Smoothed Efficient Algorithms and Reductions for Network Coordination Games

Shant Boodaghians
University of Illinois at Urbana-Champaign, Urbana, IL, USA
boodagh2@illinois.edu

Rucha Kulkarni
University of Illinois at Urbana-Champaign, Urbana, IL, USA
ruchark2@illinois.edu

Ruta Mehta
University of Illinois at Urbana-Champaign, Urbana, IL, USA
rutamehta@cs.illinois.edu

Abstract

We study the smoothed complexity of finding pure Nash equilibria in Network Coordination Games, a PLS-complete problem in the worst case, even when each player has two strategies. This is a potential game where the *sequential-better-response* algorithm is known to converge to a pure NE, albeit in exponential time. First, we prove polynomial (respectively, quasi-polynomial) smoothed complexity when the underlying game graph is complete (resp. arbitrary), and every player has constantly many strategies. The complete graph assumption is reminiscent of perturbing *all parameters*, a common assumption in most known polynomial smoothed complexity results. We develop techniques to bound the probability that an (adversarial) better-response sequence makes slow improvements to the potential. Our approach combines and generalizes the local-max-cut approaches of Etscheid and Röglin (SODA '14; ACM TALG, '17) and Angel, Bubeck, Peres, and Wei (STOC '17), to handle the multi-strategy case. We believe that the approach and notions developed herein could be of interest in addressing the smoothed complexity of other potential games.

Further, we define a notion of a *smoothness-preserving reduction* among search problems, and obtain reductions from 2-strategy network coordination games to local-max-cut, and from k-strategy games (k arbitrary) to local-max-bisection. The former, with the recent result of Bibak, Chandrasekaran, and Carlson (SODA '18) gives an alternate $O(n^8)$-time smoothed algorithm when $k = 2$. These reductions extend smoothed efficient algorithms from one problem to another.

2012 ACM Subject Classification Theory of computation → Algorithmic game theory

Keywords and phrases Network Coordination Games, Smoothed Analysis

Digital Object Identifier 10.4230/LIPIcs.ITCS.2020.73

Related Version A full version of this paper is available at `https://arxiv.org/abs/1809.02280`.

Funding All three authors acknowledge support from NSF grant CCF 1750436.

Acknowledgements We would like to thank Pravesh Kothari for the insightful discussions in the initial stages of this work.

1 Introduction

Coordination games are a widely studied class of games, where players receive equal payoffs, and so are incentivized to coordinate. Network coordination games are a succinctly represented, natural multi-player extension of coordination games. The players simultaneously play multiple two-player coordination games, and receive the sum of their payoffs from these individual games. As a caveat, the players must choose the same strategy to play in all games.

These games naturally arise in various settings like social and biological networks [3, 32, 29], and have been extensively studied in various areas like Game theory and economics, Learning, Networks etc [25, 11, 20, 14, 2].

The natural dynamics in such a game imply that agents will change their strategy choices if this increases their payoff. Because these are coordination games, this also increases the total sum of payoffs. This sum is then a proxy for the progression towards an equilibrium, where no player can improve, hence is a *potential function* for the game, and the game becomes a *potential game*. When no player can benefit by deviating, or equivalently the potential function reaches a *local maximum*, this is a *pure Nash equilibrium*, and the standard search problem for most potential games is to find such an equilibrium.

Finding a pure Nash equilibrium in a network coordination game is complete for the class PLS (Polynomial Local Search) [12]. Although it is widely conjectured that PLS is unlikely to lie in P [6, 9, 38], problems in this class admit local-search algorithms [27], which have been observed to be empirically fast [27, 15, 18], but requiring exponential time in the worst case [40, 39]. To understand this discrepancy, we naturally turn to a beyond worst-case analysis technique called *smoothed analysis*, which "continuously interpolates between the worst-case and average-case analyses of algorithms," [41] (see Section 1.2 for a detailed discussion). Informally, we wish to show that adversarial instances are "scattered" in a probabilistic sense. We say that an algorithm is smoothed-efficient if it is efficient with high probability when all input parameters are perturbed by small random noise. This is one of the strongest performance metrics beyond worst-case performance. We ask the following:

▶ **Question.** *Can we design smoothed efficient algorithms for finding pure Nash equilibria for network coordination games?*

In this paper we answer the question in the affirmative. In particular, we obtain smoothed (quasi-)polynomial time algorithms to find pure Nash equilibria (PNE) in network-coordination games (NetCoordNash) with a constant number of strategies. We also introduce a notion for a *smoothness-preserving reduction*, and show that a special case of NetCoordNash admits such a reduction to local-max-cut, and the general case admits a reduction to local-max-bisection (see Section 2.2 for the problem definitions).

To the best of our knowledge, no smoothed efficient algorithm for a worst-case hard Nash equilibrium problem was known prior to this work, apart from the party affiliation games, the smoothed complexity of which directly follows from local-max-cut [23]. Similarly, to the authors' knowledge, no notion of smoothness-preserving reduction has been shown in the past, and we believe that such reductions are of independent interest.

Local-max-cut is a PLS-complete problem, where the goal is to find a cut in a graph that is maximal up to switching one vertex. In a recent series of results, the smoothed complexity of local-max-cut was shown to be first quasi-polynomial for arbitrary graphs [22], and then polynomial for complete graphs [1]. Both results and a recent (simultaneous) work [8] follow a common high-level framework: using anti-concentration bounds for linear combinations of random variables to argue that bad events are unlikely. Our analysis extends this high-level approach to NetCoordNash. However, local-max-cut is a special case of NetCoordNash where every player has two strategies and the matrix on every edge is off-diagonal (see Figure 1). To handle the extra complexity of NetCoordNash in general, we need to define novel bounds.

1.1 Our Results

A network coordination game is represented by an undirected *game graph* $G = (V, E)$, where the nodes are the players, and each player $v \in V$ simultaneously plays a two-player coordination game with each of its neighbors. If players have k strategies to choose from,

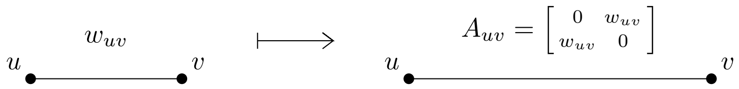

w_{uv}
u
v
$A_{uv} = \begin{bmatrix} 0 & w_{uv} \\ w_{uv} & 0 \end{bmatrix}$
u
v

Figure 1 Local-max-cut to 2-strategy network coordination games: mapping of edge (u, v).

the game on each edge uv can be represented by a $k \times k$ payoff matrix. Once every player chooses a strategy, the payoff value for each edge is fixed, and players receive the sum of the payoffs from incident edges. The goal is to find a PNE of this game. We show the natural *better-response algorithm* converges quickly with high probability on perturbed instances.

Smoothed Analysis of NetCoordNash. A strategy profile is at equilibrium if no single player can gain by deviating while others' unilaterally. Better-response dynamics/algorithms are an iterative procedure where any player who benefits by deviating, does so, one at a time, until an equilibrium is reached. This need not converge in general, *e.g.* for a game of rock-paper-scissors, where the players would infinitely cycle through the three strategies. In our setting, however, the sum of payoffs of all players acts as a *potential function*, which increases with every better-response move (Section 2.1). Thus, starting from any initial choice of strategies, *better-response algorithm (BRA)* will converge to a PNE in network coordination games, since it can be shown that the potential function is bounded.

We show that the BRA is an efficient algorithm with probability $1 - 1/poly(n)$ for perturbed instances: when the payoff values are independently sampled from distributions with density bounded by ϕ, the runtime will be polynomial in ϕ and the input size with high probability. One may interpret ϕ as the inverse of the minimum allowed perturbation. The exact relation between perturbation size and running time are as follows:

▶ **Theorem 1.** *Let $G = (V, E)$ be a game graph for an instance of NetCoordNash, with $k \times k$ payoff matrices, whose entries are independently distributed, continuous, random variables, with densities $f_{u,v,i,j} : [-1, 1] \to [0, \phi]$. Let $n := |V|$. If G is a complete graph, then with probability $1 - (nk)^{-3}$, all valid executions of the BRA (even adversarial) will converge to a PNE in at most $(nk\phi)^{O(k)}$ steps, and the expected running time $(nk\phi)^{O(k)}$ as well.*

If G is arbitrary, all valid executions of the BRA, from all starting points, will converge to a PNE in at most $\phi \cdot (nk)^{O(k \log(nk))}$ steps with probability $1 - (nk)^{-2}$ over the payoff entries. Furthermore, the expected running time is also at most $\phi \cdot (nk)^{O(k \log(nk))}$.

The proof of this is discussed in Section 3. The polynomial running time requires the graph to be complete so that *all* parameters can be perturbed. This seems to be unavoidable as all known results on *polynomial* smoothed complexity so far, *e.g.,* linear-programming [41], local-max-cut [1], etc., require this.

The above performance guarantees are only (quasi-)polynomial in the input size for k constant. It is an open problem to improve this to general k. This can be achieved either by showing that local-max-bisection has polynomial smoothed complexity (see below), or by directly tightening the bounds in the proof presented in this paper (Section 3).

Smoothness-Preserving Reductions. Note that standard Karp reductions do not suffice to extend a smoothed efficient algorithm across problems. This is because, among other things, such a reduction needs to ensure that independently perturbed parameters of the original problem produce independent perturbations of all parameters in the reduced problem. In this work, we introduce a notion of a *smoothness-preserving reduction*, which to the knowledge of the authors, has not been studied prior to this work. We obtain two such reductions:

▶ **Theorem 2.** *NetCoordNash with 2×2 payoff matrices admits a weak smoothness-preserving reduction to the local-max-cut problem. Furthermore, NetCoordNash with $k \times k$ matrices for general k admits a weak smoothness-preserving reduction to the local-max-bisection problem. For both results, an instance of NetCoordNash with a general or complete game graph reduces to an instance of local-max-cut/bisection on a general or complete graph, respectively.*

The definition of weak reductions is given in Section 4, and a formal statement of the local-max-cut and -bisection problems is in Section 2.2. In this writeup, we provide an outline of the reductions in Section 4, and a formal description and proofs of correctness are given in the full version of this paper [10]. The first reduction, together with smoothed efficient algorithms for local-max-cut, gives alternate smoothed efficient algorithms for the $k = 2$ instance; namely, the result of [8] implies an $O(n^8)$ algorithm when the game graph is complete. For general network coordination games, the smoothed complexity of local-max-bisection is open, and so any conclusion on the complexity of NetCoordNash is conditional.

1.2 Related Work

The works most closely related to ours are [22] and [1], where the smoothed complexity of local-max-cut was first analyzed, and [8] which refined the analyses. As discussed above, local-max-cut is a special case of NetCoordNash, therefore the techniques do not immediately apply. Independently, [8] also obtained smoothed polynomial algorithms for local-max-3-cut on complete graphs, and quasi-polynomial algorithms in general for local-max-k-cut with constant k. Local-max-k-cut naturally reduces to NetCoordNash with $k \times k$ payoff matrices. However, our result does not subsume theirs as the reduction is not *smoothness preserving.*

Complexity of Equilibrium Computation. There has been extensive work on various potential games, equivalently congestion games (e.g., [35, 31, 36, 23]), capturing routing and traffic situations (e.g., [37, 25]), and resource allocation under strategic agents (e.g., [26, 24]). A potential function ensures that these games always have a pure NE [35]. Finding pure NE is typically PLS-complete in the worst case [23, 12]. Our approach provides tools to prove smoothed analysis results for such games.

For the general games, NE computation is PPAD-complete [17, 13], even (1/poly)-additive approximation. This latter result also implies smoothed complexity does not lie in P unless RP = PPAD [13]. Towards average case analysis, Bárány, Vempala, and Vetta [5] showed that a game picked uniformly at random has a NE with support size 2 for both the players with inverse-logarithmic probability The average case complexity of a random potential game was shown to be polynomial in the number of players and strategies by Durand and Gaujal [19].

Smoothed Analysis. The work of Spielman and Teng [41] introduced the smoothed analysis framework to study the empirical performance of the Simplex method for linear programming. They showed that introducing independent random perturbations to any given (adversarial) LP instance, ensures that Simplex terminates fast with high probability, with polynomial dependence on the inverse of the magnitude of perturbation. Performance on such perturbed instances has since been known as *smoothed complexity* of the problem – one of the strongest guarantees one can hope for *beyond worst-case* analysis. In the past decade and a half, much work has sought to obtain smoothed efficient algorithms when worst-case efficiency seems infeasible [16, 7, 30, 34, 4, 21, 22, 1], including for integer programming, binary search trees, iterative-closest-point (ICP) algorithms, the 2-OPT algorithm for the Traveling Salesman problem (TSP), the knapsack problem, and the local-max-cut problem.

2 Preliminaries: Game Model and Smoothed Analysis

In what follows, the set $\{1, 2, \ldots, k\}$ is denoted as $[k]$, and $\langle \cdot, \cdot \rangle$ denotes inner product.

2.1 Nash Equilibria in Network Coordination Games

A two-player game, where each player has finitely many strategies to choose from is given by two payoff matrices A and B. Assume without loss of generality that both players have k strategies, and thus the matrices are $k \times k$. It is called a *coordination game* if $A = B$.

A *network coordination game* is a multi-player extension of coordination games. The game is specified by an underlying undirected graph $G = (V, E)$, where the nodes are players, and each edge represents a two-player coordination game between its endpoints. It is a *k-network coordination game* if each player has k strategies. For disambiguation, we represent the payoff values as an $|E|k^2$-dimensional vector A, and denote as $A((u, i)(v, j))$ the payoff that players u and v get for the game-edge $uv \in E$, when u chooses strategy i, and v, strategy j. As Nash equilibria are invariant to shifting and scaling of the payoffs, assume without loss of generality that every entry of A is contained in $[-1, 1]$.

Potential Function. Below we show that it suffices to only consider pure strategies, *i.e.* players needn't randomize. Let n be the number of players; a *strategy profile* is a vector $\boldsymbol{\sigma} \in [k]^n$, assigning to each player a strategy in $[k]$. The payoff to player u is given by

$$\text{payoff}_u(\boldsymbol{\sigma}) := \textstyle\sum_{v:\, uv \in E} A((u, \sigma_u)(v, \sigma_v))$$

Define the *potential function* $\Phi : [k]^n \to \mathbb{R}$ to be the sum of all payoffs. Formally,

$$\Phi(\boldsymbol{\sigma}) := \textstyle\sum_{(u,v)\in E} A((u, \sigma_u), (v, \sigma_v)) = \frac{1}{2} \sum_{u\in V} \text{payoff}_u(\boldsymbol{\sigma}) \tag{1}$$

The potential function is of interest since it captures the possible improvements to all players' payoffs in the following sense [12]: if player u changes their strategy, $\Phi(\boldsymbol{\sigma})$ and $\text{payoff}_u(\boldsymbol{\sigma})$ vary by the same amount. Formally, for all $u \in V$, $\sigma_u, \sigma'_u \in [k]$, and $\boldsymbol{\sigma}_{-u} \in [k]^{n-1}$

$$\Phi(\sigma_u, \boldsymbol{\sigma}_{-u}) - \Phi(\sigma'_u, \boldsymbol{\sigma}_{-u}) \;=\; \text{payoff}_u(\sigma_u, \boldsymbol{\sigma}_{-u}) - \text{payoff}_u(\sigma'_u, \boldsymbol{\sigma}_{-u})$$

where $\boldsymbol{\sigma}_{-u} \in [k]^{n-1}$ denotes the strategy profile on $V \setminus u$. Network coordination games are termed *potential games* because they admit such a potential function. As a consequence, they must admit pure Nash equilibria [35].

Nash Equilibrium and Better-Response Algorithm (BR alg., or BRA). At a Nash equilibrium (NE), no player gains by deviating unilaterally.

$$\textit{NE:} \quad \forall u \in V, \quad \text{payoff}_u(\sigma_u, \boldsymbol{\sigma}_{-u}) \geq \text{payoff}_u(\sigma'_u, \boldsymbol{\sigma}_{-u}), \quad \forall \sigma'_u \in [k]$$

Such a $\boldsymbol{\sigma}$ is called *pure* NE (PNE) as every player is playing a deterministic strategy. By the discussion above, $\boldsymbol{\sigma}$ is a PNE if and only if it is a local maximum for Φ, where $\boldsymbol{\sigma}'$ is in the local neighbourhood of $\boldsymbol{\sigma}$ when they differ in exactly one entry. A deviation for one player is termed a *better-response (BR) move* if their individual payoff strictly increases. Note that if $\boldsymbol{\sigma}'$ is a BR deviation from $\boldsymbol{\sigma}$, differing in a single player, then $\Phi(\boldsymbol{\sigma}') > \Phi(\boldsymbol{\sigma})$. The *better-response algorithm* (BRA) repeatedly makes better-response moves, increasing the Φ value in every step. The terminating point has to be a local maximum of Φ, and thereby a PNE. Since Φ may only take k^n different values, this procedure must terminate at a PNE.

2.2 Smoothed Analysis and Reductions

The notion of smoothed analysis was introduced by Spielman and Teng [41] to bridge the gap between average- and worst-case analysis. The parameters of the problem are perturbed by some small noise, and the performance is measured as a function of the perturbation size. We present first a formal definition of smoothed-efficient algorithms in our setting.

▶ **Definition 3** (Independent distributions with bounded density)**.** *Let X be a random vector in $[-1,1]^d$. We say it is* independently distributed with density bounded by ϕ *if the entries are independently distributed, and the p.d.f. for the i-th entry is a function $f_i : [-1,1] \to [0,\phi]$. Observe that the joint distribution on X has p.d.f. upper-bounded by ϕ^d.*

Intuitively, a bounded-density X is "spread" by at least $1/\phi$. Running-time bounds will be defined as a function of ϕ. We define here polynomial smoothed complexity in our setting:

▶ **Definition 4** (Polynomial Smoothed Complexity)**.** *Let $\mathcal{P}$ be a search problem, whose instances consist of some structural information D – e.g. a graph – and some real-valued information X – e.g. edge weights. We say $\mathcal{A}$ is a* smoothed efficient algorithm *for $\mathcal{P}$ if, $\mathcal{A}(D,X)$ returns a correct solution with probability 1, and there exist constants $c, c' > 0$ such that whenever $X \in \mathbb{R}^d$ is an independently distributed random vector with density bounded by ϕ,*

$$\max_D \Pr_X \left[\text{running time of } \mathcal{A} \text{ on } (D,X) \ \geq (d \cdot |D| \cdot \phi)^c\right] \ \leq (d \cdot |D|)^{-c'} \ .$$

$\mathcal{P}$ is said to have polynomial smoothed complexity *if a smoothed efficient algorithm exists. It has* quasi-poly smoothed complexity *if the above holds for running time $(d \cdot |D| \cdot \phi)^{O(\log(d \cdot |D|))}$.*

An algorithm is smoothed-efficient in expectation if the expected running time over a worst-case choice of ϕ-bounded distributions is (quasi-)polynomial in d, ϕ, and $|D|$; a problem $\mathcal{P}$ is said to have (quasi-)polynomial smoothed complexity in expectation similarly.

Local-max-cut and -bisection. In this paper, we define *smoothness preserving reductions* which allow the extension of smoothed-complexity results, as defined above, from one problem to another. Namely, we obtain reductions to the local-max-cut and -bisection problems. These problems are defined as FLIP and SWAP respectively in [40]. Given a weighted graph $G = (V,E)$, local-max-cut is the problem of finding a cut which is maximal up to flipping one vertex across the cut, and local-max-bisection is the problem of finding a balanced cut of the nodes into two sets of equal size, whose cut value is maximal up to swapping a pair of nodes across the cut. Both problems are shown to be PLS-hard in [40], and the smoothed complexity of local-max-cut has been studied at length, as discussed in the introduction.

3 Smoothed Performance of the BR algorithm

This section is a partial proof of Theorem 1. All missing details may be found in the full version [10]. Recall that Theorem 1 stated that for ϕ-bounded random variables, the BRA converges in $(nk\phi)^{O(k)}$ steps with probability $1-(nk)^{-3}$ on complete graphs, and $\phi \cdot (nk)^{O(k \log(nk))}$ steps with probability $1-(nk)^{-2}$ over the payoff entries. Only the payoff values are randomized, the BR moves may be chosen adversarially after sampling.

Recall that a profile $\boldsymbol{\sigma}$ is a PNE if and only if it is a local maximum of the potential Φ. Note that Φ may only take values in the interval $[-n^2, n^2]$, since the payoffs are in $[-1,1]$. It suffices then to show that with high probability, *every* linear-length sequence of BR moves has significant increase in Φ. We show the following:

▶ **Theorem 5.** *Let $G = (V, E)$ be a game graph, with random payoff vector A, and $\boldsymbol{\sigma}^0 \in [k]^n$ be an arbitrary strategy profile. With probability $1 - (nk)^{-2}$ over the values of A, all BR sequences of length at least $2nk$, initiated at any choice of $\boldsymbol{\sigma}^0$, must have at least one step in which the potential increases by $\epsilon = \phi^{-1}(2n^2k^3)^{-20k\log(nk)}$. If G is a complete graph, then with probability $1 - (nk)^{-3}$, all BR sequences of length at least $2nk$, will have at least one step increasing by $\epsilon' = (20\phi^2n^3k^3)^{-4k-4}$.*

From the above discussion, this implies that the BRA must terminate in $\phi(nk)^{O(k\log(nk))}$ steps with probability $1-(nk)^{-2}$ in general, and $(\phi nk)^{O(k)}$ steps with probability $1-(nk)^{-3}$ on complete game graphs, implying Theorem 1. We proceed with a proof of Theorem 5.

The high level approach of this paper, and also that of [22, 1, 8], is as follows: Express the increase in potential as a linear combination of the payoff values, and conclude Theorem 5 via an anti-concentration inequality and a union bound. Each step of the BRA is some player u, deviating to a new $\sigma \in [k]$, denoted as the (player,strategy) pair (u, σ). Thus, an execution of the BRA is fully specified by a sequence of pairs $S = (u_1, \sigma_1), (u_2, \sigma_2), \dots,$ along with an initial strategy vector $\boldsymbol{\sigma}^0 \in [k]^n$. The strategy profile at time t is given by $\boldsymbol{\sigma}^t := (\sigma_t, \boldsymbol{\sigma}^{t-1}_{-u_t})$.

We wish to control the value of $\Phi(\boldsymbol{\sigma}^t) - \Phi(\boldsymbol{\sigma}^{t-1})$ as a function of the payoff values. To this end, define the *potential-change indicator matrix* for a BR sequence as follows:

▶ **Definition 6.** *For any fixed BR sequence S of length ℓ, let $L(S, \boldsymbol{\sigma}^0) := \{\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \dots, \boldsymbol{\lambda}_\ell\}$, where $\boldsymbol{\lambda}_t \in \{-1, 0, 1\}^{(|E|\times k^2)}$, for all t. The entries of $\boldsymbol{\lambda}_t$ are indexed by indices of payoff matrix entries, denoted $((v, i)(w, j))$. The values of its entries are chosen as follows:*

$$\boldsymbol{\lambda}_t((v,i)(w,j)) := \begin{cases} 1 & \textit{if:} \quad u_t \in \{v,w\} \quad \textit{and} \quad \sigma^t_v = i \quad \textit{and} \quad \sigma^t_w = j. \\ -1 & \textit{if:} \quad u_t \in \{v,w\} \quad \textit{and} \quad \sigma^{t-1}_v = i \quad \textit{and} \quad \sigma^{t-1}_w = j. \\ 0 & \textit{otherwise.} \end{cases}$$

Each entry denotes whether the payoff values remain in the payoff (0), get added to the total payoff $(+1)$, or removed (-1). This set of vectors – or equivalently the matrix whose rows consist of the $\boldsymbol{\lambda}_t$'s – is termed the potential-change indicator matrix of a sequence.

The arguments $S, \boldsymbol{\sigma}^0$ are omitted if they are clear from context. Observe $\Phi(\boldsymbol{\sigma}^t) - \Phi(\boldsymbol{\sigma}^{t-1}) = \langle \boldsymbol{\lambda}_t, A\rangle$, where A is the vector of payoff values, so the product LA represents the sequence of changes in Φ along an execution of the BRA. Bounding the probability of $LA \notin [0, \epsilon]^\ell$ for all sequences of length $\ell \geq 2nk$ then implies Theorem 5. We use the following.

▶ **Lemma 7** ([33])**.** *Let $X \in \mathbb{R}^d$ be a random vector such that the joint probability on any $a \leq d$ coordinates of X is upper-bounded by ϕ^a at all points. Let M be a rank r matrix in $\eta\mathbb{Z}^{\ell\times d}$, i.e. all entries are multiples of η. Then the joint density of the vector MX is bounded by $(\phi/\eta)^r$, and for any given $b_1, b_2, \dots, b_\ell \in \mathbb{R}$ and $\epsilon > 0$,*

$$\Pr\Big[MX \in [b_1, b_1+\epsilon] \times \cdots \times [b_\ell, b_\ell + \epsilon]\Big] \leq (\phi\epsilon/\eta)^r \tag{2}$$

Observe that if X is a vector whose entries are independently distributed, and each X_i has probability density bounded by ϕ, then the joint distribution over any ℓ coordinates of X has density bounded by ϕ^ℓ. This statement is a generalization of a lemma from [33], but the proof therein easily extends.

To conclude Theorem 5, we first show that L has large enough rank, then apply the above lemma with $M = L(S, \boldsymbol{\sigma}^0)$ and $X = A$, and finally take a union bound over the choice of S and $\boldsymbol{\sigma}^0$. The following parameters are introduced for clarity of exposition.

▶ **Definition 8** (Active, Inactive, Repeating, and Non-Repeating players.). *Let S be a BR sequence, then player u is said to be* active *if she appears in the sequence, and* inactive *otherwise. An active player u is said to be* repeating *if there exists some strategy i such that (u,i) appears at least twice in S, or if (u,σ_u^0) appears in S at all. An active player who is not repeating is said to be* non-repeating. *We introduce the following notation:*

$p(S)$	*number of active players in S,*	$d(S)$	*number of distinct (u,i) moves in S,*
$p_1(S)$	*num. non-repeating players in S,*	$d_1(S)$	*distinct moves by non-rept. players in S,*
$p_2(S)$	*number of repeating players in S,*	$q_0(S)$	*number of distinct (u,σ_u^0) moves in S.*

Observe that $p = p_1 + p_2$, $k \cdot p \geq d \geq p$, $k \cdot p_1 \geq d_1 \geq p_1$, and $q_0 \leq p_2$. We will often use the quantity $d(S) - q_0(S)$, which is the number of "new" strategies played by the players.

3.1 Inactive Players, Rank Bounds, and Union Bounds

As discussed above, the goal is to show that $L(S)$ has large rank, and apply Lemma 7, taking a union bound over all possible choices of S and $\boldsymbol{\sigma}^0$. Naïvely, there are $k^n(nk)^\ell$ choices for sequences of length ℓ. However, if $p(S) \ll n$, the rank bound cannot exceed $d(S) \leq k \cdot p(S)$ in our model, which does not match the union bound. To circumvent this issue, we modify the matrix L in two different ways, for the cases $p_1(S) \geq p_2(S)$ and $p_2(S) \geq p_1(S)$, respectively. This case analysis is similar to the proofs in [1, 8], however these two papers each use only one of the two constructions, whereas our analyses require both, as the large strategy space allows for more complex interactions between the rows of L, and each construction only allows bounds of one kind.

3.1.1 Control by rounding.

The first modification to L builds on a technique of [1]. While the construction is valid for arbitrary graphs, the rank bounds hold only for complete graphs. Let $V_0 \subset V$ be the set of inactive players, and V_1 the active players. For any $u \in V_1$ and i fixed, all $((u,i)(w,\sigma_w^0))$ rows of L for $w \in V_0$ are identical, up to flipping a row's signs, since w's strategy remains unchanged. Therefore, in the inner product $\langle \boldsymbol{\lambda}_t, A\rangle$, these $((u,i),(w,\sigma_v^0))$ terms are added or subtracted together, and we may simply "guess" this value approximately, and take a union bound on the guesses, rather than guessing strategy choices. This is formalized below.

The quantity of interest is the change in $\Phi(\boldsymbol{\sigma}^t)$ over t, so it is equivalent to instead consider $\Phi(\boldsymbol{\sigma}^t) - \Phi(\boldsymbol{\sigma}^0)$, a constant shift. Let $\widetilde{A}_{uv}^{t-0} = A((u,\boldsymbol{\sigma}_u^t)(v,\boldsymbol{\sigma}_v^t)) - A((u,\boldsymbol{\sigma}_u^0)(v,\boldsymbol{\sigma}_v^0))$.

$$\begin{aligned}\Phi(\boldsymbol{\sigma}^t) - \Phi(\boldsymbol{\sigma}^0) &= \sum_{u,v\in V} A((u,\boldsymbol{\sigma}_u^t)(v,\boldsymbol{\sigma}_v^t)) - A((u,\boldsymbol{\sigma}_u^0)(v,\boldsymbol{\sigma}_v^0))\\ &= \sum_{u,v\in V_1} \widetilde{A}_{uv}^{t-0} + \sum_{w,w'\in V_0} \widetilde{A}_{ww'}^{t-0} + \sum_{u\in V_1}\sum_{w\in V_0} \widetilde{A}_{uw}^{t-0}\end{aligned}$$

For $w \in V_0$, $\boldsymbol{\sigma}_w^t = \boldsymbol{\sigma}_w^0$, so the "$\widetilde{A}_{ww'}^{t-0}$" terms are 0. Furthermore, the rightmost terms are in fact constants, depending only on σ_u. Let then $C(u,\sigma) := \sum_{w\in V_0} A((u,\sigma)(w,\boldsymbol{\sigma}_w^0)) - A((u,\boldsymbol{\sigma}_u^0)(w,\boldsymbol{\sigma}_w^0))$. Note $C(u,\boldsymbol{\sigma}_u^0) = 0$. The above sum is therefore equivalent to

$$\Phi(\boldsymbol{\sigma}^t) - \Phi(\boldsymbol{\sigma}^0) \;=\; \sum_{u,v\in V_1} \widetilde{A}_{uv}^{t-0} \;+\; 0 \;+\; \sum_{u\in V_1} C(u,\sigma_u^t)$$

For $p(S)$ fixed, we need only take a union bound over the nk^ℓ choices of BR sequence, $k^{p(S)}$ choices of initial strategy, and the $d(S) - q_0(S)$ different C values for each $(u, \sigma \neq \sigma_u^0)$. We will round these to the nearest multiple of ϵ, leaving $2n/\epsilon$ choices for each. With this, we have a union bound of size $k^{p(S)}(nk)^\ell(2n/\epsilon)^{d(S)-q_0(S)}$.

Critical Subsequences and Rank Bounds. It remains to argue that $L(S, \boldsymbol{\sigma}^0)$ has sufficiently large rank bounds. The statements below are presented informally, in the interest of space. See the full version for complete details [10]. To this end, we define *critical subsequences* as follows, based on the definition of critical block from [1]:

▶ **Definition 9** (Critical Subsequence). *Let B be a contiguous subsequence of S. If $\ell(B) \geq 2\big(d(B) - q_0(B)\big)$, but for every $B' \subsetneq B$, $\ell(B') < 2\big(d(B') - q_0(B')\big)$, we say that B is* critical*. Note that a return move – i.e. q_0-type move – for a subsequence B which starts at time t_B is a move $(u, \boldsymbol{\sigma}_u^{t_B})$, as opposed to a $(u, \boldsymbol{\sigma}_u^0)$ move.*

Observe that for any BR sequence S of length $2nk$, $\ell(S) \geq 2(d(S) - q_0(S))$, and so a maximal (up-to-inclusion) subsequence satisfying $\ell \leq 2(d - q_0)$ will be critical. Thus, any sequence of length $2nk$ must contain a critical subsequence B, with $\ell(B) = 2(d(B) - q_0(B))$. Critical subsequences are key to the analysis, since their length and number of distinct moves are perfectly correlated. This will be used in conjunction with the notion of *separated blocks*.

▶ **Definition 10** (Separated Blocks). *Fix a BR sequence S. For any active, non-repeating, player u, let T_u be the set of indices where the moving player is u, and $T := \{t_1 < t_2 < \cdots < t_m\}$ be the union of the T_u's. Note that $|T| = d_1(S)$. Let S_i for $0 \leq i \leq m$ be the subsequences of S from time t_i to t_{i+1} excluding boundaries, respectively, where $t_0 = 0$ and $t_{m+1} = |S|$. We say these S_i's are the* separated blocks *of S.*

The separated blocks allow us to isolate large-rank submatrices and combine their ranks. Let w be any inactive player, and (u_i, σ_i) be the non-repeating move which begins separated block i for all i. Then, up to column permutations, the submatrix with rows from T and columns indexed by $((u_i, \sigma_i)(w, \sigma_w))$ is upper-triangular with a non-zero diagonal. The diagonal is necessarily nonzero because the graph is complete. Furthermore, for a separated block S' which begins at move (u_i, σ_i), the submatrix of rows from S' and columns $((v, \sigma_v)(u_i, \sigma_i))$ also form an upper-triangular matrix with non-zero diagonals, where the columns are sorted by the first appearance of v, σ_v in S'. Finally, if we sort the rows of T first, then by chronological order of separated blocks, we can form a block-upper-triangular submatrix with non-zero diagonal terms, of size $d_1(S) + \sum_{S' \text{ separated}} (d(S') - q_0(S'))$. As above, graph completeness is necessary for the diagonal to be nonzero. This submatrix is block-upper-triangular because $((\cdot, \cdot), (u_i, \sigma_i))$ rows are all 0 before u_i first enters σ_i, since u_i is non-repeating. Assuming that the overall sequence S is critical, this gives

$$rank(L) \;\geq\; d_1(S) + \sum_{S' \text{ sep.}} d(S') - q_0(S') \;>\; \tfrac{1}{2} d_1(S) + \tfrac{1}{2} d_1(S) + \sum_{S' \text{ sep.}} \tfrac{1}{2} \ell(S')$$

However, $\ell(S) = d_1(S) + \sum_{S' \text{ sep.}} \ell(S')$, so we have $rank(L) \geq \frac{1}{2} d_1(S) + \frac{1}{2}\ell(S)$. By criticality, $\ell(S) = 2(d(S) - q_0(S))$, which gives a rank bound of $\frac{1}{2}(d_1(S) + \ell(S)) = \frac{1}{2} d_1(S) + d(S) - q_0(S)$.

Combining Rank and Union bounds. Recall that we have shown, for any BR sequence of length $2nk$, it must contain a critical subsequence B. Assuming B has length ℓ, there are $k^{p(B)}(nk)^\ell$ choices for the (modified) matrix $L(B, \boldsymbol{\sigma}^0)$, and $(4n/\epsilon)^{d(B) - q_0(B)}$ choices for the approximate $C(u, \sigma)$ values. The $C(u, \sigma)$'s are rounded to the nearest ϵ, so the true improvements lie in $(0, \epsilon)$ if the approximated improvements lie in $(-\epsilon, 2\epsilon)$, since there can be at most two C terms in each change, and rounding introduces error at most $\epsilon/2$.

We have shown that $L(B, \boldsymbol{\sigma}^0)$ has rank at least $\frac{1}{2} d_1(B) + d(B) - q_0(B)$. Applying Lemma 7, we conclude that the probability that every critical subsequence of a valid BR sequence of length at least $2nk$ has all improvements in $(0, \epsilon)$ is upper-bounded by

$k^{p(B)}(nk)^{\ell}(4n/\epsilon)^{d(B)-q_0(B)}(3\phi\epsilon)^{d_1(B)/2+d(B)-q_0(B)}$. Recall $d_1(B) \geq p_1(B)$, and assume $p_1(B) \geq p_2(B) \implies d_1(B) \geq p_1(B) \geq \frac{1}{2}p(B)$. Also, $d(B) - q_0(B) \leq k \cdot p(B)$, and recall $\ell = 2(d(B) - q_0(B))$. Thus the probability that all improvements are small, for complete graphs, when $p_1(B) \geq p_2(B)$ is at most $\left((20n^3k^3\phi^2)^k\epsilon^{1/4}\right)^{p(B)}$. We later combine this bound with one for the case $p_2(B) \geq p_1(B)$, and take a union bound over the choice of $p(B)$.

3.1.2 Control by cyclic sums

The second rank bound argument is more intricate, and is loosely based on a construction in [22]. The bounds proved here hold for arbitrary graphs. We construct a new matrix Q whose columns lie in the span of L, which cancels the contributions of inactive players, but still captures improvement. Let (u, σ) be some move which appears twice in S, or suppose some (u, σ_u^0) appears in S. Let τ_0 be the index of the first occurrence of (u, σ) in the BR sequence ($\tau_0 = 0$ in the latter case), and let $\tau_1, \tau_2, \ldots$ be all subsequent appearances of $(u, \cdot)$ in the sequence. Suppose τ_m is the second occurrence of (u, σ) in the BR sequence. Then we let $\boldsymbol{q}_{u,\sigma} := \sum_{j=1}^{m} \boldsymbol{\lambda}_{\tau_j}$, noting that the τ_0 is omitted. Let $Q(S, \boldsymbol{\sigma}^0)$ be the matrix whose columns consist of the $\boldsymbol{q}_{u,\sigma}$'s. Observe, for any inactive player w,

$$\boldsymbol{q}_{u,\sigma}\big((u,\sigma')(w,\sigma_w^0)\big) \;=\; \sum_{j=1}^{m} \boldsymbol{\lambda}_{\tau_j}\big((u,\sigma')(w,\sigma_w^0)\big) \;=\; \sum_{j=1}^{m} \mathbb{I}[\sigma_u^{\tau_j} = \sigma'] - \mathbb{I}[\sigma_u^{\tau_{j-1}} = \sigma'] \;=\; 0$$

Thus, the $\boldsymbol{q}_{u,\sigma}$'s are all 0 in entries corresponding to edges with inactive players, and therefore the Q matrix does not depend on the initial strategies of the inactive players. Thus, to union bound the possible Q matrices, it suffices to fix the initial strategy of only the active players and the BR sequence. Furthermore, $L \cdot A \in [0, \epsilon]^{\ell} \implies Q \cdot A \in [0, \ell\epsilon]^{d-d_1}$, so it suffices to apply Lemma 7 on the matrix Q, with only polynomial blowup, since $\ell \leq O(nk)$.

Fixing a critical subsequence B, there are at most $(nk)^{\ell(B)}$ choices of the BR sequence, and $k^{p(B)}$ choices of initial strategy profiles for the active players.

Rank Bounds. It remains to show that Q has large rank relative to this smaller union bound. Since we have eliminated inactive players, the high-rank submatrix will consist of interactions between active players. As in [22], construct an auxiliary *directed* graph $G' = (V, E')$, where V is the set of players, and E' is constructed as follows: for any repeating move (u, σ), add the directed edge (u, w) for *every* $w \in V$, such that $\exists \sigma' \in [k] : \boldsymbol{q}_{u,\sigma}\big((u,\sigma)(w,\sigma')\big) \neq 0$. Note that u and w must be connected in the game graph, and we do not need completeness.

Let $P_2 \subseteq V$ be the set of repeating players, and note that they all have non-zero out-degree, since $\boldsymbol{q}_{u,\sigma} \neq \boldsymbol{0}$ as this would contradict the fact that the sequence is improving. Repeatedly perform the following: choose $r_1 \in P_2$, and let T_1 be the BFS arborescence rooted at r_1 which spans all nodes reachable from r_1 in G'. Delete $V(T_1)$ from G' and repeat, picking an arbitrary root vertex $r_2 \in P_2 \setminus V(T_1)$. Stop when every vertex of P_2 is covered by some arborescence. For all i, let T_i^0 and T_i^1 be a partition of the nodes of T_i which are of even or odd distance from the root, respectively. Let P_i' be the larger of $V(T_i^0) \cap P_2$ and $V(T_i^1) \cap P_2$, and let $P_2' := \bigcup_{i=1}^{\infty} P_i'$, noting that $|P_2'| \geq |P_2|/2 = p_2(S)/2$. We claim that the collection $\mathcal{V} := \{\boldsymbol{q}_{u,\cdot} : u \in P_2'\}$ is linearly independent.

For every $u \in P_2'$, either u was an internal node to the arborescence that covers it, or a leaf. If it is internal, we may associate to u any of its children in the arborescence, which are not contained in P_2', by definition. Let w be the chosen child, then the only $\mathcal{V}$ vectors which have nonzero $((u, \cdot)(w, \cdot))$ entries are the $\boldsymbol{q}_{u,\cdot}$ vectors.

Conversely if u is a leaf, then its out-neighbours must be in previously constructed arborescences. Let w be any such neighbour, then $\boldsymbol{q}_{w,\cdot}$ can not contain a non-zero $((u,\cdot)(w,\cdot))$ entry, as otherwise u would have been in w's arborescence. Therefore, $\boldsymbol{q}_{u,\cdot}$ is the only vector in $\mathcal{V}$ to contain a nonzero $((u,\cdot)(w,\cdot))$ entry. Observe that these matrix entries are necessarily nonzero even if the graph is not complete. The above discussion demonstrates a $|\mathcal{V}| \times |\mathcal{V}|$ diagonal submatrix of $\mathcal{V}$, which implies Q has rank at least $|\mathcal{V}| \geq p_2(S)/2$.

Combining Rank and Union Bounds. This section considers the complete graph case. These bounds will be extended to general graphs in the next section. Section 3.1.1 gives bounds for the case $p_1(B) \geq p_2(B)$, and so we restrict our attention to the case $p_2(B) \geq p_1(B) \implies p_2(B) \geq p(B)/2$.

For any fixed critical subsequence B, we have argued that the probability that $L(B, \boldsymbol{\sigma}^0)$ has all entries in $(0, \epsilon)$ is at most the probability that $Q(B, \boldsymbol{\sigma}^0)$ has all entries in $(0, \ell\epsilon)$. By Lemma 7 and the above rank bounds, this occurs with probability at most $(\ell\phi\epsilon)^{p_2(B)/2}$. Recall that by criticality, $\ell = 2(d(B) - q_0(B)) \leq 2k \cdot p(B)$. Thus, with the above union bound, the probability that all critical subsequences with $p(B)$ active players and $p_2(B) \geq p_1(B)$ have bad improvements is at most $k^{p(B)}(nk)^{2kp(B)}(2k \cdot p(B) \cdot \epsilon\phi)^{p(B)/4} \leq \big(2(nk)^{2k}k^{5/4}(n\phi\epsilon)^{1/4}\big)^{p(S)}$.

3.2 (Quasi-)Polynomial Smoothed Complexity for NetCoordNash

Complete Game Graphs. From Sections, 3.1.1 and 3.1.2, the probability that *all* valid BR sequences of length at least $2nk$ have all improvements in $(0, \epsilon)$ is at most

$$\sum_{p=1}^{n} \big((20n^3k^3\phi^2)^k \epsilon^{1/4}\big)^p + \big(2(nk)^{2k}(nk^5\phi\epsilon)^{1/4}\big)^p$$

by simply taking a union bound over the value of $p(B)$, and whether $p_1(B) \geq p_2(B)$ or the converse hold. Setting $\epsilon = (20\phi^2 n^3 k^3)^{-4k-4}$, we conclude that this probability is at most $1/(20\phi^2 n^3 k^4 - 1)$, which concludes the complete-graph portion of Theorem 5 noting that $\phi > \frac{1}{2}$. We omit calculation details and handling edge-cases where all players are active. These are included in the full version [10].

General Game Graphs. For general game graphs, we will use the cyclical-sum construction, as it does not require completeness. Note that it was phrased in terms of critical subsequences, but never used this property. However, the cyclic-sum construction only gives bounds in terms of $p_2(S)$. As in [22], we use the following lemma:

▶ **Lemma 11** ([22], Lemma 3.4)**.** *Any valid BR sequence of length $5nk$ contains a contiguous subsequence S' with $|S'|/(5\log(nk))$ repeating moves (pairs).*

This implies that any $5nk$-length BR sequence S contains a subsequence S' with $p_2(S') \geq |S'|/(5\log(nk))$. Therefore, the probability that all $5nk$-length sequences have improvements in $(0, \epsilon)$ is at most $\sum_{\ell=1}^{5nk} k^n (nk)^\ell (\phi\epsilon)^{\ell/10\log(nk)}$. Setting $\epsilon = \phi^{-1}(2n^2k^3)^{-20k\cdot\log(nk)}$, the probability is at most $1/(nk)^2$. This concludes the general-graph portion of Theorem 5. As above, the details of calculation are omitted.

Performance in Expectation. The performance analysis of BRA in expectation is discussed in the full-version. It is derived as a consequence of the with-high-probability bounds, integrating over the choice of ϵ.

4 Smoothness-Preserving Reduction to Local-Max-Cut and -Bisection

In this section, we refine standard Karp reductions to define *smoothness preserving reductions*, and outline the two reductions of Theorem 2. Recall from Section 2.2 that an algorithm is said to be smoothed-efficient if, on applying independent random perturbations to all inputs of an adversarially chosen instance, the algorithm runs in time polynomial in the input size and inverse perturbation size, with high probability.

▶ **Definition 12** (Strong and Weak Smoothness-Preserving Reductions). *A* weak (randomized) smoothness-preserving reduction *from a search problem $\mathcal{P}$ to problem $\mathcal{Q}$ is defined by poly-time computable functions f_1 and f_2, a full-row-rank, integer, matrix M with polynomially bounded entries, a constant $\eta \geq 1/poly$, and a real probability space $\Omega \subseteq \mathbb{R}^d$; such that the following holds: For any $I = (D, X) \in \mathcal{P}$ and $R \in \Omega$, $J = (f_1(D), \eta M(X \circ R))$ is an instance of $\mathcal{Q}$ whose solutions $\boldsymbol{\sigma}$ map to solutions $f_2(\boldsymbol{\sigma})$ for I. Here, $\circ$ denotes concatenation.*

We require that $|f_1(D)|$, the dimension of R, and the size of M, be polynomial in $|I|$; that the probability density of the entries of R be polynomial in $|I|$ and the density bound on X; and that the entries of R be independently distributed.

If M is a diagonal matrix, then this is a strong *smoothness-preserving reduction.*

The R variables may seem superfluous at first, but are included to ensure that M has full-rank. A key fact to the proof of Lemma 7, the anti-concentration bound at the heart of this paper and previous local-max-cut papers, is the following:

▶ **Proposition 13** ([33]). *Let $X \in \mathbb{R}^d$ be a random vector such that the joint probability on any $a \leq d$ coordinates of X is upper-bounded by ϕ^a at all points, and let $M \in \mathbb{R}^{\ell \times d}$ be full-rank, with entries which are multiples of η, for $\ell \leq d$. Then the random variable $Y := MX$ also has bounded joint density $f_Y(y) \leq (\phi/\eta)^\ell$ for all $y \in \mathbb{R}^d$.*

Thus, if the entries of X and R have bounded density, and $|\det(\eta M)| \geq \eta^d$, then the joint distribution on $M(X \circ R)$ has polynomially bounded density. A proof of this statement is provided in the full version of the paper [10].

When M is diagonal, the reduced instance trivially has independent entries, which is sufficient for most smoothed-analysis results to hold. Hence, strong reductions easily extend smoothed efficient algorithms. We conjecture that for many problems, upper-bounding the joint density of the input values suffices for smoothed-efficient algorithms to exist.

▶ **Lemma 14.** *(a) Suppose problem $\mathcal{Q}$ has (quasi-)polynomial smoothed complexity. If $\mathcal{P}$ admits a strong smoothness-preserving reduction to $\mathcal{Q}$, then $\mathcal{P}$ also has (quasi-)polynomial smoothed complexity. (b) If $\mathcal{Q}$ has a smoothed efficient algorithms when the input distribution has joint density bounds as in Proposition 13 and Lemma 7, then if $\mathcal{P}$ admits a weak smoothness-preserving reduction to $\mathcal{Q}$, $\mathcal{P}$ has (quasi-)polynomial smoothed complexity.*

The proof is straightforward modulo technicalities, and is included for completeness in the full version of this paper [10]. It is key that the matrix M has full (row) rank, since this ensures that the joint density on the reduced parameters is sufficiently bounded.

Observe that local-max-cut satisfies the conditions of part (b), since the proofs of [22, 1] simply apply Lemma 7 to the input, similarly to the argument in Section 3. Thus weak reductions to local-max-cut do imply smoothed efficient algorithms. This would not be an interesting notion of reduction if it only held for reductions to max-cut. As mentioned above, we conjecture that many smoothed-analysis results satisfy the conditions for part (b).

As an example, we note that the analyses for the smoothed efficient algorithms for a TSP 2-approximation [21] and for multidimensional bin-packing [28], are robust to this form of input assumption.

The smoothed complexity of local-max-bisection is open, but we believe that the natural local search procedure may admit a similar smoothed analysis to local-max-cut. This would imply a smoothed efficient algorithm for k-NetCoordNash for non-constant k.

4.1 Outline of Reductions

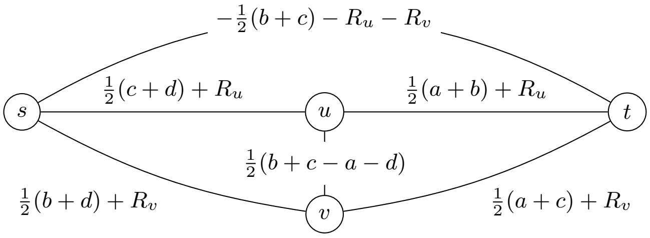
It suffices then, to provide weak smoothness-preserving reductions from NetCoordNash to local-max-cut and -bisection, as stated in Theorem 2. The main technical part involves the construction of M such that its rows are independent, and solutions of the resulting instance map to solutions of the original.

In this writeup, we give only a sketch of the reductions. The formal definitions and rank analyses may be found in the full version of this paper [10]. The reduction from 2-NetCoordNash to local-max-cut will be given first and in more detail, as it is cleaner, and is the basis for the reduction from k-NetCoordNash to local-max-bisection.

2-NetCoordNash reduces to Local-max-cut. Let $G = (V, E)$ be the game graph, with payoff vector A. Construct a weighted cut graph $H = (V', E')$ where $V' = V \cup \{s, t\}$, and $E' = E \cup \bigcup_{u \in V} \{su, ut\}$. We will define edge weights such that (1) every locally maximal cut is an s-t cut, and (2) the value of the cut (S, T) with $s \in S$ and $t \in T$ is equal to the total payoff of the game when $\sigma_u = 1$ if $u \in S$, and 2 if $u \in T$. Thus changing a player's strategy is equivalent to flipping its vertex across the cut. In the reverse direction, local-max-cuts are exactly the local maxima of the game's potential function, and thereby pure NE.

The following figure gives the edge weights for a small 2-player example which achieves the above properties, with the payoff matrix given as follows:

uv game payoffs: $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$



The general construction consists of placing copies of the above gadget on H for every game edge in E, taking the sum of the edge-weights for the su, ut, and st edges.

Observe that the above construction has edge weights which are linear combinations of the payoff values. Furthermore, for all values of R_u and R_v, the cut values are equal to payoff values. The R values are added only to increase the rank of the reduction matrix, and choosing them negative ensures that only s-t cuts are maximal. We show [10] by induction on $|V|$ that the matrix rows are independent, and so the reduction satisfies the necessary conditions. Note that the cut graph is complete if and only if the game graph is.

***k*-NetCoordNash reduces to Local-max-bisection.** The reductions from games with k strategies are not as straightforward. Let $G = (V, E)$ be the game graph with payoff vector A. We construct a weighted cut graph $H = (V', E')$ where $V' = (V \times [k]) \cup \{s_0, s_1, \ldots, s_{n(k-2)}, t\}$, the (player,strategy) pairs, and construct E' as follows: for every node (u, i) and $0 \leq a \leq n(k-2)$, we add an $\{s_a, (u, i)\}$ and $\{(u, i), t\}$ edge; for every $u \in V$

and $i \neq j$, we add a $\{(u,i),(u,j)\}$ edge; and for every $uv \in E$ and $i,j \in [k]$, we add a $\{(u,i),(v,j)\}$ edge. Call a cut (S,T) *valid* if $s_\ell \in S$ for all ℓ, $t \in T$, and S contains exactly one (u,i) node for all $u \in V$. Note that all valid cuts are balanced.

By construction, there is a natural strategy profile associated with each valid cut, namely if node (u,i) is in S then $\sigma_u = i$. We wish to choose edge weights such that (1) all locally maximal bisections are valid, and (2) the cut value is equal to $\Phi(\boldsymbol{\sigma})$. (1) will be achieved by giving low weight to the $\{(u,i),(u,j)\}$ edges, and higher weight to the $\{s_a,(u,i)\}$ edges, using the extra randomness available. This respectively ensures that it is always in our interest to have a small number of $(u,\cdot)$ nodes in S, but not none. As above, we will introduce extra randomness to the edge weights to ensure that M is full-rank. In this case, we will show M is full rank by arguing that it is upper-triangular after basic row operations.

The cut graph is again complete if and only if the game graph is, and thereby we have shown the second part of Theorem 2. The edge weights are not given directly, but are instead found by solving for the total cut values. As it would not be of any value to the reader to be given the values of the edge-weights without the full exposition, the details are left to the full version of this paper [10].

References

1 Omer Angel, Sébastien Bubeck, Yuval Peres, and Fan Wei. Local max-cut in smoothed polynomial time. In *ACM Symposium on Theory of Computing*, pages 429–437, 2017.

2 Elliot Anshelevich, Anirban Dasgupta, Eva Tardos, and Tom Wexler. Near-optimal network design with selfish agents. In *ACM Symposium on Theory of Computing*, pages 511–520, 2003.

3 Radhika Arava. Social Network Analysis Using Coordination Games. *arXiv preprint*, 2017. `arXiv:1708.09570`.

4 D. Arthur and S. Vassilvitskii. Worst-Case and Smoothed Analysis of the ICP Algorithm, with an Application to the k-Means Method. *SIAM J. on Computing*, 39(2):766–782, 2009.

5 Imre Bárány, Santosh Vempala, and Adrian Vetta. Nash equilibria in random games. *Random Struct. Algorithms*, 31(4):391–405, December 2007.

6 Paul Beame, Stephen Cook, Jeff Edmonds, Russell Impagliazzo, and Toniann Pitassi. The Relative Complexity of NP Search Problems. *J. Comput. Sys. Sci.*, 57(1):3–19, 1998.

7 Rene Beier and Berthold Vöcking. Random Knapsack in Expected Polynomial Time. *J. Comput. Syst. Sci.*, 69(3):306–329, November 2004.

8 Ali Bibak, Charles Carlson, and Karthekeyan Chandrasekaran. Improving the smoothed complexity of FLIP for max cut problems. In *ACM-SIAM SODA*, pages 897–916, 2019.

9 N. Bitansky, O. Paneth, and A. Rosen. On the Cryptographic Hardness of Finding a Nash Equilibrium. In *IEEE FOCS*, pages 1480–1498, 2015.

10 Shant Boodaghians, Rucha Kulkarni, and Ruta Mehta. Nash Equilibrium in Smoothed Polynomial Time for Network Coordination Games. *arXiv preprint*, 2018. `arXiv:1809.02280`.

11 Joris Broere, Vincent Buskens, Jeroen Weesie, and Henk Stoof. Network effects on coordination in asymmetric games. *Scientific reports*, 7(1):17016, 2017.

12 Yang Cai and Constantinos Daskalakis. On minmax theorems for multiplayer games. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 217–234, 2011.

13 X. Chen, X. Deng, and S.-H. Teng. Computing Nash Equilibria: Approximation and Smoothed Complexity. In *IEEE Symposium on Foundations of Computer Science*, pages 603–612, 2006.

14 Syngjoo Choi, Douglas Gale, Shachar Kariv, and Thomas Palfrey. Network architecture, salience and coordination. *Games and Economic Behavior*, 73(1):76–90, 2011.

15 Bruno Codenotti, Stefano De Rossi, and Marino Pagan. An Experimental Analysis of Lemke-Howson Algorithm. arXiv preprint, 2008. `arXiv:0811.3247`.

16 Valentina Damerow, Friedhelm Meyer auf der Heide, Harald Räcke, Christian Scheideler, and Christian Sohler. Smoothed Motion Complexity. In *Algorithms – ESA*, pages 161–171, 2003.

17 C. Daskalakis, P. Goldberg, and C. Papadimitriou. The Complexity of Computing a Nash Equilibrium. *SIAM Journal on Computing*, 39(1):195–259, 2009. doi:10.1137/070699652.

18 Argyrios Deligkas, John Fearnley, Tobenna Peter Igwe, and Rahul Savani. An Empirical Study on Computing Equilibria in Polymatrix Games. In *AAMAS*, pages 186–195, 2016.

19 Stéphane Durand and Bruno Gaujal. Complexity and Optimality of the Best Response Algorithm in Random Potential Games. In *Algorithmic Game Theory*, pages 40–51, 2016.

20 Glenn Ellison. Learning, local interaction, and coordination. *Econometrica: Journal of the Econometric Society*, pages 1047–1071, 1993.

21 Matthias Englert, Heiko Röglin, and Berthold Vöcking. Worst Case and Probabilistic Analysis of the 2-Opt Algorithm for the TSP. *Algorithmica*, 68(1):190–264, 2014.

22 Michael Etscheid and Heiko Röglin. Smoothed analysis of local search for the maximum-cut problem. *ACM Transactions on Algorithms (TALG)*, 13(2):25, 2017.

23 Alex Fabrikant, Christos Papadimitriou, and Kunal Talwar. The Complexity of Pure Nash Equilibria. In *ACM Symposium on Theory of Computing*, pages 604–612, 2004.

24 Michal Feldman and Tami Tamir. Conflicting congestion effects in resource allocation games. *Operations research*, 60(3):529–540, 2012.

25 Tobias Harks, Martin Hoefer, Max Klimm, and Alexander Skopalik. Computing pure Nash and strong equilibria in bottleneck congestion games. *Math. Prog.*, 141(1):193–215, 2013.

26 Ramesh Johari and John N Tsitsiklis. Efficiency loss in a network resource allocation game. *Mathematics of Operations Research*, 29(3):407–435, 2004.

27 David S. Johnson, Christos H. Papadimitriou, and Mihalis Yannakakis. How easy is local search? *Journal of Computer and System Sciences*, 37(1):79–100, 1988.

28 David Karger and Krzysztof Onak. Polynomial approximation schemes for smoothed and random instances of multidimensional packing problems. In *ACM-SIAM Symposium on Discrete Algorithms*, volume 7, pages 1207–1216, 2007.

29 Dharshana Kasthurirathna, Mahendra Piraveenan, and Michael Harré. Influence of topology in the evolution of coordination in complex networks under information diffusion constraints. *The European Physical Journal B*, 87(1):3, 2014.

30 Bodo Manthey and Rüdiger Reischuk. Smoothed Analysis of Binary Search Trees. In *Algorithms and Computation*, pages 483–492, 2005.

31 Dov Monderer and Lloyd S Shapley. Potential games. *Games and economic behavior*, 14(1):124–143, 1996.

32 Andrea Montanari and Amin Saberi. The spread of innovations in social networks. *National Academy of Sciences*, 107(47):20196–20201, 2010.

33 Heiko Röglin. The Complexity of Nash Equilibria, Local Optima, and Pareto-Optimal Solutions. *Fakultät für Math., Informatik und Naturw. der RWTH*, 2008.

34 Heiko Röglin and Berthold Vöcking. Smoothed analysis of integer programming. *Mathematical Programming*, 110(1):21–56, June 2007.

35 Robert W. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International J. of Game Theory*, 2(1):65–67, 1973.

36 T. Roughgarden and É. Tardos. How Bad is Selfish Routing? *J. ACM*, 49(2):236–259, 2002.

37 Tim Roughgarden. Routing Games, 2007.

38 Aviad Rubinstein. Settling the Complexity of Computing Approximate Two-player Nash Equilibria. *SIGecom Exch.*, 15(2):45–49, February 2017.

39 Rahul Savani and Bernhard von Stengel. Exponentially Many Steps for Finding a Nash Equilibrium in a Bimatrix Game. In *IEEE FOCS*, pages 258–267, 2004.

40 Alejandro A. Schäffer and Mihalis Yannakakis. Simple Local Search Problems That Are Hard to Solve. *SIAM J. Comput.*, 20(1):56–87, February 1991.

41 Daniel A. Spielman and Shang-Hua Teng. Smoothed Analysis of Algorithms: Why the Simplex Algorithm Usually Takes Polynomial Time. *J. ACM*, 51(3):385–463, May 2004.