Convergent Adaptive Discretization Methods for Computing Correlated Equilibria of Polynomial Games

Noah D. Stein, Asuman Ozdaglar, Pablo A. Parrilo

Abstract— **We construct a family of iterative discretization algorithms for computing sequences of finitely-supported ϵ-correlated equilibria of n-player games with polynomial utility functions. These algorithms can be implemented efficiently using semidefinite programming and sum of squares techniques. They converge in the sense that they drive ϵ to zero in the limit as points are added to the discretization. We show how a natural discretization scheme proposed previously can be viewed as a limiting case of this new family of algorithms. Finally we provide a counterexample showing that this limiting case is singular, i.e. ϵ need not converge to zero.**

I. Introduction

This paper extends our previous work on computing correlated equilibria in games with polynomial utility functions [12]. Most research on computing equilibria has focused on games with finite strategy spaces. However many games of practical importance have infinite strategy spaces. Our goal is to make these games computationally tractable and hence more viable as a modeling tool.

Research along these lines began in the 1950's with a series of publications by Dresher, Karlin, and Shapley in which they examined structural properties of equilibria in games with polynomial and polynomial-like utility functions [1], [2], [4], [5]. Only ad hoc techniques of limited applicability were known for computing equilibria of such games until Parrilo showed that equilibria of two-player zero-sum polynomial games could be computed using semidefinite programming and sum of squares techniques [8]. These methods have been extended to algorithmically approximate Nash and correlated equilibria of nonzero-sum games [10], [11], [12].

In this paper we provide an efficient method for computing approximate correlated equilibria. In earlier work we have shown that it is possible to compute a nested sequence of outer approximations to the set of correlated equilibria (when probability measures are described in terms of joint moments) of a polynomial game which converge in a well-defined sense to a description of this set of equilibria [12]. In the same paper we presented an adaptive discretization method intended to iteratively compute a sample correlated equilibrium as the limit of a sequence of finitely supported ϵ-correlated equilibria. The algorithm is adaptive in the sense that new points are added to the discretization (i.e. new strategies are allowed to be chosen with positive probability) according to a heuristic which seems to drive ϵ to zero quickly. This convergence behavior was conjectured to occur in general based on empirical evidence.

Here we construct a parametrized family of related discretization algorithms and prove that under these procedures ϵ goes to zero as the number of points in the discretization increases. We show that it is possible to view the adaptive discretization algorithm of [12] as a limiting case of certain parameter choices for these new algorithms. Our convergence proof does not go through in this limiting case and we show that this is due to a fundamental flaw in the earlier adaptive discretization algorithm. To do so we construct a counterexample in which ϵ remains bounded away from zero, refuting our conjecture in [12].

The remainder of this paper is organized as follows. We define the class of games and equilibria we will study in Section II. We present the main algorithm in Section III and show how to implement it using semidefinite programs and sum of squares techniques in Section IV. The counterexample showing that the earlier adaptive discretization algorithm does not converge comprises Section V, after which we finish with conclusions.

II. Continuous games and correlated equilibria

We begin by introducing the class of games we will study and showing how to define correlated equilibria in this setting. Some notational conventions used throughout are that subscripts refer to players, while superscripts are frequently used for other indices (it will be clear from context when they represent exponents). If S_j are sets for $j = 1, \ldots, n$ then $S = \prod_{j=1}^n S_j$ and $S_{-i} = \prod_{j \neq i} S_j$. The n-tuple s and the $(n-1)$-tuple s_{-i} are formed from the points s_j similarly. For simplicity we will write $\pi(s)$ in place of $\pi(\{s\})$ for the probability mass assigned to a singleton $\{s\} \subseteq S$. All polynomials will be assumed to have real coefficients.

Definition 2.1: A **continuous game** is a normal form game with n players, each of whom has a **pure strategy space** C_i which is a compact metric space and a continuous utility function $u_i : C \to \mathbb{R}$ ($C = \prod C_i$). A **polynomial game** is a continuous game with $C_i = [-1, 1]$ for all i and polynomial utility functions. Given the other players' strategy choices, each player seeks to maximize his own utility.

The results below on polynomial games also apply to more general semialgebraic strategy spaces (e.g. boxes or spheres in place of intervals) and piecewise polynomial or piecewise rational utilities, but for simplicity we will not consider such cases here.

Massachusetts Institute of Technology / Cambridge, MA 02139
nstein@mit.edu

Since the strategy spaces are infinite, there is no known way to express the entire set of correlated equilibria in terms of finitely many parameters; this is generally handled by means of infinite-dimensional families of departure functions [3] or alternatively by our recent method of test functions [12]. We will not need to consider these extra complications here, because we will only be working with finitely supported probability distributions, i.e. those which assign all probability mass to some finite set. In this case the correlated equilibrium conditions can be written in a much simpler equivalent form, as in the case of finite pure strategy spaces. Nonetheless it is important to note that the definition below excludes many ϵ-correlated equilibria which are not finitely supported.

Definition 2.2: A probability measure π on $C = \prod C_i$ with support contained in some finite set $\tilde{C} = \prod \tilde{C}_i$ is a **[finitely supported] ϵ-correlated equilibrium** if

$$\sum_{s_{-i}\in\tilde{C}_{-i}} \pi(s)\left[u(t_i, s_{-i}) - u_i(s)\right] \leq \epsilon_{i,s_i} \text{ for all } s_i \in \tilde{C}_i,$$
$$\text{all } t_i \in C_i, \text{ and all } i$$
$$\sum_{s_i\in\tilde{C}_i} \epsilon_{i,s_i} \leq \epsilon \text{ for all } i.$$

If $\epsilon = 0$ then we call π an **[exact, finitely supported] correlated equilibrium**.

To interpret this definition suppose a random variable $R = (R_1, \ldots, R_n)$ were drawn according to the distribution π which is known to all players. Suppose each player i is given R_i as a "recommended" strategy but has no knowledge of the other R_j aside from that which he can infer from the value of R_i and π via Bayes' rule. Then the above definition states that if all players beside i play according to their recommendations, player i cannot expect to improve his payoff by more than ϵ by deviating from his recommendation.

If the above definition is properly extended to arbitrary probability distributions which are not necessarily finitely supported, it can be shown that the set of ϵ-correlated equilibria is compact with respect to the weak topology on the space of probability distributions. We omit the precise definitions and proofs here, but note this fact because it has important implications for our algorithms. In particular any sequence of ϵ^k-correlated equilibria with ϵ^k going to zero as k goes to infinity (such as those produced by the algorithms below) has a subsequence which converges weakly to some probability distribution. Furthermore, any such limiting probability distribution is an exact correlated equilibrium.

III. A family of convergent adaptive discretization algorithms

Fix a continuous game. In this section we provide two algorithms (the second is in fact a parametrized family of algorithms which generalizes the first) to compute a sequence of ϵ^k-correlated equilibria such that $\lim_{k\to\infty} \epsilon^k = 0$. In the following section we will show that for polynomial games these algorithms can be implemented efficiently using semidefinite programming.

Informally, these algorithms work as follows. Each iteration k begins with a finite set $\tilde{C}_i^k \subseteq C_i$ of strategies which each player i is allowed to play with positive probability in that iteration; the initial choice of this set at iteration $k = 0$ is arbitrary. We then compute the "best" ϵ-correlated equilibrium which uses only these allowed strategies, i.e. the one which minimizes ϵ (subject to some extra technical conditions needed to ensure convergence).

Given the optimal objective value ϵ^k and optimal probability distribution π^k, there is some player i who can improve his payoff by ϵ^k if he switches from his recommended strategies to certain other strategies. We interpret these other strategies as good choices for that player to use to help make ϵ^k smaller in later iterations k. Therefore we add these strategies to $\tilde{C}_i^k$ to get $\tilde{C}_i^{k+1}$ and repeat this process for iteration $k + 1$.

Algorithm 3.1: Let $k = 0$ and for each player fix a finite subset $\tilde{C}_i^0 \subseteq C_i$.

- Let π^k be an ϵ^k-correlated equilibrium of the game having minimal ϵ^k subject to two extra conditions. First, π^k must be supported on $\tilde{C}^k$. Second, we require that π^k be an exact correlated equilibrium of the finite game induced when deviations from the recommended strategies are restricted to the set $\tilde{C}^k$, i.e. when we replace the condition $t_i \in C_i$ in Definition 2.1 with $t_i \in \tilde{C}_i^k$.
  That is to say, let ϵ^k be the optimal value of the following optimization problem, and π^k be an optimal assignment to the decision variables.
$$\begin{aligned} \text{minimize} \quad & \epsilon \\ \text{s.t.} \quad & \\ \sum_{s_{-i}\in\tilde{C}^k_{-i}} \pi(s)\left[u_i(t_i, s_{-i}) - u_i(s)\right] \leq 0 \quad & \text{for all } i \text{ and} \\ & s_i, t_i \in \tilde{C}_i^k \\ \sum_{s_{-i}\in\tilde{C}^k_{-i}} \pi(s)\left[u_i(t_i, s_{-i}) - u_i(s)\right] \leq \epsilon_{i,s_i} \quad & \text{for all } i,\ s_i \in \tilde{C}_i^k \\ & \text{and } t_i \in C_i \\ \sum_{s_i\in\tilde{C}_i^k} \epsilon_{i,s_i} \leq \epsilon \quad & \text{for all } i \\ \pi(s) \geq 0 \quad & \text{for all } s \in \tilde{C} \\ \sum_{s\in\tilde{C}^k} \pi(s) = 1 \quad & \end{aligned}$$
- If $\epsilon^k = 0$, terminate.
- For each player i for whom $\sum_{s_i\in\tilde{C}_i} \epsilon_{i,s_i} = \epsilon$, form $\tilde{C}_i^{k+1}$ from $\tilde{C}_i^k$ by adding in at least one strategy t_i which makes $\sum_{s_{-i}\in\tilde{C}_{-i}} \pi(s)\left[u_i(t_i, s_{-i}) - u_i(s)\right] = \epsilon_{i,s_i}$ for each $s_i \in \tilde{C}_i^k$ such that $\epsilon_{i,s_i} > 0$.
- For all other players i, let $\tilde{C}_i^{k+1} = \tilde{C}_i^k$.
- Let $k = k + 1$ and repeat.

Note that all steps of this algorithm are well-defined. First, the optimization problem is feasible. To see this let π^k be any exact correlated equilibrium of the finite game with strategy spaces $\tilde{C}_i^k$ and utilities u_i restricted to $\tilde{C}^k$; such an equilibrium exists because all finite games have correlated equilibria [3]. The u_i are bounded on C (being continuous functions on a compact set), so by making ϵ and the ϵ_{i,s_i} large, we see that π^k is a feasible solution of the problem. Second, the optimal objective value is achieved by some π^k because the space of probability measures on $\tilde{C}^k$ is compact and ϵ is bounded below by zero. Third, the set of $t_i \in C_i$ making the ϵ-correlated equilibrium constraints tight at the optimum is nonempty by optimality of π^k and continuity of u_i. This set consists only of strategies which are not in $\tilde{C}_i^k$ because we have the constraint that the deviations in utility be nonpositive for $t_i \in \tilde{C}_i^k$.

To show that Algorithm 3.1 converges, we will view it as a member of the following family of algorithms with the parameters set to $\alpha = 0$ and $\beta = 1$. Varying these parameters corresponds to adding some slack in the exact correlated equilibrium constraints and allowing some degree of suboptimality in the choice of strategies added to $\tilde{C}_i^k$ to form $\tilde{C}_i^{k+1}$. Such changes make little conceptual difference, but could be helpful in practice by making the optimization problem strictly feasible and allowing it to be solved to within a known fraction of the optimal objective value rather than all the way to optimality. We will prove that all algorithms in this family converge.

Algorithm 3.2: Fix parameters $0 \leq \alpha < \beta \leq 1$. Let $k = 0$ and for each player fix a finite subset $\tilde{C}_i^0 \subseteq C_i$.

- Choose ϵ^k to be the smallest number for which there exists π^k such that:
  - π^k is a probability distribution supported on $\tilde{C}^k$,
  - π^k is an ϵ^k-correlated equilibrium of the game,
  - π^k is not an ϵ-correlated equilibrium for any $\epsilon < \epsilon^k$,
  - π^k is an $\alpha\epsilon^k$-correlated equilibrium of the game when strategy deviations are restricted to $\tilde{C}^k$ (i.e. when the condition $t_i \in C_i$ is changed to $t_i \in \tilde{C}_i^k$ in Definition 2.1).
- If $\epsilon^k = 0$, terminate.
- For at least one value of i, form $\tilde{C}_i^{k+1}$ from $\tilde{C}_i^k$ by adding strategies $t_{i,s_i} \in C_i$ such that
$$\sum_{s \in \tilde{C}^k} \pi^k(s)\left[u_i(t_{i,s_i}, s_{-i}) - u_i(s)\right] \geq \beta\epsilon^k$$
- For all other values of i, let $\tilde{C}_i^{k+1} = \tilde{C}_i^k$.
- Let $k = k + 1$ and repeat.

Theorem 3.3: Algorithms 3.1 and 3.2 converge in the sense that $\epsilon^k \to 0$.

Proof: Suppose not, so there exists $\epsilon > 0$ and infinitely many values of k such that $\epsilon^k \geq \epsilon$. For each i let $B_i^1, \ldots, B_i^{l_i}$ be a finite open cover of C_i such that $u_i(s_i, s_{-i}) - u_i(t_i, s_{-i}) \leq \frac{1}{2}(\beta - \alpha)\epsilon$ when s_i and t_i belong to the same set B_i^l and $s_{-i} \in C_{-i}$. Such a cover exists by the compactness of the C_i and the continuity of the u_i. There are finitely many sets B_i^l in total, so there is some iteration k, which we can take to satisfy $\epsilon^k \geq \epsilon$, such that all the sets B_i^l which will ever contain an element of $\tilde{C}_i^k$ at some iteration k already do.

Note that π^k is an $\alpha\epsilon^k$-correlated equilibrium when strategy choices are restricted to $\tilde{C}_i^k$, and $\epsilon^k > 0$ so we have $\beta\epsilon^k > \alpha\epsilon^k$. By the minimality of ϵ^k, the set $\tilde{C}_i^{k+1} \setminus \tilde{C}_i^k$ is nonempty for some player i (that is to say, it is always possible to perform the third step of the algorithm). Choose such an i and let $t_{i,s_i} \in \tilde{C}_i^{k+1}$ satisfy
$$\sum_{s \in \tilde{C}^k} \pi^k(s)\left[u_i(t_{i,s_i}, s_{-i}) - u_i(s)\right] \geq \beta\epsilon^k.$$
By assumption, for any choice of $r_{i,s_i} \in \tilde{C}_i^k$ we have
$$\sum_{s \in \tilde{C}^k} \pi^k(s)\left[u_i(r_{i,s_i}, s_{-i}) - u_i(s)\right] \leq \alpha\epsilon^k,$$
so
$$\sum_{s \in \tilde{C}^k} \pi^k(s)\left[u_i(t_{i,s_i}, s_{-i}) - u_i(r_{i,s_i}, s_{-i})\right] \geq (\beta - \alpha)\epsilon^k.$$
By construction of k, we can choose $r_{i,s_i} \in \tilde{C}_i^k$ to lie in the same set B_i^l as t_{i,s_i} for each $s_i \in \tilde{C}_i^k$. Thus
$$\begin{aligned}
(\beta - \alpha)\epsilon &\leq (\beta - \alpha)\epsilon^k \\
&\leq \left|\sum_{s \in \tilde{C}^k} \pi^k(s)\left[u_i(t_{i,s_i}, s_{-i}) - u_i(r_{i,s_i}, s_{-i})\right]\right| \\
&\leq \sum_{s \in \tilde{C}^k} \pi^k(s)\left|u_i(t_{i,s_i}, s_{-i}) - u_i(r_{i,s_i}, s_{-i})\right| \\
&\leq \sum_{s \in \tilde{C}^k} \pi^k(s)\frac{(\beta - \alpha)\epsilon}{2} = \frac{(\beta - \alpha)\epsilon}{2},
\end{aligned}$$
a contradiction. ■

IV. Implementing these algorithms with semidefinite programs

To implement these algorithms for polynomial games, we must be able to do two things. First, we need to solve optimization problems with finitely many decision variables, linear objective functions and two types of constraints: nonnegativity constraints on linear functionals of the decision variables, and nonnegativity constraints on univariate polynomials whose coefficients are linear functionals of the decision variables. That is to say, we must be able to handle constraints of the form $p(t) \geq 0$ for all $t \in [-1, 1]$, where the coefficients of the polynomial p are linear in the decision variables. Second, we need to extract values of t for which such polynomial inequalities are tight at the optimum.

Both of these tasks can be done simultaneously by casting the problem as a **semidefinite program (SDP)**. An SDP is an optimization problem in which there are finitely many real decision variables, linear equality and inequality constraints, positive semidefiniteness constraints on symmetric matrices of decision variables, and a linear objective function. SDPs can be solved efficiently by interior point methods [14]. Polynomial inequalities of the above type can be expressed using semidefinite constraints via sum of squares techniques

[7]. We will summarize the necessary results, both of which are classical.

Proposition 4.1 (Markov-Lukács [6]): A univariate polynomial $p(x)$ is nonnegative on the interval $[-1, 1]$ if and only if $p(x) = s(x) + (1 - x^2)t(x)$ where s and t are both sums of squares of polynomials.

Proof: Direct algebraic manipulations show that the set of polynomials of the form $s(x) + (1 - x^2)t(x)$ where s and t are sums of squares of polynomials in x is closed under multiplication and contains all polynomials of the following forms: a for $a \geq 0$, $(x - a)^2 + b^2$ for $a, b \in \mathbb{R}$, $x - a$ for $a \leq -1$, and $a - x$ for $a \geq 1$. By assumption $p(x)$ factors as a product of terms of these types, because any real root of p in the interval $(-1, 1)$ must have even multiplicity. ∎

Proposition 4.2: A univariate polynomial $p(x) = \sum_{k=0}^{2d} p_k x^k$ of degree at most $2d$ is a sum of squares of polynomials if and only if there exists a symmetric positive semidefinite matrix $Q \in \mathbb{R}^{(d+1)\times(d+1)}$ such that $p_k = \sum_{i+j=k} Q_{ij}$ (numbering the rows and columns of Q from 0 to d).

Proof: Relating the coefficients of $p(x)$ to the entries of Q in this way is the same as writing $p(x) = \mathbf{x}^T Q \mathbf{x}$ where $\mathbf{x} = \begin{bmatrix} 1 & x & x^2 & \cdots & x^d \end{bmatrix}^T$. Thought of in this way, saying that $p(x)$ is a sum of squares is the same as saying that $Q = \sum_i q_i q_i^T$ for some column vectors q_i and in this case Q is clearly positive semidefinite. Conversely, if Q is positive semidefinite then there exists a matrix F such that $Q = F^T F$, so $p(x) = \mathbf{x}^T Q \mathbf{x} = \sum_i [F\mathbf{x}]_i^2$. ∎

We can now express the optimization problem in Algorithm 3.1 as an SDP. In this problem we have a finite number of univariate polynomials in t_i whose coefficients are linear in the decision variables $\pi(s)$ and ϵ_{i,s_i}. We wish to constrain these coefficients to allow only polynomials which are nonnegative for all $t_i \in [-1, 1]$. By Propositions 4.1 and 4.2 this is the same as asking that these coefficients equal certain linear functions of matrices (i.e. sums along antidiagonals) which are constrained to be positive semidefinite.

As a special case of convex programs, semidefinite programs have a rich duality theory which is useful for theoretical and computational purposes. In particular, SDP solvers keep track of feasible primal and dual solutions in order to determine when optimality is reached. It can be shown that the dual data obtained by an SDP solver run on this optimization problem will encode the values of t_i making the polynomial inequalities tight at the optimum [7].

The process of generating an SDP from the optimization problem in the algorithms above, solving it, and extracting an optimal solution along with t_i values from the dual can all be automated. We have done so using the SOSTOOLS MATLAB toolbox for the pre- and post-processing and SeDuMi for solving the semidefinite programs efficiently [9], [13].

Example 4.3: Consider a polynomial game with three players, choosing strategies x, y, and $z \in [-1, 1]$. Choose the utilities to be polynomials with terms up to degree 4 in all the variables with coefficients independently and identically distributed according to a normal distribution with zero mean and unit variance. Under one instantiation of the utilities Algorithm 3.1 proceeds as in Figure 4.3, which shows the value of ϵ^k and the new strategies added on each iteration. The terminal probability distribution π^6 does not display any obvious structure; in particular it is not a Nash equilibrium (product distribution).

k	ϵ^k	$\tilde{C}_x^k \setminus \tilde{C}_x^{k-1}$	$\tilde{C}_y^k \setminus \tilde{C}_y^{k-1}$	$\tilde{C}_z^k \setminus \tilde{C}_z^{k-1}$
0	0.99	$\{0\}$	$\{0\}$	$\{0\}$
1	4.16			$\{0.89\}$
2	5.76	$\{-1\}$		
3	0.57		$\{1\}$	
4	0.28	$\{0.53\}$		$\{0.50, 0.63\}$
5	0.16		$\{0.49, 0.70\}$	
6	10^{-7}		$\{-1, 0.60\}$	$\{-0.60, 0.47\}$

Fig. 1. Output of Algorithm 3.1 on a three player polynomial game with utilities of degree 4 and randomly chosen coefficients.

V. A NONCONVERGENT LIMITING CASE

Note that in the algorithms above the convergence of the sequence ϵ^k is not necessarily monotone. If we were to let $\alpha = \beta$ (which we did not allow above), the sequence would become monotone nonincreasing, but the above convergence proof would fail. This is not merely an artifact of the chosen proof; in fact when $\alpha = \beta$ Algorithm 3.2 may not converge, in the sense that ϵ^k may remain bounded away from zero. To see this, consider the following version of this algorithm for $\alpha = \beta = 1$, which is exactly the procedure proposed in [12]. The only difference between this procedure and Algorithm 3.1 is that Algorithm 3.1 has additional control over the change in utility due to deviations to strategies in $\tilde{C}_i^k$.

Algorithm 5.1 ([12]): Let $k = 0$ and for each player fix a finite subset $\tilde{C}_i^0 \subseteq C_i$.

- Minimize ϵ^k subject to the condition that π^k be an ϵ^k-correlated equilibrium supported on $\tilde{C}^k$. That is to say, let ϵ^k be the optimal value of the following optimization problem, and π^k an optimal assignment to the decision variables.

$$\begin{array}{rl} \text{minimize} & \epsilon \\ \text{s.t.} & \\ \displaystyle\sum_{s_{-i}\in\tilde{C}_{-i}^k} \pi(s)\left[u_i(t_i, s_{-i}) - u_i(s)\right] \leq \epsilon_{i,s_i} & \text{for all } i,\ s_i \in \tilde{C}_i^k \\ & \text{and } t_i \in C_i \\ \displaystyle\sum_{s_i\in\tilde{C}_i^k} \epsilon_{i,s_i} \leq \epsilon & \text{for all } i \\ \pi(s) \geq 0 & \text{for all } s \in \tilde{C}^k \\ \displaystyle\sum_{s\in\tilde{C}^k} \pi(s) = 1 & \end{array} \tag{1}$$

- If $\epsilon^k = 0$, terminate.
- For each player i for whom the second constraint is tight, let $\tilde{C}_i^{k+1}$ consist of all strategies in $\tilde{C}_i^k$ along with any strategies $t_i \in C_i$ such that the first inequality is tight for some s_i satisfying $\epsilon_{i,s_i} > 0$.
- For all other players let $\tilde{C}_i^{k+1} = \tilde{C}_i^k$.

- Let $k = k + 1$ and repeat.

It is worth noting that this procedure is still somewhat underspecified; for example there is the possibility that (1) may have multiple optima and we have not specified which is to be chosen. Nonetheless we will show that this procedure need not converge, even in games for which such subtleties do not arise.

	a	b	c
a	0	1	0
b	1	5	7
c	0	7	0

Fig. 2. A finite symmetric game with identical utilities for which Algorithm 5.1 does not converge when started with strategy sets $\tilde{C}_1^0 = \tilde{C}_2^0 = \{a\}$.

Example 5.2: Consider the game shown in Figure 2, which is symmetric and has identical utilities for both players. Let $\tilde{C}_1^0 = \tilde{C}_2^0 = \{a\}$ and use Algorithm 5.1. The only probability distribution supported on $\tilde{C}^0$ is $\delta_{(a,a)}$ which has an objective value of $\epsilon^0 = 1$. It is easy to see that $\tilde{C}_i^1$ is formed by simply adding each player's best response to a, so that $\tilde{C}_1^1 = \tilde{C}_2^1 = \{a, b\}$. We will argue that the unique optimal solution to (1) for $k = 1$ is also $\delta_{(a,a)}$, hence $\tilde{C}_i^2 = \tilde{C}_i^1$ and the algorithm gets "stuck", so that $\epsilon^k = \epsilon^0 = 1$ for all k.

For a probability distribution π, let π^T denote π with the players interchanged. By symmetry and convexity, if π is an optimal solution of (1) then so is $\frac{\pi+\pi^T}{2}$, which is a symmetric probability distribution with respect to the two players. Hence an optimal solution which is symmetric always exists. We will parametrize such distributions by $\pi = p\delta_{(a,a)} + q\delta_{(a,b)} + q\delta_{(b,a)} + r\delta_{(b,b)}$, where $p, q, r \geq 0$ and $p + 2q + r = 1$. Define a function $\zeta : C_1 \to C_1$ by $\zeta(a) = b$, $\zeta(b) = \zeta(c) = c$. Then for π to be an ϵ-correlated equilibrium it must satisfy the following condition, obtained by summing several of the inequalities in (1).

$$\begin{aligned}\epsilon &\geq \sum_{s_1 \in \tilde{C}_1^1} \epsilon_{1,s_1} \geq \sum_{s \in \tilde{C}^1} \pi(s)(u_1(\zeta(s_1), s_2) - u_1(s_1, s_2)) \\ &= p + 4q - q + 2r = 1 + q + r\end{aligned}$$

We know we can achieve $\epsilon = 1$ with $p = 1$ (i.e. $\pi = \pi^0 = \delta_{(a,a)}$), and this inequality shows that if $p < 1$ then $\epsilon > 1$. Therefore the minimal ϵ value in (1) is unity and is achieved by $\pi = \delta_{(a,a)}$. Furthermore we have shown that this is the unique symmetric probability distribution which achieves the minimal value of ϵ. Hence any other (not necessarily symmetric) optimal solution $\hat{\pi}$ satisfies $\frac{\hat{\pi}+\hat{\pi}^T}{2} = \delta_{(a,a)}$. But $\delta_{(a,a)}$ is an extreme point of the convex set of probability distributions on $\tilde{C}^1$, so we must in fact have $\hat{\pi} = \delta_{(a,a)}$. Therefore $\pi^1 = \pi^0 = \delta_{(a,a)}$ is the unique optimal solution to (1) on iteration $k = 1$, so Algorithm 5.1 gets stuck as claimed. That is, $\tilde{C}_i^k = \{a, b\}$ and $\epsilon^k = 1$ for all $k \geq 1$.

The same behavior can occur in polynomial games, as can be shown by "embedding" the above finite game in a polynomial game. For example, we can take $C_x = C_y = [-1, 1]$ and

$$\begin{aligned}u_x(x, y) = u_y(x, y) = (1 - x^2)(3y^2 + 6y + 5) \\ + (1 - y^2)(3x^2 + 6x + 5).\end{aligned}$$

Then if $\tilde{C}_x^0 = \tilde{C}_y^0 = \{-1\}$ the same analysis as above shows that $\tilde{C}_x^k = \tilde{C}_y^k = \{-1, 0\}$ and $\epsilon^k = 2$ for all $k \geq 1$.

k	ϵ^k	$\tilde{C}_x^k \setminus \tilde{C}_x^{k-1}$	$\tilde{C}_y^k \setminus \tilde{C}_y^{k-1}$
0	2	$\{-1\}$	$\{-1\}$
1	4	$\{0\}$	$\{0\}$
2	0	$\{1\}$	$\{1\}$

Fig. 3. Output of Algorithm 3.1 for a polynomial game on which Algorithm 5.1 does not converge.

Example 5.3: If we run Algorithm 3.1 on this polynomial game, the iterations proceed as in Figure 5.3. The correlated equilibrium obtained in iteration 2 is

$$\begin{aligned}\pi^2 = 0.4922\delta(x = 0, y = 1) + 0.4922\delta(x = 1, y = 0) \\ + 0.0156\delta(x = 1, y = 1),\end{aligned}$$

i.e. a probability of 0.4922 is assigned to each of the outcomes $(x, y) = (0, 1)$ and $(x, y) = (1, 0)$ and a probability of 0.0156 is assigned to $(x, y) = (1, 1)$.

VI. Conclusions

We have shown how to iteratively compute a sequence of ϵ^k-correlated equilibria of a polynomial game such that ϵ^k goes to zero. This leaves several open questions. First, in practice the ϵ^k go to zero very quickly, with ϵ falling below the numerical precision of the optimization solver within a handful of iterations. It remains to support this observation with theoretical performance guarantees. Second, the effect of the parameters α and β on the performance of Algorithm 3.2 is unknown, experimentally or theoretically. Finally, since the algorithms are already cast in an optimization framework, there is hope that they could be modified to steer the computation toward a correlated equilibrium which optimizes some quantity (say, the sum the utilities), but we leave this extension for future work.

References

[1] M. Dresher and S. Karlin. Solutions of convex games as fixed points. In H. W. Kuhn and A. W. Tucker, editors, *Contributions to the Theory of Games II*, number 28 in Annals of Mathematics Studies, pages 75 – 86. Princeton University Press, Princeton, NJ, 1953.

[2] M. Dresher, S. Karlin, and L. S. Shapley. Polynomial games. In H. W. Kuhn and A. W. Tucker, editors, *Contributions to the Theory of Games I*, number 24 in Annals of Mathematics Studies, pages 161 – 180. Princeton University Press, Princeton, NJ, 1950.

[3] S. Hart and D. Schmeidler. Existence of correlated equilibria. *Mathematics of Operations Research*, 14(1), February 1989.

[4] S. Karlin. *Mathematical Methods and Theory in Games, Programming, and Economics*, volume 2: Theory of Infinite Games. Addison-Wesley, Reading, MA, 1959.

[5] S. Karlin and L. S. Shapley. *Geometry of Moment Spaces*. American Mathematical Society, Providence, RI, 1953.

[6] M. G. Kreĭn and A. A. Nudel'man. *The Markov Moment Problem and Extremal Problems*, volume 50 of *Translations of Mathematical Monographs*. American Mathematical Society, 1977.

[7] P. A. Parrilo. *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*. PhD thesis, California Institute of Technology, May 2000.

[8] P. A. Parrilo. Polynomial games and sum of squares optimization. In *Proceedings of the 45th IEEE Conference on Decision and Control (CDC)*, 2006.

[9] S. Prajna, A. Papachristodoulou, P. Seiler, and P. A. Parrilo. *SOSTOOLS: Sum of squares optimization toolbox for MATLAB*, 2004.

[10] N. D. Stein. Characterization and computation of equilibria in infinite games. Master's thesis, Massachusetts Institute of Technology, May 2007.

[11] N. D. Stein, A. Ozdaglar, and P. A. Parrilo. Separable and low-rank continuous games. *To appear*, 2008.

[12] N. D. Stein, P. A. Parrilo, and A. Ozdaglar. Characterization and computation of correlated equilibria in infinite games. In *Proceedings of the 46th IEEE Conference on Decision and Control*, pages 759 – 764, 2007.

[13] Jos F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optim. Methods Softw.*, 11/12(1-4):625–653, 1999. Interior point methods.

[14] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49 – 95, 1996.