OPTIMAL CORRELATED EQUILIBRIA IN GENERAL-SUM EXTENSIVE-FORM GAMES: FIXED-PARAMETER ALGORITHMS, HARDNESS, AND TWO-SIDED COLUMN-GENERATION

Brian Hu Zhang, Gabriele Farina
Computer Science Department
Carnegie Mellon University
{bhzhang, gfarina}@cs.cmu.edu

Andrea Celli
Computing Sciences Department
Bocconi University
andrea.celli2@unibocconi.it

Tuomas Sandholm
Computer Science Department, CMU
Strategic Machine, Inc.
Strategy Robot, Inc.
Optimized Markets, Inc.
sandholm@cs.cmu.edu

ABSTRACT

We study the problem of finding *optimal correlated equilibria* of various sorts: *normal-form coarse correlated equilibrium (NFCCE)*, *extensive-form coarse correlated equilibrium (EFCCE)*, and *extensive-form correlated equilibrium (EFCE)*. This is NP-hard in the general case and has been studied in special cases, most notably *triangle-free games* (Farina & Sandholm, 2020), which include all two-player games with public chance moves. However, the general case is not well understood, and algorithms usually scale poorly. In this paper, we make two primary contributions.

First, we introduce the *correlation DAG*, a representation of the space of correlated strategies whose structure and size are dependent on the specific solution concept desired. It extends the *team belief DAG* of Zhang et al. (2022) to general-sum games. For each of the three solution concepts, its size depends exponentially only on a parameter related to the information structure of the game. We also prove a fundamental complexity gap: while our size bounds for NFCCE are similar to those achieved in the case of team games by Zhang et al. (2022), this is impossible to achieve for the other two concepts under standard complexity assumptions.

Second, we propose a *two-sided column generation approach* to compute optimal correlated strategies in extensive-form games. Our algorithm improves upon the *one-sided* approach of Farina et al. (2021a) by means of a new decomposition of correlated strategies which allows players to re-optimize 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, and that our two families of approaches have complementary strengths: the correlation DAG is fast when the parameter is small and the two-sided column generation approach is superior when the parameter is large.

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 (Moravčík et al., 2017; Brown & Sandholm, 2018). However, in general-sum and/or multi-player games, computing Nash equilibria is hard even in normal-form games (Chen et al., 2009). Further, the assumption in Nash equilibrium that players' strategies are *independent* may not apply in real-world situations, where, for example, agents may have access to a shared random seed or to a trusted mediator. Both of these concerns motivate the definition and computational study of notions of *correlated equilibria*.

In notions of 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, 1974) is a distribution of strategy profiles for which no player is ever incentivized to deviate. In a *normal-form coarse correlated equilibrium* (NFCCE) (Moulin & Vial, 1978; Celli et al., 2019), 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 & Forges (2008), and Farina et al. (2020) 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.

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 & Forges, 2008). Some special cases are known to be solvable efficiently. Most notably, Farina & Sandholm (2020) show that in so-called *triangle-free games*, which include all two-player games with public chance actions, optimal equilibria in all three equilibrium notions can be computed in polynomial time.

Correlated equilibria have a close relationship with *adversarial team games*, that is, games where two teams compete against each other. 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., 2021a). Recently, Zhang & Sandholm (2022) and Zhang et al. (2022) 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, it does not immediately extend to general-sum correlation: for that, we are not only interested in the 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.

We make the following two primary contributions.

First, we extend the construction of Zhang et al. (2022) to the case of general-sum correlation, by explicitly accounting for the relevant marginal probabilities. Our construction, like theirs, is based on a decomposition of the game into public states. It immediately yields a fast LP-based algorithm for optimal correlation. The size of our construction depends exponentially on the parameter that Zhang et al. (2022) call k, which here we refer to as the *information complexity*. The information complexity of a game represents the number of possible *player private states*, across all players, in each *public state*. Like Zhang et al. (2022), in NFCCE, we achieve a bound of[1] $O^*((b+1)^k)$ in games of branching factor b. However, this bound does not extend to EFCCE or EFCE, where instead we achieve $O^*((b+d)^k)$ and $O^*((bd)^k)$ respectively, where d is the depth of the game. In fact, we show that matching the bound for NFCCE and teams is impossible for EFCCE and

[1] Throughout this paper, O^* hides factors polynomial in the size of the game.

EFCE under standard complexity assumptions, demonstrating a fundamental complexity-theoretic gap between correlation in normal and extensive forms.

Second, we propose a new approach to computing optimal correlated equilibria which we call *two-sided column generation*. We start by deriving an LP formulation based on the strategy polytope by von Stengel & Forges (2008), and on the notion of *semi-randomized correlation plan* introduced by Farina et al. (2021a) in the context of team games. As in those two papers, this LP formulation is for two-player games. In the latter of those two prior approaches, one player is chosen to play a normal-form strategy, and the other plays a sequence-form strategy. Our approach improves upon this by, in effect, inserting a root node that allows *both* players to act simultaneously as the normal-form player, increasing the space of correlation plans reachable from any given support. We show that we can express any valid correlation plan as a convex combination of semi-randomized correlation plans. Since their number may be exponential in the size of the game, we incrementally generate through a mixed-integer-programming-based pricing oracle the support set of semi-randomized correlation plans which are used in the convex combination. Moreover, we propose a new decomposition of correlation plans which allows players to make an effective use of the current support, by letting each player re-optimize their strategy with respect to the marginals of correlation plans already in the support.

Our two 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, 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 1999]—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

Definition 2.1. An *extensive-form game* Γ with n *players*, which we will identify with the positive integers $[n] = \{1, \dots, n\}$ consists of the following: **1)** A rooted tree of *histories* $\mathcal{H}$, where the edges are labelled with *actions*. The root node of $\mathcal{H}$ will be denoted h_0. 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 $\mathcal{H}_0, \mathcal{H}_1, \dots, \mathcal{H}_n$ 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*, or *infosets*. 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 function* $u_i : \mathcal{Z} \to \mathbb{R}$, 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.

We will use $\preceq$ to denote the typical precedence relation induced by the game tree—that is, $h \preceq h'$ 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 $s \preceq s'$ for some $s \in S, s' \in S'$. We will use $h \wedge h'$ to denote the lowest common ancestor of h and h'.

The *sequence* $\sigma_i(h)$ and *private state* $\bar{\sigma}_i(h)$ of player i at node h are the sequence of information sets reached and actions played by i on the root $\to$ h path. If player i plays at h, then $\sigma_i(h)$ does *not* include the infoset at h, while $\bar{\sigma}_i(h)$ does[2]. 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 $\Sigma_i := \{\sigma_i(h) : h \in \mathcal{H}\}$. In perfect-recall games, a sequence can be identified with infoset-action pair Ia. We will use this identification.

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* of the pure strategy is the vector $\boldsymbol{\pi}_i \in \{0,1\}^{\Sigma_i}$, where $\pi_i[\sigma_i] = 1$ if player i plays every action on the path from $\varnothing_i$ to σ_i. For infosets or nodes v, we will use $\pi_i[v]$ as overloaded notation for $\pi_i[\sigma_i(v)]$. A *mixed strategy* $\boldsymbol{\mu}_i$ is a distribution over pure strategies. The *reach probability* of σ_i under $\boldsymbol{\mu}_i$ is $\mu_i[\sigma_i] := \mathbb{E}_{\pi_i \sim \boldsymbol{\mu}_i} \pi_i[\sigma_i]$. The set of mixed strategies of player i is denoted by Q_i. It is a convex polytope characterized by a linear constraint system of size[3] $O(|\Sigma_i|)$ (Koller et al., 1994).

A *pure strategy profile* $\boldsymbol{\pi} = (\boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_n)$ is a collection of pure strategies, one per player. For a pure strategy profile $\boldsymbol{\pi}$, we define $\pi[h] := \prod_{i \in N} \pi_i[h] \in \{0,1\}$ to be the indicator that all players play all actions on root $\to h$ path. The *expected utility* under $\boldsymbol{\pi}$ is $u(\pi) := \sum_{z \in \mathcal{Z}} p(z) u_i(z) \pi[z]$.

A *correlated strategy profile* $\boldsymbol{\mu}$ is a distribution over pure strategy profiles. The *reach probability* of h under $\boldsymbol{\mu}$ is $\mu[h] := \mathbb{E}_{\boldsymbol{\pi} \sim \boldsymbol{\mu}} \pi[h]$. The *expected utility* under $\boldsymbol{\mu}$ is $u(\boldsymbol{\mu}) := \mathbb{E}_{\boldsymbol{\pi} \sim \boldsymbol{\mu}} u(\boldsymbol{\pi})$.

2.2 Public States

Throughout this paper, we will require the concept of *public states*, defined as follows.

Definition 2.2. Two nonterminal nodes h, h' in the same layer of Γ are *connected* if: **1)** $\bar{\sigma}_i(h) = \bar{\sigma}_i(h')$ for all players i, *or* **2)** there is an infoset I such that $h \preceq I$ and $h' \preceq I$.

The *connectivity graph* of Γ is the graph whose nodes are the nonterminal histories in Γ and whose edges are given by the connectivity relation. A *public state* is a connected component of the connectivity graph. We will use $\mathcal{P}$ to denote the collection of public states. Intuitively, a public state P is a subset of nodes such that whether P has been reached is common knowledge among all players[4]. Notice that terminal nodes, under this definition, are *not* assigned public states.

2.3 Sequential Decision Making on DAGs

In this section, we review a paradigm for sequential decision making (Farina et al., 2019a) on directed acyclic graphs (DAGs), also used by Zhang et al. (2022).

Definition 2.3. A *DAG-form sequential decision problem* (DFSDP) is a DAG $\mathcal{D} = (\mathcal{S}, \mathcal{E})$, where **1)** $\mathcal{D}$ has a unique source node $s_0 \in \mathcal{S}$, **2)** each node $s \in \mathcal{D}$ is either a *decision node*, an *observation node*, or a sink, and **3)** for every pair of paths from the root leading to the same node, the last node common to both paths is a decision node.

The edges out of decision nodes are called *actions*, and the edges out of observation nodes are *observations*. Like in games, a *pure strategy* is an assignment of one action to each decision node of $\mathcal{D}$. The *sequence form* of a pure strategy is a vector which we will also denote by $\boldsymbol{\pi} \in \{0,1\}^{\mathcal{S}}$ where $\pi[s] = 1$ if the pure strategy plays all the actions on *some* root $\to s$ path[5], which we will call the *active path* to s. A *mixed strategy* μ is a distribution over pure strategies, and its sequence form is the vector $\boldsymbol{\mu} \in [0,1]^{\mathcal{S}}$ for which $\mu[s] = \mathbb{E}_{\pi \sim \mu} \pi[s]$. Like sequence-form mixed strategies in games, we will use $Q_{\mathcal{D}}$ to denote the set of sequence-form mixed strategies in a DFSDP, and this set is a polytope that can be described by a linear constraint system of size $O(|\mathcal{E}|)$ via *scaled extensions* (Farina et al., 2019c; Zhang et al., 2022).

[2]Some prior works on team games, such as Zhang et al. (2022), call $\bar{\sigma}_i(h)$ the *sequence* and do not use $\sigma_i(h)$—to avoid confusion, we differentiate the two.

[3]Throughout this paper, the *size* of a constraint system $\boldsymbol{A}\boldsymbol{x} \le \boldsymbol{b}$ is $\mathrm{nnz}(\boldsymbol{A})$.

[4]With our definitions of private states and public states, it is possible for two nodes in different public states to share the private state of some player. Intuitively, this is because public states are defined to be "as fine as possible", while private states are defined to be "as coarse as possible". Our analysis, like that of Zhang et al. (2022), relies on the *number of private states per public state*, so it is beneficial to define these terms in this way

[5]Condition **3)** in Definition 2.3 ensures that there is at most one such path.

2.4 Correlated Equilibria in Games

In this section, we formally define the various *solution concepts* of correlated equilibria that we will be working with in this paper: *normal-form coarse correlated equilibrium* (NFCCE), *extensive-form coarse correlated equilibrium* (EFCCE), and *extensive-form correlated equilibrium* (EFCE). We first give some intuition. These notions of correlated equilibria can be thought of as correlated strategies of play that can be *enforced by a mediator*. The mediator selects a strategy profile $\pi \sim \mu$, and, when each player reaches an infoset I, the mediator gives a *recommendation* that player to play the action prescribed by π 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. **1)** 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. **2)** 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. **3)** In EFCE, a player may deviate at each of her infosets *after* seeing a recommendation, by instead playing a different action. To formalize these notions, we use the language of *trigger agents* introduced by Gordon et al. (2008).

Definition 2.4. A *trigger* τ is: **1)** for NFCCE, the empty sequence $\varnothing_i$ for some player $i \in [n]$; **2)** for EFCCE, an infoset; and **3)** 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.Given a trigger $\tau \in \mathcal{T}^c$, we use $\mathcal{H}[\tau]$ to denote the set of nodes at which trigger τ may be activated. Formally, **1)** for NFCCE, $\mathcal{H}[\varnothing_j] = \{h_0\}$; and **2)** for EFCCE, $\mathcal{H}[I] = I$; and **3)** for EFCE, $\mathcal{H}[Ia] = I$. A *pure deviation* $\boldsymbol{\pi}'_i$ following a trigger τ of player i is a pure strategy defined on all infosets $I \succeq \mathcal{H}[\tau]$. Mixed deviations and their sequence forms are defined analogously. We will denote by $Q[\tau]$ the set of all sequence-form mixed deviations following τ. A *trigger agent* $\alpha = (\tau, \boldsymbol{\pi}'_i)$ consists of a trigger τ of player i and a pure deviation $\boldsymbol{\pi}'_i$ following τ.

For a trigger agent $\alpha = (\tau, \boldsymbol{\pi}'_i)$ and pure strategy $\boldsymbol{\pi}_i$, define $\boldsymbol{\pi}^\alpha_i$ to be identical to $\boldsymbol{\pi}_i$ unless $\pi_i[\tau] = 1$, in which case $\boldsymbol{\pi}'_i$ replaces $\boldsymbol{\pi}_i$ where it is defined. That is, the player i plays according to the original strategy $\boldsymbol{\pi}_i$ unless it prescribes τ, in which case she replaces her strategy with $\boldsymbol{\pi}'_i$ where the latter is defined. For a pure strategy profile $\boldsymbol{\pi}$, define $\pi^\alpha[h] := \pi^\alpha_i[h] \prod_{j \neq i} \pi_j[h]$. For a correlated strategy $\boldsymbol{\mu}$, define $\mu^\alpha[h] := \mathbb{E}_{\boldsymbol{\pi} \sim \boldsymbol{\mu}} \pi^\alpha[h]$. That is, $\mu^\alpha[h]$ is the reach probability of h if all players play $\boldsymbol{\mu}$, except player i, who deviates from $\boldsymbol{\mu}$ according to the trigger agent.

Definition 2.5. A trigger agent α is *profitable* if $u_i(\boldsymbol{\mu}^\alpha) > u_i(\boldsymbol{\mu})$.

Definition 2.6. NFCCEs, EFCCEs, and EFCEs are correlated strategy profiles $\boldsymbol{\mu}$ that have no profitable trigger agents of their respective types.

In this paper, we focus on the problem of computing *optimal* correlated equilibria of the given sort. That is, given an *objective function* $u : \mathcal{Z} \to \mathbb{R}$, our goal is to find the correlated equilibrium $\boldsymbol{\mu}$ of the desired concept that maximizes $u(\boldsymbol{\mu})$. One example special case of this problem is the *social welfare-maximizing* equilibrium, which is the equilibrium maximizing $u_{\text{SW}}(\boldsymbol{\mu}) := \sum_{i \in [n]} u_i(\boldsymbol{\mu})$.

3 Correlation Plans

In this section, we introduce our main contribution of this part of the paper, the *correlation DAG*. It is a generalization of the team belief DAG of Zhang et al. (2022) to representing the polytopes Ξ^c instead of just a team polytope[6]. Before we begin, we must first define the polytope we are trying to represent—namely, the polytope of correlation plans.

3.1 Correlation Plans and Optimal Correlation via Linear Programs

In this section, we introduce the *correlation polytope* Ξ, and its projections Ξ^c for each of the three solution concepts we are concerned with, *i.e., $c \in \{\text{NFCCE}, \text{EFCCE}, \text{EFCE}\}$*. Unlike previous works [e.g., Farina & Sandholm 2020], we define Ξ^c to be different for each solution concept. This allows us to give refined bounds and hardness results that depend on c.

[6] Indeed, if we disallow all triggers, our correlation DAG recovers effectively the team polytope of Zhang et al. (2022).

Definition 3.1. A *trigger history* h^τ is a history h augmented with an *optional* trigger τ such that $\mathcal{H}[\tau] \preceq h$. If there is no trigger, we write $\tau = \perp$.

If $\tau \neq \perp$, we say that the player to which τ belongs is *triggered* at h^τ. The set of trigger histories for concept c will be denoted $\mathcal{H}^c$, and the set of trigger histories z^τ for which z is terminal will be denoted $\mathcal{Z}^c$.

The *sequence* of a player i at a terminal trigger history h^τ is defined by $\sigma_i(h^\tau) = \sigma_i(\tau)$ if i is triggered at τ, and $\sigma_i(h^\tau) = \sigma_i(h)$ otherwise. The *joint sequence* $\boldsymbol{\sigma}(h^\tau)$ is the tuple of sequences of all players, that is, $\boldsymbol{\sigma}(h^\tau) := (\sigma_1(h^\tau), \dots, \sigma_n(h^\tau))$. We call a joint sequence *relevant* if it appears as the joint sequence of any h^τ for any solution concept, and use Σ to denote the set of relevant joint sequences. That is, $\Sigma := \{\boldsymbol{\sigma}(h^\tau) : h^\tau \in \mathcal{H}^{\text{NFCCE}} \cup \mathcal{H}^{\text{EFCCE}} \cup \mathcal{H}^{\text{EFCE}}\}$. This notion of relevance is a generalization of the notion of von Stengel & Forges (2008) to games with more than two players, and the two notions coincide in the two-player case.

The *reach probability* $\xi^{\boldsymbol{\mu}}[\boldsymbol{\sigma}]$ of terminal sequence $\boldsymbol{\sigma}$ under correlated strategy profile $\boldsymbol{\mu}$ is defined to be $\xi^{\boldsymbol{\mu}}[\boldsymbol{\sigma}] = \mathbb{E}_{\pi\sim\boldsymbol{\mu}} \prod_{i\in[n]} \pi_i[\sigma_i]$. That is, $\xi^{\boldsymbol{\mu}}[\boldsymbol{\sigma}]$ is the probability that all players play to $\boldsymbol{\sigma}$. As usual, we will use $\xi[h^\tau]$ as notational shorthand for $\xi[\boldsymbol{\sigma}(h^\tau)]$. The resulting vector $\boldsymbol{\xi}^{\boldsymbol{\mu}} \in [0,1]^\Sigma$ is called the *correlation plan* for $\boldsymbol{\mu}$ in concept c. We will use Ξ to denote the set of correlation plans.

We now define the restrictions Ξ^c for each solution concept c. A joint sequence $\boldsymbol{\sigma}$ is called *terminal* if $\boldsymbol{\sigma} = \boldsymbol{\sigma}(z^\tau)$ for some $z^\tau \in \mathcal{Z}^c$. We will use Σ^c to denote the set of terminal joint sequences in concept c, that is, $\Sigma^c := \{\boldsymbol{\sigma}(z^\tau) : z^\tau \in \mathcal{Z}^c\}$. We will call the subvector on Σ^c of a correlation plan $\boldsymbol{\xi} \in [0,1]^\Sigma$ a *correlation plan in concept* c, and use Ξ^c to denote the set of such correlation plans. Intuitively, Ξ^c is the projection of Ξ onto only the joint sequences that are needed to define that particular solution concept.

We define the correlation plan polytopes in this way because they have different characteristics: in particular, Ξ^{NFCCE} is a much lower-dimensional object than Ξ^{EFCCE}, Ξ^{EFCE}, and Ξ. In Section 4.1, we will show that this difference marks the transition between efficient parameterized algorithms being possible and impossible in our parameterization. The important fact about Ξ^c is the following:

Theorem 3.2 (Farina et al. 2019b; 2020)**.** *For each solution concept c, the set of solutions Ξ^c_* in concept c can be written in the form $\Xi^c_* = \{\boldsymbol{\xi} \in \Xi^c, \boldsymbol{A}\boldsymbol{\xi} \leq \boldsymbol{b}\}$, where $\boldsymbol{A} \in \mathbb{R}^{m\times\mathcal{Z}^c}$ with $m = \operatorname{poly}(|\mathcal{H}|)$. Therefore, in particular, if Ξ^c has a description of size D, then optimizations over Ξ^c_*, such as computing* welfare-optimal *solutions of concept c, can be done in time* $\operatorname{poly}(D)$.

The remainder of the paper will focus on ways of representing Ξ or Ξ^c. With such a representation, Theorem 3.2 can be applied directly to yield an efficient algorithm for optimal correlation.

4 The Correlation DAG

Definition 4.1. A trigger history h^τ is *active* if a mediator must recommend an action at h, that is, if $\tau = \perp$ or the player acting at h is not chance and not triggered.

We say that $h^\tau \in I$ if and only if $h \in I$ and h^τ is active. Given a set B of trigger histories, an infoset I is active if there exists an active trigger history $h^\tau \in B \cap I$. A *prescription* $\boldsymbol{a}$ at B is an assignment of one action to each active infoset I.

Definition 4.2. For a trigger history $h^\tau \in \mathcal{H}^c$ with trigger $\tau \neq \perp$, define the *trigger point* $t(h^\tau) = h'$ be the node h' that caused the trigger τ to be activated. Formally: **1)** For NFCCE, $t(h^{\varnothing_i}) = h_0$. **2)** For EFCCE, $t(h^I) = h'$, where $h' \preceq h$ is such that $h' \in I$. **3)** For EFCE, $t(h^{Ia}) = h'a$, where $h' \preceq h$ is such that $h' \in I$. Note that, in the case of EFCE, we have $t(h^\tau) \not\preceq h$, since h^τ can only arise when the mediator attempts to recommend an action that is then *not* played by the player.

We now extend the precedence order $\preceq$ to $\mathcal{H}^c$, as follows. For a trigger history h^τ, define the *path to* h^τ as the path $(h_0^{\tau^0}, h_1^{\tau^1}, \dots, h^\tau)$, where $(h_0, h_1, \dots, h)$ is the path to h in $\mathcal{H}$, and $\tau^\ell = \perp$ if $\tau = \perp$ or $h_\ell \prec t(h^\tau)$, and $\tau^\ell = \tau$ otherwise. Put simply, the path to h^τ is the path of trigger histories that occurs if node h has been reached and trigger τ was activated. Now if $\widetilde{h^\tau}$ is another trigger history, we say that $\widetilde{h^\tau} \preceq h^\tau$ if it is on the path to h^τ. We say that h^τ is *fresh* if $\tau \neq \perp$ and h is at the same layer as $t(h^\tau)$, that is, the trigger was just activated[7].

We are now ready to define the correlation DAG. Like Zhang et al. (2022), we will assume in this section that the game is *timed—i.e.*, information sets do not span multiple levels of the game tree[8].

Definition 4.3. The *correlation DAG* is a DFSDP $\mathcal{D}^c = (\mathcal{S}^c, \mathcal{E}^c)$, whose player we call the *mediator* and whose nodes are identified with *subsets of trigger histories* $S \subseteq \mathcal{H}^c$, defined as follows. **1)** The source node is an observation node $S_0 = \{h_0^\perp\}$. **2)** At an observation node O, let $\hat{O} = O \cup \{h^\tau \text{ fresh} : t(h^\tau)^\perp \in O\}$. Let $\mathcal{P}_S = \{P \in \mathcal{P} : h \in P, h^\tau \in \hat{O}\}$ be the set of public states containing nodes in $\hat{O}$. The mediator observes some public state $P \in \mathcal{P}_S$ and transitions to the decision node $\hat{O}P := \{h^\tau \in \hat{O} : h \in P\}$. Additionally, if $z^\tau \in \hat{O}$ is terminal[9], the mediator may observe z^τ and transition to the singleton terminal node $\{z^\tau\}$. Intuitively, at an observation node S, we first construct the set $\hat{S}$ of trigger-augmented histories that *could have been reached* by a mediator who attempted to reach S. Then, the mediator observes a public state and conditions the world on that public state, and the game continues. **3)** At a decision node B, the mediator chooses a prescription $\boldsymbol{a}$, and transitions to the observation node

$$B\boldsymbol{a} := \underbrace{\{ha[I]^\tau : h \in I, h^\tau \in B \text{ active}\}}_{\text{the player at } h \text{ is recommended action } a[I]} \cup \underbrace{\{h\tilde{a}^\tau : h^\tau \in B \text{ inactive}, \tilde{a} \in A_h\}}_{\text{the player at } h \text{ did not receive a recommendation}}.$$

The sets corresponding to decision nodes are called *beliefs*. The DAG $\mathcal{D}^c$ is created by merging decision nodes that have the same belief.

Pseudocode can be found in Algorithm 2, and an example can be found in Appendix D, both in the appendix.

To check that $\mathcal{D}^c$ is a valid DFSDP, we need to ensure that no two paths from the root to any node $S \in \mathcal{S}^c$ split off at an observation node. This is true because distinct observations at an observation node result in intersection with distinct public states, which are disjoint. Therefore, they lead to *disjoint* beliefs, which can never again meet.

If we do not allow any triggers at all and only consider nodes belonging to a subset of players that form a team, we recover the *team belief DAG* of Zhang et al. (2022). However, the team belief DAG is not sufficient to parameterize Ξ^c for any of the correlation concepts c, because it lacks—among other things—the marginal strategies of each individual player. As Zhang et al. (2022) have discussed the case of adversarial team games at length, we do not elaborate on this point in this section and instead focus our attention on the general-sum correlation case.

Theorem 4.4 (Correctness). *Let Γ be a game, and $\mathcal{D}^c$ its correlation DAG for any of the three concepts c. For a sequence-form mixed strategy $\boldsymbol{\mu} \in Q_{\mathcal{D}^c}$, define the vector $\boldsymbol{\xi}^{\boldsymbol{\mu}} \in [0,1]^{\Sigma^c}$ by $\xi^{\boldsymbol{\mu}}[\boldsymbol{\sigma}(z^\tau)] = \mu[\{z^\tau\}]$. Then we have $\Xi^c = \{\boldsymbol{\xi}^{\boldsymbol{\mu}} : \boldsymbol{\mu} \in Q_{\mathcal{D}^c}\}$. That is, the correlation plan polytope Ξ^c is a projection of the set $Q_{\mathcal{D}^c}$ of sequence-form mixed strategies in $\mathcal{D}^c$.*

Corollary 4.5. *Optimization problems over Ξ^c and Ξ^c_*, including but not limited to computing optimal equilibria, can be written as linear programs of size with $O^*(|\mathcal{E}^c|)$ constraints, and thus solved in time* $\mathrm{poly}(|\mathcal{E}^c|, |\mathcal{H}|)$.

We call this linear program the *correlation DAG LP*.

4.1 The Size of the Correlation DAG

In light of Corollary 4.5, it is important to discuss the number of edges $|\mathcal{E}^c|$ for each possible concept c. From a similar analysis to Zhang et al. (2022), we can do this by directly counting beliefs and edges. We will use the following parameters in our analysis**1)** k is

[7] For NFCCE and EFCCE, this is equivalent to saying $h = t(h^\tau)$. For EFCE, it is equivalent to h and $t(h^\tau)$ sharing a parent.

[8] This assumption is not without loss of generality, but any game in which the precedence order $\preceq$ on infosets is a partial order can be converted to a timed game by adding dummy nodes.

[9] Since multiple terminal trigger histories may have the same joint sequence, we do not need to do this for *all* terminal z^τ; indeed, we only need to pick one representative for each joint sequence, and ignore all other terminal trigger histories. This results in a slightly more compact representation.

the largest number of player private states (across all players) in any public state. Formally, $k = \max_{P \in \mathcal{P}} |\{\bar{\sigma}_i(h) : h \in P, i \in [n]\}|$. We will call k the *information complexity* of the game. **2)** b is the largest branching factor of any non-chance node in Γ. Formally, $b = \max_{h \in \mathcal{H} \setminus \mathcal{H}_0} |A_h|$. **3)** d is the depth of Γ, where a game with only a root node is defined to have depth 1.

Theorem 4.6. $\left|\mathcal{E}^{\text{NFCCE}}\right| = O^*\big((b+1)^k\big)$, $\left|\mathcal{E}^{\text{EFCCE}}\right| = O^*\big((b+d-2)^k\big)$, *and* $\left|\mathcal{E}^{\text{EFCE}}\right| = O^*\big((bd)^k\big)$.

The bound for NFCCE is the same bound achieved by Zhang et al. (2022) in the setting of team games. For EFCCE and EFCE, the bounds also depend on the number of triggers that could have been activated for a given player at a given node. For EFCCE, this is at most d; for EFCE, it is bd.

In games with public actions, we can remove the dependencies on b for NFCCE and EFCCE.

Definition 4.7. A game has *public actions* if, for all $P \in \mathcal{P}$ containing at least one non-chance node, for all actions $a \in \bigcup_{h \in P} A_h$, 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.

Theorem 4.8. *In games with public actions, the correlation DAG construction can be modified to achieve* $\left|\mathcal{E}^{\text{NFCCE}}\right| = O^*\big(3^k\big)$ *and* $\left|\mathcal{E}^{\text{EFCCE}}\right| = O^*\big(d^k\big)$.

The bound on $\left|\mathcal{E}^{\text{EFCE}}\right|$ is not improved, because the $(bd)^k$ term in that analysis comes from counting the number of triggers at a given node, which has not changed. Once again, the bound for NFCCE matches that of Zhang et al. (2022) up to polynomial factors.

5 Two-Sided Column Generation and Experiments

Due to lack of space, we defer the description of a new *two-sided column generation approach* to Appendix B. Briefly, our column generation approach builds on and improves that of Farina et al. (2021a), by allowing *both* players to take the role of the sequence-form player, thereby broadening the space of strategies available for a given set of support strategies and speeding up convergence.

Discussion of experiments is deferred to Appendix F. Briefly, our experimental results show that 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).

6 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. Possible directions of future research include the following. **1)** Our correlation DAG does not necessarily have polynomial size even in games with no chance nodes, so it does not subsume the earlier analyses of Farina & Sandholm (2020) and von Stengel & Forges (2008). We leave it to future research to devise a construction that subsumes all of these. **2)** An intelligent combination—rather than a selection of one versus the other—of the correlation DAG and the column generation algorithm may lead to faster practical algorithms. **3)** 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 can be made to lead to *optimal* correlated equilibria.

ACKNOWLEDGMENTS

This material is based on work supported by the National Science Foundation under grants IIS-1718457, IIS-1901403, and CCF-1733556, and the ARO under award W911NF2010081.

REFERENCES

Robert Aumann. Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, 1:67–96, 1974.

Noam Brown and Tuomas Sandholm. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359(6374):418–424, 2018.

Noam Brown and Tuomas Sandholm. Superhuman AI for multiplayer poker. *Science*, 365(6456): 885–890, 2019.

Andrea Celli, Stefano Coniglio, and Nicola Gatti. Computing optimal ex ante correlated equilibria in two-player sequential games. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 909–917, 2019.

Andrea Celli, Alberto Marchesi, Gabriele Farina, and Nicola Gatti. No-regret learning dynamics for extensive-form correlated equilibrium. *Advances in Neural Information Processing Systems*, 33: 7722–7732, 2020.

Jianer Chen, Benny Chor, Mike Fellows, Xiuzhen Huang, David Juedes, Iyad A Kanj, and Ge Xia. Tight lower bounds for certain parameterized np-hard problems. *Information and Computation*, 201(2):216–231, 2005.

Xi Chen, Xiaotie Deng, and Shang-Hua Teng. Settling the complexity of computing two-player Nash equilibria. *Journal of the ACM*, 2009.

Gabriele Farina and Tuomas Sandholm. Polynomial-time computation of optimal correlated equilibria in two-player extensive-form games with public chance moves and beyond. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

Gabriele Farina, Christian Kroer, and Tuomas Sandholm. Online convex optimization for sequential decision processes and extensive-form games. In *AAAI Conference on Artificial Intelligence*, 2019a.

Gabriele Farina, Chun Kai Ling, Fei Fang, and Tuomas Sandholm. Correlation in extensive-form games: Saddle-point formulation and benchmarks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019b.

Gabriele Farina, Chun Kai Ling, Fei Fang, and Tuomas Sandholm. Efficient regret minimization algorithm for extensive-form correlated equilibrium. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019c.

Gabriele Farina, Tommaso Bianchi, and Tuomas Sandholm. Coarse correlation in extensive-form games. In *AAAI Conference on Artificial Intelligence*, 2020.

Gabriele Farina, Andrea Celli, Nicola Gatti, and Tuomas Sandholm. Connecting optimal ex-ante collusion in teams to extensive-form correlation: Faster algorithms and positive complexity results. In *International Conference on Machine Learning*, 2021a.

Gabriele Farina, Andrea Celli, Alberto Marchesi, and Nicola Gatti. Simple uncoupled no-regret learning dynamics for extensive-form correlated equilibrium. *arXiv preprint arXiv:2104.01520*, 2021b.

L. R. Ford and D. R. Fulkerson. A suggested computation for maximal multi-commodity network flows. *Management Science*, 5(1):97–101, 1958. doi: 10.1287/mnsc.5.1.97.

Matthew L Ginsberg. Gib: Steps toward an expert-level bridge-playing program. In *IJCAI*, pp. 584–593. Citeseer, 1999.

Geoffrey J Gordon, Amy Greenwald, and Casey Marks. No-regret learning in convex games. In *25th international conference on Machine learning*, pp. 360–367. ACM, 2008.

Wan Huang and Bernhard von Stengel. Computing an extensive-form correlated equilibrium in polynomial time. In *International Workshop on Internet and Network Economics*, pp. 506–513. Springer, 2008.

Albert Xin Jiang and Kevin Leyton-Brown. Polynomial-time computation of exact correlated equilibrium in compact games. *Games and Economic Behavior*, 91:347–359, 2015.

Daphne Koller, Nimrod Megiddo, and Bernhard von Stengel. Fast algorithms for finding randomized strategies in game trees. In *ACM Symposium on Theory of Computing (STOC)*, 1994.

Daphne Koller, Nimrod Megiddo, and Bernhard von Stengel. Efficient computation of equilibria for extensive two-person games. *Games and Economic Behavior*, 14(2), 1996.

H. W. Kuhn. A simplified two-person poker. In H. W. Kuhn and A. W. Tucker (eds.), *Contributions to the Theory of Games*, volume 1 of *Annals of Mathematics Studies, 24*, pp. 97–103. Princeton University Press, Princeton, New Jersey, 1950.

Viliam Lisỳ, Marc Lanctot, and Michael H Bowling. Online monte carlo counterfactual regret minimization for search in imperfect information games. In *Autonomous Agents and Multi-Agent Systems*, 2015.

Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisý, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science*, May 2017.

H. Moulin and J.-P. Vial. Strategically zero-sum games: The class of games whose completely mixed equilibria cannot be improved upon. *International Journal of Game Theory*, 7(3-4):201–221, 1978.

Christos H Papadimitriou and Tim Roughgarden. Computing correlated equilibria in multi-player games. *Journal of the ACM*, 55(3):14, 2008.

Sheldon M Ross. Goofspiel—the game of pure strategy. *Journal of Applied Probability*, 8(3): 621–625, 1971.

Finnegan Southey, Michael Bowling, Bryce Larson, Carmelo Piccione, Neil Burch, Darse Billings, and Chris Rayner. Bayes' bluff: Opponent modelling in poker. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, July 2005.

Bernhard von Stengel. Efficient computation of behavior strategies. *Games and Economic Behavior*, 14(2):220–246, 1996.

Bernhard von Stengel and Françoise Forges. Extensive-form correlated equilibrium: Definition and computational complexity. *Mathematics of Operations Research*, 33(4):1002–1022, 2008.

Brian Hu Zhang and Tuomas Sandholm. Team correlated equilibria in zero-sum extensive-form games via tree decompositions. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2022.

Brian Hu Zhang, Gabriele Farina, and Tuomas Sandholm. Team belief dag form: A concise representation for team-correlated game-theoretic decision making. *arXiv preprint arXiv:2202.00789*, 2022.

Youzhi Zhang, Bo An, and Jakub Černỳ. Computing ex ante coordinated team-maxmin equilibria in zero-sum multiplayer extensive-form games. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

A 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 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 A.1. 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 $f(k)\,\text{poly}(|x|)$, where $|x|$ is the bit length of x and $f : \mathbb{N} \to \mathbb{N}$ is an *arbitrary function*.

The k-CLIQUE problem[10] 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[1]*, and is implied by the exponential time hypothesis (Chen et al., 2005). We now show that this conjecture implies lower bounds on the complexity of representing the polytopes Ξ^{EFCCE} and Ξ^{EFCE}.

Theorem A.2. *Assuming FPT $\neq$ W[1], there is no FPT algorithm for linear optimization over Ξ^{EFCCE} or Ξ^{EFCE} parameterized by information complexity, even in two-player games with constant branching factor.*

Technically speaking, Theorem A.2 does not establish parameterized hardness of computing welfare-optimal EFCCEs or EFCEs, as there could hypothetically be a method for doing so that does not need to construct the correlation plan polytope. However, we know of no technique that circumvents this need. Therefore, Theorem A.2 is a lower bound that applies to all known techniques for computing welfare-optimal EFCCEs and EFCEs.

B Two-Sided Column Generation Approach

In this section, we propose a scalable iterative method which exploits the particular combinatorial structure of the polytope of correlation plans. Such polytope is typically low-dimensional, but may have an exponential number of facets. Nonetheless, its geometrical structure can be exploited to compute solutions even for large game instances. In the analysis, we exploit a particular polytope introduced by von Stengel & Forges (2008). In line with previous literature on such polytope, we describe our framework for games with two players. In principle one could extend the notion of polytope introduced by von Stengel & Forges (2008) to the case of an arbitrary number of players, but that is beyond the scope of the present work. The *two-sided column generation approach* which we describe is capable of computing *optimal* NFCCE, EFCCE, EFCE in any two-player general-sum imperfect-information extensive-form game.

B.1 Correlation as a Linear Program over Ξ

In this section, we write out the LP in Theorem 3.2, since we will need it when defining column generation. Given a solution concept c, the problem of computing optimal equilibrium point with respect to an arbitrary objective $\boldsymbol{g}^\top \boldsymbol{\xi}$ amounts to solving the following optimization problem

$$\begin{cases} \operatorname{argmax}_{\boldsymbol{\xi}} \quad \boldsymbol{g}^\top \boldsymbol{\xi}, \quad \text{subject to:} \\ \text{1 Incentive constraints} \\ \text{2 } \boldsymbol{\xi} \in \Xi^c \end{cases} . \tag{$*$}$$

Then, the family of constraints 1 is defined depending on the solution concept c. In particular, for all $\tau \in \mathcal{T}^c$ belonging to any player i, we have the constraint

$$\min_{\boldsymbol{\mu}'_i \in Q[\tau]} \sum_{z \succeq \mathcal{H}[\tau]} u_i(z)\, p(z) (\xi[z^\tau]\, \mu'_i[z] - \xi[z]) \leq 0.$$

These constraints can be converted to a system of linear constraints by dualizing, yielding a linear program. Therefore, we have:

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

B.2 SEMI-RANDOMIZED CORRELATION PLANS

We introduce the following notation for two-player games, which was also used by Farina et al. (2021a): we write $\sigma_1 \bowtie \sigma_2$ to denote a relevant sequence pair $(\sigma_1, \sigma_2) \in \Sigma$. For a P1 sequence σ and P2 infoset I, we write $\sigma \bowtie I$ if $\sigma \bowtie Ia$ for each action a. Similarly, for a P1 infoset I and P2 infoset σ, we write $I \bowtie \sigma$ if $Ia \bowtie \sigma$.

Now, we introduce the strategy representation which we employ in our algorithm. We observe that variables in LP ($*$) (Section 3.1) belong to the convex polytope Ξ, but that polytope cannot be compactly represented in general. Therefore, we tackle LP ($*$) by adopting the notion of *semi-randomized correlation plan* proposed by Farina et al. (2021a). For completeness, we show how *semi-randomized correlation plan* can be derived from the *von Stengel-Forges polytope* (von Stengel & Forges, 2008) representing interlaced sequence-form "probability mass conservation" constraints for the two players.

Definition B.1. The *von Stengel-Forges polytope*, denoted $\mathcal{V}$, is the polytope of all vectors $\boldsymbol{\zeta} \in \mathbb{R}^{\Sigma}_{\geq 0}$ (*i.e.*, indexed over relevant sequence pairs) such that: Ⓐ $\zeta[\varnothing_1, \varnothing_2] = 1$; Ⓑ $\sum_{a \in A_I} \zeta[Ia, \sigma_2] = \zeta[\sigma_1(I), \sigma_2] \;\; \forall I \bowtie \sigma_2 \in \mathcal{I}_1 \times \Sigma_2$; and Ⓒ $\sum_{a \in A_I} \zeta[\sigma_1, Ia] = \zeta[\sigma_1, \sigma_2(I)] \;\; \forall I \bowtie \sigma_2 \in \mathcal{I}_2 \times \Sigma_1$.

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 & Forges, 2008).

Finally, a *semi-randomized correlation plan* is composed of a deterministic sequence form strategy for one player, while the other player independently plays a mixed strategy.

Definition B.2 ((Farina et al., 2021a))**.** The sets of semi-randomized correlation plans are

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

Given $i \in \{1,2\}$, a point $\boldsymbol{\xi} \in \Xi_i^{\mathrm{sr}}$ 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. (2021a) that $\Xi = \mathrm{co}(\Xi_1^{\mathrm{sr}}) = \mathrm{co}(\Xi_2^{\mathrm{sr}}) = \mathrm{co}(\Xi_1^{\mathrm{sr}} \cup \Xi_2^{\mathrm{sr}})$.

B.3 COMPUTING DEVIATIONS

Given a solution concept $c \in \{\text{NFCCE}, \text{EFCCE}, \text{EFCE}\}$, and a correlation plan $\boldsymbol{\xi}$, we can compute the maximum possible deviation (across all players) by exploiting constraints ① of LP ($*$). In particular, given a trigger $\tau \in \mathcal{T}^c$, we observe that any constraint of type ① can be written as $\boldsymbol{\xi}^\top \boldsymbol{A}_\tau \boldsymbol{\mu} - \boldsymbol{b}_\tau^\top \boldsymbol{\xi} \leq 0$, where $\boldsymbol{\mu} \in Q[\tau]$ is an arbitrary mixed sequence-form deviation for trigger τ, and $\boldsymbol{A}_\tau, \boldsymbol{b}_\tau$ are suitably defined sparse matrices/vectors that only depend on τ. Since $Q[\tau]$ is the set of sequence-form strategies following τ, we can compactly represent it as $Q[\tau] := \{\boldsymbol{\mu} : \boldsymbol{F}_\tau \boldsymbol{\mu} = \boldsymbol{f}_\tau, \boldsymbol{\mu} \geq 0\}$, for some suitable choice of $\boldsymbol{F}_\tau$ and $\boldsymbol{f}_\tau$ (see, *e.g.*, (Koller et al., 1996; von Stengel, 1996)). Then, we can compute the maximum deviation through the following LP.

$$\begin{cases} \max_{\boldsymbol{\lambda}, (\tilde{\boldsymbol{\mu}})_{\mathcal{T}^c}} & \sum_{\tau \in \mathcal{T}^c} \left(\boldsymbol{\xi}^\top \boldsymbol{A}_\tau \tilde{\boldsymbol{\mu}}_\tau - \lambda[\tau] \boldsymbol{b}_\tau^\top \boldsymbol{\xi} \right) \\ \text{s.t.} & \boldsymbol{F}_\tau \tilde{\boldsymbol{\mu}}_\tau = \lambda[\tau] \boldsymbol{f}_\tau \quad \forall \tau \in \mathcal{T}^c \\ & \mathbf{1}^\top \boldsymbol{\lambda} = 1 \\ & \lambda[\tau] \geq 0, \tilde{\boldsymbol{\mu}}_\tau \geq 0 \; \forall \tau \in \mathcal{T}^c \end{cases},$$

Then, by taking the dual of the above LP we obtain the following.

$$\begin{cases} \min_{(\boldsymbol{v}_\tau)_{\mathcal{T}^c}, u} & u \\ \text{s.t.} & \boldsymbol{F}_\tau^\top \boldsymbol{v}_\tau - \boldsymbol{A}_\tau^\top \boldsymbol{\xi} \geq 0 \quad \forall \tau \in \mathcal{T}^c \\ & u - \boldsymbol{f}_\tau^\top \boldsymbol{v}_\tau + \boldsymbol{\xi}^\top \boldsymbol{b}_\tau \geq 0 \quad \forall \tau \in \mathcal{T}^c \\ & \boldsymbol{v}, u \text{ free} \end{cases}. \tag{D}$$

By strong duality, the value of LP (D) is the same as the value of the primal problem (*i.e.*, the maximum 'deviation benefit' across all players and all possible deviations for triggers in $\mathcal{T}^c$).

B.4 Correlation-Plan Decomposition and Iterative Framework

We say that a correlation plan $\boldsymbol{\xi}$ 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 semi-randomized correlation plan corresponds to a distribution of play where one team member plays a pure sequence-form strategy, while the other plays a mixed sequence-form strategy, $\boldsymbol{\xi} \in \Xi_i^{\mathrm{sr}}$ is guaranteed to be a product correlation plan for any i (see (Farina et al., 2021a, Lemma 3)). Given $\boldsymbol{\xi} \in \Xi_1^{\mathrm{sr}} \cup \Xi_2^{\mathrm{sr}}$, let $\boldsymbol{m}_{\boldsymbol{\xi},1} \in \mathbb{R}^{\Sigma_1}$ be the vector such that $\boldsymbol{m}_{\boldsymbol{\xi},1} := \xi[\cdot, \varnothing]$, and let $\boldsymbol{m}_{\boldsymbol{\xi},2}$ be defined analogously. Moreover, for any $\boldsymbol{x}_1 \in \mathbb{R}^{\Sigma_1}, \boldsymbol{x}_2 \in \mathbb{R}^{\Sigma_2}$, let $\boldsymbol{\xi} := \boldsymbol{x}_1 \otimes \boldsymbol{x}_2$ be the correlation plan given by $\xi[\sigma_1, \sigma_2] = x_1[\sigma_1]\, x_2[\sigma_2]$. Then, we can decompose any correlation plan $\boldsymbol{\xi} \in \Xi_1^{\mathrm{sr}} \cup \Xi_2^{\mathrm{sr}}$ as

$$\boldsymbol{\xi} = \lambda\, \boldsymbol{\mu}_1 \otimes \boldsymbol{m}_{\boldsymbol{\xi},2} + (1-\lambda)\, \boldsymbol{m}_{\boldsymbol{\xi},1} \otimes \boldsymbol{\mu}_2,$$

for some appropriate choice of $\lambda \in [0,1]$, and mixed strategies $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ for Player 1 and Player 2, respectively. Moreover, given $\boldsymbol{\xi} \in \Xi_1^{\mathrm{sr}} \cup \Xi_2^{\mathrm{sr}}$, $\lambda\, \boldsymbol{\mu}_1 \otimes \boldsymbol{m}_{\boldsymbol{\xi},2} + (1-\lambda)\, \boldsymbol{m}_{\boldsymbol{\xi},1} \otimes \boldsymbol{\mu}_2 \in \mathrm{co}(\Xi_1^{\mathrm{sr}} \cup \Xi_2^{\mathrm{sr}})$ for any $\lambda \geq 0$ and well-formed sequence-form strategies $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$.

We define the *support* $S \subseteq \Xi_1^{\mathrm{sr}} \cup \Xi_2^{\mathrm{sr}}$ to be an arbitrary subset of semi-randomized correlation plans. Then, we consider the following optimization problem, which we call the *master LP*.

$$\begin{cases} \max \quad \boldsymbol{g}^\top \boldsymbol{\xi} \\ \quad ① \; \boldsymbol{F}_\tau^\top \boldsymbol{v}_\tau - \boldsymbol{A}_\tau^\top \boldsymbol{\xi} \geq 0 & \forall \tau \in \mathcal{T}^c \\ \quad ② \; u - \boldsymbol{f}_\tau^\top \boldsymbol{v}_\tau + \boldsymbol{\xi}^\top \boldsymbol{b}_\tau \geq 0 & \forall \tau \in \mathcal{T}^c \\ \quad ③ \; \boldsymbol{\xi} = \sum_{\tilde{\boldsymbol{\xi}} \in S} \left(\tilde{\boldsymbol{\mu}}_{\tilde{\boldsymbol{\xi}},1} \otimes \boldsymbol{m}_{\tilde{\boldsymbol{\xi}},2} + \boldsymbol{m}_{\tilde{\boldsymbol{\xi}},1} \otimes \tilde{\boldsymbol{\mu}}_{\tilde{\boldsymbol{\xi}},2} \right) \\ \quad ④ \; \boldsymbol{F}_i \tilde{\boldsymbol{\mu}}_{\tilde{\boldsymbol{\xi}},i} = \lambda_i[\tilde{\boldsymbol{\xi}}]\, \boldsymbol{f}_i & \forall i \in \{1,2\}, \tilde{\boldsymbol{\xi}} \in S \\ \quad ⑤ \; u \leq 0 \\ \quad ⑥ \; \sum_{i \in \{1,2\}} \sum_{\tilde{\boldsymbol{\xi}} \in S} \lambda_i[\tilde{\boldsymbol{\xi}}] = 1 \\ \quad ⑦ \; \boldsymbol{\lambda}_i \geq 0 \,\forall i, \;\; \boldsymbol{\mu}_{\tilde{\boldsymbol{\xi}},i} \geq 0 \,\forall (i, \tilde{\boldsymbol{\xi}}), \;\; \boldsymbol{v}_\tau \text{ free } \; \forall \tau \end{cases} \tag{M}$$

As we argued in Section B.2, we could replace constraint ② of LP ($*$) by forcing all $\boldsymbol{\xi}$ to belong to $\mathrm{co}(\Xi_1^{\mathrm{sr}} \cup \Xi_2^{\mathrm{sr}})$. Then, if we set $S = \mathrm{co}(\Xi_1^{\mathrm{sr}} \cup \Xi_2^{\mathrm{sr}})$, the above LP would have value equal to the optimal value of LP ($*$). To see that, we make the following observations:

1. each correlation plan $\boldsymbol{\xi} \in S$ in the support is represented through the decomposition which we defined above, where we perform the change of variables $\tilde{\boldsymbol{\mu}}_{\tilde{\boldsymbol{\xi}},i} := \lambda_i[\tilde{\boldsymbol{\xi}}] \boldsymbol{\mu}_{\tilde{\boldsymbol{\xi}},i}$ for each i;
2. from the analysis of LP (D), we have that constraint ⑤ (together with ① and ②), implies that the maximum deviation benefit is non-positive, that is, family of incentive constraints ① in LP ($*$) is satisfied;
3. constraints ④ and ⑦ imply that $\tilde{\boldsymbol{\mu}}_{\boldsymbol{\xi},i}$ are well-defined mixed strategies rescaled appropriately according to $\boldsymbol{\lambda}$ coefficients;
4. constraints ⑥ and ⑦ imply that $\boldsymbol{\lambda}$ is a point in the $2|S|$-dimensional simplex.

If we could afford to set $S = \mathrm{co}(\Xi_1^{\mathrm{sr}} \cup \Xi_2^{\mathrm{sr}})$, finding an optimal NFCCE, EFCCE, EFCE for an arbitrary objective $\boldsymbol{g}$ would amount to solving LP (M) once. However, the size of $\Xi_1^{\mathrm{sr}} \cup \Xi_2^{\mathrm{sr}}$ is oftentimes prohibitively large, since it has size exponential in the size of the game. Therefore, we follow the approach by Ford & Fulkerson (1958) and generate the support S iteratively. We say that the column-generation algorithm of Farina et al. (2021a) 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 $\boldsymbol{\lambda}$ 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. (2021a). In particular, we remark that $\tilde{\boldsymbol{\mu}}_{\tilde{\boldsymbol{\xi}},1}, \tilde{\boldsymbol{\mu}}_{\tilde{\boldsymbol{\xi}},1}$ are continuous variables in LP (M). Therefore, each player is allowed to re-optimize their mixed strategies $\boldsymbol{\mu}_{\tilde{\boldsymbol{\xi}},i}$, enabling them to reach a richer set of correlation plans starting from the same support set.

Algorithm 1 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, we solve the master LP (M) with the current support S. Each time we solve the master LP, we keep track of the resulting primal and dual variables. In particular, when solving (M), the algorithm keeps track of the current solution $\boldsymbol{\xi}^{\text{c}}$ (*i.e.*, the correlation plan corresponding to the optimal decomposition), and the vector of dual variables $\boldsymbol{\gamma}^{\text{c}}$. We observe that, in order to obtain dual variables, we do not need to solve the dual LP of (M), since modern LP solvers already keep track of dual values. New correlation plans to be added to S are determined through the function Pricer(), 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). Such correlation plan can be computed from the solution to the dual of the master LP, which we denote by $\boldsymbol{\gamma}^{\text{c}}$. Let $\boldsymbol{\gamma}^{\text{c}}_{1,\tau}$ be the sub-vector of the dual variables corresponding to constraints 1 and trigger τ of (M), and $\boldsymbol{\gamma}^{\text{c}}_{2,\tau}$ be the sub-vector of dual variables corresponding to constraints 2 and trigger τ. Then, by letting $\boldsymbol{w} := \sum_{\tau\in\mathcal{T}^c}\left(A_\tau\boldsymbol{\gamma}^{\text{c}}_{1,\tau} - \boldsymbol{b}_\tau\boldsymbol{\gamma}^{\text{c}}_{2,\tau}\right)$, the pricing problem amounts to solving $\max_{\boldsymbol{\xi}\in\Xi}\left(\boldsymbol{g}^\top\boldsymbol{\xi}^{\text{c}} - \boldsymbol{\xi}^\top\boldsymbol{w}\right)$. We know that $\Xi = \text{co}(\Xi_i^{\text{sr}})$, for any i. Therefore, by linearity of the objective and by convexity, we have $\max_{\boldsymbol{\xi}\in\Xi}\left(\boldsymbol{g}^\top\boldsymbol{\xi}^{\text{c}} - \boldsymbol{\xi}^\top\boldsymbol{w}\right) = \max_{\boldsymbol{\xi}\in\Xi_i^{\text{sr}}}\left(\boldsymbol{g}^\top\boldsymbol{\xi}^{\text{c}} - \boldsymbol{\xi}^\top\boldsymbol{w}\right)$. The right hand side 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 $\hat{\boldsymbol{\xi}}$ a correlation plan attaining such value (see Line 9). When the objective value δ is non-positive, there is no correlation plan which can be added to the support S which would result in an increase in the value of the master LP. When that happens, we know that the optimal solution to the master LP is also an optimal solution to the original problem ($*$) and the main loop terminates.

ALGORITHM 1: Two-Sided Column Generation

1 **input:** game Γ, concept $c \in \{\text{NFCCE}, \text{EFCCE}, \text{EFCE}\}$, objective $\boldsymbol{g}$
2 **function** ComputeOpt()
3 **initialization phase:** initialize S
4 $\boldsymbol{\xi}^{\text{c}}, \boldsymbol{\gamma}^{\text{c}} \leftarrow$ solve master LP (M) ▷ *Solving (M) yields primal solution $\boldsymbol{\xi}^{\text{c}}$ and dual variables $\boldsymbol{\gamma}^{\text{c}}$.*
5 **if** (M) is not feasible **then** go to Line (13)
6 **while** within computational budget **do**
7 $\boldsymbol{\xi}^{\text{c}}, \boldsymbol{\gamma}^{\text{c}} \leftarrow$ solve master LP (M)
8 **if** (M) is feasible **then**
9 $\delta, \hat{\boldsymbol{\xi}} \leftarrow$ Pricer($\boldsymbol{\gamma}^{\text{c}}$) ▷ *$\delta$: maximum deviation benefit, $\hat{\boldsymbol{\xi}}$: new correlation plan to be added.*
10 **if** $\delta <$ Tolerance **then return** $\boldsymbol{\xi}^{\text{c}}$
11 $S \leftarrow S \cup \hat{\boldsymbol{\xi}}$, $\boldsymbol{m}_{\hat{\boldsymbol{\xi}},\cdot} \leftarrow$ marginalize $\hat{\boldsymbol{\xi}}$
12 **else** ▷ *The current support S is insufficient to generate any equilibrium*
13 Drop constraint 5 and substitute objective with $\min u$. Then solve (M) to obtain $\boldsymbol{\gamma}^{\text{c}}$
14 $\_, \hat{\boldsymbol{\xi}} \leftarrow$ Pricer($\boldsymbol{\gamma}^{\text{c}}$)
15 $S \leftarrow S \cup \hat{\boldsymbol{\xi}}$, $\boldsymbol{m}_{\hat{\boldsymbol{\xi}},\cdot} \leftarrow$ marginalize $\hat{\boldsymbol{\xi}}$

C Further Discussion of Prior Works

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 & von Stengel, 2008) via a variation of the *Ellipsoid Against Hope* algorithm (Papadimitriou & Roughgarden, 2008; Jiang & Leyton-Brown, 2015). 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., 2020; Farina et al., 2021b). Furthermore, NFCCE is the equilibrium notion that gets satisfied when all players in a game play according to any regret minimizer. Using regret minimizers to play large multi-player games has already led to superhuman practical performance in multi-player poker (Brown & Sandholm, 2019). As stated above, however, computing optimal equilibria is much harder.

This paper does not address *normal-form correlated equilibrium* (NFCE). In NFCE, the mediator tells each player her entire pure strategy π_i at the start of the game, at which point the player may choose to deviate. Unlike the other three concepts, because NFCE involves the whole recommendation being given upfront, it is impossible to express it in the language of trigger agents we have introduced. Indeed, it is known computing optimal NFCEs is NP-hard even in two-player games without chance nodes (von Stengel & Forges, 2008), making it a distinctly difficult problem that is out of the scope of this paper.

Zhang & Sandholm (2022) introduced a framework based on *tree decompostions* for solving zero-sum team games. The correlation DAG can also be viewed as the polytope generated by a certain tree decomposition of the constraint system. Namely, there is a tree decomposition of a (nonlinear) constraint system defining pure correlation plans such that the bags of the tree decomposition are exactly the sets $P \cup P'$ where P is the set of trigger histories in a public state, and P' is the set of all their descendants. Correctness of the decomposition, that is, Theorem 4.4, would then follow immediately from the junction tree theorem. To avoid introducing machinery and notation needlessly, we will not elaborate on this view for this paper and instead focus on the DAG representation.

D Example

An example of the correlation DAG can be found in Figure 1. The figure uses several notational simplifications for cleanliness:

- For NFCCE: If a non-terminal node contains $h^\perp$, it is also understood to contain $h^{\varnothing_1}$ and $h^{\varnothing_2}$. If an observation node contains $ha^\perp$ for a node h of player i, it is also understood to contain $h\tilde{a}^{\varnothing_i}$ for every action $\tilde{a} \in A_h$.
- For EFCCE: If any non-terminal node contains h^I, then it is also understood to contain $h^{I'}$ for every $I' \preceq I$. If a decision node contains $h^\perp$ for $h \in I$, it is also understood to contain h^I. If an observation node contains $ha^\perp$ for $h \in I$, it is also understood to contain $h\tilde{a}^I$ for every $\tilde{a} \in A_h$.
- In this particular game, a player (for NFCCE) or a layer (for EFCCE or EFCE) uniquely identifies a trigger. We therefore use $1, 2$ in NFCCE to identify he player triggered, and $1, 2, 3$ in EFCCE and EFCE to identify triggers by the layer on which they occurred.

In the example game, the three DAGs are nearly identical; in particular, the first four layers of nodes are identical. The only difference between the NFCCE and EFCCE DAGs is that the latter contains an explicit terminal trigger history for the reach probability of P^3, which is an invalid deviation in NFCCE. The only difference between the EFCCE and EFCE DAGs is the definition of trigger history: h^ℓ has a different meaning in each. In EFCCE, h^ℓ means that the player at level ℓ deviated *before* seeing the recommendation; in EFCE, it means that the player at level ℓ deviated *after* being recommended the move that did not lead to h. The three DAGs will grow more dissimilar in games with large branching factor or large depth.

E Proofs

E.1 Theorem 4.4 (Correctness of Correlation DAG Construction)

Theorem 4.4 (Correctness). *Let Γ be a game, and $\mathcal{D}^c$ its correlation DAG for any of the three concepts c. For a sequence-form mixed strategy $\boldsymbol{\mu} \in Q_{\mathcal{D}^c}$, define the vector $\boldsymbol{\xi}^{\boldsymbol{\mu}} \in [0,1]^{\Sigma^c}$ by $\xi^{\boldsymbol{\mu}}[\boldsymbol{\sigma}(z^\tau)] = \mu[\{z^\tau\}]$. Then we have $\Xi^c = \{\boldsymbol{\xi}^{\boldsymbol{\mu}} : \boldsymbol{\mu} \in Q_{\mathcal{D}^c}\}$. That is, the correlation plan polytope Ξ^c is a projection of the set $Q_{\mathcal{D}^c}$ of sequence-form mixed strategies in $\mathcal{D}^c$.*

Proof. The proof is very similar to the correctness proof of Zhang et al. (2022). We prove that the vertices of $Q_{\mathcal{D}^c}$ project onto all the vertices of Ξ^c, which is sufficient because of convexity. Let $z^\tau \in \mathcal{H}^c_\perp$ be an arbitrary terminal trigger history.

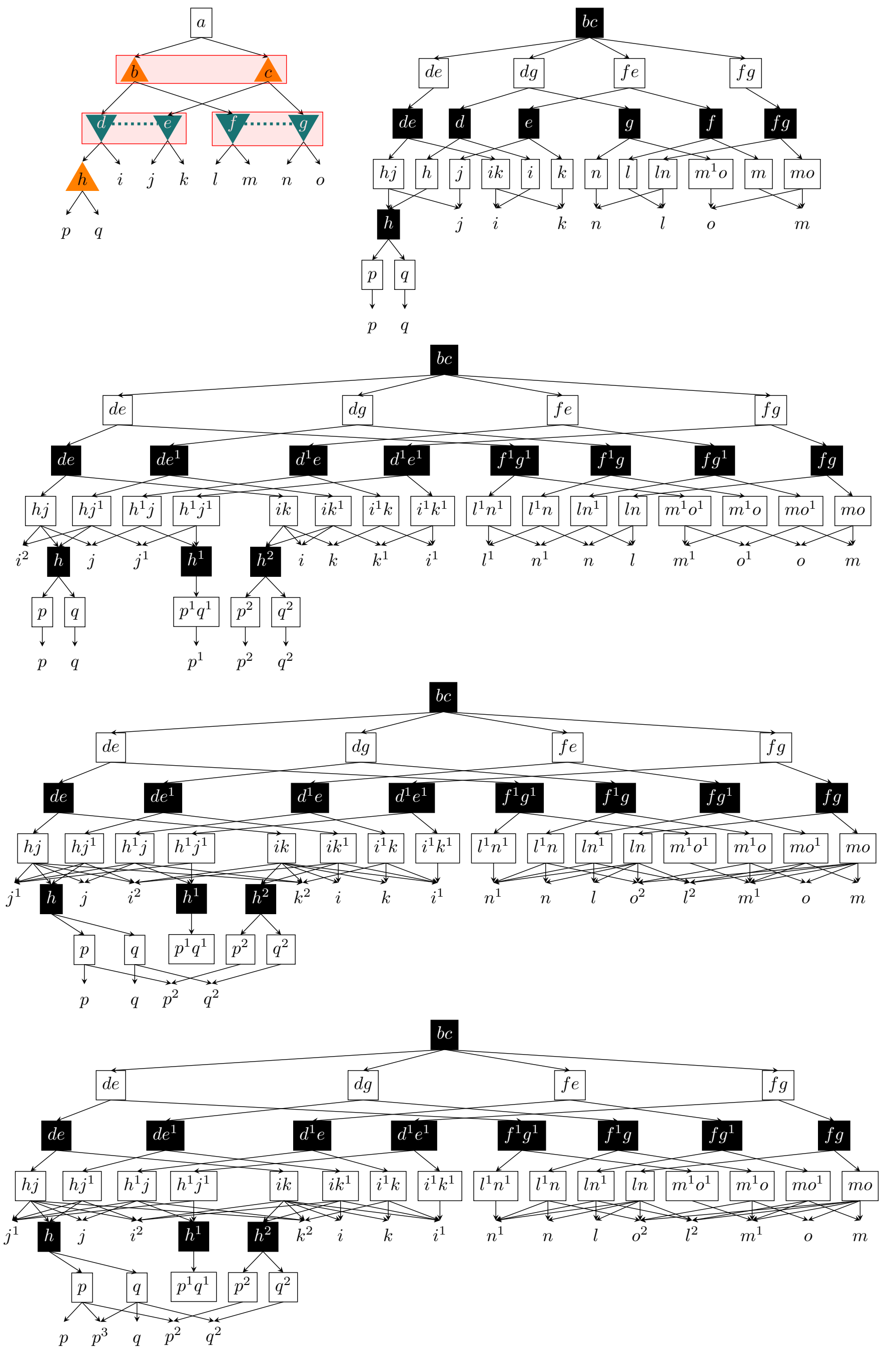


Figure 1: An example two-player game (top left), its team DAG (top right; Zhang et al. 2022), and its EFCE DAG. The NFCCE and EFCCE DAGs are very similar, and can be found in Appendix D. Square nodes are nature, and triangle nodes are the nodes for the two players. Dotted lines connect nodes in the same information set. In all the DAGs, nodes with black text on white background are observation nodes, and nodes with white text on black background are decision nodes. Red shaded regions connect nodes in the same public states. For each joint terminal sequence, an arbitrary representative was selected. Nodes $h^\perp$ are written without the $\perp$ to reduce notational clutter. Triggers are identified by the layer on which they occurred. The top three layers of all DAGs are also omitted to reduce clutter; they are $\boxed{a} \to \boxed{a} \to \boxed{bc} \to \boxed{bc}$.

ALGORITHM 2: Constructing the correlation DAG, and using it to write a constraint system for Ξ^c

input: game Γ, concept $c \in \{\text{NFCCE}, \text{EFCCE}, \text{EFCE}\}$
function MAKECONSTRAINTSYSTEM()
  $s_0 \leftarrow$ MAKEOBSERVATIONNODE($\{h_0\}$)
  for each *node s in the decision problem* **do** introduce variable $\mu[s] \geq 0$
  add constraint $\mu[s_0] = 1$
  for each *decision node s* **do** add constraint $\sum_{p \text{ parent of } s} \mu[p] = \sum_{a \in A_s} \mu[sa]$
  for each *key $\boldsymbol{\sigma}$ in* Terminal **do**
    $(s, z^\tau) \leftarrow$ Terminal[$\boldsymbol{\sigma}$]
    $\xi[\boldsymbol{\sigma}] \leftarrow \mu[s]$
function MAKEOBSERVATIONNODE(*set of nodes O*)
  $s \leftarrow$ new observation node
  if $c =$ *NFCCE* **then** $\hat{O} \leftarrow \{h_0\} \cup \{h_0^{\varnothing_i} : i \in [n]\}$ if $O = \{h_0\}$, else O
  if $c =$ *EFCCE* **then** $\hat{O} \leftarrow O \cup \{h^I : h \in I, h^\perp \in O\}$
  if $c =$ *EFCE* **then** $\hat{O} \leftarrow O \cup \{h^{Ia}\tilde{a} : h \in I, ha^\perp \in O, \tilde{a} \neq a \in A_h\}$
  for each $z^\tau \in \hat{O} \cap \mathcal{Z}^c$ **do**
    if Terminal[$\boldsymbol{\sigma}(z^\tau)$] *is not defined* **then**
      $s' \leftarrow$ new terminal node
      Terminal[$\boldsymbol{\sigma}(z^\tau)$] $\leftarrow (s', z^\tau)$
    $(s', \widetilde{z^\tau}) \leftarrow$ Terminal[$\boldsymbol{\sigma}(z^\tau)$]
    ▷ *Only one representative for each terminal joint sequence $\boldsymbol{\sigma}$ is needed. If z^τ is not it, do nothing.*
    if $z^\tau = \widetilde{z^\tau}$ **then** add edge $s \to s'$
  for each *public state P intersecting* $\{h : h^\tau \in \hat{O}\}$ **do**
    $\hat{O}P \leftarrow \{h^\tau \in \hat{O} : h \in P\}$
    add edge $s \to$ MAKEDECISIONNODE($\hat{O}P$)
  return s
function MAKEDECISIONNODE(*belief B*)
  if *there is already a decision node s with belief B* **then return** s
  $s \leftarrow$ new decision node with belief B
  $\mathcal{I} \leftarrow \{I : h^\tau \in B \text{ active}, h \in I\}$
  for each *prescription* $\boldsymbol{a} \in \bigtimes_{I \in \mathcal{I}} I$ **do**
    $B\boldsymbol{a} \leftarrow \{ha[I]^\tau : h \in I, h^\tau \in B \text{ active}\} \cup \{h\tilde{a}^\tau : h^\tau \in B \text{ inactive}, \tilde{a} \in A_h\}$
    add edge $s \to$ MAKEOBSERVATIONNODE($B\boldsymbol{a}$)
  return s

($\Leftarrow$) Let $\boldsymbol{\pi}$ be a pure strategy profile in Γ. Let $\boldsymbol{\pi}'$ be the pure strategy in $\mathcal{D}^c$ that always chooses the prescription $\boldsymbol{a}$ according to $\boldsymbol{\pi}$.

Suppose $\pi[z^\tau] = 1$. To show that $\pi'[\{z^\tau\}] = 1$, we need to demonstrate that there exists a path from the root $\{h_0\} \in \mathcal{S}^c$ to $\{z^\tau\}$ such that $\boldsymbol{\pi}'$ gives every prescription along this path. By a simple induction, the path through $\mathcal{D}^c$ generated by following prescriptions at belief nodes and always picking the public state observation containing an ancestor of z can be shown to end at $\{z^\tau\}$, which is what we need.

Conversely, suppose $\pi'[\{z^\tau\}] = 1$. Then, again by induction, every belief B on the active path to $\{z^\tau\}$ in $\mathcal{D}^c$ contains a trigger history $\tilde{h}^{\tilde{\tau}}$ on the path to z^τ. Therefore, in particular, for every infoset I on the path to z^τ in $\mathcal{H}$, I must be active in some belief B on the path to $\{z^\tau\}$. Thus, by definition, $\boldsymbol{\pi}$ must play the action at I consistent with z^τ. Thus, we have $\pi[z^\tau] = 1$.

($\Rightarrow$) Let $\boldsymbol{\pi}'$ be a pure strategy in $\mathcal{D}^c$. At each public state P, $\boldsymbol{\pi}'$ must reach (at most) one unique belief B. Let $\boldsymbol{a}$ be the prescription of $\boldsymbol{\pi}'$ at B. Define the pure strategy $\boldsymbol{\pi}$ for Γ to play the actions consistent with $\boldsymbol{a}$ at active infosets I in B, and arbitrarily otherwise.

Suppose $\pi'[\{z^\tau\}] = 1$. That is, there is a path to $\{z^\tau\}$ at which $\boldsymbol{\pi}'$ plays every prescription. Then, in particular, every belief B along this path was used to construct $\boldsymbol{\pi}$. Thus, π must also play every action on the path to z^τ; that is, $\pi[z^\tau] = 1$.

Conversely, suppose $\pi[z^\tau] = 1$. Then, again by induction, the belief B in every public state $P \preceq z$ must contain an ancestor of z^τ, and therefore must be played to. Thus, π' plays to $\{z^\tau\}$. This completes the proof. □

E.2 THEOREM 4.6 (CORRELATION DAG SIZE)

Theorem 4.6. $\left|\mathcal{E}^{\text{NFCCE}}\right| = O^*\big((b+1)^k\big)$, $\left|\mathcal{E}^{\text{EFCCE}}\right| = O^*\big((b+d-2)^k\big)$, *and* $\left|\mathcal{E}^{\text{EFCE}}\right| = O^*\big((bd)^k\big)$.

Proof. First, we observe that $|\mathcal{E}^c| = O^*(|\mathcal{S}^c|)$, since every observation point in $\mathcal{S}$ has exactly one parent, and at most as many children as there are public states. Thus, we will bound $|\mathcal{S}^c|$.

NFCCE: To specify a belief-prescription pair $B\boldsymbol{a}$ within a given public state P, it suffices, for each player state in P, to specify, for each private state $\bar\sigma$ at B, whether the player (1) does not play to $\bar\sigma$ at all, or (2) plays to $\bar\sigma$ and chooses one of the b actions available therein. There are at most $(b+1)^k$ such choices, which is what we needed to show.

EFCCE: For each of the k private states $\bar\sigma$ at P, we need to specify whether the player played to reach $\bar\sigma$ and then played one of the (at most) b actions available there, or she deviated at one of the (at most) $d-2$ infosets $I \prec \bar\sigma$. There are at most $b+d-2$ ways to do this.

EFCE: For EFCE, we need to additionally specify which action was recommended at the deviation point, of which there are at most b possibiliities, for a total of $b(d-2)+b = b(d-1)$ options.

□

E.3 THEOREM 4.8 (GAMES WITH PUBLIC ACTIONS)

Theorem 4.8. *In games with public actions, the correlation DAG construction can be modified to achieve* $\left|\mathcal{E}^{\text{NFCCE}}\right| = O^*\big(3^k\big)$ *and* $\left|\mathcal{E}^{\text{EFCCE}}\right| = O^*\big(d^k\big)$.

Proof. Zhang et al. (2022) 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 of Γ. □

E.4 THEOREM A.2 (W[1]-HARDNESS)

Theorem A.2. *Assuming FPT $\neq$ W[1], there is no FPT algorithm for linear optimization over* Ξ^{EFCCE} *or* Ξ^{EFCE} *parameterized by information complexity, even in two-player games with constant branching factor.*

Proof. We reduce from k-CLIQUE. Let $G = (V, E)$ be a graph with n nodes (identified with the positive integers $[n]$), and construct the following two-player game Γ:

- 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).

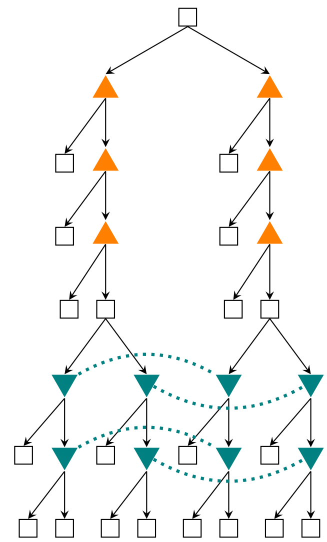


Figure 2: The game used in the proof of Theorem A.2, for $n = k = 2$. P1 is ▲, P2 is ▼, and square nodes are chance or terminal. Dotted lines connect nodes in the same infoset.

- Finally, $(j_1, j_2, n + 1)$ is a terminal node for all j_1, j_2.

Since we are only concerned with representing the correlation plan polytope, we do not need to specify utilities or chance probabilities.

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 $m^{\boldsymbol{\xi}}[j_1, v_1, j_2, v_2]$ 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 $v_1, v_2 \in [n]$, $m^{\boldsymbol{\xi}}[j_1, v_1, j_2, v_2]$ is a linear function of both the correlation plan spaces Ξ^{EFCCE} and Ξ^{EFCE}: for $\boldsymbol{\xi} \in \Xi^{\text{EFCCE}}$, it is exposed as $\xi[(j_1, j_2, v_2, \mathsf{E})^{(j_1,v_1+1)}] - \xi[(j_1, j_2, v_2, \mathsf{E})^{(j_1,v_1)}]$; for $\boldsymbol{\xi} \in \Xi^{\text{EFCE}}$, it is exposed as $\xi[(j_1, j_2, v_2, \mathsf{E})^{(j_1,v_1,\mathsf{E})}]$. (For Ξ^{NFCCE}, $\boldsymbol{m}^{\boldsymbol{\xi}}$ is not a linear function of $\boldsymbol{\xi}$, so, as expected, the argument fails here.)

Let $M = \{\boldsymbol{m}^{\boldsymbol{\xi}} : \boldsymbol{\xi} \in \Xi^c\} \subseteq [0, 1]^{[k]\times[n]\times[k]\times[n]}$ be the polytope of vectors $\boldsymbol{m}$ corresponding to correlated strategies. At this point, since M does not depend on the notion of equilibrium, we have no more need to distinguish between EFCCE and EFCE. It suffices to show that linear optimization on M can decide k-CLIQUE. First, we characterize the vertices of M. A vertex of M is characterized by, for each player $i \in \{1, 2\}$ and each $j \in [k]$, picking at most one vertex $v_{i,j} \in [n]$, and constructing $\boldsymbol{m}$ by setting $m[j_1, v_1, j_2, v_2] = \mathbf{1}\{v_{1,j_1} = v_1 \text{ and } v_{2,j_2} = v_2\}$. Now consider the objective function $f : M \to \mathbb{R}$ defined by

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

where the expectation is over a uniformly random sample. We claim that $\max_{\boldsymbol{m}\in M} f(\boldsymbol{m}) = 1$ if and only if G has a clique of size k.

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

($\Rightarrow$) If $f(\boldsymbol{m}) = 1$, then for all j we must have $\sum_{v\in[n]} m[j, v, j, v] = 1$, i.e., $v_{1,j} = v_{2,j}$. But then $\{v_{1,j}, \dots, v_{1,k}\}$ must be a clique by construction, because otherwise there would be some $j_1 \ne j_2$ for which $m[j_1, v_{1,j}, j_2, v_{1,j}] = 0$.

This completes the proof. □

F 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 allocateed 4 threads, 64 GB of RAM, and 6 hours of runtime. We used Gurobi 9.5 to solve LPs and MIPs.

Implementation details In the implementation of the two-sided column-generation algorithm (Algorithm 1), before solving a pricing problem via its MIP formulation, we try to solve the linear relaxation in which $\boldsymbol{\xi} \in \mathcal{V}$. If the solution to such LP is a semi-randomized correlation plan we can avoid the overhead of solving a MIP. Moreover, our implementation makes use of Gurobis'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., 2021a).

F.1 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. (2022).

1. ^{3}Kr is 3-player *Kuhn poker* (Kuhn, 1950) with r ranks.
2. ^{3}Lbrs is 3-player *Leduc poker* (Southey et al., 2005) with b bets per round, r ranks, and s suits.
3. ^{3}GL is 3-player *Goofspiel* (Ross, 1971) with 3 ranks and imperfect information.
4. ^{3}Dd is 3-player *Liar's Dice* (Lisý et al., 2015) with one d-sided die per player.
5. ^{2}Bhwr is 2-player *Battleship* (Farina et al., 2019b) on a grid of size $h \times w$, one unit-size ship per player, and r rounds.
6. ^{2}Snbr is a simplified version of the 2-player *Sherrif of Nottingham* (Farina et al., 2019b) 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. (2021a) and Farina et al. (2019b). We also introduce two new parametric families of games: **1)** ^{3}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[11]. ^{3}TP 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. **2)** ^{2}RSiT 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 relaxation of von Stengel & Forges (2008) 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 G.

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

F.2 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 relaxation by von Stengel & Forges (2008) (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. (2021a) (denoted by **[FCGS21]**), which we adapted from the team domain. Table 1 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}B323 and ^{2}S133), and it can solve in less than 1 hour instances that previously were not computationally feasible (*e.g.*, ^{2}B324). On the other hand, when k is large (*e.g.*, in ^{2}RS23), 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.

We also ran experiments on three-player games using the correlation DAG LP (see Table 3 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.

F.3 Adversarial Team Games

We compared our new column generation approach (Algorithm 1) to prior approaches for finding team-correlated equilibria (TMECor) in zero-sum adversarial team games. Specifically, we compared to the CFR algorithm on the team DAG introduced by Zhang et al. (2022) and the prior column generation-based approach of Farina et al. (2021a). Results can be found in Table 2. Our results clearly give several conclusions. First, our algorithm is an improvement upon Farina et al. (2021a), achieving speedups of more than an order of magnitude in some games. Second, our algorithm, like Zhang et al. (2021), scales well in the information complexity k compared to that of Zhang et al. (2022): 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 1999].

G Games in Experiments

G.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 T3 to refer to the trick-taking game, where the 3 stands for the number of ranks. In the *perfect information* variant TP3, 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 T3[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. Note that $L = 9!/(3!)^3 = 1680$ is the full game.

G.2 Ride-Sharing Game Instances

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

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 $G^{\text{rs}} = (V^{\text{rs}}, E^{\text{rs}})$. Each vertex $v \in V^{\text{rs}}$ 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 $v \in V^{\text{rs}}$ 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 computed as 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.

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 & Sandholm, 2020). 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.

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 3 we report the structure of the two maps. The value between curly brackets is the reward for a request on that node.

G.3 Payoff Space Plots

In Figure 4, we show plots of space the feasible payoffs in several tested games. All three-player games we tested on were constant sum, so we show a 2D projection of the space of payoffs.

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, e.g., Farina et al. (2020), and demonstrates the importance of defining the various notions as separate.

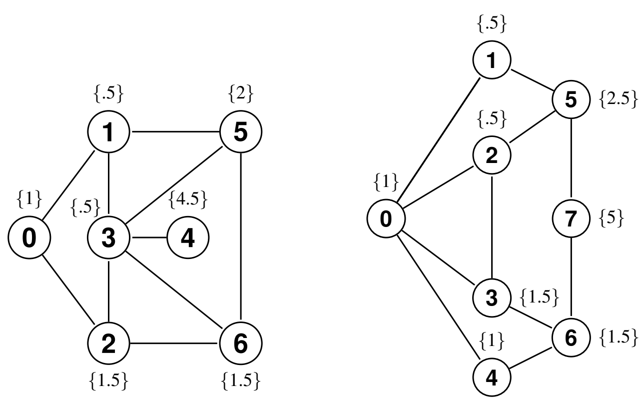


Figure 3: The two road network configurations which we consider. *Left*: map 1 (used for ^{2}RS13, ^{2}RS14). *Right*: map 2 (used for ^{2}RS23). In both cases the position of the two drivers is randomly chosen at the beginning of the game, edge costs are unitary, and one reward per node is indicated between curly brackets.

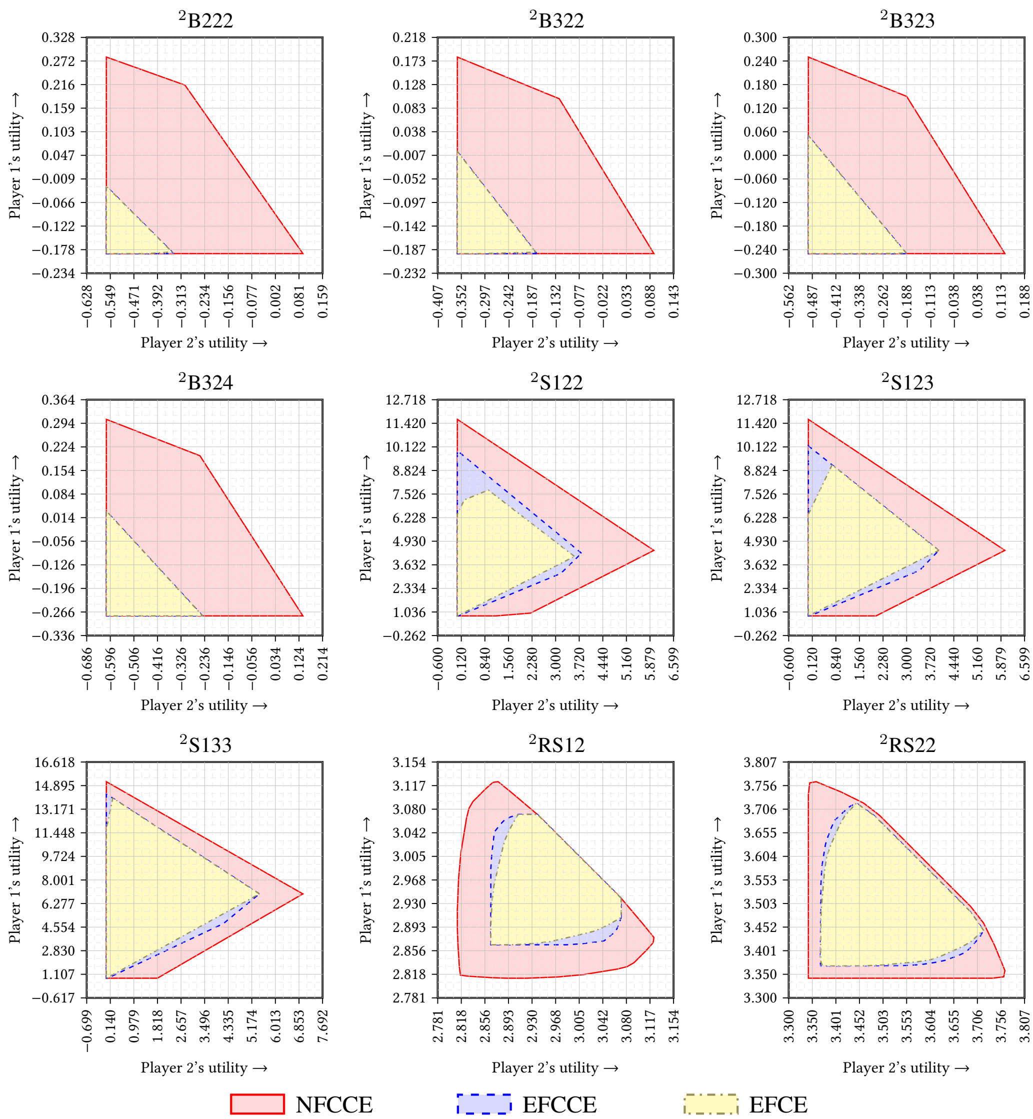
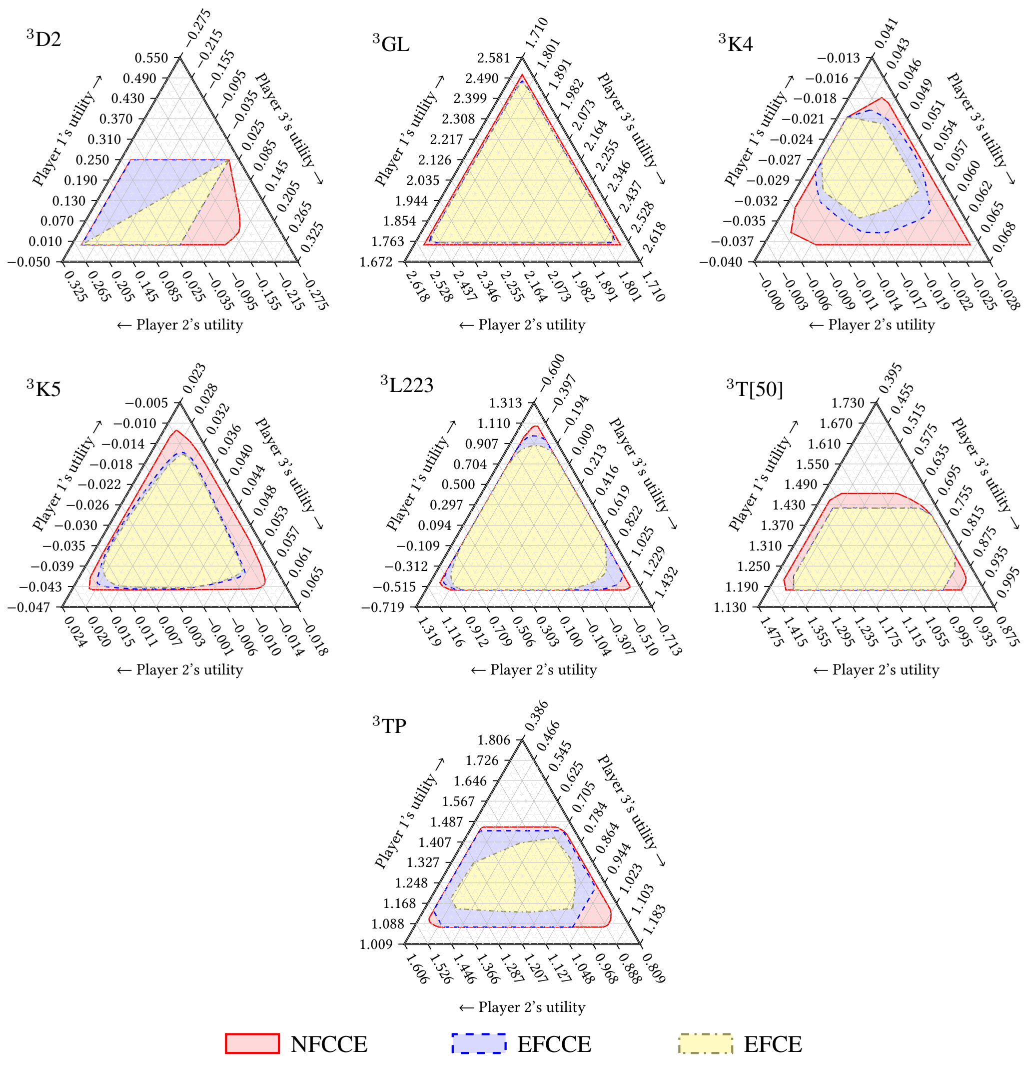
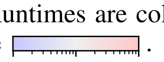


Figure 4: Payoff spaces in several games, with all three notions of equilibrium. Continued onto the next page.



^{3}D2
^{3}GL
^{3}K4
^{3}K5
^{3}L223
^{3}T[50]
^{3}TP
Player 1's utility →
Player 3's utility →
← Player 2's utility
NFCCE
EFCCE
EFCE

Game			Concept	$\lvert\mathcal{E}^c\rvert$	Value	[vSF08]	Column generation		DAG
							[FCGS21]	This paper	This paper
	$\lvert\mathcal{Z}\rvert$	1,072	NFCCE	7,093	0.000	0.07s	0.97s	0.24s	0.02s
^{2}B222	$\lvert\Sigma\rvert$	11,049	EFCCE	22,821	−0.525	0.15s	1m 56s	19.57s	0.05s
	k	8	EFCE	20,226	−0.525	0.28s	36m 46s	2m 1s	0.17s
	$\lvert\mathcal{Z}\rvert$	19,116	NFCCE	93,121	0.000	2.50s	13m 21s	13.65s	0.21s
^{2}B322	$\lvert\Sigma\rvert$	264,541	EFCCE	368,773	−0.317	4.30s	> 6h	1h 5m	1.38s
	k	12	EFCE	499,667	−0.317	15.23s	> 6h	> 6h	5.83s
	$\lvert\mathcal{Z}\rvert$	191,916	NFCCE	1,060,277	0.000	1m 5s	> 6h	> 6h	2.82s
^{2}B323	$\lvert\Sigma\rvert$	3,893,341	EFCCE	5,820,762	−0.375	2m 14s	> 6h	> 6h	32.94s
	k	12	EFCE	7,236,102	−0.375	oom	> 6h	oom	1m 55s
	$\lvert\mathcal{Z}\rvert$	969,516	NFCCE	6,531,150	0.000	oom	oom	oom	26.27s
^{2}B324	$\lvert\Sigma\rvert$	26,443,741	EFCCE	43,551,248	−0.489	oom	oom	oom	13m 15s
	k	12	EFCE	49,063,556	−0.489	oom	oom	oom	57m 12s
	$\lvert\mathcal{Z}\rvert$	396	NFCCE	3,059	13.636	0.07s	0.70s	0.28s	0.01s
^{2}S122	$\lvert\Sigma\rvert$	3,717	EFCCE	8,177	9.565	0.04s	12.00s	0.96s	0.02s
	k	12	EFCE	6,227	9.078	0.08s	51.49s	4.09s	0.04s
	$\lvert\mathcal{Z}\rvert$	2,376	NFCCE	19,187	13.636	0.26s	18.18s	3.03s	0.04s
^{2}S123	$\lvert\Sigma\rvert$	33,633	EFCCE	75,479	10.000	0.59s	1h 1m	5m 52s	0.23s
	k	12	EFCE	52,559	10.000	1.22s	1h 11m	7m 6s	0.65s
	$\lvert\mathcal{Z}\rvert$	5,632	NFCCE	46,755	18.182	1.05s	3m 21s	9.01s	0.04s
^{2}S133	$\lvert\Sigma\rvert$	95,768	EFCCE	179,571	15.000	1.94s	> 6h	1h 26m	1.51s
	k	12	EFCE	165,859	15.000	6.45s	> 6h	> 6h	2.46s
	$\lvert\mathcal{Z}\rvert$	400	NFCCE	10,366	6.010	n/a	0.04s	0.04s	0.02s
^{2}RS12	$\lvert\Sigma\rvert$	613	EFCCE	10,366	6.010	n/a	0.08s	0.06s	0.01s
	k	15	EFCE	8,846	6.010	n/a	0.54s	0.13s	0.01s
	$\lvert\mathcal{Z}\rvert$	4,356	NFCCE	oom	9.398	n/a	3.32s	2.82s	oom
^{2}RS13	$\lvert\Sigma\rvert$	15,063	EFCCE	oom	9.385	n/a	2m 10s	1m 28s	oom
	k	40	EFCE	oom	9.367	n/a	> 6h	12m 31s	oom
	$\lvert\mathcal{Z}\rvert$	484	NFCCE	34,947	7.188	n/a	0.08s	0.28s	0.20s
^{2}RS22	$\lvert\Sigma\rvert$	701	EFCCE	34,947	7.176	n/a	0.13s	0.08s	0.20s
	k	15	EFCE	31,503	7.176	n/a	0.56s	0.41s	0.16s
	$\lvert\mathcal{Z}\rvert$	4,096	NFCCE	oom	10.961	n/a	2.63s	3.12s	oom
^{2}RS23	$\lvert\Sigma\rvert$	13,277	EFCCE	oom	10.820	n/a	1m 51s	56.31s	oom
	k	44	EFCE	oom	10.791	n/a	> 6h	6m 35s	oom

Table 1: 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 & Forges (2008), which is only correct in triangle-free games (Farina & Sandholm, 2020). RS is not triangle-free, so vSF fails in that game. **FCGS21** is the one-sided column generation approach of Farina et al. (2021a). k is the information complexity. All runs were performed to convergence. 'oom': out of memory. Runtimes are colored according to the ratio with the best runtime in that row, according to the scale .

Game $\{\ominus\}$	**Leaves**	$\oplus$ **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	1.17s	0.36s	0.06s
^{3}K12 $\{3\}$	17,160	-0.014	24	oom	1.45s	0.47s
^{3}L132 $\{3\}$	4,500	0.293	6	0.01s	51.65s	4.62s
^{3}L133 $\{3\}$	6,477	0.215	6	0.01s	1m 22s	8.49s
^{3}L151 $\{3\}$	10,020	-0.019	10	0.06s	29.30s	1.14s
^{3}L223 $\{3\}$	8,762	0.516	4	0.01s	2m 0s	6.78s
^{3}L523 $\{3\}$	775,148	0.953	4	3.51s	$>$ 6h	4h 17m
^{3}D2 $\{3\}$	504	0.250	4	0.00s	0.08s	0.05s
^{3}D3 $\{3\}$	13,797	0.284	6	0.07s	6m 7s	5.76s
^{3}GL $\{3\}$	1,296	1.252	2	0.00s	0.71s	0.22s
^{3}T[50] $\{2\}$	10,300	0.600	5	0.00s	5.49s	1.02s
^{3}T[100] $\{2\}$	20,992	0.710	18	0.14s	13.34s	1.64s
^{3}T[840] $\{2\}$	190,228	0.569	141	oom	26m 43s	1m 51s
^{3}T $\{2\}$	379,008	≈ 0.573	141	oom	$>$ 6h	53m 23s
^{3}TP $\{2\}$	379,008	0.658	2	0.39s	7m 22s	1m 6s

Table 2: 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. (2022). **FCGS21** is the one-sided column generation approach of Farina et al. (2021a). 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 ± 0.003. 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 .
$\leq 1/10$ 1 ≥ 10

Game	**Leaves**	k	**Concept**	$\lvert\mathcal{E}^c\rvert$	**Runtime**	**Optimal value** P1	P2	P3
^{3}K4	312	12	NFCCE	233,023	0.73s	-0.018	-0.007	0.064
			EFCCE	253,007	0.93s	-0.020	-0.012	0.057
			EFCE	263,024	0.87s	-0.021	-0.013	0.055
^{3}K5	780	15	NFCCE	3,090,854	21.71s	-0.011	0.017	0.057
			EFCCE	3,378,914	46.81s	-0.016	0.015	0.052
			EFCE	4,095,962	1m 20s	-0.016	0.013	0.052
^{3}L223	8,762	6	NFCCE	625,432	2.49s	1.079	0.992	1.146
			EFCCE	6,213,623	39.09s	0.984	0.959	1.033
			EFCE	5,592,535	1m 0s	0.887	0.883	0.860
^{3}D2	504	6	NFCCE	33,926	0.06s	0.250	0.250	0.131
			EFCCE	52,651	0.09s	0.250	0.250	0.000
			EFCE	635,689	1.48s	0.250	0.250	0.000
^{3}GL	1,296	10	NFCCE	31,125	0.04s	2.505	2.505	2.505
			EFCCE	38,397	0.06s	2.476	2.476	2.476
			EFCE	17,382	0.05s	2.467	2.467	2.467
^{3}T[50]	10,300	15	NFCCE	139,091,992	1m 26s	1.463	1.380	0.887
			EFCCE	139,314,061	1m 38s	1.420	1.360	0.840
			EFCE	139,721,910	1m 53s	1.420	1.360	0.840
^{3}TP	379,008	3	NFCCE	6,838,696	11.99s	1.466	1.477	1.037
			EFCCE	12,786,328	22.07s	1.451	1.442	0.922
			EFCE	6,714,256	16.52s	1.423	1.360	0.886

Table 3: 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.