Computing Team-Maxmin Equilibria in Zero-Sum Multiplayer Extensive-Form Games

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

Abstract

The study of finding the equilibrium for multiplayer games is challenging. This paper focuses on computing Team-Maxmin Equilibria (TMEs) in zero-sum multiplayer Extensive-Form Games (EFGs), which describes the optimal strategies for a team of players who share the same goal but they take actions independently against an adversary. TMEs can capture many realistic scenarios, including: 1) a team of players play against a target player in poker games; and 2) defense resources schedule and patrol independently in security games. However, the study of efficiently finding TMEs within any given accuracy in EFGs is almost completely unexplored. To fill this gap, we first study the inefficiency caused by computing the equilibrium where team players correlate their strategies and then transforming it into the mixed strategy profile of the team and show that this inefficiency can be arbitrarily large. Second, to efficiently solve the non-convex program for finding TMEs directly, we develop the Associated Recursive Asynchronous Multiparametric Disaggregation Technique (ARAMDT) to approximate multilinear terms in the program with two novel techniques: 1) an asynchronous precision method to reduce the number of constraints and variables for approximation by using different precision levels to approximate these terms; and 2) an associated constraint method to reduce the feasible solution space of the mixed-integer linear program resulting from ARAMDT by exploiting the relation between these terms. Third, we develop a novel iterative algorithm to efficiently compute TMEs within any given accuracy based on ARAMDT. Our algorithm is orders of magnitude faster than baselines in the experimental evaluation.

1 Introduction

The design of algorithms for agents to make complex decisions in an interactive environment is an important component in artificial intelligence (Russell and Norvig 2016). Recently, the computational study in the non-cooperative environment has achieved many results for two-player games, for example, security games (Sinha et al. 2018) and poker games (Brown and Sandholm 2018). The key solution concepts behind these results are the well-known Nash Equilibrium (NE) (Nash 1951) and Stackelberg equilibrium (Conitzer and Sandholm 2006). However, there are fewer results about multiplayer games except for games with special structures, e.g., congestion games (Shoham and Leyton-Brown 2008), or the algorithm without theoretical guarantee (Brown and Sandholm 2019). It is not only because it is hard to compute the NE, which is PPAD-complete (Chen and Deng 2005) even for 3-player games, but also because the NE is not unique (or not exchangeable) in multiplayer games, which will make it difficult for each player to select a strategy independently and then form an NE with other players' strategies (Brown and Sandholm 2019).

The Team-Maxmin Equilibrium (TME) is a solution concept for the situation where a team of players with the same utility function take actions independently against an adversary (von Stengel and Koller 1997; Basilico et al. 2017; Celli and Gatti 2018). TMEs have some fascinating properties, e.g., it is the NE maximizing the utility of the team and always exists. Specifically, it is unique in general, which can avoid the equilibrium selection problem. Moreover, TMEs can capture many real-world scenarios. For example, in multiplayer poker games, all except one of them may form a team and play against the target player, i.e., they share the same goal but take actions independently (e.g., the setting of poker games in Brown and Sandholm (2019) avoids that players can communicate to correlate their actions). In addition to recreational applications, in security games, the US Coast Guard and other different police departments schedule and patrol independently against the attacker at major ports such as the New York port (Jiang et al. 2013).

However, it is challenging to compute a TME in Extensive-Form Games (EFGs), which is FNP-hard and formulated as a non-convex programming problem (Celli and Gatti 2018) even each player's (sequence-form) strategy space is linear in the size of EFGs. The current approach (Celli and Gatti 2018) to compute this TME is only to solve the non-convex problem directly by using the state-of-the-art global optimization solver BARON (Khajavirad and Sahinidis 2018). Unfortunately, BARON does not scale well (Celli and Gatti 2018). On the other hand, in EFGs, some efficient algorithms (Celli and Gatti 2018; Farina et al. 2018) are proposed to compute Correlated-Team Maxmin Equilibria (CTMEs), where team players correlate their strategies. However, if team players transform a CTME into the mixed strategy profile, they may obtain an arbitrarily large loss (see the next section for details). Therefore, the study of efficiently finding TMEs within any given accuracy, especially in EFGs, is almost completely unexplored. This paper will develop very scalable algorithms to compute TMEs.

Main Contributions. We first study the inefficiency caused by computing a CTME and then transforming it into the mixed strategy profile of the team and theoretically show that this inefficiency can be arbitrarily large in EFGs. Second, to efficiently solve the non-convex program for finding TMEs directly, we develop the Associated Recursive Asynchronous Multiparametric Disaggregation Technique (ARAMDT) to approximate multilinear terms in the program by using a digit-wise discretization of one variable in the term and then transform the program into a mixed-integer linear program (MILP) with two novel techniques: 1) an asynchronous precision method, where we can use different precision levels to approximate these terms even they include the same discretized variable to reduce the number of constraints and variables for approximation; and 2) an associated constraint method, where we exploit the relation between these terms to reduce the feasible solution space of MILP in ARAMDT. To our best knowledge, both techniques have not been considered in the literature of global optimization. Third, we develop a novel algorithm to efficiently compute TMEs within any given accuracy based on ARAMDT, which uses novel techniques to iteratively increase the precision levels for part of nonlinear terms. Finally, we evaluate our algorithm by experiments showing that our algorithm is dramatically faster than baselines.

2 Related Work

The solution concept similar to the TME is the CTME, where team players correlate their strategies by exploiting a mediator recommending actions to them through communication (Basilico et al. 2017). CTMEs in EFGs (Celli and Gatti 2018) include: 1) CTMEs with communication device (CTMECom) for the situation where team players can communicate and correlate their actions before the play of the game and during the game's execution; and 2) CTMEs with correlation device (CTMECor) for the situation where team players communicate and correlate their actions only before the play (ex ante coordination (Farina et al. 2018)). A CTMECom be computed in polynomial time, but computing a CTMECor is FNP-hard, where each team member's (normal-form) strategy space is exponential in the size of EFGs and efficient algorithms (Celli and Gatti 2018; Farina et al. 2018) are proposed. However, team players in a TME have no communication and then do not correlate their actions (Celli and Gatti 2018). To compute a TME in EFGs, one approach is to compute a CTME first and then transform it into the mixed strategy profile of the team, motivated by the algorithm in normal-form games (Basilico et al. 2017). Unfortunately, this approach may give the team a huge loss because of the large gap between the team's utility obtained by this approach and the TME value as shown in the previous experiments (Basilico et al. 2017). We theoretically show that this loss can be arbitrarily large.

In the domain of global optimization, the Multiparametric Disaggregation Technique (MDT) (Kolodziej, Castro, and Grossmann 2013; Castro 2016; Andrade et al. 2019) for approximating bilinear terms is a strong competitor of BARON. MDT outperforms the piecewise McCormick relaxation (Kolodziej, Castro, and Grossmann 2013; Castro 2016) and has been applied for solving game-theoretical problems (Wang, Guo, and An 2017; Čermák et al. 2018). To approximate multilinear terms, we can recursively transform multilinear terms to bilinear terms and then apply MDT inspired by the recursive McCormick relaxation (Ryoo and Sahinidis 2001). However, the current MDT approach cannot compute TMEs efficiently because the number of bilinear terms in EFGs is large (e.g., there are hundreds of bilinear terms in small Kuhn poker games) and MDT will use a large number of constraints and integer variables for approximating each term. Therefore, instead of applying MDT directly, we develop novel techniques.

3 Preliminaries

An imperfect-information Extensive-Form Game (EFG) (Shoham and Leyton-Brown 2008) is a tuple $(N, A, H, L, \chi, \rho, \mu, u, I)$, where: $N = \{1, \ldots, n\}$ is a set of players, A is a set of actions, H is a set of nonterminal nodes, L is a set of terminal nodes, $\chi : H \to 2^A$ is the action function assigning a set of possible actions to each nonterminal node, $\rho : H \to N$ is the player function assigning an acting player to each nonterminal node ($H_i = \{h \mid \rho(h) = i, h \in H\}, \forall i \in N$), $\mu : H \times A \leftarrow H \cup L$ is the successor function, which maps a nonterminal node and an action to a new node, $u = (u_1, \ldots, u_n)$ is the set of players' utility functions where $u_i : L \to \mathbb{R}$ assigns utilities to terminal nodes for player i, and $I = (I_1, \ldots, I_n)$ is the set of information sets where I_i is a partition of H_i such that, $\rho(h_1) = \rho(h_2)$ and $\chi(h_1) = \chi(h_2)$ for any $h_1, h_2 \in H_i$ whenever there exists $I_{i,j} \in I_i$ with $h_1 \in I_{i,j}$ and $h_2 \in I_{i,j}$. Without loss of generality, we assume, for each action $a \in A$, there is unique $I_{i,j}$ such that $a \in \chi(I_{i,j})$. In games with perfect recall, for each player i and each $I_{i,j} \in I_i$, nodes in $I_{i,j}$ share the same sequence of moves of player i on the paths from the root.

A pure normal-form plan of player i is a tuple $\pi \in \Pi_i = \times_{I_{i,j} \in I_i} \chi(I_{i,j})$ assigning an action to each information set of player i. A normal-form strategy x_i is a probability distribution over Π_i, i.e., $x_i \in \Delta(\Pi_i)$. A behavioral strategy β_i defines a probability distribution over $\chi(I_{i,j})$ for each information set of player i. A sequence (σ_i) of actions of player i, defined by a node $h \in H \cup L$ of the game tree, is the ordered set of player i's actions that are on the path from the root to h. Let Σ_i define the set of sequences of player i. Let $\emptyset$ denote the fictitious sequence corresponding to the root. A realization plan $r_i : \Sigma \to [0, 1]$ is a function assigning a probability to be played to each sequence, which satisfies the following constraints.

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

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

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

where $\mathrm{seq}_i(I_{i,j})$ is the sequence leading to $I_{i,j}$.

Example 1. *In the 3-player Kuhn poker game, the information set includes the card the payer has and the observed actions taking by players in turn. For example, J:/ccrc: is information set of player 2, where player 2 holds card J and she has observed the actions 'c,c,r,c' of all players in turn. Now, in information set J:/ccrc: reached by sequence J:/c:c of player 2, player 2 has two actions: calling (c) and folding (f). Therefore, by Eq.(1b), we have* $r_2(J\text{:/c:c}) = r_2(J\text{:/ccrc:c}) + r_2(J\text{:/ccrc:f})$.

The Team-Maxmin Equilibrium (TME) defined as $\arg\max_{r_1,\ldots,r_{n-1}}\min_{r_n}\sum_{\sigma=\times_{i\in N}\sigma_i,\sigma\in\Sigma}U_T(\sigma)\prod_{i=1}^{n}r_i(\sigma_i)$ (Celli and Gatti 2018), captures the scenario where a single team $T = \{1,\ldots,n-1\}$ plays against an adversary n with $u_i = u_j(\forall i,j \in T)$ and $u_n = -u_T = -\sum_{i\in T}u_i$, and team players take actions independently in a zero-sum EFG. Here, U_T denotes the team's utility with $U_T(\sigma) = \sum_{l\in L'}u_T(l)c(l)$ if at least a terminal node $l \in L' \subseteq L$ is reached by the joint plan σ (L' is the set of those terminal nodes) with the chance $c(l)$ determined by chance nodes, and $U_T(\sigma) = 0$ otherwise. A TME is a Nash Equilibrium (NE) maximizing the team's utility and is unique in general. Let the TME value be the utility of the team under any TME. For an ϵ-TME, both the team and the adversary cannot gain more than ϵ if only one player deviates from his strategy, and the difference between the ϵ-TME value and the TME value is not larger than ϵ. A TME can be computed by the following non-convex program (Celli and Gatti 2018):

$$\max_{r_1,\ldots,r_{n-1}} v(\mathcal{I}_n(\emptyset)) \tag{2a}$$

$$v(\mathcal{I}_n(\sigma_n)) - \sum_{I_{n,j}\in I_n:\text{seq}_n(I_{n,j})=\sigma_n} v(I_{n,j}) \tag{2b}$$

$$\leq \sum_{\sigma_T\in\Sigma_T} U_T(\sigma_T,\sigma_n)\prod_{i\in T} r_i(\sigma_T(i)) \quad \forall \sigma_n \in \Sigma_n$$

$$\text{Eqs.}(1a) - (1c) \quad \forall i \in T \tag{2c}$$

where $\mathcal{I}_n(\sigma_n)$ is the information set where player n takes the last action of sequence σ_n, $v : I_n \to \mathbb{R}$ where $v(I_{n,j})$ is the team's expected utility in information set $I_{n,j}$, σ_T is the team's joint sequence (i.e., $\times_{i\in T}\sigma_i$), $\sigma_T(i)$ is the sequence of player i in σ_T, and Σ_T is the set of these joint sequences. By Eq.(2b), the adversary chooses the action minimizing the team's utility in each information set $\mathcal{I}_n(\sigma_n)$.

4 The Inefficiency of Correlated Strategies

It is very difficult to solve the non-convex Problem (2) (Celli and Gatti 2018) even using the state-of-the-art global optimization solver BARON (Khajavirad and Sahinidis 2018). Instead of solving Problem (2) to compute a TME, one potential approach is to compute a CTME (see Section 2) first and then transform it into the team's Mixed Strategy Profile (Transformed-MSP(TMSP)), as done in normal-form games (Basilico et al. 2017). This section shows that this approach can cause an arbitrarily large loss.

To measure the inefficiency of this TMSP, we define the Price of Correlated strategies (PoC)[1], i.e., the inefficiency caused by that team players use this TMSP when they take actions independently. Formally, PoC $= \frac{v_m}{v_c}$, where v_m is the TME value and v_c is the team's utility obtained from the TMSP. The best algorithm (Basilico et al. 2017), to the best of our knowledge, to obtain a TMSP is: Given a CTME strategy for the team: $x \in \Delta(\Pi_T)$ (the probability distribution over the set of joint normal-formal strategies), player 1's mixed strategy $x_1(\pi_1) = \sum_{\pi'\in\Pi_{T\setminus\{1\}}} x(\pi_1,\pi')(\forall\pi_1 \in \Pi_1)$; player j's mixed strategy $x_j(\pi_j) = \frac{1}{|\text{sup}_j|}$ if $|\text{sup}_j| > 0$, otherwise $x_j(\pi_j) = 0$ $(\forall j \in T \setminus \{1\}, \pi_j \in \Pi_j)$ where $\text{sup}_j = \{\pi_j \mid \pi_j \in \Pi_j, \exists\pi_T, \pi_j \in \pi_T, x(\pi_T) > 0\}$, i.e., the set of strategies player j plays with nonzero probability in x. This algorithm returns the best strategy for the team among the strategies obtained by exchanging each player of the team with player 1. From previous experiments (Basilico et al. 2017), we can see the large gap between v_m and v_c in normal-form games. Now we theoretically show that PoC can be arbitrarily large.

Theorem 1. *PoC can be arbitrarily large in EFGs.*

Proof. Consider an EFG with n players ($n-1$ teammates) and $m(>2)$ actions per player's decision node where each level of the game tree forms a unique information set belonging to one player (see Figure 1 for a game tree with $n = 3$), and the payoffs of the team at terminal nodes ($\pi_i = a(1 \leq a \leq m)$ is a pure normal-form strategy) are:

$$U_T(\pi_1,\ldots,\pi_n) = \begin{cases} 1 & \text{if } \pi_1 = \pi_2 = \cdots = \pi_n \\ \frac{1}{m+1} & \text{if } \pi_1 = 1, \pi_i = 2(\forall i \in T\setminus\{1\}) \\ -\frac{m+2}{m+1} & \text{if } \pi_1 = 2, \pi_i = 1(\forall i \in T\setminus\{1\}) \\ 0 & \text{otherwise} \end{cases}$$

To obtain a TMSP, we need to find a CTME first. A CTME (the CTMECom and CTMECor (see Section 2) are the same in this case) is the strategy profile prescribing that the team plays each joint strategy $\pi_1 = \cdots = \pi_{n-1} = a(1 \leq a \leq m)$ with probability $\frac{1}{m}$.[2] Now, by the aforementioned transformed algorithm (Basilico et al. 2017), the corresponding (unique) TMSP prescribes that player $i(\in T)$ plays $\pi_i = a(1 \leq a \leq m)$ with probability $\frac{1}{m}$, while player n is indifferent between playing any pure strategies given this TMSP. Then, $v_c = 0$.

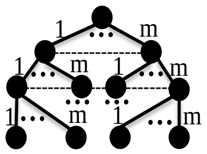

Figure 1: Example

Even though it is difficult to determine a TME analytically, we can find a good lower bound for the TME value through an NE because a TME is an NE maximizing the team's utility. This equilibrium strategy $(1, 2, \ldots, 2, 3)$ prescribes that player 1 plays $\pi_1 = 1$, player $i(2 \leq i \leq n-1)$

[1]PoC is different from the price of uncorrelation ($\frac{v'_c}{v_m}$) (Basilico et al. 2017), i.e., the inefficiency caused by that team players do not correlate their strategies when they can (and then obtain utility v'_c).

[2]First, the team, obtaining utility $\frac{1}{m}$ in this profile, cannot improve its utility by playing other joint strategies with nonzero probability because all other joint strategies are weakly dominated as they provide a utility less than $\frac{1}{m}$ to the team for every adversary strategy. Second, the team must play each joint strategy of the form $\pi_1 = \cdots = \pi_{n-1}$ with nonzero probability because the team will receive utility 0 if any such a form with a' is played with probability 0, but the adversary n plays a'. Therefore, each such a form is played with probability $\frac{1}{m}$ by symmetry.

plays $\pi_i = 2$, while player n plays $\pi_n = 3$. The team's utility is $U_T(1, 2, \ldots, 2, 3) = \frac{1}{m+1}$. Basically, any team member will obtain 0 from unilateral deviation while the adversary is indifferent between playing any pure strategies.

Therefore, PoC $= \frac{v_m}{v_c} \geq \frac{\frac{1}{m+1}}{0} = \infty$. □

5 Associated Recursive Asynchronous MDT

This section proposes novel techniques to directly solve the non-convex Problem (2) to compute TMEs. Specifically, we develop the Associated Recursive Asynchronous Multiparametric Disaggregation Technique (ARAMDT) to approximate multilinear terms in Eq.(2b). ARAMDT is developed based on MDT (Kolodziej, Castro, and Grossmann 2013; Andrade et al. 2019), which uses a digit-wise discretization of one variable in the bilinear term for approximation and then transforms the bilinear program into a mixed-integer linear program (MILP). ARAMDT has three additional important features: 1) the Asynchronous MDT (AMDT) (i.e., adding the asynchronous precision method to MDT) to reduce the number of constraints and new variables for approximating bilinear terms; 2) the Recursive AMDT (RAMDT) (i.e., recursively deploying AMDT) to approximate multilinear terms which cannot be directly approximated by AMDT; and 3) the Associated RAMDT (ARAMDT) (i.e., adding associated constraints to RAMDT) to reduce the feasible solution space of MILP in RAMDT.

5.1 Asynchronous MDT

To transform the bilinear program into MILP while reducing the number of constraints and new variables for approximating bilinear terms, we develop AMDT which can use different precision levels to approximate bilinear terms even they include the same discretized variable. The idea is that: If we do not need high precision levels to approximate some bilinear terms, AMDT can just use low precision levels to do that, which reduces the number of constraints and new variables because this number of constraints and variables increases with the precision levels. Given a bilinear term $w = y_i y_j$ ($y_i, y_j \in [0, 1]$), we approximate y_i by using a binary number with powers (precision levels) $\mathbb{Z}_{y_i} = \{-1, \ldots, -Z_{y_i}\}$:

$$y_i = \textstyle\sum_{z \in \mathbb{Z}_{y_i}} 2^z \lambda_{i,z} + \Delta y_i \tag{3}$$

where $\lambda_{i,z} \in \{0, 1\}$ and $\Delta y_i \in [0, 2^{-Z_{y_i}}]$ is the slack variable. After that, we approximate w by using powers $\mathbb{Z}_w = \{-1, \ldots, -Z_w\}$ with $Z_w \leq Z_{y_i}$:

$$w = \textstyle\sum_{z \in \mathbb{Z}_w} 2^z y_{i,j,z} + \Delta w \tag{4}$$

where $y_{i,j,z} = \lambda_{i,z} y_j$ can be represented by:

$$0 \leq y_{i,j,z} \leq \lambda_{i,z} \quad \forall z \in \mathbb{Z}_w \tag{5a}$$

$$0 \leq y_j - y_{i,j,z} \leq 1 - \lambda_{i,z} \quad \forall z \in \mathbb{Z}_w \tag{5b}$$

$\Delta w = (y_i - \sum_{z \in \mathbb{Z}_w} 2^z \lambda_{i,z}) y_j$ ($0 \leq y_i - \sum_{z \in \mathbb{Z}_w} 2^z \lambda_{i,z} = \Delta y_i + \sum_{-Z_{y_i} \leq z < -Z_w} 2^z \lambda_{i,z} \leq 2^{-Z_w}$) cannot be represented exactly by linear constraints and then is approximated by using the McCormick relaxation (McCormick 1976):

$$0 \leq \Delta w \leq y_i - \textstyle\sum_{z \in \mathbb{Z}_w} 2^z \lambda_{i,z} \tag{6a}$$

$$2^{-Z_w}(y_j - 1) + y_i - \textstyle\sum_{z \in \mathbb{Z}_w} 2^z \lambda_{i,z} \leq \Delta w \leq 2^{-Z_w} y_j \tag{6b}$$

Here, AMDT replaces Δy_i in the standard MDT (Kolodziej, Castro, and Grossmann 2013) with $y_i - \sum_{z \in \mathbb{Z}_w} 2^z \lambda_{i,z}$ for representing Δw because $Z_w \leq Z_{y_i}$. Therefore, AMDT can use different powers to approximate bilinear terms even they include the same discretized variable, which will reduce the number of constraints if we do not need to use more powers to approximate some bilinear terms, e.g., when w is already equal to $y_i y_j$.

5.2 Recursive Asynchronous MDT

EFGs with $n > 3$ will result in multilinear terms in Eq.(2b), where AMDT cannot be applied directly. Therefore, we develop RAMDT, which recursively uses a new variable to replace each bilinear part of the multilinear term in Eq.(2b) until the multilinear term is replaced by a variable (representing a bilinear term), and then the bilinear term is approximated by AMDT. More specifically, for each $\sigma_T \in \Sigma'_T (\subseteq \Sigma_T)$ reaching a terminal node with the multilinear term $\prod_{i \in T} r_i(\sigma_T(i))$ in Eq.(2b), we recursively define a set of bilinear terms $BL_{\sigma_T} = \{w_1(\sigma_{T_1}), \ldots, w_{n-2}(\sigma_{T_{n-2}})\}$ such that:

$$w_i(\sigma_{T_i}) = r_i(\sigma_{T_i}(i)) w_{i+1}(\sigma_{T_{i+1}}) = \textstyle\prod_{i \leq j \leq n-1} r_j(\sigma_{T_i}(j))$$

where $w_i(\sigma_{T_i}) \in [0, 1] (1 \leq i \leq n-2)$, σ_{T_i} represents the joint sequence of players $i, \ldots, n-1$ as part of σ_T with $\sigma_{T_1} = \sigma_T$, $\sigma_{T_i}(j)$(i.e., $\sigma_T(j)$) is the sequence of player j in σ_{T_i}, and $w_{n-1}(\sigma_{T_{n-1}}) = r_{n-1}(\sigma_{T_{n-1}}(n-1))$. Therefore, each multilinear term can be represented by a bilinear term. To be consistent, we discretize the variable on the left hand side of each bilinear term in BL_{σ_T}.

Denote $BL = \bigcup_{\sigma_T \in \Sigma'_T} BL_{\sigma_T}$. By using $w_1(\sigma_T)$ to replace $\prod_{i \in T} r_i(\sigma_T(i))$ in Eq.(2b) through RAMDT, an upper bound of Problem (2) is computed by:

$$\max_{r_1, \ldots, r_{n-1}} v(\mathcal{I}_n(\emptyset))_{RAMDT} \tag{7a}$$

$$v(\mathcal{I}_n(\sigma_n)) - \textstyle\sum_{I_{n,j} \in I_n : \mathrm{seq}_n(I_{n,j}) = \sigma_n} v(I_{n,j}) \leq \sum_{\sigma_T \in \Sigma_T} U_T(\sigma_T, \sigma_n) w_1(\sigma_T) \quad \forall \sigma_n \in \Sigma_n \tag{7b}$$

$$\text{Eqs.}(1a) - (1c) \quad \forall i \in T \tag{7c}$$

$$\text{Eq.}(3) \quad \forall r_i(\sigma_T(i)), i \in T \backslash \{n-1\}, \sigma_T \in \Sigma'_T \tag{7d}$$

$$\text{Eqs.}(4) - (6b) \quad \forall w_i(\sigma_{T_i}) \in BL \tag{7e}$$

Theorem 2. *The solution of Problem (7) yields an upper bound for Problem (2), i.e., $v(\mathcal{I}_n(\emptyset))_{RAMDT} \geq v(\mathcal{I}_n(\emptyset))$.*

Proof. The variable $w_i(\sigma_{T_i})$ in Problem (7) may not be exactly equal to $r_i(\sigma_{T_i}(i)) w_{i+1}(\sigma_{T_{i+1}})$ due to the power $Z_{w_i(\sigma_{T_i})}$. In fact, by Eqs.(3)–(6b), $|w_i(\sigma_{T_i}) - r_i(\sigma_{T_i}(i)) w_{i+1}(\sigma_{T_{i+1}})| \leq 2^{-Z_w} w_{i+1}(\sigma_{T_{i+1}})$. It means that, some assignments for $w_i(\sigma_{T_i})$, $r_i(\sigma_{T_i}(i))$, and $w_{i+1}(\sigma_{T_{i+1}})$ without satisfying $w_i(\sigma_{T_i}) = r_i(\sigma_{T_i}(i)) w_{i+1}(\sigma_{T_{i+1}})$ are feasible in Problem (7). Thus, Problem (7) is a relaxation of Problem (2). Therefore, the solution of Problem (7) yields an upper bound for Problem (2), i.e., $v(\mathcal{I}_n(\emptyset))_{RAMDT} \geq v(\mathcal{I}_n(\emptyset))$. □

Theorem 3. *As $Z_{w_i(\sigma_{T_i})}$ ($\forall w_i(\sigma_{T_i}) \in BL$) in Problem (7) approaches ∞, $v(\mathcal{I}_n(\emptyset))_{RAMDT}$ approaches $v(\mathcal{I}_n(\emptyset))$.*

Proof. As $Z_{w_i(\sigma_{T_i})}$ approaches ∞, we have: $\lim_{Z_{w_i(\sigma_{T_i})}\to\infty} |w_i(\sigma_{T_i}) - r_i(\sigma_{T_i}(i))w_{i+1}(\sigma_{T_{i+1}})| \leq \lim_{Z_{w_i(\sigma_{T_i})}\to\infty} 2^{-Z_{w_i(\sigma_{T_i})}} w_{i+1}(\sigma_{T_{i+1}}) = 0$, which means that $w_i(\sigma_{T_i})$ approaches $r_i(\sigma_{T_i}(i))w_{i+1}(\sigma_{T_{i+1}})$. Then, $v(\mathcal{I}_n(\emptyset))_{RAMDT}$ approaches $v(\mathcal{I}_n(\emptyset))$. □

The above property shows that $v(\mathcal{I}_n(\emptyset))_{RAMDT} = v(\mathcal{I}_n(\emptyset))$ when $Z_{w_i(\sigma_{T_i})}$ ($\forall w_i(\sigma_{T_i}) \in BL$) is large enough.

5.3 Associated Recursive Asynchronous MDT

By Theorem 2, the solution of Problem (7) is an upper bound for Problem (2) because $w_i(\sigma_{T_i})$ may not be equal to $r_i(\sigma_{T_i}(i))w_{i+1}(\sigma_{T_{i+1}})$, which also makes the feasible solution space very large and then makes it difficult to solve Problem (7). Here we aim to reduce the feasible solution space in Problem (7) and then solve it efficiently. To do that, we develop ARAMDT, which adds associated constraints to RAMDT by exploiting the relation between bilinear terms.

The relation between bilinear terms is based on the constraints for the realization plan in Eqs.(1a)-(1c). Basically, for all sequences with the last action in the same information set, the sum of probabilities to play these sequences is equal to the probability to play the sequence reaching that information set. For example, in Example 1, for bilinear terms $w_1(\sigma_1, \textit{J:/c:c})$, $w_1(\sigma_1, \textit{J:/ccrc:c})$, and $w_1(\sigma_1, \textit{J:/ccrc:f})$, we have $w_1(\sigma_1, \textit{J:/c:c}) = w_1(\sigma_1, \textit{J:/ccrc:c}) + w_1(\sigma_1, \textit{J:/ccrc:f})$. In addition, given $w_1(\sigma_1, \textit{J:/c:r})$, we have $w_1(\sigma_1, \textit{J:/c:c}) + w_1(\sigma_1, \textit{J:/c:r}) = r_1(\sigma_1)r_2(\emptyset) = r_1(\sigma_1)$. That is, we look for the equivalence relation between bilinear terms until these terms are connected to the played probability of a sequence, which is represented by associated constraints. After adding these constraints, the solutions which cannot satisfy these constraints will be ruled out immediately, and then the feasible solution space in Problem (7) is smaller.

Algorithm 1 generates associated constraints for the equivalence relation between bilinear terms. This relation is based on the realization plan, which is also based on information sets of the game tree (see Eq.(1b)). Therefore, for a set of bilinear terms, to check whether one associated constraint of a player based on Eq.(1b) can be created, we can fix other players' sequences, and then check whether this player has the sequences in this set of bilinear terms involving all actions in the related information set. To do that, we create subsets of bilinear terms $BL_{\sigma_{T_i \setminus \{j\}}, \mathcal{I}_j(\sigma_{T_i}(j))}$ for each player j in Lines 1–2, which are indexed by all other involved players' sequences $\sigma_{T_i \setminus \{j\}}$ and player j's information set where the last action of his sequence $\sigma_{T_i}(j)$ is taken. Now we can add associated constraints based on these subsets starting from $i = n-2$ (Line 6). For example, in Example 1, if we have $BL_{\sigma_1, \textit{J:/ccrc:}} = \{w_1(\sigma_1, \textit{J:/ccrc:c}), w_1(\sigma_1, \textit{J:/ccrc:f})\}$, where $|BL_{\sigma_1, \textit{J:/ccrc:}}| = 2$ is the number of actions in player 2's information set *J:/ccrc:*, then there exists an associated constraint $w_1(\sigma_1, \textit{J:/c:c}) = w_1(\sigma_1, \textit{J:/ccrc:c}) + w_1(\sigma_1, \textit{J:/ccrc:f})$ (we can generate an auxiliary variable $w_1(\sigma_1, \textit{J:/c:c})$ if it does not exist) among these bilinear terms. That is, if player j's sequence $\sigma_{T_i}(j)$ is on the left hand side of bilinear terms in $BL_{\sigma_{T_i \setminus \{j\}}, \mathcal{I}_j(\sigma_j)}$ whose number of terms (corresponding to the number of sequences of player j) is equal to the number of actions in $\mathcal{I}_j(\sigma_j)$ (Line 8), or $BL_{\sigma_{T_i \setminus \{i\}}} (= \{w_{i+1}(\sigma_{T_i \setminus \{i\}}) \mid w_i(\sigma_{T_i}) \in BL_{\sigma_{T_i \setminus \{j\}}, \mathcal{I}_j(\sigma_j)}\}$ representing the set of terms about the right hand side of each bilinear term $w_i(\sigma_{T_i})$) is in the domain of the relation mapping RM (Line 10, initialized by $RM[\{w_{n-1}(\sigma_{n-1}a) \mid a \in \chi(I_{n-1,j})\}] = w_{n-1}(\sigma_{n-1})$ based on Eq.(1b) and updated at Line 24), we can add an associated constraint for the equivalence relation (Line 23) between this subset of bilinear terms and the bilinear term involving the sequence reaching this information set (Line 9 or 11). If this term is new, the related subsets and the set of bilinear terms are updated (Lines 15–16). At Lines 17–22, this new bilinear term is connected to one variable if the other variable in this term is certainly 1. The loop (Lines 4–25) will terminate if no new associated constraints are found.

Algorithm 1: Generate associated constraints

```
1  for $w_i(\sigma_{T_i}) \in BL$, and $j \leftarrow i, \ldots, n-1$ do
2      $BL_{\sigma_{T_i \setminus \{j\}}, \mathcal{I}_j(\sigma_{T_i}(j))} \leftarrow BL_{\sigma_{T_i \setminus \{j\}}, \mathcal{I}_j(\sigma_{T_i}(j))} \cup \{w_i(\sigma_{T_i})\}$;
3  isContinue $\leftarrow$ true, $BL' \leftarrow BL$;
4  while isContinue = true do
5      isContinue $\leftarrow$ false;
6      for $i \leftarrow n-2, \ldots, 1$; $j \in \{i, \ldots, n-1\}$; each $BL_{\sigma_{T_i \setminus \{j\}}, \mathcal{I}_j(\sigma_{T_i}(j))}$ do
7          $w_i(\sigma'_{T_i}) \leftarrow \varnothing, \sigma_j \leftarrow \sigma_{T_i}(j)$;
8          if $i = j \wedge |BL_{\sigma_{T_i \setminus \{j\}}, \mathcal{I}_j(\sigma_j)}| = |\chi(\mathcal{I}_j(\sigma_j))|$ then
9              $w_i(\sigma'_{T_i}) \leftarrow w_i(seq_j(\mathcal{I}_j(\sigma_j)), \sigma_{T_i \setminus \{j\}})$;
10         if $i \neq j \wedge BL_{\sigma_{T_i \setminus \{i\}}}$ in $RM$ then
11             $w_i(\sigma'_{T_i}) \leftarrow r_i(\sigma_i) RM[BL_{\sigma_{T_i \setminus \{i\}}}]$;
12         if $w_i(\sigma'_{T_i}) \neq \varnothing$ then
13             isContinue $\leftarrow$ true;
14             if $w_i(\sigma'_{T_i}) \notin BL'$ then
15                 Line 2, $j \leftarrow i, \ldots, n-1$;
16                 $BL' \leftarrow BL' \cup \{w_i(\sigma'_{T_i})\}$;
17                 if $r_i(\sigma'_{T_i}(i)) = 1$ then
18                     Add $w_i(\sigma'_{T_i}) = w_{i+1}(\sigma'_{T_i \setminus \{i\}})$;
19                 if $w_{i+1}(\sigma'_{T_i \setminus \{i\}}) = 1$ then
20                     Add $w_i(\sigma'_{T_i}) = r_i(\sigma'_{T_i}(i))$;
21                 if $r_i(\sigma'_{T_i}(i)) = w_{i+1}(\sigma'_{T_i \setminus \{i\}}) = 1$ then
22                     Add $w_i(\sigma'_{T_i}) = 1$;
23             Add $w_i(\sigma'_{T_i}) = \sum_{w'_i \in BL_{\sigma_{T_i \setminus \{j\}}, \mathcal{I}_j(\sigma_{T_i}(j))}} w'_i$;
24             $RM[BL_{\sigma_{T_i \setminus \{j\}}, \mathcal{I}_j(\sigma_{T_i}(j))}] \leftarrow w_i(\sigma'_{T_i})$;
25         Remove $BL_{\sigma_{T_i \setminus \{j\}}, \mathcal{I}_j(\sigma_j)}$;
```

ARAMDT, remaining to be the upper bound of Problem (2), adds associated constraints for the equivalence relation between bilinear terms to RAMDT (Problem (7)), i.e.,

$$\max_{r_1,\ldots,r_{n-1}} v(\mathcal{I}_n(\emptyset))_{ARAMDT} \tag{8a}$$

$$\text{Eqs.}(7b) - (7e) \tag{8b}$$

$$\text{Constraints generated by Algorithm 1} \tag{8c}$$

Algorithm 2: Iterative ARAMDT (IARAMDT)

```
1  Initialize $\epsilon_0, \epsilon_1, \epsilon_2, Z_w = Z_r = Z = 0, \forall Z_w, Z_r$;
2  repeat
3      $(r_T, \overline{v}) \leftarrow$ Problem (8) with gap $\epsilon_1$;
4      $\underline{v} \leftarrow$ BR, given $r_T$;
5      $Z \leftarrow Z + 1, BL_I \leftarrow \emptyset$;
6      while $BL_I = \emptyset \wedge |\overline{v} - \underline{v}| > \epsilon_0$ do
7          for each $w_i(\sigma_{T_i}) \in BL$ do
8              if $|w_i(\sigma_{T_i}) - r_i(\sigma_i) w_{i+1}(\sigma_{T_{i+1}})| > \epsilon_2$ then
9                  $BL_I \leftarrow BL_I \cup \{w_i(\sigma_{T_i})\}$;
10                 $Z_{w_i(\sigma_{T_i})} \leftarrow Z$ if $Z_{w_i(\sigma_{T_i})} < \lceil -\log_2 \epsilon_2 \rceil$;
11                 $Z_{r_i(\sigma_i)} \leftarrow Z$ if $Z_{r_i(\sigma_i)} < \lceil -\log_2 \epsilon_2 \rceil$;
12         if $BL_I = \emptyset$ then $\epsilon_2 \leftarrow \frac{\epsilon_2}{10}$;
13     Generate a constraint: $v(\mathcal{I}_n(\emptyset))_{ARAMDT} \geq \underline{v}$;
14 until $|\overline{v} - \underline{v}| \leq \epsilon_0$;
15 return $r_T$
```

Theorem 4. *The feasible solution of Problem (2) is feasible in Problem (8).*

Proof. Suppose the realization plan r_i of player i is the feasible solution of Problem (2) but is infeasible in Problem (8). By Theorem 2, r_i is feasible in Problem (7). Therefore, r_i does not satisfy the constraints generated by Algorithm 1. Given $Z_{w_i(\sigma_{T_i})}$ ($\forall w_i(\sigma_{T_i}) \in BL$) in Problem (8), we have $|w_i(\sigma_{T_i}) - r_i(\sigma_{T_i}(i)) w_{i+1}(\sigma_{T_{i+1}})| \leq 2^{-Z_{w_i(\sigma_{T_i})}} w_{i+1}(\sigma_{T_{i+1}})$. Now we consider the feasible solutions when $w_i(\sigma_{T_i}) = r_i(\sigma_{T_i}(i)) w_{i+1}(\sigma_{T_{i+1}})$ for all bilinear terms. Without loss of generality, we assume $\sum_{a \in \chi(I_{i,j})} r_i(\sigma_{T_i}(i)a) = r_i(\sigma_{T_i}(i))$, but $\sum_{a \in \chi(I_{i,j})} w_i(\sigma_{T_i}(i)a, \sigma_{T_i \setminus \{i\}}) \neq w_i(\sigma_{T_i})$. We have:

$$\sum_{a \in \chi(I_{i,j})} w_i(\sigma_{T_i}(i)a, \sigma_{T_i \setminus \{i\}}) \neq w_i(\sigma_{T_i})$$
$$\Rightarrow \sum_{a \in \chi(I_{i,j})} r_i(\sigma_{T_i}(i)a) w_{i+1}(\sigma_{T_i \setminus \{i\}}) \neq r_i(\sigma_{T_i}(i)) w_{i+1}(\sigma_{T_i \setminus \{i\}})$$
$$\Rightarrow \sum_{a \in \chi(I_{i,j})} r_i(\sigma_{T_i}(i)a) \neq r_i(\sigma_{T_i}(i))$$

which causes a contradiction. Therefore, the feasible solution of Problem (2) is feasible in Problem (8). □

Theorem 5. *$v(\mathcal{I}_n(\emptyset))_{ARAMDT} \geq v(\mathcal{I}_n(\emptyset))$, and $v(\mathcal{I}_n(\emptyset))_{ARAMDT}$ approaches $v(\mathcal{I}_n(\emptyset))$ as $Z_{w_i(\sigma_{T_i})}$ ($\forall w_i(\sigma_{T_i}) \in BL$) approaches ∞ in Problem (8).*

Proof. This is immediately obtained from Theorems 2–4 □

6 An Iterative Algorithm for Computing TMEs Based on ARAMDT

Even though ARAMDT can compute TMEs, it is difficult to decide which precision levels for different bilinear terms could achieve the given accuracy (i.e., ϵ in 'ϵ-TME'). This section develops a novel iterative algorithm to efficiently compute TMEs within any given accuracy based on ARAMDT. Specifically, we iteratively increase the precision levels for part of bilinear terms until the desired accuracy is achieved, as shown in Algorithm 2. This is based on the fact that the upper bound of the TME value in Problem (8) will approach the optimal value when the powers for approximating bilinear terms in ARAMDT increase by Theorem 5. Moreover, given the joint realization plan r_T of the team (i.e., $\times_{i \in T} r_i$) corresponding to the upper bound, we can obtain the lower bound of the optimal value by computing the Best Response (BR) of the adversary. Therefore, we can iteratively increase powers for approximating bilinear terms to make the lower bound and the upper bound tighter to guarantee the accuracy.

Here, the upper bound of the TME value can be computed by solving Problem (8), while the lower bound can be obtained through the BR of the adversary, e.g., by using the public-state algorithm (Johanson et al. 2011). For convenience, we use $Z = 0$ to represent the McCormick relaxation (McCormick 1976), i.e., approximating a bilinear term $w = y_i y_j$ ($y_i, y_j \in [0,1]$) with loose bounds: $\max\{0, y_i + y_j - 1\} \leq w \leq \min\{y_i, y_j\}$. Line 1 initializes the powers to 0, which means that each bilinear term is approximated by the McCormick relaxation, i.e., constraints of AMDT in Problem (8) are replaced by constraints of the McCormick relaxation. In each iteration, the lower and upper bounds are computed at Lines 3 and 4, respectively. Here, the gap ϵ_1 is set for MILP, i.e., Problem (8). After that, the powers are updated at Lines 5–12. Specifically, we increase the powers for bilinear terms (in BL_I) whose difference values between $w_i(\sigma_{T_i})$ and $r_i(\sigma_i) w_{i+1}(\sigma_{T_{i+1}})$ (here $\sigma_i = \sigma_{T_i}(i)$) are larger than ϵ_2 (Lines 8 and 9). The powers for bilinear terms and the variables are updated separately with the bound $\lceil -\log_2 \epsilon_2 \rceil$ (Lines 10 and 11) because $|w_i(\sigma_{T_i}) - r_i(\sigma_i) w_{i+1}(\sigma_{T_{i+1}})| \leq 2^{-Z_{w_i(\sigma_{T_i})}} w_{i+1}(\sigma_{T_{i+1}}) \leq 2^{-Z_{w_i(\sigma_{T_i})}}$. If no bilinear terms fulfill the requirements, ϵ_2 will decrease at Line 12 to guarantee that we can achieve the given accuracy. The lower bound of the objective value of Problem (8) is updated at Line 13. The loop will terminate if the gap between the lower bound and the upper bound is not larger than ϵ_0 (Line 14) and then the joint realization plan of team players are returned (Line 15). The following theorem shows that the output r_T is part of an $(\epsilon_0 + \epsilon_1)$-TME.

Theorem 6. *The output r_T of Algorithm 2 is part of an $(\epsilon_0 + \epsilon_1)$-TME.*

Proof. When we obtain the output r_T of Algorithm 2, we also obtain the optimal value $\overline{v}$ of Problem (8). Given the constraints in Problem (8) and constraints in Eqs.(1a)-(1c) for the adversary, $\overline{v}$ is also the optimal objective value in Problem: $\max_{r_T} \min_{r_n} \sum_{\sigma_n \in \Sigma_n} \sum_{\sigma_T \in \Sigma_T} U_T(\sigma_T, \sigma_n) w_1(\sigma_T) r_n(\sigma_n)$. Let r_n be the optimal solution in this problem. Suppose $v^\star$ is the team's utility value under profile (r_T, r_n). We know that $\underline{v} \leq v^\star \leq \overline{v} + \epsilon_1$. Consequently, given r_n, for any strategy r'_T by the unilateral deviation from r_T with the team's utility v', we have $v' \leq \overline{v} + \epsilon_1$ and then $v' - v^\star \leq \overline{v} + \epsilon_1 - \underline{v} \leq \epsilon_0 + \epsilon_1$. Given r_T, for any adversary strategy r'_n with the adversary's utility v', we have $v' \leq -\underline{v}$ and then $v' - (-v^\star) \leq$

ϵ	10^{-1}	$\frac{10^{-1}}{2}$	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-1}	$\frac{10^{-1}}{2}$	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
IARAMDT	<1s	<1s	<1s	<1s	<1s	<1s	<1s	<1s	<1s	<1s	19s	319s	**649s**	**2460s**
IARMDT-1	<1s	<1s	<1s	<1s	<1s	<1s	<1s	<1s	<1s	<1s	>2h			
IARMDT+1	<1s	<1s	<1s	<1s	<1s	<1s	<1s	<1s	<1s	<1s	41	492s	964s	4786s
IRAMDT	4s	47s	>2h					52s	>2h					
IARMDT	<1s	<1s	<1s	<1s	<1s	<1s	<1s	<1s	<1s	<1s	7s	252	714s	4425s
IRMDT	4s	172s	>2h					61s	>2h					
BARON	30s	690s	>10h					30s	>10h					
IARAMDT	<1s	<1s	1s	4s	54s	**133s**	**522s**	<1s	<1s	<1s	35s	141s	**399s**	**3242s**
IARMDT-1	<1s	<1s	<1s	4s	14s	>2h		<1s	<1s	2s	3082s	5982s	>2h	
IARMDT+1	<1s	<1s	1s	4s	50s	707s	707s	<1s	<1s	1s	19s	100s	1885s	>2h
IRAMDT	319s	>2h						>2h						
IARMDT	<1s	<1s	2s	5s	65s	>2h		<1s	<1s	1s	39s	320s	919s	>2h
IRMDT	436s	>2h						>2h						
BARON	60s	>10h						540s	>10h					

Table 1: The top left part is 3K4 with $|BL| = 120$, the top right part is 3K5 with $|BL| = 200$, the bottom left part is 3K6 with $|BL| = 300$, the bottom right part is 3K7 with $|BL| = 420$, $\Delta_U = 6$, and '$>$ #h': algorithms are terminated after # hours.

$-\underline{v}+\overline{v}+\epsilon_1 \leq \epsilon_0+\epsilon_1$. Then r_T is part of $(\epsilon_0+\epsilon_1)$ Nash equilibrium. In addition, $\overline{v}+\epsilon_1$ is the upper bound of the TME value $v^{\star\star}$ and then $v^{\star\star} - v^\star \leq \overline{v}+\epsilon_1 - v^\star \leq \overline{v}+\epsilon_1 - \underline{v} \leq \epsilon_0+\epsilon$. Therefore, r_T is part of an $(\epsilon_0+\epsilon_1)$-TME. □

7 Experimental Evaluation

We evaluate our algorithm (IARAMDT) through experiments. We use CPLEX (version 12.9) to solve the linear program. All the algorithms are performed on a machine with 6-core 3.2GHz CPU and 16GB memory. Because the default gap for MILP in CPLEX is 10^{-6}, we set $\epsilon_1 = 10^{-6}$ with $\epsilon_2 = 5 \times 10^{-7}$ unless otherwise specified.

We use BARON (version 19.3.24) (Khajavirad and Sahinidis 2018) to solve Problem (2) directly as a baseline. Other baselines include: 1) IARAMDT-1: at each iteration, we only select bilinear terms with the largest difference between $w_i(\sigma_{T_i})$ and $r_i(\sigma_i)w_{i+1}(\sigma_{T_{i+1}})$ similar to the previous approach (Čermák et al. 2018); 2) IARAMDT+1: we increase the power with 1 instead of Z at lines 10 and 11 of Algorithm 2, which is similar to many approaches (Čermák et al. 2018; Andrade et al. 2019); 3) IRAMDT: we use RAMDT instead of ARAMDT; 4) IARMDT: we use MDT to replace AMDT; and 5) IRMDT: we increase the powers with 1 for all bilinear terms starting with $Z = 1$, which is the original MDT approach (Kolodziej, Castro, and Grossmann 2013; Wang, Guo, and An 2017).

Our experiments run on the standard games, the multi-player Kuhn poker and the Leduc Hold'em poker games (see (Farina et al. 2018) for their rules), where nKr and nLr denote an n-player Kuhn instance with r ranks (i.e., r cards) and an n-player Leduc instance with r ranks (i.e., $2r$ cards), respectively. Δ_U is the difference between the maximum possible utility and the minimum possible utility of the team. In addition to the current game rules, the game ends if the adversary takes the action of folding because the team will certainly win after that. Without loss of generality, player n is the adversary.

We show the time to compute $\epsilon\Delta_U$-TMEs for different accuracies. Here $\epsilon\Delta_U$ is $\epsilon_0 + \epsilon_1$ in Algorithm 2 or the

ϵ	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-1}	10^{-2}	10^{-3}	$\frac{10^{-3}}{2}$
1	2s	**44s**	**481s**	**7047s**	3s	218s	**1831s**	**5591s**
2	2s	>3h			3s	>3h		
3	2s	274s	1303s	>3h	3s	155s	5121s	>3h
4	2s	83s	4442s	>3h	3s	209s	6924s	6924s

Table 2: The left part is 4K5 with $|BL| = 1840$ and $\Delta_U = 8$, the right part is 3L3 with $|BL| = 1380$ and $\Delta_U = 21$ ($\epsilon_1 = \epsilon_2 = 0.001$), and rows 1–4 are IARAMDT, IARAMDT-1, IARAMDT+1, IARMDT, respectively, which are terminated after 3 hours.

difference between the lower bound and the upper bound in BARON. Results in Table 1 show that 1) IARAMDT dramatically outperforms the exiting MDT approach and BARON; 2) our approximation techniques, especially the associated constraint method, are extremely important for the scalability of IARAMDT by comparing it with IRAMDT and IARMDT; and 3) our novel techniques in our iterative algorithm improve the scalability of IARAMDT by comparing it with IARAMDT-1 and IARAMDT+1. Specifically, after using our associated constraint method with the same initial setting (i.e., using the McCormick relaxation) in the iterative algorithm, IARAMDT-1, IARAMDT+1, and IARMDT perform similarly to IARAMDT when we only need low precision levels to achieve the given accuracy, but they perform significantly worse than IARAMDT when we need high precision levels to achieve the given accuracy, which is further confirmed in larger problems as shown in Table 2.

Comparison to TMSP. Table 3 shows the comparison to the TMSP obtained from an $\epsilon\Delta_U$-CTMECor, which is computed by algorithm Hybrid Column Generation (HCG) (Celli and Gatti 2018) or algorithm Fictitious Team-Play (FTP) (Farina et al. 2018). The gap is the relative distance between the team's best utility (v_m) of IARAMDT (the lower bound when $\epsilon = 10^{-6}$) and the team's lower bound (v) under different values of ϵ, i.e., $\frac{|v-v_m|}{|v|} \times 100\%$. The larger the gap is, the more utility the team will lose. We can see that: 1) the gaps of IARAMDT monotonically de-

ϵ	10^{-1}	10^{-2}	10^{-3}	10^{-6}	10^{-1}	10^{-2}	10^{-3}	10^{-6}
1	**74**%	**0**%	**0**%	**0**%	**60**%	**60**%	**10**%	**0**%
5	<1s	3s	6s	6s	5s	38s	169s	169s
5	*75%*	*70%*	*80%*	*80%*	*76%*	*80%*	*84%*	*84%*
6	<1s	4s	>2h		<1s	10s	>2h	
6	*87%*	*69%*			*91%*	*69%*		
1	*86%*	**20**%	**17**%	**0**%	**77**%	**67**%	**12**%	**0**%
5	24s	201s	1305s	2208s	59s	1154s	>2h	
5	*84%*	*86%*	*84%*	*82%*	*83%*	*82%*		
6	<1s	7s	>2h		<1s	23s	>2h	
6	*88%*	*76%*			*93%*	*77%*		

Table 3: The gaps and runtime of IARAMDT (label 1) (see Table 1 for its runtime), HCG (label 5), and FTP (label 6) on 3K4, 3K5, 3K6, and 3K7, respectively.

crease with ϵ to 0, but the gaps of the TMSP may not decrease with ϵ and are always very large; and 2) given the same ϵ, IARAMDT runs faster than HCG and FTP in most cases. About the TMSP obtained from a CTMECom, it can be computed within 1s in four games of Table 3, but the gaps (i.e., 88%, 94%, 95%, and 92%) are extremely large. Therefore, the TMSP can cause a huge loss, which is consistent with our theoretical result shown in Theorem 1.

8 Conclusions

This paper proposes an efficient algorithm to compute TMEs within any given accuracy for zero-sum multiplayer EFGs. We first show that the inefficiency of correlated strategies can be arbitrarily large in EFGs. To efficiently solve the non-convex program for finding TMEs directly, we develop novel approximation techniques (i.e., an asynchronous precision method and an associated constraint method) and the novel iterative algorithm. Our algorithm is dramatically faster than baselines in the experimental evaluation. In addition, our novel techniques could shed light on the study of imperfect-recall games involving bilinear terms (Čermák et al. 2018) and global optimization.

Acknowledgments

This research was supported by NSOE-TSS2019-01, AISG-RP-2019-0013. This research was conducted in collaboration with Singapore Telecommunications Limited and partially supported by the Singapore Government through the Industry Alignment Fund - Industry Collaboration Projects Grant.

References

Andrade, T.; Oliveira, F.; Hamacher, S.; and Eberhard, A. 2019. Enhancing the normalized multiparametric disaggregation technique for mixed-integer quadratic programming. *Journal of Global Optimization* 73(4):701–722.

Basilico, N.; Celli, A.; De Nittis, G.; and Gatti, N. 2017. Team-maxmin equilibrium: Efficiency bounds and algorithms. In *AAAI*, 356–362.

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.

Castro, P. M. 2016. Normalized multiparametric disaggregation: An efficient relaxation for mixed-integer bilinear problems. *Journal of Global Optimization* 64(4):765–784.

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

Čermák, J.; Bošanskỳ, B.; Horák, K.; Lisỳ, V.; and Pěchouček, M. 2018. Approximating maxmin strategies in imperfect recall games using A-loss recall property. *International Journal of Approximate Reasoning* 93:290–326.

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.

Jiang, A. X.; Procaccia, A. D.; Qian, Y.; Shah, N.; and Tambe, M. 2013. Defender (mis)coordination in security games. In *IJCAI*, 220–226.

Johanson, M.; Waugh, K.; Bowling, M.; and Zinkevich, M. 2011. Accelerating best response calculation in large extensive games. In *IJCAI*, 258–265.

Khajavirad, A., and Sahinidis, N. V. 2018. A hybrid LP/NLP paradigm for global optimization relaxations. *Mathematical Programming Computation* 10(3):383–421.

Kolodziej, S.; Castro, P. M.; and Grossmann, I. E. 2013. Global optimization of bilinear programs with a multiparametric disaggregation technique. *Journal of Global Optimization* 57(4):1039–1063.

McCormick, G. P. 1976. Computability of global solutions to factorable nonconvex programs: Part I–Convex underestimating problems. *Mathematical Programming* 10(1):147–175.

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.

Ryoo, H. S., and Sahinidis, N. V. 2001. Analysis of bounds for multilinear functions. *Journal of Global Optimization* 19(4):403–424.

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.

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

Wang, X.; Guo, Q.; and An, B. 2017. Stop nuclear smuggling through efficient container inspection. In *AAMAS*, 669–677.