Odd Cycle Games and Connected Rules

Jan Corsten LSE
Adva Mond TAU
Alexey Pokrovskiy Birkbeck
Christoph Spiegel UPC
Tibor Szabó FUB

GAPCOMB Workshop

Campelles, 17th of June 2019

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH
Freie Universität Berlin
אוניברסיטת תל-אביב TEL AVIV UNIVERSITY

Variants of Positional Games

Definition (Positional Games)

Positional games are two-player games of perfect information played on a finite **board** X equipped with a family of **winning sets** $\mathcal{F} \subset 2^X$.

Variants of Positional Games

Definition (Positional Games)

Positional games are two-player games of perfect information played on a finite **board** X equipped with a family of **winning sets** $\mathcal{F} \subset 2^X$.

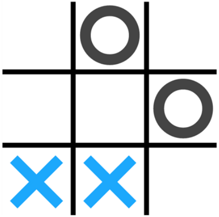
In the **strong game**, two players take turns claiming elements in X. The first player to claim all elements of a winning set in $\mathcal{F}$ wins.

Variants of Positional Games

Definition (Positional Games)

Positional games are two-player games of perfect information played on a finite **board** X equipped with a family of **winning sets** $\mathcal{F} \subset 2^X$.

In the **strong game**, two players take turns claiming elements in X. The first player to claim all elements of a winning set in $\mathcal{F}$ wins.



Variants of Positional Games

Definition (Positional Games)

Positional games are two-player games of perfect information played on a finite **board** X equipped with a family of **winning sets** $\mathcal{F} \subset 2^X$.

In the **strong game**, two players take turns claiming elements in X. The first player to claim all elements of a winning set in $\mathcal{F}$ wins.

Definition (Maker-Breaker Games)

Maker and **Breaker** take turns claiming elements from the board X. Maker wins if she claims a winning set and Breaker wins otherwise.

Variants of Positional Games

Definition (Positional Games)

Positional games are two-player games of perfect information played on a finite **board** X equipped with a family of **winning sets** $\mathcal{F} \subset 2^X$.

In the **strong game**, two players take turns claiming elements in X. The first player to claim all elements of a winning set in $\mathcal{F}$ wins.

Definition (Maker-Breaker Games)

Maker and **Breaker** take turns claiming elements from the board X. Maker wins if she claims a winning set and Breaker wins otherwise. In the **biased** version, Breaker claims b elements each round.

Variants of Positional Games

Definition (Positional Games)

Positional games are two-player games of perfect information played on a finite **board** X equipped with a family of **winning sets** $\mathcal{F} \subset 2^X$.

In the **strong game**, two players take turns claiming elements in X. The first player to claim all elements of a winning set in $\mathcal{F}$ wins.

Definition (Maker-Breaker Games)

Maker and **Breaker** take turns claiming elements from the board X. Maker wins if she claims a winning set and Breaker wins otherwise. In the **biased** version, Breaker claims b elements each round.

Definition (Client-Waiter Games)

Every round **Waiter** offers **Client** $1 \leq t \leq b+1$ elements. Client claims one of these and Waiter the rest. Client wins if she claims a winning set and Waiter wins otherwise.

Variants of Positional Games

Definition (Maker-Breaker Games)

Maker and **Breaker** take turns claiming elements from the board X. Maker wins if she claims a winning set and Breaker wins otherwise. In the **biased** version, Breaker claims b elements each round.

Definition (Client-Waiter Games)

Every round **Waiter** offers **Client** $1 \leq t \leq b+1$ elements. Client claims one of these and Waiter the rest. Client wins if she claims a winning set and Waiter wins otherwise.

Variants of Positional Games

Definition (Maker-Breaker Games)

Maker and **Breaker** take turns claiming elements from the board X. Maker wins if she claims a winning set and Breaker wins otherwise. In the **biased** version, Breaker claims b elements each round.

Definition (Client-Waiter Games)

Every round **Waiter** offers **Client** $1 \leq t \leq b+1$ elements. Client claims one of these and Waiter the rest. Client wins if she claims a winning set and Waiter wins otherwise.

For what values of b do Breaker and Waiter win?

The point where the winner switches is referred to as the **bias threshold**, denoted by b_{mb} and b_{cw}.

Examples of Positional Games

Let the board X be given by all edges of the complete graph on n vertices.

Example (Connectivity and Hamiltonicity Games)

The winning sets of the **connectivity game** consist of all spanning trees of K_n. Gebauer and Szabó showed that $b_{\mathrm{mb}} \approx n/\ln n$.

Examples of Positional Games

Let the board X be given by all edges of the complete graph on n vertices.

Example (Connectivity and Hamiltonicity Games)

The winning sets of the **connectivity game** consist of all spanning trees of K_n. Gebauer and Szabó showed that $b_{\mathrm{mb}} \approx n/\ln n$.

The winning sets of the **Hamiltonicity game** consist of all Hamiltonian cycles in K_n. Krivelevich showed that $b_{\mathrm{mb}} \approx n/\ln n$.

Examples of Positional Games

Let the board X be given by all edges of the complete graph on n vertices.

Example (Connectivity and Hamiltonicity Games)

The winning sets of the **connectivity game** consist of all spanning trees of K_n. Gebauer and Szabó showed that $b_{\mathrm{mb}} \approx n/\ln n$.

The winning sets of the **Hamiltonicity game** consist of all Hamiltonian cycles in K_n. Krivelevich showed that $b_{\mathrm{mb}} \approx n/\ln n$.

Example (Triangle and H-Games)

The winning sets of the **triangle game** are all triangles in K_n.

Examples of Positional Games

Let the board X be given by all edges of the complete graph on n vertices.

Example (Connectivity and Hamiltonicity Games)

The winning sets of the **connectivity game** consist of all spanning trees of K_n. Gebauer and Szabó showed that $b_{\text{mb}} \approx n/\ln n$.

The winning sets of the **Hamiltonicity game** consist of all Hamiltonian cycles in K_n. Krivelevich showed that $b_{\text{mb}} \approx n/\ln n$.

Example (Triangle and H-Games)

The winning sets of the **triangle game** are all triangles in K_n. The winning sets of the **H-game** are all copies of H in K_n, where H is fixed. Bednarska and Łuczak showed that $b_{\text{mb}} \approx n^{1/m_2(H)}$.

Examples of Positional Games

Let the board X be given by all edges of the complete graph on n vertices.

Example (Connectivity and Hamiltonicity Games)

The winning sets of the **connectivity game** consist of all spanning trees of K_n. Gebauer and Szabó showed that $b_{\mathrm{mb}} \approx n/\ln n$.

The winning sets of the **Hamiltonicity game** consist of all Hamiltonian cycles in K_n. Krivelevich showed that $b_{\mathrm{mb}} \approx n/\ln n$.

Example (Triangle and H-Games)

The winning sets of the **triangle game** are all triangles in K_n. The winning sets of the **H-game** are all copies of H in K_n, where H is fixed. Bednarska and Łuczak showed that $b_{\mathrm{mb}} \approx n^{1/m_2(H)}$.

Example (Cycle Games)

The winning sets of the **cycle game** are all cycles in K_n.

Examples of Positional Games

Let the board X be given by all edges of the complete graph on n vertices.

Example (Connectivity and Hamiltonicity Games)

The winning sets of the **connectivity game** consist of all spanning trees of K_n. Gebauer and Szabó showed that $b_{\mathrm{mb}} \approx n/\ln n$.

The winning sets of the **Hamiltonicity game** consist of all Hamiltonian cycles in K_n. Krivelevich showed that $b_{\mathrm{mb}} \approx n/\ln n$.

Example (Triangle and H-Games)

The winning sets of the **triangle game** are all triangles in K_n. The winning sets of the ***H*-game** are all copies of H in K_n, where H is fixed. Bednarska and Łuczak showed that $b_{\mathrm{mb}} \approx n^{1/m_2(H)}$.

Example (Cycle Games)

The winning sets of the **cycle game** are all cycles in K_n. In the **odd (even) cycle game** the winning sets are all odd (even) cycles.

Maker-Breaker Cycle Games

Theorem (Bednarska and Pikhurko 2005)

In the Maker-Breaker cycle game $b_{mb} = \lceil n/2 \rceil - 1$.

Maker-Breaker Cycle Games

Theorem (Bednarska and Pikhurko 2005)

In the Maker-Breaker cycle game $b_{mb} = \lceil n/2 \rceil - 1$.

Theorem (Bednarska and Pikhurko 2008)

In the Maker-Breaker **even** *cycle game*

$$b_{mb} = n/2 - o(n).$$

Maker-Breaker Cycle Games

Theorem (Bednarska and Pikhurko 2005)

In the Maker-Breaker cycle game $b_{mb} = \lceil n/2 \rceil - 1$.

Theorem (Bednarska and Pikhurko 2008)

In the Maker-Breaker **even** *cycle game*

$$b_{mb} = n/2 - o(n).$$

In the Maker-Breaker **odd** *cycle game*

$$b_{mb} \geq n - n/\sqrt{2} - o(n) \approx 0.2928n.$$

Maker-Breaker Cycle Games

Theorem (Bednarska and Pikhurko 2005)

In the Maker-Breaker cycle game $b_{mb} = \lceil n/2 \rceil - 1$.

Theorem (Bednarska and Pikhurko 2008)

In the Maker-Breaker **even** *cycle game*

$$b_{mb} = n/2 - o(n).$$

In the Maker-Breaker **odd** *cycle game*

$$b_{mb} \geq n - n/\sqrt{2} - o(n) \approx 0.2928n.$$

Theorem 1 (Corsten, Mond, Pokrovskiy, S. and Szabó ′19+)

In the Maker-Breaker odd cycle game

$$b_{mb} \geq \left(\frac{4-\sqrt{6}}{5} - o(1) \right) n \approx 0.3101n.$$

A Strategy for Maker

Observations

A Strategy for Maker

Observations

1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.

A Strategy for Maker

Observations

1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.

A Strategy for Maker

Observations

1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.

A Strategy for Maker

Observations

1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.

A Strategy for Maker

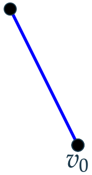
Observations

1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.

A Strategy for Maker

Observations

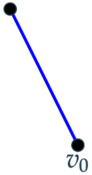
1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.



A Strategy for Maker

Observations

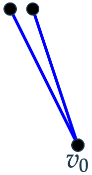
1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.



A Strategy for Maker

Observations

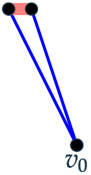
1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.



A Strategy for Maker

Observations

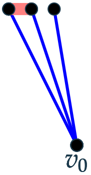
1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.



A Strategy for Maker

Observations

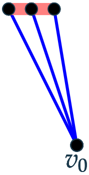
1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.



A Strategy for Maker

Observations

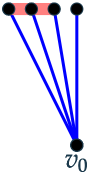
1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.



A Strategy for Maker

Observations

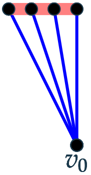
1. Maker’s graph is bipartite if she hasn’t yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker’s graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker’s graph.
4. To maximise the number of such edges, one part should be large.



A Strategy for Maker

Observations

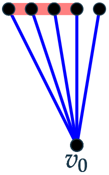
1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.



A Strategy for Maker

Observations

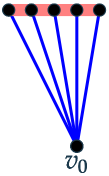
1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.



A Strategy for Maker

Observations

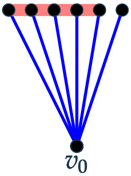
1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.



A Strategy for Maker

Observations

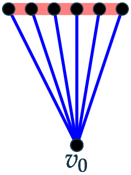
1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.



A Strategy for Maker

Observations

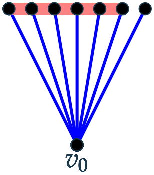
1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.



A Strategy for Maker

Observations

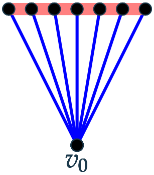
1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.



A Strategy for Maker

Observations

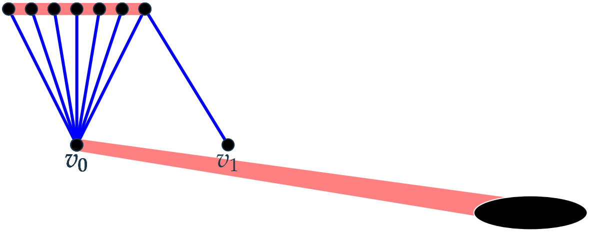
1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.



A Strategy for Maker

Observations

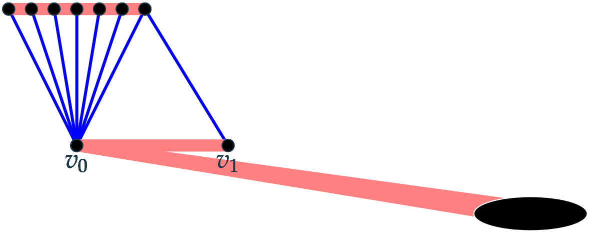
1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.



A Strategy for Maker

Observations

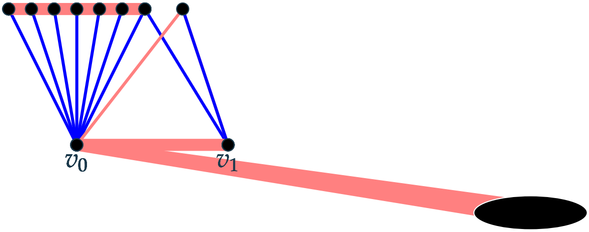
1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.



A Strategy for Maker

Observations

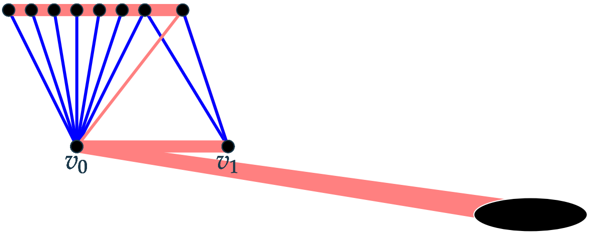
1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.



A Strategy for Maker

Observations

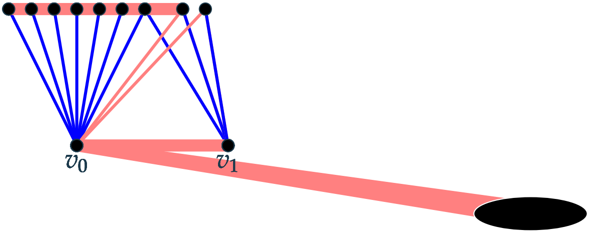
1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.



A Strategy for Maker

Observations

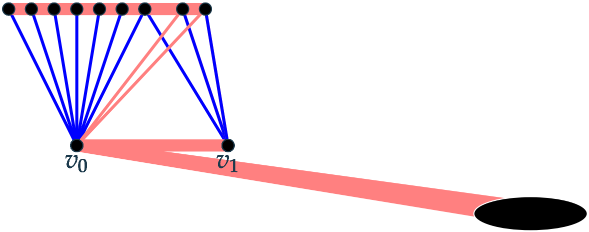
1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.



A Strategy for Maker

Observations

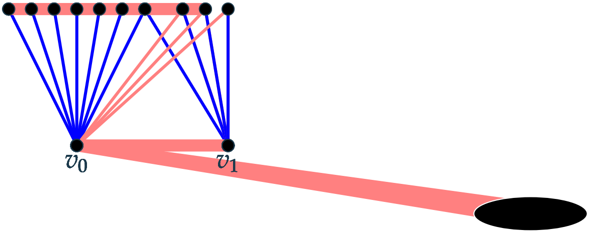
1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.



A Strategy for Maker

Observations

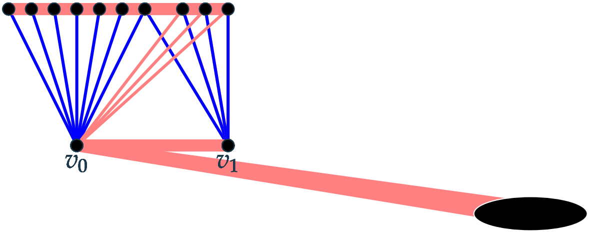
1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.



A Strategy for Maker

Observations

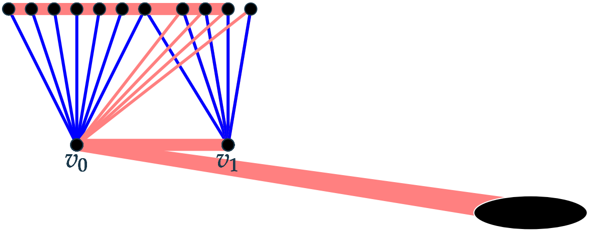
1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.



A Strategy for Maker

Observations

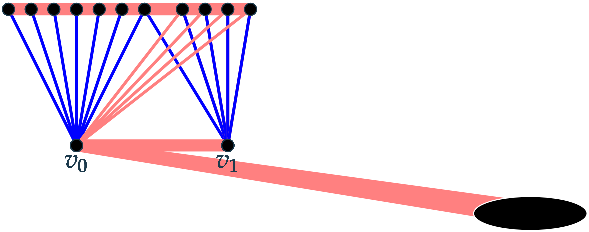
1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.



A Strategy for Maker

Observations

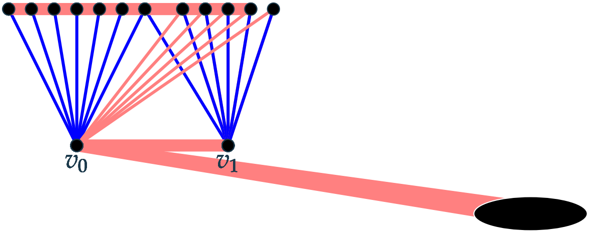
1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.



A Strategy for Maker

Observations

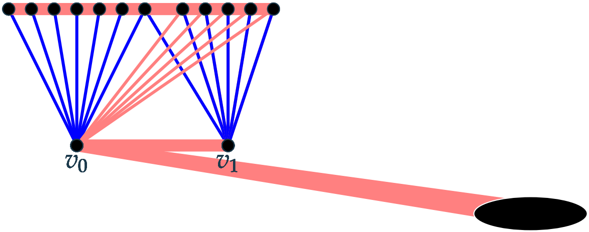
1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.



A Strategy for Maker

Observations

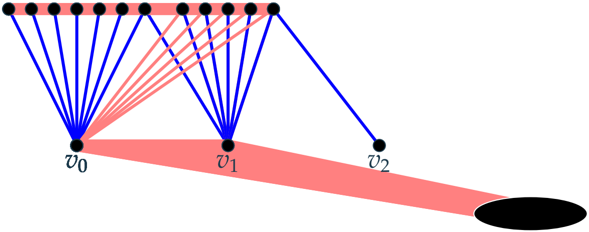
1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.



A Strategy for Maker

Observations

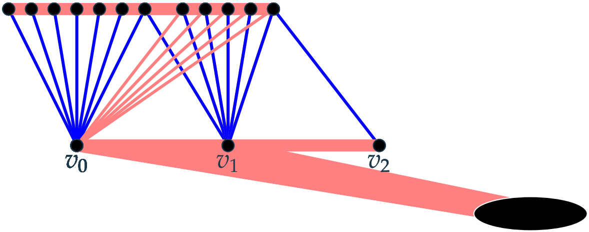
1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.



A Strategy for Maker

Observations

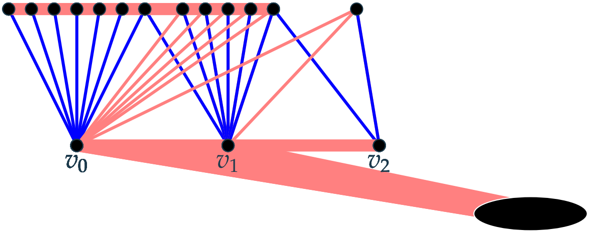
1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.



A Strategy for Maker

Observations

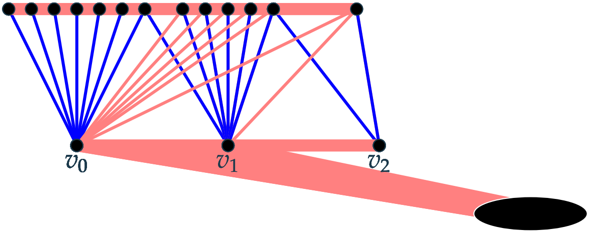
1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.



A Strategy for Maker

Observations

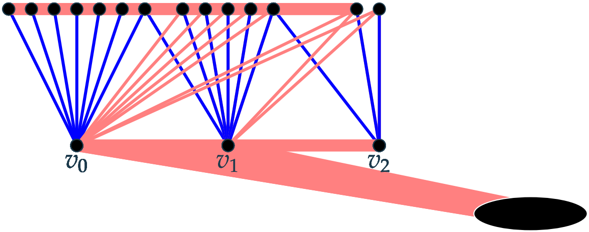
1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.



A Strategy for Maker

Observations

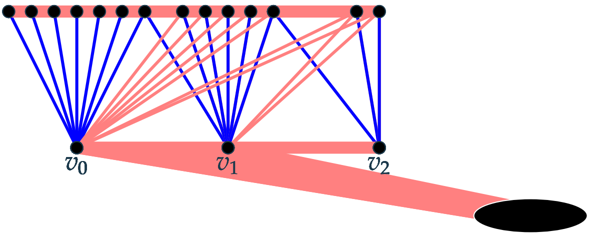
1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.



A Strategy for Maker

Observations

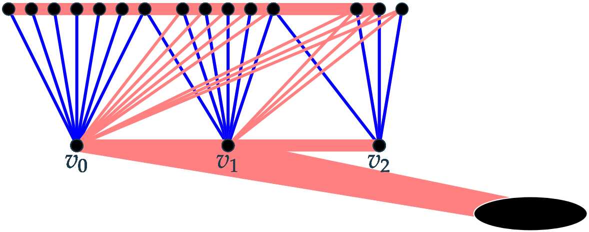
1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.



A Strategy for Maker

Observations

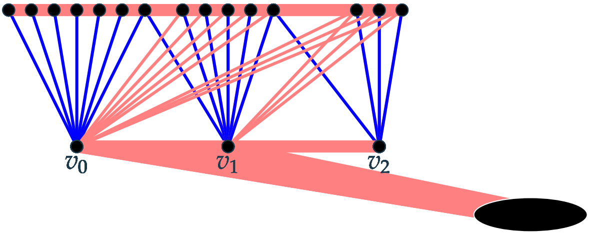
1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.



A Strategy for Maker

Observations

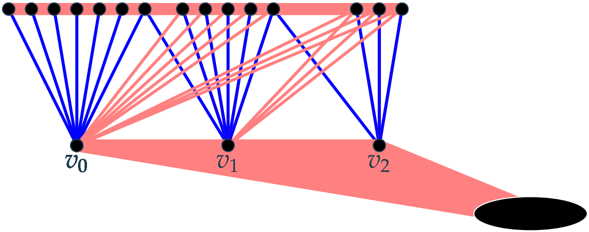
1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.



A Strategy for Maker

Observations

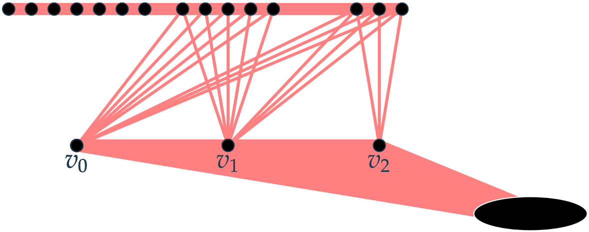
1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.



A Strategy for Maker

Observations

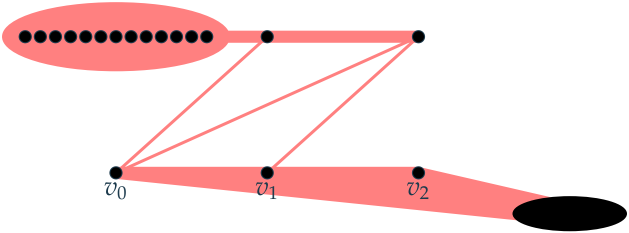
1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.



A Strategy for Maker

Observations

1. Maker's graph is bipartite if she hasn't yet claimed an odd cycle.
2. Claiming an **even** cycle never helps Maker.
3. If Maker's graph is connected, Breaker is forced to claim every edge within each part of the bipartition of Maker's graph.
4. To maximise the number of such edges, one part should be large.



Connected Maker-Breaker Cycle Games

Theorem (Bednarska and Pikhurko 2008)

In the Maker-Breaker odd cycle game $b_{mb} \geq 0.2928n$.

Theorem 1 (Corsten, Mond, Pokrovskiy, S. and Szabó '19+)

In the Maker-Breaker odd cycle game $b_{mb} \geq 0.3101n$.

Connected Maker-Breaker Cycle Games

Theorem (Bednarska and Pikhurko 2008)

In the Maker-Breaker odd cycle game $b_{mb} \geq 0.2928n$.

Theorem 1 (Corsten, Mond, Pokrovskiy, S. and Szabó ′19+)

In the Maker-Breaker odd cycle game $b_{mb} \geq 0.3101n$.

Question (Bednarska and Pikhurko)

Do we have $b_{mb} = n/2 - o(n)$ *in the odd cycle Maker-Breaker game?*

Connected Maker-Breaker Cycle Games

Theorem (Bednarska and Pikhurko 2008)

In the Maker-Breaker odd cycle game $b_{mb} \geq 0.2928n$.

Theorem 1 (Corsten, Mond, Pokrovskiy, S. and Szabó ′19+)

In the Maker-Breaker odd cycle game $b_{mb} \geq 0.3101n$.

Question (Bednarska and Pikhurko)

Do we have $b_{mb} = n/2 - o(n)$ *in the odd cycle Maker-Breaker game?*

Definition (Connected Maker-Breaker Games)

Maker has to claim edges incident to her previously claimed edges.

Connected Maker-Breaker Cycle Games

Theorem (Bednarska and Pikhurko 2008)

In the Maker-Breaker odd cycle game $b_{mb} \geq 0.2928n$.

Theorem 1 (Corsten, Mond, Pokrovskiy, S. and Szabó '19+)

In the Maker-Breaker odd cycle game $b_{mb} \geq 0.3101n$.

Question (Bednarska and Pikhurko)

Do we have $b_{mb} = n/2 - o(n)$ *in the odd cycle Maker-Breaker game?*

Definition (Connected Maker-Breaker Games)

Maker has to claim edges incident to her previously claimed edges.

Theorem 2 (Corsten, Mond, Pokrovskiy, S. and Szabó '19+)

In the **connected** *Maker-Breaker odd cycle game* $b^c_{mb} \leq 0.47n$.

A Strategy for Breaker under Connected Rules

Theorem 2 (Corsten, Mond, Pokrovskiy, S. and Szabó ′19+)
In the **connected** *Maker-Breaker odd cycle game* $b^c_{mb} \leq 0.47n$.

Proof Idea

A Strategy for Breaker under Connected Rules

Theorem 2 (Corsten, Mond, Pokrovskiy, S. and Szabó '19+)
In the **connected** *Maker-Breaker odd cycle game* $b^c_{mb} \leq 0.47n$.

Proof Idea

1. Maker's graph will again be bipartite as long as she hasn't won the game.

A Strategy for Breaker under Connected Rules

Theorem 2 (Corsten, Mond, Pokrovskiy, S. and Szabó '19+)
In the **connected** *Maker-Breaker odd cycle game* $b^c_{mb} \leq 0.47n$.

Proof Idea

1. Maker's graph will again be bipartite as long as she hasn't won the game.
2. Besides blocking any immediate threats of Maker creating an odd cycle, Breaker's goal will be to connect the vertices not yet touched by Maker in as even a way as possible to the two parts.

A Strategy for Breaker under Connected Rules

Theorem 2 (Corsten, Mond, Pokrovskiy, S. and Szabó '19+)
In the **connected** *Maker-Breaker odd cycle game* $b^c_{mb} \leq 0.47n$.

Proof Idea

1. Maker's graph will again be bipartite as long as she hasn't won the game.
2. Besides blocking any immediate threats of Maker creating an odd cycle, Breaker's goal will be to connect the vertices not yet touched by Maker in as even a way as possible to the two parts.
3. This way Breaker minimises the number of edges ending up between the two parts of Maker's graph.

Client-Waiter Cycle Games

Theorem (Hefetz, Krivelevich, and Tan 2016)

In the Client-Waiter cycle game $b_{cw} = \lceil n/2 \rceil - 1$.

Client-Waiter Cycle Games

Theorem (Hefetz, Krivelevich, and Tan 2016)

In the Client-Waiter cycle game $b_{cw} = \lceil n/2 \rceil - 1$.

In the Client-Waiter **odd** *cycle game* $b_{cw} \geq n/(4 \log 2) - o(n) \approx 0.3606n$.

Client-Waiter Cycle Games

Theorem (Hefetz, Krivelevich, and Tan 2016)

In the Client-Waiter cycle game $b_{cw} = \lceil n/2 \rceil - 1$.

In the Client-Waiter **odd** *cycle game* $b_{cw} \geq n/(4\log 2) - o(n) \approx 0.3606n$.

Conjecture (Hefetz, Krivelevich and Tan)

We have $b_{cw} = n/2 - o(n)$ *in the odd cycle Client-Waiter game.*

Client-Waiter Cycle Games

Theorem (Hefetz, Krivelevich, and Tan 2016)

In the Client-Waiter cycle game $b_{cw} = \lceil n/2 \rceil - 1$.

In the Client-Waiter **odd** *cycle game* $b_{cw} \geq n/(4\log 2) - o(n) \approx 0.3606n$.

Conjecture (Hefetz, Krivelevich and Tan)

We have $b_{cw} = n/2 - o(n)$ *in the odd cycle Client-Waiter game.*

Definition (Connected Client-Waiter Games)

Waiter has to offer edges incident to Client's previously claimed edges.

Client-Waiter Cycle Games

Theorem (Hefetz, Krivelevich, and Tan 2016)

In the Client-Waiter cycle game $b_{cw} = \lceil n/2 \rceil - 1$.

In the Client-Waiter **odd** *cycle game* $b_{cw} \geq n/(4\log 2) - o(n) \approx 0.3606n$.

Conjecture (Hefetz, Krivelevich and Tan)

We have $b_{cw} = n/2 - o(n)$ *in the odd cycle Client-Waiter game.*

Definition (Connected Client-Waiter Games)

Waiter has to offer edges incident to Client's previously claimed edges.

Theorem 3 (Corsten, Mond, Pokrovskiy, S. and Szabó '19+)

In the **connected** *Client-Waiter odd cycle game* $b^c_{cw} = \lceil n/2 \rceil - 1$.

A Strategy for Client under Connected Rules

Theorem 3 (Corsten, Mond, Pokrovskiy, S. and Szabó '19+)
In the **connected** *Client-Waiter odd cycle game* $b^c_{cw} = \lceil n/2 \rceil - 1$.

Proof Idea

A Strategy for Client under Connected Rules

Theorem 3 (Corsten, Mond, Pokrovskiy, S. and Szabó '19+)
In the **connected** *Client-Waiter odd cycle game* $b^c_{cw} = \lceil n/2 \rceil - 1$.

Proof Idea

1. Client's graph will again be bipartite as long as she hasn't won the game.

A Strategy for Client under Connected Rules

Theorem 3 (Corsten, Mond, Pokrovskiy, S. and Szabó '19+)
In the **connected** *Client-Waiter odd cycle game* $b^c_{cw} = \lceil n/2 \rceil - 1$.

Proof Idea

1. Client's graph will again be bipartite as long as she hasn't won the game.
2. If at any point there is an unclaimed edge inside either of the two parts, Waiter will loose.

A Strategy for Client under Connected Rules

Theorem 3 (Corsten, Mond, Pokrovskiy, S. and Szabó '19+)
In the **connected** *Client-Waiter odd cycle game* $b_{cw}^{c} = \lceil n/2 \rceil - 1$.

Proof Idea

1. Client's graph will again be bipartite as long as she hasn't won the game.
2. If at any point there is an unclaimed edge inside either of the two parts, Waiter will loose.
3. Whenever offering an edge incident to a vertex not yet in Client's graph, Waiter must either *offer all unclaimed edges between that vertex and Client's graph*

A Strategy for Client under Connected Rules

Theorem 3 (Corsten, Mond, Pokrovskiy, S. and Szabó ′19+)
In the **connected** *Client-Waiter odd cycle game* $b^c_{cw} = \lceil n/2 \rceil - 1$.

Proof Idea

1. Client's graph will again be bipartite as long as she hasn't won the game.
2. If at any point there is an unclaimed edge inside either of the two parts, Waiter will loose.
3. Whenever offering an edge incident to a vertex not yet in Client's graph, Waiter must either *offer all unclaimed edges between that vertex and Client's graph* or he must have *previously claimed all edges between that vertex and one part of the bipartition*.

A Strategy for Client under Connected Rules

Theorem 3 (Corsten, Mond, Pokrovskiy, S. and Szabó '19+)
In the **connected** *Client-Waiter odd cycle game* $b^c_{cw} = \lceil n/2 \rceil - 1$.

Proof Idea

1. Client's graph will again be bipartite as long as she hasn't won the game.
2. If at any point there is an unclaimed edge inside either of the two parts, Waiter will loose.
3. Whenever offering an edge incident to a vertex not yet in Client's graph, Waiter must either *offer all unclaimed edges between that vertex and Client's graph* or he must have *previously claimed all edges between that vertex and one part of the bipartition*.
4. Client tries to reduce the number of times the later occurs.

Open Question

Q1. What is the threshold bias for other variants of the odd cycle games, for example Avoider-Enforcer or Waiter-Client?

Q2. What is the threshold bias for the connected Maker-Breaker H-game?

Q3. One can view the odd cycle game as the non-2-colourability game. It was proved by Hefetz et al. that the threshold bias for the Maker-Breaker non-k-colourability game satisfies $b_{mb} = \Theta_k(n)$. Do we have $b_{mb} \approx b^c_{mb}$?

Thank you for your attention!