Computing Correlated Equilibria in Multi-Player Games*

Christos H. Papadimitriou† Tim Roughgarden‡

January 5, 2008

Abstract

We develop polynomial-time algorithms for finding correlated equilibria—a well-studied notion of rationality that generalizes the Nash equilibrium—in a broad class of succinctly representable multiplayer games, encompassing graphical games, anonymous games, polymatrix games, congestion games, scheduling games, local effect games, as well as several generalizations. Our algorithm is based on a variant of the existence proof due to Hart and Schmeidler [23], and employs linear programming duality, the ellipsoid algorithm, Markov chain steady state computations, as well as application-specific methods for computing multivariate expectations over product distributions.

For anonymous games and graphical games of bounded tree-width, we provide a different polynomial-time algorithm for optimizing an arbitrary linear function over the set of correlated equilibria of the game. In contrast to our sweeping positive results for computing an arbitrary correlated equilibrium, we prove that optimizing over correlated equilibria is NP-hard in all of the other classes of games that we consider.

1 Introduction

The Nash equilibrium [35] is the standard notion of rationality in game theory. It is standard in at least three dictionary senses: everybody uses it; its many refinements and generalizations are always measured against it; and finally, it has served as the flagship open computational problem in the emerging area at the interface between game theory and algorithms. In this latter regard, it was very recently established that computing a mixed Nash equilibrium is PPAD-complete, even in a two-person game [9, 12, 37].

*Preliminary versions of these results appeared in the Proceedings of the 16th Annual Symposium on Discrete Algorithms, January 2005, and the Proceedings of the 37th Annual Symposium on Theory of Computing, May 2005.

†UC Berkeley, Computer Science Division, Soda Hall, Berkeley, CA 94720. Supported by an NSF grant CCR-0121555 and a Microsoft Research grant. Email: `christos@cs.berkeley.edu`.

‡Department of Computer Science, Stanford University, 462 Gates Building, 353 Serra Mall, Stanford, CA 94305. Supported in part by an NSF CAREER Award, an Alfred P. Sloan Fellowship, and an ONR Young Investigator Award. Email: `tim@cs.stanford.edu`. Most of this research was done while the author was at UC Berkeley and supported by an NSF Postdoctoral Fellowship.

As mentioned above, there are several other competing notions of rationality; chief among them is the *correlated equilibrium*, proposed by Aumann [3]. While the mixed Nash equilibrium is a distribution on the strategy space that is "uncorrelated" (that is, the product of independent distributions, one of each player), a correlated equilibrium is a general distribution over strategy profiles. It must however possess an equilibrium (quiescence) property: If a strategy profile is drawn from this distribution (presumably by a trusted external agent), and each player is told separately his/her own component, then no player has an incentive to choose a different strategy, because, assuming that all other players also obey, the suggested strategy is the best in expectation. (See Section 2 for a precise definition and examples.) The correlated equilibrium has several important advantages: It is a perfectly reasonable, simple, and plausible concept; it is guaranteed to always exist (simply because the Nash equilibrium is an example of a correlated equilibrium); it arises from simple and natural dynamics in senses that the Nash equilibrium does not (see [8, 21, 22]); and it can be found in polynomial time for any number of players and strategies by linear programming, since the inequalities specifying the quiescence property above are linear. In fact, the correlated equilibrium that optimizes any linear function (such as the sum) of the players' utilities can be computed this way. Such optimization is obviously useful in games where player payoffs can be meaningfully compared, and also ensures Pareto optimality (within the set of correlated equilibria) in games where they cannot.

A parenthesis: But why should one be interested in *algorithms* for computing equilibria (in the absence of a professional bias, that is)? Equilibria are concepts of rationality, models of player behavior, coveted existence theorems that demonstrate the modeling power of games — albeit in a context that is not explicitly computational. Indeed, several game theorists cannot understand our community's obsession with this aspect. *We believe that the complexity of equilibria is of fundamental importance in game theory,* and not just a computer scientist's afterthought. Intractability of an equilibrium concept would make it implausible as a model of behavior. In the words of Kamal Jain: "If your PC cannot find it, then neither can the market." In more detail, most interpretations of an equilibrium concept involve someone (either the participants or a third party), at some point, determining an equilibrium, and there is no clear reason why a group of agents cannot be efficiently simulated by a machine.

But there *is* a disconnect between the focus of current algorithmic research in game theory and the motivating consideration explained above: We study games because we believe they are productive models of markets, auctions, networks. And still, most algorithmic research in game theory has been aimed at the 2-player case. Clearly, we need to expand our scope to include *multiplayer games.* That is the focus of this paper.

But applying our field's methodology and norms to multiplayer games is tricky. Many computational problems related to multiplayer games (computing correlated equilibria, for instance) are "easy" in our way of thinking, but for the wrong reason: *The input length is exponential* — one number for each combination of strategies. Clearly, this is not an acceptable situation. If a computational problem is important, the description of its typical instance cannot require an astronomical number of bits.

In conclusion: Equilibrium problems in games, of which the correlated equilibrium is a quite prominent example, are worthy objects of study from the algorithmic point of view. Multiplayer games are the most compelling specimens in this regard. But, to be of interest, they must be somehow represented succinctly. *This paper is about, and largely settles, the question of the existence of polynomial-time algorithms for computing correlated equilibria in succinctly representable multiplayer games.*

Succinct Games

In the recent literature we have seen the introduction of many examples of important classes of multiplayer games with succinct representation (see below). But the most ancient such class is that of *symmetric games*, studied by von Neumann and Nash themselves. These are games in which players are identical and indistinguishable, in that a player's utility depends on the player's choice (but not identity) and on the number of other players who have chosen each of the strategies. Thus, a symmetric game with n players and $m << n$ strategies needs about n^m, as opposed to m^n, numbers for its description (the number of partitions, with repetitions, of n strategy choices to m bins). We show that correlated equilibria of a symmetric game can be optimized over—and in particular, one can be computed—in time polynomial in this succinct representation (for all m and n). Our results also hold for *anonymous games* (e.g. [5]), a generalization of symmetric games with player-specific utility functions (each a symmetric function of the actions chosen by the other players). Finally, we prove that even mixed Nash equilibria can be efficiently computed in symmetric games when the number of players is large relative to the number of strategies; this stands in contrast to the PPAD-completeness of the problem when there is a constant number of players [7, 9, 12, 19]. (See also [6] for a recent work on the complexity of *pure* Nash equilibria in symmetric games.)

Another important class of succinct games is that of *graphical games* [25, 26], where we are given a network connecting the players, and each player's utility depends only on the choices of the players in its neighborhood. Here, we show that optimizing over correlated equilibria is NP-hard, even when the underlying graph is bipartite and has bounded degree. Since correlated equilibria are linear programs, objects in which existence and linear optimization are typically computationally equivalent, it is natural to be pessimistic about efficiently computing correlated equilibria in general graphical games. Nevertheless, we provide a polynomial-time algorithm for computing a correlated equilibrium in *every* graphical game. We also show that optimization is tractable provided the graph has bounded tree-width, generalizing a result of Kakade et al. [25].

In a *polymatrix game* [44, 24, 14] each player plays a 2-person game with each other player, and his/her choices are the same in each of these games; the utilities are then added. B. von Stengel (personal communication, July 2004) observed that it is NP-hard to compute the correlated equilibrium that optimizes the sum of all utilities in such games. Once more, a polynomial-time algorithm for computing a correlated equilibrium in such a game follows from our results.

In fact, we can solve a class of games that generalizes both graphical games and polymatrix games: In a *hypergraphical game* several subsets of the players are involved in separate games with utilities added. If the hypergraph is a clique graph, we have a polymatrix game; if every vertex has nonzero utility in only one hyperedge, and these nonzero hyperedges have a certain symmetry property, we have a graphical game. We can compute correlated equilibria for this general class of games in polynomial time.

In a *congestion game* [40] we have a set of resources, and the strategies of each player are subsets of these resources. Each resource has a delay that is a function of the number of players using it, and the (negative) utility of a player is the sum of the delays of the resources in the set that s/he chose. Such games are guaranteed to have pure Nash equilibria, but finding them is in general PLS-complete [15]. The nonatomic version of such games were studied extensively by Roughgarden and Tardos [41]. Again, we prove that finding an optimal correlated equilibrium is NP-hard, but give a polynomial-time algorithm for finding *some* correlated equilibrium. In fact, the algorithm works for the more general case of *local effect games* [31], as well as a version of congestion games, with different equilibrium properties, known as *scheduling games* [18].

Finally, there is a rather obvious, yet not treated in the literature, succinct representation of games which could be called *sparse games*, in which some utility values are given explicitly, and the rest are assumed to be zero (in analogy with sparse matrices). All our positive results are easily seen to hold for this representation, which will not be mentioned any further.

Note that, in all these applications of our techniques, ours is the first polynomial-time algorithm for computing *any* kind of equilibrium (without making severe additional restrictions). And we know of no other natural classes of succinct games for which this problem is still open — with the exception of certain "succinct games of superpolynomial type" (see Section 2) in which the number of strategies itself is exponential in the description, and for which radically different methods seem to be required, as well as a generic class of games defined by circuits [42].

There is a related research tradition in Game Theory concerned with learning equilibria via repeated play, during which players modify their strategy in response to the observed behavior of other players. For example, in *fictitious play*, each player perceives the other players' history of play as a histogram of some fixed mixed strategy, and plays best response to that. The question is, does such a process converge to any kind of equilibrium? Fictitious play converges to the Nash equilibrium in the case of (two-player) zero-sum games, but not in the general case. However, it was shown recently by Hart and Mas-Colell [22], by an extension of the proof of Blackwell's Approachability Theorem [4], that another natural variant, in which a player switches from the current strategy to another with probability proportional to the player's *regret* for having played the present strategy instead of the other in the past, converges to the set of correlated equilibria; a similar result for a different type of learning dynamics had been known [17]. Such learning processes apply very generally, including to all of the succinct games studied here, and can be interpreted as *approximating* equilibria within some ϵ — in this sense, they have the same goal as the present paper. However, seen this way, the learning methods require an exponential number of iterations to converge, proportional to $(\frac{1}{\epsilon})^k$ for a constant $k > 1$. Our ellipsoid-based algorithm, on the other hand, is polynomial in $\log \frac{1}{\epsilon}$. In addition, there is no known way to optimize over the set of correlated equilibria via the learning approach in [17, 22].

The Main Ideas

The ingredients of our basic algorithm are: Linear programming duality; a novel use of the ellipsoid algorithm; the steady state distribution of a Markov process; and calculating multivariate expectations. Our work builds on an ingenious observation on correlated equilibria made some time ago by Hart and Schmeidler [23], motivated by the following natural question: why is the feasibility of the linear program that describes the set of correlated equilibria (see (CE) below) typically proved so indirectly, via Nash's and Brouwer's theorems? There surely must be a more direct proof. They give such a proof by a sort of "doubly nested" application of game-theoretic duality. We present a version of their proof that is based on linear programming duality and is computationally explicit. It turns out that the dual object of interest is the steady-state distribution of a certain Markov chain (a concept also mentioned in [36], an independent proof of the same existence result, and is further explored in [34]).

A correlated equilibrium of a multiplayer game is an exponential object; a polynomial algorithm must find a *succinct representation* of such. Our algorithm employs *polynomial mixtures of products*, or *pmp's*: A distribution over the strategy space is given as the convex combination of polynomially many product (independent, Nash equilibrium-like) distributions; in other words, an n-dimensional tensor with polynomially small (positive) rank. Such a correlated equilibrium has an attractive implementation (sampling algorithm): First sample the products, and then sample the strategies of each player according to the resulting product distribution. *The correlated equilibria computed by our algorithm are pmp's.*

Briefly, our algorithm works on the dual (D) of (CE), which has polynomially many variables and exponentially many constraints. The existence proof implies (D) is infeasible; despite this fact, we run the ellipsoid algorithm on it — a maneuver dubbed "ellipsoid algorithm against hope." We use Markov chain computations at each step of the ellipsoid algorithm to find a violated convex combination of the constraints of (D). At the conclusion of the algorithm, we have a polynomial number of such combinations (the cuts of the ellipsoid algorithm) that are themselves infeasible, call those (D′). Solving the dual of this new linear program, call it (P′), gives us the required correlated equilibrium as a pmp.

Even though (P′) has polynomial dimensions, its constraint matrix must be computed as the product of two exponentially large matrices (the constraints of (P) times the matrix whose columns are the computed pmp's). The computation implicit in the matrix multiplication formula for each element of the constraint matrix of (P′) suggests an expectation calculation; for each class of succinct games we need a problem-dependent trick to carry this out. Almost magically, in all cases outlined above, this turns out to always be doable in polynomial time, by utilizing one or two of three techniques: Linearity of expectation (polymatrix games, hypergraphical games), dynamic programming (congestion, scheduling, and local effect games), or implementing the definition of expectation on polynomially small domains (graphical and hypergraphical games).

To *optimize* over the correlated equilibria of a succinctly represented game, we require further ideas. We give a general optimization framework based on dimensionality reduction of the above linear programming formulations. In more detail, for each player, the strategy profiles of all opponents are divided into equivalence classes, in which the player's utility depends only on his/her own choice. This reduces the number of variables of the linear program to a polynomial, but introduces new "consistency" issues between the equivalence classes of different players. We prove that this reduced linear program can be optimized over efficiently provided a suitable "consistency problem" can be implemented in polynomial time, and we show how to efficiently solve the consistency problem for anonymous games and graphical games of bounded tree-width. The correlated equilibria computed by our optimization algorithm are a degenerate type of pmp: distributions with polynomial support. We complement this positive result by proving that, for all other classes of games that we study, optimizing over the set of correlated equilibria is NP-hard.

Finally, for anonymous games and tree graphical games, we refine the preceding positive results and identify explicit polynomial-size linear programs that characterize the set of correlated equilibria. As a consequence, interior-point methods can be used in lieu of the ellipsoid method for computing optimal correlated equilibria in such games.

2 Definitions

In a *(finite) game* we have n *players* $1, \dots, n$. Each player $p \leq n$ has a finite set of *strategies* or *choices*, S_p, with $|S_p| \geq 2$. The set $\mathcal{S} = \prod_{p=1}^{n} S_p$ is called the *set of strategy profiles.* We shall denote $\prod_{q \neq p} S_q$ by S_{-p}. We use m for the maximum of all the $|S_p|$'s. The *utility* or *payoff* function of player p is a function u^p mapping $\mathcal{S}$ to the integers.

Example 2.1 The *chicken game* has 2 players (think of them as very competitive drivers speeding from different streets to an intersection), each with two strategies: $S_1 = S_2 = \{\text{stop}, \text{go}\}$. The utilities, tabulated below, reflect the situation (in the format $u^1(s), u^2(s)$, where the strategies of player 1 are the rows and of 2 the columns):

	stop	go
stop	$4,4$	$1,5$
go	$5,1$	$0,0$

A *distribution on* $\mathcal{S}$ is a vector of nonnegative real numbers, one for each strategy profile in $\mathcal{S}$, adding up to 1. A distribution x on $\mathcal{S}$ is a *product* (or *of rank one*) if for each player p there is a distribution x^p on S_p such that for all $s = (s_1, \dots, s_n) \in \mathcal{S}$, $x_s = \prod_{p=1}^{n} x^p_{s_p}$.

A *correlated equilibrium* is a distribution x on $\mathcal{S}$ such that for all players p and all strategies $i, j \in S_p$ the following is true: Conditioned on the p-th component of a strategy profile drawn from x being i, the expected utility for p of playing i is no smaller than that of playing j:

$$\sum_{s \in S_{-p}} [u^p_{is} - u^p_{js}] x_{is} \geq 0, \qquad (CE),$$

where we denote by is the strategy profile resulting from $s \in S_{-p}$ by adding the component $i \in S_p$. Intuitively, if a trusted intermediary were to draw a strategy profile s from this distribution and announce to each player p separately (and privately) p's own component i, p will have no incentive to choose another strategy j—assuming that the other players also conform to the intermediary's suggestion, i is the optimum response in expectation. Finally, a *(mixed) Nash equilibrium* is a correlated equilibrium that happens to be a product distribution.

Example 2.2 The following five distributions are correlated equilibria in the chicken game.

0	1
0	0

0	0
1	0

1/4	1/4
1/4	1/4

0	1/2
1/2	0

1/3	1/3
1/3	0

The first two are pure Nash equilibria, and the third is the only other Nash equilibrium, corresponding to both players playing the mixed strategy $\{1/2, 1/2\}$. The fourth correlated equilibrium can be interpreted as a *traffic light:* a trusted intermediary tosses a fair coin and, depending on the outcome, suggests a strategy to each player; neither player has an incentive to disobey. Notice that, in terms of utility expectations, it does better than the Nash equilibrium. The last correlated equilibrium (it takes a second to verify that it is one) is the one that maximizes the expected sum of utilities, obtained by a linear maximization over (CE).

A *succinct game* $G = (I, T, U)$ is defined, like all computational problems, in terms of a set of efficiently recognizable inputs I, and two polynomial algorithms T and U. For each $z \in I$, $T(z)$ returns a *type*, that is, an integer $n \geq 2$ (the number of players) and an n-tuple of integers $(t_1, \dots, t_n)$, each at least 2 (the cardinalities of the strategy sets). If n and the t_p's are polynomially bounded in $|z|$, the game is said to be *of polynomial type.* Given any n-tuple of positive integers $s = (s_1, \dots, s_n)$, with $s_p \leq t_p$ for all $p \leq n$, $U(z, p, s)$ returns an integer standing for the utility $u^p(s)$. The resulting game is denoted $G(z)$.

Examples of polynomial succinct games were discussed in the introduction and will be defined more formally in Section 4.

Finally, given a succinct game G, a *polynomial correlated equilibrium scheme* consists of two polynomial-time algorithms A and R. A on input $z \in I$ produces (deterministically or not) an output y. R is a randomized algorithm which, on input z and y, will select an element s of the strategy space of $G(z)$ with probability corresponding to a correlated equilibrium of $G(z)$. We give numerous examples in Section 4.

3 The Basic Algorithm

The Existence Proof

The following result is usually proved via Nash's Theorem. Our proof is a more constructive variant of that in [23]:

Theorem 3.1 *Every game has a correlated equilibrium.*

Proof: Consider the linear program

$$\begin{aligned} \max \sum_s x_s & \\ Ux \geq 0 & \qquad\qquad (P) \\ x \geq 0 & \end{aligned}$$

where the objective is the sum of all x's and by $Ux \geq 0$ we mean the constraints of (CE), one for every player and pair of strategies. The program (P) is either trivial (with maximum 0) or unbounded, the latter case occurring precisely when the game has a correlated equilibrium. Therefore, by duality, to prove the theorem, it suffices to show that the dual of (P)

$$\begin{aligned} U^T y \leq & \ -1 \qquad\qquad (D) \\ y \geq & \ \ \ 0 \end{aligned}$$

is always infeasible. We now need the following lemma.

Lemma 3.2 *For every $y \geq 0$ there is a product distribution x such that $xU^Ty = 0$.*

To see that the lemma establishes the theorem, notice that xU^Ty is a convex combination of left-hand sides of constraints of (D), and hence for every feasible y it should evaluate to something negative.

We now turn to the proof of the lemma. We look at the expression for xU^Ty setting $x_s = x^1_{s_1} \cdot \ldots \cdot x^n_{s_n}$ (since the lemma promises a solution in product form). This product is linear and homogeneous in the utilities (since the entries of the matrix U are linear in the utilities); consider therefore the coefficient in this expression of the typical utility u^p_{is}, where $s \in S_{-p}$ and $i \in S_p$. This coefficient is

$$\left[\prod_{q \neq p} x^q_{s_q}\right] \cdot \left[x^p_i \sum_{j \in S_p} y^p_{ji} - \sum_{j \in S_p} x^p_j y^p_{ij}\right].$$

(Recall that the variables y^p_{ij} of (D) are one for each player p and pair of strategies (i, j) of this player.) Now the second term of this expression is recognized as the equilibrium equation of the steady-state distribution x_i^p of a Markov chain y_{ij}^p (normalizing the y's appropriately so they add to one or less per row). Hence, by setting, for each player p, the x_i^p's to be the steady state distribution of the Markov chain defined by the normalized y_{ij}^p's (one of the possible ones if the chain happens to be periodic, or to any distribution if they are all zero), all second factors become zero, and so does the expression xU^Ty. This concludes the proof of the lemma and the theorem. ■

It is worth reflecting a little on this. That the certificate of existence of a correlated equilibrium is in product form might appear to bring us a little closer to computing a Nash equilibrium, but, apparently, not close enough [9, 12]. This proof leads us to an interesting interpretation of the dual [34]: The dual variable y_{ij}^p may be thought of as the tendency player p might have to switch from strategy i to strategy j. The steady state of this process suggests then a distribution that exposes dual infeasibility. A final observation is that this intriguing mapping from y values to x values is universal for all games of the same type (strategy set cardinalities), since it does not depend on the u_s^p's.

Ellipsoid Against Hope

The challenge now is to turn this existence proof into a polynomial algorithm. The idea is to apply the ellipsoid algorithm [27, 38, 20] to the dual (D), *which is guaranteed to be infeasible.* Notice that (D) has polynomially many variables (their number, denoted by N, is about nm^2) and exponentially many constraints ($M \approx m^n$), while for (P) the opposite holds, and hence the ellipsoid algorithm is appropriate for (D). At each step i we have a candidate solution y_i; we use the lemma to obtain x_i and the violated inequality $x_iU^Ty \leq -1$. We then proceed to the next step. Naturally, since we know that (D) is infeasible, the algorithm will end up reporting "infeasible" after the ellipsoid has shrunk too much to contain anything of interest. But by then we will have a polynomial collection of product distributions — the ingredients of the sought correlated equilibrium.

In particular, after the termination of the ellipsoid algorithm at the Lth step, we have L product distributions $x_1, \ldots, x_L$ such that, for each $i \leq L$, $[x_iU^T]y \leq -1$ is violated by y_i. This (modulo a complication related to arithmetic precision discussed in the next subsection) means that $[XU^T]y \leq -1$, where X is the matrix whose rows are the x_i's, is itself an infeasible linear program — one of polynomially many constraints.

But this implies in turn that the dual program $[UX^T]\alpha \geq 0, \alpha \geq 0$ is unbounded. *Such a nonzero α vector provides the desired correlated equilibrium, a convex combination of the x_i's that satisfies (P).*

The Issue of Arithmetic Precision

As any student of the ellipsoid algorithm knows, there are nontrivial, if usually benign, issues of arithmetic precision involved in it. This application of the method, however, presents us with two unusually sticky complications.

The first complication is that the eigenvector computations required to compute x_i from the y_i, via matrix inversion, introduce an increase at each step in the number of bits of the rational numbers involved by a factor of m —the dimension of the inverted matrices. This, of course, becomes problematic after polynomially many steps.

The second problem is this: The ellipsoid method starts with a ball of radius u^N, where u is the largest in absolute value coefficient of the constraints (assumed integer), and ends when the ellipsoid has volume smaller than $u^{-\Theta(N)}$, after $L = O(N^2 \log u)$ steps. Our assertion that $[XU^T]y \leq -1$ is infeasible rests on the presumption that $y_1, \ldots, y_L$ is a legitimate run of the ellipsoid algorithm also on the new program. However, the largest integer coefficient of the new constraint matrix is much larger than before, and thus the ellipsoid algorithm would have run much longer on it.

The following manoeuvre takes care of both complications: At each step we round the components of the product distribution x_i to the nearest multiple of $\epsilon = 1/(4nMNu^{N+1})$, to obtain $\tilde{x}_i$, and we consider the inequality $\tilde{x}_i U^T y \leq -\frac{1}{2}$. We *claim* that it is a valid inequality of (D) that is violated at y_i. In proof, suppose that there is a y within the current ellipsoid violating $\tilde{x}_i U^T y \leq -\frac{1}{2}$ yet satisfying $xU^T y \leq -1$. Then $(\tilde{x}_i - x_i)U^T y > \frac{1}{2}$ implying $2n\epsilon MNu||y|| > \frac{1}{2}$ (since the rounding error in the components of x is about n times that of the constituent distributions of the product), or, plugging in the value of ϵ, $||y|| > u^N$, which contradicts our assumption that the violating y is within the current ellipsoid. By the opposite sequence of inequalities, y_i must violate $\tilde{x}_i U^T y \leq -\frac{1}{2}$. We conclude that $\tilde{x}_i U^T y \leq -\frac{1}{2}$ is a proper inequality for use at the ith step of the ellipsoid algorithm.

Which brings us to the appropriate number of steps for the algorithm. We take it to be $L' = O(N^2 \log \frac{1}{\epsilon})$, enough so that it is also a valid limit for the modified dual $[\tilde{x}U^T]y \leq -\frac{1}{2}$.

Polynomial Combinations of Products

We can now state our main result of this section, a strengthening of the Existence Theorem (Theorem 3.1) and its constructive proof that is pregnant with the algorithmic results of the next section:

Theorem 3.3 *Every game has a correlated equilibrium that is a convex combination of product distributions, whose number is polynomial in n and m.*

Notice again that this theorem follows trivially from Nash's; the advantage is that, as we shall see in the next section, its proof is polynomially constructive. Also note that the strengthening is trivial without the polynomial bound (any correlated equilibrium is the convex combination of products of pure strategies, one for each strategy profile in its support).

Proof: The ingredients of the proof are in the discussion above. Running the ellipsoid algorithm on (D) with separating hyperplanes derived from the ellipsoid centers via steady-state calculation and rounding, and extending the algorithm to L' steps, yields an $L' \times N$ infeasible system $[\tilde{X}U^T]y \leq -\frac{1}{2}$, $y \geq 0$, call it (D′). The dual of (D′) is an $N \times L'$ system $[U\tilde{X}^T]\alpha \geq 0$, $\alpha \geq 0$, call it (P′), which must have a nonzero solution α. This solution, scaled to be a distribution, provides the L' multipliers such that the product $\tilde{X}^T\alpha$ is clearly a solution of (P), and thus a correlated equilibrium of the required form. That the number of products does not depend on $\log u$ is a consequence of the fundamental theorem of linear programming: (P$'$) has nm^2 constraints. ■

Notice that all steps of this proof are polynomially constructive except for the penultimate one, the construction of the matrix $[U\tilde{X}^T]$ needed for the solution of (P$'$) (as well as the calculation of the normal vector to the cut at each step of the ellipsoid algorithm, which, however, is easily seen to be a column of this product). The problem is that it is a $N \times M$ times $M \times L'$ matrix product; and, even though N and L' are polynomially large, $M \approx m^n$ is not. In the next section we show how to overcome this obstacle for a certain kind of succinct game that encompasses essentially all known classes.

4 Computing A Correlated Equilibrium

The Main Result

Consider a succinct game G. We say that G has the *polynomial expectation property* if there is a polynomial algorithm $\mathcal{E}$ which, given $z \in I$, $p \leq n$ and a *product* distribution $\{x_s : s \in \mathcal{S}\}$ with rational coefficients over the set $\mathcal{S}$ of strategy profiles of $G(z)$, returns in polynomial time the expectation of the utility u^p_s under the given product distribution:

$$\mathcal{E}(z, p, x_s) = \mathbf{E}_{x_s}\left[u^p_s : s \in \mathcal{S}\right].$$

Theorem 4.1 *If G is a succinct game of polynomial type and has the polynomial expectation property, then it has a polynomial correlated equilibrium scheme.*

Proof: A polynomial algorithm calculating a polynomial mixture of products is definitely a polynomial correlated equilibrium scheme: The sampling algorithm R first samples from the products and then for each player separately.

Hence we shall try to adapt the algorithm of the previous subsection to this situation. Since all other steps are trivially polynomial (by polynomial type), it suffices to show that the polynomial expectation property enables a polynomial solution of the penultimate step in the previous proof: Computing the product $U\tilde{X}^T$. In fact, it suffices to show how to compute each of the polynomially many entries, so let us concentrate on the entry corresponding to the p, i, j row of U and the ℓ row of $\tilde{X}$, the product distribution $\tilde{x}^\ell$ (for notational clarity we switch to superscripts for the rows of $\tilde{X}$). By inspection the term is

$$\sum_{s \in S_{-p}} [u^p_{is} - u^p_{js}]\tilde{x}^\ell_{is},$$

which is easily seen to be $\tilde{x}^{\ell,p}_i \cdot [\mathcal{E}(z; \tilde{x}^\ell | p \leftarrow i) - \mathcal{E}(z; \tilde{x}^\ell | p \leftarrow j)]$, where by $\tilde{x}^\ell | p \leftarrow i$ we mean the product distribution $\tilde{x}^\ell$ except for the p-th factor which is replaced by the distribution concentrated on strategy i. Since this expression can be computed by two invocations of the polynomial algorithm $\mathcal{E}$, this completes the proof of the theorem. ■

We next show that essentially all known succinct game genres of polynomial type have the polynomial expectation property. Interestingly, optimization over correlated equilibria is an NP-hard problem in many of these classes of games (see Section 5).

Classes of Succinct Games

Graphical Games. In a graphical game G [26], input z describes an undirected graph H (where $H(p)$ is the set of neighbors of p, including p) and, for each p a game G_p, explicitly represented, with players in $H(p)$.

All graphical games have the polynomial expectation property. Given z, p, and $x^1, \ldots, x^n$, $\mathcal{E}$ will calculate explicitly the expected utility under this play for the game G_p, going over all strategy profiles of G_p and ignoring the mixed strategies of players not in $H(p)$. This is obviously polynomial in z.

In sharp contrast, efficient optimization over the set of correlated equilibria of a graphical game is possible only in special cases. We show in Section 5 that for a graphical game with bounded tree-width, an optimal correlated equilibrium (with respect to any linear function of players' payoffs) can be computed in polynomial time. We also show that this problem is NP-hard even in bounded-degree bipartite graphical games.

Polymatrix Games. In a polymatrix game G [24], input z describes $\binom{n}{2}$ 2-person games G_{pq} with each player p having the same strategy set S_p in each. The algorithm U on input $p, s_1, \ldots, s_n$ computes $\sum_{q \neq p} u^p_{pq}(s_p, s_q)$, where we represent the utilities of G_{pq} by u^p_{pq}.

Polymatrix games have the polynomial expectation property. Given z, a player p, and a product distribution $x^1, \ldots, x^n$, algorithm $\mathcal{E}$ will return the sum (by linearity of expectation) over all $q \neq p$ of $\mathbf{E}_{x^p, x^q} u^p$. Notice that each term can be exhaustively computed in time proportional to the representation of G_{pq}, and thus the total calculation is polynomial in z.

Hypergraphical Games. In a hypergraphical game z describes a hypergraph $H = ([n], E)$ and, for each hyperedge $h \in E$, an explicit game G_h involving the players in h. Player p has the same set of strategies S_p in all games s/he is involved. The payoff $U(z, p, s_1, \ldots, s_n)$ is the sum of all payoffs of p in all games involving him/her.

These games generalize polymatrix games and graphical games but also have the polynomial expectation property.

Congestion Games and Local Effect Games. In congestion games [40] z describes a set E of resources, and, for each player p, a set of strategies $S_p \subseteq 2^E$; that is, each strategy is a subset of E. Input z also describes explicitly, for each $e \in E$, a *delay function* $d_e : \{1, \ldots, n\} \mapsto Z^+$. The algorithm U, on input z, $p \leq n$, and $s_q \in S_q$ for each $q \leq n$, will compute the total delay $\sum_{e \in s_p} d_e(c_e(s_1, \ldots, s_n))$, where $c_e(s_1, \ldots, s_n)$, the congestion of e in the strategy profile, is the cardinality of $\{q \leq n : e \in s_q\}$. (Thus U computes the "negative utility" of a player in congestion games.)

To see that congestion games have the polynomial expectation property, we need to show how to compute the expectation of U when that s_q's are distributed according to some product distribution $x^1, \ldots, x^n$. This expectation equals

$$\sum_{s_p \in S_p} x^p_{s_p} \cdot \mathbf{E}_{s_{-p}} \left[\sum_{e \in s_p} d_e(c_e(s_1, \ldots, s_n)) \,\middle|\, s_p \right] = \sum_{s_p \in S_p} x^p_{s_p} \cdot \sum_{e \in s_p} \mathbf{E}_{s_{-p}} \left[d_e(1 + |\{q \neq p \,:\, e \in s_q\}|) \right].$$

We compute each of the expectations on the right-hand side as follows. For each $e \in E$ and $q \leq n$, we define $w_q(e)$ to be $\sum_{s \in S_q \,:\, e \in s} x^q_s$ — that is, the probability that player q will use a strategy containing e. Then we compute for each $r \leq n-1$, via straightforward dynamic programming, the probability $P_r(e)$ that $\sum_{q \neq p} b_q(e) = r$, where the $b_q(e)$'s are independent Bernoulli random variables equal to 1 with probability $w_q(e)$ and zero otherwise. With these ingredients in place, it is easy now to compute the expectation of $d_e(1 + |\{q \neq p \,:\, e \in s_q\}|)$ as $\sum_{r=0}^{n-1} d_e(1+r) \cdot P_r(e)$. Note that this argument applies even in congestion games with player-specific payoffs (as studied by Milchtaich [33]).

Local effect games [31] generalize congestion games in that they allow the cost of e to p to be the sum of terms of the form $d_{f,e}(c_f(s_1, \ldots, s_n))$ for all $f \in E$; that is, the congestion of each resource affects additively the cost of the other resources. It is not hard to see that, combining the above ideas with linearity of expectation, we have the polynomial expectation property in this larger class of games as well. Again, this argument applies even with player-dependent costs.

Facility Location and Network Design Games. In a *facility location game*, see for example [10], players choose one of a set of facility locations, each with a given cost; the cost of each facility is then divided equally between the players who chose it, while each player also pays her distance to the facility she chose. In the more general network design game from [1], players choose paths in a graph to connect their terminals (we assume here that the paths are given explicitly in the input so that the game has polynomial type), and the cost of each edge is shared equally among the players who included it in their path. Both of these types of games correspond to congestion games with *non-increasing* delay functions; the preceding argument carries over and proves that both have the polynomial expectation property.

Scheduling Games. In a *scheduling game* (generalizing a definition in [18]) strategies are machines in a finite set M, and the cost of choosing machine $m \in M$ is the sum over all players p who also chose machine m, of given terms of the form $t(m, p)$ (the time it takes to execute p's job on m). Adapting the argument for congestion games shows that scheduling games have the polynomial expectation property.

Symmetric Games. In a symmetric game the type is of the form $n, m, m, \ldots, m$ (that is, all players have the same strategies) and z explicitly lists an integer utility for each $i \leq n$ (the strategy of the player) and each partition of $n-1$ into m parts (the strategy distribution of the other players).

In Sections 5 and 6 we show how to efficiently optimize over the correlated equilibria of a symmetric game. It is interesting to note, as a matter of curiosity, that the approach in this section for finding a single equilibrium also works for symmetric games. But it needs some work, rather than a direct application of Theorem 4.1: First we must make sure that our algorithm can be coached to return a convex combination of *symmetric* product distributions (where all players play the same). And, when the input is restricted to such distributions, it turns out that symmetric games do have the polynomial expectation property: The basic idea is to compute the probability that a particular partition of $n-1$ into m parts will be realized under a symmetric product distribution, which can be done via an extension of the dynamic programming algorithm described for congestion games.

We summarize the above discussion as follows:

Theorem 4.2 *The following classes of succinct games have a polynomial correlated equilibrium scheme:*

1. *Graphical games.*
2. *Polymatrix games.*
3. *Hypergraphical games.*
4. *Congestion games.*
5. *Local effect games.*
6. *Facility location games.*
7. *Network design games.*
8. *Scheduling games.*
9. *Symmetric games.*

5 Computing Optimal Correlated Equilibria

The previous section shows how to efficiently compute *some* correlated equilibrium of a succinct game. But a stronger result is true for explicitly represented games: the *optimal* correlated equilibrium, with respect to an arbitrary linear function of the players' expected payoffs, can be computed in polynomial time. Is the same true for succinct games? This section presents both positive (Sections 5.3 and 5.4) and negative (Section 5.5) results for this question.

5.1 Motivation

Why doesn't the primal-dual pair (P) and (D) from Section 3 extend to the more general optimization problem? As an example, suppose we want to maximize the sum of the players' expected payoffs. The obvious approach is to supplement the primal problem (P) with the objective

$$\max \sum_{s \in \mathcal{S}} u_s x_s,$$

where u_s denotes $\sum_p u_s^p$, subject to the additional constraint $\sum_s x_s = 1$. The new constraint supplies the dual problem with a new variable—z, say—and the new coefficients in the primal objective are passed on to the right-side of the dual. As a consequence, the new dual constraints are of the form

$$(U_s)^T y \leq -u_s + z, \tag{1}$$

where U_s denotes the column of the matrix U corresponding to strategy s. (In the previous dual (D), the right-hand side is -1 for every constraint.) The ellipsoid method applies only if there is a separation oracle for these dual constraints. Previously, given a candidate dual solution y, it was enough to exhibit a non-negative and non-zero vector x such that $xU^T y = 0$. Now, given a candidate dual solution (y, z), the right-hand side of (1) may be either positive or negative, and computing such a vector x is no longer sufficient. Even when y is identically zero, the separation problem specializes to checking whether or not there is a strategy profile with total utility greater than a given threshold z—an NP-hard problem for many of the succinct games treated in Section 4 (see Section 5.5).

On the other hand, in important examples such as symmetric and graphical games, the exponentially many dual constraints (1) can be classified into a polynomial number of player-specific "equivalence classes", such that feasibility within a class can be summarized by a single constraint. The next section formalizes this idea and shows that, provided a certain "consistency problem" across equivalence classes of different players can be solved efficiently, optimal correlated equilibria can be computed in polynomial time.

5.2 Reduced Forms of Games

Motivated by the preceding discussion, we begin with a definition of "equivalent outcomes". Such a definition is necessarily player-specific, as generally no two outcomes are equivalent from every player's perspective.

Definition 5.1 Let $G = (S_1, \ldots, S_n, u^1, \ldots, u^n)$ be a game. For $p = 1, 2, \ldots, n$, let $P_p = \{C_p^1, \ldots, C_p^{r_p}\}$ be a partition of S_{-p} into r_p classes.

(a) For a player p, two strategy profiles s and s' are *p-equivalent* if $s_p = s'_p$, and both s_{-p} and s'_{-p} belong to the same class of the partition P_p.

(b) The set $\mathcal{P} = \{P_1, \ldots, P_n\}$ of partitions is a *reduced form of* G if $u_p(s) = u_p(s')$ whenever s and s' are p-equivalent.

The *size* of a reduced form of a game is the total number of classes in its partitions, plus the number of bits needed to describe player payoffs.

Example 5.2 Let G be an anonymous game in which each player has the strategy set S and player p has utility function u^p. Then G has a natural reduced form with size equal to that of its succinct description: for each player p, there is one partition class of P_p for each ordered partition of $n-1$ (the other players) into m parts (the strategies).

Example 5.3 In the natural reduced form of a graphical game G, the partition P_p of a player p has a class C_p^j for each assignment j of strategies to the players that are neighbors of p in the graphical game. The size of this reduced form is the same as that of the succinct description of G.

Succinct games need not have succinct reduced forms in the sense of Definition 5.1. Indeed, a reduced form of a game requires at least one partition class per distinct payoff that can be earned by a player. In games where payoffs are defined as a function (such as the sum) of the input, including polymatrix and congestion games, the number of distinct payoffs (and hence the size of a reduced form) can be exponential in that of the game's natural description. This should not be viewed as a failure of Definition 5.1, however, as we show in Section 5.5 that computing optimal correlated equilibria is NP-hard in such games.

5.3 Algorithmic Framework

Our algorithm for computing optimal correlated equilibria is based on the following linear programming relaxation for a game G and reduced form $\mathcal{P} = \{P_p\}$. Write $\mathcal{S}_p(j,\ell)$ for the set of strategy profiles s with $s_p = \ell$ and $s_{-p} \in C_p^j$, and $u^p(j,\ell)$ for the payoff to player p in all of these profiles (well defined by Definition 5.1). Consider maximizing

$$\sum_{p=1}^{n} w_p \left(\sum_{j=1}^{r_p} \sum_{\ell \in S_p} u^p(j,\ell) x_p(j,\ell) \right), \tag{2}$$

where the w_p's are arbitrary real-valued player weights, subject to

$$\sum_{j=1}^{r_p} \left[u^p(j,\ell) - u^p(j,\ell') \right] x_p(j,\ell) \geq 0 \tag{3}$$

for all players p and strategies $\ell, \ell' \in S_p$;

$$\sum_{j=1}^{r_p} \sum_{\ell \in S_p} x_p(j,\ell) = 1 \tag{4}$$

for all players p; and

$$x_p(j,\ell) \geq 0 \tag{5}$$

for all players p, indices $j \in \{1, 2, \ldots, r_p\}$, and strategies $\ell \in S_p$. We interpret the decision variable $x_p(j, \ell)$ as the aggregate probability assigned to the strategy profiles of $\mathcal{S}_p(j, \ell)$. The size of this linear program is polynomial in the size of the given reduced form $\mathcal{P}$.

The linear program defined by (2)–(5) is closely related to the primal problem (P) from Section 3. Roughly, the former program can be obtained from the latter one in two steps: first, each decision variable (one per strategy profile) is replicated n times, one for each player; second, groups of the decision variables (strategy profiles) corresponding to a player p are added together according to the classes of the partition P_p. The definition of a reduced form ensures that the second step is well defined, in that only decision variables with identical coefficients are grouped and added together.

The next lemma records the fact that the linear program defined by (2)–(5) is a relaxation for the problem of computing an optimal correlated equilibrium; we omit the straightforward proof.

Lemma 5.4 *Let G have a reduced form $\{P_p\}$ and let $\{z_s\}_{s \in \mathcal{S}}$ be a correlated equilibrium of G. For each player p, index $j \in \{1, \ldots, r_p\}$, and strategy $\ell \in S_p$, define $x_p(j, \ell)$ by*

$$x_p(j, \ell) = \sum_{s \in \mathcal{S}_p(j,\ell)} z_s. \tag{6}$$

Then $\{x_p(j, \ell)\}$ satisfies (3)–(5)*, and the objective function value of x in* (2) *is the weighted expected payoff $\sum_{p=1}^{n} w_p \sum_{s \in \mathcal{S}} u^p(s) z_s$ under z.*

Why do we only claim that (2)–(5) is a *relaxation* of the problem of computing an optimal correlated equilibrium, rather than an exact *formulation*? Consider a feasible solution x to (3)–(5). We say that x *extends* to a non-negative vector z, defined on $\mathcal{S}$, if x is induced by z according to (6). The following converse to Lemma 5.4 is easy to show.

Lemma 5.5 *Let G have a reduced form $\{P_p\}$, let x be a feasible solution to* (3)–(5)*, and suppose that x extends to z. Then z is a correlated equilibrium of G, and the weighted expected payoff $\sum_{p=1}^{n} w_p \sum_{s \in \mathcal{S}} u^p(s) z_s$ under z equals the objective function value* (2) *of x.*

Unfortunately, because the strategy profiles corresponding to two variables $x_p(j, \ell)$ and $x_{p'}(j', \ell')$ for distinct player p, p' generally overlap, a feasible solution to (3)–(5) need not extend to a vector z defined on $\mathcal{S}$. (See also Example 6.1.) We therefore would like to optimize only over a subset of (3)–(5)—the feasible solutions that are also extensible. This motivates our final definition, which is for a procedure that efficiently generates supplementary inequalities that enforce extensibility. Theorem 5.7 shows that the complexity of this procedure controls that of the problem of computing optimal correlated equilibria.

Definition 5.6 Let $\mathcal{P}$ be a reduced form of a game G. The *separation problem for $\mathcal{P}$* is the following algorithmic problem:

> Given rational numbers $y_p(j, \ell)$ for all $p \le n$, $j \le r_p$, and $\ell \in S_p$, is there a strategy profile s such that
>
> $$\sum_{(p,j,\ell)\,:\, s \in \mathcal{S}_p(j,\ell)} y_p(j, \ell) < 0? \tag{7}$$

Note that the left-hand side of (7) involves precisely n summands, one for each player.

Section 5.4 gives concrete examples of such separation problems. We conclude this section by proving that a tractable separation problem is all that is required for the efficient computation of a correlated equilibrium.

Theorem 5.7 *Let G be a game, $\mathcal{P}$ a reduced form of G, and w a set of real-valued player weights. If the separation problem for $\mathcal{P}$ can be solved in polynomial time, then a correlated equilibrium of G that maximizes the weighted sum of players' expected payoffs can be computed in time polynomial in the size of $\mathcal{P}$.*

Proof: Let $G = (S_1, \ldots, S_n, u^1, \ldots, u^n)$ be a game and $\mathcal{P}$ a reduced form such that the separation problem for $\mathcal{P}$ is solvable in polynomial time. Consider the linear program given by (2)–(5) and a feasible solution x. The problem of checking whether or not x extends to a vector z—a correlated equilibrium of G, by Lemma 5.5—is naturally formulated as a linear system $Az = x, z \geq 0$. The matrix A has exponentially many columns (one per strategy profile of G) but polynomially many rows (one per decision variable $x_p(j, \ell)$ of (2)–(5)).

By Farkas's Lemma, the solution x can be extended to a correlated equilibrium of G if and only if for every y with $y^T A \geq 0$, $y^T x \geq 0$. Note that the vector y is indexed by the variables in (3)–(5).

This observation suggests extra inequalities to add to (3)–(5) to ensure extensibility: for every vector y with $y^T A \geq 0$, include the inequality $y^T x \geq 0$. Every such inequality is valid in the sense that every correlated equilibrium of G induces a solution to (3)–(5) that also satisfies this extra inequality.

Naively, there are infinitely many such extra inequalities to add. Fortunately, we need only include those inequalities that can arise as an optimal solution to the following problem:

(*) Given x satisfying (3)–(5), minimize $y^T x$ subject to $y^T A \geq 0$.

Since problem (*) is a linear program, the minimum is always attained by one of the finitely many basic solutions [11]. Moreover, in all such basic solutions, the vector y can be described with a number of bits polynomial in $\mathcal{P}$ [20, §6.2].

We have therefore defined a finite linear system, which we will call the *full linear system* for $\mathcal{P}$, so that x is a solution to the full linear system if and only if x can be extended to a correlated equilibrium of G. By Lemmas 5.4 and 5.5, the optimal solution to the full linear system extends to an optimal correlated equilibrium of G.

We can efficiently compute a solution to the full linear system via the ellipsoid algorithm [20, 27], provided we can define a polynomial-time separation oracle—an algorithm that takes as input a candidate solution and, if the solution is not feasible, produces a violated constraint. Such a separation oracle is tantamount to a polynomial-time algorithm for problem (*), a linear program with exponentially many constraints (indexed by strategy profiles). We can solve problem (*) with a second application of the ellipsoid method. Here, the separation oracle required is precisely the separation problem for $\mathcal{P}$ (Definition 5.6) which, by assumption, admits a polynomial-time algorithm. The proof is therefore complete. ■

The optimal correlated equilibrium computed by the above algorithm can be sampled from in polynomial time. In other words, the algorithm provides a refinement of a polynomial correlated equilibrium scheme (Section 2) that is guaranteed to sample the strategy space of the given game G according to an optimal correlated equilibrium. Indeed, the algorithm explicitly computes a solution x that can be extended to an optimal correlated equilibrium z. As in the proof of Theorem 5.7, the problem of computing z from x can be formulated as a linear program (with the all-zero objective, say), the dual of which can be solved in polynomial time via the ellipsoid method. The ellipsoid method will only generate a polynomial number of dual constraints during its execution, and these constraints suffice to compute an optimal dual solution. The primal variables (strategy profiles) that correspond to these dual constraints then suffice to solve the primal problem optimally. Since this "reduced primal" has a polynomial number of variables and constraints, it can be solved in polynomial time, yielding an optimal correlated equilibrium z. (Strategy profiles not included in the "reduced primal" are understood to have zero probability.) Since z has polynomial-sized support, it can be directly sampled from in polynomial time.

5.4 Applications

We now demonstrate the power of Theorem 5.7 by applying it to anonymous (and hence symmetric) games, as well as to graphical games.

Theorem 5.8 *Optimal correlated equilibria of anonymous games can be computed in polynomial time.*

Proof: Given an anonymous game, we consider the reduced form $\mathcal{P}$ described in Example 5.2. The separation problem for $\mathcal{P}$ is then: given rational numbers $y_p(j,\ell)$ for each player p, each ordered partition j of $n-1$ into m parts, and each choice ℓ for player p's strategy, is there a strategy profile s with

$$\sum_{(p,j,\ell)\,:\,s\in\mathcal{S}_p(j,\ell)} y_p(j,\ell) < 0? \tag{8}$$

We can solve this problem in polynomial time as follows. We first enumerate all ordered partitions of n into m parts; the number of such partitions is polynomial in the size of $\mathcal{P}$. For each such partition $(k_1,\ldots,k_m)$, we minimize the left-hand side of (8) over all ways of assigning k_ℓ players to strategy ℓ (for each ℓ); this step can be implemented in polynomial time using min-cost flow. Theorem 5.7 now implies the theorem. ■

Theorem 5.9 *Optimal correlated equilibria of graphical games with bounded tree-width can be computed in polynomial time.*

Proof: Let G be a graphical game and let $\mathcal{P}$ be the natural reduced form described in Example 5.3. The separation problem for $\mathcal{P}$ is the following: given rational numbers $y_p(j,\ell)$ for each player p, each set j of strategy choices of p's neighbors, and each choice ℓ for player p's strategy, is there a strategy profile s with

$$\sum_{(p,j,\ell)\,:\,s\in\mathcal{S}_p(j,\ell)} y_p(j,\ell) < 0? \tag{9}$$

If the graph H in the graphical game G has bounded tree-width, then this problem can be solved in polynomial time by straightforward dynamic programming; we sketch the main step for completeness.

Consider a node w in a (rooted) tree decomposition T of H; if such a decomposition is not explicitly given, one can be computed in polynomial time provided H has bounded tree-width (see e.g. [29]). Let A_w and B_w denote the vertices of H that correspond to w and to the subtree of T rooted at w, respectively. Let $\Gamma(A_w)$ denote the vertices that are either in A_w or have a neighbor in A_w.

There is one subproblem for each node $w \in T$ and each assignment of strategies to a subset of the vertices of $\Gamma(A_w)$; since $|A_w| = O(1)$, the number of subproblems is polynomial in the description of the graphical game G. For each subproblem, the goal is to compute a strategy profile $s \in \mathcal{S}$ that minimizes

$$\sum_{(p,j,\ell)\,:\,p\in B_w, s\in\mathcal{S}_p(j,\ell)} y_p(j,\ell)$$

subject to the additional constraint that the strategy profile s should adopt the prescribed assignments for players of $\Gamma(A_w)$. To solve such a subproblem, given solutions to the subproblems corresponding to the children of w in T, the algorithm first enumerates all possible extensions of the prescribed strategy assignments to assignments for *all* players of $\Gamma(A_w)$, thereby fixing the value of $\sum_{(p,j,\ell)\,:\,p\in A_w} y_p(j,\ell)$, and then computes strategies for the rest of the players in B_w using the appropriate precomputed solutions for the children of w in T. Since $|A_w| = O(1)$, each subproblem can be solved in time polynomial in the description of G. The final solution—the strategy profile minimizing the left-hand side of (9)—is the solution to the subproblem at the root of T that does not pre-assign strategies to any of the players. ■

As we noted in the Introduction, Theorem 5.9 was first proved by Kakade et al. [25] for tree graphical games using tools from probabilistic inference.

5.5 Hardness Results for Optimal Correlated Equilibria

Our results for computing optimal correlated equilibria in succinct games (Theorem 5.7) are less all-encompassing than our results for computing arbitrary correlated equilibria (Sections 3 and 4). We now show that this difference is fundamental: in all of the classes of games not covered by Theorems 5.8 and 5.9, computing an optimal correlated equilibrium is an NP-hard problem. This negative result dispels any lingering concern that our approach to optimizing over correlated equilibria was too modest: for these other classes of games, there is *no* linear system that characterizes the correlated equilibria and can be optimized over in polynomial time (assuming $P \neq NP$).

Theorem 5.10 *In the following classes of succinct games, it is NP-hard to compute a correlated equilibrium that maximizes the sum of the players' expected payoffs:*

1. *Bounded-degree, bipartite graphical games.*
2. *Polymatrix games.*
3. *Hypergraphical games.*
4. *Congestion games.*
5. *Local effect games.*
6. *Facility location games.*
7. *Network design games.*
8. *Scheduling games.*

The hardness result for polymatrix games is due to B. von Stengel (personal communication, July 2004); we provide proofs for all of the other cases. Most of our reductions are based on the following lemma. We say that an outcome of a game is *dominant* if it simultaneously maximizes the payoff earned by each of the players.

Lemma 5.11 *The problem of checking whether or not a game of polynomial type has a dominant outcome reduces to the problem of computing a correlated equilibrium that maximizes the sum of the players' expected payoffs.*

Proof: Given a game of polynomial type, the maximum-possible payoff that can be earned by each player, and the sum A of these payoffs, can be computed in polynomial time. If the game has a dominant outcome, then this outcome is also a correlated equilibrium with total payoff A. If there is no dominant outcome, then there is no outcome with total payoff at least A, and hence no distribution over outcomes (correlated equilibrium or otherwise) with total expected payoff at least A. ■

Proof of Theorem 5.10: We first show the NP-hardness of deciding the existence of a dominant outcome (and hence computing optimal correlated equilibria) in bounded-degree, bipartite graphical games. This hardness result obviously carries over to hypergraphical games. We reduce from E3SAT-5, the special case of SAT in which each clause has exactly 3 literals and each variable occurs in at most 5 clauses. Given such a SAT formula, we construct a bipartite graph $H = (V_1, V_2, E)$ where vertices of V_1 correspond to clauses ("clause vertices") and vertices of V_2 correspond to variables ("variable vertices"). Edge (v, w) is present in H if and only if $v \in V_1$, $w \in V_2$, and the variable corresponding to w appears in the clause corresponding to v. Each variable vertex has two strategies, "true" and "false". Each clause vertex has three strategies, one for each of the variables in the clause. Variable vertices always receive zero payoff. The payoff of a clause vertex is 1 if its strategy corresponds to a variable vertex that has picked a strategy (i.e., a truth assignment) that satisfies the clause, and zero otherwise. Since H has bounded degree, its description size is polynomial in that of the given SAT formula. Since H has a dominant outcome if and only if the give SAT formula is satisfiable, the reduction is complete.

We proceed to congestion (and hence local effect) games. The reduction is from EXACT COVER BY 3-SETS (X3C). Suppose the given instance has ground set X with $3n$ elements and m 3-sets $A_1, \dots, A_m$. We construct a congestion game with ground set $E = X \cup \{x_0\}$. Players in the congestion game correspond to sets A_i of the X3C instance, and each has two strategies: A_i and $\{x_0\}$. The delay function d_e for every element $e \in E$ other than x_0 is given by $d_e(1) = 1$ and $d_e(y) = 2$ for all $y \geq 2$. Element x_0 has delay function $d_{x_0}(y) = 3$ for $y \leq m-n$ and $d_{x_0}(y) = 4$ for $y > m-n$. This congestion game has a dominant outcome if and only if the X3C instance admits an exact cover. Lemma 5.11 then implies that computing optimal correlated equilibria in congestion games is NP-hard. Similar reductions from X3C establish hardness for facility location and network design games.

Finally, we address scheduling games. Checking whether or not such a game has a dominant outcome can be accomplished in polynomial time, so we furnish a direct reduction from X3C, based on one of Lenstra, Shmoys, and Tardos [30]. The m machines correspond to sets, the $3n$ jobs (players) to elements of the X3C instance. The job p corresponding to an element x requires 1 unit of processing time on machines that correspond to sets that contain x, and 2 units on other machines. There are also $m-n$ "dummy players" whose jobs require 3 units of processing time on every machine. Clearly, outcomes with makespan 3 are in bijective correspondence with exact covers of the given X3C instance. When such an outcome exists, it is a correlated equilibrium. Moreover, in this case, the only correlated equilibria that minimize the sum of the players' costs are job assignments with makespan 3. It is therefore NP-hard to compute optimal correlated equilibria in scheduling games. ■

6 Explicit Descriptions of Correlated Equilibria

Theorem 5.7 provides a general algorithm for computing optimal correlated equilibria, but it falls short of the classical guarantee for explicitly represented games in one respect: it does not give an *explicit* polynomial-size linear system that characterizes the correlated equilibria of a succinct game. Explicit linear characterizations are necessarily more problem-specific than the generic algorithms presented in Sections 3–5, but they remain important for two reasons. The first is conceptual: an explicit description provides a concrete understanding of the set of correlated equilibria. The second is computational: given an explicit linear system describing the correlated equilibria of a game, optimal correlated equilibria can be efficiently computed using any polynomial-time linear programming algorithm (such as interior-point methods), and the ellipsoid method is not required. This section gives explicit polynomial-size linear systems for the correlated equilibria of both anonymous games and tree graphical games.

6.1 Anonymous Games

Recall from Example 5.2 that an anonymous game with n players and m strategies has a natural reduced form $\mathcal{P}$, where there is one partition class of P_p for each ordered partition j of $n-1$ (the other players) into m parts (the strategies). We denote the set of all such ordered partitions by $\Psi(n-1, m)$ and identify each element $j \in \Psi(n-1, m)$ with a nonnegative integral m-vector with sum of components equal to $n-1$.

We begin with the linear system (3)–(5) from the previous section, with one decision variable $x_p(j, \ell)$ for each player p, each $j \in \Psi(n-1, m)$, and each strategy $\ell \in S$. Recall that a feasible solution to this system is *extensible*, and corresponds to a correlated equilibrium of the anonymous game, if and only if there is a probability distribution z over the strategy profiles $\mathcal{S}$ that satisfies the intended aggregations given in (6).

Example 6.1 Not all feasible solutions to (3)–(5) extend to correlated equilibria, even for 2-player, 2-strategy symmetric games. To see this, set all utilities to zero and set $x_1(\{2\}, 1) = x_2(\{2\}, 1) = 1$, where $\{2\}$ denotes the partition class in which the other player selects the second strategy.

To home in on the extensible solutions, we add a polynomial number of additional decision variables and constraints. We include a new nonnegative decision variable $x(k)$ for each ordered partition $k \in \Psi(n, m)$ of n into m parts; these are intended to keep track of the distribution of all players among the m strategies. Looking ahead, we have in mind a two-stage sampling procedure: first a partition of $\Psi(n, m)$ is chosen, and then an assignment of players to classes of the partition is selected. To implement this, we add the following new constraints, where e_ℓ denotes the ℓth standard basis vector in $\mathcal{R}^m$:

$$\sum_{\ell \,:\, k_\ell > 0} x_p(k - e_\ell, \ell) = x(k) \tag{10}$$

for every $k \in \Psi(n, m)$ and player p;

$$\sum_{p=1}^{n} x_p(k - e_\ell, \ell) = k_\ell \cdot x(k) \tag{11}$$

for every $k \in \Psi(n, m)$ and every strategy ℓ with $k_\ell > 0$; and

$$\sum_{k \in \Psi(n,m)} x(k) = 1. \tag{12}$$

Given these new constraints, the original normalization constraints (4) are redundant and can be dropped.

The new constraints are valid in the sense that they are satisfied by correlated equilibria of the anonymous game.

Lemma 6.2 *Let G be an n-player, m-strategy anonymous game and let $\{z_s\}_{s\in\mathcal{S}}$ be a correlated equilibrium of G. For each player p, ordered partition $j \in \Psi(n-1,m)$, and strategy $\ell \in S$, define $x_p(j,\ell)$ as in* (6). *For each $k \in \Psi(n,m)$, let $\mathcal{S}(k)$ denote the strategy profiles in which k_ℓ players choose strategy ℓ for each $\ell \in S$, and define $x(k)$ by*

$$x(k) = \sum_{s\in\mathcal{S}(k)} z_s.$$

Then $\{x_p(j,\ell)\}_{p,j,\ell}$ and $\{x(k)\}_k$ satisfy (10)–(12).

Proof: Since z is a probability distribution on $\mathcal{S}$ and the $\mathcal{S}(k)$'s form a partition of $\mathcal{S}$, constraint (12) is clearly satisfied. Similarly, the constraints (10) hold because, for each p and $k \in \Psi(n,m)$, the collection $\{\mathcal{S}_p(k-e_\ell,\ell)\}_{\ell : k_\ell > 0}$ is a partition of $\mathcal{S}(k)$. Finally, the constraints (11) hold because, for each $k \in \Psi(n,m)$ and $\ell \in S$, each strategy profile $s \in \mathcal{S}(k)$ inhabits precisely k_ℓ of the sets $\{\mathcal{S}_p(k-e_\ell,\ell)\}_{p=1}^n$—those of the k_ℓ players choosing strategy ℓ in the profile s. ■

We now show that the additional variables and constraints in (10)–(12) provide the desired explicit description of correlated equilibria in anonymous games.

Theorem 6.3 *For every anonymous game G, every nonnegative feasible solution to* (3) *and* (10)–(12) *extends to a correlated equilibrium of G. Moreover, strategy profiles can be sampled from this correlated equilibrium in polynomial time.*

Proof: Let $\{x_p(j,\ell)\}_{p,j,\ell}$ and $\{x(k)\}_k$ denote a nonnegative feasible solution to (3) and (10)–(12). We prove the theorem by giving a polynomial-time algorithm that outputs a random strategy profile $s \in \mathcal{S}$ with the properties that

$$\mathbf{Pr}[s \in \mathcal{S}(k)] = x(k) \tag{13}$$

for every $k \in \Psi(n,m)$ and

$$\mathbf{Pr}[s \in \mathcal{S}_p(j,\ell)] = x_p(j,\ell) \tag{14}$$

for every $p \in \{1,\ldots,n\}$, $j \in \Psi(n-1,m)$, and $\ell \in S$. (The extension z of x is then implicitly defined by $z_s = \mathbf{Pr}[s]$ for every $s \in \mathcal{S}$.)

The algorithm first samples, by enumeration, an ordered partition $k \in \Psi(n,m)$ according to the distribution $\{x(k)\}$. Given k, it constructs the complete bipartite graph $K_{n,n} = (V_1, V_2, E)$. Vertices of V_1 correspond to players; for each $\ell \in S$, k_ℓ vertices of V_2 are deemed "ℓ-vertices". Perfect matchings in this graph are in bijective correspondence with strategy profiles of $\mathcal{S}(k)$.

We now construct a fractional perfect matching in this bipartite graph. We can assume that $x(k) > 0$. For each player $p \in V_1$ and ℓ-vertex $w \in V_2$, we assign fractional weight

$$y_{p,w} = \frac{x_p(k-e_\ell,\ell)}{k_\ell \cdot x(k)} \tag{15}$$

to the edge (p, w). For each player p and strategy ℓ, the total amount of y-weight on edges between p and ℓ-vertices is $x_p(k - e_\ell, \ell)/x(k)$; by (10), it follows that the total amount of y-weight incident to each player is 1. Also, constraints (11) ensure that the total amount of y-weight incident to each vertex of V_2 is 1. By Birkhoff's Theorem, we can express y as a probability distribution over at most n^2 perfect matchings; moreover, this decomposition can be computed in polynomial time. Finally, we sample from this distribution over perfect matchings (by enumeration) and return the corresponding strategy profile.

By the definition of its first step, this sampling procedure meets requirement (13). To verify (14) and complete the proof, fix p, $j \in \Psi(n-1, m)$, and $\ell \in S$. We have

$$\begin{aligned}
\mathbf{Pr}[s \in \mathcal{S}_p(j,\ell)] &= \sum_{k \in \Psi(n,m)} \mathbf{Pr}[s \in \mathcal{S}_p(j,\ell) | s \in \mathcal{S}(k)] \cdot \mathbf{Pr}[s \in \mathcal{S}(k)] \\
&= x(j+e_\ell) \cdot \mathbf{Pr}[s \in \mathcal{S}_p(j,\ell) | s \in \mathcal{S}(j+e_\ell)] \qquad (16) \\
&= x(j+e_\ell) \cdot \frac{x_p(j,\ell)}{x(j+e_\ell)} \qquad (17) \\
&= x_p(j,\ell),
\end{aligned}$$

where (16) follows from (13) and the fact that $\mathbf{Pr}[s \in \mathcal{S}_p(j,\ell) | s \in \mathcal{S}(k)] = 0$ unless $k = j + e_\ell$, and (17) follows from the definition (15) of the fractional matching y and the fact that the probability that p is assigned to ℓ equals the y-weight on edges between the vertex p and the ℓ-vertices of V_2. ■

6.2 Tree Graphical Games

We now give an explicit polynomial-size linear system describing the correlated equilibria of a tree graphical game. This result appears, essentially, in Kakade et al. [25]. Here, we give a new and arguably more direct proof.

Let G be a tree graphical game with graph T. Recall (Example 5.3) that this game has a natural reduced form $\mathcal{P}$, where the partition P_p of each player p has r_p classes, one for each assignment of strategies to the players that are neighbors of p. A pair (j, ℓ), with $j \in \{1, 2, \ldots, r_p\}$ and $\ell \in S_p$, thus dictates a strategy to all of the players of $H(p)$. As with anonymous games, we must augment the linear system (3)–(5) to ensure the extensibility of a feasible solution.

For a neighboring pair p, q of vertices in T, let S_{pq} denote the possible joint strategy assignments to p and q. For $s_{pq} \in S_{pq}$ and a pair (j, ℓ) (with either $j \leq r_p$ and $\ell \in S_p$, or $j \leq k_q$ and $\ell \in S_q$), we abuse notation and write $(j, \ell) = s_{pq}$ to mean that the strategy assignments dictated to $\{p.q\}$ by the pair (j, ℓ) agree with s_{pq}. We then add the following constraints, called *Intersection Consistency Constraints* in [25]:

$$\sum_{j \leq r_p, \ell \in S_p \,:\, (j,\ell) = s_{pq}} x_p(j,\ell) = \sum_{j \leq k_q, \ell \in S_q \,:\, (j,\ell) = s_{pq}} x_q(j,\ell) \qquad (18)$$

for every neighboring pair p, q of players and $s_{pq} \in S_{pq}$. Every solution x induced by a probability distribution z over strategy profiles according to (6) satisfies (18).

The linear system defined by (3)–(5) and (18) has size polynomial in the length of the description of the tree graphical game G. We now show that it characterizes the correlated equilibria of G.

Theorem 6.4 *For every tree graphical game G, every feasible solution to* (3)–(5) *and* (18) *extends to a correlated equilibrium of G. Moreover, strategy profiles can be sampled from this correlated equilibrium in polynomial time.*

Proof: Let $\{x_p(j,\ell)\}_{p,j,\ell}$ denote a feasible solution to (3)–(5) and (18). As in the proof of Theorem 6.3, we proceed by giving a randomized polynomial-time algorithm that outputs a strategy profile s such that

$$\mathbf{Pr}[s \in \mathcal{S}_p(j,\ell)] = x_p(j,\ell) \tag{19}$$

for every $p \in \{1,\ldots,n\}$, $j \in \{1,2,\ldots,r_p\}$, and $\ell \in S_p$.

Let T be the graph of the given graphical game G, rooted at an arbitrary player p. The idea is to randomly and irrevocably assign strategies to the players via breadth-first search. The algorithm first randomly chooses strategy assignments for $H(p)$—the root p and its children—according to the distribution $\{x_p(j,\ell)\}_{j,\ell}$. (This is a probability distribution by (4)–(5).) The algorithm then considers the children of p in an arbitrary order. For such a player q, let $s_{pq} \in S_{pq}$ denote the strategies chosen for p and q in the first step. The algorithm randomly assigns strategies to the children of q according to the distribution

$$\left\{ \frac{x_q(j,\ell)}{\sum_{(j',\ell')\,:\,(j',\ell')=s_{pq}} x_q(j',\ell')} \right\}_{(j,\ell)\,:\,(j,\ell)=s_{pq}}. \tag{20}$$

The induction below, which relies on the Intersection Consistency Constraints (18), justifies assuming that the denominator in (20) is non-zero. Iterating this procedure for all of the players of T in breadth-first order defines a random strategy profile.

We prove that the output of this algorithm satisfies (19) by structural induction on T. This identity holds at the root by definition. Consider a player q with parent p in T and suppose that (19) holds for p. Fix $j \le k_q$ and $\ell \in S_q$, and let s_{pq} denote the unique strategy assignment to p and q that is consistent with the pair (j,ℓ). We conclude that

$$\begin{aligned}
\mathbf{Pr}[s \in \mathcal{S}_q(j,\ell)] &= \mathbf{Pr}[s \in \mathcal{S}_q(j,\ell) | s_{pq}] \cdot \mathbf{Pr}[s_{pq}] \\
&= \left(\frac{x_q(j,\ell)}{\sum_{(j',\ell')\,:\,(j',\ell')=s_{pq}} x_q(j',\ell')} \right) \cdot \left(\sum_{j' \le r_p, \ell' \in S_p\,:\,(j',\ell')=s_{pq}} x_p(j',\ell') \right) \qquad (21) \\
&= \left(\frac{x_q(j,\ell)}{\sum_{(j',\ell')\,:\,(j',\ell')=s_{pq}} x_q(j',\ell')} \right) \cdot \left(\sum_{j' \le k_q, \ell' \in S_q\,:\,(j',\ell')=s_{pq}} x_q(j',\ell') \right) \qquad (22) \\
&= x_q(j,\ell),
\end{aligned}$$

where (21) follows from the rounding procedure (19) (at q) and the inductive hypothesis (at p), and (22) follows from the Intersection Consistency Constraints (18). ■

7 Computing Nash Equilibria in Symmetric Games

Finally, in this section we present a polynomial-time algorithm for computing a symmetric Nash equilibrium —one in which all players employ the same mixed strategy— in succinctly represented symmetric games, provided the number of players is large relative to the number of strategies.

Theorem 7.1 *The problem of computing a symmetric Nash equilibrium in a symmetric game with n players and m strategies can be solved to arbitrary precision in time polynomial in n^m, the number of bits required to describe the utility functions, and the number of bits of precision desired.*

Proof: Let G be an n-player, m-strategy symmetric game. Nash [35] proved that G has a symmetric Nash equilibrium $x^* = (x_1^*, \ldots, x_m^*)$. We can "guess" the support of x^* (i.e., try all possibilities) in time exponential in m but independent of n — and thus polynomial in n^m. (The support of x^* is the set of strategies i for which $x_i^* > 0$.) So suppose we know the support of x^*, which without loss of generality is $\{1, 2, \ldots, j\}$ for some $1 \leq j \leq n$. Let U_ℓ denote the expected payoff to player p, if player p chooses strategy ℓ and every other player chooses a strategy at random according to the distribution x^*. Since G is symmetric, U_ℓ is a polynomial in the j variables $x_1^*, \ldots, x_j^*$ of degree $n - 1$ that is independent of the player p.

Since x^* is a Nash equilibrium, it must satisfy the equations $U_\ell = U_{\ell+1}$ for $1 \leq \ell < j$ and the inequalities $U_j \leq U_\ell$ for $\ell > j$. Conversely, every vector $(x_1, \ldots, x_j)$ with non-negative components that sum to 1 and that satisfies these equations and inequalities yields a symmetric Nash equilibrium. Finding such a vector amounts to solving $O(m)$ simultaneous equations and inequalities with degree $O(n)$ in $O(m)$ variables. It is known — see Renegar [39] and the references therein — that this problem can be solved in time polynomial in n^m, the number of bits of the numbers in the input, and in the number of bits of precision desired. ■

Since the succinct representation of a symmetric game has size $\Omega(\text{poly}(n^m))$ when $m = O(\log n / \log\log n)$, we have the following corollary of Theorem 7.1.

Corollary 7.2 *The problem of computing a Nash equilibrium of a succinctly represented n-player m-strategy symmetric game with $m = O(\log n / \log\log n)$ is in P.*

Corollary 7.2 stands in contrast to symmetric games with a constant number of players: combining recent hardness results [9, 12] with a classical reduction [7, 19] from general to symmetric games shows that it is PPAD-complete to compute a symmetric Nash equilibrium in such games.

Incidentally, a different application of the decision procedure for the first-order theory of the reals to games was developed independently by Lipton and Markakis [32].

8 Open Problems

Our work suggests a number of interesting open questions.

- Now that an ellipsoid-based algorithm for computing correlated equilibria has been discovered, we should pursue faster, more combinatorial versions. One possible approach would be to employ the techniques in [2, 28] (which generalize several combinatorial algorithms for specially structured linear programs).

- Is there an approach to finding correlated equilibria in games of exponential type, such as extensive-form games [43], congestion and network design games in which player strategies are given implicitly in terms of the $s - t$ paths in a graph, or for classes of circuit games [16]? What kinds of super-succinct representations of correlated equilibria are needed? Note that there are no standard techniques that work on linear programs that have both dimensions exponential. Daskalakis and Papadimitriou [13] explore certain problems of *doubly* exponential type defined in terms of highly regular graphs, such as the 2-dimensional grid; in such games correlated equilibria can be computed efficiently by exploiting symmetry.

- It would be very interesting to investigate whether there is a polynomial algorithm for correlated equilibria in the case of games in which the utility functions are given as a circuit [42], a generic succinct representation generalizing all of the ones we consider here. For such games computing expected utility for product distributions is a $\#P$-complete problem, and hence our approach does not apply. However, showing that, under some complexity assumptions, there is no polynomial-time correlated equilibrium scheme for such games seems a very intriguing and challenging problem.

- Our approach works for all kinds of succinct multiplayer games of polynomial type in the literature known to us. There are, however, artificial examples for which it does not seem to: A simple one is a variant of polymatrix games in which a player's utility is the *maximum* utility over all games played, as opposed to the sum. The maximum destroys, of course, linearity of expectation. Is there a correlated equilibrium scheme for this class?

- The previous two questions bring about a third, a rather novel complexity-theoretic problem: What kinds of negative complexity results can be shown about the existence of polynomial correlated equilibrium schemes (recall the definition)? By which kinds of reductions can one rule out the existence of a polynomial sampling algorithm R driven by a polynomially computable input?

- Theorem 5.10 shows that computing optimal correlated equilibria is an NP-hard problem in many important classes of succinct games. The approximability of this problem should be systematically studied.

Acknowledgments

Special thanks to Dick Karp for reminding us about some of the dynamic programming tricks in Section 4, Vangelis Markakis for notifying us about [32], Grant Schoenebeck for directing us to [19], Jiri Sgall for a suggestion that led to a big simplification in our proof of Theorem 6.3, and Bernhard von Stengel for suggesting the study of polymatrix games. We received excellent suggestions from a number of other researchers, and we thank them: Kamalika Chaudhuri, Vincent Conitzer, James Lee, Laci Lovász, Nimrod Megiddo, Satish Rao, Martha Sideri, Kunal Talwar, Éva Tardos, Boriska Toth, and Umesh Vazirani. Finally, we thank the anonymous referees from the conference versions of this work and from JACM for many helpful comments.

References

[1] E. Anshelevich, A. Dasgupta, J. Kleinberg, É. Tardos, T. Wexler, and T. Roughgarden. The price of stability for network design with fair cost allocation. In *Proceedings of the 45th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 295–304, 2004.

[2] S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: a meta algorithm and applications. Working paper, 2005.

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

[4] D. Blackwell. An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, 6(1):1–8, 1956.

[5] M. Blonski. Characterization of pure-strategy equilibria in finite anonymous games. *Journal of Mathematical Economics*, 34(2):225–233, 2000.

[6] F. Brandt, F. Fischer, and M. Holzer. Symmetries and the complexity of pure Nash equilibrium. In *Proceedings of the 24th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 4393 of *Lecture Notes in Computer Science*, pages 212–223, 2007.

[7] J. W. Brown and J. von Neumann. Solutions of games by differential equations. In H. W. Kuhn and A. W. Tucker, editors, *Contributions to the Theory Games*, volume 1, pages 73–79. Princeton University Press, 1950.

[8] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games.* Cambridge University Press, 2006.

[9] X. Chen and X. Deng. Settling the complexity of two-player Nash equilibrium. In *Proceedings of the 47th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 261–270, 2006.

[10] B. G. Chun, K. Chaudhuri, H. Wee, M. Barreno, C. H. Papadimitriou, and J. Kubiatowicz. Selfish caching in distributed systems: a game-theoretic analysis. In *Proceedings*

of the 23rd ACM Symposium on Principles of Distributed Computing (PODC), pages 21–30, 2004.

[11] V. Chvátal. *Linear Programming.* Freeman, 1983.

[12] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou. The complexity of computing a Nash equilibrium. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC)*, pages 71–78, 2006.

[13] C. Daskalakis and C. H. Papadimitriou. The complexity of games on highly regular graphs. In *Proceedings of the 13th Annual European Symposium on Algorithms (ESA)*, pages 71–82, 2005.

[14] B. C. Eaves. Polymatric games with joint constraints. *SIAM Journal on Applied Mathematics*, 24(3):418–423, 1973.

[15] A. Fabrikant, C. H. Papadimitriou, and K. Talwar. The complexity of pure Nash equilibria. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC)*, pages 604–612, 2004.

[16] L. Fortnow, R. Impagliazzo, V. Kabanets, and C. Umans. On the complexity of succinct zero-sum games. In *Proceedings of the 20th Annual IEEE Conference on Computational Complexity (CCC)*, pages 323–332, 2005.

[17] D. Foster and R. Vohra. Calibrated learning and correlated equilibrium. *Games and Economic Behavior*, 21(1-2):40–55, 1997.

[18] D. Fotakis, S. C. Kontogiannis, E. Koutsoupias, M. Mavronicolas, and P. G. Spirakis. The structure and complexity of Nash equilibria for a selfish routing game. In *Proceedings of the 29th Annual International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 2380 of *Lecture Notes in Computer Science*, pages 123–134, 2002.

[19] D. Gale, H. W. Kuhn, and A. W. Tucker. On symmetric games. In H. W. Kuhn and A. W. Tucker, editors, *Contributions to the Theory Games*, volume 1, pages 81–87. Princeton University Press, 1950.

[20] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization.* Springer, 1988. Second Edition, 1993.

[21] S. Hart and Y. Mansour. The communication complexity of uncoupled Nash equilibrium procedures. *Games and Economic Behavior*, 2008. To appear.

[22] S. Hart and A. Mas-Colell. A simple adaptive pocedure leading to correlated equilibria. *Econometrica*, 68(5):1127–1150, 2000.

[23] S. Hart and D. Schmeidler. Existence of correlated equilibria. *Mathematics of Operations Research*, 14(1):18–25, 1989.

[24] Jr. J. T. Howson. Equilibria of polymatrix games. *Management Science*, 18(5):312–318, 1972.

[25] S. Kakade, M. Kearns, J. Langford, and L. Ortiz. Correlated equilibria in graphical games. In *Proceedings of the 4th ACM Conference on Electronic Commerce*, pages 42–47, 2003.

[26] M. Kearns, M. L. Littman, and S. Singh. Graphical models for game theory. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 253–260, 2001.

[27] L. G. Khachiyan. A polynomial algorithm in linear programming. *Soviet Mathematics Doklady*, 20(1):191–194, 1979.

[28] R. Khandekar. *Lagrangian Relaxation based Algorithms for Convex Programming Problems.* PhD thesis, IIT Delhi, 2004.

[29] J. Kleinberg and É. Tardos. *Algorithm Design.* Addison-Wesley, 2005.

[30] J. K. Lenstra, D. B. Shmoys, and É. Tardos. Approximation algorithms for scheduling unrelated parallel machines. *Mathematical Programming*, 46(3):259–271, 1990.

[31] K. Leyton-Brown and M. Tennenholtz. Local-effect games. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 772–780, 2003.

[32] R. J. Lipton and E. Markakis. Nash equilibria via polynomial equations. In *Proceedings of the Sixth Conference on Latin American Theoretical Informatics (LATIN)*, pages 413–422, 2004.

[33] I. Milchtaich. Congestion games with player-specific payoff functions. *Games and Economic Behavior*, 13(1):111–124, 1996.

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

[35] J. F. Nash, Jr. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, 1951.

[36] R. F. Nau and K. F. McCardle. Coherent behavior in noncooperative games. *Journal of Economic Theory*, 50(2):424–444, 1990.

[37] C. H. Papadimitriou. The complexity of finding Nash equilibria. In N. Nisan, T. Roughgarden, É. Tardos, and V. V. Vazirani, editors, *Algorithmic Game Theory*, chapter 2, pages 29–51. Cambridge, 2007.

[38] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity.* Prentice-Hall, 1982. Dover Edition, 1998.

[39] J. Renegar. On the computational complexity and geometry of the first-order theory of the reals, parts I–III. *Journal of Symbolic Computation*, 13(3):255–299, 301–327, 329–352, 1992.

[40] R. W. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2(1):65–67, 1973.

[41] T. Roughgarden and É. Tardos. Bounding the inefficiency of equilibria in nonatomic congestion games. *Games and Economic Behavior*, 49(2):389–403, 2004.

[42] G. Schoenebeck and S. P. Vadhan. The computational complexity of Nash equilibria in concisely represented games. In *Proceedings of the 7th ACM Conference on Electronic Commerce (EC)*, pages 270–279, 2006.

[43] B. von Stengel. Computational complexity of correlated equilibria for extensive games. Technical Report LSE-CDAM-2001-03, Centre for Discrete and Applicable Mathematics, London School of Economics, 2001.

[44] E. B. Yanovskaya. Equilibrium situations in multi-matrix games. *Litovskii Matematicheskii Sbornik*, 8:381–384, 1968. In Russian.