CrossMark

Approximate Well-supported Nash Equilibria Below Two-thirds

John Fearnley[1] · Paul W. Goldberg[2] ·
Rahul Savani[1] · Troels Bjerre Sørensen[3]

Abstract   In an ϵ-Nash equilibrium, a player can gain at most ϵ by changing his behaviour. Recent work has addressed the question of how best to compute ϵ-Nash equilibria, and for what values of ϵ a polynomial-time algorithm exists. An ϵ-*well-supported* Nash equilibrium (ϵ-WSNE) has the additional requirement that any strategy that is used with non-zero probability by a player must have payoff at most ϵ less than a best response. A recent algorithm of Kontogiannis and Spirakis shows how to compute a 2/3-WSNE in polynomial time, for bimatrix games. Here we introduce a new technique that leads to an improvement to the worst-case approximation guarantee.

Keywords   Bimatrix games · Nash equilibria · Well-supported approximate equilibria

1 Introduction

In a bimatrix game, a Nash equilibrium is a pair of strategies in which the two players only assign probability to best response strategies. The apparent hardness of computing an exact Nash equilibrium [3,5] has led to work on computing approximate Nash equilibria, and two notions of approximate Nash equilibria have been developed. The

✉ Rahul Savani
  rahul.savani@liverpool.ac.uk

[1] Department of Computer Science, University of Liverpool, Liverpool, UK

[2] University of Oxford, Wolfson Building, Parks Road, Oxford, UK

[3] ITU Copenhagen, Rued Langgaards Vej 7, 2300 Copenhagen, Denmark

first, and more widely studied, notion is of an ϵ-*approximate Nash equilibrium* (ϵ-Nash), where each player is required to achieve an expected payoff that is within ϵ of a best response. A line of work [2,6,7] has investigated the value of ϵ that can be guaranteed in polynomial time. The current best result in this setting is a polynomial time algorithm that always finds a 0.3393-Nash equilibrium [13].

However, ϵ-Nash equilibria have a drawback: since they only require that the expected payoff is within ϵ of a pure best response, it is possible that a player could be required to place probability on a strategy that is arbitrarily far from being a best response. This issue is addressed by the second notion of an approximate Nash equilibrium. An ϵ-*well supported approximate Nash equilibrium* (ϵ-WSNE), requires that both players only place probability on strategies that have payoff within ϵ of a pure best response. This is a stronger notion of equilibrium, because every ϵ-WSNE is an ϵ-Nash, but the converse is not true.

In contrast to ϵ-Nash, there has been relatively little work ϵ-WSNE. The first result on the subject gave a $\frac{5}{6}$ additive approximation [7], but this only holds if a certain a graph-theoretic conjecture is true. The best-known polynomial-time additive approximation algorithm was given by Kontogiannis and Spirakis, and achieves a $\frac{2}{3}$-approximation [11]. We will call this algorithm the KS algorithm. In [10], which is an earlier conference version of [11], the authors presented an algorithm that they claimed was polynomial-time and achieves a ϕ-WSNE, where $\phi = \frac{\sqrt{11}}{2} - 1 \approx 0.6583$, but this was later withdrawn, and instead the polynomial-time $\frac{2}{3}$-approximation algorithm was presented in [11]. Recently, it has been shown that for every $\delta > 0$, a $\left(\frac{1}{2} + \delta\right)$-WSNE can be found in polynomial times for *symmetric* bimatrix games [4]. It has also been shown that there is a PTAS for ϵ-WSNE if and only if there is a PTAS for ϵ-Nash [3, Lemma 3.2].

1.1 Our Contribution

In this paper, we develop analgorithm for finding an ϵ-WSNE with $\epsilon < \frac{2}{3}$. Our approach to modifying the KS algorithm for finding a $\frac{2}{3}$-WSNE, by adding two additional procedures: we perform a brute-force search that finds the best WSNE with a 2×2 support, and we attempt to improve the ϵ-WSNE returned by the KS algorithm by shifting the probabilities of the two players. We show that one of these two approaches will always find an ϵ-WSNE with $\epsilon = \frac{2}{3} - 0.005913759$. Our results are particularly interesting when compared to a recent support size lower bound of Anbalagan, Norin, Savani, and Vetta, who showed that there exist games in which all ϵ-WSNE with $\epsilon < \frac{2}{3}$ have super-constant sized supports [1].

A preliminary version of this paper was published in the proceedings of SAGT 2012 [8]. In that version of the paper, we gave a polynomial time algorithm for finding an ϵ-WSNE with $\epsilon = \frac{2}{3} - 0.004735$. It turns out that one of the inequalities used to show this result[1] was not as strong as it could have been, and correcting this led to

---

[1] The inequality in question appeared in Proposition 16 of the preliminary version, and is now part of Proposition 12.

the improved bound in this version of the paper. We have also greatly simplified the computer-assisted proof that is used at the end of the paper. The preliminary version of the paper used a rather opaque method involving sensitivity analysis of a linear program. We have reformulated the LP so that the relevant values can be read directly from the solution of the LP. The algorithm has since been implemented as part of an experimental study [9].

The paper will proceed as follows. In Sect. 2 we give the basic definitions that will be needed in this paper. In Sect. 3 we give a high level overview of our algorithms, along with the intuition behind our two modifications. In Sect. 4, we formally define our algorithm and state our main theorem. In Sect. 5 we give a high level overview of the proof, before then proceeding with the proof in Sects. 6 through 9.

2 Definitions

A *bimatrix game* is a pair (R, C) of two $n \times n$ matrices: R gives payoffs for the *row player*, and C gives payoffs for the *column player*. We assume that all payoffs are in the range [0, 1]. We use $[n] = \{1, 2, \ldots n\}$ to denote the *pure strategies* for each player. To play the game, both players simultaneously select a pure strategy: the row player selects a row $i \in [n]$, and the column player selects a column $j \in [n]$. The row player then receives $R_{i,j}$, and the column player receives $C_{i,j}$.

A *mixed strategy* is a probability distribution over $[n]$. We denote a mixed strategy as a vector $\mathbf{x}$ of length n, such that $\mathbf{x}_i$ is the probability that the pure strategy i is played. The *support* of mixed strategy $\mathbf{x}$, denoted Supp($\mathbf{x}$), is the set of pure strategies i with $\mathbf{x}_i > 0$. If $\mathbf{x}$ and $\mathbf{y}$ are mixed strategies for the row and column player, respectively, then we call $(\mathbf{x}, \mathbf{y})$ a *mixed strategy profile*.

Let $\mathbf{y}$ be a mixed strategy for the column player. The *best responses* against $\mathbf{y}$ for the row player is the set of pure strategies that maximize the payoff against $\mathbf{y}$. More formally, a pure strategy $i \in [n]$ is a best response against $\mathbf{y}$ if, for all pure strategies $i' \in [n]$ we have: $\sum_{j \in [n]} \mathbf{y}_j \cdot R_{i,j} \geq \sum_{j \in [n]} \mathbf{y}_j \cdot R_{i',j}$. Column player best responses are defined analogously. A mixed strategy profile $(\mathbf{x}, \mathbf{y})$ is a *mixed Nash equilibrium* if every pure strategy in Supp($\mathbf{x}$) is a best response against $\mathbf{y}$, and every pure strategy in Supp($\mathbf{y}$) is a best response against $\mathbf{x}$. Nash [12] showed that all bimatrix games have a mixed Nash equilibrium.

An *approximate well-supported Nash equilibrium* weakens the requirements of a mixed Nash equilibrium. For a mixed strategy $\mathbf{y}$ of the column player, a pure strategy $i \in [n]$ is an ϵ-*best response* for the row player if, for all pure strategies $i' \in [n]$ we have: $\sum_{j \in [n]} \mathbf{y}_j \cdot R_{i,j} \geq \sum_{j \in [n]} \mathbf{y}_j \cdot R_{i',j} - \epsilon$. We define ϵ-best responses for the column player analogously. A mixed strategy profile $(\mathbf{x}, \mathbf{y})$ is an ϵ-*well-supported Nash equilibrium* (ϵ-WSNE) if every pure strategy in Supp($\mathbf{x}$) is an ϵ-best response against $\mathbf{y}$, and every pure strategy in Supp($\mathbf{y}$) is an ϵ-best response against $\mathbf{x}$.

3 Outline

Before we give a technical presentation of our algorithm, we begin by giving the high level ideas behind our techniques. Our approach builds upon the algorithm of
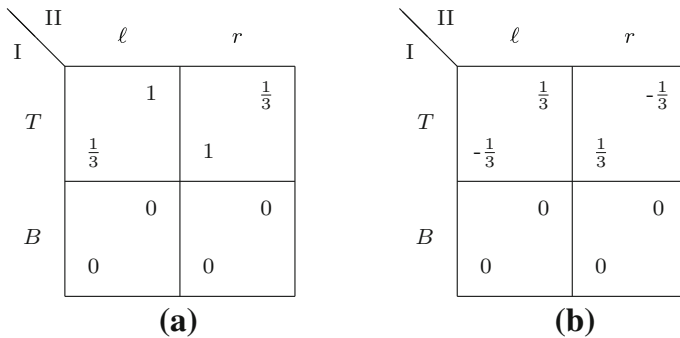
Fig. 1 The *left figure* shows a worst case example for the KS algorithm. The *right figure* shows the corresponding zero-sum game $(D, -D)$

Kontogiannis and Spirakis for finding a $\frac{2}{3}$-WSNE, so let us begin by describing their algorithm. Given a bimatrix game (R, C), the KS algorithm performs two steps:

1. Check if there is a pure strategy profile under which both players get payoff at least $\frac{1}{3}$. If so, that pure strategy profile is a $\frac{2}{3}$-WSNE.
2. Construct the zero-sum game $(D, -D)$ where $D = \frac{1}{2}(R - C)$, and let $(\mathbf{x}, \mathbf{y})$ be a Nash-equilibrium of $(D, -D)$.

Kontogiannis and Spirakis showed that if step 1 failed to find a pure $\frac{2}{3}$-WSNE of (R, C), then $(\mathbf{x}, \mathbf{y})$ is a $\frac{2}{3}$-WSNE of (R, C). Our goal is to show that the WSNEs found by the KS algorithm can be improved: either by *shifting probabilities,* or by finding a *matching pennies* sub-game. We now show the motivation behind these two procedures.

3.1 Shifting Probabilities

Figure 1a shows an example for which the KS algorithm actually produces a $\frac{2}{3}$-WSNE. For simplicity of exposition, we have ignored the first part of the algorithm here: note that in (T, ℓ) and (T, r) both players have payoff greater than or equal to $\frac{1}{3}$. If we replace both $\frac{1}{3}$ payoffs with $\frac{1}{3} - \delta$, for some arbitrarily small $\delta > 0$, then this issue is avoided, and the properties of the example do not significantly change.

Figure 1b shows the corresponding zero-sum game. Let $(\mathbf{x}, \mathbf{y})$ be a strategy profile in which the row player plays B, and the column player mixes equally between ℓ and r. Observe that $(\mathbf{x}, \mathbf{y})$ is a Nash equilibrium of the zero-sum game, and that it is $\frac{2}{3}$-WSNE of (R, C), and no better. Therefore, this example is a worst-case example for the KS-algorithm.

Our observation is that $(\mathbf{x}, \mathbf{y})$ can be improved by shifting probabilities. We can improve things for the row player by transferring some of the *column* player's probability from r to ℓ. This reduces the payoff of T while leaving the payoff of B unchanged. Thus, B becomes an ϵ-best response for $\epsilon < \frac{2}{3}$, and we obtain a better WSNE.
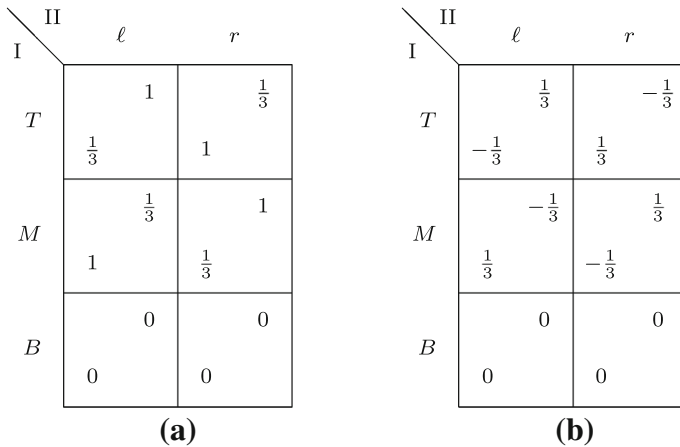
Fig. 2 The *left figure* shows an example for which the approach of shifting probabilities fails. The *right figure* shows the corresponding zero-sum game

3.2 Matching Pennies

Figure 2 shows a game in which the approach of shifting probabilities does not work. To see this, observe that the strategy profile $(\mathbf{x}, \mathbf{y})$ where the row player plays B, and the column player mixes uniformly over ℓ and r is a Nash equilibrium of the game shown in Fig. 2b. When $(\mathbf{x}, \mathbf{y})$ is played in the original game in Fig. 2a, this gives a $\frac{2}{3}$-WSNE and no better. However, in this case, the column player cannot make the row player happier by shifting probabilities: if probability is shifted to ℓ, then the payoff of strategy T will increase, and if probability is shifted to r then the payoff of strategy M will increase.

In this case, we use a different approach. We observe that the 2×2 subgame induced by ℓ, r, T, and M, is similar to a matching pennies game. If the row player mixes uniformly over M and T, while the column player mixes uniformly of ℓ and r, then both players will obtain payoff at least 0.5, which yields a 0.5-WSNE.

3.3 Our Approach

We will show that one of these two techniques can always be applied. Our algorithm will first perform a brute force search over all 2×2 sub-games in order to determine whether there is a matching pennies sub game. If such a game is not found, then we run the KS algorithm and attempt to shift probabilities in the resulting strategy profile. Ultimately, we show that this algorithm always produces a $\left(\frac{2}{3} - 0.005913759\right)$-WSNE.

4 Our Algorithm

In this section we formally describe our algorithm for finding a WSNE. We begin by describing a method for finding the best WSNE on a given pair of supports, and then move on to describe the three procedures that make up our algorithm.

4.1 The Best WSNE on a Pair of Supports

Let S_c and S_r be supports for the column and row player, respectively. We first define an LP, which assumes that the row player uses a strategy with support S_r, and then finds a strategy on S_c that minimizes the difference between the row player's best response payoff, and the payoff of the strategies in S_r.

Definition 1 For each $S_r, S_c \subseteq [n]$, we define $BestR(S_r, S_c)$ to be the following linear program with variables $\epsilon \in \mathbb{R}$ and $\mathbf{y} \in \mathbb{R}^n$:

$$
\begin{aligned}
&\textbf{Minimize:} && \epsilon \\
&\textbf{Subject to:} && R_{i'} \cdot \mathbf{y} - R_i \cdot \mathbf{y} \le \epsilon && i \in S_r, \ i' \in [n] \\
& && \mathbf{y}_j = 0 && j \notin S_c \\
& && \mathbf{y}_j \ge 0 && j \in [n] \\
& && \textstyle\sum_{j \in [n]} \mathbf{y}_j = 1 &&
\end{aligned}
\tag{1}
$$

Similarly, the following LP assumes that the column player uses a strategy with support S_c, and finds a strategy on S_r that minimizes the difference between the column player's best response payoff, and the payoff of the strategies in S_c.

Definition 2 For each $S_r, S_c \subseteq [n]$, we define $BestC(S_r, S_c)$ to be the following linear program with variables $\epsilon \in \mathbb{R}$ and $\mathbf{x} \in \mathbb{R}^n$:

$$
\begin{aligned}
&\textbf{Minimize:} && \epsilon \\
&\textbf{Subject to:} && C_{j'}^T \cdot \mathbf{x} - C_j^T \cdot \mathbf{x} \le \epsilon && j \in S_c, \ j' \in [n] \\
& && \mathbf{x}_i = 0 && i \notin S_r \\
& && \mathbf{x}_i \ge 0 && i \in [n] \\
& && \textstyle\sum_{j \in [n]} \mathbf{x}_i = 1 &&
\end{aligned}
\tag{2}
$$

We now prove that these two LPs give a WSNE that is at least as good as the best WSNE on the supports S_r and S_c. Let $(\mathbf{y}^*, \epsilon_{\mathbf{y}})$ be a solution of $BestR(S_r, S_c)$, let $(\mathbf{x}^*, \epsilon_{\mathbf{x}})$ be a solution of $BestC(S_r, S_c)$, and let ϵ^* to be $\max(\epsilon_{\mathbf{x}}, \epsilon_{\mathbf{y}})$.

Proposition 3 *We have:*

1. $(\mathbf{x}^*, \mathbf{y}^*)$ *is an ϵ^*-WSNE.*
2. *For every ϵ-WSNE $(\mathbf{x}, \mathbf{y})$ with $Supp(\mathbf{x}) = S_r$ and $Supp(\mathbf{y}) = S_c$, we have $\epsilon^* \le \epsilon$.*

Proof The first claim is straightforward, because Constraint 1 ensures that every strategy $i \in \mathrm{Supp}(\mathbf{x}^*) \subseteq S_r$ is an $\epsilon_{\mathbf{y}}$-best response against $\mathbf{y}^*$, and every strategy $j \in \mathrm{Supp}(\mathbf{y}^*) \subseteq S_c$ is an $\epsilon_{\mathbf{x}}$-best response against $\mathbf{x}^*$. Therefore $(\mathbf{x}^*, \mathbf{y}^*)$ is a ϵ^*-WSNE.

For the second claim, let $(\mathbf{x}, \mathbf{y})$ be an ϵ-WSNE on the supports S_r and S_c. Since every row $i \in \mathrm{Supp}(\mathbf{x}) = S_r$ is an ϵ-best response against $\mathbf{y}$, we must have that $\mathbf{y}$ and ϵ are feasible in $BestR(S_r, S_c)$. For the same reason, we have that $\mathbf{x}$ and ϵ are feasible in $BestC(S_r, S_c)$. Therefore, we must have $\epsilon^* \le \epsilon$. □

Proposition 3 implies that $(\mathbf{x}^*, \mathbf{y}^*)$ is at least as good as the best WSNE on the supports S_r and S_c. Note that it is possible that $(\mathbf{x}^*, \mathbf{y}^*)$ may actually be better than

any WSNE on these supports, because the LPs do not require that $\mathbf{x}^*$ places probability on all strategies in S_r, or that $\mathbf{y}^*$ places probability on all strategies in S_c.

4.2 Our Algorithm

We now describe our algorithm for finding a WSNE in a bimatrix game. Our algorithm for finding a WSNE consists of three distinct procedures.

- *Procedure 1: find the best pure WSNE* The KS algorithm requires a preprocessing step that eliminates all games that have a pure $\frac{2}{3}$-WSNE, and this is a generalisation of that step. Suppose that the row player plays row i, and that the column player plays column j. Let: $\epsilon_r = \max_{i'}(R_{i',j}) - R_{i,j}$, and $\epsilon_c = \max_{j'}(C_{i,j'}) - C_{i,j}$. Thus i is an ϵ_r-best response against j, and that j is an ϵ_c-best response against i. Therefore, (i, j) is a $\max(\epsilon_r, \epsilon_c)$-WSNE. We can find the best pure WSNE by checking all $O(n^2)$ possible pairs of pure strategies. Let ϵ_p be the best approximation guarantee that is found by this procedure.
- *Procedure 2: find the best WSNE with 2×2 support* We can use the linear programs from Definitions 1 and 2 to implement this procedure. For each of the $O(n^4)$ possible 2×2 supports, we solve the LPs to find a WSNE. Proposition 3 implies that this WSNE is at least as good as the best WSNE on those supports. Let ϵ_m be the best approximation guarantee that is found by this procedure.
- *Procedure 3: find an improvement over the KS algorithm* The KS algorithm finds an exact Nash equilibrium $(\mathbf{x}, \mathbf{y})$ of the zero-sum game $(D, -D)$, where $D = \frac{1}{2}(R - C)$. To find an improvement over the KS algorithm we use the linear programs from Definitions 1 and 2 with parameters $S_r = \mathrm{Supp}(\mathbf{x})$ and $S_c = \mathrm{Supp}(\mathbf{y})$. Let $(\mathbf{x}^*, \mathbf{y}^*)$ be the mixed strategy profile returned by the LPs, and let ϵ_i be the smallest value such that $(\mathbf{x}^*, \mathbf{y}^*)$ is a ϵ_i-WSNE.

Note that one could replace Procedures 1 and 2 with a single procedure that searches for the best WSNE on supports of size *up to* 2. However, since Procedures 1 and 2 play different roles in our proof, we have chosen to keep them separate here.

After executing these three procedures, we take the smallest of ϵ_p, ϵ_m, and ϵ_i, and return the corresponding WSNE. Since all three procedures can be implemented in polynomial time, this is a polynomial time algorithm. The rest of this paper is dedicated to proving the following theorem.

Theorem 4 *Our algorithm finds a $\left(\frac{2}{3} - 0.005913759\right)$-WSNE.*

5 Proof Outline

In order for our proof to be as informative as possible, we will parameterize it using a constant $z > 0$. We will show the conditions under which our algorithm can produce a $\left(\frac{2}{3} - z\right)$-WSNE. At the end of the proof we will show that these conditions are satisfied for $z = 0.005913759$, which provides a proof of Theorem 4.

Our approach is to assume that Procedures 1 and 2 did not produce a $\left(\frac{2}{3} - z\right)$-WSNE, and then to use that assumption to determine the conditions under which Procedure 3 does find a $\left(\frac{2}{3} - z\right)$-WSNE. This comprises of the following steps.

– *Reanalyze the KS algorithm* The original analysis for the KS algorithm assumed that the game does not have a pure $\frac{2}{3}$-WSNE. However, in our analysis, we have assumed only that Procedure 1 did not find a pure $\left(\frac{2}{3} - z\right)$-WSNE, so the original KS analysis is no longer valid. In Sect. 6 we show that, assuming there is no pure $\left(\frac{2}{3} - z\right)$-WSNE, the KS algorithm will produce a strategy profile $(\mathbf{x}, \mathbf{y})$ where all strategies have payoff at most $\frac{2}{3} + 2z$, and therefore $(\mathbf{x}, \mathbf{y})$ is a $\left(\frac{2}{3} + 2z\right)$-WSNE.

– *Study the bad strategies* Our goal is to show that $(\mathbf{x}, \mathbf{y})$ can be improved from a $\left(\frac{2}{3} + 2z\right)$-WSNE to a $\left(\frac{2}{3} - z\right)$-WSNE. To achieve this, we show how to reduce the payoffs of all strategies from $\frac{2}{3} + 2z$ to $\frac{2}{3} - z$. While describing our approach, we will focus on the row player, but all of our techniques will actually be applied to both players. We define a *bad* row to be a row that has payoff strictly more than $\frac{2}{3} - z$. In Sect. 7 we study the properties of bad rows, and we prove that all bad rows are similar in structure to the games shown in Figs. 1 and 2. That is, most of the columns in a bad row are either *big* (ie. close to 1,) or *small* (ie. close to $\frac{1}{3}$.) We prove lower bounds on the amount of probability that the column player's strategy assigns to big and small payoffs. We then define a new strategy for the column player $\mathbf{y}^{\mathrm{imp}}$, that finds the *worst* bad row $\bar{\imath}$ (ie. the row with the largest payoff,) and shifts all probability from the big columns in $\bar{\imath}$ to the small columns in $\bar{\imath}$.

– *Apply the matching pennies argument* In Sect. 8 we use the fact that Procedure 2 did not find an ϵ-WSNE on a 2×2 support with $\epsilon < \frac{2}{3} - z$. Intuitively, this corresponds to ruling out cases like the one shown in Fig. 2. We prove that, if Procedure 2 failed to find a $\left(\frac{2}{3} - z\right)$-WSNE, then the bad rows cannot be arranged like they are in Fig. 2. This gives a formal condition on how the probability of $\mathbf{y}$ can be distributed over the columns of the bad rows, which will be used later in the proof.

– *Find an improved strategy* Since the strategy shifts all probability from big payoffs in row $\bar{\imath}$ to small payoffs in row $\bar{\imath}$, by definition, we must have that the payoff of $\bar{\imath}$ against $\mathbf{y}^{\mathrm{imp}}$ is small. However, the payoff of other rows may increase as we move from $\mathbf{y}$ and $\mathbf{y}^{\mathrm{imp}}$. We must find a trade-off between the bad rows decreasing in payoff, and other rows increasing in payoff, so we define a strategy $\mathbf{y}(t) = (1 - t) \cdot \mathbf{y} + t \cdot \mathbf{y}^{\mathrm{imp}}$, which mixes between $\mathbf{y}$ and $\mathbf{y}^{\mathrm{imp}}$. We show that there exists a t such that all rows i have payoff less than or equal to $\frac{2}{3} - z$ against $\mathbf{y}(t)$. In Sect. 9 we develop a computer-assisted proof for this task. For each z and t we formulate a linear program that gives the largest possible payoff of a row against $\mathbf{y}(t)$, and then we perform a grid search over z and t in order to find a strategy $\mathbf{y}(t)$ against which all rows have payoff at most $\frac{2}{3} - z$. Ultimately, we find that this occurs for $z = 0.005913759$, which proves Theorem 4.

Before we continue with the proof, we justify why it is possible to treat the two players independently in our analysis. In our proof, we will start with a strategy profile $(\mathbf{x}, \mathbf{y})$. At a high level, the idea is to rearrange the probabilities in $\mathbf{x}$ to create $\mathbf{x}'$ such that the column player is happier when he plays $\mathbf{y}$ against $\mathbf{x}'$. Simultaneously, we rearrange the probabilities in $\mathbf{y}$ to create $\mathbf{y}'$ such that the row player is happier when he plays $\mathbf{x}$ against $\mathbf{y}'$. We then claim that both players are happier in the profile $(\mathbf{x}', \mathbf{y}')$. To see why, observe that an approximate well supported Nash equilibrium is defined entirely by the supports that the strategies use. Since we only rearrange probabilities, ie. we

have $\text{Supp}(\mathbf{x}') \subseteq \text{Supp}(\mathbf{x})$ and $\text{Supp}(\mathbf{y}') \subseteq \text{Supp}(\mathbf{y})$, it is sufficient to consider only $\mathbf{x}'$ played against $\mathbf{y}$ and $\mathbf{y}'$ played against $\mathbf{x}$ in order to prove properties of $(\mathbf{x}', \mathbf{y}')$.

6 Reanalyzing the KS Algorithm

In this section we analyse the KS algorithm under the assumption that Procedure 1 did not find a $\left(\frac{2}{3} - z\right)$-WSNE. Note that if there is a pure strategy profile (i, j), such that $R_{i,j} \geq \frac{1}{3} + z$ and $C_{i,j} \geq \frac{1}{3} + z$, then (i, j) is a $\left(\frac{2}{3} - z\right)$-WSNE. Therefore, our assumption allows us to conclude that for all i and j we have:

$$0 \leq R_{i,j} + C_{i,j} \leq \frac{4}{3} + z. \tag{3}$$

This inequality replaces the inequality $0 \leq R_{i,j} + C_{i,j} \leq \frac{4}{3}$, which was used in the original analysis.

From now on, our analysis will be stated for the row player, with the understanding that all of our proofs can be apply symmetrically to the column player. Our goal is to show that all "worst-case" examples for the KS algorithm are similar to Figs. 1 and 2. More precisely, if $(\mathbf{x}, \mathbf{y})$ is the strategy profile returned by the KS algorithm, then we are interested in the following properties of Figs. 1 and 2:

– There exists a row $i \in \text{Supp}(\mathbf{x})$ such that $R_i \cdot \mathbf{y} = 0$ and $C_i \cdot \mathbf{y} = 0$.
– Every row i with $R_i \cdot \mathbf{y} = \frac{2}{3}$ also has $C_i \cdot \mathbf{y} = \frac{2}{3}$.

We will show that our "worst-case" examples have similar properties.

We begin with the first property. Here we show that, if $(\mathbf{x}, \mathbf{y})$ is not a $\left(\frac{2}{3} - z\right)$-WSNE, then there exists a row in the row player's support where both players have payoff close to 0.

Proposition 5 *If $(\mathbf{x}, \mathbf{y})$ is a solution of $(D, -D)$ such that there is an $i \in \text{Supp}(\mathbf{x})$ where i is not a $\left(\frac{2}{3} - z\right)$-best response against $\mathbf{y}$ in (R, C), then there is a row $i \in \text{Supp}(\mathbf{x})$ such that both of the following hold:*

$$R_i \cdot \mathbf{y} < 3z, \quad C_i \cdot \mathbf{y} < 3z.$$

Proof We begin by noting that, since $D = \frac{1}{2}(R - C)$, if $X = -\frac{1}{2}(R + C)$, then we have two equalities:

$$R = D - X, \quad C = -D - X.$$

Since $\mathbf{x}$ is a min-max strategy in $(D, -D)$, if i is a row in $\text{Supp}(\mathbf{x})$, then for all rows i' we have:

$$D_i \cdot \mathbf{y} \geq D_{i'} \cdot \mathbf{y},$$
$$(R + X)_i \cdot \mathbf{y} \geq (R + X)_{i'} \cdot \mathbf{y},$$
$$R_i \cdot \mathbf{y} \geq R_{i'} \cdot \mathbf{y} - (X_i - X_{i'}) \cdot \mathbf{y}.$$

Let $i \in \text{Supp}(\mathbf{x})$ be a row that is not a $\left(\frac{2}{3} - z\right)$-best response against $\mathbf{y}$, which exists by assumption, and let i' be a best-response against $\mathbf{y}$. We have:

$$R_{i'} \cdot \mathbf{y} - \left(\frac{2}{3} - z\right) > R_i \cdot \mathbf{y},$$

$$\geq R_{i'} \cdot \mathbf{y} - (X_i - X_{i'}) \cdot \mathbf{y}.$$

Hence, we have:

$$(X_i - X_{i'}) \cdot \mathbf{y} > \left(\frac{2}{3} - z\right).$$

Note that, by Eq. (3), all entries of X must lie in the range $[-\frac{2}{3} - \frac{1}{2}z, 0]$. In particular, this implies that:

$$-\frac{2}{3} - \frac{1}{2}z \leq X_{i'} \cdot \mathbf{y} < X_i \cdot \mathbf{y} - \left(\frac{2}{3} - z\right).$$

This implies that $-\frac{3}{2}z < X_i \cdot \mathbf{y} \leq 0$. Now, using the definition of X we obtain:

$$-\frac{1}{2}(R + C)_i \cdot \mathbf{y} > -\frac{3}{2}z,$$

which is equivalent to:

$$(R + C)_i \cdot \mathbf{y} < 3z.$$

Since both R and C are non-negative, we have completed the proof. □

We now consider the second property. Here we show that, if $(\mathbf{x}, \mathbf{y})$ is not a $\left(\frac{2}{3} - z\right)$-WSNE, then every row has payoff at most $\frac{2}{3} + 2z$, and that for all rows i we have that $R_i \cdot \mathbf{y} - C_i \cdot \mathbf{y}$ is small.

Proposition 6 *If $(\mathbf{x}, \mathbf{y})$ is a solution of $(D, -D)$ such that there is an $i \in \text{Supp}(\mathbf{x})$ where i is not a $\left(\frac{2}{3} - z\right)$-best response against $\mathbf{y}$ in (R, C), then for all rows i' both of the following hold:*

$$R_{i'} \cdot \mathbf{y} \leq \frac{2}{3} + 2z, \quad R_{i'} \cdot \mathbf{y} - C_{i'} \cdot \mathbf{y} \leq 3z.$$

Proof Let i be the row in $\text{Supp}(\mathbf{x})$ whose existence is implied by Proposition 5. This proposition, along with the fact that all entries in R and C are non-negative, implies that:

$$0 \leq R_i \cdot \mathbf{y} < 3z, \quad 0 \leq C_i \cdot \mathbf{y} < 3z.$$

By definition we have $D = \frac{1}{2}(R - C)$, and therefore, we have:

$$-\frac{3}{2}z < D_i \cdot \mathbf{y} < \frac{3}{2}z.$$

Now, since x is a min-max strategy for the zero-sum game $(D, -D)$, we must have, for all rows i':

$$D_{i'} \cdot \mathbf{y} \le D_i \cdot \mathbf{y} < \frac{3}{2}z.$$

Thus, we have:

$$\frac{1}{2}(R_{i'} - C_{i'}) \cdot \mathbf{y} < \frac{3}{2}z.$$

Rearranging this yields one of our two conclusions:

$$R_{i'} \cdot \mathbf{y} < C_{i'} \cdot \mathbf{y} + 3z. \tag{4}$$

We can obtain the other conclusion by rearranging Eq. (3), to argue that for all rows i and all columns j we have:

$$C_{i,j} \le \frac{4}{3} + z - R_{i,j}.$$

Then, Eq. (4) implies that:

$$R_{i'} \cdot \mathbf{y} < C_{i'} \cdot \mathbf{y} + 3z \le \frac{4}{3} + 4z - R_{i'} \cdot \mathbf{y}.$$

This implies that $2 \cdot R_{i'} \cdot \mathbf{y} \le \frac{4}{3} + 4z$, and so we have $R_{i'} \cdot \mathbf{y} \le \frac{2}{3} + 2z$. □

Proposition 6 shows that $R_{i'} \cdot \mathbf{y} \le \frac{2}{3} + 2z$ holds for all rows i'. Using the same argument symmetrically, we can also show that $C_{j'} \cdot \mathbf{x} \le \frac{2}{3} + 2z$ for all columns j'. Thus, we have shown that if there is no pure $\left(\frac{2}{3} - z\right)$-WSNE, then the KS algorithm will produce a mixed strategy pair $(\mathbf{x}, \mathbf{y})$ that is a $\left(\frac{2}{3} + 2z\right)$-WSNE.

The main goal of our proof is to show that the probabilities in $\mathbf{x}$ and $\mathbf{y}$ can be rearranged to construct a $\left(\frac{2}{3} - z\right)$-WSNE. From this point onwards, we only focus on improving the strategy $\mathbf{y}$, with the understanding that all of our techniques can be applied in the same way to improve the strategy $\mathbf{x}$. For the rest of the paper, we will fix $(\mathbf{x}, \mathbf{y})$ to be the strategy profile produced by the KS algorithm, and we will assume that it is not a $\left(\frac{2}{3} - z\right)$-WSNE.

7 Bad Rows

In order to transform $(\mathbf{x}, \mathbf{y})$ to a $\left(\frac{2}{3} - z\right)$-WSNE, we will ensure that there are no rows with payoff greater than $\frac{2}{3} - z$. Thus, we define a *bad* row to be a row i whose payoff lies in the range $\frac{2}{3} - z < R_i \cdot \mathbf{y} \le \frac{2}{3} + 2z$. Furthermore, we classify the bad rows according to how bad they are.

Definition 7 A row i is *q-bad* if $R_i \cdot \mathbf{y} = \frac{2}{3} + 2z - qz$.

Since $(\mathbf{x}, \mathbf{y})$ is a $\left(\frac{2}{3} + 2z\right)$-WSNE, we have that every row is q-bad for some $q \ge 0$. Moreover, we are interested in improving the q-bad rows with $0 \le q < 3$. In this

section, we study the properties of q-bad rows, and we show that they must look similar to the bad rows in Figs. 1 and 2.

To begin, we observe that if i is a q-bad row, then we can apply the second inequality of Proposition 6 to obtain:

$$C_i \cdot \mathbf{y} \geq \frac{2}{3} - z - qz. \tag{5}$$

Now consider a q-bad row i with $q < 3$. We can deduce the following three properties about row i.

– Definition 7 tells us that $R_i \cdot \mathbf{y}$ is close to $\frac{2}{3}$.
– Equation (5) tells us that $C_i \cdot \mathbf{y}$ is close to $\frac{2}{3}$.
– The fact that there are no pure $\left(\frac{2}{3} - z\right)$-WSNEs implies that, for each column j, we must either have $R_{i,j} < \frac{1}{3} + z$ or $C_{i,j} < \frac{1}{3} + z$, because otherwise (i, j) would be a pure $\left(\frac{2}{3} - z\right)$-WSNE.

In order to satisfy all three of these conditions simultaneously, the row i must have a very particular form: approximately half of the probability assigned by $\mathbf{y}$ must be given to columns j where $R_{i,j}$ is close to 1 and $C_{i,j}$ is close to $\frac{1}{3}$, and approximately half of the probability assigned by $\mathbf{y}$ must be given to columns j where $R_{i,j}$ is close to $\frac{1}{3}$ and $C_{i,j}$ is close to 1.

Building on this observation, we split the columns of each row i into three sets. We define the set B_i of *big* columns to be $B_i = \{j : R_{i,j} \geq \frac{2}{3} + 2z\}$, and the set S_i of *small* columns to be $S_i = \{j : C_{i,j} \geq \frac{2}{3} + 2z\}$. Finally, we have the set of *other* columns $O_i = \{1, 2, \ldots, n\} \setminus (B_i \cup S_i)$, which contains all columns that are neither big nor small.

We now formalise our observations by giving inequalities about the amount of probability that $\mathbf{y}$ can assign to B_i, S_i, and O_i, for every q-bad row i. The following proposition proves three inequalities. The first inequality is proved using Markov's inequality. The second and third inequalities arise from substituting the first inequality into Definition 7 and Eq. (5), respectively. The full proof of this proposition is presented in 1.

Proposition 8 *If i is a q-bad row then:*

$$\sum_{j \in O_i} \mathbf{y}_j \leq \frac{2qz}{\frac{1}{3} - 2z},$$

$$\sum_{j \in B_i} \mathbf{y}_j \geq \frac{\frac{1}{3} + z - qz - \left(\frac{1}{3} + z\right) \sum_{j \in O_i} \mathbf{y}_j}{\frac{2}{3} - z},$$

$$\sum_{j \in S_i} \mathbf{y}_j \geq \frac{\frac{1}{3} - 2z - qz - \left(\frac{1}{3} + z\right) \sum_{j \in O_i} \mathbf{y}_j}{\frac{2}{3} - z}.$$

7.1 The Strategies

$\mathbf{y}^{\mathrm{imp}}$ **and** $\mathbf{y}(t)$ We now define our improved strategies. Let $\bar{\imath}$ to be a *worst* bad row. That is $\bar{\imath}$ is a row that satisfies $\arg\max_i (R_i \cdot \mathbf{y})$, and therefore $\bar{\imath}$ is a $\bar{q}$-bad row such that there is no q-bad row with $q < \bar{q}$. We fix $\bar{\imath}$ and $\bar{q}$ to be these choices for the rest of this paper. Note that we can assume that $\bar{q} < 3$, because if this is not the case, then all rows have payoff less than or equal to $\frac{2}{3} - z$, and $\mathbf{y}$ does not need to be improved.

We begin by defining a strategy that improves row $\bar{\imath}$. We will improve row $\bar{\imath}$ by moving the probability assigned to $B_{\bar{\imath}}$ to $S_{\bar{\imath}}$. Formally, we define the strategy $\mathbf{y}^{\mathrm{imp}}$ as follows. For each j with $1 \le j \le n$, we have:

$$
\mathbf{y}_j^{\mathrm{imp}} = \begin{cases} 0 & \text{if } j \in B_{\bar{\imath}}, \\ \mathbf{y}_j + \frac{\mathbf{y}_j \cdot \sum_{k \in B_{\bar{\imath}}} \mathbf{y}_k}{\sum_{k \in S_{\bar{\imath}}} \mathbf{y}_k} & \text{if } j \in S_{\bar{\imath}}, \\ \mathbf{y}_j & \text{otherwise.} \end{cases} \tag{6}
$$

The strategy $\mathbf{y}^{\mathrm{imp}}$ improves the specific bad row $\bar{\imath}$, but other rows may not improve, or even get worse in $\mathbf{y}^{\mathrm{imp}}$. Therefore, we will study convex combinations of $\mathbf{y}$ and $\mathbf{y}^{\mathrm{imp}}$. More formally, for the parameter $t \in [0, 1]$, we define the strategy $\mathbf{y}(t)$ to be $(1 - t) \cdot \mathbf{y} + t \cdot \mathbf{y}^{\mathrm{imp}}$.

8 Applying the Matching Pennies Argument

So far, we have not used the assumption that Procedure 2 did not find a $\left(\frac{2}{3} - z\right)$-WSNE. In this section we will see how this assumption can be used to prove properties about the q-bad rows. We begin by defining the concept of a matching pennies sub-game.

Definition 9 (*Matching Pennies*) Let $\mathbf{y}$ be a column player strategy, let i and i' be two rows, and let j and j' be two columns. If $j \in B_i \cap S_{i'}$ and $j' \in B_{i'} \cap S_i$, then we say that $i, i', j,$ and j' form a matching pennies sub-game in $\mathbf{y}$.

An example of a matching pennies sub-game is given by $l, r, T,$ and M in Fig. 2, because we have $l \in B_M \cap S_T$, and we have $r \in B_T \cap S_M$. In this example, we can obtain an exact Nash equilibrium by making the row player mix uniformly between T and M, and making the column player mix uniformly between l and r. However, in general we can only expect to obtain an $\left(\frac{2}{3} - z\right)$-WSNE using this technique, as the following proposition shows.

Proposition 10 *Let $\mathbf{y}$ be a column player strategy. If there is a matching pennies sub-game in $\mathbf{y}$, then we can construct a $\left(\frac{2}{3} - z\right)$-WSNE with a 2×2 support.*

Proof Let $i, i', j,$ and j' be a matching pennies sub-game in $\mathbf{y}$. We define two strategies $\mathbf{x}'$ and $\mathbf{y}'$ as follows:

$$
\mathbf{x}_k' = \begin{cases} 0.5 & \text{if } k = i \text{ or } k = i', \\ 0 & \text{otherwise.} \end{cases} \qquad \mathbf{y}_k' = \begin{cases} 0.5 & \text{if } k = j \text{ or } k = j', \\ 0 & \text{otherwise.} \end{cases}
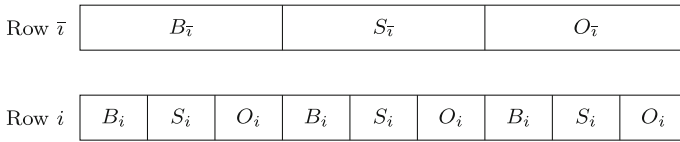$$

Row $\bar{\imath}$	$B_{\bar{\imath}}$			$S_{\bar{\imath}}$			$O_{\bar{\imath}}$		

Row i	B_i	S_i	O_i	B_i	S_i	O_i	B_i	S_i	O_i

Fig. 3 The nine possible intersections between the partition $(B_{\bar{\imath}}, S_{\bar{\imath}}, O_{\bar{\imath}})$, and the partition (B_i, S_i, O_i) for the rows i and $\bar{\imath}$

We will prove that $(\mathbf{x}', \mathbf{y}')$ is a $\left(\frac{2}{3} - z\right)$-WSNE. Note that when the column player plays $\mathbf{y}'$, the payoff to the row player from row i is:

$$R_i \cdot \mathbf{y}' = 0.5 \cdot R_{i,j} + 0.5 \cdot R_{i,j'}.$$

Since $j \in B_i$ we have $R_{i,j} \geq \frac{2}{3} + 2z$, Hence, we have:

$$R_i \cdot \mathbf{y}' \geq 0.5 \times \left(\frac{2}{3} + 2z\right) + 0.5 \times 0 = \frac{1}{3} + z.$$

An identical argument can be used to argue that $R_{i'} \cdot \mathbf{y}'$, $C^T_{\ j} \cdot \mathbf{x}'$, and $C^T_{\ j'} \cdot \mathbf{x}'$ are all greater than or equal to $\frac{1}{3} + z$.

Thus, we have shown that all pure strategies in the support of $\mathbf{x}'$ and $\mathbf{y}'$ are $\left(\frac{2}{3} - z\right)$-best responses. Hence, $(\mathbf{x}', \mathbf{y}')$ is a $\left(\frac{2}{3} - z\right)$-WSNE. $\square$

Proposition 10 allows us to assume that the game does not contain a matching pennies sub-game, because otherwise Procedure 2 would have found a $\left(\frac{2}{3} - z\right)$-WSNE. Note that, by definition, if the game does not contain a matching pennies sub-game, then for all rows i we must have either $B_{\bar{\imath}} \cap S_i = \emptyset$, or $B_i \cap S_{\bar{\imath}} = \emptyset$.

9 An Improved Strategy Exists

Our goal is to show that there exists a t in the range $0 \leq t \leq 1$ and a $z > 0$ such that for every row i, we have $R_i \cdot \mathbf{y}(t) \leq \frac{2}{3} - z$. In this section, we develop a computer-assisted proof of this fact.

Recall that the strategy $\mathbf{y}^{\mathrm{imp}}$ is defined by moving all probability from the columns in $B_{\bar{\imath}}$ to the columns in $S_{\bar{\imath}}$. We are interested in how other rows i are affected by this operation. This will depend on how much probability mass is shared between the partition (B_i, S_i, O_i), and the partition $(B_{\bar{\imath}}, S_{\bar{\imath}}, O_{\bar{\imath}})$. Figure 3 shows the nine possible intersections.

We are interested in the amount of probability that $\mathbf{y}$ assigns to each of these nine sets. We define a shorthand for this purpose:

$$\mathbf{bb} = \sum_{j \in B_{\bar{\imath}} \cap B_i} \mathbf{y}_j, \qquad \mathbf{sb} = \sum_{j \in S_{\bar{\imath}} \cap B_i} \mathbf{y}_j, \qquad \mathbf{ob} = \sum_{j \in O_{\bar{\imath}} \cap B_i} \mathbf{y}_j,$$

$$\mathbf{bs} = \sum_{j \in B_{\bar{\imath}} \cap S_i} \mathbf{y}_j, \qquad \mathbf{ss} = \sum_{j \in S_{\bar{\imath}} \cap S_i} \mathbf{y}_j, \qquad \mathbf{os} = \sum_{j \in O_{\bar{\imath}} \cap S_i} \mathbf{y}_j,$$

$$\mathbf{bo} = \sum_{j \in B_{\bar{\imath}} \cap O_i} \mathbf{y}_j, \qquad \mathbf{so} = \sum_{j \in S_{\bar{\imath}} \cap O_i} \mathbf{y}_j, \qquad \mathbf{oo} = \sum_{j \in O_{\bar{\imath}} \cap O_i} \mathbf{y}_j.$$

As t is increased away from 0, probability will be shifted from **bb**, **bs**, and **bo** to **sb**, **ss**, and **so**, while the amount of probability assigned to **ob**, **os**, and **oo** will remain constant.

For each fixed t in the range $0 \leq t \leq 1$, and each fixed $z > 0$, we are interested in the worst-case value of $R_i \cdot \mathbf{y}(t)$. We will show that an upper bound on $R_i \cdot \mathbf{y}(t)$ can be obtained by solving a linear program. The linear program has eleven variables. We use nine variables, **bb**, **bs**, **bo**, **sb**, **ss**, **so**, **ob**, **os**, and **oo**, to represent the amount of probability assigned to the columns in i. We use two additional variables **q** and **q̄** to represent how bad the rows $\bar{\imath}$ and i are. These two variables should be interpreted as follows: row $\bar{\imath}$ is a **q̄**-bad row and row i is a **q**-bad row.

We can now define the linear program. We begin by defining a helper function $\phi(z, \bar{q})$ as follows:

$$\phi(z, \bar{q}) = 1 + \frac{\frac{1}{3} + z + \bar{q}z}{\frac{1}{3} - 2z - \bar{q}z - \left(\frac{1}{3} + z\right) \cdot \frac{2\bar{q}z}{\frac{1}{3} - 2z}}.$$

9.1 The Linear Program

$LP(z, t, k)$.

   Maximize:

$$(1 - t)\left(\frac{2}{3} + 2z - \mathbf{q} \cdot z\right) + t \cdot \left(\phi(z, 3)\left(\mathbf{sb} + \left(\frac{1}{3} + z\right) \cdot \mathbf{ss} + \left(\frac{2}{3} + 2z\right) \cdot \mathbf{so}\right)\right.$$
$$\left. + \mathbf{ob} + \left(\frac{1}{3} + z\right) \cdot \mathbf{os} + \left(\frac{2}{3} + 2z\right) \cdot \mathbf{oo}\right)$$

Subject to:

$$\mathbf{bb} + \mathbf{bs} + \mathbf{bo} \geq \frac{\frac{1}{3} + z - \bar{\mathbf{q}} \cdot z - \left(\frac{1}{3} + z\right)(\mathbf{ob} + \mathbf{os} + \mathbf{oo})}{\frac{2}{3} - z} \tag{7}$$

$$\mathbf{bb} + \mathbf{sb} + \mathbf{ob} \geq \frac{\frac{1}{3} + z - \mathbf{q} \cdot z - \left(\frac{1}{3} + z\right)(\mathbf{bo} + \mathbf{so} + \mathbf{oo})}{\frac{2}{3} - z} \tag{8}$$

$$\mathbf{sb} + \mathbf{ss} + \mathbf{so} \geq \frac{\frac{1}{3} - 2z - \bar{\mathbf{q}} \cdot z - \left(\frac{1}{3} + z\right)(\mathbf{ob} + \mathbf{os} + \mathbf{oo})}{\frac{2}{3} - z} \tag{9}$$

$$\mathbf{bs} + \mathbf{ss} + \mathbf{os} \geq \frac{\frac{1}{3} - 2z - \mathbf{q} \cdot z - \left(\frac{1}{3} + z\right)(\mathbf{bo} + \mathbf{so} + \mathbf{oo})}{\frac{2}{3} - z} \tag{10}$$

$$\mathbf{ob} + \mathbf{os} + \mathbf{oo} \leq \frac{2 \cdot \bar{\mathbf{q}} \cdot z}{\frac{1}{3} - 2z} \tag{11}$$

$$\mathbf{bo} + \mathbf{so} + \mathbf{oo} \leq \frac{2 \cdot \mathbf{q} \cdot z}{\frac{1}{3} - 2z} \tag{12}$$

$$0 = \begin{cases} \mathbf{bs} & \text{if } k = 0 \\ \mathbf{sb} & \text{if } k = 1 \end{cases} \tag{13}$$

$$\bar{\mathbf{q}} \leq 3 \tag{14}$$

$$\bar{\mathbf{q}} \leq \mathbf{q} \tag{15}$$

$$1 = \mathbf{bb} + \mathbf{bs} + \mathbf{bo} + \mathbf{sb} + \mathbf{ss} + \mathbf{so} + \mathbf{ob} + \mathbf{os} + \mathbf{oo} \tag{16}$$

$$0 \leq \mathbf{bb}, \mathbf{bs}, \mathbf{bo}, \mathbf{sb}, \mathbf{ss}, \mathbf{so}, \mathbf{ob}, \mathbf{os}, \mathbf{oo}, \mathbf{q}, \bar{\mathbf{q}} \tag{17}$$

Our linear program will be parameterised: for each z with $z \geq 0$, each t in the range $0 \leq t \leq 1$, and each $k \in \{0, 1\}$ we define $LP(z, t, k)$ to be the linear program shown here. The rest of this section is dedicated to showing that this linear program can be used to find an upper bound on $R_i \cdot \mathbf{y}(t)$, for all rows i.

9.2 The Constraints

We begin by arguing that all of the constraints in the LP are valid. Firstly, since z and t are both constants, it can be seen that all of the constraints are indeed linear. Constraints (7) through (12) are taken directly from Proposition 8. Each inequality in Proposition 8 appears twice: once for the row i and once for the row $\bar{i}$.

Constraint (13) encodes the matching pennies argument. By Proposition 10 if we have both $\mathbf{bs} > 0$ and $\mathbf{sb} > 0$, then we can find a $\left(\frac{2}{3} - z\right)$-WSNE. Thus, we can assume that either $\mathbf{sb} = 0$ or $\mathbf{sb} = 0$. Constraint (13) encodes this using the parameter k: if $k = 0$ then $\mathbf{bs}$ is constrained to be 0, and if $k = 1$, then $\mathbf{sb}$ is constrained to be 0.

Constraints (14) and (15) provide bounds for $\mathbf{q}$ and $\bar{\mathbf{q}}$. Recall that a row i is q-bad if $R_i \cdot \mathbf{y} = \frac{2}{3} + 2z - qz$. Since $\bar{\mathbf{q}}$ is the q-value for a *worst* bad row, and since a worst bad row $\bar{i}$ must have $R_{\bar{i}} \cdot \mathbf{y} \geq \frac{2}{3} - z$, we must have $\bar{\mathbf{q}} \leq 3$. This is encoded in Constraint (14). Constraint (15) again uses the fact that $\bar{\mathbf{q}}$ is the q-value of a worst bad row: the q value for every other row must be greater than or equal to $\bar{\mathbf{q}}$.

Finally, Constraints (16) and (17) specify that the nine variables must be a probability distribution. They also specify that both $\mathbf{q}$ and $\bar{\mathbf{q}}$ must be non-negative, which is valid because Proposition 6 implies that no row i can have $R_i \cdot \mathbf{y} > \frac{2}{3} + 2z$.

9.3 The Objective

We now show that the objective function of the linear program provides an upper bound on $R_i \cdot \mathbf{y}(t)$. To prove this, we first observe that by definition we have

$$R_i \cdot \mathbf{y}(t) = (1 - t) \cdot R_i \cdot \mathbf{y} + t \cdot R_i \cdot \mathbf{y}^{\text{imp}}. \tag{18}$$

Since i is a $\mathbf{q}$-bad row, we have that $R_i \cdot \mathbf{y} = \frac{2}{3} + 2z - \mathbf{q}z$. In the following proposition, we show an upper bound for $R_i \cdot \mathbf{y}^{\text{imp}}$.

Proposition 11 *We have that $R_i \cdot \mathbf{y}^{\text{imp}}$ is less than or equal to:*

$$\left(1 + \frac{\mathbf{bb} + \mathbf{bs} + \mathbf{bo}}{\mathbf{sb} + \mathbf{ss} + \mathbf{so}}\right)\left(\mathbf{sb} + \left(\frac{1}{3} + z\right) \cdot \mathbf{ss} + \left(\frac{2}{3} + 2z\right) \cdot \mathbf{so}\right)$$
$$+ \mathbf{ob} + \left(\frac{1}{3} + z\right) \cdot \mathbf{os} + \left(\frac{2}{3} + 2z\right) \cdot \mathbf{oo}.$$

Proof Since $\mathbf{y}^{\text{imp}}$ is obtained from $\mathbf{y}$ by shifting all probability from $B_{\bar{i}}$ to $S_{\bar{i}}$, we have that:

$$R_i \cdot \mathbf{y}^{\text{imp}} = \sum_{j \in S_{\bar{i}}} R_{i,j} \cdot \mathbf{y}^{\text{imp}} + \sum_{j \in O_{\bar{i}}} R_{i,j} \cdot \mathbf{y}^{\text{imp}}$$

$$= \sum_{j \in S_{\bar{i}}} R_{i,j} \cdot \mathbf{y}^{\text{imp}} + \sum_{j \in O_{\bar{i}}} R_{i,j} \cdot \mathbf{y}$$

$$= \left(1 + \frac{\mathbf{bb} + \mathbf{bs} + \mathbf{bo}}{\mathbf{sb} + \mathbf{ss} + \mathbf{so}}\right) \sum_{j \in S_{\bar{i}}} R_{i,j} \cdot \mathbf{y} + \sum_{j \in O_{\bar{i}}} R_{i,j} \cdot \mathbf{y}. \qquad (19)$$

The second and third equalities were obtained directly from the definition of $\mathbf{y}^{\text{imp}}$ given in (6).

Now to obtain the claimed result we split the two sums into their constituent parts. Firstly, we have that $\sum_{j \in S_{\bar{i}}} \mathbf{y} = \mathbf{sb} + \mathbf{ss} + \mathbf{so}$, and by definition we have that:

– $R_{i,j} \leq 1$ for each $j \in B_i$,
– $R_{i,j} \leq \frac{1}{3} + z$ for each $j \in S_i$, and
– $R_{i,j} \leq \frac{2}{3} + 2z$ for each $j \in O_i$.

Similarly, we split the sum $\sum_{j \in O_{\bar{i}}} \mathbf{y}$ into $\mathbf{ob} + \mathbf{os} + \mathbf{oo}$, and apply the same bounds as above. Combining all of these bounds and substituting them into Eq. (19) yields the claimed result. $\qquad\square$

Substituting our two bounds into Eq. (18) does give an upper bound on $R_i \cdot \mathbf{y}(t)$, but this upper bound is not linear in the variables of the linear program. To resolve this, in the next proposition we provide a constant upper bound for one of the terms in the LP, using the auxiliary function $\phi(z, \bar{q})$ that was defined earlier.

Proposition 12 *If $z < \frac{13 - 3\sqrt{17}}{24} \approx 0.02627$, then*

$$\left(1 + \frac{\mathbf{bb} + \mathbf{bs} + \mathbf{bo}}{\mathbf{sb} + \mathbf{ss} + \mathbf{so}}\right) \leq \phi(z, 3).$$

Proof Proposition 8 implies that $\mathbf{sb} + \mathbf{ss} + \mathbf{so} \geq \dfrac{\frac{1}{3} - 2z - \bar{q}z - \left(\frac{1}{3} + z\right)(\mathbf{ob} + \mathbf{os} + \mathbf{oo})}{\frac{2}{3} - z}$. We can apply this in order to determine the following upper bound for $\mathbf{bb} + \mathbf{bs} + \mathbf{bo}$.

$$\mathbf{bb} + \mathbf{bs} + \mathbf{bo} = 1 - (\mathbf{sb} + \mathbf{ss} + \mathbf{so}) - (\mathbf{ob} + \mathbf{os} + \mathbf{oo})$$

$$\leq 1 - \frac{\frac{1}{3} - 2z - \bar{q}z - \left(\frac{1}{3} + z\right)(\mathbf{ob} + \mathbf{os} + \mathbf{oo})}{\frac{2}{3} - z} - (\mathbf{ob} + \mathbf{os} + \mathbf{oo})$$

$$= 1 - \frac{\frac{1}{3} - 2z - \bar{q}z + \left(-\frac{1}{3} - z + \frac{2}{3} - z\right)(\mathbf{ob} + \mathbf{os} + \mathbf{oo})}{\frac{2}{3} - z}$$

$$= 1 - \frac{\frac{1}{3} - 2z - \bar{q}z + \left(\frac{1}{3} - 2z\right)(\mathbf{ob} + \mathbf{os} + \mathbf{oo})}{\frac{2}{3} - z}$$

$$= \frac{\frac{1}{3} + z + \bar{q}z - \left(\frac{1}{3} - 2z\right)(\mathbf{ob} + \mathbf{os} + \mathbf{oo})}{\frac{2}{3} - z}$$

$$\leq \frac{\frac{1}{3} + z + \bar{q}z}{\frac{2}{3} - z}.$$

Substituting this gives the following upper bound.

$$\left(1 + \frac{\mathbf{bb} + \mathbf{bs} + \mathbf{bo}}{\mathbf{sb} + \mathbf{ss} + \mathbf{so}}\right) \leq \left(1 + \frac{\frac{1}{3} + z + \bar{q}z}{\left(\frac{2}{3} - z\right) \cdot (\mathbf{sb} + \mathbf{ss} + \mathbf{so})}\right). \tag{20}$$

In order to proceed we must now use a lower bound for $\left(\frac{2}{3} - z\right) \cdot (\mathbf{sb} + \mathbf{ss} + \mathbf{so})$. By Proposition 8 we have that:

$$\left(\frac{2}{3} - z\right) \cdot (\mathbf{sb} + \mathbf{ss} + \mathbf{so}) \geq \frac{1}{3} - 2z - \bar{q}z - \left(\frac{1}{3} + z\right)(\mathbf{ob} + \mathbf{os} + \mathbf{oo})$$

$$\geq \frac{1}{3} - 2z - \bar{q}z - \left(\frac{1}{3} + z\right) \cdot \frac{2\bar{q}z}{\frac{1}{3} - 2z}. \tag{21}$$

In order to substitute Inequality (21) into Inequality (20), we must have that $2z + \bar{q}z + \left(\frac{1}{3} + z\right) \cdot \frac{2\bar{q}z}{\frac{1}{3} - 2z} < \frac{1}{3}$, because otherwise the denominator of Inequality (20) will be 0 or negative. Since $\bar{q}$ can be at most 3, this holds whenever:

$$5z + \left(\frac{1}{3} + z\right) \cdot \frac{6z}{\frac{1}{3} - 2z} < \frac{1}{3}.$$

Solving this inequality for z gives that $z < \frac{13 \pm 3\sqrt{17}}{24}$. Taking the smaller of the two solutions gives $z < \frac{13 - 3\sqrt{17}}{24} \approx 0.02627$.

So, if we have $z < \frac{13 - 3\sqrt{17}}{24}$, then we can conclude:

$$\left(1 + \frac{\mathbf{bb} + \mathbf{bs} + \mathbf{bo}}{\mathbf{sb} + \mathbf{ss} + \mathbf{so}}\right) \leq \left(1 + \frac{\frac{1}{3} + z + \bar{q}z}{\frac{1}{3} - 2z - \bar{q}z - \left(\frac{1}{3} + z\right) \cdot \frac{2\bar{q}z}{\frac{1}{3} - 2z}}\right)$$

$$= \phi(z, \bar{q}).$$

To complete the proof we observe that, so long as $0 \leq z < \frac{13-3\sqrt{17}}{24}$, we have that $\phi(z, \bar{q})$ is monotonically increasing in $\bar{q}$. This holds because $\bar{q}$ only occurs positively in the numerator and negatively in the denominator, and because the denominator is strictly positive. Thus, since $\bar{q}$ can be at most 3, we have $\phi(z, \bar{q}) \leq \phi(z, 3)$. □

Combining the upper bound from Proposition 12 with the upper bound from Proposition 11, and substituting the result into Eq. (18) gives a linear upper bound on $R_i \cdot \mathbf{y}(t)$, and this linear bound is used as the objective function of the LP.

9.4 The Upper Bound

We can now prove that the linear program provides an upper bound on the quality of WSNE provided by $\mathbf{y}(t)$. For each problem $LP(z, t, k)$, let $Sol(LP(z, t, k))$ be the value of the objective function in the solution of $LP(z, t, k)$.

Proposition 13 *For every $z > 0$ and t in the range $0 \leq t \leq 1$:*

- *if $\sum_{j \in B_{\bar{i}} \cap S_i} \mathbf{y} = 0$ then $R_i \cdot \mathbf{y}(t) \leq Sol(LP(z, t, 0))$.*
- *if $\sum_{j \in S_{\bar{i}} \cap B_i} \mathbf{y} = 0$ then $R_i \cdot \mathbf{y}(t) \leq Sol(LP(z, t, 1))$.*

Proof We will prove only the case where $\sum_{j \in B_{\bar{i}} \cap S_i} \mathbf{y} = 0$, because the other case is entirely symmetric. Let i be a row that maximizes $R_i \cdot \mathbf{y}(t)$, and let $\bar{i}$ be the worst bad row in $\mathbf{y}$. It is not difficult to construct a feasible point in $LP(z, t, 0)$ that represents these two rows: the variables $\mathbf{bb}$, $\mathbf{bs}$, ... are set according to the probability assigned to the corresponding intersection sets by $\mathbf{y}$, while $\mathbf{q}$ and $\bar{\mathbf{q}}$ are set to be the q-values of i and $\bar{i}$, respectively.

Obviously, this point satisfies $\bar{\mathbf{q}} \leq 3$ and $\bar{\mathbf{q}} \leq 3$, and it also satisfies the non-negativities and the sum-to-one constraint. Furthermore, by assumption we have that $\mathbf{bs} = 0$, so Constraint (13) is satisfied. Since all other constraints of the LP were derived from the properties of either $\mathbf{q}$-bad or $\bar{\mathbf{q}}$-bad rows, we have that the point is feasible in $LP(z, t, 0)$.

Since Propositions 11 and 12 show that the objective function of the LP provides an upper bound on $R_i \cdot \mathbf{y}(y)$, and since the LP is a maximization problem, we must have $R_i \cdot \mathbf{y}(t) \leq Sol(LP(z, t, 0))$. □

9.5 Finding

z We now describe how the linear programs can be used to determine a value of z such that $\mathbf{y}(t)$ is a $\left(\frac{2}{3} - z\right)$-WSNE. For every $z > 0$, if we want to prove that we can produce a $\left(\frac{2}{3} - z\right)$-WSNE, we require a witness (z, t_0, t_1) that satisfies both of the following conditions:

- We have that t_0 is in the range $0 \leq t_0 \leq 1$ and that $Sol(z, t_0, 0) \leq \frac{2}{3} - z$.
- We have that t_1 is in the range $0 \leq t_1 \leq 1$ and that $Sol(z, t_1, 1) \leq \frac{2}{3} - z$.

If a pair (t_0, t_1) can be found that satisfy these properties, then $\mathbf{y}(t_0)$ is a $\left(\frac{2}{3} - z\right)$-WSNE in the case where $\mathbf{bs} = 0$, and $\mathbf{y}(t_1)$ is a $\left(\frac{2}{3} - z\right)$-WSNE in the case where $\mathbf{sb} = 0$.

Our strategy for finding a witness (z, t_0, t_1) was to perform a grid search over all possible values for z, t_0, and t_1 using a suitably small increment. We implemented this approach in Mathematica[2], where for each candidate witness, we solved the two linear programs in exact arithmetic. Ultimately, we were able to find a witness $(0.005913759, 0.120, 0.168)$, and we were unable to find a witness for $z = 0.005913760$. Thus, we have completed the proof of Theorem 4.

10 Conclusion

We have shown that our algorithm always finds a $\left(\frac{2}{3} - 0.005913759\right)$-WSNE. Our computer assisted proof relied upon a linear program. We tried several ways to improve this analysis, all of which were ultimately unsuccessful.

The current proof finds two values of t: one for the case where $\mathbf{bb} = 0$, and one for the case where $\mathbf{sb} = 0$. One obvious approach towards improving the analysis is to split the analysis into more cases, and compute a t for each case. One of our unsuccessful attempts in this direction was to parameterise the LP for different values of $\bar{\mathbf{q}}$. The existing LP allows $\bar{\mathbf{q}}$ to take any value in the range $[0, 3]$, but we could, for example, use one LP for the case where $\bar{\mathbf{q}} \in [0, 1.5]$ and another for the case where $\bar{\mathbf{q}} \in [1.5, 3]$, and then compute two different values of t for these two cases. Unfortunately, this did not yield a better analysis no matter how many different bands we used.

The objective function of the LP uses a linear upper bound on the non-linear expression from Proposition 11. We could, in principle, attempt to solve the non-linear optimization problem that is obtained when the expression from Proposition 11 is used directly as the objective function. Unfortunately, it seems that this task is beyond current technology. In particular, the need to solve the problem in exact arithmetic thwarted our attempts to solve the problem within a reasonable running time.

Appendix: Proof of Proposition 8

We begin by proving the inequality for O_i. The first thing that we note is that, if a column j is in O_i, then $R_{i,j} + C_{i,j}$ must be significantly smaller than $\frac{4}{3} + z$.

Proposition 14 *For each row i, and each column $j \in O_i$, we have $R_{i,j} + C_{i,j} < 1 + 3z$.*

Proof For each column $j \in O_i$ we have both of the following properties:

- Since $j \notin B_i$, we have $R_{i,j} < \frac{2}{3} + 2z$.
- Since $j \notin S_i$, we have $C_{i,j} < \frac{2}{3} + 2z$.

Furthermore, our assumption that Procedure (1) does not find a pure $\left(\frac{2}{3} - z\right)$-WSNE implies that:

- If $R_{i,j} \geq \frac{1}{3} + z$, then $C_{i,j} < \frac{1}{3} + z$.
- If $C_{i,j} \geq \frac{1}{3} + z$, then $R_{i,j} < \frac{1}{3} + z$.

This is because, if these inequalities did not hold for some pair i and j, then it is easy to show that (i, j) is a $(\frac{2}{3} - z)$-WSNE. From these properties it is easy to see that $R_{i,j} + C_{i,j} < \frac{2}{3} + 2z + \frac{1}{3} + z = 1 + 3z$. □

We now use this proposition, along with Markov's inequality, to prove the bound for O_i specified in Proposition 8.

Proposition 15 *If i is a q-bad row, then $\sum_{j \in O_i} \mathbf{y}_j \leq \frac{2qz}{\frac{1}{3} - 2z}$.*

Proof Consider the random variable $T = \frac{4}{3} + z - R_{i,j} - C_{i,j}$, where i is fixed and j is sampled from $\mathbf{y}$. From Eq. (3), we have that T takes values in the range $[0, \frac{4}{3} + z]$. Utilizing Definition 7 along with Eq. (5) gives the following:

$$R_i \mathbf{y} + C_i \mathbf{y} \geq \frac{4}{3} + (1 - 2q)z.$$

Therefore, we have the following expression for the expectation of T:

$$\begin{aligned} E[T] &= \frac{4}{3} + z - E_{j \sim \mathbf{y}}[R_{i,j} + C_{i,j}] \\ &\leq \frac{4}{3} + z - \left(\frac{4}{3} + (1 - 2q)z \right) = 2qz \end{aligned}$$

By Proposition 14, for each $j \in O_i$, we have $R_{i,j} + C_{i,j} \leq 1 + 3z$. Hence, we have $T \geq \frac{4}{3} + z - (1 + 3z) = \frac{1}{3} - 2z$ for each $j \in O_i$. Therefore, we must have $\Pr(T \geq \frac{1}{3} - 2z) \geq \sum_{j \in O_i} \mathbf{y}_j$. Applying Markov's inequality completes the proof:

$$\Pr \left(T \geq \frac{1}{3} - 2z \right) \leq \frac{E[T]}{\frac{1}{3} - 2z} \leq \frac{2qz}{\frac{1}{3} - 2z}.$$

□

Now we prove the inequality that was given for B_i in Proposition 8.

Proposition 16 *If i is a q-bad row, then $\sum_{j \in B_i} \mathbf{y}_j \geq \frac{\frac{1}{3} + z - qz - \left(\frac{1}{3} + z \right) \sum_{j \in O_i} \mathbf{y}_j}{\frac{2}{3} - z}$.*

Proof Since the sets B_i, S_i, and O_i are disjoint, we can write Definition 7 as:

$$\sum_{j \in B_i} \mathbf{y}_j R_{i,j} + \sum_{j \in S_i} \mathbf{y}_j R_{i,j} + \sum_{j \in O_i} \mathbf{y}_j R_{i,j} \geq \frac{2}{3} + 2z - qz.$$

We know that $R_{i,j} \leq 1$ for each $j \in B_i$, that $R_{i,j} \leq \frac{2}{3} + 2z$ for each $j \in O_i$, and that $R_{i,j} \leq \frac{1}{3} + z$ for each $j \in S_i$. Therefore we obtain the following inequality:

$$1 \cdot \sum_{j \in B_i} \mathbf{y}_j + \left(\frac{1}{3} + z \right) \cdot \sum_{j \in S_i} \mathbf{y}_j + \left(\frac{2}{3} + 2z \right) \cdot \sum_{j \in O_i} \mathbf{y}_j \geq \frac{2}{3} + 2z - qz.$$

Furthermore, since $\sum_{j \in S_i} \mathbf{y}_j = 1 - \sum_{j \in B_i} \mathbf{y}_j - \sum_{j \in O_i} \mathbf{y}_j$, we have:

$$\sum_{j \in B_i} \mathbf{y}_j + \left(\frac{1}{3} + z\right) \cdot \left(1 - \sum_{j \in B_i} \mathbf{y}_j - \sum_{j \in O_i} \mathbf{y}_j\right) + \left(\frac{2}{3} + 2z\right) \sum_{j \in O_i} \mathbf{y}_j \geq \frac{2}{3} + 2z - qz.$$

Rearranging this gives:

$$\left(\frac{2}{3} - z\right) \cdot \sum_{j \in B_i} \mathbf{y}_j \geq \frac{1}{3} + z - qz - \left(\frac{1}{3} + z\right) \sum_{j \in O_i} \mathbf{y}_j.$$

Finally, this allows us to conclude that:

$$\sum_{j \in B_i} \mathbf{y}_j \geq \frac{\frac{1}{3} + z - qz - \left(\frac{1}{3} + z\right) \sum_{j \in O_i} \mathbf{y}_j}{\frac{2}{3} - z}.$$

$\square$

Finally, we prove the inequality that was given for S_j in Proposition 8. This proof is very similar to the proof of Proposition 16, except that we substitute into Eq. (5) rather than Definition 7.

Proposition 17 *If i is a q-bad row then* $\sum_{j \in S_i} \mathbf{y}_j \geq \frac{\frac{1}{3} - 2z - qz - \left(\frac{1}{3} + z\right) \sum_{j \in O_i} \mathbf{y}_j}{\frac{2}{3} - z}$.

Proof Since the sets B_i, S_i, and O_i are disjoint, we can rewrite Eq. (5) as:

$$\sum_{j \in B_i} \mathbf{y}_j C_{i,j} + \sum_{j \in S_i} \mathbf{y}_j C_{i,j} + \sum_{j \in O_i} \mathbf{y}_j C_{i,j} \geq \frac{2}{3} - z - qz.$$

We know that $C_{i,j} \leq 1$ for each $j \in S_i$, that $C_{i,j} \leq \frac{2}{3} + 2z$ for each $j \in O_i$, and that $C_{i,j} \leq \frac{1}{3} + z$ for each $j \in B_i$. Therefore we obtain the following inequality:

$$1 \cdot \sum_{j \in S_i} \mathbf{y}_j + \left(\frac{1}{3} + z\right) \cdot \sum_{j \in B_i} \mathbf{y}_j + \left(\frac{2}{3} + 2z\right) \cdot \sum_{j \in O_i} \mathbf{y}_j \geq \frac{2}{3} - z - qz.$$

Furthermore, since $\sum_{j \in B_i} \mathbf{y}_j = 1 - \sum_{j \in S_i} \mathbf{y}_j - \sum_{j \in O_i} \mathbf{y}_j$, we have:

$$\sum_{j \in S_i} \mathbf{y}_j + \left(\frac{1}{3} + z\right) \cdot \left(1 - \sum_{j \in S_i} \mathbf{y}_j - \sum_{j \in O_i} \mathbf{y}_j\right) + \left(\frac{2}{3} + 2z\right) \sum_{j \in O_i} \mathbf{y}_j \geq \frac{2}{3} - z - qz.$$

Rearranging this gives:

$$\left(\frac{2}{3} - z\right) \cdot \sum_{j \in S_i} \mathbf{y}_j \geq \frac{1}{3} - 2z - qz - \left(\frac{1}{3} + z\right) \sum_{j \in O_i} \mathbf{y}_j.$$

Finally, this allows us to conclude that:

$$\sum_{j \in S_i} \mathbf{y}_j \geq \frac{\frac{1}{3} - 2z - qz - \left(\frac{1}{3} + z\right) \sum_{j \in O_i} \mathbf{y}_j}{\frac{2}{3} - z}.$$

$\square$

Now that we have shown Propositions 15, 16, and 17, we have completed the proof of Proposition 8

References

1. Anbalagan, Y., Norin, S., Savani, R., Vetta, A.: Polylogarithmic supports are required for approximate well-supported Nash equilibria below 2/3. In: Proceedings of WINE, pp. 15–23 (2013)
2. Bosse, H., Byrka, J., Markakis, E.: New algorithms for approximate Nash equilibria in bimatrix games. Theoret. Comput. Sci. **411**(1), 164–173 (2010)
3. Chen, X., Deng, X., Teng, S.H.: Settling the complexity of computing two-player Nash equilibria. J. ACM **56**(3), 14:1–14:57 (2009)
4. Czumaj, A., Fasoulakis, M., Jurdziński, M.: Approximate well-supported Nash equilibria in symmetric bimatrix games. In: Proceedings of SAGT, pp. 244–254 (2014)
5. Daskalakis, C., Goldberg, P.W., Papadimitriou, C.H.: The complexity of computing a Nash equilibrium. SIAM J. Comput. **39**(1), 195–259 (2009)
6. Daskalakis, C., Mehta, A., Papadimitriou, C.H.: Progress in approximate Nash equilibria. In: Proceedings of ACM-EC, pp. 355–358 (2007)
7. Daskalakis, C., Mehta, A., Papadimitriou, C.H.: A note on approximate Nash equilibria. Theoret. Comput. Sci. **410**(17), 1581–1588 (2009)
8. Fearnley, J., Goldberg, P.W., Savani, R., Sørensen, T.B.: Approximate well-supported Nash equilibria below two-thirds. In: Proceedings of SAGT, pp. 108–119 (2012)
9. Fearnley, J., Igwe, T.P., Savani, R.: An empirical study of finding approximate equilibria in bimatrix games. In: Proceedings of SEA, pp. 339–351 (2015)
10. Kontogiannis, S.C., Spirakis, P.G.: Efficient algorithms for constant well supported approximate equilibria in bimatrix games. In: Proceedings of ICALP, pp. 595–606 (2007)
11. Kontogiannis, S.C., Spirakis, P.G.: Well supported approximate equilibria in bimatrix games. Algorithmica **57**(4), 653–667 (2010)
12. Nash, J.: Non-cooperative games. Ann. Math. **54**(2), 286–295 (1951)
13. Tsaknakis, H., Spirakis, P.G.: An optimization approach for approximate Nash equilibria. Intern. Math. **5**(4), 365–382 (2008)