Distributed Methods for Finding Approximate Equilibria

Artur Czumaj[1] Argyrios Deligkas[2] Michail Fasoulakis[1]
John Fearnley[2] Marcin Jurdziński[1] Rahul Savani[2]

[1]University of Warwick

[2]University of Liverpool

Bimatrix games

I \ II	L	R
t	0 1	1 0
b	1 0	0 1

Mixed strategy: probability distribution over actions

Best response: action that maximizes payoff

Regret: best response payoff - expected payoff

Bimatrix games

I \ II	L	R
t	0 1	1 0
b	1 0	0 1

Nash equilibrium: A pair of mixed strategies where

- Both players have regret zero
- Both players only play best responses

eg. row player plays $(0.5, 0.5)$ and column player plays $(0.5, 0.5)$

Bimatrix games

I \ II	L	R
t	0 1	1 0
b	1 0	0 1

ϵ-Approximate Nash equilibrium (ϵ-NE)

- Nash: Both players have regret 0
- ϵ-NE: Both players have regret at most ϵ

eg. row player plays $(0.5, 0.5)$ and column player plays R

Bimatrix games

I \ II	L	R
t	0 1	1 0
b	1 0	0 1

ϵ-Approximate Well Supported Nash equilibrium (ϵ-WSNE)

- Nash: Both players only play best responses
- ϵ-WSNE: Both players only play ϵ-best responses

eg. row player plays $(0.4, 0.6)$ and column player plays $(0.6, 0.4)$

Bimatrix games

I \ II	L	R
t	0 1	1 0
b	1 0	0 1

Goal: Design an algorithm

- Finds an ϵ-NE or ϵ-WSNE
- Runs in polynomial time

Previous work

ϵ-NE

- 0.75 (Kontogiannis, Panagopoulou, and Spirakis, 2006)
- 0.5 (Daskalakis, Mehta, and Papadimitriou, 2006)
- 0.38 (Kontogiannis, Panagopoulou, and Spirakis, 2007)
- 0.36 (Bosse, Byrka, and Markakis, 2007)
- 0.3393 (Tsaknakis and Spirakis, 2007)

Previous work

ϵ-NE

- 0.75 (Kontogiannis, Panagopoulou, and Spirakis, 2006)
- 0.5 (Daskalakis, Mehta, and Papadimitriou, 2006)
- 0.38 (Kontogiannis, Panagopoulou, and Spirakis, 2007)
- 0.36 (Bosse, Byrka, and Markakis, 2007)
- 0.3393 (Tsaknakis and Spirakis, 2007)

ϵ-WSNE

- 0.6667 (Kontogiannis and Spirakis, 2007)
- 0.6608 (Fearnley, Goldberg, Savani, and Sørensen, 2012)

Previous work

ϵ-NE

- 0.75 (Kontogiannis, Panagopoulou, and Spirakis, 2006)
- 0.5 (Daskalakis, Mehta, and Papadimitriou, 2006)
- 0.38 (Kontogiannis, Panagopoulou, and Spirakis, 2007)
- 0.36 (Bosse, Byrka, and Markakis, 2007)
- 0.3393 (Tsaknakis and Spirakis, 2007)

ϵ-WSNE

- 0.6667 (Kontogiannis and Spirakis, 2007)
- 0.6608 (Fearnley, Goldberg, Savani, and Sørensen, 2012)
- 0.6528 (CDFFJS, 2015)

Our results

We give a polynomial time algorithm for finding a 0.6528-WSNE

Communication complexity

- 0.6528-WSNE with polylogarithmic communication
- 0.3820-NE with polylogarithmic communication
- 0.5-WSNE with polylogarithmic communication in win-lose games

Query complexity

- 0.6528-WSNE with $n \cdot \log n$ queries

Outline

1. WSNE results
   - A simple algorithm to get a 2/3-WSNE
   - An improved algorithm to get a 0.6528-WSNE

2. Communication complexity results
   - Communication efficient algorithm for 0.6528-WSNE

3. Query complexity results
   - Query efficient algorithm for 0.6528-WSNE

An algorithm for 2/3-WSNE

1. Solve the zero-sum games $(R, -R)$ and $(-C, C)$.
   - $(\mathbf{x}^*, \mathbf{y}^*)$ is a NE of $(R, -R)$ that secures v_R
   - $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ is a NE of $(-C, C)$ that secures v_C
   - Assume that $v_R \geq v_C$
   - If $v_R \leq 2/3$, return $(\hat{\mathbf{x}}, \mathbf{y}^*)$

An algorithm for 2/3-WSNE

1. Solve the zero-sum games $(R, -R)$ and $(-C, C)$.
   - $(\mathbf{x}^*, \mathbf{y}^*)$ is a NE of $(R, -R)$ that secures v_R
   - $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ is a NE of $(-C, C)$ that secures v_C
   - Assume that $v_R \geq v_C$
   - If $v_R \leq 2/3$, return $(\hat{\mathbf{x}}, \mathbf{y}^*)$

2. If all columns have payoff $\leq 2/3$ against $\mathbf{x}^*$ return $(\mathbf{x}^*, \mathbf{y}^*)$

An algorithm for 2/3-WSNE

1. Solve the zero-sum games $(R, -R)$ and $(-C, C)$.
   - $(\mathbf{x}^*, \mathbf{y}^*)$ is a NE of $(R, -R)$ that secures v_R
   - $(\hat{\mathbf{x}}\ , \hat{\mathbf{y}}\ )$ is a NE of $(-C, C)$ that secures v_C
   - Assume that $v_R \geq v_C$
   - If $v_R \leq 2/3$, return $(\hat{\mathbf{x}}, \mathbf{y}^*)$

2. If all columns have payoff $\leq 2/3$ against $\mathbf{x}^*$ return $(\mathbf{x}^*, \mathbf{y}^*)$

3. Otherwise:
   - Let j^* be a pure best response to $\mathbf{x}^*$.
   - Find a row i such that $R_{i\mathrm{j}^*} \geq 1/3$ and $C_{i\mathrm{j}^*} \geq 1/3$.
   - Return (i, j^*).

An algorithm for 2/3-WSNE

1. Solve the zero-sum games $(R, -R)$ and $(-C, C)$.
   - $(\mathbf{x}^*, \mathbf{y}^*)$ is a NE of $(R, -R)$ that secures v_R
   - $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ is a NE of $(-C, C)$ that secures v_C
   - Assume that $v_R \geq v_C$
   - If $v_R \leq 2/3$, return $(\hat{\mathbf{x}}, \mathbf{y}^*)$
2. If all columns have payoff $\leq 2/3$ against $\mathbf{x}^*$ return $(\mathbf{x}^*, \mathbf{y}^*)$
3. Otherwise:
   - Let j^* be a pure best response to $\mathbf{x}^*$.
   - Find a row i such that $R_{i\mathrm{j}^*} \geq 1/3$ and $C_{i\mathrm{j}^*} \geq 1/3$.
   - Return (i, j^*).

Row i exists because:

- $\mathbf{x}^*$ secures payoff $> \frac{2}{3}$ against j^*
- j^* secures payoff $> \frac{2}{3}$ against $\mathbf{x}^*$

An algorithm for 2/3-WSNE

1. Solve the zero-sum games $(R, -R)$ and $(-C, C)$.
   - $(\mathbf{x}^*, \mathbf{y}^*)$ is a NE of $(R, -R)$ that secures v_R
   - $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ is a NE of $(-C, C)$ that secures v_C
   - Assume that $v_R \geq v_C$
   - If $v_R \leq 2/3$, return $(\hat{\mathbf{x}}, \mathbf{y}^*)$
2. If all columns have payoff $\leq 2/3$ against $\mathbf{x}^*$ return $(\mathbf{x}^*, \mathbf{y}^*)$
3. Otherwise:
   - Let j^* be a pure best response to $\mathbf{x}^*$.
   - Find a row i such that $R_{i\mathrm{j}^*} \geq 1/3$ and $C_{i\mathrm{j}^*} \geq 1/3$.
   - Return (i, j^*).

Theorem: This algorithm finds a 2/3-WSNE

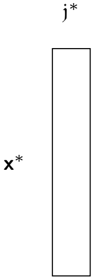
An algorithm for 2/3-WSNE

1. Solve the zero-sum games $(R, -R)$ and $(-C, C)$.
   - $(\mathbf{x}^*, \mathbf{y}^*)$ is a NE of $(R, -R)$ that secures v_R
   - $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ is a NE of $(-C, C)$ that secures v_C
   - Assume that $v_R \geq v_C$
   - If $v_R \leq 2/3$, return $(\hat{\mathbf{x}}, \mathbf{y}^*)$
2. If all columns have payoff $\leq 2/3$ against $\mathbf{x}^*$ return $(\mathbf{x}^*, \mathbf{y}^*)$
3. Otherwise:
   - Let j^* be a pure best response to $\mathbf{x}^*$.
   - Find a row i such that $R_{i\mathrm{j}^*} \geq 1/3$ and $C_{i\mathrm{j}^*} \geq 1/3$.
   - Return (i, j^*).

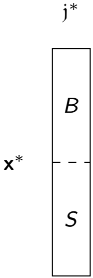
Theorem: This algorithm finds a 2/3-WSNE

We will improve it by replacing Step 3

Improve the algorithm to get a $< 2/3$-WSNE



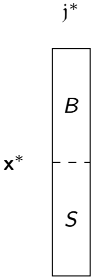
Improve the algorithm to get a $< 2/3$-WSNE



Idea: Shift probabilities

- B - set of rows with payoff $> 1/3$
- S - set of rows with payoff $\leq 1/3$
- Shift all probability from S to B

Improve the algorithm to get a $< 2/3$-WSNE

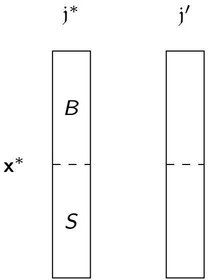


Idea: Shift probabilities

- B - set of rows with payoff $> 1/3$
- S - set of rows with payoff $\leq 1/3$
- Shift all probability from S to B

Row player will be happy, column player...

Improve the algorithm to get a $< 2/3$-WSNE



Initially, j^* is a best response, as probability is shifted:

- Payoff of j^* may fall
- Payoff of another row j' may rise
- We may not have a $< 2/3$-WSNE...

Improve the algorithm to get a $< 2/3$-WSNE

This can only happen if the game looks like this:

I \ II	j^*	j'
B	$\approx \frac{1}{3}$ ≈ 1	≈ 1 $\approx \frac{1}{3}$
S	≈ 1 $\approx \frac{1}{3}$	$\approx \frac{1}{3}$ ≈ 1

Improve the algorithm to get a $< 2/3$-WSNE

This can only happen if the game looks like this:

I \ II	j^*	j'
B	≈ 1, $\approx \frac{1}{3}$	$\approx \frac{1}{3}$, ≈ 1
S	$\approx \frac{1}{3}$, ≈ 1	≈ 1, $\approx \frac{1}{3}$

Because:

- $\mathbf{x}^*$ secures payoff $\geq 2/3$ against j^*
- $\mathbf{x}^*$ secures payoff $\geq 2/3$ against j'
- j^* has payoff $\geq 2/3$ against $\mathbf{x}^*$, and $< 1/3$ against B
- The payoff of j' is $\geq 2/3$ more than the payoff of j^* against B
- The game contains no pure $2/3$-WSNE

Improve the algorithm to get a $< 2/3$-WSNE

This can only happen if the game looks like this:

I \ II	j^*	j'
B	$\approx \frac{1}{3}$ ≈ 1	≈ 1 $\approx \frac{1}{3}$
S	≈ 1 $\approx \frac{1}{3}$	$\approx \frac{1}{3}$ ≈ 1

In fact we can select two rows b and s with:

I \ II	j^*	j'
b	0 ≈ 1	≈ 1 0
s	≈ 1 0	0 ≈ 1

When both players mix uniformly this is a 0.5-WSNE

The algorithm

1. Solve the zero-sum games $(R, -R)$ and $(-C, C)$.
   - $(\mathbf{x}^*, \mathbf{y}^*)$ is a NE of $(R, -R)$ that secures v_R
   - $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ is a NE of $(C, -C)$ that secures v_C
   - Assume that $v_R \geq v_C$
   - If $v_R \leq 2/3 - z$, return $(\hat{\mathbf{x}}, \mathbf{y}^*)$

2. If all columns have payoff $\leq 2/3 - z$ against $\mathbf{x}^*$, return $(\mathbf{x}^*, \mathbf{y}^*)$

The algorithm

1. Solve the zero-sum games $(R, -R)$ and $(-C, C)$.
   - $(\mathbf{x}^*, \mathbf{y}^*)$ is a NE of $(R, -R)$ that secures v_R
   - $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ is a NE of $(C, -C)$ that secures v_C
   - Assume that $v_R \geq v_C$
   - If $v_R \leq 2/3 - z$, return $(\hat{\mathbf{x}}, \mathbf{y}^*)$

2. If all columns have payoff $\leq 2/3 - z$ against $\mathbf{x}^*$, return $(\mathbf{x}^*, \mathbf{y}^*)$

3. Otherwise:
   - Let j^* be a pure best response to $\mathbf{x}^*$.
   - Shift probability from the rows of j^* with payoff $< 1/3 + \mathrm{z}$
   - If the result is a (2/3 - z)-WSNE, return it

The algorithm

1. Solve the zero-sum games $(R, -R)$ and $(-C, C)$.
   - $(\mathbf{x}^*, \mathbf{y}^*)$ is a NE of $(R, -R)$ that secures v_R
   - $(\hat{\mathbf{x}}\ , \hat{\mathbf{y}}\ )$ is a NE of $(C, -C)$ that secures v_C
   - Assume that $v_R \geq v_C$
   - If $v_R \leq 2/3 - z$, return $(\hat{\mathbf{x}}, \mathbf{y}^*)$

2. If all columns have payoff $\leq 2/3 - z$ against $\mathbf{x}^*$, return $(\mathbf{x}^*, \mathbf{y}^*)$

3. Otherwise:
   - Let j^* be a pure best response to $\mathbf{x}^*$.
   - Shift probability from the rows of j^* with payoff $< 1/3 + \mathrm{z}$
   - If the result is a (2/3 - z)-WSNE, return it

4. Otherwise
   - Check that the game has no pure $(2/3 - z)$-WSNE
   - Find a 2×2 matching pennies subgame and return it
   - Return a strategy profile where both players mix uniformly

Analysis

What is the best z we can acheive?

- Steps 1 - 3 return a (2/3 - z)-WSNE
- Step 4 returns a

$$(1 - \frac{1 - 39z + 360z^2}{2 - 33z - 117z^2})\text{-WSNE}$$

Analysis

What is the best z we can acheive?

- Steps 1 - 3 return a (2/3 - z)-WSNE
- Step 4 returns a

$$(1 - \frac{1 - 39z + 360z^2}{2 - 33z - 117z^2})\text{-WSNE}$$

So we want the largest value of z such that

$$1 - \frac{1 - 39z + 360z^2}{2 - 33z - 117z^2} \leq \frac{2}{3} - z$$

Analysis

$$z = \frac{1}{117}\sqrt{3}\Bigg(\sqrt{2434}\sqrt{3}\cos\left(\frac{1}{3}\arctan\left(\frac{39}{240073}\sqrt{9749}\sqrt{3}\right)\right)$$
$$-3\sqrt{2434}\sin\left(\frac{1}{3}\arctan\left(\frac{39}{240073}\sqrt{9749}\sqrt{3}\right)\right)-48\sqrt{3}\Bigg)$$

Analysis

$$z = \frac{1}{117}\sqrt{3}\Bigg(\sqrt{2434}\sqrt{3}\cos\left(\frac{1}{3}\arctan\left(\frac{39}{240073}\sqrt{9749}\sqrt{3}\right)\right)$$
$$- 3\sqrt{2434}\sin\left(\frac{1}{3}\arctan\left(\frac{39}{240073}\sqrt{9749}\sqrt{3}\right)\right) - 48\sqrt{3}\Bigg)$$

$z \approx 0.013906376$

So we get a 0.6528-WSNE

Outline

1. WSNE results
   - A simple algorithm to get a 2/3-WSNE
   - An improved algorithm to get a 0.6528-WSNE

2. Communication complexity results
   - Communication efficient algorithm for 0.6528-WSNE

3. Query complexity results
   - Query efficient algorithm for 0.6528-WSNE

Communication Complexity

The setup:

- Both players know their own payoff matrix
- They do not know their opponent's
- They can communicate with each other
- They eventually output a strategy profile

Goal:

- Use poly-logarithmic communication
- Output an ϵ-NE or ϵ-WSNE

Communication Complexity

The setup:

- Both players know their own payoff matrix
- They do not know their opponent's
- They can communicate with each other
- They eventually output a strategy profile

Goal:

- Use poly-logarithmic communication
- Output an ϵ-NE or ϵ-WSNE

Previous best: (Goldberg & Pastink, 2012)

- 0.438-NE
- 0.732-WSNE

Communication Complexity

$(R, -R)$ and $(-C, C)$ can be solved with no communication

Communication Complexity

$(R, -R)$ and $(-C, C)$ can be solved with no communication

A strategy can be transmitted with $O(\log^2 n/\epsilon^2)$ communication
(Goldberg & Pastink, 2012)

- First sample $\log n/\epsilon^2$ pure strategies
- Transmit the samples
- whp. the transmitted strategy is within ϵ of the original

Communication Complexity

$(R, -R)$ and $(-C, C)$ can be solved with no communication

A strategy can be transmitted with $O(\log^2 n/\epsilon^2)$ communication
(Goldberg & Pastink, 2012)

- First sample $\log n/\epsilon^2$ pure strategies
- Transmit the samples
- whp. the transmitted strategy is within ϵ of the original

All other steps can be carried out using $O(\frac{\log^2 n}{\epsilon^2})$ communication

Communication Complexity

We have $O(\log^2 n/\epsilon^2)$ communication algorithms for:

- Finding a 0.6528-WSNE

Communication Complexity

We have $O(\log^2 n/\epsilon^2)$ communication algorithms for:

- Finding a 0.6528-WSNE
- Finding a 0.3820-NE
- Finding a 0.5-WSNE in a win-lose game

Outline

1. WSNE results
   - A simple algorithm to get a 2/3-WSNE
   - An improved algorithm to get a 0.6528-WSNE
2. Communication complexity results
   - Communication efficient algorithm for 0.6528-WSNE
3. Query complexity results
   - Query efficient algorithm for 0.6528-WSNE

Query complexity

The setup:

- The algorithm knows the size of the game, but not the payoffs
- It can make payoff queries
  - Propose a pure strategy profile (i, j)
  - Receive R_{ij} and C_{ij}

Goal:

- Use $< n^2$ queries
- Output an ϵ-NE or ϵ-WSNE

Query complexity

The setup:

- The algorithm knows the size of the game, but not the payoffs
- It can make payoff queries
  - Propose a pure strategy profile (i, j)
  - Receive R_{ij} and C_{ij}

Goal:

- Use $< n^2$ queries
- Output an ϵ-NE or ϵ-WSNE

Previous best: (Fearnley & Savani, 2014)

- 0.3820-NE using $O(n \cdot \log n/\epsilon^2)$ queries
- 0.6667-WSNE using $O(n \cdot \log n/\epsilon^4)$ queries

Query complexity

A zero-sum game can be solved using $O(n \cdot \log n/\epsilon^2)$ queries

(Goldberg & Roth, 2013)

Query complexity

A zero-sum game can be solved using $O(n \cdot \log n/\epsilon^2)$ queries
(Goldberg & Roth, 2013)

So:

- We solve $(R, -R)$ and $(-C, C)$ using $O(n \cdot \log n/\epsilon^2)$ queries
- We query j^* and j' using $2 \cdot n$ queries
- Carry out our algorithm

Query complexity

A zero-sum game can be solved using $O(n \cdot \log n/\epsilon^2)$ queries
(Goldberg & Roth, 2013)

So:

- We solve $(R, -R)$ and $(-C, C)$ using $O(n \cdot \log n/\epsilon^2)$ queries
- We query j^* and j' using $2 \cdot n$ queries
- Carry out our algorithm

We obtain a 0.6528-WSNE using $O(n \cdot \log n/\epsilon^2)$ queries

Conclusion

We gave a polynomial time algorithm for finding a 0.6528-WSNE

Communication complexity

- 0.6528-WSNE with polylogarithmic communication
- 0.3820-NE with polylogarithmic communication
- 0.5-WSNE with polylogarithmic communication in win-lose games

Query complexity

- 0.6528-WSNE with $n \cdot \log n$ queries