The Game World Is Flat: The Complexity of Nash Equilibria in Succinct Games

Constantinos Daskalakis*, Alex Fabrikant**, and Christos H. Papadimitriou***

UC Berkeley, Computer Science Division
costis@cs.berkeley.edu
alexf@cs.berkeley.edu
christos@cs.berkeley.edu

Abstract. A recent sequence of results established that computing Nash equilibria in normal form games is a PPAD-complete problem even in the case of two players [11,6,4]. By extending these techniques we prove a general theorem, showing that, for a far more general class of families of succinctly representable multiplayer games, the Nash equilibrium problem can also be reduced to the two-player case. In view of empirically successful algorithms available for this problem, this is in essence a positive result — even though, due to the complexity of the reductions, it is of no immediate practical significance. We further extend this conclusion to extensive form games and network congestion games, two classes which do not fall into the same succinct representation framework, and for which no positive algorithmic result had been known.

1 Introduction

Nash proved in 1951 that every game has a mixed Nash equilibrium [15]. However, the complexity of the computational problem of finding such an equilibrium had remained open for more than half century, attacked with increased intensity over the past decades. This question was resolved recently, when it was established that the problem is PPAD-complete [6] (the appropriate complexity level, defined in [18]) and thus presumably intractable, for the case of 4 players; this was subsequently improved to three players [5,3] and, most remarkably, two players [4].

In particular, the combined results of [11,6,4] establish that the general Nash equilibrium problem for normal form games (the standard and most explicit representation) and for graphical agames (an important succinct representation, see the next paragraph) can all be reduced to 2-player games. 2-player games in turn can be solved by several techniques such as the Lemke-Howson algorithm [14,20], a simplex-like technique that is known empirically to behave well even though exponential counterexamples do exist [19]. *In this paper we extend these results to essentially all known kinds of succinct representations of games, as well as to more sophisticated concepts of equilibrium.*

Besides this significant increase in our understanding of complexity issues, computational considerations also led to much interest in *succinct representations of games.*

* Supported by NSF ITR Grant CCR-0121555.
** Supported by the Fannie and John Hertz Foundation.
*** Supported by NSF ITR CCR-0121555 grant and a Microsoft Research grant.

M. Bugliesi et al. (Eds.): ICALP 2006, Part I, LNCS 4051, pp. 513–524, 2006.

Computer scientists became interested in games because they help model networks and auctions; thus we should mainly focus on games with many players. However, multi-player games in normal form require in order to be described an amount of data that is exponential in the number of players. When the number of players is large, the resulting computational problems are hardly legitimate, and complexity issues are hopelessly distorted. This has led the community to consider broad classes of *succinctly representable games*, some of which had been studied by traditional game theory for decades, while others (like the graphical games [13]) were invented by computer scientists spurred by the motivations outline above. (We formally define succinct games in the next section, but also deal in this paper with two cases, network congestion games and extensive form games, that do not fit within this definition).

The first general positive algorithmic result for succinct games was obtained only recently [17]: a polynomial-time algorithm for finding a correlated equilibrium (an important generalization of the Nash equilibrium due to Aumann [1]). The main result in [17] states that a family of succinct games has a polynomial-time algorithm for correlated equilibria provided that there is a polynomial time oracle which, given a strategy profile, computes the expected utility of each player.

In this paper, using completely different techniques inspired from [11], we show a general result (Theorem 2) that is remarkably parallel to that of [17]: The Nash equilibrium problem of a family of succinct games can be reduced to the 2-player case provided that a (slightly constrained) *polynomial-length straight-line arithmetic program* exists which computes, again, the expected utility of a given strategy profile (notice the extra algebraic requirement here, necessitated by the algebraic nature of our techniques). We proceed to point out that *for all major known* families of succinct games such a straight-line program exists (Corollary 1).

We also extend these techniques to two other game classes, Network congestion games [7] and extensive form games, which do not fit into our succinctness framework, because the number of strategies is exponential in the input, and for which the result of [17] does not apply, Theorems 3 and 4, respectively).

2 Definitions and Background

In a *game in normal form* we have $r \geq 2$ players (and for each player $p \leq r$ a finite set S_p of pure strategies. We denote the Cartesian product of the S_p's by S (the set of *pure strategy profiles*) and the Cartesian product of the pure strategy sets of players other than p by S_{-p}. Finally, for each $p \leq r$ and $s \in S$ we have a *payoff* u_s^p.

A *mixed strategy* for player p is a distribution on S_p, that is, $|S_p|$ nonnegative real numbers adding to 1. Call a set of r mixed strategies $x_j^p, p = 1, \ldots, r, j \in S_p$ a *Nash equilibrium* if, for each p, its expected payoff, $\sum_{s \in S} u_s^p \prod_{q=1}^r x_{s_q}^q$ is maximized over all mixed strategies of p. That is, a Nash equilibrium is a set of mixed strategies from which no player has an incentive to deviate. For $s \in S_{-p}$, let $x_s = \prod_{q \neq p} x_{s_q}^q$. It is well-known (see, e.g., [16]) that the following is an equivalent condition for a set of mixed strategies to be a Nash equilibrium:

$$\forall p, j \quad \sum_{s \in S_{-p}} u_{js}^p x_s > \sum_{s \in S_{-p}} u_{j's}^p x_s \Longrightarrow x_{j'}^p = 0. \tag{1}$$

Also, a set of mixed strategies is an *ε-Nash equilibrium* for some $\varepsilon > 0$ if the following holds:

$$\sum_{s \in S_{-p}} u^p_{js} x_s > \sum_{s \in S_{-p}} u^p_{j's} x_s + \varepsilon \Longrightarrow x^p_{j'} = 0. \quad (2)$$

We next define the complexity class PPAD. An *FNP search problem* $\mathcal{P}$ is a set of inputs $I_\mathcal{P} \subseteq \Sigma^*$ such that for each $x \in I_\mathcal{P}$ there is an associated set of solutions $\mathcal{P}_x \subseteq \Sigma^{|x|^k}$ for some integer k, such that for each $x \in I_\mathcal{P}$ and $y \in \Sigma^{|x|^k}$ whether $y \in \mathcal{P}_x$ is decidable in polynomial time (notice that this is precisely NP with an added emphasis on finding a witness). For example, r-NASH is the search problem $\mathcal{P}$ in which each $x \in I_\mathcal{P}$ is an r-player game in normal form together with a binary integer A (the *accuracy specification*), and $\mathcal{P}_x$ is the set of $\frac{1}{A}$-Nash equilibria of the game.

A search problem is *total* if $\mathcal{P}_x \neq \emptyset$ for all $x \in I_\mathcal{P}$. For example, Nash's 1951 theorem [15] implies that r-NASH is total. The set of all total FNP search problems is denoted TFNP. TFNP seems to have no generic complete problem, and so we study its subclasses: PLS [12], PPP, PPA and PPAD [18]. In particular, PPAD is the class of all total search problems reducible to the following:

END OF THE LINE: Given two circuits S and P with n input bits and n output bits, such that $P(0^n) = 0^n \neq S(0^n)$, find an input $x \in \{0,1\}^n$ such that $P(S(x)) \neq x$ or $S(P(x)) \neq x \neq 0^n$.

Intuitively, END OF THE LINE creates a directed graph with vertex set $\{0,1\}^n$ and an edge from x to y whenever $P(y) = x$ and $S(x) = y$ (S and P stand for "successor candidate" and "predecessor candidate"). This graph has indegree and outdegree at most one, and at least one source, namely 0^n, so it must have a sink. We seek either a sink, or a source other than 0^n. Thus, PPAD is the class of all total functions whose totality is proven via the simple combinatorial argument outlined above.

A polynomially computable function f is a *polynomial-time reduction* from total search problem $\mathcal{P}$ to total search problem $\mathcal{Q}$ if, for every input x of $\mathcal{P}$, $f(x)$ is an input of $\mathcal{Q}$, and furthermore there is another polynomially computable function g such that for every $y \in \mathcal{Q}_{f(x)}$, $g(y) \in \mathcal{P}_x$. A search problem $\mathcal{P}$ in PPAD is called *PPAD-complete* if all problems in PPAD reduce to it. Obviously, END OF THE LINE is PPAD-complete; we now know that 2-NASH is PPAD-complete [6,4].

In this paper we are interested in *succinct games*. A succinct game [17] $G = (I, T, U)$ is a set of inputs $I \in \mathrm{P}$, and two polynomial algorithms T and U. For each $z \in I$, $T(z)$ returns a *type*, that is, the number of players $r \leq |z|$ and an r-tuple $(t_1, \ldots, t_r)$ where $|S_p| = t_p$. We say that G is of *polynomial type* if all t_p's are bounded by a polynomial in $|z|$. In this paper we are interested in games of both polynomial (Section 3) and non-polynomial type (Sections 5 and 4). Finally, for any r-tuple of positive integers $s = (s_1, \ldots, s_r)$, where $s_p \leq t_p$, and $p \leq r$, $U(z, p, s)$ returns an integer standing for the utility u^p_s. The game in normal form thus encoded by $z \in I$ is denoted by $G(z)$.

Examples of succinct games (due to space constraints we omit the formal definitions, see [17] for more details) are:

- *graphical games* [13], where players are nodes on a graph, and the utility of a player depends only on the strategies of the players in its neighborhood.

- *congestion games* [7], where strategies are sets of *resources*, and the utility of a player is the sum of the delays of the resources in the set it chose, where the delay is a resource-specific function of the number of players who chose this resource.
- *network congestion games*, where the strategies of each player are given implicitly as paths from a source to a sink in a graph; since the number of strategies is potentially exponential, this representation is not of polynomial type; we treat network congestion games in Section 4.
- *multimatrix games* where each player plays a different 2-person game with each other player, and the utilities are added.
- *semi-anonymous games* (a generalization of symmetric games not considered in [17]) in which all players have the same set of strategies, and each player has a utility function that depends solely on the *number* of other players who choose each strategy (and not the identities of these players).
- several other classes such as *local effect games, scheduling games, hypergraphical games, network design games, facility location games*, etc., as catalogued in [17].

Our main result, shown in the next section, implies that the problem finding a Nash equilibrium in all of these classes of games can be reduced to 2-player games (equivalently, belongs to the class PPAD).

Lastly, we define a *bounded (division-free) straight-line program* to be an arithmetic binary circuit with nodes performing addition, subtraction, or multiplication on their inputs, or evaluating to pre-set constants, with the additional constraint that the values of all the nodes remain in $[0, 1]$. This restriction is not severe, as it can be shown that an arithmetic circuit of size n with intermediate nodes bounded in absolute value by $2^{\mathrm{poly}(n)}$ can be transformed in polynomial time to fit the above constraint (with the output scaled down by a factor dependent only on the bound).

3 The Main Result

Given a succinct game, the following problem, called EXPECTED UTILITY, is of interest: Given a mixed strategy profile $x^1, \ldots, x^r$, compute the expected utility of player p. Notice that the result sought is a polynomial in the input variables. It was shown in [17] that a polynomial-time algorithm for EXPECTED UTILITY (for succinct games of polynomial type) implies a polynomial-time algorithm for computing correlated equilibria for the succinct game. Here we show a result of a similar flavor.

3.1 Mapping Succinct Games to Graphical Games

Theorem 1. *If for a succinct game G of polynomial type there is a bounded division-free straight-line program of polynomial length for computing* EXPECTED UTILITY*, then G can be mapped in polynomial time to a graphical game $\mathcal{G}$ so that there is a polynomially computable surjective mapping from the set of Nash equilibria of $\mathcal{G}$ to the set of Nash equilibria of G.*

Proof. Let G be a succinct game for which there is a bounded straight-line program for computing EXPECTED UTILITY. In time polynomial in $|G|$, we will construct a

graphical game $\mathcal{G}$ so that the statement of the theorem holds. Suppose that G has r players, $1, \ldots, r$, with strategy sets $S_p = \{1, \ldots, t_p\}, \forall p \leq r$. The players of game $\mathcal{G}$, which we shall call *nodes* in the following discussion to distinguish them from the players of G, will have two strategies each, strategy 0 and strategy 1. We will interpret the probability with which a node x of $\mathcal{G}$ chooses strategy 1 as a real number in $[0, 1]$, which we will denote, for convenience, by the same symbol x that we use for the node.

Below we describe the nodes of $\mathcal{G}$ as well as the role of every node in the construction. We will describe $\mathcal{G}$ as a directed network with vertices representing the nodes (players) of $\mathcal{G}$ and directed edges denoting directed flow of information as in [11,6].

1. For every player $p = 1, \ldots, r$ of G and for every pure strategy $j \in S_p$, game $\mathcal{G}$ has a node x_j^p. Value x_j^p should be interpreted as the probability with which player p plays strategy j; in fact, we will establish later that, given a Nash equilibrium of $\mathcal{G}$, this interpretation yields a Nash equilibrium of G. As we will see in Item 4 below, our construction will ensure that, at any Nash equilibrium, $\sum_{j=1}^{t_p} x_j^p = 1, \forall p \leq r$. Therefore, it is legitimate to interpret the set of values $\{x_j^p\}_j$ as a mixed strategy for player p in G.
2. For every player $p = 1, \ldots, r$ of G and for every pure strategy $j \in S_p$, game $\mathcal{G}$ has nodes U_j^p and $U_{\leq j}^p$. The construction of $\mathcal{G}$ will ensure that, at a Nash equilibrium, value U_j^p equals the utility of player p for playing pure strategy j if every other player $q \neq p$ plays the mixed strategy specified by the distribution $\{x_j^q\}_j$. Also, the construction will ensure that $U_{\leq j}^p = \max_{j' \leq j} U_{j'}^p$. Without loss of generality, we assume that all utilities in G are scaled down to lie in $[0, 1]$.
3. For every node of type U_j^p there is a set of nodes in $\mathcal{G}$ that simulate the intermediate variables used by the straight-line program computing the expected utility of player p for playing pure strategy j when the other players play according to the mixed strategies specified by $\{\{x_j^q\}_j\}_{q \neq p}$. This is possible due to our constraint on the straight-line program.
4. For every player p of G, there is a set of nodes Ψ_p defining a component $\mathcal{G}_p$ of $\mathcal{G}$ whose purpose is to guarantee the following at any Nash equilibrium of $\mathcal{G}$:
   (a) $\sum_{j=1}^{t_p} x_j^p = 1$
   (b) $U_j^p > U_{j'}^p \Longrightarrow x_{j'}^p = 0$
   The structure and the functionality of $\mathcal{G}_p$ are described in section 3 of [11], so its details will be omitted here. Note that the nodes of set Ψ_p interact only with the nodes $\{U_j^p\}_j$, $\{U_{\leq j}^p\}_j$ and $\{x_j^p\}_j$. The nodes of types U_j^p and $U_{\leq j}^p$ are not affected by the nodes in Ψ_p and should be interpreted as "input" to $\mathcal{G}_p$, whereas the nodes of type x_j^p are only affected by $\mathcal{G}_p$ and not by the rest of the game and are the "output" of $\mathcal{G}_p$. The construction of $\mathcal{G}_p$ ensures that they satisfy Properties 4a and 4b.

Having borrowed the construction of the components $\mathcal{G}_p, p \leq r$, from [11], the only components of $\mathcal{G}$ that remain to be specified are those that compute expected utilities. With the bound on intermediate variable values, the construction of these components can be easily done using the games $\mathcal{G}_=$, $\mathcal{G}_\zeta$, $\mathcal{G}_+$, $\mathcal{G}_-$, $\mathcal{G}_*$ for assignment, assignment of a constant ζ, addition, subtraction and multiplication that were defined in [11]. Finally, the components of $\mathcal{G}$ that give values to nodes of type $U_{\leq j}^p$ can be easily constructed using games $\mathcal{G}_{\max}$ from [11]. It remains to argue that, given a Nash equilibrium of $\mathcal{G}$, we

can find in polynomial time a Nash equilibrium of G and moreover that this mapping is onto. The first claim follows from the following lemma and the second is easy to verify.

Lemma 1. *At a Nash equilibrium of game $\mathcal{G}$, values $\{\{x_j^p\}_j\}_p$ constitute a Nash equilibrium of game G.*

Proof. From the correctness of games $\mathcal{G}_p, p \leq r$, it follows that, at any Nash equilibrium of game $\mathcal{G}$, $\sum_{j=1}^{t_p} x_j^p = 1, \forall p$. Moreover, from the correctness of games $\mathcal{G}_=$, $\mathcal{G}_\zeta$, $\mathcal{G}_+$, $\mathcal{G}_-$, $\mathcal{G}_*$, it follows that, at any Nash equilibrium of game $\mathcal{G}$, U_j^p will be equal to the utility of player p for playing pure strategy j when every other player $q \neq p$ plays as specified by the values $\{x_j^q\}_j$. From the correctness of $\mathcal{G}_{\max}$ it follows that, at any Nash equilibrium of game $\mathcal{G}$, $U_{\leq j}^p = \max_{j' \leq j} U_{j'}^p, \forall p, j$. Finally, from the correctness of games $\mathcal{G}_p, p \leq r$, it follows that, at any Nash equilibrium of game $\mathcal{G}$, for every $p \leq r$ and for every $j, j' \in S_p, j \neq j'$: $U_j^p > U_{j'}^p \Longrightarrow x_{j'}^p = 0$. By combining the above it follows that $\{\{x_j^p\}_j\}_p$ constitute a Nash equilibrium of game G. □

3.2 Succinct Games in PPAD

We now explore how the mapping described in Theorem 1 can be used in deriving complexity results for the problem of computing a Nash equilibrium in succinct games.

Theorem 2. *If for a succinct game G of polynomial type there is a bounded division-free straight-line program of polynomial length for computing* EXPECTED UTILITY*, then the problem of computing a Nash equilibrium in the succinct game polynomially reduces to the problem of computing a Nash equilibrium of a 2-player game.*

Proof. We will describe a reduction from the problem of computing a Nash equilibrium in a succinct game to the problem of computing a Nash equilibrium in a graphical game. This is sufficient since the latter can be reduced to the problem of computing a Nash equilibrium in a 2-player game [6,4]. Note that the reduction sought does not follow trivially from Theorem 1; the mapping there makes sure that the exact equilibrium points of the graphical game can be efficiently mapped to exact equilibrium points of the succinct game. Here we seek something stronger; we want every approximate Nash equilibrium of the former to be efficiently mapped to an approximate Nash equilibrium of the latter. This requirement turns out to be more delicate than the previous one.

Formally, let G be a succinct game for which there is a straight line program for computing EXPECTED UTILITY and let ε be an accuracy specification. Suppose that G has r players, $1, \ldots, r$, with strategy sets $S_p = \{1, \ldots, t_p\}, \forall p \leq r$. In time polynomial in $|G| + |1/\varepsilon|$, we will specify a graphical game $\mathcal{G}$ and an accuracy ε' with the property that, given an ε'-Nash equilibrium of $\mathcal{G}$, one can recover in polynomial time an ε-Nash equilibrium of G. In our reduction, the graphical game $\mathcal{G}$ will be the same as the one described in the proof of Theorem 1, while the accuracy specification will be of the form $\varepsilon' = \varepsilon/2^{p(n)}$, where $p(n)$ is a polynomial in $n = |G|$ that will be be specified later. Using the same notation for the nodes of game $\mathcal{G}$ as we did in Theorem 1, let us consider if the equivalent of Lemma 1 holds for approximate Nash equilibria.

Observation 1. *For any $\varepsilon' > 0$, there exist ε'-Nash equilibria of game $\mathcal{G}$ in which the values $\{\{x_j^p\}_j\}_p$* **do not** *constitute an ε-Nash equilibrium of game G.*

Proof. A careful analysis of the mechanics of gadgets $\mathcal{G}_p$, $p \leq r$, shows that property (2) which is the defining property of an approximate Nash equilibrium is not guaranteed to hold. In fact, there are ε'-equilibria of $\mathcal{G}$ in which $\sum_{s \in S_{-p}} u^p_{js} x_s > \sum_{s \in S_{-p}} u^p_{j's} x_s + \varepsilon'$ for some $p \leq r$, j and j', and, yet, $x^p_{j'}$ is any value in $[0, t_p \cdot \varepsilon']$. The details are omitted. □

Moreover, the values $\{x^p_j\}_j$ do not necessarily constitute a distribution as specified by the following observation.

Observation 2. *For any $\varepsilon' > 0$, for any $p \leq r$, at an ε'-Nash equilibrium of game $\mathcal{G}$, $\sum_j x^p_j$ is not necessarily equal to* 1.

Proof. Again by carefully analyzing the behavior of gadgets $\mathcal{G}_p$, $p \leq r$, at an ε'-Nash equilibrium of game $\mathcal{G}$, it can be shown that there are equilibria in which $\sum_j x^p_j$ can be any value in $1 \pm 2t_p\varepsilon'$. The details are omitted. □

Therefore, the extraction of an ε-Nash equilibrium of game G from an ε'-Nash equilibrium of game $\mathcal{G}$ cannot be done by just interpreting the values $\{x^p_j\}$ as the probability distribution of player p. What we show next is that, for the right choice of ε', a *trim and renormalize* strategy succeeds in deriving an ε-Nash equilibrium of game G from an ε'-Nash equilibrium of game $\mathcal{G}$. For any $p \leq r$, suppose that $\{\hat{x}^p_j\}_j$ are the values derived from $\{x^p_j\}_j$ as follows: make all values smaller than $t_p\varepsilon'$ equal to zero (trim) and renormalize the resulting values so that $\sum_j \hat{x}^p_j = 1$. The argument will rely on the tightness of the bounds mentioned above, also obtained from the gadgets' properties:

Observation 3. *In an ε'-Nash equilibrium of game $\mathcal{G}$, $|\sum_j x^p_j - 1| \leq 2t_p\varepsilon'$, and, if $\sum_{s \in S_{-p}} u^p_{js} x_s > \sum_{s \in S_{-p}} u^p_{j's} x_s + \varepsilon'$, then $x^p_{j'} \in [0, t_p \cdot \varepsilon']$.*

Lemma 2. *There exists a polynomial $p(n)$ such that, if $\varepsilon' = \varepsilon/2^{p(n)}$, then, at an ε'-Nash equilibrium of game $\mathcal{G}$, the values $\{\{\hat{x}^p_j\}_j\}_p$ constitute an ε-Nash equilibrium of game G.*

Proof. We will denote by $\mathcal{U}^p_j(\cdot)$ the function defined by the straight-line program that computes the utility of player p for choosing pure strategy j. We need to compare the values $\mathcal{U}^p_j(\hat{x})$ with the values of the nodes U^p_j of the graphical game $\mathcal{G}$ at an ε'-Nash equilibrium. For convenience, let $\hat{U}^p_j \triangleq \mathcal{U}^p_j(\hat{x})$ be the expected utility of player p for playing pure strategy j when the other players play according to $\{\{\hat{x}^q_j\}_j\}_{q \neq p}$. Our ultimate goal is to show that, at an ε'-Nash equilibrium of game $\mathcal{G}$, for all $p \leq r, j \leq t_p$

$$\hat{U}^p_j > \hat{U}^p_{j'} + \varepsilon \Longrightarrow \hat{x}^p_{j'} = 0 \tag{3}$$

Let us take $c(n)$ to be the polynomial bound on $2t_p$. Using Observation 3, we get that, for all p, j,

$$\begin{aligned} & \hat{x}^p_j(1 - c(n)\varepsilon') \leq x^p_j \leq \max\{c(n)\varepsilon', \hat{x}^p_j(1 + c(n)\varepsilon')\} \\ \Rightarrow \quad & \hat{x}^p_j - c(n)\varepsilon' \leq x^p_j \leq \hat{x}^p_j + c(n)\varepsilon' \end{aligned} \tag{4}$$

To carry on the analysis, note that, although $\hat{U}_j^p$ is the output of function $\mathcal{U}_j^p(\cdot)$ on input $\{\hat{x}_j^p\}_{j,p}$, U_j^p is not the correct output of $\mathcal{U}_j^p(\cdot)$ on input $\{x_j^p\}_{j,p}$. This is, because, at an ε'-Nash equilibrium of game $\mathcal{G}$, the games that simulate the gates of the arithmetic circuit introduce an additive error of absolute value up to ε' per operation. So, to compare U_j^p with $\hat{U}_j^p$, we shall compare the "erroneous" evaluation of the arithmetical circuit on input $\{x_j^p\}_{j,p}$ carried inside $\mathcal{G}$ against the ideal evaluation of the circuit on input $\{\hat{x}_j^p\}_{j,p}$. Let us assign a nonnegative "level" to every wire of the arithmetical circuit in the natural way: the wires to which the input is provided are at level 0 and a wire out of a gate is at level one plus the maximum level of the gate's input wires. Since the arithmetical circuits that compute expected utilities are assumed to be of polynomial length the maximum level that a wire can be assigned to is $q(n)$, $q(\cdot)$ being some polynomial. The "erroneous" and the "ideal" evaluations of the circuit on inputs $\{x_j^p\}_{j,p}$ and $\{\hat{x}_j^p\}_{j,p}$ respectively satisfy the following property which can be shown by induction:

Lemma 3. *Let v, $\hat{v}$ be the values of a wire at level i of the circuit in the erroneous and the ideal evaluation respectively. Then*

$$\hat{v} - g(i)\varepsilon' \leq v \leq \hat{v} + g(i)\varepsilon'$$

where $g(i) = 3^i \cdot (c(n) + \frac{1}{2}) - \frac{1}{2}$.

By this lemma, the outputs of the two evaluations will satisfy

$$\hat{U}_j^p - (2^{q(n)} \cdot (c(n)+1) - 1)\varepsilon' \leq U_j^p \leq \hat{U}_j^p + (2^{q(n)} \cdot (c(n)+1) - 1)\varepsilon'$$

Thus, setting $\varepsilon' = \frac{\varepsilon}{8c(n)3^{q(n)}}$ yields $|U_j^p - \hat{U}_j^p| \leq \varepsilon/4$. After applying the same argument to $U_{j'}^p$ and $\hat{U}_{j'}^p$, we have that $\hat{U}_j^p > \hat{U}_{j'}^p + \varepsilon$ implies $U_j^p + \varepsilon/4 \geq \hat{U}_j^p > \hat{U}_{j'}^p + \varepsilon \geq U_{j'}^p + 3\varepsilon/4$, and thus $U_j^p > U_{j'}^p + \varepsilon/2 > U_{j'}^p + \varepsilon'$. Then, from Observation 3, it follows that $x_{j'}^p < t_p\varepsilon'$ and, from the definition of our trimming process, that $\hat{x}_{j'}^p = 0$. So (3) is satisfied, therefore making $\{\{\hat{x}_j^p\}_j\}_p$ an ε-Nash equilibrium. □

In Section 3.4 we point out that the EXPECTED UTILITY problem in typical succinct games of polynomial type is very hard. However, in all well known succinct games in the literature, it turns out that there is a straight-line program of polynomial length that computes EXPECTED UTILITY:

Corollary 1. *The problem of computing a Nash equilibrium in the following families of succinct games can be polynomially reduced to the same problem for 2-player games: graphical games, congestion games, multimatrix games, semi-anonymous games, local effect games, scheduling games, hypergraphical games, network design games, and facility location games.*

Proof. It turns out that, for all these families, there is indeed a straight-line program as specified in Theorem 2. For graphical games, for example, the program computes explicitly the utility expectation of a player with respect to its neighbors; the other mixed strategies do not matter. For multimatrix games, the program computes one quadratic

form per constituent game, and adds the expectations (by linearity). For hypergraphical games, the program combines the previous two ideas. For the remaining kinds, the program combines results of several instances of the following problem (and possibly the two previous ideas, linearity of expectation and explicit expectation calculation): Given n Bernoulli variables $x_1, \ldots, x_n$ with $\Pr[x_i = 1] = p_i$, calculate $q_j = \Pr[\sum_{i=1}^{n} x_i = j]$ for $j = 0, \ldots, n$. This can be done by dynamic programming, letting $q_j^k = \Pr[\sum_{i=1}^{k} x_i = j]$ (and omitting initializations): $q_{j+1}^k = (1 - p_i)q_j^{k-1} + p_i q_{j-1}^{k-1}$, obviously a polynomial division-free straight-line program. □

3.3 An Alternative Proof

We had been looking for some time for an alternative proof of this result, not relying on the machinery of [11]. This proof would start by reducing the Nash equilibrium problem to Brouwer by the reduction of [10]. The Brouwer function in [10] maps each mixed strategy profile $x = (x_1, \ldots, x_n)$ to another $(y_1, \ldots, y_n)$, where $y_i = \arg\max (E_{(x_{-i}, y_i)}[U_i] - ||y_i - x_i||^2)$. That is, y_i optimizes a trade-off between utility and distance from x_i. It should be possible, by symbolic differentiation of the straight-line program, to approximate this optimum and thus the Brouwer function. There are, though, difficulties in proceeding, because the next step (reduction to Sperner's Lemma) seems to require precision incompatible with guarantees obtained this way.

3.4 Intractability

Let us briefly explore the limits of the upper bound in this section.

Proposition 1. *There are succinct games of polynomial type for which* EXPECTED UTILITY *is #P-hard.*

Proof. Consider the case in which each player has two strategies, `true` and `false`, and the utility of player 1 is 1 if the chosen strategies satisfy a given Boolean formula. Then the expected utility, when all players play each strategy with probability $\frac{1}{2}$ is the number of satisfying truth assignments divided by 2^n, a #P-hard problem. □

Thus, the sufficient condition of our Theorem is nontrivial, and there are games of polynomial type that do not satisfy it. *Are there games of polynomial type for which computing Nash equilibria is intractable beyond PPAD?* This is an important open question. Naturally, computing a Nash equilibrium of a general succinct game is EXP-hard (recall that it is so even for 2-person zero-sum games [8,9], and the nonzero version can be easily seen to be complete for the exponential counterpart of PPAD).

Finally, it is interesting to ask whether our sufficient condition (polynomial computability of EXPECTED UTILITY by a bounded division-free straight-line program) is strictly weaker than the condition in [17] for correlated equilibria (polynomial computability of EXPECTED UTILITY by Turing machines). It turns out[1] that it is, unless $\oplus$P is in nonuniform polynomial time [2]. Determining the precise complexity nature of this condition is another interesting open problem.

[1] Many thanks to Peter Bürgisser for pointing this out to us.

4 Network Congestion Games

A network congestion game [7] is specified by a network with delay functions, that is, a directed graph (V, E) with a pair of nodes (a_p, b_p) for each player p, and also, for each edge $e \in E$, a *delay function* d_e mapping $[n]$ to the positive integers; for each possible number of players "using" edge e, d_e assigns a delay. The set of strategies for player p is the set of all paths from a_p to b_p. Finally, the payoffs are determined as follows: If $s = (s_1, \ldots, s_n)$ is a pure strategy profile, define $c_e(s) = |\{p : e \in s_p\}|$ (here we consider paths as sets of edges); then the utility of player p under s is simply $-\sum_{e \in s_p} d_e(c_e(s))$, the negation of the total delay on the edges in p's strategy. It was shown in [7] that a *pure* Nash equilibrium of a network congestion game (known to always exist) can be found in polynomial time when the game is *symmetric* ($a_p = a_1$ and $b_p = b_1$ for all p), and PLS-complete in the general case. There is no known polynomial-time algorithm for finding Nash equilibria (or *any* kind of equilibria, such as correlated [17]) in general network congestion games. We prove:

Theorem 3. *The problem of computing a Nash equilibrium of a network congestion game polynomially reduces to the problem of computing a Nash equilibrium of a 2-player game.*

Proof. (Sketch.) We will map a network congestion game to a graphical game $\mathcal{G}$. To finish the proof one needs to use techniques parallel to Section 3.2. To simulate network congestion games by graphical games we use a nonstandard representation of mixed strategy: We consider a mixed strategy for player p to be a *unit flow* from a_p to b_p, that is, an assignment of nonnegative values $f_p(e)$ to the edges of the network such that all nodes are balanced except for a_p who has a deficit of 1 and b_p who has a gain of 1. Intuitively, $f_p(e)$ corresponds to the sum of the probabilities of all paths that use e.

It turns out that such flow can be set up in the simulating graphical game by a gadget similar to the one that sets up the mixed strategy of each player. In particular, for every player p and for every edge e of the network there will be a player in the graphical game whose value will represent $f_p(e)$. Moreover, for every node $v \neq a_p, b_p$ of the network, there will be a player S_v^p in the graphical game whose value will be equal to the sum of the flows of player p on the edges entering node v; there will also be a gadget $\mathcal{G}_v^p$ similar to the one used in proof of Theorem 1, whose purpose will be to distribute the flow of player p entering v, i.e. value S_v^p, to the edges leaving node v, therefore guaranteeing that Kirchhoff's first law holds. The distribution of the value S_v^p on the edges leaving v will be determined by finding the net delays between their endpoints and node b_p as specified by the next paragraphs. Finally, note that the gadgets for nodes a_p and b_p are similar but will inject a gain of 1 at a_p and a deficit of 1 at b_p. Some scaling will be needed to make sure that all computed values are in $[0, 1]$.

The rest of the construction is based on the following Lemma, whose simple proof we omit. Fix a player p and a set of unit flows f_q for the other players. These induce an expected delay on each edge e, $\mathrm{E}[d_e(c_e(k))]$ where k is 1 (for player p) plus the sum of $n - 1$ variables that are 1 with probability $f_q(e)$ and else 0. Call this quantity $D_p(e)$.

Lemma 4. *A set of unit flows $f_p(e), p = 1, \ldots, n$ is an ε-Nash equilibrium if and only if $f_p(e) > 0$ implies that e lies on a path whose length (defined as net delay under D_p), is at most ε above the length of the shortest path from a_p to b_p.*

We shall show that these conditions can be calculated by a straight-line program in polynomial time; this implies the Theorem. This is done as follows: First we compute the distances $D_p(e)$ for all edges and players by dynamic programming, as in the proof of Corollary 1. Then, for each player p and edge (u, v) we calculate the shortest path distances, under D_p, (a) from a_p to b_p; (b) from a_p to u and (c) from v to b_p. This is done by the Bellman-Ford algorithm, which is a straight line program with the additional use of the min operator (see [11] for gadget). The condition then requires that the sum of the latter two and $D_p(u, v)$ be at most the former plus ε. This completes the proof. □

5 Extensive form Games

An r-player *extensive form game* (see, e.g., [16]) is represented by a game tree with each non-leaf vertex v assigned to a player $p(v)$, who "plays" by choosing one of the outgoing labeled edges, and with a vector of payoffs u_x^p at each leaf x (let X be the set of leaves). All edges have labels, with the constraint that $l(v, v') \neq l(v, v'')$. The vertex set is partitioned into *information sets* $I \in \mathcal{I}$, with all $v \in I$ owned by the same player $p(I)$, and having identical sets of outgoing edge labels L_I. We also define $\mathcal{I}_p = \{I \in \mathcal{I} | p(I) = p\}$. Information sets represent a player's knowledge of the game state. A *behavioral strategy* σ^p for player p is an assignment of distributions $\{\sigma_j^{p,I}\}_{j \in L_I}$ over the outgoing edge labels of each $I \in \mathcal{I}_p$. A behavioral strategy profile $\sigma = (\sigma^1, \ldots, \sigma^r)$ induces a distribution over the leaves of the game tree, and hence expected utilities. A *behavioral Nash equilibrium* is the natural equivalent of the normal form's mixed Nash equilibrium: a σ such that no player p can change σ^p and increase his expected payoff.

Theorem 4. *The problem of computing a behavioral Nash equilibrium (and, in fact, a subgame perfect equilibrium [16]) in an extensive form game Γ is polynomially reducible to computing a mixed Nash equilibrium in a 2-player normal form game.*

Proof. (Sketch.) As in Section 4, we will map an extensive form congestion game to a graphical game, and omit the rest of the argument, which is also akin to Section 3.2. The graphical game construction is similar to that in Section 3.1. Using nodes with strategy sets $\{0, 1\}$,

1. For every information set I with $p(I) = p$ and an outgoing edge label $j \in L_I$, make a node $\sigma_j^{p,I}$, to represent the probability of picking j.
2. For every information set I and every $j \in L_I$ make a node $\mathcal{U}_j^I$; the value of $\mathcal{U}_j^I$ will represent the utility of player $p(I)$ resulting from the optimal choice of distributions player $p(I)$ can make in the part of the tree below information set I given that the player arrived at information set I and chose j and assuming that the other players play as prescribed by the values $\{\sigma_j^{q,I'}\}_{q=p(I') \neq p}$; the weighting of the vertices of I when computing $\mathcal{U}_j^I$ is defined by the probabilities of the other players on the edges that connect I to the closest information set of $p(I)$ above I. Let $\mathcal{U}^I$ be the maximum over $\mathcal{U}_j^I$. Assuming the values $\mathcal{U}^{I'}$ for the information sets I' below I are computed, value $\mathcal{U}_j^I$ can be found by arithmetic operations.
3. Finally, for every information set, take a gadget $\mathcal{G}_I$ similar to $\mathcal{G}_p$ above that guarantees that (*i*) $\sum_{j \in L_I} \sigma_j^{p(I),I} = 1$, and (*ii*) $\mathcal{U}_j^I > \mathcal{U}_{j'}^I \Longrightarrow \sigma_{j'}^{p(I),I} = 0$.

Further details are omitted. The construction works by arguments parallel to the proof of Theorem 1. □

References

1. R. J. Aumann. "Subjectivity and Correlation in Randomized Strategies," *Journal of Mathematical Economics,* 1 , pp. 67-95, 1974.
2. P. Bürgisser. "On the structure of Valiant's complexity classes," *Discr. Math. Theoret. Comp. Sci.*, **3**, pp. 73–94, 1999.
3. X. Chen and X. Deng. "3-NASH is PPAD-Complete," *ECCC*, TR05-134, 2005.
4. X. Chen and X. Deng. "Settling the Complexity of 2-Player Nash-Equilibrium," *ECCC*, TR05-140, 2005.
5. C. Daskalakis and C. H. Papadimitriou. "Three-Player Games Are Hard," *ECCC*, TR05-139, 2005.
6. C. Daskalakis, P. W. Goldberg and C. H. Papadimitriou. "The Complexity of Computing a Nash Equilibrium," *Proceedings of 38th STOC*, 2006.
7. A. Fabrikant, C.H. Papadimitriou and K. Talwar. "The Complexity of Pure Nash Equilibria," *Proceedings of 36th STOC*, 2004.
8. J. Feigenbaum, D. Koller and P. Shor. "A game-theoretic classification of interactive complexity classes," *IEEE Conference on Structure in Complexity Theory*, 1995.
9. L. Fortnow, R. Impagliazzo, V. Kabanets and C. Umans. "On the Complexity of Succinct Zero-Sum Games," *IEEE Conference on Computational Complexity*, 2005.
10. J. Geanakoplos. "Nash and Walras Equilibrium via Brouwer," *Economic Theory, 21*, 2003.
11. P. W. Goldberg and C. H. Papadimitriou. "Reducibility Among Equilibrium Problems," *Proceedings of 38th STOC*, 2006.
12. D. S. Johnson, C. H. Papadimitriou and M. Yannakakis, "How Easy is Local Search?," *J. Comput. Syst. Sci. 37*, 1, pp. 79–100,1988.
13. M. Kearns, M. Littman and S. Singh. "Graphical Models for Game Theory," *In UAI*, 2001.
14. C. E. Lemke and J. T. Howson Jr.. "Equilibrium points of bimatrix games," *Journal of the Society for Industrial and Applied Mathematics*, 12, 413-423, 1964.
15. J. Nash. "Noncooperative Games," *Annals of Mathematics*, 54, 289-295, 1951.
16. M.J. Osborne and A. Rubinstein. *A Course in Game Theory*, MIT Press, 1994.
17. C. H. Papadimitriou. "Computing Correlated Equilibria in Multiplayer Games," *STOC*, 2005.
18. C. H. Papadimitriou. "On the Complexity of the Parity Argument and Other Inefficient Proofs of Existence," *J. Comput. Syst. Sci.* 48, 3, pp. 498–532, 1994.
19. R. Savani and B. von Stengel. "Exponentially many steps for finding a Nash equilibrium in a Bimatrix Game". *Proceedings of 45th FOCS*, 2004.
20. B. von Stengel, Computing equilibria for two-person games. In R.J. Aumann and S. Hart, editors, *Handbook of Game Theory, Vol. 3*, pp. 1723–1759. North-Holland, Amsterdam, 2002.