CONSTANT RANK TWO-PLAYER GAMES ARE PPAD-HARD∗

RUTA MEHTA∗†

Abstract. Finding Nash equilibrium in a two-player normal form game (2-Nash) is one of the most extensively studied problem within mathematical economics as well as theoretical computer science. Such a game can be represented by two payoff matrices $A$ and $B$, one for each player. 2-Nash is PPAD-complete in general, while in case of zero-sum games ($B = -A$) the problem reduces to LP and hence is in P. Extending the notion of zero-sum, in 2005, Kannan and Theobald defined rank of game $(A, B)$ as $\text{rank}(A + B)$, e.g., rank-0 are zero-sum games. They gave an FPTAS for constant rank games, and asked if there exists a polynomial time algorithm to compute an exact Nash equilibrium (NE). Adsul et al. (2011) answered this question affirmatively for rank-1 games, leaving rank-2 and beyond unresolved.

In this paper we show that NE computation in games with rank $\geq 3$ is PPAD-hard, settling a decade long open problem. Interestingly, this is the first instance that a problem with an FPTAS turns out to be PPAD-hard. Our reduction bypasses graphical games and game gadgets, and provides a simpler proof of PPAD-hardness for NE computation in bimatrix games. In addition, we get:

• An equivalence between 2D-Linear-FIXP and PPAD, improving on a result of Etessami and Yannakakis (2007) on equivalence between Linear-FIXP and PPAD.
• NE computation in a bimatrix game with convex set of Nash equilibria is as hard as solving a simple stochastic game [16].
• Computing a symmetric NE of a symmetric bimatrix game with rank $\geq 6$ is PPAD-hard.
• Computing a $\frac{1}{\text{poly}(n)}$-approximate fixed-point of a (Linear-FIXP) piecewise-linear function is PPAD-hard.

The status of rank-2 games remains unresolved.

1. Introduction. Two player finite non-cooperative games constitute the most simple and fundamental model within game theory [33], and have been studied extensively for their computational and structural properties. Such a game can be represented by two payoff matrices $(A, B)$, one for each player, and therefore are also known as bimatrix games. von Neumann (1928) showed that in games where one player’s loss is other player’s gain ($B = -A$, zero-sum), the min-max strategies are stable [41]. This turned out to be equivalent to linear programming (LP) [18, 3] and therefore polynomial-time computable. In 1950, John Nash [34] extended this notion to formulate an equilibrium concept, and proved its existence for finite multi-player games. It has since been named Nash equilibrium (NE) and is perhaps the most important and well-studied solution concept in game theory.

The classical Lemke-Howson algorithm (1964) [28], to compute Nash equilibrium in general bimatrix games, performs very well in practice. However it may take exponential time in the worst case [37]. Other methods that followed [27, 40] are also similar in nature [8, 24], and a complexity theoretic study of the problem was called for. Henceforth, by 2-Nash we mean computing a Nash equilibrium of a bimatrix game.

An early result of Megiddo and Papadimitriou [31] indicated that the complexity of computing Nash equilibrium cannot be NP-hard unless NP = co-NP, and hence new classes were needed to understand its complexity. In 1994 Papadimitriou introduced complexity class PPAD [35], Polynomial Parity Argument for Directed graph, for problems with path following argument for existence, like Sperner’s lemma [38]. He showed that 2-Nash, among many other problems, is in PPAD. After more than a

∗A preliminary version of this paper appeared in the Proceedings of the 46th ACM Symposium on Theory of Computing.
†College of Computing, Georgia Institute of Technology, Atlanta. (mehta.ruta@gmail.com)
Research supported by NSF Grant CCF-1216019.
decade, the problem was shown to be PPAD-hard in a remarkable series of works [19, 15]. Chen et. al. [15] showed that even \( \frac{1}{\text{poly}(n)} \)-approximation of 2-Nash is PPAD-hard, i.e., if there is a fully polynomial-time approximation scheme (FPTAS) for 2-Nash then PPAD = P. This was followed by PPAD-hardness results for special classes of bimatrix games, like sparse games [14] and win-lose games [1], and their approximation were also shown to be PPAD-hard.

On the positive side, polynomial-time algorithms were developed for many special classes of games; see Section 1.2 for an overview of previous results. Among these, one of the most significant is the class of constant rank games introduced by Kannan and Theobald (2005); rank of game \((A, B)\) is defined as \(\text{rank}(A + B)\). They gave an FPTAS for constant rank games,\(^1\) and asked if there is an efficient algorithm to compute an exact NE in these games. Note that, rank-0 are zero-sum games, and therefore are polynomial-time solvable. For rank-1 games, Adsul et. al. [5] gave a polynomial time algorithm, by reducing the problem to 1-dimensional fixed-point, however rank-2 and beyond remained unresolved.

In this paper we show that NE computation in games with rank \(\geq 3\) is PPAD-hard, settling a decade long open problem. Since there is an FPTAS for constant rank games, this result comes as a surprise, because until now whenever a problem, in games or markets, was shown to be PPAD-hard, so was its approximation (i.e., no FPTAS unless PPAD = P) [15, 14, 1, 12, 26].

To obtain the result, we reduce 2D-Brouwer, a two dimensional discrete fixed-point problem which is known to be PPAD-hard [11], to a rank-3 game. The reduction is done in two steps. First we reduce 2D-Brouwer to 2D-Linear-FIXP; Linear-FIXP [22] is a class of fixed-point problems with polynomial piecewise-linear functions, and \(kD\)-Linear-FIXP is its subclass consisting of \(k\)-dimensional fixed-point problems. In the second step, we reduce an instance of 2D-Linear-FIXP to a rank-3 bimatrix game, such that a linear function of Nash equilibrium strategies of the resulting game gives fixed-points of the 2D-Linear-FIXP instance.

Our reduction completely bypasses the machinery of graphical games and game gadgets, central to the previous approaches, and instead exploits relations between LPs, linear complementarity problems (LCPs) and bimatrix games. Such a conceptual leap seems necessary to show hardness of constant rank games, since the game gadgets used previously inherently give rise to higher rank games. Our approach also provides a simpler proof for PPAD-hardness of 2-Nash, and may be of independent interest to show hardness for other problems as well as to understand connections between parameterized LPs and bimatrix games. We achieve further simplification by avoiding even the parameterized LP, but the resulting game turns out to be of high rank.

Apart from the hardness of constant rank games, a number of results follow as corollaries from our reduction. The first step shows PPAD-hardness of 2D-Linear-FIXP and thereby improves the equivalence result Linear-FIXP = PPAD of Etessami and Yannakakis to 2D-Linear-FIXP = PPAD. This also implies 2D-Linear-FIXP = Linear-FIXP: in other words the class of Linear-FIXP remains unchanged even when functions are restricted to two dimensions. Since an instance of 1D-Linear-FIXP can be solved in polynomial time using binary search, our result establishes a dichotomy between 1D and \(kD\), \(k \geq 2\) Linear-FIXP problems; the former are in P and the latter are PPAD-complete.

Our approach can be extended to reduce \(kD\)-Brouwer to \(kD\)-Linear-FIXP to

\(^1\) \(O(1/\epsilon)^{\text{poly}(n)}\) time algorithm to compute an \(\epsilon\)-approximate Nash equilibrium in a rank-\(k\) \(n \times n\) game of bit size \(L\).
rank-\((k+1)\) games, where the reduction from \(kD\)-Linear-FIXP to rank-\((k+1)\) games (almost) preserves the number of solutions. Using this, together with a result from [22], we show that bimatrix games with convex set of NE are no easier. In fact they are as hard as solving simple stochastic games, which are known to be in \(\text{NP} \cap \text{coNP}\) [16], however despite significant efforts its exact complexity remains open [17, 7]. Further, we can show that computing weak\(^2\) \(\frac{1}{\text{poly}(n)}\)-approximate fixed-point of a function in Linear-FIXP is also PPAD-hard. It will be interesting to see if this can be extended to show hardness of approximation in \(2\)-Nash.

Since NE computation in a rank-\(k\) game can be reduced to computing symmetric NE of a symmetric game with rank-\(2\) [36], we get that computing symmetric NE in symmetric games with rank \(\geq 6\) is PPAD-hard. Again computing symmetric NE in symmetric rank-0 games can be solved using LP, and for rank-1 a polynomial-time algorithm was obtained recently by Mehta et. al. [32]. This leaves the status of symmetric games with rank between 2 and 5 unresolved. Also the status of rank-2 bimatrix games remains unresolved.

1.1. Overview of the Reduction. In this section we explain the main intuition behind our reductions. We first obtain a simpler reduction from \(2D\)-Brouwer to \(2\)-Nash, and later extend it to ensure rank-3 for the resulting game. A common step in both is to reduce \(2D\)-Brouwer to \(2D\)-Linear-FIXP. We start with a brief description of both of these problems.

\(2D\)-Brouwer is a class of 2-dimensional discrete fixed-point problems, known to be PPAD-hard [13]. An instance of \(2D\)-Brouwer consists of a grid \(G_n = \{0, \ldots, 2^n - 1\} \times \{0, \ldots, 2^n - 1\}\) and a valid coloring function \(g : G_n \to \{0, 1, 2\}\) which satisfies some boundary conditions, and thereby ensures existence of a trichromatic unit square in the grid.\(^3\) The problem is to find one such trichromatic square (see Section 2.2 for details). Function \(g\) is specified by a Boolean circuit \(B\) with \(2n\) input bits; \(n\) bits to represent each of the two co-ordinate of a grid point.\(^4\)

Linear-FIXP [22] is a class of fixed-point problems with polynomial piecewise-linear functions. A function \(F : [0, 1]^n \to [0, 1]^n\) in Linear-FIXP is defined by a circuit, say \(C\), with \(n\) real inputs and outputs, and \{\(\max\), \(+, \ast\)\} operations, where \(\ast\) is multiplication by a rational constant (see Section 2.3 for details). Such a function has rational fixed-points of size polynomial in the input size [22]. We denote the class of \(k\)-dimensional fixed-point problems in Linear-FIXP by \(kD\)-Linear-FIXP.

Given circuit \(B\) of a \(2D\)-Brouwer instance, in Section 3 we construct a \(2D\)-Linear-FIXP circuit \(C\) such that all the fixed-points of the function \(F\) defined by \(C\) are in trichromatic unit squares of the grid \(G_n\). It is easy to simulate \(B\) in \(C\) by replacing \(\wedge, \vee\) and \(\neg\) with \(\min, \max\) and \((1 - x)\) respectively, if input to this simulation is guaranteed to be Boolean. To guarantee this, we need to extract bit representation of \(\lfloor p \rfloor\), for a \(p \in [0, 2^n - 1]\). Since \(\text{floor}\) is a discontinuous function it can not be simulated using Linear-FIXP circuit, whose operations can generate only continuous functions. However, we design a bit extraction gadget which does the job for almost all the points, except those that are close to the boundary of unit squares of \(G_n\). Finally, using a sampling lemma similar to that of [15] we ensure that the fixed-points of the function defined by the resulting circuit \(C\) are always in trichromatic unit squares of grid \(G_n\).

**Theorem 1.1 (Informal).** Computing a fixed-point of a Linear-FIXP instance

\(^2\)Vector \(x\) is a weak \(\varepsilon\)-approximate fixed-point of function \(f\) if \(\|x - f(x)\|_{\infty} \leq \varepsilon\)

\(^3\)This is similar to the Sperner’s lemma

\(^4\)We use super-script \(b\) to differentiate Boolean circuits from Linear-FIXP circuits that will follow.
with $k$ inputs and $k$ outputs, with $k > 1$, is PPAD-hard. In other words, 2D-Linear-FIXP = PPAD.

Etessami and Yannakakis [22] showed that Linear-FIXP = PPAD. Theorem 1.1 improves this to 2D-Linear-FIXP = PPAD, and in turn we get Linear-FIXP = 2D-Linear-FIXP, i.e., fixed-point problems with polynomial piecewise-linear functions in constant (two) dimension are as hard as those in $n$-dimension.

In the simpler reduction, which gives arbitrary rank game, next we reduce Linear-FIXP problem to a two-player game, via LCP. Let $\lambda = (\lambda_1, \ldots, \lambda_k)$ denote the $k$ inputs of circuit $C$ of the given $kD$-Linear-FIXP instance. In Section 4 we construct an LCP that exactly captures the fixed-points of this circuit. LCP stands for Linear Complementarity Problem where given a square matrix $M$ and a vector $q$ find $y$ such that $My \leq q$, $y \geq 0$ and $((My)_i - q_i)y_i = 0$, $\forall i$.

This is done as follows: There is an ordering among max gates since $C$ forms a DAG. Suppose $x_i$ captures the output of $i^{th}$ max gate. Since the + and $\zeta$ operations of circuit $C$ generates only linear expressions, for $x_j = \max\{L, R\}$, $L$ and $R$ both are linear expression in $x_1, \ldots, x_{j-1}$ and $\lambda$. Further, such a max operation is equivalent to $x_j \geq L, x_j \geq R, (x_j - L)(x_j - R) = 0$. This is like a complementarity condition. To bring it in a standard LCP form we replace $x_j$ with $y_j + L$, $y_j$ being a new variable, and therefore the conditions instead become $y_j \geq 0, y_j + L - R \geq 0$, and $y_j(y_j + L - R) = 0$, i.e., exactly in the form of LCP. Note that still expressions $L$ and $R$ have $\lambda$ as variables, which do not have any complementarity condition. To get rid of them we use the fact that at a fixed-point input is same as output, in particular $\lambda_i = i^{th}$ output.

Next we show that matrix of the above LCP is strictly semi-monotone and therefore, using the result of Adler and Verma [4], the LCP can be reduced to finding a symmetric NE of a symmetric two-player game. It turns out that symmetric NE of the constructed game are in one-to-one correspondence with the fixed-point of the original Linear-FIXP function. A standard reduction from symmetric NE to NE gives PPAD-hardness for bimatrix games. As consequences we also get that Nash equilibrium computation in bimatrix games with convex set of Nash equilibria (Theorem 4.7), and computing a unique symmetric NE of a symmetric game (Corollary 4.6), both are as hard as solving a simple stochastic game, since the latter reduces to finding a unique fixed-point of a Linear-FIXP problem [22].

**THEOREM 1.2 (Informal).** The problem of computing a Nash equilibrium of a bimatrix game is PPAD-hard. Even if the set of NE is convex, the problem remains at least as hard as solving a simple stochastic game.

To get a lower rank game, in Section 5 we reduce circuit $C$ defining a $kD$-Linear-FIXP problem to rank $(k+1)$ game. Instead of an LCP we construct a parameterized linear program $LP(\lambda)$, so that circuit evaluation for a given input $\lambda$ is same as solving the LP. The two linear conditions of the above LCP, namely $y_j \geq 0$ and $y_j + L - R \geq 0$, defines feasible region of $LP(\lambda)$. Here we keep $\lambda$ in expressions $L$ and $R$ as is, and therefore r.h.s. of the constraints of LP is parameterized by $(\lambda_1, \ldots, \lambda_k)$. Further, since $L$ and $R$ both are linear expressions in $y_1, \ldots, y_j$ the constraint matrix is lower triangular. Using this property we show that $\exists \lambda$ such that for all $\lambda$, $\min : c^T x$ over this feasible region will ensure the quadratic constraints as well for each max gate. This gives the $LP(\lambda)$ which can replace circuit $C$.

Since primal-dual feasibility, and complementary slackness characterizes solutions

---

5To keep the presentation complete, we describe the reduction from (our) LCP to a symmetric game in Appendix A.
of an LP, LP is a special case of LCP. Using this connection for $LP(\lambda)$, we construct an $LCP_C$ whose solutions exactly capture the fixed-point of the given $kD$-Linear-FIXP instance (Section 5.2). Further, the matrix of the LCP turns out to be off-block-diagonal, with the two blocks in off-diagonal adding up to a rank-$k$ matrix. Finally, using the fact that the LCP capturing Nash equilibria of a bimatrix game also has an off-block-diagonal matrix, we construct a bimatrix game, whose Nash equilibria are in one-to-one correspondence with the solutions of $LCP_C$. The rank of the resulting game turns out to be $(k + 1)$, and one of its payoff matrix is upper-triangular.

**Theorem 1.3 (Informal).** Nash equilibrium computation in bimatrix games with rank $\geq 3$ is PPAD-hard, even when one of the payoff matrix is lower/upper triangular.

Theorem 1.3, together with the reduction from 2-Nash to symmetric 2-Nash [36], implies that computing symmetric NE of a symmetric game with rank $\geq 6$ is PPAD-hard. Further, this gives a simpler proof of PPAD-hardness of 2-Nash, i.e., without using the graphical games and game gadgets.

Finally, since we know that $\frac{1}{\text{poly}(L)}$-approximation of 2-Nash is also PPAD-hard [15], in an attempt to achieve similar result, in Section 6 we extend the first step of the reduction, to reduce $kD$-Brouwer to approximate $kD$-Linear-FIXP. We show that when an instance of $kD$-Brouwer with a $k$-dimensional grid $\{0, \ldots, 2^n - 1\}^k$ is reduced to an instance of $kD$-Linear-FIXP, not only exact fixed-points but also all the $\frac{1}{\text{poly}(k)}$-approximate fixed-points are in panchromatic unit cube of the grid. Chen et al. [15] showed that a class of $kD$-Brouwer with $n = 3$ is PPAD-hard, where $k$ is an input parameter and not a constant. Therefore, we get that $\frac{1}{\text{poly}(L)}$-approximation of Linear-FIXP is PPAD-hard (Theorem 6.5), where $L$ is the size of the input instance.

**Open:** It will be interesting to see if the above reduction can be extended to inapproximability in 2-Nash using techniques developed in Section 4. Importantly, our work leaves the status of rank-2 games, and symmetric games with rank between 2 and 5, unresolved.

### 1.2. Related Work.

Efficient algorithms have been designed for many special classes of bimatrix games. Lipton et. al. [29] gave a pseudo-polynomial time algorithm, which remains the best known bound till now. In addition, they gave a polynomial time algorithm for games where $\max\{\text{rank}(A), \text{rank}(B)\}$ is a constant. Later Garg et. al. [29] improved it to $\min\{\text{rank}(A), \text{rank}(B)\}$ being constant. Note that, these classes are restrictive and do not capture even all of zero-sum games. For random games, Bárány et. al. [9] showed that there exists a NE with support size 2 with $O(1 - 1/\log n)$ probability, and using this gave an algorithm which is efficient with high probability. A game is called win-lose game, if all the entries of $A$ and $B$ are either zero or one. Chen et. al. [14] gave a polynomial-time algorithm for win-lose sparse games, and Addario-Berry et. al. [2] gave one for win-lose planar games.

Many algorithms are designed to achieve constant factor approximation for 2-Nash [20, 10, 39]; the best known factor till now is 0.3393 due to Tsaknakis and Spirakis [39]. Although designing a polynomial time approximation scheme (PTAS) remains open, PTASs were designed for special classes, like Daskalakis and Papadimitriou [21] gave one for sparse games and games whose equilibria are guaranteed to have small-$O(1/n)$-values, and Alon et. al. [6] gave a PTAS for games with rank-$(\log n)$.

### 2. Preliminaries.

To show the hardness of rank-3 games, we start with $2D$-Brouwer, reduce it to Linear-FIXP and then to a bimatrix game. In this section we
discuss each of these problems separately. First we describe a characterization of Nash equilibria in bimatrix games, and the class of 2D-Brouwer problems. Both problems are known to be PPAD-complete [15, 13]. Next, we describe Linear-FIXP [22], and define its subclass $kD$-Linear-FIXP.

**Notation:** All the vectors are in bold-face letters, and are considered as column vectors. To denote a row vector we use $x^T$. The $i^{th}$ coordinate of the vector $x$ is denoted by $x_i$. $\mathbf{1}$ and $\mathbf{0}$ represent all ones and all zeros vector respectively of appropriate dimension. We use $[n]$ to denote the set $\{1, \ldots, n\}$.

### 2.1. Bimatrix games and Nash equilibrium

A bimatrix game is a two player game, each player having finitely many pure strategies (moves). Let $S_i$, $i = 1, 2$ be the set of strategies of player $i$, and let $m \overset{\text{def}}{=} |S_1|$ and $n \overset{\text{def}}{=} |S_2|$. Then such a game can be represented by two payoff matrices $A$ and $B$, each of $m \times n$ dimension. If the first player plays strategy $i$ and the second plays $j$, then the payoff of the first player is $A_{ij}$ and that of the second player is $B_{ij}$. Note that the rows of these matrices correspond to strategies of the first player and columns to strategies of the second player.

Players may randomize among their strategies; a randomized play is called a mixed strategy. Let the set of mixed strategies for the first player be $X = \{x = (x_1, \ldots, x_m) \mid x \geq 0, \sum_{i=1}^m x_i = 1\}$, and for the second player be $Y = \{y = (y_1, \ldots, y_n) \mid y \geq 0, \sum_{j=1}^n y_j = 1\}$. By playing $(x, y) \in X \times Y$ we mean strategies are picked independently at random as per $x$ by the first-player and as per $y$ by the second-player. Therefore the expected payoffs of the first-player and second-player are, respectively

$$\sum_{i,j} A_{ij} x_i y_j = x^T A y \quad \text{and} \quad \sum_{i,j} B_{ij} x_i y_j = x^T B y$$

**Definition 2.1.** (Nash Equilibrium [42]) A strategy profile is said to be a Nash equilibrium strategy profile (NESP) if no player achieves a better payoff by a unilateral deviation [34]. Formally, $(x, y) \in X \times Y$ is a NESP iff $\forall x' \in X, x^T A y \geq x'^T A y$ and $\forall y' \in Y, x^T B y \geq x^T B y'$.

Given strategy $y$ for the second-player, the first-player gets $(Ay)_k$ from her $k^{th}$ strategy. Clearly, her best strategies are $\arg \max_k (Ay)_k$, and a mixed strategy fetches the maximum payoff only if she randomizes among her best strategies. Similarly, given $x$ for the first-player, the second-player gets $(x^T B)_k$ from $k^{th}$ strategy, and same conclusion applies. These can be equivalently stated as the following complementarity type conditions,

$$\forall i \in S_1, \ x_i > 0 \Rightarrow (Ay)_i = \max_{k \in S_1} (Ay)_k$$
$$\forall j \in S_2, \ y_j > 0 \Rightarrow (x^T B)_j = \max_{k \in S_2} (x^T B)_k$$

The next lemma follows from the above discussion.

**Lemma 2.2.** Strategy profile $(x, y) \in X \times Y$ is a NE of game $(A, B)$ if and only if the following holds, where $\pi_1$ and $\pi_2$ are scalars capturing respective payoffs at $(x, y)$.

$$\forall i \in S_1, \ (Ay)_i \leq \pi_1; \quad x_i((Ay)_i - \pi_1) = 0$$
$$\forall j \in S_2, \ (x^T B)_j \leq \pi_2; \quad y_j((x^T B)_j - \pi_2) = 0$$
Game \((A, B)\) is said to be symmetric if \(B = A^T\). In a symmetric game the strategy sets of both the players are identical, i.e., \(m = n\), \(S_1 = S_2\) and \(X = Y\). We will use \(n\), \(S\) and \(X\) to denote number of strategies, the strategy set and the mixed strategy set respectively of the players in such a game. A Nash equilibrium profile \((x, y) \in X \times X\) is called symmetric if \(x = y\). Note that at a symmetric strategy profile \((x, x)\) both the players get payoff \(x^T A x\). Using Lemma 2.2 we get the following.

**Lemma 2.3.** Strategy profile \(x \in X\) is a symmetric NE of game \((A, A^T)\), with payoff \(\pi\) to both players, if and only if,

\[
\forall i \in S, (Ax)_i \leq \pi; \quad x_i((Ax)_i - \pi) = 0
\]

The problem of computing a Nash equilibrium strategy in bimatrix games is PPAD-complete [15, 19]. This also implies that computing symmetric NE of a symmetric bimatrix game is PPAD-hard, because NE of game \((A, B)\) are in one-to-one correspondence with the symmetric NE of game \((S, S^T)\) with \(S = \begin{bmatrix} 0 & A \\ B^T & 0 \end{bmatrix} \) [36].

**Rank of game \((A, B)\):** Motivated from the fact that \(A + B = 0\) in case of zero-sum games, which are polynomial time solvable [41, 18], Kannan and Theobald [25] defined rank of game \((A, B)\) as \(\text{rank}(A + B)\) and considered rank-based hierarchy of games with zero-sum (rank-0) at the base.

Next we describe a discrete version of a two-dimensional fixed-point problem, which has resemblance to Sperner’s Lemma.

**2.2. 2D-Brouwer.** Let \(G_n\) be a two dimensional grid \(\{0, \ldots, 2^n - 1\} \times \{0, \ldots, 2^n - 1\}\). Let \(g\) be a function that assigns colors 0, 1 or 2 to vertices of \(G_n\). Function \(g\) is said to be valid if for every vertex \((p_1, p_2)\) on the boundary of \(G_n\), we have

- If \(p_2 = 0\) then \(g(p) = 2\), else if \(p_2 > 0\) & \(p_1 = 0\) then \(g(p) = 1\), else \(g(p) = 0\)

Let \(K_p\) denote the unit square with \(p\) at the bottom left corner. Using Sperner’s Lemma [38] it follows that for any valid coloring \(g\) of \(G_n\) there exists a vertex \(p \in G_n\) such that vertices of square \(K_p\) has all the three colors - trichromatic.

**2D-Brouwer Mapping Circuit:** Valid coloring function \(g\) is thought of defined by a Boolean circuit \(B\) with \(2n\) input bits, \(n\) bits for each of the two integers representing a grid point, and 4 output bits \(\Delta^+_1, \Delta^-_1, \Delta^+_2, \Delta^-_2\).\(^{6}\) It is a valid Brouwer-mapping circuit if the following is true:

- For every \(p \in G_n\), the 4 output bits of \(B\) satisfies one of the following 3 cases:
  - Case 0: \(i = 1, 2, \Delta^-_1 = 1\) and \(\Delta^+_1 = 0\).
  - Case \(i, i = 1, 2, \Delta^+_1 = 1\) and all the other 3 bits are zero.
- For every \(p\) on the boundary of \(G_n\), if \(p_2 = 0\) then Case 2 is satisfied, if \(p_1 = 0\) and \(p_2 \neq 0\) then Case 1 is satisfied, and for the rest Case 0 is satisfied.

Such a circuit \(B\) defines a valid color assignment \(g_B : G_n \rightarrow \{0, 1, 2\}\) by setting \(g_B(p) = i\) if Case \(i\) is satisfied when the circuit is evaluated at \(p\).

**Definition 2.4 (2D-Brouwer [13]).** Input to a 2D-Brouwer problem consists of a valid Brouwer-mapping circuit \(B\) that produces a valid coloring on \(G_n\). The problem is to find a point \(p \in G_k\) such that \(K_p\) is trichromatic.

---

\(^6\)We use the definitions and some of the terminologies of [15] to remain consistent.
We denote size of a 2D-Brouwer problem defined by circuit $B$ as $\text{size}[B]$, which is
$\#\text{input nodes} + \#\text{output nodes} + \#\text{gates}$. Given $B$ in Section 3 we will construct a continuous function from the convex-hull of $G_n$ to itself. As a step towards this it helps to think of $B$ defining a discrete function from the $G_n$ to itself as follows: The outputs of a circuit (defining colors) can also be mapped to incremental vector $(\Delta^1_i - \Delta^{-1}_i, \Delta^2_i - \Delta^{-2}_i)$. Let $e^i$ be the incremental vector corresponding to Case (color) $i$, i.e., $e^0 = (-1, -1), e^1 = (1, 0)$ and $e^2 = (0, 1)$. Define a discrete function $H$, such that $H(p) = p + e^{g_k}(p)$. It is easy to see that if $B$ is a valid Brouwer-mapping circuit, then $H$ is $G_n \rightarrow G_n$, and vertices of a trichromatic square $K_p$ goes in each of the $e^i$ direction under $H$. Chen and Deng showed that finding such a square is PPAD-hard [13].

2.3. Linear-FIXP. Etessami and Yannakakis [22] defined a complexity class FIXP to capture complexity of finding exact fixed-point of an algebraic function; here solution consists of algebraic numbers but they may be irrational. Formally, an instance $I$ of FIXP consists of an algebraic circuit $C_I$ defining a function $F_I : \{0, 1\}^d \rightarrow \{0, 1\}^d$, and the problem is to compute a fixed-point of $F_I$. The circuit is a finite representation of function $F_I$ (like a formula), consisting of $\{\text{max}, +, \ast\}$ operations, rational constants, and $d$ inputs and outputs.

The circuit $C_I$ is a sequence of gates $g_1, \ldots, g_m$, where first $d$ of these correspond to the inputs, i.e., $\forall i \in [d]$, $g_i := \lambda_i$ is an input variable. Let there be $r$ constants used in the circuit, then for $d < i \leq d + r$, $g_i := c_i \in \mathbb{Q}$ is a rational constant, with numerator and denominator encoded in binary. For $i > d + r$ we have $g_i = g_j \circ g_k$, where $j, k < i$ and the binary operator $\circ \in \{\text{max}, +, \ast\}$. The last $d$ gates are the output gates. Note that the circuit forms a directed acyclic graph (DAG), when gates are considered as nodes and there is an edge from $g_j$ and $g_k$ to $g_i$ if $g_i = g_j \circ g_k$. Since circuit $C_I$ represents function $F_I$ it has to be the case that if we input $\lambda \in [0, 1]^d$ to $C_I$ then all the gates are well defined and the circuit outputs $C_I(\lambda) = F_I(\lambda)$ in $[0, 1]^d$. We note that a circuit representing a problem in FIXP operates on real numbers, but the underlying model of computation is still the standard discrete Turing machine. In other words, an algorithm for problems in FIXP is not allowed to do any exact computation on irrational numbers.

For piecewise-linear functions [22] defined a subclass Linear-FIXP. Let $\ast_\zeta$ denote multiplication by a rational constant $\zeta \in \mathbb{Q}$, then operations in a Linear-FIXP problem are restricted to $\circ \in \{\text{max}, +, \ast_\zeta\}$. A function defined by a Linear-FIXP circuit is polygonal piecewise-linear, and all its fixed-points are rational numbers of size $\text{poly}(L)$ [22], where $L$ is the size of the circuit which is $\#\text{inputs} + \#\text{gates} + \text{total size of the constants used in the circuit}$. Etessami and Yannakakis showed that PPAD = Linear-FIXP. Next, we define a subclass of Linear-FIXP based on the number of inputs and outputs.

**Definition 2.5.** For a $k \geq 1$, an instance $I$ is in $kD$-Linear-FIXP if $F_I : \{0, 1\}^k \rightarrow \{0, 1\}^k$. i.e., $F_I$ is defined by a circuit with $k$ inputs and $k$ outputs.

Since fixed-point of a 1-dimensional piecewise-linear function can be computed in polynomial time using a binary search, $kD$-Linear-FIXP is in $P$ for $k = 1$. But for any constant $k > 1$ it is not clear if the problem is in $P$ or it is hard. In the next section, we show that the problem is PPAD-hard even for $k = 2$.

3. **PPAD-hardness for 2D-Linear-FIXP.** In this section we construct a Linear-FIXP circuit with two inputs and two outputs from an instance of 2D-Brouwer. We show that the function defined by the resulting 2D-Linear-FIXP circuit is such that
all its fixed-points are in trichromatic squares of the 2D-Brouwer instance, thereby proving PPAD-hardness of 2D-Liner-FIXP using [13].

Let $B$ be the valid Brouwer-mapping circuit of a given 2D-Brouwer instance on grid $G_n$, and $H$ be the discrete function defined by circuit $B$. We extend $H$ to a continuous function $F : [0, 2^n - 1]^2 \rightarrow [0, 2^n - 1]^2$ defined by a Linear-FIXP circuit $C$.

Recall that given a bit representation of a grid point $p \in G_n$, circuit $B$ outputs four bits $\Delta^+_1, \Delta^-_1, \Delta^+_2, \Delta^-_2$, so that for $I = (\Delta^+_1 - \Delta^-_1, \Delta^+_2 - \Delta^-_2)$, $H(p) = p + I$. Similarly, for every non-grid point $p = (p_1, p_2) \in Kq$, we need to compute an incremental vector based on the incremental vectors of the vertices of $Kq$. For this we need to extract the integer parts of $p_1$ and $p_2$, i.e., compute $\lfloor p_1 \rfloor$ and $\lfloor p_2 \rfloor$, and then its bit representation. Since floor is a discontinuous function, it can not be computed using Linear-FIXP operations, which are inherently continuous. Instead of floor we construct a function which is same as floor except for points very near to an integer.

Recall that the operations allowed in a Linear-FIXP circuit are $\{\max, +, *, \zeta\}$. Clearly, $\{\min, -\}$ can be simulated using the allowed operations. Let $L > 16$ be a large integer with value being a power of 2, and at most polynomial in $\text{size}(B)$, i.e., $L = 2^L \leq \text{poly}(\text{size}(B))$. Consider the $\text{ExtractBits}$ procedure of Table 1.

| Table 1 |
| Extract Bits of the Integer Part |
| ExtractBits(a) |
| $x \leftarrow a$ |
| for $i=n-1$ to 0 do |
| $b_i \leftarrow \min\{\max\{(x - 2^i) \ast L^2 + 1, 0\}, 1\}$ |
| $x \leftarrow x - 2^ib_i$ |
| endfor |
| Output bit vector $b = (b_{n-1}, \ldots, b_0)$ |

**Definition 3.1.** We say that $a \in \mathbb{R}_+$ is poorly-positioned if for some integer $t \in \mathbb{Z}_+$, $a = t + \epsilon$, where $1 - \frac{1}{2^L} < \epsilon < 1$. A point $p \in \mathbb{R}^2_+$ is said to be poorly-positioned, if any of its coordinates is poorly-positioned, otherwise it is called well-positioned.

**Lemma 3.2.** Given a well-positioned number $a \in [0, 2^n)$, vector $b = \text{ExtractBits}(a)$ is a bit representation of $\lfloor a \rfloor$.

**Proof.** Let $a = a' + \epsilon$, where $a' \in \mathbb{Z}_+$ and $0 \leq \epsilon \leq 1 - \frac{1}{2^L}$. We show that every $b_i$ is either 0 or 1, and is set correctly. Proof is by induction. If $a' \geq 2^{n-1}$, then clearly, $(a - 2^{n-1}) \ast L^2 + 1 \geq 1$ and $b_{n-1}$ will be one. If $a' < 2^{n-1}$, then $(a - 2^{n-1}) \ast L^2 + 1 \leq (-1 + \epsilon) \ast L^2 + 1 \leq (-1 + 1 - \frac{1}{2^L})L^2 + 1 \leq 0$ and hence $b_{n-1}$ will be zero. In either case $x = a - b_{n-1}2^{n-1}$ will satisfy the hypothesis, and we can apply the same argument for bit $b_{n-2}$. □

Given a well-positioned point $p \in Kq$, we can extract bit representations of each of the coordinates of $q$ due to Lemma 3.2, and hence of all the vertices of $Kq$. Next task is to obtain each of their incremental vectors by simulating circuit $B$ in Linear-FIXP. Circuit $B$ is a Boolean circuit with operations $\wedge, \lor$ and $\neg$ and takes only Boolean input. These operations are easy to simulate in Linear-FIXP: If $a, b \in \{0, 1\}$, then clearly $a \wedge b = \min(a, b), a \lor b = \max(a, b)$ and $\neg a = (1 - a)$.

Thus, if $p$ is well-positioned, then incremental vectors of the vertices of $Kq$ can be computed using a Linear-FIXP circuit. However, if $p$ is poorly-positioned, then Lemma 3.2 provides no guarantee and indeed the $\text{ExtractBits}$ procedure may produce
vector $b$ where each $b_i$ may take any value in $[0, 1]$. This is expected due to continuity property of Linear-FIXP operations. Similar difficulty arises in the approaches of Daskalakis et al. [19] and Chen et al. [15]. Both resort to a sampling argument, first proposed in [19], and later improved in [15]. Next we describe a version of [15] argument.

Given a set of points $S = \{p_i : 1 \leq i \leq l\}$, let $I_w(S)$ and $I_p(S)$ denote the set of indices $i$ of the well and poorly positioned points of $S$ respectively. For $p \in \mathbb{R}^n_+$, let $\pi(p) = \{q : q_1, q_2$ are the largest integers from $\{0, \ldots, 2^n - 1\}$ s.t. $q_1 \leq p_1$ and $q_2 \leq p_2\}$. For $e^1 = (1, 0), e^2 = (0, 1)$ and $e^0 = (-1, -1)$, let $\zeta(p) = e^i$, where $i = g_S(\pi(p))$.

**Lemma 3.3.** Given $p \in [0, 2^n - 1]^2$, consider the set $S = \{p^1, \ldots, p^{16}\}$ such that

$$p^j = p + (j - 1)\left(\frac{1}{L}, \frac{1}{L}\right), \quad j \in [16]$$

For each $j \in I_p(S)$, let $r^j \in \mathbb{R}^2$ be a vector with $\|r^j\|_\infty \leq 1$. And for each $j \in I_w(S)$, let $r^j = \zeta(p^j)$. If $\|\sum_{j=1}^{16} r^j\|_\infty = 0$ then $K_\pi(p)$ is trichromatic.

**Proof.** Let $Q = \{q^j = \pi(p^j) : p^j \in S\}$. Since $\frac{1}{L} << 1$ the set crosses boundaries of cells at most twice. In other words, for each $i = 1, 2$, there is at most one $j_i$ such that $q^j_i = q^{j_i - 1}_i + 1$. Therefore, set $Q$ can have at most three elements, and they are part of the same square which has to be $K_\pi(p)$.

Further, since $\frac{1}{2^s} << \frac{1}{L} << 1$, there can be at most two poorly-positioned points in $S$. So, we have $|I_w(S)| \geq 14$. Let $r^G = \sum_{j \in I_w(S)} r^j$, then we have $\|r^G + \sum_{j \in I_p(S)} r^j\|_\infty = 0 \Rightarrow \|r^G\|_\infty \leq \|\sum_{j \in I_p(S)} r^j\|_\infty \leq 2$, because $|I_p(S)| \leq 2$ and $\|r^j\|_\infty \leq 1$ for each $j \in I_p(S)$.

Let $W_i$ be the number of indices of $I_w(S)$ with $r^j = e^i$. Using the above fact, we will show that $W_i > 0, i = 0, 1, 2$, to prove the lemma.

If $W_0 = 0$ then $W_i \geq 7$ for either $i = 1$ or $i = 2$. In that case, $r^G_i \geq 7$, a contradiction. If $W_i = 0$ for $t = 1$ or 2, then $W_0 < 3$ or else $r^G_t \geq 3$. Let $i^* = \arg \max_{0 \leq i \leq 2} W_i$, then clearly $W_t, \geq 7$ and $i^* \neq 0$. Then, $r^G_{i^*} \geq 7 - 2 = 5$, again a contradiction. □

**Remark 3.4.** Note that, in Lemma 3.3, it suffices to assume $\|\sum_{j=1}^{16} r^j\|_\infty < 1$ for $p$ to be in a trichromatic square. We use this fact to derive inapproximability results in Section 6.

Lemma 3.3 implies that even if point $p$ is poorly-positioned, we can make sure that it forms a fixed-point only when it is in a trichromatic square by sampling 16 carefully chosen points near it. A complete construction of a $2D$-Linear-FIXP circuit $C$ from a given $2D$-Brouwer function defined by $B$ is given in Table 2. We then show its correctness using Lemmas 3.2 and 3.3.

The number of gates used in steps $S_1$, $S_4$, $S_5$ and $S_6$ of Table 2 is a constant. We use $O(n)$ gates in step $S_2$, and 16 times as many as the number of gates in $B$ in step $S_3$. Further, since value of $L$ is polynomial in $\text{size}[B]$, the constants used in steps $S_1$, $S_2$ and $S_5$ are polynomial sized. Thus, the total size of the resulting $2D$-Linear-FIXP circuit $C$ constructed by the above procedure is polynomial in $\text{size}[B]$. Next we show that each of the fixed-points of function $F$ represented by circuit $C$ are in trichromatic squares of the grid $G_n$.

**Lemma 3.5.** Every fixed-point of $F$ is inside a trichromatic square of $G_n$.

**Proof.** Let $p \in [0, 2^n - 1]^2$ be a fixed-point of $F$. If $p \in (0, 2^n - 1)^2$ then for it to be a fixed-point, the final incremental vector $r$ has to be $(0, 0)$. Let $S = \{p^j : j \in [16]\}$. Due to Lemma 3.2, we know that for each $j \in I_w(S)$ we have $r^j = \zeta(p^j)$. Further,
Table 2
Construction of a 2D-Linear-FIXP circuit

<table>
<thead>
<tr>
<th>S.</th>
<th>Statement</th>
</tr>
</thead>
</table>
| S1. | Let \( p = (p_1, p_2) \) denote the input of the Linear-FIXP circuit. These are any real number from \([0, 2^n - 1]\). Compute 16 points using the + gates and rational constants:  
\[
p' = p + (j - 1)\left(\frac{1}{2}, \frac{1}{2}\right), \quad j \in [16]
\] |
| S2. | Call ExtractBits\((p'_1, t = 1, 2)\) and \(j \in [16]\), and let the output vector be \(b^{j,t}\). |
| S3. | For each \( j \leq [16] \), feed \( b_{i,1}^{j,1}, \ldots, b_{i,1}^{j,2}, b_{i,2}^{j,2}, \ldots b_{i,2}^{j,2} \) to a simulation of circuit \( B \), where \( \lor, \land \), and \( x \) are replaced with \( \max, \min \), and \( 1 - x \) respectively. Note that, there are total of 16 simulations of circuit \( B \). Let \( \Delta^+_{1,1}, \Delta^-_{1,1}, \Delta^+_{2,1}, \Delta^-_{2,1} \) be the output values of these. |
| S4. | For each \( j \in [16] \), compute \( r_1^j = \min\{\max\{\Delta^+_{1,1} - \Delta^-_{1,1}, -1\}, 1\} \) and \( r_2^j = \min\{\max\{\Delta^+_{2,1} - \Delta^-_{2,1}, -1\}, 1\} \). |
| S5. | Compute \( r_1 = \frac{1}{16} \sum_{j \in [16]} r_1^j \) and \( r_2 = \frac{1}{16} \sum_{j \in [16]} r_2^j \). |
| S6. | Output \( p'_1 = \max\{\min\{p_1 + r_1, 2^n - 1\}, 0\} \) and \( p'_2 = \max\{\min\{p_2 + r_2, 2^n - 1\}, 0\} \). |

Due to step \((S_4)\) for each \( k \in I_p(S) \), \( \|r^j\|_\infty \leq 1 \). Therefore, using the fact that \( r = \frac{1}{16} \sum_{j = 1}^{16} r^j \) and Lemma 3.3 it follows that \( K_\pi(p) \) is trichromatic.

For the remaining case, \( p \) has to be on a boundary of the grid. Since \( B \) is a valid circuit, vertices on the boundary have specific incremental vectors: Let \( q \) be such a vertex then if \( q_2 = 0 \) then \( \zeta(q) = e^2 = (0, 1) \), else if \( q_1 = 0 \) then \( \zeta(q) = e^3 = (1, 0) \), otherwise \( \zeta(q) = e^0 = (-1, -1) \). Using this fact, and that \( |I_w(S)| \geq 14 \) (Lemma 3.3), next we show \( p \) can not be a fixed-point in that case.

If \( p_2 = 0 \), then for each \( k \in I_w(S) \), \( r^j = (0, 1) \). Therefore, we have \( r_2 > 0 \) and in turn \( p_2' > p_2 \). If \( p_2 > 0 \) and \( p_1 = 2^n - 1 \), then for each \( k \in I_w(S) \), \( r^j \) is either \((0, 1)\) or \((-1, -1)\), and one of them occurs at least 7 times. Therefore, either \( r_2 > 0 \) and in turn \( p_2' > p_2 \), or \( r_1 < 0 \) and in turn \( p_1' < p_1 \).

If \( 0 < p_1 < 2^n - 1 \) and \( p_2 = 2^n - 1 \), then for each \( j \in I_w(S) \), \( r^j \) is either \((1, 0)\) or \((-1, -1)\). Therefore, we have either \( r_1 > 0 \) and in turn \( p_1' > p_1 \), or \( r_2 < 0 \) and in turn \( p_2' < p_2 \). If \( p_1 = 0 \) and \( 1 < p_2 < 2^n - 1 \), then for each \( j \in I_w(S) \), \( r_1^j = 1, r_2^j = 0 \). Therefore, we have \( r_1 > 0 \) and in turn \( p_1' > p_1 \). Further, if \( p_1 = 0 \) and \( 0 < p_2 < 1 \), then by similar argument either \( p_1' > p_1 \) or \( p_2' > p_2 \). □

Remark 3.6. Note that every fixed-point of \( F \) is in a trichromatic square whose vertices with the three colors form a right-angled triangle with north-east oriented hypotenuse.

It is easy to shrink the range of \( F \) from \([0, 2^n - 1]\) to \([0, 1]\). Consider a function \( F' : [0, 1]^2 \to [0, 1]^2 \), such that \( F'(\lambda_1, \lambda_2) = \frac{1}{2^n - 1} F((2^n - 1)\lambda_1, (2^n - 1)\lambda_2) \), then clearly, \((\lambda_1, \lambda_2)\) is a fixed-point of \( F' \) if and only if \(((2^n - 1)\lambda_1, (2^n - 1)\lambda_2)\) is a fixed-point of \( F \). Thus we get the following theorem using Lemma 3.5 and the fact that
size[C] = poly(size[B]).

Theorem 3.7. The class of $kD$-Linear-FIXP with $k > 1$ is PPAD-hard.

Remark 3.8. All the known PPAD-hardness proofs for games go through generalized circuits \([19, 15]\), which allows feedback-loops and approximate computation for each operation. However, in \([19]\) and \([15]\) feedback-loops are used only to connect the "output nodes" to the "input nodes" to ensure that their values are almost same (approximate solution). Further, each exact operation of generalized circuit may be simulated using \(\{\max, +, \ast \zeta\}\), and polynomial sized rational numbers. The reduction discussed in this section may be obtained using these observations as well from the previous approaches of reducing 2D or 3D-Brouwer to generalized circuit.

4. Reduction: Linear-FIXP to 2-Nash. Before we show hardness for low rank games, to convey intuition in this section we describe a reduction from Linear-FIXP to arbitrary bimatrix game there by giving an alternative proof of PPAD-hardness for the latter using Theorem 3.7. Some of the techniques we develop here will be useful to show PPAD-hardness for low rank games.

Let $C$ be a $kD$-Linear-FIXP circuit representing function $F : [0, 1]^k \rightarrow [0, 1]^k$. Recall that a Linear-FIXP circuit allows only three operations, namely max, + and $\ast \zeta$ where $\zeta$ is a rational number, and it forms a DAG. The size of $C$ is $\#$ inputs $+ \#$ gates $+ \#$ total bit lengths of the constants in the circuit.

If $C$ is considered as a function from $\mathbb{R}^k$ to $\mathbb{R}^k$, then it is same as function $F$ on $[0, 1]^k$, but can be anything outside this range and hence may have fixed-points outside $[0, 1]^k$ as well. To prevent this, we add two max gates for every output of the circuit, as follows: Let $\tau_1, \ldots, \tau_k$ be the $k$ outputs of circuit $C$. Without loss of generality (wlog), we will add two max gates for each $l \in [k]$ to ensure that each output value is in $[0, 1]$.

\begin{equation}
\max\{0, \min\{1, \tau_l\}\} = \max\{0, -1 \ast \max\{-1, -1 \ast \tau_l\}\}
\end{equation}

The above transformation ensures that the output vector of $C$ is always in $[0, 1]^k$, and hence fixed-points of $C$ are exactly the fixed-points of $F$. Next, we show that it is wlog to assume that one of the inputs of every max gate is zero.

Lemma 4.1. Given a circuit $C$, it can be transformed to an equivalent polynomial sized circuit where one of the inputs of every max gate is zero.

Proof. Consider a max gate, and let $a$ and $b$ be the inputs and $c$ be the output, then we have $c = \max\{a, b\}$ which is equivalent to $c = \max\{0, b-a\} + a$. Therefore, we can transform circuit $C$ such that one input of every max gate is $0$. This transformation requires $3$ extra gates per max gate, two $+$ and one $\ast \zeta$ where $\zeta = -1$. Clearly, the increase in the size of the circuit is polynomial.

Wlog we assume that every max gate of circuit $C$ has exactly one non-trivial input, and the other input is always zero (due to Lemma 4.1). Let $m$ be the number of max gates in $C$. Since $C$ is a DAG, there is an ordering among the max gates, say $g_1, \ldots, g_m$, such that if there is a path from $g_i$ to $g_j$ in $C$ then $i < j$; ties are broken arbitrarily. Let $n = m - 2k$ be the number of max gate in the original circuit, before the addition of (4.1) per output. Let the ordering be such that these are the first $n$ gates $g_1, \ldots, g_n$. In (4.1), let $g_{n+2l-1}$ denote the inner max gate and $g_{n+2l}$ denote the outer one, then $k$ outputs of circuit are the outputs of gates $g_{n+2l}$, $l \in [k]$.

Let the $k$ inputs of circuit $C$ be denoted by $\lambda = (\lambda_1, \ldots, \lambda_k)$, and let $x_i$ capture the output of the $i^{th}$ max gate. Note that, $x_{n+2l}$, $\forall l \in [k]$ constitute output of the circuit. Except for the max, rest of the two operations give rise to linear expressions.
in the $\lambda$ and $x_i$s of the previous max gates. We use this observation crucially in the rest of the construction.

Note that, for each $i \in [m]$, the non-trivial input of $g_i$ is a linear expression in $x_1, \ldots, x_{i-1}, \lambda_1, \ldots, \lambda_k$, with a constant term. We denote this expression by $L_i(x_1, \ldots, x_{i-1}, \lambda)$, then the following conditions exactly capture the operation of $g_i$.

\begin{equation}
\forall i \in [m], \quad x_i \geq 0, \quad x_i \geq L_i(x_1, \ldots, x_{i-1}, \lambda) \tag{4.2}
\end{equation}

\begin{equation}
\forall i \in [m], \quad x_i(x_i - L_i(x_1, \ldots, x_{i-1}, \lambda)) = 0 \tag{4.3}
\end{equation}

The next lemma follows by construction.

**Lemma 4.2.** Given $\lambda \in \mathbb{R}^k$, $(x, \lambda)$ satisfies (4.2) and (4.3) iff when $\lambda$ is given as the input to circuit $C$, the $i^{th}$ max gate evaluates to $x_i$ for all $i \in [m]$.

**Proof.** Reverse direction follows just by construction. For the forward direction we will argue by induction. Suppose, $(x, \lambda)$ satisfies (4.2) and (4.3). Then, for $\lambda$ as input to $C$, clearly $L_1$ evaluates to exactly the input of the (first max) gate $g_1$. In that case, (4.2) forces that $x_1$ is at least as large as inputs of $g_1$, and (4.3) forces that it equals one of the input. Thus, $x_1$ captures output of $g_1$. Now, suppose this is true for first $k \geq 1$ max gates. Then for $(k+1)^{th}$ max gate, again $L_{k+1}$ is exactly the input of $g_{k+1}$, and the lemma follows by the same argument. \[\square\]

Constraints of (4.2) gives a system of linear inequalities,

\begin{equation}
Ax \geq \sum_{l=1}^{k} \lambda_l u_l^l + b, \quad x \geq 0 \tag{4.4}
\end{equation}

where, $b$ and $u_l^l$, $l \in [k]$ are $m$-dimensional rational vectors, and $A$ is an $m \times m$ lower-triangular rational matrix with ones on the diagonal. Once we plugin some values for $\lambda_1, \ldots, \lambda_k$, (4.4) becomes a polyhedron in $x$. Let it be denoted by $P(\lambda)$. For any $\lambda \in \mathbb{R}^k$ and $x \in P(\lambda)$, vector $(x, \lambda)$ satisfies (4.2). If it also satisfies (4.3) then using Lemma 4.2 it follows that $x_i$ captures output of $i^{th}$ max game of circuit $C$ when it is evaluated at $\lambda$, and in particular $x_{n+2l}$, $l \in [k]$ captures $l^{th}$ output. Therefore, at a fixed-point of $C$ we also have $\lambda_l = x_{n+2l}, \forall l \in [k]$. Using this fact next we replace each $\lambda_l$ with $x_{n+2l}$, in order to construct a linear complementarity problem formulation (LCP) that exactly captures the fixed-points. LCP is a generalization of complementarity slackness conditions of linear programming, where given a square matrix $M$ and a vector $q$ the problem is to find a vector $y$ such that $My \leq q$, $y \geq 0$ and $(Mx_i - q_i)y_i = 0, \forall i$.

Let $v_i^\top \in \mathbb{R}^m$ be a unit vector with 1 on $(n+2l)^{th}$ co-ordinate and zeros otherwise, i.e., $v_i^\top x = x_{n+2l}$. Let $A' = A - \sum_{l \in [k]} u_l^l v_i^\top$, and consider the following LCP.

\begin{equation}
\forall i \in [m], \quad x_i \geq 0; \quad A'x \geq b \tag{4.5}
\end{equation}

**Lemma 4.3.** Vector $x \in \mathbb{R}^m$ is a solution of LCP 4.5 iff $\lambda$, where $\lambda_l = x_{n+2l}, \forall l \in [k]$, is a fixed-point of the Linear-FIXP function $F$.

**Proof.** Since $\forall l \in [k], \lambda_l = x_{n+2l} = v_i^\top x$ the forward direction follows, because $(x, \lambda)$ satisfies both (4.2) and (4.3) by construction (Lemma 4.2). For the reverse
direction let $\lambda$ be a fixed-point of $F$ and $x$ be a vector such that $x_i$ is the output value of $i^{th}$ max gate when $\lambda$ is the input to the circuit $C$. Clearly $(x, \lambda)$ satisfies (4.2) and (4.3) (Lemma 4.2). The lemma follows using the fact that $\lambda_l = x_{n+2l} = v^T_\lambda x$, $\forall l \in [k]$ because $\lambda$ is a fixed-point. 

Adler and Verma [4] showed that solving an LCP can be reduced to finding a symmetric NE of a symmetric two-player game, if the LCP matrix is strictly semi-monotone. Next we show that the matrix of LCP 4.5 is strictly semi-monotone, and then PPAD-hardness for symmetric NE will follow using the result of [4]. However, for the sake of completeness the direct reduction from LCP 4.5 to a symmetric game is given in Appendix A; matrix of the resulting game is as follows:

$$
\begin{bmatrix}
-A' & b + 1 \\
0^T & 1
\end{bmatrix}
$$

To show semi-monotonicity of $A'$ we need to understand (4.2) for the last $2k$ max gates that we added in (4.1) to ensure that the outcome of the circuit is in $[0,1]^k$. Due to Lemma 4.1 wlog we have assumed that one of the inputs of every max gate of the circuit is zero. For this to be the case, (4.1) has to be transformed as follows,

$$
\forall l \in [k], \max\{0, -1 \ast (\max\{0, -\tau_l + 1\} - 1)\}
$$

Here, $\forall l \in [k], \tau_l$ is a linear expression in $x_1, \ldots, x_n, \lambda$, and in turn so is $L_{n+2l-1} = 1 - \tau_l$. Recall that $x_{n+2l-1}$ captures the output of the inner max gate and $x_{n+2l}$ captures the output of the outer max gate. Therefore, we have

\begin{align*}
\forall l \in [k], & x_{n+2l-1} \geq 0, \quad x_{n+2l-1} \geq L_{n+2l-1}(x_1, \ldots, x_n, \lambda) \\
\forall l \in [k], & x_{n+2l} \geq 0, \quad x_{n+2l} \geq 1 - x_{n+2l-1} \Rightarrow x_{n+2l-1} + x_{n+2l} \geq 1
\end{align*}

The following property is easy to obtain using (4.6); we will obtain more such properties of $A, b$ and $u$’s using (4.6) in the next section as and when needed.

$P_1$. $\forall l \in [k], (Ax)_{n+2l} = x_{n+2l-1} + x_{n+2l}, b_{n+2l} = 1$, and $u''_{n+2l} = 0, \forall l \in [k]$.

We show the next lemma using Property $P_1$ and the fact that $A$ is lower-triangular.

**Lemma 4.4.** Matrix $A'$ is strictly semi-monotone, i.e., given an $x \geq 0, x \neq 0, \exists j$ such that $x_j > 0$ and $(A'x)_j > 0$.

**Proof.** If $x_{n+2l} > 0$ for any $l \in [k]$, then we are done using Property $(P_1)$. Suppose, $x_{n+2l} = 0, \forall l \in [k]$ then we have $A'x = Ax$. In that case for $j = \min\{i \mid x_i > 0\}$ we have $x_j > 0$ and $(A'x)_j = (Ax)_j > 0$ because $A$ is lower-triangular. 

For hardness to follow, the size of reduced game should be polynomial in the size$|C|$. Since the game constructed in [4] is linear in the size of $A'$ and $b$, it suffice to show that these are of polynomial size. We show this in the next section in Lemma 5.4. The next theorem follows using Theorem 3.7 and Lemmas 5.4, 4.3 and 4.4 together with the reduction of [4] from LCP with strictly semi-monotone matrices to symmetric bimatrix games.

**Theorem 4.5.** The problem of computing a symmetric Nash equilibrium of a symmetric bimatrix game is PPAD-hard.

As discussed in Section 5.2, [22] showed that solving simple stochastic games [16] reduces to finding a unique fixed-point of a Linear-FIXP problem. Further, reduction from Liner-FIXP problem to LCP (Lemma 4.3), and LCP to symmetric game [4] (also
Lemma A.1) are solutions preserving. Using this together with Theorem 4.5 we get the next corollary.

**Corollary 4.6.** Computing a unique symmetric NE of a symmetric game is as hard as solving a simple stochastic game.

McLannen and Tourky [30] showed that the symmetric Nash equilibria of a symmetric game \((S, S^T)\) are in one-to-one correspondence with the Nash equilibrium strategies of the second player of game \((S, I)\), where \(I\) is an identity matrix. In that case, NE strategies of the first player should form a convex set because they are essentially solutions of a feasibility lp (follows Lemma 2.2). Using this together with the result of [22], Theorem 4.5 and Corollary 4.6, we get the following.

**Theorem 4.7.** The problem of computing a Nash equilibrium of a bimatrix game is PPAD-hard. Even if the set of NE is convex, the problem remains at least as hard as solving a simple stochastic game.

Chen et. al. [15] showed PPAD-hardness for NE computation in bimatrix games (2-Nash), which also implies that symmetric NE computation in symmetric bimatrix game is PPAD-hard (symmetric 2-Nash) as the former reduces to the latter (discussed in Section 2). Theorem 4.7 gives an alternative proof of these facts. The Chen et. al. reduction goes through generalized circuit (similar to Linear-FIXP circuit) with fuzzy gates, graphical games, and game gadgets to simulate each gate of the generalized circuit separately. Our reduction bypasses all of these completely, and provides a simpler reduction using the connections between LCPs and bimatrix games.

Building on the constructs of this section, in the next section we reduce \(kD\)-Linear-FIXP to rank-\((k+1)\) game, and thereby obtain PPAD-hardness for computing NE in a game with rank \(\geq 3\) using the fact that \(2D\)-Linear-FIXP is PPAD-hard (Theorem 3.7).

5. **Reduction: \(kD\)-Linear-FIXP to Rank-\((k+1)\) Game.** Given a \(kD\)-Linear-FIXP instance, with circuit \(C\) representing function \(F: [0, 1]^k \rightarrow [0, 1]^k\), in this section we construct a rank-\((k+1)\) bimatrix game whose Nash equilibria are almost\(^7\) in one-to-one correspondence with the fixed-points of \(F\). We do this in two steps. First we replace the circuit \(C\) by a parametric linear program (LP) with \(k\)-parameters, where inputs of circuit \(C\) become parameters of the LP. Given values of the \(k\) inputs, we show that the \(k\) outputs of the circuit \(C\) are linear function of a solution of the LP. This defines a function \(F_{lp}\) from \(\mathbb{R}^k\) to \(\mathbb{R}^k\), and we show that the fixed-points of \(F_{lp}\) are in one to one correspondence with the fixed-points of \(F\). Later, we construct a rank-\((k+1)\) game using the LP and its dual, such that Nash equilibria of the resulting game exactly capture the fixed-points of \(F_{lp}\).

**Remark 5.1.** Recall that linear programs are equivalent to zero-sum games [18, 3]. However, the reductions from LP to zero-sum games constructs a symmetric game, and require to compute a symmetric Nash equilibrium. There are no such restrictions in our construction, however our reduction is not general enough and uses the fact that the parametric LP has been constructed from a Linear-FIXP circuit. It will be interesting to reduce an LP to a non-symmetric zero-sum game, and also a fixed-point problem with parametric LP to a constant rank game in general.

5.1. **Replacing Linear-FIXP circuit with a linear program.** Given circuit \(C\) of a \(kD\)-Linear-FIXP instance in Section 4 we constructed equivalent system (4.2) and (4.3) (Lemma 4.2). Recall that here \(x_i\) is the output of \(i^{th}\) max game and

---

\(^7\)Essentially, Nash equilibrium strategies of the first player are in one-to-one correspondence with the fixed-points
(l₁, . . . , lₖ) represent the inputs of circuit C. Considering \( \lambda \) as parameters, linear inequalities of (4.2) defines polyhedron \( \mathcal{P}(\lambda) \) succinctly represented in (4.4), where \( A \) is an \( m \times m \) lower-triangular matrix, and \( b \) and \( u' \) are \( m \)-dimensional vectors.

Given any \( \lambda \in [0, 1]^k \), by construction we know that for any \( x \in \mathcal{P}(\lambda) \), \((x, \lambda)\) satisfies (4.2). Next, we construct a cost vector \( c \in \mathbb{R}^m \), such that minimizing \( x^T \cdot c \) over \( \mathcal{P}(\lambda) \) will give an \( x \) such that, it together with \( \lambda \), satisfies (4.3) as well.

**ConstructCost(A)**

- \( c_m \leftarrow 1 \), \( \beta_m \leftarrow 1 \)
- for \( i = m - 1 \) to 1 do
- \( c_i \leftarrow \sum_{j > i} |a_{ji}| \beta_j + 1 \), \( \beta_i \leftarrow c_i + \sum_{j > i} |a_{ji}| \beta_j \)
- end for

Output \( c \)

For \( c = \text{ConstructCost}(A) \), consider the following parameterized LP and its dual.

**LP(\( \lambda \)):**

\[
\begin{align*}
\text{min} & : \quad c^T \cdot x \\
\text{s.t.,} & : \quad Ax \geq \sum_{l \in [k]} \lambda_l u'_l + b \\
& : \quad x \geq 0
\end{align*}
\]

**DLP(\( \lambda \)):**

\[
\begin{align*}
\text{max} & : \quad (\sum_{l \in [k]} \lambda_l u'_l + b)^T \cdot y \\
\text{s.t.,} & : \quad A^T y \leq c \\
& : \quad y \geq 0
\end{align*}
\]

The complementary slackness requires that solutions of \( LP(\lambda) \) and \( DLP(\lambda) \) satisfy (KKT conditions),

\[
\forall i \in [m], \ y_i(Ax - \sum_{l \in [k]} \lambda_l u'_l - b)_i = 0, \quad x_i(A^T y - c)_i = 0
\]

**Lemma 5.2.** Given \( \lambda \in \mathbb{R}^k \), \( x \) is a solution of \( LP(\lambda) \) iff \((x, \lambda)\) satisfies (4.2) and (4.3).

Proof. \((\Rightarrow)\) Let \( y \) be the dual solution corresponding to \( x \), i.e., \((x, y)\) satisfies (5.2). Since \( x \) is a feasible point of \( LP(\lambda) \), clearly, \((x, \lambda)\) satisfies (4.2). For (4.3), it suffices to show that \( \forall i \in [m] \), \( x_i > 0 \Rightarrow y_i > 0 \), then the proof follows using (5.2).

Let \( \beta \in \mathbb{R}^m \) be the vector calculated in \( \text{ConstructCost}(A) \) of Table 3. We do the proof by induction, where we show that \( \forall i \in [m] \), \( y_i \leq \beta_i \), and \( x_i > 0 \Rightarrow y_i > 0 \). Recall that \( A \) is lower-triangular with ones on the diagonal. Therefore, \( A^T \) is upper-triangular with ones on the diagonal.

Our base case is when \( i = m \): If \( x_m > 0 \), then due to (5.2) we have \( y_m = (A^T y)_m = c_m = 1 > 0 \). Further, \((A^T y)_m \leq c_m \Rightarrow y_m \leq 1 \). Since \( \beta_m = 1 \) we get \( y_m \leq \beta_m \).

Now, let the hypothesis be true for \( j > r \). For \( r \) if \( x_r > 0 \) then \((A^T y)_r = c_r \Rightarrow (A^T y)_r = y_r + \sum_{j > r} a_{jr} y_j = c_r \) (due to (5.2)). Since \( \forall j > r, 0 \leq y_j \leq \beta_j \) and \( c_r = \sum_{j > r} |a_{jr}| \beta_j + 1 \), we have \( \sum_{j > r} a_{jr} y_j < c_r \). Therefore, for the equality to hold we must have \( y_r > 0 \). Further, \((A^T y)_r = y_r + \sum_{j > r} a_{jr} y_j \leq c_r \Rightarrow y_r \leq c_r - \sum_{j > r} a_{jr} y_j \leq c_r + \sum_{j > r} |a_{jr}| y_j \leq c_r + \sum_{j > r} |a_{jr}| \beta_j = \beta_r \).

\((\Leftarrow)\) If \((x, \lambda)\) satisfies (4.2) and (4.3) then clearly \( x \) is feasible in \( LP(\lambda) \). Construct \( y \), from \( y_m \) to \( y_1 \) as follows: if \( x_r = 0 \) then set \( y_r = 0 \), else set \( y_r = c_r - \sum_{j > r} a_{jr} y_j \).
It is easy to see that \( y \) is feasible in \( DLP(\lambda) \), and it, together with \( (x, \lambda) \) satisfies (5.2). □

Lemmas 4.2 and 5.2 imply that \( LP(\lambda) \) simulates the circuit. Next, we show that the circuit can be replaced by \( LP(\lambda) \) without effecting the fixed-points of \( F \). Consider function \( F^{lp} : \mathbb{R}^k \rightarrow [0, 1]^k \), such that,

\[
\lambda \in \mathbb{R}^k, \quad F^{lp}(\lambda) = (x_{n+2l})_{l \in [k]}, \quad \text{where} \ x = LP(\lambda)
\]

We show that function \( F^{lp} \) is well-defined, and its fixed-points are exactly the fixed-points of \( F \).

**Lemma 5.3.** \( F^{lp} \) is well defined. Further, \( \lambda \in \mathbb{R}^k \) is a fixed-point of \( F^{lp} \) iff it is a fixed-point of \( F \).

**Proof.** For any given \( \lambda \in \mathbb{R}^k \), using Lemmas 4.2 and 5.2, it follows that \( LP(\lambda) \) has a unique solution, and if it is \( x \) then \( 0 \leq x_{n+2l} \leq 1, \, \forall l \in [k] \). Thus, \( F^{lp} \) is well defined.

For the second part, it suffices to show that \( F^{lp}(\lambda) = C(\lambda), \, \forall \lambda \in \mathbb{R}^k \), as function \( F \) is represented by circuit \( C \). In other words, when vector \( \lambda \) is the input to circuit \( C \), then \( i^{th} \) max gate evaluates to \( x_i, \, \forall i \in [m] \), where \( x = LP(\lambda) \). This follows using Lemmas 4.2 and 5.2. □

Finally, for a reduction to give hardness we need to show that it is a polynomial-time procedure. Our process of reducing \( C \) to \( LP(\lambda) \), or to a bimatrix game in Section 4 involves computation of \( A, b, c \) and \( u \)'s. If sizes of these are polynomial in \( size[C] \), then the reduction is valid.

**Lemma 5.4.** Size of matrix \( A \), and vectors \( c, b \) and \( u \), \( \forall l \in [k] \) are polynomial in \( size[C] \).

**Proof.** By construction \( A, b \) and \( u \), \( \forall l \in [k] \), are formed by the coefficients of the linear expressions \( L_i \)'s of (4.2). These linear expressions are constructed due to the + and *\( \zeta \) gates of the circuit \( C \), therefore, the absolute value of any of its coefficient is at most \( \zeta_{max} \), where \( v \) is the number of *\( \zeta \) gates in \( C \), and \( \zeta_{max} \) is the maximum absolute rational constant used in \( C \). For rational constants \( \zeta_1, \zeta_2 \), since \( size(\zeta_1 * \zeta_2) = size(\zeta_1) + size(\zeta_2) \), we have that the size of every co-efficient of \( L_i \) is at most \( size[C] \). Thus, sizes of \( A \), \( b \) and \( u \), \( \forall l \in [k] \) are at most polynomial in \( size[C] \).

Let \( A_{max} = \max_{i \in [m]} A_{ij} \), then by construction \( c_1 = \max_{j \in [m]} c_j \leq (2A_{max} + 1)^n \) (see Table 3). Therefore, the size of \( c \) is also bounded by a polynomial in \( size[C] \). □

From Lemmas 5.3 and 5.4 we can conclude that finding a fixed-point of \( F \) is equivalent to finding one for function \( F^{lp} \), which can be represented using polynomially many bits in the \( size[C] \). Next we reduce the fixed-point computation for \( F^{lp} \) to Nash equilibrium computation for a rank-(\( k+1 \)) game, such that the size of resulting game is polynomial in the size of parameters of function \( F^{lp} \).

### 5.2. Constructing Rank-(\( k+1 \)) Game.

Since feasibility and complementary slackness are necessary and sufficient conditions for the solutions of an LP, it is well known that an LP can be formulated as a linear complementarity problem (LCP). Using this, next we construct an LCP whose solutions are exactly the fixed points of \( F^{lp} \). Before we do this, note that since all the co-ordinates of the cost vector \( c \) are strictly positive, we can make it all ones vector by dividing \( j^{th} \) column of \( A \) by \( c_j \).

Let \( H \) be the transformed matrix, i.e.,

\[
H_{ij} = A_{ij}/c_j
\]
Before we connect the solutions of LCP\((\lambda)\) in the complementary slackness conditions of the LP\((\lambda)\) to get an LCP (as done in Section 4 to get LCP \((\ref{eq:4.1})\)). For \(l \in [k]\), let \(v^l\) be an \(m\)-dimensional vector with \((n + 2l)^{th}\) co-ordinate set to \(1/c_{n+2l}\) and rest all set to zero, then \(v^T \cdot x' = x_{n+2l}/c_{n+2l}\). Lemma 5.5 implies that, at a fixed-point of \(F^{lp}\) we have \(\lambda_l = x_{n+2l} = x_{n+2l}/c_{n+2l} = v^T \cdot x', \ \forall l \in [k]\), where \(x = \text{LP}(\lambda)\) and \(x' = \text{LP}(\lambda)\). Using this as a motivation, we replace \(\lambda_l\) with \((v^T \cdot x)\) in the constraints of \(\text{LP}(\lambda)\). The resulting matrix will be

\[
H' = H - \sum_{l=1}^{k} u^l \cdot v^T, \ \forall(i, j)
\]

Using the above observation, and feasibility and complementary slackness conditions for \((\ref{eq:5.4})\), we construct the following LCP, called LCP\(_C\),

\[
\begin{align*}
\text{LP}^C(\lambda) : & \quad \min : \sum_{i} x_i \\
& \text{s.t.}, \quad Hx \geq \sum_{l \in [k]} \lambda_l u^l + b \\
& \quad x \geq 0
\end{align*}
\]

\[
\begin{align*}
\text{DLP}^C(\lambda) : & \quad \max : (\sum_{l \in [k]} \lambda_l u^l + b)^T \cdot y \\
& \text{s.t.}, \quad H^T y \leq 1 \\
& \quad y \geq 0
\end{align*}
\]

The next lemma follows by construction.

**Lemma 5.5.** Given \(\lambda \in \mathbb{R}^k\), \(x\) and \(y\) are solutions of \(\text{LP}(\lambda)\) and \(\text{DLP}(\lambda)\) respectively iff, \(x', y'\), where \(x'_j = x_j c_j\), \(\forall j \in [m]\), and \(y\) are solutions of \(\text{LP}^C(\lambda)\) and \(\text{DLP}^C(\lambda)\) respectively.

Using the fact that at a fixed-point input equals output, next we replace \(C\) for \((\ref{eq:5.4})\), we construct the following LCP, called LCP\(_P\),

\[
\begin{align*}
\text{LP}^P(\lambda) : & \quad \min : \sum_{i} x_i \\
& \text{s.t.}, \quad Hx \geq \sum_{l \in [k]} \lambda_l u^l + b \\
& \quad x \geq 0
\end{align*}
\]

\[
\begin{align*}
\text{DLP}^P(\lambda) : & \quad \max : (\sum_{l \in [k]} \lambda_l u^l + b)^T \cdot y \\
& \text{s.t.}, \quad H^T y \leq 1 \\
& \quad y \geq 0
\end{align*}
\]

The above properties and Property \(P_1\) (defined in Section 4) are crucial to the overall reduction.

**Lemma 5.6.** Vector \((x', y')\) is a solution of LCP\(_P\) of \((\ref{eq:5.5})\) if and only if \(\lambda \in [0, 1]^k\), where \(\lambda_l = x'_{n+2l}\), \(\forall l \in [k]\), is a fixed-point of \(F^{lp}\).

**Proof.** \((\Rightarrow)\) Let \((x', y')\) be a solution of LCP\(_P\). Then by construction of LCP\(_P\), clearly \(x'\) and \(y'\) are solutions of \(\text{LP}^P(\lambda)\) and \(\text{DLP}^P(\lambda)\) respectively, where \(\lambda_l = v^T \cdot x' = x'_{n+2l}\), \(\forall l \in [k]\) (using \(P_3\)). Set \(y = y'\), and \(x\) be such that \(x_j = x'_{j/c_j}\), then
using Lemma 5.5 we get that, $x$ and $y$ are solutions of $LP(\lambda)$ and $DLP(\lambda)$. Further, property $P_2$ ensures that $x_{n+2l} = x_{n+2l}' = \lambda_l, \forall l \in [k]$. Thus, $\lambda$ is a fixed-point of $F^{lp}$.

(\Rightarrow) Let $\lambda$ be a fixed-point of $F^{lp}$ and let $x$ and $y$ be the solutions of $LP(\lambda)$ and $DLP(\lambda)$. Let $y' = y$ and $x' = c_jx_j, \forall j \in [m]$, then using Lemma 5.5 we get that $x'$ and $y'$ are solutions of $LP'(\lambda)$ and $DLP'(\lambda)$ respectively. Using the fact that $\lambda$ is a fixed-point of $F^{lp}$ and property $P_3$, we get $v^T \cdot x' = x_{n+2l} = x_{n+2l}' = \lambda_l, \forall l \in [k]$. In that case, feasibility and complementary slackness of $LP'(\lambda)$ and $DLP'(\lambda)$, ensures that $(x', y')$ is a solution of $LCP_C$. $\square$

Next, we capture solutions of $LCP_C$ as Nash equilibria of a bimatrix game. Consider the following game:

\begin{equation}
\tilde{A} = \begin{bmatrix}
H^T & 0 \\
0^T & 1
\end{bmatrix}, \quad \tilde{B} = \begin{bmatrix}
-H^T & 0 \\
B^T + 1^T & 1
\end{bmatrix}
\end{equation}

where $1$ and $0$ are $m$-dimensional vectors of $1$s and $0$s respectively. Number of strategies of both the players is $m + 1$. Let $(\tilde{x}, s)$ and $(\tilde{y}, t)$ denote mixed-strategy vectors of the first player and the second player, then we have,

\begin{equation}
(\tilde{x}, s) \geq 0; \quad (\tilde{y}, t) \geq 0; \quad s + \sum_{i=1}^{m} \tilde{x}_i = 1; \quad t + \sum_{j=1}^{m} \tilde{y}_j = 1
\end{equation}

**Remark 5.7.** Adler and Verma [4], used this idea of adding an extra column/row to handle the r.h.s., in their reduction from 'solving' some special LCPs to symmetric game.

Semi-monotone like property of the matrix of $LCP_C$, shown in the next lemma, is important to derive equivalence between the NE of $(\tilde{A}, \tilde{B})$ and the solutions of $LCP_C$.

**Lemma 5.8.** Let $M = \begin{bmatrix} 0 & H^T \\ -H' & 0 \end{bmatrix}$ be the matrix of $LCP_C$. For any $q \in \mathbb{R}^{2m}$ with $q > 0$, the only solution of $LCP \{Mz \leq q; \ z \geq 0; \ z^T(Mz - q) = 0\}$ is $z = 0$.

Proof. It suffices to show that for any $z \geq 0, z \neq 0$, there is a $d \in [2m]$ such that $z_d > 0$ and $(Mz)_d \leq 0$. Partition $z$ as $(x, y)$. If $\forall l \in [k]$, $z_{n+2l} = x_{n+2l} = 0$, then $H'x = Hx$. Therefore, $z^T Mz = x^T H'Ty - y^T Hx = 0$. For all $d \in [m]$, if we have, $z_d > 0 \Rightarrow (Mz)_d > 0$ then $z^T Mz > 0$, a contradiction.

On the other hand, $\exists l \in [k]$ with $z_{n+2l} > 0$ and $(Mz)_{n+2l} \leq 0$ then done. Otherwise, we have $z_{n+2l} > 0$ and $(Mz)_{n+2l} > 0$. This together with Property $P_4$ gives,

\begin{align*}
(Mz)_{n+2l} &= y_{n+2l} = z_{m+n+2l} > 0 & \text{and} \\
(Mz)_{m+n+2l} &= -(H'x)_{n+2l} = -\frac{x_{n+2l}}{c_{n+2l}} - x_{n+2l} = \frac{x_{n+2l} - z_{n+2l}}{c_{n+2l}} - z_{n+2l} < 0
\end{align*}

$\square$

If $((\tilde{x}, s), (\tilde{y}, t))$ is a Nash equilibrium of game $(\tilde{A}, \tilde{B})$, the following have to be satisfied (see Lemma 2.2 for the NE characterization), where $\pi_1$ and $\pi_2$ are the scalars capturing payoffs of the first and the second player respectively.

\begin{align*}
t &\leq \pi_1; \quad & s(t - \pi_1) = 0 \\
s &\leq \pi_2; \quad & t(s - \pi_2) = 0 \\
\forall i \in [m], \ (H^T\tilde{y}_i) &\leq \pi_1; \quad & \tilde{x}_i((H^T\tilde{y}_i) - \pi_1) = 0 \\
\forall j \in [m], \ (-\tilde{x}^T H')_j + b_j s &\leq \pi_2; \quad & \tilde{y}_j((-\tilde{x}^T H')_j + b_j s + s - \pi_2) = 0
\end{align*}
Lemma 5.9. If $((\tilde{x}, s), (\tilde{y}, t))$ is a Nash equilibrium of game $(\tilde{A}, \tilde{B})$ with $s > 0$ and $t > 0$, then $(\tilde{x}, \tilde{y})$ is a solution of LCP$_C$. Further, if $(x, y)$ is a solution of LCP$_C$ then $((x^{(1)}, y^{(1)}), s)$ is a NE of game $(\tilde{A}, \tilde{B})$.

Proof. Since $s > 0$ and $t > 0$, we have $\pi_1 = t$ and $\pi_2 = s$ respectively (using (5.8)). Replacing $\pi_1$ and $\pi_2$ accordingly in the inequalities of (5.8), we get

$$\forall i \in [m], \quad (H^T \tilde{y})_i \leq t; \quad \tilde{x}_i((H^T \tilde{y})_i - t) = 0$$

$$\forall j \in [m], \quad (H' \tilde{x})_j \geq b_js; \quad \tilde{y}_j((H' \tilde{x})_j - b_js) = 0$$

Dividing the first expression of first line by $t$ and of second line by $s$, and the second expression in both lines by $s \ast t$, we get constraints of LCP. The second part is easy to verify using the formulation of LCP$_C$ (5.5) and NE conditions (5.7) and (5.8).

Lemma 5.9 shows that NE of game $(\tilde{A}, \tilde{B})$ with $s > 0, t > 0$ exactly capture the solutions of LCP$_C$. Next lemma shows that these are the only NE of this game.

Lemma 5.10. If $((\tilde{x}, s), (\tilde{y}, t))$ is a Nash equilibrium of game $(\tilde{A}, \tilde{B})$ then $s > 0$ and $t > 0$.

Proof. We will derive a contradiction for each of the three cases separately.

Case 1: $s > 0$ and $t = 0$

Then, we have $\pi_1 = t = 0$ and therefore, $H^T \tilde{y} \leq 0$. Since $H^T$ is upper-triangular with strictly positive values on the diagonal, the only solution of $H^T \tilde{y} \leq 0$ is $\tilde{y} = 0$, which contradicts the fact that co-ordinates of vector $(\tilde{y}, t)$ sums to one (see (5.7)).

Case 2: $s = 0$ and $t > 0$

Then, we have $\pi_2 = s = 0$ and therefore, $-H' \tilde{x} \leq 0$. Recall that $H' = H - \sum_{l=1}^{k} u^l \cdot v^l$ and $v^T \cdot \tilde{x} = \tilde{x}_{n+2l}$. Further, due to property $(P_4)$, $\forall l \in [k], \quad (H' \tilde{x})_{n+2l} = \tilde{x}_{n+2l-1} + \tilde{x}_{n+2l+1}$. And, due to $(P_2)$ we have $(H^T \tilde{y})_{n+2l} = \tilde{y}_{n+2l}$.

Now, for an $l \in [k]$ if $\tilde{x}_{2+l} > 0$, then $(H^T \tilde{y})_{n+2l} = \pi_1 \Rightarrow \tilde{y}_{n+2l} = \pi_1 > 0$ (using (5.8) and $t > 0$). However, the $n + 2l$th strategy of the second player is not fetching the maximum payoff, because $-H' \tilde{x}_{n+2l} \leq 0$, a contradiction.

Thus, we have $\tilde{x}_{2+l} = 0, \forall l \in [k]$. Then $H' \tilde{x} = H \tilde{x}$. Further, the best response condition of the first player gives $(\tilde{x}, s)^T A(\tilde{y}, t) = \pi_1 \Rightarrow \tilde{x}^T H^T \tilde{y} = \pi_1 > 0$, and the best response condition of the second player gives $(\tilde{x}, s)^T B(\tilde{y}, t) = \pi_2 \Rightarrow \tilde{x}^T H^T \tilde{y} = 0$ a contradiction.

Case 3: $s = 0$ and $t = 0$

If $\pi_1 > 0$ and $\pi_2 > 0$, then due to conditions (5.7) and (5.8), vector $\tilde{x} = (\tilde{x}, \tilde{y}) \neq 0$ is a solution of LCP $Mz \leq q, z \geq 0, z^T(Mz - q) = 0$, where $M = \begin{bmatrix} 0 & H^T \\ -H' & 0 \end{bmatrix}$ and $q = (\pi_1 \ast 1, \pi_2 \ast 1) > 0$. This contradicts Lemma 5.8.

If $\pi_1 = 0$, then the argument is similar to Case 1. If $\pi_1 > 0$ and $\pi_2 = 0$, then it is similar to Case 2. □

Now, we have established all the required facts to obtain the main theorems. Using Lemmas 5.10, 5.9, 5.6, 5.3, and 5.4, we show the next theorem.

Theorem 5.11. Given a kD-Linear-FIXP function $F$ defined by circuit $C$, there exists a bimatrix game $(\hat{A}, \hat{B})$ with rank$(\hat{A} + \hat{B}) \leq (k + 1)$, and $\hat{A}$ upper-triangular, such that the Nash equilibrium strategies of the first player in game $(\hat{A}, \hat{B})$ are in one-to-one correspondence with the fixed-points of $F$, where size$(\hat{A}) + \text{size}(\hat{B}) \leq \text{poly(size}(C))$. 

20
Proof. From circuit \(C\) of \(F\) construct \(F^{lp}\) of (5.3), then LCP of (5.5) from \(F^{lp}\), and finally game \((\hat{A}, \hat{B})\) of (5.6) from the LCP. Using Lemmas 5.3 and 5.6 it follows that solution vectors \(x\) of LCP are in one-to-one correspondence with the fixed-point of \(F^{lp}\), which are exactly the fixed-points of function \(F\) in Linear-FIXP that we started with.

Further, the Nash equilibrium \((\hat{x}, s), (\hat{y}, t)\) of LCP (due to Lemmas 5.9 and 5.10). And, two NE with distinct first players strategies \((\hat{x}, s) \neq (\hat{x}', s')\) can not map to the same \(x\) in a solution of LCP. If they do, then we have \(\frac{\hat{x}}{s} = \frac{\hat{x}'}{s'} \Rightarrow s' \sum_i \hat{x}_i = s \sum_i \hat{x}_i' \Rightarrow s'(1-s) = s(1-s') \Rightarrow s' = s \Rightarrow \hat{x} = \hat{x}'\), a contradiction.

Thus we get a game \((\hat{A}, \hat{B})\) whose Nash equilibrium strategies of the first player are in one-to-one correspondence with the fixed-points of \(F\). Since \(H' = H - \sum_{l=1}^{k} u lv_l^T\), \(\text{rank}(\hat{A} + \hat{B}) \leq k+1\), and since \(H\) is upper-triangular, \(\hat{A}\) is also upper-triangular. The size of matrices \(A\) and \(B\) is bounded by polynomial in size of \(A, b, c\) and \(u_l\), \(\forall l \in [k]\), and hence the theorem follows using Lemma 5.4. \(\square\)

Using Theorems 3.7 and 5.11, we get the next theorem.

**Theorem 5.12.** Nash equilibrium computation in bimatrix games with rank-\(k\), \(k \geq 3\) is PPAD-hard.

Since matrix \(\hat{A}\) is upper-triangular, we get the following corollary.

**Corollary 5.13.** Nash equilibrium computation in constant rank bimatrix games with one of the matrix being lower/upper-triangular is PPAD-hard.

NE computation in a bimatrix game \((A, B)\) can be reduced to computing a symmetric NE of a symmetric bimatrix game \((S, S^T)\) where \(S = \begin{bmatrix} 0 & \hat{A} \\ B^T & 0 \end{bmatrix}\) [36]. Note that if, \(\text{rank}(A + B) = k\) then \(\text{rank}(S + S^T) = 2k\), and therefore using Theorem 5.12 we get,

**Corollary 5.14.** Computing a symmetric Nash equilibrium of a symmetric game with rank-\(k\), \(k \geq 6\), is PPAD-hard.

Theorem 5.12 settles a decade long open problem. Since there is an FPTAS for constant rank games [25], this result comes as a surprise, because until now whenever a problem, in games or markets, was shown to be PPAD-hard, so was its approximation (i.e., no FPTAS unless PPAD = P) [15, 14, 1, 12, 26]. The approach of graphical games and game gadgets, central to the previous reductions, inherently give rise to higher rank games. Therefore, to obtain a lower rank game a conceptual leap seemed necessary. We bypass all these machinery and instead use connections between LPs, LCPs and bimatrix games to ensure low rank.

On the other hand previous reductions also give hardness of approximation. In particular, computing \(\frac{1}{poly(n)}\)-approximate Nash equilibrium of a bimatrix game is PPAD-hard [15]. Towards obtaining alternative proof of this fact, in the next section we show that approximating Linear-FIXP problem is hard. Extending it to show hardness for computing approximate Nash equilibrium remains open. It will be interesting to see if the techniques developed in this section or Section 4 can be used for the same.

### 6. Linear-FIXP: Hardness of Approximation

Chen et. al. [15] showed that higher dimensional discrete fixed-point problem (defined below) is PPAD-hard even when the associated grid has constant length in each dimension. Using this result, in this section we show inapproximability results for Linear-FIXP, by reducing a discrete fixed-point problems to finding an approximate solution of a Linear-FIXP
problem; the reduction is similar to that of Section 3. An approximate fixed-point can be defined as follows:

**Definition 6.1. Vector** \( \mathbf{x} \in [0, 1]^k \) is an \( \epsilon \)-approximate fixed-point of function \( F : [0, 1]^k \to [0, 1]^k \) if \( \| \mathbf{x} - F(\mathbf{x}) \|_{\infty} \leq \epsilon \).

Similar to 2D-Brouwer, let \( kD \)-Brouwer represent the class of \( k \)-dimensional discrete fixed-point problems. An instance of \( kD \)-Brouwer consists of a grid \( G^k_n = \{0, \ldots, 2^n - 1\}^k \), and a valid coloring function \( g : G^k_n \to \{0, 1, \ldots, k\} \), which satisfies the following: Let \( \partial(G^k_n) \) denote the set of points \( \mathbf{p} \in G^k_n \) with \( p_i \in \{0, 2^n - 1\} \) for some \( i \), i.e., boundary points, then,

For \( \mathbf{p} \in \partial(G^k_n) \), if \( p_i > 0, \forall i \in [k] \) then \( g(\mathbf{p}) = 0 \), otherwise \( g(\mathbf{p}) = \max\{i \mid p_i = 0, i \in [k]\} \)

Let \( K_{\mathbf{p}} = \{q \mid q_i \in \{p_i, p_i + 1\}\} \) be the set of vertices of a unit hyper-cube with \( \mathbf{p} \) at the lowest-corner. As discussed in [15], given any valid coloring \( g \) of \( G^k_n \), \( \exists \mathbf{p} \in G^k_n \) such that the vertices of hyper-cube \( K_{\mathbf{p}} \) have all \( k + 1 \) colors; \( K_{\mathbf{p}} \) is called a panchromatic cube. However, since there are \( 2^k \) vertices in a hyper-cube, given \( \mathbf{p} \) there is no efficient way to check if \( K_{\mathbf{p}} \) is panchromatic. Therefore, Chen et. al. introduces the following notion of discrete fixed-points.

**Definition 6.2 (Panchromatic Simplex [15]).** A subset \( P \subset G^k_n \) is accommodated if \( P \subset K_{\mathbf{p}} \) for some point \( \mathbf{p} \in G^k_n \). It is a panchromatic simplex of a color assignment \( g \) if it is accommodated and contains exactly \( k + 1 \) points with \( k + 1 \) distinct colors.

From the above discussion it follows that for any valid coloring \( g \) on \( G^k_n \), there exists a panchromatic simplex in \( G^k_n \) [15]. Similar to 2D-Brouwer the coloring function \( g \) is specified by a \( kD \)-Brouwer mapping circuit \( B \).

**kD-Brouwer Mapping Circuit:** The circuit has \( kn \) input bits, \( n \) bits for each of the \( k \) integers representing a grid point, and \( 2k \) output bits \( \Delta^+\), \( \Delta^- \), \( \forall i \in [k] \). It is a valid Brouwer-mapping circuit if the following is true:

- For every \( \mathbf{p} \in G_n \), the \( 2k \) output bits of \( B \) satisfies one of the following \( k + 1 \) cases:
  - Case 0: \( \forall i \in [k], \Delta^+_i = 1 \) and \( \Delta^-_i = 0 \).
  - Case \( i \in [k] \): \( \Delta^+_i = 1 \) and all the other \( 2k - 1 \) bits are zero.
- For every \( \mathbf{p} \in \partial G^k_n \), if \( \exists i \in [k] \) with \( p_i = 0 \) then letting \( i_{\text{max}} = \max\{i \mid p_i = 0\} \), the output bits satisfy Case \( i_{\text{max}} \), otherwise they satisfy Case 0.

Such a circuit \( B \) defines a valid color assignment \( g_B : G^k_n \to \{0, 1, \ldots, k\} \) by setting \( g_B(\mathbf{p}) = i \), if the output bits of \( B \) evaluated at \( \mathbf{p} \) satisfy Case \( i \). Let \( e^i \) be a \( k \)-dimensional unit vector with 1 on \( i^{th} \) coordinate, and \( e^0 \) be a vector with all \( k \) coordinates set to \(-1\). Then, a \( k \)-dimensional vector \( I \) set to \( I_i = \Delta^+_i - \Delta^-_i \), \( \forall i \in [k] \) is \( e^i \) for Case \( i \). This defines a discrete function \( H : G^k_n \to G^k_n \) where \( H(\mathbf{p}) = \mathbf{p} + e^{g_B(\mathbf{p})} \).

Given a \( kD \)-Brouwer mapping circuit \( B \) on grid \( G^k_n \), next we construct a \( kD \)-Linear-FIXP circuit \( C \) defining a function \( F : [0, 2^n - 1]^k \to [0, 2^n - 1]^k \), which is an extension of function \( H \). We show that all the \( \frac{1}{\text{poly}(\mathcal{L})} \)-approximate fixed-points of \( F \) are in panchromatic cubes of \( G^k_n \), where \( \mathcal{L} \) is the size of circuit \( B \). Further, we give a polynomial time procedure to compute a panchromatic simplex from an approximate fixed-point. When we reduce the range from \([0, 2^n - 1]^k \) to \([0, 1]^k \), to bring the function in to a standard form of Linear-FIXP, the approximation factor becomes \( \frac{1}{2^n \text{poly}(\mathcal{L})} \).

Recall that circuit \( C \) has \( k \) real inputs and outputs, \( \{\max, +, \ast, \zeta\} \) operations, and rational constants. The construction is almost same as that in Section 3. Let
Let $L > k^4$ be a large integer with value being a power of 2, and at most polynomial in $\text{size}(\mathcal{B})$, i.e., $L = 2^l \leq \text{poly}(\text{size}(\mathcal{B}))$. As in Definition 3.1 well-positioned and poorly-positioned points of $\mathbb{R}^k$ may be defined. Further, for $p \in [0, 2^n)^k$, let $\pi(p) = \{p\}$ and $\zeta(p) = e^{2\pi i \pi(p)}$.

For a well-positioned point $p \in [0, 2^n)^k$ the bit representation of each coordinate of $\pi(p)$ can be computed in $C$ using ExtractBits procedure of Table 1 (due to Lemma 3.2). This bit representation, when fed to a simulation of $B$ where $\land$, $\lor$ and $\neg$ are replaced by min, max and $(1 - x)$ respectively, outputs $2k$ values which is exactly $B(\pi(p))$. However, it is still not clear how to efficiently check if hyper-cube $K_Q$ is panchromatic because it has $2^k$ vertices. Further, if $p$ is poorly-positioned to start with, then it is not clear how to compute even the bit representation of $\pi(p)$ using operations of Linear-FIXP.

To circumvent these issues we use a geometric lemma proved by Chen et. al. [15], described next. For a finite set $S \subseteq \mathbb{R}^k_+$, let $I_w(S)$ contain the indices of the well-positioned points of $S$ and $I_p(S)$ contain indices of poorly-positioned points.

**Lemma 6.3.** [15] Given $p \in [0, 2^n - 1]^k$, consider the set $S = \{p^1, \ldots, p^{k^4}\}$ such that

$$p^j = p + \frac{(j - 1)}{L} \sum_{i \in [k]} e^i, \quad j \in [k^4]$$

For each $j \in I_p(S)$, let $r^j \in \mathbb{R}^k$ be a vector with $\|r^j\|_{\infty} \leq 1$. And for each $j \in I_w(S)$, let $r^j = \zeta(p^j)$. If $\|\sum_{j=1}^{k^4} r^j\|_{\infty} < 1$ then $Q_w = \{\pi(p^j) \mid j \in I_w(S)\}$ is panchromatic simplex.

**Proof.** Let $Q = \{q^i = \pi(p^i) \mid p^i \in S\}$. Since $\frac{k^4}{L} << 1$ the set crosses boundaries of the unit cells at most $k$ times. In other words, for each $i \in [k]$, there is at most one $j_i$ such that $q_i^{j_i} = q_i^{j_i - 1} + 1$. Therefore, set $Q$ can have at most $k + 1$ elements, and they are part of the same unit hyper-cube, which has to be $K_{\pi(p)}$. Clearly, $Q_w \subseteq Q$.

Further, since $\frac{1}{L} << \frac{1}{k} << 1$, there can be at most $k$ poorly-positioned points in $S$. So, we have $|I_w(S)| \geq k^4 - k$. Let $r^G = \sum_{j \in I_w(S)} r^j$, then we have $\|r^G + \sum_{j \in I_p(S)} r^j\|_{\infty} < 1 \Rightarrow \|r^G\|_{\infty} < 1 + \|\sum_{j \in I_p(S)} r^j\|_{\infty} < k + 1$, because $|I_p(S)| \leq k$, and $\|r^j\|_{\infty} \leq 1$ for each $j \in I_p(S)$.

Let $\forall i \in [k]$, $W_i$ be the number of indices of $I_w(S)$ with $r^k = e^i$. Using the above fact, we will show that $W_i \neq 0, \forall i$, to prove the lemma.

If $W_0 = 0$ then $W_i > k^2$ for some $i \in [k]$. In that case, $r^G_i \geq k^2$, a contradiction. If $W_i = 0$ for a $t \in [k]$, then $W_0 < k + 1$ or else $r^G_i \geq k + 1$. Let $i^* = \arg \max_{0 \leq i \leq k} W_i$, then clearly, $W_{i^*} > k^3 - 1$ and $i^* \neq 0$. Then, $r^G_{i^*} \geq k^3 - 1 - k$, again a contradiction.

Using Lemma 6.3 we can construct circuit $C$ as done in steps $(S_1)$ to $(S_6)$ in Section 3, where instead of $16$, $k^4$ points have to be sampled, and finally in step $(S_6)$ the incremental vector $\sum_j r^j$ has to be divided by $k^4$ in order to take an average. This circuit will define a piecewise-linear function $F : [0, 2^n - 1]^k \rightarrow [0, 2^n - 1]^k$.

Next, we show that it suffices to compute a $\frac{1}{L}$-approximate fixed-point of $F$ in order to find a panchromatic simplex.

**Lemma 6.4.** Every $\frac{1}{L}$-approximate fixed-point of $F$ is in a panchromatic hyper-cube of $G_{n,k}$.

**Proof.** Let $p$ be a $\frac{1}{L}$-approximate fixed-point of $F$, and $p' = F(p)$. Then, the set $S$ of sampled points is $S = \{p^j = p + \frac{(j - 1)}{L} \sum_{i \in [k]} e^i \mid j \in [k^4]\}$, and $r^j$ is the
outcome vector in step (S₄) for  \( \mathbf{p}' \). By construction, we have  \( r^j = \zeta(\mathbf{p}') \),  \( \forall j \in I_\mathcal{u}(S) \), and  \( \|r^j\|_\infty \leq 1 \),  \( \forall j \in [k^4] \). Further,  \( r \) is the average of  \( r^j \)'s, and hence  \( \|r\|_\infty \leq 1 \).

Suppose,  \( \mathbf{p} \) is not inside a panchromatic hyper-cube, then  \( \|r\|_\infty \geq \frac{1}{\mathcal{T}} \) by Lemma 6.3. If  \( \mathbf{p} \) is at least  \( \frac{1}{\mathcal{T}} \) distance away from the boundary of  \([0, 2^n - 1]^k\), i.e.,  \( \frac{1}{\mathcal{T}} \leq p_i \leq 2^n - 1 - \frac{1}{\mathcal{T}} \);  \( \forall i \in [k] \), then clearly  \( \|\mathbf{p} - F(\mathbf{p})\| \geq \frac{1}{\mathcal{T}} \), a contradiction.

For the points near boundary it may happen that  \( \|r\|_\infty \geq \frac{1}{\mathcal{T}} \), but still due to rounding in step (S₆), they generate dummy fixed-points. Using the fact that  \( \mathcal{B} \) generates a valid coloring, we show that this can never happen. Let  \( \mathbf{p} \) be such that for some  \( i \) either  \( p_i < \frac{1}{\mathcal{T}} \) or  \( p_i > 2^n - 1 - \frac{1}{\mathcal{T}} \).

-  \( \exists i \in [k], p_i < \frac{1}{\mathcal{T}} \): Let  \( i_{\text{max}} = \max\{i \mid p_i < \frac{1}{\mathcal{T}}\} \), then  \( \forall j \in [k^4]p_{i_{\text{max}}} < 1 \).
  Therefore,  \( \exists i' \geq i_{\text{max}} \) such that  \( p_i < 1 \) and  \( r_{i'} > 0 \), implying that  \( p_{i'} > p_i \).
-  \( \forall i \in [k], p_i > 1 \): Since  \( \exists i' \) with  \( p_{i'} > 2^n - 1 - \frac{1}{\mathcal{T}} \), except for  \( \mathbf{p}' \) all other  \( \mathbf{p}^j \) are outside of  \([0, 2^n - 1]^k\), and  \( \pi(\mathbf{p}') > 0 \). Therefore,  \( \forall j \in I_\mathcal{u}(S), j \neq 1 \), we have  \( r^j = e^0 < 0 \). Hence  \( \exists \iota \), such that  \( p_{\iota} < p_i \).
-  \( \exists \rho, i' \in [k], p_{\rho} < 1 \) and  \( p_{i'} > 2^n - 1 - \frac{1}{\mathcal{T}} \): Let  \( i_{\text{max}} = \max\{i \mid p_i < 1\} \), then  \( \exists \iota'' \leq i_{\text{max}} \) such that either  \( p_{\iota''} < 1 \) and  \( r_{\iota''} > 0 \) implying that  \( p_{\iota''} > p_i \), or  \( r_{\iota''} < 0 \) implying that  \( p_{\iota''} < p_i \).

\[ \square \]

If  \( \mathbf{p} \) is a  \( \frac{1}{\mathcal{T}} \)-approximate fixed-point of  \( F \), then it is in panchromatic hyper-cube of  \( G_{n,k} \) (Lemma 6.4), and the panchromatic simplex containing  \( \mathbf{p} \) is  \( \{\pi(\mathbf{p}') \mid \mathbf{p}' = \mathbf{p} + \frac{1}{L-1}\mathbf{1} = \sum_{i \in [k]} e^i\} \) (Lemma 6.3). Therefore given a  \( \frac{1}{\mathcal{T}} \)-approximate fixed-point of  \( F \) a panchromatic simplex of  \( G_{n,k} \) can be computed in polynomial time.

We can shrink the range of function  \( F \) from  \([0, 2^n - 1]\) to  \([0, 1]^k\) by multiplying and dividing the inputs and outputs respectively by  \( 2^n - 1 \). For the modified function,  \( \frac{1}{\mathcal{T}} \)-approximate fixed-points are guaranteed to be in panchromatic hyper-cubes. Note that, Lemmas 6.3 and 6.4 holds for any  \( \mathcal{T} \) strictly greater than  \( k^4 \), hence  \( \frac{1}{\mathcal{T}} = \frac{1}{2^n \text{poly}(k)} \). Further, by construction size\([C] = (\#\text{inputs} + \# \text{gates} + \text{total size of the constant used in } C)\), is polynomial in size\([\mathcal{B}]\).

Therefore, a  \( kD \)-Brouwer problem of computing a panchromatic simplex reduces to finding a  \( \frac{1}{\gamma \text{poly}(k)} \)-approximate fixed-point of a  \( kD \)-Linear-FIXP function, where  \( \gamma \) is the largest absolute constant used in the circuit. Chen et. al. [15] proved that  \( kD \)-Brouwer with  \( n = 3 \) and  \( k \) not a constant is PPAD-hard (Brouwer  \( f^j \) in [15]). Since the largest absolute constant used in the  \( kD \)-Linear-FIXP circuit constructed from such an instance is of  \( O(1) \), the next theorem follows.

**Theorem 6.5.** Let  \( F \) be a piecewise-linear function defined by a Linear-FIXP circuit  \( C \), and let  \( \mathcal{L} = \text{size}[C] \). Then Computing a  \( \frac{1}{\gamma \text{poly}(\mathcal{L})} \)-approximate fixed-point of  \( F \) is PPAD-hard.

**Remark 6.6.** We note that Theorem 6.5 may also follow from the 2-Nash to Linear-FIXP reduction shown by Etessami and Yannakakis [22].

**7. Discussion.** In this paper we showed that Nash equilibrium computation in bimatrix games with rank  \( \geq 3 \) is PPAD-hard by reducing  \( 2D \)-Brouwer to rank-3 games. Given an instance of  \( 2D \)-Brouwer first we reduced it to  \( 2D \)-Linear-FIXP, a 2-dimensional fixed-point problem defined by a Linear-FIXP circuit with two inputs. Next we replaced the circuit by a parameterized linear program with two parameters, and finally using the connections between LPs and LCPs, and LCPs and bimatrix games, we constructed a rank-3 game. This, last step of the reduction uses the fact that the parameterized linear program was constructed from a Linear-FIXP circuit. It will be interesting to reduce a fixed-point problem, defined by a parameterized LP,
to a bimatrix game in general. This will extend the classical construction of zero-sum games from linear programs by Dantzig [18]. If fixed-point problem with $k$-parameter LP can be reduced to a rank-$k$ game, then it will imply that rank-2 games are also PPAD-hard, settling the only unresolved case.

As corollaries of our reduction, we get that $2D$-Linear-FIXP = PPAD = Linear-FIXP, and in turn a sharp dichotomy on complexity of Linear-FIXP problems; $1D$-Linear-FIXP is in P, while for $k \geq 2$, $kD$-Linear-FIXP is PPAD-complete. We also give an explicit construction of a rank-$(k+1)$ game from a $kD$-Linear-FIXP problem. This construction (almost) preserves the number of solutions (in terms of NE strategies of the first player), and is different from all the previous approaches. This may be of independent interest to understand the connections between problems that reduces to Linear-FIXP and bimatrix games. One such example is Theorem 4.7 which shows that even if Nash equilibrium set of a bimatrix game is guaranteed to be convex, finding one is as hard as solving a simple stochastic game, an extensively studied problem [22].

In Section 6 we show hardness of approximation for Linear-FIXP problems. It is not immediately clear how to extend this to bimatrix games through our reduction. If done, it will provide an alternate (simpler) proof of inapproximability in 2-Nash [15].

For the case of symmetric games, the PPAD-hardness of rank-3 games imply that computing a symmetric Nash equilibrium in a symmetric rank-6 games is PPAD-hard. The polynomial time algorithm for computing symmetric NE of a rank-1 symmetric games by Mehta et. al. [32] leaves the status of symmetric games with rank-2 to rank-5 unresolved.

Acknowledgments. I wish to thank Vijay V. Vazirani, Milind Sohoni, Bernhard von Stengel, Mihalis Yannakakis, and Jugal Garg for valuable discussions.

REFERENCES

Consider the symmetric game \((S, S^T)\). Using Lemma 2.3 we get that a mixed strategy vector \(z = (x, t) \in \mathbb{R}^{(m+1)}\) is a symmetric NE of game \((S, S^T)\) if and only if
\[
\begin{align*}
\mathbf{A}x &= b; \quad t \geq 0; \\
\mathbf{A}^T x + bt &= 0; \\
\sum_{i \in [m]} x_i &= 1; \\
-t &= \pi;
\end{align*}
\]  
(A.1)

where \(\pi\) is the payoff \(z^T S z\) of both the agents at NE \((z, z)\).

**Lemma A.1.** Strategy \(z = (x, t)\), with \(t > 0\), is a symmetric NE of game \((S, S^T)\) iff \(x' = \frac{x}{t}\) is a solution of LCP (4.5). Further, for symmetric NE \(z = (x, t)\) and \(\hat{z} = (\hat{x}, \hat{t})\) with \(\hat{t} > 0\) if \(\frac{\hat{x}}{\hat{t}} = \frac{x}{t}\) then \(z = \hat{z}\).

**Proof.** If \(t > 0\) then the third condition of (A.1) ensures that \(\pi = t\). In that case, the second inequality becomes \(\mathbf{A} x' \geq \mathbf{b}\), and the third equality becomes \(\frac{1}{\hat{t}}(\mathbf{A} x' - \mathbf{b})_i = 0, \forall i \in [m]\), which are exactly the conditions of LCP (4.5). Therefore, \(x' = \frac{x}{t}\) is a solution of the LCP (4.5).

Further, if \(x'\) is a solution of the LCP, then for \(t = \frac{1}{1 + \sum_{i} x_i}\), \(x_i = tx'_i\) and \(\pi = t = (x, t, \pi)\) satisfies all the conditions of (A.1), and hence the lemma follows.

For the second part suppose \(\hat{t} = \alpha \ast t\) for some \(\alpha > 0\). Then \(\forall i, \frac{x_i}{\hat{t}} = \frac{x_i}{t} \Rightarrow \hat{x}_i = \alpha x_i\) and therefore \(\hat{z} = \alpha z\). However, \(z\) and \(\hat{z}\) being probability distributions imply \(\alpha = 1\), and the lemma follows. \(\square\)

**Lemma A.1** shows that symmetric NE of game \((S, S^T)\) with \(t > 0\) are in one-to-one correspondence with the solutions of LCP (4.5). One-to-one because clearly no two symmetric NE of game \((S, S^T)\) maps to the same solution of LCP (4.5). Next, we show that these are the only Symmetric NE of this game.

**Lemma A.2.** If \(z = (x, t)\) is a symmetric NE of game \((S, S^T)\) then \(t > 0\).

**Proof.** To the contrary suppose \(t = 0\), then \(\pi \geq t = 0\), and \(-A' x \leq \pi\). Recall that \((-A' x)_{n+2l} = -x_{n+2l-1} - x_{n+2l}, \forall l \in [k]\) using (P1). Now, if \(x_{n+2l} > 0\) then \((-A' x)_{n+2l} < 0\) which contradicts the third condition of (A.1). Therefore, we have \(\forall l \in [k], x_{n+2l} = 0\) implying that \(A' x = Ax\) because \(v^T x = x_{n+2l} = 0, \forall l \in [k]\). Let \(i^*\) be the first strategy played with the non-zero probability, i.e., \(i^* = \text{argmin}_{x_i > 0, i \in [m]} x_i\). The payoff from \(i^*\) should be maximum and hence \((-A' x)_{i^*} = x_{i^*} < 0\), a contradicting \(0 = t \leq \pi\). \(\square\)