New Solution Concepts and Algorithms for Equilibrium Computation and Learning in Extensive-Form Games and Beyond

Brian Hu Zhang

CMU-CS-25-121

August 2025

Computer Science Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Tuomas Sandholm (Chair)
Vincent Conitzer
J. Zico Kolter
Kevin Leyton-Brown (University of British Columbia)

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science.

This research was sponsored by the Vannevar Bush Faculty Fellowship under award number ONR N00014-23-1-2876; the National Science Foundation under award numbers RI-2312342, RI-1901403, RI-1718457, and AitF-1733556; the U.S. Army Contracting Command under award numbers W911NF2210266 and W911NF2010081; the National Institute of Health under award number A240108S001; and a CMU Computer Science Department Hans Berliner PhD Student Fellowship. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government, or any other entity.

Keywords: computational game theory, extensive-form games, online learning, optimization

For my friends and family who have supported my journey the whole way.

Abstract

Computational game theory has led to significant breakthroughs in AI dating back to the start of AI as a discipline. For example, it has been instrumental in enabling superhuman AI from recreational games such as two-player zero-sum games chess, go, and heads-up poker to multiplayer games such as six-player poker and *Hanabi*, and even in games involving human language such as *Diplomacy*. It has also empowered a growing range of non-recreational applications, such as trading, machine learning robustness and safety, negotiation, conflict resolution, mechanism (*e.g.*, auction) design, information design, security, political campaigning, and self-driving cars.

This thesis pushes the boundary on computational game theory, especially in imperfect-information sequential (extensive-form) games, which are most prevalent in practical applications both in zero-sum games and beyond. We will present new theoretical concepts and frameworks, state-of-the-art and often provably optimal algorithms for computing and learning equilibria, and new ways to apply such algorithms to real-world problems, including problems in economics such as mechanism and information design. We will also draw connections to the broader literature on optimization, yielding new and more efficient algorithms for solving variational inequalities.

The thesis contains two parts. We now highlight selected significant results from each part.

Part I covers the computation of optimal solutions to *extensive-form games*. We derive new scalable algorithms, solution concepts, and complexity results for computing *optimal equilibria* in a variety of extensive-form settings including two-player zero-sum games, team games, extensive-form correlated equilibria, and mechanism design. Though seemingly unrelated, the solutions to several of these problems turn out to rest on similar ideas surrounding the construction of a *mediator* that facilitates correlation or communication among the players, and often involve *reductions to two-player zero-sum games*—enabling the toolbox of zero-sum techniques, including those in this part, to be applied much more broadly. Among other results, we use our algorithm for hidden-role games to implement the first superhuman agent for *Fog of War chess* (also known as "dark chess"), a popular imperfect-information variant of chess in which common-knowledge closures are too large to be tackled by prior subgame-solving techniques; and to compute exact optimal strategies for several variants of *The Resistance: Avalon*.

Part II covers algorithms for *learning agents* in games, as well as novel connections to optimization more broadly speaking. The performance of a learning algorithm can be measured by the agent's *regret*. Different notions of regret can be characterized by different sets of *strategy transformation functions* ("deviations")—larger sets result in tighter notions of regret. We develop new algorithms that achieve more robust notions of equilibrium, in particular robustness to *arbitrary low-dimensional* sets of deviations, along with matching lower bounds. We also show how to use techniques from the theory of learning in games to solve *variational inequalities*—leading to, among other results, the first $\mathrm{polylog}(1/\epsilon)$-time algorithm for solving variational inequalities assuming only the *Minty property*. We also connect our earlier methods for computation of optimal equilibria to the theory of learning in games, resulting in the first algorithms for *steering* no-regret learning agents to desirable equilibria.

Acknowledgments

To my advisor, Tuomas, I could not thank you enough for all the guidance and support that you have given me throughout my PhD. Without all the illuminating conversations and late-night writing sessions, I would not be the researcher I am today. Thank you also for the freedom you have given me throughout my PhD to choose my own directions and problems. I am grateful not only for the implied trust in me as a researcher that this shows, but also for the opportunities that it has afforded me to take charge of the process of formulating problems and directions—experience that will surely help me in the future.

To my other committee members—Vince, Zico, and Kevin—thank you for your insightful feedback and assistance throughout this process. It has helped shape this thesis and my thought process during the final part of my degree. To my other amazing collaborators on the work in this thesis—Gabriele, Ioannis, Stephen, Emanuel, Emin, Luca, Federico, Andrea, Nicola, Andy, Noah, Costis, Tao, and Yiling—I am fortunate to have worked with you all, and this thesis would certainly have not been remotely possible without you. A shout out also to my other labmates during my time here: Carlos, Sidd, and Ellen. A special thanks to Gabriele, Vince, and Costis for your mentorship both past and future, and for being shining examples of excellence in all areas that I strive every day to emulate.

To the CMU Pool Club, thank you for being a source of joy for me during my time here. Thanks to Nimo, Zixin, Dean, Larry, and Peter for keeping the club running and active all these years, to Mike Shamos for being the club's faculty advisor and probably the most interesting person on campus to play 8-ball against, and to everyone else I've had the pleasure of playing over the years. I will miss this.

To my friends at CMU who I haven't mentioned yet: I will surely reflect fondly on our time together for many years to come. To my officemates in my final year—Hoai-an, Juli, Korinna, Apurva, and Alicia—I feel fortunate to have been assigned an office with you all, even if just for a single year. I will miss the hangouts, trivia nights, chess games, dinners, and deep conversations. To our Covid bubble—Peter, Magdalen, Jackson, Dorian, Steven, and Hugo—thanks for keeping me sane through Covid and after. A special thanks to Peter and Magdalen for our many F1 and Star Wars watch parties, and for allowing me to third wheel your whole relationship.

To my friends outside of CMU, especially to Jon, Gili, and Stella: thank you for being a constant source of support, for the endless conversations about, well, anything and everything, and for reminding me that life outside of Pittsburgh exists.

Last, but certainly not least, to my parents: I would not have made it this far without your everlasting love and support, for which I will be eternally grateful. I love you.

Contents

List of Figures

List of Tables

List of Algorithms

Introduction

Chapter 1

Summary

Intelligent agents—abundant across many domains such as artificial intelligence (AI), economics, e-commerce, social science, and distributed computing—need to make strategic decisions when interacting with each other. These strategic decision-making processes are fundamentally games. Hence, vital to the success of multi-agent systems is the ability to tackle game-theoretic challenges, such as computing strategies (equilibria), developing algorithms for learning in games, designing games to achieve desired outcomes (mechanism design), and steering learning agents to such outcomes.

To this end, computational game theory has led to numerous breakthroughs in AI dating back to the start of AI as a discipline. Perhaps most notably, it has led to the first superhuman-level AI agents for various games including classic two-player zero-sum games such as chess (*e.g.*, [150]), go [271], and heads-up poker [37]; multiplayer games such as multiplayer poker [39]; identical-interest games such as *Hanabi* [196]; and even expert-level play in games involving human language such as *Diplomacy* [18]. Despite these major advances, there remain many interesting problems to resolve in computational game theory, both in theory and in practice. This thesis aims to address some of these important outstanding problems.

This thesis is partitioned into two parts by general topic area. However, many of the results, even in different parts, are closely related to each other. We will point out relationships between the sections as they arise throughout this summary.

1.1 Part I: Computing Optimal Solutions to Extensive-Form Games

Unlike most efforts that focus on single-step or perfect-information games, the real world typically features agents that interact over multiple timesteps and lack perfect information about each other. In the game theory literature, these are conventionally modeled as *imperfect-information extensive-form games* (hereafter simply *extensive-form games*).

In two-player zero-sum (2p0s) games, the Nash equilibrium is the standard solution concept.

Moreover, many of the algorithms developed throughout the remainder of the thesis in particular for hidden-role games and optimal mechanism design (which we will introduce later) result in reductions *to* 2p0s games. As such, 2p0s games are important to study not only in their own right but also for their particular importance to the practical application of algorithms discussed in this thesis.

Outside two-player zero-sum games, notions of equilibrium run into issues of *non-exchangeability* and *multiplicity*: games can have more than one equilibrium, and if two players in a game independently compute (for example) Nash equilibria, their joint strategy may be arbitrarily bad unless they happen to have computed the same equilibrium, and there is no general way to pick a "best equilibrium". This phenomenon limits the ability to apply natural equilibrium concepts beyond two-player zero-sum games. This part of the thesis will focus on the following overarching question:

> *In what extensive-form games, and for what solution concepts,*
> *is there a reasonable notion of* optimal *or* best *solution, and*
> *when is such an solution efficiently computable?*

This part will give several answers to this question, both positive and negative.

1.1.1 Subgame Solving in Zero-Sum Games

As mentioned above, the most basic setting in which a reasonable notion of *optimal solution* exists is the two-player zero-sum game, where Nash equilibria are optimal solutions. Indeed, zero-sum games are the starting point of this thesis.

In computational equilibrium finding, subgame solving is the idea that one should *refine* a strategy online while playing the game, instead of playing solely from some precomputed strategy such as a policy network. As an idea, it is perhaps older than AI as a field[1.1] and has been vital in all of the breakthroughs mentioned in the first paragraph. In perfect-information settings, it has been used since the beginnings of AI as a field and has been fundamental to the success of strong agents—for example, the superhuman chess agent *Leela Chess Zero* drops to "only" human expert level without subgame solving, but is easily superhuman with subgame solving. However, the application of subgame solving to imperfect-information settings, especially in a game-theoretically safe manner, is much more challenging, and has only been studied recently [36, 43, 221]. These techniques were one of the core ingredients of the superhuman breakthroughs in no-limit Texas hold'em (NLTH) poker [37, 39].

All prior techniques for safe subgame solving suffer from a shared weakness that limits their applicability: they require reasoning about the *common-knowledge closure* of the player's current information set—that is, the smallest set of states in which it is common knowledge that the current state lies. In NLTH, this set is manageable; however, in many other games, it is not.

[1.1] For example, Alan Turing and David Champernowne wrote a chess engine *Turochamp* in 1948 using minimax search and node heuristics, which can be considered a form of subgame solving.

In Chapter 3, we develop *knowledge-limited subgame solving* (KLSS), which is the first known technique that does not have this weakness. Instead, this technique can work by only expanding the nodes that are still reachable in the game tree from the player's current information set. We demonstrate that, although our algorithm is not generally game-theoretically sound in theory, it is reasonably sound in practice and therefore can be used to create strong practical agents. We use our techniques to implement the first superhuman agent for *Fog of War chess*[1.2], a popular imperfect-information variant of chess in which common-knowledge closures are too large to be tackled by prior subgame-solving techniques.

1.1.2 Adversarial Team Games

In Chapter 4, we will discuss adversarial (*i.e.*, zero-sum) *team games*, that is, games in which there are two teams competing against each other, and their utilities are opposite (*e.g.*, one team wins and the other team loses). In such games, the key challenge is *asymmetric information* between different members of the same team—if all team members had the same information, we could simply treat the team as a single player with that common information. The most natural solution concept for adversarial team games is the *correlated team max-min equilibrium* (TMECor) [19]. TMECor represents the solution concept in which team members are allowed to discuss their strategy before the game (including flipping random coins which are not observed by the opposing team), but are not allowed to communicate once the game begins except as explicitly permitted by the game rules. We develop *parameterized algorithms* for computing TMECor in extensive-form adversarial team games. Our algorithm is based on the enumeration of the possible *common-knowledge belief states* for a given team, and its time complexity scales accordingly. In particular, our algorithm scales at $O^*((b+1)^k)$, where

- b is the branching factor,
- k is the *information complexity*, a natural parameter that we define that characterizes in a sense the extent to which the information states of different members of the same team are *asymmetric*, and
- O^* hides factors polynomial in the game size.

We show that these bounds are in a sense optimal: setting $b = O(1)$ and $k = O(n)$ can solve n-variable SAT, and the dependence on d for EFCCE and EFCE cannot be removed under ETH.

Our algorithm also enables the use of *regret minimization* for adversarial team games with the same time complexity. This is important in practice because regret minimizers are the fastest practical game solvers, and indeed we empirically show state-of-the-art performance across a wide variety of games using modern regret minimization techniques in combination with our construction.

Adversarial team games are equivalent to (timeable) two-player zero-sum games of imperfect recall. Thus, our results above can be thought of as a way of *representing the strategy space of an imperfect-recall player* with a size that is parameterized by the amount of asymmetric information.

[1.2] also known as "dark chess"

Along the way, we also make two other contributions of independent interest in Zhang et al. [306].

- We classify precisely the complexity of TMECor and TME in adversarial team games. In particular, we show that computing the TME value[1.3] is Σ_2^{P}-complete, and computing the TMECor value is Δ_2^{P}-complete[1.4], thus exhibiting a strict separation between the two problems assuming that the polynomial hierarchy does not collapse.
- We define a notion of *DAG-form decision problem* that generalizes tree-form decision problems in a way that the strategy set is still a polytope and regret minimization is still possible. Our notion of DAG-form decision problem will be used multiple times throughout the remainder of this thesis in various settings that may at first seem unrelated.

1.1.3 Hidden Team Assignments (Hidden-Role Games)

The above results about adversarial team games assume that both teams know the identities of their teammates. But this is frequently not the case in the real world. For example consider a group of computers performing a task that requires information sharing among them. Some computers may have been corrupted by an adversary, but each computer may not know which other computers have been corrupted. Inadvertently sharing information with the adversary could lead to negative outcomes. How should the computers communicate and collaborate to achieve the best possible outcome? This discussion motivates the study of *hidden-role games*, which are games in which one team does *not* know the identity of its teammates. Hidden-role games also include well-known recreational games such as *Mafia* or *The Resistance*. Although hugely popular, this class of games has lacked formal study: it was previously not even clear how to define a solution concept. A reasonable solution concept should take into account that teams need to *collaborate*, and that *intra-team communication* can be compromised due to the presence of hidden roles. For example, the team-correlated equilibrium is unsuitable because it is unreasonable to allow players to correlate with teammates whose identities they do not even know.

In Chapter 5, we develop a solution concept, which we call the *hidden-role equilibrium*, that addresses these issues. Informally, the hidden-role equilibrium is the best strategy for the uninformed team to commit to playing. We prove that hidden-role equilibria can—surprisingly—be found in polynomial time, by reduction to a two-player zero-sum game. We also show bounds on the *price of hidden roles*, which we define as the factor by which a team would benefit if it knew the identities of all the adversaries. (This is analogous to the *price of anarchy* and *price of stability* that are common quantities to study in more traditional games.) From a practical standpoint, we use our techniques to exactly solve five- and six-player variants of the popular game *The Resistance: Avalon*. Our techniques therefore draw heavily from the literature on both *team games* (such as in the previous section) and cryptography (secure multi-party computation).

[1.3] *i.e.*, deciding whether the value is at least some threshold t, up to exponentially-small error tolerance

[1.4] even with no error tolerance

1.1.4 A Unified Framework for Optimal Correlated Equilibria and Mechanism Design in Extensive-Form Games

In Chapter 6, we develop a framework that unifies a large family of game-theoretic problems for extensive-form games under a single umbrella. An incomplete list of the problems that fall under the framework is the following.

- Computing an optimal (*e.g.,* social welfare-maximizing) *correlated* equilibrium in a general-sum game: specifically, optimal *extensive-form correlated equilibrium* (EFCE) [291], *extensive-form coarse correlated equilibrium* (EFCCE) [102], or *normal-form coarse correlated equilibrium* (NFCCE) [226].[1.5]
- Computing an optimal *communication equilibrium* [109, 228]. This problem includes—among others—the popular economic settings of *optimal sequential* mechanism design and Bayesian persuasion (information design) [167] as special cases, and can be referred to as *generalized mechanism design*[1.6].

The framework is based on the observation that all of these problems essentially boil down to *representing the strategy space for a certain "meta-agent", or "mediator"*. (For example: for mechanism design, the mediator is the mechanism designer. For correlated equilibria, the mediator is the correlation device.)

We develop techniques for the problems in this framework that allowed for the first time the application of deep reinforcement learning (RL) to this large family of problems, thus allowing for the possibility of far greater scalability. Our techniques are based on reducing the general family of problems, via a Lagrangian relaxation, to a *zero-sum game*—two-player in the case of communication equilibria, and team-vs-player in the case of correlated equilibria. Thus, *if one can solve zero-sum games in extensive form, one can also compute solutions in the general framework described above, including optimal sequential generalized mechanisms.*

Our techniques here can be thought of as a generalization of the framework of *mechanism design with deep learning* first introduced by Dütting et al. [90]. Compared to that line of work, our techniques make two improvements. The first is, as above, generality: our techniques work for arbitrary communication equilibria ("generalized mechanisms") and in sequential settings. The second is an *improved Lagrangian formulation* that does not depend on a Lagrange multiplier that needs to either be known *a priori* or grow arbitrarily large. This results in a method that is significantly easier to work with, especially in a deep learning setting where a large Lagrange multiplier would correspond to the need for deep RL to achieve extremely precise results. In experiments, we show that the the improved Lagrangian formulation is critical to performance with deep RL.

[1.5] In particular, we purposefully exclude the *normal-form correlated equilibrium* [15], which is more difficult to reason about in extensive-form games and which we will discuss more in Part II.

[1.6] For example, Forges and Ray [111] uses "generalized mechanism design" to refer to the special case of a *single-stage* generalized mechanism design problem, *i.e.*, a communication equilibrium problem for a single-stage Bayesian game. Our use in this thesis is even more general, encompassing multi-stage problems as well and aligned with the setup of Forges [109] and Myerson [228]

Correlated equilibria are special in the above discussion in that the LP-based algorithm described above is not necessarily efficient. In our framework, this is justified by the fact that the mediator for correlated equilibrium has *imperfect recall*, and representing imperfect-recall decision spaces is hard in general (since solving adversarial team games is hard). The relationship between imperfect-recall mediators and correlated equilibria gives rise to a different interpretation of correlated equilibria as *generalized mechanism design with privacy constraints*: in a sense, the imperfect recall of the mediator represents precisely the constraint that the mediator cannot *leak information between players*. Indeed, we use this and other observations to write down an entire family of equilibria that include the three above bullets as special cases.

In Section 6.6, we conduct a more in-depth study of correlated equilibrium notions specifically. In particular, the parameter of *information complexity* that we discussed in the context of team games can be generalized to also capture general games and optimal correlated equilibria. In this setting, we prove the bounds $O^*((b+1)^k)$ for NFCCE, $O^*((b+d)^k)$ for EFCCE, and $O^*((bd)^k)$ for EFCE, where b and k are the same as the parameters for team games, and d is the game's depth. Like the general imperfect-recall bounds for team games, we show that these bounds are in a sense optimal: setting $b = O(1)$ and $k = O(n)$ can solve NP-complete problems, and the dependence on d for EFCCE and EFCE cannot be removed under ETH.

An overarching technical theme throughout this part is the notion and usage of a *revelation principle*. Informally, the revelation priniciple, in the broadest sense, is the notion that *unstructured* communication between agents, under the pressure of optimal strategic behavior, can often be assumed to be *structured*. In particular, even though the messages that can be sent among players (and the mediator) are usually arbitrary, the revelation principle allows us to assume without loss of generality that, at optimality, messages have meaning. Indeed, we will crucially rely on this principle, directly or indirectly, in multiple chapters in this part.

1. For hidden-role games, the revelation principle allows us to assume that messages between players amount to (encrypted) reports of private information or action recommendations.

2. For communication equilibria, the revelation principle allows us to assume that, in equilibrium, players report honest information to the mediator, and the mediator replies with action recommendations that should be followed.

3. For correlated equilibria, the revelation principle allows us to assume that the mediator's messages to the players are action recommendations.

In all three cases above, the revelation principle is crucial in developing efficient algorithms, and whether or not a version of the revelation principle holds will often precisely demarcate the line between what is computable efficiently and what is not.

1.2 Part II: Learning Agents in Games, Correlated Equilibria, and Optimization

So far, we have discussed a *centralized* model of computational game theory: there is a fixed, known game, and the algorithmic task is for a central authority to, given the game, compute an equilibrium. But, in many practical settings, it is unrealistic to assume the existence of such a central authority; instead, the players are simply playing the game repeatedly and learning from their observations, rewards, and experience. The players are not necessarily even aware of the existence of other players in the game, and they adapt their response only in respose to the rewards that they receive as a function of their play. This setting is known as *uncoupled learning*, and one frontier of research in computational game theory studies how such agents behave over time.

The performance of a learning algorithm faced with a strategy set $\mathcal{X} \subset \mathbb{R}^d$ can be measured by its *regret*—that is, the improvement in reward that the agent would have experienced had it played other strategies instead. Different notions of regret therefore can be characterized by different sets of *deviations* $\Phi \subseteq \mathcal{X}^{\mathcal{X}}$ mapping the strategy played by the agent to other strategies that could have been more profitable—accommodating larger sets Φ results in more robust notions of regret. Different notions of regret correspond to notions of *correlated equilibrium* in games: if agents play algorithms achieving *no* Φ*-regret*, then they will converge the set of Φ*-equilibria*.

This part will cover new algorithms for learning agents in games, both better algorithms for the learning agents themselves (*i.e.*, algorithms achieving better notions of Φ-regret), as well as novel algorithms for *steering* the dynamics of learning agents to desirable outcomes. It will also explore the deep connection between learning in games and optimization more broadly, leading to novel algorithms for solving *variational inequalities*. The overarching questions for this section is the following:

How can we design better algorithms for learning agents in games?
What implications do the general techniques involved in the proofs
of such results have for optimization problems beyond games?

Learning algorithms in games are fundamental to this thesis in several ways. In Part I, we were interested in learning algorithms primarily *as tools* for equilibrium computation, because they are the fastest-known algorithms both in theory[1.7] and in practice for solving games. In this part, learning algorithms will play a more fundamental role: we are interested in learning algorithms *for their own sake*, that is, we seek to answer questions regarding what happens when learning agents play games.

In a further departure from the previous part, this section will sometimes tackle *convex games*, which contain extensive-form games as a special case but are significantly more general. (We will, however, still mention specific applications to extensive-form games where these are relevant.) We will also generally not be interested in computing *optimal* equilibria in this part: indeed, optimal correlated equilibria in general cannot be reached by independent learning algorithms, and moreover, as we have seen in the previous part, they are often hard to compute.

[1.7] at least, with respect to dependence on the game size

1.2.1 Steering Learning Agents to Optimal Equilibria

Having established algorithms to compute optimal equilibria in general-sum games, and having established that independent learning algorithms in general cannot hope to find such equilibria, we now ask the question of how we can ever get learning agents to play desirable equilibria.

In Chapter 7, we ask whether a mediator can *steer* players to an arbitrary equilibrium of its choice. We show that a mediator with the power to provide *payments* to the players can accomplish such steering, and we show that the ability of the mediator to succeed in steering depends on how much budget the mediator has as a function of the time. If the mediator's budget is constant, no steering is possible; if the mediator's budget grows linearly with time, the mediator can trivially steer the players toward any behavior by simply providing large enough payments. The case of *sublinearly* growing payments is therefore the most interesting case, and indeed we show that, with reasonable assumptions, a mediator can steer any no-regret players toward any equilibrium of its choice with only a total budget that increases sublinearly with the time horizon.

1.2.2 Φ-Regret and Φ-Equilibria in General Convex Domains

In Chapter 8, we develop general Φ-regret minimization algorithms for arbitrary convex sets and arbitrary sets Φ. In particular, we develop a general Φ-regret minimization algorithm that achieves average regret ϵ after $\mathsf{poly}(d, k)/\epsilon^2$ rounds of play, where d is the dimension of the strategy set $\mathcal{X}$, and k is the dimension of the deviation set Φ. We also give an algorithm for *computing* (ϵ-approximate) Φ-equilibria in time $\mathsf{poly}(d, k, \log(1/\epsilon))$.

Technically, our results build on earlier upper bounds for the special case where Φ contains only linear functions [96]. At the heart of our approach is a polynomial-time algorithm for computing an *expected fixed point* of any $\phi : \mathcal{X} \to \mathcal{X}$—that is, a distribution $\mu \in \Delta(\mathcal{X})$ such that $\mathbb{E}_{\boldsymbol{x}\sim\mu}[\phi(\boldsymbol{x}) - \boldsymbol{x}] \approx 0$—based on the seminal *ellipsoid against hope (EAH)* algorithm of Papadimitriou and Roughgarden [237]. This definition of fixed point and efficient algorithm allows us to circumvent the (PPAD-hard) fixed-point computation inherent to most past attempts at Φ-regret minimization, such as Gordon et al. [132]. In particular, our algorithm for computing Φ-equilibria in time $\mathsf{poly}(d, k, \log(1/\epsilon))$ is based on executing EAH in a nested fashion—each step of EAH itself being implemented by invoking a separate call to EAH!

In Section 8.8, we will discuss the special case when the game is extensive form and Φ contains only linear functions. In this case, we develop a perhaps-surprising relationship between the set of linear functions and the set of *untimed communication deviations*, which intuitively resemble the set of deviations in a communication equilibrium (as in Section 1.1.4), except that players are not constrained to send a single message at every timestep. We show that the set of linear deviations corresponds *exactly* to the set of untimed communication deviations, and we use this connection to develop faster algorithms for regret minimization in this special case, based on DAG regret minimization as established in Section 1.1.2.

In Section 8.10, we establish nearly-matching information-theoretic *lower bounds* on Φ-regret minimization. In particular, we show that achieving average swap regret ϵ against an oblivious adversary in a general convex game (in fact, even an extensive-form game) requires at least

$\Omega(\min\{k, \exp(d^{1/14}), \exp(\epsilon^{-1/6})\})$ rounds, providing a nearly-matching lower bound. Technically, our approach builds on earlier lower bounds [71, 242] that were developed for the special case when the game is normal form (*i.e.*, $\mathcal{X}$ is the simplex $\Delta(d)$), and Φ consists of all functions $\mathcal{X} \to \mathcal{X}$ ("swap regret").

1.2.3 Game Theory and Variational Inequalities

We finally investigate a connection between correlated equilibria in games and *variational inequalities* (VIs). Informally, variational inequalities (which will be formally defined in Part II) are a broad class of optimization problems that capture, among other things, first-order function optimization, fixed points of arbitrary functions, and Nash equilibria in games. The connection between games and VIs allows us to adapt several ideas from computational game theory to the more general setting of VIs, leading to new results for that setting.

The expressivity of the general VI problem comes, of course, at the cost of computational hardness. As a result, most research has focused on carving out specific subclasses that elude those intractability barriers. A classical property that goes back to the 1960s is the *Minty condition*, which postulates that the *Minty* VI problem—the weak dual of the VI problem—admits a solution.

In Chapter 9 we establish the first polynomial-time algorithm—that is, with complexity growing polynomially in the dimension d and $\log(1/\epsilon)$—for solving VIs for Lipschitz continuous mappings under the Minty condition. Prior approaches either incurred an exponentially worse dependence on $1/\epsilon$ (and other natural parameters of the problem) or made overly restrictive assumptions—such as strong monotonicity. To do so, we introduce a new variant of the ellipsoid algorithm wherein separating hyperplanes are obtained after taking a gradient descent step from the center of the ellipsoid. It succeeds even though the set of VI solutions can be nonconvex and not full-dimensional. Moreover, when our algorithm is applied to an instance with no MVI solution and fails to identify an VI solution, it produces a succinct *certificate of Minty infeasibility*. We also show that deciding whether the Minty condition holds is coNP-complete, thereby establishing that the disjunction of those two problems—computing VI solutions and ascertaining Minty infeasibility—is polynomial-time solvable even though each problem is individually intractable.

We provide several extensions and new applications of our main results. Specifically, we obtain the first polynomial-time algorithms for i) solving monotone VIs, ii) globally minimizing a (potentially nonsmooth) *quasar-convex* function, iii) computing Nash equilibria in multi-player *harmonic games*, and iv) computing *either* a Nash equilibrium *or* a *strict* coarse correlated equilibrium in two-player general-sum concave games.

In Section 9.6, we take a deeper look at the aforementioned certificates of Minty infeasibility. In particular, these take the form of solutions to an *in-expectation* version of the VI problem, which we coin the *expected VI* (EVI) problem. These EVIs, informally, are to VIs what correlated equilibria are to Nash equilibria in games. Like correlated equilibria, we show that the expected VI problem can be parameterized by a set of deviations Φ. By extending the aforementioned ellipsoid against hope and Φ-regret frameworks beyond game theory, we show that the Φ-expected VI problem can be solved efficiently in two ways, at least when Φ consists only of linear functions: they can be *learned* by a Φ-regret-minimizing algorithm in time $\mathsf{poly}(d)/\epsilon^2$, or computed directly using a generalization of the EAH algorithm in time poly$(d, 1/\epsilon)$. We also employ our framework to capture and generalize several existing disparate results, including from settings such as smooth games, and games with coupled constraints or nonconcave utilities, and to develop a novel solution concept for games which we call the *anonymous correlated equilibrium* and which lies between linear correlated equilibria and Nash equilibria in the inclusion hierarchy.

1.3 Research Covered in This Thesis

This thesis includes work that has appeared in the literature and that has been completed in collaboration with others, as follows. (*: Equal contribution; $\alpha\beta$: alphabetial ordering of authors)

Chapter 3:

- Brian Hu Zhang and Tuomas Sandholm. Subgame solving without common knowledge. *Neural Information Processing Systems (NeurIPS)*, 2021 [301]
- Brian Hu Zhang and Tuomas Sandholm. General-purpose search techniques without common knowledge for imperfect-information games, and application to superhuman Fog of War chess. *Under submission*, 2025 [304]

Chapter 4:

- Luca Carminati*, Brian Hu Zhang*, Federico Cacciamani, Junkang Li, Gabriele Farina, Nicola Gatti, and Tuomas Sandholm. Efficient representations for team and imperfect-recall equilibrium computation. *Under submission*, 2025 [55] (Subsumes Zhang et al. [306] and Zhang and Sandholm [303])

Chapter 5:

- Luca Carminati*, Brian Hu Zhang*, Gabriele Farina, Nicola Gatti, and Tuomas Sandholm. Hidden-role games: Equilibrium concepts and computation. *ACM Conference on Economics and Computation (EC)*, 2024 [54]

Chapter 6:

- Brian Hu Zhang and Tuomas Sandholm. Polynomial-time optimal equilibria with a mediator in extensive-form games. *Neural Information Processing Systems (NeurIPS)*, 2022 [302]
- Brian Hu Zhang*, Gabriele Farina*, Ioannis Anagnostides, Federico Cacciamani, Stephen McAleer, Andreas Haupt, Andrea Celli, Nicola Gatti, Vincent Conitzer, and Tuomas Sandholm. Computing optimal equilibria and mechanisms via learning in zero-sum extensive-form games. *Neural Information Processing Systems (NeurIPS)*, 2023 [316]
- Brian Hu Zhang, Gabriele Farina, Andrea Celli, and Tuomas Sandholm. Optimal correlated equilibria in general-sum extensive-form games: Fixed-parameter algorithms, hardness, and two-sided column-generation. *Mathematics of Operations Research*, 2025 [308] (Subsumes Zhang et al. [305])

Chapter 7:

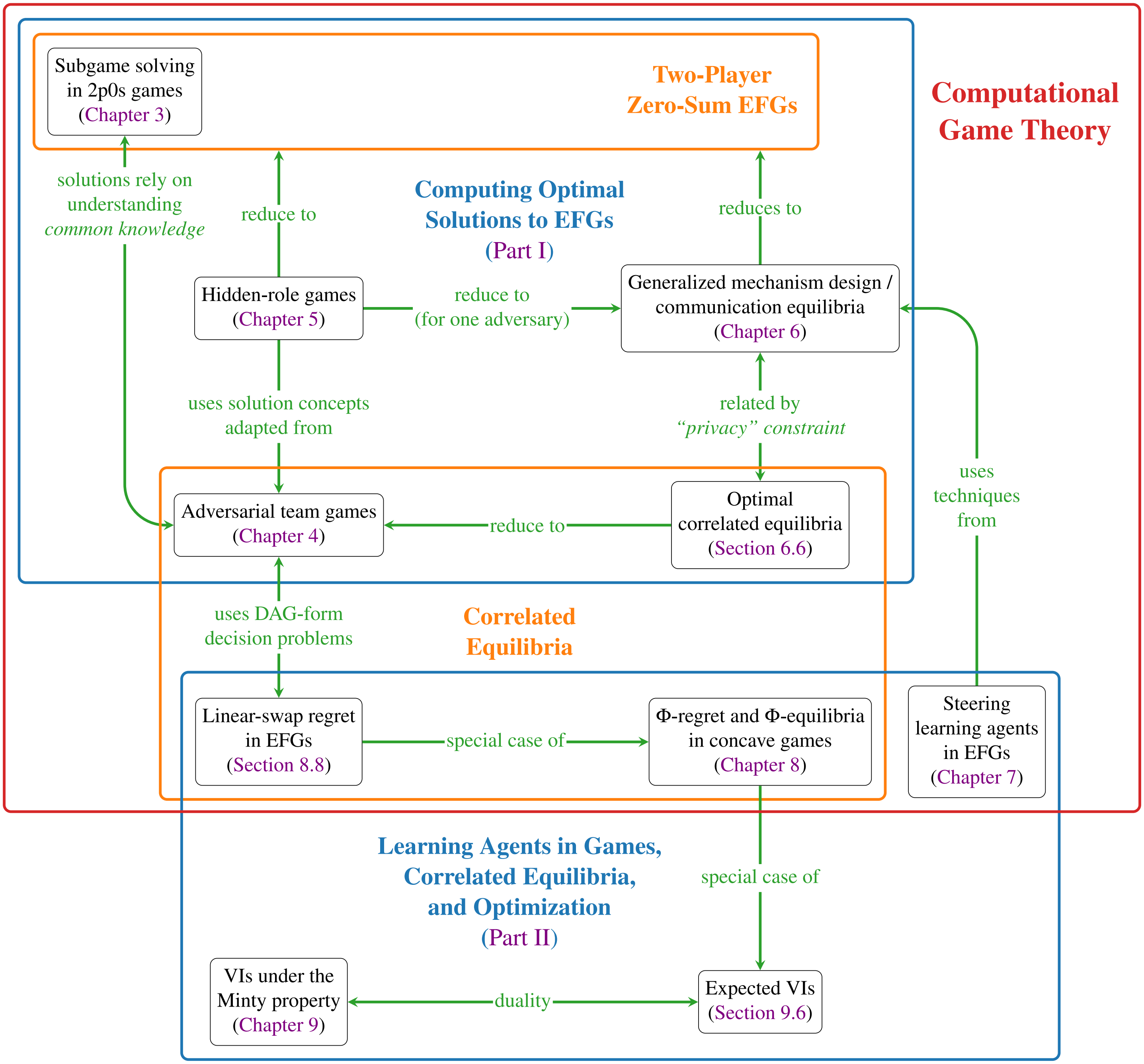


Figure 1.1: *Some of the relationships between the various chapters of this thesis. "EFGs": Extensive-form games; "VIs": Variational inequalities.*

- Brian Hu Zhang*, Gabriele Farina*, Ioannis Anagnostides, Federico Cacciamani, Stephen Marcus McAleer, Andreas Alexander Haupt, Andrea Celli, Nicola Gatti, Vincent Conitzer, and Tuomas Sandholm. Steering no-regret learners to optimal equilibria. *ACM Conference on Economics and Computation (EC)*, 2024 [318]
- Brian Hu Zhang*, Tao Lin*, Yiling Chen, and Tuomas Sandholm. Learning a game by paying the agents. *arXiv:2503.01976*, 2025 [321]

Chapter 8:

- Brian Hu Zhang*, Ioannis Anagnostides*, Emanuel Tewolde, Ratip Emin Berker, Gabriele Farina, Vincent Conitzer, and Tuomas Sandholm. Learning and computation of Φ-equilibria at the frontier of tractability. *ACM Conference on Economics and Computation (EC)*, 2025 [320]
- Brian Hu Zhang*, Ioannis Anagnostides*, Gabriele Farina, and Tuomas Sandholm. Efficient Φ-regret minimization with low-degree swap deviations in extensive-form games. *Neural Information Processing Systems (NeurIPS)*, 2024 [317]
- ($\alpha\beta$) Constantinos Daskalakis, Gabriele Farina, Noah Golowich, Tuomas Sandholm, and Brian Hu Zhang. A lower bound on swap regret in extensive-form games. *arXiv:2406.13116*, 2024 [80]
- Brian Hu Zhang, Gabriele Farina, and Tuomas Sandholm. Mediator interpretation and faster learning algorithms for linear correlated equilibria in general sequential games. *International Conference on Learning Representations (ICLR)*, 2024 [307]

Chapter 9:

- ($\alpha\beta$) Ioannis Anagnostides, Gabriele Farina, Tuomas Sandholm, and Brian Hu Zhang. A polynomial-time algorithm for variational inequalities under the Minty condition. *arXiv:2504.03432*, 2025 [13]
- Brian Hu Zhang*, Ioannis Anagnostides*, Emanuel Tewolde, Ratip Emin Berker, Gabriele Farina, Vincent Conitzer, and Tuomas Sandholm. Expected variational inequalities. *International Conference on Machine Learning (ICML)*, 2025 [319] (A part of this paper also appears in Section 8.8.)

Chapter 2

Preliminaries

Here, we introduce the various pieces of background information that will be repeatedly referenced throughout this thesis. Background information that is more specialized to a single section of the thesis is deferred to that section.

2.1 General Notation and Terminology

Unless otherwise stated, we will use the following notation:

- Vectors $\boldsymbol{x} \in \mathbb{R}^n$ will be in italic boldface, as will generic indices into such vectors, which will be denoted either $\boldsymbol{x}(i)$ or $\boldsymbol{x}_i$. Matrices will be in non-italic boldface, *e.g.*, $\mathbf{A}$.
- $\|\cdot\|$ denotes the ℓ_2-norm of a vector and the Frobenius norm of a matrix.
- $\mathcal{B}_r(\boldsymbol{x})$ is the (closed) Euclidean ball centered at $\boldsymbol{x}$ with radius $r > 0$.
- $f \lesssim g$ means $f = O(g)$. Similarly, $f \gtrsim g$ means $f = \Omega(g)$, and $f \sim g$ means $f = \Theta(g)$.
- $\circ$ denotes element-wise multiplication of vectors or matrices.
- If A, B are sets then A^B is the set of functions $f : B \to A$, and 2^A is the power set of A.
- $\Delta(S)$ is the set of *finite-support* probability distributions on set S. In particular, for finite S we have $\Delta(S) := \{\boldsymbol{x} \in \mathbb{R}^S_{\geq 0} : \sum_{s \in S} \boldsymbol{x}(s) = 1\}$. If $\boldsymbol{x} \in \Delta(S)$ then $\operatorname{supp} \boldsymbol{x} \subseteq S$ denotes the support of $\boldsymbol{x}$. We will never need to deal with probability distributions of infinite support.
- $\delta(s)$ for $s \in S$ is the Dirac delta distribution on s, that is, the distribution that determinstically returns s.
- If $f : B \to A$ is a function and $\mu \in \Delta(B)$ is a probability measure, then $f(\mu)$ is the pushforward measure, that is, $f(\mu) \in \Delta(A)$ is the distribution given by sampling $b \sim \mu$ and returning $f(b) \in A$.

- If $\mu \in \Delta(A)$ and $\nu \in \Delta(B)$, then $\mu \times \nu \in \Delta(A) \times \Delta(B)$ is the product distribution, that is, the distribuition for which sampling $A \times B \ni (x, y) \sim \mu \times \nu$ results in x and y being independent variables with marginals μ and ν.
- $\operatorname{conv} S$ denotes the convex hull of a set S.
- $\tilde{O}, \tilde{\Omega}, \tilde{\Theta}$ hide logarithmic factors. That is, $f = \tilde{O}(g)$ if $f = O(g \log^k g)$; $f = \tilde{\Omega}(g)$ if $f = \Omega(g \log^{-k} g)$ (where in both cases k is an absolute constant), and $f = \tilde{\Theta}(g)$ if $f = \tilde{O}(g)$ and $f = \tilde{\Omega}(g)$.
- O_x (and similar symbols, such as Ω_x or $\tilde{\Theta}_x$) is used to hide dependence on all parameters except x.
- For any set S, $\mathrm{Id} : S \to S$ is the identity function.
- $\mathbb{1}\{b\}$ is the indicator of the condition b
- $[n] = \{1, \dots, n\}$.
- $[x]^+ = \max(x, 0)$.

2.2 Imperfect-Information Extensive-Form Games

Imperfect-information extensive-form games are the focus of the majority of this thesis. They model games in which the players interact *over time*, and may have *imperfect information* about each others' actions.

A finite extensive-form game (hereafter simply *game*) Γ with n *players* (also known as *agents*) is modeled via a *game tree*. The game tree consists of a tree of *histories*, or *nodes*, $\mathcal{H}$, rooted at a *root history* $\varnothing \in \mathcal{H}$. Histories $h \in \mathcal{H}$ represent states of the game. Leaves of the tree are called *terminal nodes*, and the set of such leaves is $\mathcal{Z}$. Edges are identified by *actions* a, and the set of actions legal at node h is denoted $\mathcal{A}(h)$. The child of h reached by following action a is denoted ha, so that a history h may also be denoted as $h = a_1 a_2 \dots a_\ell$ where $a_1, \dots, a_\ell \in \mathcal{A}$ are the actions leading from $\varnothing$ to h. The *branching factor* of a game, often denoted b, is the maximum number of legal actions at any node. That is, $b = \max_{h \in \mathcal{H}} |\mathcal{A}(h)|$. Each player $i \in [n]$ has a *utility function* $u_i : \mathcal{Z} \to \mathbb{R}$ that indicates how much reward player i receives if the game ends in node $z \in \mathcal{Z}$.

The game tree induces a natural ordering $\preceq$ on sets of nodes: we will write $S \preceq S'$ if there are histories $h \in S, h' \in S'$ such that h' is a descendant of h. If either S or S' is a singleton, we will omit the braces: for example, $h \preceq h'$ denotes that h' is a descendant of h. We will use $|h|$ to denote the depth of history h: that is, $|\varnothing| = 0$ and $|ha| = |h| + 1$. The depth of game Γ is the maximum depth of any history.

At each (non-terminal) history $h \in \mathcal{H} \setminus \mathcal{Z}$, either a player, or *chance* (also known as *nature*) has the right to select the action a that is taken at h. That is, the set of nonterminal nodes $\mathcal{H} \setminus \mathcal{Z}$ is partitioned as $\mathcal{H} \setminus \mathcal{Z} = \mathcal{H}_\mathsf{C} \sqcup \mathcal{H}_1 \sqcup \dots \sqcup \mathcal{H}_n$, where $\mathcal{H}_i$ is the set of nodes at which player i acts, and player C is chance (or *nature*). Chance plays according to a fixed distribution: for every history $h \in \mathcal{H}_\mathsf{C}$, there is a probability distribution $p(\cdot|h) \in \Delta(\mathcal{A}(h))$ denoting how chance selects its action.

Imperfect information. To model imperfect information, each player $i \in [n]$ has a partition $\mathcal{I}_i$ of its set of decision points $\mathcal{H}_i$ into *information sets* (or *infosets* for short). Any two nodes h, h' in the same infoset $I \in \mathcal{I}_i$ are indistinguishable to player i, who must therefore play the same action at each infoset. In particular, this implies that the set of legal actions must depend only on the infoset and not on the action, that is, $\mathcal{A}(h) = \mathcal{A}(h') =: \mathcal{A}(I)$. For simplicity, we will assume unless otherwise stated that actions at different infosets have different labels—that is, $\mathcal{A}(I) \cap \mathcal{A}(I') = \varnothing$ for $I \neq I'$.

The extensive-form setup above does not natively allow for simultaneous moves. This follows the standard notational convention for extensive-form games. However, simultaneous moves can be simulated using imperfect information: if player j moves after player i and does not observe player i's move, then the two players are in effect moving simultaneously.

Timeability. An extensive-form game is *timeable* if any path from the root to any node in the same infoset has the same length (*i.e.* all histories belonging to the same infoset have the same depth). Formally, the game is is timeable if for every infoset $I \in \mathcal{I}$ and every $h, h' \in I$, we have $|h| = |h'|$.

Timeability is a natural condition that is equivalent to stating that there is a shared *clock* that is always visible to all the players (the depth of a node h corresponds to the time indicated by the clock). It is thus a very natural condition on the information structure of a game.

This thesis will often restrict attention to timeable games; for example, all the results in Part I work only for timeable games. Moreover, even the results in Part II that do not explicitly assume timeability are already interesting for timeable games; therefore, unless otherwise explicitly indicated by the text, or unless the reader is explicitly interested in games that are not timeable, it is safe for purposes of reading this thesis to assume that all games discussed in this thesis are timeable.

Perfect recall. Perfect recall is, intuitively, the condition that no player ever forgets any information. At a history $h \in \mathcal{H}$, the *sequence* $\sigma_i(h)$ of player i is the list of information sets $I \in \mathcal{I}_i$ encountered by player i on the path to h, and actions taken at those information sets, not including at h itself. Intuitively, $\sigma_i(h)$ indicates the things that player i has observed so far. We say that a player i has *perfect recall* if nodes with different sequences always belong to different infosets. Formally, player i has perfect recall if for every infoset $I \in \mathcal{I}_i$, every history $h \in I$ has the same sequence, which we will denote $\sigma_i(I)$ and call the *parent sequence* of I. The game Γ has perfect recall if all of its players do. We will denote by Σ_i the set of sequences of a player i.

Like timeability, perfect recall is a natural condition on players' actions. However, it will often be useful to us to discuss games without perfect recall. In particular, imperfect-recall games will be used in this thesis to model *teams* of players with aligned interests (identical utility functions) but asymmetric information, in the following intuitive fashion: a team of players $T \subseteq [n]$ is modeled as a single player, the *team coordinator*. When play reaches a node $h \in \mathcal{H}_i$ for any team player $i \in T$, the team coordinator forgets all information except player i's information, and then plays an action for player i. In this fashion, any strategy implementable by the team coordinator is also implementable by the players on the team working with their own private information. This intuition will be used repeatedly throughout Part I, most prominently in Chapter 4 which is specifically about team games.

Strategies. A *pure strategy* of player i is a choice of one action per infoset of player i. That is, a pure strategy is a selection $\pi_i \in \bigtimes_{I \in \mathcal{I}_i} \mathcal{A}(I)$, where $\pi_i(I) \in \mathcal{A}(I)$ indicates the action that player i plays at infoset I. Working with pure strategies in the above form is cumbersome; instead, a more convenient representation is the *realization form* of a pure strategy, which is the vector $\boldsymbol{x}_i \in \{0, 1\}^{\mathcal{Z}}$ where $\boldsymbol{x}_i(z) = 1$ if and only if the player plays all the actions on the $\varnothing \to z$ path, that is,

$$\boldsymbol{x}_i(z) := \prod_{\substack{h \in I \in \mathcal{I}_i \\ ha \preceq z}} \mathbb{1}\{\pi_i(I) = a\}.$$

We will use Π_i to denote the set of realization-form pure strategies.

A *mixed strategy* of player i is a distribution $\mu_i \in \Delta(\Pi_i)$. In many cases, we will only care about the realization form of a mixed strategy, which is simply defined to be $\mathbb{E}_{\boldsymbol{x}_i \sim \mu_i} \boldsymbol{x}_i$. The set of realization-form mixed strategies is hence $\mathcal{X}_i := \operatorname{conv} \Pi_i$. A mixed strategy is *behavioral* if its action choices at different information sets are independent.

A *correlated strategy profile* (or simply *correlated profile*) is a distribution $\mu \in \Delta(\Pi_1 \times \cdots \times \Pi_n)$. If μ factors as a product distribution $\mu = (\mu_1, \dots, \mu_n) \in \Delta(\Pi_1) \times \cdots \times \Delta(\Pi_n)$, we will drop the word *correlated* and simply call μ a *strategy profile* or *profile*. If the word *correlated* is not used, all profiles are assumed to be uncorrelated. For uncorrelated profiles, we will usually circumvent writing the distribution at all, by experessing each player's mixed strategy μ_i as a realization-form mixed strategy and thus expressing μ as a tuple $\boldsymbol{x} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$.

Every profile induces a distribution over terminal nodes, that results from sampling a pure profile $\boldsymbol{x} \sim \mu$ and following those actions through the game, sampling chance actions where needed. We will use $z \sim \mu$ (or $z \sim \boldsymbol{x}$) to denote a sample from this distribution. The *expected value* of player i under profile μ, denoted $u_i(\mu)$, is defined in the natural manner:

$$u_i(\mu) := \mathop{\mathbb{E}}_{z \sim \mu} u_i(z) = \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mu} \sum_{z \in \mathcal{Z}} u_i(z) p(z) \prod_{i \in [n]} \boldsymbol{x}_i(z)$$

where

$$p(z) := \prod_{h \in \mathcal{H}_{\mathrm{C}}, ha \preceq z} p(a|h)$$

is the probability that chance plays all the actions on the path to z. In particular, it is critical to note that players' utilities depend only on the realization forms of their strategies, and that the dependence is linear in $\boldsymbol{x}_i$.

Multiple strategies can have the same realization form. If so, we will call those strategies *(realization-)equivalent*. Unless otherwise stated, since it is not relevant for utility, we will not distinguish between realization-equivalent strategies. *Kuhn's theorem* [188] guarantees that, for players with perfect recall, every mixed strategy is realization-equivalent to a behavioral strategy, and thus for perfect-recall players it is *usually* without loss of generality to work with behavioral strategies (although we will see in Part II that it is not *always* the case!)

Two-player zero-sum games. A game is *two-player zero-sum*, if there are two players (which will always be denoted ▲ and ▼), and $u_{\blacktriangle} = -u_{\blacktriangledown}$. In this case, we will generally use the notation $u := u_{\blacktriangle}$, $\Pi = \Pi_{\blacktriangle}$, and $\mathcal{Y} = \Pi_{\blacktriangledown}$.

Normal-form games. A *normal-form game* is the special case of an extensive-form game in which each player takes a single action simultaneously, and then the game ends. In normal-form games, the (realization-form, mixed) strategy sets $\mathcal{X}_i$ are isomorphic to simplices, $\mathcal{X}_i = \Delta(\mathcal{A}_i)$. Normal-form games are usually much more simple than extensive-form games, and most of the results in this paper are trivial in the normal-form case. Moreover, any extensive-form game is equivalent to a normal-form game whose action set is equal to the pure strategy set of the extensive-form game. However, this equivalence incurs an exponential blowup, since $|\Pi_i|$ is in the general case exponential in $|\mathcal{H}|$. Thus, for computational purposes, converting a game to normal form should be avoided.

2.2.1 Equilibria

For our purposes, to *solve* a game will mean to find an *equilibrium* of it, for some *equilibrium concept* of interest. Here we identify some equilibrium concepts that we will use throughout this thesis.

The *Nash equilibrium* [230] is the oldest and best-known notion of equilibrium for general games. An ϵ-Nash equilibrium is an uncorrelated strategy profile $\boldsymbol{x} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$ such that no player can improve by more than ϵ using any unilateral deviation:

$$u_i(\boldsymbol{x}_i', \boldsymbol{x}_{-i}) \le u_i(\boldsymbol{x}_i, \boldsymbol{x}_{-i}) + \epsilon$$

for every $i \in [n]$ and $\boldsymbol{x}_i' \in \Pi_i$. A *Nash equilibrium* is a 0-Nash equilibrium. Every game has a Nash equilibrium in mixed strategies [230].

Throughout this thesis, in various places we will also be interested in various notions of *correlated equilibria*. In the greatest possible generality, a notion of correlated equilibrium is defined by a tuple of sets of transformations $\Phi = (\Phi_1, \dots, \Phi_n)$, where $\Phi_i \subseteq \mathcal{X}_i^{\Pi_i}$ is a set of transformations of player i's strategies. Then an ϵ-approximate Φ-equilibrium is a *correlated* profile for which

$$\mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mu} [u_i(\phi_i(\boldsymbol{x}_i), \boldsymbol{x}_{-i}) - u_i(\boldsymbol{x}_i, \boldsymbol{x}_{-i})] \le \epsilon$$

for every $i \in [n]$ and $\boldsymbol{x}_i' \in \Pi_i$. Two extremes of this definition are the *normal-form coarse-correlated equilibrium* (NFCCE), for which Φ_i is the set of all constant transformations $\{\phi_{\boldsymbol{x}_i^*} :$

$\boldsymbol{x}_i \mapsto \boldsymbol{x}_i^* \mid \boldsymbol{x}_i^* \in \Pi_i\}$, and the *normal-form correlated equilibrium* (NFCE), for which $\Phi_i = \mathcal{X}_i^{\Pi_i}$ is the set of all possible transformations. Other common notions include the *extensive-form correlated equilibrium* (EFCE) [291] and *extensive-form coarse correlated equilibrium* [102], which will be defined precisely in Section 6.6.

In zero-sum games, all the notions of correlated equilibria collapse to Nash equilibria[2.1], and the Nash equilibria are precisely the saddle-point solutions $(\boldsymbol{x}, \boldsymbol{y})$ to the convex bilinear saddle-point problem

$$\max_{\boldsymbol{x}\in\mathcal{X}} \min_{\boldsymbol{y}\in\mathcal{Y}} u(\boldsymbol{x}, \boldsymbol{y}) = \max_{\boldsymbol{x}\in\mathcal{X}} \min_{\boldsymbol{y}\in\mathcal{Y}} \sum_{z\in\mathcal{Z}} p(z)u(z)\boldsymbol{x}(z)\boldsymbol{y}(z) = \max_{\boldsymbol{x}\in\mathcal{X}} \min_{\boldsymbol{y}\in\mathcal{Y}} \boldsymbol{x}^\top \mathbf{A}\boldsymbol{y} \tag{2.1}$$

where $p(z)$ is again the probability that chance plays all actions on the path to z, and the matrix $\mathbf{A}$ is defined by

$$\mathbf{A}(i, j) = \sum_{z\in\mathcal{Z}:\sigma_{\blacktriangle}(z)=i,\sigma_{\blacktriangledown}(z)=j} p(z)u(z).$$

We will call the saddle-point value of (2.1) the *equilibrium value* of Γ, and denote it u^*. Nash equilibria in zero-sum games are hence *exchangeable*: if $(\boldsymbol{x}^1, \boldsymbol{y}^1)$ and $(\boldsymbol{x}^2, \boldsymbol{y}^2)$ are Nash equilibria, then so are $(\boldsymbol{x}^1, \boldsymbol{y}^2)$ and $(\boldsymbol{x}^2, \boldsymbol{y}^1)$.

2.2.2 Tree-Form Decision Making

It will be convenient at various points in this thesis to abstract away the majority of a game and focus solely on the decision problem faced by a single player. When this happens, we will generally omit the subscript i; for example, $\boldsymbol{x}$ will denote a generic strategy for the player. For a perfect-recall player, this decision problem can be described as a *tree-form decision problem*. A tree-form decision problem consists of a tree of nodes $\mathcal{T}$, that are each one of two types:

- *decision points* $j \in \mathcal{J}$, at which the player must select an action $a \in A(j)$, and
- *observation points* $\sigma \in \Sigma$, at which the player makes an observation.

For a perfect-recall player in an extensive-form game, the decision and observation points correspond respectively to the information sets and sequences of that player. Unless otherwise stated, we will assume that decision and observation points alternate, and that the root $\varnothing$ is an observation point—both of these are without loss of generality. The observation point child of j reached by taking action a is denoted ja, and the parent of j is denoted p_j. The set of children of σ is denoted C_σ. For notational simplicity, when $\boldsymbol{x} \in \mathbb{R}^\Sigma$ is any vector indexed by observation points and j is a decision point, we will use $\boldsymbol{x}(j*) \in \mathbb{R}^{\mathcal{A}(j)}$ to denote the subvector of $\boldsymbol{x}$ indexed only by the children of j.

We now define strategies in tree-form decision problems analogously to strategies in games. A *pure strategy* is a choice of one action at every decision point. The *sequence form* of a pure strategy is the vector $\boldsymbol{x} \in \Pi$ indexed by sequences $\sigma \in \Sigma$, for which $\boldsymbol{x}_i(\sigma) = 1$ if and only if the

[2.1] In particular, one can show that, for any ϵ-NFCCE, the product distribution with the same marginals is a 2ϵ-Nash equilibrium player plays all actions on the $\varnothing \to \sigma$ path in $\mathcal{T}$. The sequence-form mixed strategies are then, once again, the convex hull of Π. Conveniently, the sequence-form mixed strategies are precisely the strategies obeying a natural family of linear constraints [177, 253, 290]:

$$\mathcal{X} = \left\{ \boldsymbol{x} \in \mathbb{R}^{\mathcal{S}}_{\geq 0} \;\middle|\; \boldsymbol{x}(\varnothing) = 1, \quad \boldsymbol{x}(p_j) = \sum_{a \in A(j)} \boldsymbol{x}(ja) \quad \forall j \in \Sigma \right\}.$$

Clearly, the sequence-form and realization-form representations are equivalent: given a sequence-form vector $\boldsymbol{x}_i$ for a player i, one recovers the realization form by $\boldsymbol{x}_i(z) := \boldsymbol{x}_i(\sigma_i(z))$. Which we choose to use will depend on which is most convenient. In both cases we will denote the set of pure strategies by Π_i.

2.3 No-Regret Learning

No-regret learning is a popular framework for decision making in repeated settings. As we will see, algorithms based on no-regret learning are the most popular and fastest practical algorithms for equilibrium computation. In this section we will discuss only algorithms for so-called *external* regret minimization in extensive-form games; we defer the extension to the more general notion of Φ-*regret* to Part II.

A decision maker is faced with the following interation with an adversary. There is a convex compact *strategy set* $\mathcal{X}$, which will often be the set of mixed sequence-form strategies of some tree-form decision problem. The interaction lasts for T timesteps. At each timestep t, the decision maker selects a point $\boldsymbol{x}^t \in \mathcal{X}$. The adversary, observing $\boldsymbol{x}^t$, selects a utility vector $\boldsymbol{u}^t \in \mathbb{R}^n$ such that $\langle \boldsymbol{u}^t, \boldsymbol{x} \rangle \in [-1, 1]$ for all $\boldsymbol{x} \in \mathcal{X}$. After T timesteps, the *(averaged, external) regret* is defined as

$$\textsc{Reg}(T) := \max_{\boldsymbol{x} \in \mathcal{X}} \frac{1}{T} \sum_{t=1}^{T} \langle \boldsymbol{u}^t, \boldsymbol{x} - \boldsymbol{x}^t \rangle.$$

That is, the regret is the difference between the utility that the decision maker has actually achieved, and the utility that the decision maker *could have* achieved in hindsight by playing a fixed action $\boldsymbol{x}$ in all timesteps.

2.3.1 Relation to Equilibrium Finding

There is a well-known, tight connection between no-regret learning and equilibria in games. In particular, we have the following folk result whose proof follows almost directly from the definitions of NFCCE and regret:

Proposition 2.1 (No-regret learning converges to CCE). *In any game, if all players run no-regret learning algorithms over their strategy sets $\mathcal{X}_i$ with utilities $u_i^t(\boldsymbol{x}_i) := u_i(\boldsymbol{x}_i, \boldsymbol{x}_{-i}^t)$, then after T rounds, the* correlated *average strategy profile $\mu := \mathrm{unif}(\{\boldsymbol{x}^1, \dots, \boldsymbol{x}^T\})$ is an ϵ-NFCCE, where $\epsilon = \max_{i \in [n]} \textsc{Reg}_i(T)$ and $\textsc{Reg}_i(T)$ is the external regret of player i.*

Algorithm 2.1 (MWU): Multiplicative weight update on $\Delta(n)$.

```
initialize $\boldsymbol{z}^1 \leftarrow \mathbf{1}, t \leftarrow 0$
procedure NextStrategy(): return $\boldsymbol{x}^t := \boldsymbol{z}^t / \|\boldsymbol{z}^t\|_1$
procedure ObserveUtility($\boldsymbol{u}^t$): $\boldsymbol{z}^{t+1} \leftarrow \boldsymbol{z}^t \circ \exp(\eta \boldsymbol{u}^t)$
```

Algorithm 2.2 (RM+): Regret matching plus on $\Delta(n)$.

```
initialize $\boldsymbol{z}^1 \leftarrow \mathbf{0},\ t \leftarrow 0$
procedure NextStrategy():
    if $\boldsymbol{z}^t = \mathbf{0}$ then return $\boldsymbol{x}^t := \boldsymbol{z}^t / \|\boldsymbol{z}^t\|_1$
    else return $\boldsymbol{x}^t :=$ any strategy
procedure ObserveUtility($\boldsymbol{u}^t$): $\boldsymbol{z}^{t+1} \leftarrow [\boldsymbol{z}^t + \boldsymbol{u}^t - \langle \boldsymbol{u}^t, \boldsymbol{x}^t \rangle]^+$
```

In zero-sum games, using the fact that NFCCEs collapse to Nash, we have the following analogous result.

Proposition 2.2 (No-regret learning in zero-sum games converges to Nash equilibrium)**.** *In any zero-sum game, if both players run no-regret learning algorithms, then after T rounds, the* uncorrelated *average strategy profile $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$, where $\bar{\boldsymbol{x}} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{x}^t$ (and analogous for $\bar{\boldsymbol{y}}$) is an ϵ-equilibrium, where $\epsilon = \textsc{Reg}_{\blacktriangle}(T) + \textsc{Reg}_{\blacktriangledown}(T)$.*

Any no-regret learning algorithm for zero-sum games can be run with either *simultaneous* or *alternating* updates. While the above theoretical results apply only to the simultaneous versions, certain algorithms are also known to converge with alternating updates[2.2].

We will now introduce many fundamental algorithms for no-regret learning in normal- and extensive-form games, which will be referenced repeatedly throughout the remainder of the thesis.

2.3.2 Regret Minimization on Simplices

The most basic setting for no-regret learning is the setting in which $\mathcal{X}$ is the simplex $\Delta(n)$. Here, we introduce two simple no-regret learning algorithms on the simplex. Here, we review two common regret minimization algorithms which we will refer to repeatedly throughout this thesis, and some important variants of them.

Multiplicative Weight Update. The *multiplicative weights* (MWU) algorithm is parameterized by a single hyperparameter $\eta > 0$, called the *step size*. Multiplicative weights satisfies the following regret bound.

[2.2] For example, this is known to be true for CFR+ [44], but is nontrivial to show: the original proof attempt by Tammelin et al. [283] was flawed.

Algorithm 2.3 (GD): (Projected) gradient descent on a general convex compact set $\mathcal{X}$

```
1: initialize $\boldsymbol{x}^1 \in \mathcal{X}$ arbitrarily
2: procedure NextStrategy(): return $\boldsymbol{x}^t$
3: procedure ObserveUtility($\boldsymbol{u}^t$): $\boldsymbol{x}^{t+1} \leftarrow \Pi_{\mathcal{X}}(\boldsymbol{x}^t + \eta \boldsymbol{u}^t)$
```

Proposition 2.3. *The average external regret of MWU satisfies:*

$$\textsc{Reg}_{\mathsf{MWU}}(T) \lesssim \frac{\log n}{\eta T} + \eta \lesssim \sqrt{\frac{\log n}{T}}$$

where the second inequality follows by taking the step size $\eta = \sqrt{(\log n)/T}$.

Regret Matching Plus. The *regret matching* algorithm [143] is a simple, hyperparameter-free no-regret learning algorithm. Here, we will introduce a better, more recent variants of it, known as *regret matching plus* (RM+) [282].

Proposition 2.4 ([282])**.** *The average external regret of* RM+ *satisfies* $\textsc{Reg}_{\mathsf{RM+}}(T) \lesssim \sqrt{n/T}$.

As alluded to above, a major advantage of RM+ is that (unlike MWU) it is *hyperparameter-free*: there are no step sizes or other hyperparameters to tune. Similarly, RM+ is also *scale-invariant*: if given utility sequence $\boldsymbol{u}^1, \dots, \boldsymbol{u}^T$, it would produce the same iterates as if it had been given $C\boldsymbol{u}^1, \dots, C\boldsymbol{u}^T$ for any constant $C > 0$. These properties make RM+ extremely powerful in practice. In particular, despite a worse theoretical dependence on n, RM+ is almost always practically superior to MWU. Therefore, we will use it in almost all our experiments.

2.3.3 Regret Minimization in General Convex Domains

We now discuss regret minimization in arbitrary convex sets $\mathcal{X}$. The most standard regret minimization algorithm for such a setting is *(projected) gradient descent* (GD). Like multiplicative weights, it takes a single parameter, the step size η, and has the following regret guarantee.

Proposition 2.5. *The average external regret of* GD *satisfies*

$$\textsc{Reg}_{\mathsf{GD}}(T) \lesssim \frac{B^2}{\eta T} + \eta G^2 \lesssim \frac{BG}{\sqrt{T}}$$

where $B = \max_{\boldsymbol{x} \in \mathcal{X}} \|\boldsymbol{x}\|$ *bounds the diameter of* $\mathcal{X}$, $G = \max_{t \in [T]} \|\boldsymbol{u}^t\|$ *bounds the norm of the utility vectors, and the last inequality comes from setting* $\eta = B/G\sqrt{T}$.

Algorithm 2.4 (OMWU): Predictive (optimistic) multiplicative weight update on $\Delta(n)$.

```
initialize $\boldsymbol{z}^1 \leftarrow \mathbf{1}$, $t \leftarrow 0$
procedure NextStrategy($\tilde{\boldsymbol{u}}^t$)
    $\tilde{\boldsymbol{z}}^t \leftarrow \boldsymbol{z}^t \circ \exp(\eta \hat{\boldsymbol{u}}^t)$
    return $\boldsymbol{x}^t := \tilde{\boldsymbol{z}}^t / \|\tilde{\boldsymbol{z}}^t\|_1$
procedure ObserveUtility($\boldsymbol{u}^t$): $\boldsymbol{z}^{t+1} \leftarrow \boldsymbol{z}^t \circ \exp(\eta \boldsymbol{u}^t)$
```

Algorithm 2.5 (OGD): Predictive (optimistic) gradient descent.

```
initialize $\boldsymbol{z}^1 \in \mathcal{X}$ arbitrarily
procedure NextStrategy($\tilde{\boldsymbol{u}}^t$): return $\boldsymbol{x}^t := \Pi_{\mathcal{X}}(\boldsymbol{z}^t + \eta \tilde{\boldsymbol{u}}^t)$
procedure ObserveUtility($\boldsymbol{u}^t$): $\boldsymbol{z}^{t+1} \leftarrow \Pi_{\mathcal{X}}(\boldsymbol{z}^t + \eta \boldsymbol{u}^t)$
```

2.3.4 Predictive (Optimistic) Algorithms

Predicitions can be used to speed up regret minimization algorithms even further. In essence, predictive algorithms take an additional input on every timestep t, which is a *prediction* $\tilde{\boldsymbol{u}}^t$ of the utility vector that it will observe. The algorithm then uses the predicted utility vector to perform a temporary update before returning its strategy. The predictive variants of MWU and RM+ are known respectively as *optimistic multiplicative weights* (OMWU, [61, 250, 252, 281]), *optimistic gradient descent* (OGD, [246]), and *predictive regret matching plus* (PRM+, [104]).[2.3] Note that by setting $\tilde{\boldsymbol{u}}^t = \mathbf{0}$, the predictive variants collapse to the non-predictive variants. Conventionally (*i.e.*, unless otherwise stated), the predicition is set to the previous observed utility, that is, $\tilde{\boldsymbol{u}}^t = \boldsymbol{u}^{t-1}$.

Predictive regret matching has the same worst-case guarantee as non-predictive regret matching, but can be significantly faster, both in theory and in practice, if the predictions are accuarate.

2.3.5 Regret Minimization in Extensive-Form Games: Counterfactual Regret Minimization

In this section, we will introduce *counterfactual regret minimization* (CFR) [323], following the more recent exposition of Farina et al. [100]. Intuitively, CFR allows one to *build* a regret minimizer on a tree-form strategy set $\mathcal{X}$ by running *local* regret minimizers at each decision point, and combining them in a clever way. The guarantee given by CFR can be expressed as follows. Call a subset $P \subseteq \mathcal{J}$ *playable* if there is a pure strategy that reaches every decision point in P, that is, there is a pure strategy $\boldsymbol{x} \in \mathcal{X}$ such that $\boldsymbol{x}(p_j) = 1$ for every $j \in P$. Then:

[2.3] We use different wording (optimistic vs predictive) to be consistent with usage of past authors.

Algorithm 2.6 (PRM+): Predictive (optimistic) regret matching plus on $\Delta(n)$.

```
initialize $\boldsymbol{z}^1 \leftarrow \mathbf{0}$, $t \leftarrow 0$
procedure NextStrategy($\tilde{\boldsymbol{u}}^t$):
    $\tilde{\boldsymbol{z}}^t \leftarrow [\boldsymbol{z}^t + \tilde{\boldsymbol{u}}^t - \langle \tilde{\boldsymbol{u}}^t, \boldsymbol{x}^{t-1} \rangle]^+$
    if $\tilde{\boldsymbol{z}}^t = \mathbf{0}$ then return $\boldsymbol{x}^t := \tilde{\boldsymbol{z}}^t / \|\tilde{\boldsymbol{z}}^t\|_1$
    else return $\boldsymbol{x}^t :=$ any strategy
procedure ObserveUtility($\boldsymbol{u}^t$): $\boldsymbol{z}^{t+1} \leftarrow [\boldsymbol{z}^t + \boldsymbol{u}^t - \langle \boldsymbol{u}^t, \boldsymbol{x}^t \rangle]^+$
```

Algorithm 2.7 (CFR): Counterfactual regret minimization on tree-form decision problems $\mathcal{T}$. For each decision point j, $\mathcal{R}_j$ is a regret minimizer on $\Delta(\mathcal{A}(j))$.

```
initialize $t \leftarrow 0$
procedure NextStrategy()
    $\boldsymbol{x}^t(\varnothing) \leftarrow 1$
    for each decision point $j$, in top-down order do
        $\boldsymbol{r}_j^t \leftarrow \mathcal{R}_j$.NextStrategy()
        $\boldsymbol{x}^t(j*) \leftarrow \boldsymbol{x}^t(p_j)\boldsymbol{r}_j^t$
    return $\boldsymbol{x}^t$
procedure ObserveUtility($\boldsymbol{u}^t$)
    $\boldsymbol{v}^t \leftarrow \boldsymbol{u}^t$
    for each decision point $j$, in bottom-up order do
        $\mathcal{R}_j$.ObserveUtility($\boldsymbol{v}^t(j*)$)
        $\boldsymbol{v}^t(p_j) \leftarrow \boldsymbol{v}^t(p_j) + \langle \boldsymbol{r}_j^t, \boldsymbol{v}^t(j*) \rangle$
```

Proposition 2.6 ([100, 323]). *The average external regret of* CFR *satisfies*

$$\textsc{Reg}_{\mathsf{CFR}}(T) \le \max_P \sum_{j \in P} \textsc{Reg}_j(T) \le \sum_{j \in \mathcal{J}} \textsc{Reg}_j(T)$$

where the max is taken over all playable sets P, and $\textsc{Reg}_j(T)$ is the regret of the local regret minimizer at decision point j.

In particular, with (O)MWU and (P)RM+ as the regret minimizers, we get the regret bounds

$$\textsc{Reg}_{\mathsf{CFR\text{-}(O)MWU}}(T) \lesssim |\mathcal{J}|\sqrt{\frac{\log b}{T}} \le \frac{|\Sigma|}{\sqrt{T}} \quad \text{and} \quad \textsc{Reg}_{\mathsf{CFR\text{-}(P)RM+}}(T) \lesssim |\mathcal{J}|\sqrt{\frac{b}{T}} \le \frac{|\Sigma|}{\sqrt{T}}$$

where b is the branching factor. Several variants of CFR with specific choices of local regret minimizer have special common names. In particular, CFR with RM+ or PRM+ is known as CFR+ or PCFR+ respectively. The latter is currently the fastest regret minimizer in practice in most settings, including game solving [103][2.4].

[2.4] A notable exception is poker and variants thereof, where *discounted CFR* [38], which we will not need for this thesis, is sometimes faster.

Part I

Computing Optimal Solutions to Extensive-Form Games

Chapter 3

New Algorithms for Subgame Solving in Two-Player Zero-Sum Games, and Application to Superhuman Fog of War Chess

3.1 Introduction

Subgame solving is the standard technique for playing perfect-information games that has been used by strong agents in a wide variety of games, including chess [51, 276] and go [271]. Methods for subgame solving in perfect-information games exploit the fact that a solution to a subgame can be computed independently of the rest of the game. However, this condition fails in the imperfect-information setting, where the optimal strategy in a subgame can depend on strategies outside that subgame.

Recently, subgame solving techniques have been extended to imperfect-information games [121, 157]. Some of those techniques are provably *safe* in the sense that, under reasonable conditions, incorporating them into an agent cannot make the agent more exploitable [36, 40, 41, 43, 181, 221, 222, 279]. These techniques formed the core ingredient toward recent superhuman breakthroughs in AIs for no-limit Texas hold'em poker [37, 39]. However, all of the prior techniques have a shared weakness that limits their applicability: as a first step, they enumerate the entire *common-knowledge closure* of the player's current infoset, which is the smallest set of states within which it is common knowledge that the current node lies. In two-player community-card poker (in which each player is dealt private hole cards, and all actions are public, e.g., Texas hold'em), for example, the common-knowledge closure contains one node for each assignment of hole cards to both players. This set has a manageable size in such poker games, but in other games, it is unmanageably large.

We introduce a different technique to avoid having to enumerate the entire common-knowledge closure. We enumerate only the set of nodes corresponding to kth-order knowledge for finite

k—in the present work, we focus mostly on the case $k = 1$ and $k = 2$, for it already gives us interesting results. This allows an agent to only conduct subgame solving on still-reachable states, which in general is a much smaller set than the whole common-knowledge subgame.

We prove that, as is, the resulting algorithm, 1-KLSS, does not guarantee safety, but we develop three avenues by which safety can be guaranteed. First, safety is guaranteed if the results of subgame solves are incorporated back into the blueprint strategy. Second, we provide a method by which safety is achieved by limiting the infosets at which subgame solving is performed. Third, we prove that our approach, when applied at every infoset reached during play, achieves a weaker notion of equilibrium, which we coin *affine equilibrium* and which may be of independent interest. We show that affine equilibria cannot be exploited by any Nash strategy of the opponent: an opponent who wishes to exploit an affine equilibrium must open herself to counter-exploitation. Even without these three safety-guaranteeing additions, experiments on medium-sized games show that 1-KLSS always reduced exploitability in practical games even when applied at every infoset.

We used these techniques to create *Obscuro*, an AI that achieved superhuman performance in *Fog of War (FoW) chess* (aka. *dark chess*), the most popular variant of imperfect-information chess. Over 120 games against humans of varying skill levels—including the #1-ranked human—and 1,000 games against the previous state-of-the-art FoW chess AI from the previous chapter, we conclusively demonstrate that *Obscuro* is stronger than any other current agent—human or artificial—for FoW chess. FoW chess is now the largest (measured by amount of imperfect information, that is, the typical size of an information set) turn-based game in which superhuman performance has been achieved and the largest game in which imperfect-information search techniques have been successfully applied.

3.1.1 Challenges in Imperfect-Information Games such as Fog of War Chess

Imperfect-information versions of chess have captured the imagination of chess players and scientists alike for over a century. To our knowledge, the first imperfect-information version of chess was *Kriegspiel*, invented in 1899 and based on the earlier game *Kriegsspiel*, a war game used by the Prussian army in the early 19th century for training [247]. In the modern day, there are multiple imperfect-information variants of chess, including *Kriegspiel, reconnaissance blind chess* (RBC), and *Fog of War (FoW) chess*.[3.1] Imperfect-information chess is a recognized challenge problem in AI. Although there has been AI research in Kriegspiel [64, 240, 260] and RBC [126], strong performance has not been achieved in Kriegspiel, and RBC is not played competitively by humans. By comparison, FoW chess has surged in popularity due to its implementation on the major chess website chess.com, and strong human experts have emerged among thousands of active players.[3.2] It is the most popular variant of imperfect-information chess by far, and strong human experts exist who can serve as challenging benchmarks of progress.

FoW chess presents a unique combination of challenges that did not exist in prior superhuman AI

[3.1] Despite its similar name, *Chinese dark chess* has no private information, and thus does not require the types of reasoning that are required in FoW chess.

[3.2] As of April 2025, the Fog of War chess leaderboard on chess.com [3] listed 19,150 active players.

milestones.[3.3] First, chess itself is a highly tactical game often requiring careful lookahead, and FoW chess is no different: there are often positions where one player has perfect or near-perfect information and can execute a sequence of moves that results in an advantage. Thus, a strong agent must have solid lookahead capability. Lookahead in other games is usually accomplished by subgame solving. Thus it would be desirable to be able to conduct subgame solving in FoW chess too.

Second, private information is rapidly gained and lost. It is possible for the size of a player's *information set* (infoset)—*i.e.*, set of indistinguishable positions given a player's observations—to rapidly increase and then decrease again, for example, from hundreds up to millions and then back down to hundreds, in a matter of a few moves. Thus, a strong agent must have the ability to reason about this rapidly-changing information.

Third, a strong agent must at least somewhat play a *mixed strategy*—that is, it must randomize its actions. Otherwise, an adversary who knows the strategy, or has learned the strategy from past observation, can easily exploit that knowledge.

Finally, in games like FoW chess, reasoning about *common knowledge* is difficult. This is a key challenge because most algorithms for subgame solving—including those that led to breakthroughs in no-limit Texas hold'em poker—rely on the ability to reason about common knowledge, or often even the ability to *enumerate* the entire common-knowledge set—that is, the smallest set of histories C with the property that it is common knowledge that the true history lies in C [37, 39]. So, to prepare for solving a subgame, prior algorithms need to reason about what the agent knows about what the opponent knows about what the agent knows, and so on. This need can dramatically expand the set of states that need to be incorporated into the subgame solving algorithm, making such methods impractical for games much larger than no-limit Texas hold'em.

For example, consider the two FoW chess positions in Figure 3.1.[3.4] Although seemingly completely distinct, it is possible to show (see Section 3.9) that these two positions are connected by no fewer than nine levels of "I think that you think that..." reasoning. Prior techniques would require the ability to generate this complex connection before starting subgame solving from either of the two positions.

Such intricacies make it difficult to reason about common knowledge efficiently. For example, common-knowledge sets in FoW chess can quickly grow prohibitively large, so they cannot be held directly in memory. In FoW chess, individual *infosets* often have size as large as 10^6 and can have size 10^9. Common-knowledge sets can have size 10^{18}—far too large to be enumerated in reasonable time or space during search.[3.5] Perhaps even more troubling is the fact that it is not even clear that it is possible to efficiently decide whether two histories can be distinguished by common knowledge, so in some sense reasoning about common knowledge may *require* enumerating the common-knowledge set in the worst case.

This is in sharp contrast to poker, which has special structure that has driven the success of past

[3.3] The complete rules of FoW chess can be found in Section 3.9

[3.4] The sequence of moves in the figure is purely for the purpose of illustrating common knowledge, and does not represent strong play. For example, *Obscuro* never plays **1... g5** or **2... Qh4**.

[3.5] Detailed calculations for these lower bounds can be found in Section 3.9.

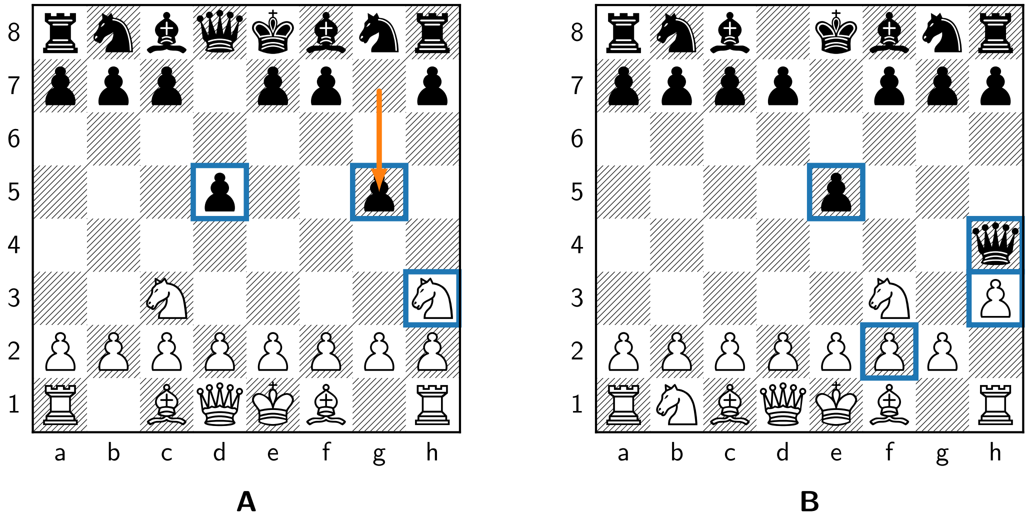


Figure 3.1: *Two FoW chess positions in the same common-knowledge set.* **(A)** *position after moves* **1. Nc3 g5 2. Nh3 d5***;* **(B)** *position after moves* **1. Nf3 e5 2. h3 Qh4***. The boxed squares mark pieces visible to the opponent.*

efforts in that game. First, at least in two-player (heads-up) Texas hold'em poker, common-knowledge sets are not very large. They have size at most $\binom{52}{2}\binom{50}{2} \approx 1.6 \times 10^6$, and can thus easily be held in memory. Moreover, thanks to poker-specific optimizations [163], subgame solving in poker can be implemented in such a way that its complexity depends not on the size of the common-knowledge set but merely on the size of the infoset, enabling feasible subgame solving even when the common-knowledge sets are large, as is the case in multi-player poker.[3.6] In more general games where these domain-specific techniques do not apply—such as FoW chess—the complexity of traditional subgame-solving techniques for imperfect-information games would scale with the size of the common-knowledge set, which in our case renders such techniques totally infeasible.

3.2 Preliminaries

In this section, we consider *timeable* two-player zero-sum games of imperfect recall with explicitly-defined observations. That is, each player has a function $o_i : \mathcal{H} \to \mathbb{R}$ defining the observation that player i makes at history h. The sequence $s_i(h)$ consists of all observations made by the player at nodes up to *and including* h. The set of sequences of player i is Σ_i.

We say that two states $h = \varnothing a_1 \dots a_t$ and $h' = \varnothing b_1 \dots b_t$ are *indistinguishable to player i*, denoted $h \sim_i h'$, if $s_i(h) = s_i(h')$. An equivalence class of nodes $h \in \mathcal{H}$ under $\sim_i$ is an infoset. Notice that infosets are well-defined here even for the player not moving—this will be critical later on.

[3.6] Specifically, *Pluribus* [39] would not have been feasible without these poker-specific optimizations.

If u, u' are nodes or sequences, $u \preceq u'$ means u is an ancestor of u' (or $u' = u$). If S is a set of nodes, $h \succeq S$ means $h \succeq h'$ for some $h' \in S$, and $\overline{S} = \{z : z \succeq S\}$.

The *conditional value* $u(\boldsymbol{x}, \boldsymbol{y}|S)$ is the conditional expectation given that some node in the set S is hit. The *(conditional) best-response value* $u^*(\boldsymbol{x}|Ja)$ to a ▲-strategy $\boldsymbol{x} \in \mathcal{X}$ upon playing action a at ▼-infoset J is the best possible conditional value that ▼ against x after playing a at J:

$$u^*(\boldsymbol{x}|Ja) = \min_{\boldsymbol{y} \in \mathcal{Y} : \boldsymbol{y}(Ja)=1} u(\boldsymbol{x}, \boldsymbol{y}|J).$$

The *best-response value* $u^*(\boldsymbol{x})$ (without specifying an infoset) is the best-response value at the root, i.e., $\min_{\boldsymbol{y} \in \mathcal{Y}} u(\boldsymbol{x}, \boldsymbol{y})$. Analogous definitions hold for ▼-strategy $\boldsymbol{y}$ and ▲-infoset I.

The *counterfactual value* $u_{\text{cf}}(\boldsymbol{x}, \boldsymbol{y}; Ja)$ for ▼ is the conditional value, scaled by the probability that ▲ (and nature) plays to reach J:

$$u_{\text{cf}}(\boldsymbol{x}, \boldsymbol{y}; Ja) = u(\boldsymbol{x}, \boldsymbol{y}|Ja) \sum_{h \in J} p(h)\boldsymbol{x}(h).$$

The counterfactual best-response values are defined as the analogously scaled versions of the conditional best response values.

We will distinguish between *states* and *histories*. A *state* is a sufficient statistic for future play of the game. That is, all data about the subtree rooted at a history h is uniquely determined by the state at h. Multiple histories can have the same state; such histories are called *transpositions*. For example, ignoring draw rules, two chess positions are transpositions if they have equal piece locations, castling rights, and *en passant* rights.

3.3 Common-Knowledge Subgame Solving

In this section we discuss prior work on subgame solving. First, ▲ computes a blueprint strategy $\boldsymbol{x}$ for the full game. During a playthrough, ▲ reaches an infoset I, and would like to perform subgame solving to refine her strategy for the remainder of the game. All prior subgame solving methods that we are aware of require, as a first step, constructing [36, 40, 41, 43, 181, 221, 222, 279], or at least approximating via samples [280], the *common-knowledge closure* of I.

Definition 3.1. The *infoset hypergraph* $\mathcal{G}$ of a game Γ is the hypergraph whose vertices are the nodes of Γ, and whose hyperedges are information sets.

Definition 3.2. Let S be a set of nodes in Γ. The *order-k knowledge set* S^k is the set of nodes that are at most distance $k-1$ away from S in $\mathcal{G}$. The *common-knowledge closure* S^∞ is the connected component of $\mathcal{G}$ containing S.

Intuitively, if we know that the true node is in S, then we know that the opponent knows that the true node is in S^2, we know that the opponent knows that we know that the true node is in S^3, etc., and it is common knowledge that the true node is in S^∞. After constructing I^∞ (where I, as above, is the infoset ▲ has reached), standard techniques then construct the subgame $\overline{I^\infty}$ (or an abstraction of it), and solve it to obtain the refined strategy. In this section we describe three variants: *resolving* [43], *maxmargin* [221], and *reach subgame solving* [36].

Let $\mathcal{H}_{\text{top}}$ be the set of root nodes of I^∞, that is, the set of nodes $h \in I^\infty$ for which the parent of h is not in I^∞. In *subgame resolving*, the following gadget game is constructed. First, nature chooses a node $h \in \mathcal{H}_{\text{top}}$ with probability proportional to $p(h)\boldsymbol{x}(h)$. Then, ▼ observes her infoset $I_{\blacktriangledown}(h)$, and is given the choice to either *exit* or *play*. If she exits, the game ends at a terminal node z with $u(z) = u^*_{\text{cf}}(\boldsymbol{x}; I_{\blacktriangledown}(h))$. This payoff is called the *alternate payoff* at $I_{\blacktriangledown}(h)$. Otherwise, the game continues from node h. In *maxmargin* solving, the objective is changed to instead find a strategy $\boldsymbol{x}'$ that maximizes the minimum *margin* $M(I) := u^*_{\text{cf}}(\boldsymbol{x}'; I) - u^*_{\text{cf}}(\boldsymbol{x}; I)$ associated with any ▼-infoset I intersecting $\mathcal{H}_{\text{top}}$. (Resolving only ensures that all margins are positive). This can be accomplished by modifying the gadget game. In *reach subgame solving*, the alternative payoffs $u^*_{\text{cf}}(\boldsymbol{x}; I)$ are decreased by the *gift* at I, which is a lower bound on the magnitude of error that ▼ has made by playing to reach I in the first place. Reach subgame solving can be applied on top of either resolving or maxmargin.

The full game Γ is then replaced by the gadget game, and the gadget game is resolved to produce a strategy $\boldsymbol{x}'$ that ▲ will use to play to play after I. To use nested subgame solving, the process repeats when another new infoset is reached.

3.4 Knowledge-Limited Subgame Solving

In this section we introduce the main contribution of this chapter, *knowledge-limited subgame solving*. The core idea is to reduce the computational requirements of safe subgame solving methods by discarding nodes that are "far away" (in the infoset hypergraph $\mathcal{G}$) from the current infoset.

Fix an odd positive integer k. In *order-k knowledge-limited subgame solving* (k-*KLSS*), we *fix* ▲'s strategy outside $\overline{I^k}$, and then perform subgame solving as usual. This carries many advantages:

1. Since ▲'s strategy is fixed outside $\overline{I^k}$, ▼'s best response outside $\overline{I^{k+1}}$ is also fixed. Thus, all nodes outside $\overline{I^{k+1}}$ can be pruned and discarded.

2. At nodes $h \in \overline{I^{k+1}} \setminus \overline{I^k}$, ▲'s strategy is again fixed. Thus, the payoff at these nodes is only a function of ▼'s strategy in the subgame and the blueprint strategy. These payoffs can be computed from the blueprint and added to the row of the payoff matrix corresponding to ▲'s empty sequence. These nodes can then also be discarded, leaving only $\overline{I^k}$.

3. Transpositions can be accounted for if $k = 1$ and we allow a slight amount of incorrectness. Suppose that $h, h' \in I$ are transpositions. Then ▲ cannot distinguish h from h' ever again. Further, ▼'s information structure after h in $\overline{I^k}$ is identical to her information structure in h' in $\overline{I^k}$. Thus, in the payoff matrix of the subgame, h and h' induce two disjoint sections of the payoff matrix A_h and $A_{h'}$ that are identical except for the top row (thanks to Item 2 above). We can thus remove one (say, at random) without losing too much. If one section of the matrix contains entries that are all not larger than the corresponding entries of the

other part, then we can remove the latter part without any loss since it is weakly dominated.

The transposition merging may cause incorrect behavior (over-optimism) in games such as poker, but we believe that its effect in a game like FoW chess, where information is transient at best and the evaluation of a position depends more on the actual position than on the players' information, is minor. Other abstraction techniques can also be used to reduce the size of the subgame, if necessary. We will denote the resulting gadget game $\Gamma[I^k]$.

In games like FoW chess, even individual infosets can have size 10^7, which means even I^2 can have size 10^{14} or larger. This is wholly unmanageable in real time. Further, very long shortest paths can exist in the infoset hypergraph $\mathcal{G}$. As such, it may be difficult to even determine whether a given node is in I^∞, much less expand all its nodes, even approximately. Thus, being able to reduce to I^k for finite k is a large step in making subgame solving techniques practical.

The benefit of KLSS can be seen concretely in the following parameterized family of games which we coin *N-matching pennies*. We will use it as a running example in the rest of this chapter. Nature first chooses an integer $n \in \{1, \dots, N\}$ uniformly at random. ▲ observes $\lfloor n/2 \rfloor$ and ▼ observes $\lfloor (n+1)/2 \rfloor$. Then, ▲ and ▼ simultaneously choose heads or tails. If they both choose heads, ▲ scores n. If they both choose tails, ▲ scores $N - n$. If they choose opposite sides, ▲ scores 0. For any infoset I just after nature makes her move, there is no common knowledge whatsoever, so $\overline{I^\infty}$ is the whole game except for the root nature node. However, I^k consists of only $\Theta(k)$ nodes.

On the other hand, in community-card poker, I^∞ itself is quite small: indeed, in heads-up Texas Hold'Em, I^∞ always has size at most $\binom{52}{2} \cdot \binom{50}{2} \approx 1.6 \times 10^6$ and even fewer after public cards have been dealt. Furthermore, game-specific tricks or matrix sparsification [163, 300] can make game solvers behave as if $I^\infty \approx 10^3$. This is manageable in real time, and is the key that has enabled recent breakthroughs in AIs for no-limit Texas hold'em [37, 39, 222]. In such settings, we do not expect our techniques to give improvement over the current state of the art.

The rest of this section addresses the *safety* of KLSS. The techniques in Section 3.3 are *safe* in the sense that applying them at every infoset reached during play in a nested fashion cannot increase exploitability compared to the blueprint strategy [36, 43, 221]. KLSS is not safe in that sense:

Proposition 3.3. *There exists a game and blueprint for which applying* 1*-KLSS at every infoset reached during play increases exploitability by a factor linear in the size of the game.*

Proof. Consider the following game. Nature chooses an integer $n \in \{1, \dots, N\}$, and tells ▲ but not ▼. Then the two players play matching pennies, with ▼ winning if the pennies match. Consider the blueprint strategy for ▲ that plays heads with probability exactly $1/2 + 2/N$, regardless of n. This strategy is a $\Theta(1/N)$-equilibrium strategy for ▲. However, if maxmargin 1-KLSS is applied independently at every infoset reached, ▲ will deviate to playing tails for all n, because she is treating her strategy at all $m \neq n$ as fixed, and the resultant strategy is more balanced. This strategy is exploitable by ▼ always playing tails. □

Despite the above negative example, we now give multiple methods by which we can obtain safety guarantees when using KLSS.

3.4.1 Safety by Updating the Blueprint

Our first method of obtaining safety is to immediately and permanently update the blueprint strategy after every subgame solution is computed. Proofs of the results in this section can be found in the appendix.

Theorem 3.4. *Suppose that whenever k-KLSS is performed at infoset I (e.g., it can be performed at every infoset reached during play in a nested manner), and that subgame strategy is immediately and permanently incorporated into the blueprint, thereby overriding the blueprint strategy in $\overline{I^k}$. Then the resulting sequence of blueprints has non-increasing exploitability.*

> *Proof.* We begin with a lemma that will be useful for many of the remaining theoretical results.
>
> **Lemma 3.5.** *Let (x, y) be a blueprint strategy, and I be an infoset for player 1 with $\boldsymbol{x}(I) > 0$. Then fixing strategies for both players at all nodes $h \not\succeq I$; performing resolving, maxmargin, or reach subgame solving at only $\overline{I^k}$; and then playing according to that strategy in $\overline{I^k}$ and $\boldsymbol{x}$ elsewhere, results in a strategy $\boldsymbol{x}'$ that is not more exploitable than x.*
>
> > *Proof.* Identical to the proof of safety of subgame resolving [43]: we always have access to our blueprint strategy, which by design makes all margins nonnegative. □
>
> The theorem now follows by applying the lemma repeatedly. □

To recover a full safety guarantee from Theorem 3.4, the blueprint—not the subgame solution—should be used during play, and the only function of the subgame solve is to update the blueprint for later use. One way to track the blueprint updates is to store the computed solutions to all subgames that the agent has ever solved. In games where only a reasonably small number of paths get played in practice (this can depend on the strength and style of the players), this is feasible. In other games this might be prohibitively storage intensive.

It may seem unintuitive that we cannot use the subgame solution on the playthrough on which it is computed, but we can use it forever after that (by incorporating it into the blueprint), while maintaining safety. This is because, if we allow the *choice of information set I* in Theorem 3.4 to depend on the opponent's strategy, the resulting strategy is exploitable due to Proposition 3.3. By only using the subgame solve result at later playthroughs, the choice of I no longer depends on the opponent strategy at the later playthrough, so we recover a safety guarantee.

One might further be concerned that what the opponent or nature does in some playthrough of the game affects our strategy in later playthroughs and thus the opponent can learn more about, or affect, the strategy she will face in later playthroughs. However, this is not a problem. If the blueprint is an ϵ-NE, the opponent (or nature) can affect *which* ϵ-NE we will play at later playthroughs, but because we will always play from *some* ϵ-NE, we remain unexploitable.

In the rest of this section we prove forms of safety guarantees for 1-KLSS that do not require the blueprint to be updated at all.

3.4.2 Safety by Allocating Deviations from the Blueprint

We now show that another way to achieve safety of 1-KLSS is to carefully allocate how much it is allowed to deviate from the blueprint. Let $\mathcal{G}'$ be the graph whose nodes are infosets for ▲, and in which two infosets I and I' share an edge if they contain nodes that are in the same ▼-infoset. In other words, $\mathcal{G}'$ is the infoset hypergraph $\mathcal{G}$, but with every ▲-infoset collapsed into a single node.

Theorem 3.6. *Let $\boldsymbol{x}$ be an ϵ-NE blueprint strategy for ▲. Let $\mathcal{I}$ be an independent set in $\mathcal{G}'$ that is closed under ancestor (that is, if $I \succeq I'$ and $I \in \mathcal{I}$, then $I' \in \mathcal{I}$). Suppose that 1-KLSS is performed at every infoset in $\mathcal{I}$, to create a strategy $\boldsymbol{x}'$. Then $\boldsymbol{x}'$ is also an ϵ-NE strategy.*

Proof. By induction on the infoset structure. Assume WLOG that ▲ has a root infoset I_0.

Base case. If ▲ has only one infoset, then Lemma 3.5 applies.

Inductive case. Let $\mathcal{I}' \subset \mathcal{I}_1$ be the collection of infosets that could be the next infosets reached after I_0. Formally, $\mathcal{I}' = \{I \in \mathcal{I}_1 : I \succ I_0$ and there is no I' such that $I \succ I' \succ I_0\}$. Since $\mathcal{I}$ is closed under ancestors, for each infoset $I \in \mathcal{I}' \setminus \mathcal{I}$, the downward closure $\bar{I}$ does not intersect with $\mathcal{I}$. Thus, the strategy in $\bar{I}$ will be left untouched, and is treated as fixed by all subgame solves.

Subgame solving is then performed at every information set $I \in \mathcal{I} \cap \mathcal{I}'$. By inductive hypothesis, for each I, this gives a Nash equilibrium $\boldsymbol{x}_I$ of $\Gamma[I]$, which, by definition of $\Gamma[I]$, makes all margins in that subgame nonnegative. Since $\mathcal{I}$ is an independent set, the margin of each ▼-infoset is only dependent on at most one of the subgame solves. Thus, replacing the strategy in $\bar{I}$ with $\boldsymbol{x}_I$ for each $I \in \mathcal{I} \cap \mathcal{I}'$ still leaves all nonnegative margins in the original game, which completes the proof. □

To apply this method safely, we may select beforehand a distribution π over independent sets of $\mathcal{G}'$, which induces a map $p : V(\mathcal{G}') \to \mathbb{R}$ where $p(I) = \Pr_{\mathcal{I} \sim \pi}[I \in \mathcal{I}]$. Then, upon reaching infoset I, with probability $1 - p(I)$, play the blueprint until the end of the game; otherwise, run 1-KLSS at I (possibly resulting in more nested subgame solves) and play that strategy instead. It is always safe to set $p(I) \le 1/\chi(I^\infty)$ where $\chi(I^\infty)$ denotes the chromatic number of the subgraph of $\mathcal{G}'$ induced by the infosets in the common-knowledge closure I^∞. For example, if the game is perfect information, then $\mathcal{G}'[I^\infty]$ is the trivial graph with only one node I, so, as expected, it is safe to set $p(I) = 1$, that is, perform subgame solving everywhere.

3.4.3 Affine Equilibrium, which Guarantees Safety against All Equilibrium Strategies

We now introduce the notion of *affine equilibrium*. We will show that such equilibrium strategies are safe against all NE strategies, which implies that they are only exploitable by playing non-NE strategies, that is, by opening oneself up to counter-exploitation. We then show that 1-KLSS finds such equilibria.

Definition 3.7. A vector $\boldsymbol{x}$ is an *affine combination* of vectors $\boldsymbol{x}_1, \dots, \boldsymbol{x}_k$ if $\boldsymbol{x} = \sum_{i=1}^k \alpha_i \boldsymbol{x}_i$ with $\sum_i \alpha_i = 1$, where the coefficients α_i can have arbitrary magnitude and sign.

Definition 3.8. An *affine equilibrium strategy* is an affine combination of NE strategies.

In particular, if the NE is unique, then so is the affine equilibrium. Before stating our safety guarantees, we first state another fact about affine equilibria that illuminates their utility.

Proposition 3.9. *Every affine equilibrium is a best response to every NE strategy of the opponent.*

Proof. Let $\boldsymbol{y}^*$ be a ▼-NE strategy. Let $\boldsymbol{x}$ be an affine equilibrium for ▲, and write $\boldsymbol{x} = \sum_i \alpha_i \boldsymbol{x}_i^*$ where $\boldsymbol{x}_i^*$ are Nash equilibria, and $\sum_i \alpha_i = 1$ (but α_i are not necessarily positive). Then we have

$$u(\boldsymbol{x}, \boldsymbol{y}^*) = \sum_i \alpha_i u(\boldsymbol{x}_i^*, \boldsymbol{y}^*) = u^*. \qquad \square$$

In other words, every affine equilibrium is an NE of the restricted game Γ' in which ▼ can only play her NE strategies in Γ. That is, affine equilibria are not exploitable by NE strategies of the opponent, not even by safe exploitation techniques [122]. So, the only way for the opponent to exploit an affine equilibrium is to open herself up to counter-exploitation. Affine equilibria may be of independent interest as a reasonable relaxation of NE in settings where finding an exact or approximate NE strategy may be too much to ask for.

Theorem 3.10. *Let $\boldsymbol{x}$ be a blueprint strategy for ▲, and suppose that $\boldsymbol{x}$ happens to be an NE strategy. Suppose that we run 1-KLSS using the blueprint $\boldsymbol{x}$, at every infoset in the game, to create a strategy $\boldsymbol{x}'$. Then $\boldsymbol{x}'$ is an affine equilibrium strategy.*

Proof. By induction on the infoset structure. As above, assume WLOG that ▲ has a root infoset I_0.

Base case. If ▲ has only one infoset, then Lemma 3.5 applies.

Inductive case. Let $\mathcal{I}'$ be as in the previous proof. By inductive hypothesis, for each $I \in \mathcal{I}'$, running subgame solving on $\bar{I}$ yields a strategy $\boldsymbol{x}_I$ that is an affine equilibrium in $\Gamma[I]$. By definition of affine equilibrium, write $\boldsymbol{x}_I = \sum_j \alpha_{I,j} \boldsymbol{x}_{I,j}$ where $\boldsymbol{x}_{I,j}$ are Nash equilibria of $\Gamma[I]$. Let $\boldsymbol{x}'_I$ be the strategy in Γ defined by playing according to $\boldsymbol{x}_I$ in $\bar{I}$, and the blueprint everywhere else.

> Then each $\boldsymbol{x}'_I$ is an affine equilibrium, because it is an affine combination of the strategies $\boldsymbol{x}'_{I,j}$, which by Lemma 3.5 are Nash equilibria of Γ. But then the strategy created by running subgame solving at every $I \in \mathcal{I}'$, which is $\boldsymbol{x} + \sum_{I \in \mathcal{I}'}(\boldsymbol{x}'_I - \boldsymbol{x})$, is an affine combination of affine equilibria, and hence itself an affine equilibrium. □

The theorem could perhaps be generalized to approximate equilibria, but the loss of a large factor (linear in the size of the game, in the worst case) in the approximation would be unavoidable: the counterexample in the proof of Proposition 3.3 has a $\Theta(1/N)$-NE becoming a $\Theta(1)$-NE, in a game where the Nash equilibria are already affine-closed (that is, all affine combinations of Nash equilibria are Nash equilibria). Furthermore, it is nontrivial to even define ϵ-affine equilibrium.

Theorem 3.10 and Proposition 3.3 together suggest that 1-KLSS may make mistakes when $\boldsymbol{x}$ suffers from *systematic* errors (e.g., playing a certain action a too frequently *overall* rather than in a particular infoset). 1-KLSS may overcorrect for such errors, as the counterexample clearly shows. Intuitively, if the blueprint plays action a too often (e.g., folds in poker), 1-KLSS may try to correct for that game-wide error fully in each infoset, thereby causing the strategy to overall be very far from equilibrium (e.g., folding way too infrequently in poker). However, we will demonstrate that this overcorrection never happens in our experiments in practical games, even if the blueprint contains very systematic errors.

Strangely, the proofs of both Theorem 3.10 and Theorem 3.6 do not work for k-KLSS when $k > 1$, because it is no longer the case that the strategies computed by subgame solving are necessarily played—in particular, for $k > 1$, k-KLSS on an infoset I computes strategies for infosets I' that are no longer reachable, and such strategies may never be played. For $k = \infty$—that is, for the case of common knowledge—it is well known that the theorems hold via different proofs [36, 43, 221]. We leave the investigation of the case $1 < k < \infty$ for future research.

3.4.4 2-KLUSS: Unfreezing the Order-2 Subgame

We now introduce one additional change to KLSS: we allow ▲-nodes in $\overline{I^2} \setminus \overline{I}$ to be unfrozen and hence re-optimized in the subgame. We call this variant 2-knowledge-limited *unfrozen* subgame solving (KLUSS),[3.7] since its complexity depends on the order-2 subgame $\overline{I^2}$. 2-KLUSS essentially amounts to pretending that $\overline{I^2} = \overline{I^\infty}$.

We now make a few remarks about KLUSS, as it is applied to our FoW chess agent *Obscuro*:

1. Like 1-KLSS, 2-KLUSS lacks safety guarantees in the worst case. However, KLSS is often safe in practice, and KLUSS outperforms KLSS in FoW chess as we will show in the ablation experiments in Section 3.7. There are two further considerations:
   (a) *Obscuro* does not *have* a full-game blueprint: its blueprint is simply the strategy from the previous timestep, which is depth limited. Thus, we *must* use some form of subgame solving to play the game. KL(U)SS is currently the only variation of subgame

[3.7] This can be easily generalized to k-KLUSS for any k.

solving that is both somewhat game-theoretically motivated for imperfect-information games and computationally feasible in a game like FoW chess.

(b) Although both KLSS and KLUSS are unsafe in the worst case, it should be heuristically intuitive that they should improve performance *more* when the blueprint itself is of low quality. Indeed, we *expect* our "blueprints" (strategies carried over from the previous timestep) to have rather low quality, especially deep in the search tree where such strategies are based on very low-depth search! So, we believe heuristically that using KL(U)SS in this manner should usually be game-theoretically sound.

2. Since our equilibrium-finding module for *Obscuro* is based on CFR instead of linear programming—in particular, it uses the full game tree $\tilde{\Gamma}$ instead of a sequence-form representation—it does not benefit from freezing the ▲-nodes in $\overline{I^2} \setminus \overline{I^1}$, since those nodes would still need to be maintained. Thus, there is less reason for us to freeze those nodes. Further, with straightforward pruning techniques (namely, *partial pruning* [35]), CFR iterations usually take *sublinear* time in the size of the game tree (unlike linear programming, which takes at least linear time in the representation size), reducing the need to optimize the size of the game representation.

3. Again since we use CFR, the solutions that are computed by the equilibrium-finding module are inherently *approximate*, and especially at levels deep in the tree, their approximation can be relatively poor. As such, allowing these infosets to be unfrozen gives them the chance to learn better actions.

4. 1-KLSS removes the nodes in $\overline{I^2} \setminus \overline{I^1}$, folding them into the sequence-form representation for efficiency. In contrast, our approach of maintaining these nodes allows them to be *selected for expansion*. This fixes a weakness of KLSS when applied in the fashion that we apply it in *Obscuro*: if we removed these nodes, we would only capable of searching for bluff opportunities "locally", since any ▲-node in $\overline{I^2} \setminus \overline{I^1}$ would cease to be in the tree once the search horizon was passed. In contrast, *Obscuro* is capable of maintaining ▲-nodes in $\overline{I^2} \setminus \overline{I^1}$ for a long time, allowing deeper bluff opportunities.

5. Liu et al. [201] introduced a *safe* variant of KLSS, which they call *safe KLSS*, in which the subgame solver attempts to find a subgame strategy $\boldsymbol{x}'$ that maintains at least the same value for every *opponent strategy* y, instead of against every *infoset* J. This is a much stricter condition that is much more difficult to satisfy and thus substantially constrains the strategy to be close to the blueprint. Therefore, the safety requirement significantly decreases the power and value of subgame solving, especially when the blueprint is bad. Moreover, safe KLSS drops all nodes outside $\overline{I^1}$, which once again introduces the problem of the previously-listed item: if we were to use safe KLSS in our setting, our AI would not be capable of exploiting long bluff opportunities.

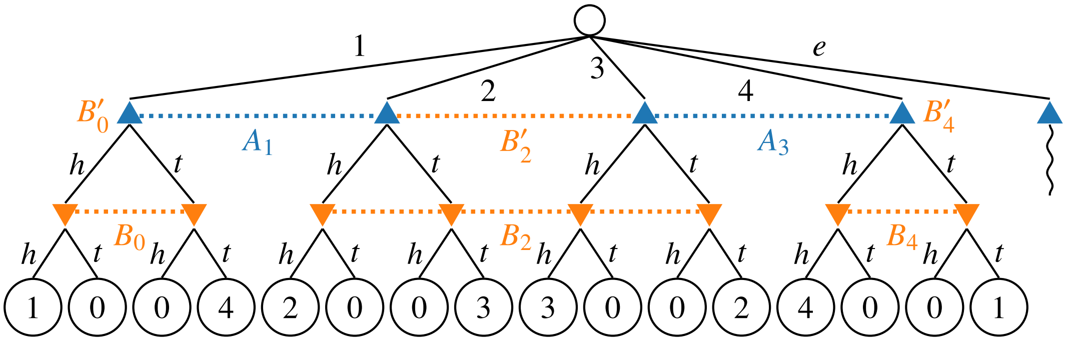


Figure 3.2: *A simple game that we use in our example. The game is a modified version of* 4*-matching pennies. Circles are nature or terminal; terminal nodes are labeled with their utilities. Nodes will be referred to by the sequence of edges leading to that node; for example, the rightmost terminal node is* $1tt$*. The details of the subgame at e are irrelevant. Nature's strategy at the root node is uniform random.*

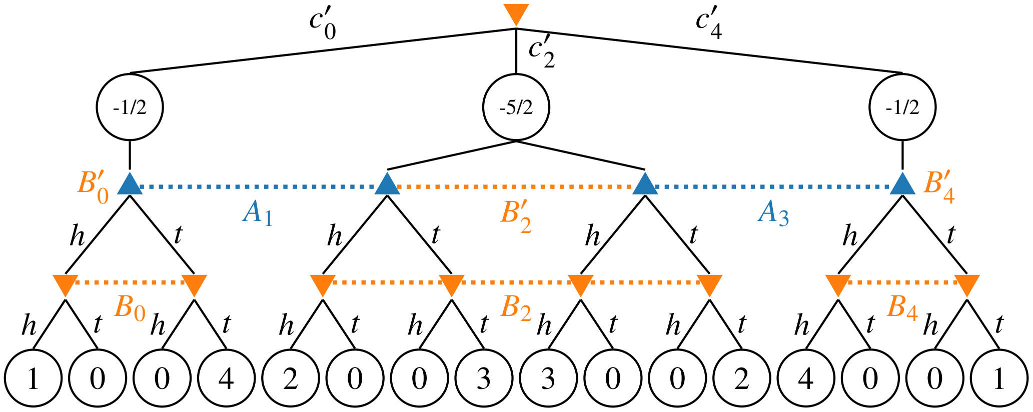


Figure 3.3: *The common-knowledge subgame at* A_1*,* $\Gamma[A_1^\infty]$*. Nature's strategy at all its nodes, once again, is uniform random. The nodes* c'_0 *and* c'_4 *are redundant because nature only has one action, but we include these for consistency with the exposition.*

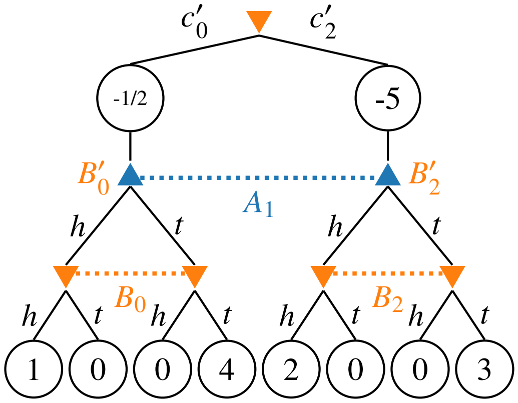


Figure 3.4: *The subgame for* 1*-KLSS at* A_1*. Once again, both nature nodes are redundant, but included for consistency with the exposition. The counterfactual value at* c'_2 *is scaled up because the other half of the subtree is missing. In addition to this,* ▼ *gains value* $3/2$ *for playing* h *and* 1 *for playing* t *at* B_2*, accounting for that missing subtree.*

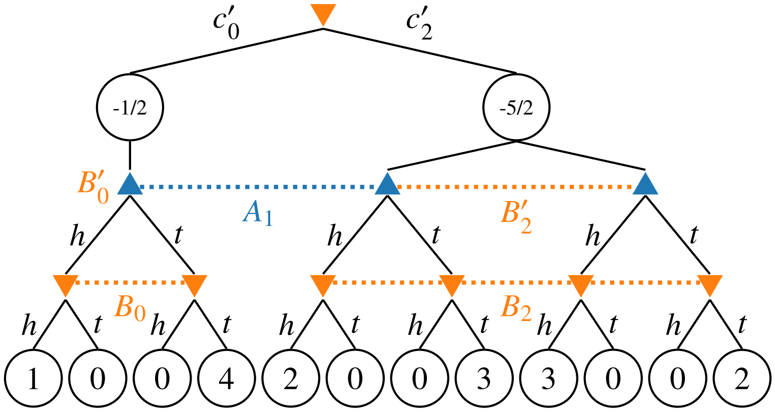


Figure 3.5: *The subgame for* 1*-KLUSS at* A_1*.*

3.5 Example of How KL(U)SS Works

Figure 3.2 shows a small example game. Suppose that the ▲-blueprint is uniform random, and consider an agent who has reached infoset A_1 and wishes to perform subgame solving. Under the given blueprint strategy, ▼ has the following counterfactual values: $1/2$ at B'_0 and C'_4, and $5/2$ at B'_2.

The common-knowledge maxmargin gadget subgame $\Gamma[A_1^\infty]$ can be seen in Figure 3.3. The 1-KLSS and 2-KLUSS maxmargin gadget subgames can be seen in Figure 3.4 and Figure 3.5 respectively.

The advantage of KL(U)SS is clearly demonstrated in this example: while both KLSS and common-knowledge subgame solving prune out the subgame at node 5, KL(U)SS further prunes the subgames at node 4 (because it is outside the order-2 set A_1^2 and thus does not directly affect A_1), and KLSS further prunes node 3 (because it only depends on ▼'s strategy in the subgame—and not on ▲'s strategy—and thus can be added to a single row of B).

3.6 Description of our AI Agent *Obscuro* and the New Algorithms Therein

The technical innovations of *Obscuro* are in its search algorithms. At a high level, they operate as follows. At all times, the program maintains the full set P of possible positions[3.8] given the observations that it has seen so far in the game, as well as a partial game tree $\hat{\Gamma}$ consisting of its calculations from the previous move. At the beginning of the game, P contains only the starting position s_0, and $\hat{\Gamma}$ consists of a single node s_0, since the program has done no calculation. Although P is small enough to fit in memory (usually $|P| \le 10^6$), it is too large to feasibly allow nontrivial reasoning about every single position in P on every move. Therefore, the program instead samples a small subset $I \subseteq P$ at random, whose size is no more than a few hundred positions.

Given a subset I, the program at a high level executes the following steps.

1. Construct an imperfect-information subgame Γ incorporating the saved computation from the previous move ($\hat{\Gamma}$), as well as the positions in the sampled subset I.
2. Compute an (approximately) optimal strategy profile (*i.e.*, an approximate Nash equilibrium) of Γ.
3. Use the Nash equilibrium to expand the game tree Γ.
4. Repeat the above two steps until a time budget is exceeded.
5. Select a move.

We now elaborate on each step individually. Full detail can be found in Section 3.8.

[3.8] A position describes where pieces are as well as the castling and *en passant* rights.

3.6.1 Step 1: Generating the Initial Game Tree at the Beginning of a Turn

The imperfect-information subgame Γ is constructed from the old game tree $\hat{\Gamma}$ and the sampled additional positions $s \in I$ according to KLUSS.

3.6.2 Step 2: Equilibrium Computation

The remaining steps are inspired by the *growing-tree counterfactual regret minimization* (GT-CFR) algorithm [267]: a game tree Γ is simultaneously solved using an iterative equilibrium-finding algorithm and expanded using an expansion policy.

For equilibrium finding we PCFR+. At all times t, PCFR+ maintains a profile $(\boldsymbol{x}^t, \boldsymbol{y}^t)$, where $\boldsymbol{x}^t$ is our strategy and $\boldsymbol{y}^t$ is the opponent's strategy.

PCFR+ has only been proven to converge *in average strategies*. That is, the empirical strategy profile $(\bar{\boldsymbol{x}}^t, \bar{\boldsymbol{y}}^t) := (\frac{1}{t}\sum_{s=1}^t \boldsymbol{x}^s, \frac{1}{t}\sum_{s=1}^t \boldsymbol{y}^s)$ converges to Nash equilibrium as $t \to \infty$. However, instead of computing the empirical average strategy, we circumvent this step and maintain only the last iterate $(\boldsymbol{x}^t, \boldsymbol{y}^t)$. There are several reasons for this choice, which are detailed in Section 3.8.

3.6.3 Step 3: Expanding the Game Tree

Nodes are selected for expansion by using carefully-designed *expansion policies* that balance exploration and exploitation. Our program chooses a node to expand by the following process. Fix one player to be the *exploring player*. (The choice of which player is exploring alternates: on odd-numbered iterations, P1 is the exploring player; on even-numbered iterations, P2 is the exploring player.) For this exposition, we will take P1 to be the exploring player. The *non-exploring* player will play according to its current strategy as computed by PCFR+, in this case $\boldsymbol{y}^t$. The *exploring* player will play a perturbed version $\tilde{\boldsymbol{x}}^t$ of its current strategy $\boldsymbol{x}^t$. The strategy $\tilde{\boldsymbol{x}}^t$ is designed to balance between exploitation and exploration. *Exploitation* here means playing actions with high possible reward, that is, actions that have positive probability in $\boldsymbol{x}^t$. *Exploration* means assigning positive probability to every possible action, to hedge against the possibility that the current tree incorrectly estimates the value of the action due to lacking search depth. For this, we use a method based on the *polynomial upper confidence bounds for trees* (PUCT) algorithm [271]. Finally, a leaf node of the current tree Γ is selected for expansion according to the strategy profile $(\tilde{\boldsymbol{x}}^t, \boldsymbol{y}^t)$.

One major difference between our algorithm and the GT-CFR algorithm lies in having only one player use the exploring strategy $\tilde{\boldsymbol{x}}^t$, rather than both. Intuitively, this remains sound, because tree nodes that *neither* player plays to reach are irrelevant to equilibrium play. Thus, allowing one player to play directly from their equilibrium strategy (here, $\boldsymbol{y}^t$) allows the tree expansion to be more focused.

Once a leaf node z is chosen by the above process, its children are evaluated by a node heuristic and added to the game tree. The node heuristic is an estimate of the perfect-information value of z, as evaluated by the chess engine *Stockfish 14* [276]. If z is the first node in its infoset that has been expanded, a local regret minimizer is created for PCFR+, and it is initialized to pick the action with highest value according to the node heuristic.[3.9]

3.6.4 Step 4: Repeat

The above two steps are repeated, in parallel using a multi-threaded implementation, until a time budget is exceeded. Our implementation uses one thread running CFR and two threads expanding the game tree, which is shared across all three threads. The node expansion threads use locks to avoid expanding the same node, but the equilibrium computation thread uses no locks and only works on the already-expanded portion of the game tree. The time budget is set heuristically based on the amount of time remaining on the player's clock. Once the time budget is exceeded, the tree expansion threads (Step 3) are stopped first, and then, after a delay, the equilibrium computation thread (Step 2). The added time allocated to equilibrium computation is present so that a more precise equilibrium can be computed without the tree constantly changing.

3.6.5 Step 5: Selecting a Move

After those computations have stopped, a move is selected based on the (possibly mixed) strategy that PCFR+ has computed. Instead of directly sampling from this distribution, we first *purify* it [123]—that is, we limit the amount of randomness. In particular, we sample from only the m highest-probability actions, where $1 \le m \le 3$ is chosen based on the computed strategies. We only allow mixing ($m > 1$) when the algorithm believes that its computed strategy is *safe*—intuitively, this is when the algorithm's final strategy $\boldsymbol{x}^t$ can guarantee expected value at least as good as what the algorithm thought to be possible before the turn. This purification technique made a significant difference in practice, detailed via an ablation test in Section 3.7.

3.7 Experimental Evaluation

To evaluate our techniques, we conducted several experiments. The first was a 1,000-game match against an early version of *Obscuro*, which we will refer to as *ZS21* in reference to its date of publication [301]. Our new AI scored 85.1% (+834 =33 -133)[3.10], confidently establishing its superiority. ZS21 used KLSS instead of KLUSS, iterative-deepening linear programming instead of GT-CFR, and none of the techniques against which we perform ablations in Section 3.7.2.

We then ran two experiments against human players. The first of these was a series of games against human players of varying skill levels. *Obscuro* played a total of 117 games (with time control 3 minutes + 2 seconds per move).[3.11] The skill levels of the players, measured by their chess.com Fog of War chess ratings, ranged from 1450 to 2006. We excluded 17 of the games

[3.9] Theoretically, the guarantees of PCFR+ do not depend on the initialization, which can be arbitrary. However, practically, we find that initializing to a "good guess" of a good action leads to faster empirical convergence to equilibrium. More details can be found in Section 3.8.

[3.10] This notation means 834 wins, 33 draws, and 133 losses.

[3.11] This time control was selected because it was the most popular time control played on the most popular website for FoW chess (chess.com) at the time of the experiment. While in regular chess both fast and slow games are common, in FoW chess slow games are typically not played.

for various reasons such as disconnections, the opponent leaving before the game finished, or the opponent clearly losing on purpose, leaving 100 completed games. *Obscuro* scored 97% (+97 =0 -3), establishing conclusively that it is stronger than humans of this level.

Finally, we invited the top FoW chess player to a 20-game match (again at 3+2 time control). At the time of our match,[3.12] this player was rated 2318 and ranked #1 on the chess.com Fog of War blitz leaderboard. In this match, *Obscuro* scored 80% (+16 =0 -4), a conclusive and statistically significant[3.13] victory against the world's strongest player. We thus conclude that *Obscuro* is superhuman.

The 20 games played against the top human are available at this link. A curated sample of particularly interesting games from our 100 games played against humans of varying skill levels, including all three games lost by *Obscuro*, is available at this link.

3.7.1 Hardware

Obscuro, for its human matches, ran on a single desktop machine with a 6-core Intel i5 CPU. Ablations and further matches were run on an AMD EPYC 64-core server machine using 10 cores (5 per side). We now report statistics about the computational performance of *Obscuro*. These statistics were collected over the course of a 1,000-game sample, at a time control of 5 seconds per move.[3.14]

- Average game length: 116.6 plies (58.3 full moves)
- Average search depth: 10.7 plies
- Average search tree size: 1,070,552 nodes, 14,404 infosets
- Average search tree size carried over from previous search: 181,421 nodes, 3,162 infosets
- Average number of possible positions: 17,264

3.7.2 Ablations

In addition to the experiments described earlier in this chapter, we conducted multiple experiments with *Obscuro* as follows. In each of these experiments, we turned off one or more of the new techniques introduced in this chapter in order to evaluate the contributions of the different techniques to the performance of *Obscuro*. All ablations were run at a time control of 5 seconds per move. Recall that *Obscuro* with all techniques turned on scored 85.1% against ZS21 and 80% against the top human.

1. *Purification off.* This version allowed mixing among all stable actions, even if the current margin is negative or there are more than three of them.

[3.12] *I.e.*, as of the rating list on August 16, 2024 [1]

[3.13] $p = 0.011$ using an exact binomial test.

[3.14] For this and all other AI-vs-AI matches in this chapter, the stated time control, usually 5 seconds per move, is the time limit allocated to the main search loop, and does *not* include the time it takes to enumerate the set of all legal positions.

In a 1,000-game match, *Obscuro* scored 70.2% (+662 =79 -259).

2. *KLUSS off.* In this version, the strategies in infosets not touching our infoset were frozen, as in 1-KLSS.

   In a 1,000-game match, *Obscuro* scored 58.0% (+532 =96 -372).

3. *Non-uniform Resolve distribution off.* In this version, when using *Resolve*, the distribution over root nodes is set uniformly to $\alpha_J = 1/m$, as in ZS21.

   In a 10,000-game match, *Obscuro* scored 53.3% (+4595 =1478 -3927).

4. *One-sided GT-CFR off.* In this version we use the two-sided node expansion algorithm proposed by the original GT-CFR paper [267].

   In a 10,000-game match, *Obscuro* scored 53.3% (+4535 =1583 -3882).

5. *Two-sided GT-CFR only, against ZS21.* In this ablation, we turned off all the above improvements 1, 2, 3, and 4, and matched the resulting agent against that of ZS21. This serves to isolate the effect of using GT-CFR compared to using the LP-based equilibrium computation and iterative deepening node expansion as in ZS21.

   In a 1,000-game match, the GT-CFR version scored 72.6% (+711 =30 -259).

All results are highly statistically significant ($z > 5$). The results suggest that each improvement played a significant role in the improvement of *Obscuro* over the previous state-of-the-art AI.

3.7.3 Further experiments

Finally, we conducted several other experiments to test different properties of *Obscuro*.

1. *Weaker evaluation function.* To test the impact of the evaluation function, we hand-crafted a simple evaluation function that takes into account only the material difference and number of squares visible to each player. We substituted this evaluation function in place of *Stockfish 14*'s neural network-based evaluation function, creating a new agent that we call *simple-eval Obscuro*. This evaluation function is very simplistic, and would not be well-suited to regular chess. We tested *simple-eval Obscuro* against both *Obscuro* and ZS21.

   In a 1,000-game match, *Obscuro* scored 81.9% (+787 =63 -150) against *simple-eval Obscuro*.

   In a 10,000-game match, *simple-eval Obscuro* scored 55.0% (+5258 =486 -4256) against ZS21.

   This experiment shows that the evaluation function has a significant impact on the performance of *Obscuro*. Yet, the search algorithm is also vital: even a simplistic evaluation function with our improved search techniques is enough to be superior to ZS21.

2. *Random agent.* As a sanity check, we also tested *Obscuro* against a random opponent.[3.15]

[3.15] The only realistic way for *Obscuro* to lose to a random opponent is by not defending against **Qa4+** or **Qa5+**

In a 1,000-game match, *Obscuro* scored 100% (+1000 =0 -0).

3. *Time scaling.* To test the effect of the time limit on the performance of *Obscuro*, we tested versions of *Obscuro* with different time limits against each other. The results were as follows. All matches consisted of 10,000 games.

   Obscuro with $\frac{1}{8}$ s/move scored 56.4% (+5162 =943 -3895) against *Obscuro* with $\frac{1}{16}$ s/move.

   Obscuro with $\frac{1}{4}$ s/move scored 56.5% (+5031 =1231 -3738) against *Obscuro* with $\frac{1}{8}$ s/move.

   Obscuro with $\frac{1}{2}$ s/move scored 56.7% (+4923 =1503 -3574) against *Obscuro* with $\frac{1}{4}$ s/move.

   Obscuro with 1 s/move scored 54.0% (+4617 =1566 -3817) against *Obscuro* with $\frac{1}{2}$ s/move.

   Obscuro with 2 s/move scored 53.7% (+4589 =1561 -3850) against *Obscuro* with 1 s/move.

   Obscuro with 4 s/move scored 52.3% (+4463 =1530 -4007) against *Obscuro* with 2 s/move.

   Obscuro with 8 s/move scored 52.4% (+4501 =1482 -4017) against *Obscuro* with 4 s/move.

   Obscuro with 16 s/move scored 52.3% (+4448 =1563 -3989) against *Obscuro* with 8 s/move.

These results, converted to the standard Elo scale, are visualized in Figure 3.6. As expected and in line with known results for other settings (*e.g.*, for regular chess [272]), increasing search time has a significant impact on playing strength, but with diminishing returns.

3.7.4 Exact tabular experiments with 1-KLSS

We conducted experiments on various small and medium-sized games to test the practical performance of 1-KLSS. To do this, we created a blueprint strategy for ▲ that is intentionally weak by forcing ▲ to play an ϵ-uniform strategy (i.e., at every infoset I, every action a must be played with probability at least ϵ/m where m is the number of actions at I). The blueprint is computed as the least-exploitable strategy under this condition. During subgame solving, the same restriction is applied at every infoset except the root, which means theoretically that it is possible for any strategy to arise from nested solving applied to every infoset in the game. The mistakes made by playing with this restriction are highly systematic (namely, playing bad actions with positive probability ϵ); thus, the argument at the end of Section 3.4 suggests that we may expect order-1 subgame solving to perform poorly in this setting.

We tested on a wide variety of games, including some implemented in the open-source library *OpenSpiel* [192]. All games were solved with Gurobi 9.0 [139], and subgames were solved in a nested fashion at every information set using *maxmargin* solving. We found that, in all practical games (i.e., all games tested except the toy game 100-matching pennies) 1-KLSS in practice always decreases the exploitability of the blueprint, suggesting that 1-KLSS decreases exploitability in practice, despite the lack of matching theoretical guarantees. Experimental results can be found in Table 3.7. We also conducted experiments at $\epsilon = 0$ (so that the blueprint

in the opening as previously discussed, which happens with only very small probability. As previously discussed, occasionally losing to a weak (here, random) player would not in itself evidence that *Obscuro* is playing suboptimally, since even an exact equilibrium player should lose to a random player with positive probability.

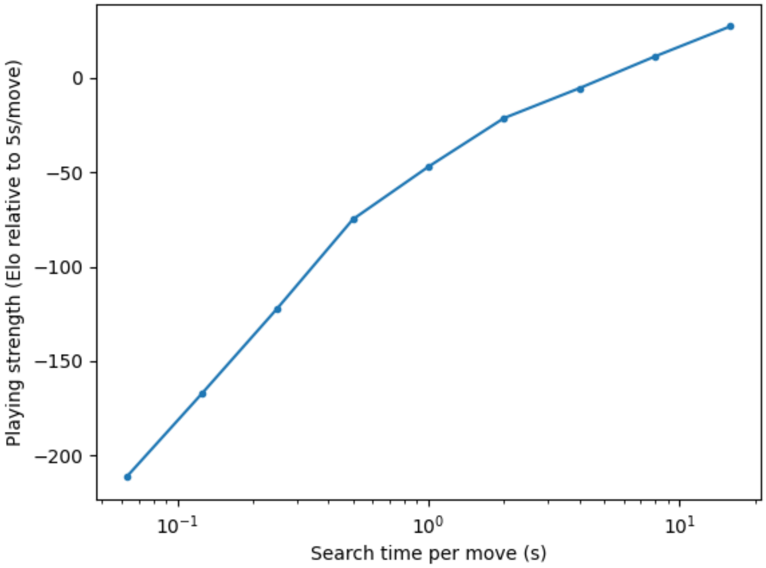


Figure 3.6: *Visualization of time scaling of* Obscuro. *The y-axis is relative to the playing strength of* Obscuro *with 5 seconds per move.*

is an exact NE strategy, and all the subgame solving needs to do is not inadvertently ruin the equilibrium), and found that, in all games tested, the equilibrium strategy was indeed not ruined (that is, exploitability remained 0). Gurobi was reset before each subgame solution was computed, to avoid warm-starting the subgame solution at equilibrium.

All games in this subsection, except k-matching pennies (which is described in the paper body), are implemented in *OpenSpiel* [192].

Kuhn poker [187] and *Leduc poker* [275] are small variants of poker. In Kuhn poker, each player is dealt one of three cards, and a single round of betting ensues with a fixed bet size and a one-bet limit. There are no community cards. In Leduc poker, there is a deck of six cards. Each player is dealt a hole card, and there is a single community card. There are two rounds of betting, one before and one after the community card is dealt. There is a two-bet limit per round, and the raise sizes are fixed.

Abrupt dark hex is the board game *Hex*, except that a player does not observe the opponent's moves. If a player attempts to play an illegal move, she is notified, and she loses her turn.

k-card Goofspiel is played as follows. At time t (for $t = 1, \dots, k$), players simultaneously place bids for a prize of value v_t. The possible bids are the integers $1, \dots, k$. Each player must use each bid exactly once. The higher bid wins the prize; in the event of a tie, the prize is split. The players learn who won the prize, but do not learn the exact bid played by the opponent. In the *random card order* variant, the list $\{v_t\}$ is a random permutation of $\{1, \dots, k\}$. In the *fixed increasing card order* variant, $v_t = t$.

	exploitability		
game	blueprint	after 1-KLSS	ratio
2x2 Abrupt Dark Hex	0.07	0.06	1.09
4-card Goofspiel, random order	0.17	0.08	2.2
4-card Goofspiel, increasing order	0.17	0.0	∞
Kuhn poker	0.01	1.5	8.3
Kuhn poker (ϵ-bet)	3.5	0.0	∞
3-rank limit Leduc poker	0.02	0.02	1.09
3-rank limit Leduc poker (ϵ-fold)	6.5	5.7	1.09
3-rank limit Leduc poker (ϵ-bet)	9.7	9.6	1.01
Liar's Dice, 5-sided die	0.18	0.13	1.45
100-Matching pennies	1.3	9.8	0.13

Table 3.7: *Experimental results in medium-sized games. Reward ranges in all games were normalized to lie in* $[-1, 1]$. Ratio *is the blueprint exploitability divided by the post-subgame-solving exploitability. The value* ϵ *was set to* 0.25 *in all experiments, but the results are qualitatively similar with smaller values of* ϵ *such as* 0.1. *In the* ϵ*-bet/fold variants, the blueprint is the least-exploitable strategy that always plays that action with probability at least* ϵ *(Kuhn poker with* 0.25*-fold has an exact Nash equilibrium for P1, so we do not include it). Descriptions and statistics about the games can be found in the appendix.*

Liar's dice. Two players roll independent dice. The players then alternate making claims about the value of their own die (e.g., "my die is at least 3"). Each claim must be larger than the previous one, until someone calls *liar*. If the last claim was correct, the claimant wins.

The experimental results suggest that despite the behavior of 1-KLSS in our counterexample to Proposition 3.3, in practice 1-KLSS can be applied at every infoset without increasing exploitability despite lacking theoretical guarantees.

3.8 Further Details about *Obscuro*

3.8.1 Dealing with Lost Particles

Upon reaching a new infoset I in a playthrough, because we are performing non-uniform iterative deepening, it is likely that some nodes in I do not appear in the subgame search tree. It is even possible that *no* node in I appears in the subgame search tree. For this reason, in addition to nested subgame solving, we maintain the exact set I (up to transpositions, as per Section 3.4). The set I rarely exceeds size 10^7, making it reasonable to maintain and update in real time. Let I' be the set of game nodes currently being considered by the player. We set a lower limit L on the number of "particles" (subgame root states) being considered. If $|I'| \le L$ and $I' \subsetneq I$, then we sample at most $L - |I'|$ nodes uniformly at random without replacement from $I \setminus I'$, and add them as roots of the subgame tree. At such nodes h, our agent assumes that the opponent knows the exact node. The

				average $\lvert I^k \rvert$ for $k = \dots$				
game	nodes	infosets	diameter	1	2	3	4	∞
2x2 Abrupt Dark Hex	471	94	13	5.23	12.00	18.17	22.04	29.58
4-card Goofspiel, random	26773	3608	4	5.84	8.90	9.19	9.20	
4-card Goofspiel, increasing	1077	162	4	5.83	9.05	9.31	9.32	
Kuhn poker	58	12	3	2.50	3.50	4.00		
3-rank limit Leduc poker	9457	936	3	6.14	14.71	15.40		
Liar's Dice, 5-sided die	51181	5120	2	7.00	15.00			
100-Matching pennies	701	101	99	3.63	4.29	4.93	5.57	35.97

Table 3.8: *Game statistics of games in this subsection. The averages are taken over* nodes*; that is, they are the average size of I^k for uniformly-sampled nodes h in the game tree, where I is the infoset containing h. "diameter" is the diameter of the infoset hypergraph—equivalently, the smallest k such that $I^k = I^\infty$ for all I. We note that the main purpose of the experiments on these games was to demonstrate practical safety, not necessarily to exhibit games of large diameter or in which the average common-knowledge size is necessarily large.*

alternate payoff at h is defined to be $\min(\tilde{u}(h), \hat{u})$ where $\hat{u}$ is the estimate of our current value in the game, as deduced from the previous subgame solve. This alternate payoff setting prevents the agent from over-valuing states with $\tilde{u}(h)$ values that are unattainable due to lack of information.

We set $L = 200$, which we find gives a reasonable balance between achievable depth in subgame solving and representative coverage of root nodes. To prevent the set I from growing too large to manage, we explicitly incentivize the agent to discover information: for each action a available to the agent at the root infoset of the subgame, let $H(a)$ denote the binary entropy of the next observation after playing action a, assuming that the true root is uniformly randomly drawn from I'. Then we give an explicit penalty of $2^{-H(a)}|I|/M$ if the agent plays action a, where M is a tunable hyperparameter. In our experiments, we set $M = 10^7$. The only purpose of this explicit penalty is to prevent the agent from running out of memory or time trying to compute I; typically $|I|$ is small enough that it is a non-factor and the agent is able to seek information without much explicit incentive.

Performing particle filtering over I^∞ was suggested as an alternative in parallel work [280]. We believe that particle filtering would not work as well as our method in FoW chess. If we maintained I^∞ instead of I, the L particles would have to cover the entire common-knowledge closure I^∞, not just I, which means a coarser and thus inferior approximation of I^∞. In a domain like FoW chess where managing one's own uncertainty of the position is a critical part of playing good moves (since good moves in chess are highly position dependent), this will degrade performance, especially when I^∞ is large compared to I (which will typically be the case in FoW chess).

3.8.2 Choice of Subgame Solving Variant

The choice of subgame solving variant is a nontrivial one in our setting. Due to the various approximations and heuristics used, it is often impossible to make all margins positive in a subgame.

Thus, we make a hybrid decision: we first attempt *reach-maxmargin* subgame solving [36], which is a generalization of maxmargin subgame solving that incorporates the fact that we can give back the gifts the opponent has given us and still be safe (Section 3.3). Using reach reasoning (i.e., mistakes reasoning) gives us a larger safe strategy space to optimize over and thus larger margins. If all margins in that optimization are positive, we stop. Otherwise, we use reach-resolving instead. This makes our agent *pessimistic on offense* (if margins are positive, it assumes that the opponent is able to exactly minimize the margin), and *optimistic on defense* (in the extreme case when all margins are negative, the distribution of root nodes is assumed to be uniform random). This guarantees that all margins are made positive whenever possible, and thus, that at least modulo all the approximations, the theoretical guarantees of Theorem 3.10 are maintained. We find that this gives the best practical performance in experiments.

3.8.3 Better Alternate Values and Gift Values

For alternate values in both *Resolve* and *Maxmargin*, in *Obscuro* we use $u(\boldsymbol{x}, \boldsymbol{y}|J)$ instead of the best-response value $u^*(\boldsymbol{x}|J)$ which is more typically used in subgame solving as we described before. Similarly, we use the counterfactual values $u_{\text{cf}}(\boldsymbol{x}, \boldsymbol{y}; Ja)$ and $u_{\text{cf}}(\boldsymbol{x}, \boldsymbol{y}; J)$ to define the gift instead of the counterfactual best responses $u^*(\boldsymbol{x}|Ja)$ and $u^*(\boldsymbol{x}|J)$, resulting in the gift estimate

$$\hat{g}(J) := \sum_{J'a' \preceq J} [u_{\text{cf}}(\boldsymbol{x}, \boldsymbol{y}; J'a') - u_{\text{cf}}(\boldsymbol{x}, \boldsymbol{y}; J')]^+$$

These changes are for stability reasons: especially late in the tree, the current strategy x may be inaccurate, and the best-response value $u^*(\boldsymbol{x}|J)$ may not be an accurate reflection of the quality of the blueprint strategy x, especially near the top of the tree. Of course, if $(\boldsymbol{x}, \boldsymbol{y})$ is actually an equilibrium of the constructed subgame, then these values are the same.

3.8.4 Better Root Distribution for *Resolve*

When using *Resolve*[3.16] for the subgame solve in games with no chance actions, the standard algorithm for *Resolve* will choose an opponent infoset J uniformly at random from the distribution of possible infosets. In reality, the correctness of *Resolve* does not depend on the distribution chosen, so long as it is fully mixed. To be more optimistic, we therefore use a different distribution. We choose an infoset J via an even mixture of a uniformly random distribution and the distribution of infosets generated from the opponent strategy in the blueprint. That is, the probability of the subgame root being infoset J is

$$\alpha(J) := \frac{1}{2}\left(\frac{\boldsymbol{y}(J)}{\sum_{J'} \boldsymbol{y}(J')} + \frac{1}{m}\right),$$

where m is the number of ▼-infosets in the current subgame and the sum is taken over those same infosets. In other words, the *Resolve* objective becomes

$$\max_{\boldsymbol{x}'} \sum_{J \in \mathcal{J}_0} \alpha(J)[M(\boldsymbol{x}', J)]^-.$$

[3.16] For *Maxmargin*, there is no prior distribution because the adversary picks the distribution.

In this manner, more weight is given to those positions that were found to be likely in the previous iteration, while maintaining at least some positive weight on every strategy.

3.8.5 Better Node Expansion via GT-CFR

Growing-tree CFR (GT-CFR) [267] is a general technique for computing good strategies in games. Intuitively, it works, like PUCT, by maintaining a current game $\tilde{\Gamma}$ and simultaneously executing two subroutines: one that attempts to solve the game $\tilde{\Gamma}$, and one that expands leaf nodes of $\tilde{\Gamma}$. As mentioned in the body, we use PCFR+ for game solving.

For expansion, we use a new variant of GT-CFR which we call *one-sided GT-CFR*, which, unlike PUCT and GT-CFR, may only expand a small fraction of nodes in the tree. As stated in the body, our one-sided GT-CFR algorithm selects the node to expand according to the profile $(\tilde{\boldsymbol{x}}^t, \boldsymbol{y}^t)$, where $\boldsymbol{y}^t$ is the *non-expanding player's current CFR strategy* and $\tilde{\boldsymbol{x}}^t$ is an exploration profile constructed from the expanding player's current strategy.[3.17] As in GT-CFR, the expanding player's strategy $\tilde{\boldsymbol{x}}^t$ is a mixture of a strategy $\tilde{\boldsymbol{x}}^t_{\text{Max}}(a|I)$ derived from the player's current strategy $\boldsymbol{x}^t$ and an exploration strategy $\tilde{\boldsymbol{x}}^t_{\text{PUCT}}(a|I)$ derived from PUCT [271]. In particular, we define

$$\tilde{\boldsymbol{x}}^t_{\text{Max}}(a|I) \propto \mathbf{1}\{\boldsymbol{x}^t(a|I) > 0\}$$

to be the uniform distribution over the support of the current CFR strategy, and

$$\tilde{\boldsymbol{x}}^t_{\text{PUCT}}(a|I) = \mathbf{1}\{a = \operatorname*{argmax}_{a'} \bar{Q}(I, a)\}$$

where

$$\bar{Q}(I, a) = u(\boldsymbol{x}^t, \boldsymbol{y}^t | I, a) + C\sigma^t(I, a)\frac{\sqrt{N^t(I)}}{1 + N^t(I, a)}.$$

Here, C is a tuneable parameter (which we set to 1); $\sigma^t(I, a)$ is the empirical variance of $u(\boldsymbol{x}^t, \boldsymbol{y}^t | I, a)$ over the previous times we have visited I during expansion (with two prior samples of -1 and $+1$ to ensure it is never zero); $N^t(I)$ is the number of times infoset I has been visited during expansion; and $N^t(I, a)$ is the number of times action a has been selected. Finally, as in GT-CFR, we define

$$\tilde{\boldsymbol{x}}^t_{\text{sample}}(a|I) = \frac{1}{2}\tilde{\boldsymbol{x}}^t_{\text{Max}}(a|I) + \frac{1}{2}\tilde{\boldsymbol{x}}^t_{\text{PUCT}}(a|I).$$

Unlike GT-CFR as originally described [267], our one-sided GT-CFR works on the *game tree itself*, not the *public tree*. The public tree in our setting would be difficult to work with since the amount of common knowledge is very low.

Our one-sided GT-CFR, unlike PUCT and GT-CFR [175, 267], is *not* guaranteed to eventually expand the whole game tree. For example, suppose that our game $\tilde{\Gamma}$ is as in Figure 3.2, and that

[3.17] In this presentation, ▲ is the expanding player. When ▼ is the expanding player, the roles of x and y are also flipped. As stated in the body, the expanding player alternates between ▲ and ▼ after every node expansion.

both players are currently playing the strategy “always play left”. Then node $1hh$ is reached by both players, nodes $1ht$ and $1th$ are reached by only one of the two players (▲ and ▼ respectively), and node $1tt$ is reached by neither player. As such, $1tt$ *will not be expanded*, and if the current strategy is an equilibrium, this can be proven without knowing the details of any subtree that may exist at $1tt$.

Nonetheless, we can still show an asymptotic convergence result:

Theorem 3.11. *For any given $\epsilon > 0$, the average strategy profile $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ in one-sided GT-CFR eventually converges to an ϵ-Nash equilibrium of any finite two-player zero-sum Γ.*[3.18]

Proof. Since Γ is finite, eventually one-sided GT-CFR stops expanding nodes. At this time, let $\tilde{\Gamma}$ be the expanded game tree. Since no more nodes are expanded, and CFR is correct, one-sided GT-CFR eventually converges to an approximate Nash equilibrium $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ of $\tilde{\Gamma}$. At this time, it is perhaps the case that there remain unexpanded nodes in the current tree $\tilde{\Gamma}$. However, any such nodes must have been played with asymptotic probability 0 by *both* players; otherwise, if (say) ▼ plays to an unexpanded node h with asymptotically positive probability, then h would have been expanded at some point when ▲ was the expander. Thus, best-response values in $\tilde{\Gamma}$ are the same as they are in Γ, and therefore $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ is also an approximate equilibrium in Γ. □

3.8.6 Evaluating New Leaves

When a (non-terminal) leaf node z of $\tilde{\Gamma}$ is selected, it is expanded. That is, all of its children are added to the tree. To assign utility values the children of z, we run the open-source engine *Stockfish 14* [276], in *MultiPV* mode, at depth 1 on node z, which gives evaluations for all children of z in a single call,[3.19] and clamp its result to $[-1, +1]$ in the same manner as in Chapter 3.

When the children of z are added to the tree, z becomes a nonterminal node and hence will be placed in an infoset. If z is the first node of its infoset to be expanded in $\tilde{\Gamma}$, we also need to initialize a new regret minimizer to be used by PCFR+ at this new infoset. Doing so naively would cause a sort of instability: the evaluation of z will be (approximately) equal to the largest evaluation of any child of z (due to how regular perfect-information evaluation functions work), but PCFR+ normally would initialize its strategy uniformly at random. Thus, the evaluation of z would suddenly change to being the *average* of the evaluations of the children of z, which could be very different from the maximum (for example, if the move at z is essentially forced). To mitigate this instability, we exploit the property that, in CFR (and all its variants, including PCFR+), the first strategy can be arbitrary. Conventionally it is set to the uniform random strategy, but we instead set it by placing all weight on the best child of z as evaluated by *Stockfish*.

[3.18] Technically, $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ is only a partial strategy in Γ, since it does not specify how to play after any unexpanded nodes. However, this is fine: *any* extension of $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ will be an equilibrium of Γ, and unexpanded nodes are not reached by either player.

[3.19] Using a single call has two minor advantages: first, it takes advantage of slight extensions that may be used in Stockfish at low depth; second, it reduces the overhead of calling Stockfish to one call per node being expanded, instead of one call per child of that node.

3.8.6.1 Selecting an Action

As mentioned in the body, *Obscuro* selects its action using the *last iterate* of PCFR+, rather than the average iterate which is known to converge to a Nash equilibrium. We do this for two reasons.

1. The stopping time of the algorithm, due to the inherent randomness of processor speeds, is already slightly randomized. Thus, stopping on the last iterate does not actually stop at the same timestep T every time: it in effect mixes among the last few strategies. Thus, we do not need to actually randomize ourselves to gain the benefit of randomization.
2. PCFR+ is conjectured (*e.g.*, [107]) to exhibit last-iterate convergence as well. Indeed, we measured the Nash gap of the last iterate $(\boldsymbol{x}^t, \boldsymbol{y}^T)$ (in the expanded game $\tilde{\Gamma}$), and the typical Nash gap was approximately equivalent to half a pawn—much less than the reward range of the game. This suggests that assuming last-iterate convergence is not unreasonable for our setting.

3.8.6.2 Strategy Purification

As mentioned in the body, we partially *purify* our strategy before playing. When *Maxmargin* is used as the subgame solving algorithm (*i.e.*, when the margins are all nonnegative), we allow mixing between $k = 3$ actions; when *Resolve* is used, we deterministically play the top action. Moreover, we only allow mixing among actions other than the highest-probability action if they have appeared continuously in the support of $\boldsymbol{x}^t$ for every iteration $t > T_{1/2}$, where $T_{1/2}$ is chosen to be the iteration number when half the time budget elapsed.[3.20] We call such actions "stable". These restrictions reduce the chance that transient fluctuations in the strategy of the player, which occur commonly during game solving especially with an algorithm like PCFR+, would affect the final action that is played. Any probability mass that was assigned to actions that are excluded in the above manner is shifted to the action with highest probability.

3.9 Further Detail and Rules of FoW chess

3.9.1 Rules of FoW chess

FoW chess is identical to regular chess, except for the following differences [4].

- A player wins by capturing the opposing king. There is no check or checkmate. Thus:
    - Moving into (or failing to escape) a check is legal and thus results in immediate loss.
    - Castling into, out of, or through check is legal (though, of course, castling into check loses immediately).
    - Stalemate is a forced win for the stalemating player.

[3.20] It will almost always be the case that $T_{1/2} < T/2$. This is because, as the game tree grows larger, PCFR+ iterations, whose time complexity scales with the size of the game tree, get slower.

  - There is no draw by insufficient material. In particular, KN vs K is a strong position for the KN, and even K vs K is not an immediate draw (although K vs K is drawn in equilibrium except in some literal edge cases where one king is on the edge of the board and cannot immediately escape.)

- After every move, each player observes all squares onto which her pieces can legally move.

- If a pawn is blocked from moving forward by an opposing piece (or pawn), the square on which the opposing piece/pawn sits is *not* observed. Thus, the player knows that the pawn is blocked, but not what is blocking it (unless, of course, some other piece can capture it.)

- If a pawn can capture *en passant*, the pawn that can be captured *en passant* is visible.

  In particular, the above rules imply that both players always know their exact set of legal moves.

- Threefold repetition and 50-move-rule draws do not need to be claimed. In particular, a draw under either rule can happen without either player knowing for certain until it happens and the game ends.

3.9.2 Size of Infosets and Common-Knowledge Sets

Here we elaborate on the discussions about common-knowledge sets and infosets, alluded to in the introduction.

Consider the family of positions in which both sides have spent the first eight moves playing **1. a4 a5 2. b4 b5 ... 8. h4 h5**, and subsequently shuffle all their remaining pieces around their first three ranks. An example of such a position is in Figure 3.9. Each player must have one bishop on a light square (12 ways), one bishop on a dark square (12 ways), one queen, one king, two knights, and two rooks ($22 \cdot 21 \cdot 20 \cdot 19 \cdot 18 \cdot 17/2^2$ ways). When multiplied, this gives a total of approximately $M = 2 \times 10^9$ ways. This is a lower bound on the maximum size of an infoset. For common-knowledge sets, *both* players can arrange their pieces arbitrarily along the first three ranks, yielding approximately $M^2 \approx 4 \times 10^{18}$ different arrangements, which provides a lower bound on the maximum size of a common-knowledge set.[3.21]

Although infosets *can* get this large, they almost never *do* in practical games, because both sides are making effort to obtain information.

We now elaborate on Figure 3.1. In particular, we will show that the two positions in that figure are in the same common-knowledge set. Consider the sequence of positions in Figure 3.10, read in order from top-left to bottom-right. The positions marked A and B are the same as those in in Figure 3.1. Each position is connected to the next one by an infoset of one of the players: the first pair by a White infoset, the second pair by a Black infoset, and so on. A computer search showed that the depicted path, which has length 9, is the shortest path between these two positions.[3.22]

[3.21] These common-knowledge sets are measured with respect to *states*, not *histories*. Measuring common-knowledge sets with *histories* would result in a significantly larger number, because the order of the moves would matter.

[3.22] A similar computer search shows that this is nearly the longest possible shortest path between any pair of nodes after two moves from each side: there is a shortest path of length 10, but no shortest paths longer than that.

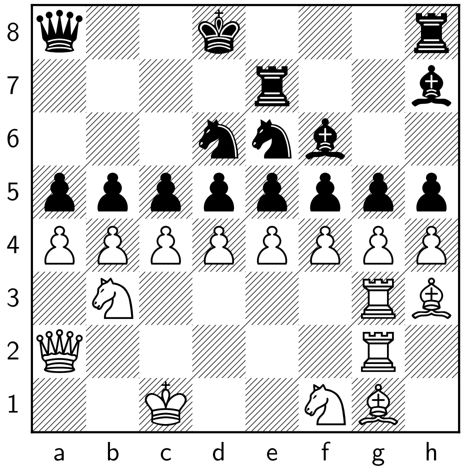


Figure 3.9: *FoW chess position illustrating the existence of large infosets and common-knowledge sets. A full explanation is given in the text.*

Hence, if the true position is A, then the statement Y = "The true position is not B" is 8th-order knowledge for both players. That is, it is true that

$$\underbrace{\text{everyone knows everyone knows ... everyone knows}}_{\text{8 repetitions}} Y$$

yet the same statement would be false if there were 9 repetitions, so Y is not common knowledge.

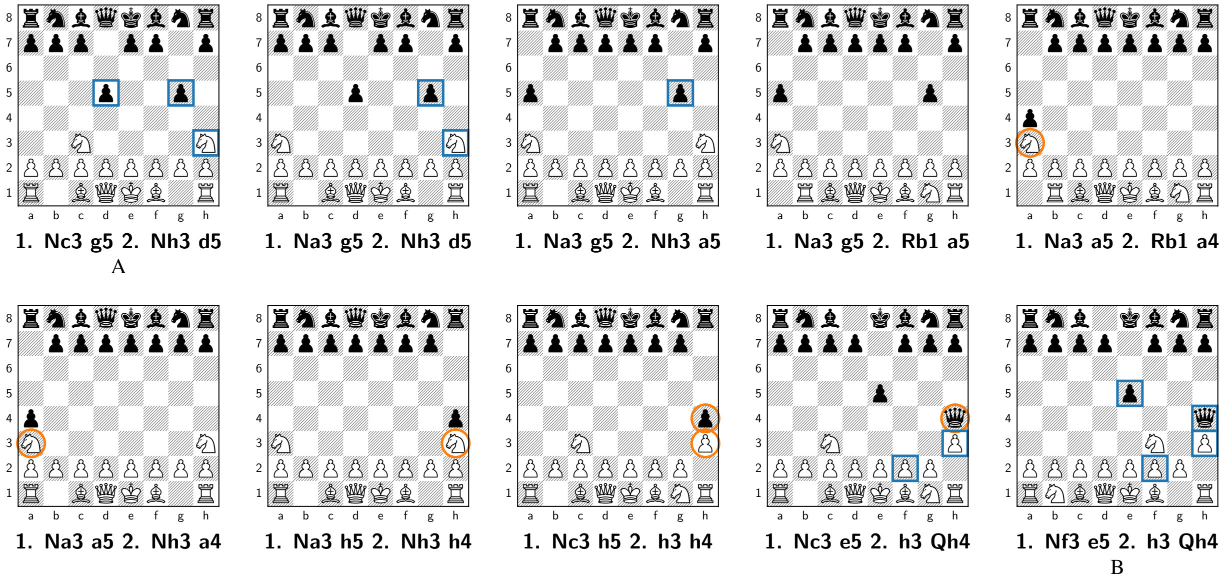
3.9.3 Mixed strategies

Playing a mixed strategy is a fundamental part of strong play in almost any imperfect information game, and it is particularly important in games like FoW chess where there is no private information assigned by chance, such as private cards in poker. Indeed, in small poker endgames, deterministic strategies exist for playing near-optimally [98]. However, in FoW chess, if a player plays a pure strategy that the opponent knows, the opponent would essentially be playing regular chess, because the opponent can predict with full certainty what the player would play. This is a significant disadvantage that will result in a rapid loss against any competent opponent.

Consider, for example, the position in Figure 3.11A. White can win almost a full pawn (in expectation) by mixing between the moves **2. Qa4** with low probability and **2. Nc3** with high probability. No move for Black simultaneously defends the threats against both the king and the pawn. (**2... c6** may look like it does, but after **3. cxd5**, Black cannot recapture the pawn without risking hanging a king or queen.)[3.23]

This necessity of playing a mixed strategy explains why we do not adopt full purification of our strategy and instead opt to allow mixing.

[3.23] *Obscuro* prefers to also include **3. Nf3** and **3. e3** in its mixed strategy to dissuade **2... d4**.



1. Nc3 g5 2. Nh3 d5
A

1. Na3 g5 2. Nh3 d5

1. Na3 g5 2. Nh3 a5

1. Na3 g5 2. Rb1 a5

1. Na3 a5 2. Rb1 a4

1. Na3 a5 2. Nh3 a4

1. Na3 h5 2. Nh3 h4

1. Nc3 h5 2. h3 h4

1. Nc3 e5 2. h3 Qh4

1. Nf3 e5 2. h3 Qh4
B

Figure 3.10: *Sequence of positions illustrating the connectivity between the two positions in Figure 3.1. Circles mark squares that the opponent knows are occupied by* some *piece, but not by* which *piece. A full explanation is given in the text.*

3.9.4 First-Mover Advantage

We evaluated the first-mover advantage in FoW chess by running 10,000 games with *Obscuro* playing against itself at a time control of 5 seconds per move. Of these games, White scored 57.5% (+4935 =1623 -3442). This is, with statistical significance ($z > 5$), larger than the empirical first-move advantage in regular chess, which is about 55% [2]. We believe that the fundamental reason for this discrepancy is the weakness of the **a4-e8** diagonal, as already exhibited in Figure 3.11A, discussed above. This risk presents Black from developing in a natural manner against **1. c4** or **1. d4**, allowing White a healthy opening lead.

Indeed, our 10,000-game sample included 10 games with length 12 ply (6 moves from each player) or fewer; all 10 of these games ended with either Black failing to cover **Qa4+** or White failing to cover **Qa5+**:

- **1. c4 d5 2. Qa4+ d4** 1-0
- **1. c4 c6 2. d4 d5 3. cxd5 Qa5+ 4. Qa4** 0-1
- **1. c4 Nc6 2. d4 d5 3. Qa4 dxc4 4. d5 Nb8** 1-0
- **1. d4 c6 2. c4 d5 3. cxd5 Bf5 4. Qa4 cxd5** 1-0 (*This play-through occurred three times.*)
- **1. d4 c6 2. c3 e6 3. e4 d5 4. e5 c5 5. Qa4+ cxd4** 1-0
- **1. c4 e6 2. d4 c5 3. d5 Qa5+ 4. Nd2 Nf6 5. e4 Nxe4 6. Nxe4** 0-1

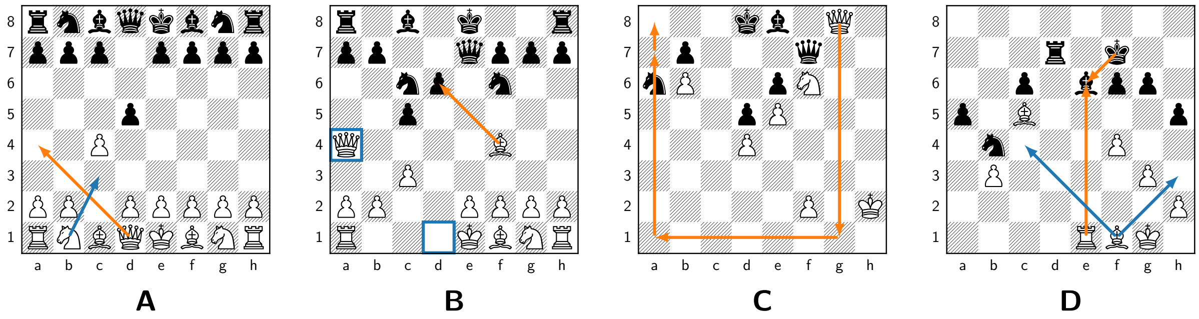


Figure 3.11: *FoW chess positions from actual gameplay illustrating common themes.* **(A)** *Opening position after the common trap* **1. c4 d5?!** **(B)** *An early-game bluff. White bluffs that its attacking bishop is defended by the queen on* **d1**. **(C)** *A highly-risky queen maneuver from a losing position.* **(D)** *An endgame position in which the disadvantaged side sacrifices material for a chance at the opposing king. Details can be found in the text.*

White		Black	
d4	66.4%	**Nc6**	32.5%
c4	29.6%	**c6**	25.1%
e4	1.9%	**e6**	20.7%
Nc3	1.4%	**Nf6**	15.9%
c3	0.4%	**c5**	4.8%
Nf3	0.2%	**d5**	0.9%

Table 3.12: *Distribution of first moves played by* Obscuro *as both White and Black, over a 10,000-game sample. Percentages may not add up to 100% due to rounding.*

- **1. d4 c6 2. Nc3 d5 3. Qd3 Nf6 4. e4 dxe4 5. Nxe4 Qa5+ 6. Nxf6+** 0-1
- **1. c4 d5 2. Qa4+ c6 3. cxd5 Nf6 4. dxc6 Nxc6 5. Nf3 e5 6. Nxe5 Nxe5** 1-0

These games may seem like they contain major mistakes, but that is not so. It is rather likely that *most or all of these play-throughs are part of optimal play*: after all, bluffs must sometimes get called!

In Table 3.12 we give *Obscuro*'s mixed strategy on the first move for both White and Black, over the 10,000-game sample. The above observation about the **a4-e8** diagonal has a large effect on opening choices. We believe that this explains why White strongly prefers opening with **d4** and **c4** rather than **e4** which is equally favored in regular chess, and why Black almost never opens with **d5** and instead prefers to immediately close the dangerous diagonal by moving something to **c6**.

3.9.5 Bluffs

Obscuro bluffs. An example bluff is in Figure 3.11B, which is from the aforementioned 10,000-game sample. White knows that **d6** is defended (in fact, White knows the exact position). Black does not know the location of the white queen (for example, it could be on **d1** instead). This allows White to play **Bxd6**, exploiting the fact that Black cannot recapture without risking losing the queen.

3.9.6 Probabilistic Tactics and Risk-Taking

The existence of hidden information in FoW chess allows tactics that would not work in regular chess. An example of this phenomenon as early as move 2 has already been described above, where mixing allows White to win a pawn after **1. c4 d5**. We now give additional examples.

Figure 3.11C depicts a position encountered during our 20-game match against the top-rated human. *Obscuro* (White) was in a losing position, down a minor piece. It decided to play the highly risky queen maneuver **Qg8-g1-a1-a7-a8**, leaving its own king exposed in order to attempt to hunt the opposing king. This risky tactic worked: the game played out **68. Qg1 Qe7 69. Qa1 Nb8 70. Qa7 Nd7 71. Qa8+ Nxb6** 1-0.[3.24] This sequence of moves heavily exploits the opponent's imperfect information: if Black knew that White was attempting this attack, Black could easily either defend the attack or launch a counterattack on the completely undefended white king.

For another example, consider the position in Figure 3.11D, again from the aforementioned 10,000-game sample, and suppose for the sake of the example that White has perfect information. White faces a slight material disadvantage in an endgame. However, *Obscuro* as White finds the tactical blow **1. Rxe6! Kxe6** upon which mixing evenly between **2. Bc4+** and **2. Bh3+** wins on the spot with 50% probability.

3.9.7 Exploitative vs. Equilibrium Play

The position in Figure 3.11D is also an example of the difference between *exploitative* play and *equilibrium* play in FoW chess. The above tactic has expected value at least 50% against any player, because it wins on the spot with probability at least 50%. It is likely the best move if playing against a perfect opponent. However, against a substantially weaker player, it may be far from the best move: against a weak player, one can argue that the endgame is probably a win even with the slight material disadvantage, whereas the tactic will lead to a significant disadvantage (down three points of material) if it fails to win. Therefore, if one knew the strength of one's opponent, one may opt to not go for this tactic and instead attempt to win the endgame in a "safer" manner. Another example of this phenomenon was also seen above. *Obscuro*, with small probability, can lose in two moves (**1. c4 d5 2. Qa4+ d4**). Any player, no matter how weak, can therefore beat *Obscuro* with positive probability as White by simply playing the above move

[3.24] The immediate **70. Qa8** would have worked in this position as well, but it was not played, likely because it would have risked losing the queen in case the king were on **b8**.

sequence. However, against opponents below a certain level, playing the above moves as Black may be considered a needless risk.

Obscuro does not know or attempt to model the opponent. It will simply play what it believes to be a near-equilibrium strategy. Therefore, it may not do as well against weak players as an agent designed specifically to exploit weak players. This design choice was intentional, and follows other efforts in superhuman game-playing AI such as those mentioned in the introduction, most of which attempted to find and play equilibria rather than to exploit a particular opponent.

3.9.8 Volatility

FoW chess is a highly volatile, highly stochastic game. Indeed, the previous two observations regarding risk taking and exploitative play are evidence of this. Most games, including a majority of our 20 games against the world #1 player, are ultimately decided by one side outright "blundering material" because of lack of knowledge of the opponent's position. We emphasize, however, that this is not a sign of poor quality of play; rather, we believe that strong play in FoW chess involves calculated risk-taking that, with nontrivial probability, leads to such "blunders". More skilled players are better at taking calculated risks while restricting the probability of losing material, and at forcing their opponents into more risky situations.

3.9.9 Endgame Analysis

To make some of the above discussions about mixing, volatility, and equilibrium play more concrete, we include here a partial analysis of the king-vs-king endgame, assuming the starting positon of the kings is common knowledge. While this endgame is an immediate draw in the rules of regular chess (because a lone king cannot checkmate), FoW chess allows such endgames to play out, and not all such endgames are immediately drawn; in fact, the analysis turns out rather intricate already. In the below discussion, 0 is a draw, $+1$ is a certain win for White, and -1 is a certain win for Black.

Claim 3.12. *Suppose that there are two legal moves for the black king that are 1) guaranteed to be safe (i.e., do not put our king next to the opponent's king), and 2) adjacent to each other. Then Black secures at least a draw.*

Proof. Black randomly moves to one of them on their first move, and shuffles between them forever thereafter. The white king cannot approach without being captured half of the time. □

Thus, it remains only to discuss the case where one king is on the edge of the board. Assume, without loss of generality, that this is the black king, and that it is on the 8th rank.

Claim 3.13. *If the white king prevents the black king from immediately moving off the back rank (*e.g.*,* `a6` *and* `a8`*), the equilibrium value is* strictly *positive, regardless of which side is to move.*

Proof. We will show that Black has no strategy that achieves expected value 0. Consider two cases.

Case 1. Black's strategy involves attempting to move off the back rank with positive probability on some move t (but not earlier). Then consider the following strategy for White. Let x**7** (for $x \in \{$**a, b, ..., h**$\}$) be the square on the 7th rank with maximal probability $p > 0$ for the black king after t moves. White places its king on square x**6** before Black's tth move. With probability p, White wins immediately. Otherwise, White runs away downwards, executing the strategy from Claim 1, forcing a draw.

Case 2. Black's strategy is to always stay on the back rank. Then consider the following strategy for White. Let x**8** be the square on the back rank with *minimal* probability $q \le 1/4$ for the black king, at the time when White makes its 8th move. White places its king on x**7** on its 8th move, then moves left and right on the 7th rank until it wins. We claim that White has expected value at least $1 - 2q = 1/2$ with this strategy. To see this, note that, since Black always stays on the back rank, the *parity* of its rank alternates between moves; therefore, if the black king is *not* on x**8**, then White will not lose on its 8th move. Further, also by a parity argument, White will eventually chase down the black king and win the game. □

If the black king is on the edge of the board, it is always the case that either White can force the kings to be two squares apart with common knowledge (Claim 3.13) or Black has a safe pair of adjacent moves (Claim 3.12), so this completes the analysis.

We complete this section by pointing out an interesting special case: If the black king starts in the corner (**a8**), the white king starts on either **b6, c7**, or **c6**, and it is White to move, then White can secure value strictly larger than 1/2: randomize between **Kb6, Kc7, Ka6**, or **Kc8** (whichever are legal moves) on the first move. This wins with probability 1/2 immediately, and otherwise immediately forces the kings to be two squares apart (Claim 3.13).

3.9.10 Conclusions and future research

We presented the first superhuman agent for FoW chess, *Obscuro*. Our agent is completely based on real-time search, so—unlike prior superhuman game-playing AI agents—required no large-scale computation to learn a value function or blueprint strategy. This demonstrates the power of search alone: *Obscuro* required no large-scale computational effort and ran on regular consumer hardware, in contrast to most prior superhuman efforts involving search that we have discussed, which have run on large computing clusters with far more computing power at play time. FoW chess is now the largest (measured by amount of imperfect information) turn-based game in which superhuman performance has been achieved and the largest game in which imperfect-information search techniques have been successfully applied.

Since FoW chess is somewhat similar to regular chess, it was sufficient to combine a perfect-information evaluation function from regular chess (namely, that used by *Stockfish*) with our game-independent state-of-the-art search algorithms for imperfect-information games. Also, *Obscuro* stores at all times the entire set of possible states in memory. While these techniques were feasible for FoW chess—due to the similarity to regular chess and the relatively small infosets—one can imagine even more complex games on which they will not work directly.

Even more complex settings could be tackled by merging our techniques with deep reinforcement learning to learn the evaluation function, instead of using a perfect-information-game evaluation function (in our case, from *Stockfish*), and/or using *continuation strategies* [39] to mitigate game-theoretic issues caused by using node-based evaluation functions in imperfect-information games. In a different direction, higher playing strength and scalability could be achieved by sampling from an infoset using a model of opponent behavior instead of doing so uniformly.

Chapter 4

Solution Concepts, Algorithms, and Complexity of Adversarial Team Games

4.1 Introduction

In two-player zero-sum games, *Nash equilibria in mixed strategies* are the most natural solution concept for modeling rational value-maximizing players. Mixed strategies specify the behavior of a player as a distribution over *pure (deterministic) strategies*. However, the exponential number of such strategies makes the computation of Nash equilibria potentially inefficient. A key assumption to circumvent this issue is *perfect recall*. In a perfect-recall game, the players never forget previously received information or played actions. When this assumption is satisfied,

1. *Kuhn's theorem* [186] states that *mixed strategies* are equivalent to *behavioral strategies*, which are the strategies expressible as a product of distributions over actions at each decision point.
2. The sequence-form representation of the strategy spaces enables efficient computation of Nash equilibria via a wide variety of different methods. In particular, uncoupled learning dynamics such as CFR converge to a Nash equilibrium by employing a regret minimizer at each decision point of the strategy tree.

There have been significant recent speed improvements to CFR-based techniques [38, 104, 283, 315], and other techniques have been built on top of CFR-based techniques, for example, abstraction algorithms [261, 262], subgame solving [36, 37, 39, 121, 129, 221, 222], further enhancing scalability. Notable results on large-scale games include poker [31, 37, 39, 222], Stratego [243], and Diplomacy [94].

This work seeks to extend these techniques beyond the perfect-recall two-player zero-sum setting. In particular, we focus on computing mixed Nash equilibria in the two equivalent settings of *imperfect-recall games* and *adversarial team games*[4.1], for which it is known that computing a

[4.1] This equivalence is formalized in Section 4.2.3.

Nash equilibrium is NP-hard [176].

Two-player zero-sum imperfect-recall games are characterized by players who may forget information at some point in the game. In this case, a mixed strategy corresponds to a distribution over pure strategies, while a behavioral strategy corresponds to a distribution that performs an independent sampling procedure at each decision point. Unlike for perfect-recall games, Kuhn's theorem does not apply in imperfect-recall games: mixed strategies can in general be more expressive than behavioral strategies. Imperfect-recall games have been employed in the literature to compress a game representation through forgetfulness (this is the case of some abstraction techniques [185, 190, 295]), or by considering human-like agents with imperfect memories [50].

Adversarial team games portray two teams of agents facing adversarially. Each team member has utilities identical to her teammates and opposite to members of the opposing team. Effective team coordination is a non-trivial challenge in this setting because team members may have different imperfect information about the current node and no communication channels are available during the game. Intuitively, the player cannot distinguish nodes that are different due to private information revealed to a teammate (such as private cards revealed to them solely). In this case, mixed strategies correspond to strategies coordinated *before the start of the game* through *ex-ante coordination*, while behavioral strategies represent strategies that are not coordinated, in the sense that each agent samples their actions independently from other teammates. Recreational and non-recreational examples of team games include Bridge, security games with multiple defenders and attackers [160], and poker with colluding agents.

Overall, team games are a more common application setting than imperfect-recall games, have many competing works in the equilibrium computation literature, and allow a more intuitive game description. On the other hand, imperfect-recall games yield a cleaner formalism. As these two perspectives are equivalent for our purposes, we choose to adopt an imperfect-recall perspective throughout the rest of this chapter to simplify the notation, while using team games to make more intuitive examples for some of the notions introduced.

The main objective of this chapter is to propose a novel representation for team and imperfect-recall games by constructing an equivalent two-player zero-sum perfect-recall game. This enables the use of all the solving techniques previously developed for perfect-recall two-player zero-sum games.

We now summarize the contributions of this chapter. In Sections 4.3 and 4.3.1, we present an algorithm that converts any two-player zero-sum imperfect-recall game into a strategically-equivalent *perfect-recall game* which we call the *belief game*. We formally prove the equivalence between the two games, and in Section 4.3.2 we show worst-case bounds on the size of the belief game in terms of the number of histories of the original game. In particular, we show that the worst case the number of histories of the belief game is $O(b^{dk})$, where b is the maximum branching factor of the original game, d is its depth, and k is a parameter we introduce called the *information complexity*, which intuitively measures the amount of information that can be forgotten by the player—or, in the case of team games, the amount of *information asymmetry* between players on the team.

In Section 4.4, we introduce a notion of DAG-form decision-making that we use to generalize counterfactual regret minimization (CFR) beyond tree-form games. While we introduce it for the purpose of applying it to imperfect-recall games, we believe it to be of independent interest as well.

In Section 4.5, we use DAG-form decision problems to efficiently represent each player's strategy space in the belief game through a construction we call the *team-belief DAG* (TB-DAG). We show that the TB-DAG representation of a game with imperfect recall can be exponentially smaller than the size of the belief game and that it can be constructed directly from the original game without first constructing the belief game, thus leading to exponentially faster algorithms in the worst case. This construction improves the worst-case efficiency[4.2] of our technique to $O(|\mathcal{H}|(b+1)^k)$, where $|\mathcal{H}|$ is the number of nodes in the original game. We also show that this bound is essentially optimal: under reasonable computational assumptions (namely, the exponential time hypothesis), we show that there cannot exist an algorithm for solving even single-player games of imperfect recall whose runtime is of the form $f(k)\mathsf{poly}(|\mathcal{H}|)$, for *any* function f.

In Section 4.6 we investigate the computational complexity of computing mixed Nash equilibria with imperfect recall. We prove that computing a max-min strategy in mixed or behavioral strategies in games where both players have imperfect recall is Δ_2^P-complete and Σ_2^P-complete respectively.

Section 4.7 presents further discussions comparing different notions presented in this chapter, providing further insights on the technical decisions made.

In Section 4.8, we evaluate our methods empirically by benchmarking our construction on a standard testbed of imperfect-information games, compared to state-of-the-art baselines. We find that our technique allows much faster equilibrium computation when the information complexity k of the game is low.

We have defined equilibrium concepts for team games by using an "equivalent" coordinator game that is two-player zero-sum imperfect recall. It turns out that, in fact, every two-player zero-sum imperfect-recall game Γ' has an ATG whose coordinator game is Γ': indeed, given such a Γ', consider the ATG Γ in which every information set is assigned to a different player. Therefore, team games and imperfect-recall games are in a very strong sense equivalent. All of the results of this section, unless otherwise stated, therefore apply equally to team games and to two-player zero-sum imperfect-recall games.

In this section, we opt to consider the point of view of two-player zero sum games with imperfect recall. A summary of the different equivalent terms that are used in the two settings can be found in Table 4.2.

We now introduce the fundamental contribution of this chapter: a novel technique to compute a mixed Nash equilibrium in two-player zero-sum imperfect-recall games (or equivalently to compute a TMECor in adversarial team games) based on the construction of an equivalent two-player zero-sum game with perfect recall.

[4.2] By *efficiency* here we mean the size of the representation of the strategy spaces of the players. Algorithms such as CFR have per-iteration complexity that scales linearly in this size.

Our technique attains the perfect-recall condition by suitably changing the information available to the players, as well as their action sets. The main intuition behind the belief game is to consider the point of view of a perfect recall player in place of the imperfect-recall one. Differently from the imperfect-recall player, this player reasons only using information the player would never forget due to imperfect recall and chooses an action for every possible information set the imperfect-recall player may be in. The game then transitions by applying the action corresponding to the information set of the current node. Crucially, the perfect-recall player can strategically refine the set of reached nodes over time by carefully considering reachable nodes given the played strategy and the perfect-recall results of her actions.

After introducing the main concepts and the construction algorithm, we prove that the original and the belief games are strategically equivalent. This means that the perfect-recall player we introduce is an equivalent representation of both the imperfect-recall player and the corresponding preplay coordinated team (thanks to the considerations from Section 4.2.3).

4.2 Preliminaries

Since this part deals with equilibrium computation in *team* games and *games with imperfect recall*, we first introduce some notation and definitions that pertain to these. For this part, unless otherwise stated, all games are assumed to be timeable.

4.2.1 Behavioral and Mixed Max-Min Strategies

Recall first the definition of a mixed-strategy Nash equilibrium for a game:

Definition 4.1 (Mixed-strategy Nash equilibrium). In a two-player zero-sum game, a (realization-form) Nash equilibrium is a saddle-point solution to the optimization problem

$$\max_{\boldsymbol{x}\in\mathcal{X}} \min_{\boldsymbol{y}\in\mathcal{Y}} u(\boldsymbol{x}, \boldsymbol{y}).$$

Since this problem is a bilinear saddle-point problem and $\mathcal{X}$ and $\mathcal{Y}$ are convex, the minimax theorem applies, and the maximinization and minimization can be freely swapped without changing the value of the game. The optimal value of the above program is the Nash equilibrium value of the game.

For games with imperfect recall, restricting to *behavioral strategies* is a nontrivial restriction. Recall that a behavioral strategy is a mixed strategy that mixes independently at each information set. Thus, the realization form of a behavioral strategy is obtained by multiplying the probability of picking each action of the player on the $\varnothing \to z$ path. We will use $\hat{\mathcal{X}}_i$ to denote the set of realization-form behavioral strategies of a player i. Recall that Kuhn's theorem states that, in games with perfect recall, behavioral and mixed strategies are realization-equivalent. That is, $\hat{\mathcal{X}}_i = \mathcal{X}_i$.

Definition 4.2 (Behavioral max-min strategy)**.** In a two-player zero-sum game, a *behavioral max-min strategy* $\boldsymbol{x} \in \hat{\mathcal{X}}$ is a solution to the optimization problem

$$\max_{\boldsymbol{x} \in \hat{\mathcal{X}}} \min_{\boldsymbol{y} \in \hat{\mathcal{Y}}} u(\boldsymbol{x}, \boldsymbol{y})$$

The *behavioral max-min value* is the optimal value of the above problem. Since $\hat{\mathcal{X}}$ and $\hat{\mathcal{Y}}$ are not necessarily convex sets, the minimax theorem does not apply, so the maximization and minimization can not necessarily be swapped. Therefore—unlike the mixed-strategy Nash—the behavioral max-min strategy is not an *equilibrium*. Further, in games with imperfect recall, the tree-form decision problem is not a valid representation of the set of realization-form strategies. Therefore, we will need different techniques to tackle such games.

4.2.2 Equivalence Across Games

The contributions presented in this chapter will rely on auxiliary games to represent strategy optimization problems. In order for the results obtained in the auxiliary game to map to the original one we want to solve, we need to define what it means for two games to be equivalent. Let Γ and Γ' be extensive-form games with the same set of players. Let Π_i and Π'_i be player i's pure strategy set in G and G' respectively, and similarly let u_i and u'_i be player i's utility function in G and G' respectively.

Definition 4.3 (Strategic Equivalence)**.** Two games Γ and Γ' are *strategically equivalent* if there are bijective *strategy maps* $\rho_i : \Pi_i \to \Pi'_i$ for each player i such that, for every profile $\boldsymbol{x} \in \Pi$ and every player i we have $u_i(\boldsymbol{x}) = u'_i(\rho(\boldsymbol{x}))$ where $\rho(\boldsymbol{x}) := (\rho_i(\boldsymbol{x}_i))_{i \in \mathbb{N} \setminus 0}$.

This definition is a very strong notion of equivalence: if two extensive-form games are equivalent in the above sense, then every strategy $\boldsymbol{x}_i$ in one of the games is equivalent to some strategy $\rho(\boldsymbol{x}_i)$ in the other game. Thus, in particular, a solution to one game will give a solution to the other game.

4.2.3 Adversarial Team Games

The general framework of adversarial team games has first been studied by von Stengel and Koller [292] in the context of normal form games, while Celli and Gatti [56] first addressed them in an extensive-form setting. Adversarial team games describe situations where multiple agents are organized in two-teams receiving zero-sum payoffs. This chapter focuses on the setting in which no extra communication channel is available to the players during the game, but they are allowed to communicate freely before the start of the game. This means that the only form of coordination across players' strategies available is *preplay coordination*, *i.e.* any coordination has to be prepared before the start of the game.

Adversarial team games can be modeled as extensive-form games as follows:

Definition 4.4 (Adversarial team game)**.** An extensive-form, perfect-recall game is said to be an *adversarial team game* (ATG), or *two-team zero-sum game* iff:

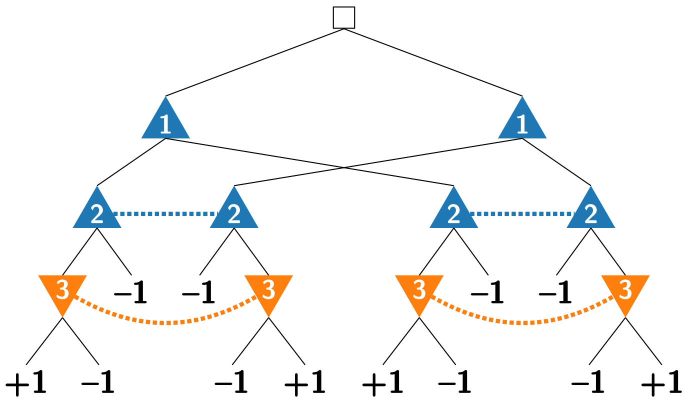


Figure 4.1: *An example of an adversarial team game. There are three players: P1 and P2 are on team $\triangle$, and P3 is on team $\triangledown$. Dotted lines connect nodes in the same information set. The (total) utility of $\triangle$ is listed on each terminal node. The root node is a nature node, at which nature selects uniformly at random.*

- the player set is partitioned in two sets called *teams*, symbolized by $\triangle$ and $\triangledown$. Formally, $[n] = \triangle \cup \triangledown$;
- the utilities of the players belonging to the same team are identical, and the total utilities of the two team are opposites. Formally:

$$\begin{aligned} u_i = u_j &\quad \text{for all } i, j \in \triangle \\ u_i = u_j &\quad \text{for all } i, j \in \triangledown \\ \sum_{i \in \triangle} u_i = - \sum_{j \in \triangledown} u_j & \end{aligned}$$

In adversarial team games, the Nash equilibrium fails to take into account the fact that teams can coordinate among themselves. Indeed, it is possible for there to be a Nash equilibrium in which two teammates could profit by *jointly* switching strategies, but no *individual* player can profit from a unilateral deviation. To take into account these joint deviations, it is most natural to reformulate an adversarial team game as a two-player zero-sum game of imperfect recall, in which a *team coordinator* plays on behalf of all members of that team. In this manner, deviations of the team coordinator correspond to *simultaneous, joint* deviations of all team members. We now formalize this conversion.[4.3]

Definition 4.5 (Coordinator game)**.** Let Γ be an adversarial team game. The *coordinator game* Γ' corresponding to Γ is the two-player zero-sum imperfect-recall game Γ', where

$$\mathcal{I}'_{\blacktriangle} = \bigcup_{i \in \triangle} \mathcal{I}_i, \qquad \mathcal{I}'_{\blacktriangledown} = \bigcup_{i \in \triangledown} \mathcal{I}_i, \qquad u'_{\blacktriangle} = \sum_{i \in \triangle} u_i, \quad \text{and} \quad u'_{\blacktriangledown} = \sum_{i \in \triangledown} u_i.$$

The coordinator game merges all members of a team ($\triangle$ or $\triangledown$) into a coordinator ($\blacktriangle$ or $\blacktriangledown$). Therefore:

[4.3] Recall that we are assuming timeability, so in particular there are no issues of absentmindedness.

- Pure strategies of a coordinator correspond to *pure profiles* of the team.
- Behavioral strategies of a coordinator correspond to *behavioral profiles* of the members of the team. Since behavioral strategies enforce actions at different infosets to be independently sampled, this means that team members can *privately* sample randomness for their own personal use but cannot share that randomness with teammates.
- Mixed strategies of a coordinator correspond to *correlated* strategy profiles of the members of the team. In a correlated profile, team members may *jointly* sample randomness that they use to correlate their actions.

We remark on the role that preplay coordination has in allowing the coordination capabilities modeled by the coordinator game. In fact, before starting the game, players are allowed to jointly sample a pure plan from their coordinator's mixed strategy and then individually play the specified actions at the infoset in which they play. This allows the team to play any randomized strategy of the coordinator effectively.

The coordinator game allows us to define notions of equilibrium specialized for team games:

Definition 4.6. A *team max-min equilibrium with correlation* (TMECor) of an ATG Γ is a mixed-strategy Nash equilibrium of Γ'.

Definition 4.7. A *team max-min equilibrium* (TME) of an ATG Γ is a behavioral max-min strategy of Γ'.

The *TMECor value* and *TME value* are defined analogous to the Nash value and behavioral max-min value. As discussed before, behavioral max-min strategies in Γ' are not equilibria in Γ', so one may wonder about the name "team max-min equilibrium". However, there *is* a sense in which TMEs are equilibria: von Stengel and Koller [292] showed that, at least in the case where $|\triangledown| = 1$, the TMEs are precisely the Nash equilibria of the team in Γ that maximize the utility of team $\triangle$.

An example adversarial team game in which the difference between TME and TMECor is relevant can be found in Figure 4.1. The coordinator game is constructed simply by erasing the player labels, creating a two-player zero-sum game. This game is a simple signaling game: nature selects a bit, which is privately revealed to P1. P1 then communicates a single bit, which is publicly revealed. Then P2 and P3 both attempt to guess nature's selected bit, and $\triangle$ wins if and only if P2's guess is correct. Therefore, the goal of P1 and P2 is for P1 to "securely" communicate the bit to P2 without also revealing it to P3. With a behavioral profile, this is impossible, since P1 and P2 cannot correlate their strategies; therefore, the TME value is $-1/2$. However, if P1 and P2 are allowed to correlate their strategies, they can do the following: jointly flip a coin. If that coin landed heads, P1 communicates the true bit, and P2 plays what P1 communicates. If that coin landed tails, P1 communicates the opposite of the true bit, and P2 plays the opposite of what P1 communicates. In this way, P2 will always play the true bit, but P3 (who does not know the outcome of the correlating coinflip) does not learn any information. Therefore, the value of this strategy for $\triangle$ is 0 (since P2 wins half the time by randomly guessing the bit).

Adversarial Team Games	**Imperfect-Recall Games**
Team $\triangle\triangledown$	Player $\blacktriangle\blacktriangledown$
Correlated team strategy	Mixed strategy
Uncorrelated team strategy	Behavioral strategy
TMECor	Mixed-strategy Nash equilibrium
TME	Behavioral max-min strategy

Table 4.2: *Translation table between terms commonly employed in the adversarial team games and two-player imperfect recall games. The translation happens through the introduction of coordinator games (Definition 4.5).*

4.3 Beliefs and Observations

The main purpose of this section is to formally define *beliefs*, which are sets of nodes $B \subseteq \mathcal{H}$ derived from information sets $\mathcal{I}_i$ of player $i \in \{\blacktriangle, \blacktriangledown\}$. Informally, beliefs are the "information sets" that player i would have if she could not distinguish nodes that cannot be distinguished using information from a later stage. This notion is formalized by putting in the same belief any two nodes that have descendent nodes in the same information set (even if they belong to different information sets). Similarly to information sets:

1. nodes in beliefs would be indistinguishable to i,
2. one action is chosen at each belief, and then this action is followed in all nodes in the belief, and
3. if the player knows that the current node of the game h lies in a set H of candidates, and the player observes that her current belief is B, then the set of candidates can be refined to $B \cap H$ (*i.e.* similarly to information sets, beliefs correspond to observations over the state of the game).

Crucially, beliefs can be organized in the tree-like structure needed by algorithms finding Nash equilibria in two-player zero-sum games, as we will see in Sections 4.3.1 and 4.3.3. This is thanks to the guarantee that once a group of nodes is split among two different distinguishable beliefs, then any group of descendent nodes from one belief will be distinguishable from any group of descendants from the other.

In the following, we formalize the notion of beliefs and observations. We consider a two-player zero-sum game with imperfect recall Γ.

Connectivity graph. We say that two nodes h and h' are *unforgettably distinguished* by i if they do not belong to the same infoset and no pair of children of those two nodes belong to the same infoset, *i.e.* i will never be in an information set where these two ancestors are both possible. This condition guarantees that if the set of candidates is $H = \{h, h'\}$, then the player is able to discern h from h' and will never forget which of the two nodes has been reached in the next steps of the

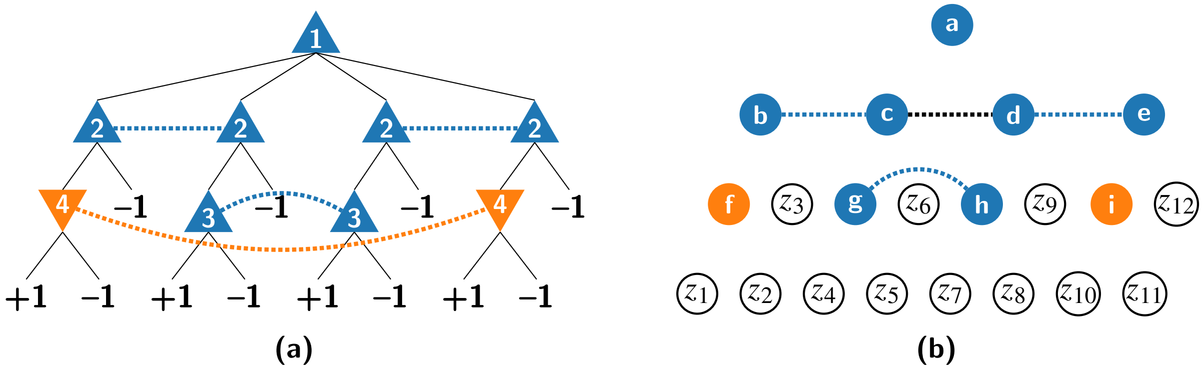


Figure 4.3: *An example of team game (a) (using the same notation as Figure 4.1) and its corresponding connectivity graph for player ▲ (b). Nodes in the two figures correspond in position.*

game.[4.4]

For the purpose of our definitions, we are concerned with pair of nodes that are *not* distinguishable. This can be represented through a *connectivity graph* over $\mathcal{H}$ as follows.

Definition 4.8 (Connectivity graph)**.** The *connectivity graph* $\mathcal{G}_i = (\mathcal{H}, \mathcal{E}_i)$ for player $i \in \{▲, ▼\}$ is the graph with nodes $\mathcal{H}$ and edges $\mathcal{E}_i$, where $(h, h') \in \mathcal{E}_i$ if h and h' are at the same depth in Γ and there exists $I \in \mathcal{I}_i$ such that $h \preceq I$ and $h' \preceq I$.

Consider Figure 4.3b as an example of connectivity graph for a game. Note the blue edges, which correspond to connections due to infosets, and the black edge $c - d$ due to g, h belonging to the same infoset.

Beliefs. Consider now a set H of nodes such that the induced subgraph $\mathcal{G}_i[H]$ is connected. Player i has no way of distinguishing any subset of H from the others, because any node cannot be distinguished from its neighbors. *Beliefs* are defined as these sets of indistinguishable nodes.[4.5]

Definition 4.9 (Belief)**.** A set of nodes $B \subseteq \mathcal{H}$ is a *belief* for player i if the induced subgraph $\mathcal{G}_i[B]$ is connected.

We remark that the timeablility property assumed on Γ implies that any node belonging to the same belief has the same depth. Notice that a direct consequence of the definition of beliefs is that $\{\varnothing\}$ and $\{z\}$ for $z \in \mathcal{Z}$ are singleton beliefs for both teams.

Observations. Consider instead a set H of nodes such that the induced subgraph $\mathcal{G}_i[B]$ has different connected components. In this case, player i can distinguish those components one from the other, thus partitioning H into multiple beliefs. Intuitively, the unforgettable information is

[4.4] h, h' being distinguishable implies that in the corresponding team game any team member can recall whether h or h' was reached upon reaching h or h'.

[4.5] From a team game perspective, beliefs are sets of nodes with the guarantee that once reached all team members know that any node $\mathcal{H} \setminus B$ is not reached, *i.e.* it is team-common knowledge that the game reached a node in B.

enough to distinguish every node in a component from any node in other components. The player can, therefore, exclude nodes from components that are distinguishable from the current reached node. We say that upon reaching a node h among possible candidates H, player i *observes belief* $B \subseteq H$, meaning that player i uses the newly acquired unforgettable information acquired in h to refine its imperfect information from H to B. We formalize this notion of observation through the function $\textsc{SplitBelief}_i$:[4.6]

Definition 4.10 (Observation)**.** The *observation* for player $i \in \{\blacktriangle, \blacktriangledown\}$ when reaching node h among a set H of candidate nodes is:

$$\textsc{SplitBelief}_i(H, h) := \text{the connected component of } \mathcal{G}_i[H] \text{ containing } h.$$

The set of all possible observations given a set of candidates is denoted[4.7] by

$$\mathcal{B}_H := \{\textsc{SplitBelief}_i(H, h) : h \in H\}.$$

An example of observation can be given by considering the team game depicted in Figure 4.3 and a candidate set $H = \{b, c, e\}$. This candidate set is possible when player $\blacktriangle$ plays a strategy where player 1 plays a mixed strategy excluding d from its support. This, in turn, implies that player 2, at the next step, knows that the reached node h in the game is in H. Moreover, player 2 observes her current information set $I = b, c$ if $h \in \{b, c\}$ of $I = \{d, e\}$ if $h \in \{d, e\}$. I can be used to further refine H *as long as the information used will be known at player 1 next.* This is formalized in $\textsc{SplitBelief}_i(H, b) = \textsc{SplitBelief}_i(H, c) = \{b, c\}$ and $\textsc{SplitBelief}_i(H, e) = \{e\}$, which intuitively correspond to the fact that given those candidates, $\blacktriangle$ unforgettably distinguishes b and c from e. From the equivalent team game perspective: player 2 is active and can check her current infoset to distinguish the two beliefs; player 1 has stopped playing and therefore it is not relevant in terms of team knowledge; player 3 either will not play or will know that the current node was c once the game reached g, so she can safely assume that the game is in c. This means that every player distinguishes e from b, c.

Team public states. We compare our notion of beliefs with *public states*, an alternative customarily used in the related literature. A public state P for player i is a connected component of the connectivity graph $\mathcal{G}_i$. The set of all public states of i is denoted as $\mathcal{P}_i$.

Public states identify sets of nodes that are distinguishable to a player without considering a possibly pruned subgraph of $\mathcal{G}_i$ as instead done for team observations. Therefore, every belief is contained in a public state. In Figure 4.3 we have that $\mathcal{P}_\blacktriangle = \{\{a\}, \{b, c, d, e\}, \{g, h\}, \{f\}, \{i\}\} \cup \{\{z\} : z \in \mathcal{Z}\}$.

[4.6] In team games, the belief returned by $\textsc{SplitBelief}_i$ is the team-common knowledge update happening when reaching h among a set of candidates H.

[4.7] We remark that the belief-based constructions employed by this chapter would also work when allowing $\textsc{SplitBelief}_i$ to return any superset of connected components. For example, in the framework of *factored-observation games* [182], it is valid to define $\textsc{SplitBelief}_i$ using the explicitly-given public observations. However, since the efficiency of the proposed algorithms depends on the size of the beliefs employed, we opt not to allow, by definition, the use of beliefs larger than needed. As we show in Section 4.3.2, any reduction in the size of the beliefs in a game brings exponential benefits in the size of the belief game obtained.

Public states are the customarily adopted alternative to observations when partitioning a set H of candidates in beliefs by splitting $\mathcal{H}$ in $\{\mathcal{H} \cap P : P \in \mathcal{P}_i\}$. However, public states may return a coarser partition than the one returned by observations, as the absence of specific nodes from H may disconnect components in $\mathcal{G}$. We will, therefore, use observations in place of public states whenever possible. An example illustrating the difference between the two definitions is available in Section 4.7.1.

Prescriptions. Restricting the information available to player i to her beliefs also affects the set of actions available. In fact, multiple infosets may intersect a given belief, and the player does not know in which infoset she finds herself. Therefore, she does not know what actions are available to her.

We overcome this issue by associating to each belief B a set of meta-actions $\mathcal{A}_i(B)$ such that an action is specified for each possible infoset that intersects the belief. We call such structured meta-actions *prescriptions* and use a symbol $\boldsymbol{a}$ to indicate them. The concept is formally defined as follows.

Definition 4.11 (Prescription)**.** Consider a belief B of a player $i \in \{▲, ▼\}$. A *prescription* $\boldsymbol{a}$ is a selection of one action at each infoset having a nonempty intersection with B:

$$\boldsymbol{a} \in \bigtimes_{I \in \mathcal{I}_i[B]} \mathcal{A}(I) \quad \text{where} \quad \mathcal{I}_i[B] = \{I \in \mathcal{I}_i : I \cap B \neq \varnothing\}.$$

Given a prescription $\boldsymbol{a}$ for a belief B and an infoset I such that $I \cap B \neq \varnothing$, we denote as $\boldsymbol{a}(I)$ the action relative to infoset I which is specified by prescription $\boldsymbol{a}$. Note that we have *empty prescriptions* at beliefs containing no active nodes for a player.

As we will see in the next section, our equivalent belief game introduces one perfect-recall player per team, with information sets associated with beliefs corresponding to the perfect-recall part of the information available to this unique player. Prescriptions will allow this player to have an identical expressive power in terms of actions without accessing the exact information set of the player, which is her imperfect-recall information. Moreover, specifying a prescription at each reached belief for i incrementally defines a pure strategy of player i. This allows us to consider a reduced set of candidate nodes H for the reached node h from which the belief is observed, as a non-played action implies that all the descendant nodes are not reached and, therefore, excluded from the candidates.

For example, consider a 3-player poker instance where two players collude to form a team. At any time of the game, we can consider the point of view of a team coordinator, who acts as the single imperfect recall player. We can imagine this coordinator as sitting at the same table as the players, and therefore, she cannot access the private cards given to the players but can access the same public information as the players, that is, the bet, fold, and check actions of the players. Her belief at any point regards the private cards that each team member has. At the start of the game, this belief is uniform over all pairs of cards, as no information regarding these cards is available from an external point of view. The coordinator emits prescriptions for the players to follow as the game progresses. Since the coordinator does not know the card held by a player, she has to prescribe an action for each possible card the current player may hold. The player receives this prescription and follows the part of it that matches the private card. By observing the action played by the player, the coordinator can exclude from her belief the cards for which she prescribed different actions. While there are no means of communicating prescriptions during play at the poker table, this mechanism can be implemented *ex ante*; that is, each team of players jointly samples a pure strategy of this coordinator before the start of the game, and each member simulates the coordinator locally.

Information complexity. We quantify the number of information sets reaching a belief through the notion of *information complexity* k. This quantity will allow us to bound the size of the belief games in Sections 4.3.2 and 4.5.1.

We first characterize the notion of *remembered* information sets and the set of *last-infosets* at a node. Intuitively, an infoset J remembers another infoset I if reaching a node in J implies traversing a node in I and picking a specific action. Therefore, knowing to be at a node in J allows the player to recall having traversed I and have played action a there. The last-infosets of player i at h are the information sets traversed by h and not remembered by any following information set of the player up to h.[4.8] This set quantifies the knowledge lost by the player at a node due to imperfect recall.

Definition 4.12. An infoset J *remembers* another infoset I if there exists an action $a \in \mathcal{A}(I)$ such that, for every $h \in J$, we have $h'a \preceq h$ for some $h' \in I$.

Definition 4.13. The set of *last-infosets* at node h for player i is the set of infosets $I \in \mathcal{I}_i$ such that $I \preceq h$ and there is no other infoset $J \in \mathcal{I}_i$ such that $J \preceq h$ and J remembers I.

We will use $\mathrm{LI}_i(h)$ to denote the set of last-infosets at h for player i. Note that if $h \in \mathcal{H}_i$ then $I_h \in \mathrm{LI}_i(h)$.

Now define the *information complexity* k of a two-player game Γ as follows.

$$k = \max_{\substack{i \in \{\blacktriangle, \blacktriangledown\},\\ P \in \mathcal{P}_i}} \left| \bigcup_{h \in P} \mathrm{LI}_i(h) \right|$$

Intuitively, k is a representation of *how much information can be worst-case forgotten* by player i. In the team game interpretation, k is a representation of how *asymmetric* the information is among team members. Note that $k = 1$ if and only if both players have perfect recall.

The information complexity characterizes both the number of beliefs in a public state P, and the number of prescriptions that are available at such beliefs. In fact, the actions played at information sets in $\bigcup_{h \in P} \mathrm{LI}_i(h)$ determine which nodes in P are reached (that is, a belief $B \subseteq P$).

As an example, consider the game from Figure 4.3 and the public state $P = \{g, h\}$. We have that

[4.8] From the perspective of an adversarial team game, the last-infosets at a node for team $t \in \{\triangle, \triangledown\}$ are the most recent infosets of each player in t, minus the infosets of players that are implied by other players' infosets.

Algorithm 4.4 (MakeBeliefGame): Belief game construction

```
procedure $\textsc{MakeNode}_\blacktriangle(h, B_\blacktriangle, B_\blacktriangledown, \tilde\sigma_\blacktriangle, \tilde\sigma_\blacktriangledown)$
    create node $\tilde h \in \tilde{\mathcal{H}}_\blacktriangle$
    add $\tilde h$ to infoset labeled $(\tilde\sigma_\blacktriangle, B_\blacktriangle)$
    for each prescription $\boldsymbol{a}_\blacktriangle \in \mathcal{A}_\blacktriangle(B_\blacktriangle)$ do
        $\tilde h \boldsymbol{a}_\blacktriangle \leftarrow \textsc{MakeNode}_\blacktriangledown(h, B_\blacktriangle, B_\blacktriangledown, \tilde\sigma_\blacktriangle, \tilde\sigma_\blacktriangledown, a_\blacktriangle)$
    return $\tilde h$
procedure $\textsc{MakeNode}_\blacktriangledown(h, B_\blacktriangle, B_\blacktriangledown, \tilde\sigma_\blacktriangle, \tilde\sigma_\blacktriangledown, \boldsymbol{a}_\blacktriangle)$
    create node $\tilde h \boldsymbol{a}_\blacktriangle \in \tilde{\mathcal{H}}_\blacktriangledown$
    add $\tilde h \boldsymbol{a}_\blacktriangle$ to infoset labeled $(\tilde\sigma_\blacktriangledown, B_\blacktriangledown)$
    for each prescription $\boldsymbol{a}_\blacktriangledown \in \mathcal{A}_\blacktriangledown(B_\blacktriangledown)$ do
        $\tilde h \boldsymbol{a}_\blacktriangle \boldsymbol{a}_\blacktriangledown \leftarrow \textsc{MakeNode}_\mathsf{C}(h, B_\blacktriangle, B_\blacktriangledown, \tilde\sigma_\blacktriangle, \tilde\sigma_\blacktriangledown, \boldsymbol{a}_\blacktriangle, \boldsymbol{a}_\blacktriangledown)$
    return $\tilde h \boldsymbol{a}_\blacktriangle$
procedure $\textsc{MakeNode}_\mathsf{C}(h, B_\blacktriangle, B_\blacktriangledown, \tilde\sigma_\blacktriangle, \tilde\sigma_\blacktriangledown, \boldsymbol{a}_\blacktriangle, \boldsymbol{a}_\blacktriangledown)$
    if $h$ is terminal node then
        create new terminal node $\tilde h \boldsymbol{a}_\blacktriangle \boldsymbol{a}_\blacktriangledown \in \tilde{\mathcal{Z}}$
        $u_i(\tilde h \boldsymbol{a}_\blacktriangle \boldsymbol{a}_\blacktriangledown) \leftarrow u_i(h)$ for each player $i$
        $\boldsymbol{p}(\tilde h \boldsymbol{a}_\blacktriangle \boldsymbol{a}_\blacktriangledown) \leftarrow \boldsymbol{p}(h)$
        return $\tilde h \boldsymbol{a}_\blacktriangle \boldsymbol{a}_\blacktriangledown$
    create new chance node $\tilde h \boldsymbol{a}_\blacktriangle \boldsymbol{a}_\blacktriangledown \in \mathcal{H}_\mathsf{C}$
    if $h$ is a chance node then $S \leftarrow \{ha : a \in \mathcal{A}(h)\}$
    else $S \leftarrow \{h\boldsymbol{a}_i(I_h)\}$ where $h \in \mathcal{H}_i$
    for each node $ha \in S$ do
        $B_i' \leftarrow \textsc{SplitBelief}_i(B_i \boldsymbol{a}_i, ha)$ for each player $i$
        $\tilde h \boldsymbol{a}_\blacktriangle \boldsymbol{a}_\blacktriangledown a \leftarrow \textsc{MakeNode}_\blacktriangle(ha, B_\blacktriangle{}', B_\blacktriangledown{}', \tilde\sigma_\blacktriangle + (\tilde\sigma_\blacktriangle, a_\blacktriangle), \tilde\sigma_\blacktriangledown + (\tilde\sigma_\blacktriangledown, a_\blacktriangledown))$
    return $\tilde h \boldsymbol{a}_\blacktriangle \boldsymbol{a}_\blacktriangledown$
```

the strategy played at

$$\bigcup_{h\in\{g,h\}} \mathrm{LI}_i(h) = \{I_a, I_c.I_g\} \cup \{I_a, I_d, I_h\} = \{I_a, I_c, I_d, I_g\}$$

is enough to characterize a belief $B \in P$ and a prescription at that belief. In fact, the action at I_a decides whether c and d are reached, the actions at I_c (respectively I_d) decide whether g (respectively h) is reached, and the action at $I_g = I_h$ is the prescription.

It is instructive to understand how k behaves in a simple game. Suppose that Γ is a team game such that there are n players on each team, each player is assigned one of t "private types" (in poker, these are the private hands) and all other information in the game is common knowledge. Then at each public state $P \in \mathcal{P}_i$, there are at most t last-infosets per player, so $k = nt$.

4.3.1 Belief Game Construction

We now introduce an algorithm that explicitly constructs a belief game given any two-player game. We will use $\tilde{\Gamma}$ to denote the belief game and distinguish components of the original game Γ from components of the belief game by writing tildes: for example, a generic history is $\tilde{h} \in \tilde{\mathcal{H}}$, a generic information set is $\tilde{I}_i \in \tilde{\mathcal{I}}_i$, and so on.

Here, for cleanliness, we will describe the evolution of the belief game as a game of *simultaneous moves*. Algorithm 4.4 (MakeBeliefGame) describes the procedure that constructs an extensive-form game (without simultaneous moves) that is equivalent to it.[4.9] In particular, $\textsc{MakeNode}_{\blacktriangle}(\varnothing, \{\varnothing\}, \{\varnothing\}, \varnothing, \varnothing)$ constructs the whole belief game.

A node $\tilde{h} \in \tilde{\mathcal{H}}$ in the belief game is identified by a tuple $(h, B_{\blacktriangle}, B_{\blacktriangledown})$ such that $h \in B_{\blacktriangle} \cap B_{\blacktriangledown}$, where $h \in \mathcal{H}$ is the corresponding node in the original game describing the underlying state of the game, $B_{\blacktriangle}, B_{\blacktriangledown}$ are the current beliefs of $\blacktriangle$ and $\blacktriangledown$ respectively. At $\tilde{h}$, each player $i \in \{\blacktriangle, \blacktriangledown\}$ has a (possibly empty) collection of infosets, $\mathcal{I}[B_i]$, at which it needs to prescribe an action. The two players simultaneously submit actions $\boldsymbol{a}_i \in \mathcal{A}_i(B_i)$. The next belief game node is $(ha, B'_{\blacktriangle}, B'_{\blacktriangledown})$, where:

(i) The action a is the one taken by the player at h: if h is chance node, then a is sampled from chance's action distribution at h; otherwise, $a = \boldsymbol{a}_i(I_h)$.

(ii) the beliefs evolve as follows. For each player i, the set of candidate next histories in the original game compatible with i's current belief B_i and its prescription $\boldsymbol{a}_i$ is given by

$$B_i\boldsymbol{a}_i := \underbrace{\{ha : h \in B_{\blacktriangle} \cap \mathcal{H}_i, a = \boldsymbol{a}(I_h)\}}_{\substack{\text{when player } i \text{ acts,} \\ \text{it must be according to the prescription}}} \cup \underbrace{\{ha : h \in B_{\blacktriangle} \setminus \mathcal{H}_i, a \in \mathcal{A}(h)\}}_{\substack{\text{when player } i \text{ does not act,} \\ i \text{ does not know what action is taken}}},$$

Next, player i observes the information revealed by the next history ha, thus arriving at belief

$$\tilde{B}'_i := \textsc{SplitBelief}_i(B_i\boldsymbol{a}_i, ha).$$

We remark some characteristics of $\tilde{\Gamma} :=$ MakeBeliefGame(Γ).

- Multiple different tree nodes $\tilde{h}$ can correspond to the same $(h, B_{\blacktriangle}, B_{\blacktriangledown})$ tuple. In particular, for each terminal node $z \in \mathcal{Z}$ there is only one state $(z, \{z\}, \{z\})$.
- Information sets in $\tilde{\Gamma}$ are associated to sequences of beliefs and prescriptions. In particular, such infosets can be described by tuples of the form $(B_i^1 = \{\varnothing\}, \boldsymbol{a}_i^1, B_i^2, \boldsymbol{a}_i^2, \dots, B_i^L)$, where $\boldsymbol{a}_i^\ell \in \mathcal{A}(B_i^\ell)$ and $B_i^{\ell+1} = \textsc{SplitBelief}_i(B_i^\ell \boldsymbol{a}_i^\ell, h)$ for some $h \in B_i^\ell \boldsymbol{a}_i^\ell$.
- By construction of MakeBeliefGame we have that $\tilde{\Gamma}$ is a perfect-recall game. In fact, nodes with different sequences are associated to different information sets thanks to including sequences in each information set's label;

[4.9] We implement simultaneous actions by representing each step in the game as a sequence of one node per player $\blacktriangle$, $\blacktriangledown$, C where everyone acts; the effects of the actions taken are applied at the end.

- When h is terminal, the belief game does not stop until both players have observed the trivial belief $\{h\}$ at h and then submitted their empty prescriptions at that belief. This is for notational convenience: it ensures that terminal sequences for a player i will always end with singleton beliefs, which will make the later analysis cleaner.
- Modulo trivial reformulations (namely, the insertion of nodes with a single child), if Γ is perfect recall then $\tilde{\Gamma}$ is identical to Γ.

Given a pure strategy $\tilde{\boldsymbol{x}}_i \in \tilde{X}_i$, we say that $\tilde{\boldsymbol{x}}_i$ plays to a belief B_i of player i if $\tilde{\boldsymbol{x}}_i$ plays to some node corresponding to (h, B_i, B_{-i}).

Theorem 4.14. *Let Γ be any two-player imperfect-recall extensive-form game, and $\tilde{\Gamma}$ be the belief game constructed by* MakeBeliefGame. *Γ and $\tilde{\Gamma}$ are strategically equivalent.*

Proof. The requirements for strategic equivalence as per Definition 4.3 that we will show are: i) it exists a function ρ mapping pure strategies in the two games ii) ρ is a bijective function iii) ρ is value-preserving, that is, the mapped strategies have the same expected utilities.

We first construct the strategy maps $\rho_i : \Pi_i \to \tilde{\Pi}_i$. A pure strategy $\boldsymbol{x}_i \in \Pi_i$ in G assigns one action to each information set.[4.10] From such a strategy we construct a strategy $\tilde{\boldsymbol{x}}_i = \rho_i(\boldsymbol{x}_i)$ which plays prescriptions consistent with $\boldsymbol{\pi}_i$. At belief B_i, $\tilde{\boldsymbol{x}}_i$ plays the prescription $\boldsymbol{a}_i$ given by $\boldsymbol{a}_i(I) = \boldsymbol{x}_i(I)$ for each $I \in \mathcal{I}_{B_i}$ reached by $\tilde{\boldsymbol{x}}$.

We now show that ρ_i is injective. This will follow from the following lemma.

Lemma 4.15. *Let $\tilde{\boldsymbol{x}}_i = \rho_i(\boldsymbol{x}_i) \in \tilde{\Pi}_i$. Then for every $z \in \mathcal{Z}$, $\tilde{\boldsymbol{x}}_i$ plays to belief $\{z\}$ if and only if $\boldsymbol{x}_i$ plays to z.*

> *Proof.* First suppose $\tilde{\boldsymbol{x}}_i = \rho_i(\boldsymbol{x}_i) \in \tilde{\Pi}_i$ plays to $\{z\}$. Thus $\tilde{\boldsymbol{x}}_i$ plays some sequence of beliefs and prescriptions $(B_i^1 = \{\varnothing\}, \boldsymbol{a}_i^1, \dots, B_i^2, \boldsymbol{a}_i^2, \dots, B_i^L = \{z\})$. But then, by construction, every ancestor $h \prec z$ is included in one of the B_i^ℓs, and for $ha \preceq z$ to appear in $B_i^\ell \boldsymbol{a}_i^\ell$, if $h \in \mathcal{H}_i$ it must be the case that $\boldsymbol{x}_i$ plays a. Thus $\boldsymbol{x}_i$ plays to z.
>
> Conversely suppose $\boldsymbol{x}_i$ plays to z. Then, construct a sequence of beliefs and prescriptions as follows. Let $\boldsymbol{a}_i^\ell$ be the prescrption played by $\tilde{\boldsymbol{x}}_i$ at belief B_i^ℓ, and $B_i^{\ell+1} = \textsc{SplitBelief}_i(B_i^\ell \boldsymbol{a}_i^\ell, h)$ where $h \in B_i^\ell \boldsymbol{a}_i^\ell$ and $h \preceq z$. (The fact that $\boldsymbol{x}_i$ plays to z ensures that such h must exist). Then, by induction, $\tilde{\boldsymbol{x}}_i$ plays to this sequence, and the sequence must eventually terminate at $\{z\}$ because it always contains at least one predecessor of z. Thus $\tilde{\boldsymbol{x}}_i$ plays to $\{z\}$. □

Since different pure strategies (by definition) play to different sets of terminal nodes, this immediately shows that ρ is injective. We now show that ρ_i is a surjection (and hence a bijection), that is, any $\tilde{\boldsymbol{x}}_i \in \tilde{\Pi}_i$ is the image of some $\boldsymbol{x}_i \in \Pi_i$. We remark that a pure strategy $\tilde{\boldsymbol{x}}_i \in \tilde{\Pi}_i$ assigns one prescription to every reached infoset of player i in $\tilde{G}$.

We will require the following lemma. Informally, it states that no player can play to two nodes

[4.10] At infosets not reached by $\boldsymbol{x}_i$, actions can be selected arbitrarily.

with intersecting beliefs.

Lemma 4.16. *Let $\tilde{\boldsymbol{x}}_i \in \tilde{\Pi}_i$ be any pure strategy. Let $\tilde{I}, \tilde{I}' \in \tilde{\mathcal{I}}_i$ be two distinct infosets of player i that are simultaneously reached by $\tilde{\boldsymbol{x}}_i$. Let B_i and B_i' be the beliefs for player i at $\tilde{I}$ and $\tilde{I}'$ respectively. Then B_i and B_i' are not connected and do not intersect. That is, there do not exist nodes $h \in B_i, h' \in B_i'$ with $(h, h') \in \mathcal{E}_i$ or $h = h'$.*

> *Proof.* Consider the tree-form decision problem $\tilde{\mathcal{T}}$ of player i in $\tilde{G}$. Since $\tilde{G}$ is perfect-recall, $\tilde{\mathcal{T}}$ is indeed a valid representation of player i's strategy set in $\tilde{G}$. Let s be the lowest common ancestor of $\tilde{I}$ and $\tilde{I}'$ in $\tilde{\mathcal{T}}$.
>
> Since $\tilde{\boldsymbol{x}}_i$ plays to both $\tilde{I}$ and $\tilde{I}'$, the node s must be an observation point, as a pure strategy plays a single action at each decision point. At observation points, the next observation made by player i is the next belief. Let C_i and C_i' be the different observed beliefs at node s that lead to $\tilde{I}$ and $\tilde{I}'$, respectively. Then, by construction of SPLITBELIEF$_i$, C_i and C_i' are disconnected and disjoint. Since every node in B_i is a descendant of some node in C_i (and the same for C_i'), it follows that B_i and B_i' are also disconnected and disjoint. □

Thus, in particular, for any infoset $I \in \mathcal{I}_i$ in the original game, $\tilde{\boldsymbol{x}}_i$ can only play to one infoset $\tilde{I} \in \tilde{\mathcal{I}}_i$ whose belief B_i overlaps I. At B_i, the prescription chosen by $\tilde{\boldsymbol{x}}_i$ includes an action $\boldsymbol{a}(I)$ at I (by construction of prescriptions at a node). Thus, consider the strategy $\boldsymbol{x}_i$ defined such that $\boldsymbol{x}_i(I) = \boldsymbol{a}(I)$ for every infoset I such that $\boldsymbol{x}_i$ plays to a belief B_i overlapping I.

Lemma 4.17. *$\boldsymbol{x}_i$ is a well-defined strategy. That is, if $\boldsymbol{x}_i$ plays to a node $h \in \mathcal{H}_i$, then $\tilde{\boldsymbol{x}}_i$ plays to a belief $B_i \ni h$, and hence, if $h \in I \in \mathcal{I}_i$ then $\boldsymbol{x}_i(I)$ is defined.*

> *Proof.* By induction on the length of the history h. For $h = \varnothing$ this is trivial. Now let $h = h'a'$ be a non-root node, and suppose $\boldsymbol{x}_i$ plays to h. Then by inductive hypothesis, $\tilde{\boldsymbol{x}}_i$ plays to a belief $B_i \ni h'$. Let $\boldsymbol{a}$ be the prescription played by $\tilde{\boldsymbol{x}}_i$ at B_i.
>
> - If $h' \in I \in \mathcal{I}_i$, then by construction of $\boldsymbol{x}_i$, it must be the case that $\tilde{\boldsymbol{x}}_i$'s prescription $\boldsymbol{a}$ at B_i satisfies $\boldsymbol{a}(I) = a'$, so $B_i\boldsymbol{a} \ni h$.
> - If $h' \notin \mathcal{H}_i$, then for *any* prescription $\boldsymbol{a}$ at B_i we have that $h \in B_i\boldsymbol{a}$.
>
> In either case, we have $B_i\boldsymbol{a} \ni h$. Thus, $\tilde{\boldsymbol{x}}_i$ must also play to the belief SPLITBELIEF$_i(B_i\boldsymbol{a}, h) \ni h$. □

Further, from the definition of ρ_i it follows immediately that $\rho_i(\boldsymbol{x}_i) = \tilde{\boldsymbol{x}}_i$. It only remains to show that ρ_i is value-preserving. But this is easy: $\rho_i(\boldsymbol{x}_i)$ prescribes the same actions as $\boldsymbol{x}_i$. Thus, for any pure strategy profile $\boldsymbol{x} \in \Pi$, following profile $\boldsymbol{x}$ through G will yield exactly the same trajectory as following profile $\rho(\boldsymbol{x})$ through $\tilde{G}$. Thus, their expected utilities will also coincide, and the proof is complete. □

4.3.2 Worst-Case Dimension of the Belief Game

The per-iteration time complexity of CFR depends linearly on the size of the game on which the algorithm is applied. Thus, it is critical for complexity analysis to bound the size of the belief

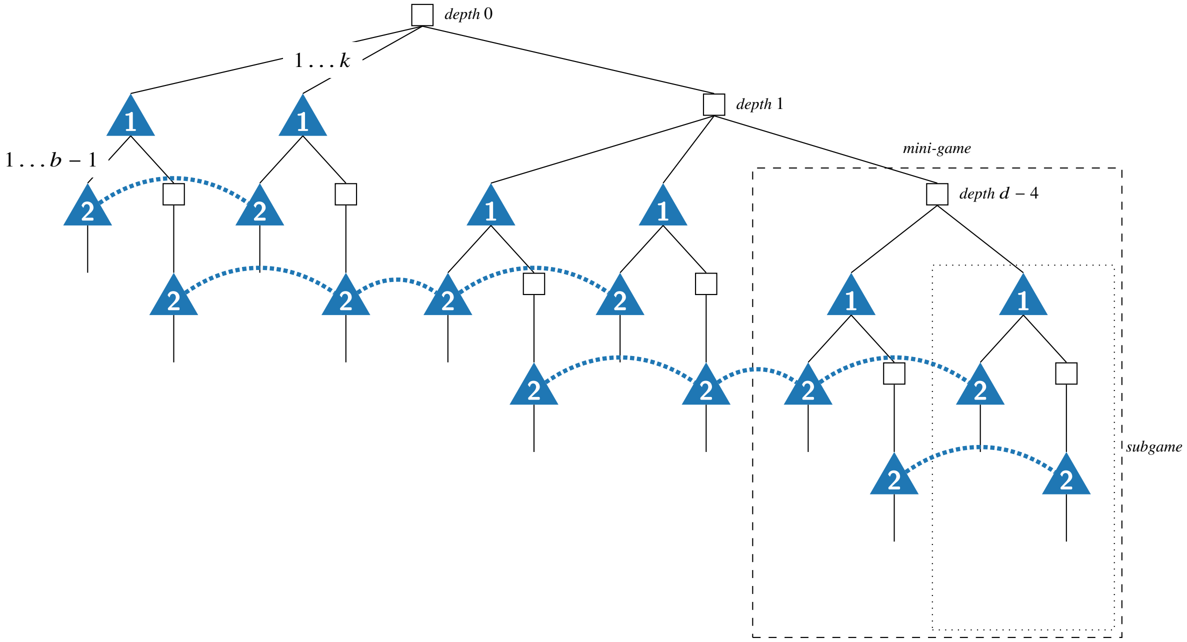


Figure 4.5: *An example of imperfect-recall game derived from a team game whose rationale is described in the proof of Theorem 4.18. We omit terminal values because they are not relevant. The boxes indicated by* mini-game *and* subgame *correspond to the terms used in the description.*

game produced by MakeBeliefGame.

Lower Bound. We first present a lower bound of the worst-case size of the belief game, *i.e.* a worst-case instance of game whose belief game has a large number of histories.

Theorem 4.18. *There exists a game G with depth d, information complexity k, and maximum branching factor at a node b such that the number of nodes in the belief game $\tilde{G}$ is $|\tilde{\mathcal{H}}| \geq b^{2k(d-4)}$.*

Proof. We construct a parametric game for depth $d \geq 4$, information complexity k, and branching factor $b \geq k + 1$.

Consider a game that consists of $d - 3$ repetitions of the following *mini-game*. There is a *"root"* nature node with k nodes of ▲, k nodes of ▼ and the root of the next repetition of the mini-game as children. Each of ▲'s and ▼'s nodes belong to different information sets. Each of those is the root of *subgames* with identical structures. Their children are $b - 1$ player nodes followed by a single terminal node, and there is a chance node followed by a player node with a single terminal node. These player nodes belong to the same player as the root of the subtree, namely ▲ for the first k children and ▼ for the second k children of the *"root"* of the mini-game. All nodes of ▲ and ▼ belong to the same level of the game apart from the $2k$ nodes that belong to the same information set.

A representation of such a game for $k = 2, b = 2, d = 6$ is given in Figure 4.5, where nodes of ▼ have been omitted to improve clarity. Those have the same structure as the nodes of ▲. The main point of this game is that at any level $l = 1, \ldots, d - 3$, both ▲ and ▼ have a public state containing the k infosets corresponding to the nodes after C's *"root"* of the *mini-game* at depth $l - 1$. At each of those public states, both ▲ and ▼ have a single belief containing k information sets and the chance node that leads to the next *mini-game*. Therefore, there are b^k prescriptions at this belief, and each prescription played leads, among the others, to the belief containing k information sets of the next *mini-game*.

Multiplying this factor for each step of the level gives $|\mathcal{H}'| \geq \left(b^{2k}\right)^{d-4}$. □

Upper Bound. We now present an upper bound on the number of histories of the belief game.

Theorem 4.19. *Let Γ be a game with depth d, information complexity k and maximum branching factor at a node b. The number of nodes in the belief game $\tilde{\Gamma}$ is $|\tilde{\mathcal{H}}| \leq b^{2kd+d}$.*

Proof. Consider the algorithm MakeBeliefGame. Grouping levels of the belief game $\tilde{G}$ three by three, we have that ▲, ▼, and Ceach play at a different level in each group, and we have d groups. Moreover, no more than b actions for the chance player and b^k prescriptions are available to each player. We, therefore, have that the number of nodes in-game tree $\tilde{G}$ is $|\tilde{\mathcal{H}}| \leq b^{d(2k+1)}$. □

Discussion. The bounds presented in this section highlight the main computational limitation of MakeBeliefGame, the explicit dependence on depth introduced by explicitly using sequences to distinguish information sets in the belief game.

We remark that we can replace k here with the maximum number of infosets (not the last-infosets) in any public state. We opted not to introduce two different notions of information complexity to have bounds comparable with the TB-DAG ones in Section 4.5.1. We will explore the effects of introducing the different definitions of k in Section 4.7.3.

4.3.3 Regret Minimization on Team Games

This section shows how to find a mixed Nash equilibrium in a generic two-player zero-sum game with imperfect recall Γ by applying CFR on the belief game $\tilde{\Gamma}$ obtained by running MakeBeliefGame on Γ.

Let $\tilde{\mathcal{X}}$ and $\tilde{\mathcal{Y}}$ be the realization-form *mixed* strategy spaces for ▲ and ▼ in $\tilde{\Gamma}$ derived from the sequence-form representation as in Section 2.2.2. Specifically, vectors $\tilde{\boldsymbol{x}} \in \tilde{\mathcal{X}}$ are indexed by terminal sequences for ▲ in $\tilde{\Gamma}$ (similarly for ▼). Such a sequence σ can be identified by a list of beliefs and prescriptions, ending in a singleton belief $\{z\}$ for terminal node $z \in \mathcal{Z}$. For any terminal node z, let $\Sigma^z_{▲}$ be the set of terminal sequences for ▲ that end at belief $\{z\}$. Then

Algorithm 4.6 (DAG-Generic): Generic construction of a regret minimizer $\mathcal{R}$ on $\mathcal{Q}$ from a regret minimizer $\hat{\mathcal{R}}$ on its tree form $\hat{\mathcal{Q}}$.

```
procedure NextStrategy
    $\hat{\boldsymbol{x}}^t \leftarrow \hat{\mathcal{R}}.\textsc{NextStrategy}()$
    return $\mathbf{D}\hat{\boldsymbol{x}}^t$
procedure ObserveUtility($\boldsymbol{u}^t$)
    $\hat{\mathcal{R}}.\textsc{ObserveUtility}(\mathbf{D}^\top \boldsymbol{u}^t)$
```

computing a Nash equilibrium in $\tilde{\Gamma}$ (and hence a mixed Nash in Γ) can be done by solving the max-min problem

$$\max_{\tilde{\boldsymbol{x}}\in\tilde{\mathcal{X}}} \min_{\tilde{\boldsymbol{y}}\in\tilde{\mathcal{Y}}} \sum_{z\in\mathcal{Z}} u(z) \sum_{\tilde{\sigma}_\blacktriangle\in\Sigma^z_\blacktriangle} \tilde{\boldsymbol{x}}(\tilde{\sigma}_\blacktriangle) \sum_{\tilde{\sigma}_\blacktriangledown\in\Sigma^z_\blacktriangledown} \tilde{\boldsymbol{y}}(\tilde{\sigma}_\blacktriangledown). \tag{4.1}$$

This is equivalent to the max-min problem for the coordinator game by setting $\boldsymbol{x}(z) := \sum_{\tilde{\sigma}_\blacktriangle\in\Sigma^z_\blacktriangle} \tilde{\boldsymbol{x}}(\tilde{\sigma}_\blacktriangle)$ (and similar for $\boldsymbol{y}$). That is, from an optimization perspective, what has happened is that we have constructed sets $\tilde{\mathcal{X}}$ and $\tilde{\mathcal{Y}}$ that are described by linear constraints, just like the sequence form, and *project* onto $\mathcal{X}$ and $\mathcal{Y}$ respectively, allowing the reformulation and equivalence of problems.

We now analyze the time complexity and regret of running CFR on $\tilde{\Gamma}$. Fix a player, say, $\blacktriangle$. (The same analysis will apply to $\blacktriangledown$.) First, recall from Section 2.2.2 that, in a decision problem, a set P of $\blacktriangle$-decision points is called *playable* if there exists a pure strategy of $\blacktriangle$ that plays to all the decision points in P. But the size of any playable set P of $\blacktriangle$ is at most $|\mathcal{H}|$. Further, the branching factor of $\tilde{\Gamma}$ is at most b^k, where b is the branching factor of Γ and k is the information complexity (see Section 4.3.2). Thus, applying multiplicative weights (MWU) as the local regret minimizer at each decision point and using Proposition 2.6, we have:

Theorem 4.20. *After T iterations of CFR on $\tilde{\Gamma}$ with MWU as the local regret minimizer, the average strategy profile $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ is an $O(\epsilon)$-Nash equilibrium of Γ, where*

$$\epsilon = |\mathcal{H}|\sqrt{\frac{k\log b}{T}}.$$

The per-iteration complexity is linear in the size of $\tilde{\Gamma}$.

While the regret above is polynomial in $\mathcal{H}$, the per-iteration complexity depends on the size of $\tilde{\Gamma}$, which is worst-case exponentially larger than Γ, as shown in Section 4.3.2.

4.4 DAG Decision Problems

In this section, we will develop a general theory of DAG-form decision problems, and regret minimization on them, analogous to the tree-form theory in Section 2.2.2. Although our main interest in DAG-form decision-making is its application to two-player imperfect-recall games

Algorithm 4.7 (DAG-CFR): Counterfactual regret minimization on DAG-form decision problems Q. For each decision point j, $\mathcal{R}_j$ is a regret minimizer on $\Delta(\mathcal{A}(j)))$.

```
procedure NextStrategy
    $\boldsymbol{x}^t(\varnothing) \leftarrow 1$
    for each decision point $j$, in top-down order do
        $\boldsymbol{r}^t_j \leftarrow \mathcal{R}_j$.NextStrategy()
        $\boldsymbol{x}^t(j*) \leftarrow \sum_{p \in P_j} \boldsymbol{x}^t(p) \boldsymbol{r}^t_j$
    return $\boldsymbol{x}^t$
procedure ObserveUtility($\boldsymbol{u}^t$)
    $\boldsymbol{v}^t \leftarrow \boldsymbol{u}^t$
    for each decision point $j$, in bottom-up order do
        $\mathcal{R}_j$.ObserveUtility($\boldsymbol{v}^t(j*)$)
        for $p \in P_j$ do $\boldsymbol{v}^t(p) \leftarrow \boldsymbol{v}^t(p) + \left\langle \boldsymbol{r}^t_j, \boldsymbol{v}^t(j*) \right\rangle$
    $t \leftarrow t + 1$
```

(which we will develop in Section 4.5), the observations made in this section also have general applicability beyond this setting, as we will see repeatedly throughout this thesis.

As one may expect, DAG-form decision problems are identical to tree-form decision problems except that the graph of nodes is allowed to be a DAG, albeit with some restrictions.

Definition 4.21. A *DAG-form decision problem* is a DAG with a unique source (root node) $\varnothing$, wherein each node is either a decision point ($j \in \mathcal{J}$) or an observation point ($s \in \mathcal{S}$)[4.11], with the following properties:

1. Observation points other than the root have exactly one incoming edge.
2. For any two paths p_1 and p_2 from the root that end at the same node, the last node in common between p_1 and p_2 is a decision point.

As with tree-form decision problems, we will also assume (WLOG) that decision and observation points alternate along every path, and that both the root node and all terminal nodes are observation points. A *pure strategy* is once again an assignment of one action to each decision point. The *DAG form* of a pure strategy is the vector $\boldsymbol{x} \in \{0,1\}^{\mathcal{S}}$, where $\boldsymbol{x}(s) = 1$ if there is *some* $\varnothing \to s$ path along which the player plays all actions. A mixed strategy $\boldsymbol{x} \in Q$ is a convex combination of pure strategies. Since decision points can now have multiple parents, we will use P_j to denote the set of parents of a decision point j.

Like tree-form decision problems, the mixed strategy set in a DAG-form decision problem has a

[4.11] For most of this thesis, observation points are denoted Σ; however, here we will need to distinguish between observation points $s \in \mathcal{S}$ and sequences in the original game. We hence choose different notation.

convenient representation using linear constraints, namely:

$$\begin{cases} \boldsymbol{x}(\varnothing) = 1 \\ \sum_{p \in P_j} \boldsymbol{x}(p) = \sum_{a \in \mathcal{A}(j)} \boldsymbol{x}(ja) \qquad \text{for all} \quad j \in \mathcal{J}. \end{cases} \tag{4.2}$$

DAG-form decision problems and tree-form decision problems are closely related. Of course, all tree-form decision problems are DAG-form decision problems. Conversely, any DAG-form decision problem can be thought of as a "compressed" representation of the tree-form decision problem created by separating out all the different paths through the DAG. While this tree will generally be exponentially larger than the DAG, we will find it useful to compare the DAG and tree representations.

We now formulate a general theory of regret minimization for DAG-form decision problems. We will use hats ($\hat{Q}, \hat{\mathcal{J}}, \hat{\mathcal{S}}, \hat{\boldsymbol{x}}$) to denote components of the tree form of a generic DAG-form regret minimizer. For each tree-form observation point $s \in \hat{\mathcal{S}}$ let $\delta(s) \in \mathcal{S}$ be the corresponding observation point in $\mathcal{S}$. Note that, by construction, δ is surjective but not injective unless the DAG happens to be a tree.

We now show how tree-form strategies and utilities correspond to DAG-form strategies and utilities. Concretely, we define a matrix $\mathbf{D} \in \mathbb{R}^{\mathcal{S} \times \hat{\mathcal{S}}}$ by $\mathbf{D}\hat{\boldsymbol{x}}(s) = \sum_{\hat{s}:\delta(\hat{s})=s} \hat{\boldsymbol{x}}(\hat{s})$ for all $\hat{\boldsymbol{x}} \in \mathbb{R}^{\hat{\mathcal{S}}}$. This is the matrix of the linear map that transforms tree-form strategies to their corresponding DAG-form strategies. That is, $\mathbf{D} : \hat{\mathcal{X}} \to \mathcal{X}$ is a bijection.

Dually, for DAG-form utility vectors $\boldsymbol{u} \in \mathbb{R}^{\mathcal{S}}$, the vector $\mathbf{D}^\top \boldsymbol{u} \in \mathbb{R}^{\hat{\mathcal{S}}}$ is a utility vector on the tree form, with the property that $\langle \mathbf{D}^\top \boldsymbol{u}, \hat{\boldsymbol{x}} \rangle = \langle \boldsymbol{u}, \mathbf{D}\hat{\boldsymbol{x}} \rangle$ by definition of the inner product. That is, the DAG-form strategy $\mathbf{D}\hat{\boldsymbol{x}}$ achieves the same utility against DAG-form utility vector $\boldsymbol{u}$ as the tree-form strategy $\hat{\boldsymbol{x}}$ achieves against the utility $\mathbf{D}^\top \hat{\boldsymbol{u}}$.

The relationship between trees and DAGs allows us to use *any* regret minimizer on $\hat{Q}$ to construct a regret minimizer with the same guarantee on Q. We do this in Algorithm 4.6 (DAG-Generic).

Proposition 4.22. *Let $\mathcal{R}$ and $\hat{\mathcal{R}}$ be as in* DAG-Generic. *Then the regret of $\mathcal{R}$ with utility sequence $\boldsymbol{u}^1, \dots, \boldsymbol{u}^T$ is equal to the regret of $\hat{\mathcal{R}}$ with utility sequence $\mathbf{D}^\top \boldsymbol{u}^1, \dots, \mathbf{D}^\top \boldsymbol{u}^T$.*

Proof. Using the fact that $\mathbf{D}$ is a bijection, we have

$$\begin{aligned} R_Q^T &= \max_{\boldsymbol{x} \in Q} \sum_{t=1}^T \left\langle \boldsymbol{u}^t, \boldsymbol{x} - \boldsymbol{x}^t \right\rangle \\ &= \max_{\hat{\boldsymbol{x}} \in \hat{Q}} \sum_{t=1}^T \left\langle \boldsymbol{u}^t, \mathbf{D}\boldsymbol{x} - \mathbf{D}\hat{\boldsymbol{x}}^t \right\rangle \\ &= \max_{\hat{\boldsymbol{x}} \in \hat{Q}} \sum_{t=1}^T \left\langle \mathbf{D}^\top \boldsymbol{u}^t, \boldsymbol{x} - \boldsymbol{x}^t \right\rangle = R_Q^T. \end{aligned}$$

□

Applying the transformation DAG-Generic with CFR as the tree-form regret minimizer $\hat{\mathcal{R}}$, we arrive at a DAG form of CFR, which can be simulated efficiently: Algorithm 4.7 (DAG-CFR). One can think of DAG-CFR as a *more efficient implementation* of CFR when the decision tree happens to have a DAG structure. Of course, the regret bound $O(|\mathcal{S}|\sqrt{T})$ is only a worst-case bound; in special cases (such as Theorem 4.20), CFR does much better than its worst case, and therefore so will DAG-CFR.

Call a utility vector $\hat{\boldsymbol{u}}$ *consistent* if it is in the image of $\mathbf{D}^\top$. That is (expanding the definition of $\mathbf{D}^\top$), $\hat{\boldsymbol{u}} \in \mathbb{R}^{\hat{O}}$ is consistent if $\hat{\boldsymbol{u}}(\hat{s}) = \hat{\boldsymbol{u}}(\hat{s}')$ if $\delta(\hat{s}) = \delta(\hat{s}')$. In essence, a DAG-form regret minimizer is able to "simulate" a tree-form regret minimizer so long as the tree-form regret minimizer's utilities are always consistent. We now formalize this idea.

Theorem 4.23 (DAG regret minimization via CFR)**.** DAG-CFR *produces the same iterates as* DAG-Generic *with* CFR *as* $\hat{\mathcal{R}}$. *Therefore, in particular, the regret of* DAG-CFR *with utility sequence* $\boldsymbol{u}^1, \dots, \boldsymbol{u}^T$ *is the same as that of* CFR *on the tree form with utility sequence* $\hat{\boldsymbol{u}}^1 := \mathbf{D}^\top \boldsymbol{u}^1, \dots, \hat{\boldsymbol{u}}^t := \mathbf{D}^\top \boldsymbol{u}^T$. *Moreover, the per-iteration runtime of* DAG-CFR *is linear in the number of edges in the DAG. In particular, taking any reasonably efficient regret minimizer over simplices, the regret of* DAG-CFR *after T iterations is at most $O(|\mathcal{S}|\sqrt{T})$.*

Proof. Let $s \in \mathcal{D}$ be any decision point of Q, and let $\hat{s} \in \hat{\mathcal{D}}$ be any decision point in $\hat{Q}$ with $\delta(\hat{s}) = s$. It is enough to show that the sequence of utility vectors observed by $\mathcal{R}_{\hat{s}}$ when running DAG-Generic with CFR as $\hat{\mathcal{R}}$ is the same as the sequence of utility vectors observed by $\mathcal{R}_s$ in DAG-CFR. We show this by induction on the decision points $\hat{s}$, leaves first.

First, if $\hat{s}$ has no decision point descendants, then the claim is trivial because, by construction of $\mathbf{D}^\top$, we have $\hat{\boldsymbol{u}}^t(\hat{s}a) = \boldsymbol{u}^t(sa)$ for every $a \in A_s$. Now let $\hat{s} \in \hat{\mathcal{D}}$ be any internal node and $s = \delta(\hat{s})$. By inductive hypothesis, for every decision point descendant $\hat{s}'$ of $\hat{s}$, at every timestep t, $\mathcal{R}_{\hat{s}'}$ receives the same utility vector as $\mathcal{R}_{s'}$ where $s' = \delta(\hat{s}')$, and thus produces the same behavioral strategy $\boldsymbol{r}^t_{s'} = \hat{\boldsymbol{r}}^t_{\hat{s}'}$. Thus, at any timestep t the utility vector $\hat{\boldsymbol{v}}^t(\hat{s}*)$ that is passed to $\mathcal{R}_{\hat{s}}$ is given by

$$\begin{aligned}
\hat{\boldsymbol{v}}^t(\hat{s}a) &= \hat{\boldsymbol{u}}^t(\hat{s}a) + \sum_{\hat{s}' : p_{\hat{s}'} = \hat{s}a} \left\langle \hat{\boldsymbol{r}}^t_{\hat{s}'}, \hat{\boldsymbol{v}}^t(\hat{s}'*) \right\rangle \\
&= \boldsymbol{u}^t(sa) + \sum_{\hat{s}' : p_{\hat{s}'} = \hat{s}a} \left\langle \boldsymbol{r}^t_{s'}, \boldsymbol{v}^t[s'*] \right\rangle \\
&= \boldsymbol{u}^t(sa) + \sum_{s' : sa \in P_s} \left\langle \boldsymbol{r}^t_{s'}, \boldsymbol{v}^t[s'*] \right\rangle \\
&= \boldsymbol{v}^t(sa)
\end{aligned}$$

where once again we use the notation $s' := \delta(\hat{s}')$, and the inductive hypothesis is used in the second equality on every term in the sum. □

Algorithm 4.8 (ConstructTB-DAG): Constructing the TB-DAG. Inputs: imperfect-recall game Γ, player i

```
procedure MakeDecisionPoint($B$)        ▷ $B \subseteq \mathcal{H}$ is a belief
    if a decision point $j$ with belief $B$ already exists then return $j$
    if $B = \{z\}$ for $z \in \mathcal{Z}$ then return new terminal node with belief $\{z\}$
    $j \leftarrow$ new decision point with belief $B$
    for each prescription $\boldsymbol{a} \in \mathcal{A}_i(B)$ do
        add edge $j \rightarrow$ MakeObservationPoint($B\boldsymbol{a}$)
    return $j$
procedure MakeObservationPoint($H$)
    $s \leftarrow$ new observation point
    for each $B \in$ SplitBelief$_i(H)$ do
        add edge $s \rightarrow$ MakeDecisionPoint($B$)
    return $s$
```

4.5 DAG Decision Problems in Team Games

In Section 4.3.3, it emerged that applying the CFR procedure to the belief game produced by MakeBeliefGame suffers from the size of the game to solve, which may grow exponentially fast as shown in Section 4.3.2. In this section, we show how DAG decision problems can greatly reduce the inefficiencies caused by the previous construction.

The main observation is that MakeBeliefGame enforces perfect recallness of the belief game by including the players' sequences in the infoset definition. On the other hand, the strategic aspect of the game is governed solely by the nodes contained in beliefs. Once the set of possible nodes is fixed, the exact sequence of prescriptions and observations is not relevant, as the game will evolve identically from that point onwards. This observation leads to considering a DAG structure for the decision problems, where decision nodes are identified by beliefs.

The Nash equilibrium problem in $\tilde{\Gamma}$, namely (4.1), indeed guarantees that both players' utility vectors will be consistent with respect to these DAG-form decision problems. We will call the resulting DAG decision problems the *team belief DAGs* (TB-DAGs)[4.12]. Therefore, using DAG-CFR as the regret minimizer for both players, we recover the regret guarantee of Theorem 4.20 with *per-iteration complexity* proportional to the total size of both DAGs.

However, this proposed algorithm still depends on the size of $\tilde{\Gamma}$, because, naively, to construct the DAG representations, one first constructs the augmented game $\tilde{\Gamma}$, and only then does the merging of decision points to create the DAGs. We therefore describe an algorithm ConstructTB-DAG that recursively constructs the team belief DAGs *directly from the original game* Γ, thus bypassing the construction of $\tilde{\Gamma}$. Therefore, we have the following result. For each player $i \in \{▲, ▼\}$, let E_i be the number of edges in the TB-DAG of player i.

[4.12] We keep the name *team belief DAG* for continuity with previous versions of the work, even though it applies equally well in the team and imperfect recall settings.

Theorem 4.24 (TB-DAG and CFR)**.** *Suppose that both players run* ConstructTB-DAG *to construct their strategy spaces $\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}$, and then run* DAG-CFR*. Then their average strategy profile converges at the rate shown in Theorem 4.20, and the per-iteration runtime complexity is $O(E_{\blacktriangle} + E_{\blacktriangledown})$.*

Proof. ConstructTB-DAG is designed to construct a DAG-form decision problem whose tree form corresponds precisely to the decision problem faced by player i in the belief game. Thus, it only remains to show that Theorem 4.23 applies. That is, we need to show that, in the belief game $\tilde{G}$, the utility vectors $\hat{\boldsymbol{u}}$ that would be observed by CFR for ▲ (the same proof would hold for ▼) are such that $\hat{\boldsymbol{u}}(\tilde{z}) = \hat{\boldsymbol{u}}(\tilde{z}')$ whenever histories $\tilde{z}, \tilde{z}' \in \tilde{\mathcal{Z}}$ of the belief game represent the same history $z \in \mathcal{Z}$ of the original game. But indeed, for any pure strategy $\tilde{\boldsymbol{y}} \in \tilde{\mathcal{Y}}$, by Theorem 4.14 there is a pure strategy $\boldsymbol{y} \in \tilde{\mathcal{Y}}$ such that $\tilde{\boldsymbol{y}}(\tilde{z}) = \tilde{\boldsymbol{y}}(\tilde{z}') = \boldsymbol{y}(z)$. Thus, if ▼ plays $\tilde{\boldsymbol{y}}$, the utility observed by ▲ will be given by

$$\hat{\boldsymbol{u}}(\tilde{z}) = \tilde{\boldsymbol{p}}(\tilde{z})\tilde{\boldsymbol{y}}(\tilde{z}) = \boldsymbol{p}(z)\boldsymbol{y}(z) = \tilde{\boldsymbol{p}}(\tilde{z}')\tilde{\boldsymbol{y}}(\tilde{z}') = \hat{\boldsymbol{u}}(\tilde{z}')$$

which is indeed consistent in the required sense. □

4.5.1 Size Analysis of the TB-DAG

The per-iteration runtime above depends on the number of edges in the TB-DAGs, so it is important to bound this number. We will do so now. Here, we use the same notation as in Section 4.3.2.

Theorem 4.25. *For each player i, we have $E_i \le |\mathcal{H}|(b+1)^k$.*

Proof. Let P be a (nonterminal) public state of player i, and P' be the set of descendants of P. let $\mathcal{I}_i(P) = \bigcup_{h \in P} \mathcal{I}_i(h)$ be the set of last-infosets. Consider a pure strategy $\boldsymbol{\pi} \in \Pi_i$. For each last-infoset $I \in \mathcal{I}_i(P)$, let $\boldsymbol{\pi}_P$ be the partial strategy defined only on infosets $I \in \mathcal{I}_i(P)$, by $\pi_P[I] = \boldsymbol{\pi}(I) \in A_I$ if $\boldsymbol{\pi}$ plays to at least one node in I, and $\pi_P[I] = \bot$ if it does not. There are thus at most $(b+1)^k$ such possible partial strategies, since $|\mathcal{I}_i(h)| \le k$ by definition. By construction, each partial strategy $\boldsymbol{\pi}_P$ completely determines which nodes in P are reached by $\boldsymbol{\pi}$, as well as the actions played at any such nodes. Thus, $\boldsymbol{\pi}_P$ induces a disjoint collection of observation points $S \subseteq \mathcal{S}$ such that we have $\boldsymbol{Ba} \in S$ for each observation point $B \subseteq P$. Now, each observation point $\boldsymbol{Ba}$ has at most one incoming edge and at most $|\boldsymbol{Ba}|$ outcoming edges. Since the observation points $\boldsymbol{Ba}$ are disjoint and have a total size at most $|P'|b$, the total number of edges at public state P is at most $|P'|(b+1)^k$. The proof finishes by summing over public states, noting that every history is in exactly one public state. □

Thus, from Theorem 4.24 and Theorem 4.25, it follows that:

Theorem 4.26 (Main theorem)**.** *Any given imperfect-recall game Γ can be solved by constructing the TB-DAGs using* ConstructTB-DAG *and running* DAG-CFR*. After T iterations, the average strategy profile will be an $O(\epsilon)$-Nash equilibrium where ϵ is as in Theorem 4.20. The per-iteration complexity is $O(|\mathcal{H}|(b+1)^k)$.*

Before proceeding, it is instructive to briefly compare Theorem 4.26 to Theorem 4.19. The latter result gives a per-iteration complexity that is $O(b^{d(2k+1)})$. Thus, Theorem 4.26 is strictly superior: for Theorem 4.19 to be superior, we would need to have $b^{d(2k+1)} < |\mathcal{H}|(b+1)^k$, which is impossible for $d \geq 1, k \geq 1, b \geq 2$. We give a more detailed comparison between the two bounds in Section 4.7.3.

4.5.2 Fixed-Parameter Hardness

Given the above result, one may ask whether the b can be removed more generally. It turns out that it cannot. Before proceeding, we need to introduce some basic concepts surrounding *fixed-parameter tractability*.

Definition 4.27. A problem is *fixed-parameter tractable* with respect to a parameter k if it admits an algorithm whose runtime on inputs of length N is $f(k)\mathsf{poly}(N)$, for some arbitrary function f.

The *k-CLIQUE* problem is to, given a graph Γ and an integer k, decide where Γ has a k-clique. The computational assumption $\mathsf{FPT} \neq \mathsf{W}[1]$ states that k-CLIQUE is not fixed-parameter tractable. It is implied by the exponential time hypothesis [59].

Theorem 4.28. *Assuming $\mathsf{FPT} \neq \mathsf{W}[1]$, there is no algorithm for computing the mixed Nash value of a* one-player *game of imperfect recall whose runtime has the form $f(k)\mathsf{poly}(|\mathcal{H}|)$ where f is an arbitrary function.*

Proof. We reduce from k-CLIQUE. Given a graph $G = (V, E)$, we construct the following game with a single team with two members. Nature selects two vertices $v_1, v_2 \in V$ independently and uniformly at random. Then, both team members privately observe the vertices that have been assigned to them, and select bits $b_1, b_2 \in \{0, 1\}$.

Utilities are defined as the sum of the following terms.

- If $v_1 = v_2$ and $b_1 \neq b_2$ then the team scores $-|V|$.
- If $b_1 = b_2 = 1$ and $(v_1, v_2) \neq E$ and $v_1 \neq v_2$, then the team scores $-|V|$.
- If $b_1 = b_2 = 1$ then the team scores 1.

We claim that the value of this game is $\geq k$ if and only if G has a k-clique. Clearly if G has a k-clique then the value of the game is at least k: if both members play bits according to the k-clique (*i.e.*, $b_i = 1$ if v_i is in the clique) then they will score k.

Conversely, suppose there is no k-clique. Note first that we can assume WLOG that the two members play the same strategy. (If they do not, they get utility at most 0, but playing the all-zeros vector also gets utility 0.) A pure strategy is identified by the subset $S \subseteq V$ of vertices at which the player plays 1. Since both members are playing the same strategy, the utility function reduces to $|S| - |V| \cdot m$, where $m = |\{(i, j) : i \neq j; i, j \in S; (i, j) \notin E\}|$ is the number of times the clique constraint is violated. Once again, this number is ≤ 0 unless S is a clique, and if S is a clique then $|S| < k$ since there is no k-clique. □

Thus, it is impossible to replace the b in Theorem 4.26 with any absolute constant.

4.5.3 Branching Factor Reduction

Despite the worst-case hardness of removing the b in Theorem 4.26, it turns out that, for a natural class of games, we *can* remove b. In this subsection we will discuss games with *action recall*, and prove that in such games, it is without loss of generality to assume that the branching factor is 2. Intuitively, a player i has action recall if it remembers the full sequence of actions she has taken in the past (including the timesteps at which such actions were taken). More formally:

Definition 4.29. At a node $h \in \mathcal{H}$, let $(a_1, \dots, a_L) \in \mathcal{A}^L$ be the list of actions taken on the $\varnothing \to h$ path. Define the *action sequence* of player i as the sequence $(a'_1, \dots, a'_L) \in (\mathcal{A} \sqcup \{\perp\})^L$ where $a'_\ell = a_\ell$ if action a_ℓ was taken by player i, and $a'_\ell = \perp$ otherwise. We say that player i has *action recall* if, for every infoset I of player i, every node in I shares the same action sequence.

Theorem 4.30. *Given a two-player imperfect-recall game Γ where both players have perfect action recall, there exists another strategically-equivalent game Γ' such that the branching factor of Γ' is at most 2 at each $h \in \mathcal{H}_\blacktriangle \cup \mathcal{H}_\blacktriangledown$, the parameter k in Γ' is the same as it in Γ, and the size of the game has increased by a factor of $O(\log |\mathcal{A}|)$.*

Proof. To every action $a \in \mathcal{A}$ we associate a unique bitstring of length at most $\ell = O(\log |\mathcal{A}|)$. Assume without loss of generality, for simplicity of notation, that all such bitstrings end with a 0. We will call bitstrings $\tilde{a} \in \{0, 1\}^{<\ell}$ "partial actions".

We replace every internal node $h \in \mathcal{H} \setminus \mathcal{Z}$ with a binary tree of depth ℓ, where bitstrings that are not prefixes of any action $a \in A_h$ are pruned. If h and h' are in the same infoset I, then for every partial action $\tilde{a} \in \{0, 1\}^{<\ell}$ we connect $h\tilde{a}$ and $h'\tilde{a}$ in an infoset, which we will call $I\tilde{a}$. This creates a new game G', whose parameters we must now analyze.

For each node $h \in \mathcal{H} \setminus \mathcal{Z}$ and action a, G' has createad at most $O(\log |\mathcal{A}|)$ additional nodes (namely, the nodes $h\tilde{a}$ where $\tilde{a}$ is a prefix of a). Thus, the size of G' is at most $O(|\mathcal{H}| \log |\mathcal{A}|)$.

It thus remains only to bound the information complexity of G'. Let P be a public state of player i in G'. By construction of action sequences, P contains either only nodes in $\mathcal{H}_i$, or only nodes not in $\mathcal{H}_i$. In the latter case there is nothing to check. In the former case, we have $P \subseteq \{h\tilde{a} : h \in P'\}$ for some public state P' of G, and partial action $\tilde{a} \in \{0, 1\}^{<\ell}$. Now let I be a last-infoset of P' in G. Then I induces at most one last-infoset in P: namely, if I overlaps P, then this infoset is simply $I\tilde{a}$; otherwise, it is the infoset $I\tilde{a}'$ where $\tilde{a}' \in \{0, 1\}^{\ell-1}$ is the partial

	Team vs Player	Team vs Team
TMECor	NP-complete [176]	Δ_2^{P}-complete (Theorems 4.36 and 4.37)
TME	NP-complete [176]	Σ_2^{P}-complete (Theorems 4.33 and 4.34)

Table 4.9: *Summary of most of the complexity results shown in Section 4.6.*

action such that action $\tilde{a}'0$ leads to P (which must be uniquely defined by definition of action recall). Thus, P has as many last-infosets as P', so the information complexity of G' is, at most, the information complexity of G. □

Corollary 4.31. *In games where both players have action recall, Theorem 4.26 applies with the per-iteration runtime replaced with $O(3^k|\mathcal{H}|\log|\mathcal{A}|)$.*

4.6 Complexity of Adversarial Team Games

Here, we state and prove several results about the *complexity* of finding various equilibria in timeable two-player zero-sum games of imperfect recall.

In all cases, the goal is to solve the following promise problem: given game Γ, threshold value v, and error $\epsilon > 0$ (where all the numbers are rational), determine whether the (mixed or behavioral) value of the game is $\geq v$, or $< v - \epsilon$. The allowance of an exponentially-small error is to circumvent issues of bit complexity that arise due to the fact that exact behavioral max-min strategies may not have rational coefficients [176]. Throughout this section, it will often be convenient to formulate the hardness gadgets in terms of adversarial team games. We will thus freely utilize the analogy between adversarial team games and coordinator games. For mixed Nash and behavioral Nash respectively, we will refer to the problems as MIXED and BEHAVIORAL.

Although we do not explicitly state it in the theorem statements, all the hardness results are proven by constructing adversarial team games in which both teams have a constant number of players.

Theorem 4.32 ([63, 176, 291])**.** *Finding the optimal strategy in a one-player, timeable game of imperfect recall is* NP*-hard.*

The above result also shows, by the PCP theorem [144], that there exists an absolute constant ϵ such that computing the optimal value in a team game with no adversary to accuracy ϵ is NP-hard. Finally, the information complexity of the game used in the above construction is[4.13] $k = n$, and the branching factor can be made an absolute constant by splitting the root chance node into $\Theta(\log m)$ layers. Finally, the size of the game is $O(mn)$. Thus, Theorem 4.26 implies a SAT-solving algorithm whose runtime is $2^{O(n)}$. Thus, in particular, the appearance of k in

[4.13] Here we use the ordering of the players: namely, we have $k = n$ only because P1 plays before P2. If the order of the players were flipped, we would instead have $k = m$.

the exponent in Theorem 4.26 is unavoidable: if the k were replaced by any $o(k)$ term, then SAT would have an $2^{o(n)}$-time algorithm, violating the commonly-believed exponential time hypothesis.

4.6.1 Behavioral Max-Min Strategies

We first show results for BEHAVIORALMAXMIN. In particular, we will show that it is Σ_2^{P}-complete, first by showing inclusion and then constructing a gadget game to show completeness. (Recall that the inclusion will require an ϵ-approximation because exact behavioral max-min strategies may contain irrational values.)

Theorem 4.33. BEHAVIORALMAXMIN *is in* Σ_2^{P}. *If* ▼ *has perfect recall, it is in* NP.

Proof. Consider a behavioral max-min strategy represented by a distribution over the actions at each information set I. Let $\delta > 0$, and consider rounding each entry of the behavioral-form strategy by at most an additive δ so that the resulting strategy is rational. Let $\boldsymbol{x}'$ be the correlation plan of the resulting strategy. Thus, for any given terminal node s, the resulting reach probability $x'[s]$ is perturbed by at most an additive $O(N\delta)$ where N is the number of nodes in the game. Thus, $\|\boldsymbol{x}' - \boldsymbol{x}\|_1 \le O(N^2\delta)$. Thus, for any realization-form strategy $\boldsymbol{y}$ for the opponent, we have $|\langle \boldsymbol{x}' - \boldsymbol{x}, \mathbf{A}\boldsymbol{y}\rangle| \le \|\boldsymbol{x}' - \boldsymbol{x}\|_1 \|\mathbf{A}\boldsymbol{y}\|_\infty \le O(N^2\delta)$, so x' is $O(N^2\delta)$-close to the optimal solution. Taking $\delta < O(\epsilon/N^2)$ thus concludes the proof. □

Theorem 4.34. BEHAVIORALMAXMIN *is* Σ_2^{P}*-hard, even for team games with a constant number of players and no chance.*

Proof. We first give a reduction involving chance, then show how to relax this condition. We reduce from ∃∀3-SAT, which is known to be Σ_2^{P}-complete [265]. The ∃∀3-SAT problem is to, given a 3-DNF formula $\phi(X, Y)$, determine whether $\exists X\ \forall Y\ \phi(X, Y)$ holds.

Given a 3-DNF formula ϕ with m clauses, n_1 variables in X, and n_2 variables in Y, construct the following game between △ with 3 players and ▽ with 3 players. Nature chooses three variables x_1, x_2, x_3 from X and three variables y_1, y_2, y_3 from Y. For each variable x_i (respectively y_i), Player i of △ (respectively ▽) is asked for an assignment to the variable.

If any two players of △ (respectively ▽) have the same variable but differ in their assignment, △ gets value $-M$ (respectively M) where M is a large value. In addition, △ gets value 1 if at least one term in the 3-DNF ϕ is satisfied by the assignments of △ and ▽.

Let $n = \max(n_1, n_2)$. We complete the proof by showing that △ gets at least $1/n^3$ if and only if $\exists x\ \forall y\ \phi(x, y)$ holds; otherwise, their value is at most 0. We first show that for large enough M, since players of △ cannot correlate, △'s pure strategies are dominant over non-pure ones.

Lemma 4.35. *Let $x \in X$ be a variable and $p \le 1/2$ be the probability that Player i plays their less-likely action for x in a behavioral strategy. If $p > 0$ and $M \ge n_1$, then this strategy is* *strictly dominated by the strategy under which Player i only plays their more-likely action.*

Proof. Whenever variable x is picked for Player i and one of their teammate (probability strictly larger than $1/n_1^2$), the penalty incurred by the two players is strictly more than $(M/n_1^2)(p(1-q)+q(1-p)) \geq (M/n_1^2)(p+q(1/2-p)) \geq (M/n_1^2)p$. On the other hand, Player i gains no more than 1 by playing their less-likely action (probability p/n_1). Hence, if $M \geq n_1$, any strategy with $p > 0$ is dominated by a pure strategy. □

The pure strategies of a player of △ (respectively ▽) are precisely the assignments in $\{0,1\}^X$ (respectively in $\{0,1\}^Y$). By a similar argument, it is straightforward to show that it is a dominant strategy for △ (respectively for ▽) to let all players pick the same assignment to avoid a large penalty.

The hardness then follows from the following observation. If $\exists X\ \forall Y\ \phi(X,Y)$ holds, then △ can play the corresponding assignment to force a value of at least $1/n^3$: no matter what assignment ▽ picks, at least one term in ϕ is true, which is discovered with a probability of at least $1/n^3$ (whenever all the variables in such a term are picked by Nature). On the other hand, if $\exists X\ \forall Y\ \phi(X,Y)$ does not hold, then no matter what assignment △ picks, there is an assignment that ▽ can pick such that none of the terms is satisfied, which forces a value of 0 for △.

To show that the same hardness holds even when there is no chance, we use the following gadget to eliminate the need of Nature. Let us introduce a new △-player called △-Nature and a new ▽-player called ▽-Spoiler. The gadget will be such that △-Nature can incur a large penalty whenever they do not mimic perfectly Nature's behavior. More concretely, as Nature in the construction above, △-Nature picks $s = (x_1, x_2, x_3, y_1, y_2, y_3) \in X^3 \times Y^3$. ▽-Spoiler then guesses △-Nature's choice by picking $s' \in X^3 \times Y^3$. △ receives $-N(n_1^3 n_2^3 - 1)$ If $s = s'$, otherwise N, where N is a large number. The game then continues as in the construction above.

By a similar argument to the one used in the proof of the lemma above, △-Nature's dominant strategy is to pick s uniformly at random. Since △ cannot correlate, the game plays exactly like the construction above; △ can force a value of $1/n^3$ if and only if $\exists X\ \forall Y\ \phi(X,Y)$ holds. □

4.6.2 Mixed Nash Equilibria

We now show results for MixedNash, namely, we will show that MixedNash is Δ_2^{P}-complete, again by showing inclusion first and then completeness. Unlike for BehavioralMaxMin (Theorem 4.33), here we will directly construct a separation oracle, and thus be able to recover algorithms for *exact* computation.

Theorem 4.36. MixedNash *is in* Δ_2^{P}, *even for exact computation* ($\epsilon = 0$).

Proof. Let $\mathcal{X} \subset \mathbb{R}^m, \mathcal{Y} \subset \mathbb{R}^n$ be the space of realization-form pure strategies of both players, and $\mathcal{A}$ be the payoff matrix. Then our goal is to decide whether the polytope

$$\mathcal{X}^* := \left\{ \boldsymbol{x} \in \mathbb{R}^m : \begin{array}{cc} \text{①} & \boldsymbol{x} \in \operatorname{conv} \mathcal{X}, \\ \text{②} & \boldsymbol{y}^\top \mathbf{A} \boldsymbol{x} \le v \ \forall \boldsymbol{y} \in \mathcal{Y} \end{array} \right\}$$

is empty. We will show how to separate over $\mathcal{X}^*$ with a mixed-integer linear programming oracle, which suffices to complete the proof because such a separating oracle can be used to run the ellipsoid algorithm.

Given a candidate solution $\boldsymbol{x}$, we check both constraints. If ② is violated for some $\boldsymbol{y}^* \in \mathcal{Y}$, then $\mathbf{A}\boldsymbol{y}^*$ is a separating direction; such $\boldsymbol{y}^*$ can be found by an integer programming oracle. If ① is violated, then a separating direction can be found because (strong) separation and optimization are equivalent for well-described polytopes [138], and optimization over $\operatorname{conv} \mathcal{X}$ is an integer program. □

Theorem 4.37. Mixed Nash *is* Δ_2^{P}*-hard, even for team games with a constant number of players and no chance.*

Proof. We reduce from Last-SAT, which is known to be Δ_2^{P}-complete [183]. The Last-SAT problem is to, given a 3-CNF formula $\phi(x)$, decide whether the lexicographically last satisfying assignment of ϕ has a 1 in the least-significant bit.

Given a 3-CNF formula ϕ with m clauses over a set of n variables $X = \{x_1, \dots x_n\}$, we construct the following zero-sum game with 3 players on each team. First, nature chooses 6 variables $x_1^{\triangle}, x_2^{\triangle}, x_3^{\triangle}, x_1^{\triangledown}, x_2^{\triangledown}, x_3^{\triangledown} \in X$ independently and uniformly at random. Player i of △ (respectively of ▽) is asked concurrently and independently to assign either true or false to the variable $x_i^{\triangle}$ (respectively $x_i^{\triangledown}$). The payoff for △ is a sum of terms, determined by the following conditions.

- If any two players of △ (respectively of ▽) assign different values to the same variable, △ receives $-N^2$ (respectively $+N^2$), where N is a large number.
- If any clause in ϕ is rendered false by the assignment of △ (respectively ▽), △ receives $-(n^2 + N)$ (respectively $+N$).
- If the variable shown to player 1 of △ (resp. of ▽) is $x_k \in X$ and they assign true to this variable, then △ receives $+2^{n-k}$ (respectively -2^{n-k}).
- If the variable shown to player 1 of △ is the last variable x_n and they assign false to this variable, then △ receives an additional penalty of -1.

It is straightforward to verify if N is large enough (e.g. $N = n^2 2^n$), for both △ and ▽, the dominant pure strategy is to let all the players in the same team pick the lexicographically last maximum-satisfying assignment $X' \subseteq X$. In particular, this strategy is also the dominant mixed strategy. Now consider △'s payoff when both teams play this pure strategy.

- If ϕ is not satisfiable, then every clause that is false under the assignment X' is detected with a probability of at least $1/n^3$ (whenever all the variables in the clause are picked by Nature for the same team). This means the second term yields to △ an expected payoff at most $-1/n$. Other terms yields a non-positive expected payoff.

- If ϕ is satisfiable, then the only non-zero expected payoff for △ comes from the last term, which is 0 if $x_n \in X'$, otherwise $-1/n$.

Therefore, the value of the game is at least 0 if and only if the Last-SAT instance is true; otherwise, the value is at most $-1/n$.

To eliminate nature from the construction, we use a similar gadget to the one in the proof of 4.33. Let us introduce a new △-player called △-Nature and two new ▽-player called ▽-Spoiler and ▽-Anticorrelator. The gadget will be such that ▽-Spoiler (respectively ▽-Anticorrelator) can impose a strictly negative expected payoff whenever △-Nature does not mimic perfectly Nature's behavior (respectively whenever other players of △ correlate with △-Nature).

More concretely, the game proceeds as follows: First, △-Nature picks a sextuple from X^6. ▽-Spoiler then decides whether to guess the choice of △-Nature without observing it. If they do, △ receives $+1$, and an additional $-n^6$ if ▽-Spoiler guesses correctly. If ▽-Spoiler decides not to guess, then the game continues as before: each player is shown their variable and nothing else, to which they assign either true or false. Then, ▽-Anticorrelator can choose to do nothing, or to pick an $i \in 1, 2, 3$. In the former case, the payoff of △ is computed as in the construction above. In the latter case, ▽-Anticorrelator observes the variable $x_i^{\triangle}$ shown to player i of △ and the truth value that player i assigns to this variable. ▽-Anticorrelator then guesses the other 5 variables picked by △-Nature; △ receives $+1$, and an additional $-n^5$ if ▽-Anticorrelator guesses correctly.

To see that this construction works, notice that if △-Nature does not pick the sextuple uniformly at random, then ▽-Spoiler can guess correctly the sextuple with a probability strictly larger than $1/n^6$, thus yielding a strictly negative reward to △. Similarly, if player i's assignment to $x_i^{\triangle}$ depends on the other 5 variables (which they cannot observe in the construction with chance above), then ▽-Anticorrelator can guess correctly with a probability strictly larger than $1/n^5$. Therefore, △-Nature's dominant strategy is to pick the sextuple uniformly at random; it is also dominant for the other 3 players of △ not to correlate to △-Nature. It is then straightforward to see that this game is equivalent to the game with chance above, and the value of the game is 0 if and only if the Last-SAT instance is true. □

4.7 Discussion

In the following section we discuss important details that may help the interested reader in clarifying some technical aspects of our contributions.

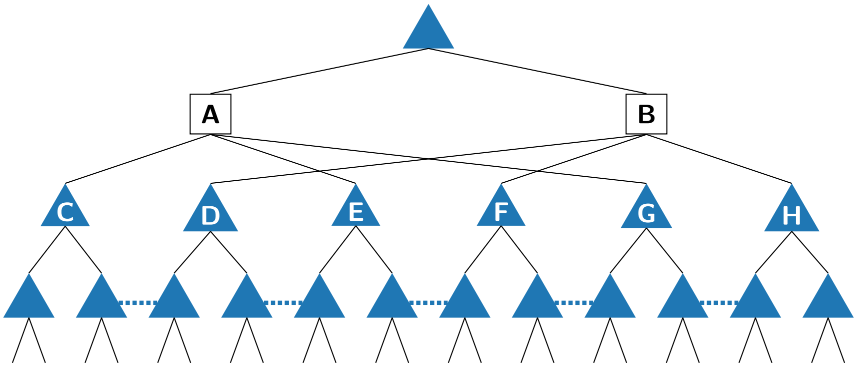


Figure 4.10: *A game showing that public state-based approaches do not subsume inflation.*

4.7.1 Public States vs Observations

In this section, we discuss in depth the difference between public *states* and public *observations*. Intuitively, the difference is that observations are *localized* to a particular node in the TB-DAG: if a fact is public to the team *conditional on the part of the team strategy that has been played to reach this point*, then it is an observation. On the other hand, public *states* only encode *unconditionally* public information. As we will see, using observations is strictly preferable to public states from both conceptual and theoretical perspectives.

Comparision to using public states. We envision an alternative construction of the TB-DAG in which the team coordinator observes only the *public state* containing the current node. That is, the definition of SPLITBELIEF is replaced by:

$$\textsc{SplitBelief}_i^{\text{pub}}(H, h) := H \cap P \text{ where } h \in P \in \mathsf{P}_i.$$

and $\textsc{SplitBelief}_i^{\text{pub}}(H)$ defined analogously. Then, in ConstructTB-DAG, we replace $\textsc{SplitBelief}_i(H)$ with $\textsc{SplitBelief}_i^{\text{pub}}(H)$. We will call this new construction the *public-state TB-DAG* and spend the rest of this subsection contrasting it with the (observation) TB-DAG constructed by ConstructTB-DAG.

Our first result is that the TB-DAG can never be too much larger than the public state TB-DAG:

Proposition 4.38. *Let N and N' be the number of nodes in the TB-DAG and public state TB-DAG respectively. Then $N \le 2pN'$, where p is the largest size (in number of nodes) of any belief in the public state TB-DAG.*

Proof. Let B be any belief in the public state TB-DAG. In the (non-public-state) TB-DAG, B splits into disjoint beliefs $B_1, \dots, B_m$. Let $A_1, \dots, A_m$ be the sizes of the prescription spaces at $B_1, \dots, B_m$ respectively. Then B has $A_1 A_2 \dots A_m$ children, so B induces $1 + A_1 A_2 \dots A_m$ nodes in the public state TB-DAG. On the other hand, the beliefs $B_1, \dots, B_m$ in the TB-DAG will have $A_1, \dots, A_m$ children respectively, accounting for a total of $m + A_1 + \dots + A_m \le 2mA_1 \dots A_m$

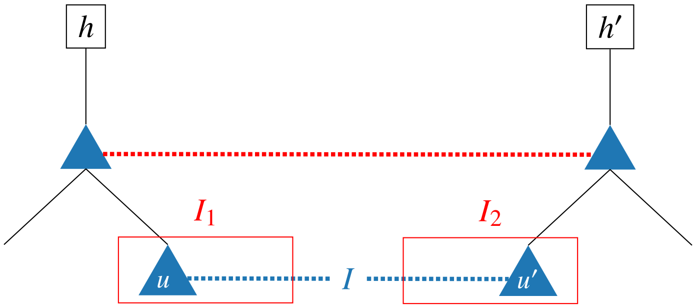


Figure 4.11: *A pictorial representation of the proof of Proposition 4.39. Since h and h' can be played simultaneously but u and u' cannot, there must be an infoset like the red dotted one connecting a child of h to a child of h'. Therefore, inflation cannot break existing edges between played nodes.*

nodes. Now, observing simply that $m \le p$ completes the proof. □

Thus, using observations is never much worse than using public states.

Comparision to using inflated public states. *Complete inflation* [168], which we simply call *inflation* for short, is an algorithm that splits an infoset I into two infosets $I = I_1 \sqcup I_2$ if no pure strategy of the team can simultaneously play to a node in I_1 and a node in I_2, and repeats this process until no more such splits are possible. This preserves strategic equivalence. However, inflation can lead to the break-up of public states, which, in turn, reduces the size of public state TB-DAG.

Indeed, consider the game in Figure 4.10. Due to the information sets marked in the last layer of the game tree, the connectivity graph contains a path C—D—E—...—H. Therefore, {C, D, ..., H} form a public state. Also, it is possible for the combinations CEG and DFH to be reached (if the player at the root plays left or right, respectively). Therefore, CEG and DFH are beliefs in the public-state TB-DAG. In the observation TB-DAG, consider, for example, what happens if the left action is played at the root so that C, E, and G are all reached. Note that there are no edges connecting C, E, and G—the path connecting C to E in the connectivity graph passes through D, which is not reached; therefore, C, E, and G are three different observations and hence three different beliefs, resulting in an exponentially-smaller TB-DAG. Inflation would remove the nontrivial information sets in the second black layer, which would ultimately have the same effect in this example as using observations.

The number 3 is not special in this construction; it can be increased arbitrarily by simply increasing the number of children of A and B. Therefore, in particular, one can construct a family of games in which the public state TB-DAG (without inflation) has exponential size, while the (observation) TB-DAG has polynomial size.

The use of observations, however, removes the need for this step:

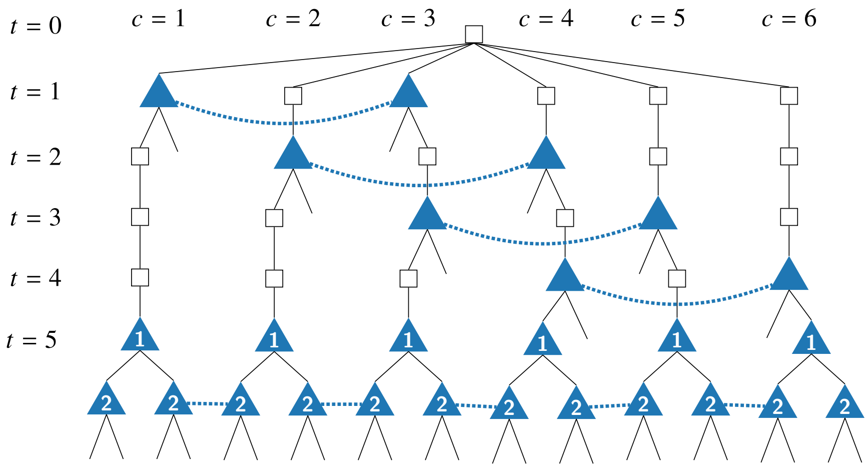


Figure 4.12: *The counterexample for Proposition 4.40, for $C = 6$.*

Proposition 4.39. *Given any team decision problem $\mathcal{T}$, the TB-DAG of $\mathcal{T}$ is the same no matter whether inflation is applied to $\mathcal{T}$ before the construction.*

Proof. Inflation operations affect the connectivity graph $\mathcal{G}_i$, thus changing the results of SPLITBELIEF$_i$ operations at a terminal node. Consider, therefore, any observation node O in the TB-DAG, and let $h, h' \in O$, such that $I = I_1 \sqcup I_2$ is an inflatable infoset and $h \preceq u \in I_1$ and $h' \preceq u' \in I_2$. We need to show that inflating I into I_1 and I_2 cannot remove the (h, h') edge in $\mathcal{G}_i[O]$.

Assume for contradiction that inflating would remove the (h, h') edge and that therefore SPLITBELIEF$_i$ would split h, h' into two different beliefs. We have that O is a valid observation node, so it is possible for the player to play to both nodes h and h' simultaneously. But then there must be an infoset I' connecting some node on the $h \to u$ path to some node on the $h' \to u'$ path—otherwise, it would be possible for the player to play to both u and u' simultaneously, which violates inflatability of I. But then there is still an (h, h') edge in $\mathcal{G}_i[O]$, which is a contradiction. □

Although inflation *can* be performed efficiently, not requiring it as a preprocessing step simplifies the code and makes for a conceptually cleaner construction. However, the benefits of observations go beyond making inflation unnecessary. In fact, even with inflation, there are still cases in which using observations instead represents an exponential improvement.

Proposition 4.40. *There exists a family of team decision problems in which the TB-DAG has polynomial size, but the public state TB-DAG has exponential size, even if inflation is applied as a preprocessing step before building the latter.*

Proof. The counterexample in Figure 4.10 would work if it were not for the fact that all of the infosets in the last layer inflate. Therefore, we use a similar gadget at the bottom of the game to prove this result but ensure that inflation does nothing.

Consider the following family of games, parameterized by an integer $C > 1$. First, nature picks an integer $c \in \{1, \dots, C\}$. Over the next $C - 2$ layers $t = 1, 2, \dots, C - 2$, if $c \in \{t, t + 2\}$, a player who cannot distinguish the two cases chooses an action $a \in \{0, 2\}$. If $c = t + a$, then the game continues; otherwise, the game ends.

Finally, P1, who has perfect information about c chooses between two actions numbered either c or $c + 1$. Then, player P2, observing P1's action number but *not* the value c, picks one of two options.

The resulting game is visualized in Figure 4.12. We observe the following things about it.

1. No infoset inflates: all nontrivial infosets have size 2, and it is easy to check that for all such infosets it is always possible to play to both nodes in them. This starkly contrasts the earlier counterexample, in which inflation was enough to achieve a small representation.
2. Every P2-node in layer $C - 1$ is in the same public state, and it is always possible to play to at least $C/2$ of them. Therefore, if using public-state-based beliefs, there will be a belief with $2^{C/2}$ prescriptions. Thus, the public-state-based team belief DAG, will have a size of at least $2^{C/2}$.

We claim that layer $t \le C - 1$ does not have too many beliefs. Let $B \subseteq \mathcal{H}_t$ be a belief, and in the below discussion, let $[a..b]$ denote the set of integers $\{a, \dots, b\}$.

1. $B \subseteq [t + 2..C]$. Since all nodes $j \ge t + 2$ must be played to, we have $B = [t + 2..C]$.
2. $B \nsubseteq [t + 2..C]$. Then let $j = \max(B \setminus [t + 2..C])$. Since $j \le t + 1$, we have $j - 2 \notin B$, since j and $j - 2$ are descendants of different actions taken at the infoset on layer $j - 2 < t$. Thus B does not contain any node $j' < j - 2$ either, since in $\mathcal{G}[\mathcal{H}_t]$ such nodes j' are only connected to j through $j - 2$.

   Thus, $B \cap [1..t + 1]$ is either $\{j\}$ or $\{j, j - 1\}$. Further, since all nodes $j \ge t + 2$ are played to and connected in $\mathcal{G}[\mathcal{H}_t]$, it follows that $B \cap [t + 2..C]$ is either $[t + 2..C]$ or empty. Thus, for each possible choice of $j \le t + 1$, there are at most 4 valid beliefs.

Thus, the number of beliefs in layer t is at most $4(t + 1) + 1 + 2 = 4t + 7$, where the +2 comes from counting the terminal beliefs, of which there are at most two.

Further, at each belief B, we claim that the number of active information sets is, at most, a constant. For $t < C - 1$ this is obvious since $\mathcal{H}_t$ contains only one information set (namely $\{t, t + 2\}$). For $t = C + 1$, by the above argument, we have $|B| \le 2$, so B overlaps at most two information sets.

Overall, we have that there are $O(C)$ beliefs at each of the C layers of the game, and such beliefs never touch more than $O(1)$ different infosets. This is also true at the final layer because each infoset contains only two nodes at most. It is therefore proven that the team belief DAG has a size of at most $O(C^2)$. □

A practical experiment backs up these results. When $C = 16$, using observations generates a DAG with around 1000 edges; using public states generates a DAG with 30 million edges.

4.7.2 Tree vs DAG Representation

Here we give an explicit example in which the TB-DAGs will be exponentially smaller than the game tree generated by MakeBeliefGame. This construction would work for most nontrivial adversarial team games, but for concreteness, consider the game Γ depicted in Figure 4.1. Call the leftmost terminal node in that diagram z. Consider adding another copy of Γ rooted at node z, and then repeating this process until ℓ copies of the game tree have been created, thus forming a game Γ^ℓ. That is, Γ^ℓ is the game in which Γ is played repeatedly until ℓ repetitions have been reached, or the terminal node reached is not z.

Note that, when running MakeBeliefGame on Γ, multiple copies of node z will appear. Thus, the number of nodes in the auxiliary game will be exponential in ℓ. However, in the TB-DAG, after the ith repetition of the game finishes, the belief will always be $\{z_i\}$ (where z_i is the copy of z in the ith repetition of the game). Thus, the size of the TB-DAG will scale linearly with ℓ. Thus, as ℓ grows, the TB-DAG will be exponentially smaller than the auxiliary game, and in particular the TB-DAG will have polynomial size while the auxiliary game will have exponential size.

4.7.3 Definition of Information Complexity and Comparison of Bounds

We discuss the comparison between the bounds from Theorem 4.19 and Theorem 4.25 in more detail.

In Section 4.3, we defined the information complexity as the maximum number of *last-infosets* in any public state. This definition was made with Theorem 4.26 in mind, because it is the correct parameterization for that result. For Theorem 4.19, however, we could have used a tighter parameterization. In particular, we could have defined a parameter κ as the number of infosets (not last-infosets) in any public state. Then $O(b^{2\kappa d+d})$ would be a valid upper bound in Theorem 4.19. One might ask how this new upper bound compares to that of Theorem 4.26. To this end, we now compare the two bounds.

Lemma 4.41. *$k \leq \kappa d$.*

Proof. Every last-infoset at a public state P will be an infoset intersecting some public state ancestor of P. Thus, there can be at most κd of these. □

Thus, the bound in Theorem 4.26 is at most

$$|\mathcal{H}|(b+1)^k \leq |\mathcal{H}|(b+1)^{\kappa d} < |\mathcal{H}|b^{2\kappa d} \leq b^{2\kappa d+d}$$

where we use the bounds $b \geq 2$ (which holds for every nontrivial game) and $|\mathcal{H}| \leq b^d$. Thus, we conclude that the bound in Theorem 4.26 is always strictly tighter than the bound in Theorem 4.19.

We also remark that in any case $\kappa \leq |\mathcal{H}|$ is a loose bound that still ensures that the overall bound in Theorem 4.20 is polynomial in $|\mathcal{H}|$.

4.7.4 Connection with Tree Decomposition

The public *state* TB-DAG can be viewed from the perspective of graphical models, specifically, using *tree decompositions*. Here, we review tree decompositions and show the tree decomposition-based perspective of the public-state TB-DAG.

Definition 4.42. Given a (simple) graph $\mathcal{G} = (V, E)$, a *tree decomposition*[4.14] is a tree $\mathcal{J}$, with the following properties:

1. the nodes of $\mathcal{J}$ are subsets of V, called *bags*;
2. for each edge $(u, v) \in E$, there is a bag containing both u and v; and
3. for each vertex $u \in V$, the subset of nodes of $\mathcal{J}$ whose bags contain u is connected.

Consider an arbitrary set of the form

$$\Pi = \{\boldsymbol{x} \in \{0,1\}^n : g_k(\boldsymbol{x}) = 0 \;\; \forall k \in [m]\}$$

where the g_ks are arbitrary constraints. Each constraint g_k has a *scope* $S_k \subseteq [n]$ of variables on which it depends. The *dependency graph* of Π is the graph $\mathcal{G}_\Pi$ whose nodes are the integers $1, \dots, n$, and where there is an edge (i, j) if there is a constraint whose scope S_k contains both i and j. For a subset $U \subseteq [n]$, a vector $\tilde{\boldsymbol{x}} \in \{0,1\}^U$ is *locally feasible* if $\tilde{\boldsymbol{x}} = \boldsymbol{x}_H$ for some $\boldsymbol{x} \in \Pi$. We will use Π_U to denote the set of all locally feasible vectors on U. Of course, $\Pi_{[n]} = \Pi$.

The main result of interest to us is a corollary of the junction tree theorem (*e.g.*, [293]), which allows an arbitrary set $\operatorname{conv}\Pi$ to be described with a constraint system whose size is related to the sizes of tree decompositions of $\mathcal{G}_\Pi$.

Theorem 4.43 ([293]). *Let $\mathcal{J}$ be a tree decomposition of $\mathcal{G}_\Pi$. Then $\boldsymbol{x} \in \Pi$ if and only if there are vectors $\boldsymbol{\lambda}_U \in \Delta(\Pi_H)$ for each bag U of $\mathcal{J}$, such that:*

$$\boldsymbol{x}_U = \sum_{\tilde{\boldsymbol{x}} \in \Pi_U} \boldsymbol{\lambda}_U(\tilde{\boldsymbol{x}}) \cdot \tilde{\boldsymbol{x}} \qquad \forall \text{ bags } U \text{ in } \mathcal{J}$$

$$\sum_{\substack{\tilde{\boldsymbol{x}} \in \Pi_U \\ \tilde{\boldsymbol{x}}_{U \cap V} = \tilde{\boldsymbol{x}}^*}} \boldsymbol{\lambda}_U(\tilde{\boldsymbol{x}}) = \sum_{\substack{\tilde{\boldsymbol{x}} \in \Pi_V \\ \tilde{\boldsymbol{x}}_{U \cap V} = \tilde{\boldsymbol{x}}^*}} \boldsymbol{\lambda}_V(\tilde{\boldsymbol{x}}) \qquad \forall \text{ edges } (U, V) \text{ of } \mathcal{J} \text{ and } \tilde{\boldsymbol{x}}^* \in \Pi_{U \cap V}$$

Intuitively, the first constraint says that every $\boldsymbol{x}_U$ must be a convex combination of locally feasible $\tilde{\boldsymbol{x}} \in \Pi_U$. This is of course a necessary condition. The second constraint says that marginal probabilities on edges (U, V) must be consistent with each other. This is also clearly a necessary condition, so the difficulty of proving the above result lies in showing that these two constraints are *sufficient*. We will not prove the result here, but we will use it as a black box.

[4.14] also known as a *clique tree* or *junction tree*

In this section, we will work with a representation slightly different from the realization form. For a player i in a coordinator game Γ and a pure strategy of that player, the *history form* of the strategy as the vector $\boldsymbol{x} \in \{0,1\}^{\mathcal{H}}$ where $\boldsymbol{x}(h) = 1$ if and only if the team plays all actions on the $\varnothing \to h$ path. (Of course, the realization form is just the subvector of $\boldsymbol{x}$ indexed by $\mathcal{Z}$.) As usual we will use Π for the set of pure strategies in history form, and $\boldsymbol{x} = \operatorname{conv} \Pi$. The history form is the set of vectors $\boldsymbol{x} \in \{0,1\}^{\mathcal{H}}$ satisfying the following constraint system.

$$\begin{aligned}
\boldsymbol{x}(\varnothing) &= 1 \\
\boldsymbol{x}(ha) &= \boldsymbol{x}(h) & &\text{if} \quad h \notin \mathcal{H}_i \\
\boldsymbol{x}(h) &= \sum_{a \in \mathcal{A}(h)} \boldsymbol{x}(ha) & &\text{if} \quad h \in \mathcal{H}_i \\
\boldsymbol{x}(ha)\boldsymbol{x}(h') &= \boldsymbol{x}(h'a)\boldsymbol{x}(h) & &\text{if} \quad h, h' \in I \in \mathcal{I}_i; a \in \mathcal{A}(h)
\end{aligned}$$

This constraint system defines a dependency graph $\mathcal{G}_\Pi$, whose nodes are nodes of the tree, and in which there is an edge (h, h') if either h' is a child of h, or h and h' are in the same infoset of player i.

Now consider the following tree decomposition of $\mathcal{J}$ of $\mathcal{G}_\Pi$. For each public state P, the tree decomposition $\mathcal{J}$ has a bag U_P that contains all nodes in P and all children of nodes in P. The edges of $\mathcal{J}$ are the obvious edges, connecting each U_P to $U_{P'}$ if $U_P \cap U_{P'} \neq \varnothing$.

One can check that, up to trivial reformulations (that is, removal of redundant variables and constraints), the constraint system from Theorem 4.43 associated with $\mathcal{J}$ is identical to the constraint system associated with the public state TB-DAG (via (4.2)). Thus, it is possible to interpret the public state TB-DAG entirely from the point of view of tree decompositions. We do not take this perspective here because using beliefs is more interpretable and understandable from a game-theoretic perspective.

4.7.5 Postprocessing Techniques that can be Used to Shrink the TB-DAG

In practice, ConstructTB-DAG is suboptimal in several ways. Here, we state some straightforward postprocessing techniques that can be used to shrink the size of the TB-DAG. These do not affect the theoretical statements as the primary focus of those is isolating the dependency on our parameters of interest, but they can significantly affect the practical performance, so we apply them in the experiments.

1. If two terminal nodes z, z' have the same sequence, we remove one of them (say, z') from our DAG because it is redundant, and alias $\boldsymbol{x}(\{z'\})$ to $\boldsymbol{x}(\{z\})$. If this removal causes a section of the DAG to no longer contain any terminal descendants, we also remove that section.

2. If a decision point in the TB-DAG has (at most) one parent and (at most) one child, we remove the decision point and directly connect the parent observation node to the grandchild decision points.

In particular, if the team has perfect recall, the above two optimizations are sufficient for the TB-DAG to coincide with the sequence form.

	Original game Γ							Belief Game $\tilde{\Gamma}$					Team $\blacktriangle$'s DAG		Team $\blacktriangledown$'s DAG	
	Nodes	Infosets		Sequences		Information		Nodes	Infosets		Sequences		Vertices	Edges	Vertices	Edges
Γ	$\lvert\mathcal{H}\rvert$	$\lvert\mathcal{I}_\blacktriangle\rvert$	$\lvert\mathcal{I}_\blacktriangledown\rvert$	$\lvert\Sigma_\blacktriangle\rvert$	$\lvert\Sigma_\blacktriangledown\rvert$	$\max_{\mathcal{P}}\lvert P\rvert$	k	$\lvert\tilde{\mathcal{H}}\rvert$	$\lvert\tilde{\mathcal{I}}_\blacktriangle\rvert$	$\lvert\tilde{\mathcal{I}}_\blacktriangledown\rvert$	$\lvert\tilde{\Sigma}_\blacktriangle\rvert$	$\lvert\tilde{\Sigma}_\blacktriangledown\rvert$	$\lvert\mathcal{J}_\blacktriangle \cup \mathcal{S}_\blacktriangle\rvert$	$\lvert E_\blacktriangle\rvert$	$\lvert\mathcal{J}_\blacktriangledown \cup \mathcal{S}_\blacktriangledown\rvert$	$\lvert E_\blacktriangledown\rvert$
^{3}K3 {3}	151	24	12	48	24	6	6	2119	486	12	1062	24	487	918	37	36
^{3}K4 {3}	601	32	16	64	32	12	8	45,049	4487	16	9800	32	2100	6711	49	48
^{3}K6 {3}	3001	48	24	96	48	30	12	6,768,601	267,184	24	574,588	48	54,255	336,944	73	72
^{3}K8 {3}	8401	64	32	128	64	56	16	617,929,873	13,194,749	32	27,978,704	64	1,783,926	15,564,765	97	96
^{3}K12 {3}	33,001	96	48	192	96	132	24	—	—	—	—	—	—	—	—	—
^{4}K5 {3,4}	7801	80	80	160	160	20	10	577,764,601	102,725	10,385	221,810	21,740	26,566	124,875	4621	15,415
^{4}K5 {4}	7801	120	40	240	80	60	15	174,273,721	11,739,640	40	25,581,730	80	998,471	4,658,070	121	120
^{3}L133 {3}	12,688	456	228	912	456	9	6	1,293,658	96,115	228	208,136	456	23,983	49,005	685	684
^{3}L143 {3}	40,409	800	400	1600	800	16	8	52,745,745	2,625,209	400	5,736,592	800	139,964	417,027	1201	1200
^{3}L151 {3}	19,981	1000	500	2000	1000	20	10	152,692,141	16,564,617	500	36,016,124	1000	150,707	496,196	1501	1500
^{3}L153 {3}	98,606	1240	620	2480	1240	25	10	1,833,113,016	67,400,747	500	147,671,104	1240	855,397	3,486,091	1861	1860
^{3}L223 {3}	15,659	1260	630	2884	1442	4	4	521,285	47,579	812	100,420	1624	32,750	45,913	2437	2436
^{3}L523 {3}	1,299,005	99,168	49,584	246,304	123,152	4	4	178,141,285	19,499,329	73,568	40,224,140	147,136	2,911,352	4,183,685	220,705	220,704
^{4}L133 {3,4}	159,001	1632	1632	3264	3264	9	6	985,916,371	475,081	135,322	1,011,500	292,400	79,351	158,058	75,157	155,475
^{3}D3 {3}	27,622	1023	513	2046	1020	9	6	70,704,118	3,235,954	765	5,501,789	1272	91,858	215,967	1522	1521
^{3}D4 {3}	524,225	10,924	5460	21,840	10,920	16	8	—	—	—	—	—	4,043,377	13,749,608	16,381	16,380
^{4}D3 {2,4}	663,472	6144	6144	12,285	12,285	9	6	—	—	—	—	—	514,120	1,217,310	486,442	1,155,144
^{6}D2 {2,4,6}	524,225	4096	4096	8190	8190	8	6	5,879,066,753	1,094,865	701,001	1,869,170	1,202,948	254,758	457,795	218,570	389,995
^{6}D2 {4,6}	524,225	5704	2488	10,920	5460	16	8	4,992,649,921	15,032,900	33,905	25,363,692	57,194	991,861	2,029,546	46,236	60,717
^{6}D2 {6}	524,225	6584	1608	12,922	3458	32	10	2,126,796,737	126,748,497	2532	208,964,598	4382	3,158,364	7,395,885	5551	5550

Table 4.13: *Game sizes of the equivalent representations proposed in this chapter (*i.e.*, belief game and TB-DAG) on several standard parametric benchmark team games. See Section 4.8 for a description of the games, and for a detailed description of the meaning of each column. Values denoted with '—' are missing due to out-of-time or out-of-memory errors.*

4.8 Experiments

This section investigates the empirical benefits brought about by applying the TB-DAG when computing mixed-Nash equilibria. As highlighted in Section 4.1, the literature on team games has been the one most concerned with the efficient computation of mixed Nash, with different works establishing benchmarks and proposing algorithms. We will, therefore, focus on comparing our approach against those previous related works. Our main results are reported in Table 4.13, which reports the size of the original games and our derived representations, and in Table 4.14, which reports the time required to solve those instances up to an approximation factor.

4.8.1 Experimental Setting

First, we give a complete description of the experimental setting in which the different algorithms are tested.

Game instances. We run experiments on commonly adopted parametric benchmarks in the team games literature. The following is the naming convention adopted for the instances considered:

- ^{n}**K**r: n-player Kuhn poker with r ranks [187].
- ^{n}**L**brs: n-player Leduc poker with a b-bet maximum in each betting round, r ranks, and s suits [275].
- ^{n}**D**d: n-player Liar's Dice with one d-sided die for each player [199].

The full description of these games can be found in Farina et al. [103]. For each game, the players belonging to team ▽ are represented along with the name. For example, ^{4}L133 {3,4} indicates a 4-player Leduc poker game with 1 bet each round, 3 ranks, 3 suits, where players 3 and 4 belong to team ▽ and are therefore coordinated by player ▼.

CFR Variant used. We used PCFR+. We remark that applying the CFR algorithm on the belief game and on the TB-DAG leads to identical iterations since the two representations are structurally equivalent (as proven in Section 4.5), and CFR is a deterministic algorithm. We therefore focus on the TB-DAG representation due to its efficiency. We also remark that the optimizations discussed in Sections 4.5.3 and 4.7.5 are applied during the experiments.

Baselines. We use the column generation framework of Farina et al. [103] and refined by Zhang et al. [305] (henceforth "**ZFCS22**") as the prior state-of-the-algorithm to compare the performance of CFR on the team belief DAG. ZFCS22 belongs to the family of column generation approaches adopted in the past literature in team games. ZFCS22 iteratively refines the strategy of each team by solving best-response problems using a tight integer program derived from the theory of extensive-form correlation [291]. We used the original code by the authors, which was implemented for three-player games in which a team of two players faces an opponent.

Hardware used. All experiments were run on a 64-core AMD EPYC 7282 processor. Each algorithm was allocated a maximum of 4 threads, 60GBs of RAM, and a time limit of 6 hours. ZFCS22 uses the commercial solver Gurobi to solve linear and integer linear programs. All CFR implementations are single-threaded, while we allowed Gurobi to use up to four threads.

4.8.2 Discussion of the Results

We now discuss the empirical results obtained by our algorithms.

Representation vs Game size. We analyze the size results from Table 4.13. The different orders of magnitude of the size of each representation and the original game highlight how the belief game construction increases the size of the game. Moreover, the striking difference between the two equivalent approaches of belief game and TB-DAG motivates the introduction of the latter: the direct construction of a decision problem and the more efficient representation brought by the DAG structure allow the construction of a substantially smaller representation. The benefits of the DAG imperfect-recall structure are especially beneficial in the case of Liar's Dice instances, which have a larger depth of the game tree. Overall, this comparison confirms the results from the worst-case bounds from Sections 4.3.2, 4.5.1 and 4.7.3. The exponential factor of inefficiency between the two representations agrees with the results from the discussion in Section 4.7.2.

There are also some minor remarks that are worth to be made. Whenever ▼ is a perfect-recall player (equivalently, when the team ▽ is composed of a single player), our constructions never increase the size of its decision problem. In the case of the belief game, we have that the adversary retains an identical number of information sets and sequences. In the case of the TB-DAG, the correspondence is $|\mathcal{D}_{\blacktriangledown}| = |\tilde{\mathcal{I}}_{\blacktriangledown}| + 1$ and $|\mathcal{S}_{\blacktriangledown}| = |\tilde{\Sigma}_{\blacktriangledown}|$

Running time. We focus on the time performance of CFR applied to the games from Table 4.14. The main observation is that the TB-DAG approach combined with the CFR algorithm has good performance in most of the games traditionally employed in the team game literature. In particular, impressive performance is achieved in games where the information complexity is low. This is the case of Leduc and Liar's Dice benchmarks (whose number of infosets and sequences in the original game are reported in Table 4.13). On the other hand, the column generation approaches struggle since the dimension of the pure strategy space depends exponentially on the number of information sets. The performance of our method depends crucially on having low information complexity. In fact, in games such as ^{3}K8 and ^{3}K12 where the information complexity is high, we observe poor performance even though the game tree is small. On the other hand, column generation techniques avoid this cost by considering an incrementally larger action space.

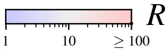
4.9 Conclusion

We have proposed a novel two-player zero-sum representation called the *team-belief DAG* for the computation of mixed Nash equilibria in timeable two-player zero-sum imperfect-recall games and team max-min equilibria with correlation in adversarial team games. We proposed a conversion mechanism that can be interpreted from the point of view of a perfect-recall coordinator which manages all the player's strategic choices while not accessing any information destined to be imperfectly recalled. The behavior of such a coordinator is defined based on beliefs and observations, novel concepts that allow an intuitive yet effective characterization. We also introduced a DAG decision problem structure for the TB-DAG to characterize more efficiently our conversion, by avoiding the pitfalls of an extensive-form characterization of the equivalent game. We theoretically analyzed the efficiency of our method through worst-case bounding of the size of the converted game, and we experimentally tested it on a set of customary benchmark games against a state-of-the-art approach from the literature. Our results are accompanied by novel complexity results that further characterize the hardness of computing equilibria in imperfect-recall games. In particular, we prove that computing a max-min strategy in behavioral strategies is Σ_2^{P}-hard even when the number of players is constant and there is no chance. Similarly, we prove that computing a Nash equilibrium in mixed strategies is Δ_2^{P}-hard.

Many directions departing from this work can be interesting for further development of the literature on imperfect-recall and team games. In particular, designing an algorithm able to exploit both the TB-DAG representation and the incrementality of column generation is an interesting approach to surpass the previous developments. Moreover, the TB-DAG construction may possibly be improved by preprocessing the game to reduce its information complexity, mitigating the exponential blowup due, while generalizing the notion of *triangle-free games* [97] to DAG games may extend the class of games that can be solvable in polynomial time. Another possible direction follows the more traditional two-player zero-sum literature. It aims to develop specific abstraction, dynamic pruning, and subgame-solving techniques tailored to our conversion's resulting two-player zero-sum games.

Finally, the question whether some of the results presented in this chapter can be extended to the non-timeable or absentminded imperfect-recall case is open. Timeability is used fundamentally in our method even to define a belief, so our methods break down completely for non-timeable games. Indeed, if we allow for *absentmindedness*, it is no longer even necessarily the case that distributions over pure strategies suffice to define the optimal strategy for a single-player game—for example, in the *absentminded driver problem* [244], the optimal strategy is a behavioral strategy that cannot be expressed as a mixture of pure strategies.

	Original game				TB-DAG			EFG		
Game {▼}	▲ Value	Nodes	Information			This chapter			CG	
Γ	u^*	$\lvert\mathcal{H}\rvert$	$\max_{\mathcal{P}} \lvert P\rvert$	k	Init	ϵ=10^{-3}	ϵ=10^{-4}	Init	ϵ=10^{-3}	ϵ=10^{-4}
^{3}K3 {3}	0.000	151	6	6	0.00s	0.00s	0.00s	0.00s	0.00s	0.00s
^{3}K4 {3}	-0.042	601	12	8	0.01s	0.00s	0.00s	0.00s	0.01s	0.02s
^{3}K6 {3}	-0.024	3001	30	12	1.03s	0.03s	0.12s	0.00s	0.14s	0.14s
^{3}K8 {3}	-0.019	8401	56	16	1m6s	4.73s	32.36s	0.01s	0.23s	0.32s
^{3}K12 {3}	-0.014	33,001	132	24	—	oom	oom	0.01s	0.84s	1.39s
^{4}K5 {3,4}	-0.037	7801	20	10	0.55s	0.03s	0.05s	—	—	—
^{4}K5 {4}	-0.030	7801	60	15	13.71s	1.59s	6.34s	—	—	—
^{3}L133 {3}	0.215	12,688	9	6	0.49s	0.02s	0.05s	0.02s	24.89s	45.96s
^{3}L143 {3}	0.107	40,409	16	8	1.39s	0.10s	0.48s	0.05s	2m 4s	6m 3s
^{3}L151 {3}	-0.019	19,981	20	10	1.54s	0.18s	0.50s	0.04s	3.06s	13.98s
^{3}L153 {3}	0.024	98,606	25	10	16.03s	1.24s	4.94s	0.12s	7m 23s	28m 13s
^{3}L223 {3}	0.516	15,659	4	4	0.13s	0.03s	0.08s	0.05s	13.48s	18.53s
^{3}L523 {3}	0.953	1,299,005	4	4	18.02s	11.26s	24.86s	6.83s	> 6h	> 6h
^{4}L133 {3,4}	0.147	159,001	9	6	2.03s	0.21s	0.92s	—	—	—
^{3}D3 {3}	0.284	27,622	9	6	0.80s	0.11s	0.40s	0.09s	11.05s	11.05s
^{3}D4 {3}	0.284	524,225	16	8	1m3s	22.54s	1m 28s	1.57s	3h 19m	3h 19m
^{4}D3 {2,4}	0.200	663,472	9	6	27.05s	2.31s	4.70s	—	—	—
^{6}D2 {2,4,6}	0.072	524,225	8	6	10.74s	1.72s	4.26s	—	—	—
^{6}D2 {4,6}	0.265	524,225	16	8	16.55s	3.80s	11.09s	—	—	—
^{6}D2 {6}	0.333	524,225	32	10	31.00s	30.20s	1m 11s	—	—	—

Table 4.14: *Runtime of our CFR-based algorithm (column '**CFR on TB-DAG**') using the team belief DAG form, compared to the prior state-of-the-art algorithms based on linear programming and column generation by Zhang et al. [305] ('**ZFCS22**'), on several standard parametric benchmark games. See Section 4.8 for a description of the games. Column* "Init" *represents the time needed to construct the structures needed for solving the games. This corresponds to fully exploring the TB-DAG and computing its full representation in memory in the TB-DAG case. Missing or unknown values are denoted with '—'. For each row, the background color of each runtime column is set proportionally to the ratio with the best runtime for the row, according to the logarithmic color scale* 

Runtimes that are more than two orders of magnitude larger than the best runtime for the row (i.e., *for which* $R > 10^2$*) are colored as if* $R = 10^2$.

Chapter 5

Solution Concepts, Algorithms, and Complexity of Hidden-Role Games

5.1 Introduction

Consider a multiagent system with communication where the majority of agents share incentives, but there are also hidden defectors who seek to disrupt their progress.

This chapter adopts the lens of game theory to characterize and solve a class of games called *hidden-role games*[5.1]. Hidden-role games model multi-agent systems in which a team of "good" agents work together to achieve some desired goal, but a subset of adversaries hidden among the agents seeks to sabotage the team. Customarily (and in this chapter), the "good" agents make up a majority of the players, but they will not know who the adversaries are. On the other hand, the adversaries know each other.

Hidden-role games offer a framework for developing optimal strategies in systems and applications that face deception. They have a strong emphasis on communication: players need to communicate in order to establish trust, coordinate actions, exchange information, and distinguish teammates from adversaries.

Hidden-role games can be used to model a wide range of recreational and real-world applications. Notable recreational examples include the popular tabletop games *Mafia* (also known as *Werewolf*) and *The Resistance*, of which *Avalon* is the best-known variant. As an example, consider the game *Mafia*. The players are split in an uninformed majority called *villagers* and an informed minority called *mafiosi*. The game proceeds in two alternating phases, *night* and *day*. In the night phase, the mafiosi privately communicate and eliminate one of the villagers. In the day phase, players vote to eliminate a suspect through majority voting. The game ends when one of the teams is completely eliminated.

We now provide several non-recreational examples of hidden-role games. In many cybersecurity

[5.1] These games are often commonly called *social deduction games*.

applications [124, 125, 286], an adversary compromises and controls some nodes of a distributed system whose functioning depends on cooperation and information sharing among the nodes. The system does not know which nodes have been compromised, and yet it must operate in the presence of the compromised nodes.

Another instance of problems that can be modeled as hidden-role games arises in *AI alignment*, *i.e.*, the study of techniques to steer AI systems towards humans' intended goals, preferences, or ethical principles [154, 158, 322]. In this setting, there is risk that a misaligned AI agent may attempt to deceive a human user into trusting its suggestions [239, 266]. AI debate [155] aims at steering AI agents by using an adversarial training procedure in which a judge has to decide which is the more trustful between two hidden agents, one of which is a deceptor trained to fool the judge. Miller et al. [216] proposes an experimental setting consisting of a chess game in which one side is controlled by a player and two advisors, which falls directly under our framework. The advisors pick action suggestions for the player to choose from, but one of the two advisors has the objective of making the team lose.

Hidden-role games also include general scenarios where agents receive inputs from other agents which may be compromised. For example, in *federated learning* (a popular category of distributed learning methods), a central server aggregates machine learning models trained by multiple distributed local agents. If some of these agents are compromised, they may send doctored input with the goal of disrupting the training process [225].

We aim to characterize optimal behavior in these settings, and analyze its computability.

Related work. To the best of our knowledge, there have been no previous works on general hidden-role games. On the other hand, there has been a limited amount of prior work on solving specific hidden-role games. Braverman et al. [33] propose an optimal strategy for *Mafia*, and analyze the win probability when varying the number of players with different roles. Similarly, Christiano [62] proposes a theoretical analysis for *Avalon*, investigating the possibility of *whispering*, *i.e.* any two players being able to communicate without being discovered. Both of those papers describe game-specific strategies that can be adopted by players to guarantee a specific utility to the teams. In contrast, we provide, to our knowledge, the first rigorous definition of a reasonable solution concept for hidden-role games, an algorithm to find such equilibria, and an experimental evaluation with a wide range of parameterized instances.

Deep reinforcement learning techniques have also been applied to various hidden-role games [8, 179, 269], but with no theoretical guarantees and usually with no communication allowed between players. A more recent stream of works focused on investigating the deceptive capabilities of large language models (LLMs) by having them play a hidden-role game [234, 297]. The agents, being LLM-based, communicate using plain human language. However, as before, these are not grounded in any theoretical framework, and indeed we will illustrate that optimal strategies in hidden-role games are likely to involve communication that does not bear resemblance to natural language, such as the execution of cryptographic protocols.

5.1.1 Main Modeling Contributions

We first here give an informal, high-level description of our game model. We also introduce our main solution concept of interest, called *hidden-role equilibrium*, and discuss the challenges it addresses. We will define these concepts in more formality beginning in Section 5.3.

We define a (finite) *hidden-role game* as an n-player finite extensive-form game Γ in which the players are partitioned at the start of the game into two teams. Members of the same team share the same utility function, and the game is zero-sum, *i.e.* any gain for one team means a loss for the other. We thus identify the teams as △ and ▽, since teams share the same utility function, but have opposite objectives. At the start of the game, players are partitioned at random into two teams. A crucial assumption is that one of the two teams is *informed*, *i.e.* all the members of that particular team know the team assignment of all the players, while this is not true for all players belonging to the other team. Without loss of generality, we use △ to refer to the uninformed team, and ▽ to refer to the informed one.[5.2]

To allow our model to cover *communication* among players, we formally define the *communication extensions* of a game Γ. The communication extensions are games like Γ except that actions allowing messages to be sent between players are explicitly encoded in the game. In the *public communication extension*, players are able to publicly broadcast messages. In the *private communication extension*, in addition to the public broadcast channel, the players have pairwise private communication channels.[5.3] In all cases, communication channels are *synchronous* and *authenticated*: messages sent on one timestep are received at the next timestep, and are tagged with their sender. Communication presents the main challenge of hidden role games: △-players wish to share information with teammates, but *not* with ▽-players.

In defining communication extensions, we must bound the *length* of the communication, that is, how many rounds of communication occur in between every move of the game, and how many distinct messages can be sent on each round. To do this, we fix a finite message space[5.4] of size M and length of communication R, and in our definition of equilibrium we will take a supremum over M and R. This will allow us to consider arbitrarily complex message spaces (*i.e.*, M and R arbitrarily large) while still only analyzing finite games: for any *fixed* M and R, the resulting game is a finite hidden-role game. We will show that our positive results (upper bounds) only require $\log M = R = \mathsf{polylog}(|H|, 1/\epsilon)$, where $|H|$ is the number of nodes (histories) in the game tree and ϵ is the desired precision of equilibrium.

We characterize optimal behavior in the hidden-role setting by converting hidden-role games into team games in a way that preserves the strategic aspect of hidden-roles. This team game is called *split-personality form* of a given hidden-role game. Given a (possibly communication-extended) hidden-role game Γ, we define and analyze two possible variants:

[5.2] For example, in *Mafia*, the villagers are △ while mafiosi are ▽.

[5.3] If players are assumed to be computationally bounded, pairwise private channels can be created from the public broadcast channels through public-key cryptography. However, throughout this chapter, for the sake of conceptual cleanliness, we will not assume that players are computationally bounded, and therefore we will distinguish the public-communication case from the private-communication case.

[5.4] Note that, if the message space is of size M, a message can be sent in $O(\log M)$ bits.

- the *uncoordinated split-personality form* USPLIT(Γ) is a team games with $2n$ players, derived by splitting each player i in the original game in two distinct players, i^+ and i^-, that pick actions for i in Γ if the player is assigned to team △ or ▽ respectively.
- the *coordinated split-personality form* CSPLIT(Γ) is the $(n+1)$-player team game in which the additional player, who we refer to simply as the *adversary* or ▽*-player*, takes control of the actions of all players who have been assigned to the ▽-team. On the contrary, the players from 1 to n control the players as usual only if they belong to the team △.

The coordinated split-personality variant encodes an extra assumption on ▽'s capabilities, namely, that the ▽-team is controlled by a single player and is therefore perfectly coordinated. Trivially, when only one player is on team ▽, the uncoordinated and coordinated split-personality forms coincide.

In either case, the resulting game is a team game in which each player has a fixed team assignment. We remark that the split-personality form maintains the strategic aspects of hidden roles, since i^+ and i^- share identity when interacting with the environment. For example, players may observe that i has done an action, but do not know if the controller was i^+ or i^-. Similarly, messages sent by i^+ and i^- are signed by player i, since the communication extension is applied on Γ *before* splitting personalities.

Picking which split-personality variant to use is a modeling assumption that depends on the game instance that one wants to address. For example, in many recreational tabletop games, USPLIT is the more reasonable choice because ▽-players are truly distinct; however, in a network security game where a single adversary controls the corrupted nodes, CSPLIT is the more reasonable choice. The choice of which variant also affects the complexity of equilibrium computation: as we will detail in later, CSPLIT yields a more tractable solution concept. In certain special cases, however, CSPLIT and USPLIT will coincide. For example, we will later show that this is the case in *Avalon*, which is key to allowing our algorithms to work in that game.

With these pieces in place, we define the *hidden-role equilibria* (HRE) of a hidden-role game Γ as the *team max-min equilibria* (TMEs) of the split-personality form of Γ. That is, the hidden-role equilibria are the optimal joint strategies for team △ in the split-personality game, where optimality is judged by the expected value against a jointly-best-responding ▽-team. The *value* of a hidden-role game is the expected value for △ in any hidden-role equilibrium. If communication (private or public) is allowed, we define hidden-role equilibria and values by taking the supremum over M and R of the expected value at the equilibrium, that is, the △-team is allowed to set the parameters of the communication.

Our new solution concept encodes, by design, a pessimistic assumption for the △-team. △ picks M, R and its strategy considering a worst-case ▽ adversary that knows this strategy and best-responds to it. Throughout our proofs, we will heavily make use of this fact. In particular, we will often consider ▽-players that "pretend to be △-players" under certain circumstances, which is only possible if ▽-players know △-players' strategies. It is *not* allowed for △-players to know ▽-players' strategies in the same fashion. This is in stark contrast to usual zero-sum game analysis, where various versions of the *minimax theorem* promise that the game is unchanged no matter which side commits first to a strategy. Indeed, we discuss in Section 5.6.2 the fact that, for hidden-role games, the asymmetry is in some sense necessary: a minimax theorem *cannot* hold for nontrivial hidden-role games. We argue, however, that this asymmetry is natural and *inherent* in the the hidden-role setting. If we assumed the contrary and inverted the order of the teams so that ▽ commits first to its strategy, △ could discover the roles immediately by agreeing to message a passphrase unknown to ▽ in the first round, thus spoiling the whole purpose of hidden-role games. This argument will be made formal in Section 5.6.2.

Existing solution concepts failures. We defined our equilibrium notion as a team max-min equilibrium (TME) of the split-personality form of a communication-extended hidden role game. Here, we will argue why some other notions would be less reasonable.

- *Nash Equilibrium.* A *Nash equilibrium* [230] is a strategy profile for all players from which no player can improve its own utility by deviating. This notion is unsuitable for our purposes because it fails to capture *team coordination*. In particular, in pure coordination games (in which all players have the same utility function), which are a special case of hidden-role games (with no hidden roles and no adversary team at all), a Nash equilibrium would only be locally optimal in the sense that no player can improve the team's value alone. In contrast, our notion will lead to the optimal team strategy in such games.
- *Team-correlated equilibrium.* The *team max-min equilibrium with correlation* [56, 292] (TMECor), is a common solution concept used in team games. It arises from allowing each team the ability to communicate in private (in particular, to generate correlated randomness) *before* the game begins. For team games, TMECor is arguably a more natural notion than TME, as the corresponding optimization problem is a bilinear saddle-point problem, and therefore in particular the minimax theorem applies, avoiding the issue of which team ought to commit first. However, for hidden-role games, TMECor is undesirable, because it does not make sense for a team to be able to correlate with teammates that have not even been assigned yet. The *team max-min equilibrium with communication* (TMECom) [56] makes an even stronger assumption about communication among team members, and therefore suffers the same problem.

5.1.2 Main Computational Contributions

We now introduce computational results, both positive and negative, for computing the hidden-role value and hidden-role equilibria of a given game.

Polynomial-time algorithm. Our main positive result is summarized in the following informal theorem statement.

Theorem 5.1 (Main result, informal; formal result in Theorem 5.12)**.** *If the number of players is constant, private communication is available, the ▽-team is a strict minority* (i.e., *strictly less than half of the players are on the ▽-team), and the adversary is coordinated, there is a polynomial-time algorithm for exactly computing the hidden-role value.*

This result should be surprising, for multiple reasons. First, team games are generally hard to solve, as we have seen, so any positive result for computing equilibria in team games is fairly surprising. Further, it is *a priori* not obvious that the value of any hidden-role game with private communication and coordinated adversary is even a *rational* number[5.5], much less computable in polynomial time: for example, there exist adversarial team games with no communication whose TME values are irrational [292].

There are two key ingredients to the proof of Theorem 5.1. The first is a special type of game which we call a *mediated* game. In a mediated game, there is a player, the *mediator*, who is always on team △. △-players can therefore communicate with it and treat it as a trusted party. We show that, when a mediator is present (and all the other assumptions of Theorem 5.1 also hold), the hidden-role value is computable in polynomial time. To do this, we state and prove a form of *revelation principle*. Informally, our revelation principle states that, without loss of generality, it suffices to consider △-team strategies in which, at every timestep of the game,

1. all △-players send their honest information to the mediator,
2. the mediator sends action recommendations to all players (regardless of their team allegiance; remember that the mediator may not know the team assignment), and
3. all △-players play their recommended actions.

▽-team players are, of course, free to pretend to be △-team players and thus send false information to the mediator; the mediator must deal with this possibility. However, ▽-team players cannot just send *any* message; they must send messages that *are consistent with some △-player*, lest they be immediately revealed as ▽-team. These observations are sufficient to construct a *two-player zero-sum game* Γ_0, where the mediator is the △-player and the coordinated adversary is the ▽-player. The value of Γ_0 is the value of the original hidden-role game, and the size of Γ_0 is at most polynomially larger than the size of the original game. Since two-player zero-sum extensive-form games can be solved in polynomial time, it follows that mediated hidden-role games can also be solved in polynomial time.

The second ingredient is to invoke results from the literature on *secure multi-party computation* to *simulate* a mediator in the case that one is not already present. A well-known result from that literature states that so long as strictly more than half of the players are honest, essentially any interactive protocol—such as the ones used by our mediator to interact with other players—can be simulated efficiently such that the adversary can cause failure of the protocol or leakage of information [21, 249].[5.6] Chaining such a protocol with the argument in the previous paragraph

[5.5] assuming all game values and chance probabilities are also rational numbers

[5.6] In this part of the argument, the details about the communication channels become important: in particular, the MPC results that we use for our main theorem statement assume that the network is synchronous (*i.e.*, messages sent in round r arrive in round $r + 1$), and that there are pairwise private channels and a public broadcast channel that are all authenticated (*i.e.*, message receivers know who sent the message). This is enough to implement MPC so long as $k < n/2$, where k is the number of adversaries and n is the number of players. Our results, however, do not depend on the specific assumptions about the communication channel, so long as said assumptions enable secure MPC with guaranteed outcome delivery. For a recent survey of MPC, see Lindell [198]. For example, if $k < n/3$ then MPC does not require a public broadcast channel, so neither do our results. For cleanliness, and to avoid introducing extra formalism, we will stick to one model of communication.

concludes the proof of the main theorem.

Related works on MPC and communication equilibria. The *communication equilibrium* [109, 228] is a notion of equilibrium with a mediator, in which the mediator has two-way communication with all players, and players need to be incentivized to report information honestly and follow recommendations. Communication equilibria include all Nash equilibria, and therefore are unfit for general hidden-role games for the same reason as Nash equilibria, as discussed in the previous subsection.

However, when team ▽ has only one player and private communication is allowed, the hidden-role equilibria coincide with the △-team-optimal communication equilibria in the original game Γ. Our main result covers this case, but an alternative way of computing a hidden-role equilibrium in this special case is to apply the optimal communication equilibrium algorithms of Chapter 6. However, those algorithms either involve solving linear programs, solving many zero-sum games, or solving zero-sum games with large reward ranges, which will be less efficient than directly solving a single zero-sum game Γ_0.

We are not the first to observe that multi-party computation can be used to implement a mediator for use in game theory. In various settings and for various solution concepts, it is known to be possible to implement a mediator using only cheap-talk communication among players [*e.g.*, [6, 156, 200, 287]]. For additional reading on the connections between game theory and cryptography, we refer the reader to the survey of Katz [171], and papers citing and cited by this survey.

Lower bounds. We also show *lower bounds* on the complexity of computing the hidden-role value, even for a constant number of players, when any of the assumptions in Theorem 5.1 is broken.

Theorem 5.2 (Lower bounds, informal; formal statement in Theorems 5.18, 5.19 and 5.21)**.** *If private communication is disallowed, the hidden-role value problem is* NP*-hard. If the* ▽*-team is uncoordinated, the problem is* coNP*-hard. If both, the problem is* Σ_2^{P}*-hard. All hardness reductions hold even when the* ▽*-team is a minority and the number of players is an absolute constant.*

Price of hidden roles. Finally, we define and compute the *price of hidden roles*. It is defined (analogously to the price of anarchy and price of stability, which are common quantities of study in game theory) as the ratio between the value of a hidden-role game, and the value of the same game with team assignments made public. We show the following:

Theorem 5.3 (Price of hidden roles; formal statement in Theorem 5.25)**.** *Let D be a distribution of team assignments. For the class of games where teams are drawn according to distribution D, the price of hidden roles is equal to $1/p$, where p is the probability of the most-likely team in D.*

Intuitively, in the worst case, the △-team can be forced to guess at the beginning of the game all the members of the △-team, and win if and only if its guess is correct. In particular, for the class of n-player games with k adversaries, the price of hidden roles is exactly $\binom{n}{k}$.

5.1.3 Experiments: *Avalon*

We ran experiments on the popular tabletop game *The Resistance: Avalon* (or simply *Avalon*, for short). As discussed earlier, despite the adversary team in *Avalon* not being coordinated in the sense used in the rest of the chapter, we show that, at least for the 5- and 6-player variants, the adversary would not benefit from being coordinated; therefore, our polynomial-time algorithms can be used to solve the game. This observation ensures that our main result applies. Game-specific simplifications allow us to reduce the game tree from roughly 10^{56} nodes [269] to 10^8 or even fewer, enabling us to compute exact equilibria. Our experimental evaluation demonstrates the practical efficacy of our techniques on real game instances. Our results are discussed in Section 5.7, and further detail on the game-specific reductions used, as well as a complete hand-analysis of a small *Avalon* variant, can be found in the appendix of the full paper [54].

5.1.4 Examples

In this section we present three examples that will hopefully help the reader in understanding our notion of equilibrium and justify some choices we have made in our definition.

A hidden-role matching pennies game. Consider a n-player version of matching pennies (with $n > 2$), which we denote as $\mathsf{MP}(n)$. One player is chosen at random to be the adversary (team ▽). All n players then simultaneously choose bits $b_i \in \{0, 1\}$. Team △ wins (gets utility 1) if and only if all n bits match; else, team △ loses (gets utility 0).

With no communication, the value of this game is $1/2^{n-1}$: it is an equilibrium for everyone to play uniformly at random. Public communication does not help, because, conditioned on the public transcript, bits chosen by players must be mutually independent. Thus, the adversary can do the following: pretend to be on team △, wait for all communication to finish, and then play 0 if the string of all ones is more conditionally likely than the string of all 1s, and vice-versa.

With private communication, however, the value becomes $1/(n + 1)$. Intuitively, the △-team should attempt to guess who the ▽-player is, and then privately discuss among the remaining $n - 1$ players what bit to play. We defer formal proofs of the above game values to Section 5.5, because they rely on results in Section 5.4.1.

Simultaneous actions. In typical formulations of extensive-form imperfect-information games, it is without loss of generality to assume that games are *turn-based*, *i.e.*, only one player acts at any given time. To simulate simultaneous actions with sequential ones, one can simply forbid players from observing each others' actions. However, when communication is allowed arbitrarily throughout the game, the distinction between simultaneous and sequential actions suddenly becomes relevant, because *players can communicate when one—but not all—the players have decided on an action*.

To illustrate this, consider the game MP(n) defined in the previous section, with public communication, except that the players act in sequence in order of index $(1, 2, \ldots, n)$. We claim that the value of this game is not $1/2^{n-1}$, but at least $1/2n$. To see this, consider the following strategy for team △. The △ players wait for P1 to (privately) pick an action. Then, P2 publicly declares a bit $b \in \{0, 1\}$, and all remaining players play b if they are on team △. If P1 was the ▽ player, this strategy wins with probability at least 1/2, so the expected value is at least $1/2n$. This example illustrates the importance of allowing simultaneous actions in our game formulations.

Correlated randomness matters. We use our third and final example to discuss a nontrivial consequence of the definition of hidden-role equilibrium that may appear strange at first: it is possible for seemingly-useless information to affect strategic decisions and the game value.

To illustrate, consider the following simple game Γ: there are three players, and three role cards. Two of the three cards are marked △, and the third is marked ▽. The cards are dealt privately and randomly to the players. Then, after some communication, all three players simultaneously cast votes to elect a winner. If no player gains a majority of votes, ▽ wins. Otherwise, the elected winner's team wins. Clearly, △ can win no more than 2/3 of the time in this game: ▽ can simply pretend to be on team △, and in that case △ cannot gain information, and the best they can do is elect a random winner.

Now consider the following seemingly-meaningless modification to the game. We will modify the two △ cards so that they are distinguishable. For example, one card has △ written on it, and the other has △′. We argue that this, perhaps surprisingly, affects the value of the game. In fact, the △ team can now win deterministically, even with only public communication. Indeed, consider following strategy. The two players on △ publicly declare what is written on their cards (*i.e.*, △ or △′). The player elected now depends on what the third player did. If one player does not declare △ or △′, elect either of the other two players. If two players declared △, elect the player who declared △′. If two players declare △′, elect the player who declared △. This strategy guarantees a win: no matter what the ▽-player does, any player who makes a unique declaration is guaranteed to be on the △-team.

What happens in the above example is that making the cards distinguishable introduces a piece of *correlated randomness* that △ can use: the two △ players receive cards whose labels are (perfectly negatively) correlated with each other. Since our definition otherwise prohibits the use of such correlated randomness (because players cannot communicate only with players on a specific team), introducing some into the game can have unintuitive effects. In Section 5.6.2, we expand on the effects of allowing correlated randomness: in particular, we argue that allowing correlated randomness essentially ruins the point of hidden-role games by allowing the △ team to learn the entire team assignment.

5.2 Preliminaries

Our notation in this part differs slightly from other parts of this thesis. For reasons alluded to in the introduction, we explicitly allow simultaneous moves in our formulation. More specifically, at each history $h \in \mathcal{H} \setminus \mathcal{Z}$, *every* player (including chance) selects an action $a \in \mathcal{A}_i(h)$, and the edges leaving h are identified with *joint actions* $a \in \bigtimes_{i \in [n] \cup \mathsf{C}} A_i(h)$. Thus, each player's infoset partition $\mathcal{I}$ is a partition of $\mathcal{H} \setminus \mathcal{Z}$.

An extensive-form game is an *adversarial team game* (ATG) if there is a team assignment $t \in \{\triangle, \triangledown\}^n$ and a team utility function $u : \mathcal{Z} \to \mathbb{R}$ such that $u_i(z) = u(z)$ if $t_i = \triangle$, and $u_i(z) = -u(z)$ if $t_i = \triangledown$. That is, each player is assigned to a team, all members of the team get the same utility, and the two teams are playing an adversarial zero-sum game[5.7]. In this setting, we will write $\boldsymbol{x}_i \in \mathcal{X}_i$ and $\boldsymbol{y}_j \in \mathcal{Y}_j$ for a generic behavioral strategy of a player on team $\triangle$ and $\triangledown$ respectively. ATGs are fairly well studied. In particular, team maxmin equilibria (TMEs) [56, 292] and their variants are the common notion of equilibrium employed. The *value* of a given strategy profile $\boldsymbol{x}$ for team $\triangle$ is the value that $\boldsymbol{x}$ achieves against a best-responding opponent team. The *TME value* is the value of the best strategy profile of team $\triangle$. That is, the TME value is defined as

$$\mathsf{TMEVal}(\Gamma) := \max_{\boldsymbol{x} \in \bigtimes_i \mathcal{X}_i} \min_{\boldsymbol{y} \in \bigtimes_j \mathcal{Y}_j} u(\boldsymbol{x}, \boldsymbol{y}), \tag{5.1}$$

and the *TMEs* are the strategy profiles $\boldsymbol{x}$ that achieve the maximum value. Notice that the TME problem is nonconvex, since the objective function u is nonlinear as a function of $\boldsymbol{x}$ and $\boldsymbol{y}$. As such, the minimax theorem does not apply, and swapping the teams may not preserve the solution. Computing an (approximate) TME is Σ_2^{P}-complete in extensive-form games (Theorems 4.33 and 4.34).

5.3 Equilibrium Concepts for Hidden-Role Games

While the notion of TME is well-suited for ATGs, it is not immediately clear how to generalize it to the setting of hidden-role games. We do so by formally defining the concepts of *hidden-role game*, *communication* and *split-personality form* first introduced in Section 5.1.1.

Definition 5.4. An extensive-form game is a *zero-sum hidden-role team game*, or *hidden-role game* for short, if it satisfies the following additional properties:

1. At the root node, only chance has a nontrivial action set. Chance chooses a string $t \sim \mathcal{D} \in \Delta(\{\triangle, \triangledown\}^n)$, where t_i denotes the team to which player i has been assigned. Each player i privately observes (at least[5.8]) their team assignment t_i. In addition, $\triangledown$-players privately observe the entire team assignment t.

[5.7] This is a slight abuse of language: if the $\triangle$ and $\triangledown$ teams have different sizes, the sum of all players' utilities is not zero. However, such a game can be made zero-sum by properly scaling each player's utility. The fact that such a rescaling operation does not affect optimal strategies is a basic result for von Neumann–Morgenstern utilities [211, Chapter 2.4]. We will therefore generally ignore this detail.

[5.8] It is allowable for $\triangle$-players to also have more observability of the team assignment, *e.g.*, certain $\triangle$-players may know who some $\triangledown$-players are.

2. The utility of a player i is defined completely by its team: there is a $u : \mathcal{Z} \to \mathbb{R}$ for which $u_i(z) = u(z)$ if player i is on team △ at node z, and $-u(z)$ otherwise.[5.9]

In some games, players observe additional information beyond just their team assignments. For example, in *Avalon*, one △-player is designated *Merlin*, and Merlin has additional information compared to other △-players. In such cases, we will distinguish between the *team assignment* and *role* of a player: the team assignment is just the team that the player is on (△ or ▽), while the role encodes the extra private information of the player as well, which may affect what actions that player is allowed to legally take. For example, the team assignment of the player with role *Merlin* is △. We remark that additional imperfect information of the game may be observed after the root node.[5.10]

We will use k to denote the largest number of players on the ▽-team, that is, $k = \max_{t \in \text{supp}(\mathcal{D})} |\{i : t_i = \triangledown\}|$.

5.3.1 Models of Communication

The bulk of this chapter concerns notions of equilibrium that allow communication between the players. We distinguish between *public* and *private communication*:

1. *Public communication*: There is an open broadcast channel on which all players can send messages.
2. *Private communication*: In addition to the open broadcast channel, each pair of players has access to a private communication channel. The private communication channel reveals to all players when messages are sent, but only reveals the message contents to the intended recipients.

Assuming that public-key cryptography is possible (*e.g.*, assuming the discrete logarithm problem is hard) and players are polynomially computationally bounded, public communication and private communication are equivalent, because players can set up pairwise private channels via public-key exchange. However, in this chapter, we assume that agents are computationally unbounded and thus treat the public and private communication cases as different. Our motivation for making this distinction is twofold. First, it is conceptually cleaner to explicitly model private communication, because then our equilibrium notion definitions do not need to reference computational complexity. Second, perhaps counterintuitively, equilibria with public communication only may be *more* realistic to execute in practice in human play, precisely *because* public-key cryptography breaks. That is, the computationally unbounded adversary renders more "complex" strategies of the △-team (involving key exchanges) useless, thus perhaps resulting in a *simpler* strategy. We emphasize that, in all of our positive results, the △-team's strategy *is* efficiently computable.

[5.9] While at a first look this condition is similar to the one in ATGs, we remark that in this case the number of players in a team depends on the roles assigned at the start. The same considerations as Footnote 5.7 on the zero-sum rescaling of the utilities hold.

[5.10] This is an important difference with respect to Bayesian games [142], which assume all imperfect information to be the initial *types* of the players. Conversely, we have an imperfect information structure that evolves throughout the game, while only the teams are assigned and observed at the start.

To formalize these notions of communication, we now introduce the *communication extension*.

Definition 5.5. The *public* and *private* (M, R)*-communication extensions* corresponding to a hidden-role game Γ are defined as follows. Informally, between every step of the original game Γ, there will be R rounds of communication; in each round, players can send a public broadcast message and private messages to each player. The communication extension starts in state $h = \varnothing \in \mathcal{H}_\Gamma$. At each game step of Γ:

1. Each player $i \in [n]$ observes its infoset $I_i \ni h$.
2. For each of R successive communication rounds:
   (a) Each player i simultaneously chooses a message $m_i \in [M]$ to broadcast publicly.

   (b) If private communication is allowed, each player i also chooses messages $m_{i\to j} \in [M] \cup \{\perp\}$ to send to each player $j \neq i$. $\perp$ denotes that the player does not send a private message at that time.

   (c) Each player j observes the messages $m_{i\to j}$ that were sent to it, as well as all messages m_i that were sent publicly. That is, by notion of communication, the players observe:
      - *Public*: player j observes the ordered tuple $(m_1, \dots, m_n)$.
      - *Private*: player j also observes the ordered tuple $(m_{1\to j}, \dots, m_{n\to j})$, and the set $\{(i, k) : m_{i\to k} \neq \perp\}$. That is, players observe messages sent to them, and players see when other players send private messages to each other (but not the contents of those messages)
3. Each player, including chance, simultaneously plays an action $a_i \in \mathcal{A}_i(h)$. (Chance plays according to its fixed strategy.) The game state h advances accordingly.

We denote the (M, R)-extensions as $\textsc{Comm}^{M,R}_{\mathsf{priv}}(\Gamma)$, and $\textsc{Comm}^{M,R}_{\mathsf{pub}}(\Gamma)$. To unify notation, we also define $\textsc{Comm}^{M,R}_{\mathsf{none}}(\Gamma) = \Gamma$. When the type of communication allowed and number of rounds are not relevant, we use $\textsc{Comm}(\Gamma)$ to refer to a generic extension.

5.3.2 Split Personalities

We introduce two different *split-personality* forms $\text{US}\textsc{plit}(\Gamma)$ and $\text{CS}\textsc{plit}(\Gamma)$ of a hidden-role game Γ, The split-personality forms are adversarial team games which preserve the characteristics of Γ.

Definition 5.6. The *uncoordinated split-personality form*[5.11] of an n-player hidden-role game Γ is the $2n$-player adversarial team game $\text{US}\textsc{plit}(\Gamma)$ in which each player i is split into two players, i^+ and i^-, which control player i's actions when i is on team △ and team ▽ respectively.

Unlike the original hidden-role game Γ, the split-personality game is an adversarial team game without hidden roles: players i^+ are on the △ team, and i^- are on the ▽-team. Therefore, we are

[5.11] In the language of Bayesian games, the split-personality form would almost correspond to the *agent form*.

able to apply notions of equilibrium for ATGs to $\textsc{USplit}(\Gamma)$. We also define the *coordinated split-personality form*:

Definition 5.7. The *coordinated split-personality form* of an n-player hidden-role game Γ is the $(n+1)$-player adversarial team game $\textsc{CSplit}(\Gamma)$ formed by starting with $\textsc{USplit}(\Gamma)$ and merging all ▽-players into a single adversary player, who observes all their observations and chooses all their actions.

Assuming ▽ to be *coordinated* is a worst-case assumption for team △, which however can be justified. In many common hidden-role games, such as the *Mafia* or *Werewolf* family of games and most variants of *Avalon*, such an assumption is not problematic, because the ▽-team has essentially perfect information already. In the appendix of the full paper [54], we justify why this assumption is safe also in some more complex *Avalon* instances considered. The coordinated split-personality form will be substantially easier to analyze, and in light of the above equivalence for games like *Avalon*, we believe that it is important to study it.

When team ▽ in Γ is already coordinated, that is, if every ▽-team member has the same observation at every timestep, the coordinated and uncoordinated split-personality games will, for all our purposes, coincide: in this case, any strategy of the adversary in $\textsc{CSplit}(\Gamma)$ can be matched by a joint strategy of the ▽-team members in $\textsc{USplit}(\Gamma)$. This is true in particular if there is only one ▽-team member. But, we insert here a warning: even when the base game Γ has a coordinated adversary team, the private communication extension $\textsc{Comm}_{\mathsf{priv}}(\Gamma)$ will not. Thus, with private-communication extensions of Γ, we must distinguish the coordinated and uncoordinated split-personality games even if Γ itself is coordinated.

5.3.3 Equilibrium Notions

We now define the notions of equilibrium that we will primarily study.

Definition 5.8. The *uncoordinated value* of a hidden-role game Γ with notion of communication c is defined as

$$\mathsf{UVal}_c(\Gamma) := \sup_{M,R} \mathsf{UVal}_c^{M,R}(\Gamma)$$

where $\mathsf{UVal}_c^{M,R}(\Gamma)$ is the TME value of $\textsc{USplit}(\textsc{Comm}_c^{M,R}(\Gamma))$. The *coordinated value* $\mathsf{CVal}_c(\Gamma)$ is defined analogously by using $\textsc{CSplit}$.

Definition 5.9. An *ϵ-uncoordinated hidden-role equilibrium* of Γ with a particular notion of communication $c \in \{\mathsf{none}, \mathsf{pub}, \mathsf{priv}\}$ is a tuple $(M, R, \boldsymbol{x})$ where $\boldsymbol{x}$ is a △-strategy profile in $\textsc{USplit}(\textsc{Comm}_c^{M,R}(\Gamma))$ of value at least $\mathsf{UVal}_c(\Gamma) - \epsilon$. The *$\epsilon$-coordinated hidden-role equilibria* is defined analogously, again with $\textsc{CSplit}$ and CVal instead of $\textsc{USplit}$ and UVal.

As discussed in Section 5.1.1, our notion of equilibrium is inherently asymmetric due to its max-min definition. The △-team is the first to commit to a strategy and a communication scheme, and ▽ is allowed to know both how much communication will be used (*i.e.*, M and R) as well as △'s entire strategy x. As mentioned before, this asymmetry is fundamental in our setting, and we will formalize it in Section 5.6.2.

5.4 Computing Hidden-Role Equilibria

In this section, we show the main computational results regarding the complexity of computing an hidden-role equilibrium in different settings. We first provide positive results for the private-communication case in Section 5.4.1 while the negative computational results for the no/public-communications cases are presented in Section 5.4.3. The results are summarized in Table 5.1.

5.4.1 Computing Private-Communication Equilibria

In this section, we show that it is possible under some assumptions to compute equilibria efficiently for hidden-role games. In particular, in this section, we assume that

1. there is private communication,
2. the adversary is coordinated, and
3. the adversary is a minority ($k < n/2$).

Games with a publicly-known △-player. First, we consider a special class of hidden-role games which we call *mediated*. In a mediated game, there is a player, who we call the *mediator*, who is always assigned to team △. The task of the mediator is to coordinate the actions and information transfer of team △. Our main result of this subsection is the following:

Theorem 5.10 (Revelation Principle). *Let Γ^* be a mediated hidden-role game. Then, for $R \geq 2$ and $M \geq |H|$, there exists a coordinated private-communication equilibrium in which the players on △ have a TME profile in which, at every step, the following events happen in sequence:*

1. *every player on team △ sends its observation privately to the mediator,*
2. *the mediator sends to every player (△ and ▽) a recommended action, and*
3. *all players on team △ play their recommended actions.*

Proof. We follow the usual proof structure of revelation principle proofs. Let $x = (x_1, \dots, x_n)$ be any strategy profile for team △ in $\textsc{CSplit}(\textsc{Comm}_{\mathsf{priv}}(\Gamma^*))$. Consider the strategy profile $x' = (x'_1, \dots, x'_n)$ that operates as follows. For each player i, the mediator instantiates a simulated version of each player i playing according to strategy x_i. These simulated players are entirely "within the imagination of the mediator".

1. When a (real) player i sends an observation $o_i(h)$ to the mediator, the mediator forwards observation $o_i(h)$ to the simulated player i.
2. When a simulated player i wants to send a message to another player j, the mediator forwards the message to the *simulated* player j.
3. When a simulated player i plays an action a_i, the mediator forwards the action as a

message to the real player i.

Since the strategy x_i is only well-defined on sequences that can actually arise in $\textsc{CSplit}(\textsc{Comm}_{\mathsf{priv}}(\Gamma^*))$, the simulated player i may crash if it receives an impossible sequence of observations. If player i's simulator has crashed, it will no longer send simulated messages, and the mediator will no longer send messages to player i.

It suffices to show that team ▽ cannot exploit x' more than x. Let y' be any best-response strategy profile for ▽ against x'.

We will show that there exists a strategy y such that (x', y') is equivalent to (x, y). Consider the strategy y for team ▽ in which each player i maintains simulators of both x_i and y'_i, and acts as follows.

1. Upon receiving an observation or message, forward it to y'_i
2. If y'_i wants to send an observation to the mediator, forward that observation to x_i.
3. If x_i sends a message, send that message.
4. If x_i plays an action a_i, forward that action to y'_i as a message from the mediator. If x_i crashes, send empty messages to y'_i from the mediator. In either case, when y'_i outputs an action, play that action.

By definition, the profiles (x, y) and (x', y') have the same expected utility (in fact, they are equivalent, in the sense that they induce the same outcome distribution over the terminal nodes of Γ), so we are done. □

Players on team ▽, of course, can (and will) lie or deviate from recommendations as they wish. The above revelation principle imples the following algorithmic result:

Theorem 5.11. *Let Γ^* be a mediated hidden-role game, $R \geq 2$, and $M \geq |H|$. An (exact) coordinated private-communication hidden-role equilibrium of Γ^* can be computed by solving an extensive-form zero-sum game Γ_0 with at most $|H|^{k+1}$ nodes, where H is the history set of Γ^*.*

Proof. Consider the zero-sum game Γ_0 that works as follows. There are two players: the *mediator*, representing team △, and a single *adversary*, representing team ▽. The game state of Γ_0 consists of a history h in Γ (initially the root), and n *sequences* s_i. Define a sequence s_i to be *consistent* if it is a prefix of a terminal sequence $s_i(z)$ for some terminal node z of Γ. For each sequence s_i which ends with an action, let $O(s_i)$ be the set of observations that could be the next observations of player i in Γ.

1. For each player i on team △, let $\tilde{o}_i = o_i(h)$ be the true observation of player i. The adversary observes all recommendations $o_i(h)$ for players i on its team. Then, for each such player, the adversary picks an observation $\tilde{o}_i \in O(s_i) \cup \{\perp\}$. Each $\tilde{o}_i$ is appended to the corresponding s_i.

2. The mediator observes $(\tilde{o}_1, \dots, \tilde{o}_n)$, and picks action recommendations $a_i \in A_i(\tilde{o}_i)$ to recommend to each player i, and appends each a_i to the corresponding s_i.
3. Players on team △ automatically play their recommended actions. The adversary observes all actions $(a_i : t_i = \triangledown)$ recommended to players on team ▽, and selects the action played by each member of team ▽.

The size of this game is given by the number of tuples of the form $(h, s_1, \dots, s_k)$ where s_i is the sequence of adversary i and h is a node of the original game Γ. There are at most $|H|^{k+1}$ of these, so we are done. □

We give a sketch of how the two-player zero-sum game is structured. Theorem 5.10 allows us to simplify the game by fixing the actions of all players on team △, leaving two strategic players, the mediator and the adversary. Any node from the original game is expanded into three levels:

1. the adversary picks messages on behalf of all ▽-players to send to the mediator,
2. the mediator picks recommended actions to send to all players, and
3. the adversary acts on behalf of all ▽-players.

The key to proving Theorem 5.11 is that, in the first step above, the adversary's message space is not too large. Indeed, any message sent by the adversary must be a message that *could have plausibly been sent by a △-player*: otherwise the mediator could automatically infer that the sender must be the adversary. It is therefore possible to exclude all other messages from the game since they belong to dominated strategies. Carefully counting the number of such messages would complete the proof.

It is crucial in the above argument that the ▽-team is coordinated; indeed, otherwise, it would not be valid to model the ▽-team as a single adversary in Γ_0. It turns out (Theorem 5.19) that, in fact, without the coordination assumption, the problem is hard.

In practice, zero-sum extensive-form games can be solved very efficiently in the tabular setting with linear programming, or algorithms from the CFR family. Thus, Theorem 5.11 gives an efficient algorithm for solving hidden-role games with a mediator.

Simulating mediators with multi-party computation. In this section, we show that the previous result essentially generalizes (up to exponentially-small error) to games *without* a mediator, so long as the ▽ team is also a minority, that is, $k < n/2$. Informally, the main result of this subsection states that, when private communication is allowed, one can efficiently *simulate* the existence of a mediator using secure multi-party computation (MPC), and therefore team △ can achieve the same value. The form of secure MPC that we use is *information-theoretically* secure; that is, it is secure even against computationally-unbounded adversaries.

Theorem 5.12 (Main theorem)**.** *Let Γ be a hidden-role game with $k < n/2$. Then* $\mathsf{CVal}_{\mathsf{priv}}(\Gamma) = \mathsf{CVal}_{\mathsf{priv}}(\Gamma^*)$*, where Γ^* is Γ with a mediator added, and moreover this value can be computed in $|H|^{O(k)}$ time by solving a zero-sum game of that size. Moreover, an ϵ-hidden-role equilibrium with private communication and* $\log M = R = \mathsf{polylog}(|H|, 1/\epsilon)$ *can be computed and executed by the △-players in time* $\mathsf{poly}(|H|^k, \log(1/\epsilon))$.

The proof uses MPC to simulate the mediator and then executes the equilibrium given by Theorem 5.11. The proof of Theorem 5.12, as well as requisite background on multi-party computation, are deferred to the next subsection. We emphasize that Theorems 5.11 and 5.12 are useful not only for algorithmically computing an equilibrium, but also for manual analysis of games: instead of analyzing the infinite design space of possible messaging protocols, it suffices to analyze the finite zero-sum game Γ_0. Our experiments on *Avalon* use both manual analysis and computational equilibrium finding algorithms to solve instances.

Comparison with communication equilibria. As mentioned in Section 5.1.2, our construction simulating a mediator bears resemblance to the construction used to define *communication equilibria* [109, 228]. At a high level, a communication equilibrium of a game Γ is a Nash equilibrium of Γ augmented with a mediator that is playing according to some fixed strategy μ. Indeed, when team ▽ has only one player, it turns out that the two notions coincide:

Theorem 5.13. *Let Γ be a hidden-role game with $k = 1$. Then* $\mathsf{CVal}_{\mathsf{priv}}(\Gamma)$ *is exactly the value for △ of the △-optimal communication equilibrium of* Γ.

However, in the more general case where ▽ can have more than one player, Theorem 5.13 does not apply: in that case, communication equilibria include all Nash equilibria in particular, and therefore fail to enforce *joint* optimality of the ▽-team, so our concepts and methods are more suitable.

Before proving this result, we must first formally define a communication equilibrium. Given an arbitrary game Γ with n players, consider the $(n + 1)$-player game in which the extra player is the mediator. Consider a private-communication extension $\tilde{\Gamma}$ of that game, with $R = 2$ rounds and $M = |H|$, in which only communication with the mediator is allowed. In Γ^*, each player has a *direct strategy* x_i^* in which, at every timestep, the player sends its honest information to the mediator, interprets the mediator's message in reply as an action recommendation, and plays that action recommendation.

Definition 5.14. A *communication equilibrium* of Γ is a strategy profile $\mu = (x_1, \dots, x_n)$ for the mediator of $\tilde{\Gamma}$ such that, with μ held fixed, the profile $(x_1, \dots, x_n)$ is a Nash equilibrium of the resulting n-player game.

The *revelation principle for communication equilibria* [109, 228] states that, without loss of generality in the above definition, it can be assumed that $x = x^*$, *i.e.*, players are direct in equilibrium.

We now prove the theorem.

Proof of Theorem 5.13. Let Γ^* be Γ with a mediator who is always on team △, and let Γ_0 be the zero-sum game constructed by Theorem 5.11. Because $\mathsf{CVal}_{\mathsf{priv}}(\Gamma)$ is the zero-sum value of Γ_0 by Theorem 5.12, it is enough to prove the inequality chain

$$\mathsf{CVal}_{\mathsf{priv}}(\Gamma) \le \mathsf{CommVal}(\Gamma) \le \mathsf{Val}(\Gamma_0),$$

where $\mathsf{CommVal}$ is the value of the △-optimal communication equilibrium and Val is the zero-sum game value.

By Theorem 5.12, the hidden-role equilibria of Γ are (up to an arbitrarily small error $\epsilon > 0$) TMEs of $\textsc{CSplit}(\Gamma^*)$. By von Stengel and Koller [292], since ▽ has only one player, the TMEs of $\textsc{CSplit}(\Gamma^*)$ are precisely the △-team-optimal Nash equilibria of $\textsc{CSplit}(\Gamma^*)$. Thus, for a TME $(\mu, x_1, \dots, x_n)$ of that game (where μ is a strategy of the mediator), there exists an adversary strategy y such that $(\mu, x_1, \dots, x_n, y)$ is a Nash equilibrium. But then $(\mu, x_1, \dots, x_n, y)$ is also a communication equilibrium. Thus $\mathsf{TMEVal}(\textsc{CSplit}(\Gamma^*)) \le \mathsf{CommVal}(\Gamma)$.

We now show the second inequality. If Γ is an adversarial hidden-role game, each player i's strategy can be expressed as a tuple (x_i, y_i) where x_i is player i's strategy in the subtree where i is on team △, and y_i is the same on team ▽. In that case, the problem of finding a △-optimal communication equilibrium can be expressed as:

$$\begin{aligned} \max_{\mu} \quad & u(\mu, x^*, y^*) \\ \text{s.t.} \quad & \forall i \quad \max_{x_i} u(\mu, x_i, x^*_{-i}, y^*) \le u(\mu, x^*, y^*) \\ & \forall i \quad \min_{y_i} u(\mu, x^*, y_i, y^*_{-i}) \ge u(\mu, x^*, y^*) \end{aligned}$$

where u is the △-team utility function. That is, no player can increase their team's utility by deviating from the direct profile (x^*, y^*), regardless of which team they are assigned to. Now, using the fact that ▽ has only one player, we can write △'s utility function u as a sum $u = \sum_i u_i$ where u_i is the utility of △ when the ▽-player is player i (weighted by the probability of that happening). Each term u_i depends only on x and the y_is. Thus, the above program can be rewritten as

$$\begin{aligned} \max_{\mu} \quad & \sum_i u_i(\mu, x^*, y^*_i) \\ \text{s.t.} \quad & \forall i \quad \max_{x_i} u(\mu, x_i, x^*_{-i}, y^*) \le u(\mu, x^*, y^*) \\ & \forall i \quad \min_{y_i} u_i(\mu, x^*, y_i) \ge u_i(\mu, x^*, y^*_i) \end{aligned}$$

or, equivalently,

$$\begin{aligned} \max_{\mu} \quad & \sum_i \min_{y_i} u_i(\mu, x^*, y_i) \\ \text{s.t.} \quad & \forall i \quad \max_{x_i} u(\mu, x_i, x^*_{-i}, y^*) \le u(\mu, x^*, y^*). \end{aligned}$$

or, equivalently,

$$\begin{aligned} \max_{\mu} \min_{y} \quad & u(\mu, x^*, y) \\ \text{s.t.} \quad & \forall i \;\; \max_{x_i} u(\mu, x_i, x^*_{-i}, y^*) \leq u(\mu, x^*, y^*). \end{aligned}$$

This is precisely the problem of computing a zero-sum equilibrium in the game Γ_0, except with an extra constraint, so the inequality follows. □

5.4.2 Multi-Party Computation and Proof of Theorem 5.12

We first formalize the usual setting of multi-party computation (MPC). Let X be the set of binary strings of length ℓ, and let λ be a security parameter. Rabin and Ben-Or [249] claims in their Theorem 4 that essentially any protocol involving a mediator can be efficiently simulated without a mediator so long as more than half the players follow the protocol and we allow some exponentially small error. However, they do not include a proof of this result. In the interest of completeness, we prove the version of their result that is needed for our setting, based only on the primitives of *secure multi-party computation* and *verifiable secret sharing*.

5.4.2.1 Secure MPC

In *secure MPC*, there is a (possibly randomized) function $f : X^n \to X^n$ defined by a circuit with N nodes. A subset $K \subset [n]$ of size $< n/2$ has been *corrupted*. Each *honest* player $i \in [n] \setminus K$ holds an input $x_i \in X$. The goal is to design a randomized messaging protocol, with *private* communication, such that, regardless of what the corrupted players do, there exist inputs $\{x_j : j \in K\}$ such that:

1. (Output delivery) At the end of the protocol, each player i learns its own output, $y_i := f(x_1, \dots, x_n)_i$, with probability $1 - 2^{-\lambda}$.
2. (Privacy) No subset of $< n/2$ players can learn any information except their own output y_i. That is, the players cannot infer any extra information from analyzing the transcripts of the message protocol than what they already know.

   This can be intuitively modeled as follows. If any minority of colluded players were to analyze the empirical distributions of transcripts of the protocol for any input-output tuple, then such distribution would be fully explained in terms of their input-outputs, leaving no conditional dependence on other players' input-outputs. Formally, for any such subset $K \subseteq [n]$ of size $< n/2$, there exists a randomized algorithm Sim_K that takes the inputs and outputs of the players in set K, and reconstructs transcripts T, such that for all $x \in X^n$, we have

   $$\sum_T |\Pr[\mathsf{Sim}_K(x_K, y_K) = T] - \Pr[\mathsf{View}_K(x) = T]| \leq 2^{-\lambda}$$

   where $\mathsf{View}_K(x)$ is the distribution of transcripts observed by players in set K when running the protocol with input x.

In other words, the players in set K cannot do anything except pass to f inputs of their choice. For our specific application, this implies that introducing an MPC protocol to simulate the mediator is equivalent to having a mediator, because no extra information aside from the intended function will be leaked to the players.

Theorem 5.15 ([21, 249]). *Secure MPC is possible, with polytime (in ℓ, λ, n, N) algorithms that take at most polynomially many rounds and send at most polynomially many bits in each round.*

5.4.2.2 Verifiable Secret Sharing

In the *verifiable secret sharing* (VSS) problem, the goal is to design a function $\mathsf{Share} : X \to \bar{X}^n$, where $\bar{X} = \{0, 1\}^{\mathsf{poly}(\ell,\lambda,n)}$, that *shares* a secret $x \in X$ by privately informing each player i of its piece $\mathsf{Share}(x)_i$, such that:

1. (Reconstructibility) Any subset of $> n/2$ players can recover the secret fully, even if the remaining players are adversarial. That is, there exists a function $\mathsf{Reconstruct} : X^n \to X$ such that, where $(x_1, \dots, x_n) \leftarrow \mathsf{Share}(x)$, we have $\mathsf{Reconstruct}(x'_1, \dots, x'_n) = x$ so long as $x'_i = x_i$ for $> n/2$ players i.
2. (Privacy) No subset of $< n/2$ players can learn any information about the secret x. That is, for any such subset K and any secret x, there exists a distribution $\mathsf{Sim}_K \in \Delta(X^n)$ such that

$$\|\mathsf{Share}(x)_K - \mathsf{Sim}_K\|_1 \le 2^{-\lambda},$$

where $\|\cdot\|_1$ denotes the ℓ_1-norm on probability distributions.

Theorem 5.16 ([248, 249]). *There exist algorithms* Share *and* $\mathsf{Reconstruct}$*, with runtime* $\mathsf{poly}(\ell, \lambda, n)$*, that implement robust secret sharing.*

VSS is a primitive used in a fundamental way to build secure MPC protocols; to be formally precise, we will require VSS as a separate primitive as well to maintain the state of the mediator throughout the game.

5.4.2.3 Simulating a Mediator

We will assume that the game Γ_0 in Theorem 5.11 has been solved, and that its solution is given by a (possibly randomized) function

$$f : \Sigma \times O^n \to \Sigma \times A^n \tag{5.2}$$

where Σ is the set of information states of the mediator in Γ_0. At each step, the mediator takes n observations $o_1, \dots, o_n$ and its current infostate s as input, and updates its infostate and outputs action recommendations according to the function f.

Consider the function $\hat{f} : (\bar{X} \times O)^n \to (\bar{X} \times A)^n$ defined by

$$\hat{f}((x_1, o_1), \ldots, (x_n, o_n)) = ((x'_1, a_1), \ldots, (x'_n, a_n))$$

where

$$(s', a_1, \ldots, a_n) = f(\mathsf{Reconstruct}(x_1, \ldots, x_n), o_1, \ldots, o_n)$$

$$(x'_1, \ldots, x'_n) = \mathsf{Share}(s').$$

That is, $\hat{f}$ operates the mediator with its state secret-shared across the various players. The players will run secure MPC on $\hat{f}$ at every timestep. By the properties of MPC and secret sharing, this securely implements the mediator in such a way that the players on team ▽ can neither break privacy nor cause the protocol to fail, with probability better than $O(|H| \cdot 2^{-\lambda})$, where H is the set of histories of the game.

We have thus shown our main theorem, which is more formally stated as follows:

Theorem 5.17 (Formal version of Theorem 5.12)**.** *Let:*

- Γ *be a hidden-role game with a* ▽*-team of size* $k < n/2$*,*
- Γ^* *be identical to* Γ *except that there is an additional player who takes no nontrivial actions but is always on team* △*;*
- Γ_0 *be the zero-sum game defined by Theorem 5.11 based on* Γ^**;*
- x *be any strategy of the* △ *player (mediator) in* Γ_0*, represented by an arithmetic circuit* f *as in* (5.2) *with* $N = \mathsf{poly}(|H|^k)$ *gates; and*
- λ *be a security parameter.*

Then there exists a strategy profile x' *of the* △ *players in* $\mathrm{CS}\textsc{plit}(\mathrm{C}\textsc{omm}^{M,R}_{\mathsf{priv}}(\Gamma))$*, where* $\log M = R = \mathsf{poly}(\lambda, N, \log |H|)$ *such that:*

1. (Equivalence of value) *the value of* x' *is within* $2^{-\lambda}$ *of the value of* x *in* Γ_0*, and*
2. (Efficient execution) *there is a* $\mathsf{poly}(r)$*-time randomized algorithm* $\mathcal{A}_\Gamma$ *that takes as input an infostate* s_i *of* $\mathrm{CS}\textsc{plit}(\mathrm{C}\textsc{omm}^{M,R}_{\mathsf{priv}}(\Gamma))$ *that ends with an observation, and returns the (possibly random) action that player* i *should play at* s_i*.*

5.4.3 Computing No/Public-Communication Equilibria

In this section, we consider games with no communication or with public-communication and a coordinated minority. Conversely to the private-communication case of Section 5.4.1, in this case the problem of computing the value of a hidden-role equilibrium is in general NP-hard.

For the remainder of this section, when discussing the problem of "computing the value of a game", we always mean the following promise problem: given a game, a threshold v, and an allowable error $\epsilon > 0$ (both expressed as rational numbers), decide whether the hidden-role value of Γ is $\geq v$ or $\leq v - \epsilon$. Our hardness results will hold even when $\epsilon = 1/\mathsf{poly}(|H|)$. Further, utilities will be given *unnormalized* by chance probability. That is, if we say that a player gets utility 1, what we really mean is that the player gets utility $1/p$, where p is the probability that chance sampled all actions on the path to z. Thus the contribution to the expected value from this terminal node will be 1. This makes calculations easier. Finally, the utility range of the games used in the reductions will usually be of the form $[-M, M]$ where M is large but polynomial in the size of the game. Our definition of an extensive-form game allows only games with reward range $[-1, 1]$. This discrepancy is easily remedied by dividing all utility values in the proofs by M.

Theorem 5.18. *Even in* 2*-vs-*1 *games with public roles and* $\epsilon = 1/\mathsf{poly}(|H|)$, *computing the TME value (and hence also the hidden-role value, since adversarial team games are a special case of hidden-role games) with public communication is* NP*-hard.*

Intuitively, the proofs work by constructing gadgets that prohibit any useful communication, thus reducing to the case of no communication.

Proof. We show that given any graph G, it is possible to construct a hidden-role game based on G whose value correspond to the size the graph's max-cut. This reduces MAX-CUT to the TME value problem.

Let G be an arbitrary graph with n nodes and m edges, and consider the following team game (no hidden roles). There are 3 players, 2 of whom are on team △. The game progresses as follows.

1. Chance chooses two vertices v_1, v_2 in G, independently and uniformly at random. The two players on team △ observe v_1 and v_2 respectively.
2. The two players on team △ select bits b_1, b_2, and the ▽ player selects a pair (v_1', v_2'). There are now several things that can happen: (L is a large number to be picked later)
   (a) (*Agreement of players*) If $v_1 = v_2$ and $b_1 \neq b_2$, team △ gets utility $-L$.
   (b) (*Objective*) If $v_1 \neq v_2$, $b_1 \neq b_2$, and (v_1, v_2) is an edge in G, then team △ gets utility 1.
   (c) (*Non-leakage*) If $(v_1', v_2') = (v_1, v_2)$ then team △ gets utility $-(n^2 - 1)L$. Otherwise, team △ gets utility L.[5.12]

Consider a sufficiently large $L = \mathsf{poly}(m)$. The game is designed in such a way that ▽'s objective is to guess the vertices v_1, v_2 sampled by chance, but she has no information apart from the transcript of communication to guess it. Therefore, ▽'s optimal strategy is to punish any communication attempt between the players and play the most likely pair of vertices v_1, v_2. If no communication happens, her best strategy is to play a random pair of vertices. On the other hand, △ optimal strategy must ensure that under no circumstance players play the same bit when assigned to the same vertex (lest they incur the large penalty L). Therefore, the strategy of both △ players is to play a fixed bit in each vertex, and the optimal strategy is the one that assigns a bit to the vertices in such a way that the number of edges connecting vertices with different bits is maximized. This corresponds to finding a max cut and therefore the value of the game is (essentially) c^* where c^* is the true size of the maximum cut. Moreover, any communication attempt would be immediately shut down by ▽ strategy since any leak of the observation received on the public channel implies to receive a fraction of the large penalty L.

We now formalize this intuition.

First, note that △ can achieve utility exactly c^* by playing according to a maximum cut. To see that △ cannot do significantly better, consider the following strategy for team ▽. Observe the entire transcript τ of messages shared between the two △ players. Pick (v_1, v_2) maximizing the probability $p(\tau | v_1, v_2)$ that the players would have produced τ.[5.13]

First, suppose that △ does not communicate. Then ▽'s choice is independent of △'s, so △ can WLOG play a pure strategy. If $b_1 \neq b_2$ for any pair (v_1, v_2), then △ loses utility L in expectation. For $L > n^2$, this makes any such strategy certainly inferior to playing the maximum cut. Therefore, △ should play the maximum cut, achieving value c^*.

Now suppose that △ uses communication. Fix a transcript τ, and let

$$\delta := \max_{v_1, v_2} p(v_1, v_2 | \tau) - \frac{1}{n^2}.$$

Now note that we can write $p(\cdot|\tau) = (1-\alpha)q_0 + \alpha q_1$, where $q_0, q_1 \in \Delta(V^2)$, q_0 is uniform, and $\alpha \leq \Theta(n^4 \delta)$. Now consider any strategy that △ could play, given transcript τ. Such a strategy has the form $x_1(b_1|v_1)$ and $x_2(b_2|v_2)$. Since q is α-close to uniform, the utility of △ under this strategy conditioned on τ must be bounded above by

$$(1-\alpha)u_0 + \alpha \leq (1-\alpha)c^* + \alpha \leq c^* + \alpha \leq c^* + \Theta(n^4\delta)$$

where u_0 is the expected value of profile (x_1, x_2) given τ if $v_1, v_2 | \tau$ were truly uniform. But now, in expectation over τ, team ▽ can gain utility $Ln^2\delta$ by playing $\operatorname{argmax}_{v_1, v_2} p(v_1, v_2|\tau)$. So, △'s utility is bounded above by

$$c^* + \Theta(n^4\delta) - Ln^2\delta \leq c^*$$

by taking L sufficiently large. □

Since there is only one ▽-player in the above reduction, the result applies regardless of whether the adversary is coordinated.

The next result illustrates the difference between the uncoordinated hidden-role value and the coordinated hidden-role value which is the focus of our positive results. Whereas the coordinated hidden-role value with private communication can be computed in polynomial time when k is

[5.12] Note that ▽ playing uniformly at random means that the expected utility of this term is 0.

[5.13] Note again, as in Section 5.5, that this computation may take time exponential in r and the size of G, but we allow the players to perform unbounded computations.

	Communication Type		
Adversary Team Assumptions	**None**	**Public**	**Private**
Coordinated, Minority	NP-complete [292]	NP-hard [Thm. 5.18]	P [Thm. 5.12]
Coordinated			*open problem*
Minority	Σ_2^{P}-complete [Thms. 5.19, 4.33]	Σ_2^{P}-hard [Thm. 5.21]	coNP-hard [Thm. 5.21]
None			

Table 5.1: *Complexity results for computing hidden-role value with a constant number of players, for various assumptions about the adversary team and notions of communication. The results shaded in green are new.*

constant (Theorem 5.12), the uncoordinated hidden-role value cannot, even when $k = 2$:

Theorem 5.19. *Even in* 3*-vs-*2 *hidden-role games, the uncoordinated hidden-role value problem with private communication is* coNP*-hard.*

Proof. We reduce from UNSAT. Let ϕ be any 3-CNF-SAT formula, and consider the following 5-player hidden-role game. Two players are chosen uniformly at random to be on team ▽; the rest are on team △. The players on team ▽ know each other. The players on team ▽ play the SAT gadget game described by Koller and Megiddo [176]. Namely:

1. The players on team ▽ are numbered P1 and P2, at random, by chance.
2. Chance selects a clause C in ϕ and tells P1.
3. P1 selects a variable x_i in C, and that variable (but not its sign in C, nor the clause C itself) is revealed to P2.
4. P2 selects an assignment $b_i \in \{0, 1\}$ to x_i. ▽ wins the gadget game if the assignment b_i matches the sign of x_i in C.

The value of this game is decreasing with M and R since △ does nothing, so it is in the best interest of △ to select $R = 0$ (*i.e.*, allow no communication). In that case, the best probability with which ▽ can win the game is exactly the maximum fraction of clauses satisfied by any assignment, which completes the proof. □

The above result is fairly straightforward: it is known that optimizing the joint strategy of a team with asymmetric information[5.14] is hard [176], and private communication does not help if △ does not allow its use. However, next, we will show that the result even continues to apply when ▽ has *symmetric* information, that is, when the original game Γ is coordinated. This may seem mysterious at first, but the intuition is the following. Just because Γ has symmetric information for the ▽-team, does not mean $\textsc{USplit}(\textsc{Comm}_{\mathsf{priv}}(\Gamma))$ does. Indeed, △-players can send different private messages to different ▽-players, resulting in asymmetric information among ▽-players. This result illustrates precisely the reason that we define two different split-personality

[5.14] For our purposes, we will say that ▽ has *symmetric information* if all players ▽ have the same observation at every timestep. This implies that they can be merged into a single player without loss of generality.

games, rather than simply dealing with the special case where the original game Γ has symmetric information for the ▽-team.

Theorem 5.20. *Even in* 3*-vs-*2 *hidden-role games with a mediator in which no* ▽*-player has any information beyond the team assignment, the uncoordinated hidden-role value problem with private communication is* coNP*-hard.*

Proof. We will reduce from (the negation of) MAX-CUT. Let G be an arbitrary graph with n nodes and m edges, and consider the following 5-player hidden-role game with 2 players on the ▽-team and 3 players on the △-team. Player 5 is always on team △ and is the mediator. The other four players are randomly assigned teams so that two are on team △ and two are on team ▽. The game proceeds as follows.

1. Chance chooses vertices $v_1, \dots, v_4$ uniformly at random from G. The mediator privately observes the whole tuple $(v_1, \dots, v_4)$.
2. For notational purposes, call the players on team △ 3 and 4, and ▽ 1 and 2. (The mediator does not know these numbers.) After some communication, the following actions happen simultaneously:
   (a) P1 and P2 select bits $b_1, b_2 \in \{0, 1\}$;
   (b) P3 and P4 select vertices v'_3, v'_4 of G and players $i_3, i_4 \in \{1, 2, 3, 4\}$.
3. The following utilities are given: (L is a large number to be picked later, and each item in the list represents an additive term in the utility function)
   (a) (*Correct vertex identification*) For each player $i \in \{3, 4\}$, if $v_i \neq v'_i$ then △ gets utility $-L^4$
   (b) (*Agreement of* ▽*-players*) If $v_1 = v_2$ and $b_1 \neq b_2$ then team △ gets utility L.
   (c) (*Objective*) If $v_1 \neq v_2$, $b_1 \neq b_2$, and (v_1, v_2) is an edge in G, then team △ gets utility -1.
   (d) (*Privacy*) For each $j \in \{3, 4\}$, if i^*_j is on team ▽, then △ gets utility L. Otherwise, △ gets utility $-L/2$.

We claim that this game has value (essentially) $-c^*$, where c^* is the actual size of the maximum cut of G. To see this, observe first that the mediator *must* tell all players their true vertices v_i, lest it risk incurring the large negative utility $-L^2$. Further, any player except the mediator who sends a message must be on team ▽. This prevents team ▽ from communicating. Thus, the mediator's messages force ▽ to play an asymmetric-information identical-interest game, which is hard.

We now formalize this intuition. First, consider the following strategy for team △: The mediator sends all players their true types, and △-players play their types. If any △-player sees a message sent from anyone except the mediator, the △-player guesses that that player is on team ▽.

Now consider any (pure) strategy profile of team ▽. First, ▽ achieves utility $-c^*$ by observing the mediator's message and playing bits according to a maximum cut. We now show that this is the best that ▽ can do. Sending messages is, as before, a bad idea. Thus, a pure strategy profile of ▽ is given by four $f_1, f_2, f_3, f_4 : V \to \{0, 1\}$ denoting how player i should pick its bits. But then $f_1 = f_2 = f_3 = f_4$; otherwise, the agreement of ▽-players would guarantee that ▽ is not playing optimally for large enough L.

Now, for any (possibly mixed) strategy profile of team △, consider the following strategy profile for each ▽-player. Let $f : V \to \{0, 1\}$ be a maximum cut. Pretend to be a △-player, and let v_i' be the vertex that would be played by that △-player. Play $f(v_i')$.

First, consider any △-player strategy profile for which, for some player i and some v_i, the probability that $v_i' \neq v_i$ exceeds $1/L^2$. Then △ gets a penalty of roughly L^2 in expectation, but now setting L large enough would force △ to have utility worse than -1, so that △ would rather simply play v_i with probability 1.

Now, condition on the event that $v_i = v_i'$ for all i (probability at least $1 - \Theta(1/L^2)$). In that case, the utility of △ is exactly $-c^*$, because ▽ is playing according to the maximum cut. Thus, the utility of △ is bounded by $-(1 - \Theta(1/L^2))c^* + \Theta(1/L^2 \cdot L) \leq -c^*/n^2 + \Theta(n/L) < c^* + 1/2$ for sufficiently large L. Thus, solving the hidden-role game to sufficient precision and rounding the result would give the maximum cut, completing the proof. □

Theorem 5.21. *Even in* 5*-vs-*4 *hidden-role games, the uncoordinated hidden-role value problem with public communication is* Σ_2^{P}*-hard.*

Proof. We reduce from ∃∀3-DNF-SAT, which is Σ_2^{P}-complete [144]. The ∃∀3-DNF-SAT problem is the following. Given a 3-DNF formula $\phi(x, y)$ with k clauses, where $x \in \{0, 1\}^m$ and $y \in \{0, 1\}^n$, decide whether $\exists x \forall y \; \phi(x, y)$. Consider the following game. There are 9 players, 5 on team △ and 4 on team ▽. One designated player, who we will call P0, is △ and has no role in the game. (The sole purpose of this player is so that △ is a majority.) The other players are randomly assigned teams. These other players are randomly assigned teams. For the sake of analysis, we number the remaining players P1 through P8 such that P1, P3, P4, P5 are on team △ and P2, P6, P7, P8 are on team ▽. ▽ knows the entire team assignment, whereas △ dos not. We will call P3–P8 "regular players", and P1–P2 "guessers". The game proceeds as follows.

1. For each regular △-player, chance selects a literal (either x_j or $\neg x_j$), uniformly at random. For each regular ▽-player (P6–8), chance selects a literal (either y_j or $\neg y_j$), also uniformly at random. Each player privately observes the *variable* (index j), but not the sign of that variable.
2. After some communication, the following actions happen simultaneously.
   (a) P3–P8 select assignments (0 or 1) to their assigned variables.

(b) P1 (who observes nothing) guesses a player (among the six players P3–P8) that P1 believes is on team ▽.

(c) P2 (who observes nothing) guesses one literal for each △-player (there are $K := (2m)^3$ such possible guesses.)

3. The following utilities are assigned. (L is a large number to be picked later, and each item in the list represents an additive term in the utility function)

   (a) (*Satisfiability*) Chance selects three regular players at random. If the three literals given to those players form a clause in ϕ, and that clause is satisfied, △ gets utility 1.

   (b) (*Consistency*) If chance selected the same variable three times, if the three players did not give the same assignment, then the team (△ if the variable was an x_i and ▽ if the variable was a y_j) gets utility $-L^2$.

   (c) (*Privacy for* △) If P2 guesses the three literals correctly, ▽ gets utility L^3K; otherwise, ▽ gets utility $-L^3$.[5.15]

   (d) (*Privacy for* ▽) If P1 guesses a ▽-player, △ gets utility L; otherwise, △ gets utility $-L$.

We claim that the value of this game with public communication is at least 1 if and only if ϕ is $\exists\forall$-satisfiable. Intuitively, the rest of the proof goes as follows:

1. △ will not use the public communication channels if ϕ is $\exists\forall$-satisfiable. By the *Privacy for* ▽ term, ▽-players therefore cannot do so either without revealing themselves immediately. Thus, △ will get utility at least 1 if ϕ is $\exists\forall$-satisfiable.

2. If ϕ is not $\exists\forall$-satisfiable, then consider any △-team strategy profile. By the *Privacy for* △ term, team △ cannot make nontrivial use of the public communication channel without leaking information to P9. By the *Consistency* term, P1 through P3 must play the same assignment, or else incur a large penalty. So, △ must play essentially an assignment to the variables in x, but such an assignment cannot achieve positive utility because there will exist a y that makes ϕ unsatisfied.

We now formalize this intuition. Suppose first that ϕ is $\exists\forall$-satisfiable, and let x be the satisfying assignment. Suppose that △-players never communicate and assign according to x, and P4 guesses any player that sends a message. Then any ▽-strategy that sends a message is bad for sufficiently large L because it guarantees a correct guess from △; any ▽-strategy that is inconsistent is bad because it will lose utility at least L; and any ▽-strategy that is consistent will cause △ to satisfy at least one clause. Thus △ guarantees utility at least 1.

Now suppose that ϕ is not $\exists\forall$-satisfiable. Consider any strategy profile for △. Suppose that the ▽-players play as follows. During the public communication phase, each ▽-player samples a literal from the set $\{x_1, \neg x_1, \dots, x_m, \neg x_m\}$ uniformly at random and pretends to be a △-player given that literal. P8 observes the public transcript, and selects the triplet of literals that is conditionally most likely given the transcript. By an identical argument to that used in Theorem 5.18, △ then cannot profit from using the communication channel for sufficiently large L. Therefore we can assume that △ does not use the communication channels, and therefore by the argument in the previous paragraph, neither does ▽.

Now, the strategy of each △-player i can be described by a vector $s_i \in [0,1]^m$, where s_{ij} is the probability that player i assigns 1 to variable x_t. For sufficiently large L, we have $s_i \in [0, \epsilon] \cup [1-\epsilon, 1]$ where $\epsilon = 1/L$, because otherwise △ would incur a penalty proportional to $L^2\epsilon > k$ and would rather just play (for example) the all-zeros profile, which guarantees value 0. Condition on the event that every player at every variable chooses to play the most-likely assignment according to the s_is. This happens with probability at least $1 - \Theta(m\epsilon)$. These assignments must be consistent (*i.e.*, every player must have the same most-likely assignment), or else the players would once again incur a large penalty proportional to L^2. Call that assignment x, and let y be such that $\phi(x, y)$ is unsatisfied. Suppose ▽ plays according to y. Then △'s expected utility is bounded above by $\Theta(m\epsilon k)$: with probability $1 - \Theta(m\epsilon)$ it is bounded above by 0; otherwise it is bounded above by k. For $\epsilon < \Theta(1/mk)$ this completes the proof. □

5.5 Worked Example

This section includes a worked example of value computation to illustrate the differences among the notions of equilibrium discussed in this chapter and illustrates the utility of having a mediator for private communication. Consider a n-player version of matching pennies $\mathsf{MP}(n)$ as defined in Section 5.1.4.

Proposition 5.22. *Let* $\mathsf{MP}(n)$ *be the* n*-player matching pennies game.*

1. *The TMECor and TMECom values of* PUBLICTEAM$(\mathsf{MP}(n))$ *are both* $1/2$.
2. *Without communication or with only public communication, the value of* $\mathsf{MP}(n)$ *is* $1/2^{n-1}$.
3. *With private communication, the value of* $\mathsf{MP}(n)$ *is* $1/(n+1)$.

Proof. The first claim, as well as the no-communication value, is known [19].

For the public-communication value, observe that, conditioned on the transcript, the bits chosen by the players must be mutually independent of each other. Thus, the adversary can do the following: pretend to be on team △, wait for all communication to finish, and then play 0 if the string of all ones is more conditionally likely than the string of all 1s, and vice-versa[5.16].

It thus only remains to prove the third claim.

(*Lower bound*) The players simulate a mediator using multi-party computation (see Theo-

[5.15] These utilities are once again selected so that a uniformly random guess gets utility 0.

rems 5.11 and 5.12). Consider the following strategy for the mediator. Sample a string $b \in \{0, 1\}^n$ uniformly at random from the set of $2n + 2$ strings that has at most one mismatched bit. Recommend to each player i that they play b_i.

Consider the perspective of the adversary. The adversary sees only a recommended bit b_i. Assume WLOG that $b_i = 0$. Then there are $n + 1$ possibilities:

1. b is all zeros (1 way)
2. All other bits of b are 1 (1 way)
3. Exactly one other bit of b is 1 ($n - 1$ ways).

The adversary wins in the third case automatically (since the team has failed to coordinate), and, regardless of what the adversary does, it can win only one of the first two cases. Thus the adversary can win at most $n/(n + 1)$ of the time, that is, this strategy achieves value $1/(n + 1)$.

(*Upper bound*) Consider the following adversary strategy. The adversary communicates as it would do if it were on team △. Let b_i be the bit that the adversary would play if it were on team △. The adversary plays b_i with probability $1/(n + 1)$ and $1 - b_i$ otherwise. We need only show that no pure strategy of the medaitor achieves value better than $1/(n + 1)$ against this adversary. A strategy of the mediator is identified by a bitstring b. If b is all zeros or all ones, the team wins if and only if the adversary plays b_i (probability $1/(n + 1)$). If b has a single mismatched bit, the team wins if and only if the mismatched bit is the adversary (probability $1/n$) and the adversary flips b_i (probability $n/(n + 1)$). □

5.6 Properties of Hidden-role Equilibria

In the following, we discuss interesting properties of hidden-role equilibria given the definition we provided in Section 5.1.1, and that make them fairly unique relative to other notions of equilibrium in team games.

5.6.1 The Price of Hidden Roles

One interesting question arising from hidden-role games is the *price* of having them. That is, how much value does △ lose because roles are hidden? In this section, we define this quantity and derive reasonably tight bounds on it.

Definition 5.23. The *public-team refinement* of an n-player hidden-role game Γ is the adversarial team game $\text{PUBLICTEAM}(\Gamma)$ defined by starting with the (uncoordinated) split-personality game, and adding the condition that all team assignments t_i are publicly observed by all players.

[5.16] In general, computing the conditional probabilities could take exponential time, but when defining the notion of value here, we are assuming that players have unbounded computational resources. This argument not work for computationally-bounded adversaries. Indeed, if the adversary were computationally bounded, △ would be able to use cryptography to build private communication channels and thus implement a mediator, allowing our main positive result Theorem 5.12 to apply.

Definition 5.24. For a given hidden-role game Γ in which △ is guaranteed a nonnegative value (*i.e.*, $u_i(z) \geq 0$ whenever i is on team △), the *price of hidden roles* $\mathsf{PoHR}(\Gamma)$ is the ratio between the TME value of $\textsc{PublicTeam}(\Gamma)$ and the hidden-role value of $\textsc{USplit}(\Gamma)$.

For a given class of hidden-role games $\mathcal{G}$, the price of hidden roles $\mathsf{PoHR}(\mathcal{G})$ is the supremum of the price of hidden roles across all games $\Gamma \in \mathcal{G}$.

Theorem 5.25. *Let $D \in \Delta(\{\triangle, \triangledown\}^n)$ be any distribution of teams assignments. Let $\mathcal{G}_{n,D}$ be the class of all hidden-role games with n players and team assignment distribution D. Then the price of hidden roles of $\mathcal{G}_{n,D}$ is exactly the largest probability assigned to any team by D, that is,*

$$\mathsf{PoHR}(\mathcal{G}_{n,D}) = \max_{t \in \{\triangle, \triangledown\}^n} \Pr_{t' \sim D}[t' = t].$$

The lower bound is achieved even in the presence of private communication.

Proof. Let t^* be the team to which D assigns the highest probability, and let p^* be that probability. Our goal is to show that the price of hidden roles is $1/p^*$.

(*Upper bound*) Team △ assumes that the true △-team is exactly the team t^*. Then △ gets utility at most a factor of $1/p^*$ worse than the TME value of $\textsc{PublicTeam}(\Gamma)$: if the assumption is correct, then △ gets the TME value; if the assumption is incorrect, △ gets value at least 0 thanks to the condition on △'s utilities in Definition 5.24.

(*Lower bound*) Consider the following game Γ. Nature first selects a team assignment $t \sim D$ and each player privately observes its team assignment. Then, all players are simultaneously asked to announce what they believe the true team assignment is. The △-team wins if every △-player announces the true team assignment. If △ wins, △ gets utility 1; otherwise △ gets utility 0.

Clearly, if teams are made public, △ wins easily. With teams not public, suppose that we add a mediator to the game so that Theorem 5.10 applies. This cannot decrease △'s value. The mediator's strategy amounts to selecting what team each player should announce. Mediator strategies in which different players announce different teams are dominated. The mediator strategy in which the mediator tells every player to announce team t wins if and only if t is the true team, which happens with probability at most p^* (if $t = t^*$). Thus, even the game with a mediator added has value at most p^*, completing the proof. □

This implies immediately:

Corollary 5.26. *Let $\mathcal{G}_{n,k}$ be the class of all hidden-role games where the number of players and adversaries are always exactly n and k respectively. The price of hidden roles in $\mathcal{G}_{n,k}$ is exactly $\binom{n}{k}$.*

In particular, when $k = 1$, the price of hidden roles is at worst n. This is in sharp contrast to the

Variant	**5 Players**	**6 Players**
No special roles (*Resistance*)	3 / 10 = 0.3000*	1 / 3 ≈ 0.3333*
Merlin	2 / 3 ≈ 0.6667*	3 / 4 = 0.7500*
Merlin + Mordred	731 / 1782 ≈ 0.4102	6543 / 12464 ≈ 0.5250
Merlin + 2 Mordreds	5 / 18 ≈ 0.2778	99 / 340 ≈ 0.2912
Merlin + Mordred + Percival + Morgana	67 / 120 ≈ 0.5583	—

Table 5.2: *Exact equilibrium values for 5- and 6-player* Avalon. *The values marked * were also manually derived by Christiano [62]; we match their results. '—': too large to solve.*

price of communication and *price of correlation* in ATGs, both of which can be arbitrarily large even when $n = 3$ and $k = 1$ [19, 56].

5.6.2 Order of Commitment and Duality Gap

In Definition 5.8, when choosing the TME as our solution concept and defining the split-personality game, we explicitly choose that △ should pick its strategy before ▽—that is, the team committing to a strategy is the same one that has incomplete information about the roles. One may ask whether this choice is necessary or relevant: for example, what happens when the TME problem (5.1) satisfies the minimax theorem? Perhaps surprisingly, the answer to this question is that, at least with private communication, *the minimax theorem in hidden-role games only holds in "trivial" cases*, in particular, when the hidden-role game is equivalent to its public-role refinement (Definition 5.23).

Proposition 5.27. *Let Γ be any hidden-role game. Define $\mathsf{UVal}'_{\mathsf{priv}}(\Gamma)$ identically to $\mathsf{UVal}_{\mathsf{priv}}(\Gamma)$, except that ▽ commits before △—that is, in (5.1), the maximization and minimization are flipped. Then $\mathsf{UVal}'_{\mathsf{priv}}(\Gamma)$ is equal to the TME value of $\textsc{PublicTeam}(\Gamma)$ with communication—that is, the equilibrium value of the zero-sum game in which teams are public and intra-team communication is private and unlimited.*

Proof. It suffices to show that team △ can always cause the teams to be revealed publicly if ▽ commits first. Let s be a long random string. All members of team △ broadcast s publicly at the start of the game. Since ▽ commits first, ▽ cannot know or guess s if it is sufficiently long; thus, with exponentially-good probability, this completely reveals the teams publicly. Then, using the private communication channels, team △ can play a TMECom of $\textsc{PublicTeam}(\Gamma)$. □

Therefore, the choice of having △ commit to a strategy before ▽ is forced upon us: flipping the order of commitment would ruin the point of hidden-role games.

5.7 Experimental Evaluation: *Avalon*

In this section, we apply Theorem 5.11 to instances of the popular hidden-role game *The Resistance: Avalon* (hereafter simply *Avalon*). We solve various versions of the game with up to six players.

A game of *Avalon* proceeds, generically speaking, as follows. There are n players, $\lceil n/3 \rceil$ of which are randomly assigned to team ▽ and the rest to team △. Team ▽ is informed. Some special rules allow players observe further information; for example, *Merlin* is a △-player who observes the identity of the players on team ▽, except the ▽-player *Mordred*, and the △-player *Percival* knows *Merlin* and *Morgana* (who is on team ▽), but does not know which is which. The game proceeds in five rounds. In each round, a *leader* publicly selects a certain number of people (defined as a function of the number of players and current round number) to go on a *mission*. Players then publicly vote on whether to accept the leader's choice. If a strict majority vote to accept, the mission begins; otherwise, leadership goes to the player to the left. If four votes pass with no mission selected, there is no vote on the fifth mission (it automatically gets accepted). If a ▽-player is sent on a mission, they have the chance to *fail* the mission. The goal of △ is to have three missions pass. If *Merlin* is present, ▽ also wins by correctly guessing the identity of Merlin at the end of the game. *Avalon* is therefore parameterized by the number of players and the presence of the extra roles *Merlin, Mordred, Percival*, and *Morgana*.

Avalon is far too large to be written in memory: Serrino et al. [269] calculates that 5-player *Avalon* has at least 10^{56} information sets. However, in *Avalon* with ≤ 6 players, many simplifications can be made to the zero-sum game given by Theorem 5.11 without changing the equilibrium. These are detailed in the appendix of the full paper [54], but here we sketch one of them which has theoretical implications. Without loss of generality, in the zero-sum game in Theorem 5.11, the mediator completely dictates the choice of missions by telling everyone to propose the same mission and vote to accept missions, and ▽ can do nothing to stop this. Therefore, team ▽ always has symmetric information in the game: they know each others' roles (at least when $n \leq 6$), and the mediator's recommendations to the players may as well be public. Therefore, *Avalon* is already natively without loss of generality a game with a coordinated adversary in the sense of Section 5.3.2, so the seemingly strong assumptions used in Definition 5.6 are in fact appropriate in *Avalon*. Even after our simplifications, the games are fairly large, *e.g.*, the largest instance we solve has 2.2 million infosets and 26 million terminal nodes.

Our results are summarized in Table 5.2. Games were solved using a CPU compute cluster machine with 64 CPUs and 480 GB RAM, using two algorithms:

1. A parallelized version of PCFR+. PCFR+ was able to find an approximate equilibrium with exploitability $< 10^{-3}$ in less than 10 minutes in the largest game instance, and was able to complete 10,000 iterations in under two hours for each game.

2. An implementation of the simplex algorithm with exact (rational) precision, which was warmstarted using incrementally higher-precision solutions obtained from configurable finite-precision floating-point arithmetic implementation of the simplex algorithm, using an algorithm similar to that of Farina et al. [99]. This method incurred significantly higher

runtimes (in the order of hours to tens of hours), but had the advantage of computing *exact* game values at equilibrium.

Table 5.2 shows exact game values for the instances we solved.

Findings. We solve *Avalon* exactly in several instances with up to six players. In the simplest instances (*Resistance* or only Merlin), Christiano [62] previously computed equilibrium values by hand. The fact that we match those results is positive evidence of the soundness of both our equilibrium concepts and our algorithms.

Curiously, as seen in Table 5.2, the game values are not "nice" fractions: this suggests to us that most of the equilibrium strategies will likely be inscrutable to humans. The simplest equilibrium not previously noted by Christiano, namely Merlin + 2 Mordreds with 5 players, is scrutable, and is analyzed in detail in the appendix of the full paper [54].

Also curiously, having Merlin and two Mordreds (*i.e.*, having a Merlin that does not actually know anything) is not the same as having no Merlin. If it were, we would expect the value of Merlin and two Mordreds to be $0.3 \times 2/3 = 0.2$ (due to the 1/3 probability of $\triangledown$ randomly guessing Merlin). But, the value is actually closer to 0.28. The discrepancy is due to the "special player" implicit correlation discussed in Section 5.1.4.

5.8 Conclusions and Future Research

In this chapter, we have initiated the formal study of hidden-role games from a game-theoretic perspective. We build on the growing literature on ATGs to define a notion of equilibrium, and give both positive and negative results surrounding the efficient computation of these equilibria. In experiments, we completely solve real-world instances of *Avalon*. As this chapter introduces a new and interesting class of games, we hope that it will be the basis of many future papers as well. We leave many interesting questions open.

1. From our results, it is not even clear that hidden-role equilibria and values can be computed in *finite* time except as given by Theorem 5.12. Is this possible? For example, is there a revelation-principle-like characterization for *public* communication that would allow us to fix the structure of the communication? We believe this question is particularly important, as humans playing hidden-role games are often restricted to communicating in public and cannot reasonably run the cryptographic protocols necessary to build private communication channels or perform secure MPC.

2. Changing the way in which communication works can have a ripple effect on all the results. One particular interesting change that we do not investigate is *anonymous* messaging, in which players can, publicly or privately, send messages that do not leak their own identity. How does the possibility of anonymous messaging affect the central results of this chapter?

3. We do not investigate or define hidden-role games where *both* teams have imperfect information about the team assignment. What difference would that make? In particular, is there a way to define an equilibrium concept in that setting that is "symmetric" in the sense that it

does not require a seemingly-arbitrary choice of which team ought to commit first to its strategy?

Chapter 6

A Framework for Optimal Correlated Equilibria and Mechanism Design in Extensive-Form Games

6.1 Introduction

Various equilibrium notions in general-sum extensive-form games are used to describe situations where the players have access to a trusted third-party *mediator*, who can communicate with the players. Depending on the power of the mediator and the form of communication, these notions include the *normal-form* [15] and *extensive-form correlated equilibrium* (NFCE and EFCE) [291], the *normal-form* [226] and *extensive-form* [102] *coarse-correlated equilibrium* (NFCCE and EFCCE), the *communication equilibrium* [109, 228], and the *certification equilibrium* [110].

Several of these notions, in particular the EFCE and EFCCE, were defined for mainly *computational* reasons: the EFCE as a computationally-reasonable relaxation to NFCE, and the EFCCE as a computationally-reasonable relaxation of EFCE. When the goal is to compute a *single* correlated equilibrium, these relaxations are helpful: there are polynomial-time algorithms for computing an EFCE [153]. However, from the perspective of computing *optimal* equilibria—that is, equilibria that maximize the expected value of a given function, such as the social welfare—even these relaxations fall short: for all of the *correlation* notions above, computing an optimal equilibrium of an extensive-form game is NP-hard [102, 291].

On the other hand, notions of equilibrium involving *communication* in games have arisen. These differ from the notions of *correlation* in that the mediator can receive and remember information from the players, and therefore pass information *between* players as necessary to back up their suggestions. *Certification equilibria* [110] further strengthen communication equilibria by allowing players to *prove* certain information to the mediator. To our knowledge, the computational complexity of optimal communication or certification equilibria has never been studied. We do so in this chapter.

We make several contributions in this chapter.

Section 6.3: General equilibrium concepts. We define a general class of equilibrium concepts, parameterized by a *communication protocol* that governs how the communication between the mediator and the player should take place, and *privacy constraints* for the mediator that define, intuitively, what messages the mediator is not allowed to share with other players. Our class of equilibrium concepts includes, as special cases, all of the concepts discussed so far except the *normal-form correlated equilibrium* (NFCE). Intuitively, our class includes essentially any concept that can be formulated in terms of a communication device (mediator) that gives *local action recommendations* to players.

Section 6.4: Algorithms for optimal direct equilibria. A *direct* equilibrium is one in which the players all follow some predetermined strategy, namely, always reporting information honestly and always obeying action recommendations. We give an algorithm that recovers an optimal *direct* equilibrium for any of these equilibrium concepts in polynomial time in the size of the original game, as long as the mediator has perfect recall.[6.1]

Critically, we employ a *reduced mediator-augmented game* that has size polynomial in the size of the original game, yet captures all meaningful strategic considerations. Our reduced mediator-augmented game improves upon earlier constructions for communication equilibria in several critical ways, that together allow it to be of polynomial size:

1. Players who have provably deviated (*e.g.*, by sending messages that cannot possibly be consistent with truthful play) are assumed to not receive any further recommendations.
2. Only one player is allowed to deviate. That is, once any one player has deviated, other players are assumed to be direct for the remainder of the game.

Section 6.5: Revelation principle. We state and prove a general version of the *revelation principle* for our equilibrium class, which depends on an appropriate generalization of the *nested range condition* [134]. Therefore, in particular, under this condition, the algorithm from the previous section computes an optimal equilibrium.

One consequence of the previous two results is that an optimal communication equilibrium in an extensive-form game can be computed in polynomial time. To our knowledge, this chapter is the first to show this result.

Section 6.6: Optimal correlated equilibria. We then take a deep dive into *optimal correlated equilibria* as a special case of our framework. We will show that the *correlated* equilibrium concepts arise in our framework from considering mediators with *imperfect* recall, justifying the complexity gap between optimal EFCE (which is NP-hard) and the polynomial-time notions. As such, the algorithms from Chapter 4 can be used to develop parameterized algorithms for

[6.1] Mediator imperfect recall, as we will detail later, is used to represent *correlated* equilibrium concepts. For such concepts, our algorithm runs in polynomial time if the mediator's strategy space has a polynomial representation (*e.g.*, a polynomial-time separation oracle).

computing optimal correlated equilibria, with runtimes $O^*((b+1)^k)$ for NFCCE, $O^*((b+d)^k)$ for EFCCE, and $O^*((bd)^k)$ for EFCE, where b, d, k are as in Chapter 4 (branching factor, depth, and information complexity respectively). We also show that these bounds are essentially optimal: NP-complete problems can be solved by setting $b = O(1)$ and $k = O(n)$, and even removing the dependence on d for EFCCE and EFCE would break the exponential time hypothesis (ETH). We also show that our algorithms run in polynomial time in *two-player games of public chance*, matching the previously-known result of Farina and Sandholm [97].

Section 6.7: Other special cases. We discuss several other solution concepts and problems, including the communication and correlation notions discussed above, Bayes-correlated equilibria (with application to information design), and automated mechanism design. We show that these notions are all expressible in our framework, and therefore optimal equilibria in all of these concepts can be found in polynomial time as long as the mediator has perfect recall. (The imperfect-recall case, as above, leads to correlated equilibria.)

Section 6.9: Experiments. We empirically test the above claims via experiments on a standard set of game instances.

6.2 Applications and Related Work

Correlated and communication equilibria have various applications that have been well-documented. Here, we discuss just a few of them, as motivation.

Bargaining, negotiation, and conflict resolution [58, 101]. Two parties with asymmetric information wish to arrive at an agreement, say, the price of an item. A mediator, such as a central third-party marketplace, does not know the players' information but can communicate with the players.

Crowdsourcing and ridesharing [118, 204, 305]. A group of players each has individual goals (*e.g.*, to make money by serving customers at specific locations). The players are coordinated by a central party (*e.g.*, a ridesharing company) that has more information than any one of the players, but the players are free to ignore recommendations if they so choose.

Information design in games [57, 119, 167, 208, 296]. The mediator (in that literature, usually "sender") has more information than the players ("receivers"), and wishes to tell information to the receivers so as to persuade them to act in a certain way.

Automated mechanism design [67, 68, 172, 173, 235, 310, 311, 312]. Players have private information unknown to the mediator. The mediator wishes to commit to a strategy—that is, set a mechanism—such that players are incentivized to honestly reveal their information.

In fact, in Section 6.7.2 we will see that we recover many of the results in the above papers as special cases of our main result.

Some of the above examples are often used to motivate correlated equilibria. However, when the mediator is a rational agent with the ability to remember information that it is told and pass the information between players as necessary, we will argue that communication or certification equilibrium should be the notion of choice, for both conceptual and computational reasons.

6.3 Extensive-Form Communication Games

The central notion of interest in this chapter is a broad class of equilibrium concepts for extensive-form games which, as we will later show, captures many interesting problems—such as automated mechanism design, information design, correlated equilibria, communication and certification equilibria, and so on—as special cases. We will define notions of equilibrium by defining *mediator-augmented games*. These games will be created by augmenting the game with a mediator. The mediator has the power to send messages to and from the players, and *may* additionally have the power of forcing players to take certain actions to be taken in response to certain messages.

Formally, we start from a base game Γ, which is a general-sum timeable extensive-form game. There is a large, finite *message space* M that both the mediator and the players use to communicate. In particular, we will assume that messages can at least be either empty ($\perp$), information sets, or actions—formally, $\{\perp\} \sqcup \mathcal{I} \sqcup \mathcal{A} \subseteq M$. When a player reaches one of its infosets, it first selects a message from M and sends that message the mediator. The mediator replies with some other message, and then the player selects an action. No messages can be sent out-of-turn—that is, the only player allowed to communicate with the mediator is the one whose turn it is to play[6.2]. Players always remember all messages that they have sent and received so far.

The message $\perp$ is special: it indicates that a player wishes to immediately exit the communication protocol. That is, after a player sends $\perp$, all future messages between the player and mediator are forced to be $\perp$, and the player may play any action. The message $\perp$ exists because we will explore "coarse" equilibrium notions in which the player must choose whether or not *commit* to following the mediator's recommendations *before* knowing the recommendation: in such scenarios, sending the message $\perp$ allows a player to choose not to commit (and not to receive the recommendation).

We will use $\boldsymbol{\tau} = (\tau_1, \dots, \tau_n)$ to denote a transcript of messages, where τ_i is the list of messages exchanged between player i and the mediator. That is, a history of the mediator-augmented game is specified by a pair $(h, \boldsymbol{\tau})$ where h is the current history (in the original game) and $\boldsymbol{\tau}$ is the tuple of transcripts. For any message $m \in M$, we will use juxtaposition ($\tau_i m$) to denote appending m to τ_i.

6.3.1 Communication Protocols

We will restrict how the agents will communicate in all stages of the communication protocol: what messages the player can send to the mediator, what messages the mediator can send to the player, and what actions the player can take in response to receiving certain messages.

[6.2] This restriction in particular implies that the mediator will always know precisely how many actions each player has taken so far in the game, and in what order

Definition 6.1. A *communication protocol* is a tuple $(S, R, \mathcal{A})$, where

1. $S : I \to 2^M$ defines what messages the player is allowed to send, as a function of its current infoset. This allows some messages to be *certifiable*, *a la* Forges and Koessler [110]: for example, if there is only one information set I for which $m \in S(I)$, then sending m confirms to the mediator that the player's true infoset is I. Players are always allowed to honestly report its information. That is, $I \in S(I)$ for all infosets I. If communication between the mediator and a player has been terminated (*i.e.*, the player has sent $\perp$), the player must continue to send $\perp$ even if $\perp \notin S(I)$.

2. $R : M \to 2^M$ defines what messages the mediator can send to the player, as a function of the message it is sent. That is, if the mediator receives message m, it may only reply with messages $m' \in R(m)$. We will impose two conditions on R:

   a) If a player sends an infoset I, the mediator can send an action recommendation at I: $\mathcal{A}(I) \subseteq R(I)$.

   b) The mediator can always respond $\perp$: $\perp \in R(m)$ for all m

3. $\mathcal{A} : I \times M \to 2^{\mathcal{A}}$ defines[6.3] what actions a player is allowed to take as a function of its current *true* information (not the information that it reported) and what message it receives. If the player receives message m at information set I, it may only take actions in set $\mathcal{A}(I, m) \subseteq \mathcal{A}(I)$. We make two restrictions on A:

   a) Players can follow recommendations: $a \in \mathcal{A}(I, a)$ for all a and I.

   b) Players who get no recommendation can play any move: $\mathcal{A}(I, \perp) = \mathcal{A}(I)$.

Furthermore, we impose a restriction that any player who could have deviated (*i.e.*, by failing to send honest information, or failing to follow a recommendation) could also choose to exit the protocol. Formally, let I be an i-infoset of player i. If there is some i-sequence $I'a \prec I$ for which $\{I', \perp\} \not\supseteq S(I')$, or $a \in \mathcal{A}(I', a')$ for some action $a' \in \mathcal{A}(I') \setminus \{a\}$, then we insist that $\perp \in S(I)$. Intuitively, this condition states that *no player can be compelled to send a (nonempty) message proving that it has deviated.* We will call this the *Miranda condition*[6.4].

6.3.2 Imperfect Recall

We will sometimes consider cases where the mediator does not have a perfect memory. Formally, let $\mathcal{P} \subseteq [n]$ be a set of players at which messages will be considered *private*. When the mediator is deciding what message to send to a player i, the mediator remembers the transcripts with other players $j \neq i$ only if $j \notin \mathcal{P}$, as well as the entire transcript with the player i. Intuitively, the reason we call such messages *private* is that, by restricting the mediator's information partition in this fashion, the mediator is effectively prevented from *sharing* the information sent

[6.3] We are overloading the notation A in three ways: $\mathcal{A}$ (with zero arguments) is the set of all possible actions, $\mathcal{A}(I)$ is the set of actions legal at infoset I, and $\mathcal{A}(I, m)$ is the subset of actions legal at infoset I after being told message m. Which is meant should be clear from context.

[6.4] Unlike the right against self-incrimination in criminal law, the mediator very well may infer from a player's silence that the player has deviated! We thank Gabriele Farina for suggesting the name *Miranda condition*.

by players $i \in \mathcal{P}$ with any other player. In symbols, at a history $(h, \boldsymbol{\tau})$ where h is an i-node, the mediator's information is then the subvector $\boldsymbol{\tau}_{([n]\setminus\mathcal{P})\cup\{i\}}$, and any other history $(h', \tau_i, \boldsymbol{\tau}'_{-i})$ with $\boldsymbol{\tau}_{([n]\setminus\mathcal{P})\cup\{i\}} = \boldsymbol{\tau}'_{([n]\setminus\mathcal{P})\cup\{i\}}$ lies in the same infoset.

6.3.3 Ex-Post Incentive Compatibility

To capture notions surrounding *ex-post* incentive compatibility from the mechanism design literature, we introduce a *deviator information partition* $\mathcal{J} = \mathcal{J}_1 \sqcup \cdots \sqcup \mathcal{J}_n$, where each $\mathcal{J}_i$ is a refinement of the corresponding player information partition $\mathcal{I}_i$. When considering whether a profile is an equilibrium, we insist that no player i would have a profitable deviation, even if it observed information according to $\mathcal{J}_i$. For simplicity, we will also assume that $\mathcal{J}_i$ is perfect recall. Intuitively, working with two different partitions $\mathcal{J}_i$ and $\mathcal{I}_i$ allows us to separate the information partition in which the player must disclose information and act in equilibrium from the information partition that a possible deviator has. This is precisely the difference between *Bayes-incentive compatibility* and *ex-post incentive compatibility* in the mechanism design literature, a connection we will make clearer in Section 6.7.2.

Since the deviator information partition is defined over the *original* game Γ, it does not allow players to observe the messages exchanged between other players and the mediator—such messages remain private.

6.3.4 Generalized Communication Equilibria

The above communication protocol and information partition for the mediator, together, define an $(n+1)$-player *mediator-augmented game* $\hat{\Gamma}(S, R, \mathcal{A}, \mathcal{P})$ where the extra player is the mediator[6.5]. When $S, R, \mathcal{A}$, and $\mathcal{P}$ are clear from context, we will omit them and simply write $\hat{\Gamma}$ to mean the mediator-augmented game.

We will consistently use hats to distinguish the mediator-augmented game $\hat{\Gamma}$ from the original game Γ—for example, $\hat{h}$ and $\hat{I}$ will denote histories and infosets in $\hat{\Gamma}$, respectively. We will use the index 0 for the mediator, so that, for example, $\hat{\mathcal{I}}_0$ is the information partition of the mediator. We will write $\hat{\mathcal{J}}_i$ for the information partition given by extending player i's deviator information partition $\mathcal{J}_i$ to the augmented game. Finally, we will use Ξ to denote the set of all sequence-form mixed strategies for the mediator in $\hat{\Gamma}$, and $\boldsymbol{\xi} \in \Xi$ to denote a generic mediator strategy,

We are now ready to define the central solution concept of this chapter.

Definition 6.2. An $(S, R, \mathcal{A}, \mathcal{P}, \mathcal{J})$-*communication equilibrium* is a strategy profile $(\boldsymbol{\xi}, \hat{\boldsymbol{x}}) := (\boldsymbol{\xi}, \hat{\boldsymbol{x}}_1, \dots, \hat{\boldsymbol{x}}_n)$ in $\hat{\Gamma}(S, R, \mathcal{A}, \mathcal{P})$ such that, for every $i \in [n]$, player i's strategy is a best response even if its information partition were $\hat{\mathcal{J}}_i$ instead of $\hat{\mathcal{I}}_i$.

We have not specified any incentive constraints for the mediator. That is, the mediator is conceptualized as having the power to *commit* to a strategy $\boldsymbol{\xi}$. Thus, when evaluating whether a profile

[6.5] The deviator's information partition $\mathcal{J}$ is not part of the definition of the mediator-augmented game, because $\mathcal{J}$ only affects the incentive constraints, not what strategies are feasible.

$(\boldsymbol{\xi}, \boldsymbol{x})$ is an equilibrium, the mediator's strategy $\boldsymbol{\xi}$ should be treated as *fixed*.[6.6]

Every strategy profile $(\boldsymbol{\xi}, \hat{\boldsymbol{x}})$ of $\hat{\Gamma}$ induces a distribution over terminal nodes $z \in Z$ of the original game Γ. We will write a sample from this distribution as $z \sim (\boldsymbol{\xi}, \hat{\boldsymbol{x}})$. If two strategy profiles induce the same distribution, we call them *outcome-equivalent*. We will only be concerned with equilibria up to outcome-equivalence.

In many situations involving communication and correlation in games, it is desirable to have a *direct* equilibrium—that is, one in which the strategy of every player is simply "always report honest information and always follow mediator recommendations".

We will assume that the rules of Γ and the parameters $S, R, \mathcal{A}, \mathcal{P}, \mathcal{J}$ are common knowledge among all players including the mediator.

Definition 6.3. A strategy profile $(\boldsymbol{\xi}, \hat{\boldsymbol{x}})$ is *direct* if:

1. (*Mediator directness*) If the transcript τ_i of a player i is a sequence of player i, and player i sends an infoset I with $\sigma(I) = \tau_i$, then the mediator replies with an action $a \in \mathcal{A}(I)$. Otherwise[6.7], the mediator replies $\perp$.
2. (*Player directness*) Players always send their true information I, and, upon receiving an action $a \in \mathcal{A}(I)$, always play that action.

The directness constraints on each player specify a unique *direct strategy* $\hat{\boldsymbol{o}}_i \in \hat{\mathcal{X}}_i$, where $\hat{\mathcal{X}}_i \subset \mathbb{R}^{\hat{\Sigma}_i}$ is the realization-form strategy space of player i in $\hat{\Gamma}_{\text{red}}$, and $\hat{\mathcal{Z}}$ is the set of terminal nodes in $\hat{\Gamma}_{\text{red}}$: namely, we define $\hat{\boldsymbol{o}}_i$ to be the pure strategy in which player i always reports honest information and follows recommendations. That is, $(\boldsymbol{\xi}, \hat{\boldsymbol{x}})$ is direct if and only if $\hat{\boldsymbol{x}} = \hat{\boldsymbol{o}}$.

We will call any action by a player that violates the second condition above (*i.e.*, any misreporting of information, or disobedience of action recommendations) a *deviation* by that player.

6.4 Algorithm for Optimal Direct Equilibria

Given an objective function $u : \mathcal{Z} \to \mathbb{R}$, an *optimal* $(S, R, \mathcal{A}, \mathcal{P}, \mathcal{J})$*-communication equilibrium* is an equilibrium that maximizes the expected objective, $\mathbb{E}_{z\sim(\boldsymbol{\xi},\hat{\boldsymbol{x}})}\, u(z)$. In this section, we will be intersted in the following problem. Given a game, how should one compute an optimal equilibrium? That is, given a game Γ and the sets $S, R, \mathcal{A}, \mathcal{P}, \mathcal{J}$, is is it always possible to compute an optimal equilibrium efficiently?

We will start by focusing on the restriction to *direct* equilibria. By *optimal direct equilibrium*, we mean an optimal equilibrium among direct strategies. Such an equilibrium may not be optimal

[6.6] To emphasize that the mediator's strategy is fixed, Forges [109] calls the mediator a *communication device*, with the language suggesting that it is merely a fixed device used by the players rather than an entity with agency. The two perspectives are equivalent; here, we will also be interested in computing *optimal* equilibria, in which case it will make more sense to think of the mediator as a mechanism designer who can choose and commit to its strategy $\boldsymbol{\xi}$.

[6.7] This condition is necessary because, if the mediator does not know what infoset the player is in, the mediator may not be *able* to send the player a valid action, because action sets may differ by infoset.

across *all* equilibria, as it is possible in general for there to be an equilibrium that is not outcome-equivalent to any direct equilibrium. In the next section, we will give sufficient conditions on S and R such that *any* equilibrium can be converted to a direct equilibrium—if such conditions hold, then the optimal direct equilibrium computed by the above theorem is also optimal among all equilibria.

Theorem 6.4 (Algorithm for optimal direct equilibria)**.** *For any $S, R, \mathcal{A}, \mathcal{J}$, and objective u, an optimal direct $(S, R, \mathcal{A}, \mathcal{P}, \mathcal{J})$-communication equilibrium can be found by solving a linear program of size*[6.8] *$O(|\mathcal{Z}||\Sigma| + M)$, where M is the size of the constraint matrix describing the mediator's sequence-form mixed strategy set in $\hat{\Gamma}$. In particular, when $\mathcal{P} = \varnothing$, we have $M \le O(|\mathcal{Z}||\Sigma|)$, so this algorithm runs in polynomial time.*

Proof. Running computations directly on the mediator-augmented game $\hat{\Gamma}(S, R, \mathcal{A}, \varnothing)$ is hard, because the game may have a number of nodes that is exponential in $|\mathcal{Z}|$. To prove Theorem 6.4, we will therefore first construct a *reduced mediator-augmented game* $\hat{\Gamma}_{\text{red}}$ that has size polynomial in $|\mathcal{Z}|$. Then, we will show that an optimal direct communication equilibrium in $\hat{\Gamma}_{\text{red}}$ can be computed by a linear program of polynomial size. Finally, we will show that every communication equilibrium of $\hat{\Gamma}_{\text{red}}$ is outcome-equivalent to a communication equilibrium of $\hat{\Gamma}$.

Starting from the full mediator-augmented game $\hat{\Gamma}$, to construct the reduced game $\hat{\Gamma}_{\text{red}}$, we introduce the following restrictions.

1. A player can never send any message that proves that it has deviated. Formally, if the current transcript τ_i is a sequence of Γ for player i, then the player must either send $\perp$, or send an infoset I with $\sigma(I) = \tau_i$.[6.9]
2. When a player sends an infoset I to the mediator, the mediator may only send messages $a \in \mathcal{A}(I)$.
3. If some player has already deviated, then all other players always send honest information and obey action recommendations. (these nodes are replaced by nature nodes in $\hat{\Gamma}_{\text{red}}$, with one legal transition)

This construction is more general, and slightly stronger, than usual constructions used in special cases. For example, in mechanism design, players do not take actions, so directness only means that players report information honestly. In correlated equilibria, players take actions but do not send information. Compared to the reduced game implied by the revelation principle of Forges [109], ours is tighter, in two ways:

1. Players cannot send messages that prove that they have not been direct.
2. We completely ignore the part of the game tree that corresponds to the possibility that multiple players may deviate.

[6.8] The *size* of a linear program is the number of nonzero entries in its constraint matrix.

Both of these ingredients are critical to the small size of our reduced game. Without either one, the game would have size that could grow exponentially with $|\mathcal{Z}|$.

The above three conditions are enough for $\hat{\Gamma}_{\text{red}}$ to have $|\hat{\mathcal{Z}}| \le |\mathcal{Z}||\Sigma|$ terminal nodes. Indeed, in every terminal node $\hat{z} = (z, \boldsymbol{\tau})$ of $\hat{\Gamma}_{\text{red}}$, there must be at most one i for which $\tau_i \ne \sigma_i(z)$, since the third condition ensures that at most one player has deviated. Moreover, by the first two conditions above, τ_i must coincide with *some* sequence $\sigma_i \in \Sigma_i$ (where $\sigma_i \ne \sigma_i(z)$). Therefore, $\hat{z}$ can be uniquely identified by specifying a terminal node $z \in Z$ and a deviation sequence $\sigma \in \Sigma$, which is what we needed.

We now show that an optimal direct equilibrium $\boldsymbol{\xi}$ of $\hat{\Gamma}_{\text{red}}$ indeed gives an optimal direct equilibrium in the full mediator-augmented game $\hat{\Gamma}$. We show both directions.

To construct a direct equilibrium in $\hat{\Gamma}$ from $\boldsymbol{\xi}$, we must specify how the mediator acts upon receiving a message m that does not appear in $\hat{\Gamma}_{\text{red}}$. This is simple: the mediator treats such messages the same as $\bot$. The resulting mediator strategy yields a direct equilibrium in $\hat{\Gamma}$, because the only way for a player to deviate in $\hat{\Gamma}$ that does not already exist in $\hat{\Gamma}_{\text{red}}$ is to send such a message m. By the Miranda condition, the player can already send $\bot$ whenever such a message m exists, so the player gains no advantage from sending m compared to sending $\bot$.

To see that $\boldsymbol{\xi}$ is optimal among direct equilibria, consider any other direct equilibrium $\boldsymbol{\xi}'$ in $\hat{\Gamma}$. By construction, the strategy $\boldsymbol{\xi}'$ is a valid strategy in $\hat{\Gamma}_{\text{red}}$, except that $\boldsymbol{\xi}'$ may not always respond $\bot$ when a player sends $\bot$. Then the restriction of $\boldsymbol{\xi}'$ to $\hat{\Gamma}_{\text{red}}$ must also be a direct equilibrium in $\hat{\Gamma}_{\text{red}}$, because players have only less ability to deviate in $\hat{\Gamma}_{\text{red}}$ than in $\hat{\Gamma}$.

We now claim that an optimal direct $\hat{\Gamma}_{\text{red}}$-communication equilibrium can be computed in polynomial time. To see this, we first write the following program, which directly encodes the desired problem.

$$\begin{aligned} \max_{\boldsymbol{\xi} \in \hat{X}_0} \quad & \sum_{\hat{z}=(z,\boldsymbol{\tau}) \in \hat{\mathcal{Z}}} u(z) p(\hat{z}) \cdot \boldsymbol{\xi}(\hat{z}) \prod_{i \in [n]} \hat{\boldsymbol{o}}_i(\hat{z}) \\ \text{s.t.} \quad & \max_{\hat{\boldsymbol{y}}_i \in \hat{Y}_i} \sum_{\hat{z}=(z,\boldsymbol{\tau}) \in \hat{\mathcal{Z}}} u_i(z) p(z) \cdot \boldsymbol{\xi}(\hat{z}) (\hat{\boldsymbol{y}}_i(\hat{z}) - \hat{\boldsymbol{o}}_i(\hat{z})) \prod_{j \ne i} \hat{\boldsymbol{o}}_j(\hat{z}) \le 0 \quad \forall i \in [n] \end{aligned} \tag{6.1}$$

where $\hat{Y}_i$ is the realization-form representation of player i's decision space when player i is deviating, that is, the decision space given by the deviator's information partition $\hat{\mathcal{J}}_i$. Since $\hat{\boldsymbol{o}}_i$ is a constant in the program, the objective function is linear in $\boldsymbol{\xi}$ and the constraints are bilinear in $\boldsymbol{\xi}$ and $\hat{\boldsymbol{y}}_i$. That is, Program (6.1) can be written in the form

$$\max_{\boldsymbol{\xi}:\mathbf{F}_0\boldsymbol{\xi}=\boldsymbol{f}_0, \boldsymbol{\xi} \ge 0} \boldsymbol{g}^\top \boldsymbol{\xi} \quad \text{s.t.} \quad \max_{\hat{\boldsymbol{y}}_i:\mathbf{F}_i\hat{\boldsymbol{y}}_i=\boldsymbol{f}_i, \hat{\boldsymbol{y}}_i \ge 0} \boldsymbol{\xi}^\top \mathbf{A}_i \hat{\boldsymbol{y}}_i \le 0 \ \forall i \in [n] \tag{6.2}$$

where $\boldsymbol{g}$ and the $\mathbf{A}_i$s are appropriately chosen to match (6.1), and $\hat{Y}_i = \{\hat{\boldsymbol{y}}_i \in \mathbb{R}^{\hat{\Sigma}_i} : \mathbf{F}_i \hat{\boldsymbol{y}}_i = \boldsymbol{f}_i, \hat{\boldsymbol{y}}_i \ge 0\}$.

Dualizing the inner maximizations, we have the following linear program.

$$\begin{aligned} \max\ & \boldsymbol{g}^\top \boldsymbol{\xi} \\ \text{s.t.}\ & \mathbf{F}_i^\top \boldsymbol{v}_i \ge \mathbf{A}_i^\top \boldsymbol{\xi}, \ \ \boldsymbol{f}_i^\top \boldsymbol{v}_i \le 0 \ \ \forall i \in [n] \\ & \mathbf{F}_0 \boldsymbol{\xi} = \boldsymbol{f}_0, \ \ \boldsymbol{\xi} \ge 0 \end{aligned} \tag{6.3}$$

It remains only to count the number of nonzero entries in the constraint matrices.

- For each player i, the matrix $\mathbf{A}_i$ contains (at most) one nonzero entry for every $\hat{z} \in \hat{\mathcal{Z}}$ with $\hat{\boldsymbol{y}}_i(\hat{z}) - \hat{\boldsymbol{o}}_i(\hat{z})$, *i.e.*, every $\hat{z}$ in which player i has deviated. Since only one player has deviated, the total number of nonzeros across all the $\mathbf{A}_i$s must therefore be at most $|\hat{\mathcal{Z}}|$.
- Each player's constraint matrix $\mathbf{F}_i$ contains $O(|\hat{\Sigma}_i|)$ nonzero entries. We claim that $\sum_i |\hat{\Sigma}_i| \le O(|\hat{\mathcal{Z}}|)$. To see this, note that each player's sequence set $\hat{\Sigma}_i$ consists of two types of sequences: those in which player i has not deviated, and those in which player i has deviated. We count the types separately. For the former type, there are $|\Sigma_i|$. For the latter type, since only one player deviates, there are at most $O(|\hat{\mathcal{Z}}|)$ total across all players. Thus, the total number of nonzeros is at most $\sum_i |\Sigma_i| + O(|\hat{\mathcal{Z}}|) \le O(|\hat{\mathcal{Z}}|)$ across all players.
- The mediator's constraint matrix $\mathbf{F}_0$ has M nonzeros, by assumption.

Thus, the total size of the LP is at most $O(|\hat{\mathcal{Z}}|) + M \le O(|\mathcal{Z}||\Sigma| + M)$. □

6.4.1 Discussion

When $\mathcal{P} \ne \emptyset$, we can use the same argument; however, the mediator will in general have imperfect recall. Therefore, the constraint matrix $\mathbf{F}_0$ representing the mediator's decision space will in general have exponential size, unless $\mathsf{P} = \mathsf{NP}$ [63]. This exponential blowup is unavoidable in general: indeed, computing an optimal extensive-form correlated equilibrium, which we will later see is a special case of our framework with $\mathcal{P} = \mathcal{I}$, is NP-complete [291].

When $\mathcal{J} \ne \mathcal{I}$, in general, direct equilibria may not exist: indeed, consider a simple one-player game in which nature flips a coin and the player then guesses the coin without observing it. If the player does not observe the coin in $\mathcal{I}$ but does observe the coin in $\mathcal{J}$, then the mediator cannot recommend the player to guess correctly, but a deviating player can guess correctly, so there will always be a profitable deviation. In such a case, *if* there is a direct equilibrium, then solving (6.3) will produce an optimal direct equilibrium; otherwise, (6.3) will be infeasible. A simple additional condition guarantees the existence of (direct) equilibria. Whenever a player i has a nontrivial decision to make, we will insist that it does not know in $\mathcal{J}_i$ than it would know in $\mathcal{I}_i$. That is, for all $I \in \mathcal{I}_i$, if $|\mathcal{A}(I)| > 1$, then $I \in \mathcal{J}_i$. If $\mathcal{J}$ satisfies this condition, we will say that it is *simple*. If $\mathcal{J}$ is simple, then in particular all Nash equilibria are direct $(S, R, \mathcal{A}, \mathcal{P}, \mathcal{J})$-communication equilibria (for any $S, R, \mathcal{A}$, and $\mathcal{P}$).[6.10]

[6.9] The Miranda condition ensures that the player will still always have at least one legal message to send.

[6.10] The condition $|\mathcal{A}(I)| > 1$—or, indeed, the mere concept that what happens at trivial information sets could be

6.5 The Revelation Principle

Theorem 6.4 recovers an optimal *direct* equilibrium. However, in the general case, there may be indirect equilibria that are better (with respect to the objective u) than the best direct equilibrium. In this section, we state and prove a revelation principle for our setting, that will allow us to ensure that, under reasonable conditions, the optimal direct equilibrium recovered by solving Program (6.3) in fact *cannot* be improved upon by an indirect equilibrium.

To do this, we introduce a condition on the communication protocols S and R. The condition we will need is a generalization of the *nested range condition* (NRC) [134]. Intuitively, the nested range condition states that the honest message I is the *most informative* message that a player could possibly send about its information. We will need the analogue of this condition for action recommendations as well. Informally, that analogue states that the direct action recommendation a is the *most restricting* message that the mediator could send that still allows the player to play a.

Definition 6.5 (Nested Range Condition for Generalized Communication Equilibria)**.** A communication protocol $(S, R, \mathcal{A})$ satisfies the *nested range condition* if:

1. (*NRC for S*) Whenever a player at infoset I can lie about its information by sending $I' \neq I$, it can also lie at I by sending any message that it could have sent at I'. Formally, if $I' \in S(I)$ and $m \in S(I)$ then $m \in S(I')$.
2. (*NRC for A*) Whenever a player at infoset I who receives a direct recommendation to play action $a \in \mathcal{A}(I')$ (possibly $I' \neq I$!) may instead play an action $a' \in \mathcal{A}(I)$, any other message $m \in R(I')$ that would allow a *direct* player to play a in I' must also allow the player to play a' in I. Formally, if $a' \in \mathcal{A}(I, a)$, $m \in R(I)$, and $a \in \mathcal{A}(I', m)$, then $a' \in \mathcal{A}(I, m)$.

The nested range condition can be visualized by viewing the communication protocol $(S, R, \mathcal{A})$ as a graph. Let $G(S)$ be the directed graph whose node set is M, and for which there is an edge $I \to m$ if $m \in S(I)$. Similarly, let $G(R, \mathcal{A})$ be the directed graph whose node set is M and for which there is an edge $m \to a$ if $a \in \mathcal{A}(I, m)$. Then, the nested range condition states that:

1. $G(S)$ is transitive, in the sense that for every path $I \to I' \to m$ in $G(S)$ there is an edge $I \to m$, and
2. $G(R, \mathcal{A})$ is almost transitive, in the sense that for every path $m \to a \to a'$ in $G(R, \mathcal{A})$ where $m \in R(I')$ and $a \in \mathcal{A}(I')$, there is an edge $m \to a'$.

This is visualized in Figure 6.1. The condition required for R and A to satisfy NRC is in some sense *global* rather than *local*: we require it to hold even when a and a' are actions at different infosets, so long as $m \in R(I')$. This is because, for the revelation principle to hold, the mediator must always be able to send an action recommendation a to a player in lieu of any other message m, without fear that sending a would give the player more leeway to play a different action than

relevant—may seem strange. However, in our framework, "trivial" information sets have an effect: they induce a round of communication with the mediator. Thus, if $|\mathcal{A}(I)| > 1$ and $I \notin \mathcal{J}_i$, that means that player i at infoset I could, at nodes $h \in I$ condition its message to the mediator on the information set $J \in \mathcal{J}_I$ containing h, which is a strict subset of I.

$$\begin{tikzcd}
I \arrow[r] \arrow[dr, dashed, bend right] \arrow[loop left] & J \arrow[d] \arrow[loop right] \\
& m
\end{tikzcd}$$

$$\begin{tikzcd}
R(J) \ni m \arrow[d] \arrow[dr, dashed, bend left] & \\
a \arrow[r] \arrow[loop left] \arrow[d, phantom, sloped, "\in"] & a' \arrow[loop right] \\
\mathcal{A}(J) &
\end{tikzcd}$$

Figure 6.1: *A visualization of the nested range condition as transitivity of the graphs $G(S)$ (left) and $G(R, A)$ (right). In both diagrams, if the solid edges exist, then NRC states that the dashed edge must exist as well. The self-loops always exist because the conditions in Section 6.3.1 demand that players always be allowed to report honestly and obey recommendations. In the diagram on the right, a' need not be in $\mathcal{A}(J)$.*

sending m would—even if the player reported false information and therefore the action a is at the completely wrong information set!

We now state the main result of this section.

Theorem 6.6 (Revelation principle)**.** *Let $(S, R, \mathcal{A})$ be a communication protocol satisfying NRC. Then, for every $\mathcal{P}$, every $(S, R, \mathcal{A}, \mathcal{P}, \mathcal{J})$-communication equilibrium is outcome-equivalent to a direct $(S, R, \mathcal{A}, \mathcal{P}, \mathcal{J})$-communication equilibrium.*

The above result is a generalization of the revelation principles commonly seen in the various special cases that we will later discuss, and the proof follows the usual structure of revelation principle proofs.

We dedicate the rest of this section to proving Theorem 6.6. For this proof, we will use the notation $a = x(S)$ to denote the action taken by pure strategy x given information S. If x is a mixed strategy, the player replaces x with a sampled pure strategy from from the mixed strategy at the beginning of the game. If x does not specify an action at S, $\boldsymbol{x}(S)$ is set arbitrarily.

Given an equilibrium $(\boldsymbol{\xi}, \hat{\boldsymbol{x}})$ in $\hat{\Gamma}$, we first define *translators* for each player i, that work as follows. The translator operates as an intermediary between player i and the mediator. It maintains separate transcripts with the mediator (τ_i) and the player (π_i). When i sends an infoset I to the translator, the translator samples a message $s = \hat{x}_i[\tau_i, I]$. That is, the translator assumes that the player reported honestly, and samples the message s that the player would have sent if playing according to $\hat{\boldsymbol{x}}_i$. If the player sends a message that is not an infoset, the translator sets $s = \perp$. The message s is sent to the mediator. When the mediator sends a message r to the translator, the translator computes $a = \hat{x}_i[\tau_i sr, I]$, and sends a to the player as an action recommendation, unless $\sigma_i(I') \neq \pi_i$, in which case the translator sets $a = \perp$ instead. The translator then updates the transcripts, by setting $\tau_i \leftarrow \tau_i sr$ and $\pi_i \leftarrow \pi_i I' a$. A visualization of the translator is given in Figure 6.2. The key to the proof is showing that *both the mediator and player i can implement player i's translator*.

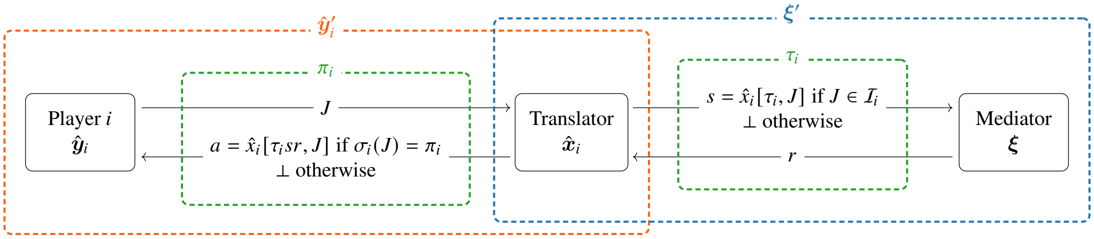


Figure 6.2: *A visualization of the translator used in the proof of the revelation principle. The message J reported by the (indirect) player $\hat{\boldsymbol{y}}_i$ may not be its true information (indeed, it may not be an infoset at all). In the direct equilibrium $\boldsymbol{\xi}'$, the mediator takes the role of the translator, becoming the entire shaded box. If player i has a profitable deviation $\hat{\boldsymbol{y}}_i$ against $\boldsymbol{\xi}'$, giving the player the role of the translator (dashed box) yields a profitable deviation $\hat{\boldsymbol{y}}_i'$ against the original mediator.*

First, let us give the mediator the role of implementing the translator, thereby creating another mediator strategy $\boldsymbol{\xi}'$. The construction of the mediator information structure via $\mathcal{P}$ ensures that $\boldsymbol{\xi}'$ is able to compute τ_i and $\tilde{\boldsymbol{\tau}}_{-i}$ from its current memory $(\pi_i, \tilde{\boldsymbol{\pi}}_{-i})$. Therefore, the mediator can still compute the messages $\boldsymbol{\xi}(\tau_i, \tilde{\boldsymbol{\tau}}_{-i})$ that it should send to to the players. Thus, $\boldsymbol{\xi}'$ is indeed a valid mediator strategy, and $(\boldsymbol{\xi}', \hat{\boldsymbol{o}})$, where $\hat{\boldsymbol{o}}$ is any[6.11] direct strategy profile of the players, is outcome-equivalent to $(\boldsymbol{\xi}, \hat{\boldsymbol{x}})$.

It remains only to show that $(\boldsymbol{\xi}', \hat{\boldsymbol{o}})$ is an equilibrium. To see this, suppose for contradiction that some player i had a profitable deviation $\hat{\boldsymbol{y}}_i \in \hat{Y}_i$ against $(\boldsymbol{\xi}', \hat{\boldsymbol{o}}_{-i})$. We will construct a profitable deviation $\hat{\boldsymbol{y}}_i'$ for player i against the original equilibrium $(\boldsymbol{\xi}, \hat{\boldsymbol{x}})$, by allowing the *player* to implement its own translator. To verify that this is possible, we need to check the following.

1. If $\hat{\boldsymbol{y}}_i$ specifies sending false information $I' \neq I$ at infoset I, then the translator would send $s = \boldsymbol{x}_i(\tau_i, I')$. The nested range condition on S is sufficient for this to be legal for $\hat{\boldsymbol{y}}_i'$: we have $I' \in S(I)$ and $s \in S_{I'}$, and NRC guarantees that $s \in S(I)$. If $\hat{\boldsymbol{y}}_i$ specifies sending a message m at I that is not an infoset, then the translator would send $s = \perp$. This is legal for $\hat{\boldsymbol{y}}_i'$ because of the Miranda condition.

2. If $\hat{\boldsymbol{y}}_i$ at an infoset I specifies playing an action a' when it receives action recommendation a after sending (possibly false) information I', then a' must be a legal action to play after any message $r \in R(I')$ from which the translator can output $a = \hat{\boldsymbol{x}}_i(\tau_i, I')$. The nested range condition on R is sufficient for this to be legal for $\hat{\boldsymbol{y}}_i'$: we have $a \in \mathcal{A}(I', r)$ and $a' \in \mathcal{A}(I, a)$, and NRC guarantees $a' \in \mathcal{A}(I, r)$.

Therefore, $\hat{\boldsymbol{y}}_i'$ achieves the same outcome distribution against $(\boldsymbol{\xi}, \hat{\boldsymbol{x}}_{-i})$ that $\hat{\boldsymbol{y}}_i$ achieves against $(\boldsymbol{\xi}', \hat{\boldsymbol{o}}_{-i})$, so in particular $\hat{\boldsymbol{x}}_i''$ is a profitable deviation against the equilibrium $(\boldsymbol{\xi}', \hat{\boldsymbol{o}})$. This is a

[6.11] Since we are working with the non-reduced game for the moment, there is technically not a unique direct strategy for any given player, since the directness constraints do not specify how the player should act upon receiving a recommendation that is not an action. However, for this proof, that is a meaningless distinction, since $\boldsymbol{\xi}'$ never sends messages that are not actions.

contradiction, so the proof is complete.

Theorems 6.4 and 6.6 together give the following immediate corollary:

Corollary 6.7. *If $(S, R, \mathcal{A})$ satisfies NRC and $\mathcal{J}$ is simple, then an optimal $(S, R, \mathcal{A}, \varnothing, \mathcal{J})$-communication equilibrium can be computed by solving a linear program of size $O(|\mathcal{Z}||\Sigma|)$.*

6.6 Optimal Correlated Equilibria

We now take a deep dive into notions of correlated equilibrium, through the lens of the communication game framework that we have introduced. The problem of computing *one* EFCE (and, therefore, one NFCCE/EFCCE) can be solved in polynomial time in the size of the game tree [153] via a variation of the *ellipsoid against hope* algorithm [159, 237]. Moreover, there exist decentralized no-regret learning dynamics guaranteeing that the empirical frequency of play after T rounds is an $O(1/\sqrt{T})$-approximate EFCE with high probability, and an EFCE almost surely in the limit [106]. Using regret minimizers to play large multi-player games has already led to superhuman practical performance in multi-player poker [39]. However, the problem of computing an *optimal* correlated equilibrium is NP-complete [291], and it is this latter problem that is of interest to us in this section.

Contributions and structure. This section makes a number of contributions related to the computation of *optimal* (*i.e.*, one that maximizes a given linear objective function, such as social welfare or any weighted sum of expected player utilities) NFCCE, EFCCE, and EFCE in general multi-player general-sum extensive-form games.

In particular, we show how our techniques for team games from Chapter 4 can be used to give new parameterized algorithms for computing optimal correlated equilibria. Specifically, we give bounds on the size of the public states of mediator-augmented games for each of the solution concepts as a function of the depth d, the maximum branching factor b, and a suitably-defined *information-complexity* k of the input game that is independent of the solution concept. However, our overall complexity bounds are different depending on the solution concept: the bound for NFCCE in particular does not depend exponentially on the depth of the game, whereas the bounds for EFCCE and EFCE do. We show that this difference is inherent, therefore contributing new complexity-theoretic separations between the solution concepts.

- We show that an optimal EFCE in an extensive-form game can be computed by solving a linear program of size $O^*((bd)^k)$, where the notation O^* suppresses factors polynomial in the size of the game (Theorem 6.15). For optimal EFCCE and optimal NFCCE, we establish bounds of $O^*((b + d - 1)^k)$ and $O^*((b + 1)^k)$, respectively.
- In games with *public player actions*, we show that the bounds for NFCCE and EFCCE can be further improved to $O^*(3^k)$ and $O^*(d^k)$, respectively (Theorem 6.17). We show that the bound for EFCE *cannot* be improved in this manner.

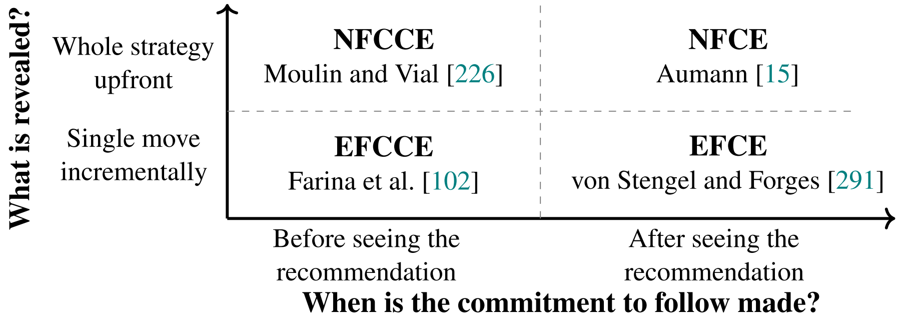

What is revealed?
Whole strategy upfront
Single move incrementally
NFCCE
Moulin and Vial [226]
NFCE
Aumann [15]
EFCCE
Farina et al. [102]
EFCE
von Stengel and Forges [291]
Before seeing the recommendation
After seeing the recommendation
When is the commitment to follow made?

Figure 6.3: *Comparison of different notions of correlation in extensive-form games.*

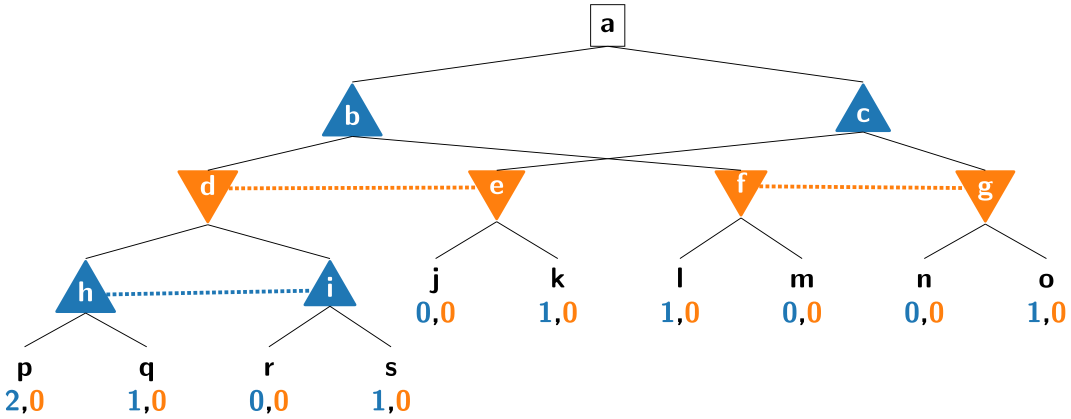


Figure 6.4: *An example game, between two players ▲ (P1) and ▼ (P2). The root node is a chance node, at which chance moves uniformly at random. Dotted lines connect nodes in the same information set. Bold lowercase letters are the names of nodes. We will refer to infosets by naming all the nodes within them; for example,* **b** *and* **de** *are infosets. At terminal nodes, the utility of ▲ is listed below the name of the node. ▼ has utility zero at every terminal node, and in this game the only role of ▼ is to incentivize ▲ to act in a certain way.*

- In *two-player* games with *public chance actions*, our algorithm runs in polynomial time (Theorem 6.19) for all three solution concepts. The problem in this setting had already been shown to be solvable in polynomial time using a different technique by Farina and Sandholm [97]; we match their results and discuss the relationship between our algorithm and theirs in Section 6.6.5.1.

- We show that the gap between the NFCCE bound and the EFCCE and EFCE bounds is fundamental. Matching the bound for NFCCE—in particular, removing the dependence on d—is impossible for EFCCE and EFCE under standard complexity assumptions, demonstrating a *fundamental complexity-theoretic gap* for coarse correlation between normal and extensive form (Theorem 6.23).

6.6.1 Example of Solution Concepts

In this section, we give an example that illustrates the difference between NFCCE, EFCCE, and EFCE. Consider the extensive-form game in Figure 6.4. The game has two players ($n = 2$), whose nodes are pictorially marked with ▲ for Player 1 and ▼ for Player 2 respectively, and 19 nodes (denoted **a** through **s**), of which nine (**a** through **i**) are nonterminal. The root node is a chance node, at which the chance player moves uniformly at random. Player 1 (▲) observes the outcome of the chance node, and can pick between a left or a right action. Player 2 (▼) however does *not* observe the outcome of the chance node; rather, the player only observes the choice of Player 1. This imperfect knowledge of the state is encoded by the information partition $\mathcal{I}_2$ of Player 2, which contains the two information sets $\{\{\mathbf{d}, \mathbf{e}\}, \{\mathbf{f}, \mathbf{g}\}\}$, denoted in the figure with dotted lines connecting the nodes in the same information set. If the game hits state **d**, then Player 1 (▲) gets to play a second move. However, Player 1 will not observe the action chosen by Player 2 at **d**; this is captured again by the information set $\{\mathbf{h}, \mathbf{i}\}$. Nodes **b** and **c** do not bear any uncertainty, and are therefore singleton elements in their corresponding information sets. In summary, the information partitions of the players are $\mathcal{I}_1 = \{\{\mathbf{b}\}, \{\mathbf{c}\}, \{\mathbf{h}, \mathbf{i}\}\}$ and $\mathcal{I}_2 = \{\{\mathbf{d}, \mathbf{e}\}, \{\mathbf{f}, \mathbf{g}\}\}$. At terminal nodes, the payoffs for ▲, ▼ are listed below the node. ▼ has utility zero at every terminal node.

This game represents a signalling game between two players, ▲ and ▼. ▼ has no rewards and will therefore never have incentives to deviate from recommendations. ▲ scores a point if ▼ plays the same action as chance played at the root, but chance's action is only privately revealed to ▲, so ▼ relies on ▲ to signal the chance action through ▲'s own action. ▲ also has the opportunity to receive a bonus point for guessing ▼'s action in case **d** is reached.

We will refer to the pure profiles in this game using the notation $\boxed{\mathbf{bcdfh}}$, where the letters indicate which actions were played at the respective infosets containing those nodes. For example, $\boxed{\mathsf{LRLRL}}$ means that ▲ plays left at **b**, right at **c**, and left at infoset **hi**; while ▼ plays left at **de** and right at **fg**—in particular, ▲ copies chance, and ▼ copies ▲. If ▲ plays right at **b**, we leave ▲'s action at **hi** unspecified since it is irrelevant; for example, $\boxed{\mathsf{RLRL}}$ is a valid pure strategy.

We make the following observations about our example game.

- The correlated profile $\mu_1 := \frac{1}{2}\boxed{\mathsf{LRLRR}} + \frac{1}{2}\boxed{\mathsf{RLRL}}$ is an NFCCE: ▲ is getting utility 1, which is larger than any utility it can get by unilaterally deviating without seeing any recommendations: since ▼'s marginal strategy is uniform random, a best unilateral deviation for ▲ is to always play left, securing expected utility 3/4. However, μ_1 is not an EFCCE, because ▲ can profitably deviate at trigger **hi** by playing left instead of right. This deviation cannot be expressed as an NFCCE deviation, because it requires ▲ to follow recommendations at **b** and **c**.
- The correlated profile $\mu_2 := \frac{1}{2}\boxed{\mathsf{LRLRL}} + \frac{1}{2}\boxed{\mathsf{RRRR}}$ is an EFCCE. ▲ still gets total expected utility 1. ▲ is already getting the optimal utility at **c** and **hi**; and at **b**, ▲ is currently getting a conditional utility of 1, and she cannot improve upon this without seeing the recommendation at **b**. However, μ_2 is not an EFCE, because ▲ can profitably deviate upon being recommended to play right at **b** by instead playing left at **b** and right at **hi**. This deviation cannot be expressed as an EFCCE deviation, because, in the deviation, ▲

conditions her action at infoset **hi** on the recommendation that she received at **b**.

- The pure profile $\boxed{\textbf{LRLRL}}$ is an EFCE (in fact, being uncorrelated, it is a Nash equilibrium).

6.6.2 Mediator-Augmented Games for Correlated Equilibria

Notions of correlation in extensive-form games are recovered from our mediator-augmented framework by forcing all information to be private, and forcing players to disclose information. That is, $\mathcal{P} = [n]$. We then create different notions of correlated equilibrium by varying the communication protocol $(S, R, \mathcal{A})$.

1. To recover *extensive-form correlated equilibrium* (EFCE) [291], set $S(I) = \{I, \perp\}$ and $\mathcal{A}(I, m) = \mathcal{A}(I)$ for all m. That is, the player is allowed to play any action.
2. To recover *extensive-form coarse correlated equilibrium* (EFCCE) [102], set[6.12] $S(I) = \{I, \perp\}$ and $\mathcal{A}(I, a) = \{a\}$ for all $a \in \mathcal{A}(I)$. That is, players are forced to obey any recommendation they see, but may still choose not to see a recommendation.
3. To recover *normal-form coarse correlated equilibrium* (NFCCE) [226], add n dummy nodes $d_1, \dots, d_n$ to the top of the game tree. Node d_i belongs to player i, and has a single legal action that leads to node d_{i+1}. Node d_1 is the root of the modified game tree, and d_{n+1} is the root of the original game tree. These nodes serve the purpose of forcing a round of communication with the mediator before the game begins. Now set $S(\{d_i\}) = \{\{d_i\}, \perp\}$ for every player i, $S(I) = \{I\}$ for all other infosets $I \succ d_n$, and $\mathcal{A}(I, a) = \{a\}$ for all $a \in \mathcal{A}(I)$ and all I. That is, players are only allowed to decide whether to follow the mediator at the root of the game (send $\{d_i\}$), or completely play on their own (send $\perp$).

The choice to use *mixed* strategies rather than *behavioral* strategies as the set of valid strategies for the mediator is what allows our notion to recover *correlated* equilibria instead of merely Nash equilibria.

Unlike all the other special cases we will discuss, correlated equilibria involve an imperfect-recall mediator—that is, $\mathcal{P} \neq \emptyset$. As mentioned before, this is unsurprising, because optimal correlated equilibria are hard to compute. In this light, our results could be interpreted as reducing the problem of computing optimal correlated equilibria to the problem of representing the strategy space of an imperfect-recall player (the mediator).

Our methods do *not* encompass stronger notions of correlated equilibrium such as the *autonomous correlated equilibrium* (ACE) [109, 291] or *normal-form correlated equilibrium* (NFCE) [15]. This is because the revelation principle in those settings would require the mediator to recommend to the player an action at every infoset that the player could possibly have reached, breaking the "local" nature of our reduced game and revelation principle. Indeed, the computational complexity of computing even *one* equilibrium in an extensive-form game remains an open problem for both ACE and NFCE.

[6.12] Here, we do not need to specify what $R_I(m)$ is for $m \notin \mathcal{A}(I)$: this is pointless, because the revelation principle (Theorem 6.6) holds and players cannot lie, so without loss of generality the mediator's recommendations can be assumed to always be actual actions $a \in \mathcal{A}(I)$

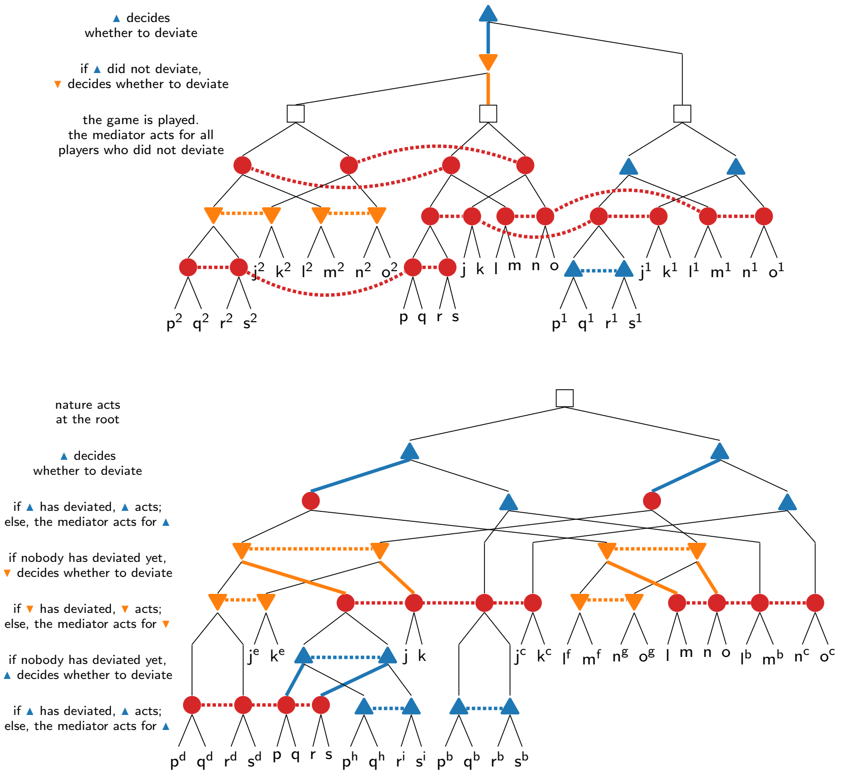

▲ decides whether to deviate
if ▲ did not deviate, ▼ decides whether to deviate
the game is played. the mediator acts for all players who did not deviate
j^2 k^2 l^2 m^2 n^2 o^2 j k l m n o j^1 k^1 l^1 m^1 n^1 o^1
p^2 q^2 r^2 s^2 p q r s p^1 q^1 r^1 s^1
nature acts at the root
▲ decides whether to deviate
if ▲ has deviated, ▲ acts; else, the mediator acts for ▲
if nobody has deviated yet, ▼ decides whether to deviate
if ▼ has deviated, ▼ acts; else, the mediator acts for ▼
if nobody has deviated yet, ▲ decides whether to deviate
if ▲ has deviated, ▲ acts; else, the mediator acts for ▲
j^e k^e j k j^c k^c l^f m^f n^g o^g l m n o l^b m^b n^c o^c
p^d q^d r^d s^d p q r s p^h q^h r^i s^i p^b q^b r^b s^b

Figure 6.5: *Augmented games Γ^c for NFCCE (top) and EFCCE (bottom), where Γ is the example game in Figure 6.4. The obedient strategies $\boldsymbol{o}_1, \boldsymbol{o}_2$ are given by the thick colored lines below ▲ and ▼'s decision points. Red circles denote decision points of the mediator. Augmented histories are labeled as h^τ, where h is the true node and τ is the transcript of the deviating player. If no superscript is present, there is no deviating player. For cleanliness, τ is abbreviated in all three diagrams. For NFCCE, τ is the player i who deviated—for example, p^2 means terminal node p was reached, but ▼ deviated. For EFCCE, τ is the node at which the player deviated—for example, p^{d} means terminal node p was reached, but ▼ deviated at node* **d**.

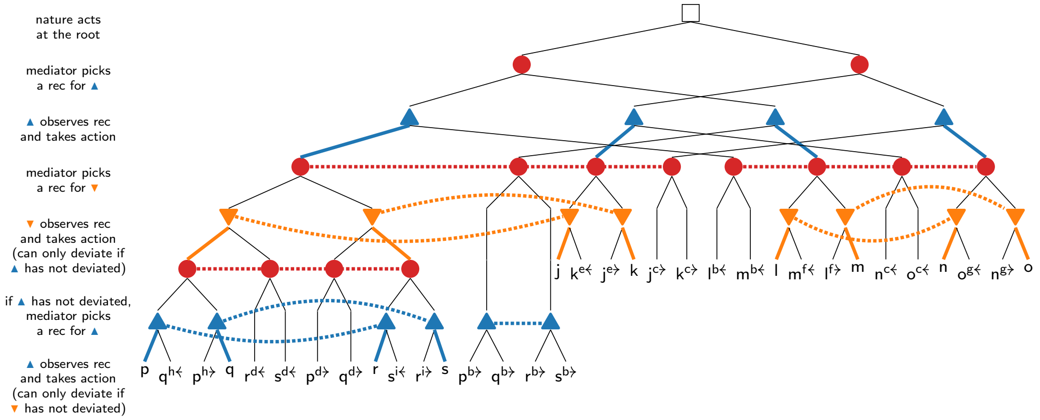


Figure 6.6: *Augmented game Γ^c for EFCE, where Γ is the example game in Figure 6.4. Notation is as in Figure 6.5. The trigger τ is the node at which the player deviated, followed by the recommendation (↚ or ↛) given to the player at that node—for example, $\mathsf{q}^{\mathsf{h}↚}$ means terminal node q was reached but ▲ deviated after being recommended to play ↚ at* **h**.

For this section only, it will be important to distinguish the different augmented games. Therefore, instead of using hats, we will use the notation Γ^c , where $c \in \{\text{NFCCE}, \text{EFCCE}, \text{EFCE}\}$, to denote the augmented game for solution concept c. Similarly, superscripts c will denote generic components, strategies, ‖within that augmented game, *e.g.*, $\boldsymbol{x}_i^c$ is a strategy for player i.

For concreteness, Figures 6.5 and 6.6 depicts the augmented games derived from the example of Figure 6.4 for the three solution concepts of interest.

6.6.3 Comparison to Relevant Sequence-Based Construction of Ξ

Our construction via the mediator-augmented game uses a vector $\boldsymbol{\xi} \in \Xi^c$ to represent a correlated profile. It is instructive to compare this representation to other representations of correlated profiles, in particular, the *correlation plan* defined and used by von Stengel and Forges [291]. In this section, we will review the notion of correlation plan defined by that paper, and compare it to our construction.

Definition 6.8. A sequence tuple $(I_1 a_1, \dots, I_n a_n) \in \Sigma_1 \times \cdots \times \Sigma_n$ is *relevant* if there is a history h in Γ such that either $\sigma_i(h) = I_i a_i$ for every player i, or there is a player j—the *deviator*—such that $\sigma_i(h) = I_i a_i$ for all $i \neq j$ and $I_j \preceq h$.

This definition was first proposed by von Stengel and Forges [291] in the two-player case; here, we generalize it to arbitrarily many players. Intuitively, the relevant tuples are those that appear in the linear program defining *any* of the three notions.

Definition 6.9 ([291])**.** For a correlated profile $\mu \in \Delta(X_1 \times \cdots \times X_n)$, the *correlation plan* is the vector $\boldsymbol{\xi} \in \mathbb{R}^\Sigma$ defined by $\boldsymbol{\xi}(\sigma_1, \dots, \sigma_n) = \mathbb{E}_{\boldsymbol{x} \sim \mu} \prod_{i \in [n]} \boldsymbol{x}_i(\sigma_i)$. We denote by Ξ the set of all correlation plans.

von Stengel and Forges [291] go on to show that correlation plans are a sufficient representation for computing (optimal) EFCE, in the sense that, if one could efficiently represent the set of all correlation plans, then one can compute optimal EFCE efficiently. Farina et al. [101, 102] generalizes this observation to NFCCE and EFCCE as well. Our linear program (6.3) achieves the same claim: if Ξ^c is efficiently representable then optimal equilibria in notion c can be computed efficiently. One may wonder, therefore, about the relationship between the two.

It turns out that each of our Ξ^c polytopes is in some sense merely a sub-vector of Ξ with the indices renamed. That is, there is a natural injectiion from sequences of the mediator in Γ^c to relevant tuples $(\sigma_1, \dots, \sigma_n) \in \Sigma$. A mediator sequence in Γ^c corresponds to some history h^τ. If $\tau = \perp$ then h^τ corresponds to $(\sigma_1(h), \dots, \sigma_n(h))$, that is, $\boldsymbol{\xi}(h^\tau) = \boldsymbol{\xi}(\sigma_1(h), \dots, \sigma_n(h))$; if τ is a nonempty trigger (say, P1 WLOG), then h^τ corresponds to $(\sigma_1(\tau), \sigma_2(h), \dots, \sigma_n(h))$, where $\sigma_1(\tau)$ is the last sequence of player i before τ. By construction of Γ^c, this must be a relevant tuple.

In some sense, Ξ^c is therefore a *refined* notion of correlation plan that is specific to the equilibrium concept c, only requiring the sequence tuples that are relevant for that concept. In the next section, we will show that, in fact, the differences between the various Ξ^cs result in separations in the complexity of representing each polytope, and therefore separations in the complexity of computing optimal equilibria.

6.6.4 Representing Imperfect-Recall Decision Spaces

The key barrier to computing optimal equilibria, in a sense, is that *the mediator in the augmented game has imperfect recall*. We will now describe a methods of overcoming this imperfect recall and thus of arriving at algorithms for computing optimal equilibria, by applying the TB-DAG from Chapter 4.

Definition 6.10. A set $B \subseteq \mathcal{H}$ is a *(reachable) belief* of player i if

1. B contains at least one decision point for player i, that is, $B \cap \mathcal{H}_i \neq \emptyset$
2. there exists a pure strategy $\boldsymbol{x}_i$ for player i such that B is a connected component of $\mathcal{G}_i[\{h \in \mathcal{H} : x_i[h] = 1\}]$ where $\mathcal{G}_i[\cdot]$ denotes an induced connected component of $\mathcal{G}_i$.

We will use $\mathcal{B}_i$ to denote the set of (reachable) beliefs of player i.

Intuitively, beliefs represent sets of nodes that an imperfect-recall player *will always be able to distinguish in the future*: that is, if B is a belief corresponding to pure strategy $\boldsymbol{x}_i$, then, upon reaching the belief B, player i knows that it has reached belief B, and player i knows that it will never forget having reached B. We first show the following useful lemma about the TB-DAG from Chapter 4:

Algorithm 6.7 (CorrelationDAG): Optimal Correlated Equilibria via Correlation DAG

```
1: input: extensive-form game $\Gamma$, desired solution concept $c$, objective $g : \mathcal{Z} \to \mathbb{R}$
2: construct the augmented game $\Gamma^c$
3: compute a polytope representation of the mediator's strategy space, $\Xi^c$, using Proposition 6.11
4: solve the LP (6.3)
5: return $\boldsymbol{\xi}$
```

Proposition 6.11. *There exists a representation of player i's decision space as DAG decision problem with $O^*(R_i)$ entries, where*

$$R_i := \sum_{B \in \mathcal{B}_i} \prod_{\substack{I \in \mathcal{I}_i: \\ I \cap B \neq \varnothing}} |\mathcal{A}(I)| \tag{6.4}$$

Proof. R_i counts, up to a constant factor, the number of nodes in i's TB-DAG. To see that the number of edges is $O^*(R_i)$, notice that each observation point $B\boldsymbol{a}$ has exactly one incoming edge and at most $|B\boldsymbol{a}|$ outgoing edges. □

We use the above result to construct a representation of the mediator's decision space, Ξ^c, in the augmented game Γ^c. We call the representation of Ξ^c using Proposition 6.11 the *correlation DAG* for notion c. Proposition 6.11 immediately gives an algorithm for solving the program (6.3). This algorithm is given in Algorithm 6.7 (CorrelationDAG).

6.6.4.1 Analyzing the Size of the Representation

To analyze the complexity of CorrelationDAG, it suffices to bound the quantity in (6.4). Notationally, we will use R_0^c to denote the quantity R_0 in (6.4) in the augmented game Γ^c. We first introduce some useful definitions.

Definition 6.12. A *public state* is a connected component of $G = G_{[n]}$.

Definition 6.13. Given a node h and a player i, the *last infoset* $I_i(h)$ is the lowest (*i.e.*, most recent) infoset reached by player i on the path to h.

Definition 6.14. The *information complexity* k of an extensive-form game is the greatest number of unique last infosets in any public state. In symbols, $k = \max_{P \in \mathcal{P}} |\{I_i(h) : h \in P, i \in [n]\}|$.

Notice that it is possible for k to be much smaller than $n|P|$, because the set of last infosets may contain duplicates. For example, in normal-form games (converted to extensive form in the canonical manner), we have $k = n$ since each terminal node is a public state and each player has only one infoset. As an example, the information complexity of the game in Figure 6.4 is 3: the public state **de** has three last infosets, namely **b**, **c**, and **de** itself.

Chapter 4 uses the definition of information complexity to bound the representation size of Proposition 6.11.In this section, we show similar bounds in our setting. Note that b and k, as defined above, are the branching factor and information complexity of the *original game* Γ, not of the mediator in Γ^c—therefore, we cannot directly apply Theorem 4.25 to arrive at a bound of $(b+1)^k$. Indeed, the mediator in Γ^c can have much higher information complexity than Γ. Thus, we need to be more careful in our analysis.

Theorem 6.15. *Let k be the information complexity of a timeable game Γ, b be its branching factor, and d be its depth. Then $R_0^{\mathrm{NFCCE}} \le O^*\big((b+1)^k\big)$, $R_0^{\mathrm{EFCCE}} \le O^*\big((b+d-1)^k\big)$, and $R_0^{\mathrm{EFCE}} \le O^*\big((bd)^k\big)$.*

Proof. The expression (6.4) counts the number of pairs $(B, \boldsymbol{a})$ where B is a belief of the mediator Γ^c and $\boldsymbol{a} \in \prod_{I \in \mathcal{I}_0, I \cap B \neq \varnothing} A_I$. Our goal will therefore be to bound this number.

NFCCE: It suffices, for each last-infoset I of P, to specify whether the player (1) does not play to I at all, or (2) plays to I and chooses one of the b actions available therein. There are at most $(b+1)^k$ such choices. Each choice induces a disjoint collection of pairs $(B, \boldsymbol{a})$; that is, surely at most $|P|$ pairs. Thus, $R_0^{\mathrm{NFCCE}} = O^*((b+1)^k)$.

EFCCE: For each of the k last-infosets I at P, we need to specify whether the player played to reach I and then played one of the (at most) b actions available there, or she deviated at one of the (at most) $d-2$ infosets $I' \prec I$. There are at most $b+d-1$ ways to do this, so, by the above argument, we have $R_0^{\mathrm{EFCCE}} = O^*((b+d-1)^k)$.

EFCE: For EFCE, we need to additionally specify which action was recommended at the deviation point, of which there are at most b possibilities, for a total of $b(d-1)+b = bd$ options. Thus, again by the same argument, $R_0^{\mathrm{EFCE}} = O^*((bd)^k)$. □

As an example, consider an extensive-form game of the following form. Chance first samples and privately reveals types $t_i \in [T]$ to each player i. Thereafter, there is no further privacy: all actions by the players and chance after the root are public. By definition, we see that this game is a public-action game, and we have $k = nT$ because each sequence of post-root actions induces a public state with T private states for each of the n players. Thus, Theorem 6.15 gives an algorithm for computing optimal EFCEs that runs in time $\mathrm{poly}(|\mathcal{H}|, (bd)^{nT})$; in particular, if $n = T = O(1)$ then the algorithm runs in polynomial time. To our knowledge, we are the first to give a polynomial-time algorithm for this setting, even when $n = T = 2$.

We now show two settings in which we can improve our bounds from Theorem 6.15. They both depend on certain information being *public*.

6.6.4.2 Public Player Actions

First, we discuss the setting in which *player* actions are public.

Definition 6.16. A game has *public player actions* if, for all public states $P \in \mathcal{P}$ containing at least one non-chance node, for all actions $a \in \bigcup_{h \in P} \mathcal{A}(h)$, the set $\{ha : h \in P, a \in \mathcal{A}(h)\}$ is a union of public states.

Poker, for example, has this structure: the root public state contains only a chance node, and every action thereafter is fully public. In this setting, we can remove the dependencies on b for NFCCE and EFCCE:

Theorem 6.17. *In games with public player actions, $R_0^{\text{NFCCE}} = O^*(3^k)$ and $R_0^{\text{EFCCE}} = O^*(d^k)$.*

Proof. Theorem 4.30 constructs, starting with a game Γ with public actions, a new strategically equivalent game Γ' with branching factor 2, no higher parameter k, and at most polynomially larger size. The method works by breaking up each high-branching-factor node into several successive binary decisions, in such a way that public state size is preserved. For NFCCE, this is sufficient to immediately conclude the desired result. For EFCCE, it suffices to additionally observe that one only needs to care about trigger histories h^τ in Γ' where τ is a valid trigger in Γ. The number of these is at most the depth d of the original game Γ. □

Once again, the bound for NFCCE matches the bound for team gamess, up to polynomial factors. The bound on R_0^{EFCE} cannot be improved in this fashion, for two reasons. First, the $(bd)^k$ term in that analysis comes from counting the number of triggers at a given node, which has not changed. Second, as above, the proof of Theorem 6.17 modifies the original game tree to have lower branching factor. This is an invalid transformation for EFCE, because some EFCE triggers present in the original game would not be expressible in the new game.

6.6.5 Two-Player Games with Public Chance

We now discuss the case where *chance* actions are public. Since it is already NP-hard to compute optimal equilibria in three-player games with no chance nodes [291], we restrict our attention to *two-player* games. Farina and Sandholm [97] showed, via a different construction, that in games with public chance, Ξ has a polynomial-sized representation and therefore optimal NFCCEs, EFCCEs, and EFCEs can be computed in polynomial time. In this section, we show that our correlation DAG matches this bound.

Definition 6.18. A game has *public chance actions* if, for every two nodes h, h' in the same public state, the lowest common ancestor $h \wedge h'$ is not a chance node.

We will assume for the rest of this section that levels in Γ uniquely specify whose move it is—that is, for every level of the game tree, there exists a player i (possibly nature) such that every node in the level is a decision node of player i. Since we have already assumed timeability, this additional assumption is without loss of generality by adding dummy nodes. Most practical games, including the games we use in our experiments, already satisfy this assumption without further modification.

Theorem 6.19. *In two-player timeable games with public chance actions, we have $R_0^c = \operatorname{poly}(|\mathcal{H}|)$ for all three notions c.*

Initially, one may ask whether it is possible to prove this result by directly applying Theorem 6.15. In particular, if it were the case that all two-player games of public chance had constant information

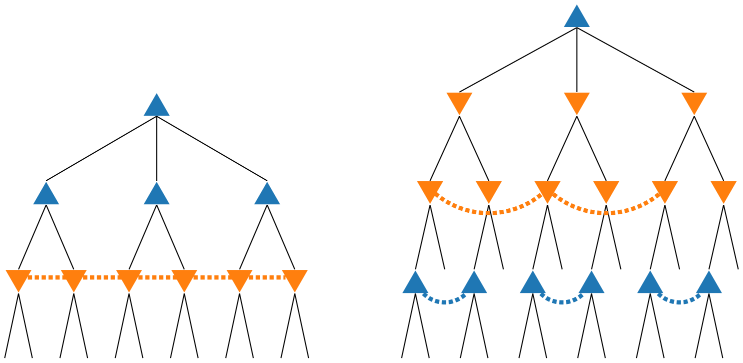


Figure 6.8: *Two examples of two-player extensive-form game trees with no chance moves and large information complexity k. In both examples, k can be increased arbitrarily by increasing the branching factor of the root node. The left example would be easily reparable with a tighter definition of information complexity (that takes into account the fact that only one of the infosets in the second layer is reachable in any pure strategy profile), but the right example is not so easily reparable, and examples such as these are the reason that the proof of Theorem 6.19 is more involved than one may initially expect.*

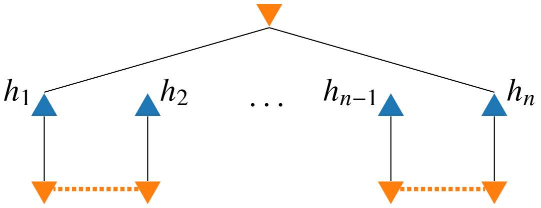


Figure 6.9: *Visualization of the proof of Lemma 6.20 (other nodes, such as the ancestors of h_2, are not shown). Since both ▼ infosets are used to make connections in $\mathcal{G}^c[B]$, they must both be played to by ▼, which is impossible since this would require ▼ to make two different moves at the top node.*

complexity, Theorem 6.19 would follow immediately. Unfortunately, this is not the case: in Figure 6.8, we exhibit two families of two-player extensive-form games with *no* chance actions and information complexity that is linear in the size of the game.

Proof. Let Γ be a two-player game with public chance actions, and call the two players ▲ and ▼. Let c be any of the three solution concepts. For a node h^τ in Γ^c, we will use $\tilde{\sigma}_i(h^\tau)$ to denote the sequence infosets reached and recommendations received by player i has received on the path from the root to h^τ, not including at h itself. $\tilde{\sigma}_i(h^\tau)$ is always a valid player i sequence in Γ. However, it is not the same as player i's sequence $\sigma_i(h^\tau)$: for example, for NFCCE, if player i deviated at h^τ then $\tilde{\sigma}_i(h^\tau) = \varnothing_i$ (because a deviating player receives no recommendations) but player i still sees information sets and actions on the path to h.

Throughout this proof, for notational shorthand, we write $h^\tau \in I$, where I is an infoset of player

i in Γ, if $h \in I$ and τ is not a trigger of player i.

Lemma 6.20. *Let B be a mediator belief in Γ^c, and suppose (WLOG) that the mediator is giving a recommendation to player* 1*. Then there exists a unique information set $I \in \mathcal{I}_\blacktriangle$, and a sequence $\sigma \in \Sigma_\blacktriangledown$, such that:*

- *the mediator only gives a recommendation at information set I: for every $h^\tau \in B$, either $h \in I$ or τ is a trigger of $\blacktriangle$;*
- *for every $h^\tau \in B \cap I$, we have $\tilde{\sigma}_\blacktriangledown(h^\tau) \preceq \sigma$; and*
- *there is an $h^\tau \in B \cap I$ with $\tilde{\sigma}_\blacktriangledown(h^\tau) = \sigma$.*

Further, the map $B \mapsto (I, \sigma)$ is injective.

Proof. Let $h_1^{\tau_1} \in B$ be any decision point for the mediator, and let $I \ni h_1$.

We first claim that there is no other node $h_n^{\tau_n} \in I' \neq I$ and player 1 not having deviated. (See Figure 6.9 for a visual representation of the argument in this paragraph.) Let $h_1^{\tau_1}$—$h_2^{\tau_2}$—$\cdots$—$h_n^{\tau_n}$ be a path through the induced connectivity graph $\mathcal{G}_0^c[B]$. Further, assume WLOG that $h_2^{\tau_2} \notin I$ and $h_{n-1}^{\tau_{n-1}} \notin I'$; otherwise, move $h_1^{\tau_1}$ and $h_n^{\tau_n}$ along the path toward each other until this is true. We first ask: what is $h_1 \wedge h_n$? By definition of public chance, it cannot be a chance node, or else h_1 and h_n could not be in the same public state, much less the same belief. It cannot be a $\blacktriangle$-node, because then the mediator cannot recommend $\blacktriangle$ to play to both h_1 and h_n. It must therefore be a $\blacktriangledown$-node. We now ask: how are $h_1^{\tau_1}$ and $h_2^{\tau_2}$ connected? $h_1^{\tau_1} \in I$ and $h_2^{\tau_2} \notin I$; therefore, h_1 and h_2 must be connected by an infoset at which the mediator recommends to $\blacktriangledown$. Therefore, the mediator must recommend $\blacktriangledown$ to play to h_1. The same applies to $h_n^{\tau_n}$. But this is a contradiction, because it implies that $\blacktriangledown$ must have been recommended two distinct actions at $h_1 \wedge h_n$.

Now let $h_2^{\tau_2} \in B$ with $h_2 \in I$. We claim that either $\tilde{\sigma}_\blacktriangledown(h_1^{\tau_1}) \preceq \tilde{\sigma}_\blacktriangledown(h_2^{\tau_2})$ or $\tilde{\sigma}_\blacktriangledown(h_1^{\tau_1}) \succeq \tilde{\sigma}_\blacktriangledown(h_2^{\tau_2})$. Consider the node $h := h_1 \wedge h_2$. Since $h_1, h_2 \in I$ and I belongs to $\blacktriangle$, h must be a $\blacktriangledown$-node (again, it cannot be a chance node, because chance is public). Let a_1 and a_2 be the actions at h leading to h_1 and h_2 respectively. There are two cases:

- The mediator does not recommend $\blacktriangledown$ to play to h, or recommends an action at h that is neither a_1 nor a_2. Then $\tau_1 = \tau_2 = \sigma_\blacktriangledown(h_1^{\tau_1}) = \sigma_\blacktriangledown(h_1^{\tau_2})$.
- At h, the mediator recommends one of a_1 or a_2 (WLOG, a_1). Then $\tilde{\sigma}_\blacktriangledown(h_2^{\tau_2}) = I(h)a_1 \preceq \tilde{\sigma}_\blacktriangledown(h_1^{\tau_1})$.

Therefore, the set $\{\tilde{\sigma}_2(h^\tau) : h^\tau \in B \cap I\}$ is totally ordered, and so it has a maximum element, which we call σ. Then, by definition, σ satisfies the desired properties.

We therefore have a map $\phi : \mathcal{B}_0^c \to (\mathcal{I}_1 \times \Sigma_2) \sqcup (\mathcal{I}_2 \times \Sigma_1)$ associating each mediator belief to a pair consisting of an infoset of one player and a sequence of the other player.

It remains to show that ϕ is injective. Let $(I, \sigma) \in \mathcal{I}_1 \times \Sigma_2$ (WLOG). Let $h^\tau \in I$ with $\sigma_\blacktriangledown(h^\tau) = \sigma$, and pick an h^τ so that $\tau = \bot$ if one exists. First, suppose $\tau = \bot$. There is only

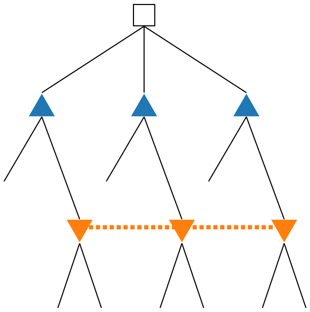


Figure 6.10: *An example of a timeable triangle-free game in which our construction will be exponentially-sized (in the branching factor of the root node). In this game, the algorithm of Farina and Sandholm [97] works by essentially "re-ordering" the game tree so that ▼'s decision point is moved to the root, at which point the chance decision can be treated as public, thereby removing the exponentiality.*

one way to reach $h^\perp$: at every belief B', the mediator must play the action leading to h, and then observe the public observation containing h'. Thus, the belief containing $h^\perp$ must be unique.

If $B \cap I$ contains no trigger-less node, then it contains only nodes with ▼-triggers. But then $B \subseteq I$, because no node with a ▼-trigger can ever be connected to a node with a ▲-trigger, and every node in $B \setminus I$ must have a ▲-trigger because of Lemma 6.20. But this precisely fixes what B is: namely, $B = \{h^\tau \in I : \sigma_{\blacktriangledown}(h^\tau) \preceq \sigma\}$, because all such h^τ must be in B, and Lemma 6.20 states that no others can be. □

Thus, the number of beliefs is polynomial in the size of the game. Since every belief overlaps exactly one mediator information set, it follows that R_0^c is polynomial in the game size. □

6.6.5.1 Discussion: Relationship to Triangle-Freeness

Theorem 6.19 implies that CorrelationDAG runs in polynomial time in two-player games of public chance. As we mentioned, we are not the first to exhibit a polynomial-time algorithm in this setting; Farina and Sandholm [97] has exhibited one using a different technique, namely by showing that the *von Stengel–Forges* (vSF) polytope [291] is tight. It is instructive to compare the two approaches. The approach of Farina and Sandholm [97] carries many similarities to our approach for this special case—in particular, their approach also works by effectively constructing a DAG representation of Ξ^{EFCE}. However, while their approach dynamically chooses which information set to expand next on the fly, our approach uses the fixed ordering provided by the timeable game to decide which information set is "next". When the game is timeable, our approaches give essentially the same representation: indeed, the proof in the previous section shows that there is a decision point of the mediator in Γ^c for every relevant pair (I_1, σ_2) or (σ_1, I_2), which are precisely the branching points in the representation of Farina and Sandholm [97].

Unlike their approach, our correlation DAG algorithm provides an FPT guarantee on any game.

However, it is limited to timeable games, whereas theirs generalizes beyond timeable games to a family they coin *triangle-free games*. Here, for the sake of completeness, we include a definition of triangle-freeness.

Definition 6.21. In a two-player game, two information sets $I_1 \in \mathcal{I}_1$ and $I_2 \in \mathcal{I}_2$ are *connected*, denoted $I_1 \bowtie I_2$, if there exists a node h with $h \succeq I_1$ and $h \succeq I_2$. A *triangle* is a collection of four infosets $I_1, I_1' \in \mathcal{I}_1$ and $I_2, I_2' \in \mathcal{I}_2$ such that $I_1 \bowtie J_1$, $I_2 \bowtie J_2$, and $I_1 \bowtie J_2$.

Intuitively, triangle-freeness is useful because it guarantees the existence of some "branching order" that can be used to fill in the polytope Ξ^{EFCE}. We refer the reader to the paper of Farina and Sandholm [97] for more details. It is not difficult to construct triangle-free games in which our construction would be exponentially-sized; see Figure 6.10. We leave to future research the question of whether it is possible to extend our algorithm so that it is also runs in polynomial time in all triangle-free games, achieving the best of both worlds.

6.6.5.2 Fixed-Parameter Hardness of Representing Ξ^{EFCCE} and Ξ^{EFCE}

A natural question is whether it is possible to achieve the same bound for EFCCE and EFCE as achieved for NFCCE and team games—namely, a construction whose exponential term depends only on b and k. It turns out that our construction does *not* accomplish this, and in fact, *no* representation of Ξ^c for $c =$ EFCCE or $c =$ EFCE can have size $O^*(f(k))$ for any function f under standard complexity assumptions even when $b = 2$. To do this, we first review some fundamental notions of *parameterized complexity*.

Definition 6.22. A *fixed-parameter tractable* (FPT) algorithm for a problem is an algorithm that takes as input an instance x and a *parameter* $k \in \mathbb{N}$, and runs in time $f(k)\mathsf{poly}(|x|)$, where $|x|$ is the bit length of x and $f : \mathbb{N} \to \mathbb{N}$ is an *arbitrary function*.

The k-CLIQUE problem[6.13] is widely conjectured to not admit an FPT algorithm parameterized by the clique size k. In the literature on parameterized complexity, this conjecture is known as $\mathsf{FPT} \neq \mathsf{W}[1]$, and is implied by the exponential time hypothesis [59]. We now show that this conjecture implies lower bounds on the complexity of representing the polytopes Ξ^{EFCCE} and Ξ^{EFCE}.

Theorem 6.23. *Assuming* $\mathsf{FPT} \neq \mathsf{W}$*[1], there is no FPT algorithm for linear optimization over* Ξ^{EFCCE} *or* Ξ^{EFCE} *parameterized by information complexity, even in two-player games with constant branching factor.*

Proof. We reduce from k-CLIQUE. Let $G = (V, E)$ be a graph with n nodes (identified with the positive integers $[n]$), and construct the following two-player game Γ (see also Figure 6.11):

- Chance chooses an integer $j_1 \in [k]$ and tells ▲ but not ▼. Transition to the node $(j_1, 1)$.
- For each $v_1 \in [n+1]$, the node (j_1, v_1) is a decision node for ▲. ▲ may *exit* or *continue*.

[6.13] The k-CLIQUE problem is to decide whether a given graph contains a clique of size at least k.

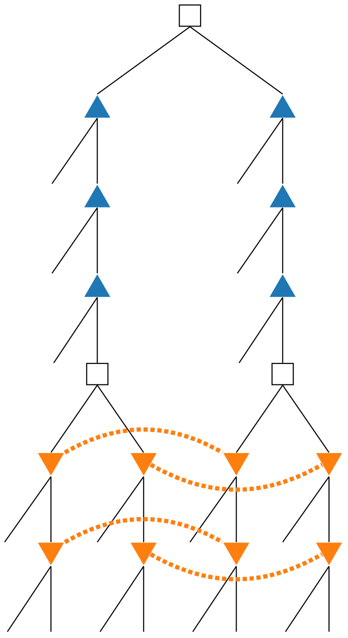


Figure 6.11: *The game used in the proof of Theorem 6.23, for $n = k = 2$.*

If ▲ exits, transition to the terminal node (j_1, v_1, E). Otherwise, transition to $(j_1, v_1 + 1)$.

- At the node $(j_1, n + 2)$, Chance chooses an integer $j_2 \in [k]$ and tells ▼, Transition to the node (j_1, j_2, E).
- For each $v_2 \in [n]$, the node (j_1, j_2, v_2) is a decision node for ▼. ▼ may *exit* or *continue*. If ▼ exits, transition to the terminal node $(j_1, j_2, v_2, \mathsf{E})$. Otherwise, transition to $(j_1, v_1 + 1)$.
- Finally, $(j_1, j_2, n + 1)$ is a terminal node for all j_1, j_2.

Since this result is only concerned with representing the correlation plan polytope (not necessarily with computing optimal equilibria), we do not need to specify utilities or chance probabilities—these do not affect the construction of the augmented game Γ^c nor the polytope Ξ^c.[6.14] We will identify the information sets of both players i by (j_i, v_i) for $j \in [k]$, and the infoset-action pairs by (j_i, v_i, E) and (j_i, v_i, C) for exiting and continuing respectively.

Γ has information complexity $2k$ since every public state has at most k sequences for each player. Every non-chance node has branching factor exactly 2.

Given a correlation plan $\boldsymbol{\xi}$, define the vector $\boldsymbol{m}^{\boldsymbol{\xi}} \in [0, 1]^{[k]\times[n]\times[k]\times[n]}$ where $\boldsymbol{m}^{\boldsymbol{\xi}}(j_1, v_1, j_2, v_2)$ is the probability that each player i exits at exactly the v_ith opportunity conditioned on observing j_i. Notice that, for $j_1, j_2 \in [k]$ and $v_1, v_2 \in [n]$, $\boldsymbol{m}^{\boldsymbol{\xi}}(j_1, v_1, j_2, v_2)$ is a linear function of both the correlation plan spaces Ξ^{EFCCE} and Ξ^{EFCE}: for $\boldsymbol{\xi} \in \Xi^{\text{EFCCE}}$, it is exposed as $\xi[(j_1, j_2, v_2, \mathsf{E})^{(j_1,v_1+1)}] - \xi[(j_1, j_2, v_2, \mathsf{E})^{(j_1,v_1)}]$; for $\boldsymbol{\xi} \in \Xi^{\text{EFCE}}$, it is exposed as $\xi[(j_1, j_2, v_2, \mathsf{E})^{(j_1,v_1,\mathsf{E})}]$. (For Ξ^{NFCCE}, $\boldsymbol{m}^{\boldsymbol{\xi}}$ is not a linear function of $\boldsymbol{\xi}$, so, as expected, the argument fails here.)

Let $M = \{\boldsymbol{m}^{\boldsymbol{\xi}} : \boldsymbol{\xi} \in \Xi^c\} \subseteq [0,1]^{[k]\times[n]\times[k]\times[n]}$ be the polytope of vectors $\boldsymbol{m}$ corresponding to correlated strategies. At this point, since M does not depend on the notion of equilibrium, we have no more need to distinguish between EFCCE and EFCE. It suffices to show that linear optimization on M can decide k-CLIQUE. First, we characterize the vertices of M. A vertex of M is characterized by, for each player $i \in \{1,2\}$ and each $j \in [k]$, picking at most one vertex $v_{i,j} \in [n]$, and constructing $\boldsymbol{m}$ by setting $\boldsymbol{m}(j_1, v_1, j_2, v_2) = \mathbf{1}\big\{v_{1,j_1} = v_1 \text{ and } v_{2,j_2} = v_2\big\}$. Now consider the objective function $g : M \to \mathbb{R}$ defined by

$$g(\boldsymbol{m}) = \mathop{\mathbb{E}}_{\substack{j_1,j_2\in[k]\\ v_1,v_2\in[n]}} \begin{cases} \boldsymbol{m}(j_1, v_1, j_2, v_2) & \text{if } j_1 = j_2 \text{ and } v_1 = v_2 \le n;\ \text{or } j_1 \ne j_2 \text{ and } (v_1, v_2) \in E \\ 0 & \text{otherwise} \end{cases}$$

where the expectation is over a uniformly random sample. We now claim that $\max_{\boldsymbol{m}\in M} g(\boldsymbol{m}) = 1$ if and only if G has a clique of size k, which will complete the proof.

($\Leftarrow$) If G has a k-clique $\{v_1^*, \dots, v_k^*\}$, then we set $v_{i,j} = v_j^*$ for both players $i \in \{1,2\}$, and indeed this achieves $g(\boldsymbol{m}) = 1$ by construction.

($\Rightarrow$) If $g(\boldsymbol{m}) = 1$, then for all j we must have $\sum_{v\in[n]} \boldsymbol{m}(j, v, j, v) = 1$, *i.e.*, $v_{1,j} = v_{2,j}$. But then $\{v_{1,j}, \dots, v_{1,k}\}$ must be a clique by construction, because otherwise there would be some $j_1 \ne j_2$ for which $\boldsymbol{m}(j_1, v_{1,j}, j_2, v_{1,j}) = 0$. □

Technically speaking, this result does not establish parameterized hardness of computing optimal EFCCEs or EFCEs, as there could hypothetically be a method for doing so that exploits the special nature of the (6.3). Indeed, the proof of Theorem 6.23 exploits the fact that the objective coefficient $g[h^\tau]$ may depend on τ as well as h, which is not the case for the LP (6.3). However, we know of no technique for optimal equilibria that would not also imply the ability to optimize over Ξ^c. Therefore, Theorem 6.23 is a lower bound that applies to all known techniques for computing optimal EFCCEs and EFCEs.

6.7 Other Special Cases

We will now discuss several other equilibrium concepts and problems, beyond optimal correlation, that are special cases of our framework. In most cases, the choice of R does not affect the equilibrium, and we will leave it unspecified. Also, unless otherwise specified, we consider the case $\mathcal{J} = \mathcal{I}$—that is, the player's information partition in the equilibrium matches the player's information partition for deviations. A summary and inclusion diagram for the equilibrium notions discussed and introduced in this section can be found in Figure 6.12.

[6.14] Note that Ξ^c is not the set of EFCEs—it is a representation of the set of correlation plans. That set does not depend on utilities or chance probabilities.

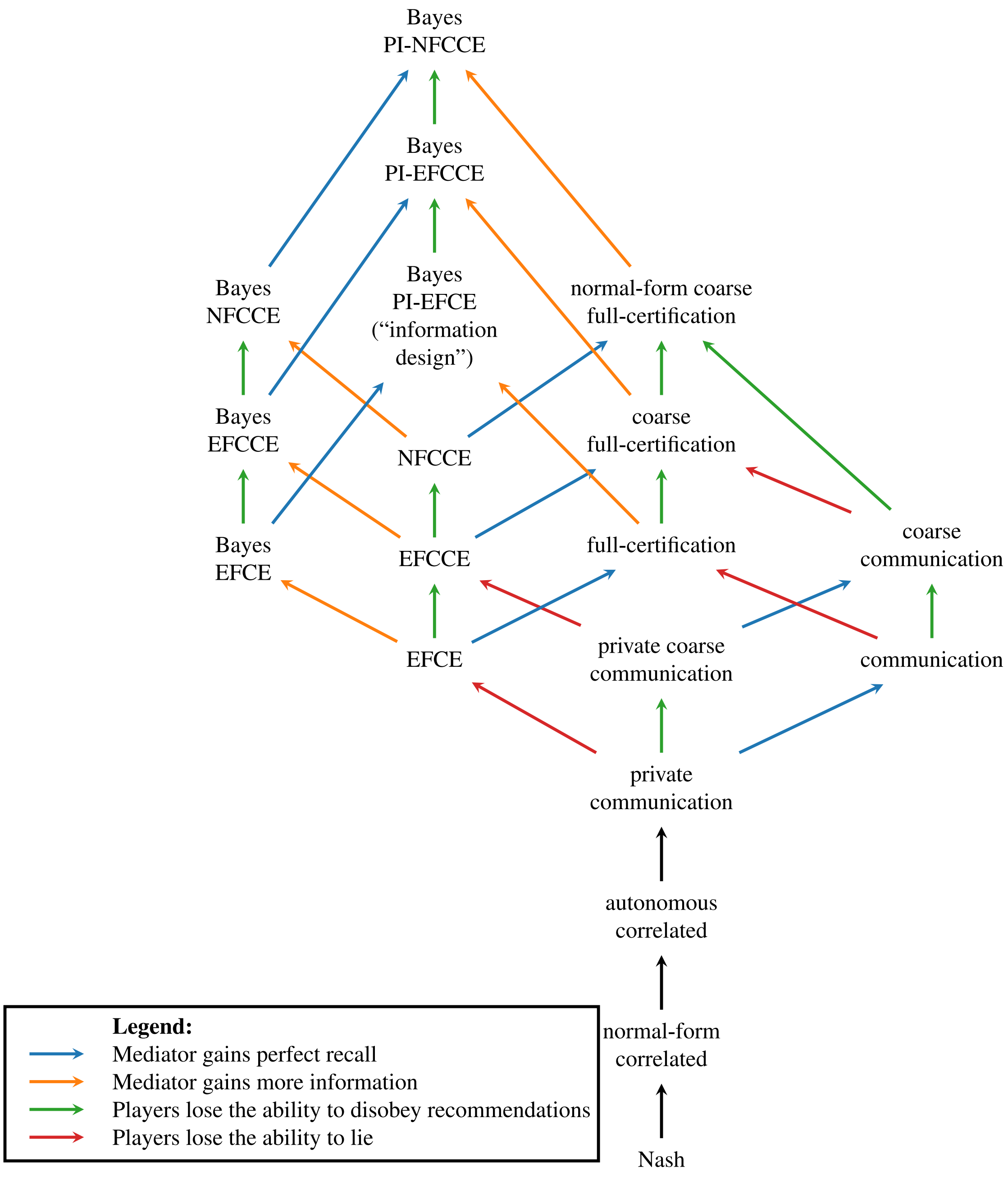

Bayes PI-NFCCE
Bayes PI-EFCCE
Bayes NFCCE
Bayes PI-EFCE ("information design")
normal-form coarse full-certification
Bayes EFCCE
NFCCE
coarse full-certification
Bayes EFCE
EFCCE
full-certification
coarse communication
EFCE
private coarse communication
communication
private communication
autonomous correlated
normal-form correlated
Nash
Legend:
Mediator gains perfect recall
Mediator gains more information
Players lose the ability to disobey recommendations
Players lose the ability to lie

Figure 6.12: *Inclusion diagram for the equilibrium notions discussed in Section 6.7. Arrows indicate subset relationships.*

6.7.1 Communication in Games

Games with simultaneous moves. There are settings in which it is useful for *multiple* players to simultaneously engage in a round of communication with the mediator before selecting moves, also simultaneously. To model this case in our framework, add dummy nodes so that every player has a round of communication with the mediator after they learn their information and before any player has to select a move.

S-certification equilibria. Forges and Koessler [110] defined a notion of certification equilibria for single-step Bayesian games. To recover that setting in our framework, we transform the Bayesian game into an extensive-form game in the natural way, with simultaneous moves (see above). We set $\mathcal{A}(I, m) = \mathcal{A}(I)$ for all I (*i.e.*, no action restrictions) and $\mathcal{P} = \emptyset$ (mediator has perfect memory), and finally we use the set of legal action reports $S(I)$ to control what information is certifiable. As such, the special case $\mathcal{A}(I, m) = \mathcal{A}(I)$ and $\mathcal{P} = \emptyset$ may be thought of as a generalization of the framework of Forges and Koessler [110] to extensive-form games.

Communication equilibria. Communication equilibria, in the sense of Forges [109] and Myerson [228], are S-certification equilibria where $S(I) = M$ for all I. That is, there are no restrictions on the communication protocol. To our knowledge, our framework gives the first algorithm for the polynomial-time computation of optimal communication equilibria.

Full-certification equilibria. A special case of certification equilibria of particular interest is what we coin the *full-certification equilibrium*, in which $S(I) = \{I, \perp\}$ for all I. That is, players may choose to withhold information, but they may not lie. For full-certification equilibria, the size of game $\hat{\Gamma}$ reduces dramatically. Indeed, in all terminal histories $(z, \boldsymbol{\tau})$ of $\hat{\Gamma}_{\text{red}}$, we must have $\tau_i = Ia$ for $I \preceq z$ for (at most) one player i, and $\boldsymbol{\tau}_{-i} = \boldsymbol{\sigma}_{-i}(z)$. Thus, $(z, \boldsymbol{\tau})$ is uniquely determined by selecting a terminal node $z \in Z$, an infoset $I \preceq z$, and an action $a \in \mathcal{A}(I)$. Therefore, we have $|\hat{\mathcal{Z}}| \le |Z|BD$, where B is the maximum branching factor and D is the depth of the game tree. That is, the size of $\hat{\Gamma}$ (and hence the size of the LP given by Theorem 6.4) goes from almost quadratic to almost linear in $|Z|$. Thus, optimal full-certification equilibria are at once easy to compute and easy to motivate.

Coarseness. Our framework enables us to naturally adapt the notion of *coarseness* from correlation to communication and certification equilibria. Therefore, we make the following definitions:

- An *extensive-form coarse S-certification equilibrium* is a $(S, R, \mathcal{A}, \emptyset, \mathcal{I})$-communication equilibrium where, for every I, we have $R(I) = \mathcal{A}(I)$, $\mathcal{A}(I, m) = \mathcal{A}(I)$ for $m \notin \mathcal{A}(I)$, and $\mathcal{A}(I, a) = \{a\}$ for $a \in \mathcal{A}(I)$. That is, a player who has reported honest information so far is bound to obey the mediator's recommendation; a player who has been dishonest is not. This is the only special case in this section for which we will need to specify the mediator's message set $R(I)$ explicitly: it matters because the mediator is not allowed here to "accidentally" force an action by sending a recommendation $a \in \mathcal{A}(I) \cap R(I')$ in response to being told false information I' at infoset I. Indeed, if we try to set $R(I) = M$ for all I, the nested range condition (and with it the revelation principle) ceases to hold, because a path $a_{\in \mathcal{A}(I)} \to a'_{\in \mathcal{A}(I')} \to a''_{\in \mathcal{A}(I)}$ would exist in the graph $G(R, \mathcal{A})$ (cf. Figure 6.1).

We define *extensive-form coarse communication equilibria* and *extensive-form coarse full-certification equilibria*, analogously to the non-coarse cases, by setting $S(I) = M$ and $S(I) = \{I, \perp\}$ for all I, respectively. In the latter case, the discussion about setting $R(I) = \mathcal{A}(I)$ is not relevant, because the NRC failure case described in the previous paragraph only happens if $I' \in S(I)$ for $I' \neq I$.

- A *normal-form coarse certification equilibrium* is recovered by adding dummy nodes and setting $S, R, \mathcal{A}$ as in NFCCE, and using $\mathcal{P} = \varnothing$ instead of $\mathcal{P} = \mathcal{I}$. That is, in normal-form coarse certification equilibrium, each player can choose to either play completely on its own, or delegate its play, including observations, completely to the mediator. The distinction between communication and certification disappears in the normal-form coarse case, because normal-form coarseness already enforces that players cannot deviate.

  Coarse certification equilibria, both normal-form and extensive-form, generalize the notion of *mediated equilibrium* [220] to extensive-form games. That is, in normal-form games (converted to extensive form using the above simultaneous move construction), the three notions coincide.

Private communication equilibria. The above discussion gives an interpretation of correlated equilibria as *certification equilibria with privacy constraints*. That is, correlated equilibrium notions arise from certification equilibria if we add the condition that *the mediator should regard the information sent by players as private, and not leak such messages to other players*. Analogously, we can add privacy constraints to the notions of communication equilibria by setting $\mathcal{P} = \mathcal{I}$, creating two new equilibrium concepts, the *private communication equilibrium* and *private coarse communication equilibrium*[6.15]. While it is still possible to apply the TB-DAG to these notions, the resulting algorithm would *not* have fixed-parameter runtime, because the mediator cannot trust any information it is told by players and therefore cannot even learn information that is publicly known to all players.

6.7.2 Mediator Surrogates: Mechanism Design and Information Design

In settings such as mechanism design and information design, the mediator itself has abilities in the game that are not directly specifiable in our language. For example, in mechanism design, the mediator has the power to dictate the outcome of the game (and players take no actions); in information design, the mediator has information that players do not have. Therefore, to model these situations in our framework, we introduce in this subsection the concept of a *mediator surrogate*, M. The mediator surrogate is an extra player in Γ, added in to represent the mediator. The mediator surrogate does not have incentives—that is, $u_{\mathsf{M}}(z) = 0$ for all z. If the mediator should have information that players do not have, the mediator surrogate observes that information. If the mediator should have power to take actions in the game (such as dictate outcomes), the

[6.15] Private normal-form coarse communication equilibria are equivalent to NFCCEs.

mediator surrogate takes those actions on behalf of the mediator. The mediator surrogate has no choice but to report honest information and follow recommendations, and all information reported by the mediator surrogate is usable—that is, we have $\mathcal{I}_\mathsf{M} \cap \mathcal{P} = \varnothing$, and for $I \in \mathcal{I}_\mathsf{M}$, we have $S(I) = \{I\}$ and $R_I(a) = \{a\}$ for each $a \in \mathcal{A}(I)$.

Since the mediator has full control over the actions taken by its surrogate, we will now cease to distinguish between the mediator and the surrogate and refer to both collectively as *the mediator*.

The settings of *mechanism design*, where the mediator has the power to take actions and the players have an informational advantage, and *information design*, where the mediator has an informational advantage and the players take actions, are usually treated separately in the literature. We promote a more unified approach. Indeed, in this section, we will show that both mechanism design and information design are special cases of our general framework. We are not the first to point out the connections between communication equilibria, mechanism design, and information design: for example, Bergemann and Morris [24] discuss in depth the relationship between the three notions in the context of Bayesian games. On top of that paper, our contribution is that we extend their ideas of unification to the more general setting of arbitrary extensive-form games and provide efficient algorithms for all of those cases.

6.7.2.1 Mechanism Design

In *mechanism design*, the players do not take any actions in the game, but have an informational advantage over the mediator. That is, the surrogate's information partition $\mathcal{I}_\mathsf{M}$ is the trivial partition in which it only knows what actions it has taken so far. The players have no actions—that is, every decision node for every (non-surrogate) player has only one child, but have information about the world state that the mediator does not have. As such, the sole purpose of the decision points of the players is to create a round of communication where the players may report their information to the mediator. The goal of automated mechanism design is to compute an *optimal mechanism*, which in our framework means an optimal communication equilibrium[6.16].

Payments. One particular common desirable feature in mechanism design is *payments*: the mediator can collect payments from or give payments to players, as a means of incentivizing players to report truthfully or of simply gaining revenue.

We can make this concrete as follows. When the mediator picks an outcome, chance picks a random player $i \in [n]$, reveals it to the mediator. The mediator then selects a payment $p \in \{L, U\}$ to be given to i, where $L, U \in \mathbb{R}$ are a minimum or maximum payment to that player[6.17]. The player gains utility np and the mediator loses utility np. By varying the probability with which the mediator makes the two payments, the mediator can (in expectation) pay any player any amount in the range $[L, U]$. The size of the mediator-augmented game increases by a factor of $O(n)$. Assuming that players' utilities are a linear function of the amount of money they earn or pay, this

[6.16] We do not consider optimal mechanism design in (weakly-)dominant strategies—we only insist that the profile in which players report honestly is a Nash equilibrium. However, we *do* consider *ex-post* incentive compatibility, which in many settings is equivalent to weakly-dominant incentive compatibility.

[6.17] If the player is paying money, then $p < 0$.

construction suffices to model payments to all n players in the range $[L, U]$. The ability of the mediator to "commit" to its strategy $\boldsymbol{\xi}$ is critical when payments are involved: if a mediator could not credibly commit, then it would never make a payment because it has negative incentive to do so.

This method of allocating payments assumes that the utilities of the players and the objective functions are both *quasilinear*: that is, that they can be expressed in the form $u(z, p) = v(z) + p$, where $z \in Z$ is the terminal node reached in the game and p is the amount of money paid to that player (or mediator). This assumption is typical in settings involving mechanism design.

Ex-post incentive compatibility. A common problem in mechanism design is designing mechanisms that are *ex-post incentive compatible*—that is, mechanisms that are incentive-compatible even when players know each others' private information. In our framework, this is accomplished by setting $\mathcal{J}$ to be the perfect-information partition in Γ—thus allowing players to observe each others' private information before deciding whether or not to truthfully report their information. Thus, in all settings described below, we could either work with *Bayes*-incentive compatibility by setting $\mathcal{J} = \mathcal{I}$, or *ex-post* incentive compatibility by setting $\mathcal{J}$ to be perfect-information.

Individual rationality. At a particular point in the game, we say that a player satisfies *individual rationality* if, conditioned on its information at that point (that is, its infoset $J \in \mathcal{J}$ and its transcript with the mediator), its expected utility is nonnegative. In our framework, we can impose individual rationality constraints by explicitly adding, in the original game Γ, decision nodes that allow the corresponding players to leave the game, accepting no further payoffs. In particular, the common types of individual rationality discussed in the mechanism design literature can be naturally incorporated:

1. *ex-ante* individual rationality, by adding such a decision node for each player before the root node of Γ,
2. *ex-interim* individual rationality, by adding such a decision node for each player after players observe their own types, and
3. *ex-post* individual rationality, by adding such a decision node for each player when the game terminates, and revealing to that player all the information in the game except the mediator's payment.

This modification will increase the size of Γ by a factor of around 2^n, where the exact factor will depend on what it means in the game for a player to leave the game. Keeping the mediator's payment hidden for *ex-post* individual rationality is critical: since the mediator can only choose a minimum or maximum payment (L or U), the "hidden" payment is what allows the mediator to essentially select the payment by randomizing between the extreme payments L and U.

This formulation of *ex-post* individual rationality leads to two perhaps-unintuitive properties that differ slightly from those usually considered in traditional mechanism design.

1. In certain settings (for example, if every outcome has very negative reward for each player and the mediator cannot pay the players), it may be impossible, or undesirable for the

mediator, to actually incentivize players to not leave the game. In that case, when an optimal equilibrium is computed in Γ, the mediator will sometimes *recommend* a player to leave.

2. Since our notion of individual rationality is formulated by explicitly allowing players to walk away, the mechanism must be robust to "double deviations" of the following form: a player *reports dishonestly*, and then *only accepts desirable outcomes*. For example, in an auction setting, this would mean that the bidder places an overbid, and then rejects the outcome if the proposed price is higher than its valuation. This type of deviation is typically not considered in mechanism design, but our framework results in mechanisms that are also robust to them.

One way to recover the more "usual" notion of individual rationality is to explicitly include constraints on the mediator's and players' strategies. In particular, (1) is circumvented by directly preventing the mediator from selecting outcomes that are not ex-post individually rational for an honest player, and (2) is circumvented by directly preventing a deviating player in the augmented game from leaving the game after sending false information. Since such general constraints are not a part of our framework, we will not discuss these circumvention techniques further.

Single-stage automated mechanism design. A consequence of the above analysis is that we recover the Bayes-Nash randomized automated mechanism design algorithm of Conitzer and Sandholm [67, 68] as a special case, as follows. That paper considers an automated mechanism design scenario with T types per player and n players. A type assignment $\boldsymbol{\theta} \in [T]^n$ is sampled from some joint distribution, and each type θ_i is revealed privately to player i. A round of communication ensues, in which each player informs the mediator about its own type (or lies about it). The mediator then chooses an outcome $o \in [O]$, and each player receives a utility that is a function of their own type θ_i and the outcome o. The resulting game has size $T^n O$, and each player has at most T^2 sequences, so the overall LP (6.3), after including payments, has size $\mathsf{poly}(O, T^n)$, which is the same result observed by Conitzer and Sandholm [68].

Automated multi-stage mechanism design. Zhang and Conitzer [310] consider an automated dynamic mechanism design setting in a Markov game with one player. Their main positive result is an LP for the setting of short-horizon MDPs, which can be viewed as a special case of our framework by simply unrolling the (short-horizon) MDP into an extensive-form game.

Papadimitriou et al. [235] study a dynamic auction design setting. In their most general setup, there are k players (bidders) with independent valuations for each of D items to be sold in sequence. The agents in their setting know *all* their valuations upfront, but only *report* their valuations for the item currently being sold. That is, in $\mathcal{I}$, the players only learn their current valuations, whereas in $\mathcal{J}$, the players learn their future valuations as well. In this setting, our algorithm matches their positive results, Theorems 7 and 8, that use a "dynamic programming LP" to compute the optimal randomized mechanism.

Sandholm et al. [264] study multi-stage mechanism design, but they use multiple stages for the purpose of reducing the amount of communication necessary for preference elicitation in what would otherwise be a single-stage setting. Their work is therefore orthogonal to ours.

Automated mechanism design with partially-verifiable types. Zhang et al. [311] give an algorithm for finding the optimal *direct* mechanism in a single-agent, single-stage setting with partially-verifiable types. The positive results in their paper focus on the case when all types have the same preferences over outcomes. Our work differs from that one in that we consider extensive-form (multi-stage) settings and a more general setup (in which the players can also take actions and can have arbitrary utility functions). However, we share in common with that paper the fact that we only compute the optimal *direct* mechanism, and hence rely on the revelation principle holding for that mechanism to be optimal across communication structures.

Kephart and Conitzer [172] analyze mechanism design in a single-stage, single-agent setting with costly reporting and not assuming the revelation principle. Their goal is to, given a social choice function, determine whether that function can be implemented. They analyze a large spectrum of cases, and discuss for each case whether this implementation problem is easy or (NP-)hard. In a follow-up paper [173], the same authors carefully investigate when the revelation principle does or does not hold (still in the single-stage, single-agent setting with reporting costs). When the revelation principle does hold, as we have discussed above, our framework matches, as a special case, the polynomial-time algorithm of Conitzer and Sandholm [67, 68], and both can be used to compute whether a social choice function is implementable, simply by adding the appropriate linear constraints to the linear program formulation. However, when the revelation principle fails, that approach will fail to find an implementation for any social choice function that cannot be implemented by a direct mechanism.

6.7.2.2 Information Design and Bayes-Correlated Equilibria

In *information design* [167] (also known as *Bayesian persuasion*), the roles of the mediator and player are reversed compared to automated mechanism design: the mediator ("principal") has informational advantage, and the player(s) take the actions. The difference between the two settings lies in who has the commitment power: in automated mechanism design, the side with the power to take actions has the commitment power; in information design, the side with the informational advantage has commitment power.

Information design is closely related to the concept of *Bayes-correlated equilibria* [23]. A Bayes-correlated equilibrium in a single-stage game is modeled in our framework as follows. Nature samples a state of the world $\boldsymbol{\theta} = (\theta_1, \dots, \theta_n) \in \Theta$ from a known prior, and reveals the full vector $\boldsymbol{\theta}$ to the mediator. Each player i then observes θ_i and takes an action. A Bayes-correlated equilibrium is a communication equilibrium of this game[6.18]. Information design is then the problem of computing an *optimal* Bayes-correlated equilibrium. The main result of this section thus implies that optimal information design in a single-stage setting with $|\mathcal{A}|$ actions per player and T types can be done in time $\mathsf{poly}(T, |\mathcal{A}|^n)$. For more on the connection between these concepts in the single-stage setting, we refer the reader to Bergemann and Morris [24].

There are at several reasonable generalizations of Bayes-correlated equilibria to the extensive-form setting, depending on what information the mediator observes. Indeed, adding a mediator surrogate

[6.18] In this setting, communication and certification equilibria coincide because the players have no information to misreport.

that observes extra information to the notions of correlated and communication equilibrium notions discussed in the previous section yields Bayes-versions of correlated, communication, and certification equilibria in extensive-form games.

1. Celli et al. [57] define a different notion of Bayes-correlated equilibria that is closer to normal-form correlated equilibrium. In their notion, there is still a world state $\boldsymbol{\theta}$ that is fully known to the mediator at the start of the game, but the mediator can only send a single message to the players after observing $\boldsymbol{\theta}$. They work with both normal-form coarse and normal-form versions of their equilibrium concept, which they call Bayes-NFCCE and Bayes-NFCE, respectively. In our framework, Bayes-NFCCE is recovered by starting from NFCCE and introducing a mediator surrogate that observes $\boldsymbol{\theta}$. Bayes-NFCE is not captured by our framework for the same reason that NFCE is not. In addition, our framework allows us to immediately define Bayes-EFCCE and Bayes-EFCE by starting from EFCCE and EFCE respectively and adding the same surrogate.

2. The original paper of Bergemann and Morris [23] motivates Bayes-correlated equilibria as the set of equilibria that can arise across *all possible information structures* that the players could possibly have. With that motivation, the logical extension of Bayes-correlated equilibria in extensive-form games is to allow the mediator to have *perfect* information about the current history h[6.19]. In this case, S does not matter (since the mediator already knows the players' information), but players are free to play whatever action they wish. That is, we set[6.20] $\mathcal{A}(I, m) = \mathcal{A}(I)$. To avoid name collisions with the notions already defined by Celli et al. [57], we will call the resulting notion *Bayes-perfect-information EFCE*, or *Bayes-PI-EFCE*. Similarly, we can define Bayes-PI-EFCCE and Bayes-PI-NFCCE by combining the same surrogate with the constructions for EFCCE and NFCCE in the previous section[6.21].

Gan et al. [119] study information design in Markov games with a single player, in which the principal can send messages to the player at every time step. The extensive-form analogue of their setting is, once again, a special case of our framework. The complexity gap between myopic, advice-myopic, and far-sighted players, discussed in that paper for Markov games, disappears in extensive-form games because extensive form naturally allows history-dependent strategies for both the mediator and the players.

Wu et al. [296] study information design in Markov games with myopic players. That is, a new player arrives at every timestep. After performing a single action, the player gains a reward dependent on the true state (which, of course, the player may not actually know) and its action, and then leaves the system forever. The authors devise an online learning algorithm that provably has low regret for the sender. Compared to their setting, our setting is significantly more

[6.19] If the extensive-form game is being used to model a case with simultaneous moves, one could only have the surrogate observe the history at the beginning of each simultaneous stage of the game.

[6.20] The choices of S, R, and $\mathcal{P}$ do not affect the equilibrium notion, because the mediator already knows the player's information anyway. Therefore players never have any incentive to misreport, and the mediator gains no useful information from the player's report.

[6.21] Bayes-PI-NFCE is meaningless as a notion: NFCEs are fundamentally restricted by the mediator only being allowed to send one message to the players, whereas the sequential nature of the Bayes-PI family of equilibria depends on the mediator's ability to send multiple messages.

general—allowing multiple non-myopic agents as well as other forms of limited information and communication—but ours is based on linear programming instead of online learning, and works with extensive-form games instead of Markov games.

Zhang et al. [312, 313] consider an automated mechanism design setting in which both the mediator and the (single) player have the ability to quit the process at each timestep (immediately ending the game), and there is no information asymmetry. They study Markov games, which in this setting are significantly more involved than extensive-form games. In our framework, the extensive-form analogue of this setting is more appropriately formulated as an "information-design-like" setting. We can do this by explicitly adding an exit action to every player decision node that immediately ends the game, and setting $S(I) = I$ (mediator learns player's information) and $\mathcal{A}(I, a) = \{a, \textsc{Exit}\}$ (player may only obey the recommendation or exit) for every I and $a \in \mathcal{A}(I)$.

6.8 Learning Optimal Equilibria

In this section, we introduce a new paradigm of learning in games for *computing* optimal equilibria. It applies to extensive-form settings with any number of players, including information design, and solution concepts such as correlated, communication, and certification equilibria. Further, our framework is general enough to also capture optimal mechanism design and optimal incentive design problems in sequential settings.

Summary of Our Results. A key insight that underpins our results is that computing *optimal* equilibria in multi-player extensive-form games can be cast via a Lagrangian relaxation as a two-player zero-sum extensive-form game. This unlocks a rich technology, both theoretical and experimental, developed for computing minimax equilibria for the more challenging—and much less understood—problem of computing optimal equilibria.

We thus focus on computing an optimal equilibrium by employing regret minimization techniques in order to solve the induced bilinear saddle-point problem. Such considerations are motivated in part by the remarkable success of no-regret algorithms for computing minimax equilibria in large two-player zero-sum games (*e.g.*, see [31, 37]), which we endeavor to transfer to the problem of computing optimal equilibria in multi-player games.

In this context, we show that employing standard regret minimizers, such as OGD or CFR, leads to a rate of convergence of $T^{-1/4}$ to optimal equilibria by appropriately tuning the magnitude of the Lagrange multipliers (Corollary 6.27). We also leverage the technique of *optimism*, pioneered by Chiang et al. [61], Rakhlin and Sridharan [252] and Syrgkanis et al. [281], to obtain an accelerated $T^{-1/2}$ rate of convergence (Corollary 6.28). These are the first learning dynamics that (provably) converge to optimal equilibria. Our bilinear formulation also allows us to obtain *last-iterate* convergence to optimal equilibria via optimistic gradient descent/ascent (Theorem 6.29), instead of the time-average guarantees traditionally derived within the no-regret framework. As such, we bypass known barriers in the traditional learning paradigm by incorporating an additional player, a *mediator*, into the learning process. Furthermore, we also study an alternative Lagrangian relaxation which, unlike our earlier approach, consists of solving a sequence of zero-sum games (*cf.* [101]). While the latter approach is less natural, we find that it is preferable when used in conjunction with deep RL solvers since it obviates the need for solving games with large reward ranges—a byproduct of employing the natural Lagrangian relaxation.

Experimental results. We demonstrate the practical scalability of our approach for computing optimal equilibria and mechanisms. First, we obtain state-of-the-art performance in a suite of 23 different benchmark game instances for seven different equilibrium concepts. Our algorithm significantly outperforms existing LP-based methods, typically by more than one order of magnitude. We also use our algorithm to derive an optimal mechanism for a sequential auction design problem, and we demonstrate that our approach is naturally amenable to modern deep RL techniques.

6.8.1 Preliminaries

In this section, we will work only with games that are already mediator-augmented; therefore, we will drop the hats and refer to the augmented game itself as Γ. Thus, for us, a game has n players, a mediator $\mathsf{0}$, and chance C. We will also assume that $\mathcal{J} = \mathcal{I}$, *i.e.*, the player's information partition is the same whether or not the player is deviating. This assumption is for simplicity of notation only; our results generalize immediately to the cases $\mathcal{J} \neq \mathcal{I}$ that are disucssed in Section 6.7.

Revelation principle. The *revelation principle* allows us, without loss of generality, to restrict our attention to equilibria where each player is playing some fixed pure strategy $\boldsymbol{o}_i \in \mathcal{X}_i$.

Definition 6.24. The game Γ satisfies the *revelation principle* if there exists a *direct* pure strategy profile $\boldsymbol{o} = (\boldsymbol{o}_1, \dots, \boldsymbol{o}_n)$ for the players such that, for all strategy profiles $(\boldsymbol{\xi}, \boldsymbol{x})$ for all players including the mediator, there exists a mediator strategy $\boldsymbol{\xi}' \in \Xi$ and functions $f_i : \mathcal{X}_i \to \mathcal{X}_i$ for each player i such that:

1. $f_i(\boldsymbol{o}_i) = \boldsymbol{x}_i$, and
2. $u_j(\boldsymbol{\xi}', \boldsymbol{x}'_i, \boldsymbol{o}_{-i}) = u_j(\boldsymbol{\xi}, f_i(\boldsymbol{x}'_i), \boldsymbol{x}_{-i})$ for all $\boldsymbol{x}'_i \in \mathcal{X}_i$, and players $j \in [n] \cup \{0\}$.

The function f_i in the definition of the revelation principle can be seen as a *simulator* for Player i: it tells Player i that playing $\boldsymbol{x}'_i$ if other players play $(\boldsymbol{\xi}, \boldsymbol{o}_{-i})$ would be equivalent, in terms of all the payoffs to all agents (including the mediator), to playing $f(\boldsymbol{x}'_i)$ if other agents play $(\boldsymbol{\xi}, \boldsymbol{x}_{-i})$. It follows immediately from the definition that if $(\boldsymbol{\xi}, \boldsymbol{x})$ is an ϵ-equilibrium, then so is $(\boldsymbol{\xi}', \boldsymbol{o})$—that is, every equilibrium is payoff-equivalent to a direct equilibrium. This version of

This revelation principle generalizes the revelation principle from Section 6.5, and applies and covers many cases of interest in economics and game theory. For example, in (single-stage or dynamic) mechanism design, the direct strategy $\boldsymbol{o}_i$ of each player is to report all information truthfully, and the revelation principle guarantees that for all non-truthful mechanisms $(\boldsymbol{\xi}, \boldsymbol{x})$ there exists a truthful mechanism $(\boldsymbol{\xi}', \boldsymbol{o})$ with the same utilities for all players.[6.22] For correlated equilibrium, the direct strategy $\boldsymbol{o}_i$ consists of obeying all (potentially randomized) recommendations

[6.22] In a mechanism design context, a strategy for the mediator $\boldsymbol{\xi}$ induces a mechanism; here we slightly abuse terminology by referring to $(\boldsymbol{\xi}, \boldsymbol{o})$ also as a mechanism.

that the mediator gives, and the revelation principle states that we can, without loss of generality, consider only correlated equilibria where the signals given to the players are what actions they should play. In both these cases (and indeed in general for the notions we consider in this section), it is therefore trivial to specify the direct strategies $\boldsymbol{o}$ without any computational overhead. Indeed, we will assume throughout the section that the direct strategies $\boldsymbol{o}$ are given.

Although the revelation principle is a very useful characterization of optimal equilibria, as long as we are given $\boldsymbol{o}$, all of the results in this section actually apply regardless of whether the revelation principle is satisfied: when it fails, our algorithms will simply yield an *optimal direct equilibrium* which may not be an optimal equilibrium.

Recall the linear program (6.2) that appeared in the proof of Theorem 6.4:

$$\max_{\boldsymbol{\xi} \in \Xi} \quad \boldsymbol{g}^\top \boldsymbol{\xi} \quad \text{s.t.} \quad \max_{\boldsymbol{x}_i \in \mathcal{X}_i} \boldsymbol{\xi}^\top \mathbf{A}_i \boldsymbol{x}_i \le 0 \;\; \forall i \in [n]. \tag{6.5}$$

Theorem 6.4 now proceeds by taking the dual linear program of the inner maximization, which suffices to show that (6.5) can be solved using linear programming.[6.23]

Finally, although our main focus in this section is on games with discrete action sets, it is worth pointing out that some of our results readily apply to continuous games as well using, for example, the discretization approach of Kroer and Sandholm [184].

6.8.2 Lagrangian Relaxations and a Reduction to a Zero-Sum Game

Our approach relies on Lagrangian relaxations of the linear program (6.5). In particular, in this section we introduce two different Lagrangian relaxations. The first one (Section 6.8.2.1) reduces computing an optimal equilibrium to solving a *single* zero-sum game. We find that this approach performs exceptionally well in benchmark extensive-form games in the tabular regime, but it may struggle when used in conjunction with deep RL solvers since it increases significantly the range of the rewards. This shortcoming is addressed by our second method, introduced in Section 6.8.2.2, which instead solves a *sequence* of suitable zero-sum games.

6.8.2.1 "Direct" Lagrangian

Directly taking a Lagrangian relaxation of the LP (6.5) gives the following saddle-point problem:

$$\max_{\boldsymbol{\xi} \in \Xi} \min_{\substack{\lambda \in \mathbb{R}_{\ge 0}, \\ \boldsymbol{x}_i \in \mathcal{X}_i : i \in [n]}} \quad \boldsymbol{g}^\top \boldsymbol{\xi} - \lambda \sum_{i=1}^{n} \boldsymbol{\xi}^\top \mathbf{A}_i \boldsymbol{x}_i. \tag{L1}$$

We first point out that the above saddle-point optimization problem admits a solution $(\boldsymbol{\xi}^*, \boldsymbol{x}^*, \lambda^*)$:

[6.23] Computing optimal equilibria can be phrased as a linear program, and so in principle Adler's reduction could also lead to an equivalent zero-sum game [7]. However, that reduction does not yield an *extensive-form* zero-sum game, which is crucial for our purposes; see Section 6.8.2.

Proposition 6.25. *The problem* (L1) *admits a finite saddle-point solution* $(\boldsymbol{\xi}^*, \boldsymbol{x}^*, \lambda^*)$. *Moreover, for all fixed* $\lambda > \lambda^*$, *the problems* (L1) *and* (6.5) *have the same value and same set of optimal solutions.*

Proof. Let v be the optimal value of (6.3). The Lagrangian of (6.3) is

$$\max_{\boldsymbol{\xi}\in\Xi} \min_{\substack{\lambda_i\in\mathbb{R}_{\geq 0},\\ \boldsymbol{x}_i\in\mathcal{X}_i : i\in[n]}} \quad \boldsymbol{g}^\top\boldsymbol{\xi} - \sum_{i=1}^{n} \lambda_i \boldsymbol{\xi}^\top \mathbf{A}_i \boldsymbol{x}_i.$$

Now, making the change of variables $\bar{\boldsymbol{x}}_i := \lambda_i \boldsymbol{x}_i$, the above problem is equivalent to

$$\max_{\boldsymbol{\xi}\in\Xi} \min_{\bar{\boldsymbol{x}}_i\in\bar{\mathcal{X}}_i : i\in[n]} \quad \boldsymbol{g}^\top\boldsymbol{\xi} - \sum_{i=1}^{n} \boldsymbol{\xi}^\top \mathbf{A}_i \bar{\boldsymbol{x}}_i. \tag{6.6}$$

where $\bar{\mathcal{X}}_i$ is the conic hull of $\mathcal{X}_i$: $\bar{\mathcal{X}}_i := \{\lambda_i \boldsymbol{x}_i : \boldsymbol{x}_i \in \mathcal{X}_i\}$. Note that, when $\mathcal{X}_i$ is a polytope of the form $\mathcal{X}_i := \{\mathbf{F}_i \boldsymbol{x}_i = \boldsymbol{f}_i, \boldsymbol{x}_i \geq 0\}$, its conic hull can be expressed as $\bar{\mathcal{X}}_i = \{\mathbf{F}_i \boldsymbol{x}_i = \lambda_i \boldsymbol{f}_i, \boldsymbol{x}_i \geq \mathbf{0}, \lambda_i \geq 0\}$. Thus, (6.6) is a bilinear saddle-point problem, where Ξ is compact and convex and $\bar{\mathcal{X}}_i$ is convex. Thus, Sion's minimax theorem [273] applies, and we have that the value of (6.6) is equal to the value of the problem

$$\min_{\bar{\boldsymbol{x}}_i\in\bar{\mathcal{X}}_i : i\in[n]} \max_{\boldsymbol{\xi}\in\Xi} \quad \boldsymbol{g}^\top\boldsymbol{\xi} - \sum_{i=1}^{n} \boldsymbol{\xi}^\top \mathbf{A}_i \bar{\boldsymbol{x}}_i. \tag{6.7}$$

Since this is a linear program[6.24] with a finite value, its optimum value must be achieved by some $\bar{\boldsymbol{x}} := (\bar{\boldsymbol{x}}_1, \dots, \bar{\boldsymbol{x}}_n) := (\lambda_1 \boldsymbol{x}_1, \dots, \lambda_n \boldsymbol{x}_n)$. Let $\lambda^* := \max_i \lambda_i$. Using the fact that $\boldsymbol{\xi}^\top \mathbf{A}_i \boldsymbol{o}_i = 0$ for all $\boldsymbol{\xi}$, the profile

$$\bar{\boldsymbol{x}}' := \left(\lambda^* \boldsymbol{x}_1', \dots, \lambda^* \boldsymbol{x}_n'\right) \quad \text{where} \quad \boldsymbol{x}_i' = \boldsymbol{o}_i + \frac{\lambda_i}{\lambda^*}(\boldsymbol{x}_i - \boldsymbol{o}_i)$$

is also an optimal solution in (6.7). Therefore, for any $\lambda \geq \lambda^*$, $\boldsymbol{x}' := (\boldsymbol{x}_1', \dots, \boldsymbol{x}_n')$ is an optimal solution for the minimizer in (2.1) that achieves the value of (6.3), so (6.3) and (2.1) have the same value.

Now take $\lambda > \lambda^*$, and suppose for contradiction that (2.1) admits some optimal $\boldsymbol{\xi} \in \Xi$ that is not optimal in (6.3). Then, either $\boldsymbol{g}^\top \boldsymbol{\xi} < v$, or $\boldsymbol{\xi}$ violates some constraint $\max_{\boldsymbol{x}_i} \boldsymbol{\xi}^\top \mathbf{A}_i \boldsymbol{x}_i \leq 0$. The first case is impossible because then setting $\boldsymbol{x}_i = \boldsymbol{o}_i$ for all i yields value less than v in (2.1). In the second case, since we know that (2.1) and (6.3) have the same value when $\lambda = \lambda^*$, we have

$$\boldsymbol{g}^\top\boldsymbol{\xi} - \lambda \max_{\boldsymbol{x}\in X} \sum_{i=1}^{n} \boldsymbol{\xi}^\top \mathbf{A}_i \boldsymbol{x}_i < \boldsymbol{g}^\top\boldsymbol{\xi} - \lambda^* \max_{\boldsymbol{x}\in X} \sum_{i=1}^{n} \boldsymbol{\xi}^\top \mathbf{A}_i \boldsymbol{x}_i \leq v. \qquad \square$$

[6.24] This holds by taking a dual of the inner minimization.

We will call the smallest possible λ^* the *critical Lagrange multiplier*.

Proposition 6.26. *For any fixed value λ, the saddle-point problem* (L1) *can be expressed as a zero-sum extensive-form game.*

Proof. Consider the zero-sum extensive-form game $\hat{\Gamma}$ between two players, the *mediator* and the *deviator*, with the following structure:

1. Nature picks, with uniform probability, whether or not there is a deviator. If nature picks that there should be a deviator, then nature samples, also uniformly, a deviator $i \in [n]$. Nature's actions are revealed to the deviator, but kept private from the mediator.
2. The game Γ is played. All players, except i if nature picked a deviator, are constrained to according to $\boldsymbol{o}_i$. The deviator plays on behalf of Player i.
3. Upon reaching terminal node z, there are two cases. If nature picked a deviator i, the utility is $-2\lambda n \cdot u_i(z)$. If nature did not pick a deviator, the utility is $2g(z) + 2\lambda \sum_{i=1}^n u_i(z)$.

The mediator's expected utility in this game is

$$g(\boldsymbol{\xi}, \boldsymbol{o}) - \lambda \sum_{i=1}^{n} [u_i(\boldsymbol{\xi}, \boldsymbol{x}_i, \boldsymbol{o}_{-i}) - u_i(\boldsymbol{\xi}, \boldsymbol{o})]. \qquad \square$$

This characterization enables us to exploit technology used for extensive-form zero-sum game solving to compute optimal equilibria for an entire hierarchy of equilibrium concepts

We will next focus on the computational aspects of solving the induced saddle-point problem (L1) using regret minimization techniques.

The first challenge that arises in the solution of (L1) is that the domain of the minimizing player is unbounded—the Lagrange multiplier is allowed to take any nonnegative value. Nevertheless, we show that it suffices to set the Lagrange multiplier to a fixed value (that may depend on the time horizon); appropriately setting that value will allow us to trade off between the equilibrium gap and the optimality gap. We combine this theorem with standard regret minimizers (such as variants of CFR employed in the experiments) to guarantee fast convergence to optimal equilibria.

Corollary 6.27. *There exist regret minimization algorithms such that when employed in the saddle-point problem* (L1)*, the average strategy of the mediator $\bar{\boldsymbol{\xi}} := \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{\xi}^{(t)}$ converges to the set of optimal equilibria at a rate of $T^{-1/4}$. Moreover, the per-iteration complexity is polynomial for communication equilibria, while for NFCCE, EFCCE and EFCE, implementing each iteration admits a fixed-parameter tractable algorithm.*

Proofs from this section require additional machinery, and are therefore deferred to the appendix of the full paper [316].

Furthermore, we leverage the technique of *optimism*, pioneered by Chiang et al. [61], Rakhlin and Sridharan [252], Syrgkanis et al. [281], to obtain a faster rate of convergence.

Corollary 6.28 (Improved rates via optimism)**.** *There exist regret minimization algorithms that guarantee that the average strategy of the mediator* $\bar{\boldsymbol{\xi}} := \frac{1}{T}\sum_{t=1}^{T}\boldsymbol{\xi}^{(t)}$ *converges to the set of optimal equilibria at a rate of* $T^{-1/2}$*. The per-iteration complexity is analogous to Corollary 6.27.*

While this rate is slower than the (near) T^{-1} rates known for converging to some of those equilibria [10, 79, 105, 245], Corollaries 6.27 and 6.28 additionally guarantee convergence to *optimal* equilibria; improving the $T^{-1/2}$ rate of Corollary 6.28 is an interesting direction for future research.

Last-iterate convergence. The convergence results we have stated thus far apply for the *average* strategy of the mediator—a typical feature of traditional guarantees in the no-regret framework. Nevertheless, an important advantage of our mediator-augmented formulation is that we can also guarantee *last-iterate convergence* to optimal equilibria in general games. Indeed, this follows readily from our reduction to two-player zero-sum games, leading to the following guarantee.

Theorem 6.29 (Last-iterate convergence to optimal equilibria in general games)**.** *There exist algorithms that guarantee that the last strategy of the mediator* $\boldsymbol{\xi}^{(T)}$ *converges to the set of optimal equilibria at a rate of* $T^{-1/4}$*. The per-iteration complexity is analogous to Corollaries 6.27 and 6.28.*

As such, our mediator-augmented paradigm bypasses known hardness results in the traditional learning paradigm since iterate convergence is no longer tied to Nash equilibria.

6.8.2.2 Thresholding and Binary Search

A significant weakness of the above Lagrangian is that the multiplier λ^* can be large. This means that, in practice, the zero-sum game that needs to be solved to compute an optimal equilibrium could have a large reward range. While this is not a problem for most tabular methods that can achieve high precision, more scalable methods based on reinforcement learning tend to be unable to solve games to the required precision. In this section, we will introduce another Lagrangian-based method for solving the program (6.5) that will not require solving games with large reward ranges.

Specifically, let $\tau \in \mathbb{R}$ be a fixed threshold value, and consider the bilinear saddle-point problem

$$\max_{\boldsymbol{\xi}\in\Xi} \min_{\substack{\boldsymbol{\lambda}\in\Delta(n+1),\\ \boldsymbol{x}_i\in\mathcal{X}_i : i\in[n]}} \boldsymbol{\lambda}_0(\boldsymbol{g}^\top\boldsymbol{\xi} - \tau) - \sum_{i=1}^{n} \boldsymbol{\lambda}_i \boldsymbol{\xi}^\top \mathbf{A}_i \boldsymbol{x}_i, \tag{L2}$$

This Lagrangian was also stated—but not analyzed—by Farina et al. [101], in the special case of correlated equilibrium concepts (NFCCE, EFCCE, EFCE). Compared to that paper, ours contains a more complete analysis, and is general to more notions of equilibrium.

Like (L1), this Lagrangian is also a zero-sum game, but unlike (L1), the reward range in this Lagrangian is bounded by an absolute constant:

Proposition 6.30. *Let Γ be a (mediator-augmented) game in which the reward for all agents is bounded in $[0,1]$. For any fixed $\tau \in [0,1]$, the saddle-point problem (L2) can be expressed as a zero-sum extensive-form game whose reward is bounded in $[-2,2]$.*

Proof. Consider the zero-sum extensive-form game $\hat{\Gamma}$ between two players, the *mediator* and the *deviator*, with the following structure:

1. The deviator picks an index $i \in [n] \cup \{0\}$.
2. If $i \neq 0$, nature picks whether Player i can deviate, uniformly at random.
3. The game Γ is played. All players, except i if $i \neq 0$ and nature selected that i can deviate, are constrained to play according to $\boldsymbol{o}_i$. The deviator plays on behalf of Player i.
4. Upon reaching terminal node z, there are three cases. If nature picked $i = 0$, the utility is $g(z) - \tau$. Otherwise, if nature picked that Player $i \neq 0$ can deviate, the utility is $-2u_i(z)$. Finally, if nature picked that Player $i \neq 0$ cannot deviate, the utility is $2u_i(z)$.

The mediator's expected utility in this game is exactly

$$\boldsymbol{\lambda}_0 \cdot g(\boldsymbol{\xi}, \boldsymbol{o}) - \sum_{i=1}^{n} \boldsymbol{\lambda}_i [u_i(\boldsymbol{\xi}, \boldsymbol{x}_i, \boldsymbol{o}_{-i}) - u_i(\boldsymbol{\xi}, \boldsymbol{o})]$$

where $\boldsymbol{\lambda} \in \Delta^{n+1}$ is the deviator's mixed strategy in the first step. □

The above observations suggest a binary-search-like algorithm for computing optimal equilibria; the pseudocode is given as Algorithm 6.13 (BinSearch). The algorithm solves $O(\log(1/\epsilon))$ zero-sum games, each to precision ϵ. Let v^* be the optimal value of (6.5). If $\tau \leq v^*$, the value of (L2) is 0, and we will therefore never branch low, in turn implying that $u \geq v^*$ and $\ell \geq v^* - \epsilon$. As a result, we have proven:

Theorem 6.31. BinSearch *returns an ϵ-approximate equilibrium $\boldsymbol{\xi}$ whose value to the mediator is at least $v^* - 2\epsilon$. If the underlying game solver used to solve* (L2) *runs in time $f(\Gamma, \epsilon)$, then* BinSearch *runs in time $O(f(\Gamma, \epsilon) \log(1/\epsilon))$.*

The differences between the two Lagrangian formulations can be summarized as follows:

1. Using (L1) requires only a single game solve, whereas using (L2) requires $O(\log(1/\epsilon))$ game solves.
2. Using (L2) requires only an $O(\epsilon)$-approximate game solver to guarantee value $v^* - \epsilon$, whereas using (L1) would require an $O(\epsilon/\lambda^*)$-approximate game solver to guarantee the same, even assuming that the critical Lagrange multiplier λ^* in (L1) is known.

Which is preferred will therefore depend on the application. In practice, if the games are too large to be solved using tabular methods, one can use approximate game solvers based on deep reinforcement learning. In this setting, since reinforcement learning tends to be unable to achieve the high precision required to use (L1), using (L2) should generally be preferred. In Section 6.9, we back up these claims with concrete experiments.

Algorithm 6.13 (BinSearch): Pseudocode for binary search-based algorithm

```
input: game $\Gamma$ with mediator reward range $[0, 1]$, target precision $\epsilon > 0$
$\ell \leftarrow 0, u \leftarrow 1$
while $u - \ell > \epsilon$ do
    $\tau \leftarrow (\ell + u)/2$
    run an algorithm to solve game (L2) until either
        (1) it finds a $\boldsymbol{\xi}$ achieving value $\geq -\epsilon$ in (L2), or
        (2) it proves that the value of (L2) is $< 0$
    if case (1) happened then $\ell \leftarrow \tau$
    else $u \leftarrow \tau$
return the last $\boldsymbol{\xi}$ found
```

6.9 Experiments

Here, we describe some of the experiments that we have run using the algorithms described in this part. Since all the techniques in this part are interrelated, this section is standalone rather than a subsection.

6.9.1 Optimal Equilibria in Tabular Games

We first extensively evaluate the empirical performance of our two-player zero-sum reduction (Section 6.8.2.1) for computing seven equilibrium solution concepts across 23 game instances; the results using the method of Section 6.8.2.2 are slightly inferior, and are included in the appendix of Zhanget al. [316].

The game instances we use are also described in detail in the appendix of Zhanget al. [316], and belong to following eight different classes of established parametric benchmark games, each identified with an alphabetical mnemonic: `B` – Battleship [101], `D` – Liar's dice [199], `GL` – Goofspiel [254], `K` – Kuhn poker [187], `L` – Leduc poker [275], `RS` – ridesharing game [305], `S` – Sheriff [101], `TP` – double dummy bridge game [305].

In Figure 6.15, we have plotted the payoff spaces of some representative instances. The plots show how the polytopes of communication and full-certification equilibria behave relative to correlated equilibria. In the *battleship* and *sheriff* instances, the space of communication equilibrium payoffs is a single point, which implies that the space of NFCE (and hence Nash) equilibrium payoffs is also that single point. Unfortunately, that point is the Pareto-least-optimal point in the space of EFCEs. In the *ridesharing* instances, communication allows higher payoffs. This is because the mediator is allowed to "leak" information between players.

6.9.2 Exact Sequential Auction Design

Next, we use our approach to derive the optimal mechanism for a sequential auction design problem. In particular, we consider a two-round auction with two bidders, each starting with a budget of 1. The valuation for each item for each bidder is sampled uniformly at random from

Game	# Nodes	NFCCE		EFCCE		EFCE		COMM		CERT	
		LP	CFR	LP	CFR	LP	CFR	LP	CFR	LP	CFR
B2222	1573	0.00s	0.00s	0.00s	0.01s	0.00s	0.02s	2.00s	1.49s	0.00s	0.02s
B2322	23,839	0.00s	0.01s	3.00s	0.69s	9.00s	1.60s	timeout	4m 41s	2.00s	1.24s
B2323	254,239	6.00s	0.33s	1m 21s	14.23s	3m 40s	44.87s	timeout	timeout	37.00s	40.45s
B2324	1,420,639	38.00s	2.73s	timeout	3m 1s	timeout	10m 48s	timeout	timeout	timeout	6m 14s
D32	1017	0.00s	0.01s	0.00s	0.02s	12.00s	0.40s	0.00s	0.06s	0.00s	0.01s
D33	27,622	2m 17s	12.93s	timeout	1m 46s	timeout	timeout	timeout	4m 37s	4.00s	3.14s
GL3	7735	0.00s	0.01s	1.00s	0.02s	0.00s	0.01s	timeout	7.72s	0.00s	0.02s
K35	1501	49.00s	0.76s	46.00s	0.67s	57.00s	0.55s	1.00s	0.03s	0.00s	0.01s
L3132	8917	26.00s	0.59s	8m 43s	5.13s	8m 18s	6.10s	8.00s	3.46s	1.00s	0.10s
L3133	12,688	38.00s	0.94s	20m 26s	8.88s	21m 25s	6.84s	12.00s	3.40s	1.00s	0.22s
L3151	19,981	timeout	15.12s	timeout	timeout	timeout	timeout	timeout	16.73s	2.00s	0.21s
L3223	15,659	4.00s	0.44s	1m 10s	2.94s	2m 2s	5.52s	19.00s	18.19s	1.00s	0.61s
L3523	1,299,005	timeout	1m 7s	timeout	timeout	timeout	timeout	timeout	timeout	timeout	2m 58s
S2122	705	0.00s	0.00s	0.00s	0.01s	0.00s	0.02s	2.00s	0.35s	0.00s	0.02s
S2123	4269	0.00s	0.01s	1.00s	0.06s	1.00s	0.15s	1m 33s	59.63s	1.00s	0.15s
S2133	9648	1.00s	0.02s	3.00s	0.11s	3.00s	0.49s	timeout	12m 11s	2.00s	0.92s
S2254	712,552	1m 58s	7.43s	timeout	22.01s	timeout	3m 34s	timeout	timeout	timeout	2m 42s
S2264	1,303,177	3m 43s	11.74s	timeout	39.23s	timeout	timeout	timeout	timeout	timeout	timeout
TP3	910,737	1m 38s	7.44s	timeout	13.76s	timeout	13.46s	timeout	timeout	timeout	26.70s
RS212	598	0.00s	0.00s	0.00s	0.00s	0.00s	0.00s	2.00s	0.01s	0.00s	0.00s
RS222	734	0.00s	0.00s	0.00s	0.00s	0.00s	0.00s	3.00s	0.01s	0.00s	0.00s
RS213	6274	timeout	14.68s	timeout	15.54s	timeout	23.37s	6m 25s	8.74s	0.00s	0.02s
RS223	6238	timeout	timeout	timeout	timeout	timeout	timeout	8m 54s	4.00s	1.00s	0.01s

Table 6.14: *Experimental comparison between our learning-based approach ('CFR', Section 6.8.2.1) and our linear-programming-based method ('LP', Section 6.4). Within each pair of cells corresponding to 'LP' vs 'CFR,' the faster algorithm is shaded blue while the hue of the slower algorithm depends on how much slower it is. If both algorithms timed out, they are both shaded gray.*

the set $\{0, 1/4, 1/2, 3/4, 1\}$. We consider a mediator-augmented game in which the principal chooses an outcome (allocation and payment for each player) given their reports (bids). We use CFR+ [283] as learning algorithm and a fixed Lagrange multiplier $\lambda := 25$ to compute the optimal communication equilibrium that corresponds to the optimal mechanism. We terminated the learning procedure after 10000 iterations, at a duality gap for (L1) of approximately 4.2×10^{-4}. Figure 6.16 (left) summarizes our results. On the y-axis we show how exploitable (that is, how incentive-incompatible) each of the considered mechanisms are, confirming that for this type of sequential settings, second-price auctions (SP) with or without reserve price, as well as the first-price auction (FP), are typically incentive-incompatible. On the x-axis, we report the hypothetical revenue that the mechanism would extract assuming truthful bidding. Our mechanism is provably incentive-compatible and extracts a larger revenue than all considered second-price mechanisms. It also would extract less revenue than the hypothetical first-price auction if the bidders behaved truthfully (of course, real bidders would not behave honestly in the first-price auction but rather would shade their bids downward, so the shown revenue benchmark in Figure 6.16 is actually not

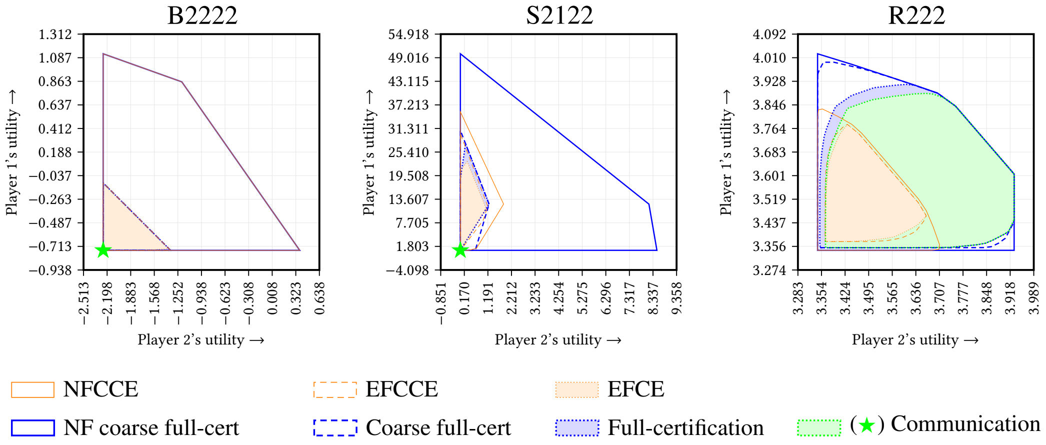


Figure 6.15: *Payoff spaces for various games and notions of equilibrium. The symbol ★ indicates that the set of communication equilibrium payoffs for that game is (at least, modulo numerical precision) that single point. In the battleship instance, many of the notions overlap.*

achievable). Intriguingly, we observed that 8% of the time the mechanism gives an item away for free. Despite appearing irrational, this behavior can incentivize bidders to use their budget earlier in order to encourage competitive bidding, and has been independently discovered in manual mechanism design recently [85].

6.9.3 Scalable Sequential Auction Design via Deep Reinforcement Learning

We also combine our framework with deep-learning-based algorithms for scalable equilibrium computation in two-player zero-sum games to compute optimal mechanisms in two sequential auction settings. To compute an optimal mechanism using our framework, we use the PSRO algorithm [191], a deep reinforcement learning method based on the double oracle algorithm that has empirically scaled to large games such as Starcraft [288] and Stratego [212], as the game solver in BinSearch.[6.25] To train the best responses, we use proximal policy optimization (PPO) [268].

First, to verify that the deep learning method is effective, we replicate the results of the tabular experiments in Section 6.9.2. We find that PSRO achieves the same best response values and optimal equilibrium value computed by the tabular experiment, up to a small error. These results give us confidence that our method is correct.

Second, to demonstrate scalability, we run our deep learning-based algorithm on a larger auction environment that would be too big to solve with tabular methods. In this environment, there are four rounds, and in each round the valuation of each player is sampled uniformly from $\{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$. The starting budget of each player is, again, 1. We find that, like the smaller setting, the optimal revenue of the mediator is ≈ 1.1 (right-side of Figure 6.16). This

[6.25] We also tested PSRO on the Lagrangian (L1), but this proved to be incompatible with deep learning due to the large reward range induced by the multiplier λ.

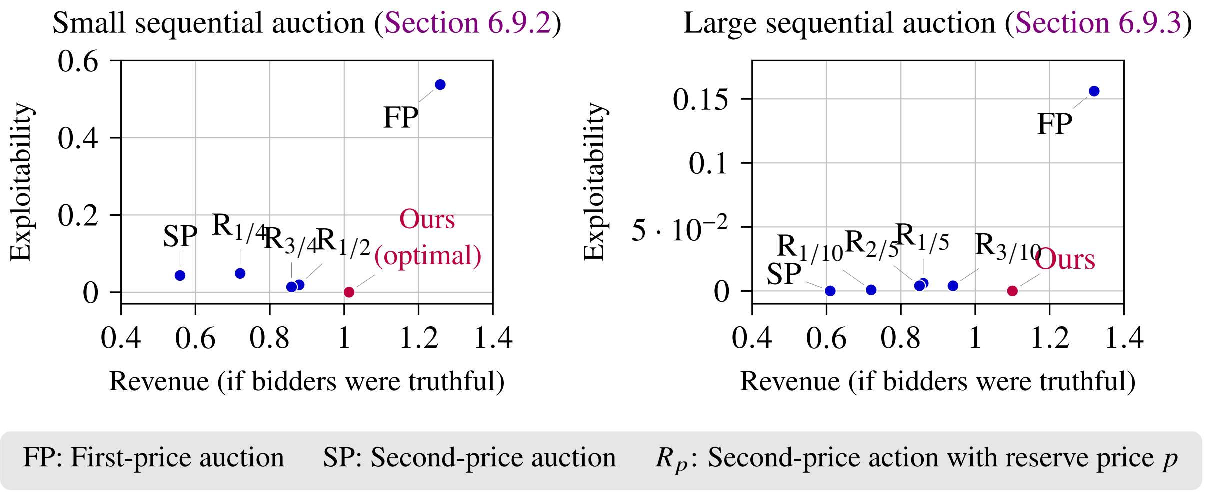


Figure 6.16: *Exploitability is measured by summing the best response for both bidders to the mechanism. Zero exploitability corresponds to incentive compatibility. In a sequential auction with budgets, our method is able to achieve higher revenue than second-price auctions and better incentive compatibility than a first-price auction.*

revenue exceeds the revenue of every second-price auction (none of which have revenue greater than 1).[6.26]

6.10 Conclusion and Future Research

We proposed a new paradigm of learning in games. It applies to mechanism design, information design, and solution concepts in multi-player extensive-form games such as correlated, communication, and certification equilibria. Leveraging a Lagrangian relaxation, our paradigm reduces the problem of computing optimal equilibria to determining minimax equilibria in zero-sum extensive-form games. We also demonstrated the scalability of our approach for *computing* optimal equilibria by attaining state-of-the-art performance in benchmark tabular games, and by solving a sequential auction design problem using deep reinforcement learning. Along the way, we have shown parameterized complexity results—both upper and lower bounds—for the special case of computing optimal *correlated* equilibria.

Possible directions of future research include the following.

1. Is there a better-than-quadratic-size linear program or similar algorithm for communication equilibria?
2. Is it possible to extend our augmented game construction to also cover *normal-form* correlated equilibria while maintaining efficiency?

[6.26] We are inherently limited in this setting by the inexactness of best responses based on deep reinforcement learning; as such, it is possible that these values are not exact. However, because of the success of above tabular experiment replications, we believe that our results should be reasonably accurate.

3. Investigate further the comparison between communication and correlation in games. For example, when and why do communication equilibria achieve higher social welfare than extensive-form correlated equilibria?

4. Extend CorrelationDAG in such a way that it also has polynomial size in all triangle-free games.

5. An intelligent combination—rather than merely a selection of one versus the other—of the correlation DAG and the column generation algorithm may lead to faster practical algorithms.

6. Investigate possible use of the payoff structure in the game; for example, investigate extensions of the concept of *smooth games* [256].

Part II

Learning Agents in Games, Correlated Equilibria, and Optimization

Chapter 7

Steering Learning Agents to Optimal Equilibria

7.1 Introduction

Any student of game theory learns that games can have multiple equilibria of different quality—for example, in terms of social welfare. As such, a foundational problem that has received tremendous interest in the literature revolves around characterizing the quality of the equilibrium reached under *no-regret* learning dynamics. The outlook that has emerged from this endeavor, however, is discouraging: typical learning algorithms can fail spectacularly at reaching desirable equilibria. This is rather dramatically illustrated in the example of Figure 7.1 (second panel). Learning agents initialized at either A, B, or C will in fact converge to the *Pareto-pessimal* Nash equilibrium of the game (bottom-left corner); only an initialization close to the Pareto-dominant equilibrium (such as D in the top-right corner) will end up with the desired outcome.

Our goal in this chapter is to develop methods to *steer* learning agents toward better equilibrium outcomes. To do so, we will use a *mediator* that can observe the agents playing the game, give *advice* to the agents (in the form of action recommendations), and *pay* the agents as a function of what actions they played. Our goal is to develop algorithms that allow the mediator to steer agents to a target equilibrium, while not spending too much money doing so. Critically, our only assumption on the agents' behavior is that they have no regret in hindsight. This is a fairly mild assumption compared to the assumptions made by many past papers on similar topics. We elaborate on the comparison to related work in the full paper [318].

Beyond the obvious relation to equilibrium selection, our model also has implications for the problem of *information design* and Bayesian persuasion (*e.g.*, [167]). Indeed, we will show that we can steer players not only to any Nash equilibrium but to any *Bayes-correlated equilibrium (BCE)*—the solution concept most naturally associated with the problem of information design.

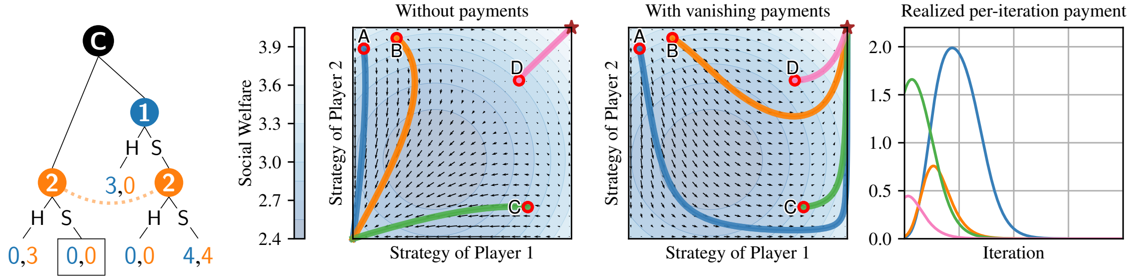


Figure 7.1: ***Left:*** *An extensive-form version of a stag hunt. Chance plays uniformly at random at the root note, and the dotted line connecting the two nodes of Player 2 indicates an infoset: Player 2 cannot distinguish the two nodes. The game has two equilibria: one at the bottom-left corner, and one at the top-right corner (star). The latter is Pareto-dominant. Introducing* vanishing *realized payments alters the gradient landscape, steering players to the optimal equilibrium (star) instead of the suboptimal one (opposite corner). The capital letters show the players' initial strategies. Lighter color indicates higher welfare and the star shows the highest-welfare equilibrium. Further details are in the appendix of the full paper [318].*

7.2 Summary of Our Results

Here we summarize our model and results. There is a fixed, arbitrary extensive-form game Γ, being played repeatedly over rounds $t = 1, \dots, T$. Players' rewards are assumed to be normalized to range $[0, 1]$. The players are assumed to play in such a way that their regret increases sublinearly as a function of T. This is a fairly natural and mild assumption (as discussed in the previous paragraph), and moreover there are many well-known algorithms that players can use to efficiently achieve sublinear regret in extensive-form games, perhaps the best-known of which is CFR, which has regret $T^{1/2}$ ignoring game-dependent constants.[7.1]

Broadly speaking, the goal of this chapter is to design methods of *steering* the learning behavior of the players so that they reach desirable equilibria instead of undesirable ones. We do this by introducing a *mediator* to the game. After each round, the mediator observes how the players played the game, and has the power to give nonnegative *payments* $p_i^{(t)}$ to each player i at each round t. We will first consider the case where a target *pure Nash equilibrium* is given as part of the problem instance.

A few observations follow easily. If the mediator's payments are not bounded, the mediator can trivially steer the players toward any outcome at all—not just equilibrium outcomes—by simply paying the players to play that outcome. We must therefore somehow bound the budget of the mediator. We will study two different budgets: a *per-round budget*, which constrains the individual payments $p_i^{(t)}$, and a *total budget*, which constrains their sum over time. We start by showing that the total budget must be allowed to grow with time.

[7.1] Throughout the introduction, game-dependent constants are omitted for clarity and to emphasize the dependence on T. In all cases, the omitted game-dependent constant is polynomial in the number of nodes in the game tree.

Proposition 7.1 (Informal version of Proposition 7.9)**.** *For any fixed total budget B, there is a time horizon T large enough that the steering problem is impossible.*

As a result, the total budget must be allowed to grow with the time horizon, but yet, for the problem to be interesting, the budget cannot be allowed to grow too fast. We thus focus on the regime where the budget is allowed to grow with T, but only *sublinearly*—that is, the *average* per-round payment must vanish in the limit $T \to \infty$. We are interested in algorithms for which both the average budget and rate of convergence to the desired equilibrium can both be bounded by T^{-c} for some absolute constant $c > 0$. We show the following.

Theorem 7.2 (Informal version of Theorems 7.11 and 7.16)**.** *Steering to pure-strategy equilibria is possible in normal-form or Bayesian games, with absolute constant per-round budget. The average budget and rate of convergence to equilibrium are both $T^{-1/4}$.*

Intuitively, the mediator sends payments in such a way as to 1) reward the player a small amount for playing the equilibrium, and 2) *compensate* the player for deviations of other players. The goal of the mediator is to set the payments in such a way that the target equilibrium actions become *strictly dominant* for the players, and therefore the players must play them.

Next we turn to the extensive-form setting. Settings such as information design, in which first a signal is designed, and then players take actions, are naturally extensive-form games. We distinguish between two settings: the *full feedback* setting, in which the mediator observes every player's entire strategy at every round, and the *trajectory-feedback setting*, in which the mediator only observes the trajectories that are actually played by the players.[7.2]

The *full feedback* setting yields results similar to the normal-form setting.

Theorem 7.3 (Informal version of Theorem 7.13)**.** *Steering to pure-strategy equilibria is possible in extensive-form games with full feedback, with absolute constant per-round budget. The average budget and rate of convergence to equilibrium are both $T^{-1/4}$.*

The *trajectory feedback* case, however, is quite different.

Theorem 7.4 (Informal version of Theorem 7.17)**.** *With only trajectory feedback and absolute constant per-round budget, steering in general extensive-form games is impossible, even to the welfare-maximizing pure Nash equilibrium.*

Intuitively, the discrepancy is because, with only trajectory feedback, it is not possible to make the target equilibrium dominant using only nonnegative, vanishing-on-average payments, so the techniques used for the previous results cannot apply. This phenomenon can already be observed in the "stag hunt" game in Figure 7.1: for Player 2, Stag (S) cannot be a weakly-dominant strategy

[7.2] This distinction becomes only meaningful for extensive-form games. For normal-form games, the two settings above coincide, because the "trajectory" in a normal-form game *is* just a list consisting of each player's chosen action.

unless a payment is given at the boxed node, which would be problematic because such payments would also appear in the welfare-optimal equilibrium (S, S). Thus, one needs to be more clever. Fortunately, steering is still possible in this setting, but only if the per-round budget is also allowed to grow:

Theorem 7.5 (Informal version of Theorem 7.19)**.** *Steering to pure-strategy equilibria is possible in extensive-form games with bandit feedback. The average budget and rate of convergence to equilibrium are both $T^{-1/8}$, and the per-round budget grows at rate $T^{1/8}$.*

Next, we generalize our results beyond pure Nash equilibria. To do this, we will require the mediator to have the additional ability to give *advice* to the players, in the form of action recommendations. First, we show that using advice is a *necessary* condition for steering to even mixed Nash equilibria.

Theorem 7.6 (Informal version of Theorem 7.21)**.** *Without advice, there exists a normal-form game in which the unique optimal Nash equilibrium is mixed, and it is impossible to steer players toward it.*

If we allow advice, it turns out to be possible to steer players not just to mixed Nash equilibria but to a far broader set of equilibria known as the *Bayes-correlated equilibria*.

Theorem 7.7 (Informal version of Theorem 7.23)**.** *With advice, steering to Bayes-correlated equilibria is possible in extensive-form games. The conditions and rates are the same as those for pure Nash equilibria.*

Intuitively, the result follows because Bayes-correlated equilibria can be viewed as the pure Nash equilibria of an *augmented game* in which the advice is treated as part of the game's observations. Bayes-correlated equilibria are a very general solution concept that include, for example, all the extensive-form correlated equilibria [291] and communication equilibria [109, 228], among other notions.

In Section 7.7, we study the problem of steering *without* prior knowledge of the utility functions. Since the equilibrium in this setting is not known *a priori*, we show that, at least for normal-form games, it is possible for a mediator to first *learn* the utilities, and *then* compute the optimal equilibrium and steer players toward it. Along the way, we define a new solution concept, the *correlated equilibrium with payments*, that characterizes the optimal outcome for the mediator, given a particular utility function of the mediator.

Finally, we complement our theoretical analysis by implementing and testing our steering algorithms in several benchmark games in Section 7.8.

	Per-Iteration Payment	
	Absolute Constant	Growing with T
Normal Form and Bayesian Games	$T^{-1/4}$ (Theorems 7.13 and 7.16)	
Extensive Form Games	Impossible (Theorem 7.17)	$T^{-1/8}$ (Theorem 7.19)

Table 7.2: *Summary of our positive algorithmic results for steering to a known equilibrium with trajectory feedback. In this table, we hide polynomial game-dependent constants, and assume that regret minimizers incur regret* $T^{-1/2}$.

7.3 The Steering Problem

In this section, we introduce what we call the *steering* problem. Informally, the steering problem asks whether a mediator can always steer players to any given equilibrium of an extensive-form game.

Definition 7.8 (Steering Problem for Pure-Strategy Nash Equilibrium)**.** Let Γ be an extensive-form game with payoffs bounded in $[0, 1]$. Let $\boldsymbol{o}$ be an arbitrary pure-strategy Nash equilibrium of Γ, which we will call the *target equilibrium*. The mediator knows the game Γ, as well as a function $R(T) = o(T)$, which may be game-dependent, that bounds the regret of each player. At each round $t \in [T]$, the mediator picks *payment functions* for each player, $p_i^{(t)} : \mathcal{X}_1 \times \cdots \times \mathcal{X}_n \to [0, P]$, where $p_i^{(t)}$ is linear in $\boldsymbol{x}_i$ and continuous in $\boldsymbol{x}_{-i}$, and P defines the largest allowable per-iteration payment. Then, players pick strategies $\boldsymbol{x}_i^{(t)} \in \mathcal{X}_i$. Each player i then gets utility $v_i^{(t)}(\boldsymbol{x}_i) := u_i(\boldsymbol{x}_i, \boldsymbol{x}_{-i}^{(t)}) + p_i^{(t)}(\boldsymbol{x}_i, \boldsymbol{x}_{-i}^{(t)})$. The mediator has two desiderata.

(S1) (Payments) The time-averaged realized payments to the players, defined as

$$\max_{i \in [n]} \frac{1}{T} \sum_{t=1}^{T} p_i^{(t)}(\boldsymbol{x}^{(t)}),$$

converges to 0 as $T \to \infty$.

(S2) (Target Equilibrium) Players' actions are indistinguishable from the Nash equilibrium $\boldsymbol{o}$. That is, for every terminal node z, the *directness gap*, defined as

$$\sum_{z \in \mathcal{Z}} \left| \frac{1}{T} \sum_{t=1}^{T} \hat{\boldsymbol{x}}^{(t)}(z) - \hat{\boldsymbol{o}}(z) \right| = \left\| \frac{1}{T} \sum_{t=1}^{T} \hat{\boldsymbol{x}}^{(t)} - \hat{\boldsymbol{o}} \right\|_1,$$

converges to 0 as $T \to \infty$, where $\hat{\boldsymbol{x}} \in \Delta(\mathcal{Z})$ is the probability distribution on $\mathcal{Z}$ induced by profile $\boldsymbol{x}$.

The assumption imposed on the payment functions in Definition 7.8 ensures the existence of Nash equilibria in the payment-augmented game (*e.g.*, [116], p. 34). Throughout this chapter, we will refer to players as *direct* if they are playing actions prescribed by the target equilibrium strategy $\boldsymbol{o}$. Critically, (S2) does not require that the strategies themselves converge to the direct strategies, *i.e.*, $\boldsymbol{x}_i^{(t)} \to \boldsymbol{o}_i$, in iterates or in averages. They may differ on nodes off the equilibrium path. Instead, the requirement defined by (S2) is that the *outcome distribution over terminal nodes* converges to that of the equilibrium. Similarly, (S1) refers to the *realized* payments $p_i^{(t)}(\boldsymbol{x}^{(t)})$, not the *maximum offered payment* $\max_{\boldsymbol{x}\in\mathcal{X}} p_i^{(t)}(\boldsymbol{x})$.

For now, we assume that the pure Nash equilibrium is part of the instance, and therefore our only task is to steer the agents toward it. In Section 7.6 we show how our steering algorithms can be extended to other equilibrium concepts such as *mixed* or *(Bayes-)correlated* equilibria, and to the case where the mediator needs to compute the equilibrium.

The mediator does not know anything about how the players pick their strategies, except that they will have regret bounded by a function that vanishes in the limit and is known to the mediator. This condition is a commonly adopted behavioral assumption [49, 178, 231]. The regret of Player $i \in [n]$ in this context is defined as

$$\textsc{Reg}_{\mathcal{X}_i}(T) := \frac{1}{P+1}\left[\max_{\boldsymbol{x}_i^*\in\mathcal{X}_i} \sum_{t=1}^{T} v_i^{(t)}(\boldsymbol{x}_i^*) - \sum_{t=1}^{T} v_i^{(t)}(\boldsymbol{x}_i^{(t)})\right].$$

That is, regret takes into account the payment functions offered to that player. (The division by $1/(P+1)$ is for normalization, since $v_i^{(t)}$s has range $[0, P+1]$.)

How large payments are needed to achieve (S1) and (S2)? If the mediator could provide totally unconstrained payments, it could enforce any arbitrary outcome. On the other hand, if the total payments are restricted to be bounded, the steering problem is information-theoretically impossible:

Proposition 7.9. *There exists a game and some function $R(T) = O(\sqrt{T})$ such that, for all $B \geq 0$, the steering problem is impossible if we add the constraint $\sum_{t=1}^{\infty}\sum_{i=1}^{n} p_i^{(t)}(\boldsymbol{x}^{(t)}) \leq B$.*

Proof. Suppose that the mediator's goal is for the players to coordinate on the equilibrium (B, B) in the coordination 2-player game with the following payoff matrix.

	A	B
A	0.5, 0.5	0,0
B	0,0	1,1

Set $R(T) = 2\sqrt{T}$. We will show that, regardless of the mediator's strategy, it is possible for the players to play (A, A) for all but finitely many rounds.

Suppose the players play as follows. Let $\Gamma^{(t)}$ be the game at time t induced by the mediator's payoff function $p^{(t)}$. For the first B^2 rounds, play an arbitrary Nash equilibrium of $\Gamma^{(t)}$. After that, if (A, A) is a Nash equilibrium of $\Gamma^{(t)}$, play it. Otherwise, play a strategy profile $\boldsymbol{x}^{(t)}$ for which $\sum_{i=1}^{n} p_i^{(t)}(\boldsymbol{x}^{(t)}) > \frac{1}{2}$ (Such a strategy profile must exist, for otherwise (A, A) would be a Nash equilibrium).

The total regret of the players after T rounds is (at most) 0 for $T \le B^2$, since we have assumed that they are playing a Nash equilibrium of $\Gamma^{(t)}$, and at most $(P+1)k$ for $T > B^2$, where k is the number of times that the final case triggers, since the reward range of $\Gamma^{(t)}$ is at most $[0, P+1]$. But the final case can only trigger at most $2B$ times since the mediator only has a total budget of B. Therefore, the regret is bounded by $2(P+1)\sqrt{T}/(P+1) = 2\sqrt{T}$ for any T, and for all but $2B + B^2$ rounds, the players are playing a suboptimal equilibrium. So, desideratum (S2) in Definition 7.8 cannot be satisfied. □

Hence, a weaker requirement on the size of the payments is needed. Between these extremes, one may allow the *total* payment to be unbounded, but insist that the *average* payment per round must vanish in the limit.

7.4 Steering in Normal-Form Games

We start with the simpler setting of *normal-form games*, that is, extensive-form games in which every player has one information set, and the set of histories correspond precisely to the set of pure profiles. This setting is much simpler than the general extensive-form setting (we consider in the next section), and we can appeal to a special case of a result in the literature [219].

Proposition 7.10 (Costless implementation of pure Nash, special case of k-implementation, [219]). *Let $\boldsymbol{o}$ be a pure Nash equilibrium in a normal-form game. Then there exist functions $p_i^* : \mathcal{X}_1 \times \cdots \times \mathcal{X}_n \to [0,1]$, with $p_i^*(\boldsymbol{o}) = 0$, such that in the game with utilities $v_i := u_i + p_i^*$, the profile $\boldsymbol{o}$ is weakly dominant: $v_i(\boldsymbol{o}_i, \boldsymbol{x}_{-i}) \ge v_i(\boldsymbol{x}_i, \boldsymbol{x}_{-i})$ for every profile $\boldsymbol{x}$.*

Proof. The payment function

$$p_i^*(\boldsymbol{x}) := (\boldsymbol{o}_i^\top \boldsymbol{x}_i)\Big(1 - \prod_{j \ne i} \boldsymbol{o}_j^\top \boldsymbol{x}_j\Big),$$

which on pure profiles $\boldsymbol{x}$ returns 1 if and only if $\boldsymbol{x}_i = \boldsymbol{o}_i$ and $\boldsymbol{x}_j \ne \boldsymbol{o}_j$ for some $j \ne i$ makes equilibrium play weakly dominant. □

This is *almost* enough for steering: the only problem is that $\boldsymbol{o}$ is only *weakly* dominant, so no-regret players *may* play other strategies than $\boldsymbol{o}$. This can be fixed by adding a small reward $\alpha \ll 1$ for playing $\boldsymbol{o}_i$. That is, we set

$$p_i(\boldsymbol{x}) := \alpha \boldsymbol{o}_i^\top \boldsymbol{x}_i + p_i^*(\boldsymbol{x}) = (\boldsymbol{o}_i^\top \boldsymbol{x}_i)\Big(\alpha + 1 - \prod_{j \ne i} \boldsymbol{o}_j^\top \boldsymbol{x}_j\Big). \tag{7.1}$$

On a high level, the structure of the payment function guarantees that the average strategy of any no-regret learner $i \in [n]$ should be approaching the direct strategy $\boldsymbol{o}_i$ by making $\boldsymbol{o}_i$ the strictly dominant strategy of player i. At the same time, it is possible to ensure that the average payment will also be vanishing by appropriately selecting parameter α. With an appropriate choice of α, this is enough to solve the steering problem for normal-form games:

Theorem 7.11 (Normal-form steering)**.** *Let $p_i(\boldsymbol{x})$ be defined as in (7.1), set $\alpha = \sqrt{\epsilon}$, where $\epsilon := nR(T)/T$, and let T be large enough that $\alpha \le 1$. Then players will be steered toward equilibrium, with both payments and directness gap bounded by $2\sqrt{\epsilon}$.*

Proof. By construction of the payments, the utility for player i is at least α higher for playing $\boldsymbol{o}_i$ than for any other action, regardless of the actions of the other players. Let $\epsilon := nR(T)/T$ and $\delta_i^{(t)} := 1 - \boldsymbol{o}_i^\top \boldsymbol{x}_i^{(t)}$. Then the above property ensured by the payments implies that $R(T)/T = \epsilon/n \ge \alpha\, \mathbb{E}_{t\in[T]}\, \delta_i^{(t)}$. Let z^* be the terminal node induced by profile $\boldsymbol{o}$. Then the directness gap is

$$2\mathop{\mathbb{E}}_t \left[1 - \hat{\boldsymbol{x}}^{(t)}(z^*)\right] = 2 - 2\mathop{\mathbb{E}}_t \prod_i (1 - \delta_i^{(t)}) \le 2\mathop{\mathbb{E}}_t \sum_i \delta_i^{(t)} \le 2\epsilon/\alpha,$$

and the payments are bounded by

$$\mathop{\mathbb{E}}_t p_i(\boldsymbol{x}) \le \alpha + \mathop{\mathbb{E}}_t \left[1 - \prod_{j\ne i} (1 - \delta_i^{(t)})\right] \le \alpha + \epsilon/\alpha.$$

So, taking $\alpha = \sqrt{\epsilon}$ completes the proof. □

We note that no effort was made throughout this chapter to optimize the game-dependent or constant factors, so long as they remained polynomial in $|\mathcal{Z}|$—they can very likely be improved.

7.5 Steering in Extensive-Form Games

This section considers steering in extensive-form games. We will first consider a model in which steering payments can condition on full player strategies (Section 7.5.1). Next, we consider a model in which only realized trajectories are considered (Section 7.5.2).

Tbere are two main reassons why the extensive-form version of the steering problem is significantly more challenging than the normal-form version.

First, in extensive form, the strategy spaces of the players are no longer simplices. Therefore, if we wanted to write a payment function p_i with the property that $p_i(\boldsymbol{x}) = \alpha\mathbb{1}\{\boldsymbol{x} = \boldsymbol{o}\} + \mathbb{1}\{\boldsymbol{x}_i = \boldsymbol{o}_i; \exists j\ \boldsymbol{x}_j \ne \boldsymbol{o}_j\}$ for pure $\boldsymbol{x}$ (which is what was needed by Theorem 7.11), such a function would not be linear (or even convex) in player i's strategy $\boldsymbol{x}_i \in \mathcal{X}_i$ (which is a sequence-form strategy, not a distribution over pure strategies). As such, even the meaning of extensive-form regret minimization becomes suspect in this setting.

Second, in extensive form, a desirable property would be that the mediator give payments conditioned only on what actually happens in gameplay, *not* on the players' full strategies—in particular, if a particular information set is not reached during play, the mediator should not know what action the player *would have* selected at that information set. We will call this the *trajectory* setting, and distinguish it from the *full-feedback* setting, where the mediator observes the players' full strategies.[7.3] This distinction is meaningless in the normal-form setting: since terminal nodes in normal form correspond to (pure) profiles, observing gameplay is equivalent to observing strategies. (We will discuss this point in more detail when we introduce the trajectory-feedback setting in Section 7.5.2.)

7.5.1 Steering with Full Feedback

In this section, we introduce a steering algorithm for extensive-form games under full feedback, summarized below.

Definition 7.12 (FullFeedbackSteer)**.** At every round, set the payment function $p_i(\boldsymbol{x}_i, \boldsymbol{x}_{-i})$ as

$$\underbrace{\alpha \boldsymbol{o}_i^\top \boldsymbol{x}_i}_{\text{directness bonus}} + \underbrace{[u_i(\boldsymbol{x}_i, \boldsymbol{o}_{-i}) - u_i(\boldsymbol{x}_i, \boldsymbol{x}_{-i})]}_{\text{sandboxing payments}} - \underbrace{\min_{\boldsymbol{x}_i' \in \mathcal{X}_i} \left[u_i(\boldsymbol{x}_i', \boldsymbol{o}_{-i}) - u_i(\boldsymbol{x}_i', \boldsymbol{x}_{-i})\right]}_{\text{payment to ensure nonnegativity}}, \tag{7.2}$$

where $\alpha \le 1/|\mathcal{Z}|$ is a hyperparameter that we will select appropriately.

By construction, p_i satisfies the conditions of the steering problem (Definition 7.8): it is linear in $\boldsymbol{x}_i$, continuous in $\boldsymbol{x}_{-i}$, nonnegative, and bounded by an absolute constant (namely, 3). The payment function defined above has three terms:

1. The first term is a *reward for directness*: a player gets a reward proportional to α if it plays $\boldsymbol{o}_i$.
2. The second term *compensates the player* for the indirectness of other players. That is, the second term ensures that players' rewards are *as if* the other players had acted directly.
3. The final term simply ensures that the overall expression is nonnegative.

We claim that this protocol solves the basic version of the steering problem, as formalized below.

Theorem 7.13. *Set $\alpha = \sqrt{\epsilon}$, where $\epsilon := 4nR(T)/T$, and let T be large enough that $\alpha \le 1/|\mathcal{Z}|$. Then,* FullFeedbackSteer *results in average realized payments and directness gap at most $3|\mathcal{Z}|\sqrt{\epsilon}$.*

Proof. We start with a useful lemma.

Lemma 7.14. *Let $\bar{\boldsymbol{x}}_i := \mathbb{E}_{t\in[T]}\, \boldsymbol{x}_i^{(t)}$ for any player $i \in [n]$ and $\delta := \sum_{i=1}^n \boldsymbol{o}_i^\top(\boldsymbol{o}_i - \bar{\boldsymbol{x}}_i)$. Then, $\mathbb{E}_{t\in[T]} \left\|\hat{\boldsymbol{x}}_N^{(t)} - \hat{\boldsymbol{o}}_N\right\|_1 \le |\mathcal{Z}|\delta$ for every $N \subseteq [n]$. Moreover, if the payments are defined according to (7.2), the average payment to every player can be bounded by $\mathbb{E}_{t\in[T]}\, p_i(\boldsymbol{x}^{(t)}) \le |\mathcal{Z}|(2\delta + \alpha)$.*

[7.3] To be clear, the settings are differentiated by what the *mediator* observes, not what the *players* observe. That is, it is valid to consider the full-feedback steering setting with players running bandit-feedback regret minimizers, or the trajectory-feedback steering setting with players running full-feedback regret minimizing algorithms.

Proof. Let $\delta_i := \boldsymbol{o}_i^\top(\boldsymbol{o}_i - \bar{\boldsymbol{x}}_i)$ for any player $i \in [n]$. Then, we have that

$$\min_{z:\boldsymbol{o}_i(z)=1} \bar{\boldsymbol{x}}_i(z) \geq 1 - \delta_i,$$

which in turn implies that $\max_{z:\boldsymbol{o}_i(z)=0} \bar{\boldsymbol{x}}_i(z) \leq \delta_i$. Now let $N \subseteq [n]$. If $z \in \mathcal{Z}$ is such that $\boldsymbol{o}_N(z) = 1$,

$$\bar{\boldsymbol{x}}_N(z) = \mathop{\mathbb{E}}_{t\in[T]} \boldsymbol{x}_N^{(t)}(z) = \mathop{\mathbb{E}}_{t\in[T]} \prod_{j\in N} \boldsymbol{x}_j^{(t)}(z) \geq \mathop{\mathbb{E}}_{t\in[T]} \prod_{j\in N} \left(1 - \delta_j\right) \geq 1 - \sum_{j\in N} \delta_j = 1 - \delta.$$

Further, if $\boldsymbol{o}_j(z) = 0$ for some $j \in N$,

$$\bar{\boldsymbol{x}}_N(z) \leq \bar{\boldsymbol{x}}_j(z) \leq \delta_j \leq \delta.$$

Thus,

$$\begin{aligned}\mathop{\mathbb{E}}_{t\in[T]} \left\| \hat{\boldsymbol{x}}_N^{(t)} - \hat{\boldsymbol{o}}_N \right\|_1 &= \mathop{\mathbb{E}}_{t\in[T]} \left(\sum_{z:\hat{\boldsymbol{o}}_N(z)=0} (\hat{\boldsymbol{x}}_N^{(t)}(z) - \hat{\boldsymbol{o}}_N(z)) + \sum_{z:\hat{\boldsymbol{o}}_N(z)=1} (\hat{\boldsymbol{o}}_N(z) - \hat{\boldsymbol{x}}_N^{(t)}(z)) \right) \\ &= \left\| \mathop{\mathbb{E}}_{t\in[T]} \hat{\boldsymbol{x}}_N^{(t)} - \hat{\boldsymbol{o}}_N \right\|_1 = \|\bar{\boldsymbol{x}}_N - \hat{\boldsymbol{o}}_N\|_1 \leq |\mathcal{Z}|\delta, \qquad (7.3)\end{aligned}$$

since we have shown that $|\bar{\boldsymbol{x}}_N(z) - \hat{\boldsymbol{o}}_N(z)| \leq \delta$ for any $z \in \mathcal{Z}$. This establishes the first part of the claim. Next, the average payments (7.2) can by bounded for any player $i \in [n]$ as

$$\begin{aligned}\mathop{\mathbb{E}}_{t\in[T]} \Big[\left[u_i(\boldsymbol{x}_i^{(t)}, \boldsymbol{o}_{-i}) - u_i(\boldsymbol{x}_i^{(t)}, \boldsymbol{x}_{-i}^{(t)}) \right] & \\ - \min_{\boldsymbol{x}_i'\in X_i} \left[u_i(\boldsymbol{x}_i', \boldsymbol{o}_{-i}) - u_i(\boldsymbol{x}_i', \boldsymbol{x}_{-i}^{(t)}) \right] + \alpha \boldsymbol{o}_i^\top \boldsymbol{x}_i^{(t)} \Big] & \\ \leq 2 \mathop{\mathbb{E}}_{t\in[T]} \left\| \hat{\boldsymbol{x}}_{-i}^{(t)} - \hat{\boldsymbol{o}}_{-i} \right\|_1 + \alpha|\mathcal{Z}| \leq |\mathcal{Z}|(2\delta + \alpha),\end{aligned}$$

where we used the normalization assumption $|u_i(\cdot)| \leq 1$, and the fact that $\boldsymbol{o}_i^\top \boldsymbol{x}_i^{(t)} \leq |\mathcal{Z}|$. This concludes the proof. □

The utility of each player $i \in [n]$ reads

$$v_i(\boldsymbol{x}_i, \boldsymbol{x}_{-i}) := \alpha \boldsymbol{o}_i^\top \boldsymbol{x}_i + u_i(\boldsymbol{x}_i, \boldsymbol{o}_{-i}) - \min_{\boldsymbol{x}_i'\in X_i} [u_i(\boldsymbol{x}_i', \boldsymbol{o}_{-i}) - u_i(\boldsymbol{x}_i', \boldsymbol{x}_{-i})].$$

Given that $\boldsymbol{o}$ is an equilibrium, it follows that $\boldsymbol{o}_i$ is a strict best response for any player $i \in [n]$. That is, the regret of each player $i \in [n]$ after T iterations can be lower bounded as

$$\sum_{t=1}^{T} \left(\alpha \boldsymbol{o}_i^\top (\boldsymbol{o}_i - \boldsymbol{x}_i^{(t)}) + u_i(\boldsymbol{o}) - u_i(\boldsymbol{x}_i^{(t)}, \boldsymbol{o}_{-i}) \right) \geq \alpha T \boldsymbol{o}_i^\top (\boldsymbol{o}_i - \bar{\boldsymbol{x}}_i),$$

where we used that $u_i(\boldsymbol{o}) - u_i(\boldsymbol{x}_i^{(t)}, \boldsymbol{o}_{-i}) \geq 0$ since $\boldsymbol{o}$ is an equilibrium. Thus,

$$\sum_{i=1}^{n} \boldsymbol{o}_i^\top (\boldsymbol{o}_i - \bar{\boldsymbol{x}}_i) \leq \frac{nR(T)}{\alpha T} = \frac{\epsilon}{\alpha}.$$

We can now apply Lemma 7.14 to obtain that $\mathbb{E}_{t \in [T]} \left\| \hat{\boldsymbol{x}}^{(t)} - \hat{\boldsymbol{o}} \right\|_1 \leq |\mathcal{Z}|\delta$, where $\delta := \epsilon/\alpha$. Thus, the directness gap is bounded by

$$\left\| \mathop{\mathbb{E}}_t \hat{\boldsymbol{x}}^{(t)} - \hat{\boldsymbol{o}} \right\|_1 = \mathop{\mathbb{E}}_t \left\| \hat{\boldsymbol{x}}^{(t)} - \hat{\boldsymbol{o}} \right\|_1 \leq \frac{n|\mathcal{Z}|R(T)}{\alpha T},$$

where the first equality follows because $\hat{\boldsymbol{o}}$ is an extreme point of X (as in (7.3)). Furthermore, by Lemma 7.14, the payment to each player $i \in [n]$ can be bounded by

$$2|\mathcal{Z}|(2\delta + \alpha) = 2|\mathcal{Z}|\frac{\epsilon}{\alpha} + |\mathcal{Z}|\alpha.$$

As a result, setting $\alpha = \sqrt{\epsilon}$ for T sufficiently large so that $\alpha \leq 1/|\mathcal{Z}|$, we guarantee that the payment to each player is bounded by $3n|\mathcal{Z}|\sqrt{\epsilon}$ and the directness gap is bounded by $|\mathcal{Z}|\sqrt{\epsilon}$, as desired. □

7.5.2 Steering with Trajectory Feedback

In FullFeedbackSteer, payments depend on full strategies $\boldsymbol{x}$, not the realized game trajectories. In particular, the mediator in Theorem 7.13 observes what the players *would have played* even at infosets that other players avoid. To allow for an algorithm that works without knowledge of full strategies, $p_i^{(t)}$ must be structured so that it could be induced by a payment function that only gives payments for terminal nodes reached during play. To this end, we now formalize *trajectory-feedback steering*.

Definition 7.15 (Trajectory-feedback steering problem)**.** Let Γ be an extensive-form game in which rewards are bounded in $[0, 1]$ for all players. Let $\boldsymbol{o}$ be an arbitrary pure-strategy Nash equilibrium of Γ. The mediator knows Γ and a regret bound $R(T) = o(T)$. At each $t \in [T]$, the mediator selects a payment function $q_i^{(t)} : \mathcal{Z} \to [0, P]$. The players select strategies $\boldsymbol{x}_i^{(t)}$. A terminal node $z^{(t)} \sim \boldsymbol{x}^{(t)}$ is sampled, and all agents observe the terminal node that was reached, $z^{(t)}$. The players get payments $q_i^{(t)}(z^{(t)})$, so that their expected payment is $p_i^{(t)}(\boldsymbol{x}) := \mathbb{E}_{z \sim \boldsymbol{x}}\, q_i^{(t)}(z)$. The desiderata are as in Definition 7.8.

The trajectory-feedback steering problem is more difficult than the full-feedback steering problem in two ways. First, as discussed above, the mediator does not observe the strategies $\boldsymbol{x}$, only a terminal node $z^{(t)} \sim \boldsymbol{x}$. Second, the form of the payment function $q_i^{(t)} : \mathcal{Z} \to [0, P]$ is restricted: this is already sufficient to rule out FullFeedbackSteer. Indeed, p_i as defined in (7.2) cannot be written in the form $\mathbb{E}_{z \sim \boldsymbol{x}}\, q_i(z)$: $p_i(\boldsymbol{x}_i, \boldsymbol{x}_{-i})$ is nonlinear in $\boldsymbol{x}_{-i}$ due to the nonnegativity-ensuring payments, whereas every function of the form $\mathbb{E}_{z \sim \boldsymbol{x}}\, q_i(z)$ will be linear in each player's strategy.

We remark that, despite the above algorithm containing a sampling step, the payment function is defined *deterministically*: the payment is defined as the *expected value* $p_i^{(t)}(\boldsymbol{x}) := \mathbb{E}_{z\sim\boldsymbol{x}}\, q_i^{(t)}(z)$. Thus, the theorem statements in this section will also be deterministic.

For normal-form games, the payments p_i defined by (7.1) already satisfy the condition of trajectory-feedback steering. In particular, if $\boldsymbol{a} = (a_1, \dots, a_n)$ is the joint action, we have

$$p_i(\boldsymbol{x}) = \mathop{\mathbb{E}}_{\boldsymbol{a}\sim\boldsymbol{x}} \left[\alpha \mathbb{1}\{\boldsymbol{a} = \boldsymbol{o}\} + \mathbb{1}\big\{\boldsymbol{a}_i = \boldsymbol{o}_i; \exists j\ \boldsymbol{a}_j \neq \boldsymbol{o}_j\big\}\right].$$

Therefore, for normal-form games, Theorem 7.11 applies to both full-feedback steering and trajectory-feedback steering, and we have no need to distinguish between the two.

7.5.3 Bayesian Games

In this section, we will discuss the special case of Bayesian games. For notation, we will let θ_i denote the type of player i, and $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_n)$ denote the joint type. Then a terminal node z can be identified by a tuple $(\boldsymbol{\theta}, \boldsymbol{a})$ where $\boldsymbol{a}$ is the joint action. Now consider the payment function

$$p_i(\boldsymbol{x}) = \mathop{\mathbb{E}}_{(\boldsymbol{\theta},\boldsymbol{a})\sim\boldsymbol{x}} \left[\alpha \mathbb{1}\{\boldsymbol{a}_i = \boldsymbol{o}_i(\theta_i) \forall i\} + \mathbb{1}\big\{\boldsymbol{a}_i = \boldsymbol{o}_i(\theta_i); \exists j\ \boldsymbol{a}_j \neq \boldsymbol{o}_j(\theta_j)\big\}\right] \tag{7.4}$$

where $\boldsymbol{o}_i(\theta_i)$ is the pure action played by $\boldsymbol{o}_i$ with type θ_i. We claim that this payment function steers players toward the target profile $\boldsymbol{o}$. Indeed, we have the following result.

Theorem 7.16 (Trajectory-feedback steering in Bayesian games)**.** *Set payments as in (7.4), with hyperparameters $\alpha = \sqrt{\epsilon}$ and $\epsilon = nR(T)/T$. Then players will be steered toward equilibrium with both payments and directness gap bounded by $2\sqrt{\epsilon}$.*

Proof. As in the normal-form case, the pure strategy $\boldsymbol{o}_i$ is dominant for each player i: whenever player i has type θ_i, action $\boldsymbol{o}_i(\theta_i)$ has expected utility α higher than any other action. That is, playing any other action incurs regret at least $\alpha \cdot p_i(\theta_i)$ in expectation, where $p_i(\theta_i)$ is the prior probability of type θ_i. Thus, we have

$$\mathop{\mathbb{E}}_t \sum_{\boldsymbol{\theta}} p(\boldsymbol{\theta})\alpha \cdot \left(1 - \boldsymbol{x}_i^{(t)}(\boldsymbol{o}_i(\theta_i))\right) \leq \frac{R(T)}{T}.$$

Summing over all players, we have

$$\begin{aligned}
\frac{nR(T)}{T} &\geq \mathop{\mathbb{E}}_t \sum_{\boldsymbol{\theta}} p_i(\boldsymbol{\theta})\alpha \sum_{i=1}^n (1 - \boldsymbol{x}_i^{(t)}(\boldsymbol{o}_i(\theta_i))) \\
&\geq \mathop{\mathbb{E}}_t \sum_{\boldsymbol{\theta}} p_i(\boldsymbol{\theta})\alpha \left(1 - \prod_{i=1}^n \boldsymbol{x}_i^{(t)}(\boldsymbol{o}_i(\theta_i))\right) \\
&= \frac{1}{2}\alpha \mathop{\mathbb{E}}_t \left\| \hat{\boldsymbol{x}}^{(t)} - \boldsymbol{o} \right\|_1
\end{aligned}$$

so the directness gap is bounded by $2nR(T)/\alpha T = 2\epsilon/\alpha = 2\sqrt{\epsilon}$. Moreover, the payments are bounded by $\alpha + \mathbb{E}_t\, \delta_i^{(t)} = \alpha + \epsilon/\alpha$ whee $\delta_i^{(t)} = \frac{1}{2}\left\|\hat{\boldsymbol{x}}^{(t)} - \boldsymbol{o}\right\|_1$ as before. □

This result is a strict generalization of Theorem 7.11.

7.5.4 Lower Bound for Extensive-Form Games

However, in extensive form, as discussed above, the story is not so clear. Unlike in the full-feedback or normal-form settings, in the trajectory-feedback setting, steering is impossible in the general case in the sense that per-iteration payments bounded by any constant do not suffice.

Theorem 7.17. *For every $P > 0$, there exists an extensive-form game Γ with $O(P)$ players, $O(P^2)$ nodes, and rewards bounded in $[0, 1]$ such that, with payments $q_i^{(t)} : \mathcal{Z} \to [0, P]$, it is impossible to steer players to the welfare-maximizing Nash equilibrium, even when $R(T) = 0$.*

For intuition, consider the extensive-form game in Figure 7.3, which can be seen as a three-player version of Stag Hunt. Players who play Hare (H) get a value of $1/2$ (up to constants); in addition, if all three players play Stag (S), they all get expected value 1. The welfare-maximizing equilibrium is “everyone plays Stag”, but “everyone plays Hare” is also an equilibrium. In addition, if all players are playing Hare, the only way for the mediator to convince a player to play Stag without accidentally also paying players in the Stag equilibrium is to pay players at one of the three boxed nodes. But those three nodes are only reached with probability $1/n$ as often as the three nodes on the left, so the mediator would have to give a bonus of more than $n/2$. The full proof essentially works by deriving an algorithm that the players could use to exploit this dilemma to achieve either large payments or bad convergence rate, generalizing the example to $n > 3$, and taking $n = \Theta(P)$.

Proof. For any $n > 0$, consider the following n-player extensive-form game Γ, which has $O(n^2)$ nodes. Every player has only a single information set with two actions, and we will (for good reason, as we will see later) refer to the actions as Stag and Hare. Chance first picks some $j \in [n] \cup \{\perp\}$ uniformly at random.

If $j \neq \perp$, then player j plays an action (which is either Stag or Hare). If i plays Hare, it gets utility $1/2$; otherwise, it gets utility 0. All other players get utility 0.

If $k = \perp$, chance samples another player k uniformly at random from $[n]$. Then, in the order $k, k + 1, \dots, n, 1, 2, \dots, k - 1$, the players play their actions. If any player at any point plays Hare, then the game ends and all players get 0. If all players play Stag, then all players get 1.

The normal form of this game is an n-player generalization of the Stag Hunt game: if all players play Stag then all players have (expected) payoff $1/(n + 1)$; if any player plays Hare then every player has expected payoff $(1/2)/(n + 1)$ for playing Hare and 0 for playing Stag. In particular, the welfare-optimal profile, “everyone plays Stag”, is a Nash equilibrium, and hence is also the welfare-optimal EFCE, with social welfare $n/(n + 1)$. “Everyone plays Hare” is also an equilibrium, with social welfare $(1/2)n/(n + 1)$. The game tree when $n = 3$ is depicted in Figure 7.3.

Intuitively, the rest of the proof works as follows. Suppose that all players are currently playing

Hare. The mediator needs to incentivize players to play Stag, but it has a dilemma. It cannot give a large payment to i for playing Stag when $j = i$—then the average payment for each player would diverge if the players were to move to the Stag equilibrium. The only other location that the mediator could possibly give a payment to i is when $j = \perp$, $k = i$, player i plays Stag, and the next player plays Hare. But this node is only reached with probability $O(1/n^2)$—therefore, to outweigh i's current incentive of $\Theta(1/n)$ of playing Hare, the payment at this node would have to be $\Theta(n)$, at which point taking $n = \Theta(P)$ would complete the proof.

We now formalize this intuition. Take $n = \lceil 4P \rceil$. Consider players who play as follows. At each timestep t, the players consider the extensive-form game $\Gamma^{(t)}$ induced by adding the payment functions $q_i^{(t)}$ that the mediator would play, and ignoring mediator recommendations. That is, $\Gamma^{(t)}$ is identical to Γ except that $q_i^{(t)}$ has been added to player i's utility function. If "everyone plays Hare" is a Nash equilibrium of $\Gamma^{(t)}$, all players play Hare. Otherwise, the players play according to an arbitrary Nash equilibrium of $\Gamma^{(t)}$.

Since the players are playing according to a Nash equilibrium at every step, they all have regret at most 0. Now consider two cases.

1. There is a player i such that plays Stag with probability less than $1/2$. Then the social welfare is at most $(3/4)n/(n+1)$, which is lower than the optimal social welfare by $(1/4)n/(n+1)$.
2. All players play Stag with probability at least $1/2$. Then, in particular, "everyone plays Hare" is not a Nash equilibrium in $\Gamma^{(t)}$. So, if everyone were to play Hare, there is some player i who would rather deviate and play Stag. Thus, the mediator must be giving an expected payment to i of at least $(1/2)/(n+1)$. As discussed above, there are only two nodes z for which the setting of $q_i^{(t)}(z)$ increases i's utility for playing Stag relative to its utility for playing Hare. The first is when $j = \perp$, $k = i$, i plays Stag, and the next player plays Hare. Since $P \le n/4$ and this node occurs with probability $1/(n(n+1))$, even the maximum payment at this node contributes at most $(1/4)/(n+1)$ to the expected payment. Therefore, the remainder of the payment, $(1/2)/(n+1)$, must be given when $j = i$ and then i plays Stag. But Player i plays Stag with probability at least $1/2$, so i's observed expected payment is at least $(1/4)/(n+1)$.

Therefore, we have

$$\left(u_0^* - \mathbb{E}\, u_0(z^{(t)})\right) + \mathbb{E} \sum_{i \in [n]} q_i^{(t)}(z^{(t)}) \ge \frac{1}{4(n+1)}$$

where u_0 is the social welfare function, so it is impossible for both quantities to tend to 0 as $T \to \infty$. □

7.5.5 Upper Bound for Extensive-Form Games

To circumvent the lower bound in Theorem 7.17, in this subsection, we allow the payment bound $P \ge 1$ to depend on both the time limit T and the game.

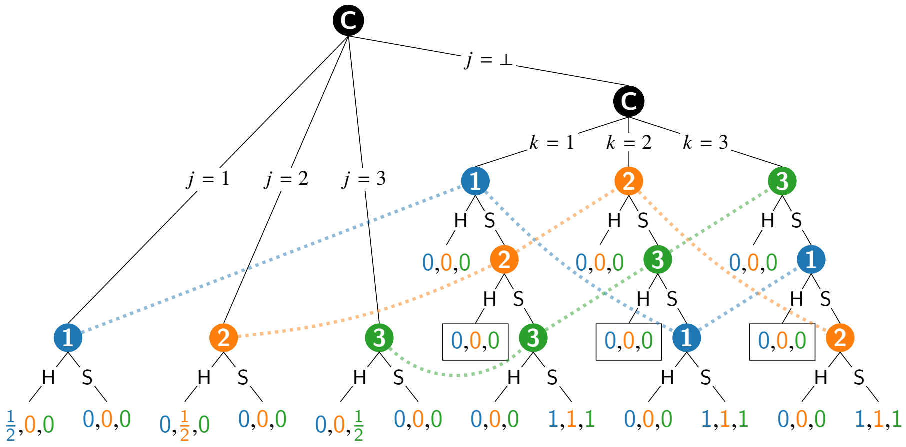


Figure 7.3: *The counterexample for Theorem 7.17, for $n = 3$. Chance always plays uniformly at random. Infosets are linked by dotted lines (all nodes belonging to the same player are in the same infoset).*

Definition 7.18 (TrajectorySteer)**.** Let α, P be hyperparameters. Then, for all rounds $t = 1, \dots, T$, sample $z \sim \boldsymbol{x}^{(t)}$ and pay players as follows. If all players have been direct (*i.e.*, if $\hat{\boldsymbol{o}}(z) = 1$), pay all players α. If at least one player has not been direct, pay P to all players who have been direct. That is, set $q_i^{(t)}(z^{(t)}) = \alpha\hat{\boldsymbol{o}}(z) + P\boldsymbol{o}_i(z)(1 - \hat{\boldsymbol{o}}(z))$.

Theorem 7.19. *Set the hyperparameters $\alpha = 4|\mathcal{Z}|^{1/2}\epsilon^{1/4}$ and $P = 2|\mathcal{Z}|^{1/2}\epsilon^{-1/4}$, where $\epsilon := R(T)/T$, and let T be large enough that $\alpha \leq 1$. Then, running* TrajectorySteer *for T rounds results in average realized payments bounded by $8|\mathcal{Z}|^{1/2}\epsilon^{1/4}$, and directness gap by $2\epsilon^{1/2}$.*

As alluded to in the introduction, the proof of this result is more involved than those for previous results, because one cannot simply make the target equilibrium dominant as in the full-feedback case. One may hope that—as in FullFeedbackSteer—the desired equilibrium can be made dominant by adding payments. In fact, a sort of "chicken-and-egg" problem arises: (S2) requires that all players converge to equilibrium. But for this to happen, other players' strategies must first converge to equilibrium so that i's incentives are as they would be in equilibrium. The main challenge in the proof of Theorem 7.19 is therefore to carefully set the hyperparameters to achieve convergence despite these apparent problems.

Proof. We use the following notation.

- The set D_S is the set of nodes at which all players in set S have played directly: $D_S = \{z \in \mathcal{Z} : \boldsymbol{o}_i(z) = 1 \forall i \in S\}$. The set $D'_S = Z \setminus D_S$ is its complement.
- $\boldsymbol{x}$ is a random variable for the correlated strategy profile played by all players through the T timesteps. That is, $\boldsymbol{x}$ is a uniform sample from $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(T)}\}$.
- $\pi(S|\boldsymbol{x})$ is the probability that a terminal node from set S is reached, given that the

mediator plays $\boldsymbol{\xi}$ and the players play the (possibly correlated) strategy profile $\boldsymbol{x}$. That is, $\pi(S|\boldsymbol{x}) = \Pr_{z\sim(\boldsymbol{\xi},\boldsymbol{x})}[z \in S]$.

- $\tilde{u}_i(\boldsymbol{x}) = u_i(\boldsymbol{x}) + \mathbb{E}_{z\sim(\boldsymbol{\xi},\boldsymbol{x})}\, q_i(z)$ is the expected utility for player i, including payment, under profile $(\boldsymbol{\xi}, \boldsymbol{x})$.
- $u_i(\boldsymbol{y}_i - \boldsymbol{x}_i, \boldsymbol{x}_{-i}) := u_i(\boldsymbol{y}_i, \boldsymbol{x}_{-i}) - u_i(\boldsymbol{x}_i, \boldsymbol{x}_{-i})$ is player i's advantage for playing $\boldsymbol{y}_i$ instead of $\boldsymbol{x}$. $\tilde{u}_i(\boldsymbol{y}_i - \boldsymbol{x}_i, \boldsymbol{x}_{-i})$ and $\pi(z|\boldsymbol{y}_i - \boldsymbol{x}_i, \boldsymbol{x}_{-i})$ are defined similarly.

Let $\epsilon = R(T)/T$. Then after T timesteps, since the players are no-regret learners, their average joint strategy profile will be an $P\epsilon$-NFCCE of the extensive-form game with the payments added.

Intuitively, the proof will go as follows. We will show that, for P sufficiently large, each player's incentive to be direct will be *at least* as great as it would have been if everyone else were also direct, plus α. Then it will follow from the fact that $\boldsymbol{\xi}$ is an equilibrium, and picking $\alpha \gg P\epsilon$, that all players must therefore be direct. We first prove a lemma. Informally, the lemma states that, when any player i deviates, all other players must be direct.

Lemma 7.20. *Let z be any node with $\boldsymbol{o}_i(z) = 0$, that is, any node at which player i has deviated. Then $|\pi(z|\boldsymbol{x}_i, \boldsymbol{o}_{-i} - \boldsymbol{x}_{-i})| \le \gamma := n\epsilon + \sum_j \delta_j/P$, where $\delta_j := u_j(\boldsymbol{x}_j - \boldsymbol{o}_j, \boldsymbol{x}_{-j})$ is player j's current deviation benefit.*

Proof. Assume without loss of generality that $i = 1$, and consider two cases.

1. $\boldsymbol{o}_j(z) = 0$ for some $j \neq i$—that is, some other player has also deviated. Then $\pi(z|\boldsymbol{x}_i, \boldsymbol{o}_{-i}) = 0$. Assume for contradiction that $\pi(z|\boldsymbol{x}) > \gamma$. Let $h_i, h_j \prec z$ be the two deviation points—that is, $\boldsymbol{o}_i(h_i) = 1$ but $\boldsymbol{o}_i(h_i a_i) = 0$ where $h_i a_i \preceq z$, and similar for h_j. Suppose without loss that $h_i \prec h_j$. Now consider player j's incentive. If player j were to switch to playing $\boldsymbol{o}_j$, its expected payment increases by at least γP, and its expected utility (sans payment) decreases by δ_j, by definition. When $\gamma \ge \epsilon + \delta_j/P$, this produces a contradiction.
2. $\boldsymbol{o}_j(z) = 1$ for all $j \neq i$. Then $\pi(z|\boldsymbol{x}_i, \boldsymbol{o}_{-i}) \ge \pi(z|\boldsymbol{x})$, so we need to show that $\pi(z|\boldsymbol{x}_i, \boldsymbol{o}_{-i}) - \pi(z|\boldsymbol{x}^{(t)}) \le \gamma$. That is, other players will almost always play to catch player i deviating, whenever possible. Suppose not. Let $h \prec z$ be the point where player i deviated (that is, $\boldsymbol{o}_i(h) = 1$ but $\boldsymbol{o}_i(ha_1) = 0$ where $ha_1 \preceq z$). Let a_0 be the direct action at h. Notice that, for any player $j \neq i$, if j shifts to playing the direct strategy, the probability of leaving the path to ha before reaching ha itself cannot increase by more than $\epsilon + \delta_i/P$: otherwise, player j's expected utility would be increasing by more than δ_i, a contradiction. If all $n-1$ players allocate their deviations in this manner, and even if the remaining $(n-1)\delta_i/P$ probability of leaving path ha is then all allocated to node z, the reach probability of z could not have increased by more than $\sum_j(\epsilon + \delta_j/P)$. Thus, when γ is larger than this value, we have a contradiction. □

The rest of the proof is structured as follows. We will first show, roughly speaking, that *player i's deviation benefit*—that is, its advantage for playing $\boldsymbol{x}_i^{(t)}$ at each timestep t instead of playing

$\boldsymbol{o}_i$—is *smaller* against the opponent strategies $\boldsymbol{x}_{-i}^{(t)}$ than it would be against $\boldsymbol{o}_{-i}^{(t)}$, modulo a small additive error. Then, the proof will follow from the fact that $\boldsymbol{o}$ is an equilibrium against $\boldsymbol{\xi}$, so therefore all players should play according to $\boldsymbol{o}$.

$$
\begin{aligned}
&\tilde{u}_i(\boldsymbol{x}) - \tilde{u}_i(\boldsymbol{x}_i, \boldsymbol{o}_{-i}) \\
&= \sum_{z \in D_i \cap D_{-i}} \tilde{u}_i(z)[\pi(z|\boldsymbol{x}) - \pi(z|\boldsymbol{x}_i, \boldsymbol{o}_{-i})] + \sum_{z \in D_i \cap D'_{-i}} \tilde{u}_i(z)\pi(z|\boldsymbol{x}) \\
&\quad \underbrace{+ \sum_{z \in D'_i} u_i(z)[\pi(z|\boldsymbol{x}) - \pi(z|\boldsymbol{x}_i, \boldsymbol{o}_{-i})]}_{\leq \gamma|\mathcal{Z}|} \\
&\leq \sum_{z \in D_i \cap D_{-i}} \tilde{u}_i(z)[\pi(z|\boldsymbol{x}) - \pi(z|\boldsymbol{x}_i, \boldsymbol{o}_{-i})] + \sum_{z \in D_i \cap D'_{-i}} \tilde{u}_i(z)\pi(z|\boldsymbol{x}) + \gamma|\mathcal{Z}|
\end{aligned}
$$

where we use, in order, the definition of expected utility, the fact that $u_i(z) = \tilde{u}_i(z)$ when $\boldsymbol{o}_i(z) = 0$ and $\pi(z|\boldsymbol{x}) = 0$ whenever $\boldsymbol{x}_i(z) = 0$ for any i, and finally Lemma 7.20. Similarly,

$$
\begin{aligned}
&\tilde{u}_i(\boldsymbol{o}_i, \boldsymbol{x}_{-i}) - \tilde{u}_i(\boldsymbol{o}) \\
&= \sum_{z \in D_i \cap D_{-i}} \tilde{u}_i(z)[\pi(z|\boldsymbol{o}_i, \boldsymbol{x}_{-i}) - \pi(z|\boldsymbol{o})] + \sum_{z \in D_i \cap D'_{-i}} \tilde{u}_i(z)\pi(z|\boldsymbol{o}_i, \boldsymbol{x}_{-i}).
\end{aligned}
$$

Thus,

$$
\begin{aligned}
&P\epsilon - [\tilde{u}_i(\boldsymbol{o}) - \tilde{u}_i(\boldsymbol{x}_i, \boldsymbol{o}_{-i})] \\
&\geq [\tilde{u}_i(\boldsymbol{o}_i, \boldsymbol{x}_{-i}) - \tilde{u}_i(\boldsymbol{x})] - [\tilde{u}_i(\boldsymbol{o}) - \tilde{u}_i(\boldsymbol{x}_i, \boldsymbol{o}_{-i})] \\
&\geq \sum_{z \in D_i \cap D_{-i}} \tilde{u}_i(z) \underbrace{[\pi(z|\boldsymbol{o}_i - \boldsymbol{x}_i, \boldsymbol{x}_{-i}) - \pi(z|\boldsymbol{o}_i - \boldsymbol{x}_i, \boldsymbol{o}_{-i})]}_{\leq 0} \\
&\quad + 2 \sum_{z \in D_i \cap D'_{-i}} \pi(z|\boldsymbol{o}_i - \boldsymbol{x}_i, \boldsymbol{x}_{-i}) - \gamma|\mathcal{Z}| \\
&\geq 2 \sum_{z \in D_i \cap D_{-i}} [\pi(z|\boldsymbol{o}_i - \boldsymbol{x}_i, \boldsymbol{x}_{-i}) - \pi(z|\boldsymbol{o}_i - \boldsymbol{x}_i, \boldsymbol{o}_{-i})] \\
&\quad + 2 \sum_{z \in D_i \cap D'_{-i}} \pi(z|\boldsymbol{o}_i - \boldsymbol{x}_i, \boldsymbol{x}_{-i}) - \gamma|\mathcal{Z}| \\
&= 2\big[\pi(D'_i|\boldsymbol{x}_i, \boldsymbol{o}_{-i}) - \pi(D'_i|\boldsymbol{x})\big] - \gamma|\mathcal{Z}| \geq -3\gamma|\mathcal{Z}|.
\end{aligned}
$$

The first inequality uses the fact $\pi(z|\boldsymbol{o}_i, \boldsymbol{x}_{-i}) - \pi(z|\boldsymbol{x}) \geq 0$ when $\boldsymbol{o}_i(z) = 1$ and $\tilde{u}_i(z) \geq P \geq 2$ when $\boldsymbol{o}_i(z) = 1$ and $\boldsymbol{o}_{-i}(z) = 0$. The quantity in braces is nonpositive because for any profile $\boldsymbol{x}$, setting $\boldsymbol{x}_{-i} = \boldsymbol{o}$ only increases the probability that player i is the one to deviate from the path to z. The second inequality uses the nonpositivity of the quantity in the braces, and the fact that $\tilde{u}_i(z) = u_i(z) + \alpha \leq 2$.

Now we look at the remaining quantity, $\tilde{u}_i(\boldsymbol{o}) - \tilde{u}_i(\boldsymbol{x}_i, \boldsymbol{o}_{-i})$, which is simply the negative deviation of benefit of Player i's strategy $\boldsymbol{x}_i$ if all other players were direct. Indeed, since we know that $\boldsymbol{\xi}$ is an equilibrium, we have

$$\begin{aligned}
&\tilde{u}_i(\boldsymbol{o}) - \tilde{u}_i(\boldsymbol{x}_i, \boldsymbol{o}_{-i}) \\
&= \underbrace{[\tilde{u}_i(\boldsymbol{o}) - u_i(\boldsymbol{o})]}_{=\alpha} - \underbrace{[\tilde{u}_i(\boldsymbol{x}_i, \boldsymbol{o}_{-i}) - u_i(\boldsymbol{x}_i, \boldsymbol{o}_{-i})]}_{=\alpha(1-\Delta_i(\boldsymbol{x}_i,\boldsymbol{o}_{-i}))} + \underbrace{[u_i(\boldsymbol{o}) - u_i(\boldsymbol{x}_i, \boldsymbol{o}_{-i})]}_{\geq 0} \\
&\geq \alpha \Delta_i(\boldsymbol{x}_i, \boldsymbol{o}_{-i}) \geq \alpha \Delta_i(\boldsymbol{x}) - \gamma |\mathcal{Z}|,
\end{aligned}$$

where the final inequality again uses Lemma 7.20 and $\Delta_i(\boldsymbol{x}) := \sum_{z:\boldsymbol{o}_i(z)=0} \pi(z|\boldsymbol{x})$

Now, notice that $\delta_i \leq \Delta_i(\boldsymbol{x})$, by definition. Substituting into the previous inequality and Lemma 7.20, we have

$$\alpha \Delta_i(\boldsymbol{x}) - 4\left(n\epsilon + \frac{\sum_j \Delta_j(\boldsymbol{x})}{P}\right)|\mathcal{Z}| \leq P\epsilon,$$

or, rearranged,

$$\alpha \Delta_i(\boldsymbol{x}) - 4|\mathcal{Z}|\frac{\sum_j \Delta_j(\boldsymbol{x})}{P} \leq (P + 4n)\epsilon \leq 2P\epsilon$$

when $P \geq 4n$. Summing over all players i yields

$$\alpha \Delta - 4|\mathcal{Z}|\frac{\Delta}{P} \leq (P + 4n)\epsilon \leq 2P\epsilon$$

where $\Delta = \sum_i \Delta_i(\boldsymbol{x})$, or, rearranging,

$$\Delta \leq \frac{2P\epsilon}{\alpha - 4|\mathcal{Z}|/P}.$$

Both the payments from the mediator and the gap to optimal value are thus bounded by

$$\alpha + P\Delta \leq \alpha + \frac{2P^2\epsilon}{\alpha - 4|\mathcal{Z}|/P}.$$

Now taking $\alpha = 4|\mathcal{Z}|^{1/2}\epsilon^{1/4}$ and $P = 2|\mathcal{Z}|^{1/2}/\epsilon^{1/4}$ gives the desired bounds. □

7.6 Other Equilibrium Notions

So far, Theorems 7.13 and 7.19 handle only the case where the equilibrium is a *pure-strategy* Nash equilibrium of the game, given as part of the input. This section extends our analysis to other equilibrium notions and considers settings in which an *objective for the mediator* is given instead of a target equilibrium. For the former, we will show that many types of equilibrium can be viewed as pure-strategy equilibria in an *augmented game* in which the mediator has the ability to give *advice* to the players in the form of action recommendations. Then, in the original game, the goal is to guide the players to the pure strategy profile of following recommendations.

7.6.1 Necessity of Advice

We first show that without the possibility to give advice, steering is impossible with sublinear payments.

Theorem 7.21. *There exists a normal-form game, and objective function u_0 of the mediator, such that the unique optimal equilibrium is mixed, and it is impossible to steer players toward that equilibrium using only sublinear payments (and no advice).*

Proof. Consider a 2-player, binary action coordination game, with actions A and B. Players receive utility 1 point for playing the same action, and -1 otherwise. The mediator's goal is to *minimize* the welfare of the players.[7.4]

The welfare-minimizing equilibrium in this game is the fully-mixed one. So, we claim that, using sublinear payments alone, it is impossible to steer players to the mixed equilibrium. Consider the following algorithm for the players: Let $\Gamma^{(t)}$ be the game at time t induced by the mediator's payoff function $p^{(t)}$. Play an arbitrary Nash equilibrium of $\Gamma^{(t)}$, pure if possible. The total regret of the players after T rounds is at most 0 since the players always play a Nash equilibrium. There are three cases:

1. The players play $(\mathsf{A},\ \mathsf{A})$ or $(\mathsf{B},\ \mathsf{B})$. In this case, the players get social welfare 2.
2. The players play $(\mathsf{A},\ \mathsf{B})$ or $(\mathsf{B},\ \mathsf{A})$. In this case, the players get social welfare -2 in the game itself, but in order for either of these to be a Nash equilibrium, there must be a payment of at least 2 to each player.
3. The players play a mixed strategy. This means that $\Gamma^{(t)}$ had no pure strategy Nash equilibrium. Since $(\mathsf{A},\ \mathsf{A})$ is not an equilibrium, suppose WLOG that $v_1^{(t)}(\mathsf{B},\mathsf{A}) > v_1^{(t)}(\mathsf{A},\mathsf{A})$. Then $p_i^{(t)}(\mathsf{B},\mathsf{A}) > 2$. Since $(\mathsf{B},\ \mathsf{A})$ is also not a Nash equilibrium, we have $v_2^{(t)}(\mathsf{B},\mathsf{B}) > v_2^{(t)}(\mathsf{B},\mathsf{A})$. Since $(\mathsf{B},\ \mathsf{B})$ is also not a Nash equilibrium, we have $v_1^{(t)}(\mathsf{A},\mathsf{B}) > v_1^{(t)}(\mathsf{B},\mathsf{B})$, so $p_1^{(t)}(\mathsf{A},\mathsf{B}) > 2$. Thus, all four strategy profiles have either high welfare for the players, or nontrivial payments.

In all three cases, as a result, we must have $\sum_i u_i(\boldsymbol{x}^{(t)}) + 2p_i^{(t)}(\boldsymbol{x}^{(t)}) > 1$ for all timesteps t. Therefore, summing over $t = 1, \dots, T$, it is impossible for both quantities to grow sublinearly in T, which is what would be required for successful steering. □

Given this result, we will analyze a setting in with the mediator is allowed to provide "advice," and show a broad possibility result for steering.

[7.4] One could construct an example in which the mediator's goal is to *maximize* the players' utility, by simply adding a third player, with one action, whose utility is -10 if P1 and P2 play the same action and 0 otherwise.

7.6.2 More General Equilibrium Notions: Bayes-Correlated Equilibrium

Throughout this subsection, there will be two games: the original game $\hat{\Gamma}$, and the augmented game Γ. We will use hats to distinguish the various components of them. For example, a history of $\hat{\Gamma}$ is $\hat{h} \in \hat{\mathcal{H}}$, a strategy of Player i is $\hat{\boldsymbol{x}}_i \in \hat{\mathcal{X}}_i$, and so on. Given an n-player game $\hat{\Gamma}$, the *mediator-augmented game* Γ is the $n + 1$-player game constructed as follows. Γ is identical to $\hat{\Gamma}$, except that there is an extra player, namely, the mediator itself. We will denote the mediator as Player 0. For each (non-chance) player i, every decision point $\hat{h} \in \hat{\mathcal{H}}_i$ is replaced with the following gadget. First, the mediator selects an action $\hat{a} \in \hat{A}(\hat{h})$ to *recommend* to Player i. Player i privately observes the recommendation, and only then is allowed to choose an action. The mediator is assumed to have perfect information in the game. To ensure that the size of Γ is not too large, we make the following restriction: once two players have disobeyed action recommendations ("deviated"), the mediator ceases to give further action recommendations. Finally, upon reaching a terminal node $\hat{z} \in \hat{\mathcal{Z}}$, each player gets utility $\hat{u}_i(\hat{z})$.

We first analyze the size of Γ. A terminal node in Γ can be uniquely identified by a tuple $(\hat{z}, \hat{h}_1, \hat{h}_2, \hat{a}_1, \hat{a}_2)$ where $\hat{z}$ is the terminal node in the original game that was reached, $\hat{h}_1, \hat{h}_2$ are predecessors of $\hat{z}$ at which players deviated (or $\varnothing$ if the deviations did not happen), and $\hat{a}_1$ and $\hat{a}_2$ are the recommendations that the mediator gave at $\hat{h}_1, \hat{h}_2$ respectively (again, $\varnothing$ if the deviations did not happen). Thus, a (very loose) bound on the number of terminal nodes in Γ is $|\mathcal{Z}| \le |\hat{\mathcal{Z}}|^3$, *i.e.*, it is polynomial. (This is where we use the fact that only two deviations were allowed.)

As in the previous section, the mediator is able to *commit* to a strategy $\boldsymbol{\xi} \in \Xi$ upfront on each iteration. For a fixed mediator strategy $\boldsymbol{\xi}$, we will use $\Gamma^{\boldsymbol{\xi}}$ to refer to the n-player game resulting from treating the mediator as a nature player that plays according to $\boldsymbol{\xi}$.

The *direct strategy* $\boldsymbol{o}_i \in \mathcal{X}_i$ of each player i is the strategy that follows all mediator recommendations. The goal of the mediator is to find a *Bayes-correlated equilibrium*, which is defined as follows.

Definition 7.22. A *Bayes-correlated equilibrium* Γ is a strategy $\boldsymbol{\xi} \in \Xi$ for the mediator such that $\boldsymbol{o}$ is a Nash equilibrium of $\Gamma^{\boldsymbol{\xi}}$. An equilibrium $\boldsymbol{\xi}$ is *optimal* if, among all equilibria, it maximizes the mediator's objective $u_0(\boldsymbol{\xi}, \boldsymbol{o})$.

Bayes-correlated equilibria (BCEs) were introduced first by Bergemann and Morris [23] in single-step games. In sequential (extensive-form) games, BCEs were explored first, to our knowledge, by Makris and Renou [206] in the economics literature, and in independent work in the computer science literature as a special case of the general framework introduced in Chapter 6. Bayes-correlated equilibria are easily seen to be a superset of most other equilibrium notions, including (mixed) Nash equilibria, *extensive-form correlated equilibria* (EFCE) [291], *communication equilibria* [109, 228], and many more. The *revelation principle* assures us that the assumption that players will be direct in equilibrium is without loss of generality: for every possible Nash equilibrium $\boldsymbol{x}$ of $\Gamma^{\boldsymbol{\xi}}$, then there is some $\boldsymbol{\xi}'$ such that $u_i(\boldsymbol{\xi}', \boldsymbol{o}) = u_i(\boldsymbol{\xi}, \boldsymbol{x})$.

BCEs naturally capture the problems of *information design* and *Bayesian persuasion* (*e.g.*, Kamenica and Gentzkow [167]). In particular, the results in this section can therefore be thought of as a version of information design/Bayesian persuasion that does not need to assume that players will play a certain profile ($\boldsymbol{o}$), but instead *steers* the players to play that profile.

Since Γ^{ξ} is just an n-player game with pure Nash equilibrium $\boldsymbol{o}$, all of the results in the previous sections apply. Therefore, it follows immediately that is possible to steer players toward *any* BCE (and thus any mixed Nash equilibrium, any EFCE, or any communication equilibrium) so long as the mediator is allowed to give advice to the players. We therefore have the following result.

Theorem 7.23. *Algorithms* FullFeedbackSteer *and* TRAJECTORYSTEER *can be used to steer players to an arbitrary Bayes-correlated equilibrium, with (up to a polynomial loss in the dependence on* $|\hat{\mathcal{Z}}|$*, because* $|\mathcal{Z}| = \mathsf{poly}(|\hat{\mathcal{Z}}|)$*) the same bounds.*

7.7 Steering Without Prior Knowledge of Utilities

In this section, we study the problem of steering *without prior knowledge of utilities*, in the special case of normal-form games. That is, we assume that the mediator does not initially know the players' utility functions $u_i : \mathcal{Z} \to \mathbb{R}$, and yet we still wish to steer the players to an equilibrium.

Since the mediator does not know the utility functions *a priori*, it does not make sense for the mediator to know the desired equilibrium either. Therefore, one initial question to ask is what solution concept we *ought* to steer the players to, given freedom over this choice. We define a solution concept called *correlated equilibrium with payments* (CEP), in which the mediator has a utility function, and wishes to optimize its utility minus the amount of payment that it must give. It is therefore possible for the amount of payment to grow linearly, so long as the corresponding increase in mediator utility is large enough to justify the payments. We then show that the mediator-optimal CEP exactly characterizes the value (averaged across timesteps) that the mediator can achieve in the limit $T \to \infty$:

Theorem 7.24 (Informal summary of results for steering without utilities in normal-form games)**.** *Let F^* be the objective value for the mediator in the mediator-optimal CEP. Then:*

- *no mediator—even if the mediator knows the game Γ exactly on the first round—can achieve time-averaged value better than $F^* + \mathsf{poly}(|\mathcal{Z}|) \cdot T^{-1/2}$, and*
- *there exists a mediator that can achieve time-averaged value at least $F^* - \mathsf{poly}(|\mathcal{Z}|) \cdot T^{-1/4}$, with no prior knowledge of the players' utilities.*

All our algorithms are implementable by the mediator with $\mathsf{poly}(|\mathcal{Z}|)$ runtime complexity.

As per Section 7.6.2, we model the signaling scheme by turning the normal-form game into an extensive-form game in which each player first observes their own signal. Since the origial game is normal form, the new game is a Bayesian game, so Theorem 7.16 applies. Therefore, informally speaking, to show Theorem 7.24, it suffices for the mediator to figure out what the optimal CEP is, and then use Theorem 7.16 to steer the players toward it. However, the mediator must do this without prior knowledge of the players' utility functions! We therefore break the problem down into two steps. First, we learn the players' utilities, and second, we run the steering algorithm.

7.7.1 Learning Utilities

In this subsection, we formulate and study the subproblem of *learning utilities* from no-regret learning algorithms. Specifically, we study the following problem. There is an underlying normal-form game in which each player $i = 1, \dots, n$ has action set $\mathcal{A}_i$ of size m_i. Let $M = \prod_i m_i$. On each round $t = 1, \dots, T$, the following events happen, in order:

1. The mediator selects *payment function* $p_i^t : \mathcal{A}_i \to \mathbb{R}_{\geq 0}$ for each player i. The payment $p_i^t(a_i)$ is added to player i's reward, creating a new game Γ^t with utility functions given by $u_i^t(a) := u_i(a) + p_i^t(a_i)$. The mediator may also send *signals* $s_i^t \in S_i$ to each player i, where S_i is a finite signal set.
2. Each player i simultaneously selects an action $a_i^t \in \mathcal{A}_i$.
3. The mediator observes the joint strategy a^t. Each player i gets reward $u_i^t(a^t)$.

For simplicity, in this section, since players are playing Bayesian games, we will assume that their external regret *in every information set* is at most $R(T)$. That is, we assume that they run independent regret minimizers at every information set, so that

$$R_i(t, s_i) := \max_{a_i \in \mathcal{A}_i} \sum_{\substack{\tau \leq t \\ s_i^\tau = s_i}} \left[u_i^\tau(a_i, \mathcal{A}_{-i}^\tau) - u_i^\tau(a^\tau) \right] \leq R(T).$$

for every player i, time $t \leq T$, and signal s_i. Although not fully general, this assumption encompasses basically every known technique for performing regret minimization in Bayesian games, including all algorithms based on CFR.

7.7.1.1 Game Equivalence and Formal Goal Statement

Our goal is to design algorithms for the mediator to learn the players' utility functions u_i by designing the payment functions p_i^t and sending signals s_i^t for all players. This goal as currently stated is impossible. To see this, note that players' actions in all the behavioral models are only affected by their utility *differences*, that is, the differences $u_i(a_i, \mathcal{A}_{-i}) - u_i(a_i', \mathcal{A}_{-i})$. In other words, if we create another game Γ' with $u_i'(a_i, \mathcal{A}_{-i}) = u_i(a_i, \mathcal{A}_{-i}) + w_i(a_{-i})$ for all $a \in A$, where $w_i : \mathcal{A}_{-i} \to \mathbb{R}$ is an arbitrary function not depending on i's action, there is no way to distinguish Γ from Γ' using only behavioral data. Thus, we can only determine utility functions *up to* additive w_i terms. We can thus formally state our goal as follows.

Goal. Given a game Γ and precision ϵ, we say that the mediator *ϵ-learns* the game Γ if it outputs utility functions $\tilde{u}_i : A \to \mathbb{R}$ such that there exist functions $w_i : \mathcal{A}_{-i} \to \mathbb{R}$ satisfying

$$|u_i(a) + w_i(a_{-i}) - \tilde{u}_i(a)| \leq \epsilon$$

for all players i and action profiles $a \in A$. The goal of the mediator is to ϵ-learn Γ in as few rounds as possible. Fortunately, learning the utilities in this sense is enough to steer players to an optimal equilibrium, as we will see later.

Algorithm 7.4 (LearnUtility1P): Mediator's algorithm for learning a single-player game in the no-regret model

```
p^1 ← 1
for each time t = 1, ..., T do
    mediator selects payment vector $p^t$, observes action $a^t$ played by the player
    mediator sets $p^{t+1} \leftarrow \Pi_{\mathcal{P}}\left[p^t - \eta e_{a^t}\right]$        ▷ $\eta = \sqrt{m/T}$ is the step size
return $-\frac{1}{T}\sum_{t=1}^{T} p^t$
```

7.7.1.2 The Single-Player Case

In this subsection, it will be convenient to view the single player's utility function as a vector $\boldsymbol{u} \in [0,1]^m$, and similarly the payment $p^t : [m] \to \mathbb{R}$ as vector $\boldsymbol{p}^t \in \mathbb{R}^m$ and total utility $\boldsymbol{u}^t := \boldsymbol{u} + \boldsymbol{p}^t$. To simplify notations, we subtract the average utility of all actions from the utility of each action: $\boldsymbol{u} \leftarrow \boldsymbol{u} - \langle \mathbf{1}, \boldsymbol{u}\rangle \mathbf{1}/m$, so that $\boldsymbol{u} \in [-1,1]^m$ and $\langle \mathbf{1}, \boldsymbol{u}\rangle = 0$. By the discussion in Section 7.7.1.1, this does not change the mediator's learning problem.

For single-player games, we will also not use signaling, *i.e.*, it will be enough to set $|S_i| = 1$.

The key idea in our algorithm is to imagine the mediator and player as playing a zero-sum game where the mediator selects the payment function $\boldsymbol{p}$ from some set $\mathcal{P}$ to be specified later, the player selects $\boldsymbol{x} \in \Delta(m)$, the player's utility is given by $\langle \boldsymbol{u} + \boldsymbol{p}, \boldsymbol{x}\rangle$, and the mediator's utility is $-\langle \boldsymbol{u} + \boldsymbol{p}, \boldsymbol{x}\rangle$. Call this game Γ_0. In particular, if we set $\mathcal{P} = \{\boldsymbol{p} \in [0,2]^m : \langle \mathbf{1}, \boldsymbol{p}\rangle = m\}$, we have the following:

Lemma 7.25. *In the zero-sum game Γ_0, every ϵ-Nash equilibrium strategy for the mediator has the form $\boldsymbol{p} = \mathbf{1} - \boldsymbol{u} + \boldsymbol{z}$, where $\|\boldsymbol{z}\|_1 \le 4m\epsilon$.*

> *Proof.* Setting $\boldsymbol{p} = \mathbf{1} - \boldsymbol{u}$ guarantees $\langle \boldsymbol{u} + \boldsymbol{p}, \boldsymbol{x}\rangle = \langle \mathbf{1}, \boldsymbol{x}\rangle = 1$ for every $\boldsymbol{x} \in \Delta(m)$. Thus, in every ϵ-Nash equilibrium, the player's utility is at most $1+\epsilon$. Now suppose for contradiction that $(\boldsymbol{p}, \boldsymbol{x})$ is an ϵ-Nash equilibrium with $\|\boldsymbol{p} + \boldsymbol{u} - \mathbf{1}\|_1 > 4m\epsilon$. Then since $\langle \boldsymbol{p} + \boldsymbol{u} - \mathbf{1}, \mathbf{1}\rangle = 0$ by construction, there must be an action a for which $(\boldsymbol{p}+\boldsymbol{u}-\mathbf{1})(a) > 2\epsilon$, *i.e.*, $(\boldsymbol{u}+\boldsymbol{p})(a) > 1+2\epsilon$. But then the player has an ϵ-profitable deviation to action a. □

It is well known that no-regret learning algorithms converge on average to Nash equilibria in zero-sum games. In particular, if both mediator and player run no-regret algorithms, and R_0 is the regret after T timesteps for the mediator, then the average mediator strategy $\frac{1}{T}\sum_{t=1}^{T} \boldsymbol{p}^t$ is an ϵ-Nash equilibrium for $\epsilon \lesssim (R_0 + C\sqrt{T})/T$. Here, we will use the projected gradient descent algorithm for the mediator. Note that, although the mediator's utility function $\boldsymbol{p} \mapsto \langle \boldsymbol{u} + \boldsymbol{p}, \boldsymbol{x}\rangle$ depends on $\boldsymbol{u}$ (which the mediator does not know), the gradient of the mediator's utility function is $-\boldsymbol{x}$, which does not depend on $\boldsymbol{u}$ and can be unbiasedly estimated by $-\boldsymbol{e}_a$ where a is an action sampled according to $\boldsymbol{x}$ and $\boldsymbol{e}_a$ is the unit vector whose a-th component is 1. Thus, the mediator can run projected gradient descent without the knowledge of $\boldsymbol{u}$. The resulting algorithm is formalized in Algorithm 7.4.

Algorithm 7.5 (LearnUtilityMP): mediator's algorithm for learning a multi-player game in the no-regret model

```
$t \leftarrow 1$
for each player $i = 1, \dots, n$ do
    for each action profile $a_{-i} \in A_{-i}$ do
        $\boldsymbol{p}^1 \leftarrow \mathbf{1} \in \mathbb{R}^{A_i}$
        for timestep $\ell = 1, \dots, L$ do
            mediator sets $p_i^t(\cdot) = \boldsymbol{p}^\ell(\cdot)$ and $p_j^t(a_j') = 2\mathbb{1}\{*\}a_j' = a_j$ for every $j \neq i$
            mediator sends signals $s_i^t = \perp$ and $s_j^t = a_j$ for every $j \neq i$
            mediator observes action profile $a^t$ played by players
            mediator sets $\boldsymbol{p}^{\ell+1} \leftarrow \Pi_{\mathcal{P}}\left[\boldsymbol{p}^\ell - \eta \boldsymbol{e}_{a_i^t}\right]$        ▹ $\eta = \sqrt{m_i/L}$ is the step size
            $t \leftarrow t + 1$
        $\tilde{u}_i(\cdot, a_{-i}) = -\frac{1}{L}\sum_{\ell=1}^L \boldsymbol{p}^\ell$
return $\tilde{u}$
```

Theorem 7.26. LearnUtility1P *ϵ-learns any single-player game Γ using $O(m^3 + C^2m^2)/\epsilon^2$ rounds.*

Proof. Let $\bar{\boldsymbol{p}} = \frac{1}{T}\sum_{t=1}^T \boldsymbol{p}^t$ be the average payment. From the preliminaries, the regret bound of the mediator is given by $R_0 \le BG\sqrt{T}$ where $B \lesssim \sqrt{m}$ and $G = 1$. Then, by Lemma 7.25 and the argument in the previous paragraph, we have

$$\epsilon = \|\bar{\boldsymbol{p}} + \boldsymbol{u} - \mathbf{1}\|_\infty \le \|\bar{\boldsymbol{p}} + \boldsymbol{u} - \mathbf{1}\|_1 \lesssim \frac{m}{T}(R_0 + C\sqrt{T}) \lesssim \frac{m}{\sqrt{T}}(C + \sqrt{m})$$

upon which solving for T yields the desired result. □

7.7.1.3 The Multi-Player Case

The no-regret learning case for multiple players is the only case in which we take advantage of signaling. Intuitively, our algorithm uses signals to induce the action profile a_{-i} among other players without increasing their regret by too much. More precisely, we set the signal sets as $S_i := A_i \sqcup \{\perp\}$ where $\perp$ is a special signal indicating that i's utility is the one being learned at the moment. Then, when learning the utility $u_i(\cdot, a_{-i})$, we send signal $\perp$ to player i and the desired action profile a_j for each player $j \neq i$. This idea is formalized in Algorithm 7.5.

Theorem 7.27. *For some appropriate choice of the hyperparameter L,* LearnUtilityMP *ϵ-learns any game in* $\mathsf{poly}(M)/\epsilon^2$ *rounds.*

Proof. As in Section 7.7.1.2, we will assume without loss of generality that $\sum_{a_i \in A_i} u_i(a_i, a_{-i}) = 0$ for all players i and opponent profiles a_{-i}.

We claim first that, for any player i and any action $a_i \in A_i$, the number of times that i does *not* play $a_i^t = a_i$ when given signal $s_i^t = a_i$ is bounded by $C\sqrt{T}$. To see this, note that whenever the mediator sends signal a_i, the payment is always set such that $u_i^t(a_i, a_{-i}) \ge 1 + u_i^t(a_i', a_{-i})$. Thus, the number of times i does not play $a_i^t = a_i$ quantity lower-bounds the regret $R(T, a_i)$. The claim follows from the regret guarantee $R(T, a_i) \le C\sqrt{T}$.

We will refer to the iterations of the inner loop over action profiles a_{-i} as *phases*. Fix a player i, and number the phases for that player using integers $k = 1, \dots, M_i := \prod_{j \ne i} m_j$, corresponding respectively to strategy profiles $\bar{a}_{-i}^1, \dots, \bar{a}_{-i}^{M_i} \in A_{-i}$. Let $\mathcal{T}_i(k) = \{\underline{T}_i(k), \dots, \bar{T}_i(k)\}$ be the set of timesteps in player i's kth phase. Let B_k be the total numer of rounds in phases $1, \dots, k$ in which $a_{-i}^t \ne a_{-i}$. By the mediator's regret bound in each phase, we have

$$\sum_{t \in \mathcal{T}_i(k)} u_i^t(a_i^t, \bar{a}_{-i}^k) = \sum_{t \in \mathcal{T}_i(k)} \left[u_i(a_i^t, \bar{a}_{-i}^k) + p_i^t(a_i^t) \right] \le L + R_0$$

or else the mediator has a profitable deviation to $p_i^t(\cdot) = 1 - u_i(\cdot, \bar{a}_{-i}^k)$.

Fix some $K \le M_i$ and $a_i \in A_i$. By the anytime regret bound of player i under signal $\perp$, we have

$$\begin{aligned} \sum_{\substack{k \le K \\ t \in \mathcal{T}_i(k)}} u_i^t(a_i, a_{-i}^t) &\le \sum_{\substack{k \le K \\ t \in \mathcal{T}_i(k)}} u_i^t(a_i, \bar{a}_{-i}^k) + 2\mathbb{1}\left\{ a_{-i}^t \ne \bar{a}_{-i}^k \right\} \\ &\le LK + R_0 K + R_i(T_i(k), \perp) + 2B_K \end{aligned}$$

$$\underbrace{\sum_{k=1}^{K} \frac{1}{L} \sum_{t \in \mathcal{T}_i(k)} \left[u_i(a_i, a_{-i}^t) + p_i^t(a_i) - 1 \right]}_{:= \epsilon_i(k, a_i)} \le \frac{1}{L}\left(R_0 K + 3nC\sqrt{T} \right).$$

The error we need to bound is $\|\epsilon_i(k, \cdot)\|_\infty$. Since this holds for any a_i, and $\sum_{a_i} \epsilon_i(k, \cdot) = 0$ by definition, it follows that

$$\left\| \sum_{k=1}^{K} \epsilon_i(k, a_i) \right\|_\infty \le \frac{m}{L}\left(R_0 K + 3nC\sqrt{T} \right).$$

By an inductive application of the triangle inequality, we have

$$\|\epsilon_i(k, \cdot)\|_\infty \le \frac{km}{L}\left(R_0 M + 3nC\sqrt{T} \right).$$

Finally, substituting $R_0 \lesssim \sqrt{mL}$ and $T \le nML$, we arrive at

$$\|\epsilon_i(k, \cdot)\|_\infty \lesssim \frac{Mm}{\sqrt{L}}\left(\sqrt{m}M + nC\sqrt{nM} \right)$$

upon which taking $L \le \mathsf{poly}(M)/\epsilon^2$ is enough to complete the proof. □

Signals are vital to this analysis. Without them, it would be possible for players to incur large *negative* regret, which harms the learning process because it allows the players to "delay" the learning until their regrets once again become non-negative. For example, if we were to execute our algorithm without signals, then by the time $\underline{T}_n(0)$ at which the outer loop reaches player n, player n could have $\Omega(\underline{T}_n(0))$ regret for *every* action, making it impossible to say anything about how player n will act for the next $\Omega(\underline{T}_n(0))$ rounds. Using signals allows us to separate out the regret of player n in previous rounds from the regret of player n when its own utility function is being learned.

7.7.1.4 What Outcome Should We Steer To?

The steering problem stipulates that the mediator knows in advance, or be able to compute, the desired outcome that we wish to induce. Of course, in the setting where utilities are unknown, such a stipulation is unreasonable: the mediator does not initially know the players' utilities in the game Γ, so it cannot know what outcome it wishes to induce. We thus take a more direct approach: we try to maximize the average reward, less payments, of the mediator. That is, we will assume that the mediator has a utility function $u_0 : \mathcal{A} \to \mathbb{R}$, and we will attempt to optimize the mediator's objective, defined as the mediator utility minus payments:

$$F(T) := \frac{1}{T} \sum_{t=1}^{T} \left[u_0(a^t) - \sum_{i=1}^{n} p_i^t(s^t, a^t) \right].$$

We now introduce a solution concept which we call the *correlated equilibrium with payments* (CEP). Intuitively, a CEP is a distribution of *signals* and *payment functions*, which may be correlated both with each other and across the players, that satisfies the usual incentive compatibility constraints. Formally, we have the following definition.

Definition 7.28. A *correlated profile with payments* is a pair $(\mu, p) \in \Delta(\mathcal{A}) \times \mathbb{R}_+^{[n]\times\mathcal{A}\times\mathcal{A}}$. The vector P is to be interpreted as a collection of n payment functions $p_i : A \times \mathcal{A} \to \mathbb{R}_+$, where $p_i(s, a)$ is the payment to player i given joint signal s and joint action a. Given such a distribution μ, the *objective value* for the mediator is defined as

$$F(\mu) := \mathop{\mathbb{E}}_{a\sim\mu} [u_0(a) - p(a)],$$

where for notational simplicity we set $p(a) := \sum_i p_i(a, a)$. An *ϵ-correlated equilibrium with payments* (CEP) is a pair (μ, p) satisfying the incentive compatibility (IC) constraints

$$\mathop{\mathbb{E}}_{a\sim\mu} \left[u_i^p(a, \phi_i(a_i), a_{-i}) - u_i^p(a, a) \right] \le \epsilon,$$

where $u_i^p(s, a) := u_i(a) + p_i(s, a)$, for every player i and deviation function $\phi_i : \mathcal{A}_i \to \mathcal{A}_i$. An ϵ-CEP is *optimal* if it maximizes the mediator objective $F(\mu)$ over all ϵ-CEPs.

Note that it is without loss of generality to assume that the mediator never gives payments to non-equilibrium actions, *i.e.*, we have $p(s, a) = 0$ for $s \neq a$. Giving such payments would only decreases the mediator objective and worsens incentive compatibility.

7.7.2 Properties of correlated equilibria with payments

Before proceeding with our analysis of steering learners toward CEPs, we state some simple results about CEPs in general. First, optimal (ϵ-)CEPs can be efficiently computed when the game is known.

Proposition 7.29. *There is a* $\mathsf{poly}(M)$*-time algorithm for computing an optimal* ϵ*-CEP, given a game* Γ *with rational utility values and rational parameter* $\epsilon \ge 0$.

Proof. Use the change of variables

$$q_i(a_i) := \mu(a_i) \cdot \mathop{\mathbb{E}}_{a \sim \mu | a_i} p_i(a, a).$$

That is, $q_i(a_i)$ is the μ-weighted total payment given to player i across all strategy profiles on which player i is recommended action a_i. Then the following LP precisely represents the problem of computing an optimal ϵ-CEP.

$$\begin{aligned}
\max \quad & \sum_{a \in \mathcal{A}} \mu(a) u_0(a) - \sum_{\substack{i \in [n] \\ a_i \in \mathcal{A}_i}} q_i(a_i, a) \quad \text{s.t.} \\
\sum_{a_{-i} \in \mathcal{A}_{-i}} \mu(a) \big[u_i(a_i', a_{-i}) - u_i(a) \big] - q_i(a_i) &\le \epsilon_i(a_i) && \forall i \in [n], a_i, a_i' \in \mathcal{A}_i \\
\sum_{a_i \in \mathcal{A}_i} \epsilon_i(a_i) &\le \epsilon && \forall i \in [n] \\
\sum_{a \in \mathcal{A}} \mu(a) &= 1 \\
0 \le q_i(a_i) &\le \mu(a_i) && \forall i \in [n], a_i, a_i' \in \mathcal{A}_i
\end{aligned} \tag{7.5}$$

This LP has $\mathsf{poly}(M)$ variables and constraints, so the proof is complete. □

The above result also implies that it is WLOG to restrict the payment functions to output range $[0, 1]$, because this is sufficient to satisfy all incentive constraints.

Second, the fact that the payments can be signal-dependent is not innocuous, except when the payment at equilibrium is zero.

Proposition 7.30 (Correlation does not help when no payments are allowed in equilibrium)**.** *The* 0*-CEPs with* $\mathbb{E}_{a \sim \mu}\, p(a, a) = 0$ *are exactly the correlated equilibria.*

Proof. In the LP (7.5), this is equivalent to setting $q_i(\cdot) = 0$ for every player i, in which case (7.5) is just the LP characterizing correlated equilibria. □

However, when the payment at equilibrium is positive, it is possible for signal-dependent payments to help the mediator.

Proposition 7.31 (Signal-dependent payments can help in general)**.** *There exists a game Γ, and pricipal utility function u_0, such that the optimal value of (7.5) is greater than the objective value of the optimal CEP in which $p(s, a)$ depends only on a.*

Proof sketch. In the normal-form game below, P1 and P2 play matching pennies, and the mediator is willing to pay a large amount to avoid a particular pure profile.

	X	Y
X	$-\infty, 0, 1$	$0, 1, 0$
Y	$0, 1, 0$	$0, 0, 1$

P1 chooses the row, P2 chooses the column. In each cell, the mediator's utility is listed first, then P1's, then P2's. Now consider the following CEP: The mediator mixes evenly between recommending (X, Y), (Y, X), and (Y, Y). If the mediator recommends (Y, X), it also promises a payment of 1 to P2 if P2 follows the recommendation X. This CEP has mediator objective value $-1/3$, and no signal-independent CEP can match that value. □

The proof is formalized in the full paper [321].

In the language of Monderer and Tennenholtz [218], a CEP with $k = \mathbb{E}_{a\sim\mu}\, p(a)$ is called a *k-implementable correlated equilibrium.*[7.5] They show that all correlated equilibria are 0-implementable, but do not show the converse. Our results improve upon theirs by 1) showing the converse (Proposition 7.30), and 2) analyzing the $k > 0$ case, in particular, by incorporating a mediator objective and showing how to compute the optimal CEP.

Finally, in Definition 7.28, we set the signal set to be identical to the action set. In the appendix of the full paper [321], we show that this is without loss of generality, that is, a sort of *revelation principle* holds for CEPs.

7.7.3 CEPs and optimal steering

We now show that the objective value of the optimal (0-)CEP is exactly the maximum value attainable (in the limit $T \to \infty$) by a mediator in our model. We start with the upper bound. Intuitively, the upper bound holds because, for any algorithm for the mediator, the players can always compute and play a Nash equilibrium of the payment- and signal-augmented game Γ^t, which leads to zero regret in expectation and value bounded above by the optimal CEP value. We now formalize this argument.

[7.5] Instead of our condition of *ex-interim* IC, Monderer and Tennenholtz [218] insist on *dominant-strategy* IC, that is, they insist that $u_i^p(s, s_i, a_{-i}) \geq u_i^p(s, a)$ for *every* s and a. However, this requirement does not change anything in equilibrium, because one can always set $p(s, s_i, a_{-i})$ when $s_i = a_i$ and $s \neq a$ to be so large that playing a_i becomes dominant. Indeed, Monderer and Tennenholtz [218] do this to establish their results on implementation, and this was the main idea in our earlier steering algorithms for normal-form games.

Algorithm 7.6 (LearnUtilitiesThenSteer): Principal's algorithm for steering without prior knowledge of utilities

```
using Algorithm 7.5, estimate the utility functions to precision $\epsilon$
using the LP (7.5), compute an optimal $2\epsilon$-CEP $(\mu^*, p^*)$ of the estimated game $\tilde{\Gamma}$
for remaining rounds do
    set $\mu^t = \mu^*$ and $p_i^t(s, a) = \begin{cases} p_i^*(a, a) + 2\epsilon + \rho & \text{if} \quad s = a \\ 2 & \text{if} \quad s \neq a, a_i = s_i \\ 0 & \text{otherwise} \end{cases}$.
```

Theorem 7.32. *Let Γ be any game, and suppose the signal sets have size $|S_i| \le \text{poly}(m)$. Then for any possible algorithm for the mediator, there exists some algorithm that the players can use, for which, with probability $1 - \delta$,*

1. *no player ever plays an action that is not rationalizable,*
2. *each player's regret $R_i(t, s_i)$ is bounded by $C\sqrt{T}$ for every $t \le T$ and signal s_i, and*
3. *the mediator objective value $F(T)$ is bounded above by the objective value F^* of the optimal 0-CEP.*

Proof. The algorithm for the mediator, on each round, selects payments function P_i^t and signal distribution $\mu^t \in \Delta(S)$. Together, these induce a Bayesian game Γ^t, where the players' strategies correspond to functions $\pi_i^t : S_i \to \Delta(A_i)$. Suppose that the players play according to a Nash equilibrium of Γ^t. Then the players incur no regret, and the mediator's utility is bounded above by F^*. □

We now show that the mediator *can* achieve utility F^* in the limit $T \to \infty$. Intuitively, the algorithm will work in two stages. In the first stage, the mediator uses Algorithm 7.5 to learn the utility functions of the players. Then, the mediator computes an optimal CEP and steers the players to it. The steering algorithm is adapted from Theorem 7.16, and presented in full here for the sake of self-containment. Perhaps most notably, since the mediator only knows the game up to an error $\epsilon > 0$, it must give extra payments of at least ϵ to ensure that players do not deviate.

Theorem 7.33. *For appropriate choices of the hyperparameters L (from Algorithm 7.5) and ρ, LearnUtilitiesThenSteer guarantees mediator objective $F(T) \ge F^* - \text{poly}(M)/T^{1/4}$ rounds.*

Proof. From the analysis of Theorem 7.27, Algorithm 7.5 learns a game to precision ϵ, where $\epsilon = \text{poly}(M)\sqrt{T}/L$. (Notice that we cannot assume $T = \text{poly}(M) \cdot L$, because T is the total number of rounds across *both* stages of the algorithm.)

Since $\tilde{U}$ and U differ by only ϵ (up to player-independent terms), every CEP of Γ is a 2ϵ-CEP of $\tilde{\Gamma}$. The payment function p_i^t for the steering stage then ensures that, when given signal s_i, it is *dominant* for player i to play a_i. Formally, regardless of how other players act, we have

$$u_i^t(s, s_i, a_{-i}) - u_i^t(s, a) \geq \rho$$

for every $s \in S$ and $a \in A$ with $a_i \neq s_i$. Further, from the analysis of Theorem 7.27, player i's regret against following signals $s_i \neq \perp$ is always nonnegative. Therefore, by player i's regret bound, there are at most $C\sqrt{T}/\rho$ rounds on which player i fails to obey recommendation s_i in the steering stage. By a union bound, there are therefore $mnC\sqrt{T}/\rho$ rounds in the steering stage on which $a^t \neq s^t$. Thus, the mediator's suboptimality is bounded by

$$F^* - F(T) \leq \underbrace{\frac{(2n+1)nmL}{T}}_{(1)} + \underbrace{n(2\epsilon + \rho)}_{(2)} + \underbrace{\frac{(2n+1)mnC}{\rho\sqrt{T}}}_{(3)} \leq \mathsf{poly}(M) \cdot \left(\frac{L}{T} + \frac{\sqrt{T}}{L} + \rho + \frac{1}{\rho\sqrt{T}}\right)$$

where the three terms are:

1. the suboptimality and payments in the utility learning stage,
2. the bonus payments to ensure strict incentive compatibility in the steering stage, and
3. the suboptimality and payments in rounds on which $a^t \neq s^t$.

Setting $\rho = T^{-1/4}$ and $L = T^{3/4}$ then completes the proof. □

7.8 Experimental Results

We ran experiments with our TrajectorySteer algorithm (Definition 7.18) on various notions of equilibrium in extensive-form games. Since the hyperparameter settings suggested by Definition 7.18 are very extreme, in practice we fix a constant P and set α dynamically based on the currently-observed gap to directness. We used CFR+ [282] as the regret minimizer for each player, and precomputed a welfare-optimal equilibrium with the LP algorithm from Chapter 6. In most instances tested, a small constant P (say, $P \leq 8$) is enough to steer CFR+ regret minimizers to the exact equilibrium in a finite number of iterations. Two plots exhibiting this behavior are shown in Figure 7.7. More experiments, as well as descriptions of the game instances tested, can be found in the appendix of the full paper [318].

7.9 Conclusions and Future Research

We established that it is possible to steer no-regret learners to optimal equilibria using vanishing rewards, even under trajectory feedback. There are many interesting avenues for future research. First, this chapter did not attempt to provide optimal rates, and their improvement is a fruitful direction for future work. Second, are there algorithms with less demanding knowledge assumptions for the principal, *e.g.*, steering without full knowledge of the players' information? Similarly, can the results of Section 7.7 be applied to extensive-form games as well? Finally, our main

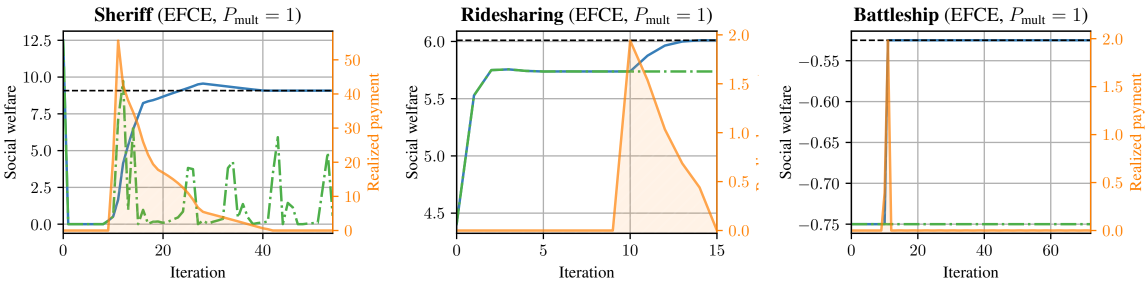


Figure 7.7: *Sample experimental results. The blue line in each figure is the social welfare (left y-axis) of the players* with *steering enabled. The green dashed line is the social welfare* without *steering. The yellow line gives the payment (right y-axis) paid to each player. The flat black line denotes the welfare of the optimal equilibrium. The panels show the game, the equilibrium concept (in this figure, always EFCE). In all cases, the first ten iterations are a "burn-in" period during which no payments are issued; steering only begins after that.*

behavioral assumption throughout this chapter is that the regret players incur vanishes in the limit. Yet, stronger guarantees could be possible when specific no-regret learning dynamics are in place, such as mean-based learning [34]; see [127, 128, 289] for recent results in the presence of *strict* equilibria. Concretely, it would be interesting to understand the class of learning dynamics under which the steering problem can be solved with a finite cumulative budget.

Chapter 8

Efficient Φ-Regret Algorithms for General Domains

8.1 Introduction

The long-standing absence of efficient algorithms for computing an NFCE shifted the focus to natural relaxations thereof, which can be understood through the notion of *Φ-regret* [135, 251, 278]. In particular, Φ represents a set of strategy deviations; the richer the set of deviations, the stronger the induced solution concept. When Φ contains all possible transformations, one recovers the notion of NFCE—corresponding to *swap regret*, while at the other end of the spectrum, *coarse correlated equilibria* correspond to Φ consisting solely of constant transformations (aka. *external regret*). Perhaps the most notable relaxation is the *extensive-form correlated equilibrium (EFCE)* [291], which can be computed exactly in time polynomial in the representation of the game tree [153]. Considerable interest in the literature has recently been on *learning dynamics* minimizing Φ-regret (*e.g.*, [17, 25, 73, 89, 95, 117, 132, 207, 223, 224, 233]). A key reference point in this line of work is the recent construction of Farina and Pipis [95], an efficient algorithm minimizing *linear swap regret*—that is, the notion of Φ-regret where Φ contains all *linear* deviations. Such algorithms lead to an ϵ-equilibrium in time polynomial in the game's description and $1/\epsilon$—aka. a fully polynomial-time approximation scheme (FPTAS).

Yet, virtually nothing was known beyond those special cases until recent breakthrough results by Dagan et al. [71] and Peng and Rubinstein [242], who introduced a new approach for reducing swap to external regret; unlike earlier reductions [29, 132, 277], their algorithm can be implemented efficiently even in certain settings with an exponential number of pure strategies. For extensive-form games, their reduction implies a polynomial-time approximation scheme (PTAS) for computing an ϵ-correlated equilibrium; their algorithm has complexity $N^{\tilde{O}(1/\epsilon)}$ for games of size N, which is polynomial only when ϵ is an absolute constant. Instead, we focus here on algorithms with better complexity $\mathsf{poly}(N, 1/\epsilon)$, the typical guarantee one hopes for within the no-regret framework.

While this result does not rule out the existence of efficient algorithms beyond the adversarial

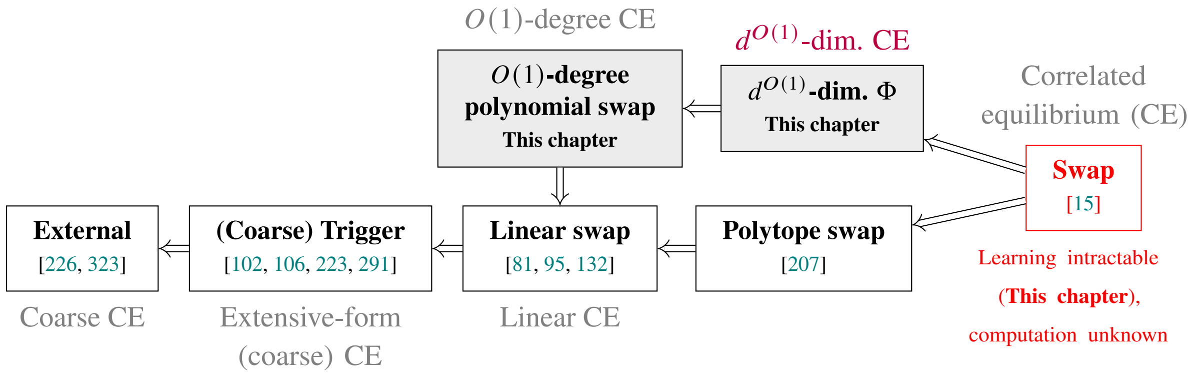


Figure 8.1: *The arrows $A \implies B$ denote that minimizing the notion of regret A implies minimizing the notion of regret B. In other words, A defines a superset of deviations that the learner considers compared to B, and hence leads to a stronger notion of equilibrium. The gray text below or above a notion of regret denotes the name of the corresponding notion of equilibrium, if applicable.*

regime, it does immediately bring to the fore a well-studied but pressing question: *what notions of hindsight rationality are efficiently learnable?*

Hindsight rationality in online learning can be understood through a set of functions, Φ, so that no deviation according to a function in Φ can retrospectively improve the cumulative utility; such a learner is said to be consistent with minimizing Φ-*regret* [132, 136, 278]. The broader the set of deviations Φ, the more appealing the ensuing concept of hindsight rationality. The usual notion of external regret is an instantiation of that framework for which Φ contains solely constant functions—referred to as *coarse* deviations. On the other end of the spectrum, when Φ contains all possible deviations, one finds the powerful notion of swap regret—associated with (normal-form) correlated equilibria. The fundamental question thus is to characterize the structure of Φ that enables efficient learnability—and, indeed, computation.

Much of the recent research in the context of learning in extensive-form games has focused on this exact problem. This can be traced back to CFR and its variants, which are at the heart of recent landmark results in AI benchmarks such as poker [31, 37, 39, 222]. CFR is an online algorithm for minimizing external regret—associated with (normal-form) coarse correlated equilibria. Moving forward, efficient algorithms eventually emerged for *extensive-form correlated equilibria (EFCE)* [17, 89, 106, 153] (*cf.* Morrill et al. [223, 224]), and more broadly, when Φ contains solely *linear* functions [95, 96]—corresponding to *linear correlated equilibria (LCE)*. Daskalakis et al. [81] recently took a step even further by strengthening those results whenever the underlying constraint set $\mathcal{X} \subseteq \mathbb{R}^d$ admits a membership oracle. Figure 8.1 summarizes the landscape that has emerged.

8.1.1 Our results: Φ-equilibria and Φ-Regret at the Frontier of Tractability

The primary focus of this chapter is to expand the scope of that prior research beyond linear-swap regret—associated with linear correlated equilibria—toward the frontier of tractability. We are

	Upper bound	Lower bound
Learning	$\mathsf{poly}(k)/\epsilon^2$ (Theorem 8.3)	$\min\{\sqrt{k}/4, \exp(\Omega(\epsilon^{-1/6}))\}$ (Theorem 8.5)
Computation	$\mathsf{poly}(k, \log(1/\epsilon))$ (Theorem 8.2)	Open question

Table 8.2: *Our main results for the k-dimensional set Φ^m of Definition 8.1; $k \gg d$.*

able to cope with the broad class of functions introduced below.

Definition 8.1. Given a map $m : \mathcal{X} \to \mathbb{R}^{k'}$, the set of deviations Φ^m is defined as the set of all maps $\phi : \mathcal{X} \to \mathcal{X}$ that can be can be expressed by the matrix-vector product $\mathbf{K}(\phi)m(\boldsymbol{x}) + \boldsymbol{c}(\phi)$ for some $\mathbf{K}(\phi) \in \mathbb{R}^{d \times k'}$ and $\boldsymbol{c}(\phi) \in \mathbb{R}^d$. The set of functions Φ^m has dimension at most $k := k' \cdot d + d$.

We think of k as a measure of the complexity of Φ^m; in what follows, one may imagine $k \leq \mathsf{poly}(d)$. There is a clear sense in which going beyond Definition 8.1 is daunting: even representing such functions becomes prohibitive. Indeed, we also establish lower bounds that preclude going beyond the set of deviations in Definition 8.1, showing that our results cannot be significantly improved.

As a canonical example, one can capture degree-ℓ polynomials by taking $m(\boldsymbol{x})$ to be the function that outputs all ℓ-wise (and lower) products of entries in $\boldsymbol{x}$ (hence $k = d^{O(\ell)}$), and $\mathbf{K}$ the matrix of coefficients of the polynomial. (For technical reasons, we actually consider a certain orthonormal basis for polynomials introduced formally in Definition 8.25.)

8.1.1.1 Upper Bounds

We begin by stating our results in the usual no-regret framework in the centralized model of computation, and then proceed with online learning. Our first result establishes an algorithm with running time growing polynomially in $\log(1/\epsilon)$ for the problem of computing a Φ^m-equilibrium.

Theorem 8.2 (Computation; precise version in Theorem 8.34). *Consider an n-player multilinear game Γ such that, for each player $i \in [n]$, we are given $\mathsf{poly}(n, k)$-time algorithms for the following:*

- *an oracle to compute the gradient, that is, the vector $\boldsymbol{g}_i = \boldsymbol{g}_i(\boldsymbol{x}_{-i}) \in \mathbb{R}^{d_i}$ for which $\langle \boldsymbol{g}_i(\boldsymbol{x}_{-i}), \boldsymbol{x}_i \rangle = u_i(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$ (polynomial expectation property); and*
- *a membership oracle for the strategy set $\mathcal{X}_i$.*

Suppose further that each Φ^{m_i} is k_i-dimensional per Definition 8.1 and $\|\boldsymbol{g}_i\| \leq B$. Then, an ϵ-approximate Φ^m-equilibrium of Γ can be computed in $\mathsf{poly}(n, k, \log(B/\epsilon))$ time.

We now continue with our main result for online learning.

Theorem 8.3 (Online learning; precise version in Theorem 8.38)**.** *Suppose that $\mathcal{X} \subseteq \mathbb{R}^d$ admits a membership oracle and Φ^m is k-dimensional per Definition 8.1. There is an online algorithm that guarantees at most ϵ average Φ^m-regret after* $\mathsf{poly}(k)/\epsilon^2$ *rounds with* $\mathsf{poly}(k, 1/\epsilon)$ *running time.*

The result above holds even when the learner is facing an adversary, thereby being readily applicable when learning in n-player *mutlilinear games*. In such games, each player $i \in [n]$ has a convex and compact strategy set $\mathcal{X}_i \subseteq \mathbb{R}^{d_i}$ and utility function $u_i : \mathcal{X}_1 \times \cdots \times \mathcal{X}_n \to \mathbb{R}$ that is linear in $\mathcal{X}_i$, so that $u_i(\boldsymbol{x}) = \langle \boldsymbol{g}_i, \boldsymbol{x}_i \rangle$ for some $\boldsymbol{g}_i = \boldsymbol{g}_i(\boldsymbol{x}_{-i}) \in \mathbb{R}^{d_i}$. (Extensive-form games constitute a canonical example of this framework.) In this context, Theorem 8.3 implies a fully polynomial-time algorithm (FPTAS) for computing ϵ-approximate Φ^m-equilibria in convex games.

We find these results surprising; we originally surmised that even when Φ is the set of quadratic functions (for which $k = \mathsf{poly}(d)$), the underlying online problem would be hard in the regime $\epsilon = 1/\mathsf{poly}(d)$.

In Section 8.8, we elaborate on the interesting special case where $\mathcal{X}$ is an extensive-form strategy set and Φ consists of only linear deviations. In this special case, we establish a surprising connection between Φ-regret and *communication equilibria* from Chapter 6. In particular, we will define a set of *untimed communication deviations*, and show that these are equivalent to the set of linear deviations. This relationship, as we will show, allows for the construction of significantly more efficient regret minimizers for this special case, via the machinery of DAG decision problems from Chapter 4.

8.1.1.2 Lower Bounds

A salient aspect of the above results is that the learner is allowed to output a *probability distribution* over $\mathcal{X}$. In stark contrast, and perhaps surprisingly, when the learner is constrained to output *behavioral* strategies, that is to say, points in $\mathcal{X}$, we show that regret minimization PPAD-hard even for quadratics (Theorem 8.4). We are not aware of any such hardness results pertaining to a natural online learning problem.

The key connection behind our lower bound is an observation by Hazan and Kale [146], which reveals that any Φ-regret minimizer is inadvertedly able to compute approximate fixed points of any deviation in Φ. Computing fixed points is in general a well-known (presumably) intractable problem, being PPAD-hard. In our context, the set Φ does not contain arbitrary (continuous) functions $\mathcal{X} \to \mathcal{X}$, but instead contains multilinear functions from $\mathcal{X}$ to $\mathcal{X}$. We arrive at the following hardness result.

Theorem 8.4. *If a regret minimizer $\mathcal{R}$ outputs strategies in $\mathcal{X}$, it is* PPAD*-hard to guarantee* $\textsc{Reg}_\Phi \le \epsilon$*, even with respect to quadratic deviations and* $\epsilon = 1/\mathsf{poly}(d)$.

In Section 8.10, we establish generic lower bounds against minimizing swap regret over generic strategy sets $\mathcal{X}$. Our results in that section extend the known impossibility results for swap regret over the simplex (Theorem 8.14) to extensive-form games, establishing an exponential lower bound for extensive-form games as well.

Theorem 8.5. *For any k and any $d \geq \Theta(\log^{14} k)$, there is an extensive-form decision problem with dimension d and an adversary such that the Φ-regret of the learner with respect to a k-dimensional Φ is at least ϵ when $T < \min\{\sqrt{k}/4, \exp(\Omega(\epsilon^{-1/6}))\}$.*

What is important is that, up to constant factors in the exponent of k, Theorem 8.5 matches the upper bound of Theorem 8.3. In doing so, we establish for the first time a class of deviations that characterizes—in the previous sense—no-Φ-regret learning in the adversarial setting.

8.1.2 Technical approach

Theorems 8.2 and 8.3 build on and extend certain recent developments due to Daskalakis et al. [81]. Below, we outline our key technical contributions.

Expected fixed points. The first key ingredient one requires in the framework of Gordon et al. [132] is an algorithm for computing an approximate *fixed point* of any function within the set of deviations. This fixed point computation is—at least in some sense—inherent: Hazan and Kale [146] observed that minimizing Φ-regret is computationally equivalent to computing approximate fixed points of transformations in Φ. Specifically, an efficient algorithm minimizing Φ-regret—with respect to any sequence of utilities—can be used to compute an approximate fixed point of any transformation in Φ. Given that functions in Φ in general could be nonlinear or even discontinous, this would seem to preclude the possibility of (efficient) regret minimization. Indeed, although functions in Φ have a particular structure not directly compatible with prior reductions, we show that they can still simulate generalized circuits even under low-degree deviations. At first glance, this argument seems to contradict the recent positive results of Dagan et al. [71] and Peng and Rubinstein [242].

It turns out that there is a delicate precondition on the reduction of Hazan and Kale [146] that makes all the difference: computing approximate fixed points is only necessary if the learner outputs points on $\operatorname{conv} \mathcal{X}$. In stark contrast, a crucial observation that drives our approach is that a learner who selects a probability distribution over $\mathcal{X}$ does *not* have to compute (approximate) fixed points of functions in Φ. Instead, we show that it is enough to determine what we refer to as an approximate fixed point *in expectation*. More precisely, for a deviation $\Phi \ni \phi : \mathcal{X} \to \mathcal{X}$ with an efficient representation, it is enough to compute a distribution $\mu \in \Delta(\mathcal{X})$ such that $\mathbb{E}_{\boldsymbol{x}\sim\mu}\, \phi(\boldsymbol{x}) \approx \mathbb{E}_{\boldsymbol{x}\sim\mu}\, \boldsymbol{x}$. It is quite easy to compute an approximate fixed point in expectation: take any $\boldsymbol{x}_1 \in \mathcal{X}$, and consider the sequence $\boldsymbol{x}_1, \dots, \boldsymbol{x}_L \in \mathcal{X}$ such that $\boldsymbol{x}_{\ell+1} := \phi(\boldsymbol{x}'_\ell)$ for all ℓ. Then, for $\mu := \operatorname{unif}\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_L\}$, we have

$$\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mu}[\phi(\boldsymbol{x}) - \boldsymbol{x}] = \frac{1}{L}\sum_{\ell=1}^{L}[\phi(\boldsymbol{x}'_\ell) - \boldsymbol{x}'_\ell] = \frac{1}{L}\mathop{\mathbb{E}}_{\boldsymbol{x}'_L\sim\delta(\boldsymbol{x}_L)}[\phi(\boldsymbol{x}'_L) - \boldsymbol{x}_1] = O\left(\frac{1}{L}\right).$$

This procedure can replace the fixed point oracle required by the template of Gordon et al. [132], which is prohibitive when Φ contains nonlinear functions. In fact, even in normal-form games where considering linear deviations suffices, computing a fixed point is relatively expensive, amounting to solving a linear system, dominating the per-iteration complexity.

It is worth noting that the discrepancy that has arisen between operating over $\Delta(\mathcal{X})$ versus $\mathcal{X}$ is quite singular when it comes to regret minimization in extensive-form games. Kuhn's theorem [188] is often invoked to argue about their equivalence, but in our setting it is the nonlinear nature of deviations in Φ that invalidates that equivalence.[8.1] To tie up the loose ends, we adapt the reduction of Hazan and Kale [146] to show that minimizing Φ-regret over $\Delta(\mathcal{X})$ necessitates computing approximate fixed points in expectation (Proposition 8.59), and we observe that the reductions of Dagan et al. [71] and Peng and Rubinstein [242] are indeed compatible with computing approximate fixed points in expectation (Section 8.11.2).

While the $\mathsf{poly}(1/\epsilon)$-time algorithm implied by the above argument would suffice for regret minimization, Theorem 8.2 concerns the complexity of computing expected fixed points in the regime where ϵ is exponentially small, and therefore requires a more precise expected fixed point. We address this problem by showing that expected fixed points can be computed in time polynomial in the dimension and $\log(1/\epsilon)$.

Theorem 8.6 ($\log(1/\epsilon)$-time expected fixed points)**.** *Given oracle access to $\mathcal{X}$ and $\phi : \mathcal{X} \to \mathcal{X}$, there is a $\mathsf{poly}(d, \log(1/\epsilon))$-time algorithm that computes an ϵ-expected fixed point of ϕ.*

We have described so far the role of expected fixed points when learning in an online environment (*cf.* Theorem 8.3). Going back to Theorem 8.2, expected fixed points also serve a crucial purpose in that context. Namely, Theorem 8.2 relies on the *ellipsoid against hope (*EAH*)* algorithm of Papadimitriou and Roughgarden [237], which in turn is based on running the ellipsoid algorithm on an infeasible program—the rationale being that a correlated equilibrium can be extracted from the certificate of infeasibility of that program. Now, to execute ellipsoid, one needs a separation oracle. For normal-form games, this amounts to a fixed point oracle: for any ϕ, compute $\boldsymbol{x} \in \mathcal{X}$ such that $\phi(\boldsymbol{x}) = \boldsymbol{x}$. And, as we saw earlier, ϕ is a just a stochastic matrix, and so it suffices to identify a stationary distribution of the corresponding Markov chain.

However, there are two main obstacles, which manifest themselves in each iteration of the ellipsoid, when $\mathcal{X}$ is a general convex set and Φ is allowed to contain nonlinear endomorphisms:[8.2]

1. computing a fixed point of a nonlinear ϕ is intractable; and
2. separating over the set Φ is also intractable even with respect to linear endomorphisms [81], let alone under Definition 8.1.

[8.1] Kuhn's theorem is also invalidated in extensive-form games with *imperfect recall* [189, 244, 284], in which there is also a genuine difference between mixed and behavioral strategies. In such settings, it is NP-hard to even minimize external regret.

[8.2] Recall that an *endomorphism* on $\mathcal{X}$ is a function mapping $\mathcal{X}$ to $\mathcal{X}$.

With regard to Item 1, we show that, during the execution of the ellipsoid, one might as well use *expected* fixed points, which are tractable by virtue of Theorem 8.6 we described earlier. What is intriguing is that our proof of Theorem 8.6 also relies on (a different instantiation of) EAH, and so the overall algorithm that we develop uses EAH in a nested fashion—each separation oracle as part of the execution of EAH is internally implemented via EAH!

For Item 2, we build on the framework of Daskalakis et al. [81]. In light of the inability to efficiently separate over linear endomorphisms, they observed that one can still execute EAH with access to a weaker oracle, which they refer to as a *semi-separation oracle*. Moreover, they developed a polynomial-time semi-separation oracle with respect to the set of linear endomorphisms. Building on Theorem 8.6, we significantly extend their result, establishing a semi-separation oracle for general functions, not just linear ones.

Theorem 8.7 (Semi-separation oracle for general functions)**.** *Given oracle access to $\mathcal{X}$ and $\phi : \mathcal{X} \to \mathbb{R}^d$, there is a* $\mathsf{poly}(d, \log(1/\epsilon))$*-time algorithm that either computes an ϵ-expected fixed point of ϕ, or identifies a point $\boldsymbol{x} \in \mathcal{X}$ such that $\phi(\boldsymbol{x}) \notin \mathcal{X}$.*

(In particular, in the usual case where ϕ maps to $\mathcal{X}$, the algorithm above always returns an ϵ-expected fixed point of ϕ.)

We now turn to Theorem 8.3, which revolves around the online learning setting. Equipped with our semi-separation oracle for general functions, we show that the framework of Daskalakis et al. [81] can be extended from linear endomorphisms to ones satisfying Definition 8.1. The technical pieces underpinning Theorem 8.3 are exposed in depth in Section 8.7. Importantly, the dimension of Φ turns out to be a fundamental barrier for no-regret learning in the following sense.

8.1.3 Further Related Work

The existence of no-regret algorithms goes back to the pioneering work of Blackwell [28]; the stronger notion of swap regret was crystallized and analyzed more recently [29, 143, 277]. Part of the impetus of that line of work revolves around the connection to correlated equilibria, highlighted earlier. Unfortunately, beyond online decision problems on the simplex, such no-swap-regret algorithms become inefficient when the number of pure strategies is exponential in the natural parameters of the problem—as is the case, for example, in Bayesian games, wherein $\mathcal{X} := [0, 1]^d$. Recent breakthrough results by Dagan et al. [71] and Peng and Rubinstein [242] establish a new algorithmic paradigm for minimizing swap regret, applicable even when the number of pure strategies is exponential. However, that comes at the expense of introducing an exponential dependence on $1/\epsilon$, which is unavoidable in the adversarial regime [80]. Our main interest here is in online algorithms with complexity scaling polynomially in both the dimension and $1/\epsilon$.

Besides the game-theoretic implications concerning convergence to correlated equilibria, swap regret is a fundamental concept in its own right, being intimately tied to the notion of *calibration*; namely, it has been known since the foundational work of Foster and Vohra [112] that best-responding to calibrated forecasters guarantees no-swap-regret (*cf.* Foster and Hart [113]); in relation to that connection, it is worth noting an important, recent body of work that bypasses the intractability of calibration in high dimensions [151, 233, 255]. Swap regret is also more robust against exploitation, in a sense formalized in a series of recent papers [14, 84, 140, 207].

In particular, within that line of work, Mansour et al. [207] introduced the notion of *polytope swap regret*, which comprises deviations that allow the vertices of the underlying polytope to be swapped with each other—points inside the polytope are mapped in accordance with the (worst-case) convex combination of vertices. It is currently unknown whether there is an efficient algorithm for minimizing polytope swap regret.

The more flexible framework of Φ-regret, which has been gaining considerable traction in recent years, allows one to circumvent this possible intractability restricting the set of deviations. In addition to the research highlighted above, chiefly in the context of extensive-form games, we now provide some further pointers for the interested reader. Bernasconi et al. [25] considered the more challenging setting of so-called "pseudo-games," wherein players have joint constraint sets. Φ-equilibria in such settings have certain counterintuitive properties; for example, they are not necessarily convex. Φ-equilibria have also garnered attention in the context of Markov (aka. stochastic) games, going back to the work of Greenwald and Hall [135]; for more recent research, we refer to Cai et al. [47], Erez et al. [91], Jin et al. [162], and references therein. Even more broadly, we refer to Cai et al. [46], Şeref Ahunbay [70] for efficient solution concepts in nonconcave games [75].

Finally, we remark that the hardness result of Daskalakis et al. [81] for separating over linear endomorphisms does not apply to polytopes represented with a polynomial number of constraints. Indeed, it is relatively straightforward to implement a membership oracle for such polytopes (Theorem 8.53). In contrast, it is computationally hard to decide membership for low-degree polynomials [81].

8.2 Preliminaries

Before proceeding, we must introduce some notation and background that will be fundamental in this chapter. Most of this chapter will be concerned with *general multilinear games*, not just extensive-form games as in the previous chapters. Therefore, the pure strategy set is $\Pi_i \subset \mathbb{R}^{d_i}$. To align with extensive-form terminology, we will continue to refer to $\mathcal{X}_i := \operatorname{conv} \Pi_i$ as the set of *behavioral strategies*. We will still demand throughout the chapter (unless otherwise stated) that the utility functions $u_i : \mathcal{X}_1 \times \cdots \times \mathcal{X}_n \to \mathbb{R}$ are multilinear, that is, linear in each $\boldsymbol{x}_i$. Since extensive-form games are multilinear games (via the sequence form), the results of this section for multilinear games apply directly to extensive-form games, although they also apply more generally as well.

As usual, the subscript i will be dropped when there is only one player involed, for example, when talking about single-player regret minimization.

Oracle access. Throughout this chapter, we assume that we have access to the convex and compact constraint set $\mathcal{X}$ via an oracle. (For multi-player games, oracle access is posited for the constraint set $\mathcal{X}_i$ of each player, which can be thereby extended to $\mathcal{X} := \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$.) In particular, the following three types of oracles are commonly considered in the literature:

- *membership*: given a point $\boldsymbol{x} \in \mathbb{R}^d$, decide whether $\boldsymbol{x} \in \mathcal{X}$;
- *separation*: given a point $\boldsymbol{x} \in \mathbb{R}^d$, decide whether $\boldsymbol{x} \in \mathcal{X}$, and if not, output a hyperplane $\boldsymbol{w} \in \mathbb{R}^d$ separating $\boldsymbol{x}$ from $\mathcal{X}$: $\langle \boldsymbol{x}, \boldsymbol{w} \rangle > \langle \boldsymbol{x}', \boldsymbol{w} \rangle$ for all $\boldsymbol{x}' \in \mathcal{X}$;
- *linear optimization*: given a point $\boldsymbol{u} \in \mathbb{R}^d$, output any point in $\operatorname{argmax}_{\boldsymbol{x} \in \mathcal{X}} \langle \boldsymbol{x}, \boldsymbol{u} \rangle$.

Under the assumption that $\mathcal{B}_r(\cdot) \subseteq \mathcal{X} \subseteq \mathcal{B}_R(\mathbf{0})$, the three oracles described above are known to be (polynomially) equivalent [138]—up to logarithmic factors in R and $1/r$. The previous geometric condition can always be met by bringing $\mathcal{X}$ into *isotropic position*, which means that, for a uniformly sampled $\boldsymbol{x} \sim \mathcal{X}$, we have $\mathbb{E}[\boldsymbol{x}] = 0$ and $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top] = \mathbf{I}_{d \times d}$. This can be achieved in polynomial time through an affine transformation [30, 169, 202]; it is easy to see that minimizing Φ-regret after applying that transformation suffices in order to minimize Φ-regret in the original space.[8.3] As a result, we can assume throughout that, for example, $\mathcal{B}_1(\mathbf{0}) \subseteq \mathcal{X} \subseteq \mathcal{B}_d(\mathbf{0})$ [202].

Remark 8.8 (Weak oracles)**.** When dealing with general convex sets, the oracles posited above can return points supported on irrational numbers. To address this issue in the usual Turing model of computation, it suffices to consider weaker versions of those oracles that allow for some small slackness $\epsilon > 0$. Our analysis in the sequel can be extended to account for such imprecision.

8.2.1 Φ-Regret Minimization

We first recall the usual framework of online learning. The interaction lasts for T rounds. On round t, the learner selects a point $\boldsymbol{x}^t \in \mathcal{X}$.

In the framework of online learning, a learner interacts with an adversary over a sequence of rounds. In each round, the learner selects a strategy $\mu^t \in \Delta(\mathcal{X})$, whereupon the adversary constructs a utility function which is subsequently observed by the learner. The adversary is allowed to be strongly adaptive, so that the utility function at the tth round $u^t : \mathcal{X} \ni \boldsymbol{x} \mapsto \langle \boldsymbol{u}^t, \boldsymbol{x} \rangle$ can depend on the strategy μ^t of the learner at that round. We assume that utilities belong to $\mathcal{U} := \{\boldsymbol{u} : |\langle \boldsymbol{u}, \boldsymbol{x} \rangle| \le 1 \ \forall \boldsymbol{x} \in \mathcal{X}\}$. It will be convenient to use $\|\boldsymbol{x}\|_{\mathcal{X}} := \max_{\boldsymbol{u} \in \mathcal{U}} \langle \boldsymbol{u}, \boldsymbol{x} \rangle$ for the induced norm.

We measure the performance of an online learning algorithm as follows. Suppose that $\Phi \subseteq \mathcal{X}^{\mathcal{X}}$ is a set of deviations. If the learner outputs in each round a *mixed strategy* $\mu^t \in \Delta(\mathcal{X})$, its (time-average) Φ-*regret* [135, 278] is defined as

$$\textsc{Reg}_\Phi(T) := \frac{1}{T} \max_{\phi \in \Phi} \sum_{t=1}^{T} \left\langle \boldsymbol{u}^t, \mathop{\mathbb{E}}_{\boldsymbol{x}^t \sim \mu^t} [\phi(\boldsymbol{x}^t) - \boldsymbol{x}^t] \right\rangle. \tag{8.1}$$

In the special case where Φ contains only *constant transformations*, one recovers the notion of *external regret*. On the other extreme, *swap regret* corresponds to Φ containing all functions $\mathcal{X} \to \mathcal{X}$.

[8.3] The formal argument is deferred to the appendix of the full paper [320].

Algorithm 8.3 (GGM): Construction of a Φ-regret minimizer from a fixed point oracle and an external regret minimizer

```
procedure NextStrategy()
    $\phi^t \leftarrow \mathcal{R}_\Phi.\textsc{NextStrategy}()$
    return $\boldsymbol{x}^t \leftarrow$ an $\epsilon$-fixed point of $\phi^t$
procedure ObserveUtility($\boldsymbol{u}^t$)
    $\mathcal{R}_\Phi.\textsc{ObserveUtility}(\Phi \ni \phi \mapsto \langle \boldsymbol{u}^t, \mathit{\Phi}(\boldsymbol{x}^t) \rangle)$
```

We are interested in algorithms whose regret is bounded by ϵ after $T = \mathsf{poly}(N, 1/\epsilon)$ rounds. We refer to such algorithms as *fully-polynomial no-regret learners*.

Remark 8.9 (Swap versus internal regret)**.** When it comes to defining correlated equilibria in normal-form games, there are two prevalent definitions appearing in the literature; one is based on *internal regret*, while the other on *swap regret* (*e.g.*, [120, 130]). The key difference is that internal regret only contains deviations that swap a *single action*—thereby being weaker. Nevertheless, it is not hard to see that swap regret can only be larger by a factor of $|\Pi|$ [29], where we recall that Π denotes the set of pure strategies. So, in normal-form games those two definitions are polynomially equivalent, and in most applications one can safely switch from one to the other.

However, this is certainly not the case in games with an exponentially large action space, such as extensive-form games. In fact, the definition of internal regret itself is problematic when the action set is exponentially large: the uniform distribution always attains an error of at most $1/|\Pi|$. Consequently, any guarantee for $\epsilon \geq 1/|\Pi|$ is vacuous. That is, if $|\Pi|$ is exponentially large, an algorithm that requires a number of iterations polynomial in $1/\epsilon$—which is what we expect to get from typical no-regret dynamics—would need an exponential number of iterations to yield a non-trivial guarantee; this issue with internal regret was also observed by Fujii [117]. Nevertheless, internal regret in the context of games with an exponentially large action set was used in a recent work by Chen et al. [60], who provided oracle-efficient algorithms for minimizing internal regret.

8.2.1.1 The GGM Construction

Gordon et al. [132], building on earlier work by Blum and Mansour [29] and Stoltz and Lugosi [277], came up with a general recipe for minimizing Φ-regret. That construction relies on a no-regret learning algorithm on the set of deviations Φ, which we denote by $\mathcal{R}_\Phi$. Then GGM is a Φ-regret minimizer on $\mathcal{X}$. It has the following regret guarantee.

Theorem 8.10 (Regret of GGM [132])**.** *Suppose that* $\textsc{Reg}(T)$ *is the external regret incurred by* $\mathcal{R}_\Phi$*. After T rounds of* GGM*, we have*

$$\max_{\phi \in \Phi} \frac{1}{T} \sum_{t=1}^{T} \langle \boldsymbol{u}^t, \phi(\boldsymbol{x}^t) - \boldsymbol{x}^t \rangle \leq \textsc{Reg}(T) + \epsilon.$$

Deviations Φ	**Equilibrium concept**	**References**
Constant (external), $\Phi = \{\phi : \boldsymbol{x} \mapsto \boldsymbol{x}_0 \mid \boldsymbol{x}_0 \in \mathcal{X}\}$	Normal-form coarse correlated	Moulin and Vial [226]
Trigger (see Section 8.8)	Extensive-form correlated	von Stengel and Forges [291]
Communication (see Section 8.8)	Communication	Forges [109], Myerson [228]
Linear / Untimed communication	Linear correlated	Farina and Pipis [95]; Section 8.8
Swap, $\Phi = \mathcal{X}^{\mathcal{X}}$	Normal-form correlated	Aumann [15]

Table 8.4: *Some examples of deviation sets Φ and corresponding notions of correlated equilibrium, in increasing order of size of Φ (and thus increasing tightness of the equilibrium concept)*

In Chapter 8, we will relax the requirement of needing (approximate) fixed points, while at the same time maintaining the guarantee of Theorem 8.10.

8.2.1.2 Convergence to Correlated Equilibria

Notions of Φ-regret correspond naturally to notions of correlated equilibria. Therefore, our results also have implications for no-regret learning algorithms that converge to correlated equilibria. Indeed, the following standard result follows immediately from the definitions of equilibrium and regret.

Proposition 8.11 (No-Φ-regret learners converge to Φ-equilibrium)**.** *Suppose that every player i plays according to a regret minimizer whose Φ_i-regret is at most ϵ after T rounds. Let $\mu_i^t \in \Delta(\mathcal{X}_i)$ be the distribution played by player i at round t. Let $\mu^t \in \Delta(\mathcal{X}_1)\times\cdots\times\Delta(\mathcal{X}_n)$ be the product distribution whose marginal on $\mathcal{X}_i$ is μ_i^t. Then the* average strategy profile, *that is, the uniform distribution on $\{\mu^1, \dots, \mu^T\}$, is an ϵ-Φ-equilibrium.*

Some common choices of Φ, and corresponding equilibrium notions, are in Table 8.4.

8.2.2 Swap Regret

Swap regret is the important special case of Φ-regret when Φ consists of *all* maps $\mathcal{X} \to \mathcal{X}$. It captures, via Proposition 8.11, the notion of *correlated equilibrium*, which is the strongest possible notion expressible in the language of Φ-regret and Φ-equilibria. In this section, we review known results about no-swap-regret learning algorithms.

8.2.2.1 Normal-form games

In a *normal-form game*, every player's decision problem consists of a single decision point with N actions, that is, $\mathcal{X} = \Delta(N)$ where $\boldsymbol{e}_k$ is the kth standard basis vector in $\mathbb{R}^N$. Blum and Mansour [29] showed that efficient algorithms exist for minimizing swap regret over the simplex.

Theorem 8.12 ([29])**.** *There exists a no-regret learning algorithm for simplices that achieves average swap regret ϵ within $T = \tilde{O}(N/\epsilon^2)$ rounds.*

One may wonder whether this is optimal, *e.g.*, whether it is possible to achieve a logarithmic dependence on m. Recent simultaneous work by Dagan et al. [71] and Peng and Rubinstein [242] has essentially completely answered this question for normal-form games.

Theorem 8.13 ([71, 242], upper bound)**.** *There exists a no-regret learning algorithm for simplices that achieves average swap regret ϵ within $T = (\log N)^{\tilde{O}(1/\epsilon)}$ rounds.*

Both papers also provided (nearly-)matching lower bounds. Here we state a particularly simple-to-state lower bound proven by Dagan et al. [71].

Theorem 8.14 (Theorem 4.1 of Dagan et al. [71], lower bound)**.** *Let $T < N/4$. Then, in the there exists an oblivious adversary such that the swap regret of any learner for the N-simplex is $\Omega(\log^{-5} T)$.*

8.2.2.2 Extensive-Form Games and Tree-Form Strategy Sets

For more general extensive-form games, the picture is less clear. For an upper bound, one can consider a tree-form strategy set $\mathcal{X}$ with N pure strategies (*i.e.*, $|\mathcal{X}| = N$) as simply an "easier version" of a normal-form decision problem where each pure strategy is treated as a different action, *i.e.*, where the strategy set is the N-simplex. Theorem 8.13 therefore implies a similar bound on swap regret for tree-form decision problems.[8.4]

Corollary 8.15 ([71, 242], tree-form upper bound)**.** *Let $\mathcal{X} \subset \{0, 1\}^m$ be a tree-form strategy set. There exists a no-regret learning algorithm for tree-form decision problems that achieves swap regret ϵ after $T = (\log |\mathcal{X}|)^{\tilde{O}(1/\epsilon)} \le d^{\tilde{O}(1/\epsilon)}$ rounds.*

Showing a matching lower bound for extensive form, however, remained open. The main difficulty is that the adversary is restricted to *linear* utility functions $u^t : \mathcal{X} \to \mathbb{R}$; the adversary in Theorem 8.14 does not use linear utility functions when the extensive-form game is interpreted as a normal-form game over N actions as described above. We will close this discrepancy in Section 8.10 by showing a lower bound that almost matches Corollary 8.15.

8.2.3 Ellipsoid Against Hope

The *ellipsoid against hope (*EAH*)* algorithm was famously introduced by Papadimitriou and Roughgarden [237] to compute correlated equilibria in succinct, multi-player games—under the polynomial expectation property. A further crucial assumption in the approach of Papadimitriou and Roughgarden [237] is that the game is of *polynomial type*, in that the number of actions (or pure strategies) is polynomial in the representation of the game. In contrast to normal-form games, extensive-form games—and many other natural classes of games—are *not* of polynomial

[8.4] As stated, the bound is only information-theoretic. However, the information-theoretic bound is implementable by an efficient (*i.e.*, poly$(d, 1/\epsilon)$-time-per-iteration) algorithm, which is described by Dagan et al. [71] and Peng and Rubinstein [242], and is beyond the scope of this thesis.

type. Farina and Pipis [96] recently showed how to apply EAH in the context of extensive-form games—albeit only for LCE; as we have seen, the complexity of NFCE remains open. We begin by recalling their framework, which crystallizes the approach of Papadimitriou and Roughgarden [237]. We then proceed by introducing the more powerful approach of Daskalakis et al. [81], which is crucial to compute LCE under general convex constraint sets, and which will form the basis for our approach as well.

Consider an arbitrary optimization problem of the form

$$\text{find} \quad \mu \in \Delta(\mathcal{X}) \quad \text{s.t.} \quad \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mu} \langle \boldsymbol{y}, G(\boldsymbol{x})\rangle \geq 0 \quad \forall \boldsymbol{y} \in \mathcal{Y}, \tag{8.2}$$

where $\mathcal{X} \subseteq \mathbb{R}^d$, $\mathcal{Y} \subseteq \mathbb{R}^k$, and $G : \mathcal{X} \to \mathbb{R}^k$ is a function such that $\|G(\boldsymbol{x})\| \leq B$ for all $\boldsymbol{x} \in \mathcal{X}$. The crux in (8.2) lies in the fact that μ resides in a high-dimensional (indeed, an infinite-dimensional) space, making standard approaches of little use. EAH addresses that challenge, as we describe next.

Suppose that we are given a $\mathsf{poly}(d, k)$-time evaluation oracle for G and a separation oracle (SEP) for $\mathcal{Y}$, assumed to be *well-bounded*: $\mathcal{B}_r(\cdot) \subseteq \mathcal{Y} \subseteq \mathcal{B}_R(\mathbf{0})$. In addition, we assume that we have access to a *good-enough-response* (GER) oracle, which, given any $\boldsymbol{y} \in \mathcal{Y}$, returns $\boldsymbol{x} \in \mathcal{X}$ such that $\langle \boldsymbol{y}, G(\boldsymbol{x})\rangle \geq 0$. The EAH algorithm allows us to solve problems of the form (8.2) with just the above tools. In particular, EAH proceeds by considering an ϵ-approximate version of the dual of (8.2).

$$\text{find} \quad \boldsymbol{y} \in \mathcal{Y} \quad \text{s.t.} \quad \langle \boldsymbol{y}, G(\boldsymbol{x})\rangle \leq -\epsilon \quad \forall \boldsymbol{x} \in \mathcal{X}. \tag{8.3}$$

Since a GER oracle exists, (8.3) is infeasible. Moreover, a certificate of infeasibility of (8.3) provides an ϵ-approximate solution to (8.2). Thus, it suffices to run the ellipsoid algorithm on (8.3) and extract a certificate of infeasibility. This is precisely what EAH does, as formalized in Theorem 8.16.

Theorem 8.16 (Generalized form of EAH [96, 237])**.** *Suppose that we have* $\mathsf{poly}(d, k)$*-time algorithms for the following:*

- *an evaluation oracle for G, where $\|G(\boldsymbol{x})\| \leq B$ for all $\boldsymbol{x} \in \mathcal{X}$;*
- *a GER oracle for (8.2); and*
- *a separation oracle (SEP) for the well-bounded set $\mathcal{Y}$.*

Then, there is an algorithm that runs in time $\mathsf{poly}(d, k, \log(B/\epsilon))$ *and returns an ϵ-approximate solution to (8.2).*

8.3 Weakening the Assumptions of Ellipsoid Against Hope

When it comes to computing LCE under general constraint sets, Theorem 8.16 is not enough: Daskalakis et al. [81] showed that separating over $\mathcal{Y}$—the set of linear endomorphisms—is hard. In light of this fact, their key observation was that an ϵ-approximate solution to (8.2) can still

Algorithm 8.5 (EAH): Ellipsoid against hope [96, 237]

```
input:
    parameters $R_y, r_y > 0$ such that $\mathcal{B}_{r_y}(\cdot) \subseteq \mathcal{Y} \subseteq \mathcal{B}_{R_y}(\mathbf{0})$
    precision parameter $\epsilon > 0$
    parameter $B > 0$ such that $\|G(\boldsymbol{x})\| \le B$ for all $\boldsymbol{x} \in \mathcal{X}$
    a GER oracle for (8.2)
    a SEP oracle for $\mathcal{Y}$
output: A sparse, $\epsilon$-approximate solution $\mu \in \Delta(\mathcal{X})$ of (8.2)
initialize the ellipsoid $\mathcal{E} := \mathcal{B}_{R_y}(\mathbf{0})$
initialize $\tilde{\mathcal{Y}} := \mathcal{B}_{R_y}(\mathbf{0})$
while $\mathrm{vol}(\mathcal{E}) \ge \mathrm{vol}(\mathcal{B}_{\epsilon/B}(\cdot))$ do
    let $\boldsymbol{y}$ be the center of $\mathcal{E}$
    if $\boldsymbol{y} \in \mathcal{Y}$ then
        let $\boldsymbol{x}$ be a good enough response with respect to $\boldsymbol{y}$ (via the GER oracle)
        let $H$ be the halfspace $\{\boldsymbol{y} \in \mathbb{R}^k : \langle \boldsymbol{y}, G(\boldsymbol{x}) \rangle \le 0\}$
    else let $H$ be a halfspace that separates $\boldsymbol{y}$ from $\mathcal{Y}$ (via the SEP oracle)
    update $\mathcal{E}$ to the minimum volume ellipsoid containing $\mathcal{E} \cap H$
let $\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(T)}$ be the GER oracle responses produced in the process above
define $\mathbf{G} := [G(\boldsymbol{x}^{(1)}) \mid \dots \mid G(\boldsymbol{x}^{(T)})] \in \mathbb{R}^{k \times T}$
compute a solution $\boldsymbol{\lambda}$ to the convex program

        $\text{find} \quad \boldsymbol{\lambda} \in \Delta(T) \quad \text{s.t.} \quad \min_{\boldsymbol{y} \in \tilde{\mathcal{Y}}} \boldsymbol{\lambda}^\top \mathbf{G}^\top \boldsymbol{y} \ge -\epsilon$

return the distribution $\mu \in \Delta(\mathcal{X})$ that outputs $\boldsymbol{x}^{(t)}$ with probability $\lambda^{(t)}$
```

be computed given access to a weaker oracle. Namely, instead of requiring both a GER and a SEP oracle, as in Theorem 8.16, Daskalakis et al. [81] showed that it suffices to implement the following oracle: for any given $\boldsymbol{y} \in \mathbb{R}^k$ (*not* necessarily in $\mathcal{Y}$), compute *either*

1. a good-enough response $\boldsymbol{x} \in \mathcal{X}$, *or*
2. a hyperplane separating $\boldsymbol{y}$ from $\mathcal{Y}$.

Although separating over $\mathcal{Y}$ is hard, this weaker oracle suffices to recover the guarantee of Theorem 8.16, and this is enough to compute linear correlated equilibria in games. Yet, for our purposes, it will be necessary to relax the aforedescribed oracle even further, as formalized below.

Definition 8.17 (ExpectedGERorSEP)**.** Consider problem (8.2). The oracle ϵ-ExpectedGERorSEP works as follows. It takes as input $\boldsymbol{y} \in \mathbb{R}^k$, and it computes *either*

1. an ϵ-approximate good-enough-response *in expectation* $\mu \in \Delta(\mathcal{X})$, such that $\mathbb{E}_{\boldsymbol{x} \sim \mu} \langle \boldsymbol{y}, G(\boldsymbol{x}) \rangle \ge -\epsilon$, such that $\mathrm{supp}(\mu)$, *or*
2. a hyperplane ϵ-approximately separating $\boldsymbol{y}$ from $\mathcal{Y}$.

Algorithm 8.6 (ExpectedEAH): Ellipsoid against hope under ExpectedGERorSEP oracle [81]

```
input:
    parameters $R_y, r_y > 0$ such that $\mathcal{B}_{r_y}(\cdot) \subseteq \mathcal{Y} \subseteq \mathcal{B}_{R_y}(\mathbf{0})$
    precision parameter $\epsilon > 0$
    parameter $B > 0$ such that $\|G(\boldsymbol{x})\| \le B$ for all $\boldsymbol{x} \in \mathcal{X}$
    a ExpectedGERorSEP oracle (Definition 8.17)
output: A sparse, $\epsilon$-approximate solution $\mu \in \Delta(\mathcal{X})$ of (8.2)
initialize the ellipsoid $\mathcal{E} := \mathcal{B}_{R_y}(\mathbf{0})$
initialize $\tilde{\mathcal{Y}} := \mathcal{B}_{R_y}(\mathbf{0})$
while $\mathrm{vol}(\mathcal{E}) \ge \mathrm{vol}(\mathcal{B}_{\epsilon/B}(\cdot))$ do
    query the ExpectedGERorSEP oracle on the center of $\mathcal{E}$
    if it returns a good-enough-response $\mu \in \Delta(\mathcal{X})$ then
        let $H$ be the halfspace $\{\boldsymbol{y} \in \mathbb{R}^k : \mathbb{E}_{\boldsymbol{x} \sim \mu}\langle \boldsymbol{y}, G(\boldsymbol{x}) \rangle \le 0\}$
    else
        let $H$ be a halfspace that separates $\boldsymbol{y}$ from $\mathcal{Y}$
        update $\tilde{\mathcal{Y}} := \tilde{\mathcal{Y}} \cap H$
    update $\mathcal{E}$ to the minimum volume ellipsoid containing $\mathcal{E} \cap H$
let $\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(T)}$ be the GER oracle responses produced in the process above
define $\mathbf{G} := [G(\boldsymbol{x}^{(1)}) \mid \dots \mid G(\boldsymbol{x}^{(T)})] \in \mathbb{R}^{k \times T}$
compute a solution $\boldsymbol{\lambda}$ to the convex program

    $$\text{find} \quad \boldsymbol{\lambda} \in \Delta(T) \quad \text{s.t.} \quad \min_{\boldsymbol{y} \in \tilde{\mathcal{Y}}} \boldsymbol{\lambda}^\top \mathbf{G}^\top \boldsymbol{y} \ge -\epsilon$$

return $\frac{1}{T} \sum_{t=1}^{T} \boldsymbol{\lambda}(t) \mu^{(t)}$
```

Compared to the oracle described earlier (Items 1 and 2), Definition 8.17 makes two further concessions: first, the good-enough-response can now be a distribution, so long as it has polynomial support; and second, both GER and SEP can have some small slack $\epsilon > 0$. Both of those relaxations will be essential for our applications. We now summarize the key guarantee.

Theorem 8.18 (Generalization of Theorem 8.16 and Daskalakis et al. [81]). *Suppose that we have* $\mathsf{poly}(d, k, \log(1/\epsilon))$*-time algorithms for the following:*

- *an ϵ-ExpectedGERorSEP oracle with respect to the well-bounded set $\mathcal{Y}$, and*
- *an evaluation oracle for G, where $\|G(\boldsymbol{x})\| \le B$ for all $\boldsymbol{x} \in \mathcal{X}$.*

Then ExpectedEAH *runs in time* $\mathsf{poly}(d, k, \log(B/\epsilon))$ *and returns an ϵ-approximate solution to* (8.2).

8.4 Sets of Deviations with Polynomial Dimension

In this section, we formally introduce the assumptions we make concerning the feature map m of Definition 8.1, and we then provide a canonical example that satisfies our blanket assumptions.

Assumption 8.19. We make the following assumptions regarding Φ^m and m of Definition 8.1:

- $m : \mathcal{X} \to \mathbb{R}^{k'}$ is computable in $\mathsf{poly}(k)$ time.
- $\operatorname{conv} m(\mathcal{B}_1(\mathbf{0})) \supseteq \mathcal{B}_\delta(\mathbf{0})$ for some $\delta \geq \mathsf{poly}(1/k)$.
- $\|m(\boldsymbol{x})\| \leq \mathsf{poly}(k)$ for all $\boldsymbol{x} \in \mathcal{X}$, and $m(\mathbf{0}) = \mathbf{0}$.
- Φ^m contains the identity map.

Remark 8.20 (Functions on the vertices)**.** Let Π be the set of extreme points of $\mathcal{X}$. Our positive results (Theorems 8.34 and 8.38) only evaluate ϕ at extreme points, so they would operate identically if we instead defined our maps ϕ to be $\Pi \to \mathcal{X}$.

The definition above places some minimal assumptions on the feature mapping m to ensure that Φ^m is geometrically well behaved. Indeed, we first show that the set of transformations Φ^m under Assumption 8.19 is well-bounded.

Proposition 8.21. *Let $\mathcal{X} \subseteq \mathbb{R}^d$ be a convex and compact set such that $\mathcal{B}_r(\mathbf{0}) \subseteq \mathcal{X} \subseteq \mathcal{B}_R(\mathbf{0})$, with $R \geq 1$ and $r < R$. Suppose further that $\|m(\boldsymbol{x})\| \leq M$ for all $\boldsymbol{x} \in \mathcal{B}_R(\mathbf{0})$, with $M = M(R) \geq 1$; $\operatorname{conv} m(\mathcal{B}_r(\mathbf{0})) \supseteq \mathcal{B}_\delta(\mathbf{0})$ for some $\delta = \delta(r) > 0$; and $m(\mathbf{0}) = \mathbf{0}$. Then,*

$$\mathcal{B}_{r'}(\mathbf{0}) \subseteq \Phi^m \subseteq \mathcal{B}_{R'}(\mathbf{0}),$$

where $r' := r/2M(R)$ and $R' := R\left(\frac{2\sqrt{d}}{\delta(r)} + 1\right)$.

Proof. Below, for convex and compact $\mathcal{A}, \mathcal{B} \subseteq \mathbb{R}^d$, we use the notation

$$\Phi^m(\mathcal{A}, \mathcal{B}) := \left\{(\mathbf{K}, \boldsymbol{c}) \in \mathbb{R}^{k+d} : \mathbf{K} m(\boldsymbol{x}) + \boldsymbol{c} \in \mathcal{B} \quad \forall \boldsymbol{x} \in \mathcal{A}\right\}.$$

Lemma 8.22. *Let $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}$ be convex and compact sets. If $\mathcal{A} \supseteq \mathcal{C}$ and $\mathcal{B} \subseteq \mathcal{D}$, then $\Phi^m(\mathcal{A}, \mathcal{B}) \subseteq \Phi^m(\mathcal{C}, \mathcal{D})$.*

Proof. Consider any $(\mathbf{K}, \boldsymbol{c}) \in \Phi^m(\mathcal{A}, \mathcal{B})$. By definition, it holds that $\mathbf{K}\boldsymbol{x} + \boldsymbol{c} \in \mathcal{B}$ for all $\boldsymbol{x} \in \mathcal{A}$. Since $\mathcal{C} \subseteq \mathcal{A}$, it follows that $\mathbf{K}\boldsymbol{x} + \boldsymbol{c} \in \mathcal{B}$ for all $\boldsymbol{x} \in \mathcal{C}$, and in particular, $\mathbf{K}\boldsymbol{x} + \boldsymbol{c} \in \mathcal{D}$ since $\mathcal{B} \subseteq \mathcal{D}$. □

Lemma 8.23. *Let $\mathcal{X} \subseteq \mathbb{R}^d$ be a convex and compact set such that $\mathcal{B}_r(\mathbf{0}) \subseteq \mathcal{X} \subseteq \mathcal{B}_R(\mathbf{0})$, with $R \geq 1$. Suppose further that $\|m(\boldsymbol{x})\| \leq M$ for all $\boldsymbol{x} \in \mathcal{B}_R(\mathbf{0})$, where $M = M(R) \geq 1$. Then, $\Phi^m \supseteq \mathcal{B}_{r'}(\phi_\mathbf{0})$, where $r' := r/2M(R)$ and $\phi_\mathbf{0} := (\mathbf{0}, \mathbf{0})$ is the constant transformation $\boldsymbol{x} \mapsto \mathbf{0}$.*

Proof. By Lemma 8.22, it suffices to prove $\mathcal{B}_{r'}(\phi_{\mathbf{0}}) \subseteq \Phi^m(\mathcal{X}, \mathcal{B}_r(\mathbf{0}))$. Consider any $(\mathbf{K}, \boldsymbol{c}) \in \mathcal{B}_{r'}(\phi_{\mathbf{0}})$, which means that $\|\mathbf{K}\|_F^2 + \|\boldsymbol{c}\|^2 \leq (r/2M(R))^2$. Then, for any $\boldsymbol{x} \in \mathcal{X}$,

$$\|\mathbf{K}m(\boldsymbol{x}) + \boldsymbol{c}\| \leq \|\mathbf{K}\|_F \|m(\boldsymbol{x})\| + \|\boldsymbol{c}\| \leq r.$$

This means that $\mathcal{B}_{r'}(\phi_{\mathbf{0}}) \subseteq \Phi^m(\mathcal{X}, \mathcal{B}_r(\mathbf{0}))$, and the proof follows. □

Lemma 8.24. *Suppose that* $\operatorname{conv} m(\mathcal{B}_r(\mathbf{0})) \supseteq \mathcal{B}_\delta(\mathbf{0})$ *for some* $\delta = \delta(r) > 0$ *and* $m(\mathbf{0}) = \mathbf{0}$. *Then, assuming that* $r < R$,

$$\Phi^m(\mathcal{B}_r(\mathbf{0}), \mathcal{B}_R(\mathbf{0})) \subseteq \mathcal{B}_{R'}(\mathbf{0}),$$

where $R' := R\left(\frac{2\sqrt{d}}{\delta(r)} + 1\right)$.

Proof. Consider any $(\mathbf{K}, \boldsymbol{c}) \in \Phi^m(\mathcal{B}_r(\mathbf{0}), \mathcal{B}_R(\mathbf{0}))$. By definition, we have $\|\mathbf{K}m(\boldsymbol{x}) + \boldsymbol{c}\| \leq R$ for all $\boldsymbol{x} \in \mathcal{B}_r(\mathbf{0})$. Since $m(\mathbf{0}) = \mathbf{0}$, it follows that $\|\boldsymbol{c}\| \leq R$. Thus, $\|\mathbf{K}m(\boldsymbol{x})\| \leq \|\mathbf{K}m(\boldsymbol{x}) + \boldsymbol{c}\| + \|\boldsymbol{c}\| \leq 2R$ for all $\boldsymbol{x} \in \mathcal{B}_r(\mathbf{0})$. Now, let $\boldsymbol{x}' \in \mathbb{R}^{k'}$ with $\|\boldsymbol{x}'\| = 1$ be such that $\|\mathbf{K}\boldsymbol{x}'\| = \|\mathbf{K}\|$, where $\|\mathbf{K}\|$ is the spectral norm of $\mathbf{K}$. Since we have assumed that $\operatorname{conv} m(\mathcal{B}_r(\mathbf{0})) \supseteq \mathcal{B}_\delta(\mathbf{0})$, it follows that there exist $\lambda_1, \dots, \lambda_{k'+1}$, with $\lambda_1, \dots, \lambda_{k'+1} \geq 0$ and $\sum_{j=1}^{k'+1} \lambda_j = 1$, and $\boldsymbol{x}_1, \dots, \boldsymbol{x}_{k'+1} \in \mathcal{B}_r(\mathbf{0})$ (by Carathéodory's theorem) such that $\sum_{j=1}^{k'+1} \lambda_j m(\boldsymbol{x}_j) = \delta \boldsymbol{x}'$. As a result,

$$\delta\|\mathbf{K}\| = \|\mathbf{K}(\delta\boldsymbol{x}')\| = \left\|\mathbf{K}\left(\sum_{j=1}^{k'+1} \lambda_j m(\boldsymbol{x}_j)\right)\right\| \leq \sum_{j=1}^{k'+1} \lambda_j \|\mathbf{K}m(\boldsymbol{x}_j)\| \leq 2R.$$

Finally, we have $\|\mathbf{K}\|_F \leq \sqrt{d}\|\mathbf{K}\|$, and the claim follows. □

The proof of Proposition 8.21 follows directly by combining Lemmas 8.22 to 8.24. □

We are now ready to provide a canonical, concrete example of deviations that satisfy Definition 8.1 under Assumption 8.19. As we alluded to earlier in our introduction, it is the family of low-degree polynomials; in particular, it will be convenient to work with the Legendre basis.

Definition 8.25. Let $P_0(x) = 1$ and $P_1(x) = x$. The $(\ell + 1)$th *Legendre polynomial* is given by the recurrence $(\ell + 1)P_{\ell+1}(x) - (2\ell + 1)xP_\ell(x) + \ell P_{\ell-1}(x) = 0$.

These polynomials have a convenient orthogonality property over $[-1, 1]$:

$$\int_{-1}^{1} P_\ell(x)P_{\ell'}(x)dx = \begin{cases} \frac{1}{2\ell+1} & \text{if } \ell = \ell', \\ 0 & \text{otherwise.} \end{cases} \tag{8.4}$$

For convenience, we shall consider the rescaled polynomial $\bar{P}_\ell := \sqrt{2\ell + 1}P_\ell$, so that $\int_{-1}^{1} \bar{P}_\ell(x)^2 dx = 1$. We now define

$$m(\boldsymbol{x}) := \left(\prod_{j=1}^{d} \bar{P}_{\ell_j}(\boldsymbol{x}(j))\right)_{1 \leq \ell_1 + \dots \ell_d \leq \ell}. \tag{8.5}$$

We now show that the above mapping satisfies Assumption 8.19.

Proposition 8.26. *Let $m : \mathcal{X} \to \mathbb{R}^{k'}$ per (8.5), where $k' = \binom{d+\ell}{\ell} - 1$. $\bar{m} : \boldsymbol{x} \mapsto m(\sqrt{d}\boldsymbol{x})$ satisfies Assumption 8.19 with $M \le d^{O(\ell)}$ and $\delta = 1/M$.*

Proof. We will use a simple, auxiliary lemma.

Lemma 8.27. *Let X be a random variable such that $\mathbb{E}[X] = 0$, $\operatorname{Var}[X] = 1$, and $X \in [-R, R]$ almost surely. Then, $\Pr[X \ge 1/R] > 0$.*

It is clear that $\bar{m}(\mathbf{0}) = \mathbf{0}$. The bound on M is also immediate. We thus focus on proving that $\delta := 1/M$ suffices.

For the sake of contradiction, suppose that $\operatorname{conv} \bar{m}(\mathcal{B}_1(\mathbf{0}))$ does not contain $\boldsymbol{x}'$ for some $\boldsymbol{x}' \in \mathbb{R}^{k'}$ with $\|\boldsymbol{x}'\| \le \delta$. Then, we consider a hyperplane that separates $\operatorname{conv} \bar{m}(\mathcal{B}_1(\mathbf{0}))$ from $\boldsymbol{x}'$, and we let $\boldsymbol{v}$ be the normal vector to that hyperplane, so that $\langle \boldsymbol{v}, \boldsymbol{x}' \rangle > \langle \boldsymbol{v}, \bar{m}(\boldsymbol{x}) \rangle$ for all $\boldsymbol{x} \in \mathcal{B}_1(\mathbf{0})$. Now, let $\mathcal{U}$ be the uniform product distribution over $[-1, 1]^d$. By (8.4), we have $\mathbb{E}_{\boldsymbol{x} \sim \mathcal{U}}[m(\boldsymbol{x})] = 0$ and $\mathbb{E}_{\boldsymbol{x} \sim \mathcal{U}}[m(\boldsymbol{x}) m(\boldsymbol{x})^\top] = \mathbf{I}_{k' \times k'}$ (by ortogonality). As a result, we have $\mathbb{E}_{\boldsymbol{x} \sim \mathcal{U}}[\langle \boldsymbol{v}, m(\boldsymbol{x}) \rangle^2] = \mathbb{V}_{\boldsymbol{x} \sim \mathcal{U}}[\langle \boldsymbol{v}, m(\boldsymbol{x}) \rangle] = \|\boldsymbol{v}\|^2 = 1$. Lemma 8.27, applied for the random variable $\langle \boldsymbol{v}, m(\boldsymbol{x}) \rangle$ with range $[-M, M]$, implies that there exists $\boldsymbol{x} \in [-1, 1]^d$ such that $\langle \boldsymbol{v}, m(\boldsymbol{x}) \rangle \ge 1/M$, which in turn implies that there exists $\bar{\boldsymbol{x}} \in \mathcal{B}_1(\mathbf{0})$—namely, $\bar{\boldsymbol{x}} := \boldsymbol{x}/\sqrt{d}$—such that $\langle \boldsymbol{v}, \bar{m}(\bar{\boldsymbol{x}}) \rangle \ge 1/M$. But this yields $\delta \le \langle \boldsymbol{v}, \bar{m}(\bar{\boldsymbol{x}}) \rangle < \langle \boldsymbol{v}, \boldsymbol{x}' \rangle \le \|\boldsymbol{v}\| \|\boldsymbol{x}'\| = \delta$, a contradiction. □

It will thus follow from Theorem 8.2 that the regret against the set of degree-ℓ polynomials can be minimized in time $d^{O(\ell)} \mathsf{polylog}(1/\epsilon)$.

8.4.1 Behavioral vs Mixed Strategies

The above formulation allows the functions $\phi : \mathcal{X} \to \mathcal{X}$ to be arbitrary. A possible alternative is to consider $\phi : \Pi \to \mathcal{X}$ instead, and force the learner to play a distribution $\mu^t \in \Delta(\Pi)$. Here, we argue that the two are, in fact, equivalent, that is, a regret minimizer that plays distributions on $\mathcal{X}$ can be converted to one that plays only distributions on Π. Let $\Phi \subseteq \mathcal{X}^\Pi$ be a set of deviations. Given some $\phi : \Pi \to \mathcal{X}$, define $\phi^\beta(\boldsymbol{x}) := \mathbb{E}_{\boldsymbol{x}' \sim \beta(\boldsymbol{x})}[\phi(\boldsymbol{x}')]$, where $\beta : \mathcal{X} \to \Delta(\Pi)$ is a function that is *consistent* in the sense that $\mathbb{E}_{\boldsymbol{x}' \sim \beta(\boldsymbol{x})}[\boldsymbol{x}'] = \boldsymbol{x}$. Let $\Phi^\beta = \{\phi^\beta : \phi \in \Phi\} \subset \mathcal{X}^\mathcal{X}$. We construct a Φ-regret minimizer $\hat{\mathcal{R}}$ that plays distributions on Π from a Φ^β-regret minimizer $\mathcal{R}$ that plays distributions on $\mathcal{X}$, as follows: when $\mathcal{R}$ plays $\mu^t \in \Delta(\mathcal{X})$, $\hat{\mathcal{R}}$ plays the pushforward $\beta(\mu)$. Therefore whether we define our functions ϕ as functions $\Pi \to \mathcal{X}$ or $\mathcal{X} to \mathcal{X}$ is immaterial for the results of this section.

We now give two methods of constructing consistent and efficient maps $\beta : \mathcal{X} \to \Delta(\Pi)$ for tree-form strategy sets $\mathcal{X}$. The first is the behavioral strategy map, for extensive-form games.

Definition 8.28. When $\mathcal{X}$ is a sequence-form strategy set, the *behavioral strategy map* $\beta : \mathcal{X} \to \Delta(\Pi)$ is defined as follows: $\beta(\boldsymbol{x})$ is the distribution of pure strategies generated by sampling, at each decision point j for which $\boldsymbol{x}(j) > 0$, an action a according to the probabilities $\boldsymbol{x}(ja)/\boldsymbol{x}(j)$.

Formally,

$$\beta(\boldsymbol{x})[\boldsymbol{y}] := \prod_{ja:\boldsymbol{x}(j)>0, \boldsymbol{y}(ja)=1} \frac{\boldsymbol{x}(ja)}{\boldsymbol{x}(j)}.$$

It is possible for ϕ^β to be not a polynomial even when ϕ is a polynomial, because β is *itself* not a polynomial. It is clear that β is consistent. For efficiency, we show the following claim.

Proposition 8.29. *Let $\beta : \mathcal{X} \to \Delta(\Pi)$ be the behavioral strategy map. Let $\phi : \mathcal{X} \to \mathcal{X}$ be expressed as a polynomial of degree at most ℓ, in particular, as a sum of at most $O(d^\ell)$ terms. Then there is an algorithm running in time $d^{O(\ell)}$ that, given ϕ and $\boldsymbol{x} \in \mathcal{X}$, computes $\phi^\beta(\boldsymbol{x})$.*

Proof. To compute $\mathbb{E}_{\boldsymbol{x}' \sim \beta(\boldsymbol{x})} \phi(\boldsymbol{x}')$, since ϕ is a polynomial, it suffices to compute $\mathbb{E}_{\boldsymbol{x}' \sim \beta(\boldsymbol{x})} m(\boldsymbol{x}')$ for multilinear monomials m of degree at most ℓ, that is, functions of the form $m_S(\boldsymbol{x}) := \prod_{z \in S} \boldsymbol{x}(z)$ where $S \subseteq \mathcal{Z}$ has size at most ℓ. There are two cases. First, there are monomials that are clearly identically zero: in particular, if there are two nodes $ja, ja' \preceq S$ for $a \neq a'$, then $m_S \equiv 0$ because a player cannot play two different actions at j. For monomials that are not identically zero, we have

$$\mathop{\mathbb{E}}_{\boldsymbol{x}' \sim \beta(\boldsymbol{x})} \prod_{ja \in S} \boldsymbol{x}(ja) = \prod_{ja \preceq S : \boldsymbol{x}(j)>0} \frac{\boldsymbol{x}(ja)}{\boldsymbol{x}(j)},$$

which is computable in time $O(\ell d)$. Thus, the overall time complexity is $O(kd^{\ell+1}) \leq d^{O(\ell)}$. □

The behavioral strategy map is in some sense the *canonical* strategy map: when one writes a tree-form strategy $\boldsymbol{x} \in \mathcal{X}$ without further elaboration on what distribution $\Delta(\Pi)$ it is meant to represent, it is often implicitly or explicitly assumed to mean the behavioral strategy.

The behavioral strategy map has the unfortunate property that it usually outputs distributions of exponentially-large support; indeed, if $\boldsymbol{x} \in \operatorname{relint} \mathcal{X}$ then $\beta(\boldsymbol{x})$ is *full*-support. The second example we propose, which we call a *Carathéodory map*, always outputs low-support distributions. In particular, for any $\boldsymbol{x} \in \mathcal{X}$, Carathéodory's theorem on convex hulls guarantees that $\boldsymbol{x}$ is a convex combination of $d + 1$. Grötschel et al. [137, Theorem 3.9] moreover showed that there exists an efficient algorithm for computing the appropriate convex combination. Thus, fixing some efficient algorithm for this computational problem, we define a *Carathéodory map* $\beta : \mathcal{X} \to \Delta(\Pi)$ to be the map given by this algorithm. Given such a mapping, computing $\phi^\beta(\boldsymbol{x})$ is easy: one simply writes $\boldsymbol{x} = \sum_i \alpha_i \boldsymbol{x}_i$ by computing $\beta(\boldsymbol{x})$, and returns $\phi^\beta(\boldsymbol{x}) = \sum_i \alpha_i \phi(\boldsymbol{x}_i)$. This only requires a poly(d)-time computation of β, and $d + 1$ evaluations of the function ϕ. As before, when ϕ is a degree-ℓ polynomial, the time complexity of computing ϕ^β is bounded by $d^{O(\ell)}$.

8.5 Polynomial-Time Expected Fixed Points and Semi-Separation

We will now start connecting the framework we laid out in Section 8.2.3 with the problem of computing Φ-equilibria. That a Φ-equilibrium can be cast as (8.2)—by linking $\mathcal{Y}$ to the set of deviations Φ—is not hard to see, and will be spelled out in the next section. This section concerns the question of implementing ϵ-ExpectedGERorSEP (per Definition 8.17), which is the main precondition of Theorem 8.16.

The key to implementing the ϵ-ExpectedGERorSEP oracle, and the main subject of this section, is the notion of an *ϵ-expected fixed point (EFP)* (Definition 8.30). This is crucial because, unlike fixed points which are intractable beyond linear maps, there is a simple, $O(1/\epsilon)$-time algorithm for computing ϵ-expected fixed points. When it comes to computing Φ-equilibria in games, our contribution here is twofold.

1. we give a $\mathsf{poly}(d, \log(1/\epsilon))$-time algorithm for computing an ϵ-expected fixed point, and
2. we show that expected fixed points can be naturally coupled with the EAH framework, and in particular, with the recent generalization of Daskalakis et al. [81].

This section establishes Item 1, while the next section formalizes Item 2. Going back to Section 8.2.3 and the ExpectedGERorSEP oracle, the connection with (expected) fixed points lies in the observation that, when it comes to Φ-equilibria in games, the GER part of the oracle can be implemented by computing an expected fixed point. This will become clear in the upcoming section.

Definition 8.30 (Expected fixed points)**.** Let $\mathcal{X} \subseteq \mathbb{R}^d$ be convex and compact and a function $\phi : \mathcal{X} \to \mathcal{X}$ to which we are given oracle access. The *ϵ-expected fixed point (EFP)* problem asks for a distribution $\mu \in \Delta(\mathcal{X})$ such that[8.5]

$$\left\| \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mu} [\phi(\boldsymbol{x}) - \boldsymbol{x}] \right\|_1 \le \epsilon.$$

This definition departs from the usual notion of a fixed point by measuring the fixed-point error *in expectation* over samples $\boldsymbol{x}$ from μ. Definition 8.30 is natural in its own right, but our key motivation is computational: our main computational result regarding EFPs is a polynomial-time algorithm based on EAH.

Theorem 8.31. *There exists a* $\mathsf{poly}(d, \log(1/\epsilon))$*-time algorithm that, given oracle access to a set $\mathcal{X}$ and a map $\phi : \mathcal{X} \to \mathcal{X}$, that computes an ϵ-EFP of ϕ.*

Proof. We observe that an EFP can be equivalently expressed through the optimization problem

$$\text{find} \quad \mu \in \Delta(\mathcal{X}) \quad \text{s.t.} \quad \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mu} \langle \boldsymbol{y}, \phi(\boldsymbol{x}) - \boldsymbol{x} \rangle \ge 0 \quad \forall \boldsymbol{y} \in [-1, 1]^d.$$

We will now apply Theorem 8.16. The set $\mathcal{Y} := [-1, 1]^d$ clearly admits a separation oracle

[8.5] The choice of the 1-norm (instead of, say, another p-norm) here is unimportant, because one can always take ϵ to be exponentially small.

(SEP). Further, for any $\boldsymbol{y} \in [-1, 1]^d$, taking $\boldsymbol{x}^* = \text{argmin}_{\boldsymbol{x} \in \mathcal{X}} \langle \boldsymbol{y}, \boldsymbol{x} \rangle$ (using an optimization oracle for $\mathcal{X}$) guarantees $\langle \boldsymbol{y}, \phi(\boldsymbol{x}^*) - \boldsymbol{x}^* \rangle \geq 0$ since $\phi(\boldsymbol{x}^*) \in \mathcal{X}$, thereby implementing the GER oracle. We thus find that the preconditions of Theorem 8.16 are satisfied, and ExpectedEAH returns $\mu \in \Delta(\mathcal{X})$, with $\text{supp}(\mu) \leq \text{poly}(d, \log(1/\epsilon))$, such that

$$\mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mu} \langle \boldsymbol{y}, \phi(\boldsymbol{x}) - \boldsymbol{x} \rangle \geq -\epsilon \quad \forall \boldsymbol{y} \in [-1, 1]^d.$$

Taking $\boldsymbol{y} = \text{sgn}(\mathbb{E}_{\boldsymbol{x} \sim \mu}(\boldsymbol{x} - \phi(\boldsymbol{x})))$ (coordinate-wise) completes the proof. □

As it will become clear, Theorem 8.31 yields a polynomial-time implementation of the GER oracle in the context of Section 8.2.3, which can be employed in EAH. With a slight modification in the proof of Theorem 8.31, we shall see how one can also recover an ϵ-ExpectedGERorSEP oracle (Definition 8.17), which will then enable us to harness Theorem 8.18 for computing Φ-equilibria in games. Following the nomenclature of Daskalakis et al. [81], we refer to this oracle as a *semi-separation oracle*.

Definition 8.32 (Semi-separation oracle)**.** The *semi-separation* problem is the following. Given a convex and compact $\mathcal{X}$ and a function $\phi : \mathcal{X} \to \mathbb{R}^d$, compute

1. *either* a distribution $\mu \in \Delta(\mathcal{X})$ such that $\| \mathbb{E}_{\boldsymbol{x} \sim \mu}[\phi(\boldsymbol{x}) - \boldsymbol{x}] \|_1 \leq \epsilon$,
2. *or* a point $\boldsymbol{x} \in \mathcal{X}$ with $\phi(\boldsymbol{x}) \notin \mathcal{X}$.

Unlike Definition 8.30, here we allow ϕ to map outside of $\mathcal{X}$. This more general framing is essential to arrive at the ExpectedGERorSEP oracle. In particular, we note that Item 2 yields a hyperplane separating ϕ from the set of endomorphisms on $\mathcal{X}$. Namely, since $\phi(\boldsymbol{x}) \notin \mathcal{X}$, we can use the separation oracle on $\mathcal{X}$ to separate $\mathcal{X}$ from $\phi(\boldsymbol{x})$; that is, there is a $\boldsymbol{w}$ such that $\langle \phi(\boldsymbol{x}), \boldsymbol{w} \rangle > \langle \boldsymbol{x}, \boldsymbol{w} \rangle$ for all $\boldsymbol{x} \in \mathcal{X}$. But this also implies that $\langle \phi(\boldsymbol{x}), \boldsymbol{w} \rangle > \langle \phi'(\boldsymbol{x}), \boldsymbol{w} \rangle$ for any endomorphism ϕ', as promised.

Theorem 8.33. *Given oracle access to $\mathcal{X}$ and ϕ, there is a* $\text{poly}(d, \log(1/\epsilon))$*-time algorithm for implementing the semi-separation oracle of Definition 8.32.*

Proof. As in the proof of Theorem 8.31, we proceed by running the ellipsoid algorithm (per EAH) on the problem

$$\text{find} \quad \boldsymbol{y} \in [-1, 1]^d \quad \text{s.t.} \quad \langle \boldsymbol{y}, \phi(\boldsymbol{x}) - \boldsymbol{x} \rangle \leq -\epsilon \quad \forall \boldsymbol{x} \in \mathcal{X}. \tag{8.6}$$

For any $\boldsymbol{y} \in [-1, 1]^d$ during the execution of the ellipsoid, take $\boldsymbol{x}^*(\boldsymbol{y}) \in \text{argmin}_{\boldsymbol{x} \in \mathcal{X}} \langle \boldsymbol{y}, \boldsymbol{x} \rangle$. If $\phi(\boldsymbol{x}^*(\boldsymbol{y})) \notin \mathcal{X}$, the algorithm can terminate and return $\boldsymbol{x}^*(\boldsymbol{y})$. Otherwise, it follows that $\langle \boldsymbol{y}, \phi(\boldsymbol{x}^*(\boldsymbol{y})) - \boldsymbol{x}^*(\boldsymbol{y}) \rangle \geq 0$, by definition of $\boldsymbol{x}^*$, and so we can use $\boldsymbol{x}^*$ to get a separation oracle for (8.6).

Now, if every $\boldsymbol{x}^*(\boldsymbol{y}) \in \mathcal{X}$ generated above satisfies the constraint $\phi(\boldsymbol{x}^*(\boldsymbol{y})) \in \mathcal{X}$, then EAH returns a certificate of infeasibility for (8.6) in $\text{poly}(d, \log(1/\epsilon))$ time, which is an ϵ-expected fixed point of ϕ. On the other hand, if at some point there is $\boldsymbol{y} \in [-1, 1]^d$ such that $\phi(\boldsymbol{x}^*(\boldsymbol{y})) \notin$

$\mathcal{X}$, then the algorithm returns a point $\boldsymbol{x}^*(\boldsymbol{y}) \in \mathcal{X}$ such that $\phi(\boldsymbol{x}^*(\boldsymbol{y})) \notin \mathcal{X}$. This completes the proof. □

This semi-separation oracle amounts to the ϵ-ExpectedGERorSEP oracle needed in Theorem 8.18, as we shall see next in the context of games. Compared to the semi-separation oracle of Daskalakis et al. [81] that only works for linear functions, ours (Theorem 8.33) places no restrictions on ϕ.

8.6 A Polynomial-Time Algorithm for Φ^m-Equilibria in Games

Armed with the powerful semi-separation oracle of Theorem 8.33, we now establish a polynomial-time algorithm for computing Φ^m-equilibria in general multilinear games (Theorem 8.34).

Let us recall the basic setting of an n-player multilinear game Γ. Each player $i \in [n]$ has a convex and compact strategy set $\mathcal{X}_i \subseteq \mathbb{R}^{d_i}$ in isotropic position (Section 8.2). Player i has a utility function $u_i : \mathcal{X}_1 \times \cdots \times \mathcal{X}_n \to \mathbb{R}$ that is linear in $\mathcal{X}_i$, so that $u_i(\boldsymbol{x}) = \langle \boldsymbol{g}_i, \boldsymbol{x}_i \rangle$ for some $\boldsymbol{g}_i = \boldsymbol{g}_i(\boldsymbol{x}_{-i}) \in \mathbb{R}^{d_i}$. Furthermore, for each player $i \in [n]$, we let $\Phi^{m_i} \subseteq \mathcal{X}_i^{\mathcal{X}_i}$ be the k_i-dimensional set of deviations in the sense of Definition 8.1; that is, there exists a function $m_i \in \mathcal{X}_i \to \mathbb{R}^{k'_i}$, with $k_i = k'_i \cdot d_i + d_i$, such that for each $\phi_i \in \Phi^{m_i}$ and $\boldsymbol{x}_i \in \mathcal{X}_i$, the function output $\phi_i(\boldsymbol{x}_i)$ can be expressed as the matrix-vector product $\mathbf{K}_i(\phi_i) m_i(\boldsymbol{x}_i) + \boldsymbol{c}_i$ for some matrix $\mathbf{K}_i \in \mathbb{R}^{d_i \times k'_i}$ and $\boldsymbol{c}_i \in \mathbb{R}^{d_i}$. It is assumed throughout that Φ^{m_i} contains the identity map. For notational simplicity, we let $k := \sum_{i=1}^n k_i$ and $d := \sum_{i=1}^n d_i$.

In this context, we next state the main result of this section, and proceed with its proof.

Theorem 8.34 (Precise version of Theorem 8.2)**.** *Consider an n-player multilinear game Γ such that, for each player $i \in [n]$, we are given* $\mathsf{poly}(n, k)$*-time algorithms for the following:*

- *an oracle to compute the gradient, that is, the vector $\boldsymbol{g}_i = \boldsymbol{g}_i(\boldsymbol{x}_{-i}) \in \mathbb{R}^{d_i}$ for which $\langle \boldsymbol{g}_i(\boldsymbol{x}_{-i}), \boldsymbol{x}_i \rangle = u_i(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$ (polynomial expectation property); and*
- *a membership oracle for the strategy set $\mathcal{X}_i$, assumed to be in isotropic position.*

Suppose further that each k_i-dimensional set Φ^{m_i} satisfies Assumption 8.19 and $\|\boldsymbol{g}_i\| \leq B$. Then, an ϵ-approximate Φ^m-equilibrium of Γ can be computed in $\mathsf{poly}(n, k, \log(B/\epsilon))$ *time.*

Proof. An ϵ-approximate Φ^m-equilibrium of Γ is a distribution $\mu \in \Delta(\mathcal{X}_1 \times \cdots \times \mathcal{X}_n)$ such that

$$\mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mu} [u_i(\phi_i(\boldsymbol{x}_i), \boldsymbol{x}_{-i}) - u_i(\boldsymbol{x})] \leq \epsilon$$

for every player $i \in [n]$ and deviation $\phi_i \in \Phi^{m_i}$. Using multilinearity and Definition 8.1, it suffices to find a distribution $\mu \in \Delta(\mathcal{X}_1 \times \cdots \times \mathcal{X}_n)$ satisfying

$$\mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mu} \left[\sum_{i=1}^{n} \langle \boldsymbol{g}_i(\boldsymbol{x}_{-i}), \mathbf{K}_i m_i(\boldsymbol{x}_i) + \boldsymbol{c}_i - \boldsymbol{x}_i \rangle \right] \le \epsilon$$

for every $(\mathbf{K}_1(\phi_1), \dots, \mathbf{K}_n(\phi_n))$ and $(\boldsymbol{c}_1(\phi_1), \dots, \boldsymbol{c}_n(\phi_n))$, where $(\phi_1, \dots, \phi_n) \in \Phi^m$. (This derivation uses the fact that Φ^{m_i} contains the identity map.) We will now apply Theorem 8.18 with respect to $\mathbb{R}^d \supseteq \mathcal{X} := \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$ and

$$\mathbb{R}^k \supseteq \mathcal{Y} := \{(\mathbf{K}_1, \boldsymbol{c}_1, \dots, \mathbf{K}_n, \boldsymbol{c}_n) : \mathbf{K}_i m_i(\boldsymbol{x}_i) + \boldsymbol{c}_i \in \mathcal{X}_i \quad \forall \boldsymbol{x}_i \in \mathcal{X}_i\}.$$

By the polynomial expectation property, we can evaluate the term $\sum_{i=1}^{n} \langle \boldsymbol{g}_i(\boldsymbol{x}_{-i}), \mathbf{K}_i m_i(\boldsymbol{x}_i) + \boldsymbol{c}_i - \boldsymbol{x}_i \rangle$, for each $\boldsymbol{x} \in \mathcal{X}$, in $\mathsf{poly}(n, k)$ time. It thus suffices to show how to implement the ϵ-$\mathsf{ExpectedGERorSEP}$ oracle, which yields a separation oracle for the program

$$\begin{aligned} &\text{find} \quad \mathbf{K}_1, \boldsymbol{c}_1, \dots, \mathbf{K}_n, \boldsymbol{c}_n \quad \text{s.t.} \\ &\sum_{i=1}^{n} \langle \boldsymbol{g}_i(\boldsymbol{x}_{-i}), \mathbf{K}_i \boldsymbol{m}_i(\boldsymbol{x}_i) + \boldsymbol{c}_i - \boldsymbol{x}_i \rangle \ge -\epsilon \quad \forall \boldsymbol{x} \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_n, \\ &\mathbf{K}_i \boldsymbol{m}_i(\boldsymbol{x}_i) + \boldsymbol{c}_i \in \mathcal{X}_i \quad \forall \boldsymbol{x}_i \in \mathcal{X}_i. \end{aligned}$$

Consider any $\mathbb{R}^k \ni \phi = (\mathbf{K}_1, \boldsymbol{c}_1, \dots, \mathbf{K}_n, \boldsymbol{c}_n)$. We apply the semi-separation oracle of Theorem 8.33 for each function $\boldsymbol{x}_i \mapsto \mathbf{K}_i m_i(\boldsymbol{x}_i) + \boldsymbol{c}_i$. This returns *either* an ϵ'-expected fixed point, that is, a distribution $\nu_i \in \Delta(\mathcal{X}_i)$ such that

$$\left\| \mathop{\mathbb{E}}_{\boldsymbol{x}_i \sim \nu_i} [\mathbf{K}_i \boldsymbol{m}_i(\boldsymbol{x}_i) + \boldsymbol{c}_i - \boldsymbol{x}_i] \right\|_1 \le \epsilon',$$

or a point $\boldsymbol{x}_i \in \mathcal{X}_i$ such that $\mathbf{K}_i m_i(\boldsymbol{x}_i) + \boldsymbol{c}_i \notin \mathcal{X}_i$. If any of those semi-separation oracles returned $\boldsymbol{x}_i \in \mathcal{X}_i$ with $\mathbf{K}_i m_i(\boldsymbol{x}_i) + \boldsymbol{c}_i \notin \mathcal{X}_i$, we can use it to obtain a hyperplane separating $(\mathbf{K}, \boldsymbol{c})$ from the set of deviations $\mathcal{Y}$. Otherwise, let $\nu := \nu_1 \times \cdots \times \nu_n \in \Delta(\mathcal{X}_1) \times \cdots \times \Delta(\mathcal{X}_n)$ be the induced product distribution. Then, we have

$$\mathop{\mathbb{E}}_{\boldsymbol{x} \sim \nu} \sum_{i=1}^{n} \langle \boldsymbol{g}_i(\boldsymbol{x}_{-i}), \mathbf{K}_i \boldsymbol{m}_i(\boldsymbol{x}_i) + \boldsymbol{c}_i - \boldsymbol{x}_i \rangle = \sum_{i=1}^{n} \Big\langle \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \nu} \boldsymbol{g}_i(\boldsymbol{x}_{-i}), \mathop{\mathbb{E}}_{\boldsymbol{x}_i \sim \nu_i} [\mathbf{K}_i \boldsymbol{m}_i(\boldsymbol{x}_i) + \boldsymbol{c}_i - \boldsymbol{x}_i] \Big\rangle \le nB\epsilon', \tag{8.7}$$

where we used the fact that ν is a product distribution in the equality above. Thus, we have identified an $(\epsilon' nB)$-approximate good-enough-response, yielding an ϵ-$\mathsf{ExpectedGERorSEP}$ oracle by rescaling ϵ', and the proof follows from Theorem 8.18. □

It is worth stressing that it is crucial for our proof that the *expected* VI problem above corresponds to a game. It allows each player to be treated *independently*, which yields a *product distribution* $\nu = \nu_1 \times \cdots \times \nu_n$ when we apply the semi-separation oracle of Theorem 8.33 (for each player).

Algorithm 8.7 (NestedEAH): Polynomial-time algorithm for Φ^m-equilibria

```
input:
    an $n$-player multilinear game $\Gamma$
    a precision parameter $\epsilon > 0$
    a membership oracle for each $\mathcal{X}_i$
    an oracle for computing the gradient $\boldsymbol{g}_i = \boldsymbol{g}_i(\boldsymbol{x}_{-i}) \in \mathbb{R}^{d_i}$ for each $i \in [n]$
    a $k_i$-dimensional set $\Phi^{m_i}$ under Assumption 8.19 for each $i \in [n]$
output: an $\epsilon$-approximate $\Phi^m$-equilibrium of $\Gamma$ in $\mathrm{poly}(k, \log(1/\epsilon))$ time
define $G : \mathbb{R}^d \to \mathbb{R}^k$ such that $\langle G(\boldsymbol{x}), (\mathbf{K}, \boldsymbol{c}) \rangle = \sum_{i=1}^n \langle \boldsymbol{g}_i(\boldsymbol{x}_{-i}), \mathbf{K}_i m_i(\boldsymbol{x}_i) + \boldsymbol{c}_i - \boldsymbol{x}_i \rangle$
use the semi-separation oracle of Theorem 8.33 to construct an $\epsilon$-ExpectedGERorSEP oracle $\mathcal{O}$
apply ExpectedEAH with $\mathcal{O}$ as the $\epsilon$-ExpectedGERorSEP oracle
```

That ν is a product distribution is crucial to implement the separation oracle for the dual because it allows us to push the expectation into the inner product in (8.7), as we saw in the last step of the proof.

8.7 Algorithm for Minimizing Φ^m-Regret

We now switch gears to the online learning setting, recalled in Section 8.2.1.1. We will establish both upper and lower bounds on Φ-regret minimization. Our main positive result, Theorem 8.38, is an efficient online algorithm for minimizing Φ^m-regret with respect to any $\mathrm{poly}(d)$-dimensional set Φ^m (under Assumption 8.19), which applies even in the adversarial regime.

In what follows, we build on the framework of Daskalakis et al. [81], itself a refinement of the template of Gordon et al. [132]. As we have seen, Daskalakis et al. [81] showed that separating even over the set of linear endomorphisms is hard. In light of this, they proceed as follows. Instead of operating over the set of linear endomorphisms, their key idea is to consider a sequence of "shell sets," each of which contains the original set. Each shell set must also satisfy two basic properties:

- it is sufficiently structured so that it is possible to optimize over that set, and
- it contains a transformation with a fixed point inside $\mathcal{X}$.

Here, we show that by replacing fixed points with *expected* fixed points in the above template, it is possible to extend their main result to handle any $\mathrm{poly}(d)$-dimensional set under Assumption 8.19.

Overview. Our main construction is ShellRM. It is an instantiation of ShellGD (Section 8.7.2), which is projected gradient descent but with the twist that the constraint set is changing over time—reflecting the fact that a new shell set is computed at every round. To execute ShellGD, ShellProject (Section 8.7.3) provides an efficient projection oracle together with an approximate expected fixed point thereof, which is ultimately the output of our Φ^m-regret minimizer. ShellProject crucially relies on ShellEllipsoid, introduced next in Section 8.7.1. It strengthens our semi-separation oracle

Algorithm 8.8 (ShellEllipsoid): Shell ellipsoid algorithm

```
input:
    oracle access to convex set $\mathcal{X} \subseteq \mathbb{R}^d$
    oracle access to a $k$-dimensional convex set $\mathcal{F} \subseteq \mathcal{B}_D(\mathbf{0})$
    precision parameter $\epsilon > 0$
initialize $\mathcal{E} := \mathcal{B}_D(\mathbf{0})$ and $Q := \mathbb{R}^k$
while vol$(\mathcal{E}) \geq \epsilon$ do
    set $\phi \in Q \cap \mathcal{F}$ as the center of $\mathcal{E}$
    run the semi-separation oracle of Theorem 8.33 with respect to $\phi$
    if it returned an $\epsilon$-expected fixed point $\mu \in \Delta(\mathcal{X})$ of $\phi$ then return $\phi$
    else
        let $H$ be the halfspace that separates $\phi$ from $\Phi^m$
        Set $Q := Q \cap H$
    set $\mathcal{E}$ to be the minimum volume ellipsoid containing $Q \cap \mathcal{F}$
return $Q$
```

of Theorem 8.33 by again using expected fixed points. Section 8.7.4 combines those ingredients to arrive at our main result (Theorem 8.38).

8.7.1 Shell Ellipsoid

Continuing from our semi-separation oracle of Theorem 8.33, ShellEllipsoid takes a step further: it takes as input a convex set of transformations $\mathcal{F} \subseteq \mathcal{B}_D(\mathbf{0})$—for which we have efficient oracle access, unlike Φ^m—and returns *either* a function $\phi \in \mathcal{F}$ and an ϵ-expected fixed point thereof in $\Delta(\mathcal{X})$, *or* it provides a certificate—in the form of a polytope expressed as the intersection of a polynomial number of halfspaces—establishing that $\text{vol}(\mathcal{F} \cap \Phi^m) \approx 0$. ShellEllipsoid will be used later as part of the ShellProject algorithm so as to shrink the shell set.

Lemma 8.35. *For any k-dimensional convex set $\mathcal{F} \subseteq \mathcal{B}_D(\mathbf{0})$ with efficient oracle access and $\epsilon > 0$, ShellEllipsoid$(\mathcal{F})$ runs in time $\text{poly}(k, \log(1/\epsilon), \log D)$, and*

- *either it returns a transformation $\phi \in \mathcal{F}$ with an ϵ-expected fixed point in $\mathcal{X}$,*
- *or it returns a polytope $Q \subseteq \mathbb{R}^k$, specified as the intersection of at most $\text{poly}(k, \log(1/\epsilon), \log D)$ halfspaces, such that $\Phi^m \subseteq Q$ and $\text{vol}(Q \cap \mathcal{F}) < \epsilon$.*

Coupled with Theorem 8.33 pertaining to the semi-separation oracle, the proof of correctness of Lemma 8.35 is immediate. That Q can be expressed as the intersection of a polynomial number of halfspaces follows from the usual analysis of ellipsoid, as in Daskalakis et al. [81, Lemma 4.2].

8.7.2 Shell Gradient Descent

Instead of minimizing external regret with respect to the set Φ^m, which is hard even under linear endomorphisms [81], the overarching idea is to run (projected) gradient descent but with respect

Algorithm 8.9 (ShellGD): Shell gradient descent [81]

```
input: learning rate $\eta$, convex and compact sets $\tilde{\mathcal{Y}}^{(1)}, \dots, \tilde{\mathcal{Y}}^{(T)} \subseteq \mathcal{B}_D(\mathbf{0})$
initialize $\boldsymbol{y}^{(0)} \in \tilde{\mathcal{Y}}^{(1)}$ and $\boldsymbol{u}^{(0)} := \mathbf{0}$
for $t = 1, \dots, T$ do
    obtain efficient oracle access to $\tilde{\mathcal{Y}}^{(t)}$
    update $\boldsymbol{y}^{(t)} := \Pi_{\tilde{\mathcal{Y}}^{(t)}}(\boldsymbol{y}^{(t-1)} + \eta \boldsymbol{u}^{(t-1)})$
    output $\boldsymbol{y}^{(t)}$ as the next strategy and receive feedback $\boldsymbol{u}^{(t)} \in [-1, 1]^k$
```

to a sequence of changing shell sets, $(\tilde{\mathcal{Y}}^{(t)})_{t=1}^T$, of Φ^m (each of which contains Φ^m); this process is ShellGD. So long as $\Phi^m \subseteq \tilde{\mathcal{Y}}^{(t)}$, ShellGD indeed minimizes external regret with respect to deviations in Φ^m—of course, ShellGD is not a genuine regret minimizer for Φ^m in that it is allowed to output strategies not in Φ^m, but Lemma 8.36 below is in fact enough for the purpose of minimizing Φ^m-regret.

Lemma 8.36 ([81]). *Suppose that the sequence of shell sets $(\tilde{\mathcal{Y}}^{(t)})_{t=1}^T$ is such that $\Phi^m \subseteq \tilde{\mathcal{Y}}^{(t)} \subseteq \mathcal{B}_D(\mathbf{0})$ for all $t \in [T]$. For any sequence of utilities $\boldsymbol{u}^{(1)}, \dots, \boldsymbol{u}^{(T)} \in [-1, 1]^k$, ShellGD satisfies*

$$\max_{\boldsymbol{y}^* \in \Phi^m} \sum_{t=1}^T \langle \boldsymbol{y}^* - \boldsymbol{y}^{(t)}, \boldsymbol{u}^{(t)} \rangle \leq \frac{D^2}{2\eta} + \eta \sum_{t=1}^T \|\boldsymbol{u}^{(t)}\|^2.$$

8.7.3 Shell Projection

To implement ShellGD, we will make use of ShellProject, the algorithm that is the subject of this subsection. There are two main desiderata for the sequence of shell sets taken as input in ShellGD. First, each shell set must be structured or simple enough to allow projecting onto it—this is the whole rationale of expanding Φ^m through shell sets. But, of course, this is not enough, for one could just consider the entire space. The second crucial concern is that each transformation given by ShellGD needs to admit (approximate) expected fixed points, so as to use the framework of Gordon et al. [132] (Theorem 8.10) and minimize Φ^m-regret. Lemma 8.37 below, concerning ShellProject, shows how to accomplish that goal; its proof is similar to that of Daskalakis et al. [81, Theorem 4.4].

Lemma 8.37. *Let $\mathcal{X}$ be a convex and compact set such that $\mathcal{B}_r(\mathbf{0}) \subseteq \mathcal{X} \subseteq \mathcal{B}_R(\mathbf{0})$ and $\mathcal{M}$ be a convex set such that $\Phi^m \subseteq \mathcal{M} \subseteq \mathcal{B}_D(\mathbf{0})$. For any $\phi \in \mathcal{B}_D(\mathbf{0}) \subseteq \mathbb{R}^k$ and $\epsilon > 0$, ShellProject runs in time $\mathsf{poly}(k, 1/\epsilon, R/r, D)$ and returns*

1. *a shell set $\tilde{\Phi}$ satisfying $\Phi^m \subseteq \tilde{\Phi}$, expressed by intersecting $\mathcal{M}$ with at most $\mathsf{poly}(d, k, 1/\epsilon, R/r, D)$ halfspaces, and*
2. *a transformation $\tilde{\phi} \in \tilde{\Phi}$ such that $\|\tilde{\phi} - \Pi_{\tilde{\Phi}}(\phi)\| \leq \epsilon$, together with an ϵ-expected fixed point of $\tilde{\phi}$, $\mu \in \Delta(\mathcal{X})$.*

Algorithm 8.10 (ShellProject): Project ϕ to a shell of Φ

```
input:
    convex and compact set $\mathcal{X} \subseteq \mathbb{R}^d$ such that $\mathcal{B}_r(\mathbf{0}) \subseteq \mathcal{X} \subseteq \mathcal{B}_R(\mathbf{0})$
    convex set $\mathcal{M}$ such that $\Phi^m \subseteq \mathcal{M} \subseteq \mathcal{B}_D(\mathbf{0})$
    transformation $\phi \in \mathcal{B}_D(\mathbf{0})$
    precision parameter $\epsilon > 0$
output:
    convex set $\tilde{\Phi}$ such that $\Phi^m \subseteq \tilde{\Phi} \subseteq \mathcal{M}$
    transformation $\tilde{\phi} \in \tilde{\Phi}$ such that $\|\tilde{\phi} - \Pi_{\tilde{\Phi}}(\phi)\| \le \epsilon$
    an $\epsilon$-expected fixed point $\mu \in \Delta(\mathcal{X})$ of $\tilde{\phi}$
set $\epsilon' = \frac{\epsilon r}{32M(R)D^2}$
initialize $\tilde{\Phi} := \mathcal{M}$
for $q = 0, \dots$ incremented by $\delta := \epsilon/4D$ do
    run ShellEllipsoid($\tilde{\Phi} \cap \mathcal{B}_q(\phi)$) with precision vol($\mathcal{B}_{\epsilon'}(\cdot)$)
    if it finds $\tilde{\phi}$ with an $\epsilon$-expected fixed point $\mu \in \Delta(\mathcal{X})$ then return $\tilde{\Phi}, \tilde{\phi}, \mu$
    else
        set $Q$ be the polytope returned by ShellEllipsoid
        set $\tilde{\Phi} := \tilde{\Phi} \cap Q$
```

8.7.4 Putting Everything Together

We now combine all the previous pieces to obtain an efficient algorithm for minimizing Φ^m-regret—when Φ^m is poly(d)-dimensional—under a general convex and compact set $\mathcal{X}$. The overall construction is depicted in Algorithm 8.11. In effect, it runs ShellGD with respect to the sequence of shell sets $(\tilde{\Phi}^{(t)})_{t=1}^T$. Indeed, by the correctness guarantee of ShellProject (Item 1 of Lemma 8.37), we have the invariance $\Phi(\mathcal{X}) \subseteq \tilde{\Phi}^{(t)}$ for all $t \in [T]$. Furthermore, Item 2 of Lemma 8.37 implies that $(\mathbf{K}^{(t+1)}, \boldsymbol{c}^{(t+1)}) \in \tilde{\Phi}^{(t+1)}$, returned by ShellProject in ShellRM, is within distance ϵ of the projection prescribed by ShellGD. As a result, we can apply Lemma 8.36 (up to some some slackness proportional to ϵ) to bound the external regret $\text{Reg}_{\Phi^m}^{(T)}$ of $((\mathbf{K}^{(t)}, \boldsymbol{c}^{(t)}))_{t=1}^T$ with respect to comparators from Φ^m; combined with the fact that $\mu^{(t)} \in \Delta(\mathcal{X})$ is an ϵ-expected fixed point of the function $\boldsymbol{x} \mapsto \mathbf{K}^{(t)} m(\boldsymbol{x}) + \boldsymbol{c}^{(t)}$ (as promised by Item 2), it follows that the Φ^m-regret of the learner (ShellRM) can be bounded by $\text{Reg}_{\Phi^m}^{(T)} + \epsilon T$ (as in Theorem 8.10). We thus arrive at our main result.

Theorem 8.38 (Precise version of Theorem 8.3)**.** *Let $\mathcal{X} \subseteq \mathbb{R}^d$ be a convex and compact set in isotropic position for which we have a membership oracle.* ShellRM *has per-round running time of* poly(k, T) *and guarantees average Φ^m-regret of at most* poly$(k)/\sqrt{T}$*, where k is the dimension of Φ^m under Assumption 8.19.*

Unlike the algorithm of Daskalakis et al. [81], a salient aspect of ShellRM is that it outputs a sequence of *mixed* strategies in $\Delta(\mathcal{X})$. As we saw earlier in Section 8.2.1.1, this turns out to be necessary: as we will see later, a learner restricted to output strategies in $\mathcal{X}$ cannot efficiently

Algorithm 8.11 (ShellRM): Φ^m-regret minimizer for convex strategy sets

```
input:
    convex and compact set $\mathcal{X} \subseteq \mathbb{R}^d$ in isotropic position
    $k$-dimensional set $\Phi^m$ under Assumption 8.19 with respect to $m : \mathcal{X} \to \mathbb{R}^{k'}$, where $k = k' \cdot d + d$
    time horizon $T \in \mathbb{N}$
set the learning rate $\eta \propto \frac{1}{\sqrt{T}}$ and $\epsilon = 1/\mathsf{poly}(k, T)$ to be sufficiently small
initialize $\mu^{(1)} \in \Delta(\mathcal{X})$ and $\mathbf{K}^{(1)} := \mathbf{I}_{d \times k'}$ to be the identity map and $\boldsymbol{c}^{(1)} := \mathbf{0}$
initialize $\mathcal{M} := \mathcal{B}_R(\mathbf{0})$ for a large enough $R \le \mathsf{poly}(k)$
for $t = 1, \dots, T$ do
    output $\mu^{(t)} \in \Delta(\mathcal{X})$ and receive feedback $\boldsymbol{u}^{(t)} \in [-1, 1]^d$
    define $\mathbb{R}^{d \times k' + d} \ni \mathbf{U}^{(t)} := (\mathbb{E}_{\boldsymbol{x}^{(t)} \sim \mu^{(t)}} \boldsymbol{u}^{(t)} \otimes m(\boldsymbol{x}^{(t)}), \boldsymbol{u}^{(t)})$
    set $\tilde{\Phi}^{(t+1)}, (\mathbf{K}^{(t+1)}, \boldsymbol{c}^{(t+1)}), \mu^{(t+1)} := \mathsf{ShellProject}_\Phi((\mathbf{K}^{(t)}, \boldsymbol{c}^{(t)}) + \eta \mathbf{U}^{(t)})$ with input $\mathcal{M}$ and precision $\epsilon$, where $\mu^{(t+1)} \in \Delta(\mathcal{X})$ is an $\epsilon$-expected fixed point of $\boldsymbol{x} \mapsto \mathbf{K}^{(t+1)} m(\boldsymbol{x}) + \boldsymbol{c}^{(t+1)}$
```

minimize Φ-regret even with respect to low-degree swap deviations (assuming PPAD ≠ P).

8.8 Special Case: Linear Swap Regret in Extensive-Form Games

In this section, we consider an interesting special case of the Φ-regret and Φ-equilibrium framework, namely, the case where $\Pi \subset \{0, 1\}^d$ is a sequence-form strategy set and $\Phi = \Phi_{\mathsf{LIN}}$ consists of the linear maps $\Pi \to \mathcal{X}$. We make two main contributions.

The first contribution is conceptual: we give, for extensive-form games, an *interpretation* of the set of linear deviations. More specifically, we will first introduce a set of deviations, which we will call the *untimed communication (UTC) deviations* that, a priori, seems very different from the set of linear deviations at least on a conceptual level. The deviation set, rather than being defined *algebraically* (linear functions), will be defined in terms of an interaction between a *deviator*, who wishes to evaluate the deviation function at a particular input, and a *mediator*, who answers queries about the input. We will show the following result, which is our first main theorem:

Theorem 8.39. *The untimed communication deviations are precisely the linear deviations.*

The mediator-based framework is more in line with other extensive-form deviation sets—indeed, all prior notions of regret for extensive form, to our knowledge, including all the notions discussed above, can be expressed in terms of the framework. As such, the above theorem places linear deviations firmly within the same framework usually used to study deviations in extensive form.

We will then demonstrate, perhaps surprisingly, that the set of UTC deviations is expressible in terms of DAG-form decision problems (from Chapter 4), opening up access to a wide range of extremely fast algorithms for regret minimization, both theoretically and practically, for UTC deviations and thus also for linear deviations. Our second main theorem is as follows.

Theorem 8.40 (Faster linear-swap regret minimization)**.** *There exists a regret minimizer with regret $O(d^2\sqrt{T})$ against all linear deviations, and whose per-iteration complexity is dominated by the complexity of computing a fixed point of a linear map $\phi^{(t)} : \mathcal{X} \to \mathcal{X}$.*

In particular, using the algorithm of Cohen et al. [65] to solve the linear program of finding a fixed point, our per-iteration complexity is $\tilde{O}(d^\omega)$, where $\omega \approx 2.37$ is the current matrix multiplication constant and $\tilde{O}$ hides logarithmic factors. We elaborate on the fixed-point computation in Section 8.8.3. This improves substantially on the result of Farina and Pipis [95], which has the same regret bound but whose per-iteration computation involved a *quadratic* program (namely, an ℓ_2 projection), which has higher complexity than a linear program (they give a bound of $\tilde{O}(d^{10})$). Finally, we demonstrate via experiments that our method is also empirically faster than the prior method.

8.8.1 Mediators and UTC Deviations

For extensive-form games, linear-swap regret was recently studied in detail by Farina and Pipis [95]: they provide a characterization of the set Φ_{LIN} when $\mathcal{X}$ is a sequence-form polytope, and thus derive an algorithm for minimizing Φ_{LIN}-regret over $\mathcal{X}$. Their paper is the starting point of ours.

With the notable exception the deviations Φ^m we have studied so far, most sets of deviations Φ for extensive-form games are defined by interactions between a *mediator* who holds a strategy $\boldsymbol{x} \in \Pi$, and a *deviator*, who should compute the function $\phi(\boldsymbol{x})$ by making queries to the mediator. The set of deviations is then defined by what queries that the player is allowed to make. Before continuing, we will first formulate the sets Φ mentioned in Section 8.2.1.2 in this paradigm, for intuition. For a given decision point j, call an action $a \in A_j$ the *recommended action at* j, denoted $a(\boldsymbol{x}, j)$, if $\boldsymbol{x}(ja) = 1$. Since $\boldsymbol{x}$ is a sequence-form strategy, it is possible for a decision point to have no recommended action if its parent p_j is itself not recommended.

- Constant (NFCCE): The deviator cannot to make any queries to the mediator.
- Trigger (EFCE): The deviator, upon reaching a decision point j, learns the recommended action (if any) at j before selecting its own action.
- Communication: The deviator maintains a *state* with the mediator, which is a sequence σ, initially $\varnothing$. Upon reaching a decision point j, the deviator selects a decision point $j' \in C_\sigma$ (possibly $j' \neq j$) at which to query the mediator, the deviator observes the recommendation $a' = a(\boldsymbol{x}, j')$, then the deviator must pick an action $a \in A_j$. The state is updated to $j'a'$.
- Swap (NFCE): The deviator learns the whole strategy $\boldsymbol{x}$ before selecting its strategy.

An example of a communication deviation can be found in Section 8.8.2. Of these, the closest notion to ours is the notion of communication deviation, and that is the starting point of our construction. One critical property of communication deviations is that the mediator and deviator "share a clock": for every decision point reached, the deviator must make exactly one query to the mediator. As the name suggests, our set of *untimed* deviations results from removing this timing restriction, and therefore allowing the deviator to make *any number* (zero, one, or more than one) of queries to the mediator for every decision point reached. We formally define the decision problem faced by an untimed deviator as follows.

Definition 8.41. The *UTC decision problem* corresponding to a given tree-form decision problem is the DAG-form decision problem defined as follows. Nodes are identified with pairs $(s, \tilde{s})$ where $s, \tilde{s} \in \Sigma \cup \mathcal{J}$. s represents the state of the real decision problem, and $\tilde{s}$ represents the state of the mediator. The root is $(\varnothing, \varnothing) \in \Sigma \times \Sigma$.

1. $(\sigma, \tilde{\sigma}) \in \Sigma \times \Sigma$ is an observation point. The deviator observes the next decision point $j \in C_\sigma$, and the resulting decision point is $(j, \tilde{\sigma})$
2. $(j, \tilde{j}) \in \mathcal{J} \times \mathcal{J}$ is an observation point. The deviator observes the recommendation $a = a(\boldsymbol{x}, \tilde{j})$, and the resulting decision point is $(j, \tilde{j}a)$.
3. $(j, \tilde{\sigma}) \in \mathcal{J} \times \Sigma$ is a decision point. The deviator can choose to either play an action $a \in A_j$, or to query a decision point $\tilde{j} \in C_{\tilde{\sigma}}$. In the former case, the resulting observation point is $(ja, \tilde{\sigma})$ for $a \in A_j$; in the latter case, the resulting observation point is $(j, \tilde{j})$.

Any mixed strategy of the deviator in this decision problem defines a function $\phi : \Pi \to \mathcal{X}$, where $\phi(\boldsymbol{x})[\sigma]$ is the probability that an untimed deviator plays all the actions on the path to σ when the mediator recommends according to pure strategy $\boldsymbol{x}$. We thus define:

Definition 8.42. An *UTC deviation* is any function $\phi : \Pi \to \mathcal{X}$ induced by a mixed strategy of the deviator in the UTC decision problem.

Clearly, the set of UTC deviations is at least as large as the set of communication deviations, and at most as large as the set of swap deviations. In the next section, we will discuss how to represent UTC deviations, and show that UTC deviations coincide precisely with linear deviations.

The set of DAG-form strategies can be parameterized as follows. A DAG-form strategy is given by two matrices $(\mathbf{A}, \mathbf{B})$ where $\mathbf{A} \in \{0, 1\}^{\Sigma \times \Sigma}$ encodes the part corresponding to sequences $(\sigma, \tilde{\sigma})$, and $\mathbf{B} \in \{0, 1\}^{\mathcal{J} \times \mathcal{J}}$ encodes the part corresponding to decision points $(j, \tilde{j})$. $\mathbf{A}(\sigma, \tilde{\sigma}) = 1$ if the deviator plays all the actions on *some* path to observation point $(\sigma, \tilde{\sigma})$, and similarly $\mathbf{B}(j, \tilde{j}) = 1$ if the deviator plays all the actions on some path to observation node $(j, \tilde{j})$. Since the only possible way for two paths to end at the same observation point is for the deviator to have changed the order of actions and queries, for any given pure strategy of the deviator, at most one path can exist for both cases. Therefore, the set of mixed sequence-form deviations can be expressed using the following set of constraints:

$$\begin{aligned} \mathbf{A}(p_j, \tilde{\sigma}) + \mathbf{B}(j, p_{\tilde{\sigma}}) &= \sum_{a \in A_j} \mathbf{A}(ja, \tilde{\sigma}) + \sum_{\tilde{j} \in C_{\tilde{\sigma}}} \mathbf{B}(j, \tilde{j}) \qquad && \forall j \in \mathcal{J}, \tilde{\sigma} \in \Sigma \\ \mathbf{A}(\varnothing, \varnothing) &= 1 && \\ \mathbf{A}(\varnothing, \tilde{\sigma}) &= 0 && \forall \tilde{\sigma} \neq \varnothing \\ \mathbf{A}, \mathbf{B} &\geq 0 && \end{aligned} \tag{8.8}$$

where, in a slight abuse of notation, we define $\mathbf{B}(j, p_\varnothing) := 0$ for every $j \in \mathcal{J}$. Moreover, for any pair of matrices $(\mathbf{A}, \mathbf{B})$ satisfying the constraint system and therefore defining some deviation $\phi : \Pi \to \mathcal{X}$, it is easy to compute how ϕ acts on any $\boldsymbol{x} \in \Pi$: the probability that the deviator plays all the actions on the $\varnothing \to \sigma$ path is simply given by

$$\sum_{\tilde{\sigma} \in \Sigma} \boldsymbol{x}(\tilde{\sigma}) \mathbf{A}(\sigma, \tilde{\sigma}) = (\mathbf{A}\boldsymbol{x})[\sigma],$$

and therefore ϕ is nothing more than a matrix multiplication with $\mathbf{A}$, that is, $\phi(\boldsymbol{x}) = \mathbf{A}\boldsymbol{x}$. We have thus shown that every UTC deviation is linear, that is, $\Phi_{\mathsf{UTC}} \subseteq \Phi_{\mathsf{LIN}}$. In fact, the reverse inclusion holds too:

Theorem 8.43. *The UTC deviations are precisely the linear deviations. That is,* $\Phi_{UTC} = \Phi_{\mathsf{LIN}}$.

Proof. We start with a lemma.

Lemma 8.44. *Let $f : \mathcal{X} \to \mathbb{R}_{\geq 0}$ be a linear map, where $\mathcal{X}$ is a sequence-form strategy space. Then there exists a unique vector $\boldsymbol{c}$ such that:*

1. *$f(\boldsymbol{x}) = \boldsymbol{c}^\top \boldsymbol{x}$ for all $\boldsymbol{x} \in \mathcal{X}$,*
2. *$\boldsymbol{c}$ has all nonnegative entries, and*
3. *for every decision point I, there is at least one action a such that $\boldsymbol{c}(ja) = 0$.*

Proof. Let $f(\boldsymbol{x}) = \boldsymbol{c}^\top \boldsymbol{x}$, where $\boldsymbol{c}$ is currently arbitrary (*i.e.*, it may not satisfy (2) and (3)). Then, for each decision point j in bottom-up order, let $\boldsymbol{c}^*(j) := \min_a \boldsymbol{c}(ja)$. Subtract $\boldsymbol{c}^*(j)$ from $\boldsymbol{c}(ja)$ for every action a, and add $\boldsymbol{c}^*(j)$ to $\boldsymbol{c}(p_j)$. Since $\boldsymbol{x}$ satisfies the constraint $\boldsymbol{x}(p_j) = \sum_a x(ja)$, this does not change the validity of $\boldsymbol{c}$, and by the end of the algorithm, (2) and (3) will be satisfied except possibly that $\boldsymbol{c}(\varnothing) \geq 0$. To see that $\boldsymbol{c}(\varnothing) \geq 0$, let $\boldsymbol{x}$ be the pure strategy that plays the zeroing action a specified by (3) at every decision point. Then, by construction, $\boldsymbol{c}^\top \boldsymbol{x} = \boldsymbol{c}(\varnothing) \geq 0$. To see that $\boldsymbol{c}$ is unique, note that there was no choice at any step in the above process: the transformation performed at each decision point is the only way to satisfy conditions (2) and (3) without changing the linear map. □

Now let $\mathbf{A}$ represent a linear map $\mathcal{X} \to \mathcal{X}$, where the rows of $\mathbf{A}$ are represented according to the above lemma. That is, $\mathbf{A}$ has all nonnegative entries, and moreover for any $\tilde{j} \in \mathcal{J}$ and $\sigma \in \Sigma$, we have $\mathbf{A}(\sigma, \tilde{j}a) = 0$ for some action a. It remains only to show there exists a matrix $\mathbf{B}$ such that $(\mathbf{A}, \mathbf{B})$ satisfies all constraints in the constraint system (8.8).

We have $\mathbf{A}(\varnothing, \varnothing) = 1$ and $\mathbf{A}(\varnothing, \tilde{\sigma}) = 0$ for $\tilde{\sigma} \neq \varnothing$ follow from the fact that $(\mathbf{A}\boldsymbol{x})(\varnothing) = 1$ for all $\boldsymbol{x}$, that is $\mathbf{A}(\varnothing, \cdot) : \mathcal{X} \to [0, 1]$ is the identically-1 function, which by Lemma 8.44 has the above form. We are thus left with the main constraint,

$$\mathbf{A}(p_j, \tilde{\sigma}) + \mathbf{B}(j, p_{\tilde{\sigma}}) = \sum_{a \in A_j} \mathbf{A}(ja, \tilde{\sigma}) + \sum_{\tilde{j} \in C_{\tilde{\sigma}}} \mathbf{B}(j, \tilde{j}) \tag{8.9}$$

for every $(j, \tilde{\sigma}) \in \mathcal{J} \times \Sigma$.

Define $\mathbf{B}$ and another matrix $\tilde{\mathbf{B}} \in \mathbb{R}^{\mathcal{J}\times\Sigma}$ as follows:

$$\begin{aligned}\tilde{\mathbf{B}}(j,\tilde\sigma) &= \sum_{a\in A_j} \mathbf{A}(ja,\tilde\sigma) + \sum_{\tilde\jmath\in C_{\tilde\sigma}} \mathbf{B}(j,\tilde\jmath) - \mathbf{A}(p_j,\tilde\sigma)\\ \mathbf{B}(j,\tilde\jmath) &= \min_{a\in A_{\tilde\jmath}} \tilde{\mathbf{B}}(j,\tilde\jmath a)\end{aligned}$$

To see that $\mathbf{B}$ satisfies all the constraints (8.9), let $\boldsymbol{x}$ be any fully-mixed strategy, and I be any decision point. Then:

$$\begin{aligned}0 &= \sum_{a\in A_j} (\mathbf{A}\boldsymbol{x})(ja) - (\mathbf{A}\boldsymbol{x})(p_j)\\ &= \sum_{\tilde\sigma\in\Sigma} \boldsymbol{x}(\tilde\sigma)\left(\sum_{a\in A_j} \mathbf{A}(ja,\tilde\sigma) - \mathbf{A}(p_j,\tilde\sigma)\right)\\ &= \sum_{\tilde\sigma\in\Sigma} \boldsymbol{x}(\tilde\sigma)\left(\tilde{\mathbf{B}}(j,\tilde\sigma) - \sum_{\tilde\jmath\in C_{\tilde\sigma}} \mathbf{B}(j,\tilde\jmath)\right)\\ &= \sum_{\tilde\jmath}\left(\sum_{a\in A_{\tilde\jmath}} \boldsymbol{x}(\tilde\jmath a)\tilde{\mathbf{B}}(j,\tilde\jmath a) - \boldsymbol{x}(p_{\tilde\jmath})\mathbf{B}(j,\tilde\jmath)\right)\\ &\ge \sum_{\tilde\jmath} \mathbf{B}(j,\tilde\jmath)\left(\sum_{a\in A_{\tilde\jmath}} \boldsymbol{x}(\tilde\jmath a) - \boldsymbol{x}(p_{\tilde\jmath})\right) = 0\end{aligned}$$

Thus, the inequality must in fact be an equality, and since all its terms are nonnegative, we thus have $\tilde{\mathbf{B}}(j,\tilde\jmath a) = \mathbf{B}(j,\tilde\jmath)$ for all $a \in A_{\tilde\jmath}$, so the constraints (8.9) are satisfied by definition of $\tilde{\mathbf{B}}$.

To see that $\mathbf{B} \ge 0$, suppose not. Let $(j,\tilde\jmath)$ be a last (*i.e.*, farthest from the root, with respect to the ordering of the DAG) pair for which $\mathbf{B}(j,\tilde\jmath) < 0$. Then, for any action $\tilde a \in A_{\tilde\jmath}$, we have

$$\mathbf{A}(p_j,\tilde\jmath\tilde a) + \mathbf{B}(j,\tilde\jmath) = \sum_{a\in A_j} \mathbf{A}(ja,\tilde\jmath\tilde a) + \sum_{\tilde\jmath'\in C_{\tilde\jmath\tilde a}} \mathbf{B}(j,\tilde\jmath') \ge 0$$

where the inequality is because $(j,\tilde\jmath)$ is farthest from the root so all the terms on the right-hand side are nonnegative. Therefore, $\mathbf{A}(p_j,\tilde\jmath\tilde a) > 0$. But this should hold for every action $\tilde a$, contradicting the construction of $\mathbf{A}$, which includes the condition that there must exist a $\tilde a$ for which $\mathbf{A}(p_j,\tilde\jmath\tilde a) = 0$. □

Since the two sets are equivalent, in the remainder of the section, we will use the terms *UTC deviation* and *linear deviation* (similarly, *UTC regret* and *linear-swap regret*) interchangeably.

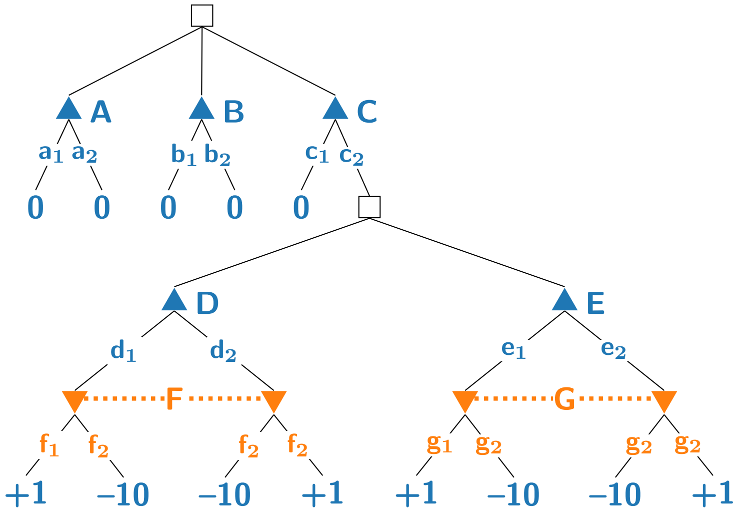


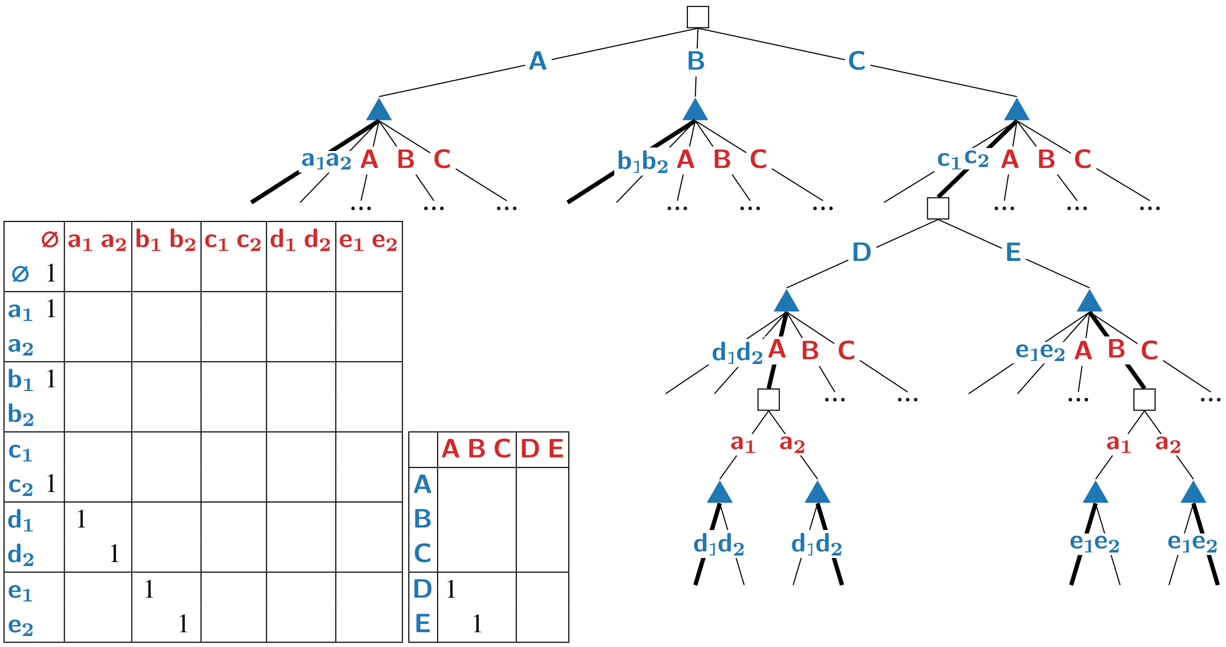
Figure 8.12: *An example extensive-form game in which communication deviations are a strict subset of UTC deviations. There are two players, P1 (▲) and P2 (▼). Infosets for both players are labeled with capital letters* (e.g., A) *and joined by dotted lines. Actions are labeled with lowercase letters and subscripts* (e.g., a_1). *P1's utility is labeled on each terminal node. P2's utility is zero everywhere (not labeled). Boxes are chance nodes, at which chance plays uniformly at random.*

8.8.2 Example

In this section, we provide two examples in which the UTC deviations are strictly more expressive than the communication deviations. Consider the game in Figure 8.12. The subgames rooted at D and E are guessing games, where ▲ must guess ▼'s action, with a large penalty for guessing wrong. Consider the correlated profile that mixes uniformly among the four pure profiles (a_i, b_j, c_1, f_i, g_j) for $i, j \in \{1, 2\}$. In this profile, the information that ▲ needs to guess perfectly is contained in the recommendations: the recommendation at A tells it how to guess at D, and the recommendation at B tells it how to guess at E. With a communication deviation, ▲ cannot access this information in a profitable way, since upon reaching C, ▲ must immediately make its first mediator query. Hence, this profile is a communication equilibrium. However, with an *untimed* communication deviation, ▲ can profit: it should, upon reaching[8.6] C, play action c_2 *without making a mediator query*, and then query A if it observes D, and B if it observes E. This deviation is allowed only due to the untimed nature of UTC deviations allows the deviating player to *delay* its query to the mediator until it reaches either D or E. In a *timed* communication deviation, this deviation is impossible, because the player must make its first query (A, B, or C) *before* reaching D or E, and thus that query cannot be conditioned on which one of D or E will be reached.

For a second example, consider the game in Figure 8.14. Consider the correlated profile that mixes uniformly between the pure profiles (a_1, b_1, c_1) and (a_1, b_2, c_2). This is a communication

[8.6] The actions/queries ▲ makes at A and B are irrelevant, because ▲ only cares about maximizing utility, and it always gets utility 0 regardless of what it does. In the depiction of this deviation in Figure 8.13, the deviator always plays action 1 at A and B.



	∅	a₁	a₂	b₁	b₂	c₁	c₂	d₁	d₂	e₁	e₂
∅	1										
a₁	1										
a₂											
b₁	1										
b₂											
c₁											
c₂	1										
d₁		1									
d₂			1								
e₁				1							
e₂					1						

	A	B	C	D	E
A					
B					
C					
D	1				
E		1			

Figure 8.13: *A part of the UTC decision problem for* ▲ *corresponding to the same game. Nodes labeled* ▲ *are decision points for* ▲*; boxes are observation points. "..." denotes that the part of the decision problem following that edge has been omitted. Terminal nodes are unmarked. Red edge labels indicate interactions with the mediator; blue edge labels indicate interactions with the game. The profitable untimed deviation discussed in Section 5.5 is indicated by the thick lines. The first action taken in that profiable deviation,* c_2*, is not legal for a timed deviator, because a timed deviator must query the mediator once before taking its first action. The matrices (lower-left corner) are the pair of matrices* $(\mathbf{A}, \mathbf{B})$ *corresponding to that same deviation. All blank entries are 0.*

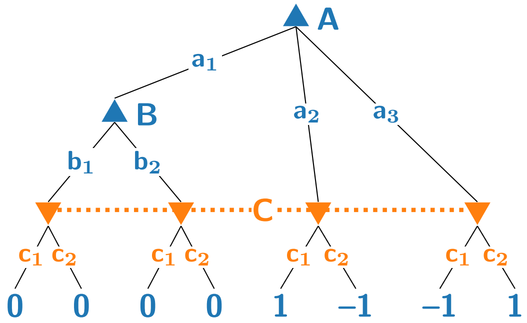


Figure 8.14: *Another example. The notation is shared with Figure 8.12. In this example, ▲'s strategy set is equivalent to a simplex, so the linear deviations coincide with its swap deviations. As such, we will not bother to depict the UTC decision problem or matrices.*

equilibrium: ▲ cannot profitably deviate, because its utility in the B subgame is always 0, and if it chooses to disobey the recommendation a_1 its expected utility will be also 0, because it cannot ask for another recommendation before choosing what action to play. However, ▲ has the following profitable UTC deviation: ask for the recommendation at B *before* deciding which action to play at A. If the recommendation is b_1, play a_2; if the recommendation is b_2, play a_3. Notice that, in this example, ▲'s decision problem is essentially that of a normal-form game; therefore, its linear deviations coincide with its swap deviations. However, due to the timing restriction on communication deviations, the communication deviations are more restricted than the swap deviations. This example also shows that the untimed private communication equilibria (see Section 8.8.4.6) are not outcome-equivalent to the timed private communication equilibria: in this game, every correlated profile is a distribution over terminal nodes (outcomes), so the fact that there exists a private communication equilibrium with a profitable UTC deviation is enough to disprove outcome equivalence.

8.8.3 Regret Minimization on Φ_{UTC}

In this section, we discuss how Theorem 8.43 can be used to construct very efficient Φ_{LIN}-regret minimizers, both in theory and in practice.

Theorem 8.45 (CFR for Φ_{LIN}, special case of Theorem 4.23)**.** DAG-CFR *can be applied to Φ_{UTC} (and thus also on Φ_{LIN}) with $O(d^2\sqrt{T})$ regret and $O(d^2)$ per-iteration complexity.*

Applying Theorem 8.10 now yields:

Theorem 8.46. *CFR-based algorithms can be used to construct a Φ_{LIN}-regret minimizer with $O(d^2\sqrt{T})$ regret, and per-iteration complexity dominated by the complexity of computing a fixed point of a linear transformation $\phi^{(t)} : \operatorname{conv} \mathcal{X} \to \operatorname{conv} \mathcal{X}$.*

As mentioned, this significantly improves the per-iteration complexity of linear-swap regret

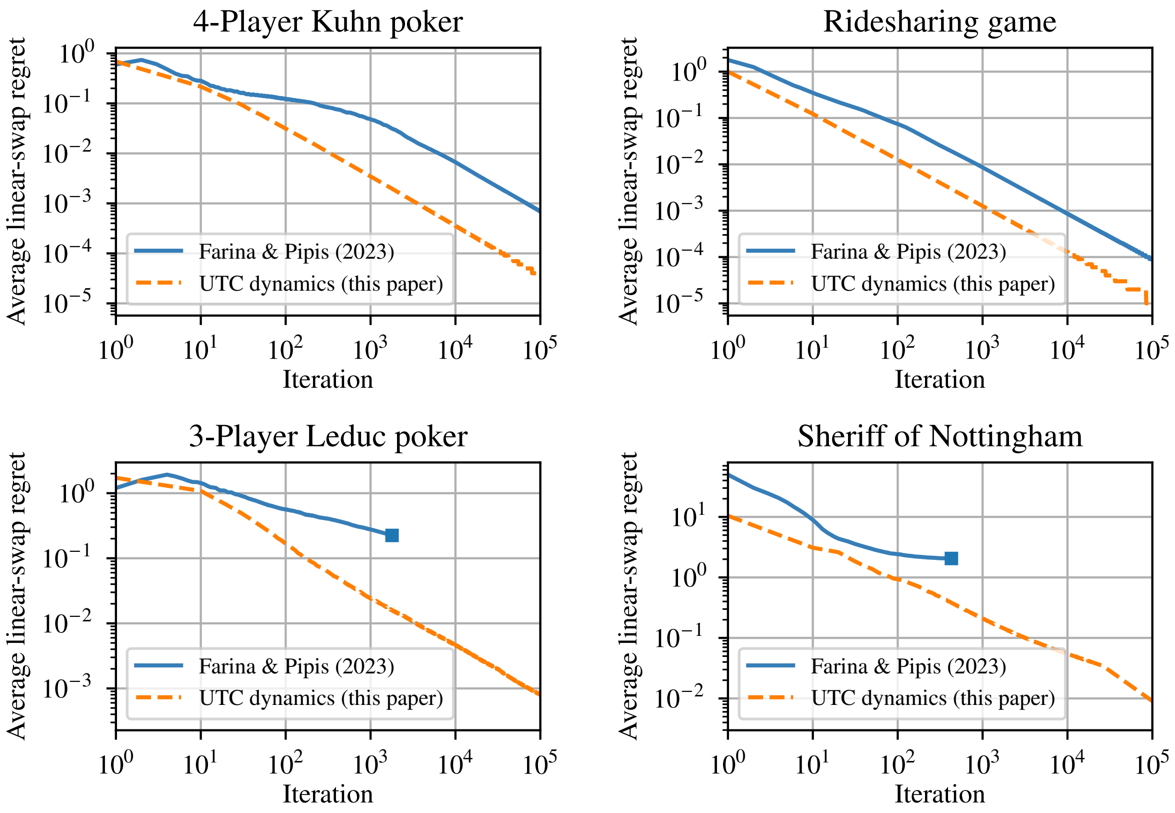


Figure 8.15: *Experimental comparison between our dynamics and those of Farina and Pipis [95] for approximating a linear correlated equilibrium in extensive-form games. Each algorithm was run for a maximum of* 100,000 *iterations or 6 hours, whichever was hit first. Runs that were terminated due to the time limit are marked with a square* ■.

minimization. Fixed points can be computed by finding a feasible solution to the constraint system $\{\boldsymbol{x} \in \mathcal{X}, \mathbf{A}\boldsymbol{x} = \boldsymbol{x}\}$, where $\boldsymbol{x} \in \mathcal{X}$ is expressed using the sequence-form constraints. This is a linear program with $O(d)$ variables and constraints, so the LP algorithm of Cohen et al. [65] yields a fixed-point computation algorithm with runtime $\tilde{O}(d^\omega)$.

For comparison, the algorithm of Farina and Pipis [95] requires an ℓ_2 projection onto $\mathcal{X}$ on every iteration, which requires solving a convex quadratic program; the authors of that paper derive a bound of $\tilde{O}(d^{10})$, which, although polynomial, is much slower than our algorithm. CFR-based algorithms are currently the fastest practical regret minimizers [38, 103]—therefore, showing that our method allows such algorithms to be applied is also a significant practical step. In Section 8.8.6, we will show empirically that the resulting algorithm is significantly better than the previously-known state of the art, in terms of both per-iteration time complexity and number of iterations.

8.8.4 Discussion

Here, we discuss a few points of interest about UTC and linear deviations.

8.8.4.1 The Convex Hull of Pure Deviations

In our definitions, we were careful to allow transformations $\phi \in \Phi$ to map from the set of pure strategies, Π, to its convex hull $\mathcal{X}$, instead of insisting that every pure strategy map onto another pure strategy. One might ask whether this makes a difference in our definitions. For example, if in (8.1) we restrict our attention to $\phi : \Pi \to \Pi$, does the definition change? In symbols, for a given set of transformations Φ, is $\Phi \subseteq \operatorname{conv} \hat{\Phi}$ where $\hat{\Phi} = \{\phi \in \Phi : \phi(\boldsymbol{x}) \in \Pi \ \forall \boldsymbol{x} \in \Pi\}$? For the other sets of deviations mentioned in this chapter (external, swap, trigger, and communication), the answer is already known to be positive.

Our equivalence theorem between UTC and linear deviations gives an answer to this question for the set of linear deviations as well. Since the UTC deviations are defined by a decision problem, every mixed UTC deviation is by definition equivalent to a distribution over pure UTC deviations. That is, the vertices of Φ_{UTC} are the pure UTC deviations: they map pure strategies Π to pure strategies. Since $\Phi_{\mathsf{UTC}} = \Phi_{\mathsf{LIN}}$, this proves:

Corollary 8.47. *When Π is a sequence-form polytope, the extreme points of Φ_{LIN} are the linear maps $\phi : \Pi \to \Pi$,* i.e.*, the linear maps that map pure strategies to pure strategies. Thus, $\Phi_{\mathsf{LIN}} = \hat{\Phi}_{\mathsf{LIN}}$*

This result is not obvious *a priori*. For example, it fails to generalize to other sets of functions Φ, or to Φ_{LIN} for polytopes Π that are not sequence-form polytopes:

- Other sets of functions Φ: Let $\Phi = \{\phi\}$ consist of a single constant function $\phi : \boldsymbol{x} \mapsto \boldsymbol{x}^*$, where $\boldsymbol{x}^* \in \mathcal{X} \setminus \Pi$. Then $\hat{\Phi}$ is empty, so $\Phi \not\subseteq \operatorname{conv} \hat{\Phi}$.
- Non-sequence-form polytopes: Take Π to be a trapezoid $ABCD$ where AB is the longer of the two bases and consider the linear map ϕ with $\phi(A) = D$, $\phi(D) = A$, and $\phi(B) = C$. Then ϕ is an extreme point of Φ_{LIN}, but $\phi(C)$ will lie somewhere along segment AB, but at neither endpoint—that is, not at a vertex. Figure 8.16 has a visual depiction.

8.8.4.2 Generalization to Arbitrary Pairs of Polytopes

Our main result characterizes the set of linear maps $\phi : \Pi \to \Pi$ for sequence-form polytopes Π. However, it is actually more general than this: an identical proof works to characterize the set of linear maps $\phi : \mathcal{Y} \to \mathcal{X}$ for any (possibly different!) sequence-form polytopes $\mathcal{X}$ and $\mathcal{Y}$. Hence, we have shown:

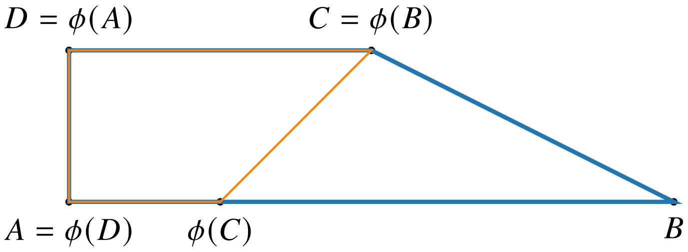


Figure 8.16: *A visual depiction of the argument that Corollary 8.47 cannot generalize to all polytopes. The affine map ϕ maps the large blue polygon onto the small orange polygon, and ϕ is a vertex of the set of linear maps from polygon $ABCD$ to itself, yet $\phi(C)$ is not a vertex of $ABCD$.*

Theorem 8.48. *Let $\Pi, \mathcal{Y}$ be sequence-form strategy sets. The linear maps $\phi : \mathcal{Y} \to \mathcal{X}$ are precisely the functions induced by strategies in the DAG decision problem whose nodes are identified with pairs $(s, \tilde{s})$, where $s \in \Sigma_{\mathcal{X}} \cup \mathcal{J}_{\mathcal{X}}$ and $\tilde{s} \in \Sigma_{\mathcal{Y}} \cup \mathcal{J}_{\mathcal{Y}}$, and which behaves analogously to Definition 8.41.*

Although we are mostly concerned with the case $\mathcal{X} = \mathcal{Y}$ in this paper, we state this extension in the hope that it may be of independent interest. We will also use it in the proof of the revelation principle (Theorem 8.52).

8.8.4.3 Uniqueness of Representation

The statement of Theorem 4.26 discusses Φ_{UTC} and Φ_{LIN} as *sets of functions* $\Phi \subseteq \mathcal{X}^{\Pi}$. It does *not* imply that for every linear map $\phi : \Pi \to \mathcal{X}$ there is *exactly* one representation of ϕ as a deviator strategy in the UTC decision problem, only that there is *at least* one representation. Indeed, the external deviations (constant functions $\phi : \boldsymbol{x} \mapsto \boldsymbol{x}^*$ for fixed $\boldsymbol{x}^* \in \mathcal{X}$ can be represented via a large number of different strategies in the UTC decision problem: the deviator may send any number of queries to the mediator, before eventually deciding to ignore the queries and play according to $\boldsymbol{x}^*$, and such a deviator would still represent the external deviation ϕ.

Similarly, Theorem 4.26 also does not imply that every matrix $\mathbf{A} \in \mathbb{R}^{\Sigma \times \Sigma}$ representing a linear map $\phi_{\mathbf{A}} : \Pi \to \mathcal{X}$ is part of a pair $(\mathbf{A}, \mathbf{B})$ satisfying the system of equations (8.8). Indeed, the proof of Theorem 4.26 only shows that for every linear $\phi : \Pi \to \mathcal{X}$, there is *at least one* pair $(\mathbf{A}, \mathbf{B})$ satisfying (8.8) where $\mathbf{A}$ represents ϕ. It is easy to construct matrices $\mathbf{A}$ that represent linear maps, yet cannot satisfy (8.8), by changing the first row of $\mathbf{A}$ to some other vector $\boldsymbol{c}$ with $\boldsymbol{c}^\top \boldsymbol{x} = 1$ for all $\boldsymbol{x} \in \Pi$.

8.8.4.4 When Untimed and Timed Communication Deviations Coincide

If all players have only one layer of decision nodes, the game is a single-stage *Bayesian game*—in that special case, the communication deviations and linear deviations will coincide[8.7]. This property was also proven by Fujii [117], but our framework gives a particularly simple proof via Theorem 4.26: for any UTC deviation in a single-stage game, the deviator makes either no queries or one query to the mediator. A communication deviator can simulate the same function by making the same query (if any), or, if the UTC deviator makes no query, by making an arbitrary query and ignoring the reply. It turns out that the converse is also essentially true:

Theorem 8.49. *Consider any decision problem with no nontrivial decision points—that is, the player has at least two legal actions at every decision point. The (timed) communication deviations coincide with the untimed communication deviations (and hence also the linear deviations)* if and only if *every path through the decision problem contains at most one decision point.*

Proof. The *if* direction was shown above and by Fujii [117], so it suffices to show the *only if* direction. Suppose there are two decision points, A and B, such that B is a child of action a_1 at A. Let a_2 be another action at A, and let b_1 and b_2 be two actions at B. (The game in Figure 8.14 has such a structure). Consider any deviation ϕ that plays action a_i if it is recommended b_i, for $i \in \{1, 2\}$. It is easy to construct untimed deviations with this behavior, but timed deviations cannot have this behavior, because a timed deviation cannot know the recommendation at B while still at decision point A. □

8.8.4.5 Relation between Our Representation and That of Farina and Pipis [95]

This chapter and the paper of Farina and Pipis [95] both take similar approaches to minimizing Φ_{LIN}-regret: both use the framework of Gordon et al. [132] to reduce the problem to minimizing external regret over the set of linear maps, and then derive a system of constraints for the set of matrices $\mathbf{A} \in \mathbb{R}^{\Sigma \times \Sigma}$ that represent linear maps. The representations are, however, significantly different:

- The representation of Farina and Pipis [95] cannot be expressed in scaled extensions. As such, that paper was forced to resort to less efficient regret minimization techniques. This difference is what allows us to improve upon their results.
- As a technical note, the representation of Farina and Pipis [95] will always result in a matrix $\mathbf{A} \in \mathbb{R}^{\Sigma \times \Sigma}$ where the columns of $\mathbf{A}$ corresponding to nonterminal sequences are filled with zeros. While this is without loss of generality due to the constraints defining the sequence form, it sometimes results in intuitively-strange representations: for example, their representation does not represent the identity map $\mathrm{Id} : \Pi \to \Pi$ as the identity matrix $\mathbf{I} \in \mathbb{R}^{\Sigma \times \Sigma}$, whereas our representation will.

[8.7] Here, by two sets of deviations *coinciding*, we mean that the same set of deviation functions $\phi : \Pi \to \mathcal{X}$ is available to both deviators.

- While our representation generalizes to arbitrary pairs of sequence-form polytopes according to Section 8.8.4.2, theirs generalizes even further, to functions $\phi : \mathcal{Y} \to \mathcal{X}$ as long as $\mathcal{Y}$ is sequence form and $\mathcal{X}$ has *some* small set of linear constraints (not necessarily sequence form) describing it. We likely cannot hope for our representation to generalize as far: our proof of equivalence relies fundamentally on both input and output being sequence-form strategy sets.

8.8.4.6 Untimed Communication Equilibria

The UTC deviations, like all sets of deviations, give rise to a notion of equilibrium. We define:

Definition 8.50. In an extensive-form game, an *untimed private communication equilibrium* is a correlated profile that is a (Φ_i)-equilibrium where Φ_i is player i's set of UTC deviations.

We add the word "private" here in the name to emphasize the fact that the mediator must have a separate interaction with each player—that is, the mediator cannot use its interactions with one player to inform how it gives recommendations to another player. This is enforced by the fact that the equilibrium is a correlated profile. See Chapter 6 regarding why this distinction is important.

Defining untimed communication equilibrium without such a privacy restriction seems to be a subtle task, and is orthogonal to and beyond the scope of the present work. However, we will make a few informal comments here. Untimed communication equilibria (without the privacy constraint) are difficult to define in a way that does not quickly collapse to the regular notion of communication equilibrium. In games with three or more players, the mediator is always guaranteed that two of the players have not deviated, and those two players will have messages synchronized with the game clock. Therefore, under reasonable assumptions on how often each player makes moves, the mediator will immediately know if the deviating player is sending out-of-order messages, and this concept would reduce immediately to the regular communication equilibrium. It is entirely unclear how to define a notion of untimed (non-private) communication equilibrium that does not exhibit such a collapse.

In two-player games, it is possible that there is a reasonable way to define untimed communication equilibria. The above collapse does not apply, because the mediator will not know which player is the one sending out-of-timing messages. However, this definition would still be rather subtle—for example, when do the out-of-order messages arrive to the mediator, relative to the *other player's* messages? We leave these issues to future work.

8.8.4.7 Revelation Principle for Untimed Private Commnunication Equilibrium

All the other notions of equilibrium involving a mediator, discussed in Section 5.4.1, obey a *revelation principle*, which we now discuss using the example of normal-form correlated equilibrium. The original definition of Aumann [15] did not initially refer to correlated profiles; instead, the definition posited an arbitrary joint distribution of correlated signals. In this section, we break down the notions of equilibrium that we have defined so far, and reconstruct them from the perspective of this arbitrary set of signals, and show that the resulting notions are equivalent for our notion of untimed private communication equilibrium.

We start with NFCE as an illustrative example. Let $\mu \in \Delta(\mathcal{S}_1 \times \cdots \times \mathcal{S}_n)$, where $\mathcal{S}_i$ is an arbitrary set of signals for player i. The mediator samples a joint signal $(s_1, \dots, s_n) \sim \mu$, and then each player privately observes its own signal s_i and selects (possibly at random) a strategy $\boldsymbol{x} \in \Pi$. An NFCE is then a tuple $(\mu, \phi_1, \dots, \phi_n)$, where $\phi_i : \mathcal{S}_i \to \mathcal{X}$ is the function by which player i selects its strategy given a signal, such that each player's choice of ϕ_i maximizes that player's utility given μ and the other ϕ_is, that is,

$$\mathop{\mathbb{E}}_{(s_1,\dots,s_n)\sim\mu} [u_i(\phi_i'(s_i), \phi_{-i}(s_{-i}) - u_i(\phi_i(s_i), \phi_{-i}(s_{-i})] \le 0$$

for every player i and other possible function $\phi_i' : \mathcal{S}_i \to \mathcal{X}$. An NFCE is *direct* if $\mathcal{S}_i = \Pi_i$ and $\phi_i : \Pi_i \to \mathcal{X}_i$ is the identity function. The *revelation principle* states that every NFCE is outcome-equivalent[8.8] to a direct equilibrium.

To generalize this to extensive-form and communication equilibria (timed and untimed), we follow the approach of Forges [109], Myerson [228]. In their approach, a mediator of player i is a map $M_i : \mathcal{S}_i^{\le H} \to \mathcal{S}_i$ (where $\mathcal{S}_i^{\le H}$ is the set of all sequences over $\mathcal{S}_i$ of length $\le H$, and H is some large but finite number (at least the depth of player i's decision problem.) that determines what message the mediator sends in reply to a player whose message history with the mediator is a finite sequence $\boldsymbol{s} = \{s_i\}$. We will assume that $\mathcal{S}_i$ at least is expressive enough to send an empty message, a decision point, or an observation point: $\mathcal{S}_i \supseteq \mathcal{J}_i \sqcup \Sigma_i \sqcup \{\bot\}$. The three notions of extensive-form correlated equilibrium, private communication equilibrium, and untimed private communication equilibrium will differ in how the player interacts with the mediator. We will describe player i's interactions by a set of functions $\Phi_i \subseteq (\mathcal{X}_i)^{\mathcal{M}_i}$ where $\mathcal{M}_i$ is a set of mediators: each function $\phi_i \in \Phi_i$ represents the player i choosing how it interacts with the mediator and how it uses those interactions to inform its choices of action. Then, as before, an equilibrium is a tuple $(\mu, \phi_1, \dots, \phi_n)$ where $\phi \in \Delta(\mathcal{M}_1 \times \dots \mathcal{M}_n)$ is a distribution over mediators and no player i can profit by switching to a different $\phi_i' \in \Phi_i$. The three notions above then differ in the choice of set Φ_i:

- Extensive-form correlated equilibria are equilibria where Φ_i is the set of interactions in which the player, upon reaching a decision point j, must send that decision point to the mediator.
- Private communication equilibria are equilibria where Φ_i is the set of interactions in which the player, upon reaching a decision point j, must send a single message (which may or may not be the decision point j) to the mediator.
- Untimed private communication equilibria are equilibria where Φ_i is the set of interactions in which the player, upon reaching a decision point j, may send any number of messages to the mediator.

The *direct mediator* $M_i^{\boldsymbol{x}_i}$ for a pure strategy $\boldsymbol{x}_i \in \Pi_i$ is the mediator who acts by sending the recommendation a at infoset j if and only if the message history matches the $\varnothing_i \to j$ path, otherwise $\bot$. Formally, $M_i^{\boldsymbol{x}_i}(\boldsymbol{s}) = a(\boldsymbol{x}_i, j)$ if $\boldsymbol{s} = (j^{(1)}, a^{(1)}, j^{(2)}, \dots, j)$ is the path to j in player

[8.8] By *outcome-equivalent*, we mean that the distribution over terminal nodes in the extensive-form game is the same in both equilibria.

i's decision tree, and $\perp$ otherwise. We write $\mathcal{M}_i^* := \{M_i^{\boldsymbol{x}_i} \mid \boldsymbol{x}_i \in \Pi_i\}$ for the set of direct mediators on Π_i. Notice that, for direct M_i, the sets of interactions valid for each of the three equilibrium notions reduces to the sets of deviations defined in Section 5.4.1. Analogous to the NFCE case, an equilibrium $(\mu, \phi_1^*, \dots, \phi_n^*)$ (in any of the previous three notions) is called *direct* if μ is a distribution over direct mediators, and ϕ_i^* is the map $M_i^{\boldsymbol{x}_i} \mapsto \boldsymbol{x}_i$ (which is the analogy of the identity map). We are now ready to state the revelation principle for these notions.

Theorem 8.51 (Revelation principle: for EFCE, proven by von Stengel and Forges [291]; for communication equilibrium, proven by Forges [109], Myerson [228] and refined in Chapter 6)**.** *For EFCE and (private) communication equilibrium, every equilibrium is outcome-equivalent to a direct equilibrium.*

Our main result in this section is that the same holds for untimed private communication equilibrium:

Theorem 8.52 (Revelation principle for untimed private communication equilibrium)**.** *Every untimed private communication equilibrium is outcome-equivalent to a direct untimed private communication equilibrium.*

Proof. Let $(\mu, \phi_1, \dots, \phi_n)$ be some (possibly indirect) equilibrium. Observe that we can view the mediator as holding a strategy $\boldsymbol{y} \in \mathcal{Y}_i$, where $\mathcal{Y}_i$ is the decision problem whose nodes correspond to sequences $\boldsymbol{s} \in \mathcal{S}_i^{\le H}$, *i.e.*, to message histories. Notice that, by construction of the message set $\mathcal{S}_i$, $\mathcal{Y}$ contains a copy of each Π_i within it, and that direct mediators $M_i^{\boldsymbol{x}_i}$ constrain themselves to states within this copy of Π_i by terminating the interaction (sending $\perp$ forever) if the history of communication fails to match a state in player i's decision problem. We will use this fact later.

By Theorem 8.48, each player's strategy set Φ_i is the set of linear maps $\mathcal{Y}_i \to \mathcal{X}_i$. Now, consider the direct profile $(\mu^*, \phi_i^*, \dots, \phi_n^*)$ where $\mu^* \in \Delta(\mathcal{M}_1^* \times \cdots \times \mathcal{M}_n^*)$ is given by sampling $(M_1, \dots, M_n) \sim \mu$, sampling $\boldsymbol{x}_i \in \Pi_i$ from any distribution whose expectation is $\phi_i(M_i)$ for every player i, and finally outputting $(M_1^{\boldsymbol{x}_1}, \dots, M_n^{\boldsymbol{x}_n})$. Clearly, this profile is outcome-equivalent to the original profile, so it only remains to show that it is also an equilibrium. Consider any deviation ϕ_i' of player i from the direct equilibrium.

We proceed by contrapositive. Suppose that $(\mu^*, \phi_i^*, \dots, \phi_n^*)$ is not an equilibrium: player i has profitable deviation ϕ_i'. Since a direct mediator is constrained, as above, to act within player i's decision problem, ϕ_i' can be expressed as a UTC deviation $\phi_i' : \Pi_i \to \mathcal{X}_i$. Since all UTC deviations are linear, ϕ_i' is itself linear, and can also be extended to a function $\phi_i' : \mathcal{X}_i \to \mathcal{X}_i$. Now let $\psi_i : \mathcal{Y}_i \to \mathcal{X}_i$ be given by $\psi_i = \phi_i' \circ \phi_i$, and observe that, since the composition of linear functions is linear, ψ_i is a linear map, that is, $\psi_i \in \Phi_i$. Moreover, by construction, the profiles (μ, ψ_i, ϕ_{-i}) and $(\mu^*, \phi_i', \phi_{-i}^*)$ must induce the same outcome distributions—and therefore, ψ_i is a profitable deviation against the original equilibrium $(\mu, \phi_1, \dots, \phi_n)$. □

This result justifies the definitions of equilibrium we have been using throughout the section before reaching this point. We remark that, although the proof is usually not difficult, the revelation principle is not a given or automatic fact that can be assumed without proof: there are other settings where it fails, such as when the deviator's set of allowable messages depends on its true type in a nontrivial manner (*e.g.*, [110, 173]).

8.8.5 Generalization Beyond Extensive-Form Games

We now show that the ability to write Φ_{LIN} as a system of linear constraints is not special to extensive-form games. In fact, we have the following result:

Theorem 8.53. *Let $\mathcal{X}$ be a polytope given by explicit linear constraints, $\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^d : \mathbf{M}\boldsymbol{x} \le \boldsymbol{u}\}$. Let $\mathbf{A} \in \mathbb{R}^{d\times d}$. Then $\mathbf{A}\boldsymbol{x} \in \mathcal{X}$ for all $\boldsymbol{x} \in \mathcal{X}$ if and only if there is a matrix $\mathbf{B} \in \mathbb{R}^{m\times m}$ satisfying the constraints*

$$\mathbf{BM} = \mathbf{MA}, \quad \mathbf{B}\boldsymbol{u} \le \boldsymbol{u}, \quad \mathbf{B} \ge \mathbf{0}. \tag{8.10}$$

Proof. Let $\mathbf{A} \in \mathbb{R}^{d\times d}$, and let $\boldsymbol{m}_i^\top \boldsymbol{x} \le b_i$ be the ith constraint that defines $\mathcal{X}$. Then, the claim that $\boldsymbol{m}_i^\top \mathbf{A}\boldsymbol{x} \le b_i$ for every $\boldsymbol{x} \in \mathcal{X}$ is equivalent to the claim that the linear program

$$\max_{\boldsymbol{x}} \quad \boldsymbol{m}_i^\top \mathbf{A}\boldsymbol{x} \quad \text{s.t.} \quad \mathbf{M}\boldsymbol{x} \le \boldsymbol{u} \tag{8.11}$$

has value at most b_i. By strong duality, (8.11) has the same value as

$$\min_{\boldsymbol{u}_i} \quad \boldsymbol{u}^\top \boldsymbol{u}_i \quad \text{s.t.} \quad \mathbf{M}^\top \boldsymbol{u}_i = \mathbf{A}^\top \boldsymbol{m}_i, \quad \boldsymbol{u} \ge \mathbf{0}.$$

The theorem follows now by setting $\mathbf{B} = \begin{bmatrix} \boldsymbol{u}_1 & \dots & \boldsymbol{u}_k \end{bmatrix}^\top$. □

Furthermore, assuming that $\mathcal{B}_1(\mathbf{0}) \subseteq \mathcal{X} \subseteq \mathcal{B}_R(\mathbf{0})$ with $R \le \mathsf{poly}(d)$, it follows that $\|\mathbf{A}\|_2, \|\mathbf{B}\|_2 \le \mathsf{poly}(d)$, where $\|\cdot\|_2$ denotes the spectral norm. Indeed, for $\|\mathbf{A}\|_2$, take any $\boldsymbol{x} \in \mathbb{R}^d$ with $\|\boldsymbol{x}\| = 1$. Since $\mathcal{B}_1(\mathbf{0}) \subseteq \mathcal{X}$, we have $\boldsymbol{x} \in \mathcal{X}$, in turn implying that $\mathbf{A}\boldsymbol{x} \in \mathcal{X}$. As a result, $\|\mathbf{A}\boldsymbol{x}\| \le \mathsf{poly}(d)$, from which it follows that $\|\mathbf{A}\|_2 \le \mathsf{poly}(d)$. Further, one can take each b_i to be such that $1 \le b_i \le \mathsf{poly}(d)$, and so the bound $\|\mathbf{B}\|_2 \le \mathsf{poly}(d)$ follows from the fact that $\mathbf{B}\boldsymbol{u} \le \boldsymbol{u}$ and $\mathbf{B} \ge 0$.

Combining these bounds with Theorem 8.53, we are able to use standard techniques for minimizing regret over Φ_{LIN}—such as projected gradient descent. As a consequence, we can instantiate the regret minimizer operating over Φ_{LIN} with projected gradient descent.

Corollary 8.54. *There is a deterministic algorithm that guarantees $\textsc{Reg}_{\Phi_{\mathsf{LIN}}}(T) \le \epsilon$ after $\mathsf{poly}(d, m)/\epsilon^2$ rounds, and requires solving a convex quadratic program with $O(d^2 + m^2)$ variables and constraints in each iteration.*

For comparison, let us compare to the approach of Daskalakis et al. [81] for the case where $\mathcal{X}$ is given explicitly. To do so, we recall the following definition.

Definition 8.55. We say that a polytope $\mathcal{X}$ has an *H-representation* of size m if it is given as the intersection of m halfspaces: $\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^d : \mathbf{A}\boldsymbol{x} \leq \boldsymbol{b}\}$ for some $\mathbf{A} \in \mathbb{Q}^{m \times d}$ and $\boldsymbol{b} \in \mathbb{Q}^m$. It has a *V-representation* of size m if it is given as the convex hull of m vertices: $\mathcal{X} = \operatorname{conv}(\{\boldsymbol{v}_1, \dots, \boldsymbol{v}_m\})$ for $\boldsymbol{v}_1, \dots, \boldsymbol{v}_m \in \mathbb{Q}^d$.

In this context, they make the following crucial observation [81, Lemmas 3.1 and 3.2].

Lemma 8.56 ([81])**.** *If $\mathcal{X}$ has either an H-representation of size m or a V-representation of size m, there is a* $\mathsf{poly}(d, m)$*-time membership oracle for* Φ_{LIN}.

Using a membership oracle for Φ_{LIN}, it is also possible to construct a linear optimization oracle [138]. As a result, coupled with Lemma 8.56, standard algorithms—such as *follow-the-perturbed-leader* [145]—can be applied to minimize regret over Φ_{LIN}. However, the main limitation is that constructing a linear optimization oracle using a membership oracle relies on the ellipsoid algorithm, which is impractical. In contrast, Theorem 8.53 allows us to bypass using the ellipsoid algorithm, resulting in a more practical approach. Therefore, we obtain the best-known algorithm for linear-swap regret minimization over explicitly-represented polytopes, improving on Daskalakis et al. [81] by reducing the per-iteration complexity.

Remark 8.57. The construction in Theorem 8.53 is *not* equivalent to our DAG-based construction of Φ_{LIN} in the case of extensive-form games. Indeed, Theorem 8.53 implies that *every* matrix $\mathbf{A}$ representing a linear endomorphism on $\mathcal{X}$ induces a feasible solution to (8.10), and Section 8.8.4.3 guarantees that this is not true of the DAG representation. Our DAG representation is significantly more efficient in the special case of extensive-form games, as it permits DAG-CFR-based algorithms, which have time complexity linear in the size of the DAG.

8.8.6 Experimental Evaluation

We empirically investigate the performance of our learning dynamics for linear correlated equilibrium, compared to the recent algorithm by Farina and Pipis [95]. We test on four benchmark games:

- 4-player Kuhn poker, a multiplayer variant of the classic benchmark game introduced by Kuhn [187]. The deck has 5 cards. This game has 3,960 terminal states.
- A ridesharing game, a two-player general-sum game introduced as a benchmark for welfare-maximizing equilibria by Zhang et al. [305]. This game has 484 terminal states.
- 3-player Leduc poker, a three-player variant of the classic Leduc poker introduced by Southey et al. [275]. Only one bet per round is allowed, and the deck has 6 cards (3 ranks, 2 suits). The game has 4,500 terminal states.
- Sheriff of Nottingham, a two-player general-sum game introduced by Farina et al. [101] for its richness of equilibrium points. The smuggler has 10 items, a maxmimum bribe of 2, and 2 rounds to bargain. The game has 2,376 terminal states.

We run our algorithm based on the UTC polytope, and that of Farina and Pipis [95] (with the learning rate $\eta = 0.1$ as used by the authors), for a limit of 100,000 iterations or 6 hours, whichever

Game	Our algorithm	Farina and Pipis [95]	Speedup
4-Player Kuhn poker	5.65ms ± 0.30ms	195ms ± 7ms	35×
Ridesharing game	676µs ± 80µs	160ms ± 7ms	237×
3-Player Leduc poker	42.0ms ± 0.7ms	12.1s ± 1.0s	287×
Sheriff of Nottingham	114ms ± 16ms	50.2s ± 9.6s	442×

Table 8.17: *Comparison of average time per iteration. For each combination of game instance and algorithm, the mean and standard deviation of the iteration runtime are noted.*

Game	Target gap	Our algorithm	Farina and Pipis [95]	Speedup
4-Player Kuhn poker	7×10^{-4}	32.8s	5h 25m	595×
Ridesharing game	9×10^{-5}	8.89s	4h 07m	1667×
3-Player Leduc poker	0.224	2.12s	6h 00m	10179×
Sheriff of Nottingham	2.06	2.00s	6h 00m	10800×

Table 8.18: *Comparison of time taken to achieve a particular linear swap equilibrium gap. The gap is whatever gap was achieved by the algorithm of Farina and Pipis [95] before termination.*

is hit first. Instead of solving linear programs to find the fixed points, we use power iteration, which is faster in practice. All experiments were run on the same machine with 32GB of RAM and a processor running at a nominal speed of 2.4GHz. For our learning dynamics, we employed the CFR algorithm instantiated with the regret matching$^+$ [282] regret minimizer at each decision point (see Theorem 8.45). Experimental results are shown in Figure 8.15.

One of the most appealing features of our algorithm is that allows CFR-based methods to apply. CFR-based methods are the fastest regret minimizers in practice, so it is unsurprising that using them results in better convergence as seen in Figure 8.15. Another appealing feature is that our method sidesteps the need of projecting onto the set of transformations. This is in contrast with the algorithm of Farina and Pipis [95], which requires an expensive projection at every iteration. We observe that this difference results in a dramatic reduction in iteration runtime between the two algorithms, which we quantify in Table 8.17. So, we remark that when accounting for *time* instead of iterations on the x-axis of the plots in Figure 8.15, the difference in performance between the algorithms appears even stronger. Such a plot is available in the appendix of the full paper [307].

8.9 Hardness of Minimizing Φ-Regret in Behavioral Strategies

In this section, we show that if the learner is constrained to output in reach round a strategy in $\mathcal{X}$ (*i.e.*, $\mu^{(t)}$ is restricted to be a point distribution), then there is no efficient algorithm (under standard complexity assumptions) minimizing Φ-regret (Theorem 8.4). The key connection is an observation by Hazan and Kale [146], which reveals that any Φ-regret minimizer is inadvertedly able to compute approximate fixed points of any deviations in Φ. We then show that the set of induced deviations, even on the hypercube $\mathcal{X} = [0, 1]^d$, is rich enough to approximate PPAD-hard fixed-point problems.

In this context, consider a transformation $\Phi \ni \phi : \mathcal{X} \to \mathcal{X}$ for which we want to compute an approximate fixed point $\boldsymbol{x} \in \operatorname{conv} \mathcal{X}$; that is, $\|\phi(\boldsymbol{x}) - \boldsymbol{x}\|_2 \leq \epsilon$, for some precision parameter $\epsilon > 0$. (It is convenient in the construction below to measure the fixed-point error with respect to $\| \cdot \|_2$.) Hazan and Kale [146] observed that a Φ-regret minimizer can be readily turned into an algorithm for computing fixed points of any function in Φ, as stated formally below. Before we proceed, we remind that here and throughout we operate under a strongly adaptive adversary, which is quite crucial in the construction of Hazan and Kale [146].

Proposition 8.58 ([146]). *Consider a regret minimizer $\mathcal{R}$ operating over an arbitrary strategy set $\mathcal{X}$. If $\mathcal{R}$ runs in time $\mathsf{poly}(d, 1/\epsilon)$ and guarantees $\textsc{Reg}_\Phi(T) \leq \epsilon$ for any sequence of utilities, then there is a $\mathsf{poly}(d, 1/\epsilon)$ algorithm for computing an ϵ-fixed point of any $\phi \in \Phi$ with respect to $\| \cdot \|_2$, assuming that ϕ can be evaluated in polynomial time.*

Proposition 8.58 significantly circumscribes the class of problems for which efficient Φ-regret minimization is possible, at least when operating in behavioral strategies. Indeed, computing fixed points is in general a well-known (presumably) intractable problem. In our context, the set Φ does not contain arbitrary (Lipschitz continuous) functions $\mathcal{X} \to \mathcal{X}$, but instead contains multilinear functions from $\mathcal{X}$ to $\mathcal{X}$. To show Theorem 8.4, we start with a *generalized circuit*, and we show that all gates can be approximately simulated using exclusively gates involving multilinear operations; we defer the formal argument to the appendix of the full paper [317]. As a result, we arrive at the main hardness result of this section.

We also obtain a stronger hardness result under a stronger complexity assumption put forward by Babichenko et al. [16], which also can be found in the appendix of the full paper [317]. As we have already discsused, the above result justifies the importance of expected fixed points in the positive results for computing equilibria and regret minimization.

Since Proposition 8.58 can be circumvented, one may wonder whether it is possible “fix” it by finding an expected fixed point instead of a regular fixed point. Indeed, this is possible:

Proposition 8.59. *Consider a regret minimizer $\mathcal{R}$ operating over $\Delta(\mathcal{X})$, where $\mathcal{X} \subseteq \mathcal{B}_R(0)$. If $\mathcal{R}$ runs in time $\mathsf{poly}(d, R/\epsilon)$-time and guarantees $\textsc{Reg}_\Phi(T) \leq \epsilon$ for any sequence of utilities, then there is a $\mathsf{poly}(d, R/\epsilon)$ algorithm for computing ϵ-expected fixed points of $\phi \in \Phi$, assuming that we can efficiently compute $\mathbb{E}_{\boldsymbol{x}^{(t)} \sim \mu^{(t)}}[\phi(\boldsymbol{x}^{(t)}) - \boldsymbol{x}^{(t)}]$ at any time t.*

Proof. The proof proceeds similarly to Proposition 8.58. Suppose that $\mathcal{R}$ outputs a strategy $\mu^{(t)} \in \Delta(\mathcal{X})$. At each time $t \in [T]$, we can terminate if $\| \mathbb{E}_{\boldsymbol{x}^{(t)} \sim \mu^{(t)}}[\phi(\boldsymbol{x}^{(t)}) - \boldsymbol{x}^{(t)}]\|_2 \leq \epsilon R$; that is, we have identified an (ϵR)-expected fixed point. Otherwise, we construct the utility function

$$u^{(t)} : \mathcal{X} \ni \boldsymbol{x} \mapsto \frac{1}{R} \frac{1}{\| \mathbb{E}_{\boldsymbol{x}^{(t)} \sim \mu^{(t)}} [\phi(\boldsymbol{x}^{(t)}) - \boldsymbol{x}^{(t)}] \|_2} \left\langle \mathop{\mathbb{E}}_{\boldsymbol{x}^{(t)} \sim \mu^{(t)}} [\phi(\boldsymbol{x}^{(t)}) - \boldsymbol{x}^{(t)}], \boldsymbol{x} - \mu^{(t)} \right\rangle,$$

which indeed satisfies the normalization constraint $|u^{(t)}(\boldsymbol{x})| \leq 1$. Now, if at all iterations it was the case that $\| \mathbb{E}_{\boldsymbol{x}^{(t)} \sim \mu^{(t)}} [\phi(\boldsymbol{x}^{(t)}) - \boldsymbol{x}^{(t)}] \|_2 > \epsilon R$, we have

$$\textsc{Reg}_\Phi^T \geq \frac{1}{T} \sum_{t=1}^{T} u^{(t)} \left(\mathop{\mathbb{E}}_{\boldsymbol{x}^{(t)} \sim \mu^{(t)}} [\phi(\boldsymbol{x}^{(t)})] \right) - \frac{1}{T} \sum_{t=1}^{T} u^{(t)}(\mu^{(t)}) > \epsilon$$

since $u^{(t)}(\mu^{(t)}) = 0$ and

$$u^{(t)} \left(\mathop{\mathbb{E}}_{\boldsymbol{x}^{(t)} \sim \mu^{(t)}} [\phi(\boldsymbol{x}^{(t)})] \right) = \frac{1}{R} \left\| \mathop{\mathbb{E}}_{\boldsymbol{x}^{(t)} \sim \mu^{(t)}} [\phi(\boldsymbol{x}^{(t)}) - \boldsymbol{x}^{(t)}] \right\|_2$$

for all $t \in [T]$. This contradicts the assumption that $\textsc{Reg}_\Phi^T \leq \epsilon$ for any sequence of utilities, in turn implying that there exists $t \in [T]$ such that $\mu^{(t)}$ is an ϵ-expected fixed point. Given that, by assumption, we can compute $\mathbb{E}_{\boldsymbol{x}^{(t)} \sim \mu^{(t)}} [\phi(\boldsymbol{x}^{(t)}) - \boldsymbol{x}^{(t)}]$ for any time t, the claim follows. □

8.10 Impossibility of Minimizing Swap Regret in Extensive-Form Games

We now show generic lower bounds on minimizing swap regret, that is, regret against the set of all functions $\Phi = \Pi^\Pi$. For notation, in this subsection, we will define

$$V(\phi) := \frac{1}{T} \sum_{t=1}^{T} \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mu^t} \left\langle \boldsymbol{u}^t, \phi(\boldsymbol{x}) \right\rangle.$$

Thus, in particular, the total utility experienced by the learner is $V(\mathrm{Id})$. After T rounds, the *(average) swap-regret* is

$$\textsc{Reg}_{\textsc{SWAP}}(T) := \max_{\phi:\Pi \to \Pi} V(\phi) - V(\mathrm{Id}) = \max_{\phi:\Pi \to \Pi} \frac{1}{T} \sum_{t=1}^{T} \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mu^t} \left\langle \boldsymbol{u}^t, \phi(\boldsymbol{x}) - \boldsymbol{x} \right\rangle.$$

The learner's goal is then to achieve small swap regret after a small number of rounds: for example, one may hope to achieve swap regret ϵ after $T = \mathsf{poly}(d, 1/\epsilon)$ rounds.

The main result of this section is the following.

Theorem 8.60 (Extensive-form lower bound)**.** *There exist arbitrarily large tree-from strategy sets* $\Pi \subset \{0,1\}^d$ *with the following property. Let* $\epsilon > 0$ *and suppose* $T \leq \exp(\Omega(\min\{d^{1/14}, \epsilon^{-1/6}\}))$. *Then there exists an oblivious adversary running for T iterations against which no learner can achieve expected swap regret better than* ϵ.

Intuitively, the proof of Theorem 8.60 works by finding an "embedding" of the adversary of Theorem 8.14 into a tree-form decision problem such that the utility functions u^t do remain linear. This works by choosing random vectors in $\{-1, 1\}^n$ (for some appropriately chosen dimension n) to simulate the "actions" in the (exponentially large) normal-form decision problem, and then exploiting the concentration property that an exponentially large number of such vectors $\{\boldsymbol{a}_i\}_{i=1}^N$ can be chosen such that $\langle \boldsymbol{a}_i, \boldsymbol{a}_j \rangle \approx 0$ for all $i \neq j$.

Like Theorem 8.14, our lower bound is *information-theoretic*: it does not rely on computational hardness results, and thus applies to *any* no-regret learning algorithm no matter how much computation it might perform.

Before proving Theorem 8.60, we first state a more detailed version of the normal-form lower bound (Theorem 8.14).[8.9] This restatement changes the notation so as to avoid mixing the notation between tree-form and normal-form decision problems, and extracts some useful properties of the adversary. In particular, a key property of the adversary that we exploit (specified in Item 1 in the below theorem) is that, with probability 1, each of the vectors $\boldsymbol{u}^t$ that it chooses is in fact a unit vector $\boldsymbol{e}_i$. (In general, in the normal-form game setting, the vectors $\boldsymbol{u}^t$ may have coordinates in $[-1, 1]$.)

Theorem 8.61 ([71], expanded version of Theorem 8.14)**.** *Let $\mathcal{A} = \Delta(N)$ be the N-simplex, and let $T < N/4$. Then there exists an adversary on $\mathcal{A}$ with the following properties:*

1. *The adversary selects a sequence $(\mathbf{u}^1, \dots, \mathbf{u}^T) \sim \mathcal{D}$ from some distribution $\mathcal{D} \in \Delta(\mathcal{A}^T)$, and then outputs utility vector $\mathbf{u}^t$ at time t regardless of the sequence of distributions played by the learner.*
2. *There exists an action $\boldsymbol{a}^* \in \mathcal{A}$ that is never used by the adversary.*[8.10]
3. *There exists a partition $\mathcal{A} = \mathcal{A}_1 \sqcup \cdots \sqcup \mathcal{A}_\ell$ where $\ell \le O(\log T)$ with the following property. Within each set $\mathcal{A}_i$, number the actions $\mathcal{A}_i = \{\boldsymbol{a}_{i1}, \dots, \boldsymbol{a}_{iN_i}\}$. For any sequence $(\boldsymbol{u}^1, \dots, \boldsymbol{u}^T) \in \operatorname{supp} \mathcal{D}$, the adversary plays actions in $\mathcal{A}_i$ only in increasing order. That is, if $\mathbf{u}^t = \boldsymbol{a}_{ij}$ and $\mathbf{u}^{t'} = \boldsymbol{a}_{ij'}$ and $t \le t'$, then $j \le j'$.*
4. *The swap regret of any learner against this adversary is*[8.11] $\Omega(\ell^{-6}) = \Omega(\log^{-6} T)$.

Proof of Theorem 8.60. We proceed via a reduction from Theorem 8.61.

Extensive-form game instances. Consider the following family of tree-from strategy sets, parameterized by natural numbers ℓ and n. First the learner picks an index $i \in [\ell]$. Then the environment picks $j \in [n]$, and finally the learner picks a binary action. This family of decision

[8.9] The discussion in Dagan et al. [71] on pages 37–38 of specifies the adversary which satisfies the properties listed in Theorem 8.61.

[8.10] This can always be assumed WLOG.

[8.11] The reason that the −5 in Theorem 8.14 has been changed to a −6 here is that the adversary is now constrained to pick a sequence of *actions*, *i.e.*, ℓ_1-bounded losses, instead of ℓ_∞-bounded losses. See [71], Theorems 1.7 and 4.1

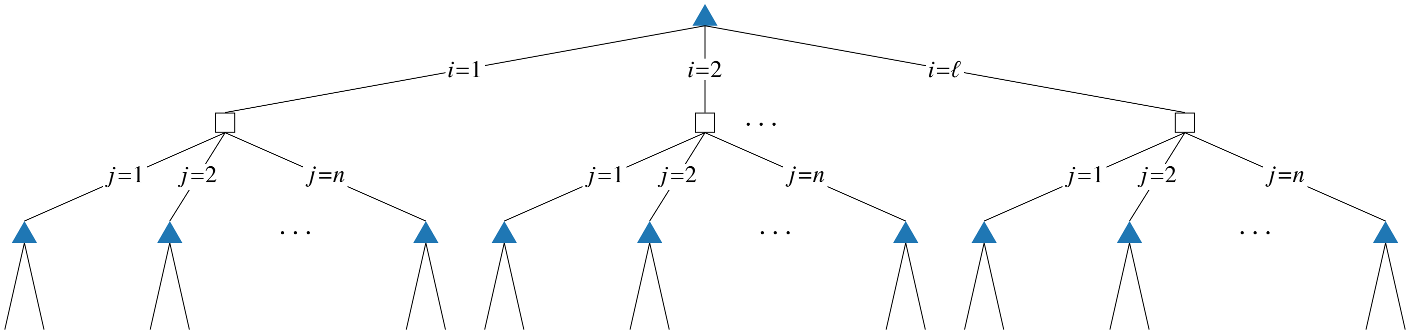


Figure 8.19: *A depiction of the class of tree-form decision problems used in the proof of Theorem 8.60. Triangles (▲) are decision points and boxes (□) are observation points.*

problems is depicted in Figure 8.19.

A pure strategy is identified (up to linear transformations) by a vector $\boldsymbol{x} \in \mathbb{R}^{\ell \times n}$ where, for some $i \in [\ell]$, $\boldsymbol{x}(i, \cdot) \in \{-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}\}^n$ and $\boldsymbol{x}(i', \cdot) = \mathbf{0}$ if $i' \neq i$ (*i.e.*, $\boldsymbol{x}$ interpreted as a matrix with exactly one nonzero row). For convenience, we will use $\Pi_i \subset \Pi$ to denote the set of (pure) strategies where the learner plays i at the root. Let C be an absolute constant large enough to make the asymptotic bounds in Theorem 8.61 true.

The adversarial environment. The adversary used to prove Theorem 8.60 works as follows. First, for each $i \in [\ell]$, it populates $\mathcal{A}_i$ with N_i uniformly randomly chosen strategies in Π_i. Formally, we let $\psi : \mathcal{A} \to \Pi$ denote the (random) mapping which associates each action in $\mathcal{A}_i \subset \mathcal{A}$ with the corresponding action in Π_i: the image of $\mathcal{A}_i$ under ψ consists of actions we denote by $\tilde{\boldsymbol{a}}_{i1}, \dots, \tilde{\boldsymbol{a}}_{iN_i} \in \Pi_i$, which are chosen independently and uniformly in Π_i. The adversary in Theorem 8.61 produces a random sequence $\boldsymbol{u}^1, \dots, \boldsymbol{u}^T \in \mathcal{A}$; we consider the adversary which draws a sequence $\boldsymbol{u}^t$ from that distribution and outputs the sequence consisting of $\tilde{\boldsymbol{u}}^t := \psi(\boldsymbol{u}^t)$ for $t \in [T]$.

Analysis. Let $\epsilon > 0$ be a parameter to be selected later. We start with a simple concetration bound.

Lemma 8.62. *Let $\delta = N^2 e^{-n\epsilon^2/2}$. With probability $1 - \delta$, for all $\boldsymbol{a}, \boldsymbol{a}' \in \mathcal{A}$, we have $|\langle \tilde{\boldsymbol{a}}, \tilde{\boldsymbol{a}}' \rangle - \mathbb{1}\{\boldsymbol{a} = \boldsymbol{a}'\}| \leq \epsilon$.*

Proof. If $\boldsymbol{a} = \boldsymbol{a}'$ then the claim holds trivially because then $\tilde{\boldsymbol{a}} = \tilde{\boldsymbol{a}}'$, and they are both unit vectors. For a fixed $\boldsymbol{a} \neq \boldsymbol{a}' \in \mathcal{A}$, the claim holds with probability $2e^{-n\epsilon^2/2}$ by Hoeffding's inequality. The lemma then follows by union bounding over the $\binom{N}{2} \leq N^2/2$ pairs. □

We will claim that, for any learner against this adversary, there exists a learner against the adversary of Theorem 8.61 that achieves a similar swap regret—and thus the swap regret of the former learner must be large. First, we will construct the latter learner.

Let $\mu^1, \ldots, \mu^T \in \Delta(\Pi)$ be the sequence of distributions played by the learner. Note that μ^t can depend on the utilities $\boldsymbol{u}^{1:t-1} \in \mathcal{A}$ that are played by the adversary. Consider the sequence $\bar{\mu}^1, \ldots, \bar{\mu}^T \in \Delta(\mathcal{A})$, where $\bar{\mu}^t$ is the distribution that samples $\boldsymbol{x} \sim \mu^t$ and plays according to $p_{\boldsymbol{x}} \in \Delta(\mathcal{A})$, defined as follows. Let $\boldsymbol{x} \in \Pi_i$ be any strategy. There are two cases.

1. $\langle \boldsymbol{x}, \tilde{\boldsymbol{a}}_{ij} \rangle \le \epsilon$ for every $i \in [\ell], j \in [N_i]$. Then define $p_{\boldsymbol{x}} = \boldsymbol{a}^*$ deterministically (i.e., $p_{\boldsymbol{x}}$ is the distribution which puts all of its mass on $\boldsymbol{a}^*$).
2. $\langle \boldsymbol{x}, \tilde{\boldsymbol{a}}_{ij} \rangle > \epsilon$ for some $i \in [\ell], j \in [N_i]$. Let j be the *largest* such index, let $\beta = \langle \boldsymbol{x}, \tilde{\boldsymbol{a}}_{ij} \rangle$, and define $p_{\boldsymbol{x}}$ as the distribution that is $\boldsymbol{a}^*$ with probability $1 - \beta$ and $\boldsymbol{a}_{ij}$ with probability β. Note that $\beta \in [0, 1]$ since $\boldsymbol{x}, \tilde{\boldsymbol{a}}_{ij} \in \{-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}\}^n$ and in this case we have assumed that $\langle \boldsymbol{x}, \tilde{\boldsymbol{a}}_{ij} \rangle > \epsilon > 0$.

A critical property for us will be that the learner cannot "guess in advance" what future unobserved $\tilde{\boldsymbol{a}}_{ij}$s will be, since these are sampled uniformly at random. That is, in Case 2, $\boldsymbol{x}$ can only be played with large probability once the adversary has played $\tilde{\boldsymbol{a}}_{ij}$.

To be more formal, we first define some notation. For every $i \in [\ell], j \in [N_i]$, let t_{ij} be the first iteration on which the adversary plays $\tilde{\boldsymbol{a}}_{ij}$ (or $t_{ij} = T$ if this never happens). For $\boldsymbol{x} \in \Pi_i$, if $\boldsymbol{x}$ is in Case 1 above then define $t_{\boldsymbol{x}} = 0$, and otherwise define $t_{\boldsymbol{x}} = t_{ij}$, where j is as in Case 2.

There are two properties that we will critically need to use about $t_{\boldsymbol{x}}$. The first states that the learner cannot place large mass on $\boldsymbol{x}$ until after $t_{\boldsymbol{x}}$, because doing so would require the learner to guess a vector heavily correlated with $\tilde{\boldsymbol{a}}_{ij}$ before the learner observes $\tilde{\boldsymbol{a}}_{ij}$.

Lemma 8.63. $\mathbb{E} \dfrac{1}{T} \displaystyle\sum_{\boldsymbol{x} \in \Pi} \sum_{t=1}^{t_{\boldsymbol{x}}} \mu^t(\boldsymbol{x}) \le \delta.$

Proof. Since the learner has not yet observed $\tilde{\boldsymbol{a}}_{ij}$ at time t_{ij}, its prior strategy sequence $\mu^{1:t_{ij}}(\boldsymbol{x})$ must be independent of $\tilde{\boldsymbol{a}}_{ij}$. Moreover, if $t \le t_{\boldsymbol{x}}$ then there must exist some j with $t_{ij} \ge t$ and $\langle \boldsymbol{x}, \tilde{\boldsymbol{a}}_{ij} \rangle \ge \epsilon$—namely, the j defining Case 2. Thus we have:

$$\begin{aligned}
\mathbb{E} \frac{1}{T} \sum_{\boldsymbol{x} \in \Pi} \sum_{t=1}^{t_{\boldsymbol{x}}} \mu^t(\boldsymbol{x}) &\le \mathbb{E} \frac{1}{T} \sum_{i=1}^{d} \sum_{\boldsymbol{x} \in \Pi_i} \sum_{t=1}^{T} \mu^t(\boldsymbol{x}) \sum_{j : t_{ij} \ge t} \mathbb{1}\{\langle \boldsymbol{x}, \tilde{\boldsymbol{a}}_{ij} \rangle \ge \epsilon\} \\
&= \underbrace{\frac{1}{T} \sum_{i=1}^{d} \sum_{\boldsymbol{x} \in \Pi_i} \sum_{j=1}^{N_i} \mathbb{E}\left[\sum_{t \le t_{ij}} \mu^t(\boldsymbol{x})\right]}_{\le N} \underbrace{\mathbb{E}\left[\mathbb{1}\{\langle \boldsymbol{x}, \tilde{\boldsymbol{a}}_{ij} \rangle \ge \epsilon\}\right]}_{\le e^{-n\epsilon^2/2}} \le \delta,
\end{aligned}$$

where in the last line we use the fact that $\tilde{\boldsymbol{a}}_{ij}$ is independent of $\mu^{1:t_{ij}}(\boldsymbol{x})$ and then Hoeffding's inequality. Moreover, we have used in the final inequality that for each $i \in [\ell]$ and $j \le N_i \le N$, we have $\frac{1}{T} \sum_{\boldsymbol{x} \in \Pi} \sum_{t=1}^{T} \mathbb{E}[\mu^t(\boldsymbol{x})] \le 1$. □

The second property is that, for $t > t_{\boldsymbol{x}}$, utilities of $\boldsymbol{x}$ under $\tilde{\boldsymbol{u}}^t$ are approximately the same as those of $p_{\boldsymbol{x}}$ under the utilies $\boldsymbol{u}^t$ of Theorem 8.61.

Lemma 8.64. *For $t > t_{\boldsymbol{x}}$, we have $\left\langle \boldsymbol{x}, \tilde{\boldsymbol{u}}^t \right\rangle \le p_{\boldsymbol{x}}(\boldsymbol{u}^t) + \epsilon$.*

Proof. Let $\boldsymbol{x} \in \Pi_i$. There are two cases. In the first case, we suppose that $\left\langle \boldsymbol{x}, \tilde{\boldsymbol{a}}_{ij} \right\rangle \le \epsilon$ for every $i \in [\ell], j \in [N_i]$. Then for every t, we have $\boldsymbol{u}^t \notin \operatorname{supp} p_{\boldsymbol{x}} = \{\boldsymbol{a}^*\}$ (because the adversary of Theorem 8.61 never plays $\boldsymbol{a}^*$), and $\left\langle \boldsymbol{x}, \tilde{\boldsymbol{u}}^t \right\rangle \le \epsilon$ by definition, so we are done.

Otherwise, let j be the largest index for which $\left\langle \boldsymbol{x}, \tilde{\boldsymbol{a}}_{ij} \right\rangle > \epsilon$. Then $t_{\boldsymbol{x}} = t_{ij}$ by definition, and since $t > t_{ij}$, by Property 4, for time steps following t_{ij} the adversary of Theorem 8.61 is no longer allowed to play $\boldsymbol{a}_{ij'}$ for $j' < j$. Thus, either $\boldsymbol{u}^t = \boldsymbol{a}_{ij}$, or else $\boldsymbol{u}^t \notin \{\boldsymbol{a}_{i1}, \dots, \boldsymbol{a}_{ij}\}$. Since j is defined to be the largest index for which $\langle \boldsymbol{x}, \tilde{\boldsymbol{a}}_{ij} \rangle > \epsilon$, in the latter case we must have $p_{\boldsymbol{x}}$ puts all its mass on $\boldsymbol{a}^* \neq \boldsymbol{u}^t$, meaning that $\langle \boldsymbol{x}, \tilde{\boldsymbol{u}}^t \rangle \le \epsilon = p_{\boldsymbol{x}}(\boldsymbol{u}^t) + \epsilon$. In the former case, we have $\langle \boldsymbol{x}, \tilde{\boldsymbol{u}}^t \rangle = \beta = p_{\boldsymbol{x}}(\boldsymbol{a}_{ij}) = p_{\boldsymbol{x}}(\boldsymbol{u}^t)$. □

For the rest of this proof we will use $\bar{V}(\phi)$ to denote the utilities experienced by $\bar{\mu}^t$ under the utilities $\boldsymbol{u}^t$ in Theorem 8.61. That is,

$$\bar{V}(\phi) = \frac{1}{T} \sum_{t=1}^{T} \sum_{\boldsymbol{a} \in \mathcal{A}} \bar{\mu}^t(\boldsymbol{a}) \mathbb{1}\left\{\phi(\boldsymbol{a}) = \boldsymbol{u}^t\right\} = \frac{1}{T} \sum_{t=1}^{T} \sum_{\boldsymbol{x} \in \Pi} \mu^t(\boldsymbol{x}) \Pr_{\boldsymbol{a} \sim p_{\boldsymbol{x}}}[\phi(\boldsymbol{a}) = \boldsymbol{u}^t]$$

By Theorem 8.61, there exists a function $\bar{\phi} : \mathcal{A} \to \mathcal{A}$ such that[8.12] $\mathbb{E}[\bar{V}(\bar{\phi}) - \bar{V}(\mathrm{Id})] \ge 1/C\ell^6$. We define a deviation $\phi : \Pi \to \mathcal{X}$ by setting[8.13] $\phi(\boldsymbol{x}) := \mathbb{E}_{\boldsymbol{a} \sim p_{\boldsymbol{x}}} \psi(\bar{\phi}(\boldsymbol{a}))$.

It suffices to show that $\mathbb{E}[V(\phi) - V(\mathrm{Id})]$ is large. To do this, we will show that, in expectation and up to small errors, $V(\mathrm{Id}) \le \bar{V}(\mathrm{Id})$ and $V(\phi) \ge \bar{V}(\bar{\phi})$.

For the first approximation, we have

$$\begin{aligned}
V(\mathrm{Id}) &= \frac{1}{T} \sum_{\boldsymbol{x} \in \Pi} \sum_{t=1}^{T} \mu^t(\boldsymbol{x}) \left\langle \boldsymbol{x}, \tilde{\boldsymbol{u}}^t \right\rangle \\
&\le \frac{1}{T} \sum_{\boldsymbol{x} \in \Pi} \sum_{t > t_{\boldsymbol{x}}} \mu^t(\boldsymbol{x}) \left\langle \boldsymbol{x}, \tilde{\boldsymbol{u}}^t \right\rangle + \delta \\
&\le \frac{1}{T} \sum_{\boldsymbol{x} \in \Pi} \sum_{t > t_{\boldsymbol{x}}} \mu^t(\boldsymbol{x}) p_{\boldsymbol{x}}(\boldsymbol{u}^t) + \epsilon + \delta \\
&\le \frac{1}{T} \sum_{\boldsymbol{x} \in \Pi} \sum_{t=1}^{T} \mu^t(\boldsymbol{x}) p_{\boldsymbol{x}}(\boldsymbol{u}^t) + \epsilon + 2\delta = \bar{V}(\mathrm{Id}) + \epsilon + 2\delta,
\end{aligned} \tag{8.12}$$

where the first and third inequalities use Lemma 8.63, and the second inequality uses Lemma 8.64.

For the second, conditional on the event in Lemma 8.62, we have

$$\begin{aligned}
V(\phi) &= \frac{1}{T}\sum_{\boldsymbol{x}\in\Pi}\sum_{t=1}^{T}\mu^t(\boldsymbol{x})\left\langle\phi(\boldsymbol{x}),\tilde{\boldsymbol{u}}^t\right\rangle \\
&\geq \frac{1}{T}\sum_{\boldsymbol{x}\in\Pi}\sum_{t>t_{\boldsymbol{x}}}\mu^t(\boldsymbol{x})\left\langle\phi(\boldsymbol{x}),\tilde{\boldsymbol{u}}^t\right\rangle - \delta \\
&\geq \frac{1}{T}\sum_{\boldsymbol{x}\in\Pi}\sum_{t>t_{\boldsymbol{x}}}\mu^t(\boldsymbol{x})\Pr_{\boldsymbol{a}\sim p_{\boldsymbol{x}}}[\bar{\phi}(\boldsymbol{a})=\boldsymbol{u}^t] - \epsilon - \delta \\
&\geq \sum_{\boldsymbol{x}\in\Pi}\frac{1}{T}\sum_{t=1}^{T}\mu^t(\boldsymbol{x})\Pr_{\boldsymbol{a}\sim p_{\boldsymbol{x}}}[\bar{\phi}(\boldsymbol{a})=\boldsymbol{u}^t] - \epsilon - 2\delta = \bar{V}(\bar{\phi}) - \epsilon - 2\delta,
\end{aligned}$$

where the first and third inequalities use Lemma 8.63. To establish the second inequality above, we note that,

$$\left\langle\phi(\boldsymbol{x}),\tilde{\boldsymbol{u}}^t\right\rangle = \mathop{\mathbb{E}}_{\boldsymbol{a}\sim p_{\boldsymbol{x}}}\left\langle\psi(\bar{\phi}(\boldsymbol{a})),\tilde{\boldsymbol{u}}^t\right\rangle \geq \Pr_{\boldsymbol{a}\sim p_{\boldsymbol{x}}}[\bar{\phi}(\boldsymbol{a})=\boldsymbol{u}^t] - \epsilon$$

by Lemma 8.62, since $\bar{\phi}(\boldsymbol{a}),\tilde{\boldsymbol{u}}^t \in \mathcal{A}$. Thus, accounting for the probability δ in which Lemma 8.62 fails, we have

$$\mathbb{E}[V(\phi)-\bar{V}(\bar{\phi})] = \underbrace{\mathbb{E}[V(\phi)-\bar{V}(\bar{\phi})|F]}_{\geq-\epsilon-2\delta}\cdot\underbrace{\Pr[F]}_{\leq 1}+\underbrace{\mathbb{E}[V(\phi)-\bar{V}(\bar{\phi})|\neg F]}_{\geq -1}\cdot\underbrace{\Pr[\neg F]}_{\leq\delta} \geq -\epsilon-3\delta \tag{8.13}$$

where F is the event in Lemma 8.62. Combining (8.12) and (8.13),

$$\mathbb{E}[V(\phi)-V(\mathrm{Id})] \geq \mathbb{E}\left[\bar{V}(\bar{\phi})-\bar{V}(\mathrm{Id})\right] - 2\epsilon - 5\delta \geq \frac{1}{C\ell^6} - 2\epsilon - 5\delta \geq \frac{1}{4C\ell^6} = \epsilon$$

by setting the parameters

$$\epsilon = \frac{1}{4C\ell^6}, \quad \text{and} \quad n = \frac{2\log 20CN^2d^6}{\epsilon^2} \quad \text{so that} \quad 5\delta = 5N^2e^{-n\epsilon^2/2} \leq \frac{1}{4C\ell^6}.$$

The resulting tree-form decision problem hence has dimension $d = \ell\cdot n = O(\log^{14}N)$, and since $\ell = \Theta(\epsilon^{-1/6}) \leq O(\log T)$ we have that the swap regret is at least ϵ for all $T < \min\left\{N/4, \exp(\Omega(\epsilon^{-1/6}))\right\}$, where $N = \exp(\Omega(d^{1/14}))$, as desired. □

Finally, we conclude by providing a lower bound that matches our upper bound (Theorem 8.38) up to a constant factor in the exponent of k. We first observe that that there is a simple way to parameterize the above lower bound in terms of the dimension of the set of deviations:

[8.12] Technically ϕ is a random variable dependent on $\boldsymbol{u}^1,\ldots,\boldsymbol{u}^T$.

[8.13] Note that if a profitable deviation $\Pi\to\mathcal{X}$ exists, then by linearity, so must a profitable deviation $\Pi\to\Pi$.

Corollary 8.65. *Consider a learner operating on the simplex $\Delta(\mathcal{A})$. There is a k-dimensional set of deviations $\Phi \subseteq \Delta(\mathcal{A})^{\Delta(\mathcal{A})}$ such that for any $T < \sqrt{k}/4$, there is an adversary that forces the Φ-regret of the learner to be $\Omega(\log^{-6} T)$.*

Indeed, one can first identify an arbitrary subset $\mathcal{A}'$ of $\mathcal{A}$ with cardinality $\sqrt{k}$, and then employ the adversary of Theorem 8.14 with respect to $\mathcal{A}'$ while rendering all other actions dominated by assigning to them very small utility. That Φ is k-dimensional in this case follows because the set of stochastic matrices mapping $\Delta(\mathcal{A}')$ to $\Delta(\mathcal{A}')$—which contains all relevant swap deviations—is $(\sqrt{k})^2$-dimensional.

Combining Corollary 8.65 with Theorem 8.60, we have proven Theorem 8.5.

8.11 Discussion

In this section, we discuss several remarks about the framework and results that we have introduced.

8.11.1 Strict Hierarchy of Equilibrium Concepts

Let Ψ^ℓ be the set of degree-ℓ polynomial maps on strategy set $\Pi = \{0, 1\}^d$. For every $\ell \geq 0$, let $\mathcal{S}^\ell(\Gamma)$ be the set of Ψ^ℓ-equilibria in Γ. It is clear from definitions that $\mathcal{S}^\ell(\Gamma) \subseteq \mathcal{S}^{\ell-1}(\Gamma)$. Further, even for normal-form games, it is known that coarse-correlated equilibria are not generally equivalent to correlated equilibria, so at least one of these inclusions is strict in some games. We now show that *all* of these inclusions are strict, so that the deviations Ψ^ℓ form a *strict* hierarchy of equilibria.[8.14]

Proposition 8.66. *For every $\ell \geq 1$, there exists a game Γ such that $\mathcal{S}^\ell(\Gamma) \subsetneq \mathcal{S}^{\ell-1}(\Gamma)$.*

Proof. Consider the two-player game Γ defined as follows.

- P1's strategy space is $\mathcal{X} = [-1, 1]^\ell$. Player 2's strategy space is simply $\mathcal{Y} = [-1, 1]$.
- P1's utility function is $u_1(\boldsymbol{x}, y) = x_1 y$. That is, P1 would like to set $x_1 = y$. P2 gets no utility.

Consider the correlated profile μ defined as follows: μ is uniform over the 2^ℓ pure profiles $(\boldsymbol{x}, y) \in \mathcal{X} \times \mathcal{Y}$ such that $y = x_1 x_2 \dots x_\ell$. P1's expected utility is clearly 0, and there is a swap (*i.e.*, Ψ^ℓ) deviation that yields a profit of 1, namely $\boldsymbol{x} \mapsto (x_1 x_2 \dots x_\ell, \dots)$. (it does not matter what the swap deviation plays at coordinates other than the first one.) But, since all the x_is are independent, no function of degree less than ℓ can have positive correlation with $x_1 x_2 \dots x_\ell$,

[8.14] The below result constructs a game that depends on ℓ. It is *not* the case that there exists a single game for which the inclusion hierarchy is strict: for example, when $\Pi = \{0, 1\}^d$, for $\ell \geq d$, the set Ψ^ℓ will already contain all the swap deviations, so $\mathcal{S}^\ell(\Gamma) = \mathcal{S}^d(\Gamma)$ for every $\ell \geq d$.

and thus, there are no profitable deviations of degree less than ℓ. Thus, μ is a $\Psi^{\ell-1}$-equilibrium, but not a Ψ^{ℓ}-equilibrium. □

8.11.2 Characterization of Recent Low-Swap-Regret Algorithms in Our Framework

We have, throughout this chapter, introduced and used a framework of Φ-regret that involves fixed points in expectation. Proposition 8.59 shows that the ability to compute fixed points in expectation is in some sense necessary for the ability to minimize Φ-regret. It is instructive to briefly discuss how the recent swap-regret-minimizing algorithm of Dagan et al. [71] and Peng and Rubinstein [242] fits into this framework. Their algorithm makes no explicit reference to fixed-point computation, nor to the minimization of external regret over swap deviations ϕ—they do not explicitly invoke the framework we use in this chapter, nor that of Gordon et al. [132]. Where is the expected fixed point hidden, then? While we will not present their entire construction here, it suffices to state the following property of it. At every round t, the learner outputs a distribution $\mu^{(t)} \in \Delta(\mathcal{X})$ that is uniform on L strategies $\boldsymbol{x}^{(t,1)}, \dots, \boldsymbol{x}^{(t,L)}$. The way to map this into our framework is to consider $\mu^{(t)}$ an approximate fixed point in expectation of the "function"[8.15] $\phi^{(t)}$ that maps $\boldsymbol{x}^{(t,\ell)} \mapsto \boldsymbol{x}^{(t,\ell+1)}$ for each $\ell = 1, \dots, L-1$. With this choice of $\phi^{(t)}$, their algorithm indeed fits into our framework.

8.11.3 Representation of Strategies

In this section, we discuss how mixed strategy profiles $\mu \in \Delta(\mathcal{X}_1 \times \cdots \times \mathcal{X}_n)$ are represented for the purposes of all of the results in this chapter. In both cases, at each timestep, each player's strategy $\mu_i^{(t)}$ is a mixture of at most $L = \operatorname{poly}(d, \log(1/\epsilon))$ strategies $\boldsymbol{x}_i^{(t,1)}, \dots, \boldsymbol{x}_i^{(t,L)}$. That is,

$$\mu = \sum_{t=1}^{T} \lambda^{(t)} \bigtimes_{i=1}^{n} \left(\sum_{\ell=1}^{L} \lambda_i^{(t,\ell)} \delta(\boldsymbol{x}_i^{(t,\ell)}) \right),$$

where $\lambda^{(t)}$ and $\lambda_i^{(t,\ell)}$ are weights satisfying $\sum_{t=1}^{T} \lambda^{(t)} = 1$, and $\sum_{\ell=1}^{L} \lambda_i^{(t,\ell)} = 1$ for every t and every i.

If we impose a slightly more stringent restriction on the output format, namely, we want μ to be a uniform mixture of products of mixtures of *pure* strategies, we can use the Carathéodory construction from Section 8.4.1 to force the $\boldsymbol{x}_i^{(t,\ell)}$s to be pure strategies without loss of generality. Now, writing $\beta(\boldsymbol{x}_i^{(t,\ell)}) = \sum_{j \in [N]} \lambda_i^{(t,\ell,j)} \delta(\boldsymbol{x}_i^{(t,\ell,j)})$ for $\boldsymbol{x}_i^{(t,\ell,j)} \in \Pi_i$, where $N = d + 1$ for the Carathéodory map, we set

$$\mu = \sum_{t=1}^{T} \lambda^{(t)} \bigtimes_{i=1}^{n} \left(\sum_{\ell=1}^{L} \lambda_i^{(t,\ell)} \sum_{j=1}^{N} \lambda_i^{(t,\ell,j)} \delta(\boldsymbol{x}_i^{(t,\ell,j)}) \right).$$

[8.15] "Function" is in quotes because the stated ϕ may not be a function at all; for example, the sequence $\boldsymbol{x}^{(t,1)}, \dots, \boldsymbol{x}^{(t,L)}$ may contain repeats yet be aperiodic.

Note that we *cannot* convert these to mixtures of products of pure straegies (*i.e.*, $\mu = \sum_t \lambda^{(t)} \delta(\boldsymbol{z}^{(t)})$ with $\boldsymbol{z}^{(t)} \in \Pi_1 \times \cdots \times \Pi_n$) without incurring an exponential blowup.

8.12 Conclusions and Open Problems

In summary, we established efficient algorithms for minimizing Φ-regret and computing Φ-equilibria with respect to any set of deviations with a polynomial dimension. For the online learning setting, our upper bounds are tight up to constant factors in the exponents, crystallizing for the first time a family of deviations that characterizes the learnability of Φ-regret.

There are many important avenues for future research. First, we did not attempt to optimize the (polynomial) dependence of the running time (in Theorems 8.34 and 8.38) on k and d; improving the overall complexity of our algorithms is an interesting direction. Moreover, developing more practical algorithms—that refrain from using ellipsoid—would also be a valuable contribution. In particular, are there polynomial-time algorithms for computing Φ-equilibria without resorting to the EAH framework? But the most pressing open question is to understand the complexity of computing (normal-form) correlated equilibria in the centralized model.

Chapter 9

Game Theory and Variational Inequalities

9.1 Introduction

In this chapter, we will generalize several of the ideas we have explored so far even further—beyond games—to optimization problems, specifically *variational inequalities (VIs)*. VIs are a mainstay framework at the heart of optimization that unifies a host of foundational problems in diverse areas ranging from engineering to economics [22, 93, 174]. The basic problem underpinning VIs—when restricted to Euclidean spaces—can be formulated as follows.

Definition 9.1 (SVIs)**.** Let $\mathcal{X}$ be a convex and compact subset of $\mathbb{R}^d$ and a (single-valued) mapping $F : \mathcal{X} \to \mathbb{R}^d$. The ϵ-approximate *Stampacchia variational inequality (SVI)* problem[9.1] asks for a point $\boldsymbol{x} \in \mathcal{X}$ such that

$$\langle F(\boldsymbol{x}), \boldsymbol{x}' - \boldsymbol{x} \rangle \geq -\epsilon \quad \forall \boldsymbol{x}' \in \mathcal{X}. \tag{9.1}$$

Assuming F is continuous, an exact SVI solution (that is, with $\epsilon = 0$) always exists [93]—by Brouwer's fixed-point theorem applied on the function $\boldsymbol{x} \mapsto \Pi_{\mathcal{X}}(\boldsymbol{x} - \eta F(\boldsymbol{x}))$, where $\Pi_{\mathcal{X}}$ is the (Euclidean) projection mapping. For computational purposes, we have introduced a slackness $\epsilon > 0$ in the right-hand side of (9.1); without this relaxation, the only SVI solution can be irrational. It is assumed that ϵ is given (in binary) as part of the input, and the goal is to design algorithms with complexity growing polynomially in $\log(1/\epsilon)$ and the dimension d.

A canonical example of Definition 9.1 is the problem of computing fixed points of gradient descent—that is, first-order optima—in constrained optimization. In particular, for a differentiable function f, the ϵ-SVI problem corresponding to $F := \nabla f$ can be expressed as $\langle \nabla f(\boldsymbol{x}), \boldsymbol{x}' - \boldsymbol{x} \rangle \geq -\epsilon$ for all $\boldsymbol{x}' \in \mathcal{X}$, which captures precisely the first-order optimality conditions [32]. Another standard example of Definition 9.1 is the problem of computing (approximate) *Nash equilibria* in multi-player games [230]: the problem of computing a Nash equilibrium in a multilinear game can be expressed as a VI by setting $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$ to be the joint strategy set, and $F(\boldsymbol{x}) = (-\nabla_{\boldsymbol{x}_1} u_1(\boldsymbol{x}), \dots, -\nabla_{\boldsymbol{x}_n} u_n(\boldsymbol{x}))$.

[9.1] It is common to refer to SVIs as simply VIs, but we adopt the former nomenclature here to disambiguate with the *Minty VI* problem, introduced in Definition 9.2.

Unfortunately, by virtue of encompassing (approximate) Nash equilibria, solving general SVIs is computationally intractable—namely, PPAD-hard [236]—even when F is linear and ϵ is an absolute constant [259]; recent work by Bernasconi et al. [26], Kapron and Samieefar [170] characterizes the exact complexity of SVIs, as well as generalizations thereof. In fact, even in the special case discussed above wherein $F := \nabla f$, computing ϵ-SVI solutions is also hard—under certain well-believed complexity assumptions—when ϵ is exponentially small in the dimension [108].

In light of the intractability of solving general SVIs, most research has restricted its attention to more structured classes of problems. Following a long and burgeoning line of work that goes back to the 1960s, we operate under the *Minty condition* (introduced below as Assumption 9.3); it is based on a variant of SVIs (Definition 9.1) tracing back to Minty [217], known as the *Minty* VI problem.

Definition 9.2 (MVIs)**.** Let $\mathcal{X}$ be a convex and compact subset of $\mathbb{R}^d$ and a mapping $F : \mathcal{X} \to \mathbb{R}^d$. The *Minty variational inequality (MVI)* problem asks for a point $\boldsymbol{x} \in \mathcal{X}$ such that

$$\langle F(\boldsymbol{x}'), \boldsymbol{x}' - \boldsymbol{x} \rangle \geq 0 \quad \forall \boldsymbol{x}' \in \mathcal{X}. \tag{9.2}$$

Unlike SVIs, an MVI solution may not exist even when F is continuous—indeed, the Minty condition is precisely the requirement that an MVI solution exists:

Assumption 9.3 (Minty condition)**.** A variational inequality problem $\mathrm{VI}(\mathcal{X}, F)$ satisfies the *Minty condition* if there exists $\boldsymbol{x} \in \mathcal{X}$ that satisfies (9.2).

Assuming continuity, Definition 9.2 refines Definition 9.1 in the following sense.

Lemma 9.4 (Minty's lemma)**.** *If F is continuous and $\mathcal{X}$ is convex and compact, then any MVI solution is also an SVI solution.*

On the other hand, an SVI solution need not be an MVI solution—otherwise Assumption 9.3 would always hold under a continuous mapping. One well-known special case in which the set of MVI solutions does coincide with the set of SVI solutions is when F is *monotone*; that is, when $\langle F(\boldsymbol{x}) - F(\boldsymbol{x}'), \boldsymbol{x} - \boldsymbol{x}' \rangle \geq 0$ for all $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$. This holds more generally when F is *pseudomonotone*, which means that $\langle F(\boldsymbol{x}'), \boldsymbol{x} - \boldsymbol{x}' \rangle \geq 0 \implies \langle F(\boldsymbol{x}), \boldsymbol{x} - \boldsymbol{x}' \rangle \geq 0$ for all $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$. Importantly, the Minty condition is more permissive than monotonicity, encompassing a broader class of problems; indeed, it captures a host of nonconvex optimization problems (*cf.* Section 9.3).

By now, it is well-known that under the Minty condition, there are algorithms with complexity scaling polynomially in $1/\epsilon$ (and all other natural parameters of the problem) for computing an ϵ-SVI solution. This goes back to the early, pioneering work of Sibony [270], Martinet [210], and Korpelevich [180]; in particular, the latter showed that the extra-gradient method converges in all monotone VIs—this stands in stark contrast to the usual gradient descent algorithm, which can cycle even in two-player zero-sum games [215], which are monotone. Motivated by many applications in machine learning and reinforcement learning, there has been renewed interest in the Minty condition recently (*e.g.*, [42, 48, 78, 195, 197, 213, 214, 241, 274, 298]). However,

	Result	Reference
Upper bounds	• ϵ-SVIs under the Minty condition • ϵ-MVIs under a monotone F • ϵ-SVIs *or* $\Omega_\epsilon(\epsilon^2)$-strict EVIs • ϵ-global optimization under quasar-convexity† • ϵ-Nash equilibria in harmonic games • ϵ-Nash *or* $\Omega_\epsilon(\epsilon^4)$-strict CCEs in two-player concave games	Theorem 9.5 Corollary 9.6 Theorem 9.8 Theorem 9.10 Corollary 9.11 Theorem 9.12
Lower bounds	• Deciding MVI feasibility (*i.e.*, Minty condition) • Solving MVIs under the Minty condition	Theorem 9.13 Proposition 9.15

Table 9.1: *Summary of main results. All algorithms (upper bounds) have runtimes that depend polynomially on both d and $\log(1/\epsilon)$, and, to our knowledge, are the first algorithms for their respective settings with this dependence. †: Unlike our other positive results, our result for quasar-convex optimization holds even when $F = \nabla f$ is not Lipschitz continuous.*

those existing results become vacuous when ϵ is exponentially small in terms of the dimension d. On the other end of the spectrum, all existing algorithms with $\mathsf{poly}(d, \log(1/\epsilon))$ complexity make restrictive assumptions that are significantly stronger than the Minty condition, akin to *strong* monotonicity or some type of an "error bound" (for example, we refer to [203, 205, 274, 298], discussed further in Section 8.1.3).

9.1.1 Our results

We establish the first polynomial-time algorithm for solving ϵ-SVIs under the Minty condition. In what follows, we assume that the constraint set $\mathcal{X}$ is accessed through a (weak) separation oracle (Definition 9.18) and $\mathcal{B}_1(\mathbf{0}) \subseteq \mathcal{X} \subseteq \mathcal{B}_R(\mathbf{0})$ for some $R \leq \mathsf{poly}(d)$; the latter can be met be bringing $\mathcal{X}$ into isotropic position, a transformation that does not affect our main result (as formalized in the appendix of the full paper [13]). With regard to the mapping F, we assume that it can be evaluated in polynomial time, it is L-Lipschitz continuous, and $\|F(\boldsymbol{x})\|$ is bounded by $B > 0$ (Assumption 9.20). We are now ready to state our main result.

Theorem 9.5 (Precise version in Theorem 9.46)**.** *Under the Minty condition (Assumption 9.3), there is an algorithm that runs in $\mathsf{poly}(d, \log(B/\epsilon), \log L)$ time and returns an ϵ-SVI solution.*

Compared to all previous results cited earlier under the Minty condition, this result improves exponentially the dependence on $1/\epsilon$ and L. We clarify that, although the underlying algorithm posits the Minty condition, its execution does not hinge on knowing an MVI solution. Indeed, a peculiar feature of Theorem 9.5 is that while its main precondition concerns MVIs, the output itself is an approximate SVI solution; the reason for this discrepancy will become clear shortly.

Furthermore, Theorem 9.5 implies the first, to our knowledge, polynomial-time algorithm for solving monotone VIs—perhaps the most well-studied structural assumption in the literature.

Corollary 9.6. *There is an algorithm that runs in* $\mathsf{poly}(d, \log(B/\epsilon), \log L)$ *time and returns an* ϵ*-MVI solution when* F *is monotone.*

Indeed, as we saw earlier, monotonicity implies that the set of MVI solutions coincides with the set of SVI solutions, so Corollary 9.6 directly follows from Theorem 9.5. Before we present some more results that build on Theorem 9.5 (gathered in Table 9.1), we dive into our technical approach.

9.1.1.1 Technical Approach: Proof of Theorem 9.5

The proof of Theorem 9.5 relies on the usual (central-cut) ellipsoid algorithm, but with certain unusual twists.

Challenge 1: lack of convexity. The first immediate conundrum lies in the fact that the set of SVI solutions is generally not convex even when the Minty condition holds. By contrast, the set of MVI solutions is convex, thereby being a better candidate on which to execute ellipsoid. But this approach also runs into an apparent difficulty: while a point $\boldsymbol{x} \in \mathcal{X}$ can be confirmed to be an ϵ-SVI solution by invoking a (linear) optimization oracle, this is not so for MVIs. Indeed, as we show later (Section 9.1.1.4), ascertaining MVI membership is coNP-complete.

In summary, the set of SVI solutions admits an efficient membership oracle, while the set of MVI solutions is convex. A natural approach now presents itself: execute the ellipsoid with respect to the set of MVIs, but only verify SVI membership. To do so, the first key idea is this: if a point $\boldsymbol{x} \in \mathcal{X}$ is *not* an ϵ-SVI solution, then $F(\boldsymbol{x})$ yields a hyperplane separating $\boldsymbol{x}$ from the set of MVIs. As a result, this allows us to run the ellipsoid algorithm with respect to the convex set of MVI solutions, with the peculiarity that we only test for SVI membership during the execution of the algorithm. So long as the algorithm fails to identify an approximate SVI solution, the volume of the ellipsoid shrinks geometrically.

Challenge 2: lack of full dimensionality. This now brings us to the next key challenge: the set of MVI solutions is generally not fully dimensional. It is therefore unclear whether the volume of the ellipsoid can be used as a yardstick to measure the progress of the algorithm. There is a standard approach for addressing this issue, at least for rational polyderal sets (for example, [138, Chapter 6]): restrict the execution of the ellipsoid to suitable subspaces whenever one of the ellipsoid's axis gets too small. However, this standard approach falls short in our problem; we provide a concrete numerical example in Section 9.7.3 illustrating that the usual ellipsoid algorithm fails to identify ϵ-SVI solutions.

To address this problem, we introduce a new algorithmic idea. Namely, we show that we can produce, in polynomial time, what we refer to as a *strict* separating hyperplane; this key ingredient underpinning Theorem 9.5 is summarized below (the precise version is Lemma 9.44). (Technically,

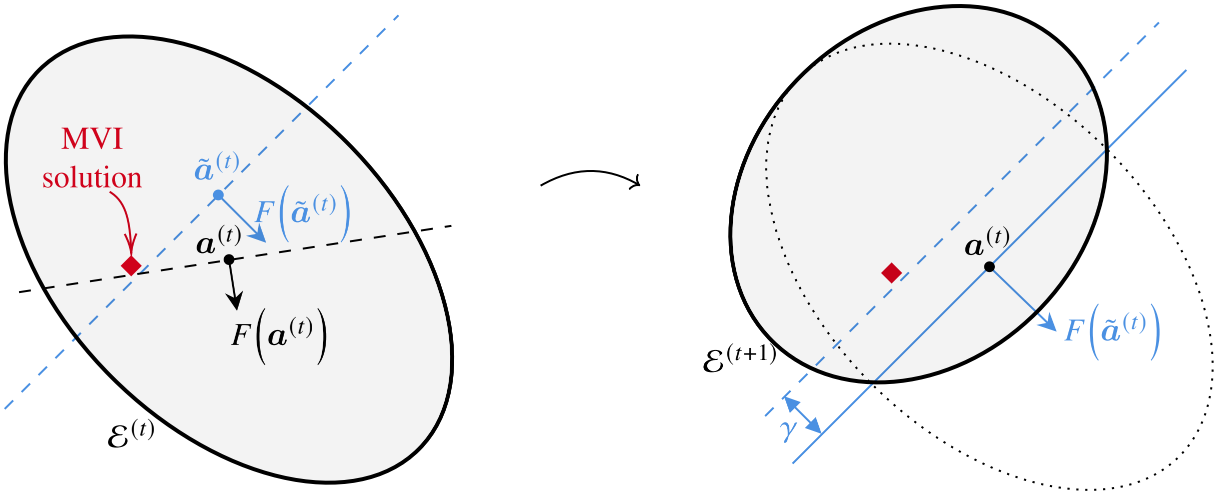

MVI
solution
$\tilde{\boldsymbol{a}}^{(t)}$
$F\left(\tilde{\boldsymbol{a}}^{(t)}\right)$
$\boldsymbol{a}^{(t)}$
$F\left(\boldsymbol{a}^{(t)}\right)$
$\mathcal{E}^{(t)}$
$\boldsymbol{a}^{(t)}$
$\mathcal{E}^{(t+1)}$
$F\left(\tilde{\boldsymbol{a}}^{(t)}\right)$
γ

Figure 9.2: *One step of our ellipsoid algorithm—when the current ellipsoid center $\boldsymbol{a}^{(t)}$ is not already an ϵ-SVI solution. While $F(\boldsymbol{a}^{(t)})$ separates $\boldsymbol{a}^{(t)}$ from the set of MVI solutions (in this case a single point), $F(\tilde{\boldsymbol{a}}^{(t)})$ yields a γ-*strict *separating hyperplane, which turns out to be crucial; see Section 9.7.3.*

we need to allow $\boldsymbol{a}$ to be approximately in $\mathcal{X}$ since we are dealing with general convex sets, but we do not dwell on this issue in the introduction.)

Lemma 9.7 (SVI membership or MVI *strict* separation)**.** *Given a point $\boldsymbol{a} \in \mathcal{X} \cap \mathbb{Q}^d$ and $\epsilon \in \mathbb{Q}_{>0}$, there is a polynomial-time algorithm that either*

1. *ascertains that $\boldsymbol{a}$ is an ϵ-SVI solution or*
2. *returns $\mathbf{c} \in \mathbb{Q}^d$, with $\|\mathbf{c}\|_\infty = 1$, such that $\langle \mathbf{c}, \boldsymbol{x} \rangle \leq \langle \mathbf{c}, \boldsymbol{a} \rangle - \gamma$ for any point $\boldsymbol{x} \in \mathcal{X}$ that satisfies the Minty VI (9.2), where $\gamma = \epsilon^2 \cdot \mathsf{poly}(R^{-1}, L^{-1}, B^{-1})$.*

To obtain a hyperplane that *strictly* separates $\boldsymbol{a} \in \mathcal{X}$ from the set of MVI solutions (in the sense of Item 2), assuming that $\boldsymbol{a}$ is not an ϵ-SVI solution, we perform a gradient descent step starting from $\boldsymbol{a}$. Since $\boldsymbol{a}$ is not an ϵ-SVI solution, it can be shown that the resulting point, say $\tilde{\boldsymbol{a}} \in \mathcal{X}$, is such that $F(\tilde{\boldsymbol{a}})$ strictly separates $\boldsymbol{a}$ from the set of MVIs (Lemma 9.44); this is illustrated in Figure 9.2.

Interestingly, this simple algorithmic maneuver is, at least conceptually, similar to the extra-gradient method [180] (and variants thereof), which is known to converge—albeit at an inferior rate that grows polynomially in $1/\epsilon$—under the Minty condition. Accordingly, we call the overall algorithm ExtraGradientEllipsoid; it is an incarnation of the central-cut ellipsoid endowed with additional gradient descent steps. We stress again that this step cannot be avoided: the usual ellipsoid algorithm without the extra-gradient step fails (Section 9.7.3) due to its inability to generate strict separating hyperplanes. This phenomenon mirrors the behavior of first-order methods, whereby regular gradient descent fails to converge to an SVI solution—even in monotone problems—whereas the extra-gradient method succeeds [180]. As such, we find that there is an intriguing analogy between the behavior we uncover for ellipsoid-based algorithms and what has been known for decades pertaining to gradient-based algorithms.

To finish the proof of Theorem 9.5, we observe that Lemma 9.7—and in particular the sequence of strict separating hyperplanes produced during the execution of the ellipsoid—implies that the volume of the ellipsoid cannot shrink too much (Lemma 9.45). Indeed, since the ellipsoid always contains the set of MVI solutions, and the separating hyperplanes are *strict*, every MVI solution is in some sense *far* from the boundary of the ellipsoid, which in turn implies that the ellipsoid must have nontrivial volume. In other words, the algorithm will necessarily terminate with an ϵ-SVI solution, as promised—so long as an MVI solution exists.

If no MVI solution exists, the above algorithm might fail, that is, the volume may end up shrinking too much. But in this case, as we shall see, a small adaptation to ExtraGradientEllipsoid can produce a polynomial *certificate of MVI infeasibility* (*cf.* Theorem 9.8).

9.1.1.2 A certificate of MVI infeasibility

We now treat the general setting in which the Minty condition can be altogether violated. Our next result shows how to employ ExtraGradientEllipsoid so as to produce a certificate of MVI infeasibility. But how does such a "certificate" look like?

To answer this, we rely on a novel notion of *expected VIs (EVIs)* (Definition 9.29). This is a relaxation of Definition 9.1 that only imposes the SVI constraint *in expectation* for points draw from a distribution. For readers familiar with equilibrium concepts in game theory, it is instructive to have in mind that EVIs are to SVIs what *(average) coarse correlated equilibria (ACCEs)* (Definition 9.31) are to Nash equilibria. The key point is that there is a strong duality between MVIs and EVIs (Proposition 9.30); namely, the Minty condition holds if and only if no EVI solution with *negative gap* exists—we refer to the latter object as a *strict* EVI solution. In other words, the mere existence of a strict EVI exposes MVI infeasibility. This brings us to the following key refinement of Theorem 9.5.

Theorem 9.8 (SVI or strict EVI; precise version in Theorem 9.55)**.** *There is an algorithm that runs in* $\mathsf{poly}(d, \log(B/\epsilon), \log L)$ *time and returns either*

1. *an ϵ-SVI solution or*
2. *an $\Omega_\epsilon(\epsilon^2)$-strict EVI solution.*

This clearly strengthens Theorem 9.5: under the Minty condition no strict EVIs exist, so Item 2 in Theorem 9.8 will never arise under Assumption 9.3. A key reference point here is a result by Anagnostides et al. [12], who provided an algorithm with a similar output guarantee, but with complexity scaling polynomially—rather than logarithmically—in $1/\epsilon$; as such, Theorem 9.8 yields again an exponential improvement over existing results.

The proof of Theorem 9.8 is based on an application of duality between MVIs and EVIs. In particular, the minimax theorem implies that, once the volume of the ellipsoid becomes sufficiently small, there is a distribution supported on $\tilde{\boldsymbol{a}}^{(1)}, \dots, \tilde{\boldsymbol{a}}^{(T)}$ that is a $\gamma/2$-strict EVI, where γ is the strictness parameter per Lemma 9.7 and $\tilde{\boldsymbol{\alpha}}^{(t)}$ is obtained after a gradient descent step starting from the center of the ellipsoid at the tth iteration (Figure 9.2); coupled with Lemma 9.7, this explains the $\Omega_\epsilon(\gamma) = \Omega_\epsilon(\epsilon^2)$ strictness in Item 2. We are thus left with the simple problem of optimizing the mixing weights of a distribution with a polynomial support. This algorithmic maneuver closely resembles the celebrated "ellipsoid against hope" algorithm of Papadimitriou and Roughgarden [237] (*cf.* [81, 96, 159]), which is also based on running ellipsoid on an infeasible program.

A strict EVI, besides certifying MVI infeasibility, is an interesting object in its own right. It is, by definition, a solution concept with negative gap, thereby being particularly stable—any possible deviation is not just suboptimal, but significantly so. Indeed, its incarnation in the context of games has been already used to address the equilibrium selection problem, as we explain more in Section 8.1.3. Furthermore, in certain applications, the EVI gap translates to a performance guarantee in terms of some underlying objective function; the *smoothness* framework of Roughgarden [256]—and its extension for general VI problems given in Definition 9.26—is a prime example of this in the context of multi-player games. It turns out that an EVI with a negative gap yields a *strict* improvement over the bound predicted by the smoothness framework.

But there is something more that is especially notable about Theorem 9.8: each of the computational problems in Items 1 and 2 is computationally hard on its own, yet Theorem 9.8 shows that their disjunction is easy! In particular, computing ϵ-SVI solutions is a well-known PPAD-complete problem. With regard to strict EVIs, we provide a characterization of its complexity in this chapter, establishing NP-completeness (Theorem 9.13).[9.2] To put this into context, we highlight that there has been interest in characterizing the complexity of the union of two problems, especially in the realm of TFNP. A notable contribution here is the work of Daskalakis and Papadimitriou [76] that examined the complexity of problems in PPAD ∩ PLS, where PLS stands for "polynomial local search" [164]. They observed that if a problem A is PPAD-complete and B is PLS-complete, then the problem that must return *either* a solution to A or to B is PPAD ∩ PLS-complete—that is, CLS-complete [108]; unlike Theorem 9.8, this observation by Daskalakis and Papadimitriou [76] assumes that A and B are defined with respect to different instances. In this context, Theorem 9.8 provides an example in which the disjunction of two hard problems—one PPAD-complete and the other NP-complete—defined with respect to the *same* instance is easy.

9.1.1.3 Implications and Extensions

Moving forward, we discuss several important consequences and extensions of our main results for optimization and game theory.

Quasar-convex optimization. The first implication concerns optimizing *quasar-convex* functions; this is a relaxation of convexity that has attracted significant interest recently (for example, [53, 72, 115, 133, 141, 147, 294]).

(We elaborate more on how this relates to other properties in Section 9.3.1.) Not only does $\mathrm{VI}(\mathcal{X}, \nabla f)$ (under Definition 9.24) satisfy the Minty condition, but every approximate SVI solu-

[9.2] Strict EVIs are dual to MVI solutions; Theorem 9.13 shows that deciding MVI feasibility is coNP-complete, so deciding strict EVI existence is NP-complete.

tion is also an approximate global minimum of f (Proposition 9.25); combined with Theorem 9.5, we obtain the first polynomial-time algorithm for globally minimizing smooth quasar-convex functions.

Corollary 9.9. *There is a* $\mathsf{poly}(d, \log(B/\epsilon), \log(1/\lambda), \log L)$*-time algorithm that outputs a point* $\boldsymbol{x} \in \mathcal{X}$ *such that* $f(\boldsymbol{x}) \leq \min_{\boldsymbol{x}' \in \mathcal{X}} f(\boldsymbol{x}') + \epsilon$ *for any* λ*-quasar-convex function* f*.*

This result can be generalized (*cf.* Proposition 9.28) by considering a broader class of VI problems (Definition 9.26), beyond quasar-convex functions. Interestingly, this class of problems encompasses (a special case of) *smooth games*, famously introduced by Roughgarden [256]; we explain this in more detail in Section 9.3.1.

Furthermore, as we shall now see, Corollary 9.9 can be significantly strengthened by relaxing the assumption that ∇f is continuous (Theorem 9.10). This follows a long line of research in *nonsmooth* optimization (*e.g.*, [82, 165, 285, 314]). In this context, we show that under quasar-convexity—and, more generally, its extension based on Definition 9.26—it is possible to entirely eliminate the (logarithmic) dependence on L.

Theorem 9.10 (Precise version in Theorem 9.54)**.** *There is a* $\mathsf{poly}(d, \log(B/\epsilon), \log(1/\lambda))$*-time algorithm that outputs a point* $\boldsymbol{x} \in \mathcal{X}$ *such that* $f(\boldsymbol{x}) \leq \min_{\boldsymbol{x}' \in \mathcal{X}} f(\boldsymbol{x}') + \epsilon$ *for any* λ*-quasar-convex function* f*.*

(Since we are considering additive approximations, a dependence on B is necessary since one can always rescale f.) The key idea here is that quasar-convexity yields a strict separating hyperplane *without* requiring an extra-gradient step, which is where the Lipschitz continuity of F came into play in Lemma 9.7. Theorem 9.10 should be compared with the result of Lee and Valiant [193] pertaining to optimizing *star-convex* functions—the special case of Definition 9.24 where $\lambda = 1$.

Weak Minty condition. We also strengthen Theorem 9.5 along another axis. Perhaps the most immediate question is how far can one relax the assumption that $\text{VI}(\mathcal{X}, F)$ satisfies the Minty condition—while accepting the fact that, in light of the intractability of general VIs, imposing some assumptions is inevitable. Naturally, this question has received ample attention in contemporary research (*cf.* Section 9.4.3). One permissive condition that has emerged from that line of work is the *weak* Minty property put forward by Diakonikolas et al. [87] in the unconstrained setting. In Definition 9.49, we introduce a natural version of that notion for the constrained setting. And we show, in Theorem 9.52, that Theorem 9.5 can be applied in a certain regime of the weak Minty condition.

Harmonic games. We next discuss two implications and extensions of our main results for game theory. The first concerns *harmonic games* (Definition 9.36), a class of multi-player games at the heart of the seminal decomposition of Candogan et al. [52], covered in more detail in Section 9.3.3. Leveraging Theorem 9.5, we obtain the first polynomial-time algorithm for computing ϵ-Nash equilibria (per Definition 9.32, captured by ϵ-SVIs) in multi-player harmonic games under the polynomial expectation property [237]; this latter condition postulates that one can efficiently compute utility gradients (equivalently, the underlying mapping F can be evaluated efficiently), which holds in most succinct classes of games.

Corollary 9.11. *There is a polynomial-time algorithm for computing ϵ-Nash equilibria in (succinct) multi-player harmonic games under the polynomial expectation property.*

This algorithm is based on the observation that harmonic games satisfy a weighted version of the Minty condition. In particular, after applying a suitable transformation, we show that the induced VI problem satisfies the usual Minty condition under a Lipschitz continuous mapping (Proposition 9.38), at which point Corollary 9.11 follows from Theorem 9.5. Crucial to this argument is the fact that the weights in harmonic games cannot be too close to 0 (Lemma 9.37), for otherwise the Lipschitz continuity parameter would blow up; this issue is the crux in our refinement for two-player games, which is the subject of the next paragraph.

Nash or strict coarse correlated equilibria in two-player games. Our next application concerns equilibrium computation in two-player concave games. As we alluded to, EVIs are closely related to the notion of a coarse correlated equilibrium (CCE) from game theory. More precisely, when the underlying VI problem corresponds to a multi-player game, the EVI gap equates to the *sum* of the players' deviation benefits under a correlated distribution; this does not quite capture the usual notion of a CCE in which one bounds the *maximum* of the deviation benefits (Section 9.3.2). In light of this, we also provide the following refinement of Theorem 9.8 in two-player concave games.

Theorem 9.12 (Precise version in Theorem 9.57)**.** *In a two-player concave game, there is an algorithm that runs in time* $\mathsf{poly}(d, \log(B/\epsilon), \log L)$ *and returns either*

1. *an ϵ-Nash equilibrium or*
2. *an $\Omega_\epsilon(\epsilon^4)$-strict CCE.*

Unlike Theorem 9.8, which can be applied to multi-player games to find Nash or strict *ACCE*s, Theorem 9.57 cannot be extended to games with more than two players—one can always include a third player who always obtains zero utility. Theorem 9.12 provides an exponential improvement over a known result by Anagnostides et al. [11], who gave a $\mathsf{poly}(d, B, L, 1/\epsilon)$-time algorithm for the same problem via *optimistic mirror descent*.

As in harmonic games, the proof of Theorem 9.12 is based on analyzing a weighted version of the Minty condition—this allows transitioning from the sum to the maximum deviation benefit. But, unlike harmonic games, here we need to handle the case where one of the weights is arbitrarily close to 0, in which case the Lipschitz continuity parameter blows up, which in turn neutralizes the strict separation oracle of Lemma 9.7. We address this by providing a tailored separation oracle for two-player games when one of the weights gets too close to 0 (Lemma 9.59).

Expected VIs. In Section 9.6, we take a deeper look at the expected VI (EVI) problem in its own right. In particular, we will define an entire class of Φ-EVI problems that generalize the notion to Φ-equilibria in games to variational inequalities. We show that, at least when Φ contains only linear functions, Φ-EVIs are both efficiently computable and efficiently learnable, but—in a departure from the learnability of Φ-equilibria from Chapter 8—*not* efficiently computable when Φ contains nonlinear functions, even if they have constant degree. Finally, we use our notion of Φ-EVI to define a new notion of correlated equilibrium in games that we call the *anonymous (linear) correlated equilibrium* (ALCE). ALCEs are efficiently computable and learnable, but, surprisingly, lie somewhat outside the usual hierarchy of Φ-equilibria. We conclude by comparing our notion to the more usual notions of correlation that we have already introduced.

9.1.1.4 Lower Bounds

As promised, we complement Theorem 9.8 by proving that determining whether the Minty condition holds is coNP-complete.

Theorem 9.13 (Precise version in Proposition 9.86 and Theorems 9.87 and 9.89)**.** *Determining whether a VI problem satisfies the Minty condition (Assumption 9.3) is* coNP*-complete. Hardness holds even for two-player concave games or multi-player (succinct) normal-form games and even when a constant approximation error ϵ is allowed.*

Inclusion in coNP follows because a strict EVI solution is itself an efficiently verifiable witness of MVI infeasibility. On the other hand, for explicitly represented (normal-form) games, the duality between EVIs and MVIs (Proposition 9.30) enables us to show the following positive result.

Proposition 9.14. *There is a polynomial-time algorithm that determines whether an explicitly represented (normal-form) game satisfies the Minty condition.*

In particular, for such problems, it is well-known that one can efficiently optimize over the polytope of EVI solutions, so one can in particular minimize the equilibrium gap.

One final natural question that arises from Theorem 9.5 concerns the complexity of computing *Minty* VI solutions (per Definition 9.2), *when promised that such a solution exists*; by Lemma 9.4, this would be stronger than computing SVI solutions. With a straightforward construction, we observe that this is information-theoretically impossible.

Proposition 9.15 (Precise version in Propositions 9.92 and 9.94)**.** *Computing an ϵ-MVI solution requires $\Omega(1/\epsilon)$ oracle evaluations to F even when an MVI solution is guaranteed to exist and $d = 1$. When d is large, it requires $\Omega(2^d)$ oracle evaluations even when ϵ is an absolute constant.*

In particular, this lower bound implies that the dependence on λ in Theorem 9.10 and Corollary 9.9 cannot be removed (Proposition 9.92).

9.2 Preliminaries

Before moving on, we lay out some basic notation and background on optimization, and then proceed to formally state our blanket assumptions.

Notation. We denote by $\mathsf{size}(r)$ the *encoding length* of a rational number $r \in \mathbb{Q}$ in binary. For a matrix $\mathbf{A} \in \mathbb{R}^{d \times d'}$, $\|\mathbf{A}\|$ denotes its spectral norm.

We distinguish between a VI problem, denoted by $\mathrm{VI}(\mathcal{X}, F)$—which is given by a mapping $F : \mathcal{X} \to \mathbb{R}^d$ that can be evaluated in polynomial time (Assumption 9.20) and a constraint set that is implicitly accessed through a (weak) separation oracle (Definition 9.18)—and a solution thereof, be it an SVI (Definition 9.1) or an MVI (Definition 9.2).

Let $\mathcal{X}$ be a convex and compact set in $\mathbb{R}^d$. We define

$$\mathcal{X}^{-\epsilon} := \{\boldsymbol{x} \in \mathcal{X} : \mathcal{B}_\epsilon(\boldsymbol{x}) \subseteq \mathcal{X}\}, \tag{9.3}$$

where we recall that $\mathcal{B}_\epsilon(\boldsymbol{x})$ is the (closed) Euclidean ball centered at $\boldsymbol{x} \in \mathbb{R}^d$ with radius $\epsilon > 0$. (9.3) describes all points that are "ϵ-deep" inside $\mathcal{X}$. Similarly, we define

$$\mathcal{X}^{+\epsilon} := \{\boldsymbol{x}' \in \mathbb{R}^d : \|\boldsymbol{x}' - \boldsymbol{x}\| \le \epsilon \text{ for some } \boldsymbol{x} \in \mathcal{X}\} = \bigcup_{\boldsymbol{x} \in \mathcal{X}} \mathcal{B}_\epsilon(\boldsymbol{x}), \tag{9.4}$$

the set of all points that are "ϵ-close" to $\mathcal{X}$. Throughout this chapter, we work with a general convex set $\mathcal{X}$, which might even be supported solely on irrational points; it will thus be necessary to consider (9.3) and (9.4)—in place of $\mathcal{X}$—in the definitions that follow to ensure the existence of points with rational coordinates.

We say that $\mathcal{X}$ is in *isotropic position* if for a uniformly sampled $\boldsymbol{x} \sim \mathcal{X}$, we have $\mathbb{E}[\boldsymbol{x}] = 0$ and $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top] = \mathbf{I}_{d \times d}$. There is a polynomial-time algorithm that brings any constraint set $\mathcal{X}$ into isotropic position (for example, [202]). This transformation does not affect our main result (Theorem 9.46) or implications thereof, so we assume it without loss of generality. If $\mathcal{X}$ is in isotropic position, we can assume that $\mathcal{B}_1(\mathbf{0}) \subseteq \mathcal{X} \subseteq \mathcal{B}_R(\mathbf{0})$ for some $R \le \mathsf{poly}(d)$; in fact, all our positive results apply even when $R \le \exp(\mathsf{poly}(d))$.

Remark 9.16. We assume throughout that $\mathcal{X}$ is well-bounded, in that $\mathcal{B}_r(\boldsymbol{a}) \subseteq \mathcal{X} \subseteq \mathcal{B}_R(\mathbf{0})$, which in particular implies that $\mathcal{X}$ is fully dimensional. Under this assumption, it is known (for example, see Grötschel et al. [138, Lemma 3.2.35]) that if $\langle \boldsymbol{c}, \boldsymbol{x} \rangle \le \gamma$ holds for all $\boldsymbol{x} \in \mathcal{X}^{-\delta}$, it follows that

$$\langle \boldsymbol{c}, \boldsymbol{x} \rangle \le \gamma + \frac{2R}{r}\|\boldsymbol{c}\|\delta \quad \forall \boldsymbol{x} \in \mathcal{X}. \tag{9.5}$$

As a result, we can safely consider deviations in $\mathcal{X}$, not merely in $\mathcal{X}^{-\delta}$, by suitably rescaling the precision parameters—by virtue of (9.5).

Continuing with some basic geometric definitions, we say that a set $\mathcal{E} \subseteq \mathbb{R}^d$ is an *ellipsoid* if there exists a vector $\boldsymbol{a} \in \mathbb{R}^d$ and a positive definite matrix $\mathbf{A} \in \mathbb{R}^{d \times d}$ such that

$$\mathcal{E} = \mathcal{E}(\mathbf{A}, \boldsymbol{a}) := \{\boldsymbol{x} \in \mathbb{R}^d : \langle \boldsymbol{x} - \boldsymbol{a}, \mathbf{A}^{-1}(\boldsymbol{x} - \boldsymbol{a}) \rangle \le 1\};$$

above, we use the inverse of $\mathbf{A}$ so as to be consistent with Grötschel et al. [138]. $\text{vol}(\mathcal{E})$ is the *volume* of the ellipsoid $\mathcal{E}$.

We are now ready to introduce some basic oracles, which are known to be (polynomial-time) equivalent when $\mathcal{X}$ is well-bounded; in light of Remark 9.16, we can handle deviations $\boldsymbol{x}' \in \mathcal{X}$ instead of $\boldsymbol{x}' \in \mathcal{X}^{-\epsilon}$.[9.3] The first one simply ascertains (approximate) membership.

Definition 9.17 (Weak membership; [138])**.** Given a point $\boldsymbol{x} \in \mathbb{Q}^d$ and a rational number $\epsilon \in \mathbb{Q}_{>0}$, decide whether $\boldsymbol{x} \in \mathcal{X}^{+\epsilon}$.

A separation oracle takes a step further: if the point is not (approximately) in the set, it proceeds by producing a separating hyperplane, as defined next.

Definition 9.18 (Weak separation; [138])**.** Given a point $\boldsymbol{x} \in \mathbb{R}^d$ and a rational number $\epsilon \in \mathbb{Q}_{>0}$, either

- assert that $\boldsymbol{x} \in \mathcal{X}^{+\epsilon}$ or
- find a vector $\boldsymbol{c} \in \mathbb{Q}^d$ with $\|\boldsymbol{c}\|_\infty = 1$ such that $\langle \boldsymbol{c}, \boldsymbol{x}' \rangle < \langle \boldsymbol{c}, \boldsymbol{x} \rangle + \epsilon$ for every $\boldsymbol{x}' \in \mathcal{X}^{-\epsilon}$.

The final useful oracle (approximately) optimizes linear functions with respect to the constraint set; it enables verifying whether a point is an approximate SVI solution.

Definition 9.19 (Weak optimization; [138])**.** Given a vector $\boldsymbol{c} \in \mathbb{Q}^d$ and a rational number $\epsilon \in \mathbb{Q}_{>0}$, either

- assert that $\mathcal{X}^{-\epsilon}$ is empty or
- find a vector $\boldsymbol{x} \in \mathcal{X}^{+\epsilon} \cap \mathbb{Q}^d$ such that $\langle \boldsymbol{c}, \boldsymbol{x} \rangle \leq \langle \boldsymbol{c}, \boldsymbol{x}' \rangle + \epsilon$ for all $\boldsymbol{x}' \in \mathcal{X}^{-\epsilon}$.

Our main assumption concerning $\mathcal{X}$ is that it is given implicitly through access to any of those computationally equivalent oracles.

With regard to the mapping F of the underlying variational inequality problem, we gather our assumptions below; some of our results, such as Theorem 9.54, weaken these assumptions.

Assumption 9.20. For a fixed $\epsilon_0 \in \mathbb{Q}_{>0}$, the mapping $F : \mathcal{X} \to \mathbb{R}^d$ satisfies the following:

1. for any rational $\boldsymbol{x} \in \mathcal{X} \cap \mathbb{Q}^d$, $F(\boldsymbol{x})$ is a rational number that can be evaluated exactly in $\mathsf{poly}(d)$ time, with $\mathsf{size}(F(\boldsymbol{x})) \leq \mathsf{poly}(\mathsf{size}(\boldsymbol{x}))$;
2. for any $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$, $\|F(\boldsymbol{x}) - F(\boldsymbol{x}')\| \leq L\|\boldsymbol{x} - \boldsymbol{x}'\|$ for some $L \in \mathbb{Q}_{>0}$; and
3. for any $\boldsymbol{x} \in \mathcal{X}$, $\|F(\boldsymbol{x})\| \leq B$ for some $B \in \mathbb{Q}_{>0}$.

Regarding Item 1, a weaker assumption, which suffices for our purposes, is that we have access to an oracle that, for any $\boldsymbol{x} \in \mathcal{X} \cap \mathbb{Q}^d$ and rational $\delta \in \mathbb{Q}_{>0}$, returns $\boldsymbol{g} \in \mathbb{Q}^d$ such that $\|F(\boldsymbol{x}) - \boldsymbol{g}\| \leq \delta$. When specialized to multi-player games, Item 1 is precisely the *polynomial expectation property* of Papadimitriou and Roughgarden [237].

[9.3] For simplicity, we posit the strong versions of the oracles, that is, with $\epsilon = 0$. We explain in the appendix of the full paper [13] how to generalize this to weak oracles

For simplicity, our algorithm takes as input L and B, under the promise that F satisfies Assumption 9.20; this is not necessary: one can run the algorithm starting from L_0, B_0; if the output does not satisfy the desired property, it suffices to repeat, setting $L_1 = 2L_0$ and $B_1 = 2B_0$, and so on.

We next state a simple, well-known result regarding optimizing convex functions with respect to a constraint set that admits a separation oracle.

Theorem 9.21 ([138])**.** *Let $\mathcal{X} \subseteq \mathbb{R}^d$ be in isotropic position, given by a (weak) separation oracle, and $f : \mathcal{X} \to \mathbb{R}$ a convex function that can be evaluated exactly in* $\mathsf{poly}(d)$ *time. There is a* $\mathsf{poly}(d, \log(1/\epsilon))$*-time algorithm that outputs $\boldsymbol{x} \in \mathcal{X}^{+\epsilon}$ such that $f(\boldsymbol{x}) \leq \min_{\boldsymbol{x}' \in \mathcal{X}^{-\epsilon}} f(\boldsymbol{x}') + \epsilon$.*

Remark 9.22. In addition, suppose that f is differentiable and μ-strongly convex: $f(\boldsymbol{x}) \geq f(\boldsymbol{x}') + \langle \nabla f(\boldsymbol{x}'), \boldsymbol{x} - \boldsymbol{x}' \rangle + \frac{\mu}{2} \|\boldsymbol{x} - \boldsymbol{x}'\|^2$. If $\boldsymbol{x}'$ is the global minimum of f with respect to $\mathcal{X}^{-\epsilon}$, we have $\|\Pi_{\mathcal{X}^{-\epsilon}}(\boldsymbol{x}) - \boldsymbol{x}'\| \leq 2\epsilon/\mu + 2(f(\Pi_{\mathcal{X}^{-\epsilon}}(\boldsymbol{x})) - f(\boldsymbol{x}'))/\mu$; if f is Lipschitz continuous, it follows that $\boldsymbol{x}$ is also geometrically close to $\boldsymbol{x}'$.

9.3 Consequences and Manifestations of the Minty Condition

The Minty condition (Assumption 9.3) is intimately related to several important and seemingly disparate concepts in optimization and game theory. This section gathers a number of such connections, most of which are new. In Section 9.3.1, we begin by exploring connections in optimization, mostly revolving around quasar-convexity (Definition 9.24) and a more general incarnation thereof in general VI problems (Definition 9.26). Section 9.3.2 establishes a duality between MVIs and a relaxation of SVIs called *expected VIs* (Proposition 9.30), which are closely related to the notion of coarse correlated equilibria from game theory. We then leverage this duality to show that in explicitly represented multi-player games, ascertaining whether the Minty condition holds can be phrased as a linear program of polynomial size; this is to be contrasted with our coNP-hardness for succinct games (Theorem 9.89). Section 9.3.3 examines classes of games that satisfy the Minty condition, which notably includes *harmonic games*—after applying a suitable transformation (Proposition 9.38).

9.3.1 Optimization

We first turn our attention to optimizing a single function. Let $f : \mathcal{X} \to \mathbb{R}$ be a differentiable function to be minimized.[9.4] As we discussed earlier, an ϵ-SVI solution $\boldsymbol{x} \in \mathcal{X}$ to the problem arising when $F := \nabla f(\boldsymbol{x})$ is an approximate stationary point of gradient descent applied on f; namely, $\langle \nabla f(\boldsymbol{x}), \boldsymbol{x}' - \boldsymbol{x} \rangle \geq -\epsilon$ for all $\boldsymbol{x}' \in \mathcal{X}$. In particular, if there exists $r > 0$ such that $\mathcal{B}_r(\boldsymbol{x}) \subseteq \mathcal{X}$—that is, $\boldsymbol{x}$ is in the interior of $\mathcal{X}$—it follows that $\|\nabla f(\boldsymbol{x})\| \leq \epsilon/r$. Of course, when f is nonconvex, ϵ-SVIs are not generally approximate global minima—they can even be saddle points. This is in stark contrast to MVI solutions, although their existence is not guaranteed in

[9.4] We always assume that f is differentiable on an open superset $\hat{\mathcal{X}} \supset \mathcal{X}$.

general; indeed, the following was observed, for example, by Huang and Zhang [152, Theorem 2.10].

Proposition 9.23 ([152]). *Consider the optimization problem* $\min_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x})$*, where* f *is differentiable and* $\mathcal{X}$ *is convex and compact. If* $\boldsymbol{x} \in \mathcal{X}$ *is an MVI solution with respect to* $F := \nabla f$*, then* $\boldsymbol{x}$ *is a global minimum of* f*.*

On the other hand, a global minimum of f is not necessarily an MVI solution—otherwise our main result (Theorem 9.46) would imply $\mathsf{CLS} \subseteq \mathsf{P}$ (*cf.* [108]); see Huang and Zhang [152, Remark 2.11] for a concrete example.

One special case in which SVI solutions do correspond to global minima is when f is *quasar-convex*, in the following sense.

Definition 9.24 (Quasar-convexity). Let $\lambda \in (0, 1]$ and $\boldsymbol{x}$ be a minimizer of a differentiable function $f : \mathcal{X} \to \mathbb{R}$. We say that f is *λ-quasar-convex* with respect to $\boldsymbol{x}$ if

$$f(\boldsymbol{x}) \geq f(\boldsymbol{x}') + \frac{1}{\lambda}\langle \nabla f(\boldsymbol{x}'), \boldsymbol{x} - \boldsymbol{x}' \rangle \quad \forall \boldsymbol{x}' \in \mathcal{X}. \tag{9.6}$$

Quasar-convexity is a generalization of the usual notion of convexity: if (9.6) holds for any $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$ and $\lambda = 1$, one recovers precisely convexity for differentiable functions. Further, when (9.6) holds only with respect to $\boldsymbol{x}$ (as in Definition 9.24) but with $\lambda = 1$, we get the notion of *star-convexity* [193, 232]. It follows immediately from the definition that quasar-convexity is in fact a strengthening of the Minty condition, which further guarantees that all approximate SVI solutions are approximately global minima in terms of value—this latter property does not necessarily hold under solely the Minty condition.[9.5]

Proposition 9.25. *Let* $F := \nabla f$ *for a differentiable,* λ*-quasar-convex function* $f : \mathcal{X} \to \mathbb{R}$ *with respect to a global minimum* $\boldsymbol{x} \in \mathcal{X}$ *of* f*. Then,* $\boldsymbol{x}$ *is a solution to the Minty VI problem. Furthermore, any* ϵ*-SVI solution satisfies* $f(\boldsymbol{x}') \leq f(\boldsymbol{x}) + \epsilon/\lambda$*.*

Proof. The fact that $\boldsymbol{x}$ satisfies the Minty VI follows since

$$\langle \nabla f(\boldsymbol{x}'), \boldsymbol{x}' - \boldsymbol{x} \rangle \geq \lambda(f(\boldsymbol{x}') - f(\boldsymbol{x})) \geq 0$$

for any $\boldsymbol{x}' \in \mathcal{X}$, where we used the fact that $f(\boldsymbol{x}) \leq f(\boldsymbol{x}')$ ($\boldsymbol{x}$ is a global minimum of f). Now, let $\boldsymbol{x}'$ be an ϵ-SVI solution, which implies that $\langle \nabla f(\boldsymbol{x}'), \boldsymbol{x} - \boldsymbol{x}' \rangle \geq -\epsilon$. Combining with (9.6), we have

$$f(\boldsymbol{x}') \leq f(\boldsymbol{x}) + \frac{\epsilon}{\lambda},$$

as promised. □

[9.5] Hinder et al. [147] established a lower bound of $\Omega(\lambda^{-1}\epsilon^{-1/2})$ in terms of the number of gradient evaluations required to minimize a λ-quasar convex function; this does not contradict Theorem 9.10 because their lower bound only applies if the dimension is large enough as a function of ϵ; in particular, their lower bound targets "dimension-free" algorithms.

Smooth VIs. The notion of quasar-convexity given in Definition 9.24 can be significantly generalized to general VI problems. As we shall see, this captures a special case of the seminal notion of *smoothness*, introduced by Roughgarden [256].[9.6] (For convenience, we take the perspective of maximization in the following definition.)

Definition 9.26 (Smoothness for VIs)**.** Let $\lambda > 0$ and $\nu > -1$. Consider further a function $Q : \mathcal{X} \to \mathbb{R}$ and a global maximum $\boldsymbol{x} \in \mathcal{X}$ of Q. A VI problem with respect to the mapping $F : \mathcal{X} \to \mathbb{R}^d$ is called (λ, ν)*-smooth* with respect to Q and $\boldsymbol{x}$ if

$$\langle F(\boldsymbol{x}'), \boldsymbol{x}' - \boldsymbol{x} \rangle \geq \lambda Q(\boldsymbol{x}) - (\nu + 1) Q(\boldsymbol{x}') \quad \forall \boldsymbol{x}' \in \mathcal{X}. \tag{9.7}$$

In particular, $(\lambda, \lambda - 1)$-smoothness equates to λ-quasar-convexity when we define $Q := -f$ and $F := \nabla f$. Furthermore, in the special case of multi-player games, Definition 9.26 is equivalent to the seminal concept of smoothness introduced by Roughgarden [256]—in particular, Definition 9.26 is a direct extension of the more general concept of "local smoothness" per Roughgarden and Schoppmann [257].

Definition 9.27 (Smoothness for games; [256])**.** Let $\lambda > 0$ and $\nu > -1$, and $\boldsymbol{x} \in \arg\max_{\boldsymbol{x}' \in \mathcal{X}} \mathsf{SW}(\boldsymbol{x}')$ with respect to an n-player game Γ. Γ is called (λ, ν)*-smooth* with respect to $\boldsymbol{x}$ if

$$\sum_{i=1}^{n} u_i(\boldsymbol{x}_i, \boldsymbol{x}'_{-i}) \geq \lambda \mathsf{SW}(\boldsymbol{x}) - \nu \mathsf{SW}(\boldsymbol{x}') \quad \forall \boldsymbol{x}' \in \mathcal{X}. \tag{9.8}$$

By defining $Q := \mathsf{SW}$—the social welfare function, we see that (9.8) is equivalent to (9.7) due to multilinearity. The key motivation behind Definition 9.27 is that it enables bounding the social welfare of any (A)CCE in terms of the optimal welfare [256]. Smoothness manifests itself prominently in a host of important applications; for example, we refer to the survey of Roughgarden et al. [258]. For our purposes, the key point is that Definition 9.26 enables generalizing Proposition 9.25 to a broader family of problems beyond (single-function) optimization:

Proposition 9.28. *Let $\lambda > 0$, a function $Q : \mathcal{X} \to \mathbb{R}$ with a global maximum at $\boldsymbol{x} \in \mathcal{X}$, and a mapping $F : \mathcal{X} \to \mathbb{R}^d$. If the corresponding VI problem is $(\lambda, \lambda - 1)$-smooth with respect to Q and $\boldsymbol{x}$, then $\boldsymbol{x}$ is a solution to the Minty VI problem. Furthermore, any ϵ-SVI solution satisfies $Q(\boldsymbol{x}') \leq Q(\boldsymbol{x}) + \epsilon/\lambda$.*

The proof is analogous to that of Proposition 9.25.

9.3.2 Expected VIs and Duality with MVIs

The next connection is that MVIs are, in a precise sense, duals of a certain relaxation of SVIs which we call *expected VIs*; we refer to Cai et al. [46] and Şeref Ahunbay [70] for some precursors of that definition in the context of nonconcave games.

We begin by stating the definition of expected VIs.

[9.6] We caution that smoothness per Definition 9.26 is different than the usual notion of smoothness in optimization; we chose to overload notation so as to be consistent with the terminology of Roughgarden [256].

Definition 9.29. In the context of Definition 9.1, the ϵ-*expected* VI problem asks for a *distribution* $\mu \in \Delta(\mathcal{X})$ such that

$$\mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mu} \langle F(\boldsymbol{x}), \boldsymbol{x}' - \boldsymbol{x} \rangle \geq -\epsilon \quad \forall \boldsymbol{x}' \in \mathcal{X}. \tag{9.9}$$

That is, in an expected VI, it suffices if the (S)VI constraint holds *in expectation* when $\boldsymbol{x}$ is drawn from a distribution μ. Unlike SVIs, expected VIs can be solved in $\text{poly}(d, \log(1/\epsilon))$ time, as we will see in Section 9.6. Definition 9.29 places no restriction on ϵ being nonnegative; indeed, expected VIs with a negative gap may exist (*cf.* Proposition 9.30)—this is obviously not possible for SVIs. For $\epsilon < 0$, an ϵ-EVI solution will also be referred to as a $(-\epsilon)$-*strict EVI* solution.

In this context, starting from (9.9), we observe that

$$\max_{\mu \in \Delta(\mathcal{X})} \min_{\boldsymbol{x}' \in \mathcal{X}} \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mu} \langle F(\boldsymbol{x}), \boldsymbol{x}' - \boldsymbol{x} \rangle = \min_{\boldsymbol{x}' \in \mathcal{X}} \max_{\mu \in \Delta(\mathcal{X})} \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mu} \langle F(\boldsymbol{x}), \boldsymbol{x}' - \boldsymbol{x} \rangle,$$

where the equality follows by the minimax theorem [273]; indeed, the function $\mathbb{E}_{\boldsymbol{x} \sim \mu} \langle F(\boldsymbol{x}), \boldsymbol{x} - \boldsymbol{x}' \rangle$ is bilinear in terms of μ and $\boldsymbol{x}'$. Equivalently,

$$- \max_{\mu \in \Delta(\mathcal{X})} \min_{\boldsymbol{x}' \in \mathcal{X}} \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mu} \langle F(\boldsymbol{x}), \boldsymbol{x}' - \boldsymbol{x} \rangle = \max_{\boldsymbol{x}' \in \mathcal{X}} \min_{\boldsymbol{x} \in \mathcal{X}} \langle F(\boldsymbol{x}), \boldsymbol{x} - \boldsymbol{x}' \rangle, \tag{9.10}$$

where we used the fact that, for a given $\boldsymbol{x}' \in \mathcal{X}$, $\min_{\mu \in \Delta(\mathcal{X})} \mathbb{E}_{\boldsymbol{x} \sim \mu} \langle F(\boldsymbol{x}), \boldsymbol{x} - \boldsymbol{x}' \rangle = \min_{\boldsymbol{x} \in \mathcal{X}} \langle F(\boldsymbol{x}), \boldsymbol{x} - \boldsymbol{x}' \rangle$. By (9.10), we arrive at the following characterization of the Minty condition.

Proposition 9.30. *The Minty condition (Assumption 9.3) holds if and only if there is no ϵ-expected VI solution (Definition 9.29) with $\epsilon < 0$.*

Proof. If the Minty condition holds, there exists $\boldsymbol{x}' \in \mathcal{X}$ such that $\min_{\boldsymbol{x} \in \mathcal{X}} \langle F(\boldsymbol{x}), \boldsymbol{x} - \boldsymbol{x}' \rangle \geq 0$. By (9.10), this implies that $\max_{\mu \in \Delta(\mathcal{X})} \min_{\boldsymbol{x}' \in \mathcal{X}} \mathbb{E}_{\boldsymbol{x} \sim \mu} \langle F(\boldsymbol{x}), \boldsymbol{x}' - \boldsymbol{x} \rangle \leq 0$, which means that every ϵ-expected SVI solution μ must satisfy $\epsilon \geq 0$. The converse is also immediate. □

Coarse correlated equilibria in games. Proposition 9.30 has a particularly notable consequence in the context of n-player (normal-form) games. Here, each player $i \in [n]$ selects as (mixed) strategy a probability distribution $\boldsymbol{x}_i \in \Delta(\mathcal{A}_i) =: \mathcal{X}_i$ over a finite set of available actions $\mathcal{A}_i$. Under a joint strategy $\boldsymbol{x} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_n) \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_n =: \mathcal{X}$, we denote by $u_i(\boldsymbol{x})$ the expected utility of a player i. Expected VIs (per Definition 9.29) correspond to the following equilibrium concept; this can be readily extended to general concave games (Section 9.5 does so for two-player games).

Definition 9.31 (Average CCE)**.** For an n-player game, a distribution $\mu \in \Delta(\mathcal{A}_1 \times \cdots \times \mathcal{A}_n)$ is an ϵ-*average coarse correlated equilibrium (ϵ-ACCE)* if

$$\sum_{i=1}^{n} \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mu} [u_i(\boldsymbol{x}_i', \boldsymbol{x}_{-i}) - u_i(\boldsymbol{x})] \leq \epsilon \quad \forall \boldsymbol{x}_1', \dots, \boldsymbol{x}_n' \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_n. \tag{9.11}$$

Several remarks are in order. An ACCE is a relaxation of a CCE [226], which in turn relaxes *correlated equilibria (CE) à la* Aumann [15]. The key difference between ACCEs and CCEs is that the former only insists on bounding the *cumulative* deviation benefit over all players, whereas in a CCE one bounds the *maximum* deviation benefit; the term "average" CCE is due to Nadav and Roughgarden [229], who also separated ACCEs from CCEs. Now, as we shall see, expected VIs are to ACCEs what SVIs are to *Nash equilibria*; we now recall the latter definition.

Definition 9.32. For an n-player game, a strategy $\boldsymbol{x} \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$ is an *ϵ-Nash equilibrium* if[9.7]

$$\sum_{i=1}^{n} (u_i(\boldsymbol{x}'_i, \boldsymbol{x}_{-i}) - u_i(\boldsymbol{x})) \leq \epsilon \quad \forall \boldsymbol{x}'_1, \dots, \boldsymbol{x}'_n \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_n.$$

As we alluded to, Definition 9.31 can be naturally cast as an expected VI problem per Definition 9.29. Indeed, we define

$$F : \boldsymbol{x} \mapsto ((-u_i(a_i, \boldsymbol{x}_{-i}))_{a_i \in \mathcal{A}_i})_{i=1}^{n} \text{ and } \mathcal{X} := \Delta(\mathcal{A}_1) \times \cdots \times \Delta(\mathcal{A}_n). \tag{9.12}$$

That (9.11) is equivalent to the resulting expected VI problem follows immediately from the definitions, noting that one can always—without any loss of generality—restrict μ to be a distribution over pure strategies.

In this context, a direct consequence of Proposition 9.30 is that there is a linear program with a number of variables and constraints polynomial in $\prod_{i=1}^{n} |\mathcal{A}_i|$, whose output determines whether the Minty property holds. In particular, one can compute the ACCE that minimizes the equilibrium gap per (9.11). Let ϵ be the output of that linear program. By Proposition 9.30, the Minty condition—with respect to the corresponding mapping F—holds if and only if $\epsilon = 0$ (of course, there is always a 0-ACCE simply because an exact Nash equilibrium exists).

Proposition 9.33. *For any n-player normal-form game, there is an algorithm polynomial in $\prod_{i=1}^{n} |\mathcal{A}_i|$ (and the number of bits needed to encode the payoff tensor) that determines whether the Minty condition with respect to the corresponding VI problem per* (9.12) *holds.*

As a result, there is a polynomial-time algorithm for determining whether the Minty condition holds in explicitly represented (normal-form) games, meaning that the input fully specifies each entry of the utility tensors; as we show in Theorem 9.89, this is no longer the case in succinct games (with the polynomial expectation property). Moreover, we also state the following immediate consequence.

Proposition 9.34. *For any n-player game that satisfies the Minty condition, there is an algorithm polynomial in $\prod_{i=1}^{n} |\mathcal{A}_i|$ that determines an MVI solution (per Definition 9.2).*

[9.7] It is more common to bound the maximum deviation benefit (as opposed to the cumulative one), but—unlike CCEs—the two are equivalent up to a factor of n in the approximation.

Games with nonnegative sum of regrets. Moreover, the condition that every ϵ-expected VI solution has nonnegative gap (Proposition 9.30) is precisely the condition put forward by Anagnostides et al. [12], which was used to define the class of games with "nonnegative sum of regrets." To be precise, the *regret* of a player $i \in [n]$ who has observed the sequence of utilities $(\boldsymbol{u}_i^{(t)}(\boldsymbol{x}_{-i}^{(t)}))_{1 \le t \le T}$ and has selected the sequence of strategies $(\boldsymbol{x}_i^{(t)})_{1 \le t \le T}$ is defined as

$$\textsc{Reg}_i^{(T)} := \max_{\boldsymbol{x}_i \in \mathcal{X}_i} \sum_{t=1}^{T} \alpha^{(t)} \langle \boldsymbol{x}_i - \boldsymbol{x}_i^{(t)}, \boldsymbol{u}_i^{(t)} \rangle, \tag{9.13}$$

where $\alpha^{(1)}, \dots, \alpha^{(T)} \ge 0$ are weights such that $\sum_{t=1}^{T} \alpha^{(t)} = 1$; it is common to define regret when $\alpha^{(1)} = \cdots = \alpha^{(T)} = 1/T$, but we will operate under the more flexible definition given in (9.13).

Observation 9.35. *We say that a game Γ has nonnegative sum of regrets if for any $T \in \mathbb{N}$, weights $(\alpha^{(t)})_{1 \le t \le T} \in \Delta(T)$, and sequence of joint strategies $(\boldsymbol{x}^{(t)})_{1 \le t \le T} \in \mathcal{X}^T$, it holds that $\sum_{i=1}^{n} \textsc{Reg}_i^{(T)} \ge 0$. This is equivalent to any ϵ-ACCE in Γ satisfying $\epsilon \ge 0$. By Proposition 9.30, it is also equivalent to the Minty property with respect to Γ.*

Among others implications, in such games it is possible to show that a broad class of no-regret dynamics—namely, ones satisfying the *RVU bound* of Syrgkanis et al. [281], such as *optimistic* mirror descent—guarantees optimal per-player (average) regret vanishing at a rate of T^{-1}; it remains an open question whether this is possible in general multi-player games (*cf.* [79]).

9.3.3 Harmonic Games

Moving forward, we observe that (a weighted version of) the Minty condition manifests itself in *harmonic games*. This is a class of games introduced by Candogan et al. [52], who famously provided a decomposition of any game—based on Helmholtz decomposition—into a direct sum of a *potential* game and a harmonic game; this decomposition is unique up to an affine transformation that preserves the equilibria of the game. The potential component captures games with aligned interests, whereas the harmonic component captures games with conflicting interests.

Following recent follow-up work [5, 194], we give below a more general definition of harmonic games than the one introduced by Candogan et al. [52].[9.8]

Definition 9.36 (Harmonic games; [5, 52, 194])**.** A finite game Γ is called *harmonic* if for each player $i \in [n]$ there exists $\boldsymbol{\sigma}_i \in \mathbb{R}_{>0}^{\mathcal{A}_i}$ such that

$$\sum_{i=1}^{n} \sum_{a_i \in \mathcal{A}_i} \boldsymbol{\sigma}_i(a_i)(u_i(\boldsymbol{a}') - u_i(a_i, \boldsymbol{a}'_{-i})) = 0 \quad \forall \boldsymbol{a}' \in \mathcal{A}_1 \times \cdots \times \mathcal{A}_n. \tag{9.14}$$

To cast this as a special case of the Minty property, we observe that (9.14) can be equivalently reformulated as asking for a collection of (strictly) positive weights $w_1, \dots, w_n$ and fully mixed

[9.8] Under the original definition of Candogan et al. [52], the strategy profile in which each player mixes uniformly at random is a Nash equilibrium, thereby trivializing equilibrium computation.

strategies $\boldsymbol{x}_1 \in \Delta(\mathcal{A}_1) \cap \mathbb{R}^{\mathcal{A}_1}_{>0}, \dots, \boldsymbol{x}_n \in \Delta(\mathcal{A}_n) \cap \mathbb{R}^{\mathcal{A}_n}_{>0}$ such that[9.9]

$$\sum_{i=1}^{n} w_i(u_i(\boldsymbol{a}') - u_i(\boldsymbol{x}_i, \boldsymbol{a}'_{-i})) = 0 \quad \forall \boldsymbol{a}' \in \mathcal{A}_1 \times \cdots \times \mathcal{A}_n. \tag{9.15}$$

We can now recognize this as a special case of the Minty property after we suitably rescale the utilities in the definition of F. But there are two lingering issues with this observation: first, we do not know the weights $w_1, \dots, w_n$; and second, even if we could rescale the utilities, the complexity of the algorithm would be polynomial in $\log(\rho)$, where $\rho := \max_{1 \le i \le n} w_i / \min_{1 \le i \le n} w_i$.

To address these issues, we first make a simple observation regarding the bit complexity of $\boldsymbol{\sigma}$ that satisfies (9.14).

Lemma 9.37. *Let* $\mathsf{size}(u_i(\boldsymbol{a})) \le \mathsf{poly}(n, \max_{1 \le i \le n} |\mathcal{A}_i|)$ *for all* $i \in [n]$ *and* $\boldsymbol{a} \in \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$. *Suppose further that* (9.14) *is feasible. Then, it can be satisfied with respect to* $\boldsymbol{\sigma} = (\boldsymbol{\sigma}_1, \dots, \boldsymbol{\sigma}_n) \in \mathbb{R}^{\mathcal{A}_1}_{>0} \times \cdots \times \mathbb{R}^{\mathcal{A}_n}_{>0}$ *such that* $\mathsf{size}(\boldsymbol{\sigma}) \le \mathsf{poly}(n, \max_{1 \le i \le n} |\mathcal{A}_i|)$.

> *Proof.* (9.14) induces a linear program with a polynomial number of variables (and exponential number of constraints). The number of active constraints required to define a vertex is thus polynomial. Since the coefficients of each constraint have polynomial bit complexity (by assumption), the claim follows.[9.10] □

Returning to (9.15), we can assume that $\sum_{i=1}^{n} w_i = 1$ (by rescaling); Lemma 9.37 implies that $w_i \ge 1/2^{\mathsf{poly}(n, \max_{1 \le i \le n} |\mathcal{A}_i|)}$. Having established this lower bound, we consider the mapping

$$G : \mathcal{P}_{\ge \alpha} \ni (w_1, \dots, w_n, w_1 \boldsymbol{x}_1, \dots, w_n \boldsymbol{x}_n) := (u_1(\boldsymbol{x}), \dots, u_n(\boldsymbol{x}), -(u_i(a_i, \boldsymbol{x}_{-i})_{a_i \in \mathcal{A}_i})_{i=1}^{n}), \tag{9.16}$$

where

$$\mathcal{P}_{\ge \alpha} := \left\{ (w_1, \dots, w_n, w_1 \boldsymbol{x}_1, \dots, w_n \boldsymbol{x}_n) : \boldsymbol{w} \in \Delta(n) \cap \mathbb{R}^n_{\ge \alpha}, \boldsymbol{x}_1 \in \Delta(\mathcal{A}_1), \dots, \boldsymbol{x}_n \in \Delta(\mathcal{A}_n) \right\} \tag{9.17}$$

for a sufficiently small $\alpha = 1/2^{\mathsf{poly}(n, \max_{1 \le i \le n} |\mathcal{A}_i|)}$ (per Lemma 9.37). The following now follows directly from the definitions.

[9.9] Combining Lemma 9.4 and Proposition 9.38, it follows that $\boldsymbol{x}$ is an exact Nash equilibrium. In particular, this means that any harmonic game admits a fully mixed Nash equilibrium, but we are yet not aware of any prior polynomial-time algorithm for computing Nash equilibria in harmonic games. To elaborate on this point further, in certain classes of games, such as two-player (general-sum) games, knowing the support of the equilibrium reduces the problem to a linear system, which can be in turn solved in polynomial time (*e.g.*, [263]). This is not so in multi-player games: Etessami and Yannakakis [92, Corollary 13] proved certain hardness results based on a three-player game promised to have a unique, fully mixed Nash equilibrium.

[9.10] For explicitly represented (normal-form) games, it is evident from (9.14) that there is a polynomial-time algorithm for computing $\boldsymbol{\sigma}$, and hence a Nash equilibrium of that game. Our focus here is on succinct games (with the polynomial expectation property), in which case the LP induced by (9.14) has exponentially many constraints.

Proposition 9.38. *Consider any harmonic game Γ per Definition 9.36. If we define* $\mathrm{VI}(\mathcal{P}_{\geq\alpha}, G)$ *per* (9.16) *and* (9.17)*, the following properties hold:*

- $\mathrm{VI}(\mathcal{P}_{\geq\alpha}, G)$ *satisfies the Minty condition;*
- G *is* $2^{\mathsf{poly}(n, \max_{1 \leq i \leq n} |\mathcal{A}_i|)}$*-Lipschitz continuous; and*
- *an* ϵ*-SVI of* $\mathrm{VI}(\mathcal{P}_{\geq\alpha}, G)$ *is an* $\epsilon \cdot 2^{\mathsf{poly}(n, \max_{1 \leq i \leq n} |\mathcal{A}_i|)}$*-Nash equilibrium of* Γ.

As a result, our main result (Theorem 9.46) implies a polynomial-tme algorithm for computing ϵ-Nash equilibria in harmonic games.

Some simple examples of games that adhere to Definition 9.36 include "cyclic games," in the sense of Hofbauer and Schlag [149], the buyer-seller game of Friedman [114], and the crime deterrence game analyzed by Cressman et al. [69].

Polymatrix zero-sum games and beyond. We continue with a related but distinct class of games known as *polymatrix games* [45]; in particular, it is easy to see that any two-player zero-sum game with a fully mixed Nash equilibrium is harmonic per Definition 9.36. We first make an observation with regard to general MVI problems, providing a sufficient condition under which Assumption 9.3 holds.

Proposition 9.39. *Consider a problem* $\mathrm{VI}(\mathcal{X}, F)$ *such that* F *is linear and* $\langle F(\boldsymbol{x}), \boldsymbol{x} \rangle = 0$ *for all* $\boldsymbol{x} \in \mathcal{X}$*. Then, the Minty condition (Assumption 9.3) holds.*

Proof. The Minty condition is equivalent to $\max_{\boldsymbol{x} \in \mathcal{X}} \min_{\boldsymbol{x}' \in \mathcal{X}} \langle F(\boldsymbol{x}'), \boldsymbol{x}' - \boldsymbol{x} \rangle \geq 0$. But under our assumptions, the function $(\boldsymbol{x}, \boldsymbol{x}') \mapsto \langle F(\boldsymbol{x}'), \boldsymbol{x}' - \boldsymbol{x} \rangle$ is bilinear, which in turn implies that

$$\max_{\boldsymbol{x} \in \mathcal{X}} \min_{\boldsymbol{x}' \in \mathcal{X}} \langle F(\boldsymbol{x}'), \boldsymbol{x}' - \boldsymbol{x} \rangle = \min_{\boldsymbol{x}' \in \mathcal{X}} \max_{\boldsymbol{x} \in \mathcal{X}} \langle F(\boldsymbol{x}'), \boldsymbol{x}' - \boldsymbol{x} \rangle \geq 0,$$

by the minimax theorem [273]. □

When specialized to multi-player games, the two preconditions of Proposition 9.39 are satisfied when i) the game is (globally) zero-sum, meaning that $\mathsf{SW}(\boldsymbol{x}) := \sum_{i=1}^{n} u_i(\boldsymbol{x}) = -\langle F(\boldsymbol{x}), \boldsymbol{x} \rangle = 0$ (by multilinearity) for all $\boldsymbol{x}$, and ii) the utility gradient of each player is linear in the joint strategy. Those two assumptions are satisfied in zero-sum polymatrix games [45]. A polynomial-time algorithm for computing Nash equilibria in zero-sum polymatrix games was obtained by Cai et al. [45], who observed that taking the marginals of any CCE yields a Nash equilibrium; this approach falls short more generally if one merely assumes that the Minty condition holds (Proposition 9.95). On the other hand, without the zero-sum restriction, computing ϵ-Nash equilibria in polymatrix games is PPAD-hard [83, 259].

9.4 Solving SVIs under the Minty Condition

In this section, we establish our main result. To begin with, we gather some basic facts about the central-cut ellipsoid in Section 9.4.1, without assuming that the underlying convex set is fully dimensional (Theorem 9.40). We then show how to adapt this basic paradigm (in ExtraGradientEllipsoid) by introducing some new key ideas, arriving at our main result in Theorem 9.46: a polynomial-time algorithm for computing ϵ-SVI solutions under the Minty condition. Sections 9.4.3 and 9.4.4 concern two basic extensions of our main result: the former relaxes the Minty condition following the weaker property put forward by Diakonikolas et al. [87], while the latter relaxes the assumption that the mapping F is continuous, imposing instead an assumption generalizing quasar-convexity—itself a strengthening of the Minty condition (Proposition 9.28). Finally, Section 9.4.5 deals with the most general setting wherein the Minty condition (and relaxations thereof per Section 9.4.3) can be altogether violated. It shows how the execution of our main algorithm (ExtraGradientEllipsoid) can produce a polynomial certificate—in the form of a *strict* EVI solution—that the Minty condition is violated.

9.4.1 Central-Cut Ellipsoid

We will use the following standard result concerning one incarnation of the central-cut ellipsoid [138, Theorem 3.2.1]. It is suited to our purposes as it does not rest on the usual assumption that the underlying constraint set is fully dimensional.

Theorem 9.40 ([138]). *Let $\epsilon \in \mathbb{Q}_{>0}$ and $\mathcal{K} \subseteq \mathcal{B}_R(\mathbf{0})$ be a circumscribed closed and convex set, with $R \geq 1$, given by a polynomial-time oracle $\mathsf{SEP}_{\mathcal{K}}$ such that for any $\boldsymbol{x} \in \mathbb{Q}^d$ and $\delta \in \mathbb{Q}_{>0}$, either asserts that $\boldsymbol{x} \in \mathcal{K}^{+\delta}$ or finds a vector $\boldsymbol{c} \in \mathbb{Q}^d$ with $\|\boldsymbol{c}\|_\infty = 1$ with $\langle \boldsymbol{c}, \boldsymbol{x}' \rangle \leq \langle \boldsymbol{c}, \boldsymbol{x} \rangle + \delta$ for every $\boldsymbol{x}' \in \mathcal{K}$. There is a polynomial-time algorithm that returns one of the following:*

- *a point in $\mathcal{K}^{+\epsilon}$ or*
- *an ellipsoid $\mathcal{E} \subseteq \mathbb{R}^d$, described by a positive definite matrix $\mathbf{A} \in \mathbb{Q}^{d \times d}$ and a point $\boldsymbol{a} \in \mathbb{Q}^d$, such that $\mathcal{K} \subseteq \mathcal{E}$ and $\operatorname{vol}(\mathcal{E}) \leq \epsilon$.*

Theorem 9.40 is based on the *central-cut* ellipsoid method (Ellipsoid). It produces a sequence of ellipsoids, $\mathcal{E}^{(0)}, \mathcal{E}^{(1)}, \dots, \mathcal{E}^{(T)}$, each of which contains the underlying set $\mathcal{K}$, such that either at least one of their centers belongs to $\mathcal{K}^{+\epsilon}$, or the last ellipsoid $\mathcal{E}^{(T)}$ has volume at most ϵ. We clarify that, in Algorithm 9.3, we use the notation $\approx_p$ to mean that the left-hand side is obtained by truncating the binary expansions of the numbers on the right-hand side after p digits behind the binary point. The correctness of Ellipsoid boils down to the following lemma.

Lemma 9.41 ([138]). *At every iteration t of Ellipsoid, the following properties hold:*

- *the matrix $\mathbf{A}^{(t)}$ is positive definite with $\|\boldsymbol{a}^{(t)}\| \leq R2^t$, $\|\mathbf{A}^{(t)}\| \leq R^2 2^t$, and $\|(\mathbf{A}^{(t)})^{-1}\| \leq R^{-2}4^t$;*
- *$\mathcal{K} \subseteq \mathcal{E}^{(t)}$; and*

Algorithm 9.3 (Ellipsoid): Central-cut ellipsoid algorithm [138]

```
input: a separation oracle $\mathsf{SEP}_{\mathcal{K}}$ for $\mathcal{K}$ per Theorem 9.40, rational $\epsilon > 0$.
output: a point in $\mathcal{K}^{+\epsilon}$ or an ellipsoid $\mathcal{E} \subseteq \mathbb{R}^d$ such that $\mathcal{K} \subseteq \mathcal{E}$ and $\mathrm{vol}(\mathcal{E}) \leq \epsilon$.
set the maximum number of iterations as $T := \lceil 5d \log(1/\epsilon) + 5d^2 \log(2R) \rceil$
set the precision parameter as $p := 8T$
set the error tolerance for $\mathsf{SEP}_{\mathcal{K}}$ as $\delta := 2^{-p}$
initialize the ellipsoid $\mathcal{E}^{(0)} := \mathcal{E}^{(0)}(\mathbf{A}^{(0)}, \boldsymbol{a}^{(0)})$ as $\boldsymbol{a}^{(0)} := \mathbf{0}$ and $\mathbf{A}^{(0)} := R^2 \mathbf{I}_d$
for $t = 0, \dots, T-1$ do
    invoke $\mathsf{SEP}_{\mathcal{K}}$ with input $\boldsymbol{a}^{(t)}$ and error $\delta$
    if $\boldsymbol{a}^{(t)} \in \mathcal{K}^{+\delta}$ then return $\boldsymbol{a}^{(t)}$
    else
        $\mathsf{SEP}_{\mathcal{K}}$ has returned $\boldsymbol{c} \in \mathbb{Q}^d$, with $\|\boldsymbol{c}\|_\infty = 1$, such that $\langle \boldsymbol{c}, \boldsymbol{x} \rangle \leq \langle \boldsymbol{c}, \boldsymbol{a}^{(t)} \rangle + \delta$ for all $\boldsymbol{x} \in \mathcal{K}$
        update the ellipsoid:
        $$\boldsymbol{a}^{(t+1)} \approx_p \boldsymbol{a}^{(t)} - \frac{1}{d+1} \frac{\mathbf{A}^{(t)} \boldsymbol{c}}{\sqrt{\langle \boldsymbol{c}, \mathbf{A}^{(t)} \boldsymbol{c} \rangle}} \text{ and } \mathbf{A}^{(t+1)} \approx_p \frac{2d^2+3}{2d^2} \left( \mathbf{A}^{(t)} - \frac{2}{d+1} \frac{\mathbf{A}^{(t)} \boldsymbol{c} \boldsymbol{c}^\top \mathbf{A}^{(t)}}{\sqrt{\langle \boldsymbol{c}, \mathbf{A}^{(t)} \boldsymbol{c} \rangle}} \right).$$
return $\mathcal{E}^{(T)}$
```

- $\mathrm{vol}(\mathcal{E}^{(t+1)}) / \mathrm{vol}(\mathcal{E}^{(t)}) \leq e^{-1/(5d)}$.

Armed with this lemma, Theorem 9.40 follows by noting that $\mathrm{vol}(\mathcal{E}^{(T)}) \leq e^{-T/(5d)} \mathrm{vol}(\mathcal{E}^{(0)})$ and $\mathrm{vol}(\mathcal{E}^{(0)}) \leq (2R)^d$; by the choice of T in Algorithm 9.3, we conclude that, if the algorithm failed to terminate (in Algorithm 9.3) with a point in $\mathcal{K}^{+\epsilon}$ (the value of δ in Algorithm 9.3 implies that $\mathcal{K}^{+\delta} \subseteq \mathcal{K}^{+\epsilon}$), we have $\mathrm{vol}(\mathcal{E}^{(T)}) \leq \epsilon$, as promised.

9.4.2 Our Algorithm and its Analysis

We will now show how to leverage Ellipsoid to compute ϵ-SVI solutions under the Minty property. To do so, we are first faced with an immediate concern: the set of SVI solutions is not necessarily convex even when the Minty property holds (see the function behind Proposition 9.92). On the other hand, while the set of MVI solutions is convex (Claim 9.50), it is hard to verify whether a point satisfies the Minty VI, as we show in Theorems 9.87 and 9.89.

We address this by executing the following hybrid version of the ellipsoid. We let $\mathcal{K}$ be the set of MVI solutions—points that satisfy (9.2); for now, we assume that $\mathcal{K} \neq \emptyset$, although we will relax that assumption later (Sections 9.4.3 and 9.4.5). At each iteration, we evaluate whether the center of the ellipsoid—when it belongs to $\mathcal{X}$—is an ϵ-SVI solution, which boils down to a call to the optimization oracle (Definition 9.19); if not, the key observation is that we can *strictly* separate that point from the set of MVIs—in the sense of Definition 9.42.

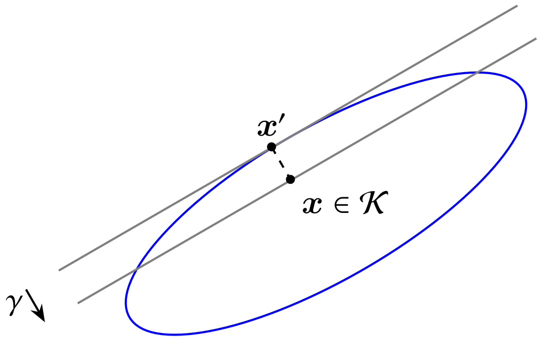


Figure 9.4: *A sequence of γ-strict separating hyperplanes implies that any $\boldsymbol{x} \in \mathcal{K}$ is far from the boundary of the ellipsoid, assuming that the closest point outside the ellipsoid, labeled $\boldsymbol{x}'$, belongs to $\mathcal{X}$. Claim 9.47 shows how to make this argument when $\boldsymbol{x}' \notin \mathcal{X}$ by considering instead a point $\boldsymbol{z} \in \mathcal{X}$ that is close to $\boldsymbol{x}'$.*

9.4.2.1 Strict Separation Oracle

The basic building block of our algorithm is what we refer to as a *strict* separation oracle—a strengthening of the second item of Definition 9.18:

Definition 9.42 (Strict separation)**.** Given a point $\boldsymbol{x} \in \mathbb{R}^d$ and a rational number $\gamma \in \mathbb{Q}_{>0}$, we say that a vector $\boldsymbol{c} \in \mathbb{Q}^d$, with $\|\boldsymbol{c}\|_\infty = 1$, *$\gamma$-strictly separates $\boldsymbol{x}'$* from a convex set $\mathcal{K}$ if $\langle \boldsymbol{c}, \boldsymbol{x}' \rangle \leq \langle \boldsymbol{c}, \boldsymbol{x} \rangle - \gamma$ for all $\boldsymbol{x} \in \mathcal{K}$.

The upshot is that a strict separation oracle for the set of MVIs can be indeed implemented in polynomial time assuming that the point to be separated is in $\mathcal{X}$ but is *not* an approximate SVI solution.

Lemma 9.43 (Semi-separation oracle)**.** *Given a point $\boldsymbol{a} \in \mathcal{X} \cap \mathbb{Q}^d$ and $\epsilon \in \mathbb{Q}_{>0}$, there is a polynomial-time algorithm that either*

- *ascertains that $\boldsymbol{a}$ is an ϵ-SVI solution; or*
- *returns $\boldsymbol{c} \in \mathbb{Q}^d$, with $\|\boldsymbol{c}\|_\infty = 1$, such that $\langle \boldsymbol{c}, \boldsymbol{x} \rangle \leq \langle \boldsymbol{c}, \boldsymbol{a} \rangle - \gamma$ for any point $\boldsymbol{x}$ that satisfies the Minty VI (9.2), where $\gamma := \epsilon^2 L B^{-1}/(B + 4RL)^2$.*

We proceed with the proof of this lemma. We first determine whether $\boldsymbol{a} \in \mathcal{X}$ is an ϵ-SVI solution; this can be done in polynomial time by invoking an optimization oracle for $\mathcal{X}$ (Definition 9.19). If so, the algorithm can terminate since $\boldsymbol{a}$ is an ϵ-SVI solution. Otherwise, we define $\tilde{\boldsymbol{a}} \in \mathbb{Q}^d$ per the gradient descent step $\Pi_{\mathcal{X}}(\boldsymbol{a} - \eta F(\boldsymbol{a}))$; such $\tilde{\boldsymbol{a}}$ guarantees the following, establishing Lemma 9.43.

Lemma 9.44. *Suppose that $\boldsymbol{a} \in \mathcal{X}$ is not an ϵ-SVI solution. If $\tilde{\boldsymbol{a}} := \Pi_{\mathcal{X}}(\boldsymbol{a} - \eta F(\boldsymbol{a}))$ with $\eta = 1/(2L)$, then $\langle F(\tilde{\boldsymbol{a}}), \boldsymbol{a} - \boldsymbol{x} \rangle \geq \gamma$ for any $\boldsymbol{x} \in \mathcal{X}$ that satisfies the Minty VI (9.2), where $\gamma := \epsilon^2 L/(B + 4RL)^2$. Furthermore, $\langle F(\tilde{\boldsymbol{a}}^{(t)}), \boldsymbol{a}^{(t)} - \tilde{\boldsymbol{a}}^{(t)} \rangle \geq \gamma$.*

Proof. By the first-order optimality conditions, we have

$$\left\langle F(\boldsymbol{a}) + \frac{1}{\eta}(\tilde{\boldsymbol{a}} - \boldsymbol{a}), \boldsymbol{a} - \tilde{\boldsymbol{a}} \right\rangle \geq 0,$$

which in turn implies that

$$\langle F(\boldsymbol{a}), \boldsymbol{a} - \tilde{\boldsymbol{a}} \rangle \geq \frac{1}{\eta}\|\boldsymbol{a} - \tilde{\boldsymbol{a}}\|^2. \tag{9.18}$$

Moreover, for any MVI solution $\boldsymbol{x} \in \mathcal{X}$,

$$\begin{aligned}
\langle F(\tilde{\boldsymbol{a}}), \boldsymbol{a} - \boldsymbol{x} \rangle &= \langle F(\tilde{\boldsymbol{a}}), \tilde{\boldsymbol{a}} - \boldsymbol{x} \rangle + \langle F(\tilde{\boldsymbol{a}}), \boldsymbol{a} - \tilde{\boldsymbol{a}} \rangle \\
&\geq \langle F(\tilde{\boldsymbol{a}}), \boldsymbol{a} - \tilde{\boldsymbol{a}} \rangle = \langle F(\boldsymbol{a}), \boldsymbol{a} - \tilde{\boldsymbol{a}} \rangle + \langle F(\tilde{\boldsymbol{a}}) - F(\boldsymbol{a}), \boldsymbol{a} - \tilde{\boldsymbol{a}} \rangle && (9.19) \\
&\geq \frac{1}{\eta}\|\boldsymbol{a} - \tilde{\boldsymbol{a}}\|^2 - \|F(\tilde{\boldsymbol{a}}) - F(\boldsymbol{a})\| \|\boldsymbol{a} - \tilde{\boldsymbol{a}}\| && (9.20) \\
&\geq \frac{1}{\eta}\|\boldsymbol{a} - \tilde{\boldsymbol{a}}\|^2 - L\|\boldsymbol{a} - \tilde{\boldsymbol{a}}\|^2 && (9.21) \\
&\geq \frac{1}{2\eta}\|\boldsymbol{a} - \tilde{\boldsymbol{a}}\|^2, && (9.22)
\end{aligned}$$

where (9.19) uses the fact that $\boldsymbol{x}$ satisfies (9.2); (9.20) follows from (9.18) and Cauchy-Schwarz; and (9.21) uses the fact that F is L-Lipschitz continuous. Finally, using again the first-order optimality conditions, we have that for any $\boldsymbol{x} \in \mathcal{X}$,

$$\left\langle F(\boldsymbol{a}) + \frac{1}{\eta}(\tilde{\boldsymbol{a}} - \boldsymbol{a}), \boldsymbol{x} - \tilde{\boldsymbol{a}} \right\rangle \geq 0 \implies \langle F(\boldsymbol{a}), \boldsymbol{x} - \boldsymbol{a} \rangle + \langle F(\boldsymbol{a}), \boldsymbol{a} - \tilde{\boldsymbol{a}} \rangle + \frac{1}{\eta}\langle \tilde{\boldsymbol{a}} - \boldsymbol{a}, \boldsymbol{x} - \tilde{\boldsymbol{a}} \rangle \geq 0. \tag{9.23}$$

But $\boldsymbol{a} \in \mathcal{X}$ is not an ϵ-SVI solution, which implies that there exists $\boldsymbol{x} \in \mathcal{X}$ such that $\langle F(\boldsymbol{a}), \boldsymbol{x} - \boldsymbol{a} \rangle < -\epsilon$. So, continuing from (9.23),

$$\epsilon < \langle F(\boldsymbol{a}), \boldsymbol{a} - \tilde{\boldsymbol{a}} \rangle + \frac{1}{\eta}\langle \tilde{\boldsymbol{a}} - \boldsymbol{a}, \boldsymbol{x} - \tilde{\boldsymbol{a}} \rangle \leq B\|\boldsymbol{a} - \tilde{\boldsymbol{a}}\| + \frac{2R}{\eta}\|\boldsymbol{a} - \tilde{\boldsymbol{a}}\|,$$

which in turn yields

$$\|\boldsymbol{a} - \tilde{\boldsymbol{a}}\| \geq (B + 4LR)^{-1}\epsilon.$$

Combining with (9.22), the proof follows. □

Lemma 9.43 now follows from Lemma 9.44 since $\|F(\tilde{\boldsymbol{a}})\| \leq B$.

The algorithm. We are now ready to describe our main construction, given as ExtraGradientEllipsoid. It is based on the central-ellipsoid we saw in Ellipsoid. In every iteration, it checks whether the center of the current ellipsoid is an ϵ-SVI solution. If so, the algorithm can terminate (Algorithm 9.5). Otherwise, if the center of the current ellipsoid lies in $\mathcal{X}$, it proceeds by producing a γ-strict separating hyperplane with respect to the set of MVIs (Algorithm 9.5)—by taking an extra-gradient step per Lemma 9.44. If the center does not belong to $\mathcal{X}$, it suffices to invoke the

Algorithm 9.5 (ExtraGradientEllipsoid): Efficient algorithm for SVIs under the Minty condition

```
input:
    oracle access to a convex, compact set $\mathcal{X} \subseteq \mathcal{B}_R(\mathbf{0})$ in isotropic position;
    oracle access to $F : \mathcal{X} \to \mathbb{R}^d$ satisfying Assumption 9.20;
    rational $\epsilon > 0$.
output: An $\epsilon$-SVI solution per Definition 9.1.
set the strictness parameter $\gamma := \epsilon^2 L/(B + 4RL)^2$
set the termination volume $v := r^d/d^d$, where $r := \gamma/(16RB)$
set the maximum number of iterations as $T := \lceil 5d \log(1/v) + 5d^2 \log(2R) \rceil$
initialize Ellipsoid with initial ellipsoid $\mathcal{B}_R(\mathbf{0}) \subset \mathbb{R}^d$ and target volume $v$
for $t = 0, \dots, T - 1$ do        ▷ $T$ is set as in Ellipsoid
    $\boldsymbol{a}^{(t)} \leftarrow$ current Ellipsoid center
    if $\boldsymbol{a}^{(t)}$ is an $\epsilon$-SVI solution then return $\boldsymbol{a}^{(t)}$
    else if $\boldsymbol{a}^{(t)} \notin \mathcal{X}$ then let $\boldsymbol{c} \in \mathbb{Q}^d$ be a hyperplane separating $\boldsymbol{a}^{(t)}$ from $\mathcal{X}$
    else
        compute $\tilde{\boldsymbol{a}}^{(t)} := \Pi_{\mathcal{X}}(\boldsymbol{a}^{(t)} - \eta F(\boldsymbol{a}^{(t)}))$, where $\eta := 1/(2L)$
        set $\boldsymbol{c} := F(\tilde{\boldsymbol{a}}^{(t)})/\|F(\tilde{\boldsymbol{a}}^{(t)})\|$
    pass $\boldsymbol{c}$ as the separating direction to Ellipsoid
return "there are no MVI solutions"
```

separation oracle of $\mathcal{X}$ (Algorithm 9.5). The algorithm continues by updating the ellipsoid.

Having analyzed our semi-separation oracle (Lemma 9.43), we conclude the analysis by showing that the number of iterations prescribed in Algorithm 9.5, which is polynomial in all relevant parameters, suffices to identify an ϵ-SVI solution.

9.4.2.2 Putting Everything Together

The set of MVIs, denoted by $\mathcal{K} \neq \emptyset$, is generally not fully dimensional; nevertheless, by virtue of having a *strict* separating hyperplane throughout the execution of ExtraGradientEllipsoid (whenever the center of the ellipsoid belongs to $\mathcal{X}$), we will show that the volume of the ellipsoid can indeed by used as a yardstick to track the progress of the algorithm. The basic idea is that every axis of the ellipsoid needs to have a non-trivial length (Figure 9.4)—as dictated by the strictness parameter γ, thereby implying that the volume of the ellipsoid cannot shrink too much; formally, we show the following.

Lemma 9.45. *Suppose that $\mathcal{K} \neq \emptyset$—that is, the Minty condition (Assumption 9.3) holds. For any t during the execution of* ExtraGradientEllipsoid*, the ellipsoid $\mathcal{E}^{(t)}$ contains a (Euclidean) ball of radius $r := \gamma/(16RB)$, where $\gamma > 0$ is the strictness parameter per Lemma 9.44.*

We are now ready to complete the proof of correctness of ExtraGradientEllipsoid, summarized in the theorem below.

Theorem 9.46. *Let $\mathcal{X}$ be a convex and compact set in isotropic position to which we have oracle access, $F : \mathcal{X} \to \mathbb{R}^d$ a mapping satisfying Assumption 9.20, and $\epsilon \in \mathbb{Q}$ a rational number. Under the Minty condition (Assumption 9.3),* ExtraGradientEllipsoid *can be implemented in time $\mathrm{poly}(d, \log(B/\epsilon), \log L)$ and returns an ϵ-SVI solution to* $\mathrm{VI}(\mathcal{X}, F)$.

Proof. That ExtraGradientEllipsoid can be implemented in time $\mathrm{poly}(d, \log(B/\epsilon), \log L)$ is immediate. We thus focus on proving correctness. For the sake of contradiction, suppose that the algorithm never identified an ϵ-SVI solution. The volume of a d-dimensional Euclidean ball with radius $r > 0$ is given by

$$\frac{\pi^{d/2}}{\Gamma(\frac{d}{2}+1)} r^d > \frac{1}{d^d} r^d,$$

where $\Gamma(\cdot)$ is the gamma function. By our choice of parameters in Algorithm 9.5 and Theorem 9.40, it follows that the short axis of the Tth ellipsoid will have radius strictly smaller than $r = \gamma/(16RB)$. Let $\boldsymbol{x} \in \mathcal{X}$ be any point inside the final ellipsoid and $\boldsymbol{c}$ the unit vector in the direction of the short axis of the ellipsoid. We will show that $\boldsymbol{x}$ strictly violates the MVI constraint. Thus, assuming $\mathcal{K} \neq \emptyset$, this will imply that the algorithm must have terminated at some earlier iteration with an ϵ-SVI solution.

Let $\mathcal{Z}$ be the union of the two $(d-1)$-dimensional disk of points $\boldsymbol{z}$ lying in the planes $\langle \boldsymbol{c}, \boldsymbol{z} - \boldsymbol{x} \rangle = \pm 2r$, and within $r' := \gamma/(4B)$ of $\boldsymbol{x}$. That is,

$$\mathcal{Z} = \left\{ \boldsymbol{z} \in \mathbb{R}^d : |\langle \boldsymbol{c}, \boldsymbol{z} - \boldsymbol{x} \rangle| = 2r, \|\boldsymbol{x} - \boldsymbol{z}\| \leq r' \right\}.$$

Claim 9.47. *$\mathcal{Z}$ intersects $\mathcal{X}$.*

Proof. Since $\mathcal{X}$ contains a ball of radius 1, there must be a point $\boldsymbol{y} \in \mathcal{X}$ such that $|\langle \boldsymbol{c}, \boldsymbol{y} - \boldsymbol{x} \rangle| = 1$. Assume $\langle \boldsymbol{c}, \boldsymbol{y} - \boldsymbol{x} \rangle = 1$ (The case $\langle \boldsymbol{c}, \boldsymbol{y} - \boldsymbol{x} \rangle = -1$ is symmetric). Let $\boldsymbol{z}$ be the point on line segment $[\boldsymbol{x}, \boldsymbol{y}]$ such that $\langle \boldsymbol{c}, \boldsymbol{z} - \boldsymbol{x} \rangle = 2r$, that is, let $\boldsymbol{z} = \boldsymbol{x} + 2r(\boldsymbol{y} - \boldsymbol{x})$. Since $\mathcal{X}$ is convex, we have $\boldsymbol{z} \in \mathcal{X}$. Since $\mathcal{X} \subseteq \mathcal{B}_R(0)$, we have $\|\boldsymbol{y} - \boldsymbol{x}\| \leq 2R$. Thus, $\|\boldsymbol{z} - \boldsymbol{x}\| \leq 2r \cdot 2R = \gamma/(4B) = r'$, so $\boldsymbol{z} \in \mathcal{Z}$. □

Lemma 9.48. *If the algorithm fails to return an ϵ-SVI solution after T rounds, where T is as defined in Algorithm 9.5, any point $\boldsymbol{x} \in \mathcal{X}$ strictly violates the MVI constraint. In particular, there exists a timestep t such that $\tilde{\boldsymbol{a}}^{(t)} \in \mathcal{X}$ and $\langle F(\tilde{\boldsymbol{a}}^{(t)}), \tilde{\boldsymbol{a}}^{(t)} - \boldsymbol{x} \rangle \leq -\gamma/2$.*

Proof. It suffices to consider $\boldsymbol{x} \in \mathcal{E}^{(T)}$. Let $\boldsymbol{z} \in \mathcal{Z} \cap \mathcal{X}$, which must exist by Claim 9.47. By definition, we have $\|\boldsymbol{z} - \boldsymbol{x}\| \leq r' < \gamma/(2B)$. Since the short axis of the final ellipsoid has radius less than r and $|\langle \boldsymbol{c}, \boldsymbol{z} - \boldsymbol{x} \rangle| = 2r$, it follows that $\boldsymbol{z}$ is not in the ellipsoid. Thus, at some point, there must have been a separating hyperplane that has $\boldsymbol{z}$ on the opposite side. This separating hyperplane could not have come from the separation oracle of $\mathcal{X}$, because

$\boldsymbol{z} \in \mathcal{X}$. Thus, it must have come from an extra-gradient step, *i.e.*, the separating hyperplane must have the form

$$\left\langle F(\tilde{\boldsymbol{a}}^{(t)}), \boldsymbol{z} - \boldsymbol{a}^{(t)} \right\rangle \geq 0.$$

for some timestep t. From Lemma 9.44 and the construction of $\tilde{\boldsymbol{a}}^{(t)}$, we also have $\left\langle F(\tilde{\boldsymbol{a}}^{(t)}), \boldsymbol{a}^{(t)} - \tilde{\boldsymbol{a}}^{(t)} \right\rangle \geq \gamma$. Thus, we have

$$\begin{aligned} \left\langle F(\tilde{\boldsymbol{a}}^{(t)}), \boldsymbol{x} - \tilde{\boldsymbol{a}}^{(t)} \right\rangle &= \left\langle F(\tilde{\boldsymbol{a}}^{(t)}), \boldsymbol{z} - \boldsymbol{a}^{(t)} \right\rangle + \left\langle F(\tilde{\boldsymbol{a}}^{(t)}), \boldsymbol{a}^{(t)} - \tilde{\boldsymbol{a}}^{(t)} \right\rangle + \left\langle F(\tilde{\boldsymbol{a}}^{(t)}), \boldsymbol{x} - \boldsymbol{z} \right\rangle \\ &\geq \gamma - B \cdot \frac{\gamma}{2B} \geq \frac{\gamma}{2}. \end{aligned}$$

□

This concludes the proof of Lemma 9.48, and Theorem 9.46 follows. □

9.4.3 Weak Minty Condition

Moving forward, we first observe that the previous analysis can be extended beyond the Minty condition (Assumption 9.3). In particular, we lean on the more permissive assumption put forward by Diakonikolas et al. [87]; we will shortly discuss how it relates to other conditions. Below, we use the notation $\mathsf{SVIGap} : \mathcal{X} \ni \boldsymbol{x} \mapsto \max_{\boldsymbol{x}' \in \mathcal{X}} \langle F(\boldsymbol{x}), \boldsymbol{x} - \boldsymbol{x}' \rangle$, so that Definition 9.1 can be equivalently expressed as $\mathsf{SVIGap}(\boldsymbol{x}) \leq \epsilon$.

Definition 9.49 (Weak Minty)**.** A problem $\mathrm{VI}(\mathcal{X}, F)$ satisfies the *ρ-weak Minty condition*, with $\rho > 0$, if there exists $\boldsymbol{x} \in \mathcal{X}$ such that

$$\langle F(\boldsymbol{x}'), \boldsymbol{x}' - \boldsymbol{x} \rangle \geq -\rho(\mathsf{SVIGap}(\boldsymbol{x}'))^2 \quad \forall \boldsymbol{x}' \in \mathcal{X}. \tag{9.24}$$

Diakonikolas et al. [87] focused on the unconstrained setting, positing that the right-hand side of (9.24) instead reads $-\rho\|F(\boldsymbol{x}')\|^2$ (we have removed a factor of 2 compared to the definition given by [87], which amounts to simply rescaling ρ); Definition 9.49 can be seen as the natural counterpart of that condition to the constrained setting. For the unconstrained setting, this is weaker than another well-studied condition; namely, F is called $(-\rho)$-*comonotone* [20] (see also cohypomonotone operators per [66]) if for all $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^d$, $\langle F(\boldsymbol{x}) - F(\boldsymbol{x}'), \boldsymbol{x} - \boldsymbol{x}' \rangle \geq -\rho\|F(\boldsymbol{x}) - F(\boldsymbol{x}')\|^2$; since $F(\boldsymbol{x}) = 0$ for any SVI solution (in the unconstrained setting), the condition of Diakonikolas et al. [87] is weaker.

In this context, the purpose of this subsection is to show that our previous analysis can be readily extended under Definition 9.49—in place of Assumption 9.3. For completeness, we begin with a simple claim showing that the set of points satisfying the ρ-weak Minty condition is convex.

Claim 9.50. *Let $\mathcal{K}$ be the set of solutions to* (9.24)*. $\mathcal{K}$ is a convex set.*

Proof. Suppose $\mathcal{K} \neq \emptyset$. Let $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{K}$ and $\lambda \in [0, 1]$. We need to show that for all $\boldsymbol{x}' \in \mathcal{X}$,

$$\lambda \langle F(\boldsymbol{x}'), \boldsymbol{x}' - \boldsymbol{x}_1 \rangle + (1 - \lambda) \langle F(\boldsymbol{x}'), \boldsymbol{x}' - \boldsymbol{x}_2 \rangle \geq -\rho(\mathsf{SVIGap}(\boldsymbol{x}'))^2.$$

This follows since $\langle F(\boldsymbol{x}'), \boldsymbol{x}' - \boldsymbol{x}_1 \rangle \geq -\rho(\mathsf{SVIGap}(\boldsymbol{x}'))^2$ and $\langle F(\boldsymbol{x}'), \boldsymbol{x}' - \boldsymbol{x}_2 \rangle \geq -\rho(\mathsf{SVIGap}(\boldsymbol{x}'))^2$.

□

Now, to show that an ϵ-SVI solution can be computed in polynomial time even under the weak Minty condition, for small enough ρ, it suffices to adjust Lemma 9.44 according to the statement below.

Lemma 9.51. *Suppose that $\boldsymbol{a} \in \mathcal{X}$ is not an ϵ-SVI solution. Suppose further that $\tilde{\boldsymbol{a}} := \Pi_{\mathcal{X}}(\boldsymbol{a} - \eta F(\boldsymbol{a}))$ is also not an ϵ-SVI solution. Then, $\langle F(\tilde{\boldsymbol{a}}), \boldsymbol{a} - \boldsymbol{x}\rangle \geq \epsilon^2 L/(B + 4RL)^2 - \rho\epsilon^2$ for any $\boldsymbol{x} \in \mathcal{X}$ that satisfies the ρ-weak Minty condition of Definition 9.49.*

The proof is similar to that of Lemma 9.44; unlike Lemma 9.44, Lemma 9.51 further assumes that $\tilde{\boldsymbol{a}} \in \mathcal{X}$ is not an ϵ-SVI solution, which can be again ascertained during the execution of the algorithm by invoking a linear optimization oracle. Assuming that $B = L$, Lemma 9.51 yields a strict separating hyperplane when $\rho \leq C(1 + 4R)^{-2}/L$ for some constant $C < 1$. The rest of the argument is analogous to Theorem 9.46.

Theorem 9.52. *Let $\mathcal{X}$ be a convex and compact set in isotropic position to which we have oracle access, $F : \mathcal{X} \to \mathbb{R}^d$ a mapping satisfying Assumption 9.20, and $\epsilon \in \mathbb{Q}$ a rational number. Under the ρ-weak Minty condition (Definition 9.49) with $\rho \leq CL/(B + 4RL)^2$, for some constant $C < 1$, there is a $\mathsf{poly}(d, \log(B/\epsilon), \log L)$-time algorithm that returns an ϵ-SVI solution to $\mathrm{VI}(\mathcal{X}, F)$.*

9.4.4 Relaxing Continuity

Another natural question is whether one can relax the assumption that F is Lipschitz continuous. Under an additional assumption—namely, a generalization of quasar-convexity (Definition 9.24) based on Definition 9.26—we show that this is indeed possible by obviating the need for the extra-gradient step in ExtraGradientEllipsoid (Lemma 9.44), which is where Lipschitz continuity was used. In particular, the following lemma shows that, under a suitable strengthening of the Minty condition, $F(\boldsymbol{a})$ already yields a strict separating hyperplane. We again call attention to the fact that smoothness per Definition 9.26, which accords with the terminology of Roughgarden [256], is different from the usual notion of a smooth function in optimization.

Lemma 9.53. *Let $\mathrm{VI}(\mathcal{X}, F)$ be a $(\lambda, \lambda - 1)$-smooth VI problem (Definition 9.26) with respect to Q and a global maximizer $\boldsymbol{x} \in \mathcal{X}$ thereof. If $\boldsymbol{a} \in \mathcal{X}$ is such that $Q(\boldsymbol{a}') \leq Q(\boldsymbol{x}) - \epsilon$, then*

$$\langle F(\boldsymbol{a}), \boldsymbol{a} - \boldsymbol{x}\rangle \geq \lambda\epsilon.$$

Proof. By Definition 9.26 (with $\nu = \lambda - 1$), we have $\langle F(\boldsymbol{a}), \boldsymbol{a} - \boldsymbol{x}\rangle \geq \lambda(Q(\boldsymbol{x}) - Q(\boldsymbol{a})) \geq \lambda\epsilon$. □

In this case, we define $\mathcal{K}$ to contain any point $\boldsymbol{x} \in \mathcal{X}$ such that $\langle F(\boldsymbol{x}'), \boldsymbol{x}' - \boldsymbol{x}\rangle \geq \lambda(\mathsf{OPT} - Q(\boldsymbol{x}'))$ for all $\boldsymbol{x}' \in \mathcal{X}$, where OPT is the maximum value attained by Q. This is still a convex set. Further, $\mathcal{K} \neq \emptyset$—in particular, this means that the Minty condition is satisfied.

We give the overall construction in SmoothVIEllipsoid. Compared to ExtraGradientEllipsoid, we call attention to the following differences. First, we do not check in each iteration t whether the center of the current ellipsoid $\boldsymbol{a}^{(t)}$ satisfies $Q(\boldsymbol{a}^{(t)}) \geq \mathsf{OPT} - \epsilon$; we do not know the value of OPT,

Algorithm 9.6 (SmoothVIEllipsoid): Ellipsoid for $(\lambda, \lambda - 1)$-smooth VIs.

```
input:
    oracle access to a convex, compact set $\mathcal{X} \subseteq \mathcal{B}_R(\mathbf{0})$ in isotropic position;
    oracle access to $F : \mathcal{X} \to \mathbb{R}^d$ satisfying Items 1 and 3 of Assumption 9.20;
    oracle access to $Q : \mathcal{X} \to \mathbb{R}$ such that $\langle F(\boldsymbol{x}'), \boldsymbol{x}' - \boldsymbol{x} \rangle \geq \lambda(Q(\boldsymbol{x}) - Q(\boldsymbol{x}')$ for all $\boldsymbol{x}' \in \mathcal{X}$,
    where $\boldsymbol{x} \in \mathcal{X}$ is some global maximum of $Q$ and $\lambda \in \mathbb{Q} \cap (0, 1]$;
    rational $\epsilon > 0$.
output: a point $\boldsymbol{a} \in \mathcal{X}$ such that $Q(\boldsymbol{a}) \geq \max_{\boldsymbol{x} \in \mathcal{X}} Q(\boldsymbol{x}) - \epsilon$.
set the strictness parameter $\gamma := \lambda\epsilon$
set the termination volume $v := r^d / d^d$, where $r := \gamma/(16RB)$
initialize Ellipsoid with initial ellipsoid $\mathcal{B}_R(\mathbf{0}) \subset \mathbb{R}^d$ and target volume $v$
for $t = 0, \dots, T - 1$ do        ▷ $T$ is set as in Ellipsoid
    $\boldsymbol{a}^{(t)} \leftarrow$ current Ellipsoid center
    if $\boldsymbol{a}^{(t)} \notin \mathcal{X}$ then set $\boldsymbol{c} \in \mathbb{Q}^d$ be a hyperplane separating $\boldsymbol{a}^{(t)}$ from $\mathcal{X}$
    elseset $\boldsymbol{c} := F(\boldsymbol{a}^{(t)}) / \|F(\boldsymbol{a}^{(t)})\|$
    pass $\boldsymbol{c}$ as the separating direction to Ellipsoid
return $\operatorname{argmax}_{1 \leq t \leq T} Q(\boldsymbol{a}^{(t)})$
```

so instead the algorithm eventually returns the point attaining the highest value throughout the execution (Algorithm 9.6). The second difference is that $F(\boldsymbol{a}^{(t)})$ (Algorithm 9.6) already yields a strict separating hyperplane (Lemma 9.53), so there is no need for an extra-gradient step.

By our previous analysis in Section 9.4.2 and Lemma 9.53, it follows that there must be some iteration such that $Q(\boldsymbol{a}^{(t)}) \geq \mathsf{OPT} - \epsilon$, for otherwise the underlying promise—namely, $(\lambda, \lambda - 1)$-smoothness per Definition 9.26—would be violated; Algorithm 9.6 returns such a point. We summarize the guarantee of SmoothVIEllipsoid below.

Theorem 9.54. *Let $\mathcal{X}$ be a convex and compact set in isotropic position to which we have oracle access, $F : \mathcal{X} \to \mathbb{R}^d$ a mapping satisfying Items 1 and 3 of Assumption 9.20, and $\epsilon \in \mathbb{Q}$ a rational number. If* $\mathrm{VI}(\mathcal{X}, F)$ *is* $(\lambda, \lambda - 1)$*-smooth (Definition 9.26) with respect to Q, SmoothVIEllipsoid can be implemented in time* $\mathsf{poly}(d, \log(B/\epsilon), \log(1/\lambda))$ *and returns a point $\boldsymbol{a} \in \mathcal{X}$ such that $Q(\boldsymbol{a}) \geq \max_{\boldsymbol{x} \in \mathcal{X}} Q(\boldsymbol{x}) - \epsilon$.*

9.4.5 A Certificate of MVI Infeasibility: Strict EVIs

Computing ϵ-SVI solutions is generally PPAD-hard, so certain VI problems violate the Minty conditions (and relaxations thereof per Section 9.4.3). In such cases, the transcript of ExtraGradientEllipsoid itself provides a certificate of MVI infeasibility. In fact, as we observe in this subsection, the certificate of infeasibility can be expressed as an expected VI solution (in the sense of Definition 9.29) *with negative gap*; the mere existence of such an object implies that the Minty condition is violated due to the duality between EVIs and MVIs (Proposition 9.30).

To make this argument formal, we first observe that if ExtraGradientEllipsoid fails to identify an

Algorithm 9.7 (SVIorStrictEVI): Augment ExtraGradientEllipsoid to return a certificate of strict Minty infeasibility in case of failure

```
input:
    Oracle access to a convex, compact set $\mathcal{X} \subseteq \mathcal{B}_R(\mathbf{0})$ in isotropic position;
    oracle access to $F : \mathcal{X} \to \mathbb{R}^d$ satisfying Assumption 9.20;
    rational $\epsilon > 0$.
output: An $\epsilon$-SVI solution per Definition 9.1 or a strict EVI solution per Definition 9.29.
run ExtraGradientEllipsoid
if ExtraGradientEllipsoid returns an $\epsilon$-SVI $\boldsymbol{a}^{(t)}$ then return $\boldsymbol{a}^{(t)}$
else return $\operatorname{argmax}_{\mu \in \Delta([T])} \min_{\boldsymbol{x} \in \mathcal{X}} \mathbb{E}_{t \sim \mu} \langle F(\tilde{\boldsymbol{a}}^{(t)}), \boldsymbol{x} - \tilde{\boldsymbol{a}}^{(t)} \rangle$
▷ $T$ and $\tilde{\mathbf{a}}^{(t)}$ are as in ExtraGradientEllipsoid
```

ϵ-SVI solution, Lemma 9.48 implies that

$$\min_{\boldsymbol{x} \in \mathcal{X}} \max_{t \in [T]} \langle F(\tilde{\boldsymbol{a}}^{(t)}), \boldsymbol{x} - \tilde{\boldsymbol{a}}^{(t)} \rangle \geq \frac{\gamma}{2},$$

which, by strong duality, is equivalent to

$$\max_{\mu \in \Delta(T)} \min_{\boldsymbol{x} \in \mathcal{X}} \mathbb{E}_{t \sim \mu} \langle F(\tilde{\boldsymbol{a}}^{(t)}), \boldsymbol{x} - \tilde{\boldsymbol{a}}^{(t)} \rangle \geq \frac{\gamma}{2}. \tag{9.25}$$

In particular, a distribution μ over $[T]$ that satisfies (9.25) is, by definition, a $\gamma/2$-strict EVI solution, a certificate of MVI infeasibility. Further, such a distribution can be computed in polynomial time (*e.g.*, [161]). The crucial point here is that the maximization in (9.25) simply optimizes over the mixing weights with respect to a fixed support of size T, which is polynomial. This approach resembles the celebrated "ellipsoid against hope" algorithm of Papadimitriou and Roughgarden [237], which applies the ellipsoid algorithm on a certain program guaranteed to be infeasible; similarly to our case, the certificate of infeasibility produces a correlated equilibrium.

The resulting construction is SVIorStrictEVI. It is almost the same as ExtraGradientEllipsoid; the key difference is that, when the algorithm fails to identify an ϵ-SVI solution, the last step (Algorithm 9.7) performs an additional optimization to compute a $\gamma/2$-strict EVI solution. The guarantee of SVIorStrictEVI is given below; it is the precise version of Theorem 9.8.

Theorem 9.55. *Let $\mathcal{X}$ be a convex and compact set in isotropic position to which we have oracle access, a mapping $F : \mathcal{X} \to \mathbb{R}^d$ satisfying Assumption 9.20, and a rational number $\epsilon \in \mathbb{Q}^d$. SVIorStrictEVI can be implemented in time $\operatorname{poly}(d, \log(B/\epsilon), \log L)$ and returns either*

- *an ϵ-SVI solution to $\mathrm{VI}(\mathcal{X}, F)$ or*
- *an $\Omega_\epsilon(\epsilon^2)$-strict EVI solution to $\mathrm{VI}(\mathcal{X}, F)$.*

9.5 Two-player Smooth Concave Games

This section provides a refinement of Theorem 9.55 in the context of two-player games. We begin by formally introducing this class of problems.

Definition 9.56 (Two-player smooth concave games)**.** A *two-player concave game* is given by two convex and compact strategy sets $\mathcal{X} \subseteq \mathbb{R}^{d_1}$, $\mathcal{Y} \subseteq \mathbb{R}^{d_2}$, and two utility functions $u_1, u_2 : \mathcal{X}, \mathcal{Y} \to \mathbb{R}$, such that the utility of each player is concave (in the player's own strategy), differentiable, and L-smooth. Formally, $u_1(\cdot, \boldsymbol{y}) : \mathcal{X} \to \mathbb{R}$ is concave for every fixed $\boldsymbol{y}$, the gradient operator $\nabla_{\boldsymbol{x}} u_1 : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^{d_1}$ is L-Lipschitz continuous, and the symmetric guarantees hold for the second player as well.

A *(pure) strategy profile* is a pair $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{X} \times \mathcal{Y}$. A strategy profile is an ϵ*-Nash equilibrium* if each player is ϵ-best responding to the other player; that is,

$$\max_{\boldsymbol{x}'} u_1(\boldsymbol{x}', \boldsymbol{y}) - u_1(\boldsymbol{x}, \boldsymbol{y}) \le \epsilon \quad \text{and} \quad \max_{\boldsymbol{y}'} u_2(\boldsymbol{x}, \boldsymbol{y}') - u_2(\boldsymbol{x}, \boldsymbol{y}) \le \epsilon.$$

A *correlated strategy profile* is a distribution $\mu \in \Delta(\mathcal{X} \times \mathcal{Y})$. A correlated strategy profile is a ϵ*-coarse correlated equilibrium* (CCE) if no player can profit by more than ϵ using a *unilateral* deviation, that is,

$$\max_{\boldsymbol{x}' \in \mathcal{X}} \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{y}) \sim \mu} [u_1(\boldsymbol{x}', \boldsymbol{y}) - u_1(\boldsymbol{x}, \boldsymbol{y})] \le \epsilon \quad \text{and} \quad \max_{\boldsymbol{y}' \in \mathcal{Y}} \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{y}) \sim \mu} [u_2(\boldsymbol{x}, \boldsymbol{y}') - u_2(\boldsymbol{x}, \boldsymbol{y})] \le \epsilon.$$

We will call a $(-\epsilon)$-CCE an ϵ*-strict CCE*. In this section, we will show the following result for smooth concave games.

Theorem 9.57. *Assume that $\mathcal{X}$ and $\mathcal{Y}$ are in isotropic position and given by separation oracles. Assume also the gradient operators $\nabla_{\boldsymbol{x}} u_1 : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^{d_1}$ and $\nabla_{\boldsymbol{y}} u_2 : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^{d_2}$ satisfy Assumption 9.20. Let $d = d_1 + d_2$. Then there exists a $\mathsf{poly}(d) \cdot \mathsf{polylog}(B, L, R, 1/\epsilon)$-time algorithm that outputs* either *an ϵ-Nash equilibrium* or *an ϵ'-strict CCE, where $\epsilon' \ge \epsilon^4 / \mathsf{poly}(B, L, R)$.*

We first note that this result is *not* an immediate corollary of Theorem 9.55. ϵ-Nash equilibria indeed correspond to ϵ-SVI solutions. However, as per the discussion in Section 9.3, Theorem 9.55 would only give a strict *average* CCE (Definition 9.31), not a strict CCE. Circumventing this problem therefore requires a few new insights.

Proof. First, we need to modify the SVI problem so that strict EVI solutions correspond to strict CCEs. Consider the set

$$\mathcal{P} := \left\{ \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_1 \boldsymbol{x} \\ \lambda_2 \boldsymbol{y} \end{pmatrix} : \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} \in \Delta(2), \boldsymbol{x} \in \mathcal{X}, \boldsymbol{y} \in \mathcal{Y} \right\} \subseteq \mathbb{R}^{2+d}.$$

It is easy to see that $\mathcal{P}$ is convex and compact, since $\mathcal{X}$ and $\mathcal{Y}$ are. Moreover, consider the operator $G : \mathcal{P} \to \mathbb{R}^{2+d}$ given by

$$G\begin{pmatrix}\lambda_1\\ \lambda_2\\ \lambda_1\boldsymbol{x}\\ \lambda_2\boldsymbol{y}\end{pmatrix} := \begin{pmatrix}\langle \nabla_{\boldsymbol{x}} u_1(\boldsymbol{x},\boldsymbol{y}),\boldsymbol{x}\rangle\\ \left\langle \nabla_{\boldsymbol{y}} u_2(\boldsymbol{x},\boldsymbol{y}),\boldsymbol{y}\right\rangle\\ -\nabla_{\boldsymbol{x}} u_1(\boldsymbol{x},\boldsymbol{y})\\ -\nabla_{\boldsymbol{y}} u_2(\boldsymbol{x},\boldsymbol{y})\end{pmatrix}.$$

We now immediately encounter our first problem: G is not Lipschitz, or even well-defined, when $\lambda_1 = 0$ or $\lambda_1 = 1$. The solution to this problem is to restrict ourselves to the slightly smaller set

$$\mathcal{P}_\alpha := \left\{\begin{pmatrix}\lambda_1\\ \lambda_2\\ \lambda_1\boldsymbol{x}\\ \lambda_2\boldsymbol{y}\end{pmatrix} : \begin{pmatrix}\lambda_1\\ \lambda_2\end{pmatrix} \in \Delta(2)\cap\mathbb{R}^2_{\geq\alpha}, \boldsymbol{x}\in\mathcal{X}, \boldsymbol{y}\in\mathcal{Y}\right\}$$

over which $G : \mathcal{P}_\alpha \to \mathbb{R}^{2+d}$ is indeed well-defined, bounded, and $L' := \mathsf{poly}(B, L, R)/\alpha$-Lipschitz.

We start by noting that, for any $\boldsymbol{z} = (\lambda_1, \lambda_2, \lambda_1\boldsymbol{x}, \lambda_2\boldsymbol{y}), \boldsymbol{z}' = (\lambda_1', \lambda_2', \lambda_1'\boldsymbol{x}', \lambda_2'\boldsymbol{y}') \in \mathcal{P}$, we have

$$\begin{aligned}-\langle G(\boldsymbol{z}),\boldsymbol{z}'\rangle &= \begin{pmatrix}\langle \nabla_{\boldsymbol{x}} u_1(\boldsymbol{x},\boldsymbol{y}),\boldsymbol{x}\rangle\\ \left\langle \nabla_{\boldsymbol{y}} u_2(\boldsymbol{x},\boldsymbol{y}),\boldsymbol{y}\right\rangle\\ -\nabla_{\boldsymbol{x}} u_1(\boldsymbol{x},\boldsymbol{y})\\ -\nabla_{\boldsymbol{y}} u_2(\boldsymbol{x},\boldsymbol{y})\end{pmatrix}^{\top}\begin{pmatrix}\lambda_1'\\ \lambda_2'\\ \lambda_1'\boldsymbol{x}'\\ \bar{\lambda}_1'\boldsymbol{y}'\end{pmatrix}\\ &= \lambda_1'\langle \nabla_{\boldsymbol{x}} u_1(\boldsymbol{x},\boldsymbol{y}),\boldsymbol{x}'-\boldsymbol{x}\rangle + \lambda_2'\left\langle \nabla_{\boldsymbol{y}} u_y(\boldsymbol{x},\boldsymbol{y}),\boldsymbol{y}'-\boldsymbol{y}\right\rangle.\end{aligned}$$

That is, $-\langle G(\boldsymbol{z}),\boldsymbol{z}'\rangle$ is the weighted sum of deviation benefits for the two players if they deviate to $\boldsymbol{z}'$ from $\boldsymbol{z}$. In particular, $\langle G(\boldsymbol{z}),\boldsymbol{z}\rangle = 0$ for all $\boldsymbol{z}$.

Lemma 9.58. *Any $2\alpha BR$-strict EVI solution to $\mathrm{VI}(\mathcal{P}_\alpha, G)$ is a αBR-strict CCE.*

Proof. Let $\mu \in \Delta(\mathcal{P}_\alpha)$ be a 2α-strict EVI solution. Then, for any $\boldsymbol{z}' = (\lambda_1', \lambda_2'\lambda_1'\boldsymbol{x}', \lambda_2'\boldsymbol{y}') \in \mathcal{P}_\alpha$, we have

$$-2\alpha \geq \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mu}\langle G(\boldsymbol{z}),\boldsymbol{z}-\boldsymbol{z}'\rangle = \lambda_1' \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mu}\langle \nabla_{\boldsymbol{x}} u_1(\boldsymbol{x},\boldsymbol{y}),\boldsymbol{x}'-\boldsymbol{x}\rangle + \lambda_2' \mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mu}\left\langle \nabla_{\boldsymbol{y}} u_y(\boldsymbol{x},\boldsymbol{y}),\boldsymbol{y}'-\boldsymbol{y}\right\rangle.$$

Since this holds for any $\boldsymbol{z}'$, in particular it holds if we set $\boldsymbol{\lambda}' = (1-\alpha, \alpha)$, which gives

$$(1-\alpha)\mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mu}\langle \nabla_{\boldsymbol{x}} u_1(\boldsymbol{x},\boldsymbol{y}),\boldsymbol{x}'-\boldsymbol{x}\rangle \leq -2\alpha BR + \alpha\mathop{\mathbb{E}}_{\boldsymbol{z}\sim\mu}\left\langle \nabla_{\boldsymbol{y}} u_y(\boldsymbol{x},\boldsymbol{y}),\boldsymbol{y}'-\boldsymbol{y}\right\rangle \leq -\alpha BR,$$

where the final inequality follows from Cauchy-Schwarz. Dividing by $1-\alpha$ and noting that $1-\alpha \leq 1$ completes the proof. □

It may now be tempting to try to run ExtraGradientEllipsoid on $\mathrm{VI}(\mathcal{P}_\alpha, G)$ with ϵ large enough to recover $2\alpha BR$-strict EVI solutions via Theorem 9.55. Unfortunately, this is impossible, since the Lipschitz constant L' of G scales as $O_\alpha(1/\alpha)$. So, we would be required to take ϵ to be of (game-dependent) constant size. To account for this, we directly modify the semi-separation oracle (Lemma 9.43). Intuitively, the problematic case is when λ_1 or λ_2 is close to 0. For intuition, let us discuss an extreme case, $\lambda_2 = 0$.[9.11] Let $\boldsymbol{a} = (1, 0, \boldsymbol{x}, \boldsymbol{0}) \in \mathcal{P}$. Our goal is to either (1) find a Nash equilibrium, or (2) separate $\boldsymbol{a}$ strictly from the set of Minty solutions to $\mathrm{VI}(\mathcal{P}_\alpha, G)$. Consider the following process. Let $\boldsymbol{y}'$ be a best response to $\boldsymbol{x}$. If $\boldsymbol{x}$ is also a best response to $\boldsymbol{y}'$, then $(\boldsymbol{x}, \boldsymbol{y}')$ is a Nash equilibrium, and we are done. Otherwise, let $\boldsymbol{x}' = \Pi_{\mathcal{X}}(\boldsymbol{x} + \eta \nabla_{\boldsymbol{x}} u_1(\boldsymbol{x}, \boldsymbol{y}'))$. Then, any point of the form $\boldsymbol{a}' := (\lambda_1, \lambda_2, \lambda_1 \boldsymbol{x}', \lambda_2 \boldsymbol{y}')$ with $\lambda_1, \lambda_2 > 0$ certifies that $\boldsymbol{a}$ is not Minty: the Minty constraint induced by $\boldsymbol{a}'$ is

$$\langle G(\boldsymbol{a}'), \boldsymbol{a}' - \boldsymbol{a} \rangle = \langle \nabla_{\boldsymbol{x}} u_1(\boldsymbol{x}', \boldsymbol{y}'), \boldsymbol{x} - \boldsymbol{x}' \rangle \geq 0,$$

but this constraint can be refuted by choosing the step size η small enough that $\nabla_{\boldsymbol{x}} u_1(\boldsymbol{x}', \boldsymbol{y}') \approx \nabla_{\boldsymbol{x}} u_1(\boldsymbol{x}, \boldsymbol{y}')$. We now formalize this intuition.

Lemma 9.59. *There is a polynomial-time algorithm that, given $\boldsymbol{a} = (\lambda_1, \lambda_2, \lambda_1 \boldsymbol{x}, \lambda_2 \boldsymbol{y}) \in \mathcal{P}_\alpha$ and $\epsilon > 0$, either*

- *outputs an ϵ-Nash equilibrium (not necessarily $(\boldsymbol{x}, \boldsymbol{y})$), or*
- *returns $\boldsymbol{a}' \in \mathcal{P}_\alpha$ such that $\langle G(\boldsymbol{a}'), \boldsymbol{a}' - \boldsymbol{a} \rangle \geq \gamma' := \epsilon^4 / \mathsf{poly}(B, L, R)$.*

Proof. We split the analysis into two cases.

Case 1. $\min\{\lambda_1, \lambda_2\} \leq \gamma/(4BR)$. Assume without loss of generality that $\lambda_2 \leq \gamma/(4BR)$ (and $\lambda_2 \leq 1/2$). Compute a best response $\boldsymbol{y}'$ to $\boldsymbol{x}$, and check if $(\boldsymbol{x}, \boldsymbol{y}')$ is an ϵ-Nash equilibrium by computing a best response to $\boldsymbol{y}'$. (These are both convex optimization problems, solvable using standard techniques.) If so, output $(\boldsymbol{x}, \boldsymbol{y}')$. Otherwise, let $\boldsymbol{x}' = \Pi_{\mathcal{X}}(\boldsymbol{x} + \eta \nabla_{\boldsymbol{x}} u_1(\boldsymbol{x}, \boldsymbol{y}'))$ with $\eta = 1/(2L)$ as before. By Lemma 9.44 applied to the VI problem $(\mathcal{X}, -\nabla_{\boldsymbol{x}} u_1(\cdot, \boldsymbol{y}'))$, we have

$$\langle \nabla_{\boldsymbol{x}} u_1(\boldsymbol{x}', \boldsymbol{y}'), \boldsymbol{x}' - \boldsymbol{x} \rangle \geq \gamma$$

But then, setting $\boldsymbol{a}' = (\lambda_1', \lambda_2', \lambda_1' \boldsymbol{x}', \lambda_2' \boldsymbol{y}')$ (for any $\lambda_1', \lambda_2' > 0$), we have

$$\langle G(\boldsymbol{a}'), \boldsymbol{a}' - \boldsymbol{a} \rangle = \lambda_1 \langle \nabla_{\boldsymbol{x}} u_1(\boldsymbol{x}', \boldsymbol{y}'), \boldsymbol{x} - \boldsymbol{x}' \rangle + \lambda_2 \left\langle \nabla_{\boldsymbol{y}} u_1(\boldsymbol{x}', \boldsymbol{y}'), \boldsymbol{y} - \boldsymbol{y}' \right\rangle \leq -\lambda_1 \cdot \gamma + \lambda_2 \cdot 2BR$$

Thus, for $\lambda_2 \leq \min\{1/2, \gamma/(4BR)\}$ we have $\langle G(\boldsymbol{a}'), \boldsymbol{a}' - \boldsymbol{a} \rangle \leq -\gamma/4$.

Case 2. $\min\{\lambda_1, \lambda_2\} > \gamma/(4BR)$. Then let $\eta = 1/(2L')$, where $L' \leq \mathsf{poly}(B, L, R) \cdot 8BR/\gamma$ is the Lipschitz constant of G on $\mathcal{P}_{\gamma/8BR}$. Let $\boldsymbol{a}' = (\lambda_1', \lambda_2', \lambda_1' \boldsymbol{x}', \lambda_2' \boldsymbol{y}') = \Pi_{\mathcal{P}_\alpha}(\boldsymbol{a} - \eta G(\boldsymbol{a}))$. Then, since $\eta \leq \gamma/(8B^2 R)$, we have $\min\{\lambda_1', \lambda_2'\} \geq \gamma/(8BR)$, so $\boldsymbol{a}, \boldsymbol{a}' \in \mathcal{P}_{\gamma/(8BR)}$. Since L' is the Lipschitz constant of G on $\mathcal{P}_{\gamma/(8BR)}$, Lemma 9.44 implies

$$\langle G(\boldsymbol{a}'), \boldsymbol{a} - \boldsymbol{a}' \rangle \geq \frac{\epsilon^2 L'}{(B + 4RL')^2} \geq \frac{\epsilon^4}{\mathsf{poly}(B, L, R)}. \qquad \square$$

[9.11] Since $\alpha > 0$, it is actually impossible for $\lambda_2 = 0$ to arise in our algorithm; nonetheless, studying this case will be instrutive for intuition.

> Taking $\alpha = \gamma'/(4BR)$ and following the remainder of the analysis of Theorem 9.55 yields Theorem 9.57. □

It should be noted that Theorem 9.57 only applies for games with two players. This restriction is fundamental. Indeed, given any two-player game Γ, consider the $(n+1)$-player game Γ' created by adding a player to Γ whose utility is identically zero. Then the CCE gap for player $n+1$ is always zero, so Γ' has no strict CCEs, but a Nash equilibrium of Γ' would immediately also yield a Nash equilibrium of Γ. Thus, solving the ϵ-Nash-or-strict-CCE problem for $n+1$ players is at least as hard as finding n-player Nash equilibria.

9.6 A Deep Dive on Expected Variational Inequalities

We now take a closer look at the problem of expected variational inequalities (Definition 9.29) in their own right, as a relaxation of the (S)VI problem. We will start by defining a generalized version of the EVI problem that we call the Φ-EVI problem. Intuitively, one should think of the Φ-EVI problem as the generalization of Φ-equilibria (from Chapter 8) to variational inequalities.

Definition 9.60. Given a set of *deviations* $\Phi \subseteq \mathcal{X}^{\mathcal{X}}$, the *$\epsilon$-approximate Φ-expected variational inequality (Φ-EVI) problem asks for a distribution $\mu \in \Delta(\mathcal{X})$ such that*

$$\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mu} \langle F(\boldsymbol{x}), \phi(\boldsymbol{x}) - \boldsymbol{x}\rangle \geq -\epsilon \quad \forall \phi \in \Phi.$$

Φ-EVIs are no harder than VIs[9.12] (assuming that solutions exist), regardless of the choice of Φ—indeed, if $\boldsymbol{x}$ is a VI solution, then $\mu = \delta(\boldsymbol{x})$ is a Φ-EVI solution regardless of Φ. However, as we shall see, a primary justification of Φ-EVIs is that they can be easier than VIs.

Definition 9.60 is crucially parameterized by Φ; the larger the set of deviations Φ, the tighter the set of solutions. As will become clear, Definition 9.60 is intimately connected with notions of *correlated equilibrium (CE)* from game theory (*e.g.*, [15]). The more permissive case where Φ comprises only constant functions, $\Phi = \Phi_{\mathsf{CON}} = \{\phi_{\boldsymbol{x}} : \boldsymbol{x} \in \mathcal{X}\}$ where $\phi_{\boldsymbol{x}}(\boldsymbol{x}') = \boldsymbol{x}$ for all $\boldsymbol{x}' \in \mathcal{X}$, is perhaps the most basic relaxation of Definition 9.1; this is equivalent to the EVI problem as defined in Definition 9.29, and we will refer to it as simply the "EVI problem".

Algorithms and complexity for Φ-EVIs. As it turns out, imposing no constraints on Φ results in an impasse: Φ-EVIs are in general tantamount to regular VIs—thereby being PPAD-hard (Corollaries 9.70 and 9.72). On the other hand, unlike general VIs, one of our key contributions is to show that when Φ contains only linear maps, Φ_{LIN}, Φ-EVIs can be solved in time polynomial in the dimension d and $\log(1/\epsilon)$ (Theorem 9.74), establishing the promised computational property that separates EVIs from VIs. This result is once again based on EAH. (Section 8.2.3 gives a self-contained overview of EAH.) In doing so, we extend the scope of that algorithm to a much broader class of problems well beyond the realm of game theory. Notably, Theorem 9.74 applies even when $\mathcal{X}$ is given implicitly through a membership oracle; this extension makes use of the recent technical approach of Daskalakis et al. [81], discussed in more detail in Section 9.6.2.

[9.12] For this section, "VI" means SVI.

Result	Description	Reference
Existence of (ϵ-approx.) solutions	Under Lipschitz cont. for Φ and bounded F	Theorem 9.61
Complexity with nonlinear Φ	PPAD-hardness with linear F and $\epsilon = \Theta(1)$	Corollaries 9.70 and 9.72
Algorithms for linear Φ	• poly$(d, \log(1/\epsilon))$-time via EAH	Theorem 9.74
	• poly$(d, 1/\epsilon)$-time via Φ-regret minimization	Theorem 9.76

Table 9.8: *Our main results concerning Φ-EVIs (Definition 9.60).*

One limitation of Theorem 9.74 is that it relies on the EAH algorithm, which is slow in practice. We address this by also establishing more scalable algorithms that use Φ_{LIN}-regret minimization (from Theorem 8.53) instead of the ellipsoid algorithm.

In addition to their more favorable computational properties, we further show that Φ-EVIs admit (approximate) solutions under more general conditions than their associated VIs—namely, without F being continuous (Theorem 9.61); Section 9.6.1 documents further interesting aspects on existence.

Connection to other solution concepts. As we have alluded to, Φ-EVIs generalize the seminal concept of a *(coarse) correlated equilibrium à la* Aumann [15] and Moulin and Vial [226] in finite games, and more generally Φ-*equilibria* [132, 136, 278] of concave games. What is more surprising is that Φ_{LIN}-EVIs *refine* CEs even in normal-form games; we give illustrative examples, together with an interpretation, in Section 9.6.3. We also note that Φ-EVIs can be used even in games with nonconcave utilities [46, 75] or noncontinuous gradients (as in nonsmooth optimization), as well as in (pseudo-)games with *coupled constraints* (*cf.* [25] and Section 8.1.3 for related work).

Taken together, these properties provide compelling justification for Φ-EVIs as a solution concept *in lieu of* VIs. Table 9.8 gathers our main results.

9.6.1 Existence and Complexity Barriers

Perhaps the most basic question about Φ-EVIs concerns their *totality*—the existence of solutions. If one is willing to tolerate an arbitrarily small imprecision $\epsilon > 0$, we show that solutions exist under very broad conditions.

Theorem 9.61. *Suppose that $F : \mathcal{X} \to \mathbb{R}^d$ is measurable and there exists $L > 0$ such that every $\phi \in \Phi$ is L-Lipschitz continuous. Then, for any $\epsilon > 0$, there exists an ϵ-approximate solution to the Φ-EVI problem.*

Proof. We define a function $\hat{F}_\delta : \mathcal{X} \to \mathbb{R}^d$ as

$$\hat{F}_\delta : \boldsymbol{x} \mapsto \frac{1}{|\mathcal{B}_\delta(\boldsymbol{x}) \cap \mathcal{X}|} \int_{\mathcal{B}_\delta(\boldsymbol{x}) \cap \mathcal{X}} F(\hat{\boldsymbol{x}}) d\nu(\hat{\boldsymbol{x}});$$

this is a rescaled Lebesgue integral, which represents a multivariate local average. Above,

- $\delta > 0$ is a sufficiently small parameter, to be defined shortly;
- $\mathcal{B}_\delta(\boldsymbol{x}) \subseteq \mathbb{R}^d$ is the (closed) Euclidean ball of radius δ centered at $\boldsymbol{x}$; and
- $|\cdot|$ denotes the set's Borel measure.

Given that F is assumed to be bounded, we can define $B \in \mathbb{R}$ such that $\max_{\boldsymbol{x} \in \mathcal{X}} \|F(\boldsymbol{x})\| \leq B$. For the proof below, it will suffice to set $\delta := \epsilon/(L+1)B$.

We first observe that $\hat{F}_\delta$ is continuous.

Lemma 9.62. *$\hat{F}_\delta$ is continuous.*

Proof. We will show that for any $\boldsymbol{x} \in \mathcal{X}$ and $\epsilon' > 0$, we can choose $\delta' = \delta'(\epsilon')$ such that for any $\boldsymbol{x}' \in \mathcal{X}$ with $\|\boldsymbol{x} - \boldsymbol{x}'\| < \delta'$,

$$\|\hat{F}_\delta(\boldsymbol{x}) - \hat{F}_\delta(\boldsymbol{x}')\| \leq \epsilon'.$$

By the triangle inequality, the difference $\|\hat{F}_\delta(\boldsymbol{x}) - \hat{F}_\delta(\boldsymbol{x}')\|$ can be decomposed as the sum of

$$\textcircled{A} := \left| \frac{1}{|\mathcal{B}_\delta(\boldsymbol{x}) \cap \mathcal{X}|} - \frac{1}{|\mathcal{B}_\delta(\boldsymbol{x}') \cap \mathcal{X}|} \right| \int_{\mathcal{B}_\delta(\boldsymbol{x}) \cap \mathcal{X}} \|F(\hat{\boldsymbol{x}})\| d\nu(\hat{\boldsymbol{x}})$$

and

$$\textcircled{B} := \frac{1}{|\mathcal{B}_\delta(\boldsymbol{x}') \cap \mathcal{X}|} \left\| \int_{\mathcal{B}_\delta(\boldsymbol{x}) \cap \mathcal{X}} F(\hat{\boldsymbol{x}}) d\nu(\hat{\boldsymbol{x}}) - \int_{\mathcal{B}_\delta(\boldsymbol{x}') \cap \mathcal{X}} F(\hat{\boldsymbol{x}}) d\nu(\hat{\boldsymbol{x}}) \right\|.$$

Now, $\textcircled{A}$ can be bounded as

$$\textcircled{A} \leq B \left| 1 - \frac{|\mathcal{B}_\delta(\boldsymbol{x}) \cap \mathcal{X}|}{|\mathcal{B}_\delta(\boldsymbol{x}') \cap \mathcal{X}|} \right| \leq \frac{1}{2}\epsilon',$$

where we selected δ' small enough so that

$$\left(1 - \frac{\epsilon'}{B}\right) |\mathcal{B}_\delta(\boldsymbol{x}') \cap \mathcal{X}| \leq |\mathcal{B}_\delta(\boldsymbol{x}) \cap \mathcal{X}| \leq \left(1 + \frac{\epsilon'}{B}\right) |\mathcal{B}_\delta(\boldsymbol{x}') \cap \mathcal{X}|.$$

Moreover, by selecting δ' small enough so that

$$|(\mathcal{B}_\delta(\boldsymbol{x}) \cap \mathcal{X}) \setminus (\mathcal{B}_\delta(\boldsymbol{x}') \cap \mathcal{X})| + |(\mathcal{B}_\delta(\boldsymbol{x}') \cap \mathcal{X}) \setminus (\mathcal{B}_\delta(\boldsymbol{x}) \cap \mathcal{X})| \leq \frac{1}{2B} \epsilon' |\mathcal{B}_\delta(\boldsymbol{x}') \cap \mathcal{X}|, \quad (9.26)$$

we have

$$\left\| \int_{\mathcal{B}_\delta(\boldsymbol{x})\cap\mathcal{X}} F(\hat{\boldsymbol{x}})d\nu(\hat{\boldsymbol{x}}) - \int_{\mathcal{B}_\delta(\boldsymbol{x}')\cap\mathcal{X}} F(\hat{\boldsymbol{x}})d\nu(\hat{\boldsymbol{x}}) \right\|$$
$$\leq \int_{(\mathcal{B}_\delta(\boldsymbol{x})\cap\mathcal{X})\setminus(\mathcal{B}_\delta(\boldsymbol{x}')\cap\mathcal{X})} \|F(\hat{\boldsymbol{x}})\| d\nu(\hat{\boldsymbol{x}}) + \int_{(\mathcal{B}_\delta(\boldsymbol{x}')\cap\mathcal{X})\setminus(\mathcal{B}_\delta(\boldsymbol{x})\cap\mathcal{X})} \|F(\hat{\boldsymbol{x}})\| d\nu(\hat{\boldsymbol{x}})$$
$$\leq B|(\mathcal{B}_\delta(\boldsymbol{x})\cap\mathcal{X})\setminus(\mathcal{B}_\delta(\boldsymbol{x}')\cap\mathcal{X})| + B|(\mathcal{B}_\delta(\boldsymbol{x}')\cap\mathcal{X})\setminus(\mathcal{B}_\delta(\boldsymbol{x})\cap\mathcal{X})| \leq \frac{1}{2}\epsilon'|\mathcal{B}_\delta(\boldsymbol{x}')\cap\mathcal{X}|,$$

where the last inequality uses (9.26). As a result, we have shown that Ⓐ + Ⓑ $\leq \epsilon'$, thereby implying that $\|\hat{F}_\delta(\boldsymbol{x}) - \hat{F}_\delta(\boldsymbol{x}')\| \leq \epsilon'$. This completes the proof. □

Having established that $\hat{F}_\delta$ is continuous, we can now apply Brouwer's fixed point theorem on the map $\boldsymbol{x} \mapsto \Pi_{\mathcal{X}}(\boldsymbol{x} - \hat{F}_\delta(\boldsymbol{x}))$, where we recall that $\Pi_{\mathcal{X}}$ denotes the Euclidean projection onto $\mathcal{X}$. This implies that there is a point $\boldsymbol{x} \in \mathcal{X}$ such that $\boldsymbol{x} = \Pi_{\mathcal{X}}(\boldsymbol{x} - \hat{F}_\delta(\boldsymbol{x}))$. Moreover, such a point satisfies the VI constraint with respect to $\hat{F}_\delta$:

$$\langle \hat{F}_\delta(\boldsymbol{x}), \boldsymbol{x}' - \boldsymbol{x}\rangle \geq 0 \quad \boldsymbol{x}' \in \mathcal{X};$$

for example, see Kinderlehrer and Stampacchia [174, Section 3] for the derivation. Finally, we define $\mu \in \Delta(\mathcal{X})$ to be the uniform distribution over $\mathcal{B}_\delta(\boldsymbol{x}) \cap \mathcal{X}$. Then, for any $\phi \in \Phi$,

$$\begin{aligned}\langle \hat{F}_\delta(\boldsymbol{x}), \phi(\boldsymbol{x}) - \boldsymbol{x}\rangle &= \mathop{\mathbb{E}}_{\hat{\boldsymbol{x}}\sim\mu} \langle F(\hat{\boldsymbol{x}}), \phi(\boldsymbol{x}) - \boldsymbol{x}\rangle \\ &= \mathop{\mathbb{E}}_{\hat{\boldsymbol{x}}\sim\mu} \langle F(\hat{\boldsymbol{x}}), \hat{\boldsymbol{x}} - \boldsymbol{x}\rangle + \mathop{\mathbb{E}}_{\hat{\boldsymbol{x}}\sim\mu} \langle F(\hat{\boldsymbol{x}}), \phi(\boldsymbol{x}) - \phi(\hat{\boldsymbol{x}})\rangle + \mathop{\mathbb{E}}_{\hat{\boldsymbol{x}}\sim\mu} \langle F(\hat{\boldsymbol{x}}), \phi(\hat{\boldsymbol{x}}) - \hat{\boldsymbol{x}}\rangle. \end{aligned} \tag{9.27}$$

The first term in (9.27) can be bounded as

$$\mathop{\mathbb{E}}_{\hat{\boldsymbol{x}}\sim\mu} \langle F(\hat{\boldsymbol{x}}), \hat{\boldsymbol{x}} - \boldsymbol{x}\rangle \leq \sqrt{\mathop{\mathbb{E}}_{\hat{\boldsymbol{x}}\sim\mu} \|F(\hat{\boldsymbol{x}})\|^2} \sqrt{\mathop{\mathbb{E}}_{\hat{\boldsymbol{x}}\sim\mu} \|\hat{\boldsymbol{x}} - \boldsymbol{x}\|^2} \leq \delta B, \tag{9.28}$$

where we used the Cauchy-Schwarz inequality, the fact that $\|F(\hat{\boldsymbol{x}})\| \leq B$ for all $\hat{\boldsymbol{x}} \in \mathcal{X}$, and $\|\hat{\boldsymbol{x}} - \boldsymbol{x}\| \leq \delta$ for all $\hat{\boldsymbol{x}}$ in the support of μ. Similarly, the second term in (9.27) can be bounded as

$$\begin{aligned}\mathop{\mathbb{E}}_{\hat{\boldsymbol{x}}\sim\mu} \langle F(\hat{\boldsymbol{x}}), \phi(\boldsymbol{x}) - \phi(\hat{\boldsymbol{x}})\rangle &\leq \sqrt{\mathop{\mathbb{E}}_{\hat{\boldsymbol{x}}\sim\mu} \|F(\hat{\boldsymbol{x}})\|^2} \sqrt{\mathop{\mathbb{E}}_{\hat{\boldsymbol{x}}\sim\mu} \|\phi(\hat{\boldsymbol{x}}) - \phi(\boldsymbol{x})\|^2} \\ &\leq L\sqrt{\mathop{\mathbb{E}}_{\hat{\boldsymbol{x}}\sim\mu} \|F(\hat{\boldsymbol{x}})\|^2} \sqrt{\mathop{\mathbb{E}}_{\hat{\boldsymbol{x}}\sim\mu} \|\hat{\boldsymbol{x}} - \boldsymbol{x}\|^2} \leq \delta BL, \end{aligned} \tag{9.29}$$

where we additionally used the assumption that ϕ is L-Lipschitz continuous. Combining (9.28) and (9.29) with (9.27), we have

$$\mathop{\mathbb{E}}_{\hat{\boldsymbol{x}}\sim\mu} \langle F(\hat{\boldsymbol{x}}), \phi(\hat{\boldsymbol{x}}) - \hat{\boldsymbol{x}}\rangle \geq -\delta(L+1)B + \langle \hat{F}_\delta(\boldsymbol{x}), \phi(\boldsymbol{x}) - \boldsymbol{x}\rangle \geq -\delta(L+1)B,$$

and this holds for any $\phi \in \Phi$. Setting $\delta := \epsilon/(L+1)B$ completes the proof. □

In particular, our existence proof does not rest on F being continuous. Instead, we consider the continuous function $\hat{F}$ that maps $\boldsymbol{x} \mapsto \mathbb{E}_{\hat{\boldsymbol{x}} \sim \Delta(\mathcal{B}_\delta(\boldsymbol{x}) \cap \mathcal{X})} F(\hat{\boldsymbol{x}})$ (Lemma 9.62), where $\mathcal{B}_\delta(\boldsymbol{x})$ is the Euclidean ball centered at $\boldsymbol{x}$ with radius $\delta = \delta(\epsilon)$. It then suffices to invoke Brouwer's fixed-point theorem for the gradient mapping $\boldsymbol{x} \mapsto \Pi_{\mathcal{X}}(\boldsymbol{x} - \hat{F}(\boldsymbol{x}))$, where $\Pi_{\mathcal{X}}$ is the Euclidean projection with respect to $\mathcal{X}$.

Theorem 9.61 implies that a Φ-EVI can have approximate solutions even when the associated VI problem does not.[9.13]

Proposition 9.63. *There exists a VI problem that does not admit approximate solutions when $\epsilon = \Theta(1)$, but the corresponding ϵ-approximate Φ-EVI is total for any $\epsilon > 0$.*

By contrast, if one insists on exact solutions, EVIs do not necessarily admit solutions.

Proposition 9.64. *When F is not continuous, there exists an EVI problem with no solutions.*

We provide an example that will establish both of those claims.

Example 9.65 (Discontinuous F)**.** Let $F(x)$ be the sign function,

$$F(x) = \text{sgn}(x) := \begin{cases} -1 & \text{if} \quad x < 0, \\ 1 & \text{otherwise}, \end{cases}$$

and $\mathcal{X} = [-1, 1]$. We first claim that there is no ϵ-approximate VI solution for $\epsilon < 1$. Indeed,

- for any $x < 0$, picking $x' = 1$ ensures $\langle F(x), x' - x \rangle = x - 1 < -1$;
- for any $x \geq 0$, picking $x' = -1$ ensures $\langle F(x), x' - x \rangle = -1 - x \leq -1$.

There is also no *exact* EVI solution to this problem. Indeed, consider any $\mu \in \Delta(\mathcal{X})$.

- If $\mathbb{P}_{x \sim \mu}[x = 0] = 1$, then taking $x' = -1$ ensures $\mathbb{E}_{x \sim \mu} \langle F(x), x' - x \rangle = \langle F(0), x' \rangle = -1$.
- Otherwise, taking $x' = 0$, we have

$$\mathop{\mathbb{E}}_{x \sim \mu} \langle F(x), x' - x \rangle = \mathop{\mathbb{E}}_{x \sim \mu} [-|x|] < 0.$$

On the other hand, for any $\epsilon > 0$, there exists an ϵ-approximate EVI solution (as promised by Theorem 9.61). In particular, suppose that μ uniformly picks between $-\epsilon$ and ϵ. Then, for any $x' \in \mathcal{X}$,

$$\mathop{\mathbb{E}}_{x \sim \mu} \langle F(x), x' - x \rangle = -\frac{1}{2}(x' + \epsilon) + \frac{1}{2}(x' - \epsilon) = -\epsilon.$$

It is worth pointing out that the above example can be slightly modified so that exact EVI solutions do exist, as we explain below.

[9.13] Noncontinuity of F manifests itself prominently in *nonsmooth* optimization (*e.g.*, [82, 165, 285, 314]); recent research there focuses on *Goldstein* stationary points [131].

Example 9.66 (Modification of Example 9.65 with exact VI)**.** We define $F(x)$ identically to Example 9.65, except $F(1/2) = -1$. We claim that there is no VI solution for $\epsilon < 1/2$: any $x \neq 1/2$ by Example 9.65, and $x = 1/2$ is not a solution since $y = 1$ ensures $\langle F(x), x' - x\rangle = -1/2$.

However, there is an exact EVI solution: fix any $x^* \in [0, 1/2)$ and consider μ that uniformly mixes between $x = x^*$ and $x = 1/2$. Then, for any $x' \in \mathcal{X}$,

$$\mathop{\mathbb{E}}_{x\sim\mu} \langle F(x), x' - x\rangle = \frac{1}{2}(x' - x^*) - \frac{1}{2}\left(x' - \frac{1}{2}\right) = \frac{1}{2}\left(\frac{1}{2} - x^*\right) > 0.$$

Furthermore, Theorem 9.61 raises the question of whether it is enough to instead assume that every $\phi \in \Phi$ is continuous. Our next result dispels any such hopes.

Theorem 9.67. *There are Φ-EVI instances that do not admit ϵ-approximate solutions even when $\epsilon = \Theta(1)$, F is piecewise constant, and Φ contains only continuous functions.*

Proof. As before, let $F(x)$ be the sign function,

$$F(x) = \operatorname{sgn}(x) := \begin{cases} -1 & \text{if} \quad x < 0, \\ 1 & \text{otherwise,} \end{cases}$$

and $\mu \in \Delta([-1, 1])$ be any distribution. For $\delta > 0$, let $\phi_\delta : [-1, 1] \to [-1, 1]$ be given by

$$\phi_\delta(x) = \begin{cases} 1 & \text{if} \quad x < -2\delta, \\ -(x + \delta)/\delta & \text{if} \quad -2\delta \leq x \leq 0, \\ -1 & \text{if} \quad x > 0. \end{cases}$$

Further, let $\phi_0(x) := -\operatorname{sgn}(x)$. Every ϕ_δ (with $\delta > 0$) is continuous, by construction. Now, note that $\phi_\delta \to \phi_0$ pointwise when $\delta \downarrow 0$, and every ϕ_δ is bounded. As a result, by the dominated convergence theorem, we have

$$\begin{aligned} \lim_{\delta\to 0} \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mu} [F(x)(\phi_\delta(x) - x)] &= \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mu} [F(x)(\phi_0(x) - x)] \\ &= \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mu} [-1 - F(x) \cdot x] \leq -1, \end{aligned}$$

where the last line uses the fact that $F(x)\phi_0(x) = -\operatorname{sgn}(x)^2 = -1$ and $F(x) \cdot x = \operatorname{sgn}(x) \cdot x = |x|$ for all x. Thus, for any $\epsilon < 1$, there must be some $\delta > 0$ for which $\mathbb{E}[F(x)(\phi_\delta(x) - x)] < -\epsilon$, so μ cannot be an ϵ-approximate EVI solution. □

Our final result on existence complements Theorems 9.61 and 9.67 by showing that, when Φ is finite-dimensional, it is enough if every $\phi \in \Phi$ admits a fixed point (this holds, for example, when ϕ is continuous—by Brouwer's theorem).

Theorem 9.68. *Suppose that*

1. *Φ is finite-dimensional, that is, there exists $k \in \mathbb{N}$ and a kernel map $m : \mathcal{X} \to \mathbb{R}^k$ such that every $\phi \in \Phi$ can be expressed as $\mathbf{K}m(\boldsymbol{x})$ for some $\mathbf{K} \in \mathbb{R}^{d\times k}$; and*
2. *every $\phi \in \Phi$ admits a fixed point, that is, a point $\mathcal{X} \ni \boldsymbol{x} = \text{fix}(\phi)$ such that $\phi(\boldsymbol{x}) = \boldsymbol{x}$.*

Then, the Φ-EVI problem admits an ϵ-approximate solution with support size at most $1 + dk$ for every $\epsilon > 0$.

Proof. We assume, without loss of generality, that (as functions) the coordinates $m_i : \mathcal{X} \to \mathbb{R}$ for $1 \le i \le k$ are linearly independent. We further assume that m is bounded, again without loss of generality. (Indeed, if for example m_i is unbounded, then column i of $\mathbf{K}$ must contain all zeros, or else $\phi_{\mathbf{K}}(\boldsymbol{x}) := \mathbf{K}m(\boldsymbol{x})$ would be unbounded; we can thus freely remove such coordinates m_i.)

Now, let $\mathcal{K} := \text{conv}\{\mathbf{K} : \phi_{\mathbf{K}} \in \Phi\}$ be the set of matrices corresponding to maps in Φ; we can assume that $\mathcal{K}$ is closed. We can now rewrite the Φ-EVI problem as

$$\text{find} \quad \mu \in \Delta(\mathcal{X}) \quad \text{s.t.} \quad \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mu} \left\langle F(\boldsymbol{x})m(\boldsymbol{x})^\top, \mathbf{K} - \mathbf{I} \right\rangle \ge 0$$

for all $\mathbf{K} \in \mathcal{K}$, where above $\mathbf{I}$ is the identity matrix and the inner product is the usual Frobenius inner product of matrices.[9.14] Further, let $\mathcal{A} := \text{conv}\{F(\boldsymbol{x})m(\boldsymbol{x})^\top : \boldsymbol{x} \in \mathcal{X}\}$. Then, the Φ-EVI problem can be in turn expressed as

$$\text{find} \quad \mathbf{A} \in \mathcal{A} \quad \text{s.t.} \quad \langle \mathbf{A}, \mathbf{K} - \mathbf{I} \rangle \ge 0$$

for all $\mathbf{K} \in \mathcal{K}$. Since F and m are bounded, by assumption, so is $\mathcal{A}$. Moreover, since the coordinates m_i are linearly independent, $\mathcal{K}$ is also bounded. Thus, letting $\bar{\mathcal{A}}$ denote the closure of $\mathcal{A}$, the max-min problem

$$\max_{\mathbf{A}\in\bar{\mathcal{A}}} \min_{\mathbf{K}\in\mathcal{K}} \langle \mathbf{A}, \mathbf{K} - \mathbf{I} \rangle \tag{9.30}$$

satisfies the conditions of the minimax theorem. Moreover, for any $\mathbf{K} \in \mathcal{K}$, the fixed point $\boldsymbol{x} := \text{fix}(\phi_{\mathbf{K}})$ satisfies

$$\left\langle F(\boldsymbol{x})m(\boldsymbol{x})^\top, \mathbf{K} - \mathbf{I} \right\rangle = \langle F(\boldsymbol{x}), \phi_{\mathbf{K}}(\boldsymbol{x}) - \boldsymbol{x} \rangle = 0,$$

so the zero-sum game (9.30) has a nonnegative value; that is, there exists $\mathbf{A} \in \bar{\mathcal{A}}$ such that $\min_{\mathbf{K}\in\mathcal{K}} \langle \mathbf{A}, \mathbf{K} - \mathbf{I} \rangle \ge 0$. Thus, for every $\epsilon > 0$, there exists $\mathbf{A} \in \mathcal{A}$ such that $\min_{\mathbf{K}\in\mathcal{K}} \langle \mathbf{A}, \mathbf{K} - \mathbf{I} \rangle \ge -\epsilon$. Moreover, by Carathéodory's theorem, $\mathbf{A}$ can be expressed as a convex combination of at most $1 + dk$ matrices of the form $F(\boldsymbol{x})m(\boldsymbol{x})^\top$. This convex combination is thus an ϵ-approximate EVI solution. □

The only reason the above proof breaks when $\epsilon = 0$ is that $\mathcal{A}$ may not be closed. Indeed, this

[9.14] To avoid measurability issues, it is enough to consider here only distributions μ with finite support.

issue is fundamental: there are instances where no exact EVI solutions exist even when ϕ contains only constant functions (Proposition 9.64). Notably, this theorem guarantees the existence of solutions with finite support.

Complexity. Having established some basic existence properties, we now turn to the complexity of Φ-EVIs. Let us define the VI gap function $\text{VIGap}(\boldsymbol{x}) := -\min_{\boldsymbol{x}' \in \mathcal{X}} \langle F(\boldsymbol{x}), \boldsymbol{x}' - \boldsymbol{x} \rangle$, which is nonnegative. If we place no restrictions on Φ, it turns out that Φ-EVIs are tantamount to regular VIs:

Proposition 9.69. *If Φ contains all measurable functions from $\mathcal{X}$ to $\mathcal{X}$, then any solution $\mu \in \Delta(\mathcal{X})$ to the ϵ-approximate Φ-EVI problem satisfies*

$$\underset{\boldsymbol{x} \sim \mu}{\mathbb{E}} \text{VIGap}(\boldsymbol{x}) \leq \epsilon. \tag{9.31}$$

Proof. We can define a measurable map $\phi : \mathcal{X} \to \mathcal{X}$ such that $\phi(\boldsymbol{x})$ is an element selected from $\text{argmin}_{\boldsymbol{x}' \in \mathcal{X}} \langle F(\boldsymbol{x}), \boldsymbol{x}' - \boldsymbol{x} \rangle$ by utilizing the measurable maximum theorem [9, Theorem 18.19]. To satisfy the conditions of this theorem, we need to define—using Aliprantis and Border's notation— the weakly measurable set-valued function $\psi : \mathcal{X} \twoheadrightarrow \mathcal{X}$ as $\psi(\boldsymbol{x}) = \mathcal{X}$ and the (Carathéodory) function $f : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ as $f(\boldsymbol{x}, \boldsymbol{x}') = -\langle F(\boldsymbol{x}), \boldsymbol{x}' - \boldsymbol{x} \rangle$. Due to this map ϕ, a Φ-EVI solution $\mu \in \Delta(\mathcal{X})$ must then, in particular, satisfy

$$\underset{\boldsymbol{x} \sim \mu}{\mathbb{E}} \langle F(\boldsymbol{x}), \phi(\boldsymbol{x}) - \boldsymbol{x} \rangle = \underset{\boldsymbol{x} \sim \mu}{\mathbb{E}} \underset{\boldsymbol{x}' \in \mathcal{X}}{\text{argmin}} \langle F(\boldsymbol{x}), \boldsymbol{x}' - \boldsymbol{x} \rangle \geq 0.$$

Therefore, there must exist $\boldsymbol{x}^* \in \mathcal{X}$ with $\text{argmin}_{\boldsymbol{x}' \in \mathcal{X}} \langle F(\boldsymbol{x}^*), \boldsymbol{x}' - \boldsymbol{x}^* \rangle \geq 0$, that is, a VI solution $\boldsymbol{x}^*$. If μ has finite support, then such a $\boldsymbol{x}^*$ exists within that support. The ϵ-approximation case follows analogously. □

When μ must be given explicitly, Proposition 9.69 immediately implies that Φ-EVIs are computationally hard, because (9.31) implies that $\text{VIGap}(\boldsymbol{x}) \leq \epsilon$ for some $\boldsymbol{x}$ in the support of μ, and such a point can be identified in polynomial time.[9.15]

Corollary 9.70. *The ϵ-approximate Φ-EVI problem is* PPAD*-hard even when ϵ is an absolute constant and F is linear.*

Coupled with Proposition 9.69, this follows from the hardness result of Rubinstein [259] concerning Nash equilibria in (multi-player) polymatrix games (for binary-action, graphical games, [83] recently showed that PPAD-hardness persists up to $\epsilon < 1/2$). Corollary 9.70 notwithstanding, it is easy to see that the set of solutions to Φ-EVIs is convex for any $\Phi \subseteq \mathcal{X}^{\mathcal{X}}$.

Remark 9.71. Let $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$, as in an n-player game. Whether Corollary 9.70 applies

[9.15] This argument carries over without restricting the support of μ, by assuming instead access to a sampling oracle from μ: a standard Chernoff bound implies that the empirical distribution (w.r.t. a large enough sample size) approximately satisfies (9.31).

under deviations that can be decomposed as $\phi : \boldsymbol{x} \mapsto \phi(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}_1), \dots, \phi_n(\boldsymbol{x}_n))$ is equivalent to asking whether computing an NFCE is PPAD-hard—a major open question, as we have established in past chapters.

Viewed differently, a special case of the Φ-EVI problem arises when $\Phi = \{\phi\}$ and $F(\boldsymbol{x}) = \boldsymbol{x} - \phi(\boldsymbol{x})$, for some fixed map $\phi : \mathcal{X} \to \mathcal{X}$. In this case, the Φ-EVI problem reduces to finding a $\mu \in \Delta(\mathcal{X})$ such that

$$\mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mu} \langle F(\boldsymbol{x}), \phi(\boldsymbol{x}) - \boldsymbol{x} \rangle = - \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mu} \|\phi(\boldsymbol{x}) - \boldsymbol{x}\|^2 \geq -\epsilon. \tag{9.32}$$

As a result, μ must contain in its support an ϵ-approximate fixed point of ϕ, a problem which is PPAD-hard already for quadratic functions [317].

Corollary 9.72. *The ϵ-approximate Φ-EVI problem is* PPAD*-hard even when ϵ is an absolute constant, F is quadratic, and $\Phi = \{\phi\}$ for a quadratic map $\phi : \mathcal{X} \to \mathcal{X}$.*

It is also worth noting that, unlike Corollary 9.70, Φ in the corollary above contains only continuous functions.

It also follows from (9.32) that, for $\epsilon = 0$, Φ-EVIs capture exact fixed points. The complexity class FIXP characterizes such problems [92].

Corollary 9.73. *The Φ-EVI problem is* FIXP*-hard, assuming that* $\operatorname{supp}(\mu) \leq \operatorname{poly}(d)$.

Exponential lower bounds in terms of the number of function evaluations of F also follow from Hirsch et al. [148].

On a positive note, the next section establishes polynomial-time algorithms when Φ contains only *linear* maps.[9.16]

9.6.2 Efficient Computation with Linear Φ

The hardness results of the previous section highlight the need to restrict the set Φ in order to make meaningful progress. Our main result here establishes a polynomial-time algorithm when Φ contains only linear maps.

Theorem 9.74. *If Φ contains only linear maps, the ϵ-approximate Φ-EVI problem can be solved in time* $\operatorname{poly}(d, \log(B/\epsilon))$ *given a membership oracle for $\mathcal{X}$.*

Proof. It suffices to show how to run EAH, that is, it suffices to construct a GERorSEP oracle. But this is precisely the *semi-separation oracle* solved by Daskalakis et al. [81, Lemma 4.1], stated below.

[9.16] We do not distinguish between affine and linear maps because we can always set $\mathcal{X} \leftarrow \mathcal{X} \times \{1\}$, in which case affine and linear maps coincide.

Lemma 9.75 ([81]). *There is an algorithm that takes* $\mathbf{K} \in \mathbb{R}^{d \times d}$*, runs in* $\mathsf{poly}(d)$ *time, makes* $\mathsf{poly}(d)$ *oracle queries to* $\mathcal{X}$*, and either returns a fixed point* $\mathcal{X} \ni \boldsymbol{x} = \mathbf{K}\boldsymbol{x}$*, or a hyperplane separating* $\mathbf{K}$ *from* Φ_{LIN}.

The theorem now follows from an identical argument to the correctness proof of EAH. □

On a separate note, Theorem 9.74 only accounts for approximate solutions. We cannot hope to improve that in the sense that exact solutions might be supported only on irrational points even in concave maximization.

9.6.2.1 Regret minimization for EVIs on polytopes

One caveat of Theorem 9.74 is that it relies on the impractical EAH algorithm. To address this limitation, we will show that Φ-EVIs are also amenable to the more scalable approach of *regret minimization*—albeit with an inferior complexity growing as $\mathsf{poly}(1/\epsilon)$.

Specifically, in our context, the regret minimization framework can be applied as follows. At any time $t \in \mathbb{N}$, we think of a "learner" selecting a point $\boldsymbol{x}^{(t)} \in \mathcal{X}$, whereupon $F(\boldsymbol{x}^{(t)})$ is given as feedback from the "environment," so that the utility at time t reads $-\langle \boldsymbol{x}^{(t)}, F(\boldsymbol{x}^{(t)}) \rangle$. Φ-*regret* is a measure of performance in online learning, defined as

$$\textsc{Reg}_{\Phi}(T) := \max_{\phi \in \Phi} \sum_{t=1}^{T} \langle F(\boldsymbol{x}^{(t)}), \phi(\boldsymbol{x}^{(t)}) - \boldsymbol{x}^{(t)} \rangle.$$

The uniform distribution μ on $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(T)}\}$ is clearly a $\textsc{Reg}_{\Phi}(T)/T$-approximate Φ-EVI solution. Therefore, the following result follows from Daskalakis et al. [81] (or Corollary 8.54, for explicitly-represented $\mathcal{X}$):

Theorem 9.76. *There is a deterministic algorithm that finds an* ϵ*-approximate EVI solution after* $\mathsf{poly}(d, m)/\epsilon^2$ *rounds, and requires solving a convex quadratic program with* $O(d^2 + m^2)$ *variables and constraints in each iteration.*

9.6.3 Game Theory Applications of EVIs

A major motivation for studying Φ-EVIs lies in a strong connection to *(C)CEs* [15] in games. Indeed, it is immmediate from the definitions that Φ-EVIs are enough to capture arbitrary notions of Φ-equilibria for general multilinear games.

9.6.3.1 Coupled Constraints

We observe that (Φ_{LIN}-)EVIs can be used even in "pseudo-games," in which $\mathcal{X}$ does not necessarily decompose into $\mathcal{X}_1 \times \cdots \times \mathcal{X}_n$; this means that $\boldsymbol{x}_i \in \mathcal{X}_i(\boldsymbol{x}_{-i})$. As we discuss in Section 8.1.3, most prior work in such settings has focused on generalized Nash equilibria, with the exception of Bernasconi et al. [25]. (Φ_{LIN}-)EVIs induce an interesting notion of LCEs/CCEs in pseudo-games, albeit not directly comparable to the one put forward by Bernasconi et al. [25]. It is worth noting that Bernasconi et al. [25] left open whether efficient algorithms for computing their notion of (coarse) correlated equilibria exist.

Definition 9.77. Given an n-player pseudo-game with concave, differentiable utilities and joint constraints $\mathcal{X}$, a distribution $\mu \in \Delta(\mathcal{X})$ is an *ϵ-ALCE* if

$$\max_{\phi \in \Phi_{\mathsf{LIN}}} \mathbb{E}_{\boldsymbol{x} \sim \mu} \sum_{i=1}^{n} u_i(\phi(\boldsymbol{x})_i, \boldsymbol{x}_{-i}) - \sum_{i=1}^{n} u_i(\boldsymbol{x}) \leq \epsilon.$$

By virtue of our main result (Theorem 9.74), such an equilibrium can be computed in polynomial time.

9.6.3.2 Noncontinuous Gradients

In fact, our results do not rest on the usual assumption that each player's gradient is a continuous function, thereby significantly expanding the scope of prior known results even in games. For example, we refer to Bichler et al. [27], Dasgupta and Maskin [74], Martin and Sandholm [209] for pointers to some applications.

9.6.3.3 Nonconcave Games

Φ-EVIs give rise to a notion of *local* Φ-equilibrium (Definition 9.78) in nonconcave games. It turns out that this captures a recent result by Cai et al. [46], but our framework has certain important advantages. First, we give a $\mathsf{poly}(d, \log(1/\epsilon))$-time algorithm (Theorem 9.74), while theirs scale polynomially in $1/\epsilon$. Second, our results do not assume continuity of the gradients. And finally, we consider our formulation more natural.

Consider an n-player game in which each player $i \in [n]$ has a convex and compact strategy set $\mathcal{X}_i$, and a differentiable utility function $u_i : \mathcal{X}_1 \times \cdots \times \mathcal{X}_n \to \mathbb{R}$. Crucially, there is now no assumption that u_i is concave. In this setting, our framework suggests the following definition.

Definition 9.78. Given sets of functions $\Phi \subseteq \mathcal{X}_i^{\mathcal{X}_i}$, an *$\epsilon$-approximate local $(\Phi_i)_{i=1}^n$-equilibrium* in an n-player nonconcave game is a distribution $\mu \in \Delta(\mathcal{X}_1 \times \cdots \times \mathcal{X}_n)$ such that for any player $i \in [n]$ and deviation $\phi_i \in \Phi_i$,

$$\mathbb{E}_{\boldsymbol{x} \sim \mu} \left\langle \nabla_{\boldsymbol{x}_i} u_i(\boldsymbol{x}), \phi_i(\boldsymbol{x}_i) - \boldsymbol{x} \right\rangle \leq \epsilon.$$

Theorem 9.74 immediately implies the following result when $\Phi_i = \Phi_{\mathsf{LIN}}(\mathcal{X}_i, \mathcal{X}_i)$; as before, in what follows, we assume a membership oracle for each $\mathcal{X}_i$.

Corollary 9.79. *Suppose $\|\nabla u_i(\boldsymbol{x})\| \leq B$ for every player $i \in [n]$ and profile $\boldsymbol{x} \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$. Then, there is a $\mathsf{poly}(d, \log(B/\epsilon))$-time algorithm that outputs an ϵ-approximate local $(\Phi_i)_{i=1}^n$-equilibrium.*

Similarly, the existence of linear swap-regret minimizers for arbitrary polytopes $\mathcal{X}_i$ [81] immediately implies the following.

Corollary 9.80. *There is an independent no-regret learning algorithm that computes ϵ-approximate local $(\Phi_i)_{i=1}^n$-equilibria in $\mathsf{poly}(d, 1/\epsilon)$ rounds and $\mathsf{poly}(d, 1/\epsilon)$ per-round runtime.*

Cai et al. [46] also studied the problem of computing local $(\Phi_i)_{i=1}^n$-equilibria in nonconcave games. They defined ϵ-local $(\Phi_i)_{i=1}^n$-equilibria instead by restricting the magnitudes of the deviations to the "first-order" regime where local deviations cannot change the gradients by too much. In particular, they assume that utility functions u_i are smooth, in the sense that

$$\left\|\nabla_{\boldsymbol{x}_i} u_i(\boldsymbol{x}_i, \boldsymbol{x}_{-i}) - \nabla_{\boldsymbol{x}_i} u_i(\boldsymbol{x}_i', \boldsymbol{x}_{-i})\right\|_2 \le L\left\|\boldsymbol{x}_i - \boldsymbol{x}_i'\right\| \quad \forall \boldsymbol{x}_i, \boldsymbol{x}_i' \in \mathcal{X}_i, \forall \boldsymbol{x}_{-i} \in \bigtimes_{i' \neq i} \mathcal{X}_{i'},$$

where $L > 0$ is a constant. Then, they restrict deviations to only slightly perturb the strategies, that is, for a given set $\Phi_i \subseteq \mathcal{X}_i^{\mathcal{X}_i}$, they define a set

$$\Phi_i(\delta) := \{\lambda \phi_i + (1 - \lambda)\,\mathrm{Id} : \phi_i \in \Phi_i, \lambda \le \delta / D_i\},$$

where $\mathrm{Id} : \mathcal{X} \to \mathcal{X}$ is the identity function and D_i is the ℓ_2-diameter of $\mathcal{X}_i$, *i.e.*, $\|\boldsymbol{x} - \boldsymbol{x}'\|_2 \le D_i$ for all $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}_i$. With this restriction, they show [46, Lemma 1 and Theorem 10] that Φ-regret minimizers converge to $\Phi(\delta)$-equilibria, in the sense that

$$\mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mu} [u_i(\phi_i(\boldsymbol{x}_i), \boldsymbol{x}_{-i}) - u_i(\boldsymbol{x})] \le \frac{\delta}{D_i} \frac{\textsc{Reg}_{\Phi,i}(T)}{T} + \frac{\delta^2 L}{2},$$

where $\textsc{Reg}_{\Phi,i}$ is the Φ_i-regret of Player $i \in [n]$, for all players i and deviations $\phi_i \in \Phi_i(\delta)$. Our results imply theirs, in the following sense.

Proposition 9.81. *Any ϵ-approximate local $(\Phi_i)_{i=1}^n$-equilibrium μ (per Definition 9.78) satisfies*

$$\mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mu} [u_i(\phi_i(\boldsymbol{x}_i), \boldsymbol{x}_{-i}) - u_i(\boldsymbol{x})] \le \frac{\delta \epsilon}{D_i} + \frac{\delta^2 L}{2}$$

for any player $i \in [n]$ and deviation $\phi_i \in \Phi_i(\delta)$.

Proof. Write $\phi_i = \lambda \phi_i^* + (1 - \lambda)\,\mathrm{Id}$ for some $\phi_i^* \in \Phi_i$. Then,

$$\begin{aligned} u_i(\phi_i(\boldsymbol{x}_i), \boldsymbol{x}_{-i}) - u_i(\boldsymbol{x}) &\le \left\langle \nabla_{\boldsymbol{x}_i} u_i(\boldsymbol{x}), \phi_i(\boldsymbol{x}_i) - \boldsymbol{x}_i \right\rangle + \frac{L}{2}\|\phi_i(\boldsymbol{x}_i) - \boldsymbol{x}_i\|_2^2 \\ &\le \frac{\delta}{D_i}\left\langle \nabla_{\boldsymbol{x}_i} u_i(\boldsymbol{x}), \phi_i^*(\boldsymbol{x}_i) - \boldsymbol{x}_i \right\rangle + \frac{\delta^2 L}{2}, \end{aligned}$$

where the last inequality uses the fact that $\lambda \le \delta / D_i$ and therefore $\|\phi_i(\boldsymbol{x}_i) - \boldsymbol{x}_i\|_2 \le \lambda \left\|\phi_i^*(\boldsymbol{x}_i) - \boldsymbol{x}_i\right\|_2 \le \lambda D_i \le \delta$. Taking expectations over μ and applying the definition of ϵ-approximate local $(\Phi_i)_{i=1}^n$-equilibrium completes the proof. □

However, our results improve on theirs in several ways:

- We believe that the formulation of local $(\Phi_i)_{i=1}^n$-equilibria using gradients directly instead of restricting to small perturbations is more natural and more directly conveys what it means for a distribution to be a local $(\Phi_i)_{i=1}^n$-equilibrium without introducing too many hyperparameters.
- Our results do not require the smoothness of the utility functions u_i.
- We have an ellipsoid-based algorithm that computes local $(\Phi_i)_{i=1}^n$-equilibria with convergence rate depending on $\log(1/\epsilon)$, whereas no-regret algorithms only achieve $\mathsf{poly}(1/\epsilon)$ convergence rate.
- Although we do not explicitly state it here, Definition 9.78 and Corollary 9.79 extend directly to the case where $\Phi_i = \Phi_{\mathsf{LIN}}(\mathcal{X}, \mathcal{X}_i)$ (instead of $\Phi_{\mathsf{LIN}}(\mathcal{X}_i, \mathcal{X}_i)$). Per Section 9.6.3.4, this can yield an even smaller set of equilibria.

9.6.3.4 Anonymous LCEs Refining Correlated Equilibria

Perhaps surprisingly, Φ_{LIN}-EVI solutions can be a strict subset of LCEs. This separation can already be appreciated in the setting of normal-form games, and manifests itself in at least two distinct ways. First, there exist games for which a CE need not be a solution to the Φ_{LIN}-EVI. In this sense, Φ_{LIN}-EVIs yield a computationally tractable superset of Nash equilibria that is tighter than CEs. Second, computation suggests that the set of solutions of the Φ_{LIN}-EVI for the game need not be a polyhedron, unlike the set of CEs. We provide a graphical depiction of this phenomenon in Figure 9.9. The figure depicts the set of Φ_{LIN}-EVI solutions to a simple "Bach or Stravinsky" game, in which the players receive payoffs $(3, 2)$ if they both pick Bach, $(2, 3)$ if they both pick Stravinsky, and $(0, 0)$ otherwise.

Interpretation. The reason for this separation is that, for a map $\phi : \mathcal{X} \to \mathcal{X}$, each player's mapped strategy $\phi(\boldsymbol{x})_i$ can also depend (linearly) on *other players' strategies* $\boldsymbol{x}_{-i}$. Indeed, the EVI formulation of a game does not take into account the identities of the players. For this reason, we will call the set of Φ_{LIN}-EVI solutions in a concave game *anonymous linear correlated equilibria*, or *ALCE* for short. We give two game-theoretic interpretations of ALCEs.

First, the ALCEs of a game Γ are the *symmetric* LCEs of the "symmetrized" game in which the players are randomly shuffled before the game begins. That is, consider the n-player game Γ^{sym} defined as follows. Each player's strategy set is $\mathcal{X}$. For strategy profile $(\boldsymbol{x}^1, \dots, \boldsymbol{x}^n) \in \mathcal{X}^n$, the utility to player i is given by

$$u_i^{\text{sym}}(\boldsymbol{x}^1, \dots, \boldsymbol{x}^n) = \frac{1}{n!} \sum_{\sigma \in \mathfrak{S}_n} u_{\sigma(i)}(\boldsymbol{x}_1^{\sigma^{-1}(1)}, \dots, \boldsymbol{x}_n^{\sigma^{-1}(n)}),$$

where $\mathfrak{S}_n$ is the set of permutations $\sigma : [n] \to [n]$. The following result then follows almost by definition.

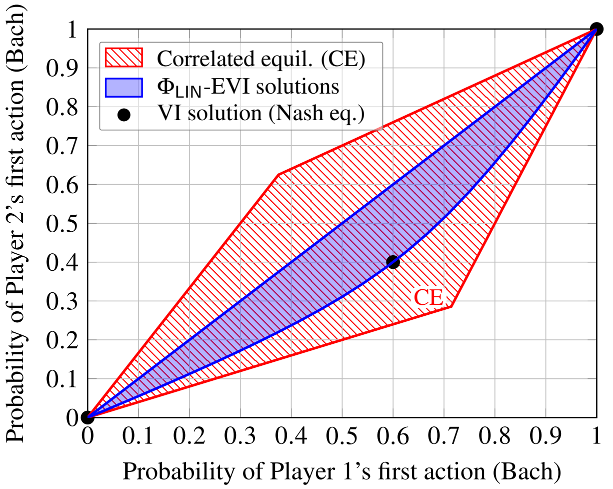


Figure 9.9: *Marginals of the set of correlated equilibria (CE) and of the set of solutions to Φ_{LIN}-EVI in the simple 2×2 game "Bach or Stravinsky." The x- and y-axes show the probability with which the two players select the first action (Bach). The set of marginals of Φ_{LIN}-EVI solutions appears to have a curved boundary corresponding, we believe, to the hyperbola $10x^2 - 25xy + 10y^2 - 6x + 11y = 0$.*

Proposition 9.82. *For a given distribution $\mu \in \Delta(\mathcal{X})$, define the distribution $\mu^n \in \Delta(\mathcal{X}^n)$ by sampling $\boldsymbol{x} \sim \mu$ and outputting $(\boldsymbol{x}, \dots, \boldsymbol{x}) \in \mathcal{X}^n$. Then, μ is a ALCE of Γ if and only if μ^n is an LCE of Γ^{sym}.*

Second, for normal-form games, the ALCEs are the distributions $\mu \in \Delta(\mathcal{X})$ such that no player i has a profitable deviation of the following form. The correlation device first samples $\boldsymbol{x} \sim \mu$, and samples recommendations $a_j \sim \boldsymbol{x}_j$ for each player j. Then, i selects player j (possibly $j = i$) whose recommendation it wishes to see. Player i then observes a sample $a'_j \sim \boldsymbol{x}_j$ that is *independently* sampled from a_j.[9.17] Finally, the player chooses an action $a^*_i \in \mathcal{A}_i$, and each player j gets reward $u_j(a^*_i, a_{-i})$. Thus, players are allowed (modulo the independent sampling) to *spy* on each others' recommendations.

For the special case where Φ_i consists of all linear maps $\mathcal{X} \to \mathcal{X}_i$, we coin the resulting $(\Phi_i)_{i=1}^n$-equilibrium notion an *anonymous linear correlated equilibrium* (ALCE). We now compare ALCEs and LCEs in concave games, and point out some intriguing properties of ALCEs, especially compared to LCEs and CEs.

In normal-form games Γ, LCEs and CEs coincide, and ALCEs lie strictly between LCEs and Nash equilibria, as can be seen in Figure 9.9. We now elaborate on the normal-form specific game-theoretic interpretation of ALCEs by giving an augmented game-based definition. For any fixed $\mu \in \Delta(\mathcal{X})$, consider the augmented game Γ^μ that proceeds as follows.

1. A correlation device samples $\boldsymbol{x} \sim \mu$.

[9.17] This independence is crucial: without it, μ would actually need to be a distribution over pure Nash equilibria!

2. Each player i chooses a player j (possibly not itself) and observes a sample $a_j \sim \boldsymbol{x}_j$, *independently from the samples of other players*. (In particular, if multiple players choose the same player j, then they get independent samples from $\boldsymbol{x}_j$).
3. Each player selects an action $a_i \in \mathcal{A}_i$ and gets utility $u_i(a_1, \dots, a_n)$.

Proposition 9.83. *A distribution $\mu \in \Delta(\mathcal{X})$ is a ALCE of Γ if and only if the strategy profile in which every player requests an action for itself and then plays that action is a Nash equilibrium of Γ^μ.*

The proof will use critically the following characterization of linear maps.

Lemma 9.84 ([117]). *Let $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$ where each $\mathcal{X}_i$ is a simplex $\mathcal{X}_i = \Delta(m_i)$. Then every linear map $\phi : \mathcal{X} \to \mathcal{X}_i$ is a convex combination of linear maps $\phi_j : \mathcal{X} \to \mathcal{X}_i$ that only depend on a single $\boldsymbol{x}_j$.*

Proof of Proposition 9.83. Fix some $\mu \in \Delta(\mathcal{X})$ and suppose that it is not a ALCE, that is, there is some profitable deviation $\phi : \mathcal{X} \to \mathcal{X}_i$ for some player i. By Lemma 9.84, it suffices to assume that ϕ only depends on one player's strategy $\boldsymbol{x}_j$. Moreover, a linear map $\phi : \mathcal{X}_j \to \mathcal{X}_i$ can be represented as $\boldsymbol{x}_j \mapsto \mathbf{A}\boldsymbol{x}_i$, where $\mathbf{A} \in \mathbb{R}^{m_i \times m_j}$ is column-stochastic. Again, it suffices to assume that ϕ is a vertex of the set of column-stochastic matrices, that is, $\mathbf{A}$ has exactly one 1 in each column. Now player i's deviation benefit under deviation ϕ is given by

$$\mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mu} [u_i(\phi_j(\boldsymbol{x}_j), \boldsymbol{x}_{-i}) - u_i(\boldsymbol{x})] = \mathop{\mathbb{E}}_{\substack{\boldsymbol{x} \sim \mu \\ a \sim \boldsymbol{x}}} \left[\mathop{\mathbb{E}}_{a'_j \sim \boldsymbol{x}_j} u_i(\phi_j(a'_j), a_{-i}) - u_i(a) \right],$$

where the equality uses multilinearity of a. This is precisely the deviation benefit of the strategy in Γ^μ for player i in which player i chooses to sample a'_j and then plays an action according to $\phi_j : [m_j] \to [m_i]$. The proposition now follows by observing that these are precisely the possible pure strategy deviations of player i in Γ^μ. □

We make several more observations about the relationship between ALCEs and other notions of equilibrium in games.

- Proposition 9.83 generalizes beyond normal-form games, but needs to be modified. For example, for (single-step) Bayesian games where each $\mathcal{X}_i$ is itself a product of simplices, it follows from a similar proof that, in the augmented game Γ^μ, player i should be allowed to observe its own type first, and then select both another player j and a type θ_j of that player at which to ask for a recommendation. (Another way to see this is that the EVI formulation does not distinguish Bayesian games from their *agent form* [188], where each player-type pair is treated as a separate player.)

  Even more generally, for extensive-form games, we can generalize ALCEs using our characterization of the linear maps $\mathcal{X} \to \mathcal{X}_i$ from Section 8.8: in Γ^μ, player i first may observe its first recommendation at any time of its choosing, and may delay its choice of which player j to observe until that point.

- The above result is a "revelation principle-like" result that characterizes ALCEs. However, unlike the usual revelation principle, it is *not* without loss of generality in this result to take μ to be a distribution over vertices. Formally, in normal-form games, CEs can be without loss of generality defined as distributions over *pure* action profiles $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$ instead of distributions over mixed strategy profiles $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$ [15]. By *without loss of generality* here, we mean the following: given any $\mu \in \Delta(\mathcal{X})$, define $\mu' \in \Delta(\mathcal{A})$ by sampling $\boldsymbol{x} \sim \mu$, then $a_i \sim \boldsymbol{x}_i$ for each i. Then μ is a correlated equilibrium if and only if μ' is.

  This phenomenon is *not* true for ALCEs. Indeed, for two-player games, if $\mu' \in \Delta(\mathcal{A})$ is a ALCE, then in fact μ' is a distribution over *pure* Nash equilibria, which in general may not even exist! It is thus critical in our definition that μ be allowed to be a distribution over *mixed* strategy profiles, not just *pure* strategy profiles.

- We have shown that there is an efficient algorithm for computing *one* (approximate) ALCE. We leave as an open question the complexity of computing an *optimal* (*e.g.*, welfare-maximizing) ALCE (when the number of players n is a constant). Optimal CEs can be computed efficiently in this setting, because the set of CEs $\mu \in \Delta(\mathcal{A})$ is bounded by a small number of linear constraints; however, this fails for ALCEs because, as above, we need to optimize over $\mu \in \Delta(\mathcal{X})$.

9.7 Lower Bounds

We now turn to proving certain hardness results concerning MVIs. In Section 9.7.1, we will show that deciding whether the Minty condition holds is coNP-complete, with coNP-hardness holding even for two-player concave games (Theorem 9.87) or succinct multi-player (normal-form) games (Theorem 9.89) when a constant error ϵ is allowed. Our results complement Proposition 9.33, which showed that in *explicitly* represented games there is a polynomial-time algorithm for that problem. In particular, Theorems 9.87 and 9.89 imply that determining the EVI that minimizes the equilibrium gap—that is, the strictest ACCE—is intractable. From the point of view of the duality exposed in Proposition 9.30, this hardness result is perhaps surprising: as we have seen, an EVI can always be computed in polynomial time, but the dual program turns out to be hard. Section 9.7.1.3 provides a simpler proof that deciding the Minty condition is hard—albeit not applicable to games—based on the hardness of deciding membership in the copositive cone. Finally, in Section 9.7.2, we formalize a straightforward query hardness result for solving MVIs under the promise that the Minty condition holds.

9.7.1 Existence of MVI Solutions

We first characterize the complexity of deciding whether a VI problem satisfies the Minty condition (Assumption 9.3). We will begin by showing membership in coNP, then show hardness results even in the special case of games.

Definition 9.85. The *ϵ-approximate Minty decision problem* is the following. Given a VI problem $\mathrm{VI}(\mathcal{X}, F)$ where $\mathcal{X}$ is assumed to be bounded and in isotropic position and F satisfies Assumption 9.20, and a precision parameter $\epsilon > 0$, decide whether (+) $\mathrm{VI}(\mathcal{X}, F)$ satisfies the Minty condition, or (−) an ϵ-strict EVI solution exists.

Proposition 9.86. *The ϵ-approximate Minty decision problem is in* coNP.

Proof. By Carathéodory's theorem on convex hulls, an ϵ-approximate EVI solution can always have support $\mathsf{poly}(d)$. Further, an approximate EVI solution can be checked in polynomial time with a single call to a linear optimization oracle over $\mathcal{X}$. □

Having established coNP-membership, we now turn to hardness.

9.7.1.1 Hardness for Two-Player Concave Games

To begin with, we show that deciding this problem is hard even for two-player games if the strategy sets $\mathcal{X}, \mathcal{Y}$ are allowed to be arbitrary polytopes.

Theorem 9.87. *Consider a problem $\mathrm{VI}(\mathcal{X} \times \mathcal{Y}, F)$ associated with a two-player concave game per (9.12) such that F satisfies Assumption 9.20, where $\mathcal{X} \subseteq \mathbb{R}^{d_1}$ and $\mathcal{Y} \subseteq \mathbb{R}^{d_2}$ are the strategy sets of the two players. Then the ϵ-approximate Minty decision problem is* coNP*-hard, even when $\mathcal{X}, \mathcal{Y}$ are polytopes given by explicit linear constraints, the utility functions $u_1, u_2 : \mathcal{X} \times \mathcal{Y} \to [-1, 1]$ are bilinear, ϵ is an absolute constant, and we are promised that if there is a Minty point then $(\mathbf{0}, \mathbf{0}) \in \mathcal{X} \times \mathcal{Y}$ is Minty.*

We reduce from the following problem.

Lemma 9.88 (Bilinear optimization is hard)**.** *There exists an absolute constant $\epsilon > 0$ for which the following problem is* NP*-complete: given a bilinear map $f : [0, 1]^{d_1} \times [0, 1]^{d_2} \to [-1, 1]$ and target value $v \in [-1, 1]$, decide whether (+) there exists $(\boldsymbol{x}, \boldsymbol{y}) \in [0, 1]^{d_1} \times [0, 1]^{d_2}$ such that $f(\boldsymbol{x}, \boldsymbol{y}) \geq v + \epsilon$, or (−) for all $(\boldsymbol{x}, \boldsymbol{y}) \in [0, 1]^{d_1} \times [0, 1]^{d_2}$, we have $f(\boldsymbol{x}, \boldsymbol{y}) \leq v$.*

This result is easy to show via reduction from MAX-2-SAT; a proof can be found in the appendix of the full paper [13].

Proof of Theorem 9.87. We will reduce from bilinear optimization. Given an instance (f, v), construct the following two-player game. $\mathcal{X} = \{(\boldsymbol{x}, s) \in [0, 1]^{d_1} \times [0, 1] : \mathbf{0} \leq \boldsymbol{x} \leq \mathbf{1}s\}$, $\mathcal{Y}$ is defined similarly, and the utility functions are given by

$$u_2((\boldsymbol{x}, s), (\boldsymbol{y}, t)) = 0, \quad \text{and} \quad u_1((\boldsymbol{x}, s), (\boldsymbol{y}, t)) = f(\boldsymbol{x}, \boldsymbol{y}) + (1 - s)v - 2(1 - t)s.$$

Notationally, we will use $\tilde{\boldsymbol{x}} = (\boldsymbol{x}, s)$ and $\tilde{\boldsymbol{y}} = (\boldsymbol{y}, t)$. This corresponds to taking the normal-form game in which each player's strategy set is $\{0, 1\}^{d_i}$ and P1's utility function is f, and adding to it one strategy $\tilde{\mathbf{0}}$ for each player such that $u_1(\tilde{\mathbf{0}}, \tilde{\boldsymbol{y}}) = v$ for all $\tilde{\boldsymbol{y}}$, and $u_1(\tilde{\boldsymbol{x}}, \tilde{\mathbf{0}}) = -2$ for all pure strategies (vertices of $\mathcal{X}$) $\tilde{\boldsymbol{x}} \neq \tilde{\mathbf{0}}$. Let F be the operator corresponding to this game. Then deciding if $\mathrm{VI}(\mathcal{X} \times \mathcal{Y}, F)$ satisfies the Minty condition amounts to deciding whether P1 has a dominant strategy.

Suppose first that $f(\boldsymbol{x}, \boldsymbol{y}) \leq v$ for every $(\boldsymbol{x}, \boldsymbol{y})$. Then we see that $\tilde{\mathbf{0}}$ is dominant; indeed, $u_1(\tilde{\mathbf{0}}, \tilde{\boldsymbol{y}}) = v \geq u_1(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}})$ for every $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}})$. Thus, in this case, $\mathrm{VI}(\mathcal{X} \times \mathcal{Y}, F)$ satisfies the Minty condition. Conversely, suppose that there is some $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ such that $f(\boldsymbol{x}^*, \boldsymbol{y}^*) \geq v + \epsilon$. We claim here that P1 has no $\epsilon/4$-approximately dominant (hereafter $\epsilon/4$-dominant) strategy. Suppose for contradiction that $\tilde{\boldsymbol{x}} := (\boldsymbol{x}, s)$ were $\epsilon/4$-dominant. Then in particular we have

$$v(1-s) - 2s = u_1(\tilde{\boldsymbol{x}}, \tilde{\mathbf{0}}) \geq u_1(\tilde{\mathbf{0}}, \tilde{\mathbf{0}}) - \epsilon/4 = v - \epsilon/4,$$

so $s \leq \epsilon/4$. But then we have

$$u_1(\tilde{\boldsymbol{x}}, (\boldsymbol{y}^*, 1)) \leq s + (1-s)v \leq v + \epsilon/2 < u_1((\boldsymbol{x}^*, 1), (\boldsymbol{y}^*, 1)) - \epsilon/4,$$

so $\tilde{\boldsymbol{x}}$ is not $\epsilon/4$-dominant. Thus, in this case, $\mathrm{VI}(\mathcal{X} \times \mathcal{Y}, F)$ admits an $\epsilon/4$-strict EVI solution. □

9.7.1.2 Hardness for Multi-Player Normal-Form Games

Next, we show that, even if the games are normal form, that is, each player's strategy set $\mathcal{X}_i$ is a simplex $\Delta(\mathcal{A}_i)$, deciding the Minty condition becomes hard when the number of players is large.

Theorem 9.89. *Consider a problem* $\mathrm{VI}(\mathcal{X}, F)$ *associated with a multi-player normal-form game per* (9.12) *such that* F *satisfies* *Assumption 9.20. Then the* ϵ*-approximate Minty decision problem is* coNP*-hard, even when each player has two actions,* ϵ *is an absolute constant, and "everyone plays their first action" is a Minty solution if a Minty solution exists.*

Proof. We again reduce from bilinear optimization (Lemma 9.88). Given a bilinear map $f : [0,1]^{d_1} \times [0,1]^{d_2} \to [-1,+1]$, create a game as follows. There are $n := d_1 + d_2 + 2$ players, and two actions per player. We will use $\boldsymbol{x}_1(i) \in [0,1]$ to denote the mixed strategy of Player $i \in \{1, \dots, d_1\}$, $\boldsymbol{x}_2(i) \in [0,1]$ to denote the mixed strategy of $i + d_1 \in \{d_1 + 1, \dots, d_1 + d_2\}$, and $s, t \in [0,1]$ to denote the mixed strategy of the remaining two players. We will overload notation and also use s, t to refer to these two final players. The utility of every player except player s is 0. The utility of player s is given by

$$u_s(\boldsymbol{x}_1, \boldsymbol{x}_2, s, t) = f(\boldsymbol{x}_1, \boldsymbol{x}_2) \cdot st + (1-s)v - 2(1-t)s.$$

(This is precisely the utility function given in the previous proof, except reparameterized.) The proof now proceeds similarly to the proof of the previous result. Since only s has nonzero utility, the Minty condition is equivalent to s having a dominant strategy.

Suppose first that $f(\boldsymbol{x}_1, \boldsymbol{x}_2) \leq v$ for every $(\boldsymbol{x}_1, \boldsymbol{x}_2)$. Then we also have $u_s(\boldsymbol{x}_1, \boldsymbol{x}_2, s, t) \leq v = u_s(\boldsymbol{x}_1, \boldsymbol{x}_2, 0, t)$ for all $(\boldsymbol{x}_1, \boldsymbol{x}_2, s, t)$, so $s = 0$ is dominant. Conversely, suppose that there is some $(\boldsymbol{x}_1^*, \boldsymbol{x}_2^*)$ such that $f(\boldsymbol{x}_1^*, \boldsymbol{x}_2^*) \geq v + \epsilon$. Suppose for contradiction that $s \in [0,1]$ is $\epsilon/4$-dominant. Then

$$u_s(\boldsymbol{x}_1^*, \boldsymbol{x}_2^*, s, 0) = v(1-s) - 2s \geq u_1(\boldsymbol{x}_1^*, \boldsymbol{x}_2^*, 0, 0) - \epsilon/4 = v - \epsilon/4,$$

so $s \le \epsilon/4$. But then, once again, we have

$$u_s(\boldsymbol{x}_1^*, \boldsymbol{x}_2^*, s, 1) \le s + (1-s)v \le v + \epsilon/2 < u_1(\boldsymbol{x}_1^*, \boldsymbol{x}_2^*, 1, 1) - \epsilon/4,$$

so s is not $\epsilon/4$-dominant. Thus, in this case, $\mathrm{VI}(\mathcal{X}_1 \times \mathcal{X}_2, F)$ admits an $\epsilon/4$-strict EVI solution. □

To put this into context, we saw earlier in Proposition 9.33 that in explicitly given normal-form games, there is a polynomial-time algorithm—one whose running time scales polynomially in $\prod_{i=1}^n |\mathcal{A}_i|$—that decides whether the Minty property holds. As such, Theorem 9.89 separates the complexity of deciding the Minty condition under succinct descriptions from that under explicitly-represented games.

9.7.1.3 Hardness for MVI Membership Beyond Games

Beyond VI problems induced by games, treated in Sections 9.7.1.1 and 9.7.1.2, there is an immediate, simpler hardness argument for deciding MVI membership based on the complexity of deciding membership to the *copositive cone* [88, 227]. In particular, a matrix $\mathbf{A} \in \mathbb{R}^{d \times d}$ is said to be copositive if for all nonnegative vectors $\boldsymbol{x} \in \mathbb{R}^d$, we have $\langle \boldsymbol{x}, \mathbf{A}\boldsymbol{x} \rangle \ge 0$; this is a coNP-complete problem. We can then construct the mapping $F : [0,1]^d \ni \boldsymbol{x} \mapsto \boldsymbol{x}(\langle \boldsymbol{x}, \mathbf{A}\boldsymbol{x} \rangle)$. Then, verifying whether $\mathbf{0} \in [0,1]^d$ is an MVI solution is equivalent to checking whether $\mathbf{A}$ is copositive.

9.7.2 Solving Minty VIs

Another natural question is whether one can compute Minty VI solutions—as opposed to SVI solutions, treated in Theorem 9.46—in polynomial time under the promise that the set of MVI solutions is non-empty. In this subsection, we show lower bounds on this problem in both ϵ and d. The upcoming lower bounds are straightforward; it is likely that they have appeared elsewhere, but we include them for completeness since we are not aware of an explicit reference.

9.7.2.1 Lower Bound on ϵ in One Dimension

Our next hardness result gives an exponential lower bound (in $\log(1/\epsilon)$) for that problem, even in a single dimension.

For a rational $\epsilon \ll 1$, we define

$$\phi_\epsilon : [\epsilon, 3\epsilon] \ni x \mapsto (x-\epsilon)^2 (x - 3\epsilon)^2 (x^2(x^2 - 8\epsilon^2) - 1). \tag{9.33}$$

We begin with an elementary calculation that characterizes the derivative of ϕ_ϵ.

Claim 9.90. *For any $x \in (\epsilon, 2\epsilon)$ it holds that $\phi'_\epsilon(x) < 0$, whereas for any $x \in (2\epsilon, 3\epsilon)$ it holds that $\phi'_\epsilon(x) > 0$. In particular, ϕ_ϵ obtains its minimum at $x = 2\epsilon$, with $\phi_\epsilon(2\epsilon) = -\epsilon^4(1 + 16\epsilon^4)$.*

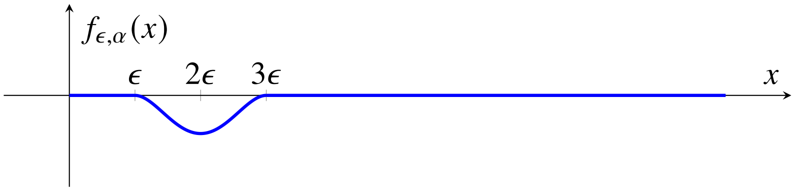


Figure 9.10: *The function $f_{\epsilon,\alpha}(x)$, with $\epsilon = 0.1$ and $\alpha = 0$, over the domain $[0, 1]$; the y-axis is at a larger scale for the sake of the illustration.*

Proof. The derivative $\phi'_\epsilon(x)$ can be expressed as

$$2(x-\epsilon)(x-3\epsilon)^2(x^2(x^2-8\epsilon^2)-1)+2(x-\epsilon)^2(x-3\epsilon)(x^2(x^2-8\epsilon^2)-1)+(x-\epsilon)^2(x-3\epsilon)^2(4x^3-16x\epsilon^2).$$

For $x \in (\epsilon, 3\epsilon)$, we have

$$2(x-\epsilon)(x-3\epsilon)^2(x^2(x^2-8\epsilon^2)-1) < -2(x-\epsilon)^2(x-3\epsilon)(x^2(x^2-8\epsilon^2)-1) \iff (3\epsilon-x) > (x-\epsilon)$$

and

$$(x-\epsilon)^2(x-3\epsilon)^2(4x^3-16x\epsilon^2) \le 0 \iff x \le 2\epsilon.$$

□

Now, let $\alpha \in [-\epsilon, 1-3\epsilon]$. We extend (9.33) as follows.

$$f_{\epsilon,\alpha} : [0,1] \ni x \mapsto \begin{cases} \phi_\epsilon(x-\alpha) & \text{if } \epsilon \le x-\alpha \le 3\epsilon, \\ 0 & \text{otherwise.} \end{cases}$$

An example of $f_{\epsilon,\alpha}$ is illustrated in Figure 9.10. We then define

$$F_{\epsilon,\alpha} : [0,1] \ni x \mapsto \begin{cases} \phi'_\epsilon(x-\alpha) & \text{if } \epsilon < x-\alpha < 3\epsilon, \\ 0 & \text{otherwise.} \end{cases} \tag{9.34}$$

As we show next, the induced VI problem satisfies the Minty condition and is also Lipschitz continuous.

Lemma 9.91. $\mathrm{VI}([0,1], F_{\epsilon,\alpha})$ *satisfies the Minty condition. Furthermore, $F_{\epsilon,\alpha}$ is $O(\epsilon^2)$-Lipschitz continuous.*

Proof. We claim that $x := \alpha + 2\epsilon$ is a solution to the Minty VI problem with respect to $F_{\epsilon,\alpha}$. Indeed, consider any $x' \neq x$. When $x'-\alpha \ge 3\epsilon$ or $x'-\alpha \le \epsilon$, it follows that $F_{\epsilon,\alpha}(x')(x'-x) = 0$. Suppose $x'-\alpha \in (\epsilon, 2\epsilon)$. By Claim 9.90, we have $F_{\epsilon,\alpha}(x') < 0$ while $x'-x < 0$, in turn implying that $F_{\epsilon,\alpha}(x')(x'-x) > 0$. Similarly, when $x'-\alpha \in (2\epsilon, 3\epsilon)$, we have $F_{\epsilon,\alpha}(x') > 0$ while $x'-x > 0$; this shows that $x = \alpha + 2\epsilon$ is indeed a solution to the Minty VI problem.

We continue with the argument that $F_{\epsilon,\alpha}$ is Lipschitz continuous. Let $x, x' \in [0,1]$ such that $x < x'$. We consider the following cases:

- If $x \leq \alpha + \epsilon$ and $x' \geq \alpha + 3\epsilon$, it follows that $F_{\epsilon,\alpha}(x) = F_{\epsilon,\alpha}(x') = 0$.
- If $x \leq \alpha + \epsilon$ and $x' \in (\alpha + \epsilon, \alpha + 3\epsilon)$, it suffices to show that $|\phi'_\epsilon(x' - \alpha)| \leq L|x' - \epsilon|$ since $|x' - \epsilon| \leq |x' - x|$. By the definition of ϕ'_ϵ, this holds with $L = O(\epsilon^2)$.
- If $x \in (\alpha + \epsilon, \alpha + 3\epsilon)$ and $x' \geq \alpha + 3\epsilon$, it suffices to show that $|\phi'_\epsilon(x - \alpha)| \leq L|x - 3\epsilon|$ since $|x - 3\epsilon| \leq |x - x'|$. This again holds with $L = O(\epsilon^2)$.
- Finally, we treat the case where $x, x' \in (\alpha + \epsilon, \alpha + 3\epsilon)$. We can expand $\phi'_\epsilon(x)$ as $24\epsilon^3 - x(144\epsilon^6 + 44\epsilon^2) + x^2(576\epsilon^5 + 24\epsilon) - x^3(668\epsilon^4 + 4) + 200\epsilon^3 x^4 + 84\epsilon^2 x^5 - 56\epsilon x^6 + 8x^7$. From this expression it is easy to see that $|\phi'_\epsilon(x - \alpha) - \phi'_\epsilon(x' - \alpha)| \leq L|x - x'|$ for some $L = O(\epsilon^2)$. □

It is worth noting that identifying an exact *Stampacchia* VI solution to $\mathrm{VI}([0,1], F_{\epsilon,\alpha})$ is trivial: any point outside the region $(\alpha + \epsilon, \alpha + 3\epsilon)$ suffices since $F_{\epsilon,\alpha}$ is defined as 0 in such points. On the other hand, any algorithm that outputs a non-trivial approximation to the global minimum of f needs to output a point in the region $(\alpha + \epsilon, \alpha + 3\epsilon)$. Now, if α is selected initially unbeknownst to the algorithm, it follows that any algorithm that succeeds with constant probability needs to submit $\Omega(1/\epsilon) = \Omega(2^{\log(1/\epsilon)})$ queries to the evaluation oracle for F. We arrive at the following information-theoretic lower bound.

Proposition 9.92. *For any sufficiently small $\epsilon > 0$, any algorithm that computes with constant probability a point $\boldsymbol{x} \in \mathcal{X}$ such that $f(\boldsymbol{x}) \leq \min_{\boldsymbol{x}' \in \mathcal{X}} f(\boldsymbol{x}') + \epsilon^4$ of a function whose associated VI problem satisfies the Minty condition requires $\Omega(1/\epsilon)$ gradient evaluations, even when $\mathcal{X} = [0,1]$, $L = O(\epsilon^2)$ and $B = O(\epsilon^3)$.*

Furthermore, suppose that one is instead looking for an approximate MVI solution $x \in [0,1]$; namely, $F(x')(x' - x) \geq -C\epsilon^4$ for all $x' \in [0,1]$, where C is a sufficiently small constant. We claim that this forces x to belong in $(\alpha + \epsilon, \alpha + 3\epsilon)$, at which point the hardness of Proposition 9.92 kicks in. For the sake of contradiction, suppose first that $x \geq \alpha + 3\epsilon$. Taking then $x' := \alpha + 2.5\epsilon$ leads to a violation since $x' - x < 0$, $F(x') > 0$ (by Claim 9.90), and $|x' - x| = \Theta(\epsilon)$, $|F(x')| = \Theta(\epsilon^3)$. Similar reasoning applies if $x \leq \alpha + \epsilon$ by deviating to $x' := \alpha + 1.5\epsilon$.

Corollary 9.93. *For any sufficiently small $\epsilon > 0$, any algorithm that computes with constant probability a point $\boldsymbol{x} \in \mathcal{X}$ such that $\langle F(\boldsymbol{x}'), \boldsymbol{x}' - \boldsymbol{x} \rangle \geq -C\epsilon^4$, for a sufficiently small constant $C > 0$, of a problem $\mathrm{VI}(\mathcal{X}, F)$ that satisfies the Minty condition requires $\Omega(1/\epsilon)$ gradient evaluations, even when $\mathcal{X} = [0,1]$, $L = O(\epsilon^2)$ and $B = O(\epsilon^3)$.*

9.7.2.2 Lower Bound in High Dimensions

We now address the problem of finding an approximate Minty VI solution in high dimensions, given the promise that such a solution exists. As in the previous subsection, it will suffice to use polynomial optimization problems.

Proposition 9.94. *There exists an absolute constant $\epsilon > 0$ for which the following holds: any algorithm that computes with constant probability an ϵ-approximate global optimum of a function $f : \mathcal{X} \to \mathbb{R}$ whose associated VI problem satisfies the Minty problem requires* $\exp(d)$ *gradient evaluations, even when $\mathcal{X} = [-1, 1]^d$ and L, B are absolute constants.*

Proof. Let $\boldsymbol{c} \in \{-1, +1\}^d$, and $f_{\boldsymbol{c}} : [-1, 1]^d \to \mathbb{R}$ be defined by $f_{\boldsymbol{c}}(\boldsymbol{x}) := -(\max\{1 - \|\boldsymbol{c} - \boldsymbol{x}\|, 0\})^2$. Then $f_{\boldsymbol{c}}$ has minimum $\boldsymbol{x} = \boldsymbol{c}$. Moreover, if we let $F = \nabla f_{\boldsymbol{c}}$, then F satisfies the Minty condition and has constant Lipschitz constant L and bound B. Finally, $F(\boldsymbol{x}) := 0$ if $\operatorname{sgn}(\boldsymbol{x}) \neq \boldsymbol{c}$. Therefore, any algorithm that optimizes this family of functions $f_{\boldsymbol{c}}$ must take $\Omega(2^d)$ gradient evaluations before it successfully learns any information about $f_{\boldsymbol{c}}$. □

9.7.2.3 On Equilibrium Collapse

The class of instances behind Proposition 9.92 also precludes a natural approach for computing SVI solutions under the Minty condition; namely, the positive result of Cai et al. [45] (*cf.* [166, 238]) concerning zero-sum, polymatrix games is based on the observation that there is an "equilibrium collapse," meaning that taking the marginals of any CCE results in a Nash equilibrium; in fact, it is easy to see that this holds more generally under the preconditions of Proposition 9.39. On the other hand, we find that this approach falls short when one merely posits the Minty condition:

Proposition 9.95. *There is a problem* $\mathrm{VI}([0, 1], F)$ *that satisfies the Minty condition, but there exists an exact expected VI solution $\mu \in \Delta([0, 1])$ such that $\mathbb{E}_{\boldsymbol{x} \sim \mu}\, \boldsymbol{x}$ is not a $\Theta(1)$-SVI solution.*

In proof, we take $F := F_{\epsilon,0}$ per (9.34), where $\epsilon = \Theta(1)$. We consider the distribution $\mu \in \Delta([0, 1])$ that samples uniformly between $x_1 := 0.1\epsilon$ and $x_2 := 3.1\epsilon$. By definition, we have $F(x_1) = F(x_2) = 0$. This implies that μ is an exact expected VI (per Definition 9.29). However, it is clear that there exists a constant $C = C(\epsilon)$ such that $\mathbb{E}_{\boldsymbol{x} \sim \mu}\, \boldsymbol{x} = 1.7\epsilon$ is not a C-SVI solution, as claimed in Proposition 9.95.

9.7.3 Ellipsoid Without Extra-Gradient

It is reasonable to ask whether the extra-gradient step in our main algorithm (ExtraGradientEllipsoid) is necessary. In other words, if the extra-gradient step were to be removed, would we still be able to have $\log(1/\epsilon)$ convergence toward an SVI solution under the Minty condition? In this section, we give strong computational evidence that such a guarantee is impossible. We ran the ellipsoid algorithm on polytope $\mathcal{X} = [-1, 1]^2$ and starting ellipse $B_{\sqrt{2}}(\mathbf{0})$, which is the smallest ellipse containing $\mathcal{X}$. At each timestep, given ellipsoid center $\boldsymbol{a}^{(t)} \in \mathcal{X}$, we generated a separating direction $F(\boldsymbol{a}^{(t)})$ such that:

- when t is even, $F(\boldsymbol{a}^{(t)})$ has positive y-component, and $\boldsymbol{a}^{(t+1)}$ has x-component $1/8$, and
- when t is odd, $F(\boldsymbol{a}^{(t)})$ has negative y-component, and $\boldsymbol{a}^{(t+1)}$ has x-component $-1/8$, and

As shown in Figure 9.11, the ellipse quickly grows extremely thin along one axis. After $T = 1293$ iterations,[9.18] we observe the following properties.

- The algorithm has not proven that no Minty solution exists. That is, the red polytope in Figure 9.11 remains nonempty, and in fact the current center $\boldsymbol{a}^{(T)}$ is a candidate Minty solution.
- The radius of the short axis of the ellipse is less than 1.498×10^{-219}.
- No approximate SVI point has been found: $\|F(\boldsymbol{a}^{(t)})\| = 1$ and $\|\boldsymbol{a}^{(t)}\| < 0.481$ for every timestep t, so the minimum SVI gap of any queried point is at least 0.519.
- No Lipschitz violation has been found: for every $0 \le s < t \le T$, we have

$$\frac{\|F(\boldsymbol{a}^{(s)}) - F(\boldsymbol{a}^{(t)})\|}{\|\boldsymbol{a}^{(s)} - \boldsymbol{a}^{(t)}\|} < 7.997.$$

A common idea when using the ellipsoid method for low-dimensional sets is to, once the shortest axis of the ellipse gets small, simply project the problem onto a subspace of smaller dimension and continue. However, this will not work in our setting. Indeed, after such a projection is made, it is possible that $F(\boldsymbol{a}^{(t)})$ has large magnitude in a direction orthogonal to the subspace (so that $\boldsymbol{a}^{(t)}$ is not an SVI point) and yet $F(\boldsymbol{a}^{(t)})$ gives no separation direction, so ellipsoid cannot proceed.

Thus, even in two dimensions, this counterexample demonstrates that ellipsoid fails to find an approximate SVI solution even when the Minty property holds and the given function is Lipschitz. This demonstrates the need for the additional ideas we introduce in Section 9.4.2 (namely, the extra-gradient step) beyond the naive ellipsoid method.

9.8 Conclusion and Future Research

In summary, we established the first algorithm for computing ϵ-SVI solutions under the Minty condition that has polynomial dependence on both d and $\log(1/\epsilon)$. We also provided several new applications of our main results in optimization and game theory, including the first polynomial-time algorithm for quasar-convex optimization. Our results raise a number of interesting questions for future work. First, one of our main contributions was a technique to tackle the lack of full dimensionality when using the ellipsoid algorithm by using strict separation; it is possible that our approach can be employed to other problems as well. Furthermore, our result concerning strict CCEs only applies to two-player games; while it cannot be generalized to games with more than two players or to tighter equilibrium concepts such as correlated equilibrium [15], it would still be interesting to characterize classes of games and solution concepts for which such extensions are possible. Finally, our main result provides further motivation for characterizing what problems satisfy the Minty condition and related concepts.

[9.18] We ran this algorithm with 2048-bit floating-point arithmetic, and numerical precision issues arose at this point.

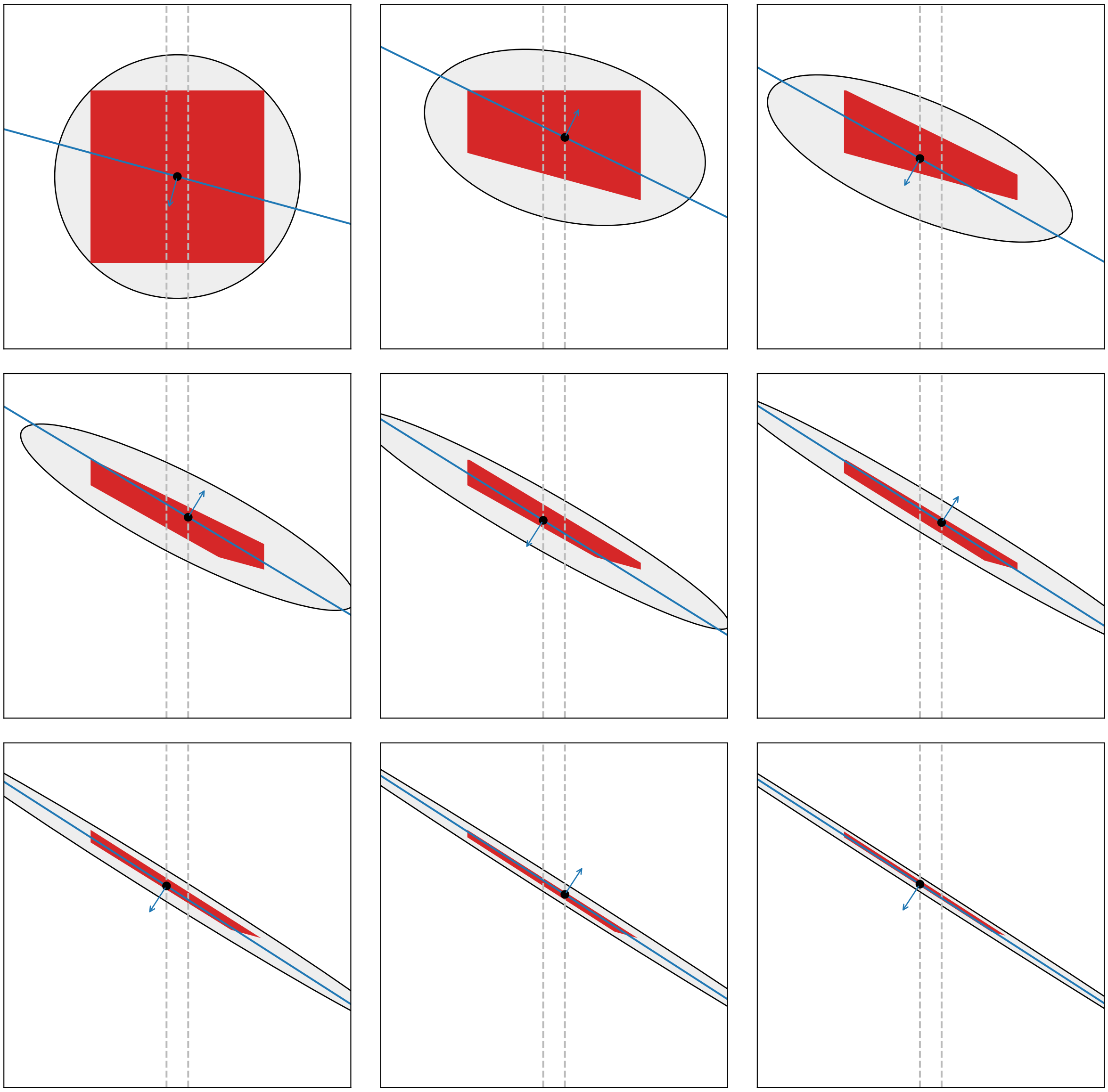


Figure 9.11: *The counterexample for the ellipsoid algorithm without extragradient as described in Section 9.7.3, after t iterations, for* $t = 0, 1, \ldots, 7, 8$. *In each plot, the red polygon is the feasible region where Minty points might still lie, and the blue line/arrow indicates the separating direction* $F(\boldsymbol{a}^{(t)})$. *The two dashed vertical lines correspond to* $x = \pm 1/8$.

Conclusions and Future Research

In this thesis, we have sought to develop new algorithms for equilibrium computation and learning, in extensive-form games and beyond. In doing so, we have uncovered novel relationships, connections, and reductions between problems previously treated separately, both in game theory and beyond, leading to several fundamental new results in both theory and practice. I will conclude with a few more thoughts regarding possible directions of future research and problems that are most exciting to me.

In Part I we developed new theoretical techniques, solution concepts, algorithms, and complexity results for computing *optimal equilibria*, suitably defined, in a variety of game classes. including generalized mechanism design (itself a class of problems that includes classical mechanism design and information design as special cases) and optimal correlated equilibria, under a unified umbrella. In this framework, generalized mechanism design reduces to solving two-player zero-sum games, while optimal correlated equilibrium computation reduces to solving adversarial team games and thus admits a parameterized algorithm. The framework we have developed is extremely general and powerful.

I am excited to see how these techniques can be applied in practice, to real-world mechanism design problems. This comes with many inherent challenges, only some of which I will list here. First, the real world is large scale, and we can expect the scale itself to come with its own challenges and require novel techniques. Second, the real world often demands mechanisms that are *simple* or *interpretable*, so that they can be explained to the humans that will partake as agents in the mechanism (*e.g.*, buyers in an auction). Optimal mechanisms, however, are often not easily interpretable and explainable in this fashion. Is it possible to do automated optimal generalized mechanism design under some constraint of simplicity or explainability? How could one even formalize such a constraint? Finally, the real world is filled with agents that learn over time, such as autobidders in auctions. How do we perform mechanism design with such learning agents in mind? The framework of steering learning agents in Chapter 7 is one possible answer to this question, but certainly not a complete one nor the only possible one.

In Part II, we developed new theoretical techniques, algorithms, and complexity results for learning agents in games and the corresponding questions about computation of various kinds of correlated equilibrium. We also exploited the deep connection between game theory and optimization to use techniques from game theory to arrive at new and state-of-the-art algorithms for computing solutions to certain kinds of variational inequalities, leading to the first algorithm with $\mathsf{polylog}(1/\epsilon)$ dependence in several settings, as well as a novel notion of *variational inequality in expectation*. Finally, we connected our earlier methods for computation of optimal equilibria to the theory of learning in games, resulting in the first algorithms for *steering* learning agents to arbitrary equilibria, including optimal equilibria.

The idea of steering learning agents to equilibria is a new and interesting direction in itself, for which we have only just scratched the surface. Perhaps the most interesting question to ask is whether it is possible for a mediator to steer in extensive-form games without knowledge of the players' private information and/or utility functions. This would be critical to practical applications where mediators are usually not all-knowing.

There is a deep overlap between game theory and optimization. I believe that this interface is a fruitful area for future research, especially in the direction of applying techniques originally discovered in game theory to problems in optimization. The results in this thesis on variational inequalities, which, among other ideas, rely on the ellipsoid against hope algorithm from game theory, are a step in this broad direction. Variational inequalities enjoy a wide range of applications in game theory and beyond. It would be interesting to explore in more detail how expected VIs and the Minty property apply to this wide range of applications. For example, what other classes of games (or other applications beyond games) satisfy the Minty property and therefore are efficiently solvable via the techniques in Chapter 9? In what other applications are expected VIs a reasonable solution concept, in the same way that correlated equilibria are a reasonable solution concept for games?

The most natural open question regarding the computation of Φ-equilibria is whether it is possible to compute (or learn) an optimal *normal-form correlated equilibrium* in $\mathsf{poly}(d, 1/\epsilon)$ time or even $\mathsf{poly}(d, \log(1/\epsilon))$ time in concave games of dimension d. As we have stated, the best known results along these lines are the $d^{\tilde{O}(1/\epsilon)}$-time algorithm of Peng and Rubinstein [242] and Dagan et al. [71], and the matching lower bound from Section 8.10 for no-regret algorithms against worst-case adversaries. I would conjecture that no $\mathsf{poly}(d, 1/\epsilon)$-time algorithm exists, learning or otherwise, exists for computing NFCE; however, proving hardness would likely require new techniques. For example, the PPAD-hardness proofs for problems such as Nash equilibrium [77] rely fundamentally on the fact that the set of Nash equilibria is disconnected in general, but this is simply not true of correlated equilibria.

More broadly, our current theoretical understanding of learning dynamics in games is still very limited in scale. For example, in large collaborative games like *Hanabi*, independent learning dynamics empirically converge to strong joint strategies [299]. Similarly, in complex games such as auctions, independent learning dynamics empirically find Nash equilibria [86]. Notably, both phenomena manifest only at large scale (there are small-scale counterexamples in which independent learning fails to find optimal equilibria in team games and fails to find Nash equilibria in general-sum multiplayer games), and current theory cannot fully explain these phenomena. Namely, under what circumstances do "simple" learning dynamics suffice for strong performance at scale? This line of thinking carries similarity to the theory of supervised learning, which suggests that increasing the size of the network itself ("overparameterization") is key to the performance of supervised learning models [309]. However, the corresponding theory for learning in games is yet undeveloped.

Finally, while I have not discussed them yet in the present thesis, large language models (LLMs) have revolutionized AI as a field in the past few years. Computational game theory can help us understand and improve the study of LLMs in several ways. First, an understanding of game theory and learning dynamics can help *train* better language models. For example, multi-agent dynamics in games ("self-play") can be used to break down problems into multiple pieces, or to train models to be more adversarially robust. Second, as LLMs become more and more common in our lives, they will more frequetly interact *with each other*. Many of the questions that I have alluded to above for large-scale learning dynamics also apply to language models. For example, what learning dynamics occur when LLMs interact with each other? How does one do mechanism design when language models are involved? How can we formally study games, including but not limited to certain recreational hidden-role games, that inherently involve natural-language communication? These are all open-ended questions that are ripe for future study.

Bibliography

[1] Fog of War Chess - Leaderboards - Play Chess Variants Online - chess.com. https://www.chess.com/variants/fog-of-war/leaderboards, 2024. [Online; accessed 16-August-2024]. 43

[2] Chess Statistics. https://www.chessgames.com/chessstats.html, 2024. [Online; accessed 16-Sep-2024]. 55

[3] Fog of War Chess - Leaderboards - Play Chess Variants Online - chess.com. https://www.chess.com/variants/fog-of-war/leaderboards, 2025. [Online; accessed 13-April-2025]. 27

[4] What is Fog Of War chess? — Chess.com Help Center. https://support.chess.com/en/articles/8708650-what-is-fog-of-war-chess, 2025. [Online; accessed 13-April-2025]. 52

[5] Joseph M. Abdou, Nikolaos Pnevmatikos, Marco Scarsini, and Xavier Venel. Decomposition of games: Some strategic considerations. *Mathematics of Operations Research*, 47(1): 176–208, 2022. 291

[6] Ittai Abraham, Danny Dolev, Rica Gonen, and Joe Halpern. Distributed computing meets game theory: robust mechanisms for rational secret sharing and multiparty computation. *ACM Symposium on Principles of Distributed Computing*, 2006. 110

[7] Ilan Adler. The equivalence of linear programs and zero-sum games. *Int. J. Game Theory*, 42(1):165–177, 2013. 177

[8] Matthew Aitchison, Lyndon Benke, and Penny Sweetser. Learning to deceive in multi-agent hidden role games. *Deceptive AI*, 2021. 105

[9] Charalambos D. Aliprantis and Kim C. Border. *Infinite Dimensional Analysis: a Hitchhiker's Guide*. Springer, 2006. 314

[10] Ioannis Anagnostides, Constantinos Daskalakis, Gabriele Farina, Maxwell Fishelson, Noah Golowich, and Tuomas Sandholm. Near-optimal no-regret learning for correlated equilibria in multi-player general-sum games. *Symposium on Theory of Computing (STOC)*, 2021. 180

[11] Ioannis Anagnostides, Gabriele Farina, Ioannis Panageas, and Tuomas Sandholm. Optimistic mirror descent either converges to nash or to strong coarse correlated equilibria in bimatrix games. *Neural Information Processing Systems (NeurIPS)*, 2022. 282

[12] Ioannis Anagnostides, Ioannis Panageas, Gabriele Farina, and Tuomas Sandholm. On last-iterate convergence beyond zero-sum games. *International Conference on Machine Learning (ICML)*, 2022. 279, 291

[13] Ioannis Anagnostides, Gabriele Farina, Tuomas Sandholm, and Brian Hu Zhang. A polynomial-time algorithm for variational inequalities under the Minty condition. *arXiv:2504.03432*, 2025. 13, 276, 285, 323

[14] Angelos Assos, Yuval Dagan, and Constantinos Daskalakis. Maximizing utility in multi-agent environments by anticipating the behavior of other learners. *Neural Information Processing Systems (NeurIPS)*, 2024. 226

[15] Robert Aumann. Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, 1:67–96, 1974. 6, 138, 152, 154, 220, 229, 258, 290, 307, 308, 316, 322, 329

[16] Yakov Babichenko, Christos H. Papadimitriou, and Aviad Rubinstein. Can almost everybody be almost happy? *Innovations in Theoretical Computer Science (ITCS)*, 2016. 264

[17] Yu Bai, Chi Jin, Song Mei, Ziang Song, and Tiancheng Yu. Efficient phi-regret minimization in extensive-form games via online mirror descent. *Neural Information Processing Systems (NeurIPS)*, 2022. 219, 220

[18] Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, et al. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624): 1067–1074, 2022. 2

[19] Nicola Basilico, Andrea Celli, Giuseppe De Nittis, and Nicola Gatti. Team-maxmin equilibrium: efficiency bounds and algorithms. *AAAI Conference on Artificial Intelligence (AAAI)*, 2017. 4, 131, 134

[20] Heinz H. Bauschke, Walaa M. Moursi, and Xianfu Wang. Generalized monotone operators and their averaged resolvents. *Mathematical Programming*, 189(1):55–74, 2021. 300

[21] Donald Beaver. Multiparty protocols tolerating half faulty processors. *International Cryptology Conference (CRYPTO)*. Springer, 1990. 109, 123

[22] Alain Bensoussan and J-L Lions. *Applications of variational inequalities in stochastic control*, volume 12. Elsevier, 2011. 274

[23] Dirk Bergemann and Stephen Morris. Bayes correlated equilibrium and the comparison of information structures in games. *Theoretical Economics*, 11(2):487–522, 2016. 173, 174, 207

[24] Dirk Bergemann and Stephen Morris. Information design: A unified perspective. *Journal of Economic Literature*, 57(1):44–95, 2019. 170, 173

[25] Martino Bernasconi, Matteo Castiglioni, Alberto Marchesi, Francesco Trovò, and Nicola Gatti. Constrained Φ-equilibria. *International Conference on Machine Learning (ICML)*, 2023. 219, 226, 308, 316, 317

[26] Martino Bernasconi, Matteo Castiglioni, Andrea Celli, and Gabriele Farina. On the role of constraints in the complexity of min-max optimization. *arXiv:2411.03248*, 2024. 275

[27] Martin Bichler, Maximilian Fichtl, Stefan Heidekrüger, Nils Kohring, and Paul Sutterer. Learning equilibria in symmetric auction games using artificial neural networks. *Nature Machine Intelligence*, 3(8):687–695, 2021. 317

[28] David Blackwell. An analog of the minmax theorem for vector payoffs. *Pacific Journal of Mathematics*, 6:1–8, 1956. 225

[29] Avrim Blum and Yishay Mansour. From external to internal regret. *Journal of Machine Learning Research*, 8:1307–1324, 2007. 219, 225, 228, 229

[30] Jean Bourgain. Random points in isotropic convex sets. *Convex geometric analysis*, 34: 53–58, 1996. 227

[31] Michael Bowling, Neil Burch, Michael Johanson, and Oskari Tammelin. Heads-up limit hold'em poker is solved. *Science*, 347(6218), January 2015. 61, 175, 220

[32] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. 274

[33] Mark Braverman, Omid Etesami, and Elchanan Mossel. Mafia: A theoretical study of players and coalitions in a partial information environment. *Annals of Applied Probability*, 18:825–846, 2006. 105

[34] Mark Braverman, Jieming Mao, Jon Schneider, and Matt Weinberg. Selling to a no-regret buyer. *ACM Conference on Economics and Computation (EC)*, 2018. 218

[35] Noam Brown and Tuomas Sandholm. Regret-based pruning in extensive-form games. *Neural Information Processing Systems (NeurIPS)*, 2015. 37

[36] Noam Brown and Tuomas Sandholm. Safe and nested subgame solving for imperfect-information games. *Neural Information Processing Systems (NeurIPS)*, 2017. 3, 26, 30, 31, 32, 36, 49, 61

[37] Noam Brown and Tuomas Sandholm. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359(6374):418–424, 2018. 2, 3, 26, 28, 32, 61, 175, 220

[38] Noam Brown and Tuomas Sandholm. Solving imperfect-information games via discounted regret minimization. *AAAI Conference on Artificial Intelligence (AAAI)*, 2019. 24, 61, 254

[39] Noam Brown and Tuomas Sandholm. Superhuman AI for multiplayer poker. *Science*, 365 (6456):885–890, 2019. 2, 3, 26, 28, 29, 32, 60, 61, 151, 220

[40] Noam Brown, Tuomas Sandholm, and Brandon Amos. Depth-limited solving for imperfect-information games. *Neural Information Processing Systems (NeurIPS)*, 2018. 26, 30

[41] Noam Brown, Anton Bakhtin, Adam Lerer, and Qucheng Gong. Combining deep reinforcement learning and search for imperfect-information games. *Neural Information Processing Systems (NeurIPS)*, 2020. 26, 30

[42] Regina S Burachik and R Diaz Millan. A projection algorithm for non-monotone variational inequalities. *Set-Valued and Variational Analysis*, 28(1):149–166, 2020. 275

[43] Neil Burch, Michael Johanson, and Michael Bowling. Solving imperfect information games using decomposition. *AAAI Conference on Artificial Intelligence (AAAI)*, 2014. 3, 26, 30, 31, 32, 33, 36

[44] Neil Burch, Matej Moravcik, and Martin Schmid. Revisiting CFR+ and alternating updates. *Journal of Artificial Intelligence Research*, 64:429–443, 2019. 21

[45] Yang Cai, Ozan Candogan, Constantinos Daskalakis, and Christos H. Papadimitriou. Zero-sum polymatrix games: A generalization of minmax. *Mathematics of Operations Research*, 41(2):648–655, 2016. 293, 328

[46] Yang Cai, Constantinos Daskalakis, Haipeng Luo, Chen-Yu Wei, and Weiqiang Zheng. On tractable Φ-equilibria in non-concave games. *Neural Information Processing Systems (NeurIPS)*, 2024. 226, 288, 308, 317, 318

[47] Yang Cai, Haipeng Luo, Chen-Yu Wei, and Weiqiang Zheng. Near-optimal policy optimization for correlated equilibrium in general-sum markov games. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2024. 226

[48] Yang Cai, Argyris Oikonomou, and Weiqiang Zheng. Accelerated algorithms for constrained nonconvex-nonconcave min-max optimization and comonotone inclusion. *International Conference on Machine Learning (ICML)*, 2024. 275

[49] Modibo K. Camara, Jason D. Hartline, and Aleck C. Johnsen. Mechanisms for a no-regret agent: Beyond the common prior. *Symposium on Foundations of Computer Science (FOCS)*, 2020. 193

[50] Colin Camerer. *Behavioral Game Theory: Experiments in Strategic Interaction*. Princeton University Press, 2003. 62

[51] Murray Campbell, A Joseph Hoane Jr, and Feng-hsiung Hsu. Deep Blue. *Artificial Intelligence*, 134(1-2):57–83, 2002. 26

[52] Ozan Candogan, Ishai Menache, Asuman E. Ozdaglar, and Pablo A. Parrilo. Flows and decompositions of games: Harmonic and potential games. *Mathematics of Operations Research*, 36(3):474–503, 2011. 281, 291

[53] Constantine Caramanis, Dimitris Fotakis, Alkis Kalavasis, Vasilis Kontonis, and Christos Tzamos. Optimizing solution-samplers for combinatorial problems: the landscape of policy-gradient methods. *Neural Information Processing Systems (NeurIPS)*, 2024. 280

[54] Luca Carminati, Brian Hu Zhang, Gabriele Farina, Nicola Gatti, and Tuomas Sandholm. Hidden-role games: Equilibrium concepts and computation. *ACM Conference on Economics and Computation (EC)*, 2024. 11, 111, 116, 135, 136

[55] Luca Carminati, Brian Hu Zhang, Federico Cacciamani, Junkang Li, Gabriele Farina, Nicola Gatti, and Tuomas Sandholm. Efficient representations for team and imperfect-recall equilibrium computation. *Under submission*, 2025. 11

[56] Andrea Celli and Nicola Gatti. Computational results for extensive-form adversarial team games. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. 65, 108, 113, 134

[57] Andrea Celli, Stefano Coniglio, and Nicola Gatti. Private Bayesian persuasion with sequential games. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. 140, 174

[58] Michal Chalamish and Sarit Kraus. Automed: an automated mediator for multi-issue bilateral negotiations. *Autonomous Agents and Multi-Agent Systems*, 24(3):536–564, 2012. 140

[59] Jianer Chen, Benny Chor, Mike Fellows, Xiuzhen Huang, David Juedes, Iyad A Kanj, and Ge Xia. Tight lower bounds for certain parameterized NP-hard problems. *Information and Computation*, 201(2):216–231, 2005. 85, 164

[60] Xinyi Chen, Angelica Chen, Dean Foster, and Elad Hazan. AI safety by debate via regret minimization, 2023. 228

[61] Chao-Kai Chiang, Tianbao Yang, Chia-Jung Lee, Mehrdad Mahdavi, Chi-Jen Lu, Rong Jin, and Shenghuo Zhu. Online optimization with gradual variations. *Conference on Learning Theory*, 2012. 23, 175, 179

[62] Paul F. Christiano. Solving avalon with whispering. 2018. Blog post. Available at https://sideways-view.com/2018/08/25/solving-avalon-with-whispering/. xxiv, 105, 134, 136

[63] Francis Chu and Joseph Halpern. On the NP-completeness of finding an optimal strategy in games with common payoffs. *International Journal of Game Theory*, 2001. 87, 147

[64] Paolo Ciancarini and Gian Piero Favini. Monte Carlo tree search techniques in the game of Kriegspiel. *International Joint Conference on Artificial Intelligence (IJCAI)*, 2009. 27

[65] Michael B Cohen, Yin Tat Lee, and Zhao Song. Solving linear programs in the current matrix multiplication time. *Journal of the ACM*, 68(1):1–39, 2021. 247, 254

[66] Patrick L. Combettes and Teemu Pennanen. Proximal methods for cohypomonotone operators. *SIAM Journal on Control and Optimization*, 43(2):731–742, 2004. 300

[67] Vincent Conitzer and Tuomas Sandholm. Complexity of mechanism design. *Conference on Uncertainty in Artificial Intelligence (UAI)*, Edmonton, Canada, 2002. 140, 172, 173

[68] Vincent Conitzer and Tuomas Sandholm. Self-interested automated mechanism design and implications for optimal combinatorial auctions. *ACM Conference on Electronic Commerce (ACM-EC)*, New York, NY, 2004. 140, 172, 173

[69] Ross Cressman, William G Morrison, and Jean-François Wen. On the evolutionary dynamics of crime. *Canadian Journal of Economics*, pages 1101–1117, 1998. 293

[70] Mete Şeref Ahunbay. First-order (coarse) correlated equilibria in non-concave games. *arXiv:2403.18174*, 2025. 226, 288

[71] Yuval Dagan, Constantinos Daskalakis, Maxwell Fishelson, and Noah Golowich. From external to swap regret 2.0: An efficient reduction for large action spaces. *Symposium on Theory of Computing (STOC)*, 2024. 10, 219, 223, 224, 225, 230, 266, 272, 332

[72] Marina Danilova, Pavel Dvurechensky, Alexander Gasnikov, Eduard Gorbunov, Sergey Guminov, Dmitry Kamzolov, and Innokentiy Shibaev. Recent theoretical advances in non-convex optimization. *High-Dimensional Optimization and Probability: With a View Towards Data Science*, pages 79–163. Springer, 2022. 280

[73] Christoph Dann, Yishay Mansour, Mehryar Mohri, Jon Schneider, and Balasubramanian Sivan. Pseudonorm approachability and applications to regret minimization. *Internatinal Conference on Algorithmic Learning Theory (ALT)*, 2023. 219

[74] Partha Dasgupta and Eric Maskin. The existence of equilibrium in discontinuous economic games 1: Theory. *Review of Economic Studies*, 53:1–26, 1986. 317

[75] Constantinos Daskalakis. Non-concave games: A challenge for game theory's next 100 years. *Nobel symposium" One Hundred Years of Game Theory: Future Applications and*

Challenges, 2022. 226, 308

[76] Constantinos Daskalakis and Christos Papadimitriou. Continuous local search. *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2011. 280

[77] Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a nash equilibrium. *SIAM Journal on Computing*, 39(1), 2009. 332

[78] Constantinos Daskalakis, Dylan J. Foster, and Noah Golowich. Independent policy gradient methods for competitive reinforcement learning. *Neural Information Processing Systems (NeurIPS)*, 2020. 275

[79] Constantinos Daskalakis, Maxwell Fishelson, and Noah Golowich. Near-optimal no-regret learning in general games. *Neural Information Processing Systems (NeurIPS)*, 2021. 180, 291

[80] Constantinos Daskalakis, Gabriele Farina, Noah Golowich, Tuomas Sandholm, and Brian Hu Zhang. A lower bound on swap regret in extensive-form games. *arXiv:2406.13116*, 2024. 13, 225

[81] Constantinos Daskalakis, Gabriele Farina, Maxwell Fishelson, Charilaos Pipis, and Jon Schneider. Efficient learning and computation of linear correlated equilibrium in general convex games. *Symposium on Theory of Computing (STOC)*, 2025. xxvi, 220, 223, 224, 225, 226, 231, 232, 233, 238, 239, 240, 242, 243, 244, 245, 261, 262, 280, 307, 315, 316, 318

[82] Damek Davis, Dmitriy Drusvyatskiy, Yin Tat Lee, Swati Padmanabhan, and Guanghao Ye. A gradient sampling method with complexity guarantees for Lipschitz functions in high and low dimensions. *Neural Information Processing Systems (NeurIPS)*, 2022. 281, 311

[83] Argyrios Deligkas, John Fearnley, Alexandros Hollender, and Themistoklis Melissourgos. Tight inapproximability for graphical games. *AAAI Conference on Artificial Intelligence (AAAI)*, 2023. 293, 314

[84] Yuan Deng, Jon Schneider, and Balasubramanian Sivan. Strategizing against no-regret learners. *Neural Information Processing Systems (NeurIPS)*, 2019. 226

[85] Yuan Deng, Vahab Mirrokni, and Song Zuo. Non-clairvoyant dynamic mechanism design with budget constraints and beyond. *ACM Conference on Economics and Computation (EC)*, 2021. 184

[86] Greg d'Eon, Neil Newman, and Kevin Leyton-Brown. Understanding iterative combinatorial auction designs via multi-agent reinforcement learning. *ACM Conference on Economics and Computation (EC)*, 2024. 332

[87] Jelena Diakonikolas, Constantinos Daskalakis, and Michael I. Jordan. Efficient methods for structured nonconvex-nonconcave min-max optimization. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021. 281, 294, 300

[88] Peter J.C. Dickinson and Luuk Gijben. On the computational complexity of membership problems for the completely positive cone and its dual. *Computational Optimization and Applications*, 57:403–415, 2014. 325

[89] Miroslav Dudík and Geoffrey J. Gordon. A sampling-based approach to computing

equilibria in succinct extensive-form games. *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2009. 219, 220

[90] Paul Dütting, Zhe Feng, Harikrishna Narasimhan, David Parkes, and Sai Srivatsa Ravindranath. Optimal auctions through deep learning. *International Conference on Machine Learning (ICML)*, 2019. 6

[91] Liad Erez, Tal Lancewicki, Uri Sherman, Tomer Koren, and Yishay Mansour. Regret minimization and convergence to equilibria in general-sum Markov games. *International Conference on Machine Learning (ICML)*, 2023. 226

[92] Kousha Etessami and Mihalis Yannakakis. On the complexity of Nash equilibria and other fixed points (extended abstract). *Symposium on Foundations of Computer Science (FOCS)*, 2007. 292, 315

[93] Francisco Facchinei and Jong-Shi Pang. *Finite-dimensional variational inequalities and complementarity problems*. Springer, 2003. 274

[94] Meta Fundamental AI Research Diplomacy Team (FAIR), Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, Athul Paul Jacob, Mojtaba Komeili, Karthik Konath, Minae Kwon, Adam Lerer, Mike Lewis, Alexander H. Miller, Sasha Mitts, Adithya Renduchintala, Stephen Roller, Dirk Rowe, Weiyan Shi, Joe Spisak, Alexander Wei, David Wu, Hugh Zhang, and Markus Zijlstra. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074, 2022. 61

[95] Gabriele Farina and Charilaos Pipis. Polynomial-time linear-swap regret minimization in imperfect-information sequential games. *Neural Information Processing Systems (NeurIPS)*, 2023. xiv, xxi, xxiv, 219, 220, 229, 247, 254, 257, 262, 263

[96] Gabriele Farina and Charilaos Pipis. Polynomial-time computation of exact Phi-equilibria in polyhedral games. *Neural Information Processing Systems (NeurIPS)*, 2024. xxvi, 9, 220, 231, 232, 280

[97] Gabriele Farina and Tuomas Sandholm. Polynomial-time computation of optimal correlated equilibria in two-player extensive-form games with public chance moves and beyond. *Neural Information Processing Systems (NeurIPS)*, 2020. xix, 101, 140, 152, 160, 163, 164

[98] Gabriele Farina and Tuomas Sandholm. Fast payoff matrix sparsification techniques for structured extensive-form games. *AAAI Conference on Artificial Intelligence (AAAI)*, 2022. 54

[99] Gabriele Farina, Nicola Gatti, and Tuomas Sandholm. Practical exact algorithm for trembling-hand equilibrium refinements in games. *Neural Information Processing Systems (NeurIPS)*, 2018. 135

[100] Gabriele Farina, Christian Kroer, and Tuomas Sandholm. Regret circuits: Composability of regret minimizers. *International Conference on Machine Learning*, 2019. 23, 24

[101] Gabriele Farina, Chun Kai Ling, Fei Fang, and Tuomas Sandholm. Correlation in extensive-form games: Saddle-point formulation and benchmarks. *Neural Information Processing Systems (NeurIPS)*, 2019. 140, 157, 176, 180, 182, 262

[102] Gabriele Farina, Tommaso Bianchi, and Tuomas Sandholm. Coarse correlation in extensive-form games. *AAAI Conference on Artificial Intelligence*, 2020. 6, 19, 138, 152, 154, 157, 220

[103] Gabriele Farina, Andrea Celli, Nicola Gatti, and Tuomas Sandholm. Connecting optimal ex-ante collusion in teams to extensive-form correlation: Faster algorithms and positive complexity results. *International Conference on Machine Learning*, 2021. 24, 99, 100, 254

[104] Gabriele Farina, Christian Kroer, and Tuomas Sandholm. Faster game solving via predictive Blackwell approachability: Connecting regret matching and mirror descent. *AAAI Conference on Artificial Intelligence (AAAI)*, 2021. 23, 61

[105] Gabriele Farina, Ioannis Anagnostides, Haipeng Luo, Chung-Wei Lee, Christian Kroer, and Tuomas Sandholm. Near-optimal no-regret learning dynamics for general convex games. *Neural Information Processing Systems (NeurIPS)*, 2022. 180

[106] Gabriele Farina, Andrea Celli, Alberto Marchesi, and Nicola Gatti. Simple uncoupled no-regret learning dynamics for extensive-form correlated equilibrium. *Journal of the ACM*, 69(6):41:1–41:41, 2022. 151, 220

[107] Gabriele Farina, Julien Grand-Clément, Christian Kroer, Chung-Wei Lee, and Haipeng Luo. Regret matching+: (in)stability and fast convergence in games. *Neural Information Processing Systems (NeurIPS)*, 2024. 52

[108] John Fearnley, Paul Goldberg, Alexandros Hollender, and Rahul Savani. The complexity of gradient descent: CLS = PPAD ∩ PLS. *Journal of the ACM*, 70(1):7:1–7:74, 2023. 275, 280, 287

[109] Francoise Forges. An approach to communication equilibria. *Econometrica: Journal of the Econometric Society*, 54(6):1375–1385, 1986. 6, 110, 120, 138, 144, 145, 154, 168, 191, 207, 229, 259, 260

[110] Françoise Forges and Frédéric Koessler. Communication equilibria with partially verifiable types. *Journal of Mathematical Economics*, 41(7):793–811, 2005. 138, 142, 168, 261

[111] Françoise Forges and Indrajit Ray. "subjectivity and correlation in randomized strategies": Back to the roots. *Journal of Mathematical Economics*, 114:103044, 2024. 6

[112] Dean Foster and Rakesh Vohra. Calibrated learning and correlated equilibrium. *Games and Economic Behavior*, 21:40–55, 1997. 225

[113] Dean P. Foster and Sergiu Hart. Smooth calibration, leaky forecasts, finite recall, and nash dynamics. *Games and Economic Behavior*, 109:271–293, 2018. 225

[114] Daniel Friedman. Evolutionary games in economics. *Econometrica*, 59(3):637–666, 1991. 293

[115] Qiang Fu, Dongchu Xu, and Ashia Camage Wilson. Accelerated stochastic optimization methods under quasar-convexity. *International Conference on Machine Learning (ICML)*, 2023. 280

[116] Drew Fudenberg and Jean Tirole. *Game Theory*. MIT Press, 1991. 192

[117] Kaito Fujii. Bayes correlated equilibria and no-regret dynamics. *arXiv:2304.05005*, 2023.

219, 228, 257, 321

[118] Masabumi Furuhata, Maged Dessouky, Fernando Ordóñez, Marc-Etienne Brunet, Xiaoqing Wang, and Sven Koenig. Ridesharing: The state-of-the-art and future directions. *Transportation Research Part B: Methodological*, 57:28–46, 2013. 140

[119] Jiarui Gan, Rupak Majumdar, Goran Radanovic, and Adish Singla. Bayesian persuasion in sequential decision-making. *AAAI Conference on Artificial Intelligence (AAAI)*, 2022. 140, 174

[120] Anat Ganor and Karthik C. S. Communication complexity of correlated equilibrium with small support. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, 2018. 228

[121] Sam Ganzfried and Tuomas Sandholm. Endgame solving in large imperfect-information games. *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2015. 26, 61

[122] Sam Ganzfried and Tuomas Sandholm. Safe opponent exploitation. *ACM Transaction on Economics and Computation*, 3(2):8:1–28, 2015. 35

[123] Sam Ganzfried, Tuomas Sandholm, and Kevin Waugh. Strategy purification and thresholding: Effective non-equilibrium approaches for playing large games. *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2012. 42

[124] Oscar Garcia Morchon, Heribert Baldus, Tobias Heer, and Klaus Wehrle. Cooperative security in distributed sensor networks. *International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, 2007. 105

[125] Oscar Garcia-Morchon, Dmitriy Kuptsov, Andrei Gurtov, and Klaus Wehrle. Cooperative security in distributed networks. *Computer Communications*, 36(12):1284–1297, 2013. 105

[126] Ryan W Gardner, Gino Perrotta, Anvay Shah, Shivaram Kalyanakrishnan, Kevin A Wang, Gregory Clark, Timo Bertram, Johannes Fürnkranz, Martin Müller, Brady P Garrison, et al. The machine reconnaissance blind chess tournament of NeurIPS 2022. *NeurIPS 2022 Competition Track*, 2023. 27

[127] Angeliki Giannou, Emmanouil-Vasileios Vlatakis-Gkaragkounis, and Panayotis Mertikopoulos. On the rate of convergence of regularized learning in games: From bandits and uncertainty to optimism and beyond. *Neural Information Processing Systems (NeurIPS)*, 2021. 218

[128] Angeliki Giannou, Emmanouil-Vasileios Vlatakis-Gkaragkounis, and Panayotis Mertikopoulos. Survival of the strictest: Stable and unstable equilibria under regularized learning with partial information. *Conference on Learning Theory (COLT)*. PMLR, 2021. 218

[129] Andrew Gilpin and Tuomas Sandholm. A competitive Texas hold'em poker player via automated abstraction and real-time equilibrium computation. *National Conference on Artificial Intelligence (AAAI)*, 2006. 61

[130] Paul W. Goldberg and Aaron Roth. Bounds for the query complexity of approximate

equilibria. *ACM Transaction on Economics and Computation*, 4(4):24:1–24:25, 2016. 228

[131] Allen Goldstein. Optimization of Lipschitz continuous functions. *Mathematical Programming*, 13:14–22, 1977. 311

[132] Geoffrey J Gordon, Amy Greenwald, and Casey Marks. No-regret learning in convex games. *International Conference on Machine Learning (ICML)*, 2008. 9, 219, 220, 223, 224, 228, 242, 244, 257, 272, 308

[133] Robert M. Gower, Othmane Sebbouh, and Nicolas Loizou. SGD for structured nonconvex functions: Learning rates, minibatching and interpolation. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021. 280

[134] J Green and J-J Laffont. Characterization of satisfactory mechanisms for the revelation of preferences for public goods. *Econometrica*, 45:427–438, 1977. 139, 148

[135] Amy Greenwald and Keith Hall. Correlated Q-learning. *International Conference on Machine Learning (ICML)*, Washington, DC, USA, 2003. 219, 226, 227

[136] Amy Greenwald and Amir Jafari. A general class of no-regret learning algorithms and game-theoretic equilibria. *Conference on Learning Theory (COLT)*, Washington, D.C., 2003. 220, 308

[137] Martin Grötschel, László Lovász, and Alexander Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1:169–197, 1981. 237

[138] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimizations*. Springer-Verlag, 1993. xxvi, 90, 227, 262, 277, 284, 285, 286, 294, 295

[139] Gurobi Optimization, LLC. Gurobi optimizer reference manual, 2020. 45

[140] Guru Guruganesh, Yoav Kolumbus, Jon Schneider, Inbal Talgam-Cohen, Emmanouil-Vasileios Vlatakis-Gkaragkounis, Joshua R. Wang, and S. Matthew Weinberg. Contracting with a learning agent. *Neural Information Processing Systems (NeurIPS)*, 2024. 226

[141] Moritz Hardt, Tengyu Ma, and Benjamin Recht. Gradient descent learns linear dynamical systems. *Journal of Machine Learning Research*, 19(1):1025–1068, 2018. 280

[142] John Harsanyi. Game with incomplete information played by Bayesian players. *Management Science*, 14:159–182; 320–334; 486–502, 1967–68. 114

[143] Sergiu Hart and Andreu Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68:1127–1150, 2000. 22, 225

[144] Johan Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001. 87, 129

[145] Elad Hazan. Introduction to online convex optimization. *Foundations and Trends in Optimization*, 2(3-4):157–325, 2016. 262

[146] Elad Hazan and Satyen Kale. Computational equivalence of fixed points and no regret algorithms, and convergence to equilibria. *Neural Information Processing Systems (NeurIPS)*, 2007. 222, 223, 224, 263, 264

[147] Oliver Hinder, Aaron Sidford, and Nimit Sharad Sohoni. Near-optimal methods for

minimizing star-convex functions and beyond. *Conference on Learning Theory (COLT)*, 2020. 280, 287

[148] Michael D. Hirsch, Christos H. Papadimitriou, and Stephen A. Vavasis. Exponential lower bounds for finding Brouwer fix points. *Journal of Complexity*, 5(4):379–416, 1989. 315

[149] Josef Hofbauer and Karl H Schlag. Sophisticated imitation in cyclic games. *Journal of Evolutionary Economics*, 10(5):523–543, 2000. 293

[150] Feng-Hsiung Hsu. *Behind Deep Blue: Building the Computer that Defeated the World Chess Champion*. Princeton University Press, 2002. 2

[151] Lunjia Hu and Yifan Wu. Predict to minimize swap regret for all payoff-bounded tasks. *Symposium on Foundations of Computer Science (FOCS)*, 2024. 226

[152] Kevin Huang and Shuzhong Zhang. Beyond monotone variational inequalities: Solution methods and iteration complexities. *arXiv:2304.04153*, 2023. 287

[153] Wan Huang and Bernhard von Stengel. Computing an extensive-form correlated equilibrium in polynomial time. *International Workshop on Internet and Network Economics*. Springer, 2008. 138, 151, 219, 220

[154] Evan Hubinger, Carson Denison, Jesse Mu, Mike Lambert, Meg Tong, Monte MacDiarmid, Tamera Lanham, Daniel M. Ziegler, Tim Maxwell, Newton Cheng, Adam Jermyn, Amanda Askell, Ansh Radhakrishnan, Cem Anil, David Duvenaud, Deep Ganguli, Fazl Barez, Jack Clark, Kamal Ndousse, Kshitij Sachan, Michael Sellitto, Mrinank Sharma, Nova DasSarma, Roger Grosse, Shauna Kravec, Yuntao Bai, Zachary Witten, Marina Favaro, Jan Brauner, Holden Karnofsky, Paul Christiano, Samuel R. Bowman, Logan Graham, Jared Kaplan, Sören Mindermann, Ryan Greenblatt, Buck Shlegeris, Nicholas Schiefer, and Ethan Perez. Sleeper agents: Training deceptive llms that persist through safety training. *arXiv:2401.05566*, 2024. 105

[155] Geoffrey Irving, Paul F. Christiano, and Dario Amodei. AI safety via debate. *arXiv:1805.00899*, 2018. 105

[156] Sergei Izmalkov, Silvio Micali, and Matt Lepinski. Rational secure computation and ideal mechanism design. *Symposium on Foundations of Computer Science (FOCS)*, 2005. 110

[157] Eric Jackson. A time and space efficient algorithm for approximately solving large imperfect information games. *AAAI Workshop on Computer Poker and Imperfect Information*, 2014. 26

[158] Jiaming Ji, Tianyi Qiu, Boyuan Chen, Borong Zhang, Hantao Lou, Kaile Wang, Yawen Duan, Zhonghao He, Jiayi Zhou, Zhaowei Zhang, Fanzhi Zeng, Kwan Yee Ng, Juntao Dai, Xuehai Pan, Aidan O'Gara, Yingshan Lei, Hua Xu, Brian Tse, Jie Fu, Stephen McAleer, Yaodong Yang, Yizhou Wang, Song-Chun Zhu, Yike Guo, and Wen Gao. AI alignment: A comprehensive survey. *arXiv:2310.19852*, 2023. 105

[159] Albert Xin Jiang and Kevin Leyton-Brown. Polynomial-time computation of exact correlated equilibrium in compact games. *Games and Economic Behavior*, 91:347–359, 2015. 151, 280

[160] Albert Xin Jiang, Ariel Procaccia, Yundi Qian, Nisarg Shah, and Milind Tambe. De-

fender (mis)coordination in security games. *International Joint Conference on Artificial Intelligence (IJCAI)*, 2013. 62

[161] Haotian Jiang, Yin Tat Lee, Zhao Song, and Sam Chiu-wai Wong. An improved cutting plane method for convex optimization, convex-concave games, and its applications. *Symposium on Theory of Computing (STOC)*, 2020. 303

[162] Chi Jin, Qinghua Liu, Yuanhao Wang, and Tiancheng Yu. V-learning - A simple, efficient, decentralized algorithm for multiagent reinforcement learning. *Mathematics of Operations Research*, 49(4):2295–2322, 2024. 226

[163] Michael Johanson, Kevin Waugh, Michael Bowling, and Martin Zinkevich. Accelerating best response calculation in large extensive games. *International Joint Conference on Artificial Intelligence (IJCAI)*, 2011. 29, 32

[164] David S. Johnson, Christos H. Papadimitriou, and Mihalis Yannakakis. How easy is local search? *Journal of Computer and System Sciences*, 37(1):79–100, 1988. 280

[165] Michael I. Jordan, Guy Kornowski, Tianyi Lin, Ohad Shamir, and Manolis Zampetakis. Deterministic nonsmooth nonconvex optimization. *Conference on Learning Theory (COLT)*, 2023. 281, 311

[166] Fivos Kalogiannis and Ioannis Panageas. Zero-sum polymatrix markov games: Equilibrium collapse and efficient computation of nash equilibria. *Neural Information Processing Systems (NeurIPS)*, 2023. 328

[167] Emir Kamenica and Matthew Gentzkow. Bayesian persuasion. *American Economic Review*, 101(6):2590–2615, 2011. 6, 140, 173, 188, 207

[168] Mamoru Kaneko and J Jude Kline. Behavior strategies, mixed strategies and perfect recall. *International Journal of Game Theory*, 24(2):127–145, 1995. 93

[169] Ravi Kannan, László Lovász, and Miklós Simonovits. Random walks and an $o^*(n^5)$ volume algorithm for convex bodies. *Random Structures and Algorithms*, 11(1):1–50, 1997. 227

[170] Bruce M Kapron and Koosha Samieefar. The computational complexity of variational inequalities and applications in game theory. *arXiv:2411.04392*, 2024. 275

[171] Jonathan Katz. Bridging game theory and cryptography: Recent results and future directions. *Theory of Cryptography (TCC)*, 2008. 110

[172] Andrew Kephart and Vincent Conitzer. Complexity of mechanism design with signaling costs. *Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2015. 140, 173

[173] Andrew Kephart and Vincent Conitzer. The revelation principle for mechanism design with signaling costs. *ACM Transaction on Economics and Computation*, 9(1):1–35, 2021. 140, 173, 261

[174] David Kinderlehrer and Guido Stampacchia. *An introduction to variational inequalities and their applications*. SIAM, 2000. 274, 310

[175] Levente Kocsis and Csaba Szepesvári. Bandit based Monte-Carlo planning. *European Conference on Maching Learning (ECML)*, pages 282–293. Springer, 2006. 50

[176] Daphne Koller and Nimrod Megiddo. The complexity of two-person zero-sum games in extensive form. *Games and Economic Behavior*, 4(4):528–552, October 1992. 62, 87, 127

[177] Daphne Koller, Nimrod Megiddo, and Bernhard von Stengel. Fast algorithms for finding randomized strategies in game trees. *ACM Symposium on Theory of Computing (STOC)*, 1994. 20

[178] Yoav Kolumbus and Noam Nisan. Auctions between regret-minimizing agents. Frédérique Laforest, Raphaël Troncy, Elena Simperl, Deepak Agarwal, Aristides Gionis, Ivan Herman, and Lionel Médini, editors, *ACM Web Conference (WWW)*, 2022. 193

[179] Kavya Kopparapu, Edgar A. Duéñez-Guzmán, Jayd Matyas, Alexander Sasha Vezhnevets, John P. Agapiou, Kevin R. McKee, Richard Everett, Janusz Marecki, Joel Z. Leibo, and Thore Graepel. Hidden agenda: a social deduction game with diverse learned equilibria. *arXiv:2201.01816*, 2022. 105

[180] Galina M Korpelevich. The extragradient method for finding saddle points and other problems. *Matecon*, 12:747–756, 1976. 275, 278

[181] Vojtěch Kovařík, Dominik Seitz, and Viliam Lisỳ. Value functions for depth-limited solving in imperfect-information games. *AAAI Reinforcement Learning in Games Workshop*, 2021. 26, 30

[182] Vojtěch Kovařík, Martin Schmid, Neil Burch, Michael Bowling, and Viliam Lisỳ. Rethinking formal models of partially observable multiagent decision making. *Artificial Intelligence*, 303:103645, 2022. 70

[183] Mark W. Krentel. The complexity of optimization problems. *Journal of Computer and System Sciences*, 36(3):490–509, 1988. 90

[184] Christian Kroer and Tuomas Sandholm. Discretization of continuous action spaces in extensive-form games. *Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2015. 177

[185] Christian Kroer and Tuomas Sandholm. Imperfect-recall abstractions with bounds in games. *ACM Conference on Economics and Computation (EC)*, 2016. 62

[186] H. W. Kuhn. Extensive games. *Proceedings of the National Academy of Sciences*, 36: 570–576, 1950. 61

[187] H. W. Kuhn. A simplified two-person poker. H. W. Kuhn and A. W. Tucker, editors, *Contributions to the Theory of Games*, volume 1 of *Annals of Mathematics Studies, 24*, pages 97–103. Princeton University Press, Princeton, New Jersey, 1950. 46, 99, 182, 262

[188] H. W. Kuhn. Extensive games and the problem of information. H. W. Kuhn and A. W. Tucker, editors, *Contributions to the Theory of Games*, volume 2 of *Annals of Mathematics Studies, 28*, pages 193–216. Princeton University Press, Princeton, NJ, 1953. 18, 224, 321

[189] Nicolas S. Lambert, Adrian Marple, and Yoav Shoham. On equilibria in games with imperfect recall. *Games and Economic Behavior*, 113:164–185, 2019. 224

[190] Marc Lanctot, Richard Gibson, Neil Burch, Martin Zinkevich, and Michael Bowling. No-regret learning in extensive-form games with imperfect recall. *International Conference on Machine Learning (ICML)*, 2012. 62

[191] Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning. *Neural Information Processing Systems (NeurIPS)*, 2017. 184

[192] Marc Lanctot, Edward Lockhart, Jean-Baptiste Lespiau, Vinicius Zambaldi, Satyaki Upadhyay, Julien Pérolat, Sriram Srinivasan, Finbarr Timbers, Karl Tuyls, Shayegan Omidshafiei, Daniel Hennes, Dustin Morrill, Paul Muller, Timo Ewalds, Ryan Faulkner, János Kramár, Bart De Vylder, Brennan Saeta, James Bradbury, David Ding, Sebastian Borgeaud, Matthew Lai, Julian Schrittwieser, Thomas Anthony, Edward Hughes, Ivo Danihelka, and Jonah Ryan-Davis. OpenSpiel: A framework for reinforcement learning in games. *arXiv:1908.09453*, 2019. 45, 46

[193] Jasper C. H. Lee and Paul Valiant. Optimizing star-convex functions. *Symposium on Foundations of Computer Science (FOCS)*, 2016. 281, 287

[194] Davide Legacci, Panayotis Mertikopoulos, Christos H. Papadimitriou, Georgios Piliouras, and Bary S. R. Pradelski. No-regret learning in harmonic games: Extrapolation in the face of conflicting interests. *Neural Information Processing Systems (NeurIPS)*, 2024. 291

[195] Ming Lei and Yiran He. An extragradient method for solving variational inequalities without monotonicity. *Journal of Optimization Theory and Applications*, 188:432–446, 2021. 275

[196] Adam Lerer, Hengyuan Hu, Jakob Foerster, and Noam Brown. Improving policies via search in cooperative partially observable games. *Proceedings of the AAAI conference on artificial intelligence*, 2020. 2

[197] Tianyi Lin and Michael I Jordan. Perseus: A simple and optimal high-order method for variational inequalities. *Mathematical Programming*, 209:1–42, 2024. 275

[198] Yehuda Lindell. Secure multiparty computation (MPC). *Cryptology ePrint Archive*, 2020. 109

[199] Viliam Lisỳ, Marc Lanctot, and Michael H Bowling. Online Monte Carlo counterfactual regret minimization for search in imperfect information games. *Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2015. 99, 182

[200] Heng Liu. Correlation and unmediated cheap talk in repeated games with imperfect monitoring. *International Journal of Game Theory*, 46:1037–1069, 2017. 110

[201] Weiming Liu, Haobo Fu, Qiang Fu, and Yang Wei. Opponent-limited online search for imperfect information games. *International Conference on Machine Learning (ICML)*, 2023. 37

[202] László Lovász and Santosh S. Vempala. Simulated annealing in convex bodies and an $o^*(n^4)$ volume algorithm. *Journal of Computer and System Sciences*, 72(2):392–417, 2006. 227, 284

[203] Hans-Jakob Lüthi. On the solution of variational inequalities by the ellipsoid method. *Mathematics of Operations Research*, 10(3):515–522, 1985. 276

[204] Hongyao Ma, Fei Fang, and David C Parkes. Spatio-temporal pricing for ridesharing

platforms. *Operations Research*, 2021. 140

[205] Thomas L Magnanti and Georgia Perakis. A unifying geometric solution framework and complexity analysis for variational inequalities. *Mathematical Programming*, 71(3): 327–351, 1995. 276

[206] Miltiadis Makris and Ludovic Renou. Information design in multistage games. *Theoretical Economics*, 18(4):1475–1509, 2023. 207

[207] Yishay Mansour, Mehryar Mohri, Jon Schneider, and Balasubramanian Sivan. Strategizing against learners in Bayesian games. *Conference on Learning Theory (COLT)*, 2022. 219, 220, 226

[208] Yishay Mansour, Alex Slivkins, Vasilis Syrgkanis, and Zhiwei Steven Wu. Bayesian exploration: Incentivizing exploration in Bayesian games. *Operations Research*, 70(2): 1105–1127, 2022. 140

[209] Carlos Martin and Tuomas Sandholm. Joint-perturbation simultaneous pseudo-gradient, 2024. 317

[210] Bernard Martinet. Brève communication. Régularisation d'inéquations variationnelles par approximations successives. *Revue française d'informatique et de recherche opérationnelle. Série rouge*, 4(R3):154–158, 1970. 275

[211] Michael Maschler, Shmuel Zamir, and Eilon Solan. *Game Theory*. Cambridge University Press, 2020. 113

[212] Stephen McAleer, John B. Lanier, Roy Fox, and Pierre Baldi. Pipeline PSRO: A scalable approach for finding approximate Nash equilibria in large games. *Neural Information Processing Systems (NeurIPS)*, 2020. 184

[213] Panayotis Mertikopoulos and Zhengyuan Zhou. Learning in games with continuous action sets and unknown payoff functions. *Mathematical Programming*, 173(1-2):465–507, 2019. 275

[214] Panayotis Mertikopoulos, Bruno Lecouat, Houssam Zenati, Chuan-Sheng Foo, Vijay Chandrasekhar, and Georgios Piliouras. Optimistic mirror descent in saddle-point problems: Going the extra (gradient) mile. *International Conference on Learning Representations (ICLR)*, 2018. 275

[215] Panayotis Mertikopoulos, Christos H. Papadimitriou, and Georgios Piliouras. Cycles in adversarial regularized learning. *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2018. 275

[216] James D. Miller, Roman Yampolskiy, Olle Häggström, and Stuart Armstrong. Chess as a testing grounds for the oracle approach to AI safety. Huáscar Espinoza, John A. McDermid, Xiaowei Huang, Mauricio Castillo-Effen, Xin Cynthia Chen, José Hernández-Orallo, Seán Ó hÉigeartaigh, Richard Mallah, and Gabriel Pedroza, editors, *Proceedings of the Workshop on Artificial Intelligence Safety 2021 co-located with the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI 2021), Virtual, August, 2021*. CEUR-WS.org, 2021. 105

[217] George J. Minty. On the generalization of a direct method of the calculus of variations.

Bulletin of the American Mathematical Society, 73(3):315 – 321, 1967. 275

[218] Dov Monderer and Moshe Tennenholtz. k-Implementation. *ACM Conference on Electronic Commerce (ACM-EC)*, San Diego, CA, 2003. 215

[219] Dov Monderer and Moshe Tennenholtz. K-implementation. *Journal of Artificial Intelligence Research*, 21:37–62, 2004. 194

[220] Dov Monderer and Moshe Tennenholtz. Strong mediated equilibrium. *Artificial Intelligence*, 173(1):180–195, 2009. 169

[221] Matej Moravcik, Martin Schmid, Karel Ha, Milan Hladik, and Stephen Gaukrodger. Refining subgames in large imperfect information games. *AAAI Conference on Artificial Intelligence (AAAI)*, 2016. 3, 26, 30, 31, 32, 36, 61

[222] Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisý, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. DeepStack: Expert-level artificial intelligence in heads-up no-limit poker. *Science*, 356(6337):508–513, May 2017. 26, 30, 32, 61, 220

[223] Dustin Morrill, Ryan D'Orazio, Reca Sarfati, Marc Lanctot, James R. Wright, Amy R. Greenwald, and Michael Bowling. Hindsight and sequential rationality of correlated play. *AAAI Conference on Artificial Intelligence (AAAI)*, 2021. 219, 220

[224] Dustin Morrill, Ryan D'Orazio, Marc Lanctot, James R Wright, Michael Bowling, and Amy R Greenwald. Efficient deviation types and learning for hindsight rationality in extensive-form games. *International Conference on Machine Learning (ICML)*. PMLR, 2021. 219, 220

[225] Viraaji Mothukuri, Reza M Parizi, Seyedamin Pouriyeh, Yan Huang, Ali Dehghantanha, and Gautam Srivastava. A survey on security and privacy of federated learning. *Future Generation Computer Systems*, 115:619–640, 2021. 105

[226] H. Moulin and J.-P. Vial. Strategically zero-sum games: The class of games whose completely mixed equilibria cannot be improved upon. *International Journal of Game Theory*, 7(3-4):201–221, 1978. 6, 138, 152, 154, 220, 229, 290, 308

[227] Katta G. Murty and Santosh N. Kabadi. Some NP-complete problems in quadratic and nonlinear programming. *Math. Program.*, 39(2):117–129, June 1987. 325

[228] Roger B Myerson. Multistage games with communication. *Econometrica: Journal of the Econometric Society*, pages 323–358, 1986. 6, 110, 120, 138, 168, 191, 207, 229, 259, 260

[229] Uri Nadav and Tim Roughgarden. The limits of smoothness: A primal-dual framework for price of anarchy bounds. *International Workshop On Internet And Network Economics (WINE)*, 2010. 290

[230] John Nash. Equilibrium points in n-person games. *National Academy of Sciences*, 36: 48–49, 1950. 18, 108, 274

[231] Denis Nekipelov, Vasilis Syrgkanis, and Éva Tardos. Econometrics for learning agents. *ACM Conference on Economics and Computation (EC)*, 2015. 193

[232] Yurii Nesterov and Boris T Polyak. Cubic regularization of Newton method and its global

performance. *Mathematical programming*, 108(1):177–205, 2006. 287

[233] Georgy Noarov, Ramya Ramalingam, Aaron Roth, and Stephan Xie. High-dimensional prediction for sequential decision making. *CoRR*, abs/2310.17651, 2023. 219, 226

[234] Aidan O'Gara. Hoodwinked: Deception and cooperation in a text-based game for language models. *arXiv:2308.01404*, 2023. 105

[235] Christos Papadimitriou, George Pierrakos, Alexandros Psomas, and Aviad Rubinstein. On the complexity of dynamic mechanism design. *Games and Economic Behavior*, 2022. 140, 172

[236] Christos H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and system Sciences*, 48(3):498–532, 1994. 275

[237] Christos H Papadimitriou and Tim Roughgarden. Computing correlated equilibria in multi-player games. *Journal of the ACM*, 55(3):14, 2008. xxvi, 9, 151, 224, 230, 231, 232, 280, 282, 285, 303

[238] Chanwoo Park, Kaiqing Zhang, and Asuman Ozdaglar. Multi-player zero-sum Markov games with networked separable interactions. *Neural Information Processing Systems (NeurIPS)*, 2023. 328

[239] Peter S. Park, Simon Goldstein, Aidan O'Gara, Michael Chen, and Dan Hendrycks. AI deception: A survey of examples, risks, and potential solutions. *arXiv:2308.14752*, 2023. 105

[240] Austin Parker, Dana Nau, and VS Subrahmanian. Game-tree search with combinatorially large belief states. *International Joint Conference on Artificial Intelligence (IJCAI)*, 2005. 27

[241] Nikolas Patris and Ioannis Panageas. Learning Nash equilibria in rank-1 games. *International Conference on Learning Representations (ICLR)*, 2024. 275

[242] Binghui Peng and Aviad Rubinstein. Fast swap regret minimization and applications to approximate correlated equilibria. *Symposium on Theory of Computing (STOC)*, 2024. 10, 219, 223, 224, 225, 230, 272, 332

[243] Julien Perolat, Bart De Vylder, Daniel Hennes, Eugene Tarassov, Florian Strub, Vincent de Boer, Paul Muller, Jerome T. Connor, Neil Burch, Thomas Anthony, Stephen McAleer, Romuald Elie, Sarah H. Cen, Zhe Wang, Audrunas Gruslys, Aleksandra Malysheva, Mina Khan, Sherjil Ozair, Finbarr Timbers, Toby Pohlen, Tom Eccles, Mark Rowland, Marc Lanctot, Jean-Baptiste Lespiau, Bilal Piot, Shayegan Omidshafiei, Edward Lockhart, Laurent Sifre, Nathalie Beauguerlange, Remi Munos, David Silver, Satinder Singh, Demis Hassabis, and Karl Tuyls. Mastering the game of Stratego with model-free multiagent reinforcement learning. *Science*, 378(6623):990–996, 2022. 61

[244] Michele Piccione and Ariel Rubinstein. On the interpretation of decision problems with imperfect recall. *Games and Economic Behavior*, pages 3–24, 1997. 102, 224

[245] Georgios Piliouras, Ryann Sim, and Stratis Skoulakis. Beyond time-average convergence: Near-optimal uncoupled online learning via clairvoyant multiplicative weights update. *Neural Information Processing Systems (NeurIPS)*, 2022. 180

[246] Leonid Denisovich Popov. A modification of the Arrow-Hurwitz method of search for saddle points. *Mat. Zametki*, 28(5):777–784, 1980. 23

[247] David Brine Pritchard. *The encyclopedia of chess variants*. Games & Puzzles Publications, 1994. 27

[248] Tal Rabin. Robust sharing of secrets when the dealer is honest or cheating. *Journal of the ACM*, 41(6):1089–1109, 1994. 123

[249] Tal Rabin and Michael Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. *Symposium on Theory of Computing (STOC)*, 1989. 109, 122, 123

[250] Alexander Rakhlin and Karthik Sridharan. Online learning with predictable sequences. *Conference on Learning Theory (COLT)*, 2013. 23

[251] Alexander Rakhlin, Karthik Sridharan, and Ambuj Tewari. Online learning: Beyond regret. *Conference on Learning Theory (COLT)*, 2011. 219

[252] Sasha Rakhlin and Karthik Sridharan. Optimization, learning, and games with predictable sequences. *Advances in Neural Information Processing Systems*, 2013. 23, 175, 179

[253] I. Romanovskii. Reduction of a game with complete memory to a matrix game. *Soviet Mathematics*, 3, 1962. 20

[254] Sheldon M Ross. Goofspiel—the game of pure strategy. *Journal of Applied Probability*, 8 (3):621–625, 1971. 182

[255] Aaron Roth and Mirah Shi. Forecasting for swap regret for all downstream agents. *ACM Conference on Economics and Computation (EC)*, 2024. 226

[256] Tim Roughgarden. Intrinsic robustness of the price of anarchy. *J. ACM*, 62(5):32:1–32:42, 2015. 186, 280, 281, 288, 301

[257] Tim Roughgarden and Florian Schoppmann. Local smoothness and the price of anarchy in splittable congestion games. *Journal of Economic Theory*, 156:317–342, 2015. 288

[258] Tim Roughgarden, Vasilis Syrgkanis, and Éva Tardos. The price of anarchy in auctions. *Journal of Artificial Intelligence Research*, 59:59–101, 2017. 288

[259] Aviad Rubinstein. Inapproximability of Nash equilibrium. *Symposium on Theory of Computing (STOC)*, 2015. 275, 293, 314

[260] Stuart Russell and Jason Wolfe. Efficient belief-state AND-OR search, with application to Kriegspiel. *National Conference on Artificial Intelligence (AAAI)*, 2005. 27

[261] Tuomas Sandholm. Abstraction for solving large incomplete-information games. *AAAI Conference on Artificial Intelligence (AAAI)*, 2015. Senior Member Track. 61

[262] Tuomas Sandholm. Solving imperfect-information games. *Science*, 347(6218):122–123, 2015. 61

[263] Tuomas Sandholm, Andrew Gilpin, and Vincent Conitzer. Mixed-integer programming methods for finding Nash equilibria. *National Conference on Artificial Intelligence (AAAI)*, 2005. 292

[264] Tuomas Sandholm, Vincent Conitzer, and Craig Boutilier. Automated design of multistage mechanisms. *International Joint Conference on Artificial Intelligence (IJCAI)*, 2007. 172

[265] Marcus Schaefer and Christopher Umans. Completeness in the polynomial-time hierarchy: A compendium. *SIGACT News*, 33(3):32–49, 2002. 88

[266] Jérémy Scheurer, Mikita Balesni, and Marius Hobbhahn. Technical report: Large language models can strategically deceive their users when put under pressure. *arXiv:2311.07590*, 2023. 105

[267] Martin Schmid, Matej Moravčík, Neil Burch, Rudolf Kadlec, Josh Davidson, Kevin Waugh, Nolan Bard, Finbarr Timbers, Marc Lanctot, G Zacharias Holland, et al. Student of games: A unified learning algorithm for both perfect and imperfect information games. *Science Advances*, 9(46):eadg3256, 2023. 41, 44, 50

[268] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv:1707.06347*, 2017. 184

[269] Jack Serrino, Max Kleiman-Weiner, David C Parkes, and Josh Tenenbaum. Finding friend and foe in multi-agent games. *Neural Information Processing Systems (NeurIPS)*, 2019. 105, 111, 135

[270] Moïse Sibony. Méthodes itératives pour les équations et inéquations aux dérivées partielles non linéaires de type monotone. *Calcolo*, 7:65–183, 1970. 275

[271] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484, 2016. 2, 26, 41, 50

[272] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359, 2017. 45

[273] Maurice Sion. On general minimax theorems. *Pacific Journal of Mathematics*, 8(4): 171–176, 1958. 178, 289, 293

[274] Chaobing Song, Zhengyuan Zhou, Yichao Zhou, Yong Jiang, and Yi Ma. Optimistic dual extrapolation for coherent non-monotone variational inequalities. *Neural Information Processing Systems (NeurIPS)*, 2020. 275, 276

[275] Finnegan Southey, Michael Bowling, Bryce Larson, Carmelo Piccione, Neil Burch, Darse Billings, and Chris Rayner. Bayes' bluff: Opponent modelling in poker. *Conference on Uncertainty in Artificial Intelligence (UAI)*, July 2005. 46, 99, 182, 262

[276] Stockfish. `https://stockfishchess.org/`. 26, 41, 51

[277] Gilles Stoltz and Gábor Lugosi. Internal regret in on-line portfolio selection. *Machine Learning*, 59(1-2):125–159, 2005. 219, 225, 228

[278] Gilles Stoltz and Gábor Lugosi. Learning correlated equilibria in games with compact sets of strategies. *Games and Economic Behavior*, 59(1):187–208, 2007. 219, 220, 227, 308

[279] Michal Šustr, Vojtěch Kovařík, and Viliam Lisỳ. Monte Carlo continual resolving for online strategy computation in imperfect information games. *Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2019. 26, 30

[280] Michal Šustr, Vojtěch Kovařík, and Viliam Lisỳ. Particle value functions in imperfect information games. *AAMAS Adaptive and Learning Agents Workshop*, 2021. 30, 48

[281] Vasilis Syrgkanis, Alekh Agarwal, Haipeng Luo, and Robert E Schapire. Fast convergence of regularized learning in games. *Advances in Neural Information Processing Systems*, 2015. 23, 175, 179, 291

[282] Oskari Tammelin. Solving large imperfect information games using CFR+. *arXiv:1407.5042*, 2014. 22, 217, 263

[283] Oskari Tammelin, Neil Burch, Michael Johanson, and Michael Bowling. Solving heads-up limit Texas hold'em. *International Joint Conference on Artificial Intelligence (IJCAI)*, 2015. 21, 61, 183

[284] Emanuel Tewolde, Caspar Oesterheld, Vincent Conitzer, and Paul W. Goldberg. The computational complexity of single-player imperfect-recall games. *International Joint Conference on Artificial Intelligence (IJCAI)*, 8 2023. 224

[285] Lai Tian, Kaiwen Zhou, and Anthony Man-Cho So. On the finite-time complexity and practical computation of approximate stationarity concepts of Lipschitz functions. *International Conference on Machine Learning (ICML)*, 2022. 281, 311

[286] Dipty Tripathi, Amit Biswas, Anil Kumar Tripathi, Lalit Kumar Singh, and Amrita Chaturvedi. An integrated approach of designing functionality with security for distributed cyber-physical systems. *The Journal of Supercomputing*, 78(13):14813–14845, sep 2022. 105

[287] Amparo Urbano and Jose E Vila. Computational complexity and communication: Coordination in two–player games. *Econometrica*, 70(5):1893–1927, 2002. 110

[288] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575 (7782):350–354, 2019. 184

[289] Emmanouil-Vasileios Vlatakis-Gkaragkounis, Lampros Flokas, Thanasis Lianeas, Panayotis Mertikopoulos, and Georgios Piliouras. No-regret learning and mixed nash equilibria: They do not mix. *Neural Information Processing Systems (NeurIPS)*, 2020. 218

[290] Bernhard von Stengel. Efficient computation of behavior strategies. *Games and Economic Behavior*, 14(2):220–246, 1996. 20

[291] Bernhard von Stengel and Françoise Forges. Extensive-form correlated equilibrium: Definition and computational complexity. *Mathematics of Operations Research*, 33(4):1002–1022, 2008. 6, 19, 87, 100, 138, 147, 151, 152, 154, 156, 157, 160, 163, 191, 207, 219, 220, 229, 260

[292] Bernhard von Stengel and Daphne Koller. Team-maxmin equilibria. *Games and Economic Behavior*, 21(1):309–321, 1997. 65, 67, 108, 109, 113, 121, 127

[293] Martin J Wainwright and Michael Irwin Jordan. *Graphical models, exponential families, and variational inference*. Now Publishers Inc, 2008. 97

[294] Jun-Kun Wang and Andre Wibisono. Continuized acceleration for quasar convex functions

in non-convex optimization. *International Conference on Learning Representations (ICLR)*, 2023. 280

[295] Kevin Waugh. Abstraction in large extensive games. Master's thesis, University of Alberta, 2009. 62

[296] Jibang Wu, Zixuan Zhang, Zhe Feng, Zhaoran Wang, Zhuoran Yang, Michael I Jordan, and Haifeng Xu. Sequential information design: Markov persuasion process and its efficient reinforcement learning. *ACM Conference on Economics and Computation (EC)*, 2022. 140, 174

[297] Zelai Xu, Chao Yu, Fei Fang, Yu Wang, and Yi Wu. Language agents with reinforcement learning for strategic play in the werewolf game. *CoRR*, abs/2310.18940, 2023. 105

[298] Minglu Ye. An infeasible projection type algorithm for nonmonotone variational inequalities. *Numerical Algorithms*, 89(4):1723–1742, 2022. 275, 276

[299] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative multi-agent games. *Neural Information Processing Systems (NeurIPS)*, 35:24611–24624, 2022. 332

[300] Brian Hu Zhang and Tuomas Sandholm. Sparsified linear programming for zero-sum equilibrium finding. *International Conference on Machine Learning (ICML)*, 2020. 32

[301] Brian Hu Zhang and Tuomas Sandholm. Subgame solving without common knowledge. *Neural Information Processing Systems (NeurIPS)*, 2021. 11, 42

[302] Brian Hu Zhang and Tuomas Sandholm. Polynomial-time optimal equilibria with a mediator in extensive-form games. *Neural Information Processing Systems (NeurIPS)*, 2022. 11

[303] Brian Hu Zhang and Tuomas Sandholm. Team correlated equilibria in zero-sum extensive-form games via tree decompositions. *AAAI Conference on Artificial Intelligence (AAAI)*, 2022. 11

[304] Brian Hu Zhang and Tuomas Sandholm. General-purpose search techniques without common knowledge for imperfect-information games, and application to superhuman Fog of War chess. *Under submission*, 2025. 11

[305] Brian Hu Zhang, Gabriele Farina, Andrea Celli, and Tuomas Sandholm. Optimal correlated equilibria in general-sum extensive-form games: Fixed-parameter algorithms, hardness, and two-sided column-generation. *ACM Conference on Economics and Computation (EC)*, 2022. xxiv, 11, 100, 103, 140, 182, 262

[306] Brian Hu Zhang, Gabriele Farina, and Tuomas Sandholm. Team belief DAG form: A concise representation for team-correlated game-theoretic decision making. *International Conference on Machine Learning (ICML)*, 2023. 5, 11

[307] Brian Hu Zhang, Gabriele Farina, and Tuomas Sandholm. Mediator interpretation and faster learning algorithms for linear correlated equilibria in general sequential games. *International Conference on Learning Representations (ICLR)*, 2024. 13, 263

[308] Brian Hu Zhang, Gabriele Farina, Andrea Celli, and Tuomas Sandholm. Optimal correlated equilibria in general-sum extensive-form games: Fixed-parameter algorithms, hardness,

and two-sided column-generation. *Mathematics of Operations Research*, 2025. 11

[309] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016. 332

[310] Hanrui Zhang and Vincent Conitzer. Automated dynamic mechanism design. *Neural Information Processing Systems (NeurIPS)*, 34:27785–27797, 2021. 140, 172

[311] Hanrui Zhang, Yu Cheng, and Vincent Conitzer. Automated mechanism design for classification with partial verification. *AAAI Conference on Artificial Intelligence (AAAI)*, 2021. 140, 173

[312] Hanrui Zhang, Yu Cheng, and Vincent Conitzer. Planning with participation constraints. *AAAI Conference on Artificial Intelligence (AAAI)*, 2022. 140, 175

[313] Hanrui Zhang, Yu Cheng, and Vincent Conitzer. Efficiently solving turn-taking stochastic games with extensive-form correlation. *ACM Conference on Economics and Computation (EC)*, 2023. 175

[314] Jingzhao Zhang, Hongzhou Lin, Stefanie Jegelka, Suvrit Sra, and Ali Jadbabaie. Complexity of finding stationary points of nonconvex nonsmooth functions. *International Conference on Machine Learning (ICML)*, 2020. 281, 311

[315] Naifeng Zhang, Stephen McAleer, and Tuomas Sandholm. Faster game solving via hyperparameter schedules. *arXiv:2404.09097*, 2024. 61

[316] Brian Hu Zhang, Gabriele Farina, Ioannis Anagnostides, Federico Cacciamani, Stephen McAleer, Andreas Haupt, Andrea Celli, Nicola Gatti, Vincent Conitzer, and Tuomas Sandholm. Computing optimal equilibria and mechanisms via learning in zero-sum extensive-form games. *Neural Information Processing Systems (NeurIPS)*, 2023. 11, 179, 182

[317] Brian Hu Zhang, Ioannis Anagnostides, Gabriele Farina, and Tuomas Sandholm. Efficient Φ-regret minimization with low-degree swap deviations in extensive-form games. *Neural Information Processing Systems (NeurIPS)*, 2024. 13, 264, 315

[318] Brian Hu Zhang, Gabriele Farina, Ioannis Anagnostides, Federico Cacciamani, Stephen Marcus McAleer, Andreas Alexander Haupt, Andrea Celli, Nicola Gatti, Vincent Conitzer, and Tuomas Sandholm. Steering no-regret learners to optimal equilibria. *ACM Conference on Economics and Computation (EC)*, 2024. xx, 13, 188, 189, 217

[319] Brian Hu Zhang, Ioannis Anagnostides, Emanuel Tewolde, Ratip Emin Berker, Gabriele Farina, Vincent Conitzer, and Tuomas Sandholm. Expected variational inequalities. *International Conference on Machine Learning (ICML)*, 2025. 13

[320] Brian Hu Zhang, Ioannis Anagnostides, Emanuel Tewolde, Ratip Emin Berker, Gabriele Farina, Vincent Conitzer, and Tuomas Sandholm. Learning and computation of Φ-equilibria at the frontier of tractability. *ACM Conference on Economics and Computation (EC)*, 2025. 13, 227

[321] Brian Hu Zhang, Tao Lin, Yiling Chen, and Tuomas Sandholm. Learning a game by paying the agents. *arXiv:2503.01976*, 2025. 13, 215

[322] Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul F. Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv:1909.08593*, 2019. 105

[323] Martin Zinkevich, Michael Bowling, Michael Johanson, and Carmelo Piccione. Regret minimization in games with incomplete information. *Neural Information Processing Systems (NeurIPS)*, 2007. 23, 24, 220