Connecting Optimal Ex-Ante Collusion in Teams to Extensive-Form Correlation: Faster Algorithms and Positive Complexity Results

Gabriele Farina [1] **Andrea Celli** [2] **Nicola Gatti** [2] **Tuomas Sandholm** [1 3 4 5]

Abstract

We focus on the problem of finding an optimal strategy for a team of players that faces an opponent in an imperfect-information zero-sum extensive-form game. Team members are not allowed to communicate during play but can coordinate before the game. In this setting, it is known that the best the team can do is sample a profile of potentially randomized strategies (one per player) from a joint (a.k.a. correlated) probability distribution at the beginning of the game. In this paper, we first provide new modeling results about computing such an optimal distribution by drawing a connection to a different literature on extensive-form correlation. Second, we provide an algorithm that allows one for capping the number of profiles employed in the solution. This begets an anytime algorithm by increasing the cap. We find that often a handful of well-chosen such profiles suffices to reach optimal utility for the team. This enables team members to reach coordination through a simple and understandable plan. Finally, inspired by this observation and leveraging theoretical concepts that we introduce, we develop an efficient column-generation algorithm for finding an optimal distribution for the team. We evaluate it on a suite of common benchmark games. It is three orders of magnitude faster than the prior state of the art on games that the latter can solve and it can also solve several games that were previously unsolvable.

[*]Equal contribution [1]Computer Science Department, Carnegie Mellon University, Pittsburgh PA 15213 [2]DEIB, Politecnico di Milano, Milano, Italy [3]Strategic Machine, Inc. [4]Strategy Robot, Inc. [5]Optimized Markets, Inc.. Correspondence to: Gabriele Farina <gfarina@cs.cmu.edu>, Andrea Celli <andrea.celli@polimi.it>, Nicola Gatti <nicola.gatti@polimi.it>, Tuomas Sandholm <sandholm@cs.cmu.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

1. Introduction

Much of the computational game theory literature has focused on finding strong strategies for large two-player zero-sum extensive-form games. In that setting, perfect game playing corresponds to playing strategies that belong to a Nash equilibrium, and such strategies can be found in polynomial time in the size of the game. Recent landmark results, such as superhuman agents for heads-up limit and no-limit Texas hold'em poker (Bowling et al., 2015; Brown & Sandholm, 2019; Moravčík et al., 2017) show that the problem of computing strong strategies in two-player zero-sum games is well understood both in theory and in practice. The same cannot be said for almost any type of strategic multi-player interaction, where computing strong strategies is generally hard in the worst case.

In this paper, we study *adversarial team games*, that is, games in which a team of coordinating (colluding) players faces an opponent. We will focus on a two-player team coordinating against a third player. Team members can plan jointly at will before the game, but are not allowed to communicate during the game (other than through their actions in the game). These games are a popular middle ground between two-player zero-sum games and multiplayer games (von Stengel & Koller, 1997; Celli & Gatti, 2018). They can be used to model many strategic interactions of practical relevance. For example, how should two players colluding against a third at a poker table play? Or, how would the two defenders in Bridge (who are prohibited from communicating privately during the game) play optimally against the declarer? Even though adversarial team games are conceptually zero-sum interactions between two entities—the team and the opponent—computing optimal strategies is hard in this setting. Even finding a best-response strategy for the team given a fixed strategy for the opponent is hard (Celli & Gatti, 2018).

One might think that finding the optimal strategy for the team simply amounts to finding an optimal profile of potentially mixed (a.k.a. randomized) strategies, one strategy per team members. A solution of this type that yields maximum expected sum of utilities for the team players against a rational (that is, best-responding) opponent is known as a *team-maxmin equilibrium* (TME) strategy (Basilico et al., 2017; Zhang & An, 2020a;b).

In this paper, we are interested in a more powerful model. Before the game starts, the team members are able to sample a profile from a joint (a.k.a. correlated) *distribution*. This form of *ex-ante coordination* is known to be the best a team can do and comes with two major advantages. First, it offers the team larger (or equal) expected utility than TME—sometimes with dramatic gains (Celli & Gatti, 2018). Second, it makes the problem of computing the optimal team strategy convex—and thus more amenable to the plethora of convex optimization algorithms that have been developed over the past 80 years—whereas the problem of computing a TME strategy is not convex. In our model, an optimal distribution for the team is known as a *team-maxmin equilibrium with coordination device* (TMECor) strategy (Celli & Gatti, 2018; Farina et al., 2018).

Our contributions. We introduce the notion of *semi-randomized correlation plan*, and propose a natural formulation for the problem of finding a TMECor strategy by drawing connections with the extensive-form strategy polytope defined by von Stengel & Forges (2008). Second, we propose an algorithm for computing a TMECor strategy when only a fixed number of pairs of semi-randomized correlation plans is allowed. This begets an anytime algorithm by increasing that fixed number. Surprisingly, we find that often a handful of well-chosen semi-randomized correlation plans is enough to reach optimal utility. This enables team members to reach coordination through simple and understandable strategies. Finally, by leveraging our new representation, we develop a column-generation algorithm for finding a TMECor strategy. The core of our algorithm is a new best-response (BR) oracle for computing joint team best-response strategies. We show that, in contrast with the previous state-of-the-art BR oracles for that problem, our oracle enables provably polynomial-time computation of a TMECor in some notable classes of games including, for example, Goofspiel. This result constitutes the first example of efficient computation of optimal ex-ante coordinated strategies in adversarial team games, and cannot be achieved by employing previous BR oracles. We evaluate our column-generation algorithm on a suite of common benchmark games. It is three orders of magnitude faster than the prior state of the art on games that the latter can solve. It can also solve many games that were previously unsolvable.

2. Preliminaries

Extensive-form games (EFGs) model games that are played on a game tree, and can capture both sequential and simultaneous moves, as well as private information. In this paper, we focus on three-player zero-sum games where two players—$\mathsf{T1}$ and $\mathsf{T2}$—play as a team against the opponent player, denoted by O.

Each node v in the game tree belongs to exactly one player $i \in \{\mathsf{T1}, \mathsf{T2}, \mathsf{O}\} \cup \{\mathrm{C}\}$ whose turn is to move. Player C is a special player, called the *chance player*. It models exogenous stochasticity in the environment, such as drawing a card from a deck or tossing a coin. The edges leaving v represent the actions available at that node. Any node without outgoing edges is called a *leaf* and represents an end state of the game. We denote the set of such nodes by Z. Each $z \in Z$ is associated with a tuple of payoffs specifying the payoff $u_i(z)$ of each player $i \in \{\mathsf{T1}, \mathsf{T2}, \mathsf{O}\}$ at z. The product of the probabilities of all actions of C on the path from the root of the game to leaf z is denoted by $p_{\mathrm{C}}(z)$.

Private information is represented via *information set* (infoset). In particular, the set of nodes belonging to $i \in \{\mathsf{T1}, \mathsf{T2}, \mathsf{O}\}$ is partitioned into a collection $\mathcal{I}_i$ of non-empty sets: each $I \in \mathcal{I}_i$ groups together nodes that Player i cannot distinguish among, given what they have observed. Necessarily, for any $I \in \mathcal{I}_i$ and $v, w \in I$, nodes v and w must have the same set of available actions. Consequently, we denote the set of actions available at all nodes of I by A_I. As it is customary in the related literature, we assume *perfect recall*, that is, no player forgets what he/she knew earlier in the game. Finally, given players i and j, two infosets $I_i \in \mathcal{I}_i, I_j \in \mathcal{I}_j$ are *connected*, denoted by $I_i \rightleftharpoons I_j$, if there exist $v \in I_i$ and $w \in I_j$ such that the path from the root to v passes through w or vice versa.

Sequences. The set of *sequences* of Player i, denoted by Σ_i, is defined as $\Sigma_i := \{(I, a) : I \in \mathcal{I}_i, a \in A_I\} \cup \{\varnothing\}$, where the special element $\varnothing$ is called the *empty sequence* of Player i. The *parent sequence* of a node v of Player i, denoted $\sigma(v)$, is the last sequence (information set-action pair) for Player i encountered on the path from the root of the game to that node. Since the game has perfect recall, for each $I \in \mathcal{I}_i$, nodes belonging to I share the same *parent sequence*. So, given $I \in \mathcal{I}_i$, we denote by $\sigma(I) \in \Sigma_i$ the unique parent sequence of nodes in I. Additionally, we let $\sigma(I) = \varnothing$ if Player i never acts before infoset I.

Relevant sequences. A pair of sequences $\sigma_i \in \Sigma_i, \sigma_j \in \Sigma_j$ is *relevant* if either one is the empty sequence, or if they can be written as $\sigma_i = (I_i, a_i)$ and $\sigma_j = (I_j, a_j)$ with $I_i \rightleftharpoons I_j$. We write $\sigma_i \bowtie \sigma_j$ to denote that they form a pair of relevant sequences. Given two players i and j, we let $\Sigma_i \bowtie \Sigma_j := \{(\sigma_i, \sigma_j) : \sigma_i \in \Sigma_i, \sigma_j \in \Sigma_j, \sigma_i \bowtie \sigma_j\}$. Similarly, given σ_i and $I_j \in \mathcal{I}_j$, we say that (σ_i, I_j) forms a relevant sequence-information set pair ($\sigma_i \bowtie I_j$), if $\sigma_i = \varnothing$ or if $\sigma_i = (I_i, a_i)$ and $I_i \rightleftharpoons I_j$.

Reduced-normal-form plans. 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. The set of reduced-normal-form plans of Player i is denoted 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 $I \in \mathcal{I}_i$. Similarly, given $\sigma = (I, a) \in \Sigma_i$, let $\Pi_i(\sigma) \subseteq \Pi_i(I)$ be the set of reduced-normal-form plans belonging to $\Pi_i(I)$ where Player i plays action a at I, and let $\Pi_i(\varnothing) := \Pi_i$. Finally, given a leaf $z \in Z$, we denote with $\Pi_i(z) \subseteq \Pi_i$ the set of reduced-normal-form plans where Player i plays so as to reach z.

Sequence-form strategies. A *sequence-form strategy* is a compact strategy representation for perfect-recall players in EFGs (Romanovskii, 1962; Koller et al., 1996). Given a player $i \in \{\mathsf{T1}, \mathsf{T2}, \mathsf{O}\}$ and a normal-form strategy $\mu \in \Delta(\Pi_i)$,[1] the sequence-form strategy induced by μ is the real vector $\boldsymbol{y}$, indexed over $\sigma \in \Sigma_i$, defined as $y[\sigma] := \sum_{\pi \in \Pi_i(\sigma)} \mu(\pi)$. The set of sequence-form strategies that can be induced as μ varies over $\Delta(\Pi_i)$ is denoted by $\mathcal{Y}_i$ and is known to be a convex polytope (called the *sequence-form polytope*) defined by a number of constraints equal to $|\mathcal{I}_i|$ (Koller et al., 1996).

TMECor as a Bilinear Saddle-Point Problem. A TMECor strategy is a probability distribution μ_T over the set of randomized strategy profiles $\mathcal{Y}_{\mathsf{T1}} \times \mathcal{Y}_{\mathsf{T2}}$ that guarantees maximum expected utility for the team against the best-responding opponent O. Since each player has perfect recall, any randomized strategy for a player is equivalent to a distribution over reduced-normal-form pure strategies (Kuhn, 1953). Hence, any distribution over profiles of randomized strategies of the team members can be expressed in an equivalent way as a distribution over *deterministic* strategy profiles $\Pi_{\mathsf{T1}} \times \Pi_{\mathsf{T2}}$. The benefit of this transformation is that $\Pi_{\mathsf{T1}} \times \Pi_{\mathsf{T2}}$ is a finite set, unlike $\mathcal{Y}_{\mathsf{T1}} \times \mathcal{Y}_{\mathsf{T2}}$. For this reason, TMECor is usually defined in the literature as a distribution over $\Pi_{\mathsf{T1}} \times \Pi_{\mathsf{T2}}$ without loss of generality. We will follow the same approach in our characterization.

For each leaf z, let $\hat{u}_\mathsf{T}(z) := (u_{\mathsf{T1}}(z) + u_{\mathsf{T2}}(z)) p_\mathsf{C}(z)$. The expected utility of the team can be written as the following function of the distributions of play $\mu_\mathsf{T} \in \Delta(\Pi_{\mathsf{T1}} \times \Pi_{\mathsf{T2}}), \mu_\mathsf{O} \in \Delta(\Pi_\mathsf{O})$:

$$u_\mathsf{T}(\mu_\mathsf{T}, \mu_\mathsf{O}) := \sum_{z \in Z} \hat{u}_\mathsf{T}(z) \left(\sum_{\substack{\pi_{\mathsf{T1}} \in \Pi_{\mathsf{T1}}(z) \\ \pi_{\mathsf{T2}} \in \Pi_{\mathsf{T2}}(z)}} \mu_\mathsf{T}(\pi_{\mathsf{T1}}, \pi_{\mathsf{T2}}) \right) \left(\sum_{\pi \in \Pi_\mathsf{O}(z)} \mu_\mathsf{O}(\pi) \right).$$

By definition, a *team-maxmin equilibrium with coordination device* (TMECor) is a Nash equilibrium of the game where the team plays according to the coordinated strategy $\mu_\mathsf{T} \in \Delta(\Pi_{\mathsf{T1}} \times \Pi_{\mathsf{T2}})$. In the zero-sum setting, this amounts to finding a solution of the optimization problem

$$\underset{\mu_\mathsf{T} \in \Delta(\Pi_{\mathsf{T1}} \times \Pi_{\mathsf{T2}})}{\arg\max} \; \min_{\mu_\mathsf{O} \in \Delta(\Pi_\mathsf{O})} u_\mathsf{T}(\mu_\mathsf{T}, \mu_\mathsf{O}). \tag{1}$$

[1] $\Delta(X)$ denotes the probability simplex over the finite set X.

The opponent's strategy μ_O can be compactly represented through its equivalent sequence-form representation. This is not the case for μ_T, which cannot be represented concisely through the sequence form as shown by Farina et al. (2018).

3. A New Formulation of TMECor Based on Extensive-Form Correlation Plans

We propose using a different representation of the correlated distribution of play μ_T, inspired by the growing body of literature on extensive-form correlated equilibria. Like the *realization form* by Farina et al. (2018), in our approach we represent μ_T as a vector with only a polynomial number of components. However, unlike the realization form, the number of components scales as the product of the number of sequences of the two players, which can be significantly larger than the number of leaves. This downside is amply outweighed by the following benefits. First, we show that in practice our proposed representation of μ_T enables us to compute best responses for the team significantly faster than the prior representations. Second, in certain classes of games, we even show that our proposed representation enables the computation of a TMECor in polynomial time. This is the case, for example, in Goofspiel, a popular benchmark game in computational game theory (Ross, 1971).

3.1. Extensive-Form Correlation Plans

Our representation is based on the concept of *extensive-form correlation plans*, introduced by von Stengel & Forges (2008) in their seminal paper on extensive-form correlation. In particular, we map the correlated distribution of play μ_T of the team to the vector $\boldsymbol{\xi}_\mathsf{T}$ indexed over pairs of sequences $(\sigma_{\mathsf{T1}}, \sigma_{\mathsf{T2}}) \in \Sigma_{\mathsf{T1}} \bowtie \Sigma_{\mathsf{T2}}$, where each entry is defined as

$$\xi_\mathsf{T}[(\sigma_{\mathsf{T1}}, \sigma_{\mathsf{T2}})] := \sum_{\substack{\pi_{\mathsf{T1}} \in \Pi_{\mathsf{T1}}(\sigma_{\mathsf{T1}}) \\ \pi_{\mathsf{T2}} \in \Pi_{\mathsf{T2}}(\sigma_{\mathsf{T2}})}} \mu_\mathsf{T}[(\pi_{\mathsf{T1}}, \pi_{\mathsf{T2}})]. \tag{2}$$

Here $\boldsymbol{\xi}_\mathsf{T}$ is not indexed over *all* pairs of sequences $(\sigma_{\mathsf{T1}}, \sigma_{\mathsf{T2}})$—only *relevant* sequence pairs. While there are games in which this distinction is meaningless (that is, games in which all sequences pairs for the team members are relevant), in practice the number of *all* sequence pairs is usually significantly bigger than the number of relevant sequence pairs, as shown in Table 1(b).

The set of extensive-form correlation plans $\boldsymbol{\xi}_\mathsf{T}$ that can be induced as μ_T varies over the set of all correlated distributions of play for the team members is a convex polytope. We denote it as Ξ_T and call it the *polytope of correlation plans*. We will recall existing results and provide new ones about the structure of Ξ_T in Section 4.

Game instance	Num. sequences $\lvert\Sigma_1\rvert$	$\lvert\Sigma_2\rvert$	$\lvert\Sigma_3\rvert$	Num. leaves $\lvert Z\rvert$	$\frac{\lvert\Sigma_{T1}\bowtie\Sigma_{T2}\rvert}{\lvert Z\rvert}$	$\frac{\lvert\Sigma_{T1}\times\Sigma_{T2}\rvert}{\lvert\Sigma_{T1}\bowtie\Sigma_{T2}\rvert}$	Triangle-free? $O=1$	$O=2$	$O=3$
[A] Kuhn poker (3 ranks)	25	25	25	78	3.40	2.36	✗	✗	✗
[B] Kuhn poker (4 ranks)	33	33	33	312	1.59	2.19	✗	✗	✗
[C] Kuhn poker (12 ranks)	97	97	97	17 160	0.29	1.90	✗	✗	✗
[D] Goofspiel (3 ranks, limited info)	934	934	934	1296	9.54	70.59	✓	✓	✓
[E] Goofspiel (3 ranks)	1630	1630	1630	1296	15.54	131.96	✓	✓	✓
[F] Liar's dice (3 faces)	1021	1021	1021	13 797	5.27	14.43	✗	✗	✗
[G] Liar's dice (4 faces)	10 921	10 921	10 921	262 080	6.25	72.79	✗	✗	✗
[H] Leduc poker (3 ranks, 1 raise)	457	457	457	6477	1.82	17.70	✗	✗	✗
[I] Leduc poker (4 ranks, 1 raise)	801	801	801	20856	1.08	28.36	✗	✗	✗
[J] Leduc poker (2 ranks, 2 raises)	1443	1443	1443	8762	3.14	75.59	✗	✗	✗
(a) — Game instances and sizes					**(b)**		**(c)**		

Table 1: **(a)** Size of the game instances used in our experiments, in terms of number of sequences $|\Sigma_i|$ for each player i, and number of leaves $|Z|$. **(b)** Ratio between the number of leaves $|Z|$, number of sequence pairs for the team members $|\Sigma_{T1}\times\Sigma_{T2}|$, and number of *relevant* sequence pairs for the team members $|\Sigma_{T1}\bowtie\Sigma_{T2}|$ in various benchmark games. For all games reported in the subtable, we chose the first two players to act as the team members. **(c)** The subtable reports whether the interaction of the team members is triangle-free (Farina & Sandholm, 2020), given the opponent player O.

3.2. Computing a TMECor using Correlation Plans

Extensive-form correlation plans encode a *superset* of the information encoded by realization plans. Indeed, for all z, $\xi_T[\sigma_{T1}(z),\sigma_{T2}(z)] = \rho_T[z]$. Using the previous identity, we can rewrite the problem of computing a TMECor of a constant-sum game (1) as

$$\arg\max_{\boldsymbol{\xi}_T\in\Xi_T}\ \min_{\boldsymbol{y}_O\in\mathcal{Y}_O} \sum_{z\in Z} \hat{u}_T(z)\xi_T[\sigma_{T1}(z),\sigma_{T2}(z)]y[\sigma_O(z)].$$

By dualizing the inner linear minimization problem over $\boldsymbol{y}_O$, we get the following proposition that shows that a TMECor can be found as the solution to a linear program (LP) with a polynomial number of variables. (All the proofs of this paper can be found in the appendix.)

Proposition 1. *An extensive-form correlation plan $\boldsymbol{\xi}_T$ is a TMECor if and only if it is a solution to the LP*

$$\begin{cases} \arg\max_{\boldsymbol{\xi}_T}\ v_\varnothing, \quad \text{subject to:} \\ ①\ v_I - \sum\limits_{\substack{I'\in\mathcal{I}_O \\ \sigma_O(I')=(I,a)}} v_{I'} \le \sum\limits_{\substack{z\in Z \\ \sigma_O(z)=(I,a)}} \hat{u}_T(z)\xi_T[\sigma_{T1}(z),\sigma_{T2}(z)] \qquad \forall\,(I,a)\in\Sigma_O\setminus\{\varnothing\} \\ ②\ v_\varnothing - \sum\limits_{\substack{I'\in\mathcal{I}_O \\ \sigma_O(I')=\varnothing}} v_{I'} \le \sum\limits_{\substack{z\in Z \\ \sigma_O(z)=\varnothing}} \hat{u}_T(z)\xi_T[\sigma_{T1}(z),\sigma_{T2}(z)] \\ ③\ v_\varnothing \text{ free}, v_I \text{ free} \quad \forall\, I\in\mathcal{I}_O \\ ④\ \boldsymbol{\xi}_T\in\Xi_T. \end{cases}$$

As a direct consequence of Proposition 1, a TMECor can be found in polynomial time whenever Ξ_T can be represented as the intersection of a set of polynomially many linear constraints. In Section 4, we recall when that is the case.

4. Semi-Randomized Correlation Plans and the Structure of Ξ_T

Even though Ξ_T is a convex polytope, the set of (potentially exponentially many) linear constraints that define it is not known in general. So, alternative characterizations of the set Ξ_T are needed before the LP in Proposition 1 can be solved. In this section, we recall two known results about the structure of Ξ_T, and propose a new one (Proposition 3). We will use our result to arrive at two different approaches to tackle the LP of Proposition 1 in Sections 5 and 6, respectively.

4.1. Containment in the von Stengel-Forges Polytope

The first result about the structure of Ξ_T has to do with a particular polytope that was introduced by von Stengel & Forges (2008).

Definition 1. *The* von Stengel-Forges polytope *of the team, denoted $\mathcal{V}_T$, is the polytope of all vectors $\boldsymbol{\xi}\in\mathbb{R}_{\ge 0}^{|\Sigma_{T1}\bowtie\Sigma_{T2}|}$ indexed over relevant sequence pairs that satisfy the following polynomially-sized set of linear constraints.*

$$① \quad \xi[\varnothing,\varnothing]=1$$

$$② \quad \sum_{a_{T1}\in A_{I_{T1}}} \xi[(I_{T1},a_{T1}),\sigma_{T2}] = \xi[\sigma(I_{T1}),\sigma_{T2}] \quad \forall I_{T1}\bowtie\sigma_{T2}$$

$$③ \quad \sum_{a_{T2}\in A_{I_{T2}}} \xi[\sigma_{T1},(I_{T2},a_{T2})] = \xi[\sigma_{T1},\sigma(I_{T2})] \quad \forall \sigma_{T1}\bowtie I_{T2}.$$

These can be interpreted as "probability mass conservation" constraints. They are interlaced sequence-form constraints.

The following result by von Stengel & Forges (2008) is immediate from the definition of ξ_T in (2).

Proposition 2 (von Stengel & Forges (2008))**.** *The set of extensive-form correlation plans is a subset of the von Stengel-Forges polytope. Formally, $\Xi_T\subseteq\mathcal{V}_T$.*

4.2. Triangle-Freeness and Polynomial-Time Computation of TMECor

Proposition 2 shows that Ξ_{T} is a subset of the von Stengel-Forges polytope. There are games where the reverse inclusion does not hold. Farina & Sandholm (2020) gave a sufficient condition—called *triangle-freeness*—for the reverse inclusion to hold. We state the condition for our setting.

Definition 2 (Farina & Sandholm (2020))**.** *The interaction of the team members* T1 *and* T2 *is* triangle-free *if, for any choice of distinct information sets* $I_1, I_2 \in \mathcal{I}_{\mathsf{T1}}$ *with* $\sigma_{\mathsf{T1}}(I_1) = \sigma_{\mathsf{T1}}(I_2)$ *and any choice of distinct information sets* $J_1, J_2 \in \mathcal{I}_{\mathsf{T2}}$ *with* $\sigma_{\mathsf{T2}}(J_1) = \sigma_{\mathsf{T2}}(J_2)$*, it is never the case that* $(I_1 \rightleftharpoons J_1) \wedge (I_2 \rightleftharpoons J_2) \wedge (I_1 \rightleftharpoons J_2)$.

The above condition can be easily checked in polynomial-time by iterating over all possible quadruplets of information sets I_1, I_2, J_1, J_2 and checking whether the condition (which itself can be checked by performing a standard traversal of the input game tree) holds. Farina & Sandholm (2020) show that when the information structure of correlating players (in our case, the team members) is triangle-free, then $\Xi_{\mathsf{T}} = \mathcal{V}_{\mathsf{T}}$. So, when the interaction of the team is triangle-free, a TMECor can be found in polynomial time by substituting constraint ④ in the LP in Proposition 1 with the von Stengel-Forges constraints of Definition 1. As far as we are aware, this positive complexity result has not been noted before in the literature. We show in Table 1(c) that Goofspiel is triangle free, since all chance outcomes are public (and that none of the other common benchmark games that we consider are).

4.3. Semi-Randomized Correlation Plans

We now give a third result about the structure of Ξ_{T}, which will enable us to replace Constraint ④ of Proposition 1 with something more practical. First, we introduce *semi-randomized correlation plans*, which are elements of a subset of the von Stengel-Forges polytope of the team, as we formalize shortly. A semi-randomized correlation plan represents a strategy profile in which one of the players plays a deterministic strategy, while the other player in the team independently plays a randomized strategy. Formally, we define the set of semi-randomized correlation plans for T1 and T2 as

$$\Xi^*_{\mathsf{T1}} = \{\boldsymbol{\xi} \in \mathcal{V}_{\mathsf{T}} : \xi[\varnothing, \sigma_{\mathsf{T2}}] \in \{0,1\} \quad \forall\, \sigma_{\mathsf{T2}} \in \Sigma_{\mathsf{T2}}\},$$
$$\Xi^*_{\mathsf{T2}} = \{\boldsymbol{\xi} \in \mathcal{V}_{\mathsf{T}} : \xi[\sigma_{\mathsf{T1}}, \varnothing] \in \{0,1\} \quad \forall\, \sigma_{\mathsf{T1}} \in \Sigma_{\mathsf{T1}}\},$$

respectively. Crucially, a point $\boldsymbol{\xi} \in \Xi^*_i$ for $i \in \{\mathsf{T1}, \mathsf{T2}\}$ can be expressed using real and binary variables, in addition to the linear constraints the define $\mathcal{V}$ (Definition 1).

With that, we can show the following structural result for the polytope of extensive-form correlation plans Ξ_{T}.

Proposition 3. *In every game,* Ξ_{T} *is the convex hull of the set* Ξ^*_{T1}*, or equivalently of the set* Ξ^*_{T2}*. Formally,* $\Xi_{\mathsf{T}} = \operatorname{co}\Xi^*_{\mathsf{T1}} = \operatorname{co}\Xi^*_{\mathsf{T2}} = \operatorname{co}(\Xi^*_{\mathsf{T1}} \cup \Xi^*_{\mathsf{T2}})$.

Our notion of semi-randomized correlation plans is reminiscent of the *auxiliary game* construction of Farina et al. (2018), in which only one of the team members (the *pivot player*) is required to play a deterministic strategy. Our setting is very different, however, since we have a different representation of team strategies with many more variables and stronger combinatorial structure.

5. Computing TMECor with a Small Support of Semi-Randomized Plans of Fixed Size

From Proposition 3, it is known that Ξ_{T} is the convex hull of Ξ^*_{T1} and Ξ^*_{T2}. Furthermore, the polytopes Ξ^*_{T1} and Ξ^*_{T2} can be described via a number of linear constraints that is quadratic in the game size and a number of integer variables that is linear in the game size. So, we can replace Constraint ④ in Proposition 1 with the constraint that $\boldsymbol{\xi}_{\mathsf{T}}$ be a convex combination of elements from Ξ^*_{T1} and Ξ^*_{T2}. We introduce variables $\boldsymbol{\xi}^{(1)}_{\mathsf{T}}, \dots, \boldsymbol{\xi}^{(n)}_{\mathsf{T}} \in \Xi^*_{\mathsf{T1}} \cup \Xi^*_{\mathsf{T2}}$ and the corresponding convex combination coefficients $\lambda^{(1)}, \dots, \lambda^{(n)}$, and replace Constraint ④ with the linear constraint $\boldsymbol{\xi}_{\mathsf{T}} = \sum_{i=1}^{n} \lambda^{(i)} \boldsymbol{\xi}^{(i)}_{\mathsf{T}}$. Here, n is a parameter with which we can cap the number of semi-randomized correlation plans that can be included in the strategy. This gives the following mixed integer LP.

$$\begin{cases} \displaystyle\mathop{\arg\max}_{\boldsymbol{\xi}^{(1)}_{\mathsf{T}},\dots,\boldsymbol{\xi}^{(n)}_{\mathsf{T}},\lambda^{(1)},\dots,\lambda^{(n)}} \quad v_{\varnothing}, \quad \text{subject to:} \\ \text{constraints ① ② ③ as in Proposition 1} \\ \text{④ } \boldsymbol{\xi}_{\mathsf{T}} = \sum_{i=1}^{n} \lambda^{(i)} \boldsymbol{\xi}^{(i)}_{\mathsf{T}} \\ \text{⑤ } \boldsymbol{\xi}^{(1)}_{\mathsf{T}} \in \Xi^*_{\mathsf{T1}}, \boldsymbol{\xi}^{(2)}_{\mathsf{T}} \in \Xi^*_{\mathsf{T2}}, \boldsymbol{\xi}^{(3)}_{\mathsf{T}} \in \Xi^*_{\mathsf{T1}}, \boldsymbol{\xi}^{(4)}_{\mathsf{T}} \in \Xi^*_{\mathsf{T2}}, \dots^{\ddagger} \\ \text{⑥ } \sum_{i=1}^{n} \lambda^{(i)} = 1,\ \lambda^{(i)} \geq 0 \quad \forall i \in \{1, \dots, n\}. \end{cases}$$

The larger n is, the higher the solution value obtained, but the slower the program. We can make this into an anytime algorithm by solving the integer program for increasing values of n. By Caratheodory's theorem, this program already yields an optimal solution to the LP in Proposition 1 when $n \geq |\Sigma_1 \bowtie \Sigma_2| + 1$. As we show in detail in Section 7, in practice we found that near-optimal coordination can be achieved through strategies with a significantly smaller value of n. Hence, oftentimes the team does not need a large number of complex profiles of randomized strategies to play optimally: a handful (often one or two) of carefully selected simple strategies often result in optimal coordination. That

[‡] In Constraint ⑤ we alternate the set of semi-randomized correlation plans (i.e., we alternate which player's turn it is to play a deterministic strategy). Empirically, this increases the diversity of the strategies of Ξ_{T} that can be represented with small values of n and leads to higher utilities for the team.

Game		Opponent player O = 1				Opponent player O = 2				Opponent player O = 3			
		$n=1$	$n=2$	$n=3$	$n=\infty$	$n=1$	$n=2$	$n=3$	$n=\infty$	$n=1$	$n=2$	$n=3$	$n=\infty$
Kuhn poker	**[A]**	0	★	★	0	0	★	★	0	0	★	★	0
	[B]	0.0208	0.0379	★	0.0379	0.0018	0.0246	0.0265	0.0265	−0.0417	★	★	−0.0417
	[C]	0.0470	0.0655	0.0663	0.0664	0.0128	0.0367	0.0376	0.0380	−0.0227	−0.0153	−0.0141	−0.0140
Goofspiel	**[D]**	0.2389	0.2524	★	0.2524	0.2389	0.2524	★	0.2524	0.2389	0.2524	★	0.2524
	[E]	0.2389	0.2534	★	0.2534	0.2389	0.2534	★	0.2534	0.2389	0.2534	★	0.2534
Liar's dice	**[F]**	0	★	★	0	0.2099	0.2554	0.2562	0.2562	0.2716	0.2840	★	0.2840
	[G]	0.0625	★	★	0.0625	0.2500	0.2656	0.2656	—	0.2656	—	—	—
Leduc poker	**[H]**	0.0326	0.1934	0.1987	0.1987	0.1333	0.1899	—	0.2530	0.1461	0.1672	0.1910	0.2148
	[I]	—	—	—	0.1859	0.0841	—	—	0.1826	−0.0532	—	—	0.1073
	[J]	0.2609	0.3767	—	0.5493	0.3125	0.5660	0.6274	0.6284	0.2609	0.3682	0.4703	0.5155

Table 2: Expected utility of the team for varying support sizes (n). All values for $n \in \{1, 2, 3\}$ were computed using the MIP of Section 5, while the values corresponding to $n = \infty$ were computed using our column generation approach (Section 6). '★': A provably optimal utility has already been obtained with a lower value of the support size n. '—': We were unable to compute the exact value, because the corresponding algorithm hits the time limit.

empirical observation complements the theoretical statement by Celli & Gatti (2018, Proposition 3), who proved that an optimal TMECor with support of size *at most* Σ_{O} always exist. This sections shows that the theoretical bound of Celli & Gatti (2018) is way too pessimistic in practice: for example, in the Goofspiel game [E], the theoretical bound would predict that a support of size at most $n = 1630$ is necessary to guarantee optimality, but in Table 2 we find that $n = 2$ is already enough.

6. A Fast Column Generation Approach

In this section, we present a scalable approach to solving the LP in Proposition 1—using column generation (Ford & Fulkerson, 1958). First, we proceed with a *seeding* phase. We pick a set S containing one or more points $\boldsymbol{\xi}_{\mathsf{T}}^{(1)}, \boldsymbol{\xi}_{\mathsf{T}}^{(2)}, \dots, \boldsymbol{\xi}_{\mathsf{T}}^{(m)}$ that are known to belong to Ξ_{T}. Then, the main loop starts. First, for $i \in \{1, \dots, |S|\}$, let

$$\beta^{(i)}(\sigma_{\mathsf{O}}) := \sum_{\substack{z \in Z \\ \sigma_{\mathsf{O}}(z) = \sigma_{\mathsf{O}}}} \hat{u}_{\mathsf{T}}(z) \xi_{\mathsf{T}}^{(i)}[\sigma_{\mathsf{T1}}(z), \sigma_{\mathsf{T2}}(z)] \quad \forall\, \sigma_{\mathsf{O}} \in \Sigma_{\mathsf{O}}.$$

Then we solve the LP of Proposition 1 where Constraint ④ has been substituted with $\boldsymbol{\xi}_{\mathsf{T}} \in \operatorname{co} S$:

$$(*) : \begin{cases} \displaystyle\operatorname*{arg\,max}_{\lambda^{(1)}, \dots, \lambda^{(|S|)}} \quad v_{\varnothing}, \quad \text{subject to:} \\ \text{①}\ \displaystyle v_I - \sum_{\substack{I' \in \mathcal{I}_{\mathsf{O}} \\ \sigma_{\mathsf{O}}(I') = \sigma_{\mathsf{O}}}} v_{I'} - \sum_{i=1}^{|S|} \beta^{(i)}(\sigma_{\mathsf{O}})\, \lambda^{(i)} \le 0 \\ \hfill \forall\, \sigma_{\mathsf{O}} \in \Sigma_{\mathsf{O}} \setminus \{\varnothing\} \\ \text{②}\ \displaystyle v_{\varnothing} - \sum_{\substack{I' \in \mathcal{I}_{\mathsf{O}} \\ \sigma_{\mathsf{O}}(I') = \varnothing}} v_{I'} - \sum_{i=1}^{|S|} \beta^{(i)}(\varnothing)\, \lambda^{(i)} \le 0 \\ \text{③}\ \sum_{i=1}^{|S|} \lambda^{(i)} = 1 \\ \text{④}\ \lambda^{(i)} \ge 0 \quad \forall\, i \in \{1, \dots, |S|\} \\ \text{⑤}\ v_{\varnothing} \text{ free}, v_I \text{ free} \quad \forall\, I \in \mathcal{I}_{\mathsf{O}}. \end{cases}$$

This is called the *master LP*.[2]

Given the solution to the master LP, a *pricing problem* is created. The goal of the pricing problem is to generate a new element $\boldsymbol{\xi}_{\mathsf{T}}^{|S|+1}$ to be added to S so as to increase the team utility in the next iteration, that is, the next solve of the master LP that then has an additional variable. This main loop of solving the larger and larger master LP keeps repeating until termination (discussed later).

6.1. The Pricing Problem

The pricing problem consist of finding a correlation plan $\hat{\boldsymbol{\xi}}_{\mathsf{T}} \in \Xi_{\mathsf{T}}$ which, if included in the convex combination computed by $(*)$, would lead to the maximum gradient of the objective (that is, the maximum *reduced cost*). By exploiting the theory of linear programming duality, such a correlation plan can be computed starting from the solution of the dual of $(*)$. In particular, let $\boldsymbol{\gamma}$ be the $|\Sigma_{\mathsf{O}}|$-dimensional vector of dual variables corresponding to Constraints ① and ② of $(*)$, and $\gamma' \in \mathbb{R}$ be the dual variable corresponding to Constraint ③. Then, the reduced cost of any candidate $\hat{\boldsymbol{\xi}}_{\mathsf{T}}$ is

$$c(\hat{\boldsymbol{\xi}}_{\mathsf{T}}) := -\gamma' + \sum_{z \in Z} \hat{u}_{\mathsf{T}}(z) \hat{\xi}_{\mathsf{T}}[\sigma_{\mathsf{T1}}(z), \sigma_{\mathsf{T2}}(z)] \gamma[\sigma_{\mathsf{O}}(z)].$$

Now comes our crucial observation. Since $c(\hat{\boldsymbol{\xi}}_{\mathsf{T}})$ is a linear function, and since from Proposition 3 we know that $\Xi_{\mathsf{T}} = \operatorname{co} \Xi_{\mathsf{T1}}^{*}$, by convexity

$$\max_{\hat{\boldsymbol{\xi}}_{\mathsf{T}} \in \Xi_{\mathsf{T}}} c(\hat{\boldsymbol{\xi}}_{\mathsf{T}}) = \max_{\hat{\boldsymbol{\xi}}_{\mathsf{T}} \in \Xi_{\mathsf{T1}}^{*}} c(\hat{\boldsymbol{\xi}}_{\mathsf{T}}).$$

We want to solve the LP on the left hand side, but—as discussed in Section 4—the constraints defining Ξ_{T} are not

[2] In $(*)$ the convex combination is among *given* correlation plans, while in the MIP of Section 5, the elements to combine are themselves variables.

known. The above equality enables us to solve the problem because the right hand side is a well-defined mixed integer LP (MIP). We can use a commercial solver such as Gurobi to solve it. When the objective value of the pricing problem is non-positive, there is no variable that can be added to the master LP which would increase its value. Thus, the optimal solution to the master LP is guaranteed to be optimal for the LP in Proposition 1 and the main loop terminates.

6.2. Implementation Details

We further speed up the solution of the pricing problem in our implementation by the following techniques.

Seeding phase. To avoid having to go through many iterations of the main loop, each of which requires solving the pricing problem, we want to seed the master LP up front with a set of good candidate variables. While any seeding maintains optimality of the overall algorithm, seeding it with variables that are likely to be part of the optimal solution increases speed the most. We initialize the set of correlation plans S by running m iterations of a self-play no-external-regret algorithm. Specifically, we let each player run CFR+ (Tammelin et al., 2015; Bowling et al., 2015) and, at each iteration of that algorithm, we sample a pair of pure normal-form plans for the two team members according to the current strategies of the two players. At each iteration of that no-regret method, we set the utility of each team member to $u_{\mathsf{T}1} + u_{\mathsf{T}2}$. Finally, for each pair $(\pi_{\mathsf{T}1}, \pi_{\mathsf{T}2}) \in \Pi_{\mathsf{T}1} \times \Pi_{\mathsf{T}2}$ of normal-form plans generated by that no-regret algorithm, we compute and add to S the correlation plan corresponding to the distribution μ that assigns probability 1 to $(\pi_{\mathsf{T}1}, \pi_{\mathsf{T}2})$ using Eq. (2). While self-play no-regret methods guarantee convergence to Nash equilibrium in two-player zero-sum game, no guarantee is available in our setting. However, we empirically find that this seeding strategy leads to a strong initial set of correlation plans.

Linear relaxation. Before solving the MIP formulation of the pricing problem, we first try to solve its linear relaxation $\arg\max_{\hat{\boldsymbol{\xi}}_{\mathsf{T}} \in \mathcal{V}_{\mathsf{T}}} c(\hat{\boldsymbol{\xi}}_{\mathsf{T}})$. We found that in many cases it outputs semi-randomized correlation plans, thus avoiding the overhead of having to solve a MIP.

Solution pools. Modern commercial MIP solvers such as Gurobi keep track of additional suboptimal feasible solutions (in addition to the optimal one) that were found during the process of solving a MIP. Since accessing those additional solutions is essentially free computationally, we add to S all the solutions (even suboptimal ones) that were produced in the process of solving the MIP. This can be viewed as a form of dynamic seeding and does not affect the optimality of the overall algorithm.

Termination. Because fast integer and LP solvers work with real-valued variables, near the end of the column-generation loop the new variables that are generated in the pricing problem have reduced costs that are very close to zero. It is not clear whether they are actually positive or zero. Therefore, we set the numeric tolerance so that we stop the column-generation loop if the value of the pricing problem solution is less than 10^{-6}.

Dual values. To obtain the dual values used in the pricing problem, we do not need to formulate and solve a dual LP as modern LP solvers already keep track of dual values.

7. Experimental Evaluation

We computationally evaluate the algorithms proposed in Section 5 and Section 6. We test on the common parametric games shown in Table 1. Appendix B provides additional detail about the games. We ran the experiments on a machine with a 16-core 2.80GHz CPU, and allow a maximum of four threads and 32GB of RAM to each experiment. We used Gurobi 9.1.1 to solve LPs and MIPs.

7.1. Small-Supported TMECor in Practice

Table 2 describes the maximum expected utility that the team can obtain by limiting the support of its distribution to $n \in \{1, 2, 3\}$ semi-randomized correlation plans. Columns denoted by $n = \infty$ show the optimal expected utility of the team at the TMECor (without any limit on the support size). We ran experiments with the opponent as the first ($\bigcirc = 1$), second ($\bigcirc = 2$), and third player ($\bigcirc = 3$) of each game. In all the games, distributions with as few as two or three semi-randomized coordination plans gave the team near-optimal expected utility. Moreover, in several games, one or two carefully selected semi-randomized coordination plans are enough to reach an optimal solution.

7.2. Column-Generation in Practice

We evaluate our column-generation algorithm against the two prior state-of-the art algorithms for computing a TMECor: the column-generation technique by Celli & Gatti (2018) (henceforth CG-18), and the fictitious-team-play algorithm by Farina et al. (2018) (denoted FTP). Like our algorithm, CG-18 uses a column generation approach that lets $\bigcirc$ play sequence-form strategies, while the team's strategy is directly represented as a distribution over joint normal-form plans $\Pi_{\mathsf{T}1} \times \Pi_{\mathsf{T}2}$. FTP is based on the bilinear saddle-point formulation of the problem and is essentially a variation of *fictitious play* (Brown, 1951). FTP operates on the bilinear formulation of TMECor (1): the team and the opponent are treated as two entities that converge to equilibrium in self-play. FTP only guarantees convergence in the limit to a TMECor, while our algorithm certifies optimality. So, the run-time comparison between our algorithm to FTP must be done with care, as the latter never stops, whereas our

Game	Ours		Fictitious Team Play (FTP)			CG-18	Pricers		Team utility after seeding			TMECor value
	Seeded	Not seed.	$\epsilon = 50\%$	$\epsilon = 10\%$	$\epsilon = 1\%$		Relax.	MIP	$m = 0$	100	1000	
[A]	1ms	2ms	2s†	10s†	1m 08s†	175ms	1	0	–0.556	0	0	0
[B]	1ms	14ms	3m 52s	37m 51s	> 6h	26.81s	2	0	–0.406	–0.042	–0.042	–0.042
[C]	4.96s	13.40s	4h 42m	> 6h	> 6h	> 6h	4	22	–0.343	–0.030	–0.021	–0.014
[D]	325ms	517ms	50s	9m 21s	> 6h	3m 09s	18	0	–1.000	0.247	0.252	0.252
[E]	1.18s	1.48s	4m 51s	2h 02m	> 6h	29m 38s	45	0	–2.933	0.239	0.248	0.253
[F]	1m 12s	4m 03s	> 6h	> 6h	> 6h	> 6h	40	8	0.000	0.276	0.284	0.284
[G]	> 6h	> 6h	> 6h	> 6h	> 6h	> 6h	—	—	–0.688	0.277	oom	—
[H]	2m 23s	3m 23s	> 6h	> 6h	> 6h	> 6h	20	171	–1.783	0.065	0.151	0.215
[I]	1h 07m	1h 19m	> 6h	> 6h	> 6h	> 6h	7	610	–1.216	–0.149	0.019	0.107
[J]	3m 29s	1m 50s	> 6h	> 6h	> 6h	> 6h	1346	18	–6.000	–0.222	0.387	0.516

(a) — Comparison of run times **(b)** **(c)**

Table 3: **(a)** Runtime comparison between our column generation algorithm, FTP, and CG-18. The seeded version of our algorithm runs $m = 1000$ iterations of CFR+ (Section 6.2), while the non seeded version runs $m = 0$. '†': since the TMECor value for the game is exactly zero, we measure how long it took the algorithm to find a distribution with expected value at least $-\epsilon/10$ for the team. **(b)** Number of times the pricing problem for our column-generation algorithm was solved to optimality by the linear relaxation ('Relax') and by the MIP solver ('MIP') when using our column-generation algorithm. **(c)** Quality of the initial strategy of the team obtained for varying sizes of S compared to the expected utility of the team at the TMECor. 'oom': out of memory.

algorithm and CG-18 terminate after a finite number of iterations with an *exact* optimal strategy. We report the run time of FTP reaching solution quality that is $\epsilon = 50\%$, 10%, and 1% off the optimal value (determined by the other two algorithms). Finally, the concurrent work by Zhang et al. (2020) proposes an alternative approach for computing a TMECor. According to the run times reported in their paper, our algorithm is significantly faster. In particular, our algorithm takes about 2m 00s to solve game [H], while their algorithm takes about 1h 21m, even though they conducted their experiments on a more powerful machine.[3]

We set a time limit of 6 hours, a memory limit of 32GB, and a cap of four threads for each algorithm. Table 3 shows the results with the opponent playing as the third player. According to Table 2, this is almost always the hardest setting. The results for the other two settings are in Appendix C.

Our column-generation algorithm dramatically outperforms FTP and CG-18. For example, in Liar's dice instance [F], our algorithm finds an optimal TMECor in a few seconds while the prior algorithms exceed six hours. The last column of Table 3(c) shows the final team utility. Even when the opponent is playing as the third player, the team is able to reach positive expected utility. Finally, we identify Liar's dice instance [G] as the current boundary of problem that just cannot be handled with current TMECor technology: it does not complete within six hours.

[3]The machine which they used has a 3.6GHz CPU, 32GB of memory, and they dedicated 12 threads to the algorithm. Moreover, we observe that the number of terminal nodes reported for 3L3 in their paper is inconsistent with the description of the game, which corresponds to our game [H]; in particular, their paper reports that the game has a larger number of terminal nodes than it actually has. This was confirmed by the authors in private communications.

Using the linear relaxation of the pricing problem ("implementation details" in Section 6.2) often obviated the need to run the slower MIP pricing (see Table 3(b)). In all Goofspiel instances (games [D] and [E]) and in small Kuhn poker instances, the MIP pricing is never invoked.

Regret-based seeding further improves the performance of the algorithm. In the Liar's dice instance [F], it reduced run time by roughly a factor of ten. The objective value of the master solution immediately after seeding (that is, before the first column generation step) increases significantly with the number of iterations of the no-regret algorithm that is used for seeding.

8. Conclusions

We studied finding an optimal strategy for a team with two members facing an opponent in an imperfect-information, zero-sum, extensive-form game. We focused on the setting where the team members cannot communicate during play but can coordinate before the game. We provided modeling results by drawing a connection to prior results on extensive-form correlation. Then, we developed an algorithm that computes an optimal joint distribution by just using profiles where only one of the team members gets to randomize in each profile. We can cap the number of such profiles we allow in the solution. This begets an anytime algorithm by increasing the cap. Moreover, we showed that often a handful of well-chosen such profiles suffice to reach optimal utility for the team. Inspired by this observation and leveraging theoretical concepts that we introduced, we developed an efficient column-generation algorithm for finding an optimal strategy for the team. We tested our algorithm on a suite of standard games, showing that it is three order of magnitudes faster than the prior state of the art and also solves many games that were previously unsolvable.

Acknowledgments

This material is based on work supported by the National Science Foundation under grants IIS-1718457, IIS-1901403, and CCF-1733556, the ARO under award W911NF2010081, and the Italian MIUR PRIN 2017 Project ALGADIMAR "Algorithms, Games, and Digital Market". Gabriele Farina is supported by a Facebook fellowship.

References

Basilico, N., Celli, A., Nittis, G. D., and Gatti, N. Team-maxmin equilibrium: efficiency bounds and algorithms. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pp. 356–362, 2017.

Bowling, M., Burch, N., Johanson, M., and Tammelin, O. Heads-up limit hold'em poker is solved. *Science*, 347 (6218), January 2015.

Brown, G. W. Iterative solutions of games by fictitious play. In Koopmans, T. C. (ed.), *Activity Analysis of Production and Allocation*, pp. 374–376. John Wiley & Sons, 1951.

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

Celli, A. and Gatti, N. Computational results for extensive-form adversarial team games. In *AAAI Conference on Artificial Intelligence (AAAI)*, pp. 965–972, 2018.

Farina, G. and Sandholm, T. Polynomial-time computation of optimal correlated equilibria in two-player extensive-form games with public chance moves and beyond. In *ArXiv preprint*, 2020.

Farina, G., Celli, A., Gatti, N., and Sandholm, T. Ex ante coordination and collusion in zero-sum multi-player extensive-form games. In *Advances in Neural Information Processing Systems*, pp. 9638–9648, 2018.

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

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

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

Kuhn, H. W. Extensive games and the problem of information. In Kuhn, H. W. and Tucker, A. W. (eds.), *Contributions to the Theory of Games*, volume 2 of *Annals of Mathematics Studies, 28*, pp. 193–216. Princeton University Press, Princeton, NJ, 1953.

Lanctot, M., Waugh, K., Zinkevich, M., and Bowling, M. Monte Carlo sampling for regret minimization in extensive games. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, 2009.

Lisỳ, V., Lanctot, M., and Bowling, M. Online Monte Carlo counterfactual regret minimization for search in imperfect information games. In *Proceedings of the 2015 international conference on autonomous agents and multiagent systems*, pp. 27–36, 2015.

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

Romanovskii, I. Reduction of a game with complete memory to a matrix game. *Soviet Mathematics*, 3, 1962.

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

Southey, F., Bowling, M., Larson, B., Piccione, C., Burch, N., Billings, D., and Rayner, C. Bayes' bluff: Opponent modelling in poker. In *Proceedings of the 21st Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, July 2005.

Tammelin, O., Burch, N., Johanson, M., and Bowling, M. Solving heads-up limit Texas hold'em. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

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

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

von Stengel, B. and Koller, D. Team-maxmin equilibria. *Games and Economic Behavior*, 21(1-2):309–321, 1997.

Zhang, Y. and An, B. Computing team-maxmin equilibria in zero-sum multiplayer extensive-form games. In *AAAI*

Conference on Artificial Intelligence (AAAI), pp. 2318–2325, 2020a.

Zhang, Y. and An, B. Converging to team-maxmin equilibria in zero-sum multiplayer games. In *International Conference on Machine Learning (ICML)*, 2020b.

Zhang, Y., An, B., and Černý, J. Computing ex ante coordinated team-maxmin equilibria in zero-sum multiplayer extensive-form games. In *ArXiv preprint:2009.12629*, 2020.