Smoothed Efficient Algorithms and Reductions for Network Coordination Games

Submission 163

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.

Second, 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 (with arbitrary k) to local-max-bisection. The former together with the recent result of Bibak, Chandrasekaran, and Carlson (SODA '18) gives an alternate $O(n^8)$-time smoothed algorithm for the 2-strategy case. This notion of reduction allows for the extension of smoothed efficient algorithms from one problem to another.

For the first set of results, we develop techniques to bound the probability that an (adversarial) better-response sequence makes slow improvements on the potential. Our approach combines and generalizes the local-max-cut approaches of Etscheid and Röglin (ACM TALG, 2017) 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.

Contents

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 networks, biological networks, routing and congestion on roads etc. [Ara17, MS10, KPH14, Smi79, Rou07], and have been extensively studied in various areas like Game theory and economics, Learning, Networks etc [HHKS13, BBWS17, Ell93, CGKP11, ADTW03].

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) [CD11]. Although it is widely conjectured that PLS is unlikely to lie in P [BCE$^+$98, BPR15, Rub17], problems in this class admit local-search algorithms [JPY88], which have been observed to be empirically fast [JPY88, CDRP08, DFIS16], but requiring exponential time in the worst case [SY91, SvS04]. 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," [ST04] (see Section 6 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 the inputs are randomly perturbed–this is one of the strongest guarantees one can hope for beyond worst case. The above gives rise to the following question:

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 [FPT04].

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 [ER17], and then polynomial for complete graphs [ABPW17]. Both results and a recent (simultaneous) work [BCC19] follow a common high-level framework. 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 design novel techniques to obtain the appropriate bounds

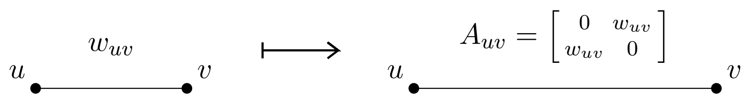


Figure 1: Local-max-cut to 2-strategy network coordination games: mapping of edge (u, v).

and combine them.

Furthermore, to the knowledge of the authors, no notion of smoothness-preserving reduction has been shown in the past, and we believe that such reductions are of independent interest.

1.1 Our Results

A network coordination game is represented by an undirected *game graph* $G = (V, E)$, where the nodes represent players, and each player $v \in V$ simultaneously plays a two-player coordination game with each of its neighbors. If every player has k strategies to choose from, then the game on each edge (u, v) can be represented by a $k \times k$ payoff matrix A_{uv}. Once every player chooses a strategy, the payoff value for each edge is fixed, and each player gets the sum of the payoffs on its incident edges. The goal is to find a PNE of this game. We will show that the natural *better-response algorithm* converges quickly with high probability for a perturbed instance.

Smoothed Analysis of NetCoordNash. A choice of strategies is at equilibrium if no player can gain by deviating unilaterally. Better-response dynamics/algorithms are an iterative procedure where any player who can gain by changing strategy, does so, one at a time until an equilibrium is reached. Such a procedure need not converge in general[1]. In our setting, however, the sum of payoffs of all players acts as a *potential function*, measuring the progress of this algorithm (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 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. Formally, we show the following:

Theorem 1.1. *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 will converge to a PNE in at most $(nk\phi)^{O(k)}$ steps, and the expected maximum number of steps of any valid execution is polynomial in ϕ^k and n^k.*

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 maximum number of steps of any valid execution is polynomial in ϕ and $n^{k \log(nk)}$.

An outline of the proof is given in Section 3.1, and the missing details in Section 4. 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 [ST04], local-max-cut[ABPW17], etc., require this.

The above performance guarantees are only (quasi-)polynomial in the input size for k fixed. We leave it as an open problem to improve this. 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 3.1).

[1]for instance, consider zero-sum games where the total sum of payoffs is zero for any choice of strategies.

Smoothness-Preserving Reductions. Note that standard Karp reductions do not suffice to extend a smoothed efficient algorithm from one problem to another. 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 1.2. *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 3.2, and a formal statement of the local-max-cut and -bisection problems is in Section 2.2. An outline of the proof is given in Section 3.2, and the missing details in Section 5. The first reduction, together with smoothed efficient algorithms for local-max-cut, gives alternate smoothed efficient algorithms for the $k = 2$ instance; in particular, the recent result of [BCC19] gives 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 related to ours are [ER17] and [ABPW17], who first analyzed the smoothed complexity of local-max-cut, and [BCC19] who refined the analysis of the latter. As discussed earlier local-max-cut is a special case of NetCoordNash, therefore techniques of the former do not directly apply. Independently, [BCC19] 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 we note that our result does not subsume theirs as the reduction is not *smoothness preserving*.

Beyond-worst case complexity of NE computation. For Nash equilibrium (NE) computation, the smoothed complexity of two-player games is known to not lie in P unless RP = PPAD [CDT06a], which follows from the hardness of (1/poly)-additive approximation. On the contrary, for most PLS-complete problems, the natural local-search algorithm often finds an additive approximate solution efficiently. There is always a "potential function" that the algorithm improves in each step. Intuitively, until an approximate solution is reached, the algorithm will improve the associated potential function significantly in every local-search step.

Towards average case analysis, Bárány, Vempala, and Vetta [BVV07] showed that a game picked uniformly at random has a NE with support size 2 for both the players with high 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 [DG16].

Due to space constraints, we list broader related work on smoothed analysis in general, and worst-case results for Nash equilibrium computation in Section 6, following the technical discussion.

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, can be represented by two payoff matrices A and B. It is without loss of generality to assume 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. We term it a *k-network coordination game* if each player has k strategies. For disambiguation, we will 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, we assume without loss of generality that every entry of A is contained in $[-1, 1]$.

Potential Function. We will see below that we may restrict our attention to pure strategies, *i.e.* no randomization by the players. Let n be the number of players; a *strategy profile* is a vector $\boldsymbol{\sigma} \in [k]^n$, implicitly 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 [CD11]: if player u changes their strategy, $\Phi(\boldsymbol{\sigma})$ and $\text{payoff}_u(\boldsymbol{\sigma})$ change 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 [Ros73].

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

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

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 change in strategies 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) consists of repeatedly making 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 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 [ST04] 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.

In Section 3.2, we will define a notion of reduction — not unlike Karp reductions — which allow smoothed complexity results to be translated from one problem to another. Note that standard Karp reductions do not achieve this, as they do not give any guarantees on the distribution of the reduced variables when in the input variables are randomly perturbed. We provide here a formal definition of problems which admit smoothed efficient algorithms.

Definition 2.1 (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.

The intuition of this notion is that X must be "spread" by at least $1/\phi$ in each point, and so we define running-time bounds as a function of ϕ. A fact that will be of interest, and which is central to the proof of the anti-concentration bound at the heart of this paper and previous local-max-cut papers, is that composing random variables with integer matrices will only reduce the joint density.

Proposition 2.1 ([Rög08])**.** *Let $X \in \mathbb{R}^d$ be a random vector such that the joint probability on any $a \le d$ 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 $\ell \le d$. Then the random variable $Y := MX$ also has bounded joint density $f_Y(y) \le (\phi/\eta)^\ell$ for all $y \in \mathbb{R}^d$.*

A proof of this proposition is given in Section 5.1. We continue with a formal definition of polynomial smoothed complexity in our setting.

Definition 2.2 (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, as in the previous definition,

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

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

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 [SY91]. 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 [SY91], and the smoothed complexity of local-max-cut has been studied at length, as discussed in the introduction.

3 High-Level Proof Ideas for Theorems 1.1 and 1.2

In this section, we give the high-level structure for the proofs of our main results. The remaining sections of the paper provide the missing details. We first outline the analysis of the BRA local search procedure (Theorem 1.1), then outline the smoothness-preserving reductions (Theorem 1.2).

3.1 Theorem 1.1: Smoothed performance of the BRA

We begin with the proof of Theorem 1.1. Observe that in the theorem statement, the randomness is only on the values in the payoff matrices, and not on the choice of BR moves. These results hold if the BR moves are chosen *adversarially* in response payoffs. Recall that a profile $\boldsymbol{\sigma}$ is a PNE if and only if it is a local maximum of the potential Φ, and note that Φ may only take values in the interval $[-n^2, n^2]$, since the payoffs are in $[-1, 1]$. Therefore it suffices to show significant increase in Φ for *every* linear-length sequence of BR moves, with high probability. Formally,

Theorem 3.1. *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 $\epsilon' = (20\phi^2n^3k^3)^{-4k-4}$.*

This theorem, along with the above observations, implies that the BRA must terminate in $\phi(nk)^{O(k\log(nk))}$ steps with probability $1 - (nk)^{-2}$, and if the graph is complete, in $(\phi nk)^{O(k)}$ steps with probability $1 - (nk)^{-3}$. The results in expectation follow from the high-probability results, as proved in Section 4.5. Therefore, Theorem 1.1 follows from Theorem 3.1.

Following the pattern of [ER17, ABPW17, BCC19], we will begin by expressing the increase in potential as a linear combination of the payoff values, and reduce the proof of Theorem 3.1 to the application of an anti-concentration inequality and a union bound.

Each step of the BRA consists of some player, u, deviating from their previously chosen strategy to a new $\sigma \in [k]$, which we will denote as the (player,strategy) pair (u, σ). Thus, an execution of the BRA is fully specified by a sequence of pairs $S = (u_1, \sigma_1), (u_2, \sigma_2), \ldots$, 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 introduce next the *potential-change matrix* for a BR sequence, which allows us to control the value of $\Phi(\boldsymbol{\sigma}^t) - \Phi(\boldsymbol{\sigma}^{t-1})$ as a function of the payoff values.

Definition 3.1. For any fixed BR sequence S of length ℓ, we define the set of vectors $L(S, \boldsymbol{\sigma}^0) = \{\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \ldots, \boldsymbol{\lambda}_\ell\}$, 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:

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

That is, every entry signifies if the corresponding payoff value remains in consideration (0), gets added to the total payoff (+1), or removed (−1). We term this set of vectors, or equivalently the matrix whose columns consist of the $\boldsymbol{\lambda}_t$'s, as the *potential-change matrix* of a sequence.

The arguments $S, \boldsymbol{\sigma}^0$ will be 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 vector $(L \cdot A)$ represents the sequence of changes in Φ along an execution of the BRA. Theorem 3.1 is then equivalent to bounding the probability of $LA \notin [0, \epsilon]^\ell$ for all sequences of length $\ell \geq 2nk$. We will apply the following lemma:

Lemma 3.2 ([Rög08]). *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 $b_1, b_2, \ldots, b_\ell \in \mathbb{R}$ and $\epsilon > 0$,*

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

The proof of this lemma is given in Section 5.1, as the statement is more general than the original result. However, the proof remains the same.

Remark 1. 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 a coordinates of X has density bounded by ϕ^a. In [Rög08], the above lemma has been stated under this assumption of independence, but the proof of the lemma only requires this density bound.

It suffices, then, to show that L has large enough rank, applying the above lemma with $M = L(S, \boldsymbol{\sigma}^0)$ and $X = A$, and taking a union bound over the choice of S and $\boldsymbol{\sigma}^0$. The right rank bound would imply Theorem 3.1. The task at hand then is to get the largest possible rank bounds and tight union bounds to get good running time overall. We introduce here some parameters.

Definition 3.2 (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 otherwise, the player is termed *inactive*. 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 which 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. of non-repeating players in S,	$d_1(S)$	distinct moves by non-repeating players,
$p_2(S)$	number of repeating players in S,	$q_0(S)$	number of distinct (u, σ_u^0) moves

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.

3.1.1 Inactive Players, Rank Bounds, and Union Bounds

As discussed above, the goal is to show that $L(S)$ has large rank, and apply Lemma 3.2, taking a union bound over all the possible sequences S of size, say, ℓ. Naïvely, we have $k^n(nk)^\ell$ choices of sequence of length ℓ and initial strategy profile. 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 fix this, we will modify the matrix L in two different ways.

The two modified matrices will be relevant in the remaining analysis 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 [ABPW17, BCC19], however these two papers each use only one of the two constructions for both cases. In our analysis, the rank bounds for the case $p_1 \geq p_2$ were only attained on the rounding construction, and the bounds for the converse case were only attained on the other. This distinction arises from the size of the strategy space, which allows for more complex interactions between the rows of L, and both constructions are required in different arguments.

Control by rounding. The first builds on a construction introduced in [ABPW17]. While the construction works for arbitrary graphs, the rank bounds hold only for complete graphs. Observe that if $V_0 \subset V$ is the set of inactive players, and $u \in V$ is active, then for i fixed, all $((u, i)(v, \sigma_v^0))$ rows of L for $v \in V_0$ are identical, modulo flipping a row's signs. This is because v's strategy never changes. Therefore, in the inner product $\langle \boldsymbol{\lambda}_t, A \rangle$, these $((u, i), (v, \sigma_v^0))$ terms are added or subtracted together, and we may simply take a union bound on an approximation of this value, instead of controlling for strategy choices. This idea is formalized in Section 4.1.

For $p(S)$ fixed, there are at most $(nk)^\ell$ choices of the BR sequence, $k^{p(S)}$ choices of initial strategy profiles for the active players, and $d(S) - q_0(S)$ different "buckets" of the payoffs with $4n/\epsilon$

choices for the approximate value. Thus, we have a union bound of size $k^{p(S)}(nk)^{\ell}(4n/\epsilon)^{d(S)-q_0(S)}$. In Section 4.1, we show that, *whenever the graph is complete*, $L(S, \boldsymbol{\sigma}^0)$ has rank at least $d(S) - q_0(S) + d_1(S)/2$, after we consider a "bucketing" operation.

To bound the probability of success for all BR sequences, we restrict our attention to *critical subsequences*, as used in [ABPW17, BCC19]. These are maximal (up to inclusion) continuous subsequences S' satisfying $\ell(S') \geq 2(d(S') - q_0(S'))$, formally defined in Section 4.1. As we will show, these must exist, and satisfy $\ell(S') = 2(d(S') - q_0(S'))$, which by definition is at most $2kp(S)$. For a fixed choice of p, p_1, p_2, d, d_1, q_0, we bound the probability of *any* sequence having all improvements between 0 and ϵ, by $(nk\phi)^{\ell}\epsilon^{d_1(S)/2} \leq (nk\phi)^{O(kp(S))}\epsilon^{p_1(S)/2}$, using the above rank bounds and Lemma 3.2. Summing over all choices of parameters only introduces a polynomial blow-up.

Control by cyclic sums. The second is more intricate, and is loosely based on a construction in [ER17]. The bounds proved here hold for arbitrary graphs. The intent is to construct a new matrix Q whose columns lie in the span of L, which cancels the contributions of inactive players, but allows us to perform a similar analysis as above.

Suppose the move (u, i) appears twice in S, or (u, σ_u^0) appears in S. Let τ_0 be the index of the first occurrence of (u, i) 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, i) in the BR sequence. Then we let $\boldsymbol{q}_{u,i} := \sum_{j=1}^{m} \boldsymbol{\lambda}_{\tau_j}$, noting that the τ_0 is omitted, and we show in Section 4.2 that the entries of $\boldsymbol{q}$ corresponding to inactive players are all 0. Let $Q(S, \boldsymbol{\sigma}^0)$ be the matrix whose columns consist of the $\boldsymbol{q}$'s. To take a union bound on all Q matrices, it suffices to fix the initial strategy of only the active players. Furthermore, we have that $L \cdot A \in [0, \epsilon]^{\ell} \implies Q \cdot A \in [0, \ell\epsilon]^{d-d_1}$, so we may apply Lemma 3.2 on the matrix Q.

Fixing $p(S)$, there are at most $(nk)^{\ell}$ choices of the BR sequence, and $k^{p(S)}$ choices of initial strategy profiles for the active players. Thus, we have a union bound of size $k^{p(S)}(nk)^{\ell}$. In Section 4.2, we show that, on any graph, $Q(S, \boldsymbol{\sigma}^0)$ has rank at least $p_2(S)/2$. Thus, for a fixed choice of p, p_1, p_2, d, d_1, q_0, the probability of *any* sequence having all improvements being between 0 and ϵ is then bounded by $(nk\phi)^{O(kp(S))}(\ell\phi\epsilon)^{p_2(S)/2}$, by Lemma 3.2. Summing over all choices of the fixed parameters only introduces a polynomial blow-up.

Conclusion. We conclude Theorem 3.1 from the above bounds, which we summarize here:

Graph	Rank Bound	Union Bound	Probability of Success $\forall S$
complete	$d(S) - q_0(S) + d_1(S)/2$	$k^{p(S)}(4n/\epsilon)^{d(S)-q_0(S)}(nk)^{\ell}$	$1 - (nk\phi)^{O(k \cdot p(S))}\epsilon^{d_1(S)/2}$
general	$p_2(S)/2$	$k^{p(S)}(nk)^{\ell}$	$1 - (nk)^{O(k \cdot p(S))}(\ell\phi\epsilon)^{p_2(S)/2}$

The result on general graphs uses only the bounds from the cyclical sum construction, and a lemma of [ER17] which ensures that any sequence must contain a sub-sequence S' such that $p_2(S') \geq \Omega(p(S')/\log(nk))$. Applying the above bounds with $\epsilon = 1/(nk\phi)^{O(k \log(nk))}$ gives the desired result for general graphs. For complete graphs, we restrict ourselves to critical blocks as described above, using rounding when $p_1(S) \geq p_2(S)$, and cyclic sums when $p_2(S) \geq p_1(S)$. Setting $\epsilon = 1/(nk\phi)^{O(k)}$ for both gives the second half of Theorem 3.1. Along with the details of Sections 4, this concludes our proof of Theorem 3.1, and as a result, Theorem 1.1.

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

In this section, we define the notion of smoothness-preserving reduction, and outline the two reductions of Theorem 1.2.

3.2.1 Smoothness-Preserving Reductions

We refine standard Karp reductions to define *smoothness preserving reductions.* Recall from Section 2.2 that an algorithm is said to be smoothed-efficient if, on adversarially chosen combinatorial information, and random real-valued inputs, the algorithm runs in time polynomial in the input size and the degree of perturbation, with high probability. The idea behind the reductions is to allow a perturbed instance of $\mathcal{P}$ to be mapped to a perturbed instance of $\mathcal{Q}$, preserving sufficient randomness to allow for a smoothed efficient algorithm for $\mathcal{Q}$ to be applied. The output of the algorithm is then mapped back to a solution for the original $\mathcal{P}$ instance. The usual definition of Karp reduction does not allow such conclusions.

Definition 3.3 (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 matrix M with polynomially bounded entries, a constant η such that $1/\eta$ is polynomial in the input size, 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 $\mathcal{Q}$, such that if $\boldsymbol{\sigma}$ is a solution to J, then $f_2(\boldsymbol{\sigma})$ is a solution to 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 maximum density 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.

At first blush, the extra randomness R is superfluous. These variables are introduced to ensure that M has full-rank. The result of Proposition 2.1 ensures that 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. When M is diagonal, the random input to the reduced instance has independently distributed entries, which are required by most smoothed analysis results, and so strong reductions easily extend smoothed efficient algorithms. We conjecture that for most smoothed analysis, an upper-bound on the joint density of the input values suffices for efficient performance of the algorithm. We formalize as follows the properties of smoothness-preserving reductions.

Lemma 3.3. *(a) Suppose problem $\mathcal{Q}$ has (quasi-)polynomial smoothed complexity. Then, if problem $\mathcal{P}$ admits a strong smoothness-preserving reduction to $\mathcal{Q}$, then $\mathcal{P}$ also has (quasi-)polynomial smoothed complexity. (b) If $\mathcal{Q}$ still has smoothed efficient algorithms when the input is arbitrarily distributed with a bound on the joint density as in Proposition 2.1 and Lemma 3.2, then if $\mathcal{P}$ admits a weak smoothness-preserving reduction to $\mathcal{Q}$, $\mathcal{P}$ has (quasi-)polynomial smoothed complexity.*

A formal proof is given in Section 5. The results follows almost by definition, modulo technicalities. Observe that local-max-cut satisfies the conditions of part (b), since the proofs of [ER17, ABPW17] simply apply Lemma 3.2 to the input, similarly to the argument in Section 3.1. Thus weak reductions to local-max-cut do imply smoothed efficient algorithms. We highlight again that the key property of smoothness-preserving reductions is that the matrix M is full rank, since this ensures that the joint density on the reduced parameters is sufficiently bounded.

Note that the smoothed complexity of local-max-bisection is not yet known, 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.

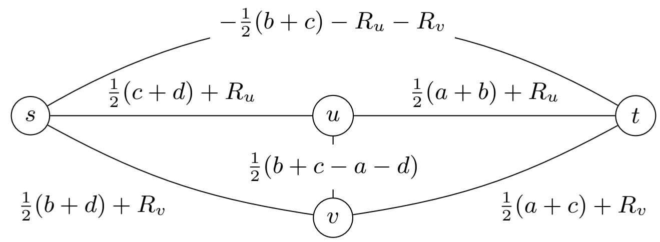
3.2.2 Outline of Reductions

We continue with a proof of Theorem 1.2, providing weak smoothness-preserving reductions from NetCoordNash to local-max-cut and -bisection. The reductions themselves are not particularly novel. What is more important is defining the matrix M and showing that it is indeed full-rank.

2-NetCoordNash reduces to Local-max-cut. We begin with the first reduction from network coordination games with 2×2 payoff matrices, to graph cuts. Let $G = (V, E)$ be the game graphs, with payoff vector A. We will construct a weighted cut graph $H = (V', E')$ where $V' = V \cup \{s, t\}$, and E' is obtained from E by adding su and ut edges for all $u \in V$. We wish to select 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 the vertex across the cut, and so solving for a local max cut is equivalent to solving for a local max of the game's potential function.

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

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



The general construction, which is specified in Section 5.2, places a copy of this gadget for each of the game-edges in the network game, and sums the edge weights. Observe that the above construction indeed has edge weights which are linear combinations of the payoff values. Furthermore, even if $R_u = R_v = 0$, cut values are equal to payoff values, and the maximal cuts are s-t cuts. The w_u and w_v values are added to increase the rank of the reduction matrix, which as discussed above, is the key property of these reductions. In Section 5.2, we show by induction on the number of players that the matrix has full rank, which implies that it is a valid reduction.

Observe that the cut graph is complete if and only if the game graph is, as all su and ut edges are present in the cut graph, and there is a uv edge in the cut graph whenever there is a uv game.

k-NetCoordNash reduces to Local-max-bisection. Reductions from games with k strategies, to show the second part of Theorem 1.2, is not as straight-forward. Let $G = (V, E)$ be the game graph again, and we will construct a weighted cut graph $H = (V', E')$. V' is given by the set of all (player,strategy) pairs $V \times [k]$, with two extra vertices s and t. E' is obtained as follows: for every node (u, i), we add an $\{s, (u, i)\}$ 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 $s \in S$, $t \in T$, and S contains exactly one (u, i) node for all $u \in V$. To balance the cuts, we will replace s with $n(k-2)+1$ copies $s_0, s_1, \ldots, s_{n(k-2)}$, and require them all to lie in S.

To each valid cut is associated the natural strategy profile where σ_u is given by the unique (u, i) node in S. 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 making the $\{(u, i), (u, j)\}$ edges bad, and the $\{s_a, (u, i)\}$ edges good, 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 1.2.

4 Smoothed Analysis of the BR Algorithm

In this section we provide the missing details of the proof of Theorem 1.1, as outlined in Section 3.1. We use the notation of Section 3.1, and begin with the case where the game graph is *complete*.

We will recall some notation. Fix a game graph $G = (V, E)$, and random vector of payoff values $A \in [-1,1]^{|E|k^2}$, where, for all $uv \in E$, $A((u,i)(v,j))$ is the payoff that players u and v receive in the uv game when u plays $i \in [k]$ and v plays $j \in [k]$. For a sequence of BR moves $S = (u_1, \sigma_1), \ldots, (u_\ell, \sigma_\ell)$ starting at initial strategy profile $\boldsymbol{\sigma}^0 \in [k]^n$, the strategy profiles over the sequence are defined as $\boldsymbol{\sigma}^t := (\sigma_t, \boldsymbol{\sigma}^{t-1}_{-u_t})$. Recall, $L(S, \boldsymbol{\sigma}^0) := \{\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_\ell\}$ where

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

As outlined in Section 3.1.1, we will provide two modifications of the matrix L with good union bounds, and show sufficient bounds on the rank of L. We wish to prove Theorem 3.1 — that for any sufficiently long sequence, at least one move must have increased the potential significantly Thus, we define the following:

Definition 4.1 (Minimum Improvement)**.** For a fixed sequence S and initial state $\boldsymbol{\sigma}^0$, recall that the t-th entry of $L(S, \boldsymbol{\sigma}^0) \cdot A$ is the value $\Phi(\boldsymbol{\sigma}^t) - \Phi(\boldsymbol{\sigma}^{t-1})$. Since A is random, these values are random, and not necessarily positive. If some entry is negative, then S is not a BR sequence. We define Δ_N as the increase in potential of the worst BR sequence. Formally,

$$\Delta_N = \min_{S, \boldsymbol{\sigma}^0} \|L(S, \boldsymbol{\sigma}^0) \cdot A\|_\infty \quad \text{subject to} \quad |S| = N, \quad L(S, \boldsymbol{\sigma}^0) \cdot A \geq \mathbf{0}$$

Observe that for $N = O(nk)$, if $\Delta_N = 1/poly(n^k, \phi)$, then the running time is polynomial, and if $\Delta_N = 1/poly(n^{k \log nk}, \phi)$, then the running time is quasi-polynomial. Thus, Theorem 3.1 is equivalent to showing that

$$\Pr_A \left[\Delta_{2nk} < \phi^{-1} (2n^2k^3)^{-20k \log(nk)} \right] \leq \tfrac{1}{(nk)^2} \qquad \text{in general} \qquad (3a)$$

$$\Pr_A \left[\Delta_{2nk} < (20\phi^2 n^3 k^3)^{-4k-4} \right] \leq \tfrac{1}{(nk)^3} \qquad \text{for complete game graphs} \qquad (3b)$$

It remains to give the proof of this result, which we will do by case analysis. We wish to distinguish between the cases when $p_1(S) \leq p_2(S)$, and the converse, following the outline of Section 3.1. We define $\overline{\Delta}_N$ and $\underline{\Delta}_N$ to be defined similarly, but requiring that $p_1(S) \geq p_2(S)$ and $p_1(S) \leq p_2(S)$, respectively.

4.1 Rank Bounds and Union Bounds via Rounding

We begin with the "rounding" construction of Section 3.1.1. We'll first define the matrix $\widetilde{L}$ for the rounded values, then introduce the concept of *critical block*, which will allow us to show the desired rank bounds. For this section, we may assume that the graph is *complete*. We will use completeness to ensure that between any two nodes, there is a game edge we may use for rank bounds, specifically in the proof of Lemma 4.2.

4.1.1 Matrix Construction

Fix a BR sequence S, and let V_0 be the set of inactive players, and V_1 the set of active players. Since we are looking to control the rate at which $\Phi(\boldsymbol{\sigma}^t)$ grows with t, we may without loss of generality assume $\Phi(\boldsymbol{\sigma}^0) = 0$ by adding a constant shift. Formally, let $\Psi(t) := \Phi(\boldsymbol{\sigma}^t) - \Phi(\boldsymbol{\sigma}^0)$, which satisfies $\Psi(t) - \Psi(t-1) = \Phi(\boldsymbol{\sigma}^t) - \Phi(\boldsymbol{\sigma}^{t-1})$. Further, define $\widetilde{A}((u,\sigma_u)(v,\sigma_v)) = A((u,\sigma_u)(v,\sigma_v)) - A((u,\boldsymbol{\sigma}_u^0)(v,\boldsymbol{\sigma}_v^0))$. Then

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

Rounding the effect of inactive players. Now, for $w \in V_0$, $\boldsymbol{\sigma}_w^t = \boldsymbol{\sigma}_w^0$, so middle terms on the second line are 0. Furthermore, the rightmost terms are in fact constants, depending only on σ_u. Let then $C(u,\sigma) := \sum_{w\in V_0} \widetilde{A}((u,\sigma)(w,\boldsymbol{\sigma}_w^0))$. Then the above sum can be expressed as

$$\Psi(t) \;:=\; \sum_{u,v\in V_1} \widetilde{A}((u,\boldsymbol{\sigma}_u^t)(v,\boldsymbol{\sigma}_v^t)) \;+\; 0 \;+\; \sum_{u\in V_1} C(u,\sigma_u^t)$$

Observe also that $C(u,\sigma_u^0) = 0$, since the $\widetilde{A}$ terms cancel. Finally, instead of controlling for $\Psi(t) - \Psi(t-1)$ exactly, it suffices to control for an approximation thereof.

We round the C values to the nearest multiple of ϵ, as was first introduced in [ABPW17]. Let $C'(u,\sigma)$ be the nearest multiple of ϵ to $C(u,\sigma)$. Since $C(u,\sigma) \in [-n,n]$ for all $u \in V_1$ and $\sigma \in [k]$, then there are $2n/\epsilon$ possible choices for $C'(u,\sigma)$. Let $\Psi'(t) := \sum_{u,v\in V_1} \widetilde{A}((u,\boldsymbol{\sigma}_u^t)(v,\boldsymbol{\sigma}_v^t)) + \sum_{u\in V_1} C'(u,\sigma_u^t)$, the approximation to Ψ obtained by using C' terms instead of C. Since $\Psi(t) - \Psi(t-1)$ depends only on two C terms, namely $C(u,\sigma_u^t)$ and $C(u,\sigma_u^{t-1})$, we have

$$|C(u,i) - C'(u,i)| < \epsilon/2 \implies \Big|\big(\Psi(t) - \Psi(t-1)\big) - \big(\Psi'(t) - \Psi'(t-1)\big)\Big| \le \epsilon$$

And therefore $\Pr[\Phi(\boldsymbol{\sigma}^t) - \Phi(\boldsymbol{\sigma}^{t-1}) \in (0,\epsilon)] \le \Pr[\Psi'(t) - \Psi'(t-1) \in (-\epsilon, 2\epsilon)]$. This new definition of potential Ψ will allow us to reduce the needed union bounds.

Union bound size. Let $\widetilde{L}$ be obtained from L where for $u \in V_1$, and $i \in [k]$, we replace the set of rows $\{(u,i)(w,j) : w \in V_0, j \in [k]\}$ with the single row for $C(u,i)$ — and therefore $C'(u,i)$. To define $\widetilde{L}$, it suffices to control $\boldsymbol{\sigma}^0$ restricted to the active players, the sequence S, and the values of the C' terms. Recall that $C'(u,\sigma_u^0) = 0$. Furthermore, the only $C(u,i)$ terms which are considered are those for moves (u,i) which appear in the sequence. Therefore, we only have $d(S)$ terms to control, $q_0(S)$ of which are 0, so we get a union bound of size $k^{p(S)}(nk)^\ell(4n/\epsilon)^{d(S)-q_0(S)}$.

4.1.2 Critical Subsequences

As discussed in Section 3.1, to get our rank bounds, we will want a sub-sequence S' whose length is proportional to $d(S') - q_0(S')$. To this end, we define *critical subsequences* below. These are closely based on the definition of a critical block in [ABPW17].

Definition 4.2 (Critical Subsequence)**.** For every contiguous subsequence B of S, let $\ell(B)$, $d(B)$, and $q_0(B)$ be length, number of distinct pairs, and number of return moves, in B, respectively, as defined in Section 3.1. Such a subsequence is termed *critical* if $\ell(B) \ge 2\big(d(B) - q_0(B)\big)$, but for every $B' \subseteq B$, $\ell(B') < 2\big(d(B') - q_0(B')\big)$.

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. We show that critical subsequences must exist.

Lemma 4.1. *A critical subsequence always exists in any sequence S of length $2nk$. Furthermore, if B is a critical subsequence, then $\ell(B) = 2(d(B) - q_0(B))$.*

Proof. As there are at most nk distinct player-strategy pairs possible, the entire sequence S satisfies the relation $\ell(S) \geq 2d(S) \geq 2(d(S) - q_0(S))$. Conversely, for every subsequence B of length 1 (*i.e.* a single move), $d(B) = 1, q_0(B) = 0 \Rightarrow 1 = \ell(B) < 2(d(B) - q_0(B)) = 2$. Thus, it suffices to take an inclusion-minimal subsequence which satisfies $\ell(B) \geq 2(d(B) - q_0(B))$ and obtain a critical subsequence.

It remains to show that for B critical $\ell(B) = 2d(B) - 2q_0(B)$. Suppose not, then it is strictly larger. Let B' be obtained from B by dropping the last column. Then,

$$\ell(B') \;=\; \ell(B) - 1 \;\geq\; 2d(B) - 2q_0(B) + 1 - 1$$

Now, we claim $d(B) - q_0(B) \geq d(B') - q_0(B')$. Clearly $d(B) - 1 \leq d(B') \leq d(B)$, and $q_0(B) - 1 \leq q_0(B') \leq q_0(B)$. However, if $q_0(B') = q_0(B) - 1$, then we must also have $d(B') = d(B) - 1$. Thus, in all cases, $d(B) - q_0(B) \geq d(B') - q_0(B')$. This implies $\ell(B') \geq 2(d(B') - q_0(B'))$, contradicting the criticality of B. □

The tight bound $\ell(B) = 2(d(B) - q_0(B))$ is key for proving the main rank lemma from this section, below. We show that critical subsequences have high rank, which by Lemma 4.1 extends to any length-$2nk$ sequence.

We extend the definition of Δ_N (Def. 4.1) to include the use of critical sequences.

Definition 4.3. We define $\Delta'(p)$ to be the minimum total increase due to any critical subsequence with exactly p active players, where the initial strategy profile is chosen arbitrarily. Formally,

$$\Delta'(p) = \min_{\boldsymbol{\sigma}^0} \|L(S, \boldsymbol{\sigma}^0) \cdot A\|_\infty \quad \text{subject to} \quad S \text{ critical}, \quad p(S) = p, \quad L(S, \boldsymbol{\sigma}^0) \cdot A \geq \mathbf{0}$$

Similarly, $\overline{\Delta}'(p)$ and $\underline{\Delta}'(p)$ represent the same value, with the extra restriction that $p_1(S) \geq p_2(S)$ and $p_1(S) \leq p_2(S)$, respectively. Observe, $\Delta'(p) = \min\{\overline{\Delta}'(p), \underline{\Delta}'(p)\}$.

Since any sequence must have a critical subsequence, $\Delta_N \geq \min_{p=1}^{N} \Delta'(p)$, we will take a union bound to show that the probability that $\Delta_N < 1/(nk\phi)^{O(k)}$ is small. As outlined in Section 3.1, this bound is performed separately for the two cases specified in Definition 4.3, $\min\{\overline{\Delta}'(p)$ and $\underline{\Delta}'(p)\}$.

4.1.3 Rank Bounds from Separated Blocks

We provide here the main rank bound of this section, which we begin with a definition

Definition 4.4 (Separated Blocks)**.** Fix a BR sequence S, and let $P_1(S)$ be the set of non-repeating active players. For $u \in P_1$, let T_u be the set of indices where the moving player is u. Let $T = \bigcup_{u \in P_1} T_u$, and denote without loss of generality $T = \{t_1 < t_2 < \cdots < t_m\}$. We will show below how the t_i's "separate" the sequence S, since we will be able to control their ranks separately. To this end, let S_i for $i = 0, 1, \ldots, m$ 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, and denote their collection as $\mathbb{S} = \{S_0, S_1, \ldots, S_m\}$. Furthermore, note that $|T| = d_1(S)$.

The following lemma allows us to take advantage of this notion of separated block, to break up the rank bounds into simpler subproblems.

Lemma 4.2. *Assume the game graph is complete, and let S be a BR sequence with at least one inactive player, and let $L = L(S, \boldsymbol{\sigma}^0) = \{\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \ldots\}$. Then L contains at least $d_1(S) + \sum_{S' \in \mathbb{S}} d(S') - q_0(S')$ linearly independent vectors.*

Proof. Let w be some inactive player, which we have assumed exists. Let $T = \{t_1 < t_2 < \cdots < t_m\}$ be the endpoints of the separated blocks, as in Definition 4.4 above. For $i = 0, 1, \ldots, m$, let D_i be the set of distinct (player,strategy) moves which occur in S_i, which are not *return moves* of S_i, *i.e.* $(u, \boldsymbol{\sigma}_u^{t_i})$ moves.

For all i, the move at t_i must be some non-repeating player of S, which we denote v_i, and call the strategy it moves to as σ^i, letting v_0 be the inactive player w, and $\sigma^0 := \sigma_w^0$. For all $(u, \sigma) \in D_i$, let $\tau^i_{(u,\sigma)}$ be the time of the *first* occurrence of (u, σ) in the subsequence S_i, and let $H_i = \{\tau^i_{(u,\sigma)} : (u, \sigma) \in D_i\}$. Let $H = \bigcup_{i=0}^{|\mathbb{S}|-1} H_i \cup \{t_1, \ldots, t_{|\mathbb{S}|-1}\}$. For each $t \in H$, if $t = \tau^i_{(u,\sigma)} \in H_i$ for some i, u, σ, then *associate* to L_t the row $((u, \sigma)(v_i, \sigma^i))$. If, instead, $t = t_i$ for some i, then *associate* to L_t the row $((v_i, \sigma^i)(w, \sigma^0))$. Note that this row exists because the game graph is complete.

Consider the submatrix of L consisting of all columns $\{\boldsymbol{\lambda}_t : t \in H\}$, sorted in "chronological" order, and all of their associated rows, in the same order as their respectively associated columns. We claim that this matrix is upper-triangular, and its diagonal entries are non-zero. For each column $\boldsymbol{\lambda}_t$, the diagonal entry in the submatrix is the entry for the associated row, which we have chosen to be nonzero. Furthermore, if $t = t_i \in H$, then v_i all previous columns have 0 entries in the $((v_i, \sigma^i)(\cdot, \cdot))$ rows, since v_i is non-repeating. Thus, $\boldsymbol{\lambda}_{t_i}$ is the first column where the associated row has a nonzero entry. If, instead, $t = \tau^i_{(u,\sigma)} \in H_i$, then the associated row $((u, \sigma)(v_i, \sigma^i))$ must have been 0 up until column $\boldsymbol{\lambda}_{t_i}$ as described above. Furthermore, since $\tau^i_{(u,\sigma)}$ is the first occurrence of $(u, \sigma \neq \boldsymbol{\sigma}_u^{t_i})$ after time t_i, we must have had the row $((u, \sigma)(v_i, \sigma^i))$ be 0 before the $\tau^i_{(u,\sigma)}$-th column.

These observations imply that our $|H| \times |H|$ submatrix, with the given row-ordering, must be upper-triangular with nonzero diagonal terms. Therefore, it must be full-rank. Since $|H_i| = d(S_i) - q_0(S_i)$, then $|H| = d_1(S) + \sum_{S' \in \mathbb{S}} d(S') - q_0(S')$, and we conclude the desired bound. □

We also extend this proof to the case of all players active.

Corollary 4.3. *Let S be a BR sequence where all players are active, and let $L = L(S, \boldsymbol{\sigma}^0)$. Then L contains at least $\left(1 - \frac{1}{n}\right)(d(S) - q_0(S))$ linearly independent vectors.*

Proof. Consider the above proof method with $|T| = 0$, and $S_0 = S$. Note that now, $H = D_0$. It is still correct if some arbitrary player is chosen to be the w player, and all $((u, \sigma)(v_0, \sigma^0))$ terms are replaced with $((u, \sigma)(w, \boldsymbol{\sigma}^{\tau^0_{(u,\sigma)}}))$ terms. We must further restrict H not to contain any moves of player w. Suppose we choose, as our w player, the player which appears the least number of times in H, then we suffer a $\left(\frac{1}{n}\right)$-fraction loss in the size of H, concluding the proof. □

This above lemma and corollary, along with the notion of critical block, will give us our desired bound.

Lemma 4.4. *Assume a complete game graph, and let S be a BR sequence of length $2nk$ which has at least one inactive player. Let B be some critical subsequence of S starting at t_0, and let $L = L(B, \boldsymbol{\sigma}^{t_0})$. Then L contains at least $\frac{1}{2} d_1(B) + d(B) - q_0(B)$ linearly independent vectors.*

Proof. Since S has an inactive player, then so must B. Therefore, Lemma 4.2 applies. Recall, Lemma 4.2 shows that L contains at least $d_1(B) + \sum_{S' \in \mathbb{S}(B)} d(S') - q_0(S')$ linearly independent vectors. If $p_1(B) = d_1(B) = 0$, then we are done. Otherwise, since B is critical, then for all $S' \in \mathbb{S}(B)$, $\ell(S') < 2(d(S') - q_0(S'))$. Hence,

$$rank(L) \;\geq\; d_1(B) + \sum_{S' \in \mathbb{S}(B)} d(S') - q_0(S') \;>\; \tfrac{1}{2} d_1(B) + \tfrac{1}{2} d_1(B) + \sum_{S' \in \mathbb{S}(B)} \tfrac{1}{2}\ell(S')$$

However, $\ell(B) = d_1(B) + \sum_{S' \in \mathbb{S}(B)} \ell(S')$, and so this implies $rank(L) \geq \frac{1}{2} d_1(B) + \frac{1}{2}\ell(B)$. By criticality and Lemma 4.1, $\ell(B) \geq 2(d(B) - q_0(B))$, giving us our desired bound. ☐

This concludes the first rank- and union-bound of Section 3.1.1. Using this, and Lemma 3.2, we show our first result:

Theorem 4.5. $\Pr\left[\, \overline{\Delta}'(p) \in (0, \epsilon)\right] \leq \left(\left(20\phi^2 n^3 k^3\right)^k \epsilon^{1/4}\right)^p$.

Proof. From the above lemmas, it remains to apply Lemma 3.2. For a fixed critical subsequence S with p active players, if $p_1 \geq p_2$, then by Lemma 4.4, the improvement of each step of the *approximate* potential Ψ along the sequence will lie in $(-\epsilon, 2\epsilon)$ with probability $(3\phi\epsilon)^{d(S) - q_0(S) + p(S)/4}$. Taking a union bound over *all* approximated sequences, this event holds with probability

$$k^{p(S)} (nk)^{\ell(S)} (2n/\epsilon)^{d(S) - q_0(S)} (3\phi\epsilon)^{d(S) - q_0(S) + p(S)/4}$$

Noting that $d(S) - q_0(S) \leq k \cdot p(S)$, and by criticality of S, $\ell(S) \leq 2d(S) - 2q_0(S) \leq 2kp(S)$, so

$$\begin{aligned}
\Pr\left[\, \overline{\Delta}'(p) \in (0, \epsilon)\right] &\leq k^{p(S)} (nk)^{\ell(S)} (2n/\epsilon)^{d(S) - q_0(S)} (3\phi\epsilon)^{d(S) - q_0(S) + p(S)/4} \\
&\leq 20^{k \cdot p(S)} (nk\phi)^{2k \cdot p(S)} (nk)^{k \cdot p(S)} \epsilon^{p(S)/4} \\
&= \left((20 n^3 k^3 \phi^2)^k \epsilon^{1/4}\right)^{p(S)}
\end{aligned}$$

as desired. ☐

4.2 Rank Bounds and Union Bounds via Cyclic Sums

In this subsection, we will prove the second half of the results from Section 3.1.1. Unlike the rank bounds of the previous section, all statements in this section hold for arbitrary game graphs.

Recall that, for a fixed BR sequence S, we have defined the matrix $Q(S, \boldsymbol{\sigma}^0)$, whose columns consist of sums of columns of $L(S, \boldsymbol{\sigma}^0) = \{\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \ldots, \boldsymbol{\lambda}_\ell\}$. We recall its definition here: Let (u, i) be a move which appears twice in S — possibly a (u, σ_u^0) return move. Let τ_0 be the index of the first occurrence of (u, i) in the BR sequence, setting $\tau_0 = 0$ for return moves. Let $\tau_1, \tau_2, \ldots$ be the indices of all subsequent moves by player u in the sequence, and suppose τ_m is the second occurrence of (u, i) in the BR sequence, or first, if it is a return move. Define $\boldsymbol{q}_{u,i} := \sum_{j=1}^m \boldsymbol{\lambda}_{\tau_j}$, noting that the τ_0 is omitted, and let $Q(S, \boldsymbol{\sigma}^0)$ be the matrix whose columns consist of the $\boldsymbol{q}$'s.

We wish to show that $Q(S, \boldsymbol{\sigma}^0)$ does not depend on the strategies of the inactive players. Intuitively, this holds because we are taking a "cyclic sum" of the moves of a player u, and therefore we are cancelling out the entering and exiting payoff values. Formally, note that if player u is moving at time t, $\boldsymbol{\lambda}_t$ has only nonzero entries in $(u, \cdot)(\cdot, \cdot)$ rows. Furthermore, if w is inactive, and σ is any strategy played by u, then the $(u, \sigma)(w, \sigma_w^0)$ row of $\boldsymbol{q}_{u,i}$ is given by

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

Thus, we have that to fully specify $Q(S, \boldsymbol{\sigma}^0)$, it suffices to know S and the initial strategy profiles of the active players. It remains to show $Q(S, \boldsymbol{\sigma}^0)$ has large rank, as follows.

Lemma 4.6. *Fix a BR sequence S and starting configuration $\boldsymbol{\sigma}^0$. Then $Q = Q(S, \boldsymbol{\sigma}^0)$ contains at least $p_2(S)/2$ linearly independent vectors.*

Proof. We begin by constructing an auxiliary *directed* graph $G' = (V, E')$, where V is the set of players, and E' will be defined as follows: let (u, σ) be some repeating move. We cannot have $\boldsymbol{q}_{u,\sigma}$ be the all-zero vector, as otherwise $\boldsymbol{\sigma}^{\tau_0} = \boldsymbol{\sigma}^{\tau_m}$, which cannot hold for a BR sequence. For *every* player $w \in V$ such that $\boldsymbol{q}\big((u, \sigma)(w, \sigma')\big) \neq 0$ for some $\sigma' \in [k]$, add the directed edge (u, w) to E'.

Let $P_2 \subseteq V$ be the set of repeating players, and note that they all have non-zero out-degree. Consider the following procedure: pick a vertex $r_1 \in P_2$, and let T_1 be the BFS arborescence rooted at r_1 which spans all nodes reachable from r_1 in G'. Then delete $V(T_1)$ from G' and repeat, picking an arbitrary root vertex $r_2 \in P_2 \setminus V(T_1)$, and get the arborescence T_2 on everything reachable from r_2. We may continue this until every vertex of P_2 is covered by some arborescence. For each $i = 1, 2, \ldots$, let T_i^0 and T_i^1 be the set of nodes of T_i which are of even or odd distance from r_i along T_i, respectively. Let P_i' be the larger of $V(T_i^0) \cap P_2$ and $V(T_i^1) \cap P_2$, and $P_2' := \bigcup_{i=1}^{\infty} P_i'$.

We must have that $|P_2'| \geq |P_2|/2 = p_2(S)/2$. We wish to show that the collection $\mathcal{V} := \{\boldsymbol{q}_{u,\cdot} : u \in P_2'\}$ is independent. Every $u \in P_2'$ must have some out-neighbour w. If u was not a leaf of the arborescence it was selected in, then it must have some out-neighbour along the arborescence, and we may choose this neighbour. This out-neighbour can not also be in P_2'. In this case, $\boldsymbol{q}_{u,\cdot}$ will be the only vector from $\mathcal{V}$ to contain a non-zero $((u, \cdot)(w, \cdot))$ entry, since w was not taken in P_2'. If, instead, u was a leaf of its arborescence, 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 the other arborescence. Therefore, $\boldsymbol{q}_{u,\cdot}$ is the only vector in $\mathcal{V}$ to contain a nonzero $((u, \cdot)(w, \cdot))$ entry. Thus, $\mathcal{V}$ must contain a $|\mathcal{V}| \times |\mathcal{V}|$ diagonal submatrix, and therefore has rank at least $|\mathcal{V}| \geq p_2(S)/2$, as desired. □

using the above lemma along with the appropriate union bound discounting the inactive players, we show the following Theorem:

Theorem 4.7. $\Pr\left[\underline{\Delta}'(p) \in (0, \epsilon)\right] \leq \left(2(nk)^{2k} k^{5/4} (n\phi\epsilon)^{1/4}\right)^p$.

Proof. Fix S, $\boldsymbol{\sigma}^0$, and let $L = L(S, \boldsymbol{\sigma}^0)$. Let A be the payoff vector of the network coordination game. Let $\mathcal{V}$ be a collection of $p_2(S)/2$ independent vectors from Lemma 4.6. Let $\boldsymbol{q} \in \mathcal{V}$ and recall $\boldsymbol{q} = \sum_{j=1}^m \boldsymbol{\lambda}_{t_j}$ for some collection of indices $t_1 < \cdots < t_m$. We have $\Pr[\bigwedge_{i=1}^m \langle \boldsymbol{\lambda}_{t_i}, A \rangle \in (0, \epsilon)] \leq \Pr[\langle \boldsymbol{q}, A \rangle \in (0, m\epsilon)]$. Since $m \leq \ell$, taking the collection of all $\boldsymbol{q}$ vectors and applying Lemma 3.2, we have

$$\Pr\left[\bigwedge_{t=1}^{\ell(S)} \langle \boldsymbol{\lambda}_t, A \rangle \in (0, \epsilon)\right] \leq \Pr\left[\bigwedge_{\boldsymbol{q} \in \mathcal{V}} \langle \boldsymbol{q}, A \rangle \in (0, \ell\epsilon)\right] \leq (\ell\phi\epsilon)^{p_2(S)/2}$$

There are at most $k^{p(S)}(nk)^{\ell(S)}$ possible collections $\mathcal{V}$. The quantity $\underline{\Delta}'(p)$ assumes we are in a critical subsequence, which implies $\ell(S) = 2(d(S) - q_0(S))$ by Lemma 4.1 Since $\ell(S) = 2(d(S) - q_0(S)) \leq k \cdot p(S)$, and $p_2 \geq p_2 \implies p_2(S) \geq \frac{1}{2} p(S)$, we have

$$\begin{aligned}
\Pr\left[\underline{\Delta}'(p) \in (0, \epsilon)\right] &\leq k^{p(S)} (nk)^{\ell(S)} (\ell\phi\epsilon)^{p_2(S)/2} \\
&\leq n^{2k \cdot p(S)} k^{(2k+1)p(S)} (2kp)^{p(S)/4} (\phi\epsilon)^{p(S)/4} \\
&\leq \left(2(nk)^{2k} k^{5/4} (n\phi\epsilon)^{1/4}\right)^{p(S)}
\end{aligned}$$

as desired. □

4.3 Polynomial Smoothed Complexity for Complete Game Graphs

We have shown above that $\overline{\Delta}'(p)$ and $\underline{\Delta}'(p)$ have vanishing probability of lying in $(0, \epsilon)$. In this section, we use these results to show that the BRA will terminate in time polynomial in n^k, k and ϕ, with high probability, when the game graph is complete. The following lemma combines our two previous results:

Lemma 4.8. *With probability $1 - 1/O(\phi^2 n^3 k^4)$, every BR sequence of length $2nk$ must have an improvement of at least $\epsilon = O\big((\phi^2 n^3 k^3)^{-4k-4}\big)$.*

Proof. We will perform a case analysis based on the values of $p_1(S)$ and $p_2(S)$, with cases for $p(S) = n$, $p(S) < n$ and $p_1(S) \geq p_2(S)$, and $p_2(S) \geq p_1(S)$.

If $p(S) = n$, we apply the rank bound of Corollary 4.3 and take a union bound over all initial strategy profiles, and all possible sequences to get

$$\Pr[\Delta'(n) \in (0, \epsilon)] \leq k^n (nk)^{2nk} (\phi\epsilon)^{n-1} \leq \left(k^{3k} n^{2k} \phi\epsilon\right)^n \Big/ \phi\epsilon \tag{4}$$

This union bound over-counts the number of sequences with $p(S) = n$, but this isn't a problem. Setting $\epsilon = \phi^{-1} \left(n^2 k^3\right)^{-2k}$ gives $\Pr[\Delta'(n) \in (0, \epsilon)] \leq \left(\frac{1}{n^2 k^3}\right)^{n-2}$.

In the converse case, we combine Theorems 4.5 and 4.7, then take a union bound over all possible values of p to bound the probability for any sequence of the given length. As defined previously, $\Delta'(p) = \min\{\overline{\Delta}'(p), \underline{\Delta}'(p)\}$ and so,

$$\Pr[\Delta'(p) \in (0, \epsilon)] \;\leq\; \left((20\phi^2 n^3 k^3)^k \epsilon^{1/4}\right)^p + \left(2(\phi\epsilon)^{1/4} n^{2k+1/4} k^{2k+5/4}\right)^p \;\leq\; 2\left((20\phi^2 n^3 k^3)^k \epsilon^{1/4}\right)^p \tag{5}$$

Since any sequence of length $2nk$ must contain a critical subsequence, it suffices to set $\epsilon = \left(20\phi^2 n^3 k^3\right)^{-4k-4}$, and taking the union bound over all choices of p, we get

$$\Pr[\Delta_{2nk} \in (0, \epsilon)] \leq \sum_{p=1}^{n} \left(20\phi^2 n^3 k^4\right)^{-p} \;\leq\; \frac{1}{(20\phi^2 n^3 k^4) - 1}$$

Combining the two cases of $p = n$ and $p < n$ gives us our desired result. $\square$

This concludes the proof of the complete-game-graphs part of Theorem 3.1, noting that $\phi \geq \frac{1}{2}$. As outlined in Section 3.1, and at the top of this section, this implies that with probability $1 - 1/poly(n, k, \phi)$, any correct implementation of the BRA will converge to a PNE of the network coordination game in at most $(nk\phi)^{O(k)}$ steps.

4.4 Quasipolynomial Smoothed Complexity for General Game Graphs

In this section we show the quasi-polynomial running time when the game graph G is incomplete, and thus prove the remaining part of Theorem 1.1. The analysis mostly uses the lemmas from Section 4.2, paired with the following definition and lemma from [ER17]:

Definition 4.5. Recall the random variable Δ from Definition 4.1. Call a sequence of length ℓ log-repeating if it contains at least $\ell/(5\log(nk))$ repeating moves (pairs). We denote as $\Delta''(\ell)$ the minimum total potential-improvement after any log-repeating BR sequence of length exactly ℓ.

Lemma 4.9 ([ER17], Lemma 3.4)**.** *Let Δ_N and $\Delta''(\ell)$ be as above. Then $\Delta_{5nk} := \min_{1\leq\ell\leq 5nk} \Delta''(\ell)$*

The proof of the above lemma shows that any sequence on $5nk$ pairs must contain some contiguous sub-sequence which is log-repeating. Thus, for the remainder of the analysis, it suffices to bound $\Delta''(\ell)$. Since a sequence captured by $\Delta''(\ell)$ must have at least $\ell/(5\log(nk))$ repeated terms, it must have $p_2 \geq \ell/(5k\log(nk))$. Therefore, as we have shown in the proof of Theorem 4.7, we have $\Pr[\Delta''(\ell) \in (0,\epsilon)] \leq k^\ell (nk)^\ell (\ell\phi\epsilon)^{\ell/10k\log(nk)}$. It suffices, then to simply take the union bound over all possible values of ℓ.

Theorem 4.10. *Given a smoothed instance of k-NetCoordNash with an arbitrary initial strategy profile, then any execution of a BR algorithm where improvements are chosen arbitrarily will converge to a PNE in at most $\phi \cdot (nk)^{O(k\log(nk))}$ steps, with probability $1-(nk)^{-2}$.*

Proof. As discussed above,

$$\begin{aligned} Pr[\Delta''(\ell) \in (0,\epsilon)] &\leq k^\ell (nk)^\ell (\ell\phi\epsilon)^{\ell/10k\log(nk)} \\ &\leq \left(k^2 n (5nk\phi\epsilon)^{1/(10k\log(nk))}\right)^\ell && (\ell \leq 5nk) \\ &\leq \left(2k^3 n^2 (\phi\epsilon)^{1/(10k\log(nk))}\right)^\ell . && (5^{1/10} \leq 2) \end{aligned} \tag{6}$$

Setting $\epsilon = \phi^{-1}(2n^2k^3)^{-2\cdot 10k\log(nk)}$, this gives

$$Pr[\Delta''(\ell) \in (0,\epsilon)] \leq \left(\frac{1}{2n^2k^3}\right)^\ell$$

Let Δ_{5nk} be the improvement in potential in any length $5nk$ BR sequence. Then using Lemma 4.9, and taking the union bound over all choices of ℓ, we have,

$$\Pr[\Delta_{5nk} \in (0,\epsilon)] \leq \sum_{\ell=1}^{5nk} \Pr[\Delta''(\ell) \in (0,\epsilon)] \leq \sum_{\ell=1}^{5nk} (2n^2k^3)^{-\ell} \leq \frac{(2n^2k^3)^{-1}}{1-(2n^2k^3)^{-1}} = \frac{1}{2n^2k^3-1} \leq \frac{1}{(nk)^2}$$

Hence, with probability $1 - 1/poly(n,k)$ (over the draw of payoff vector A), *all* BR sequences of length $5nk$ will have total improvement at least ϵ. In that case, any execution of BR algorithm makes an improvement of at least ϵ every $5nk$ moves. Since the total improvement is at most $2n^2$, we conclude that the total number of steps is at most $5nk \cdot 2n^2/\epsilon = 10n^3k(2n^2k^3)^{20k\log(nk)} \cdot \phi = \phi \cdot (nk)^{O(k\log(nk))}$, and this occurs with probability $1 - 1/poly(n,k)$. □

This concludes the proof of the arbitrary-graphs part of Theorem 3.1, noting that $\phi \geq \frac{1}{2}$. As outlined in Section 3.1, and at the top of the previous section, this implies that with probability $1 - 1/poly(n,k,\phi)$, any correct implementation of the BRA will converge to a PNE of the network coordination game in at most $(nk\phi)^{O(k\log(nk))}$ steps.

This completes our analysis of the smoothed performance of BRA for finding pure Nash equilibria in network coordination games. In the next section, we show that this result indeed holds in expectation, and then go on to show a notion of smoothness-preserving reduction which allows us to prove alternative, conditional, algorithms for this problem.

4.5 (Quasi)Polynomial Running time in Expectation

The analysis in the previous section establishes smoothed complexity of network coordination games with respect to the *with high probability* notion. Another aspect of smoothed analysis is to analyze the expected time of completion of the algorithm. Observe that the expected running time of an algorithm can not be immediately concluded from the high-probability running time, and this performance will depend on the explicit bounds computed. In this section, we provide a theorem to obtain expected time results from the with high probability bounds. The results are presented in a general form to allow application to any problem in PLS that has a bounded total improvement in potential value. The following theorem is a generalization of the statement of a result found in [ER17]. We include the analysis for completeness.

Theorem 4.11. *Given a PLS problem with input size N, potential function range $[-N^{r_1}, N^{r_2}]$, and a local-search algorithm $\mathcal{A}$ to solve it, let d be the number of distinct choices the algorithm has in each step and let Λ be the total size of the search space of the algorithm. For an instance I drawn at random with maximum density ϕ, suppose the probability that any length-N^β sequence of improving moves of $\mathcal{A}$ results in total improvement in the potential value at most ϵ, is at most $\sum_{q=1}^{N^\beta}(N^{f(N)}(\phi^{g'(N)}\epsilon)^{1/g(N)})^q$. Then the expected running-time of the algorithm is $O(\phi^{g'(N)} \cdot N^{\beta+r} \cdot g(N) \cdot N^{f(N)g(N)} \cdot \ln \Lambda)$. Here, $f(N)$ $g'(N)$ and $g(N)$ are functions of N.*

Proof. The maximum improvement possible before $\mathcal{A}$ terminates is the maximum change in the potential function value, given by $N^{r_2} + N^{r_1}$. For any integer $t \geq 1$, if the algorithm requires more than t steps to terminate, then there must exist some subsequence of length N^β that results in an improvement in the potential value of less than $N^\beta(N^{r_2} + N^{r_1})/t \leq 2N^{\beta+\max\{r_2,r_1\}}/t$. We denote $r := \max\{r_1, r_2\}$.

We define a random variable T as the number of steps $\mathcal{A}$ requires to terminate. Using the notation $\Delta(N^\beta)$ to denote the minimum total improvement in a length-N^β sequence of the algorithm $\mathcal{A}$, this gives the probability of $\mathcal{A}$ running for more than t steps as:

$$\Pr[T \geq t] \leq Pr[\Delta(N^\beta) \in (0, N^{r+\beta}/t)] \leq \sum_{q=1}^{N^\beta} \left(N^{f(N)} \left(\phi^{g'(N)} \cdot \frac{N^{\beta+r}}{t} \right)^{1/g(N)} \right)^q .$$

We define $t = \gamma i$, for $\gamma = N^{f(N)g(N)}(\phi^{g'(N)}N^{r+\beta}) = \phi^{g'(N)}N^{f(N)g(N)+\beta+r}$, and compute the probability of $T \geq \gamma i$ for any integer i:

$$\Pr[T \geq \gamma i] \leq \sum_{q=1}^{N^\beta} \left(N^{f(N)} \left(\phi^{g'(N)} \cdot \frac{N^{r+\beta}}{\gamma i} \right)^{1/g(N)} \right)^q \leq \sum_{q=1}^{\infty} \left(\frac{1}{i}\right)^{q/g(N)} \leq g(N) \sum_{q'=1}^{\infty} \left(\frac{1}{i}\right)^{q'} \leq \frac{g(N)}{i-1}.$$

We now sum over all values of t, by using that $\Pr[T \geq t] \leq \Pr[T \geq t \cdot \lceil t/\gamma \rceil]$, and compute the expected time steps as:

$$\mathbb{E}[T] = \sum_{t=1}^{\Lambda} \Pr[T \geq t] \leq \sum_{i=1}^{\Lambda/\gamma} \sum_{t=1}^{\gamma} \Pr[T \geq (i+1)\gamma] \leq \sum_{i=2}^{\Lambda/\gamma} \frac{g(N)\gamma}{i-1} = O(g(N) \cdot \gamma \cdot \ln \Lambda)$$

Thus, replacing the value for γ, the expected runtime is at most $O(\phi^{g'(N)}N^{\beta+r}g(N)N^{f(N)g(N)} \ln \Lambda)$. □

Corollary 4.12. *The smoothed expected time for BR to terminate for all network coordination games is polynomial in $(n^{(k \log(nk))}, \phi)$.*

Proof. From (6) in Theorem 4.10, we know that the probability that the minimum improvement in a fixed BR sequence of length $5nk$ is no more than ϵ is at most $\sum_{\ell=1}^{5nk} \left(2n^2k^3(\phi\epsilon)^{1/(10k\log(nk))}\right)^l$.

Applying Theorem 4.11, for $N = nk$ and $\Lambda \leq k^n$, we get $f(N) = O(1)$, $N^{r+\beta} \leq N^3$, $g'(N) = 1$ and $g(N) = O(k\log(nk))$, and the expected running time is

$$O(\phi(nk)^{O(1)}O(k\log(nk))(nk)^{O(1)\cdot O(k\log(nk))}ln(k^n)) \equiv O(\phi n^{O(k\log(nk))}).$$

□

Corollary 4.13. *For complete graphs, the smoothed expected time for BR to terminate for network coordination games is polynomial in* (n^k, ϕ).

Proof. From (4) in Lemma 4.8, for the case of complete graphs when a BR sequence has all active players, we have:

$$\Pr[\Delta(p) \in (0,\epsilon)] \leq \left(k^{3k}n^{2k}\phi\epsilon\right)^n \Big/ \phi\epsilon \leq \sum_{i=1}^{n} \left(k^{3k}n^{2k}\phi^{1/2}\epsilon^{1/2}\right)^i \Big/ \phi\epsilon.$$

Similarly, from (5) in Lemma 4.8, the probability that the minimum improvement in a BR sequence of length $2nk$ is at most ϵ, is given by:

$$\Pr[\Delta(p) \in (0,\epsilon)] \leq \sum_{p=1}^{n} 2\left((20\phi^2n^3k^4)^k\epsilon^{1/4}\right)^p$$

Combining these sums, we get the probability that a BR sequence of length $2nk$ has improvement at most ϵ is:

$$\begin{aligned}\Pr[\Delta(p) \in (0,\epsilon)] &\leq \max\left\{\sum_{p=1}^{n} 2\left((20\phi^2n^3k^5)^k\epsilon^{1/4}\right)^p, \sum_{i=1}^{n}\left(k^{3k}n^{2k}\phi^{1/2}\epsilon^{1/2}\right)^i\right\} \\ &\leq \sum_{j=1}^{n}(((nk)^{c_1k}(\phi^{c_2k}\epsilon)^{1/c_3})^j,\end{aligned}$$

for $c_1 \leq 5, c_2 \leq 8$ and $c_3 \leq 4$.

Applying Theorem 4.11, for $N = nk$, $N^{r+\beta} \leq N^3$, and $\Lambda \leq k^n$, we get $f(N) = O(k)$, $g'(N) = 8k$ and $g(N) = O(1)$. The expected running time is $O(\phi^{8k} \cdot (nk)^3 \cdot O(1) \cdot (nk)^{O(k)}nln(k))$, which is polynomial in (n^k, ϕ^k). □

5 Smoothness-Preserving Reductions to Local-Max-Cut and -Bisection

Recall Definition 3.3 in Section 3.2, where we have defined a notion of smoothness preserving reductions. $\mathcal{P}$ admits a smoothness-preserving reduction to $\mathcal{Q}$ if they admit a Karp reduction where the real-valued parameters of the reduced instance are given by a full-rank, linear combination of the real-valued parameters of $\mathcal{P}$, with coefficients of polynomial size. These preserve density bounds, and a weaker form of independence. The reduction is *strong* if the linear combination is simply a rescaling of the entries, *i.e.* given by a diagonal matrix. We also recall the definitions of local-max-cut and local-max-bisection, which consist of finding a cut in a graph whose cut value is maximal up to flipping one node across the cut, or swapping a pair of nodes across the cut, respectively.

In this section, we prove that smoothness-preserving reductions are sufficient to translate smoothed-efficient algorithms, and give the two reductions outlined in Section 3.2, proving that they are smoothness-preserving.

5.1 Smoothness-Preserving Reductions

We begin by including, for completeness, a proof of Proposition 2.1 and Lemma 3.2, which is exactly the proof given in [ER17], a simplification of the proof from [Rög08].

Proposition 2.1. ([Rög08]) *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 $y \in \mathbb{R}^d$.*

Proof, from [ER17]. Let $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_d$ be the standard basis vectors of $\mathbb{R}^d$, and let $\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_\ell$ denote the (linearly independent) rows of M. Without loss of generality, $\{\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_\ell, \boldsymbol{e}_{\ell+1} \ \ldots, \boldsymbol{e}_d\}$ is a basis for $\mathbb{R}^d$, and let $\widetilde{M}$ be the matrix whose rows are given by this basis. Let $\boldsymbol{x} \in \mathbb{R}^\ell$, and define $C_\epsilon(\boldsymbol{x}) := [x_1, x_1 + \epsilon] \times \cdots \times [x_\ell, x_\ell + \epsilon]$, a rectangular region in $\mathbb{R}^\ell$, and $\widetilde{C}_\epsilon(\boldsymbol{x}) := C_\epsilon(\boldsymbol{x}) \times \mathbb{R}^{d-\ell}$. We have $\Pr[MX \in C_\epsilon(\boldsymbol{x})] = \Pr[\widetilde{M}X \in \widetilde{C}_\epsilon(\boldsymbol{x})] = \Pr[X \in \widetilde{M}^{-1}\widetilde{C}_\epsilon(\boldsymbol{x})]$.

Observe that $\widetilde{M}^{-1}$ is the identity on the coordinates $d - \ell, \ldots, d$, since M is as well. Thus, we have that $\widetilde{M}^{-1}\widetilde{C}_\epsilon(\boldsymbol{x}) = C' \times \mathbb{R}^{d-\ell}$ for some region C' of volume at most ϵ^ℓ/η^ℓ. Therefore,

$$\begin{aligned}\Pr[X \in \widetilde{M}^{-1}\widetilde{C}_\epsilon(\boldsymbol{x})] &= \int_{C'} \mathrm{d}f_{X_1,\ldots,X_\ell} \cdot \int_{\mathbb{R}^{d-\ell}} \mathrm{d}f_{X_{\ell+1},\ldots,X_d} \\ &\leq \phi^\ell \cdot (\epsilon/\eta)^\ell \cdot 1\end{aligned}$$

Where the first integral bound comes from our assumption on the joint densities of collections of entries of X, and the second is simply integrating a probability density over the whole domain. Now taking the limit

$$\lim_{\epsilon \to 0} \frac{\Pr[MX \in C_\epsilon(\boldsymbol{x})}{\mathrm{vol}(C_\epsilon(\boldsymbol{x}))} \leq (\phi/\eta)^\ell$$

which gives our desired bound on the density. □

Lemma 3.2. ([Rög08]) *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 $b_1, b_2, \ldots, b_\ell \in \mathbb{R}$ and $\epsilon > 0$,$J_i \subset \mathbb{R}$ have measure ϵ, then*

$$\Pr\Big[MX \in [b_1, b_1 + \epsilon] \times \cdots \times [b_\ell, b_\ell + \epsilon]\Big] \leq (\phi\epsilon/\eta)^r \tag{7}$$

The proof of Proposition 2.1 (before taking the limit) is a proof for this lemma. We may now proceed to show that our definition of reduction does indeed preserve smoothed analysis results.

Lemma 5.1. *Let $\mathcal{Q}$ be a search problem with (quasi-)polynomial smoothed complexity, as defined in Section 2.2. Let $\mathcal{P}$ be a problem which admits a strong smoothness-preserving reduction to $\mathcal{Q}$, given by f_1, f_2, M, as in Definition 3.3. Then $\mathcal{P}$ has (quasi)polynomial smoothed complexity.*

Proof. The algorithm for instances of $\mathcal{P}$ is as follows:
1) Perform the randomized reduction,
2) Run the smoothed-(quasi-)polynomial-time algorithm for $\mathcal{Q}$ on the reduced instance,
3) Compute the solution to the instance of $\mathcal{P}$ given the solution to the reduced problem.

By the definition of smoothness-preserving reductions and (quasi-)polynomial smoothed complexity, step (2) will always correctly solve the reduced instance in finite time, and therefore step (3) will output a correct solution to the instance of $\mathcal{P}$.

It remains then to show that the algorithm runs in polynomial time with high probability, which we do via Markov's inequality, to control for the effect of the newly introduced randomness. Let (D, X) be an arbitrary instance of $\mathcal{P}$ where D is fixed, and X is a random vector whose entries are independently distributed with density bound ϕ. Let R be a random vector also independently distributed with density bound $\phi' = poly(\phi, |D|, |X|)$. Without loss of generality, $\phi' \geq \phi$. Since the reduction is *strong*, and the matrices are assumed to have integer entries, and therefore the entry-wise densities of the rescaling $M \cdot (X \circ R)$ also has densities bounded by ϕ'.

Let $\mathcal{A}$ be the smoothed efficient algorithm for $\mathcal{Q}$. Thus, there exist constants $c, c' > 0$ such that on random input (C, Y) with density bound ϕ', $\mathcal{A}$ runs in time $(\phi'|C||Y|)^c$ with probability $1 - 1/|C|^{c'}$. By definition, $|f_1(D)| \leq poly(|D|, |X|)$. We wish to show that with $1 - 1/poly(|D|, |X|)$ over the randomness in X and R, the reduced instance given by $C := f_1(D)$ and $Y := M \cdot (X \circ R)$ will be solved by $\mathcal{A}$ in time $poly(\phi, |D|, |X|)$, or quasi-polynomial time. By the assumptions on the performance of $\mathcal{A}$ for instances of $\mathcal{Q}$, this holds by definition, since the entries of $M(X \circ R)$ are independently distributed, and ϕ', $|C|$, and $|Y|$ are polynomial in ϕ, $|D|$, and $|X|$. $\square$

Corollary 5.2. *Let $\mathcal{Q}$ be a search problem with (quasi-)polynomial smoothed complexity when the input is arbitrarily distributed with a bound on the joint density as in the statements of Proposition 2.1 and Lemma 3.2. Let $\mathcal{P}$ be a problem which admits a* weak *smoothness-preserving reduction to $\mathcal{Q}$, then $\mathcal{P}$ has (quasi-)polynomial smoothed complexity.*

The proof of this corollary is identical to the above, combined with Proposition 2.1.

Corollary 5.3. *Let $\mathcal{P}$ be a problem which admits a* weak *smoothness-preserving reduction to local-max-cut, then $\mathcal{P}$ has quasi-polynomial smoothed complexity. If it admits a* weak *reduction to local-max-cut on a complete graph, then it has polynomial smoothed complexity.*

Proof. Following the discussion from Section 3.2, the proofs of the local-max-cut smoothed results from [ER17, ABPW17] consist of applying Lemma 3.2 directly to the edge weights of the graph, and finding bounds on the rank of the linear transformation. By Proposition 2.1, a weak reduction satisfies the conditions for the application of Lemma 3.2, and therefore the local-max-cut satisfies the conditions of the previous corollary, as desired. $\square$

We note, as discussed in previous sections, that it would also have sufficed for X to have joint density bounded by $\phi^{|X|}$. Observe that if it were possible to weakly reduce k-NetCoordNash to local-max-cut, then this would imply a (quasi-)polynomial smoothed complexity for k-NetCoordNash, where the degree of the polynomial does not depend on k. Unfortunately, we only achieve a weak reduction to local-max-bisection, which we believe has similar smoothed complexity to local-max-cut, though this is not as of yet known. We leave this as an open problem.

5.2 Reduction from 2-NetCoordNash to Local-Max-Cut

In this section, we give our first reduction from 2-NetCoordNash to local-max-cut, and show it satisfies the conditions of a smoothness-preserving reduction.

Let $G = (V, E)$ be a game graph, with payoff vector $A \in [0.5, 1]^{4|E|}$. As the inputs are assumed to lie in $[-1, 1]$, this is without loss of generality since we can make this transformation while preserving all distributional assumptions, and at most quadrupling the probability density in each coordinate. As defined in Section 3.2, we construct a cut graph $H = (V \cup \{s, t\}, E')$, where E' consists of the edges E over V, with an additional su and ut edge for all $u \in V$. We will define the edge weights below, where $w(u, v)$ denotes the weight of edge $uv \in E'$. Let $\Gamma(u)$ denote the neighbours of u in graph G, and $\underline{W}$ be a $|V|$-dimensional vector of extra randomness, assumed to be uniformly distributed in $[-1,-0.5]^{|V|}$. We set:

$$w(u,v) = \tfrac{1}{2}\Big(A((u,1)(v,2)) + A((u,2)(v,1)) - A((u,1)(v,1)) - A((u,2)(v,2))\Big) \quad \forall u,v \in V \tag{8a}$$

$$w(s,u) = \sum_{v:uv\in E} \Big[\tfrac{1}{2}\big(A((u,2)(v,1)) + A((u,2)(v,2))\big) + \underline{W}(u)\Big] \quad \forall u \in V \tag{8b}$$

$$w(u,t) = \sum_{v:uv\in E} \Big[\tfrac{1}{2}\big(A((u,1)(v,1)) + A((u,1)(v,2))\big) + \underline{W}(u)\Big] \quad \forall u \in V \tag{8c}$$

$$w(s,t) = (-1)\cdot \sum_{uv\in E} \Big[\tfrac{1}{2}\big(A((u,1)(v,2)) + A((u,2)(v,1))\big) + \underline{W}(u) + \underline{W}(v)\Big] \tag{8d}$$

Observe that the above are linear combinations of the input values, and the coefficients are $O(|E|)$-sized multiples of $\eta = \frac{1}{2}$.

Lemma 5.4. *The above construction satisfies the following conditions:*

1. *Cut values of s-t cuts are equal to the potential function of the associated strategy profiles,*
2. *All locally maximal cuts are s-t cuts,*
3. *The construction is full-rank*

Proof. Let (S,T) be a cut such that $s \in S$ and $t \in T$. We will do a quick case analysis for each payoff term. Note first that the $\underline{W}$ terms get cancelled by the st edge, since they must appear exactly once for s or t. We say u is "playing i according to the cut" if $u \in S$ when $i = 1$ or if $u \in T$ when $i = 2$. Suppose u is playing i and v is playing j, then the $A((u,i)(v,j))$ term is added with total weight 1 in the su, sv, ut, and vt edges, and if $i \neq j$, it is also added and removed in the uv and st edges, respectively, so it appears with total weight 1. If u is playing i but v is not playing j, then $A((u,i)(v,j))$ is added with weight $\frac{1}{2}$ in the su and ut edges, and it is subtracted with weight $\frac{1}{2}$ in uv if $i = j$, or st if $i \neq j$. Finally if u is not playing i and v is not playing j, then the term does not appear if $i = j$. Thus, the only terms that appear are the correct ones, and they appear with weight 1.

To show condition 2, first recall that all entries of $\underline{W}$ lie in $[-1,-0.5]$, as it is uniformly distributed in $[-1,-0.5]^{|V|}$. As observed above, in any s-t cut, the $\underline{W}$ terms are cancelled out, and by our assumption on the payoff values, we have cut values between $\frac{1}{2}|E|$ and $|E|$. Consider any cut (S,T) where $s,t \in S$. Then for every $u \in T$, we are contributing $+2\underline{W}(u)\cdot|\Gamma(u)|$ to the cut from su and ut edges, and at most four A terms for each uv edge with weight $\frac{1}{2}$ each, so the cut value must be non-positive, and switching node t to the other side will improve the cut value. Therefore, all locally maximal cuts are s-t cuts.

To show condition 3, we explicitly write out the matrix and show it has full row-rank by induction on the number of players. Let Id_n denote the $n \times n$ identity matrix, $\mathbf{1}_n$ denote the $1 \times n$ row of 1's, $\boldsymbol{\gamma}$ denote the vector of $|\Gamma(u)|$ values, and $\boldsymbol{\Gamma}$ be the diagonal matrix with diagonal $\boldsymbol{\gamma}$. Let $B_i \in \{0,1\}^{|V|\times 4|E|}$ denote the payoff-node incidence matrix of G, where the 1's are in the$(u,(u,i)(v,\cdot))$ entries. We have

$$\begin{pmatrix} \vdots \\ w(u,v) \\ \vdots \\ \hline \vdots \\ w(u,t) \\ \vdots \\ \hline w(s,t) \\ \hline \vdots \\ w(s,u) \\ \vdots \end{pmatrix} = \frac{1}{2} \left(\begin{array}{c|c} -\mathsf{Id} \otimes (-1,+1,+1,-1) & \mathbf{0} \\ \hline B_1 & 2\boldsymbol{\Gamma} \\ \hline \mathbf{1} \otimes (0,-1,-1,0) & -2\boldsymbol{\gamma} \\ \hline B_2 & 2\boldsymbol{\Gamma} \end{array}\right) \begin{pmatrix} \vdots \\ A((u,1)(v,1)) \\ A((u,1)(v,2)) \\ A((u,2)(v,1)) \\ A((u,2)(v,2)) \\ \vdots \\ \hline \vdots \\ \underline{W}(u) \\ \vdots \end{pmatrix} \tag{9}$$

Note that B_1 and B_2 have disjoint support, and $B_1 + B_2$ is an edge-node incidence matrix, tensored with $(1,1,1,1)$. We will show by induction on $|V|$ that this matrix is full-rank. For $n=2$, the matrix is explicitly

$$\begin{pmatrix} -1 & 1 & 1 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 2 & 0 \\ 1 & 0 & 1 & 0 & 0 & 2 \\ 0 & -1 & -1 & 0 & -2 & -2 \\ 0 & 0 & 1 & 1 & 2 & 0 \\ 0 & 1 & 0 & 1 & 0 & 2 \end{pmatrix}$$

which can be verified to have rank 6. Suppose then that the rank property holds for $|V| = n-1$, and we introduce a new node v. Then the new edges introduced to the matrix, restricted to the columns for edges with v and $\underline{W}(v)$, are of the form

$$\left(\begin{array}{ccccccccc|c} -1 & +1 & +1 & -1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & -1 & +1 & +1 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \hline\hline +1 & +1 & 0 & 0 & +1 & +1 & 0 & 0 & \cdots & -2\gamma_v \\ 0 & 0 & +1 & +1 & 0 & 0 & +1 & +1 & \cdots & -2\gamma_v \end{array}\right)$$

which has full row-rank. Since the entries in the omitted columns are all 0, then we have a block-upper-triangular matrix where the first diagonal block is the matrix for the graph $V \setminus v$, and the second diagonal block is this above matrix. Therefore, by induction, the matrix has full row rank, as desired. □

5.3 Reduction from k-NetCoordNash to Local-Max-Bisection

We present in this section the final reduction from k-NetCoordNash to Local-Max-Bisection.

The reduction will introduce some extra-randomness variables with density $2nk$. Recall, given a game graph $G = (V,E)$ and payoff vector $A \in \mathbb{R}^{|E|k^2}$, we have constructed in Section 3.2 the cut graph $H = (\{s_0, s_1, \ldots, s_{n(k-2)}, t\} \cup V \times [k], E')$, where there is an edge $\{s_z, (u,i)\}$ and $\{(u,i), t\}$ for all $u \in V$, $i \in [k]$, and $0 \le z \le n(k-2)$, there is an edge $\{(u,i),(u,j)\}$ for all $i \ne j \in [k]$, and an edge $\{(u,i),(v,j)\}$ for all $uv \in E$ and $i,j \in [k]$. Note that for complete game graphs, the resulting cut graph is also complete. We say a bisection (S,T) is valid if $|S| = |T| = |V(H)|/2$, $s_z \in S$ for all z, $t \in T$, and for each $u \in V$, there is exactly one $(u, \cdot)$ in S. To each valid bisection, we can associate the natural strategy profile $\boldsymbol{\sigma}(S,T)$ where $\sigma_u = i$ if $(u,i) \in S$, which is well-defined.

We wish to choose edge weights such that, as in the previous reduction, (a) the cut value of a valid bisection is equal to $\Phi(\boldsymbol{\sigma}(S,T))$, (b) all locally maximal bisections are valid, and (c) the construction is full-rank. As in the previous section, we assume without loss of generality that the entries of A are supported in $[2.5, 3]$. Let

- $\underline{W}^z(u,i) \sim \mathcal{U}[-\frac{3}{4}, -\frac{1}{2}]$, *i.i.d.* for all $0 \le z \le n(k-2)$, $u \in V$, and $i \in [k]$
- $\underline{R}(u,ij) \sim \mathcal{U}[-1, -\frac{1}{2}]$ *i.i.d.* for all $u \in V$ and $i \neq j \in [k]$
- $\underline{Y}(u,i) \sim \mathcal{U}[2, 2.5]$ *i.i.d.* for all $u \in V$ and $i \in [k]$
- $\underline{A}_0^z \sim \mathcal{U}[0, \frac{1}{2nk})$ *i.i.d.* for all $0 \le z \le n(k-2)$.

Note that the $\underline{A}_0^z$'s have density $2nk$, and all other variables have constant density. Define $\underline{A}_0 := \sum_{z=0}^{n(k-2)} \underline{A}_0^z$. Let $S_0 := \{s_0, \ldots, s_{n(k-2)}\}$, and for any valid cut (S,T), let $\psi(S) := \Phi(\boldsymbol{\sigma}(S,T))$. We will also extend ψ to be defined on invalid cuts. If there is no $(u,\cdot)$ node in S, say that $\sigma_u(S) = 0$, and in the definition of $\psi(S)$, let $A((u,i)(v,0)) := \underline{Y}(u,i)$ for all $u, v \in V$ and $i \in [k]$, and let $A((u,0)(v,0)) := \underline{A}_0$ for all $u, v \in V$. Let $\delta(S)$ denote the cut value of $(S, V \setminus S)$. We will construct edge weights with the following properties:

(i) For every valid (S,T), $\delta(S) = \psi(S)$. (From above)

(ii) For every $u \in V$, and $i \neq j \in [k]$, $\delta(S_0 \cup \{(u,i),(u,j)\}) = 2\underline{R}(u,ij)$

(iii) For every $u \in V$, $i \in [k]$, and $0 \le z \le n(k-2)$, $w(s_z,(u,i)) = \underline{W}^z(u,i)$.

Furthermore, we simply assume that for all $0 \le z < z' \le n(k-2)$, the weight of the $s_z s_{z'}$ edge is given by the random variable $\overline{W}(z,z')$, which are distributed *i.i.d.* uniformly along $[-1, -0.5]$. The correctness of the reduction will be proved using two lemmas, established using the following claim.

Claim 5.5. *Condition (i) is satisfied if (a) $\delta(S_0 \cup \{(u,i)\}) = \psi(S_0 \cup \{(u,i)\})$ for all players u and $1 \le i \le k$, and (b) $w((u,i),(v,j)) = \underline{Y}(u,i) - \underline{Y}(v,j) - A((u,i)(v,j)) - \underline{A}_0$.*

Proof. Let $S := S_0 \cup \{(u_1,i_1), \ldots, (u_\ell,i_\ell)\}$. We begin by showing the following:

$$\begin{aligned}\psi(S) = \Big[\sum_{j=1}^{\ell} \psi(S_0 \cup \{(u_j,i_j)\})\Big] - (\ell-1)\psi(S_0) \\ - \sum_{\substack{(u,i),(v,j)\in S \\ uv\in E}} 2\left[A((u,i)(v,0)) + A((u,0)(v,j)) - \underline{A}_0 - A((u,i)(v,j))\right] \qquad (10)\end{aligned}$$

$$\delta(S) = \Big[\sum_{j=1}^{\ell} \delta(S_0 \cup \{(u_j,i_j)\})\Big] - (\ell-1)\delta(S_0) - \sum_{\substack{(u,i),(v,j)\in S \\ uv\in E}} 2w((u,i),(v,j)) \qquad (11)$$

For (10), note first if there is no uv edge in the game graph, then $A((u,\cdot)(v,\cdot))$ does not appear on either side of the equality, and we may restrict our attention to pairs which form game edges. Now, for every v and w which do not appear in S, the left-hand-side has $2A((v,0)(w,0))$, and the right-hand-side has $2(\ell-(\ell-1))\underline{A}_0$ from the first line. If u appears with strategy i, and v does not appear in S, then the left-hand-side has $2A((u,i)(v,0))$, and the right-hand-side has $2A((u,i)(v,0))$ from the $\psi(S_0 \cup \{(u,i)\})$ term. If u appears with strategy i, and v appears with strategy j, then the left-hand-side has $2A((u,i)(v,j))$, and the right-hand-side has $2A((u,i)(v,0))$ and $2A((u,0)(v,j))$ from the $\psi(S_0 \cup \{(u,i)\})$ and $\psi(S_0 \cup \{(v,j)\})$ terms which are canceled out by the second line, $2(\ell-2-(\ell-1))\underline{A}_0$ terms from the first line which is canceled out by the second line, and the term $2A((u,i)(v,j))$ from the second line. A similar argument shows the validity of (11).

Since condition (i) requires that $\pi(S_0 \cup \{u,i\}) = \delta(S_0 \cup \{(u,i)\})$, this is necessary, and along with $\delta(S_0) = \psi(S_0)$ from (14), the above analysis shows it is sufficient. We are required to set $w((u,i)(v,j)) = A((u,i)(v,0)) + A((u,0)(v,j)) - \underline{A}_0 - A((u,i)(v,j))$, which is equal to $\underline{Y}(u,i) + \underline{Y}(v,j) - \underline{A}_0 - A((u,i)(v,j))$. Observe that this is supported on the interval $[1, 2.5]$. □

Lemma 5.6. *There exist edge weights w which satisfy conditions (i), (ii), and (iii). Moreover, these are a full-rank, square, integer-valued, linear combinations of the entries of A, $\underline{Y}(u,i)$, $\underline{A}_0^z$, $\underline{R}(u,ij)$, $\underline{W}^z(u,i)$, and $\overline{W}(z,z')$.*

Proof. We may ignore the rows indexed by $s_z s_{z'}$, as they depend only on the $\overline{W}$ values, and these values do not appear anywhere else, so they are independent, and do not affect the dependence of other rows. Next, we derive edge weights such that the conditions of Claim 5.5 hold, using the following system:

$$w(s_z,(u,i)) = \underline{W}^z(u,i) \qquad \text{(by def'n)} \tag{12}$$

$$w((u,i)(v,j)) = \underline{Y}(u,i) + \underline{Y}(v,j) - A((u,i)(v,j)) - \underline{A}_0 \tag{13}$$

$$\psi(S_0) = \delta(S_0) = \sum_{z=0}^{n(k-2)} w(s_z,t) + \sum_{u\in V}\sum_{i=1}^{k}\sum_{z=0}^{n(k-2)} \underline{W}^z(u,i) \tag{14}$$

$$\implies \sum_{z=0}^{n(k-2)} w(s_z,t) = \psi(S_0) - \sum_{u\in V}\sum_{i=1}^{k}\sum_{z=0}^{n(k-2)} \underline{W}^z(u,i)$$

$$\forall z \text{ we choose} : w(s_z,t) = \psi(S_0) - \sum_{u\in V}\sum_{i=1}^{k} \underline{W}^z(u,i) \tag{15}$$

$$\begin{aligned} 2\underline{R}(u,ij) = \delta(S_0 \cup \{(u,i),(u,j)\}) &= \delta(S_0 \cup \{(u,i)\}) + \delta(S_0 \cup \{(u,j)\}) - \delta(S_0) - 2w((u,i)(u,j)) \\ \implies w((u,i)(u,j)) &= \tfrac{1}{2}\big(\, \psi(S_0 \cup \{(u,i)\}) + \psi(S_0 \cup \{(u,j)\}) - \psi(S_0) - 2\underline{R}(u,i,j)\,\big) \end{aligned} \tag{16}$$

$$\begin{aligned} &\psi(S_0 \cup \{(u,i)\}) = \\ &\delta(S_0 \cup \{(u,i)\}) = \sum_{z=0}^{n(k-2)} w(s_z,t) + w((u,i),t) + \sum_{(v,j)\neq(u,i)} \Big[\textstyle\sum_{z=0}^{n(k-2)} \underline{W}^z(v,j) + w((u,i)(v,j)) \Big] \\ &\implies w((u,i),t) = \psi(S_0 \cup \{(u,i)\}) - \sum_{z=0}^{n(k-2)} w(s_z,t) - \sum_{(u,i)\neq(v,j)} \Big[\textstyle\sum_{z=0}^{n(k-2)} \underline{W}^z(v,j) + w((u,i)(v,j)) \Big] \\ &\qquad = \psi(S_0 \cup \{(u,i)\}) - \psi(S_0) + \textstyle\sum_{z=0}^{n(k-2)} \underline{W}^z(u,i) - \sum_{(u,i)\neq(v,j)} w((u,i)(v,j)) \end{aligned} \tag{17}$$

We observe first that (17) contains as terms the previous numbered equations. Thus, it suffices to perform simple row-elimination to get $w((u,i),t) = \psi(S_0 \cup \{u,i\}) - \sum_{v,j}\sum_z \underline{W}^z(v,j)$. Now, let $G = (V, E)$ be the underlying game graph, and let $d(u)$ be the degree of u in G. Then $\psi(S_0) = 2|E|\underline{A}_0$, and $\psi(S_0 \cup \{(u,i)\}) = \psi(S_0) + 2d(u)[\underline{Y}(u,i) - \underline{A}_0]$. Finally, we have

$$\left(\begin{array}{c} w((u,i)(v,j)) \\ \vdots \\ \hline w((u,i)(u,j)) \\ \vdots \\ \hline \widehat{w}((u,i),t) \\ \vdots \\ \hline w(s_z,t) \\ \vdots \\ \hline w(s,(u,i)) \\ \vdots \end{array}\right) = \left(\begin{array}{c|c|c|c|c} -\mathsf{Id}_{|E|k^2} & \mathbf{0} & * & -\mathbf{1}_{|E|k^2 \times n(k-2)+1} & \mathbf{0} \\ \hline \mathbf{0} & -\mathsf{Id}_{n\binom{k}{2}} & * & * & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & 2d(u)\mathsf{Id}_{nk} & * & \mathsf{Id} \otimes \mathbf{1} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & 2|E| \cdot \mathsf{Id}_{n(k-2)+1} & -\mathbf{1} \otimes \mathsf{Id} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathsf{Id} \otimes \mathbf{1} \end{array}\right) \left(\begin{array}{c} A_{uv}(i,j) \\ \vdots \\ \hline \underline{R}(u,ij) \\ \vdots \\ \hline \underline{Y}(u,i) \\ \vdots \\ \hline \underline{A}_0^z \\ \vdots \\ \hline \underline{W}^z(u,i) \\ \vdots \end{array}\right) \tag{18}$$

Where $\otimes$ denotes the tensor product, namely,

$$\mathsf{Id} \otimes \mathbf{1} = \begin{pmatrix} 1 & \cdots & 1 & 0 & \cdots & 0 & 0 & \cdots & 0 & \cdots \\ 0 & \cdots & 0 & 1 & \cdots & 1 & 0 & \cdots & 0 & \cdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & 1 & \cdots & 1 & \cdots \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots \end{pmatrix} \qquad \mathbf{1} \otimes \mathsf{Id} = \begin{pmatrix} 1 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & \cdots \\ 0 & 1 & \cdots & 0 & 0 & 1 & \cdots & 0 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots \\ 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 1 & \cdots \end{pmatrix}$$

It is easy to check that the $*$ values are integral, since the ψ values must be even combinations of the A values. Therefore, after the row-operations leading to $\widehat{w}((u,i),t)$ values, the matrix is upper-triangular, which implies that the system is full-rank, square, and integral, as desired. □

Lemma 5.7. *If conditions (i), (ii), and (iii) are satisfied, then all local-max-bisections are valid cuts, and their associated strategy profiles are Nash equilibria.*

Proof. Recall that we have assumed that $0.5 \leq A_{uv}(i,j) \leq 1$ for all edges uv and for all $1 \leq i,j \leq k$, that $0 \leq \underline{A}_0, \underline{Y}(u,i) < 0.5$ for all players u and $1 \leq i \leq k$ (Since $\underline{A}_0 = \sum_{z=0}^{n(k-2)} \underline{A}_0^z$, and the latter is contained in $[0, \frac{1}{2nk})$), and that $-1 \leq \underline{R}(u,ij), \underline{W}^z(u,i), \overline{W}(z,z') < -0.5$ for all players u, $1 \leq i < j \leq k$, $0 \leq z < z' \leq n(k-2)$.

We will show that from any non-valid cut, there will be a single flip operation towards a valid cut which improves the total cut value, then argue that they may be paired up into swap operations. Fix a bisection (S,T), and consider the following cases:

Case I. ***$t \in T$, and S contains at least half the s_z's, but $s_z \in T$ for some z.*** — Let s_z in T, we argue that $\delta(S \cup \{s_z\}) - \delta(S) > 0$. The two cuts may only differ on edges incident to s_z. The positive term includes $w(s_z,t)$, $\underline{W}^z(u,i)$ for all $(u,i) \notin S$, and the $\overline{W}(z,z')$ for all $s_{z'} \in T$, and the negative term includes $\underline{W}^z(u,i)$ for all $(u,i) \in S$ and the $\overline{W}(z,z')$ for all $s_{z'} \in S$. However, we know from (15) that $w(s_z,t) = \psi(S_0) - \sum_{(u,i)} \underline{W}^z(u,i)$. Therefore, we get

$$\delta(S \cup \{s_z\}) - \delta(S) \quad = \quad 2|E| \underbrace{\underline{A}_0}_{\geq 0} - 2\sum_{(u,i) \in S} \underline{W}^z(u,i) + \sum_{z: s_z \in T} \overline{W}(z,z') - \sum_{z: s_z \in S} \overline{W}(z,z')$$

Now, note that, since these are bisections, $|\{(u,i) : (u,i) \in S\}| + |\{z' : s_{z'} \in S\}| = n(k-1)+1$, so

$$-2\sum_{(u,i) \in S} \underline{W}^z(u,i) - \sum_{z: s_z \in S} \overline{W}(z,z') \quad \geq \quad \tfrac{1}{2}(n(k-1)+1)$$

and $\sum_{z:s_z\in T}\overline{W}(z,z')\geq -\frac{1}{2}(n(k-2)+1)$, since we have assumed more than half lie in S, so therefore the above sum is non-negative, and moving s_z into S was an improvement.

Case II. ***$t\in T$ and S contains fewer than half the s_z's.*** — We wish to show that in this case, $\delta(S\cup\{t\})-\delta(\{S\})\geq 0$. Since there are fewer than $\frac{1}{2}(n(k-2)+1)$ s_z nodes in S, there must be at least $\frac{1}{2}(nk-1)$ (u,i) nodes on the S side. Without loss of generality, we may assume n or k is even, and so at least half of the (u,i) nodes are in S. Therefore, it would suffice to show that $w(t,(u,i))<0$, and $w(s_z,t)\geq 0$, so t would benefit from moving to the side with fewer s_z and more (u,i) nodes. We have $w(s_z,t)=\psi(S_0)-\sum_{v\in V}\sum_{i=1}^{k}\underline{W}^z(u,i)$. Since $\underline{W}^z(u,i)\leq 0$, and $\psi(S_0)=2|E|k^2\cdot\underline{A}_0\geq 0$, we have that $w(s_z,t)\geq 0$. Conversely,

$$\begin{aligned}
w(t,(u,i)) &= \psi(S_0\cup\{(u,i)\})-\psi(S_0)+\textstyle\sum_{z=0}^{n(k-2)}\underline{W}^z(u,i)-\sum_{(u,i)\neq(v,j)}w((u,i)(v,j))\\
&\leq \sum_{v:uv\in E}\underline{Y}(u,i)-\tfrac{1}{2}n(k-2)-\tfrac{1}{2}-k\sum_{v:uv\in E}(\tfrac{1}{2}+\tfrac{1}{2nk})-\sum_{j\neq i}w((u,i)(u,j))\\
&< d(u)\cdot\tfrac{5}{2}-\tfrac{k}{2}d(u)-\tfrac{1}{2}n(k-2)\\
&\leq 0
\end{aligned}$$

For $k\geq 3$ and $n\geq 2k$. This concludes the proof of the claim.

Case III. ***$t\in T$, S contains at least half of the s_z's, and also (u,i) and (u,j) for some u and $i\neq j$.*** — Recall from (17) that

$$\begin{aligned}
w((u,i),t) &= \psi(S_0\cup\{(u,i)\})-\psi(S_0)+\textstyle\sum_{z=0}^{n(k-2)}\underline{W}^z(u,i)-\sum_{(u,i)\neq(v,j)}w((u,i)(v,j))\\
&\geq \textstyle\sum_{z=0}^{n(k-2)}\underline{W}^z(u,i)-\sum_{(u,i)\neq(v,j)}w((u,i)(v,j))
\end{aligned}$$

Now,

$$\begin{aligned}
&\delta(S)-\delta(S\setminus\{(u,j)\})\\
&\quad= -w((u,i),t)+\sum_{s_z\in S}\underline{W}^z(u,j)-\sum_{s_z\in T}\underline{W}^z(u,j)+\sum_{(v,i)\neq(u,j)\in S}w((u,j)(v,i))-\sum_{(v,i)\in T}w((u,j)(v,i))\\
&\quad= -2\sum_{s_z\in T}\underline{W}^z(u,j)-2\sum_{(v,i)\neq(u,j)\in S}w((u,j)(v,i))\\
&\quad\leq \tfrac{3}{4}|S_0\cap T|-|\{(v,i)\neq(u,j)\in S: v\neq u\}|-\sum_{i\neq j:(u,i)\in S}w((u,j)(u,i))
\end{aligned}$$

Note that, since the cut is a bisection, we have $|\{(v,i)\neq(u,j)\in S:v\neq u\}|=n-1+|S_0\cap T|$, and also $w((u,i)(u,j))\geq 0$. Therefore the sum overall is ≥ 0 if $n\geq k-1$, as desired.

Therefore, from any non-valid cut, it is always an improvement to (1) ensure that t is opposite the majority of s_z nodes by swapping it with some redundant (u,i) node, which must exist, (2) ensure that all s_z nodes are on the same side by swapping them with redundant (u,i) nodes, which must exist. If all the s_z are on one side, and t on the other, there must be exactly n nodes of the form (u,i). If some player appears twice, then another player does not appear, and it is in our interest to swap the redundant node with any node of the missing player. Entering the node of the missing player into the cut is an improvement because all $\underline{Y}$ and $\underline{A}_0$ values are smaller than all in-game payoffs, with probability 1. Therefore, all locally-maximal bisections are valid.

Note that we have required $\underline{A}_0^z$ to be distributed over the interval $[0,\frac{1}{2nk})$, so the density bound must be at least $2nk$, which is polynomial. □

Claim 5.5, and Lemmas 5.6 and 5.7 prove the correctness of the reduction, thus establishing the following theorem.

Theorem 5.8. *There is a smoothness preserving reduction from k-NetCoordNash to Local-Max-Bisection. The reduction maps k-NetCoordNash instances defined on complete graphs to Local-Max-Bisection instances on complete graphs.*

6 Additional Related Work

Smoothed Analysis. The work of Spielman and Teng [ST04] introduced the smoothed analysis framework to study good empirical performance of classical Simplex method for linear programs (LP). They showed that introducing independent random perturbations to any given (adversarial) LP instance, ensures that the Simplex method 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 [DMadHR+03, BV04, MR05, RV07, AV09, ERV14, ER17, ABPW17], 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.

Worst-case analysis of equilibrium computation. There has been extensive work on various potential games, equivalently congestion games (e.g., [Ros73, MS96, RT02, FPT04, CMN05]), capturing routing and traffic situations (e.g., [Smi79, DN84, Rou07, HS10, HHKS13, ADTW03]), and resource allocation under strategic agents (e.g., [JT04, FT12a, FT12b]). Unlike general games, existence of the potential function ensures that these games always have a pure NE [Ros73]. Finding pure NE is typically PLS-complete [FPT04, CD11], while finding any NE, mixed or pure, is known to be in CLS (Continuous Local Search) [DP11], a class in the intersection of PPAD (Polynomial Parity Argument on Directed graphs) and PLS. Another series of remarkable works have studied the loss in welfare at NE through the notions of Price-of-Anarchy and Price-of-Stability (e.g., [KP99, RT02, CK05, ADG+06, ADK+08, AFM09, RST17]). Our approach should help provide ways to obtain smoothed efficient algorithms for these games.

Worst case complexity of NE computation in general non-potential games has been studied extensively. The computation is typically PPAD-complete, even for various special cases (e.g., [AKV05, CDT06b, Meh14, FTC10]) and approximation (e.g., [CDT06a, Rub18]). On the other hand efficient algorithms have been designed for interesting sub-classes (e.g., [KT07, TS07, IKL+11, AGMS11, CD11, CCDP15, ADH+16, BB17, Bar18]), exploiting the structure of NE for the class to either enumerate, or through other methods such as parametrized LP and binary search. For two-player games, Lipton, Mehta, and Markakis gave a quasi-polynomial time algorithm to find a constant approximate Nash equilibrium [LMM03]. Recently, Rubinstein [Rub17] showed this to be the best possible assuming exponential time hypothesis for PPAD, and Kothari and Mehta [KM18] showed a matching unconditional hardness under the powerful algorithmic framework of Sum-of-Squares with oblivious rounding and enumeration. These results are complemented by communication [BR17, GR18] and query complexity lower bounds [Bab16, GR16, FS16]. Lower bounds in approximation under well-accepted assumptions have been studied for the decision versions [GZ89, CS08, HK11, BKW15, DFS16].

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

References

[ABPW17] Omer Angel, Sébastien Bubeck, Yuval Peres, and Fan Wei. Local max-cut in smoothed polynomial time. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 429–437. ACM, 2017.

[ADG+06] Sebastian Aland, Dominic Dumrauf, Martin Gairing, Burkhard Monien, and Florian Schoppmann. Exact price of anarchy for polynomial congestion games. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 218–229. Springer, 2006.

[ADH+16] AmirMahdi Ahmadinejad, Sina Dehghani, MohammadTaghi Hajiaghayi, Brendan Lucier, Hamid Mahini, and Saeed Seddighin. From duels to battlefields: Computing equilibria of Blotto and other games. In *AAAI*, pages 376–382, 2016.

[ADK+08] Elliot Anshelevich, Anirban Dasgupta, Jon Kleinberg, Eva Tardos, Tom Wexler, and Tim Roughgarden. The price of stability for network design with fair cost allocation. *SIAM Journal on Computing*, 38(4):1602–1623, 2008.

[ADTW03] Elliot Anshelevich, Anirban Dasgupta, Eva Tardos, and Tom Wexler. Near-optimal network design with selfish agents. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 511–520. ACM, 2003.

[AFM09] Nir Andelman, Michal Feldman, and Yishay Mansour. Strong price of anarchy. *Games and Economic Behavior*, 65(2):289–317, 2009.

[AGMS11] Bharat Adsul, Jugal Garg, Ruta Mehta, and Milind Sohoni. Rank-1 bimatrix games: A homeomorphism and a polynomial time algorithm. In *ACM Symposium on the Theory of Computing (STOC)*, pages 195–204, 2011.

[AKV05] T. Abbott, D. Kane, and P. Valiant. On the complexity of two-player win-lose games. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pages 113–122, Oct 2005.

[Ara17] Radhika Arava. Social network analysis using coordination games. *arXiv preprint arXiv:1708.09570*, 2017.

[AV09] D. Arthur and S. Vassilvitskii. Worst-case and smoothed analysis of the ICP algorithm, with an application to the k-means method. *SIAM Journal on Computing*, 39(2):766–782, 2009.

[Bab16] Yakov Babichenko. Query complexity of approximate nash equilibria. *Journal of the ACM (JACM)*, 63(4):36, 2016.

[Bar18] Siddharth Barman. Approximating Nash equilibria and dense subgraphs via an approximate version of Carathéodory's theorem. *SIAM Journal on Computing*, 47(3):960–981, 2018.

[BB17] Maria-Florina Balcan and Mark Braverman. Nash equilibria in perturbation-stable games. *Theory of Computing*, 13(1):1–31, 2017.

[BBWS17] Joris Broere, Vincent Buskens, Jeroen Weesie, and Henk Stoof. Network effects on coordination in asymmetric games. *Scientific reports*, 7(1):17016, 2017.

[BCC19] Ali Bibak, Charles Carlson, and Karthekeyan Chandrasekaran. Improving the smoothed complexity of flip for max cut problems. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 897–916. SIAM, 2019.

[BCE+98] Paul Beame, Stephen Cook, Jeff Edmonds, Russell Impagliazzo, and Toniann Pitassi. The relative complexity of NP search problems. *Journal of Computer and System Sciences*, 57(1):3 – 19, 1998.

[BKW15] Mark Braverman, Young Kun Ko, and Omri Weinstein. Approximating the best Nash equilibrium in $n^o(\log n)$-time breaks the exponential time hypothesis. In *ACM-SIAM Annual Symposium on Discrete Algorithms (SODA)*, pages 970–982, 2015.

[BPR15] N. Bitansky, O. Paneth, and A. Rosen. On the cryptographic hardness of finding a Nash equilibrium. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 1480–1498, Oct 2015.

[BR17] Yakov Babichenko and Aviad Rubinstein. Communication complexity of approximate nash equilibria. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 878–889. ACM, 2017.

[BV04] Rene Beier and Berthold Vöcking. Random knapsack in expected polynomial time. *J. Comput. Syst. Sci.*, 69(3):306–329, November 2004.

[BVV07] Imre Bárány, Santosh Vempala, and Adrian Vetta. Nash equilibria in random games. *Random Struct. Algorithms*, 31(4):391–405, December 2007.

[CCDP15] Yang Cai, Ozan Candogan, Constantinos Daskalakis, and Christos Papadimitriou. A multiplayer generalization of the minmax theorem. *Mathematics of Operation Research, Forthcoming*, 2015.

[CD11] Yang Cai and Constantinos Daskalakis. On minmax theorems for multiplayer games. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 217–234, 2011.

[CDRP08] Bruno Codenotti, Stefano De Rossi, and Marino Pagan. An experimental analysis of Lemke-Howson algorithm. arXiv preprint arXiv:0811.3247, 2008.

[CDT06a] X. Chen, X. Deng, and S.-H. Teng. Computing Nash equilibria: Approximation and smoothed complexity. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 603–612, Oct 2006.

[CDT06b] Xi Chen, Xiaotie Deng, and Shang-Hua Teng. Sparse games are hard. In *Internet and Network Economics*, pages 262–273, 2006.

[CGKP11] Syngjoo Choi, Douglas Gale, Shachar Kariv, and Thomas Palfrey. Network architecture, salience and coordination. *Games and Economic Behavior*, 73(1):76–90, 2011.

[CK05] George Christodoulou and Elias Koutsoupias. The price of anarchy of finite congestion games. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 67–73. ACM, 2005.

[CMN05] Deeparnab Chakrabarty, Aranyak Mehta, and Viswanath Nagarajan. Fairness and optimality in congestion games. In *Proceedings of the 6th ACM Conference on Electronic Commerce*, pages 52–57, 2005.

[CS08] Vincent Conitzer and Tuomas Sandholm. New complexity results about Nash equilibria. *Games and Economic Behavior*, 63(2):621–641, 2008.

[DFIS16] Argyrios Deligkas, John Fearnley, Tobenna Peter Igwe, and Rahul Savani. An empirical study on computing equilibria in polymatrix games. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, AAMAS '16, pages 186–195, Richland, SC, 2016. International Foundation for Autonomous Agents and Multiagent Systems.

[DFS16] Argyrios Deligkas, John Fearnley, and Rahul Savani. Inapproximability results for approximate Nash equilibria. In *12th International Conference Web and Internet Economics (WINE)*, pages 29–43, 2016.

[DG16] Stéphane Durand and Bruno Gaujal. Complexity and optimality of the best response algorithm in random potential games. In Martin Gairing and Rahul Savani, editors, *Algorithmic Game Theory*, pages 40–51, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[DMadHR+03] Valentina Damerow, Friedhelm Meyer auf der Heide, Harald Räcke, Christian Scheideler, and Christian Sohler. Smoothed motion complexity. In Giuseppe Di Battista and Uri Zwick, editors, *Algorithms - ESA 2003*, pages 161–171, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[DN84] Stella Dafermos and Anna Nagurney. On some traffic equilibrium theory paradoxes. *Transportation Research Part B: Methodological*, 18(2):101–110, 1984.

[DP11] Constantinos Daskalakis and Christos Papadimitriou. Continuous local search. In *Proceedings of the Twenty-second Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '11, pages 790–804, Philadelphia, PA, USA, 2011. Society for Industrial and Applied Mathematics.

[Ell93] Glenn Ellison. Learning, local interaction, and coordination. *Econometrica: Journal of the Econometric Society*, pages 1047–1071, 1993.

[ER17] Michael Etscheid and Heiko Röglin. Smoothed analysis of local search for the maximum-cut problem. *ACM Transactions on Algorithms (TALG)*, 13(2):25, 2017.

[ERV14] 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, Jan 2014.

[FPT04] Alex Fabrikant, Christos Papadimitriou, and Kunal Talwar. The complexity of pure Nash equilibria. In *Proceedings of the Thirty-sixth Annual ACM Symposium on Theory of Computing*, STOC '04, pages 604–612, New York, NY, USA, 2004. ACM.

[FS16] John Fearnley and Rahul Savani. Finding approximate nash equilibria of bimatrix games via payoff queries. *ACM Transactions on Economics and Computation (TEAC)*, 4(4):25, 2016.

[FT12a] Michal Feldman and Tami Tamir. Conflicting congestion effects in resource allocation games. *Operations research*, 60(3):529–540, 2012.

[FT12b] Michal Feldman and Tami Tamir. Conflicting congestion effects in resource allocation games. *Operations research*, 60(3):529–540, 2012.

[FTC10] Uriel Feige and Inbal Talgam-Cohen. A direct reduction from k-player to 2-player approximate nash equilibrium. In *International Symposium on Algorithmic Game Theory*, pages 138–149. Springer, 2010.

[GR16] Paul W Goldberg and Aaron Roth. Bounds for the query complexity of approximate equilibria. *ACM Transactions on Economics and Computation (TEAC)*, 4(4):24, 2016.

[GR18] Mika Göös and Aviad Rubinstein. Near-optimal communication lower bounds for approximate nash equilibria. In *FOCS*, 2018.

[GZ89] I. Gilboa and E. Zemel. Nash and correlated equilibria: Some complexity considerations. *Games Econ. Behav.*, 1:80–93, 1989.

[HHKS13] Tobias Harks, Martin Hoefer, Max Klimm, and Alexander Skopalik. Computing pure Nash and strong equilibria in bottleneck congestion games. *Mathematical Programming*, 141(1):193–215, 2013.

[HK11] Elad Hazan and Robert Krauthgamer. How hard is it to approximate the best Nash equilibrium? *SIAM Journal on Computing*, 40(1):79–91, 2011.

[HS10] Martin Hoefer and Alexander Souza. Tradeoffs and average-case equilibria in selfish routing. *ACM Trans. Comput. Theory*, 2(1):2:1–2:25, 2010.

[IKL$^+$11] Nicole Immorlica, Adam Tauman Kalai, Brendan Lucier, Ankur Moitra, Andrew Postlewaite, and Moshe Tennenholtz. Dueling algorithms. In *Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing*, pages 215–224, 2011.

[JPY88] 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.

[JT04] Ramesh Johari and John N Tsitsiklis. Efficiency loss in a network resource allocation game. *Mathematics of Operations Research*, 29(3):407–435, 2004.

[KM18] Pravesh K. Kothari and Ruta Mehta. Sum-of-squares meets Nash: Lower bounds for finding any equilibrium. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, pages 1241–1248, 2018.

[KP99] Elias Koutsoupias and Christos Papadimitriou. Worst-case equilibria. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 404–413. Springer, 1999.

[KPH14] 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.

[KT07] Ravi Kannan and Thorsten Theobald. Games of fixed rank: A hierarchy of bimatrix games. In *ACM-SIAM Annual Symposium on Discrete Algorithms (SODA)*, pages 1124–1132, 2007. Available on arxiv since 2005.

[LMM03] Richard J. Lipton, Evangelos Markakis, and Aranyak Mehta. Playing large games using simple strategies. In *Proceedings of the 4th ACM Conference on Electronic Commerce*, EC '03, pages 36–41, New York, NY, USA, 2003. ACM.

[Meh14] Ruta Mehta. Constant rank bimatrix games are PPAD-hard. In *Proceedings of the Forty-sixth Annual ACM Symposium on Theory of Computing*, STOC '14, pages 545–554, New York, NY, USA, 2014. ACM.

[MR05] Bodo Manthey and Rüdiger Reischuk. Smoothed analysis of binary search trees. In Xiaotie Deng and Ding-Zhu Du, editors, *Algorithms and Computation*, pages 483–492, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[MS96] Dov Monderer and Lloyd S Shapley. Potential games. *Games and economic behavior*, 14(1):124–143, 1996.

[MS10] Andrea Montanari and Amin Saberi. The spread of innovations in social networks. *Proceedings of the National Academy of Sciences*, 107(47):20196–20201, 2010.

[Rög08] Heiko Röglin. The complexity of nash equilibria, local optima, and pareto-optimal solutions. *Fakultät für Mathematik, Informatik und Naturwissenschaften der Rheinisch-Westfälischen Technischen Hochschule*, 2008.

[Ros73] Robert W. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2(1):65–67, 1973.

[Rou07] Tim Roughgarden. Routing games, 2007.

[RST17] Tim Roughgarden, Vasilis Syrgkanis, and Eva Tardos. The price of anarchy in auctions. *Journal of Artificial Intelligence Research*, 59:59–101, 2017.

[RT02] Tim Roughgarden and Éva Tardos. How bad is selfish routing? *J. ACM*, 49(2):236–259, March 2002.

[Rub17] Aviad Rubinstein. Settling the complexity of computing approximate two-player Nash equilibria. *SIGecom Exch.*, 15(2):45–49, February 2017.

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

[RV07] Heiko Röglin and Berthold Vöcking. Smoothed analysis of integer programming. *Mathematical Programming*, 110(1):21–56, Jun 2007.

[Smi79] Mt J Smith. The existence, uniqueness and stability of traffic equilibria. *Transportation Research Part B: Methodological*, 13(4):295–304, 1979.

[ST04] 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.

[SvS04] Rahul Savani and Bernhard von Stengel. Exponentially many steps for finding a Nash equilibrium in a bimatrix game. In *IEEE Annual Symposium on Foundations of Computer Science*, pages 258–267, 2004.

[SY91] Alejandro A. Schäffer and Mihalis Yannakakis. Simple local search problems that are hard to solve. *SIAM J. Comput.*, 20(1):56–87, February 1991.

[TS07] Haralampos Tsaknakis and Paul Spirakis. An optimization approach for approximate Nash equilibria. In *Web and Internet Economics (WINE)*, pages 42–56, 2007.