Static Symmetry and Dominance Breaking for Pseudo-Boolean Optimization

Daimy Van Caudenberg and Bart Bogaerts

Vrije Universiteit Brussel

Benelux Conference on Artificial Intelligence '22

ARTIFICIAL
INTELLIGENCE
RESEARCH GROUP

WHAT IS SYMMETRY?

Context

Combinatorial search and optimization problems

Assign values to variables so that constraints are satisfied and objective is minimized

WHAT IS SYMMETRY?

Context

Combinatorial search and optimization problems

Assign values to variables so that constraints are satisfied and objective is minimized

- Symmetry: Permutation of variables that preserves the solutions.
- Occurs e.g. when sets of variables in the problem are interchangeable.
- Examples of such problems include:
  - Pigeon hole problem
  - Graph colouring problem
  - Scheduling problem
- Can be detected (automatically) or declared (manually)

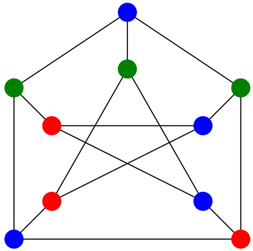
WHAT IS SYMMETRY?

Context

Combinatorial search and optimization problems

Assign values to variables so that constraints are satisfied and objective is minimized

- Symmetry: Permutation of variables that preserves the solutions.
- Occurs e.g. when sets of variables in the problem are interchangeable.
- Examples of such problems include:
  - Pigeon hole problem
  - Graph colouring problem
  - Scheduling problem
- Can be detected (automatically) or declared (manually)



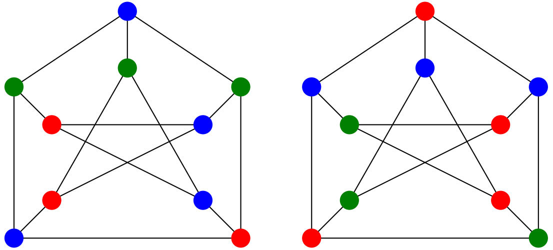
WHAT IS SYMMETRY?

Context

Combinatorial search and optimization problems

Assign values to variables so that constraints are satisfied and objective is minimized

- Symmetry: Permutation of variables that preserves the solutions.
- Occurs e.g. when sets of variables in the problem are interchangeable.
- Examples of such problems include:
  - Pigeon hole problem
  - Graph colouring problem
  - Scheduling problem
- Can be detected (automatically) or declared (manually)



WHAT IS SYMMETRY HANDLING?

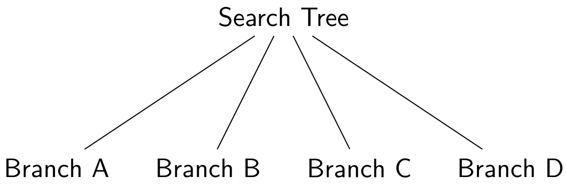
- Many symmetric branches: problem!
  - Exploding search tree.

WHAT IS SYMMETRY HANDLING?

- Many symmetric branches: problem!
  - Exploding search tree.
- Goal of symmetry handling: prune the search tree.

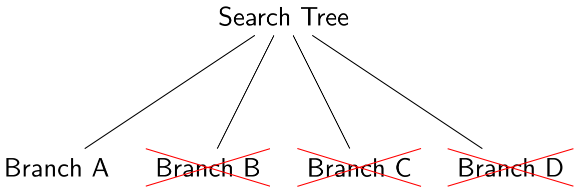
WHAT IS SYMMETRY HANDLING?

- Many symmetric branches: problem!
  - Exploding search tree.
- Goal of symmetry handling: prune the search tree.



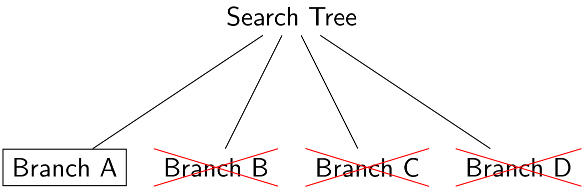
WHAT IS SYMMETRY HANDLING?

- Many symmetric branches: problem!
  - Exploding search tree.
- Goal of symmetry handling: prune the search tree.



WHAT IS SYMMETRY HANDLING?

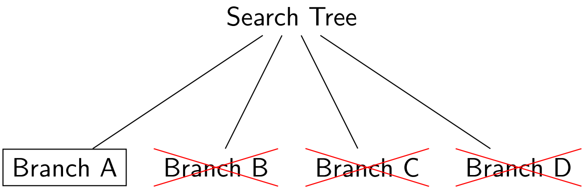
- Many symmetric branches: problem!
  - Exploding search tree.
- Goal of symmetry handling: prune the search tree.



WHAT IS SYMMETRY HANDLING?

- Many symmetric branches: problem!
  - Exploding search tree.
- Goal of symmetry handling: prune the search tree.
- Static symmetry breaking
  - Static: during a preprocessing phase.
  - Breaking: possibly prunes solutions.
  - How? Adding *lex-leader constraints* LL:

    LL discards assignments that are lexicographically larger than their symmetrical counterpart.



SYMMETRY AND PSEUDO-BOOLEAN OPTIMIZATION

Pseudo-Boolean Optimization Problem

Consists of:

- Constraints $C = \sum_{i=1}^{j} w_i l_i \geq A, A \in \mathbb{N}$
- Formula $F = \bigwedge_{i=1}^{k} C_i,$
- Objective $O = \sum_{i=1}^{m} w_i l_i$

SYMMETRY AND PSEUDO-BOOLEAN OPTIMIZATION

Pseudo-Boolean Optimization Problem

Consists of:

- Constraints $C = \sum_{i=1}^{j} w_i l_i \geq A, A \in \mathbb{N}$
- Formula $F = \bigwedge_{i=1}^{k} C_i$,
- Objective $O = \sum_{i=1}^{m} w_i l_i$

Symmetries of a pseudo-Boolean optimization problem:

- A **strong** symmetry σ is a symmetry of both F and O,
  - i.e. $\sigma(O) = O$ and $\sigma(F) = F$.

SYMMETRY AND PSEUDO-BOOLEAN OPTIMIZATION

Pseudo-Boolean Optimization Problem

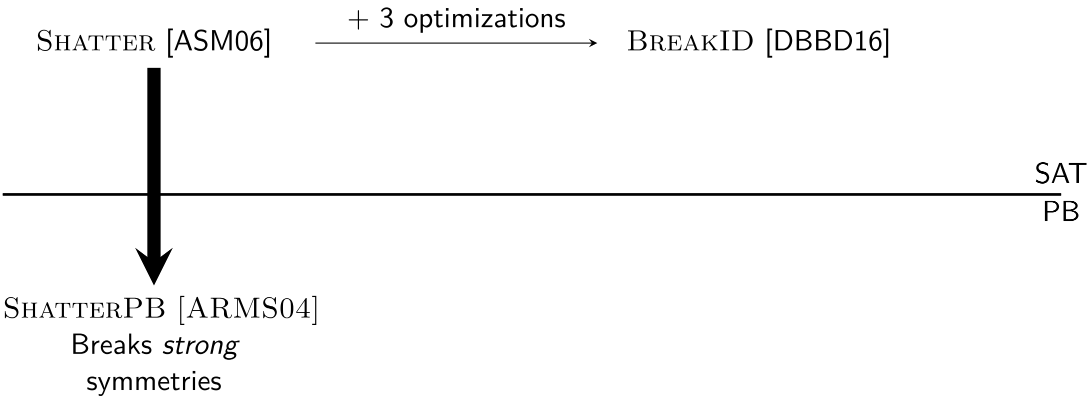
Consists of:

- Constraints $C = \sum_{i=1}^{j} w_i l_i \geq A, A \in \mathbb{N}$
- Formula $F = \bigwedge_{i=1}^{k} C_i$,
- Objective $O = \sum_{i=1}^{m} w_i l_i$

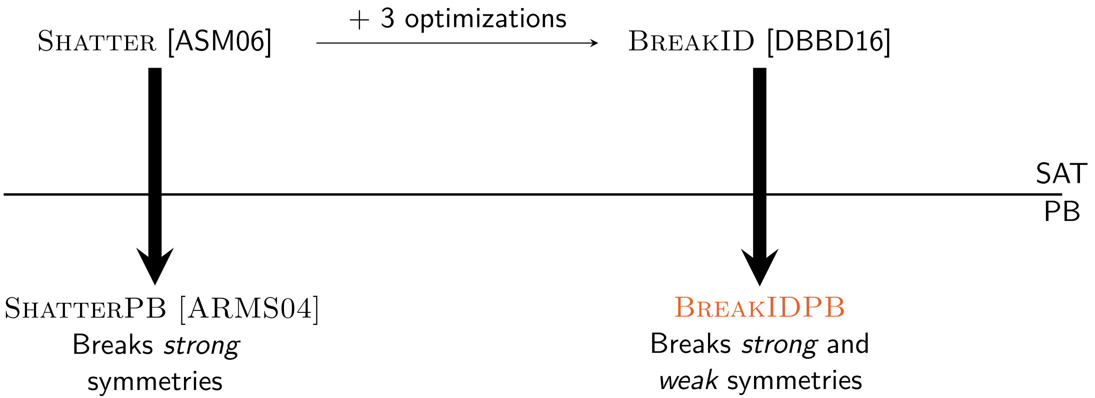
Symmetries of a pseudo-Boolean optimization problem:

- A **strong** symmetry σ is a symmetry of both F and O,
  - i.e. $\sigma(O) = O$ and $\sigma(F) = F$.
- (new) A **weak** symmetry ω is a symmetry of F,
  - i.e. $\omega(F) = F$.
- All strong symmetries are weak symmetries.

EXTENDING BREAKID



EXTENDING BREAKID



EXTENDING BREAKID

Symmetry Breaking

Symmetry Detection

Adding symmetry breaking constraints

SYMMETRY DETECTION FOR PSEUDO-BOOLEAN OPTIMIZATION

$$F = C_1 \land C_2 \land C_3 \land C_4$$

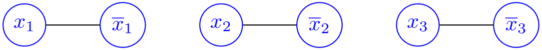
$$O = 2x_2 + 3\overline{x}_3$$

$$C_1 = x_1 + x_2 + \overline{x}_3 \geq 1$$

$$C_2 = x_3 + \overline{x}_1 + \overline{x}_2 \geq 1$$

$$C_3 = 2x_1 + x_2 + x_3 \geq 2$$

$$C_4 = x_1 + 2x_2 + x_3 \geq 2$$



SYMMETRY DETECTION FOR PSEUDO-BOOLEAN OPTIMIZATION

$F = C_1 \wedge C_2 \wedge C_3 \wedge C_4$

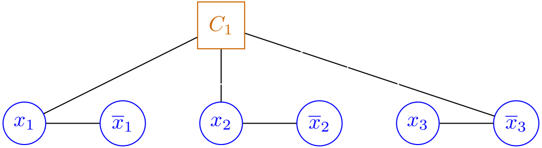
$O = 2x_2 + 3\overline{x}_3$

$C_1 = x_1 + x_2 + \overline{x}_3 \geq 1$

$C_2 = x_3 + \overline{x}_1 + \overline{x}_2 \geq 1$

$C_3 = 2x_1 + x_2 + x_3 \geq 2$

$C_4 = x_1 + 2x_2 + x_3 \geq 2$



SYMMETRY DETECTION FOR PSEUDO-BOOLEAN OPTIMIZATION

$F = C_1 \wedge C_2 \wedge C_3 \wedge C_4$

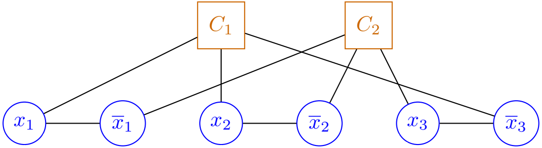
$O = 2x_2 + 3\overline{x}_3$

$C_1 = x_1 + x_2 + \overline{x}_3 \geq 1$

$C_2 = x_3 + \overline{x}_1 + \overline{x}_2 \geq 1$

$C_3 = 2x_1 + x_2 + x_3 \geq 2$

$C_4 = x_1 + 2x_2 + x_3 \geq 2$



SYMMETRY DETECTION FOR PSEUDO-BOOLEAN OPTIMIZATION

$F = C_1 \wedge C_2 \wedge C_3 \wedge C_4$

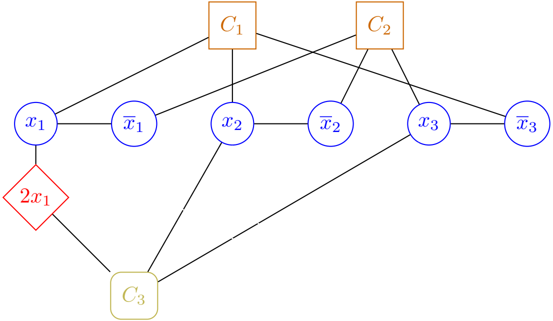
$O = 2x_2 + 3\overline{x}_3$

$C_1 = x_1 + x_2 + \overline{x}_3 \geq 1$

$C_2 = x_3 + \overline{x}_1 + \overline{x}_2 \geq 1$

$C_3 = 2x_1 + x_2 + x_3 \geq 2$

$C_4 = x_1 + 2x_2 + x_3 \geq 2$



SYMMETRY DETECTION FOR PSEUDO-BOOLEAN OPTIMIZATION

$F = C_1 \wedge C_2 \wedge C_3 \wedge C_4$

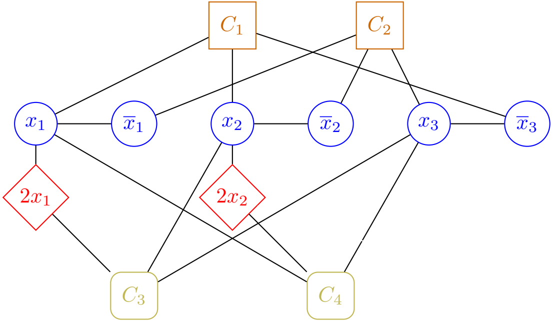
$O = 2x_2 + 3\overline{x}_3$

$C_1 = x_1 + x_2 + \overline{x}_3 \geq 1$

$C_2 = x_3 + \overline{x}_1 + \overline{x}_2 \geq 1$

$C_3 = 2x_1 + x_2 + x_3 \geq 2$

$C_4 = x_1 + 2x_2 + x_3 \geq 2$



SYMMETRY DETECTION FOR PSEUDO-BOOLEAN OPTIMIZATION

$F = C_1 \wedge C_2 \wedge C_3 \wedge C_4$

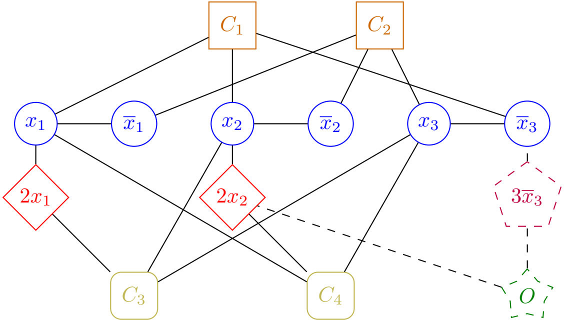
$O = 2x_2 + 3\overline{x}_3$

$C_1 = x_1 + x_2 + \overline{x}_3 \geq 1$

$C_2 = x_3 + \overline{x}_1 + \overline{x}_2 \geq 1$

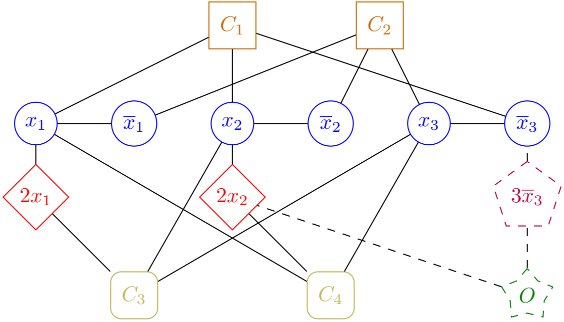
$C_3 = 2x_1 + x_2 + x_3 \geq 2$

$C_4 = x_1 + 2x_2 + x_3 \geq 2$



SYMMETRY DETECTION FOR PSEUDO-BOOLEAN OPTIMIZATION

$$F = C_1 \wedge C_2 \wedge C_3 \wedge C_4$$
$$O = 2x_2 + 3\overline{x}_3$$
$$C_1 = x_1 + x_2 + \overline{x}_3 \geq 1$$
$$C_2 = x_3 + \overline{x}_1 + \overline{x}_2 \geq 1$$
$$C_3 = 2x_1 + x_2 + x_3 \geq 2$$
$$C_4 = x_1 + 2x_2 + x_3 \geq 2$$



- Strong symmetries are automorphisms of the full graph

SYMMETRY DETECTION FOR PSEUDO-BOOLEAN OPTIMIZATION

$F = C_1 \wedge C_2 \wedge C_3 \wedge C_4$

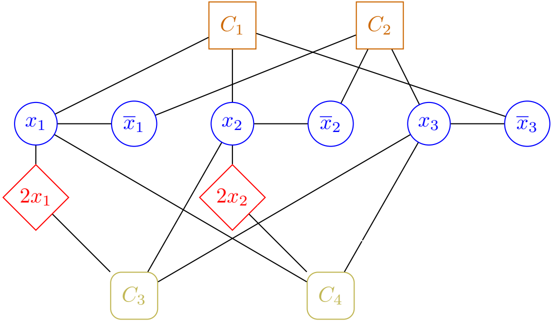
$O = 2x_2 + 3\overline{x}_3$

$C_1 = x_1 + x_2 + \overline{x}_3 \geq 1$

$C_2 = x_3 + \overline{x}_1 + \overline{x}_2 \geq 1$

$C_3 = 2x_1 + x_2 + x_3 \geq 2$

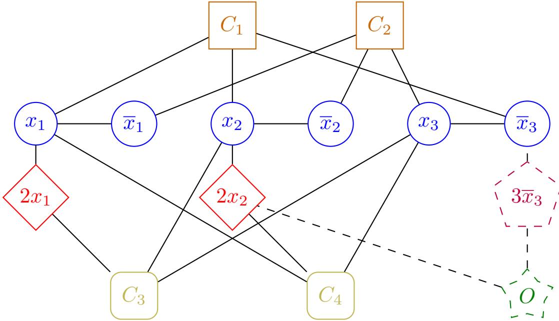
$C_4 = x_1 + 2x_2 + x_3 \geq 2$



- Strong symmetries are automorphisms of the full graph
- Weak symmetries are automorphisms of the solid graph

SYMMETRY DETECTION FOR PSEUDO-BOOLEAN OPTIMIZATION

$$F = C_1 \wedge C_2 \wedge C_3 \wedge C_4$$
$$O = 2x_2 + 3\overline{x}_3$$
$$C_1 = x_1 + x_2 + \overline{x}_3 \geq 1$$
$$C_2 = x_3 + \overline{x}_1 + \overline{x}_2 \geq 1$$
$$C_3 = 2x_1 + x_2 + x_3 \geq 2$$
$$C_4 = x_1 + 2x_2 + x_3 \geq 2$$



- **Strong** symmetries are automorphisms of the **full** graph
- **Weak** symmetries are automorphisms of the **solid** graph
- **Detect** graph automorphisms with SAUCY [KSM10].

BREAKING CONSTRAINTS FOR PSEUDO-BOOLEAN OPTIMIZATION

Lex-leader breaking constraints LL (e.g. in SAT)

Discard assignments that are lexicographically larger than their symmetric counterpart.

BREAKING CONSTRAINTS FOR PSEUDO-BOOLEAN OPTIMIZATION

Lex-leader breaking constraints LL (e.g. in SAT)

Discard assignments that are lexicographically larger than their symmetric counterpart.

Breaking Strong Symmetries σ of (F, O)= [ARMS04]

Guaranteed that $\sigma(O) = O$ and $\sigma(F) = F$.

$B = LL_\sigma$

BREAKING CONSTRAINTS FOR PSEUDO-BOOLEAN OPTIMIZATION

Lex-leader breaking constraints LL (e.g. in SAT)

Discard assignments that are lexicographically larger than their symmetric counterpart.

Breaking Strong Symmetries σ of (F, O)= [ARMS04]

Guaranteed that $\sigma(O) = O$ and $\sigma(F) = F$.

$B = LL_\sigma$

Breaking Weak Symmetries ω of (F, O)

Guaranteed that $\omega(F) = F$, still need to take O into account.

$$D = \left\{ \begin{array}{c} O \leq \omega(O) \\ (O = \omega(O)) \Rightarrow LL_\omega \end{array} \right\}$$

EXPERIMENTS: CRAFTED BENCHMARK SET [Nor22]

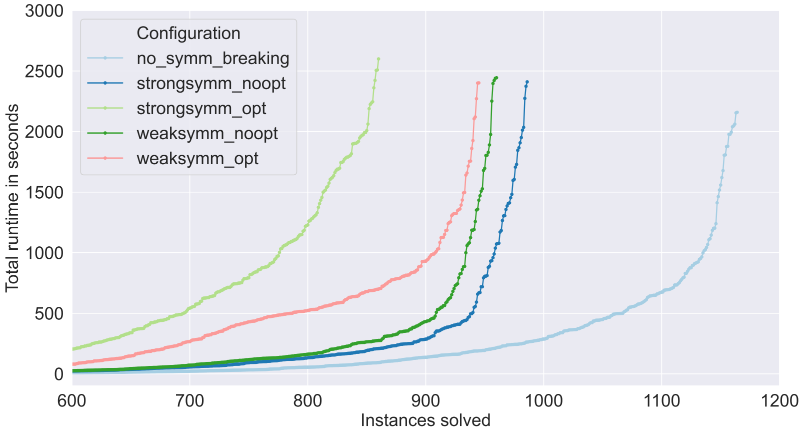


Figure: Results for different BreakIDPB configurations with the core-guided optimization configuration of RoundingSAT [EN18].

EXPERIMENTS: CRAFTED BENCHMARK SET [Nor22]

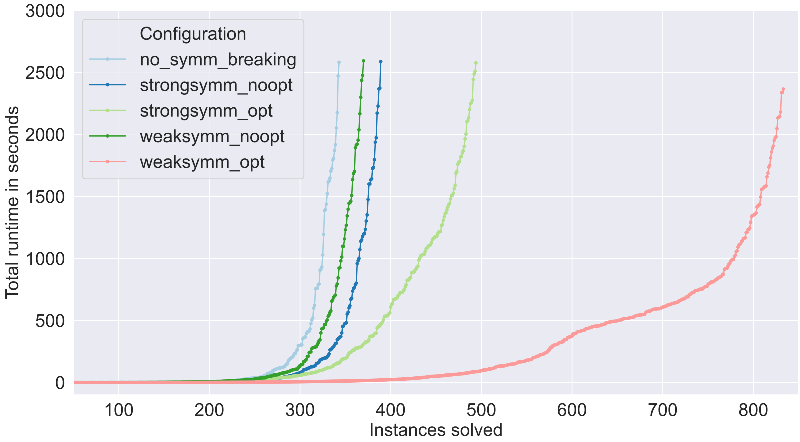


Figure: Results for different BREAKIDPB configurations with the linear SAT–UNSAT configuration of ROUNDINGSAT [EN18].

CONCLUSIONS

- Defined the new notion of **weak** symmetries and how to detect and break them.

CONCLUSIONS

- Defined the new notion of **weak** symmetries and how to detect and break them.
- Effect of breaking (weak) symmetries depends greatly on the type of solving algorithm used.

CONCLUSIONS

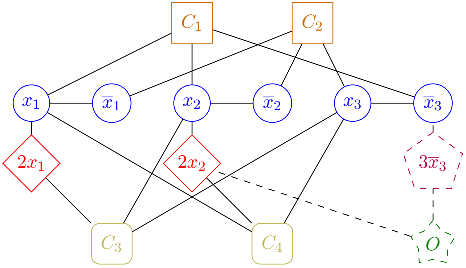
- Defined the new notion of **weak** symmetries and how to detect and break them.
- Effect of breaking (weak) symmetries depends greatly on the type of solving algorithm used.
- Suprisingly, adding weak symmetry breaking can have a positive effect on instances that do not exhibit weak symmetries.

CONCLUSIONS

- Defined the new notion of **weak** symmetries and how to detect and break them.
- Effect of breaking (weak) symmetries depends greatly on the type of solving algorithm used.
- Suprisingly, adding weak symmetry breaking can have a positive effect on instances that do not exhibit weak symmetries.
  - Simpeler and smaller graph makes structure in symmetry group more visible.

CONCLUSIONS

- Defined the new notion of **weak** symmetries and how to detect and break them.
- Effect of breaking (weak) symmetries depends greatly on the type of solving algorithm used.
- Suprisingly, adding weak symmetry breaking can have a positive effect on instances that do not exhibit weak symmetries.
  - Simpeler and smaller graph makes structure in symmetry group more visible.



Breaking Weak Symmetries ω of (F, O)

Guaranteed that $\omega(F) = F$, still need to take O into account.

$$D = \left\{ \begin{array}{c} O \leq \omega(O) \\ (O = \omega(O)) \Rightarrow LL_\omega \end{array} \right\}$$

REFERENCES

[ARMS04] F.A. Aloul, A. Ramani, I.L. Markov, and K.A. Sakallah. Shatterpb: symmetry-breaking for pseudo-boolean formulas. In *ASP-DAC 2004: Asia and South Pacific Design Automation Conference 2004 (IEEE Cat. No.04EX753)*, pages 884–887, 2004.

[ASM06] Fadi A. Aloul, Karim A. Sakallah, and Igor L. Markov. Efficient symmetry breaking for Boolean satisfiability. *IEEE Transactions on Computers*, 55(5):549–558, 2006.

[DBBD16] Jo Devriendt, Bart Bogaerts, Maurice Bruynooghe, and Marc Denecker. Improved static symmetry breaking for SAT. In Nadia Creignou and Daniel Le Berre, editors, *Theory and Applications of Satisfiability Testing - SAT 2016 - 19th International Conference, Bordeaux, France, July 5-8, 2016, Proceedings*, volume 9710 of *Lecture Notes in Computer Science*, pages 104–122. Springer, 2016.

[EN18] Jan Elffers and Jakob Nordström. Divide and conquer: Towards faster pseudo-Boolean solving. In Jérôme Lang, editor, *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 1291–1299. ijcai.org, 2018.

[KSM10] Hadi Katebi, Karem A. Sakallah, and Igor L. Markov. Symmetry and satisfiability: An update. In Ofer Strichman and Stefan Szeider, editors, *SAT*, volume 6175 of *LNCS*, pages 113–127. Springer, 2010.

[Nor22] Jakob Nordström. Supplementary material for cutting to the core of pseudo-boolean optimization: Combining core-guided search with cutting planes reasoning, 2022.

COMPATIBILITY WITH OPTIMIZATIONS

- Compact encoding for lex-leader constraints (i.e. the used symmetry breaking constraints).
- Exploitation of row-interchangeability symmetries.
- Generation of binary symmetry breaking clauses.
  - Only added for strong symmetries.

BREAKING CONSTRAINTS: WEAK SYMMETRY

$$\forall \alpha \models D \Rightarrow \left\{ \begin{array}{c} O(\alpha) \leq \omega(O(\alpha)) \\ (O(\alpha) = \omega(O(\alpha))) \Rightarrow \alpha \leq_{lex} \omega(\alpha) \end{array} \right\}$$

BREAKING CONSTRAINTS: WEAK SYMMETRY

$$\forall \alpha \models D \Rightarrow \left\{ \begin{array}{c} O(\alpha) \leq \omega(O(\alpha)) \\ (O(\alpha) = \omega(O(\alpha))) \Rightarrow \alpha \leq_{lex} \omega(\alpha) \end{array} \right\}$$

$$D = \left\{ \begin{array}{c} O \leq \omega(O) \\ (O = \omega(O)) \Rightarrow LL_\omega \end{array} \right\} \Longleftrightarrow \left\{ \begin{array}{c} O \leq \omega(O) \\ y_0 \Longleftrightarrow (O = \omega(O)) \\ y_0 \Rightarrow LL_\omega \end{array} \right\} \Longleftrightarrow$$

$$\left\{ \begin{array}{c} \omega(O) - O \geq 0 \\ cy_0 + \omega(O) - O \geq 1 \\ -cy_0 + O - \omega(O) \geq -c \\ y_0 \Rightarrow LL_\omega \end{array} \right\}$$