A Polynomial-Time Algorithm for 1/2-Well-Supported Nash Equilibria in Bimatrix Games*

Argyrios Deligkas† Michail Fasoulakis‡ Evangelos Markakis§

Abstract

Since the seminal PPAD-completeness result for computing a Nash equilibrium even in two-player games, an important line of research has focused on relaxations achievable in polynomial time. In this paper, we consider the notion of ε-well-supported Nash equilibrium, where $\varepsilon \in [0, 1]$ corresponds to the approximation guarantee. Put simply, in an ε-well-supported equilibrium, every player chooses with positive probability actions that are within ε of the maximum achievable payoff, against the other player's strategy. Ever since the initial approximation guarantee of 2/3 for well-supported equilibria, which was established more than a decade ago, the progress on this problem has been extremely slow and incremental. Notably, the small improvements to 0.6608, and finally to 0.6528, were achieved by algorithms of growing complexity. Our main result is a simple and intuitive algorithm, that improves the approximation guarantee to 1/2. Our algorithm is based on linear programming and in particular on exploiting suitably defined zero-sum games that arise from the payoff matrices of the two players. As a byproduct, we show how to achieve the same approximation guarantee in a query-efficient way.

1 Introduction

The computation of a Nash equilibrium has been one of the most fundamental problems in the intersection of computer science and economics. The non-constructive proof of existence by Nash for finite games [38], motivated a significant volume of works to focus on the quest for efficient algorithms. This quest was terminated by the landmark results of [17, 10], establishing PPAD-completeness, even for two-player games.

The above intractability results naturally led to the study of polynomial-time algorithms for approximate Nash equilibria. But as this is not a typical optimization problem with a single objective to optimize, there exist in the literature different ways of defining approximate solutions. Among them, the two most prevalent notions of approximability are those of ε-Nash equilibria and ε-well-supported Nash equilibria (ε-WSNE), for $\varepsilon > 0$, assuming that all payoffs of the game have been scaled to be in $[0, 1]$. In an ε-Nash equilibrium, the expected payoff of any player is at most ε less than the best-response payoff. In an ε-well-supported Nash equilibrium, any pure strategy that is played with a positive probability, should have a payoff of at most ε less than the best-response payoff. It is easy to see that the latter is a stronger approximation concept, as any ε-well-supported Nash equilibrium is an ε-Nash equilibrium, but the other direction does not always hold.

The research efforts on these two notions have not evolved equally well. For ε-Nash equilibria, there was a steady progress, starting with a $\frac{3}{4}$-approximation in [28], and gradual improvements to $\frac{1}{2}$ by [19], 0.382 in [18], 0.364 by [7], and $(0.3393 + \delta)$ by [42], for any constant $\delta > 0$. Finally, a very recent improvement by [20], has obtained the currently best approximation of $(\frac{1}{3} + \delta)$, for any constant $\delta > 0$. On the other hand, the first result for ε-well-supported Nash equilibria was given by Kontogiannis and Spirakis [29], with a $\frac{2}{3}$-approximation. This was obtained by a quite "clean" and easy to implement algorithm, requiring only a single call to a linear programming solver. After this, the progress on this front has been very incremental and with algorithms of growing complexity. A refinement of the algorithm by [29] was analyzed in [24], obtaining a 0.6608-approximation. Later on, a more involved algorithm was provided by [14], achieving an additional improvement to 0.6528. The guarantee of 0.6528 has been the state of the art for the last 7 years [13].

For both approximation notions, it is well known that there exist quasi-polynomial time algorithms [32, 29] for any constant $\varepsilon > 0$, running in time $n^{O(\log n/\epsilon^2)}$, for a game with n available pure strategies per player.

*The full version of the paper can be accessed at https://arxiv.org/abs/2207.07007

†Royal Holloway, University of London.

‡Foundation for Research and Technology-Hellas and Athens University of Economics and Business.

§Athens University of Economics and Business and Input Output Global (IOG).

Quite interestingly, for well-supported equilibria, there exists an improved quasi-polynomial time algorithm for computing a $(\frac{1}{2}+\delta)$-approximation, for any constant $\delta > 0$ [23], making the exponent dependent on $\log \log n$, rather than $\log n$. This can be viewed as supporting evidence that further improvements on the running time might be feasible for obtaining a $(\frac{1}{2}+\delta)$-well-supported approximation, and even a polynomial time algorithm could be anticipated.

Finally, regarding impossibility results, we know by the work of Rubinstein [41] that, assuming the exponential-time hypothesis for PPAD, then ε-Nash equilibria, and consequently ε-well-supported equilibria as well, require quasi-polynomial time, when ε is below some very small, yet unspecified, constant. This rules out a PTAS for both approximation notions. But especially for well-supported equilibria, it also leaves open a very large gap from the 0.6528-approximation.

1.1 Our contribution We improve upon the state of the art on the computation of approximate well-supported Nash equilibria in two respects. Firstly, we derive a polynomial-time algorithm (Algorithm 1) that, for every constant δ, computes a $(\frac{1}{2}+\delta)$-WSNE. Our algorithm is more intuitive and conceptually much simpler compared to the algorithm of [14] that finds a 0.6528-WSNE. Then, we use Algorithm 1 as a basis for a query-efficient algorithm (Algorithm 2) that achieves the same approximation guarantee.

Algorithm 1 (main result): To provide a high level description, the first step in Algorithm 1 is to compute a Nash equilibrium in the zero-sum games $(R, -R)$ and $(-C, C)$, where R and C are the payoff matrices of the row and column player respectively. These equilibrium-strategies, and their corresponding payoffs, are utilized to partition the remaining analysis into three cases that the algorithm considers sequentially. The first and easier case occurs when the solutions of the two zero-sum games induce a strategy profile, where the maximum possible payoff for each player under the chosen strategy of their opponent is "low", i.e., it is bounded by $\frac{1}{2}$. This directly yields a 1/2-approximation. The remaining two cases (referred to as the low-high and the high-high cases) are more interesting and in these cases we utilize either a linear system solver, or a support enumeration argument, to construct the desirable strategy profile.

Algorithm 2: query-efficient algorithm. In order to convert Algorithm 1 into a query-efficient protocol we utilize the fact that ε-WSNEs in zero-sum games can be found with $O(\frac{n \cdot \log n}{\varepsilon^2})$ queries. Then, using the approximate well-supported Nash equilirbia we found, we resort to *sampling* in order to create a subgame with $O(\log n)$ rows and $O(n)$ columns (or the other way around, depending on the case we are handling). This subgame is guaranteed to have a strategy profile that is a $(\frac{1}{2}+\delta+3\varepsilon)$-WSNE for the original game. In order to find such a profile the algorithm simply queries the whole subgame and it proceeds as our main algorithm.

1.2 Further related work Apart from results on general two-player games, there have been several studies focusing on the approximability of Nash equilibria in specific classes of games. For instance, for constant-rank games, where the matrix defined by the sum of the two payoff matrices has constant rank [1, 27, 36]; for win-lose games [11, 12, 33]; for sparse games [9, 5], for imitation games [34, 35, 37]; for random games [4, 39]; for symmetric games [15, 30] and for games with symmetric payoff matrices [16].

There have also been results on new quasi-polynomial algorithms (QPTAS). We mention three new QPTASs that have been obtained: [5] gave a refined, parameterized, approximation scheme; [3] gave a QPTAS that can be applied to multi-player games as well; [21] gave a more general approach for approximation schemes for the existential theory of the reals. Furthermore, more negative results for ε-NE were derived: [31] gave an unconditional lower bound, based on the sum of squares hierarchy; [6] proved PPAD-hardness in the smoothed analysis setting; [2, 8, 22] gave quasi-polynomial time lower bounds for constrained ε-NE, under the exponential time hypothesis.

2 Preliminaries

In what follows, for any natural number n, we denote by $[n]$ the set $\{1, 2, \dots, n\}$ and by Δ^n the $(n-1)$-dimensional simplex. An $n \times n$ *bimatrix* game (R, C) is defined by two *payoff* matrices R and C of size $n \times n$ each: R defines the payoffs of the *row* player and C defines the payoffs of the *column* player. When the row player picks a row $i \in [n]$ and the column player picks a column $j \in [n]$, then they receive a payoff of R_{ij} and C_{ij}, respectively. We follow the usual assumption in the relevant literature that the matrices are normalized, so that all entries are in $[0, 1]$.

A *mixed strategy* is a probability distribution over $[n]$. We use $\mathbf{x} \in \Delta^n$ to denote a mixed strategy for the row player and $\mathbf{y} \in \Delta^n$ to denote a mixed strategy for the column player. If $\mathbf{x}$ and $\mathbf{y}$ are mixed strategies for the row and the column player respectively, then we call $(\mathbf{x}, \mathbf{y})$ a *strategy profile.* We denote by e_i the n-dimensional vector that has 1 at index i and 0 elsewhere. Hence, e_i corresponds to a pure strategy, where a player assigns probability one to play the pure strategy i. A useful notion in equilibrium computation is the *support* of a strategy $\mathbf{x}$, denoted by $\text{supp}(\mathbf{x})$, which is the set of pure strategies that are played with positive probability under $\mathbf{x}$. Formally, $\text{supp}(\mathbf{x}) := \{i \in [n] : \mathbf{x}(i) > 0\}$.

Given a strategy profile $(\mathbf{x}, \mathbf{y})$, the expected payoff of the row player is $\mathcal{R}(\mathbf{x}, \mathbf{y}) := \mathbf{x}^T R\mathbf{y}$, and the expected payoff of the column player is $\mathcal{C}(\mathbf{x}, \mathbf{y}) := \mathbf{x}^T C\mathbf{y}$. Hence, for a pure strategy e_i, the term $\mathcal{R}(e_i, \mathbf{y}) := \sum_j R_{ij} y_j$, denotes the expected payoff of the row player, when she plays the pure strategy i against strategy $\mathbf{y}$ of the column player. Similarly, $\mathcal{C}(\mathbf{x}, e_j)$ is the expected payoff of the column player when she plays the pure strategy j against $\mathbf{x}$. A pure strategy is a *best-response strategy* for a player against a chosen strategy of her opponent, if it maximizes her expected payoff; a pure strategy is an ε-best response, for $\varepsilon \in [0, 1]$, if it achieves a payoff that is at most ε less than the maximum possible payoff against the opponent's chosen strategy.

Well-supported Nash equilibria. A strategy profile $(\mathbf{x}, \mathbf{y})$ is an *ε-well-supported Nash equilibrium*, henceforth ε-WSNE, for some $\varepsilon \in [0, 1]$, if every player plays with positive probability only pure strategies that are ε-best responses. Put formally, $(\mathbf{x}, \mathbf{y})$ is an ε-WSNE if the following conditions are satisfied:

$$\mathcal{R}(e_i, \mathbf{y}) \geq \max_{k \in [n]} \mathcal{R}(e_k, \mathbf{y}) - \varepsilon \quad \text{for all } i \in \text{supp}(\mathbf{x});$$
$$\mathcal{C}(\mathbf{x}, e_j) \geq \max_{k \in [n]} \mathcal{C}(\mathbf{x}, e_k) - \varepsilon \quad \text{for all } j \in \text{supp}(\mathbf{y}).$$

If $\varepsilon = 0$, then 0-WSNE is an *exact* Nash equilibrium.

k-uniform strategies. For any natural number k, a *k-uniform strategy* plays each pure strategy with probability that is a multiple of $\frac{1}{k}$; i.e. $\mathbf{x}$ is k-uniform if for every $i \in [n]$ it holds that $\mathbf{x}(i) \in \{0, \frac{1}{k}, \frac{2}{k}, \ldots, \frac{k-1}{k}, 1\}$. If both $\mathbf{x}$ and $\mathbf{y}$ are k-uniform strategies, then we will say that $(\mathbf{x}, \mathbf{y})$ is a k-uniform strategy profile. We will use k-uniform strategies for different values of k in several parts of our algorithms. Observe that for any fixed value of k, the size of the support of a k-uniform strategy is at most k. This implies that in a $n \times n$ bimatrix game, there are at most $O(n^k)$ k-uniform strategy profiles. In addition, k-uniform strategies are used to produce *sampled strategies* as explained below.

Sampled strategies. Given a strategy $\mathbf{x}$, a k-uniform *sampled strategy* $\mathbf{x}_s$ is produced by taking k independent samples from $[n]$ according to $\mathbf{x}$, and by setting $\mathbf{x}_s(i)$ equal to the frequency with which pure strategy i was sampled. Observe that $\text{supp}(\mathbf{x}_s) \subseteq \text{supp}(\mathbf{x})$. Sampled strategies can approximate well any arbitrary strategy if enough samples are taken; roughly, if the sampled strategy has to satisfy m constraints with an ε-additive error, then $O(\frac{\log m}{\text{poly}(\varepsilon)})$ samples suffice.

Next, we list some lemmas that involve k-uniform sampled strategies that will be used later in the analysis of our algorithms.

LEMMA 2.1. (IMPLIED BY THE PROOF OF THEOREM 1 OF [32]) *Let $(\mathbf{x}, \mathbf{y})$ be a strategy profile of an $n \times n$ bimatrix game. Then, there exist $O(\frac{\log n}{\varepsilon^2})$-uniform sampled strategies $\mathbf{x}_s, \mathbf{y}_s$ such that:*

$$|\mathcal{R}(\mathbf{x}, \mathbf{y}) - \mathcal{R}(\mathbf{x}_s, \mathbf{y}_s)| \leq \varepsilon \quad \textit{and} \quad |\mathcal{R}(e_i, \mathbf{y}) - \mathcal{R}(e_i, \mathbf{y}_s)| \leq \varepsilon \quad \forall i \in [n];$$
$$|\mathcal{C}(\mathbf{x}, \mathbf{y}) - \mathcal{C}(\mathbf{x}_s, \mathbf{y}_s)| \leq \varepsilon \quad \textit{and} \quad |\mathcal{C}(\mathbf{x}, e_j) - \mathcal{C}(\mathbf{x}_s, e_j)| \leq \varepsilon \quad \forall j \in [n].$$

LEMMA 2.2. *Let $(\mathbf{x}, \mathbf{y})$ be an ε-WSNE of an $n \times n$ bimatrix game and let $\mathbf{x}_s, \mathbf{y}_s$ be $O(\frac{\log n}{\varepsilon^2})$-uniform strategies sampled from $\mathbf{x}$ and $\mathbf{y}$ respectively. Then, $(\mathbf{x}_s, \mathbf{y}_s)$ is a 3ε-WSNE for the game.*

Proof. Assume that we are considering the bimatrix game (R, C). We will focus on the row player, since the analysis for the second player is identical. Recall that since $\mathbf{x}_s$ is sampled from $\mathbf{x}$ it holds that $\text{supp}(\mathbf{x}_s) \subseteq \text{supp}(\mathbf{x})$.

So, consider an $i \in \text{supp}(\mathbf{x}_s)$:

$$\begin{aligned}
\mathcal{R}(e_i, \mathbf{y}_s) &\geq \mathcal{R}(e_i, \mathbf{y}) - \varepsilon && \text{(From Lemma 2.1)}\\
&\geq \max_{k\in[n]} \mathcal{R}(e_k, \mathbf{y}) - 2\varepsilon && \text{(Since } (\mathbf{x}, \mathbf{y}) \text{ is an } \varepsilon\text{-WSNE)}\\
&\geq \mathcal{R}(e_{k^*}, \mathbf{y}) - 2\varepsilon && (k^* \in \text{argmax}_k \mathcal{R}(e_k, \mathbf{y}_s))\\
&\geq \mathcal{R}(e_{k^*}, \mathbf{y}_s) - 3\varepsilon && \text{(From Lemma 2.1).}\\
&= \max_{k\in[n]} \mathcal{R}(e_k, \mathbf{y}_s) - 3\varepsilon
\end{aligned}$$

□

3 The algorithm

In this section we provide our main algorithm, that computes a $(\frac{1}{2}+\delta)$-WSNE in polynomial time. This algorithm will form the basis for obtaining an additional algorithm that still computes a $(\frac{1}{2}+\delta)$-WSNE and is also query efficient.

Algorithm 1: Main Algorithm for $(\frac{1}{2}+\delta)$-WSNE

```
Input: A bimatrix game $(R, C)$ and constants $\delta \in (0, 1]$, and $\kappa(\delta) = \left\lceil \frac{2\ln(1/\delta)}{\delta^2} \right\rceil$.
Output: A $(\frac{1}{2}+\delta)$-WSNE for $(R, C)$.
1. Compute a Nash equilibrium $(\mathbf{x}^*, \mathbf{y}^*)$ for the zero-sum game $(R, -R)$.
2. Compute a Nash equilibrium $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ for the zero-sum game $(-C, C)$.
3. If $\mathcal{R}(\mathbf{x}^*, \mathbf{y}^*) \geq \mathcal{C}(\hat{\mathbf{x}}, \hat{\mathbf{y}})$:
   (a) If $\mathcal{R}(\mathbf{x}^*, \mathbf{y}^*) \leq \frac{1}{2}$, then return $(\hat{\mathbf{x}}, \mathbf{y}^*)$.
   (b) Else, if there exists a strategy $\mathbf{x}'$ such that $\text{supp}(\mathbf{x}') \subseteq \text{supp}(\mathbf{x}^*)$, and $\mathcal{C}(\mathbf{x}', e_j) \leq \frac{1}{2}$ for every $j \in [n]$,
       then return $(\mathbf{x}', \mathbf{y}^*)$.
   (c) Else, exhaustively search over all $\kappa(\delta)$-uniform strategy profiles and find a $(\frac{1}{2}+\delta)$-WSNE.
4. Else if $\mathcal{C}(\hat{\mathbf{x}}, \hat{\mathbf{y}}) > \mathcal{R}(\mathbf{x}^*, \mathbf{y}^*)$:
   (a) If $\mathcal{C}(\hat{\mathbf{x}}, \hat{\mathbf{y}}) \leq \frac{1}{2}$, then return $(\hat{\mathbf{x}}, \mathbf{y}^*)$.
   (b) Else if there exists a strategy $\mathbf{y}'$ such that $\text{supp}(\mathbf{y}') \subseteq \text{supp}(\hat{\mathbf{y}})$, and $\mathcal{R}(e_i, \mathbf{y}') \leq \frac{1}{2}$ for every $i \in [n]$,
       then return $(\hat{\mathbf{x}}, \mathbf{y}')$.
   (c) Else, exhaustively search over all $\kappa(\delta)$-uniform strategy profiles and find a $(\frac{1}{2}+\delta)$-WSNE.
```

THEOREM 3.1. *For any constant $\delta > 0$, Algorithm 1 computes in polynomial time a $(\frac{1}{2}+\delta)$-WSNE.*

Before proving Theorem 3.1, we provide first a high level description. Algorithm 1 partitions the solution space into three cases that considers then sequentially. If the algorithm fails to find a solution within a specific case, this rules out the existence of certain strategy profiles and proceeds to the next case, where it exploits the extra constraints that are imposed from the failure of the previous cases. Before reaching the case-analysis though, the algorithm computes a Nash equilibrium $(\mathbf{x}^*, \mathbf{y}^*)$ for the zero-sum game $(R, -R)$ and a Nash equilibrium $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ for the zero-sum game $(-C, C)$. These equilibrium-strategies, and their corresponding payoffs, are utilized to provide bounds for the payoffs in the game (R, C).

- **Case (a): low payoffs.** Firstly, the algorithm checks if the solutions of the two zero-sum games can be used to produce a strategy profile, $(\hat{\mathbf{x}}, \mathbf{y}^*)$, where the maximum payoff for every player under the chosen strategy of their opponent is "low", i.e., it is bounded by $\frac{1}{2}$. If this is the case, then under the strategy profile $(\hat{\mathbf{x}}, \mathbf{y}^*)$, the regret of *any* action is bounded by $\frac{1}{2}$, and we have a $\frac{1}{2}$-WSNE.

- **Case (b): low-high payoffs.** If the first case fails, then the algorithm deduces that there is a strategy profile among $(\mathbf{x}^*, \mathbf{y}^*)$ and $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$, where at least one of the players is guaranteed a "high" payoff from every action they play with positive probability, i.e., every action in her support yields a payoff of at least $\frac{1}{2}$. Then, it formulates a linear-feasibility system that seeks a strategy for the high-payoff player yielding "low" maximum payoff for her opponent. For example, if the algorithm finds out that under $(\mathbf{x}^*, \mathbf{y}^*)$, the row player is guaranteed to get high payoff, it fixes the strategy $\mathbf{y}^*$ for the column player, since every action that is assigned positive probability under $\mathbf{x}^*$ is guaranteed to yield a payoff of at least $\frac{1}{2}$ for the row player. Then, it formulates the linear-feasibility system that checks if there is a strategy $\mathbf{x}'$ that restricts the row player to place probability only on actions in the support of $\mathbf{x}^*$ and at the same time bounds the maximum payoff of the column player by $\frac{1}{2}$. If the linear system has indeed a feasible solution, the algorithm returns the profile $(\mathbf{x}', \mathbf{y}^*)$, and this will be a $\frac{1}{2}$-WSNE, by the preceding discussion.
- **Case (c): high payoffs.** If the previous two cases fail, then we can prove that there exists a strategy profile $(\mathbf{x}, \mathbf{y})$, under which both players are playing only actions that yield a payoff of at least $\frac{1}{2}$, thus $(\mathbf{x}, \mathbf{y})$ is a $\frac{1}{2}$-WSNE. In order to prove this, we first define a "subgame" of (R, C) that depends on the Nash equilibria we have computed for the zero-sum games $(R, -R)$ and $(-C, C)$. Then, using the properties of Nash equilibria, and the fact that the previous two cases of the algorithm failed, we prove that in *any* Nash equilibrium $(\mathbf{x}, \mathbf{y})$ of this subgame, both players get a payoff of at least $\frac{1}{2}$. On the other hand though, it is not straightforward to find such a strategy profile. Luckily, we can show that we can approximate this by a strategy profile $(\mathbf{w}, \mathbf{z})$, where each player plays with positive probability *constantly-many* actions and every such action yields a payoff of at least $\frac{1}{2} - \delta$. Thus, we obtain a $(\frac{1}{2} + \delta)$-WSNE by exhaustively searching over the strategy profiles with constant support.

3.1 Proof of Theorem 3.1 Firstly, observe that Steps 1 and 2 can be performed in polynomial time since it is known that zero-sum games can be solved via a linear program. The remaining steps can also be performed in polynomial time, as we will argue in the sequel. In what follows we will assume that without loss of generality $\mathcal{C}(\hat{\mathbf{x}}, \hat{\mathbf{y}}) \le \mathcal{R}(\mathbf{x}^*, \mathbf{y}^*)$ and thus we will focus on the analysis of Step 3, since the analysis of Step 4 is symmetric.

Before we proceed with the analysis of Steps 3(a) - 3(c), we state the following simple observation for Nash equilibria of zero-sum games.

Observation 1. *Let $(\mathbf{x}^*, \mathbf{y}^*)$ be a Nash equilibrium of the zero-sum game $(R, -R)$. Then, for any other strategy $\mathbf{y}$ of the column player, we have:*

$$\mathcal{R}(\mathbf{x}^*, \mathbf{y}^*) \le \mathcal{R}(\mathbf{x}^*, \mathbf{y}).$$

Similarly, for an equilibrium $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ of the zero-sum game $(-C, C)$ we have that for any strategy $\mathbf{x}$ of the row player:

$$\mathcal{C}(\hat{\mathbf{x}}, \hat{\mathbf{y}}) \le \mathcal{C}(\mathbf{x}, \hat{\mathbf{y}}).$$

Proof. This is a simple implication of well known properties for 0-sum games. Let $(\mathbf{x}^*, \mathbf{y}^*)$ be a Nash equilibrium of the zero-sum game $(R, -R)$, and let v_R be the value of this game. Then we know that $v_R = \mathcal{R}(\mathbf{x}^*, \mathbf{y}^*)$. We also know that in a zero-sum game, the equilibrium strategy $\mathbf{x}^*$ for the row player is a max-min strategy, and therefore the row player can guarantee at least the value v_R by playing $\mathbf{x}^*$, independently of the column player's choice of strategy. Hence, we have that $\mathcal{R}(\mathbf{x}^*, \mathbf{y}) \ge \mathcal{R}(\mathbf{x}^*, \mathbf{y}^*)$. The second inequality is proved in the same way by applying the same argument for the column player in the zero-sum game $(-C, C)$. ◻

Now we are ready to prove the correctness of our algorithm. We will consider every case separately. For the cases 3(a) and 3(b), the analysis is rather straightforward.

Lemma 3.1. (Case 3(a)) *If $\mathcal{C}(\hat{\mathbf{x}}, \hat{\mathbf{y}}) \le \mathcal{R}(\mathbf{x}^*, \mathbf{y}^*) \le \frac{1}{2}$, then $(\hat{\mathbf{x}}, \mathbf{y}^*)$ is a $\frac{1}{2}$-WSNE for the game (R, C).*

Proof. Since $(\mathbf{x}^*, \mathbf{y}^*)$ is a Nash equilibrium of the game $(R, -R)$, we know that for every $i \in [n]$, it holds that $\mathcal{R}(e_i, \mathbf{y}^*) \le \mathcal{R}(\mathbf{x}^*, \mathbf{y}^*) \le \frac{1}{2}$. Hence, the best-response payoff of the row player against $\mathbf{y}^*$ is at most $\frac{1}{2}$, and thus, *any* strategy for the row player satisfies the constraints of $\frac{1}{2}$-WSNE, when the column player plays $\mathbf{y}^*$. Similarly, for the column player we get that for every $j \in [n]$, $\mathcal{C}(\hat{\mathbf{x}}, e_j) \le \mathcal{C}(\hat{\mathbf{x}}, \hat{\mathbf{y}}) \le \mathcal{R}(\mathbf{x}^*, \mathbf{y}^*) \le \frac{1}{2}$. So, the best-response payoff of the column player against $\hat{\mathbf{x}}$ is at most $\frac{1}{2}$, and thus, any mixed strategy for the column player satisfies the constraints of $\frac{1}{2}$-WSNE, when the row player selects $\hat{\mathbf{x}}$. ◻

Lemma 3.2. (Case 3(b)) *Suppose that $\mathcal{C}(\hat{\mathbf{x}}, \hat{\mathbf{y}}) \leq \mathcal{R}(\mathbf{x}^*, \mathbf{y}^*)$ and $\mathcal{R}(\mathbf{x}^*, \mathbf{y}^*) > \frac{1}{2}$. If there exists a strategy $\mathbf{x}'$ with $\operatorname{supp}(\mathbf{x}') \subseteq \operatorname{supp}(\mathbf{x}^*)$ such that $\mathcal{C}(\mathbf{x}', e_j) \leq \frac{1}{2}$ for any $j \in [n]$, then $(\mathbf{x}', \mathbf{y}^*)$ is a $\frac{1}{2}$-WSNE for (R, C). In addition, we can find such a strategy $\mathbf{x}'$, or decide that no such strategy exists, in polynomial time by solving a system of linear equations.*

Proof. Assume that such a strategy $\mathbf{x}'$ exists. Recall that since $\mathbf{x}^*$ is a best response against $\mathbf{y}^*$, any pure strategy in $\operatorname{supp}(\mathbf{x}^*)$ is also a best response against $\mathbf{y}^*$. Thus, any $\mathbf{x}'$ with $\operatorname{supp}(\mathbf{x}') \subseteq \operatorname{supp}(\mathbf{x}^*)$, is a best-response strategy against $\mathbf{y}^*$. Hence, the strategy profile $(\mathbf{x}', \mathbf{y}^*)$ satisfies the constraints of $\frac{1}{2}$-WSNE for the row player.

Now, for the column player, observe the following. Since the strategy $\mathbf{x}'$ guarantees that $\mathcal{C}(\mathbf{x}', e_j) \leq \frac{1}{2}$, for any $j \in [n]$, this means that the best-response payoff for the column player against $\mathbf{x}'$ is at most $\frac{1}{2}$. Thus, any mixed strategy for the column player satisfies the constraints of the $\frac{1}{2}$-WSNE, when the row player selects $\mathbf{x}'$.

Finally, it only remains to argue that we can decide in polynomial time if such a strategy exists, and if it does, to compute it efficiently. Indeed, this can be formulated as a feasibility system of linear equations. In particular, it corresponds to finding a solution for the linear system defined by:

$$\begin{aligned} \mathcal{C}(\mathbf{x}', e_j) &\leq \frac{1}{2} && \text{for every } j \in [n]; \\ \mathbf{x}'(i) &\geq 0 && \text{for every } i \in \operatorname{supp}(\mathbf{x}^*); \\ \mathbf{x}'(i) &= 0 && \text{for every } i \notin \operatorname{supp}(\mathbf{x}^*); \\ \sum_i \mathbf{x}'(i) &= 1. \end{aligned}$$

□

Next we focus on Case 3(c), whose correctness seems less intuitive compared to the other two cases. In order to argue about the correctness of the algorithm, we will focus on the subgame $(R_{\mathbf{x}^*}, C_{\mathbf{x}^*})$ of (R, C) that has size $|\operatorname{supp}(\mathbf{x}^*)| \times n$. More specifically, in $(R_{\mathbf{x}^*}, C_{\mathbf{x}^*})$ the row player will be constrained to use only pure strategies in $\operatorname{supp}(\mathbf{x}^*)$, while the column player will be allowed to choose from her complete set of pure strategies. Then, we will prove that the bimatrix game $(R_{\mathbf{x}^*}, C_{\mathbf{x}^*})$ possesses a Nash equilibrium $(\mathbf{x}, \mathbf{y})$ where both players get payoff at least $\frac{1}{2}$.

Lemma 3.3. *If Algorithm 1 reaches Case 3(c), then for every equilibrium profile $(\mathbf{x}, \mathbf{y})$ of the bimatrix game $(R_{\mathbf{x}^*}, C_{\mathbf{x}^*})$, it holds that $\mathcal{R}(\mathbf{x}, \mathbf{y}) > 1/2$ and $\mathcal{C}(\mathbf{x}, \mathbf{y}) > 1/2$.*

Proof. Let $(\mathbf{x}, \mathbf{y})$ be a Nash equilibrium of the bimatrix game $(R_{\mathbf{x}^*}, C_{\mathbf{x}^*})$. Firstly, we will prove that $\mathcal{R}(\mathbf{x}, \mathbf{y}) > 1/2$. Since the algorithm has reached Case 3(c), it must be true that $\mathcal{R}(\mathbf{x}^*, \mathbf{y}^*) > 1/2$. Thus, from Observation 1, we get that $\mathcal{R}(\mathbf{x}^*, \mathbf{y}) > 1/2$ for every possible $\mathbf{y}$. Hence, it must be true that $\mathcal{R}(\mathbf{x}, \mathbf{y}) > 1/2$ since otherwise the row player could deviate to $\mathbf{x}^*$, which is a valid strategy for the game $(R_{\mathbf{x}^*}, C_{\mathbf{x}^*})$, and increase her payoff; this would contradict the assumption that $(\mathbf{x}, \mathbf{y})$ is a Nash equilibrium.

Before we prove that $\mathcal{C}(\mathbf{x}, \mathbf{y}) > \frac{1}{2}$, we will prove an intermediate claim. Consider the zero-sum game $(-C_{\mathbf{x}^*}, C_{\mathbf{x}^*})$ and let $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ be a Nash equilibrium for this game. We argue that

$$\mathcal{C}(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) > \frac{1}{2}. \tag{3.1}$$

For the sake of contradiction, assume that $\mathcal{C}(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) \leq \frac{1}{2}$. Then, since $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ is a Nash equilibrium of the game $(-C_{\mathbf{x}^*}, C_{\mathbf{x}^*})$, we get that $\mathcal{C}(\tilde{\mathbf{x}}, e_j) \leq \frac{1}{2}$ for every $j \in [n]$. But this would mean that Algorithm 1 satisfies the conditions from Case 3(b) and contradicts the fact that the algorithm reached Case 3(c).

Now we are ready to prove that $\mathcal{C}(\mathbf{x}, \mathbf{y}) > \frac{1}{2}$. To obtain a contradiction, assume that $\mathcal{C}(\mathbf{x}, \mathbf{y}) \leq \frac{1}{2}$. From Equation (3.1) and Observation 1, we get that $\mathcal{C}(\mathbf{x}, \tilde{\mathbf{y}}) > \frac{1}{2}$. Hence the column player can deviate from $\mathbf{y}$ to $\tilde{\mathbf{y}}$ and increase her payoff. But this would contradict the assumption that $(\mathbf{x}, \mathbf{y})$ is a Nash equilibrium of the game $(R_{\mathbf{x}^*}, C_{\mathbf{x}^*})$. □

In addition to Lemma 3.3, we will need the following lemma that uses k-uniform sampled strategies for *constant* values of k, which was proved in [15].

Lemma 3.4. (Theorem 2 from [15]) *Let $(\mathbf{x}, \mathbf{y})$ be a Nash equilibrium for a bimatrix game (R, C). Then, for any constant $\delta > 0$, there exists a $\kappa(\delta)$-uniform strategy profile $(\mathbf{x}_s, \mathbf{y}_s)$, with $\kappa(\delta) = \left\lceil \frac{2\ln(1/\delta)}{\delta^2} \right\rceil$, where $\mathbf{x}_s$ is sampled from $\mathbf{x}$, and $\mathbf{y}_s$ is sampled from $\mathbf{y}$ such that:*

1. *$\mathcal{R}(e_i, \mathbf{y}_s) \geq \mathcal{R}(\mathbf{x}, \mathbf{y}) - \delta$, for every $i \in \operatorname{supp}(\mathbf{x}_s)$;*
2. *$\mathcal{C}(\mathbf{x}_s, e_j) \geq \mathcal{C}(\mathbf{x}, \mathbf{y}) - \delta$, for every $j \in \operatorname{supp}(\mathbf{y}_s)$;*

Lemma 3.4 tells us two things that are useful for our purposes. Firstly, Points 1-2 provide an approximation guarantee for the strategy profile $(\mathbf{x}_s, \mathbf{y}_s)$ as a well-supported NE. Then, the fact that both $\mathbf{x}_s$ and $\mathbf{y}_s$ are $\kappa(\delta)$-uniform tells us that we can find this profile by enumeration in polynomial time. The lemma below summarizes the above and it completes the proof of correctness of our algorithm.

Lemma 3.5. (Case 3(c)) *If Algorithm 1 reaches Case 3(c), then we can compute in polynomial time a $\kappa(\delta)$-uniform strategy profile $(\mathbf{w}, \mathbf{z})$, with $\kappa(\delta) = \left\lceil \frac{2\ln(1/\delta)}{\delta^2} \right\rceil$, that is a $(\frac{1}{2} + \delta)$-WSNE for the game (R, C).*

Proof. The proof of the lemma follows from the combination of Lemma 3.3 and Lemma 3.4. From Lemma 3.3, we get that there exists a strategy profile $(\mathbf{x}, \mathbf{y})$, that is a Nash equilibrium for the subgame $(R_{\mathbf{x}^*}, C_{\mathbf{x}^*})$, and additionally under this strategy profile both players get payoff at least $\frac{1}{2}$. Then, Lemma 3.4, applied to the profile $(\mathbf{x}, \mathbf{y})$ for the game $(R_{\mathbf{x}^*}, C_{\mathbf{x}^*})$, implies the existence of a strategy profile $(\mathbf{w}, \mathbf{z})$, such that:

- $\mathcal{R}(e_i, \mathbf{z}) \geq \frac{1}{2} - \delta$ for every $i \in \operatorname{supp}(\mathbf{w})$ and $\mathcal{C}(\mathbf{w}, e_j) \geq \frac{1}{2} - \delta$ for every $j \in \operatorname{supp}(\mathbf{z})$;
- $\operatorname{supp}(\mathbf{w}) \subseteq \operatorname{supp}(\mathbf{x}^*)$.

This profile satisfies all the constraints we want. Observe that the first bullet above implies that $(\mathbf{w}, \mathbf{z})$ is a $(\frac{1}{2} + \delta)$-WSNE for the game (R, C); this is straightforward to see since the maximum payoff is at most 1. Thus, it only remains to prove that we can compute this strategy profile in polynomial time. This is again easy to see: there are $O(n^{\kappa(\delta)}) = n^{O(\frac{\ln(1/\delta)}{\delta^2})}$ $\kappa(\delta)$-uniform strategy profiles, which is polynomially bounded, for every constant $\delta > 0$, and we can check if such a profile is a $(\frac{1}{2} + \delta)$-WSNE in polynomial time. □

4 A Query Efficient Algorithm

In this section we consider the query complexity of the problem and we show how Algorithm 1 can be converted to a query-efficient algorithm that achieves the same approximation for well-supported Nash equilibria using $O(n \cdot \log n)$ payoff queries.

Query Complexity. In the setting, we assume that the algorithm only knows that the players will play an $n \times n$ game (R, C), but it does not know any of the payoff entries of R and C. The algorithm learns the payoff entries via *payoff queries*: if the algorithm queries the pure strategy profile (e_i, e_j), it is told the payoffs R_{ij} and C_{ij} the players get under this strategy profile. After each payoff query, the algorithm can perform arbitrary computations –our algorithm will perform polynomial-time computations– in order to decide the next strategy profile to query (if any). In the end, the algorithm outputs a mixed strategy profile $(\mathbf{x}, \mathbf{y})$. The goal is to ensure that $(\mathbf{x}, \mathbf{y})$ has a good approximation guarantee, while keeping the number of queries as small as possible.

In order to prove our main result for this section, we will use the following lemma, proven by Fearnley and Savani [25], which essentially shows that with high probability we can find an ε-WSNE of a zero-sum game with $O(\frac{n \cdot \log n}{\varepsilon^4})$ payoff queries.

Lemma 4.1. *[Corollary 5.5 from [25]] Given an $n \times n$ zero-sum game, with probability at least $(1 - n^{-\frac{1}{8}})(1 - \frac{2}{n})^2$, we can compute an ε-WSNE $(\mathbf{x}, \mathbf{y})$ using $O(\frac{n \cdot \log n}{\varepsilon^4})$ payoff queries.*

While Lemma 4.1 suffices to simulate Steps 1 and 2, and Case 3(a) of Algorithm 1, there is no obvious way to implement the remaining cases in a query-efficient way. One idea would be to perform an exhaustive search under the constraint that we consider only the subgame $(R_{\mathbf{x}^*}, C_{\mathbf{x}^*})$, where $\mathbf{x}^*$ is the strategy computed at Step 1. This would not break the correctness of the algorithm, since from Lemmas 3.2 and 3.3 we know that it suffices to focus on this subgame if the algorithm reaches Cases 3(b) and 3(c). This idea would indeed allow us to simulate Case

3(b), and Case 3(c), with $O(n \cdot \log n)$ payoff queries, if $(R_{\mathbf{x}^*}, C_{\mathbf{x}^*})$ had $O(n \cdot \log n)$ payoff entries; for example, everything would work fine if $|\text{supp}(\mathbf{x}^*)| = O(\log n)$. However, Lemma 4.1 does not provide any bound on the support size of $\mathbf{x}$; in fact it may even return a fully-mixed strategy for every player. Luckily though, we can sample an $O(\log n)$-uniform strategy $\mathbf{x}_s^*$ that approximates well enough $\mathbf{x}^*$.

We describe formally below all the adjustments that need to be done to Algorithm 1.

```
Algorithm 2: Query-Efficient implementation of Algorithm 1

Input: A bimatrix game $(R, C)$ and constants $\varepsilon, \delta \in (0, 1]$, and $\kappa(\delta) = \left\lceil \frac{2\ln(1/\delta)}{\delta^2} \right\rceil$.
Output: A $(\frac{1}{2} + 3\varepsilon + \delta)$-WSNE for $(R, C)$ using $O(\frac{n \cdot \log n}{\varepsilon^4})$ payoff queries.
1. Use Lemma 4.1 to compute an $\varepsilon$-WSNE $(\mathbf{x}^*, \mathbf{y}^*)$ for the zero-sum game $(R, -R)$.
2. Use Lemma 4.1 to compute an $\varepsilon$-WSNE $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ for the zero-sum game $(-C, C)$.
3. If $\mathcal{R}(\mathbf{x}^*, \mathbf{y}^*) \geq \mathcal{C}(\hat{\mathbf{x}}, \hat{\mathbf{y}})$, use Lemma 2.1 to sample the strategy profile $(\mathbf{x}_s^*, \mathbf{y}_s^*)$.
   Query all strategy profiles $(e_i, e_j)$, with $i \in \text{supp}(\mathbf{x}_s^*)$, and create the subgame $(R_{\mathbf{x}_s^*}, C_{\mathbf{x}_s^*})$.
   (a) If $\mathcal{R}(\mathbf{x}^*, \mathbf{y}^*) \leq \frac{1}{2}$, then return $(\hat{\mathbf{x}}, \mathbf{y}^*)$.
   (b) Else if there exists a strategy $\mathbf{x}'$ such that $\text{supp}(\mathbf{x}') \subseteq \text{supp}(\mathbf{x}_s^*)$ and $\mathcal{C}(\mathbf{x}', e_j) \leq \frac{1}{2}$ for every $j \in [n]$,
       then return $(\mathbf{x}', \mathbf{y}_s^*)$.
   (c) Else, exhaustively search over $\kappa(\delta)$-uniform strategy profiles of $(R_{\mathbf{x}_s^*}, C_{\mathbf{x}_s^*})$ and find a $(\frac{1}{2} + 3\varepsilon + \delta)$-
       WSNE of $(R, C)$.
4. Else if $\mathcal{C}(\hat{\mathbf{x}}, \hat{\mathbf{y}}) > \mathcal{R}(\mathbf{x}^*, \mathbf{y}^*)$, use Lemma 2.1 to sample the strategy profile $(\hat{\mathbf{x}}_s, \hat{\mathbf{y}}_s)$.
   Query all strategy profiles $(e_i, e_j)$, with $j \in \text{supp}(\hat{\mathbf{y}}_s)$, and create the subgame $(R_{\hat{\mathbf{x}}_s}, C_{\hat{\mathbf{x}}_s})$.
   (a) If $\mathcal{C}(\hat{\mathbf{x}}, \hat{\mathbf{y}}) \leq \frac{1}{2}$, then return $(\hat{\mathbf{x}}, \mathbf{y}^*)$.
   (b) Else if there exists a strategy $\mathbf{y}'$ such that $\text{supp}(\mathbf{y}') \subseteq \text{supp}(\hat{\mathbf{y}}_s)$ and $\mathcal{R}(e_i, \mathbf{y}') \leq \frac{1}{2}$ for every $i \in [n]$,
       then return $(\hat{\mathbf{x}}_s, \mathbf{y}')$.
   (c) Else, exhaustively search over $\kappa(\delta)$-uniform strategy profiles of $(R_{\hat{\mathbf{x}}_s}, C_{\hat{\mathbf{x}}_s})$ and find a $(\frac{1}{2} + 3\varepsilon + \delta)$-
       WSNE for $(R, C)$.
```

THEOREM 4.1. *For any constants, $\varepsilon, \delta > 0$, Algorithm 2 computes in polynomial time, and with probability at least $(1 - n^{-\frac{1}{8}})(1 - \frac{2}{n})^2$, a $(\frac{1}{2} + 3\varepsilon + \delta)$-WSNE using $O(\frac{n \cdot \log n}{\varepsilon^4})$ payoff queries.*

Note that in order to compute a $(1/2 + \delta')$-WSNE for some given δ', it suffices to set the constants ε, δ of Algorithm 2, so that $\delta' = 3\varepsilon + \delta$, e.g., using $\varepsilon = \delta = \delta'/4$ suffices.

Before we prove Theorem 4.1, we state first the following simple observation, which follows directly from the definition of ε-WSNE.

OBSERVATION 2. *Fix $\varepsilon \geq 0$, and let $(\mathbf{x}^*, \mathbf{y}^*)$ be an ε-WSNE of the zero-sum game $(R, -R)$. Then, we have:*

$$\mathcal{R}(\mathbf{x}^*, \mathbf{y}^*) \geq \mathcal{R}(e_i, \mathbf{y}^*) - \varepsilon, \ \forall i \in [n], \quad \textit{and} \quad \mathcal{R}(\mathbf{x}^*, \mathbf{y}^*) - \varepsilon \leq \mathcal{R}(\mathbf{x}^*, e_j), \ \forall j \in [n]. \tag{4.2}$$

Similarly, for any ε-WSNE $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ of the zero-sum game $(-C, C)$ we have:

$$\mathcal{C}(\hat{\mathbf{x}}, \hat{\mathbf{y}}) - \varepsilon \leq \mathcal{C}(e_i, \hat{\mathbf{y}}), \ \forall i \in [n], \quad \textit{and} \quad \mathcal{C}(\hat{\mathbf{x}}, \hat{\mathbf{y}}) \geq \mathcal{C}(\hat{\mathbf{x}}, e_j) - \varepsilon, \ \forall j \in [n]. \tag{4.3}$$

Proof. [Proof of Theorem 4.1] We begin by establishing the number of payoff queries the algorithm requires. Steps 1 and 2, according to Lemma 4.1, require $O(\frac{n \cdot \log n}{\varepsilon^4})$ queries each. In addition, Steps 3 and 4 require $O(\frac{n \cdot \log n}{\varepsilon^2})$ queries each; this is because we know from Lemma 2.1 that $|\text{supp}(\mathbf{x}_s^*)| = O(\frac{\log n}{\varepsilon^2})$ and $|\text{supp}(\mathbf{x}_s^*)| = O(\frac{\log n}{\varepsilon^2})$.

In order to prove the correctness of Algorithm 2, we will follow a similar approach as in the proof of Theorem 3.1. Without loss of generality assume that $\mathcal{R}(\mathbf{x}^*, \mathbf{y}^*) \geq \mathcal{C}(\hat{\mathbf{x}}, \hat{\mathbf{y}})$, hence the algorithm will proceed according to Step 3. We will distinguish between the three possible cases.

- **Case 3(a) occurs.** Then, using exactly the same arguments as in Lemma 3.1, and by Observation 2, we get that $(\hat{\mathbf{x}}, \mathbf{y}^*)$ is a $(\frac{1}{2} + \varepsilon)$-WSNE, since $(\mathbf{x}^*, \mathbf{y}^*)$ and $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ are ε-WSNEs and not exact Nash equilibria.

- **Case 3(b) occurs.** Then, we follow the proof of Lemma 3.2 and we get a $\frac{1}{2}$-WSNE. This is because we are using the sampled strategy profile $(\mathbf{x}^*_s, \mathbf{y}^*_s)$ which, according to Lemma 2.1, is a 3ε-WSNE. Thus, we get that $\mathbf{x}'$ will satisfy the constraints of a 3ε-WSNE for the row player, while the linear system of Lemma 3.2 will satisfy the constraints of a $\frac{1}{2}$-WSNE.

- **Case 3(c) occurs.** Here, we can reprove Lemma 3.3, using the subgame $(R_{\mathbf{x}^*_s}, C_{\mathbf{x}^*_s})$ this time, and get that this subgame possesses a Nash equilibrium $(\mathbf{x}, \mathbf{y})$ such that $\mathcal{R}(\mathbf{x}, \mathbf{y}) > \frac{1}{2} - 3\varepsilon$, and $\mathcal{C}(\mathbf{x}, \mathbf{y}) > \frac{1}{2} - 3\varepsilon$; the loss of 3ε comes from the use of the sampled strategies and Lemma 2.1. Then, using verbatim the analysis of Lemma 3.5, we get that there exists a $\kappa(\delta)$-uniform strategy profile in $(R_{\mathbf{x}^*_s}, C_{\mathbf{x}^*_s})$ that is a $(\frac{1}{2}+3\varepsilon+\delta)$-WSNE for (R, C).

□

5 Discussion

We have presented a new algorithm for computing a $\frac{1}{2}$-well-supported Nash equilibrium in bimatrix games. Our algorithm not only significantly improves the previously-best approximation guarantee, but it is conceptually much simpler and intuitive. As a byproduct, we have showed how we can convert our main algorithm into a query-efficient protocol that achieves the same approximation guarantee with $O(n \cdot \log n)$ queries.

Below we identify three orthogonal directions for future work.

- Close the gap between polynomial-time upper bounds and the quasi-polynomial lower bound from [41]. While we conjecture that $1/2$ is not the limit of polynomial-time tractability for ε-WSNE, we believe that new techniques and ideas will be required for achieving a better approximation. Indeed, the shifting-probabilities ideas that were used in [24] and [14] had as a natural limit $\varepsilon = 1/2$. Hence, some fresh ideas seem to be required to go below $\frac{1}{2}$.

- Prove a matching query-complexity lower bound. In [25] it was shown that any deterministic algorithm requires $\Omega(n^2)$ payoff queries to find an ε-NE, and thus an ε-WSNE, for every $\varepsilon < 1/2$. Steps 1 and 2 from Algorithm 2 are randomized, thus the above-mentioned lower bound is not binding; in [25] it has been shown that for $\varepsilon = 1/6n$, any randomized algorithm requires $\Omega(n^2)$ queries. We conjecture that any randomized algorithm that finds an ε-WSNE, with $\varepsilon < 1/2$, requires $\Omega(n^2)$ queries.

- Improve the approximation guarantee of ε-WSNE with polylogarithmic communication. In the communication complexity model, introduced in [26], every player knows her own payoff matrix, but she does not know the entries in the payoff matrix of her opponent. The players have to follow a protocol that works in rounds and in every round the players exchange a bit of information. Algorithm 1 can be converted to a communication protocol that finds a $(\frac{1}{2} + \delta)$-WSNE with $O(n \cdot \log n)$ communication, using a similar approach as in [14]. However, there is a fundamental barrier that does not allow us to implement it with polylogarithmic communication, which is the desirable bound in communication complexity. Although we can implement almost every step of the algorithm with $O(\log^2 n)$ communication, the last step seems to inevitably require $O(n)$ communication. In particular, the Case 3(c) of the algorithm can be easily reduced to the so-called Disjointness problem, which is known to be "hard" with respect to communication [40].

Acknowledgements.

This research has been supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the "First Call for H.F.R.I. Research Projects to support faculty members and researchers and the procurement of high-cost research equipment" grant (Project Number: HFRI-FM17-3512).

References

[1] Bharat Adsul, Jugal Garg, Ruta Mehta, and Milind Sohoni. Rank-1 bimatrix games: a homeomorphism and a polynomial time algorithm. In *Proceedings of STOC*, pages 195–204, 2011.

[2] Per Austrin, Mark Braverman, and Eden Chlamtáč. Inapproximability of NP-complete variants of Nash equilibrium. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 13–25. Springer, 2011.

[3] Yakov Babichenko, Siddharth Barman, and Ron Peretz. Empirical distribution of equilibrium play and its testing application. *Math. Oper. Res.*, 42(1):15–29, 2017.

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

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

[6] Shant Boodaghians, Joshua Brakensiek, Samuel B. Hopkins, and Aviad Rubinstein. Smoothed complexity of 2-player Nash equilibria. In *Proceedings of FOCS*, pages 271–282, 2020.

[7] Hartwig Bosse, Jaroslaw Byrka, and Evangelos Markakis. New algorithms for approximate Nash equilibria in bimatrix games. *Theoretical Computer Science*, 411(1):164–173, 2010.

[8] Mark Braverman, Young Kun-Ko, and Omri Weinstein. Approximating the best Nash equilibrium in $n^{o(\log n)}$-time breaks the exponential time hypothesis. In *Proceedings of SODA*, pages 970–982. SIAM, 2015.

[9] Xi Chen, Xiaotie Deng, and Shang-Hua Teng. Sparse games are hard. In *Proceedings of WINE*, pages 262–273, 2006.

[10] Xi Chen, Xiaotie Deng, and Shang-Hua Teng. Settling the complexity of computing two-player Nash equilibria. *Journal of the ACM*, 56(3), 2009.

[11] Xi Chen, Shang-Hua Teng, and Paul Valiant. The approximation complexity of win-lose games. In *Proceedings of SODA*, volume 7, pages 159–168, 2007.

[12] Bruno Codenotti and Daniel Štefankovič. On the computational complexity of Nash equilibria for (0,1) bimatrix games. *Information Processing Letters*, 94(3):145–150, 2005.

[13] Artur Czumaj, Argyrios Deligkas, Michail Fasoulakis, John Fearnley, Marcin Jurdzinski, and Rahul Savani. Distributed methods for computing approximate equilibria. *CoRR*, abs/1512.03315, 2015.

[14] Artur Czumaj, Argyrios Deligkas, Michail Fasoulakis, John Fearnley, Marcin Jurdziński, and Rahul Savani. Distributed methods for computing approximate equilibria. *Algorithmica*, 81(3):1205–1231, 2019.

[15] Artur Czumaj, Michail Fasoulakis, and Marcin Jurdziński. Approximate well-supported Nash equilibria in symmetric bimatrix games. In *Proceedings of SAGT*, volume 8768, pages 244–254, 2014.

[16] Artur Czumaj, Michail Fasoulakis, and Marcin Jurdzinski. Zero-sum game techniques for approximate Nash equilibria. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems (AAMAS'17)*, pages 1514–1516, 2017.

[17] Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a Nash equilibrium. In *Proceedings of STOC*, pages 71–78, 2006.

[18] Constantinos Daskalakis, Aranyak Mehta, and Christos Papadimitriou. Progress in approximate Nash equilibria. In *Proceedings of EC*, pages 355–358, 2007.

[19] Constantinos Daskalakis, Aranyak Mehta, and Christos Papadimitriou. A note on approximate Nash equilibria. *Theoretical Computer Science*, 410(17):1581–1588, 2009.

[20] Argyrios Deligkas, Michail Fasoulakis, and Evangelos Markakis. A polynomial-time algorithm for 1/3-approximate Nash equilibria in bimatrix games. *Arxiv*, abs/2204.11525, 2022.

[21] Argyrios Deligkas, John Fearnley, Themistoklis Melissourgos, and Paul G. Spirakis. Approximating the existential theory of the reals. *J. Comput. Syst. Sci.*, 125:106–128, 2022.

[22] Argyrios Deligkas, John Fearnley, and Rahul Savani. Inapproximability results for constrained approximate Nash equilibria. *Inf. Comput.*, 262:40–56, 2018.

[23] Michail Fasoulakis and Evangelos Markakis. An improved quasi-polynomial algorithm for approximate well-supported Nash equilibria. In *Proceedings of the Thirty-Third Conference on Artificial Intelligence (AAAI'19)*, pages 1926–1932, 2019.

[24] John Fearnley, Paul W. Goldberg, Rahul Savani, and Troels Bjerre Sørensen. Approximate well-supported Nash equilibria below two-thirds. *Algorithmica*, 76(2):297–319, 2016.

[25] John Fearnley and Rahul Savani. Finding approximate Nash equilibria of bimatrix games via payoff queries. *ACM Transactions on Economics and Computation (TEAC)*, 4(4):1–19, 2016.

[26] Paul W Goldberg and Arnoud Pastink. On the communication complexity of approximate Nash equilibria. *Games and Economic Behavior*, 85:19–31, 2014.

[27] Ravi Kannan and Thorsten Theobald. Games of fixed rank: A hierarchy of bimatrix games. *Economic Theory*, 42(1):157–173, 2010.

[28] Spyros C. Kontogiannis, Panagiota N. Panagopoulou, and Paul G. Spirakis. Polynomial algorithms for approximating Nash equilibria of bimatrix games. In *Proceedings of WINE*, pages 286–296, 2006.

[29] Spyros C. Kontogiannis and Paul G. Spirakis. Well supported approximate equilibria in bimatrix games. *Algorithmica*, 57(4):653–667, 2010.

[30] Spyros C. Kontogiannis and Paul G. Spirakis. Approximability of symmetric bimatrix games and related experiments. In *Proceedings of SEA*, volume 6630, pages 1–20, 2011.

[31] Pravesh K. Kothari and Ruta Mehta. Sum-of-squares meets Nash: lower bounds for finding any equilibrium. In *Proceedings of STOC*, pages 1241–1248. ACM, 2018.

[32] Richard Lipton, Evangelos Markakis, and Aranyak Mehta. Playing large games using simple strategies. In *Proceedings of EC*, pages 36–41, 2003.

[33] Zhengyang Liu and Ying Sheng. On the approximation of Nash equilibria in sparse win-lose games. In *Proceedings of AAAI*, volume 32, 2018.

[34] Andrew McLennan and Rabee Tourky. Imitation games and computation. *Games and Economic Behavior*, 70(1):4–11, 2010.

[35] Andrew McLennan and Rabee Tourky. Simple complexity from imitation games. *Games and Economic Behavior*, 68(2):683–688, 2010.

[36] Ruta Mehta. Constant rank two-player games are PPAD-hard. *SIAM J. Comput.*, 47(5):1858–1887, 2018.

[37] Aniket Murhekar and Ruta Mehta. Approximate Nash equilibria of imitation games: Algorithms and complexity. In *Proceedings of AAMAS*, pages 887–894, 2020.

[38] John Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, 1951.

[39] Panagiota N. Panagopoulou and Paul G. Spirakis. Random bimatrix games are asymptotically easy to solve (a simple proof). *Theory of Computing Systems*, 54(3):479–490, 2014.

[40] Tim Roughgarden et al. Communication complexity (for algorithm designers). *Foundations and Trends® in Theoretical Computer Science*, 11(3–4):217–404, 2016.

[41] Aviad Rubinstein. Settling the complexity of computing approximate two-player Nash equilibria. In *Proceedings of FOCS*, pages 258–265, 2016.

[42] Haralambos Tsaknakis and Paul G. Spirakis. An optimization approach for approximate Nash equilibria. *Internet Mathematics*, 5(4):365–382, 2008.