

A Polynomial-Time Algorithm for 1/3-Approximate Nash Equilibria in Bimatrix Games

Argyrios Deligkas ✉

Royal Holloway, University of London, Egham, UK

Michail Fasoulakis ✉

Foundation for Research and Technology-Hellas (FORTH), Heraklion, Greece
Athens University of Economics and Business, Greece

Evangelos Markakis ✉

Athens University of Economics and Business, Greece

Abstract

Since the celebrated PPAD-completeness result for Nash equilibria in bimatrix games, a long line of research has focused on polynomial-time algorithms that compute ε -approximate Nash equilibria. Finding the best possible approximation guarantee that we can have in polynomial time has been a fundamental and non-trivial pursuit on settling the complexity of approximate equilibria. Despite a significant amount of effort, the algorithm of Tsaknakis and Spirakis [36], with an approximation guarantee of $(0.3393 + \delta)$, remains the state of the art over the last 15 years. In this paper, we propose a new refinement of the Tsaknakis-Spirakis algorithm, resulting in a polynomial-time algorithm that computes a $(\frac{1}{3} + \delta)$ -Nash equilibrium, for any constant $\delta > 0$. The main idea of our approach is to go beyond the use of convex combinations of primal and dual strategies, as defined in the optimization framework of [36], and enrich the pool of strategies from which we build the strategy profiles that we output in certain bottleneck cases of the algorithm.

2012 ACM Subject Classification Theory of computation → Exact and approximate computation of equilibria

Keywords and phrases bimatrix games, approximate Nash equilibria

Digital Object Identifier 10.4230/LIPIcs.ESA.2022.41

Related Version *Full Version*: <https://arxiv.org/abs/2204.11525>

Funding *Evangelos Markakis*: This work 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).

Acknowledgements The authors would like to thank Marcin Jurdziński for some initial discussions on the Tsaknakis-Spirakis algorithm.

1 Introduction

The notion of *Nash equilibrium* has been undoubtedly a fundamental solution concept in strategic games, ever since the seminal result of Nash [33], on the existence of equilibria for all finite games. Nash’s theorem however is only *existential*; it only shows that such an equilibrium always exists, but it does not provide an efficient algorithm to find one. In fact, many years after the work of Nash, in a series of breakthrough results, it was proven that computing a Nash equilibrium is PPAD-complete [16], even for *bimatrix* games [10], which provides strong evidence that computing an equilibrium is an intractable problem.

These negative results have naturally led to the study of *approximate* Nash equilibria. In an ε -approximate Nash equilibrium (ε -NE), no player can increase her payoff more than ε , by unilaterally changing her strategy. In contrast to exact Nash equilibria, the relaxation



© Argyrios Deligkas, Michail Fasoulakis, and Evangelos Markakis;
licensed under Creative Commons License CC-BY 4.0

30th Annual European Symposium on Algorithms (ESA 2022).

Editors: Shiri Chechik, Gonzalo Navarro, Eva Rotenberg, and Grzegorz Herman; Article No. 41; pp. 41:1–41:14

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

to ϵ -NE does admit subexponential algorithms. More precisely, the quasi polynomial-time approximation scheme (QPTAS) of [27] can find an ϵ -NE in time $n^{O(\log n/\epsilon^2)}$, for a game with n available pure strategies per player. One can then wonder whether the QPTAS could be improved to a PTAS or even a FPTAS. Unfortunately this does not seem to be the case, as the result of Chen, Deng, and Teng [10] already ruled out the existence of an FPTAS, unless $\text{PPAD}=\text{P}$. Some years later, in another breakthrough result, Rubinfeld [35] showed that, assuming the exponential-time hypothesis for PPAD, there exists a very small, yet unspecified, constant ϵ^* such that finding an ϵ -NE requires quasi polynomial time for every constant $\epsilon < \epsilon^*$. This would rule out a PTAS too.

Although it seems unlikely to have a polynomial time algorithm for any $\epsilon > 0$, it is still important to identify the best constant ϵ for which we can have an efficient algorithm. In fact, this has been one of the fundamental questions of algorithmic game theory, that is still unresolved. Soon after the initial PPAD-hardness results of [10, 16], there was a flourish of works along this direction. Kontogiannis, Panagopoulou, and Spirakis [23] derived a polynomial-time algorithm for $\epsilon = 3/4$; Daskalakis, Mehta, and Papadimitriou [17, 18] improved it to $\epsilon = 1/2$ and $\epsilon \approx 0.382$; Bosse, Byrka, and Markakis [7] achieved $\epsilon = 0.364$; and finally Tsaknakis and Spirakis [36] attained a bound of $\epsilon = 0.3393 + \delta$, for any constant $\delta > 0$. Ever since this last work however, the progress on this front has stalled, and the result of Tsaknakis and Spirakis (referred to as the TS algorithm from now on) remains the state of the art over the last 15 years. It is particularly puzzling that so far, it has remained an open problem to even improve the approximation to $1/3 + \delta$ (even though it has been conjectured that such an approximation should be feasible). To make things worse, in the very recent work of [12], it was shown that the TS algorithm and its analysis are tight.

In order to beat the 0.3393-guarantee of the TS algorithm, it is instructive to understand first its bottleneck cases. At a high level, we can think of the algorithm as consisting of two phases: the *Descent phase* and the *Strategy-construction phase*. In the Descent phase, it performs “gradient descent” on the maximum regret among the two players, i.e., the maximum additional gain that a player can have by a unilateral deviation to another strategy. This process terminates at an approximate “stationary” point, i.e., a strategy profile such that any local change does not decrease the value of the maximum regret. When we reach a δ -stationary point for some small constant δ , the Strategy-construction phase begins. This phase performs a case analysis, based on certain relevant parameters of the game, and tries to decide which strategy profile to output in each of the five cases that arise.

In doing so, the algorithm has at its disposal the δ -stationary profile, along with a “dual” strategy profile (produced by solving the dual of the linear program used in the Descent phase). A close inspection reveals that one of these two profiles suffices to guarantee a $(\frac{1}{3} + \delta)$ -NE in three out of the five cases. In the remaining two cases, the algorithm outputs a convex combination of the stationary and the dual strategies, and this is where the bottleneck occurs, causing the algorithm to output a $(0.3393 + \delta)$ -NE.

Our contribution

We improve upon the state of the art and provide a polynomial-time algorithm for computing a $(\frac{1}{3} + \delta)$ -NE in bimatrix games, for any constant $\delta > 0$. More specifically, we modify sufficiently the TS algorithm by designing an *improved* Strategy-construction phase to handle the problematic cases of TS. Our main insights in doing so are as follows:

- Apart from convex combinations between primal (stationary) and dual strategies, we also consider best response strategies to such convex combinations. Hence, we enrich the pool of strategies, out of which we choose the profile to output in each case. As a result,

in the cases where the δ -stationary point or the dual profile (or their combinations) do not have the desired guarantee, we have one of the players use a carefully chosen convex combination between our newly defined strategies and her dual strategy.

- We produce a more refined case analysis, that is based on the values of some new auxiliary parameters (e.g., the quantities v_r, t_r and $\hat{\mu}$, defined in Section 4). These parameters encode payoff differences or regrets of the players for using specific strategies, and they help us in two ways. First, they are used to obtain improved upper bounds on the maximum regret of the δ -stationary profile (Section 4.1). Secondly, their values greatly help us in decomposing our analysis into convenient subcases in order to establish the approximation guarantee.

Further related work

A different notion of approximation of NE is that of ε -well-supported NE (ε -WSNE). In an ε -WSNE every player is required to place positive probability only to actions that are within ε of being best responses. Hence, ε -WSNE are more constrained than ε -NE, where the players can place a positive probability on any strategy. After a series of papers on the topic [24, 21], the currently best approximation is for $\varepsilon = 0.6528$ due to [14].

Another line of research has focused on more structured classes of bimatrix games such as: constant-rank games, where the matrix defined by the sum of the two payoff matrices has constant rank [1, 22, 31]; win-lose games, where the payoff for every pure action is either 0 or 1 [11, 13, 28]; sparse games, where there are only “a few” outcomes that yield a non-zero payoff for each player [9], imitation games, where the payoff matrix for one of the players is the identity matrix [29, 30, 32]; random games, where the payoff entries are drawn from certain distributions [4, 34]; symmetric games, where the payoff matrix of one player is the transpose of the other [15, 25]. In most of these classes, it has been possible to obtain improved approximation guarantees and have a better understanding of how to construct approximate equilibria.

Concerning quasi-polynomial algorithms, in addition to the QPTAS of [27], three new QPTASs have been obtained, which contain the original result of [27] as a special case: [5] gave a refined, parameterized, approximation scheme; [3] gave a QPTAS that can be applied to multi-player games as well; [19] gave a more general approach for approximation schemes for the existential theory of the reals. More recently, more negative results for ε -NE were derived: [26] gave an unconditional lower bound, based on the sum of squares hierarchy; [6] proved PPAD-hardness in the smoothed analysis setting; [8, 20, 2] gave quasi-polynomial time lower bounds for constrained ε -NE, under the exponential time hypothesis.

2 Preliminaries

In what follows, let $[n] := \{1, 2, \dots, n\}$, and let Δ^n denote the $(n - 1)$ -dimensional simplex. We focus on $n \times n$ bimatrix games, where n denotes the number of available pure strategies per player. Such games are defined by a pair $(R, C) \in [0, 1]^{n \times n}$ of two matrices: R and C are the payoff matrices for the *row* player and the *column* player respectively. We follow the usual assumption in the relevant literature that the matrices are normalized, so that all entries are in $[0, 1]$. It is also assumed without loss of generality, that both players have the same number of pure strategies, since otherwise one can add dummy strategies to equalize the rows and columns. The semantics of the payoff matrices are that 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.

41:4 A Polynomial-Time Algorithm for 1/3-NE in Bimatrix Games

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 x_i to denote the probability the player assigns to the pure strategy i . For the column player, we use $\mathbf{y} \in \Delta^n$ and y_i , respectively. If \mathbf{x} and \mathbf{y} are mixed strategies for the row and the column player, then we call (\mathbf{x}, \mathbf{y}) a (*mixed*) *strategy profile*. It is often also convenient to represent pure strategies as vectors. Hence, we will use the vector e_i , which has 1 at index i and zero elsewhere, to denote the i -th pure strategy, in other words the distribution where a player assigns probability one to play the pure strategy i .

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 $\mathcal{C}(\mathbf{x}, \mathbf{y}) := \mathbf{x}^T C \mathbf{y}$. Thus, 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} . We say that a pure strategy is a *best-response strategy* for a player if it maximizes her expected payoff against a chosen strategy of her opponent. So, under a strategy profile (\mathbf{x}, \mathbf{y}) , the set of pure best responses for the row player is $B_r(\mathbf{y}) := \{i \in [n] : \mathcal{R}(e_i, \mathbf{y}) = \max_{i' \in [n]} \mathcal{R}(e_{i'}, \mathbf{y})\}$, and for the column player, it is $B_c(\mathbf{x}) := \{j \in [n] : \mathcal{C}(\mathbf{x}, e_j) = \max_{j' \in [n]} \mathcal{C}(\mathbf{x}, e_{j'})\}$.

The *regret* of the row player at a profile (\mathbf{x}, \mathbf{y}) , is $\text{reg}_r(\mathbf{x}, \mathbf{y}) = \max_i \mathcal{R}(e_i, \mathbf{y}) - \mathcal{R}(\mathbf{x}, \mathbf{y})$ and the regret of the column player is $\text{reg}_c(\mathbf{x}, \mathbf{y}) = \max_j \mathcal{C}(\mathbf{x}, e_j) - \mathcal{C}(\mathbf{x}, \mathbf{y})$. The strategy profile (\mathbf{x}, \mathbf{y}) is an ε -*Nash equilibrium*, or ε -NE, if the regret of both players is bounded by $\varepsilon \in [0, 1]$, formally $\max\{\text{reg}_r(\mathbf{x}, \mathbf{y}), \text{reg}_c(\mathbf{x}, \mathbf{y})\} \leq \varepsilon$. If $\varepsilon = 0$, then the strategy profile (\mathbf{x}, \mathbf{y}) is an exact Nash equilibrium.

3 The Tsaknakis-Spirakis algorithm

In this section we give a description of the algorithm by [36] and we highlight the bottleneck cases, where it fails to provide a $(\frac{1}{3} + \delta)$ -approximation. In order to have a self-contained exposition, we also present some of the lemmas that are used in the analysis of [36], which are needed for our work as well.

The core of the algorithm is to consider the function $g(\mathbf{x}, \mathbf{y}) = \max\{\text{reg}_r(\mathbf{x}, \mathbf{y}), \text{reg}_c(\mathbf{x}, \mathbf{y})\}$, i.e., the maximum regret among the two players. Clearly, if we arrive at a profile (\mathbf{x}, \mathbf{y}) such that $g(\mathbf{x}, \mathbf{y}) \leq \varepsilon$, then (\mathbf{x}, \mathbf{y}) is an ε -Nash equilibrium. At a high level, one can think of TS as consisting of two phases: the *Descent* phase, and the *Strategy-construction* phase.

Descent Phase. During this phase, TS performs “gradient descent” on the function $g(\mathbf{x}, \mathbf{y})$, until it reaches a “*stationary point*”, i.e., a strategy profile such that any local change does not decrease the value of g . More concretely, every iteration of the Descent phase performs a series of steps: given the current profile under consideration, it equalizes the regrets of the players, then it solves an appropriate linear program to identify a feasible direction, and finally depending on the solution of the LP, it either updates the strategy profile, or it decides that it has reached an approximate stationary point.

The first step runs the RegretEqualization procedure described below. This procedure is based on solving a single linear program to equalize the regrets of the two players, and most importantly, it guarantees that the maximum regret does not increase.

RegretEqualization($\mathbf{x}_0, \mathbf{y}_0$)

Input: A strategy profile $(\mathbf{x}_0, \mathbf{y}_0)$.

Output: A strategy profile (\mathbf{x}, \mathbf{y}) such that $\text{reg}_r(\mathbf{x}, \mathbf{y}) = \text{reg}_c(\mathbf{x}, \mathbf{y}) \leq g(\mathbf{x}_0, \mathbf{y}_0)$.

1. If $\text{reg}_r(\mathbf{x}_0, \mathbf{y}_0) \geq \text{reg}_c(\mathbf{x}_0, \mathbf{y}_0)$, keep \mathbf{y}_0 fixed and solve the following linear program:

Minimize $\text{reg}_r(\mathbf{x}, \mathbf{y}_0)$

Such that $\text{reg}_r(\mathbf{x}, \mathbf{y}_0) \geq \text{reg}_c(\mathbf{x}, \mathbf{y}_0)$ and $\mathbf{x} \in \Delta^n$.

Return $(\mathbf{x}, \mathbf{y}_0)$, where \mathbf{x} is the solution of the linear program.

2. If $\text{reg}_r(\mathbf{x}_0, \mathbf{y}_0) < \text{reg}_c(\mathbf{x}_0, \mathbf{y}_0)$, keep \mathbf{x}_0 fixed and solve the following linear program:

Minimize $\text{reg}_c(\mathbf{x}_0, \mathbf{y})$

Such that $\text{reg}_c(\mathbf{x}_0, \mathbf{y}) \geq \text{reg}_r(\mathbf{x}_0, \mathbf{y})$ and $\mathbf{y} \in \Delta^n$.

Return $(\mathbf{x}_0, \mathbf{y})$, where \mathbf{y} is the solution of the linear program.

Given the output (\mathbf{x}, \mathbf{y}) of RegretEqualization, the next step is to either find a feasible direction to follow so as to decrease the maximum regret, or to decide that (\mathbf{x}, \mathbf{y}) is an approximate stationary point. This is enforced by solving the following linear program.

Primal Linear Program: Primal(\mathbf{x}, \mathbf{y})

minimize γ

s.t. $\gamma \geq \mathcal{R}(e_i, \mathbf{y}') - \mathcal{R}(\mathbf{x}, \mathbf{y}') - \mathcal{R}(\mathbf{x}', \mathbf{y}) + \mathcal{R}(\mathbf{x}, \mathbf{y}), \quad \forall i \in B_r(\mathbf{y}),$

$\gamma \geq \mathcal{C}(\mathbf{x}', e_j) - \mathcal{C}(\mathbf{x}', \mathbf{y}) - \mathcal{C}(\mathbf{x}, \mathbf{y}') + \mathcal{C}(\mathbf{x}, \mathbf{y}), \quad \forall j \in B_c(\mathbf{x}),$

$\mathbf{x}' \in \Delta^n, \quad \mathbf{y}' \in \Delta^n.$

It is proved in [36] that the solution of Primal(\mathbf{x}, \mathbf{y}) guarantees one of the following:

1. it either identifies a strategy profile $(\mathbf{x}', \mathbf{y}')$ such that the maximum regret can be strictly decreased by a constant fraction, if we move from (\mathbf{x}, \mathbf{y}) towards $(\mathbf{x}', \mathbf{y}')$;
2. or it decides that (\mathbf{x}, \mathbf{y}) is a δ -stationary point¹, which is the termination criterion of the descent.

Putting everything together, the Descent phase of the TS algorithm is described below, starting from some arbitrary initial strategy profile, and its main properties are captured by the following lemma.

► **Lemma 1** ([36]). *For any constant $\delta > 0$, the Descent phase computes a δ -stationary point, in time polynomial in $1/\delta$ and in the size of the game.*

Descent Phase

Input: A strategy profile (\mathbf{x}, \mathbf{y}) , a small constant $\delta > 0$ ($\delta \ll 1/3$).

Output: A δ -stationary profile $(\mathbf{x}_s, \mathbf{y}_s)$ with equal regrets.

1. Equalize the regrets of the two players, i.e., set $(\mathbf{x}, \mathbf{y}) \leftarrow \text{RegretEqualization}(\mathbf{x}, \mathbf{y})$.

2. Solve Primal(\mathbf{x}, \mathbf{y}) and compute \mathbf{x}', \mathbf{y}' , and γ .

3. If $\gamma - g(\mathbf{x}, \mathbf{y}) \geq -\delta$, set $\mathbf{x}_s \leftarrow \mathbf{x}, \mathbf{y}_s \leftarrow \mathbf{y}$ and stop.

4. Else, set $\mathbf{x} \leftarrow (1 - \frac{\delta}{\delta+2}) \cdot \mathbf{x} + \frac{\delta}{\delta+2} \cdot \mathbf{x}'$, $\mathbf{y} \leftarrow (1 - \frac{\delta}{\delta+2}) \cdot \mathbf{y} + \frac{\delta}{\delta+2} \cdot \mathbf{y}'$ and go to Step 1.

¹ This means that the directional derivative of $g(\mathbf{x}, \mathbf{y})$ is at least $-\delta$. For the definition of directional derivative, see [36].

Strategy-construction Phase. In this phase, the algorithm utilizes the dual linear program of Primal(\mathbf{x}, \mathbf{y}), in order to identify some alternative candidate strategies for the players.

Dual Linear Program: Dual(\mathbf{x}, \mathbf{y})

$$\begin{aligned}
 & \text{maximize} && P \cdot \mathcal{R}(\mathbf{x}, \mathbf{y}) + Q \cdot \mathcal{C}(\mathbf{x}, \mathbf{y}) + a + b \\
 & \text{s.t.} && p_i \geq 0, \quad i \in B_r(\mathbf{y}), \\
 & && q_j \geq 0, \quad j \in B_c(\mathbf{x}), \\
 & && P = \sum_{i \in B_r(\mathbf{y})} p_i, \quad Q = \sum_{j \in B_c(\mathbf{x})} q_j, \\
 & && P + Q = 1, \\
 & && a \leq \sum_{i \in B_r(\mathbf{y})} -\mathcal{R}(e_k, \mathbf{y}) \cdot p_i + \sum_{j \in B_c(\mathbf{x})} [-\mathcal{C}(e_k, \mathbf{y}) + C_{kj}] \cdot q_j, \quad 1 \leq k \leq n \\
 & && b \leq \sum_{j \in B_c(\mathbf{x})} -\mathcal{C}(\mathbf{x}, e_l) \cdot q_j + \sum_{i \in B_r(\mathbf{y})} [-\mathcal{R}(\mathbf{x}, e_l) + R_{il}] \cdot p_i, \quad 1 \leq l \leq n.
 \end{aligned}$$

Given the δ -stationary profile $(\mathbf{x}_s, \mathbf{y}_s)$ from the Descent phase, the algorithm solves Dual($\mathbf{x}_s, \mathbf{y}_s$) and computes the following (from the optimal dual variables).

- The dual strategy \mathbf{w} for the row player, where $w_i = p_i / \sum_{j \in B_r(\mathbf{y}_s)} p_j$, for $i \in B_r(\mathbf{y}_s)$, and 0 elsewhere; note that by construction, \mathbf{w} is a best-response strategy against \mathbf{y}_s .
- The dual strategy \mathbf{z} for the column player, where $z_i = q_i / \sum_{j \in B_c(\mathbf{x}_s)} q_j$, for $i \in B_r(\mathbf{x}_s)$, and 0 elsewhere; by construction, \mathbf{z} is a best-response strategy against \mathbf{x}_s .
- The dual variables $P, Q \in [0, 1]$, that are useful for the approximation analysis.

In addition, we define the following two quantities λ and μ , that help in parameterizing the maximum regret bound. These quantities are equal to the payoff difference of a player between the dual and the primal strategies, when the other player uses her dual strategy:

$$\lambda = \mathcal{R}(\mathbf{w}, \mathbf{z}) - \mathcal{R}(\mathbf{x}_s, \mathbf{z}), \quad \mu = \mathcal{C}(\mathbf{w}, \mathbf{z}) - \mathcal{C}(\mathbf{w}, \mathbf{y}_s). \quad (1)$$

Fact. Obviously, $\lambda \leq 1$, and $\mu \leq 1$ and furthermore, $\mathcal{R}(\mathbf{w}, \mathbf{z}) \geq \lambda$, and $\mathcal{C}(\mathbf{w}, \mathbf{z}) \geq \mu$. The algorithm then constructs and outputs a strategy profile as follows.

Strategy-Construction Phase

Input: A δ -stationary strategy profile $(\mathbf{x}_s, \mathbf{y}_s)$ from the Descent phase, the dual strategies \mathbf{w}, \mathbf{z} , and the parameters λ, μ .

1. If $\min\{\lambda, \mu\} \leq \frac{1}{2}$, then return $(\mathbf{x}_s, \mathbf{y}_s)$.
2. If $\min\{\lambda, \mu\} \geq \frac{2}{3}$, then return (\mathbf{w}, \mathbf{z}) .
3. If $\min\{\lambda, \mu\} > \frac{1}{2}$ and $\max\{\lambda, \mu\} \leq \frac{2}{3}$, then return $(\mathbf{x}_s, \mathbf{y}_s)$.
4. Else if $\lambda \geq \mu$, then return the strategy profile with the minimum regret between $\left(\frac{1}{1+\lambda-\mu} \cdot \mathbf{w} + \frac{\lambda-\mu}{1+\lambda-\mu} \cdot \mathbf{x}_s, \mathbf{z}\right)$ and $(\mathbf{x}_s, \mathbf{y}_s)$.
5. Else if $\lambda < \mu$, then return the strategy profile with the minimum regret between $\left(\mathbf{w}, \frac{1}{1+\mu-\lambda} \cdot \mathbf{z} + \frac{\mu-\lambda}{1+\mu-\lambda} \cdot \mathbf{y}_s\right)$ and $(\mathbf{x}_s, \mathbf{y}_s)$.

► **Theorem 2** ([36]). *For any constant $\delta > 0$, the TS algorithm computes in polynomial time a $(0.3393 + \delta)$ -NE.*

► **Remark 3.** One could also check all the proposed profiles of this phase at every iteration of the Descent phase, as presented in [36], and stop if we have reached already the desired approximation. But this does not affect the worst-case running time, which occurs when the Descent phase terminates at a δ -stationary point.

We present below some important lemmas from [36] that are needed in our analysis too.

The first, and most important, lemma below shows how $\text{Primal}(\mathbf{x}_s, \mathbf{y}_s)$ and $\text{Dual}(\mathbf{x}_s, \mathbf{y}_s)$ can be used to bound the value of the maximum regret, $g(\mathbf{x}_s, \mathbf{y}_s)$.

► **Lemma 4** (implied by [36]). *Let $(\mathbf{x}_s, \mathbf{y}_s)$ be a δ -stationary point produced by the Descent Phase, for a constant $\delta > 0$. Let also \mathbf{w}, \mathbf{z} and P , be derived by an optimal solution to $\text{Dual}(\mathbf{x}_s, \mathbf{y}_s)$, as seen before. Then, for any strategy profile $(\mathbf{x}', \mathbf{y}')$, it holds that*

$$g(\mathbf{x}_s, \mathbf{y}_s) \leq P \cdot (\mathcal{R}(\mathbf{w}, \mathbf{y}') - \mathcal{R}(\mathbf{x}', \mathbf{y}_s) - \mathcal{R}(\mathbf{x}_s, \mathbf{y}') + \mathcal{R}(\mathbf{x}_s, \mathbf{y}_s)) + (1 - P) \cdot (\mathcal{C}(\mathbf{x}', \mathbf{z}) - \mathcal{C}(\mathbf{x}', \mathbf{y}_s) - \mathcal{C}(\mathbf{x}_s, \mathbf{y}') + \mathcal{C}(\mathbf{x}_s, \mathbf{y}_s)) + \delta.$$

Lemma 4 plays a crucial role as it allows us to bound $g(\mathbf{x}_s, \mathbf{y}_s)$ in terms of λ, μ and P , by making appropriate choices for \mathbf{x}' and \mathbf{y}' . This is used both in the following lemma and in Lemma 11 of Section 4.

► **Lemma 5** ([36]). *Let $(\mathbf{x}_s, \mathbf{y}_s)$ be a δ -stationary point produced by the Descent phase, for a constant $\delta > 0$, and let P be obtained by an optimal solution of $\text{Dual}(\mathbf{x}_s, \mathbf{y}_s)$. It holds that $g(\mathbf{x}_s, \mathbf{y}_s) \leq \min\{P \cdot \lambda, (1 - P) \cdot \mu\} + \delta \leq \frac{\lambda \cdot \mu}{\lambda + \mu} + \delta \leq \frac{\lambda + \mu}{4} + \delta$.*

One may worry that the bound $\frac{\lambda \cdot \mu}{\lambda + \mu}$ is not well-defined when $\lambda + \mu = 0$. However, as we explain below, this is not a concern.

► **Corollary 6.** *We can assume that both $\lambda > 0$ and $\mu > 0$, otherwise $(\mathbf{x}_s, \mathbf{y}_s)$ is a δ -Nash equilibrium.*

The definitions of λ and μ , along with Lemma 5 can immediately be used to prove that Cases 1-3 from the Strategy-construction phase return a $(\frac{1}{3} + \delta)$ -Nash equilibrium. Hence, the bottleneck of the TS algorithm comes from Cases 4 and 5. In fact, it was also recently shown in [12] that the analysis of these cases in [36] is tight, and therefore one needs to come up with a different construction in order to obtain an improvement.

► **Lemma 7** ([36]). *Cases 1-3 from the Strategy-construction phase return a $(\frac{1}{3} + \delta)$ -Nash equilibrium.*

Proof. We will consider every case independently.

- If $\min\{\lambda, \mu\} \leq \frac{1}{2}$, by Lemma 5 we have that $g(\mathbf{x}_s, \mathbf{y}_s) \leq \frac{\lambda \cdot \mu}{\lambda + \mu} + \delta \leq \frac{\min\{\lambda, \mu\}}{\min\{\lambda, \mu\} + 1} + \delta \leq \frac{1/2}{1/2 + 1} + \delta \leq \frac{1}{3} + \delta$. Here, the second inequality comes from the fact that $\frac{\lambda \cdot \mu}{\lambda + \mu}$ is an increasing function of $\max\{\lambda, \mu\}$, and also $\max\{\lambda, \mu\} \leq 1$.
- If $\min\{\lambda, \mu\} \geq \frac{2}{3}$, then $g(\mathbf{w}, \mathbf{z}) \leq \max\{1 - \mathcal{R}(\mathbf{w}, \mathbf{z}), 1 - \mathcal{C}(\mathbf{w}, \mathbf{z})\}$. But since $\mathcal{R}(\mathbf{w}, \mathbf{z}) \geq \lambda$ and $\mathcal{C}(\mathbf{w}, \mathbf{z}) \geq \mu$, we have that the regret is at most $1 - \min\{\lambda, \mu\} \leq \frac{1}{3}$.
- If $\min\{\lambda, \mu\} > \frac{1}{2}$ and $\max\{\lambda, \mu\} \leq \frac{2}{3}$, by Lemma 5 we have $g(\mathbf{x}_s, \mathbf{y}_s) \leq \frac{\lambda \cdot \mu}{\lambda + \mu} + \delta \leq \frac{\frac{2}{3} \cdot \frac{2}{3}}{\frac{2}{3} + \frac{2}{3}} + \delta \leq \frac{1}{3} + \delta$, since $\frac{\lambda \cdot \mu}{\lambda + \mu}$ is an increasing function of λ and μ . ◀

Thus, in the next section, we will focus on the remaining cases, when $\min\{\lambda, \mu\} \in (\frac{1}{2}, \frac{2}{3}]$ and $\max\{\lambda, \mu\} \in (\frac{2}{3}, 1]$.

4 Improved Strategy-construction Phase

In this section we replace Cases 4 and 5 from the original TS algorithm in order to bypass the bottleneck in the approximation. To do so, we utilize the δ -stationary point $(\mathbf{x}_s, \mathbf{y}_s)$, the dual strategies \mathbf{w}, \mathbf{z} , their convex combinations and best-response strategies to such

combinations. We then perform a more refined analysis and prove that in every case we can efficiently construct a tailored strategy profile that is a $(\frac{1}{3} + \delta)$ -Nash equilibrium. We note that all missing proofs from this section can be found in the full version of our work.

Our new Strategy-construction phase works as follows.

Improved Strategy-construction Phase

Input: A δ -stationary point $(\mathbf{x}_s, \mathbf{y}_s)$ from the Descent phase, the dual strategies \mathbf{w} , \mathbf{z} , and the parameters λ, μ .

1. If $\min\{\lambda, \mu\} \leq \frac{1}{2}$, then return $(\mathbf{x}_s, \mathbf{y}_s)$.
2. If $\min\{\lambda, \mu\} \geq \frac{2}{3}$, then return (\mathbf{w}, \mathbf{z}) .
3. If $\min\{\lambda, \mu\} > \frac{1}{2}$ and $\max\{\lambda, \mu\} \leq \frac{2}{3}$, then return $(\mathbf{x}_s, \mathbf{y}_s)$.
4. If $\frac{1}{2} < \lambda \leq \frac{2}{3} < \mu$:
 - Set $\hat{\mathbf{y}} = \frac{1}{2} \cdot \mathbf{y}_s + \frac{1}{2} \cdot \mathbf{z}$.
 - Find a best response $\hat{\mathbf{w}}$ against $\hat{\mathbf{y}}$.
 - Set $t_r = \mathcal{R}(\hat{\mathbf{w}}, \hat{\mathbf{y}}) - \mathcal{R}(\mathbf{w}, \hat{\mathbf{y}})$; $v_r = \mathcal{R}(\mathbf{w}, \mathbf{y}_s) - \mathcal{R}(\hat{\mathbf{w}}, \mathbf{y}_s)$; $\hat{\mu} = \mathcal{C}(\hat{\mathbf{w}}, \mathbf{z}) - \mathcal{C}(\hat{\mathbf{w}}, \mathbf{y}_s)$.
 - 4.1 If $v_r + t_r \geq \frac{\mu - \lambda}{2}$ and $\hat{\mu} \geq \mu - v_r - t_r$, then set $p = \frac{2 \cdot (v_r + t_r) - (\mu - \lambda)}{2 \cdot (v_r + t_r)}$ and return the strategy profile with the minimum regret among $(p \cdot \mathbf{w} + (1 - p) \cdot \hat{\mathbf{w}}, \mathbf{z})$ and $(\mathbf{x}_s, \mathbf{y}_s)$.
 - 4.2 Else, set $q = \frac{1 - \mu/2 - t_r}{1 + \mu/2 - \lambda - t_r}$ and return the strategy profile with the minimum regret among $(\mathbf{w}, (1 - q) \cdot \hat{\mathbf{y}} + q \cdot \mathbf{z})$ and $(\mathbf{x}_s, \mathbf{y}_s)$.
5. If $\frac{1}{2} < \mu \leq \frac{2}{3} < \lambda$ (symmetric to Case 4):
 - Set $\hat{\mathbf{x}} = \frac{1}{2} \cdot \mathbf{x}_s + \frac{1}{2} \cdot \mathbf{w}$.
 - Find a best response $\hat{\mathbf{z}}$ against $\hat{\mathbf{x}}$.
 - Set $t_c = \mathcal{C}(\hat{\mathbf{x}}, \hat{\mathbf{z}}) - \mathcal{C}(\hat{\mathbf{x}}, \mathbf{z})$; $v_c = \mathcal{C}(\mathbf{x}_s, \mathbf{z}) - \mathcal{C}(\mathbf{x}_s, \hat{\mathbf{z}})$; $\hat{\lambda} = \mathcal{R}(\mathbf{w}, \hat{\mathbf{z}}) - \mathcal{R}(\mathbf{x}_s, \hat{\mathbf{z}})$.
 - 5.1 If $v_c + t_c \geq \frac{\lambda - \mu}{2}$ and $\hat{\lambda} \geq \lambda - v_c - t_c$, then set $p = \frac{2 \cdot (v_c + t_c) - (\lambda - \mu)}{2 \cdot (v_c + t_c)}$ and return the strategy profile with the minimum regret among $(\mathbf{w}, p \cdot \mathbf{z} + (1 - p) \cdot \hat{\mathbf{z}})$ and $(\mathbf{x}_s, \mathbf{y}_s)$.
 - 5.2 Else, set $q = \frac{1 - \lambda/2 - t_c}{1 + \lambda/2 - \mu - t_c}$ and return the strategy profile with the minimum regret among $((1 - q) \cdot \hat{\mathbf{x}} + q \cdot \mathbf{w}, \mathbf{z})$ and $(\mathbf{x}_s, \mathbf{y}_s)$.

Note that Cases 1-3 are identical to the Strategy-construction phase of the TS algorithm. Thus, by Lemma 7 they return a $(\frac{1}{3} + \delta)$ -Nash equilibrium. The new part concerns Cases 4 and 5. The analysis in both cases is based on certain auxiliary parameters (v_r, t_r and $\hat{\mu}$ for Case 4 and analogously for Case 5), that we define in the statement of the algorithm. These parameters encode payoff differences or regrets of the players for using specific strategies, and they help us decompose the problem into convenient subcases, so as to obtain better upper bounds on the maximum regret.

Our main result is as follows:

► **Theorem 8.** *For any constant $\delta > 0$, we can compute in polynomial-time a $(\frac{1}{3} + \delta)$ -Nash equilibrium.*

To prove the theorem, it suffices to analyze Case 4, where $\frac{1}{2} < \lambda \leq \frac{2}{3} < \mu$, since Case 5 is symmetric to Case 4 and is analyzed in exactly the same way.

Intuition and Roadmap. The overall analysis in the sequel looks rather technical, therefore, we will first provide some elaboration on the choices that the algorithm makes in Case 4. The first crucial component in the design of the new algorithm is that the upper bounds on the regret of the δ -stationary point $(\mathbf{x}_s, \mathbf{y}_s)$, obtained in Lemma 5, can be further refined based on the values of the parameters $\lambda, \mu, \hat{\mu}, v_r$. This is precisely implemented in Section 4.1 with Lemmas 11, 12, and 13. Once this is done, we then try to answer the following question: Whenever $(\mathbf{x}_s, \mathbf{y}_s)$ does not provide a $(\frac{1}{3} + \delta)$ -approximation, which profiles can form alternative candidates for a better performance? One idea is to exploit the dual strategies \mathbf{w} , and \mathbf{z} , as was also done in [36]. However, the profile (\mathbf{w}, \mathbf{z}) may not be a $(\frac{1}{3} + \delta)$ -equilibrium either (in most cases). A next attempt then is to consider appropriate convex combinations of the primal and the dual strategy for each player, i.e., a combination of \mathbf{x}_s and \mathbf{w} for the row player and \mathbf{y}_s and \mathbf{z} for the column player. Unfortunately, this again does not work in all cases. But one next step is to also take into consideration best-response strategies against such convex combinations. E.g., the strategy $\hat{\mathbf{w}}$ defined in Case 4 is a best response to the equiprobable combination of \mathbf{y}_s and \mathbf{z} . This completes our weaponry, and at the end, in all subcases of Case 4, we consider profiles where the row player uses a convex combination of \mathbf{w} and $\hat{\mathbf{w}}$, and the column player selects a combination between her primal and dual strategies, \mathbf{y}_s and \mathbf{z} . Analogous profiles with the roles of the players reversed are constructed for Case 5 too. Finally, we also know that whenever $(\mathbf{x}_s, \mathbf{y}_s)$ does not attain a $(\frac{1}{3} + \delta)$ -approximation, this restricts the relation between the parameters $\lambda, \mu, \hat{\mu}$ and v_r due to the lemmas of Section 4.1. This is exploitable for us in the sense that it allows us to construct the exact coefficients for the convex combinations that we use so as to have the desired approximation.

To proceed, we start with two helpful observations, which are used repeatedly for the analysis of Cases 4.1 and 4.2.

► **Lemma 9.** *It holds that $\mathcal{R}(\hat{\mathbf{w}}, \mathbf{z}) \geq \lambda + v_r + 2t_r$.*

Proof. By the definition of t_r , inside Case 4, we have that it holds that $\mathcal{R}(\hat{\mathbf{w}}, \hat{\mathbf{y}}) = \mathcal{R}(\mathbf{w}, \hat{\mathbf{y}}) + t_r$. Hence,

$$\frac{\mathcal{R}(\hat{\mathbf{w}}, \mathbf{y}_s)}{2} + \frac{\mathcal{R}(\hat{\mathbf{w}}, \mathbf{z})}{2} = \mathcal{R}(\hat{\mathbf{w}}, \hat{\mathbf{y}}) = \frac{\mathcal{R}(\mathbf{w}, \mathbf{y}_s)}{2} + \frac{\mathcal{R}(\mathbf{w}, \mathbf{z})}{2} + t_r \Rightarrow$$

$$\mathcal{R}(\hat{\mathbf{w}}, \mathbf{z}) = (\mathcal{R}(\mathbf{w}, \mathbf{y}_s) - \mathcal{R}(\hat{\mathbf{w}}, \mathbf{y}_s)) + \mathcal{R}(\mathbf{w}, \mathbf{z}) + 2t_r \geq \lambda + v_r + 2t_r,$$

since $\mathcal{R}(\mathbf{w}, \mathbf{y}_s) - \mathcal{R}(\hat{\mathbf{w}}, \mathbf{y}_s) = v_r$, and $\mathcal{R}(\mathbf{w}, \mathbf{z}) \geq \lambda$ (by the fact after Equation (1)). ◀

► **Corollary 10.** *It holds that $v_r \leq 1 - \lambda - 2t_r$, or equivalently $t_r \leq \frac{1 - \lambda - v_r}{2}$.*

Proof. By the previous lemma we have $\mathcal{R}(\hat{\mathbf{w}}, \mathbf{z}) \geq \lambda + v_r + 2t_r \Rightarrow v_r \leq 1 - \lambda - 2t_r$, since $\mathcal{R}(\hat{\mathbf{w}}, \mathbf{z}) \leq 1$. ◀

4.1 Bounding the regret of δ -stationary points

In this subsection, we establish three important lemmas that provide different ways of bounding the maximum regret of any δ -stationary point. The first of these lemmas is an improvement over [36], where we add a third upper bound for the δ -stationary point, in addition to the bounds stated in Lemma 5 from Section 3.

► **Lemma 11.** *Let $(\mathbf{x}_s, \mathbf{y}_s)$ be a δ -stationary point with $\delta \geq 0$, and let P be obtained by an optimal solution of $\text{Dual}(\mathbf{x}_s, \mathbf{y}_s)$, as the sum of the dual variables: $P = \sum_{i \in B_r(\mathbf{y}_s)} p_i$. It holds that $g(\mathbf{x}_s, \mathbf{y}_s) \leq \min\{P \cdot \lambda, (1 - P) \cdot \mu, P \cdot v_r + (1 - P) \cdot \hat{\mu}\} + \delta$.*

41:10 A Polynomial-Time Algorithm for 1/3-NE in Bimatrix Games

Proof. By Lemma 5 it holds that $g(\mathbf{x}_s, \mathbf{y}_s) \leq \min\{P \cdot \lambda, (1-P) \cdot \mu\} + \delta$. So, it suffices to prove that $g(\mathbf{x}_s, \mathbf{y}_s) \leq P \cdot v_r + (1-P) \cdot \hat{\mu} + \delta$. This follows from Lemma 4 when we set $(\mathbf{x}', \mathbf{y}') = (\hat{\mathbf{w}}, \mathbf{y}_s)$. Indeed, in this case we have $g(\mathbf{x}_s, \mathbf{y}_s) \leq P \cdot (\mathcal{R}(\mathbf{w}, \mathbf{y}_s) - \mathcal{R}(\hat{\mathbf{w}}, \mathbf{y}_s) - \mathcal{R}(\mathbf{x}_s, \mathbf{y}_s) + \mathcal{R}(\mathbf{x}_s, \mathbf{y}_s)) + (1-P) \cdot (\mathcal{C}(\hat{\mathbf{w}}, \mathbf{z}) - \mathcal{C}(\hat{\mathbf{w}}, \mathbf{y}_s) - \mathcal{C}(\mathbf{x}_s, \mathbf{y}_s) + \mathcal{C}(\mathbf{x}_s, \mathbf{y}_s)) = P \cdot v_r + (1-P) \cdot \hat{\mu} + \delta$, by the definitions of v_r and $\hat{\mu}$. \blacktriangleleft

The remaining two lemmas help in attaining a more fine-grained analysis on upper bounding the regret of the players, under the restrictions on the values of λ and μ in Case 4.

► **Lemma 12.** *Let $(\mathbf{x}_s, \mathbf{y}_s)$ be a δ -stationary point with $\delta \geq 0$, and let $\hat{\mu} \geq v_r$, and $\lambda > \frac{1}{2}$. Then, it holds that $g(\mathbf{x}_s, \mathbf{y}_s) \leq \frac{\hat{\mu} \cdot \lambda}{\lambda + \hat{\mu} - v_r} + \delta$.*

Proof. By Lemma 11 we have $g(\mathbf{x}_s, \mathbf{y}_s) \leq \min\{P \cdot \lambda, P \cdot v_r + (1-P) \cdot \hat{\mu}\} + \delta$. Note that $P \cdot \lambda$ is an increasing linear function of P , and $P \cdot v_r + (1-P) \cdot \hat{\mu}$ is a decreasing linear function of P , because $\hat{\mu} \geq v_r$. Therefore, the maximum of the minimum of these two functions is achieved at the point where they are equal, which is for $P' = \frac{\hat{\mu}}{\lambda + \hat{\mu} - v_r}$, as long as $P' \in [0, 1]$ (recall that P is constrained to belong to this interval). To check that P' is a valid point, observe first that since $\lambda > \frac{1}{2}$ and $\hat{\mu} \geq v_r$, the denominator of P' is positive. Also, again using that $\lambda > \frac{1}{2}$, Corollary 10 implies that $v_r \leq 1 - \lambda \leq \frac{1}{2}$, hence $\lambda > v_r$, which means that $P' \in [0, 1]$. Thus, $g(\mathbf{x}_s, \mathbf{y}_s) \leq \lambda \cdot P' + \delta = \frac{\hat{\mu} \cdot \lambda}{\lambda + \hat{\mu} - v_r} + \delta$. \blacktriangleleft

► **Lemma 13.** *Let $(\mathbf{x}_s, \mathbf{y}_s)$ be a δ -stationary point with $\delta \geq 0$, and let $\hat{\mu} < v_r$, $\lambda > \frac{1}{2}$, and $\mu > \frac{2}{3}$. Then, it holds that $g(\mathbf{x}_s, \mathbf{y}_s) \leq \frac{v_r \cdot \mu}{\mu - \hat{\mu} + v_r} + \delta$.*

4.2 Case 4.1 of the Improved Strategy-construction Phase

We now analyze the approximation we obtain, when we fall into Case 4.1 of the algorithm. We establish that either the δ -stationary point has the desired approximation or otherwise, this is achieved by having the row player use an appropriate convex combination of \mathbf{w} and $\hat{\mathbf{w}}$ and the column player play the dual strategy \mathbf{z} .

► **Lemma 14.** *If $v_r + t_r \geq \frac{\mu - \lambda}{2}$, and $\hat{\mu} \geq \mu - v_r - t_r$, then for the strategy profile $(p \cdot \mathbf{w} + (1-p) \cdot \hat{\mathbf{w}}, \mathbf{z})$, with $p = \frac{2 \cdot (v_r + t_r) - (\mu - \lambda)}{2 \cdot (v_r + t_r)}$, the payoff of both the row and the column player is at least $\frac{\lambda + \mu}{2}$.*

Proof. Note first that under the assumptions of the lemma, and since $\mu > \lambda$, the parameter p is a valid probability. For the row player, we have that her payoff is

$$\begin{aligned} \mathcal{R}(p \cdot \mathbf{w} + (1-p) \cdot \hat{\mathbf{w}}, \mathbf{z}) &= p \cdot \mathcal{R}(\mathbf{w}, \mathbf{z}) + (1-p) \cdot \mathcal{R}(\hat{\mathbf{w}}, \mathbf{z}) \\ &\geq p \cdot \lambda + (1-p) \cdot \lambda + (1-p) \cdot (v_r + t_r) \quad (\text{from Lemma 9}) \\ &= \lambda + (1-p) \cdot (v_r + t_r) \\ &= \lambda + \frac{(\mu - \lambda)}{2} \quad \left(\text{since } 1-p = \frac{\mu - \lambda}{2 \cdot (v_r + t_r)} \right) \\ &= \frac{\lambda + \mu}{2}. \end{aligned}$$

For the column player we have that her payoff is

$$\begin{aligned}
\mathcal{C}(p \cdot \mathbf{w} + (1-p) \cdot \hat{\mathbf{w}}, \mathbf{z}) &= p \cdot \mathcal{C}(\mathbf{w}, \mathbf{z}) + (1-p) \cdot \mathcal{C}(\hat{\mathbf{w}}, \mathbf{z}) \\
&\geq p \cdot \mu + (1-p) \cdot \hat{\mu} \quad (\text{Since } \hat{\mu} = \mathcal{C}(\hat{\mathbf{w}}, \mathbf{z}) - \mathcal{C}(\hat{\mathbf{w}}, \mathbf{y}_s) \leq \mathcal{C}(\hat{\mathbf{w}}, \mathbf{z})) \\
&\geq p \cdot \mu + (1-p) \cdot \mu - (1-p) \cdot (v_r + t_r) \quad (\text{Since } \hat{\mu} \geq \mu - v_r - t_r) \\
&= \mu - \frac{(\mu - \lambda)}{2} \quad \left(\text{since } 1-p = \frac{\mu - \lambda}{2 \cdot (v_r + t_r)} \right) \\
&= \frac{\mu + \lambda}{2}. \quad \blacktriangleleft
\end{aligned}$$

► **Lemma 15.** *Let $p \in [0, 1]$, be such that $\mathcal{R}(p \cdot \mathbf{w} + (1-p) \cdot \hat{\mathbf{w}}, \mathbf{z}) \geq \frac{\lambda + \mu}{2}$, and $\mathcal{C}(p \cdot \mathbf{w} + (1-p) \cdot \hat{\mathbf{w}}, \mathbf{z}) \geq \frac{\lambda + \mu}{2}$. Then, either $(\mathbf{x}_s, \mathbf{y}_s)$ is a $(\frac{1}{3} + \delta)$ -Nash equilibrium, or $(p \cdot \mathbf{w} + (1-p) \cdot \hat{\mathbf{w}}, \mathbf{z})$ is a $\frac{1}{3}$ -Nash equilibrium.*

Proof. The regret of either player at the strategy profile $(p \cdot \mathbf{w} + (1-p) \cdot \hat{\mathbf{w}}, \mathbf{z})$ is at most $1 - \frac{\lambda + \mu}{2}$, since the payoff of any player is no less than $\frac{\lambda + \mu}{2}$ and the best-response payoff is at most 1. On the other hand, by Lemma 5 the regret of each player at the δ -stationary point $(\mathbf{x}_s, \mathbf{y}_s)$ is at most $\frac{\lambda + \mu}{4} + \delta$. Thus, if $\lambda + \mu \leq \frac{4}{3}$, then $g(\mathbf{x}_s, \mathbf{y}_s) \leq \frac{1}{3} + \delta$. Otherwise, the maximum regret at the profile $(p \cdot \mathbf{w} + (1-p) \cdot \hat{\mathbf{w}}, \mathbf{z})$ is at most $1 - \frac{\lambda + \mu}{2} \leq \frac{1}{3}$. ◀

4.3 Case 4.2 of the Strategy-construction phase

In this case it holds that either $v_r + t_r < \frac{\mu - \lambda}{2}$ or $\hat{\mu} < \mu - v_r - t_r$. It turns out that this is a technically more intriguing case, and the reason is that the parameters are less constrained, compared to Case 4.1. As a result, we need to consider different subcases in order to have tighter upper bounds. We recall that the algorithm in this case outputs either $(\mathbf{x}_s, \mathbf{y}_s)$, or a profile where the row player selects her dual strategy \mathbf{w} , which is a best response against \mathbf{y}_s , and the column player plays a convex combination between $\hat{\mathbf{y}}$ and \mathbf{z} , which by the definition of $\hat{\mathbf{y}}$, is a convex combination of her primal strategy \mathbf{y}_s and her dual strategy \mathbf{z} .

► **Lemma 16.** *The regret of the row player at $(\mathbf{w}, \hat{\mathbf{y}})$ is t_r , and the regret of the column player is at most $1 - \frac{\mu}{2}$.*

Proof. By definition, the regret of the row player is t_r , since $\hat{\mathbf{w}}$ is a best-response strategy against $\hat{\mathbf{y}}$. On the other hand, recall by the definition of μ , that $\mathcal{C}(\mathbf{w}, \mathbf{z}) \geq \mu$. So, we have that $\mathcal{C}(\mathbf{w}, \hat{\mathbf{y}}) = \frac{\mathcal{C}(\mathbf{w}, \mathbf{y}_s)}{2} + \frac{\mathcal{C}(\mathbf{w}, \mathbf{z})}{2} \geq \frac{\mathcal{C}(\mathbf{w}, \mathbf{z})}{2} \geq \frac{\mu}{2}$. Thus, since the maximum payoff is less than or equal to 1, we have that the regret of the column player is at most $1 - \frac{\mu}{2}$. ◀

We now quantify the regret of the players at the profile $(\mathbf{w}, (1-q) \cdot \hat{\mathbf{y}} + q \cdot \mathbf{z})$, that is considered by the algorithm. In particular, we obtain an upper bound as a function of the parameters λ, μ , and t_r .

► **Lemma 17.** *Consider the strategy profile $(\mathbf{w}, (1-q) \cdot \hat{\mathbf{y}} + q \cdot \mathbf{z})$ with $q = \frac{1 - \mu/2 - t_r}{1 + \mu/2 - \lambda - t_r}$. Then, the regret of each player is no greater than $q \cdot (1 - \lambda) + (1 - q) \cdot t_r = \frac{1 - \mu/2 - t_r - \lambda + \mu \cdot \lambda/2 + \mu \cdot t_r}{1 + \mu/2 - \lambda - t_r}$.*

We now come to the core of the proof and establish that either the δ -stationary point, or the strategy profile $(\mathbf{w}, (1-q) \cdot \hat{\mathbf{y}} + q \cdot \mathbf{z})$ yields a good approximation. This is established by the following lemma.

► **Lemma 18.** *Under the assumptions of Case 4.2, either $(\mathbf{x}_s, \mathbf{y}_s)$ is a $(\frac{1}{3} + \delta)$ -Nash equilibrium, or $(\mathbf{w}, (1-q) \cdot \hat{\mathbf{y}} + q \cdot \mathbf{z})$ with $q = \frac{1 - \mu/2 - t_r}{1 + \mu/2 - \lambda - t_r}$, is a $\frac{1}{3}$ -Nash equilibrium.*

41:12 A Polynomial-Time Algorithm for 1/3-NE in Bimatrix Games

Proof. Since we are in Case 4.2, where either $v_r + t_r < \frac{\mu - \lambda}{2}$, or $\hat{\mu} < \mu - v_r - t_r$, we will split the analysis into further subcases, so that we have a more concrete relation between the relevant parameters in each subcase. More precisely, we will consider the following three subcases.

4.2(i) $v_r + t_r < \frac{\mu - \lambda}{2}$.

4.2(ii) $\hat{\mu} < \mu - v_r - t_r$, and $\hat{\mu} \geq v_r$.

4.2(iii) $\hat{\mu} < \mu - v_r - t_r$, and $\hat{\mu} < v_r$.

So far, we have not been able to have a unifying argument for all these different subcases. Consequently, we need to proceed with a separate analysis for each of them. Due to space constraints, we defer the first two subcases to the full version of this work, and we present here only Subcase 4.2(iii). We note also that among them, it was Subcase 4.2(ii) that turned out to be the lengthier and technically most subtle in obtaining the approximation bound.

Subcase 4.2(iii). Let us begin by observing that if $v_r \leq \frac{1}{3}$, then the strategy profile $(\mathbf{x}_s, \mathbf{y}_s)$ is a $(\frac{1}{3} + \delta)$ -NE. Indeed, from Lemma 13, and since $\hat{\mu} < v_r$, we get that the approximation guarantee of the stationary strategy profile is $\frac{v_r \cdot \mu}{\mu - \hat{\mu} + v_r} + \delta \leq v_r + \delta \leq \frac{1}{3} + \delta$. Hence, in what follows, we will assume that $v_r > \frac{1}{3}$. So, from Corollary 10 we have that

$$t_r \leq \frac{1 - \lambda - v_r}{2} < \frac{\frac{2}{3} - \lambda}{2} = \frac{1}{3} - \frac{\lambda}{2}. \quad (2)$$

Assume now for the sake of contradiction, that $(\mathbf{w}, (1 - q) \cdot \hat{\mathbf{y}} + q \cdot \mathbf{z})$ is not a $\frac{1}{3}$ -NE, i.e., the maximum regret is higher than $1/3$. Combining this with the bound of Lemma 17 on the regret, and by simplifying the resulting expression, we get the following inequality:

$$\frac{\lambda\mu}{2} - \frac{2\mu}{3} - \frac{2\lambda}{3} + \left(\mu - \frac{2}{3}\right) \cdot t_r + \frac{2}{3} > 0. \quad (3)$$

We will focus on the LHS of (3) and we will upper bound it by a non-positive value. Suppose that the regret at the stationary point $(\mathbf{x}_s, \mathbf{y}_s)$ is greater than $(\frac{1}{3} + \delta)$. This means from Lemma 5 that $\frac{\lambda \cdot \mu}{\lambda + \mu} > \frac{1}{3}$, which implies that $\mu > \frac{\lambda}{3\lambda - 1}$. In addition, observe that since $\mu > 2/3$, the LHS of (3) is increasing with t_r . Hence, if we use Inequality (2), we get that the LHS of (3) is upper bounded by

$$\begin{aligned} \frac{\lambda\mu}{2} - \frac{2\mu}{3} - \frac{2\lambda}{3} + \left(\mu - \frac{2}{3}\right) \cdot \left(\frac{1}{3} - \frac{\lambda}{2}\right) + \frac{2}{3} &= -\frac{\mu}{3} - \frac{\lambda}{3} + \frac{4}{9} \\ &< -\frac{\lambda}{9\lambda - 3} - \frac{\lambda}{3} + \frac{4}{9} \quad \left(\text{since } \mu > \frac{\lambda}{3\lambda - 1}\right). \end{aligned}$$

It is easy to verify though that this quantity is non-positive for every $\lambda \in [\frac{1}{2}, \frac{2}{3}]$, which in turn contradicts Inequality (3). \blacktriangleleft

5 Discussion

Our algorithm is the first improvement for a foundational problem after 15 years, during which progress had stalled. We hope that our result will again ignite the spark for actively studying ε -NE in bimatrix games. There is still a large gap between the quasi polynomial-time lower bound for some very small constant ε^* from [35], and our newly-established upper bound of $1/3$. We conjecture that closing this gap requires radically new ideas.

Our result has some extra positive consequences for games with more than two players. In [7] it was shown that if we have an algorithm that finds an α -Nash equilibrium in a $(k - 1)$ -player game, then in polynomial time we can compute a $(\frac{1}{2-\alpha})$ -NE for any k -player game. Thus, our algorithm improves the state of the art for k -player normal-form games, for any $k > 2$. Namely, we get a $(0.6 + \delta)$ -NE for three-player games, a $(5/7 + \delta)$ -NE for four-player games, and so on.

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 Rubinfeld. 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 Zhaohua Chen, Xiaotie Deng, Wenhan Huang, Hanyu Li, and Yuhao Li. On tightness of the Tsaknakis-Spirakis algorithm for approximate Nash equilibrium. In *Proceedings of SAGT*, volume 12885, pages 97–111, 2021.
- 13 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.
- 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 Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a Nash equilibrium. In *Proceedings of STOC*, pages 71–78, 2006.
- 17 Constantinos Daskalakis, Aranyak Mehta, and Christos Papadimitriou. Progress in approximate Nash equilibria. In *Proceedings of EC*, pages 355–358, 2007.
- 18 Constantinos Daskalakis, Aranyak Mehta, and Christos Papadimitriou. A note on approximate Nash equilibria. *Theoretical Computer Science*, 410(17):1581–1588, 2009.

- 19 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. doi:10.1016/j.jcss.2021.11.002.
- 20 Argyrios Deligkas, John Fearnley, and Rahul Savani. Inapproximability results for constrained approximate Nash equilibria. *Inf. Comput.*, 262:40–56, 2018.
- 21 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.
- 22 Ravi Kannan and Thorsten Theobald. Games of fixed rank: A hierarchy of bimatrix games. *Economic Theory*, 42(1):157–173, 2010.
- 23 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.
- 24 Spyros C. Kontogiannis and Paul G. Spirakis. Well supported approximate equilibria in bimatrix games. *Algorithmica*, 57(4):653–667, 2010.
- 25 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.
- 26 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.
- 27 Richard Lipton, Evangelos Markakis, and Aranyak Mehta. Playing large games using simple strategies. In *Proceedings of EC*, pages 36–41, 2003.
- 28 Zhengyang Liu and Ying Sheng. On the approximation of Nash equilibria in sparse win-lose games. In *Proceedings of AAAI*, volume 32(1), 2018.
- 29 Andrew McLennan and Rabee Tourky. Imitation games and computation. *Games and Economic Behavior*, 70(1):4–11, 2010.
- 30 Andrew McLennan and Rabee Tourky. Simple complexity from imitation games. *Games and Economic Behavior*, 68(2):683–688, 2010.
- 31 Ruta Mehta. Constant rank two-player games are PPAD-hard. *SIAM J. Comput.*, 47(5):1858–1887, 2018.
- 32 Aniket Murhekar and Ruta Mehta. Approximate Nash equilibria of imitation games: Algorithms and complexity. In *Proceedings of AAMAS*, pages 887–894, 2020.
- 33 John Nash. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, 1951.
- 34 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.
- 35 Aviad Rubinfeld. Settling the complexity of computing approximate two-player Nash equilibria. In *Proceedings of FOCS*, pages 258–265, 2016.
- 36 Haralambos Tsaknakis and Paul G. Spirakis. An optimization approach for approximate Nash equilibria. *Internet Mathematics*, 5(4):365–382, 2008.