Tight Inapproximability for Graphical Games

Argyrios Deligkas
Royal Holloway, United Kingdom
argyrios.deligkas@rhul.ac.uk

John Fearnley
University of Liverpool, United Kingdom
john.fearnley@liverpool.ac.uk

Alexandros Hollender
University of Oxford, United Kingdom
alexandros.hollender@cs.ox.ac.uk

Themistoklis Melissourgos
University of Essex, United Kingdom
themistoklis.melissourgos@essex.ac.uk

Abstract

We provide a complete characterization for the computational complexity of finding approximate equilibria in two-action graphical games. We consider the two most well-studied approximation notions: ε-Nash equilibria (ε-NE) and ε-well-supported Nash equilibria (ε-WSNE), where $\varepsilon \in [0, 1]$. We prove that computing an ε-NE is PPAD-complete for any constant $\varepsilon < 1/2$, while a very simple algorithm (namely, letting all players mix uniformly between their two actions) yields a 1/2-NE. On the other hand, we show that computing an ε-WSNE is PPAD-complete for any constant $\varepsilon < 1$, while a 1-WSNE is trivial to achieve, because any strategy profile is a 1-WSNE. All of our lower bounds immediately also apply to graphical games with more than two actions per player.

1 Introduction

Graphical games were introduced more than twenty years ago by Kearns, Littman, and Singh [KLS01] as a succinct model of a multi-player game. These games have found a wide variety of applications. On the theoretical side, they have served as a fundamental tool for showing seminal PPAD-completeness results in algorithmic game theory [DGP09, CDT09]. Practically, graphical games have been used as a foundation for the game theoretic analysis of networks [GGJ+10, JZ15], social networks, and multi-agent systems [Kea07, Jac11].

A graphical game is specified by a directed graph with n vertices. Each vertex represents a player, and each player has m distinct actions. The edges of the graph specify the interactions between the players: the payoff to player i is determined entirely by the actions chosen by player i and the in-neighbours of player i. Formally, the payoffs for a player are given by a payoff tensor, which maps the actions chosen by that player and their in-neighbours to a payoff in $[0, 1]$.

Graphical games are more succinct than standard normal form games when the maximum in-degree d is constant. An n-player m-action game requires $n \cdot m^n$ payoffs to be written down, which becomes infeasibly large as n grows. On the other hand, each tensor in a graphical game has m^{d+1} payoff entries, giving $n \cdot m^{d+1}$ payoffs in total, which provides much more reasonable scaling as n grows when d is constant.

The complexity of finding equilibria. Unfortunately it is known that finding a Nash equilibrium in a graphical game is a PPAD-hard problem [DGP09] and thus considered to be intractable. This has left open the question of finding *approximate* Nash equilibria, and two notions of approximate equilibrium have been studied in the literature. An *ε-Nash equilibrium* (ε-NE) requires that no player can improve their payoff by more than ε by unilaterally changing their strategy, while an *ε-well-supported Nash equilibrium* (ε-WSNE) requires that all players only place positive probability on actions that are ε-best responses. Every ε-WSNE is also an ε-NE, but the reverse is not true.

In this paper we make the standard assumption that all payoffs lie in the range $[0, 1]$, which then gives us a scale on which we can measure the additive approximation factor ε. A 0-NE or 0-WSNE is an exact Nash equilibrium, while a 1-NE or 1-WSNE can be trivially obtained, since the requirements will be satisfied no matter what strategies the players use.

For many years, the best known lower bounds for approximate equilibria in graphical games were given by Rubinstein [Rub18], who proved that there is some unspecified small constant ε for which finding an ε-NE is PPAD-complete, and there is a different but still unknown small constant ε' for which finding an ε'-WSNE is PPAD-complete. In fact, Rubinstein's result applies to games that are simultaneously graphical games of constant degree and also *polymatrix games*, namely in which each edge represents a two-player game and a player's payoff is the sum of payoffs from these games against her in-neighbours.

This was recently improved by a result of Deligkas et al. [DFHM22]. They showed that it is PPAD-complete to find a $1/48$-NE of a two-action polymatrix game, and it is PPAD-complete to find an ε-WSNE of a two-action polymatrix game for all $\varepsilon < 1/3$ (the latter result being tight). Since these hardness results hold even for constant-degree polymatrix games, they also apply to graphical games.[1]

On the other hand, only trivial upper bounds are known for approximate equilibria in graphical games, even when the players only have two actions. For ε-WSNE the upper bound is 1, since any strategy profile is a 1-WSNE. For ε-NE, the upper bound is $1/2$ in two-action games and is achieved when all players uniformly mix over their two actions; the upper bound simply follows from the fact that every player plays their best response with probability at least 0.5 and that the maximum payoff is bounded by 1.

Our Contribution. In this paper we show that the aforementioned trivial upper bounds are in fact the *best possible*, by providing matching lower bounds for finding approximate equilibria in graphical games.

For the problem of finding an ε-WSNE, we show that it is PPAD-complete to find an ε-WSNE in a two-action graphical game for every constant $\varepsilon < 1$. Since finding a 1-WSNE is trivial, we obtain a striking characterization for constant ε: no polynomial-time algorithm can find a non-trivial WSNE of a graphical game unless PPAD $=$ P.

In fact, we present a more fine-grained analysis that provides a complete dichotomy of the complexity of finding a WSNE in a two-action graphical game of maximum in-degree d:

- a $\left(1 - \frac{2}{2^d+1}\right)$-WSNE can be found in polynomial time;
- for any $\varepsilon < 1 - \frac{2}{2^d+1}$, it is PPAD-complete to compute a ε-WSNE.

Thus, for any constant $\varepsilon < 1$ there exists a sufficiently large constant in-degree d, such that the problem becomes intractable.

For ε-NE we show that it is PPAD-complete to find an ε-NE of a two-action graphical game for any constant $\varepsilon < 0.5$; this complements the straightforward algorithm for finding a 0.5-NE in a two-action graphical game. We note that our lower bounds, both for ε-WSNE and ε-NE, also hold for graphical games with more than two actions.[2]

All of our lower bounds are shown via reductions from the PURE-CIRCUIT problem that was recently introduced by Deligkas et al. [DFHM22]. In that paper, PURE-CIRCUIT was used to show the aforementioned lower bounds for polymatrix games. We show that PURE-CIRCUIT can likewise be used to show stronger and tight lower bounds for graphical games.

[1]A polymatrix game of constant degree can be turned into its graphical game representation in polynomial time.

[2]We can simply add additional "dummy" actions that are just copies of the original two actions.

Further Related Work. The class PPAD was defined by Papadimitriou [Pap94]. Many years later, Daskalakis, Goldberg, and Papadimitriou [DGP09] proved that finding an ε-NE in graphical games and 3-player normal form games is PPAD-complete for an exponentially small ε. These results were further extended to polynomially small ε for 2-player games and two-action polymatrix games with bipartite underlying graph by Chen, Deng, and Teng [CDT09].

On the positive side, Elkind, Goldberg, and Goldberg [EGG06] derived a polynomial-time algorithm on two-action graphical games on paths and cycles, and Ortiz and Irfan [OI17] derived an approximation scheme for constant-action graphical games on trees. For polymatrix games, Barman, Ligett, and Piliouras [BLP15] derived a quasi-PTAS on trees, and Deligkas, Fearnley, and Savani [DFS17] derived a quasi-PTAS for constant-action games on bounded treewidth graphs. These results where complemented by the same authors [DFS20] who showed that finding an *exact* NE is PPAD-complete for polymatrix games on trees when every player has 20 actions.

2 Preliminaries

For every natural number k, let Δ^k denote the k-dimensional simplex, i.e., $\Delta^k := \{x \in \mathbb{R}^{k+1} : x \geq 0, \sum_{i=1}^{k+1} x_i = 1\}$, and let $[k] := \{1, 2, \ldots, k\}$.

2.1 Graphical Games

An n-player m-action graphical game is defined by a directed graph $G = (V, E)$, where $|V| = n$, each node of G corresponds to a player, and the maximum number of actions per player is m. We define the in-neigbourhood of node i to be $N^-(i) := \{j \in V : (j, i) \in E\}$, and similarly its out-neighbourhood to be $N^+(i) := \{j \in V : (i, j) \in E\}$. We also define the neighbourhood of i as $N(i) := N^-(i) \cup N^+(i) \cup \{i\}$. We include i in its own neighbourhood for notational convenience.

In a graphical game, each player i participates in a normal form game G_i whose player set is $N(i)$, but she affects only the payoffs of her out-neighbours. Player i has m_i *actions*, or *pure strategies*, and her payoffs are represented by a function $R_i : [m_i] \times \prod_{j \in N^-(i)} [m_j] \mapsto [0, 1]$ which will be referred to as the *payoff tensor* of i. If the codomain of R_i is $\{0, 1\}$ for all $i \in V$, we have a *win-lose* graphical game.

A *mixed strategy* s_i for player i specifies a probability distribution over player i's actions. Thus, the set of mixed strategies for player i corresponds to the $(m_i - 1)$-dimensional simplex $\Delta^{m_i - 1}$. The *support* of a mixed strategy $s_i = (s_i(1), s_i(2), \ldots, s_i(m_i)) \in \Delta^{m_i-1}$ is given by $\mathsf{supp}(s_i) = \{j \in [m_i] : s_i(j) > 0\}$. In other words, it is the set of pure strategies that are played with non-zero probability in strategy s_i.

An action profile $\mathbf{a}(H) := (a_i)_{i \in H}$ over a player set H is a tuple of actions, one for each player in H, and so the set of these action profiles is given by $A(H) = \prod_{i \in H} [m_i]$. Similarly, a *strategy profile* $\mathbf{s}(H)$ over the same set is a tuple of mixed strategies, and so the set of these strategy profiles is given by $\prod_{i \in H} \Delta^{m_i - 1}$. We define the *partial action profile* $\mathbf{a}_{-i}$ to be the tuple of all players' actions except i's action, and similarly we define the *partial strategy profile* $\mathbf{s}_{-i}$. The expected payoff of i when she plays action $k \in [m_i]$, and all other players play according to $\mathbf{s}_{-i}$ is

$$u_i(k, \mathbf{s}_{-i}) := \sum_{\mathbf{a} \in A(N^-(i))} R_i(k; \mathbf{a}) \cdot \prod_{j \in N^-(i)} s_j(a_j).$$

Notice that this depends only on the in-neighbours of i.

The expected payoff to player i under $\mathbf{s}$ is therefore

$$u_i(\mathbf{s}) := \sum_{k \in [m_i]} u_i(k, \mathbf{s}_{-i}) \cdot s_i(k).$$

u	v
0	1
1	0
$\perp$	$\{0,1,\perp\}$

NOT gate

u	v	w
1	1	1
0	$\{0,1,\perp\}$	0
$\{0,1,\perp\}$	0	0
Else		$\{0,1,\perp\}$

AND gate

u	v	w
0	0	0
1	1	1
$\perp$	At least one output in $\{0,1\}$	

PURIFY gate

Figure 1: The truth tables of the three gates of Pure-Circuit.

A pure strategy k is a *best response* for player i against a partial strategy profile $\mathbf{s}_{-i}$ if it achieves the maximum payoff over all her pure strategies, that is,

$$u_i(k, \mathbf{s}_{-i}) = \max_{\ell \in [m_i]} u_i(\ell, \mathbf{s}_{-i}).$$

Pure strategy k is an ε-*best response* if the payoff it yields is within ε of a best response, meaning that

$$u_i(k, \mathbf{s}_{-i}) \geq \max_{\ell \in [m_i]} u_i(\ell, \mathbf{s}_{-i}) - \varepsilon.$$

Finally, the *best response payoff* for player i is

$$\mathrm{br}_i(\mathbf{s}_{-i}) := \max_{\ell \in [m_i]} u_i(\ell, \mathbf{s}_{-i}).$$

(Approximate) Nash equilibria. A strategy profile $\mathbf{s}$ is a *Nash equilibrium* if $\mathrm{br}_i(\mathbf{s}) = u_i(\mathbf{s})$ for all players i, i.e., every player achieves their best response payoff. A strategy profile $\mathbf{s}$ is an ε-*Nash equilibrium* (ε-NE) if every player's payoff is within ε of their best response payoff, meaning that $u_i(\mathbf{s}) \geq \mathrm{br}_i(\mathbf{s}) - \varepsilon$. A strategy profile $\mathbf{s}$ is an ε-*well supported Nash equilibrium* (ε-WSNE) if every player only plays strategies that are ε-best responses, meaning that for all i we have that $\mathsf{supp}(s_i)$ contains only ε-best response strategies.

2.2 The Pure-Circuit Problem

An instance of the Pure-Circuit problem is given by a node set $V = [n]$ and a set G of gate-constraints (or just *gates*). Each gate $g \in G$ is of the form $g = (T, u, v, w)$ where $u, v, w \in V$ are distinct nodes, and $T \in \{\mathsf{NOT}, \mathsf{AND}, \mathsf{PURIFY}\}$ is the type of the gate, with the following interpretation.

- If $T = \mathsf{NOT}$, then u is the input of the gate, and v is its output. (w is unused)
- If $T = \mathsf{AND}$, then u and v are the inputs of the gate, and w is its output.
- If $T = \mathsf{PURIFY}$, then u is the input of the gate, and v and w are its outputs.

We require that each node is the output of exactly one gate.

A solution to instance (V, G) is an assignment $\mathbf{x} : V \to \{0, 1, \perp\}$ that satisfies all the gates (see Fig. 1), i.e., for each gate $g = (T, u, v, w) \in G$ we have the following.

- If $T = \mathsf{NOT}$ in $g = (T, u, v)$, then $\mathbf{x}$ satisfies

$$\mathbf{x}[u] = 0 \implies \mathbf{x}[v] = 1$$
$$\mathbf{x}[u] = 1 \implies \mathbf{x}[v] = 0.$$

- If $T = \mathsf{AND}$ in $g = (T, u, v, w)$, then $\mathbf{x}$ satisfies

$$\begin{aligned} \mathbf{x}[u] = \mathbf{x}[v] = 1 &\implies \mathbf{x}[w] = 1 \\ x[u] = 0 \vee x[v] = 0 &\implies x[w] = 0. \end{aligned}$$

- If $T = \mathsf{PURIFY}$, then $\mathbf{x}$ satisfies

$$\begin{aligned} &\{\mathbf{x}[v], \mathbf{x}[w]\} \cap \{0,1\} \neq \emptyset \\ &\mathbf{x}[u] \in \{0,1\} \implies \mathbf{x}[v] = \mathbf{x}[w] = \mathbf{x}[u]. \end{aligned}$$

The structure of a Pure-Circuit instance is captured by its *interaction graph*. This graph is constructed on the vertex set $V = [n]$ by adding a directed edge from node u to node v whenever v is the output of a gate with input u. The total degree of a node is the sum of its in- and out-degrees.

Theorem 2.1 ([DFHM22]). Pure-Circuit *is* PPAD*-complete, even when every node of the interaction graph has in-degree at most 2 and total degree at most 3.*

3 Well-Supported Nash Equilibria

An easy upper bound. We present a polynomial time algorithm to compute a $\left(1 - \frac{2}{2^d+1}\right)$-WSNE in any two-action graphical game with maximum in-degree $d \geq 2$. The algorithm relies on a simple and natural approach that has been used for similar problems by Liu et al. [LLD21] and Deligkas et al. [DFHM22].

The algorithm proceeds in two steps. In the first step, it iteratively checks for a player that has an action that is ε*-dominant*; an action which, if played with probability 1, will satisfy the ε-WSNE conditions no matter what strategies the in-neighbours play. Here, we will set $\varepsilon = 1 - \frac{2}{2^d+1}$, where d is the maximum in-degree of the graph. If such a player with an ε-dominant action exists, the algorithm fixes the strategy of that player, updates the game accordingly, and iterates until there is no such player left. In the second step, the algorithm lets all remaining players mix uniformly, i.e., every player without an ε-dominant action plays each of its two actions with probability 1/2.

Theorem 3.1. *There is a polynomial-time algorithm that finds a* $\left(1 - \frac{2}{2^d+1}\right)$*-WSNE in a two-action graphical game with maximum in-degree d.*

Proof. It is easy to see that the algorithm described above runs in polynomial time. In particular, we can check if a player has an ε-dominant action by simply going over all possible action profiles of its in-neighbours. We will prove it computes an ε-WSNE for $\varepsilon = 1 - \frac{2}{2^d+1}$. By definition of ε-dominance, the players whose actions were fixed in the first step satisfy the constraints of ε-WSNE. Notice that after fixing any such player to play an ε-dominant action, we get a smaller graphical game. Thus, it suffices to consider the graphical game we obtain after the end of the first step, and to show that if all (remaining) players mix uniformly, this is an ε-WSNE.

So, consider a player i in the graphical game we obtain after the end of the first step, and denote its in-degree by $k := |N^-(i)| \leq d$. Since all in-neighbours of player i are mixing uniformly, the expected payoff of player i for playing action 0 is $\sum_{\mathbf{a} \in A(N^-(i))} R_i(0; \mathbf{a})/2^k$, and for playing action 1, it is $\sum_{\mathbf{a} \in A(N^-(i))} R_i(1; \mathbf{a})/2^k$. The constraints of ε-WSNE for player i are satisfied if both actions are ε-best responses, i.e., if

$$\left| \sum_{\mathbf{a} \in A(N^-(i))} R_i(0; \mathbf{a})/2^k - \sum_{\mathbf{a} \in A(N^-(i))} R_i(1; \mathbf{a})/2^k \right| \leq \varepsilon.$$

This can be rewritten as $\frac{1}{2^k}\left|\sum_{\mathbf{a}\in A(N^-(i))} f_i(\mathbf{a})\right| \le \varepsilon$, where, for any action profile $\mathbf{a} \in A(N^-(i))$, we let

$$f_i(\mathbf{a}) := R_i(0;\mathbf{a}) - R_i(1;\mathbf{a}).$$

Note that since all payoffs lie in $[0,1]$, we always have $f_i(\mathbf{a}) \in [-1,1]$.

Let $M := \sum_{\mathbf{a}\in A(N^-(i))} f_i(\mathbf{a})$. To prove the correctness of the algorithm, it suffices to prove that $|M| \le 2^k - 1 - \varepsilon$; since then $\frac{1}{2^k}\cdot|M| \le 1 - \frac{1+\varepsilon}{2^k} \le 1 - \frac{1+\varepsilon}{2^d} = \varepsilon$. To see why this is indeed the case, observe the following. Since player i does not have an ε-dominant action (otherwise, it would have been removed in the first step), it means that there exist $\mathbf{a}, \mathbf{a}' \in A(N^-(i))$ such that

$$f_i(\mathbf{a}) > \varepsilon \quad \text{and} \quad f_i(\mathbf{a}') < -\varepsilon. \tag{1}$$

Given that M is the sum of 2^k terms, each of them upper bounded by 1, and at least one of them upper bounded by $-\varepsilon$ by (1), it follows that $M \le 2^k - 1 - \varepsilon$. Similarly, since each term is also lower bounded by -1, and one of them is lower bounded by ε, we also obtain that $M \ge -2^k - 1 + \varepsilon$. Thus, $|M| \le 2^k - 1 - \varepsilon$, as desired, and this completes the proof of correctness. □

The lower bound. In this section we prove a matching lower bound for Theorem 3.1, which essentially proves that computing an ε-WSNE in two-action graphical games is PPAD-complete for every constant $\varepsilon \in (0,1)$.

We will prove our result by a reduction from Pure-Circuit. For the remainder of this section, we fix $\varepsilon < 1 - \frac{2}{2^d+1}$. Given a Pure-Circuit instance with in-degree 2, we build a two-action graphical game, where the two actions will be named zero and one. For any given $d \ge 2$, the game will have in-degree at most d. Each node v of the Pure-Circuit instance will correspond to a player in the game – the game will have some additional auxiliary players too – whose strategy in any ε-WSNE will encode a solution to the Pure-Circuit problem as follows. Given a strategy s_v for the player that corresponds to node v, we define the assignment $\mathbf{x}$ for Pure-Circuit such that:

- if $s_v(\mathsf{zero}) = 1$, then $\mathbf{x}[v] = 0$;
- if $s_v(\mathsf{one}) = 1$, then $\mathbf{x}[v] = 1$;
- otherwise, $\mathbf{x}[v] = \perp$.

We now give implementations for NOT, AND, and PURIFY gates. We note that, in all three cases, the payoff received by player v is only affected by the actions chosen by the players representing the inputs to the (unique) gate g that outputs to v. Thus, we can argue about the equilibrium condition at v by only considering the players involved in gate g, and we can ignore all other gates while doing this.

NOT gates. For a gate $g = (\mathsf{NOT}, u, v)$ – where recall that u is the input variable and v is the output variable – we create a gadget involving players u and v, where player v has a unique incoming edge from u. The payoffs of v are defined as follows.

- If u plays zero, then v gets payoff 0 from playing zero and payoff 1 from playing one.
- If u plays one, then v gets 1 from zero and 0 from one.

This gadget appeared in [DFHM22], but we include it here for completeness. We claim that this gadget works correctly.

- If $s_u(\mathsf{zero}) = 1$, i.e., u encodes 0, observe that for player v action zero yields payoff 0, while action one yields payoff 1. Hence, by the constraints imposed by ε-WSNE it must hold that $s_v(\mathsf{one}) = 1$, and thus v encodes 1.
- Using identical reasoning, we can prove that if $s_u(\mathsf{one}) = 1$, then $s_v(\mathsf{one}) = 0$ in any ε-WSNE.

AND gates. For a gate $g = (\mathsf{AND}, u, v, w)$ we create the following gadget with players u, v and w, where u and v are the in-neighbors of w. The payoffs of w are as follows.

- If $s_u(\mathsf{one}) = 1$ and $s_v(\mathsf{one}) = 1$, then w gets payoff 0 from playing zero and payoff 1 from playing one.
- For any other action profile of u and v, player w gets 1 from zero and 0 from one.

Next we argue that this gadget works correctly.

- If $s_u(\mathsf{one}) = 1$ and $s_v(\mathsf{one}) = 1$, i.e. both u and v encode 1, observe that for player w action zero yields payoff 0, while action one yields payoff 1. Hence, by the constraints imposed by ε-WSNE it must hold that $s_w(\mathsf{one}) = 1$, and thus w encodes 1.
- If at least one of u or v encodes 0, then for player w action zero yields expected payoff 1 while action one yields expected payoff 0. Hence, by the constraints imposed by ε-WSNE it must hold that $s_w(\mathsf{zero}) = 1$, and thus w encodes 0.

PURIFY gates. For a gate $g = (\mathsf{PURIFY}, u, v, w)$ we create the following gadget with $d + 3$ players. We introduce auxiliary players $u_1, u_2, \dots, u_d$. Each player u_i has a unique incoming edge from u. The idea is that in any ε-WSNE, every player u_i "copies" the strategy of player u.

- If u plays zero, then u_i gets 1 from zero and 0 from one.
- If u plays one, then u_i gets 0 from zero and 1 from one.

Lemma 3.2. *At any ε-WSNE the following hold for every $i \in [d]$: if $s_u(\mathsf{zero}) = 1$, then $s_{u_i}(\mathsf{zero}) = 1$; if $s_u(\mathsf{one}) = 1$, then $s_{u_i}(\mathsf{one}) = 1$.*

Proof. If $s_u(\mathsf{zero}) = 1$, then for player u_i action zero yields payoff 1, while action one yields payoff 0. Thus, the constraints of ε-WSNE dictate that $s_{u_i}(\mathsf{zero}) = 1$. If $s_u(\mathsf{one}) = 1$, then for player u_i action zero yields payoff 0, while action one yields payoff 1. Thus, the constraints of ε-WSNE dictate that $s_{u_i}(\mathsf{one}) = 1$. □

Next, we describe the payoff tensors of players v and w; each one of them has in-degree d with edges from all $u_1, u_2, \dots, u_d$. In what follows, fix $\lambda := 1 - \frac{2}{2^d+1}$. The payoffs of player v are as follows.

- If v plays zero and at least one of $u_1, \dots, u_d$ plays zero, then the payoff for v is 1.
- If v plays zero and every one of $u_1, \dots, u_d$ plays one, then the payoff for v is 0.
- If v plays one and at least one of $u_1, \dots, u_d$ plays zero, then the payoff for v is 0.
- If v plays one and every one of $u_1, \dots, u_d$ plays one, then the payoff for v is λ.

The payoffs of player w are as follows.

- If w plays zero and every one of $u_1, \dots, u_d$ plays zero, then the payoff for w is λ.
- If w plays zero and at least one of $u_1, \dots, u_d$ plays one, then the payoff for w is 0.

- If w plays one and every one of $u_1, \dots, u_d$ plays zero, then the payoff for w is 0.
- If w plays one and at least one of $u_1, \dots, u_d$ plays one, then the payoff for w is 1.

We are now ready to prove that this construction correctly simulates a PURIFY gate. We consider the different cases that arise depending on the value encoded by u.

– $s_u(\mathsf{zero}) = 1$, i.e., u encodes 0. From Lemma 3.2 we know that $s_{u_i}(\mathsf{zero}) = 1$ for every $i \in [d]$. Then, we have the following for players v and w.

  - Player v gets payoff 1 from action zero and payoff 0 from action one. Hence, in an ε-WSNE we get that $s_v(\mathsf{zero}) = 1$, and thus v encodes 0.
  - Player w gets payoff λ from action zero and payoff 0 from action one. Hence, since $\varepsilon < \lambda$, in an ε-WSNE we get that $s_w(\mathsf{zero}) = 1$, and thus w encodes 0.

– $s_u(\mathsf{one}) = 1$, i.e., u encodes 1. From Lemma 3.2 we know that $s_{u_i}(\mathsf{one}) = 1$ for every $i \in [d]$. Then, we have the following for players v and w.

  - Player v gets payoff 0 from action zero and payoff λ from action one. Hence, since $\varepsilon < \lambda$, in an ε-WSNE we get that $s_v(\mathsf{one}) = 1$, and thus v encodes 1.
  - Player w gets payoff 0 from action zero and payoff 1 from action one. Hence, in an ε-WSNE we get that $s_w(\mathsf{one}) = 1$, and thus w encodes 1.

– $s_u(\mathsf{one}) \in (0, 1)$, i.e., u encodes $\perp$. Then each one of the auxiliary players $u_1, \dots, u_d$ can play a different strategy. For each $i \in [d]$, denote $s_{u_i}(\mathsf{one}) = p_i$, i.e., $p_i \in [0, 1]$ is the probability player u_i assigns on action one. Let $P := \prod_{i\in[d]} p_i$ and $Q := \prod_{i\in[d]}(1 - p_i)$. Then, we have the following two cases.

  - $P \le 2^{-d}$. Then we focus on player v: action zero yields expected payoff $1-P \ge 1-2^{-d}$, while action one yields expected payoff $P \cdot \lambda \le 2^{-d} \cdot \lambda$. Then, since $\varepsilon < \lambda$, we get that in an ε-WSNE it must hold that $s_v(\mathsf{zero}) = 1$, i.e., v encodes 0.
  - $P > 2^{-d}$. Then, it holds that $Q < 2^{-d}$; this is because $P \cdot Q = \prod_{i\in[d]} p_i \cdot (1 - p_i) \le (1/4)^d$. In this case we focus on player w: action zero yields payoff $\lambda \cdot Q < \lambda \cdot 2^{-d}$, while action one yields payoff $1 - Q > 1 - 2^{-d}$. Then, again since $\varepsilon < \lambda$, in any ε-WSNE it must hold that $s_w(\mathsf{one}) = 1$, i.e., w encodes 1.

From the arguments given above, we have that in any ε-WSNE of the graphical game, with $\varepsilon < 1 - \frac{2}{2^d+1}$, the players correctly encode a solution to the Pure-Circuit instance.

Theorem 3.3. *Computing an ε-WSNE in two-action graphical games with maximum in-degree $d \ge 2$ is* PPAD*-complete for any $\varepsilon < 1 - \frac{2}{2^d+1}$.*

We can see that the constructed game is not win-lose since there is a payoff $\lambda \notin \{0, 1\}$ in the gadget that simulates PURIFY gates. However, if we set $\lambda = 1$, and use verbatim the analysis from above, we will get PPAD-hardness for ε-WSNE with $\varepsilon < 1 - \frac{1}{2^{d-1}}$.

Theorem 3.4. *Computing an ε-WSNE in two-action win-lose graphical games with maximum in-degree $d \ge 2$ is* PPAD*-complete for any $\varepsilon < 1 - \frac{1}{2^{d-1}}$.*

Since for every constant $\varepsilon < 1$ we can find a constant d such that Theorem 3.4 holds, we get the following corollary.

Corollary 3.5. *For any constant $\varepsilon < 1$, computing an ε-WSNE in two-action win-lose graphical games is* PPAD*-complete.*

4 Approximate Nash Equilibria

A straightforward upper bound. We first show that a 0.5-NE can easily be found in any two-action graphical game.

Theorem 4.1. *There is a polynomial-time algorithm that finds a* 0.5*-NE in a two-action graphical game.*

Proof. Let $\mathbf{s}$ be the strategy profile in which all players mix uniformly over their two actions. Then, for each player i we have

$$\begin{aligned} u_i(\mathbf{s}) &\geq 0.5 \cdot \mathrm{br}_i(\mathbf{s}_{-i}) \\ &\geq \mathrm{br}_i(\mathbf{s}_{-i}) - 0.5, \end{aligned}$$

where the final inequality used the fact that $\mathrm{br}_i(\mathbf{s}_{-i}) \in [0, 1]$. Thus, $\mathbf{s}$ is a 0.5-NE. □

The lower bound. We will show that computing a $(0.5 - \varepsilon)$-NE of a graphical game is PPAD-hard for any constant $\varepsilon > 0$ by a reduction from Pure-Circuit.

Given a Pure-Circuit instance, we build a two-action graphical game, where the two actions will be named zero and one. Each node v of the Pure-Circuit instance will be represented by a set of k players in the game, named $v_1, v_2, \ldots, v_k$, where we fix k to be an odd number satisfying $k \geq \ln(12/\varepsilon) \cdot 18/\varepsilon^2$. Therefore, since ε is constant, k is also constant.

The strategies of these players will encode a solution to the Pure-Circuit problem in the following way. Given a strategy profile $\mathbf{s}$, we define the assignment $\mathbf{x}$ such that

- If $s_{v_i}(\mathsf{zero}) \geq 0.5 + \varepsilon/3$ for all i, then $\mathbf{x}[v] = 0$.
- If $s_{v_i}(\mathsf{one}) \geq 0.5 + \varepsilon/3$ for all i, then $\mathbf{x}[v] = 1$.
- In all other cases, $\mathbf{x}[v] = \bot$.

We now give implementations for NOT, AND, and PURIFY gates. We note that, in all three cases, the payoff received by player v_i is only affected by the actions chosen by the players representing the inputs to the (unique) gate g that outputs to v. Thus, we can argue about the equilibrium condition at v_i by only considering the players involved in gate g, and we can ignore all other gates while doing this.

NOT gates. For a gate $g = (\mathsf{NOT}, u, v)$, we use the following construction, which specifies the games that will be played between the set of players that represent u and the set of players that represent v.

Each player v_i has incoming edges from all players $u_1, u_2, \ldots, u_k$, and v_i's payoff tensor is set as follows.

- If strictly more than[3] $k/2$ of the players u_1 through u_k play zero, then v_i receives payoff 0 for strategy zero and payoff 1 for strategy one.
- If strictly less than $k/2$ of the players u_1 through u_k play zero, then v_i receives payoff 1 for strategy zero and payoff 0 for strategy one.

We now show the correctness of this construction. We start with a technical lemma that we will use repeatedly throughout the construction.

Lemma 4.2. *Suppose that player p has two actions a and b. In any $(0.5 - \varepsilon)$-NE, if the payoff of action a is at most $\varepsilon/3$, and the payoff of action b is at least $1 - \varepsilon/3$, then player p must play action b with probability at least $0.5 + \varepsilon/3$.*

[3]Since k is odd, it is not possible for exactly $k/2$ players to play zero.

Proof. Since action b has payoff at least $1 - \varepsilon/3$, we have that the best response payoff to p is also at least $1 - \varepsilon/3$. Hence, in any strategy profile $\mathbf{s}$ that is an $(0.5 - \varepsilon)$-NE, we have

$$\begin{aligned} u_p(\mathbf{s}) &\geq \mathrm{br}_p(\mathbf{s}) - 0.5 + \varepsilon \\ &\geq 1 - \varepsilon/3 - 0.5 + \varepsilon \\ &= 0.5 + 2\varepsilon/3. \end{aligned}$$

Since the payoff of a is bounded by $\varepsilon/3$, and the payoff of b is bounded by 1 (since all payoffs are in the range $[0, 1]$), we obtain

$$\begin{aligned} u_p(\mathbf{s}) &\leq s_p(a) \cdot \varepsilon/3 + s_p(b) \cdot 1 \\ &= (1 - s_p(b)) \cdot \varepsilon/3 + s_p(b) \cdot 1 \\ &= s_p(b)(1 - \varepsilon/3) + \varepsilon/3 \\ &\leq s_p(b) + \varepsilon/3. \end{aligned}$$

Joining the two previous inequalities gives $s_p(b) + \varepsilon/3 \geq 0.5 + 2\varepsilon/3$, and therefore $s_p(b) \geq 0.5 + \varepsilon/3$. □

We can now prove that the NOT gadget operates correctly.

Lemma 4.3. *In every $(0.5 - \varepsilon)$-NE, the following properties hold.*

- *If the players representing u encode 0, then the players representing v encode 1.*
- *If the players representing u encode 1, then the players representing v encode 0.*

Proof. Let $\mathbf{s}$ be a $(0.5 - \varepsilon)$-NE. We begin with the first claim. Since the players representing u encode a 0, we have that $s_{u_j}(\mathsf{zero}) \geq 0.5 + \varepsilon/3$ for all $j \in [k]$.

We start by proving an upper bound on the payoff of action zero for player v_i. The payoff of this action increases as the players u_j place less probability on action zero, so we can assume $s_{u_j}(\mathsf{zero}) = 0.5 + \varepsilon/3$, since this minimizes the payoff of zero to v_i.

Under this assumption, the number N of players u_j that play action zero is distributed binomially according to $N \sim B(k, 0.5 + \varepsilon/3)$. Applying the standard Hoeffding bound [Hoe94] for the binomial distribution, and using the fact that $k \geq \ln(6/\varepsilon) \cdot 9/2\varepsilon^2$ yields the following

$$\begin{aligned} \Pr(N \leq k/2) &\leq 2 \cdot \exp\left(-2k\left(0.5 + \varepsilon/3 - \frac{k/2}{k}\right)^2\right) \\ &= 2 \cdot \exp\left(-2k \cdot \varepsilon^2/9\right) \\ &\leq \exp\left(-\ln(3/\varepsilon)\right) \\ &= \varepsilon/3. \end{aligned}$$

Hence the payoff of action zero to player v_i is at most $\varepsilon/3$, and therefore the payoff of action one to v_i is at least $1 - \varepsilon/3$.

So we can apply Lemma 4.2 to argue that $s_{v_i}(\mathsf{one}) \geq 0.5 + \varepsilon/3$. Since this holds for all i, we have that the players representing v encode the value 1 in the PURE-CIRCUIT instance, as required.

The second case can be proved in an entirely symmetric manner. □

AND gates. For a gate $g = (\mathsf{AND}, u, v, w)$ we use the following construction. Each player w_i has in-degree $2k$ and has incoming edges from all of the players $u_1, u_2, \ldots, u_k$, and all of the players $v_1, v_2, \ldots, v_k$. The payoff tensor of w_i is as follows.

- If strictly more than $k/2$ of the players u_1 through u_k play one, and strictly more than $k/2$ of the players v_1 through v_k play one, then w_i receives payoff 0 from action zero and payoff 1 from action one.
- If this is not the case, then w_i receives payoff 1 from action zero and payoff 0 from action one.

Correctness of this construction is shown in the following pair of lemmas.

Lemma 4.4. *In every $(0.5-\varepsilon)$-NE of the game, if the players representing u encode value 1, and the players representing v encode value 1, then the players representing w will encode value 1.*

Proof. Let $\mathbf{s}$ be a $(0.5-\varepsilon)$-NE. From the assumptions about u and v, we have that $s_{u_j}(\mathsf{one}) \geq 0.5+\varepsilon/3$ for all j, and $s_{v_j}(\mathsf{one}) \geq 0.5+\varepsilon/3$ for all $j \in [k]$.

We start by proving an upper bound on the payoff of zero to w_i. Since this payoff decreases as the players u_j and v_j place more probability on one, we can assume that $s_{u_j}(\mathsf{one}) = 0.5+\varepsilon/3$ for all j, and $s_{v_j}(\mathsf{one}) = 0.5+\varepsilon/3$ for all j, since this maximizes the payoff of zero to w_i.

The number N of players u_j that play one is distributed binomially according to $N \sim B(k, 0.5+\varepsilon/3)$. Similarly, the number of players v_j that play one, are distributed according to the same distribution, that is why we focus only in the former. Using the standard Hoeffding bound for the binomial distribution, along with the fact that $k \geq \ln(12/\varepsilon)\cdot 9/2\varepsilon^2$, we get

$$\begin{aligned}\Pr(N \leq k/2) &\leq 2\cdot\exp\left(-2k\left(0.5+\varepsilon/3-\frac{k/2}{k}\right)^2\right)\\ &= 2\cdot\exp\left(-2k\cdot\varepsilon^2/9\right)\\ &\leq \exp\left(-\ln(6/\varepsilon)\right)\\ &= \varepsilon/6.\end{aligned}$$

We can then use the union bound to prove that the probability that strictly less than $k/2$ of the players u_1 through u_k play one, or strictly less than $k/2$ of the players v_1 through v_k play one is at most $\varepsilon/3$.

Hence, the payoff of zero to player w_i is at most $\varepsilon/3$, and so the payoff of one to player w_i is at least $1-\varepsilon/3$. Thus we can apply Lemma 4.2 to argue that w_i must play one with probability at least $0.5+\varepsilon/3$. Since this holds for all i, we have that $w_1, w_2, \dots w_k$ correctly encode value 1. □

Lemma 4.5. *In every $(0.5-\varepsilon)$-NE of the game, if the players representing u encode value 0, or the players representing v encode value 0, then the players representing w will encode value 0.*

Proof. We will provide a proof for the case where the players representing u encode value 0, since the other case is entirely symmetric.

Let $\mathbf{s}$ be an $(0.5-\varepsilon)$-NE. By assumption we have that $s_{u_j}(\mathsf{zero}) \geq 0.5+\varepsilon/3$ for all j. We start by proving an upper bound on the payoff of one to w_i. Since the payoff of this strategy decreases as u_j places more probability on zero, we can assume that $s_{u_j}(\mathsf{zero}) = 0.5+\varepsilon/3$ for all j, since this minimizes the payoff of one to w_i.

The number of players u_j that play one is distributed binomially according to $N \sim B(k, 0.5+\varepsilon/3)$. Using the standard Hoeffding bound for the binomial distribution, along with the fact that $k \geq \ln(6/\varepsilon)\cdot 9/2\varepsilon^2$, we get

$$\Pr(N \leq k/2) \leq 2\cdot\exp\left(-2k\left(0.5+\varepsilon/3-\frac{k/2}{k}\right)^2\right)$$

$$
\begin{aligned}
&= 2 \cdot \exp\left(-2k \cdot \varepsilon^2/9\right) \\
&\leq \exp\left(-\ln(3/\varepsilon)\right) \\
&= \varepsilon/3.
\end{aligned}
$$

Hence, the payoff of one to player w_i is at most $\varepsilon/3$, and so the payoff of zero to player w_i is at least $1 - \varepsilon/3$. Thus, we can apply Lemma 4.2 to argue that w_i must play zero with probability at least $0.5 + \varepsilon/3$. Since this holds for all i, we have that $w_1, w_2, \ldots w_k$ correctly encode value 0. □

PURIFY gates. For a gate $g = (\mathsf{PURIFY}, u, v, w)$, we use the following construction. Each player v_i will have incoming edges from all players $u_1, u_2, \ldots, u_k$, with their payoff tensor set as follows.

- If strictly more than $(0.5 - \varepsilon/6) \cdot k$ of the players u_1 through u_k play one, then v_i receives payoff 0 from action zero and payoff 1 from action one.
- If this is not the case, then v_i receives payoff 1 from action zero and payoff 0 from action one.

Each player w_i will have incoming edges from all players $u_1, u_2, \ldots, u_k$, with their payoff tensor set as follows.

- If strictly more than $(0.5 + \varepsilon/6) \cdot k$ of the players u_1 through u_k play one, then w_i receives payoff 0 from action zero and payoff 1 from action one.
- If this is not the case, then w_i receives payoff 1 from action zero and payoff 0 from action one.

The following lemma shows that v is correctly simulated.

Lemma 4.6. *Let* $\mathbf{s}$ *be a* $(0.5 - \varepsilon)$*-NE, and let* $E[X]$ *denote the expected number of players* u_i *that play strategy* one *under* $\mathbf{s}$.

- *If* $E[X] \leq (0.5 - \varepsilon/3) \cdot k$*, then the players representing* v *will encode value* 0.
- *If* $E[X] \geq 0.5 \cdot k$*, then the players representing* v *will encode value* 1.

Proof. We will prove only the first case, since the second case can be proved in an entirely symmetric manner. Using the fact that $E[X] \leq (0.5 - \varepsilon/3) \cdot k$, then applying Hoeffding's inequality, and using the fact that $k \geq \ln(3/\varepsilon) \cdot 18/\varepsilon^2$ we get

$$
\begin{aligned}
\Pr(X \geq (0.5 - \varepsilon/6) \cdot k) &\leq \Pr(X - E[X] \geq \varepsilon/6 \cdot k) \\
&\leq \exp\left(\frac{-2(k \cdot \varepsilon/6)^2}{k}\right) \\
&= \exp\left(-k \cdot \varepsilon^2/18\right) \\
&\leq \varepsilon/3.
\end{aligned}
$$

Therefore, the payoff of strategy one to v_i is at most $\varepsilon/3$, and so the payoff of strategy zero to v_i is at least $1 - \varepsilon/3$. Thus we can apply Lemma 4.2 to argue that $s_{v_i}(\mathsf{zero}) \geq 0.5 + \varepsilon/3$. Since this applies for all i, we have that the players representing v encode value 0, as required. □

The next lemma shows that w is also correctly simulated; its proof is omitted since it is entirely symmetric to the proof of Lemma 4.6.

Lemma 4.7. *Let* s *be a* $(0.5-\varepsilon)$*-NE, and let* $E[X]$ *denote the expected number of players* u_i *that play strategy* one *under* s*.*

- *If* $E[X] \leq 0.5 \cdot k$*, then the players representing* w *will encode value* 0*.*
- *If* $E[X] \geq (0.5+\varepsilon/3) \cdot k$*, then the players representing* w *will encode value* 1*.*

Combining the two previous lemmas, we can see that the construction correctly simulates a PURIFY gate.

- If the players representing u encode value 0, then $E[X] \leq 0.5 - \varepsilon/3$, and so both the players representing v and those representing w encode value 0.
- If the players representing u encode value 1, then $E[X] \geq 0.5 + \varepsilon/3$, and so both the players representing v and those representing w encode value 1.
- In all other cases we can verify that either the players representing v or the players representing w encode a 0 or a 1. Specifically, if $E[X] \leq 0.5 \cdot k$ then the players representing w encode value 0, while if $E[X] \geq 0.5 \cdot k$, then the players representing v encode value 1.

The hardness result. From the arguments given above, we have that in an $(0.5-\varepsilon)$-NE of the graphical game, the players correctly encode a solution to the Pure-Circuit instance. Note also that, since Theorem 2.1 gives hardness for Pure-Circuit even when the total degree of each node is 3, the graphical game that we have built has total degree at most $3k$. Thus, the game can be built in polynomial time.

Theorem 4.8. *It is* PPAD*-hard to find a* $(0.5-\varepsilon)$*-NE in a two-action graphical game for any constant* $\varepsilon > 0$*.*

In fact, the payoff entries in all gadgets are 0 or 1. Thus, our PPAD-hardness result holds for win-lose games too.

Corollary 4.9. *For any constant* $\varepsilon > 0$*, it is* PPAD*-hard to find a* $(0.5-\varepsilon)$*-NE in a two-action win-lose graphical game.*

5 Conclusions

We have resolved the computational complexity of finding approximate Nash equilibria in two-action graphical games, by providing complete characterizations for both ε-NE and ε-WSNE. Our results show that finding approximate Nash equilibria in graphical games is much harder when compared to the special case of (constant-degree) polymatrix games: for two-action polymatrix games the tractability threshold for ε-WSNE is $1/3$ [DFHM22].

Below we identify two research questions that deserve more research.

- What is the intractability threshold for ε-NE in graphical games with more than two actions? We have shown that 0.5 is the threshold for two-action games, and we conjecture that 0.5 is the correct answer for the multi-action case as well. Hence, we view the main open problem as finding a polynomial-time algorithm that finds a 0.5-NE in any graphical game. We note that such an algorithm is already known for the special case of polymatrix games [DFSS17].
- What is the intractability threshold for ε-NE and ε-WSNE in polymatrix games? For ε-NE our understanding is far from complete, even in the two-action case, since there is a substantial gap between the $1/48$ lower bound and the $1/3$ upper bound (where the latter actually comes from the tractability of $1/3$-WSNE) given in [DFHM22]. For ε-WSNE, although the problem is completely resolved for the two-action case, the gap in multi-action polymatrix games is still large, and it seems that improving either the known lower bound of $1/3$, or the trivial upper bound of 1, would require significantly new ideas.

Acknowledgements

The second author was supported by EPSRC grant EP/W014750/1 “New Techniques for Resolving Boundary Problems in Total Search”. The third author was supported by the Swiss State Secretariat for Education, Research and Innovation (SERI) under contract number MB22.00026.

References

[BLP15] Siddharth Barman, Katrina Ligett, and Georgios Piliouras. Approximating Nash equilibria in tree polymatrix games. In *Proc. of SAGT*, pages 285–296, 2015.

[CDT09] Xi Chen, Xiaotie Deng, and Shang-Hua Teng. Settling the complexity of computing two-player Nash equilibria. *Journal of the ACM*, 56(3):14:1–14:57, 2009. doi:10.1145/1516512.1516516.

[DFHM22] Argyrios Deligkas, John Fearnley, Alexandros Hollender, and Themistoklis Melissourgos. Pure-circuit: Strong inapproximability for PPAD. In *Proc. of FOCS*, page (to appear), 2022.

[DFS17] Argyrios Deligkas, John Fearnley, and Rahul Savani. Computing constrained approximate equilibria in polymatrix games. In *Proc. of SAGT*, pages 93–105, 2017.

[DFS20] Argyrios Deligkas, John Fearnley, and Rahul Savani. Tree polymatrix games are PPAD-hard. In *Proc. of ICALP*, volume 168, pages 38:1–38:14, 2020. doi:10.4230/LIPIcs.ICALP.2020.38.

[DFSS17] Argyrios Deligkas, John Fearnley, Rahul Savani, and Paul G. Spirakis. Computing approximate Nash equilibria in polymatrix games. *Algorithmica*, 77(2):487–514, 2017.

[DGP09] Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a Nash equilibrium. *SIAM Journal on Computing*, 39(1):195–259, 2009. doi:10.1137/070699652.

[EGG06] Edith Elkind, Leslie Ann Goldberg, and Paul W. Goldberg. Nash equilibria in graphical games on trees revisited. In *Proc. of EC*, pages 100–109, 2006.

[GGJ$^+$10] Andrea Galeotti, Sanjeev Goyal, Matthew O Jackson, Fernando Vega-Redondo, and Leeat Yariv. Network games. *The review of economic studies*, 77(1):218–244, 2010.

[Hoe94] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. In *The collected works of Wassily Hoeffding*, pages 409–426. Springer, 1994.

[Jac11] Matthew O Jackson. An overview of social networks and economic applications. *Handbook of social economics*, 1:511–585, 2011.

[JZ15] Matthew O Jackson and Yves Zenou. Games on networks. In *Handbook of game theory with economic applications*, volume 4, pages 95–163. Elsevier, 2015.

[Kea07] Michael Kearns. Graphical games. In Noam Nisan, Tim Roughgarden, Éva Tardos, and Vijay V. Vazirani, editors, *Algorithmic Game Theory*, pages 159–180. Cambridge University Press, 2007. doi:10.1017/cbo9780511800481.009.

[KLS01] Michael Kearns, Michael L. Littman, and Satinder Singh. Graphical models for game theory. In *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 253–260, 2001. doi:10.48550/arXiv.1301.2281.

[LLD21] Zhengyang Liu, Jiawei Li, and Xiaotie Deng. On the approximation of Nash equilibria in sparse win-lose multi-player games. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, pages 5557–5565, 2021. URL: https://ojs.aaai.org/index.php/AAAI/article/view/16699.

[OI17] Luis E. Ortiz and Mohammad Tanvir Irfan. Tractable algorithms for approximate Nash equilibria in generalized graphical games with tree structure. In *Proc. of AAAI*, pages 635–641, 2017.

[Pap94] Christos H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, 48(3):498–532, 1994. doi:10.1016/S0022-0000(05)80063-7.

[Rub18] Aviad Rubinstein. Inapproximability of Nash equilibrium. *SIAM Journal on Computing*, 47(3):917–959, 2018. doi:10.1137/15M1039274.