Multilinear Games

Hau Chan[1], Albert Xin Jiang[1], Kevin Leyton-Brown[2], and Ruta Mehta[3]

[1] Dept. of Computer Science, Trinity University. {hchan,xjiang}@trinity.edu
[2] Dept. of Computer Science, U. of British Columbia. kevinlb@cs.ubc.ca
[3] Dept. of Computer Science, UIUC. rutamehta@cs.illinois.edu

Abstract. In many games, players' decisions consist of multiple subdecisions, and hence can give rise to an exponential number of pure strategies. However, this set of pure strategies is often structured, allowing it to be represented compactly, as in network congestion games, security games, and extensive form games. Reduction to the standard normal form generally introduces exponential blow-up in the strategy space and therefore are inefficient for computation purposes. Although individual classes of such games have been studied, there currently exists no general purpose algorithms for computing solutions such as Nash equilibrium (NE) and (coarse) correlated equilibrium.

To address this, we define *multilinear games* generalizing all. Informally, a game is multilinear if its utility functions are linear in each player's strategy, while fixing other players' strategies. Thus, if pure strategies, even if they are exponentially many, are vectors in polynomial dimension, then we show that mixed-strategies have an equivalent representation in terms of *marginals* forming a polytope in polynomial dimension.

The canonical representation for multilinear games can still be exponential in the number of players, a typical obstacle in multi-player games. Therefore, it is necessary to assume additional structure that allows computation of certain sub-problems in polynomial time. Towards this, we identify two key subproblems: computation of *utility gradients*, and optimizing linear functions over strategy polytope. Given a multilinear game, with polynomial time subroutines for these two tasks, we show the following: (a) We can construct a polynomially-computable and polynomially-continuous fixed-point formulation, and show that its approximate fixed-points are approximate NE. This gives containment of approximate NE computation in PPAD, and settles its complexity to PPAD-complete. (b) Even though a coarse correlated equilibrium can potentially have exponential representation (being a probability distribution of pure strategy profiles), through LP duality and a carefully designed separation oracle, we provide a polynomial-time algorithm to compute one with polynomial representation. (c) Finally, we show existence of an approximate NE with support-size logarithmic in the strategy polytope dimensions.

1 Introduction

The computation of game-theoretic solution concepts is a central problem at the intersection of game theory and computer science. For games with large numbers of players, the standard normal form game representation requires exponential space even if the number of strategies per players is two, and is thus not a practical option as a basis for computation. Fortunately, most such games of practical interest have highly structured *utility functions*, and thus it is possible to represent them compactly. A line of research thus exists to look for *compact game representations* that are able to succinctly describe structured games, including work on graphical games [17], multi-agent influence diagrams [19] and action-graph games [16].

On the other hand, in many real-world domains, each player needs to make a decision that consists of multiple sub-decisions (e.g., assigning a set of resources, ranking a set of options, or finding a path in a network), and hence the number of pure strategies per player itself can be exponential. The single-player versions of these decision problems have been well studied in the field of combinatorial optimization, with mature general modeling languages such as AMPL and solvers like CPLEX. For the multi-player case, several classes of games studied in the recent literature have structured strategy spaces, including network congestion games [5, 7], simultaneous auctions and other multi-item auctions [32, 29], dueling algorithms [13], integer programming games [20], Blotto games [1], and security games [21, 31]. These papers proposed compact game representations suitable for their specific domains, and corresponding algorithms for computing solution concepts, which take advantage of the specific structure in the representations. However, it is not obvious whether algorithmic techniques developed for one domain can be transferred to another.

One general approach that has been quite successful in the study of efficient computation for compact representations is the following: identify subtasks that are required for most existing algorithms of these solution concepts, and then speed up these subtasks by exploiting the structure of the compact representation. In particular, [7, 26] identified *expected utility computation* given a mixed strategy as the subtask to compute correlated equilibrium efficiently, and to show NE computation is in PPAD. Furthermore, they showed that games like graphical, polymatrix, and symmetric, this subtask can be done in polynomial time. However, a crucial assumption behind these results is *polynomial type*: roughly, it is feasible to enumerate pure strategies of all the players. This is not the case for games with structured strategies, in which such explicit strategy enumeration can take exponential time. [7] showed PPAD membership of NE computation for two additional subclasses: network congestion and extensive form games, but the general case remained open. A related obstacle is that even specifying a mixed strategy as a distribution over pure strategies can require exponential space.

In this paper, we present a unified algorithmic framework for games with structured strategy spaces, even when the number of pure strategies is exponential. We focus on games with polytopal strategy spaces, in which each player's set of pure strategies is defined to be integer points in a polytope. We summarize our contributions as follows.

1. We identify *multilinearity* as an important property of games that enables us to represent the players' mixed strategies compactly. Informally, a game is multilinear if its utility functions are linear in each player's strategy, while

fixing other players' strategies. We show that many existing game forms, like Bayesian, congestion, security, etc., are multilinear (see Appendix A).
2. The canonical representation of multilinear games still grows exponentially in the number of players. Therefore, it is necessary to assume additional structure that allows some computation in polynomial time, like done in [7, 26]. Towards this, we identify two key subproblems: computation of *utility gradients*, and optimizing linear functions over strategy polytopes. Given a multilinear game, with polynomial time subroutines for these two tasks, we show the following:
   (a) computing an approximate Nash equilibrium is in PPAD.
   (b) a coarse correlated equilibrium can be computed in polynomial time.
   These results are generalizations of [7] and [26], respectively, from games of polynomial type to multilinear games.
3. We prove that given a multilinear game, there exists an approximate NE with support-size logarithmic in the strategy polytope dimensions. This generalizes [2], which gave bounds logarithmic in the number of strategies.

1.1 Technical Overview

Our approach is based on a compact representation of mixed strategies as *marginal vectors*, which is a point in the strategy polytope induced by the mixed strategy distributions. When the game is multilinear, all mixed strategies with the same marginal vector are payoff-equivalent (Lemmas 1 and 2), and therefore we can work in the space of marginal vectors instead of the exponentially higher-dimensional space of mixed strategies.[4] Next we adapt existing algorithmic approaches such that whenever the algorithm calls for enumeration of pure strategies (e.g., for computing a best response), we instead solve a linear optimization problem in the space of marginal vectors, which can in turn be reduced to the two subproblems, namely computation of utility gradient given a marginal strategy profile, and optimizing a linear function over the polytope of marginal strategies. Assuming polynomial-time procedure for these two, we show a number of computational results.

Next we analyze complexity of computing an equilibrium. Since normal-form games are subcase of multilinear games, irrationality of NE [24], and PPAD-hardness for NE computation [8, 6] follows. Due to exponentially many pure strategies per player, containment of approximate NE computation in PPAD does not carry forward to multilinear games. Towards this, we design a fixed-point formulation to capture NE in marginal profiles, and show that corresponding approximate fixed-points exactly capture approximate NE. Furthermore, we show polynomial-continuity and polynomial-computability (see the Appendix or [9] for definitions) of the function by finding its equivalent representation in terms of projection operator, and obtaining a convex quadratic formulation

[4] This technique has been used extensively in the study of subclasses of games, including extensive-form games (sequence form), dueling algorithms[13], and security games. We provide a unified treatment and identify multilinearity as the key property that enables this technique.

for function evaluation, respectively. Finally, due to a result of [9], all of these together implies containment of finding an approximate NE in PPAD for multilinear games. This provides hope of extending algorithms and heuristics for PPAD problems to multilinear games.

For computing CCE (Theorem 2), we adapt the Ellipsoid Against Hope approach of [26] and its refinement [15]. Applied directly to our setting, this approach would involve running the ellipsoid method in a space whose dimension is roughly the total number of pure strategies of all the players, yielding an exponential-time algorithm. We instead use a related but different convex programming formulation, and then (through use of the multilinear property) transform it into a linear program of polynomial number of variables, which is then amenable to the ellipsoid method. Although the final output is not in terms of mixed strategies or marginal vectors (instead it is a correlated distribution with small support), a crucial intermediate step (the separation oracle of the ellipsoid method) requires linear optimization over the space of marginal vectors.

Finally, we show existence of approximate NE with logarithmic support using the probabilistic method, together with applying concentration inequalities on marginals to avoid union bound on exponentially many terms (Theorem 4).

Due to space constraint next we give an overview of our results, while all the proofs and some of the details may be found in the Appendix.

2 Preliminaries

Notations. We use boldface letters, like $\boldsymbol{x}$, to denote vectors, and x_i to denote its i^{th} coordinate. To denote the set of $\{1, \ldots, m\}$ we use $[m]$. We use $\mathbb{Z}_+$ and $\mathbb{R}_+$ to denote the sets of non-negative integers and reals, respectively.

A game is specified by (N, S, u), where $N = \{1, \ldots, n\}$ is the set of players. Each player $i \in N$ chooses from a finite set of pure strategies S_i. Denote by $s_i \in S_i$ a pure strategy of player i. Then $S = \prod_i S_i$ is the set of pure-strategy profiles. Moreover, $u = (u_1, \ldots, u_n)$ are the utility functions of the players, where the utility function of player i is $u_i : S \to \mathbb{R}$.

In normal-form games, strategy sets S_is and utility functions u_is are specified explicitly. Thus, the size of the representation is of the order of $n|S| = n \prod_i |S_i|$.

A mixed strategy $\boldsymbol{\sigma}_i$ of player i is a probability distribution over her pure strategies. Let $\Sigma_i = \Delta(S_i)$ be i's set of mixed strategies, where $\Delta(\cdot)$ denotes the set of probability distributions over a finite set. Denote by $\boldsymbol{\sigma} = (\boldsymbol{\sigma}_1, \ldots, \boldsymbol{\sigma}_n)$ a mixed strategy profile, and $\Sigma = \prod_i \Sigma_i$ the set of mixed strategy profiles. Denote by $\boldsymbol{\sigma}_{-i}$ the mixed strategy profile of players other than i. $\boldsymbol{\sigma}$ induces a probability distribution over pure strategy profiles. Denote by $u_i(\boldsymbol{\sigma})$ the expected utility of player i under $\boldsymbol{\sigma}$:

$$u_i(\boldsymbol{\sigma}) = E_{\boldsymbol{s} \sim \boldsymbol{\sigma}}[u_i(\boldsymbol{s})] = \sum_{\boldsymbol{s} \in S} u_i(\boldsymbol{s}) \prod_{k \in N} \boldsymbol{\sigma}_k(s_k),$$

where $\boldsymbol{\sigma}_k(s_k)$ is player k's probability of playing the pure strategy s_k.

Nash equilibrium (NE). Player i's strategy $\boldsymbol{\sigma}_i$ is a best response to $\boldsymbol{\sigma}_{-i}$ if $\boldsymbol{\sigma}_i \in \arg\max_{\boldsymbol{\sigma}'_i \in \Sigma_i} u_i(\boldsymbol{\sigma}'_i, \boldsymbol{\sigma}_{-i})$. A mixed strategy profile $\boldsymbol{\sigma}$ is a Nash equilibrium if for each player $i \in N$, $\boldsymbol{\sigma}_i$ is a best response to $\boldsymbol{\sigma}_{-i}$.

Another important solution concept is Coarse Correlated Equilibrium (CCE). Consider a distribution over the set of pure-strategy profiles. This can be represented by a vector $\boldsymbol{x}$, satisfying $\boldsymbol{x} \geq 0$, $\sum_{\boldsymbol{s} \in S} x_{\boldsymbol{s}} = 1$. The expected utility for player i under $\boldsymbol{x}$ is $u_i(\boldsymbol{x}) = \sum_{\boldsymbol{s} \in S} x_{\boldsymbol{s}} u_i(\boldsymbol{s})$. Given $\boldsymbol{x}$, the expected utility for player i if he deviates to strategy s_i is: $u_i^{s_i}(\boldsymbol{x}) = \sum_{\boldsymbol{s}_{-i}} x_{\boldsymbol{s}_{-i}} u_i(s_i, \boldsymbol{s}_{-i})$, where $x_{\boldsymbol{s}_{-i}} = \sum_{s_i \in S_i} x_{(s_i, \boldsymbol{s}_{-i})}$ is the marginal probability of $\boldsymbol{s}_{-i}$ in distribution $\boldsymbol{x}$. Let

$$g_i(\boldsymbol{x}) = \max_{s_i \in S_i} u_i^{s_i}(\boldsymbol{x}), \tag{1}$$

i.e. player i's expected utility if he deviates to a best response against $\boldsymbol{x}$.

Definition 1. *A distribution $\boldsymbol{x}$ satisfying $\boldsymbol{x} \geq 0$, $\sum_{\boldsymbol{s} \in S} x_{\boldsymbol{s}} = 1$ is a Coarse Correlated Equilibrium (CCE) if it satisfies the following* incentive constraints*:*

$$u_i(\boldsymbol{x}) \geq g_i(\boldsymbol{x}), \quad \forall i \in N.$$

A *rational polytope* is defined by a set of inequalities with integer coefficients; formally $P = \{\boldsymbol{x} \in \mathbb{R}^m | D\boldsymbol{x} \leq \boldsymbol{f}\}$ is a rational polytope if matrix D and vector $\boldsymbol{f}$ consist of integers.

3 Multilinear Games

3.1 Polytopal Strategy Space

We are interested in games in which a pure strategy has multiple components. Without loss of generality, if each pure strategy of player i has m_i components, we can associate each such pure strategy with an m_i-dimensional nonnegative integer vector. Then the set of pure strategies for each player i is $S_i \subset \mathbb{Z}_+^{m_i}$. In general the number of integer points in S_i can grow exponentially in m_i. Thus, we need a compact representation of S_i.

In most studies of games with structured strategy spaces, each S_i can be expressed as the set of integer points in a rational polytope $P_i \subset \mathbb{R}_+^{m_i}$, i.e., $S_i = P_i \cap \mathbb{Z}_+^{m_i}$. We call such an S_i a *polytopal pure strategy set*. We assume P_i is nonempty, bounded and contained in the nonnegative quadrant $R_+^{m_i}$. Then, in order to represent the strategy space, we only need to specify the set of linear constraints defining $P_i = \{\boldsymbol{p} \in \mathbb{R}_+^{m_i} \mid D_i \boldsymbol{p} \leq \boldsymbol{f}_i\}$, with each linear constraint requiring us to store $O(m_i)$ integers.[5] We call a game with this property *a game with polytopal strategy spaces.*

For example, one common scenario is when there are k finite sets $S_i^1, \ldots S_i^k$, and player i needs to simultaneously select one action in each of these sets.

[5] We also allow the possibility that P_i requires an exponential number of linear constraints to specify, but it is represented in another compact way. We discuss properties of P_i that enable efficient computation in Section 4.2.

This happens in Bayesian games in which a player needs to choose an action for each of his type, extensive form games in which a player needs to choose an action in each information set, and simultaneous auctions, among others. The player's pure strategy set S_i is a polytopal strategy space with P_i being the product of k simplices. Second common type of strategy set is a uniform matroid: given a universe $[m_i]$, player i's pure strategy is a subset of size k. This can (e.g.) represent security scenarios in which a defender player i in charge of protecting m_i target, but due to limited resources can only cover k of targets [21]. Then player i's strategy can be represented as the 0–1 vector encoding the subset, and the strategy set can be represented as a polytopal strategy set with $P_i = \{\boldsymbol{p} \in \mathbb{R}^{m_i} | \sum_{j\in[m_i]} p_j = k\}$. Third common type of strategy is to select a path in a network, from a given source to a given destination. This can model routing of data traffic in an network congestion game, or patrol / attack routes in security settings [14, 33]. Here, s_i can be modeled as a 0–1 vector specifying the subset of edges forming the chosen path. S_i can be represented as a polytopal strategy space, where P_i consists of a set of flow constraints, as in [7].

3.2 Mixed Strategies and Multilinearity

In this paper, we are focusing on computation of solution concepts in which players are playing mixed strategies, such as Nash equilibrium. The first challenge we face is the representation of mixed strategies. Recall that a mixed strategy $\boldsymbol{\sigma}_i$ of player i is a probability distribution over the set of pure strategies S_i. When $|S_i|$ is exponential, representing $\boldsymbol{\sigma}_i$ explicitly would take exponential space. Thus we would like a compact representation of mixed strategies, i.e., a way to represent a mixed strategy using only polynomial number of bits. One approach would be to only use mixed strategies of polynomial-sized *support*, where support is the set of pure strategies played with non-zero probability. Such strategies can be stored as sparse vectors requiring polynomial space; however, the space of small-support mixed strategies is not convex, and this is problematic for computation.

We list a set of desirable features for a compact representation of mixed strategies: (1) the expected utilities of the game can be expressed in terms of this compact representation; (2) the space of the resulting compactly-represented strategies is convex; (3) given this compact representation, we can efficiently recover a mixed strategy (e.g., as a mixture over a small number of pure strategies, or by providing a way to efficiently sample pure strategies from the mixed strategy). We show that such a compact representation is possible if the game is *multilinear.*

Definition 2. *Consider a game Γ with polytopal strategy sets, with $S_i = P_i \cap \mathbb{Z}_+^{m_i}$ for each player i. Γ is a multilinear game if*

1. *for each player i, there exists $U^i \in \mathbb{R}^{\prod_{k\in N} m_k}$ such that for all $\boldsymbol{s} \in S$,*

$$u_i(\boldsymbol{s}) = \sum_{(j_1 \ldots j_n)\in\prod_k [m_k]} U^i_{j_1\ldots j_n} \prod_{k\in N} s_{k,j_k},$$

where $[m_k] = \{1, \ldots, m_k\}$;

2. *The extreme points (i.e. vertices) of P_i are integer vectors, which implies that $P_i = conv(S_i)$, where $conv(S_i)$ is the convex hull of S_i.*

In particular, given a fixed $\boldsymbol{s}_{-j}$, u_i is a linear function of $\boldsymbol{s}_j$. In other words, a multilinear game's utility functions are multilinear in the players' strategies.

Condition 2 of Definition 2 is satisfied if P_i's constraint matrix D_i is *totally unimodular*. Total unimodularity is a well-studied property satisfied by the constraint matrices of many polytopal strategy spaces studied in the literature, including the network flow constraint matrix of network congestion games, the uniform matroid constraints of security games [21], and the doubly-stochastic constraints representing rankings in the search engine ranking duel [13]. When Condition 2 is not satisfied, we can redefine P_i to be $conv(S_i)$, but the new P_i may have exponentially more constraints. Indeed, dropping Condition 2 would allow us to express various NP-hard single-agent combinatorial optimization problems (e.g. set cover, knapsack). Examples 1, 2 and 3 in Appendix A demonstrates how security, congestion, extensive-form, and Bayesian games are multilinear.

Remark 1. The utility functions of a multilinear game can be represented by $\{U^i\}$, with space complexity to the order of $n \prod_{k \in N} m_k$. This is more compact than the normal form but still exponential in n. We are not proposing multilinear games as a concrete compact representation; we are interested in multilinearity as a desirable property for all compact games because of its implications for efficient computation.

Given a mixed strategy $\boldsymbol{\sigma}_i$, define the *marginal vector* $\boldsymbol{\pi}_i$ corresponding to $\boldsymbol{\sigma}_i$ as the expectation over the pure strategy space S_i induced by the distribution $\boldsymbol{\sigma}_i$, i.e., $\boldsymbol{\pi}_i = E_{\boldsymbol{\sigma}_i}[\boldsymbol{s}_i] = \sum_{\boldsymbol{s}_i \in S_i} \boldsymbol{\sigma}_i(\boldsymbol{s}_i)\boldsymbol{s}_i$. Denote by π_{ij} the $j-th$ component of $\boldsymbol{\pi}_i$. The set of marginal vectors is exactly $conv(S_i) = P_i$. Given a mixed strategy profile $\boldsymbol{\sigma}$, we call the corresponding collection of marginal vectors $\boldsymbol{\pi} = (\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_n) \in P = \times_i P_i$ the *marginal strategy profile*. By slight abuse of notation let us denote by

$$u_i(\boldsymbol{\pi}) = \sum_{(j_1 \ldots j_n) \in \prod_k [m_k]} U^i_{j_1 \ldots j_n} \prod_{k \in N} \pi_{k,j_k} \tag{2}$$

player i's expected utility under marginal strategy profile $\boldsymbol{\pi}$.

Lemma 1. *Given a mixed strategy profile $\boldsymbol{\sigma} \in \Sigma$ and a marginal vector $\boldsymbol{\pi} \in P$, if $\forall i,\ \boldsymbol{\pi}_i = \sum_{\boldsymbol{s}_i \in S_i} \boldsymbol{\sigma}_i(\boldsymbol{s}_i)\boldsymbol{s}_i$ then $\forall i,\ u_i(\boldsymbol{\sigma}) = u_i(\boldsymbol{\pi})$.*

That is, marginal vectors capture all payoff-relevant information about mixed strategies, and thus we can use them to compactly represent the space of mixed strategies. We note that this property does not hold for arbitrary games.

Suppose a mixed strategy profile $\boldsymbol{\sigma}$ with marginals $\boldsymbol{\pi} = (\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_n)$ is a Nash equilibrium of a multilinear game. By multilinearity any mixed strategy profile having the same marginals are payoff-equivalent to $\boldsymbol{\sigma}$, and therefore also a Nash equilibrium. Let us define Nash equilibrium in terms of marginals:

Marginal NE. $\boldsymbol{\pi} \in P$ is a *marginal NE* iff $\forall i,\ u_i(\boldsymbol{\pi}) \geq u_i(\boldsymbol{\pi}'_i, \boldsymbol{\pi}_{-i}),\ \forall \boldsymbol{\pi}'_i \in P_i$.

The next lemma follows easily using Lemma 1, and the fact that any vector $\boldsymbol{\pi}_i \in P_i$ can be represented as a convex combination of extreme points of P_i, and extreme points of P_i are in S_i.

Lemma 2. *A mixed-strategy profile $\boldsymbol{\sigma} \in \Sigma$ is a NE iff corresponding marginal strategy profile $\boldsymbol{\pi} \in P$, where $\boldsymbol{\pi}_i = \sum_{\boldsymbol{s}_i \in S_i} \boldsymbol{\sigma}_i(\boldsymbol{s}_i)\boldsymbol{s}_i,\ \forall i \in N$, is a marginal NE.*

The existence of marginal NE follows from the existence of Nash equilibrium in mixed strategies.

4 Computation with Multilinear Games

We now show that many algorithmic results for computing various solutions for normal form games and other game representations of polynomial type can be adapted to multilinear games, with strategies represented as marginals. We follow a "modular" approach, similar to [7, 26]'s treatment of computation of Nash equilibrium and correlated equilibrium in games of polynomial type: we first identify certain key subproblems, then develop general algorithmic results assuming these subproblems can be efficiently computed (likely by exploiting specific structure of the representation). We note that a wide variety of games do has such specific structure (see Appendix A).

4.1 Utility Gradient

Recall that we can express the expected utilities of players using marginal vectors by Equation (2) (Lemma 1). However, a direct computation of expected utility using (2) would require summing over a number of terms exponential in n. Furthermore, computing expected utilities may not be enough: consider the task of determining if a mixed strategy profile (as marginals) is a Nash equilibrium. One needs to compute the expected utility for each pure strategy deviation of i in order to verify that i is playing a best response, but that would require enumerating all pure strategies. Instead, we identify a related but different computational problem as the key subtask for equilibrium computation for multilinear games.

Due to multilinearity, after fixing the strategies of players $N \setminus \{k\}$, $u_i(\boldsymbol{\pi})$ is a linear function of $\pi_{k1}, \ldots, \pi_{km_k}$. We define the *utility gradient* of player i with respect to player k's marginal, $\nabla_k(u_i(\boldsymbol{\pi}_{-k})) \in \mathbb{R}^{m_k}$, to be the vector of coefficients of this linear function. Formally, $\forall j_k \in [m_k]$,

$$(\nabla_k u_i(\boldsymbol{\pi}_{-i}))_{j_k} \equiv \sum_{(j_1,..,j_{k-1},j_{k+1},..,j_n) \in \prod_{\ell=1}^{N\setminus\{k\}}[m_\ell]} U^i_{j_1 \ldots j_n} \prod_{\ell \in N\setminus\{k\}} \pi_{\ell,j_\ell}.$$

Intuitively, $(\nabla_k u_i(\boldsymbol{\pi}_{-i}))_{j_k}$ is the rate of change to i's utility when player k increases it's marginal in dimension j_k.

Problem 1 (UtilGradient). Given a compactly represented game that satisfies multilinearity, given players $i, k \in N$, and $\boldsymbol{\pi}_{-k}$, compute $\nabla_k(u_i(\boldsymbol{\pi}_{-k}))$.

Consider the problem of computing the utility gradients. As with expected utility computation, direct summation would require time exponential in n. Nevertheless, with a compact game representation this problem could be solved in polynomial time. Example 4 in Appendix A shows computation of UtilGradient for network congestion games.

4.2 PolytopeSolve and Decomposing Marginals

The other key subproblem we identify, PolytopeSolve, is the optimization of an arbitrary linear objective in each player's strategy polytope.

Problem 2 (PolytopeSolve). Given a compactly represented game with polytopal strategy space, player i, and a vector $\boldsymbol{d} \in \mathbb{R}^{m_i}$, compute $\arg\max_{\boldsymbol{x} \in P_i} \boldsymbol{d}^T \boldsymbol{x}$.

To motivate this problem, let us consider the issue of constructing a mixed strategy given a marginal vector. First of all, since we have assumed that the extreme points of the polytope P_i are integer points, and thus $P_i = conv(S_i)$, this becomes the problem of describing a point in a polytope by a convex combination of extreme points of the polytope. By Caratheodory theorem, given $\boldsymbol{\pi}_i \in \mathbb{R}^{m_i}$ there exists a mixed strategy of support size at most $(m_i + 1)$ that matches the marginals. There has been existing work that provides efficient constructions for different types of polytopes, including the Birkhoff-von Neumann theorem and its generalizations [4]. The most general result by Grostchel, Lovasz and Schrijver [12] reduces the problem to the task of optimizing an arbitrary linear objective over the polytope, i.e., PolytopeSolve.

Theorem 1 (Grostchel, Lovasz and Schrijver [12]). *Suppose the PolytopeSolve can be solved in polynomial time. Then, the following problem DECOMPOSE(P_i) can be solved in polynomial time: Given $\boldsymbol{\pi}_i \in P_i$, find a polynomial number of extreme points of P_i (i.e., pure strategies) $\boldsymbol{s}_i^1, \ldots \boldsymbol{s}_i^K \in S_i$ and weights $\lambda_1, \ldots, \lambda_K \geq 0$ such that $\sum_{k=1}^K \lambda_k = 1$ and $\boldsymbol{\pi}_i = \sum_{k=1}^K \lambda_k \boldsymbol{s}_i^k$.*

We note that the computational complexity of PolytopeSolve depends only on the strategy polytopes P_is of the game, and not on the utility functions. PolytopeSolve can be definitely solved in polynomial time by linear programming if P_i is given by a polynomial number of linear constraints; this holds for all examples we discussed in this paper. Since the objective is linear, $\arg\max_{\boldsymbol{x} \in P_i} \boldsymbol{d}^T \boldsymbol{x} = \arg\max_{\boldsymbol{x} \in S_i} \boldsymbol{d}^T \boldsymbol{x}$, i.e., we can alternatively solve the optimization problem over S_i, which may be more amenable to combinatorial methods.

For the case when P_i has exponentially many constraints, Grostchel, Lovasz and Schrijver [12] also showed that PolytopeSolve is equivalent to the SEPARATION problem (also known as a *separation oracle*): Given a vector $\boldsymbol{\pi}_i \in \mathbb{R}^{m_i}$, either answers that $\boldsymbol{\pi}_i \in P_i$, or produces a hyperplane that separates $\boldsymbol{\pi}_i$ and P_i (e.g., a constraint of P_i violated by $\boldsymbol{\pi}_i$).

4.3 Best Response

We observe that by construction, $u_i(\boldsymbol{\pi}) = \boldsymbol{\pi}_i^T \nabla_i u_i(\boldsymbol{\pi}_{-i})$. Then given $\boldsymbol{\pi}$, the best response for player i is the solution of the following optimization

$$\begin{aligned} \text{maximize } & \boldsymbol{\pi}_i^T \nabla_i u_i(\boldsymbol{\pi}_{-i}) \\ \text{subject to } & \boldsymbol{\pi}_i \in P_i \end{aligned}$$

This is a linear program with feasible region P_i, which is an instance of the problem PolytopeSolve. The coefficients of the linear objective are exactly the utility gradient $\nabla_i u_i(\boldsymbol{\pi}_{-i})$.

Proposition 1. *Suppose we have a compact game representation with polynomial-time procedures for both UtilGradient and PolytopeSolve. Then the best response problem can be computed in polynomial time.*

As a corollary, under the same assumptions, we get that checking if a given profile $\boldsymbol{\pi}$ is a Nash equilibrium can be done in polynomial time.

4.4 Computing Coarse Correlated Equilibrium

Another important solution concept is Coarse Correlated Equilibrium (CCE), defined in Definition 1.

Approximate CCE. Given a multilinear game, an approximate CCE can be computed by simulating no-regret dynamics (a.k.a. online convex programming) for each player. For example, one such no-regret dynamic is Generalized Infinitesimal Gradient Ascent (GIGA) [34], where in each iteration, for each player i we move $\boldsymbol{\pi}_i$ along the direction of the utility gradient $\nabla_i u_i(\boldsymbol{\pi}_{-i})$, and then project the resulting point back to P_i. The projection step is a convex optimization problem on P_i, and can be solved efficiently given an efficient separation oracle, or equivalently a procedure for PolytopeSolve. Therefore, under the same assumptions as Proposition 1, approximate CCE can be found efficiently.

Exact CCE. The above procedure does not guarantee exact CCE in polynomial-time. Next we obtain such a procedure, using LP duality and carefully designed separation oracle to get the following theorem.

Theorem 2. *Consider a multilinear game, with polynomial time subroutines for UtilGradient and PolytopeSolve. Then an exact Coarse Correlated Equilibrium (CCE) can be computed in polynomial time.*

Recall that a distribution over the set of pure-strategy profiles can be represented by a vector $\boldsymbol{x}$, satisfying $\boldsymbol{x} \geq 0$, $\sum_{\boldsymbol{s} \in S} x_{\boldsymbol{s}} = 1$. Given a multilinear game, the expected utility for player i under $\boldsymbol{x}$ is

$$u_i(\boldsymbol{x}) = \sum_{\boldsymbol{s}} x_{\boldsymbol{s}} u_i(\boldsymbol{s}) = \sum_{\boldsymbol{s}} \sum_{j_1,\ldots,j_n} U^i_{j_1,\ldots,j_n} x_{\boldsymbol{s}} \prod_k s_{k,j_k}$$

Given $\boldsymbol{x}$, the expected utility for player i if he deviates to strategy $\boldsymbol{s}_i$ is:

$$u_i^{\boldsymbol{s}_i}(\boldsymbol{x}) = \sum_{\boldsymbol{s}_{-i}} x_{\boldsymbol{s}_{-i}} u_i(\boldsymbol{s}_i, \boldsymbol{s}_{-i}) = \sum_{\boldsymbol{s}_{-i}} \sum_{j_1,\ldots,j_n} U^i_{j_1,\ldots,j_n} x_{\boldsymbol{s}_{-i}} \prod_k s_{k,j_k},$$

where $x_{\boldsymbol{s}_{-i}} = \sum_{\boldsymbol{s}_i \in S_i} x_{(\boldsymbol{s}_i, \boldsymbol{s}_{-i})}$ is the marginal probability of $\boldsymbol{s}_{-i}$ in distribution $\boldsymbol{x}$. We observe that $u_i^{\boldsymbol{s}_i}(\boldsymbol{x})$ is linear in $\boldsymbol{s}_i$. Specifically, $u_i^{\boldsymbol{s}_i}(\boldsymbol{x}) = \sum_{j_i} s_{i,j_i} \sum_{\boldsymbol{s}_{-i}} \sum_{j_{-i}} U^i_{j_1,\ldots,j_n} x_{\boldsymbol{s}_{-i}} \prod_{k \neq i} s_{k,j_k}$. We can extend the definition of $u_i^{\boldsymbol{s}_i}(\boldsymbol{x})$ beyond $\boldsymbol{s}_i \in S_i$ to any vector in the convex hull P_i; specifically for $\boldsymbol{p}_i \in P_i$, $u_i^{\boldsymbol{p}_i}(\boldsymbol{x})$ is defined to be $\sum_{j_i} p_{i,j_i} \sum_{\boldsymbol{s}_{-i}} \sum_{j_{-i}} U^i_{j_1,\ldots,j_n} \; x_{\boldsymbol{s}_{-i}} \prod_{k \neq i} s_{k,j_k}$. Recall from (1) that $g_i(\boldsymbol{x}) = \max_{\boldsymbol{s}_i \in S_i} u_i^{\boldsymbol{s}_i}(\boldsymbol{x})$, i.e. player i's expected utility if he deviates to a best response against $\boldsymbol{x}$. Since $u_i^{\boldsymbol{s}_i}(\boldsymbol{x})$ is linear in $\boldsymbol{s}_i$, we can write $g_i(\boldsymbol{x}) = \max_{\boldsymbol{p}_i \in P_i} u_i^{\boldsymbol{p}_i}(\boldsymbol{x})$. Recall that a distribution $\boldsymbol{x}$ is a Coarse Correlated Equilibrium (CCE) if it satisfies the *incentive constraints*: $u_i(\boldsymbol{x}) \geq g_i(\boldsymbol{x}), \quad \forall i$.

Consider the following optimization problem:

$$\max \sum_i z_i \tag{3}$$

$$\boldsymbol{x} \geq 0, \sum_{\boldsymbol{s}} x_{\boldsymbol{s}} = 1, \tag{4}$$

$$u_i(\boldsymbol{x}) - g_i(\boldsymbol{x}) - z_i \geq 0, \forall i \tag{5}$$

$$z_i \leq 0, \forall i \tag{6}$$

The feasible region correspond to a relaxation of CCE, due to the introduction of slack variables z. A feasible solution (x, z) with $z = 0$ is an optimal solution of the above problem (since $z \leq 0$); furthermore such a solution corresponds to a CCE $\boldsymbol{x}$ by construction.

This optimization problem is convex, but is difficult to handle directly because it has exponential number of variables $x_{\boldsymbol{s}}$ for each $\boldsymbol{s} \in S$. Take the dual optimization problem:

$$\min_{\boldsymbol{y} \geq 0} \max_{\boldsymbol{x} \in \Delta, z \leq 0} \sum_i z_i + \sum_i y_i (u_i(\boldsymbol{x}) - g_i(\boldsymbol{x}) - z_i) \tag{7}$$

$$= \min_{\boldsymbol{y} \geq 0} \max_{\boldsymbol{x} \in \Delta, z \leq 0} \sum_i (1 - y_i) z_i + \sum_i \min_{\boldsymbol{p}_i \in P_i} y_i (u_i(\boldsymbol{x}) - u_i^{\boldsymbol{p}_i}(\boldsymbol{x})) \tag{8}$$

$$= \min_{0 \leq \boldsymbol{y} \leq 1} \max_{\boldsymbol{x} \in \Delta} \min_{\boldsymbol{p}_1 \in P_1, \ldots, \boldsymbol{p}_n \in P_n} \sum_i y_i (u_i(\boldsymbol{x}) - u_i^{\boldsymbol{p}_i}(\boldsymbol{x})) \tag{9}$$

$$= \min_{0 \leq \boldsymbol{y} \leq 1} \min_{\boldsymbol{p}_1 \in P_1, \ldots, \boldsymbol{p}_n \in P_n} \max_{\boldsymbol{x} \in \Delta} \sum_i y_i (u_i(\boldsymbol{x}) - u_i^{\boldsymbol{p}_i}(\boldsymbol{x})) \tag{10}$$

where $\Delta = \{\boldsymbol{x} \in \mathbb{R}^{|S|} : \boldsymbol{x} \geq 0, \mathbf{1}^T \boldsymbol{x} = 1\}$. Going from (8) to (9), we used the fact that if $y_i > 1$, the maximizer can take z_i towards $-\infty$ and get arbitrarily high objective value. Therefore the outer minimizer should keep $y_i \leq 1$, in which case it is optimal for the maximizer to set $z = 0$ and the term $(1 - y_i) z_i$ disappears. In the last line we used the Minimax Theorem to switch the min and max operators. Since $\sum_i y_i (u_i(\boldsymbol{x}) - u_i^{\boldsymbol{p}_i}(\boldsymbol{x}))$ is linear in $\boldsymbol{x}$, it attains its maximum at one of the extreme points of Δ, i.e., one of the pure strategy profiles. Thus the dual problem is equivalent to

$$\min_{\boldsymbol{y},\boldsymbol{p}_1\cdots\boldsymbol{p}_n,t} t \tag{11}$$

$$0 \leq \boldsymbol{y} \leq 1; \quad \boldsymbol{p}_i \in P_i \quad \forall i \tag{12}$$

$$t \geq \sum_i y_i(u_i(\boldsymbol{s}') - u_i(\boldsymbol{p}_i, \boldsymbol{s}'_{-i})), \quad \forall \boldsymbol{s}' \in S \tag{13}$$

This is a nonlinear optimization problem due to the multiplication of y_i and $\boldsymbol{p}_i$ in (13), but can be transformed to a linear optimization problem via the following variable substitution: let $\boldsymbol{w}_i = y_i \boldsymbol{p}_i$. We now try to express the dual problem in terms of y_i and $\boldsymbol{w}_i$. Recall that $P_i = \{p \in \mathbb{R}^{m_i} | D_i p \leq \boldsymbol{f}_i\} \subset \mathbb{R}_+^{m_i}$. Then $\boldsymbol{w}_i$ satisfies $D_i \boldsymbol{w}_i \leq y_i \boldsymbol{f}_i$. For positive y_i, given $\boldsymbol{w}_i$ we can recover $\boldsymbol{p}_i = \boldsymbol{w}_i / y_i$. When $y_i = 0$, we need to make sure that $\boldsymbol{w}_i$ is also 0. This can be achieved using the constraints $\boldsymbol{w}_i \geq 0$ and $w_{ij} \leq M_{ij} y_i$, where the constant $M_{ij} = \max_{\boldsymbol{p}_i \in P_i} p_{ij}$ for all $j \in [m_i]$. Note that this a valid bound on w_{ij} when $y_i > 0$. M_{ij} can be computed in polynomial time by calling PolytopeSolve, and hence is polynomial-sized. The dual problem is then equivalent to

$$\min_{\boldsymbol{y},\boldsymbol{w}_1\ldots\boldsymbol{w}_n,t} t \tag{14}$$

$$0 \leq \boldsymbol{y} \leq 1; \quad D_i \boldsymbol{w}_i \leq y_i \boldsymbol{f}_i \quad \forall i \tag{15}$$

$$\boldsymbol{w}_i \geq 0, w_{ij} \leq M_{ij} y_i \quad \forall i, \forall j \in [m_i] \tag{16}$$

$$t \geq \sum_i y_i u_i(\boldsymbol{s}') - u_i(\boldsymbol{w}_i, \boldsymbol{s}'_{-i}), \quad \forall \boldsymbol{s}' \in S \tag{17}$$

where $u_i(\boldsymbol{w}_i, \boldsymbol{s}'_{-i})$ is the linear extension of $u_i(\boldsymbol{s}_i, \boldsymbol{s}'_{-i})$; i.e. $u_i(\boldsymbol{w}_i, \boldsymbol{s}'_{-i}) = \sum_{j_i} w_{i,j_i} \sum_{j_{-i}} U^i_{j_1\ldots j_n} \prod_{k \neq i} s'_{k,j_k}$. This is a linear program, with polynomial number of variables and exponential number of constraints. Since we know the primal objective is less or equal to 0, by LP duality, the optimal t in the dual is less or equal to 0. The following lemma establishes the existence of CCE in a way that does not use the existence of NE.

Lemma 3. *The dual LP (and therefore the primal LP) has optimal objective 0.*

This lemma says that for every candidate solution with $t < 0$, we can produce a hyperplane that separates it from the feasible set of the dual LP. We can use this lemma as a separation oracle in an algorithm similar to Papadimitriou & Roughgarden's [26] Ellipsoid Against Hope method to compute a CCE. However it would encounter similar numerical precision issues as discussed in [15], essentially due to the use of a convex combination of constraints which has a higher bit complexity than the individual constraints.

On the other hand, if we use a *pure separation oracle* that given $y, \boldsymbol{w}_1 \ldots \boldsymbol{w}_n$, finds $\boldsymbol{s}'$ such that $\sum_i y_i u_i(\boldsymbol{s}') - u_i(\boldsymbol{w}_i, \boldsymbol{s}'_{-i}) \geq 0$, we can use the approach as described in [15] to compute a CCE.

Lemma 4. *Consider a multilinear game, with polynomial-time subroutines for UtilGradient and PolytopeSolve. Then there is a polynomial-time algorithm for* *the following pure separation oracle problem: given* $y, \boldsymbol{w}_1 \dots \boldsymbol{w}_n$*, find pure strategy profile* $\boldsymbol{s}' \in S$ *such that* $\sum_i y_i u_i(\boldsymbol{s}') - u_i(\boldsymbol{w}_i, \boldsymbol{s}'_{-i}) \geq 0$*.*

Using Lemmas 3 and 4, in Appendix B we extend the approach of approach of [26, 15] and complete the proof of Theorem 2.

5 Complexity of Approximate NE: Membership in PPAD

In this section we analyze complexity of computing Nash equilibrium in multilinear games. Existence of a NE in multilinear game follows from [10] makes the problem total. On the other hand, since multilinear games contains normal-form multi-player games as a subcase, the Nash equilibria may be irrational [24]. In such a case the standard approach is to try approximation.

ϵ-approximate NE (ϵ-NE) Given a rational $\epsilon > 0$ in binary, a mixed strategy profile $\boldsymbol{\sigma}$ is an ϵ-approximate NE iff $\forall i \in N, u_i(\boldsymbol{\sigma}) \geq \max_{\boldsymbol{\sigma}'_i \in \Sigma_i} u_i(\boldsymbol{\sigma}'_i, \boldsymbol{\sigma}_{-i}) - \epsilon$. In case of multilinear games, due to Lemma 1, this is iff corresponding marginal strategy profile $\boldsymbol{\pi}$ satisfy $u_i(\boldsymbol{\pi}) \geq \max_{\boldsymbol{\pi}'_i \in P_i} u_i(\boldsymbol{\pi}'_i, \boldsymbol{\pi}_{-i}) - \epsilon$.

It is well known that even in two player normal form games, computing approximate NE is PPAD-complete [27, 8, 6]. Roughly speaking, PPAD captures the class of total search problems that can be reduced to `End-Of-Line` [27], which includes computing approximate fixed-points. Since normal form games are contained in multilinear games, the next corollary follows:

Corollary 1. *Given a rational* $\epsilon > 0$ *in binary, computing* ϵ*-approximate NE in multilinear games is PPAD-hard.*

Due to exponential size of the strategy spaces, it seems that computing an ϵ-NE in multilinear games could be a much harder problem given its generality. However, as we will show, it is no harder than computing a NE in 2-player games.

We note that, there has been recent efforts on showing PPAD membership for different classes of games [7]. However, the techniques are for games with polynomial type property, i.e. polynomial time computation of expected utility given mixed-strategy, inapplicable to multilinear games. Instead, we will use the characterization result (Proposition 2.2) of [9] to show that computing NE in multilinear games is in PPAD. See Appendix C for relevant definitions, and the proposition statement (Proposition 2).

Proposition 2 implies that to show membership of computing ϵ-NE in PPAD, it is enough to capture them as approximate fixed-points of a polynomially continuous and polynomially computable function. Next we will construct such a function for multi-linear games.

Consider the following function $\varphi : \Sigma \to \Sigma$ from [10] where $\varphi = (\varphi_1, ..., \varphi_n)$ and $\varphi_i : \Sigma \to \Sigma_i$ such that, $\varphi_i(\boldsymbol{\sigma}_i, \boldsymbol{\sigma}_{-i}) = \text{argmax}_{\overline{\boldsymbol{\sigma}}_i \in \Sigma_i}[u_i(\overline{\boldsymbol{\sigma}}_i, \boldsymbol{\sigma}_{-i}) - ||\overline{\boldsymbol{\sigma}}_i - \boldsymbol{\sigma}_i||^2]$. It was used to show existence of NE in concave games which includes multilinear games. However, notice that for multilinear games, description of mixed strategies is of exponential size, hence the function is not polynomially-computable. Its' polynomial-continuity is unclear.

Instead, once again we will use marginal strategies. Moreover, we can compute the expected utilities using the marginal strategies efficiently as long as there is polynomial-time procedure to compute the utility gradient. Let $P = \prod_{i \in N} P_i$, we redefine $\varphi : P \to P$ where $\varphi = (\varphi_1, ..., \varphi_n)$ and $\varphi_i : P \to P_i$ is

$$\varphi_i(\boldsymbol{\pi}_i, \boldsymbol{\pi}_{-i}) = \underset{\overline{\boldsymbol{\pi}}_i \in \boldsymbol{\pi}_i}{\operatorname{argmax}} [u_i(\overline{\boldsymbol{\pi}}_i, \boldsymbol{\pi}_{-i}) - ||\overline{\boldsymbol{\pi}}_i - \boldsymbol{\pi}_i||^2]. \quad (18)$$

Clearly, φ is a continuous function and therefore has a fixed-point. Next we show that its approximate fixed-points give approximate NE of the corresponding game. As the approximation goes to zero in the former we get exact NE in the latter, in other words exact fixed-points of (18) captures exact NE.

Lemma 5. *Given a rational* $\epsilon > 0$, *let* $\epsilon' = \frac{\epsilon}{|S| U_{max} H^n}$, *where* $H = \max_{i, \boldsymbol{p}_i \in P_i} ||\boldsymbol{p}_i||_1$ *and* $U_{max} = \max_{i, (j_1, ..., j_n) \in \prod_k [m_k]} |U^i_{j_1, ..., j_n}|$. *Then if* $\boldsymbol{\pi} \in P$ *is an* ϵ'*-approximate fixed-point of (18),* i.e., $||\varphi(\boldsymbol{\pi}) - \boldsymbol{\pi}||_\infty < \epsilon'$ *then it is a* 2ϵ*-approximate NE of the corresponding multilinear game.*

Lemma 5 implies that, for computation of approximate NE, it is enough to compute approximate fixed-point of function φ. Next we show that this function is polynomially continuous and polynomially computable and therefore computing its approximate fixed-point is contained in PPAD using Proposition 2, and therefore containment of approximate NE computation in PPAD follows. Next lemma shows polynomial-continuity by establishing equivalence of φ_i and a projection operator.

Lemma 6. *The function* φ *is polynomially continuous.*

By establishing connection to convex quadratic programming, next we show polynomially-computability of $\varphi(\boldsymbol{\pi})$.

Lemma 7. *The function* φ *is polynomially computable.*

Due to the assumption that *PolytopeSolve* has polynomial-time sub-routine, the size of $\max_{\boldsymbol{p}_i \in P_i} p_{ij}, \ \forall i, \ \forall j \in [m_i]$ is polynomial in the description of the game. Furthermore, $|S|$ is $2^{poly(n \sum_i m_i)}$. Therefore, if L is the size of the game description, then in Lemma 5 bit-length of H is polynomially bounded, and hence $size(\epsilon') = O(\log(1/\epsilon), poly(size(L)))$. Therefore, next theorem follows using Lemmas 5, 6 and 7, together with Proposition 2 (or [9]), and Corollary 1.

Theorem 3. *Given a multilinear game with polynomial-time subroutines for* PolytopeSolve *and* UtilGradient, *and* $\epsilon > 0$ *in binary, computing an* ϵ*-approximate NE of the game is in PPAD. Furthermore, it is PPAD-complete.*

Small Support Approximate NE. Using discussion of Section 4.2, given an ϵ-approximate NE $\boldsymbol{\pi} \in P$, each $\boldsymbol{\pi}_i$ can be represented as distribution over $m_i + 1$ pure strategies from S_i. However, existence of smaller support approximate NE is not clear. In Appendix D we study the same and obtain the following result.

Theorem 4. *Given a multilinear game, and given an* $\epsilon > 0$, *there exists an* ϵ*-approximate NE with support size* $O(M^2 \frac{\log(n) + \log(m) - \log(\epsilon)}{\epsilon^2})$ *for each player, where* $m = \max_i m_i$ *and* $M = (\max_{i, \boldsymbol{\pi} \in P} ||\nabla_i u_i(\boldsymbol{\pi})||_\infty) \max_{i, \boldsymbol{\pi}_i \in P_i} ||\boldsymbol{\pi}_i||_1$.

Note that M upper bounds the magnitude of the game's utilities $u_i(\boldsymbol{s}), \forall i, \forall \boldsymbol{s} \in S$. Finally we provide discussion in Appendix E.

References

1. Ahmadinejad, A., Dehghani, S., Hajiaghayi, M., Lucier, B., Mahini, H., Seddighin, S.: From Duels to Battlefields: Computing Equilibria of Blotto and Other Games. In: AAAI: Proceedings of the AAAI Conference on Artificial Intelligence (2016)
2. Babichenko, Y., Barman, S., Peretz, R.: Empirical distribution of equilibrium play and its testing application. CoRR abs/1310.7654 (2013), `http://arxiv.org/abs/1310.7654`
3. Bertsekas, D.: Convex Optimization Theory. Athena Scientific optimization and computation series, Athena Scientific (2009), `https://books.google.com/books?id=0H1iQwAACAAJ`
4. Budish, E., Che, Y.K., Kojima, F., Milgrom, P.: Designing random allocation mechanisms: Theory and applications. The American Economic Review 103(2), 585–623 (2013)
5. Chan, H., Jiang, A.X.: Congestion Games with Polytopal Strategy Spaces. In: IJCAI: Proceedings of the International Joint Conference on Artificial Intelligence. pp. 165–171 (2016)
6. Chen, X., Deng, X., Teng, S.H.: Computing Nash equilibria: Approximation and smoothed complexity. In: FOCS: Proceedings of the Annual IEEE Symposium on Foundations of Computer Science. pp. 603–612. IEEE Computer Society, Los Alamitos, CA, USA (2006)
7. Daskalakis, C., Fabrikant, A., Papadimitriou, C.: The game world is flat: The complexity of Nash equilibria in succinct games. In: ICALP: Proceedings of the International Colloquium on Automata, Languages and Programming. pp. 513–524 (2006)
8. Daskalakis, C., Goldberg, P.W., Papadimitriou, C.H.: The complexity of computing a Nash equilibrium. In: STOC: Proceedings of the Annual ACM Symposium on Theory of Computing. pp. 71–78 (2006)
9. Etessami, K., Yannakakis, M.: On the Complexity of Nash Equilibria and Other Fixed Points (Extended Abstract). In: FOCS: Proceedings of the Annual IEEE Symposium on Foundations of Computer Science. pp. 113–123 (2007)
10. Geanakoplos, J.: Nash and walras equilibrium via brouwer. Economic Theory 21(2), 585–603 (2003)
11. Govindan, S., Wilson, R.: Structure theorems for game trees. Proceedings of the National Academy of Sciences 99(13), 9077–9080 (2002)
12. Grötschel, M., Lovász, L., Schrijver, A.: The ellipsoid method and its consequences in combinatorial optimization. Combinatorica 1(2), 169–197 (1981)
13. Immorlica, N., Kalai, A.T., Lucier, B., Moitra, A., Postlewaite, A., Tennenholtz, M.: Dueling algorithms. In: STOC: Proceedings of the Annual ACM Symposium on Theory of Computing (2011)
14. Jain, M., Korzhyk, D., Vanek, O., Conitzer, V., Pechoucek, M., Tambe, M.: A double oracle algorithm for zero-sum security games on graphs. In: AAMAS (2011)
15. Jiang, A.X., Leyton-Brown, K.: Polynomial computation of exact correlated equilibrium in compact games. In: EC: Proceedings of the ACM Conference on Electronic Commerce (2011)
16. Jiang, A.X., Leyton-Brown, K., Bhat, N.: Action-graph games. Games and Economic Behavior 71(1), 141–173 (January 2011)
17. Kearns, M., Littman, M., Singh, S.: Graphical models for game theory. In: UAI: Proceedings of the Conference on Uncertainty in Artificial Intelligence. pp. 253–260 (2001)

18. Koller, D., Megiddo, N., Von Stengel, B.: Efficient computation of equilibria for extensive two-person games. Games and Economic Behavior 14(2), 247–259 (1996)
19. Koller, D., Milch, B.: Multi-agent influence diagrams for representing and solving games. In: IJCAI: Proceedings of the International Joint Conference on Artificial Intelligence (2001)
20. Köppe, M., Ryan, C.T., Queyranne, M.: Rational generating functions and integer programming games. Oper. Res. 59(6), 1445–1460 (Nov 2011)
21. Korzhyk, D., Conitzer, V., Parr, R.: Complexity of computing optimal stackelberg strategies in security resource allocation games. In: AAAI (2010)
22. Kozlov, M., Tarasov, S., Khachiyan, L.: The polynomial solvability of convex quadratic programming. USSR Computational Mathematics and Mathematical Physics 20(5), 223 – 228 (1980)
23. Murty, K.: Linear Complementarity, Linear and Non Linear Programming. Sigma series in applied mathematics, Heldermann Verlag (1988)
24. Nash, J.: Non-cooperative games. Annals of Mathematics 54(2), 289–295 (September 1951)
25. Nesterov, Y., Nemirovskii, A.: Interior-Point Polynomial Algorithms in Convex Programming. Society for Industrial and Applied Mathematics (1994)
26. Papadimitriou, C., Roughgarden, T.: Computing correlated equilibria in multi-player games. Journal of the ACM 55(3), 14 (July 2008)
27. Papadimitriou, C.H.: On the complexity of the parity argument and other inefficient proofs of existence. Journal of Computer and System Sciences 48(3), 498 – 532 (1994)
28. Pia, A.D., Ferris, M., Michini, C.: Totally unimodular congestion games. CoRR abs/1511.02784 (2015), http://arxiv.org/abs/1511.02784
29. Rabinovich, Z., Gerding, E., Polukarov, M., Jennings, N.R.: Generalised fictitious play for a continuum of anonymous players. In: IJCAI: Proceedings of the International Joint Conference on Artificial Intelligence. pp. 245–250 (2009)
30. Skorin-Kapov, J.: On strongly polynomial algorithms for some on strongly polynomial algorithms for some classes of quadratic programming problems. Mathematical Communications 2, 95–105 (1997)
31. Tambe, M.: Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned. Cambridge University Press (2011)
32. Vorobeychik, Y.: Mechanism Design and Analysis Using Simulation-Based Game Models. Ph.D. thesis, University of Michigan (2008)
33. Yin, Z., Jiang, A.X., Johnson, M.P., Tambe, M., Kiekintveld, C., Leyton-Brown, K., Sandholm, T., Sullivan, J.: TRUSTS: Scheduling randomized patrols for fare inspection in transit systems. In: IAAI (2012)
34. Zinkevich, M.: Online convex programming and generalized infinitesimal gradient ascent. In: ICML (2003)

A Examples

Example 1 (Security Games). Consider a security game [31] between a defender and an attacker. The defender and attacker want to protect and attack, respectively, some targets in T. For simplicity, the attacker can choose one target to attack while the defender can protect m ($< |T|$) targets. The defender and attacker's utilities depend on whether the attacked target is covered. The attacker's strategy space and the defender's strategy space can be captured using polytopes and the extreme points of the polytopes are clearly integer vectors. Moreover, the utilities of the attacker and defender are linear with respect to other players' strategies. Therefore, security games are multilinear.

Example 2 (Network Congestion Games). A network congestion game, as, e.g., in [5, 7, 28], of n players is defined on a directed graph $G = (V, E)$ with vertex set V and edge set E. Each player i's goal is to select a path from its source $s_i \in V$ to its destination $d_i \in V$ that minimizes the sum of the delay on each edge where the delay depends on the number of players selects the edge. The strategy set of each player i is the possible number of paths from s_i to d_i in G. For each player i, its constraint matrix D_i of P_i is the $|V|$ by $|E|$ incidence matrix of G and D_i is totally unimodular. Moreover, it is easy to see that the utility functions of the players are multilinear in the players' strategies (i.e., for each player i, the values $U^i \in \mathbb{R}^{|E|^n}$ correspond to the costs of the delay of the edges). Thus, network congestion games are a subclass of multilinear games.

Example 3 (Extensive-Form Games and Bayesian Games). Consider extensive form games (Bayesian games can be represented as extensive form, and thus is a special case). The space of behavior strategies can be represented as a product of simplices; however the utility functions are not multilinear in behavior strategies. The sequence form representation [18] was proposed to circumvent this. Player's mixed strategy space are represented using realization probabilities, which satisfy a polynomial number of linear constraints. The resulting utility function is multilinear. The case for n-players is discussed in [11].

Example 4 (Computing UtilGradient for Network Congestion Games). Using either of the dynamic programming techniques described in [5] for a general class of congestion games or in [7] for network congestion games, the utility gradients of network congestion games can be computed in polynomial time given the marginals.

B Missing Proofs

Proof of Lemma 1.
For each agent i we have the following:

$$u_i(\boldsymbol{\sigma}) = \sum_{\boldsymbol{s}} \boldsymbol{\sigma}(\boldsymbol{s}) u_i(\boldsymbol{s}) \tag{19}$$

$$= \sum_{\boldsymbol{s}} \prod_{k \in N} \boldsymbol{\sigma}_k(\boldsymbol{s}_k) \sum_{(j_1 \ldots j_n) \in \prod_k [m_k]} U^i_{j_1 \ldots j_n} \prod_{k \in N} s_{k,j_k} \tag{20}$$

$$= \sum_{(j_1 \ldots j_n) \in \prod_k [m_k]} U^i_{j_1 \ldots j_n} \sum_{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n} \prod_{k \in N} \boldsymbol{\sigma}_k(\boldsymbol{s}_k) s_{k,j_k} \tag{21}$$

$$= \sum_{(j_1 \ldots j_n) \in \prod_k [m_k]} U^i_{j_1 \ldots j_n} \Big(\sum_{\boldsymbol{s}_1 \in S_1} \boldsymbol{\sigma}_1(\boldsymbol{s}_1) s_{1,j_1} \Big) \ldots \Big(\sum_{\boldsymbol{s}_n \in S_n} \boldsymbol{\sigma}_n(\boldsymbol{s}_n) s_{n,j_n} \Big) \tag{22}$$

$$= \sum_{(j_1 \ldots j_n) \in \prod_k [m_k]} U^i_{j_1 \ldots j_n} \prod_{k \in N} \pi_{k,j_k}. \tag{23}$$

Proof of Lemma 3.

Suppose not, that is the optimal objective (i.e., t) is negative. Then there exists a feasible solution $\boldsymbol{y}, \boldsymbol{w}_1 \dots \boldsymbol{w}_n$ such that $\sum_i y_i u_i(\boldsymbol{s}') - u_i(\boldsymbol{w}_i, \boldsymbol{s}'_{-i}) < 0$ for all $\boldsymbol{s}' \in S$. Let $\boldsymbol{p}_i = \boldsymbol{w}_i / y_i$ if $y_i > 0$. If $y_i = 0$, (16) ensures that $\boldsymbol{w}_i = 0$, and we can take $\boldsymbol{p}_i$ to be an arbitrary vector in P_i. In either case, we have $\boldsymbol{w}_i = y_i \boldsymbol{p}_i$. Let $\boldsymbol{x}^*$ be the product distribution (i.e. mixed strategy profile) where each player i independently plays a mixed strategy with marginal $\boldsymbol{p}_i$. Consider the convex combination of $\sum_i y_i u_i(\boldsymbol{s}') - u_i(\boldsymbol{w}_i, \boldsymbol{s}'_{-i})$ by $\boldsymbol{x}^*$, i.e.

$$\sum_{\boldsymbol{s}'} x^*_{\boldsymbol{s}'} \sum_i (y_i u_i(\boldsymbol{s}') - u_i(\boldsymbol{w}_i, \boldsymbol{s}'_{-i})) \tag{24}$$

$$= \sum_i \sum_{\boldsymbol{s}'} x^*_{\boldsymbol{s}'} u_i(\boldsymbol{s}') y_i - x^*_{\boldsymbol{s}'} u_i(\boldsymbol{p}_i, \boldsymbol{s}'_{-i}) y_i \tag{25}$$

$$= \sum_i (u_i(\boldsymbol{x}^*) - u_i^{\boldsymbol{p}_i}(\boldsymbol{x}^*)) y_i \tag{26}$$

$$= 0 \tag{27}$$

In the last line we use the fact that since $\boldsymbol{x}^*$ is a product distribution with marginals $(\boldsymbol{p}_1, \dots, \boldsymbol{p}_n)$, we have $u_i^{\boldsymbol{p}_i}(\boldsymbol{x}^*) = u_i(\boldsymbol{p}_i, \boldsymbol{p}_{-i}) = u_i(\boldsymbol{x}^*)$. This contradicts the assumption that $\sum_i y_i u_i(\boldsymbol{s}') - u_i(\boldsymbol{w}_i, \boldsymbol{s}'_{-i}) < 0$ for all $\boldsymbol{s}'$. $\square$

Proof of Lemma 4.

We start with the product distribution $\boldsymbol{x}^*$ as described in Lemma 3 and then proceed player by player, changing the player's strategy to a pure strategy while maintaining the inequality $\sum_{\boldsymbol{s}'} x^*_{\boldsymbol{s}'} \sum_i y_i u_i(\boldsymbol{s}') - u_i(\boldsymbol{w}_i, \boldsymbol{s}'_{-i}) \geq 0$.

1. given $\boldsymbol{y}, \boldsymbol{w}_1 \dots \boldsymbol{w}_n$, compute product distribution $\boldsymbol{x}^*$ as represented by $\boldsymbol{p}_1, \dots, \boldsymbol{p}_n$, as described in Lemma 3.
2. for $k = 1 \dots n$:
   (a) find $\boldsymbol{s}_k = \arg\max_{\boldsymbol{s}'_k \in S_k} \sum_{\boldsymbol{s}'_{-k}} x^*_{\boldsymbol{s}'_k, \boldsymbol{s}'_{-k}} \sum_i y_i u_i(\boldsymbol{s}'_k, \boldsymbol{s}'_{-k}) - u_i^{\boldsymbol{w}_i}(\boldsymbol{s}'_k, \boldsymbol{s}'_{-k})$
   (b) set $\boldsymbol{p}_k = \boldsymbol{s}_k$.
3. output the pure strategy profile $\boldsymbol{p}_1, \dots, \boldsymbol{p}_k$.

The correctness of the procedure follows from the fact that the value of the expression $\sum_{\boldsymbol{s}'} x^*_{\boldsymbol{s}'} \sum_i y_i u_i(\boldsymbol{s}') - u_i(\boldsymbol{w}_i, \boldsymbol{s}'_{-i})$ is 0 at the beginning of procedure (by Lemma 3) and weakly increases in each iteration. To show computational efficiency, the only non-obvious step is 2a, where the maximization is over all pure strategies $\boldsymbol{s}'_k$ of player k. Instead of enumerating the pure strategies, we observe that the objective is linear in $\boldsymbol{s}'_k$ and thus we can solve the corresponding LP over P_k using our PolytopeSolve procedure. What remains is to calculate the coefficients in the objective. Recall that $\boldsymbol{x}^*$ is a product distribution $(\boldsymbol{p}_1, \dots, \boldsymbol{p}_n)$. Therefore the argmax can be rewritten as $\text{argmax}_{\boldsymbol{p}'_k \in P_k} \sum_i y_i (u_i(\boldsymbol{p}'_k, \boldsymbol{p}_{-k}) -$

$u_i^{\boldsymbol{p}_i}(\boldsymbol{p}'_k, \boldsymbol{p}_{-k})$). Recall that we can write $u_i(\boldsymbol{p}'_k, \boldsymbol{p}_{-k}) = \boldsymbol{p}'^T_k \nabla_k u_i(\boldsymbol{p}_{-k})$, and similarly for $u_i^{\boldsymbol{p}_i}(\boldsymbol{p}'_k, \boldsymbol{p}_{-k})$. Therefore the coefficients of the objective of step 2a can be written in terms of the utility gradients. □

Proof of Theorem 2. What remains is an application of the approach of [26, 15] to our primal and dual optimization problems. We sketch the arguments for completeness. Apply the ellipsoid method to the LP (14). Whenever the ellipsoid method takes a candidate solution $y, \boldsymbol{w}_1 \dots \boldsymbol{w}_n, t$ with $t < 0$, apply the separation oracle described in Lemma 4 to find $\boldsymbol{s}'$ such that $\sum_i y_i u_i(\boldsymbol{s}') - u_i(\boldsymbol{w}_i, \boldsymbol{s}'_{-i}) \geq 0$, which implies that the corresponding constraint in (17) is violated. Therefore the ellipsoid method will end (after a polynomial number K of iterations) with an optimal value of $t = 0$. Take the series of pure strategy profiles generated by the separation oracle during the run of ellipsoid method, $\boldsymbol{s}^{(1)}, \boldsymbol{s}^{(2)}, \dots \boldsymbol{s}^{(K)}$. We now take the primal optimization problem (3), but with $\boldsymbol{x}$ now restricted to the support $\boldsymbol{s}^{(1)}, \boldsymbol{s}^{(2)}, \dots \boldsymbol{s}^{(K)}$. This restricted primal problem has the same optimal value of 0 as the original primal problem, because running the ellipsoid method on their respective duals yield the same sequence of iterates and the same result. The restricted primal problem now has only a polynomial number (K) of variables. To solve this convex optimization problem, we just need a separation oracle for constraint (5), which can be done efficiently by solving the LP $\max_{\boldsymbol{p}_i \in P_i} u_i^{\boldsymbol{p}_i}(\boldsymbol{x})$. Since $\boldsymbol{x}$ now has polynomial support, the coefficients of the LP can be computed in polynomial time, and we can apply PolytopeSolve. The result of solving the restricted primal is a correlated distribution supported on $\boldsymbol{s}^{(1)}, \boldsymbol{s}^{(2)}, \dots \boldsymbol{s}^{(K)}$. Since it achieves the optimal value of 0, it is a CCE. □

Proof of Lemma 5.
To the contrary suppose $\boldsymbol{\pi}$ is an ϵ'-approximate fixed-point of (18), but not a 2ϵ-approximate NE. For ease of notation, let $a_{ij} = (\nabla_i u_i(\boldsymbol{\pi}_{-i}))_j,\ \forall i,\ \forall j \in [m_i]$, then clearly $a_{ij} \leq |S_{-i}|H^{n-1}U_{max}$. Let $\bar{\boldsymbol{\pi}} = \varphi(\boldsymbol{\pi})$, then it satisfies $|\bar{\pi}_{ij} - \pi_{ij}| < \epsilon',\ \forall i \in N, \forall j \in [m_i]$. Therefore, $\forall i$ we have,

$$\begin{aligned} |u_i(\boldsymbol{\pi}_i, \boldsymbol{\pi}_{-i}) - u_i(\bar{\boldsymbol{\pi}}_i, \boldsymbol{\pi}_{-i})| &= |\textstyle\sum_{j\in[m_i]} \pi_{ij}a_{ij} - \sum_{j\in[m_i]} \bar{\pi}_{ij}a_{ij}| \\ &= |\textstyle\sum_{j\in[m_i]} (\pi_{ij} - \bar{\pi}_{ij})a_{ij}| \\ &\leq \epsilon' \textstyle\sum_{j\in[m_i]} |a_{ij}| < \epsilon \quad (\because |a_{ij}| \leq |S_{-i}|H^{n-1}U_{max}) \end{aligned}$$

Now if we manage to show that $u_i(\bar{\boldsymbol{\pi}}_i, \boldsymbol{\pi}_{-i}) \geq \max_{\boldsymbol{p}_i \in P_i} u_i(\boldsymbol{p}_i, \boldsymbol{\pi}_{-i}) - \epsilon,\ \forall i$, then using the above derivation we get $u_i(\boldsymbol{\pi}) > u_i(\bar{\boldsymbol{\pi}}_i, \boldsymbol{\pi}_i) - \epsilon \geq \max \boldsymbol{p}_i \in P_i u_i(\boldsymbol{p}_i, \boldsymbol{\pi}_{-i}) - 2\epsilon,\ \forall i$, a contradiction to $\boldsymbol{\pi}$ not being 2ϵ-approximate NE, and the lemma follows.

Claim. $u_i(\bar{\boldsymbol{\pi}}_i, \boldsymbol{\pi}_{-i}) \geq \max_{\boldsymbol{p}_i \in P_i} u_i(\boldsymbol{p}_i, \boldsymbol{\pi}_{-i}) - \epsilon,\ \forall i$

Proof. To the contrary suppose for player i, $\exists \boldsymbol{p}_i \in P_i$ such that $u_i(\boldsymbol{p}_i, \boldsymbol{\pi}_{-i}) > u_i(\bar{\boldsymbol{\pi}}, \boldsymbol{\pi}_{-i}) + \epsilon$. Let $\tau = u_i(\boldsymbol{p}_i, \boldsymbol{\pi}_{-i}) - u_i(\bar{\boldsymbol{\pi}}, \boldsymbol{\pi}_{-i}) - \epsilon$, and $\delta = \frac{\tau+\epsilon}{8m^2H^2}$, where

$m = \max_i m_i$. Clearly, $\tau, \delta > 0$. Consider a marginal profile $\boldsymbol{p}'_i = (1-\delta)\bar{\boldsymbol{\pi}}_i + \delta \boldsymbol{p}_i$.

$$\begin{aligned}
u_i(\boldsymbol{p}'_i, \boldsymbol{\pi}_{-i}) - ||\boldsymbol{p}'_i - \boldsymbol{\pi}_i||^2 &= \textstyle\sum_{j\in[m_i]}((1-\delta)\bar{\pi}_{ij} + \delta p_{ij})a_{ij} - ||(1-\delta)\bar{\boldsymbol{\pi}}_i + \delta \boldsymbol{p}_i - \boldsymbol{\pi}_i||^2 \\
&\geq (1-\delta)u_i(\bar{\boldsymbol{\pi}}_i, \boldsymbol{\pi}_{-i}) + \delta u_i(\boldsymbol{p}_i, \boldsymbol{\pi}_{-i}) - \epsilon'^2 - \delta^2||\boldsymbol{p}_i + \bar{\boldsymbol{\pi}}_i||^2 \\
&\geq u_i(\bar{\boldsymbol{\pi}}_i, \boldsymbol{\pi}_{-i}) + (\tau+\epsilon)\delta - \epsilon'^2 - \delta^2(4m^2H^2) \\
&= u_i(\bar{\boldsymbol{\pi}}_i, \boldsymbol{\pi}_{-i}) + \delta(\tau + \epsilon - \tfrac{\epsilon^2}{\delta(|S|U_{max}H^n)^2} - \delta(4m^2H^2) \\
&= u_i(\bar{\boldsymbol{\pi}}_i, \boldsymbol{\pi}_{-i}) + \delta(\tau + \epsilon - \tfrac{\epsilon^2}{(|S|U_{max}H^n)^2}\tfrac{8m^2H^2}{\tau+\epsilon} - \tfrac{\tau+\epsilon}{2}) \\
&\geq u_i(\bar{\boldsymbol{\pi}}_i, \boldsymbol{\pi}_{-i}) + \delta(\tfrac{\tau+\epsilon}{2} - \tfrac{\epsilon^2}{(|S|U_{max}H^n)^2}\tfrac{8m^2H^2}{\epsilon}) \\
&\geq u_i(\bar{\boldsymbol{\pi}}_i, \boldsymbol{\pi}_{-i}) + \delta(\tfrac{\tau+\epsilon}{2} - \tfrac{\epsilon}{2}) \\
&= u_i(\bar{\boldsymbol{\pi}}_i, \boldsymbol{\pi}_{-i}) + \tfrac{\delta\tau}{2} \\
&> u_i(\bar{\boldsymbol{\pi}}_i, \boldsymbol{\pi}_{-i})
\end{aligned}$$

Since $u_i(\bar{\boldsymbol{\pi}}_i, \boldsymbol{\pi}_{-i}) \geq u_i(\bar{\boldsymbol{\pi}}_i, \boldsymbol{\pi}_{-i}) - ||\boldsymbol{\pi}_i - \bar{\boldsymbol{\pi}}_i||^2$, we get a contradiction to $\bar{\boldsymbol{\pi}}_i = \varphi_i(\boldsymbol{\pi}_i, \boldsymbol{\pi}_{-i})$. □

Proof of Lemma 6.
To show that the function φ is polynomially continuous, we first define $\texttt{proj}_i(\boldsymbol{v})$ to be the projection function that maps $\boldsymbol{v} \in \mathbb{R}^{m_i}$ to the closest point in P_i and provide the following claim.

Claim. For each player $i \in N$, $\varphi_i(\boldsymbol{\pi}_i, \boldsymbol{\pi}_{-i}) = \texttt{proj}_i\left(\boldsymbol{\pi}_i + \frac{1}{2}\nabla_i u_i(\boldsymbol{\pi}_{-i})\right)$.

Proof. It follows that

$$\begin{aligned}
\texttt{proj}_i\left(\boldsymbol{\pi}_i + \frac{1}{2}\nabla_i u_i(\boldsymbol{\pi}_{-i})\right) &= \operatorname*{argmin}_{\overline{\boldsymbol{\pi}}_i \in P_i} \left\|\overline{\boldsymbol{\pi}}_i - \boldsymbol{\pi}_i - \frac{1}{2}\nabla_i u_i(\boldsymbol{\pi}_{-i})\right\| \\
&= \operatorname*{argmin}_{\overline{\boldsymbol{\pi}}_i \in P_i} \left\|-\frac{1}{2}\nabla_i u_i(\boldsymbol{\pi}_{-i}) + \overline{\boldsymbol{\pi}}_i - \boldsymbol{\pi}_i\right\|^2 \\
&= \operatorname*{argmin}_{\overline{\boldsymbol{\pi}}_i \in P_i} [-\overline{\boldsymbol{\pi}}_i \nabla_i u_i(\boldsymbol{\pi}_{-i}) + \|\overline{\boldsymbol{\pi}}_i - \boldsymbol{\pi}_i\|^2] \\
&= \operatorname*{argmax}_{\overline{\boldsymbol{\pi}}_i \in P_i} [u_i(\overline{\boldsymbol{\pi}}_i, \boldsymbol{\pi}_{-i}) - ||\overline{\boldsymbol{\pi}}_i - \boldsymbol{\pi}_i||^2]. \square
\end{aligned}$$

To show the function φ is polynomially continuous, we want to show that $||\varphi(\boldsymbol{x}) - \varphi(\boldsymbol{y})||_k \leq C||\boldsymbol{x} - \boldsymbol{y}||_k$ for all $\boldsymbol{x}, \boldsymbol{y} \in P$ for some constant C and norm k. To show this, it is suffice to show that for each $i \in N$, $||\varphi_i(\boldsymbol{x}_i, \boldsymbol{x}_{-i}) - \varphi_i(\boldsymbol{y}_i, \boldsymbol{y}_{-i})||_2 \leq C||\boldsymbol{x}_i - \boldsymbol{y}_i||_2$. From Lemma B, we have that

$$\begin{aligned}
&||\varphi_i(\boldsymbol{x}_i, \boldsymbol{x}_{-i}) - \varphi_i(\boldsymbol{y}_i, \boldsymbol{y}_{-i})||_2 \\
&\qquad = \left\|\texttt{proj}_i\left(\boldsymbol{x}_i + \frac{1}{2}\nabla_i u_i(\boldsymbol{x}_{-i})\right) - \texttt{proj}_i\left(\boldsymbol{y}_i + \frac{1}{2}\nabla_i u_i(\boldsymbol{y}_{-i})\right)\right\|_2 \\
&\qquad \leq ||\boldsymbol{x}_i - \boldsymbol{y}_i||_2,
\end{aligned}$$

where the inequality due to fact that the projection function onto a convex compact domain is non-expansive [3]. □

Proof of Lemma 7.
To show that the function φ is polynomially computable, we show that, for each $i \in N$, φ_i is polynomially computable.

First notice that, for any $\boldsymbol{\pi} = (\boldsymbol{\pi}_i, \boldsymbol{\pi}_{-i}) \in P$,

$$\begin{aligned}\varphi_i(\boldsymbol{\pi}_i, \boldsymbol{\pi}_{-i}) &= \operatorname*{argmax}_{\overline{\boldsymbol{\pi}}_i \in \boldsymbol{\pi}_i} [u_i(\overline{\boldsymbol{\pi}}_i, \boldsymbol{\pi}_{-i}) - ||\overline{\boldsymbol{\pi}}_i - \boldsymbol{\pi}_i||^2] \\ &= \operatorname*{argmax}_{\overline{\boldsymbol{\pi}}_i \in \boldsymbol{\pi}_i} [\overline{\boldsymbol{\pi}}_i \nabla_i u_i(\boldsymbol{\pi}_{-i}) - ||\overline{\boldsymbol{\pi}}_i - \boldsymbol{\pi}_i||^2].\end{aligned}$$

The above problem in fact is a convex quadratic program. Moreover, because the matrix involving the quadratic term is positive definite, we can solve it using the ellipsoid method in polynomial time [25]. In addition, if there is a rational solution, the ellipsoid method is guaranteed to find it [22, 30].

Now, we argue that for any rational $\boldsymbol{\pi} \in P$, there is a rational solution of $\varphi_i(\boldsymbol{\pi})$. This follows using the fact that Karush-Kuhn-Tucker conditions of a convex quadratic program are sufficient, and are captured through a linear complementarity problem (LCP) [23] with rational parameters. And an LCP with rational input parameters always has a rational solution [23]. □

C Proposition 2.2 of [9]

Before stating the proposition, we need the following definitions.

Polynomially-continuous. Let $\mathcal{F}$ be the class of functions associated with a fixed point search problem Π where each instance of I of Π is associated with the function F_I of $\mathcal{F}$. The class F is *polynomially continuous* if there is a polynomial $q(z)$ such that for all instances I and all rational $\epsilon > 0$, there is a rational $\delta > 0$ such that $size(\delta) \leq q(|I| + size(\epsilon))$ and such that for all $\boldsymbol{x}, \boldsymbol{y} \in D_I, ||\boldsymbol{x} - \boldsymbol{y}||_\infty < \delta \implies ||F_I(\boldsymbol{x}) - F_I(\boldsymbol{y})||_\infty < \epsilon$ where $size(\mathrm{r})$ denote the number of bits in the numerator and denominator of a rational number r, D_I is the convex compact domain of the instance I, and the size $|I|$ of the instance I is the length of the string that represents it. Moreover, all Lipschitz continuous functions (functions that satisfy $||F_I(\boldsymbol{x}) - F_I(\boldsymbol{y})||_k \leq C_I||\boldsymbol{x} - \boldsymbol{y}||_k$ for all $\boldsymbol{x}, \boldsymbol{y} \in D_I$ with Lipschitz constant $C_I \leq 2^{poly(|I|)}$ for some norm k) are polynomially continuous.

Polynomially-computable. The class $\mathcal{F}$ is *polynomially computable* if there is a polynomial $q(z)$ such that (a) for every instance I, D_I is a convex polytope described by a set of linear inequalities with rational coefficients that can be computed from I in time $q(|I|)$, and (b) given a rational vector $\boldsymbol{x} \in D_I$, the image $F_I(\boldsymbol{x})$ is rational and can be computed from I and $\boldsymbol{x}$ in time $q(|I| + size(\boldsymbol{x}))$.

Finally, we are ready to state the result of [9] below.

Proposition 2. *[Proposition 2.2 of [9]] Let $\mathcal{F}$ be the class of functions associated with a fixed point search problem Π. If $\mathcal{F}$ is polynomially continuous* *and polynomially computable, then, given an instance $I \in \Pi$ and a rational $\epsilon > 0$ in binary as input, the problem of computing a rational $\boldsymbol{x}' \in D_I$ such that $||F_I(\boldsymbol{x}') - \boldsymbol{x}'||_\infty < \epsilon$ (ϵ-approximate fixed-point) is in PPAD.*

D Existence of Small Support Nash Equilibrium

The standard approach for such results is to use the probabilistic method: start with a Nash equilibrium $\boldsymbol{\sigma}$. For each player i sample k pure strategies according to distribution $\boldsymbol{\sigma}_i$. Take the empirical distribution for each player to form a mixed strategy profile. (Such empirical distribution of k pure strategies is also known as k-uniform strategies.) Then show that with positive probability, this mixed strategy profile is an ϵ-Nash equilibrium. This then establishes the existence of ϵ-Nash equilibrium in k-uniform strategies.

However, if we use the standard method to check for ϵ-Nash equilibrium by comparing our strategy to every other pure strategy, that would be exponential number of comparisons. Using the union bound over this set would yield an exponential factor on our bound. Instead, we avoid this exponential number of comparisons by exploiting multilinearity. In particular, let $\boldsymbol{\pi}$ be the marginal strategy profile of the Nash equilibrium $\boldsymbol{\sigma}$. We note that i's utility is $\boldsymbol{\pi}_i^T \nabla_i u_i(\boldsymbol{\pi})$. Let $\tilde{\boldsymbol{\sigma}}$ be the sampled strategy profile, with marginals $\tilde{\boldsymbol{\pi}}$. Using standard methods, we can show that w.h.p., $\boldsymbol{\pi}_i$ is close to $\tilde{\boldsymbol{\pi}}_i$ and $\nabla_i u_i(\boldsymbol{\pi})$ is close to $\nabla_i u_i(\tilde{\boldsymbol{\pi}})$, which then allow us to establish that $\tilde{\boldsymbol{\pi}}$ is an ϵ-Nash equilibrium.

A direct application of Hoeffding's inequality yields the following lemma.

Lemma 8. *Let $\tilde{\boldsymbol{\sigma}}_i$ (with marginals $\tilde{\boldsymbol{\pi}}_i$) be the empirical distribution of k pure strategies drawn from a mixed strategy σ_i with marginals $\boldsymbol{\pi}_i$. Then $\Pr(|\tilde{\boldsymbol{\pi}}_{ij} - \boldsymbol{\pi}_{ij}| > \epsilon) \leq 2e^{-\epsilon^2 k/(2H^2)}$, for all $j \in [m_i]$, where $H = \max_{i,\boldsymbol{\pi}_i \in P_i} ||\boldsymbol{\pi}_i||_\infty$.*

Lemma 9. $\Pr(|(\nabla_i u_i(\tilde{\boldsymbol{\pi}}))_j - (\nabla_i u_i(\boldsymbol{\pi}))_j| > \epsilon) \leq \frac{4e^{-\epsilon^2 k/(8H'^2)}}{\epsilon}$ *for all $j \in [m_i]$, where $H' = \max_{i,\boldsymbol{\pi} \in P} ||\nabla_i u_i(\boldsymbol{\pi})||_\infty$.*

Proof. We can directly apply Theorem 9 of [2], a concentration inequality for expectations of product distributions. Indeed, $\nabla_i(u_i(\boldsymbol{\pi}))_j$ is the expectation of the rate of change of u_i w.r.t j-th component of i's strategy, over the product distribution σ_{-i}. □

Proof of Theorem 4. Let $m = \max_i m_i$, and $H'' = \max_{i,\boldsymbol{\pi}_i \in P_i} ||\boldsymbol{\pi}_i||_1$, and set $k = \frac{128H'^2H''^2(\log n + \log m + \log 8 - \log \epsilon)}{\epsilon^2}$. By union bound we have

$$\Pr[\exists i, ||\tilde{\boldsymbol{\pi}}_i - \boldsymbol{\pi}_i||_\infty > \epsilon/(2H') \text{ OR } \exists i, ||\nabla_i u_i(\tilde{\boldsymbol{\pi}}) - \nabla_i u_i(\boldsymbol{\pi})||_\infty > \epsilon/(4H'')] \\ \leq nm(\frac{4e^{-\epsilon^2 k/(128H'^2H''^2)}}{\epsilon} + 2e^{-\epsilon^2 k/(8H^2H'^2)}) < 8nm\frac{e^{-\epsilon^2 k/(128H'^2H''^2)}}{\epsilon} = 1.$$

Thus with positive probability, we have $||\tilde{\boldsymbol{\pi}}_i - \boldsymbol{\pi}_i||_\infty \leq \epsilon/(2H')$ and $||\nabla_i u_i(\tilde{\boldsymbol{\pi}}) - \nabla_i u_i(\boldsymbol{\pi})||_\infty \leq \epsilon/(4H'')$ for all i. We claim $\tilde{\boldsymbol{\pi}}$ is an ϵ-Nash equilibrium. To prove this we need to show for all i, for all $\boldsymbol{\pi}'_i \in P_i$, $u_i(\boldsymbol{\pi}'_i, \tilde{\boldsymbol{\pi}}_{-i}) \leq u_i(\tilde{\boldsymbol{\pi}}) + \epsilon$. Indeed,

$$\begin{aligned} u_i(\boldsymbol{\pi}'_i, \tilde{\boldsymbol{\pi}}_{-i}) &= \boldsymbol{\pi}'_i \nabla_i u_i(\tilde{\boldsymbol{\pi}}) \\ &\leq \boldsymbol{\pi}'_i \nabla_i u_i(\boldsymbol{\pi}) + \tfrac{\epsilon}{4} \\ &\leq \boldsymbol{\pi}_i \nabla_i u_i(\boldsymbol{\pi}) + \tfrac{\epsilon}{4} \\ &\leq \boldsymbol{\pi}_i \nabla_i u_i(\tilde{\boldsymbol{\pi}}) + \tfrac{\epsilon}{4} + \tfrac{\epsilon}{4} \\ &\leq \tilde{\boldsymbol{\pi}}_i \nabla_i u_i(\tilde{\boldsymbol{\pi}}) + \tfrac{\epsilon}{4} + \tfrac{\epsilon}{4} + \tfrac{\epsilon}{2} \\ &= u_i(\tilde{\boldsymbol{\pi}}) + \epsilon. \end{aligned}$$

This yields the asymptotic bound in the theorem statement. □

E Discussion

In this paper, we defined multilinear games, that generalizes various game forms like security games, extensive-form games, congestion games, Bayesian games, and provided a unifying approach to equilibrium characterization and computation. In multilinear games, pure strategies of a player are a set of integer points in polynomial dimension, however they may be exponentially many. This introduces the primary difficulty and we handle it by moving to the space of marginals. We provide a polynomial-time procedure to compute CCE, and a polynomially-continuous and polynomially-computable fixed-point formulation of NE. Using the latter we show that approximate NE computation in multilinear games is in PPAD, *i.e.,* not harder than in normal-form games, which opens up possibility of designing path following algorithms. It will be interesting and highly useful to design such an algorithm for multilinear games. The complexity of strong approximation remains open, and we conjecture that it should be FIXP-complete like normal-form games.

We show existence of approximate NE with support size logarithmic in the dimension of the strategy polytope for each player. Such a result leads to pseudo-polynomial time algorithm in normal-form games when number of players is a contestant. However, due to exponentially many pure strategies in multilinear games, the result does not extend directly. It will be interesting to see if there is any way to exploit the marginal space to obtain such an algorithm.

We provided a procedure to compute coarse correlated equilibrium in multilinear games by exploiting linear optimization on strategy polytopes. Is there a way to extend this procedure to compute correlated equilibrium? Finally, it will interesting to analyze multilinear games with special properties, like potential games, for say analyzing convergence of natural dynamics like best response.