Converging to Team-Maxmin Equilibria in Zero-Sum Multiplayer Games

Youzhi Zhang [1] **Bo An** [1]

Abstract

Efficiently computing equilibria for multiplayer games is still an open challenge in computational game theory. This paper focuses on computing Team-Maxmin Equilibria (TMEs), which is an important solution concept for zero-sum multiplayer games where players in a team having the same utility function play against an adversary independently. Existing algorithms are inefficient to compute TMEs in large games, especially when the strategy space is too large to be represented due to limited memory. In two-player games, the Incremental Strategy Generation (ISG) algorithm is an efficient approach to avoid enumerating all pure strategies. However, the study of ISG for computing TMEs is completely unexplored. To fill this gap, we first study the properties of ISG for multiplayer games, showing that ISG converges to a Nash Equilibrium (NE) but may not converge to a TME. Second, we design an ISG variant for TMEs (ISGT) by exploiting that a TME is an NE maximizing the team's utility and show that ISGT converges to a TME and the impossibility of relaxing conditions in ISGT. Third, to further improve the scalability, we design an ISGT variant (CISGT) by using the strategy space for computing an equilibrium that is close to a TME but is easier to be computed as the initial strategy space of ISGT. Finally, extensive experimental results show that CISGT is orders of magnitude faster than ISGT and the state-of-the-art algorithm to compute TMEs in large games.

1. Introduction

Game theory is an important tool to model the interaction between agents. Now researchers have achieved many results for two-player games, e.g., computing Nash Equilibria (NEs) for zero-sum games (Nash, 1951) via linear programs (Von Neumann & Morgenstern, 1953; Von Stengel, 1996; Shoham & Leyton-Brown, 2008) and many scalable algorithms, e.g., the double oracle algorithm (McMahan et al., 2003) and the counterfactual regret minimization algorithm (Zinkevich et al., 2008), and computing Stackelberg equilibria (Conitzer & Sandholm, 2006). Based on these results, researchers have successfully applied game theory to many domains, e.g., improving the security for people and wildlife in security games (Sinha et al., 2018) and defeating top human professionals in poker games (Brown & Sandholm, 2018). However, researchers have achieved fewer results for multiplayer games except for games having very special structures, e.g., polymatrix games (Cai & Daskalakis, 2011) and congestion games (Shoham & Leyton-Brown, 2008) or algorithms having no theoretical guarantee (Brown & Sandholm, 2019). In fact, the hardness to compute NEs (it is PPAD-complete even for zero-sum three-player games (Chen & Deng, 2005)) and the equilibrium selection problem (Brown & Sandholm, 2019) (it is hard for players independently choosing strategies and then forming an NE because NEs are not exchangeable) make them remain open challenges for computing and applying NEs in multiplayer games.

This paper focuses on computing Team-Maxmin Equilibria (TMEs) (von Stengel & Koller, 1997; Basilico et al., 2017b; Celli & Gatti, 2018; Zhang & An, 2020), which is an important solution concept for zero-sum multiplayer games where players in a team having the same utility function play against an adversary independently. A TME is an NE maximizing the team's utility and always exists. More importantly, the TME is unique in general, and then it avoids the equilibrium selection problem. Moreover, TMEs can be used to model many real-world scenarios. For example, to keep the safety of New York, the Patrol Services Bureau in the New York City Police Department (NYPD) has 77 police precincts (NYPD, 2020b), and each precinct is divided into four or five fully-staffed sectors by Neighborhood Policing recently (NYPD, 2020a). They maintain "sector integrity": officers in different sectors work independently to keep the safety of their sectors, except in precinct-wide emergencies (NYPD, 2020a).

However, it is still challenging to compute a TME, which is FNP-hard (Hansen et al., 2008). Moreover, a TME is only

[1]School of Computer Science and Engineering, Nanyang Technological University, Singapore. Correspondence to: Youzhi Zhang <yzhang137@ntu.edu.sg>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

solved via a non-convex program (von Stengel & Koller, 1997) with global optimization techniques (Zhang & An, 2020), which makes it hard to solve large games. In addition, it is impossible to apply this approach when we cannot represent the problem via the game matrix due to the large strategy space. For example, in a network security game on a fully connected network with 190 edges and 20 nodes, the number of possible adversary pure strategies (paths) without any cycle is about 6.6^{18} (Jain et al., 2011), which means that the memory cost for enumerating all pure strategies would be prohibitive. In two-player games, the Incremental Strategy Generation (ISG) algorithm (including the double oracle algorithm, column generation) (McMahan et al., 2003; Jain et al., 2011; Sinha et al., 2018) is an efficient approach to avoid enumerating all pure strategies: It computes an equilibrium in a game with restricted strategy spaces for players, and then iteratively expands players' strategy spaces. ISG can converge to an NE in zero-sum two-player games. However, the study of ISG for computing TMEs is completely unexplored. We know that ISG terminates when oracles in ISG cannot find better strategies than the equilibrium strategies in the restricted game, which is consistent with the definition of NEs. Unfortunately, NEs in multiplayer games, unlike those in two-player games, are not exchangeable and may give different utilities to the team. Therefore, if ISG is used to compute TMEs, even it can converge to an NE, it may cause a loss to the team.

To fill this gap, we first study the properties of ISG for multiplayer games, showing that ISG converges to an NE but may not converge to a TME, and it can cause an arbitrarily large loss to the team. Second, we design an ISG variant (ISGT) by exploiting that a TME is an NE maximizing the team's utility and show that ISGT converges to a TME and the impossibility of relaxing conditions in ISGT. Third, to further improve the scalability, we design an ISGT variant (CISGT) by using the strategy space for computing an equilibrium that is close to a TME but is easier to be computed as the initial strategy space of ISGT. Finally, extensive experimental results show that CISGT is orders of magnitude faster than ISGT and the state-of-the-art algorithm to compute TMEs in large games.

2. Related Work

McMahan et al. (2003) propose the first ISG (also called the double oracle algorithm) for zero-sum two-player games, where the robot chooses a path to a goal location while avoiding being detected by an adversary on the road. Given the adversary strategy, the robot's best response oracle (to compute a best response against the adversary strategy) is modelled as a Markov Decision Process (MDP). Then solving the best response oracle is equivalent to solving the corresponding MDP. After that, ISG is used to solve many similar problems, including the classic network security games (Jain et al., 2011; Iwashita et al., 2016; Zhang et al., 2019) and extensive-form games (Bosansky et al., 2014). The MDP feature of the best response oracle in ISG makes it possible to deploy deep reinforcement learning (Wang et al., 2019) and be extended to multiagent learning (Lanctot et al., 2017; Muller et al., 2020). In this paper, we study the problem of extending ISG for converging to TMEs, which will be a base for developing learning algorithms for TMEs in multiplayer games.

However, the extension is not straightforward. It is well-known that ISG converges to an NE in zero-sum two-player games (McMahan et al., 2003), but, which is unclear (to our best knowledge) in multiplayer games. Then we theoretically show that ISG (Vanilla-ISG) converges to an NE in multiplayer games in Theorem 1. However, as we illustrate in Section 4.1, the existing ISG cannot guarantee to converge to a TME. Then we try to extend it to ISGT for converging to a TME. As we illustrate in Section 4.2, it is difficult to converge to a TME, e.g., we cannot simply add the team's best response to the restricted game to converge to a TME. Then we add our new operations to ISG to guarantee to converge to a TME by exploiting that a TME is an NE maximizing the team's utility. Section 4.2 also implies that the conditions in our operations cannot be further relaxed. Finally, we theoretically show that our new ISG (i.e., ISGT) can guarantee to converge to a TME in Theorem 2–4.

The Correlated TME (CTME) (Basilico et al., 2017b) is a solution concept close to the TME, where team players with the same utility function can synchronize their actions against the adversary. That is, team players can jointly plan and execute their strategies, which means that the team is equivalent to a single player with actions as the joint team action profiles. Then, a CTME can be found through a linear program similar to finding an NE in zero-sum two-player games. However, team players in a TME cannot correlate their actions (Celli & Gatti, 2018), and then cannot directly use the strategies in a CTME. To compute a TME, one approach is that the team can compute a CTME first and then transform the team's correlated strategy into the team's mixed strategy profile, where a transformation algorithm was proposed (Basilico et al., 2017b). However, this transformation cannot theoretically guarantee to obtain a team-maxmin strategy profile (in a TME) for the team and may cause a huge loss for the team, as shown in the previous experiments (Basilico et al., 2017b). In Section 5.1, we study the limitations of computing TMEs based on CTMEs. We first theoretically show that using the strategy transformed from a CTEM may cause an arbitrarily large loss to the team. Second, we show that a TME in a restricted game with the strategies for computing a CTME may not be an NE in the full game, and it may not be a TME in the full game even it is an NE in the full game. These results show that it is not straightforward to compute a TME via computing a CTME. Therefore, we develop our novel operations to improve the scalability of our algorithm ISGT by computing a CTME to initialize the strategy space and exploiting the team's utility in a CTME to terminate earlier. Finally, we theoretically show that our new CTME based ISG (e.g., CISGT) can guarantee to converge to a TME in Theorem 5–7.

3. Preliminaries

A normal-form game G (Shoham & Leyton-Brown, 2008) is a tuple (N, A, u) where: $N = \{1, \ldots, n\}$ is a set of players, $A = \times_{i\in N} A_i$ is a set of joint actions with that A_i is a finite set of player i's actions (pure strategies) with $a_i \in A_i$, and $u = (u_1, \ldots, u_n)$ is a set of players' utility functions with that $u_i : A \to \mathbb{R}$ is player i's utility function. $X = \times_{i\in N} X_i$ is the set of mixed strategy profiles with that $X_i = \Delta(A_i)$ is the set of player i's mixed strategy. For each $x_i \in X_i$ and $a_i \in A_i$, $x_i(a_i)$ is the probability that action a_i is played, $\underline{A}_{i,x_i} = \{a_i \mid x_i(a_i) > 0, a_i \in A_i\}$ is the support of x_i. For each $x \in X$, player i's expected utility is $u_i(x) = \sum_{a\in A} u_i(a) \prod_j x_j(a_j)$ and $u_i(a_i, x_{-i}) = \sum_{a_{-i}\in A_{-i}} u_i(a_i, a_{-i}) \prod_{j\in N\setminus\{i\}} x_j(a_j)$. Generally, $-i$ denotes the set of all players except player i. The Nash Equilibrium (NE) is an important solution concept for a game, which is a strategy profile x^* such that, for each player i, x^*_i is a best response to x^*_{-i} (i.e., $x^*_i = BR(x^*_{-i})$ with $u_i(x^*_i, x^*_{-i}) \geq u_i(x_i, x^*_{-i}), \forall x_i \in X_i$).

The Team-Maxmin Equilibrium (TME, and TMEs for the plural equilibria) (von Stengel & Koller, 1997; Basilico et al., 2017b) is a solution concept for zero-sum multiplayer games with that a team of players $T = \{1, \ldots, n-1\}$ with $u_i(a) = u_j(a) (\forall i, j \in T, a \in A)$ and $\sum_{i\in T} u_i(a) = u_T(a) = -u_n(a) (\forall a \in A)$ play against an adversary n, and each team player takes actions independently. We call G with such a scenario G_T. A TME is an NE with the properties that it is unique except for degenerate cases[1] and a best NE for the team. The utility of the team under the TME is called the TME value. In an ϵ-TME, the team and the adversary both cannot gain more than ϵ by the unilateral deviation of players, and the gap between the TME value and the ϵ-TME value is not greater than ϵ. The team-maxmin strategy profile x_T(i.e., $\times_{i\in T} x_i$) in a TME (x_T, x_n) can be computed by the following nonlinear program:

$$\max_{x_1,\ldots,x_{n-1}} U \tag{1a}$$
$$U \leq \sum_{a_T\in A_T} u_T(a_T, a_n) \prod_{i\in T} x_i(a_T(i)) \quad \forall a_n \in A_n \tag{1b}$$
$$\sum_{a_i\in A_i} x_i(a_i) = 1, x_i(a_i) \geq 0 \quad \forall i \in T \tag{1c}$$

[1]The situation of multiple TMEs can only occur in degenerate cases with special entries in the payoff matrix (von Stengel & Koller, 1997) because a TME gives the team the highest utility among all NEs.

where a_T is a joint action of the team (i.e., $\times_{i\in T} a_i$), $a_T(i)$ is the action of player i in a_T, and $A_T = \times_{i\in T} A_i$ is the set of these joint actions. The adversary strategy x_n in a TME (x_T, x_n) is computed by a linear program after x_T is computed (see Appendix A). For simplicity, we say that a TME is computed by solving Problem (1). The Correlated TME (CTME) (Basilico et al., 2017b) captures the situation where team players can synchronize actions in G_T. That is, the team has the set of actions A_T, and the set of mixed strategies ($\overline{x}_T \in \Delta(A_T)$), and $\underline{A}_{i,\overline{x}_T} = \{a_i \mid a_i \in A_i, \exists a_T = (a_i, a_{T\setminus\{i\}}), \overline{x}_T(a_T) > 0\}$. The CTME remains the property of the NE in zero-sum two-player games. For example, CTMEs are exchangeable and a CTME can be computed by a linear program, i.e., the multilinear term $\prod_{i\in T} x_i(a_T(i))$ in Eq.(1b) is replaced by a single variable $\overline{x}_T(a_T)$.

The Incremental Strategy Generation (ISG) algorithm (McMahan et al., 2003; Jain et al., 2011; Bosansky et al., 2014) has shown the advantage for improving the scalability for computing an NE in two-player zero-sum games. The algorithm includes the following steps, repeating until convergence: 1) creating a restricted game G' by limiting the set of actions for each player i, i.e., $A'_i \subseteq A_i$; 2) computing the equilibrium strategy profile x^* in this restricted game G'; and 3) computing a best response a_i against x^*_{-i} in the original unrestricted game G for each player i, and add a_i to A'_i if $u_i(a_i, x^*_{-i}) > u_i(x^*)$. The algorithm terminates when no actions are added to the restricted game for all players (i.e., $u_i(a_i, x^*_{-i}) \leq u_i(x^*), \forall a_i \in A_i, i \in N$). For convenience, we call this algorithm Vanilla-ISG. For CTMEs having the property of NEs in zero-sum two-player games, Vanilla-ISG can converge to a CTME. Without loss of generality, we assume that G'_T with $A' = A'_T \times A'_n$ is a restricted game of G_T with $A'_i \subseteq A_i (\forall i \in N)$, and $x_i(a_i) = 0 (\forall A_i \setminus A'_i)$ if x_i is a strategy in G'_T and is used in G_T.

4. ISG in Multiplayer Games

In this section, we show that Vanilla-ISG converges to an NE in multiplayer games. However, we show that Vanilla-ISG cannot guarantee to converge to a TME. Then, we provide a method (ISGT) to amend it by exploiting the property that a TME is an NE maximizing the team's utility.

4.1. Limitations of Vanilla-ISG

We first show that Vanilla-ISG can converge to an NE but cannot guarantee to converge to a TME (even cannot guarantee to approximate a TME).

Theorem 1. *Vanilla-ISG converges to an NE in G.*

Proof. First, Vanilla-ISG will converge because the action set for each player is finite in G. Second, when Vanilla-

ISG converges to strategy profile x^*, $u_i(a_i, x^*_{-i}) \leq u_i(x^*)$ ($\forall a_i \in A_i, i \in N$), which means that $u_i(x^*_i, x^*_{-i}) \geq u_i(x_i, x^*_{-i})$ ($\forall x_i \in X_i, i \in N$). Therefore, x^* is an NE. □

In multiplayer games, there are many different NEs, which are not exchangeable and give different utilities to players. However, a TME is an NE giving the best utility for the team, which is not considered by Vanilla-ISG. Indeed, Vanilla-ISG may not converge to a TME.

Proposition 1. *Vanilla-ISG may not converge to a TME in G_T.*

Proof. By Theorem 1, Vanilla-ISG converges to an NE in G_T. Now we only need to show that this NE may not be a best NE for the team because a TME is a best NE for the team.

Consider G_T with three players (two teammates), two actions ({1,2}) per player, and the following utility function:

$$u_T(a) = \begin{cases} 10 & \text{if } a = (1,2,2) \text{ or } (2,1,1) \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Suppose that the restricted game G'_T in Vanilla-ISG is initialized with action sets $A'_1 = \{1\}, A'_2 = \{2\}, A'_3 = \{2\}$. In G'_T, the single equilibrium is a pure strategy profile $(1, 2, 2)$. Then, A'_3 is expanded to $\{1, 2\}$ because $u_3(1, 2, 1) = 0 > -10 = u_3(1, 2, 2)$, while A'_1 and A'_2 are not expanded. Now the pure strategy profile $(1, 2, 1)$ is the new equilibrium in the new G'_T with $u_T(1, 2, 1) = 0$. For each player, A'_i will not be expanded because players 1 and 2 will gain 0 from the unilateral deviation, while player 3 will lose utility 10 from the unilateral deviation. That is, pure strategy profile $(1, 2, 1)$ is an NE in G_T, i.e., Vanilla-ISG converges to an NE.

Consider the mixed strategy profile $x = (x_1, x_2, x_3)$ with $x_i = (0.5, 0.5)$, which is an NE with utility 2.5 for the team because each player is indifferent between playing their actions with their utility under x, e.g., $u_T(1, x_{-1}) = u_T(2, x_{-1}) = 2.5$. Therefore, pure strategy profile $(1, 2, 1)$ is not a best NE for the team, which means that pure strategy profile $(1, 2, 1)$ is not a TME, concluding the proof. □

Vanilla-ISG cannot guarantee to converge to a TME and also may not approximate a TME by the following result.

Proposition 2. *Vanilla-ISG can cause an arbitrarily large loss to the team in G_T.*

Proof. Consider G_T with utilities shown in Eq.(2). As shown in the proof for Theorem 1, Vanilla-ISG can converge to the NE $(1, 2, 1)$ (pure strategy profile) with utility 0 for the team while $x = (x_1, x_2, x_3)$ with $x_i = (0.5, 0.5)$ is another NE with utility 2.5 for the team. Therefore, Vanilla-ISG can cause an arbitrarily large loss to the team because $\frac{2.5}{0} = \infty$. □

4.2. The Difficulty of Converging to a TME

Vanilla-ISG cannot guarantee to converge to a TME, so we need to extend the current ISG to converge to a TME. This section shows the difficulty of converging to a TME.

Vanilla-ISG's failure to converge to a TME shows that we cannot simply add each player's best response to obtain a TME. One straightforward extension of this idea is that we add the team's best response to the restricted game. However, the following result shows that this extension cannot guarantee to converge to a TME.

Proposition 3. *x may not be a TME in G_T if x is a TME in G'_T and an NE in G_T, and $\nexists a_T \in (A_T \setminus A'_T)$ such that $u_T(a_T, x_n) > u_T(x)$.*

Proof. Consider G_T with three players (two teammates), $A_2 = \{1, 2, 3\}$, $A_1 = A_3 = \{1, 2\}$, and the following utility function:

$$u_T(a) = \begin{cases} 10 & \text{if } a = (1,1,1) \text{ or } (2,2,2) \\ 5 & \text{if } a = (2,3,1) \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

In G'_T, $A'_1 = A'_2 = A'_3 = \{1, 2\}$. Let player i's mixed strategy be $x_i = (x_i(1), x_i(2))$. Given x_1, x_2, and the adversary's action 1, the team's utility is:

$$u_T(x_1, x_2, 1) = 10x_1(1)x_2(1)$$

Given x_1, x_2, and the adversary's action 2, the team's utility is:

$$\begin{aligned} u_T(x_1, x_2, 2) &= 10(1 - x_1(1))(1 - x_2(1)) \\ &= 10(1 - x_1(1) - x_2(1) + x_1(1)x_2(1)) \end{aligned}$$

To achieve the TME value, we need to maximize the the minimum of $u_T(x_1, x_2, 1)$ and $u_T(x_1, x_2, 2)$ (see Eq.(1)). The case that $u_T(x_1, x_2, 1) = u_T(x_1, x_2, 2)$ gives the largest minimum to the team. Let $u_T(x_1, x_2, 1) = u_T(x_1, x_2, 2)$, we have $x_1(1) = 1 - x_2(1)$. Then we have:

$$\begin{aligned} u_T(x_1, x_2, 1) &= 10x_1(1)x_2(1) \\ &= 10(-(x_2(1) - 0.5)^2 + 0.25) \end{aligned}$$

which has its maximum 2.5 at $(x_2(1) = 0.5$. Then we have $(x_1(1) = 0.5$. Now we have the team-maxmin strategy profile $x_T = (x_1, x_2) = ((0.5, 0.5), (0.5, 0.5))$ with the team utility 2.5 in G'_T. Given this x_T, we can achieve $x_3 = (0.5, 0.5)$ by the following equation (see Appendix

A):

$$u_T(x_1, 1, x_3) = u_T(x_1, 2, x_3)$$
$$\Rightarrow 0.5 \times 10 \times x_3(1) = 0.5 \times 10 \times (1 - x_3(1))$$

Then $x = (x_1, x_2, x_3)$ is a TME in G'_T. Note that $u_T(x_1, 3, x_3) = 0.5 \times 0.5 \times 5 = 1.25 < 0.5$. Therefore, strategy profile $x = (x_1, (0.5, 0.5, 0), x_3)$ is an NE in G_T. In addition, we have $u_T(2, 3, x_3) = 2.5$, which is not larger than $u_T(x) = 2.5$. However, strategy profile $x' = ((0, 1), (0, \frac{1}{3}, \frac{2}{3}), (\frac{2}{3}, \frac{1}{3}))$ is an NE in G_T with utility $\frac{10}{3}(> 2.5)$ for the team. The reason is that $u_T(1, x'_2, x'_3) = 0 < \frac{10}{3}$, and $u_T(x'_1, 1, x'_3) = 0 < \frac{10}{3} = u_T(x'_1, 2, x'_3) = u_T(x'_1, 3, x'_3) = u_T(x'_1, x'_2, 1) = u_T(x'_1, x'_2, 2)$. Therefore, x is not a TME in G_T. □

The reason for the above failure is that the adversary tries to avoid the strategy that gives a high utility for the team, which gives a low utility for himself in zero-sum games. Then we need to add more actions to the restricted game, instead of only adding the team's best response (i.e., adding a_T such that $a_T \in \arg\max_{a'_T \in A_T} u_T(a_T, x_n)$ and $u_T(a_T, x_n) > u_T(x)$, where x is a TME in G'_T) or all of the team's better responses (i.e., adding a_T such that $u_T(a_T, x_n) > u_T(x)$). To do that, we can add the team's joint actions with outcomes that are better than the utility of the equilibrium in the restricted game (i.e., adding a_T such that $u_T(a_T, a_n) > u_T(x)$). However, there may be too many joint actions satisfying this condition in the full game, and adding too many actions to the restricted games will make it hard to compute a TME. In ISG, we compute the team's best response against the adversary strategy and add it (only one joint action) to the restricted game at each iteration. To speed up, two straightforward extensions of this idea are that: 1) we only add joint actions related to the support set of the adversary strategy, i.e., adding $(a_T, a_n) \in (A_T \setminus A'_T) \times \underline{A}_{n,x_n}$ such that $u_T(a_T, a_n) > u_T(x)$ to G'_T; and 2) we only add one joint action that can affect the TME value at each iteration. The following two results show that both extensions cannot guarantee to converge to a TME.

Proposition 4. *x may not be a TME in G_T if x is a TME in G'_T and an NE in G_T, and $\nexists(a_T, a_n) \in (A_T \setminus A'_T) \times \underline{A}_{n,x_n}$ such that $u_T(a_T, a_n) > u_T(x)$.*

Proof. Consider G_T with three players (two teammates), $A_1 = A_2 = \{1, 2, 3\}$, $A_3 = \{1, 2\}$, and the following utility function:

$$u_T(a) = \begin{cases} 3 & \text{if } a = (1,1,1) \text{ or } (1,1,2) \\ 21 & \text{if } a = (2,2,1) \\ 3 & \text{if } a = (2,3,1), (3,2,1), \text{ or } (3,3,1) \\ 10 & \text{if } a = (2,3,2), (3,2,2), \text{ or } (3,3,2) \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

In G'_T, $A'_1 = A'_2 = A'_3 = \{1, 2\}$. Obviously, pure strategy profile $a = (1, 1, 1)$ is an NE in G'_T and G_T. Now we check that x is a TME in G'_T. Let player i's mixed strategy be $x_i = (x_i(1), x_i(2))$. Given x_1, x_2, and the adversary's action 1, the team's utility is:

$$u_T(x_1, x_2, 1) = 3x_1(1)x_2(1) + 21(1 - x_1(1))(1 - x_2(1))$$

Given x_1, x_2, and the adversary's action 2, the team's utility is:

$$u_T(x_1, x_2, 2) = 3x_1(1)x_2(1)$$

We can see that, if $x_1(1) < 1$ and $x_2(1) < 1$, action 1 is strictly dominated by action 2 for the adversary. However, given the adversary action 2 in G'_T, the team cannot achieve the utility that is higher than $u_T(a) = 3$. Now, given any strategy of the adversary, if player 1's strategy is $x_1(1) = 1$, player 2's best response is $x_1(1) = 1$, and vice versa. In this case, $a = (1, 1, 1)$ (i.e., $x = ((1, 0), (1, 0), (1, 0))$) is a TME in G'_T. Given x, $\nexists(a_T, a_n) \in (A_T \setminus A'_T) \times \underline{A}_{n,x_n}$ such that $u_T(a_T, a_n) > u_T(x)$. However, strategy profile $x' = ((0, 0.5, 0.5), (0, 0.5, 0.5), (\frac{5}{14}, \frac{9}{14}))$ is an NE in G_T with utility $7.5(> 3)$ for the team. The reason is that $u_T(1, x'_2, x'_3) = 0 < 7.5 = u_T(2, x'_2, x'_3) = u_T(3, x'_2, x'_3)$, and $u_T(x'_1, 1, x'_3) = 0 < 7.5 = u_T(x'_1, 2, x'_3) = u_T(x'_1, 3, x'_3) = u_T(x'_1, x'_2, 1) = u_T(x'_1, x'_2, 2)$. Therefore, x is not a TME in G_T. □

Proposition 5. *x may not be a TME in G_T if x is a TME in G'_T and an NE in G_T, and $\nexists(a_T, a_n) \in (A_T \setminus A'_T) \times A'_n$ with $u_T(a_T, a_n) > u_T(x)$ such that the TME value in G'_T changes after adding a_T to A'_T.*

Proof. Consider G_T with three players (two teammates), $A_1 = A_2 = \{1, 2, 3\}$, $A_3 = \{1, 2\}$, and the following utility function:

$$u_T(a) = \begin{cases} 3 & \text{if } a = (1,1,1) \text{ or } (1,1,2) \\ 100 & \text{if } a = (2,2,1) \text{ or } (3,3,2) \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

In G'_T, $A'_1 = A'_2 = \{1\}, A'_3 = \{1, 2\}$. Obviously, $x = ((1), (1), (0.5, 0.5))$ is an NE in G'_T and G_T with utility 3 for the team, which is also a TME in G'_T. If we add $a_T = (2, 2)$ or $(3, 3)$, according to the analysis on the case in Eq.(4), x with $x_1(1) = 1 = x_2(1)$ and $x_3(1) = x_3(2) = 0.5$ is still a TME in G'_T with the TME value 3. That is, the TME value in G'_T does not change. However, strategy profile $x' = ((0, 0.5, 0.5), (0, 0.5, 0.5), (0.5, 0.5)$ is an NE in G_T with utility $25(> 3)$ for the team. The reason is that $u_T(1, x'_2, x'_3) = 0 < 25 = u_T(2, x'_2, x'_3) = u_T(3, x'_2, x'_3)$, and $u_T(x'_1, 1, x'_3) = 0 < 25 = u_T(x'_1, 2, x'_3) = u_T(x'_1, 3, x'_3) = u_T(x'_1, x'_2, 1) = u_T(x'_1, x'_2, 2)$. Therefore, x is not a TME in G_T. □

Algorithm 1 ISG for a TME (ISGT)

1: Initialize G'_T with $A' \leftarrow A'_T \times A'_n, u_T(x^*) \leftarrow -\infty, x^* \leftarrow \emptyset$
2: **repeat**
3:   Iteration $\leftarrow 1$
4:   **repeat**
5:     $x \leftarrow$ CoreTME(G'_T)
6:     **if** $u_T(x) = u_T(x^*)$& Iteration $= 1$ **then**
7:       **return** x^*
8:     **end if**
9:     **for** $i \in N$ **do**
10:       $A'_i \leftarrow A'_i \cup \{a_i \leftarrow BR(x_{-i})\}$
11:     **end for**
12:     Iteration $\leftarrow$ Iteration $+1$
13:   **until** convergence (A' does not change)
14:   $x^* \leftarrow x$
15:   $A''_T \leftarrow \{a_T \mid u_T(a_T, a_n) > u_T(x), (a_T, a_n) \in (A_T \setminus A'_T) \times A'_n\}$
16:   $A'_T \leftarrow A'_T \cup A''_T$
17: **until** convergence (A'_T does not change)
18: **return** x.

4.3. Converging to a TME

Based on the discussion in the previous section, this section proposes our ISG algorithm for converging to a TME (ISGT). In fact, it exploits that a TME is an NE maximizing the team's utility. Basically, ISGT makes sure that its output x satisfies three conditions: 1) x is a TME in G'_T; 2) x is an NE in G_T; and 3) $\nexists (a_T, a_n) \in (A_T \setminus A'_T) \times A'_n$ such that $u_T(a_T, a_n) > u_T(x)$.

Our ISGT is shown in Algorithm 1.[2] Line 5 computes a TME for G'_T by solving Problem (1), and Line 10 expands A'_i for each player i by solving the best response oracle. If 1) x is a TME in G'_T; and 2) x is an NE in G_T (by Theorem 1), the inner loop will terminate. Line 15 looks for a_T that gives a better utility to the team outside of G'_T and updates A'_T (Line 16). If no such a_T, the outer loop will terminate. Moreover, after obtaining an NE, ISGT records it as x^* in Line 14, and immediately returns it if there are not better NEs in Line 7.

Now ISGT definitely reduces the strategy space to $A_T \times A'_n$ to find a TME. To show that the output x of ISGT is a TME, we first show that if a TME in a restricted game is an NE in the full game, then it is also a TME in a larger restricted game including all adversary strategies.

Lemma 1. *If x is a TME in G'_T and an NE in G_T, then x is a TME in G''_T with $A'' = A'_T \times A_n$.*

Proof. Suppose that there is an NE $x'(i.e., (x'_T, x'_n))$ in G''_T such that $u_T(x') > u_T(x)$ and $\underline{A}_{n,x'_n} \nsubseteq A'_n$. Then, $u_n(x') \geq u_n(x'_T, x''_n)$ ($\forall x''_n$ with $\underline{A}_{n,x''_n} \subseteq A'_n$), i.e., $\min_{x''_n, \underline{A}_{n,x''_n} \subseteq A'_n} u_T(x'_T, x''_n) \geq u_T(x')$. Therefore, in G'_T, we have $\max_{x_T} \min_{x''_n} u_T(x_T, x''_n) \geq \min_{x''_n} u_T(x'_T, x''_n) \geq u_T(x') > u_T(x)$, which means that, $u_T(x)$ is not the TME value in G'_T, i.e., x is not a TME in the game G'_T, concluding the proof. □

Based on the above result, now we show that if a TME in a restricted game is an NE in the full game and there is no pure strategy profile outside the restricted game giving larger utility than this TME value, then it is also a TME in the full game.

Lemma 2. *x is a TME in G_T if a) x is a TME in G'_T, b) x is an NE in G_T, and c) $\nexists (a_T, a_n)$ such that $u_T(a_T, a_n) > u_T(x)$ with $a_T \in A_T \setminus A'_T$.*

Proof. By Lemma 1, x is a TME of G''_T with $A'' = A'_T \times A_n$. Suppose that there is a TME $x' = (x'_T, x'_n)$ in G_T such that $u_T(x') > u_T(x)$ and $\underline{A}_{T,x'_T} \nsubseteq A'_T$. That is, there is $a_i \notin A'_i (i \in T)$ such that $x'_i(a_i) > 0$. By the condition c), we have $u_T(a_T, a_n) \leq u_T(x)$ for each $a_T \in A_T \setminus A'_T$. Then, $u_T(a_i, x'_{-i}) \leq u_T(x) < u_T(x')$, and then there exists some a'_i such that $u_T(a'_i, x'_{-i}) > u_T(x')$ (otherwise, $u_T(x')$ will not be larger than $u_T(a_i, x'_{-i})$). Then we construct a new strategy x'' which is identical to x', but $x''_i(a_i) = 0$ and $x''_i(a'_i) = x'_i(a_i) + x'_i(a'_i)$. Then $u_T(x'') - u_T(x') = x'_i(a_i)(u_T(a'_i, x'_{-i}) - u_T(a_i, x'_{-i})) > 0$, which causes a contradiction, concluding the proof. □

Theorem 2. *x is a TME in G_T if: 1) x is a TME in G'_T; 2) x is an NE in G_T; and 3) $\nexists (a_T, a_n) \in (A_T \setminus A'_T) \times A'_n$ such that $u_T(a_T, a_n) > u_T(x)$.*

Proof. By Lemma 2, x is a TME in G''_T with $A''_T = (A_T, A'_n)$. Then, with the condition 2), x is a TME in G_T with $A = (A_T, A_n)$ by Lemma 1. □

In addition, ISGT (Line 7) indeed can terminate if the TME value does not change after adding all joint actions that are better than the equilibrium strategy by the following result.

Theorem 3. *x is a TME in G_T if x is a TME in G'_T and an NE in G_T, $u_T(x)$ is the TME value in G''_T with $\nexists (a_T, a_n) \in (A_T \setminus A'_T) \times A'_n$ such that $u_T(a_T, a_n) > u_T(x)$, and G'_T is a restricted game of G''_T that is a restricted game of G_T.*

Proof. x is an NE in G'_T because x is a TME in G'_T. Then, x is an NE in G''_T because x is an NE in G_T, and G'_T is a restricted game of G''_T that is a restricted game of G_T. Moreover, x is a TME in G''_T because $u_T(x)$ is the TME value in G''_T. By Theorem 2, x is a TME in G_T. □

[2]Another algorithm framework we can develop is only using one loop. That is, we directly use the operations in Lines 15 and 16 of Algorithm 1 to replace the best response oracle for the team in Line 10 of Algorithm 1. However, experimental results show that this framework can be significantly slower than the current framework shown in Algorithm 1, which may be partially due to that this framework will add too many actions to the restricted game at early iterations. Therefore, we adopt the current framework.

More importantly, Propositions 3–5 have shown the impossibility of relaxing these conditions in Theorems 2 and 3. Now we can have our following conclusion.

Theorem 4. *ISGT converges to a TME in G_T.*

Proof. A is finite, so ISGT will terminate. ISGT terminates with output x satisfying conditions in Theorem 2 or 3, so x is a TME in G_T. □

5. The CTME Based ISGT (CISGT)

Even though ISGT can guarantee to converge to a TME, it is not efficient enough as ISGT needs to solve the non-convex Problem (1) in G'_T at each iteration. To speed up, we try to reduce the number of iterations by effectively initializing the restricted game G'_T through computing CTMEs. Now we first discuss the limitations of computing TMEs based on CTMEs.

5.1. Limitations of Computing TMEs by CTMEs

A CTME is close to a TME, and CTMEs are used to approximate TMEs (Basilico et al., 2017b) by Transforming the correlated strategy in a CTME into the team's Mixed Strategy Profile (TMSP). In this section, we show the limitations of computing TMEs based on CTMEs. We first show that using TMSPs may cause an arbitrarily large loss to the team. Second, we show that a TME in G'_T with the strategies for computing a CTME may not be an NE in G_T, and it may not be a TME in G_T even it is an NE in G_T. Moreover, we show that using the strategy profile in this TME of such G'_T may cause an arbitrarily large loss to the team as well.

To our best knowledge, the best algorithm (Basilico et al., 2017b) to obtain a TMSP is: Given the team's strategy in a CTME: $\overline{x}_T \in \Delta(A_T)$, player 1's mixed strategy $x_1(a_1) = \sum_{a' \in A_{T\setminus\{1\}}} \overline{x}_T(a_1, a')(\forall a_1 \in A_1)$; player i's mixed strategy $x_i(a_i) = \frac{1}{|\underline{A}_{i,\overline{x}_T}|}$ if $|\underline{A}_{i,\overline{x}_T}| > 0$, otherwise $x_i(a_i) = 0$ $(\forall i \in T \setminus \{1\}, a_i \in A_i)$. This algorithm returns the team's best TMSP among TMSPs obtained by exchanging player 1 with each team member. We call it TMSP-Alg. To measure the inefficiency of transforming the correlated strategies in a CTME into a TMSP, we adopt the concept of Price of Correlated strategies (PoC) (Zhang & An, 2020). Formally, PoC $= \frac{v_m}{v_c}$, where v_m is the TME value and v_c is the team utility obtained from a TMSP. Unfortunately, PoC can be arbitrarily large in G_T.[3]

Proposition 6. *PoC can be arbitrarily large in G_T.*

Proof. Consider G_T with three players (two teammates), $A_1 = A_2 = A_3 = \{1, 2, 3\}$, and the following utility function:

$$u_T(a_1,a_2,a_3) = \begin{cases} 1 & \text{if } a_1 = a_2 = a_3 \\ \frac{1}{4} & \text{if } a_1 = 1, a_2 = 2 \\ -\frac{5}{4} & \text{if } a_1 = 2, a_2 = 1 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

A CTME $\overline{x}$ is $\overline{x}_T(1,1) = \overline{x}_T(2,2) = \overline{x}_T(3,3) = \frac{1}{3}$ (note that, given any adversary strategy, one of these three pure strategies dominates other strategies, and any of them should be played with nonzero probability otherwise the adversary best response gives utility 0 to the team) and $\overline{x}_n(1) = \overline{x}_n(2) = \overline{x}_n(3) = \frac{1}{3}$. Now, by TMSP-Alg, $\overline{x}$'s unique TMSP prescribes that $x_1 = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ and $x_2 = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, while the adversary (player 3) is indifferent between playing any strategies given this TMSP. Then, $v_c = 0$.

An NE is a pure strategy profile $(1, 2, 3)$, where player 1 plays action 1, player 2 plays action 2, while player 3 plays action 3. The team's utility is $u_T(1, 2, 3) = 0.25$. The reason is that the team will obtain utility 0 if any team member unilaterally deviates to other actions, and the adversary is indifferent among all strategies.

Therefore, PoC $= \frac{v_m}{v_c} > \frac{0.25}{0} = \infty$. □

Another idea is to compute a TME in G'_T with strategies for computing a CTME. There are two cases: 1) G'_T including all support sets of all players in a CTME; and 2) G'_T including all strategies for computing a CTME. However, in each case, this TME may not be an NE in G_T, and it may not be a TME in G_T even it is an NE in G_T by following results.

Proposition 7. *x may not be an NE in G_T if $\overline{x}$ is a CTME in G_T, and x is a TME in G'_T with $A' = (\times_{i\in T}\underline{A}_{i,\overline{x}_T}) \times \underline{A}_{n,\overline{x}_n}$.*

Proof. Consider G_T with three players (two teammates), $A_1 = A_2 = \{1, 2\}$, $A_3 = \{1, 2, 3\}$, and the following utility function:

$$u_T(a) = \begin{cases} 10 & \text{if } a = (1,1,1) \text{ or } (2,2,2) \\ 10 & \text{if } a = (1,1,3) \text{ or } (2,2,3) \\ -10 & \text{if } a = (2,1,3) \text{ or } (1,2,3) \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

A CTME $\overline{x}$ is $\overline{x}_T(1,1) = \overline{x}_T(2,2) = 0.5$ and $\overline{x}_3(1) = \overline{x}_3(2) = 0.5$ with utility 5 for the team (it is easy to verify that no players would like to deviate to other strategies, e.g., action 3 for the adversary with $u_T(\overline{x}_T, 3) = 10 > 5$ is not better than $\overline{x}_3$). Then we have G'_T with with $A'_1 = A'_2 = A'_3 = \{1, 2\}$. According to the analysis on the case in Eq.(3), x with $x_i(1) = x_i(2) = 0.5$ is a TME in G'_T with utility 2.5 for the team. However, for the adversary action

[3] Omitted proofs in this section are in Appendix B.

Algorithm 2 The CTME Based ISGT (CISGT)

1: Initialize G'_T with $A' = A'_T \times A'_n, u_T(x^*) = -\infty, x^* \leftarrow \emptyset$
2: **repeat**
3: $\quad \overline{x} \leftarrow CoreCTME(G'_T)$
4: $\quad A'_T \leftarrow A'_T \cup \{a_T \leftarrow BR(\overline{x}_n)\}$
5: $\quad A'_n \leftarrow A'_n \cup \{a_n \leftarrow BR(\overline{x}_T)\}$
6: **until** convergence (A' does not change)
7: $x_T \leftarrow$TMSP-Alg($\overline{x}_T$)
8: **if** $u_T(x_T, BR(x_T)) \geq u_T(\overline{x})$ **then**
9: $\quad$ **return** $(x_T, \overline{x}_n)$
10: **end if**
11: **repeat**
12: $\quad$ Lines 4–13 in ISGT
13: $\quad$ **if** $u_T(x) \geq u_T(\overline{x})$ **then**
14: $\quad\quad$ **return** x
15: $\quad$ **end if**
16: $\quad$ Lines 14–16 in ISGT
17: **until** convergence (A'_T does not change)
18: **return** x.

3, $u_T(x_T, 3) = 0.5 \times 0.5(10 + 10 - 10 - 10) = 0 < 2.5$. Therefore, x is not an NE in the original game G_T. □

Corollary 1. *x may not be an NE in G_T if $\overline{x}$ is a CTME in G_T and is computed in G'_T, and x is a TME in G'_T.*

Proposition 8. *x may not be a TME in G_T if $\overline{x}$ is a CTME in G_T, and x is a TME in G'_T with $A' = (\times_{i\in T}\underline{A}_{i,\overline{x}_T}) \times \underline{A}_{n,\overline{x}_n}$ and an NE in G_T.*

Corollary 2. *x may not be a TME in G_T if $\overline{x}$ is a CTME in G_T and is computed in G'_T, and x is a TME in G'_T and an NE in G_T.*

Computing a TME in G'_T with strategies for computing a CTME not only cannot guarantee to obtain a TME in G_T, but also can cause an arbitrarily large loss to the team.

Proposition 9. *If $\overline{x}$ is a CTME in G_T, and x is a TME in G'_T with $A' = (\times_{i\in T}\underline{A}_{i,\overline{x}_T}) \times \underline{A}_{n,\overline{x}_n}$, then playing x_T may cause an arbitrarily large loss to the team.*

Corollary 3. *If $\overline{x}$ is a CTME in G_T, and x is a TME in G'_T where $\overline{x}$ is computed, then playing x_T may cause an arbitrarily large loss to the team.*

5.2. CISGT: Efficiently Converging to a TME

Based on our discussion in the previous section, this section proposes our CISGT. In addition to the operations in ISGT, CISGT has two new operations: 1) it initializes the restricted game through computing a CTME to reduce the number of iterations for solving Prolbem (1); and 2) it exploits that the team's utility in a CTME is an upper bound of the TME value (Basilico et al., 2017b) to terminate earlier.

Our CISGT is shown in Algorithm 2.[4] CISGT uses Vanilla-ISG to compute a CTME at Lines 2–6. Then, CISGT computes a TMSP at Line 7 and then checks whether we have obtained a TME to return it at Lines 8–9. After that, CISGT repeats the operations in ISGT to compute a TME in G'_T and makes sure that it is also an NE in G_T at Line 12. CISGT then checks whether we have obtained a TME to return it at Lines 13–14. At Line 16, CISGT adds actions by repeating the operations in ISGT.

To show the convergence of CISGT, we first show that, a TMSP is part of a TME if the utility obtained by it is not less than to the team's utility in a CTME.

Theorem 5. *In G_T, given a CTME $\overline{x} = (\overline{x}_T, \overline{x}_n)$ and a mixed strategy profile for the team x_T such that $u_T(x_T, BR(x_T)) \geq u_T(\overline{x})$, then $(x_T, \overline{x}_n)$ is a TME.*

Proof. Let v^* be the TME value of G_T. Then $v^* \leq u_T(\overline{x})$ (Basilico et al., 2017b). Now we have $v^* \leq u_T(\overline{x}) = \max_{\overline{x}'_T}\min_{\overline{x}'_n} u_T(\overline{x}'_T, \overline{x}'_n) = \max_{\overline{x}'_T} u_T(\overline{x}'_T, \overline{x}_n)$. Then, we have $u_T(x_T, BR(x_T)) = \min_{x'_n} u_T(x_T, x'_n) \leq u_T(x_T, \overline{x}_n) \leq \max_{x'_i} u_T(x_{T\setminus\{i\}}, x'_i, \overline{x}_n) \leq \max_{\overline{x}'_T} u_T(\overline{x}'_T, \overline{x}_n) = u_T(\overline{x})$. Consequently, given x_T, for any adversary strategy x'_n, we have $u_n(x_T, x'_n) - u_n(x_T, \overline{x}_n) = -u_T(x_T, x'_n) - (-u_T(x_T, \overline{x}_n)) = u_T(x_T, \overline{x}_n) - u_T(x_T, x'_n) \leq u_T(\overline{x}) - u_T(x_T, BR(x_T)) \leq 0$. Similarly, given $\overline{x}_n$ and $x_{T\setminus\{i\}}$, for any player i's strategy x'_i, we have $u_T(x_{T\setminus\{i\}}, x'_i, \overline{x}_n) - u_T(x_T, \overline{x}_n) \leq u_T(\overline{x}) - u_T(x_T, BR(x_T)) \leq 0$. Then, $(x_T, \overline{x}_n)$ is an NE. In addition, due to $u_T(x_T, BR(x_T)) = \min_{x'_n} u_T(x_T, x'_n) \leq \max_{x'_T}\min_{x'_n} u_T(x'_T, x'_n) = v^*$, we have $u_T(x_T, \overline{x}_n) - v^* \leq u_T(\overline{x}) - u_T(x_T, BR(x_T)) \leq 0$ and $v^* - u_T(x_T, \overline{x}_n) \leq u_T(\overline{x}) - u_T(x_T, BR(x_T)) \leq 0$. Therefore, $(x_T, \overline{x}_n)$ is a TME. □

Similarly, a TME in G'_T is a TME in G_T if the utility obtained by it is not less than to the team's utility in a CTME.

Theorem 6. *x is a TME in G_T if x is a TME in G'_T and an NE in G_T, $\overline{x}$ is a CTME in G_T with $u_T(x) \geq u_T(\overline{x})$.*

Proof. Let v^* be the TME value of G_T. Then $v^* \leq u_T(\overline{x})$ (Basilico et al., 2017b). Now we have $u_T(x) \leq v^* \leq u_T(\overline{x})$. Obviously, x is a TME in G_T. □

Finally, based on the above results, we have the following conclusion.

Theorem 7. *CISGT converges to a TME.*

Proof. First, CISGT will converge because the action set for each player is finite in G_T. Second, the output x is a TME by Theorems 4–6. □

[4]When we compute an ϵ-TME, we only need to set $u_T(x_T, BR(x_T)) \geq u_T(\overline{x}) - \epsilon$ at Line 8 and $u_T(x) \geq u_T(\overline{x}) - \epsilon$ at Line 13, and all properties still hold.

L×W	5×5	5×5	5×5	5×5	5×5	4×4	6×6	8×8	10×10
(p,q)	(0.8,0.6)	(0.7,0.5)	(0.6,0.4)	(0.5,0.3)	(0.4,0.2)	(0.4,0.2)	(0.4,0.2)	(0.4,0.2)	(0.4,0.2)
FullTME		∞	448s	50.4s	17.8s	0.3s	∞		
ISGT					>1000s	4s	>1000s		
CISGT	9.8s	5.9s	4.7s	3.7s	2.3s	2.2s	8.3s	24s	57s

Table 1. Computing TMEs: ∞ represents out of memory.

L×W	4×4	4×4	4×4	4×4	4×4	5×5	6×6	7×7	8×8
(p,q)	(0.4,0.2)	(0.3,0.2)	(0.3,0.1)	(0.2,0.1)	(0.1,0,1)	(0.1,0.1)	(0.1,0.1)	(0.1,0.1)	(0.1,0.1)
Vanilla-ISG	35%	38%	38%	30%	26%	53%	47%	56%	55%
TMSP-Alg	61%	50%	51%	44%	38%	75%	54%	79%	87%

Table 2. Gaps relative to CISGT.

6. Experiment Evaluation

We experimentally evaluate CISGT, comparing the performance of CISGT with that of ISGT and the state-of-the-art algorithm (Zhang & An, 2020) (FullTME) for computing TMEs (i.e., ϵ-TME with $\epsilon = 0.05$). FullTME enumerates all pure strategies and uses global optimization techniques to approximate multilinear terms in Problem (1) by a mixed-integer linear program. We use CPLEX solver (version 12.9) for solving all linear programs. All experiments are run on a machine with 6-core 3.6GHz CPU and 32GB memory.

We conduct experiments on the classic network security games [5] (Washburn & Wood, 1995; Jain et al., 2011; I-washita et al., 2016) to evaluate our approach. In a network security game, the adversary starts at a source node (he may have many possible source nodes) and travels along the path he chooses to one of his targets. That is, an action (a pure strategy) of the adversary is a path from a source to a target, and then the action space includes all possible paths. The police officers form a team, and each police occupies one of the possible edges on the network to try to catch the adversary before the target is reached. That is, an action (a pure strategy) of each team member is an edge. Similar to police officers in the NYPD, who maintain "sector integrity" (NYPD, 2020a), the action space of each police officer is disjoint with others in our setting. The adversary may have different values for different targets, and the adversary will succeed if his choosing path does not overlap with the edges chosen by the team; otherwise, the adversary will obtain nothing. All networks are generated by the grid model with random edges (Peng et al., 2014), which models the real urban network with some parameters. That is, it samples a square network with $L \times W$ nodes, and it generates horizontal/vertical edges between neighbors with probability p, and diagonal ones with probability q. By default, $n = 3$, and results are all averaged over 30 instances that are randomly generated.

Vanilla-ISG to compute CTMEs is the double oracle algorithm proposed by Jain et al. (2011), including the linear program for computing a maximin strategy for the team, and mixed-integer linear programs of best response oracles for the team and the adversary, respectively. These best response oracles can be easily extended to CISGT. TMEs for restricted games are computed by the algorithm proposed by Zhang & An (2020), i.e., FullTME computes TMEs in restricted games.

Results in Table 1 show that CISGT is orders of magnitude faster than ISGT and FullTME.[6] Specially, when FullTME, enumerating all pure strategies to compute a TME in a full game, runs out of the memory, CISGT still runs efficiently.

In addition, we compare the solution quality of CISGT with that of Vanilla-ISG and TMSP-Alg. Table 2 shows the possible gaps, which are the relative distance between the team utility (v_m) obtained by CISGT and the team utility v obtained by Vanilla-ISG or TMSP-Alg, i.e., $\frac{|v-v_m|}{|v_m|} \times 100\%$. The team will lose more utility if the gap is larger. We can see that the team may lose a large utility if Vanilla-ISG or TMSP-Alg is deployed.

7. Conclusion and Future Work

This paper proposes an efficient ISG algorithm (CISGT) to compute TMEs for zero-sum multiplayer games. Our algorithm is the first incremental strategy generation algorithm guaranteeing to converge to a TME, which significantly overcomes the limitation of state-of-the-art algorithms. Especially, our algorithm can efficiently solve the cases that the baselines cannot solve. In the future, we can extend our CISGT to extensive-form games (Bosansky et al., 2014) and use it to develop learning algorithms (Lanctot et al., 2017).

[5] Network security games can be easily extended to other games, including the robot planning problem in the adversary environment (McMahan et al., 2003), hider-seeker games (Halvorson et al., 2009), patrolling games with alarm systems (Basilico et al., 2017a), and green security games (Wang et al., 2019). Then our results will also hold in these games.

[6] In addition, ISGT's another framework we mentioned in Section 4.3 needs 898s on the smallest network with 4×4 and (0.4,0.2).

Acknowledgments

Bo An is partially supported by Singtel Cognitive and Artificial Intelligence Lab for Enterprises (SCALE@NTU), which is a collaboration between Singapore Telecommunications Limited (Singtel) and Nanyang Technological University (NTU) that is funded by the Singapore Government through the Industry Alignment Fund Industry Collaboration Projects Grant. This research is also 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), and NTU.

References

Basilico, N., Celli, A., De Nittis, G., and Gatti, N. Coordinating multiple defensive resources in patrolling games with alarm systems. In *AAMAS*, pp. 678–686, 2017a.

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

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

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

Brown, N. and Sandholm, T. Superhuman AI for multiplayer poker. *Science*, 365(6456):885–890, 2019. doi: 10.1126/science.aay2400.

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

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

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

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

Halvorson, E. D., Conitzer, V., and Parr, R. Multi-step multi-sensor hider-seeker games. In *AAAI*, pp. 159–166, 2009.

Hansen, K. A., Hansen, T. D., Miltersen, P. B., and Sørensen, T. B. Approximability and parameterized complexity of minmax values. In *WINE*, pp. 684–695. Springer, 2008.

Iwashita, H., Ohori, K., Anai, H., and Iwasaki, A. Simplifying urban network security games with cut-based graph contraction. In *AAMAS*, pp. 205–213, 2016.

Jain, M., Korzhyk, D., Vaněk, O., Conitzer, V., Pěchouček, M., and Tambe, M. A double oracle algorithm for zero-sum security games on graphs. In *AAMAS*, pp. 327–334, 2011.

Lanctot, M., Zambaldi, V., Gruslys, A., Lazaridou, A., Tuyls, K., Pérolat, J., Silver, D., and Graepel, T. A unified game-theoretic approach to multiagent reinforcement learning. In *NeurIPS*, pp. 4190–4203, 2017.

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

Muller, P., Omidshafiei, S., Rowland, M., Tuyls, K., Perolat, J., Liu, S., Hennes, D., Marris, L., Lanctot, M., Hughes, E., et al. A generalized training approach for multiagent learning. In *ICLR*, 2020.

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

NYPD. Neighborhood policing. *https://www1.nyc.gov/site/nypd/bureaus/patrol/neighborhood-coordination-officers.page*, 2020a.

NYPD. Patrol. *https://www1.nyc.gov/site/nypd/bureaus/patrol/patrol-landing.page*, 2020b.

Peng, W., Dong, G., Yang, K., and Su, J. A random road network model and its effects on topological characteristics of mobile delay-tolerant networks. *IEEE Transactions on Mobile Computing*, 13(12):2706–2718, 2014.

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

Sinha, A., Fang, F., An, B., Kiekintveld, C., and Tambe, M. Stackelberg security games: Looking beyond a decade of success. In *IJCAI*, pp. 5494–5501, 2018.

Von Neumann, J. and Morgenstern, O. *Theory of Games and Economic Behavior*. Princeton University Press, 1953.

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

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

Wang, Y., Shi, Z. R., Yu, L., Wu, Y., Singh, R., Joppa, L., and Fang, F. Deep reinforcement learning for green security games with real-time information. In *AAAI*, volume 33, pp. 1401–1408, 2019.

Washburn, A. and Wood, K. Two-person zero-sum games for network interdiction. *Operations Research*, 43(2): 243–251, 1995.

Zhang, Y. and An, B. Computing team-maxmin equilibria in zero-sum multiplayer extensive-form games. In *AAAI*, pp. 2318–2325, 2020.

Zhang, Y., Guo, Q., An, B., Tran-Thanh, L., and Jennings, N. R. Optimal interdiction of urban criminals with the aid of real-time information. In *AAAI*, volume 33, pp. 1262–1269, 2019.

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

Appendix

A. Computing the Adversary Strategy in a TME

After computing the team-maxmin strategy profile x_T, we can compte the adversary strategy x_n by minimizing the team's utility and making sure that no team members would like to deviate from their strategies in x_T (von Stengel & Koller, 1997). Then x_n can be computed by solving the following linear program (von Stengel & Koller, 1997):

$$\min_{x_n} \sum_{i \in T} z_i \tag{8a}$$

$$z_i - \sum_{a_n \in A_n} x_n(a_n) u_T(a_i, x_{T\setminus\{i\}}, a_n) \geq 0 \quad \forall i \in T, a_i \in A_i \tag{8b}$$

$$\sum_{a_n \in A_n} x_n(a_n) = 1 \tag{8c}$$

$$x_n(a_n) \geq 0 \quad \forall a_n \in A_n \tag{8d}$$

B. Omitted Proofs

Corollary 1. *x may not be an NE in G_T if $\overline{x}$ is a CTME in G_T and is computed in G'_T, and x is a TME in G'_T.*

Proof. Suppose a CTME $\overline{x}$ is computed in G'_T and $A' = (\times_{i\in T}\underline{A}_{i,\overline{x}_T}) \times \underline{A}_{n,\overline{x}_n}$. By Proposition 7, x may not be an NE in G_T, even if x is a TME in G'_T. □

Proposition 8. *x may not be a TME in G_T if $\overline{x}$ is a CTME in G_T, and x is a TME in G'_T with $A' = (\times_{i\in T}\underline{A}_{i,\overline{x}_T}) \times \underline{A}_{n,\overline{x}_n}$ and an NE in G_T.*

Proof. Consider the case in Eq.(3). A CTME $\overline{x}$ is $\overline{x}_T(1,1) = \overline{x}_T(2,2) = 0.5$ and $\overline{x}_3(1) = \overline{x}_3(2) = 0.5$ with utility 5 for the team (it is easy to verify that no players would like to deviate to other strategies). Then we have G'_T with with $A'_1 = A'_2 = A'_3 = \{1,2\}$. According to the analysis on the case in Eq.(3), x with $x_i(1) = x_i(2) = 0.5$ is a TME in G'_T with utility 2.5 for the team. x is an NE in G_T. However, according to the analysis on the case in Eq.(3), x is not a TME in G_T. □

Corollary 2. *x may not be a TME in G_T if $\overline{x}$ is a CTME in G_T and is computed in G'_T, and x is a TME in G'_T and an NE in G_T.*

Proof. Suppose a CTME $\overline{x}$ is computed in G'_T and $A' = (\times_{i\in T}\underline{A}_{i,\overline{x}_T}) \times \underline{A}_{n,\overline{x}_n}$. By Proposition 8, x may not be a TME in G_T, even if x is a TME in G'_T and an NE in G_T. □

Proposition 9. *If $\overline{x}$ is a CTME in G_T, and x is a TME in G'_T with $A' = (\times_{i\in T}\underline{A}_{i,\overline{x}_T}) \times \underline{A}_{n,\overline{x}_n}$, then playing x_T may cause an arbitrarily large loss to the team.*

Proof. Consider G_T with utilities shown in Eq.(7). As shown in the proof for Proposition 7, A CTME $\overline{x}$ is $\overline{x}_T(1,1) = \overline{x}_T(2,2) = 0.5$ and $\overline{x}_3(1) = \overline{x}_3(2) = 0.5$ with utility 5 for the team. Then we have G'_T with with $A'_1 = A'_2 = A'_3 = \{1,2\}$, and x with $x_i(1) = x_i(2) = 0.5$ is a TME in G'_T with utility 2.5 for the team. Given x_T, the adversary best response is action 3 with utility $u_T(x_T, 3) = 0.5 \times 0.5(10+10-10-10) = 0$ for the team. Now an NE $x' = ((\frac{1}{3}, \frac{2}{3}), (\frac{1}{3}, \frac{2}{3}), (\frac{2}{3}, 0, \frac{1}{3}))$ (it is easy to verify that no players would like to deviate to other strategies, e.g., action 3 for the adversary with $u_T(x_T, 2) = \frac{40}{9} > \frac{10}{9}$ is not better than x'_3) will given utility $\frac{10}{9}$ to the team. Then, playing x_T may cause an arbitrarily large loss to the team because $\frac{10/9}{0} = \infty$. □

Corollary 3. *If $\overline{x}$ is a CTME in G_T, and x is a TME in G'_T where $\overline{x}$ is computed, then playing x_T may cause an arbitrarily large loss to the team.*

Proof. Suppose a CTME $\overline{x}$ is computed in G'_T and $A' = (\times_{i\in T}\underline{A}_{i,\overline{x}_T}) \times \underline{A}_{n,\overline{x}_n}$. By Proposition 9, playing x_T may cause an arbitrarily large loss to the team. □