Proof Logging for MaxSAT Preprocessing

Based on joint work with: Andy Oertel, Yong Kiam Tan, Jeremias Berg, Matti Järvisalo, Magnus O. Myreen, and Jakob Nordström

HELSINKI INSTITUTE FOR INFORMATION TECHNOLOGY

WHOOPS May 24th 2024

Outline

- MaxSAT and Preprocessing
- Proof logging for MaxSAT preprocessing: Overview
- Proof logging for MaxSAT preprocessing: Practical examples

MaxSAT and Preprocessing

(Objective-Centric form of) MaxSAT

- Optimization variant of SAT, (F, O)
- $F = \{(x_1 \vee \bar{x}_2), (x_2 \vee \bar{x}_3), (x_3 \vee \bar{x}_1), (\bar{x}_2 \vee \bar{x}_3 \vee \bar{x}_4)\}$
- $O \equiv \bar{x}_1 + 2x_4$
- There are three solutions to (F, O):
  - $\tau_1 = \{x_1 \to 1, x_2 \to 1, x_3 \to 1, x_4 \to 0\}$
  - $\tau_2 = \{x_1 \to 0, x_2 \to 0, x_3 \to 0, x_4 \to 0\}$
  - $\tau_3 = \{x_1 \to 0, x_2 \to 0, x_3 \to 0, x_4 \to 1\}$
  - $O(\tau_1) = 2$
  - $O(\tau_2) = 1$
  - $O(\tau_3) = 3$
  - τ_2 is an optimal solution.

(Objective-Centric form of) MaxSAT

- Optimization variant of SAT, (F, O)
- $F = \{(x_1 \vee \bar{x}_2), (x_2 \vee \bar{x}_3), (x_3 \vee \bar{x}_1), (\bar{x}_2 \vee \bar{x}_3 \vee \bar{x}_4)\}$
- $O \equiv \bar{x}_1 + 2x_4$
- There are three solutions to (F, O):
  - $\tau_1 = \{x_1 \to 1, x_2 \to 1, x_3 \to 1, x_4 \to 0\}$
  - $\tau_2 = \{x_1 \to 0, x_2 \to 0, x_3 \to 0, x_4 \to 0\}$
  - $\tau_3 = \{x_1 \to 0, x_2 \to 0, x_3 \to 0, x_4 \to 1\}$
  - $O(\tau_1) = 2$
  - $O(\tau_2) = 1$
  - $O(\tau_3) = 3$
  - τ_2 is an optimal solution.

(Objective-Centric form of) MaxSAT

- Optimization variant of SAT, (F, O)
- $F = \{(x_1 \vee \bar{x}_2), (x_2 \vee \bar{x}_3), (x_3 \vee \bar{x}_1), (\bar{x}_2 \vee \bar{x}_3 \vee \bar{x}_4)\}$
- $O \equiv \bar{x}_1 + 2x_4$
- There are three solutions to (F, O):
  - $\tau_1 = \{x_1 \to 1, x_2 \to 1, x_3 \to 1, x_4 \to 0\}$
  - $\tau_2 = \{x_1 \to 0, x_2 \to 0, x_3 \to 0, x_4 \to 0\}$
  - $\tau_3 = \{x_1 \to 0, x_2 \to 0, x_3 \to 0, x_4 \to 1\}$
  - $O(\tau_1) = 2$
  - $O(\tau_2) = 1$
  - $O(\tau_3) = 3$
  - τ_2 is an optimal solution.

(Objective-Centric form of) MaxSAT

- Optimization variant of SAT, (F, O)
- $F = \{(x_1 \lor \bar{x}_2), (x_2 \lor \bar{x}_3), (x_3 \lor \bar{x}_1), (\bar{x}_2 \lor \bar{x}_3 \lor \bar{x}_4)\}$
- $O \equiv \bar{x}_1 + 2x_4$
- There are three solutions to (F, O):
  - $\tau_1 = \{x_1 \to 1, x_2 \to 1, x_3 \to 1, x_4 \to 0\}$
  - $\tau_2 = \{x_1 \to 0, x_2 \to 0, x_3 \to 0, x_4 \to 0\}$
  - $\tau_3 = \{x_1 \to 0, x_2 \to 0, x_3 \to 0, x_4 \to 1\}$
  - $O(\tau_1) = 2$
  - $O(\tau_2) = 1$
  - $O(\tau_3) = 3$
  - τ_2 is an optimal solution.

WCNF form of MaxSAT

- "Satisfy all hard clauses, minimize the total weight of unsatisfied soft clauses"
- Example:
  - $\mathcal{F} = (F_H, F_S)$
  - $F_H = \{(x_1 \vee \bar{x}_2), (x_2 \vee \bar{x}_3), (x_3 \vee \bar{x}_1)\}$
  - $F_S = \{\langle (x_1), 1 \rangle, \langle (\bar{x}_2 \vee \bar{x}_3), 2 \rangle\}$
- Conversion to objective-centric
  - $\mathcal{F}^b = (F_H^b, F_S^b)$
  - $F_H^b = \{(x_1 \vee \bar{x}_2), (x_2 \vee \bar{x}_3), (x_3 \vee \bar{x}_1), (\bar{x}_2 \vee \bar{x}_3 \vee x_4)\}$
  - $F_S^b = \{\langle (x_1), 1 \rangle, \langle (\bar{x}_4), 2 \rangle\}$
- $\rightsquigarrow$
  - $F = \{(x_1 \vee \bar{x}_2), (x_2 \vee \bar{x}_3), (x_3 \vee \bar{x}_1), (\bar{x}_2 \vee \bar{x}_3 \vee \bar{x}_4)\}$
  - $O \equiv \bar{x}_1 + 2x_4$

WCNF form of MaxSAT

- "Satisfy all hard clauses, minimize the total weight of unsatisfied soft clauses"
- Example:
  - $\mathcal{F} = (F_H, F_S)$
  - $F_H = \{(x_1 \lor \bar{x}_2), (x_2 \lor \bar{x}_3), (x_3 \lor \bar{x}_1)\}$
  - $F_S = \{\langle (x_1), 1\rangle, \langle (\bar{x}_2 \lor \bar{x}_3), 2\rangle\}$
- Conversion to objective-centric
  - $\mathcal{F}^b = (F_H^b, F_S^b)$
  - $F_H^b = \{(x_1 \lor \bar{x}_2), (x_2 \lor \bar{x}_3), (x_3 \lor \bar{x}_1), (\bar{x}_2 \lor \bar{x}_3 \lor x_4)\}$
  - $F_S^b = \{\langle (x_1), 1\rangle, \langle (\bar{x}_4), 2\rangle\}$
- $\rightsquigarrow$
  - $F = \{(x_1 \lor \bar{x}_2), (x_2 \lor \bar{x}_3), (x_3 \lor \bar{x}_1), (\bar{x}_2 \lor \bar{x}_3 \lor \bar{x}_4)\}$
  - $O \equiv \bar{x}_1 + 2x_4$

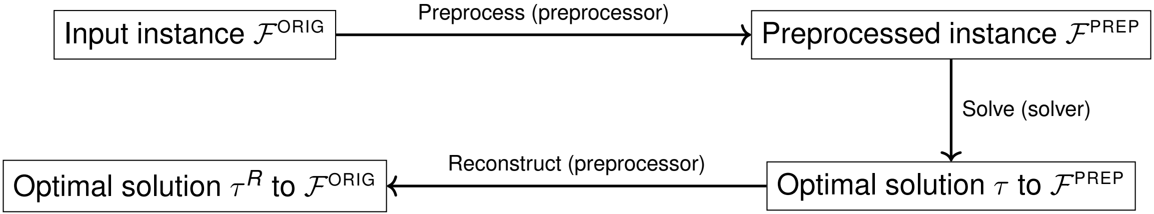
WCNF form of MaxSAT

- "Satisfy all hard clauses, minimize the total weight of unsatisfied soft clauses"
- Example:
  - $\mathcal{F} = (F_H, F_S)$
  - $F_H = \{(x_1 \lor \bar{x}_2), (x_2 \lor \bar{x}_3), (x_3 \lor \bar{x}_1)\}$
  - $F_S = \{\langle (x_1), 1 \rangle, \langle (\bar{x}_2 \lor \bar{x}_3), 2 \rangle\}$
- Conversion to objective-centric
  - $\mathcal{F}^b = (F^b_H, F^b_S)$
  - $F^b_H = \{(x_1 \lor \bar{x}_2), (x_2 \lor \bar{x}_3), (x_3 \lor \bar{x}_1), (\bar{x}_2 \lor \bar{x}_3 \lor x_4)\}$
  - $F^b_S = \{\langle (x_1), 1 \rangle, \langle (\bar{x}_4), 2 \rangle\}$
- $\rightsquigarrow$
  - $F = \{(x_1 \lor \bar{x}_2), (x_2 \lor \bar{x}_3), (x_3 \lor \bar{x}_1), (\bar{x}_2 \lor \bar{x}_3 \lor \bar{x}_4)\}$
  - $O = \bar{x}_1 + 2x_4$

WCNF form of MaxSAT

- "Satisfy all hard clauses, minimize the total weight of unsatisfied soft clauses"
- Example:
  - $\mathcal{F} = (F_H, F_S)$
  - $F_H = \{(x_1 \lor \bar{x}_2), (x_2 \lor \bar{x}_3), (x_3 \lor \bar{x}_1)\}$
  - $F_S = \{\langle (x_1), 1 \rangle, \langle (\bar{x}_2 \lor \bar{x}_3), 2 \rangle\}$
- Conversion to objective-centric
  - $\mathcal{F}^b = (F_H^b, F_S^b)$
  - $F_H^b = \{(x_1 \lor \bar{x}_2), (x_2 \lor \bar{x}_3), (x_3 \lor \bar{x}_1), (\bar{x}_2 \lor \bar{x}_3 \lor x_4)\}$
  - $F_S^b = \{\langle (x_1), 1 \rangle, \langle (\bar{x}_4), 2 \rangle\}$
- $\rightsquigarrow$
  - $F = \{(x_1 \lor \bar{x}_2), (x_2 \lor \bar{x}_3), (x_3 \lor \bar{x}_1), (\bar{x}_2 \lor \bar{x}_3 \lor \bar{x}_4)\}$
  - $O \equiv \bar{x}_1 + 2x_4$

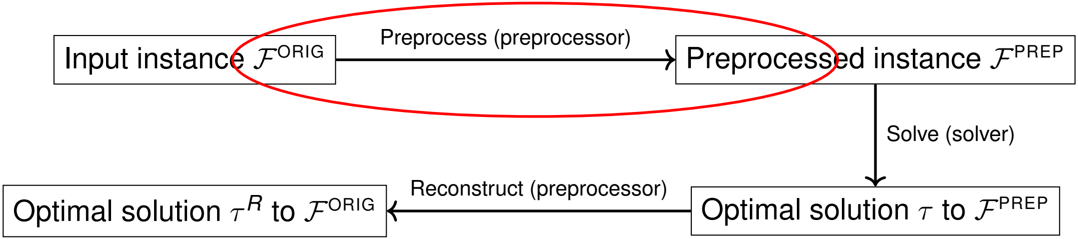
MaxSAT preprocessing



Proof logging for MaxSAT preprocessing: Overview

We want to verify equioptimality

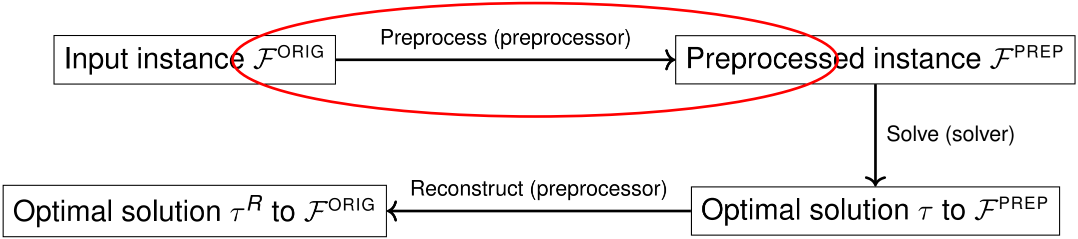
- Verify that $\mathcal{F}^{\text{ORIG}}$ and $\mathcal{F}^{\text{PREP}}$ have the same optimal cost



- What about reconstruction?
  - No proof logging for reconstruction
  - Verify that τ^R is a solution to $\mathcal{F}^{\text{ORIG}}$ and that $O^{\text{ORIG}}(\tau^R) = O^{\text{PREP}}(\tau)$.

We want to verify equioptimality

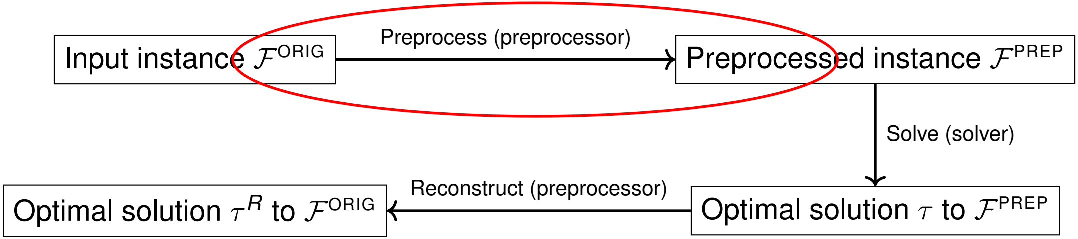
- Verify that $\mathcal{F}^{\text{ORIG}}$ and $\mathcal{F}^{\text{PREP}}$ have the same optimal cost



- What about reconstruction?
  - No proof logging for reconstruction
  - Verify that τ^R is a solution to $\mathcal{F}^{\text{ORIG}}$ and that $O^{\text{ORIG}}(\tau^R) = O^{\text{PREP}}(\tau)$.

We want to verify equioptimality

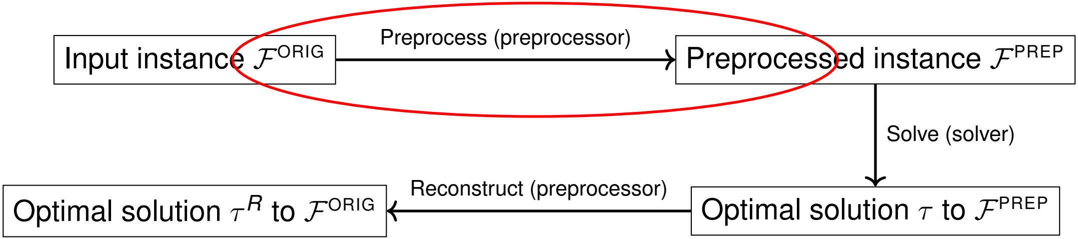
- Verify that $\mathcal{F}^{\text{ORIG}}$ and $\mathcal{F}^{\text{PREP}}$ have the same optimal cost



- What about reconstruction?
  - No proof logging for reconstruction
  - Verify that τ^R is a solution to $\mathcal{F}^{\text{ORIG}}$ and that $O^{\text{ORIG}}(\tau^R) = O^{\text{PREP}}(\tau)$.

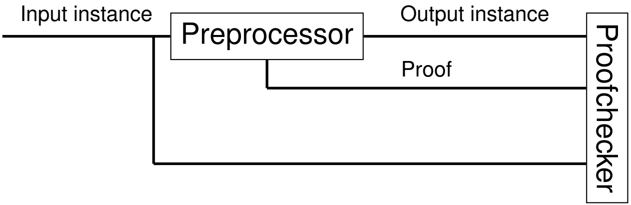
We want to verify equioptimality

- Verify that $\mathcal{F}^{\text{ORIG}}$ and $\mathcal{F}^{\text{PREP}}$ have the same optimal cost



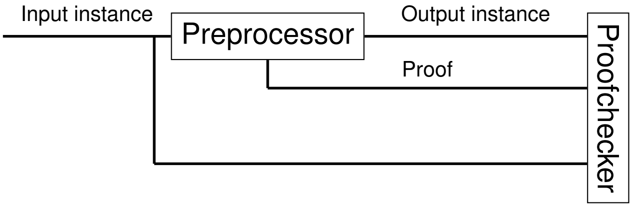
- What about reconstruction?
  - No proof logging for reconstruction
  - Verify that τ^R is a solution to $\mathcal{F}^{\text{ORIG}}$ and that $O^{\text{ORIG}}(\tau^R) = O^{\text{PREP}}(\tau)$.

Proof logging MaxPre



- MaxSAT to PBO in proofchecker-side
  - Convert to objective-centric
  - asPB$(\bigvee_i \ell_i) = \sum_i \ell_i \geq 1$.
  - Formally verified conversion
- VeriPB output section
  - Given output instance $F^o, O^o =$ asPB$(\mathcal{F}^{\text{PREP}})$, check that $\mathcal{C} = F^o$, $O = O^o$

Proof logging MaxPre



- MaxSAT to PBO in proofchecker-side
  - Convert to objective-centric
  - $\textsc{asPB}(\bigvee_i \ell_i) = \sum_i \ell_i \geq 1$.
  - Formally verified conversion
- VeriPB output section
  - Given output instance $F^o, O^o = \textsc{asPB}(\mathcal{F}^{\text{PREP}})$, check that $\mathcal{C} = F^o$, $O = O^o$

Some charasteristics of our proofs

- No objective-improving constraints
- Mainly adding and removing *core constraints*
- Heavy use of redundance-based strengthening and checked deletion
- Changes to the objective function

Some charasteristics of our proofs

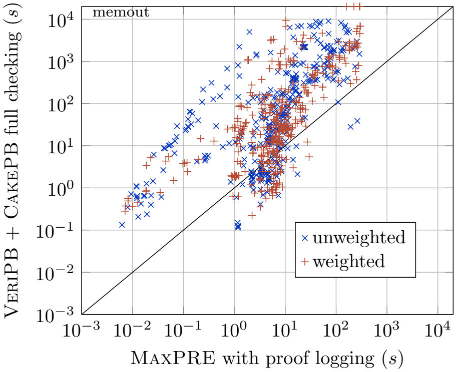
- No objective-improving constraints
- Mainly adding and removing *core constraints*
- Heavy use of redundance-based strengthening and checked deletion
- Changes to the objective function

Some charasteristics of our proofs

- No objective-improving constraints
- Mainly adding and removing *core constraints*
- Heavy use of redundance-based strengthening and checked deletion
- Changes to the objective function

Some charasteristics of our proofs

- No objective-improving constraints
- Mainly adding and removing *core constraints*
- Heavy use of redundance-based strengthening and checked deletion
- Changes to the objective function



Preprocessing flow of MaxPre

preprocessing (MaxSAT)

proof (pseudo-Boolean)

1. **Initialization**
2. **Preprocessing on WCNF**
3. **Conversion to objective-centric**
4. **Preprocessing on objective-centric**
5. **Constant removal**

Preprocessing flow of MaxPre

	preprocessing (MaxSAT)	*proof (pseudo-Boolean)*
1. Initialization	$\mathcal{F}^{\text{ORIG}}$	$\text{asPB}(\text{ObjMaxSAT}(\mathcal{F}^{\text{ORIG}}))$
2. Preprocessing on WCNF		
3. Conversion to objective-centric		
4. Preprocessing on objective-centric		
5. Constant removal		

Preprocessing flow of MaxPre

	preprocessing (MaxSAT)	*proof (pseudo-Boolean)*
1. Initialization	$\mathcal{F}^{\text{ORIG}}$	$\textsc{asPB}(\textsc{ObjMaxSAT}(\mathcal{F}^{\text{ORIG}}))$
2. Preprocessing on WCNF	$\mathcal{F}^1, \text{LB}^1$	$(\mathcal{C}^1, O^1)$
3. Conversion to objective-centric		
4. Preprocessing on objective-centric		
5. Constant removal		

Preprocessing flow of MaxPre

	preprocessing (MaxSAT)	*proof (pseudo-Boolean)*
1. Initialization	$\mathcal{F}^{\text{ORIG}}$	$\textsc{asPB}(\textsc{ObjMaxSAT}(\mathcal{F}^{\text{ORIG}}))$
2. Preprocessing on WCNF	$\mathcal{F}^1, \text{LB}^1$	$(\mathcal{C}^1, O^1)$
3. Conversion to objective-centric	(F^2, O^2)	$(\textsc{asPB}^*(F^2), O^{2*})$
4. Preprocessing on objective-centric		
5. Constant removal		

Preprocessing flow of MaxPre

	preprocessing (MaxSAT)	*proof (pseudo-Boolean)*
1. Initialization	$\mathcal{F}^{\text{ORIG}}$	$\text{ASPB}(\text{OBJMAXSAT}(\mathcal{F}^{\text{ORIG}}))$
2. Preprocessing on WCNF	$\mathcal{F}^1, \text{LB}^1$	$(\mathcal{C}^1, O^1)$
3. Conversion to objective-centric	(F^2, O^2)	$(\text{ASPB}^*(F^2), O^{2*})$
4. Preprocessing on objective-centric	(F^3, O^3)	$(\text{ASPB}^*(F^3), O^{3*})$
5. Constant removal		

Preprocessing flow of MaxPre

	preprocessing (MaxSAT)	*proof (pseudo-Boolean)*
1. Initialization	$\mathcal{F}^{\text{ORIG}}$	$\textsc{asPB}(\textsc{ObjMaxSAT}(\mathcal{F}^{\text{ORIG}}))$
2. Preprocessing on WCNF	$\mathcal{F}^1, \text{LB}^1$	$(\mathcal{C}^1, O^1)$
3. Conversion to objective-centric	(F^2, O^2)	$(\textsc{asPB}^*(F^2), O^{2*})$
4. Preprocessing on objective-centric	(F^3, O^3)	$(\textsc{asPB}^*(F^3), O^{3*})$
5. Constant removal	$(F^4, O^4) = \mathcal{F}^{\text{PREP}}$	$(\textsc{asPB}(F^4), O^4) = \textsc{asPB}(\textsc{ObjMaxSAT}(\mathcal{F}^{\text{PREP}}))$

Proof logging for MaxSAT preprocessing: Practical examples

Preprocessing flow of MaxPre

1. Initialization
2. Preprocessing on WCNF
3. Converstion to objective-centric
4. Preprocessing on objective centric
5. Constant removal + renaming variables

Stage 1: Initialization

Input MaxSAT instance

Hard clauses:

$(x_1 \vee x_2 \vee \bar{x}_3)$
$(\bar{x}_2 \vee x_3 \vee x_4)$
$(\bar{x}_1 \vee \bar{x}_2)$
$(\bar{x}_1 \vee x_2)$

Soft clauses:

$\langle(\bar{x}_4), 2\rangle$
$\langle(x_1, x_3), 4\rangle$

Stage 1: Initialization

Input MaxSAT instance

Hard clauses:

$(x_1 \vee x_2 \vee \bar{x}_3)$
$(\bar{x}_2 \vee x_3 \vee x_4)$
$(\bar{x}_1 \vee \bar{x}_2)$
$(\bar{x}_1 \vee x_2)$

Soft clauses:

$\langle(\bar{x}_4), 2\rangle$
$\langle(x_1, x_3), 4\rangle$

PBO instance (proof)

Constraints:

1: $x_1 + x_2 + \bar{x}_3 \geq 1$
2: $\bar{x}_2 + x_3 + x_4 \geq 1$
3: $\bar{x}_1 + \bar{x}_2 \geq 1$
4: $\bar{x}_1 + x_2 \geq 1$

Stage 1: Initialization

Input MaxSAT instance

Hard clauses:

$(x_1 \vee x_2 \vee \bar{x}_3)$
$(\bar{x}_2 \vee x_3 \vee x_4)$
$(\bar{x}_1 \vee \bar{x}_2)$
$(\bar{x}_1 \vee x_2)$

Soft clauses:

$\langle(\bar{x}_4), 2\rangle$
$\langle(x_1, x_3), 4\rangle$

PBO instance (proof)

Constraints:

1: $x_1 + x_2 + \bar{x}_3 \geq 1$
2: $\bar{x}_2 + x_3 + x_4 \geq 1$
3: $\bar{x}_1 + \bar{x}_2 \geq 1$
4: $\bar{x}_1 + x_2 \geq 1$

5: $x_1 + x_3 + b_5 \geq 1$

Stage 1: Initialization

Input MaxSAT instance

Hard clauses:

$(x_1 \vee x_2 \vee \bar{x}_3)$
$(\bar{x}_2 \vee x_3 \vee x_4)$
$(\bar{x}_1 \vee \bar{x}_2)$
$(\bar{x}_1 \vee x_2)$

Soft clauses:

$\langle(\bar{x}_4), 2\rangle$
$\langle(x_1, x_3), 4\rangle$

PBO instance (proof)

Constraints:

1: $x_1 + x_2 + \bar{x}_3 \geq 1$
2: $\bar{x}_2 + x_3 + x_4 \geq 1$
3: $\bar{x}_1 + \bar{x}_2 \geq 1$
4: $\bar{x}_1 + x_2 \geq 1$

5: $x_1 + x_3 + b_5 \geq 1$

minimize $O \equiv 2x_4 + 4b_5$

Stage 2: Preprocessing on WCNF, removing duplicate clauses

MaxSAT instance (preprocessor)

Soft clauses:

$\ldots$
$\langle(x_1 \lor \bar{x}_2), 2\rangle$
$\langle(x_1 \lor \bar{x}_2), 3\rangle$
$\ldots$

- Replace with a single soft clause $\langle(x_1 \lor \bar{x}_2), 5\rangle$
- **Proof:**
  - Introduce constraints to encode $b_1 = b_2$
    - $b_1 + \bar{b}_2 \geq 1, \omega = \{b_2 \to 0\}$
    - $\bar{b}_1 + b_2 \geq 1, \omega = \{b_1 \to 0\}$
  - Remove $x_1 + \bar{x}_2 + \bar{b}_2 \geq 1$ (RUP)
  - Add $-3b_2 + 3b_1$ to O
  - Remove constraints $b_1 + \bar{b}_2 \geq 1$ and $\bar{b}_1 + b_2 \geq 1$.

Stage 2: Preprocessing on WCNF, removing duplicate clauses

MaxSAT instance (preprocessor)

Soft clauses:

$\ldots$
$\langle (x_1 \vee \bar{x}_2), 2 \rangle$
$\langle (x_1 \vee \bar{x}_2), 3 \rangle$
$\ldots$

PBO instance (proof)

Minimize $O \equiv 2b_1 + 3b_2 + \ldots$
s.t.

$\ldots$
$x_1 + \bar{x}_2 + b_1 \geq 1$
$x_1 + \bar{x}_2 + b_2 \geq 1$

- Replace with a single soft clause $\langle (x_1 \vee \bar{x}_2), 5 \rangle$
- **Proof:**
  - Introduce constraints to encode $b_1 = b_2$
    - $b_1 + \bar{b}_2 \geq 1, \omega = \{b_2 \to 0\}$
    - $\bar{b}_1 + b_2 \geq 1, \omega = \{b_1 \to 0\}$
  - Remove $x_1 + \bar{x}_2 + \bar{b}_2 \geq 1$ (RUP)
  - Add $-3b_2 + 3b_1$ to O
  - Remove constraints $b_1 + \bar{b}_2 \geq 1$ and $\bar{b}_1 + b_2 \geq 1$.

Stage 2: Preprocessing on WCNF, removing duplicate clauses

MaxSAT instance (preprocessor)

Soft clauses:

$\dots$
$\langle (x_1 \vee \bar{x}_2), 2 \rangle$
$\langle (x_1 \vee \bar{x}_2), 3 \rangle$
$\dots$

PBO instance (proof)

Minimize $O \equiv 2b_1 + 3b_2 + \dots$
s.t.
$\dots$
$x_1 + \bar{x}_2 + b_1 \geq 1$
$x_1 + \bar{x}_2 + b_2 \geq 1$

- Replace with a single soft clause $\langle (x_1 \vee \bar{x}_2), 5 \rangle$
- **Proof:**
  - Introduce constraints to encode $b_1 = b_2$
    - $b_1 + \bar{b}_2 \geq 1, \omega = \{b_2 \to 0\}$
    - $\bar{b}_1 + b_2 \geq 1, \omega = \{b_1 \to 0\}$
  - Remove $x_1 + \bar{x}_2 + \bar{b}_2 \geq 1$ (RUP)
  - Add $-3b_2 + 3b_1$ to O
  - Remove constraints $b_1 + \bar{b}_2 \geq 1$ and $\bar{b}_1 + b_2 \geq 1$.

Stage 2: Preprocessing on WCNF, removing duplicate clauses

MaxSAT instance (preprocessor)

Soft clauses:

$\ldots$

$\langle(x_1 \vee \bar{x}_2), 2\rangle$

$\langle(x_1 \vee \bar{x}_2), 3\rangle$

$\ldots$

PBO instance (proof)

Minimize $O \equiv 2b_1 + 3b_2 + \ldots$

s.t.

$\ldots$

$x_1 + \bar{x}_2 + b_1 \geq 1$

$x_1 + \bar{x}_2 + b_2 \geq 1$

- Replace with a single soft clause $\langle(x_1 \vee \bar{x}_2), 5\rangle$
- **Proof:**
  - Introduce constraints to encode $b_1 = b_2$
    - $b_1 + \bar{b}_2 \geq 1, \omega = \{b_2 \to 0\}$
    - $\bar{b}_1 + b_2 \geq 1, \omega = \{b_1 \to 0\}$
  - Remove $x_1 + \bar{x}_2 + \bar{b}_2 \geq 1$ (RUP)
  - Add $-3b_2 + 3b_1$ to O
  - Remove constraints $b_1 + \bar{b}_2 \geq 1$ and $\bar{b}_1 + b_2 \geq 1$.

Stage 2: Preprocessing on WCNF, removing duplicate clauses

MaxSAT instance (preprocessor)

Soft clauses:

$\ldots$
$\langle(x_1 \lor \bar{x}_2), 2\rangle$
$\langle(x_1 \lor \bar{x}_2), 3\rangle$
$\ldots$

PBO instance (proof)

Minimize $O \equiv 2b_1 + 3b_2 + \ldots$
s.t.

$\ldots$
$x_1 + \bar{x}_2 + b_1 \geq 1$
$x_1 + \bar{x}_2 + b_2 \geq 1$

- Replace with a single soft clause $\langle(x_1 \lor \bar{x}_2), 5\rangle$
- **Proof:**
  - Introduce constraints to encode $b_1 = b_2$
    - $b_1 + \bar{b}_2 \geq 1, \omega = \{b_2 \to 0\}$
    - $\bar{b}_1 + b_2 \geq 1, \omega = \{b_1 \to 0\}$
  - Remove $x_1 + \bar{x}_2 + \bar{b}_2 \geq 1$ (RUP)
  - Add $-3b_2 + 3b_1$ to O
  - Remove constraints $b_1 + \bar{b}_2 \geq 1$ and $\bar{b}_1 + b_2 \geq 1$.

Stage 2: Preprocessing on WCNF, removing duplicate clauses

MaxSAT instance (preprocessor)

Soft clauses:

$\dots$
$\langle (x_1 \vee \bar{x}_2), 2 \rangle$
$\langle (x_1 \vee \bar{x}_2), 3 \rangle$
$\dots$

PBO instance (proof)

Minimize $O \equiv 2b_1 + 3b_2 + \dots$
s.t.
$\dots$
$x_1 + \bar{x}_2 + b_1 \geq 1$
$x_1 + \bar{x}_2 + b_2 \geq 1$

- Replace with a single soft clause $\langle (x_1 \vee \bar{x}_2), 5 \rangle$
- **Proof:**
  - Introduce constraints to encode $b_1 = b_2$
    - $b_1 + \bar{b}_2 \geq 1, \omega = \{b_2 \rightarrow 0\}$
    - $\bar{b}_1 + b_2 \geq 1, \omega = \{b_1 \rightarrow 0\}$
  - Remove $x_1 + \bar{x}_2 + \bar{b}_2 \geq 1$ (RUP)
  - Add $-3b_2 + 3b_1$ to O
  - Remove constraints $b_1 + \bar{b}_2 \geq 1$ and $\bar{b}_1 + b_2 \geq 1$.

Stage 2: Preprocessing on WCNF, removing duplicate clauses

MaxSAT instance (preprocessor)

Soft clauses:

$\ldots$
$\langle (x_1 \vee \bar{x}_2), 2 \rangle$
$\langle (x_1 \vee \bar{x}_2), 3 \rangle$
$\ldots$

PBO instance (proof)

Minimize $O \equiv 2b_1 + 3b_2 + \ldots$
s.t.
$\ldots$
$x_1 + \bar{x}_2 + b_1 \geq 1$
$x_1 + \bar{x}_2 + b_2 \geq 1$

- Replace with a single soft clause $\langle (x_1 \vee \bar{x}_2), 5 \rangle$
- **Proof:**
  - Introduce constraints to encode $b_1 = b_2$
    - $b_1 + \bar{b}_2 \geq 1, \omega = \{b_2 \to 0\}$
    - $\bar{b}_1 + b_2 \geq 1, \omega = \{b_1 \to 0\}$
  - Remove $x_1 + \bar{x}_2 + \bar{b}_2 \geq 1$ (RUP)
  - Add $-3b_2 + 3b_1$ to O
  - Remove constraints $b_1 + \bar{b}_2 \geq 1$ and $\bar{b}_1 + b_2 \geq 1$.

Stage 3: Conversion to Objective-centric

- Case where proof is needed:
  - Preprocessor has $\langle (x), w \rangle$
  - PBO instance has $x + b \geq 1$, $O \equiv \cdots + wb + \ldots$.
  - We want to remove b
  - **Proof:**
    - Introduce constraint $\bar{x} + \bar{b} \geq 1$, $\omega = \{b \to 0\}$
    - Add $w\bar{x} - wb$ to O
    - Remove $x + b \geq 1$ and $\bar{x} + \bar{b} \geq 1$

Stage 3: Conversion to Objective-centric

- Case where proof is needed:
  - Preprocessor has $\langle (x), w \rangle$
  - PBO instance has $x + b \geq 1$, $O \equiv \cdots + wb + \ldots$.
  - We want to remove b
  - **Proof:**
    - Introduce constraint $\bar{x} + \bar{b} \geq 1, \omega = \{b \to 0\}$
    - Add $w\bar{x} - wb$ to O
    - Remove $x + b \geq 1$ and $\bar{x} + \bar{b} \geq 1$

Stage 3: Conversion to Objective-centric

- Case where proof is needed:
  - Preprocessor has $\langle (x), w \rangle$
  - PBO instance has $x + b \geq 1$, $O \equiv \dots + wb + \dots$.
  - We want to remove b
  - **Proof:**
    - Introduce constraint $\bar{x} + \bar{b} \geq 1$, $\omega = \{b \rightarrow 0\}$
    - Add $w\bar{x} - wb$ to O
    - Remove $x + b \geq 1$ and $\bar{x} + \bar{b} \geq 1$

Stage 3: Conversion to Objective-centric

- Case where proof is needed:
  - Preprocessor has $\langle (x), w \rangle$
  - PBO instance has $x + b \geq 1$, $O \equiv \cdots + wb + \ldots$.
  - We want to remove b
  - **Proof:**
    - Introduce constraint $\bar{x} + \bar{b} \geq 1$, $\omega = \{b \to 0\}$
    - Add $w\bar{x} - wb$ to O
    - Remove $x + b \geq 1$ and $\bar{x} + \bar{b} \geq 1$

Stage 3: Conversion to Objective-centric

- Case where proof is needed:
  - Preprocessor has $\langle (x), w \rangle$
  - PBO instance has $x + b \geq 1$, $O \equiv \dots + wb + \dots$.
  - We want to remove b
  - **Proof:**
    - Introduce constraint $\bar{x} + \bar{b} \geq 1$, $\omega = \{b \to 0\}$
    - Add $w\bar{x} - wb$ to O
    - Remove $x + b \geq 1$ and $\bar{x} + \bar{b} \geq 1$

Stage 4: Preprocessing on Objective-Centric, 1/3: Subsumed Literal Elimination (SLE)

- Given two (non-objective) literals ℓ_1 and ℓ_2 s.t.
  1. $\{C \mid C \in F, \ell_1 \in C\} \supseteq \{C \mid C \in F, \ell_2 \in C\}$
  2. $\{C \mid C \in F, \overline{\ell_2} \in C\} \supseteq \{C \mid C \in F, \overline{\ell_1} \in C\}$
- SLE fixes $\ell_1 = 1$, $\ell_2 = 0$
- **Proof:**
  - Introduce $\ell_1 \geq 1$, $\overline{\ell}_2 \geq 1$, both with witness $\omega = \{\ell_1 \to 1, \ell_2 \to 0\}$
  - Simplify the constraint database (unit propagate)
  - Delete $\ell_1 \geq 1$ and $\overline{\ell}_2 \geq 1$

Stage 4: Preprocessing on Objective-Centric, 1/3: Subsumed Literal Elimination (SLE)

- Given two (non-objective) literals ℓ_1 and ℓ_2 s.t.
  1. $\{C \mid C \in F, \ell_1 \in C\} \supseteq \{C \mid C \in F, \ell_2 \in C\}$
  2. $\{C \mid C \in F, \overline{\ell}_2 \in C\} \supseteq \{C \mid C \in F, \overline{\ell}_1 \in C\}$
- SLE fixes $\ell_1 = 1$, $\ell_2 = 0$
- **Proof:**
  - Introduce $\ell_1 \geq 1$, $\overline{\ell}_2 \geq 1$, both with witness $\omega = \{\ell_1 \to 1, \ell_2 \to 0\}$
  - Simplify the constraint database (unit propagate)
  - Delete $\ell_1 \geq 1$ and $\overline{\ell}_2 \geq 1$

Stage 4: Preprocessing on Objective-Centric, 1/3: Subsumed Literal Elimination (SLE)

- Given two (non-objective) literals ℓ_1 and ℓ_2 s.t.
  1. $\{C \mid C \in F, \ell_1 \in C\} \supseteq \{C \mid C \in F, \ell_2 \in C\}$
  2. $\{C \mid C \in F, \overline{\ell}_2 \in C\} \supseteq \{C \mid C \in F, \overline{\ell}_1 \in C\}$
- SLE fixes $\ell_1 = 1$, $\ell_2 = 0$
- **Proof:**
  - Introduce $\ell_1 \geq 1$, $\overline{\ell}_2 \geq 1$, both with witness $\omega = \{\ell_1 \to 1, \ell_2 \to 0\}$
  - Simplify the constraint database (unit propagate)
  - Delete $\ell_1 \geq 1$ and $\overline{\ell}_2 \geq 1$

Stage 4: Preprocessing on Objective-Centric, 1/3: Subsumed Literal Elimination (SLE)

- Given two (non-objective) literals ℓ_1 and ℓ_2 s.t.
  1. $\{C \mid C \in F, \ell_1 \in C\} \supseteq \{C \mid C \in F, \ell_2 \in C\}$
  2. $\{C \mid C \in F, \overline{\ell}_2 \in C\} \supseteq \{C \mid C \in F, \overline{\ell}_1 \in C\}$
- SLE fixes $\ell_1 = 1$, $\ell_2 = 0$
- **Proof:**
  - Introduce $\ell_1 \geq 1$, $\overline{\ell}_2 \geq 1$, both with witness $\omega = \{\ell_1 \to 1, \ell_2 \to 0\}$
  - Simplify the constraint database (unit propagate)
  - Delete $\ell_1 \geq 1$ and $\overline{\ell}_2 \geq 1$

Stage 4: Preprocessing on Objective-Centric, 1/3: Subsumed Literal Elimination (SLE)

- Given two (non-objective) literals ℓ_1 and ℓ_2 s.t.
  1. $\{C \mid C \in F, \ell_1 \in C\} \supseteq \{C \mid C \in F, \ell_2 \in C\}$
  2. $\{C \mid C \in F, \overline{\ell}_2 \in C\} \supseteq \{C \mid C \in F, \overline{\ell}_1 \in C\}$
- SLE fixes $\ell_1 = 1$, $\ell_2 = 0$
- **Proof:**
  - Introduce $\ell_1 \geq 1$, $\overline{\ell}_2 \geq 1$, both with witness $\omega = \{\ell_1 \to 1, \ell_2 \to 0\}$
  - Simplify the constraint database (unit propagate)
  - Delete $\ell_1 \geq 1$ and $\overline{\ell}_2 \geq 1$

Stage 4: Preprocessing on Objective-Centric, 2/3: SLE with objective literals

- Given two literals ℓ_1 and ℓ_2 s.t.
  1. $\{C \mid C \in F, \ell_1 \in C\} \supseteq \{C \mid C \in F, \ell_2 \in C\}$
  2. $\{C \mid C \in F, \overline{\ell_2} \in C\} \supseteq \{C \mid C \in F, \overline{\ell_1} \in C\}$
  3. $O \equiv \ldots w_1\ell_1 + w_2\ell_2 + \ldots,\ w_1 \leq w_2$
- Fix $\ell_2 = 0$
- **Proof:**
  - Introduce $\overline{\ell_2} \geq 1$, $\omega = \{\ell_1 \to 1, \ell_2 \to 0\}$
  - Simplify the constraint database
  - Add $-w_2\ell_2$ to O
  - Delete $\overline{\ell_2} \geq 1$

Stage 4: Preprocessing on Objective-Centric, 2/3: SLE with objective literals

- Given two literals ℓ_1 and ℓ_2 s.t.
  1. $\{C \mid C \in F, \ell_1 \in C\} \supseteq \{C \mid C \in F, \ell_2 \in C\}$
  2. $\{C \mid C \in F, \overline{\ell_2} \in C\} \supseteq \{C \mid C \in F, \overline{\ell_1} \in C\}$
  3. $O \equiv \ldots w_1\ell_1 + w_2\ell_2 + \ldots,\ w_1 \leq w_2$
- Fix $\ell_2 = 0$
- **Proof:**
  - Introduce $\overline{\ell_2} \geq 1$, $\omega = \{\ell_1 \to 1, \ell_2 \to 0\}$
  - Simplify the constraint database
  - Add $-w_2\ell_2$ to O
  - Delete $\overline{\ell_2} \geq 1$

ℓ_1	ℓ_2	cost
0	0	0
0	1	w_2
1	0	w_1
1	1	$w_1 + w_2$

Stage 4: Preprocessing on Objective-Centric, 2/3: SLE with objective literals

- Given two literals ℓ_1 and ℓ_2 s.t.
  1. $\{C \mid C \in F, \ell_1 \in C\} \supseteq \{C \mid C \in F, \ell_2 \in C\}$
  2. $\{C \mid C \in F, \overline{\ell_2} \in C\} \supseteq \{C \mid C \in F, \overline{\ell_1} \in C\}$
  3. $O \equiv \ldots w_1\ell_1 + w_2\ell_2 + \ldots, w_1 \leq w_2$
- Fix $\ell_2 = 0$
- **Proof:**
  - Introduce $\overline{\ell_2} \geq 1$, $\omega = \{\ell_1 \to 1, \ell_2 \to 0\}$
  - Simplify the constraint database
  - Add $-w_2\ell_2$ to O
  - Delete $\overline{\ell_2} \geq 1$

ℓ_1	ℓ_2	cost
0	0	0
0	1	w_2
1	0	w_1
1	1	$w_1 + w_2$

Stage 4: Preprocessing on Objective-Centric, 2/3: SLE with objective literals

- Given two literals ℓ_1 and ℓ_2 s.t.
  1. $\{C \mid C \in F, \ell_1 \in C\} \supseteq \{C \mid C \in F, \ell_2 \in C\}$
  2. $\{C \mid C \in F, \overline{\ell_2} \in C\} \supseteq \{C \mid C \in F, \overline{\ell_1} \in C\}$
  3. $O \equiv \ldots w_1\ell_1 + w_2\ell_2 + \ldots,\ w_1 \leq w_2$
- Fix $\ell_2 = 0$
- **Proof:**
  - Introduce $\overline{\ell}_2 \geq 1$, $\omega = \{\ell_1 \to 1, \ell_2 \to 0\}$
  - Simplify the constraint database
  - Add $-w_2\ell_2$ to O
  - Delete $\overline{\ell}_2 \geq 1$

ℓ_1	ℓ_2	cost
0	0	0
0	1	w_2
1	0	w_1
1	1	$w_1 + w_2$

Stage 4: Preprocessing on Objective-Centric, 2/3: SLE with objective literals

- Given two literals ℓ_1 and ℓ_2 s.t.
  1. $\{C \mid C \in F, \ell_1 \in C\} \supseteq \{C \mid C \in F, \ell_2 \in C\}$
  2. $\{C \mid C \in F, \bar{\ell}_2 \in C\} \supseteq \{C \mid C \in F, \bar{\ell}_1 \in C\}$
  3. $O \equiv \ldots w_1\ell_1 + w_2\ell_2 + \ldots,\ w_1 \leq w_2$
- Fix $\ell_2 = 0$
- **Proof:**
  - Introduce $\bar{\ell}_2 \geq 1, \omega = \{\ell_1 \to 1, \ell_2 \to 0\}$
  - Simplify the constraint database
  - Add $-w_2\ell_2$ to O
  - Delete $\bar{\ell}_2 \geq 1$

ℓ_1	ℓ_2	cost
0	0	0
0	1	w_2
1	0	w_1
1	1	$w_1 + w_2$

Stage 4: Preprocessing on Objective-Centric, 2/3: SLE with objective literals

- Given two literals ℓ_1 and ℓ_2 s.t.
  1. $\{C \mid C \in F, \ell_1 \in C\} \supseteq \{C \mid C \in F, \ell_2 \in C\}$
  2. $\{C \mid C \in F, \overline{\ell_2} \in C\} \supseteq \{C \mid C \in F, \overline{\ell_1} \in C\}$
  3. $O \equiv \ldots w_1\ell_1 + w_2\ell_2 + \ldots, \; w_1 \leq w_2$
- Fix $\ell_2 = 0$
- **Proof:**
  - Introduce $\overline{\ell_2} \geq 1$, $\omega = \{\ell_1 \to 1, \ell_2 \to 0\}$
  - Simplify the constraint database
  - Add $-w_2\ell_2$ to O
  - Delete $\overline{\ell_2} \geq 1$

ℓ_1	ℓ_2	cost
0	0	0
0	1	w_2
1	0	w_1
1	1	$w_1 + w_2$

Stage 4: Preprocessing on Objective-Centric, 2/3: SLE with objective literals

- Given two literals ℓ_1 and ℓ_2 s.t.
  1. $\{C \mid C \in F, \ell_1 \in C\} \supseteq \{C \mid C \in F, \ell_2 \in C\}$
  2. $\{C \mid C \in F, \bar{\ell}_2 \in C\} \supseteq \{C \mid C \in F, \bar{\ell}_1 \in C\}$
  3. $O \equiv \ldots w_1 \ell_1 + w_2 \ell_2 + \ldots,\ w_1 \leq w_2$
- Fix $\ell_2 = 0$
- **Proof:**
  - Introduce $\bar{\ell}_2 \geq 1, \omega = \{\ell_1 \to 1, \ell_2 \to 0\}$
  - Simplify the constraint database
  - Add $-w_2 \ell_2$ to O
  - Delete $\bar{\ell}_2 \geq 1$

ℓ_1	ℓ_2	cost
0	0	0
0	1	w_2
1	0	w_1
1	1	$w_1 + w_2$

Stage 4: Preprocessing on Objective-Centric, 2/3: SLE with objective literals

- Given two literals ℓ_1 and ℓ_2 s.t.
  1. $\{C \mid C \in F, \ell_1 \in C\} \supseteq \{C \mid C \in F, \ell_2 \in C\}$
  2. $\{C \mid C \in F, \overline{\ell_2} \in C\} \supseteq \{C \mid C \in F, \overline{\ell_1} \in C\}$
  3. $O \equiv \ldots w_1\ell_1 + w_2\ell_2 + \ldots,\ w_1 \leq w_2$
- Fix $\ell_2 = 0$
- **Proof:**
  - Introduce $\overline{\ell}_2 \geq 1$, $\omega = \{\ell_1 \to 1, \ell_2 \to 0\}$
  - Simplify the constraint database
  - Add $-w_2\ell_2$ to O
  - Delete $\overline{\ell}_2 \geq 1$

ℓ_1	ℓ_2	cost
0	0	0
0	1	w_2
1	0	w_1
1	1	$w_1 + w_2$

Stage 4: Preprocessing on Objective-Centric, 2/3: SLE with objective literals

- Given two literals ℓ_1 and ℓ_2 s.t.
  1. $\{C \mid C \in F, \ell_1 \in C\} \supseteq \{C \mid C \in F, \ell_2 \in C\}$
  2. $\{C \mid C \in F, \bar{\ell}_2 \in C\} \supseteq \{C \mid C \in F, \bar{\ell}_1 \in C\}$
  3. $O \equiv \ldots w_1\ell_1 + w_2\ell_2 + \ldots,\ w_1 \leq w_2$
- Fix $\ell_2 = 0$
- **Proof:**
  - Introduce $\bar{\ell}_2 \geq 1$, $\omega = \{\ell_1 \to 1, \ell_2 \to 0\}$
  - Simplify the constraint database
  - Add $-w_2\ell_2$ to O
  - Delete $\bar{\ell}_2 \geq 1$

ℓ_1	ℓ_2	cost
0	0	0
0	1	w_2
1	0	w_1
1	1	$w_1 + w_2$

Stage 4: Preprocessing on Objective-Centric, 2/3: Hardening

- Given
  - τ s.t. $O(\tau) = \text{UB}$
  - b s.t. $O \equiv \cdots + wb + \ldots$, $w > \text{UB}$
- Fix $b = 0$
- In optimality proofs, objective improving constraints can be used
- We need something else
- **Proof:**
  - add $\bar{b}_i \geq 1$ with witness $\omega = \tau$

Stage 4: Preprocessing on Objective-Centric, 2/3: Hardening

- Given
  - τ s.t. $O(\tau) = \text{UB}$
  - b s.t. $O \equiv \dots + wb + \dots$, $w > \text{UB}$
- Fix $b = 0$
- In optimality proofs, objective improving constraints can be used
- We need something else
- **Proof:**
  - add $\bar{b}_i \geq 1$ with witness $\omega = \tau$

Stage 4: Preprocessing on Objective-Centric, 2/3: Hardening

- Given
  - τ s.t. $O(\tau) = \text{UB}$
  - b s.t. $O \equiv \dots + wb + \dots$, $w > \text{UB}$
- Fix $b = 0$
- In optimality proofs, objective improving constraints can be used
- We need something else
- **Proof:**
  - add $\bar{b}_i \geq 1$ with witness $\omega = \tau$

Stage 4: Preprocessing on Objective-Centric, 3/3: Label matching

- Assume that
  1. F has clauses $C \lor b_C$ and $D \lor b_D$
  2. $O \equiv \dots + wb_C + wb_D + \dots$
  3. $C \lor D$ is a tautology
  4. b_C and b_D do not appear in other clauses

- Label matching
  - Replaces $C \lor b_C$ with $C \lor b_{CD}$
  - Replaces $D \lor b_D$ with $D \lor b_{CD}$
  - Adds $wb_{CD} - wb_C - wb_D$ to O.

- **Proof**, w.l.o.g. assume $\ell \in C$ and $\bar{\ell} \in D$
  - Introduce $\bar{b}_C + \bar{b}_D \geq 1$, $\omega = \{b_C \to \bar{\ell}, b_D \to \ell\}$
  - Introduce $b_{CD} = b_C + b_D$
  - Add $wb_{CD} - wb_C - wb_D$ to O
  - Introduce ASPB($C \lor b_{CD}$) and ASPB($D \lor b_{CD}$) (RUP)
  - Delete ASPB($C \lor b_C$) and ASPB($D \lor b_D$), $\omega = \{b_C \to \bar{\ell}, b_D \to \ell\}$
  - Delete constraints encoding $b_{CD} = b_C + b_D$
  - Delete constraint $\bar{b}_C + \bar{b}_D \geq 1$

Stage 4: Preprocessing on Objective-Centric, 3/3: Label matching

- Assume that
  1. F has clauses $C \vee b_C$ and $D \vee b_D$
  2. $O \equiv \dots + wb_C + wb_D + \dots$
  3. $C \vee D$ is a tautology
  4. b_C and b_D do not appear in other clauses

- Label matching
  - Replaces $C \vee b_C$ with $C \vee b_{CD}$
  - Replaces $D \vee b_D$ with $D \vee b_{CD}$
  - Adds $wb_{CD} - wb_C - wb_D$ to O.

- **Proof**, w.l.o.g. assume $\ell \in C$ and $\bar{\ell} \in D$
  - Introduce $\bar{b}_C + \bar{b}_D \geq 1$, $\omega = \{b_C \to \bar{\ell}, b_D \to \ell\}$
  - Introduce $b_{CD} = b_C + b_D$
  - Add $wb_{CD} - wb_C - wb_D$ to O
  - Introduce ASPB($C \vee b_{CD}$) and ASPB($D \vee b_{CD}$) (RUP)
  - Delete ASPB($C \vee b_C$) and ASPB($D \vee b_D$), $\omega = \{b_C \to \bar{\ell}, b_D \to \ell\}$
  - Delete constraints encoding $b_{CD} = b_C + b_D$
  - Delete constraint $\bar{b}_C + \bar{b}_D \geq 1$

Stage 4: Preprocessing on Objective-Centric, 3/3: Label matching

- Assume that
  1. F has clauses $C \lor b_C$ and $D \lor b_D$
  2. $O \equiv \dots + w b_C + w b_D + \dots$
  3. $C \lor D$ is a tautology
  4. b_C and b_D do not appear in other clauses
- Label matching
  - Replaces $C \lor b_C$ with $C \lor b_{CD}$
  - Replaces $D \lor b_D$ with $D \lor b_{CD}$
  - Adds $w b_{CD} - w b_C - w b_D$ to O.
- **Proof**, w.l.o.g. assume $\ell \in C$ and $\bar{\ell} \in D$
  - Introduce $\bar{b}_C + \bar{b}_D \geq 1$, $\omega = \{b_C \to \bar{\ell}, b_D \to \ell\}$
  - Introduce $b_{CD} = b_C + b_D$
  - Add $w b_{CD} - w b_C - w b_D$ to O
  - Introduce ASPB($C \lor b_{CD}$) and ASPB($D \lor b_{CD}$) (RUP)
  - Delete ASPB($C \lor b_C$) and ASPB($D \lor b_D$), $\omega = \{b_C \to \bar{\ell}, b_D \to \ell\}$
  - Delete constraints encoding $b_{CD} = b_C + b_D$
  - Delete constraint $\bar{b}_C + \bar{b}_D \geq 1$

Stage 4: Preprocessing on Objective-Centric, 3/3: Label matching

- Assume that
  1. F has clauses $C \vee b_C$ and $D \vee b_D$
  2. $O \equiv \cdots + wb_C + wb_D + \ldots$
  3. $C \vee D$ is a tautology
  4. b_C and b_D do not appear in other clauses
- Label matching
  - Replaces $C \vee b_C$ with $C \vee b_{CD}$
  - Replaces $D \vee b_D$ with $D \vee b_{CD}$
  - Adds $wb_{CD} - wb_C - wb_D$ to O.
- **Proof**, w.l.o.g. assume $\ell \in C$ and $\bar{\ell} \in D$
  - Introduce $\bar{b}_C + \bar{b}_D \geq 1$, $\omega = \{b_C \to \bar{\ell}, b_D \to \ell\}$
  - Introduce $b_{CD} = b_C + b_D$
  - Add $wb_{CD} - wb_C - wb_D$ to O
  - Introduce ASPB($C \vee b_{CD}$) and ASPB($D \vee b_{CD}$) (RUP)
  - Delete ASPB($C \vee b_C$) and ASPB($D \vee b_D$), $\omega = \{b_C \to \bar{\ell}, b_D \to \ell\}$
  - Delete constraints encoding $b_{CD} = b_C + b_D$
  - Delete constraint $\bar{b}_C + \bar{b}_D \geq 1$

Stage 4: Preprocessing on Objective-Centric, 3/3: Label matching

- Assume that
  1. F has clauses $C \vee b_C$ and $D \vee b_D$
  2. $O \equiv \dots + wb_C + wb_D + \dots$
  3. $C \vee D$ is a tautology
  4. b_C and b_D do not appear in other clauses
- Label matching
  - Replaces $C \vee b_C$ with $C \vee b_{CD}$
  - Replaces $D \vee b_D$ with $D \vee b_{CD}$
  - Adds $wb_{CD} - wb_C - wb_D$ to O.
- **Proof**, w.l.o.g. assume $\ell \in C$ and $\bar{\ell} \in D$
  - Introduce $\bar{b}_C + \bar{b}_D \geq 1$, $\omega = \{b_C \to \bar{\ell}, b_D \to \ell\}$
  - Introduce $b_{CD} = b_C + b_D$
  - Add $wb_{CD} - wb_C - wb_D$ to O
  - Introduce ASPB($C \vee b_{CD}$) and ASPB($D \vee b_{CD}$) (RUP)
  - Delete ASPB($C \vee b_C$) and ASPB($D \vee b_D$), $\omega = \{b_C \to \bar{\ell}, b_D \to \ell\}$
  - Delete constraints encoding $b_{CD} = b_C + b_D$
  - Delete constraint $\bar{b}_C + \bar{b}_D \geq 1$

Stage 4: Preprocessing on Objective-Centric, 3/3: Label matching

- Assume that
  1. F has clauses $C \vee b_C$ and $D \vee b_D$
  2. $O \equiv \dots + wb_C + wb_D + \dots$
  3. $C \vee D$ is a tautology
  4. b_C and b_D do not appear in other clauses

- Label matching
  - Replaces $C \vee b_C$ with $C \vee b_{CD}$
  - Replaces $D \vee b_D$ with $D \vee b_{CD}$
  - Adds $wb_{CD} - wb_C - wb_D$ to O.

- **Proof**, w.l.o.g. assume $\ell \in C$ and $\bar{\ell} \in D$
  - Introduce $\bar{b}_C + \bar{b}_D \geq 1$, $\omega = \{b_C \to \bar{\ell}, b_D \to \ell\}$
  - Introduce $b_{CD} = b_C + b_D$
  - Add $wb_{CD} - wb_C - wb_D$ to O
  - Introduce ASPB($C \vee b_{CD}$) and ASPB($D \vee b_{CD}$) (RUP)
  - Delete ASPB($C \vee b_C$) and ASPB($D \vee b_D$), $\omega = \{b_C \to \bar{\ell}, b_D \to \ell\}$
  - Delete constraints encoding $b_{CD} = b_C + b_D$
  - Delete constraint $\bar{b}_C + \bar{b}_D \geq 1$

Stage 4: Preprocessing on Objective-Centric, 3/3: Label matching

- Assume that
  1. F has clauses $C \lor b_C$ and $D \lor b_D$
  2. $O \equiv \cdots + wb_C + wb_D + \ldots$
  3. $C \lor D$ is a tautology
  4. b_C and b_D do not appear in other clauses
- Label matching
  - Replaces $C \lor b_C$ with $C \lor b_{CD}$
  - Replaces $D \lor b_D$ with $D \lor b_{CD}$
  - Adds $wb_{CD} - wb_C - wb_D$ to O.
- **Proof**, w.l.o.g. assume $\ell \in C$ and $\bar{\ell} \in D$
  - Introduce $\bar{b}_C + \bar{b}_D \geq 1, \omega = \{b_C \to \bar{\ell}, b_D \to \ell\}$
  - Introduce $b_{CD} = b_C + b_D$
  - Add $wb_{CD} - wb_C - wb_D$ to O
  - Introduce ASPB($C \lor b_{CD}$) and ASPB($D \lor b_{CD}$) (RUP)
  - Delete ASPB($C \lor b_C$) and ASPB($D \lor b_D$), $\omega = \{b_C \to \bar{\ell}, b_D \to \ell\}$
  - Delete constraints encoding $b_{CD} = b_C + b_D$
  - Delete constraint $\bar{b}_C + \bar{b}_D \geq 1$

Stage 4: Preprocessing on Objective-Centric, 3/3: Label matching

- Assume that
  1. F has clauses $C \lor b_C$ and $D \lor b_D$
  2. $O \equiv \dots + wb_C + wb_D + \dots$
  3. $C \lor D$ is a tautology
  4. b_C and b_D do not appear in other clauses

- Label matching
  - Replaces $C \lor b_C$ with $C \lor b_{CD}$
  - Replaces $D \lor b_D$ with $D \lor b_{CD}$
  - Adds $wb_{CD} - wb_C - wb_D$ to O.

- **Proof**, w.l.o.g. assume $\ell \in C$ and $\bar{\ell} \in D$
  - Introduce $\bar{b}_C + \bar{b}_D \geq 1$, $\omega = \{b_C \to \bar{\ell}, b_D \to \ell\}$
  - Introduce $b_{CD} = b_C + b_D$
  - Add $wb_{CD} - wb_C - wb_D$ to O
  - Introduce ASPB($C \lor b_{CD}$) and ASPB($D \lor b_{CD}$) (RUP)
  - Delete ASPB($C \lor b_C$) and ASPB($D \lor b_D$), $\omega = \{b_C \to \bar{\ell}, b_D \to \ell\}$
  - Delete constraints encoding $b_{CD} = b_C + b_D$
  - Delete constraint $\bar{b}_C + \bar{b}_D \geq 1$

Stage 4: Preprocessing on Objective-Centric, 3/3: Label matching

- Assume that
  1. F has clauses $C \lor b_C$ and $D \lor b_D$
  2. $O \equiv \cdots + wb_C + wb_D + \ldots$
  3. $C \lor D$ is a tautology
  4. b_C and b_D do not appear in other clauses

- Label matching
  - Replaces $C \lor b_C$ with $C \lor b_{CD}$
  - Replaces $D \lor b_D$ with $D \lor b_{CD}$
  - Adds $wb_{CD} - wb_C - wb_D$ to O.

- **Proof**, w.l.o.g. assume $\ell \in C$ and $\bar{\ell} \in D$
  - Introduce $\bar{b}_C + \bar{b}_D \geq 1$, $\omega = \{b_C \to \bar{\ell}, b_D \to \ell\}$
  - Introduce $b_{CD} = b_C + b_D$
  - Add $wb_{CD} - wb_C - wb_D$ to O
  - Introduce ASPB($C \lor b_{CD}$) and ASPB($D \lor b_{CD}$) (RUP)
  - Delete ASPB($C \lor b_C$) and ASPB($D \lor b_D$), $\omega = \{b_C \to \bar{\ell}, b_D \to \ell\}$
  - Delete constraints encoding $b_{CD} = b_C + b_D$
  - Delete constraint $\bar{b}_C + \bar{b}_D \geq 1$

Stage 4: Preprocessing on Objective-Centric, 3/3: Label matching

- Assume that
  1. F has clauses $C \lor b_C$ and $D \lor b_D$
  2. $O \equiv \dots + wb_C + wb_D + \dots$
  3. $C \lor D$ is a tautology
  4. b_C and b_D do not appear in other clauses
- Label matching
  - Replaces $C \lor b_C$ with $C \lor b_{CD}$
  - Replaces $D \lor b_D$ with $D \lor b_{CD}$
  - Adds $wb_{CD} - wb_C - wb_D$ to O.
- **Proof**, w.l.o.g. assume $\ell \in C$ and $\bar{\ell} \in D$
  - Introduce $\bar{b}_C + \bar{b}_D \geq 1$, $\omega = \{b_C \to \bar{\ell}, b_D \to \ell\}$
  - Introduce $b_{CD} = b_C + b_D$
  - Add $wb_{CD} - wb_C - wb_D$ to O
  - Introduce ASPB($C \lor b_{CD}$) and ASPB($D \lor b_{CD}$) (RUP)
  - Delete ASPB($C \lor b_C$) and ASPB($D \lor b_D$), $\omega = \{b_C \to \bar{\ell}, b_D \to \ell\}$
  - Delete constraints encoding $b_{CD} = b_C + b_D$
  - Delete constraint $\bar{b}_C + \bar{b}_D \geq 1$

Stage 5: Constant removal and variable renaming

- Preprocessor produces a MaxSAT instance in WCNF
- VeriPB verifies that the output WNCF (converted to PBO) matches the database at the end of the proof
  - Remove the constant term from the objective function
    - Hard clause (x_{LB}), soft clause $\langle(\bar{x}_{\text{LB}}), \text{LB}\rangle$
  - Rename variables (if necessary)
    - For each x_i, reify $t_{x_i} \leftrightarrow x_i$
    - Derive constraints with t-variables, remove the original constraints
    - Repeat to get desired variable names

Stage 5: Constant removal and variable renaming

- Preprocessor produces a MaxSAT instance in WCNF
- VeriPB verifies that the output WNCF (converted to PBO) matches the database at the end of the proof
  - Remove the constant term from the objective function
    - Hard clause (x_{LB}), soft clause $\langle(\bar{x}_{\text{LB}}), \text{LB}\rangle$
  - Rename variables (if necessary)
    - For each x_i, reify $t_{x_i} \leftrightarrow x_i$
    - Derive constraints with t-variables, remove the original constraints
    - Repeat to get desired variable names

Stage 5: Constant removal and variable renaming

- Preprocessor produces a MaxSAT instance in WCNF
- VeriPB verifies that the output WNCF (converted to PBO) matches the database at the end of the proof
  - Remove the constant term from the objective function
    - Hard clause (x_{LB}), soft clause $\langle(\bar{x}_{\text{LB}}), \text{LB}\rangle$
  - Rename variables (if necessary)
    - For each x_i, reify $t_{x_i} \leftrightarrow x_i$
    - Derive constraints with t-variables, remove the original constraints
    - Repeat to get desired variable names

Stage 5: Constant removal and variable renaming

- Preprocessor produces a MaxSAT instance in WCNF
- VeriPB verifies that the output WNCF (converted to PBO) matches the database at the end of the proof
  - Remove the constant term from the objective function
    - Hard clause (x_{LB}), soft clause $\langle(\bar{x}_{\text{LB}}), \text{LB}\rangle$
  - Rename variables (if necessary)
    - For each x_i, reify $t_{x_i} \leftrightarrow x_i$
    - Derive constraints with t-variables, remove the original constraints
    - Repeat to get desired variable names

Stage 5: Constant removal and variable renaming

- Preprocessor produces a MaxSAT instance in WCNF
- VeriPB verifies that the output WNCF (converted to PBO) matches the database at the end of the proof
  - Remove the constant term from the objective function
    - Hard clause (x_{LB}), soft clause $\langle(\bar{x}_{\text{LB}}), \text{LB}\rangle$
  - Rename variables (if necessary)
    - For each x_i, reify $t_{x_i} \leftrightarrow x_i$
    - Derive constraints with t-variables, remove the original constraints
    - Repeat to get desired variable names

Conclusion

- Proof-logging for stand-alone MaxSAT preprocessor with VERIPB
  - 15+ preprocessing techniques implemented in MAXPRE
- Seems to work well
- End-to-end formally verified proof logging with CAKEPB
- First practical tool for even verifying (two-way) equisatisfiability