informs.

Optimal Correlated Equilibria in General-Sum Extensive-Form Games: Fixed-Parameter Algorithms, Hardness, and Two-Sided Column-Generation

Brian Hu Zhang,[a,*] Gabriele Farina,[b] Andrea Celli,[c] Tuomas Sandholm[a,d,e,f]

[a] Carnegie Mellon University, Pittsburgh, Pennsylvania 15232; [b] Massachusetts Institute of Technology, Cambridge, Massachusetts 02139; [c] Bocconi University, 20136 Milano MI, Italy; [d] Strategy Robot, Inc., Pittsburgh, Pennsylvania 15232; [e] StrategicMachine, Inc., Pittsburgh, Pennsylvania 15232; [f] Optimized Markets, Inc., Pittsburgh, Pennsylvania 15232
*Corresponding author
Contact: bhzhang@cs.cmu.edu, https://orcid.org/0000-0002-1267-0170 (BHZ); gfarina@cs.cmu.edu, https://orcid.org/0000-0002-3976-0061 (GF); andrea.celli2@unibocconi.it, https://orcid.org/0000-0002-2046-4019 (AC); sandholm@cs.cmu.edu, https://orcid.org/0000-0001-8861-9366 (TS)

Abstract. We study the problem of finding *optimal correlated equilibria* of various sorts in extensive-form games: *normal-form coarse correlated equilibrium (NFCCE)*, *extensive-form coarse correlated equilibrium (EFCCE)*, and *extensive-form correlated equilibrium (EFCE)*. We make two primary contributions. First, we introduce a new algorithm for computing optimal equilibria in all three notions. Its runtime depends exponentially only on a parameter related to the information structure of the game. We also prove a fundamental complexity gap between NFCCE and the other two concepts. Second, we propose a *two-sided column generation approach* for use when the runtime or memory usage of the previous algorithm is prohibitive. Our algorithm improves upon an earlier *one-sided* approach by means of a new decomposition of correlated strategies which allows players to reoptimize their sequence-form strategies with respect to correlation plans which were previously added to the support. Experiments show that our techniques outperform the prior state of the art for computing optimal general-sum correlated equilibria.

Keywords: equilibrium computation • extensive-form games • correlated equilibrium

1. Introduction

Recent algorithms for computing *Nash equilibria* in zero-sum imperfect-information extensive-form games have led to breakthroughs, most notably strong agents for two-player no-limit Texas hold'em poker (Brown and Sandholm [2], Moravčík et al. [28]). However, in general-sum and/or multiplayer games, computing Nash equilibria is hard even in normal-form games (Chen et al. [9]). Further, in real-world situations, the assumption of player strategies being *independent*, as posited in Nash equilibrium, might not hold true. For example, agents may share conventions, or communicate with a trusted mediator. Both of these concerns motivate the definition and computational study of notions of *correlated equilibria*.

In correlated equilibria, an outside *mediator* can recommend, but not enforce, certain actions. More precisely, the mediator first draws a strategy *profile* from a publicly-agreed distribution, and recommends to each player their chosen strategy. The players may then choose whether to accept the recommendation or to *deviate* and play an arbitrary action instead. A *normal-form correlated equilibrium* (NFCE) (Aumann [1]) is a distribution of profiles for which no player is ever incentivized to deviate. In a *normal-form coarse correlated equilibrium* (NFCCE) (Celli et al. [6], Moulin and Vial [29]), each player must choose to commit to following the recommendation *before* receiving it—if a player commits, she must play the recommended strategy; if she does not commit, she does not receive a recommendation.

Both above notions of correlated equilibria were originally defined only for normal-form games. More recently, von Stengel and Forges [37], and Farina et al. [14] defined and studied notions of correlated equilibria in extensive-form games. In an *extensive-form correlated equilibrium* (EFCE), each player receives recommendations

throughout the game at each of their decision point, and again can choose to follow or ignore the recommendation. In an *extensive-form coarse correlated equilibrium* (EFCCE), at each decision point, each player must commit to following the recommendation before seeing it. In both cases, a player that deviates no longer receives recommendations for the remainder of the game. We refer the reader to Figure 2 for a visual summary of the difference between the solution concepts.

Our focus is on computing *optimal* NFCCEs, EFCCEs, and EFCEs, which are the equilibria that maximize a given linear objective function. Computing optimal correlated equilibria, in any of these notions, is NP-hard in the size of the game tree, even in two-player games with chance nodes, or three-player games without chance nodes (von Stengel and Forges [37]). Some special cases are known to be solvable efficiently. von Stengel and Forges [37] show that in two-player games without chance moves, optimal equilibria in all three equilibrium notions can be computed in polynomial time. More recently, Farina and Sandholm [13] extend the positive result to so-called *triangle-free games*, which strictly include all two-player games with public chance actions.

The problem of computing *one* EFCE (and, therefore, one NFCCE/EFCCE) can be solved in polynomial time in the size of the game tree (Huang and von Stengel [21]) via a variation of the *Ellipsoid Against Hope* algorithm (Jiang and Leyton-Brown [23], Papadimitriou and Roughgarden [30]). 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 (Celli et al. [7], Farina et al. [16]). Using regret minimizers to play large multiplayer games has already led to superhuman practical performance in multiplayer poker (Brown and Sandholm [3]). As stated above, however, computing optimal equilibria is much harder.

Correlated equilibria have a close relationship with *adversarial team games*, that is, games where two teams compete against each other (Celli and Gatti [5], von Stengel and Koller [38]). An efficient algorithm for representing the space of correlated strategies of a team of players also gives an efficient algorithm for solving adversarial team games. Until recently, the state of the art for solving team games was to represent the space of correlated strategies of the team, as if to compute an extensive-form correlated equilibrium of that team (Farina et al. [15]). Recently, Zhang and Sandholm [39] and Zhang et al. [40] have developed new methods of solving team games based on *public states*. Their work gives a construction of the decision space of a team whose complexity is dependent on natural parameters of the game. However, their construction does not extend to general-sum correlation: for that, we are not only interested in the player reach probabilities of the terminal states, but also, among other things, in the *marginal* strategies of each individual player. This difference, as we will explain, creates a *critical separation between adversarial team games and general-sum correlation*.

1.1. Contributions and Paper Structure

This paper 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 multiplayer general-sum extensive-form games. At a high level, we distinguish between *conceptual*, *complexity-theoretic*, and *algorithmic* contributions.

- *Conceptual contributions.* At the conceptual level, we show that the problem of computing an optimal NFCCE, EFCCE, and EFCE, can be converted into the problem of computing an optimal strategy for a player in a suitably-constructed game. The equivalent game, which we call a *mediator-augmented game*, explicitly captures the decision problem that each player would face if the correlation device were an explicit player in the game, called the *mediator*. The action space of the mediator depends on the solution concept being analyzed: NFCCE, EFCCE, or EFCE.

While the mediator-augmented formalism greatly simplifies the treatment—providing what we hope will be an important conceptual framework for further analysis of these solution concepts—this game reformulation preserves the computational aspects of computing an optimal equilibrium, including their hardness aspects. Indeed, a key point regarding the mediator-augmented game is that the mediator faces *imperfect recall*. This is because the mediator cannot leak information across the players, so the mediator has to forget what it has observed about the other players when making a recommendation to a given player. Otherwise, the mediator's recommendations would not form a correlated profile at all, much less any equilibrium.

Optimizing for the strategy of an imperfect-recall player (here, the mediator) is known to be hard (Chu and Halpern [10], Koller and Megiddo [24]). To tackle the issue, in our paper we study effective extended formulations (in the mathematical programming sense, e.g., Conforti et al. [11]) of the decision space of the mediator, by removing the imperfect recall at the expense of a (worst-case exponential) increase in the number of decision points for the mediator player.

- *Complexity-theoretic contributions.* We then proceed to show how certain recent results regarding parameterized complexity of imperfect-recall decision problems can be applied to the mediator-augmented game. A critical

technical step in applying those results lies in characterizing the complexity of the *public states* of the decision problem faced by the mediator in the mediator-augmented game as a function of the original (not mediator-augmented) input game. 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.

    i. 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 2). For optimal EFCCE and optimal NFCCE, we establish bounds of $O^*((b + d − 1)^k)$ and $O^*((b + 1)^k)$, respectively.

    ii. 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 3). We show that the bound for EFCE *cannot* be improved in this manner.

    iii. In *two-player* games with *public chance actions*, our algorithm runs in polynomial time (Theorem 4) 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 [13]; we match their results and discuss the relationship between our algorithm and theirs in Section 4.4.

    iv. 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 5).

When k is a constant, our algorithms are, to our knowledge, the first efficient algorithms for the problem of computing an optimal correlated equilibrium in any of the three solution concepts. A comparison between our complexity results and those of past papers (Farina and Sandholm [13], von Stengel and Forges [37]) can be found in Table 1.

- *Algorithmic contributions.* We propose two main algorithms for computing optimal correlated equilibria in all three solution concepts.

    i. We operationalize the positive complexity results established above (Theorems 2 to 4) via Algorithm 1. It computes an optimal strategy for the mediator in the mediator-augmented game via linear programming. At its core, the algorithm is based on the idea that the imperfect-recall strategy space of the mediator is the projection of the set of flows in a suitable high-dimensional directed acyclic graph (DAG), called the *team belief DAG* (Zhang et al. [40]). To our knowledge, this characterization of the complicated polytope of feasible correlated equilibria as the projection of a simpler set of flows in a higher dimension is the first example of an extended formulation (in the mathematical programming sense, e.g., Conforti et al. [11]) for these solution concepts.

One cannot directly apply the fixed-parameter results of Zhang et al. [40], as that would result in a worse bound. Instead, the above results are proven by carefully analyzing the size of the resulting construction with the special structure of the mediator-augmented games in mind.

    ii. We propose a new practical approach to computing optimal correlated equilibria which we call *two-sided column generation* (Section 5). We start by deriving an LP formulation based on the strategy polytope of von

Table 1. Comparison of results in our paper with previously-known results about the computation of optimal correlated equilibria. "exp" and "poly" mean exponential time and polynomial time, respectively. "—" means that the analysis of that paper cannot handle that class of games. All no-chance games are public-chance (trivially), and all public-chance games are triangle-free (Farina and Sandholm [13]). Both these inclusions are strict. The correlation DAG algorithm requires the game to be timeable. The $O^*((bd)^k)$ result is for EFCE; the bound is better for EFCCE and NFCCE. Column generation has poor theoretical guarantees but can work well in practice compared with the correlation DAG, especially when the information complexity, k, is large.

	Game class			
Algorithm	No chance	Public chance	Triangle-free	Information complexity k
von Stengel and Forges [37]	poly	—	—	—
Farina and Sandholm [13]	poly	poly	poly	—
Correlation DAG [**this paper**]	poly (Theorem 4)	poly (Theorem 4)	exp	$O^*((bd)^k)$ (Theorem 2)

Stengel and Forges [37] and on the notion of *semirandomized correlation plan* introduced by Farina et al. [15] in the context of team games. In the latter of those two prior approaches, one player is chosen to play a normal-form strategy and the other plays a mixed (sequence-form) strategy. Our approach improves upon this by allowing the master LP to *select* which player is chosen to play the mixed strategy, thereby increasing the space of correlation plans that can be represented for any given support, and leading to a tighter master problem. In practice, we find that this change yields a speed improvement over the algorithm of Farina et al. [15] in almost all of the games tested, and this speed improvement can be greater than two orders of magnitude.

Our two solving techniques are complementary: where the parameter k is small, writing out the DAG is superior; where it is large, the two-sided column generation is faster and more frugal in its memory usage. Furthermore, the value of k can be easily computed, enabling an efficient choice between these two approaches. In experiments (Section 6), we demonstrate state-of-the-art practical performance compared to prior state-of-the-art techniques with at least one, and sometimes both, of our techniques. We also introduce two new benchmark games: a 2-vs-1 adversarial team game we call the *tricks game* which is the trick-taking (endgame) phase of the card game bridge, and the *ride-sharing game* in which two drivers seek to earn points by serving requests across a road network modeled as an undirected graph. In the tricks game, we demonstrate empirically that, even for small endgames with only three cards per player remaining, relaxing the game to be perfect information—as so-called *double dummy* bridge endgame solvers do (e.g., Ginsberg [19])—causes incorrect solutions and game values to be generated, demonstrating the need for imperfect-information game analysis.

2. Preliminaries

In this section, we review common notions for correlation in extensive-form games.

2.1. Extensive-Form Games

We start with the definition of (imperfect-information) extensive-form games, that is, tree-form games in which players might not observe all actions.

Definition 1. An *extensive-form game* Γ with *n players*, which we will identify with the positive integers $[n] = \{1, \ldots, n\}$ consists of the following:

1. A rooted tree of *nodes* $\mathcal{H}$, where the edges are labelled with *actions*. The root node of $\mathcal{H}$ will be denoted $\varnothing$. The set of leaves, or *terminal nodes* in $\mathcal{H}$ will be denoted $\mathcal{Z}$. The set of actions at a node $h \in \mathcal{H}$ will be denoted A_h. The child reached by following action a at node h will be denoted ha.

2. A partition $\mathcal{H}_0, \mathcal{H}_1, \ldots, \mathcal{H}_n$ of the set of nonterminal nodes, where $\mathcal{H}_i$ for $i > 0$ is the set of decision nodes of player i and nodes in $\mathcal{H}_0$ are chance nodes.

3. For each player $i \in [n]$, a partition $\mathcal{I}_i$ of $\mathcal{H}_i$ into *information sets*, also known as *infosets* for short. The set of actions at every node in a given infoset I must be the same, and we will denote it A_I.

4. For each player $i \in [n]$, a *utility vector* $\mathbf{u}_i \in \mathbb{R}^{\mathcal{Z}}$, where $u_i[z]$ is the utility that player i achieves upon reaching terminal node z.

5. For each chance node $h \in \mathcal{H}_0$, a fixed distribution $p(\cdot|h)$ over A_h. We will use $p(z)$ to denote the probability that chance plays all actions on the path from root to z.

Information sets $I \in \mathcal{I}_i$ contain all those nodes that Player i cannot distinguish among when acting at those nodes. This is further elucidated in the following example.

Example 1. As an example, consider the example game of Figure 1. 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. Being the only chance node, it follows that $\mathcal{H}_0 = \{\mathbf{a}\}$. 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. Examples of correlated equilibria for this game are given in Section 2.3.

Figure 1. (Color online) 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.
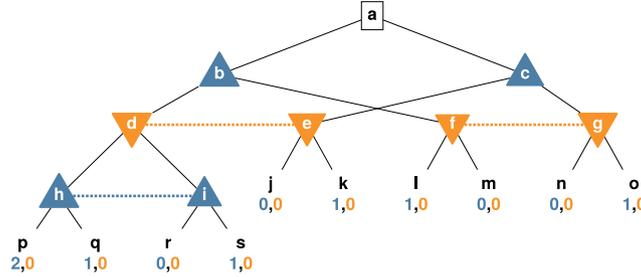
We will use $\preceq$ to denote the precedence relation induced by a tree. For example, $h \preceq h'$ if h is an ancestor of h' in the tree. If S and S' are sets of nodes, we will use $S \preceq h$ or $S \succeq h$ to mean that there exists $s \in S$ for which $s \preceq h$ or $s \succeq h$ (respectively), and $S \preceq S'$ to mean that there exist $h \in S$ and $h' \in S'$ with $h \preceq h'$. We will use $h \wedge h'$ to denote the lowest common ancestor of h and h'.

The *sequence* $\sigma_i(h)$ of player i at node h are the sequence of information sets reached and actions played by i on the root $\rightarrow h$ path, *not* including the infoset at h itself even when h is a decision node of player i. We assume that every player has *perfect recall*—that is, at every player i infoset I, every $h \in I$ has the same sequence, denoted $\sigma_i(I)$. The set of sequences of player i will be denoted $\Sigma_i := \{\sigma_i(h) : h \in \mathcal{H}\}$. The *empty sequence* of player i, $\sigma_i(\varnothing)$, will be denoted $\varnothing_i$.

In perfect-recall games, a sequence can be identified with infoset-action pair Ia. We will use this identification, and moreover, we will identify Ia with the set of nodes $Ia := \{ha : h \in I\}$. This will allow us to make use of statements such as "the nodes of Ia are one level deeper than those of I".

A *pure strategy* for a player i is an assignment of one action to each information set $I \in \mathcal{I}_i$. The *sequence form representation* of a pure strategy is the vector $x_i \in \{0, 1\}^{\Sigma_i}$, where $x_i[\sigma_i] = 1$ if player i plays every action on the path from $\varnothing_i$ to σ_i. We will use Π_i to denote the set of all sequence-form pure strategies.

For infosets or nodes v, we will use $x_i[v]$ as overloaded notation for $x_i[\sigma_i(v)]$. If $x_i[v] = 1$, we say that x_i *plays to* v. A *mixed strategy*, also denoted x_i, is a distribution over pure strategies. The sequence form of a mixed strategy is the appropriate convex combination of sequence forms of pure strategies. The set of mixed strategies of player i is denoted by $\mathcal{X}_i = \text{co}\Pi_i$. For *perfect-recall games*, $\mathcal{X}_i$ is a convex polytope characterized by a linear constraint system of size $O(|\Sigma_i|)$, containing one constraint for each information set (Koller et al. [25], Romanovskii [32], von Stengel [36]). Note that the sequence form is also well-defined, and is still a convex polytope, even for *imperfect-recall* players; however, in this setting, unless P = NP, the smallest constraint system defining the polytope may be exponential (Koller and Megiddo [24]).

A *pure profile* $x = (x_1, \ldots, x_n)$ is a collection of pure strategies, one per player. For a pure profile x, we define $x[h] := \prod_{i \in N} x_i[h] \in \{0, 1\}$ to be the indicator that all players play all actions on root $\rightarrow h$ path. The *expected utility* of player i under x is $u_i(x) := \mathbb{E}_{z \sim x} u_i[z]$, where $z \sim x$ denotes sampling a terminal node z by following the profile x, where, since x is pure, the expectation is over nature's actions. A *correlated profile* μ is a distribution over pure profiles.

The ℓth *layer* of the game tree consists of all nodes exactly distance ℓ from $\varnothing$. That is, layer 0 contains only the root, the layer 1 contains all children of the root, and so on. The *depth* of the game tree is the largest ℓ for which layer ℓ is nonempty. We will call an extensive-form game *timeable* if no infoset contains nodes in multiple layers. This is a fairly mild assumption commonly used in the extensive-form game literature (see, e.g., Jakobsen et al. [22] for a discussion) and (implicitly) universal in the reinforcement learning literature. Throughout this paper, unless otherwise stated, we consider only timeable games.

A table summarizing the notation used in this paper can be found in the appendix (Table C.2).

2.2. Correlated Equilibria in Games

Most notions of correlated equilibria in extensive-form games, including *normal-form coarse correlated equilibrium* (NFCCE), *extensive-form coarse correlated equilibrium* (EFCCE), and *extensive-form correlated equilibrium* (EFCE), can

be thought of as correlated strategies of play that can be *enforced by a mediator*. The mediator first computes and publicly announces a correlated profile μ. Then, *privately*, the mediator selects a pure profile $x \sim \mu$. Then, whenever a player i reaches an infoset I, the mediator gives a *recommendation* that i play the action played by x_i at I. The player may also choose to *deviate*, in which case they do not need to follow the recommendations of the mediator, but the mediator also no longer *gives* recommendations for the remainder of the game. The different notions of correlation are separated by what types of deviations are allowed (see also Figure 2).

• In NFCCE, a player may only deviate at the very beginning of the game. If she chooses not to deviate, she must follow all mediator recommendations for the whole game.

• In EFCCE, a player may deviate at each of her infosets *before* seeing a recommendation. However, if she chooses not to deviate, she must play the recommended action.

• In EFCE, a player may deviate at each of her infosets *after* seeing a recommendation, by instead playing a different action.

The fourth notion of equilibrium, called *normal-form correlated equilibrium* (NFCE), is often known as simply the *correlated equilibrium*. In NFCE, the mediator tells each player her entire pure strategy x_i at the start of the game, at which point the player may choose to deviate. It is known computing optimal NFCEs is NP-hard even in two-player games without chance nodes (unlike for the three notions we study in this paper) (von Stengel and Forges [37]), making it a distinctly difficult problem that is out of the scope of this paper. Thus, throughout this paper, we use *"correlated equilibrium"* to generically refer to any of the three notions of correlated equilibrium that we investigate.

2.2.1. Triggers. To formalize these notions, we use the language of *deviations* introduced by Gordon et al. [20]. Each deviation consists of a *trigger* and a *continuation strategy*, which specifies the behaviour of the player when they decide to deviate from the mediator's recommendation. The trigger determines the point of the game in which the deviating player stops following the recommendation to start playing as prescribed by the continuation strategy. Each of the solution concepts that we consider has a different set of triggers. In an NFCCE each player is allowed to deviate only at the beginning of the interaction, before any recommendation is observed. Therefore, each player i will have the empty sequence $\varnothing_i$ as their trigger. In an EFCCE triggers are the information sets of the game, while in an EFCE players may get triggered after observing a specific action recommendation at a specific information set of the game.

Definition 2. A *trigger* τ is:
• for NFCCE, the empty sequence $\varnothing_i$ for some player $i \in [n]$;
• for EFCCE, an infoset; and
• for EFCE, a sequence.

Given a solution concept $c \in \{\text{NFCCE}, \text{EFCCE}, \text{EFCE}\}$, we denote by $\mathcal{T}^c$ the set of all triggers for that concept, and $\mathcal{T}^c_i$ the set of all triggers of player i. Given a trigger $\tau \in \mathcal{T}^c$, we use $\overline{\tau}$ to denote where τ can be activated. That is, $\overline{\tau} = I$ if $\tau = Ia$ is a nonroot sequence, or else $\overline{\tau} = \tau$. We must make this distinction because EFCE triggers are activated not by reaching a part of a game tree, but by receiving a recommendation a *after* reaching a part of the game tree. We use $\Sigma^{\overline{\tau}}_i$ to denote the set of all sequences $\sigma \geq \overline{\tau}$ of player i.

A (pure) *continuation* $x'_i \in \{0,1\}^{\Sigma^{\overline{\tau}}_i}$ following a trigger τ of player i is a pure strategy defined on all infosets $I \geq \overline{\tau}$. In sequence form, x'_i is indexed by sequences $\sigma \geq \overline{\tau}$, and $x'_i[\sigma] = 1$ if the player plays all actions on the path from $\overline{\tau}$ to σ. Mixed continuation strategies are defined analogously.

2.2.2. Deviations. A pair (τ, x'_i), consisting of a trigger τ of player i and a pure continuation x'_i following $\overline{\tau}$, defines a *deviation* $\phi^{(\tau, x'_i)} : \Pi_i \to \Pi_i$ in the following manner: $\phi^{(\tau, x'_i)}(x)$ is the pure strategy that plays according to

Figure 2. Comparison of different notions of correlation in extensive-form games.
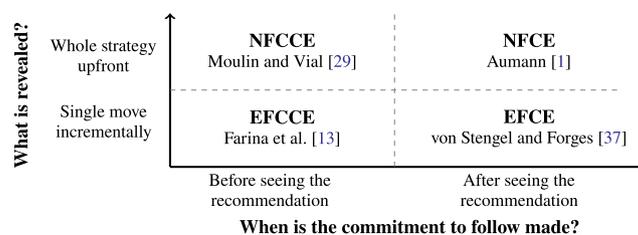
the original strategy x_i unless it prescribes τ, in which case it replaces it strategy with the continuation x_i' wherever the latter is defined. Formally,

$$\phi^{(\tau, x_i')}(x)[\sigma] := \begin{cases} x_i'[\sigma] & \text{if } I \succeq \overline{\tau} \text{ and } x_i[\tau] = 1 \\ x_i[\sigma] & \text{otherwise} \end{cases}$$

Definition 3. Given a correlated profile μ, a deviation ϕ of a player i is *profitable* if the deviating player improves its expected utility: $\mathbb{E}_{x \sim \mu} u_i(\phi(x_i), x_{-i}) > \mathbb{E}_{x \sim \mu} u_i(x)$.

Definition 4. NFCCEs, EFCCEs, and EFCEs are correlated profiles μ that have no profitable deviations of their respective types.

Here, in deciding whether to deviate, the players have common knowledge of the correlated profile μ from which their recommendations are drawn.

Given an objective function $g : Z \to \mathbb{R}$, we say that an equilibrium μ is *optimal* with respect to an objective $g : Z \to \mathbb{R}$ if μ maximizes the expected objective value $\mathbb{E}_{x \sim \mu, z \sim x} g(z)$ among all equilibria of the same notion.

Remark 1. The number of triggers available to a given player will play a fundamental role in the complexity of computing a solution according to each of the three solution concepts. In particular, for NFCCE, each player has only one trigger $(\varnothing_i)$, whereas for EFCCE and EFCE, the number of triggers for each player depends on the depth of the game. We will see in Section 4 that this difference results in a fundamental gap: under reasonable assumptions, an optimal NFCCE can be computed faster than an optimal EFCCE or an optimal EFCE.

2.3. 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 1. As described in Example 1, this game represents a signaling 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 `bcdfh`, where the letters indicate which actions were played at the respective infosets containing those nodes. For example, `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, `RLRL` is a valid pure strategy.

We make the following observations about our example game.

- The correlated profile $\mu_1 := \frac{1}{2}$`LRLRR` $+ \frac{1}{2}$`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 $\mu_2 := \frac{1}{2}$`LRLRL` $+ \frac{1}{2}$`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 `LRLRL` is an EFCE (in fact, being uncorrelated, it is a Nash equilibrium).

3. Unifying Correlated Solution Concepts via Mediator-Augmented Games

As mentioned in the previous section and summarized in Figure 2, different correlated solution concepts for extensive-form games differ in what the mediator (correlation device) reveals to the players, and whether the players' choices to commitment to follow the recommended behavior happen before or after observing the recommendation. These differences not only materialize in different equilibrium sets, but—as we will show later in this paper—also in complexity barriers that separate the solution concepts. Consequently, a unified treatment of these solution concepts needs to be approached with care.

In this section, we define augmented games in which the mediator is made explicit, which will be pivotal to our main results. Prior to presenting a precise formalization of the notion of augmented game, we provide some intuition about how the game is constructed, using the illustrative example in Figure 1. During this phase, our primary objective is to provide a straightforward intuition about the construction process, deliberately omitting certain significant details that will be formally defined in Definition 5. The augmented game explicitly represents players' choices regarding whether to adhere to mediators' recommendations or to deviate from them. Consequently, the augmented games will have different structures depending on which solution concept is desired— we will define one augmented game Γ^c for each of our target solution concepts c. Figure 3 summarizes the main connections between the computation of an optimal correlated concept c (for instance, EFCE) in the original game Γ, and the computation of a Stackelberg equilibrium in the mediator-augmented game Γ^c corresponding to c. Figure 4 depicts the augmented games derived from the example of Figure 1 for the three solution concepts of interest.

In all three augmented games, the mediator has *imperfect recall*. This is crucial to correctly capture the correlated solution concepts. The imperfect recall is necessary for the one-to-one correspondence between *mixed strategies for the mediator* in the augmented game, and *correlated profiles of the players* in the original game. Intuitively, this is because the mediator's decisions in the augmented game correspond to *recommendations* in the original game, and therefore the mediator must pick one and only one recommendation in each information set. Thus, the mediator must have one infoset in the augmented game corresponding to each infoset in the original game. If the mediator were to have perfect recall, it would have the ability to "break" information sets by sending recommendations to a player that depend on information not known to that player. Therefore, there could be a strategy for the mediator that does not correspond to a strategy profile in the original game.

NFCCE. In the case of NFCCE (Figure 4, top), the augmented game has an initial phase in which ▲ and ▼ decide whether to deviate or obey to the mediator. Only one player is allowed to deviate in the game. When player ▲ (resp., ▼) deviates, all subsequent infosets will belong to either ▲ (resp., ▼) or to the mediator. The mediator takes decisions on behalf of the obedient player. If both players are obedient (see the subtree with leaf nodes p,q,r,s, j,k,l,m,n,o), then all decisions after the initial phase are taken by the mediator.

EFCCE. In the case of EFCCE (Figure 4, middle) we can reason as follows: starting from the root of the original game Γ, we replace each information set of player ▲ or ▼ with three new infosets. The first one is a parent infoset modelling the decision of the player to obey or to deviate at the original infoset in Γ. The two children information sets encode the decision to be taken at the original infoset of Γ being replaced. The new infoset following from the decision of the player to obey (at the parent infoset) belongs to the mediator, who takes the action on behalf of the player. The new infoset following from the decision of the player to deviate (at the parent infoset) belongs to the deviating player, and it allows them for choosing the desired deviation. As before, after one player deviated, all the subsequent information sets belong to that player or to the mediator.

EFCE. In the case of EFCE (Figure 4, bottom), each of the original infosets I of Γ is duplicated and preceded by an information set of the mediator explicitly encoding the recommendation being issued at I. After observing the recommendation, the player decides whether to deviate or not. We note that actions within the same mediator's information set can represent recommendations as well as actions taken on behalf of the player. This is contingent upon whether the other player previously made a deviation or not. The information available to the mediator when recommending actions or acting on behalf of a player remains identical to what the player would have had in the original game.

Following this intuition, given a game Γ, we define the *augmented game* Γ^c corresponding to solution concept c as follows.

Figure 3. Correspondence between notions in the mediator-augmented game, and notions in the original game.

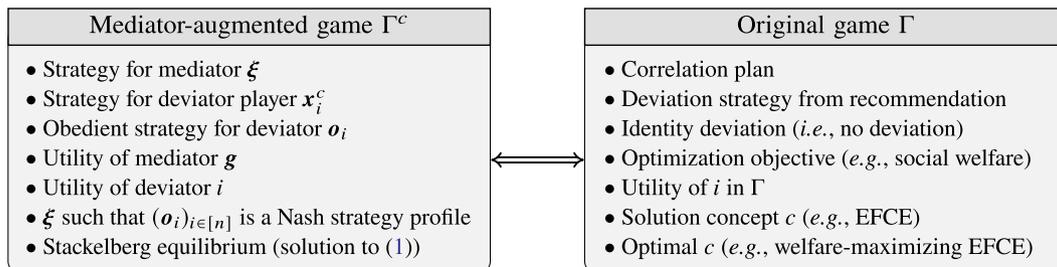| Mediator-augmented game Γ^c	Original game Γ
• Strategy for mediator $\boldsymbol{\xi}$	• Correlation plan
• Strategy for deviator player $\boldsymbol{x}_i^c$	• Deviation strategy from recommendation
• Obedient strategy for deviator $\boldsymbol{o}_i$	• Identity deviation (*i.e.*, no deviation)
• Utility of mediator $\boldsymbol{g}$	• Optimization objective (*e.g.*, social welfare)
• Utility of deviator i	• Utility of i in Γ
• $\boldsymbol{\xi}$ such that $(\boldsymbol{o}_i)_{i \in [n]}$ is a Nash strategy profile	• Solution concept c (*e.g.*, EFCE)
• Stackelberg equilibrium (solution to (1))	• Optimal c (*e.g.*, welfare-maximizing EFCE)

Figure 4. (Color online) Augmented games Γ^c for NFCCE (top), EFCCE (center), and EFCE (bottom), where Γ is the example game in Figure 1. The obedient strategies o_1, 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 trigger. If no superscript is present, there was no trigger. 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. For EFCE, τ is the node at which the player deviated, followed by the recommendation (← or →) given to the player at that node—for example, $q^{h\leftarrow}$ means terminal node q was reached but ▲ deviated after being recommended to play ← at h.
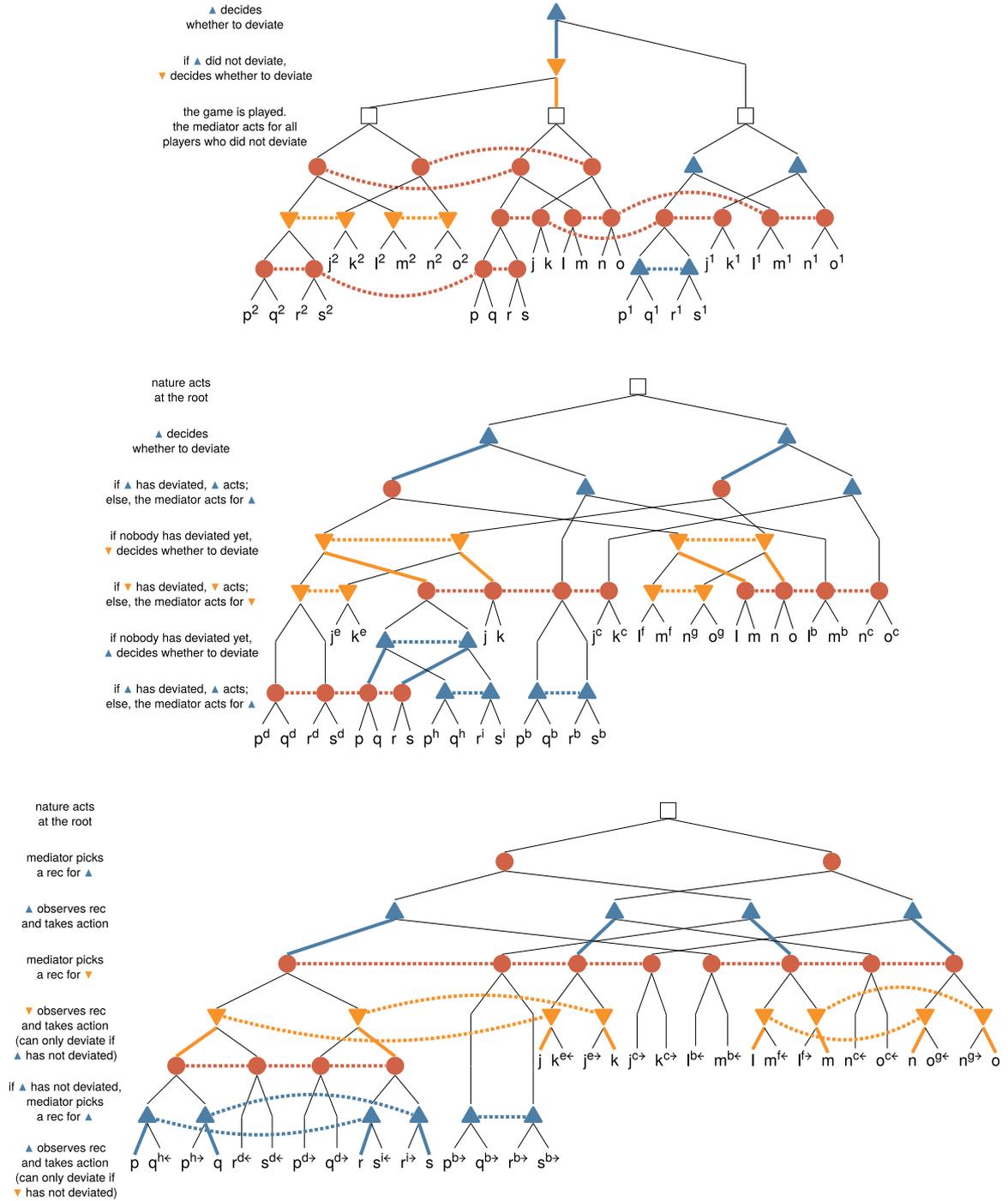
Definition 5. Given an extensive-form game Γ, a solution concept c, and an objective function $g : Z \rightarrow \mathbb{R}$, the *augmented game* Γ^c is defined as follows.
- *Players.* Γ^c has $n + 1$ players: the n players in Γ, and a *mediator*.
- *Histories.* Unless otherwise stated, histories in Γ^c are identified with tuples (h, a, τ), where:
- h is a history in Γ,
- a is either nothing ($\bot$), a special symbol $*$, or an action $a \in A_h$, and
- τ is either nothing ($\bot$) or a trigger.

Intuitively, the three components of the history represent the following.
- h is the true history of the game, representing the actions that have been taken by the players.
- a is the recommendation from the mediator at the current infoset. $\bot$ means that the mediator has yet to make a recommendation. $*$ means that the mediator has not given a recommendation.
- τ represents the trigger, if any, that has been activated. Since we need only consider deviations of one player at a time, there can be at most one active trigger—$\bot$ means there is no active trigger.
- *Terminal nodes.* If z is terminal in Γ, then $(z, \bot, \tau)$ is terminal in Γ^c for any τ. At terminal node z, each player i receives utility $u_i(z)$, and the mediator receives utility $g(z)$.
- *Chance nodes.* If h is a chance node in Γ, then $(h, \bot, \tau)$ is also a chance node in Γ^c with the same action probabilities. Chance selects an action $a \in A_h$, and the next node is $(ha, \bot, \tau)$. If $c =$ NFCCE, the next node is $(ha, \bot, \tau)$. If $c \neq$ NFCCE, the next node is a dummy node at which the sole action leads to $(ha, \bot, \tau)$.[1]
- *Information.* Players i other than the mediator have perfect recall, and their observations are specified in the game description. The mediator does *not* have perfect recall: two histories $(h_1, \cdot, \cdot)$ and $(h_2, \cdot, \cdot)$ belong to the same mediator infoset in Γ^c if and only if h_1 and h_2 belong to the same infoset in Γ.

The remainder of the construction depends on the solution concept c.

For NFCCE:
- *Preplay phase.* There is a preplay phase in which each of the n players, in order, chooses whether to deviate. Only one player may deviate: if player i deviates, then players $j > i$ are forced to not deviate. Hence this preplay phase is a tree with $n + 1$ layers and $n + 1$ leaves. The leaf in which player i deviates is $(\emptyset, \bot, \emptyset_i)$, and the leaf in which no player deviates is $(\emptyset, \bot, \bot)$.
- *Player nodes* $(h, \bot, \tau)$ (with $h \in \mathcal{H}_i$ and $i \neq 0$). Suppose $h \in \mathcal{H}_i$ with $i \neq 0$. If $\tau = \emptyset_i$ then player i observes the infoset $I \ni h$ and picks an action $a \in A_h$. Otherwise, the mediator acts by picking the action $a \in A_h$. In either case the next node is $(ha, \bot, \tau)$.
- There are no nodes (h, a, τ) with $a \neq \bot$.

For EFCCE:
- There is no preplay phase.
- *Player nodes* $(h, \bot, \tau)$ (with $h \in \mathcal{H}_i$ and $i \neq 0$). If $\tau \neq \bot$, then $(h, \bot, \tau)$ is a chance node with only one action, leading to node $(h, *, \tau)$. If $\tau = \bot$, then player i observes the infoset $I \ni h$ decides whether to deviate. If player i deviates, then the next node is $(h, *, I)$. Otherwise, the next node is $(h, *, \bot)$.
- *Player nodes* (h, a, τ) (with $h \in \mathcal{H}_i$, $i \neq 0$, and $a \neq \bot$). If τ is a trigger of player i, then player i selects an action $a \in A_h$. Otherwise, the mediator selects an action $a \in A_h$. In either case the next node is $(ha, \bot, \tau)$.

For EFCE:
- There is no preplay phase.
- *Player nodes* $(h, \bot, \tau)$ (with $h \in \mathcal{H}_i$ and $i \neq 0$). If τ is a trigger of player i, then $(h, \bot, \tau)$ is a chance node with only one action, leading to $(h, *, \tau)$. Otherwise, the mediator selects an action $a \in A_h$, and the next node is (h, a, τ).
- *Player nodes* (h, a, τ) (with $h \in \mathcal{H}_i$, $i \neq 0$, and $a \neq \bot$). If $\tau \neq \bot$ is a trigger not belonging to player i, then (h, a, τ) is a chance node with a single action, leading to $(ha, \bot, \tau)$. Otherwise, player i observes the infoset $I \ni h$ and action recommendation a, and selects an action $a' \in A_h$. If $a' = a$ or $a = *$, then the next node is $(ha', \bot, \tau)$; otherwise, the next node is $(ha', \bot, Ia)$.

For concreteness, in Figure 4 we show the augmented games derived from the example game in Figure 1 for all three solution concepts. For notational shorthand, we will use h^τ to refer to the node in Γ^c corresponding to the mediator making a recommendation with history h and trigger τ—that is, $h^\tau = (h, \bot, \tau)$ when $c =$ NFCCE or EFCE, and $h^\tau = (h, *, \tau)$ when $c =$ EFCCE.

3.1. Optimal Correlation via the Augmented Game
We now discuss how to use the augmented game Γ^c to compute optimal correlated equilibria in Γ. We first make a few critical observations:

First, the mediator has exactly one information set corresponding to each information set of the original game Γ. Therefore, *pure strategies* of the mediator correspond to *pure profiles* in Γ, and *mixed strategies* of the mediator

correspond to *correlated profiles* in Γ. We will therefore abuse notation and also use $\boldsymbol{\xi}$ to refer to mixed strategies for the mediator in Γ. Critically, the sequence form of ξ in each augmented game will have enough information about the correlated distribution to define the incentive constraints of the players. Second, each player has a unique *obedient strategy* $\boldsymbol{o}_i$, defined by always obeying recommendations (for EFCE) and never choosing to deviate (or NFCCE and EFCCE). Finally, the size of the Γ^c is polynomial in the size of Γ.

As a notational convention, where context is insufficient, we will generally use a superscript c to distinguish the augmented game from the original game—for example, $\mathcal{X}_i^c$ will denote the strategy set of player i in Γ^c, *etc.*

Now let $\boldsymbol{\xi}$ be a mediator mixed strategy in Γ^c. Then $\boldsymbol{\xi}$ represents an equilibrium in Γ if and only if, in the profile $(\boldsymbol{\xi}, \boldsymbol{o}_1^c, \ldots, \boldsymbol{o}_n^c)$, each (nonmediator) player i is playing a best response. That is, solving the following program will give an optimal equilibrium:

$$\max_{\boldsymbol{\xi} \in \Xi^c} \quad g(\boldsymbol{\xi}) \text{ s.t. } \max_{x_i^c \in \mathcal{X}_i^c} u_i(\boldsymbol{\xi}, \boldsymbol{x}_i^c, \boldsymbol{o}_{-i}^c) \leq u_i(\boldsymbol{\xi}, \boldsymbol{o}_i^c, \boldsymbol{o}_{-i}^c) \quad \forall i \in [n]$$

where Ξ^c is the mediator's sequence-form mixed strategy set in Γ^c, and $\mathcal{X}_i^c$ is player i's mixed strategy set in Γ^c. Now, by representing the mixed strategy of each player (including the mediator) in sequence form, the utility functions are linear in each strategy. Therefore, the above program can be rewritten as

$$\max_{\boldsymbol{\xi} \in \Xi^c} \quad \boldsymbol{g}^\top \boldsymbol{\xi} \text{ s.t. } \max_{x_i^c \in \mathcal{X}_i^c} \boldsymbol{\xi}^\top A_i x_i^c \leq \boldsymbol{b}_i^\top \boldsymbol{\xi} \quad \forall i \in [n] \tag{1}$$

for vectors and matrices $\boldsymbol{g}, A_i,$ and $\boldsymbol{b}_i$. Now, the inner maximization

$$\max_{x_i^c \in \mathcal{X}_i^c} \boldsymbol{\xi}^\top A_i x_i^c \tag{2}$$

is itself an LP where $\boldsymbol{\xi}$ is a constant. Moreover, since each player i has perfect recall, the sequence-form strategy sets $\mathcal{X}_i^c$ can be represented as polytopes $\mathcal{X}_i^c = \{x_i^c \geq \boldsymbol{0} : F_i^c x_i^c = f_i^c\}$ for matrix and vector F_i^c, f_i^c of size linear in the size of Γ^c. We therefore can formally take a dual of (2), resulting in the LP

$$\min_{v_i} (f_i^c)^\top v_i \text{ s.t. } A_i^\top \boldsymbol{\xi} \leq (F_i^c)^\top v_i. \tag{3}$$

By strong duality of linear programs (which holds in this case because (2) is always feasible), the programs (2) and (3) have the same value. Therefore, (1) is equivalent to the linear program

$$\begin{cases} \max_{\boldsymbol{\xi}, v_i : i \in [n]} & \boldsymbol{g}^\top \boldsymbol{\xi} \\ \text{s.t.} & A_i^\top \boldsymbol{\xi} \leq (F_i^c)^\top v_i & \forall i \in [n] \\ & (f_i^c)^\top v_i \leq \boldsymbol{b}_i^\top \boldsymbol{\xi} & \forall i \in [n] \\ \circledast & \boldsymbol{\xi} \in \Xi^c \end{cases} \tag{LP}$$

This program has size linear in the size of Γ^c and the description of the polytope Ξ^c. Unfortunately, in general, since the mediator has imperfect recall, there is no efficient way of representing Ξ^c, that is, there is no polynomial system of linear constraints describing Ξ^c. Indeed computing optimal equilibria for all three notions c is NP-hard (von Stengel and Forges [37]).

Although the *pure strategy sets* for the mediator are essentially the same in all three augmented games, the *sequence-form strategy sets* Ξ^c are substantially different. The differences arise due to more deviations being possible for some notions than for others. Consider for example the game Γ depicted in Figure 1. In the augmented game Γ^{EFCE} (Figure 4, bottom), there is a terminal node $\mathsf{p}^{\mathsf{b}\to}$ whose player reach probability $\xi[\mathsf{p}^{\mathsf{b}\to}]$ is the probability that the mediator recommends ▲ to play right at **b** and ▼ to play left at infoset **de**. There is no node in Γ^{NFCCE} whose player reach probability represents the same thing. It should therefore remain intuitively plausible that Ξ^{EFCE} should be more difficult to represent than Ξ^{NFCCE}. In the next section, we will discover that this is precisely the case.

3.2. Comparison with 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 [37]. In this section, we will review the notion of correlation plan defined by that paper, and compare it to our construction.

Definition 6. A sequence tuple $(I_1a_1, \ldots, I_na_n) \in \Sigma_1 \times \cdots \times \Sigma_n$ is *relevant* if there is a history h in Γ such that either $\sigma_i(h) = I_ia_i$ for every player i, or there is a player j—the *deviator*—such that $\sigma_i(h) = I_ia_i$ for all $i \neq j$ and $I_j \preceq h$.

This definition was first proposed by von Stengel and Forges [37] 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 7 von Stengel and Forges [37]. For a correlated profile $\mu \in \Delta(X_1 \times \cdots \times X_n)$, the *correlation plan* is the vector $\xi \in \mathbb{R}^\Sigma$ defined by $\xi[\sigma_1, \ldots, \sigma_n] = \mathbb{E}_{x \sim \mu} \prod_{i \in [n]} x_i[\sigma_i]$. We denote by Ξ the set of all correlation plans.

von Stengel and Forges [37] 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. [14, 17] generalizes this observation to NFCCE and EFCCE as well. Our linear program (LP) 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 subvector of Ξ with the indices renamed. That is, there is a natural injection from sequences of the mediator in Γ^c to relevant tuples $(\sigma_1, \ldots, \sigma_n) \in \Sigma$. A mediator sequence in Γ^c corresponds to some history h^τ. If $\tau = \bot$ then h^τ corresponds to $(\sigma_1(h), \ldots, \sigma_n(h))$, that is, $\xi[h^\tau] = \xi[\sigma_1(h), \ldots, \sigma_n(h)]$; if τ is a nonempty trigger (say, P1 WLOG), then h^τ corresponds to $(\sigma_1(\tau), \sigma_2(h), \ldots, \sigma_n(h))$, 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.

The key barrier to computing optimal equilibria, in a sense, is that *the mediator in the augmented game has imperfect recall*. In the next two sections, we will describe two methods of overcoming this imperfect recall and thus of arriving at algorithms for computing optimal equilibria. The first (Section 4) applies the recent construction of Zhang et al. [40], which is a general method of representing the sequence form of an imperfect-recall player in a timeable game. The second (Section 5) is a variant of *column generation* which is most powerful in two-player games, in which one (and only one) player is allowed to play a *mixed* strategy, thereby allowing a much greater strategy set to be available for any given support.

4. Representing Imperfect-Recall Decision Spaces

Zhang et al. [40] recently developed a method for representing the sequence-form strategy spaces for imperfect-recall players (equivalently, teams of players who cannot communicate) in timeable games. Since the augmented games Γ^c are timeable, we directly apply their main result to our problem.

Definition 8. In a timeable extensive-form game Γ', the *connectivity graph* G_S of a subset of players $S \subseteq [n]$ is the graph whose nodes are histories of Γ', and where there is an edge (h, h') if h and h' are in the same level of the tree, *and* they are *connected*, that is, there is an infoset $I \in \mathcal{I}_i$, where $i \in S$, with $h, h' \preceq I$. We will use G_i as shorthand for $G_{\{i\}}$.

Definition 9. A set of nodes $B \subseteq \mathcal{H}$ is a *belief* for 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[2] $x_i \in \Pi_i$ for player i such that B is a connected component of $G_i[\{h \in \mathcal{H} : x_i[h] = 1\}]$ where $G_i[\cdot]$ denotes an induced connected component of G_i.

We will use $\mathcal{B}_i$ to denote the set of 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 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.

Theorem 1 (Team Belief DAG; Zhang et al. [40]). *There exists a representation of player i's decision space as a polytope whose constraint matrix has $O^*(R_i)$ entries, where*

$$R_i := \sum_{B \in \mathcal{B}_i} \prod_{\substack{I \in \mathcal{I}_i: \\ I \cap B \neq \emptyset}} |A_I| \tag{4}$$

The representation uses a DAG to model the decision problem faced by player i, and then bounds the number of nodes in the DAG. For intuition, when Γ' has perfect recall, one can check that beliefs are always disjoint and

every infoset $I \in \mathcal{I}_i$ is a belief, so the above expression is linear in the size of the game—indeed, in that case, the representation reduces to the sequence-form polytope. In the interest of self-containment, Appendix A contains a description of the construction of Zhang et al. [40].

We use the above result to construct a representation of the mediator's decision space, Ξ^c, in the augmented game $\Gamma' := \Gamma^c$. We call the representation of Ξ^c using Theorem 1 the *correlation DAG* for notion c. Theorem 1 immediately gives an algorithm for solving the program (LP). This algorithm is given in Algorithm 1.

Algorithm 1 (Optimal Correlated Equilibria via Correlation DAG)
1 **input:** extensive-form game Γ, desired solution concept c, objective $g : Z \to \mathbb{R}$;
2 construct the augmented game Γ^c;
3 compute a polytope representation of the mediator's strategy space, Ξ^c, using Theorem 1;
4 solve the LP (LP);
5 **return** $\boldsymbol{\xi}$;

4.1. Analyzing the Size of the Representation

To analyze the complexity of Algorithm 1, it suffices to bound the quantity in (4). Notationally, we will use $R_\mathbf{M}^c$ to denote the quantity $R_\mathbf{M}$ in (4) in the augmented game Γ^c. We first introduce some useful definitions.

Definition 10. A *public state* is a connected component of $G = G_{[n]}$. We will use $\mathcal{P}$ to denote the set of public states.

Definition 11. 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 12. 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 1 is 3: the public state **de** has three last infosets, namely **b**, **c**, and **de** itself.

Zhang et al. [40] use the definition of information complexity to bound the representation size of Theorem 1. In particular, they show that if the decision problem for the imperfect-recall player i can be decomposed into n perfect-recall players such that the information complexity is k, then $R_i \le O^*((b+1)^k)$, where b is the branching factor of the game. In this section, we show similar bounds in our setting. Note that b and k here are the branching factor and information complexity of the original game Γ, not of Γ^c—therefore, we cannot directly apply the bound $R_i \le O^*((b+1)^k)$. 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 2. *Let k be the information complexity of a timeable game Γ, b be its branching factor, and d be its depth. Then $R_\mathbf{M}^{\mathrm{NFCCE}} \le O^*((b+1)^k)$, $R_\mathbf{M}^{\mathrm{EFCCE}} \le O^*((b+d-1)^k)$, and $R_\mathbf{M}^{\mathrm{EFCE}} \le O^*((bd)^k)$.*

Proof. The expression (4) counts the number of pairs $(B, \boldsymbol{a})$ where B is a belief of the mediator Γ^c and $\boldsymbol{a} \in \prod_{I \in \mathcal{I}_\mathbf{M}, I \cap B \ne \emptyset} A_I$. 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, $R_\mathbf{M}^{\mathrm{NFCCE}} = O^*((b+1)^k)$.

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 $R_\mathbf{M}^{\mathrm{EFCCE}} = O^*((b+d-1)^k)$.

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, $R_\mathbf{M}^{\mathrm{EFCE}} = O^*((bd)^k)$. $\square$

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 postroot actions induces a public state with T private states for each of the n players. Thus, Theorem 2 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 2. They both depend on certain information being *public*.

4.2. Public Player Actions

First, we discuss the setting in which *player* actions are public.

Definition 13. A game has *public player actions* if, for all public states $P \in \mathcal{P}$ containing at least one nonchance node, for all actions $a \in \cup_{h \in P} A_h$, the set $\{ha : h \in P, a \in A_h\}$ 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 3. *In games with public player actions, $R_{\mathbf{M}}^{\mathrm{NFCCE}} = O^*(3^k)$ and $R_{\mathbf{M}}^{\mathrm{EFCCE}} = O^*(d^k)$.*

Intuitively, the proof works by constructing a new game that reduces the branching factor of the original game to 2 while keeping all other relevant structure intact. The fact that the players' actions are public ensures that this transformation does not increase k. We defer the full proof to Appendix B.1, since it is similar to the proof in the team setting given by Zhang et al. [40].

Once again, the bound for NFCCE matches that of Zhang et al. [40] in team games, up to polynomial factors. The bound on $R_{\mathbf{M}}^{\mathrm{EFCCE}}$ 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 3 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.

4.3. 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 (von Stengel and Forges [37]), we restrict our attention to *two-player* games. Farina and Sandholm [13] 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 14. 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 (Carminati et al. [4]). Most practical games, including the games we use in our experiments, already satisfy this assumption without further modification.

Theorem 4. *In two-player timeable games with public chance actions, we have $R_{\mathbf{M}}^c = \mathrm{poly}(|\mathcal{H}|)$ for all three notions c.*

Initially, one may ask whether it is possible to prove this result by directly applying Theorem 2. In particular, if it were the case that all two-player games of public chance had constant information complexity, Theorem 4 would follow immediately. Unfortunately, this is not the case: in Figure 5, 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.

The rest of this subsection is devoted to proving this result, so, for the rest of this subsection, 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 1. *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 ▲;*
- *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 $\tilde{\sigma}_{\blacktriangledown}(h^\tau) = \sigma$.*

Figure 5. (Color online) 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 4 is more involved than one may initially expect.
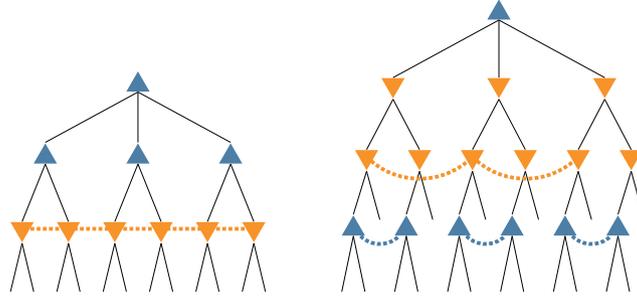
Further, the map $B \longmapsto (I, \sigma)$ is injective.

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 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 $G_\mathbf{M}^c[B]$. 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 ▲-node, because then the mediator cannot recommend ▲ to play to both h_1 and h_n. It must therefore be a ▼-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 ▼. Therefore, the mediator must recommend ▼ to play to h_1. The same applies to $h_n^{\tau_n}$. But this is a contradiction, because it implies that ▼ 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 $\tilde{\sigma}_▼(h_1^{\tau_1}) \preceq \tilde{\sigma}_▼(h_2^{\tau_2})$ or $\tilde{\sigma}_▼(h_1^{\tau_1}) \geq \tilde{\sigma}_▼(h_2^{\tau_2})$. Consider the node $h := h_1 \wedge h_2$. Since $h_1, h_2 \in I$ and I belongs to ▲, h must be a ▼-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 ▼ to play to h, or recommends an action at h that is neither a_1 nor a_2. Then $\tau_1 = \tau_2 = \sigma_▼(h_1^{\tau_1}) = \sigma_▼(h_1^{\tau_2})$.

• At h, the mediator recommends one of a_1 or a_2 (WLOG, a_1). Then $\tilde{\sigma}_▼(h_2^{\tau_2}) = I(h)a_1 \preceq \tilde{\sigma}_▼(h_1^{\tau_1})$.

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 $\phi : \mathcal{B}_\mathbf{M}^c \to (\mathcal{I}_1 \times \Sigma_2) \sqcup (\mathcal{I}_2 \times \Sigma_1)$ 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 $\sigma_▼(h^\tau) = \sigma$, and pick an h^τ so that $\tau = \bot$ if one exists. First, suppose $\tau = \bot$. There is only one way to reach $h^\bot$: 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 $h^\bot$ must be unique.

Figure 6. (Color online) Visualization of the proof of Lemma 1 (other nodes, such as the ancestors of h_2, are not shown). Since both ▼ infosets are used to make connections in $G^c[B]$, they must both be played to by ▼, which is impossible since this would require ▼ to make two different moves at the top node.
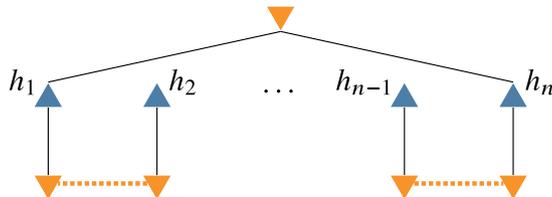
If $B \cap I$ contains no trigger-less node, then it contains only nodes with ▼-triggers. But then $B \subseteq I$, because no node with a ▼-trigger can ever be connected to a node with a ▲-trigger, and every node in $B \setminus I$ must have a ▲-trigger because of Lemma 1. But this precisely fixes what 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 1 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 $R_{\mathbf{M}}^c$ is polynomial in the game size. This completes the proof of Theorem 4.

4.4. Discussion: Relationship to Triangle-Freeness

Theorem 4 implies that Algorithm 1 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 [13] has exhibited one using a different technique, namely by showing that the *von Stengel–Forges* (vSF) polytope (von Stengel and Forges [37]) is tight. It is instructive to compare the two approaches. The approach of Farina and Sandholm [13] 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 [13].

Unlike their approach, our correlation DAG algorithm provides a parameterized 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 15. 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 $h \geq I_1$ and $h \geq I_2$. 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 [13] for more details. It is not difficult to construct triangle-free games in which our construction would be exponentially-sized; see Figure 7. 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.

4.5. 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 $c = \mathrm{EFCCE}$ or $c = \mathrm{EFCE}$ 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 16. 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 $f(k) \operatorname{poly}(|x|)$, where $|x|$ is the bit length of x and $f : \mathbb{N} \to \mathbb{N}$ is an *arbitrary function*.

Figure 7. (Color online) 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 [13] works by essentially "reordering" 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.
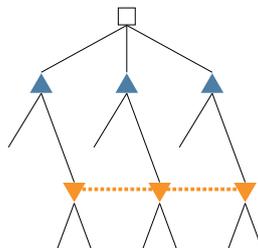
The *k*-CLIQUE problem[3] is widely conjectured to not admit an FPT algorithm parameterized by the clique size *k*. In the literature on parameterized complexity, this conjecture is known as $FPT \neq W$ (Aumann [1]), and is implied by the exponential time hypothesis (Chen et al. [8]). We now show that this conjecture implies lower bounds on the complexity of representing the polytopes Ξ^{EFCCE} and Ξ^{EFCE}.

Theorem 5. *Assuming FPT $\neq$ W (Aumann [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 8):
- 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*. 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.[4] 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 $\boldsymbol{\xi} \in \Xi^{\mathrm{EFCCE}}$, 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 $\boldsymbol{\xi} \in \Xi^{\mathrm{EFCE}}$, 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 $\boldsymbol{m}[j_1, v_1, j_2, v_2] = \mathbf{1}\{v_{1,j_1} = v_1 \text{ and } v_{2,j_2} = v_2\}$. Now consider the objective function $g : M \to \mathbb{R}$ defined by

$$g(\boldsymbol{m}) = \mathop{\mathbb{E}}_{\substack{j_1, j_2 \in [k] \\ v_1, v_2 \in [n]}} \begin{cases} \boldsymbol{m}[j_1, v_1, j_2, v_2] & \text{if } j_1 = j_2 \text{ and } v_1 = v_2 \leq n; \text{ or } j_1 \neq j_2 \text{ and } (v_1, v_2) \in E \\ 0 & \text{otherwise} \end{cases}$$

Figure 8. (Color online) The game used in the proof of Theorem 5, for $n = k = 2$.
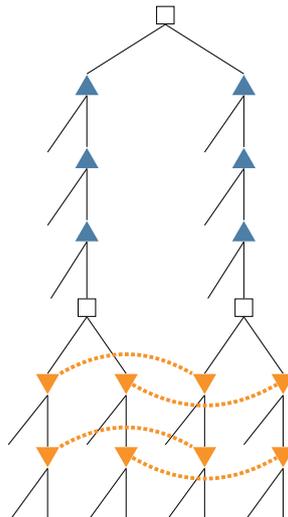
where the expectation is over a uniformly random sample. We now claim that $\max_{m \in M} g(m) = 1$ if and only if G has a clique of size k, which will complete the proof.

($\Leftarrow$) If G has a k-clique $\{v_1^*, \ldots, v_k^*\}$, then we set $v_{i,j} = v_j^*$ for both players $i \in \{1,2\}$, and indeed this achieves $g(m) = 1$ by construction.

($\Rightarrow$) If $g(m) = 1$, then for all j we must have $\sum_{v \in [n]} m[j, v, j, v] = 1$, that is, $v_{1,j} = v_{2,j}$. But then $\{v_{1,j}, \ldots, v_{1,k}\}$ must be a clique by construction, because otherwise there would be some $j_1 \neq j_2$ for which $m[j_1, v_{1,j}, j_2, v_{1,j}] = 0$. $\square$

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 (LP). Indeed, the proof of Theorem 5 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 (LP). However, we know of no technique for optimal equilibria that would not also imply the ability to optimize over Ξ^c. Therefore, Theorem 5 is a lower bound that applies to all known techniques for computing optimal EFCCEs and EFCEs.

5. Two-Sided Column Generation Approach

The approach in the previous section overcomes the imperfect recall of the mediator player in the augmented game by using an extended formulation, in which the imperfect recall is eliminated at the cost of a (controlled) exponential increase in the size of the decision problem. As we have seen, this process begets new parameterized complexity results for the problem of computing optimal correlated solution concepts in extensive-form games, as well as, in many cases, the current state-of-the-art algorithm for computing optimal correlated solution concepts in EFGs. Yet, there are some cases in which the exponential reformulation is prohibitive. In this section, we propose an alternative approach based on column generation (with some domain-specific tweaks and improvements) that might be helpful in such scenarios. To reduce the burden on notation, our presentation will focus on the case of two players, though in principle the method can be extended to multiple players without significant obstacles.

5.1. Semi-Randomized Correlation Plans

We introduce the following notation for two-player games, which was also used by Farina et al. [15]: it is worth re-emphasizing that sequence pair $(I_1 a_1, I_2 a_2) \in \Sigma_1 \times \Sigma_2$ is *relevant* if there is a history h with $h \succeq I_1$ and $h \succeq I_2$ (see also Definition 6). We write $\sigma_1 \bowtie \sigma_2$ to denote a relevant sequence pair, and $\Sigma \subseteq \Sigma_1 \times \Sigma_2$ for the set of all relevant sequence pairs. For a sequence σ of Player 1 and an infoset I of Player 2, we write $\sigma \bowtie I$ if $\sigma \bowtie Ia$ for each action a. Similarly, for a Player 1 infoset I and Player 2 sequence σ, we write $I \bowtie \sigma$ if $Ia \bowtie \sigma$. In the subsequent discussion, for ease of notation, the symbol Ξ will be used to denote Ξ^{EFCE}, as it encompasses both the EFCCE and NFCCE cases.

Now, we introduce the strategy representation which we employ in our algorithm. We observe that variables in LP (LP) belong to the convex polytope Ξ, but that polytope cannot be compactly represented in general. Therefore, we tackle LP (LP) by adopting the notion of *semirandomized correlation plan* proposed by Farina et al. [15]. For completeness, we show how *semirandomized correlation plan* can be derived from the *von Stengel-Forges polytope* (von Stengel and Forges [37]) representing interlaced sequence-form "probability mass conservation" constraints for the two players.

Definition 17. The *von Stengel-Forges polytope*, denoted $\mathcal{V}$, is the polytope of all vectors $\boldsymbol{\xi} \in \mathbb{R}_{\geq 0}^{\Sigma}$ (i.e., indexed over relevant sequence pairs) such that:

$$\text{Ⓐ} \quad \boldsymbol{\xi}[\varnothing_1, \varnothing_2] = 1$$

$$\text{Ⓑ} \quad \sum_{a \in A_I} \boldsymbol{\xi}[Ia, \sigma_2] = \boldsymbol{\xi}[\sigma_1(I), \sigma_2] \quad \forall I \bowtie \sigma_2 \in \mathcal{I}_1 \times \Sigma_2, \text{ and}$$

$$\text{©} \quad \sum_{a \in A_I} \boldsymbol{\xi}[\sigma_1, Ia] = \boldsymbol{\xi}[\sigma_1, \sigma_2(I)] \quad \forall I \bowtie \sigma_2 \in \mathcal{I}_2 \times \Sigma_1.$$

The set of linear constraints defining $\mathcal{V}$ is polynomially-sized. Moreover, the set of correlation plans is a subset of the von Stengel-Forges polytope, that is, $\Xi \subseteq \mathcal{V}$ (von Stengel and Forges [37]).

The notation in this section is different from that in the preceding section: $\boldsymbol{\xi}$ here is indexed by *relevant sequence pairs*, whereas $\boldsymbol{\xi}$ in Section 3 is indexed by *sequences for the mediator in the augmented game*. However, the two notations essentially describe the same thing: given a history h^τ of any of the augmented games, we identify the

realization-form player reach probability $\xi[h^\tau]$ with the term $\xi[\sigma_1(h^\tau), \sigma_2(h^\tau)]$, where $\sigma_i(h^\tau) = \sigma_i(\tau)$ if τ is a trigger of player i, and otherwise $\sigma_i(h)$ (that is, $\sigma_i(h^\tau)$ is the sequence of recommendations that the mediator has given to player i on the path to node h^τ in the augmented game.)

Finally, a *semirandomized correlation plan* is composed of a deterministic sequence form strategy for one player, while the other player independently plays a mixed strategy.[5]

Definition 18 (Farina et al. [15]). The sets of semirandomized correlation plans are

$$\Xi_1^{\mathrm{sr}} := \{\xi \in \mathcal{V} : \xi[\varnothing, \sigma_2] \in \{0, 1\} \quad \forall \sigma_2 \in \Sigma_2\} \quad \text{and} \quad \Xi_2^{\mathrm{sr}} := \{\xi \in \mathcal{V} : \xi[\sigma_1, \varnothing] \in \{0, 1\} \quad \forall \sigma_1 \in \Sigma_1\}.$$

Given $i \in \{1, 2\}$, a point $\xi \in \Xi_i^{\mathrm{sr}}$ can be expressed using real and binary variables, in addition to the linear constraints defining the von Stengel-Forges polytope $\mathcal{V}$. In particular, we rely on the observation by Farina et al. [15] that $\Xi = \mathrm{co}(\Xi_1^{\mathrm{sr}}) = \mathrm{co}(\Xi_2^{\mathrm{sr}}) = \mathrm{co}(\Xi_1^{\mathrm{sr}} \cup \Xi_2^{\mathrm{sr}})$.

5.2. Correlation-Plan Decomposition and Iterative Framework

We say that a correlation plan ξ is a *product correlation plan* if, for any $(\sigma_1, \sigma_2) \in \Sigma$, $\xi[\sigma_1, \sigma_2] = \xi[\sigma_1, \varnothing] \cdot \xi[\varnothing, \sigma_2]$. Since any semirandomized correlation plan corresponds to a distribution of play where one player plays a pure sequence-form strategy, while the other plays a mixed sequence-form strategy, $\xi \in \Xi_i^{\mathrm{sr}}$ is guaranteed to be a product correlation plan for any i (see Farina et al. [15, lemma 3]).

Given $\xi \in \Xi_1^{\mathrm{sr}} \cup \Xi_2^{\mathrm{sr}}$, let $m_1 \in \mathcal{X}_1$ be the *marginal vector* such that $m_1[\cdot] := \xi[\cdot, \varnothing]$, and let m_2 be defined analogously. Then, we can decompose any correlation plan $\xi \in \Xi_1^{\mathrm{sr}} \cup \Xi_2^{\mathrm{sr}}$ as

$$\xi = \lambda x_1 m_2^\top + (1 - \lambda) m_1 x_2^\top,$$

for some appropriate choice of $\lambda \in \{0, 1\}$, and mixed strategies x_1, x_2 for Player 1 and Player 2, respectively.[6] Moreover, given $\xi \in \Xi_1^{\mathrm{sr}} \cup \Xi_2^{\mathrm{sr}}$, we have $\lambda x_1 m_2^\top + (1 - \lambda) m_1 x_2^\top \in \mathrm{co}(\Xi_1^{\mathrm{sr}} \cup \Xi_2^{\mathrm{sr}})$ for any $\lambda \geq 0$ and well-formed sequence-form strategies x_1, x_2.

Our column-generation algorithm will solve a sequence of linear programs. Each program refines the previous by allowing the correlation plan variable ξ to be expressed as a convex combination of points from a growing *support set* S. In our case, the support generated after any T iterations is in the form $S = \{m_1^{(1)}, m_2^{(1)}, m_1^{(2)}, m_2^{(2)}, \ldots, m_1^{(T)}, m_2^{(T)}\}$, where $m_1^{(t)}$ and $m_2^{(t)}$ are marginal strategies for P1 and P2, respectively. With this support, we will allow the mediator to select any *mixture of semirandomized correlation plans*, where *at least one* of the players is playing a strategy in S. Formally, the mediator is allowed to select weights $\lambda_i^{(t)}$, for $i \in \{1, 2\}$ and $t \in [T]$, such that $\sum_{i=1}^2 \sum_{t=1}^T \lambda_i^{(t)} = 1$ and $\lambda_i^{(t)} \geq 0$, as well as the other player's strategy $x_i^{(t)} \in \mathcal{X}_i$, resulting in the correlation plan

$$\xi := \sum_{t=1}^T \lambda_1^{(t)}((m_1^{(t)})(x_2^{(t)})^\top + \lambda_2^{(t)}(x_1^{(t)})(m_2^{(t)})^\top).$$

We will denote by Ξ^S the set of all correlation plans admissible in the above sense for a given support S.

Compared with Farina et al. [15], this notion of mixing allows more correlation plans to be formed for any given support size T: that paper *fixes* upfront which player is allowed to play a mixed strategy and which player plays the pure strategy, whereas ours allows the master problem to decide this. We say that the column-generation algorithm of Farina et al. [15] is *one-sided* since one player has to select a pure strategy, while the other picks a sequence-form strategy after observing the pure strategy. In contrast, we call our framework *two-sided*, each player can have both roles, and the parameter λ dictates who has which role. As such, the correlation-plan decomposition which we introduced allows us to exploit correlation plans already in the support S in a more powerful way than what is possible in other *one-sided* column-generation approaches like the one by Farina et al. [15]. In particular, we remark that $\tilde{x}_1^{(t)}, \tilde{x}_2^{(t)}$ are continuous variables in LP (M). Therefore, each player is allowed to reoptimize their mixed strategies, enabling them to reach a richer set of correlation plans starting from the same support set. As a result, for any T, our master problem will be substantially tighter, leading to faster convergence.

Given an equilibrium concept c, our master problem at time T is the following linear program:

$$
\begin{cases}
\max & g^\top \xi \\
\text{s.t.} & A_i^\top \xi \leq (F_i^c)^\top v_i & \forall i \in \{1,2\} \\
& (f_i^c)^\top v_i \leq b_i^\top \xi & \forall i \in \{1,2\} \\
& \xi = \sum_{t=1}^T ((m_1^{(t)})(\tilde{x}_2^{(t)})^\top + (\tilde{x}_1^{(t)})(m_2^{(t)})^\top) \\
& F_i \tilde{x}_i^{(t)} = \lambda_i^{(t)} f_i, \quad \tilde{x}_i^{(t)} \geq 0 & \forall i \in \{1,2\}, t \in [T] \\
& \sum_{i=1}^2 \sum_{t=1}^T \lambda_i^{(t)} = 1, \quad \lambda_i^{(t)} \geq 0 \quad \forall(i,t)
\end{cases} \tag{M}
$$

where $g, F_i^c, f_i^c, v_i, A_i$ are defined in (LP), and $F_i = x_i$ are the sequence-form equality constraints for player i in the original game Γ—that is, $\mathcal{X}_i := \{x_i \geq 0 : F_i x_i = f_i\}$.
  • each correlation plan ξ is represented through the decomposition which we defined above, where we perform the change of variables $\tilde{x}_i^{(t)} := \lambda_i^{(t)} x_i^{(t)}$ for each i;
  • Constraint   and   are the same constraints as in (LP).
  • Constraints   to   define ξ according to the aforementioned semirandomized mixture. They take the place of the hard constraint $\otimes$ in (LP).

If we could afford to set $S = \Pi_1$ or $S = \Pi_2$, finding an optimal NFCCE, EFCCE, EFCE for an arbitrary objective g would amount to solving LP (M) once. However, Π_1 and Π_2 are usually exponentially large. Therefore, we follow the approach by Ford and Fulkerson [18] and generate the support S iteratively.

Algorithm 2 describes the main steps of our iterative procedure. First, we initialize the support S through a seeding phase in which S is endowed with one or more correlation plans which are known to belong to Ξ. In our experiments, we start by assigning to S the correlation plan obtained as the product of one *uniform* mixed strategy per player (i.e., a strategy such that, at each I, the player draws one action from A_I according to a uniform probability distribution). Then, at each iteration t, we solve the master LP (M) with the current support S. Each time we solve (M), we keep track of the resulting primal and dual variables. In particular, when solving (M), the algorithm keeps track of the current solution $\xi^{(T)}$ (i.e., the correlation plan corresponding to the optimal decomposition), and the dual variables for constraints   and   .

5.3. Expansion of Support: The Pricing Problem

At iteration T, the marginal strategies $m_1^{(T+1)}, m_2^{(T+1)}$ of a new correlation plan $\xi^{(T+1)}$ are added to S. The selection of $\xi^{(T+1)}$ is controlled by the function PRICER (Algorithm 2), which solves the pricing problem of finding the correlation plan that would lead to the maximum gradient of the objective (i.e., maximum *reduced cost*) if it was to be included in the convex combination computed by (M). At iteration t, such correlation plan can be computed from the solution to the dual of the master LP. We use a tuple $((\tilde{x}_i^c)^{(T)}, \gamma_i^{(T)})_{i \in [n]}$, where $(\tilde{x}_i^c)^{(T)}$ be the subvector of the dual variables corresponding to constraints   and player i, and $\gamma_i^{(T)}$ be the subvector of dual variables corresponding to constraints   and trigger τ. Then, by letting

$$
w^{(T)} := \sum_{i \in \{1,2\}} (A_i(\tilde{x}_i^c)^{(T)} - \gamma_i^{(T)} b_i),
$$

the pricing problem amounts to solving $\max_{\xi \in \Xi}(g - w^{(T)})^\top \xi$. We know that $\Xi = \mathrm{co}(\Xi_i^{\mathrm{sr}})$, for any i. Therefore, by linearity of the objective and by convexity, we have

$$
\max_{\xi \in \Xi}(g - w^{(T)})^\top \xi = \max_{\xi \in \Xi_i^{\mathrm{sr}}}(g - w^{(T)})^\top \xi.
$$

This is a well-defined mixed integer LP (MIP), which can be solved through a commercial solver such as Gurobi. We denote by δ the optimal value of the pricing problem, and by $\xi^{(t+1)}$ a correlation plan attaining such value (see Line 12).

In the initial stage of the algorithm, (M) may be infeasible as the support generated up to that point might be insufficient to generate an equilibrium. In this case, we define an LP (M') by replacing constraint   with $(f_i^c)^\top v_i \leq b_i^\top \xi + u$, where u is an introduced slack variable, and making the objective $\max(-u)$. This LP is guaranteed to be feasible.

The column generation algorithm admits a clear game-theoretic interpretation.[7] The dual variables $\tilde{x}_i^c$ correspond to an augmented-game deviator strategy $x_i^c := \tilde{x}_i^c / \gamma_i \in \mathcal{X}_i^c$, scaled by a Lagrangian multiplier $\gamma_i \geq 0$. In the initial (infeasible) phase, it solves the zero-sum game

$$\max_{\xi \in \Xi^S} \min_{\substack{i \in [n], \\ x_i^c \in \mathcal{X}_i^c}} -\xi^\top (A_i x_i^c - b_\tau) \tag{5}$$

by column generation for the maximizing player. For any given ξ, the objective value above is nothing but the maximum deviation benefit of a deviator strategy x_i^c against ξ. Therefore, since every game *has* equilibria, the equilibrium value of this game is zero, and that value is achieved when S is large enough that Ξ^S contains at least one equilibrium, at which point the algorithm moves to the second stage. In the second stage, the algorithm solves the zero-sum game

$$\max_{\xi \in \Xi^S} \min_{\substack{\gamma_i \geq 0, \\ x_i^c \in \mathcal{X}_i^c}} g^\top \xi - \sum_{i \in \{1,2\}} \gamma_i \cdot \xi^\top (A_i x_i^c - b_\tau) \tag{6}$$

again by column generation. The optimal solutions to this game are, by construction, the optimal equilibria. The feasibility of (5) guarantees that (6) has a finite equilibrium value, and moreover the equilibria of this zero-sum game are precisely the solutions to (LP), that is, the optimal equilibria with concept c. In the language of this zero-sum game, the pricer is the maximizing player (mediator)'s best response value against the current equilibrium $(\xi, \{x_i^c, \gamma_i\}_{i \in \{1,2\}}$. Thus, when equilibrium is achieved (i.e., when the pricer fails to find an improving direction), the game is solved.

Algorithm 2 (Two-Sided Column Generation)
1   **function** COMPUTEOPT (game Γ, concept $c \in \{\text{NFCCE}, \text{EFCCE}, \text{EFCE}\}$, objective g, tolerance $\epsilon \geq 0$
2   **initialization:** $T \leftarrow 1, S \leftarrow \{\text{uniformly random strategy for both players}\}$
3   **while** within computational budget **do**
4   **if** (M) is infeasible **then**
5   (M') $\leftarrow$ replace constraint   in (M) with $(f_i^c)^\top v_i \leq b_i^\top \xi + u$, where u is an introduced slack variable, and replace the objective with $\max(-u)$ [▷ see discussion in text above]
6         $\xi^{(T)}, \{(\tilde{x}_i^c)^{(T)}, \gamma_i^{(T)}\}_{i \in \{1,2\}} \leftarrow$ solve (M')   [▷ primal $\xi^{(T)}$, dual $\{(\tilde{x}_i^c)^{(T)}, \gamma_i^{(T)}\}_{i \in \{1,2\}}$]
7         $\beta, m_1^{(T+1)}, m_2^{(T+1)} \leftarrow$ PRICER($\xi^{(T)}, \mathbf{0}, \{(\tilde{x}_i^c)^{(T)}, \gamma_i^{(T)}\}_{i \in \{1,2\}})$
8         $S \leftarrow S \cup \{m_1^{(T+1)}, m_2^{(T+1)}\}$
9         $T \leftarrow T + 1$
10   **else**
11         $\xi^{(T)}, \{(\tilde{x}_i^c)^{(T)}, \gamma_i^{(T)}\}_{i \in \{1,2\}} \leftarrow$ solve (M)   [▷ primal $\xi^{(T)}$, dual $\{(\tilde{x}_i^c)^{(T)}, \gamma_i^{(T)}\}_{i \in \{1,2\}}$]
12         $\beta, m_1^{(T+1)}, m_2^{(T+1)} \leftarrow$ PRICER($\xi^{(T)}, g, \{(\tilde{x}_i^c)^{(T)}, \gamma_i^{(T)}\}_{i \in \{1,2\}})$
13   **if** $\beta \leq \epsilon$ **then return** $\xi^{(T)}$

14   **function** PRICER $\xi, g, \{\tilde{x}_i^c, \gamma_i\}_{i \in \{1,2\}}$
15     $w \leftarrow \sum_{i \in \{1,2\}} (A_i \tilde{x}_i^c - \gamma_i b_i)$
16   Select one player i
17     $\beta, \xi^* \leftarrow \max_{\bar{\xi} \in \Xi_i^{\text{sr}}} (g - w)^\top (\bar{\xi} - \xi)$
18     $m_1^*, m_2^* \leftarrow$ marginals of ξ^*
19   **return** β, m_1^*, m_2^*

6. Experiments

We ran experiments to evaluate our proposed algorithms on a suite of standard benchmark games, as well as two new benchmarks that we introduce. Each experiment was allocated 4 threads, 64 GB of RAM, and 6 hours of runtime. We used Gurobi 9.5 to solve LPs and MIPs.

6.1. Implementation Details

The correlation DAG LP sometimes causes Gurobi's presolver to produce seemingly poorly-conditioned LPs, for reasons unknown to us. We therefore run the correlation DAG twice for each experiment, once with presolver on and once with presolver off, and report only the better of the two runtimes.

In the implementation of the two-sided column-generation algorithm (Algorithm 2), before solving a pricing problem via its MIP formulation, we try to solve the linear relaxation in which $\xi \in \mathcal{V}$. If the solution to such LP is a semirandomized correlation plan we can avoid the overhead of solving a MIP. Moreover, our implementation

makes use of Gurobi's solution pools: since the MIP solver used for pricing problems is already tracking additional suboptimal feasible solutions, we add, together with the optimal one, such suboptimal correlation plans to S with no additional computational cost. This does not affect the optimality of the final solution, and was shown to improve performances in the team games domain (Farina et al. [15]).

6.2. Game Instances

We ran experiments on the following standard benchmark games. For compatibility, we use the same notation for referencing games as Zhang et al. [40].

1. $^3\text{K}r$ is 3-player *Kuhn poker* (Kuhn [26]) with r ranks.
2. $^3\text{L}brs$ is 3-player *Leduc poker* (Southey et al. [35]) with b bets per round, r ranks, and s suits.
3. ^3GL is 3-player *Goofspiel* (Ross [33]) with 3 ranks and imperfect information.
4. $^3\text{D}d$ is 3-player *Liar's Dice* (Lis' et al. [27]) with one d-sided die per player.
5. $^2\text{B}hwr$ is 2-player *Battleship* (Farina et al. [16]) on a grid of size $h \times w$, one unit-size ship per player, and r rounds.
6. $^2\text{S}nbr$ is a simplified version of the 2-player *Sheriff of Nottingham* (Farina et al. [17]) game, with n items for the smuggler, a maximum bribe of b, and r rounds of bargaining.

Detailed rules for all of these games can be found in Farina et al. [15] and Farina et al. [17]. We also introduce two new parametric families of games:

7. $^3\text{T}[L]$ is a *trick-taking game*, which emulates the trick-taking (endgame) phase of the card game *bridge* where each player only has three cards remaining. When L is given, L deals are randomly selected at the beginning of the game, and it is common knowledge that the true deal is among them.[8] ^3TP is the *perfect-information* ("double-dummy", as it is known in the bridge community) variant, which could in principle be solved by perfect-information techniques such as alpha-beta search. Nonetheless, our algorithms still run in that game, so we use them.

Bridge is one of the most well-known adversarial team games. To our knowledge, computer agents in bridge have not achieved performance comparable to top humans, making it an excellent benchmark for research. The techniques in this paper obviously will not scale to the full game of bridge, but nonetheless we can show interesting results even on small endgames.

8. $^2\text{RS}iT$ is a *ride-sharing game*. It is played on finite graph. Two *drivers* seek to earn points by reaching specific nodes of the graph and serving the requests at those nodes. Parameter i specifies the graph configuration, while T is the time horizon.

Ride sharing is of course ubiquitous in the modern day. A ride-sharing company is tasked with directing its drivers in such a way that it maximizes some objective function (say, the social welfare of all drivers). But the company has no ultimate way of enforcing behavior, only recommending it. This is exactly the scenario where correlated equilibria are the right notion. Further, to our knowledge, this game is the only benchmark in the literature in general-sum correlation in which the polytope of von Stengel and Forges [37] is not tight, and thus for which we know no polynomial-time algorithm. As such, it is a good testbed for our algorithms, which can run in all games.

Full details on our new benchmarks are given in Appendix C.

6.3. Optimal Correlation

We evaluated the performance of the DAG-based LP and the two-sided column-generation framework against the prior state-of-the art algorithms for computing optimal correlated equilibria in general-sum extensive-form games: the polytope of von Stengel and Forges [37] (denoted by **[vSF08]**), which is correct only for a certain family of games called *triangle-free games* (we denote with "n/a" when this is not the case), and the one-sided column-generation algorithm by Farina et al. [15] (denoted by **[FCGS21]**), which we adapted from the team domain.

Table 2 summarizes the comparison over two-player game instances. As expected, the correlation DAG LP has the best running times for games with small information complexity parameter k. When this is the case, it dramatically outperforms previous algorithms: it can solve in a matter of seconds instances that previously exceeded 6 hours (see, e.g., $^2\text{B}323$ and $^2\text{S}133$), and it can solve in less than 1 hour instances that previously were not computationally feasible (e.g., $^2\text{B}324$). On the other hand, when k is large (e.g., in $^2\text{RS}23$), the two-sided column-generation algorithm provides the best running times. For example, when computing optimal EFCE, it requires 6 minutes while the prior one-sided column-generation algorithm **[vSF08]** exceeds 6 hours. Combining the two techniques that we propose yields uniformly better performance than prior work for any value of the parameter k.

Table 2. Experiments on general-sum correlated equilibria, comparing both our correlation DAG LP and two-sided column generation to earlier approaches. **vSF08** is the relaxation of von Stengel and Forges [37], which is only correct in triangle-free games (Farina and Sandholm [13]). RS is not triangle-free, so vSF fails in that game. **FCGS21** is the one-sided column generation approach of Farina et al. [15]. k is the information complexity. All runs were performed to convergence. "oom" means out of memory. Runtimes are colored according to the ratio with the best runtime in that row, according to the scale [1   10   ≥ 100].
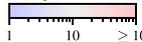
| Game			Concept	$|\mathcal{E}^c|$	Value	[vSF08]	Column generation [FCGS21]	This paper	DAG This paper
²B222	$|\mathcal{Z}|$	1,072	NFCCE	6,895	0.000	0.06s	0.98s	0.23s	0.02s
	$|\Sigma|$	11,049	EFCCE	20,909	−0.525	0.15s	1m 57s	19.61s	0.06s
	k	8	EFCE	20,547	−0.525	0.28s	36m 58s	2m 21s	0.16s
²B322	$|\mathcal{Z}|$	19,116	NFCCE	91,582	0.000	2.62s	13m 33s	11m 49s	0.13s
	$|\Sigma|$	264,541	EFCCE	331,310	−0.317	4.20s	> 6h	1h 10m	1.38s
	k	12	EFCE	503,053	−0.317	12.97s	> 6h	> 6h	5.23s
²B323	$|\mathcal{Z}|$	191,916	NFCCE	1,040,814	0.000	1m 5s	4h 26m	2h 44m	2.35s
	$|\Sigma|$	3,893,341	EFCCE	5,321,677	−0.375	2m 13s	> 6h	> 6h	35.48s
	k	12	EFCE	7,274,633	−0.375	oom	> 6h	oom	2m 1s
²B324	$|\mathcal{Z}|$	969,516	NFCCE	6,388,479	0.000	oom	oom	oom	20.01s
	$|\Sigma|$	26,443,741	EFCCE	40,732,129	−0.489	oom	oom	oom	10m 56s
	k	12	EFCE	49,299,490	−0.489	oom	oom	oom	33m 12s
²S122	$|\mathcal{Z}|$	396	NFCCE	2,861	13.636	0.04s	0.69s	0.28s	0.01s
	$|\Sigma|$	3,717	EFCCE	7,385	9.565	0.04s	11.86s	0.96s	0.02s
	k	12	EFCE	6,227	9.078	0.08s	51.10s	4.09s	0.04s
²S123	$|\mathcal{Z}|$	2,376	NFCCE	17,999	13.636	0.26s	18.17s	3.02s	0.04s
	$|\Sigma|$	33,633	EFCCE	69,539	10.000	0.59s	1h 1m	3m 11s	0.23s
	k	12	EFCE	52,559	10.000	1.24s	48m 55s	7m 5s	0.53s
²S133	$|\mathcal{Z}|$	5,632	NFCCE	43,939	18.182	1.07s	3m 22s	8.82s	0.05s
	$|\Sigma|$	95,768	EFCCE	165,491	15.000	2.10s	> 6h	1h 28m	0.67s
	k	12	EFCE	165,859	15.000	6.39s	> 6h	3h 39m	2.01s
²RS12	$|\mathcal{Z}|$	400	NFCCE	15,256	6.010	not run	0.04s	0.03s	0.02s
	$|\Sigma|$	613	EFCCE	15,256	6.010	not run	0.08s	0.08s	0.02s
	k	15	EFCE	8,846	6.010	not run	0.54s	0.12s	0.01s
²RS13	$|\mathcal{Z}|$	4,356	NFCCE	107,201,638	9.398	not run	3.33s	2.44s	1m 32s
	$|\Sigma|$	15,063	EFCCE	177,846,004	9.385	not run	2m 11s	1m 2s	2m 35s
	k	40	EFCE	135,762,741	9.367	not run	> 6h	14m 25s	2m 18s
²RS22	$|\mathcal{Z}|$	484	NFCCE	53,983	7.188	not run	0.08s	0.29s	0.06s
	$|\Sigma|$	701	EFCCE	53,983	7.176	not run	0.13s	0.10s	0.06s
	k	15	EFCE	31,503	7.176	not run	0.56s	0.24s	0.05s
²RS23	$|\mathcal{Z}|$	4,096	NFCCE	oom	10.961	not run	2.65s	3.33s	oom
	$|\Sigma|$	13,277	EFCCE	oom	10.820	not run	1m 51s	42.03s	oom
	k	44	EFCE	oom	10.791	not run	> 6h	7m 11s	oom

We also ran experiments on three-player games using the correlation DAG LP (see Table C.1 in the appendix). This shows, for the first time, that it is possible to compute optimal NFCCE/EFCCE/EFCE in practice for large game instances even when the number of players is greater than two.

6.4. Payoff Space Plots

In Figure 9, we show plots of the space of feasible payoffs in several tested games. All three-player games we tested on were constant sum, so for those games we show a 2D projection.

In most games tested, all three payoff spaces are different, and show very detailed boundaries that almost seem smooth (though, of course, they cannot be, since the payoff space is a polytope). This confirms the findings of earlier papers, for example, Farina et al. [14], and demonstrates the importance of defining the various notions as separate.

Figure 9. (Color online) Payoff spaces in several games, with all three notions of equilibrium. More such plots can be found in the appendix, in Figures C.1 and C.2.
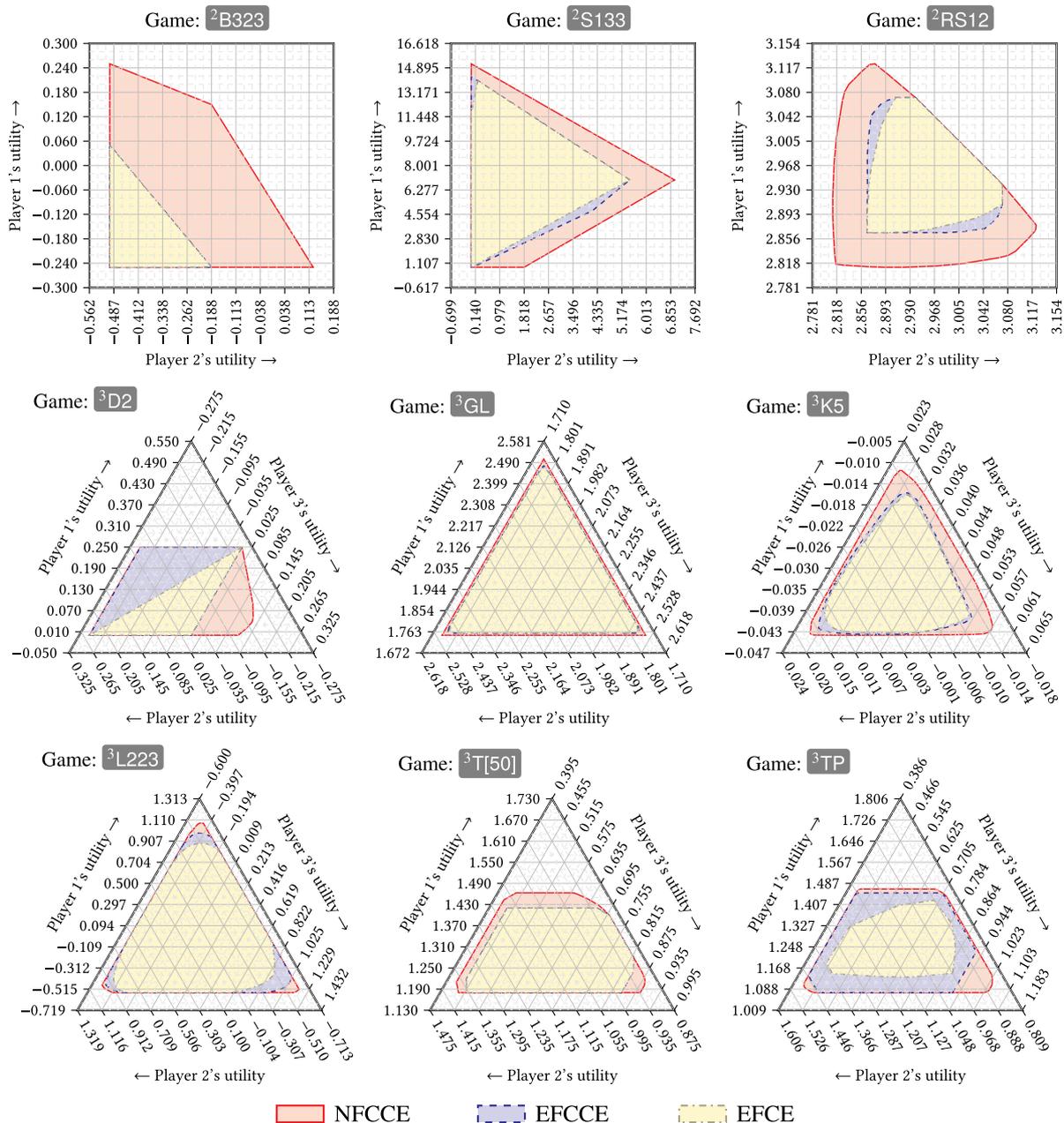
6.5. Adversarial Team Games

We compared our new column generation approach (Algorithm 2) to prior approaches for finding team-correlated equilibria (TMECor) in zero-sum adversarial team games. Specifically, we compared with the CFR algorithm on the team DAG introduced by Zhang et al. [40] and the prior column generation-based approach of Farina et al. [15]. Results can be found in Table 3. Our results clearly give several conclusions. First, our algorithm is an improvement upon Farina et al. [15], achieving speedups of more than an order of magnitude in some games. Second, our algorithm, like Zhang et al. [42], scales well in the information complexity k compared with that of Zhang et al. [40]: while ours is slower when k is small, it begins to match and quickly exceed the performance of that algorithm when k grows larger, as happens in Kuhn poker.

In the Tricks game instances, we observe that the perfect-information value, 0.66, does not match the team game value, 0.57. The discrepancy of nearly 0.1 tricks is nontrivially large given that there are only three tricks remaining. This establishes that, even in small endgames with three cards left, the fact that players do not know the cards of their teammate or opponent is still relevant information in a game of bridge, showing the importance of viewing bridge as a true imperfect-information game between two teams, rather than as a perfect-information game as *double dummy* bridge endgame solvers do (e.g., Ginsberg [19]).

Table 3. Experiments on TMECor in adversarial team games, comparing our two-sided column generation approach to earlier approaches. **DAG CFR** is the CFR-based team DAG algorithm of Zhang et al. [40]. **FCGS21** is the one-sided column generation approach of Farina et al. [15]. All runtimes are reported to a target precision of 0.005 times the reward range of the game. The game value of ^{3}T is after our new incremental algorithm ran to the time limit, and is accurate to ±0.002. All other game values are accurate to three decimals. Runtimes are colored according to the ratio with the runtime of our two-sided column generation, according to the scale [≤ 1/10 — 1 — ≥ 10].
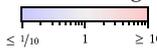
Game	{▼}	Leaves	▲ Value	k	DAG CFR [ZFS22]	Column generation [FCGS21]	This paper
^{3}K3	{3}	78	0.000	6	0.00s	0.00s	0.00s
^{3}K4	{3}	312	−0.042	8	0.00s	0.02s	0.01s
^{3}K6	{3}	1,560	−0.024	12	0.01s	0.15s	0.04s
^{3}K8	{3}	4,368	−0.019	16	0.79s	0.36s	0.06s
^{3}K12	{3}	17,160	−0.014	24	oom	1.24s	0.43s
^{3}L132	{3}	4,500	0.293	6	0.01s	52.66s	5.75s
^{3}L133	{3}	6,477	0.215	6	0.01s	1m 16s	7.71s
^{3}L151	{3}	10,020	−0.019	10	0.07s	22.03s	3.29s
^{3}L223	{3}	8,762	0.516	4	0.01s	2m 13s	4.61s
^{3}L523	{3}	775,148	0.953	4	3.60s	> 6h	4h 39m
^{3}D2	{3}	504	0.250	4	0.00s	0.11s	0.05s
^{3}D3	{3}	13,797	0.284	6	0.06s	5m 13s	5.95s
^{3}GL	{3}	1,296	1.252	2	0.00s	0.78s	0.21s
^{3}T[50]	{2}	10,300	0.600	5	0.00s	5.49s	0.67s
^{3}T[100]	{2}	20,992	0.710	18	0.07s	10.54s	1.62s
^{3}T[840]	{2}	190,228	0.569	141	oom	16m 40s	1m 51s
^{3}T	{2}	379,008	≈ 0.573	141	oom	4h 10m	22m 58s
^{3}TP	{2}	379,008	0.658	2	0.49s	6m 55s	1m 3s

7. Conclusions and Future Research

In this paper, we introduced and analyzed two new approaches for finding optimal correlated equilibria in general-sum games: the *correlation DAG* and a *two-sided column generation* algorithm. The former has complexity parameterizable by the *information complexity* k of the game. The two techniques have complementary practical strengths and weaknesses: when k is small, the correlation DAG shines; when k grows large, the column generation technique is faster and more frugal in terms of memory usage. Furthermore, the value of k can be easily computed, enabling an efficient choice between the two approaches. Our techniques are the state of the art in practice across all the games tested (except two small games where the prior column-generation approach was slightly faster). Possible directions of future research include the following.

1. Extend the correlation DAG in such a way that it also has polynomial size in all triangle-free games.

2. 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.

3. Investigate possible use of the payoff structure in the game; for example, investigate extensions of the concept of *smooth games* (Roughgarden [34]).

4. Our algorithms for optimal correlation all ultimately reduce to linear programs or mixed-integer programs. On the other hand, as we have discussed, regret minimization algorithms are known to be able to find *one* correlated equilibrium in all the notions we discuss in the paper, as well as equilibria in adversarial team games. We leave it to future research to answer whether regret minimization—with exact utility gradient or even with sampling (when only a gradient estimate is available)—can be made to lead to *optimal* correlated equilibria.

Since the publication of the initial version of this paper, some developments have been made along these lines. Zhang et al. [43] developed a method for combining the team belief DAG with column generation, which can also be applied to optimal equilibria, leading to "best-of-both-worlds" performance that combines the strengths of the two approaches. Peng and Rubinstein [31] and Dagan et al. [12] have shown that, for the stronger notion of NFCE and absolute constant gap ε, there exists a learning-based polynomial-time algorithm for computing an ε-NFCE. (It remains an open question whether there exists such an algorithm for $\varepsilon < 1/\text{poly}(|\mathcal{H}|)$.) Zhang et al. [41] developed learning-based algorithms for computing optimal equilibria including—but not limited to— NFCCE, EFCCE, and EFCE that essentially match the parameterized complexity guarantees offered by the present paper by taking a Lagrangian relaxation of (1) and viewing it as a zero-sum game.

Acknowledgments

Appendix A. Team Belief DAG

In the interest of self-containment, in this section we give a description of the construction of Zhang et al. [40]. Given a timeable extensive-form game Γ and a player i with imperfect recall, the goal is to represent the sequence form of player i. To do this, we construct a DAG $\mathcal{D}$ with two types of nodes: *decision nodes*, at which i makes a decision, and *observation nodes*, at which i observes something. Nodes in $\mathcal{D}$ will be identified with *sets* of histories in Γ, with the exception that we allow an observation node and a decision node to have the same set.

- The root node of $\mathcal{D}$ is the decision node $\{\varnothing\}$, where $\varnothing$ is the root node of Γ.
- Decision nodes $\{z\}$ for $z \in Z$ are leaves of $\mathcal{D}$.
- At a decision node B, let $I_1, \dots, I_m \in \mathcal{I}_i$ be the infosets with nonempty intersection with B. Player i may pick any *prescription*

$$\boldsymbol{a} \in \bigtimes_{i \in [m]} A_{I_i}.$$

The next node is the observation node

$$B\boldsymbol{a} := \{h\boldsymbol{a}_i : h \in I_i \cap B\} \cup \{ha : h \in B \setminus \mathcal{H}_i, a \in A_h\}$$

- At an observation node O, let $B_1, \dots, B_m$ be the connected components of the induced subgraph $G_i[O]$, where G_i is player i's connectivity graph (Definition 8).

The critical observation of Zhang et al. [40] (which we will not explicitly prove here) is that the set of sequence-form mixed strategies for player i in Γ is a projection of the set of strategies of the player in the decision problem $\mathcal{D}$, and moreover the latter set has a representation using linear constraints with size $O^*(|\mathcal{D}|)$, where $|\mathcal{D}|$ denotes the number of nodes in $\mathcal{D}$. Theorem 1 then follows by observing that the decision nodes in $\mathcal{D}$ are precisely the beliefs $B \in \mathcal{B}_i$ and for each such belief B there are $\prod_{I \in \mathcal{I}_i : B \cap \mathcal{I} \neq \varnothing} A_I$ actions.

Appendix B. Omitted Proofs
B.1. Theorem 3 (Games with Public Actions)

Theorem 3. *In games with public player actions, $R_{\mathbf{M}}^{\text{NFCCE}} = O^*(3^k)$ and $R_{\mathbf{M}}^{\text{EFCCE}} = O^*(d^k)$.*

Proof. Zhang et al. [40] devise an algorithm for constructing, 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. 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. □

Appendix C. Games in Experiments
C.1. Trick-Taking Game (Bridge Endgame)

We introduce a *trick-taking game*, which is effectively a bridge endgame scenario. There is a fixed deck of playing cards consisting of 3 ranks (2, 3, 4) of each of four suits (♠, , ♦, ♣). Spades (♠) is designated as the *trump suit*. There are four players: two *defenders*, who sit across from each other at the table, the *dummy*, and the *declarer*. The actions of the dummy will be controlled by the declarer; as such, there are actually only three players in the game. However, in this section, we will use the four-player terminology because it is easier to understand.

The whole deck is randomly dealt to four players. The dummy's cards are then publicly revealed. Play proceeds in *tricks*. The player to the left of the declarer leads the first trick. In each trick, the leader of the trick first plays a card. The suit of that card is the *lead suit*. Then, in clockwise order around the table, the other three players play a card from their hand. Players must play a card of the lead suit if they have such a card; otherwise, they may play any card. If any ♠ has been played, then whoever plays the highest ♠ wins the trick. Otherwise, the highest card of the lead suit wins the trick. The winner of one trick leads the next trick. At the end of the game, each player earns as many points as tricks they have won. For the adversarial team game, the two defenders are teammates, playing against the declarer (who controls the dummy).

We use ^3T to refer to the trick-taking game. In the *perfect information* variant ^3TP, all information is public, creating a perfect-information game. This is equivalent to what the bridge community calls a *double dummy* game. In all our games, the dummy's hand is fixed as 2♠ 2 3 .

In the *limited deals* variant $^3\text{T}[L]$, L deals are randomly selected at the beginning of the game, and it is common knowledge that the true deal is among them. This limits the size of the game tree, as well as the parameters on which the complexity of our algorithms depend. $L = 9!/(3!)^3 = 1680$ is the full game.

Table C.1. Experiments on general-sum correlated equilibria in 3-player games, with the correlation DAG. All runs were performed to convergence. Since all games tested are constant sum, instead of reporting the social welfare optimum (which is always the constant sum), we report the optimal utility for each individual agent in every solution concept.

						Optimal value		
Game	Leaves	k	Concept	$|\mathcal{E}^c|$	Runtime	P1	P2	P3
$^3\text{K4}$	312	12	NFCCE	259,176	0.63 s	−0.018	−0.007	0.064
			EFCCE	370,408	0.87 s	−0.020	−0.012	0.057
			EFCE	249,508	0.72 s	−0.021	−0.013	0.055
$^3\text{K5}$	780	15	NFCCE	4,182,981	18.44 s	−0.011	0.017	0.057
			EFCCE	7,236,161	49.89 s	−0.016	0.015	0.052
			EFCE	5,150,241	39.14 s	−0.016	0.013	0.052
$^3\text{L223}$	8,762	6	NFCCE	605,941	2.32 s	2.450	2.197	2.072
			EFCCE	6,342,970	34.84 s	1.302	1.431	1.309
			EFCE	5,251,772	50.56 s	0.877	1.009	1.000
$^3\text{D2}$	504	6	NFCCE	34,212	0.05 s	0.250	0.250	0.131
			EFCCE	54,627	0.08 s	0.250	0.250	0.000
			EFCE	621,237	1.57 s	0.250	0.250	0.000
^3GL	1,296	10	NFCCE	29,865	0.04 s	2.505	2.505	2.505
			EFCCE	36,933	0.06 s	2.476	2.476	2.476
			EFCE	16,950	0.06 s	2.467	2.467	2.467
$^3\text{T}[50]$	10,300	15	NFCCE	154,973,683	1m 44 s	1.463	1.380	0.887
			EFCCE	155,188,423	1m 40 s	1.420	1.360	0.840
			EFCE	155,340,357	1m 31 s	1.420	1.360	0.840
^3TP	379,008	3	NFCCE	6,678,856	12.91 s	1.466	1.477	1.037
			EFCCE	11,625,688	28.73 s	1.451	1.442	0.922
			EFCE	6,714,256	17.13 s	1.451	1.442	0.922

Table C.2. Summary of notation used in the paper.

Games	
$[n] = \{1, \dots, n\}$	**the number of players**
$h \in \mathcal{H}$	a node in a game
$\varnothing$	the root node
$z \in \mathcal{Z}$	a terminal node
A_h, A_I	the set of actions available at node h or infoset I
a	an action
ha	the child of h reached by playing a
$\mathcal{H}_i$	the set of nodes at which player i (possibly chance) acts
$p(z)$	the probability that chance plays all actions on the path to z
$\sigma_i \in \Sigma_i$	a sequence of player i
$\sigma_i(h)$	the sequence of player i at node h
$x_i[s]$	the probability that player i plays to the given sequence or node s
$x_i \in \mathcal{X}_i$	a (sequence-form) mixed strategy of player i
$u_i(z), u_i(x), \cdots$	the (expected) utility of player i
$\preceq$	the precedence order induced by the game tree
$h \wedge h'$	the lowest common ancestor of h and h'
$\varnothing_i$	the empty sequence of player i
Correlation and Correlated Profiles	
τ	a trigger
$\bot$	the empty trigger
$\overline{\tau}$	where τ was activated—that is, I if $\tau = Ia$ is a sequence, and τ otherwise.
(h, a, τ)	a node in the augmented game
h^τ	shorthand for $(h, \bot, \tau)$ for NFCCE and EFCE; or $(h, *, \tau)$ for EFCCE
$\boldsymbol{\xi} \in \Xi^c$	a correlation plan, in notion c
$P \in \mathcal{P}$	a public state
$R_{\mathbf{M}}^c$	the size of the representation of Ξ^c
Parameters	
b	the (nonchance) branching factor
d	the depth of the game tree
k	the information complexity
von Stengel–Forges and Column Generation	
Σ	the set of *relevant* joint sequences
$\sigma_1 \bowtie \sigma_2$	$(\sigma_1, \sigma_2) \in \Sigma$
$\sigma_1 \bowtie I$	$(\sigma_1, Ia) \in \Sigma$ for all $a \in A_I$
$\mathcal{V}$	the von Stengel–Forges polytope
Ξ_1^{sr}	the set of semirandomized plans of player i

C.2. Ride-Sharing Game Instances

We introduce a new benchmark which we call *ride-sharing game*.

C.2.1. General Rules of the Game The game models the interaction between two players (*a.k.a.*, drivers), which compete to serve requests on a road network. In particular, the network is modeled as an undirected graph $G^{\mathrm{rs}} = (V^{\mathrm{rs}}, E^{\mathrm{rs}})$. Each vertex $v \in V^{\mathrm{rs}}$ corresponds to a ride request to be served. Each ride request has a reward in $\mathbb{R}_{\geq 0}$. Each edge in the road network has some cost (representing the time incurred to traverse the edge). The first driver who arrives on node $v \in V^{\mathrm{rs}}$ serves the corresponding ride, and receives the corresponding reward. Once a node has been served, it stays clean until the end of the game. The game terminates when all requests have been served, or when a timeout is met (i.e., there's a fixed time horizon T). If the two drivers arrive on the same vertex at the same time they get reward 0. The final utility of each driver is the sum of the rewards obtained from the beginning until the end of the game. The initial position of the two drivers is randomly selected at the beginning of the game. Finally, the two drivers can observe each other's position only when they are simultaneously on the same node, or they are in adjacent nodes.

C.2.2. Objective and Remarks Ride-sharing games are particularly well-suited to study the computation of optimal correlated equilibria because they are two-player, general-sum games which are *not* triangle-free (Farina and Sandholm [13]). That is not the case for some of the existing two-player general-sum benchmarks, such as Goofspiel. We take the perspective of a *centralized platform* that has the goal of steering the drivers' behavior so as to maximize the overall social welfare. The platform can send *recommendations* to players in the form of navigation instructions. The goal of the platform is to ensure that such recommendations are incentive compatible, and maximize the SW attained at the equilibrium. Depending on the type of interaction in place between the platform and the players, the platform's goal amounts to finding an optimal (i.e., social-welfare maximizing) NFCCE/EFCCE/EFCE. For example, if the platform implemented an EFCE-like interaction protocol, at each new vertex in V^{rs} a driver would receive a suggestion about the next road to take

from there. The driver would be free to deviate as such decision point, since they could decide to take another direction, and that would come at the cost of future recommendations.

C.2.3. Implementation Details In our experiments, we employ road networks with unitary cost associated to edges. We write ^{n}RSiT to indicate the game instance has n players, and was generated from map i with time horizon T (i.e., each driver can make at most T steps). We employ two maps (map 1 and map 2), and we generate the instances ^{2}RS13, ^{2}RS14, ^{2}RS23. In Figure C.1 we report the structure of the two maps. The value between curly brackets is the reward for a request on that node.

Figure C.1. (Color online) Payoff spaces in two-player games.
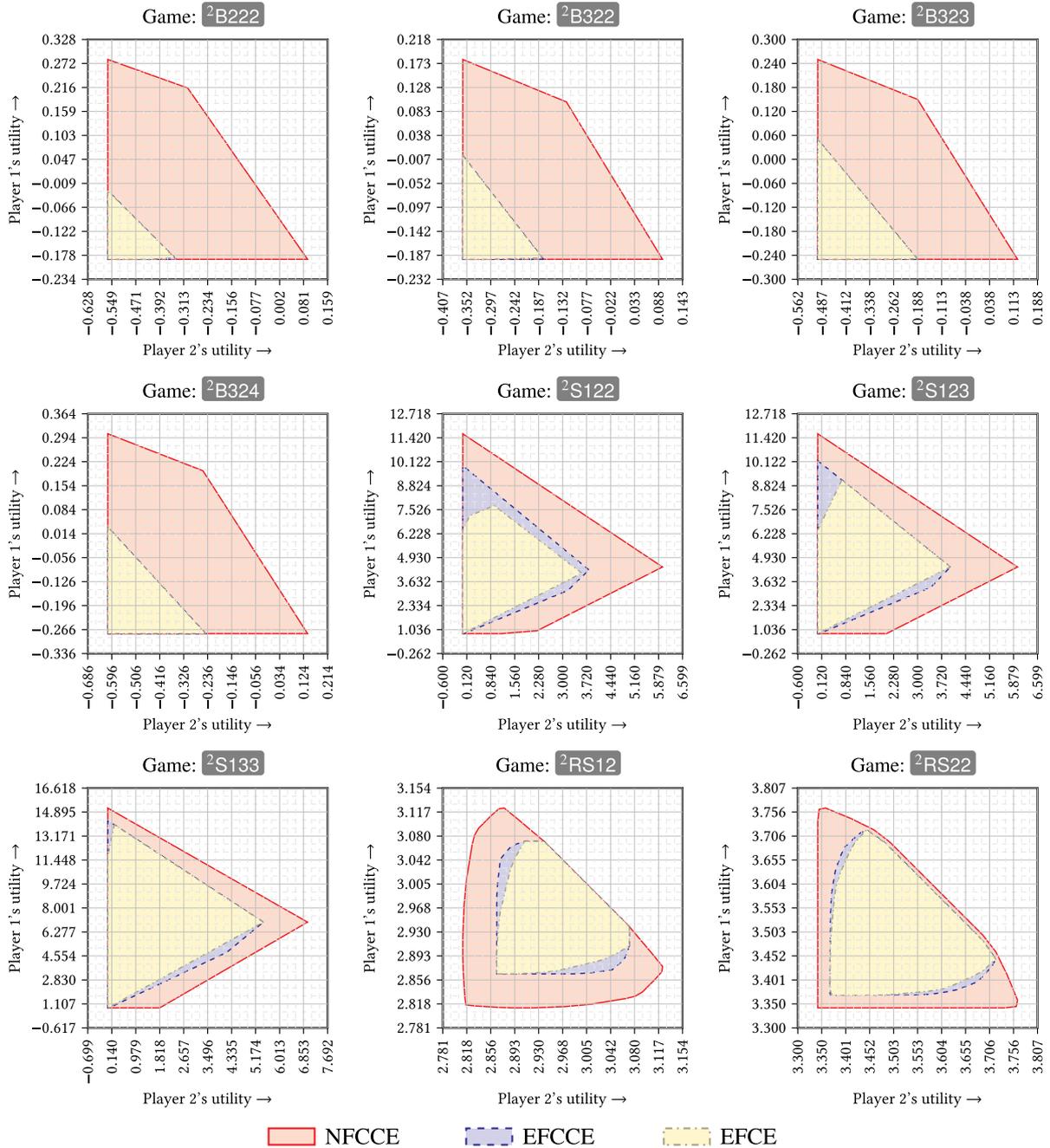
**Figure C.2.** (Color online) Payoff spaces in three-player, fixed-sum games.
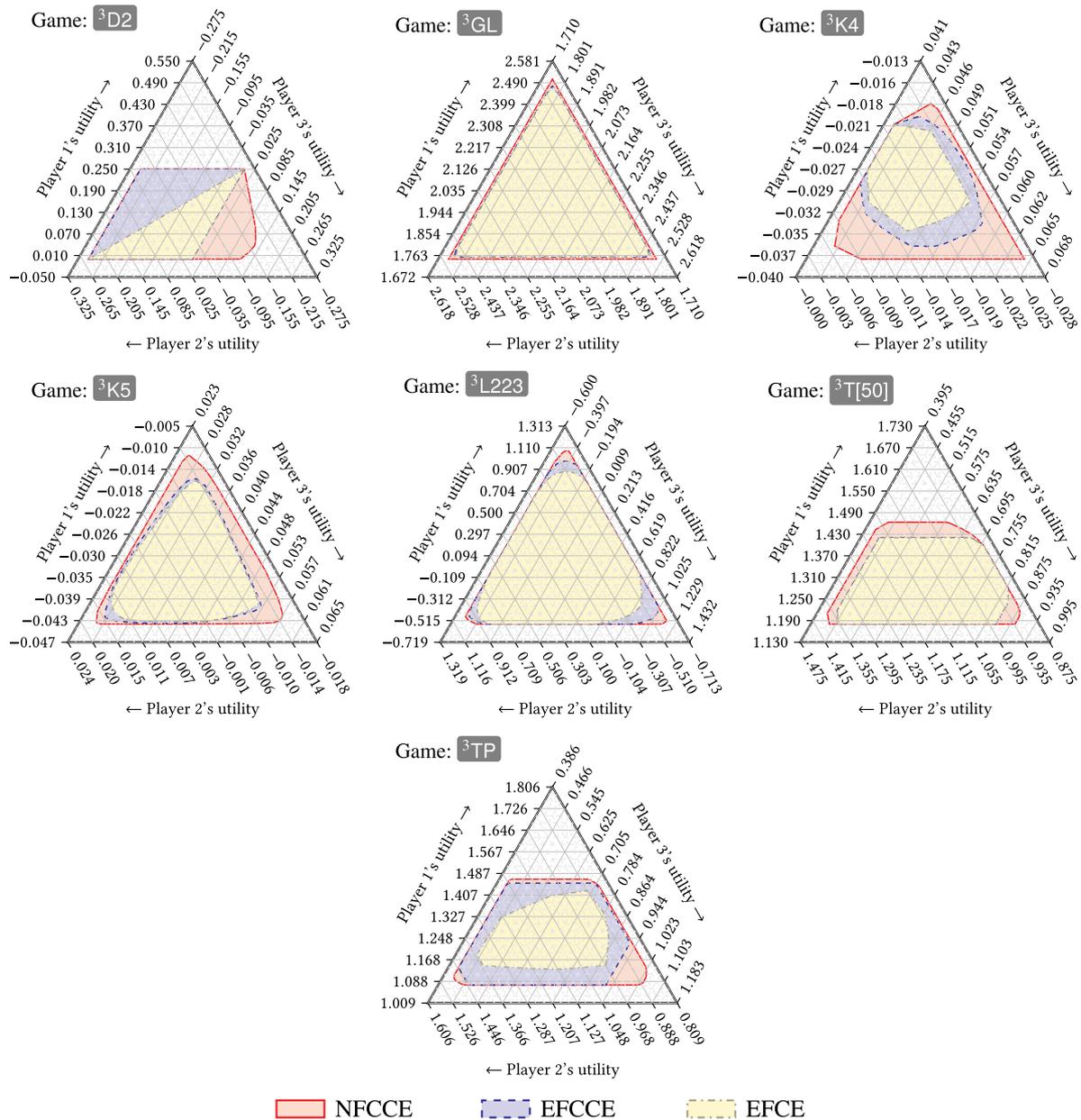
Figure C.3. The two road network configurations which we consider. *Left*: map 1 (used for ²RS13, ²RS14). *Right*: map 2 (used for ²RS23). In both cases the position of the two drivers is randomly chosen at the beginning of the game, edge costs are unitary, and one reward per node is indicated between curly brackets.
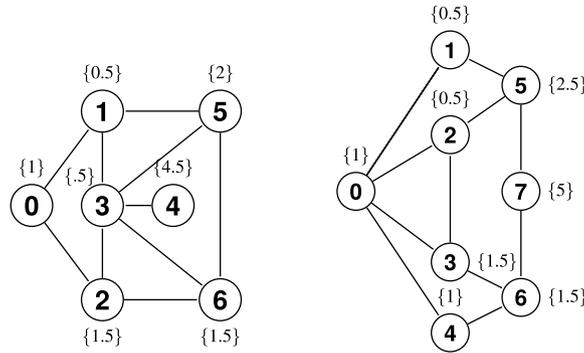
Endnotes

[1] The sole purpose of this "dummy layer" is to maintain timeability of Γ^c: for $c = $ EFCCE and $c = $ EFCE, each layer of the game tree of Γ needs to become *two* layers in Γ^c.

[2] Recall that Π_i is the set of pure strategies for player i, obeying any imperfect recall constraints.

[3] The k-CLIQUE problem is to decide whether a given graph contains a clique of size at least k.

[4] 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.

[5] The ideas in this section also naturally extend to games with more than two players, where a semi-randomized correlation plan is a correlation plan in which a single player is allowed to randomize and the other players must play pure strategies. The notation for that would be significantly more cumbersome, and the performance benefit would be less noticeable because the majority of players would still be forced to play pure strategies; as such, we restrict our attention in this section to the two-player case.

[6] Notation like $x_1 m_2^\top$ is technically an abuse of notation, since ξ is only indexed over *relevant* sequence pairs $\Sigma \subset \Sigma_1 \times \Sigma_2$. To be fully precise, by writing $ab^\top \in \mathbb{R}^\Sigma$ we refer to the vector $(ab^\top)[\sigma_1, \sigma_2] = a[\sigma_1]b[\sigma_2]$. But we will still call it an "outer product".

[7] In this paragraph only, we drop the superscript (T) for simplicity of notation.

[8] The full game has $L = 9!/(3!)^3 = 1680$.

References

[1] Aumann R (1974) Subjectivity and correlation in randomized strategies. *J. Math. Econom.* 1(1):67–96.
[2] Brown N, Sandholm T (2018) Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science* 359(6374):418–424.
[3] Brown N, Sandholm T (2019) Superhuman AI for multiplayer poker. *Science* 365(6456):885–890.
[4] Carminati L, Cacciamani F, Ciccone M, Gatti N (2022) A marriage between adversarial team games and 2-player games: Enabling abstractions, no-regret learning, and subgame solving. *Internat. Conf. Machine Learn. (ICML)* (PMLR, Baltimore).
[5] Celli A, Gatti N (2018) Computational results for extensive-form adversarial team games. *AAAI Conf. Artificial Intelligence (AAAI)* (Association for the Advancement of Artificial Intelligence (AAAI Press), Palo Alto, CA).
[6] Celli A, Coniglio S, Gatti N (2019) *Computing Optimal Ex Ante Correlated Equilibria in Two-Player Sequential Games. Autonomous Agents and Multi-Agent Systems (AAMAS)* (International Foundation for Autonomous Agents and MultiAgent Systems (IFAAMAS), Richland, SC).
[7] Celli A, Marchesi A, Farina G, Gatti N (2020) No-regret learning dynamics for extensive-form correlated equilibrium. *Neural Inform. Proc. Systems* (Neural Information Processing Systems Foundation, Inc. (NeurIPS)).
[8] Chen J, Chor B, Fellows M, Huang X, Juedes D, Kanj IA, Xia G (2005) Tight lower bounds for certain parameterized NP-hard problems. *Inform. Comput.* 201(2):216–231.
[9] Chen X, Deng X, Teng SH (2009) Settling the complexity of computing two-player Nash equilibria. *J. ACM* 56(3):14.
[10] Chu F, Halpern J (2001) On the NP-completeness of finding an optimal strategy in games with common payoffs. *Internat. J. Game Theory* 30:99–106.
[11] Conforti M, Cornuéjols G, Zambelli G (2010) Extended formulations in combinatorial optimization. *4OR* 8(1):1–48.
[12] Dagan Y, Daskalakis C, Fishelson M, Golowich N (2024) From external to swap regret 2.0: An efficient reduction and oblivious adversary for large action spaces. *Sympos. Theory Comput. (STOC)* (Association for Computing Machinery, New York).
[13] Farina G, Sandholm T (2020) Polynomial-time computation of optimal correlated equilibria in two-player extensive-form games with public chance moves and beyond. *Neural Inform. Proc. Systems* (Neural Information Processing Systems Foundation, Inc. (NeurIPS)).
[14] Farina G, Bianchi T, Sandholm T (2020) Coarse correlation in extensive-form games. *AAAI Conf. Artificial Intelligence (AAAI)* (Association for the Advancement of Artificial Intelligence (AAAI Press), Palo Alto, CA).
[15] Farina G, Celli A, Gatti N, Sandholm T (2021) Connecting optimal ex-ante collusion in teams to extensive-form correlation: Faster algorithms and positive complexity results. *Internat. Conf. Machine Learn. (ICML)* (PMLR, New York).

[16] Farina G, Celli A, Marchesi A, Gatti N (2022) Simple uncoupled no-regret learning dynamics for extensive-form correlated equilibrium. *J. ACM* 69(6):41.

[17] Farina G, Ling CK, Fang F, Sandholm T (2019) Correlation in extensive-form games: Saddle-point formulation and benchmarks. *Neural Inform. Proc. Systems* (Neural Information Processing Systems Foundation, Inc. (NeurIPS), Vancouver, BC).

[18] Ford LR, Fulkerson DR (1958) A suggested computation for maximal multi-commodity network flows. *Management Sci.* 5(1):97–101.

[19] Ginsberg ML (1999) GIB: Steps toward an expert-level bridge-playing program. *Internat. Joint Conf. Artificial Intelligence (IJCAI)* (Morgan Kaufmann Publishers Inc., Stockholm).

[20] Gordon GJ, Greenwald A, Marks C (2008) No-regret learning in convex games. *Internat. Conf. Machine Learn. (ICML)* (Association for Computing Machinery (ACM), New York).

[21] Huang W, von Stengel B (2008) Computing an extensive-form correlated equilibrium in polynomial time. *Internat. Workshop Internet Network Econom. (WINE)* (Springer, Berlin, Heidelberg).

[22] Jakobsen SK, Sørensen TB, Conitzer V (2016) Timeability of extensive-form games. *2016 ACM Conf. Innovations Theoretical Comput. Sci.* (Association for Computing Machinery (ACM), New York).

[23] Jiang AX, Leyton-Brown K (2015) Polynomial-time computation of exact correlated equilibrium in compact games. *Games Econom. Behav.* 91:347–359.

[24] Koller D, Megiddo N (1992) The complexity of two-person zero-sum games in extensive form. *Games Econom. Behav.* 4(4):528–552.

[25] Koller D, Megiddo N, von Stengel B (1994) Fast algorithms for finding randomized strategies in game trees. *Sympos. Theory Comput. (STOC)* (Association for Computing Machinery (ACM), New York).

[26] Kuhn HW (1950) A simplified two-person poker. Kuhn HW, Tucker AW, eds. *Contributions to the Theory of Games*, Annals of Mathematics Studies, 24, vol. 1 (Princeton University Press, Princeton, NJ), 97–103.

[27] Lisý V, Lanctot M, Bowling MH (2015) *Online Monte Carlo counterfactual regret minimization for search in imperfect information games. Autonomous Agents and Multi-Agent Systems (AAMAS)* (International Foundation for Autonomous Agents and MultiAgent Systems (IFAAMAS), Richland, SC).

[28] Moravčík M, Schmid M, Burch N, Lisý V, Morrill D, Bard N, Davis T, Waugh K, Johanson M, Bowling M (2017) DeepStack: Expert-level artificial intelligence in heads-up no-limit poker. *Science* 356(1667):508–513.

[29] Moulin H, Vial JP (1978) Strategically zero-sum games: The class of games whose completely mixed equilibria cannot be improved upon. *Internat. J. Game Theory* 7(3–4):201–221.

[30] Papadimitriou CH, Roughgarden T (2008) Computing correlated equilibria in multi-player games. *J. ACM* 55(3):14.

[31] Peng B, Rubinstein A (2024) Fast swap regret minimization and applications to approximate correlated equilibria. *Sympos. Theory Comput. (STOC)* (Association for Computing Machinery (ACM), New York).

[32] Romanovskii I (1962) Reduction of a game with complete memory to a matrix game. *Soviet Mathematics* 3:678–681.

[33] Ross SM (1971) Goofspiel—The game of pure strategy. *J. Appl. Probab.* 8(3):621–625.

[34] Roughgarden T (2015) Intrinsic robustness of the price of anarchy. *J. ACM* 62(5):1–42.

[35] Southey F, Bowling M, Larson B, Piccione C, Burch N, Billings D, Rayner C (2005) Bayes' bluff: Opponent modelling in poker. *Conf. Uncertainty Artificial Intelligence (UAI)* (Association for Uncertainty in Artificial Intelligence (AUAI) Press, Arlington, VA).

[36] von Stengel B (1996) Efficient computation of behavior strategies. *Games Econom. Behav.* 14(2):220–246.

[37] von Stengel B, Forges F (2008) Extensive-form correlated equilibrium: Definition and computational complexity. *Math. Oper. Res.* 33(4):1002–1022.

[38] von Stengel B, Koller D (1997) Team-maxmin equilibria. *Games Econom. Behav.* 21:309–321.

[39] Zhang BH, Sandholm T (2022) Team correlated equilibria in zero-sum extensive-form games via tree decompositions. *AAAI Conf. Artificial Intelligence (AAAI)* (Association for the Advancement of Artificial Intelligence (AAAI Press), Palo Alto, CA).

[40] Zhang BH, Farina G, Sandholm T (2023a) Team belief DAG: Generalizing the sequence form to team games for fast computation of correlated team max-min equilibria via regret minimization. *Internat. Conf. Machine Learn. (ICML)* (Proceedings of Machine Learning Research (PMLR), New York).

[41] Zhang BH, Farina G, Anagnostides I, Cacciamani F, McAleer S, Haupt A, Celli A, Gatti N, Conitzer V, Sandholm T (2023b) Computing optimal equilibria and mechanisms via learning in zero-sum extensive-form games. *Neural Inform. Proc. Systems* (Neural Information Processing Systems Foundation, Inc. (NeurIPS), New Orleans).

[42] Zhang Y, An B, Černý J (2021) Computing *ex ante* coordinated team-maxmin equilibria in zero-sum multiplayer extensive-form games. *AAAI Conf. Artificial Intelligence (AAAI)* (Association for the Advancement of Artificial Intelligence (AAAI Press), Palo Alto, CA).

[43] Zhang Y, An B, Zeng DD (2024) DAG-based column generation for adversarial team games. *Internat. Conf. Machine Learn. (ICML)* (Proceedings of Machine Learning Research (PMLR), Vienna).