

The Complexity of Computing a Nash Equilibrium

Constantinos Daskalakis* Paul W. Goldberg† Christos H. Papadimitriou‡

June 4, 2008

Abstract

In 1951, John F. Nash proved that every game has a Nash equilibrium [43]. His proof is non-constructive, relying on Brouwer’s fixed point theorem, thus leaving open the questions: Is there a polynomial-time algorithm for computing Nash equilibria? And is this reliance on Brouwer inherent? Many algorithms have since been proposed for finding Nash equilibria, but none known to run in polynomial time. In 1991 the complexity class PPAD, for which Brouwer’s problem is complete, was introduced [48], motivated largely by the classification problem for Nash equilibria; but whether the Nash problem is complete for this class remained open. In this paper we resolve these questions: We show that finding a Nash equilibrium in three-player games is indeed PPAD-complete; and we do so by a reduction from Brouwer’s problem, thus establishing that the two problems are computationally equivalent. Our reduction simulates a (stylized) Brouwer function by a graphical game [33], relying on “gadgets,” graphical games performing various arithmetic and logical operations. We then show how to simulate this graphical game by a three-player game, where each of the three players is essentially a color class in a coloring of the underlying graph. Subsequent work [8] established, by improving our construction, that even two-player games are PPAD-complete; here we show that this result follows easily from our proof.

Categories and Subject Descriptors

F. 2. 0 [Analysis of Algorithms and Problem Complexity]: General

General Terms

Theory, Algorithms, Economics

Keywords

Complexity, Nash Equilibrium, PPAD-Completeness, Game Theory

*Computer Science Division, University of California at Berkeley. Research supported by NSF ITR Grants CCR-0121555 and CCF-0515259 and a grant from Microsoft Research. **email:** costis@cs.berkeley.edu

†Department of Computer Science, University of Liverpool. Research supported by the EPSRC grant GR/T07343/01 “Algorithmics of Network-sharing Games”. This work was begun while the author was visiting UC Berkeley. **email:** P.W.Goldberg@liverpool.ac.uk

‡Computer Science Division, University of California at Berkeley. Research supported by NSF ITR grants CCR-0121555 and CCF-0515259 and a grant from Microsoft Research. **email:** christos@cs.berkeley.edu

Contents

1	Introduction	3
2	Background	7
2.1	Basic Definitions from Game Theory	7
2.2	Related Work on Computing Equilibria	8
3	The Class PPAD	10
3.1	Total Search Problems	10
3.2	Computing a Nash Equilibrium is in PPAD	11
3.3	The BROUWER Problem	18
4	Reductions Among Equilibrium Problems	23
4.1	Preliminaries: Game Gadgets	23
4.2	Reducing Graphical Games to Normal Form Games	27
4.3	Reducing Normal Form Games to Graphical Games	31
4.4	Combining the Reductions	36
4.5	Reducing to Three Players	37
4.6	Preservation of Approximate equilibria	42
4.7	Reductions Between Different Notions of Approximation	51
5	The Main Reduction	54
6	Further Results and Open Problems	62
6.1	Two Players	62
6.2	Approximate Nash Equilibria	64
6.3	Nash Equilibria in Graphical Games	65
6.4	Special Cases	65
6.5	Further Applications of our Techniques	65

1 Introduction

Game Theory is one of the most important and vibrant mathematical fields established during the 20th century. In 1928, John von Neumann, extending work by Borel, showed that any two-person zero-sum game has an equilibrium — in fact, a min-max pair of randomized strategies [44]. Two decades later it was understood that this is tantamount to Linear Programming duality [14], and thus (as it was established another three decades hence [34]) computationally tractable. However, it became clear with the publication of the seminal book [45] by von Neumann and Morgenstern that this two-player, zero-sum case is too specialized; for the more general and important non-zero sum and multi-player games no existence theorem was known.

In 1951, Nash showed that *every* game has an equilibrium in mixed strategies, hence called Nash equilibrium [43]. His argument for proving this powerful and momentous result relies on another famous and consequential result of the early 20th century, Brouwer’s fixed point theorem [35]. The original proof of that result is notoriously nonconstructive (Brouwer’s preoccupation with constructive Mathematics and Intuitionism notwithstanding); its modern combinatorial proof (based on Sperner’s Lemma, see, e.g., [48]) does suggest an algorithm for the problem of finding an approximate Brouwer fixed point (and therefore for finding a Nash equilibrium) — albeit one of exponential complexity. In fact, it can be shown that any “natural” algorithm for Brouwer’s problem (roughly, treating the Brouwer function as a black box, a property shared by all known algorithms for the problem) must be exponential [31]. Over the past half century there has been a great variety of other algorithmic approaches to the problem of finding a Nash equilibrium (see Section 2.2); unfortunately, none of these algorithms is known to run in polynomial time. Whether a Nash equilibrium in a given game can be found in polynomial time had remained an important open question.

Such an efficient algorithm would have many practical applications; however, the true importance of this question is conceptual. The Nash equilibrium is a proposed model and prediction of social behavior, and Nash’s theorem greatly enhances its plausibility. This credibility, however, is seriously undermined by the absence of an efficient algorithm. It is doubtful that groups of rational players are more powerful than computers — and it would be remarkable, and potentially very useful, if they were. To put it bluntly, “if your laptop can’t find it, then, probably, neither can the market.” Hence, whether an efficient algorithm for finding Nash equilibria exists is an important question in Game Theory, the field for which the Nash equilibrium is perhaps the most central concept.

Besides Game Theory, the 20th century saw the development of another great mathematical field, which also captured the century’s zeitgeist and has had tremendous growth and impact: Computational Complexity. However, the mainstream concepts and techniques developed by complexity theorists for classifying computational problems according to their difficulty — chief among them NP-completeness — are not directly applicable for fathoming the complexity of the problem of finding Nash equilibria, exactly because of Nash’s Theorem: Since a Nash equilibrium is always guaranteed to exist, NP-completeness does not seem useful in exploring the complexity of finding one. NP-complete problems seem to draw much of their difficulty from the possibility that a solution may not exist. How would a reduction from SATISFIABILITY to NASH (the problem of finding a Nash equilibrium) look like? Any attempt to define such a reduction quickly leads to $NP = coNP$.

Motivated mainly by this open question regarding Nash equilibria, Meggido and Papadimitriou [42] defined in the 1980s the complexity class TFNP (for “NP total functions”), consisting exactly of all search problems in NP for which every instance is guaranteed to have a solution. NASH of course belongs there, and so do many other important and natural problems, finitary versions of Brouwer’s problem included. But here there is a difficulty of a different sort: TFNP is a “semantic

class” [47], meaning that there is no easy way of recognizing nondeterministic Turing machines which define problems in TFNP —in fact the problem is undecidable; such classes are known to be devoid of complete problems.

To capture the complexity of NASH, and other important problems in TFNP, another step is needed: One has to group together into subclasses of TFNP total functions whose proofs of totality are similar. Most of these proofs work by essentially constructing an exponentially large graph on the solution space (with edges that are computed by some algorithm), and then applying a simple graph-theoretic lemma establishing the existence of a particular kind of node. The node whose existence is guaranteed by the lemma is the sought solution of the given instance. Interestingly, essentially all known problems in TFNP can be shown total by one of the following arguments:

- *In any dag there must be a sink.* The corresponding class, PLS for “polynomial local search” had already been defined in [32], and contains many important complete problems.
- *In any directed graph with outdegree one, and with one node with indegree zero, there must be a node with indegree at least two.* The corresponding class is PPP (for “polynomial pigeonhole principle”).
- *In any undirected graph with one odd-degree node, there must be another odd-degree node.* This defines a class called PPA for “polynomial parity argument” [48], containing many important combinatorial problems (unfortunately none of them are known to be complete).
- *In any directed graph with one unbalanced node (node with outdegree different from its indegree), there must be another unbalanced node.* The corresponding class is called PPAD for “polynomial parity argument for directed graphs,” and it contains NASH, BROUWER, and BORSUK-ULAM (finding approximate fixed points of the kind guaranteed by Brouwer’s Theorem and the Borsuk-Ulam Theorem, respectively, see [48]). The latter two were among the problems proven PPAD-complete in [48]. Unfortunately, NASH — the one problem which had motivated this line of research — was not shown PPAD-complete; it was conjectured that it is.

In this paper we show that NASH is PPAD-complete, thus answering the open questions discussed above. We show that this holds even for games with three players. In another result (which is a crucial component of our proof) we show that the same is true for *graphical games*. Thus, a polynomial-time algorithm for these problems would imply a polynomial algorithm for, e.g., computing Brouwer fixed points, despite the exponential lower bounds for large classes of algorithms [31], and the relativizations in [2] — oracles for which PPAD has no polynomial-time algorithm.

Our proof gives an affirmative answer to another important question arising from Nash’s Theorem, namely, whether the reliance of its proof on Brouwer’s fixed point theorem is inherent. Our proof is essentially a reduction in the opposite direction to Nash’s: An appropriately discretized and stylized PPAD-complete version of Brouwer’s fixed point problem in 3 dimensions is reduced to NASH.

The structure of the reduction is the following: We represent a point in the three-dimensional unit cube by three players each of which has two strategies. Thus, every combination of mixed strategies for these players corresponds naturally to a point in the cube. Now, suppose that we are given a function from the cube to itself represented as a circuit. We construct a graphical game in which the best responses of the three players representing a point in the cube implement the given function, so that the Nash equilibria of the game must correspond to Brouwer fixed points. This is done by decoding the coordinates of the point in order to find their binary representation

(inputs to the circuit), and then simulating the circuit that represents the Brouwer function by a graphical game — an important alternative form of games defined in [33], see Section 2.1. This part of the construction relies on certain “gadgets,” small graphical games acting as arithmetical gates and comparators. The graphical game thus “computes” (in the sense of a mixed strategy over two strategies representing a real number) the value of the circuit at the point represented by the mixed strategies of the original three players, and then induces the three players to add appropriate increments to their mixed strategy. This establishes a one-to-one correspondence between Brouwer fixed points of the given function and Nash equilibria of the graphical game and shows that NASH for graphical games is PPAD-complete.

One difficulty in this part of the reduction is related to *brittle comparators*. Our comparator gadget sets its output to 0 if the input players play mixed strategies x, y that satisfy $x < y$, to 1 if $x > y$, and to *anything* if $x = y$; moreover, it is not hard to see that no “robust” comparator gadget is possible, one that outputs a specific fixed value if the input is $x = y$. This in turn implies that no robust decoder from real to binary can be constructed; decoding will always be flaky for a non-empty subset of the unit cube and, at that set, arbitrary values can be output by the decoder. On the other hand, real to binary decoding would be very handy since the circuit representing the given Brouwer function should be simulated in binary arithmetic. We take care of this difficulty by computing the Brouwer function on a “microlattice” around the point of interest and averaging the results, thus smoothing out any effects from boundaries of measure zero.

To continue to our main result for three-player normal form games, we establish certain reductions between equilibrium problems. In particular, we show by reductions that the following three problems are equivalent:

- NASH for r -player (normal form) games, for any $r > 3$.
- NASH for three-player games.
- NASH for graphical games with two strategies per player and maximum degree three (that is, of the exact type used in the simulation of Brouwer functions).

Thus, all these problems and their generalizations are PPAD-complete (since the third one was already shown to be PPAD-complete).

Our results leave open the question of NASH for two-player games. This case had been thought to be a little easier, since linear programming-like techniques come into play and solutions consisting of rational numbers are guaranteed to exist [38]; on the contrary, as exhibited in Nash’s original paper, there are three-player games with only irrational equilibria. In the precursors of the current paper [30, 16, 19], it was conjectured that there is a polynomial algorithm for two-player NASH. Surprisingly, a few months after our proof was circulated, Chen and Deng [8] came up with a proof establishing that this problem is PPAD-complete as well. In the last section of the present paper we show how this result can be obtained by a simple modification of our proof.

The structure of the paper is as follows. In Section 2, we provide some background on game theory and survey previous work regarding the computation of equilibria. In Section 3, we review the complexity theory of total functions, we define the class PPAD which is central in our paper, and we describe a canonical version of the Brouwer Fixed Point computation problem which is PPAD-complete and will be the starting point for our main result. In Section 4, we present the game-gadget machinery needed for our proof of the main result and establish the computational equivalence of different Nash equilibrium computation problems; in particular, we describe a polynomial reduction from the problem of computing a Nash equilibrium in a normal form game of any constant number of players or a graphical game of any constant degree to that of computing a Nash equilibrium of a

three player normal form game. Finally, in Section 5 we present our main result that computing a Nash equilibrium of a 3-player normal form game is PPAD-hard. Section 6 contains some discussion of the result and future research directions.

2 Background

2.1 Basic Definitions from Game Theory

A *game in normal form* has $r \geq 2$ players, $1, \dots, r$, and for each player $p \leq r$ a finite set S_p of pure strategies. The set S of *pure strategy profiles* is the Cartesian product of the S_p 's. We denote the set of pure strategy profiles of all players other than p by S_{-p} . Also, for a subset T of the players we denote by S_T the set of pure strategy profiles of the players in T . Finally, for each p and $s \in S$ we have a *payoff* or *utility* $u_s^p \geq 0$ — also occasionally denoted $u_{j_s}^p$ for $j \in S_p$ and $s \in S_{-p}$. We refer to the set $\{u_s^p\}_{s \in S}$ as the *payoff table* of player p . Also, for notational convenience and unless otherwise specified, we will denote by $[t]$ the set $\{1, \dots, t\}$, for all $t \in \mathbb{N}$.

A *mixed strategy* for player p is a distribution on S_p , that is, real numbers $x_j^p \geq 0$ for each strategy $j \in S_p$ such that $\sum_{j \in S_p} x_j^p = 1$. A set of r mixed strategies $\{x_j^p\}_{j \in S_p, p \in [r]}$, is called a (*mixed*) *Nash equilibrium* if, for each p , $\sum_{s \in S} u_s^p x_s$ is maximized over all mixed strategies of p — where for a strategy profile $s = (s_1, \dots, s_r) \in S$, we denote by x_s the product $x_{s_1}^1 \cdot x_{s_2}^2 \cdots x_{s_r}^r$. That is, a Nash equilibrium is a set of mixed strategies from which no player has a unilateral incentive to deviate. It is well-known (see, e.g., [46]) that the following is an equivalent condition for a set of mixed strategies to be a Nash equilibrium:

$$\forall p \in [r], j, j' \in S_p : \sum_{s \in S_{-p}} u_{j_s}^p x_s > \sum_{s \in S_{-p}} u_{j'_s}^p x_s \implies x_{j'}^p = 0. \quad (1)$$

The summation $\sum_{s \in S_{-p}} u_{j_s}^p x_s$ in the above equation is the expected utility of player p if p plays pure strategy $j \in S_p$ and the other players use the mixed strategies $\{x_j^q\}_{j \in S_q, q \neq p}$. Nash's theorem [43] asserts that *every normal form game has a Nash equilibrium*.

We next turn to approximate notions of equilibrium. We say that a set of mixed strategies x is an ϵ -*approximately well supported Nash equilibrium*, or ϵ -*Nash equilibrium* for short, if the following holds:

$$\forall p \in [r], j, j' \in S_p : \sum_{s \in S_{-p}} u_{j_s}^p x_s > \sum_{s \in S_{-p}} u_{j'_s}^p x_s + \epsilon \implies x_{j'}^p = 0. \quad (2)$$

Condition (2) relaxes (1) in that it allows a strategy to have positive probability in the presence of another strategy whose expected payoff is better by at most ϵ .

This is the notion of approximate Nash equilibrium that we use in this paper. There is an alternative, and arguably more natural, notion, called ϵ -*approximate Nash equilibrium* [40], in which the expected utility of each player is required to be within ϵ of the optimum response to the other players' strategies. This notion is less restrictive than that of an approximately well supported one. More precisely, for any ϵ , an ϵ -Nash equilibrium is also an ϵ -approximate Nash equilibrium, whereas the opposite need not be true. Nevertheless, the following lemma, proved in Section 4.7, establishes that the two concepts are computationally related (a weaker version of this fact was pointed out in [9]).

Lemma 1 *Given an ϵ -approximate Nash equilibrium $\{x_j^p\}_{j,p}$ of a game \mathcal{G} we can compute in polynomial time a $\sqrt{\epsilon} \cdot (\sqrt{\epsilon} + 1 + 4(r-1)u_{\max})$ -approximately well supported Nash equilibrium $\{\hat{x}_j^p\}_{j,p}$, where r is the number of players and u_{\max} is the maximum entry in the payoff tables of \mathcal{G} .*

In the sequel we shall focus on the notion of approximately well-supported Nash equilibrium, but all our results will also hold for the notion of approximate Nash equilibrium. Notice that Nash's theorem ensures the existence of an ϵ -Nash equilibrium —and hence of an ϵ -approximate

Nash equilibrium— for every $\epsilon \geq 0$; in particular, for every ϵ there exists an ϵ -Nash equilibrium whose probabilities are integer multiples of $\epsilon/(2r \times u_{\max\text{sum}})$, where $u_{\max\text{sum}}$ is the maximum, over all players p , of the sum of all entries in the payoff table of p . This can be established by rounding a Nash equilibrium $\{x_j^p\}_{j,p}$ to a nearby (in total variation distance) set of mixed strategies $\{\hat{x}_j^p\}_{j,p}$ all the entries of which are integer multiples of $\epsilon/(2r \times u_{\max\text{sum}})$. Note, however, that a ϵ -Nash equilibrium may not be close to an exact Nash equilibrium; see [25] for much more on this important distinction.

A game in normal form requires $r|S|$ numbers for its description, an amount of information that is exponential in the number of players. A *graphical game* [33] is defined in terms of an undirected graph $G = (V, E)$ together with a set of strategies S_v for each $v \in V$. We denote by $\mathcal{N}(v)$ the set consisting of v and v 's neighbors in G , and by $S_{\mathcal{N}(v)}$ the set of all $|\mathcal{N}(v)|$ -tuples of strategies, one from each vertex in $\mathcal{N}(v)$. In a graphical game, the utility of a vertex $v \in V$ only depends on the strategies of the vertices in $\mathcal{N}(v)$ so it can be represented by just $|\mathcal{N}(v)|$ numbers. In other words, a graphical game is a succinct representation of a multiplayer game, advantageous when it so happens that the utility of each player only depends on a few other players. A generalization of graphical games are the *directed graphical games*, where G is directed and $\mathcal{N}(v)$ consists of v and the predecessors of v . The two notions are almost identical; of course, the directed graphical games are more general than the undirected ones, but any directed graphical game can be represented, albeit less concisely, as an undirected game whose graph is the same except with no direction on the edges. In the remaining of the paper, we will not be very careful in distinguishing the two notions; our results will apply to both. The following is a useful definition.

Definition 1 *Suppose that \mathcal{GG} is a graphical game with underlying graph $G = (V, E)$. The affects-graph $G' = (V, E')$ of \mathcal{GG} is a directed graph with edge $(v_1, v_2) \in E'$ if the payoff to v_2 depends on the action of v_1 , that is, the payoff to v_2 is a non-constant function of the action of v_1 .*

In the above definition, an edge (v_1, v_2) in G' represents the relationship “ v_1 affects v_2 ”. Notice that if $(v_1, v_2) \in E'$ then $\{v_1, v_2\} \in E$, but the opposite need not be true—it could very well be that some vertex v_2 is affected by another vertex v_1 , but vertex v_1 is not affected by v_2 .

Since graphical games are representations of multi-player games, it follows by Nash’s theorem that every graphical game has a mixed Nash equilibrium. It can be checked that a set of mixed strategies $\{x_j^v\}_{j \in S_v, v \in V}$, is a mixed Nash equilibrium if and only if

$$\forall v \in V, j, j' \in S_v : \sum_{s \in S_{\mathcal{N}(v)} \setminus \{v\}} u_{j's}^v x_s > \sum_{s \in S_{\mathcal{N}(v)} \setminus \{v\}} u_{j's}^v x_s \implies x_{j'}^v = 0.$$

Similarly the condition for an approximately well supported Nash equilibrium can be derived.

2.2 Related Work on Computing Equilibria

Many papers in the economic, optimization, and computer science literature over the past 50 years study the computation of Nash equilibria. A celebrated algorithm for computing equilibria in 2-player games, which appears to be efficient in practice, is the Lemke-Howson algorithm [38]. The algorithm can be generalized to multi-player games, see, e.g., the work of Rosenmüller [51] and Wilson [57], albeit with some loss of efficiency. It was recently shown to be exponential in the worst case [53]. Other algorithms are based on computing approximate fixed points, most notably algorithms that walk on simplicial subdivisions of the space where the equilibria lie [54, 27, 36, 37, 23]. None of these algorithms is known to be polynomial-time.

Lipton and Markakis [39] study the algebraic properties of Nash equilibria, and point out that standard quantifier elimination algorithms can be used to solve them, but these are not polynomial-time in general. Papadimitriou and Roughgarden [50] show that, in the case of *symmetric* games, quantifier elimination results in polynomial algorithms for a broad range of parameters. Lipton, Markakis and Mehta [40] show that, if we only require an ϵ -approximate Nash equilibrium, then a subexponential algorithm is possible. If the Nash equilibria sought are required to have any special properties, for example optimize total utility, the problem typically becomes NP-complete [29, 13]. In addition to our work, as communicated in [30, 16, 19], other researchers (see, e.g., [5, 1, 11, 55]) have explored reductions between alternative types of games.

In particular, the reductions by Bubelis [5] in the 1970s comprise a remarkable early precursor of our work; it is astonishing that these important results had not been pursued for three decades. Bubelis established that the Nash equilibrium problem for 3 players captures the computational complexity of the same problem with any number of players. In Section 4 we show the same result in an indirect way, via the Nash equilibrium problem for graphical games — a connection that is crucial for our PPAD-completeness reduction. Bubelis also demonstrated in [5] that any algebraic number can be the basis of a Nash equilibrium, something that follows easily from our results (Theorem 14).

Etessami and Yannakakis studied in [25] the problem of computing a Nash equilibrium *exactly* (a problem that is well-motivated in the context of stochastic games) and came up with an interesting characterization of its complexity (considerably higher than PPAD), along with that of several other problems. In Section 6.5, we mention certain interesting results at the interface of [25]’s approach with ours.

3 The Class PPAD

3.1 Total Search Problems

A *search problem* \mathcal{S} is a set of inputs $I_{\mathcal{S}} \subseteq \Sigma^*$ on some alphabet Σ such that for each $x \in I_{\mathcal{S}}$ there is an associated set of solutions $\mathcal{S}_x \subseteq \Sigma^{|x|^k}$ for some integer k , such that for each $x \in I_{\mathcal{S}}$ and $y \in \Sigma^{|x|^k}$ whether $y \in \mathcal{S}_x$ is decidable in polynomial time. Notice that this is precisely NP with an added emphasis on finding a witness.

For example, let us define r -NASH to be the search problem \mathcal{S} in which each $x \in I_{\mathcal{S}}$ is an r -player game in normal form together with a binary integer A (the *accuracy specification*), and \mathcal{S}_x is the set of $\frac{1}{A}$ -Nash equilibria of the game (where the probabilities are rational numbers of bounded size as discussed). Similarly, d -GRAPHICAL NASH is the search problem with inputs the set of all graphical games with degree at most d , plus an accuracy specification A , and solutions the set of all $\frac{1}{A}$ -Nash equilibria. (For $r > 2$ it is important to specify the problem in terms of a search for *approximate* Nash equilibrium — exact solutions may need to be high-degree algebraic numbers, raising the question of how to represent them as bit strings.)

A search problem is *total* if $\mathcal{S}_x \neq \emptyset$ for all $x \in I_{\mathcal{S}}$. For example, Nash’s 1951 theorem [43] implies that r -NASH is total. Obviously, the same is true for d -GRAPHICAL NASH. The set of all total search problems is denoted TFNP. A *polynomial-time reduction* from total search problem \mathcal{S} to total search problem \mathcal{T} is a pair f, g of polynomial-time computable functions such that, for every input x of \mathcal{S} , $f(x)$ is an input of \mathcal{T} , and furthermore for every $y \in \mathcal{T}_{f(x)}$, $g(y) \in \mathcal{S}_x$.

TFNP is what in Complexity is sometimes called a “semantic” class [47], i.e., it has no generic complete problem. Therefore, the complexity of total functions is typically explored via “syntactic” subclasses of TFNP, such as PLS [32], PPP, PPA and PPAD [48]. In this paper we focus on PPAD.

PPAD can be defined in many ways. As mentioned in the introduction, it is, informally, the set of all total functions whose totality is established by invoking the following simple lemma on a graph whose vertex set is the solution space of the instance:

In any directed graph with one unbalanced node (node with outdegree different from its indegree), there is another unbalanced node.

This general principle can be specialized, without loss of generality or computational power, to the case in which every node has both indegree and outdegree at most one. In this case the lemma becomes:

In any directed graph in which all vertices have indegree and outdegree at most one, if there is a source (a node with indegree zero), then there must be a sink (a node with outdegree zero).

Formally, we shall define PPAD as the class of all total search problems polynomial-time reducible to the following problem:

END OF THE LINE: *Given two circuits S and P , each 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 $G_{S,P}$ with vertex set $\{0, 1\}^n$ and an edge from x to y whenever both $y = S(x)$ and $x = P(y)$; S and P stand for “successor candidate” and “predecessor candidate”. All vertices in $G_{S,P}$ have indegree and outdegree at most one, and there is at least one source, namely 0^n , so there must be a sink. We seek either a sink, or a source other than 0^n . Notice that in this problem a sink *or a source other than 0^n* is sought; if we insist on a sink, another complexity class called PPADS, apparently larger than PPAD, results.

The other important classes PLS, PPP and PPA, and others, are defined in a similar fashion based on other elementary properties of finite graphs. These classes are of no relevance to our analysis so their definition will be skipped; the interested reader is referred to [48].

A search problem \mathcal{S} in PPAD is called *PPAD-complete* if all problems in PPAD reduce to it. Obviously, END OF THE LINE is PPAD-complete; furthermore, it was shown in [48] that several problems related to topological fixed points and their combinatorial underpinnings are PPAD-complete: BROUWER, SPERNER, BORSUK-ULAM, TUCKER. Our main result in this paper (Theorem 12) states that so are the problems 3-NASH and 3-GRAPHICAL NASH.

3.2 Computing a Nash Equilibrium is in PPAD

We establish that computing an approximate Nash equilibrium in an r -player game is in PPAD. The $r = 2$ case was shown in [48].

Theorem 1 *r -NASH is in PPAD, for $r \geq 2$.*

Proof. We reduce r -NASH to END OF THE LINE. Note that Nash’s original proof [43] utilizes Brouwer’s fixed point theorem — it is essentially a reduction from the problem of finding a Nash equilibrium to that of finding a Brouwer fixed point of a continuous function; the latter problem can be reduced, under certain continuity conditions, to END OF THE LINE, and is therefore in PPAD. The, rather elaborate, proof below makes this simple intuition precise.

Let \mathcal{G} be a normal form game with r players, $1, \dots, r$, and strategy sets $S_p = [n]$, for all $p \in [r]$, and let $\{u_s^p : p \in [r], s \in S\}$ be the utilities of the players. Also let $\epsilon < 1$. In time polynomial in $|\mathcal{G}| + \log(1/\epsilon)$, we will specify two circuits S and P each with $N = \text{poly}(|\mathcal{G}|, \log(1/\epsilon))$ input and output bits and $P(0^N) = 0^N \neq S(0^N)$, so that, given any solution to END OF THE LINE on input S , P , one can construct in polynomial time an ϵ -approximate Nash equilibrium of \mathcal{G} . This is enough for reducing r -NASH to END OF THE LINE by virtue of Lemma 1. Our construction of S , P builds heavily upon the simplicial approximation algorithm of Laan and Talman [37] for computing fixed points of continuous functions from the product space of unit simplices to itself.

Let $\Delta_n = \{x \in \mathbb{R}_+^n \mid \sum_{k=1}^n x_k = 1\}$ be the $(n - 1)$ -dimensional unit simplex. Then the space of mixed strategy profiles of the game is $\Delta_n^r := \times_{p=1}^r \Delta_n$. For notational convenience we embed Δ_n^r in $\mathbb{R}^{n \cdot r}$ and we represent elements of Δ_n^r as vectors in $\mathbb{R}^{n \cdot r}$. That is, if $(x^1, x^2, \dots, x^r) \in \Delta_n^r$ is a mixed strategy profile of the game, we identify this strategy profile with a vector $x = (x^1; x^2; \dots; x^r) \in \mathbb{R}^{n \cdot r}$ resulting from the concatenation of the mixed strategies. For $p \in [r]$ and $j \in [n]$ we denote by $x(p, j)$ the $((p - 1)n + j)$ -th coordinate of x , that is $x(p, j) := x_{(p-1)n+j}$.

We are about to describe our reduction from finding an ϵ -approximate Nash equilibrium to END OF THE LINE. The nodes of the END OF THE LINE graph will correspond to the simplices of a triangulation of Δ_n^r which we describe next.

Triangulation of the Product Space of Unit Simplices. For some d , to be specified later, we describe the triangulation of Δ_n^r induced by the regular grid of size d . For this purpose, let us denote by $\Delta_n(d)$ the set of points of Δ_n induced by the grid of size d , i.e.

$$\Delta_n(d) = \left\{ x \in \mathbb{R}_+^n \mid x = \left(\frac{y_1}{d}, \frac{y_2}{d}, \dots, \frac{y_n}{d} \right), y_j \in \mathbb{N}_0 \text{ and } \sum_j y_j = d \right\},$$

and similarly define $\Delta_n^r(d) = \times_{p=1}^r \Delta_n(d)$. Moreover, let us define the block diagonal matrix Q by

$$Q = \begin{pmatrix} Q^1 & 0 & \dots & 0 & 0 \\ 0 & Q^2 & & 0 & 0 \\ 0 & & & & \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \dots & 0 & Q^r \end{pmatrix},$$

where, for all $p \in [r]$, Q^p is the $n \times n$ matrix defined by

$$Q^p = \begin{pmatrix} -1 & 0 & \dots & 0 & 1 \\ 1 & -1 & & 0 & 0 \\ 0 & 1 & & & \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \dots & -1 & 0 \\ 0 & 0 & \dots & 1 & -1 \end{pmatrix}.$$

Let us denote by $q(p, j)$ the $((p-1)n + j)$ -th column of Q . It is clear that adding $q(p, j)^T/d$ to a mixed strategy profile corresponds to shifting probability mass of $1/d$ from strategy j of player p to strategy $(j \bmod n) + 1$ of player p .

For all $p \in [r]$ and $k \in [n]$, let us define a set of indices $I_{p,k}$ as $I_{p,k} := \{(p, j)\}_{j \leq k}$. Also, let us define a collection \mathcal{T} of sets of indices as follows

$$\mathcal{T} := \left\{ T \subseteq \bigcup_{p \in [r]} I_{p,n} \mid \forall p \in [r], \exists k \in [n-1] : T \cap I_{p,n} = I_{p,k} \right\}.$$

Suppose, now, that q_0 is a mixed strategy profile in which every player plays strategy 1 with probability 1, that is $q_0(p, 1) = 1$, for all $p \in [r]$, and for $T \in \mathcal{T}$ define the set

$$A(T) := \left\{ x \in \Delta_n^r \mid x = q_0 + \sum_{(p,j) \in T} a(p, j) q(p, j)^T/d \text{ for non-negative real numbers } a(p, j) \geq 0 \right\}.$$

Defining $T^* := \cup_{p \in [r]} I_{p, n-1}$, it is not hard to verify that

$$A(T^*) = \Delta_n^r.$$

Moreover, if, for $T \in \mathcal{T}$, we define $B(T) := A(T) \setminus \cup_{T' \in \mathcal{T}, T' \subset T} A(T')$, the collection $\{B(T)\}_{T \in \mathcal{T}}$ partitions the set Δ_n^r .

To define the triangulation of Δ_n^r let us fix some set $T \in \mathcal{T}$, some permutation $\pi : [|T|] \rightarrow T$ of the elements of T , and some $x_0 \in A(T) \cap \Delta_n^r(d)$. Let us then denote by $\sigma(x_0, \pi)$ the $|T|$ -simplex which is the convex hull of the points $x_0, \dots, x_{|T|}$ defined as follows

$$x_t = x_{t-1} + q(\pi(t))^T/d, \quad \text{for all } t = 1, \dots, |T|.$$

The following lemmas, whose proof can be found in [37], describe the triangulation of Δ_n^r . We define $A(T, d) := A(T) \cap \Delta_n^r(d)$, we denote by \mathcal{P}_T the set of all permutations $\pi : [|T|] \rightarrow T$, and we set

$$\Sigma_T := \{\sigma(x_0, \pi) \mid x_0 \in A(T, d), \pi \in \mathcal{P}_T, \sigma(x_0, \pi) \subseteq A(T)\}.$$

Lemma 2 ([37]) For all $T \in \mathcal{T}$, the collection of $|T|$ -simplices Σ_T triangulates $A(T)$.

Corollary 1 ([37]) Δ_n^r is triangulated by the collection of simplices Σ_{T^*} .

The Vertices of the END OF THE LINE Graph. The vertices of the graph in our construction will correspond to the elements of the set

$$\Sigma := \bigcup_{T \in \mathcal{T}} \Sigma_T.$$

Let us encode the elements of Σ with strings $\{0, 1\}^N$; choosing N polynomial in $|\mathcal{G}|$, the description size of \mathcal{G} , and $\log d$ is sufficient.

We proceed to define the edges of the END OF THE LINE graph in terms of a labeling of the points of the set $\Delta_n^r(d)$, which we describe next.

Labeling Rule. Recall the function $f : \Delta_n^r \rightarrow \Delta_n^r$ defined by Nash to establish the existence of an equilibrium [43]. To describe f , let $U_j^p(x) := \sum_{s \in S_{-p}} u_{js}^p x_s$ be the expected utility of player p , if p plays pure strategy $j \in [n]$ and the other players use the mixed strategies $\{x_j^q\}_{j \in [n], q \neq p}$; let also $U^p(x) := \sum_{s \in S} u_s^p x_s$ be the expected utility of player p if every player $q \in [r]$ uses mixed strategy $\{x_j^q\}_{j \in [n]}$. Then, the function f is described as follows:

$$f(x^1, x^2, \dots, x^r) = (y^1, y^2, \dots, y^r),$$

where, for each $p \in [r]$, $j \in [n]$,

$$y_j^p = \frac{x_j^p + \max(0, U_j^p(x) - U^p(x))}{1 + \sum_{k \in [n]} \max(0, U_k^p(x) - U^p(x))}.$$

It is not hard to see that f is continuous, and that $f(x)$ can be computed in time polynomial in the binary encoding size of x and \mathcal{G} . Moreover, it can be verified that any point $x \in \Delta_n^r$ such that $f(x) = x$ is a Nash equilibrium [43]. The following lemma establishes that f is λ -Lipschitz for $\lambda := [1 + 2U_{\max} r n(n + 1)]$, where U_{\max} is the maximum entry in the payoff tables of the game.

Lemma 3 For all $x, x' \in \Delta_n^r \subseteq \mathbb{R}^{n \cdot r}$ such that $\|x - x'\|_\infty \leq \delta$,

$$\|f(x) - f(x')\|_\infty \leq [1 + 2U_{\max} r n(n + 1)]\delta.$$

Proof. We use the following bound shown in Section 4.6, Lemma 14.

Lemma 4 For any game \mathcal{G} , for all $p \leq r$, $j \in S_p$,

$$\left| \sum_{s \in S_{-p}} u_{js}^p x_s - \sum_{s \in S_{-p}} u_{js}^p x'_s \right| \leq \max_{s \in S_{-p}} \{u_{js}^p\} \sum_{q \neq p} \sum_{i \in S_q} |x_i^q - x'_i{}^q|.$$

It follows that for all $p \in [r]$, $j \in [n]$,

$$\begin{aligned} |U_j^p(x) - U_j^p(x')| &\leq U_{\max} r n \delta \\ \text{and } |U^p(x) - U^p(x')| &\leq U_{\max} r n \delta. \end{aligned}$$

Denoting $B_j^p(x) := \max(0, U_j^p(x) - U^p(x))$, for all $p \in [r]$, $j \in [n]$, the above bounds imply that

$$\begin{aligned} |B_j^p(x) - B_j^p(x')| &\leq 2U_{\max}rn\delta, \\ \left| \sum_{k \in [n]} B_k^p(x) - \sum_{k \in [n]} B_k^p(x') \right| &\leq 2U_{\max}rn\delta \cdot n. \end{aligned}$$

Combining the above bounds we get that, for all $p \in [r]$, $j \in [n]$,

$$\begin{aligned} |y_j^p(x) - y_j^p(x')| &\leq |x_j^p - x_j^p(x')| + |B_j^p(x) - B_j^p(x')| + \left| \sum_{k \in [n]} B_k^p(x) - \sum_{k \in [n]} B_k^p(x') \right| \\ &\leq \delta + 2U_{\max}rn\delta + 2U_{\max}rn\delta \cdot n \\ &\leq [1 + 2U_{\max}rn(n+1)]\delta, \end{aligned}$$

where we made use of the following lemma:

Lemma 5 For any $x, x', y, y', z, z' \geq 0$ such that $\frac{x+y}{1+z} \leq 1$,

$$\left| \frac{x+y}{1+z} - \frac{x'+y'}{1+z'} \right| \leq |x-x'| + |y-y'| + |z-z'|.$$

Proof.

$$\begin{aligned} \left| \frac{x+y}{1+z} - \frac{x'+y'}{1+z'} \right| &= \left| \frac{(x+y)(1+z') - (x'+y')(1+z)}{(1+z)(1+z')} \right| \\ &= \left| \frac{(x+y)(1+z') - (x+y)(1+z) - ((x'-x) + (y'-y))(1+z)}{(1+z)(1+z')} \right| \\ &\leq \left| \frac{(x+y)(1+z') - (x+y)(1+z)}{(1+z)(1+z')} \right| + \left| \frac{((x'-x) + (y'-y))(1+z)}{(1+z)(1+z')} \right| \\ &\leq \left| \frac{(x+y)(z'-z)}{(1+z)(1+z')} \right| + |x'-x| + |y'-y| \\ &\leq \frac{x+y}{1+z} |z'-z| + |x'-x| + |y'-y| \leq |z'-z| + |x'-x| + |y'-y|. \end{aligned}$$

□

□

We describe a labeling of the points of the set $\Delta_n^r(d)$ in terms of the function f . The labels that we are going to use are the elements of the set $\mathcal{L} := \cup_{p \in [r]} I_{p,n}$. In particular,

We assign to a point $x \in \Delta_n^r$ the label (p, j) iff (p, j) is the lexicographically least index such that $x_j^p > 0$ and $f(x)_j^p - x_j^p \leq f(x)_k^q - x_k^q$, for all $q \in [r]$, $k \in [n]$.

This labeling rule satisfies the following properties:

- *Completeness:* Every point x is assigned a label; hence, we can define a labeling function $\ell : \Delta_n^r \rightarrow \mathcal{L}$.
- *Properness:* $x_j^p = 0$ implies $\ell(x) \neq (p, j)$.

- *Efficiency:* $\ell(x)$ is computable in time polynomial in the binary encoding size of x and \mathcal{G} .

A simplex $\sigma \in \Sigma$ is called *completely labeled* if all its vertices have different labels; a simplex $\sigma \in \Sigma$ is called *p -stopping* if it is completely labeled and, moreover, for all $j \in [n]$, there exists a vertex of σ with label (p, j) . Our labeling satisfies the following important property.

Theorem 2 ([37]) *Suppose a simplex $\sigma \in \Sigma$ is p -stopping for some $p \in [r]$. Then all points $x \in \sigma \subseteq \mathbb{R}^{n \cdot r}$ satisfy*

$$\|f(x) - x\|_\infty \leq \frac{1}{d}(\lambda + 1)n(n - 1).$$

Proof. It is not hard to verify that, for any simplex $\sigma \in \Sigma$ and for all pairs of points $x, x' \in \sigma$,

$$\|x - x'\|_\infty \leq \frac{1}{d}.$$

Suppose now that a simplex $\sigma \in \Sigma$ is p -stopping, for some $p \in [r]$, and that, for all $j \in [n]$, $z(j)$ is the vertex of σ with label (p, j) . Since, for any x , $\sum_{i \in [n]} x_i^p = 1 = \sum_{i \in [n]} f(x)_i^p$, it follows from the labeling rule that

$$f(z(j))_j^p - z(j)_j^p \leq 0, \forall j \in [n].$$

Hence, for all $x \in \sigma$, $j \in [n]$,

$$f(x)_j^p - x_j^p \leq f(z(j))_j^p - z(j)_j^p + (\lambda + 1)\frac{1}{d} \leq (\lambda + 1)\frac{1}{d},$$

where we used the fact that the diameter of σ is $\frac{1}{d}$ (in the infinity norm) and the function f is λ -Lipschitz. Hence, in the opposite direction, for all $x \in \sigma$, $j \in [n]$, we have

$$f(x)_j^p - x_j^p = - \sum_{i \in [n] \setminus \{j\}} (f(x)_i^p - x_i^p) \geq -(n - 1)(\lambda + 1)\frac{1}{d}.$$

Now, by the definition of the labeling rule, we have, for all $x \in \sigma$, $q \in [r]$, $j \in [n]$,

$$\begin{aligned} f(x)_j^q - x_j^q &\geq f(z(1))_j^q - z(1)_j^q - (\lambda + 1)\frac{1}{d} \\ &\geq f(z(1))_1^p - z(1)_1^p - (\lambda + 1)\frac{1}{d} \\ &\geq -(n - 1)(\lambda + 1)\frac{1}{d} - (\lambda + 1)\frac{1}{d} = -n(\lambda + 1)\frac{1}{d}, \end{aligned}$$

whereas

$$\begin{aligned} f(x)_j^q - x_j^q &= - \sum_{i \in [n] \setminus \{j\}} (f(x)_i^q - x_i^q) \\ &\leq (n - 1)n(\lambda + 1)\frac{1}{d}. \end{aligned}$$

Combining the above, it follows that, for all $x \in \sigma$,

$$\|f(x) - x\|_\infty \leq \frac{1}{d}(\lambda + 1)n(n - 1).$$

□

The Approximation Guarantee. By virtue of Theorem 2, if we choose

$$d := \frac{1}{\epsilon'} [2 + 2U_{\max} r n (n + 1)] n (n - 1),$$

then a p -stopping simplex $\sigma \in \Sigma$, for any $p \in [r]$, satisfies that, for all $x \in \sigma$,

$$\|f(x) - x\|_{\infty} \leq \epsilon',$$

which by Lemma 6 below implies that x is a $n\sqrt{\epsilon'(1 + nU_{\max})} \left(1 + \sqrt{\epsilon'(1 + nU_{\max})}\right) \max\{U_{\max}, 1\}$ -approximate Nash equilibrium. Choosing

$$\epsilon' := \frac{1}{1 + nU_{\max}} \left(\frac{\epsilon}{2n \max\{U_{\max}, 1\}} \right)^2,$$

implies that x is an ϵ -approximate Nash equilibrium.

Lemma 6 *If a vector $x = (x^1; x^2; \dots; x^r) \in \mathbb{R}^{n \cdot r}$ satisfies*

$$\|f(x) - x\|_{\infty} \leq \epsilon',$$

then x is a $n\sqrt{\epsilon'(1 + nU_{\max})} \left(1 + \sqrt{\epsilon'(1 + nU_{\max})}\right) \max\{U_{\max}, 1\}$ -approximate Nash equilibrium.

Proof. Let us fix some player $p \in [r]$, and assume, without loss of generality, that

$$U_1^p(x) \geq U_2^p(x) \geq \dots \geq U_k^p(x) \geq U^p(x) \geq U_{k+1}^p(x) \geq \dots \geq U_n^p(x).$$

For all $j \in [n]$, observe that $|f(x)_j^p - x_j^p| \leq \epsilon'$ implies

$$x_j^p \sum_{i \in [n]} B_i^p(x) \leq B_j^p(x) + \epsilon' \left(1 + \sum_{i \in [n]} B_i^p(x) \right).$$

Setting $\epsilon'' := \epsilon'(1 + nU_{\max})$, the above inequality implies

$$x_j^p \sum_{i \in [n]} B_i^p(x) \leq B_j^p(x) + \epsilon''. \quad (3)$$

Let us define $t := x_{k+1}^p + x_{k+2}^p + \dots + x_n^p$, and let us distinguish the following cases

- If $t \geq \frac{\sqrt{\epsilon''}}{U_{\max}}$, then summing Equation (3) for $j = k + 1, \dots, n$ implies

$$t \sum_{i \in [n]} B_i^p(x) \leq (n - k)\epsilon'',$$

which gives

$$B_1^p \leq \sum_{i \in [n]} B_i^p(x) \leq n\sqrt{\epsilon''}U_{\max}. \quad (4)$$

- If $t \leq \frac{\sqrt{\epsilon''}}{U_{\max}}$, then multiplying Equation (3) by x_j^p and summing over $j = 1, \dots, n$ gives

$$\sum_{j \in [n]} (x_j^p)^2 \sum_{i \in [n]} B_i^p(x) \leq \sum_{j \in [n]} x_j^p B_j^p(x) + \epsilon''. \quad (5)$$

Now observe that for any setting of the probabilities x_j^p , $j \in [n]$, it holds that

$$\sum_{j \in [n]} (x_j^p)^2 \geq \frac{1}{n}. \quad (6)$$

Moreover, observe that, since $U^p(x) = \sum_{j \in [n]} x_j^p U_j^p(x)$, it follows that

$$\sum_{j \in [n]} x_j^p (U_j^p(x) - U^p(x)) = 0,$$

which implies that

$$\sum_{j \in [n]} x_j^p B_j^p(x) + \sum_{j \geq k+1} x_j^p (U_j^p(x) - U^p(x)) = 0.$$

Plugging this into (5) implies

$$\sum_{j \in [n]} (x_j^p)^2 \sum_{i \in [n]} B_i^p(x) \leq \sum_{j \geq k+1} x_j^p (U^p(x) - U_j^p(x)) + \epsilon''.$$

Further, using (6) gives

$$\frac{1}{n} \sum_{i \in [n]} B_i^p(x) \leq \sum_{j \geq k+1} x_j^p (U^p(x) - U_j^p(x)) + \epsilon'',$$

which implies

$$\sum_{i \in [n]} B_i^p(x) \leq n(tU_{\max} + \epsilon'').$$

The last inequality then implies

$$B_1^p(x) \leq n(\sqrt{\epsilon''} + \epsilon''). \quad (7)$$

Combining (4) and (7), we have the following uniform bound

$$B_1^p(x) \leq n(\sqrt{\epsilon''} + \epsilon'') \max\{U_{\max}, 1\} =: \epsilon'''. \quad (8)$$

Since $B_1^p(x) = U_1^p(x) - U(x)$, it follows that player p cannot improve her payoff by more than ϵ''' by changing her strategy. This is true for every player, hence x is a ϵ''' -approximate Nash equilibrium. \square

The Edges of the END OF THE LINE Graph. Laan and Talman [37] describe a pivoting algorithm which operates on the set Σ , by specifying the following:

- a simplex $\sigma_0 \in \Sigma$, which is the *starting simplex*; σ_0 contains the point q_0 and is uniquely determined by the labeling rule;

- a partial one-to-one function $h : \Sigma \rightarrow \Sigma$, mapping a simplex to a neighboring simplex, which defines a *pivoting rule*; h has the following properties ¹:
 - σ_0 has no pre-image;
 - any simplex $\sigma \in \Sigma$ that has no image is a p -stopping simplex for some p ; and, any simplex $\sigma \in \Sigma \setminus \{\sigma_0\}$ that has no pre-image is a p -stopping simplex for some p ;
 - both $h(\sigma)$ and $h^{-1}(\sigma)$ are computable in time polynomial in the binary encoding size of σ , that is N , and \mathcal{G} —given that the labeling function ℓ is efficiently computable;

The algorithm of Laan and Talman starts off with the simplex σ_0 and employs the pivoting rule h until a simplex σ with no image is encountered. By the properties of h , σ must be p -stopping for some $p \in [r]$ and, by the discussion above, any point $x \in \sigma$ is an ϵ -approximate Nash equilibrium.

In our construction, the edges of the END OF THE LINE graph are defined in terms of the function h : if $h(\sigma) = \sigma'$, then there is a directed edge from σ to σ' . Moreover, the string 0^N is identified with the simplex σ_0 . Any solution to the END OF THE LINE problem thus defined corresponds by the above discussion to a simplex σ such that any point $x \in \sigma$ is an ϵ -approximate Nash equilibrium of \mathcal{G} . This concludes the construction. \square

3.3 The BROUWER Problem

In the proof of our main result we use a problem we call BROUWER, which is a discrete and simplified version of the search problem associated with Brouwer's fixed point theorem. We are given a continuous function ϕ from the 3-dimensional unit cube to itself, defined in terms of its values at the centers of 2^{3n} cubelets with side 2^{-n} , for some $n \geq 0$ ². At the center c_{ijk} of the cubelet K_{ijk} defined as

$$K_{ijk} = \{(x, y, z) : i \cdot 2^{-n} \leq x \leq (i+1) \cdot 2^{-n}, \\ j \cdot 2^{-n} \leq y \leq (j+1) \cdot 2^{-n}, \\ k \cdot 2^{-n} \leq z \leq (k+1) \cdot 2^{-n}\},$$

where i, j, k are integers in $\{0, 1, \dots, 2^n - 1\}$, the value of ϕ is $\phi(c_{ijk}) = c_{ijk} + \delta_{ijk}$, where δ_{ijk} is one of the following four vectors (also referred to as colors):

- $\delta_1 = (\alpha, 0, 0)$
- $\delta_2 = (0, \alpha, 0)$
- $\delta_3 = (0, 0, \alpha)$
- $\delta_0 = (-\alpha, -\alpha, -\alpha)$

Here $\alpha > 0$ is much smaller than the cubelet side, say 2^{-2n} .

Thus, to compute ϕ at the center of the cubelet K_{ijk} we only need to know which of the four displacements to add. This is computed by a circuit C (which is the only input to the problem) with $3n$ input bits and 2 output bits; $C(i, j, k)$ is the index r such that, if c is the center of cubelet K_{ijk} , $\phi(c) = c + \delta_r$. C is such that $C(0, j, k) = 1$, $C(i, 0, k) = 2$, $C(i, j, 0) = 3$, and

¹More precisely, the pivoting rule h of Laan and Talman is defined on a subset Σ' of Σ . For our purposes, let us extend their pivoting rule h to the set Σ by setting $h(\sigma) = \sigma$ for all $\sigma \in \Sigma \setminus \Sigma'$.

²The value of the function near the boundaries of the cubelets could be determined by interpolation—there are many simple ways to do this, and the precise method is of no importance to our discussion.

$C(2^n - 1, j, k) = C(i, 2^n - 1, k) = C(i, j, 2^n - 1) = 0$ (with conflicts resolved arbitrarily), so that the function ϕ maps the boundary to the interior of the cube. A vertex of a cubelet is called *panchromatic* if among the cubelets adjacent to it there are four that have all four displacements $\delta_0, \delta_1, \delta_2, \delta_3$. Sperner's Lemma guarantees that, for any circuit C satisfying the above properties, a panchromatic vertex exists, see, e.g., [48]. An alternative proof of this fact follows as a consequence of Theorem 3 below.

BROUWER is thus the following total problem: Given a circuit C as described above, find a panchromatic vertex. The relationship with Brouwer fixed points is that *fixed points of ϕ only ever occur in the vicinity of a panchromatic vertex*. We next show:

Theorem 3 BROUWER is PPAD-complete.

Proof. That BROUWER is in PPAD follows from the main result of this paper (Theorem 12), which is a reduction from BROUWER to r -NASH, which has been shown to be in PPAD in Theorem 1.

To show hardness, we shall reduce END OF THE LINE to BROUWER. Given circuits S and P with n inputs and outputs, as prescribed in that problem, we shall construct an “equivalent” instance of BROUWER, that is, another circuit C with $3m = 3(n + 4)$ inputs and two outputs that computes the color of each cubelet of side 2^{-m} , that is to say, the index i such that δ_i is the correct displacement of the Brouwer function at the center of the cubelet encoded into the $3m$ bits of the input. We shall first describe the Brouwer function ϕ explicitly, and then argue that it can be computed by a circuit.

Our description of ϕ proceeds as follows: We shall first describe a 1-dimensional subset L of the 3-dimensional unit cube, intuitively an embedding of the path-like directed graph $G_{S,P}$ implicitly given by S and P . Then we shall describe the 4-coloring of the 2^{3m} cubelets based on the description of L . Finally, we shall argue that colors are easy to compute locally, and that panchromatic vertices correspond to endpoints other than the standard source 0^n of $G_{S,P}$.

We assume that the graph $G_{S,P}$ is such that for each edge (u, v) , one of the vertices is even (ends in 0) and the other is odd; this is easy to guarantee by duplicating the vertices of $G_{S,P}$.

L will be orthonormal, that is, each of its segments will be parallel to one of the axes; all coordinates of endpoints of segments are integer multiples of 2^{-m} , a factor that we omit in the discussion below. Let $u \in \{0, 1\}^n$ be a vertex of $G_{S,P}$. By $\langle u \rangle$ we denote the integer between 0 and $2^n - 1$ whose binary representation is u . Associated with u there are two line segments of length 4 of L . The first, called the *principal* segment of u , has endpoints $u_1 = (8\langle u \rangle + 2, 3, 3)$ and $u'_1 = (8\langle u \rangle + 6, 3, 3)$. The other *auxiliary* segment has endpoints $u_2 = (3, 8\langle u \rangle + 6, 2^m - 3)$ and $u'_2 = (3, 8\langle u \rangle + 10, 2^m - 3)$. Informally, these segments form two dashed lines (each segment being a dash) that run along two edges of the cube and slightly in its interior (see Figure 1).

Now, for every vertex u of $G_{S,P}$, we connect u'_1 to u_2 by a line with three straight segments, with joints $u_3 = (8\langle u \rangle + 6, 8\langle u \rangle + 6, 3)$ and $u_4 = (8\langle u \rangle + 6, 8\langle u \rangle + 6, 2^m - 3)$. Finally, if there is an edge (u, v) in $G_{S,P}$, we connect u'_2 to v_1 by a jointed line with breakpoints $u_5 = (8\langle v \rangle + 2, 8\langle u \rangle + 10, 2^m - 3)$ and $u_6 = (8\langle v \rangle + 2, 8\langle u \rangle + 10, 3)$. This completes the description of the line L if we do the following perturbation: exceptionally, the principal segment of $u = 0^n$ has endpoints $0_1 = (2, 2, 2)$ and $0'_1 = (6, 2, 2)$ and the corresponding joint is $0_3 = (6, 6, 2)$.

It is easy to see that L traverses the interior of the cube without ever “nearly crossing itself”; that is, two points p, p' of L are closer than $3 \cdot 2^{-m}$ in Euclidean distance only if they are connected by a part of L that has length $8 \cdot 2^{-m}$ or less. (This is important in order for the coloring described below of the cubelets surrounding L to be well-defined.) To check this, just notice that segments of different types (e.g., $[u_3, u_4]$ and $[u'_2, u_5]$) come closer than $3 \cdot 2^{-m}$ only if they share an endpoint; segments of the same type on the $z = 3$ or the $z = 2^m - 3$ plane are parallel and at least 4 apart; and segments parallel to the z axis differ by at least 4 in either their x or y coordinates.

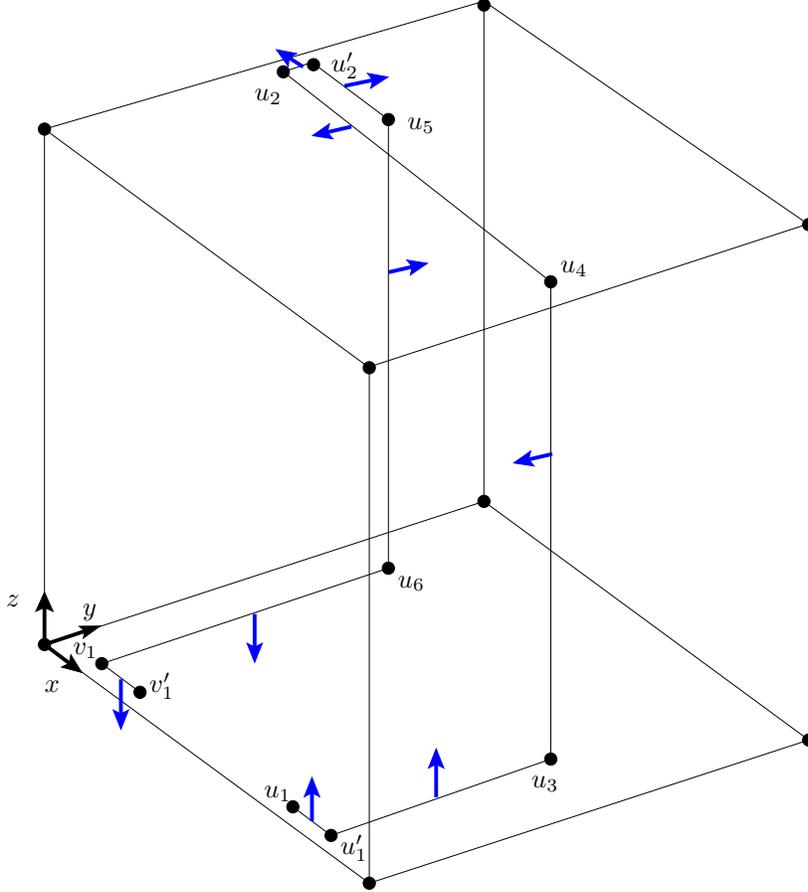


Figure 1: The orthonormal path connecting vertices (u,v) ; the arrows indicate the orientation of colors surrounding the path.

We now describe the coloring of the 2^{3m} cubelets by four colors corresponding to the four displacements. Consistent with the requirements for a BROWER circuit, we color any cubelet K_{ijk} where any one of i, j, k is $2^m - 1$, with 0. Given that, any other cubelet with $i = 0$ gets color 1; with this fixed, any other cubelet with $j = 0$ gets color 2, while the remaining cubelets with $k = 0$ get color 3. Having colored the boundaries, we now have to color the interior cubelets. An interior cubelet is always colored 0 *unless one of its vertices is a point of the interior of line L* , in which case it is colored by one of the three other colors in a manner to be explained shortly. Intuitively, at each point of the line L , starting from $(2, 2, 2)$ (the beginning of the principle segment of the string $u = 0^n$) the line L is “protected” from color 0 from all 4 sides. As a result, the only place where the four colors can meet is vertex u'_2 or $u_1, u \neq 0^n$, where u is an end of the line. . .

In particular, near the beginning of L at $(2, 2, 2)$ the 27 cubelets K_{ijk} with $i, j, k \leq 2$ are colored as shown in Figure 2. From then on, for any length-1 segment of L of the form $[(x, y, z), (x', y', z')]$ consider the four cubelets containing this segment. Two of these cubelets are colored 3, and the other two are colored 1 and 2, in this order clockwise (from the point of view of an observer at (x, y, z)). The remaining cubelets touching L are the ones at the joints where L turns. Each of these cubelets, a total of two per turn, takes the color of the two other cubelets adjacent to L with which it shares a face.

Now it only remains to describe, for each line segment $[a, b]$ of L , the direction d in which the

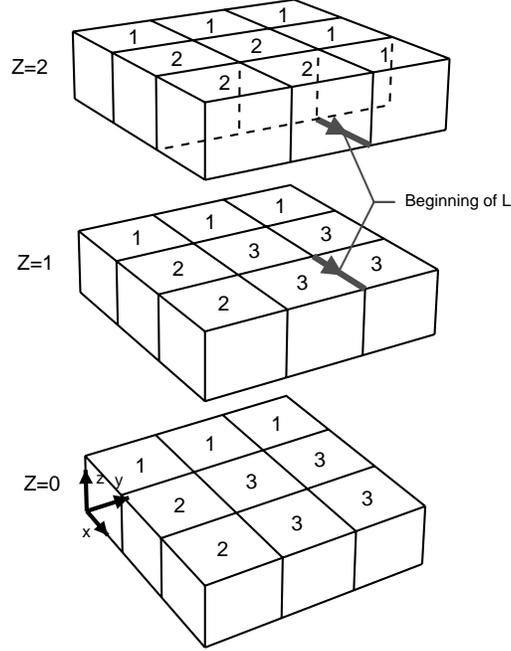


Figure 2: The 27 cubelets around the beginning of line L .

two cubelets that are colored 3 lie. The rules are these (in Figure 1 the directions d are shown as arrows):

- If $[a, b] = [u_1, u'_1]$ then $d = (0, 0, -1)$ if u is even and $d = (0, 0, 1)$ if u is odd.
- If $[a, b] = [u'_1, u_3]$ then $d = (0, 0, -1)$ if u is even and $d = (0, 0, 1)$ if u is odd.
- If $[a, b] = [u_3, u_4]$ then $d = (0, 1, 0)$ if u is even and $d = (0, -1, 0)$ if u is odd.
- If $[a, b] = [u_4, u_2]$ then $d = (0, 1, 0)$ if u is even and $d = (0, -1, 0)$ if u is odd.
- If $[a, b] = [u_2, u'_2]$ then $d = (1, 0, 0)$ if u is even and $d = (-1, 0, 0)$ if u is odd.
- If $[a, b] = [u'_2, u_5]$ then $d = (0, -1, 0)$ if u is even and $d = (0, 1, 0)$ if u is odd.
- If $[a, b] = [u_5, u_6]$ then $d = (0, -1, 0)$ if u is even and $d = (0, 1, 0)$ if u is odd.
- If $[a, b] = [u_6, v_1]$ then $d = (0, 0, 1)$ if u is even and $d = (0, 0, -1)$ if u is odd.

This completes the description of the construction. Notice that, for this to work, we need our assumption that edges in $G_{S,P}$ go between odd and even vertices. Regarding the alternating orientation of colored cubelets around L , note that we could not simply introduce “twists” to make them always point in (say) direction $d = (0, 0, -1)$ for all $[u_1, u'_1]$. That would create a panchromatic vertex at the location of a twist.

The result now follows from the following two claims:

1. A point in the cube is panchromatic in the described coloring if and only if it is
 - (a) an endpoint u'_2 of a sink vertex u of $G_{S,P}$, or
 - (b) an endpoint u_1 of a source vertex $u \neq 0^n$ of $G_{S,P}$
2. A circuit C can be constructed in time polynomial in $|S| + |P|$, which computes, for each triple of binary integers $i, j, k < 2^m$, the color of cubelet K_{ijk} .

Regarding the first claim, the endpoint u'_2 of a sink vertex u , or the endpoint u_1 of a source vertex u other than 0^n , will be a point where L meets color 0, hence a panchromatic vertex. There is no alternative way that L can meet color 0 and no other way a panchromatic vertex can occur.

Regarding the second claim, circuit C is doing the following. $C(0, j, k) = 1$, for $j, k < 2^m - 1$, $C(i, 0, k) = 2$ for $i > 0$, $i, k < 2^m - 1$, $C(i, j, 0) = 3$ for $i, j > 0$, $i, j < 2^m - 1$. Then by default, $C(i, j, k) = 0$. However the following tests yield alternative values for $C(i, j, k)$, for cubelets adjacent to L . $LSB(x)$ denotes the least significant bit of x , equal to 1 if x is odd, 0 if x is even, and undefined if x is not an integer. For example, a $[u'_1, u_3], u \neq 0^n$ segment is given by (letting $x = \langle u \rangle$):

1. If $k = 2$ and $i = 8x + 5$ and $LSB(x) = 1$ and $j \in \{3, \dots, 8x + 6\}$ then $C(i, j, k) = 2$.
2. If $k = 2$ and $i = 8x + 6$ and $LSB(x) = 1$ and $j \in \{2, \dots, 8x + 6\}$ then $C(i, j, k) = 1$.
3. If $k = 3$ and ($i = 8x + 5$ or $i = 8x + 6$) and $LSB(x) = 1$ and $j \in \{2, \dots, 8x + 5\}$ then $C(i, j, k) = 3$.
4. If $k = 2$ and ($i = 8x + 5$ or $i = 8x + 6$) and $LSB(x) = 0$ and $j \in \{2, \dots, 8x + 6\}$ then $C(i, j, k) = 3$.
5. If $k = 3$ and $i = 8x + 5$ and $LSB(x) = 0$ and $j \in \{3, \dots, 8x + 5\}$ then $C(i, j, k) = 1$.
6. If $k = 3$ and $i = 8x + 6$ and $LSB(x) = 0$ and $j \in \{2, \dots, 8x + 5\}$ then $C(i, j, k) = 2$.

A $[u'_2, u_5]$ segment uses the circuits P and S , and, in the case $LSB(x) = 1$, $x = \langle u \rangle$, is given by:

1. If ($k = 2^m - 3$ or $k = 2^m - 4$) and $j = 8x + 10$ and $S(x) = x'$ and $P(x') = x$ and $i \in \{2, \dots, 8x' + 2\}$ then $C(i, j, k) = 3$.
2. If $k = 2^m - 3$ and $j = 8x + 9$ and $S(x) = x'$ and $P(x') = x$ and $i \in \{3, \dots, 8x' + 2\}$ then $C(i, j, k) = 1$.
3. If $k = 2^m - 4$ and $j = 8x + 9$ and $S(x) = x'$ and $P(x') = x$ and $i \in \{3, \dots, 8x' + 1\}$ then $C(i, j, k) = 2$.

The other segments are done in a similar way, and so the second claim follows. This completes the proof of hardness.

□

4 Reductions Among Equilibrium Problems

In the next section we show that r -NASH is PPAD-hard by reducing BROUWER to it. Rather than r -NASH, it will be more convenient to first reduce BROUWER to d -GRAPHICAL NASH, the problem of computing a Nash equilibrium in graphical games of degree d . Therefore, we need to show that the latter reduces to r -NASH. This will be the purpose of the current section; in fact, we will establish something stronger, namely that

Theorem 4 *For every fixed $d, r \geq 3$,*

- *Every r -player normal form game and every graphical game of degree d can be mapped in polynomial time to (a) a 3-player normal form game and (b) a graphical game with degree 3 and 2 strategies per player, such that there is a polynomial-time computable surjective mapping from the set of Nash equilibria of the latter to the set of Nash equilibria of the former.*
- *There are polynomial-time reductions from r -NASH and d -GRAPHICAL NASH to both 3-NASH and 3-GRAPHICAL NASH.*

Note that the first part of the theorem establishes mappings of exact equilibrium points between different games, whereas the second asserts that computing approximate equilibrium points in all these games is polynomial-time equivalent. The proof, which is quite involved, is presented in the following subsections. In Subsection 4.1, we present some useful ideas that enable the reductions described in Theorem 4, as well as prepare the necessary machinery for the reduction from BROUWER to d -GRAPHICAL NASH in Section 5. Subsections 4.2 through 4.6 provide the proof of the theorem. In Subsection 4.7, we establish a polynomial-time reduction from the problem of computing an approximately well supported Nash equilibrium to the problem of computing an approximate Nash equilibrium. A mapping from r -player games to 3-player games was already known by Bubelis [5].

4.1 Preliminaries: Game Gadgets

We describe the building blocks of our constructions. As we have observed earlier, if a player v has two pure strategies, say 0 and 1, then every mixed strategy of that player corresponds to a real number $\mathbf{p}[v] \in [0, 1]$ which is precisely the probability that the player plays strategy 1. Identifying players with these numbers, we are interested in constructing games that perform simple arithmetical operations on mixed strategies; for example, we are interested in constructing a game with two “input” players v_1 and v_2 and another “output” player v_3 so that in any Nash equilibrium the latter plays the sum of the former, i.e., $\mathbf{p}[v_3] = \min\{\mathbf{p}[v_1] + \mathbf{p}[v_2], 1\}$. Such constructions are considered below.

Notation: We use $x = y \pm \epsilon$ to denote $y - \epsilon \leq x \leq y + \epsilon$.

Proposition 1 *Let α be a non-negative real number. Let v_1, v_2, w be players in a graphical game $\mathcal{G}\mathcal{G}$ with two strategies per player, and suppose that the payoffs to v_2 and w are as follows.*

<i>Payoffs to v_2 :</i>	v_2 plays 0 v_2 plays 1	w plays 0 w plays 1 <hr style="border: 0; border-top: 1px solid black; margin: 0;"/> 0 1 1 0
--------------------------------------	--------------------------------	--

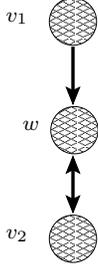


Figure 3: $\mathcal{G}_{\times\alpha}, \mathcal{G}_{=}$

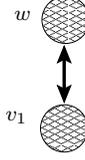


Figure 4: \mathcal{G}_{α}

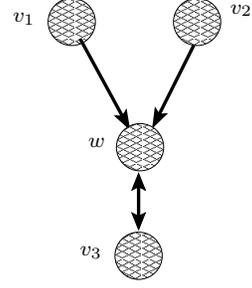


Figure 5: $\mathcal{G}_{+}, \mathcal{G}_{*}, \mathcal{G}_{-}$

Payoffs to w :

		v_2 plays 0	v_2 plays 1
w plays 0	v_1 plays 0	0	0
	v_1 plays 1	α	α
		v_2 plays 0	v_2 plays 1
w plays 1	v_1 plays 0	0	1
	v_1 plays 1	0	1

Then, for $\epsilon < 1$, in every ϵ -Nash equilibrium of game \mathcal{GG} , $\mathbf{p}[v_2] = \min(\alpha\mathbf{p}[v_1], 1) \pm \epsilon$. In particular, in every Nash equilibrium of game \mathcal{GG} , $\mathbf{p}[v_2] = \min(\alpha\mathbf{p}[v_1], 1)$.

Proof. If w plays 1, then the expected payoff to w is $\mathbf{p}[v_2]$, and, if w plays 0, the expected payoff to w is $\alpha\mathbf{p}[v_1]$. Therefore, in an ϵ -Nash equilibrium of \mathcal{GG} , if $\mathbf{p}[v_2] > \alpha\mathbf{p}[v_1] + \epsilon$ then $\mathbf{p}[w] = 1$. However, note also that if $\mathbf{p}[w] = 1$ then $\mathbf{p}[v_2] = 0$. (Payoffs to v_2 make it prefer to disagree with w .) Consequently, $\mathbf{p}[v_2]$ cannot be larger than $\alpha\mathbf{p}[v_1] + \epsilon$, so it cannot be larger than $\min(\alpha\mathbf{p}[v_1], 1) + \epsilon$. Similarly, if $\mathbf{p}[v_2] < \min(\alpha\mathbf{p}[v_1], 1) - \epsilon$, then $\mathbf{p}[v_2] < \alpha\mathbf{p}[v_1] - \epsilon$, so $\mathbf{p}[w] = 0$, which implies —again since v_2 has the biggest payoff by disagreeing with w — that $\mathbf{p}[v_2] = 1 \geq 1 - \epsilon$, a contradiction to $\mathbf{p}[v_2] < \min(\alpha\mathbf{p}[v_1], 1) - \epsilon$. Hence $\mathbf{p}[v_2]$ cannot be less than $\min(\alpha\mathbf{p}[v_1], 1) - \epsilon$. \square

We will denote by $\mathcal{G}_{\times\alpha}$ the (directed) graphical game shown in Figure 3, where the payoffs to players v_2 and w are specified as in Proposition 1 and the payoff of player v_1 is completely unconstrained: v_1 could have any dependence on other players of a larger graphical game \mathcal{GG} that contains $\mathcal{G}_{\times\alpha}$ or even depend on the strategies of v_2 and w ; as long as the payoffs of v_2 and w are specified as above the conclusion of the proposition will be true. Note in particular that using the above construction with $\alpha = 1$, v_2 becomes a “copy” of v_1 ; we denote the corresponding graphical game by $\mathcal{G}_{=}$. These graphical games will be used as building blocks in our constructions; the way to incorporate them into some larger graphical game is to make player v_1 depend (incoming edges) on other players of the game and make v_2 affect (outgoing edges) other players of the game. For example, we can make a sequence of copies of any vertex, which form a path in the graph. The copies then will alternate with distinct w vertices.

Proposition 2 Let α, β, γ be non-negative real numbers. Let v_1, v_2, v_3, w be players in a graphical game \mathcal{GG} with two strategies per player, and suppose that the payoffs to v_3 and w are as follows.

		w plays 0	w plays 1
Payoffs to v_3 :	v_3 plays 0	0	1
	v_3 plays 1	1	0

Payoffs to w :

		v_2 plays 0	v_2 plays 1
w plays 0	v_1 plays 0	0	β
	v_1 plays 1	α	$\alpha + \beta + \gamma$
w plays 1	v_3 plays 0	0	
	v_3 plays 1	1	

Then, for $\epsilon < 1$, in any ϵ -Nash equilibrium of \mathcal{GG} , $\mathbf{p}[v_3] = \min(\alpha\mathbf{p}[v_1] + \beta\mathbf{p}[v_2] + \gamma\mathbf{p}[v_1]\mathbf{p}[v_2], 1) \pm \epsilon$. In particular, in every Nash equilibrium of \mathcal{GG} , $\mathbf{p}[v_3] = \min(\alpha\mathbf{p}[v_1] + \beta\mathbf{p}[v_2] + \gamma\mathbf{p}[v_1]\mathbf{p}[v_2], 1)$.

Proof. If w plays 1, then the expected payoff to w is $\mathbf{p}[v_3]$, and if w plays 0 then the expected payoff to w is $\alpha\mathbf{p}[v_1] + \beta\mathbf{p}[v_2] + \gamma\mathbf{p}[v_1]\mathbf{p}[v_2]$. Therefore, in an ϵ -Nash equilibrium of \mathcal{GG} , if $\mathbf{p}[v_3] > \alpha\mathbf{p}[v_1] + \beta\mathbf{p}[v_2] + \gamma\mathbf{p}[v_1]\mathbf{p}[v_2] + \epsilon$ then $\mathbf{p}[w] = 1$. However, note from the payoffs to v_3 that if $\mathbf{p}[w] = 1$ then $\mathbf{p}[v_3] = 0$. Consequently, $\mathbf{p}[v_3]$ cannot be strictly larger than $\alpha\mathbf{p}[v_1] + \beta\mathbf{p}[v_2] + \gamma\mathbf{p}[v_1]\mathbf{p}[v_2] + \epsilon$. Similarly, if $\mathbf{p}[v_3] < \min(\alpha\mathbf{p}[v_1] + \beta\mathbf{p}[v_2] + \gamma\mathbf{p}[v_1]\mathbf{p}[v_2], 1) - \epsilon$, then $\mathbf{p}[v_3] < \alpha\mathbf{p}[v_1] + \beta\mathbf{p}[v_2] + \gamma\mathbf{p}[v_1]\mathbf{p}[v_2] - \epsilon$ and, due to the payoffs to w , $\mathbf{p}[w] = 0$. This in turn implies —since v_3 has the biggest payoff by disagreeing with w — that $\mathbf{p}[v_3] = 1 \geq 1 - \epsilon$, a contradiction to $\mathbf{p}[v_3] < \min(\alpha\mathbf{p}[v_1] + \beta\mathbf{p}[v_2] + \gamma\mathbf{p}[v_1]\mathbf{p}[v_2], 1) - \epsilon$. Hence $\mathbf{p}[v_3]$ cannot be less than $\min(\alpha\mathbf{p}[v_1] + \beta\mathbf{p}[v_2] + \gamma\mathbf{p}[v_1]\mathbf{p}[v_2], 1) - \epsilon$. \square

Remark 1 *It is not hard to verify that, if v_1, v_2, v_3, w are players of a graphical game \mathcal{GG} and the payoffs to v_3, w are specified as in Proposition 2 with $\alpha = 1, \beta = -1$ and $\gamma = 0$, then, in every ϵ -Nash equilibrium of the game \mathcal{GG} , $\mathbf{p}[v_3] = \max(0, \mathbf{p}[v_1] - \mathbf{p}[v_2]) \pm \epsilon$; in particular, in every Nash equilibrium, $\mathbf{p}[v_3] = \max(0, \mathbf{p}[v_1] - \mathbf{p}[v_2])$.*

Let us denote by \mathcal{G}_+ and \mathcal{G}_* the (directed) graphical game shown in Figure 5, where the payoffs to players v_3 and w are specified as in Proposition 2 taking (α, β, γ) equal to $(1, 1, 0)$ (addition) and $(0, 0, 1)$ (multiplication) respectively. Also, let \mathcal{G}_- be the game when the payoffs of v_3 and w are specified as in Remark 1.

Proposition 3 *Let $v_1, v_2, v_3, v_4, v_5, v_6, w_1, w_2, w_3, w_4$ be vertices in a graphical game \mathcal{GG} with two strategies per player, and suppose that the payoffs to vertices other than v_1 and v_2 are as follows.*

Payoffs to w_1 :

		v_2 plays 0	v_2 plays 1
w_1 plays 0	v_1 plays 0	0	0
	v_1 plays 1	1	1
w_1 plays 1	v_1 plays 0	0	1
	v_1 plays 1	0	1

		w_1 plays 0	w_1 plays 1
Payoffs to v_5 :	v_5 plays 0	1	0
	v_5 plays 1	0	1

Payoffs to w_2 and v_3 are chosen using Proposition 2 to ensure $\mathbf{p}[v_3] = \mathbf{p}[v_1](1 - \mathbf{p}[v_5]) \pm \epsilon$ ³, in every ϵ -Nash equilibrium of game \mathcal{GG} .

³We can use Proposition 2 to multiply by $(1 - \mathbf{p}[v_5])$ in a similar way to multiplication by $\mathbf{p}[v_5]$; the payoffs to w_2 have v_5 's strategies reversed.

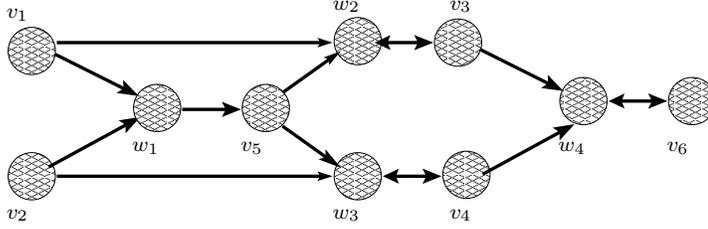


Figure 6: \mathcal{G}_{\max}

Payoffs to w_3 and v_4 are chosen using Proposition 2 to ensure $\mathbf{p}[v_4] = \mathbf{p}[v_2]\mathbf{p}[v_5] \pm \epsilon$, in every ϵ -Nash equilibrium of game $\mathcal{G}\mathcal{G}$.

Payoffs to w_4 and v_6 are chosen using Proposition 2 to ensure $\mathbf{p}[v_6] = \min(1, \mathbf{p}[v_3] + \mathbf{p}[v_4]) \pm \epsilon$, in every ϵ -Nash equilibrium of game $\mathcal{G}\mathcal{G}$.

Then, for $\epsilon < 1$, in every ϵ -Nash equilibrium of game $\mathcal{G}\mathcal{G}$, $\mathbf{p}[v_6] = \max(\mathbf{p}[v_1], \mathbf{p}[v_2]) \pm 4\epsilon$. In particular, in every Nash equilibrium, $\mathbf{p}[v_6] = \max(\mathbf{p}[v_1], \mathbf{p}[v_2])$.

The graph of the game looks as in Figure 6. It is actually possible to “merge” w_1 and v_5 , but we prefer to keep the game as is in order to maintain the bipartite structure of the graph in which one side of the partition contains all the vertices corresponding to arithmetic expressions (the v_i vertices) and the other side all the intermediate w_i vertices.

Proof. If, in an ϵ -Nash equilibrium, we have $\mathbf{p}[v_1] < \mathbf{p}[v_2] - \epsilon$, then it follows from w_1 ’s payoffs that $\mathbf{p}[w_1] = 1$. It then follows that $\mathbf{p}[v_5] = 1$ since v_5 ’s payoffs induce it to imitate w_1 . Hence, $\mathbf{p}[v_3] = \pm \epsilon$ and $\mathbf{p}[v_4] = \mathbf{p}[v_2] \pm \epsilon$, and, consequently, $\mathbf{p}[v_3] + \mathbf{p}[v_4] = \mathbf{p}[v_2] \pm 2\epsilon$. This implies $\mathbf{p}[v_6] = \mathbf{p}[v_2] \pm 3\epsilon$, as required. A similar argument shows that, if $\mathbf{p}[v_1] > \mathbf{p}[v_2] + \epsilon$, then $\mathbf{p}[v_6] = \mathbf{p}[v_1] \pm 3\epsilon$.

If $|\mathbf{p}[v_1] - \mathbf{p}[v_2]| \leq \epsilon$, then $\mathbf{p}[w_1]$ and, consequently, $\mathbf{p}[v_5]$ may take any value. Assuming, without loss of generality that $\mathbf{p}[v_1] \geq \mathbf{p}[v_2]$, we have

$$\begin{aligned} \mathbf{p}[v_3] &= \mathbf{p}[v_1](1 - \mathbf{p}[v_5]) \pm \epsilon \\ \mathbf{p}[v_4] &= \mathbf{p}[v_2]\mathbf{p}[v_5] \pm \epsilon = \mathbf{p}[v_1]\mathbf{p}[v_5] \pm 2\epsilon, \end{aligned}$$

which implies

$$\mathbf{p}[v_3] + \mathbf{p}[v_4] = \mathbf{p}[v_1] \pm 3\epsilon,$$

and, therefore,

$$\mathbf{p}[v_6] = \mathbf{p}[v_1] \pm 4\epsilon, \text{ as required.}$$

□

We conclude the section with the simple construction of a graphical game \mathcal{G}_α , depicted in Figure 4, which performs the assignment of some fixed value $\alpha \geq 0$ to a player. The proof is similar in spirit to our proof of Propositions 1 and 2 and will be skipped.

Proposition 4 *Let α be a non-negative real number. Let w, v_1 be players in a graphical game $\mathcal{G}\mathcal{G}$ with two strategies per player and let the payoffs to w, v_1 be specified as follows.*

		<i>w plays 0</i>	<i>w plays 1</i>
<i>Payoffs to v_1 :</i>	<i>v_1 plays 0</i>	0	1
	<i>v_1 plays 1</i>	1	0

Payoffs to w :		v_1 plays 0	v_1 plays 1
w plays 0		α	α
w plays 1		0	1

Then, for $\epsilon < 1$, in every ϵ -Nash equilibrium of game \mathcal{GG} , $\mathbf{p}[v_1] = \min(\alpha, 1) \pm \epsilon$. In particular, in every Nash equilibrium of \mathcal{GG} , $\mathbf{p}[v_1] = \min(\alpha, 1)$.

Before concluding the section we give a useful definition.

Definition 2 Let v_1, v_2, \dots, v_k, v be players of a graphical game \mathcal{G}_f such that, in every Nash equilibrium, it holds that $\mathbf{p}[v] = f(\mathbf{p}[v_1], \dots, \mathbf{p}[v_k])$, where f is some function with k arguments and range $[0, 1]$. We say that the game \mathcal{G}_f has error amplification at most c if, in every ϵ -Nash equilibrium, it holds that $\mathbf{p}[v] = f(\mathbf{p}[v_1], \dots, \mathbf{p}[v_k]) \pm c\epsilon$.

In particular, the games $\mathcal{G}_=, \mathcal{G}_+, \mathcal{G}_-, \mathcal{G}_*, \mathcal{G}_\alpha$ described above have error amplifications at most 1, whereas the game \mathcal{G}_{\max} has error amplification at most 4.

4.2 Reducing Graphical Games to Normal Form Games

We establish a mapping from graphical games to normal form games as specified by the following theorem.

Theorem 5 For every $d > 1$, a graphical game (directed or undirected) \mathcal{GG} of maximum degree d can be mapped in polynomial time to a $(d^2 + 1)$ -player normal form game \mathcal{G} so that there is a polynomial-time computable surjective mapping g from the Nash equilibria of the latter to the Nash equilibria of the former.

Proof. Overview:

Figure 7 shows the construction of $\mathcal{G} = f(\mathcal{GG})$. We will explain the construction in detail as well as show that it can be computed in polynomial time. We will also establish that there is a surjective mapping from the Nash equilibria of \mathcal{G} to the Nash equilibria of \mathcal{GG} . In the following discussion we will refer to the players of the graphical game as “vertices” to distinguish them from the players of the normal form game.

We first rescale all payoffs so that they are nonnegative and at most 1 (Step 1); it is easy to see that the set of Nash equilibria is preserved under this transformation. Also, without loss of generality, we assume that all vertices $v \in V$ have the same number of strategies, $|S_v| = t$. We color the vertices of G , where $G = (V, E)$ is the affects graph of \mathcal{GG} , so that any two adjacent vertices have different colors, but also any two vertices with a common successor have different colors (Step 3). Since this type of coloring will be important for our discussion we will define it formally.

Definition 3 Let \mathcal{GG} be a graphical game with affects graph $G = (V, E)$. We say that \mathcal{GG} can be legally colored with k colors if there exists a mapping $c : V \rightarrow \{1, 2, \dots, k\}$ such that, for all $e = (v, u) \in E$, $c(v) \neq c(u)$ and, moreover, for all $e_1 = (v, w), e_2 = (u, w) \in E$ with $v \neq u$, $c(v) \neq c(u)$. We call such coloring a legal k -coloring of \mathcal{GG} .

To get such coloring, it is sufficient to color the union of the underlying undirected graph G' with its square (with self-loops removed) so that no adjacent vertices have the same color; this can be done with at most d^2 colors —see, e.g., [6]— since G' has degree d by assumption; we are going to use $r = d^2$ or $r = d^2 + 1$ colors, whichever is even, for reasons to become clear shortly. We assume for simplicity that each color class has the same number of vertices, adding dummy vertices

Input: Degree d graphical game \mathcal{GG} : vertices V , $|V| = n'$, $|S_v| = t$ for all $v \in V$.

Output: Normal-form game \mathcal{G} .

1. If needed, rescale the entries in the payoff tables of \mathcal{GG} so that they lie in the range $[0, 1]$. One way to do so is to divide all payoff entries by $\max\{u\}$, where $\max\{u\}$ is the largest entry in the payoff tables of \mathcal{GG} .
2. Let $r = d^2$ or $r = d^2 + 1$; r chosen to be even.
3. Let $c : V \rightarrow \{1, \dots, r\}$ be a r -coloring of \mathcal{GG} such that no two adjacent vertices have the same color, and, furthermore, no two vertices having a common successor—in the affects graph of the game—have the same color. Assume that each color is assigned to the same number of vertices, adding to V extra isolated vertices to make up any shortfall; extend mapping c to these vertices. Let $\{v_1^{(i)}, \dots, v_{n/r}^{(i)}\}$ denote $\{v : c(v) = i\}$, where $n \geq n'$.
4. For each $p \in [r]$, game \mathcal{G} will have a player, labeled p , with strategy set S_p ; S_p will be the union (assumed disjoint) of all S_v with $c(v) = p$, i.e.,

$$S_p = \{(v, a) : c(v) = p, a \in S_v\}, \quad |S_p| = t \frac{n}{r}.$$

5. Taking S to be the cartesian product of the S_p 's, let $s \in S$ be a strategy profile of game \mathcal{G} . For $p \in [r]$, u_s^p is defined as follows:
 - (a) Initially, all utilities are 0.
 - (b) For $v_0 \in V$ having predecessors $v_1, \dots, v_{d'}$ in the affects graph of \mathcal{GG} , if $c(v_0) = p$ (that is, $v_0 = v_j^{(p)}$ for some j) and, for $i = 0, \dots, d'$, s contains (v_i, a_i) , then $u_s^p = u_{s'}^{v_0}$ for s' a strategy profile of \mathcal{GG} in which v_i plays a_i for $i = 0, \dots, d'$.
 - (c) Let $M > 2 \frac{n}{r}$.
 - (d) For odd number $p < r$, if player p plays $(v_i^{(p)}, a)$ and $p + 1$ plays $(v_i^{(p+1)}, a')$, for any i, a, a' , then add M to u_s^p and subtract M from u_s^{p+1} .

Figure 7: Reduction from graphical game \mathcal{GG} to normal form game \mathcal{G}

if needed to satisfy this property. Henceforth, we assume that n is an integer multiple of r so that every color class has $\frac{n}{r}$ vertices.

We construct a normal form game \mathcal{G} with $r \leq d^2 + 1$ players. Each of them corresponds to a color and has $t \frac{n}{r}$ strategies, the t strategies of each of the $\frac{n}{r}$ vertices in its color class (Step 4). Since r is even, we can divide the r players into pairs and make each pair play a generalized Matching Pennies game (see Definition 4 below) at very high stakes, so as to ensure that all players will randomize uniformly over the vertices assigned to them⁴. Within the set of strategies associated with each vertex, the Matching Pennies game expresses no preference, and payoffs are augmented to correspond to the payoffs that would arise in the original graphical game \mathcal{GG} (see Step 5 for the exact specification of the payoffs).

Definition 4 *The (2-player) game Generalized Matching Pennies is defined as follows. Call the 2 players the pursuer and the evader, and let $[n]$ denote their strategies. If for any $i \in [n]$ both players play i , then the pursuer receives a positive payoff $u > 0$ and the evader receives a payoff*

⁴A similar trick is used in Theorem 7.3 of [55], a hardness result for a class of circuit games.

of $-u$. Otherwise both players receive 0. It is not hard to check that the game has a unique Nash equilibrium in which both players use the uniform distribution.

Polynomial size of $\mathcal{G} = f(\mathcal{GG})$:

The input size is $|\mathcal{GG}| = \Theta(n' \cdot t^{d+1} \cdot q)$, where n' is the number of vertices in \mathcal{GG} and q the size of the values in the payoff matrices in the logarithmic cost model. The normal form game \mathcal{G} has $r \in \{d^2, d^2 + 1\}$ players, each having tn/r strategies, where $n \leq rn'$ is the number of vertices in \mathcal{GG} after the possible addition of dummy vertices to make sure that all color classes have the same number of vertices. Hence, there are $r \cdot \left(\frac{tn}{r}\right)^r \leq \left((d^2 + 1)(tn')^{d^2+1}\right)$ payoff entries in \mathcal{G} . This is polynomial in $|\mathcal{GG}|$ so long as d is constant. Moreover, each payoff entry will be of polynomial size since M is of polynomial size and each payoff entry of the game \mathcal{G} is the sum of 0 or M and a payoff entry of \mathcal{GG} .

Construction of the mapping g :

Given a Nash equilibrium $N_{\mathcal{G}} = \{x_{(v,a)}^p\}_{p,v,a}$ of $\mathcal{G} = f(\mathcal{GG})$, we claim that we can recover a Nash equilibrium $\{x_a^v\}_{v,a}$ of \mathcal{GG} , $N_{\mathcal{GG}} = g(N_{\mathcal{G}})$, as follows:

$$x_a^v := x_{(v,a)}^{c(v)} / \sum_{j \in S_v} x_{(v,j)}^{c(v)}, \quad \forall a \in S_v, v \in V. \quad (9)$$

Clearly g is computable in polynomial time.

Proof that g maps Nash equilibria of \mathcal{G} to Nash equilibria of \mathcal{GG} :

Call \mathcal{GG}' the graphical game resulting from \mathcal{GG} by rescaling the utilities so that they lie in the range $[0, 1]$. It is easy to see that any Nash equilibrium of game \mathcal{GG} is, also, a Nash equilibrium of game \mathcal{GG}' and vice versa. Therefore, it is enough to establish that the mapping g maps every Nash equilibrium of game \mathcal{G} to a Nash equilibrium of game \mathcal{GG}' .

For $v \in V$, $c(v) = p$, let “ p plays v ” denote the event that p plays (v, a) for some $a \in S_v$. We show that in a Nash equilibrium $N_{\mathcal{G}}$ of game \mathcal{G} , for every player p and every $v \in V$ with $c(v) = p$, $\Pr(p \text{ plays } v) \in [\lambda - \frac{1}{M}, \lambda + \frac{1}{M}]$, where $\lambda = \left(\frac{n}{r}\right)^{-1}$. Note that the “fair share” for v is λ .

Lemma 7 *For all $v \in V$, in a Nash equilibrium of \mathcal{G} , $\Pr(c(v) \text{ plays } v) \in [\lambda - \frac{1}{M}, \lambda + \frac{1}{M}]$.*

Proof. Suppose, for a contradiction, that in a Nash equilibrium of \mathcal{G} , $\Pr(p \text{ plays } v_i^{(p)}) < \lambda - \frac{1}{M}$ for some i, p . Then there exists some j such that $\Pr(p \text{ plays } v_j^{(p)}) > \lambda + \frac{1}{M}\lambda$.

If p is odd (a pursuer) then $p + 1$ (the evader) will have utility of at least $-\lambda M + 1$ for playing any strategy $(v_i^{(p+1)}, a)$, $a \in S_{v_i^{(p+1)}}$, whereas utility of at most $-\lambda M - \lambda + 1$ for playing any strategy $(v_j^{(p+1)}, a)$, $a \in S_{v_j^{(p+1)}}$. Since $-\lambda M + 1 > -\lambda M - \lambda + 1$, in a Nash equilibrium, $\Pr(p + 1 \text{ plays } v_j^{(p+1)}) = 0$. Therefore, there exists some k such that $\Pr(p + 1 \text{ plays } v_k^{(p+1)}) > \lambda$. Now the payoff of p for playing any strategy $(v_j^{(p)}, a)$, $a \in S_{v_j^{(p)}}$, is at most 1, whereas the payoff for playing any strategy $(v_k^{(p)}, a)$, $a \in S_{v_k^{(p)}}$ is at least λM . Thus, in a Nash equilibrium, player p should not include any strategy $(v_j^{(p)}, a)$, $a \in S_{v_j^{(p)}}$, in her support; hence $\Pr(p \text{ plays } v_j^{(p)}) = 0$, a contradiction.

If p is even, then $p - 1$ will have utility of at most $(\lambda - \frac{1}{M})M + 1$ for playing any strategy $(v_i^{(p-1)}, a)$, $a \in S_{v_i^{(p-1)}}$, whereas utility of at least $(\lambda + \frac{1}{M}\lambda)M$ for playing any strategy $(v_j^{(p-1)}, a)$,

$a \in S_{v_j^{(p-1)}}$. Hence, in a Nash equilibrium $\Pr(p-1 \text{ plays } v_i^{(p-1)}) = 0$, which implies that there exists some k such that $\Pr(p-1 \text{ plays } v_k^{(p-1)}) > \lambda$. But, then, p will have utility of at least 0 for playing any strategy $(v_i^{(p)}, a)$, $a \in S_{v_i^{(p)}}$, whereas utility of at most $-\lambda M + 1$ for playing any strategy $(v_k^{(p)}, a)$, $a \in S_{v_k^{(p)}}$. Since $0 > -\lambda M + 1$, in a Nash equilibrium, $\Pr(p \text{ plays } v_k^{(p)}) = 0$. Therefore, there exists some k' such that $\Pr(p \text{ plays } v_{k'}^{(p)}) > \lambda$. Now the payoff of $p-1$ for playing any strategy $(v_k^{(p-1)}, a)$, $a \in S_{v_k^{(p-1)}}$, is at most 1, whereas the payoff for playing any strategy $(v_{k'}^{(p-1)}, a)$, $a \in S_{v_{k'}^{(p-1)}}$ is at least λM . Thus, in a Nash equilibrium, player $p-1$ should not include any strategy $(v_k^{(p-1)}, a)$, $a \in S_{v_k^{(p-1)}}$, in her support; hence $\Pr(p-1 \text{ plays } v_k^{(p-1)}) = 0$, a contradiction.

From the above discussion, it follows that every vertex is chosen with probability at least $\lambda - \frac{1}{M}$ by the player that represents its color class. A similar argument shows that no vertex is chosen with probability greater than $\lambda + \frac{1}{M}$. Indeed, suppose, for a contradiction, that in a Nash equilibrium of \mathcal{G} , $\Pr(p \text{ plays } v_j^{(p)}) > \lambda + \frac{1}{M}$ for some j, p ; then there exists some i such that $\Pr(p \text{ plays } v_i^{(p)}) < \lambda - \frac{1}{M}\lambda$; now, distinguish two cases depending on whether p is even or odd and proceed in the same fashion as in the argument used above to show that no vertex is chosen with probability smaller than $\lambda - 1/M$. \square

To see that $\{x_a^v\}_{v,a}$, defined by (9), corresponds to a Nash equilibrium of \mathcal{GG}' note that, for any player p and vertex $v \in V$ such that $c(v) = p$, the division of $\Pr(p \text{ plays } v)$ into $\Pr(p \text{ plays } (v, a))$, for various values of $a \in S_v$, is driven entirely by the same payoffs as in \mathcal{GG}' ; moreover, note that there is some positive probability $p(v) \geq (\lambda - \frac{1}{M})^d > 0$ that the predecessors of v are chosen by the other players of \mathcal{G} and the additional expected payoff to p resulting from choosing (v, a) , for some $a \in S_v$, is $p(v)$ times the expected payoff of v in \mathcal{GG}' if v chooses action a and all other vertices play as specified by (9). More formally, suppose that $p = c(v)$ for some vertex v of the graphical game \mathcal{GG}' and, without loss of generality, assume that p is odd (pursuer) and that v is the vertex $v_i^{(p)}$ in the notation of Figure 7. Then, in a Nash equilibrium of the game \mathcal{G} , we have, by the definition of a Nash equilibrium, that for all strategies $a, a' \in S_v$ of vertex v :

$$\mathbb{E}[\text{payoff to } p \text{ for playing } (v, a)] > \mathbb{E}[\text{payoff to } p \text{ for playing } (v, a')] \Rightarrow x_{(v,a')}^p = 0. \quad (10)$$

But

$$\mathbb{E}[\text{payoff to } p \text{ for playing } (v, a)] = M \cdot \Pr(p+1 \text{ plays } v_i^{(p+1)}) + \sum_{s \in S_{\mathcal{N}(v) \setminus \{v\}}} u_{as}^v \prod_{u \in \mathcal{N}(v) \setminus \{v\}} x_{(u,s_u)}^{c(u)}$$

and similarly for a' . Therefore, (10) implies

$$\sum_{s \in S_{\mathcal{N}(v) \setminus \{v\}}} u_{as}^v \prod_{u \in \mathcal{N}(v) \setminus \{v\}} x_{(u,s_u)}^{c(u)} > \sum_{s \in S_{\mathcal{N}(v) \setminus \{v\}}} u_{a's}^v \prod_{u \in \mathcal{N}(v) \setminus \{v\}} x_{(u,s_u)}^{c(u)} \Rightarrow x_{(v,a')}^p = 0.$$

Dividing by $\prod_{u \in \mathcal{N}(v) \setminus \{v\}} \sum_{j \in S_u} x_{(u,j)}^{c(u)} = \prod_{u \in \mathcal{N}(v) \setminus \{v\}} \Pr(c(u) \text{ plays } u) = p(v)$ and invoking (9) gives

$$\sum_{s \in S_{\mathcal{N}(v) \setminus \{v\}}} u_{as}^v \prod_{u \in \mathcal{N}(v) \setminus \{v\}} x_{s_u}^u > \sum_{s \in S_{\mathcal{N}(v) \setminus \{v\}}} u_{a's}^v \prod_{u \in \mathcal{N}(v) \setminus \{v\}} x_{s_u}^u \Rightarrow x_{a'}^v = 0,$$

where we used that $p(v) \geq (\lambda - \frac{1}{M})^d > 0$, which follows by Lemma 7.

Mapping g is surjective on the Nash equilibria of \mathcal{GG}' and, therefore, \mathcal{GG} : We will show that, for every Nash equilibrium $N_{\mathcal{GG}'} = \{x_a^v\}_{v,a}$ of \mathcal{GG}' , there exists a Nash equilibrium $N_{\mathcal{G}} = \{x_{(v,a)}^p\}_{p,v,a}$ of \mathcal{G} such that (9) holds. The existence can be easily established via the existence of a Nash equilibrium in a game \mathcal{G}' defined as follows. Suppose that, in $N_{\mathcal{GG}'}$, every vertex $v \in V$ receives an expected payoff of u_v from every strategy in the support of $\{x_a^v\}_a$. Define the following game \mathcal{G}' whose structure results from \mathcal{G} by merging the strategies $\{(v,a)\}_a$ of player $p = c(v)$ into one strategy s_v^p , for every v such that $c(v) = p$. So the strategy set of player p in \mathcal{G}' will be $\{s_v^p \mid c(v) = p\}$ also denoted as $\{s_1^{(p)}, \dots, s_{n/r}^{(p)}\}$ for ease of notation. Define now the payoffs to the players as follows. Initialize the payoff matrices with all entries equal to 0. For every strategy profile s ,

- for $v_0 \in V$ having predecessors $v_1, \dots, v_{d'}$ in the affects graph of \mathcal{GG}' , if, for $i = 0, \dots, d'$, s contains $s_{v_i}^{c(v_i)}$, then add u_{v_0} to $u_s^{c(v_0)}$.
- for odd number $p < r$ if player p plays strategy $s_i^{(p)}$ and player $p + 1$ plays strategy $s_i^{(p+1)}$ then add M to u_s^p and subtract M from u_s^{p+1} (Generalized Matching Pennies).

Note the similarity in the definitions of the payoff matrices of \mathcal{G} and \mathcal{G}' . From Nash's theorem, game \mathcal{G}' has a Nash equilibrium $\{y_{s_v^p}^p\}_{p,v}$ and it is not hard to verify that $\{x_{(v,a)}^p\}_{p,v,a}$ is a Nash equilibrium of game \mathcal{G} , where $x_{(v,a)}^p := y_{s_v^p}^p \cdot x_a^v$, for all $p, v \in V$ such that $c(v) = p$, and $a \in S_v$. □

4.3 Reducing Normal Form Games to Graphical Games

We establish the following mapping from normal form games to graphical games.

Theorem 6 *For every $r > 1$, a normal form game with r players can be mapped in polynomial time to an undirected graphical game of maximum degree 3 and two strategies per player so that there is a polynomial-time computable surjective mapping g from the Nash equilibria of the latter to the Nash equilibria of the former.*

Given a normal form game \mathcal{G} having r players, $1, \dots, r$, and n strategies per player, say $S_p = [n]$ for all $p \in [r]$, we will construct a graphical game \mathcal{GG} , with a bipartite graph of maximum degree 3, and 2 strategies per player, say $\{0, 1\}$, with description length polynomial in the description length of \mathcal{G} , so that from every Nash equilibrium of \mathcal{GG} we can recover a Nash equilibrium of \mathcal{G} . In the following discussion we will refer to the players of the graphical game as “vertices” to distinguish them from the players of the normal form game. It will be easy to check that the graph of \mathcal{GG} is bipartite and has degree 3; this graph will be denoted $G = (V \cup W, E)$, where W and V are disjoint, and each edge in E goes between V and W . For every vertex v of the graphical game, we will denote by $\mathbf{p}[v]$ the probability that v plays pure strategy 1.

Recall that \mathcal{G} is specified by the quantities $\{u_s^p : p \in [r], s \in S\}$. A mixed strategy profile of \mathcal{G} is given by probabilities $\{x_j^p : p \in [r], j \in S_p\}$. \mathcal{GG} will contain a vertex $v(x_j^p) \in V$ for each player p and strategy $j \in S_p$, and the construction of \mathcal{GG} will ensure that in any Nash equilibrium of \mathcal{GG} , the quantities $\{\mathbf{p}[v(x_j^p)] : p \in [r], j \in S_p\}$, if interpreted as values $\{x_j^p\}_{p,j}$, will constitute a Nash equilibrium of \mathcal{G} . Extending this notation, for various arithmetic expressions A involving any x_j^p and u_s^p , vertex $v(A) \in V$ will be used, and be constructed such that in any Nash equilibrium of \mathcal{GG} , $\mathbf{p}[v(A)]$ is equal to A evaluated at the given values of u_s^p and with x_j^p equal to $\mathbf{p}[v(x_j^p)]$. Elements of

W are used to mediate between elements of V , so that the latter ones obey the intended arithmetic relationships.

We use Propositions (1-4) as building blocks of \mathcal{GG} , starting with r subgraphs that represent mixed strategies for the players of \mathcal{G} . In the following, we construct a graphical game containing vertices $\{v(x_j^p)\}_{j \in [n]}$, whose probabilities sum to 1, and internal vertices v_j^p , which control the distribution of the one unit of probability mass among the vertices $v(x_j^p)$. See Figure 8 for an illustration.

Proposition 5 *Consider a graphical game that contains*

- for $j \in [n]$ a vertex $v(x_j^p)$
- for $j \in [n-1]$ a vertex v_j^p
- for $j \in [n]$ a vertex $v(\sum_{i=1}^j x_i^p)$
- for $j \in [n-1]$ a vertex $w_j(p)$ used to ensure $\mathbf{p}[v(\sum_{i=1}^j x_i^p)] = \mathbf{p}[v(\sum_{i=1}^{j+1} x_i^p)](1 - \mathbf{p}[v_j^p])$
- for $j \in [n-1]$ a vertex $w'_j(p)$ used to ensure $\mathbf{p}[v(x_{j+1}^p)] = \mathbf{p}[v(\sum_{i=1}^{j+1} x_i^p)]\mathbf{p}[v_j^p]$
- a vertex $w'_0(p)$ used to ensure $\mathbf{p}[v(x_1^p)] = \mathbf{p}[v(\sum_{i=1}^1 x_i^p)]$

Also, let $v(\sum_{i=1}^n x_i^p)$ have payoff of 1 when it plays 1 and 0 otherwise. Then, in any Nash equilibrium of the graphical game, $\sum_{i=1}^n \mathbf{p}[v(x_i^p)] = 1$ and moreover $\mathbf{p}[v(\sum_{i=1}^j x_i^p)] = \sum_{i=1}^j \mathbf{p}[v(x_i^p)]$, and the graph is bipartite and of degree 3.

Proof. It is not hard to verify that the graph has degree 3. Most of the degree 3 vertices are the w vertices used in Propositions 1 and 2 to connect the pairs or triples of graph players whose probabilities are supposed to obey an arithmetic relationship. In a Nash equilibrium, $v(\sum_{i=1}^n x_i^p)$ plays 1. The vertices v_j^p split the probability $\mathbf{p}[v(\sum_{i=1}^{j+1} x_i^p)]$ between $\mathbf{p}[v(\sum_{i=1}^j x_i^p)]$ and $\mathbf{p}[v(x_{j+1}^p)]$. \square

Comment. The values $\mathbf{p}[v_j^p]$ control the distribution of probability (summing to 1) amongst the n vertices $v(x_j^p)$. These vertices can set to zero any proper subset of the probabilities $\mathbf{p}[v(x_j^p)]$.

Notation. For $s \in S_{-p}$ let $x_s = x_{s_1}^1 \cdot x_{s_2}^2 \cdots x_{s_{p-1}}^{p-1} \cdot x_{s_p}^{p+1} \cdots x_{s_r}^r$. Also, let $U_j^p = \sum_{s \in S_{-p}} u_{j_s}^p x_s$ be the utility to p for playing j in the context of a given mixed profile $\{x_s\}_{s \in S_{-p}}$.

Lemma 8 *Suppose all utilities u_s^p (of \mathcal{G}) lie in the range $[0, 1]$ for some $p \in [r]$. We can construct a degree 3 bipartite graph having a total of $O(rn^r)$ vertices, including vertices $v(x_j^p)$, $v(U_j^p)$, $v(U_{\leq j}^p)$, for all $j \in [n]$, such that in any Nash equilibrium,*

$$\mathbf{p}[v(U_j^p)] = \sum_{s \in S_{-p}} u_{j_s}^p \prod_{q \neq p} \mathbf{p}[v(x_{s_q}^q)], \quad (11)$$

$$\mathbf{p}[v(U_{\leq j}^p)] = \max_{i \leq j} \sum_{s \in S_{-p}} u_{i_s}^p \prod_{q \neq p} \mathbf{p}[v(x_{s_q}^q)]. \quad (12)$$

The vertices whose labels include U do not form part of Proposition 5; they have been included to show how the gadget fits into the rest of the construction, as described in Figure 9. Unshaded vertices belong to V , shaded vertices belong to W (V and W being the two parts of the bipartite graph). A directed edge from u to v indicates that u 's choice can affect v 's payoff.

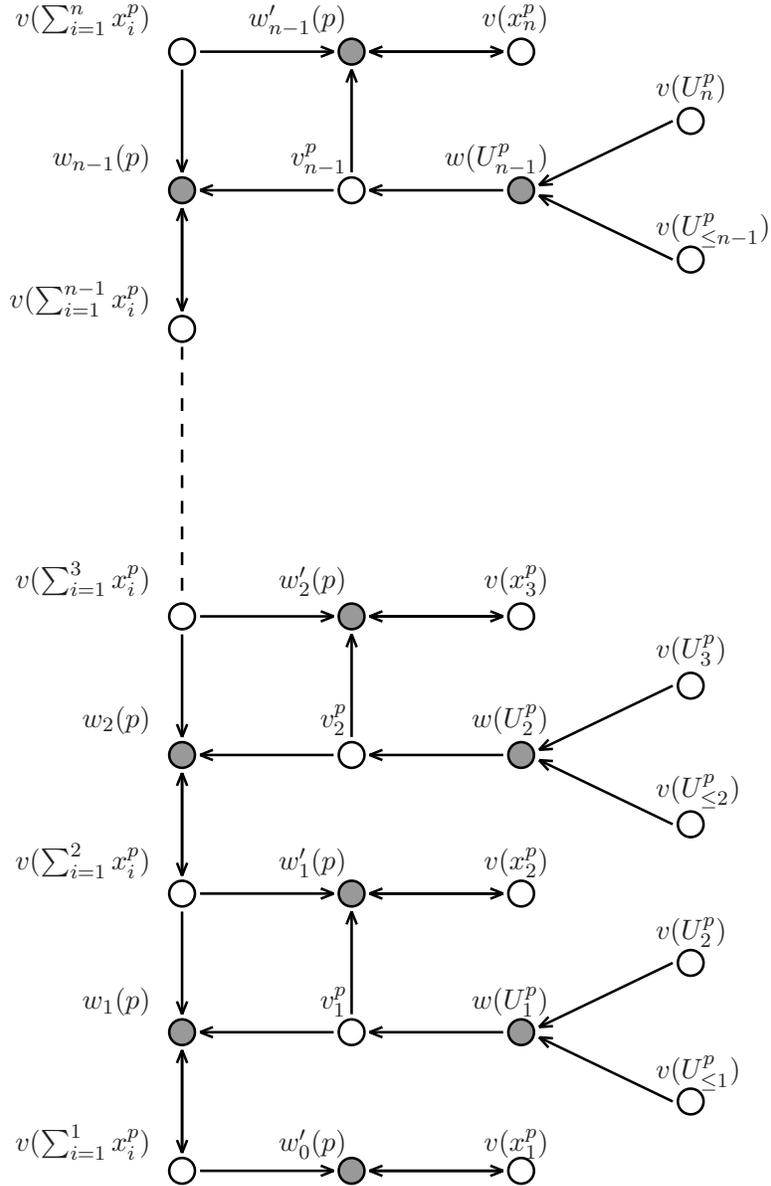


Figure 8: Diagram of Proposition 5

The general idea is to note that the expressions for $\mathbf{p}[v(U_j^p)]$ and $\mathbf{p}[v(U_{\leq j}^p)]$ are constructed from arithmetic subexpressions using the operations of addition, multiplication and maximization. If each subexpression A has a vertex $v(A)$, then using Propositions 1 through 4 we can assemble them into a graphical game such that in any Nash equilibrium, $\mathbf{p}[v(A)]$ is equal to the value of A with input $\mathbf{p}[v(x_j^p)]$, $p \in [r]$, $j \in [n]$. We just need to limit our usage to $O(rn^r)$ subexpressions and ensure that their values all lie in $[0, 1]$.

Proof. Note that

$$U_{\leq j}^p = \max\{U_j^p, U_{\leq j-1}^p\}, \quad U_j^p = \sum_{s \in S_{-p}} u_{j_s}^p x_s = \sum_{s \in S_{-p}} u_{j_s}^p x_{s_1}^{p-1} \cdots x_{s_{p-1}}^{p-1} x_{s_{p+1}}^{p+1} \cdots x_{s_r}^r.$$

Let $S_{-p} = \{S_{-p}(1), \dots, S_{-p}(n^{r-1})\}$, so that

$$\sum_{s \in S_{-p}} u_{j_s}^p x_s = \sum_{\ell=1}^{n^{r-1}} u_{j_{S_{-p}(\ell)}}^p x_{S_{-p}(\ell)}.$$

For each partial sum $\sum_{\ell=1}^z u_{j_{S_{-p}(\ell)}}^p x_{S_{-p}(\ell)}$, $1 \leq z \leq n^{r-1}$, include vertex $v(\sum_{\ell=1}^z u_{j_{S_{-p}(\ell)}}^p x_{S_{-p}(\ell)})$. Similarly, for each partial product of the summands $u_{j_s}^p \prod_{p \neq q \leq z} x_{s_q}^q$, $0 \leq z \leq r$, include vertex $v(u_{j_s}^p \prod_{p \neq q \leq z} x_{s_q}^q)$. So, for each strategy $j \in S_p$, there are n^{r-1} partial sums and $r + 1$ partial products for each summand. Then, there are n partial sequences over which we have to maximize. Note that, since all utilities are assumed to lie in the set $[0, 1]$, all partial sums and products must also lie in $[0, 1]$, so the truncation at 1 in the computations of Propositions 1, 2, 3 and 4 is not a problem. So using a vertex for each of the $2n + (r + 1)n^r$ arithmetic subexpressions, a Nash equilibrium will compute the desired quantities. \square

We repeat the construction specified by Lemma 8 for all $p \in [r]$. Note that, to avoid large degrees in the resulting graphical game, each time we need to make use of a value $x_{s_q}^q$ we create a new copy of the vertex $v(x_{s_q}^q)$ using the gadget $\mathcal{G}_=$ and, then, use the new copy for the computation of the desired partial product; an easy calculation shows that we have to make $(r - 1)n^{r-1}$ copies of $v(x_{s_q}^q)$, for all $q \leq r$, $s_q \in S_q$. To limit the degree of each vertex to 3 we create a binary tree of copies of $v(x_{s_q}^q)$ with $(r - 1)n^{r-1}$ leaves and use each leaf once.

Proof of Theorem 6: Let \mathcal{G} be a r -player normal-form game with n strategies per player and construct $\mathcal{GG} = f(\mathcal{G})$ as shown in Figure 9. The graph of \mathcal{GG} has degree 3, by the graph structure of our gadgets from Propositions 1 through 4 and the fact that we use separate copies of the $v(x_j^p)$ vertices to influence different $v(U_j^p)$ vertices (see Step 4 and discussion after Lemma 8).

Polynomial size of $\mathcal{GG} = f(\mathcal{G})$:

The size of \mathcal{GG} is polynomial in the description length $r \cdot n^r q$ of \mathcal{G} , where q is the size of the values in the payoff tables in the logarithmic cost model.

Construction of $g(N_{\mathcal{GG}})$ (where $N_{\mathcal{GG}}$ denotes a Nash equilibrium of \mathcal{GG}):

Given a Nash equilibrium $g(N_{\mathcal{GG}})$ of \mathcal{GG} , we claim that we can recover a Nash equilibrium $\{x_j^p\}_{p,j}$ of \mathcal{G} by taking $x_j^p = \mathbf{p}[v(x_j^p)]$. This is clearly computable in polynomial-time.

Proof that the reduction preserves Nash equilibria:

Call \mathcal{G}' the game resulting from \mathcal{G} by rescaling the utilities so that they lie in the range $[0, 1]$. It is easy to see that any Nash equilibrium of game \mathcal{G} is, also, a Nash equilibrium of game \mathcal{G}' and vice versa. Therefore, it is enough to establish that the mapping $g(\cdot)$ maps every Nash equilibrium

Input: Normal form game \mathcal{G} with r players, n strategies per player, utilities $\{u_s^p : p \in [r], s \in S\}$.

Output: Graphical game \mathcal{GG} with bipartite graph $(V \cup W, E)$.

1. If needed, rescale the utilities u_s^p so that they lie in the range $[0, 1]$. One way to do so is to divide all utilities by $\max\{u_s^p\}$.
2. For each player/strategy pair (p, j) let $v(x_j^p) \in V$ be a vertex in \mathcal{GG} .
3. For each $p \in [r]$ construct a subgraph as described in Proposition 5 so that in a Nash equilibrium of \mathcal{GG} , we have $\sum_j \mathbf{p}[v(x_j^p)] = 1$.
4. Use the construction of Proposition 1 with $\alpha = 1$ to make $(r-1)n^{r-1}$ copies of the $v(x_j^p)$ vertices (which are added to V). More precisely, create a binary tree with copies of $v(x_j^p)$ which has $(r-1)n^{r-1}$ leaves.
5. Use the construction of Lemma 8 to introduce (add to V) vertices $v(U_j^p), v(U_{\leq j}^p)$, for all $p \in [r], j \in [n]$. Each $v(U_j^p)$ uses its own set of copies of the vertices $v(x_j^p)$. For $p \in [r], j \in [n]$ introduce (add to W) $w(U_j^p)$ with
 - (a) If $w(U_j^p)$ plays 0 then $w(U_j^p)$ gets payoff 1 whenever $v(U_{\leq j}^p)$ plays 1, else 0.
 - (b) If $w(U_j^p)$ plays 1 then $w(U_j^p)$ gets payoff 1 whenever $v(U_{j+1}^p)$ plays 1, else 0.
6. Give the following payoffs to the vertices v_j^p (the additional vertices used in Proposition 5 whose payoffs were not specified).
 - (a) If v_j^p plays 0 then v_j^p has a payoff of 1 whenever $w(U_j^p)$ plays 0, otherwise 0.
 - (b) If v_j^p plays 1 then v_j^p has a payoff of 1 whenever $w(U_j^p)$ plays 1, otherwise 0.
7. Return the underlying undirected graphical game \mathcal{GG} .

Figure 9: Reduction from normal form game \mathcal{G} to graphical game \mathcal{GG}

of game \mathcal{GG} to a Nash equilibrium of game \mathcal{G}' . By Proposition 5, we have that $\sum_j x_j^p = 1$, for all $p \in [r]$. It remains to show that, for all p, j, j' ,

$$\sum_{s \in S_{-p}} u_{j's}^p x_s > \sum_{s \in S_{-p}} u_{j''s}^p x_s \implies x_{j'}^p = 0.$$

We distinguish the cases:

- If there exists some $j'' < j'$ such that $\sum_{s \in S_{-p}} u_{j''s}^p x_s > \sum_{s \in S_{-p}} u_{j's}^p x_s$, then, by Lemma 8, $\mathbf{p}[v(U_{\leq j'-1}^p)] > \mathbf{p}[v(U_{j'}^p)]$. Thus, $\mathbf{p}[v_{j'-1}^p] = 0$ and, consequently, $v(x_{j'}^p)$ plays 0 as required, since

$$\mathbf{p}[v(x_{j'}^p)] = \mathbf{p}[v_{j'-1}^p] \mathbf{p} \left[v \left(\sum_{i=1}^{j'} x_i^p \right) \right].$$

- The case $j < j'$ reduces trivially to the previous case.
- It remains to deal with the case $j > j'$, under the assumption that, for all $j'' < j'$, $\sum_{s \in S_{-p}} u_{j''s}^p x_s \leq$

$\sum_{s \in S_{-p}} u_{j's}^p x_s$, or, equivalently,

$$\mathbf{p}[v(U_{j''}^p)] \leq \mathbf{p}[v(U_{j'}^p)],$$

which in turn implies that

$$\mathbf{p}[v(U_{\leq j'}^p)] \leq \mathbf{p}[v(U_{j'}^p)].$$

It follows that there exists some k , $j' + 1 \leq k \leq j$, such that $\mathbf{p}[v(U_k^p)] > \mathbf{p}[v(U_{\leq k-1}^p)]$. Otherwise, $\mathbf{p}[v(U_{\leq j'}^p)] \geq \mathbf{p}[v(U_{\leq j'+1}^p)] \geq \dots \geq \mathbf{p}[v(U_{\leq j}^p)] \geq \mathbf{p}[v(U_j^p)] > \mathbf{p}[v(U_{j'}^p)]$, which is a contradiction to $\mathbf{p}[v(U_{\leq j'}^p)] \leq \mathbf{p}[v(U_{j'}^p)]$. Since $\mathbf{p}[v(U_k^p)] > \mathbf{p}[v(U_{\leq k-1}^p)]$, it follows that $\mathbf{p}[w(U_{k-1}^p)] = 1 \Rightarrow \mathbf{p}[v_{k-1}^p] = 1$ and, therefore,

$$\begin{aligned} \mathbf{p} \left[v \left(\sum_{i=1}^{k-1} x_i^p \right) \right] &= \mathbf{p} \left[v \left(\sum_{i=1}^k x_i^p \right) \right] (1 - \mathbf{p}[v_{k-1}^p]) = 0 \\ \Rightarrow \mathbf{p} \left[v \left(\sum_{i=1}^{j'} x_i^p \right) \right] &= 0 \Rightarrow \mathbf{p} [v(x_{j'}^p)] = 0. \end{aligned}$$

Mapping g is surjective on the Nash equilibria of \mathcal{G}' and, therefore, \mathcal{G} : We will show that given a Nash equilibrium $N_{\mathcal{G}'}$ of \mathcal{G}' there is a Nash equilibrium $N_{\mathcal{G}\mathcal{G}}$ of $\mathcal{G}\mathcal{G}$ such that $g(N_{\mathcal{G}\mathcal{G}}) = N_{\mathcal{G}'}$. Let $N_{\mathcal{G}'} = \{x_j^p : p \leq r, j \in S_p\}$. In $N_{\mathcal{G}\mathcal{G}}$, let $\mathbf{p}[v(x_j^p)] = x_j^p$. Lemma 8 shows that the values $\mathbf{p}[v(U_j^p)]$ are the expected utilities to player p for playing strategy j , given that all other players use the mixed strategy $\{x_j^p : p \leq r, j \in S_p\}$. We identify values for $\mathbf{p}[v_j^p]$ that complete a Nash equilibrium for $\mathcal{G}\mathcal{G}$.

Based on the payoffs to v_j^p described in Figure 9 we have

- If $\mathbf{p}[v(U_{\leq j}^p)] > \mathbf{p}[v(U_{j+1}^p)]$ then $\mathbf{p}[w(U_j^p)] = 0$; $\mathbf{p}[v_j^p] = 0$;
- If $\mathbf{p}[v(U_{\leq j}^p)] < \mathbf{p}[v(U_{j+1}^p)]$ then $\mathbf{p}[w(U_j^p)] = 1$; $\mathbf{p}[v_j^p] = 1$;
- If $\mathbf{p}[v(U_{\leq j}^p)] = \mathbf{p}[v(U_{j+1}^p)]$ then choose $\mathbf{p}[w(U_j^p)] = \frac{1}{2}$; $\mathbf{p}[v_j^p]$ is arbitrary (we may assign it any value)

Given the above constraints on the values $\mathbf{p}[v_j^p]$ we must check that we can choose them (and there is a unique choice) so as to make them consistent with the probabilities $\mathbf{p}[v(x_j^p)]$. We use the fact the values x_j^p form a Nash equilibrium of \mathcal{G} . In particular, we know that $\mathbf{p}[v(x_j^p)] = 0$ if there exists j' with $U_{j'}^p > U_j^p$. We claim that for j satisfying $\mathbf{p}[v(U_{\leq j}^p)] = \mathbf{p}[v(U_{j+1}^p)]$, if we choose

$$\mathbf{p}[v_j^p] = \mathbf{p}[v(x_{j+1}^p)] / \sum_{i=1}^{j+1} \mathbf{p}[v(x_i^p)],$$

then the values $\mathbf{p}[v(x_j^p)]$ are consistent. \square

4.4 Combining the Reductions

Suppose that we take either a graphical or a normal-form game, and apply to it both of the reductions described in the previous sections. Then we obtain a game of the same type and a surjective mapping from the Nash equilibria of the latter to the Nash equilibria of the former.

Corollary 2 *For any fixed d , a (directed or undirected) graphical game of maximum degree d can be mapped in polynomial time to an undirected graphical game of maximum degree 3 so that there is a polynomial-time computable surjective mapping g from the Nash equilibria of the latter to the Nash equilibria of the former.*

The following also follows directly from Theorems 6 and 5, but is not as strong as Theorem 7 below.

Corollary 3 *For any fixed $r > 1$, a r -player normal form game can be mapped in polynomial time to a 10-player normal form game so that there is a polynomial-time computable surjective mapping g from the Nash equilibria of the latter to the Nash equilibria of the former.*

Proof. Theorem 6 converts a r -player game \mathcal{G} into a graphical game \mathcal{GG} based on a graph of degree 3. Theorem 5 converts \mathcal{GG} to a 10-player game \mathcal{G}' , whose Nash equilibria encode the Nash equilibria of \mathcal{GG} and hence of \mathcal{G} . (Note that for d an odd number, the proof of Theorem 5 implies a reduction to a $(d^2 + 1)$ -player normal form game.) \square

We next prove a stronger result, by exploiting in more detail the structure of the graphical games \mathcal{GG} constructed in the proof of Theorem 6. The technique used here will be used in Section 4.5 to strengthen the result even further.

Theorem 7 *For any fixed $r > 1$, a r -player normal form game can be mapped in polynomial time to a 4-player normal form game so that there is a polynomial-time computable surjective mapping g from the Nash equilibria of the latter to the Nash equilibria of the former.*

Proof. Construct \mathcal{G}' from \mathcal{G} as shown in Figure 10.

Polynomial size of $\mathcal{G}' = f(\mathcal{G})$.

By Theorem 6, \mathcal{GG} (as constructed in Figure 10) is of polynomial size. The size of \mathcal{GG}' is at most 3 times the size of \mathcal{GG} since we do not need to apply Step 3 to any edges that are themselves constructed by an earlier iteration of Step 3. Finally, the size of \mathcal{G}' is polynomial in the size of \mathcal{GG}' from Theorem 5.

Construction of $g(N_{\mathcal{G}'})$ (for $N_{\mathcal{G}'}$ a Nash equilibrium of \mathcal{G}').

Let g_1 be a surjective mapping from the Nash equilibria of \mathcal{GG} to the Nash equilibria of \mathcal{G} , which is guaranteed to exist by Theorem 6. It is trivial to construct a surjective mapping g_2 from the Nash equilibria of \mathcal{GG}' to the Nash equilibria of \mathcal{GG} . By Theorem 5, there exists a surjective mapping g_3 from the Nash equilibria of \mathcal{G}' to the Nash equilibria of \mathcal{GG}' . Therefore, $g_3 \circ g_2 \circ g_1$ is a surjective mapping from the Nash equilibria of \mathcal{G}' to the Nash equilibria of \mathcal{G} . \square

4.5 Reducing to Three Players

We will strengthen Theorem 7 to reduce a r -player normal form game to a 3-player normal form game. The following theorem together with Theorems 5 and 6 imply the first part of Theorem 4.

Theorem 8 *For any fixed $r > 1$, a r -player normal form game can be mapped in polynomial time to a 3-player normal form game so that there is a polynomial-time computable surjective mapping g from the Nash equilibria of the latter to the Nash equilibria of the former.*

Proof. The bottleneck of the construction of Figure 10 in terms of the number k of players of the resulting normal form game \mathcal{G}' lies entirely on the ability or lack thereof to color the vertices of the affects graphs of \mathcal{GG} with k colors so that, for every vertex v , its neighborhood $\mathcal{N}(v)$ in the

Input: Normal form game \mathcal{G} with r players, n strategies per player, utilities $\{u_s^p : p \leq r, s \in S\}$.

Output: 4-player Normal form game \mathcal{G}' .

1. Let \mathcal{GG} be the graphical game constructed from \mathcal{G} according to Figure 9. Recall that the affects graph $G = (V \cup W, E)$ of \mathcal{GG} has the following properties:
 - Every edge $e \in E$ is from a vertex of set V to a vertex of set W or vice versa.
 - Every vertex of set W has indegree at most 3 and outdegree at most 1 and every vertex of set V has indegree at most 1 and outdegree at most 2.
2. Color the graph $(V \cup W, E)$ of \mathcal{GG} as follows: let $c(w) = 1$ for all W -vertices w and $c(v) = 2$ for all V -vertices v .
3. Construct a new graphical game \mathcal{GG}' from \mathcal{GG} as follows. While there exist $v_1, v_2 \in V, w \in W, (v_1, w), (v_2, w) \in E$ with $c(v_1) = c(v_2)$:
 - (a) Every W -vertex has at most 1 outgoing edge, so assume $(w, v_1) \notin E$.
 - (b) Add $v(v_1)$ to V , add $w(v_1)$ to W .
 - (c) Replace (v_1, w) with $(v_1, w(v_1)), (w(v_1), v(v_1)), (v(v_1), w(v_1)), (v(v_1), w)$. Let $c(w(v_1)) = 1$, choose $c(v(v_1)) \in \{2, 3, 4\} \neq c(v')$ for any v' with $(v', w) \in E$. Payoffs for $w(v_1)$ and $v(v_1)$ are chosen using Proposition 1 with $\alpha = 1$ such that in any Nash equilibrium, $\mathbf{p}[v(v_1)] = \mathbf{p}[v_1]$.
4. The coloring $c : V \cup W \rightarrow \{1, 2, 3, 4\}$ has the property that, for every vertex v of \mathcal{GG}' , its neighborhood $\mathcal{N}(v)$ in the affects graph of the game —recall it consists of v and all its predecessors— is colored with $|\mathcal{N}(v)|$ distinct colors. Rescale all utilities of \mathcal{GG}' to $[0,1]$ and map game \mathcal{GG}' to a 4-player normal form game \mathcal{G}' following the steps 3 through 5 of figure 7.

Figure 10: Reduction from normal form game \mathcal{G} to 4-player game \mathcal{G}'

affects graph is colored with $|\mathcal{N}(v)|$ distinct colors, i.e. on whether there exists a legal k coloring. In Figure 10, we show how to design a graphical game \mathcal{GG}' which is equivalent to \mathcal{GG} —in the sense that there exists a surjective mapping from the Nash equilibria of the former to the Nash equilibria of the latter— and can be legally colored using 4 colors. However, this cannot be improved to 3 colors since the addition game \mathcal{G}_+ and the multiplication game \mathcal{G}_* , which are essential building blocks of \mathcal{GG} , have vertices with indegree 3 (see Figure 5) and, therefore, need at least 4 colors to be legally colored. Therefore, to improve our result we need to redesign addition and multiplication games which can be legally colored using 3 colors.

Notation: In the following,

- $x = y \pm \epsilon$ denotes $y - \epsilon \leq x \leq y + \epsilon$
- $v : s$ denotes “player v plays strategy s ”

Proposition 6 *Let α, β, γ be non-negative integers such that $\alpha + \beta + \gamma \leq 3$. There is a graphical game $\mathcal{G}_{+,*}$ with two “input players” v_1 and v_2 , one “output player” v_3 and several intermediate players, with the following properties:*

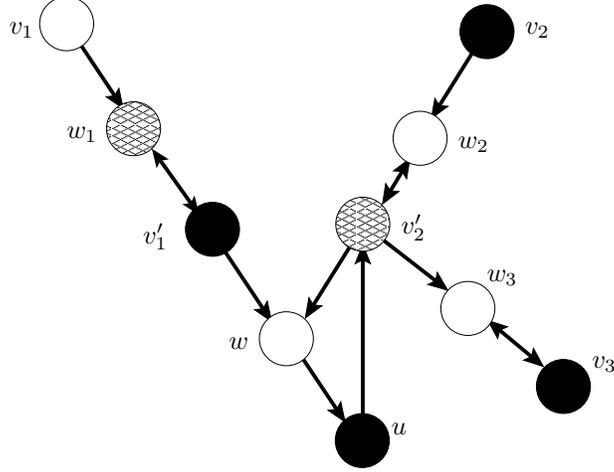


Figure 11: The new addition/multiplication game and its legal 3-coloring.

- the graph of the game can be legally colored using 3 colors
- for any $\epsilon \in [0, 0.01]$, at any ϵ -Nash equilibrium of game $\mathcal{G}_{+,*}$ it holds that $\mathbf{p}[v_3] = \min\{1, \alpha\mathbf{p}[v_1] + \beta\mathbf{p}[v_2] + \gamma\mathbf{p}[v_1]\mathbf{p}[v_2]\} \pm 81\epsilon$; in particular at any Nash equilibrium $\mathbf{p}[v_3] = \min\{1, \alpha\mathbf{p}[v_1] + \beta\mathbf{p}[v_2] + \gamma\mathbf{p}[v_1]\mathbf{p}[v_2]\}$.

Proof. The graph of the game and the labeling of the vertices is shown in Figure 11. All players of $\mathcal{G}_{+,*}$ have strategy set $\{0, 1\}$ except for player v_2' who has three strategies $\{0, 1, *\}$. Below we give the payoff tables of all the players of the game. For ease of understanding we partition the game $\mathcal{G}_{+,*}$ into four subgames:

1. Game played by players v_1, w_1, v_1' :

Payoffs to v_1' :	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding: 5px;"></td> <td style="padding: 5px;">$w_1 : 0$</td> <td style="padding: 5px;">$w_1 : 1$</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">$v_1' : 0$</td> <td style="padding: 5px;">0</td> <td style="padding: 5px;">1</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">$v_1' : 1$</td> <td style="padding: 5px;">1</td> <td style="padding: 5px;">0</td> </tr> </table>		$w_1 : 0$	$w_1 : 1$	$v_1' : 0$	0	1	$v_1' : 1$	1	0
	$w_1 : 0$	$w_1 : 1$								
$v_1' : 0$	0	1								
$v_1' : 1$	1	0								

Payoffs to w_1 :

$w_1 : 0$:	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding: 5px;"></td> <td style="padding: 5px;">$v_1' : 0$</td> <td style="padding: 5px;">$v_1' : 1$</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">$v_1 : 0$</td> <td style="padding: 5px;">0</td> <td style="padding: 5px;">0</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">$v_1 : 1$</td> <td style="padding: 5px;">1/8</td> <td style="padding: 5px;">1/8</td> </tr> </table>		$v_1' : 0$	$v_1' : 1$	$v_1 : 0$	0	0	$v_1 : 1$	1/8	1/8	$w_1 : 1$:	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding: 5px;"></td> <td style="padding: 5px;">$v_1' : 0$</td> <td style="padding: 5px;">$v_1' : 1$</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">$v_1 : 0$</td> <td style="padding: 5px;">0</td> <td style="padding: 5px;">1</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">$v_1 : 1$</td> <td style="padding: 5px;">0</td> <td style="padding: 5px;">1</td> </tr> </table>		$v_1' : 0$	$v_1' : 1$	$v_1 : 0$	0	1	$v_1 : 1$	0	1
	$v_1' : 0$	$v_1' : 1$																			
$v_1 : 0$	0	0																			
$v_1 : 1$	1/8	1/8																			
	$v_1' : 0$	$v_1' : 1$																			
$v_1 : 0$	0	1																			
$v_1 : 1$	0	1																			

2. Game played by players v_2', w_3, v_3 :

Payoffs to v_3 :	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding: 5px;"></td> <td style="padding: 5px;">$w_3 : 0$</td> <td style="padding: 5px;">$w_3 : 1$</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">$v_3 : 0$</td> <td style="padding: 5px;">0</td> <td style="padding: 5px;">1</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">$v_3 : 1$</td> <td style="padding: 5px;">1</td> <td style="padding: 5px;">0</td> </tr> </table>		$w_3 : 0$	$w_3 : 1$	$v_3 : 0$	0	1	$v_3 : 1$	1	0
	$w_3 : 0$	$w_3 : 1$								
$v_3 : 0$	0	1								
$v_3 : 1$	1	0								

Payoffs to w_3 :

$w_3 : 0$:	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding: 5px;"></td> <td style="padding: 5px;">$v_3 : 0$</td> <td style="padding: 5px;">$v_3 : 1$</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">$v_2' : 0$</td> <td style="padding: 5px;">0</td> <td style="padding: 5px;">0</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">$v_2' : 1$</td> <td style="padding: 5px;">0</td> <td style="padding: 5px;">0</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">$v_2' : *$</td> <td style="padding: 5px;">8</td> <td style="padding: 5px;">8</td> </tr> </table>		$v_3 : 0$	$v_3 : 1$	$v_2' : 0$	0	0	$v_2' : 1$	0	0	$v_2' : *$	8	8	$w_3 : 1$:	<table style="border-collapse: collapse;"> <tr> <td style="border-right: 1px solid black; padding: 5px;"></td> <td style="padding: 5px;">$v_3 : 0$</td> <td style="padding: 5px;">$v_3 : 1$</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">$v_2' : 0$</td> <td style="padding: 5px;">0</td> <td style="padding: 5px;">1</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">$v_2' : 1$</td> <td style="padding: 5px;">0</td> <td style="padding: 5px;">1</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 5px;">$v_2' : *$</td> <td style="padding: 5px;">0</td> <td style="padding: 5px;">1</td> </tr> </table>		$v_3 : 0$	$v_3 : 1$	$v_2' : 0$	0	1	$v_2' : 1$	0	1	$v_2' : *$	0	1
	$v_3 : 0$	$v_3 : 1$																									
$v_2' : 0$	0	0																									
$v_2' : 1$	0	0																									
$v_2' : *$	8	8																									
	$v_3 : 0$	$v_3 : 1$																									
$v_2' : 0$	0	1																									
$v_2' : 1$	0	1																									
$v_2' : *$	0	1																									

3. Game played by players v_2, w_2, v'_2 :

Payoffs to w_2 :

		$v_2 : 0$	$v_2 : 1$			$v_2 : 0$	$v_2 : 1$
$w_2 : 0 :$	$v'_2 : 0$	0	1/8	$w_2 : 1 :$	$v'_2 : 0$	0	0
	$v'_2 : 1$	0	1/8		$v'_2 : 1$	1	1
	$v'_2 : *$	0	1/8		$v'_2 : *$	0	0

Payoffs to v'_2 :

		$w_2 : 0$	$w_2 : 1$			$w_2 : 0$	$w_2 : 1$
$v'_2 : 0 :$	$u : 0$	0	1	$v'_2 : 1 :$	$u : 0$	1	0
	$u : 1$	0	0		$u : 1$	1	0
		$w_2 : 0$ $w_2 : 1$				$w_2 : 0$ $w_2 : 1$	
		$u : 0$	0	$u : 0$	0		
		$u : 1$	0	$u : 1$	1		

4. Game played by players v'_1, v'_2, w, u :

Payoffs to w :

		$v'_1 : 0$	$v'_1 : 1$			$v'_1 : 0$	$v'_1 : 1$
$w : 0 :$	$v'_2 : 0$	0	α	$w : 1 :$	$v'_2 : 0$	0	0
	$v'_2 : 1$	$1 + \beta$	$1 + \alpha + \beta + 8\gamma$		$v'_2 : 1$	1	1
	$v'_2 : *$	0	α		$v'_2 : *$	1	1

Payoffs to u :

		$w : 0$	$w : 1$
$u : 0$	0	1	
$u : 1$	1	0	

Claim 1 At any ϵ -Nash equilibrium of $\mathcal{G}_{+,*}$: $\mathbf{p}[v'_1] = \frac{1}{8}\mathbf{p}[v_1] \pm \epsilon$.

Proof. If w_1 plays 0, then the expected payoff to w_1 is $\frac{1}{8}\mathbf{p}[v_1]$, whereas if w_1 plays 1, the expected payoff to w_1 is $\mathbf{p}[v'_1]$. Therefore, in an ϵ -Nash equilibrium, if $\frac{1}{8}\mathbf{p}[v_1] > \mathbf{p}[v'_1] + \epsilon$, then $\mathbf{p}[w_1] = 0$. However, note also that if $\mathbf{p}[w_1] = 0$ then $\mathbf{p}[v'_1] = 1$, which is a contradiction to $\frac{1}{8}\mathbf{p}[v_1] > \mathbf{p}[v'_1] + \epsilon$. Consequently, $\frac{1}{8}\mathbf{p}[v_1]$ cannot be strictly larger than $\mathbf{p}[v'_1] + \epsilon$. On the other hand, if $\mathbf{p}[v'_1] > \frac{1}{8}\mathbf{p}[v_1] + \epsilon$, then $\mathbf{p}[w_1] = 1$ and consequently $\mathbf{p}[v'_1] = 0$, a contradiction. The claim follows from the above observations. \square

Claim 2 At any ϵ -Nash equilibrium of $\mathcal{G}_{+,*}$: $\mathbf{p}[v'_2 : 1] = \frac{1}{8}\mathbf{p}[v_2] \pm \epsilon$.

Proof. If w_2 plays 0, then the expected payoff to w_2 is $\frac{1}{8}\mathbf{p}[v_2]$, whereas, if w_2 plays 1, the expected payoff to w_2 is $\mathbf{p}[v'_2 : 1]$.

If, in an ϵ -Nash equilibrium, $\frac{1}{8}\mathbf{p}[v_2] > \mathbf{p}[v'_2 : 1] + \epsilon$, then $\mathbf{p}[w_2] = 0$. In this regime, the payoff to player v'_2 is 0 if v'_2 plays 0, 1 if v'_2 plays 1 and 0 if v'_2 plays *. Therefore, $\mathbf{p}[v'_2 : 1] = 1$ and this contradicts the hypothesis that $\frac{1}{8}\mathbf{p}[v_2] > \mathbf{p}[v'_2 : 1] + \epsilon$.

On the other hand, if, in an ϵ -Nash equilibrium, $\mathbf{p}[v'_2 : 1] > \frac{1}{8}\mathbf{p}[v_2] + \epsilon$, then $\mathbf{p}[w_2] = 1$. In this regime, the payoff to player v'_2 is $\mathbf{p}[u : 0]$ if v'_2 plays 0, 0 if v'_2 plays 1 and $\mathbf{p}[u : 1]$ if v'_2 plays *. Since $\mathbf{p}[u : 0] + \mathbf{p}[u : 1] = 1$, it follows that $\mathbf{p}[v'_2 : 1] = 0$ because at least one of $\mathbf{p}[u : 0]$, $\mathbf{p}[u : 1]$ will be greater than ϵ . This contradicts the hypothesis that $\mathbf{p}[v'_2 : 1] > \frac{1}{8}\mathbf{p}[v_2] + \epsilon$ and the claim follows from the above observations. \square

Claim 3 *At any ϵ -Nash equilibrium of $\mathcal{G}_{+,*}$: $\mathbf{p}[v'_2 : *] = \frac{\alpha}{8}\mathbf{p}[v_1] + \frac{\beta}{8}\mathbf{p}[v_2] + \frac{\gamma}{8}\mathbf{p}[v_1]\mathbf{p}[v_2] \pm 10\epsilon$.*

Proof. If w plays 0, then the expected payoff to w is $\alpha\mathbf{p}[v'_1] + (1 + \beta)\mathbf{p}[v'_2 : 1] + 8\gamma\mathbf{p}[v'_1]\mathbf{p}[v'_2 : 1]$, whereas, if w plays 1, the expected payoff to w is $\mathbf{p}[v'_2 : 1] + \mathbf{p}[v'_2 : *]$.

If, in a ϵ -Nash equilibrium, $\alpha\mathbf{p}[v'_1] + (1 + \beta)\mathbf{p}[v'_2 : 1] + 8\gamma\mathbf{p}[v'_1]\mathbf{p}[v'_2 : 1] > \mathbf{p}[v'_2 : 1] + \mathbf{p}[v'_2 : *] + \epsilon$, then $\mathbf{p}[w] = 0$ and, consequently, $\mathbf{p}[u] = 1$. In this regime, the payoff to player v'_2 is 0 if v'_2 plays 0, $\mathbf{p}[w_2 : 0]$ if v'_2 plays 1 and $\mathbf{p}[w_2 : 1]$ if v'_2 plays *. Since $\mathbf{p}[w_2 : 0] + \mathbf{p}[w_2 : 1] = 1$, it follows that at least one of $\mathbf{p}[w_2 : 0]$, $\mathbf{p}[w_2 : 1]$ will be larger than ϵ so that $\mathbf{p}[v'_2 : 0] = 0$ or, equivalently, that $\mathbf{p}[v'_2 : 1] + \mathbf{p}[v'_2 : *] = 1$. So the hypothesis can be rewritten as $\alpha\mathbf{p}[v'_1] + (1 + \beta)\mathbf{p}[v'_2 : 1] + 8\gamma\mathbf{p}[v'_1]\mathbf{p}[v'_2 : 1] > 1 + \epsilon$. Using Claims 1 and 2 and the fact that $\epsilon \leq 0.01$ this inequality implies $\frac{\alpha}{8}\mathbf{p}[v_1] + \frac{1+\beta}{8}\mathbf{p}[v_2] + \frac{\gamma}{8}\mathbf{p}[v_1]\mathbf{p}[v_2] + (\alpha + 1 + \beta + 3\gamma)\epsilon > 1 + \epsilon$ and further that $\frac{\alpha+1+\beta+\gamma}{8} + (\alpha + 1 + \beta + 3\gamma)\epsilon > 1 + \epsilon$. We supposed $\alpha + \beta + \gamma \leq 3$ therefore the previous inequality implies $\frac{1}{2} + 10\epsilon > 1 + \epsilon$, a contradiction since we assumed $\epsilon \leq 0.01$.

On the other hand, if, in a ϵ -Nash equilibrium, $\mathbf{p}[v'_2 : 1] + \mathbf{p}[v'_2 : *] > \alpha\mathbf{p}[v'_1] + (1 + \beta)\mathbf{p}[v'_2 : 1] + 8\gamma\mathbf{p}[v'_1]\mathbf{p}[v'_2 : 1] + \epsilon$, then $\mathbf{p}[w] = 1$ and consequently $\mathbf{p}[u] = 0$. In this regime, the payoff to player v'_2 is $\mathbf{p}[w_2 : 1]$ if v'_2 plays 0, $\mathbf{p}[w_2 : 0]$ if v'_2 plays 1 and 0 if v'_2 plays *. Since $\mathbf{p}[w_2 : 0] + \mathbf{p}[w_2 : 1] = 1$, it follows that $\mathbf{p}[v'_2 : *] = 0$. So the hypothesis can be rewritten as $0 > \alpha\mathbf{p}[v'_1] + \beta\mathbf{p}[v'_2 : 1] + 8\gamma\mathbf{p}[v'_1]\mathbf{p}[v'_2 : 1] + \epsilon$ which is a contradiction.

Therefore, in any ϵ -Nash equilibrium, $\mathbf{p}[v'_2 : 1] + \mathbf{p}[v'_2 : *] = \alpha\mathbf{p}[v'_1] + (1 + \beta)\mathbf{p}[v'_2 : 1] + 8\gamma\mathbf{p}[v'_1]\mathbf{p}[v'_2 : 1] \pm \epsilon$, or, equivalently, $\mathbf{p}[v'_2 : *] = \alpha\mathbf{p}[v'_1] + \beta\mathbf{p}[v'_2 : 1] + 8\gamma\mathbf{p}[v'_1]\mathbf{p}[v'_2 : 1] \pm \epsilon$. Using claims 1 and 2 this can be restated as $\mathbf{p}[v'_2 : *] = \frac{\alpha}{8}\mathbf{p}[v_1] + \frac{\beta}{8}\mathbf{p}[v_2] + \frac{\gamma}{8}\mathbf{p}[v_1]\mathbf{p}[v_2] \pm 10\epsilon$ \square

Claim 4 *At any ϵ -Nash equilibrium of $\mathcal{G}_{+,*}$: $\mathbf{p}[v_3] = \min\{1, \alpha\mathbf{p}[v_1] + \beta\mathbf{p}[v_2] + \gamma\mathbf{p}[v_1]\mathbf{p}[v_2]\} \pm 81\epsilon$.*

Proof. If w_3 plays 0, the expected payoff to w_3 is $8\mathbf{p}[v'_2 : *]$, whereas, if w_3 plays 1, the expected payoff to w_3 is $\mathbf{p}[v_3]$. Therefore, in a ϵ -Nash equilibrium, if $\mathbf{p}[v_3] > 8\mathbf{p}[v'_2 : *] + \epsilon$, then $\mathbf{p}[w_3] = 1$ and, consequently, $\mathbf{p}[v_3] = 0$, which is a contradiction to $\mathbf{p}[v_3] > 8\mathbf{p}[v'_2 : *] + \epsilon$.

On the other hand, if $8\mathbf{p}[v'_2 : *] > \mathbf{p}[v_3] + \epsilon$, then $\mathbf{p}[w_3] = 0$ and consequently $\mathbf{p}[v_3] = 1$. Hence, $\mathbf{p}[v_3]$ cannot be less than $\min\{1, 8\mathbf{p}[v'_2 : *] - \epsilon\}$.

From the above observations it follows that $\mathbf{p}[v_3] = \min\{1, 8\mathbf{p}[v'_2 : *]\} \pm \epsilon$ and, using claim 3, $\mathbf{p}[v_3] = \min\{1, \alpha\mathbf{p}[v_1] + \beta\mathbf{p}[v_2] + \gamma\mathbf{p}[v_1]\mathbf{p}[v_2]\} \pm 81\epsilon$. \square

It remains to show that the graph of the game can be legally colored using 3 colors. The coloring is shown in Figure 11. \square

Now that we have our hands on the game $\mathcal{G}_{+,*}$ of Proposition 6, we can reduce r -player games to 3-player games, for any fixed r , using the algorithm of Figure 10 with the following tweak: in the construction of game \mathcal{GG} at Step 1 of the algorithm, instead of using the addition and multiplication gadgets \mathcal{G}_+ , \mathcal{G}_* of Section 4.1, we use our more elaborate $\mathcal{G}_{+,*}$ gadget. Let us call the resulting game \mathcal{GG} . We will show that we can construct a graphical game \mathcal{GG}' which is equivalent to \mathcal{GG} in the sense that there is a surjective mapping from the Nash equilibria of \mathcal{GG}' to the Nash equilibria of \mathcal{GG} and which, moreover, can be legally colored using three colors. Then we can proceed as in Step 4 of Figure 10 to get the desired 3-player normal form game G' .

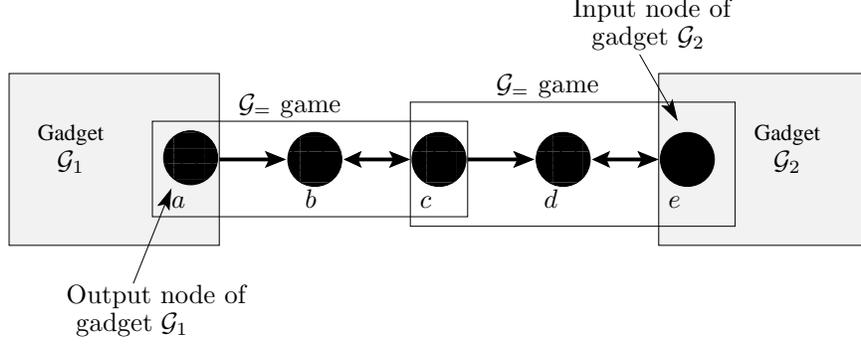


Figure 12: The interposition of two $\mathcal{G}_=$ games between gadgets \mathcal{G}_1 and \mathcal{G}_2 does not change the game.

The construction of \mathcal{GG}' and its coloring can be done as follows: Recall that all our gadgets have some distinguished vertices which are the *inputs* and one distinguished vertex which is the *output*. The gadgets are put together to construct \mathcal{GG} by identifying the output vertices of some gadgets as the input vertices of other gadgets. It is easy to see that we get a graphical game with the same functionality if, instead of identifying the output vertex of some gadget with the input of another gadget, we interpose a sequence of two $\mathcal{G}_=$ games between the two gadgets to be connected, as shown in Figure 12. If we “glue” our gadgets in this way then the resulting graphical game \mathcal{GG}' can be legally colored using 3 colors:

- i. (stage 1) legally color the vertices inside the “initial gadgets” using 3 colors
- ii. (stage 2) extend the coloring to the vertices that serve as “connections” between gadgets; any 3-coloring of the initial gadgets can be extended to a 3-coloring of \mathcal{GG}' because, for any pair of gadgets $\mathcal{G}_1, \mathcal{G}_2$ which are connected (Figure 12) and for any colors assigned to the output vertex a of gadget \mathcal{G}_1 and the input vertex e of gadget \mathcal{G}_2 , the intermediate vertices b, c and d can be also colored legally. For example, if vertex a gets color 1 and vertex e color 2 at stage 1, then, at stage 2, b can be colored 2, c can be colored 3 and d can be colored 1.

This completes the proof of the theorem. □

4.6 Preservation of Approximate equilibria

Our reductions so far map exact equilibrium points. In this section we generalize to approximate equilibria and prove the second part of Theorem 4. We claim that the reductions of the previous sections translate the problem of finding an ϵ -Nash equilibrium of a game to the problem of finding an ϵ' -Nash equilibrium of its image, for ϵ' polynomial in ϵ and inverse polynomial in the size of the game. As a consequence, we obtain polynomial-time equivalence results for the problems r -NASH and d -GRAPHICAL-NASH. To prove the second part of Theorem 4, we extend Theorems 5, 6 and 8 of the previous sections.

Theorem 9 *For every fixed $d > 1$, there is a polynomial-time reduction from d -GRAPHICAL-NASH to $(d^2 + 1)$ -NASH.*

Proof. Let $\widetilde{\mathcal{GG}}$ be a graphical game of maximum degree d and \mathcal{GG} the resulting graphical game after rescaling all utilities by $1/\max\{\tilde{u}\}$, where $\max\{\tilde{u}\}$ is the largest entry in the utility tables of game $\widetilde{\mathcal{GG}}$, so that they lie in the set $[0, 1]$, as in the first step of Figure 7. Assume that $\epsilon < 1$.

In time polynomial in $|\mathcal{GG}| + \log(1/\epsilon)$, we will specify a normal form 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 \mathcal{GG} . This will be enough, since an ϵ -Nash equilibrium of \mathcal{GG} is trivially an $\epsilon \cdot \max\{\tilde{u}\}$ -Nash equilibrium of game $\widehat{\mathcal{GG}}$ and, moreover, $|\mathcal{GG}|$ is polynomial in $|\widehat{\mathcal{GG}}|$.

We construct \mathcal{G} using the algorithm of Figure 7; recall that $M \geq 2\frac{n}{r}$, where r is the number of color classes specified in Figure 7 and n is the number of vertices in \mathcal{GG} after the possible addition of dummy vertices to make sure that all color classes have the same number of vertices (as in Step 3 of Figure 7). Let us choose $\epsilon' \leq \epsilon(\frac{r}{n} - \frac{1}{M})^d$; we will argue that from any ϵ' -Nash equilibrium of game \mathcal{G} one can construct in polynomial time an ϵ -Nash equilibrium of game \mathcal{GG} .

Suppose that $p = c(v)$ for some vertex v of the graphical game \mathcal{GG} . As in the proof of Theorem 5, Lemma 7, it can be shown that in any ϵ' -Nash equilibrium of the game \mathcal{G} ,

$$\Pr(p \text{ plays } v) \in \left[\frac{r}{n} - \frac{1}{M}, \frac{r}{n} + \frac{1}{M} \right].$$

Now, without loss of generality, assume that p is odd (pursuer) and suppose that v is vertex $v_i^{(p)}$ in the notation of Figure 7. Then, in an ϵ' -Nash equilibrium of the game \mathcal{G} , we have, by the definition of a Nash equilibrium, that for all strategies $a, a' \in S_v$ of vertex v :

$$\mathbb{E}[\text{payoff to } p \text{ for playing } (v, a)] > \mathbb{E}[\text{payoff to } p \text{ for playing } (v, a')] + \epsilon' \Rightarrow x_{(v,a')}^p = 0.$$

But

$$\mathbb{E}[\text{payoff to } p \text{ for playing } (v, a)] = M \cdot \Pr(p+1 \text{ plays } v_i^{(p+1)}) + \sum_{s \in S_{\mathcal{N}(v) \setminus \{v\}}} u_{as}^v \prod_{u \in \mathcal{N}(v) \setminus \{v\}} x_{(u,s_u)}^{c(u)}$$

and similarly for a' . Therefore, the previous inequality implies

$$\sum_{s \in S_{\mathcal{N}(v) \setminus \{v\}}} u_{as}^v \prod_{u \in \mathcal{N}(v) \setminus \{v\}} x_{(u,s_u)}^{c(u)} > \sum_{s \in S_{\mathcal{N}(v) \setminus \{v\}}} u_{a's}^v \prod_{u \in \mathcal{N}(v) \setminus \{v\}} x_{(u,s_u)}^{c(u)} + \epsilon' \Rightarrow x_{(v,a')}^p = 0$$

So letting

$$x_a^v = x_{(v,a)}^c / \sum_{j \in S_v} x_{(v,j)}^c, \quad \forall v \in V, a \in S_v,$$

as we did in the proof of Theorem 5, we get that, for all $v \in V$, $a, a' \in S_v$,

$$\sum_{s \in S_{\mathcal{N}(v) \setminus \{v\}}} u_{as}^v \prod_{u \in \mathcal{N}(v) \setminus \{v\}} x_{s_u}^u > \sum_{s \in S_{\mathcal{N}(v) \setminus \{v\}}} u_{a's}^v \prod_{u \in \mathcal{N}(v) \setminus \{v\}} x_{s_u}^u + \epsilon'/\mathcal{T} \Rightarrow x_{a'}^v = 0, \quad (13)$$

where $\mathcal{T} = \prod_{u \in \mathcal{N}(v) \setminus \{v\}} \sum_{j \in S_u} x_{(u,j)}^{c(u)} = \prod_{u \in \mathcal{N}(v) \setminus \{v\}} \Pr[c(u) \text{ plays } u] \geq (\frac{r}{n} - \frac{1}{M})^d$. By the definition of ϵ' it follows that $\epsilon'/\mathcal{T} \leq \epsilon$. Hence, from (13) it follows that $\{x_a^v\}_{v,a}$ is an ϵ -Nash equilibrium of the game \mathcal{GG} . \square

We have the following extension of Theorem 6.

Theorem 10 *For every fixed $r > 1$, there is a polynomial-time reduction from r -NASH to 3-GRAPHICAL NASH with two strategies per vertex.*

Proof. Let $\tilde{\mathcal{G}}$ be a normal form game with r players, $1, 2, \dots, r$, and strategy sets $S_p = [n]$, for all $p \in [r]$, and let $\{\tilde{u}_s^p : p \in [r], s \in S\}$ be the utilities of the players. Denote by \mathcal{G} the game constructed at the first step of Figure 9 which results from $\tilde{\mathcal{G}}$ after rescaling all utilities by $1/\max\{\tilde{u}_s^p\}$ so that they lie in $[0, 1]$; let $\{u_s^p : p \in [r], s \in S\}$ be the utilities of the players in game \mathcal{G} . Also, let $\epsilon < 1$. In time polynomial in $|\mathcal{G}| + \log(1/\epsilon)$, we will specify a graphical game \mathcal{GG} and an accuracy ϵ' with the property that, given an ϵ' -Nash equilibrium of \mathcal{GG} , one can recover in polynomial time an ϵ -Nash equilibrium of \mathcal{G} . This will be enough, since an ϵ -Nash equilibrium of \mathcal{G} is trivially an $\epsilon \cdot \max\{\tilde{u}_s^p\}$ -Nash equilibrium of game $\tilde{\mathcal{G}}$ and, moreover, $|\mathcal{G}|$ is polynomial in $|\tilde{\mathcal{G}}|$. In our reduction, the graphical game \mathcal{GG} will be the same as the one described in the proof of Theorem 6 (Figure 9), while the accuracy specification will be of the form $\epsilon' = \epsilon/p(|\mathcal{G}|)$, where $p(\cdot)$ is a polynomial that will be specified later. We will use the same labels for the vertices of the game \mathcal{GG} that we used in the proof Theorem 6.

Suppose $N_{\mathcal{GG}}$ is some ϵ' -Nash equilibrium of the game \mathcal{GG} and let $\{\mathbf{p}[v(x_j^p)]\}_{j,p}$ denote the probabilities with which the vertices $v(x_j^p)$ of \mathcal{GG} play strategy 1. In the proof of Theorem 6 we considered the following mapping from the Nash equilibria of game \mathcal{GG} to the Nash equilibria of game \mathcal{G} :

$$x_j^p := \mathbf{p}[v(x_j^p)], \text{ for all } p \text{ and } j. \quad (14)$$

Although (14) succeeds in mapping exact equilibrium points, it fails for approximate equilibria, as specified by the following remark —its justification follows from the proof of Lemma 9.

Remark 2 *For any $\epsilon' > 0$, there exists an ϵ' -Nash equilibrium of game \mathcal{GG} such that $\sum_j \mathbf{p}[v(x_j^p)] \neq 1$, for some player $p \leq r$, and, moreover, $\mathbf{p}[v(U_j^p)] > \mathbf{p}[v(U_{j'}^p)] + \epsilon'$, for some $p \leq r$, j and j' , and, yet, $\mathbf{p}[v(x_{j'}^p)] > 0$.*

Recall from Section 4.3, that, for all p, j , the probability $\mathbf{p}[v(U_j^p)]$ represents the utility of player p for playing pure strategy j , when the other players play according to $\{x_j^q := \mathbf{p}[v(x_j^q)]\}_{j,q \neq p}$ ⁵. Therefore, not only the $\{x_j^p := \mathbf{p}[v(x_j^p)]\}_j$ do not necessarily constitute a distribution —this could be easily fixed by rescaling— but, also, the defining property of an approximate equilibrium (2) is in question. The following lemma bounds the deviation from the approximate equilibrium conditions.

Lemma 9 *In any ϵ' -Nash equilibrium of the game \mathcal{GG} ,*

- (i) *for all $p \in [r]$, $|\sum_j \mathbf{p}[v(x_j^p)] - 1| \leq 2cne'$, and,*
- (ii) *for all $p \in [r]$, $j, j' \in [n]$, $\mathbf{p}[v(U_j^p)] > \mathbf{p}[v(U_{j'}^p)] + 5cne' \Rightarrow \mathbf{p}[v(x_{j'}^p)] \in [0, cne']$,*

where $c \geq 1$ is the maximum error amplification of the gadgets used in the construction of \mathcal{GG} .

Proof. Note that at an ϵ' -Nash equilibrium of game \mathcal{GG} the following properties are satisfied for all $p \in [r]$ by the vertices of game \mathcal{GG} , since the error amplification of the gadgets is at most c :

⁵Note, however, that, since we are considering an ϵ' -Nash equilibrium of game \mathcal{GG} , the Equation (11) of Section 4.3 will be only satisfied approximately as specified by Lemma 11.

$$\mathbf{p} \left[v \left(\sum_{i=1}^n x_i^p \right) \right] = 1 \quad (15)$$

$$\mathbf{p} \left[v \left(\sum_{i=1}^j x_i^p \right) \right] = \mathbf{p} \left[v \left(\sum_{i=1}^{j+1} x_i^p \right) \right] \cdot (1 - \mathbf{p}[v_j^p]) \pm c\epsilon', \forall j < n \quad (16)$$

$$\mathbf{p} [v(x_{j+1}^p)] = \mathbf{p} \left[v \left(\sum_{i=1}^{j+1} x_i^p \right) \right] \cdot \mathbf{p}[v_j^p] \pm c\epsilon', \forall j < n \quad (17)$$

$$\mathbf{p} [v(x_1^p)] = \mathbf{p} \left[v \left(\sum_{i=1}^1 x_i^p \right) \right] \pm c\epsilon' \quad (18)$$

Proof of (i): By successive applications of (16) and (17), we deduce

$$\begin{aligned} \sum_{j=1}^n \mathbf{p}[v(x_j^p)] &= \sum_{j=2}^n \left\{ \mathbf{p} \left[v \left(\sum_{i=1}^j x_i^p \right) \right] \cdot \mathbf{p}[v_{j-1}^p] \right\} + \mathbf{p} \left[v \left(\sum_{i=1}^1 x_i^p \right) \right] \pm cn\epsilon' \\ &= \sum_{j=2}^n \left\{ \mathbf{p} \left[v \left(\sum_{i=1}^j x_i^p \right) \right] \cdot \mathbf{p}[v_{j-1}^p] \right\} + \left(\mathbf{p} \left[v \left(\sum_{i=1}^2 x_i^p \right) \right] \cdot (1 - \mathbf{p}[v_1^p]) \pm c\epsilon' \right) \pm cn\epsilon' \\ &= \sum_{j=3}^n \left\{ \mathbf{p} \left[v \left(\sum_{i=1}^j x_i^p \right) \right] \cdot \mathbf{p}[v_{j-1}^p] \right\} + \mathbf{p} \left[v \left(\sum_{i=1}^2 x_i^p \right) \right] \pm c(n+1)\epsilon' \\ &= \dots \\ &= \mathbf{p} \left[v \left(\sum_{i=1}^n x_i^p \right) \right] \pm c(2n-1)\epsilon' \\ &= 1 \pm c(2n-1)\epsilon' \end{aligned}$$

Proof of (ii): Let us first observe the behavior of vertices $w(U_j^p)$ and v_j^p in an ϵ' -Nash equilibrium.

- **Behavior of $w(U_j^p)$ vertices:** The utility of vertex $w(U_j^p)$ for playing strategy 0 is $\mathbf{p}[v(U_{\leq j}^p)]$, whereas for playing 1 it is $\mathbf{p}[v(U_{j+1}^p)]$. Therefore,

$$\begin{aligned} \mathbf{p}[v(U_{\leq j}^p)] > \mathbf{p}[v(U_{j+1}^p)] + \epsilon' &\Rightarrow \mathbf{p}[w(U_j^p)] = 0 \\ \mathbf{p}[v(U_{j+1}^p)] > \mathbf{p}[v(U_{\leq j}^p)] + \epsilon' &\Rightarrow \mathbf{p}[w(U_j^p)] = 1 \\ |\mathbf{p}[v(U_{j+1}^p)] - \mathbf{p}[v(U_{\leq j}^p)]| \leq \epsilon' &\Rightarrow \mathbf{p}[w(U_j^p)] \text{ can be anything} \end{aligned}$$

- **Behavior of v_j^p vertices:** The utility of vertex v_j^p for playing strategy 0 is $1 - \mathbf{p}[w(U_j^p)]$, whereas for playing 1 it is $\mathbf{p}[w(U_j^p)]$. Therefore,

$$\begin{aligned} \mathbf{p}[w(U_j^p)] < \frac{1-\epsilon'}{2} &\Rightarrow \mathbf{p}[v_j^p] = 0 \\ \mathbf{p}[w(U_j^p)] > \frac{1+\epsilon'}{2} &\Rightarrow \mathbf{p}[v_j^p] = 1 \\ |\mathbf{p}[w(U_j^p)] - \frac{1}{2}| \leq \frac{\epsilon'}{2} &\Rightarrow \mathbf{p}[v_j^p] \text{ can be anything} \end{aligned}$$

Note that, since the error amplification of the gadget \mathcal{G}_{\max} is at most c and computing $\mathbf{p}[v(U_{\leq j}^p)]$, for all j , requires j applications of \mathcal{G}_{\max} ,

$$\mathbf{p}[v(U_{\leq j}^p)] = \max_{i \leq j} \mathbf{p}[v(U_i^p)] \pm c\epsilon' j. \quad (19)$$

To establish the second part of the claim, we need to show that, for all p, j, j' ,

$$\mathbf{p}[v(U_j^p)] > \mathbf{p}[v(U_{j'}^p)] + 5cn\epsilon' \Rightarrow \mathbf{p}[v(x_{j'}^p)] \in [0, nc\epsilon'].$$

1. Note that, if there exists some $j'' < j'$ such that $\mathbf{p}[v(U_{j''}^p)] > \mathbf{p}[v(U_{j'}^p)] + c\epsilon'n$, then

$$\begin{aligned} \mathbf{p}[v(U_{\leq j'-1}^p)] &= \max_{i \leq j'-1} \mathbf{p}[v(U_i^p)] \pm c\epsilon'(j' - 1) \\ &\geq \mathbf{p}[v(U_{j''}^p)] - c\epsilon'(j' - 1) \\ &> \mathbf{p}[v(U_{j'}^p)] + cn\epsilon' - c\epsilon'(j' - 1) \geq \mathbf{p}[v(U_{j'}^p)] + \epsilon'. \end{aligned}$$

Then, because $\mathbf{p}[v(U_{\leq j'-1}^p)] > \mathbf{p}[v(U_{j'}^p)] + \epsilon'$, it follows that $\mathbf{p}[w(U_{j'-1}^p)] = 0$ and $\mathbf{p}[v_{j'-1}^p] = 0$. Therefore,

$$\mathbf{p}[v(x_{j'}^p)] = \mathbf{p} \left[v \left(\sum_{i=1}^{j'} x_i^p \right) \right] \cdot \mathbf{p}[v_{j'-1}^p] \pm c\epsilon' = \pm c\epsilon'.$$

2. The case $j < j'$ reduces to the previous for $j'' = j$.
3. It remains to deal with the case $j > j'$, under the assumption that, for all $j'' < j'$,

$$\mathbf{p}[v(U_{j''}^p)] \leq \mathbf{p}[v(U_{j'}^p)] + c\epsilon'n.$$

which, in turn, implies

$$\mathbf{p}[v(U_{\leq j'}^p)] < \mathbf{p}[v(U_{j'}^p)] + 2c\epsilon'n. \quad (\text{by (19)})$$

Let us further distinguish the following subcases

- (a) If there exists some $k, j' + 1 \leq k \leq j$, such that $\mathbf{p}[v(U_k^p)] > \mathbf{p}[v(U_{\leq k-1}^p)] + \epsilon'$, then

$$\begin{aligned} \mathbf{p}[w(U_{k-1}^p)] = 1 &\Rightarrow \mathbf{p}[v_{k-1}^p] = 1 \\ &\Rightarrow \mathbf{p} \left[v \left(\sum_{i=1}^{k-1} x_i^p \right) \right] = \mathbf{p} \left[v \left(\sum_{i=1}^k x_i^p \right) \right] (1 - \mathbf{p}[v_{k-1}^p]) \pm c\epsilon' = \pm c\epsilon' \\ &\Rightarrow \mathbf{p} \left[v \left(\sum_{i=1}^{j'} x_i^p \right) \right] = \pm(k - j')c\epsilon' \quad \left(\begin{array}{l} \text{by successive applications} \\ \text{of equation (16)} \end{array} \right) \\ &\Rightarrow \mathbf{p} [v(x_{j'}^p)] = \pm nc\epsilon'. \quad (\text{by (17), (18)}) \end{aligned}$$

- (b) If, for all $k, j' + 1 \leq k \leq j$, it holds that $\mathbf{p}[v(U_k^p)] \leq \mathbf{p}[v(U_{\leq k-1}^p)] + \epsilon'$, we will show a contradiction; hence, only the previous case can hold. Towards a contradiction, we argue first that

$$\mathbf{p}[v(U_{\leq j'+1}^p)] \geq \mathbf{p}[v(U_{j'}^p)] - 2cn\epsilon'.$$

To show this, we distinguish the cases $j = j' + 1, j > j' + 1$.

- In the case $j = j' + 1$, we have

$$\mathbf{p}[v(U_{\leq j'+1}^p)] \geq \max \{ \mathbf{p}[v(U_{j'+1}^p)], \mathbf{p}[v(U_{\leq j'}^p)] \} - c\epsilon' \geq \mathbf{p}[v(U_{j'+1}^p)] - c\epsilon' = \mathbf{p}[v(U_j^p)] - c\epsilon'.$$

- In the case $j > j' + 1$, we have for all k , $j' + 2 \leq k \leq j$,

$$\mathbf{p}[v(U_{\leq k-1}^p)] \geq \max \{ \mathbf{p}[v(U_{\leq k-1}^p)], \mathbf{p}[v(U_k^p)] \} - \epsilon' \geq \mathbf{p}[v(U_{\leq k}^p)] - c\epsilon' - \epsilon',$$

where the last inequality holds since the game \mathcal{G}_{\max} has error amplification at most c . Summing these inequalities for $j' + 2 \leq k \leq j$, we deduce that

$$\begin{aligned} \mathbf{p}[v(U_{\leq j'+1}^p)] &\geq \mathbf{p}[v(U_{\leq j}^p)] - (c\epsilon' + \epsilon')(n-2) \\ &\geq \max \{ \mathbf{p}[v(U_j^p)], \mathbf{p}[v(U_{\leq j-1}^p)] \} - c\epsilon' - (c\epsilon' + \epsilon')(n-2) \\ &\geq \mathbf{p}[v(U_j^p)] - 2c\epsilon'n. \end{aligned}$$

It follows that

$$\mathbf{p}[v(U_{\leq j'+1}^p)] > \mathbf{p}[v(U_{j'}^p)] + 3cn\epsilon'.$$

But,

$$\mathbf{p}[v(U_{\leq j'+1}^p)] \leq \max \{ \mathbf{p}[v(U_{j'+1}^p)], \mathbf{p}[v(U_{\leq j'}^p)] \} + c\epsilon'$$

and recall that

$$\mathbf{p}[v(U_{\leq j'}^p)] < \mathbf{p}[v(U_{j'}^p)] + 2c\epsilon'n.$$

We can deduce that

$$\max \{ \mathbf{p}[v(U_{j'+1}^p)], \mathbf{p}[v(U_{\leq j'}^p)] \} = \mathbf{p}[v(U_{j'+1}^p)],$$

which combined with the above implies

$$\mathbf{p}[v(U_{j'+1}^p)] \geq \mathbf{p}[v(U_{j'}^p)] + 3cn\epsilon' - c\epsilon' > \mathbf{p}[v(U_{\leq j'}^p)] + \epsilon'.$$

□

From Lemma 9, it follows that the extraction of an ϵ -Nash equilibrium of game \mathcal{G} from an ϵ' -Nash equilibrium of game \mathcal{GG} cannot be done by just interpreting the values $\{x_j^p := \mathbf{p}[v(x_j^p)]\}_j$ as the mixed strategy of player p . What we show next is that, for the right choice of ϵ' , a *trim and renormalize* transformation succeeds in deriving an ϵ -Nash equilibrium of game \mathcal{G} from an ϵ' -Nash equilibrium of game \mathcal{GG} . Indeed, for all $p \leq r$, suppose that $\{\hat{x}_j^p\}_j$ are the values derived from $\{x_j^p\}_j$ by setting

$$\hat{x}_j^p = \begin{cases} 0, & \text{if } x_j^p \leq cn\epsilon' \\ x_j^p, & \text{otherwise} \end{cases}$$

and then renormalizing the resulting values $\{\hat{x}_j^p\}_j$ so that $\sum_j \hat{x}_j^p = 1$.

Lemma 10 *There exists a polynomial $p(\cdot)$ such that, if $\{\{x_j^p\}_j\}_p$ is an $\epsilon/p(|\mathcal{G}|)$ -Nash equilibrium of game \mathcal{GG} , then the trimmed and renormalized values $\{\{\hat{x}_j^p\}_j\}_p$ constitute an ϵ -Nash equilibrium of game \mathcal{G} .*

Proof. We first establish the following useful lemma

Lemma 11 *At an ϵ' -Nash equilibrium of game \mathcal{GG} , for all p, j , it holds that*

$$\mathbf{p}[v(U_j^p)] = \sum_{s \in S_{-p}} u_{js}^p x_{s_1}^1 \cdots x_{s_{p-1}}^{p-1} x_{s_{p+1}}^{p+1} \cdots x_{s_r}^r \pm 2n^{r-1} \zeta_r,$$

where c is the maximum error amplification of the gadgets used in the construction of \mathcal{GG} , $\zeta_r = c\epsilon' + ((1 + \zeta)^r - 1)(c\epsilon' + 1)$, $\zeta = 2r \log n c\epsilon'$.

Proof. Using the same notation as in Section 4.3, let $S_{-p} = \{S_{-p}(1), \dots, S_{-p}(n^{r-1})\}$, so that

$$\sum_{s \in S_{-p}} u_{js}^p x_s = \sum_{\ell=1}^{n^{r-1}} u_{jS_{-p}(\ell)}^p x_{S_{-p}(\ell)}.$$

Recall that in \mathcal{GG} , for each partial sum $\sum_{\ell=1}^z u_{jS_{-p}(\ell)}^p x_{S_{-p}(\ell)}$, $1 \leq z \leq n^{r-1}$, we have included vertex $v(\sum_{\ell=1}^z u_{jS_{-p}(\ell)}^p x_{S_{-p}(\ell)})$. Similarly, for each partial product of the summands $u_{js}^p \prod_{p \neq q \leq z} x_{s_q}^q$, $0 \leq z \leq r$, we have included vertex $v(u_{js}^p \prod_{p \neq q \leq z} x_{s_q}^q)$. Note that, since we have rescaled the utilities to the set $[0, 1]$, all partial sums and products must also lie in $[0, 1]$. Note, moreover, that, to avoid large degrees in the resulting graphical game, each time we need to make use of a value $x_{s_q}^q$ we create a new copy of the vertex $v(x_{s_q}^q)$ using the gadget $\mathcal{G}_=$ and, then, use the new copy for the computation of the desired partial product; an easy calculation shows that we have to make $(r-1)n^{r-1}$ copies of $v(x_{s_q}^q)$, for all $q \leq r$, $s_q \in S_q$. To limit the degree of each vertex to 3 we create a binary tree of copies of $v(x_{s_q}^q)$ with $(r-1)n^{r-1}$ leaves and use each leaf once. Then, because of the error amplification of $\mathcal{G}_=$, this already induces an error of $\pm \lceil \log(r-1)n^{r-1} \rceil c\epsilon'$ to each of the factors of the partial products. The following lemma characterizes the error that results from the error amplification of our gadgets in the computation of the partial products and can be proved easily by induction.

Lemma 12 *For all $p \leq r$, $j \in S_p$, $s \in S_{-p}$ and $z \leq r$,*

$$\mathbf{p} \left[v \left(u_{js}^p \prod_{p \neq \ell \leq z} x_{s_\ell}^\ell \right) \right] = u_{js}^p \prod_{p \neq \ell \leq z} x_{s_\ell}^\ell \pm \zeta_z, \quad (20)$$

where $\zeta_z = c\epsilon' + ((1 + \zeta)^z - 1)(c\epsilon' + 1)$, $\zeta = 2r \log n c\epsilon'$.

The following lemma characterizes the error in the computation of the partial sums and can be proved by induction using the previous lemma for the base case.

Lemma 13 *For all $p \leq r$, $j \in S_p$ and $z \leq n^{r-1}$,*

$$\mathbf{p} \left[v \left(\sum_{\ell=1}^z u_{jS_{-p}(\ell)}^p x_{S_{-p}(\ell)} \right) \right] = \sum_{\ell=1}^z u_{jS_{-p}(\ell)}^p x_{S_{-p}(\ell)} \pm (z\zeta_r + (z-1)c\epsilon'), \quad (21)$$

where ζ_r is defined as in Lemma 12.

From Lemma 13 we can deduce, in particular, that for all $p \leq r$, $j \in S_p$,

$$\mathbf{p}[v(U_j^p)] = \sum_{s \in S_{-p}} u_{js}^p x_s \pm 2n^{r-1} \zeta_r.$$

□

Lemma 14 For all $p \leq r$, $j \in S_p$,

$$\left| \sum_{s \in S_{-p}} u_{j_s}^p x_s - \sum_{s \in S_{-p}} u_{j_s}^p y_s \right| \leq \max_{s \in S_{-p}} \{u_{j_s}^p\} \sum_{q \neq p} \sum_{i \in S_q} |x_i^q - y_i^q|.$$

Proof. We have

$$\left| \sum_{s \in S_{-p}} u_{j_s}^p x_s - \sum_{s \in S_{-p}} u_{j_s}^p y_s \right| \leq \sum_{s \in S_{-p}} u_{j_s}^p |x_s - y_s| \leq \max_{s \in S_{-p}} \{u_{j_s}^p\} \sum_{s \in S_{-p}} |x_s - y_s|. \quad (22)$$

Let us denote by \mathcal{X}^q the random variable, ranging over the set S_q , which represents the mixed strategy $\{x_i^q\}_{i \in S_q}$, $q \leq r$. Similarly define the random variable \mathcal{Y}^q from the mixed strategy $\{y_i^q\}_{i \in S_q}$, $q \leq r$. Note, then, that $\frac{1}{2} \sum_{s \in S_{-p}} |x_s - y_s|$ is precisely the total variation distance between the vector random variable $(\mathcal{X}^q)_{q \neq p}$ and the vector random variable $(\mathcal{Y}^q)_{q \neq p}$. That is,

$$\frac{1}{2} \sum_{s \in S_{-p}} |x_s - y_s| = \|(\mathcal{X}^q)_{q \neq p} - (\mathcal{Y}^q)_{q \neq p}\|_{TV}. \quad (23)$$

By the coupling lemma, we have that

$$\|(\mathcal{X}^q)_{q \neq p} - (\mathcal{Y}^q)_{q \neq p}\|_{TV} \leq \Pr((\mathcal{X}^q)_{q \neq p} \neq (\mathcal{Y}^q)_{q \neq p}),$$

for any coupling of $(\mathcal{X}^q)_{q \neq p}$ and $(\mathcal{Y}^q)_{q \neq p}$. Applying a union bound to the right hand side of the above implies

$$\|(\mathcal{X}^q)_{q \neq p} - (\mathcal{Y}^q)_{q \neq p}\|_{TV} \leq \sum_{q \neq p} \Pr(\mathcal{X}^q \neq \mathcal{Y}^q). \quad (24)$$

Now let us fix a coupling between $(\mathcal{X}^q)_{q \neq p}$ and $(\mathcal{Y}^q)_{q \neq p}$ so that, for all $q \neq p$,

$$\Pr(\mathcal{X}^q \neq \mathcal{Y}^q) = \|\mathcal{X}^q - \mathcal{Y}^q\|_{TV}.$$

Such a coupling exists by the coupling lemma for each $q \neq p$ individually, and for the whole vectors $(\mathcal{X}^q)_{q \neq p}$ and $(\mathcal{Y}^q)_{q \neq p}$ it exists because also the \mathcal{X}^q 's are independent and so are the \mathcal{Y}^q 's. Then (24) implies that

$$\|(\mathcal{X}^q)_{q \neq p} - (\mathcal{Y}^q)_{q \neq p}\|_{TV} \leq \sum_{q \neq p} \|\mathcal{X}^q - \mathcal{Y}^q\|_{TV},$$

so that from (22), (23) we get

$$\left| \sum_{s \in S_{-p}} u_{j_s}^p x_s - \sum_{s \in S_{-p}} u_{j_s}^p y_s \right| \leq \max_{s \in S_{-p}} \{u_{j_s}^p\} 2 \sum_{q \neq p} \|\mathcal{X}^q - \mathcal{Y}^q\|_{TV}. \quad (25)$$

Now, note that, for all q ,

$$\|\mathcal{X}^q - \mathcal{Y}^q\|_{TV} = \frac{1}{2} \sum_{i \in S_q} |x_i^q - y_i^q|.$$

Hence, (25) implies

$$\left| \sum_{s \in S_{-p}} u_{j_s}^p x_s - \sum_{s \in S_{-p}} u_{j_s}^p y_s \right| \leq \max_{s \in S_{-p}} \{u_{j_s}^p\} \sum_{q \neq p} \sum_{i \in S_q} |x_i^q - y_i^q|.$$

□

We can conclude the proof of Lemma 10, by invoking Lemmas 11 and 14. Indeed, by the definition of the $\{\hat{x}_j^p\}$, it follows that for all $p, j \in S_p$,

$$\hat{x}_j^p = \begin{cases} \frac{x_j^p}{\Lambda^p}, & x_j^p > cn\epsilon' \\ 0, & x_j^p \leq cn\epsilon' \end{cases},$$

where

$$1 \geq \Lambda^p = \sum_{j \in S_p} x_j^p \mathcal{X}_{\{x_j^p > cn\epsilon'\}} = 1 - \sum_{j \in S_p} x_j^p \mathcal{X}_{\{x_j^p \leq cn\epsilon'\}} \geq 1 - n \cdot cn\epsilon',$$

where $\mathcal{X}_{\{\cdot\}}$ is the indicator function. Therefore,

$$|\hat{x}_j^p - x_j^p| = \begin{cases} \frac{x_j^p}{\Lambda^p} - x_j^p, & x_j^p > cn\epsilon' \\ x_j^p, & x_j^p \leq cn\epsilon' \end{cases},$$

which implies

$$|\hat{x}_j^p - x_j^p| \leq \begin{cases} \frac{1}{\Lambda^p} - 1, & x_j^p > cn\epsilon' \\ cn\epsilon', & x_j^p \leq cn\epsilon' \end{cases}.$$

So,

$$|\hat{x}_j^p - x_j^p| \leq \max \left\{ cn\epsilon', \frac{n^2 c\epsilon'}{1 - n^2 c\epsilon'} \right\} =: \delta_1,$$

which by Lemma 14 implies that

$$\left| \sum_{s \in S_{-p}} u_{js}^p x_s - \sum_{s \in S_{-p}} u_{js}^p \hat{x}_s \right| \leq \max_{s \in S_{-p}} \{u_{js}^p\} (r-1) n \delta_1 \leq (r-1) n \delta_1 =: \delta_2, \quad (26)$$

where the second inequality follows from the fact that we have rescaled the utilities so that they lie in $[0, 1]$.

Choosing $\epsilon' = \frac{\epsilon}{40cr^2n^{r+1}}$, we will argue that the conditions of an ϵ -Nash equilibrium are satisfied by the mixed strategies $\{\hat{x}_j^p\}_{p,j}$. First, note that:

$$(1 + 2r \log n \, c\epsilon')^r - 1 \leq \left(1 + \frac{\epsilon}{20rn^r}\right)^r - 1 \leq \exp \left\{ \frac{\epsilon}{20n^r} \right\} - 1 \leq \frac{\epsilon}{10n^r},$$

which implies that

$$2n^{r-1} \zeta_r \leq 2n^{r-1} \left(c\epsilon' + \frac{\epsilon}{10n^r} (c\epsilon' + 1) \right) \leq 2n^{r-1} \frac{1.5\epsilon}{10n^r} = \frac{3\epsilon}{10n} \leq \frac{0.3\epsilon}{n}.$$

Also, note that

$$\delta_1 = \max \left\{ cn\epsilon', \frac{n^2 c\epsilon'}{1 - n^2 c\epsilon'} \right\} \leq 2n^2 c\epsilon',$$

which gives

$$\delta_2 = (r-1)n\delta_1 \leq rn2n^2c \frac{\epsilon}{40cr^2n^{r+1}} \leq \frac{\epsilon}{20r}.$$

Thus, for all $p \leq r$, $j, j' \in S_p$, we have that

$$\begin{aligned}
\sum_{s \in S_{-p}} u_{j's}^p \hat{x}_s &> \sum_{s \in S_{-p}} u_{j's}^p \hat{x}_s + \epsilon \Rightarrow \sum_{s \in S_{-p}} u_{j's}^p x_s + \delta_2 > \sum_{s \in S_{-p}} u_{j's}^p x_s - \delta_2 + \epsilon && \text{(using (26))} \\
&\Rightarrow \sum_{s \in S_{-p}} u_{j's}^p x_s > \sum_{s \in S_{-p}} u_{j's}^p x_s + \epsilon - 2\delta_2 \\
&\Rightarrow \mathbf{p}[v(U_j^p)] + 2n^{r-1}\zeta_r > \mathbf{p}[v(U_{j'}^p)] - 2n^{r-1}\zeta_r + \epsilon - 2\delta_2 \text{ (using Lemma 11)} \\
&\Rightarrow \mathbf{p}[v(U_j^p)] > \mathbf{p}[v(U_{j'}^p)] - 4n^{r-1}\zeta_r + \epsilon - 2\delta_2 \\
&\Rightarrow \mathbf{p}[v(U_j^p)] > \mathbf{p}[v(U_{j'}^p)] + 5cn\epsilon' \\
&\Rightarrow x_{j'}^p \leq cn\epsilon' \text{ (using Lemma 9)} \\
&\Rightarrow \hat{x}_{j'}^p = 0.
\end{aligned}$$

Therefore, $\{\hat{x}_j^p\}$ is indeed an ϵ -Nash equilibrium of game \mathcal{G} , which concludes the proof of the lemma. \square

\square

We have the following extension of Theorem 8.

Theorem 11 *For every fixed $r > 1$, there is a polynomial-time reduction from r -NASH to 3-NASH.*

Proof. The proof follows immediately from the proofs of Theorems 9 and 10. Indeed, observe that the reduction of Theorem 10 still holds when we use the gadget $\mathcal{G}_{+,*}$ of Section 4.5 for the construction our graphical games, since the gadget $\mathcal{G}_{+,*}$ has constant error amplification. Therefore, the problem of computing an ϵ -Nash equilibrium of a r -player normal form game \mathcal{G} can be polynomially reduced to computing an ϵ' -Nash equilibrium of a graphical game \mathcal{GG}' which can be legally colored with 3 colors (after performing the “glueing” step described in the end of the proof of Theorem 8 and appropriately adjusting the ϵ' specified in the proof of Theorem 10). Observe, further, that the reduction of Theorem 9 can be used to map the latter to computing an ϵ'' -Nash equilibrium of a 3-player normal form game \mathcal{G}'' , since the number of players that are required for \mathcal{G}'' is equal to the minimum number of colors needed for a legal coloring of \mathcal{GG}' . The claim follows by combining the reductions. \square

\square

4.7 Reductions Between Different Notions of Approximation

We establish a polynomial time reduction from the problem of computing an approximately well supported Nash equilibrium to the problem of computing an approximate Nash equilibrium. As pointed out in Section 2, the reduction in the opposite direction is trivial, since an ϵ -approximately well supported Nash equilibrium is also an ϵ -approximate Nash equilibrium.

Lemma 15 *Given an ϵ -approximate Nash equilibrium $\{x_j^p\}_{j,p}$ of a game \mathcal{G} we can compute in polynomial time a $\sqrt{\epsilon} \cdot (\sqrt{\epsilon} + 1 + 4(r-1) \max\{u\})$ -approximately well supported Nash equilibrium $\{\hat{x}_j^p\}_{j,p}$, where r is the number of players in \mathcal{G} and $\max\{u\}$ is the maximum entry in the payoff tables of \mathcal{G} .*

Proof. Since $\{x_j^p\}_{j,p}$ is an ϵ -approximate Nash equilibrium, it follows that for every player $p \leq r$ and every mixed strategy $\{y_j^p\}_j$ for that player

$$\sum_{s \in S} u_s^p \cdot x_{s-p} \cdot x_{s_p}^p \geq \sum_{s \in S} u_s^p \cdot x_{s-p} \cdot y_{s_p}^p - \epsilon.$$

Equivalently,

$$\forall p \leq r, \quad \forall \{y_j^p\}_{j \in S_p} : \quad \sum_{j \in S_p} \left[\sum_{s-p \in S_{-p}} u_{j s-p}^p x_{s-p} \right] x_j^p \geq \sum_{j \in S_p} \left[\sum_{s-p \in S_{-p}} u_{j s-p}^p x_{s-p} \right] y_j^p - \epsilon. \quad (27)$$

For all $p \leq r$, denote $\mathcal{U}_j^p = \sum_{s-p \in S_{-p}} u_{j s-p}^p x_{s-p}$, for all $j \in S_p$, and $\mathcal{U}_{\max}^p = \max_j \mathcal{U}_j^p$. Then, if we choose $\{y_j^p\}_j$ to be some pure strategy from the set $\arg \max_j \mathcal{U}_j^p$, (27) implies

$$\forall p \leq r : \quad \sum_{j \in S_p} \mathcal{U}_j^p x_j^p \geq \mathcal{U}_{\max}^p - \epsilon. \quad (28)$$

Now, let us fix some player $p \leq r$. We want to upper bound the probability mass that the distribution $\{x_j^p\}_j$ assigns to pure strategies $j \in S_p$ which give expected utility \mathcal{U}_j^p more than an additive ϵk smaller than \mathcal{U}_{\max}^p , for some k to be specified later. The following bound is easy to derive using (28).

Claim 5 For all p , set

$$z^p = \sum_{j \in S_p} x_j^p \cdot \mathcal{X}_{\{\mathcal{U}_j^p < \mathcal{U}_{\max}^p - \epsilon k\}},$$

where \mathcal{X}_A is the characteristic function of the event A . Then

$$z^p \leq \frac{1}{k}.$$

Let us consider then the strategy profile $\{\hat{x}_j^p\}_{j,p}$ defined as follows

$$\forall p, j \in S_p : \quad \hat{x}_j^p = \begin{cases} \frac{x_j^p}{1-z^p}, & \mathcal{U}_j^p \geq \mathcal{U}_{\max}^p - \epsilon k \\ 0, & \text{otherwise} \end{cases}$$

We establish the following bound on the L_1 distance between the strategy profiles $\{x_j^p\}_j$ and $\{\hat{x}_j^p\}_j$.

Claim 6 For all p , $\sum_{j \in S_p} |x_j^p - \hat{x}_j^p| \leq \frac{2}{k-1}$.

Proof. Denote $S_{p,1} := \{j \mid j \in S_p, \mathcal{U}_j^p \geq \mathcal{U}_{\max}^p - \epsilon k\}$ and $S_{p,2} := S_p \setminus S_{p,1}$. Then

$$\begin{aligned}
\sum_{j \in S_p} |x_j^p - \hat{x}_j^p| &= \sum_{j \in S_{p,1}} |x_j^p - \hat{x}_j^p| + \sum_{j \in S_{p,2}} |x_j^p - \hat{x}_j^p| \\
&= \sum_{j \in S_{p,1}} \left| x_j^p - \frac{x_j^p}{1 - z^p} \right| + \sum_{j \in S_{p,2}} |x_j^p| \\
&= \sum_{j \in S_{p,1}} \left| x_j^p - \frac{x_j^p}{1 - z^p} \right| + z^p \\
&\leq \frac{z^p}{1 - z^p} \sum_{j \in S_{p,1}} x_j^p + z^p \\
&\leq \frac{1}{k-1} + \frac{1}{k} \leq \frac{2}{k-1}.
\end{aligned}$$

□

Now, for all players p , let $\hat{\mathcal{U}}_j^p$ and $\hat{\mathcal{U}}_{\max}^p$ be defined similarly to \mathcal{U}_j^p and \mathcal{U}_{\max}^p . Recall Lemma 14 from Section 4.6.

Lemma 16 For all $p, j \in S_p$,

$$|\mathcal{U}_j^p - \hat{\mathcal{U}}_j^p| \leq \max_{s \in S_{-p}} \{u_{j_s}^p\} \sum_{p' \neq p} \sum_{j \in S_{p'}} |x_j^{p'} - \hat{x}_j^{p'}|.$$

Let us then take $\Delta_2 = 2 \frac{r-1}{k-1} \max_{p,j \in S_p, s \in S_{-p}} \{u_{j_s}^p\}$. Claim 6 and Lemma 16 imply that the strategy profile $\{\hat{x}_j^p\}_{j,p}$ satisfies

$$\forall p, \forall j \in S_p : |\mathcal{U}_j^p - \hat{\mathcal{U}}_j^p| \leq \Delta_2.$$

We will establish that $\{\hat{x}_j^p\}_{j,p}$ is a $(\epsilon k + 2\Delta_2)$ -Nash equilibrium. Equivalently, we shall establish that

$$\forall p, \forall i, j \in S_p : \hat{\mathcal{U}}_j^p < \hat{\mathcal{U}}_i^p - (\epsilon k + 2\Delta_2) \Rightarrow \hat{x}_j^p = 0.$$

Indeed,

$$\begin{aligned}
\hat{\mathcal{U}}_j^p < \hat{\mathcal{U}}_i^p - (\epsilon k + 2\Delta_2) &\Rightarrow \mathcal{U}_j^p - \Delta_2 < \mathcal{U}_i^p + \Delta_2 - (\epsilon k + 2\Delta_2) \\
&\Rightarrow \mathcal{U}_j^p < \mathcal{U}_i^p - (\epsilon k + 2\Delta_2 - 2\Delta_2) \\
&\Rightarrow \mathcal{U}_j^p < \mathcal{U}_{\max}^p - \epsilon k \\
&\Rightarrow \hat{x}_j^p = 0.
\end{aligned}$$

Taking $k = 1 + \frac{1}{\sqrt{\epsilon}}$, it follows that $\{\hat{x}_j^p\}_{j,p}$ is a $\sqrt{\epsilon} \cdot (\sqrt{\epsilon} + 1 + 4(r-1) \max\{u_{j_s}^p\})$ -Nash equilibrium. □

5 The Main Reduction

We prove our main result, namely

Theorem 12 *Both 3-NASH and 3-GRAPHICAL NASH are PPAD-complete.*

Proof. That 3-NASH is in PPAD follows from Theorem 1. That 3-GRAPHICAL NASH is in PPAD follows by reducing it to 3-NASH, by Theorem 4, and then invoking Theorem 1. We hence focus on establishing the PPAD-hardness of the problems.

The reduction is from the problem BROUWER defined in Section 3.3. Given an instance of BROUWER, that is a circuit C with $3n$ input bits and 2 output bits describing a Brouwer function as specified in Section 3.3, we construct a graphical game \mathcal{G} , with maximum degree three, that simulates the circuit C , and specify an accuracy ϵ , so that, given an ϵ -Nash equilibrium of \mathcal{G} , one can find in polynomial time a panchromatic vertex of the BROUWER instance. Then, since, by Theorem 4, 3-GRAPHICAL NASH reduces to 3-NASH, this completes the proof.

The graphical game \mathcal{G} that we construct will be *binary*, in that each vertex v in it will have two strategies, and thus, at equilibrium, will represent a real number in $[0, 1]$, denoted $\mathbf{p}[v]$. (Letting 0 and 1 denote the strategies, $\mathbf{p}[v]$ is the probability that v plays 1.) There will be three distinguished vertices v_x, v_y , and v_z which will represent the coordinates of a point in the three dimensional cube and the construction will guarantee that in any Nash equilibrium of game \mathcal{G} this point will be close to a panchromatic vertex of the given BROUWER instance.

The building blocks of \mathcal{G} will be the game-gadgets $\mathcal{G}_\alpha, \mathcal{G}_{\times\alpha}, \mathcal{G}_=, \mathcal{G}_+, \mathcal{G}_-, \mathcal{G}_*$ that we constructed in Section 4.1 plus a few new gadgets. Recall from Propositions 1, 2 and 4, Figures 4, 3 and 5, that

Lemma 17 *There exist binary graphical games \mathcal{G}_α , where α is any rational in $[0, 1]$, $\mathcal{G}_{\times\alpha}$, where α is any non-negative rational, $\mathcal{G}_=, \mathcal{G}_+, \mathcal{G}_-, \mathcal{G}_*$, with at most four players a, b, c, d each, such that, in all games, the payoffs of a and b do not depend on the choices of the other vertices c, d , and, for $\epsilon < 1$,*

1. *in every ϵ -Nash equilibrium of game \mathcal{G}_α , we have $\mathbf{p}[d] = \alpha \pm \epsilon$;*
2. *in every ϵ -Nash equilibrium of game $\mathcal{G}_{\times\alpha}$, we have $\mathbf{p}[d] = \min(1, \alpha\mathbf{p}[a]) \pm \epsilon$;*
3. *in every ϵ -Nash equilibrium of game $\mathcal{G}_=$, we have $\mathbf{p}[d] = \mathbf{p}[a] \pm \epsilon$;*
4. *in every ϵ -Nash equilibrium of game \mathcal{G}_+ , we have $\mathbf{p}[d] = \min\{1, \mathbf{p}[a] + \mathbf{p}[b]\} \pm \epsilon$;*
5. *in every ϵ -Nash equilibrium of game \mathcal{G}_- , we have $\mathbf{p}[d] = \max\{0, \mathbf{p}[a] - \mathbf{p}[b]\} \pm \epsilon$;*
6. *in every ϵ -Nash equilibrium of game \mathcal{G}_* , we have $\mathbf{p}[d] = \mathbf{p}[a] \cdot \mathbf{p}[b] \pm \epsilon$;*

where by $x = y \pm \epsilon$ we denote $y - \epsilon \leq x \leq y + \epsilon$.

Let us, further, define a comparator game $\mathcal{G}_<$.

Lemma 18 *There exists a binary graphical game $\mathcal{G}_<$ with three players a, b and d such that the payoffs of a and b do not depend on the choices of d and, in every ϵ -Nash equilibrium of the game, with $\epsilon < 1$, it holds that $\mathbf{p}[d] = 1$, if $\mathbf{p}[a] < \mathbf{p}[b] - \epsilon$, and $\mathbf{p}[d] = 0$, if $\mathbf{p}[a] > \mathbf{p}[b] + \epsilon$.*

Proof. Let us define the payoff table of player d as follows: d receives a payoff of 1 if d plays 0 and a plays 1, and d receives a payoff of 0 if d plays 1 and b plays 1, otherwise d receives a payoff of 0. Equivalently, d receives an expected payoff of $\mathbf{p}[a]$, if d plays 0, and an expected payoff of $\mathbf{p}[b]$, if d plays 1. It immediately follows that, if in an ϵ -Nash equilibrium $\mathbf{p}[a] < \mathbf{p}[b] - \epsilon$, then $\mathbf{p}[d] = 1$, whereas, if $\mathbf{p}[a] > \mathbf{p}[b] + \epsilon$, $\mathbf{p}[d] = 0$. \square

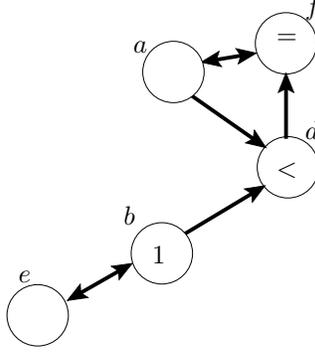


Figure 13: Brittleness of Comparator Games.

Notice that, in $\mathcal{G}_{<}$, $\mathbf{p}[d]$ is arbitrary if $|\mathbf{p}[a] - \mathbf{p}[b]| \leq \epsilon$; hence we call it the *brittle comparator*. As an aside, it is not hard to see that a robust comparator, one in which d is guaranteed, in an exact Nash equilibrium, to be, say, 0 if $\mathbf{p}[a] = \mathbf{p}[b]$, cannot exist, since it could be used to produce a simple graphical game with no Nash equilibrium, contradicting Nash's theorem. For completeness we present such a game in Figure 13, where vertices e and b constitute a \mathcal{G}_1 game so that, in any Nash equilibrium, $\mathbf{p}[b] = 1$, vertices d, f, a constitute a $\mathcal{G}_=$ game so that, in any Nash equilibrium, $\mathbf{p}[a] = \mathbf{p}[d]$ and vertices a, b, d constitute a comparator game with the hypothetical behavior that $\mathbf{p}[d] = 1$, if $\mathbf{p}[a] < \mathbf{p}[b]$ and $\mathbf{p}[d] = 0$, if $\mathbf{p}[a] \geq \mathbf{p}[b]$. Then it is not hard to argue that the game of Figure 13 does not have a Nash equilibrium contrary to Nash's theorem: indeed if, in a Nash equilibrium, $\mathbf{p}[a] = 1$, then $\mathbf{p}[d] = 0$, since $\mathbf{p}[a] = 1 = \mathbf{p}[b]$, and so $\mathbf{p}[a] = \mathbf{p}[d] = 0$, by $\mathcal{G}_=$, a contradiction; on the other hand, if, in a Nash equilibrium, $\mathbf{p}[a] < 1$, then $\mathbf{p}[d] = 1$, since $\mathbf{p}[a] < 1 = \mathbf{p}[b]$, and so $\mathbf{p}[a] = \mathbf{p}[d] = 1$, by $\mathcal{G}_=$, again a contradiction.

To continue with our reduction from BROUWER to 3-GRAPHICAL NASH, we include the following vertices to the graphical game \mathcal{G} .

- the three coordinate vertices v_x, v_y, v_z ,
- for $i \in \{1, 2, \dots, n\}$, vertices $v_{b_i(x)}, v_{b_i(y)}$ and $v_{b_i(z)}$, whose \mathbf{p} -values correspond to the i -th most significant bit of $\mathbf{p}[v_x], \mathbf{p}[v_y], \mathbf{p}[v_z]$,
- for $i \in \{1, 2, \dots, n\}$, vertices v_{x_i}, v_{y_i} and v_{z_i} , whose \mathbf{p} -values correspond to the fractional number resulting from subtracting from $\mathbf{p}[v_x], \mathbf{p}[v_y], \mathbf{p}[v_z]$ the fractional numbers corresponding to the $i - 1$ most significant bits of $\mathbf{p}[v_x], \mathbf{p}[v_y], \mathbf{p}[v_z]$ respectively.

We can extract these values by computing the binary representation of $\lfloor \mathbf{p}[v_x] 2^n \rfloor$ and similarly for v_y and v_z , that is, the binary representations of the integers i, j, k such that $(x, y, z) = (\mathbf{p}[v_x], \mathbf{p}[v_y], \mathbf{p}[v_z])$ lies in the cubelet K_{ijk} . This is done by a graphical game that simulates, using the arithmetical gadgets of Lemmas 17 and 18, the following algorithm ($\langle (a, b) \rangle$ is 1 if $a \leq b$ and 0 if $a > b$):

```

 $x_1 = x;$ 
for  $i = 1, \dots, n$  do:
   $\{b_i(x) := \langle (2^{-i}, x_i); x_{i+1} := x_i - b_i(x) \cdot 2^{-i}\};$ 
similarly for  $y$  and  $z$ ;
```

This is accomplished in \mathcal{G} by connecting these vertices as prescribed by Lemmas 17 and 18, so that $\mathbf{p}[v_{x_i}], \mathbf{p}[v_{b_i(x)}]$, etc. approximate the value of $x_i, b_i(x)$ etc. as computed by the above algorithm. The following lemma (when applied with $m = n$) shows that this device properly decodes the first

n bits of the binary expansion of $x = \mathbf{p}[v_x]$, as long as x is not too close to a multiple of 2^{-n} (suppose $\epsilon \ll 2^{-n}$ to be fixed later).

Lemma 19 *For $m \leq n$, if $\sum_{i=1}^m b_i 2^{-i} + 3m\epsilon < \mathbf{p}[v_x] < \sum_{i=1}^m b_i 2^{-i} + 2^{-m} - 3m\epsilon$ for some $b_1, \dots, b_m \in \{0, 1\}$, then, in every ϵ -Nash equilibrium of \mathcal{G} , $\mathbf{p}[v_{b_j(x)}] = b_j$, and $\mathbf{p}[v_{x_{j+1}}] = \mathbf{p}[v_x] - \sum_{i=1}^j b_i 2^{-i} \pm 3j\epsilon$, for all $j \leq m$.*

Proof. The proof is by induction on j . For $j = 1$, the hypothesis $\sum_{i=1}^m b_i 2^{-i} + 3m\epsilon < \mathbf{p}[v_x] < \sum_{i=1}^m b_i 2^{-i} + 2^{-m} - 3m\epsilon$ implies, in particular, that

$$\frac{b_1}{2} + 3\epsilon \leq \sum_{i=1}^m b_i 2^{-i} + 3m\epsilon < \mathbf{p}[v_x] < \sum_{i=1}^m b_i 2^{-i} + 2^{-m} - 3m\epsilon \leq \frac{b_1}{2} + \frac{1}{2} - 3\epsilon$$

and, since $\mathbf{p}[v_{x_1}] = \mathbf{p}[v_x] \pm \epsilon$, it follows that

$$\frac{b_1}{2} + 2\epsilon < \mathbf{p}[v_{x_1}] < \frac{b_1}{2} + \frac{1}{2} - 2\epsilon.$$

By Lemma 18, this implies that $\mathbf{p}[v_{b_1(x)}] = b_1$; note that the preparation of the constant $\frac{1}{2}$ —against which a comparator game compares the value $\mathbf{p}[v_{x_1}]$ —is done via a $\mathcal{G}_{\frac{1}{2}}$ game which introduces an error of $\pm\epsilon$. For the computation of $\mathbf{p}[v_{x_2}]$, the multiplication of $\mathbf{p}[v_{b_1(x)}]$ by $\frac{1}{2}$ and the subtraction of the product from $\mathbf{p}[v_{x_1}]$ introduce an error of $\pm\epsilon$ each and, therefore, $\mathbf{p}[v_{x_2}] = \mathbf{p}[v_{x_1}] - b_1 \frac{1}{2} \pm 2\epsilon$. And, since $\mathbf{p}[v_{x_1}] = \mathbf{p}[v_x] \pm \epsilon$, it follows that $\mathbf{p}[v_{x_2}] = \mathbf{p}[v_x] - b_1 \frac{1}{2} \pm 3\epsilon$, as required.

Supposing that the claim holds up to $j - 1 \leq m - 1$, we will show that it holds for j . By the induction hypothesis, we have that $\mathbf{p}[v_{x_j}] = \mathbf{p}[v_x] - \sum_{i=1}^{j-1} b_i 2^{-i} \pm 3(j-1)\epsilon$. Combining this with $\sum_{i=1}^m b_i 2^{-i} + 3m\epsilon < \mathbf{p}[v_x] < \sum_{i=1}^m b_i 2^{-i} + 2^{-m} - 3m\epsilon$, it follows that

$$\sum_{i=j}^m b_i 2^{-i} + 3(m - (j - 1))\epsilon < \mathbf{p}[v_{x_j}] < \sum_{i=j}^m b_i 2^{-i} + 2^{-m} - 3(m - (j - 1))\epsilon$$

which implies

$$\frac{b_j}{2^j} + 2\epsilon < \mathbf{p}[v_{x_j}] < \frac{b_j}{2^j} + \frac{1}{2^j} - 2\epsilon.$$

Continue as in the base case. □

Assuming that $x = \mathbf{p}[v_x]$, $y = \mathbf{p}[v_y]$, $z = \mathbf{p}[v_z]$ are all at distance greater than $3n\epsilon$ from any multiple of 2^{-n} , the part of \mathcal{G} that implements the above algorithm computes i, j, k such that the point (x, y, z) lies in the cubelet K_{ijk} ; that is, there are $3n$ vertices of the game \mathcal{G} whose \mathbf{p} values are equal to the n bits of the binary representation of i, j, k . Once we have the binary representations of i, j, k , we can feed them into another part of \mathcal{G} that simulates the circuit C . We could simulate the circuit by having vertices that represent gates, using addition (with ceiling 1) to simulate *or*, multiplication for *and*, and $1 - x$ for negation. However, there is a simpler way, one that avoids the complications related to accuracy, to simulate Boolean functions under the assumption that the inputs are 0 or 1:

Lemma 20 *There are binary graphical games $\mathcal{G}_{\vee}, \mathcal{G}_{\wedge}, \mathcal{G}_{-}$ with two input players a, b (one input player a for \mathcal{G}_{-}) and an output player c such that the payoffs of a and b do not depend on the choices of c , and, at any ϵ -Nash equilibrium with $\epsilon < 1/4$ in which $\mathbf{p}[a], \mathbf{p}[b] \in \{0, 1\}$, $\mathbf{p}[c]$ is also in $\{0, 1\}$, and is in fact the result of applying the corresponding Boolean function to the inputs.*

Proof. These games are in the same spirit as $G_{<}$. In G_{\vee} , for example, the payoff to c is $1/2$ if it plays 0; if c plays 1 its payoff is 1 if at least one of a, b plays 1, and it is 0 if they both play 0. Similarly for G_{\wedge} and G_{-} . \square

It would seem that all we have to do now is to close the loop as follows: in addition to the part of \mathcal{G} that computes the bits of i, j, k , we could have a part that simulates circuit C in the neighborhood of K_{ijk} and decides whether among the vertices of the cubelet K_{ijk} there is a panchromatic one; if not, the vertices v_x, v_y and v_z could be incentivized to change their \mathbf{p} values, say in the direction $\delta_{C(i,j,k)}$, otherwise stay put. To simulate a circuit evaluation in \mathcal{G} we could have one vertex for each gate of the circuit so that, in any ϵ -Nash equilibrium in which all the $\mathbf{p}[v_{b_i(x)}]$'s are 0 – 1, the vertices corresponding to the outputs of the circuit also play pure strategies, and, furthermore, these strategies correspond correctly to the outputs of the circuit.

But, as we mentioned above, there is a problem: Because of the brittle comparators, at the boundaries of the cubelets the vertices that should represent the values of the bits of i, j, k hold in fact arbitrary reals and, therefore, so do the vertices that represent the outputs of the circuit, and this noise in the calculation can create spurious Nash equilibria. Suppose for example that (x, y, z) lies on the boundary between two cubelets that have color 1, i.e. their centers are assigned vector δ_1 by C , and none of these cubelets has a panchromatic vertex. Then there ought not to be a Nash equilibrium with $\mathbf{p}[v_x] = x, \mathbf{p}[v_y] = y, \mathbf{p}[v_z] = z$. We would want that, when $\mathbf{p}[v_x] = x, \mathbf{p}[v_y] = y, \mathbf{p}[v_z] = z$, the vertices v_x, v_y, v_z have the incentive to shift their \mathbf{p} values in direction δ_1 , so that v_x prefers to increase $\mathbf{p}[v_x]$. However, on a boundary between two cubelets, some of the “bit values” that get loaded into the vertices $v_{b_i(x)}$, could be other than 0 and 1, and then there is nothing we can say about the output of the circuit that processes these values.

To overcome this difficulty, we resort to the following *averaging maneuver*: We repeat the above computation not just for the point (x, y, z) , but also for all $M = (2m + 1)^3$ points of the form $(x + p \cdot \alpha, y + q \cdot \alpha, z + s \cdot \alpha)$, for $-m \leq p, q, s \leq m$, where m is a large enough constant to be fixed later (we show below that $m = 20$ is sufficient). The vertices v_x, v_y, v_z are then incentivized to update their values according to the consensus of the results of these computations, most of which are reliable, as we shall show next.

Let us first describe this averaging in more detail. It will be convenient to assume that the output of C is a little more explicit than 3 bits: let us say that C computes six bits $\Delta x^+, \Delta x^-, \Delta y^+, \Delta y^-, \Delta z^+, \Delta z^-$, such that at most one of $\Delta x^+, \Delta x^-$ is 1, at most one of $\Delta y^+, \Delta y^-$ is 1, and similarly for z , and the increment of the Brouwer function at the center of K_{ijk} is $\alpha \cdot (\Delta x^+ - \Delta x^-, \Delta y^+ - \Delta y^-, \Delta z^+ - \Delta z^-)$, equal to one of the vectors $\delta_0, \delta_1, \delta_2, \delta_3$ specified in the definition of BROWER, where recall $\alpha = 2^{-2n}$.

The game \mathcal{G} has the following structure: Starting from (x, y, z) , some part of the game is devoted to calculating the points $(x + p \cdot \alpha, y + q \cdot \alpha, z + s \cdot \alpha)$, $-m \leq p, q, s \leq m$. Then, another part evaluates the circuit C on the binary representation of each of these points yielding $6M$ output bits, $\Delta x_1^+, \dots, \Delta z_M^-$. A final part calculates the following averages

$$(\delta x^+, \delta y^+, \delta z^+) = \frac{\alpha}{M} \sum_{t=1}^M (\Delta x_t^+, \Delta y_t^+, \Delta z_t^+), \quad (29)$$

$$(\delta x^-, \delta y^-, \delta z^-) = \frac{\alpha}{M} \sum_{t=1}^M (\Delta x_t^-, \Delta y_t^-, \Delta z_t^-), \quad (30)$$

which correspond to the average positive, respectively negative, shift of all M points.

We have already described above how to implement the bit extraction and the evaluation of a circuit using the gadgets of Lemmas 17, 18 and 20. The computation of points $(x + p \cdot \alpha, y + q \cdot \alpha, z + s \cdot \alpha)$, for all $-m \leq p, q, s \leq m$, is also easy to implement by preparing the values $\alpha|p|$, $\alpha|q|$, $\alpha|s|$, using gadgets $\mathcal{G}_{\alpha|p|}$, $\mathcal{G}_{\alpha|q|}$, $\mathcal{G}_{\alpha|s|}$, and then adding or subtracting the results to x , y and z respectively, depending on whether p is positive or not and similarly for q and s . Of course, these computations are subject to truncations at 0 and 1 (see Lemma 17).

To implement the averaging of Equations (29) and (30) we must be careful on the order of operations. Specifically, we first have to multiply the 6 outputs, Δx_t^+ , Δx_t^- , Δy_t^+ , Δy_t^- , Δz_t^+ , Δz_t^- , of each circuit evaluation by $\frac{\alpha}{M}$ using the $\mathcal{G}_{\times \frac{\alpha}{M}}$ gadget and, having done so, we then implement the additions (29) and (30). Since α will be a very small constant, by doing so we avoid undesired truncations at 0 and 1.

We can now close the loop by inserting equality, addition and subtraction gadgets, $\mathcal{G}_=$, \mathcal{G}_+ , \mathcal{G}_- , that force, at equilibrium, x to be equal to $(x' + \delta x^+) - \delta x^-$, where x' is a copy of x created using $\mathcal{G}_=$, and similarly for y and z . Note that in \mathcal{G} we respect the order of operations when implementing $(x' + \delta x^+) - \delta x^-$ to avoid undesired truncations at 0 or 1 as we shall see next. This concludes the reduction; it is clear that it can be carried out in polynomial time.

Our proof is concluded by the following claim. For the following lemma we choose $\epsilon = \alpha^2$. Recall from our definition of BROWER that $\alpha = 2^{-2n}$.

Lemma 21 *In any ϵ -Nash equilibrium of the game \mathcal{G} , one of the vertices of the cubelet(s) that contain $(\mathbf{p}[v_x], \mathbf{p}[v_y], \mathbf{p}[v_z])$ is panchromatic.*

Proof. We start by pointing out a simple property of the increments $\delta_0, \dots, \delta_3$:

Lemma 22 *Suppose that for nonnegative integers k_0, \dots, k_3 all three coordinates of $\sum_{i=0}^3 k_i \delta_i$ are smaller in absolute value than $\frac{\alpha K}{5}$ where $K = \sum_{i=0}^3 k_i$. Then all four k_i are positive.*

Proof. For the sake of contradiction, suppose that $k_1 = 0$. It follows that $k_0 < K/5$ (otherwise the negative x coordinate of $\sum_{i=0}^3 k_i \delta_i$ would be too large), and thus one of k_2, k_3 is larger than $2K/5$, which makes the corresponding coordinate of $\sum_{i=0}^3 k_i \delta_i$ too large, a contradiction. Similarly if $k_2 = 0$ or $k_3 = 0$. Finally, if $k_0 = 0$ then one of k_1, k_2, k_3 is at least $K/3$ and the associated coordinate of $\sum_{i=0}^3 k_i \delta_i$ at least $\alpha K/3$, again a contradiction. \square

Let us denote by $v_{\delta x^+}$, $\{v_{\Delta x_t^+}\}_{1 \leq t \leq M}$ the vertices of \mathcal{G} that represent the values δx^+ , $\{\Delta x_t^+\}_{1 \leq t \leq M}$. To implement the averaging

$$\delta x^+ = \frac{\alpha}{M} \sum_{t=1}^M \Delta x_t^+$$

inside \mathcal{G} , we first multiply each $\mathbf{p}[v_{\Delta x_t^+}]$ by $\frac{\alpha}{M}$ using a $\mathcal{G}_{\times \frac{\alpha}{M}}$ gadget and we then sum the results by a sequence of addition gadgets. Since each of these operations induces an error of $\pm \epsilon$ and there are $2M - 1$ operations it follows that

$$\mathbf{p}[v_{\delta x^+}] = \frac{\alpha}{M} \sum_{t=1}^M \mathbf{p}[v_{\Delta x_t^+}] \pm (2M - 1)\epsilon. \quad (31)$$

Similarly, denoting by $v_{\delta x^-}$, $\{v_{\Delta x_t^-}\}_{1 \leq t \leq M}$ the vertices of \mathcal{G} that represent the values δx^- , $\{\Delta x_t^-\}_{1 \leq t \leq M}$, it follows that

$$\mathbf{p}[v_{\delta x^-}] = \frac{\alpha}{M} \sum_{t=1}^M \mathbf{p}[v_{\Delta x_t^-}] \pm (2M - 1)\epsilon, \quad (32)$$

and similarly for directions y and z .

We continue the proof by distinguishing two subcases for the location of $(x, y, z) = (\mathbf{p}[v_x], \mathbf{p}[v_y], \mathbf{p}[v_z])$

- (a) the point $(\mathbf{p}[v_x], \mathbf{p}[v_y], \mathbf{p}[v_z])$ is further than $(m+1)\alpha$ from every face of the cube $[0, 1]^3$,
- (b) the point $(\mathbf{p}[v_x], \mathbf{p}[v_y], \mathbf{p}[v_z])$ is at distance at most $(m+1)\alpha$ from some face of the cube $[0, 1]^3$.

Case (a): Denoting by $v_{x+p\cdot\alpha}$ the player of \mathcal{G} that represents $x + p \cdot \alpha$, the small value of ϵ relative to α implies that at most one of the values $\mathbf{p}[v_{x+p\cdot\alpha}]$, $-m \leq p \leq m$, can be $3n\epsilon$ -close to a multiple of 2^{-n} , and similarly for the directions y and z . Indeed, recall that $x + p \cdot \alpha$ is computed from x by first preparing the value $|p|\alpha$ via a $\mathcal{G}_{|p|\alpha}$ gadget and then adding or subtracting the result to x —depending on whether p is positive or not—using \mathcal{G}_+ or \mathcal{G}_- . It follows that

$$\mathbf{p}[v_{x+p\cdot\alpha}] = \mathbf{p}[v_x] + p \cdot \alpha \pm 2\epsilon, \quad (33)$$

since each gadget introduces an error of $\pm\epsilon$, where note that there are no truncations at 0 or 1, because, by assumption, $(m+1)\alpha < \mathbf{p}[v_x] < 1 - (m+1)\alpha$. Consequently, for $p > p'$,

$$\mathbf{p}[v_{x+p\cdot\alpha}] - \mathbf{p}[v_{x+p'\cdot\alpha}] \geq (p - p') \cdot \alpha - 4\epsilon > 6n\epsilon,$$

and, moreover,

$$\mathbf{p}[v_{x+m\cdot\alpha}] - \mathbf{p}[v_{x-m\cdot\alpha}] \leq 2m \cdot \alpha + 4\epsilon \ll 2^{-n}, \quad (34)$$

since m is a constant, $\alpha = 2^{-2n}$, $\epsilon = \alpha^2$, and n is assumed to be large enough. Hence, from among the $M = (2m+1)^3$ circuit evaluations, all but at most $3(2m+1)^2$, or at least $K = (2m-2)(2m+1)^2$, compute legitimate, i.e. binary, Δx^+ etc. values.

Let us denote by $\mathcal{K} \subseteq \{-m, \dots, m\}^3$, $|\mathcal{K}| \geq K$, the set of values (p, q, r) for which the bit extraction from $(\mathbf{p}[v_{x+p\cdot\alpha}], \mathbf{p}[v_{y+q\cdot\alpha}], \mathbf{p}[v_{z+r\cdot\alpha}])$ results in binary outputs and, consequently, so does the circuit evaluation. Let

$$S_{\mathcal{K}} = \frac{\alpha}{M} \sum_{t \in \mathcal{K}} (\mathbf{p}[v_{\Delta x_t^+}] - \mathbf{p}[v_{\Delta x_t^-}], \mathbf{p}[v_{\Delta y_t^+}] - \mathbf{p}[v_{\Delta y_t^-}], \mathbf{p}[v_{\Delta z_t^+}] - \mathbf{p}[v_{\Delta z_t^-}]), \quad (35)$$

$$S_{\mathcal{K}^c} = \frac{\alpha}{M} \sum_{t \notin \mathcal{K}} (\mathbf{p}[v_{\Delta x_t^+}] - \mathbf{p}[v_{\Delta x_t^-}], \mathbf{p}[v_{\Delta y_t^+}] - \mathbf{p}[v_{\Delta y_t^-}], \mathbf{p}[v_{\Delta z_t^+}] - \mathbf{p}[v_{\Delta z_t^-}]). \quad (36)$$

Recall that we have inserted gadgets \mathcal{G}_+ , \mathcal{G}_- and $\mathcal{G}_=$ in \mathcal{G} to enforce that in a Nash equilibrium $x = x' + \delta x^+ - \delta x^-$, where x' is a copy of x . Because of the defection of the gadgets this will not be exactly tight in an ϵ -Nash equilibrium. More precisely, denoting by $v_{x'}$ the player of \mathcal{G} corresponding to x' , the following are true in an ϵ -Nash equilibrium

$$\begin{aligned} \mathbf{p}[v_{x'}] &= \mathbf{p}[v_x] \pm \epsilon \\ \mathbf{p}[v_x] &= \mathbf{p}[v_{x'}] + \mathbf{p}[v_{\delta x^+}] - \mathbf{p}[v_{\delta x^-}] \pm 2\epsilon, \end{aligned}$$

where for the second observe that both $\mathbf{p}[v_{\delta x^+}]$ and $\mathbf{p}[v_{\delta x^-}]$ are bounded above by $\alpha + (2M-1)\epsilon$ so there will be no truncations at 0 or 1 when adding $\mathbf{p}[v_{\delta x^+}]$ to $\mathbf{p}[v_{x'}]$ and then subtracting $\mathbf{p}[v_{\delta x^-}]$. By combining the above we get

$$\mathbf{p}[v_{\delta x^+}] - \mathbf{p}[v_{\delta x^-}] = \pm 3\epsilon$$

and, similarly, for y and z

$$\mathbf{p}[v_{\delta y^+}] - \mathbf{p}[v_{\delta y^-}] = \pm 3\epsilon$$

$$\mathbf{p}[v_{\delta z^+}] - \mathbf{p}[v_{\delta z^-}] = \pm 3\epsilon.$$

Now, if we use (31), (32), (35), (36) we derive

$$|S_{\mathcal{K}_\ell} + S_{\mathcal{K}^c_\ell}| \leq (4M + 1)\epsilon, \text{ for } \ell = x, y, z,$$

where $S_{\mathcal{K}_\ell}$, $S_{\mathcal{K}^c_\ell}$ is the ℓ coordinate of $S_{\mathcal{K}}$, $S_{\mathcal{K}^c}$. Moreover, since $|\mathcal{K}| \geq K$, the summation $S_{\mathcal{K}^c_\ell}$ has at most $M - K$ summands and because each of them is at most $\frac{\alpha}{M}$ in absolute value it follows that $|S_{\mathcal{K}^c_\ell}| \leq \frac{\alpha}{M}(M - K)$, for all $\ell = x, y, z$. Therefore, we have that

$$|S_{\mathcal{K}_\ell}| \leq (4M + 1)\epsilon + \frac{M - K}{M}\alpha, \text{ for } \ell = x, y, z.$$

Finally, note by the definition of the set \mathcal{K} that, for all $(p, q, r) \in \mathcal{K}$, the bit extraction from $(\mathbf{p}[v_{x+p\cdot\alpha}], \mathbf{p}[v_{y+q\cdot\alpha}], \mathbf{p}[v_{z+r\cdot\alpha}])$ and the following circuit evaluation result in binary outputs. Therefore, $S_{\mathcal{K}} = \frac{1}{M} \sum_{i=0}^3 k_i \delta_i$ for some nonnegative integers k_0, \dots, k_3 adding up to $|\mathcal{K}|$. From the above we get that

$$\left| \sum_{i=0}^3 k_i \delta_i \right|_{\infty} \leq (4M + 1)M\epsilon + (M - K)\alpha \leq (4M + 1)M\epsilon + 3(2m + 1)^2\alpha.$$

By choosing $m = 20$, the bound becomes less than $\alpha K/5$, and so Lemma 22 applies. It follows that, among the results of the $|\mathcal{K}|$ circuit computations, all four $\delta_0, \dots, \delta_3$ appeared. And, since every point on which the circuit C is evaluated is within ℓ_1 distance at most $3m\alpha + 6\epsilon \ll 2^{-n}$ from the point (x, y, z) , as Equation (33) dictates, this implies that among the corners of the cubelet(s) containing (x, y, z) there must be one panchromatic corner, completing the proof of Lemma 21 for case (a).

Case (b): We will show that there is no ϵ -Nash equilibrium in which $(\mathbf{p}[v_x], \mathbf{p}[v_y], \mathbf{p}[v_z])$ is within distance $(m + 1)\alpha$ from a face of $[0, 1]^3$. We will argue so only for the case

$$\mathbf{p}[v_x] \leq (m + 1)\alpha,$$

$$(m + 1)\alpha < \mathbf{p}[v_y] < 1 - (m + 1)\alpha,$$

$$(m + 1)\alpha < \mathbf{p}[v_z] < 1 - (m + 1)\alpha;$$

the other cases follow similarly.

First, we show that, for all $-m \leq p \leq m$, the bit extraction from $\mathbf{p}[v_{x+p\cdot\alpha}]$ results in binary outputs. From the proof of Lemma 19 it follows that, to show this, it is enough to establish that $\mathbf{p}[v_{x+p\alpha}] < 2^{-n} - 3n\epsilon$, for all p . Indeed, for $p \geq 0$, Equation (33) applies because there are no truncations at 1 at the addition gadget. So for $p \geq 0$ we get

$$\mathbf{p}[v_{x+p\alpha}] \leq \mathbf{p}[v_x] + p \cdot \alpha + 2\epsilon \leq (m + 1)\alpha + m\alpha + 2\epsilon \ll 2^{-n} - 3n\epsilon$$

On the other hand, for $p < 0$, there might be a truncation at 0 when we subtract the value $|p|\alpha$ from $\mathbf{p}[v_x]$. Nevertheless, we have that

$$\mathbf{p}[v_{x+p\alpha}] = \max\{0, \mathbf{p}[v_x] - (|p|\alpha \pm \epsilon)\} \pm \epsilon \leq \mathbf{p}[v_x] + 2\epsilon \leq (m + 1)\alpha + 2\epsilon \ll 2^{-n} - 3n\epsilon.$$

Therefore, for all $-m \leq p \leq m$, the bit extraction from $\mathbf{p}[v_{x+p\alpha}]$ is successful, i.e. results in binary outputs.

For the directions y and z the picture is exactly the same as in case (a) and, therefore, there exists at most one q , $-m \leq q \leq m$, and at most one r , $-m \leq r \leq m$, for which the bit extraction from $\mathbf{p}[v_{y+q.\alpha}]$ and $\mathbf{p}[v_{z+r.\alpha}]$ fails. Therefore, from among the $M = (2m + 1)^3$ points of the form $(\mathbf{p}[v_{x+p.\alpha}], \mathbf{p}[v_{y+q.\alpha}], \mathbf{p}[v_{z+r.\alpha}])$ the bit extraction succeeds in all but at most $2(2m + 1)^2$ of them.

Therefore, at least $K' = (2m - 1)(2m + 1)^2$ circuit evaluations are successful, i.e. in binary arithmetic, and, moreover, they correspond to points inside cubelets of the form K_{ijk} with $i = 0$. In particular, from Equation (34) and the analogous equations for the y and z coordinates, it follows that the successful circuit evaluations correspond to points inside at most 4 neighboring cubelets of the form K_{0jk} . Since these cubelets are adjacent to the $x = 0$ face of the cube, from the properties of the circuit C in the definition of the problem BROUWER, it follows that, among the outputs of these evaluations, one of the vectors $\delta_0, \delta_1, \delta_2, \delta_3$ is missing. Without loss of generality, let us assume that δ_0 is missing. Then, since there are K' successful evaluations, one of $\delta_1, \delta_2, \delta_3$ appears at least $K'/3$ times.

If this is vector δ_1 (similar argument applies for the cases δ_2, δ_3), then denoting by $v_{x'+\delta x^+}$ the player corresponding to $x' + \delta x^+$, the following should be true in an ϵ -Nash equilibrium.

$$\begin{aligned} \mathbf{p}[v_x] + \epsilon &\geq \mathbf{p}[v_{x'}] \geq \mathbf{p}[v_x] - \epsilon, \\ \alpha + (2M - 1)\epsilon &\geq \mathbf{p}[v_{\delta x^+}] \geq \frac{K'}{3M}\alpha - (2M - 1)\epsilon, \\ \mathbf{p}[v_{x'+\delta x^+}] &\geq \min(1, \mathbf{p}[v_{x'}] + \mathbf{p}[v_{\delta x^+}]) - \epsilon \geq \mathbf{p}[v_{x'}] + \mathbf{p}[v_{\delta x^+}] - \epsilon, \\ \frac{M - K'}{M}\alpha + (2M - 1)\epsilon &\geq \mathbf{p}[v_{\delta x^-}], \\ \mathbf{p}[v_x] &\geq \max(0, \mathbf{p}[v_{x'+\delta x^+}] - \mathbf{p}[v_{\delta x^-}]) - \epsilon \geq \mathbf{p}[v_{x'+\delta x^+}] - \mathbf{p}[v_{\delta x^-}] - \epsilon; \end{aligned}$$

in the second inequality of the third line above, we used that $\mathbf{p}[v_x] \leq (m + 1)\alpha$. Combining the above we get

$$\begin{aligned} \mathbf{p}[v_x] &\geq \mathbf{p}[v_{x'}] + \mathbf{p}[v_{\delta x^+}] - \mathbf{p}[v_{\delta x^-}] - 2\epsilon \\ &\geq \mathbf{p}[v_x] + \mathbf{p}[v_{\delta x^+}] - \mathbf{p}[v_{\delta x^-}] - 3\epsilon \end{aligned}$$

or equivalently that

$$\mathbf{p}[v_{\delta x^-}] \geq \mathbf{p}[v_{\delta x^+}] - 3\epsilon,$$

which implies

$$\frac{M - K'}{M}\alpha + (4M + 1)\epsilon \geq \frac{K'}{3M}\alpha,$$

which is not satisfied by our selection of parameters. □

To conclude the proof of Theorem 12, if we find any ϵ -Nash equilibrium of \mathcal{G} , Lemma 21 has shown that by reading off the first n binary digits of $\mathbf{p}[v_x]$, $\mathbf{p}[v_y]$ and $\mathbf{p}[v_z]$ we obtain a solution to the corresponding instance of BROUWER. □

6 Further Results and Open Problems

6.1 Two Players

Soon after our proof became available, Chen and Deng [7] showed that our PPAD-completeness result can be extended to the important two-player case. Here we present a rather simple modification of our proof from the previous section establishing this result.

Theorem 13 ([7]) *2-NASH is PPAD-complete.*

Proof. Let us define d -ADDITIVE GRAPHICAL NASH to be the problem d -GRAPHICAL NASH restricted to *bipartite graphical games with additive utility functions* defined next.

Definition 5 *Let \mathcal{GG} be a graphical game with underlying graph $G = (V, E)$. We call \mathcal{GG} a bipartite graphical game with additive utility functions if G is a bipartite graph and, moreover, for each vertex $v \in V$ and for every pure strategy $s_v \in S_v$ of that player, the expected payoff of v for playing the pure strategy s_v is a linear function of the mixed strategies of the vertices in $\mathcal{N}_v \setminus \{v\}$ with rational coefficients; that is, there exist rational numbers $\{\alpha_{u,s_u}^{s_v}\}_{u \in \mathcal{N}_v \setminus \{v\}, s_u \in S_u}$, $\alpha_{u,s_u}^{s_v} \in [0, 1]$ for all $u \in \mathcal{N}(v) \setminus \{v\}$, $s_u \in S_u$, such that the expected payoff to vertex v for playing pure strategy s_v is*

$$\sum_{u \in \mathcal{N}_v \setminus \{v\}, s_u \in S_u} \alpha_{u,s_u}^{s_v} \mathbf{p}[u : s_u],$$

where $\mathbf{p}[u : s_u]$ denotes the probability that vertex u plays pure strategy s_u .

The proof is based on the following lemmas.

Lemma 23 *BROUWER is poly-time reducible to 3-ADDITIVE GRAPHICAL NASH.*

Lemma 24 *3-ADDITIVE GRAPHICAL NASH is poly-time reducible to 2-NASH.*

Proof of Lemma 23: The reduction is almost identical to the one in the proof of Theorem 12. Recall that given an instance of BROUWER a graphical game was constructed using the gadgets $\mathcal{G}_\alpha, \mathcal{G}_{\times\alpha}, \mathcal{G}_=, \mathcal{G}_+, \mathcal{G}_-, \mathcal{G}_*, \mathcal{G}_\vee, \mathcal{G}_\wedge, \mathcal{G}_\neg$, and $\mathcal{G}_>$. In fact, gadget \mathcal{G}_* is not required, since only multiplication by a constant is needed which can be accomplished via the use of gadget $\mathcal{G}_{\times\alpha}$. Moreover, it is not hard to see by looking at the payoff tables of the gadgets defined in Section 4.1 and Lemma 18 that, in gadgets $\mathcal{G}_\alpha, \mathcal{G}_{\times\alpha}, \mathcal{G}_=, \mathcal{G}_+, \mathcal{G}_-$, and $\mathcal{G}_>$, the non-input vertices have the *additive utility functions* property of Definition 5. Let us further modify the games $\mathcal{G}_\vee, \mathcal{G}_\wedge, \mathcal{G}_\neg$ so that their output vertices have the *additive utility functions* property.

Lemma 25 *There are binary graphical games $\mathcal{G}_\vee, \mathcal{G}_\wedge, \mathcal{G}_\neg$ with two input players a, b (one input player a for \mathcal{G}_\neg) and an output player c such that the payoffs of a and b do not depend on the choices of c , c 's payoff satisfies the additive utility functions property, and, in any ϵ -Nash equilibrium with $\epsilon < 1/4$ in which $\mathbf{p}[a], \mathbf{p}[b] \in \{0, 1\}$, $\mathbf{p}[c]$ is also in $\{0, 1\}$, and is in fact the result of applying the corresponding Boolean function to the inputs.*

Proof. For \mathcal{G}_\vee , the payoff of player c is $0.5\mathbf{p}[a] + 0.5\mathbf{p}[b]$ for playing 1 and $\frac{1}{4}$ for playing 0. For \mathcal{G}_\wedge , the payoff of player c is $0.5\mathbf{p}[a] + 0.5\mathbf{p}[b]$ for playing 1 and $\frac{3}{4}$ for playing 0. For \mathcal{G}_\neg , the payoff of player c is $\mathbf{p}[a]$ for playing 0 and $\mathbf{p}[a : 0]$ for playing 1. \square

If the modified gadgets $\mathcal{G}_\vee, \mathcal{G}_\wedge, \mathcal{G}_\neg$ specified by Lemma 25 are used in the construction of Theorem 12, all vertices of the resulting graphical game satisfy the *additive utility functions* property of Definition 5. To make sure that the graphical game is also bipartite we modify the gadgets $\mathcal{G}_\vee, \mathcal{G}_\wedge, \mathcal{G}_\neg$, and $\mathcal{G}_>$ with the insertion of an extra output vertex. The modification is the same for all 4 gadgets: let c be the output vertex of any of these gadgets; we introduce a new output vertex e , whose payoff only depends on the strategy of c , but c 's payoff does not depend on the strategy of e , and such that the payoff of e is $\mathbf{p}[c]$ for playing 1 and $\mathbf{p}[c : 0]$ for playing 0 (i.e. e ‘‘copies’’ c , if c 's strategy is pure). It is not hard to see that, for every gadget, the new output vertex has the same behavior with regards to the strategies of the input vertices as the old output vertex, as specified by Lemmas 18 and 25. Moreover, it is not hard to verify that the graphical game resulting from the construction of Theorem 12 with the use of the modified gadgets $\mathcal{G}_\vee, \mathcal{G}_\wedge, \mathcal{G}_\neg$, and $\mathcal{G}_>$ is bipartite; indeed, it is sufficient to color blue the input and output vertices of all $\mathcal{G}_{\times\alpha}, \mathcal{G}_=, \mathcal{G}_+, \mathcal{G}_-, \mathcal{G}_\vee, \mathcal{G}_\wedge, \mathcal{G}_\neg$, and $\mathcal{G}_>$ gadgets used in the construction, blue the output vertices of all \mathcal{G}_α gadgets used, and red the remaining vertices. \square

Proof of Lemma 24: Let $\widetilde{\mathcal{G}\mathcal{G}}$ be a bipartite graphical game of maximum degree 3 with additive utility functions and $\mathcal{G}\mathcal{G}$ the graphical game resulting after rescaling all utilities to the set $[0, 1]$, e.g. by dividing all utilities by $\max\{\tilde{u}\}$, where $\max\{\tilde{u}\}$ is the largest entry in the payoff tables of game $\widetilde{\mathcal{G}\mathcal{G}}$. Also, let $\epsilon < 1$. In time polynomial in $|\mathcal{G}\mathcal{G}| + \log(1/\epsilon)$, we will specify a 2-player normal form 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 $\mathcal{G}\mathcal{G}$. This will be enough, since an ϵ -Nash equilibrium of $\mathcal{G}\mathcal{G}$ is trivially an $\epsilon \cdot \max\{\tilde{u}\}$ -Nash equilibrium of game $\widetilde{\mathcal{G}\mathcal{G}}$ and, moreover, $|\mathcal{G}\mathcal{G}|$ is polynomial in $|\widetilde{\mathcal{G}\mathcal{G}}|$.

The construction of \mathcal{G} from $\mathcal{G}\mathcal{G}$ is almost identical to the one described in Figure 7. Let $V = V_1 \sqcup V_2$ be the bipartition of the vertices of set V so that all edges are between a vertex in V_1 and a vertex in V_2 . Let us define $c : V \rightarrow \{1, 2\}$ as $c(v) = 1$ iff $v \in V_1$ and let us assume, without loss of generality, that $|v : c(v) = 1| = |v : c(v) = 2|$; otherwise, we can add to $\mathcal{G}\mathcal{G}$ isolated vertices to make up any shortfall. Suppose that n is the number of vertices in $\mathcal{G}\mathcal{G}$ (after the possible addition of isolated vertices) and t the cardinality of the strategy sets of the vertices in V , and let $\epsilon' = \epsilon/n$. Let us then employ the Steps 4 and 5 of the algorithm in Figure 7 to construct the normal form game \mathcal{G} from the graphical game $\mathcal{G}\mathcal{G}$; however, we choose $M = \frac{6tn}{\epsilon}$ and we modify Step 5b to read as follows

- (b)' for $v \in V$ and $s_v \in S_v$, if $c(v) = p$ and s contains (v, s_v) and (u, s_u) for some $u \in \mathcal{N}(v) \setminus \{v\}$, $s_u \in S_u$, then $u_s^p = \alpha_{u, s_u}^{s_v}$,

where we used the notation from Definition 5.

We argue next that, given an ϵ' -Nash equilibrium $\{x_{(v,a)}^p\}_{p,v,a}$ of \mathcal{G} , $\{x_a^v\}_{v,a}$ is an ϵ -Nash equilibrium of $\mathcal{G}\mathcal{G}$, where

$$x_a^v = x_{(v,a)}^{c(v)} / \sum_{j \in S_v} x_{(v,j)}^{c(v)}, \quad \forall v \in V, a \in S_v.$$

Suppose that $p = c(v)$ for some vertex v of the graphical game $\mathcal{G}\mathcal{G}$. As in the proof of Theorem 5, Lemma 7, it can be shown that in any ϵ' -Nash equilibrium of the game \mathcal{G} ,

$$\Pr(p \text{ plays } v) \in \left[\frac{2}{n} - \frac{1}{M}, \frac{2}{n} + \frac{1}{M} \right].$$

Now, without loss of generality assume that $p = 1$ (the pursuer) and suppose v is vertex $v_i^{(p)}$, in the notation of Figure 7. Then, in an ϵ' -Nash equilibrium of the game \mathcal{G} , we have, by the definition

of a Nash equilibrium, that for all strategies $s_v, s'_v \in S_v$ of vertex v :

$$\mathbb{E}[\text{payoff to } p \text{ for playing } (v, s_v)] > \mathbb{E}[\text{payoff to } p \text{ for playing } (v, s'_v)] + \epsilon' \Rightarrow x_{(v, s'_v)}^p = 0. \quad (37)$$

But

$$\mathbb{E}[\text{payoff to } p \text{ for playing } (v, s_v)] = M \cdot \Pr(p+1 \text{ plays } v_i^{(p+1)}) + \sum_{u \in \mathcal{N}_v \setminus \{v\}, s_u \in S_u} \alpha_{u, s_u}^{s_v} x_{(u, s_u)}^{c(u)}$$

and similarly for s'_v . Therefore, (37) implies

$$\sum_{u \in \mathcal{N}_v \setminus \{v\}, s_u \in S_u} \alpha_{u, s_u}^{s_v} x_{(u, s_u)}^{c(u)} > \sum_{u \in \mathcal{N}_v \setminus \{v\}, s_u \in S_u} \alpha_{u, s_u}^{s'_v} x_{(u, s_u)}^{c(u)} + \epsilon' \Rightarrow x_{(v, s'_v)}^p = 0. \quad (38)$$

Lemma 26 For all $v, a \in S_v$,

$$\left| x_a^v - \frac{x_{(v, a)}^{c(v)}}{2/n} \right| \leq \frac{n}{2M}.$$

Proof. $\left| x_a^v - \frac{x_{(v, a)}^{c(v)}}{2/n} \right| = \left| \frac{x_{(v, a)}^{c(v)}}{\Pr(c(v) \text{ plays } v)} - \frac{x_{(v, a)}^{c(v)}}{2/n} \right| = \frac{x_{(v, a)}^{c(v)}}{\Pr(c(v) \text{ plays } v)} \frac{|\Pr(c(v) \text{ plays } v) - 2/n|}{2/n} \leq \frac{n}{2M}$, where we used that $\sum_{j \in S_v} x_{(v, j)}^{c(v)} = \Pr(c(v) \text{ plays } v)$ and $|\Pr(c(v) \text{ plays } v) - 2/n| \leq \frac{1}{M}$. \square

By (38) and Lemma 26, we get that, for all $v \in V, s_v, s'_v \in S_v$,

$$\sum_{u \in \mathcal{N}_v \setminus \{v\}, s_u \in S_u} \alpha_{u, s_u}^{s_v} x_{s_u}^u > \sum_{u \in \mathcal{N}_v \setminus \{v\}, s_u \in S_u} \alpha_{u, s_u}^{s'_v} x_{s_u}^u + \frac{n}{2} \epsilon' + |\mathcal{N}_v \setminus \{v\}| t \frac{n}{M} \Rightarrow x_{s'_v}^v = 0.$$

Since $\frac{n}{2} \epsilon' + |\mathcal{N}_v \setminus \{v\}| t \frac{n}{M} \leq \epsilon$, it follows that $\{x_a^v\}_{v, a}$ is an ϵ -Nash equilibrium of the game $\mathcal{G}\mathcal{G}$. $\square \quad \square$

6.2 Approximate Nash Equilibria

Our proof establishes that it is PPAD-hard to find an approximate Nash equilibrium when the desired additive approximation ϵ is an inverse exponential in the size of the instance. What happens for larger ϵ 's? Chen, Deng and Teng show that, for any ϵ which is inverse polynomial in n , computing an ϵ -Nash equilibrium in 2-player games with n strategies per player remains PPAD-complete [9]; this is done by a modification of our reduction in which the starting BROUWER problem is defined not on the 3-dimensional cube, but in the n -dimensional hypercube. Intuitively, the difference is this: In order to create the exponentially many cells needed to embed the “line,” our construction had to resort to exponentially small cell size; in contrast, the n -dimensional hypercube contains exponentially many cells, all of reasonably large size.

The result of [9] implies that there is no *fully* polynomial time approximation scheme (a family of approximation algorithms that are polynomial in both the input size and $\frac{1}{\epsilon}$). But is there a polynomial time approximation scheme (family of polynomial algorithms with $\frac{1}{\epsilon}$ in the exponent)? This is a major open question that is left open.

And how about finitely large ϵ 's? Since the establishment of PPAD-completeness of NASH, we have seen a sequence of polynomial algorithms for finding ϵ -approximate Nash equilibria with $\epsilon = .5$ [17], $.39$ [18], $.37$ [4]; the best known ϵ at the time of writing is $.34$ [56].

6.3 Nash Equilibria in Graphical Games

Besides normal-form games, our work settles the complexity of computing a Nash equilibrium in graphical games of degree at most 3, again in the negative direction. Elkind, Goldberg and Goldberg show that a Nash equilibrium of graphical games with maximum degree 2 and 2 strategies per player can be computed in polynomial time [24]. Daskalakis and Papadimitriou describe a polynomial time approximation scheme for graphical games with a constant number of strategies per player, bounded degree and treewidth at most logarithmic in the number of players [20]. Can approximate Nash equilibria in general graphical games be computed efficiently?

6.4 Special Cases

Are there important and broad classes of games for which the Nash equilibrium problem can be solved efficiently? It has been shown that finding Nash equilibria in normal form games with all utilities either 1 or -1 (the so-called *win-lose games*) remains PPAD-complete [1, 10]. Rather surprisingly, it was also recently shown that, essentially, it is PPAD-complete to play even *repeated games* [3] (the so-called “Folk Theorem for repeated games” [52] notwithstanding).

On the positive side, Daskalakis and Papadimitriou [21, 22] develop a polynomial-time approximation scheme for *anonymous* games (games in which the utility of each player depends on her own strategy and the number of other players playing various strategies, but not the identities of these players), when the number of strategies per player is bounded. Although their algorithm is too inefficient to have a direct effect in practice, it does remove the intractability obstacle for a very large class of multiplayer games. Note that finding a Nash equilibrium in anonymous games is not known to be PPAD-complete.

6.5 Further Applications of our Techniques

What is the complexity of the Nash Equilibrium problem in other classes of succinctly representable games with many players (besides the graphical problems resolved in this paper)? For example, are these problems even in PPAD? (It is typically easy to see that they cannot be easier than the normal-form problem.) Daskalakis, Fabrikant and Papadimitriou give a general sufficient condition, satisfied by all known succinct representations of games, for membership of the Nash equilibrium problem in the class PPAD [15]. The basic idea is using the “arithmetical” gadgets in our present proof to simulate the calculation of utilities in these succinct games. However, whether computing a *sequential equilibrium* [46] in an extensive-form game is in PPAD is left open.

Our technique can be used to treat two other open problems in complexity. One is that of the complexity of simple stochastic games defined in [12], heretofore known to be in TFNP, but not in any of the more specialized classes like PPAD or PLS. Now, it is known that this problem is equivalent to evaluating combinational circuits with MAX, MIN, and AVERAGE gates. Since all three kinds of gates can be implemented by the graphical games in our construction, it follows that solving simple stochastic games is in PPAD.⁶

Similarly, by an explicit construction we can show the following.

Theorem 14 *Let $p : [0, 1] \rightarrow \mathbb{R}$ be any polynomial function such that $p(0) < 0$ and $p(1) > 0$. Then there exists a graphical game in which all vertices have two strategies, 0 and 1, and in which the mixed Nash equilibria correspond to a particular vertex v playing strategy 1 with probability equal to the roots of $p(x)$ between 0 and 1.*

⁶One has to pay some attention to the approximation; see [25] for details.

Proof Sketch. Let p be described by its coefficients $\alpha_0, \alpha_1, \dots, \alpha_n$, so that

$$p(x) := \alpha_n x^n + \alpha_{n-1} x^{n-1} + \dots + \alpha_1 x + \alpha_0.$$

Taking $A := (\sum_{i=0}^n |\alpha_i|)^{-1}$, it is easy to see that the range of the polynomial $q(x) := \frac{1}{2}Ap(x) + \frac{1}{2}$ is $[0, 1]$, that $q(0) < \frac{1}{2}$, $q(1) > 1/2$, and that every point $r \in [0, 1]$ such that $q(r) = \frac{1}{2}$ is a root of p . We define next a graphical game \mathcal{GG} in which all vertices have two strategies, 0 and 1, and a designated vertex v of \mathcal{GG} satisfies the following

- (i) in any mixed Nash equilibrium of \mathcal{GG} the probability x_1^v by which v plays strategy 1 satisfies $q(x_1^v) = 1/2$;
- (ii) for any root r of p in $[0, 1]$, there exists a mixed Nash equilibrium of \mathcal{GG} in which $x_1^v = r$;

The graphical game has the following structure:

- there is a component graphical game \mathcal{GG}_q with an “input vertex” v and an “output vertex” u such that, in any Nash equilibrium of \mathcal{GG} , the mixed strategies of u and v satisfy $x_1^u = q(x_1^v)$; a graphical game which progressively performs the computations required for the evaluation of $q(\cdot)$ on x_1^v can be easily constructed using our game-gadgets; note that the computations can be arranged in such an order that no truncations at 0 or 1 happen (recall the rescaling by $\frac{1}{2}A$ and the shifting around $1/2$ done above);
- a comparator game $\mathcal{G}_>$ (see Lemma 18) compares the mixed strategy of u with the value $\frac{1}{2}$, prepared by a $\mathcal{G}_{1/2}$ gadget (see Section 4.1), so that the output vertex of the comparator game plays 0 if $x_1^u > \frac{1}{2}$, 1 if $x_1^u < \frac{1}{2}$, and anything if $x_1^u = \frac{1}{2}$;
- we identify the output player of $\mathcal{G}_>$ with player v ;

It is not hard to see that \mathcal{GG} satisfies Properties (i) and (ii). □

As a corollary of Theorem 14, it follows that fixed points of polynomials can be computed by computing (exact) Nash equilibria of graphical games. Computing fixed points of polynomials via exact Nash equilibria in graphical games can be extended to the multi-variate case again via the use of game gadgets to evaluate the polynomial and the use of a series of $\mathcal{G}_=$ gadgets to set the output equal to the input.

Both this result and the result about simple stochastic games noted above were shown independently by [25], while Theorem 14 was already shown by Bubelis [5].

References

- [1] T. G. Abbott, D. Kane and P. Valiant. “On the Complexity of Two-Player Win-Lose Games,” In the 46th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2005.
- [2] P. Beame, S. Cook, J. Edmonds, R. Impagliazzo and T. Pitassi. “The Relative Complexity of NP Search Problems,” *Journal of Computer and System Sciences*, 57(1):13–19, 1998.
- [3] C. Borgs, J. Chayes, N. Immerlica, A. T. Kalai, V. Mirrokni and C. H. Papadimitriou. “The Myth of the Folk Theorem,” *In the 40th ACM Symposium on Theory of Computing*, STOC 2008.
- [4] H. Bosse, J. Byrka and E. Markakis. “New Algorithms for Approximate Nash Equilibria in Bimatrix Games,” *In the 3rd International Workshop on Internet and Network Economics*, WINE 2007.
- [5] V. Bubelis. “On Equilibria in Finite Games,” *International Journal of Game Theory*, 8(2):65–79, 1979.
- [6] G. J. Chang, W. Ke, D. Kuo, D. D. Liu and R. K. Yeh. “On $L(d, 1)$ -Labelings of Graphs,” *Discrete Mathematics*, 220(1–3): 57–66, 2000.
- [7] X. Chen and X. Deng. “3-NASH is PPAD-Complete,” *Electronic Colloquium in Computational Complexity*, TR05-134, 2005.
- [8] X. Chen and X. Deng. “Settling the Complexity of 2-Player Nash-Equilibrium,” *In the 47th Annual IEEE Symposium on Foundations of Computer Science*, FOCS 2006.
- [9] X. Chen, X. Deng and S. Teng. “Computing Nash Equilibria: Approximation and Smoothed Complexity,” *In the 47th Annual IEEE Symposium on Foundations of Computer Science*, FOCS 2006.
- [10] X. Chen, S. Teng and P. Valiant. “The Approximation Complexity of Win-Lose Games,” *In the 18th Annual ACM-SIAM Symposium On Discrete Algorithms*, SODA 2007.
- [11] B. Codenotti, A. Saberi, K. Varadarajan and Y. Ye. “Leontief Economies Encode Nonzero Sum Two-Player Games,” *In the 17th Annual ACM-SIAM Symposium On Discrete Algorithms*, SODA 2006.
- [12] A. Condon. “The Complexity of Stochastic Games,” *Information and Computation*, 96(2): 203–224, 1992.
- [13] V. Conitzer and T. Sandholm. “Complexity Results about Nash Equilibria,” *In the 18th International Joint Conference on Artificial Intelligence*, IJCAI 2003.
- [14] G. B. Dantzig. *Linear Programming and Extensions*, Princeton University Press, 1963.
- [15] C. Daskalakis, A. Fabrikant and C. H. Papadimitriou. “The Game World is Flat: The Complexity of Nash Equilibria in Succinct Games,” *In the 33rd International Colloquium on Automata, Languages and Programming*, ICALP 2006.
- [16] C. Daskalakis, P. W. Goldberg and C. H. Papadimitriou. “The Complexity of Computing a Nash Equilibrium,” *In the 38th ACM Symposium on Theory of Computing*, STOC 2006.

- [17] C. Daskalakis, A. Mehta and C. H. Papadimitriou. “A Note on Approximate Nash Equilibria,” *In the 2nd international Workshop on Internet and Network Economics*, WINE 2006.
- [18] C. Daskalakis, A. Mehta and C. H. Papadimitriou. “Progress in Approximate Nash Equilibria,” *In the 8th ACM Conference on Electronic Commerce*, EC 2007.
- [19] C. Daskalakis and C. H. Papadimitriou. “Three-Player Games Are Hard,” *Electronic Colloquium in Computational Complexity*, TR05-139, 2005.
- [20] C. Daskalakis and C. H. Papadimitriou. “Computing Pure Nash Equilibria via Markov Random Fields,” *In the 7th ACM Conference on Electronic Commerce*, EC 2006.
- [21] C. Daskalakis and C. H. Papadimitriou. “Computing Equilibria in Anonymous Games,” *In the 48th Annual IEEE Symposium on Foundations of Computer Science*, FOCS 2007.
- [22] C. Daskalakis and C. H. Papadimitriou. “Discretized Multinomial Distributions, Covers, and Nash Equilibria in Anonymous Games,” *ArXiv*, 2008.
- [23] B. C. Eaves. “Homotopies for Computation of Fixed Points,” *Mathematical Programming*, 3: 1–22, 1972.
- [24] E. Elkind, L. A. Goldberg and P. W. Goldberg. “Nash Equilibria in Graphical Games on Trees Revisited,” *In the 7th ACM Conference on Electronic Commerce*, EC 2006.
- [25] K. Etessami and M. Yannakakis. “On the Complexity of Nash Equilibria and Other Fixed Points,” *In the 48th Annual IEEE Symposium on Foundations of Computer Science*, FOCS 2007.
- [26] A. Fabrikant, C.H. Papadimitriou and K. Talwar. “The Complexity of Pure Nash Equilibria,” *In the 36th ACM Symposium on Theory of Computing*, STOC 2004.
- [27] C. B. Garcia, C. E. Lemke and H. J. Luthi. “Simplicial Approximation of an Equilibrium Point of Noncooperative N-Person Games,” *Mathematical Programming*, 4: 227–260, 1973.
- [28] J. Geanakoplos. “Nash and Walras Equilibrium via Brouwer,” *Economic Theory*, 21, 2003.
- [29] I. Gilboa and E. Zemel. “Nash and Correlated Equilibria: Some Complexity Considerations,” *Games and Economic Behavior*, 1(1): 80–93, 1989.
- [30] P. W. Goldberg and C. H. Papadimitriou. “Reducibility Among Equilibrium Problems,” *In the 38th ACM Symposium on Theory of Computing*, STOC 2006.
- [31] M. Hirsch, C. H. Papadimitriou and S. Vavasis. “Exponential Lower Bounds for Finding Brouwer Fixpoints,” *Journal of Complexity*, 5(4): 379–416, 1989.
- [32] D. S. Johnson, C. H. Papadimitriou and M. Yannakakis. “How Easy is Local Search?,” *Journal of Computer and System Sciences*, 37(1): 79–100, 1988.
- [33] M. Kearns, M. Littman and S. Singh. “Graphical Models for Game Theory,” *In the 17th Conference in Uncertainty in Artificial Intelligence*, UAI 2001.
- [34] L. G. Khachiyan. “A Polynomial Algorithm in Linear Programming,” *Soviet Mathematics Doklady*, 20(1): 191–194, 1979.

- [35] B. Knaster, C. Kuratowski and S. Mazurkiewicz. “Ein Beweis des Fixpunktsatzes für n-dimensionale Simplexe,” *Fundamenta Mathematicae*, 14: 132–137, 1929.
- [36] G. van der Laan and A. J. J. Talman. “A Restart Algorithm for Computing Fixed Points Without an Extra Dimension,” *Mathematical Programming*, 17: 74–84, 1979.
- [37] G. van der Laan and A. J. J. Talman. “On the Computation of Fixed Points in the Product Space of Unit Simplices and an Application to Noncooperative N Person Games,” *Mathematics of Operations Research*, 7(1): 1–13, 1982.
- [38] C. E. Lemke and J. T. Howson, Jr. “Equilibrium Points of Bimatrix Games,” *SIAM Journal of Applied Mathematics*, 12: 413–423, 1964.
- [39] R. Lipton and E. Markakis. “Nash Equilibria via Polynomial Equations,” *In the 6th Latin American Symposium, LATIN 2004*.
- [40] R. Lipton, E. Markakis and A. Mehta. “Playing Large Games Using Simple Strategies,” *In the 4th ACM Conference on Electronic Commerce, EC 2003*.
- [41] M. Littman, M. Kearns and S. Singh. “An Efficient, Exact Algorithm for Single Connected Graphical Games,” *In the 15th Annual Conference on Neural Information Processing Systems, NIPS 2001*.
- [42] N. Megiddo and C. H. Papadimitriou. “On Total Functions, Existence Theorems and Computational Complexity,” *Theoretical Computer Science*, 81(2): 317–324, 1991.
- [43] J. Nash. “Non-cooperative Games,” *Annals of Mathematics*, 54: 289–295, 1951.
- [44] J. von Neumann. “Zur Theorie der Gesellschaftsspiele,” *Mathematische Annalen*, 100: 295–320, 1928.
- [45] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*, Princeton University Press, 1944.
- [46] M.J. Osborne and A. Rubinstein. *A Course in Game Theory*, MIT Press, 1994.
- [47] C. H. Papadimitriou. *Computational Complexity*, Addison Wesley, 1994.
- [48] C. H. Papadimitriou. “On the Complexity of the Parity Argument and Other Inefficient Proofs of Existence,” *Journal of Computer and System Sciences*, 48(3): 498–532, 1994.
- [49] C. H. Papadimitriou. “Computing Correlated Equilibria in Multiplayer Games,” *In the 37th ACM Symposium on Theory of Computing, STOC 2005*.
- [50] C. H. Papadimitriou and T. Roughgarden. “Computing Equilibria in Multi-Player Games,” *In the 16th Annual ACM-SIAM Symposium On Discrete Algorithms, SODA 2005*.
- [51] J. Rosenmüller. “On a Generalization of the Lemke–Howson Algorithm to Noncooperative N-Person Games,” *SIAM Journal of Applied Mathematics*, 21(1): 73–79, 1971.
- [52] A. Rubinstein. “Equilibrium in Supergames with the Overtaking Criterion,” *Journal of Economic Theory*, 21: 1–9, 1979.

- [53] R. Savani and B. von Stengel. “Exponentially Many Steps for Finding a Nash Equilibrium in a Bimatrix Game,” *In the 45th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2004*.
- [54] H. E. Scarf. “The Approximation of Fixed Points of a Continuous Mapping,” *SIAM Journal of Applied Mathematics*, 15(5): 1328–1343, 1967.
- [55] G. Schoenebeck and S. Vadhan. “The Computational Complexity of Nash Equilibria in Concisely Represented Games,” *In the 7th ACM Conference on Electronic Commerce, EC 2006*.
- [56] H. Tsaknakis and P. G. Spirakis. “An Optimization Approach for Approximate Nash Equilibria,” *In the 3rd International Workshop on Internet and Network Economics, WINE 2007*.
- [57] R. Wilson. “Computing Equilibria of N-Person Games,” *SIAM Journal of Applied Mathematics*, 21: 80–87, 1971.