SAT-Based Enumeration of Solutions to the Yang-Baxter Equation

Daimy Van Caudenberg, Bart Bogaerts, Leandro Vendramin

KU Leuven, Vrije Universiteit Brussel

March 19, 2025

KU LEUVEN

YANG-BAXTER EQUATION

Yang-Baxter Equation [Yan67, Bax72]

A solution to the Yang-Baxter equation (YBE) is a pair (V, R), where V is a vector space and $R : V \otimes V \to V \otimes V$ is a map such that in $V \otimes V \otimes V$,

$$R_1 R_2 R_1 = R_2 R_1 R_2, \qquad \text{(the \textit{original} Yang-Baxter Equation)}$$

where $R_1 = R \otimes \mathrm{id}$ and $R_2 = \mathrm{id} \otimes R$.

SET-THEORETIC YANG-BAXTER EQUATION

Set-Theoretic Yang-Baxter Equation (YBE) [Dri92]

A set-theoretic solution to the YBE is a pair (X, r), where X is a non-empty set and $r : X^2 \to X^2$ is a map such that in X^3,

$$r_1 r_2 r_1 = r_2 r_1 r_2, \qquad \text{(the Yang-Baxter Equation)}$$

where r_i acts as r on components i and $i+1$ and as the identity on the other component.

- These solutions are a subset of the solutions to the *original* Yang-Baxter equation.

SET-THEORETIC YANG-BAXTER EQUATION

Set-Theoretic Yang-Baxter Equation (YBE) [Dri92]

A set-theoretic solution to the YBE is a pair (X, r), where X is a non-empty set and $r : X^2 \to X^2$ is a map such that in X^3,

$$r_1 r_2 r_1 = r_2 r_1 r_2, \qquad \text{(the Yang-Baxter Equation)}$$

where r_i acts as r on components i and $i+1$ and as the identity on the other component.

- These solutions are a subset of the solutions to the *original* Yang-Baxter equation.
- Two set-theoretic solutions (X, r) and (X, s) are isomorphic if there exists a bijection $f : X \to X$ such that $(f \times f)r = s(f \times f)$.

SET-THEORETIC YANG-BAXTER EQUATION

Set-Theoretic Yang-Baxter Equation (YBE) [Dri92]

A set-theoretic solution to the YBE is a pair (X, r), where X is a non-empty set and $r : X^2 \to X^2$ is a map such that in X^3,

$$r_1 r_2 r_1 = r_2 r_1 r_2, \qquad \text{(the Yang-Baxter Equation)}$$

where r_i acts as r on components i and $i+1$ and as the identity on the other component.

- These solutions are a subset of the solutions to the *original* Yang-Baxter equation.
- Two set-theoretic solutions (X, r) and (X, s) are isomorphic if there exists a bijection $f : X \to X$ such that $(f \times f)r = s(f \times f)$.
- A set-theoretic solution is called involutive if $r^2 = id_{X \times X}$.

SET-THEORETIC YANG-BAXTER EQUATION

Set-Theoretic Yang-Baxter Equation (YBE) [Dri92]

A set-theoretic solution to the YBE is a pair (X, r), where X is a non-empty set and $r : X^2 \to X^2$ is a map such that in X^3,

$$r_1 r_2 r_1 = r_2 r_1 r_2, \qquad \text{(the Yang-Baxter Equation)}$$

where r_i acts as r on components i and $i+1$ and as the identity on the other component.

- These solutions are a subset of the solutions to the *original* Yang-Baxter equation.
- Two set-theoretic solutions (X, r) and (X, s) are isomorphic if there exists a bijection $f : X \to X$ such that $(f \times f)r = s(f \times f)$.
- A set-theoretic solution is called involutive if $r^2 = id_{X \times X}$.
- A set-theoretic solution (X, r) with $r(x, y) = (\sigma_x(y), \tau_y(x))$ is called non-degenerate if the maps σ_x and τ_x are bijective for all $x \in X$.

SET-THEORETIC YANG-BAXTER EQUATION

EXAMPLE

- The pair (X, r) with X the ring of integers modulo $n \in \mathbb{N}$ and $r(x, y) = (y+1, x-1)$ is a non-degenerate, involutive solution.

$$\forall x, y, z \in X :$$

$$\begin{aligned} & r_1 r_2 r_1(x, y, z) \\ &= r_1 r_2(y+1, x-1, z) \\ &= r_1(y+1, z+1, x-2) \\ &= (z+2, y, x-2) \end{aligned}$$

$$\begin{aligned} & r_2 r_1 r_2(x, y, z) \\ &= r_2 r_1(x, z+1, y-1) \\ &= r_2(z+2, x-1, y-1) \\ &= (z+2, y, x-2) \end{aligned}$$

SET-THEORETIC YANG-BAXTER EQUATION

EXAMPLE

- The pair (X, r) with X the ring of integers modulo $n \in \mathbb{N}$ and $r(x, y) = (y + 1, x - 1)$ is a non-degenerate, involutive solution.

$$\forall x, y \in X :$$

$$r^2(x, y) = r(y + 1, x - 1) = (x, y)$$

SET-THEORETIC YANG-BAXTER EQUATION

EXAMPLE

- The pair (X, r) with X the ring of integers modulo $n \in \mathbb{N}$ and $r(x, y) = (y + 1, x - 1)$ is a non-degenerate, involutive solution.

$r : X^2 \to X^2, (x, y) \mapsto (\sigma_x(y), \tau_y(x))$ where:

$$\forall x \in X : \sigma_x : X \to X, x \mapsto x + 1$$

$$\forall x \in X : \tau_x : X \to X, x \mapsto x - 1$$

SET-THEORETIC YANG-BAXTER EQUATION

EXAMPLE

- The pair (X, r) with X the ring of integers modulo $n \in \mathbb{N}$ and $r(x, y) = (y + 1, x - 1)$ is a non-degenerate, involutive solution.
- Given $f : X \to X : x \mapsto n - x$, the solution (X, s) with $s(x, y) = (y - 1, x + 1)$ is isomorphic to (X, r).

$$\begin{array}{ccc}
(x, y) & \xrightarrow{(f \times f)} & (n - x, n - y) \\
\Big\downarrow r & & \Big\downarrow s \\
(y + 1, x - 1) & \xrightarrow[(f \times f)]{} & (n - y - 1, n - x + 1)
\end{array}$$

CYCLE SETS

Cycle Sets

A cycle set $(X, \cdot)$ is a pair consisting of a non-empty set X and a binary operation $\cdot$ on X that fulfills the following relations:

1. the map $\phi_x : X \to X : y \mapsto x \cdot y$ is bijective for all $x \in X$ and
2. for all $x, y, z \in X$:

$$(x \cdot y) \cdot (x \cdot z) = (y \cdot x) \cdot (y \cdot z). \qquad \text{(the cycloid equation)}$$

CYCLE SETS

Cycle Sets

A cycle set $(X, \cdot)$ is a pair consisting of a non-empty set X and a binary operation $\cdot$ on X that fulfills the following relations:

1. the map $\phi_x : X \to X : y \mapsto x \cdot y$ is bijective for all $x \in X$ and
2. for all $x, y, z \in X$:

$$(x \cdot y) \cdot (x \cdot z) = (y \cdot x) \cdot (y \cdot z). \qquad \text{(the cycloid equation)}$$

- Two cycle sets $(X, \cdot)$ and $(X, \times)$ are called isomorphic when there exists a bijection $f : X \to X$ such that $f(x \cdot y) = f(x) \times f(y)$.

CYCLE SETS

Cycle Sets

A cycle set $(X, \cdot)$ is a pair consisting of a non-empty set X and a binary operation $\cdot$ on X that fulfills the following relations:

1. the map $\phi_x : X \to X : y \mapsto x \cdot y$ is bijective for all $x \in X$ and
2. for all $x, y, z \in X$:

$$(x \cdot y) \cdot (x \cdot z) = (y \cdot x) \cdot (y \cdot z). \qquad \text{(the cycloid equation)}$$

- Two cycle sets $(X, \cdot)$ and $(X, \times)$ are called isomorphic when there exists a bijection $f : X \to X$ such that $f(x \cdot y) = f(x) \times f(y)$.
- A cycle set $(X, \cdot)$ is called non-degenerate if the map $T : X \to X : x \mapsto x \cdot x$ is bijective.

CYCLE SETS

RELATION TO SET-THEORETIC YBE

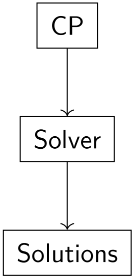
- Given:
  - the set $\mathcal{I}_n$ of finite, non-degenerate, involutive solutions to the YBE,
  - the set $\mathcal{C}_n$ of finite, non-degenerate cycle sets,
  - the map $F : \mathcal{I}_n \to \mathcal{C}_n, (X, r) \mapsto (X, \cdot)$ where:
    - $r(x, y) = (\sigma_x(y), \tau_y(x))$,
    - $x \cdot y = \tau_x^{-1}(y)$,
  - the map $G : \mathcal{C}_n \to \mathcal{I}_n, (X, \cdot) \mapsto (X, r)$ where:
    - $r(x, y) = ((y \diamond x) \cdot y, y \diamond x)$,
    - $x \diamond y = \phi_x^{-1}(y)$.

$$(X, r) \underset{G}{\overset{F}{\rightleftarrows}} (X, \cdot)$$

ENUMERATING SOLUTIONS

CONSTRAINT PROGRAMMING APPROACH

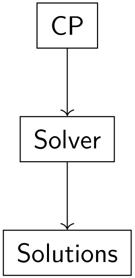
- Based on [AMV22].
- Model cycle set definition as a constraint problem (CP).
- Enumerate all solutions using a constraint solver.



ENUMERATING SOLUTIONS

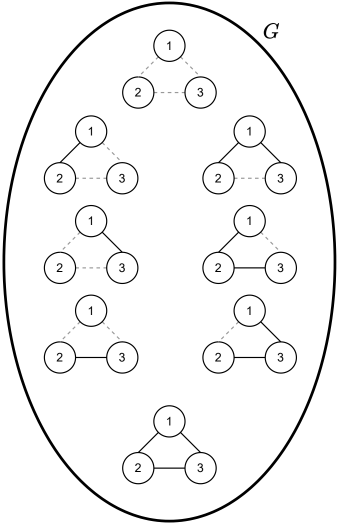
CONSTRAINT PROGRAMMING APPROACH

- Based on [AMV22].
- Model cycle set definition as a constraint problem (CP).
  - Add additional constraints to ensure that solutions are constructed up to isomorphism
- Enumerate all solutions using a constraint solver.



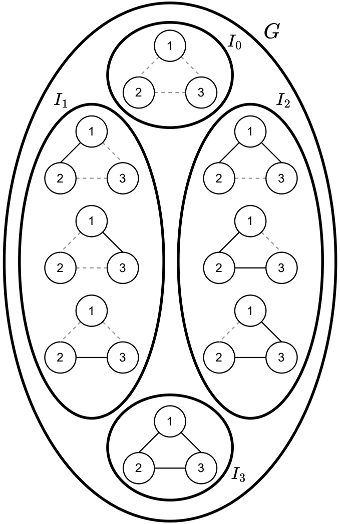
ENUMERATING UP TO ISOMORPHISM

INTERMEZZO



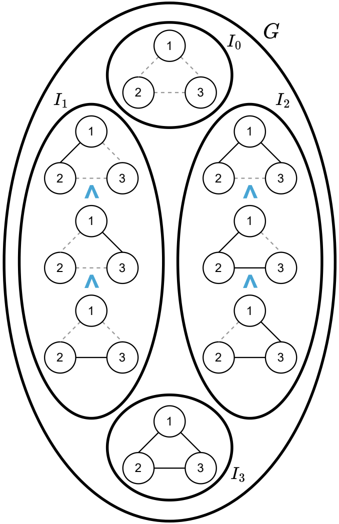
ENUMERATING UP TO ISOMORPHISM

INTERMEZZO



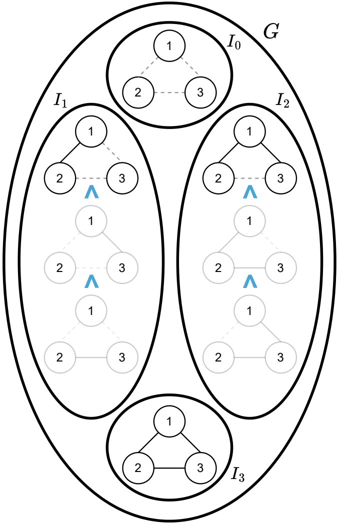
ENUMERATING UP TO ISOMORPHISM

INTERMEZZO



ENUMERATING UP TO ISOMORPHISM

INTERMEZZO



CONSTRAINT PROGRAMMING

Constraint Satisfaction Problem (CSP)

A CSP is a triple $\langle X, D, C\rangle$, where:

- $X = \{x_1, x_2, \ldots, x_n\}$ is a set of variables,
- $D = \{D_1, D_2, \ldots, D_n\}$ is a set of domains for these variables (i.e., $x_1 \in D_1, x_2 \in D_2, \ldots, x_n \in D_n$)
- and $C = \{C_1, \ldots, C_m\}$ is a set of constraints.

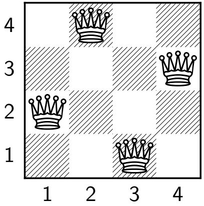
CONSTRAINT PROGRAMMING

Constraint Satisfaction Problem (CSP)

A CSP is a triple $\langle X, D, C\rangle$, where:

- $X = \{x_1, x_2, \ldots, x_n\}$ is a set of variables,
- $D = \{D_1, D_2, \ldots, D_n\}$ is a set of domains for these variables (i.e., $x_1 \in D_1, x_2 \in D_2, \ldots, x_n \in D_n$)
- and $C = \{C_1, \ldots, C_m\}$ is a set of constraints.

- Example: 4-Queens problem



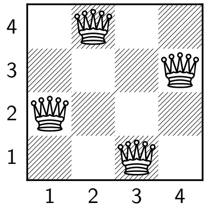
CONSTRAINT PROGRAMMING

Constraint Satisfaction Problem (CSP)

A CSP is a triple $\langle X, D, C \rangle$, where:

- $X = \{x_1, x_2, \ldots, x_n\}$ is a set of variables,
- $D = \{D_1, D_2, \ldots, D_n\}$ is a set of domains for these variables (i.e., $x_1 \in D_1, x_2 \in D_2, \ldots, x_n \in D_n$)
- and $C = \{C_1, \ldots, C_m\}$ is a set of constraints.

- Example: 4-Queens problem
  - $X = \{Q_i \mid i \in \{1, 2, 3, 4\}\}$



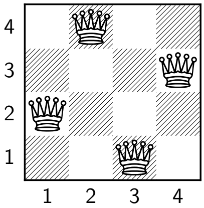
CONSTRAINT PROGRAMMING

Constraint Satisfaction Problem (CSP)

A CSP is a triple $\langle X, D, C \rangle$, where:

- $X = \{x_1, x_2, \ldots, x_n\}$ is a set of variables,
- $D = \{D_1, D_2, \ldots, D_n\}$ is a set of domains for these variables (i.e., $x_1 \in D_1, x_2 \in D_2, \ldots, x_n \in D_n$)
- and $C = \{C_1, \ldots, C_m\}$ is a set of constraints.

- Example: 4-Queens problem
  - $X = \{Q_i \mid i \in \{1, 2, 3, 4\}\}$
  - $D = \{D_i = \{1, 2, 3, 4\} \mid i \in \{1, 2, 3, 4\}\}$



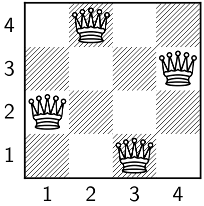
CONSTRAINT PROGRAMMING

Constraint Satisfaction Problem (CSP)

A CSP is a triple $\langle X, D, C \rangle$, where:

- $X = \{x_1, x_2, \ldots, x_n\}$ is a set of variables,
- $D = \{D_1, D_2, \ldots, D_n\}$ is a set of domains for these variables (i.e., $x_1 \in D_1, x_2 \in D_2, \ldots, x_n \in D_n$)
- and $C = \{C_1, \ldots, C_m\}$ is a set of constraints.

- Example: 4-Queens problem
  - $X = \{Q_i \mid i \in \{1,2,3,4\}\}$
  - $D = \{D_i = \{1,2,3,4\} \mid i \in \{1,2,3,4\}\}$
  - $C = \{Q_i \neq Q_j \wedge |Q_i - Q_j| \neq |i - j| \mid i, j \in \{1,2,3,4\}\}$



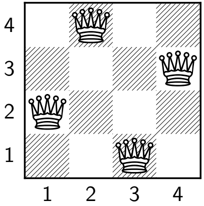
CONSTRAINT PROGRAMMING

Constraint Satisfaction Problem (CSP)

A CSP is a triple $\langle X, D, C \rangle$, where:

- $X = \{x_1, x_2, \ldots, x_n\}$ is a set of variables,
- $D = \{D_1, D_2, \ldots, D_n\}$ is a set of domains for these variables (i.e., $x_1 \in D_1, x_2 \in D_2, \ldots, x_n \in D_n$)
- and $C = \{C_1, \ldots, C_m\}$ is a set of constraints.

- Example: 4-Queens problem
  - $X = \{Q_i \mid i \in \{1, 2, 3, 4\}\}$
  - $D = \{D_i = \{1, 2, 3, 4\} \mid i \in \{1, 2, 3, 4\}\}$
  - $C = \{Q_i \neq Q_j \land |Q_i - Q_j| \neq |i - j| \mid i, j \in \{1, 2, 3, 4\}\}$
    - When enumerating: $C = C \cup$ constraints excluding previously found solutions



MODELLING CYCLE SET AS CP

- A cycle set $(X, \cdot)$ consists of a non-empty set X and a binary operation $\cdot$ on X s.t.:
  1. for all $x \in X$, the map $\phi_x : X \to X : y \mapsto x \cdot y$ is bijective,
  2. for all $x, y, z \in X$, $(x \cdot y) \cdot (x \cdot z) = (y \cdot x) \cdot (y \cdot z)$,
  3. the map $T : X \to X : x \mapsto x \cdot x$ is bijective. (non-degenerate)
- Each finite cycle set $(X, \cdot)$ can also be represented by a matrix $\mathbf{C}$ where
  - $\mathbf{C} \in X^{|X| \times |X|}$, and
  - $\mathbf{C}_{x,y} = x \cdot y$ for all $x, y \in X$.

MODELLING CYCLE SET AS CP

- A cycle set $(X, \cdot)$ consists of a non-empty set X and a binary operation $\cdot$ on X s.t.:
  1. for all $x \in X$, the map $\phi_x : X \to X : y \mapsto x \cdot y$ is bijective,
  2. for all $x, y, z \in X$, $(x \cdot y) \cdot (x \cdot z) = (y \cdot x) \cdot (y \cdot z)$,
  3. the map $T : X \to X : x \mapsto x \cdot x$ is bijective. (non-degenerate)

- Each finite cycle set $(X, \cdot)$ can also be represented by a matrix $\mathbf{C}$ where
  1. for all $x \in X$, $\mathbf{C}_{x,y} \neq \mathbf{C}_{x,z}$ for all $y, z \in X$ with $y \neq z$,
  2. for all $x, y, z \in X$, $\mathbf{C}_{\mathbf{C}_{x,y},\mathbf{C}_{x,z}} = \mathbf{C}_{\mathbf{C}_{y,x},\mathbf{C}_{y,z}}$,
  3. for all $x \in X$, $\mathbf{C}_{x,x} \neq \mathbf{C}_{y,y}$ for all $y \in X$ with $y \neq x$. (non-degenerate)

MODELLING CYCLE SET AS CP

ISOMORPHISMS

- Isomorphisms between cycle sets are well-defined,
- but what do they correspond to in the context of our CP model?

- The cycle sets $(X,\cdot)$, $(X,\times)$ are isomorphic if there exists a bijection $f: X \to X$ s.t. $f(x \cdot y) = f(x) \times f(y)$ for all $x, y \in X$.

- The cycle set matrices $C, C' \in X^{|X|\times|X|}$ are isomorphic if there exists a bijection $\pi: X \to X$ s.t. $\mathbf{C}'_{x,y} = \pi(\mathbf{C})_{x,y}$ for all $x, y \in X$, where $\pi(\mathbf{C})_{x,y} = \pi^{-1}(\mathbf{C}_{\pi(x),\pi(y)})$.

CYCLE SETS

LEXICOGRAPHIC ORDER AND PERMUTATIONS

- Given the cycle set $(\{1,2,3,4\},\cdot)$, its associated matrix $\mathbf{C}$ and an isomorphism $\pi=(12)$;
- we determine $\pi(\mathbf{C})=\pi^{-1}(\mathbf{C}_{\pi(i),\pi(j)})$.

$$\mathbf{C}=\begin{bmatrix}2&3&4&1\\4&1&2&3\\2&3&4&1\\4&1&2&3\end{bmatrix}$$

CYCLE SETS

LEXICOGRAPHIC ORDER AND PERMUTATIONS

- Given the cycle set $(\{1,2,3,4\},\cdot)$, its associated matrix $\mathbf{C}$ and an isomorphism $\pi = (12)$;
- we determine $\pi(\mathbf{C}) = \pi^{-1}(\mathbf{C}_{\pi(i),\pi(j)})$.

$$\mathbf{C} = \begin{bmatrix} 2 & 3 & 4 & 1 \\ 4 & 1 & 2 & 3 \\ 2 & 3 & 4 & 1 \\ 4 & 1 & 2 & 3 \end{bmatrix} \qquad \mathbf{C}_{\pi(i),\pi(j)} = \begin{bmatrix} 1 & 4 & 2 & 3 \\ 3 & 2 & 4 & 1 \\ 3 & 2 & 4 & 1 \\ 1 & 4 & 2 & 3 \end{bmatrix}$$

CYCLE SETS

LEXICOGRAPHIC ORDER AND PERMUTATIONS

- Given the cycle set $(\{1,2,3,4\},\cdot)$, its associated matrix $\mathbf{C}$ and an isomorphism $\pi = (12)$;
- we determine $\pi(\mathbf{C}) = \pi^{-1}(\mathbf{C}_{\pi(i),\pi(j)})$.

$$\mathbf{C} = \begin{bmatrix} 2 & 3 & 4 & 1 \\ 4 & 1 & 2 & 3 \\ 2 & 3 & 4 & 1 \\ 4 & 1 & 2 & 3 \end{bmatrix} \qquad \pi(\mathbf{C}) = \pi^{-1}(\mathbf{C}_{\pi(i),\pi(j)}) = \begin{bmatrix} 2 & 4 & 1 & 3 \\ 3 & 1 & 4 & 2 \\ 3 & 1 & 4 & 2 \\ 2 & 4 & 1 & 3 \end{bmatrix}$$

CYCLE SETS

LEXICOGRAPHIC ORDER AND PERMUTATIONS

- Given the cycle set $(\{1,2,3,4\},\cdot)$, its associated matrix $\mathbf{C}$ and an isomorphism $\pi = (12)$;
- $\mathbf{C}$ is lexicographically smaller than or equal to $\pi(\mathbf{C})$ (denoted $\mathbf{C} \preceq \pi(\mathbf{C})$) if

$$(\mathbf{C}_{1,1},\ldots,\mathbf{C}_{n,n}) \leq (\pi(\mathbf{C})_{1,1},\ldots,\pi(\mathbf{C})_{n,n})\,.$$

$$\mathbf{C} = \begin{bmatrix} 2 & 3 & 4 & 1 \\ 4 & 1 & 2 & 3 \\ 2 & 3 & 4 & 1 \\ 4 & 1 & 2 & 3 \end{bmatrix} \quad \preceq \quad \pi(\mathbf{C}) = \pi^{-1}(\mathbf{C}_{\pi(i),\pi(j)}) = \begin{bmatrix} 2 & 4 & 1 & 3 \\ 3 & 1 & 4 & 2 \\ 3 & 1 & 4 & 2 \\ 2 & 4 & 1 & 3 \end{bmatrix}$$

CYCLE SETS (UP TO ISOMORPHISM)

- A lexicographically minimal, finite, non-degenerate cycle set $(X, \cdot)$ can be represented by a matrix $\mathbf{C} \in X^{|X| \times |X|}$ where
  - for all $x \in X$, $\mathbf{C}_{x,y} \neq \mathbf{C}_{x,z}$ for all $y, z \in X$ with $y \neq z$
  - for all $x, y, z \in X$, $\mathbf{C}_{\mathbf{C}_{x,y}, \mathbf{C}_{x,z}} = \mathbf{C}_{\mathbf{C}_{y,x}, \mathbf{C}_{y,z}}$
  - for all $x \in X$, $\mathbf{C}_{x,x} \neq \mathbf{C}_{y,y}$ for all $y \in X$ with $y \neq x$
  - for all $\pi \in \mathcal{S}_n$, $\mathbf{C} \preceq \pi(\mathbf{C})$
    - the symmetry breaking constraints

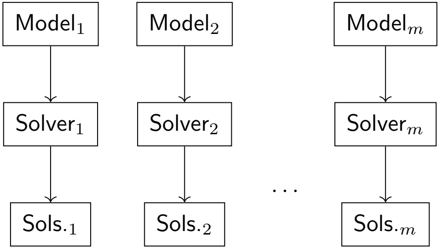
PARTITIONING THE PROBLEM

Lemma [AMV22]

Let $(X, \cdot)$ be a cycle set of size $n \in \mathbb{N}$.
Let $T : X \to X, T(x) \mapsto x \cdot x$ and $T_1 \in \mathcal{S}_n$.
If T and T_1 are conjugates, then there exists a cycle set structure $\times$ on X such that $(X, \cdot)$ and $(X, \times)$ are isomorphic and $T_1(x) = x \times x$ for all $x \in X$.

PARTITIONING THE PROBLEM

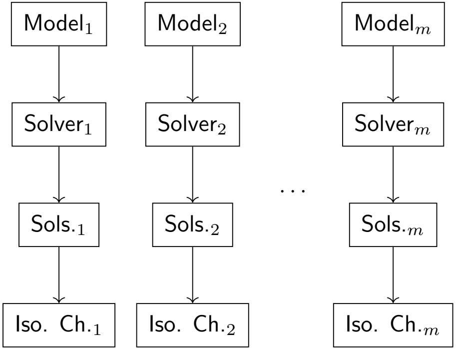
- As done by [AMV22]
- Model cycle set constraints.
- Partition the problem by fixing diagonals.
  - This decreases the search space per problem from $(n^2)^n$ to $(n^2 - n)^{(n-1)}$.
  - This allows to parallelize the search.
- Add static symmetry breaking constraints.



- Enumerate all solutions.

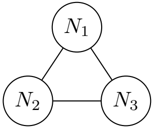
PARTITIONING THE PROBLEM

- As done by [AMV22]
- Model cycle set constraints.
- Partition the problem by fixing diagonals.
  - This decreases the search space per problem from $(n^2)^n$ to $(n^2 - n)^{(n-1)}$.
  - This allows to parallelize the search.
- Add static symmetry breaking constraints.
  - Complete symmetry breaking is unrealistic because of its encoding size...
- Enumerate all solutions.
- Perform final isomorphism check.

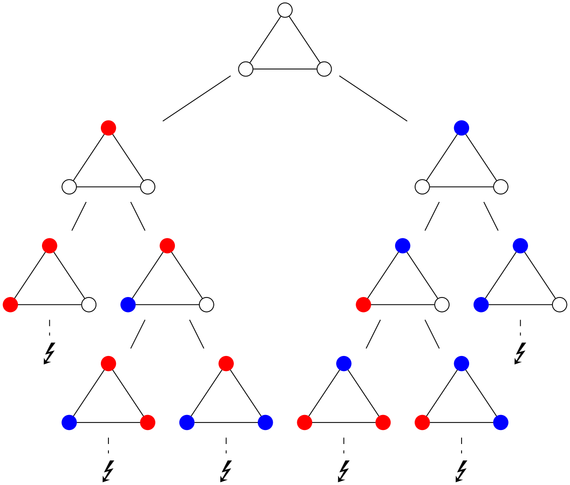


INTERMEZZO: (SAT) SOLVING

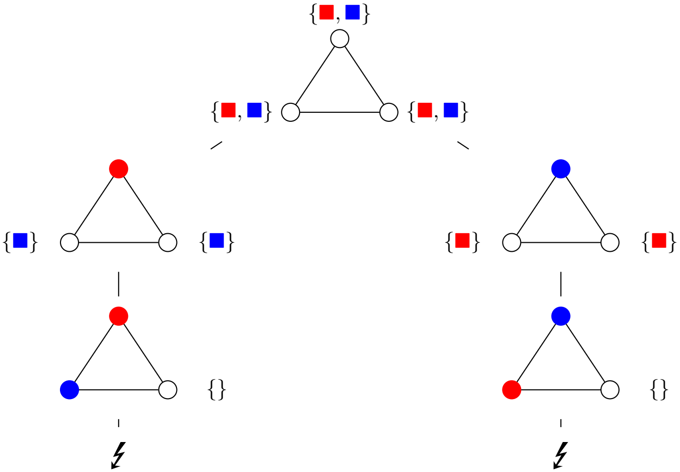
- Example: 2-colouring of a triangle graph
- Variables: $V = \{N_1, N_2, N_3\}$
- Domains: $D = \{D_i = \{$■, ■$\} \mid i \in \{1, 2, 3\}\}$
- Constraints: $C = \{N_1 \neq N_2, N_2 \neq N_3, N_3 \neq N_1\}$



INTERMEZZO: (SAT) SOLVING



INTERMEZZO: (SAT) SOLVING



INTERMEZZO: (SAT) SOLVING

- Example: a Boolean satisfaction (SAT) problem
- Variables: $V = \{a, b, c, d, r, s, w, x, y, z\}$
- Domains: $D = \{D_i = \{0, 1\} \mid i \in V\}$
- Constraints: $C = \{(r) \wedge (\neg r \vee s) \wedge (\neg w \vee a) \wedge (\neg x \vee b) \wedge (\neg y \vee \neg z \vee c) \wedge (\neg b \vee \neg c \vee d)\}$

INTERMEZZO: (SAT) SOLVING

$$(r) \wedge (\neg r \vee s) \wedge (\neg w \vee a) \wedge (\neg x \vee b) \wedge (\neg y \vee \neg z \vee c) \wedge (\neg b \vee \neg c \vee d)$$

Current assignment $\alpha = \{ \quad \}$

INTERMEZZO: (SAT) SOLVING

$$(r) \wedge (\neg r \vee s) \wedge (\neg w \vee a) \wedge (\neg x \vee b) \wedge (\neg y \vee \neg z \vee c) \wedge (\neg b \vee \neg c \vee d)$$

Current assignment $\alpha = \{r = 1 \qquad\qquad\qquad \}$

INTERMEZZO: (SAT) SOLVING

$$(\quad s) \wedge (\neg w \vee a) \wedge (\neg x \vee b) \wedge (\neg y \vee \neg z \vee c) \wedge (\neg b \vee \neg c \vee d)$$

Current assignment $\alpha = \{r = 1, s = 1 \quad \}$

INTERMEZZO: (SAT) SOLVING

$$(\neg w \vee a) \wedge (\neg x \vee b) \wedge (\neg y \vee \neg z \vee c) \wedge (\neg b \vee \neg c \vee d)$$

Current assignment $\alpha = \{r = 1, s = 1, a = 1\}$

INTERMEZZO: (SAT) SOLVING

$$(\neg x \vee b) \wedge (\neg y \vee \neg z \vee c) \wedge (\neg b \vee \neg c \vee d)$$

Current assignment $\alpha = \{r = 1, s = 1, a = 1\}$

INTERMEZZO: (SAT) SOLVING

$$(r) \wedge (\neg r \vee s) \wedge (\neg w \vee a) \wedge (\neg x \vee b) \wedge (\neg y \vee \neg z \vee c) \wedge (\neg b \vee \neg c \vee d)$$

Current assignment $\alpha = \{r = 1, s = 1, a = 1\}$

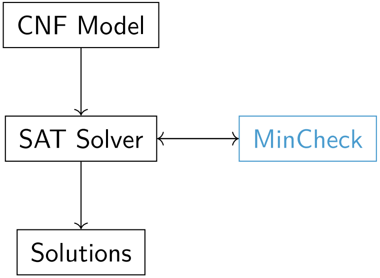
- Check if the partial solution can be extended to a solution that is lexicographically minimal
- If this is certainly not the case: force the solver to backtrack!
  - for example by adding a falsified clause: $(\neg a)$

SAT MODULO SYMMETRIES [KS21]

- Main goal:
  - Enumerate satisfying assignments of CNF up to isomorphism
  - First used to enumerate graphs with certain interesting properties
- Core idea:
  1. Model the mathematical problem at hand using propositional logic
  2. Force a SAT solver to generate only non-isomorphic solutions during the search

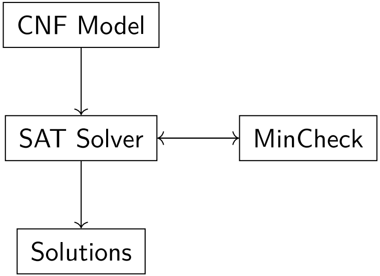
SAT MODULO SYMMETRIES [KS21]

- How:
  1. Obtain a partial interpretation from the SAT solver.
  2. Check whether the assignment can be extended to a complete assignment that is lexicographically minimal.
  3. If not, force the solver to abort the current branch of the search tree by learning a new clause.



SAT MODULO SYMMETRIES [KS21]

- How:
  1. Obtain a partial interpretation from the SAT solver.
  2. Check whether the assignment can be extended to a complete assignment that is lexicographically minimal.
  3. If not, force the solver to abort the current branch of the search tree by learning a new clause.
- This procedure needs to take into account:
  - the (encoding of) the mathematical problem,
    - i.e., the (encoding of) the cycle set definition.
  - and the structure of the set of isomorphisms.
    - i.e., all permutations that fix the diagonal.



SAT MODULO SYMMETRIES [KS21]

FOR CYCLE SETS

- Given a formula ψ over variables Σ (modelling the cycle set definition),
- With a complete, satisfying assignment α of Σ, we associate a cycle set $\mathbf{C}^{\alpha}$ where for all cells $(i, j) \in X \times X$ it holds that:
  - $\mathbf{C}_{i,j} = k$ iff $v_{i,j,k} \in \alpha$.
- We now want to introduce symmetry breaking constraints during the solving phase.
- But, during the solving phase, the full cycle set might not be known yet.
- Hence, we introduce partial cycle sets.

PARTIAL CYCLE SETS

Partial Cycle Set

A partial cycle set of size n is a matrix $\mathbf{P} \in (2^X)^{n \times n}$ with $X = \{1, \ldots, n\}$, where each cell $c \in X \times X$ of $\mathbf{P}$ represents a non-empty domain $\mathbf{P}_c \subseteq X$ of values that are still possible.

PARTIAL CYCLE SETS

Partial Cycle Set

A partial cycle set of size n is a matrix $\mathbf{P} \in (2^X)^{n \times n}$ with $X = \{1, \dots, n\}$, where each cell $c \in X \times X$ of $\mathbf{P}$ represents a non-empty domain $\mathbf{P}_c \subseteq X$ of values that are still possible.

- With a partial assignment α of Σ, we associate a partial cycle set $\mathbf{P}^\alpha$ where for all cells $(x, y) \in X \times X$ it holds that:
  - $\mathbf{P}_{x,y} = \{x \in X \mid \neg c_{i,j,x} \notin \alpha\}$.
- In other words, $\mathbf{P}^\alpha$ consist of the values that can still be true according to α.

PARTIAL CYCLE SETS

EXAMPLE

	1	2	3
1	$\{2\}$	$\{1\}$	$\{3\}$
2	$\{2,3\}$	$\{1\}$	$\{2,3\}$
3	$\{1,2\}$	$\{1,2\}$	$\{3\}$

$\mathbf{P} \in (2^X)^{n \times n}$

	1	2	3
1	2	1	3
2	2	1	3
3	1	2	3

$\mathbf{C}_1 \in \mathcal{X}(\mathbf{P})$

	1	2	3
1	2	1	3
2	3	1	2
3	2	1	3

$\mathbf{C}_2 \in \mathcal{X}(\mathbf{P})$

$\dots$

PARTIAL CYCLE SETS

LEXICOGRAPHIC MINIMALITY [KS21]

- A partial cycle set $\mathbf{P}$ is $\preceq$-minimal if it can be extended to a $\preceq$-minimal cycle set.
- If for all extended cycle sets $\mathbf{C} \in \mathcal{X}(\mathbf{P})$ there exists an isomorphism π s.t. $\pi(\mathbf{C}) \prec \mathbf{C}$:
    - $\mathbf{P}$ can not be $\preceq$-minimal.
    - But: hard to decide this...

PARTIAL CYCLE SETS

LEXICOGRAPHIC MINIMALITY [KS21]

- A partial cycle set $\mathbf{P}$ is $\preceq$-minimal if it can be extended to a $\preceq$-minimal cycle set.
- If for all extended cycle sets $\mathbf{C} \in \mathcal{X}(\mathbf{P})$ there exists an isomorphism π s.t. $\pi(\mathbf{C}) \prec \mathbf{C}$:
  - $\mathbf{P}$ can not be $\preceq$-minimal.
  - But: hard to decide this...
- If there exists an isomorphism π s.t. $\pi(\mathbf{C}) \prec \mathbf{C}$ for all extended cycle sets $\mathbf{C} \in \mathcal{X}(\mathbf{P})$:
  - $\mathbf{P}$ can not be $\preceq$-minimal.
  - We call π a witness of non-minimality of $\mathbf{P}$!

FINDING WITNESSES

- In order to find these witnesses, we need:
  1. A way to apply permutations to partial cycle sets.
  2. An order $\lhd$ over partial cycle sets,
     - s.t. if $\mathbf{P} \lhd \mathbf{P}'$ then $\mathbf{C} \prec \mathbf{C}'$ for all extensions $\mathbf{C} \in \mathcal{X}(\mathbf{P})$ and $\mathbf{C}' \in \mathcal{X}(\mathbf{P}')$.

- If we can find a permutation π for which $\pi(\mathbf{P}) \lhd \mathbf{P}$, we have that $\pi(\mathbf{C}) \prec \mathbf{C}$ for all extensions $\mathbf{C} \in \mathcal{X}(\mathbf{P})$.
- In other words, we can decide that π is a witness of non-minimality.

PARTIAL CYCLE SET

APPLYING A PERMUTATION

- Given a partial cycle set $\mathbf{P} \in (2^X)^{n \times n}$ and a permutation $\pi : X \to X$:

$$\pi(\mathbf{P}_{i,j}) = \{\pi^{-1}(x) \mid x \in \mathbf{P}_{\pi(i),\pi(j)}\}.$$

- For example, given $\mathbf{P}$ and $\pi = (12)$:

$$\mathbf{P} = \begin{bmatrix} 1 & 2 & 4 & 3 \\ 1 & 2 & 3 & 4 \\ 1 & \{2,4\} & 3 & \{2,4\} \\ 1 & \{2,3\} & \{2,3\} & 4 \end{bmatrix} \qquad \mathbf{P}_{\pi(i),(j)} = \begin{bmatrix} 2 & 1 & 3 & 4 \\ 2 & 1 & 4 & 3 \\ \{2,4\} & 1 & 3 & \{2,4\} \\ \{2,3\} & 1 & \{2,3\} & 4 \end{bmatrix}$$

PARTIAL CYCLE SET

APPLYING A PERMUTATION

- Given a partial cycle set $\mathbf{P} \in (2^X)^{n \times n}$ and a permutation $\pi : X \to X$:

$$\pi(\mathbf{P}_{i,j}) = \{\pi^{-1}(x) \mid x \in \mathbf{P}_{\pi(i),\pi(j)}\}.$$

- For example, given $\mathbf{P}$ and $\pi = (12)$:

$$\mathbf{P} = \begin{bmatrix} 1 & 2 & 4 & 3 \\ 1 & 2 & 3 & 4 \\ 1 & \{2,4\} & 3 & \{2,4\} \\ 1 & \{2,3\} & \{2,3\} & 4 \end{bmatrix} \qquad \pi(\mathbf{P}) = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 4 & 3 \\ \{1,4\} & 2 & 3 & \{1,4\} \\ \{1,3\} & 2 & \{1,3\} & 4 \end{bmatrix}$$

PARTIAL CYCLE SET

APPLYING A PERMUTATION

- Given a partial cycle set $\mathbf{P} \in (2^X)^{n\times n}$ and a permutation $\pi : X \to X$:

$$\pi(\mathbf{P}_{i,j}) = \{\pi^{-1}(x) \mid x \in \mathbf{P}_{\pi(i),\pi(j)}\}.$$

- For example, given $\mathbf{P}$ and $\pi = (12)$:

$$\mathbf{P} = \begin{bmatrix} 1 & 2 & 4 & 3 \\ 1 & 2 & 3 & 4 \\ 1 & \{2,4\} & 3 & \{2,4\} \\ 1 & \{2,3\} & \{2,3\} & 4 \end{bmatrix} \quad \triangleright ? \quad \pi(\mathbf{P}) = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 4 & 3 \\ \{1,4\} & 2 & 3 & \{1,4\} \\ \{1,3\} & 2 & \{1,3\} & 4 \end{bmatrix}$$

PARTIAL CYCLE SET

APPLYING A PERMUTATION

- Given a partial cycle set $\mathbf{P} \in (2^X)^{n \times n}$ and a permutation $\pi : X \to X$:

$$\pi(\mathbf{P}_{i,j}) = \{\pi^{-1}(x) \mid x \in \mathbf{P}_{\pi(i),\pi(j)}\}.$$

- For example, given $\mathbf{P}$ and $\pi = (12)$:

$$\mathbf{P} = \begin{bmatrix} 1 & \{3,4\} & 2 & \{3,4\} \\ 1 & 2 & 3 & 4 \\ 1 & \{2,4\} & 3 & \{2,4\} \\ 1 & \{2,3\} & \{2,3\} & 4 \end{bmatrix} \quad \triangleright? \quad \pi(\mathbf{P}) = \begin{bmatrix} 1 & 2 & 3 & 4 \\ \{3,4\} & 2 & 1 & \{3,4\} \\ \{1,4\} & 2 & 3 & \{1,4\} \\ \{1,3\} & 2 & \{1,3\} & 4 \end{bmatrix}$$

PARTIAL CYCLE SET

ORDERING PARTIAL CYCLE SETS

$\mathbf{P} \lhd \mathbf{P}'$

Given two partial cycle sets $\mathbf{P}$ and $\mathbf{P}'$ we say that $\mathbf{P} \lhd \mathbf{P}'$ iff:

- there is a cell c s.t. $\max \mathbf{P}_c < \min \mathbf{P}'_c$ and
- for all cells $c' < c$: $\max \mathbf{P}_{c'} \leq \min \mathbf{P}'_{c'}$.

PARTIAL CYCLE SET

ORDERING PARTIAL CYCLE SETS

$\mathbf{P} \lhd \mathbf{P}'$

Given two partial cycle sets $\mathbf{P}$ and $\mathbf{P}'$ we say that $\mathbf{P} \lhd \mathbf{P}'$ iff:

- there is a cell c s.t. $\max \mathbf{P}_c < \min \mathbf{P}'_c$ and
- for all cells $c' < c$: $\max \mathbf{P}_{c'} \leq \min \mathbf{P}'_{c'}$.

$\mathbf{P} \unlhd \mathbf{P}'$

Given two partial cycle sets $\mathbf{P}$ and $\mathbf{P}'$ we say that $\mathbf{P} \unlhd \mathbf{P}'$ iff:

- either $\mathbf{P} \lhd \mathbf{P}'$, or for all cells c: $\max \mathbf{P}_c \leq \min \mathbf{P}'_c$.

PARTIAL CYCLE SET

ORDERING PARTIAL CYCLE SETS

- Using this order we have that:
  - If $\mathbf{P} \lhd \mathbf{P}'$, then for all extended cycle sets $\mathbf{C} \in \mathcal{X}(\mathbf{P})$ and $\mathbf{C}' \in \mathcal{X}(\mathbf{P}')$, it holds that $\mathbf{C} \prec \mathbf{C}'$.
  - If $\pi(\mathbf{P}) \lhd \mathbf{P}$, π is a witness of non-minimality.

MINIMALITY CHECK

OVERVIEW

- Goal: decide whether $\exists \pi \in \langle \Pi \rangle$, such that $\pi(\mathbf{P}) \lhd \mathbf{P}$, given
  - a matrix $\mathbf{P}$ representing a partial cycle set, and
  - where the group $\langle \Pi \rangle$ represents the isomorphisms of the problem.
- Considering each π one-by-one is not a feasible option...

MINIMALITY CHECK

BACKTRACKING APPROACH [KS21]

- Represent all possible isomorphism of the problem.
  - i.e. $\pi(1) = [1, 2, 3, 4], \pi(2) = [1, 2, 3, 4], \dots$
- Make decision
  - i.e. $\pi(1) = [2]$
- Propagate:
  - i.e. $\pi(2) = [1, 3, 4]$
    - Ensure that the partial permutation π can be extended to an isomorphism of the problem.
    - Given the partial cycle set $\mathbf{P}$, ensure that $\pi(\mathbf{P}) \lhd \mathbf{P}$.
- Repeat until:
  - A witness is found.
  - All possibilities have failed.

MINIMALITY CHECK

BACKTRACKING APPROACH [KS21]

- Represent all possible isomorphism of the problem.
  - i.e. $\pi(1) = [1, 2, 3, 4], \pi(2) = [1, 2, 3, 4], \dots$
- Make decision
  - i.e. $\pi(1) = [2]$
- Propagate:
  - i.e. $\pi(2) = [1, 3, 4]$
    - Ensure that the partial permutation π can be extended to an isomorphism of the problem.
    - Given the partial cycle set $\mathbf{P}$, ensure that $\pi(\mathbf{P}) \lhd \mathbf{P}$.
- Repeat until:
  - A witness is found.
  - All possibilities have failed.

- Issue!
- Sometimes there is no information to propagate
- Worst case complexity of $n!$...

MINIMALITY CHECK

ISOMORPHISMS AND FIXED DIAGONALS

- Ensure that the partial permutation π can be extended to an isomorphism of the problem
- The group of isomorphisms $\langle \Pi \rangle$ is given by:
  - If no diagonal is fixed, $\langle \Pi \rangle = \mathcal{S}_n$
  - If a diagonal T is fixed, $\langle \Pi \rangle = C_{\mathcal{S}_n}(T)$
    - i.e., the isomorphisms are permutations that fix the diagonal

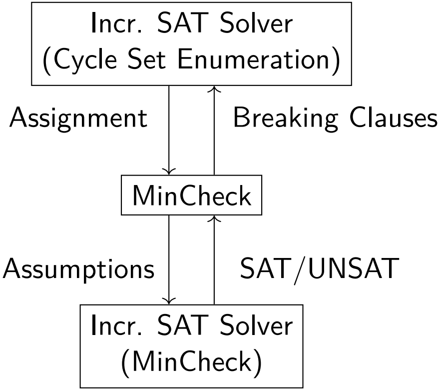
MINIMALITY CHECK

INCREMENTAL, SAT-BASED APPROACH

- Minimality check = combinatorial search problem
  - i.e. given the current (partial) cycle set, does there exist a witness?
- We chose to:
  - Express the problem in CNF.
  - Use an incremental SAT-solver to verify whether the CNF is satisfiable given the current assumptions.
  - If so, we have found a witness of non-minimality for the current cycle set!

INCREMENTAL, SAT-BASED MINIMALITY CHECK

OVERVIEW



MINIMALITY CHECK

CONSTRUCTING A CLAUSE

- π is a witness of non-minimality!
  - There exists cell $c = (i, j)$ such that:
    - for all cells $c' < c$: $\pi(\mathbf{P})_{c'} \trianglelefteq \mathbf{P}_{c'}$ and,
    - $\pi(\mathbf{P})_c \triangleleft \mathbf{P}_c$.
- So: how do we exclude the current solution (and its extensions?)

MINIMALITY CHECK

CONSTRUCTING A CLAUSE

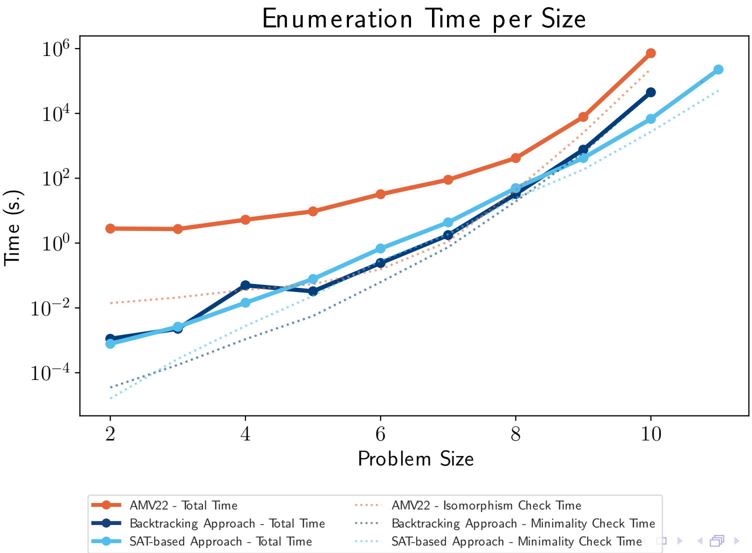
- π is a witness of non-minimality!
  - There exists cell $c = (i, j)$ such that:
    - for all cells $c' < c$: $\pi(\mathbf{P})_{c'} \trianglelefteq \mathbf{P}_{c'}$ and,
    - $\pi(\mathbf{P})_c \triangleleft \mathbf{P}_c$.
- So: how do we exclude the current solution (and its extensions?)

- We add a clause expressing that (at least) one of these conditions is different:
  - $\max \pi(\mathbf{P})_c$ becomes larger than or equal to $\min \mathbf{P}_c$,
  - or for at least one of the cells $c' < c$; $\max \pi(\mathbf{P})_{c'}$ becomes strictly larger than $\min \mathbf{P}_{c'}$,
  - or the solver needs to backtrack.

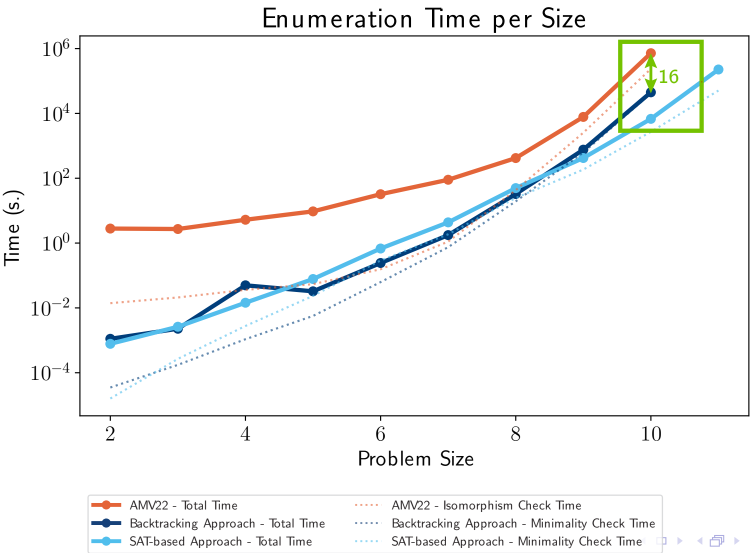
IMPLEMENTATION

- We use CaDiCaL [BFFH20] with the IPASIR-UP API [FNP+23];
  - to keep track of the current assignment,
  - to add clauses if a useful permutation is found,
  - and to find witnesses.
- The implementation and database are available on GitLab.
- Experiments were performed on a machine with
  - an AMD(R) Genoa-X CPU,
  - running Rocky Linux 8.9,
  - with Linux kernel 4.18.0.

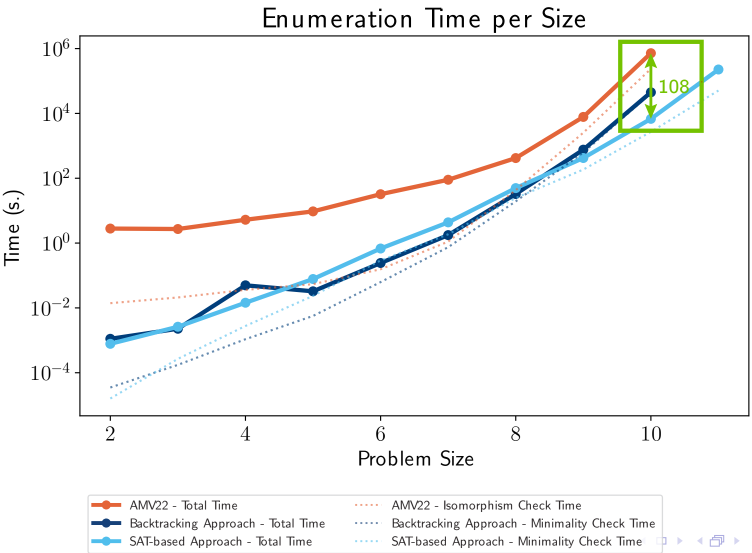
COMPARING RESULTS



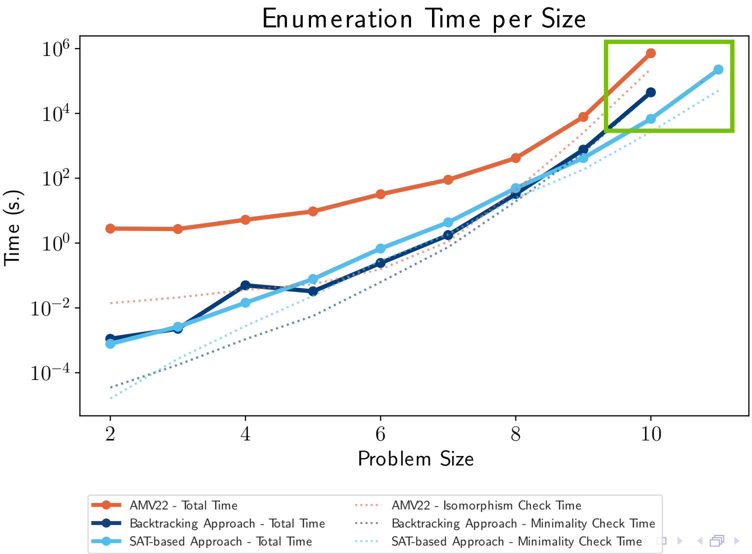
COMPARING RESULTS



COMPARING RESULTS



COMPARING RESULTS



COMPARING RESULTS

		AMV22		Backtracking Approach			Incr. SAT Approach		
Size	# Sols	Iso Check (s.)	Total (s.)	MinCheck (s.)	Total (s.)	Speedup	MinCheck (s.)	Total (s.)	Speedup
2	2	0.0	2.8	0.0	**0.0**		0.0	**0.0**	
3	5	0.0	2.7	0.0	**0.0**		0.0	**0.0**	
4	23	0.0	5.2	0.0	**0.0**		0.0	**0.0**	
5	88	0.0	9.5	0.0	**0.0**		0.0	**0.0**	
6	595	0.2	32.2	0.1	**0.2**	161.0	0.3	0.7	46.0
7	3 456	1.1	89.8	0.7	**1.8**	49.9	1.9	4.4	20.4
8	34 530	43.1	419.3	19.3	**32.6**	12.9	24.6	49.7	8.4
9	321 931	2 542.3	7 797.7	621.6	760.5	10.2	185.6	**421.2**	18.51
10	4 895 272	237 307.1	720 883.0	41 594.1	44 792.5	16.1	2 706.3	**6796.8**	108.1
11	77 182 093						50 767.2	**226 395.6**	

Table: Comparing the runtimes of the implementation of AMV22 and our approaches building on SAT Modulo Symmetries.

FUTURE WORK

- Refining incremental approach

FUTURE WORK

- Refining incremental approach
- Certifying the results
  - We obtain the same results as [AMV22], but that only means that we are either both correct or both wrong.
  - However, how do we verify this?

FUTURE WORK

- Refining incremental approach
- Certifying the results
  - We obtain the same results as [AMV22], but that only means that we are either both correct or both wrong.
  - However, how do we verify this?

- Enumerating related structures
  - Racks,
    - used to enumerate skew cycle sets.
  - Skew Cycle Sets,
    - correspond to non-degenerate set-theoretic solutions.
  - Biquandles,
    - applications in knot theory.

FUTURE WORK

- Refining incremental approach
- Certifying the results
  - We obtain the same results as [AMV22], but that only means that we are either both correct or both wrong.
  - However, how do we verify this?

- Enumerating related structures
  - Racks,
    - used to enumerate skew cycle sets.
  - Skew Cycle Sets,
    - correspond to non-degenerate set-theoretic solutions.
  - Biquandles,
    - applications in knot theory.
- Generelizing the approach?

CONCLUSION

- We have reproduced the results from [AMV22] with a significant speedup
- We have expanded these results to include size 11
- We did this by extending the SMS-framework [KS21] to reason about (partially constructed) cycle sets
- The current technique can be adapted to enumerate related mathematical structures

REFERENCES

[AMV22] Özgür Akgün, M. Mereb, and Leandro Vendramin. Enumeration of set-theoretic solutions to the yang–baxter equation. *Mathematics of Computation*, jan 2022.

[Bax72] Rodney J. Baxter. Partition function of the eight-vertex lattice model. *Annals of Physics*, 70(1):193–228, 1972.

[BFFH20] Armin Biere, Katalin Fazekas, Mathias Fleury, and Maximillian Heisinger. CaDiCaL, Kissat, Paracooba, Plingeling and Treengeling entering the SAT Competition 2020. In Tomas Balyo, Nils Froleyks, Marijn Heule, Markus Iser, Matti Järvisalo, and Martin Suda, editors, *Proc. of SAT Competition 2020 – Solver and Benchmark Descriptions*, volume B-2020-1 of *Department of Computer Science Report Series B*, pages 51–53. University of Helsinki, 2020.

[BGMN22] Bart Bogaerts, Stephan Gocht, Ciaran McCreesh, and Jakob Nordström. Certified symmetry and dominance breaking for combinatorial optimisation. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 3698–3707. AAAI Press, 2022.

[Dri92] V. G. Drinfeld. *On some unsolved problems in quantum group theory*, page 1–8. Springer Berlin Heidelberg, 1992.

REFERENCES

[FNP+23] Katalin Fazekas, Aina Niemetz, Mathias Preiner, Markus Kirchweger, Stefan Szeider, and Armin Biere. IPASIR-UP: user propagators for CDCL. In Meena Mahajan and Friedrich Slivovsky, editors, *26th International Conference on Theory and Applications of Satisfiability Testing, SAT 2023, July 4-8, 2023, Alghero, Italy*, volume 271 of *LIPIcs*, pages 8:1–8:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[KS21] Markus Kirchweger and Stefan Szeider. SAT modulo symmetries for graph generation. In Laurent D. Michel, editor, *27th International Conference on Principles and Practice of Constraint Programming, CP 2021, Montpellier, France (Virtual Conference), October 25-29, 2021*, volume 210 of *LIPIcs*, pages 34:1–34:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[KSS22] Markus Kirchweger, Manfred Scheucher, and Stefan Szeider. A SAT attack on rota's basis conjecture. In Kuldeep S. Meel and Ofer Strichman, editors, *25th International Conference on Theory and Applications of Satisfiability Testing, SAT 2022, August 2-5, 2022, Haifa, Israel*, volume 236 of *LIPIcs*, pages 4:1–4:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[Yan67] Chen Ning Yang. Some exact results for the many-body problem in one dimension with repulsive delta-function interaction. *Phys. Rev. Lett.*, 19:1312–1315, Dec 1967.

YANG-BAXTER EQUATION

DEFINITION

Yang-Baxter Equation [Yan67, Bax72]

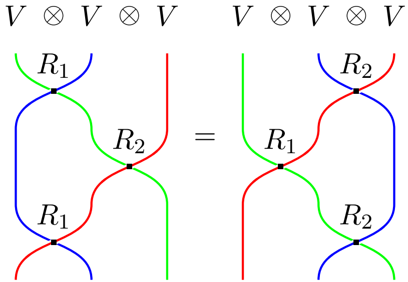
A solution to the Yang-Baxter equation (YBE) is a pair (V, R), where V is a vector space and $R : V \otimes V \to V \otimes V$ is a map such that in $(V \otimes V \otimes V)$,

$$R_1 R_2 R_1 = R_2 R_1 R_2,$$

where R_i acts as R on components i and $i+1$, and as the identity on the other component.

YANG-BAXTER EQUATION

DEFINITION



$$R_1 R_2 R_1 = R_2 R_1 R_2$$

Figure: A visual representation of the Yang-Baxter equation.

YANG-BAXTER EQUATION

DEFINITION

Set-Theoretic Yang-Baxter Equation (YBE) [Dri92]

A set-theoretic solution to the YBE is a pair (X, r), where X is a non-empty set and $r : X^2 \to X^2$ is a map such that in X^3,

$$r_1 r_2 r_1 = r_2 r_1 r_2, \qquad \text{(the Yang-Baxter Equation)}$$

where r_i acts as r on components i and $i+1$ and as the identity on the other component.

- These solutions are a subset of the solutions to the *original* Yang-Baxter equation.
- Given a set-theoretic solution (X, r), we can construct a solution to the *original* YBE through linearisation.
- A set-theoretic solution is called involutive if $r^2 = id_{X \times X}$.

SET-THEORETIC YANG-BAXTER EQUATION

EXAMPLE

- (X, r) with
  - $X = \mathbb{Z}/n\mathbb{Z}$
  - $r(x, y) = (y + 1, x - 1)$
- Finite:
  - the set X is finite

SET-THEORETIC YANG-BAXTER EQUATION

EXAMPLE

- (X, r) with
  - $X = \mathbb{Z}/n\mathbb{Z}$
  - $r(x, y) = (y + 1, x - 1)$
- Finite
- Involutive:
  - $r^2(x, y) = (x, y)$
  - $r(r(x, y)) = r(y + 1, x - 1) =$ $((x - 1) + 1, (y + 1) - 1) = (x, y)$

SET-THEORETIC YANG-BAXTER EQUATION

EXAMPLE

- (X, r) with
  - $X = \mathbb{Z}/n\mathbb{Z}$
  - $r(x, y) = (y + 1, x - 1)$
- Finite
- Involutive
- Non-degenerate:
  - Given $r(x, y) = (\sigma_x(y), \tau_y(x))$, the maps σ_x, τ_x are bijective for all $x \in X$
  - $\sigma(y) = y + 1$ and $\tau(x) = x - 1$ are bijective

SET-THEORETIC YANG-BAXTER EQUATION

EXAMPLE

- (X, r) with
  - $X = \mathbb{Z}/n\mathbb{Z}$
  - $r(x, y) = (y + 1, x - 1)$
- Finite
- Involutive
- Non-degenerate

- (X, s) with
  - $X = \mathbb{Z}/n\mathbb{Z}$
  - $s(x, y) = (y - 1, x + 1)$
- Finite
- Involutive:
  - $s(s(x, y)) = s(y - 1, x + 1) = ((x + 1) - 1, (y - 1) + 1) = (x, y)$

SET-THEORETIC YANG-BAXTER EQUATION

EXAMPLE

- (X, r) with
  - $X = \mathbb{Z}/n\mathbb{Z}$
  - $r(x, y) = (y + 1, x - 1)$
- Finite
- Involutive
- Non-degenerate

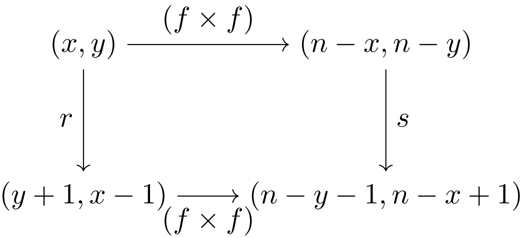
- (X, s) with
  - $X = \mathbb{Z}/n\mathbb{Z}$
  - $s(x, y) = (y - 1, x + 1)$
- Finite
- Involutive
- Non-degenerate:
  - $\sigma(y) = y - 1$ and $\tau(x) = x + 1$ are bijective

SET-THEORETIC YANG-BAXTER EQUATION

EXAMPLE

- (X, r) with
  - $X = \mathbb{Z}/n\mathbb{Z}$
  - $r(x, y) = (y + 1, x - 1)$
- (X, s) with
  - $X = \mathbb{Z}/n\mathbb{Z}$
  - $s(x, y) = (y - 1, x + 1)$

- The solutions (X, r) and (X, s) are isomorphic
  - i.e. there exists a bijection $f : X \to X$ such that $(f \times f)r = s(f \times f)$
- The isomorphism is defined by $f : X \to X, x \mapsto n - x$



CNF MODEL

- for each $i, j, x \in X$, the Boolean variable $v_{i,j,x}$ is true iff $\mathbf{C}_{i,j} = x$
- Ensure that each matrix entry is assigned exactly one value;
  - for each $i, j \in X$: $\texttt{exactlyOne}([v_{i,j,k} \mid k \in X])$

CNF MODEL

- for each $i, j, x \in X$, the Boolean variable $v_{i,j,x}$ is true iff $\mathbf{C}_{i,j} = x$
- Ensure that each matrix entry is assigned exactly one value;
  - for each $i, j \in X$: $\texttt{exactlyOne}([v_{i,j,k} \mid k \in X])$
- Rows contain unique values;
  - for each $i, k \in X$: $\texttt{exactlyOne}([v_{i,j,k} \mid j \in X])$

CNF MODEL

- for each $i, j, x \in X$, the Boolean variable $v_{i,j,x}$ is true iff $\mathbf{C}_{i,j} = x$
- Ensure that each matrix entry is assigned exactly one value;
  - for each $i, j \in X$: $\texttt{exactlyOne}([v_{i,j,k} \mid k \in X])$
- Rows contain unique values;
  - for each $i, k \in X$: $\texttt{exactlyOne}([v_{i,j,k} \mid j \in X])$
- The diagonal contains unique values;
  - $\texttt{exactlyOne}([v_{i,i,k} \mid i \in X])$

CNF MODEL

- for each $i, j, x \in X$, the Boolean variable $v_{i,j,x}$ is true iff $\mathbf{C}_{i,j} = x$
- Ensure that each matrix entry is assigned exactly one value;
  - for each $i, j \in X$: `exactlyOne`$([v_{i,j,k} \mid k \in X])$
- Rows contain unique values;
  - for each $i, k \in X$: `exactlyOne`$([v_{i,j,k} \mid j \in X])$
- The diagonal contains unique values;
  - `exactlyOne`$([v_{i,i,k} \mid i \in X])$

- for all $i, j, k, b \in X$ with $i < j$, the Boolean variable $y_{i,j,k,b}$ is true iff $\mathbf{C}_{\mathbf{C}_{i,j},\mathbf{C}_{i,k}} = \mathbf{C}_{\mathbf{C}_{k,i},\mathbf{C}_{k,j}} = b$

CNF MODEL

- for each $i, j, x \in X$, the Boolean variable $v_{i,j,x}$ is true iff $\mathbf{C}_{i,j} = x$
- Ensure that each matrix entry is assigned exactly one value;
  - for each $i, j \in X$: `exactlyOne`$([v_{i,j,k} \mid k \in X])$
- Rows contain unique values;
  - for each $i, k \in X$: `exactlyOne`$([v_{i,j,k} \mid j \in X])$
- The diagonal contains unique values;
  - `exactlyOne`$([v_{i,i,k} \mid i \in X])$

- for all $i, j, k, b \in X$ with $i < j$, the Boolean variable $y_{i,j,k,b}$ is true iff $\mathbf{C}_{\mathbf{C}_{i,j},\mathbf{C}_{i,k}} = \mathbf{C}_{\mathbf{C}_{k,i},\mathbf{C}_{k,j}} = b$
  - for all $i, j, k, x, y, b \in X$ where $i < j$:
    - $\neg v_{i,j,x} \vee \neg v_{i,k,y} \vee \neg v_{x,y,b} \vee y_{i,j,k,b}$
    - $\neg v_{j,i,x} \vee \neg v_{j,k,y} \vee \neg v_{x,y,b} \vee y_{i,j,k,b}$

CNF MODEL

- for each $i, j, x \in X$, the Boolean variable $v_{i,j,x}$ is true iff $\mathbf{C}_{i,j} = x$
- Ensure that each matrix entry is assigned exactly one value;
  - for each $i, j \in X$: $\texttt{exactlyOne}([v_{i,j,k} \mid k \in X])$
- Rows contain unique values;
  - for each $i, k \in X$: $\texttt{exactlyOne}([v_{i,j,k} \mid j \in X])$
- The diagonal contains unique values;
  - $\texttt{exactlyOne}([v_{i,i,k} \mid i \in X])$

- for all $i, j, k, b \in X$ with $i < j$, the Boolean variable $y_{i,j,k,b}$ is true iff $\mathbf{C}_{\mathbf{C}_{i,j},\mathbf{C}_{i,k}} = \mathbf{C}_{\mathbf{C}_{k,i},\mathbf{C}_{k,j}} = b$
  - for all $i, j, k, x, y, b \in X$ where $i < j$:
    - $\neg v_{i,j,x} \vee \neg v_{i,k,y} \vee \neg v_{x,y,b} \vee y_{i,j,k,b}$
    - $\neg v_{j,i,x} \vee \neg v_{j,k,y} \vee \neg v_{x,y,b} \vee y_{i,j,k,b}$
  - for all $i, j, k \in X$ where $i < j$:
    - $\texttt{exactlyOne}([y_{i,j,k,b} \mid b \in X])$

MINIMALITY CHECK

ISOMORPHISMS AND FIXED DIAGONALS

- Ensure that the partial permutation π can be extended to an isomorphism of the problem
- The group of isomorphisms $\langle \Pi \rangle$ is given by:
  - If no diagonal is fixed, $\langle \Pi \rangle = \mathcal{S}_n$
  - If a diagonal T is fixed, $\langle \Pi \rangle = C_{\mathcal{S}_n}(T)$
    - i.e., the isomorphisms are permutations that fix the diagonal

MINIMALITY CHECK

ISOMORPHISMS AND FIXED DIAGONALS

- Ensure that the partial permutation π can be extended to an isomorphism of the problem
- The group of isomorphisms $\langle \Pi \rangle$ is given by:
  - If no diagonal is fixed, $\langle \Pi \rangle = \mathcal{S}_n$
  - If a diagonal T is fixed, $\langle \Pi \rangle = C_{\mathcal{S}_n}(T)$
    - i.e., the isomorphisms are permutations that fix the diagonal

- We want to enumerate cycle sets of size 10 (i.e. $X = \{1, 2, \ldots, 10\}$) with fixed diagonal $T = [2, 3, 1, 5, 6, 4, 8, 7, 10, 9]$

MINIMALITY CHECK

ISOMORPHISMS AND FIXED DIAGONALS

- Ensure that the partial permutation π can be extended to an isomorphism of the problem
- The group of isomorphisms $\langle \Pi \rangle$ is given by:
  - If no diagonal is fixed, $\langle \Pi \rangle = \mathcal{S}_n$
  - If a diagonal T is fixed, $\langle \Pi \rangle = C_{\mathcal{S}_n}(T)$
    - i.e., the isomorphisms are permutations that fix the diagonal
- We want to enumerate cycle sets of size 10 (i.e. $X = \{1, 2, \ldots, 10\}$) with fixed diagonal $T = [2, 3, 1, 5, 6, 4, 8, 7, 10, 9]$
- To determine $\langle \Pi \rangle$ we rewrite T as a permutation:
  - $T = (123)(456)(78)(9a)$ where $a = 10$
  - Note that $(123) = (231) = (321)$

MINIMALITY CHECK

ISOMORPHISMS AND FIXED DIAGONALS

- Ensure that the partial permutation π can be extended to an isomorphism of the problem
- The group of isomorphisms $\langle \Pi \rangle$ is given by:
  - If no diagonal is fixed, $\langle \Pi \rangle = \mathcal{S}_n$
  - If a diagonal T is fixed, $\langle \Pi \rangle = C_{\mathcal{S}_n}(T)$
    - i.e., the isomorphisms are permutations that fix the diagonal

- We want to enumerate cycle sets of size 10 (i.e. $X = \{1, 2, \ldots, 10\}$) with fixed diagonal $T = [2, 3, 1, 5, 6, 4, 8, 7, 10, 9]$
- To determine $\langle \Pi \rangle$ we rewrite T as a permutation:
  - $T = (123)(456)(78)(9a)$ where $a = 10$
  - Note that $(123) = (231) = (321)$
- For a permutation to fix T, it needs to:
  - map cycles to cycles of the same length,
    - i.e. $\pi(1) \in \{1, 2, 3, 4, 5, 6\}$

MINIMALITY CHECK

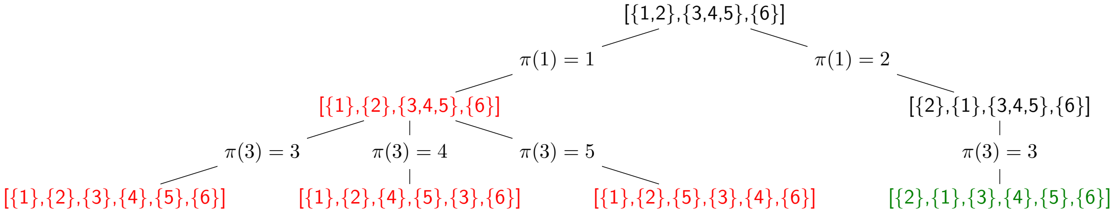
ISOMORPHISMS AND FIXED DIAGONALS

- Ensure that the partial permutation π can be extended to an isomorphism of the problem
- The group of isomorphisms $\langle\Pi\rangle$ is given by:
  - If no diagonal is fixed, $\langle\Pi\rangle = \mathcal{S}_n$
  - If a diagonal T is fixed, $\langle\Pi\rangle = C_{\mathcal{S}_n}(T)$
    - i.e., the isomorphisms are permutations that fix the diagonal

- We want to enumerate cycle sets of size 10 (i.e. $X = \{1, 2, \dots, 10\}$) with fixed diagonal $T = [2, 3, 1, 5, 6, 4, 8, 7, 10, 9]$
- To determine $\langle\Pi\rangle$ we rewrite T as a permutation:
  - $T = (123)(456)(78)(9a)$ where $a = 10$
  - Note that $(123) = (231) = (321)$
- For a permutation to fix T, it needs to:
  - map cycles to cycles of the same length,
    - i.e. $\pi(1) \in \{1, 2, 3, 4, 5, 6\}$
  - while maintaining the order between the elements of the cycle
    - i.e. if $\pi(1) = 5$, it should follow that $\pi(2) = 6$ and $\pi(3) = 4$

EXAMPLE

SEARCH TREE



MINIMALITY CHECK

CONSTRUCTING A CLAUSE, EXAMPLE

$$\bigvee_{x \geq \min \mathbf{P}_c} v_{\pi(c),x} \vee \bigvee_{x \leq \max \pi(\mathbf{P})_c} v_{c,x} \vee$$

$$\bigvee_{c' < c} \left(\bigvee_{x > \min \mathbf{P}_{c'}} v_{\pi(c'),x} \vee \bigvee_{x < \max \pi(\mathbf{P})_{c'}} v_{c',x} \right)$$

$$\mathbf{P} = \begin{bmatrix} 2 & 1 & 3 & 4 & 6 & 5 \\ 2 & 1 & 3 & 4 & 5 & 6 \\ 1 & 2 & 4 & * & * & * \\ * & * & * & 5 & * & * \\ * & * & * & * & 3 & * \\ * & * & * & * & * & 6 \end{bmatrix}$$

$$v_{2,5,6} \vee v_{1,5,5} \vee v_{1,5,4} \vee v_{1,5,3} \vee v_{1,5,2} \vee v_{1,5,1} \vee$$
$$v_{2,2,3} \vee v_{2,2,4} \vee v_{2,2,5} \vee v_{2,2,6} \vee v_{1,1,1} \vee$$
$$v_{2,1,2} \vee v_{2,1,3} \vee v_{2,1,4} \vee v_{2,1,5} \vee v_{2,1,6} \vee$$
$$v_{2,3,4} \vee v_{2,3,5} \vee v_{2,3,6} \vee v_{1,3,2} \vee v_{1,3,1} \vee$$
$$v_{2,4,5} \vee v_{2,4,6} \vee v_{1,4,3} \vee v_{1,4,2} \vee v_{1,4,1}$$

$$\pi(\mathbf{P}) = \begin{bmatrix} 2 & 1 & 3 & 4 & 5 & 6 \\ 2 & 1 & 3 & 4 & 6 & 5 \\ 1 & 2 & 4 & * & * & * \\ * & * & * & 5 & * & * \\ * & * & * & * & 3 & * \\ * & * & * & * & * & 6 \end{bmatrix}$$

MINIMALITY CHECK

CONSTRUCTING A CLAUSE, EXAMPLE

$$\bigvee_{x \geq \min \mathbf{P}_c} v_{\pi(c),x} \vee \bigvee_{x \leq \max \pi(\mathbf{P})_c} v_{c,x} \vee$$

$$\bigvee_{c' < c} \left(\bigvee_{x > \min \mathbf{P}_{c'}} v_{\pi(c'),x} \vee \bigvee_{x < \max \pi(\mathbf{P})_{c'}} v_{c',x} \right)$$

$$\mathbf{P} = \begin{bmatrix} 2 & 1 & 3 & 4 & 6 & 5 \\ 2 & 1 & 3 & 4 & 5 & 6 \\ 1 & 2 & 4 & * & * & * \\ * & * & * & 5 & * & * \\ * & * & * & * & 3 & * \\ * & * & * & * & * & 6 \end{bmatrix}$$

$$v_{2,5,6} \vee v_{1,5,5} \vee v_{1,5,4} \vee v_{1,5,3} \vee v_{1,5,2} \vee v_{1,5,1} \vee$$
$$v_{2,2,3} \vee v_{2,2,4} \vee v_{2,2,5} \vee v_{2,2,6} \vee v_{1,1,1} \vee$$
$$v_{2,1,2} \vee v_{2,1,3} \vee v_{2,1,4} \vee v_{2,1,5} \vee v_{2,1,6} \vee$$
$$v_{2,3,4} \vee v_{2,3,5} \vee v_{2,3,6} \vee v_{1,3,2} \vee v_{1,3,1} \vee$$
$$v_{2,4,5} \vee v_{2,4,6} \vee v_{1,4,3} \vee v_{1,4,2} \vee v_{1,4,1}$$

$$\pi(\mathbf{P}) = \begin{bmatrix} 2 & 1 & 3 & 4 & 5 & 6 \\ 2 & 1 & 3 & 4 & 6 & 5 \\ 1 & 2 & 4 & * & * & * \\ * & * & * & 5 & * & * \\ * & * & * & * & 3 & * \\ * & * & * & * & * & 6 \end{bmatrix}$$

MINIMALITY CHECK

CONSTRUCTING A CLAUSE, EXAMPLE

$$\bigvee_{x \geq \min \mathbf{P}_c} v_{\pi(c),x} \vee \bigvee_{x \leq \max \pi(\mathbf{P})_c} v_{c,x} \vee$$

$$\bigvee_{c' < c} \left(\bigvee_{x > \min \mathbf{P}_{c'}} v_{\pi(c'),x} \vee \bigvee_{x < \max \pi(\mathbf{P})_{c'}} v_{c',x} \right)$$

$$\mathbf{P} = \begin{bmatrix} 2 & 1 & 3 & 4 & 6 & 5 \\ 2 & 1 & 3 & 4 & 5 & 6 \\ 1 & 2 & 4 & * & * & * \\ * & * & * & 5 & * & * \\ * & * & * & * & 3 & * \\ * & * & * & * & * & 6 \end{bmatrix}$$

$v_{2,5,6} \vee v_{1,5,5} \vee v_{1,5,4} \vee v_{1,5,3} \vee v_{1,5,2} \vee v_{1,5,1} \vee$

$v_{2,2,3} \vee v_{2,2,4} \vee v_{2,2,5} \vee v_{2,2,6} \vee v_{1,1,1} \vee$

$v_{2,1,2} \vee v_{2,1,3} \vee v_{2,1,4} \vee v_{2,1,5} \vee v_{2,1,6} \vee$

$v_{2,3,4} \vee v_{2,3,5} \vee v_{2,3,6} \vee v_{1,3,2} \vee v_{1,3,1} \vee$

$v_{2,4,5} \vee v_{2,4,6} \vee v_{1,4,3} \vee v_{1,4,2} \vee v_{1,4,1}$

$$\pi(\mathbf{P}) = \begin{bmatrix} 2 & 1 & 3 & 4 & 5 & 6 \\ 2 & 1 & 3 & 4 & 6 & 5 \\ 1 & 2 & 4 & * & * & * \\ * & * & * & 5 & * & * \\ * & * & * & * & 3 & * \\ * & * & * & * & * & 6 \end{bmatrix}$$

MINIMALITY CHECK

CONSTRUCTING A CLAUSE, EXAMPLE

$$\bigvee_{x \geq \min \mathbf{P}_c} v_{\pi(c),x} \vee \bigvee_{x \leq \max \pi(\mathbf{P})_c} v_{c,x} \vee$$

$$\bigvee_{c' < c} \left(\bigvee_{x > \min \mathbf{P}_{c'}} v_{\pi(c'),x} \vee \bigvee_{x < \max \pi(\mathbf{P})_{c'}} v_{c',x} \right)$$

$$\mathbf{P} = \begin{bmatrix} 2 & 1 & 3 & 4 & 6 & 5 \\ 2 & 1 & 3 & 4 & 5 & 6 \\ 1 & 2 & 4 & * & * & * \\ * & * & * & 5 & * & * \\ * & * & * & * & 3 & * \\ * & * & * & * & * & 6 \end{bmatrix}$$

$$\pi(\mathbf{P}) = \begin{bmatrix} 2 & 1 & 3 & 4 & 5 & 6 \\ 2 & 1 & 3 & 4 & 6 & 5 \\ 1 & 2 & 4 & * & * & * \\ * & * & * & 5 & * & * \\ * & * & * & * & 3 & * \\ * & * & * & * & * & 6 \end{bmatrix}$$

$v_{2,5,6} \vee v_{1,5,5} \vee v_{1,5,4} \vee v_{1,5,3} \vee v_{1,5,1} \vee$
$v_{2,1,3} \vee v_{2,1,4} \vee v_{2,1,5} \vee v_{2,1,6} \vee$
$v_{2,3,4} \vee v_{2,3,5} \vee v_{2,3,6} \vee v_{1,3,2} \vee v_{1,3,1} \vee$
$v_{2,4,5} \vee v_{2,4,6} \vee v_{1,4,3} \vee v_{1,4,1}$

MINIMALITY CHECK

CONSTRUCTING A CLAUSE, EXAMPLE

$$\bigvee_{x \geq \min \mathbf{P}_c} v_{\pi(c),x} \vee \bigvee_{x \leq \max \pi(\mathbf{P})_c} v_{c,x} \vee$$

$$\bigvee_{c' < c} \left(\bigvee_{x > \min \mathbf{P}_{c'}} v_{\pi(c'),x} \vee \bigvee_{x < \max \pi(\mathbf{P})_{c'}} v_{c',x} \right)$$

$$\mathbf{P} = \begin{bmatrix} 2 & 1 & 3 & 4 & 6 & 5 \\ 2 & 1 & 3 & 4 & 5 & 6 \\ 1 & 2 & 4 & * & * & * \\ * & * & * & 5 & * & * \\ * & * & * & * & 3 & * \\ * & * & * & * & * & 6 \end{bmatrix}$$

$v_{2,5,6} \vee \neg v_{1,5,6} \vee$

$\neg v_{2,1,1} \vee$

$v_{2,3,4} \vee v_{2,3,5} \vee v_{2,3,6} \vee v_{1,3,2} \vee v_{1,3,1} \vee$

$v_{2,4,5} \vee v_{2,4,6} \vee v_{1,4,3} \vee v_{1,4,1}$

$$\pi(\mathbf{P}) = \begin{bmatrix} 2 & 1 & 3 & 4 & 5 & 6 \\ 2 & 1 & 3 & 4 & 6 & 5 \\ 1 & 2 & 4 & * & * & * \\ * & * & * & 5 & * & * \\ * & * & * & * & 3 & * \\ * & * & * & * & * & 6 \end{bmatrix}$$

FUTURE WORK

THE ENUMERATION OF RELATED STRUCTURES

- We have now enumerated finite, involutive, non-degenerate set-theoretic solutions to the YBE
- What about related structures?
- Only minimal adjustments are needed to enumerate:
  - Racks,
    - used to enumerate skew cycle sets
  - Skew Cycle Sets,
    - correspond to finite, non-degenerate set-theoretic solutions
  - Biquandles,
    - finite, non-degenerate, involutive, set-theoretic solutions
  - ...

FUTURE WORK

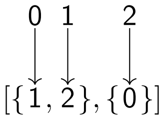
CERTIFYING THE RESULS

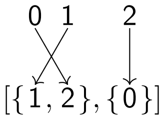
- How do we know whether these results are correct?
  - We obtain the same results as [AMV22], but that only means that we are either both correct or both wrong.
- Many SAT Solvers are verifiable
  - They produce a solution and a machine-verifiable proof for this solution
  - This proof is then verified together with the CNF formula
- This is also the case for `CaDiCaL`, even with the `SMS` framework [KSS22]
  - However: only verified if each clause is added with a good reason
- So, how do we know whether the added breaking clauses were correct?
  - `VeriPB` can verify static symmetry breaking [BGMN22]
  - `CaDiCaL` comes with `VeriPB`
- How do we verify whether we have enumerated exactly one solution per isomorphism class?
  - Non-trivial, we need information about the problem. . .
  - The symmetries of the CNF might not be equivalent to the isomorphisms of the problem...

MINIMALITY CHECK

ORDERED PARTITIONS

- An ordered partition $(X_1, X_2, \ldots, X_r)$ represents all permutations s.t.:
  - $\pi^{-1}(x_1) < \pi^{-1}(x_2)$ for all $x_1 \in X_i, x_2 \in X_j$ with $i < j$.





$\pi_1 : 0 \mapsto 1, 1 \mapsto 2, 2 \mapsto 0$

$\pi_2 : 0 \mapsto 2, 1 \mapsto 1, 2 \mapsto 0$

MINIMALITY CHECK

ITERATIVE REFINEMENT OF ORDERED PARTITION

- Start with ordered partition based on the isomorphisms of the problem.
- Make decision (i.e. split partition into a singleton and the rest).
- Propagate:
  - Ensure that the partial permutation π can be extended to an isomorphism of the problem.
  - Given the partial cycle set $\mathbf{P}$, ensure that $\pi(\mathbf{P}) \trianglelefteq \mathbf{P}$.
- Repeat until:
  - A witness or refining subset is found.
  - All possibilities have failed.

MINIMALITY CHECK

EXAMPLE

$$\mathbf{P} = \begin{bmatrix} 2 & 1 & 3 & 4 & 6 & 5 \\ 2 & 1 & 3 & 4 & 5 & 6 \\ 1 & 2 & 4 & \{5,6\} & \{3,5,6\} & \{3,5,6\} \\ \{2,3,4,6\} & \{2,3,4,6\} & \{1,2,3,4,6\} & 5 & \{1,2,3,4,6\} & \{1,2,3,4,6\} \\ \{4,5,6\} & \{1,2,4,5,6\} & \{1,2,4,5,6\} & \{1,2,4,5,6\} & 3 & \{1,2,4,5,6\} \\ \{3,4,5\} & \{2,3,4,5\} & \{1,2,3,4,5\} & \{1,2,3,4,5\} & \{1,2,3,4,5\} & 6 \end{bmatrix}$$

- Diagonal $T = (12)(345)(6)$ is fixed.
  - If a diagonal T is fixed, $\langle \Pi \rangle = C_{\mathcal{S}_n}(T)$.
  - Initial permutation: $\pi = [\{1,2\},\{3,4,5\},\{6\}]$.
- Does there exist an extension of π that can be used to exclude or refine $\mathbf{P}$?

MINIMALITY CHECK

EXAMPLE

$$[\{1,2\},\{3,4,5\},\{6\}]$$

$$\mathbf{P} = \begin{bmatrix} 2 & 1 & 3 & 4 & 6 & 5 \\ 2 & 1 & 3 & 4 & 5 & 6 \\ 1 & 2 & 4 & * & * & * \\ * & * & * & 5 & * & * \\ * & * & * & * & 3 & * \\ * & * & * & * & * & 6 \end{bmatrix}$$

$$\pi(\mathbf{P}) = \begin{bmatrix} 2 & * & * & * & * & * \\ * & 1 & * & * & * & * \\ * & * & 4 & * & * & * \\ * & * & * & 5 & * & * \\ * & * & * & * & 3 & * \\ * & * & * & * & * & 6 \end{bmatrix}$$

MINIMALITY CHECK

EXAMPLE

$$[\{1,2\},\{3,4,5\},\{6\}]$$

$$\Big\downarrow \text{Decide } 1 \mapsto 1$$

$$[\{1\},\{2\},\{3,4,5\},\{6\}]$$

$$\mathbf{P} = \begin{bmatrix} 2 & 1 & 3 & 4 & 6 & 5 \\ 2 & 1 & 3 & 4 & 5 & 6 \\ 1 & 2 & 4 & * & * & * \\ * & * & * & 5 & * & * \\ * & * & * & * & 3 & * \\ * & * & * & * & * & 6 \end{bmatrix}$$

$$\pi(\mathbf{P}) = \begin{bmatrix} 2 & 1 & * & * & * & * \\ 2 & 1 & * & * & * & * \\ * & * & 4 & * & * & * \\ * & * & * & 5 & * & * \\ * & * & * & * & 3 & * \\ * & * & * & * & * & 6 \end{bmatrix}$$

MINIMALITY CHECK

EXAMPLE

$$[\{1,2\},\{3,4,5\},\{6\}]$$

$\downarrow$ Decide $1 \mapsto 1$

$$[\{1\},\{2\},\{3,4,5\},\{6\}]$$

$\downarrow$ Decide $3 \mapsto 4$

$$[\{1\},\{2\},\{4\},\{5,3\},\{6\}]$$

$$\mathbf{P} = \begin{bmatrix} 2 & 1 & 3 & 4 & 6 & 5 \\ 2 & 1 & 3 & 4 & 5 & 6 \\ 1 & 2 & 4 & * & * & * \\ * & * & * & 5 & * & * \\ * & * & * & * & 3 & * \\ * & * & * & * & * & 6 \end{bmatrix}$$

$$\pi(\mathbf{P}) = \begin{bmatrix} 2 & 1 & 3 & * & * & * \\ 2 & 1 & 3 & * & * & * \\ * & * & 4 & * & * & * \\ * & * & * & 5 & * & * \\ * & * & * & * & 3 & * \\ * & * & * & * & * & 6 \end{bmatrix}$$

MINIMALITY CHECK

EXAMPLE

$$[\{1,2\},\{3,4,5\},\{6\}]$$

↓ Decide $1 \mapsto 1$

$$[\{1\},\{2\},\{3,4,5\},\{6\}]$$

↓ Decide $3 \mapsto 4$

$$[\{1\},\{2\},\{4\},\{5,3\},\{6\}]$$

↓ Propagate

$$[\{1\},\{2\},\{4\},\{5\},\{3\},\{6\}]$$

$$\mathbf{P} = \begin{bmatrix} 2 & 1 & 3 & 4 & 6 & 5 \\ 2 & 1 & 3 & 4 & 5 & 6 \\ 1 & 2 & 4 & * & * & * \\ * & * & * & 5 & * & * \\ * & * & * & * & 3 & * \\ * & * & * & * & * & 6 \end{bmatrix}$$

$$\pi(\mathbf{P}) = \begin{bmatrix} 2 & 1 & 3 & 6 & 5 & 4 \\ 2 & 1 & 3 & 4 & 5 & 6 \\ * & * & 4 & * & * & * \\ * & * & * & 5 & * & * \\ 1 & 2 & * & * & 3 & * \\ * & * & * & * & * & 6 \end{bmatrix}$$

MINIMALITY CHECK

EXAMPLE

$$[\{1,2\},\{3,4,5\},\{6\}]$$

$\downarrow$ Decide $1 \mapsto 1$

$$[\{1\},\{2\},\{3,4,5\},\{6\}]$$

$\downarrow$ Decide $3 \mapsto 4$

$$[\{1\},\{2\},\{4\},\{5,3\},\{6\}]$$

$\downarrow$ Propagate

$$[\{1\},\{2\},\{4\},\{5\},\{3\},\{6\}]$$

$$\mathbf{P} = \begin{bmatrix} 2 & 1 & 3 & 4 & 6 & 5 \\ 2 & 1 & 3 & 4 & 5 & 6 \\ 1 & 2 & 4 & * & * & * \\ * & * & * & 5 & * & * \\ * & * & * & * & 3 & * \\ * & * & * & * & * & 6 \end{bmatrix}$$

$$\pi(\mathbf{P}) = \begin{bmatrix} 2 & 1 & 3 & 6 & 5 & 4 \\ 2 & 1 & 3 & 4 & 5 & 6 \\ * & * & 4 & * & * & * \\ * & * & * & 5 & * & * \\ 1 & 2 & * & * & 3 & * \\ * & * & * & * & * & 6 \end{bmatrix}$$

MINIMALITY CHECK

EXAMPLE

$$[\{1,2\},\{3,4,5\},\{6\}]$$

$$\downarrow \text{Decide } 1 \mapsto 1$$

$$[\{1\},\{2\},\{3,4,5\},\{6\}]$$

$$\mathbf{P} = \begin{bmatrix} 2 & 1 & 3 & 4 & 6 & 5 \\ 2 & 1 & 3 & 4 & 5 & 6 \\ 1 & 2 & 4 & * & * & * \\ * & * & * & 5 & * & * \\ * & * & * & * & 3 & * \\ * & * & * & * & * & 6 \end{bmatrix}$$

$$\pi(\mathbf{P}) = \begin{bmatrix} 2 & * & * & * & * & * \\ * & 1 & * & * & * & * \\ * & * & 4 & * & * & * \\ * & * & * & 5 & * & * \\ * & * & * & * & 3 & * \\ * & * & * & * & * & 6 \end{bmatrix}$$

MINIMALITY CHECK

EXAMPLE

$$[\{1,2\},\{3,4,5\},\{6\}]$$

$\downarrow$ Decide $1 \mapsto 1$

$$[\{1\},\{2\},\{3,4,5\},\{6\}]$$

$\downarrow$ Decide and propagate

$$\dots$$

$$\mathbf{P} = \begin{bmatrix} 2 & 1 & 3 & 4 & 6 & 5 \\ 2 & 1 & 3 & 4 & 5 & 6 \\ 1 & 2 & 4 & * & * & * \\ * & * & * & 5 & * & * \\ * & * & * & * & 3 & * \\ * & * & * & * & * & 6 \end{bmatrix}$$

$$\pi(\mathbf{P}) = \begin{bmatrix} 2 & * & * & * & * & * \\ * & 1 & * & * & * & * \\ * & * & 4 & * & * & * \\ * & * & * & 5 & * & * \\ * & * & * & * & 3 & * \\ * & * & * & * & * & 6 \end{bmatrix}$$

MINIMALITY CHECK

EXAMPLE

$$[\{1,2\},\{3,4,5\},\{6\}]$$

$$\mathbf{P} = \begin{bmatrix} 2 & 1 & 3 & 4 & 6 & 5 \\ 2 & 1 & 3 & 4 & 5 & 6 \\ 1 & 2 & 4 & * & * & * \\ * & * & * & 5 & * & * \\ * & * & * & * & 3 & * \\ * & * & * & * & * & 6 \end{bmatrix}$$

$$\pi(\mathbf{P}) = \begin{bmatrix} 2 & * & * & * & * & * \\ * & 1 & * & * & * & * \\ * & * & 4 & * & * & * \\ * & * & * & 5 & * & * \\ * & * & * & * & 3 & * \\ * & * & * & * & * & 6 \end{bmatrix}$$

MINIMALITY CHECK

EXAMPLE

$$[\{1,2\},\{3,4,5\},\{6\}]$$

$\downarrow$ Decide $1 \mapsto 2$

$$[\{2\},\{1\},\{3,4,5\},\{6\}]$$

$$\mathbf{P} = \begin{bmatrix} 2 & 1 & 3 & 4 & 6 & 5 \\ 2 & 1 & 3 & 4 & 5 & 6 \\ 1 & 2 & 4 & * & * & * \\ * & * & * & 5 & * & * \\ * & * & * & * & 3 & * \\ * & * & * & * & * & 6 \end{bmatrix}$$

$$\pi(\mathbf{P}) = \begin{bmatrix} 2 & 1 & * & * & * & * \\ 2 & 1 & * & * & * & * \\ * & * & 4 & * & * & * \\ * & * & * & 5 & * & * \\ * & * & * & * & 3 & * \\ * & * & * & * & * & 6 \end{bmatrix}$$

MINIMALITY CHECK

EXAMPLE

$$[\{1,2\},\{3,4,5\},\{6\}]$$

$\downarrow$ Decide $1 \mapsto 2$

$$[\{2\},\{1\},\{3,4,5\},\{6\}]$$

$\downarrow$ Decide $3 \mapsto 3$ and propagate

$$[\{2\},\{1\},\{3\},\{4\},\{5\},\{6\}]$$

$$\mathbf{P} = \begin{bmatrix} 2 & 1 & 3 & 4 & 6 & 5 \\ 2 & 1 & 3 & 4 & 5 & 6 \\ 1 & 2 & 4 & * & * & * \\ * & * & * & 5 & * & * \\ * & * & * & * & 3 & * \\ * & * & * & * & * & 6 \end{bmatrix}$$

$$\pi(\mathbf{P}) = \begin{bmatrix} 2 & 1 & 3 & 4 & 5 & 6 \\ 2 & 1 & 3 & 4 & 6 & 5 \\ 1 & 2 & 4 & * & * & * \\ * & * & * & 5 & * & * \\ * & * & * & * & 3 & * \\ * & * & * & * & * & 6 \end{bmatrix}$$