

PPAD Completeness of Equilibrium Computation

**Dissertation submitted to the Department of Mathematics of the London School of Economics
and Political Science for the degree of Master of Science**

Candidate Number: 59147

London, 26 August 2008

Contents

Introduction	1
1 Background	3
1.1 Games	3
1.2 Complexity Classes	6
1.3 r -NASH is in PPAD	9
2 Reductions among equilibrium problems	11
2.1 From GRAPHICAL NASH to NASH	11
2.2 From NASH to GRAPHICAL NASH	17
2.3 4-NASH	27
2.4 3-NASH	29
3 PPAD-completeness	36
3.1 3-DIMENSIONAL BROUWER	36
3.2 3-NASH and 3-GRAPHICAL NASH	40
3.3 2-NASH	46
Conclusion	49

Abstract

This dissertation relates some results assessing the computational complexity of the problem “given a game with r players, find one of its (approximate) Nash equilibria”. Using the graphical games introduced by Kearns, Littman and Singh ([KLS]) and a series of reductions proved by Goldberg and Papadimitriou ([GP]) we will show that finding an approximate Nash equilibrium of a r -player game is PPAD-complete for $r \geq 2$, following the results of Daskalakis, Goldberg and Papadimitriou ([DGP] and [DGP2]) and Daskalakis and Papadimitriou ([DP]).

Introduction

Nash's theorem guarantees that every finite normal-form game has a Nash equilibrium. But how hard is it to find at least one equilibrium of a game? That is: how far-fetched is the hypothesis of players with unlimited computing abilities? The value of the Nash equilibrium as an effective prediction of the behaviour of rational agents relies heavily on this question. The recent results by Goldberg and Papadimitriou ([GP]), Daskalakis, Goldberg and Papadimitriou ([DGP], [DGP2]), Daskalakis and Papadimitriou ([DP]), and by Chen and Deng ([CD3], [CD]) give an insight on its answer, settling the complexity of finding a Nash equilibrium in the class PPAD. This is the class introduced by Papadimitriou in [P94] of problems whose solution relies on the fact that in a directed graph with a source there must be a sink; PPAD is also a subclass of TFNP, the class, introduced by Megiddo and Papadimitriou in [MP], of the search problems for which a solution is guaranteed to exist. Contrarily to TFNP, though, PPAD is a class for which complete problems are known to exist: we shall see that finding a Nash equilibrium (more precisely, one of its approximations) is one of these.

To prove these results we will use mainly two tools: a particular type of games called *graphical games*, first introduced by Kearns, Littman and Singh in [KLS], and a series of reductions between equilibrium problems proved by Goldberg and Papadimitriou in [GP]. The graphical games are so called because in them the payoff of a player depends on the strategies chosen by only some of the other players: thus we can imagine the players that influence each others as connected by the edges of a graph. The reductions of [GP] show that we can prove results on the complexity of any r -player normal-form game or graphical game whose graph has maximum degree d , for $d \geq 3, r \geq 4$, by considering 4-player normal-form games or graphical games with maximum degree 3. In [GP] two useful techniques are also introduced: colouring the players of a graphical game and use each colour as a “super-player” of a new game, and using special graphical games, called *gadgets*, to simulate arithmetical functions.

After [GP], Daskalakis, Goldberg and Papadimitriou ([DGP]) showed that the problem “given a game with $r \geq 4$ players, find one of its approximate Nash equilibria” is PPAD-complete; then Daskalakis and Papadimitriou ([DP]) and Chen and Deng ([CD3]) proved independently that the same holds for the $r \geq 3$ case; finally Chen and Deng ([CD]) proved that the 2-player case is PPAD-complete as well. The proof of [DGP] is a reduction from 3-DIMENSIONAL BROUWER, a discrete version of the problem of finding the fixed points of a continuous function in the unit cube, to the problem of finding an approximate Nash equilibrium in a graphical game of maximum degree 3, thus, by the results of [GP], in a 4-player normal-form game. In [DP] a gadget of [GP] is modified so that the result of [DGP] implies a result on a 3-player normal-form game as well; on the other hand, [CD3] consider the game used in [GP] to reduce the problem of finding the equilibrium of a r -player normal-form game to the problem of finding the equilibrium of a 4-player normal-form game and use a “disconnectedness” of its graph to embed it in a 3-player game: contrarily to the result of [DP], this reduction does not preserve the exactness of the equilibria, but it does preserve the approximate equilibria. The main result of [CD] is a new reduction from 3-DIMENSIONAL BROUWER to the problem of finding an approximate Nash equilibrium of a 2-

player normal-form game, without passing through a graphical game and using the “disconnectedness” of newly designed 2-player gadgets. An alternative proof of the 2-player case is given in [DGP2]: in this 3-DIMENSIONAL BROUWER is reduced to the problem of finding an approximate Nash equilibrium of a graphical game satisfying a condition on its utility function, which in turn is reduced to the problem of finding an approximate Nash equilibrium of a 2-player normal-form game.

This dissertation relates the results proved in [GP], [DGP], [DP] and [DGP2]. The first section starts by introducing the definitions from game theory and computational complexity that we will use in the following sections; a proof of the Nash equilibrium of the 2-player “matching pennies” game is given, following the one for the 3-player case in [CD3], and a shortcoming of the concept of ϵ -Nash equilibrium is illustrated with an example. In the last subsection we give a sketch of the proof that finding a Nash equilibrium is PPAD, following [G07] and [DGP2]. The second section deals with the reductions among equilibrium problems seen in [GP] and [DP]; note that these hold for exact and approximate equilibria alike. The first subsection proves the reduction from the problem of finding an equilibrium of a graphical game of maximum degree d to the problem of finding an equilibrium of a $d^2 + 1$ -player normal-form game, with a modification to correct an error present in both [GP] and [DGP2] (a counter-example is also given) and a change to give a stronger result that reunites two cases in one. In the second subsection we show the reduction from the problem of finding an equilibrium of a r -player normal-form game to the problem of finding an equilibrium of a graphical game of maximum degree 3 using the gadgets technique; we also give a proof of the “equal to a real number” gadget that was not in the articles. In the third subsection we relate the main result of [GP], which shows that finding an equilibrium of a r -player normal-form game is as hard as finding an equilibrium of a 4-player normal-form game. In the last subsection, finally, we modify one of the gadgets of the previous sections as shown in [DP] so that we can have a reduction to the problem of finding an equilibrium of a 3-player normal-form game. The third and final section concerns the PPAD-completeness of finding a Nash equilibrium, following [DGP] and [DGP2]. The first subsection presents the problem 3-DIMENSIONAL BROUWER and gives a proof of its PPAD-completeness; the second subsection gives the reduction from 3-DIMENSIONAL BROUWER to the problem of finding an approximate Nash equilibrium in a graphical game of maximum degree 3, thus, for the results of the previous section, in any r -player normal-form game and any graphical game of maximum degree d , with $r, d \geq 3$. The last subsection relates the proof of the PPAD-completeness of the 2-player normal-form case as seen in [DGP2]. The conclusion gives some further literature and open problems.

1 Background

1.1 Games

A game is a model of a strategic interaction in which players choose a strategy that influences their and their opponents' payoff. Formally, a *normal-form game* is $\mathcal{G} = (P, S, u)$ where P is the set of *players*, S_p is the set of *strategies* for each player p and $S = \times_{p \in P} S_p$ is the set of *strategy profiles*, $u^p : S \rightarrow \mathbb{R}$ is the *payoff* for each player and $u = \times_{p \in P} u^p$. Each player chooses a strategy $s^p \in S_p$ or a probability distribution over his strategies $(x_1^p, \dots, x_{|S_p|}^p)$ with $x_i^p \geq 0$ and $\sum_i x_i^p = 1$ in order to maximise the expected value of his payoff: the latter is called a *mixed strategy*, the former a *pure strategy*. The set of strategies of players other than p is denoted S_{-p} . A game is *finite* if both the set of players and the sets of pure strategies are finite.

A *Nash equilibrium* of a game is a strategy profile in which no player can improve his expected payoff by unilaterally changing his strategy, that is a mixed profile in which each player maximises his expected payoff over all his mixed strategies. A necessary and sufficient condition for this is that for every $p \in P$ and every $j, j' \in S_p$

$$\sum_{s \in S_{-p}} u^p(j, s) x_s > \sum_{s \in S_{-p}} u^p(j', s) x_s \Rightarrow x_{j'}^p = 0$$

where for $s \in S_{-p}$ we denote $x_s = \prod_{q \in P \setminus \{p\}} x_{s_q}^q$. A fundamental theorem by Nash states

Theorem 1 (*Theorem 1 in [N51]*) *Every finite game has an equilibrium.*

A basic example of game, with two players with n strategies each, is *generalised matching pennies* described in [GP]. Both players have payoff zero unless they both play the same strategy: in that case player 1 (the *pursuer*) gets payoff M , player 2 (the *evader*) gets a payoff of $-M$. With $n = 2$ the game is

		p_2	
		s_1^2	s_2^2
p_1	s_1^1	$-M$ M	0 0
	s_2^1	0 0	$-M$ M

The game can be generalised to *3-player matching pennies* as shown in [CD3]: each player $p \in \{1, 2, 3\}$ has strategies (s_1^p, \dots, s_n^p) ; player 1 gets payoff M if he plays the same strategy of 2 or 3 and payoff zero otherwise, player 2 gets payoff $-M$ if he plays the same strategy of 1 or 3 and payoff zero otherwise, player 3 gets payoff M if he plays the same strategy of player 2 and payoff zero otherwise. In the two-strategy case the three players get payoffs $(u^1(s), u^2(s), u^3(s))$ as in figure:

		2	
		s_1^2	s_2^2
1	s_1^1	$(M, -M, M)$	$(M, 0, 0)$
	s_2^1	$(0, -M, M)$	$(M, -M, 0)$

3 plays s_1^3 :

		2	
		s_1^2	s_2^2
1	s_1^1	$(M, -M, 0)$	$(0, -M, M)$
	s_2^1	$(M, 0, 0)$	$(M, -M, M)$

3 plays s_2^3 :

The equilibrium of matching pennies is described in the following

Lemma 2 (see Lemma 1 of [CD3] for the 3-player case) *There is an unique equilibrium of both generalised matching pennies and 3-player matching pennies; at this equilibrium every player plays the uniform distribution over his strategies.*

PROOF.

2-player case

Suppose by contradiction that at the equilibrium x^2 is not the uniform distribution. Then we have two nonempty sets $U_2 = \{s_i^2 \in S_2 \mid x_i^2 > \frac{1}{n}\}$ and $L_2 = \{s_i^2 \in S_2 \mid x_i^2 \leq \frac{1}{n}\}$. Since we are at the equilibrium, we have that $x_l^1 = 0$ for every l such that $s_l^2 \in L_2$ – player 1 pursues player 2. Then there is $x_u^1 \neq 0$ for some u such that $s_u^2 \in U_2$. But then player 2, being the evader, will prefer any strategy in L to s_u^2 , which contradicts the definition of U_2 and L_2 .

On the other hand, if x^1 were not uniform we would have a nonempty set $U_1 = \{s_i^1 \in S_1 \mid x_i^1 > \frac{1}{n}\}$. But then we would have $x_u^2 = 0$ for every u such that $s_u^1 \in U_1$, and this would contradict the fact that x^2 follows the uniform distribution.

3-player case

Suppose by contradiction that at the equilibrium x^2 is not the uniform distribution. As before, then we have two nonempty sets $U_2 = \{s_i^2 \in S_2 \mid x_i^2 > \frac{1}{n}\}$ and $L_2 = \{s_i^2 \in S_2 \mid x_i^2 \leq \frac{1}{n}\}$. By the definition of Nash equilibrium we have that $x_l^1 = 0$ and $x_l^3 = 0$ for every l such that $s_l^2 \in L_2$. Then there is $x_u^1 \neq 0$ for some u such that $s_u^2 \in U_2$. But then player 2, being the evader, will prefer any strategy in L to s_u^2 , which contradicts the definition of U_2 and L_2 .

Now suppose that x^3 is not uniform. Then we have two nonempty sets $U_3 = \{s_i^3 \in S_3 \mid x_i^3 > \frac{1}{n}\}$ and $L_3 = \{s_i^3 \in S_3 \mid x_i^3 \leq \frac{1}{n}\}$. Since we are at the equilibrium and player 1 pursues player 3, we have that $x_l^1 = 0$ for every l such that $s_l^3 \in L_3$. But then player 2, being the evader of player 3, would prefer any s_l^2 with $s_l^3 \in L_3$ to any s_u^2 with $s_u^3 \in U_3$; but this contradicts the fact that x^2 is the uniform distribution.

Finally, the proof that x^1 is uniform is the same as the one in the 2-player case. \square

A strategy is called a *best response* if it maximises the player's utility over all the player's strategies. Notice that a mixed strategy is a Nash equilibrium if and only if all the strategies in its support are best responses to the other players' strategies – otherwise the player could be better off playing a best

response strategy. This, as noted in [P07], makes the task of finding a Nash equilibrium an “essentially combinatorial” one, since “players combine pure best response strategies in order to create for other players a range of best responses that will sustain the equilibrium”.

There is still a problem: if for the 2-player case rational payoffs guarantee that the mixed strategies at the equilibria will be rational, for 3 or more players this no longer holds: [N51] gives an example of 3-player game with rational payoffs that has only irrational mixed equilibria. We will thus need an approximated version of the Nash equilibrium concept. A strategy profile is a ε -Nash equilibrium of a game, for $\varepsilon > 0$, if for every $p \in P$ and every $j, j' \in S_p$

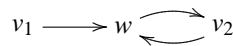
$$\sum_{s \in S_{-p}} u^p(j, s) x_s > \sum_{s \in S_{-p}} u^p(j', s) x_s + \varepsilon \Rightarrow x_{j'}^p = 0$$

where again we denote $x_s = \prod_{q \in P \setminus \{p\}} x_{s_q}^q$; that is, an ε -Nash equilibrium is a profile so that deviating from it improves the payoff of at most ε . Theorem 1 guarantees the existence of a ε -Nash equilibrium for every game, since every Nash equilibrium is also a ε -Nash equilibrium for any $\varepsilon > 0$. Notice, though, that an ε -Nash equilibrium is not always an approximation of a Nash equilibrium: an example is (s_2^1, s_2^2) in the game

		p_2	
		s_1^2	s_2^2
p_1	s_1^1	0	0
	s_2^1	0	0
	s_1^1	0	$-\varepsilon^2$
	s_2^1	0	$-\varepsilon^2$

In the next sections we will use a particular type of games, the *graphical games*, first introduced in [KLS]. These are games “played on a graph” $G = (V, E)$: the vertices $v \in V$ correspond to the players of the game, and there is an edge (v_1, v_2) if and only if the payoff of v_2 is influenced by the strategy chosen by v_1 or vice versa. More formally, following the definitions given in [GP]: a graphical game \mathcal{GG} is a undirected graph G , where each vertex $v \in V$ has an associated set of strategies S_v and utilities for mixed strategy profiles $\{u^v(s) : s \in S_{N(v) \cup \{v\}}\}$, where $N(v)$ is the set of v ’s neighbours and $S_{N(v) \cup \{v\}} = \times_{u \in N(v) \cup \{v\}} S_u$; note that [GP] assumes $v \in N(v)$ but we do not in order to simplify the notation in later sections. The *affects-graph* of \mathcal{GG} is the directed graph $G' = (V, E')$ where $(v_1, v_2) \in E'$ if and only if the payoff of v_2 depends on the action of v_1 . A definition given in [DP] that is crucial to the proof of its main result, which we will present in section 2.4, is that of *moralised graph*, the graph $G'' = (V, E'')$ built from the affects-graph as follows: $(v_1, v_2) \in E''$ if $(v_1, v_2) \in E'$, or if $(v_2, v_1) \in E'$, or if there is v_3 such that $(v_1, v_3) \in E'$ and $(v_2, v_3) \in E'$.

A nice example (taken from [DGP]) is the game played on the vertices v_1, v_2, w , with $S_p = \{0, 1\}$ for each player. The game’s affects-graph is



and its moralised graph is the 3-clique, since both (v_1, w) and (v_2, w) are arcs of the affects-graph. The payoffs for v_2 are

$v_2 \backslash w$	0	1
	0	1
1	1	0

and the payoffs for w are

$v_1 \backslash v_2$	0	1
	0	0
1	1	1

w plays 0:

$v_1 \backslash v_2$	0	1
	0	0
1	1	1

w plays 1:

$v_1 \backslash v_2$	0	1
	0	1
1	0	1

We will discuss the game's equilibria in section 2.2.

As noted in both [KLS] and [P07], graphical games have a very important propriety in terms of computational complexity: they are *succintly representable*. A normal-form game is a huge input: if there are r players and each of these players has n strategies the input (players, strategies, payoff matrices) will require rn^r numbers. The problem of finding a Nash equilibrium could be thus solved in polynomial time by checking all the possible supports of every mixed strategy, but the vastity of the space taken by the input would make this result no longer relevant. On the other hand, a graphical game with r players and underlying graph $G = (V, E)$ of maximum degree d can be described by an input of length rn^{d+1} . Notice how the complexity of a normal-form game depends mainly on the number of its players and the complexity of a graphical game depends mainly on the maximum degree of its graph: the idea will return in the next subsection.

1.2 Complexity Classes

We want to study the complexity of

n-NASH.

Input: an n -player normal-form game \mathcal{G} and an accuracy specification ε .

Output: an ε -Nash equilibrium of \mathcal{G} .

and

d-GRAPHICAL NASH.

Input: a graphical game \mathcal{GG} with graph of maximum degree d and an accuracy specification ε .

Output: an ε -Nash equilibrium of \mathcal{GG} .

We will also consider the problem “given a (graphical) game find one of its Nash equilibria”: notice how

this problem can be seen as a limit case, with $\varepsilon = 0$.

We could try to prove that these problems are NP-complete. But, as noted in [P07], the typical NP-completeness proof relies on the fact that solution to the problem may not exist, whereas Theorem 1 guarantees the existence of a Nash equilibrium.

Looking deeper into NP, we find the class TFNP (Total Function NP), first introduced in [MP]. One can think of TFNP as the class of NP problems whose answer is an example (not just “yes” or “no”) and for which an answer is always guaranteed to exist. Formally: let Σ be an alphabet with two or more symbols and $R \subseteq \Sigma^* \times \Sigma^*$ a polynomially balanced relation (that is, a relation such that $(x, y) \in R$ implies $|y| \leq p(|x|)$ for some polynomial p and the existence of a polynomial time Turing machine that verifies that $(x, y) \in R$). The *search problem* Π_R is then defined as: given $x \in \Sigma^*$ return $y \in \Sigma^*$ such that $(x, y) \in R$ if such a y exists, and “no” otherwise; FNP is the class of all search problems. R and Π_R are *total* if for every $x \in \Sigma^*$ there is always $y \in \Sigma^*$ such that $(x, y) \in R$; TFNP is the class of total search problems.

Here theorem 1 would make the class a perfect candidate for studying the problem of finding a Nash equilibrium, and indeed [MP] states that finding a Nash equilibrium of a r -player normal-form game is a problem in TFNP. But TFNP presents a problem: it is a *semantic class*, that is a class for which there are no complete problems¹. We will thus move to a subclass of TFNP, the class PPAD (Proof by Parity Argument, Directed version), first introduced in [P94] as the closure under reduction of the problems \mathcal{A} defined as follows.

Let M be a polynomial-time deterministic Turing machine; let x be an input for \mathcal{A} . The *configuration space* is $C(x) = \Sigma^{p(|x|)}$, the set of all strings of length at most polynomial in the size of x for a given polynomial p . Given $c \in C(x)$, the Turing machine M outputs in polynomial time $M(x, c)$, which is an ordered pair of configurations in $C(x)$, a single configuration or the empty set. We say that configuration c is the *predecessor* of configuration c' if c is the first component of $M(x, c')$ and c' is the second component of $M(x, c)$; in this case we also say that c' is the *successor* of c . Furthermore, we set M such that $M(x, 0 \cdots 0) = \{1 \cdots 1\}$ and $0 \cdots 0$ is the first component of $M(x, 1 \cdots 1)$. The relation “to be a predecessor” describes a graph $G(x)$ with vertices all the configurations c , such that for every configuration c we have $\text{indegree}(c) + \text{outdegree}(c) = |M(x, c)|$ and both $\text{indegree}(c)$ and $\text{outdegree}(c)$ are not greater than 1; furthermore we have one *standard source* $0 \cdots 0$. Problem \mathcal{A} associated with M is the search problem: “given x , find a node of $G(x)$ other than $0 \cdots 0$ with $\text{indegree} + \text{outdegree} = 1$ ”.

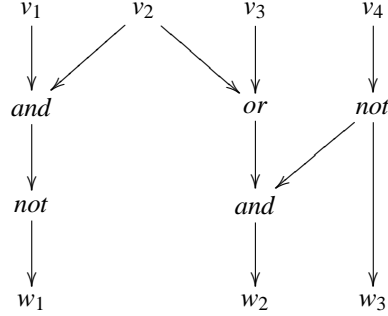
An equivalent and more practical definition of PPAD is the one given in [DGP]. A *polynomial-time reduction* is a polynomially computable function f from total search problem \mathcal{S} to total search problem \mathcal{T} such that for every input $x \in \mathcal{S}$ $f(x)$ is an input of \mathcal{T} and such that there is another function g , polynomially computable, that associates to every output $y \in \mathcal{T}_{f(x)}$ an output $g(y) \in \mathcal{S}_x$.

We define a *circuit* (see also [CR]) as a directed acyclic graph with n vertices with indegree 0 called *input nodes*, m vertices with outdegree 0 called *output nodes*, and internal nodes with indegree 1 or 2. When each input node receives an input in $\{0, 1\}$, the internal nodes with indegree 2 compute the Boolean functions *and* or *or* and the internal nodes with indegree 1 compute the Boolean function *not*, and finally

¹I am very grateful to C.H. Papadimitriou for clarifying this passage.

each output node returns a value in $\{0, 1\}$ accordingly.

We have for instance that the circuit with input bits v_1, \dots, v_4 and output bits w_1, w_2, w_3



returns $(1, 0, 0)$ when the nodes v_1, \dots, v_4 receive as input $(0, 1, 0, 1)$. Note that the input and output can also be interpreted as the binary representations of an integer in $[0, 2^n - 1]$ and $[0, 2^m - 1]$ respectively (in our example, these would be 5 and 4).

PPAD is defined as the class of problems reducible to

END OF THE LINE

Input: Two circuits S and P with n input bits and n output bits such that $P(0^n) = 0^n \neq S(0^n)$.

Output: An input x such that $P(S(x)) \neq x$ or $S(P(x)) \neq x \neq 0^n$

More informally, PPAD is the class of problems that are solved because in any directed graph with indegree and outdegree at most one if there is a source (a vertex with indegree zero and outdegree one) there must be a sink (a vertex with indegree one and outdegree zero) and there also might be another source, that is the class of problems that are solved by moving on a directed graph of maximum indegree and outdegree 1 to find a sink or a nonstandard source (a source other than 0^n). In the first definition the graph is $G(x)$; in END OF THE LINE, the graph is the one with all the n -bits inputs as vertices and an edge (x, y) whenever $S(x) = y$ and $P(y) = x$.

An example of a graph implicit in a PPAD problem could be

$$0^n \longrightarrow 1^n \longrightarrow c_1 \longrightarrow c_2 \quad c_3 \quad c_4 \rightleftarrows c_5$$

Looking at this graph, from the point of view of the first definition we have $M(x, 0 \dots 0) = \{1 \dots 1\}$, $M(x, 1 \dots 1) = (0 \dots 0, c_1)$, $M(x, c_1) = (1 \dots 1, c_2)$, $M(x, c_2) = \{c_1\}$; then $M(x, c_3) = \emptyset$ (nothing precedes nor follows c_3); then $M(x, c_4) = (c_5, c_5)$ and $M(x, c_5) = (c_4, c_4)$ (c_4 and c_5 form a cycle – they are both successor and predecessor of each other). From the point of view of the second definition, thinking of S and P as “successor” and “predecessor”, we have $P(0^n) = 0^n$ (nothing precedes 0^n), $S(0^n) = 1^n$, $P(1^n) = 0^n$ (1^n follows 0^n), $S(1^n) = c_1$, $P(c_1) = 0^n$, $S(c_1) = c_2$ (nothing follows c_2); then $P(c_3) = c_3$, $S(c_3) = c_3$; then $P(c_4) = c_5$, $S(c_4) = c_5$, $P(c_5) = c_4$, $S(c_5) = c_4$. The standard source of the graph is 0^n and its sink is c_2 .

1.3 r -NASH is in PPAD

Why do we need the class PPAD to deal with the complexity of finding a (ε) -Nash equilibrium?

As far as Nash equilibria are concerned, an answer comes from the Lemke-Howson algorithm, first introduced by Lemke and Howson ([LH]). Given a 2-player normal-form game, the Lemke-Howson algorithm moves along a directed path on the vertices of the polytope given by the inequalities describing game's strategies and payoffs, starting from the vertex $(0, \dots, 0)$ and terminating in a vertex corresponding to a Nash equilibrium; at each vertex that is not a Nash equilibrium the next vertex is calculated efficiently via a set of linear equations; for a thorough description see [vS07]. But, since a Nash equilibrium is always guaranteed to exist, this is exactly what it is required to be a problem in PPAD, with $(0, \dots, 0)$ as standard source, the Nash equilibria of the game as sinks, and the path described by the algorithm as directed graph. The algorithm has been generalised to r -player games by Rosenmüller ([R71]), with the condition that the game is nondegenerated, that is, that its polytope is nondegenerated.

On the other hand, to see that the problem of finding an ε -Nash equilibrium is a problem in PPAD we can reduce it to k -D SPERNER. This is a problem that takes as input a triangulation of the k -dimensional simplex in which every vertex of the subdivision is labelled with one of $k + 1$ colours, with the restriction that on each face of the simplex never appears the colour of the opposite vertex, and outputs a simplex in the subdivision on whose vertices all the colours are represented, called a *panchromatic* simplex. The existence of such a simplex is ensured by Sperner's lemma (see for instance [C67] and chapter 4 of [B85]). An example of this reduction can be found in [DGP2]: its main idea is that, since at a (ε) -Nash equilibrium every player "does not move away" from the mixed strategy he has chosen, an (ε) -Nash equilibrium corresponds to a fixed point of a function that measures how a player would move from the strategy where he currently is towards his best response to the other players' profile. The strategy profiles polytope can then be embedded in a k -dimensional simplex, which in turn can be triangulated with simplices of diameter less than ε . The vertices of these simplices can be coloured in a way such that each fixed point in the sense specified above corresponds to a panchromatic simplex: this is accomplished by choosing the colour of each point according to the direction in which it would "move away" towards a Nash equilibrium. Therefore, the vertices of a simplex that contains a Nash equilibrium will thus be coloured with all possible colours, since the sum of all directions in which its vertices "move" will point to the Nash equilibrium (for an illustration of the 2-dimensional case, see [G07]). We have thus reduced r -NASH to k -D SPERNER.

Once we have the reduction from r -NASH to k -D SPERNER, we can reduce the latter to END OF THE LINE using an algorithm based on the one used in the proof of Sperner's Lemma (see [C67]) to find a panchromatic simplex in the subdivision of the k -simplex. Each face of this simplex is a $(k - 1)$ -simplex subdivided in $(k - 1)$ -simplices, an odd number of which is, by induction hypothesis, panchromatic. Each one of these panchromatic $(k - 1)$ -simplices is also a face of a k -simplex in the subdivision of the k -simplex; if this is panchromatic (that is, if the colour of the vertex not on the face is the one that is not represented on the face, that is the one of the vertex opposite of the face) the algorithm stops, otherwise we "drop" the vertex on the face whose colour is the same as the one of the new vertex and we obtain

a new $(k - 1)$ -simplex, on which we repeat the procedure. Since we never cover a k -simplex twice, the algorithm is guaranteed to end, either in a panchromatic k -simplex or in a panchromatic $k - 1$ -simplex on the face we started from – the only face without the colour we were looking for from the beginning. But since by induction hypothesis the face has an odd number of panchromatic $(k - 1)$ -simplices there must be an odd number of these that taken as a starting point of the algorithm lead to a panchromatic k -simplex. Once we have given a rule to move to another $(k - 1)$ -simplex on the face once we have returned there (for instance, “consider the k -simplex obtained adding a vertex the same colour as one of the vertices delimiting the face and go on”), we have an algorithm that describes how to follow a path starting from a standard source and arriving in a sink. For a detailed description of the algorithm in the 2-dimensional case see [G07]; note, though, that as pointed out in [P94] it is not immediate to have a description of the algorithm not based on induction in the general k -dimensional simplex since there is no easy standard simplicisation even of the tetrahedron. [P94] solves the problem by proving the 3-dimensional case equivalent to finding a panchromatic cubelet in a subdivision of the unit cube, a problem that we will study in more depth in section 3.1; the generalisation to k dimensions follows similarly.

Finally, it is interesting to note that Sperner’s lemma is equivalent to Brouwer’s fixed point theorem, which ensures the existence of a fixed point of any continuous function from a k -dimensional simplex to itself (see again [B85], chapter 9); in turn, Brouwer’s fixed point theorem is the result used by Nash to prove theorem 1 in [N51].

2 Reductions among equilibrium problems

Solving an $n + 1$ -player game is always at least as hard as solving an n -player game: we can always reduce the n -player game to an $n + 1$ -player game by adding a player whose moves do not have any effect on the other player's payoffs and whose payoff is always zero. In this section, which explains the results of [GP] and [DP], we will see how the whole hierarchy of $3, \dots, n$ -NASH “collapses” onto 3-NASH: solving a 3-player game is as hard as solving any r -player game for $r > 3$. Furthermore, we will see that solving any graphical game of degree d or any normal-form r -player game ($r, d \geq 3$) is as hard as solving a graphical game of maximum degree 3 in which all players have two strategies each; a game that satisfies the latter property is called a *binary game*.

All the results of this section hold for the problems of both finding Nash and finding ε -Nash equilibria: we will first give the proof of the exact case then the proof for the approximate case. We will start by relating two results first proved in [GP]; each one of these introduces a technique that will also be used later. The first result is a reduction from d -GRAPHICAL NASH to $d^2 + 1$ -NASH; its proof introduces the technique of colouring the vertices of a graphical game then take each colour as a “super-player”. To prove this result we follow [DGP2], with a change to correct an error and another change to reunite two cases in one. The second and more complex result is a reduction from n -NASH to the problem of finding a ε -Nash equilibrium of a binary graphical game of maximum degree 3; in this proof we will introduce the technique of identifying the vertices of the graphical game with real numbers and using graphical games called *gadgets* to simulate arithmetic operations between these vertices; in this case as well we follow the more detailed proof given in [DGP2]. These two results are then combined in subsection 3 to prove the main result of [GP], a reduction from r -NASH to 4-NASH. Finally, following [DP], the gadgets are slightly modified to obtain a reduction from r -NASH to 3-NASH; this also yields a reduction from both r -NASH and d -GRAPHICAL NASH to both 3-NASH and 3-GRAPHICAL NASH for $r, d \geq 3$.

2.1 From GRAPHICAL NASH to NASH

In this first subsection given a graphical game \mathcal{GG} we construct a normal-form game \mathcal{G} such that there is a polynomially computable surjective function from the set of (ε) -Nash equilibria of \mathcal{G} to the set of (ε) -Nash equilibria of \mathcal{GG} . This is accomplished by colouring the graph of \mathcal{GG} and using each colour as a player of \mathcal{G} . As suggested in [P07], we can think the relationship of these “super-players” of the new game with the vertices of the graphical game as the one of lawyers representing their clients. To ensure that there are no “conflicts of interest”, that is to avoid that the “super-players” exploit the underlying structure of the original game to cooperate, we require that the colouring is proper not just for the graph of \mathcal{GG} but also for its moralised graph. Finally, to ensure that no vertex is privileged by its “super-player” we modify the payoffs with a matching pennies at very high stakes played by the “super-players” on the strategies “play the i -th of his vertices” (we can assume without loss of generality that each “super-player” has the same number of vertices). The theorems that we present are an original variation on the ones in [GP] and [DGP2]: whereas [GP] gives a reduction to d^2 or $d^2 - 1$ -NASH, whichever is even, and [DGP2]

gives a reduction to d^2 or $d^2 + 1$ -NASH, whichever is even, we will show that the reduction must be to $d^2 + 1$ -NASH, dropping the condition on the parity by using the 3-player matching pennies of [CD3].

Theorem 3 (see Theorem 1 of [GP], Theorem 4 of [DGP2]) *For every $d > 1$ a graphical game \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 polynomially computable surjective mapping g from the Nash equilibria of the latter to the Nash equilibria of the former.*

PROOF. Let \mathcal{GG} be a graphical game on $G = (V = (v_1, \dots, v_{n'}), E)$ with pure strategies $\{1, \dots, t\}$ for each player (add dummy strategies with a very low payoff if needed), utility functions u^v , and let d be the maximum degree of G .

Construction of the normal-form game \mathcal{G}

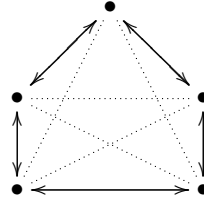
1. Rescale all utilities u^v of \mathcal{GG} so that they lie in $[0, 1]$, for instance by dividing all the utilities by $\max(u^v(a))$, the largest utility attainable in \mathcal{GG} .
2. Let $r = d^2 + 1$.
3. Let $c : V \rightarrow \{1, \dots, r\}$ be a colouring of G that is a proper colouring of the moralised graph of \mathcal{GG} as well.
4. Add isolated vertices to G until there are the same number of vertices with the same colour; let n be the new number of vertices and denote $\{v_1^i, \dots, v_{n/r}^i\}$ the set of vertices with colour i .
5. For every player p of \mathcal{G} in $\{1, \dots, r\}$ the set of pure strategies is defined as

$$S_p = \bigcup_{c(v)=p} S_v = \{(v_i, a) : c(v_i) = p, a \in S_{v_i}\}.$$

6. The utility functions of \mathcal{G} are set as follow:

- i. initially, all utilities $u^p(s)$ are zero.
- ii. (Payoffs from \mathcal{GG} .) Let $v_0 \in V$, and let $v_1, \dots, v_{d'}$ be its predecessors in the affects-graph. If $c(v_0) = p$ and for some $i \in \{0, \dots, d'\}$ the strategy s of \mathcal{G} contains a (v_i, a_i) then $u^p(s) = u^{v_0}(s')$, where s' is a strategy profile of \mathcal{GG} in which v_i plays a_i for $i \in \{0, \dots, d'\}$.
- iii. Let $M > 2\frac{n}{r}$.
- iv. (Matching pennies.) If r is odd, let $r' = r - 3$, if r is even, let $r' = r$. If any odd player $p < r'$ plays (v_i^p, a) and $p + 1$ plays (v_i^{p+1}, a') add M to $u^p(s)$ and subtract M from $u^{p+1}(s)$. If r is odd, then, if player $r - 2$ plays (v_i^{r-2}, a) and player $r - 1$ or player r plays (v_i^p, a') (with $p \in \{r - 1, r\}$) add M to $u^{r-2}(s)$; if player $r - 1$ plays (v_i^{r-1}, a) and player $r - 2$ or player r plays (v_i^p, a') (with $p \in \{r - 2, r\}$) subtract M to $u^{r-1}(s)$; finally, if player r plays (v_i^r, a) and player $r - 1$ plays (v_i^{r-1}, a') add M to $u^r(s)$.

We now show that a colouring as the one required in step 3 is possible. Let v be a vertex of the graph of \mathcal{GG} such that $\deg(v) = d$; there are at most d vertices w such that (v, w) or (w, v) is an arc of the affects-graph, and there are at most $d(d-1)$ vertices v' such that there exist a vertex w' such that (v, w') and (v', w') are arcs of the affects-graph (count at most d vertices w' at distance 1 and at most $d-1$ vertices v' that are at distance 1 from the w' s but are not v). Therefore, the degree of v in the moralised graph is at most $d + d(d-1) = d^2$, and since v was of maximum degree in the graph this implies that the maximum degree of the moralised graph is at most d^2 . Hence, the chromatic number of the moralised graph is less or equal than $d^2 + 1$, since we can colour it using a greedy algorithm that follows a breadth-first search: if v_1, \dots, v_k are the vertices of the graph we start by giving colour 1 to v_1 , then we consider all its neighbours v_{l_i} and starting from the smallest l_i we colour them with all the remaining colours using every time the smallest colour available; then we colour with the neighbours of $\min_{l_i} v_{l_i}$, always using the smallest possible colour first, and so on (see for instance Diestel, [D05]). Notice that an equality is possible, a fact that shows that the choice of [GP] and [DGP2] of a colouring in the range $\{1, \dots, d^2\}$ is wrong. We have an equality, for instance, when the affects-graph is a 5-cycle going both clockwise and anticlockwise: the degree of the undirected graph is 2, but the moralised graph is the complete graph on 5 vertices, whose chromatic number is $5 > 2^2$ (this example is taken from [AM]). In the following figure the arcs of the affects-graph are represented in solid colour, and the edges that are in the moralised graph only are represented as dotted lines:



Polynomial size of \mathcal{G}

The size of \mathcal{GG} depends on its n' vertices, their t strategies each that are played facing at most d other vertices, and their utility functions u^p . The size of \mathcal{GG} is thus $\Theta(n' \cdot t^{d+1} \cdot q)$, where q is the size, in the logarithmic cost model, of the input quantities $u^p(s)$ – the logarithmic cost of the values in the payoff matrices associated to every vertex in \mathcal{GG} .

In \mathcal{G} there are $r = d^2 + 1$ players, having $\frac{tn}{r}$ strategies each (the t strategies of each of the $\frac{n}{r}$ vertices in its colour class; recall that by construction $n \leq rn'$): hence, there are $r(\frac{tn}{r})^r \leq (d^2 + 1)(tn')^{d^2+1}$ payoff entries in \mathcal{G} . Moreover, each payoff entry is of polynomial size since M is of polynomial size in $\frac{n}{r}$ and each payoff entry is the sum of a payoff entry of \mathcal{GG} and 0 or M .

Hence, we have that \mathcal{G} is polynomial in the size of \mathcal{GG} .

Mapping of Nash equilibria

Let \mathcal{GG}' be the graphical game resulting from the rescaling of the utilities of \mathcal{GG} in step 1; there is clearly a one-on-one correspondence of the equilibria of \mathcal{GG}' and \mathcal{GG} . We will show that there is a mapping g that associates every Nash equilibrium of \mathcal{G} to a Nash equilibrium of \mathcal{GG}' : we claim that given a Nash equilibrium $\{x_{(v,a)}^p\}$ of \mathcal{G} we can recover a Nash equilibrium of \mathcal{GG}' by setting

$$x_a^v = \frac{x_{(v,a)}^{c(v)}}{\sum_{j \in S_v} x_{(v,j)}^{c(v)}}, \text{ for every vertex } v \in V \text{ and every strategy } a \in S_v. \quad (1)$$

Clearly g is computable in polynomial time; we now see that it preserves Nash equilibria.

We first see that the “matching pennies” game does not influence how each player of \mathcal{G} distributes the probability among his vertices. For $v \in V$ and $c(v) = p$ let “ p plays v ” denote the event that p plays (v, a) for some $a \in S_v$. Let $\lambda = (\frac{n}{r})^{-1}$: note that the “fair share” of probability for each vertex of a player is λ .

Lemma 4 (Lemma 2 of [DGP2]) *For every $v \in V$ we have that $\mathbf{p}[c(v) \text{ plays } v] \in [\lambda - \frac{1}{M}, \lambda + \frac{1}{M}]$.*

PROOF. Let $p = c(v)$. Suppose by contradiction that in a Nash equilibrium of \mathcal{G} for some i we have that $\mathbf{p}[p \text{ plays } v_i^p] < \lambda - \frac{1}{M}$. The part of “fair share” not used in playing i is thus distributed among the other vertices, hence there must be a j such that $\mathbf{p}[p \text{ plays } v_j^p] > \lambda + \frac{1}{M}\lambda$.

Suppose that p is an evader and let $p-1$ be its pursuer; if r is odd consider r as the pursuer of $r-1$. Then $p-1$ will get at most $M(\lambda - \frac{1}{M}) + 1 = \lambda M$ for playing v_i^{p-1} , with the additive 1 coming from the payoffs of \mathcal{GG}' , and at least $M(\lambda - \frac{1}{M}\lambda) = \lambda M + \lambda$ for playing v_j^{p-1} . Hence, in a Nash equilibrium $\mathbf{p}[p-1 \text{ plays } v_i^{p-1}] = 0$, which implies that there exists k such that $\mathbf{p}[p-1 \text{ plays } v_k^{p-1}] > \lambda$. Then p will have utility at least 0 for playing v_i^p and utility at most $-\lambda M + 1$ for playing v_k^p . Since by construction of M we have that $0 > -\lambda M + 1$, in a Nash equilibrium $\mathbf{p}[p \text{ plays } v_k^p] = 0$, which implies that there is l such that $\mathbf{p}[p \text{ plays } v_l^p] > \lambda$. The payoff of $p-1$ for playing v_k^{p-1} is at most 1, whereas his payoff for playing v_l^{p-1} is at least λM . Thus in a Nash equilibrium $\mathbf{p}[p-1 \text{ plays } v_k^{p-1}] = 0$, a contradiction.

Now suppose that $p < r$ is a pursuer, and let $p+1$ be its evader (when $p = r$ consider $r-1$ as the evader of both $r-2$ and r). Then $p+1$ will get at least $-M(\lambda + \frac{1}{M}) = -\lambda M + 1$ for playing v_i^{p+1} , and at most $-M(\lambda + \frac{1}{M}\lambda) + 1 = -\lambda M + \lambda + 1$ for playing v_j^{p+1} . Since $-\lambda M + 1 > -\lambda M - \lambda + 1$, in a Nash equilibrium $\mathbf{p}[p+1 \text{ plays } v_j^{p+1}] = 0$; therefore, there exists k such that $\mathbf{p}[p+1 \text{ plays } v_k^{p+1}] > \lambda$. The payoff of p for playing v_k^p is thus at least λM , and his payoff for playing v_j^p is at most 1. Since $\lambda M > 1$ we have that p will not play v_j^p , a contradiction.

Finally, note that if r is odd $r-2$ doesn't just play one of the vertices played by r leaving one vertex v_i^{r-2} with probability less than $\lambda - \frac{1}{M}$, because then $r-1$ would play v_i^{r-1} leaving another vertex with probability less than $\lambda - \frac{1}{M}$, contradicting what we have proved.

A similar argument proves that $\mathbf{p}[p \text{ plays } v_i^p] < \lambda + \frac{1}{M}$ for all i . \square

Lemma 4 implies that the division of $\mathbf{p}[c(v) \text{ plays } v]$ into $\mathbf{p}[c(v) \text{ plays } (v, a)]$ for $a \in S_v$ is driven entirely by the payoffs from \mathcal{GG}' . Moreover, for every v there is a probability $p(v) \geq (\lambda - \frac{1}{M})^d > 0$ that

the predecessors of v in the affects-graph of \mathcal{GG}' are chosen by the other players of \mathcal{G} . The payoff to $c(v)$ from choosing (v, a) for some $a \in S_v$ is thus $p(v)$ times the expected payoff of v in \mathcal{GG}' if all the vertices play as specified by (1).

More formally, let $p = c(v)$ and $v = v_i^p$; let $N(v)$ denote the set of predecessors of v in the affects-graph of \mathcal{GG}' (recall that contrarily to [GP] and [DGP2] we do not include v in this set, to easen the notation in the following passages). In a Nash equilibrium of \mathcal{G} by definition we must have

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

Now let $p \in \{1, \dots, r\}$ if r is even, $p \in \{1, \dots, r\} \setminus \{r-2\}$ if r is odd. If p is a pursuer let $M' = M$ and p' be the evader, if p is an evader let $M' = -M$ and p' be the pursuer: we have that for any a

$$\mathbb{E}[\text{payoff to } p \text{ for playing } (v, a)] = M' \mathbf{p}[p' \text{ plays } v_i^{p'}] + \sum_{s \in S_{N(v)}} u^v(a, s) \prod_{u \in N(v)} x_{(u, s_u)}^{c(u)}; \quad (2)$$

on the other hand, if r is odd and $p = r-2$ we have that

$$\mathbb{E}[\text{payoff to } p \text{ for playing } (v, a)] = M(\mathbf{p}[r-1 \text{ plays } v_i^{r-1}] + \mathbf{p}[r \text{ plays } v_i^r]) + \sum_{s \in S_{N(v)}} u^v(a, s) \prod_{u \in N(v)} x_{(u, s_u)}^{c(u)}. \quad (3)$$

In both cases, we have that in a Nash equilibrium of \mathcal{G}

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

dividing by $\prod_{u \in N(v)} \sum_{j \in S_u} x_{(u, j)}^{c(u)} = \prod_{u \in N(v)} \mathbf{p}[c(u) \text{ plays } u] = p(v)$ and invoking (1) yields

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

that is, the condition for a Nash equilibrium of \mathcal{GG}' .

The mapping is surjective

We will show that for every Nash equilibrium $\{x_a^v\}_{v,a}$ of \mathcal{GG} there exists a Nash equilibrium $\{x_{(v,a)}^p\}_{p,v,a}$ of \mathcal{G} such that (1) holds. Let \mathcal{G}' be the game obtained from \mathcal{G} by merging the strategies $\{(v, a)\}_a$ of player $p = c(v)$ into one strategy s_v^p , for every player p , so that the strategy set of player p is $\{s_v^p\}_{c(v)=p}$. Suppose that in the Nash equilibrium of \mathcal{GG}' described by $\{x_a^v\}_{v,a}$ every vertex v receives an expected payoff of u_v ; the payoffs of \mathcal{G}' are as follows:

- i. initially all the utilities are equal to 0.
- ii. If $v_1, \dots, v_{d'} \in V$ are the predecessors of $v_0 \in V$ in the affects-graph, and we have that s contains $s_{v_i}^{c(v_i)}$ for $i = 0, \dots, v_{d'}$, then add u_{v_0} to $u^{c(v_0)}(s)$.

- iii. Let $r' = r$ if r is even, $r' = r - 3$ if r is odd. For $p < r'$ odd if p plays strategy s_i^p and $p + 1$ plays strategy s_i^{p+1} then add M to $u^p(s)$ and subtract M to $u^{p+1}(s)$. Furthermore, if r is odd, if player $r - 2$ plays strategy s_i^{r-2} and player $r - 1$ plays strategy s_i^{r-1} or player r plays strategy s_i^r add M to $u^{r-2}(s)$; if player $r - 1$ plays strategy s_i^{r-1} and player $r - 2$ plays strategy s_i^{r-2} or player r plays strategy s_i^r subtract M from $u^{r-1}(s)$; if player r plays strategy s_i^r and player $r - 1$ plays strategy s_i^{r-1} add M to $u^r(s)$.

Nash's theorem ensures that \mathcal{G}' has a Nash equilibrium $\{y_{s_i^p}^p\}_{p,v}$, and $y_{s_i^p}^p \cdot x_a^v$ "redistributes" the probability among the strategies (v, a) returning a Nash equilibrium $\{x_{(v,a)}^p\}_{p,v,a}$ of game \mathcal{G} . \square

The theorem can be extended to the problem of finding ε -Nash equilibria.

Theorem 5 (see Theorem 4 of [GP], Theorem 8 of [DGP2]) *For every $d > 1$ there is a polynomial-time reduction from d -GRAPHICAL NASH to $d^2 + 1$ -NASH.*

PROOF. Let \mathcal{GG} be a graphical game of maximum degree d , and let \mathcal{GG}' be the game obtained by rescaling the payoffs of \mathcal{GG} so that they lie in $[0, 1]$, as in the first step of the algorithm of theorem 3. Let $\varepsilon < 1$. We will see that it is possible to construct in time polynomial in $|\mathcal{GG}| + \log(\frac{1}{\varepsilon})$ a normal form game \mathcal{G} ad an accuracy parameter ε' such that given a ε' -Nash equilibrium of \mathcal{G} one can recover in polynomial time a ε -Nash equilibrium of \mathcal{GG}' , and thus a ε -Nash equilibrium of \mathcal{GG} , since every ε -Nash equilibrium of \mathcal{GG}' is trivially a $\varepsilon \cdot \max(u^v(s))$ -Nash equilibrium of \mathcal{GG} .

Construct \mathcal{G} using the algorithm of theorem 3 and choose $\varepsilon' \leq \varepsilon(\lambda - \frac{1}{M})^d$, where $\lambda = \frac{r}{n}$. As in theorem 3 the construction of \mathcal{G} takes time $|\mathcal{GG}|$; the construction of ε' takes, in the logarithmic cost model, time proportional to the logarithm of the number of bits necessary to represent it in binary, which, since ε and ε' are in $[0, 1]$, is polynomial in $\log(\frac{1}{\varepsilon})$.

We now claim that we can recover a ε -Nash equilibrium of \mathcal{GG}' using (1). As in the proof of Lemma 4 it can be shown that in any ε' -Nash equilibrium of \mathcal{G} $\mathbf{p}[c(v) \text{ plays } v] \in [\lambda - \frac{1}{M}, \lambda + \frac{1}{M}]$. Now, in a ε' -Nash equilibrium of \mathcal{G} we must have

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

but by (2) and (3) this implies that in a ε' -Nash equilibrium of \mathcal{G}

$$\sum_{s \in S_{N(v)}} u^v(a, s) \prod_{u \in N(v)} x_{(u, s_u)}^{c(u)} > \sum_{s \in S_{N(v)}} u^v(a', s) \prod_{u \in N(v)} x_{(u, s_u)}^{c(u)} + \varepsilon' \Rightarrow x_{(v,a')}^p = 0;$$

dividing by $T = \prod_{u \in N(v)} \sum_{j \in S_u} x_{(u,j)}^{c(u)} = \prod_{u \in N(v)} \mathbf{p}[c(u) \text{ plays } u] \geq (\lambda - \frac{1}{M})^d$ and invoking (1) yields

$$\sum_{s \in S_{N(v)}} u^v(a, s) \prod_{u \in N(v)} x_{s_u}^u > \sum_{s \in S_{N(v)}} u^v(a', s) \prod_{u \in N(v)} x_{s_u}^u + \frac{\varepsilon'}{T} \Rightarrow x_{a'}^v = 0;$$

but since $\frac{\varepsilon'}{T} \leq \varepsilon$ this is the condition for a Nash equilibrium of \mathcal{GG}' . \square

In the proofs of these results is implicit the following

Corollary 6 (Lemma 2 of [DP]) *Let \mathcal{GG} be a graphical game on graph $G = (\{v_1, \dots, v_n\}, E)$ and suppose that its moralised graph is k -colourable for an integer $k > 1$. Then a k -player normal-form game \mathcal{G} can be constructed in polynomial time such that from any (ε) -Nash equilibrium of \mathcal{G} we can recover a (ε) -Nash equilibrium of \mathcal{GG} with a polynomially computable surjective mapping.*

PROOF. For Nash equilibria, apply the algorithm of theorem 3 with $r = k$; for ε -Nash equilibria apply the algorithm of theorem 5 also with $r = k$. \square

2.2 From NASH to GRAPHICAL NASH

We have proved that given a graphical game we can construct a normal-form game and a function mapping the equilibria of the latter to the equilibria of the former; now given a normal-form game \mathcal{G} we will construct a graphical game \mathcal{GG} such that there is a polynomially computable function that maps the equilibria of \mathcal{GG} to the equilibria of \mathcal{G} . A trivial reduction from n -NASH to n -GRAPHICAL NASH would be to “translate” the n -player game to a graphical game played on an n -clique, but we will prove a less immediate result: the graphical game that we will construct will have a bipartite graph of maximum degree 3 and will be binary. The latter condition yields an important consequence: if every vertex in a game has two strategies, 0 and 1, we can “identify” the vertex with the real number in $[0, 1]$ corresponding to its probability of playing 1 at the equilibrium, that we will denote $\mathbf{p}[v]$. The idea is now to use a graphical game to reproduce the calculation of the (ε) -Nash equilibria of \mathcal{G} . This is done by using vertices that simulate components of mixed strategy profiles; we will then introduce in \mathcal{GG} some vertices that at a Nash equilibrium will correspond to components of a (ε) -Nash equilibrium of \mathcal{G} , so that if $v(x_j^p)$ is the vertex corresponding to the component x_j^p of \mathcal{G} at an equilibrium of \mathcal{GG} the value of $\mathbf{p}[v(x_j^p)]$ will be the value of x_j^p at a Nash equilibrium of \mathcal{G} ; finally we will then connect these vertices via *gadgets*, subgraphs that simulate the arithmetical operations necessary to calculate the (ε) -Nash equilibrium of \mathcal{G} .

Before stating the theorems we will introduce some of these gadgets, that will be used in later sections as well. The first of these gadgets multiplies a vertex by a non-negative real number α ; by setting $\alpha = 1$ we can make a copy of the vertex; we call this game $\mathcal{G}_{\times\alpha}$ and for $\alpha = 1$ we call it $\mathcal{G}_=$. We use the notation $x = y \pm \varepsilon$ for $x \in [y - \varepsilon, y + \varepsilon]$.

Proposition 7 (Proposition 1 of [GP], Lemma 1 of [DGP], Proposition 1 of [DGP2]) *Let α be a non-negative real number. Let v_1, v_2, w be vertices in a graphical game and suppose that the payoffs for v_2 are*

		w	
		0	1
v_2	0	0	1
	1	1	0

and the payoffs for w are

		v_2		0	1
v_1	0			0	0
	1			α	α

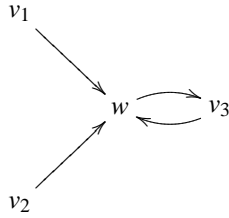
		v_2		0	1
v_1	0			0	1
	1			0	1

Then in any ε -Nash equilibrium of the game $\mathbf{p}[v_2] = \min(\alpha\mathbf{p}[v_1], 1) \pm \varepsilon$

PROOF. If w plays 1 its expected payoff is $\mathbf{p}[v_2]$, if it plays 0 its expected payoff is $\alpha\mathbf{p}[v_1]$. Therefore, if $\mathbf{p}[v_2] > \alpha\mathbf{p}[v_1] + \varepsilon$ in a ε -Nash equilibria of the game w would play 1: but then we would have $\mathbf{p}[v_2] = 0$, a contradiction. Thus $\mathbf{p}[v_2] \leq \alpha\mathbf{p}[v_1] + \varepsilon$. Similarly, if $\mathbf{p}[v_2] < \alpha\mathbf{p}[v_1] - \varepsilon$ then w would play 0, which implies $\mathbf{p}[v_2] = 1$, another contradiction. Thus $\mathbf{p}[v_2] \geq \alpha\mathbf{p}[v_1] - \varepsilon$. \square

The second gadget, a generalisation of the first, allow us to sum and multiply vertices.

Proposition 8 (Proposition 2 of [GP], Lemma 1 of [DGP], Proposition 2 of [DGP2]) Let α, β, γ be non-negative real numbers. Let v_1, v_2, v_3, w be vertices in a graphical game with affects-graph



and suppose the payoffs are as follows:
the payoffs for v_3 are

v_3	w	0	1
	0	0	1
	1	1	0

the payoffs for w playing 0 are

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

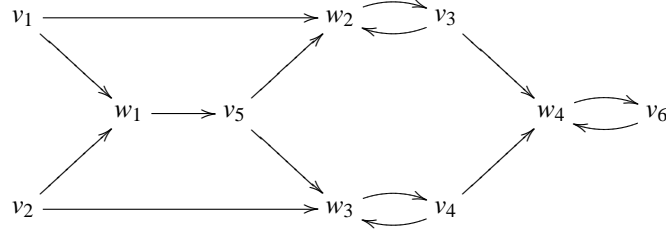
whereas if w plays 1 it gets 0 if v_3 plays 0 and it gets 1 if v_3 plays 1.

Then in any ε -Nash equilibrium of the game $\mathbf{p}[v_3] = \min(\alpha\mathbf{p}[v_1] + \beta\mathbf{p}[v_2] + \gamma\mathbf{p}[v_1]\mathbf{p}[v_2], 1) \pm \varepsilon$

PROOF. If w plays 1 its expected payoff is $\mathbf{p}[v_3]$, if w plays 0 its expected payoff is $\alpha\mathbf{p}[v_1] + \beta\mathbf{p}[v_2] + \gamma\mathbf{p}[v_1]\mathbf{p}[v_2]$. Therefore, if $\mathbf{p}[v_3] > \alpha\mathbf{p}[v_1] + \beta\mathbf{p}[v_2] + \gamma\mathbf{p}[v_1]\mathbf{p}[v_2] + \varepsilon$ in a ε -Nash equilibrium of the game $\mathbf{p}[w] = 1$. But the v_3 payoffs imply that $\mathbf{p}[v_3] = 0$, a contradiction; hence $\mathbf{p}[v_3] \leq \alpha\mathbf{p}[v_1] + \beta\mathbf{p}[v_2] + \gamma\mathbf{p}[v_1]\mathbf{p}[v_2] + \varepsilon$. Similarly, if $\mathbf{p}[v_3] < \min(\alpha\mathbf{p}[v_1] + \beta\mathbf{p}[v_2] + \gamma\mathbf{p}[v_1]\mathbf{p}[v_2], 1) - \varepsilon < \alpha\mathbf{p}[v_1] + \beta\mathbf{p}[v_2] + \gamma\mathbf{p}[v_1]\mathbf{p}[v_2] - \varepsilon$ we have that $\mathbf{p}[w] = 0$ and $\mathbf{p}[v_3] = 1 > 1 - \varepsilon$, again a contradiction. Hence, $\mathbf{p}[v_3] \geq \min(\alpha\mathbf{p}[v_1] + \beta\mathbf{p}[v_2] + \gamma\mathbf{p}[v_1]\mathbf{p}[v_2], 1) - \varepsilon$. \square

For $\alpha = \beta = 1, \gamma = 0$ we call the previous gadget \mathcal{G}_+ , for $\alpha = \beta = 0, \gamma = 1$ we call it \mathcal{G}_* ; for $\alpha = 1, \beta = -1, \gamma = 0$ a proof analogous to the one of the last proposition shows that $\mathbf{p}[v_3] = \max(0, \mathbf{p}[v_1] - \mathbf{p}[v_2]) \pm \varepsilon$, and we call the gadget \mathcal{G}_- . Now we give a gadget \mathcal{G}_{\max} to calculate the maximum.

Proposition 9 (Proposition 3 of [GP], Proposition 3 of [DGP2]) 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 with affects-graph



and suppose that the payoffs are as follows.

Payoffs for w_1 :

		v_2	
		0	1
v_1	0	0	0
	1	1	1

		v_2	
		0	1
v_1	0	0	1
	1	0	1

Payoffs for v_5 :

		w_1	
		0	1
v_5	0	1	0
	1	0	1

Payoffs to w_2 and v_3 are chosen with proposition 8 such that in any ε -Nash equilibrium of the game $\mathbf{p}[v_3] = \mathbf{p}[v_1](1 - \mathbf{p}[v_5]) \pm \varepsilon$;

payoffs to w_3 and v_4 are chosen with proposition 8 such that in any ε -Nash equilibrium of the game $\mathbf{p}[v_4] = \mathbf{p}[v_2]\mathbf{p}[v_5] \pm \varepsilon$;

payoffs to w_4 and v_6 are chosen with proposition 8 such that in any ε -Nash equilibrium of the game $\mathbf{p}[v_6] = \min(\mathbf{p}[v_3] + \mathbf{p}[v_4], 1) \pm \varepsilon$.

Then in any ε -Nash equilibrium of the game $\mathbf{p}[v_6] = \max(\mathbf{p}[v_1], \mathbf{p}[v_2]) \pm 4\varepsilon$

PROOF. If $\mathbf{p}[v_1] < \mathbf{p}[v_2] - \varepsilon$ then $\mathbf{p}[w_1] = 1$, and thus $\mathbf{p}[v_5] = 1$. Hence $\mathbf{p}[v_3] = \pm\varepsilon$ and $\mathbf{p}[v_4] = \mathbf{p}[v_2] \pm \varepsilon$, and so $\mathbf{p}[v_6] = \mathbf{p}[v_2] \pm 3\varepsilon$. A similar argument shows that $\mathbf{p}[v_1] > \mathbf{p}[v_2] + \varepsilon$ implies that $\mathbf{p}[v_6] = \mathbf{p}[v_1] \pm 3\varepsilon$.

If $|\mathbf{p}[v_1] - \mathbf{p}[v_2]| \leq \varepsilon$, we have that $\mathbf{p}[w_1]$ and thus $\mathbf{p}[v_5]$ may take any value. Assuming that $|\mathbf{p}[v_1] - \mathbf{p}[v_2]| \leq \varepsilon$ and $\mathbf{p}[v_1] \geq \mathbf{p}[v_2]$, we have $\mathbf{p}[v_3] + \mathbf{p}[v_4] = \mathbf{p}[v_1](1 - \mathbf{p}[v_5]) + \mathbf{p}[v_2]\mathbf{p}[v_5] \pm 2\varepsilon = \mathbf{p}[v_1](1 - \mathbf{p}[v_5]) + \mathbf{p}[v_1]\mathbf{p}[v_5] \pm 3\varepsilon = \mathbf{p}[v_1] \pm 3\varepsilon$, so that $\mathbf{p}[v_6] = \mathbf{p}[v_1] \pm 4\varepsilon$. \square

In the following proposition we present a gadget that assigns a fixed value α to a player; we call it \mathcal{G}_α .

Proposition 10 (Lemma 1 of [DGP], Proposition 4 of [DGP2]) *Let v, w be players in a graphical game and let their payoffs be as follows:*

		w	
		0	1
v	0	α 0	0 1
	1	α 1	1 0

Then, in every ε -Nash equilibrium of the game $\mathbf{p}[v] = \min(\alpha, 1) \pm \varepsilon$

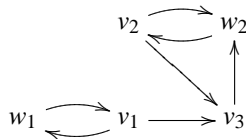
PROOF. The payoff of w is α for playing 0 and $\mathbf{p}[v]$ for playing 1. Therefore, if $\mathbf{p}[v] > \alpha + \varepsilon$ then w plays 1, which implies that v plays 0. But then $\alpha < -\varepsilon$, a contradiction. On the other hand, if $\mathbf{p}[v] < \min(\alpha - \varepsilon, 1) < \alpha - \varepsilon$ we have that w plays 0 and so v plays 1. But this implies that $1 < \min(\alpha - \varepsilon, 1)$, a contradiction. Hence, $\mathbf{p}[v] = \min(\alpha, 1) \pm \varepsilon$. \square

Finally, a gadget that simulates the relation “ $<$ ”; we call it $\mathcal{G}_<$.

Proposition 11 (Lemma 1 of [DGP], Lemma 11 of [DGP2]) *There exists a binary graphical game $\mathcal{G}_<$ with vertices v_1, v_2, w such that the payoffs of v_1 and v_2 do not depend on the choices of w , and in every ε -Nash equilibrium of the game $\mathbf{p}[w] = 1$ if $\mathbf{p}[v_1] < \mathbf{p}[v_2] - \varepsilon$ and $\mathbf{p}[w] = 0$ if $\mathbf{p}[v_1] > \mathbf{p}[v_2] + \varepsilon$.*

PROOF. The game is defined as follows: w receives a payoff of 1 if w plays 0 and v_1 plays 1, or if w plays 1 and v_2 plays 1. This is equivalent to say that w receives a payoff of $\mathbf{p}[v_1]$ if it plays 0 and a payoff of $\mathbf{p}[v_2]$ if it plays 1. Hence, in a ε -Nash equilibrium $\mathbf{p}[w] = 0$ if $\mathbf{p}[v_1] > \mathbf{p}[v_2] + \varepsilon$ and $\mathbf{p}[w] = 1$ if $\mathbf{p}[v_1] < \mathbf{p}[v_2] - \varepsilon$. \square

The gadget $\mathcal{G}_<$ is called the *brittle comparator*, since $\mathbf{p}[w]$ is arbitrary if $|\mathbf{p}[v_1] - \mathbf{p}[v_2]| \leq \varepsilon$. A robust comparator that returns a deterministic value (i.e., zero) if $\mathbf{p}[v_1] = \mathbf{p}[v_2]$ is impossible, because it could be used to produce a graphical game without Nash equilibrium. A game such as this is given in [DGP2]: it has affects-graph



where w_1, v_1 constitute a game \mathcal{G}_1 so that at the equilibrium $\mathbf{p}[v_1] = 1$, the vertices v_3, w_2, v_2 constitute a $\mathcal{G}_=$ game, and finally v_1, v_2, v_3 constitute a $\mathcal{G}_<$ game such that $\mathbf{p}[v_3] = 1$ if $\mathbf{p}[v_2] < \mathbf{p}[v_1]$ and $\mathbf{p}[v_3] = 0$ if $\mathbf{p}[v_2] \geq \mathbf{p}[v_1]$. If $\mathbf{p}[v_2] = 1$, then $\mathbf{p}[v_3] = 0$ since $\mathbf{p}[v_2] = 1 = \mathbf{p}[v_1]$, but then $\mathcal{G}_=$ implies that $\mathbf{p}[v_2] = \mathbf{p}[v_3] = 0$, a contradiction. On the other hand, if $\mathbf{p}[v_2] = 0$ then $\mathbf{p}[v_3] = 1$ since $\mathbf{p}[v_2] < \mathbf{p}[v_1] = 1$. But again $\mathcal{G}_=$ will imply that $\mathbf{p}[v_2] = \mathbf{p}[v_3] = 1$, a contradiction.

We conclude with a useful definition related to the idea of gadgets. Let v, v_1, \dots, v_k be players in a graphical game \mathcal{G}_f , where f is a function on k arguments which has range $[0, 1]$, such that in any Nash equilibrium $\mathbf{p}[v] = f(\mathbf{p}[v_1], \dots, \mathbf{p}[v_k])$. We say that \mathcal{G}_f has *error amplification at most c* if in every ε -Nash equilibrium it holds that $\mathbf{p}[v] = f(\mathbf{p}[v_1], \dots, \mathbf{p}[v_k]) \pm c\varepsilon$.

We can now state the first theorem of this subsection:

Theorem 12 (Theorem 2 of [GP], Theorem 5 of [DGP2]) *For every $r > 1$, a normal form game with r players can be mapped in polynomial time to a binary graphical game of maximum degree 3 so that there is a polynomially computable surjective mapping from the Nash equilibria of the latter to the Nash equilibria of the former.*

PROOF. Let \mathcal{G} be a normal-form game with r players with n strategies each. We will construct a graphical game \mathcal{GG} with bipartite graph $G = (V \sqcup W, E)$ of maximum degree 3 such that every Nash equilibrium in \mathcal{G} corresponds to a Nash equilibrium in \mathcal{GG} . Let $\{0, 1\}$ be the set of strategies of every vertex v of G ; as mentioned before, $\mathbf{p}[v]$ will denote the probability that v plays 1.

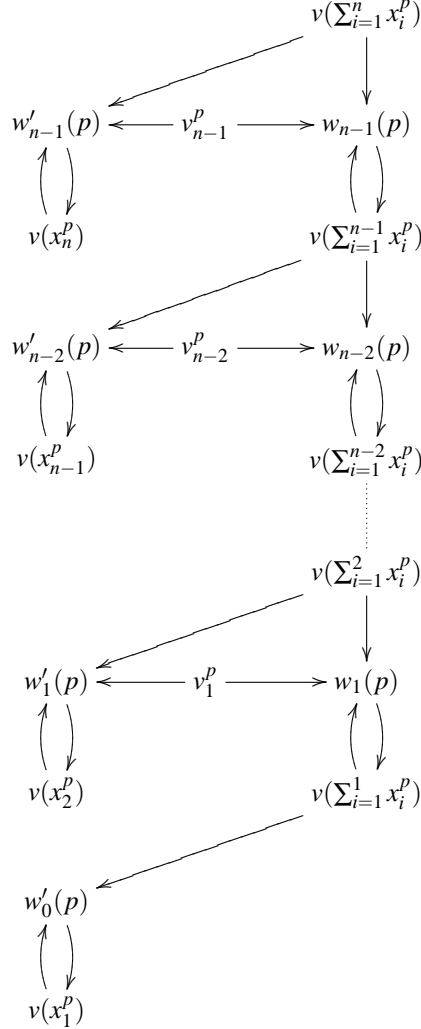
We already have the gadgets of propositions 7 to 11; we now use them to describe another building block of \mathcal{GG} , a gadget that controls that the $\{\mathbf{p}[v(x_j^p)]\}_{j \in \{1, \dots, n\}}$ s behave like a probability distribution. This is accomplished via a tree with leaves $v(x_j^p)$ that are guaranteed to sum up to 1, and with internal nodes v_j^p which control the distribution of the unit of probability among the vertices $v(x_j^p)$.

Lemma 13 (Proposition 4 of [GP], Proposition 5 of [DGP2]) *Consider a graphical game that contains:*

- *for every strategy $j \in \{1, \dots, n\}$ a vertex $v(x_j^p)$;*
- *for every strategy $j \in \{1, \dots, n-1\}$ a vertex v_j^p ;*
- *for every strategy $j \in \{1, \dots, n\}$ a vertex $v(\sum_{i=1}^j x_i^p)$;*
- *for every strategy $j \in \{1, \dots, n-1\}$ a vertex $w_j(p)$ used to ensure $\mathbf{p}[v(\sum_{i=1}^j x_i^p)] = \mathbf{p}[v(\sum_{i=1}^{j+1} x_i^p)](1 - \mathbf{p}[v_j^p])$ as in proposition 8;*
- *for every strategy $j \in \{1, \dots, n-1\}$ a vertex $w'_j(p)$ used to ensure $\mathbf{p}[v(x_{j+1}^p)] = \mathbf{p}[v(\sum_{i=1}^{j+1} x_i^p)]\mathbf{p}[v_j^p]$ as in proposition 8;*
- *a vertex $w'_0(p)$ used to ensure $\mathbf{p}[v(x_1^p)] = \mathbf{p}[v(\sum_{i=1}^1 x_i^p)]$ as in proposition 7;*
- *a vertex $v(\sum_{i=1}^n x_i^p)$ that has payoff 1 when it plays 1, zero otherwise.*

Then $\sum_{i=1}^n \mathbf{p}[v(x_i^p)] = 1$ and $\mathbf{p}[v(\sum_{i=1}^j x_i^p)] = \sum_{i=1}^j \mathbf{p}[v(x_i^p)]$. Moreover, the graph is bipartite and of maximum degree 3.

PROOF. The game's affects-graph is



The bipartition is given by adding the vertices $w_j(p)$ and $w'_j(p)$ to W and the other vertices to V ; looking at the affects-graph of the game can it is immediate to see that all the edges are between elements of V and elements of W and that in the graph of the game all nodes have degree less than or equal to 3.

In a Nash equilibrium the vertex $v(\sum_{i=1}^n x_i^p)$ must play 1; then the vertices v_j^p distribute this probability between $\mathbf{p}[v(\sum_{i=1}^j x_i^p)]$ and $\mathbf{p}[v(x_{j+1}^p)]$, so that $(v(x_1^p), \dots, v(x_n^p))$ is a probability distribution. \square

Now a proposition that we will use to deal with the behaviour of the $\mathbf{p}[v(x_j^p)]$ s in the calculation of the payoff functions and their maximisation. For $s \in S_{-p}$ let $x_s = \prod_{q \neq p} x_{s_q}^q$ and for $j \in S_p$ let $U_j^p = \sum_{s \in S_{-p}} u^p(j, s) x_s$ (the utility to p of playing j in the context of a given mixed profile) and let $U_{\leq j}^p = \max(U_j^p, U_{\leq j-1}^p)$, with $U_{\leq 1}^p = U_1^p$ (note that neither [GP] nor [DGP2] give an explicit definition of $U_{\leq 1}^p$).

Lemma 14 (Lemma 1 of [GP], Lemma 3 of [DGP2]) Suppose that all the utilities u^p lie in the range $[0, 1]$.

Then 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 $p \in \{1, \dots, r\}$ and $j \in \{1, \dots, n\}$, such that in any Nash equilibrium

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

and

$$\mathbf{p}[v(U_{\leq j}^p)] = \max_{i \leq j} \left(\sum_{s \in S_{-p}} u^p(i, s) \prod_{q \neq p} \mathbf{p}[v(x_{s_q}^q)] \right)$$

PROOF. Since the expressions for $\mathbf{p}[v(U_j^p)]$ and $\mathbf{p}[v(U_{\leq j}^p)]$ are constructed from arithmetic subexpressions that can be simulated by the gadgets we have seen in the propositions 7 to 9, there is only to check that the vertices used are $O(rn^r)$.

We enumerate the elements of S_{-p} as $\{S_{-p}(1), \dots, S_{-p}(n^{r-1})\}$, since there are $r-1$ players other than p with n strategies each. We have that

$$\sum_{s \in S_{-p}} u^p(j, s) = \sum_{l=1}^{n^{r-1}} u^p(j, S_{-p}(l)) x_{S_{-p}(l)}$$

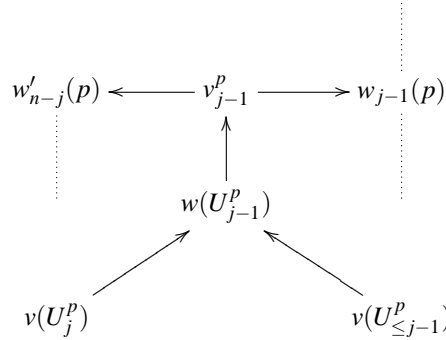
For each partial sum $\sum_{l=1}^z u^p(j, S_{-p}(l)) x_{S_{-p}(l)}$ we include a vertex $v(\sum_{l=1}^z u^p(j, S_{-p}(l)) x_{S_{-p}(l)})$: there are n^{r-1} of these for each $j \in \{1, \dots, n\}$. For each partial product $u^p(j, s) \prod_{q \neq p, q=1}^z x_{s_q}^q$ we include a vertex $v(u^p(j, s) \prod_{q \neq p, q=1}^z x_{s_q}^q)$: there are $r-1$ of these for each $j \in \{1, \dots, n\}$, $s \in S_{-p}$ ([DGP2] counts r partial products, but since $q \neq p$ we have r factors and $r-1$ products). Then we have $2n-1$ terms of comparisons to define the $U_{\leq j}^p$ ([DGP2] counts $2n$ terms of comparisons, but with our definition of $U_{\leq 1}^p$ we need a comparison to define $U_{\leq j}^p$ for $j \geq 2$ only; the remaining term is the $U_{\leq 1}^p$ that starts the induction). Moreover, to avoid large degrees in the graph, each time we need to use a value x_j^p we create a new copy of $v(x_j^p)$ using $\mathcal{G}_=$. This creates a binary tree with $(r-1)n^{r-1}$ leaves corresponding to copies of $v(x_j^p)$, each one of which is used once. Overall, we need to use $O((r-1)n^r + 2n-1 + (r-1)n^{r-1}) = O(rn^r)$ vertices to compute the desired quantities. Note that since all the quantities are in $[0, 1]$ we never use the ceiling 1 in propositions 7 and 8. \square

Construction of the graphical game \mathcal{GG}

We now have the tools to construct the graphical game \mathcal{GG} with bipartite graph $G = (V \sqcup W, E)$ from the normal-form game \mathcal{G} .

1. Rescale the utilities u^p of \mathcal{G} so that they lie in the range $[0, 1]$, for instance by dividing all the utilities by $\max(u^p(s))$.
2. For every player $p \in \{1, \dots, r\}$ and for every strategy $j \in \{1, \dots, n\} = S_p$ let $v(x_j^p) \in V$ be a vertex of \mathcal{G} .
3. For every player $p \in \{1, \dots, r\}$ construct a graph as in proposition 13 so that in a Nash equilibrium of \mathcal{GG} we have $\sum_{i=1}^n \mathbf{p}[v(x_i^p)] = 1$; add the vertices $w_j(p)$ and $w'_j(p)$ to W and the other vertices to V .

4. Using $\mathcal{G}_=$ create a binary tree whose leaves are $(r-1)n^{r-1}$ copies of $v(x_j^p)$. Add the middle vertices of the gadget to W and the other vertices to V .
5. For every player $p \in \{1, \dots, r\}$ and every strategy $j \in \{1, \dots, n\}$ add to V vertices $v(U_j^p)$ and $v(U_{\leq j}^p)$, as in proposition 14, using for each of the $v(U_j^p)$ its own set of copies of $v(x_j^p)$ constructed in the previous step. Add to V the vertices $v(\sum_{l=1}^z u^p(j, S_{-p}(l)) x_{S_{-p}(l)})$ and $v(u^p(j, s) \prod_{q \neq p, q=1}^z x_{s_q}^q)$ used in the construction of the $v(U_j^p)$ s and add to W the middle vertices of the gadgets used in the construction; when using \mathcal{G}_{\max} to construct the $v(U_{\leq j}^p)$ s add to V the vertices that in proposition 9 are denoted as v_i and add to W the vertices that are denoted w_i .
6. For every player $p \in \{1, \dots, r\}$ and every strategy $j \in \{1, \dots, n\}$ add to W vertices $w(U_j^p)$ such that $v(U_{\leq j}^p), v(U_{j+1}^p), w(U_j^p)$ form a brittle comparator $\mathcal{G}_<$ with $v(U_{\leq j}^p) = v_1, v(U_{j+1}^p) = v_2, w(U_j^p) = w$.
7. The payoffs of v_j^p are as follows:
 - if v_j^p plays 0 then it gets payoff 1 whenever $w(U_j^p)$ plays 0 and payoff 0 otherwise,
 - if v_j^p plays 1 then it gets payoff 1 whenever $w(U_j^p)$ plays 1 and payoff 0 otherwise;
that is, v_j^p imitates $w(U_j^p)$.



Polynomial size of \mathcal{GG}

Proposition 14 guarantees that the size of \mathcal{GG} is polynomial in the size of \mathcal{G} , which is $O(rn^r)$.

Mapping of Nash equilibria

We claim that from any Nash equilibrium of \mathcal{GG} we can recover a Nash equilibrium of \mathcal{G} by letting

$$x_j^p = \mathbf{p}[v(x_j^p)] \quad \text{for every } p \in \{1, \dots, r\}, j \in \{1, \dots, n\}. \quad (4)$$

This is clearly computable in polynomial time.

Let \mathcal{G}' be the game resulting from the rescaling of the utilities in step 1; since every Nash equilibrium of \mathcal{G}' is a Nash equilibrium of \mathcal{G} and vice versa, it is enough to prove that (4) recovers equilibria of \mathcal{G}'

from Nash equilibria of \mathcal{GG} . We already know from lemmas 13 and 14 that for x_j^p defined following (4) we have $x_j^p \geq 0$ and $\sum_j x_j^p = 1$; it remains to show that for every x_j^p recovered via (4) we have

$$\sum_{s \in S_{-p}} u^p(j, s) x_s > \sum_{s \in S_{-p}} u^p(j', s) x_s \Rightarrow x_{j'}^p = 0.$$

If there exists some $j'' < j'$ such that $\sum_{s \in S_{-p}} u^p(j'', s) x_s > \sum_{s \in S_{-p}} u^p(j', s) x_s$, then $\mathbf{p}[v(U_{\leq j'-1}^p)] > \mathbf{p}[v(U_{j'}^p)]$. Thus $\mathbf{p}[v_{j'-1}^p] = 0$ and $\mathbf{p}[v(x_{j'}^p)] = \mathbf{p}[v_{j'-1}^p](\mathbf{p}[v(\sum_{i=1}^{j'} x_i^p)]) = 0$. This holds in particular for the case for $j < j'$. On the other hand, if $j' < j$ and for all $j'' < j'$ we have $\mathbf{p}[v(U_{j''}^p)] < \mathbf{p}[v(U_{j'}^p)]$, this implies that $\mathbf{p}[v(U_{\leq j'}^p)] \leq \mathbf{p}[v(U_{j'}^p)]$, then that there exists some k , with $j' + 1 \leq k \leq j$ such that $\mathbf{p}[v(U_{\leq k-1}^p)] < \mathbf{p}[v(U_k^p)]$, because otherwise we would have $\mathbf{p}[v(U_{\leq j'}^p)] \geq \mathbf{p}[v(U_{\leq j'+1}^p)] \geq \dots \geq \mathbf{p}[v(U_{\leq j}^p)] \geq \mathbf{p}[v(U_j^p)] \geq \mathbf{p}[v(U_{j'}^p)]$, a contradiction. It follows that $\mathbf{p}[v(U_{k-1}^p)] = 1$, thus $\mathbf{p}[v_{k-1}^p] = 1$, so that $\mathbf{p}[v(\sum_{i=1}^{k-1} x_i^p)] = \mathbf{p}[v(\sum_{i=1}^k x_i^p)](1 - \mathbf{p}[v_{k-1}^p]) = 0$. But by lemma 13 this implies that $\mathbf{p}[v(\sum_{i=1}^{j'} x_i^p)] = 0$, which in turn implies that $\mathbf{p}[v(x_{j'}^p)] = 0$.

The mapping is surjective

Given a Nash equilibrium $\{x_j^p\}_{p \in \{1, \dots, r\}, j \in \{1, \dots, n\}}$ of \mathcal{G}' we show that there is a Nash equilibrium of \mathcal{GG} at which $\mathbf{p}[v(x_j^p)] = x_j^p$. We start by setting $\mathbf{p}[v(x_j^p)] = x_j^p$; then by lemma 14 the $\mathbf{p}[v(U_j^p)]$ s are the expected utilities to player p when the other players follow the mixed strategy profile $\{x_j^q\}_{q \in \{1, \dots, p-1, p+1, \dots, r\}, j \in S_q}$. Following the construction, if $\mathbf{p}[v(U_{\leq j}^p)] > \mathbf{p}[v(U_{j+1}^p)]$ let $\mathbf{p}[w(U_j^p)] = 0$ and $\mathbf{p}[v_j^p] = 0$; if $\mathbf{p}[v(U_{\leq j}^p)] < \mathbf{p}[v(U_{j+1}^p)]$ let $\mathbf{p}[w(U_j^p)] = 1$ and $\mathbf{p}[v_j^p] = 1$. If $\mathbf{p}[v(U_{\leq j}^p)] = \mathbf{p}[v(U_{j+1}^p)]$, then $\mathbf{p}[w(U_j^p)]$ and $\mathbf{p}[v_j^p]$ are arbitrary: choose $\mathbf{p}[w(U_j^p)] = \frac{1}{2}$ and

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

By proposition 13 the values of $\mathbf{p}[v_j^p]$ are consistent with the ones of $\mathbf{p}[v(x_j^p)]$, and we have a Nash equilibrium of \mathcal{GG} that yields the Nash equilibrium $\{x_j^p\}_{p \in \{1, \dots, r\}, j \in \{1, \dots, n\}}$ of \mathcal{G} .

Structure of the graph of \mathcal{GG}

All the vertices are included either in V or in W ; all the edges are between vertices in V and vertices in W : this makes the graph of \mathcal{GG} bipartite. \square

Like in the previous subsection, the theorem can be extended to the problem of finding ε -Nash equilibria: we give a sketch of the quite involute proof, referring to [DGP2] for details.

Theorem 15 (Theorem 5 of [GP], Theorem 9 of [DGP2]) *For every $r > 1$ there is a polynomial-time reduction from r -NASH to 3-GRAPHICAL NASH with two strategies per vertex.*

PROOF. Let \mathcal{G}' be a r -player normal-form game, and let \mathcal{G} be the game obtained from \mathcal{G}' by rescaling all the utilities so they lie in $[0, 1]$, for instance by dividing all the utilities by $\max(u^p(s))$ as in the first step of the algorithm of theorem 12. In time polynomial in $|\mathcal{G}| + \log(\frac{1}{\varepsilon})$ we will specify a graphical

game \mathcal{GG} and an accuracy parameter ε' such that given any ε' -Nash equilibrium of \mathcal{GG} we can recover an ε -Nash equilibrium of \mathcal{G} , and thus an ε -Nash equilibrium of \mathcal{G}' in polynomial time.

The game \mathcal{GG} will be the one constructed in theorem 12 and ε' will be set as $\varepsilon' = \frac{\varepsilon}{40r^2cn^{r+1}2^r}$; as in the proof of theorem 5 this is enough for the construction to take time polynomial in $|\mathcal{G}| + \log(\frac{1}{\varepsilon})$. The mapping of Nash equilibria, though, will not be done as in theorem 12 by taking $x_j^p = \mathbf{p}[v(x_j^p)]$ for all $p \in \{1, \dots, r\}$ and $j \in \{1, \dots, n\}$, since due to the approximation factors in the ε -Nash equilibrium condition in every gadget the $\mathbf{p}[v(x_j^p)]$ s do not necessarily yield neither a probability distribution nor a ε -Nash equilibrium (see Remark 2 of [DGP2] for details); instead, we will operate a *trim and normalise* transformation on the $\mathbf{p}[v(x_j^p)]$ s. Let c be the maximum error amplification of the gadgets used in the construction of \mathcal{GG} and let $x_j^p = \mathbf{p}[v(x_j^p)]$ for every $p \in \{1, \dots, r\}, j \in \{1, \dots, n\}$; the probability distributions describing a ε -Nash equilibrium of \mathcal{G} will be $\{\hat{x}_j^p\}_{p,j}$, obtained from $\{x_j^p\}_{p,j}$ by setting all the values at most $cn\varepsilon'$ equal to zero (trim) and then renormalising the resulting values so that $\sum_j \hat{x}_j^p = 1$, that is, by setting

$$\hat{x}_j^p = \begin{cases} \frac{x_j^p}{\Lambda^p} & \text{if } x_j^p > cn\varepsilon' \\ 0 & \text{if } x_j^p \leq cn\varepsilon' \end{cases} \quad (5)$$

where $\Lambda^p = \sum_{j \in S_p} x_j^p \mathcal{P}_{\{x_j^p \geq cn\varepsilon'\}}$.

The proof of the theorem relies on three lemmas, for the proof of which we refer to [DGP2]. The first lemma bounds the error amplification from the approximate equilibrium conditions applied to the x_j^p s.

Lemma 16 (Lemma 4 of [DGP2]) *In every ε' -Nash equilibrium of \mathcal{GG}*

- i. $|\sum_j \mathbf{p}[v(x_j^p)] - 1| \leq 2cn\varepsilon'$;
- ii. $\mathbf{p}[v(U_j^p)] > \mathbf{p}[v(U_j^p)] + 5cn\varepsilon' \Rightarrow \mathbf{p}[v(x_j^p)] \in [0, cn\varepsilon']$.

We already have from (5) that the $\{\hat{x}_j^p\}_{p,j}$ are a probability distribution; to show that they describe a ε -Nash equilibria of \mathcal{G} , we will use the two following lemmas: the first one guarantees that the U_j^p s work according to lemma 13 and measures their error, the second one will be used to measure the distance between the expected payoffs calculated using the x_j^p s and using the \hat{x}_j^p s with respect to the distance between the x_j^p s and the \hat{x}_j^p s.

Lemma 17 (Lemmas 6-8 of [DGP2]) *At an ε' -Nash equilibrium of \mathcal{GG} , for every $p \in \{1, \dots, r\}$ and every $j \in \{1, \dots, n\}$ it holds that*

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

where $\zeta_r = c\varepsilon' + ((1 + \zeta)^r - 1)(c\varepsilon' + 1)$ and $\zeta = 2r \log n c\varepsilon'$.

Lemma 18 (Lemma 9 of [DGP2]) *If $|x_j^p - y_j^p| < \delta < 1$ for every $p \in \{1, \dots, r\}$ and every $j \in \{1, \dots, n\}$ then for every p and every j*

$$\left| \sum_{s \in S_{-p}} u^p(j, s) x_s - \sum_{s \in S_{-p}} u^p(j, s) y_s \right| \leq \delta \cdot 2^r \cdot \sum_{s \in S_{-p}} u^p(j, s).$$

Now note that by definition of Λ^p we have $\Lambda^p = 1 - \sum_{j \in S_p} x_j^p \mathcal{R}_{\{x_j^p < cn\epsilon'\}} \geq 1 - n \cdot cn\epsilon'$ (since $|S_p| = n$), and moreover, by the definition of \hat{x}_j^p in (5)

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

so that $|\hat{x}_j^p - x_j^p| \leq \max(cn\epsilon', \frac{n^2 c \epsilon'}{1 - n^2 c \epsilon'})$. For $\delta_1 = \max(cn\epsilon', \frac{n^2 c \epsilon'}{1 - n^2 c \epsilon'})$ and $\delta_2 = \delta_1 \cdot 2^r \sum_{s \in S_{-p}} u^p(j, s)$, lemma 18 implies that

$$|\sum_{s \in S_p} u^p(j, s) x_s - \sum_{s \in S_p} u^p(j, s) \hat{x}_s| \leq \delta_2. \quad (6)$$

Note that $\delta_2 \leq \delta_1 \cdot 2^r \cdot n^{r-1}$ since all utilities lie in $[0, 1]$, and that $\delta_1 = \max(cn\epsilon', \frac{n^2 c \epsilon'}{1 - n^2 c \epsilon'}) \leq 2n^2 c \epsilon'$, so we have that

$$\delta_2 \leq \delta_1 \cdot 2^r \cdot n^{r-1} \leq 2n^2 c \cdot 2^r n^{r-1} \frac{\epsilon}{40r^2 cn^{r+1} 2^r} \leq \frac{\epsilon}{20r^2}. \quad (7)$$

We can also prove (see Lemma 5 of [DGP2]) that

$$2n^{r-1} \zeta_r \leq \frac{\epsilon}{n 2^r}. \quad (8)$$

Finally, we see that the condition for ϵ' -Nash equilibrium are satisfied by $\{\hat{x}_j^p\}_{p,j}$:

$$\begin{aligned} \sum_{s \in S_{-p}} u^p(j, s) \hat{x}_s &> \sum_{s \in S_{-p}} u^p(j', s) \hat{x}_s + \epsilon \Rightarrow \sum_{s \in S_{-p}} u^p(j, s) x_s + \delta_2 > \sum_{s \in S_{-p}} u^p(j', s) x_s - \delta_2 + \epsilon & (\text{by (6)}) \\ &\Rightarrow \mathbf{p}[v(U_j^p)] + 2n^{r-1} \zeta_r > \mathbf{p}[v(U_{j'}^p)] - 2n^{r-1} \zeta_r + \epsilon - 2\delta_2 & (\text{by Lemma 17}) \\ &\Rightarrow \mathbf{p}[v(U_j^p)] > \mathbf{p}[v(U_{j'}^p)] + \epsilon' & (\text{by (7), (8) and the definition of } \epsilon') \\ &\Rightarrow x_j^p = \mathbf{p}[v(x_j^p)] \leq cn \frac{\epsilon'}{5cn} < cn\epsilon' & (\text{by Lemma 16}) \\ &\Rightarrow \hat{x}_j^p = 0 & (\text{by the definition of } \hat{x}_j^p) \end{aligned}$$

□

2.3 4-NASH

The theorems of the previous subsections give the following immediate result:

Corollary 19 (Corollary 2 of [GP]) *For every $r > 1$ there is a polynomial-time reduction from r -NASH to 10-NASH.*

PROOF. The r -player game is reduced to a graphical game of maximum degree 3, which in turn is reduced to a $(3^2 + 1)$ -player game. □

A stronger result is the main one of [GP]: solving a 4-player game is as hard as solving any r -player game. Its proof combines the results and the techniques from the previous subsections: given a normal-form game \mathcal{G} we will construct a graphical game \mathcal{GG} as in subsection 2.2; then we will give a slightly

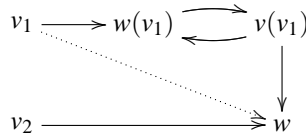
modified version of \mathcal{GG} , that we call \mathcal{GG}' , whose moralised graph has a proper 4-colouring. Finally, on the colouring of \mathcal{GG}' we will construct a 4-player normal-form game \mathcal{G}' with the “super-players” technique of subsection 2.1.

Theorem 20 (Theorem 3 of [GP], Theorem 6 of [DGP2]) *For every $r > 1$ a r -player normal-form game can be mapped in polynomial time to a 4-player normal-form game, so that there is a polynomially computable surjective mapping from the Nash equilibria of the latter to the Nash equilibria of the former.*

PROOF. Let \mathcal{G} be a r -player normal-form game.

Construction of the 4-player game

1. Construct from \mathcal{G} a graphical game \mathcal{GG} following the algorithm given in theorem 12. Let $G = (V \sqcup W, E)$ be the affects-graph of \mathcal{GG} .
2. Let $c(w) = 1$ for all $w \in W$, and let $c(v) = 2$ for all $v \in V$.
3. Construct the graphical game \mathcal{GG}' from \mathcal{GG} as follows:
while there exist $v_1, v_2 \in V$ and $w \in W$ such that $(v_1, w), (v_2, w) \in E$ and $c(v_1) = c(v_2)$, assuming that $(w, v_1) \notin E$ (this can be done without loss of generality since w has at most one outgoing edge)
 - i. add $v(v_1)$ to V , $w(v_1)$ to W ;
 - ii. replace (v_1, w) with $(v_1, w(v_1)), (w(v_1), v(v_1)), (v(v_1), w(v_1)), (v(v_1), w)$ where v_1 , $w(v_1)$ and $v(v_1)$ are connected in a $\mathcal{G}_=$ gadget, so that at the equilibrium $\mathbf{p}[v(v_1)] = \mathbf{p}[v_1]$;
 - iii. let $c(w(v_1)) = 1$. Let $c(v(v_1)) \in \{2, 3, 4\}$ such that it is different from the colour of any other vertex v' such that $(v', w) \in E$ (note that this ensures a proper colouring of the moralised graph of $\mathcal{G}_=$ as well, since the $c(v_1) = c(v_2)$ and there is still an edge (v_2, w)).



4. Map \mathcal{GG}' to a normal-form game \mathcal{G}' using steps 4 to 6 of the algorithm given in theorem 3 (or, equivalently, corollary 6).

Polynomial size of \mathcal{G}'

Theorem 12 ensures that \mathcal{GG} is polynomial in the size of \mathcal{G} . The size of \mathcal{GG}' is at most three times the size of \mathcal{GG} , since step 3 is not applied to the edges that are constructed by earlier iterations of step 3 itself. Finally, theorem 3 ensures that \mathcal{G}' is polynomial in the size of \mathcal{GG}' .

Surjective mapping of the equilibria

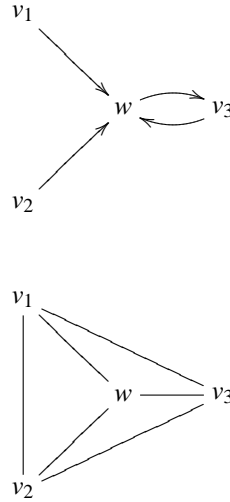
Let g_1 be the surjective mapping from the Nash equilibria of \mathcal{GG} to the Nash equilibria of \mathcal{G} given by theorem 12. Let g_2 be a mapping from the equilibria of \mathcal{GG}' to the equilibria of \mathcal{GG} (it is sufficient to ignore the vertices constructed in step 3). Finally, let g_3 be the surjective mapping from the equilibria of \mathcal{G}' to the equilibria of \mathcal{GG}' given by theorem 3. Then $g_3 \circ g_2 \circ g_1$ is a surjective mapping from the Nash equilibria of \mathcal{G}' to the Nash equilibria of \mathcal{G} . \square

As theorem 20 followed from theorems 3 and 12, from theorems 5 and 15 follows

Theorem 21 (Theorem 6 of [GP]) *For every $r > 1$ there is a polynomial-time reduction from r -NASH to 4-NASH.*

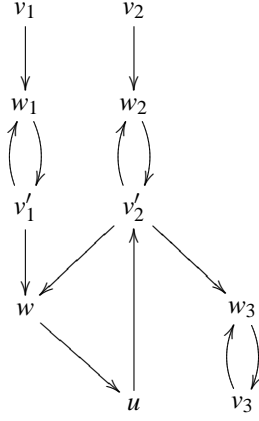
2.4 3-NASH

In the last part of this section we relate the main result of [DP]: there is a reduction from r -NASH to 3-NASH. To prove this result we will introduce a new gadget, that used instead of the one of proposition 8 when constructing a graphical game \mathcal{GG} from a normal-form game as in subsection 2.2 produces a 3-colourable moralised graph of \mathcal{GG} : the result will then follow from corollary 6. Notice that a 3-colouring of the moralised graph of \mathcal{GG} would be impossible with the gadget of proposition 8, and with $\mathcal{G}_+, \mathcal{G}_-, \mathcal{G}_*, \mathcal{G}_{\max}$ constructed from it, since its moralised graph is the 4-clique.

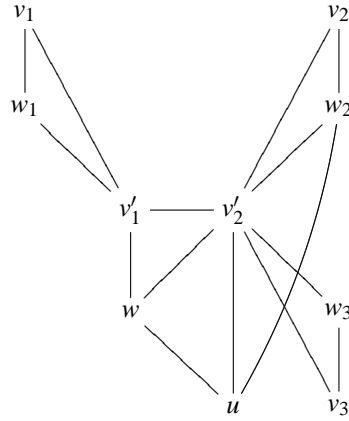


Proposition 22 (Lemma 3 of [DP], Proposition 6 of [DGP2]) *Let α, β, γ be non-negative integers such that $\alpha + \beta + \gamma \leq 3$. There is a graphical game with input players v_1, v_2 , output player v_3 and several intermediate players such that for any $\varepsilon \in [0, 0.01]$ at any ε -Nash equilibrium of the game it holds that $\mathbf{p}[v_3] = \min(\alpha \mathbf{p}[v_1] + \beta \mathbf{p}[v_2] + \gamma \mathbf{p}[v_1] \mathbf{p}[v_2], 1) \pm 81\varepsilon$, and moreover the moralised graph of the game is 3-colourable.*

PROOF. The game's affects-graph is



and its moralised graph is thus



a proper 3-colouring of which is $c(v) = 1$ for $v = v_1, w, w_2, w_3$, then $c(v) = 2$ for $v = v'_1, v'_2$, and finally $c(v) = 3$ for $v = v_2, u, v_3$. The payoffs are as follows:

1. game played by v_1, w_1, v'_1 :

payoffs for v'_1 :

$v'_1 \backslash w_1$	0	1
0	0	1
1	1	0

payoffs for w_1 :

$w_1 \backslash v'_1$	0	1
0	0	0
1	$\frac{1}{8}$	$\frac{1}{8}$

$w_1 \backslash v'_1$	0	1
0	0	1
1	0	1

2. game played by v_2, w_2, v'_2 :

payoffs for w_2 :

		v_2		0	1
w_2 plays 0:	v'_2	0	0	$\frac{1}{8}$	
		1	0	$\frac{1}{8}$	
		*	0	$\frac{1}{8}$	

		v_3		0	1
w_2 plays 1:	v'_2	0	0	0	
		1	1	1	
		*	0	0	

payoffs for v'_2 playing respectively 0, 1 and *:

		w_2		0	1
u	0	0	1		
	1	0	0		

		w_2		0	1
u	0	1	0		
	1	1	0		

		w_2		0	1
u	0	0	0		
	1	0	1		

3. game played by v'_1, v'_2, w, u :

payoffs for w , with $\delta = 1 + \alpha + \beta + 8\gamma$:

		v'_1		0	1
w plays 0:	v'_2	0	0	α	
		1	$1 + \beta$	δ	
		*	0	α	

		v'_1		0	1
w plays 1:	v'_2	0	0	0	
		1	1	1	
		*	1	1	

payoffs for u :

		w		0	1
u	0	0	1		
	1	1	0		

4. game played by v'_2, w_3, v_3 :

payoffs for v_3 :

		w_3		0	1
v_3	0	0	1		
	1	1	0		

payoffs for w_3 :

		v_3		0	1
		v_2		0	1
w_3 plays 0:	0		0	0	0
	1		0	0	0
	*		8	8	

		v_3		0	1
		v_2		0	1
w_3 plays 1:	0		0	1	
	1		0	1	
	*		0	1	

The four parts of the new gadget are analysed in the following lemmas:

Lemma 23 (Claim 1 of [DP] and [DGP2]) At any ε -Nash equilibrium of the game above, $\mathbf{p}[v'_1] = \frac{1}{8}\mathbf{p}[v_1] \pm \varepsilon$

PROOF. The expected payoff of w_1 is $\frac{1}{8}\mathbf{p}[v_1]$ for playing 0 and $\mathbf{p}[v'_1]$ for playing 1. Therefore, if $\frac{1}{8}\mathbf{p}[v_1] > \mathbf{p}[v'_1] + \varepsilon$, in any ε -Nash equilibrium we have that $\mathbf{p}[w_1] = 0$. But this implies that $\mathbf{p}[v'_1] = 1$, a contradiction. Hence, $\frac{1}{8}\mathbf{p}[v_1] \leq \mathbf{p}[v'_1] + \varepsilon$. On the other hand, if $\frac{1}{8}\mathbf{p}[v_1] < \mathbf{p}[v'_1] - \varepsilon$ then $\mathbf{p}[w_1] = 1$ and $\mathbf{p}[v'_1] = 0$, again a contradiction. Hence, $\frac{1}{8}\mathbf{p}[v_1] \geq \mathbf{p}[v'_1] - \varepsilon$. \square

Lemma 24 (Claim 2 of [DP] and [DGP2]) At any ε -Nash equilibrium of the game above, $\mathbf{p}[v'_2 \text{ plays } 1] = \frac{1}{8}\mathbf{p}[v_2] \pm \varepsilon$

PROOF. The expected payoff of w_2 is $\frac{1}{8}\mathbf{p}[v_2]$ for playing 0 and $\mathbf{p}[v'_2 \text{ plays } 1]$ for playing 1. Therefore, if $\frac{1}{8}\mathbf{p}[v_2] > \mathbf{p}[v'_2 \text{ plays } 1] + \varepsilon$, in any ε -Nash equilibrium we have that $\mathbf{p}[w_2] = 0$. In this regime, the payoff to v'_2 is 0 for playing 0, it is 1 for playing 1 and it is 0 for playing *. Therefore, $\mathbf{p}[v'_2 \text{ plays } 1] = 1$; but this gives a contradiction with $\frac{1}{8}\mathbf{p}[v_2] > \mathbf{p}[v'_2 \text{ plays } 1] + \varepsilon$. Hence, $\frac{1}{8}\mathbf{p}[v_2] \leq \mathbf{p}[v'_2 \text{ plays } 1] + \varepsilon$. On the other hand, if $\frac{1}{8}\mathbf{p}[v_2] < \mathbf{p}[v'_2 \text{ plays } 1] - \varepsilon$, in a ε -Nash equilibrium $\mathbf{p}[w_2] = 1$. In this regime, the payoff to v'_2 is $\mathbf{p}[u \text{ plays } 0] = 1 - \mathbf{p}[u]$ for playing 0, it is 0 for playing 1 and it is $\mathbf{p}[u]$ for playing *. Since at least one of $(1 - \mathbf{p}[u])$ and $\mathbf{p}[u]$ is greater than ε , we have that $\mathbf{p}[v'_2] = 0$, again a contradiction. Hence, $\frac{1}{8}\mathbf{p}[v_2] \geq \mathbf{p}[v'_2 \text{ plays } 1] - \varepsilon$. \square

Lemma 25 (Claim 3 of [DP] and [DGP2]) At any ε -Nash equilibrium of the game above, $\mathbf{p}[v'_2 \text{ plays } *] = \frac{\alpha}{8}\mathbf{p}[v_1] + \frac{\beta}{8}\mathbf{p}[v_2] + \frac{\gamma}{8}\mathbf{p}[v_1]\mathbf{p}[v_2] \pm 10\varepsilon$

PROOF. The expected payoff of w is $\alpha\mathbf{p}[v'_1] + (1 + \beta)\mathbf{p}[v'_2 \text{ plays } 1] + 8\gamma\mathbf{p}[v'_1]\mathbf{p}[v'_2 \text{ plays } 1]$ if it plays 0 and $\mathbf{p}[v'_2 \text{ plays } 1] + \mathbf{p}[v'_2 \text{ plays } *]$ if it plays 1. Therefore, if in a ε -Nash equilibrium

$$\alpha\mathbf{p}[v'_1] + (1 + \beta)\mathbf{p}[v'_2 \text{ plays } 1] + 8\gamma\mathbf{p}[v'_1]\mathbf{p}[v'_2 \text{ plays } 1] > \mathbf{p}[v'_2 \text{ plays } 1] + \mathbf{p}[v'_2 \text{ plays } *] + \varepsilon,$$

then $\mathbf{p}[w] = 0$ and consequently $\mathbf{p}[u] = 1$. In this regime, the payoff of v'_2 is 0 if it plays 0, it is $\mathbf{p}[w_2 \text{ plays } 0] = 1 - \mathbf{p}[w_2]$ if it plays 1, and it is $\mathbf{p}[w_2]$ if it plays *. Since at least one of $1 - \mathbf{p}[w_2]$, $\mathbf{p}[w_2]$ is larger than ε , we have that $\mathbf{p}[v'_2 \text{ plays } 0] = 0$ and $\mathbf{p}[v'_2 \text{ plays } 1] + \mathbf{p}[v'_2 \text{ plays } *] = 1$. The hypothesis can be thus rewritten as $\alpha\mathbf{p}[v'_1] + (1 + \beta)\mathbf{p}[v'_2 \text{ plays } 1] + 8\gamma\mathbf{p}[v'_1]\mathbf{p}[v'_2 \text{ plays } 1] > 1 + \varepsilon$, which using

the previous lemmas becomes $\frac{\alpha}{8}\mathbf{p}[v_1] + \frac{1+\beta}{8}\mathbf{p}[v_2] + \frac{\gamma}{8}\mathbf{p}[v_1]\mathbf{p}[v_2] + (\alpha + 1 + \beta + 3\gamma)\varepsilon > 1 + \varepsilon$, that since $\mathbf{p}[v_i] \leq 1$ implies that $\frac{1+\alpha+\beta+\gamma}{8} + (\alpha + 1 + \beta + 3\gamma)\varepsilon > 1 + \varepsilon$. But since we assumed that $\alpha + \beta + \gamma \leq 3$ this implies that $\frac{1}{2} + 10\varepsilon > 1 + \varepsilon$, a contradiction since we assumed $\varepsilon \leq 0.01$.

On the other hand, if in a ε -Nash equilibrium

$$\alpha\mathbf{p}[v'_1] + (1 + \beta)\mathbf{p}[v'_2 \text{ plays } 1] + 8\gamma\mathbf{p}[v'_1]\mathbf{p}[v'_2 \text{ plays } 1] < \mathbf{p}[v'_2 \text{ plays } 1] + \mathbf{p}[v'_2 \text{ plays } *] - \varepsilon,$$

then $\mathbf{p}[w] = 1$ so that $\mathbf{p}[u] = 0$. In this regime, the payoff of v'_2 is $\mathbf{p}[w_2]$ if it plays 0, it is $\mathbf{p}[w_2 \text{ plays } 0] = 1 - \mathbf{p}[w_2]$ if it plays 1, and it is 0 if it plays *. It follows that $\mathbf{p}[v'_2 \text{ plays } *] = 0$, so the hypothesis can be rewritten as $\alpha\mathbf{p}[v'_1] + \beta\mathbf{p}[v'_2 \text{ plays } 1] + 8\gamma\mathbf{p}[v'_1]\mathbf{p}[v'_2 \text{ plays } 1] < -\varepsilon$, a contradiction.

Thus we have that in any ε -Nash equilibrium

$$\alpha\mathbf{p}[v'_1] + (1 + \beta)\mathbf{p}[v'_2 \text{ plays } 1] + 8\gamma\mathbf{p}[v'_1]\mathbf{p}[v'_2 \text{ plays } 1] = \mathbf{p}[v'_2 \text{ plays } 1] + \mathbf{p}[v'_2 \text{ plays } *] \pm \varepsilon,$$

which using the previous lemmas yields

$$\mathbf{p}[v'_2 \text{ plays } *] = \frac{\alpha}{8}\mathbf{p}[v_1] + \frac{\beta}{8}\mathbf{p}[v_2] + \frac{\gamma}{8}\mathbf{p}[v_1]\mathbf{p}[v_2] \pm 10\varepsilon.$$

□

Lemma 26 (Claim 4 of [DP] and [DGP2]) *At any ε -Nash equilibrium of the game above, $\mathbf{p}[v_3] = \min(\alpha\mathbf{p}[v_1] + \beta\mathbf{p}[v_2] + \gamma\mathbf{p}[v_1]\mathbf{p}[v_2]) \pm 81\varepsilon$*

PROOF. The expected payoff to w_3 is $8\mathbf{p}[v'_2 \text{ plays } *]$ if it plays 0 and $\mathbf{p}[v_3]$ if it plays 1. Therefore, if in a ε -Nash equilibrium $\mathbf{p}[v_3] > 8\mathbf{p}[v'_2 \text{ plays } *] + \varepsilon$ then $\mathbf{p}[w_3] = 1$ and consequently $\mathbf{p}[v_3] = 0$, a contradiction. Therefore, in any ε -Nash equilibrium $\mathbf{p}[v_3] \leq 8\mathbf{p}[v'_2 \text{ plays } *] + \varepsilon$. On the other hand, if $\mathbf{p}[v_3] < 8\mathbf{p}[v'_2 \text{ plays } *] - \varepsilon$, in any ε -Nash equilibrium $\mathbf{p}[w_3] = 0$ and $\mathbf{p}[v_3] = 1$. Therefore, $\mathbf{p}[v_3] \geq \min(8\mathbf{p}[v'_2 \text{ plays } *] - \varepsilon, 1)$. Hence,

$$\mathbf{p}[v_3] = \min(8\mathbf{p}[v'_2 \text{ plays } *], 1) \pm \varepsilon = \frac{\alpha}{8}\mathbf{p}[v_1] + \frac{\beta}{8}\mathbf{p}[v_2] + \frac{\gamma}{8}\mathbf{p}[v_1]\mathbf{p}[v_2], 1) \pm (8 \cdot 10 + 1)\varepsilon.$$

□

This concludes the proof of proposition 22. □

Theorem 27 ([DP], Theorem 7 of [DGP2]) *For every $r > 1$, a r -player normal-form game can be mapped in polynomial time to a 3-player normal-form game so that there is a polynomially computable surjective mapping from the Nash equilibria of the latter to the Nash equilibria of the former.*

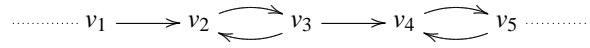
PROOF. Let \mathcal{G} be a r -player normal-form game. Construct from \mathcal{G} a graphical game \mathcal{GG} following the algorithm of theorem 12, but instead of the gadget of proposition 8 use the new gadget of proposition 22, and instead of identifying the output vertex of a gadget with the input vertex of another gadget make a “connection” interposing two $\mathcal{G}_=$ gadgets between the vertices. Proposition 22 ensures that the reduction

of theorem 12 still holds; note that lemma 14 still holds as well, so the construction of the new gadget and of the new connections do not change the polynomial time of the construction of \mathcal{GG} .

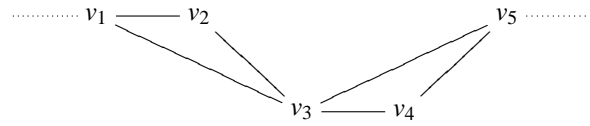
It remains to be seen that \mathcal{GG} can be given a colouring that is proper for its moralised graph as well: such a task is accomplished by

1. colouring all the gadgets except the “connections”;
2. colouring all the “connections”.

All the gadgets coloured in step 1 either consist of 3 or less vertices or are the new gadget, 3-colourable as seen in proposition 22. The connections can always be given a proper colouring that is so for the moralised graph as well, whatever the colour of their first and last vertices. For instance, in the connection with affects-graph

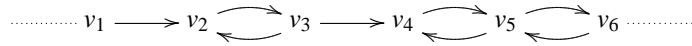


and moralised graph

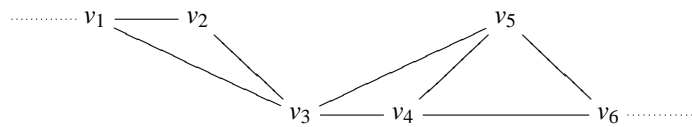


starting from any colour in v_1 and v_5 , we can give a proper colouring of the vertices v_2, v_3, v_4 : we give v_3 the colour that is neither of v_1 nor of v_5 (one of the other colours if v_1 and v_5 have the same colour) then the colours of v_2 and v_4 follow.

Notice that the colouring works also because the gadgets \mathcal{G}_α are never after a connection (we never set a quantity equal to a fixed value): if this were not the case, we could have an affects-graph



resulting in the moralised graph



in which, if the colouring of the gadgets were such that $c(v_1) = c(v_6) = 1$, $c(v_5) = 2$, a proper colouring of the moralised graph would be impossible: we should have $c(v_3) = 3$, but then it would be impossible to colour v_4 .

Now corollary 6 ensures that we can construct from \mathcal{GG} a 3-player normal-form game \mathcal{G}' and a polynomially computable surjective mapping g_1 from the Nash equilibria of \mathcal{G}' to the Nash equilibria of \mathcal{GG} . Since from the reduction of theorem 12 we have a polynomially computable surjective mapping g_2 from the Nash equilibria of \mathcal{GG} to the Nash equilibria of \mathcal{G} , we have a polynomially computable surjective mapping $g_2 \circ g_1$ from the Nash equilibria of \mathcal{G}' to the Nash equilibria of \mathcal{G} .

Finally, corollary 6 ensures that the construction of \mathcal{G}' takes time polynomial in the size of \mathcal{G} ; since we have already seen that the construction of the latter takes time polynomial in the size of \mathcal{G} we have that the construction of \mathcal{G}' takes time polynomial in the size of \mathcal{G} . \square

Using theorem 15 instead of theorem 12 allows us to extend theorem 27 to

Theorem 28 (*[DP], Theorem 10 of [DGP2]*) *For every fixed $r > 1$, there is a polynomial-time reduction from r -NASH to 3-NASH.*

The results of this section can be summed up in

Theorem 29 (*Theorem 3 of [DGP2]*) *For every fixed $r, d \geq 3$,*

- *every r -player normal-form game and every graphical game of maximum degree d can be mapped in polynomial time to both a 3-player normal-form game and a binary graphical game of maximum degree 3, such that there is a polynomially computable surjective mapping from the Nash equilibria of the latter to the Nash equilibria of the former; and*
- *there are polynomial-time reductions from r -NASH and d -GRAPHICAL NASH to both 3-NASH and 3-GRAPHICAL NASH.*

3 PPAD-completeness

Once established that any result on the complexity of 3-NASH and 3-GRAPHICAL NASH yields a result on the complexity of r -NASH and d -GRAPHICAL NASH for $r, d \geq 3$, we try to define better to which complexity class these problems belong to. We have already seen in subsection 1.3 that finding a ε -Nash equilibrium of a r -player game is a problem in PPAD; now we will show that both r -NASH and d -GRAPHICAL NASH are PPAD-complete for $r \geq 2$ and $d \geq 3$. We will start by dealing with the case $r, d \geq 3$, first proved in [DGP] and [DP], then we will relate the case $r = 2$, first proved in [CD]; in both cases we will follow the proofs given in [DGP2]. In both proofs the problem 3-DIMENSIONAL BROUWER will be pivotal; this is a discrete version of the problem of finding a fixed point of a continuous function from the 3-dimensional unit cube in itself.

We will start this section by giving a formal definition of 3-DIMENSIONAL BROUWER and the proof of its PPAD-completeness found in [DGP]. In the second subsection we will then present the reduction, first seen in [DGP] and given in greater detail in [DGP2], from 3-DIMENSIONAL BROUWER to 3-GRAPHICAL NASH; by the result of the first subsection and by theorem 29 this implies the PPAD-completeness of r -NASH and d -GRAPHICAL NASH for every $r, d \geq 3$. In the last subsection we will relate the proof given in [DGP2] of the PPAD-completeness of 2-NASH.

3.1 3-DIMENSIONAL BROUWER

Consider the 3-dimensional unit cube subdivided in 2^{3m} cubelets K_{ijk} with side 2^{-m} . Formally, let

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

A *Brouwer function* is a function ϕ defined at the centres c_{ijk} of these cubelets such that $\phi(c_{ijk}) = c_{ijk} + \delta_{ijk}$ with four possible displacements δ_{ijk} :

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

where we take $\alpha > 0$ much smaller than the cubelet side – for example, it can be 2^{-2m} . Furthermore, we require that at the boundary of the unit cube ϕ always “moves inside”, so that it is a function from the unit cube to itself. A vertex shared by eight cubelets is called *panchromatic* if in these cubelets all the possible displacements are represented; by interpolation, this corresponds to a fixed point of ϕ in the continuous version. We can now define the problem

3-DIMENSIONAL BROUWER

Input: The function ϕ computed by the circuit C with $3m$ input bits and 2 output bits and boundary conditions $C(0, j, k) = 1$, $C(i, 0, k) = 2$, $C(i, j, 0) = 3$, $C(i, j, k) = 0$ if i, j or k is $2^m - 1$.

Output: A panchromatic vertex in the unit cube.

As noted in subsection 1.3, in [P94] is shown that 3-DIMENSIONAL BROUWER is equivalent to 3-D SPERNER, thus PPAD; we now relate the proof of its PPAD-completeness as given in [DGP]. Another proof of the PPAD-completeness of k -D SPERNER for $k \geq 3$ is given in [P94]; the proof of [DGP] is less general (since it is limited to the 3-dimensional case) but more immediate.

To prove the PPAD-completeness of 3-DIMENSIONAL BROUWER we will first translate the graph implicit in END OF THE LINE into a path L each step of which joins two adjacent of the cubelets in which the unit cube is divided in 3-DIMENSIONAL BROUWER; then we will colour the cubelets with four colours corresponding to the four displacements, depending on the cubelets' position with respect to the path; finally we will check that the panchromatic vertices in this colouring correspond exactly to the non-standard sources and the sinks of the END OF THE LINE graph and that the construction requires polynomial time.

Theorem 30 (*Theorem 2 of [DGP]*) 3-DIMENSIONAL BROUWER is PPAD-complete

PROOF. Consider the problem END OF THE LINE with n input and n outputs, and the problem 3-DIMENSIONAL BROUWER on a cube divided into 2^{3m} cubelets, with $m = (n + 3)$.

Construction of the path

Let $G_{S,P}$ be the graph implicit in END OF THE LINE. We assume that for every edge (u, v) one of the vertices is even (ends in 0) and the other is odd: since we can always consider the problem on $n + 1$ vertices adding a dummy input and output bit, this assumption holds without loss of generality.

For every vertex $u \in \{0, 1\}^n$ of $G_{S,P}$ let $[u]$ be the integer between 0 and $2^n - 1$ whose binary representation is u . We omit the factor 2^{-m} when giving coordinates in the cube, that is, instead of giving the distance from the origin we consider the “coordinate” given by the division in 2^{3m} cubelets. We associate the following segments to u :

- i. the *principal segment*, with endpoints $u_1 = (8[u] + 2, 3, 3)$ and $u'_1 = (8[u] + 6, 3, 3)$;
- ii. the *auxiliary segment*, with endpoints $u_2 = (3, 8[u] + 2, 2^m - 3)$ and $u'_2 = (3, 8[u] + 6, 2^m - 3)$;
- iii. a connecting path, with joints $u'_1, u_3 = (8[u] + 6, 8[u] + 2, 3), u_4 = (8[u] + 6, 8[u] + 2, 2^m - 3)$ and u_2 .

Then, for every edge (u, v) of $G_{S,P}$ we connect u'_2 to v_1 via $u_5 = (8[v] + 2, 8[u] + 6, 2^m - 3)$ and $u_6 = (8[v] + 2, 8[u] + 6, 3)$. Finally, for $u = 0$, we set $0_1 = (2, 2, 2)$ and $0'_1 = (6, 2, 2) = 0_3$.

Notice how the hypothesis $m = n + 3$ is necessary to remain in the cube when multiplying by 8, and how we can univocally colour the cubelets because L traverses the cube without “nearly crossing itself”: if two points of L are closer than $4 \cdot 2^{-m}$ in Euclidean distance then they are connected by a part of L that has length $8 \cdot 2^{-m}$ or less, that is, they belong to the same segment or they are in the proximity of an endpoint connecting the segments to which they belong. To check this, recall that segments of different types come close only if they share an endpoint, that all segments on the plane $z = 3$ are parallel and at

least $4 \cdot 2^{-m}$ apart and that, analogously, all segments parallel to the z axis differ by at least 4 in their coordinates. This concludes the description of the path L .

Colouring of the cubelets

The main idea is now to colour the interior of the cube with 0 except when “surrounding” the interior of L and 0_1 by colours 1, 2, 3: this way we will have panchromatic vertices only at the sinks and at the nonstandard sources. Formally:

1. every K_{ijk} with any i, j, k equal to $2^m - 1$ is colored 0;
2. every K_{ijk} with any $i = 0$ is colored 1;
3. every K_{ijk} with any $j = 0$ is colored 2;
4. every K_{ijk} with any $k = 0$ is colored 3
(the steps so far correspond to the boundary conditions given for the circuit).
5. Every remaining K_{ijk} is colored 0.
6. Every K_{ijk} that has a vertex which is a point of the interior of L is coloured as follows. Two of the cubelets surrounding the segment $[a, b]$, where b is the vertex in the interior of L , are colored 3 and the other two cubelets surrounding $[a, b]$ are colored 1 and 2, clockwise, with the direction d in which the cubes colored 3 lying as follows:
 - i. 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;
 - ii. 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;
 - iii. 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;
 - iv. 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;
 - v. 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;
 - vi. 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;
 - vii. 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;
 - viii. 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.

Notice that for the construction to be consistent we need the assumption that edges of $G_{S,P}$ go between even and odd nodes; notice also that if the edges of $G_{S,P}$ were not between even and odd vertices, and if we would introduce “twists” to make all colourings at (u_1, u'_1) point in a given direction we would have a panchromatic vertex at each twist.

The computation takes polynomial time

We now show that the circuit C computing the colour of $K_{i,j,k}$ for each triple of binary integers i, j, k can be constructed in time polynomial in $|S| + |P|$. Let $LSB(x)$ be the least significant bit of x , equal to 1 if x is odd, to 0 if x is even, and undefined for x not integer; circuit C is defined as:

1. $C(0, j, k) = 1$;
2. $C(i, 0, k) = 2$ for $i > 0$;
3. $C(i, j, 0) = 3$ for $i, j > 0$;
4. $C(i, j, k) = 0$ for $i, j, k > 0$ except for cubelets adjacent to L .
5. i. For every cubelet adjacent to a segment between u_1 and u'_2 the calculation of $C(i, j, k)$ depends only on the coordinate where the circuit is calculated: this will take constant time. We describe for instance C on the cubelets surrounding a $[u'_1, u_3]$ segment; recall that in the colouring given to the cubelets surrounding such a segment the two cubelets with colour 3 lie in direction $(0, 0, 1)$ for u even and in direction $(0, 0, -1)$ for u odd:
 - if $i = 8x + 5$ with $LSB(x) = 1$, $j \in \{3, \dots, 8x + 2\}$, $k = 2$ then $C(i, j, k) = 2$;
 - if $i = 8x + 6$ with $LSB(x) = 1$, $j \in \{2, \dots, 8x + 2\}$, $k = 2$ then $C(i, j, k) = 1$;
 - if $i = 8x + 5$ or $i = 8x + 6$ with $LSB(x) = 1$, $j \in \{2, \dots, 8x + 1\}$, $k = 3$ then $C(i, j, k) = 3$;
 - if $i = 8x + 5$ with $LSB(x) = 0$, $j \in \{3, \dots, 8x + 2\}$, $k = 3$ then $C(i, j, k) = 1$;
 - if $i = 8x + 6$ with $LSB(x) = 0$, $j \in \{2, \dots, 8x + 1\}$, $k = 3$ then $C(i, j, k) = 2$;
 - if $i = 8x + 5$ or $i = 8x + 6$ with $LSB(x) = 0$, $j \in \{2, \dots, 8x + 1\}$, $k = 2$ then $C(i, j, k) = 3$.
- ii. For the cubelets adjacent to segments between u'_2 and v_1 the calculation involves also S and P , taking time polynomial in $|S| + |P|$. As an example, consider the description of the segment $[u'_2, u_5]$; recall that in the colouring given to such a segment the two cubelets with colour 3 in direction $(0, -1, 0)$ for u even and in direction $(0, 1, 0)$ for u odd:
 - if $i \in \{3, \dots, 8x' + 2\}$, $j = 8x + 5$ with $LSB(x) = 1$, $k = 2^m - 3$, and if $S(x) = x'$, $P(x') = x$ then $C(i, j, k) = 1$;
 - if $i \in \{2, \dots, 8x' + 1\}$, $j = 8x + 5$ with $LSB(x) = 1$, $k = 2^m - 4$, and if $S(x) = x'$, $P(x') = x$ then $C(i, j, k) = 2$;
 - if $i \in \{2, \dots, 8x' + 2\}$, $j = 8x + 6$ with $LSB(x) = 1$, $k = 2^m - 4$ or $k = 2^m - 3$, and if $S(x) = x'$, $P(x') = x$ then $C(i, j, k) = 1$;
 - if $i \in \{2, \dots, 8x' + 2\}$, $j = 8x + 6$ with $LSB(x) = 0$, $k = 2^m - 3$, and if $S(x) = x'$, $P(x') = x$ then $C(i, j, k) = 2$;
 - if $i \in \{2, \dots, 8x' + 1\}$, $j = 8x + 6$ with $LSB(x) = 0$, $k = 2^m - 4$, and if $S(x) = x'$, $P(x') = x$ then $C(i, j, k) = 1$;
 - if $i \in \{3, \dots, 8x' + 1\}$, $j = 8x + 5$ with $LSB(x) = 0$, $k = 2^m - 4$ or $k = 2^n - 3$, and if $S(x) = x'$, $P(x') = x$ then $C(i, j, k) = 3$.

The reduction is correct

All the panchromatic vertices correspond to sinks or nonstandard sources of L , since L is surrounded by colours 1, 2, 3 and it can meet colour 0, hence a panchromatic vertex, only at the endpoint u'_2 of a sink

vertex or at the endpoint u_1 of a source other than 0^n . Notice that the condition $0'_1 = 0_3$ ensures that the cubelets surrounding 0^n are not panchromatic. \square

3.2 3-NASH and 3-GRAPHICAL NASH

We now relate the reduction from 3-DIMENSIONAL BROUWER to 3-GRAPHICAL NASH: given a circuit C with $3n$ input bits and 2 output bits, we will construct a binary graphical game \mathcal{G} of maximum degree 3 and an accuracy parameter ε such that for every ε -Nash equilibria of \mathcal{G} it is possible to find in polynomial time a panchromatic vertex of the 3-DIMENSIONAL BROUWER instance.

More precisely, \mathcal{G} will simulate C via gadgets and in its graph that there will be three vertices v_x, v_y, v_z of \mathcal{G} that represent (as in the gadgets of subsection 2.2) the coordinates of a point that is close to a panchromatic vertex in the unit cube. The game \mathcal{G} will consist of three parts: first the gadgets introduced in subsection 2.2 will compute the binary representation, with precision to the n -th bit, of the integers (i, j, k) such that $(\mathbf{p}[v_x], \mathbf{p}[v_y], \mathbf{p}[v_z])$ lies in the cubelet K_{ijk} ; we will call this triplet (x, y, z) . The second part of \mathcal{G} will calculate the effect of the circuit C on the input (x, y, z) via gadgets simulating the Boolean functions *and*, *or* and *not*. To compensate the effects of the brittle comparator used in the analogue-digital converter, that returns an arbitrary value when $(\mathbf{p}[v_x], \mathbf{p}[v_y], \mathbf{p}[v_z])$ is near the boundary of its cubelet, these two steps will be repeated in a sufficiently large neighbourhood of (x, y, z) and the average of the displacements so obtained will be taken as displacement $\delta_{C(i,j,k)}$. Finally, we will “close the loop” forcing the vertices (v_x, v_y, v_z) to change their \mathbf{p} -values in the direction $\delta_{C(i,j,k)}$ if K_{ijk} is not a panchromatic cubelet: this way we will “move” following the (v_x, v_y, v_z) until an equilibrium, corresponding to a panchromatic cubelet, is found.

Before stating the main theorem we describe the gadgets simulating the Boolean functions. We call these gadgets *Boolean gadgets*, whereas the gadgets of subsection 2.2 will be called *arithmetical gadgets*.

Proposition 31 (Lemma 3 of [DGP], Lemma 13 of [DGP2]) *There are binary graphical games $\mathcal{G}_-, \mathcal{G}_\vee, \mathcal{G}_\wedge$, with players v_1, v_2, w (with v_2 not used for \mathcal{G}_-) such that at any ε -Nash equilibrium with $\varepsilon < 1/4$ of each game, $\mathbf{p}[w]$ is the result of applying the corresponding Boolean function to the inputs.*

PROOF. In \mathcal{G}_- the player w gets a payoff of 1 for disagreeing with v_1 , 0 otherwise:

		v_1	
		0	1
w	0	0	1
	1	1	0

In \mathcal{G}_\vee when w plays 0 it gets a payoff of $\frac{1}{2}$; when w plays 1 it gets 1 if at least one of v_1 and v_2 play 1 and gets 0 otherwise. In \mathcal{G}_\wedge when w plays 0 it gets a payoff of $\frac{1}{2}$; when w plays 1 it gets 1 if both v_1 and v_2 play 1 and gets 0 otherwise. \square

Theorem 32 (see Theorem 3 of [DGP], [DP], Theorem 11 of [DGP2]) *Both 3-NASH and 3-GRAPHICAL NASH are PPAD-complete.*

PROOF. Consider the problem 3-DIMENSIONAL BROUWER on a unit cube divided in 2^{3n} cubelets whose edges have length 2^{-n} , with a circuit C that calculates the displacements $\delta_0 = (-\alpha, -\alpha, -\alpha)$, $\delta_1 = (\alpha, 0, 0)$, $\delta_2 = (0, \alpha, 0)$, $\delta_3 = (0, 0, \alpha)$, where $\alpha = 2^{-2n}$. We now construct a binary graphical game \mathcal{G} of maximum degree 3 whose equilibria will correspond to panchromatic cubelets of the given instance of 3-DIMENSIONAL BROUWER.

Construction of the game \mathcal{G}

1. Let $m = 20$. Via arithmetical gadgets compute all the points of the form $(x + p \cdot \alpha, y + q \cdot \alpha, z + s \cdot \alpha)$, where $-m \leq p, q, s \leq m$. Let $M = (2m + 1)^3$, the number of points for which this passage is repeated.
2. For each one of these points
 - i. add vertices v_x, v_y, v_z ;
 - ii. for every $i \in \{1, \dots, n\}$ add vertices $v_{b_i(x)}, v_{b_i(y)}, v_{b_i(z)}$;
 - iii. for every $i \in \{1, \dots, n\}$ add vertices $v_{x_i}, v_{y_i}, v_{z_i}$;
 - iv. add a graphical game that using the arithmetical gadgets computes
$$\mathbf{x}_1 := \mathbf{x};$$
for $i = 1, \dots, n$

$$\{\mathbf{b}_i(\mathbf{x}) := \langle 2^{-i}, \mathbf{x}_i \rangle; \mathbf{x}_{i+1} := \mathbf{x}_i - \mathbf{b}_i(\mathbf{x}) \cdot 2^{-i};\}$$
where $a = b$ denotes $\mathbf{p}[a] = \mathbf{p}[b]$ and $\langle a, b \rangle$ denotes the brittle comparator of a and b ; similarly for y and z ;
 - v. add Boolean gadgets that calculate the circuit C with input $v_{b_i(x)}, v_{b_i(y)}, v_{b_i(z)}$ and output $\Delta x^+, \Delta x^-, \Delta y^+, \Delta y^-, \Delta z^+, \Delta z^-$ such that at most one of $\Delta x^+, \Delta x^-$ is 1, and similarly for y and z , so that the displacement calculated by C is $\alpha \cdot (\Delta x^+ - \Delta x^-, \Delta y^+ - \Delta y^-, \Delta z^+ - \Delta z^-)$.
3. Add a graphical game that calculates the averages
$$(\delta x^+, \delta y^+, \delta z^+) = \frac{\alpha}{M} \sum_{t=1}^M (\Delta x_t^+, \Delta y_t^+, \Delta z_t^+),$$

$$(\delta x^-, \delta y^-, \delta z^-) = \frac{\alpha}{M} \sum_{t=1}^M (\Delta x_t^-, \Delta y_t^-, \Delta z_t^-).$$
4. Use gadgets \mathcal{G}_+ , \mathcal{G}_- and $\mathcal{G}_=$ so that at the equilibrium we have $x = (x' + \delta x^+) - \delta x^-$, where x' is a copy of x created with $\mathcal{G}_=$; and similarly for y and z .

The \mathbf{p} -values of the vertices v_x, v_y, v_z are the coordinates of a point in the unit cube $[0, 1]^3$; the \mathbf{p} -values of $v_{b_i(x)}, v_{b_i(y)}, v_{b_i(z)}$ correspond to the i -th most significant bit of $\mathbf{p}[v_x], \mathbf{p}[v_y], \mathbf{p}[v_z]$; finally, the \mathbf{p} -values of $v_{x_i}, v_{y_i}, v_{z_i}$ correspond to the fractional numbers that results from subtracting from $\mathbf{p}[v_x], \mathbf{p}[v_y], \mathbf{p}[v_z]$ the fractional numbers corresponding to their $i - 1$ most significant bits. The point $(\mathbf{p}[v_x], \mathbf{p}[v_y], \mathbf{p}[v_z])$ thus

lies in the cubelet K_{ijk} for

$$(i, j, k) = \left(\sum_{l=1}^n 2^{n-l} \mathbf{p}[v_{b_l}(x)], \sum_{l=1}^n 2^{n-l} \mathbf{p}[v_{b_l}(y)], \sum_{l=1}^n 2^{n-l} \mathbf{p}[v_{b_l}(z)] \right). \quad (9)$$

The reduction is correct

We first see that the algorithm in step 2.iv decodes properly 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} , that is, as long as it is not near the boundary between two cubelets; this way we have that the \mathbf{p} -values of $v_{b_i}(x), v_{b_i}(y), v_{b_i}(z)$ and $v_{x_i}, v_{y_i}, v_{z_i}$ are as required above. Suppose $\varepsilon = \alpha^2 = 2^{-4n}$, so that ε is much smaller than the side of the cubelets.

Lemma 33 (*Lemma 2 of [DGP], Lemma 12 of [DGP2]*) For $m \leq n$, if

$$\sum_{i=1}^m b_i 2^{-i} + 3m\varepsilon < x < \sum_{i=1}^m b_i 2^{-i} + 2^{-m} - 3m\varepsilon \quad (10)$$

for some $b_1, \dots, b_m \in \{0, 1\}$, then at any ε -Nash equilibrium of \mathcal{G} we have that $\mathbf{p}[v_{b_j(x)}] = b_j$ and $\mathbf{p}[v_{x_{j+1}}] = x - \sum_{i=1}^j b_i 2^{-i} \pm 3j\varepsilon$ for all $j \leq m$.

PROOF. We prove the lemma by induction on j . Let $j = 1$; since $\sum_{i=2}^m 2^{-i} + 2^{-m} = 2^{-1}$, (10) implies that

$$\frac{b_1}{2} + 3\varepsilon < \mathbf{p}[v_x] < \frac{b_1}{2} + \frac{1}{2} - 3\varepsilon.$$

Since by the algorithm we have that $\mathbf{p}[v_{x_1}] = \mathbf{p}[v_x] \pm \varepsilon$, this implies that

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

now the algorithm yields that $\mathbf{p}[v_{b_1(x)}] = b_1$, where the error has been cancelled by an error of $\pm \varepsilon$ introduced by both the brittle comparator and the gadget \mathcal{G}_1 necessary to have the comparison.

The algorithm also gives that $\mathbf{p}[v_{x_1}] = \mathbf{p}[v_x] \pm \varepsilon$ and that $\mathbf{p}[v_{x_2}] = \mathbf{p}[v_{x_1}] - b_1 \cdot \frac{1}{2} \pm 2\varepsilon$, where the errors come from both the subtraction and from the multiplication by $\frac{1}{2}$: combining the two we have that $\mathbf{p}[v_{x_2}] = \mathbf{p}[v_x] - b_1 \cdot \frac{1}{2} \pm 3\varepsilon$, as required.

Suppose now that the claim holds up to $j-1 \leq m-1$. By induction hypothesis $\mathbf{p}[v_{x_j}] = \mathbf{p}[v_x] - \sum_{i=1}^{j-1} b_i 2^{-i} \pm 3(j-1)\varepsilon$, that together with (10) implies

$$\sum_{i=j}^m b_i 2^{-i} + 3(m - (j-1))\varepsilon < x < \sum_{i=j}^m b_i 2^{-i} + 2^{-m} - 3(m - (j-1))\varepsilon,$$

which in turn implies that

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

As before, by the algorithm this yields that $\mathbf{p}[v_{b_j(x)}] = b_j$. Finally, by the algorithm we have that $\mathbf{p}[v_{x_{j+1}}] = \mathbf{p}[v_{x_j}] - b_j(x) \cdot 2^{-j} \pm 3\varepsilon$, where the errors come from the subtraction, from the creation of 2^{-j} from the existing $2^{-(j-1)}$, and from the multiplication of $b_j(x)$ by 2^{-j} . By the induction hypothesis this becomes $\mathbf{p}[v_{x_{j+1}}] = \mathbf{p}[v_x] - \sum_{i=1}^{j-1} b_i(x) \cdot 2^{-i} - b_j \cdot 2^{-j} \pm 3(j-1)\varepsilon \pm 3\varepsilon = \mathbf{p}[v_x] - \sum_{i=1}^j b_i(x) \cdot 2^{-i} \pm 3j\varepsilon$. \square

It remains to see that a ε -Nash equilibrium of \mathcal{G} corresponds to a panchromatic cubelet of the instance of 3-DIMENSIONAL BROUWER.

Lemma 34 (Lemma 4 of [DGP], Lemma 14 of [DGP2]) *In any ε -Nash equilibrium of the game one of the vertices of the cubelets that contain $(\mathbf{p}[v_x], \mathbf{p}[v_y], \mathbf{p}[v_z])$ is panchromatic.*

PROOF. We start with a property of the increments $\delta_0, \delta_1, \delta_2, \delta_3$:

Lemma 35 (Lemma 5 of [DGP], Lemma 15 of [DGP2]) *If for some nonnegative integers k_0, k_1, k_2, k_3 all the 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. We prove the lemma by contradiction. Let $k_1 = 0$: then $k_0 < \frac{K}{5}$, or the x coordinate would be too large in absolute value. Thus, since $K = \sum_{i=0}^3 k_i$, one of k_2, k_3 is larger than $\frac{2K}{5}$: but this implies that the corresponding coordinate is larger than $\frac{\alpha K}{5}$. The case of k_2 and k_3 is dealt with similarly. Finally, if $k_0 = 0$, then one of the k_1, k_2, k_3 is at least $\frac{K}{3}$: but this implies that the corresponding coordinate is larger than $\frac{\alpha K}{5}$. \square

Let us denote $v_{\delta x^+}$ and $\{v_{\Delta x_t^+}\}_{t \in \{1, \dots, M\}}$ the vertices representing the values of δx^+ and $\{\Delta x_t^+\}_{t \in \{1, \dots, M\}}$ respectively. We have that

$$\mathbf{p}[v_{\delta x^+}] = \frac{\alpha}{M} \sum_{t=1}^M \mathbf{p}[\Delta x_t^+] \pm (2M-1)\varepsilon \quad (11)$$

where the error comes from the multiplication of each Δx_t^+ by $\frac{\alpha}{M}$ via a $\mathcal{G}_{\times \frac{\alpha}{M}}$ gadget and from the sum of all the results of these multiplications, considering that each operation introduces an error of $\pm \varepsilon$. Using a similar notation, we have that

$$\mathbf{p}[v_{\delta x^-}] = \frac{\alpha}{M} \sum_{t=1}^M \mathbf{p}[\Delta x_t^-] \pm (2M-1)\varepsilon, \quad (12)$$

and analogously for y and z .

We now distinguish two subcases for the location of $(x, y, z) = (\mathbf{p}[v_x], \mathbf{p}[v_y], \mathbf{p}[v_z])$: when (x, y, z) is at distance at least $(m+1)\alpha$ from every face of the unit cube $[0, 1]^3$ and when (x, y, z) is at distance at most $(m+1)\alpha$ from some face of the unit cube $[0, 1]^3$.

Let (x, y, z) be at distance at least $(m+1)\alpha$ from every face of the unit cube. Let $v_{x+p \cdot \alpha}$ be the player representing $x + p \cdot \alpha$, for $-m \leq p \leq m$. We first show that at most one of the $x + p \cdot \alpha$ values can be $3n\varepsilon$ -close to a multiple of 2^{-n} , thus that at most one of these values returns an arbitrary value in the brittle comparator in the algorithm of lemma 33. We compute the $x + p \cdot \alpha$ s from x via a gadget \mathcal{G}_α , a gadget $\mathcal{G}_{\times |p|}$ and a gadget \mathcal{G}_+ or \mathcal{G}_- depending on the sign of p : each of these gadgets introduces an error of $\pm \varepsilon$, thus $\mathbf{p}[v_{x+p \cdot \alpha}] = \mathbf{p}[v_x] + p \cdot \alpha \pm 3\varepsilon$, without truncations at 0 and 1 because we assumed $(m+1)\alpha < \mathbf{p}[v_x] < 1 - (m+1)\alpha$. Consequently, recalling that $\alpha = 2^{-2n}$ and $\varepsilon = 2^{-4n}$, we have that $\mathbf{p}[v_{x+m \cdot \alpha}] - \mathbf{p}[v_{x-m \cdot \alpha}] < 2m \cdot \alpha + 6\varepsilon \ll 2^{-n}$. This implies that from the $(2m+1)^3$ circuit evaluations all but at most $3(2m+1)^2$ are computed correctly, counting 3 coordinates and 2 evaluations per coordinate; that is, at least $K = (2m-2)(2m+1)^2$ evaluations are computed correctly.

Let $\mathcal{K} \subseteq \{-m, \dots, m\}^3$ be the set of the values (p, q, r) for which we have that the bit extraction from $(\mathbf{p}[v_{x+p \cdot \alpha}], \mathbf{p}[v_{y+q \cdot \alpha}], \mathbf{p}[v_{z+r \cdot \alpha}])$ is computed correctly, that is, from which the bit extraction returns a binary value. Define

$$S_{\mathcal{K}} = \frac{\alpha}{M} \sum_{t \in \mathcal{K}} (\mathbf{p}[\Delta x_t^+] - \mathbf{p}[\Delta x_t^-], \mathbf{p}[\Delta y_t^+] - \mathbf{p}[\Delta y_t^-], \mathbf{p}[\Delta z_t^+] - \mathbf{p}[\Delta z_t^-]) \quad (13)$$

$$S_{\mathcal{K}^c} = \frac{\alpha}{M} \sum_{t \notin \mathcal{K}} (\mathbf{p}[\Delta x_t^+] - \mathbf{p}[\Delta x_t^-], \mathbf{p}[\Delta y_t^+] - \mathbf{p}[\Delta y_t^-], \mathbf{p}[\Delta z_t^+] - \mathbf{p}[\Delta z_t^-]). \quad (14)$$

Now recall that in \mathcal{G} there are gadgets \mathcal{G}_+ , \mathcal{G}_- and $\mathcal{G}_=$ to enforce that in a Nash equilibrium we have $x = x' + \delta x^+ - \delta x^-$, where x' is a copy of x . Thus we have that $\mathbf{p}[v_{x'}] = \mathbf{p}[v_x] \pm \varepsilon$ and $\mathbf{p}[v_x] = \mathbf{p}[v_{x'}] + \mathbf{p}[v_{\delta x^+}] - \mathbf{p}[v_{\delta x^-}] \pm 2\varepsilon$; note that (11) and (12) give an upper bound of $\alpha + (2M - 1)\varepsilon$ so there are no truncations. This, in turn, implies that $\mathbf{p}[v_{\delta x^+}] - \mathbf{p}[v_{\delta x^-}] = \pm 3\varepsilon$ and similarly for y and z . Using (11), (12), (13) and (14) we obtain that

$$|S_{\mathcal{K}_x} + S_{\mathcal{K}_x^c}| = |\delta x^+ - \delta x^- \pm 2(2M - 1)\varepsilon \pm \varepsilon| \leq (4M + 1)\varepsilon,$$

and similarly for y and z . Moreover, since $|\mathcal{K}| \geq K$, it follows that $S_{\mathcal{K}_x^c} \leq \frac{\alpha}{M}(M - K)$ (at most $M - K$ summands, each one at most $\frac{\alpha}{M}$). Therefore,

$$|S_{\mathcal{K}_x}| \leq (4M + 1)\varepsilon + \frac{(M - K)\alpha}{M}, \quad (15)$$

and similarly for y and z .

Finally, recall that for all $(p, q, r) \in \mathcal{K}$ the bit extraction from $(\mathbf{p}[v_{x+p \cdot \alpha}], \mathbf{p}[v_{y+q \cdot \alpha}], \mathbf{p}[v_{z+r \cdot \alpha}])$ results in binary output. Therefore, we have that $|S_{\mathcal{K}}| = \frac{1}{M} \sum_{i=0}^3 k_i \delta_i$ for some nonnegative integers adding up to \mathcal{K} . From (15) we have that

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

By choosing $m = 20$ the bound becomes less than $\frac{\alpha K}{5}$, and so lemma 35 implies that among the results of the $|\mathcal{K}|$ computations all four increments $\delta_0, \delta_1, \delta_2, \delta_3$ appear. Since every point on which the circuit C is evaluated is within distance $3m\alpha + 9\varepsilon$ from (x, y, z) , this implies that among the corners of the cubelet(s) containing (x, y, z) there must be one panchromatic corner.

Now we deal with the second case: let (x, y, z) be at distance at most $(m + 1)\alpha$ from some face of the cube $[0, 1]^3$: we will show that there is no ε -Nash equilibria in the cubelet containing (x, y, z) . Consider for instance the case $\mathbf{p}[v_x] \leq (m + 1)\alpha$, $(m + 1)\alpha < \mathbf{p}[v_y] < 1 - (m + 1)\alpha$, $(m + 1)\alpha < \mathbf{p}[v_z] < 1 - (m + 1)\alpha$, since the other cases are analogous.

First we show that for all $p \in \{-m, \dots, m\}$ the bit extraction from $\mathbf{p}[v_{x+p \cdot \alpha}]$ results in binary outputs. For $p \geq 0$ we do not need to worry about the truncations at 0 and 1, so we have

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

On the other hand, for $p < 0$ there might be a truncation at 0 in the subtraction of $|p|\alpha$ from $\mathbf{p}[v_x]$; nevertheless, we have

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

Therefore, for all $p \in \{-m, \dots, m\}$ the algorithm of lemma 33 does not produce arbitrary values.

For directions y and z we have, as in the first case, that for at most one $q \in \{-m, \dots, m\}$ and at most one $r \in \{-m, \dots, m\}$ the bit extraction from $\mathbf{p}[v_{y+q\cdot\alpha}]$ and $\mathbf{p}[v_{z+r\cdot\alpha}]$ does not produce a binary value. Therefore from $M = (2m+1)^3$ bit extractions from $(\mathbf{p}[v_{x+p\cdot\alpha}], \mathbf{p}[v_{y+q\cdot\alpha}], \mathbf{p}[v_{z+r\cdot\alpha}])$ there are at most $2(2m+1)^2$ unsuccessful ones and at least $K' = (2m-1)(2m+1)^2$ successful ones; moreover, all the points inside cubelets of the form K_{0jk} return successful circuit evaluations.

From the definition of the problem BROWER we have that in the outputs of these evaluations is never present the displacement δ_0 ; on the other hand, since there are K' successful evaluations, we have that one of $\delta_1, \delta_2, \delta_3$ appears at least $\frac{K'}{3}$ times. Suppose that this is δ_1 , and denote by $v_{x'+\delta x}$ the player corresponding to $x' + \delta x$; then, in any ε -Nash equilibrium: $\mathbf{p}[v_{x'}] = \mathbf{p}[v_x] \pm \varepsilon$ (because of the $\mathcal{G}_=$ gadget), and we also have that

$$\frac{K'\alpha}{3M} - (2M-1)\varepsilon \leq \mathbf{p}[v_{\delta x^+}] \leq \alpha + (2M-1)\varepsilon : \quad (16)$$

to see that (16) is correct, consider the case when only the successful bit extractions in direction x result in output 1 and the case when all the bit extractions result in output 1 for the error see (11). Furthermore,

$$\mathbf{p}[v_{x'+\delta x^+}] \geq \min(1, \mathbf{p}[v_{x'}] + \mathbf{p}[v_{\delta x^+}]) - \varepsilon \geq \mathbf{p}[v_{x'}] + \mathbf{p}[v_{\delta x^+}] - \varepsilon, \quad (17)$$

where the ε comes from the \mathcal{G}_+ gadget. Finally, we have that

$$\mathbf{p}[v_{\delta x^-}] \leq \frac{(M-K')\alpha}{M} + (2M-1)\varepsilon : \quad (18)$$

to see that this is correct consider the case when all the unsuccessful bit extractions return 1 and all the successful ones return 0; for the error see (12). The inequalities (16) to (18) together with the fact that for our choice of parameters M and K' we have that $\frac{K'\alpha}{3M} > \frac{(M-K')\alpha}{M} + (4M-1)\varepsilon$, yield that

$$\begin{aligned} \mathbf{p}[v_{x'+\delta x^+}] &\geq \mathbf{p}[v_{\delta x^+}] - \varepsilon \\ &\geq \frac{K'\alpha}{3M} - 2M\varepsilon > \\ &> \frac{(M-K')\alpha}{M} + (2M-1)\varepsilon \\ &\geq \mathbf{p}[v_{\delta x^-}] \end{aligned}$$

which in turn yields

$$\begin{aligned} \mathbf{p}[v_x] &\geq \max(0, \mathbf{p}[v_{x'+\delta x^+}] - \mathbf{p}[v_{\delta x^-}]) - \varepsilon \\ &\geq \mathbf{p}[v_{x'+\delta x^+}] - \mathbf{p}[v_{\delta x^-}] - \varepsilon \geq \\ &\geq \mathbf{p}[v_{x'}] + \mathbf{p}[v_{\delta x^+}] - \mathbf{p}[v_{\delta x^-}] - 2\varepsilon \\ &\geq \mathbf{p}[v_x] + \mathbf{p}[v_{\delta x^+}] - \mathbf{p}[v_{\delta x^-}] - 3\varepsilon \end{aligned}$$

so that $\mathbf{p}[v_{\delta x^-}] \geq \mathbf{p}[v_{\delta x^+}] - 3\varepsilon$. But this implies that $\frac{(M-K')\alpha}{M} + (4M+1)\varepsilon \geq \frac{K'\alpha}{3M}$, which is impossible given our choice of M and K' . Hence there are no ε -Nash equilibria within distance $(m+1)\varepsilon$ from the given face of the cubelet and, analogously, from any face of the cubelet. \square

Given a ε -Nash equilibrium of \mathcal{G} it is thus enough to read the first n binary digits of $\mathbf{p}[v_x], \mathbf{p}[v_y], \mathbf{p}[v_z]$ to obtain with (9) a panchromatic cubelet K_{ijk} that constitutes a solution to the corresponding instance of BROUWER. \square

3.3 2-NASH

Finally, we prove that 2-NASH is PPAD-complete following the proof given in [DGP2]. Recall that $N(v)$ is the set of the predecessors of v in the affects-graph; we start with two definitions that are central to our proof.

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 every $v \in V$ and $s_v \in S_v$ there exist rational numbers $\{\alpha_{u,s_u}^{s_v}\}_{u \in N(v), s_u \in S_u} \in [0, 1]$ such that the expected payoff to v for playing s_v is

$$\sum_{u \in N(v), s_u \in S_u} \alpha_{u,s_u}^{s_v} \mathbf{p}[u \text{ plays } s_u].$$

Furthermore, let d -ADDITIVE GRAPHICAL NASH to be the problem d -GRAPHICAL NASH restricted to bipartite graphical games with additive utility functions.

We will now present a reduction from 3-DIMENSIONAL BROUWER to 3-ADDITIVE GRAPHICAL NASH and from 3-ADDITIVE GRAPHICAL NASH to 2-NASH. Since the graph of a game describing 3-DIMENSIONAL BROUWER is generally not a collection of isolated points and 2-cliques, we will no longer be able to use corollary 6; on the other hand, we will have a bipartite graph and from that bipartition we will be able to construct a 2-player normal-form game.

Theorem 36 (Theorem 12 of [DGP2]) *2-NASH is PPAD-complete.*

PROOF. As mentioned before, the proof consists of two reductions: from 3-DIMENSIONAL BROUWER to 3-ADDITIVE GRAPHICAL NASH, and from that to 2-NASH. We start by making the gadgets used in the reduction from 3-DIMENSIONAL BROUWER bipartite and with additive utility function, so that we can obtain the first reduction by simply following the one of theorem 32 with the new gadgets. First of all, notice that we will not need the gadget \mathcal{G}_* , since only multiplications by a constant are used. In gadgets $\mathcal{G}_\alpha, \mathcal{G}_{\times\alpha}, \mathcal{G}_=, \mathcal{G}_+, \mathcal{G}_-, \mathcal{G}_<$ and \mathcal{G}_\neg the non-input vertices have the additive utility functions property: for $\mathcal{G}_\alpha, \mathcal{G}_{\times\alpha}, \mathcal{G}_=, \mathcal{G}_+, \mathcal{G}_-$ it is immediate to check; in $\mathcal{G}_<$ the payoff of the output vertex is $\mathbf{p}[v_1]$ for playing 0 and $\mathbf{p}[v_2]$ for playing 1; and in \mathcal{G}_\neg the payoff of the output vertex is $\mathbf{p}[v_1]$ for playing 0 and $\mathbf{p}[v_1 \text{ plays } 0]$ for playing 1. It remains to construct $\mathcal{G}_\vee, \mathcal{G}_\wedge$ with additive utility functions: this is done in

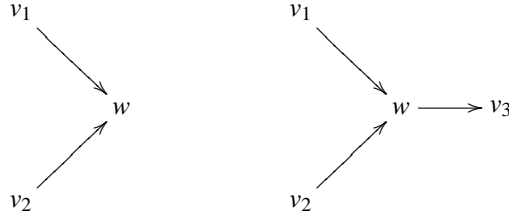
Lemma 37 (Lemma 18 of [DGP2]) *There are binary graphical games $\mathcal{G}_\vee, \mathcal{G}_\wedge$, with players v_1, v_2, w such that w 's payoff satisfies the additive utility functions property and at any ε -Nash equilibrium with $\varepsilon < 1/4$ of each game, $\mathbf{p}[w]$ is the result of applying the corresponding Boolean function to the inputs.*

PROOF. In \mathcal{G}_V when w plays 0 it gets a payoff of $\frac{1}{4}$; when w plays 1 it gets $\frac{1}{2}\mathbf{p}[v_1] + \frac{1}{2}\mathbf{p}[v_2]$. The payoff to w for playing 1 are thus

	v_2	0	1
v_1	0	0	$\frac{1}{2}$
	1	$\frac{1}{2}$	1

In \mathcal{G}_\wedge when w plays 0 it gets a payoff of $\frac{3}{4}$; when w plays 1 it gets $\frac{1}{2}\mathbf{p}[v_1] + \frac{1}{2}\mathbf{p}[v_2]$. \square

We now make sure that the graphical game describing 3-DIMENSIONAL BROUWER is bipartite: to do so we only have to modify the Boolean gadgets and $\mathcal{G}_<$ by adding a second output vertex that “copies” the behaviour of the first output vertex. For instance, $\mathcal{G}_<$ will change as in figure,



with the payoff of v_3 set as 1 if both w and v_3 play 0 or 1 and set as 0 otherwise, that is, to $\mathbf{p}[w]$ for playing 1 and to $\mathbf{p}[w \text{ plays } 0]$ for playing 0. To have the reduction from 3-DIMENSIONAL BROUWER to 3-ADDITIVE GRAPHICAL NASH now we only need to apply the construction of theorem 32 with the new gadgets; the bipartition of the graph can be seen by giving colour 1 to all the input and output vertices of the $\mathcal{G}_{\times\alpha}, \mathcal{G}_=, \mathcal{G}_+, \mathcal{G}_-, \mathcal{G}_\wedge, \mathcal{G}_V, \mathcal{G}_\neg, \mathcal{G}_<$ gadgets and to the output vertices of the \mathcal{G}_α gadgets then giving colour 2 to all the other vertices.

Finally, we reduce 3-ADDITIVE GRAPHICAL NASH to 2-NASH. Let \mathcal{GG} be a graphical game of maximum degree 3 with additive utility functions and underlying graph $G = (V = V_1 \sqcup V_2, E)$; as seen in the proofs of theorems 5 and 15, we can assume without loss of generality that all utilities of \mathcal{GG} lie in $[0, 1]$. In time polynomial in $|\mathcal{GG}| + \log(\frac{1}{\varepsilon})$ we will specify a 2-player normal-form game \mathcal{G} and an accuracy parameter ε' such that given an ε' -Nash equilibrium of \mathcal{G} we can recover in polynomial time an ε -Nash equilibrium of \mathcal{GG} .

The construction is similar to the ones of section 2.1, exploiting the bipartite structure of the graph of \mathcal{GG} . Start by adding isolated vertices to V_1 and V_2 so that $|V_1| = |V_2| = n$, if necessary, and let t be the cardinality of the strategy sets of the vertices in V , also adding dummy strategies with payoff 0 if necessary. Define a colouring $c : V \rightarrow \{1, 2\}$ such that for all $v_p \in V_p$ with $p \in \{1, 2\}$ we have $c(v_p) = p$. The strategy sets of \mathcal{G} are defined as $S_p = \{(v, s_v) | c(v) = p, s_v \in S_v\}$ with $p \in \{1, 2\}$; note that $|S_p| = \frac{tn}{2}$. We now define the payoffs of \mathcal{G} :

- i. initially the payoffs u^p are all zero;
- ii. for $v \in V$ such that $c(v) = p$, if $s \in S_1 \times S_2$ contains (v, s_v) for some $s_v \in S_v$ and (u, s_u) for some $u \in N(v)$ and $s_u \in S_u$ we set $u^p(s) = \alpha_{u, s_u}^{s_v}$;

iii. let $M = \frac{6tn}{\varepsilon}$;

iv. if in $s \in S_1 \times S_2$ player 1 plays (v_i^1, a) for some $a \in S_{v_i^1}$ and player 2 plays (v_i^2, a') for some $a' \in S_{v_i^2}$, add M to $u^1(s)$ and subtract M from $u^2(s)$.

By letting $\varepsilon' = \frac{\varepsilon}{n}$ we construct \mathcal{G} in time polynomial in $|\mathcal{G}\mathcal{G}| + \log(\frac{1}{\varepsilon})$, as seen in the proof of theorem 5.

It remains to show that given an ε' -Nash equilibrium $\{x_{(v,a)}^p\}_{p,v,a}$ of \mathcal{G} we can recover a ε -Nash equilibrium of $\mathcal{G}\mathcal{G}$ $\{x_a^v\}_{v,a}$: this is done by setting

$$x_a^v = \frac{x_{(v,a)}^{c(v)}}{\sum_{j \in S_v} x_{(v,j)}^{c(v)}}. \quad (19)$$

Note that the calculation takes polynomial time. As in the proof of lemma 4 it can be shown that in any ε' -Nash equilibrium of \mathcal{G}

$$\mathbf{p}[c(v) \text{ plays } v] \in [\frac{2}{n} - \frac{1}{M}, \frac{2}{n} + \frac{1}{M}]. \quad (20)$$

Without loss of generality, consider $p = 1$ and $v = v_i^1$. In any ε' -Nash equilibrium of \mathcal{G} we have, by definition, that for every strategy $s_v, s'_v \in S_v$

$$\mathbb{E}[\text{payoff to } p \text{ for playing } (v, s_v)] > \mathbb{E}[\text{payoff to } p \text{ for playing } (v, s'_v)] + \varepsilon' \Rightarrow x_{(v,s'_v)}^p = 0 \quad (21)$$

But

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

and similarly for s'_v , which together with (21) implies

$$\sum_{u \in N(v), s_u \in S_u} \alpha_{u,s_u}^{s_v} x_{(u,s_u)}^{c(u)} > \sum_{u \in N(v), s_u \in S_u} \alpha_{u,s_u}^{s'_v} x_{(u,s_u)}^{c(u)} + \varepsilon' \Rightarrow x_{(v,s'_v)}^p = 0 \quad (22)$$

Furthermore,

$$\begin{aligned} |x_a^v - \frac{x_{(v,a)}^{c(v)}}{\frac{2}{n}}| &= |\frac{x_{(v,a)}^{c(v)}}{\mathbf{p}[c(v) \text{ plays } v]} - \frac{x_{(v,a)}^{c(v)}}{\frac{2}{n}}| \\ &= |\frac{(x_{(v,a)}^{c(v)})(\mathbf{p}[c(v) \text{ plays } v] - \frac{2}{n})}{(\frac{2}{n})(\mathbf{p}[c(v) \text{ plays } v])}| \\ &\leq \frac{n}{2M} \end{aligned} \quad (\text{by (20) and (19)})$$

that applied to the summands in (22) yields

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

But we also have that $\frac{n}{2} \varepsilon' + \frac{|N(v)|tn}{M} = \frac{\varepsilon}{2} + \frac{|N(v)|\varepsilon}{6} \leq \varepsilon$ since $|N(v)|$ cannot be greater than the maximum degree of the graph, that is 3. Therefore, $\{x_a^v\}_{v,a}$ satisfying (19) is an ε -Nash equilibrium of $\mathcal{G}\mathcal{G}$. \square

Conclusion

Which consequences do these results yield? First of all, it seems that the connection between finding a Nash equilibrium and finding a Brouwer fixed point is very strong, not just limited to the proof of Nash's theorem in [N51]. This might indicate that finding a Nash or ε -Nash equilibrium is a task that is not easy to accomplish efficiently: Hirsch, Papadimitriou and Vavasis proved in [HPV] that any algorithm for finding a Brouwer fixed point that treats the Brouwer function as a black box (a class of algorithm that includes all general purpose algorithms) has exponential lower bounds, that is, it must perform a number of function evaluations exponential in both the number of digits of the accuracy and the dimension. This result recalls the one of Savani and von Stengel ([SvS]) that present a class of 2-player games for which the Lemke-Howson algorithm takes exponential time in the dimension of the game, even in the best case. On the other hand, finding a Nash equilibrium in a zero-sum 2-player game is a problem in P, since the maxmin theorem proved by von Neumann in [vN28] makes it a case of Linear Programming duality, thus solvable in polynomial time with the ellipsoid method of Khachiyan ([K79]): it might be premature to abandon the search for a way to solve a game in subexponential time. It might be true, that is, what has been argued by Papadimitriou in [P07], that the result on the PPAD-completeness of NASH “only shows that it is hard to find a Nash equilibrium in some far-fetched, artificial games that happen to encode Brouwer functions”. As for the result of Savani and von Stengel, it settles the case of finding a Nash equilibrium with the Lemke-Howson algorithm, but as pointed out by the same Savani and von Stengel “the problem of finding an ε -Nash equilibrium is a different problem” from finding exact equilibria. It is also important to note that the equilibrium concepts are not limited to the Nash and ε -Nash equilibria. The main lines of work left open are thus finding algorithm to compute ε -Nash equilibria for ε not exponentially small (but still as small as possible), to restrict the scope to some class of games, generally succinctly representable to avoid inputs too large to be feasible, and to look for other equilibrium concepts.

Regarding ε -Nash equilibria, Chen, Deng and Teng ([CDT]) have shown that for ε inverse polynomial in the number of strategies the problem is still PPAD, but Daskalakis, Mehta and Papadimitriou ([DMP2]) have found a linear time algorithm for the case of two players with $\varepsilon = \frac{1}{2}$. Lipton, Markakis and Mehta ([LMM]) have proved that it is possible to find an ε -Nash equilibrium in an arbitrary game in time $n^{\frac{\log n}{\varepsilon^2}}$, where n is the number of strategies, as long as the number of player is independent of the number of strategies. Finally, Daskalakis, Mehta and Papadimitriou ([DMP]) have shown that no approximation better than $\frac{1}{2}$ is possible unless support larger than the number of strategies per player are considered; the same article gives a polynomial algorithm for computing ε -Nash equilibria with $\varepsilon = .38$, considering arbitrarily large supports.

The work of Elkind, Goldberg and Goldberg ([EGG]) focuses on graphical games, and gives a polynomial time algorithm to compute Nash equilibria in graphical games with maximum degree 2 and 2 strategies per player, along with two algorithms that consider the case in which the graph is a path: one finds an equilibrium in time quadratic in length of the path, the other finds all equilibria in time cubic in the length of the path. Papadimitriou and Roughgarden ([PR]), on the other hand, consider symmet-

ric games (games in which all players are identical and indistinguishable) with r players and $O(\frac{\log r}{\log \log r})$ strategies, for which they give a polynomial time algorithm to find a Nash equilibrium. Daskalakis and Papadimitriou ([DP2]) give a polynomial-time algorithm to compute pure Nash equilibria for graphical games with treewidth logarithmic in the number of players, bounded neighbourhood size and bounded cardinality strategy sets, whose extension to approximate mixed-strategy Nash equilibria runs in time polynomial in the size of the game and exponential in the size of the approximation. Another complexity result in this line of work is theorem 4 of [EGG], that proves that the problem of finding a Nash equilibrium for a game in which the underlying graph has maximum degree 3 is PPAD-complete; this result has been extended by Daskalakis, Fabrikant and Papadimitriou, who in [DFP] have given a condition for the Nash equilibrium problem to be reducible to the 2-player case and thus in PPAD, a condition satisfied by all major known families of succinct games. Both the work of Elkind, Goldberg and Goldberg and the one of Daskalakis, Fabrikant and Papadimitriou start from the results whose proof is the main subject of this dissertation.

Other authors, finally, look for correlated equilibria, a generalisation of Nash equilibria introduced by Aumann ([A74]) and that, in the words of Papadimitriou ([P07]) may be “a computationally benign generalization of the intractable Nash equilibrium”. Papadimitriou ([P05]) shows that for most succinctly representable games, including graphical and symmetric games, correlated equilibria can be found in polynomial time; whereas Papadimitriou and Roughgarden ([PR]) describe the set of all correlated equilibria of a symmetric game with a linear system that has size polynomial in the compact representation of the game. The article by Papadimitriou and Roughgarden also shows that the problem of optimizing a linear function over the correlated equilibria of a game is solvable in polynomial time for symmetric and graphical games, but that for general graphical games there is no algorithm that optimises over the set of correlated equilibria in time polynomial in the size of the representation.

Acknowledgements

Professor Bernhard von Stengel has been an incredible supervisor, waiting for me during difficult times and providing assistance even for the most trivial problems. Professor Christos Papadimitriou explained me the concept of semantic and syntactic complexity class, an invaluable help to understand his works. Professor Paul Goldberg pointed me to the new version of his article on the complexity of computing a Nash equilibrium: without him this dissertation would have been far less thorough.

A "thank you" is also due to Simon Jolly of the Mathematics department for answering all my questions and to Jane Sedgwick of the disability office for her time and suggestions.

Finally, uncountable thanks to my husband Emmanuele who has supported me in every possible and some impossible way.

Bibliography

- [A74] R. J. Aumann, (1974). Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, 1, pp. 67–96.
- [AM] N. Alon, B. Mohar, (2002). The chromatic number of graph powers. *Combinatorics, Probability and Computing*, 11, pp. 1–10.
- [B85] K. C. Border, (1985). *Fixed Point Theorems with Applications to Economics and Game Theory*, Cambridge University Press, Cambridge.
- [C67] D. I. A. Cohen, (1967). On the Sperner lemma, *Journal of Combinatorial Theory*, 2, pp. 585–587.
- [CD3] X. Chen and X. Deng, (2005). 3-NASH is PPAD-complete. *Electronic Colloquium on Computational Complexity*, 134.
- [CD] X. Chen and X. Deng, (2005). Settling the complexity of 2-player Nash-equilibrium. *Electronic Colloquium on Computational Complexity*, 134.
- [CDT] X. Chen, X. Deng and S. Teng, (2006). Computing Nash equilibria: approximation and smoothed complexity. *Proceedings of 47th Annual IEEE Symposium on Foundations of Computer Science*.
- [CR] P. Clote, J. B. Remmel, (1995). *Feasible Mathematics II*, Birkhäuser, p. 246.
- [DFP] C. Daskalakis, A. Fabrikant and C. H. Papadimitriou, (2006). The game world is flat: the complexity of Nash equilibria in succinct games. *Proceedings of International Colloquium on Automata, Languages and Programming*.
- [DGP] C. Daskalakis, P. W. Goldberg and C. H. Papadimitriou, (2006). The complexity of computing a Nash equilibrium. *Annual ACM Symposium on Theory of Computing 2006*, pp. 71–78.
- [DGP2] C. Daskalakis, P. W. Goldberg and C. H. Papadimitriou, (2008). The complexity of computing a Nash equilibrium. Draft of the journal version,
<http://www.csc.liv.ac.uk/~pwg/publications.html>.
- [DMP] C. Daskalakis, A. Mehta and C. H. Papadimitriou, (2007). Progress in approximate Nash equilibria, *Proceedings of the ACM Conference on Electronic commerce*.
- [DMP2] C. Daskalakis, A. Mehta and C. H. Papadimitriou, (2006). A note on approximate equilibria. *Proceedings of 2nd international Workshop on Internet and Network Economics*.
- [DP] C. Daskalakis and C. H. Papadimitriou, (2005). Three-player games are hard. *Electronic Colloquium on Computational Complexity*, 139.
- [DP2] C. Daskalakis and C. H. Papadimitriou, (2006). Computing pure Nash equilibria in graphical games via Markov random fields. *Proceedings of the ACM conference on Electronic commerce*.
- [D05] R. Diestel, (2005). *Graph Theory*, Springer Verlag, p. 114.
- [EGG] E. Elkind, L. A. Goldberg, P. W. Goldberg, (2006). Nash equilibria in graphical games on trees revisited. *Proceedings of the ACM Conference on Electronic Commerce*.

- [G07] P. W. Goldberg, (2007). Introduction to PPAD. <http://www.csc.liv.ac.uk/~pwg/>.
- [GP] P. W. Goldberg and C. H. Papadimitriou, (2006). Reducibility among equilibrium problems. *Annual ACM Symposium on Theory of Computing 2006*, pp. 62–70.
- [HPV] M. Hirsch, C. H. Papadimitriou and S. Vavasis, (1989). Exponential lower bounds for finding Brouwer fixpoints. *Journal of Complexity* 5, pp. 379–416.
- [K79] L. G. Khachiyan, (1979). A polynomial algorithm in linear programming. *Soviet Mathematics Doklady*, 20(1), pp. 191–194.
- [KLS] M. Kearns, M. L. Littman and S. Singh, (2001). Graphical models for game theory. *Proceedings of the 17th Annual Conference on Uncertainty in Artificial Intelligence*.
- [LH] C. E. Lemke and J. T. Howson, Jr. (1964). Equilibrium points of bimatrix games. *Journal of the Society for Industrial and Applied Mathematics* 12, pp. 413–423.
- [LMM] L. Lipton, E. Markakis and A. Mehta, (2003). Playing large games using simple strategies. *Proceedings of the ACM Conference on Electronic Commerce*.
- [MP] N. Megiddo and C. H. Papadimitriou, (1991). On total functions, existence theorems and computational complexity. *Theoretical Computer Science*, 81, pp. 317–324.
- [N51] J. Nash, (1951). Noncooperative games. *Annals of Mathematics*, 54, pp. 289–295.
- [P94] C. H. Papadimitriou, (1994). On the complexity of the parity argument and other inefficient proofs of existence. *Journal of computer and system sciences* 48, 3, pp. 498–532.
- [P05] C. H. Papadimitriou, (2005). Computing correlated equilibria in multiplayer games. *Annual ACM Symposium on Theory of Computing 2005*.
- [P07] C. H. Papadimitriou, (2007). The complexity of finding Nash equilibria. Chapter 2 of *Algorithmic Game Theory*, edited by N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, Cambridge University Press, Cambridge, pp. 29–51.
- [PR] C. H. Papadimitriou and T. Roughgarden, (2005). Computing equilibria in multi-player games. *Proceedings of the 16th annual ACM-SIAM symposium on Discrete algorithms*.
- [R71] J. Rosenmüller, (1971). On a generalization of the Lemke-Howson algorithm to noncooperative n -person games, *SIAM Journal of Applied Mathematics*, 21, pp. 73–79.
- [SvS] R. Savani and B. von Stengel, (2004). Exponentially many steps for finding a Nash equilibrium in a bimatrix game. *Proceedings of 45th Annual IEEE Symposium on Foundations of Computer Science*, pp. 258–267.
- [vN28] J. von Neumann, (1928). Zur Theorie der Gesellschaftspiele, *Mathematische Annalen*, 100, pp.295–320.
- [vS07] B. von Stengel, (2007). Equilibrium computation for two-player games. Chapter 3 of *Algorithmic Game Theory*, edited by N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, Cambridge University Press, Cambridge, pp. 53–77.