Polynomial-Time Computation of Optimal Correlated Equilibria in Two-Player Extensive-Form Games with Public Chance Moves and Beyond

Gabriele Farina,[1] **Tuomas Sandholm**[1,2,3,4]

[1]Computer Science Department, Carnegie Mellon University,
[2]Strategic Machine, Inc., [3]Strategy Robot, Inc., [4]Optimized Markets, Inc.
gfarina@cs.cmu.edu, sandholm@cs.cmu.edu

Abstract

Unlike normal-form games, where correlated equilibria have been studied for more than 45 years, *extensive-form* correlation is still generally not well understood. Part of the reason for this gap is that the sequential nature of extensive-form games allows for a richness of behaviors and incentives that are not possible in normal-form settings. This richness translates to a significantly different complexity landscape surrounding extensive-form correlated equilibria. As of today, it is known that finding an optimal *extensive-form correlated equilibrium (EFCE)*, *extensive-form coarse correlated equilibrium (EFCCE)*, or *normal-form coarse correlated equilibrium (NFCCE)* in a two-player extensive-form game is computationally tractable when the game does not include chance moves, and intractable when the game involves chance moves. In this paper we significantly refine this complexity threshold by showing that, in two-player games, an optimal correlated equilibrium can be computed in polynomial time, provided that a certain condition is satisfied. We show that the condition holds, for example, when all chance moves are *public*, that is, both players observe all chance moves. This implies that an optimal EFCE, EFCCE and NFCCE can be computed in polynomial time in the game size in two-player games with public chance moves, providing the biggest *positive* complexity result surrounding extensive-form correlation in more than a decade.

1 Introduction

A vast body of literature in computational game theory has focused on computing Nash equilibria (NEs) in two-player zero-sum imperfect-information extensive-form games. Success stories from that endeavor include the creation of strong—in some cases superhuman—AIs for several complex games, including two-player limit Texas hold'em (Bowling et al. 2015), two-player no-limit Texas hold'em (Brown and Sandholm 2017a,b; Moravčík et al. 2017), and multiplayer no-limit Texas hold'em (Brown and Sandholm 2019). NE captures strategic interactions in which each player maximizes her own utility. The interaction in NE is assumed to be fully decentralized: no communication between players is possible and the behavior of the players is not coordinated by any external orchestrator in any way. While that assumption

is natural in games such as poker, NE is too restrictive in other types of strategic interactions in which partial forms of communication or centralized control are possible (Ashlagi, Monderer, and Tennenholtz 2008). Therefore, there has been growing interest around less restrictive solution concepts than NE.

Correlated and *coarse correlated equilibria* are classic families of solution concepts that relax the assumptions of NE to allow forms of coordination of utility-maximizing agents (Aumann 1974; Moulin and Vial 1978). In correlated and coarse correlated equilibria, a mediator that can recommend behavior—but not enforce it—complements the game. Before the interaction starts, the mediator samples a profile of recommended strategies (one for each player) from a publicly known correlated distribution. The mediator reveals the next recommended move (or sequence of moves, depending on the specific solution concept in the family) to each acting player. In correlated equilibrium, each agent must decide whether to commit to following the next recommended move (or sequence of moves) *after* such move or sequence of moves is revealed by the mediator. In coarse correlated equilibrium, each agent must decide whether to commit to following the next recommended move (or sequence of moves) *before* it is revealed by the mediator. If a player chooses not to follow the recommendation, the mediator stops issuing further recommendations to that player. Since the selfish agents are free to not follow the recommendations, it is up to the mediator to come up with a correlated distribution of recommendations such that no agent has incentive to deviate from the recommendations, assuming no other player deviates. Despite the apparent weakness of a mediator that cannot enforce behavior but only suggest it, the maximum social welfare (that is, sum of the players' utilities) that can be induced by these families of solution concepts is greater than the social welfare obtainable by NE. Examples of interactions where a mediator is natural include traffic control and load balancing (Ashlagi, Monderer, and Tennenholtz 2008).

These equilibrium concepts have typically been studied in normal-form (that is, matrix) games. The study of correlation in *extensive-form* (that is, tree-form) games is recent, and was pioneered by von Stengel and Forges (2008). Three correlated solution concepts are often used in extensive-form games: *extensive-form correlated equilibrium (EFCE)* (von Stengel and Forges 2008), *extensive-form coarse corre-*
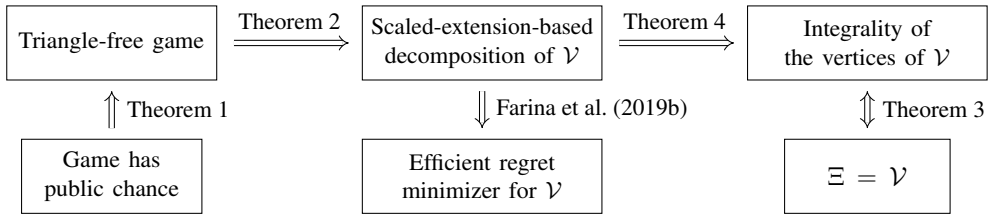
Figure 1: Overview of the connections among this paper's results.

lated equilibrium (EFCCE) (Farina, Bianchi, and Sandholm 2020), and *normal-form coarse correlated equilibrium (NFCCE)* (Moulin and Vial 1978; Celli, Coniglio, and Gatti 2019; Celli et al. 2019). Compared to normal-form (that is, one-shot) games, extensive-form correlation poses new and different challenges, especially in settings where the agents retain private information. This is unique to the sequential nature of extensive-form games, where, fundamentally, players can adjust strategically as they make observations about their opponents and the environment (Farina et al. 2019a). These challenges also translate to some negative complexity results for extensive-form correlation (Gilboa and Zemel 1989; von Stengel and Forges 2008). While a landmark positive complexity result in game theory shows that *one* EFCE, EFCCE, or NFCCE can be found in polynomial time (Papadimitriou and Roughgarden 2005; Huang and von Stengel 2008; Jiang and Leyton-Brown 2011), the computation of an *optimal* (that is, one that maximizes or minimizes a given linear objective, such as social welfare) EFCE, EFCCE, or NFCCE is computationally intractable in games with more than two players, as well as two-player games with chance moves, and tractable in two-player games without chance moves (von Stengel and Forges 2008).

In this paper we significantly refine this complexity threshold by showing that, in two-player games, an optimal correlated equilibrium can be computed in polynomial time, provided that a certain *triangle-freeness* condition—which can be checked in polynomial time—is satisfied. We prove that the condition holds, for example, when all chance moves are *public*, that is, both players observe all chance moves. This includes, for example, games where the chance outcomes amount to public dice rolls or public revelations of cards. Specifically, we show that the set of *correlation plans* Ξ of a triangle-free game coincides with the *von Stengel-Forges polytope* $\mathcal{V}$ of the game—a polytope that only requires a polynomial number of linear "probability-mass-conserving" constraints. Since $\mathcal{V}$ can be represented using a polynomial number of constraints in the input game size, optimizing over this set can be efficiently done by means of, for example, linear programming methods.

In Figure 1 we give an overview of the results in this paper and how they relate to each other. Our main result is that the polytope of correlation plans Ξ coincides with the von Stengel-Forges polytope $\mathcal{V}$ when the game satisfies the *triangle-freeness* condition that we introduce (Definition 3). As we show in Theorem 1, every two-player game with public chance moves (which includes games with no chance moves at all) is triangle-free, but not all triangle-free games

have public chance moves. So, our results also apply to some games where chance is not public. The equality Ξ = $\mathcal{V}$ in triangle-free games implies that an optimal EFCE, EFCCE and NFCCE can be computed in polynomial time. This is because $\mathcal{V}$ has a polynomial (in the game size) description (von Stengel and Forges 2008) and the computation of an EFCE, EFCCE, NFCCE can be expressed as a linear program (von Stengel and Forges 2008; Farina, Bianchi, and Sandholm 2020).

We prove Ξ = $\mathcal{V}$ in several steps. First, we show that in triangle-free games, $\mathcal{V}$ admits a structural decomposition in terms of *scaled extension* operations. This type of decomposition of $\mathcal{V}$ was introduced by Farina et al. (2019b) as a way of "unrolling" the combinatorial structure of $\mathcal{V}$ to construct an efficient regret minimization algorithm for Ξ in two-player games without chance moves. We extend their construction to handle any triangle-free game. Then, we show a deep connection between the integrality of the vertices of the von Stengel-Forges polytope $\mathcal{V}$ and Ξ. Namely, in Theorem 3, we show that Ξ = $\mathcal{V}$ holds if and only if all of $\mathcal{V}$'s vertices have integer $\{0, 1\}$ coordinates. Finally, in Section 4 we prove that $\mathcal{V}$ has integral vertices by leveraging its structural decomposition.

2 Preliminaries

Extensive-form games *Extensive-form games (EFGs)* are the standard model for games that are played on a game tree. EFGs can capture sequential and simultaneous moves as well as private information. Each node in the EFG belongs to one player. One special player, called the *chance player*, is used to model random stochastic events, such as rolling a die or drawing cards. In this paper, we only consider games that have two players in addition to potentially having a chance player.

Edges leaving from a node represent actions that a player can take at that node. To model private information, the game tree is supplemented with an *information partition*, defined as a partition of nodes into sets called *information sets*. Each node belongs to exactly one information set, and each information set is a nonempty set of tree nodes for the same Player i. An information set for Player i denotes a collection of nodes that Player i cannot distinguish among, given what she has observed so far. The symbols $\mathcal{I}_1$ and $\mathcal{I}_2$ denote the information partition of Player 1 and 2, respectively. Let I_1 and I_2 be information sets for Player 1 and 2, respectively. I_1 and I_2 are *connected*, denoted $I_1 \rightleftharpoons I_2$, if there exist nodes $u \in I_1$ and $v \in I_2$ such that u is on the path from the root to v, or vice versa.

We will only consider *perfect-recall* games, that is, no player forgets what the player knew earlier. As a consequence, all nodes that belong to an information set I share the same set of available actions (otherwise the player acting at those nodes would be able to distinguish among them), which we denote by A_I. We define the set of *sequences* of Player i as the set $\Sigma_i := \{(I, a) : I \in \mathcal{I}_i, a \in A_I\} \cup \{\varnothing\}$, where the special element $\varnothing$ is called *empty sequence*. Given an information set $I \in \mathcal{I}_i$, we denote by $\sigma(I)$ the *parent sequence* of I, defined as the last pair $(I', a') \in \Sigma_i$ encountered on the path from the root to any node $v \in I$; if no such pair exists we let $\sigma(I) = \varnothing$.

An important concept in extensive-form correlation is *relevance* of sequence pairs. Intuitively, two sequences are relevant if they belong to connected information sets or if either of them is the empty sequence. Formally, a pair of sequences $(\sigma_1, \sigma_2) \in \Sigma_1 \times \Sigma_2$ is *relevant*, denoted $\sigma_1 \bowtie \sigma_2$, if either σ_1 or σ_2 or both is the empty sequence, or if $\sigma_1 = (I_1, a_1)$ and $\sigma_2 = (I_2, a_2)$ and $I_1 \rightleftharpoons I_2$. The set of all relevant sequence pairs is denoted $\Sigma_1 \bowtie \Sigma_2$. Given $\sigma_1 \in \Sigma_1$ and $I_2 \in \mathcal{I}_2$, we say that σ_1 is relevant for I_2, and write $\sigma_1 \bowtie I_2$, if $\sigma_1 = \varnothing$ or if $\sigma_1 = (I_1, a_1)$ and $I_1 \rightleftharpoons I_2$ (an analogous statement holds for $I_1 \bowtie \sigma_2$). We say that a sequence $\sigma = (I, a) \in \Sigma_i$ for Player i is *descendent* of another sequence $\sigma' = (I', a') \in \Sigma_i$ for the same player, denoted by $\sigma \succeq \sigma'$, if $\sigma = \sigma'$ or if there is a path from the root of the game to a node $v \in I$ that passes through action a' at some node $v' \in I'$. We use the notation $\tau \succ \tau'$ to mean $\tau \succeq \tau' \wedge \tau \neq \tau'$.

A *reduced-normal-form plan* π_i for Player i defines a choice of action for every information set $I \in \mathcal{I}_i$ that is still reachable as a result of the other choices in π itself. We denote the set of reduced-normal-form plans of Player i by Π_i. We denote by $\Pi_i(I)$ the subset of reduced-normal-form plans that prescribe all actions for Player i on the path from the root to information set I.

Polytope of correlation plans (Ξ) A correlated distribution μ over combinations of plans $\Pi_1 \times \Pi_2$ of the players can be thought of as a point in probability simplex $\Delta^{|\Pi_1 \times \Pi_2|}$. Since the number of plans of each player is exponential in the game tree size, so is that representation of μ. Therefore, von Stengel and Forges (2008) introduced a more compact representation of μ, called the *correlation plan representation*. The set of all legal correlation plans is denoted Ξ and called the *polytope of correlation plans*. The set Ξ is a convex polytope in $\mathbb{R}_{\geq 0}^{|\Sigma_1 \bowtie \Sigma_2|}$, so the number of variables is at most quadratic in the game tree size. However, it might still require an exponential number of *constraints*.

An optimal EFCE, EFCCE, or NFCCE is an optimal correlation plan subject to a set of linear *incentive constraints* (von Stengel and Forges 2008; Farina et al. 2019a; Farina, Bianchi, and Sandholm 2020). These constraints encode the requirement that the set of correlated behavior be *incentive compatible* for the player, that is, that no player would be better off not following the recommended behavior than to always follow it. Hence, an optimal EFCE, EFCCE, or NFCCE can be computed as the solution of a linear program. Furthermore, the linear program can be solved in polynomial time if and only if Ξ can be described with a polynomial number of lin-

ear constraints. Thus the characterization of the constraints that define Ξ in various classes of games is important.

The von Stengel-Forges polytope ($\mathcal{V}$) The characterization of the constraints that define Ξ was initiated by von Stengel and Forges (2008) in their landmark paper on extensive-form correlation. In particular, they show that in two-player perfect-recall games without chance moves, Ξ coincides with a particular polytope $\mathcal{V}$—which we call the *von Stengel-Forges polytope*—whose description only uses a polynomial number of linear constraints, which are "probability-mass-conserving" constraints:

$$\mathcal{V} := \left\{ \boldsymbol{v} \in \mathbb{R}_{\geq 0}^{|\Sigma_1 \bowtie \Sigma_2|} : \begin{array}{l} \bullet\ v[\varnothing, \varnothing] = 1 \\ \bullet\ \sum_{a \in A_{I_1}} v[(I_1, a), \sigma_2] = v[\sigma(I_1), \sigma_2] \\ \qquad \forall\, I_1 \in \mathcal{I}_1, \sigma_2 \in \Sigma_2 \text{ s.t. } I_1 \bowtie \sigma_2 \\ \bullet\ \sum_{a \in A_{J}} v[\sigma_1, (I_2, a)] = v[\sigma_1, \sigma(I_2)] \\ \qquad \forall\, I_2 \in \mathcal{I}_2, \sigma_1 \in \Sigma_1 \text{ s.t. } \sigma_1 \bowtie I_2 \end{array} \right\}. \tag{1}$$

The polytope $\mathcal{V}$ is well defined in every game. However, the equality $\Xi = \mathcal{V}$ was known to hold only in two-player games without chance moves. In more general games, it is only known that $\Xi \subseteq \mathcal{V}$. The main contribution of our paper is to show that the equality $\Xi = \mathcal{V}$ holds in significantly more general games than two-player games without chance moves. We will isolate a condition, which we coin *triangle freeness*, that is sufficient for $\Xi = \mathcal{V}$ to hold. We also show that all two-player games where all chance moves are public (including two-player games without chance moves) are triangle free.

3 Scaled-Extension-Based Structural Decomposition for $\mathcal{V}$

Farina et al. (2019b) recently showed that in two-player games without chance moves, a particular structural decomposition theorem holds for the von Stengel-Forges polytope $\mathcal{V}$. At the core of their decomposition is a convexity-preserving operation, *scaled extension*, defined as follows.

Definition 1 ((Farina et al. 2019b))**.** *Let $\mathcal{X}$ and $\mathcal{Y}$ be nonempty, compact and convex sets, and let $h : \mathcal{X} \to \mathbb{R}_{\geq 0}$ be a nonnegative affine real function. The* scaled extension *of $\mathcal{X}$ with $\mathcal{Y}$ via h is defined as the set*

$$\mathcal{X} \stackrel{h}{\triangleleft} \mathcal{Y} := \{(\boldsymbol{x}, \boldsymbol{y}) : \boldsymbol{x} \in \mathcal{X},\ \boldsymbol{y} \in h(\boldsymbol{x})\mathcal{Y}\}.$$

Specifically, they show that in two-player games without chance moves, $\mathcal{V}$ admits a decomposition of the form $\mathcal{V} = \{1\} \triangleleft^{h_1} \mathcal{X}_1 \triangleleft^{h_2} \mathcal{X}_2 \triangleleft^{h_3} \cdots \triangleleft^{h_n} \mathcal{X}_n$, where each of the sets $\mathcal{X}_i$ is either the singleton set $\{1\}$, or a probability simplex $\Delta^{s_i} := \{\boldsymbol{x} \in \mathbb{R}_{\geq 0}^{s_i} : \|\boldsymbol{x}\|_1 = 1\}$ for some appropriate dimension s_i.

In this section, we significantly extend their result. As we will show, an analogous scaled-extension-based decomposition of $\mathcal{V}$ exists in far more general games than those without chance moves. In particular, in Section 3 we isolate a condition on the information structure of the game—which we coin *triangle freeness*—that guarantees existence of a scaled-extension-based decomposition. Then, we will present an algorithm for computing such a decomposition, that is, finding the h_i functions and sets $\mathcal{X}_i$. Since our full algorithm is
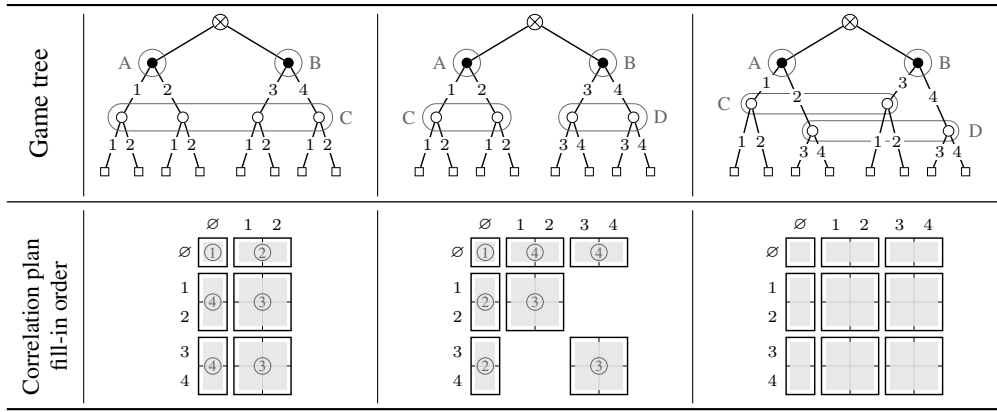
Figure 2: Three examples of extensive-form games with increasingly complex information partitions. Crossed nodes belong the chance player, black round nodes belong to Player 1, white round nodes belong to Player 2, gray round sets define information sets, white squares denote terminal states. The numbers along the edges define concise names for sequences.

rather intricate, we start by giving three examples of increasing complexity that capture the main intuition behind our structural decomposition routine.

First example The first example is shown in the first column of Figure 2. The game starts with a chance node, where two outcomes (say, heads or tails) are possible. After observing the outcome of the chance node, Player 1 chooses between two actions (say, the "left" and the "right" action). The choice as to whether to play the left or the right action can be different based on the observed chance outcome. After Player 1 has played their action, Player 2 has to pick whether to play their left or right action—however, Player 2 does not observe the chance outcome nor Player 1's action. The chance outcome is not observed by Player 2, so, this is not a public-chance game.

The only information set C for Player 2 is connected to both information sets (denoted A and B in Figure 2) of Player 1, so, all sequence pairs $(\sigma_1, \sigma_2) \in \Sigma_1 \times \Sigma_2$ are relevant. Since Player 2 only has one information set, it is easy to incrementally generate the von Stengel-Forges polytope. First, the fixed value 1 is assigned to $v[\varnothing, \varnothing]$ (step ① in the fill-in order). Then, this value is split arbitrarily into the two (non-negative) entries $v[\varnothing, 1], v[\varnothing, 2]$ so that $v[\varnothing, 1] + v[\varnothing, 2] = v[\varnothing, \varnothing]$ in accordance with the von Stengel-Forges constraints. This operation can be expressed using scaled extension as $\{(v[\varnothing, \varnothing], v[\varnothing, 1], v[\varnothing, 2])\} = \{1\} \triangleleft^h \Delta^2$, where h is the identity function (step ② in the fill-in order). Then, $v[\varnothing, 1]$ is further split into $v[1, 1] + v[2, 1] = v[\varnothing, 1]$ and $v[3, 1] + v[4, 1] = v[\varnothing, 1]$, while $v[\varnothing, 2]$ is split into $v[1, 2] + v[2, 2] = v[\varnothing, 2]$ and $v[3, 2] + v[4, 2] = v[\varnothing, 2]$ (step ③ of the fill-in order). These operations can be expressed as scaled extensions with Δ^2. Now that the eight entries $v[\sigma_1, \sigma_2]$ for $\sigma_1 \in \{1, 2, 3, 4\}, \sigma_2 \in \{1, 2\}$ have been filled out, we fill in $v[\sigma_1, \varnothing]$ for all $\sigma_1 \in \{1, 2, 3, 4\}$ in accordance with the von Stengel-Forges constraint $v[\sigma_1, \varnothing] = v[\sigma_1, 1] + v[\sigma_1, 2]$ (step ④). In this step, we are not splitting any values, but rather we are summing already-filled-in entries in v to form new entries. Specifically, we can extend the set of partially-filled-in

vectors
$$\boldsymbol{v} = (v[\varnothing, \varnothing], v[\varnothing, 1], v[\varnothing, 2], v[1, 1], v[2, 1],$$
$$2cmv[3, 1], v[4, 1], v[1, 2], v[2, 2], v[3, 2], v[4, 2])$$

with the new entry $v[1, \varnothing]$ by using the scaled extension operation $\{\boldsymbol{v}\} \triangleleft^h \{1\}$ where h is the (linear) function that extracts the sum $v[\sigma_1, 1] + v[\sigma_1, 2]$ from $\boldsymbol{v}$. By doing so, we have incrementally filled in all entries in $\boldsymbol{v}$. Furthermore, by construction, we have that all von Stengel-Forges constraints $v[\sigma_1, \varnothing] = v[\sigma_1, 1] + v[\sigma_1, 2]$ ($\sigma_1 \in \{\varnothing, 1, 2, 3, 4\}$) and $v[\varnothing, \sigma_2] = v[1, \sigma_2] + v[2, \sigma_2] = v[3, \sigma_2] + v[4, \sigma_2]$ ($\sigma_2 \in \{1, 2\}$) must hold. So, the only two von Stengel-Forges constraints that we have ignored and might potentially be violated are $v[\varnothing, \varnothing] = v[1, \varnothing] + v[2, \varnothing]$ and $v[\varnothing, \varnothing] = v[3, \varnothing] + v[4, \varnothing]$. This concern is quickly resolved by noting that those constraints are implied by the other ones that we satisfy. In particular, by construction we have $v[1, \varnothing] + v[2, \varnothing] = (v[1, 1] + v[1, 2]) + (v[2, 1] + v[2, 2]) = (v[1, 1] + v[2, 1]) + (v[1, 2] + v[2, 2]) = v[\varnothing, 1] + v[\varnothing, 2] = v[\varnothing, \varnothing]$, and an analogous statement holds for $v[3, \varnothing] + v[4, \varnothing]$. So, all constraints hold and the scaled-extension-based decomposition is finished.

Remark 1. *An approach that would start by splitting $v[\varnothing, \varnothing]$ into $v[1, \varnothing] + v[2, \varnothing] = v[\varnothing, \varnothing]$ and $v[3, \varnothing] + v[4, \varnothing] = v[\varnothing, \varnothing]$, thereby inverting the order of fill-in steps ④ and ②, would fail. Indeed, after filling $v[\sigma_1, \sigma_2]$ for all $\sigma_1 \in \{1, 2, 3, 4\}, \sigma_2 \in \{1, 2\}$), there would be no clear way of guaranteeing that $v[1, 1] + v[2, 1] = v[3, 1] + v[4, 1]$ (= $v[\varnothing, 1]$).*

Second example We now consider a variation of the game from the first example, where Player 2 observes the chance outcome but not the actions selected by Player 1. This game, shown in the middle column of Figure 2, has *public* chance moves, because the chance outcome is observed by all players. In this game, not all pairs of information sets are connected. In fact, only (A, C) and (B, D) are connected information set pairs. Correspondingly, entries such as $v[1, 3]$, $v[4, 2]$, and $v[2, 4]$ are not defined in the correlation plans for

the game. This observation is crucial, and will set apart this example from the next one. To fill in any correlation plan, we can start by splitting $v[\varnothing, \varnothing]$ into $v[1, \varnothing] + v[2, \varnothing] = v[\varnothing, \varnothing]$ and $v[3, \varnothing] + v[4, \varnothing] = v[\varnothing, \varnothing]$ (fill-in step ② in the figure). Both operations can be expressed as a scaled extension of partially-filled-in vectors with Δ^2, scaled by the affine function that extracts $v[\varnothing, \varnothing] = 1$ from the partially-filled-in correlation plans. Then, we further split those values into entries $v[\sigma_1, 1] + v[\sigma_1, 2] = v[\sigma_1, \varnothing]$ for $\sigma_1 \in \{1, 2\}$ in accordance with the von Stengel-Forges constraint. Similarly, we will in $v[\sigma_1, 3], v[\sigma_1, 4]$ for $\sigma_1 \in \{3, 4\}$ in accordance with the constraint $v[\sigma_1, 3] + v[\sigma_1, 4] = v[\sigma_1, \varnothing]$ for $\sigma_1 \in \{1, 2\}$ (fill-in step ③). Finally, we recover the values of $v[\varnothing, \sigma_2]$ for $\sigma_2 \in \{1, 2, 3, 4\}$ with a scaled extension with the singleton set $\{1\}$ as discussed in the previous example. Again, it can be checked that despite the fact that we ignored the constraints $v[\varnothing, 1] + v[\varnothing, 2] = v[\varnothing, \varnothing]$ and $v[\varnothing, 3] + v[\varnothing, 4] = v[\varnothing, \varnothing]$, those constraints are automatically satisfied by construction. In this case, we were able to sidestep the issue raised in Remark 1 because of the particular connection between the information sets.

Third example Finally, we propose a third example in the third column of Figure 2. It is a variation of the first example, where Player 2 now observes Player 1's action but *not* the chance outcome. The most significant difference with the second example is that the information structure of the game is now such that all pairs of information sets of the players are connected. Hence, the problem raised in Remark 1 cannot be avoided. Our decomposition algorithm cannot handle this example.

A Sufficient Condition for the Existence of a Scaled-Extension-Based Decomposition

The third example in the previous section highlights an unfavorable situation in which our decomposition attempt based on incremental generation of the correlation plan. In order to codify all situations in which that issue does not arise, we introduce the concept of rank of an information set.

Definition 2. *Let $i \in \{1, 2\}$ be one player, and let $-i$ denote the other player. Furthermore, let $I \in \mathcal{I}_i$ and $\sigma \in \Sigma_{-i}$. The σ-rank of I is the cardinality of the set $\{J \in \mathcal{I}_{-i} : J \rightleftharpoons I, \sigma(J) = \sigma\}$.*

The issue in Remark 1 can be stated in terms of the ranks. Consider a relevant sequence pair $(\sigma_1, \sigma_2) \in \Sigma_1 \bowtie \Sigma_2$ and two connected information sets $I_1 \rightleftharpoons I_2$ such that $\sigma(I_1) = \sigma_1, \sigma(I_2) = \sigma_2$. If the σ_1-rank of I_2 and the σ_2-rank of I_1 are both greater than 1, the issue cannot be avoided and the decomposition will fail. For example, in the third example, where our decomposition fails, all information sets have $\varnothing$-rank 2. We prove that such situations cannot occur, provided the game satisfies the following condition, which can be verified in polynomial time in the size of the EFG.

Definition 3 (Triangle-freeness)**.** *A two-player extensive-form game is* triangle-free *if, for any choice of two distinct information sets $I_1, I_2 \in \mathcal{I}_1$ such that $\sigma(I_1) = \sigma(I_2) = \sigma_1$ and two distinct information sets $J_1, J_2 \in \mathcal{I}_2$ such that $\sigma(J_1) = \sigma(J_2) = \sigma_2$, it is never the case that $I_1 \rightleftharpoons J_1 \land I_2 \rightleftharpoons J_2 \land I_1 \rightleftharpoons J_2$.*

In Theorem 1 we show that games with public chance (which includes games with no chance moves at all) always satisfy the triangle-freeness condition of Definition 3.

Theorem 1. *A two-player extensive-form game with public chance moves is triangle-free.*

However, not all triangle-free games must have public chance nodes. For example, the topmost game in Figure 2 is triangle-free, but in that game the chance outcome is not public to Player 2. So, our results apply more broadly than games with public chance moves.

Computation of the Decomposition

We present our algorithm following the same structure as (Farina et al. 2019b). Like theirs, our algorithm consists of a recursive function, DECOMPOSE. It takes three arguments: (i) a sequence pair $(\sigma_1, \sigma_2) \in \Sigma_1 \bowtie \Sigma_2$, (ii) a subset $\mathcal{S}$ of the set of all relevant sequence pairs, and (iii) a set $\mathcal{D}$ where only the entries indexed by the elements in $\mathcal{S}$ have been filled in. The decomposition for the whole von Stengel-Forges polytope $\mathcal{V}$ is computed by calling DECOMPOSE$((\varnothing, \varnothing), \{(\varnothing, \varnothing)\}, \{(1)\})$—this corresponds to the starting situation in which only the entry $v[\varnothing, \varnothing]$ has been filled in (denoted as fill-in step ① in Figure 2). Each call to DECOMPOSE returns a pair $(\mathcal{S}', \mathcal{D}')$ of updated indices and partial vectors, to reflect the new entries that were filled in during the call.

DECOMPOSE$((\sigma_1, \sigma_2), \mathcal{S}, \mathcal{D})$ operates as follows (we denote with $-i$ the opponent for Player i):

1. Let $\mathcal{J}_i := \{I \in \mathcal{I}_i : I \bowtie \sigma_{-i}, \sigma(I) = \sigma_i\}$ for all $i \in \{1, 2\}$, and $\mathcal{J}^* \leftarrow \emptyset$.
2. For each $(I_1, I_2) \in \mathcal{J}_1 \times \mathcal{J}_2$ such that $I_1 \rightleftharpoons I_2$, if the σ_2-rank of I_1 is greater than or equal to the σ_1-rank of I_2, we *branch* on Player 1, update $\mathcal{J}^* \leftarrow \mathcal{J}^* \cup \{I_1\}$. Else, update $\mathcal{J}^* \leftarrow \mathcal{J}^* \cup \{I_2\}$.
3. For each $i \in \{1, 2\}$ and $I \in \mathcal{J}_i$ such that the σ_{-i}-rank of I is 0, do $\mathcal{J}^* \leftarrow \mathcal{J}^* \cup \{I\}$.
4. For each $I \in \mathcal{J}^*$: (Below we assume that $I \in \mathcal{I}_1$, the other case is symmetrical)

   (a) Fill in all entries $\{v[(I, a), \sigma_2] : a \in A_I\}$ by splitting $v[\sigma_1, \sigma_2]$. This can be expressed using a scaled extension operation as $\mathcal{D} \leftarrow \mathcal{D} \triangleleft^h \Delta^{|A_I|}$ where h extracts $\xi[\sigma_1, \sigma_2]$ from any partially-filled-in vector.

   (b) Update $\mathcal{S} \leftarrow \mathcal{S} \cup \{((I, a), \sigma_2)\}$ to reflect that the entries corresponding to $(I, a) \bowtie \sigma_2$ have been filled in.

   (c) For each $a \in A_I$ we assign $(\mathcal{S}, \mathcal{D}) \leftarrow$ DECOMPOSE$(((I, a), \sigma_2), \mathcal{S}, \mathcal{D})$.

   (d) Let $\mathcal{K} := \{J \in \mathcal{I}_2 : I \rightleftharpoons J\}$. For all $J \in \mathcal{I}_2$ such that $\sigma(J) \succeq (J', a')$ for some $J' \in \mathcal{K}, a' \in A_{J'}$:
   - If $I \rightleftharpoons J$, then for all $a \in A_J$ we fill in the sequence pair $\xi[\sigma_1, (J, a)]$ by assigning its value in accordance with the von Stengel-Forges constraint $\xi[\sigma_1, (J, a)] = \sum_{a^* \in A_{I^*}} \xi[(I^*, a^*), (J, a)]$ via the scaled extension $\mathcal{D} \leftarrow \mathcal{D} \triangleleft^h \{1\}$ where the linear function h maps a partially-filled-in vector to the value of $\sum_{a^* \in A_{I^*}} \xi[(I^*, a^*), (J, a)]$.
   - Otherwise, we fill in the entries $\{\xi[\sigma_1, (J, a)] : a \in A_J\}$, by splitting the value $\xi[\sigma_1, \sigma(J)]$. In this case,

we let $\mathcal{D} \leftarrow \mathcal{D} \vartriangleleft^h \Delta^{|A_J|}$ where h extracts the entry $\xi[\sigma_1, \sigma(J)]$ from a partially-filled-in vector in $\mathcal{D}$.

5. At this point, all the entries corresponding to indices $\tilde{\mathcal{S}} = \{(\sigma_1', \sigma_2') : \sigma_1' \succeq \sigma_1, \sigma_2' \succeq \sigma_2\}$ have been filled in, and we return $(\mathcal{S} \cup \tilde{\mathcal{S}}, \mathcal{D})$.

The above algorithm formalizes and generalizes the first two examples of Figure 2. For example, step ② of the fill-in order in either example is captured in Step 4(a), while fill-in step ③ corresponds to Step 4(c). Finally, fill-in step ④ corresponds to Step 4(d).

Compared to the decomposition algorithm by Farina et al. (2019b), our branching steps (Step 4) are significantly more intricate. This is because, compared to their setting (that is, two-player games without chance moves) where at least one player has at most one information set with rank strictly greater than one, we have to account for multiple information sets with rank greater than one. Since two-player games without chance moves are a special case of two-player games with public chance moves, our algorithm completely subsumes that of Farina et al. (2019b).

A proof of correctness for the algorithm is in Appendix A. In particular, the following holds.

Theorem 2. *The von Stengel-Forges polytope $\mathcal{V}$ of a two-player perfect-recall triangle-free EFG can be expressed via a sequence of scaled extensions with simplexes and singleton sets:*

$$\mathcal{V} = \{1\} \overset{h_1}{\vartriangleleft} \mathcal{X}_1 \overset{h_2}{\vartriangleleft} \mathcal{X}_2 \overset{h_3}{\vartriangleleft} \cdots \overset{h_n}{\vartriangleleft} \mathcal{X}_n, \qquad (2)$$

where, for $i = 1, \ldots, n$, either $\mathcal{X}_i = \Delta^{s_i}$ or $\mathcal{X}_i = \{1\}$ and h_i is a linear function. Furthermore, an exact algorithm exists to compute such expression in linear time in the dimensionality of $\mathcal{V}$, and so, in time at most quadratic in the size of the game.

4 Bridging the Gap Between $\mathcal{V}$ and Ξ

As noted by von Stengel and Forges (2008), the inclusion $\Xi \subseteq \mathcal{V}$ holds trivially in any game. The reverse inclusion, $\Xi \supseteq \mathcal{V}$, was shown for two-player games without chance moves, but no complete characterization as to when that reverse inclusion holds was known before our paper. In Theorem 3, we contribute a new connection between the reverse inclusion $\Xi \supseteq \mathcal{V}$ and the integrality of the vertices of the von Stengel-Forges polytope (all proofs are in Appendix B).

Theorem 3. *Let Γ be a two-player perfect-recall extensive-form game, let $\mathcal{V}$ be its von Stengel-Forges polytope, and let Ξ be its polytope of correlation plans. Then, $\Xi = \mathcal{V}$ if and only if all vertices of $\mathcal{V}$ have integer $\{0, 1\}$ coordinates.*

As it turns out, the scaled-based decomposition of $\mathcal{V}$ can be used to conclude the integrality of the vertices of $\mathcal{V}$, by leveraging the following analytical result about the scaled extension operation.

Lemma 1. *Let $\mathcal{X}, \mathcal{Y}$, and h be as in Definition 1. If $\mathcal{X}$ is a convex polytope with vertices $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, and $\mathcal{Y}$ is a convex polytope with vertices $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m\}$, then $\mathcal{X} \vartriangleleft^h \mathcal{Y}$ is a convex polytope whose vertices are a nonempty subset of $\{(\boldsymbol{x}_i, h(\boldsymbol{x}_i)\boldsymbol{y}_j) : i \in \{1, \ldots, n\}, j \in \{1, \ldots, m\}\}$.*

In particular, by applying Lemma 1 inductively on the structure of the scaled-extension-based structural decomposition of $\mathcal{V}$, we obtain the following theorem.

Theorem 4. *Let $\mathcal{V}$ be the von Stengel-Forges polytope of a two-player triangle-free game (Definition 3). All vertices of $\mathcal{V}$ have integer $\{0, 1\}$ coordinates.*

Finally, combining Theorem 4 and Theorem 3, we obtain the central theorem of this paper.

Theorem 5. *In a two-player perfect-recall extensive-form game that satisfies the triangle-freeness condition (Definition 3), the polytope of correlation plans coincides with the von Stengel-Forges polytope. Consequently, an optimal EFCE, EFCCE, or NFCCE can be computed in polynomial time (in the size of the input extensive-form game) in two-player triangle-free games.*

A consequence of $\mathcal{V} = \Xi$ is that the linear programs for EFCE (von Stengel and Forges 2008), EFCCE (Farina, Bianchi, and Sandholm 2020) and NFCCE (Farina, Bianchi, and Sandholm 2020)—originally formulated for two-player games without chance moves only—hold verbatim for any triangle-free game. So, an optimal EFCE, EFCCE, and NFCCE can be computed in polynomial time as the solution of those linear programs. Furthermore, the scaled-extension-based decomposition for triangle-free games (Section 3) can be combined with the *scaled extension regret circuit* introduced by Farina et al. (2019b); Farina, Kroer, and Sandholm (2019) to construct a scalable regret minimization algorithm for $\mathcal{V} = \Xi$. That, in turn, can be used to compute an EFCE, EFCCE, and NFCCE in large triangle-free games that are too large for traditional linear programming methods.

5 Experimental Evaluation

We implemented the scaled-extension-based decomposition routine of Section 3. We test our decomposition algorithm for triangle-free games on Goofspiel (Ross 1971), a popular benchmark game in computational game theory. In Goofspiel, each player has a personal deck of cards made of k different ranks (from 1 to k). A third deck (the "prize" deck) is shuffled and put face down on the board at the beginning of the game. In each turn, the topmost card from the prize deck is *publicly* revealed. Then, each player privately picks a card from their hand—this card acts as a bid to win the card that was just revealed from the prize deck. The player that bids the highest wins the prize card. We use an established tie-breaking rule: the prize card is discarded if the players' bids are equal. Furthermore, we adopt the convention that only the winner is revealed, but not the bids, in accordance with prior computational game theory literature (Lanctot et al. 2009; Lanctot 2013). The players' scores are computed as the sum of the values of the prize cards they have won. Because of the tie-breaking rule, Goofspiel is a general-sum game. Furthermore, since all chance outcomes are public, it is a triangle-free game.

In Figure 3(left) we report the performance of our decomposition routine for $k = 3, 4, 5$, both in terms of number of scaled extension operations required in the decomposition (Theorem 2) and of runtime of our single-threaded implementation, as well as the dimensions of the games. The runtime

Deck size	Information sets		Sequences		Decomposition		
	$\lvert\mathcal{I}_1 \cup \mathcal{I}_2\rvert$	$\lvert\mathcal{I}_1 \rightleftharpoons \mathcal{I}_2\rvert$	$\lvert\Sigma_1 \cup \Sigma_2\rvert$	$\lvert\Sigma_1 \bowtie \Sigma_2\rvert$	Num $\triangleleft^h$	Runtime	
3 ranks	$4.3{\times}10^2$	$1.1{\times}10^3$	$5.2{\times}10^2$	$3.3{\times}10^3$	$2.9{\times}10^3$	2ms	
4 ranks	$1.7{\times}10^4$	$8.1{\times}10^4$	$2.1{\times}10^4$	$2.7{\times}10^5$	$2.4{\times}10^5$	1.1s	
5 ranks	$1.2{\times}10^6$	$1.1{\times}10^7$	$1.4{\times}10^6$	$3.6{\times}10^7$	$3.2{\times}10^7$	43.8s	

Figure 3: (Left) Dimensions of games and runtime of decomposition algorithm (Theorem 2). (Right) Payoffs that can be reached using an EFCE, EFCCE, or NFCCE in 3-rank Goofspiel.
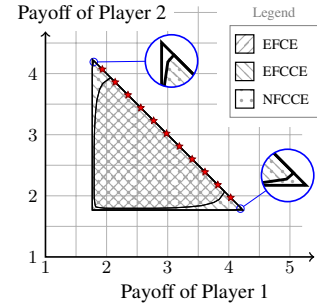
was averaged over 100 independent runs. Our decomposition algorithm performs well, and is able to scale to the largest game (Goofspiel with $k = 5$ ranks, which has 3.6×10^7 relevant sequence pairs). In Figure 3(right) we used the characterization $\Xi = \mathcal{V}$ to compute the set of all payoffs that can be reached by an EFCE, EFCCE, or NFCCE in 3-rank Goofspiel. The sets are highly non-trivial, and reinforce the observation that the behaviors and incentives that can be induced through extensive-form correlation are subtle and complex (Farina et al. 2019a). The sets of reachable payoff vectors was computed using *Polymake*, a library for computational polyhedral geometry (Gawrilow and Joswig 2000; Assarf et al. 2017).

We also implemented the linear programming formulation of EFCE described by von Stengel and Forges (2008), and the scalable regret minimization algorithm of Farina et al. (2019b). We use the Gurobi commercial linear programming solver to solve the linear program formulation. As the game size increases, the barrier algorithm is the only algorithm that can solve the linear program. However, even that quickly becomes impractical. In the largest game, Gurobi uses roughly 200GB of memory, spends approximately 90 minutes to precondition the linear program, and requires slightly more than 20 minutes to perform each iteration of the barrier method using 30 threads. The regret minimization scales significantly more favorable in the large game. It requires roughly 6 seconds per iteration, and reaches 10^{-2} infeasibility in 4 minutes, 10^{-3} infeasibility in 12 minutes, and 10^{-4} infeasibility in 110 minutes. Additional data about the experiment is available in Appendix C.

6 Conclusions

We showed that an optimal extensive-form correlated equilibrium, extensive-form coarse correlated equilibrium, and normal-form coarse correlated equilibrium can be computed in polynomial time in two-player perfect-recall games that satisfy a certain *triangle-freeness* condition that we introduced and that can be checked in polynomial time. To show that such equilibria can be found in polynomial time, we gave and combined several results that may be of independent interest: (1) the existence of a scaled-extension-based structural decomposition for the von Stengel-Forges polytope of the game, (2) a characterization of when the von Stengel-Forges polytope coincides with the polytope of correlation plans, and (3) a result about the integrality of the vertices of the von Stengel-Forges polytope in triangle-free games.

Broader Impact

In this paper we give a *positive* complexity result, showing that optimal equilibrium according to three important extensive-form imperfect-information game correlated solution concepts can be computed efficiently in settings—two-player games with public chance moves—where it was generally believed to be impossible. In fact, we showed that this can be done more broadly: in all games where a certain triangle-freeness condition holds. Correlated solution concepts have many advantages. First, they enable incentive-compatible coordination of agents. Such coordination is achieved via incentives, rather than forcing: mediators in correlated solution concepts are only able to recommend behavior, but not force it. So, it is up to the mediator to come up with a correlated distribution of recommendations such that no agent has incentive to deviate from the recommendations. Second, in some general-sum interactions these solution concepts are known to enable significantly higher social welfare than Nash equilibrium, while at the same time sidestepping some of the other shortcomings of Nash equilibrium (for example, some equilibrium selection issues). In this paper, we are particularly interested in *optimal* correlated equilibria. In other words, our technology can empower the system designer (mediator) to select, among the infinite number of correlated equilibria of the game, one that maximizes a given objective. For example, this technology could be used to find correlated equilibria than maximize social welfare, leading to highest societal good. However, like most technology, our technology has potential for abuse. If used maliciously, the ability to select particular correlated equilibria could be used to *minimize* social welfare, maximize only one of the agent's utility, or minimize all others' utilities—thereby furthering existing inequality or creating new inequality.

References

Ashlagi, I.; Monderer, D.; and Tennenholtz, M. 2008. On the value of correlation. *Journal of Artificial Intelligence Research* 33: 575–613.

Assarf, B.; Gawrilow, E.; Herr, K.; Joswig, M.; Lorenz, B.; Paffenholz, A.; and Rehn, T. 2017. Computing convex hulls and counting integer points with polymake. *Mathematical Programming Computation* 9(1): 1–38. ISSN 1867-2957. doi:10.1007/s12532-016-0104-z.

Aumann, R. 1974. Subjectivity and Correlation in Randomized Strategies. *Journal of Mathematical Economics* 1: 67–96.

Bowling, M.; Burch, N.; Johanson, M.; and Tammelin, O. 2015. Heads-up Limit Hold'em Poker is Solved. *Science* 347(6218).

Brown, N.; and Sandholm, T. 2017a. Safe and nested subgame solving for imperfect-information games. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*.

Brown, N.; and Sandholm, T. 2017b. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science* eaao1733.

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

Celli, A.; Coniglio, S.; and Gatti, N. 2019. Computing optimal ex ante correlated equilibria in two-player sequential games. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.

Celli, A.; Marchesi, A.; Bianchi, T.; and Gatti, N. 2019. Learning to Correlate in Multi-Player General-Sum Sequential Games.

Farina, G.; Bianchi, T.; and Sandholm, T. 2020. Coarse Correlation in Extensive-Form Games. In *AAAI Conference on Artificial Intelligence*.

Farina, G.; Kroer, C.; and Sandholm, T. 2019. Regret Circuits: Composability of Regret Minimizers. In *International Conference on Machine Learning*, 1863–1872.

Farina, G.; Ling, C. K.; Fang, F.; and Sandholm, T. 2019a. Correlation in Extensive-Form Games: Saddle-Point Formulation and Benchmarks. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Farina, G.; Ling, C. K.; Fang, F.; and Sandholm, T. 2019b. Efficient Regret Minimization Algorithm for Extensive-Form Correlated Equilibrium. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Gawrilow, E.; and Joswig, M. 2000. *Polymake: a Framework for Analyzing Convex Polytopes*, 43–73. Basel: Birkhäuser Basel. ISBN 978-3-0348-8438-9. doi:10.1007/978-3-0348-8438-9_2.

Gilboa, I.; and Zemel, E. 1989. Nash and Correlated Equilibria: Some Complexity Considerations. *Games and Economic Behavior* 1: 80–93.

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

Jiang, A.; and Leyton-Brown, K. 2011. Polynomial-time computation of exact correlated equilibrium in compact games. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*.

Lanctot, M. 2013. *Monte Carlo sampling and regret minimization for equilibrium computation and decision-making in large extensive form games*. Ph.D. thesis.

Lanctot, M.; Waugh, K.; Zinkevich, M.; and Bowling, M. 2009. Monte Carlo Sampling for Regret Minimization in Extensive Games. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*.

Moravčík, M.; Schmid, M.; Burch, N.; Lisý, V.; Morrill, D.; Bard, N.; Davis, T.; Waugh, K.; Johanson, M.; and Bowling, M. 2017. DeepStack: Expert-level artificial intelligence in heads-up no-limit poker. *Science* .

Moulin, H.; and Vial, J.-P. 1978. 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.

Papadimitriou, C.; and Roughgarden, T. 2005. Computing Equilibria in Multi-Player Games. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 82–91. Vancouver, BC, Canada: SIAM.

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

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

A Scaled-Extension-Based Structural Decomposition for $\mathcal{V}$

Triangle-Freeness

Lemma 2. *Consider a triangle-free game, let $(\sigma_1, \sigma_2) \in \Sigma_1 \bowtie \Sigma_2$, and let $I_1 \rightleftharpoons I_2$ be such that $\sigma(I_1) = \sigma_1, \sigma(I_2) = \sigma_2$. Then, at most one between the σ_1-rank of I_2 and the σ_2-rank of I_1 is strictly larger than 1.*

Proof. The results follows almost immediately from the definition of triangle-freeness. We prove the statement by contradiction. Let $(\sigma_1, \sigma_2) \in \Sigma_1 \bowtie \Sigma_2$ be a relevant sequence pair, and let information sets $I_1 \in \mathcal{I}_1, I_2 \in \mathcal{I}_2$ be such that $\sigma(I_1) = \sigma_1, \sigma(I_2) = \sigma_2$. Furthermore, assume that the σ_1-rank of I_2 is greater than 1, and at the same time the σ_2-rank of I_1 is greater than 1. Since the σ_2-rank of I_1 is greater than 1, there exists an information set $I_2' \in \mathcal{I}_2, \sigma(I_2') = \sigma_2$, distinct from I_2, such that $I_1 \rightleftharpoons I_2'$. Similarly, because the σ_1-rank of I_2 is greater than 1, there exists an information set $I_1' \in \mathcal{I}_1, \sigma(I_1') = \sigma_1$, distinct from I_1, such that $I_1' \rightleftharpoons I_2$. But then, we have found $I_1, I_1' \in \mathcal{I}_1$ and $I_2', I_2 \in \mathcal{I}_2$ such that $\sigma(I_1) = \sigma(I_2) = \sigma_1, \sigma(I_2') = \sigma(I_2) = \sigma_2$ such that $I_1 \rightleftharpoons I_2', I_1' \rightleftharpoons I_2$, and $I_1 \rightleftharpoons I_2$. So, the game is *not* triangle-free, contradiction. $\square$

Theorem 1. *A two-player extensive-form game with public chance moves is triangle-free.*

Proof. For contradiction, let I_1, I_2 be two distinct information sets for Player 1 such that $\sigma(I_1) = \sigma(I_2)$, let J_1, J_2 be two distinct information sets for Player 2 such that $\sigma(J_1) = \sigma(J_2)$, and assume that $I_1 \rightleftharpoons J_1, I_2 \rightleftharpoons J_2, I_1 \rightleftharpoons J_2$. By definition of connectedness, there exist nodes $u \in I_1, v \in J_1$ such that v is on the path from the root to u, or *vice versa*. Similarly, there exist nodes $u' \in I_2, v' \in J_2$ such that u' is on the path from the root to v', or *vice versa*. Let w be the lowest common ancestor of u and u'. It is not possible that $w = u$ or $w = u'$, because otherwise the parent sequences of I_1 and I_2 would be different. So w must be a strict ancestor of both u and u', and u and u' must be reached using *different* edges at w. Therefore, node w cannot belongs to Player 1, or otherwise it again would not be true that $\sigma(I_1) = \sigma(I_2)$. So, there are only two possible cases: either w belongs to Player 2, or it belongs to the chance player. We break the analysis accordingly.

- **First case:** w belongs to Player 2. From above, we know that u and u' are reached by following different branches at w. So, if both v and v' were strict descendants of w, they would need to be on two different branches of w (because they are connected to u and u' respectively), violating the condition $\sigma(J_1) = \sigma(J_2)$. So, at least one between v and v' is on the path from the root to w (inclusive). But then either v is an ancestor of v', or *vice versa*. Either case violates the hypothesis that $\sigma(J_1) = \sigma(J_2)$.

- **Second case:** w belongs to the chance player. If any between v and v' is an ancestor of w, then necessarily either v is an ancestor of v', or v' is an ancestor of v. Either case violates the condition $\sigma(J_1) = \sigma(J_2)$. So, both v and v' must be descendants of w. Because v is on the path from the root to u (or *vice versa*), and v' is on the path from the root to u' (or *vice versa*), then necessarily u, v and u', v' are on two different branches of the chance node w. To fix names, call a the action at w that must be taken to (eventually) reach u and v, and let b be the action that must be taken to (eventually) reach u' and v'. Now, we use the hypothesis that $I_1 \rightleftharpoons J_2$, that is, there exists $u'' \in I_1, v'' \in J_2$ such that u'' is on the path from the root to v'' or *vice versa*. Assume that u'' is on the path from the root to v''. Since u'' belongs to the same information set as u (that is, I_1), and since chance is public by hypothesis, then Player 1, when acting at u and u'', must have observed action a at w. In other words, the path from the root to u'' must pass through action a at w. But then, using the fact that u'' is on the path from the root to v'', this means that the path from the root to v'' passes through action a. However, the path from the root to v' passes through action b. Since chance is public, nodes v' and v'' cannot be in the same information set, because Player 2 is able to distinguish them by means of the observed chance outcome. We reached a contradiction. The symmetric case where v'' is on the path from the root to u'' is analogous. $\square$

Decomposition Algorithm

In this section, we provide pseudocode for the algorithm presented in Section 3. We will use the following conventions:

- Given a player $i \in \{1, 2\}$, we let $-i$ denote the opponent.
- We use the symbol $\sqcup$ to denote disjoint union.
- Given two infosets $I, I' \in \mathcal{I}_i$, we write $I \preceq I'$ if $\sigma(I') \succeq \sigma(I)$. We say that we iterate over a set $\mathcal{I} \subseteq \mathcal{I}_i$ *in top-down order* if, given any two $I, I' \in \mathcal{I}$ such that $I \preceq I'$, I appears before I' in the iteration.
- We use the observation that for all $I \in \mathcal{I}_1$ and $\sigma_2 \in \Sigma_2$, $I \bowtie \sigma_2$ if and only if $(I, a) \bowtie \sigma_2 \; \forall a \in A_I$. (A symmetric statement holds for $I \in \mathcal{I}_2$ and $\sigma_1 \in \Sigma_1$.)

Two Useful Subroutines We start by presenting two simple subroutines that capture fill-in step ④ of Figure 2 or equivalently Step 4(d) of Section 3. The two subroutines are symmetric and have the role of filling rows and columns of the correlation plans.

Algorithm 1: FILLOUTROW$((\sigma_1, \sigma_2), I_1, \mathcal{S}, \mathcal{D})$

Preconditions: $(\sigma_1, \sigma_2) \in \Sigma_1 \bowtie \Sigma_2, I_1 \in \mathcal{I}_1, \sigma(I_1) = \sigma_1, (\sigma_1, \sigma_2) \in \mathcal{S}$

1   **for** I_2 *such that* $\sigma(I_2) = \sigma_2$ *and* $\sigma_1 \bowtie I_2$ **do**
2   **if** $I_1 \rightleftharpoons I_2$ **then**
3   **for** $\sigma_2' \in \{(I_2, a) : a \in A_{I_2}\}$ **do**
        ▷ Fill (σ_1, σ_2') by summing up all entries $\{v[(I_1, a'), \sigma_2'] : a' \in A_{I_1}\}$ in accordance with the von Stengel-Forges constraints
4         $\mathcal{S} \leftarrow \mathcal{S} \sqcup \{(\sigma_1, \sigma_2')\}$
5         $\mathcal{D} \leftarrow \mathcal{D} \triangleleft^h \{1\}$ where $h : \boldsymbol{v} \mapsto \sum_{a' \in A_{I_1}} v[(I_1, a'), \sigma_2']$
6   **else**
        ▷ Fill all $\{v[\sigma_1, (I_2, a)] : a \in A_{I_2}\}$ by splitting $v[\sigma_1, \sigma_2]$ accordance with the von Stengel-Forges constraints
7         $\mathcal{S} \leftarrow \mathcal{S} \sqcup \{(\sigma_1, (I_2, a)) : a \in A_{I_2}\}$
8         $\mathcal{D} \leftarrow \mathcal{D} \triangleleft^h \Delta^{|A_{I_2}|}$ where $h : \boldsymbol{v} \mapsto v[\sigma_1, \sigma_2]$
9   **for** $\sigma_2' \in \{(I_2, a) : a \in A_{I_2}\}$ **do**
10   FILLOUTROW$((\sigma_1, \sigma_2'), I_1)$
11 **return** $(\mathcal{S}, \mathcal{D})$

Algorithm 2: FILLOUTCOLUMN$((\sigma_1, \sigma_2), I_2, \mathcal{S}, \mathcal{D})$

Preconditions: $(\sigma_1, \sigma_2) \in \Sigma_1 \bowtie \Sigma_2, I_2 \in \mathcal{I}_2, \sigma(I_2) = \sigma_2, (\sigma_1, \sigma_2) \in \mathcal{S}$

1   **for** I_1 *such that* $\sigma(I_1) = \sigma_1$ *and* $\sigma_2 \bowtie I_1$ **do**
2   **if** $I_1 \rightleftharpoons I_2$ **then**
3   **for** $\sigma' \in \{(I_1, a) : a \in A_{I_1}\}$ **do**
        ▷ Fill (σ_1', σ_2) by summing up all entries $\{v[\sigma_1', (I_2, a')] : a' \in A_{I_2}\}$ in accordance with the von Stengel-Forges constraints
4         $\mathcal{S} \leftarrow \mathcal{S} \sqcup \{(\sigma_1', \sigma_2)\}$
5         $\mathcal{D} \leftarrow \mathcal{D} \triangleleft^h \{1\}$ where $h : \boldsymbol{v} \mapsto \sum_{a' \in A_{I_2}} v[\sigma_1', (I_2, a')]$
6   **else**
        ▷ Fill all $\{v[(I_1, a), \sigma_2] : a \in A_{I_1}\}$ by splitting $v[\sigma_1, \sigma_2]$ accordance with the von Stengel-Forges constraints
7         $\mathcal{S} \leftarrow \mathcal{S} \sqcup \{((I_1, a), \sigma_2) : a \in A_{I_1}\}$
8         $\mathcal{D} \leftarrow \mathcal{D} \triangleleft^h \Delta^{|A_{I_1}|}$ where $h : \boldsymbol{v} \mapsto v[\sigma_1, \sigma_2]$
9   **for** $\sigma' \in \{(I_1, a) : a \in A_{I_1}\}$ **do**
10   FILLOUTCOLUMN$((\sigma_1', \sigma_2), I_2)$
11 **return** $(\mathcal{S}, \mathcal{D})$

The following inductive contract will be important for the full algortihm.

Lemma 3 (Inductive contract for FILLOUTROW)**.** *Suppose that the following preconditions hold when* FILLOUTROW$((\sigma_1, \sigma_2), I_1, \mathcal{S}, \mathcal{D}))$ *is called:*

(Pre1) $(\sigma_1, \sigma_2) \in \Sigma_1 \bowtie \Sigma_2$;

(Pre2) $I_1 \in \mathcal{I}_1$ *is such that* $\sigma(I_1) = \sigma$;

(Pre3) $\mathcal{S}$ *contains only relevant sequence pairs and* $\mathcal{D}$ *consists of vectors indexed by exactly the indices in* $\mathcal{S}$;

(Pre4) $(\sigma_1, \sigma_2) \in \mathcal{S}$, *but* $(\sigma_1, \sigma_2') \notin \mathcal{S}$ *for all* $\sigma_2' \succ \sigma_2$;

(Pre5) *For all* $a \in I_1$ *and* $\sigma_2' \succeq \sigma_2$ *such that* $I_1 \bowtie \sigma_2'$, $((I_1, a), \sigma_2') \in \mathcal{S}$;

(Pre6) *If* $I_1 \bowtie \sigma_2$, *all* $\boldsymbol{v} \in \mathcal{D}$ *satisfy the von Stengel-Forges constraint* $v[\sigma_1, \sigma_2] = \sum_{a \in I_1} v[(I_1, a), \sigma_2]$;

(Pre7) *All* $\boldsymbol{v} \in \mathcal{D}$ *satisfy the von Stengel-Forges constraints*

$$v[(I, a), \sigma(I_2)] = \sum_{a' \in A_{I_2}} v[(I, a), (I_2, a')] \quad \forall a \in I_1, \text{ and } I_2 \in \mathcal{I}_2 : I_1 \bowtie I_2, \sigma(I_2) \succeq \sigma_2.$$

Then, the sets $(\mathcal{S}', \mathcal{D}')$ *returned by the call are such that*

(Post1) $\mathcal{S}'$ *contains only relevant sequence pairs and* $\mathcal{D}'$ *consists of vectors indexed by exactly the indices in* $\mathcal{S}'$;

(Post2) $\mathcal{S}' = \mathcal{S} \sqcup \{(\sigma_1, \sigma_2') : \sigma_2' \succ \sigma_2, \sigma \bowtie \sigma_2'\}$;

(Post3) *All* $\boldsymbol{v} \in \mathcal{D}'$ *satisfy the von Stengel-Forges constraints*

$$v[\sigma_1, \sigma(I_2)] = \sum_{a' \in A_{I_2}} v[\sigma_1, (I_2, a')] \quad \forall I_2 \in \mathcal{I}_2 : \sigma \bowtie I_2, \sigma(I_2) \succeq \sigma_2$$

and all von Stengel-Forges constraints

$$v[\sigma_1, \sigma_2'] = \sum_{a \in A_{I_1}} v[(I, a), \sigma_2'] \quad \forall \sigma_2' \in \Sigma_2 : \sigma_2' \bowtie I_1, \sigma_2' \succeq \sigma_2.$$

Proof. By induction.

- **Base case.** The base case corresponds to $\sigma_2 \in \Sigma_2$ such that no information set $I_2 \in \mathcal{I}_2 : \sigma(I_2) = \sigma_2 \wedge \sigma_1 \bowtie I_2$ exists. In that case, Algorithm 1 returns immediately, so (Post1) holds trivially from (Pre3). Since no I_2 such that $\sigma(I_2) = \sigma_2 \wedge \sigma_1 \bowtie I_2$ exists, no $\sigma_2' \succ \sigma_2$ such that $\sigma_1 \bowtie \sigma_2'$ exists, so (Post2) holds as well. The first set of constraints of (Post3) is empty, and the second set reduces to (Pre6).

- **Inductive step.** Suppose that the inductive hypothesis holds when $\sigma_2' \succ \sigma_2$. We will show that it holds when $\sigma_2' = \sigma_2$ as well. In order to use the inductive hypothesis, we first need to check that the preconditions are preserved at the time of the recursive call on Line 10. (Pre1) holds since $\sigma_1 \bowtie I_2$. (Pre2) holds trivially since σ does not chance. (Pre3) holds since we are updating $\mathcal{S}$ and $\mathcal{D}$ in tandem on lines 4, 5 and 7, 8. (Pre4) holds since by the time of the recursive call we have only filled in entries (σ_1, σ_2') where σ_2' is an immediate successor of σ_2. (Pre5) at Line 10 holds trivially, since it refers to a subset of the entries for which the condition held at the beginning of the call. (Pre6) holds because $I_1 \bowtie \sigma_2' \iff I_1 \rightleftharpoons I_2$. Hence, if $I_1 \bowtie \sigma_2'$ then Lines 4 and 5 must have run. (Pre7) at Line 10 holds trivially, since it refers to a subset of the constraints for which the condition held at the beginning of the call. Using the inductive hypothesis, (Post1), (Post2), and the second set of constraints in (Post3) follow immediately. The only constraints that are left to be verified are

$$v[\sigma_1, \sigma_2] = \sum_{a' \in A_{I_2}} v[\sigma_1, (I_2, a')] \quad \forall I_2 \in \mathcal{I}_2 : \sigma \bowtie I_2, \sigma(I_2) = \sigma_2. \tag{3}$$

That constraint is guaranteed by Lines 7 and 8 for all $I_2 \neq I_1$. So, we need to verify that it holds for all those I_2 such that $\sigma(I_2) = \sigma_2, \sigma \bowtie I_2$ and $I_1 \rightleftharpoons I_2$. Let I_2 be one such information set. Then, from Lines 4 and 5 we have that

$$v[\sigma_1, (I_2, a)] = \sum_{a' \in A_{I_1}} v[(I, a'), (I_2, a)] \quad \forall a \in A_{I_2}.$$

Summing the above equations across all $a \in A_{I_2}$ and using (Pre7) yields

$$\sum_{a \in A_{I_2}} v[\sigma_1, (I_2, a)] = \sum_{a \in A_{I_2}} \sum_{a' \in A_{I_1}} v[(I, a'), (I_2, a)]$$

$$= \sum_{a' \in A_{I_1}} \sum_{a \in A_{I_2}} v[(I, a'), (I_2, a)]$$

$$= \sum_{a' \in A_{I_1}} v[(I, a'), \sigma(I_2)]$$

$$= \sum_{a' \in A_{I_1}} v[(I, a'), \sigma_2],$$

where we used the hypothesis that $\sigma(I_2) = \sigma_2$ in the last equality. Finally, since $I_1 \rightleftharpoons I_2$ and $\sigma(I_2) = \sigma_2$, it must be $I_1 \bowtie \sigma_2$ and so, using (Pre6), we obtain that

$$\sum_{a \in A_{I_2}} v[\sigma_1, (I_2, a)] = v[\sigma_1, \sigma_2],$$

completing the proof of Equation (3). So, (Post3) holds as well and the proof of the inductive step is complete. $\square$

The inductive contract for FILLOUTCOLUMN is symmetric and we omit it.

Algorithm 3: DECOMPOSE$((\sigma_1, \sigma_2), \mathcal{S}, \mathcal{D})$

Preconditions : $(\sigma_1, \sigma_2) \in \Sigma_1 \bowtie \Sigma_2, (\sigma_1, \sigma_2) \in \mathcal{S}$

1   $B \leftarrow \emptyset$

2   **for** *all* $i \in \{1, 2\}, I \in \mathcal{I}_i, \sigma(I) = \sigma_i, \sigma_{-i} \bowtie I$ **do**

3   **if** *the σ_{-i}-rank of I is 0* **then**

4       $B \leftarrow B \sqcup I$

5   **for** $(I_1, I_2) \in \mathcal{I}_1 \times \mathcal{I}_2$ *such that* $\sigma(I_1) = \sigma_1, \sigma(I_2) = \sigma_2, I_1 \rightleftharpoons I_2$ **do**

6   **if** *the σ_2-rank of I_1 is $\geq$ the σ_1-rank of I_2* **then**

7       $B \leftarrow B \sqcup I_1$

8   **else**

9       $B \leftarrow B \sqcup I_2$

10   **for** $I \in B$ **do**

11   **if** $I \in \mathcal{I}_1$ **then**

       $\triangleright$ Fill all $\{v[(I, a), \sigma_2] : a \in A_I\}$ by splitting $v[\sigma_1, \sigma_2]$ accordance with the von Stengel-Forges constraints

12       $\mathcal{S} \leftarrow \mathcal{S} \sqcup \{((I, a), \sigma_2) : a \in A_I\}$

13       $\mathcal{D} \leftarrow \mathcal{D} \triangleleft^h \Delta^{|A_I|}$ where $h : \boldsymbol{v} \mapsto v[\sigma_1, \sigma_2]$

       $\triangleright$ Recursive call

14   **for** $\sigma_1' \in \{(I, a) : a \in A_I\}$ **do**

15   DECOMPOSE$((\sigma_1', \sigma_2), \mathcal{S}, \mathcal{D})$

       $\triangleright$ Fill a portion of the row for σ_1

16   **for** $I_2 \in \mathcal{I}_2 : \sigma_1 \bowtie I_2, \sigma(I_2) = \sigma_2$ **do**

17   **for** $\sigma_2' \in \{(I_2, a') : a' \in A_{I_2}\}$ **do**

          $\triangleright$ Fill (σ_1, σ_2') by summing up all entries $\{v[(I, a'), \sigma_2'] : a' \in A_I\}$ in accordance with the von Stengel-Forges constraints

18           $\mathcal{S} \leftarrow \mathcal{S} \sqcup \{(\sigma_1, \sigma_2')\}$

19           $\mathcal{D} \leftarrow \mathcal{D} \triangleleft^h \{1\}$ where $h : \boldsymbol{v} \mapsto \sum_{a' \in A_I} v[(I, a'), \sigma_2']$

20   FILLOUTROW$((\sigma_1, \sigma_2'), I)$

21   **else**

       $\triangleright$ Fill all $\{v[\sigma_1, (I, a)] : a \in A_I\}$ by splitting $v[\sigma_1, \sigma_2]$ accordance with the von Stengel-Forges constraints

22       $\mathcal{S} \leftarrow \mathcal{S} \sqcup \{(\sigma_1, (I, a)) : a \in A_I\}$

23       $\mathcal{D} \leftarrow \mathcal{D} \triangleleft^h \Delta^{|A_I|}$ where $h : \boldsymbol{v} \mapsto v[\sigma_1, \sigma_2]$

       $\triangleright$ Recursive call

24   **for** $\sigma_2' \in \{(I, a) : a \in A_I\}$ **do**

25   DECOMPOSE$((\sigma_1, \sigma_2'), \mathcal{S}, \mathcal{D})$

       $\triangleright$ Fill a portion of the column for σ_2

26   **for** $I_1 \in \mathcal{I}_1 : \sigma_2 \bowtie I_1, \sigma(I_1) = \sigma_1$ **do**

27   **for** $\sigma_1' \in \{(I_1, a') : a' \in A_{I_1}\}$ **do**

          $\triangleright$ Fill (σ_1', σ_2) by summing up all entries $\{v[\sigma_1', (I, a')] : a' \in A_I\}$ in accordance with the von Stengel-Forges constraints

28           $\mathcal{S} \leftarrow \mathcal{S} \sqcup \{(\sigma_1', \sigma_2)\}$

29           $\mathcal{D} \leftarrow \mathcal{D} \triangleleft^h \{1\}$ where $h : \boldsymbol{v} \mapsto \sum_{a' \in A_I} v[\sigma_1', (I, a')]$

30   FILLOUTCOLUMN$((\sigma_1', \sigma_2), I)$

31   **return** $(\mathcal{S}, \mathcal{D})$

The Full Algorithm

Lemma 4 (Inductive contract for DECOMPOSE). *Assume that at the beginning of each call to* DECOMPOSE$((\sigma_1, \sigma_2), \mathcal{S}, \mathcal{D})$ *the following conditions hold*

  (Pre1) $\mathcal{S}$ *contains only relevant sequence pairs and* $\mathcal{D}$ *consists of vectors indexed by exactly the indices in* $\mathcal{S}$.

  (Pre2) $\mathcal{S}$ *does not contain any relevant sequence pairs which are descendants of* (σ_1, σ_2), *with the only exception of* (σ_1, σ_2) *itself. In formulas,*

$$\mathcal{S} \cap \{(\sigma_1', \sigma_2') \in \Sigma_1 \times \Sigma_2 : \sigma_1' \succeq \sigma_1, \sigma_2' \succeq \sigma_2\} = \{(\sigma_1, \sigma_2)\}.$$

Then, at the end of the call, the returned set $(\mathcal{S}', \mathcal{D}')$ *are such that*

(Post1) $\mathcal{S}'$ *contains only relevant sequence pairs and* $\mathcal{D}'$ *consists of vectors* $\boldsymbol{v}$ *indexed by exactly the indices in* $\mathcal{S}'$.

  (Post2) *The call has filled in exactly all relevant sequence pair indices that are descendants of* (σ_1, σ_2) *(except for* (σ_1, σ_2) *itself, which was already filled in). In formulas,*

$$\mathcal{S}' = \mathcal{S} \sqcup \{(\sigma_1', \sigma_2') \in \Sigma_1 \times \Sigma_2 : \sigma_1' \succeq \sigma_1, \sigma_2' \succeq \sigma_2, (\sigma_1', \sigma_2') \neq (\sigma_1, \sigma_2), \sigma_1' \bowtie \sigma_2'\}.$$

  (Post3) $\mathcal{D}'$ *satisfies the subset of von Stengel-Forges constraints*

$$\sum_{a\in A_I} v[(I,a),\sigma_2'] = v[\sigma(I),\sigma_2'] \quad \forall \sigma_2' \succeq \sigma_2, I \in \mathcal{I}_1 \ s.t. \ \sigma_2' \bowtie I, \sigma(I) \succeq \sigma_1$$

$$\sum_{a\in A_J} v[\sigma_1',(J,a)] = v[\sigma_1',\sigma(J)] \quad \forall \sigma_1' \succeq \sigma_1, J \in \mathcal{I}_2 \ s.t. \ \sigma_1' \bowtie J, \sigma(J) \succeq \sigma_2.$$

Proof. By induction.

- **Base case.** The base case is any (σ_1,σ_2) such that there is no $\sigma_1' \succeq \sigma_1, \sigma_2' \succeq \sigma_2, \sigma_1' \bowtie \sigma_2'$. In that case, the set B is empty, so the algorithm terminates immediately without modifying the sets $\mathcal{S}$ and $\mathcal{D}$. Consequently, (Post1) and (Post2) hold trivially from (Pre1) and (Pre2). (Post3) reduces to an empty set of constraints, so (Post3) holds as well.

- **Inductive step.** In order to use the inductive hypothesis, we will need to prove that the preconditions for DECOMPOSE hold on Lines 15 and 25. We will focus on Line 15 ($I \in \mathcal{I}_1$), as the analysis for the other case ($I \in \mathcal{I}_2$) is symmetric. (Pre1) clearly holds, since we always update $\mathcal{S}$ and $\mathcal{D}$ in tandem. Since all iterations of the **for** loop on Line 10 touch different information sets, at the time of the recursive call on Line 15, and given (Post2) for all previous recursive calls, the only relevant sequence pairs (σ_1'',σ_2'') such that $\sigma_1'' \succeq \sigma_1', \sigma_2'' \succeq \sigma_2$ that have been filled are the ones on Lines 12 and 13. So, (Pre2) holds.

  We now check that the preconditions for FILLOUTROW hold at Line 20. (Pre1), (Pre2), (Pre3), and (Pre4) are trivial. (Pre5) and (Pre7) are guaranteed by (Post2) and (Post3) of DECOMPOSE applied to Line 15. (Pre6) holds because of Lines 18 and 19.

  Using the inductive contracts of FILLOUTROW, FILLOUTCOLUMN and DECOMPOSE for the recursive calls, we now show that all postconditions hold at the end of the call. (Post1) is trivial since we always update $\mathcal{S}$ and $\mathcal{D}$ together. (Post2) holds by keeping track of what entries are filled in Lines 12, 13, 18, 19, 22, 23, 28, 29, as well as those filled in the calls to FILLOUTROW, FILLOUTCOLUMN and DECOMPOSE, as regulated by postcondition (Post2) in the inductive contracts of the functions. In order to verify (Post3), we need to verify that the constraints that are not already guaranteed by the recursive calls hold. In particular, we need to verify that

$$Ⓐ \sum_{a\in A_I} v[(I,a),\sigma_2] = v[\sigma_1,\sigma_2] \quad \forall I \in \mathcal{I}_1 \ s.t. \ \sigma_2 \bowtie I, \sigma(I) = \sigma_1, I \notin B$$

$$Ⓑ \sum_{a\in A_J} v[\sigma_1,(J,a)] = v[\sigma_1,\sigma_2] \quad \forall J \in \mathcal{I}_2 \ s.t. \ \sigma_1 \bowtie J, \sigma(J) = \sigma_2, J \notin B.$$

We will show that constraints Ⓐ hold; the proof for Ⓑ is symmetric. Using Lemma 2 together with the definition of B (Lines 1-9), any information set $I \in \mathcal{I}_i : \sigma(I) = \sigma_i, \sigma_{-i} \bowtie I$ that is not in B must have σ_{-i}-rank *exactly* 1. Let $I \in \mathcal{I}_1$ be such that $\sigma_2 \bowtie I, \sigma(I) = \sigma_1, I \notin B$, as required in Ⓐ. Since the σ_2-rank of I is 1, let J be the only information set in $\mathcal{I}_2$ such that $I \rightleftharpoons J, \sigma(J) = \sigma_2$. Note that $J \in B$. The entries $v[(I,a),\sigma_2] : a \in A_I$ were filled in Lines 28 and 29 when the **for** loop picked up $J \in B$. So, in particular,

$$v[(I,a),\sigma_2] = \sum_{a'\in A_J} v[(I,a),(J,a')] \quad \forall a \in A_I.$$

Summing the above equations across $a \in A_I$, we obtain

$$\sum_{a\in A_I} v[(I,a),\sigma_2] = \sum_{a\in A_I}\sum_{a'\in A_J} v[(I,a),(J,a')]$$
$$= \sum_{a'\in A_J}\sum_{a\in A_I} v[(I,a),(J,a')]$$
$$= \sum_{a'\in A_J} v[\sigma_1,(J,a')]$$
$$= v[\sigma_1,\sigma_2],$$

where the last equation follows from the way the entries $v[\sigma_1,(J,a')] : a' \in A_J$ were filled in (Lines 22 and 23). This shows that the set of constraints Ⓐ hold. $\square$

Theorem 2. *The von Stengel-Forges polytope $\mathcal{V}$ of a two-player perfect-recall triangle-free EFG can be expressed via a sequence of scaled extensions with simplexes and singleton sets:*

$$\mathcal{V} = \{1\} \stackrel{h_1}{\triangleleft} \mathcal{X}_1 \stackrel{h_2}{\triangleleft} \mathcal{X}_2 \stackrel{h_3}{\triangleleft} \cdots \stackrel{h_n}{\triangleleft} \mathcal{X}_n, \tag{2}$$

where, for $i = 1,\ldots,n$, either $\mathcal{X}_i = \Delta^{s_i}$ or $\mathcal{X}_i = \{1\}$ and h_i is a linear function. Furthermore, an exact algorithm exists to compute such expression in linear time in the dimensionality of $\mathcal{V}$, and so, in time at most quadratic in the size of the game.

Proof. The correctness of the algorithm follow from (Post3) in the inductive contract. Every time the set of partially-filled-in vectors $\mathcal{D}$ gets extended, it is extended with either the singleton set $\{1\}$ or a simplex. In either case the nonnegative affine functions h used are linear. So, the decomposition structure is as in the statement. Finally, since the overhead of each call (on top of the recursive calls) is linear in the number of relevant sequence pairs $(\sigma, \tau) \in \Sigma_1 \bowtie \Sigma_2$ that are filled, and each relevant sequence pair is filled only once, the complexity of the algorithm is linear in the number of relevant sequence pairs. □

B Relationship Between $\mathcal{V}$ and Ξ

Preliminaries: Definition of the Polytope of Correlation Plans

Let $\Pi_i(\sigma)$ denote the subset of reduced-normal-form plans Π_i for Player i prescribe all actions of Player i on the path from the root of the game down to the information set-action pair σ (if $\sigma =$, assign $\Pi_i(\varnothing) = \Pi_i$). The transformation from a correlated distribution μ to its correlation plan representation is achieved using a linear function

$$f : \Delta^{|\Pi_1 \times \Pi_2|} \to \mathbb{R}_{\geq 0}^{|\Sigma_1 \bowtie \Sigma_2|}.$$

Specifically, f takes a generic distribution μ over $\Pi_1 \times \Pi_2$ and maps to the vector $\boldsymbol{\xi} = f(\mu)$, called a *correlation plan*, whose components are

$$\xi[\sigma_1, \sigma_2] := \sum_{\pi_1 \in \Pi_1(\sigma_1)} \sum_{\pi_2 \in \Pi_2(\sigma_2)} \mu(\pi_1, \pi_2) \qquad \forall (\sigma_1, \sigma_2) \in \Sigma_1 \bowtie \Sigma_2. \tag{4}$$

The set of all valid correlation plans, Ξ, is defined as the image $\mathrm{Im}\, f$ of f as the distribution μ takes any possible value in $\Delta^{|\Pi_1 \times \Pi_2|}$.

Remark 2. *Since f sums up distinct entries from the distribution μ, all entries in $\boldsymbol{\xi} = f(\mu)$ are in the range $[0, 1]$.*

Proofs

Lemma 5. *Let $\mathbf{1}_{(\pi_1, \pi_2)} \in \Delta^{|\Pi_1 \times \Pi_2|}$ denote the distribution over $\Pi_1 \times \Pi_2$ that assigns mass 1 to the pair (π_1, π_2), and mass 0 to any other pair of reduced-normal-form plans. Then,*

$$\Xi = \mathrm{co}\{f(\mathbf{1}_{(\pi_1, \pi_2)}) : \pi_1 \in \Pi_1, \pi_2 \in \Pi_2\}.$$

Proof. The "deterministic" distributions $\mathbf{1}_{(\pi_1, \pi_2)}$ are the vertices of $\Delta^{|\Pi_1 \times \Pi_2|}$, so, in particular,

$$\Delta^{|\Pi_1 \times \Pi_2|} = \mathrm{co}\{\mathbf{1}_{(\pi_1, \pi_2)} : \pi_1 \in \Pi_1, \pi_2 \in \Pi_2\}.$$

Since by definition $\Xi = \mathrm{Im}\, f$, and f is a linear function, the images (under f) of the $\mathbf{1}_{(\pi_1, \pi_2)}$ are a convex basis for Ξ, which is exactly the statement. □

Lemma 6. *Let $\boldsymbol{v} \in \mathcal{V}$. For all $\sigma_1 \in \Sigma_1$ such that $v[\sigma_1, \varnothing] = 0$, $v[\sigma_1, \sigma_2] = 0$ for all $\sigma_2 \bowtie \sigma_1$. Similarly, for all $\sigma_2 \in \Sigma_2$ such that $v[\varnothing, \sigma_2] = 0$, $v[\sigma_1, \sigma_2] = 0$ for all $\sigma_1 \bowtie \sigma_2$.*

Proof. We prove the theorem by induction on the *depth* of the sequences σ_1 and σ_2. The depth $\mathrm{depth}(\sigma)$ of a generic sequence $\sigma = (I, a) \in \Sigma_i$ of Player i is defined as the number of actions that Player i plays on the path from the root of the tree down to action a at information set I included. Conventionally, we let the depth of the empty sequence be 0.

Take $\sigma_1 \in \Sigma_1$ such that $v[\sigma_1, \varnothing] = 0$. For σ_2 of depth 0 (that is, $\sigma_2 = \varnothing$), clearly $v[\sigma_1, \sigma_2] = 0$. For the inductive step, suppose that $v[\sigma_1, \sigma_2] = 0$ for all $\sigma_2 \in \Sigma_2, \sigma_1 \bowtie \sigma_2$ such that $\mathrm{depth}(\sigma_2) \leq d_2$. We will show that $v[\sigma_2, \sigma_2] = 0$ for $\mathrm{depth}(\sigma_2) \leq d_2 + 1$. Indeed, let $(I, a') = \sigma_2 \bowtie \sigma_1$ of depth $d_2 + 1$. Since $\boldsymbol{v} \in \mathcal{V}$, in particular the von Stengel-Forges constraint $\sum_{a \in A_I} v[\sigma_1, (I, a)] = v[\sigma_1, \sigma(I)]$ must hold. The depth of $\sigma(I)$ is d_2, so by the inductive hypothesis, it must be $v[\sigma_1, \sigma(I)] = 0$, and therefore $\sum_{a \in A_I} v[\sigma_1, (I, a)] = 0$. But all entries of $\boldsymbol{v}$ are nonnegative, so it must be $v[\sigma_1, (I, a)] = 0$ for all $a \in A_I$, and in particular for $(I, a') = \sigma_2$. This completes the proof by induction.

The proof for the second part is analogous. □

Lemma 7. *Let $\boldsymbol{v} \in \mathcal{V}$ have integer $\{0, 1\}$ coordinates. Then, for all $(\sigma_1, \sigma_2) \in \Sigma_1 \bowtie \Sigma_2$, it holds that*

$$v[\sigma_1, \sigma_2] = v[\sigma_1, \varnothing] \cdot v[\varnothing, \sigma_2].$$

Proof. We prove the theorem by induction on the depth of the sequences, similarly to Lemma 6.

The base case for the induction proof corresponds to the case where σ_1 and σ_2 both have depth 0, that is, $\sigma_1 = \sigma_2 = \varnothing$. In that case, the theorem is clearly true, because $v[\varnothing, \varnothing] = 1$ as part of the von Stengel-Forges constraints (1).

Now, suppose that the statement holds as long as $\mathrm{depth}(\sigma_1), \mathrm{depth}(\sigma_2) \leq d$. We will show that the statement will hold for any $(\sigma_1, \sigma_2) \in \Sigma_1 \bowtie \Sigma_2$ such that $\mathrm{depth}(\sigma_1), \mathrm{depth}(\sigma_2) \leq d + 1$. Indeed, consider $(\sigma_1, \sigma_2) \in \Sigma_1 \bowtie \Sigma_2$ such that $\mathrm{depth}(\sigma_1), \mathrm{depth}(\sigma_2) \leq d + 1$. If any of the sequences is the empty sequence, the statements holds trivially, so assume that neither is the empty sequence and in particular $\sigma_1 = (I, a), \sigma_2 = (J, b)$. If $v[\sigma_1, \varnothing] = 0$, then from Lemma 6 $v[\sigma_1, \sigma_2] = 0$ and

the statement holds. Similarly, if $v[\varnothing, \sigma_2] = 0$, then $v[\sigma_1, \sigma_2] = 0$, and the statement holds. Hence, the only remaining case given the integrality assumption on the coordinates of $\boldsymbol{v}$ is $v[\sigma_1, \varnothing] = v[\varnothing, \sigma_2] = 1$.

From the von Stengel-Forges constraints, $v[\sigma(I), \varnothing] = \sum_{a' \in A_I} v[(I, a'), \varnothing] = 1 + \sum_{a' \in A_I, a' \neq a} v[(I, a'), \varnothing] \geq 1$. Hence, because all entries of $\boldsymbol{v}$ are in $\{0, 1\}$, it must be $v[\sigma(I), \varnothing] = 1$ and $v[(I, a'), \varnothing] = 0$ for all $a' \in A_I, a' \neq a$. With a similar argument we conclude that $v[\varnothing, \sigma(J)] = 1$ and $v[\varnothing, (J, b')] = 0$ for all $b' \in A_J, b \neq b'$. Using the inductive hypothesis, $v[\sigma(I), \sigma(J)] = v[\sigma(I), \varnothing] \cdot v[\varnothing, \sigma(J)] = 1$.

Now, using the von Stengel-Forges constraints together with the equality $v[\sigma(I), \sigma(J)] = 1$ we just proved, we conclude that

$$\sum_{a' \in A_I} \sum_{b' \in A_J} v[(I, a'), (J, b')] = 1. \tag{5}$$

On the other hand, since $v[(I, a'), \varnothing] = 0$ for all $a' \in A_I, a' \neq a$ and $v[\varnothing, (J, b')] = 0$ for all $b' \in A_J, b' \neq b$, from Lemma 6 we have that

$$a' \neq a \vee b' \neq b \implies v[(I, a'), (J, b')] = 0. \tag{6}$$

From (6) and (5), we conclude that $v[(I, a), (J, b)] = v[\sigma_1, \sigma_2] = 1 = v[\sigma_1, \varnothing] \cdot v[\varnothing, \sigma_2]$, as we wanted to show. $\qquad\square$

Theorem 3. *Let Γ be a two-player perfect-recall extensive-form game, let $\mathcal{V}$ be its von Stengel-Forges polytope, and let Ξ be its polytope of correlation plans. Then, $\Xi = \mathcal{V}$ if and only if all vertices of $\mathcal{V}$ have integer $\{0, 1\}$ coordinates.*

Proof. We prove the two implications separately.

($\Rightarrow$) We start by proving that if $\Xi = \mathcal{V}$, then all vertices of $\mathcal{V}$ have integer $\{0, 1\}$ coordinates. Since $\mathcal{V} = \Xi$ by hypothesis, from 5 we can write

$$\mathcal{V} = \mathrm{co}\{f(\mathbf{1}_{(\pi_1, \pi_2)}) : \pi_1 \in \Pi_1, \pi_2 \in \Pi_2\}.$$

So, to prove this direction it is enough to show that $f(\mathbf{1}_{(\pi_1, \pi_2)})$ has integer $\{0, 1\}$ coordinates for all $(\pi_1, \pi_2) \in \Pi_1 \times \Pi_2$. To see that, we use the definition (4): each entry in $f(\mathbf{1}_{(\pi_1, \pi_2)})$ is the sum of distinct entries of $\mathbf{1}_{(\pi_1, \pi_2)}$. Given that by definition $\mathbf{1}_{(\pi_1, \pi_2)}$ has exactly one entry with value 1 and $|\Pi_1 \times \Pi_2| - 1$ entries with value 0, we conclude that all coordinates of $f(\mathbf{1}_{(\pi_1, \pi_2)})$ are in $\{0, 1\}$.

($\Leftarrow$) We now show that if all vertices of $\mathcal{V}$ have integer $\{0, 1\}$ coordinates, then $\mathcal{V} \subseteq \Xi$. This is enough, since the reverse inclusion, $\mathcal{V} \supseteq \Xi$, is trivial and already known (von Stengel and Forges 2008). Let $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$ be the vertices of $\mathcal{V}$. To conclude that $\mathcal{V} \subseteq \Xi$, we will prove that $\boldsymbol{v}_i \in \Xi$ for all $i = 1, \ldots, n$. This will be sufficient since both $\mathcal{V}$ and Ξ are convex.

Let $\boldsymbol{v} \in \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$ be any vertex of $\mathcal{V}$. By hypothesis, $v[\sigma_1, \sigma_2] \in \{0, 1\}$ for all $(\sigma_1, \sigma_2) \in \Sigma_1 \bowtie \Sigma_2$. Because $\boldsymbol{v}$ satisfies the von Stengel-Forges constraints and furthermore $\boldsymbol{v}$ has $\{0, 1\}$ entries by hypothesis, the two vectors $\boldsymbol{q}_1, \boldsymbol{q}_2$ defined according to $q_1[\sigma_1] = v[\sigma_1, \varnothing]$ $(\sigma_1 \in \Sigma_1)$ and $q_2[\sigma_2] = v[\varnothing, \sigma_2]$ $(\sigma_2 \in \Sigma_2)$ are *pure* sequence-form strategies. Now, let π_1^* and π_2^* be the reduced-normal form plans corresponding to $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$, respectively. We will show that $\boldsymbol{v} = f(\mathbf{1}_{(\pi_1^*, \pi_2^*)})$, which will immediately imply that $\boldsymbol{v} \in \Xi$ using Lemma 5.

Since $\mathbf{1}_{(\pi_1^*, \pi_2^*)}$ has exactly one positive entry with value 1 in the position corresponding to (π_1^*, π_2^*), by definition of the linear map f, for any $(\sigma_1, \sigma_2) \in \Sigma_1 \bowtie \Sigma_2$,

$$f(\mathbf{1}_{(\pi_1^*, \pi_2^*)})[\sigma_1, \sigma_2] = \mathbb{1}[\sigma_1 \in \Pi_1(\sigma_1)] \cdot \mathbb{1}[\sigma_2 \in \Pi_2(\sigma_2)]. \tag{7}$$

So, using the known properties of pure sequence-form strategies, we obtain

$$f(\mathbf{1}_{(\pi_1^*, \pi_2^*)})[\sigma_1, \sigma_2] = q_1[\sigma_1] \cdot q_2[\sigma_2] = v[\sigma_1, \varnothing] \cdot v[\varnothing, \sigma_2] = v[\sigma_1, \sigma_2],$$

where the last equality follows from Lemma 7. Since the equality holds for any $(\sigma_1, \sigma_2) \in \Sigma_1 \bowtie \Sigma_2$, we have that $\boldsymbol{v} = f(\mathbf{1}_{(\pi_1^*, \pi_2^*)})$. $\qquad\square$

Lemma 1. *Let $\mathcal{X}, \mathcal{Y}$, and h be as in Definition 1. If $\mathcal{X}$ is a convex polytope with vertices $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, and $\mathcal{Y}$ is a convex polytope with vertices $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m\}$, then $\mathcal{X} \triangleleft^h \mathcal{Y}$ is a convex polytope whose vertices are a nonempty subset of $\{(\boldsymbol{x}_i, h(\boldsymbol{x}_i)\boldsymbol{y}_j) : i \in \{1, \ldots, n\}, j \in \{1, \ldots, m\}\}$.*

Proof. Take any point $\boldsymbol{z} \in \mathcal{X} \triangleleft^h \mathcal{Y}$. By definition of scaled, extension, there exist $\boldsymbol{x} \in \mathcal{X}, \boldsymbol{y} \in \mathcal{Y}$ such that $\boldsymbol{z} = (\boldsymbol{x}, h(\boldsymbol{x})\boldsymbol{y})$. Since $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ are the vertices of $\mathcal{X}$, $\boldsymbol{x}$ can be written as a convex combination $\boldsymbol{x} = \sum_{i=1}^n \lambda_i \boldsymbol{x}_i$ where $(\lambda_1, \ldots, \lambda_n) \in \Delta^n$.

Similarly, $\boldsymbol{y} = \sum_{i=1}^{m} \mu_i \boldsymbol{y}_i$ for some $(\mu_1, \ldots, \mu_m) \in \Delta^m$. Hence, using the hypothesis that h is affine, we can write

$$\boldsymbol{z} = (\boldsymbol{x}, h(\boldsymbol{x})\boldsymbol{y})$$

$$= \left(\sum_{i=1}^{n} \lambda_i \boldsymbol{x}_i, h\left(\sum_{i=1}^{n} \lambda_i \boldsymbol{x}_i \right) \sum_{j=1}^{m} \mu_j \boldsymbol{y}_j \right)$$

$$= \left(\sum_{i=1}^{n} \lambda_i \boldsymbol{x}_i, \left(\sum_{i=1}^{n} \lambda_i h(\boldsymbol{x}_i) \right) \sum_{j=1}^{m} \mu_j \boldsymbol{y}_j \right)$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{m} \lambda_i \mu_j (\boldsymbol{x}_i, h(\boldsymbol{x}_i)\boldsymbol{y}_j).$$

Since $\lambda_i \mu_j \geq 0$ for all $i \in \{1, \ldots, n\}, j \in \{1, \ldots, m\}$ and $\sum_{i=1}^{n} \sum_{j=1}^{m} \lambda_i \mu_j = (\sum_{i=1}^{n} \lambda_i)(\sum_{j=1}^{m} \mu_j) = 1$, we conclude that $\boldsymbol{z} \in \mathrm{co}\{(\boldsymbol{x}_i, h(\boldsymbol{x}_i)\boldsymbol{y}_j) : i \in \{1, \ldots, n\}, j \in \{1, \ldots, m\}\}$. On the other hand, $(\boldsymbol{x}_i, h(\boldsymbol{x}_i)\boldsymbol{y}_j) \in \mathcal{X} \triangleleft^h \mathcal{Y}$, so

$$\mathcal{X} \overset{h}{\triangleleft} \mathcal{Y} = \mathrm{co}\{(\boldsymbol{x}_i, h(\boldsymbol{x}_i)\boldsymbol{y}_j) : i \in \{1, \ldots, n\}, j \in \{1, \ldots, m\}\}.$$

Since the vertices of a (nonempty) polytope are a (nonempty) subset of any convex basis for the polytope, the vertices of $\mathcal{X} \triangleleft^h \mathcal{Y}$ must be a nonempty subset of $\{(\boldsymbol{x}_i, h(\boldsymbol{x}_i)\boldsymbol{y}_j) : i \in \{1, \ldots, n\}, j \in \{1, \ldots, m\}\}$, which is the statement. $\square$

Theorem 4. *Let $\mathcal{V}$ be the von Stengel-Forges polytope of a two-player triangle-free game (Definition 3). All vertices of $\mathcal{V}$ have integer $\{0, 1\}$ coordinates.*

Proof. We prove the statement by induction over the scaled-extension-based decomposition

$$\mathcal{V} = \{1\} \overset{h_1}{\triangleleft} \mathcal{X}_1 \overset{h_2}{\triangleleft} \cdots \overset{h_n}{\triangleleft} \mathcal{X}_n.$$

In particular, we will show that for all $k = 0, \ldots, n$, the coordinates of the vertices of the polytope

$$\mathcal{V}_k = \{1\} \overset{h_1}{\triangleleft} \cdots \overset{h_k}{\triangleleft} \mathcal{X}_k$$

constructed by considering only the first k scaled extensions in the decomposition are all integer. Since $\mathcal{V} \subseteq [0, 1]^{|\Sigma_1 \bowtie \Sigma_2|}$ (Remark 2), this immediately implies that each coordinate is in $\{0, 1\}$.

- **Base case:** $k = 0$. In this case, $\mathcal{V}_0 = \{1\}$. The only vertex is $\{1\}$, which is integer. So, base case trivially holds.
- **Inductive step.** Suppose that the polytope $\mathcal{V}_k$ ($k < n$) has integer vertices. We will show that the same holds for $\mathcal{V}_{k+1}$. Clearly, $\mathcal{V}_{k+1} = \mathcal{V}_k \triangleleft^{h_{k+1}} \mathcal{X}_{k+1}$. From the properties of the structural decomposition, we know that $\mathcal{K}_{k+1}$ is either the singleton $\{1\}$, or a probability simplex $\Delta^{s_{k+1}}$ for some appropriate dimension s_{k+1}. We break the analysis accordingly.
  - If $\mathcal{X}_{k+1} = \{1\}$, the scaled extension represents filling in a linearly-dependent entry in $\boldsymbol{v} \in \mathcal{V}$ by summing already-filled-in entries. So, h_{k+1} takes a partially-filled-in vector from $\mathcal{V}_k$ and sums up some of its coordinates. Let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ be the vertices of $\mathcal{V}_k$. Using Lemma 1, the vertices of $\mathcal{V}_{k+1}$ are a subset of

    $$\{(\boldsymbol{v}_i, h(\boldsymbol{v}_i) \cdot 1) : i = 1, \ldots, n\}. \tag{8}$$

    Since by inductive hypothesis $\boldsymbol{v}_i$ have integer coordinates, and h sums up some of them, $h(\boldsymbol{v})_i$ is integer for all $i = 1, \ldots, n$. So, all of the vectors in (8) have integer coordinates, and in particular this must be true of the vertices of $\mathcal{V}_{k+1}$.
  - If $\mathcal{X}_{k+1} = \Delta^{s_{k+1}}$, the scaled extension represents the operation of partitioning an already-filled-in entry $v[\sigma, \tau]$ of $\mathcal{V}_k$ into s_i non-negative real values. The affine function h_{k+1} extracts the entry $v[\sigma, \tau]$ from each vector $\boldsymbol{v} \in \mathcal{V}_k$. Let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ be the vertices of $\mathcal{V}_k$. The vertices of $\Delta^{s_{k+1}}$ are the canonical basis vectors $\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_{s_{k+1}}\}$. From Lemma 1, the vertices of $\mathcal{V}_{k+1}$ are a subset of

    $$\{(\boldsymbol{v}_i, h(\boldsymbol{v}_i)\boldsymbol{e}_j) : i = 1, \ldots, n, j = 1, \ldots, s_{k+1}\}$$
    $$= \{(\boldsymbol{v}_i, v_i[\sigma, \tau]\boldsymbol{e}_j) : i = 1, \ldots, n, j = 1, \ldots, s_{k+1}\}. \tag{9}$$

    Since by inductive hypothesis the vertices $\boldsymbol{v}_i$ have integer coordinates, $v_i[\sigma, \tau]$ is an integer. Since the canonical basis vector only have entries in $\{0, 1\}$, all of the vectors in (9) have integer coordinates. So, in particular, this must be true of the vertices of $\mathcal{V}_{k+1}$. $\square$

C Additional Experimental Results

In this section we present additional computational results. Specifically, we present results on how well algorithms can solve for EFCE (and thus also EFCCE and NFCCE since they are supsets of EFCE) after our new scaled-extension-based structural decomposition has been computed for the polytope of correlation plans using the algorithm that we presented in the body. The speed of that algorithm for computing the decomposition is extremely fast, as shown in the body both theoretically and experimentally. Here we report the performance of two leading algorithms for finding an approximate optimal EFCE after the decomposition algorithm has completed. Specifically, we compare the performance of the regret-minimization method of Farina et al. (2019b) to that of the barrier algorithm for linear programming implemented by the Gurobi commercial linear programming solver, as described in the body of the paper. (On these problems, any linear programming solver could be used in principle, but simplex and dual simplex methods—even the ones in Gurobi—are prohibitively slow. Similarly, the subgradient descent method of Farina et al. (2019a) is known to be dominated by the regret-minimization method of Farina et al. (2019b).)

Both algorithms are used to converge to a feasible EFCE—that is, no objective function was set—in the largest Goofspiel instance ($k = 5$). Our implementation of the regret minimization method is single-threaded, while we allow Gurobi to use 30 threads. All experiments were conducted on a machine with 64 cores and 500GB of memory. Gurobi required roughly 200GB of memory, while the memory footprint of the regret-minimization algorithm was less than 2GB.

At all times, the regret-minimization algorithm produces *feasible* correlation plans, that is, points that belong to $\Xi = \mathcal{V}$. So, that algorithm's iterates' infeasibility is defined as how incentive-incompatible the computed correlation plan is, measured as the difference in value that each player would gain by optimally deviating from any recommendation at any information set in the game. In contrast, the barrier method does not guarantee that the correlation plan is primal feasible, that is, the correlation plans produced by the barrier algorithm might not be in $\Xi = \mathcal{V}$. Therefore, for Gurobi, we measure infeasibility as the maximum between (i) the (maximum) violation of the constraints that define $\mathcal{V}$, and (ii) the incentive-incompatibility of the iterate.

Figure 4 shows the results. The regret minimization algorithm works better as an anytime algorithm and leads to lower infeasibility for most of the run. The barrier method needs significant time to preprocess before even the first iterates are found. After that it converges rapidly.
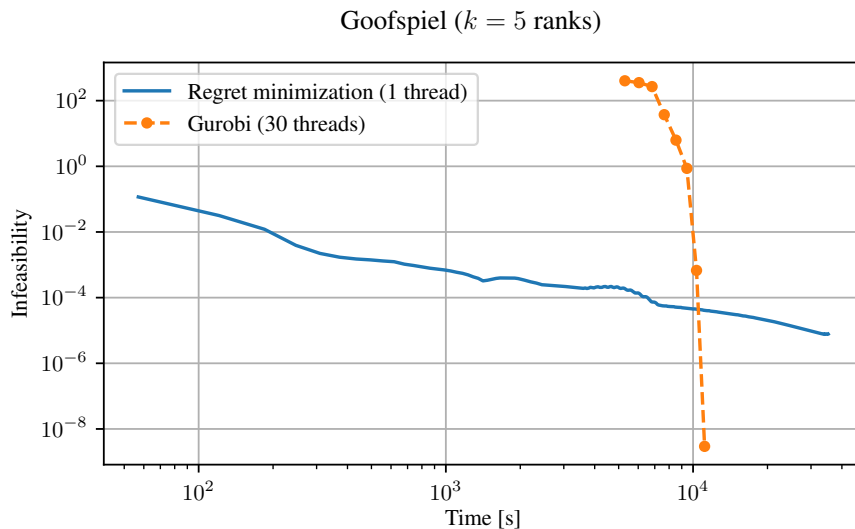

Figure 4: Performance of the regret minimization method of Farina et al. (2019b) compared to Gurobi's barrier method in the largest Goofspiel game ($k = 5$).