On the complexity of the Maker-Breaker Happy Vertex Game

Joint work with Mathieu Hilaire and Nacim Oijid

Perig Montfort

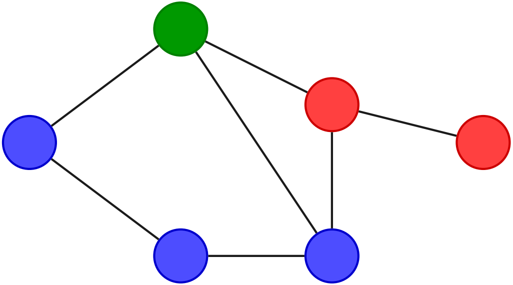
P-GASE Workshop, April 2, 2026

ENS DE LYON

LaBRI

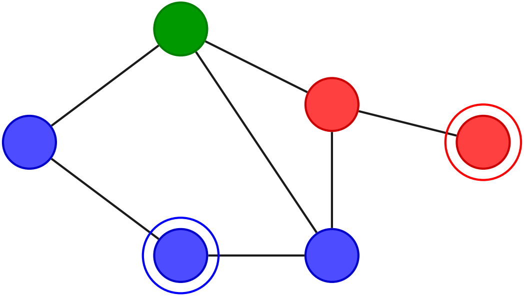
Definition (Zhang and Li – 2015)

Let $G = (V, E)$ be a graph with a vertex-coloring $c : V \to C$. A vertex $v \in V$ is called *happy* if for every neighbor $u \in N(v)$, we have $c(u) = c(v)$.



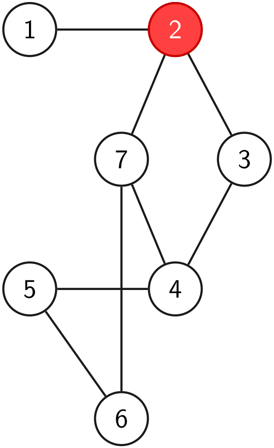
Definition (Zhang and Li – 2015)

Let $G = (V, E)$ be a graph with a vertex-coloring $c : V \to C$. A vertex $v \in V$ is called *happy* if for every neighbor $u \in N(v)$, we have $c(u) = c(v)$.



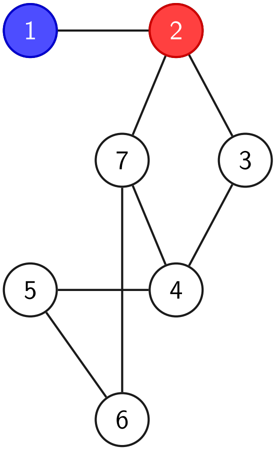
Maker–Breaker Happy Vertex Game

- Maker and Breaker alternately claim free vertices of a graph G until all vertices are claimed.



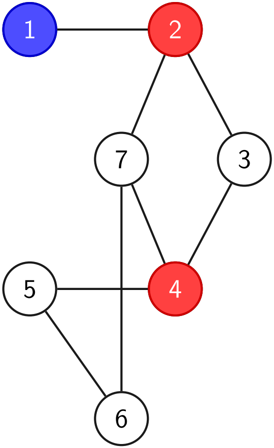
Maker–Breaker Happy Vertex Game

- Maker and Breaker alternately claim free vertices of a graph G until all vertices are claimed.



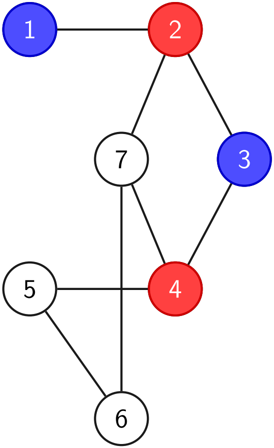
Maker–Breaker Happy Vertex Game

- Maker and Breaker alternately claim free vertices of a graph G until all vertices are claimed.
- Maker aims to maximize her number of happy vertices.



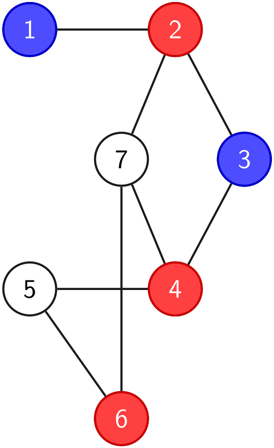
Maker–Breaker Happy Vertex Game

- Maker and Breaker alternately claim free vertices of a graph G until all vertices are claimed.
- Maker aims to maximize her number of happy vertices.



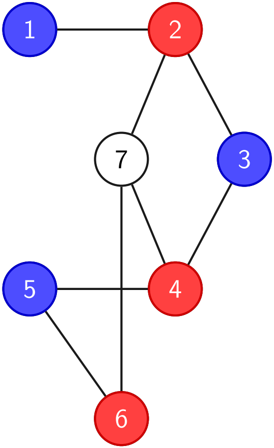
Maker–Breaker Happy Vertex Game

- Maker and Breaker alternately claim free vertices of a graph G until all vertices are claimed.
- Maker aims to maximize her number of happy vertices.
- Breaker aims to minimize the number of Maker's happy vertices.



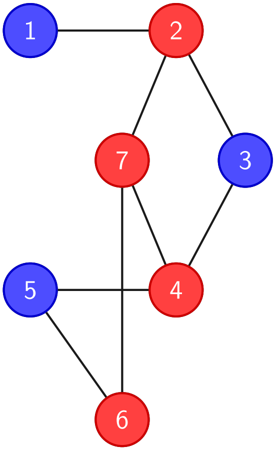
Maker–Breaker Happy Vertex Game

- Maker and Breaker alternately claim free vertices of a graph G until all vertices are claimed.
- Maker aims to maximize her number of happy vertices.
- Breaker aims to minimize the number of Maker's happy vertices.



Maker–Breaker Happy Vertex Game

- Maker and Breaker alternately claim free vertices of a graph G until all vertices are claimed.
- Maker aims to maximize her number of happy vertices.
- Breaker aims to minimize the number of Maker's happy vertices.



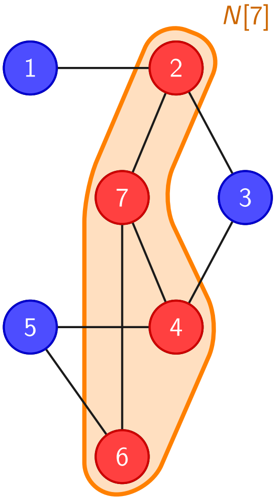
Definition

The score is the number of red happy vertices.

- Maker and Breaker alternately claim free vertices of a graph G until all vertices are claimed.
- Maker aims to maximize her number of happy vertices.
- Breaker aims to minimize the number of Maker's happy vertices.

Definition

The score is the number of red happy vertices.



Presentation of the problem

Definition (SCORING HAPPY VERTEX GAME)

Input: a graph G, an integer s, and a first player $X \in \{M, B\}$.

Question: do we have

$$Xs(G) \geq s \text{ ?}$$

Definition (Scoring Happy Vertex Game)

Input: a graph G, an integer s, and a first player $X \in \{M, B\}$.

Question: do we have

$$Xs(G) \geq s\ ?$$

- Find classes of graphs in which the problem is in P.

Presentation of the problem

Definition (Scoring Happy Vertex Game)

Input: a graph G, an integer s, and a first player $X \in \{M, B\}$.

Question: do we have

$$Xs(G) \geq s\ ?$$

- Find classes of graphs in which the problem is in P.
  - Subdivided stars graphs, maximum degree 2 graphs.

Presentation of the problem

Definition (Scoring Happy Vertex Game)

Input: a graph G, an integer s, and a first player $X \in \{M, B\}$.

Question: do we have

$$Xs(G) \geq s\,?$$

- Find classes of graphs in which the problem is in P.
  - Subdivided stars graphs, maximum degree 2 graphs.
- Find classes of graphs in which the problem is PSPACE-complete or NP-hard.

Presentation of the problem

Definition (Scoring Happy Vertex Game)

Input: a graph G, an integer s, and a first player $X \in \{M, B\}$.

Question: do we have

$$Xs(G) \geq s\ ?$$

- Find classes of graphs in which the problem is in P.
  - Subdivided stars graphs, maximum degree 2 graphs.
- Find classes of graphs in which the problem is PSPACE-complete or NP-hard.
  - Trees, caterpillars.

Definition (Scoring Happy Vertex Game)

Input: a graph G, an integer s, and a first player $X \in \{M, B\}$.

Question: do we have

$$Xs(G) \geq s\ ?$$

- Find classes of graphs in which the problem is in P.
  - Subdivided stars graphs, maximum degree 2 graphs.
- Find classes of graphs in which the problem is PSPACE-complete or NP-hard.
  - Trees, caterpillars.
- Find parameterized algorithms.

Presentation of the problem

Definition (SCORING HAPPY VERTEX GAME)

Input: a graph G, an integer s, and a first player $X \in \{M, B\}$.

Question: do we have

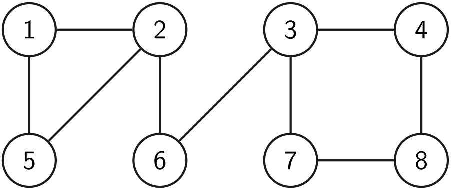
$$Xs(G) \geq s\ ?$$

- Find classes of graphs in which the problem is in P.
  - Subdivided stars graphs, maximum degree 2 graphs.
- Find classes of graphs in which the problem is PSPACE-complete or NP-hard.
  - Trees, caterpillars.
- Find parameterized algorithms.
  - FPT parameterized by neighborhood diversity.

Domination

Definition

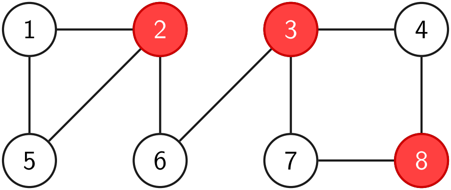
A dominating set is a subset of vertices $D \subseteq V$ such that $N[D] = V$.



Domination

Definition

A dominating set is a subset of vertices $D \subseteq V$ such that $N[D] = V$.



Example

$D = \{2, 3, 8\}$ is a dominating set.

Maker-Breaker Domination game

Definition (Duchene, Gledel, Parreau, Renault – 2020)

Two players, *Dominator* and *Staller*, alternately claim vertices of a graph G. Dominator wins if she claims a dominating set of G; otherwise, Staller wins.

Maker-Breaker Domination game

Definition (Duchene, Gledel, Parreau, Renault – 2020)

Two players, *Dominator* and *Staller*, alternately claim vertices of a graph G. Dominator wins if she claims a dominating set of G; otherwise, Staller wins.

Theorem

This game is:

Definition (Duchene, Gledel, Parreau, Renault – 2020)

Two players, *Dominator* and *Staller*, alternately claim vertices of a graph G. Dominator wins if she claims a dominating set of G; otherwise, Staller wins.

Theorem

This game is:

Maker-Breaker Domination game

Definition (Duchene, Gledel, Parreau, Renault – 2020)

Two players, *Dominator* and *Staller*, alternately claim vertices of a graph G. Dominator wins if she claims a dominating set of G; otherwise, Staller wins.

Theorem

This game is:

- PSPACE-*complete on:*
    - *Bipartite graphs of bounded degree;*
    - *split graphs.*

Maker-Breaker Domination game

Definition (Duchene, Gledel, Parreau, Renault – 2020)

Two players, *Dominator* and *Staller*, alternately claim vertices of a graph G. Dominator wins if she claims a dominating set of G; otherwise, Staller wins.

Theorem

This game is:

- PSPACE*-complete on:*
  - *Bipartite graphs of bounded degree;*
  - *split graphs.*
- *Polynomial on:*
  - *Trees;*
  - *unit interval graphs*

Link with the Happy Vertex Game

Dominator wins by claiming a set $B \iff B$ is a dominating set $\iff \forall v \in V, N[v] \cap B \neq \emptyset \iff \forall v \in V, v$ is not happy.

Link with the Happy Vertex Game

Dominator wins by claiming a set $B \iff B$ is a dominating set $\iff \forall v \in V, N[v] \cap B \neq \emptyset \iff \forall v \in V, v$ is not happy.

We have

$$s(G) = 0 \quad \iff \quad \text{Dominator wins.}$$

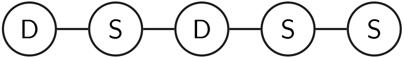
Link with the Happy Vertex Game

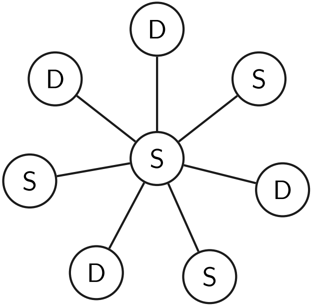
Dominator wins by claiming a set $B \iff B$ is a dominating set $\iff \forall v \in V, N[v] \cap B \neq \emptyset \iff \forall v \in V, v$ is not happy.

We have

$$s(G) = 0 \iff \text{Dominator wins.}$$

Motivation





Definition

Milnor's universe is the class of scoring games that are both *dicotic* and *non-zugzwang*:

- *dicotic*: For any position P,

$$\text{Maker can move in } P \iff \text{Breaker can move in } P.$$

- *non-zugzwang*: For any position P, $Ms(P) \geq Bs(P)$

Definition

Milnor's universe is the class of scoring games that are both *dicotic* and *non-zugzwang*:

- *dicotic*: For any position P,

$$\text{Maker can move in } P \iff \text{Breaker can move in } P.$$

- *non-zugzwang*: For any position P, $Ms(P) \geq Bs(P)$

Theorem (Bagan et al – 2024)

Every scoring positional games belong to Milnor's Universe.

Boundaries for union and neutral

Theorem (Milnor)

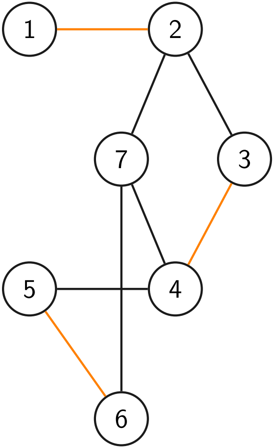
Let P_1 and P_2 be two positions in the Happy vertex game. Then

$$Ms(P_1) + Bs(P_2) \leq Ms(P_1 + P_2) \leq Ms(P_1) + Ms(P_2)$$

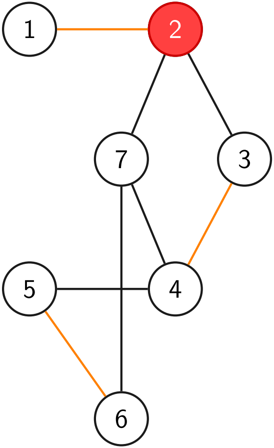
Corollary

The neutrals for the union on the Maker-Breaker Happy Vertex Game are exactly the games where *Dominator* wins in the Maker-Breaker Domination Game.

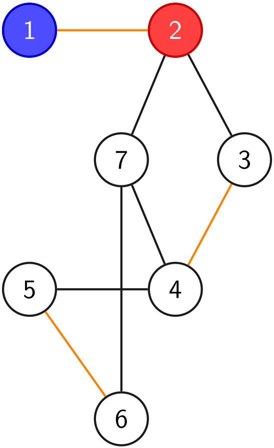
Pairing strategy



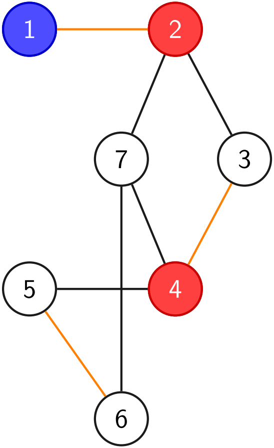
Pairing strategy



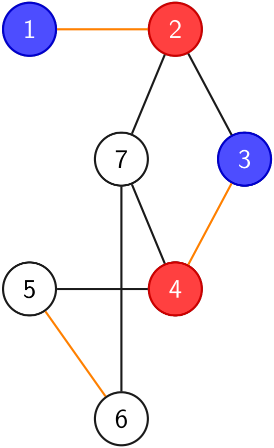
Pairing strategy



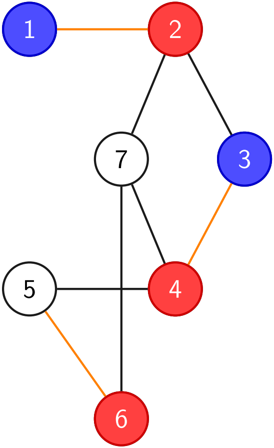
Pairing strategy



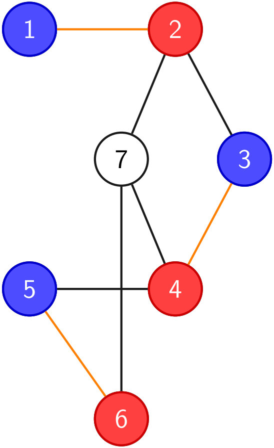
Pairing strategy



Pairing strategy



Pairing strategy

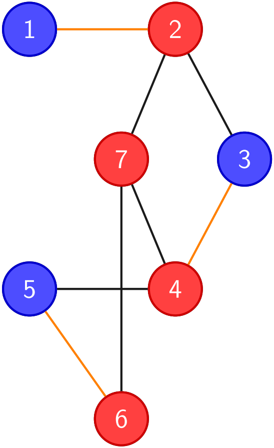


Pairing strategy

Theorem

Let G be a graph and M a matching of G:

$$s(G) \leq |V| - 2 \cdot |M|.$$



The Super Lemma

Theorem

Let (G, M, B) be a position of the game on a graph G, and let $u, v \in V_F$ be two distinct vertices such that for all $X \subseteq V_F \setminus \{u, v\}$,

$$|\{w \in V \mid N[w] \subseteq X \cup \{u\} \cup M\}| = |\{w \in V \mid N[w] \subseteq X \cup \{v\} \cup M\}| .$$

Then

$$s(G, M, B) = s(G,\ M \cup \{u\},\ B \cup \{v\}).$$

The Super Lemma

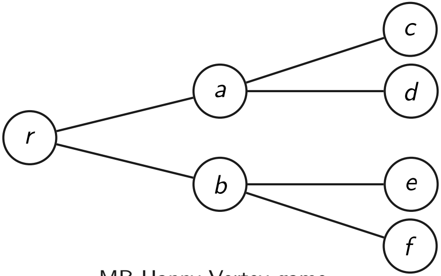
Theorem

Let (G, M, B) *be a position of the game on a graph* G*, and let* $u, v \in V_F$ *be two distinct vertices such that for all* $X \subseteq V_F \setminus \{u, v\}$,

$$|\{w \in V \mid N[w] \subseteq X \cup \{u\} \cup M\}| = |\{w \in V \mid N[w] \subseteq X \cup \{v\} \cup M\}|.$$

Then

$$s(G, M, B) = s(G,\ M \cup \{u\},\ B \cup \{v\}).$$



The Super Lemma

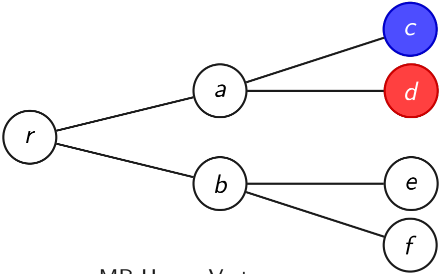
Theorem

Let (G, M, B) be a position of the game on a graph G, and let $u, v \in V_F$ be two distinct vertices such that for all $X \subseteq V_F \setminus \{u, v\}$,

$$|\{w \in V \mid N[w] \subseteq X \cup \{u\} \cup M\}| = |\{w \in V \mid N[w] \subseteq X \cup \{v\} \cup M\}|.$$

Then

$$s(G, M, B) = s(G,\ M \cup \{u\},\ B \cup \{v\}).$$



The Super Lemma

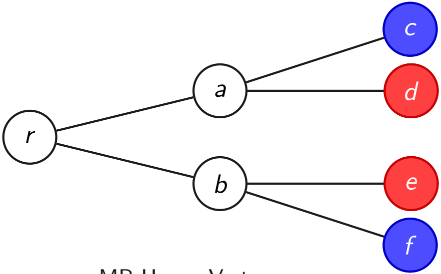
Theorem

Let (G, M, B) be a position of the game on a graph G, and let $u, v \in V_F$ be two distinct vertices such that for all $X \subseteq V_F \setminus \{u, v\}$,

$$|\{w \in V \mid N[w] \subseteq X \cup \{u\} \cup M\}| = |\{w \in V \mid N[w] \subseteq X \cup \{v\} \cup M\}|.$$

Then

$$s(G, M, B) = s(G,\ M \cup \{u\},\ B \cup \{v\}).$$



The Super Lemma

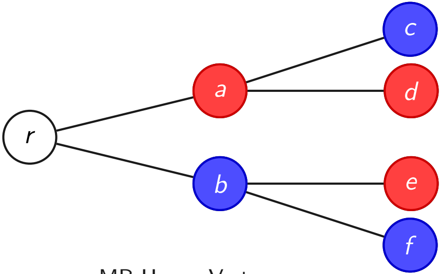
Theorem

Let (G, M, B) be a position of the game on a graph G, and let $u, v \in V_F$ be two distinct vertices such that for all $X \subseteq V_F \setminus \{u, v\}$,

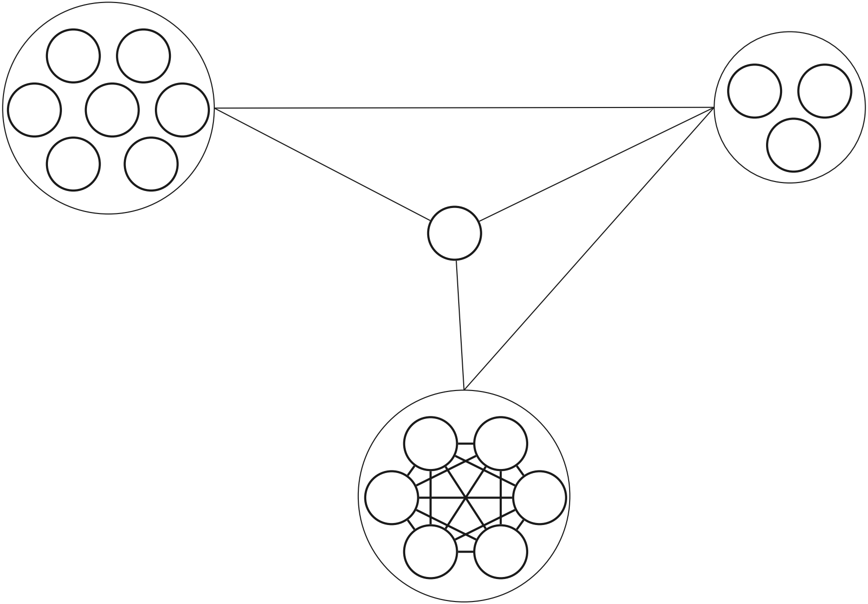
$$|\{w \in V \mid N[w] \subseteq X \cup \{u\} \cup M\}| = |\{w \in V \mid N[w] \subseteq X \cup \{v\} \cup M\}| .$$

Then

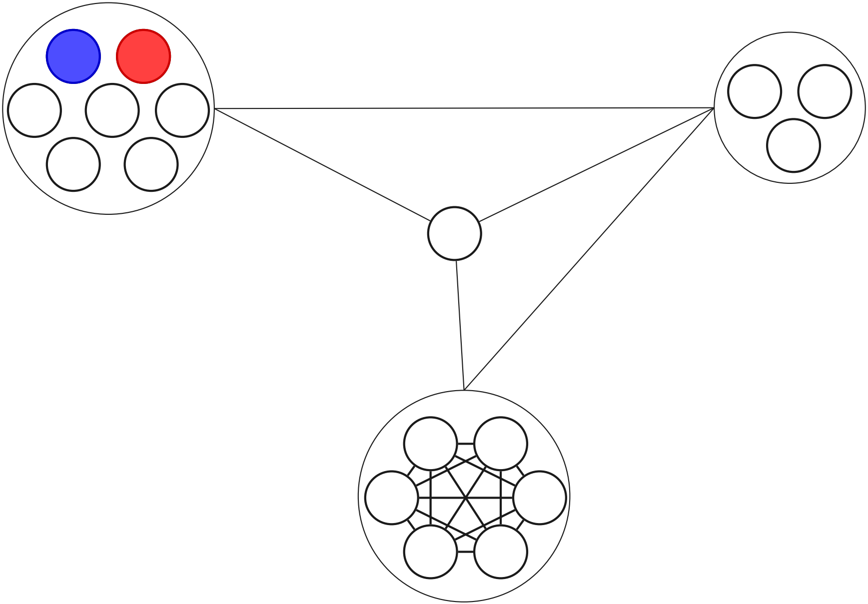
$$s(G, M, B) = s(G,\ M \cup \{u\},\ B \cup \{v\}).$$



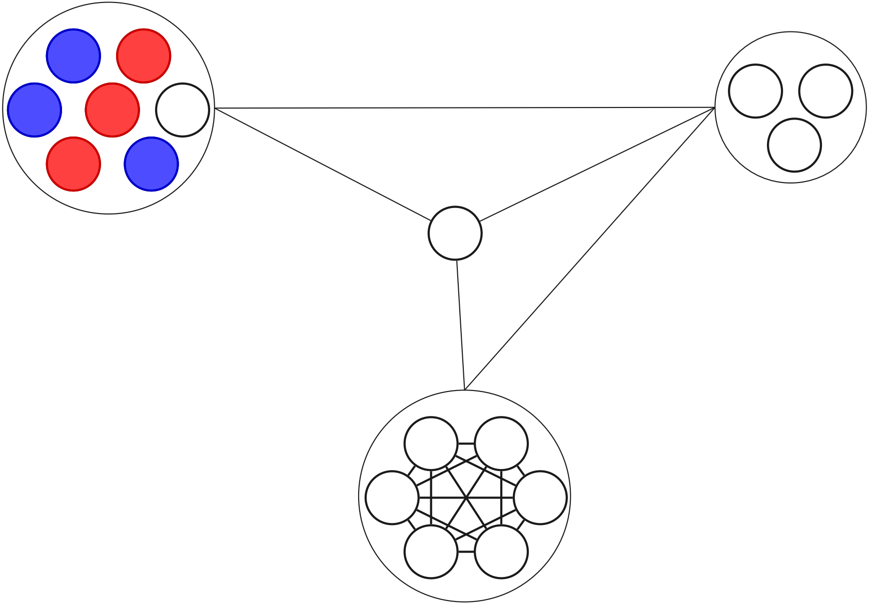
Neighborhood Diversity



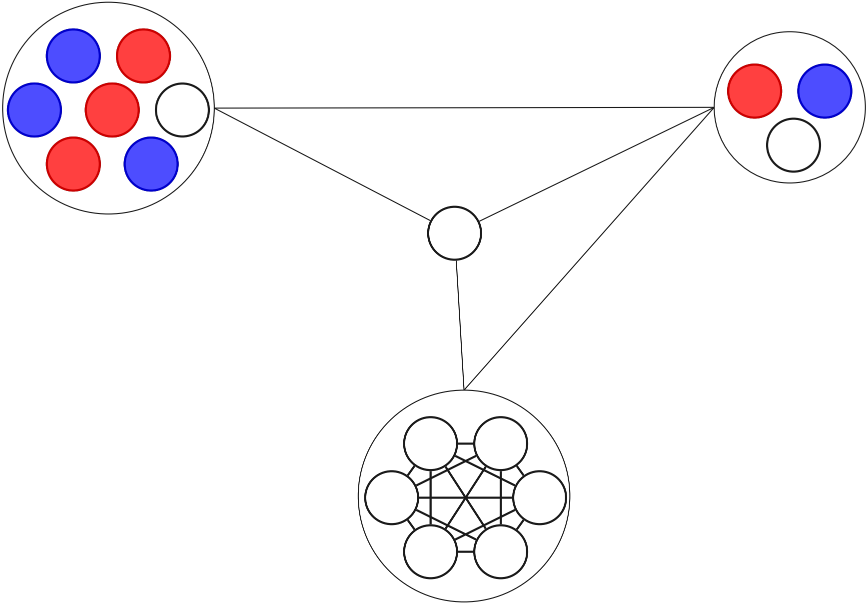
Neighborhood Diversity



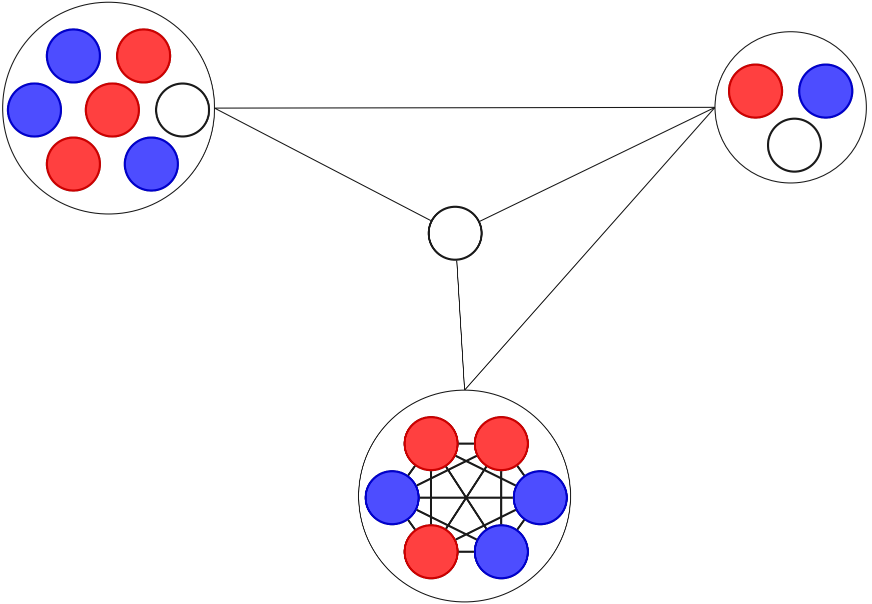
Neighborhood Diversity



Neighborhood Diversity



Neighborhood Diversity



Definition (Quantified Max 2-SAT)

Let

$$\psi = Q_1 x_1 Q_2 x_2 \ldots Q_n x_n \varphi$$

be a quantified Boolean formula where $Q_i \in \{\forall, \exists\}$ and φ is a quantifier-free 2-CNF formula over variables $\{x_1, \ldots, x_n\}$.

Definition (QUANTIFIED MAX 2-SAT)

Let

$$\psi = Q_1 x_1 Q_2 x_2 \dots Q_n x_n \varphi$$

be a quantified Boolean formula where $Q_i \in \{\forall, \exists\}$ and φ is a quantifier-free 2-CNF formula over variables $\{x_1, \dots, x_n\}$.

Question

Can at least k clauses of φ be satisfied?

Quantified Max 2-SAT

Definition (QUANTIFIED MAX 2-SAT)

Let

$$\psi = Q_1 x_1 Q_2 x_2 \dots Q_n x_n \, \varphi$$

be a quantified Boolean formula where $Q_i \in \{\forall, \exists\}$ and φ is a quantifier-free 2-CNF formula over variables $\{x_1, \dots, x_n\}$.

Question

Can at least k clauses of φ be satisfied?

Theorem (Bagan et al – 2024)

QUANTIFIED MAX 2-SAT *is* PSPACE-*complete.*

Quantified Max 2-SAT-2-2

Definition (Quantified Max 2-SAT-2-2)

Let

$$\psi = Q_1 x_1 Q_2 x_2 \dots Q_n x_n \, \varphi$$

be a quantified Boolean formula where $Q_i \in \{\forall, \exists\}$ and φ is a quantifier-free 2-CNF formula over variables $\{x_1, \dots, x_n\}$,

Quantified Max 2-SAT-2-2

Definition (Quantified Max 2-SAT-2-2)

Let

$$\psi = Q_1 x_1 Q_2 x_2 \ldots Q_n x_n \, \varphi$$

be a quantified Boolean formula where $Q_i \in \{\forall, \exists\}$ and φ is a quantifier-free 2-CNF formula over variables $\{x_1, \ldots, x_n\}$, where each variables appears no more than twice positively and twice negatively.

Quantified Max 2-SAT-2-2

Definition (Quantified Max 2-SAT-2-2)

Let

$$\psi = Q_1 x_1 Q_2 x_2 \dots Q_n x_n \varphi$$

be a quantified Boolean formula where $Q_i \in \{\forall, \exists\}$ and φ is a quantifier-free 2-CNF formula over variables $\{x_1, \dots, x_n\}$, where each variables appears no more than twice positively and twice negatively.

Question:

Can at least k clauses of φ be satisfied?

Quantified Max 2-SAT-2-2

Definition (Quantified Max 2-SAT-2-2)

Let

$$\psi = Q_1 x_1 Q_2 x_2 \dots Q_n x_n \varphi$$

be a quantified Boolean formula where $Q_i \in \{\forall, \exists\}$ and φ is a quantifier-free 2-CNF formula over variables $\{x_1, \dots, x_n\}$, where each variables appears no more than twice positively and twice negatively.

Question:

Can at least k clauses of φ be satisfied?

Theorem

Quantified Max 2-SAT-2-2 *is* NP*-hard.*

Literal–clause incidence graph

The literal–clause incidence graph G_φ with

$$\varphi = (x_1 \vee \neg x_2) \wedge (x_2 \vee x_3) \wedge (\neg x_1 \vee x_3) \wedge (\neg x_1 \vee x_2)$$

is constructed as follows.

Literal–clause incidence graph

The literal–clause incidence graph G_φ with

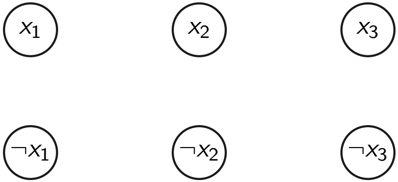
$$\varphi = (x_1 \vee \neg x_2) \wedge (x_2 \vee x_3) \wedge (\neg x_1 \vee x_3) \wedge (\neg x_1 \vee x_2)$$

is constructed as follows.

The literal–clause incidence graph G_φ with

$$\varphi = (x_1 \vee \neg x_2) \wedge (x_2 \vee x_3) \wedge (\neg x_1 \vee x_3) \wedge (\neg x_1 \vee x_2)$$

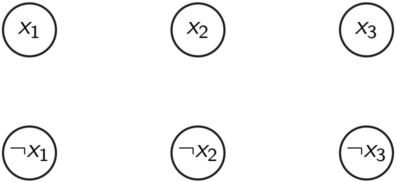
is constructed as follows.



The literal–clause incidence graph G_φ with

$$\varphi = (x_1 \lor \neg x_2) \land (x_2 \lor x_3) \land (\neg x_1 \lor x_3) \land (\neg x_1 \lor x_2)$$

is constructed as follows.

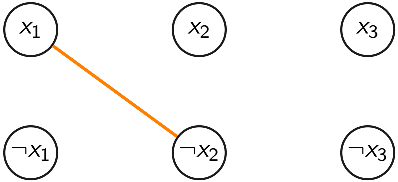


Literal–clause incidence graph

The literal–clause incidence graph G_φ with

$$\varphi = (x_1 \vee \neg x_2) \wedge (x_2 \vee x_3) \wedge (\neg x_1 \vee x_3) \wedge (\neg x_1 \vee x_2)$$

is constructed as follows.

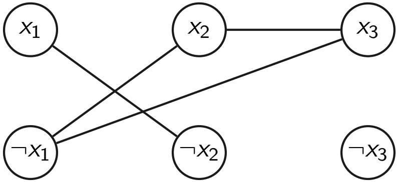


Literal–clause incidence graph

The literal–clause incidence graph G_φ with

$$\varphi = (x_1 \vee \neg x_2) \wedge (x_2 \vee x_3) \wedge (\neg x_1 \vee x_3) \wedge (\neg x_1 \vee x_2)$$

is constructed as follows.



Restricted variants

Definition

- Acyclic Quantified Max 2-SAT: the graph G_φ is acyclic.
- Acyclic Quantified Max 2-SAT-2-2: the graph G_φ is a disjoint union of paths.

Restricted variants

Definition

- ACYCLIC QUANTIFIED MAX 2-SAT: the graph G_φ is acyclic.
- ACYCLIC QUANTIFIED MAX 2-SAT-2-2: the graph G_φ is a disjoint union of paths.

Theorem

ACYCLIC QUANTIFIED MAX 2-SAT *is* PSPACE*-complete.*

Restricted variants

Definition

- Acyclic Quantified Max 2-SAT: the graph G_φ is acyclic.
- Acyclic Quantified Max 2-SAT-2-2: the graph G_φ is a disjoint union of paths.

Theorem

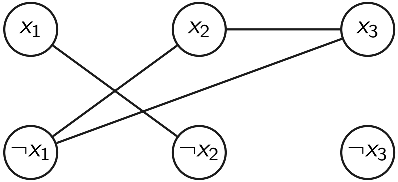
Acyclic Quantified Max 2-SAT *is* PSPACE-*complete.*

Theorem

Acyclic Quantified Max 2-SAT-2-2 *is* NP-*hard.*

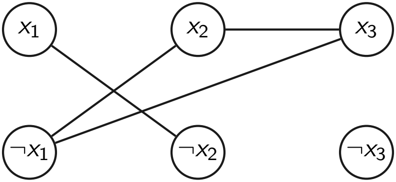
Cycle-breaking gadget

$$\varphi = (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2) \wedge (x_2 \vee x_3) \wedge (x_3 \vee \neg x_1)$$



Cycle-breaking gadget

$$\varphi = (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2) \wedge (x_2 \vee x_3) \wedge (x_3 \vee \neg x_1)$$

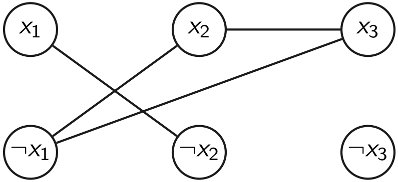


Add the gadget clauses

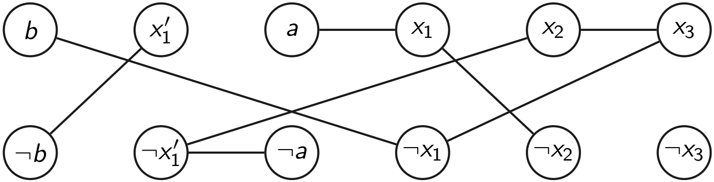
$$(x_1 \vee a), \qquad (\neg x_1 \vee b), \qquad (\neg a \vee \neg x_1'), \qquad (\neg b \vee x_1').$$

Cycle-breaking gadget

$$\varphi = (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2) \wedge (x_2 \vee x_3) \wedge (x_3 \vee \neg x_1)$$



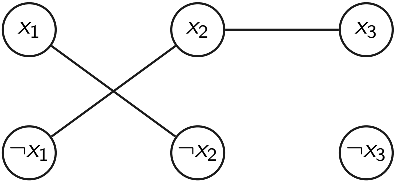
Becomes



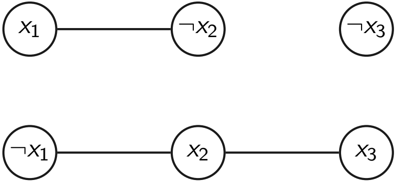
Reduction to SHVG in trees/caterpillars

$$\varphi = (x_1 \lor \neg x_2) \land (\neg x_1 \lor x_2) \land (x_2 \lor x_3)$$

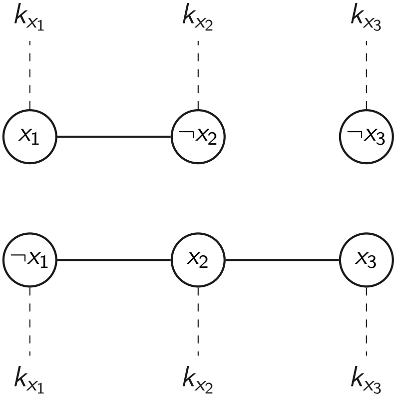
Reduction to SHVG in trees/caterpillars



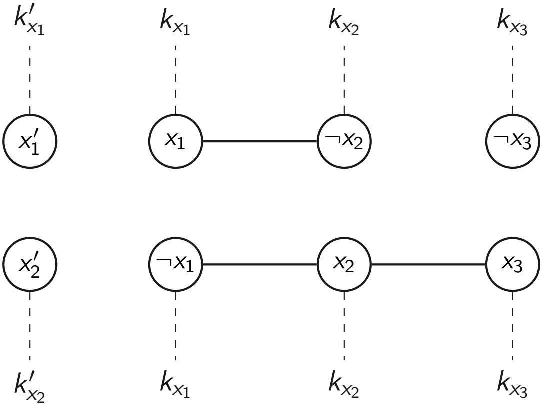
Reduction to SHVG in trees/caterpillars



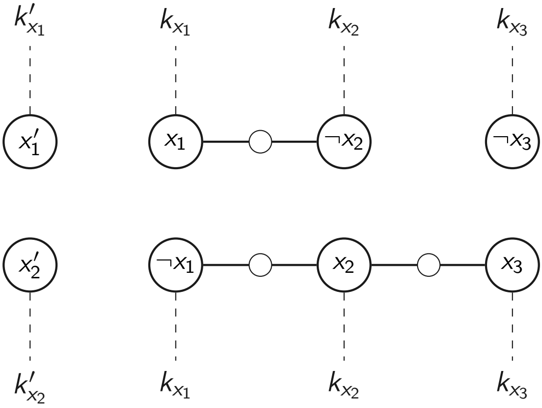
Reduction to SHVG in trees/caterpillars



Reduction to SHVG in trees/caterpillars

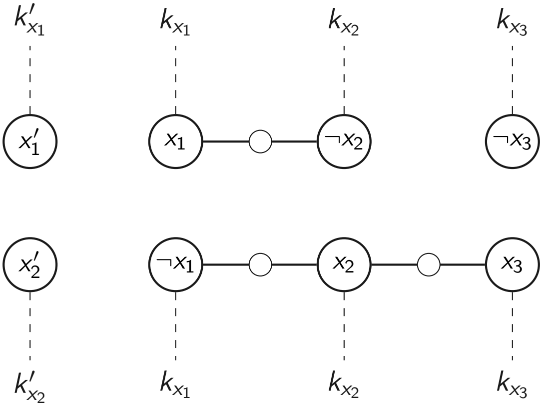


Reduction to SHVG in trees/caterpillars



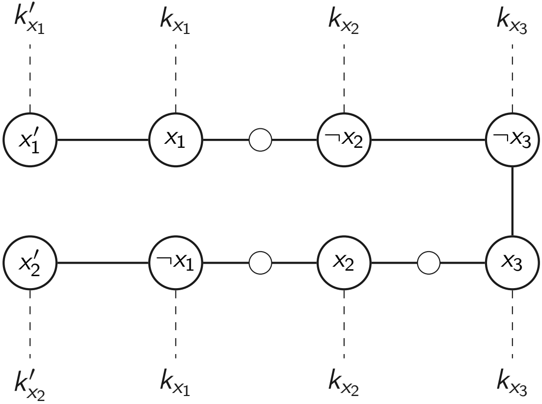
Reduction to SHVG in trees/caterpillars

$2\times$



Reduction to SHVG in trees/caterpillars

$2\times$



Open problems

Caterpillars

Is the problem on caterpillars NP-complete or already PSPACE-complete?

Open problems

Caterpillars

Is the problem on caterpillars NP-complete or already PSPACE-complete?

Larger tractable graph classes

Can the positive results be extended beyond neighborhood diversity, for instance through decomposition parameters?

Open problems

Caterpillars

Is the problem on caterpillars NP-complete or already PSPACE-complete?

Larger tractable graph classes

Can the positive results be extended beyond neighborhood diversity, for instance through decomposition parameters?

Cographs and modular decompositions

Can decomposition techniques be pushed far enough to obtain exact score computations on cographs or related classes?