Static Symmetry and Dominance Breaking for Pseudo-Boolean Optimization

Daimy Van Caudenberg and Bart Bogaerts
Vrije Universiteit Brussel

Benelux Conference on Artificial Intelligence '22
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)
Introduction

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.

▶ Exploding 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.
What is Symmetry Handling?

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

Search Tree

- Branch A
- Branch B
- Branch C
- Branch D
WHAT IS SYMMETRY HANDLING?

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

Search Tree

Branch A  Branch B  Branch C  Branch D
WHAT IS SYMMETRY HANDLING?

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

- Goal of symmetry handling: prune the search tree.

---

Search Tree

Branch A
Branch B
Branch C
Branch D
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.
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 breaking for pseudo-Boolean Optimization

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  \( \sigma \) is a symmetry of both  \( F \) and  \( O \),
  - i.e.  \( \sigma(O) = O \) and  \( \sigma(F) = F \).
Symmetry breaking for pseudo-Boolean Optimization

**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** $\sigma$ is a symmetry of both $F$ and $O$,
  - i.e. $\sigma(O) = O$ and $\sigma(F) = F$.
- **(new) A weak symmetry** $\omega$ is a symmetry of $F$,
  - i.e. $\omega(F) = F$.
- All strong symmetries are weak symmetries.
Symmetry breaking for pseudo-Boolean Optimization

EXTENDING BreakID

\textbf{Shatter} [ASM06] \quad + \quad 3 \text{ optimizations} \quad \rightarrow \quad \textbf{BreakID} [DBBD16]

\textbf{ShatterPB} [ARMS04]

Breaks \textit{strong} symmetries

SAT
PB
EXTENDING BreakID

**SHATTER** [ASM06] + 3 optimizations → **BreakID** [DBBD16]

**SHATTERPB** [ARMS04] Breaks *strong* symmetries

**BreakIDPB** Breaks *strong* and *weak* symmetries
Symmetry Detection

Symmetry Breaking

Adding symmetry breaking constraints
\[ F = C_1 \land C_2 \land C_3 \land C_4 \]
\[ O = 2x_2 + 3x_3 \]
\[ C_1 = x_1 + x_2 + x_3 \geq 1 \]
\[ C_2 = x_3 + x_1 + x_2 \geq 1 \]
\[ C_3 = 2x_1 + x_2 + x_3 \geq 2 \]
\[ C_4 = x_1 + 2x_2 + x_3 \geq 2 \]
Symmetry breaking for pseudo-Boolean Optimization

**SYMMETRY DETECTION FOR PSEUDO-BOOLEAN OPTIMIZATION**

\[ F = C_1 \land C_2 \land C_3 \land C_4 \]

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

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

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

\[ C_3 = 2x_1 + x_2 + x_3 \geq 2 \]

\[ C_4 = x_1 + 2x_2 + x_3 \geq 2 \]

---

Strong symmetries are automorphisms of the full graph.

Weak symmetries are automorphisms of the solid graph.

Detect graph automorphisms with Saucy \([\text{KSM10}]\).

---

Daimy Van Caudenberg (VUB)
\[ F = C_1 \land C_2 \land C_3 \land C_4 \]
\[ O = 2x_2 + 3\bar{x}_3 \]
\[ C_1 = x_1 + x_2 + \bar{x}_3 \geq 1 \]
\[ C_2 = x_3 + \bar{x}_1 + \bar{x}_2 \geq 1 \]
\[ C_3 = 2x_1 + x_2 + x_3 \geq 2 \]
\[ C_4 = x_1 + 2x_2 + x_3 \geq 2 \]
Symmetry breaking for pseudo-Boolean Optimization

\[ F = C_1 \land C_2 \land C_3 \land C_4 \]
\[ O = 2x_2 + 3x_3 \]
\[ C_1 = x_1 + x_2 + \bar{x}_3 \geq 1 \]
\[ C_2 = x_3 + \bar{x}_1 + \bar{x}_2 \geq 1 \]
\[ C_3 = 2x_1 + x_2 + x_3 \geq 2 \]
\[ C_4 = x_1 + 2x_2 + x_3 \geq 2 \]
\[ F = C_1 \land C_2 \land C_3 \land C_4 \]
\[ O = 2x_2 + 3\overline{x}_3 \]
\[ C_1 = x_1 + x_2 + \overline{x}_3 \geq 1 \]
\[ C_2 = x_3 + \overline{x}_1 + \overline{x}_2 \geq 1 \]
\[ C_3 = 2x_1 + x_2 + x_3 \geq 2 \]
\[ C_4 = x_1 + 2x_2 + x_3 \geq 2 \]
Symmetry breaking for pseudo-Boolean Optimization

**SYMMETRY DETECTION FOR PSEUDO-BOOLEAN OPTIMIZATION**

\[ F = C_1 \land C_2 \land C_3 \land C_4 \]

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

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

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

\[ C_3 = 2x_1 + x_2 + x_3 \geq 2 \]

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

**Strong** symmetries are automorphisms of the full graph.
\[ F = C_1 \land C_2 \land C_3 \land C_4 \]
\[ O = 2x_2 + 3\overline{x}_3 \]
\[ C_1 = x_1 + x_2 + \overline{x}_3 \geq 1 \]
\[ C_2 = x_3 + \overline{x}_1 + \overline{x}_2 \geq 1 \]
\[ C_3 = 2x_1 + x_2 + x_3 \geq 2 \]
\[ C_4 = x_1 + 2x_2 + x_3 \geq 2 \]

- Strong symmetries are automorphisms of the full graph
- Weak symmetries are automorphisms of the solid graph
\[ F = C_1 \land C_2 \land C_3 \land C_4 \]
\[ O = 2x_2 + 3\overline{x}_3 \]
\[ C_1 = x_1 + x_2 + \overline{x}_3 \geq 1 \]
\[ C_2 = x_3 + \overline{x}_1 + \overline{x}_2 \geq 1 \]
\[ C_3 = 2x_1 + x_2 + x_3 \geq 2 \]
\[ C_4 = x_1 + 2x_2 + x_3 \geq 2 \]

- **Strong** symmetries are automorphisms of the full graph
- **Weak** symmetries are automorphisms of the solid graph
- **Detect** graph automorphisms with \texttt{Saucy} [KSM10].
Lex-leader breaking constraints $LL$ (e.g. in SAT)

Discard assignments that are lexicographically larger than their symmetric counterpart.
Lex-leader breaking constraints $LL$ (e.g. in SAT)

Discard assignments that are lexicographically larger than their symmetric counterpart.

Breaking Strong Symmetries $\sigma$ of $(F, O) = [ARMS04]$

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

$B = LL_\sigma$
Lex-leader breaking constraints $LL$ (e.g. in SAT)

Discard assignments that are lexicographically larger than their symmetric counterpart.

Breaking Strong Symmetries $\sigma$ of $(F, O) = [ARMS04]$ 

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

$B = LL_\sigma$

Breaking Weak Symmetries $\omega$ of $(F, O)$

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

$D = \left\{ \begin{array}{l} O \leq \omega(O) \\ (O = \omega(O)) \Rightarrow LL_\omega \end{array} \right\}$
Figure: Results for different BreakIDPB configurations with the core-guided optimization configuration of RoundingSAT [EN18].
**Figure:** Results for different **BREAKIDPB** configurations with the **linear SAT–UNSAT** configuration of **ROUNDINGSAT** [EN18].
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.

Surprisingly, adding weak symmetry breaking can have a positive effect on instances that do not exhibit weak symmetries.

Simpler and smaller graph makes structure in symmetry group more visible.

Breaking Weak Symmetries \( \omega \) of \((F, O)\)

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

\[ D = \{ O \leq \omega(O) \} \implies LL \omega \} \]
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.
- Surprisingly, 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.
  - Simpler 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.
- Surprisingly, adding weak symmetry breaking can have a positive effect on instances that do not exhibit weak symmetries.
  - Simpler and smaller graph makes structure in symmetry group more visible.

Breaking Weak Symmetries $\omega$ of $(F, O)$

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

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


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.
\forall \alpha \models D \Rightarrow \left\{ \begin{array}{l} O(\alpha) \leq \omega(O(\alpha)) \\ (O(\alpha) = \omega(O(\alpha))) \Rightarrow \alpha \leq_{\text{lex}} \omega(\alpha) \end{array} \right\}
\[ \forall \alpha \mid D \Rightarrow \left\{ \begin{array}{l} O(\alpha) \leq \omega(O(\alpha)) \\ (O(\alpha) = \omega(O(\alpha))) \Rightarrow \alpha \leq_{lex} \omega(\alpha) \end{array} \right\} \]

\[ D = \left\{ \begin{array}{l} O \leq \omega(O) \\ (O = \omega(O)) \Rightarrow LL_{\omega} \end{array} \right\} \iff \left\{ \begin{array}{l} O \leq \omega(O) \\ y_0 \iff (O = \omega(O)) \\ y_0 \Rightarrow LL_{\omega} \end{array} \right\} \iff \left\{ \begin{array}{l} \omega(O) - O \geq 0 \\ cy_0 + \omega(O) - O \geq 1 \\ -cy_0 + O - \omega(O) \geq -c \\ y_0 \Rightarrow LL_{\omega} \end{array} \right\} \]