Polynomial-time Computation of Exact Correlated Equilibrium in Compact Games

Albert Xin Jiang Kevin Leyton-Brown
Department of Computer Science
University of British Columbia
Vancouver, BC, Canada
[jiang,kevinlb]@cs.ubc.ca

Abstract

In a landmark paper, Papadimitriou and Roughgarden described a polynomial-time algorithm ("Ellipsoid Against Hope") for computing sample correlated equilibria of concisely-represented games. Recently, Stein, Parrilo and Ozdaglar showed that this algorithm can fail to find an exact correlated equilibrium. We present a variant of the Ellipsoid Against Hope algorithm that guarantees the polynomial-time identification of exact correlated equilibrium. Our algorithm differs from the original primarily in its use of a separation oracle that produces cuts corresponding to pure-strategy profiles. Our new separation oracle can be understood as a derandomization of Papadimitriou and Roughgarden's original separation oracle via the method of conditional probabilities. We also adapt our techniques to two related algorithms that are based on the Ellipsoid Against Hope approach, Hart and Mansour's communication procedure for correlated equilibria and Huang and von Stengel's algorithm for extensive-form correlated equilibria, in both cases yielding efficient exact solutions.

JEL Classification: C63; C72

Keywords: correlated equilibrium, ellipsoid method, separation oracle, derandomization

1 Introduction

A central topic of game theory is the study of *solution concepts*, which are rules for predicting likely outcomes of the game under various models of rationality of the players. Perhaps the best-known solution concept is Nash equilibrium, a set of (randomized) strategies that are stable in the sense that no player could increase her expected utility by unilaterally deviating to a different strategy. First proposed by Aumann [1974; 1987], correlated equilibrium (CE) is another important solution concept. Whereas in a Nash equilibrium players randomize independently, in a correlated equilibrium players are allowed to coordinate their behavior based on signals from an intermediary.

A fundamental task is the *computation* of a solution concept: given a specific game instance, figuring out what the solution concept says about the likely outcomes of the game. A relatively simple such problem is the identification of a *sample*—for example, finding any Nash equilibrium of a given game. (Thus, the computational problem of finding a sample equilibrium sidesteps the issue of multiplicity of equilibria.) If the game is very small or has certain special properties, it is possible to intuit an answer to such a question in an ad hoc way. For all other cases, a methodical procedure (i.e., an *algorithm*) is required, whether the question is to be answered by pen and paper or by a computer. Economists and operations researchers have studied the computation of solution concepts since the early days of game theory, from the linear programming formulation of zero-sum games [von Neumann & Morgenstern, 1944] and Lemke and Howson's [1964] algorithm for computing a Nash equilibrium of bimatrix games, to the development of algorithms for Nash equilibria in n-player games (see, e.g., [Scarf, 1967; van der Laan *et al.*, 1987; Govindan & Wilson, 2003] and the survey by McKelvey and McLennan [1996]). One fundamental property of an algorithm is the scaling behavior of its runtime as the size of its input grows. If an algorithm runs in time polynomial in the size of its input, the algorithm is generally considered to be efficient.

In this paper we consider the problem of computing a sample correlated equilibrium given a finite, simultaneous-move game. It is known that correlated equilibria of a game can be formulated as probability distributions over pure strategy profiles satisfying certain linear constraints, and thus a CE can be found by solving a linear program (LP) without an objective function, also known as a linear feasibility program. If the game is represented in the normal form representation, in which the game's payoff function is stored as a multidimensional table with one entry for each player's payoff under each pure strategy profile, then the size of this linear feasibility program is polynomial in the size of the normal form representation of the game. Since there exist polynomial-time algorithms for solving linear feasibility programs (e.g., the ellipsoid method), a correlated equilibrium can be found in polynomial time. This attractive property of the correlated equilibrium solution concept is in contrast with the case of Nash equilibrium; recent results from the theoretical computer science community [Goldberg & Papadimitriou, 2006; Daskalakis *et al.*, 2009; Chen *et al.*, 2009] showed that the problem of finding a Nash equilibrium for games represented in normal form is unlikely to admit a polynomial-time algorithm, even if the game has only two players.[1]

The size of the normal form representation grows exponentially in the number of players. This is problematic when games involve large numbers of players: it is unreasonable to imagine even writing down such games in the normal form. Fortunately, most large games of practical interest have highly structured payoff functions, and thus it is possible to represent them compactly. Intuitively, this

[1]Some computer scientists have argued that such *complexity* results have implications on whether it is reasonable to assume that equilibria will always be reached in practice. For example, Kamal Jain has been widely quoted for the remark "if your laptop cannot find [an equilibrium], neither can the market."

helps to explain why people are able to reason about these games in the first place: we understand the payoffs in terms of simple relationships rather than in terms of enormous lookup tables. 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 [Kearns *et al.*, 2001] and action-graph games [Bhat & Leyton-Brown, 2004; Jiang *et al.*, 2011]. Thus, it is also desirable that an algorithm that computes a given solution concept be able to work directly with compactly represented games. The previously mentioned algorithm for CE no longer runs in polynomial time when the input is compactly represented, since the size of the linear feasibility program for CE can be exponential in the input size; furthermore, since a solution vector of such a linear feasibility program (i.e., a CE) has an exponential number of components, specifying such a CE can require an exponential amount of storage space.

The "Ellipsoid Against Hope" algorithm [Papadimitriou, 2005; Papadimitriou & Roughgarden, 2008] is a polynomial-time method for identifying a (polynomial-size representation of a) CE, given a game representation satisfying two properties: *polynomial type*, which requires that the number of players and the number of actions for each player are bounded by polynomials in the size of the representation, and the *polynomial expectation property*, which requires access to a polynomial-time algorithm that computes the expected utility of any player under any mixed-strategy profile. Many existing compact game representations (including graphical games, symmetric games, congestion games, polymatrix games and action-graph games) satisfy these properties. This important result extends CE's attractive computational properties to the case of compactly represented games; note in contrast that the problem of finding a Nash equilibrium remains computationally difficult for many of the same compact game representations [Goldberg & Papadimitriou, 2006; Jiang *et al.*, 2011].

At a high level, the Ellipsoid Against Hope algorithm works on an unbounded primal LP formulation (P) of CE and its infeasible dual (D). Although (D) has an exponential number of constraints, one can apply the *ellipsoid method*, an algorithm for solving LPs that terminates in polynomial time even when the number of constraints is exponential, as long as a *separation oracle* is provided [Grötschel *et al.*, 1988]. The ellipsoid algorithm performs a polynomial number of iterations, at each iteration maintaining an ellipsoid that is guaranteed to contain a solution if one exists. It starts with a ball with a large enough radius to guarantee this containment, and iteratively shrinks this ellipsoid until either a solution is found, or the volume of the ellipsoid is small enough to certify the infeasibility of the LP. At each iteration it queries a separation oracle, which determines whether the queried point is feasible, and if not provides a *cutting plane*, which is a hyperplane that separates the queried point and the feasible set. It turns out that an efficiently-computable separation oracle exists for (D), and as a result the ellipsoid method terminates in polynomial time. However, since we already know that (D) is infeasible, what do we gain from verifying this fact computationally? Papadimitriou and Roughgarden [2008] showed that such a run of the ellipsoid method on this infeasible dual (D) provides enough information to find a feasible solution of the primal (P), which yields a CE of the game. Specifically, they argue that the polynomial-sized LP (D') formed by the generated cutting planes must also be infeasible. Solving the dual of (D') yields a CE, represented as a mixture of product distributions, with each product distribution corresponding to a cutting plane generated by the separation oracle.

1.1 Recent Uncertainty About the Complexity of Exact CE

In a recent paper, Stein, Parrilo and Ozdaglar [2010] raised two interrelated concerns about the Ellipsoid Against Hope algorithm. First, they identified a symmetric 3-player, 2-action game with rational[2] utilities on which the algorithm can fail to compute an exact CE. Indeed, they showed that the same problem arises on this game for a whole class of related algorithms. Specifically, if an algorithm (a) outputs a rational solution, (b) outputs a convex combination of product distributions, and (c) outputs a convex combination of symmetric product distributions when the game is symmetric, then that algorithm fails to find an exact CE on their game, because the only CE of their game that satisfies properties (b) and (c) has irrational probabilities. This implies that any algorithm for exact rational CE must violate (b) or (c).

Second, Stein, Parrilo and Ozdaglar also showed that the original analysis by Papadimitriou and Roughgarden [2008] incorrectly handles certain numerical precision issues, which we now briefly describe. Recall that a run of the ellipsoid method requires as inputs an initial bounding ball with radius R and a volume bound v such that the algorithm stops when the ellipsoid's volume is smaller than v. To correctly certify the (in)feasibility of an LP using the ellipsoid method, R and v need to be set to appropriate values, which depend on the maximum encoding size of a constraint in the LP. However (as pointed out by Papadimitriou and Roughgarden [2008]), each cut returned by the separation oracle is a convex combination of the constraints of the original dual LP (D) and thus may require more bits to represent than any of the constraints in (D); as a result, the infeasibility of the LP (D') formed by these cuts is not guaranteed. Papadimitriou and Roughgarden [2008] proposed a method to overcome this difficulty, but Stein *et al.* showed that this method is insufficient for finding an exact CE. For the related problem of finding an approximate correlated equilibrium (ϵ-CE), Stein *et al.* gave a slightly modified version of the Ellipsoid Against Hope algorithm that runs in time polynomial in $\log \frac{1}{\epsilon}$ and the game representation size.[3] For problems that can have necessarily irrational solutions, it is typical to consider such approximations as efficient; however, the computation of a sample CE is not such a problem, as there always exists a rational CE in a game with rational utilities, since CE are defined by linear constraints. It remained an open problem to determine whether the Ellipsoid Against Hope algorithm can be modified to compute an exact, rational correlated

[2] In what follows, by "rational" we refer to rational numbers (ratios of integers) rather than an assumption about agents maximizing their own utilities.

[3] An ϵ-CE is a distribution that violates the CE incentive constraints by at most ϵ.

equilibrium.[4]

1.2 Our Results

In this paper, we resolve this open problem by deriving a variant of the Ellipsoid Against Hope algorithm algorithm that computes in polynomial time an exact (and rational) correlated equilibrium given a game representation that has polynomial type and satisfies the polynomial expectation property. Our modification to the Ellipsoid Against Hope algorithm follows an alternate approach, which completely sidesteps the issues just discussed. Specifically, our approach is based on the observation that if we use a separation oracle (for the same dual LP formulation proposed by Papadimitriou and Roughgarden [2008]) that generates cuts corresponding to pure-strategy profiles (instead of Papadimitriou and Roughgarden's separation oracle that generates nontrivial product distributions), then these cuts are actual constraints in the dual LP, as opposed to convex combinations of constraints. As a result we no longer encounter the numerical accuracy issues that prevented the previous approaches from finding exact correlated equilibria. Both the resulting algorithm and its analysis are also considerably simpler than the original: standard techniques from the theory of the ellipsoid method are sufficient to show that our algorithm computes an exact CE using a polynomial number of oracle queries.

The key issue is the identification of pure-strategy-profile cuts. It is relatively straightforward to show that such cuts always exist: since the product distribution generated by the Ellipsoid Against Hope algorithm ensures the nonnegativity of a certain expected value, then by a simple application of the probabilistic method there must exist a pure-strategy profile that also ensures the nonnegativity of that expected value. The key is to go beyond this nonconstructive proof of existence to also *compute* pure-strategy-profile cuts in polynomial time. We show how to do this by applying the method of conditional probabilities [Erdős & Selfridge, 1973; Spencer, 1994; Raghavan, 1988], an approach for derandomizing probabilistic proofs of existence. At a high level, our new separation oracle begins with the product distribution generated by Papadimitriou and Roughgarden's separation oracle, then sequentially fixes a pure strategy for each player in a way that guarantees that the corresponding conditional expectation given the choices so far remain nonnegative. Since our separation oracle goes through players sequentially, the cuts generated can be asymmetric even for symmetric games. Indeed, we can confirm (see Section 4.2) that it makes such asymmetric cuts on Stein, Parrilo and Ozdaglar's symmetric game—thus violating their condition (c)—because our algorithm always identifies a rational CE.

Another effect of our use of pure-strategy-profile cuts is that the correlated equilibria generated by our algorithm are guaranteed to have polynomial-sized supports; i.e., they are mixtures over a polynomial number of pure strategy

[4] In a recent addendum to their original paper, Papadimitriou and Roughgarden [2010] acknowledged the flaw in the original algorithm. We note also that Stein *et al.* subsequently withdrew their paper from arXiv. It is our belief that their technical results are nevertheless correct; we discuss them here because they help to motivate our alternate approach.

profiles. Correlated equilibria with polynomial-sized supports are known to exist in every game (e.g., [Germano & Lugosi, 2007]); intuitively this is because CE are defined by a polynomial number of linear constraints, so a basic feasible solution of the linear feasibility program would have a polynomial number of non-zero entries. Such small-support correlated equilibria are more natural solutions than the mixtures of product distributions produced by the Ellipsoid Against Hope algorithm: because of their simpler form they require fewer bits to represent and fewer random bits to sample from; furthermore, verifying whether a given polynomial-support distribution is a CE only requires evaluating the utilities of a polynomial number of pure strategy profiles, whereas verifying whether a mixture of product distributions is a CE requires evaluating expected utilities under product distributions, which is generally more expensive. No tractable algorithm has previously been proposed for identifying such a CE; thus, our algorithm is the first to compute in polynomial time a CE with polynomial support given a compactly-represented game. In fact, we show that any CE computed by our algorithm corresponds to a basic feasible solution of the linear feasibility program that defines CE, and is thus an extreme point of the set of CE of the game.

Since Papadimitriou and Roughgarden's [2008] proposal of the Ellipsoid Against Hope algorithm for computing a CE, other researchers have proposed algorithms for related problems based on a similar approach (which we call the Ellipsoid Against Hope approach): first "solving" an infeasible LP using the ellipsoid method with some separation oracle, then arguing that the LP formed by the cutting planes is also infeasible, and finally solving the dual of the latter polynomial-sized LP. For example, Hart and Mansour [2010] considered the setting where each player initially knows only her own utility function, and proposed a communication procedure that finds a CE with polynomial communication complexity using a straightforward adaptation of the Ellipsoid Against Hope algorithm. Huang and von Stengel [2008] proposed a polynomial-time algorithm for computing a extensive-form correlated equilibrium (EFCE) [von Stengel & Forges, 2008], a solution concept for extensive-form games, by applying the Ellipsoid Against Hope approach to the LP formulation of EFCE. For both algorithms, the separation oracle outputs a mixture of the original constraints, and hence the flaws of the Ellipsoid Against Hope algorithm pointed out by Stein *et al.* [2010] also apply. We show that our techniques can be adapted to these two algorithms, yielding in both cases exact solutions with polynomial-sized supports. In particular, we replace the original separation oracles with "purified" versions that output cutting planes corresponding to the original constraints.

The rest of the paper is organized as follows. We start with basic definitions and notation in Section 2. In Section 3 we summarize Papadimitriou and Roughgarden's Ellipsoid Against Hope algorithm. In Section 4 we describe our algorithm and prove its correctness. In Sections 5 and 6 we describe our fixes to Hart and Mansour's [2010] and Huang and von Stengel's [2008] algorithms respectively, and Section 7 concludes.

2 Preliminaries

We largely follow the notation of Papadimitriou [2005] and Papadimitriou and Roughgarden [2008]. Consider a simultaneous-move game with n players. Denote a player p, and player p's set of pure strategies (i.e., actions) S_p. Let $m = \max_p |S_p|$. Denote a pure strategy profile $s = (s_1, \ldots, s_n) \in S$, with s_p being player p's pure strategy. Denote by S_{-p} the set of partial pure strategy profiles of the players other than p. Player p's utility under pure strategy profile s is u^p_s. We assume that utilities are nonnegative integers (but results in this paper can be straightforwardly adapted to rational utilities). Denote the largest utility of the game as u. Thus each utility value of the game can be encoded by at most $\log_2 u$ bits.

A *correlated distribution* is a probability distribution over pure strategy profiles, represented by a vector $x \in \mathbb{R}^M$, where $M = \prod_p |S_p|$. Then x_s is the probability of pure strategy profile s under the distribution x. A correlated distribution x is a *product distribution* when it can be achieved by each player p randomizing independently over her actions according to some distribution x^p, i.e., $x_s = \prod_p x^p_{s_p}$. Such a product distribution is also known as a mixed-strategy profile, with each player p playing the mixed strategy x^p.

A *game representation* is a method for encoding the information needed to specify a game, i.e., to specify its number of players n, its set of pure strategies S_p for each player p, and its utilities $\{u^p_s\}$. An *instance* of a game representation is a game encoded in that representation. The *size* of an instance is the amount of data required to specify it. For example, the normal form representation explicitly stores each utility value u^p_s in a cell of a multidimensional table. The size of this table is $n \prod_p |S_p| = nM$, which dominates the amount of data required to specify n and $|S_p|$. Thus an instance of the normal form representation has size $\Theta(nM)$.

For games with structured utilities, it is possible to have a more compact game representation. One type of structure is symmetry. A game is *player-symmetric* when all players are identical and interchangeable. In a player-symmetric game, a player's utility depends only on the player's chosen action and the *configuration*, which is the vector of integers specifying the number of players choosing each of the actions. As a result, player-symmetric games can be represented more compactly than games in normal form: we only need to specify a utility value for each action and each configuration. For a player-symmetric game with n players and m actions per player, the number of configurations is $\binom{n+m-1}{m-1}$. With fixed m, this grows like n^{m-1}, and $\Theta(n^{m-1})$ numbers are required to specify the game.

Throughout the paper (except Sections 5 and 6) we assume that a game is given in a game representation satisfying two properties, following Papadimitriou and Roughgarden [2008]:

- *polynomial type*: for all instances of the game representation, the number of players and the number of pure strategies for each player are bounded by polynomials in the size of the game instance.

- *the polynomial expectation property*: we have access to an algorithm that given an instance of the game representation computes the expected utility of any player p under any product distribution x, i.e., $\sum_{s \in S} u_s^p x_s$, in time polynomial in the size of the game instance.

The normal form representation satisfies these properties, with the corresponding expected utility algorithm being the trivial one that directly computes the sum $\sum_{s \in S} u_s^p x_s$. Papadimitriou and Roughgarden [2008] showed that many compact game representations, including player-symmetric games, graphical games, polymatrix games and congestion games, also satisfy these properties; they gave a polynomial-time expected utility algorithm for each case. Jiang *et al.* [2011] showed that action-graph games—games represented in a language that unifies these and other existing compact representations—also satisfy these properties.

Definition 2.1. *A correlated distribution x is a* correlated equilibrium *(CE) if it satisfies the following* incentive constraints*: for each player p and each pair of her actions $i, j \in S_p$,*

$$\sum_{s \in S_{-p}} [u_{is}^p - u_{js}^p] x_{is} \geq 0, \tag{1}$$

where the subscript "is" (respectively "js") denotes the pure strategy profile in which player p plays i (respectively j) and the other players play according to the partial profile $s \in S_{-p}$.

Intuitively, when a trusted intermediary draws a strategy profile s from this distribution, privately announcing to each player p her own component s_p, p will have no incentive to choose another strategy, assuming others follow the suggestions. We write these incentive constraints in matrix form as $Ux \geq 0$. Thus U is an $N \times M$ matrix, where $N = \sum_p |S_p|^2$. The rows of U, corresponding to the left-hand sides of the constraints (1), are indexed by (p, i, j) where p is a player and $i, j \in S_p$ are a pair of p's actions. Denote by U_s the column of U corresponding to pure strategy profile s. These incentive constraints, together with the constraints

$$x \geq 0, \quad \sum_{s \in S} x_s = 1, \tag{2}$$

which ensure that x is a probability distribution, form a linear feasibility program that defines the set of CE. The largest value in U is at most u.

We define the *support* of a correlated equilibrium x as the set of pure strategy profiles assigned positive probability by x. Germano and Lugosi [2007] showed that for any n-player game, there always exists a correlated equilibrium with support size at most $1 + \sum_p |S_p|(|S_p| - 1) = N + 1 - \sum_p |S_p|$. Intuitively, such correlated equilibria are basic feasible solutions of the linear feasibility program for CE, i.e., vertices of the polyhedron defining the feasible region. Furthermore, these basic feasible solutions involve only rational numbers for games with rational payoffs (see e.g. Lemma 6.2.4 of [Grötschel *et al.*, 1988]).

3 The Ellipsoid Against Hope Algorithm

In this section, we summarize Papadimitriou and Roughgarden's [2008] Ellipsoid Against Hope algorithm for finding a sample CE, which can be seen as an efficiently constructive version of earlier proofs [Hart & Schmeidler, 1989; Nau & McCardle, 1990; Myerson, 1997] of the existence of CE. We will concentrate on the main algorithm and only briefly point out the numerical issues discussed at length by both Papadimitriou and Roughgarden [2008] and Stein *et al.* [2010], as our analysis will ultimately sidestep these issues.

Papadimitriou and Roughgarden's approach considers the linear program

$$\max \sum_{s\in S} x_s \qquad (P)$$
$$Ux \geq 0, \qquad x \geq 0,$$

which is modified from the linear feasibility program for CE by replacing the constraint $\sum_{s\in S} x_s = 1$ from (2) with the maximization objective. (P) either has $x = 0$ as its optimal solution or is unbounded; in the latter case, taking a feasible solution and scaling it to be a distribution yields a correlated equilibrium. Thus one way to prove the existence of CE is to show the infeasibility of the dual problem

$$U^T y \leq -1, \qquad y \geq 0. \qquad (D)$$

The Ellipsoid Against Hope algorithm uses the following lemma, versions of which were also used by Nau and McCardle [1990] and Myerson [1997].

Lemma 3.1 ([Papadimitriou & Roughgarden, 2008]). *For every dual vector $y \geq 0$, there exists a product distribution x such that $xU^T y = 0$.*[5] *Furthermore there exists an algorithm that given any $y \geq 0$, computes the corresponding x (represented by $x^1, \ldots, x^n$) in time polynomial in n and m.*

We will not discuss the details of this algorithm; we will only need the facts that the resulting x is a product distribution and can be computed in polynomial time. Note also that the resulting x is player-symmetric if the game is player-symmetric and y is player-symmetric. Lemma 3.1 implies that the dual problem (D) is infeasible (and therefore a CE must exist): $xU^T y$ is a convex combination of the left hand sides of the rows of the dual, and for any feasible y the result must be less than or equal to -1.

The Ellipsoid Against Hope algorithm runs the ellipsoid algorithm on the dual (D), with the algorithm from Lemma 3.1 as separation oracle, which we call the the Product Separation Oracle. At each step of the ellipsoid algorithm, the separation oracle is given a dual vector $y^{(i)}$. The oracle then generates the corresponding product distribution $x^{(i)}$ and indicates to the ellipsoid algorithm that $(x^{(i)}U^T)y \leq -1$ is violated by $y^{(i)}$. The ellipsoid algorithm will stop after a

[5]For notational simplicity, throughout the paper we treat a vector (in this case x) as a row vector when it is multiplied to the left of a matrix.

polynomial number of steps and determine that the program is infeasible. Let X be the matrix whose rows are the generated product distributions $x^{(1)}, \dots, x^{(L)}$.

Consider the linear program

$$[XU^T]y \le -1, \qquad y \ge 0, \tag{D'}$$

and observe that the rows of $[XU^T]y \le -1$ are the cuts generated by the ellipsoid method. If we apply the same ellipsoid method to (D') and use a separation oracle that returns the cut $x^{(i)}U^T y \le -1$ given query $y^{(i)}$, the ellipsoid algorithm would go through the same sequence of queries $y^{(i)}$ and cutting planes $x^{(i)}U^T y \le -1$ and return infeasible. Presuming that numerical problems do not arise,[6] we will find that (D') is infeasible. This implies that its dual $[UX^T]\alpha \ge 0$, $\alpha \ge 0$ is unbounded and has polynomial size, and thus can be solved for a nonzero feasible α. We can thus scale α to obtain a probability distribution. We then observe that $X^T\alpha$ satisfies the incentive constraints (1) and the probability distribution constraints (2) and is therefore a correlated equilibrium. The distribution $X^T\alpha$ is the mixture of product distributions $x^{(1)}, \dots, x^{(L)}$ with weights α, and thus can be represented in polynomial space and can be efficiently sampled from.

One issue remains. Although the matrix XU^T is polynomial sized, computing it using matrix multiplication would involve an exponential number of operations. On the other hand, entries of XU^T are differences between expected utilities that arise under product distributions. Since we have assumed that the game representation admits a polynomial-time algorithm for computing such expected utilities, XU^T can be computed in polynomial time.

Lemma 3.2 ([Papadimitriou & Roughgarden, 2008])**.** *There exists an algorithm that given a game representation with polynomial type and satisfying the polynomial expectation property, and given an arbitrary product distribution x, computes xU^T in polynomial time. As a result, XU^T can be computed in polynomial time.*

4 Our Algorithm

In this section we present our modification of the Ellipsoid Against Hope algorithm, and prove that it computes exact CE. There are two key differences between our approach and the original algorithm for computing approximate CE.

[6] Since each row of (D')'s constraint matrix XU^T may require more bits to represent than any row of the constraint matrix U^T for (D), running the ellipsoid algorithm on (D') with the original bounding ball and volume lower bound for (D) would not be sound, and as a result (D') is not guaranteed to be infeasible. Indeed, Stein *et al.* [2010] showed that when running the algorithm on their symmetric game example, (D') would remain feasible, and thus the output of the algorithm would not be an exact CE. Furthermore, since the only CE of that game that is a mixture of symmetric product distributions is irrational, there is no way to resolve this issue without breaking at least one of the symmetry and product distribution properties of the Ellipsoid Against Hope algorithm. For more on these issues and possible alternative ways to address them than that presented here, please see Papadimitriou and Roughgarden [2008]; Stein *et al.* [2010]; Papadimitriou and Roughgarden [2010].

1. Our modified separation oracle produces pure-strategy-profile cuts;

2. The algorithm is simplified, no longer requiring a special mechanism to deal with numerical issues (because pure-strategy-profile cuts can be represented directly as rows of (D)'s constraint matrix).

4.1 The Purified Separation Oracle

We start with a "purified" version of Lemma 3.1.

Lemma 4.1. *Given any dual vector $y \geq 0$, there exists a pure strategy profile s such that $(U_s)^T y \geq 0$.*

Remark. Lemma 4.1 only establishes the existence of such an s. Later in this section we will prove Lemma 4.2, an efficiently constructive version of this lemma. Although Lemma 4.1 is technically redundant, we nevertheless give its simple proof to help provide intuition.

Proof. Recall that Lemma 3.1 states that given dual vector $y \geq 0$, a product distribution x can be computed in polynomial time such that $xU^T y = 0$. Since $x[U^T y]$ is a convex combination of the entries of the vector $U^T y$, there must exist some nonnegative entry of $U^T y$. In other words, there exists a pure strategy profile s such that $(U_s)^T y \geq xU^T y = 0$. □

The proof of Lemma 4.1 is a straightforward application of the probabilistic method: since $xU^T y$ is the expected value of $(U_s)^T y$ under distribution x, which we denote $E_{s\sim x}[(U_s)^T y]$, the nonnegativity of this expectation implies the existence of some s such that $(U_s)^T y \geq 0$. Like many other probabilistic proofs, this proof is not efficiently constructive; note that there are an exponential number of possible pure strategy profiles.

It turns out that for game representations with polynomial type and satisfying the polynomial expectation property, an appropriate s can indeed be identified in polynomial time. Our approach can be seen as derandomizing the probabilistic proof using the method of conditional probabilities [Erdős & Selfridge, 1973; Spencer, 1994; Raghavan, 1988]. At a high level, for each player p our algorithm picks a pure strategy s_p, such that the conditional expectation of $(U_s)^T y$ given the choices so far remains nonnegative. This requires us to compute the conditional expectations, but this can be done efficiently using the expected utility subroutine guaranteed by the polynomial expectation property.

Lemma 4.2. *There exists a polynomial-time algorithm that given*

- *an instance of a game in a representation satisfying polynomial type and the polynomial expectation property,*

- *a polynomial-time subroutine for computing expected utility under any product distribution (as guaranteed by the polynomial expectation property), and*

Algorithm 1 Computes a pure strategy profile s such that $(U_s)^T y \geq 0$.

1. Given $y \geq 0$, identify a product distribution x satisfying $xU^T y = 0$, using the algorithm described in Lemma 3.1.
2. Sequentially for each player $p \in \{1, \dots, n\}$,
   (a) iterate through actions $s_p \in S_p$, and compute $x_{(p \to s_p)} U^T$ using the algorithm described in Lemma 3.2, until we find an action $s_p^* \in S_p$ such that $\left[x_{(p \to s_p^*)} U^T\right] y \geq 0$.
   (b) set x to be $x_{(p \to s_p^*)}$.
3. The resulting x corresponds to a pure strategy profile s. Output s.

- *a dual vector $y \geq 0$,*

finds a pure strategy profile $s \in S$ such that $(U_s)^T y \geq 0$.

Proof. Given a product distribution x, let $x_{(p \to s_p)}$ be the product distribution in which player p plays s_p and all other players play according to x. Since x is a product distribution, $x_{(p \to s_p)} U^T y$ is the conditional expectation of $(U_s)^T y$ given that p plays s_p, and furthermore we have for any p,

$$xU^T y = \sum_{s_p} \left[x_{(p \to s_p)} U^T y\right] x^p_{s_p}. \tag{3}$$

Since x^p is a distribution, the right hand side of (3) is a convex combination and thus there must exist an action $s_p \in S_p$ such that $x_{(p \to s_p)} U^T y \geq xU^T y$. Since $x_{(p \to s_p)}$ is a product distribution, this process can be repeated for each player to yield a pure strategy profile s such that $(U_s)^T y \geq xU^T y$. Since we can get a product distribution x with $xU^T y = 0$, this process yields $(U_s)^T y \geq 0$. This is formalized in Algorithm 1.

We now consider the running time of Algorithm 1. We observe that x remains a product distribution throughout the algorithm and can thus be represented by its marginals $x^1, \dots, x^n$, requiring only polynomial space. Due to the polynomial expectation property, the algorithm described in Lemma 3.2 is polynomial, which implies that in Step 2a, for each $s_p \in S_p$, $x_{(p \to s_p)} U^T$ can be computed in polynomial time. Since Step 2a requires at most $|S_p|$ such computations, and since polynomial type implies that n and $|S_p|$ are polynomial in the input size, the algorithm runs in polynomial time. □

A straightforward corollary is the following:

Corollary 4.3. *Algorithm 1 can be used as a separation oracle for the dual LP (D) in the Ellipsoid Against Hope algorithm: for each query point y, the oracle computes the corresponding pure-strategy profile s according to Algorithm 1 and* *returns the half space $(U_s)^T y \leq -1$. We call this the Purified Separation Oracle. This separation oracle has the following properties:*

- *Each returned half space is one of the constraints of (D).*
- *Since Algorithm 1 iterates through the players sequentially, the generated pure-strategy profiles can be asymmetric even for symmetric games and symmetric y.*
- *Since a pure-strategy profile is a special case of a product distribution, the resulting pure-strategy profile s also satisfies Lemma 3.1, with x being the basis vector corresponding to s, i.e., with $x_s = 1$ and its other entries being zero.*

4.2 The Simplified Ellipsoid Against Hope Algorithm

We now modify the Ellipsoid Against Hope Algorithm by replacing the Product Separation Oracle with our Purified Separation Oracle. The rows of X in (D') become basis vectors corresponding to the pure-strategy profiles generated by the oracle. Thus, we can write (D') as

$$(U')^T y \leq -1, \qquad y \geq 0, \tag{D''}$$

where the matrix $U' \equiv UX^T$ consists of the columns $U_{s^{(i)}}$ that correspond to pure-strategy profiles $s^{(i)}$ generated by the separation oracle. Note that each constraint of (D'') is also one of the constraints of (D), and as a result neither the coefficients nor the right-hand sides of (D'') have bit complexities greater than in (D). Therefore, a starting ball and volume lower bound that are valid for a run of the ellipsoid method on (D) is also valid for (D''). We thus avoid the precision issues faced by the Ellipsoid Against Hope algorithm, and it is sufficient to use standard values for the initial radius and volume lower bound, and standard perturbation methods for dealing with non-full-dimensional solutions. The resulting CE is a mixture over a polynomial number of pure strategy profiles. We can make a further conceptual simplification of the algorithm: instead of using X as in the Ellipsoid Against Hope algorithm, we can directly treat the generated pure-strategy profiles as columns of U, and use U' in place of UX^T.

We now formally state and prove our result. Note that although we only briefly discussed the way numerical issues are addressed in the original Ellipsoid Against Hope algorithm in Section 3, we do go into detail about how our algorithm ensures its own numerical accuracy. That task is comparatively easy, as it is sufficient for us to apply standard techniques from the theory of the ellipsoid method. Our analysis makes use of the following lemma from Grötschel *et al.* [1988].

Lemma 4.4 (Lemma 6.2.6, [Grötschel *et al.*, 1988])**.** *Let $P = \{y \in \mathbb{R}^N | Ay \leq b\}$ be a full-dimensional polyhedron defined by the system of inequalities, with the encoding length of each inequality at most φ. Then P contains a ball with radius $2^{-7N^3\varphi}$. Moreover, this ball is contained in the ball with radius $2^{5N^2\varphi}$ centered at 0.*

Algorithm 2 Computes an exact rational CE given a game representation satisfying polynomial type and the polynomial expectation property.

1. Apply the ellipsoid method to (D), using the Purified Separation Oracle, a starting ball with radius of $R = u^{5N^3}$ centered at 0, and stopping when the volume of the ellipsoid is below $v = \alpha_N u^{-7N^5}$, where α_N is the volume of the N-dimensional unit ball.

2. Form the matrix U' whose columns are the $U_{s^{(1)}}, \ldots, U_{s^{(L)}}$ generated by the separation oracle during the run of the ellipsoid method.

3. Compute a basic feasible solution x' of the linear feasibility program

$$U'x' \geq 0, \qquad x' \geq 0, \qquad \mathbf{1}^T x' = 1, \tag{P^*}$$

   by applying the ellipsoid method on the explicitly represented (P^*) and recovering a basis using, e.g., Algorithm 4.2 of Dantzig and Thapa [2003].

4. Output x' and $s^{(1)}, \ldots, s^{(L)}$, interpreted as a distribution over pure-strategy profiles $s^{(1)}, \ldots, s^{(L)}$ with probabilities x'.

We note that Lemma 4.4's only restriction on P is full dimensionality; we do not need to assume that P is bounded, or that P has full row rank.

Theorem 4.5. *Given a game representation with polynomial type and satisfying the polynomial expectation property, Algorithm 2 computes an exact and rational CE with support size at most* $1 + \sum_p |S_p|(|S_p| - 1)$ *in polynomial time.*

Proof. We begin by proving the correctness of Algorithm 2. This part of the proof has three steps. (i) We show that the ellipsoid method[7] in Step 1 certifies that (D) (and therefore (D'')) is either infeasible or not full dimensional. Observe that this argument is not sufficient to rule out the possibility that (D'') has a non-full-dimensional (yet non-empty) feasible set. Intuitively this is because the ellipsoid method, which relies on shrinking the volume of the candidate set, is not able to distinguish between infeasibility and non-full-dimensional feasible sets. (ii) To overcome this, we define (4), a perturbed version of (D''), and show that it is infeasible. (iii) We show that it is sufficient for our purposes to work with (4), because a feasible and normalized solution of (4)'s unbounded dual is a CE, and can be computed by Step 3. Having proven the correctness of Algorithm 2, we next show that the CE identified by the algorithm has the required support size, and finally demonstrate that the algorithm terminates in polynomial time.

First, we will show that the ellipsoid method in Step 1 correctly certifies that the feasible set of (D) is either empty or not full dimensional. Suppose

[7] Note that the ellipsoid method is used twice in Algorithm 2, once in Step 1 to generate U', and once in Step 3 on a polynomial-sized system.

the contrary, i.e., the feasible set of (D) is non-empty and full dimensional. Recall that the utilities of the games are nonnegative integers that are at most u. Since the encoding length of each constraint of (D) is at most $N \log_2 u$, then by Lemma 4.4, the feasible set must contain a ball with radius u^{-7N^4}, and thus volume $\alpha_N u^{-7N^5}$, and furthermore this ball must be contained in the ball with radius u^{5N^3} centered at 0, which is the initial ball of our ellipsoid method in Step 1. Since at the end of Step 1 the ellipsoid method certifies that the intersection of the initial ball and the feasible set has volume less than $v = \alpha_N u^{-7N^5}$, we reach a contradiction and therefore either the LP (D) must be infeasible or the feasible set must not be full dimensional. Recall that the LP (D'') is formed by collecting the generated cutting planes, as done in Step 2 of Algorithm 2. Since the largest magnitude of the coefficients in (D'') is also u, Step 1 is also a valid run for (D'') and therefore either (D'') must be infeasible or the feasible set of (D'') must not be full dimensional.

Second, we show that a perturbed version of (D'') is infeasible. Fix $\rho > 1$. Consider the following LP, which is formed by modifying the constraints $(U')^T y \leq -1$ of (D'') by multiplying the RHS by ρ:

$$\begin{aligned} &\min 0 \\ (U')^T &y \leq -\rho \mathbf{1} \\ &y \geq 0. \end{aligned} \tag{4}$$

We claim that (4) is infeasible. Suppose otherwise: then there exists a $y \in \mathbb{R}^N$ such that $y \geq 0$ and $(U')^T y \leq -\rho \mathbf{1}$. Let $y' \in \mathbb{R}^N$ be a vector such that $0 \leq y'_j - y_j \leq \frac{\rho-1}{Nu}$ for all j. Then $y' \geq 0$, and each component s of $U'^T y'$ satisfies

$$(U'_s)^T y' \leq (U'_s)^T y + \frac{\rho - 1}{Nu} \sum_j \left|U'^j_s\right| \tag{5}$$

$$\leq -\rho + (\rho - 1) \tag{6}$$

$$= -1.$$

The first inequality (5) holds because $(U'_s)^T (y' - y) = \sum_j U'^j_s (y'_j - y_j) \leq \sum_j \left|U'^j_s\right| \left|y'_j - y_j\right| \leq \sum_j \left|U'^j_s\right| \frac{\rho-1}{Nu}$. The second inequality (6) holds because each U'^j_s is an entry of the constraint matrix U of (P), with absolute value at most u. Thus, any such y' is feasible for (D''). However, the set of all such vectors y' is a full-dimensional cube. This contradicts the fact that (D'') is either infeasible or not full dimensional, and therefore (4) is infeasible.

This means that (4)'s dual

$$\begin{aligned} &\max \rho \mathbf{1}^T x' \\ &U' x' \geq 0 \\ &x' \geq 0 \end{aligned} \tag{7}$$

is unbounded (since it is feasible, e.g., with $x' = 0$). Then a nonzero feasible vector x' is (after normalization) a distribution over the pure strategy profiles corresponding to columns of U'. Treating it as a sparse representation of a correlated distribution x, it satisfies the feasibility program for CE and is therefore an exact CE.

This CE is exact but its support size could be greater than $1+\sum_p |S_p|(|S_p|-1)$ (although as we argue below it is still polynomial). To get a CE with the required support size, we notice that since (7) is unbounded, a feasible solution of the bounded linear feasibility program (P^*) is a CE, which we solve in Step 3 of Algorithm 2. Note that (P^*) has the same number of constraints as the feasibility program for CE defined by (1) and (2), and that for each player p and action $i \in S_p$, the incentive constraint (p, i, i) corresponds to deviating from action i to itself and is therefore redundant. Thus the number of bounding constraints of (P^*) is at most $1 + \sum_p |S_p|(|S_p| - 1)$ and therefore a basic feasible solution x' of (P^*), which is the output of Algorithm 2, will have the required support size. Since the coefficients and right-hand sides of (P^*) are rational, then (by e.g. Lemma 6.2.4 of Grötschel *et al.* [1988]) its basic feasible solution x' is also rational and can be represented using at most $4N^3 \log u$ bits.

We now consider the running time of Algorithm 2. Since Step 1 is a standard run of the ellipsoid method, it terminates in a polynomial number of iterations. For example if we use the ellipsoid algorithm presented in Theorem 3.2.1 of Grötschel *et al.* [1988], then by Lemma 3.2.10 of Grötschel *et al.* [1988] the ratio between volumes of successive ellipsoids $\mathrm{vol}(E_{k+1})/\mathrm{vol}(E_k) \leq e^{-1/(5N)}$. With the volume of the initial ellipsoid at most $\alpha_N R^N$ and stopping when volume is below v, the number of iterations L is at most

$$\begin{aligned}
&5N \left[\ln(\alpha_N R^N) - \ln v\right] \\
&= 5N \left[5N^4 \ln u + 7N^5 \ln u\right] \\
&= O(N^6 \ln u),
\end{aligned}$$

which is polynomial in the input size since $N \equiv \sum_p |S_p|^2$ is polynomial. Since each call to the separation oracle takes polynomial time by Lemma 4.2, Step 1 takes polynomial time. L being polynomial also ensures that (P^*) has polynomial size, and thus a basic feasible solution can be found in polynomial time in Step 3. $\square$

We note that the estimates on R and v (and thus L) can be improved, but we have satisfied our main goal here, which was proving that the running time of our algorithm is polynomial.

The reader may wonder how our algorithm would deal with Stein *et al.* [2010]'s counterexample, a symmetric game in which the only CE that is a convex combination of symmetric product distributions has irrational probabilities. Since we have proved that our algorithm computes a rational CE as a convex combination of product distributions, it must violate the symmetry property. Indeed as we discussed in Section 4.1, our Purified Separation Oracle can return asymmetric cuts for symmetric games and symmetric queries, and thus for this game it must return at least one asymmetric cut.

5 Uncoupled Dynamics with Polynomial Communication Complexity

Hart and Mansour [2010] considered the setting where each player initially knows only her own utility function, and analyzed the communication complexity for such *uncoupled* dynamics to reach various equilibrium concepts. They use a straightforward adaptation of Papadimitriou and Roughgarden's Ellipsoid Against Hope algorithm to show that a CE can be reached using polynomial communication. The recent discovery by Stein *et al.* [2010] of flaws of the Ellipsoid Against Hope algorithm imply that Hart and Mansour's procedure as proposed would not reach an exact CE. We show that our modified version of the Ellipsoid Against Hope algorithm can be straightforwardly adapted into a polynomial communication procedure for exact CE.

Formally, in Hart and Mansour's setting, each player p initially knows only her utility function u^p. No assumption is made on how the game is represented and the cost of computation is of no concern; instead, we focus on the amount of communication required to reach a CE. Hart and Mansour's approach used the following property of the Product Separation Oracle (Lemma 3.1): given $y \geq 0$, the corresponding product distribution x depends only on y and not on the utilities of the game. Although generating the cutting plane requires computing xU^T which does depend on the utilities, each entry (p, i, j) of the vector xU^T depends only on the utilities of player p.

We now describe Hart and Mansour's procedure. A *center* (a designated player that communicates with all other players) runs the Ellipsoid Against Hope algorithm; when the Product Separation Oracle generates a product distribution x, the center sends it to all players, and asks each player p to compute her segment of the vector xU^T, i.e., entries (p, i, j) for all $i, j \in S_p$, to send back to the center. This exactly simulates the Ellipsoid Against Hope algorithm, and its communication costs are those of sending the product distributions to players and each player sending back her part of xU^T.

This procedure can be modified to use the Purified Separation Oracle instead. At Step 2a of the Purified Separation Oracle (Algorithm 1), for each $s_p \in S_p$ the center sends $x_{(p \to s_p)}$ to all players and asks each to compute her segment of $x_{(p \to s_p)} U^T$. After assembling the vector $x_{(p \to s_p)} U^T$ from the segments, the center checks whether $\left[x_{(p \to s_p)} U^T\right] y \geq 0$. We call the resulting modified version of Algorithm 1 the Uncoupled Purified Separation Oracle. It is straightforward to see that this exactly simulates the Purified Separation Oracle. The communication costs are those of the center sending the product distributions and the players sending back segments of $x_{(p \to s_p)} U^T$. At most $\sum_p |S_p|$ rounds of such exchange are required for each call to the Purified Separation Oracle, therefore the total amount of communication is polynomially bounded.

Corollary 5.1. *Modify Hart and Mansour's procedure by replacing its separation oracle with the Uncoupled Purified Separation Oracle. The resulting communication procedure reaches an exact CE while both the number of bits of communication required and the size of the support are polynomial in* n, $\sum_p |S_p|$

and the bit complexity of the game utilities.

6 Computing Extensive-form Correlated Equilibria

Recently, von Stengel and Forges [2008] proposed *extensive-form correlated equilibrium* (EFCE), a solution concept for extensive-form games that is closely related to correlated equilibrium. Whereas in a CE of the induced normal form of a game the intermediary recommends a pure strategy (i.e., a move for each information set) to each player at the start of the game, in an EFCE the intermediary recommends a move to the player only when the corresponding information set is reached. Here we focus on the computational problem of finding an EFCE and refer interested readers to von Stengel and Forges [2008] for details on EFCE as a solution concept. Huang and von Stengel [2008] described a polynomial-time algorithm for computing sample extensive-form correlated equilibria. Their algorithm follows a very similar structure as Papadimitriou and Roughgarden's Ellipsoid Against Hope algorithm, and the problems pointed out by Stein *et al.* [2010] carry over. As a result, the algorithm can fail to find an exact EFCE.

We extend our fix for Papadimitriou and Roughgarden's Ellipsoid Against Hope algorithm to Huang and von Stengel's algorithm, allowing it to compute an exact EFCE with polynomial-sized support. We first give a high-level description of Huang and von Stengel's algorithm, following Huang [2011].[8] The input of the problem is an n-player extensive-form game with perfect recall. Each nonterminal node of the game tree is a decision node for either one of the players or Chance. H denotes the set of information sets, and C_h denotes the set of moves available from $h \in H$, and T denotes the set of terminal nodes. Due to the tree structure of the extensive form, for each node there exists a unique path from the root of the tree to that node. Let s be a pure-strategy profile; $s(h)$ denotes the move at information set $h \in H$. Let z be a distribution over the set of pure-strategy profiles. Generically the size of z is exponential. Huang and von Stengel [2008] showed that z is an EFCE if it satisfies a polynomial number of linear constraints, which can be written as $Az + Bv \geq 0$ where v is an auxiliary vector of polynomial size. They considered the exponential-sized primal LP

$$\begin{aligned} \max \sum_s z_s \qquad (8) \\ Az + Bv \geq 0 \\ z \geq 0, \end{aligned}$$

[8]We assume that readers are familiar with the standard concepts of extensive form games, information sets, perfect recall, and behavior strategies.

and its dual

$$
\begin{aligned}
A^T y &\leq -1 \\
B^T y &= 0 \\
y &\geq 0
\end{aligned}
\tag{9}
$$

which has a polynomial number of variables and exponential number of constraints. The following is a key lemma:

Lemma 6.1. *[Huang & von Stengel, 2008] For all $y \geq 0$ such that $B^T y = 0$, there exists a product distribution z such that $zA^T y = 0$. Given y, the corresponding z can be computed in polynomial time.*

Unlike the simultaneous-move game case, z being a product distribution (mixed-strategy profile) does not imply that it can be concisely represented, as the number of pure strategies for each player can be exponential. Fortunately the z constructed by Lemma 6.1 corresponds to a *behavior strategy profile*, which specifies a distribution (denoted z^h) over moves for each information set h. Formally, given z^h for all $h \in H$, the resulting distribution over pure-strategy profiles is given by

$$
\forall s, \qquad z_s = \sum_{t \in T : t \text{ agrees with } s} p(t) x_t,
$$

where we say t *agrees* with pure-strategy profile s if all the moves by the players on the path from the root to t are given by s, $p(t)$ is the product of probabilities of moves by Chance along the path from the root to t, and $x_t = \prod_{h \text{ precedes } t} z^h_{s(h)}$ is the product of probabilities of moves by the players along the path from the root to t. Here by "h precedes t" we mean that h is an information set on the path from the root to t. Note that perfect recall ensures that an information set h appears at most once along the path from the root to t. Such a behavior strategy profile requires only a polynomial number of values to specify.

By the same argument as for the Ellipsoid Against Hope algorithm, Lemma 6.1 implies the infeasibility of (9), and can be used as a separation oracle for applying the ellipsoid method on (9). In order to generate the cutting plane $[zA^T]y \leq -1$, the oracle needs to compute zA^T whose inner dimensions are exponential. It turns out that zA^T can be formulated as expected utility computations which can be carried out in polynomial time. Huang and von Stengel's algorithm thus proceeds similarly as in the Ellipsoid Against Hope algorithm to produce a feasible solution to (8), which can be scaled to be an EFCE.

By the same argument as our fix of the Ellipsoid Against Hope algorithm, in order to overcome the problems pointed out by Stein *et al.* [2010] it is sufficient to construct a Purified Separation Oracle that given a $y \geq 0$ such that $B^T y = 0$, computes a pure-strategy profile s such that $(A_s)^T y \geq 0$. We construct such an oracle using a similar application of the method of conditional probabilities. For a behavior strategy profile z, an information set h, and a move $d \in C_h$, define $z_{(h \to d)}$ to be the behavior strategy profile that is identical to z except at information set h, where the corresponding player deterministically chooses d instead. Our Purified Separation Oracle starts with the behavior strategy profile constructed by Lemma 6.1, and uses the same algorithm as Algorithm 1, except that instead of going through players in step 2a, we go through information sets sequentially, and for each information set h we iterate through $z_{(h\to d)}$ until we find a d^* such that $[z_{(h\to d^*)}A^T]y \geq 0$.

To show that our algorithm is correct, we use the following lemma:

Lemma 6.2. *Given a behavior strategy profile z, for each information set h,*

$$z = \sum_{d\in C_h} z_{(h\to d)} z_d^h,$$

where z_d^h is the probability of choosing d at h prescribed by z.

Proof. Recall that

$$z_s = \sum_{t\in T: t \text{ agrees with } s} p(t)x_t,$$

where $x_t = \prod_{h \text{ precedes } t} z_{s(h)}^h$. Since the moves along the path to t are uniquely determined by t, x_t is fully specified by the behavior strategies and does not depend on s. We can write this in matrix form as $z = Fx$, with $x \in \mathbb{R}^{|T|}$. Let $x_{(h\to d)} \in \mathbb{R}^{|T|}$ be the vector induced by behavior strategy profile $z_{(h\to d)}$. We then have $z_{(h\to d)} = Fx_{(h\to d)}$. Furthermore, we observe that for all h,

$$x = \sum_{d\in C_h} x_{(h\to d)} z_d^h.$$

(It is straightforward to verify the above by considering the terminal nodes t for which h precedes t and then the other terminal nodes.) We thus have

$$z = Fx = F\sum_{d\in C_h} x_{(h\to d)} z_d^h = \sum_{d\in C_h} z_{(h\to d)} z_d^h,$$

which is the required equality. $\square$

The correctness and the polynomial running time of our algorithm for Purified Separation Oracle then follow by the same argument as in the proof of Lemma 4.2. After modifying Huang and von Stengel's algorithm by replacing their separation oracle with our Purified Separation Oracle, the resulting algorithm computes in polynomial time an exact EFCE that is a mixture of a polynomial number of pure-strategy profiles.

Corollary 6.3. *Given a game in extensive form, an exact EFCE with polynomial-sized support can be computed in polynomial time.*

7 Conclusion

We have proposed a polynomial-time algorithm, a variant of Papadimitriou and Roughgarden's Ellipsoid Against Hope approach, for computing an exact CE given a game representation with polynomial type and satisfying the polynomial expectation property. A key component of our approach is a derandomization of Papadimitriou and Roughgarden's separation oracle using the method of conditional probabilities, yielding a polynomial-time separation oracle that outputs cuts corresponding to pure-strategy profiles. Our approach is then spared from dealing with the numerical precision issues that were a major focus of previous approaches, and the algorithm is considerably simplified as a result. Furthermore, the correlated equilibria returned by our algorithm have polynomial-sized supports. We expect these properties of our algorithm to be independently interesting, beyond its usefulness in resolving the recent uncertainty about the computational complexity of identifying exact CE. For example, we show that our techniques can be adapted to two existing algorithms that are based on the Ellipsoid Against Hope approach, Hart and Mansour's [2010] CE procedure with polynomial communication complexity and Huang and von Stengel's [2008] polynomial-time algorithm for extensive-form correlated equilibria, yielding in both cases exact solutions with polynomial-sized supports.

Our algorithm has additional practical benefits: the resulting cutting planes are deeper cuts than those produced by the original oracle, resulting in a smaller number of iterations required to reach convergence, albeit at the cost of more work per iteration. It is also possible to return cuts corresponding to pure strategy profiles with (e.g.) good social welfare, yielding a heuristic method for generating correlated equilibria with good social welfare; we do note, however, that finding a CE with optimal social welfare is generally NP-hard for many game representations [Papadimitriou & Roughgarden, 2008].

References

Aumann, R. (1974). Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, *1*(1), 67–96.

Aumann, R. (1987). Correlated equilibrium as an expression of Bayesian rationality. *Econometrica: Journal of the Econometric Society*, 1–18.

Bhat, N., & Leyton-Brown, K. (2004). Computing Nash equilibria of action-graph games. *UAI: Proceedings of the Conference on Uncertainty in Artificial Intelligence* (pp. 35–42).

Chen, X., Deng, X., & Teng, S.-H. (2009). Settling the complexity of computing two-player Nash equilibria. *Journal of the ACM*, *56*(3).

Dantzig, G., & Thapa, M. (2003). *Linear programming 2: Theory and extensions*. Springer.

Daskalakis, C., Goldberg, P. W., & Papadimitriou, C. H. (2009). The complexity of computing a Nash equilibrium. *SIAM Journal on Computing*, *39*(1), 195–259.

Erdős, P., & Selfridge, J. L. (1973). On a combinatorial game. *Journal of Combinatorial Theory, Series A*, *14*(3), 298 – 301.

Germano, F., & Lugosi, G. (2007). Existence of sparsely supported correlated equilibria. *Economic Theory*, *32*(3), 575–578.

Goldberg, P. W., & Papadimitriou, C. H. (2006). Reducibility among equilibrium problems. *STOC: Proceedings of the Annual ACM Symposium on Theory of Computing* (pp. 61–70).

Govindan, S., & Wilson, R. (2003). A global Newton method to compute Nash equilibria. *Journal of Economic Theory*, *110*, 65–86.

Grötschel, M., Lovász, L., & Schrijver, A. (1988). *Geometric algorithms and combinatorial optimization*. Springer-Verlag, New York, NY.

Hart, S., & Mansour, Y. (2010). How long to equilibrium? the communication complexity of uncoupled equilibrium procedures. *Games and Economic Behavior*, *69*(1), 107–126.

Hart, S., & Schmeidler, D. (1989). Existence of correlated equilibria. *Mathematics of Operations Research*, *14*(1), 18–25.

Huang, W. (2011). *Equilibrium computation for extensive games*. Doctoral dissertation, London School of Economics and Political Science.

Huang, W., & von Stengel, B. (2008). Computing an extensive-form correlated equilibrium in polynomial time. *WINE: Proceedings of the Workshop on Internet and Network Economics* (pp. 506–513).

Jiang, A. X., Leyton-Brown, K., & Bhat, N. (2011). Action-graph games. *Games and Economic Behavior*, *71*(1), 141–173.

Kearns, M., Littman, M., & Singh, S. (2001). Graphical models for game theory. *UAI: Proceedings of the Conference on Uncertainty in Artificial Intelligence* (pp. 253–260).

Lemke, C., & Howson, J. (1964). Equilibrium points of bimatrix games. *Society for Industrial and Applied Mathematics Journal of Applied Mathematics*, *12*, 413–423.

McKelvey, R., & McLennan, A. (1996). Computation of equilibria in finite games. *Handbook of Computational Economics*, *1*, 87–142.

Myerson, R. (1997). Dual reduction and elementary games. *Games and Economic Behavior*, *21*(1-2), 183–202.

Nau, R., & McCardle, K. (1990). Coherent behavior in noncooperative games. *Journal of Economic Theory*, *50*(2), 424–444.

Papadimitriou, C. (2005). Computing correlated equilibria in multiplayer games. *STOC: Proceedings of the Annual ACM Symposium on Theory of Computing* (pp. 49–56).

Papadimitriou, C., & Roughgarden, T. (2008). Computing correlated equilibria in multi-player games. *Journal of the ACM*, *55*(3), 14.

Papadimitriou, C., & Roughgarden, T. (2010). Comment on "computing correlated equilibria in multi-player games". `http://theory.stanford.edu/~tim/papers/comment.pdf`, accessed Jan. 10, 2011.

Raghavan, P. (1988). Probabilistic construction of deterministic algorithms: Approximating packing integer programs. *Journal of Computer and System Sciences*, *37*(2), 130 – 143.

Scarf, H. (1967). The approximation of fixed points of a continuous mapping. *SIAM Journal of Applied Mathematics*, *15*, 1328–1343.

Spencer, J. (1994). *Ten lectures on the probabilistic method*. CBMS-NSF regional conference series in applied mathematics. Society for Industrial and Applied Mathematics.

Stein, N. D., Parrilo, P. A., & Ozdaglar, A. (2010). Exchangeable equilibria contradict exactness of the Papadimitriou-Roughgarden algorithm. `http://arxiv.org/abs/1010.2871v1`.

van der Laan, G., Talman, A., & van der Heyden, L. (1987). Simplicial variable dimension algorithms for solving the nonlinear complementarity problem on a product of unit simplices using a general labelling. *Mathematics of Operations Research*, *12*(3), 377–397.

von Neumann, J., & Morgenstern, O. (1944). *Theory of games and economic behavior*. Princeton University Press.

von Stengel, B., & Forges, F. (2008). Extensive-form correlated equilibrium: Definition and computational complexity. *Mathematics of Operations Research*, *33*(4), 1002–1022.