On Irrelevant Literals in Pseudo-Boolean Constraint Learning

Daniel Le Berre\textsuperscript{1,2} Pierre Marquis\textsuperscript{1,2,3} Stefan Mengel\textsuperscript{1} Romain Wallon\textsuperscript{1,2}

July 8, 2019

\textsuperscript{1}CRIL-CNRS UMR 8188, Lens, France
\textsuperscript{2}Université d’Artois
\textsuperscript{3}Institut Universitaire de France
A linear pseudo-Boolean (PB) constraint is of the form

$$\sum_j a_j l_j \triangle k$$

where

- $\forall j, a_j \in \mathbb{Z}$
- $\forall j, l_j$ is a literal (i.e. a Boolean value)
- $\triangle \in \{<, \leq, =, \geq, >\}$
- $k \in \mathbb{Z}$ is the degree of the constraint

Example: $3a - 2b + c - 4d \leq -1$
Normalized PB constraints

\[ \sum_j a_j l_j \geq k \text{ with } \forall j, a_j \in \mathbb{N} \text{ and } k \in \mathbb{N} \]

Example: \(3a + 2b + c \geq 3\)
Special Cases of PB Constraints

Normalized PB constraints

\[ \sum_{j} a_j l_j \geq k \text{ with } \forall j, a_j \in \mathbb{N} \text{ and } k \in \mathbb{N} \]

Example: \( 3a + 2\tilde{b} + c \geq 3 \)

Cardinality constraints

\[ \sum_{j} l_j \geq k \text{ with } k \in \mathbb{N} \]

Example: \( a + \tilde{b} + c \geq 2 \)
Special Cases of PB Constraints

Normalized PB constraints

$$\sum_j a_j l_j \geq k \text{ with } \forall j, a_j \in \mathbb{N} \text{ and } k \in \mathbb{N}$$

Example: $3a + 2\overline{b} + c \geq 3$

Cardinality constraints

$$\sum_j l_j \geq k \text{ with } k \in \mathbb{N}$$

Example: $a + \overline{b} + c \geq 2$

Clauses

$$\sum_j l_j \geq 1$$

Example: $a + \overline{b} + c \geq 1$
Generalized Resolution [Hooker, 1988]

Most PB solvers use the following rules to learn new constraints (a.k.a. no-goods) when they encounter a conflict, so as not to do the same mistake again:

\[
\frac{a_l + \sum_{i=1}^{n} a_i l_i \geq d_1}{\sum_{i=1}^{n} (ba_i + a b_i) l_i \geq bd_1 + ad_2 - ab} \quad \text{(clashing addition)}
\]

\[
\frac{\sum_{i=1}^{n} a_i l_i \geq d}{\sum_{i=1}^{n} \min(a_i, d) l_i \geq d} \quad \text{(saturation)}
\]
Most PB solvers use the following rules to learn new constraints (a.k.a. no-goods) when they encounter a conflict, so as not to do the same mistake again.

\[
\frac{a l + \sum_{i=1}^{n} a_i l_i \geq d_1}{\sum_{i=1}^{n} (ba_i + ab_i) l_i \geq bd_1 + ad_2 - ab} \quad \text{(clashing addition)}
\]

\[
\frac{\sum_{i=1}^{n} a_i l_i \geq d}{\sum_{i=1}^{n} \min(a_i, d) l_i \geq d} \quad \text{(saturation)}
\]

This proof system is (theoretically) more powerful than classical resolution: its proofs may be exponentially shorter.
Consider the following constraints

\[ \chi_1 : \bar{a} + \bar{b} + f \geq 2 \]
\[ \chi_2 : 3\bar{x} + a + b + d + e \geq 4 \]
\[ \chi_3 : 4a + 2b + 2c + x \geq 5 \]
A Conflict Analysis with Generalized Resolution

Consider the following constraints

\[ \chi_1 : \bar{a} + \bar{b} + f \geq 2 \]
\[ \chi_2 : 3\bar{x} + a + b + d + e \geq 4 \]
\[ \chi_3 : 4a + 2b + 2c + x \geq 5 \]

\[ f = 0@1 \cdot \]
Consider the following constraints:

\begin{align*}
\chi_1 : \bar{a} + \bar{b} + f & \geq 2 \\
\chi_2 : 3\bar{x} + a + b + d + e & \geq 4 \\
\chi_3 : 4a + 2b + 2c + x & \geq 5
\end{align*}
Consider the following constraints

\( \chi_1 : \overline{a} + \overline{b} + f \geq 2 \)
\( \chi_2 : 3 \overline{x} + a + b + d + e \geq 4 \)
\( \chi_3 : 4a + 2b + 2c + x \geq 5 \)
A Conflict Analysis with Generalized Resolution

Consider the following constraints

\( \chi_1 : \bar{a} + \bar{b} + f \geq 2 \)
\( \chi_2 : 3\bar{x} + a + b + d + e \geq 4 \)
\( \chi_3 : 4a + 2b + 2c + x \geq 5 \)

We have falsified \( \chi_3 \)!
A Conflict Analysis with Generalized Resolution

Consider the following constraints

\[ \chi_1 : \bar{a} + \bar{b} + f \geq 2 \]
\[ \chi_2 : 3\bar{x} + a + b + d + e \geq 4 \]
\[ \chi_3 : 4a + 2b + 2c + x \geq 5 \]

We have falsified \( \chi_3 \)! This conflict is analyzed by resolving \( \chi_3 \) against \( \chi_2 \) which is the reason for \( \bar{x} \)

\[
\begin{array}{c|c}
\chi_3 & \chi_2 \\
13a + 7b + 6c + d + e & \geq 16
\end{array}
\]
A Conflict Analysis with Generalized Resolution

Consider the following constraints

\[ \chi_1 : \overline{a} + \overline{b} + f \geq 2 \]
\[ \chi_2 : 3\overline{x} + a + b + d + e \geq 4 \]
\[ \chi_3 : 4a + 2b + 2c + x \geq 5 \]

We have falsified \( \chi_3 \)! This conflict is analyzed by resolving \( \chi_3 \) against \( \chi_2 \) which is the reason for \( \overline{x} \)

\[
\begin{array}{c|c}
\chi_3 & \chi_2 \\
\hline
13a + 7b + 6c + d + e & \geq 16
\end{array}
\]

This constraint is learned because it propagates a to 1 at level 0
The constraint learned after conflict analysis is

\[ 13a + 7b + 6c + d + e \geq 16 \]
The constraint learned after conflict analysis is

\[ 13a + 7b + 6c + d + e \geq 16 \]

Let us have a close look at this constraint...
A Problem with the Learned Constraint?

The constraint learned after conflict analysis is

\[ 13a + 7b + 6c + d + e \geq 16 \]

Let us have a close look at this constraint...
The constraint learned after conflict analysis is

$$13a + 7b + 6c + d + e \geq 16$$

Let us have a close look at this constraint...

Literal $d$ and $e$ have no effect on the constraint: they are irrelevant!
A Problem with the Learned Constraint?

The constraint learned after conflict analysis is

\[ 13a + 7b + 6c + d + e \geq 16 \]

Let us have a close look at this constraint...

Literals \( d \) and \( e \) have no effect on the constraint: they are irrelevant!

In particular, this means that removing these literals from the constraint preserves equivalence

\[ 13a + 7b + 6c \geq 16 \]
The constraint learned after conflict analysis is

\[ 13a + 7b + 6c + d + e \geq 16 \]

Let us have a close look at this constraint...

Literals \(d\) and \(e\) have no effect on the constraint: they are irrelevant!

In particular, this means that removing these literals from the constraint preserves equivalence

\[ 13a + 7b + 6c \geq 14 \]
Irrelevant Literals in Practice (in Sat4j)

- Number of irrelevant literals in Sat4j-CP’s first 5,000 learned constraints
- Experiments conducted on the 777 decision benchmarks from PB’16
- Sat4j as an example of Generalized-Resolution-based solver
RoundingSat uses a different approach, which mainly consists in using the division rule instead of saturation

\[
\frac{\sum_{i=1}^{n} a_i l_i \geq d}{\sum_{i=1}^{n} \left[ \frac{a_i}{\alpha} \right] l_i \geq \left[ \frac{d}{\alpha} \right]} \quad (\text{division})
\]
A Conflict Analysis in RoundingSat

Consider the following constraints:

\[ \chi_1 : 2\tilde{c} + 2\tilde{d} + b + \bar{e} \geq 4 \]
\[ \chi_2 : 3a + 3b + c + d + e \geq 4 \]
\[ \chi_3 : 2\bar{a} + b + e \geq 2 \]
A Conflict Analysis in RoundingSat

Consider the following constraints:

\[ \chi_1 : 2\bar{c} + 2\bar{d} + b + \bar{e} \geq 4 \]
\[ \chi_2 : 3a + 3b + c + d + e \geq 4 \]
\[ \chi_3 : 2\bar{a} + b + e \geq 2 \]

\[ e = 1@1 \]

Observe how \( c \) and \( d \) become irrelevant, and then relevant again, and how they prevent the inference of the stronger constraint \( a \) \( \bar{b} \) \( \bar{e} \) \( 1@1 \).
A Conflict Analysis in RoundingSat

Consider the following constraints:

\[ \chi_1 : 2\bar{c} + 2\bar{d} + b + \bar{e} \geq 4 \]
\[ \chi_2 : 3a + 3b + c + d + e \geq 4 \]
\[ \chi_3 : 2\bar{a} + b + e \geq 2 \]

\[ e = 1 @ 1 \]

\[ \chi_1 \]
\[ \quad \bullet \quad c = 0 @ 1 \]
\[ \chi_1 \]
\[ \quad \bullet \quad d = 0 @ 1 \]
A Conflict Analysis in RoundingSat

Consider the following constraints:

\[ \chi_1 : 2\bar{c} + 2\bar{d} + b + \bar{e} \geq 4 \]
\[ \chi_2 : 3a + 3b + c + d + e \geq 4 \]
\[ \chi_3 : 2\bar{a} + b + e \geq 2 \]

\[ e = 1\@1 \]
\[ b = 0\@2 \]
\[ c = 0\@1 \]
\[ d = 0\@1 \]
A Conflict Analysis in RoundingSat

Consider the following constraints:

\[ \chi_1 : 2\bar{c} + 2\bar{d} + b + \bar{e} \geq 4 \]
\[ \chi_2 : 3a + 3b + c + d + e \geq 4 \]
\[ \chi_3 : 2\bar{a} + b + e \geq 2 \]
Consider the following constraints:

\( \chi_1 : 2\bar{c} + 2\bar{d} + b + \bar{e} \geq 4 \)
\( \chi_2 : 3a + 3b + c + d + e \geq 4 \)
\( \chi_3 : 2\bar{a} + b + e \geq 2 \)

We have falsified \( \chi_3 \)!
A Conflict Analysis in RoundingSat

Consider the following constraints:

\[ \chi_1 : 2\bar{c} + 2\bar{d} + b + \bar{e} \geq 4 \]
\[ \chi_2 : 3a + 3b + c + d + e \geq 4 \]
\[ \chi_3 : 2\bar{a} + b + e \geq 2 \]

We have falsified \(\chi_3\)! Before applying clashing addition, \(\chi_2\) is weakened on \(e\) and divided by 3.

\[
\frac{\chi_2}{3a + 3b + c + d \geq 3} \quad \Rightarrow \quad a + b + c + d \geq 1
\]
A Conflict Analysis in RoundingSat

Consider the following constraints:

\( \chi_1 : 2\bar{c} + 2\bar{d} + b + \bar{e} \geq 4 \)
\( \chi_2 : 3a + 3b + c + d + e \geq 4 \)
\( \chi_3 : 2\bar{a} + b + e \geq 2 \)

We have falsified \( \chi_3 \)! Before applying clashing addition, \( \chi_2 \) is weakened on \( e \) and divided by 3.

\[
\frac{\chi_2}{\begin{aligned}3a + 3b + c + d & \geq 3 \\
\quad a + b + c + d & \geq 1\end{aligned}}
\]

Observe how \( c \) and \( d \) become irrelevant, and then relevant again, and how they prevent the inference of the stronger constraint \( a + b \geq 1 \).
Irrelevant Literals in Practice (in RoundingSat)

- Number of irrelevant literals in RoundingSat’s first 100,000 weakened constraints
- Experiments conducted on the 777 decision benchmarks from PB’16
Why are Irrelevant Literals an Issue?

Irrelevant literals make coefficients bigger than necessary:

$$17a + 10b + 10c + d + e \geq 17$$
Why are Irrelevant Literals an Issue?

Irrelevant literals make coefficients \textit{bigger} than necessary:

\[ 17a + 10b + 10c + d + e \geq 17 \equiv 17a + 10b + 10c \geq 15 \]
Why are Irrelevant Literals an Issue?

Irrelevant literals make coefficients **bigger** than necessary:

\[
17a + 10b + 10c + d + e \geq 17 \equiv 17a + 10b + 10c \geq 15 \\
\equiv 15a + 10b + 10c \geq 15
\]
Why are Irrelevant Literals an Issue?

Irrelevant literals make coefficients **bigger** than necessary:

\[ 17a + 10b + 10c + d + e \geq 17 \equiv 17a + 10b + 10c \geq 15 \]
\[ \equiv 15a + 10b + 10c \geq 15 \]
\[ \equiv 3a + 2b + 2c \geq 3 \]
Why are Irrelevant Literals an Issue?

Irrelevant literals make coefficients bigger than necessary:

\[ 17a + 10b + 10c + d + e \geq 17 \equiv 17a + 10b + 10c \geq 15 \]
\[ \equiv 15a + 10b + 10c \geq 15 \]
\[ \equiv 3a + 2b + 2c \geq 3 \]

*Applying generalized resolution is harder when coefficients are big due to the need of arbitrary precision*
Why are Irrelevant Literals an Issue?

Irrelevant literals make coefficients bigger than necessary:

\[17a + 10b + 10c + d + e \geq 17 \equiv 17a + 10b + 10c \geq 15\]
\[\equiv 15a + 10b + 10c \geq 15\]
\[\equiv 3a + 2b + 2c \geq 3\]

*Applying generalized resolution is harder when coefficients are big due to the need of arbitrary precision*

Irrelevant literals hide cardinality constraints:

\[3a + 3b + 3c + 3d + e + f \geq 6\]
Why are Irrelevant Literals an Issue?

Irrelevant literals make coefficients bigger than necessary:

\[
17a + 10b + 10c + d + e \geq 17 \equiv 17a + 10b + 10c \geq 15 \\
\equiv 15a + 10b + 10c \geq 15 \\
\equiv 3a + 2b + 2c \geq 3
\]

Applying generalized resolution is harder when coefficients are big due to the need of arbitrary precision.

Irrelevant literals hide cardinality constraints:

\[
3a + 3b + 3c + 3d + e + f \geq 6 \equiv 3a + 3b + 3c + 3d \geq 4
\]
Why are Irrelevant Literals an Issue?

Irrelevant literals make coefficients **bigger** than necessary:

\[ 17a + 10b + 10c + d + e \geq 17 \equiv 17a + 10b + 10c \geq 15 \]
\[ \equiv 15a + 10b + 10c \geq 15 \]
\[ \equiv 3a + 2b + 2c \geq 3 \]

**Applying generalized resolution is harder when coefficients are big due to the need of arbitrary precision**

Irrelevant literals **hide** cardinality constraints:

\[ 3a + 3b + 3c + 3d + e + f \geq 6 \equiv 3a + 3b + 3c + 3d \geq 4 \]
\[ \equiv a + b + c + d \geq 2 \]
Why are Irrelevant Literals an Issue?

Irrelevant literals make coefficients bigger than necessary:

\[ 17a + 10b + 10c + d + e \geq 17 \equiv 17a + 10b + 10c \geq 15 \]
\[ \equiv 15a + 10b + 10c \geq 15 \]
\[ \equiv 3a + 2b + 2c \geq 3 \]

*Applying generalized resolution is harder when coefficients are big due to the need of arbitrary precision*

Irrelevant literals hide cardinality constraints:

\[ 3a + 3b + 3c + 3d + e + f \geq 6 \equiv 3a + 3b + 3c + 3d \geq 4 \]
\[ \equiv a + b + c + d \geq 2 \]

*Efficient data structures implemented in PB solvers cannot be used when cardinality constraints are hidden*
We observed that constraints derived using clashing addition or weakening may contain irrelevant literals, making these constraints weaker and harder to handle for the solver.
Towards Irrelevant-Literal-Free Constraint Learning

We observed that constraints derived using clashing addition or weakening may contain irrelevant literals, making these constraints weaker and harder to handle for the solver.

We can remove these literals from the constraints produced by the solver.
Towards Irrelevant-Literal-Free Constraint Learning

We observed that constraints derived using clashing addition or weakening may contain irrelevant literals, making these constraints weaker and harder to handle for the solver.

*We can remove these literals from the constraints produced by the solver*

Unfortunately... Checking whether a literal is relevant is NP-complete!
Towards Irrelevant-Literal-Free Constraint Learning

We observed that constraints derived using clashing addition or weakening may contain irrelevant literals, making these constraints weaker and harder to handle for the solver.

*We can remove these literals from the constraints produced by the solver*

Unfortunately... Checking whether a literal is relevant is **NP-complete**!

So, in practice, performing a complete removal on all inferred constraints seems out of reach.
Towards Irrelevant-Literal-Free Constraint Learning

We observed that constraints derived using clashing addition or weakening may contain irrelevant literals, making these constraints weaker and harder to handle for the solver.

We can remove these literals from the constraints produced by the solver.

Unfortunately... Checking whether a literal is relevant is NP-complete!

So, in practice, performing a complete removal on all inferred constraints seems out of reach.

But we can still consider an incomplete approach, treating only small learned constraints, and optimizing the detection phase.
Let us consider again the constraint we learned earlier

\[ \chi : 13a + 7b + 6c + d + e \geq 16 \]
Let us consider again the constraint we learned earlier

\[ \chi : 13a + 7b + 6c + d + e \geq 16 \]

Formally, \( e \) is irrelevant in \( \chi \) because the following equivalences hold

\[ \chi \equiv \chi | \bar{e} \equiv \chi | e \]
Let us consider again the constraint we learned earlier

\[ \chi : 13a + 7b + 6c + d + e \geq 16 \]

Formally, \( e \) is irrelevant in \( \chi \) because the following equivalences hold

\[ \chi \equiv \chi|\bar{e} \equiv \chi|e \]

In particular, observe that \( \chi|\bar{e} \models \chi \) always holds.
Let us consider again the constraint we learned earlier

$$\chi: 13a + 7b + 6c + d + e \geq 16$$

Formally, $e$ is irrelevant in $\chi$ because the following equivalences hold

$$\chi \equiv \chi|\bar{e} \equiv \chi|e$$

In particular, observe that $\chi|\bar{e} \models \chi$ always holds

So, only the following entailment has to be checked

$$\chi \models \chi|\bar{e}$$
Let us consider again the constraint we learned earlier

$$\chi : 13a + 7b + 6c + d + e \geq 16$$

Formally, $e$ is irrelevant in $\chi$ because the following equivalences hold

$$\chi \equiv \chi|\bar{e} \equiv \chi|e$$

In particular, observe that $\chi|\bar{e} \models \chi$ always holds

So, only the following entailment has to be checked

$$\chi \models 13a + 7b + 6c + d \geq 16$$
Detecting Irrelevant Literals: Relevance Check (1)

Let us consider again the constraint we learned earlier

$$\chi : 13a + 7b + 6c + d + e \geq 16$$

Formally, $e$ is irrelevant in $\chi$ because the following equivalences hold

$$\chi \equiv \chi|\bar{e} \equiv \chi|e$$

In particular, observe that $\chi|\bar{e} \models \chi$ always holds

So, only the following entailment has to be checked

$$\chi \models 13a + 7b + 6c + d \geq 16$$

This test can be achieved by verifying that this formula is unsatisfiable

$$\chi \land \neg \left(13a + 7b + 6c + d \geq 16\right)$$
Let us consider again the constraint we learned earlier
\[ \chi : 13a + 7b + 6c + d + e \geq 16 \]

Formally, \( e \) is irrelevant in \( \chi \) because the following \textit{equivalences} hold
\[ \chi \equiv \chi|\bar{e} \equiv \chi|e \]

In particular, observe that \( \chi|\bar{e} \models \chi \) always holds

So, only the following \textit{entailment} has to be checked
\[ \chi \models 13a + 7b + 6c + d \geq 16 \]

This test can be achieved by verifying that this formula is \textit{unsatisfiable}
\[ \chi \land \left(13a + 7b + 6c + d < 16\right) \]
Let us consider again the constraint we learned earlier
\[ \chi : 13a + 7b + 6c + d + e \geq 16 \]

Formally, \( e \) is irrelevant in \( \chi \) because the following \textit{equivalences} hold
\[ \chi \equiv \chi|\overline{e} \equiv \chi|e \]

In particular, observe that \( \chi|\overline{e} \models \chi \) always holds.

So, only the following \textit{entailment} has to be checked
\[ \chi \models 13a + 7b + 6c + d \geq 16 \]

This test can be achieved by verifying that this formula is \textit{unsatisfiable}
\[ \chi \land (12\overline{a} + 7\overline{b} + 6\overline{c} + \overline{d} \geq 12) \]
Detecting Irrelevant Literals: Relevance Check (1)

Let us consider again the constraint we learned earlier

\[ \chi : 13a + 7b + 6c + d + e \geq 16 \]

Formally, \( e \) is irrelevant in \( \chi \) because the following equivalences hold

\[ \chi \equiv \chi|\bar{e} \equiv \chi|e \]

In particular, observe that \( \chi|\bar{e} \models \chi \) always holds

So, only the following entailment has to be checked

\[ \chi \models 13a + 7b + 6c + d \geq 16 \]

This test can be achieved by verifying that this formula is unsatisfiable

\[ \chi \land (12\bar{a} + 7\bar{b} + 6\bar{c} + \bar{d} \geq 12) \]

*The relevance of a literal can be checked using a PB solver*
Let us consider again the constraint we learned earlier

\[ \chi : 13a + 7b + 6c + d + e \geq 16 \]
Let us consider again the constraint we learned earlier

\[ \chi : 13a + 7b + 6c + d + e \geq 16 \]

Observe the following equation (which encodes a subset-sum problem)

\[ 13a + 7b + 6c + d = (16 - 1) = 15 \]
Let us consider again the constraint we learned earlier

\[ \chi : 13a + 7b + 6c + d + e \geq 16 \]

Observe the following equation (which encodes a subset-sum problem)

\[ 13a + 7b + 6c + d = (16 - 1) = 15 \]

If it has a solution, the corresponding model can be extended to a model of \( \chi \) by satisfying \( e \), which would hence be relevant.
Let us consider again the constraint we learned earlier

\[ \chi : 13a + 7b + 6c + d + e \geq 16 \]

Observe the following equation (which encodes a subset-sum problem)

\[ 13a + 7b + 6c + d = (16 - 1) = 15 \]

If it has a solution, the corresponding model can be extended to a model of \( \chi \) by satisfying \( e \), which would hence be relevant.

Note that multiple such equations may need to be considered for one literal.
Let us consider again the constraint we learned earlier

\[ \chi : 13a + 7b + 6c + d + e \geq 16 \]

Observe the following equation (which encodes a subset-sum problem)

\[ 13a + 7b + 6c + d = (16 - 1) = 15 \]

If it has a solution, the corresponding model can be extended to a model of \( \chi \) by satisfying \( e \), which would hence be relevant

Note that multiple such equations may need to be considered for one literal (see next slide)
Let us consider again the constraint we learned earlier

$$\chi : 13a + 7b + 6c + d + e \geq 16$$

Observe the following equation (which encodes a *subset-sum* problem)

$$13a + 7b + 6c + d = (16 - 1) = 15$$

If it has a solution, the corresponding model can be extended to a model of $\chi$ by satisfying $e$, which would hence be *relevant*.

Note that *multiple* such equations may need to be considered for one literal (see next slide).

*Another alternative to implement the relevance check is to use the dynamic programming algorithm for subset-sum*
Let us consider again the constraint we learned earlier

\[ \chi : 13a + 7b + 6c + d + e \geq 16 \]

Observe the following equation (which encodes a subset-sum problem)

\[ 13a + 7b + 6c + d = (16 - 1) = 15 \]

If it has a solution, the corresponding model can be extended to a model of \( \chi \) by satisfying \( e \), which would hence be relevant

Note that multiple such equations may need to be considered for one literal (see next slide)

Another alternative to implement the relevance check is to use the dynamic programming algorithm for subset-sum

Its ability to compute “efficiently” all possible sums is crucial here
Let us consider again the constraint we learned earlier

\[ \chi : 13a + 7b + 6c + d + e \geq 16 \]
Let us consider again the constraint we learned earlier

$$\chi : 13a + 7b + 6c + d + e \geq 16$$

Because $e$ is irrelevant, so is $d$ which shares the same coefficient

$$\chi : 13a + 7b + 6c \geq 14$$

*At most one relevance check per coefficient is required*
Detecting Irrelevant Literals: Minimizing the Checks

Let us consider again the constraint we learned earlier

\[ \chi : 13a + 7b + 6c + d + e \geq 16 \]

Because \( e \) is irrelevant, so is \( d \) which shares the same coefficient

\[ \chi : 13a + 7b + 6c \geq 14 \]

*At most one relevance check per coefficient is required*

To check whether \( c \) is relevant, we consider all the equations of the following form, with \( 14 - 6 = 8 \leq n < 14 \)

\[ 13a + 7b = n \]
Detecting Irrelevant Literals: Minimizing the Checks

Let us consider again the constraint we learned earlier

\[ \chi : 13a + 7b + 6c + d + e \geq 16 \]

Because \( e \) is irrelevant, so is \( d \) which shares the same coefficient

\[ \chi : 13a + 7b + 6c \geq 14 \]

*At most one relevance check per coefficient is required*

To check whether \( c \) is relevant, we consider all the equations of the following form, with \( 14 - 6 = 8 \leq n < 14 \)

\[ 13a + 7b = n \]

There is a solution for \( n = 13 \), so all remaining literals are relevant
Let us consider again the constraint we learned earlier
\[ \chi : 13a + 7b + 6c + d + e \geq 16 \]
Because \( e \) is irrelevant, so is \( d \) which shares the same coefficient
\[ \chi : 13a + 7b + 6c \geq 14 \]
At most one relevance check per coefficient is required

To check whether \( c \) is relevant, we consider all the equations of the following form, with \( 14 - 6 = 8 \leq n < 14 \)
\[ 13a + 7b = n \]
There is a solution for \( n = 13 \), so all remaining literals are relevant

If a literal is relevant, so it is for all literals with a greater coefficient
The previous observations lead to the following algorithm

**Algorithm 1: detect-and-remove-irrelevant-literals**

**Input:** A non-valid pseudo-Boolean constraint $\chi$

**Output:** The constraint $\chi$, in which all irrelevant literals are removed

**foreach** coefficient $c$ of the constraint in ascending order **do**

Choose a literal $l$ having coefficient $c$

**if** depends($l$, $\chi$) **then**

| return $\chi$

end

Remove all literals having coefficient $c$ from $\chi$

Update the degree of $\chi$

Saturate $\chi$

end
Experiments

Settings

- Quadcore bi-processors Intel XEON E5-5637 v4 (3.5 GHz)
- 128 GB of memory
- 777 decision benchmarks submitted to PB’16
Experiments

Settings

- Quadcore bi-processors Intel XEON E5-5637 v4 (3.5 GHz)
- 128 GB of memory
- 777 decision benchmarks submitted to PB’16

Experimented approaches

- Detection and removal implemented in Sat4j-CuttingPlanes
- Using a dynamic programming algorithm or a PB solver (5-second timeout per call)
- Only applied to learned constraints having less than 1,000 literals and a degree less than 20,000
Experimental Results: Detection using Dynamic Programming

- Instances not solved at all are not presented

- A circle stands for SATISFIABLE and a square stands for UNSATISFIABLE
Experimental Results: Detection with Sat4j-CP

Instances not solved at all are not presented.

A " stands for SATISFIABLE and a " stands for UNSATISFIABLE.
Experimental Results: Detection with Sat4j-Res

- Instances not solved at all are not presented
- A □ stands for SATISFIABLE and a ◯ stands for UNSATISFIABLE
Experimental Results: Detection Runtime

Comparison of the time spent detecting irrelevant literals (in % of the runtime)

- Dynamic-Programming
- Sat4j-CP
- Sat4j-Res

Detection approach
Conclusion

- Irrelevant literals may occur in constraints learned by PB solvers
- These literals may impact the performance of PB solvers
- Removing irrelevant literals is a first step to correct this behavior
Conclusion

- Irrelevant literals may occur in constraints learned by PB solvers
- These literals may impact the performance of PB solvers
- Removing irrelevant literals is a first step to correct this behavior

Perspectives

- Improve the detection algorithm to remove more literals
- Find a proof system guaranteeing not to infer such literals
On Irrelevant Literals in Pseudo-Boolean Constraint Learning

Daniel Le Berre\textsuperscript{1,2} Pierre Marquis\textsuperscript{1,2,3} Stefan Mengel\textsuperscript{1} Romain Wallon\textsuperscript{1,2}

July 8, 2019

\textsuperscript{1}CRIL-CNRS UMR 8188, Lens, France
\textsuperscript{2}Université d’Artois
\textsuperscript{3}Institut Universitaire de France