Correlation-Based Algorithm for Team-Maxmin Equilibrium in Multiplayer Extensive-Form Games

Youzhi Zhang[1] , **Bo An**[2] , **V. S. Subrahmanian**[1,3]
[1]Department of Computer Science, Northwestern University, USA
[2]School of Computer Science and Engineering, Nanyang Technological University, Singapore
[3]Buffet Institute for Global Affairs, Northwestern University, USA
youzhi.zhang@northwestern.edu, boan@ntu.edu.sg, vss@northwestern.edu

Abstract

Efficient algorithms computing a Nash equilibrium have been successfully applied to large zero-sum two-player extensive-form games (e.g., poker). However, in multiplayer games, computing a Nash equilibrium is generally hard, and the equilibria are not exchangeable, which makes players face the problem of selecting one of many different Nash equilibria. In this paper, we focus on an alternative solution concept in zero-sum multiplayer extensive-form games called Team-Maxmin Equilibrium (TME). It is a Nash equilibrium that maximizes each team member's utility. As TME is unique in general, it avoids the equilibrium selection problem. However, it is still difficult (FNP-hard) to find a TME. Computing it can be formulated as a non-convex program, but existing algorithms are capable of solving this program for only very small games. In this paper, we first refine the complexity result for computing a TME by using a correlation plan to show that a TME can be found in polynomial time in a specific class of games according to our boundary for complexity. Second, we propose an efficient correlation-based algorithm to solve the non-convex program for TME in games not belonging to this class. The algorithm combines two special correlation plans based on McCormick envelopes for convex relaxation and von Stengel-Forges polytope for correlated equilibria. We show that restricting the feasible solution space to von Stengel-Forges polytope will strictly reduce the feasible solution space after convex relaxation of nonlinear terms. Finally, experiments show that our algorithm is about four orders of magnitude faster than the prior state of the art and can solve many previously unsolvable games.

1 Introduction

Designing efficient algorithms by which agents can make decisions in complex, interactive environmnets is an important part of AI [Russell and Norvig, 2016]. Much recent work on the computational study of non-cooperative environments has focused on computing a Nash equilibrium [Nash, 1951] for large zero-sum two-player Extensive-Form Games (EFGs) [Zinkevich *et al.*, 2008; Zhang and Sandholm, 2020; Brown and Sandholm, 2017]. In theory, a Nash equilibrium can be computed by a polynomial-time (in the size of the game) algorithm in zero-sum two-player EFGs [von Stengel, 1996; Shoham and Leyton-Brown, 2008]. In practice, the agents based on algorithms for computing a Nash equilibrium in large zero-sum two-player EFGs have defeated top human players in the heads-up limit Texas hold'em poker game [Bowling *et al.*, 2015] and the heads-up no-limit Texas hold'em poker game [Moravčík *et al.*, 2017; Brown and Sandholm, 2018]. Therefore, in both theory and practice, the problem of computing a Nash equilibrium in zero-sum two-player EFGs is well understood. However, this problem is more complicated in zero-sum multiplayer games, and there are relatively few results. Computing a Nash equilibrium in these games is generally hard [Chen and Deng, 2005], and the strategies between different Nash equilibria are not exchangeable, which impedes players from independently selecting strategies to make them form a Nash equilibrium (i.e., having the equilibrium selection problem among different Nash equilibria) [Brown and Sandholm, 2019].

In this paper, we focus on an alternative solution concept in zero-sum multiplayer EFGs, Team-Maxmin Equilibrium (TME) [von Stengel and Koller, 1997; Basilico *et al.*, 2017; Celli and Gatti, 2018; Zhang and An, 2020a], which is for the situation where several players with the same utility function form a team but take actions against an adversary independently. TME has some nice properties: (i) a TME is a Nash equilibrium that maximizes the utility of each team member among all Nash equilibria, and (ii) TME always exists and is unique and hence avoids the aforementioned equilibrium selection problem. Moreover, TME can capture many strategic interaction scenarios in the real world. For example, in poker games, several players may form a team and play against the target player, i.e., they have the same goal but take actions independently because communications are not allowed in general poker games [Brown and Sandholm, 2019]. In Bridge, two defenders, form a team but play cards independently due to the rules. In the three-player card game DouDizhu (i.e., Fighting the Landlord) [Zha *et al.*, 2021], two peasants form a team to fight against the Landlord. Moreover, in security games, different security departments independently protect major ports such as the New York port [Jiang *et al.*, 2013].

However, it is still very hard (FNP-hard [Celli and Gatti, 2018]) to compute a TME in EFGs. Computing a TME in EFGs can be formulated as a non-convex program. The previous state-of-the-art algorithm for solving this program is IARAMDT [Zhang and An, 2020a], which is based on a global optimization technique — the Multiparametric Disaggregation Technique (MDT) [Kolodziej *et al.*, 2013; Andrade *et al.*, 2019]. The MDT-based algorithms use a digit-wise discretization of continuous variables to approximate nonlinear terms and then add a large number of constraints and integer variables to the program. Our experiments show that MDT-based algorithms are inefficient to compute a close approximation of a TME in large games because they add too many constraints and integer variables to the program. To counter this inefficiency, a correlated variant of TME [Basilico *et al.*, 2017; Zhang *et al.*, 2021] where team players correlate their strategies was proposed as an approximation of TME, because the team's correlated strategy can be transformed into a mixed strategy profile [Basilico *et al.*, 2017]. Correlated TME is relatively easier to compute, but the transformed profile may not form a TME, resulting in an arbitrarily large loss for the team both in theory and practice [Basilico *et al.*, 2017; Zhang and An, 2020a; Zhang and An, 2020b].

Contributions. Our contributions are threefold. *First*, we refine the computational complexity of TME by proposing to use a correlation plan to represent the team's strategy to show that a TME can be found in polynomial time in a specific class of EFGs according to our boundary for complexity. *Second*, we propose an efficient correlation-based algorithm to solve the non-convex program for a TME in other EFGs by combining two special correlation plans based on the well-known McCormick envelope [McCormick, 1976] for convex relaxation and the well-known von Stengel-Forges Polytope [Von Stengel and Forges, 2008] for correlated equilibria: 1) We explore the relation between the proposed correlation plan and both special plans to show that our proposed correlation plan cannot be replaced by them in these EFGs. 2) We show that restricting the feasible solution space to von Stengel-Forges Polytope will strictly reduce the feasible solution space after convex relaxation of nonlinear terms. *Finally*, our experimental results show that our algorithm is about four orders of magnitude faster than baselines and can solve many previously unsolvable games.

2 Preliminaries

The Extensive-Form Game (EFG) [Shoham and Leyton-Brown, 2008; Von Stengel and Forges, 2008; Farina and Sandholm, 2020] is a standard model for games played on game trees (e.g., game trees in Figure 1), which can capture simultaneous and sequential moves. The imperfect-information EFG can also capture private information. An EFG includes a set of players $N = \{1, \dots, n\}$, and $n \geq 3$ for multiplayer games. Each node in the game tree belongs to one player except for chance nodes that are used to model stochastic events in the game (e.g., drawing cards in poker).

The edges leaving a node represent actions that the player at that node can take, and each player i has a set of actions A_i. H_i is a set of nonterminal nodes (i.e., nodes having outgoing edges) for player i. $\psi_i : H_i \to 2^{A_i}$ is the action function assigning a set of possible actions to each nonterminal node of player i. Nodes without outgoing edges are terminal nodes (also called leaves) that represent the end states of a game, and Z is a set of these terminal nodes. Each player i's utility function $u_i : Z \to \mathbb{R}$ assigns a utility to each leaf.

Information sets, partitions of nonterminal nodes, represent private information. Specifically, player i's collection $\mathcal{I}_i$ of non-empty subsets of H_i is a partition of H_i: Each $I_i \in \mathcal{I}_i$ is a set of nodes that player i cannot distinguish, given what they have observed while playing the game. Formally, if $h_i, h'_i \in I_i$, then $\psi_i(h_1) = \psi_i(h_2)$, i.e., the nodes in I_i have the same set of actions. Let $\psi_i(I_i) = \psi_i(h_i)$ for any $h_i \in I_i$ denote the set of available actions in I_i. Without loss of generality, we assume that for each action $a_i \in A_i$ there is a unique $I_i \in \mathcal{I}_i$ such that $a_i \in \psi_i(I_i)$.

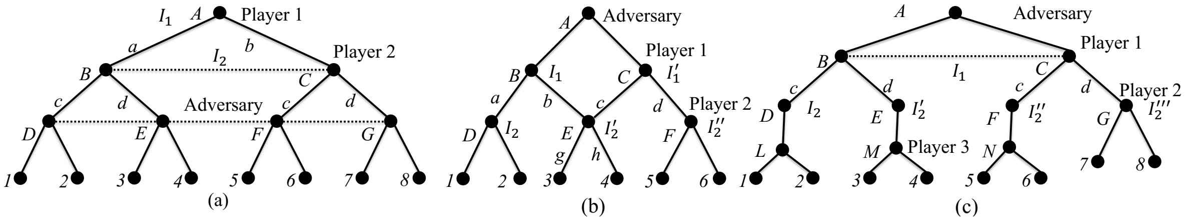
As customary in related works, we focus on perfect-recall games where all players remember the information gained earlier in the game, i.e., for each player i and each $I_i \in \mathcal{I}_i$, nodes in I_i have the same sequence of player i's moves on the paths from the root to these nodes. A sequence σ_i of moves of player i, defined for each node $h \in H \cup Z$ of the game tree, is the ordered set of actions of player i that lie on the path from the root to h. We use $\text{seq}_i(I_i)$ to represent the sequence of player i leading to I_i and use $\sigma_i = \text{seq}_i(I_i)a_i$ to denote that sequence σ_i's last move is a_i taken in information set I_i. In this case, we say σ_i belongs to I_i. $\Sigma_i = \{\text{seq}_i(I_i)a_i \mid I_i \in \mathcal{I}_i, a_i \in \psi_i(I_i)\}$ defines the set of sequences of player i. $\Sigma_{N'} = \times_{i \in N' \subseteq N} \Sigma_i$ defines the combinations of sequences of players in N', and $\sigma_{N'}(j)$ represents the sequence of player j in $\sigma_{N'} \in \Sigma_{N'}$. We write $\sigma = \sigma_N$ and $\Sigma = \Sigma_N$. Each node $h \in H \cup Z$ corresponds to a joint sequences $\sigma_N \in \Sigma_N$ of players, denoted as $\text{seq}_N(h)$, and $\text{seq}_{N'}(h)$ is the joint sequence of players in $N' \subseteq N$ leading to node h. For example, in Figure 1(a), $\text{seq}_{\{1,2\}}(D) = (a, c)$. We use $\Sigma_{N'}(I_i)$ to denote the set of joint sequences of players in N' reaching I_i. ϕ denotes the empty sequence corresponding to the root. A realization plan $r_i : \Sigma_i \to [0, 1]$ is a function assigning a probability to each sequence to be played, which satisfies:

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

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

$$\sum_{a_i \in \psi_i(I_i)} r_i(\text{seq}_i(I_i)a_i) = r_i(\text{seq}_i(I_i)) \;\; \forall I_i \in \mathcal{I}_i \tag{1c}$$

These constraints enforce the property of probability flow for r_i at each information set I_i. Let $\mathcal{R}_i$ be the set of all realization plans, and $\mathcal{R}_{N'} = \times_{i \in N' \subseteq N} \mathcal{R}_i$ define the space of joint realization plans over N'. A *behavioral strategy* β_i of player i defines a probability distribution over $\psi_i(I_i)$ for each information set $I_i \in \mathcal{I}_i$. Every realization plan gives rise to a behavioral strategy by setting: $\forall a_i \in \psi_i(I_i), \beta_i(I_i, a_i) = \frac{r_i(\text{seq}_i(I_i)a_i)}{r_i(\text{seq}_i(I_i))}$ if $r_i(\text{seq}_i(I_i)) > 0$; otherwise $\beta_i(I_i, a_i)$ could be an arbitrary value in $[0, 1]$. A behavioral strategy defines a realization plan as well, i.e., $r_i(\sigma_i) = \prod_{(I_i, a_i) \in \sigma_i} \beta_i(I_i, a_i)$, where $\{(I_i, a_i) \in \sigma_i\}$ represents all information sets and actions along the path (sequence) σ_i of player i.



(a) (b) (c)

Figure 1: Examples of EFGs with three players. In each example, uppercase letters (i.e., $A, B, \cdots$) represent nodes, lowercase letters represent actions, I_i represents the information set of player i, numbers represents leaves, and players 1, 2 (and 3) form a team. For example, in Figure (a), player 1 acts in information set I_1 with node A and two actions a and b, player 2 acts in information set I_2 with nodes B and C and two actions c and d, and the adversary acts in the information set with nodes D–G.

Given player i and player j, $I_i \rightleftharpoons I_j$ with $I_i \in \mathcal{I}_i$ and $I_j \in \mathcal{I}_j$ denotes that I_i and I_j are connected, i.e., there exist $h_i \in I_i$ and $h_j \in I_j$ such that the path from the root of the game tree to h_i passes through h_j or vice versa. For example, $I_1 \rightleftharpoons I_2$ in Figure 1(a), but I_1' and I_2 are not connected in Figure 1(b). Two sequences are *relevant* if any of them is the empty sequence or if both of them belong to connected information sets. That is, given any $\sigma_i = \text{seq}_i(I_i)a_i \in \Sigma_i$ and $\sigma_j = \text{seq}_j(I_j)a_j \in \Sigma_j$, if either σ_i or σ_j is ϕ, or if $I_i \rightleftharpoons I_j$, then σ_i and σ_j are relevant, denoted as $\sigma_i \bowtie \sigma_j$. A joint sequence $\sigma_{N'}$ is *relevant* if any two sequences $\sigma_{N'}(i)$ and $\sigma_{N'}(j)$ in $\sigma_{N'}$ are relevant. The set of relevant joint sequences is $\Sigma_{N'}^{\bowtie}$ for $N' \subseteq N$. For example, in Figure 1(a), $a \bowtie c$, and the joint sequence $\sigma_{\{1,2\}} = (a, c)$ is relevant. Given any player i and a subset of players N' ($i \notin N'$), we say that $\sigma_{N'} \in \Sigma_{N'}$ is relevant for $I_i \in \mathcal{I}_i$, denoted as $I_i \bowtie \sigma_{N'}$, if $\text{seq}_i(I_i) \bowtie \sigma_{N'}(j)$ for each $j \in N'$, and there is at least one node $h_i \in I_i$ such that the joint sequence $\text{seq}_N(h_i)$ includes the sequences of $\text{seq}_i(I_i)$ and $\sigma_{N'}$. For example, in Figure 1(a), the joint sequence $\sigma_{\{1,2\}} = (a, c)$ of players 1 and 2 is relevant for the adversary's unique information set because $\text{seq}_N(D) = (a, c, \phi)$ includes $\sigma_{\{1,2\}}$ and ϕ for the adversary to reach this information set.

Team-Maxmin Equilibrium (TME) [von Stengel and Koller, 1997; Celli and Gatti, 2018; Zhang and An, 2020a] captures scenarios where a team of players $T = \{1, \ldots, n-1\}$ play against the adversary n with $u_i(z) = u_j(z) (\forall i, j \in T)$ and $u_n(z) = -u_T(z) = -\sum_{i \in T} u_i(z)$ for each terminal node z, and team members take actions independently in a zero-sum multiplayer EFG. Here, U_T is the team's utility with $U_T(\sigma) = \sum_{z \in Z'} u_T(z)c(z)$ if a nonempty subset of terminal nodes $Z' \subseteq Z$ are reached by the joint plan σ of all players with the chance $c(z)$ determined by chance nodes, and $U_T(\sigma) = 0$ otherwise. A TME is a Nash equilibrium maximizing each team member's utility and thus always exists and is unique in general [von Stengel and Koller, 1997]. We denote the TME value as the utility of the team under a TME. In an ϵ-TME, both the adversary and the team cannot gain more than ϵ if any of the players deviates from his strategy, and the difference between the TME value and the ϵ-TME value is not larger than ϵ. The team's strategy profile in a TME is defined as $\arg\max_{r_1,\ldots,r_{n-1}} \min_{r_n} \sum_{\sigma = \times_{i \in N} \sigma_i, \sigma \in \Sigma} U_T(\sigma) \prod_{i=1}^{n} r_i(\sigma_i)$ and can be computed by a non-convex maxmin program [Celli and Gatti, 2018], whose objective maximizes the team's utility at the root of the game tree and constraints ensure that the adversary chooses the action minimizing the team's utility (i.e., maximizing the adversary's utility) in each information set of the game tree, similar to the case in two-player games [Shoham and Leyton-Brown, 2008]:

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

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

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

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

where $\mathcal{I}_n(\sigma_n)$ denotes the information set where player n takes the last action in sequence σ_n except for $\mathcal{I}_n(\phi) = \phi$ representing the root, free variable vector $v : \mathcal{I}_n \rightarrow \mathbb{R}$ where $v(I_n)$ is the team's expected utility in information set I_n except that $v(\phi)$ represents the team's expected utility in this EFG. By using the computed strategy profile of the team in a TME, the adversary strategy in a TME can be computed by a linear program through minimizing the team's utility [von Stengel and Koller, 1997; Zhang and An, 2020b]. For simplification, we say a TME is computed by Program (2).

3 Using Correlation Plan to Represent Joint Plan

We now propose an efficient algorithm to compute a TME using a correlation plan of the team to represent the combination of independent plans of team members. By using this representation, we show that a TME can be computed by a linear program with a polynomial number of variables.

In a TME, team members take actions independently, which results in using the non-convex Program (2) to compute the TME. In Eq.(2b), there is a multilinear term $\prod_{i \in T} r_i(\sigma_T(i))$. r_T is a joint realization plan of team members, i.e., the combination of independent realization plans of team members. Motivated by the extensive-form correlation plan in two-player games [Von Stengel and Forges, 2008], we propose our multilinear correlation plan $x : \Sigma_T^{\bowtie} \rightarrow [0, 1]$ to represent the joint plan σ_T of the team such that:

$$x(\sigma_T) = \prod_{i \in T} r_i(\sigma_T(i)) \quad \sigma_T \in \Sigma_T^{\bowtie}. \tag{3}$$

We use $\mathcal{X}$ to denote the space of multilinear correlation plans x. Given a correlation plan x, we can obtain a joint realization plan for all team members as well. To do so, we first obtain a behavior strategy from x. That is, for each team member i, for each $I_i \in \mathcal{I}_i$ with $\text{seq}_i(I_i) = \sigma_i$ and each $a_i \in \psi_i(I_i)$, and given any σ_T and σ'_T with $\sigma_T(i) = \sigma_i$, $\sigma'_T(i) = \sigma_i a_i$, and $\sigma_T(j) = \sigma'_T(j)$ for $i \neq j$,

$$\beta_i(I_i, a_i) = \begin{cases} \frac{x(\sigma'_T)}{x(\sigma_T)} = \frac{r_i(\sigma_i a_i)}{r_i(\sigma_i)} & \text{if } x(\sigma_T) > 0, \\ \frac{1}{|\psi_i(I_i)|} & \text{if } x(\sigma_T) = 0. \end{cases} \tag{4}$$

We can then obtain the corresponding equivalent realization plan r_i from this behavioral strategy β_i obtained from x. Therefore, the correlation plan space is equivalent to the joint realization plan space: for each joint realization plan, there is an equivalent correlation plan and vice versa. Similar to Program (2), the team's strategy in a TME can be defined as:

$$\arg\max_{x \in \mathcal{X}} \min_{r_n \in \mathcal{R}} \sum_{z \in Z} U_T(\text{seq}_N(z)) x(\text{seq}_T(z)) r_n(\text{seq}_n(z)).$$

Through the dual linear program of the inner problem over r_n, we can show that a TME can be computed by a linear program with a polynomial number of variables.

Proposition 1. *A correlation plan x is the team's strategy in a TME if and only if it is a solution of the following linear program: (All proofs are in Supplementary Appendix [1])*

$$\max_x v(\mathcal{I}_n(\phi)) \tag{5a}$$

$$v(\mathcal{I}_n(\sigma_n)) - \sum_{I_n \in \mathcal{I}_n : \text{seq}_n(I_n) = \sigma_n} v(I_n) \tag{5b}$$

$$\leq \sum_{z \in Z, \text{seq}_n(z) = \sigma_n} U_T(\text{seq}_N(z)) x(\text{seq}_T(z)) \quad \forall \sigma_n \in \Sigma_n$$

$$x \in \mathcal{X}. \tag{5c}$$

Program (5) is a linear program whose number of variables is polynomial in the game size. If we can use a polynomial-sized set of linear constraints to represent $\mathcal{X}$, then we can have a polynomial-time algorithm to compute a TME. However, this is impossible in general cases unless P = NP because computing a TME is FNP-hard [Celli and Gatti, 2018].

4 Correlation Plans with Polynomial-Sized Set of Linear Constraints

The preceding section argued that finding a polynomial-sized set of linear constraints to represent $\mathcal{X}$ in Program (5) would lead to a polynomial-time algorithm to compute a TME. In this section, we introduce two special correlation plans that do have a such a polynomial-sized representation. Based on the first special correlation plan, we obtain a boundary between polynomial and FNP instances of games with respect to TME. That is, we provide the first positive result (i.e., a polynomial-time algorithm) for complexity on computing a TME in some games. Moreover, we develop an efficient algorithm to compute a TME in other games by combining both special correlation plans.

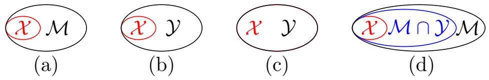


Figure 2: The relation between correlation plan spaces: (a) $\mathcal{X} \subset \mathcal{M}$ in any EFG; (b) $\mathcal{X} \subset \mathcal{Y}$ in any EFG satisfying the non-unique-path property; (c) $\mathcal{X} = \mathcal{Y}$ in any EFG satisfying the unique-path property; and (d) $\mathcal{X} \subset (\mathcal{M} \cap \mathcal{Y}) \subset \mathcal{M}$ in any EFG satisfying the non-unique-path property.

4.1 A Boundary for Complexity

We now show a boundary between the existence and nonexistence of a polynomial-time algorithm to compute a TME based on the von Stengel-Forges Polytope [Von Stengel and Forges, 2008][2].

We propose a special von Stengel-Forges correlation plan with a polynomial-sized set of linear constraints as $y : \Sigma_T^{\bowtie} \to [0, 1]$ that assigns a probability for each relevant joint sequence in multiplayer games. That is, the team's correlation plan satisfies:

$$y(\sigma_T) \geq 0 \quad \forall \sigma_T \in \Sigma_T^{\bowtie} \tag{6a}$$

$$y(\phi_T) = 1 \text{ with } \phi_T(i) = \phi, \forall i \in T \tag{6b}$$

$$\sum_{a_i \in \psi_i(I_i)} y(\sigma_{T \setminus \{i\}}, \text{seq}_i(I_i) a_i) = y(\sigma_{T \setminus \{i\}}, \text{seq}_i(I_i))$$

$$\forall i \in T, I_i \in \mathcal{I}_i, \sigma_{T \setminus \{i\}} \in \Sigma_{T \setminus \{i\}}^{\bowtie}, s.t. I_i \bowtie \sigma_{T \setminus \{i\}} \tag{6c}$$

These constraints will give y the property of the probability flow at each information set: starting with probability 1 at the root (Eq.(6b)), Eq.(6c) ensures that the incoming probability is equal to the outgoing probability at each information set. We denote the space of these plans as $\mathcal{Y}$. We now explore the relation between $\mathcal{X}$ and $\mathcal{Y}$, and find that $\mathcal{Y}$ includes $\mathcal{X}$ in any EFG.

Proposition 2. *$\mathcal{X} \subseteq \mathcal{Y}$ in any EFG.*

We now show the boundary between $\mathcal{X} = \mathcal{Y}$ and $\mathcal{X} \subset \mathcal{Y}$. Note that $\Sigma_T(I_i)$ is the set of joint sequences of team members reaching I_i, which are paths of the team from the root to I_i. We first define the **unique-path property** of an information set of a team member as: there is only one path for the team from the root to this information set, or there is only one action in this information set. Formally, I_i of team member i is unique-path if $|\Sigma_T(I_i)| = 1$ or $|\psi_i(I_i)| = 1$. Conversely, the **non-unique-path property** of an information set of a team member is: there are at least two paths for the team from the root to this information set, and this information set has at least two actions. Formally, I_i of team member i is non-unique-path if $|\Sigma_T(I_i)| \geq 2$ and $|\psi_i(I_i)| \geq 2$. A game satisfies the unique-path property if each information set of each team member satisfies the unique-path property, and a game

[1] https://github.com/Youzhi333/TMEEFG

[2] In two-player EFGs either without chance nodes [Von Stengel and Forges, 2008] or with public chance nodes [Farina and Sandholm, 2020], the von Stengel-Forges Polytope gives rise to a polynomial-time algorithm for correlated equilibrium. However, these results cannot be extended to TME because (as we show later) using the von Stengel-Forges Polytope does not lead to a polynomial-time algorithm for TME even in games without chance nodes, e.g., the game in Figure 1(a) or 1(b).

satisfies the non-unique-path property if at least one information set of any team member satisfies the non-unique-path property. For example, the EFG in Figure 1(a) satisfies the non-unique-path property because $\Sigma_T(I_2) = \{(a, \phi), (b, \phi)\}$ and $\psi_2(I_2) = \{c, d\}$; and the game in 1(b) satisfies the non-unique-path property because of the non-unique-path property of I_2'. However, the EFG in Figure 1(c) satisfies the unique-path property because $\Sigma_T(I_1) = \{(\phi, \phi)\}$, $\Sigma_T(I_2) = \{(c, \phi)\} = \Sigma_T(I_2^{''})$, $\Sigma_T(I_2') = \{(d, \phi)\} = \Sigma_T(I_2^{'''})$, and so on. These properties define the boundary between $\mathcal{X} \subset \mathcal{Y}$ and $\mathcal{X} = \mathcal{Y}$ shown in Figure 2(b) and 2(c).

Proposition 3. *In any EFG satisfying the non-unique-path property, $\mathcal{X}$ is a strict subset of $\mathcal{Y}$, i.e., $\mathcal{X} \subset \mathcal{Y}$.*

We now provide a positive result for some EFGs.

Proposition 4. *In any EFG satisfying the unique-path property, $\mathcal{X} = \mathcal{Y}$.*

Note that the assumption of the unique-path property can be easily verified in any EFG in linear time in the size of the game by checking each information set. The above results define the boundary between the existence and nonexistence of a polynomial-time algorithm for TME. That is, we have the following positive complexity result.

Theorem 1. *In any EFG satisfying the unique-path property, a TME can be computed in polynomial time.*

The EFGs satisfying the unique-path property can model many scenarios. For example, in poker, including any games using cards, a team of players may share card information with each other (at least, according to some movies such as 21 (https://en.wikipedia.org/wiki/21_(2008_film)). More specifically, in an online variant of DouDizhu [Zha *et al.*, 2021], friends can see the information of cards held by friends on the screen to fight against the adversary. Other examples include some variants of real-world security games [Wang *et al.*, 2019; Zhang *et al.*, 2019], where defenders have real-time location information about other players, e.g., through GPS. Such information enables the player to consider how other defenders try to interdict the adversary (e.g. the poacher in the forest or the criminal in the city) on the path. In these games, the unique-path property holds, and team members may use a polynomial-time Program (5) to compute a TME strategy by replacing $\mathcal{X}$ with $\mathcal{Y}$ and then transform it into independent realization plans for team members using Eq.(4).

We understand that the EFGs satisfying the unique-path property are relatively rare, and the EFGs satisfying the non-unique-path property are common. Therefore, we still need efficient algorithms to compute a TME in general EFGs. However, the significance of a boundary between the existence and nonexistence of a polynomial-time algorithm is: 1) it helps us understand the complexity of computing a TME better, i.e., knowing which case is easy and which case is hard; and 2) it helps us develop an efficient algorithm to compute a TME for general EFGs and understand the reason why the algorithm works, which is discussed in Section 4.2.

4.2 Solving General EFGs

By Proposition 3, there are some EFGs where we cannot use $\mathcal{Y}$ to replace $\mathcal{X}$, i.e., we still need to use a global optimization solver to solve the nonlinear program for a TME. To make the nonlinear program efficiently solvable, we propose our correlation-based algorithm by combining $\mathcal{Y}$ and a special correlation plan space implicitly used as the feasible solution space in the global optimization solver. We then theoretically show the efficiency of our correlation-based algorithm by proving that this combination will strictly reduce the feasible solution space in these EFGs.

We now present the special correlation plan based on the well-known McCormick envelopes [McCormick, 1976] for convex relaxation of nonlinear programs. The McCormick envelope is usually used as a basic technique to solve a bilinear program in state-of-the-art non-convex optimization solvers, e.g. Gurobi [Gurobi, 2020]. The McCormick correlation plan $m : \Sigma_T^{\bowtie} \rightarrow [0, 1]$ has a polynomial-sized set of linear constraints and assigns a probability for each relevant joint sequence. The team's plan satisfies the following a polynomial-sized set of linear constraints with $T = \{1, 2\}$:

$$m(\sigma_T) \geq 0 \quad \forall \sigma_T \in \Sigma_T^{\bowtie} \tag{7a}$$

$$m(\sigma_T) \geq r_1(\sigma_T(1)) + r_2(\sigma_T(2)) - 1 \quad \forall \sigma_T \in \Sigma_T^{\bowtie} \tag{7b}$$

$$m(\sigma_T) \leq r_1(\sigma_T(1)) \quad \forall \sigma_T \in \Sigma_T^{\bowtie} \tag{7c}$$

$$m(\sigma_T) \leq r_2(\sigma_T(2)) \quad \forall \sigma_T \in \Sigma_T^{\bowtie} \tag{7d}$$

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

That is, the McCormick correlation plan is bounded according to the realization plan, and Eqs.(7a)-(7d) represent inequalities $r_1(\sigma_1)r_2(\sigma_2) \geq 0, (1 - r_1(\sigma_1))(1 - r_2(\sigma_2)) \geq 0, r_1(\sigma_1)(1 - r_2(\sigma_2)) \geq 0$, and $r_1(\sigma_1)(1 - r_2(\sigma_2)) \geq 0$, respectively, where $r_1(\sigma_1)r_2(\sigma_2)$ is replaced by $m(\sigma_1, \sigma_2)$. We denote this McCormick correlation plan space as $\mathcal{M}$. For multilinear programs, the recursive McCormick relaxation recursively transforms each multilinear term into bilinear terms[3]. Therefore, we only present the McCormick correlation plan for two players as it is easy to extend to the multiplayer setting by recursive McCormick relaxation. We now show that $\mathcal{M}$ strictly includes $\mathcal{X}$, as illustrated in Figure 2(a).

Proposition 5. *$\mathcal{X} \subset \mathcal{M}$ in any EFG.*

Therefore, $\mathcal{M}$ is strictly larger than $\mathcal{X}$, and we cannot use Eq.(7) as the constraints of $\mathcal{X}$, i.e., we cannot have a polynomial-time algorithm for computing a TME in any EFG based on the McCormick correlation plan. However, when we use a global optimization solver, e.g., Gurobi, to solve a nonlinear program, the solver implicitly restricts the feasible solution space after convex relaxation to $\mathcal{M}$ and then searches for the optimal solution. We now propose our correlation-based algorithm for computing a TME to explicitly restrict the feasible solution space to $\mathcal{Y}$ and use nonlinear constraints to represent $\mathcal{X}$ in Program (5):

$$\max_{r_T} v(\mathcal{I}_n(\phi)) \tag{8a}$$

$$\text{Eqs.}(3), (5b), (6) \tag{8b}$$

$$\text{Eqs.}(1a) - (1c) \quad \forall i \in T. \tag{8c}$$

That is, Program (8) is a nonlinear program because of constraints (1a)-(1c) and (3) that represent the correlation plan

[3] E.g., the term $m_1 = r_1(\sigma_1)r_2(\sigma_2)r_3(\sigma_3)$ is transformed into two bilinear terms $m_1 = m_2 r_3(\sigma_3)$ and $m_2 = r_1(\sigma_1)r_2(\sigma_2)$.

	EFG	3K4	3K6	3K8	3K10	3K12	3K13	4K5	4K6*	3L2-3*	3L3-2*
Size	$\|B\|$	132	330	616	990	1452	1716	2200	4170	572	1707
	$\|\Sigma_i\|$	33	49	65	81	97	105	81	97	209	457
Algorithm	**CNLP**	**0.1s**	**0.3s**	**37s**	**47s**	**273s**	**1.5h**	**146s**	**1303s**	**20s**	**141s**
	NLP	>10h	>10h	>10h	>10h	>10h	>10h	>10h	>10h	>10h	>10h
	IARAMDT	0.2s	4.3h	>10h	>10h	>10h	>10h	>10h	>10h	>10h	>10h
	MDT	>10h	>10h	>10h	>10h	>10h	>10h	>10h	>10h	>10h	>10h

Table 1: The runtime for algorithms in different games: $|B|$ represents the number of nonlinear terms, $|\Sigma_i|$ represents the number of sequences of each player in a game, and '>10h' indicates that an algorithm did not reach the given optimization gap within 10 hours. No algorithms can reach the gap 0.0001 within 10h in games with $*$ (i.e., 4K6, 3L2-3, and 3L3-2), where the runtime for the gap 0.3 is reported.

	$\frac{1}{10}$	$\frac{1}{10^2}$	$\frac{1}{10^3}$	$\frac{1}{10^4}$	$\frac{1}{10}$	$\frac{1}{10^2}$	$\frac{1}{10^3}$	$\frac{1}{10^4}$
1	**0.3s**	**0.3s**	**0.3s**	**0.3s**	**8s**	**25s**	**35s**	**37s**
2	10h	10h	10h	10h	10h	10h	10h	10h
3	15s	68s	1.4h	4.3h	10h	10h	10h	10h
4	10h	10h	10h	10h	10h	10h	10h	10h

Table 2: The runtime to reach the gap from 0.1 to 0.0001 in 3K6 and 3K8: Rows 1-4 represent CNLP, NLP, IARAMDT, and MDT, respectively, and 10h represents >10h.

space $\mathcal{X}$. In addition, we restrict the feasible solution space to $\mathcal{Y}$ through Eq.(6) (the variable y is changed to x when Eq.(6) is inserted into Program (8)) in addition to the implicit feasible solution space $\mathcal{M}$. Our experiments show that these constraints will make our program efficiently solvable. We now theoretically show why restricting the feasible solution space to $\mathcal{Y}$ can achieve this good result by showing that this step will strictly reduce the space of $\mathcal{M}$, i.e., the implicit feasible solution space after convex relaxation of nonlinear terms, as shown in Figure 2(d).

Proposition 6. *In any EFG satisfying the non-unique-path property,* $\mathcal{X} \subset (\mathcal{M} \cap \mathcal{Y}) \subset \mathcal{M}$.

$\mathcal{X} \subset (\mathcal{M} \cap \mathcal{Y}) \subset \mathcal{M}$ means that: 1) we cannot simply use $(\mathcal{M} \cap \mathcal{Y})$ to replace $\mathcal{X}$ in Program (5) to compute a TME, i.e., Program (8) with nonlinear constraints to represent $\mathcal{X}$ is necessary; and 2) restricting the feasible solution space to $\mathcal{Y}$ will strictly reduce the space of $\mathcal{M}$, i.e., the feasible solution space after convex relaxation of nonlinear terms. By Propositions 4 and 5, $\mathcal{X} = (\mathcal{M} \cap \mathcal{Y})$, i.e., Program (8) after the convex relaxation is still a polynomial-time algorithm to solve EFGs satisfying the unique-path property. Therefore, Program (8) is efficient to solve general EFGs for TMEs.

5 Experimental Evaluation

We now evaluate the correlation-based Program (8) proposed above. All experiments ran on a machine with an 8-core 2.3GHz CPU and 16GB of RAM. All programs were solved by Gurobi 9.1.[4] We used the default optimization gap in Gurobi, i.e., $\frac{|U-u|}{|u|} = 0.0001$, where U is the upper bound, and u is the lower bound of the objective function produced by the solver while searching for the optimal solution, unless stated otherwise. Codes are available at Footnote 1.

We tested our algorithm on benchmark multiplayer poker games [Zhang *et al.*, 2021]: Kuhn poker and Leduc poker, which satisfy the non-unique-path property. We use nKc to represent an n-player Kuhn game with c different cards and nLk-c to represent an n-player Leduc game with k ranks and c cards for each rank. Generally, the team is formed by players 1 to $n-1$. We use CNLP to represent our correlation-based algorithm (i.e., Program (8)). We evaluate our CNLP against three baselines: 1) NLP: we use Gurobi to solve the nonlinear Program (2) as a baseline; 2) IARAMDT: the prior state-of-the-art algorithm for computing a TME [Zhang and An, 2020a] based on MDT; and 3) MDT: the original MDT algorithm for global optimization [Andrade *et al.*, 2019; Kolodziej *et al.*, 2013]. Note that NLP is equivalent to solving Program (8) within $\mathcal{M}$ that is implicitly used in Gurobi.

Table 1 shows the runtime to compute a TME in different games, while Table 2 provides additional details about the convergence in games 3K6 and 3K8 (similar results were achieved in other games shown in Table 1 as well). Results in Tables 1 and 2 verify that: 1) Our algorithm CNLP dramatically outperforms the baselines. 2) Our algorithm CNLP still significantly outperforms the baselines even when the optimization gap is relaxed, as shown in Table 2. 3) Our algorithm CNLP could be four or five orders faster than the prior state-of-the-art algorithm IARAMDT (see the result for 3K6 in Table 1). 4) Our algorithm CNLP can solve many games (e.g., games 3K8-3K13) that IARAMDT cannot solve. 5) Our algorithm CNLP dramatically outperforms NLP, which highlights the importance of restricting the feasible solution space to $\mathcal{Y}$. 6) IARAMDT dramatically outperforms MDT, which is consistent with previous results [Zhang and An, 2020a].

6 Conclusions

In this paper, we focus on computing a TME in zero-sum multiplayer EFGs. We first propose to use a correlation plan to represent the team's strategy to show that a TME can be found in polynomial time in specific EFGs according to our boundary for complexity. Second, we propose an efficient correlation-based algorithm for TME in other EFGs by combining two special correlation plans for smaller feasible solution space after relaxation. Finally, experimental results show that our algorithm significantly outperforms baselines.

Acknowledgments

We gratefully acknowledge support from ONR grant N00014-20-1-2407. Bo An is supported by the National Re-

[4] For games with multilinear terms, we use the recursive method mentioned in Section 4.2 to transform them into bilinear terms.

search Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG-RP-2019-0013).

References

[Andrade *et al.*, 2019] Tiago Andrade, Fabricio Oliveira, Silvio Hamacher, and Andrew Eberhard. Enhancing the normalized multiparametric disaggregation technique for mixed-integer quadratic programming. *Journal of Global Optimization*, 73(4):701–722, 2019.

[Basilico *et al.*, 2017] Nicola Basilico, Andrea Celli, Giuseppe De Nittis, and Nicola Gatti. Team-maxmin equilibrium: Efficiency bounds and algorithms. In *AAAI*, pages 356–362, 2017.

[Bowling *et al.*, 2015] Michael Bowling, Neil Burch, Michael Johanson, and Oskari Tammelin. Heads-up limit hold'em poker is solved. *Science*, 347(6218):145–149, 2015.

[Brown and Sandholm, 2017] Noam Brown and Tuomas Sandholm. Safe and nested subgame solving for imperfect-information games. In *NeurIPS*, 2017.

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

[Brown and Sandholm, 2019] Noam Brown and Tuomas Sandholm. Superhuman AI for multiplayer poker. *Science*, 365(6456):885–890, 2019.

[Celli and Gatti, 2018] Andrea Celli and Nicola Gatti. Computational results for extensive-form adversarial team games. In *AAAI*, pages 965–972, 2018.

[Chen and Deng, 2005] Xi Chen and Xiaotie Deng. 3-Nash is PPAD-complete. In *Electronic Colloquium on Computational Complexity*, volume 134, pages 2–29, 2005.

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

[Gurobi, 2020] Gurobi. Non-convex quadratic optimization. https://pages.gurobi.com/rs/181-ZYS-005/images/2020-01-14_Non%20Convex%20Quadratic%20Optimization%20in%20Gurobi%209.0%20Webinar.pdf, 2020. Accessed: 2021-08-02.

[Jiang *et al.*, 2013] Albert Xin Jiang, Ariel D Procaccia, Yundi Qian, Nisarg Shah, and Milind Tambe. Defender (mis)coordination in security games. In *IJCAI*, pages 220–226, 2013.

[Kolodziej *et al.*, 2013] Scott Kolodziej, Pedro M Castro, and Ignacio E Grossmann. Global optimization of bilinear programs with a multiparametric disaggregation technique. *Journal of Global Optimization*, 57(4):1039–1063, 2013.

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

[Moravčík *et al.*, 2017] Matěj Moravčík, Martin Schmid, Neil Burch, Viliam Lisý, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. DeepStack: Expert-level artificial intelligence in no-limit poker. *Science*, 356:508–513, 2017.

[Nash, 1951] John Nash. Non-cooperative games. *Annals of Mathematics*, pages 286–295, 1951.

[Russell and Norvig, 2016] Stuart J Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Malaysia; Pearson Education Limited, 2016.

[Shoham and Leyton-Brown, 2008] Yoav Shoham and Kevin Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2008.

[Von Stengel and Forges, 2008] Bernhard Von Stengel and Françoise Forges. Extensive-form correlated equilibrium: Definition and computational complexity. *Mathematics of Operations Research*, 33(4):1002–1022, 2008.

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

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

[Wang *et al.*, 2019] Yufei Wang, Zheyuan Ryan Shi, Lantao Yu, Yi Wu, Rohit Singh, Lucas Joppa, and Fei Fang. Deep reinforcement learning for green security games with real-time information. In *AAAI*, volume 33, pages 1401–1408, 2019.

[Zha *et al.*, 2021] Daochen Zha, Jingru Xie, Wenye Ma, Sheng Zhang, Xiangru Lian, Xia Hu, and Ji Liu. DouZero: Mastering DouDizhu with self-play deep reinforcement learning. In *ICML*, 2021.

[Zhang and An, 2020a] Youzhi Zhang and Bo An. Computing team-maxmin equilibria in zero-sum multiplayer extensive-form games. In *AAAI*, volume 34, pages 2318–2325, 2020.

[Zhang and An, 2020b] Youzhi Zhang and Bo An. Converging to team-maxmin equilibria in zero-sum multiplayer games. In *ICML*, pages 11033–11043, 2020.

[Zhang and Sandholm, 2020] Brian Zhang and Tuomas Sandholm. Sparsified linear programming for zero-sum equilibrium finding. In *ICML*, pages 11256–11267, 2020.

[Zhang *et al.*, 2019] Youzhi Zhang, Qingyu Guo, Bo An, Long Tran-Thanh, and Nicholas R Jennings. Optimal interdiction of urban criminals with the aid of real-time information. In *AAAI*, volume 33, pages 1262–1269, 2019.

[Zhang *et al.*, 2021] Youzhi Zhang, Bo An, and Jakub Černỳ. Computing ex ante coordinated team-maxmin equilibria in zero-sum multiplayer extensive-form games. In *AAAI*, volume 35, pages 5813–5821, 2021.

[Zinkevich *et al.*, 2008] Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. Regret minimization in games with incomplete information. In *NeurIPS*, pages 1729–1736, 2008.