Deep Dive into CDCL Pseudo-Boolean Solvers

Romain Wallon

May 20th, 2021

Laboratoire d'Informatique de l'X (LIX), École Polytechnique

Context

Boolean Satisfiability

The satisfiability problem (SAT) is the first problem proven to be NP-complete (Cook, 1971)

Given a CNF formula Σ, this problem is determining whether there exists an assignment of the (Boolean) variables of Σ such that this formula evaluates to true

Boolean Satisfiability

The satisfiability problem (SAT) is the first problem proven to be NP-complete (Cook, 1971)

Given a CNF formula Σ, this problem is determining whether there exists an assignment of the (Boolean) variables of Σ such that this formula evaluates to true

$$(a \vee b \vee \bar{c}) \wedge (a \vee \bar{b} \vee c) \wedge (\bar{a} \vee \bar{b} \vee c)$$

Boolean Satisfiability

The satisfiability problem (SAT) is the first problem proven to be NP-complete (Cook, 1971)

Given a CNF formula Σ, this problem is determining whether there exists an assignment of the (Boolean) variables of Σ such that this formula evaluates to true

$$(a \vee b \vee \bar{c}) \wedge (a \vee \bar{b} \vee c) \wedge (\bar{a} \vee \bar{b} \vee c)$$

Boolean Satisfiability

The satisfiability problem (SAT) is the first problem proven to be NP-complete (Cook, 1971)

Given a CNF formula Σ, this problem is determining whether there exists an assignment of the (Boolean) variables of Σ such that this formula evaluates to true

$$(a \vee b \vee \bar{c}) \wedge (a \vee \bar{b} \vee c) \wedge (\bar{a} \vee \bar{b} \vee c)$$

In the early 2000s, a revolution in the architecture of SAT solvers happened, with the wide adoption of the CDCL approach (GRASP) and the use of efficient heuristics and data structures (Chaff, MiniSat)

Boolean Satisfiability

The satisfiability problem (SAT) is the first problem proven to be NP-complete (Cook, 1971)

Given a CNF formula Σ, this problem is determining whether there exists an assignment of the (Boolean) variables of Σ such that this formula evaluates to true

$$(a \vee b \vee \bar{c}) \wedge (a \vee \bar{b} \vee c) \wedge (\bar{a} \vee \bar{b} \vee c)$$

In the early 2000s, a revolution in the architecture of SAT solvers happened, with the wide adoption of the CDCL approach (GRASP) and the use of efficient heuristics and data structures (Chaff, MiniSat)

Modern SAT solvers (Glucose, CaDiCaL, Kissat) can now deal with problems containing millions of variables and clauses

Limits of SAT Solvers

So-called "modern" SAT solvers are very efficient in practice, but some instances remain completely out of reach for these solvers, due to the weakness of the resolution proof system they use internally

Limits of SAT Solvers

So-called "modern" SAT solvers are very efficient in practice, but some instances remain completely out of reach for these solvers, due to the weakness of the resolution proof system they use internally

This is particularly for instances requiring the ability to count, such as pigeonhole-principle formulae, stating that *"n pigeons do not fit in* $n-1$ *holes"*

Limits of SAT Solvers

So-called "modern" SAT solvers are very efficient in practice, but some instances remain completely out of reach for these solvers, due to the weakness of the resolution proof system they use internally

This is particularly for instances requiring the ability to count, such as pigeonhole-principle formulae, stating that *"n pigeons do not fit in $n-1$ holes"*

While modern SAT solvers perform poorly on such instances for $n > 20$, PB solvers based on cutting-planes may solve them in linear time

Pseudo-Boolean (PB) Constraints

We consider conjunctions of linear equations or inequations over Boolean variables of the form:

$$\sum_{i=1}^{n} \alpha_i \ell_i \bigtriangleup \delta$$

in which

- the coefficients α_i are integers
- ℓ_i are literals, i.e., a variable v or its negation $\bar{v} = 1 - v$
- $\bigtriangleup$ is a relational operator among $\{<, \leq, =, \geq, >\}$
- the degree δ is an integer

Pseudo-Boolean (PB) Constraints

We consider conjunctions of linear equations or inequations over Boolean variables of the form:

$$\sum_{i=1}^{n} \alpha_i \ell_i \vartriangle \delta$$

in which

- the coefficients α_i are integers
- ℓ_i are literals, i.e., a variable v or its negation $\bar{v} = 1 - v$
- $\vartriangle$ is a relational operator among $\{<, \leq, =, \geq, >\}$
- the degree δ is an integer

For example:

$$-3a + 4b - 7c + d \leq -5$$

Normalized PB Constraints

Without loss of generality, we consider conjunctions of normalized PB constraints of the form:

$$\sum_{i=1}^{n} \alpha_i \ell_i \geq \delta$$

in which

- the coefficients α_i are non-negative integers
- ℓ_i are literals
- the degree δ is a non-negative integer

Normalized PB Constraints

Without loss of generality, we consider conjunctions of normalized PB constraints of the form:

$$\sum_{i=1}^{n} \alpha_i \ell_i \geq \delta$$

in which

- the coefficients α_i are non-negative integers
- ℓ_i are literals
- the degree δ is a non-negative integer

For example:

$$-3a + 4b - 7c + d \leq -5 \equiv 3a + 4\bar{b} + 7c + \bar{d} \geq 10$$

The CDCL Architecture

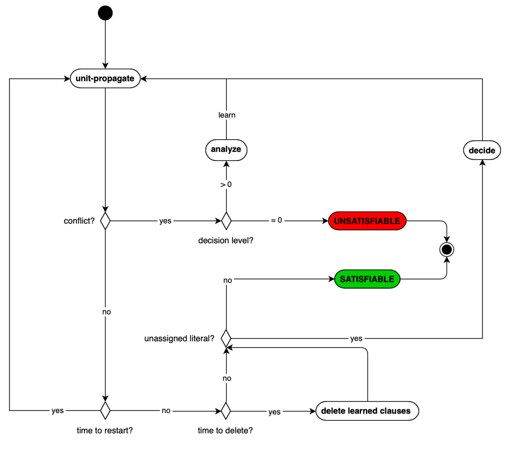


Figure 1: Overview of the CDCL algorithm

Extending the CDCL Architecture

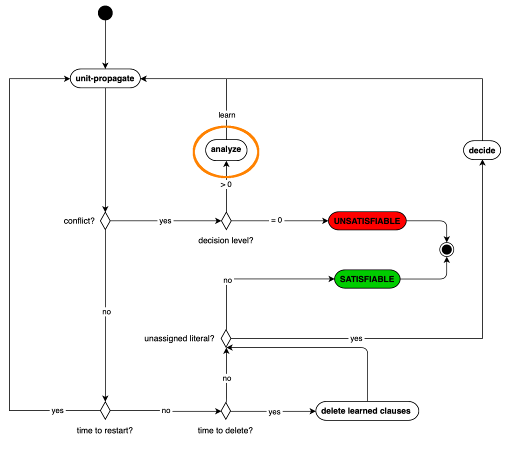


Figure 2: Use of the proof system in the CDCL algorithm

Fitting Cutting-Planes into the CDCL Architecture

Propagations in PB Constraints

The constraint $3a + 4\bar{b} + 7c + \bar{d} \geq 10$ propagates c to true

Propagations in PB Constraints

The constraint $3a + 4\bar{b} + 7c + \bar{d} \geq 10$ propagates c to true

- A PB constraint can propagate truth values without any assignment

Propagations in PB Constraints

The constraint $3a + 4\bar{b} + 7c + \bar{d} \geq 10$ propagates c to true

- A PB constraint can propagate truth values without any assignment
- A PB constraint can propagate multiple truth values at different decision levels

Propagations in PB Constraints

The constraint $3a + 4\bar{b} + 7c + \bar{d} \geq 10$ propagates c to true

- A PB constraint can propagate truth values without any assignment
- A PB constraint can propagate multiple truth values at different decision levels

The constraint above can be rewritten as $c \wedge 3a + 4\bar{b} + \bar{d} \geq 3$
but also as $c \wedge (a \vee \bar{b})$

Cutting Planes and Generalized Resolution

Many PB solvers have been designed based on the Generalized Resolution (Hooker, 1988).

$$\frac{\alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta_1 \qquad \beta\overline{\ell} + \sum_{i=1}^{n} \beta_i \ell_i \geq \delta_2}{\sum_{i=1}^{n} (\beta\alpha_i + \alpha\beta_i)\ell_i \geq \beta\delta_1 + \alpha\delta_2 - \alpha\beta} \text{(cancellation)}$$

$$\frac{\sum_{i=1}^{n} \alpha_i \ell_i \geq \delta}{\sum_{i=1}^{n} \min(\alpha_i, \delta)\ell_i \geq \delta} \text{(saturation)}$$

Cutting Planes and Generalized Resolution

Many PB solvers have been designed based on the Generalized Resolution (Hooker, 1988).

$$\frac{\alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta_1 \qquad \beta\overline{\ell} + \sum_{i=1}^{n} \beta_i \ell_i \geq \delta_2}{\sum_{i=1}^{n} (\beta\alpha_i + \alpha\beta_i)\ell_i \geq \beta\delta_1 + \alpha\delta_2 - \alpha\beta} \text{ (cancellation)}$$

$$\frac{\sum_{i=1}^{n} \alpha_i \ell_i \geq \delta}{\sum_{i=1}^{n} \min(\alpha_i, \delta)\ell_i \geq \delta} \text{ (saturation)}$$

Cutting Planes and Generalized Resolution

Many PB solvers have been designed based on the Generalized Resolution (Hooker, 1988).

$$\frac{\alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta_1 \qquad \beta\bar{\ell} + \sum_{i=1}^{n} \beta_i \ell_i \geq \delta_2}{\sum_{i=1}^{n} (\beta\alpha_i + \alpha\beta_i)\ell_i \geq \beta\delta_1 + \alpha\delta_2 - \alpha\beta} \text{ (cancellation)}$$

$$\frac{\sum_{i=1}^{n} \alpha_i \ell_i \geq \delta}{\sum_{i=1}^{n} \min(\alpha_i, \delta)\ell_i \geq \delta} \text{ (saturation)}$$

Cutting Planes and Generalized Resolution

Many PB solvers have been designed based on the Generalized Resolution (Hooker, 1988).

$$\frac{\alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta_1 \qquad \beta\bar{\ell} + \sum_{i=1}^{n} \beta_i \ell_i \geq \delta_2}{\sum_{i=1}^{n}(\beta\alpha_i + \alpha\beta_i)\ell_i \geq \beta\delta_1 + \alpha\delta_2 - \alpha\beta} \text{ (cancellation)}$$

$$\frac{\sum_{i=1}^{n} \alpha_i \ell_i \geq \delta}{\sum_{i=1}^{n} \min(\alpha_i, \delta)\ell_i \geq \delta} \text{ (saturation)}$$

Cutting Planes and Generalized Resolution

Many PB solvers have been designed based on the Generalized Resolution (Hooker, 1988).

$$\frac{\alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta_1 \qquad \beta\bar{\ell} + \sum_{i=1}^{n} \beta_i \ell_i \geq \delta_2}{\sum_{i=1}^{n}(\beta\alpha_i + \alpha\beta_i)\ell_i \geq \beta\delta_1 + \alpha\delta_2 - \alpha\beta} \text{ (cancellation)}$$

$$\frac{\sum_{i=1}^{n} \alpha_i \ell_i \geq \delta}{\sum_{i=1}^{n} \min(\alpha_i, \delta)\ell_i \geq \delta} \text{ (saturation)}$$

Cutting Planes and Generalized Resolution

Many PB solvers have been designed based on the Generalized Resolution (Hooker, 1988).

$$\frac{\alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta_1 \qquad \beta\bar{\ell} + \sum_{i=1}^{n} \beta_i \ell_i \geq \delta_2}{\sum_{i=1}^{n} (\beta\alpha_i + \alpha\beta_i)\ell_i \geq \beta\delta_1 + \alpha\delta_2 - \alpha\beta} \text{ (cancellation)}$$

$$\frac{\sum_{i=1}^{n} \alpha_i \ell_i \geq \delta}{\sum_{i=1}^{n} \min(\alpha_i, \delta)\ell_i \geq \delta} \text{ (saturation)}$$

As with the resolution rule in classical SAT solvers, these two rules can be used to learn new constraints during conflict analysis

Analyzing Conflicts

Suppose that we have the following constraints:

$6\bar{b} + 6c + 4e + f + g + h \geq 7$ $\qquad$ $5a + 4b + c + d \geq 6$

Analyzing Conflicts

Suppose that we have the following constraints:

$6\overline{b} + 6c + 4e + f + g + h \geq 7$

$5a + 4b + c + d \geq 6$

Analyzing Conflicts

Suppose that we have the following constraints:

$6\bar{b} + 6c + 4e + f + g + h \geq 7$

$5a + 4b + c + d \geq 6$

Analyzing Conflicts

Suppose that we have the following constraints:

$6\bar{b} + 6c + 4e + f + g + h \geq 7$

$5a + 4b + c + d \geq 6$

Analyzing Conflicts

Suppose that we have the following constraints:

$6\bar{b} + 6c + 4e + f + g + h \geq 7$

(reason for $\bar{b}$)

$5a + 4b + c + d \geq 6$

Analyzing Conflicts

Suppose that we have the following constraints:

$6\bar{b} + 6c + 4e + f + g + h \geq 7$ *(reason for $\bar{b}$)*

$5a + 4b + c + d \geq 6$ *(conflict)*

Analyzing Conflicts

Suppose that we have the following constraints:

$6\bar{b} + 6c + 4e + f + g + h \geq 7$ $\quad$ $5a + 4b + c + d \geq 6$

(*reason* for $\bar{b}$) $\quad$ (*conflict*)

This conflict is analyzed by applying the cancellation rule as follows:

$$\frac{6\bar{b} + 6c + 4e + f + g + h \geq 7 \qquad 5a + 4b + c + d \geq 6}{15a + 15c + 8e + 3d + 2f + 2g + 2h \geq 20}$$

Analyzing Conflicts

Suppose that we have the following constraints:

$6\bar{b} + 6c + 4e + f + g + h \geq 7$ *(reason for $\bar{b}$)*

$5a + 4b + c + d \geq 6$ *(conflict)*

This conflict is analyzed by applying the cancellation rule as follows:

$$\frac{6\bar{b} + 6c + 4e + f + g + h \geq 7 \qquad 5a + 4b + c + d \geq 6}{15a + 15c + 8e + 3d + 2f + 2g + 2h \geq 20}$$

The constraint we obtain here is no longer conflicting!

Weakening

To preserve the conflict, the weakening rule must be used:

$$\frac{\alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta}{\sum_{i=1}^{n} \alpha_i \ell_i \geq \delta - \alpha} \text{ (weakening)}$$

$$\frac{\alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq d \qquad k \in \mathbb{N} \qquad 0 < k \leq \alpha}{(\alpha - k)\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta - k} \text{ (partial weakening)}$$

Weakening

To preserve the conflict, the weakening rule must be used:

$$\frac{\alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta}{\sum_{i=1}^{n} \alpha_i \ell_i \geq \delta - \alpha} \text{ (weakening)}$$

$$\frac{\alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq d \qquad k \in \mathbb{N} \qquad 0 < k \leq \alpha}{(\alpha - k)\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta - k} \text{ (partial weakening)}$$

Weakening

To preserve the conflict, the weakening rule must be used:

$$\frac{\alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta}{\sum_{i=1}^{n} \alpha_i \ell_i \geq \delta - \alpha} \text{ (weakening)}$$

$$\frac{\alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq d \qquad k \in \mathbb{N} \qquad 0 < k \leq \alpha}{(\alpha - k)\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta - k} \text{ (partial weakening)}$$

Weakening

To preserve the conflict, the weakening rule must be used:

$$\frac{\alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta}{\sum_{i=1}^{n} \alpha_i \ell_i \geq \delta - \alpha} \text{ (weakening)}$$

$$\frac{\alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq d \qquad k \in \mathbb{N} \qquad 0 < k \leq \alpha}{(\alpha - k)\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta - k} \text{ (partial weakening)}$$

$$5a + 5b + 3c \geq 8$$

Weakening

To preserve the conflict, the weakening rule must be used:

$$\frac{\alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta}{\sum_{i=1}^{n} \alpha_i \ell_i \geq \delta - \alpha} \text{ (weakening)}$$

$$\frac{\alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq d \qquad k \in \mathbb{N} \qquad 0 < k \leq \alpha}{(\alpha - k)\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta - k} \text{ (partial weakening)}$$

$$5a + 5b + 3c \geq 8$$

Weakening

To preserve the conflict, the weakening rule must be used:

$$\frac{\alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta}{\sum_{i=1}^{n} \alpha_i \ell_i \geq \delta - \alpha} \text{ (weakening)}$$

$$\frac{\alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq d \qquad k \in \mathbb{N} \qquad 0 < k \leq \alpha}{(\alpha - k)\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta - k} \text{ (partial weakening)}$$

$$5a + 5b \geq 8 - 3$$

Weakening

To preserve the conflict, the weakening rule must be used:

$$\frac{\alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta}{\sum_{i=1}^{n} \alpha_i \ell_i \geq \delta - \alpha} \text{ (weakening)}$$

$$\frac{\alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq d \qquad k \in \mathbb{N} \qquad 0 < k \leq \alpha}{(\alpha - k)\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta - k} \text{ (partial weakening)}$$

$$5a + 5b \geq 5$$

Weakening

To preserve the conflict, the weakening rule must be used:

$$\frac{\alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta}{\sum_{i=1}^{n} \alpha_i \ell_i \geq \delta - \alpha} \text{ (weakening)}$$

$$\frac{\alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq d \qquad k \in \mathbb{N} \qquad 0 < k \leq \alpha}{(\alpha - k)\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta - k} \text{ (partial weakening)}$$

Weakening

To preserve the conflict, the weakening rule must be used:

$$\frac{\alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta}{\sum_{i=1}^{n} \alpha_i \ell_i \geq \delta - \alpha} \text{ (weakening)}$$

$$\frac{\alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq d \qquad k \in \mathbb{N} \qquad 0 < k \leq \alpha}{(\alpha - k)\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta - k} \text{ (partial weakening)}$$

Weakening

To preserve the conflict, the weakening rule must be used:

$$\frac{\alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta}{\sum_{i=1}^{n} \alpha_i \ell_i \geq \delta - \alpha} \text{ (weakening)}$$

$$\frac{\alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq d \qquad k \in \mathbb{N} \qquad 0 < k \leq \alpha}{(\alpha - k)\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta - k} \text{ (partial weakening)}$$

$$5a + 5b + 3c \geq 8$$

Weakening

To preserve the conflict, the weakening rule must be used:

$$\frac{\alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta}{\sum_{i=1}^{n} \alpha_i \ell_i \geq \delta - \alpha} \text{ (weakening)}$$

$$\frac{\alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq d \qquad k \in \mathbb{N} \qquad 0 < k \leq \alpha}{(\alpha - k)\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta - k} \text{ (partial weakening)}$$

$$5a + 5b + 3c \geq 8$$

Weakening

To preserve the conflict, the weakening rule must be used:

$$\frac{\alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta}{\sum_{i=1}^{n} \alpha_i \ell_i \geq \delta - \alpha} \text{ (weakening)}$$

$$\frac{\alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq d \qquad k \in \mathbb{N} \qquad 0 < k \leq \alpha}{(\alpha - k)\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta - k} \text{ (partial weakening)}$$

$$5a + (5-2)b + 3c \geq 8 - 2$$

Weakening

To preserve the conflict, the weakening rule must be used:

$$\frac{\alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta}{\sum_{i=1}^{n} \alpha_i \ell_i \geq \delta - \alpha} \text{ (weakening)}$$

$$\frac{\alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq d \qquad k \in \mathbb{N} \qquad 0 < k \leq \alpha}{(\alpha - k)\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta - k} \text{ (partial weakening)}$$

$$5a + 3b + 3c \geq 6$$

Weakening

To preserve the conflict, the weakening rule must be used:

$$\frac{\alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta}{\sum_{i=1}^{n} \alpha_i \ell_i \geq \delta - \alpha} \text{ (weakening)}$$

$$\frac{\alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq d \qquad k \in \mathbb{N} \qquad 0 < k \leq \alpha}{(\alpha - k)\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta - k} \text{ (partial weakening)}$$

$$5a + 3b + 3c \geq 6$$

Weakening can be applied in many different ways (Le Berre et al., 2020b)

Different Weakening Strategies

The original approach (Dixon and Ginsberg, 2002; Chai and Kuehlmann, 2003) successively weakens away literals from the reason, until the saturation rule guarantees to derive a conflicting constraint

Different Weakening Strategies

The original approach (Dixon and Ginsberg, 2002; Chai and Kuehlmann, 2003) successively weakens away literals from the reason, until the saturation rule guarantees to derive a conflicting constraint

As the operation must be repeated multiple times,
its cost is not negligible

Different Weakening Strategies

The original approach (Dixon and Ginsberg, 2002; Chai and Kuehlmann, 2003) successively weakens away literals from the reason, until the saturation rule guarantees to derive a conflicting constraint

As the operation must be repeated multiple times,
its cost is not negligible

Another solution is to take advantage of the following property:

As soon as the coefficient of the literal to cancel is equal to 1 in at least one of the constraints, the derived constraint is guaranteed to be conflicting (Dixon, 2004)

Weakening Ineffective Literals

During conflict analysis, some literals may not play a role in the conflict being analyzed: it is thus possible to weaken them away while preserving invariants

Weakening Ineffective Literals

During conflict analysis, some literals may not play a role in the conflict being analyzed: it is thus possible to weaken them away while preserving invariants

$$\cfrac{\cfrac{3\bar{a} + 3\bar{b} + c + d + e \geq 6}{3\bar{b} + c \geq 6 - 3 - 1 - 1 = 1}}{\bar{b} + c \geq 1}$$

Weakening Ineffective Literals

During conflict analysis, some literals may not play a role in the conflict being analyzed: it is thus possible to weaken them away while preserving invariants

$$\cfrac{\cfrac{3\bar{a} + 3\bar{b} + c + d + e \geq 6}{3\bar{b} + c \geq 6 - 3 - 1 - 1 = 1}}{\bar{b} + c \geq 1}$$

Weakening Ineffective Literals

During conflict analysis, some literals may not play a role in the conflict being analyzed: it is thus possible to weaken them away while preserving invariants

$$\cfrac{\cfrac{\boxed{3\bar{a}} + 3\bar{b} + c + \boxed{d} + \boxed{e} \geq 6}{3\bar{b} + c \geq 6 - 3 - 1 - 1 = 1}}{\bar{b} + c \geq 1}$$

Weakening Ineffective Literals

During conflict analysis, some literals may not play a role in the conflict being analyzed: it is thus possible to weaken them away while preserving invariants

$$\frac{\dfrac{\boxed{3\bar{a}} + 3\bar{b} + c + \boxed{d} + \boxed{e} \geq 6}{3\bar{b} + c \geq 6 - 3 - 1 - 1 = 1}}{\bar{b} + c \geq 1}$$

Weakening Ineffective Literals

During conflict analysis, some literals may not play a role in the conflict being analyzed: it is thus possible to weaken them away while preserving invariants

$$\dfrac{\dfrac{\boxed{3\bar{a}} + 3\bar{b} + c + \boxed{d} + \boxed{e} \geq 6}{3\bar{b} + c \geq 6 - 3 - 1 - 1 = 1}}{\bar{b} + c \geq 1}$$

Weakening Ineffective Literals

During conflict analysis, some literals may not play a role in the conflict being analyzed: it is thus possible to weaken them away while preserving invariants

$$\dfrac{\dfrac{\boxed{3\bar{a}} + 3\bar{b} + c + \boxed{d} + \boxed{e} \geq 6}{3\bar{b} + c \geq 6 - 3 - 1 - 1 = 1}}{\bar{b} + c \geq 1}$$

$$\dfrac{\dfrac{2a + b + c + f \geq 2}{2a + b + f \geq 2 - 1 = 1}}{a + b + f \geq 1}$$

Weakening Ineffective Literals

During conflict analysis, some literals may not play a role in the conflict being analyzed: it is thus possible to weaken them away while preserving invariants

$$\frac{\dfrac{\boxed{3\bar{a}} + 3\bar{b} + c + \boxed{d} + \boxed{e} \geq 6}{3\bar{b} + c \geq 6 - 3 - 1 - 1 = 1}}{\bar{b} + c \geq 1}$$

$$\frac{\dfrac{2a + b + \boxed{c} + f \geq 2}{2a + b + f \geq 2 - 1 = 1}}{a + b + f \geq 1}$$

Weakening Ineffective Literals

During conflict analysis, some literals may not play a role in the conflict being analyzed: it is thus possible to weaken them away while preserving invariants

$$\dfrac{\dfrac{\boxed{3\bar{a}} + 3\bar{b} + c + \boxed{d} + \boxed{e} \geq 6}{3\bar{b} + c \geq 6 - 3 - 1 - 1 = 1}}{\bar{b} + c \geq 1}$$

$$\dfrac{\dfrac{2a + b + \boxed{c} + f \geq 2}{2a + b + f \geq 2 - 1 = 1}}{a + b + f \geq 1}$$

Weakening Ineffective Literals

During conflict analysis, some literals may not play a role in the conflict being analyzed: it is thus possible to weaken them away while preserving invariants

$$\cfrac{\cfrac{\boxed{3\bar{a}} + 3\bar{b} + c + \boxed{d} + \boxed{e} \geq 6}{3\bar{b} + c \geq 6 - 3 - 1 - 1 = 1}}{\bar{b} + c \geq 1}$$

$$\cfrac{\cfrac{2a + b + \boxed{c} + f \geq 2}{2a + b + f \geq 2 - 1 = 1}}{a + b + f \geq 1}$$

Weakening Ineffective Literals

During conflict analysis, some literals may not play a role in the conflict being analyzed: it is thus possible to weaken them away while preserving invariants

$$\dfrac{\dfrac{\boxed{3\bar{a}} + 3\bar{b} + c + \boxed{d} + \boxed{e} \geq 6}{3\bar{b} + c \geq 6 - 3 - 1 - 1 = 1}}{\bar{b} + c \geq 1}$$

$$\dfrac{\dfrac{2a + b + \boxed{c} + f \geq 2}{2a + b + f \geq 2 - 1 = 1}}{a + b + f \geq 1}$$

This strategy is equivalent to that used by solvers such as *SATIRE* (Whittemore and Sakallah, 2001) or *Sat4j-Resolution* to lazily infer clauses to use resolution based reasoning

Weakening and Division

In *RoundingSat* (Elffers and Nordström, 2018), the coefficient is rounded to one thanks to the division rule, applied after having weakened away some unfalsified literals

$$\frac{\sum_{i=1}^{n} \alpha_i \ell_i \geq \delta \qquad \rho \in \mathbb{N}^*}{\sum_{i=1}^{n} \lceil \frac{\alpha_i}{\rho} \rceil \ell_i \geq \lceil \frac{\delta}{\rho} \rceil} \text{(division)}$$

Weakening and Division

In *RoundingSat* (Elffers and Nordström, 2018), the coefficient is rounded to one thanks to the division rule, applied after having weakened away some unfalsified literals

$$\frac{\sum_{i=1}^{n} \alpha_i \ell_i \geq \delta \qquad \rho \in \mathbb{N}^*}{\sum_{i=1}^{n} \lceil \frac{\alpha_i}{\rho} \rceil \ell_i \geq \lceil \frac{\delta}{\rho} \rceil} \text{(division)}$$

Weakening and Division

In *RoundingSat* (Elffers and Nordström, 2018), the coefficient is rounded to one thanks to the division rule, applied after having weakened away some unfalsified literals

$$\frac{\sum_{i=1}^{n} \alpha_i \ell_i \geq \delta \qquad \rho \in \mathbb{N}^*}{\sum_{i=1}^{n} \lceil \frac{\alpha_i}{\rho} \rceil \ell_i \geq \lceil \frac{\delta}{\rho} \rceil} \text{(division)}$$

Weakening and Division

In *RoundingSat* (Elffers and Nordström, 2018), the coefficient is rounded to one thanks to the division rule, applied after having weakened away some unfalsified literals

$$\frac{\sum_{i=1}^{n} \alpha_i \ell_i \geq \delta \qquad \rho \in \mathbb{N}^*}{\sum_{i=1}^{n} \lceil \frac{\alpha_i}{\rho} \rceil \ell_i \geq \lceil \frac{\delta}{\rho} \rceil} \text{ (division)}$$

$$\cfrac{8a + 7b + 7c + 2d + 2e + f \geq 11}{\cfrac{7b + 7c + 2d + 2e \geq 2}{b + c + d + e \geq 1}}$$

Weakening and Division

In *RoundingSat* (Elffers and Nordström, 2018), the coefficient is rounded to one thanks to the division rule, applied after having weakened away some unfalsified literals

$$\frac{\sum_{i=1}^{n} \alpha_i \ell_i \geq \delta \qquad \rho \in \mathbb{N}^*}{\sum_{i=1}^{n} \lceil \frac{\alpha_i}{\rho} \rceil \ell_i \geq \lceil \frac{\delta}{\rho} \rceil} \text{ (division)}$$

$$\cfrac{8a + 7b + 7c + 2d + 2e + f \geq 11}{\cfrac{7b + 7c + 2d + 2e \geq 2}{b + c + d + e \geq 1}}$$

Weakening and Division

In *RoundingSat* (Elffers and Nordström, 2018), the coefficient is rounded to one thanks to the division rule, applied after having weakened away some unfalsified literals

$$\frac{\sum_{i=1}^{n} \alpha_i \ell_i \geq \delta \qquad \rho \in \mathbb{N}^*}{\sum_{i=1}^{n} \lceil \frac{\alpha_i}{\rho} \rceil \ell_i \geq \lceil \frac{\delta}{\rho} \rceil} \text{ (division)}$$

$$\cfrac{8a + 7b + 7c + 2d + 2e + f \geq 11}{\cfrac{7b + 7c + 2d + 2e \geq 2}{b + c + d + e \geq 1}}$$

Weakening and Division

In *RoundingSat* (Elffers and Nordström, 2018), the coefficient is rounded to one thanks to the division rule, applied after having weakened away some unfalsified literals

$$\frac{\sum_{i=1}^{n} \alpha_i \ell_i \geq \delta \qquad \rho \in \mathbb{N}^*}{\sum_{i=1}^{n} \lceil \frac{\alpha_i}{\rho} \rceil \ell_i \geq \lceil \frac{\delta}{\rho} \rceil} \text{ (division)}$$

$$\frac{\dfrac{8a + 7b + 7c + 2d + 2e + f \geq 11}{7b + 7c + 2d + 2e \geq 2}}{b + c + d + e \geq 1}$$

Weakening and Division

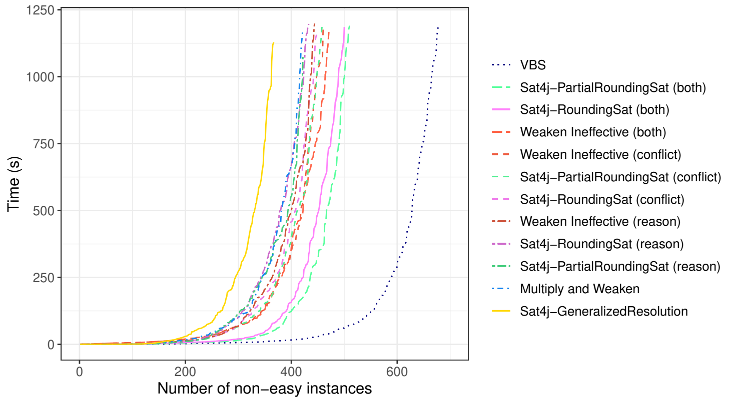
In *RoundingSat* (Elffers and Nordström, 2018), the coefficient is rounded to one thanks to the division rule, applied after having weakened away some unfalsified literals

$$\frac{\sum_{i=1}^{n} \alpha_i \ell_i \geq \delta \qquad \rho \in \mathbb{N}^*}{\sum_{i=1}^{n} \left\lceil \frac{\alpha_i}{\rho} \right\rceil \ell_i \geq \left\lceil \frac{\delta}{\rho} \right\rceil} \text{ (division)}$$

$$\cfrac{8a + 7b + 7c + 2d + 2e + f \geq 11}{\cfrac{7b + 7c + 2d + 2e \geq 2}{b + c + d + e \geq 1}}$$

It is also possible to apply partial weakening before division to infer stronger constraints

Many Different Strategies



An Achilles Heel in the Cutting Planes Proof System

Irrelevant Literals (Le Berre et al., 2020a)

Cutting planes rules may introduce irrelevant literals

$$\frac{3d + a + b + c \geq 3 \qquad 3\bar{d} + 2a + 2b \geq 3}{3a + 3b + c \geq 3}$$

Irrelevant Literals (Le Berre et al., 2020a)

Cutting planes rules may introduce irrelevant literals

$$\frac{3d + a + b + c \geq 3 \qquad 3\bar{d} + 2a + 2b \geq 3}{3a + 3b + c \geq 3}$$

Irrelevant Literals (Le Berre et al., 2020a)

Cutting planes rules may introduce irrelevant literals

$$\frac{3d + a + b + c \geq 3 \qquad 3\bar{d} + 2a + 2b \geq 3}{3a + 3b + c \geq 3}$$

Irrelevant Literals (Le Berre et al., 2020a)

Cutting planes rules may introduce irrelevant literals

$$\frac{3d + a + b + c \geq 3 \qquad 3\bar{d} + 2a + 2b \geq 3}{3a + 3b + c \geq 3}$$

A literal is said to be irrelevant in a PB constraint when its truth value does not impact the truth value of the constraint: irrelevant literals can thus be removed

Irrelevant Literals (Le Berre et al., 2020a)

Cutting planes rules may introduce irrelevant literals

$$\frac{3d + a + b + c \geq 3 \qquad 3\bar{d} + 2a + 2b \geq 3}{3a + 3b + \cancel{c} \geq 3}$$

A literal is said to be irrelevant in a PB constraint when its truth value does not impact the truth value of the constraint: irrelevant literals can thus be removed

Production of Irrelevant Literals

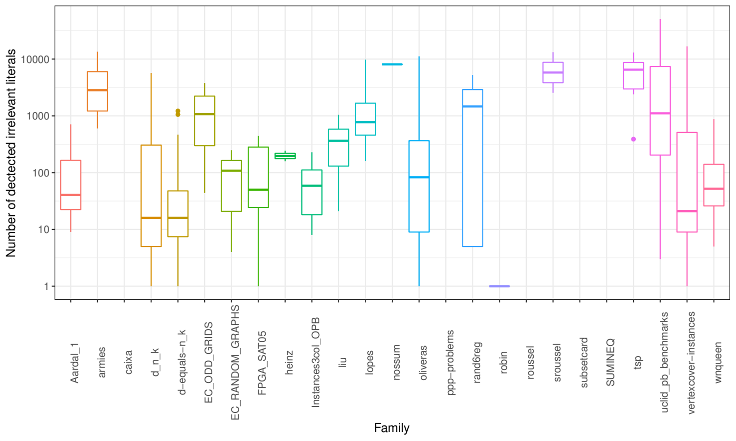


Figure 3: Statistics about the production of irrelevant literals in *Sat4j-GeneralizedResolution* for each family of benchmarks (logarithmic scale)

Artificially Relevant Literals

Irrelevant literals may become artificially relevant, in which case they may impact the strength of the derived constraints

$$\dfrac{\dfrac{3a + 3b + c \geq 3 \qquad 3\bar{a} + 3d + 2c \geq 3}{3b + 3c + 3d \geq 3}}{b + c + d \geq 1}$$

Artificially Relevant Literals

Irrelevant literals may become artificially relevant, in which case they may impact the strength of the derived constraints

$$\dfrac{\dfrac{3a + 3b + c \geq 3 \qquad 3\bar{a} + 3d + 2c \geq 3}{3b + 3c + 3d \geq 3}}{b + c + d \geq 1}$$

Artificially Relevant Literals

Irrelevant literals may become artificially relevant, in which case they may impact the strength of the derived constraints

$$\dfrac{\dfrac{3a + 3b + c \geq 3 \qquad 3\bar{a} + 3d + 2c \geq 3}{3b + 3c + 3d \geq 3}}{b + c + d \geq 1}$$

Artificially Relevant Literals

Irrelevant literals may become artificially relevant, in which case they may impact the strength of the derived constraints

$$\dfrac{\dfrac{3a + 3b + \cancel{c} \geq 3 \qquad 3\bar{a} + 3d + 2c \geq 3}{3b + 3c + 3d \geq 3}}{b + c + d \geq 1}$$

Artificially Relevant Literals

Irrelevant literals may become artificially relevant, in which case they may impact the strength of the derived constraints

$$\dfrac{\dfrac{3a + 3b + \cancel{c} \geq 3 \qquad 3\bar{a} + 3d + \cancel{2c} \geq 3}{3b + 3c + 3d \geq 3}}{b + c + d \geq 1}$$

Artificially Relevant Literals

Irrelevant literals may become artificially relevant, in which case they may impact the strength of the derived constraints

$$\cfrac{\cfrac{3a + 3b + \cancel{c} \geq 3 \qquad 3\bar{a} + 3d + \cancel{2c} \geq 3}{3b + \cancel{3c} + 3d \geq 3}}{b + c + d \geq 1}$$

Artificially Relevant Literals

Irrelevant literals may become artificially relevant, in which case they may impact the strength of the derived constraints

$$\dfrac{\dfrac{3a + 3b + \cancel{c} \geq 3 \qquad 3\bar{a} + 3d + \cancel{2c} \geq 3}{3b + \cancel{3c} + 3d \geq 3}}{b + \cancel{c} + d \geq 1}$$

Artificially Relevant Literals

Irrelevant literals may become artificially relevant, in which case they may impact the strength of the derived constraints

$$\dfrac{\dfrac{3a + 3b + \cancel{c} \geq 3 \qquad 3\bar{a} + 3d + \cancel{2c} \geq 3}{3b + \cancel{3c} + 3d \geq 3}}{b + \cancel{c} + d \geq 1}$$

Detecting irrelevant literals is NP-hard, we thus introduce an incomplete algorithm for removing them

Detecting Irrelevant Literals (1)

Irrelevant literals can be detected thanks to this reduction to subset-sum

$$\ell \text{ is irrelevant in } \alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta$$

$$\Leftrightarrow \sum_{i=1}^{n} \alpha_i \ell_i = \delta - k \text{ has no solution for } k \in \{1, \ldots, \alpha\}$$

Detecting Irrelevant Literals (1)

Irrelevant literals can be detected thanks to this reduction to subset-sum

$$\ell \text{ is irrelevant in } \alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta$$

$$\Leftrightarrow \sum_{i=1}^{n} \alpha_i \ell_i = \delta - k \text{ has no solution for } k \in \{1, \ldots, \alpha\}$$

For instance, c is irrelevant in $3a + 3b + 2c \geq 3$ because there is no solution for neither of the equalities

$$3a + 3b = 1 \text{ and } 3a + 3b = 2$$

Detecting Irrelevant Literals (1)

Irrelevant literals can be detected thanks to this reduction to subset-sum

$$\ell \text{ is irrelevant in } \alpha\ell + \sum_{i=1}^{n} \alpha_i \ell_i \geq \delta$$

$$\Leftrightarrow \sum_{i=1}^{n} \alpha_i \ell_i = \delta - k \text{ has no solution for } k \in \{1, \ldots, \alpha\}$$

For instance, c is irrelevant in $3a + 3b + 2c \geq 3$ because there is no solution for neither of the equalities

$$3a + 3b = 1 \text{ and } 3a + 3b = 2$$

A dynamic programming algorithm can decide whether any of the equalities has a solution in pseudo-polynomial time with a single run

Detecting Irrelevant Literals (2)

As coefficients and degrees may be very big in the derived PB constraints, solving subset-sum on the corresponding instances would be very inefficient

Detecting Irrelevant Literals (2)

As coefficients and degrees may be very big in the derived PB constraints, solving subset-sum on the corresponding instances would be very inefficient

We thus consider an incomplete approach for solving these instances

Detecting Irrelevant Literals (2)

As coefficients and degrees may be very big in the derived PB constraints, solving subset-sum on the corresponding instances would be very inefficient

We thus consider an incomplete approach for solving these instances

In our case, we want our algorithm to be exact when it detects that the instance has no solution, since the literal is irrelevant in this case (said differently, we accept to miss irrelevant literals, but not the contrary)

Detecting Irrelevant Literals (2)

As coefficients and degrees may be very big in the derived PB constraints, solving subset-sum on the corresponding instances would be very inefficient

We thus consider an incomplete approach for solving these instances

In our case, we want our algorithm to be exact when it detects that the instance has no solution, since the literal is irrelevant in this case (said differently, we accept to miss irrelevant literals, but not the contrary)

Our algorithm solves subset-sum modulo a fixed number, or even several numbers

Removing Irrelevant Literals

We can remove any irrelevant literal while preserving equivalence, by taking advantage that their truth value does not affect the constraint

Removing Irrelevant Literals

We can remove any irrelevant literal while preserving equivalence, by taking advantage that their truth value does not affect the constraint

$$3a + 3b + 2c \geq 3$$

Removing Irrelevant Literals

We can remove any irrelevant literal while preserving equivalence, by taking advantage that their truth value does not affect the constraint

$$3a + 3b + 2c \geq 3$$

First, we can locally assign the literal to 0, and simply remove it:

$$3a + 3b \geq 3$$

Removing Irrelevant Literals

We can remove any irrelevant literal while preserving equivalence, by taking advantage that their truth value does not affect the constraint

$$3a + 3b + 2c \geq 3$$

First, we can locally assign the literal to 0, and simply remove it:

$$3a + 3b \geq 3$$

Or, we can locally assign it to 1, and simplify the constraint accordingly:

$$3a + 3b \geq 3 - 2 = 1$$

Removing Irrelevant Literals

We can remove any irrelevant literal while preserving equivalence, by taking advantage that their truth value does not affect the constraint

$$3a + 3b + 2c \geq 3$$

First, we can locally assign the literal to 0, and simply remove it:

$$3a + 3b \geq 3$$

Or, we can locally assign it to 1, and simplify the constraint accordingly:

$$3a + 3b \geq 3 - 2 = 1$$

In practice, we use a heuristic to decide which strategy to apply, as none of them is better than the other

Impact of the Removal of Irrelevant Literals on the Proof

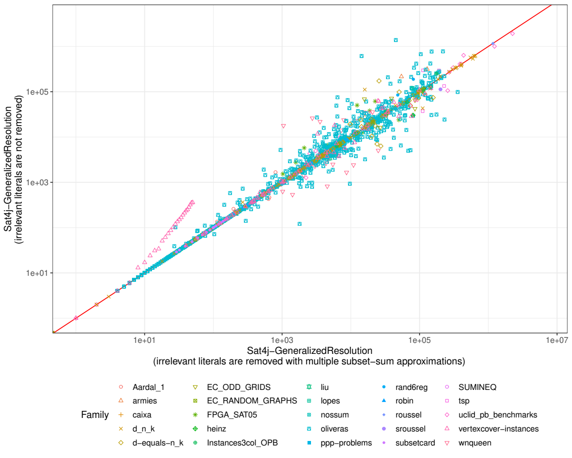


Figure 4: Comparison of the size of the proofs (number of cancellations) built by *Sat4j-GeneralizedResolution* with and without the removal of irrelevant literals on all benchmarks (logarithmic scale)

Focus on the Vertex-Cover Family: Experimental Results

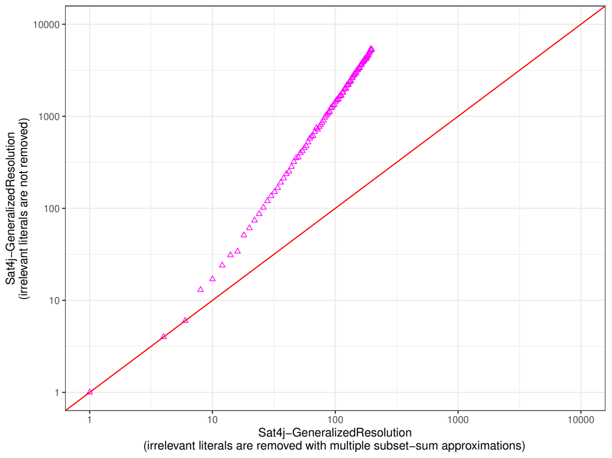


Figure 5: Comparison of the size of the proofs (number of cancellations) built by *Sat4j-GeneralizedResolution* with and without the removal of irrelevant literals on vertex-cover instances (logarithmic scale)

Focus on the Vertex-Cover Family: *Sat4j*'s Behavior

When given an instance of this family, the first constraint learned by *Sat4j* has the form

$$nx + x_1 + \ldots + x_{n-1} \geq n$$

Focus on the Vertex-Cover Family: *Sat4j*'s Behavior

When given an instance of this family, the first constraint learned by *Sat4j* has the form

$$nx + x_1 + \ldots + x_{n-1} \geq n$$

All the literals $x_1, \ldots, x_{n-1}$ are irrelevant, and this constraint is actually equivalent to the unit clause x

Focus on the Vertex-Cover Family: *Sat4j*'s Behavior

When given an instance of this family, the first constraint learned by *Sat4j* has the form

$$nx + x_1 + \ldots + x_{n-1} \geq n$$

All the literals $x_1, \ldots, x_{n-1}$ are irrelevant, and this constraint is actually equivalent to the unit clause x

No other irrelevant literals are detected in the other constraints derived by *Sat4j*

Focus on the Vertex-Cover Family: ***Sat4j*'s Behavior**

When given an instance of this family, the first constraint learned by *Sat4j* has the form

$$nx + x_1 + \ldots + x_{n-1} \geq n$$

All the literals $x_1, \ldots, x_{n-1}$ are irrelevant, and this constraint is actually equivalent to the unit clause x

No other irrelevant literals are detected in the other constraints derived by *Sat4j*

Even few irrelevant literals can lead to the production of an exponentially larger proof

Impact of the Removal of Irrelevant Literals on the Runtime

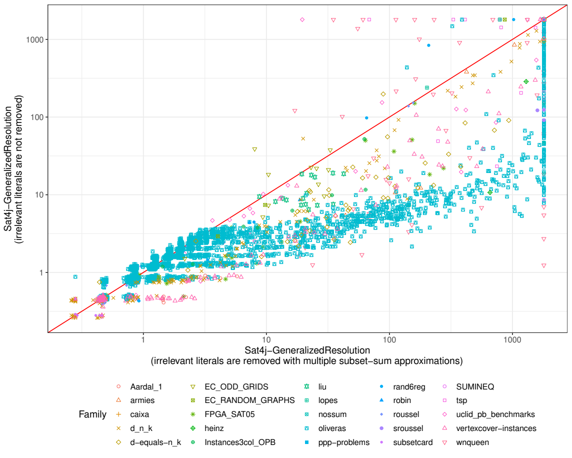


Figure 6: Comparison of the runtime of *Sat4j-GeneralizedResolution* with and without the removal of irrelevant literals on all benchmarks (logarithmic scale)

Weakening Ineffective Literals

Recall that, during conflict analysis, some literals may be ineffective

Weakening Ineffective Literals

Recall that, during conflict analysis, some literals may be ineffective

$$\boxed{3\bar{a}} + 3\bar{b} + c + \boxed{d} + \boxed{e} \geq 6$$

$$2a + b + \boxed{c} + f \geq 2$$

Weakening Ineffective Literals

Recall that, during conflict analysis, some literals may be ineffective

$$\boxed{3\bar{a}} + 3\bar{b} + c + \boxed{d} + \boxed{e} \geq 6$$

$$2a + b + \boxed{c} + f \geq 2$$

Ineffective literals can be seen as locally irrelevant, as opposed to the globally irrelevant literals presented before

Weakening Ineffective Literals

Recall that, during conflict analysis, some literals may be ineffective

$$\boxed{3\bar{a}} + 3\bar{b} + c + \boxed{d} + \boxed{e} \geq 6$$

$$2a + b + \boxed{c} + f \geq 2$$

Ineffective literals can be seen as locally irrelevant, as opposed to the globally irrelevant literals presented before

In the context of the current partial assignment, it is easy to detect ineffective literals, but they can only be weakened away (as ineffective literals may be relevant)

Adapting further PB Solvers to CDCL

CDCL Architecture Recap

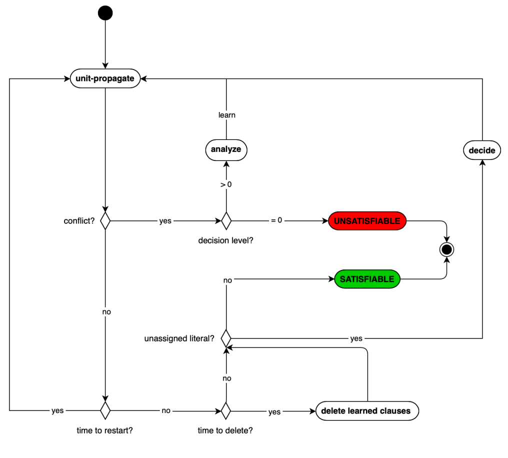


Figure 7: Overview of the CDCL Algorithm

CDCL Architecture Recap

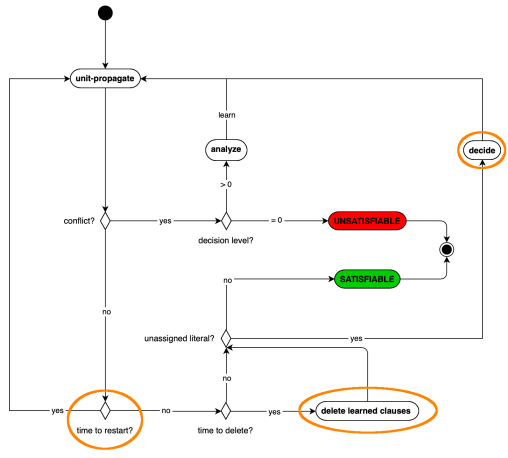


Figure 8: Use of other strategies in the CDCL Algorithm

Motivation

It is well known that, in addition to conflict analysis, several features of SAT solvers are crucial for solving problems efficiently, such as:

- branching heuristic
- learned constraint deletion strategy
- restart policy

Motivation

It is well known that, in addition to conflict analysis, several features of SAT solvers are crucial for solving problems efficiently, such as:

- branching heuristic
- learned constraint deletion strategy
- restart policy

These features are mostly reused as is by current PB solvers, without taking into account the particular properties of PB constraints

Motivation

It is well known that, in addition to conflict analysis, several features of SAT solvers are crucial for solving problems efficiently, such as:

- branching heuristic
- learned constraint deletion strategy
- restart policy

These features are mostly reused as is by current PB solvers, without taking into account the particular properties of PB constraints

Our main finding (Le Berre and Wallon, 2021) is that considering the size of the coefficients and the current partial assignment allows to significantly improve the solver

Comparison of different variants (decision)

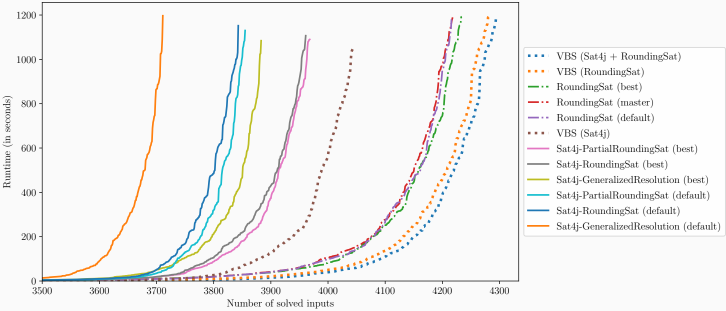


Figure 9: Cactus plot of the best configurations of different solvers

Comparison of different variants (optimization)

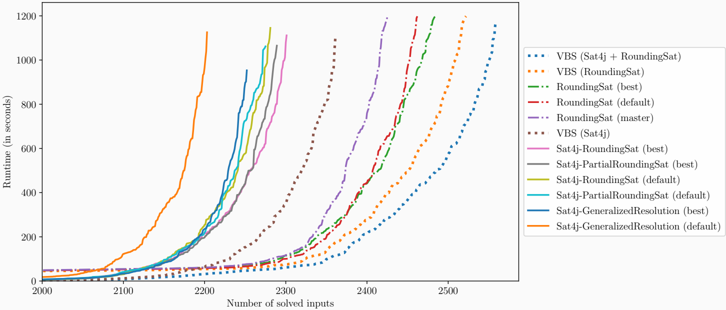


Figure 10: Cactus plot of the best configurations of different solvers

Conclusion and Perspectives

Conclusion

- Implementations of the cutting planes proof system in PB solvers are not fully satisfactory, as its strength is not fully exploited
- Irrelevant literals may be produced during conflict analysis, and lead to the inference of weaker constraints
- Applying the weakening rule on ineffective literals is a possible (aggressive) counter-measure
- Applying partial weakening and division gives better performance

Conclusion

- Implementations of the cutting planes proof system in PB solvers are not fully satisfactory, as its strength is not fully exploited
- Irrelevant literals may be produced during conflict analysis, and lead to the inference of weaker constraints
- Applying the weakening rule on ineffective literals is a possible (aggressive) counter-measure
- Applying partial weakening and division gives better performance

- Complementary heuristics implemented in CDCL PB solvers can be adapted to take into account properties of PB constraints and to improve the performance of *Sat4j*

Perspectives

- Find other strategies for applying cutting planes rules so as to exploit more power of this proof system
- Design such strategies so as to prevent the production of irrelevant literals instead of removing them

Perspectives

- Find other strategies for applying cutting planes rules so as to exploit more power of this proof system
- Design such strategies so as to prevent the production of irrelevant literals instead of removing them

- Combine the strategies to exploit their complementarity (e.g., using DAC)
- Identify possible interactions between the new heuristics

Perspectives

- Find other strategies for applying cutting planes rules so as to exploit more power of this proof system
- Design such strategies so as to prevent the production of irrelevant literals instead of removing them

- Combine the strategies to exploit their complementarity (e.g., using DAC)
- Identify possible interactions between the new heuristics

- Improve the detection of the optimal backjump level during conflict analysis

Perspectives

- Find other strategies for applying cutting planes rules so as to exploit more power of this proof system
- Design such strategies so as to prevent the production of irrelevant literals instead of removing them

- Combine the strategies to exploit their complementarity (e.g., using DAC)
- Identify possible interactions between the new heuristics

- Improve the detection of the optimal backjump level during conflict analysis

- Improve the detection of conflicts to deal with the conflictual reasons encountered during conflict analysis

Deep Dive into CDCL Pseudo-Boolean Solvers

Romain Wallon

May 20th, 2021

Laboratoire d'Informatique de l'X (LIX), École Polytechnique

Adapting further PB Solvers

Leveraging Properties of PB Constraints for Fine Tuning Sat4j

Let us consider again a confict analysis

$$3\bar{a}(?@?) + 3\bar{f}(?@?) + d(?@?) + e(?@?) \geq 5$$

$$6a(?@?) + 3b(?@?) + 3c(?@?) + 3d(?@?) + 3f(?@?) \geq 9$$

Leveraging Properties of PB Constraints for Fine Tuning Sat4j

Let us consider again a confict analysis

$$3\bar{a}(?@?) + 3\bar{f}(?@?) + d(?@?) + e(?@?) \geq 5$$

$$6a(?@?) + 3b(1@1) + 3c(?@?) + 3d(?@?) + 3f(?@?) \geq 9$$

Leveraging Properties of PB Constraints for Fine Tuning Sat4j

Let us consider again a confict analysis

$$3\bar{a}(?@?) + 3\bar{f}(?@?) + d(?@?) + e(?@?) \geq 5$$

$$6a(?@?) + 3b(1@1) + 3c(0@2) + 3d(?@?) + 3f(?@?) \geq 9$$

Leveraging Properties of PB Constraints for Fine Tuning Sat4j

Let us consider again a confict analysis

$$3\bar{a}(?@?) + 3\bar{f}(?@?) + \color{red}{d(0@3)} + e(?@?) \geq 5$$

$$6a(?@?) + \color{green}{3b(1@1)} + \color{red}{3c(0@2)} + 3d(?@?) + 3f(?@?) \geq 9$$

Let us consider again a confict analysis

$$3\bar{a}(1@3) + 3\bar{f}(1@3) + d(0@3) + e(?@?) \geq 5$$

$$6a(0@3) + 3b(1@1) + 3c(0@2) + 3d(?@?) + 3f(0@3) \geq 9$$

Let us consider again a confict analysis

$$3\bar{a}(1@3) + 3\bar{f}(1@3) + d(0@3) + e(?@?) \geq 5$$

$$6a(0@3) + 3b(1@1) + 3c(0@2) + 3d(?@?) + 3f(0@3) \geq 9$$

We now apply the cancellation rule between these two constraints:

$$\frac{3\bar{a} + 3\bar{f} + d + e \geq 5 \qquad 6a + 3b + 3c + 3d + 3f \geq 9}{3a(0@3) + 3b(1@1) + 3c(0@2) + 2\bar{d}(1@3) + e(?@?) \geq 7}$$

Leveraging Properties of PB Constraints for Fine Tuning Sat4j

Let us consider again a confict analysis

$$3\bar{a}(1@3) + 3\bar{f}(1@3) + d(0@3) + e(?@?) \geq 5$$

$$6a(0@3) + 3b(1@1) + 3c(0@2) + 3d(?@?) + 3f(0@3) \geq 9$$

We now apply the cancellation rule between these two constraints:

$$\frac{3\bar{a} + 3\bar{f} + d + e \geq 5 \qquad 6a + 3b + 3c + 3d + 3f \geq 9}{3a(?@?) + 3b(1@1) + 3c(0@2) + 2\bar{d}(?@?) + e(?@?) \geq 7}$$

Let us consider again a confict analysis

$$3\bar{a}(1@3) + 3\bar{f}(1@3) + d(0@3) + e(?@?) \geq 5$$

$$6a(0@3) + 3b(1@1) + 3c(0@2) + 3d(?@?) + 3f(0@3) \geq 9$$

We now apply the cancellation rule between these two constraints:

$$\frac{3\bar{a} + 3\bar{f} + d + e \geq 5 \qquad 6a + 3b + 3c + 3d + 3f \geq 9}{3a(?@?) + 3b(1@1) + 3c(0@2) + 2\bar{d}(?@?) + e(?@?) \geq 7}$$

The PB constraints involved in this conflict analysis have very different properties compared to clauses!

(E)VSIDS for Making Decisions: Classical Implementation

All variables encountered during conflict analysis are bumped

(E)VSIDS for Making Decisions: Classical Implementation

All variables encountered during conflict analysis are bumped

This is the case for all the variables appearing in the previous reason:

$$3\bar{a} + 3\bar{f} + d + e \geq 5$$

(E)VSIDS for Making Decisions: Classical Implementation

All variables encountered during conflict analysis are bumped

This is the case for all the variables appearing in the previous reason:

$$3\bar{a} + 3\bar{f} + d + e \geq 5$$

This means that the scores of the variables a, f, d and e are incremented

(E)VSIDS for Making Decisions: Coefficients

A first approach for adapting VSIDS to PB constraints has been proposed in (Dixon, 2004), but it only takes into account the original cardinality constraints, and thus not the reason we have here:

$$3\bar{a} + 3\bar{f} + d + e \geq 5$$

(E)VSIDS for Making Decisions: Coefficients

A first approach for adapting VSIDS to PB constraints has been proposed in (Dixon, 2004), but it only takes into account the original cardinality constraints, and thus not the reason we have here:

$$3\bar{a} + 3\bar{f} + d + e \geq 5$$

We propose to take these coefficients into account with 3 new strategies:

(E)VSIDS for Making Decisions: Coefficients

A first approach for adapting VSIDS to PB constraints has been proposed in (Dixon, 2004), but it only takes into account the original cardinality constraints, and thus not the reason we have here:

$$3\bar{a} + 3\bar{f} + d + e \geq 5$$

We propose to take these coefficients into account with 3 new strategies:

- `bump-degree`: the score of each variable is incremented by the degree of the constraint (5 for all variables)

(E)VSIDS for Making Decisions: Coefficients

A first approach for adapting VSIDS to PB constraints has been proposed in (Dixon, 2004), but it only takes into account the original cardinality constraints, and thus not the reason we have here:

$$3\bar{a} + 3\bar{f} + d + e \geq 5$$

We propose to take these coefficients into account with 3 new strategies:

- `bump-degree`: the score of each variable is incremented by the degree of the constraint (5 for all variables)
- `bump-coefficient`: the score of each variable is incremented by their coefficients in the constraint (3 for a and f)

(E)VSIDS for Making Decisions: Coefficients

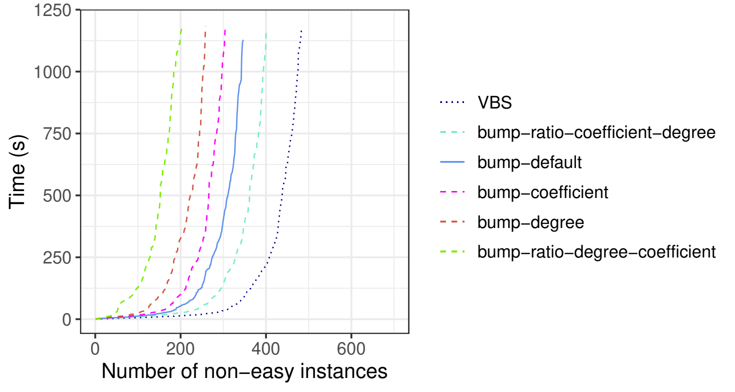
A first approach for adapting VSIDS to PB constraints has been proposed in (Dixon, 2004), but it only takes into account the original cardinality constraints, and thus not the reason we have here:

$$3\bar{a} + 3\bar{f} + d + e \geq 5$$

We propose to take these coefficients into account with 3 new strategies:

- `bump-degree`: the score of each variable is incremented by the degree of the constraint (5 for all variables)
- `bump-coefficient`: the score of each variable is incremented by their coefficients in the constraint (3 for a and f)
- `bump-ratio`: the score of each variable is incremented by the ratio of the degree by their coefficient in the constraint ($\frac{5}{3}$ for a and f)

(E)VSIDS for Making Decisions: Experiments



(E)VSIDS for Making Decisions: Assignments

Observe also that some literals are unassigned in the reason:

$$3\bar{a} + 3\bar{f} + d + e \geq 5$$

(E)VSIDS for Making Decisions: Assignments

Observe also that some literals are unassigned in the reason:

$$3\bar{a} + 3\bar{f} + d + e \geq 5$$

In an assertive clause, all literals are assigned, and all but one are falsified: these latter literals are those involved in the propagation

(E)VSIDS for Making Decisions: Assignments

Observe also that some literals are unassigned in the reason:

$$3\bar{a} + 3\bar{f} + d + e \geq 5$$

In an assertive clause, all literals are assigned, and all but one are falsified: these latter literals are those involved in the propagation

We can take the current assignement into account with 3 new strategies:

(E)VSIDS for Making Decisions: Assignments

Observe also that some literals are unassigned in the reason:

$$3\bar{a} + 3\bar{f} + d + e \geq 5$$

> *In an assertive clause, all literals are assigned, and all but one are falsified: these latter literals are those involved in the propagation*

We can take the current assignement into account with 3 new strategies:

- `bump-assigned`: the score of each assigned variable is incremented (a, f and d)

(E)VSIDS for Making Decisions: Assignments

Observe also that some literals are unassigned in the reason:

$$3\bar{a} + 3\bar{f} + d + e \geq 5$$

In an assertive clause, all literals are assigned, and all but one are falsified: these latter literals are those involved in the propagation

We can take the current assignement into account with 3 new strategies:

- `bump-assigned`: the score of each assigned variable is incremented (a, f and d)
- `bump-falsified`: the score of each falsified variable is incremented (f and d)

(E)VSIDS for Making Decisions: Assignments

Observe also that some literals are unassigned in the reason:

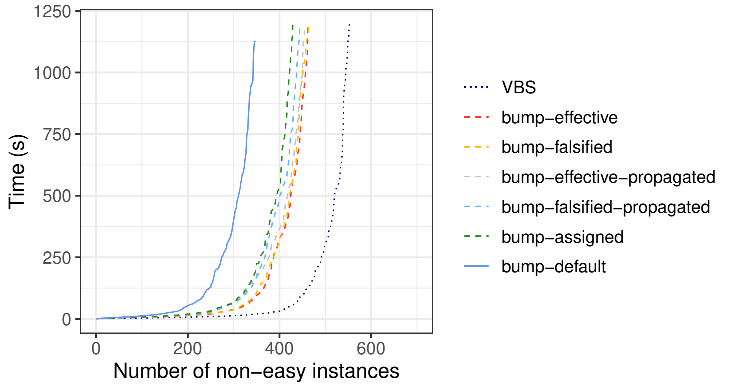
$$3\bar{a} + 3\bar{f} + d + e \geq 5$$

In an assertive clause, all literals are assigned, and all but one are falsified: these latter literals are those involved in the propagation

We can take the current assignement into account with 3 new strategies:

- `bump-assigned`: the score of each assigned variable is incremented (a, f and d)
- `bump-falsified`: the score of each falsified variable is incremented (f and d)
- `bump-effective`: the score of each effective variable is incremented (f and d)

(E)VSIDS for Making Decisions: Experiments



Quality of Learned Constraints: Classical Implementations

In SAT solvers, evaluating the quality of learned constraints is used to choose which constraints should be deleted and to decide when a restart should be triggered

Quality of Learned Constraints: Classical Implementations

In SAT solvers, evaluating the quality of learned constraints is used to choose which constraints should be deleted and to decide when a restart should be triggered

The quality measures used by SAT solvers do not take into account the properties of PB constraints

Quality of Learned Constraints: Size and Coefficients

In SAT solvers, the size of a clause is a naive measure of its quality: the longer the clause, the lower its strength

Quality of Learned Constraints: Size and Coefficients

In SAT solvers, the size of a clause is a naive measure of its quality: the longer the clause, the lower its strength

In the PB case, the length of a constraint does not reflect its strength

Quality of Learned Constraints: Size and Coefficients

In SAT solvers, the size of a clause is a naive measure of its quality: the longer the clause, the lower its strength

In the PB case, the length of a constraint does not reflect its strength

However, the size of a PB constraint also takes into account its coefficients

Quality of Learned Constraints: Size and Coefficients

In SAT solvers, the size of a clause is a naive measure of its quality: the longer the clause, the lower its strength

In the PB case, the length of a constraint does not reflect its strength

However, the size of a PB constraint also takes into account its coefficients

Consider the constraint we derived in the previous conflict analysis:

$$3a + 3b + 3c + 2\bar{d} + e \geq 7$$

Quality of Learned Constraints: Size and Coefficients

In SAT solvers, the size of a clause is a naive measure of its quality: the longer the clause, the lower its strength

In the PB case, the length of a constraint does not reflect its strength

However, the size of a PB constraint also takes into account its coefficients

Consider the constraint we derived in the previous conflict analysis:

$$3a + 3b + 3c + 2\bar{d} + e \geq 7$$

In practice, the coefficients may become very big, which requires the use of arbitrary precision encodings and slows down arithmetic operations.

Quality of Learned Constraints: Size and Coefficients

In SAT solvers, the size of a clause is a naive measure of its quality: the longer the clause, the lower its strength

In the PB case, the length of a constraint does not reflect its strength

However, the size of a PB constraint also takes into account its coefficients

Consider the constraint we derived in the previous conflict analysis:

$$3a + 3b + 3c + 2\bar{d} + e \geq 7$$

In practice, the coefficients may become very big, which requires the use of arbitrary precision encodings and slows down arithmetic operations.

We consider quality measures based on the value and size of the degree of the constraints: the lower the degree, the better the constraint

Quality of Learned Constraints: Size and Coefficients

In SAT solvers, the size of a clause is a naive measure of its quality: the longer the clause, the lower its strength

In the PB case, the length of a constraint does not reflect its strength

However, the size of a PB constraint also takes into account its coefficients

Consider the constraint we derived in the previous conflict analysis:

$$3a + 3b + 3c + 2\bar{d} + e \geq 7$$

In practice, the coefficients may become very big, which requires the use of arbitrary precision encodings and slows down arithmetic operations.

We consider quality measures based on the value and size of the degree of the constraints: the lower the degree, the better the constraint

Quality of Learned Constraints: Assignments (LBD)

In SAT solvers, the Literal Block Distance (Audemard and Simon, 2009) measures the quality of clauses by the number of decision levels appearing in this clause

Quality of Learned Constraints: Assignments (LBD)

In SAT solvers, the Literal Block Distance (Audemard and Simon, 2009) measures the quality of clauses by the number of decision levels appearing in this clause

$$3a(0@3) + 3b(1@1) + 3c(0@2) + 2\bar{d}(1@3) + e(?@?) \geq 7$$

Quality of Learned Constraints: Assignments (LBD)

In SAT solvers, the Literal Block Distance (Audemard and Simon, 2009) measures the quality of clauses by the number of decision levels appearing in this clause

$$3a(0@3) + 3b(1@1) + 3c(0@2) + 2\bar{d}(1@3) + e(?@?) \geq 7$$

There are satisfied and unassigned literals in this constraint!

Quality of Learned Constraints: Assignments (LBD)

In SAT solvers, the Literal Block Distance (Audemard and Simon, 2009) measures the quality of clauses by the number of decision levels appearing in this clause

$$3a(0@3) + 3b(1@1) + 3c(0@2) + 2\bar{d}(1@3) + e(?@?) \geq 7$$

There are satisfied and unassigned literals in this constraint!

We thus introduce 5 new definitions of LBD:

- `lbd-a`: the LBD is computed over assigned literals only
- `lbd-s`: the LBD is computed over assigned literals, and unassigned literals are considered assigned at the same (dummy) decision level
- `lbd-d`: the LBD is computed over assigned literals, and unassigned literals are considered assigned at different (dummy) decision levels
- `lbd-f`: the LBD is computed over falsified literals only
- `lbd-e`: the LBD is computed over effective literals only

Quality of Learned Constraint: Deletion

Deleting constraints is required by SAT solvers to limit the memory usage and to prevent unit propagation from slowing down

Quality of Learned Constraint: Deletion

Deleting constraints is required by SAT solvers to limit the memory usage and to prevent unit propagation from slowing down

This is also true, but to a lesser extent, for PB solvers

Quality of Learned Constraint: Deletion

Deleting constraints is required by SAT solvers to limit the memory usage and to prevent unit propagation from slowing down

This is also true, but to a lesser extent, for PB solvers

> *The constraints to delete are those having a bad score w.r.t. the quality measure used in the solver*

Quality of Learned Constraint: Deletion

Deleting constraints is required by SAT solvers to limit the memory usage and to prevent unit propagation from slowing down

This is also true, but to a lesser extent, for PB solvers

The constraints to delete are those having a bad score w.r.t. the quality measure used in the solver

We thus introduce the following deletion strategies, based on the different quality measures we presented:

- `delete-degree`
- `delete-degree-size`
- `delete-lbd-a`
- `delete-lbd-s`
- `delete-lbd-d`
- `delete-lbd-f`
- `delete-lbd-e`

Quality of Learned Constraints: Restarts

Restarting allows to forget all decisions made by the solver, so as to avoid being stuck in a subpart of the search space

Quality of Learned Constraints: Restarts

Restarting allows to forget all decisions made by the solver, so as to avoid being stuck in a subpart of the search space

Following *Glucose*'s approach (Audemard and Simon, 2012), we consider adaptive restarts based on the quality of recently learned constraints

Quality of Learned Constraints: Restarts

Restarting allows to forget all decisions made by the solver, so as to avoid being stuck in a subpart of the search space

Following *Glucose*'s approach (Audemard and Simon, 2012), we consider adaptive restarts based on the quality of recently learned constraints

Whenever the most recent constraints are of poor quality compared to all the others, a restart is performed

Quality of Learned Constraints: Restarts

Restarting allows to forget all decisions made by the solver, so as to avoid being stuck in a subpart of the search space

Following *Glucose*'s approach (Audemard and Simon, 2012), we consider adaptive restarts based on the quality of recently learned constraints

Whenever the most recent constraints are of poor quality compared to all the others, a restart is performed

We thus introduce the following restart strategies, based on the different quality measures we presented

- `restart-degree`
- `restart-degree-size`
- `restart-lbd-a`
- `restart-lbd-s`
- `restart-lbd-d`
- `restart-lbd-f`
- `restart-lbd-e`

Comparison of different variants (decision)

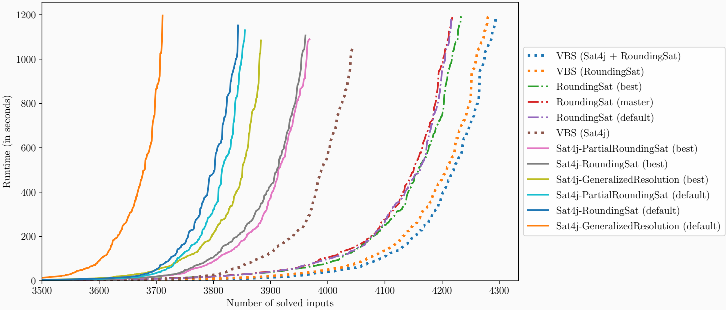


Figure 11: Cactus plot of the best configurations of different solvers

Comparison of different variants (optimization)

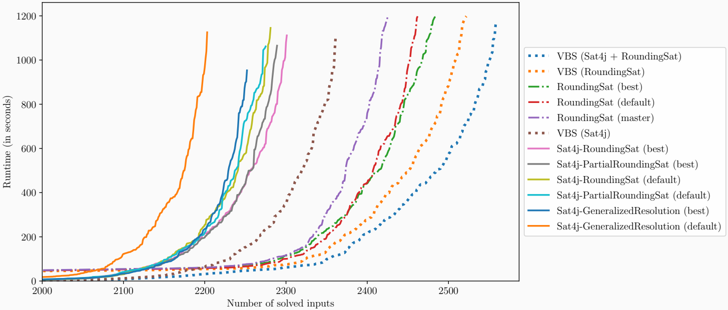


Figure 12: Cactus plot of the best configurations of different solvers

Deep Dive into CDCL Pseudo-Boolean Solvers

Romain Wallon

May 20th, 2021

Laboratoire d'Informatique de l'X (LIX), École Polytechnique

Bibliography

Audemard, G. and Simon, L. (2009). Predicting Learnt Clauses Quality in Modern SAT Solvers. In *Proceedings of IJCAI'09*, pages 399–404.

Audemard, G. and Simon, L. (2012). Refining restarts strategies for sat and unsat formulae. pages 118–126.

Chai, D. and Kuehlmann, A. (2003). A fast pseudo-boolean constraint solver. In *Proceedings of the 40th Design Automation Conference, DAC 2003, Anaheim, CA, USA, June 2-6, 2003*, pages 830–835. ACM.

Cook, S. A. (1971). The Complexity of Theorem-proving Procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, pages 151–158, New York, NY, USA. ACM.

Dixon, H. (2004). *Automating Pseudo-boolean Inference Within a DPLL Framework*. PhD thesis, Eugene, OR, USA. AAI3153782.

Bibliography

Dixon, H. E. and Ginsberg, M. L. (2002). Inference methods for a pseudo-boolean satisfiability solver. In *AAAI'02*, pages 635–640.

Elffers, J. and Nordström, J. (2018). Divide and conquer: Towards faster pseudo-boolean solving. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 1291–1299.

Hooker, J. N. (1988). Generalized resolution and cutting planes. *Annals of Operations Research*, 12(1):217–239.

Le Berre, D., Marquis, P., Mengel, S., and Wallon, R. (2020a). On irrelevant literals in pseudo-boolean constraint learning. In Bessiere, C., editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 1148–1154.

Bibliography

Le Berre, D., Marquis, P., and Wallon, R. (2020b). On weakening strategies for PB solvers. In Pulina, L. and Seidl, M., editors, *Theory and Applications of Satisfiability Testing - SAT 2020 - 23rd International Conference, Alghero, Italy, July 3-10, 2020, Proceedings*, volume 12178 of *Lecture Notes in Computer Science*, pages 322–331. Springer.

Le Berre, D. and Wallon, R. (2021). On dedicated cdcl strategies for PB solvers. In Li, C. M. and Manyà, F., editors, *Theory and Applications of Satisfiability Testing - SAT 2021 - 24th International Conference, Barcelona, Spain, July 5-10, 2021, Proceedings*, page to appear. Springer.

Bibliography

Whittemore, J. and Sakallah, J. K. K. A. (2001). SATIRE: A new incremental satisfiability engine. In *Proceedings of the 38th Design Automation Conference, DAC 2001, Las Vegas, NV, USA, June 18-22, 2001*, pages 542–545.