Computing *Ex Ante* Coordinated Team-Maxmin Equilibria in Zero-Sum Multiplayer Extensive-Form Games

Youzhi Zhang, Bo An, Jakub Černý
School of Computer Science and Engineering, Nanyang Technological University, Singapore
{yzhang137, boan}@ntu.edu.sg, cerny@disroot.org

Abstract

Computational game theory has many applications in the modern world in both adversarial situations and the optimization of social good. While there exist many algorithms for computing solutions in two-player interactions, finding optimal strategies in multiplayer interactions efficiently remains an open challenge. This paper focuses on computing the multiplayer Team-Maxmin Equilibrium with Coordination device (TMECor) in zero-sum extensive-form games. TMECor models scenarios when a team of players coordinates *ex ante* against an adversary. Such situations can be found in card games (e.g., in Bridge and Poker), when a team works together to beat a target player but communication is prohibited; and also in real world, e.g., in forest-protection operations, when coordinated groups have limited contact during interdicting illegal loggers. The existing algorithms struggle to find a TMECor efficiently because of their high computational costs. To compute a TMECor in larger games, we make the following key contributions: (1) we propose a hybrid-form strategy representation for the team, which preserves the set of equilibria; (2) we introduce a column-generation algorithm with a guaranteed finite-time convergence in the infinite strategy space based on a novel best-response oracle; (3) we develop an associated-representation technique for the exact representation of the multilinear terms in the best-response oracle; and (4) we experimentally show that our algorithm is several orders of magnitude faster than prior state-of-the-art algorithms in large games.

1 Introduction

One of the most important problems in artificial intelligence is to design algorithms for agents who make complex decisions in interactive environments (Russell and Norvig 2016). So far, researchers made significant progress mostly in non-cooperative two-player games, focusing on finding a Nash Equilibrium (NE) (Nash 1951; von Stengel 1996; Zinkevich et al. 2008) or a Stackelberg equilibrium (Conitzer and Sandholm 2006). These results paved the way for many applications, such as in security games that have high social impact (Sinha et al. 2018) or poker algorithms that defeated top human professionals (Moravčík et al. 2017; Brown and Sandholm 2018). However, the research in multiplayer games remains limited. Theoretical results were achieved only for games with specific structures (e.g., polymatrix games (Cai and Daskalakis 2011)), or there are no theoretical guarantees at all (e.g., the algorithm in Brown and Sandholm (2019)). Finding and playing NEs in multiplayer games is difficult due to the following two reasons. First, computing NEs is PPAD-complete even for three-player zero-sum games (Chen and Deng 2005). And second, NEs are neither unique nor exchangeable in multiplayer games, which makes it almost impossible for the players who choose their strategies independently to form an NE together. The research on multiplayer games hence focuses on alternative solution concepts with more favorable properties.

Team-Maxmin Equilibrium with Coordination device (TMECor, TMEsCor as a plural) (Celli and Gatti 2018; Farina et al. 2018) is a solution concept that models a situation when a team of players shares the same utility function and coordinates *ex ante* against an adversary. That is, the team members are allowed to discuss and agree on tactics before the game starts, but they cannot communicate during the game. Celli and Gatti (2018) show that *ex ante* coordination can be modelled using a coordination device, assuming that the adversary does not observe any signal from the device. The team members agree on a planned strategy (e.g., a mixed strategy) in the planning phase, and then, just before the game starts, the coordination device randomly picks a pure joint strategy (from the planned strategy) for the team members to act upon. A TMECor is an NE between the team (i.e., each team member has no incentive to deviate) and the adversary in a zero-sum multiplayer extensive-form game, and it has the properties of NEs in zero-sum two-player games (e.g., exchangeability). The study of TMECor is motivated by its ability to capture many real-world scenarios. For example, in multiplayer poker games, a team may play against an adversary player, but they cannot communicate and discuss their strategy during the game due to the rules. In Bridge, when the game reaches the phase of the play of the hand, two defenders who form a team play against the declarer. Or in security games in which several different groups (e.g., NGOs, the Madagascar National Parks, local police, and community volunteers) aim to protect forests from illegal logging (McCarthy et al. 2016), TMECor models the groups' inability to communicate while they try to interdict the escaping loggers.

TMECor has better properties than NE in multiplayer games; however, computing it is still difficult—it was shown to be FNP-hard (Celli and Gatti 2018). The problem can be formulated as a linear program (Celli and Gatti 2018), where the team plays joint normal-form strategies for all team members, but each member's normal-form strategy space is exponential in the size of the game tree. A Column Generation (CG) algorithm was hence proposed to compute a solution more efficiently (Celli and Gatti 2018). The most important component of the algorithm is a Best Response Oracle (BRO) that computes an optimal strategy of the team against the adversary's strategy, but it is in itself an NP-hard problem (Celli and Gatti 2018). A BRO can be formulated as a Mixed-Integer Linear Program (MILP) that involves a large number of integer variables (Celli and Gatti 2018; Farina et al. 2018). As a consequence, the existing approaches fail to scale up to larger scenarios (see Section 3 for details).

Main Contributions. The most significant outcome of our work is a new algorithm for computing a TMECor, which runs several orders of magnitude faster than the state-of-the-art algorithms and scales to much larger games. To design this algorithm, we make several key contributions. The first contribution is a new hybrid-form strategy representation for the team's strategy in a TMECor. Based on this representation, we develop a CG method that guarantees convergence to a TMECor in a finite number of iterations, despite the fact that the space of our hybrid-form strategies is infinite. The core component of the CG method is a novel BRO. Our BRO is formulated as a multilinear program in which the multilinear terms represent reaching probabilities for terminal nodes to reduce the number of involved integer variables. We show that the BRO can be transformed into an MILP exactly using another contribution: a novel global optimization technique called Associated Representation Technique (ART). Another essential property of ART is that it efficiently generates associated constraints for the equivalence relations between multilinear terms, which significantly speeds up the computation of the BRO's MILP formulation by reducing its space of feasible solutions. All together, our approach shows that formulating the problem as a multilinear program with global optimization techniques can be significantly faster than the direct formulation as a linear program.

2 Preliminaries

An imperfect-information extensive-form game (EFG) is defined by a tuple $(N, A, H, L, \chi, \rho, u, I)$ (Shoham and Leyton-Brown 2008). $N = \{1, \ldots, n\}$ denotes a finite set of players and A is a finite set of actions. H is a finite set of nonterminal decision nodes (sequences of actions (histories)) in the game, with L being a set of leaf (terminal) nodes. To each nonterminal decision node the function $\chi : H \to 2^A$ assigns a subset of possible actions to play, while function $\rho : H \to N \cup \{c\}$ identifies an acting player (c denotes chance). Moreover, we denote $H_i = \{h \mid \rho(h) = i, h \in H\}, \forall i \in N$. To determinate the outcomes, we use $u = (u_1, \ldots, u_n)$, where $u_i : L \to \mathbb{R}$ is player i's utility function assigning a finite utility to each terminal node. The imperfect observations are modelled through the set of information sets $I = (I_1, \ldots, I_n)$. I_i is the set of player i's information sets (a partition of H_i), such that $\rho(h_1) = \rho(h_2)$ and $\chi(h_1) = \chi(h_2)$ for any $I_{i,j} \in I_i$ with $h_1, h_2 \in I_{i,j}$. We assume that actions are unique to information sets, i.e., there exists only one information set $I_{i,j}$ such that $a \in \chi(I_{i,j})$ for any $a \in A$.

A sequence $\sigma_i \in \Sigma_i$ is an ordered list of actions taken by a single player i leading to some node h. $\varnothing$ stands for the empty sequence (i.e., a sequence with no actions). We use $\text{seq}_i(I_{i,j})$ to denote the player i's sequence leading to $I_{i,j} \in I_i$, and $\text{seq}_i(h)$ for the player i's sequence leading to $h \in H \cup L$. We assume perfect recall, i.e., for each player i and nodes $h_1, h_2 \in I_{i,j} \in I_i$, $\text{seq}_i(h_1) = \text{seq}_i(h_2)$. A realization plan (**sequence-form strategy**, also representing a behavioral strategy) of player i is a function $r_i : \Sigma \to [0, 1]$ satisfying the network-flow constraints:

$$r_i(\varnothing) = 1 \tag{1a}$$

$$\sum_{a \in \chi(I_{i,j})} r_i(\sigma_i a) = r_i(\sigma_i) \quad \forall I_{i,j} \in I_i, \sigma_i = \text{seq}_i(I_{i,j}) \tag{1b}$$

$$r_i(\sigma_i) \geq 0 \quad \forall \sigma_i \in \Sigma_i. \tag{1c}$$

Let $\mathcal{R}_i$ be the set of all (mixed) sequence-form strategies. We call r_i a pure sequence-form strategy if $r_i(\sigma_i) \in \{0, 1\}$ for all $\sigma_i \in \Sigma_i$. The set of pure sequence-form strategies is denoted as $\overline{\mathcal{R}}_i$ and naturally, we have $\overline{\mathcal{R}}_i \subseteq \mathcal{R}_i$.

A pure **normal-form strategy** of player i is a tuple $\pi_i \in \Pi_i = \times_{I_{i,j} \in I_i} \chi(I_{i,j})$ specifying one action to play in each information set of player i. In EFGs, the size of Π_i is exponential in the size of the game tree. A reduced normal-form strategy specifies actions only in reachable information sets due to earlier actions. Henceforth, we focus on reduced normal-form strategies (despite their size being still exponential in the size of the game tree) and refer to them as normal-form strategies. A mixed normal-form strategy x_i is a probability distribution over Π_i, denoted as $x_i \in \Delta(\Pi_i)$. For any pure (mixed) normal-form strategy there exists an equivalent pure (mixed) sequence-form strategy (von Stengel 1996), as shown in Example 1. We call two strategies of player i realization-equivalent (using notation $\sim$) if they induce the same probabilities for reaching nodes for all strategies of other players. Indeed, two strategies are realization-equivalent if and only if they correspond to the same realization plan (von Stengel 1996).

Team-Maxmin Equilibrium with Coordination Device (Celli and Gatti 2018; Farina et al. 2018) is a solution concept that models a scenario when a single team $T = \{1, \ldots, n-1\}$ with *ex ante* coordination plays against an adversary n. We assume the team shares the same utility $u_i(l) = u_j(l), \forall i, j \in T, l \in L$ and the utility of the adversary is $u_n(l) = -u_T(l) = -\sum_{i \in T} u_i(l), \forall l \in L$, i.e., it is a zero-sum EFG. The *ex ante* coordination means that the team players can communicate only before the game starts, through a coordination device. A pure strategy of the team in TMECor is represented by a joint normal-form strategy $\pi_T \in \Pi_T = \times_{i \in T} \Pi_i$. A mixed strategy x_T is then a probability distribution over Π_T, i.e., $x_T \in \Delta(\Pi_T)$. We use $L_{\pi_T, \sigma_n} \subseteq L$ to identify a set of terminal nodes reachable by strategy profile (π_T, σ_n), as shown in Example 1. Then, the extended utility function of the team's pure strategy π_T specifies the utility of profile (π_T, σ_n) due to chance nodes as $U_T(\pi_T, \sigma_n) = \sum_{l \in L_{\pi_T, \sigma_n}} u_T(l) c(l)$, where $c(l)$ denotes the chance probability of reaching l. For the team's mixed strat-

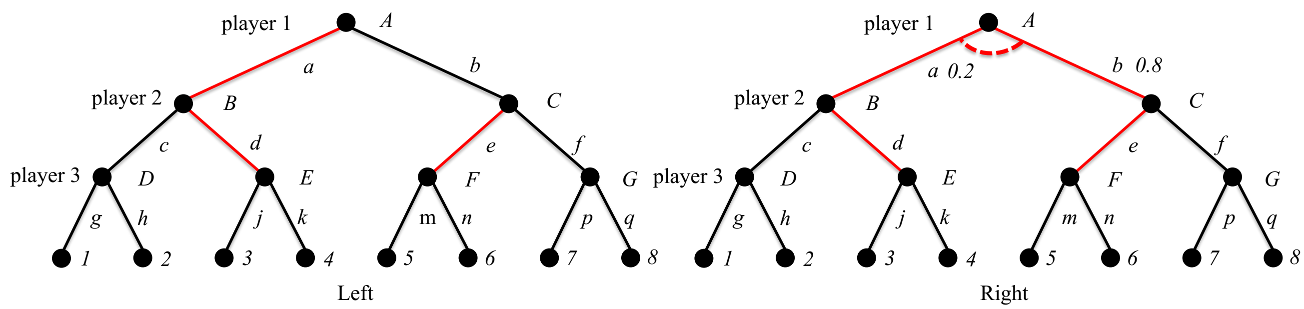

Figure 1: An example of an EFG with 3 players. Player 1 acts in information set A, player 2 acts in information sets B and C, and player 3, assuming the role of the adversary, acts in information sets D–G. Actions in the information sets are denoted a–q, and they may also represent sequences in this case. Nodes 1–8 are terminal nodes. Each side shows a different team's strategy.

egy x_T, we formulate the extended expected utility function as:

$$U_T(x_T, \sigma_n) = \sum_{\pi_T \in \Pi_T} U_T(\pi_T, \sigma_n) x_T(\pi_T). \quad (2)$$

Using a realization plan of the adversary, we write L_{π_T, r_n} to denote a set of terminal nodes reachable by strategy profile (π_T, r_n), as shown in Example 1, and $U_T(\pi_T, r_n) = \sum_{l \in L_{\pi_T, r_n}} r_n(\text{seq}_n(l)) u_T(l) c(l)$. For a mixed strategy,

$$U_T(x_T, r_n) = \sum_{\pi_T \in \Pi_T} U_T(\pi_T, r_n) x_T(\pi_T). \quad (3)$$

Example 1. *On the left side of Figure 1 we show an example of a strategy of a team consisting of players 1 and 2.* $\pi_1 = a$ *is a normal-form strategy of player 1, in which player 1 takes action* a *in his unique information set.* $\pi_2 = de$ *is then a normal-form strategy of player 2, such that player 2 takes actions* d *and* e *in his two information sets.* π_1 *is equivalent to a pure sequence-form strategy* r_1 *with* $r_1(a) = 1$*, and* π_2 *is equivalent to a pure sequence-form strategy* r_2 *with* $r_2(d) = r_2(e) = 1$*. Together, strategies* π_1 *and* π_2 *form a pure joint normal-form strategy of the team* $\pi_T = (\pi_1, \pi_2)$*. Given the adversary's sequence* $\sigma_3 = j$*, the set of terminal nodes reachable by* (π_T, σ_3) *is* $L_{\pi_T, \sigma_3} = \{3\}$*. In case that* $\sigma_3 = m$*, we have* $L_{\pi_T, \sigma_3} = \emptyset$*. Given any sequence-form strategy* r_3 *of the adversary, we have* $L_{\pi_T, r_3} = \{3, 4\}$.

A TMECor (x_T, r_n) is a Nash Equilibrium (NE) (i.e., $x_T \in \arg\max_{x_T} U_T(x_T, r_n)$ and $r_n \in \arg\max_{r_n} -U_T(x_T, r_n)$) in which the team is treated as a single player. Therefore, we have $x_T \in \arg\max_{x_T} \min_{r_n} U_T(x_T, r_n)$ due to the zero-sum assumption. Because the team members share the same utility, none of them has an incentive to deviate. A strategy profile is an ϵ-TMECor if neither the team nor the adversary is to gain more than ϵ if one of them deviates.

3 Related Work

The Team-Maxmin Equilibrium (TME) (von Stengel and Koller 1997; Celli and Gatti 2018; Zhang and An 2020b) is a solution concept closely related to TMECor, in which a team of players with the same utility function plays against an adversary independently, without coordination. Contrary to TMECor, the team members in TME are assumed to use behavioral strategies. Previous literature identified two main issues associated with TME. First, computing a TME, i.e., finding the optimal joint behavioral strategies of team members, is a non-convex FNP-hard optimization problem (Celli and Gatti 2018). Second, the equilibrium strategies of TME might be significantly suboptimal compared to TMECor because of the lack of coordination (Celli and Gatti 2018; Farina et al. 2018). Another complication arises if we attempt to coordinate the players using behavioral strategies. Due to the lack of communication between team members during the game, the team experiences imperfect recall. Because of imperfect recall, the behavioral strategies are not realization-equivalent to normal-form strategies induced by the coordination device, and thus can not capture the correlation between the team members' normal-form strategies. This was also shown to potentially result in considerable losses of utility to the team (Farina et al. 2018). Moreover, with imperfect recall, there is even no guarantee for the existence of an NE in behavioral strategies (Wichardt 2008). Using normal-formal strategies is hence crucial to TMECor.

Because the number of normal-form strategies is exponential in the size of the game, Celli and Gatti (2018) proposed a hybrid representation of players' strategies to speed up the computation of TMEsCor. In their representation, only the adversary uses sequence-form strategies. Their CG algorithm for TMEsCor formulates a BRO using an MILP with $|L|$ integer variables. However, the algorithm does not scale well as $|L|$ can be extremely large for games of moderate size (see Table 2). In an attempt to overcome this issue, Farina et al. (2018) developed a realization-form strategy representation that focuses on the probability distribution on terminal nodes of the game tree. Their representation is then used to derive an auxiliary game that represents the original game in the form of a set of subtrees. In each subtree, the adversary faces one team member, while they both use realization-form strategies. Rather than to derive a TMECor-specific strategy representation, their approach is to create a hybrid-form tree structure for the team. An essential drawback of this approach is that the auxiliary game is based on the realization-form strategy that is not executable from the game tree's root. It requires a cumbersome reconstruction algorithm to apply it (Celli et al. 2019). On the other hand, the auxiliary game enables to formulate a faster BRO with a number of integer variables equal to $\sum_{i \in T \setminus \{1\}} |\Sigma_i|$. Despite being a clear improvement over the approach of Celli and Gatti (2018), the BRO of Farina et al. (2018) is still inefficient in larger games with a higher number of sequences. Moreover, this BRO is not compatible with associated constraints (Zhang and An 2020a) that were shown to significantly improve the scalability by reducing the feasible solution space of an MILP. In this paper, we develop a hybrid-form strategy representation inspired by the previous literature, but which does not depend on the inexecutable realization-form strategies. Simultaneously, the representation gives rise to a novel BRO that considerably reduces the number of integer variables and is compatible with associated constraints. Moreover, we design a new global optimization technique called ART that does not depend on the recursive generation of associated constraints proposed by Zhang and An (2020a). ART also works for games with four or more players and significantly improves the scalability in experiments. Note that the concurrent work (Farina et al. 2020) solves games with only three players. We give more details on the related work in Appendix A.[1]

4 Hybrid-Form Column Generation with Efficiently Solvable Multilinear Oracle

Now we describe our novel approach for computing TMECor efficiently. First, we introduce a new strategy representation for the team, and we show how to use this representation to compute an exact TMECor. Then we propose a column-generation algorithm based on our representation, which iteratively calls a multilinear BRO. Finally, we develop a novel global optimization technique to solve our BRO efficiently.

4.1 Hybrid-Form Strategies for the Team

In our hybrid-form team strategy representation, one team member acts according to a (mixed) sequence-form strategy, while the other members use pure normal-form strategies.[2] Any team player can play the sequence-form strategy, and without loss of generality, we opt for player 1. A pure hybrid-form strategy of the team is defined by a tuple[3]:

$$f_T = (r_1, \pi_{T\setminus\{1\}}),$$

where $r_1 \in \mathcal{R}_1$, and $\pi_{T\setminus\{1\}} \in \Pi_{T\setminus\{1\}} = \times_{i\in T\setminus\{1\}}\Pi_i$, as shown in Example 2. We denote $\mathcal{F}_T$ the set of pure hybrid-form strategies and refer to $\overline{x}_T \in \Delta(\mathcal{F}_T)$ as a mixed hybrid-form strategy—a probability distribution over $\mathcal{F}_T$. Note that the number of pure hybrid-form strategies ($|\mathcal{F}_T|$) is infinite because each strategy in $\mathcal{R}_1$ corresponds to at least one pure hybrid-form strategy, and the size of $\mathcal{R}_1$ is infinite.

[1]Appendix of this paper can be found in the full version: https://arxiv.org/abs/2009.12629

[2]Note that all team players are free to use pure normal-form strategies–in that case, our algorithm still works, but it would significantly slow down the computation when compared with the hybrid-form strategies, as shown in the experiments. In contrast, if two or more team players use sequence-form strategies, the reaching probabilities for terminal nodes are no longer exactly representable by linear constraints (see Eqs.(6a) and (6b)).

[3]To differentiate between f_T and $\overline{x}_T$ (a probability distribution over the space of f_T defined later), we call f_T a "pure" hybrid-form strategy and $\overline{x}_T$ a "mixed" hybrid-form strategy.

Given a strategy profile (f_T, σ_n), we define an extended utility function of the team's pure strategy similarly as $U_T(\pi_T, \sigma_n)$ in Eq.(2): $U_T(f_T, \sigma_n) = \sum_{l\in L_{f_T,\sigma_n}} r_1(\text{seq}_1(l))u_T(l)c(l)$, with L_{f_T,σ_n} being the leafs reachable by the profile, as shown in Example 2. The expected utility of a mixed strategy is then:

$$U_T(\overline{x}_T, \sigma_n) = \textstyle\sum_{f_T\in\mathcal{F}_T} U_T(f_T, \sigma_n)\overline{x}_T(f_T).$$

Using a sequence-form strategy r_n of the adversary, an extended utility function of the team's pure strategy is defined (similarly to $U_T(\pi_T, r_n)$ as in Eq.(3)) as $U_T(f_T, r_n) = \sum_{l\in L_{f_T,r_n}} r_n(\text{seq}_n(l))r_1(\text{seq}_1(l))u_T(l)c(l)$, with L_{f_T,r_n} defined accordingly, as shown in Example 2. The corresponding expected utility of a mixed strategy is:

$$U_T(\overline{x}_T, r_n) = \textstyle\sum_{f_T\in\mathcal{F}_T} U_T(f_T, r_n)\overline{x}_T(f_T).$$

Example 2. *On the right side of Figure 1 we depict a sequence-form strategy of player 1 r_1 with $r_1(a) = 0.2$ and $r_1(b) = 0.8$, and a normal-form strategy of player 2 $\pi_2 = de$. The corresponding hybrid-form strategy is then $f_T = (r_1, \pi_2) = ((0.2, 0.8), de)$. r_1 is equivalent to a mixed normal-form strategy x_1 with $x_1(\pi_1 = a) = 0.2$ and $x_1(\pi_1' = b) = 0.8$, where $\pi_1 = a$ and $\pi_1' = b$ are pure normal-form strategies. The strategy f_T can be represented as a joint strategy $(x_1 = (0.2, 0.8), \pi_2 = de)$, which is equivalent to a mixed joint normal-form strategy x_T with $x_T(\pi_1, \pi_2) = 0.2$ and $x_T(\pi_1', \pi_2) = 0.8$. In this case, the mixed joint normal-form strategy x_T corresponds to a mixed hybrid-form strategy $\overline{x}_T$ with $\overline{x}_T(f_T) = 1$. Given the adversary's sequence $\sigma_3 = j$, the set of terminal nodes reachable by (f_T, σ_3) is denoted as $L_{f,\sigma_3} = \{3\}$. If $\sigma_3 = m$, the set $L_{f_T,\sigma_3} = \{5\}$. For an arbitrary sequence-form strategy r_3 of the adversary, we have $L_{f_T,r_3} = \{3, 4, 5, 6\}$.*

We define the support sets in a mixed normal-form strategy and a mixed hybrid-form strategy as:

$$S_{x_T} = \{\pi_T \mid x_T(\pi_T) > 0, \pi_T \in \Pi_T\},$$
$$S_{\overline{x}_T} = \{f_T \mid \overline{x}_T(f_T) > 0, f_T \in \mathcal{F}_T\}.$$

The first step in computing a TMECor using hybrid-form strategies is to show that the set of TMEsCor is preserved under this strategy representation. In other words, the utility of any player (the team or the adversary) in each TMECor with a hybrid-form strategy has to be the same as in a TMECor with a normal-form strategy of the team. Because we consider zero-sum games, it is enough to show that both strategy representations guarantee the same utility of the team. To prove this result, we only need to consider each sequence of the adversary once because a sequence-form strategy of the adversary is defined by the probability for taking each sequence. We first prove a more general result: for any mixed normal-form strategy of the team, there exists a mixed hybrid-form strategy such that the team obtains the same expected utility in both strategy representations with any strategy of the adversary, and vice versa. Recall that two strategies of the team are realization-equivalent if they both define the same probabilities for reaching nodes given any strategy of the adversary. Based on this definition, in the following two

$\mathcal{R}_i$	set of player i's mixed sequence-form strategies including r_i	$\mathcal{F}_T$	set of pure hybrid-form strategies including f_T
$\overline{\mathcal{R}}_i$	set of player i's pure sequence-form strategies	$\Delta(\Pi_T)$	set of mixed strategies including x_T
Π_T	set of the team's joint pure normal-form strategies including π_T	$\Delta(\mathcal{F}_T)$	set of mixed strategies including $\overline{x}_T$
$T \setminus 1$	set of the team members except player 1	$\sim$	realization equivalence
$\sigma_T(i)$	the team player i's sequence in the joint sequence σ_T	Σ_i	set of player i's sequences including σ_i
$\text{seq}_i(*)$	player i's sequence reaching a node/information set $*$	Σ_T	set of the team's joint sequences including σ_T
$w(\sigma_T)$	variable representing multilinear term $\prod_{i \in T} r_i(\sigma_T(i))$	S_x	support set of mixed strategy x with size $\lvert S_x \rvert$

Table 1: The notation used in Section 4, in addition to the standard EFG notation.

lemmas, we show the realization equivalence between the hybrid-form and the normal-form strategies, as shown in Example 2. We also show that two realization-equivalent strategies could have the same size of support sets in Lemma 1. All proofs can be found in Appendix B.

Lemma 1. *For every normal-form strategy $x_T \in \Delta(\Pi_T)$ there exists a realization-equivalent hybrid-form strategy $\overline{x}_T \in \Delta(\mathcal{F}_T)$ such that $|S_{x_T}| = |S_{\overline{x}_T}|$.*

Lemma 2. *For every hybrid-form strategy $\overline{x}_T \in \Delta(\mathcal{F}_T)$ there exists a realization-equivalent normal-form strategy $x_T \in \Delta(\Pi_T)$.*

Using the two lemmas, we can prove that our hybrid-form strategies preserve the set of TMEsCor[4].

Theorem 1. *For every normal-form strategy $x_T \in \Delta(\Pi_T)$ there exists a hybrid-form strategy $\overline{x}_T \in \Delta(\mathcal{F}_T)$ such that $U_T(x_T, \sigma_n) = U_T(\overline{x}_T, \sigma_n)$, $\forall \sigma_n \in \Sigma_n$, and vice versa.*

Because the set of TMEsCor will not change if hybrid-form strategies are played, the following linear program (LP) computes the equilibrium:

$$\max_{\overline{x}_T} v(\mathcal{I}_n(\varnothing)) \tag{4a}$$

$$v(\mathcal{I}_n(\sigma_n)) - \sum_{I_{n,j} \in I_n : \text{seq}_n(I_{n,j}) = \sigma_n} v(I_{n,j}) \le U_T(\overline{x}_T, \sigma_n) \quad \forall \sigma_n \in \Sigma_n \tag{4b}$$

$$\sum_{f_T \in \mathcal{F}_T} \overline{x}_T(f_T) = 1 \tag{4c}$$

$$\overline{x}_T(f_T) \ge 0 \quad \forall f_T \in \mathcal{F}_T. \tag{4d}$$

In this LP, $\mathcal{I}_n(\sigma_n)$ denotes an information set in which player n takes the last action of sequence σ_n, and $v(I_{n,j})$ is the expected utility of the team in each information set $I_{n,j}$. The adversary chooses the strategy minimizing the team's utility in each information set $\mathcal{I}_n(\sigma_n)$ (represented by Eq.(4b) (Bosansky et al. 2014)), and $v(\mathcal{I}_n(\varnothing))$ is hence the team's utility given $\overline{x}_T$.

4.2 Column Generation with a Multilinear Oracle

Although we can formulate the problem of finding a TMECor as LP (4), solving it requires enumerating all pure hybrid-form strategies in an infinite strategy space as variables, which is impractical. To address this problem, we introduce a CG algorithm with a Multilinear BRO (CMB), depicted in Algorithm 1. Our CMB starts from a restricted game $(\mathcal{F}'_T, \Sigma_n)$ ($\mathcal{F}'_T$ is a subset of $\mathcal{F}_T$) and proceeds iteratively. It computes a TMECor in a subgame and checks if there exists a better strategy outside the restricted game. If the answer is positive, it expands the restricted game, and terminates otherwise.

[4]Note that our proofs do not rely on the inexecutable realization-form strategies of Farina et al. (2018).

Algorithm 1: CG Algorithm with a Multilinear BRO

1 Initialize $\mathcal{F}'_T \leftarrow \{\text{any strategy}\}, \underline{v} \leftarrow 0, \overline{v} \leftarrow 1$;
2 **repeat**
3 $\quad (\underline{v}, \overline{x}_T, r_n) \leftarrow \text{CoreLP}(\mathcal{F}'_T, \Sigma_n)$;
4 $\quad (\overline{v}, f_T) \leftarrow \text{BRO}(r_n)$;
5 $\quad \mathcal{F}'_T \leftarrow \mathcal{F}'_T \cup \{f_T\}$;
6 **until** $\underline{v} = \overline{v}$;
7 **return** $(\overline{x}_T, r_n)$.

Now we describe this dynamic in more detail. In Algorithm 1, CoreLP$(\mathcal{F}'_T, \Sigma_n)$ in Line 3 computes a TMECor $(\overline{x}_T, r_n)$ with the team's utility $\underline{v}$ in the restricted game $(\mathcal{F}'_T, \Sigma_n)$ by solving LP (4). On the next line, the output f_T of our BRO(r_n) is a pure hybrid-form strategy maximizing the team's utility $\overline{v}$ when the adversary plays r_n, i.e., $\arg\max_{f_T \in \mathcal{F}_T} U_T(f_T, r_n)$. We will show how to compute it efficiently later. Thus, starting from the restricted $\mathcal{F}'_T$ (initialized at Line 1), CMB computes the equilibrium $(\overline{x}_T, r_n)$ with the team's utility $\underline{v}$ in the restricted game $(\mathcal{F}'_T, \Sigma_n)$ by calling CoreLP (Line 3), and finds the team's best response f_T with the team's utility $\overline{v}$ against the adversary's strategy r_n in the equilibrium of $(\mathcal{F}'_T, \Sigma_n)$ by calling BRO (Line 4). Then CMB expands $\mathcal{F}'_T$ if a new strategy f_T outside of $\mathcal{F}'_T$ is found (Line 5), otherwise CMB terminates (Line 7).

We aim to show that once the algorithm terminates, the equilibrium in the restricted game is an equilibrium in the full game $(\mathcal{F}_T, \Sigma_n)$. Unfortunately, most column-generation algorithms usually rely on the fact that the strategy space is finite, and the number of iterations is hence bounded to enforce a guarantee of convergence (Bosansky et al. 2014; McMahan, Gordon, and Blum 2003). Since a strategy space in our CMB is infinite, this argument can not hold. In the following proposition, we show that despite the number of pure hybrid-form strategies being infinite, the CMB will always terminate in a finite number of steps. The intuition behind our result is that if a best-response strategy f_T is added to the restricted game, instead of representing a single strategy, it stands for a whole subset of hybrid-form strategies in the BRO due to the mixed sequence-form strategy of the first player, and the number of these subsets of hybrid-form strategies is finite.

Proposition 1. *CMB converges in at most $2^{|\Pi_1|} \frac{|\Pi_T|}{|\Pi_1|}$ steps.*

Even though this result implies that there always exists a TMECor with a finite-sized support set for the team (i.e., at most $2^{|\Pi_1|} \frac{|\Pi_T|}{|\Pi_1|}$ strategies in the support set), it is a rather weak guarantee. We hence provide a tighter bound on the size of the team's support set[5]. That is, there exists a TMECor such that the size of the support set of the hybrid-form strategy will not be larger than the number of adversary's sequences, which is significantly smaller than $2^{|\Pi_1|}\frac{|\Pi_T|}{|\Pi_1|}$.

Proposition 2. *There is a TMECor $\overline{x}_T$ with $|S_{\overline{x}_T}| \leq |\Sigma_n|$.*

Our experimental results show that the actual number of iterations before CMB converges is significantly smaller than the bound from Proposition 1. We attribute this to the fact that at least one TMECor has a small support set, as shown in Proposition 2. Finally, we prove that the output of CMB is a TMECor in the full game because the team cannot find a better strategy outside the restricted game. That is, the output of CMB is a TMECor in the restricted game $(\mathcal{F}'_T, \Sigma_n)$ and also a TMECor in the full game $(\mathcal{F}_T, \Sigma_n)$.

Theorem 2. *CMB converges to a TMECor in $(\mathcal{F}_T, \Sigma_n)$.*

Moreover, we show that if the BRO checks for the existence of a best-response strategy with a gain at most ϵ outside the restricted game, the CMB converges to an approximate TMECor. This is achieved simply by changing the termination condition from $\overline{v} = \underline{v}$ to $\overline{v} - \underline{v} \leq \epsilon$. In this case, the output of CMB is a TMECor in the restricted game $(\mathcal{F}'_T, \Sigma_n)$ and also an approximate TMECor in the full game $(\mathcal{F}_T, \Sigma_n)$.

Proposition 3. *If CMB terminates with $\overline{v} - \underline{v} \leq \epsilon$, then its output $(\overline{x}_T, r_n)$ is an ϵ-TMECor in $(\mathcal{F}_T, \Sigma_n)$.*

Now we introduce our BRO. Note that:

$$\begin{aligned} U_T(f_T, r_n) &= \textstyle\sum_{l \in L_{f_T, r_n}} r_n(\mathrm{seq}_n(l)) r_1(\mathrm{seq}_1(l)) u_T(l) c(l) \\ &= \textstyle\sum_{l \in L} u_T(l) c(l) r_n(\mathrm{seq}_n(l)) \prod_{i \in T} r_i(\mathrm{seq}_i(l)). \end{aligned}$$

Then the BRO can be formulated as the following multilinear program that computes a best response f_T against a sequence-form strategy r_n of the adversary, i.e., $\arg\max_{f_T \in \mathcal{F}_T} U_T(f_T, r_n)$:

$$\max_{\times_{i \in T} r_i} \textstyle\sum_{l \in L} u_T(l) c(l) r_n(\mathrm{seq}_n(l)) \prod_{i \in T} r_i(\mathrm{seq}_i(l)) \quad \text{(5a)}$$

$$\text{Eqs.}(1a) - (1c) \quad \forall i \in T \quad \text{(5b)}$$

$$r_i(\sigma_i) \in \{0, 1\} \quad \forall \sigma_i \in \Sigma_i, i \in T \setminus \{1\} \quad \text{(5c)}$$

$$r_1(\sigma_1) \in [0, 1] \quad \forall \sigma_1 \in \Sigma_1, \quad \text{(5d)}$$

where $r_i \in \overline{\mathcal{R}}_i$ is realization-equivalent to $\pi_i \in \Pi_i$ for all i in $T \setminus \{1\}$ in a hybrid-form strategy f_T. The idea of the BRO is that it expresses the probability of all team players reaching each terminal node using a multilinear term $\prod_{i \in T} r_i(\mathrm{seq}_i(l))$ in $U_T(f_T, r_n)$ with only $\sum_{i \in T \setminus \{1\}} |\Sigma_i|$ integer variables.

4.3 Associated Representation Technique

Because problem (5) is multilinear and thus difficult to solve, we develop a novel global optimization technique for finding a solution efficiently. We call the method the Associated Representation Technique (ART). The ART represents the multilinear terms exactly through linear constraints. Moreover, it reduces the feasible solution space by using associated constraints for the equivalence relations between the individual multilinear terms. From the computational perspective, ART's two essential properties are that (i) it does not require recursive expansion to represent the multilinear terms exactly, and (ii) it generates the associated constraints efficiently.

Multilinear Representation (MR) First, we show how to transform problem (5) into an equivalent MILP exactly, without introducing new integer variables. For this purpose, for each multilinear term $w(\sigma_T) = \prod_{i \in T} r_i(\sigma_T(i))$, where $\sigma_T(i)$ is the sequence of player i in joint sequence $\sigma_T \in \Sigma_T$ with $\Sigma_T = \times_{i \in T} \Sigma_i$, $r_i \in \overline{\mathcal{R}}_i$ for all $i \in T \setminus \{1\}$, and $r_1 \in \mathcal{R}_1$, we introduce the following MR constraints:

$$0 \leq w(\sigma_T) \leq r_i(\sigma_T(i)) \quad \forall i \in T \setminus \{1\} \quad \text{(6a)}$$

$$0 \leq r_1(\sigma_T(1)) - w(\sigma_T) \leq n - 2 - \sum_{i \in T \setminus \{1\}} r_i(\sigma_T(i)). \quad \text{(6b)}$$

Note that the multilinear term $\prod_{i \in T} r_i(\sigma_T(i))$ is equal to the continuous variable $r_1(\sigma_T(1))$ if all binary variables are set to 1, and it is 0 if there is a binary variable with value 0. Now we show that variable $w(\sigma_T)$ in Eqs.(6a)–(6b) exactly represents the multilinear term $\prod_{i \in T} r_i(\sigma_T(i))$.

Proposition 4. *$\prod_{i \in T} r_i(\sigma_T(i))$ with $r_i \in \overline{\mathcal{R}}_i$ for all $i \in T \setminus \{1\}$ and $r_1 \in \mathcal{R}_1$ is exactly represented by $w(\sigma_T)$ in Eqs.(6a) and (6b).*

Efficient Generation of Associated Constraints By using the MR constraints, problem (5) becomes an MILP, which can be solved using a standard branch-and-bound approach with an LP relaxation (Morrison et al. 2016). However, relaxing the MR constraints may result in a much larger feasible solution space. To be more specific, as a consequence of making the variable $r_i(\sigma_T(i))$ real-valued, the variable $w(\sigma_T)$ may no longer exactly represent $\prod_{i \in T} r_i(\sigma_T(i))$ in Eqs.(6a) and (6b), as intended. Therefore, we aim to reduce the feasible solution space of $w(\sigma_T)$. For this purpose, we generate associated constraints enforcing equivalence relations between multilinear terms, that are based on network-flow constraints (1a)-(1c) for the sequence-form strategies. As an example, suppose that we have the multilinear terms w, w_1, w_2 and w' with $w = r_1(\sigma_1)w', w_1 = r_1(\sigma_1 a)w', w_2 = r_1(\sigma_1 b)w'$. A constraint for the sequence-form strategy r_1 requires that $r_1(\sigma_1) = r_1(\sigma_1 a) + r_1(\sigma_1 b)$. Therefore, we can introduce an associated constraint $w = w_1 + w_2$. Adding associated constraints immediately rules out some candidate solutions, which effectively reduces the MILP's solution space and results in faster computation.

Because the associated constraints are closely related to the network-flow constraints of sequence-form strategies, they can be generated in a similar manner—through information sets. And because the variable $w(\times_{i \in T}\mathrm{seq}_i(l))$ used by MR constraints to represent the product $\prod_{i \in T} r_i(\mathrm{seq}_i(l))$ involves all team members' sequences, the associated constraints are generated for all information sets of all members. For example, consider a four-player Kuhn poker game, in which a terminal node is reached by the team's joint sequence tuple (*J:/cccr:c, Q:/cccrc:c, T:/cccrcc:c*). The three sequences are taken by three team players in information

[5] A similar result was proven for normal-form strategies earlier (Celli and Gatti 2018); however, Proposition 2 is not its direct consequence as it relies on Lemma 1 that makes the connection between normal-form and hybrid-form strategies.

EFG	$\lvert L\rvert$	$\lvert\Sigma_i\rvert$	CMB	CMB/H	CMB/A	CMB/ART	CMB/ART/H	C18	F18
3K4	312	33	**0.7s**	0.7s	2.1s	4s	6s	6.8s	1.2s
3K6	1560	49	**2s**	2s	20s	191s	479s	>5h	12s
3K8	4368	65	**4s**	4s	497s	7160s	>5h		210s
3K10	9360	81	**5s**	6s	10530s	>5h			3541s
3K12	17160	97	**10s**	10s	>5h				>5h
$4L3_1$	30600	219	**68s**	165s					
$4L3_2$	638064	219	**1264s**	2155s					
3L3	249480	457	**4916s**	6500s					
4K9	99792	145	**2.6h**	3.8h					
3L5*	10020	1001	**4.4h**	>6h					

Table 2: The runtimes of algorithms computing TMECor. The difficulty of finding a solution increases from top to bottom. We use the notation '$>n$h' to indicate that an algorithm did not terminate after n hours on the current and all larger instances. We assume that the largest 3L5* instance has five cards, and team players do not take action "raising" in $4L3_1$ (6 cards) and $4L3_2$.

EFG	5K11	5K12	5K13	6K7	6K8	6K9	7K7
CMB	**35s**	**58s**	**104s**	**17s**	**64s**	**216s**	**56s**
CMBZ20	2802s	6319s	>3h	2009s	>3h	>3h	>15h

Table 3: The runtimes of the CMB algorithm and the CMBZ20 algorithm for computing TMECor in larger games. We use the same notation as in Table 2.

EFG	3K4	3K6	3K8	3K10	3K12	3L3
Iterations	14	36	47	45	52	1022
Support size	3	5	8	6	10	63

Table 4: The number of iterations until CMB converges, and the size of the support set of the team's TMECor strategy.

sets *J:/cccr:*, *Q:/cccrc:*, *T:/cccrcc:*, respectively. Assume that the information set *T:/cccrcc:* of player 3 is reachable by a sequence *T:/cc:c*. The information set contains a node (*J:/cccr:c, Q:/cccrc:c, T:/cc:c,K:/ccc:r*), specified by one sequence per each player. There are two possible actions that can be taken: action c and action f. The network-flow constraint associated with this information set is hence

$$r_3(\textit{T:/cc:c}) = r_3(\textit{T:/cccrcc:c}) + r_3(\textit{T:/cccrcc:f}).$$

The corresponding associated constraint is

$$\begin{aligned} &w(\textit{J:/cccr:c, Q:/cccrc:c, T:/cc:c}) \\ =&w(\textit{J:/cccr:c, Q:/cccrc:c, T:/cccrcc:c}) \\ &+ w(\textit{J:/cccr:c, Q:/cccrc:c, T:/cccrcc:f}). \end{aligned}$$

Now assume that there is another node (*J:/cccr:c, K:/cccrc:c, T:/cc:c, Q:/ccc:r*) in the same information set, and a terminal node with the team's joint sequence (*J:/cccr:c, K:/cccrc:c, T:/cccrcc:c*) reachable by action c of player 3. The following associated constraint is generated:

$$\begin{aligned} &w(\textit{J:/cccr:c, K:/cccrc:c, T:/cc:c}) \\ =&w(\textit{J:/cccr:c, K:/cccrc:c, T:/cccrcc:c}) \\ &+ w(\textit{J:/cccr:c, K:/cccrc:c, T:/cccrcc:f}). \end{aligned}$$

Using the same approach, we can generate associated constraints in this Kuhn poker game in all team players' information sets. More details can be found in Appendix C.

Therefore, in a general EFG, in each information set $I_{i,j}$ of a team member i, the algorithm for generating associated constraints needs to enumerate all the team's joint sequences leading to $I_{i,j}$, which correspond to different nodes in $I_{i,j}$. We denote this set of sequences as $\Sigma_T(I_{i,j})$ and use $\text{seq}_i(I_{i,j}) = \sigma_T(i)$ for all $\sigma_T = (\sigma_T(i), \sigma_{T\setminus\{i\}}) \in \Sigma_T(I_{i,j})$. The associated constraints for $I_{i,j}$ are then

$$\begin{aligned} w(\sigma_T) = &\textstyle\sum_{a\in\chi(I_{i,j})} w(\sigma_T(i)a, \sigma_{T\setminus\{i\}}) \\ &\forall \sigma_T \in \Sigma_T(I_{i,j}), I_{i,j} \in I_i, i \in T. \end{aligned} \tag{7}$$

Generating all the constraints can be thus done in time $O(\sum_{i\in T}\sum_{I_{i,j}\in I_i} |\Sigma_T(I_{i,j})|)$. The resulting MILP for representing problem (5) using Eqs.(6a)-(6b) and (7) can be formulated as follows:

$$\max_{\times_{i\in T} r_i} \sum_{l\in L} u_T(l)c(l)r_n(\text{seq}_n(l))w(\times_{i\in T}\text{seq}_i(l)) \tag{8a}$$

$$\text{Eqs.}(5b) - (5d), (7) \tag{8b}$$

$$\text{Eqs.}(6a) - (6b) \quad \forall w(\times_{i\in T}\text{seq}_i(l)), l \in L. \tag{8c}$$

Our final theorem proves that associated constraints preserve the sequence-form strategy space, making the solution of formulation (8) also a feasible solution of our BRO in CMB. The intuition is that associated constraints are consistent with the sequence-form constraints and hence do not alter the space of feasible sequence-form solutions in Problem (5).

Theorem 3. *The solution of Problem (5) solves Problem (8).*

Proposition 4 guarantees that solutions of both problems will share the same value. The result, however, is even stronger: the optimal solution of Problem (8) is also optimal for Problem (5). Thus, it is a best response against the adversary's strategy r_n.

Corollary 1. *For any strategy r_n of the adversary, the optimal solution of Problem (8) is a best response against r_n.*

5 Experimental Evaluation

Finally, we demonstrate the performance of our CMB algorithm. We compare CMB to the previous state-of-the-art algorithms: (i) the original CG algorithm in Celli and Gatti (2018) (referred to as C18); (ii) the CG with the BRO proposed by Farina et al. (2018) (referred to as F18), and (iii) the CMB with the associated constraints generation algorithm of Zhang and An (2020a) (referred to as CMBZ20). We use two standard EFG domains for evaluating the algorithms: (i)

ϵ	0.1	0.08	0.06	0.04	0.02	0.01	0.008
CMB	**0.41s**	**0.41s**	**0.41s**	**0.50s**	**0.50s**	**0.58s**	**0.58s**
FTP	0.55s	0.71s	0.82s	1.3s	3.8s	8.0s	11.0s
CMB	**0.08s**	**0.08s**	**0.16s**	**0.16s**	**0.29s**	**0.50s**	**0.87s**
FTP	4.0s	6.6s	8.1s	15.5s	72.3s	181s	243s
CMB	**0.23s**	**0.23s**	**0.31s**	**0.31s**	**0.64s**	**1.1s**	**1.1s**
FTP	34.6s	34.6s	67.7s	94.9s	171s	382s	458s
CMB	**2s**	**3s**	**6s**	**13s**	**41s**	**87s**	**111s**
FTP	228s	307s	458s	689s	1574s	4882s	>5h
CMB	**5s**	**7s**	**13s**	**28s**	**93s**	**745s**	**1533s**
FTP	188s	251s	362s	619s	>5h		
CMB	**23s**	**27s**	**37s**	**108s**	**490s**	**3357s**	**8221s**
FTP	164s	215s	371s	920s	>5h		

Table 5: The runtimes of algorithms computing $\epsilon\Delta_u$-TMEsCor. The games from the top to the bottom are 3K4, 3K8, 3K12, 3L3, 3L4, and 3L5. $\Delta_u = 6$ for $3Kr$ and $\Delta_u = 21$ for $3Lr$. The sizes and notations are the same as in Table 2. In addition, $|L| \approx 10^6$ with $|\Sigma_i| = 801$ for 3L4, and $|L| \approx 3 \cdot 10^6$ with $|\Sigma_i| = 1241$ for 3L5.

EFG		3K8	3K9	3K10	3K11	3K12
t	TMECor	**4s**	**4s**	**5s**	**10s**	**10s**
	TME	4s	5s	84s	437s	118s
u	TMECor	**-0.019**	**-0.018**	**-0.016**	**-0.015**	**-0.014**
	TME	-0.066	-0.044	-0.068	-0.050	-0.055
	Gap	71%	59%	77%	71%	74%

Table 6: The runtimes t and the team's utilities u for computed TMEs and TMEsCor solutions. We calculate the gap as the relative distance between the team's utility (u_{Cor}) in a TMECor and the one (u_{TME}) in a TME, i.e., $\frac{|u_{TME}-u_{Cor}|}{|u_{TME}|} \times 100\%$. Greater gap indicates that the team will lose more if it opts for the TME strategy.

the Kuhn poker, and (ii) the Leduc Hold'em poker. Formal definitions of the domains can be found in Appendix D. All players have the same number of sequences in these games, and we use $|\Sigma_i|$ to represent this number of sequences. We denote an n-player Kuhn instance with r ranks (i.e., r cards) as nKr, and refer to an n-player Leduc Hold'em instance with r ranks (i.e., $3r$ cards) as nLr. Without a loss of generality, the last player assumes the role of the adversary. All (MI)LPs are solved by CPLEX 12.9 on a machine with 6-core 3.6GHz CPU and 32GB of memory.

Runtimes. We present the runtime results in Table 2. We omit the runtimes of CMBZ20 because it performs similarly to CMB, but evaluate their differences further in larger games and report the results in Table 3. For assessing ablations, we compare CMB to its four variant: (i) CMB without associated constraints (referred to as CMB/A); (ii) CMB that uses continuous variables to represent reaching probabilities for terminal nodes without using our ART and BRO (referred to as CMB/ART); (iii) CMB with BRO generating joint pure normal-form strategies instead of hybrid-form strategies (referred to as CMB/H), and (iv) CMB/ART/H—a combination of (ii) and (iii). The results clearly show that CMB is several orders of magnitude faster than the reference algorithms. Moreover, CMB also outperforms all ablation algorithms, which strongly suggests that each component of CMB significantly boosts its performance.

Convergence and Supports. In Table 4, we report the number of iterations CMB needs to converge to an exact TMECor, together with the team's equilibrium strategy support size. The number of iterations is significantly smaller than the theoretical upper bound $2^{33} \times 33$ derived in Proposition 1[6]. The support sets in TMECor also remain small.

Approximation. In the next experiment, we evaluate CMB's ability to compute an $\epsilon\Delta_u$-TMEsCor, where Δ_u is the difference between the maximum and minimum achievable utility of the team. We compare CMB to Fictitious Team Play (FTP) (Farina et al. 2018). We use the setting of FTP reported in Farina et al. (2018), including their BRO's time limit of 15s. Note that their BRO cannot run on large games otherwise. The results in Table 5 show that CMB runs significantly faster than FTP. For example, CMB is at least two orders of magnitude faster than FTP on large games with small ϵ, e.g., $\epsilon = 0.01$. Moreover, according to the results, it is almost impossible for FTP to converge to an $\epsilon\Delta_u$-TMECor with an even smaller ϵ (e.g., $\epsilon = 0.0001$), let alone an exact TMECor. For 3K4—the smallest game in our experiments—FTP is unable to converge to an $\epsilon\Delta_u$-TMECor with $\epsilon = 0.0004$ in 100 hours[7]. In contrast, CMB computes an exact TMECor within 0.7s, as shown in Table 2.

Comparison to TME. The last experiment demonstrates the difference between TMECor and TME discussed in Section 3, both in runtime and the team's utility. Previous literature has shown that the team suffers large losses in utility when resorting to TME strategies instead of TMECor strategies in 3K3–3K7 (Farina et al. 2018). In Table 6, we report the results on larger 3K8–3K12. Because of the encountered difficulty to compute an exact TME, we use the state-of-the-art algorithm of Zhang and An (2020a) to approximate the TME by computing an $\epsilon\Delta_u$-TME with $\epsilon = 0.01$. The results show that approximating a TME takes significantly longer than computing a TMECor, while at the same time, the TME strategies are inferior to the TMECor strategies.

6 Conclusion and Future Work

We propose a new algorithm (CMB) for finding a TMECor in large zero-sum multiplayer EFGs. Our algorithm is based on a novel hybrid-form strategy representation of the team, which gives rise to a column-generation method with guaranteed convergence in finite time. The heart of the algorithm is a multilinear best-response oracle that can be queried efficiently using our associated representation technique. We show that our algorithm computes a TMECor significantly faster than previous state-of-the-art baselines. In the future, we can explore the tighter theoretical upper bound for the number of iterations because CMB requires very few iterations in experiments. Due to the difficulty of computing the best-response oracle, we would like to explore the possibility of improving our oracle by reinforcement learning (Timbers et al. 2020) or regret minimization (Celli et al. 2019).

[6] $|\Sigma_i| = 33$ in 3K4, whereas $|\Pi_i|$ is significantly greater.

[7] ϵ reaches 0.0005 in 2674s but then it fluctuates around 0.001.

Acknowledgements

This research is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG-RP-2019-0013), National Satellite of Excellence in Trustworthy Software Systems (Award No: NSOE-TSS2019-01), the SIMTech-NTU Joint Laboratory on Complex System, and NTU.

Broader Impact

Game-theoretic solutions, including our algorithm, have both descriptive and prescriptive applications in suitable competitive environments, including businesses, politics, or even gambling. Finding the equilibria helps to understand people's behavior when interacting in dynamic situations and makes it easier to construct effective decisions to optimize multiagent systems. For example, the manufacturers in competitive markets may find better pricing strategies if they consider the decision-making of their competitors. While abusing the theories, e.g., by gamblers in casinos, is also feasible, the same approach also allows for identifying strategic violators by predicting their behavior. A notable drawback of traditional solution concepts like TMECor is their dependence on involved players' rational behavior. In case we suspect them to behave irrationally, we have to extend our models.

References

Bosansky, B.; Kiekintveld, C.; Lisy, V.; and Pechoucek, M. 2014. An exact double-oracle algorithm for zero-sum extensive-form games with imperfect information. *Journal of Artificial Intelligence Research* 51: 829–866.

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

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

Cai, Y.; and Daskalakis, C. 2011. On minmax theorems for multiplayer games. In *SODA*, 217–234.

Celli, A.; and Gatti, N. 2018. Computational results for extensive-form adversarial team games. In *AAAI*, 965–972.

Celli, A.; Marchesi, A.; Bianchi, T.; and Gatti, N. 2019. Learning to correlate in multi-player general-sum sequential games. In *NeurIPS*, 13055–13065.

Chen, X.; and Deng, X. 2005. 3-Nash is PPAD-complete. In *Electronic Colloquium on Computational Complexity*, volume 134, 2–29.

Conitzer, V.; and Sandholm, T. 2006. Computing the optimal strategy to commit to. In *EC*, 82–90.

Farina, G.; Celli, A.; Gatti, N.; and Sandholm, T. 2018. Ex ante coordination and collusion in zero-sum multi-player extensive-form games. In *NeurIPS*, 9638–9648.

Farina, G.; Celli, A.; Gatti, N.; and Sandholm, T. 2020. Faster Algorithms for Optimal Ex-Ante Coordinated Collusive Strategies in Extensive-Form Zero-Sum Games. *arXiv preprint arXiv:2009.10061* .

McCarthy, S. M.; Tambe, M.; Kiekintveld, C.; Gore, M. L.; and Killion, A. 2016. Preventing illegal logging: Simultaneous optimization of resource teams and tactics for security. In *AAAI*, 3880–3886.

McMahan, H. B.; Gordon, G. J.; and Blum, A. 2003. Planning in the presence of cost functions controlled by an adversary. In *ICML*, 536–543.

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 no-limit poker. *Science* .

Morrison, D. R.; Jacobson, S. H.; Sauppe, J. J.; and Sewell, E. C. 2016. Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. *Discrete Optimization* 19: 79–102.

Nash, J. 1951. Non-cooperative games. *Annals of Mathematics* 286–295.

Russell, S. J.; and Norvig, P. 2016. *Artificial Intelligence: A Modern Approach*. Malaysia; Pearson Education Limited.

Shoham, Y.; and Leyton-Brown, K. 2008. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press.

Sinha, A.; Fang, F.; An, B.; Kiekintveld, C.; and Tambe, M. 2018. Stackelberg Security Games: Looking Beyond a Decade of Success. In *IJCAI*, 5494–5501.

Timbers, F.; Lockhart, E.; Schmid, M.; Lanctot, M.; and Bowling, M. 2020. Approximate exploitability: Learning a best response in large games. *arXiv preprint arXiv:2004.09677* .

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

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

Wichardt, P. C. 2008. Existence of Nash equilibria in finite extensive form games with imperfect recall: A counterexample. *Games and Economic Behavior* 63(1): 366–369.

Zhang, Y.; and An, B. 2020a. Computing team-maxmin equilibria in zero-sum multiplayer extensive-form games. In *AAAI*.

Zhang, Y.; and An, B. 2020b. Converging to Team-Maxmin Equilibria in Zero-Sum Multiplayer Games. In *ICML*, 11033–11043.

Zinkevich, M.; Johanson, M.; Bowling, M.; and Piccione, C. 2008. Regret minimization in games with incomplete information. In *NeurIPS*, 1729–1736.