

Understanding the power of Max-SAT resolution through UP-resilience

Mohamed Sami Cherif, Djamal Habet, André Abramé

▶ **To cite this version:**

Mohamed Sami Cherif, Djamal Habet, André Abramé. Understanding the power of Max-SAT resolution through UP-resilience. Artificial Intelligence, 2020, 289, pp.103397. 10.1016/j.artint.2020.103397 . hal-02977156

HAL Id: hal-02977156
https://amu.hal.science/hal-02977156

Submitted on 17 Oct 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Version of Record: https://www.sciencedirect.com/science/article/pii/S0004370220301478
Manuscript_5d251fe01bfafc5717edaec3095e93eb

Understanding the Power of Max-SAT Resolution Through UP-Resilience[⋆,⋆⋆]

Mohamed Sami Cherif[a], Djamal Habet[a], André Abramé[b]

[a] *Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France*
[b] *Logike, France*

Abstract

A typical Branch and Bound algorithm for Max-SAT computes the lower bound by estimating the number of disjoint Inconsistent Subsets (IS) of the formula. The IS detection is ensured by Simulated Unit Propagation (SUP). Then, the inference rule for Max-SAT, Max-SAT resolution, is applied to ensure that the detected IS is counted only once. Learning Max-SAT resolution transformations can be detrimental to the algorithm performance, so they are usually selectively learned if they match certain patterns. In this paper, we study the impact of the transformations by Max-SAT resolution on the SUP mechanism, indispensable for IS detection. We introduce the notion of UP-resilience of a transformation which quantifies this impact and provides, from a theoretical point of view, an explanation to the empirical efficiency of the learning schemes developed in the last ten years. We also focus on recently introduced patterns called Unit Clause Subsets (UCSs). We characterize the transformations of certain UCSs using UP-resilience and we explain how our result can help extend the current patterns. Finally, we present empirical observations that support the relevance of the UP-resilience property and further consolidate our theoretical results.

Keywords: Branch and Bound, Max-SAT resolution, Simulated Unit Propagation, UP-resilience, Pattern

⋆This paper is based on work published in [1] and [2].

⋆⋆This work has been partially funded by the French National Research Agency, reference ANR-16-C40-0028.

Email addresses: mohamedsami.cherif@lis-lab.fr (Mohamed Sami Cherif), djamal.habet@lis-lab.fr (Djamal Habet), andre.abrame@logike.fr (André Abramé)

© 2020 published by Elsevier. This manuscript is made available under the CC BY NC user license
https://creativecommons.org/licenses/by-nc/4.0/

1. Introduction

The maximum satisfiability (Max-SAT) problem is an optimization extension of the satisfiability (SAT) problem. For a given formula in Conjunctive Normal Form (CNF), it consists in finding an assignment of the variables which maximizes the number of satisfied clauses. Complete methods for this problem include SAT based approaches (e.g. RC2 [3], MAXHS [4], OPEN-WBO [5], EVA [6], WPM1 [7]) and Branch and Bound (BnB) algorithms (e.g. AHMAXSAT [8], MiniMaxSAT [9], AKMAXSAT [10], MAXSATZ [11, 12]) among others. The former which iteratively call SAT solvers are particularly efficient on industrial instances while the latter are competitive on random and crafted instances.

BnB based approaches construct a search tree and compute, at each node, the Lower Bound (LB) by counting the disjoint Inconsistent Subsets (ISs) of the formula using Simulated Unit Propagation (SUP) [13]. When an IS is found, it is either temporarily deleted or transformed by Max-SAT resolution, the inference rule for Max-SAT [14, 15], to ensure that it will be counted only once. However, learning Max-SAT resolution transformations, i.e., memorizing them in the current subtree (including the current node), may affect negatively the quality of the lower bound estimation [1, 12, 16]. Therefore, state of the art solvers learn transformations selectively mainly in the form of patterns [12].

Recently, new patterns called Unit Clause Subsets (UCSs) were introduced and empirically studied in [16]. The most significant feature of these patterns is producing unit clauses after the transformation by Max-SAT resolution. The propagation of these clauses may lead to the detection of more ISs. The empirical study of these patterns lead to the first observations on the relation between Max-SAT resolution transformations and the efficiency of the SUP mechanism which is indispensable for the lower bound estimation.

In this paper, we formally state these observations by introducing a new property called UP-resilience. More specifically, we study the impact of the

Max-SAT resolution transformations on the SUP mechanism. We show that in some cases the information which can be used by SUP in the original formula are, after transformation, fragmented in several clauses. In such situation, SUP may be less efficient in the transformed formula than in the original one. We introduce the notion of UP-resilience to characterize the transformations which are not affected by fragmentation and more generally to measure the impact of the transformations on the SUP mechanism. We show that according to this criterion, the most used learning scheme based on patterns does not affect SUP. This contributes to explain from a theoretical point of view the empirical results obtained in the last ten years on the development of inference rules [12]. Also, in the context of a new approach to extend the current learning mechanisms, we conduct a theoretical study on particular UCS patterns and, more specifically, their relation with UP-resilience: we prove that binary UCSs are UP-resilient and we generalize this result on UCSs where only one clause of any size is involved in the conflict. Furthermore, we explain how our results can help extend the current patterns by showing that the current mechanisms in BnB solvers can't ensure UP-resilience for these patterns. Finally, we present empirical observations that also support the relevance of the UP-resilience property.

This paper brings together, highlights and extends results presented in previous conference publications [1, 2]. In particular, the notion of UP-resilience was first introduced and used to quantify the impact of Max-SAT resolution transformations on the SUP mechanism in [1]. Moreover, binary UCSs and UCSs patterns where only one clause of any size is involved in the conflict were proven to be UP-resilient in [2]. In this paper, we extend those results by showing that application orders can have a direct impact not only on the UP-resilience of certain literals appearing in the detected IS but also on the UP-resilience of the whole Max-SAT resolution transformation of the IS. Furthermore, we provide a proof for the UP-resilience of one of the major patterns, which does not correspond to a UCS, thus reinforcing our observations on the relevance of this property. We also provide more detailed experiments and new empirical results on partial instances. Finally, this paper includes a thorough background

on Branch and Bound algorithms for Max-SAT and on the use of Max-SAT resolution in this context.

This paper is organized as follows. In Section 2, we give basic definitions and notations and we present a brief overview of BnB solvers for Max-SAT. In Section 3, we describe the fragmentation phenomenon as a motivation to our work. We introduce the notion of UP-resilience in Section 4 and we show how it highlights the impact of Max-SAT resolution transformations on the SUP mechanism. In Section 5, we conduct a theoretical study on particular UCS patterns and, more specifically, their relation with UP-resilience and we show the limits of the current mechanisms. In Section 6, we present empirical evidence on the relevance of UP-resilience. Finally, we conclude and we give some prospective points for future work in Section 7.

2. Preliminaries

2.1. Definitions and Notations

Let X be a set of propositional variables. A literal l is a variable $x \in X$ or its negation $\overline{x}$ and a clause is disjunction of literals, represented as a set of literals. A formula in Conjunctive Normal Form (CNF) is a conjunction of clauses and can be represented as a set of clauses. An assignment $I : X \longrightarrow \{true, false\}$ maps each variable to a Boolean value and is represented as a set of literals. For a given literal l, $var(l)$ denotes the variable appearing in l. A clause c is satisfied by an assignment I if at least one of its literals is satisfied, i.e., $\exists l \in c$ such that $l \in I$. The empty clause $\square$ is always falsified. For a unit assignment $I = \{l\}$, we denote by $\Phi|_I$ the formula obtained by reducing Φ by I, i.e, $\Phi|_I = \{c \in \Phi \mid \{l, \bar{l}\} \cap c = \emptyset\} \cup \{c \setminus \{\bar{l}\} \mid c \in \Phi, \bar{l} \in c\}$. This notation can be extended to any assignment $I = \{l_1, l_2, \ldots, l_k\}$ as follows: $\Phi|_I = (\ldots((\Phi|_{\{l_1\}})|_{\{l_2\}})\ldots|_{\{l_k\}})$. An Inconsistent Subset (IS) of a formula Φ is an unsatisfiable set of clauses $\psi \subseteq \Phi$. Solving the Max-SAT problem consists in finding an assignment which maximizes the number of satisfied clauses for a given CNF formula. There exists other variants of Max-SAT (weighted, partial and weighted partial Max-SAT)

which are not considered in this paper. Nevertheless, the presented results can be easily extended to these variants.

2.2. Branch and Bound Solvers for Max-SAT

2.2.1. Outline

Branch and Bound (BnB) algorithms for Max-SAT explore the search space by constructing a binary search tree while maintaining two values: the Upper Bound (UB) and the Lower Bound (LB) which correspond respectively to the best computed solution so far and to the estimation of the best accessible solution. In each node of the search tree, the current assignment is extended, if possible, using rules such as Pure Literals (PL) and Dominating Unit Clause (DUC) [17]. Then, the value of LB is computed and compared to UB. If it is greater, finding a better solution in the current sub-tree is not possible and, thus, the algorithm backtracks. If a complete assignment is reached, the UB value is updated and a backtrack is performed. Else, an unassigned variable is chosen using a branching heuristic [8, 9, 10, 11]. These steps are repeated until the whole search space is explored. As the efficiency of BnB algorithms depends on the quality of LB, it is important to note that state of the art BnB Max-SAT solvers use powerful mechanisms to compute a more accurate LB estimation enabling more cuts in the search tree and, thus, faster solving.

2.2.2. LB Estimation

The LB estimation is one of the most critical components of a BnB Max-SAT algorithm. Indeed, computing lower bounds with better quality entails more cuts in the search tree and, thus, faster solving. However, as these computations are done frequently during the search, the time devoted to them is closely correlated to the speed of the solver and, therefore, its efficiency. So, it is important to strike a balance between the computation time for LB and its quality. A simple estimation for LB is the number of the falsified clauses of the formula Φ by the current assignment I. Formally, $LB = \text{falsified_clauses}(\Phi|_I) = |\{c \in \Phi \mid c|_I = \square\}|$. As this simple estimation is often quite far from the best accessible

solution, BnB Max-SAT solvers refine it by calculating the number of disjoint Inconsistent Subsets (ISs) in the simplified formula $\Phi|_I$, where I is the current assignment, using Simulated Unit Propagation (SUP) [13].

SUP replicates the Unit Propagation (UP) mechanism, used in SAT solvers and which consists in iteratively satisfying the literals appearing in unit clauses until a conflict is found (an empty clause), to detect ISs. However, UP is not a valid inference rule for Max-SAT, i.e., it does not necessarily maintain the number of falsified clauses for each assignment. Thus, the variable assignments made by UP are not added to the current assignment after the LB estimation. More specifically, these variable assignments are stored in a separate temporary assignment extension. That is why we say that BnB Max-SAT algorithms simulate UP and we refer to this mechanism as Simulated Unit Propagation. The propagation steps generated by SUP can be represented by an implication graph [18] defined as follows in the context of BnB solvers for Max-SAT:

Definition 1 (Implication Graph of an IS)**.** *Let ψ be an IS of a CNF formula Φ and I an assignment. We suppose that exactly one clause is falsified by I (SUP stopped when the first empty clause is generated). An implication graph of ψ is a directed acyclic graph $G = (V, A)$ defined as follows:*

- $V = \{l \in I\} \cup \{\diamond_c \mid c \in \psi \text{ and } |c| = 1\} \cup \{\square\}$
- $A = \{(l, l', c) \mid l, l' \in I \text{ and } c \in \psi \text{ is reduced by } l \text{ and propagates } l'\} \bigcup$
  $\{(\diamond_c, l, c) \mid l \in I \text{ and } c = \{l\} \in \psi\} \bigcup$
  $\{(l, \square, c) \mid l \in I \text{ and } c \in \psi \text{ is falsified by } I \text{ and } \bar{l} \in \psi\}$

The directed edges are labeled by clauses and the nodes $\diamond$ are omitted in G.

Notation. *For an implication graph G and a literal $l \in V$ (possibly $\square$), we denote :*

- $pred_G(l)$ *(resp.* $succ_G(l)$*) the predecessors (resp. successors) of l in G*
- $neigh_G(l) = pred_G(l) \cup succ_G(l)$
- $src_G(l)$ *the clause that lead to the propagation of the literal l in G*

The index G will be omitted if there is no confusion.

When an IS is detected, it must be counted only once in the LB estimation. To this end, detected ISs are treated using two methods: they are either temporarily deleted or transformed by Max-SAT resolution. The deletion of ISs has several advantages: it is less time-consuming and it doesn't increase the size of the formula. However, it produces a formula that is not equivalent to the original and, thus, that may contain less ISs. Therefore, this method applies local changes on the formula that are only preserved during the LB estimation in the current node. In the next section we explain the second method which transforms ISs using Max-SAT resolution.

2.2.3. Max-SAT resolution Transformation

The most used method to treat ISs, outlined in Algorithm 1, is based on the Max-SAT resolution inference rule defined below. A Max-SAT resolution transformation requires an implication graph describing the sequence of propagations leading to the detection of the IS. The transformation is usually done in the reverse order of propagation. The treatment performed by Max-SAT resolution$(\Phi, c_1, c_2, var(l))$ (line 6 in Algorithm 1) consists in deleting the clauses c_1 and c_2 and adding the compensation clauses and the resolvent clause cr (also returned after the transformation) to the formula as described in the Max-SAT resolution rule defined as follows:.

Definition 2 (**Max-SAT resolution** [14, 15])**.** *The inference rule for Max-SAT, max- resolution, is defined as follows:*

$$\frac{c = \{x, y_1, ..., y_s\}, c' = \{\overline{x}, z_1, ..., z_t\}}{cr = \{y_1, ..., y_s, z_1, ..., z_t\}, cc_1, ..., cc_t, cc_{t+1}, ..., cc_{t+s}}$$

where the compensation clauses are defined as follows:

$$
\begin{array}{ll}
cc_1 = \{x, y_1, ..., y_s, \overline{z_1}\} & cc_{t+1} = \{\overline{x}, z_1, ..., z_t, \overline{y_1}\} \\
cc_2 = \{x, y_1, ..., y_s, z_1, \overline{z_2}\} & cc_{t+2} = \{\overline{x}, z_1, ..., z_t, y_1, \overline{y_2}\} \\
\ldots & \ldots \\
cc_t = \{x, y_1, ..., y_s, z_1, ..., z_{t-1}, \overline{z_t}\} & cc_{t+s} = \{\overline{x}, z_1, ..., z_t, y_1, ..., y_2, \overline{y_s}\}
\end{array}
$$

Remark. *Unlike the SAT inference rule, Max-SAT resolution replaces the premises in the rule by its conclusions. In addition, it produces an equivalent formula, i.e., it preserves the number of unsatisfied clauses for any assignment. The results established in this paper can be easily extended to weighted Max-SAT formulas (hard clauses can be included with infinite weights in the case of partial formulas) using the weighted version of Max-SAT resolution introduced in [14]. Furthermore, as Max-SAT resolution is also a sound and complete inference rule for Min-SAT [19], our results remain valid in the context of B&B Min-Sat solvers.*

Definition 3 (Max-SAT resolution transformation of an IS)**.** *Let ψ be an IS of a CNF formula Φ and $S = \langle x_1, ..., x_k \rangle$ be a sequence of variables appearing in ψ. The Max-SAT resolution transformation of ψ with respect to S, denoted $\Theta(\psi, S)$, is the set of clauses obtained from ψ after the application of Max-SAT resolution steps in accordance to the sequence S, i.e., $\Theta(\psi, S) = \theta(\theta...(\theta(\psi, x_1), x_2)..., x_k)$ where $\theta(\psi, x)$ denotes the application of the Max-SAT resolution step on two clauses c and c' such that $x \in c$ and $\overline{x} \in c'$.*

Remark. *A transformation $\Theta(\psi, S)$ is not unique in general as the application of Max-SAT resolution steps with respect to a given variable sequence S is not deterministic and, thus, $\Theta(\psi, S)$ represents any of the possible outcomes after the transformation of ψ with respect to S. However, if S represents the reverse propagation order, the transformation with respect to S becomes deterministic and $\Theta(\psi, S)$ becomes unique.*

Algorithm 1: Max-SAT resolution Transformation of an IS

Input: a CNF formula Φ, an implication graph G of an IS of Φ

Output: the transformed formula

```
1  begin
2      c_1 ← src_G(□);
3      while c_1 ≠ □ do
4          l ← the last propagated literal in G;
5          c_2 ← src_G(l);
6          cr ← max_resolution(Φ, c_1, c_2, var(l));
7          c_1 ← cr;
8      end
9      return Φ
10 end
```

Example 1. *We consider the IS* $\psi = \{\{x_1\}, \{x_2\}, \{x_3\}, \{\overline{x_3}, \overline{x_4}\}, \{\overline{x_1}, \overline{x_2}, x_4\}\}$ *detected by the sequence of unit propagations represented in the form of an implication graph on the left in Figure 1. The Max-SAT resolution transformation of* ψ *with respect to the variable sequence* $S = \langle x_4, x_3, x_2, x_1 \rangle$ *(in the reverse order of propagation) is given on the right in Figure 1.*

It is important to note that, unlike the deletion method, the changes are maintained in the sub-tree and, thus, the detected IS is computed and counted only once. This is possible since Max-SAT resolution produces an equivalent formula and this entails an incremental calculation of LB while reducing redundancy in the detection of ISs. Nevertheless, this method has several shortcomings: it is time-consuming and it may increase the size of the formula by adding the compensation clauses obtained after each Max-SAT resolution step. We explain in the next section how state of the art solvers overcome these limits.

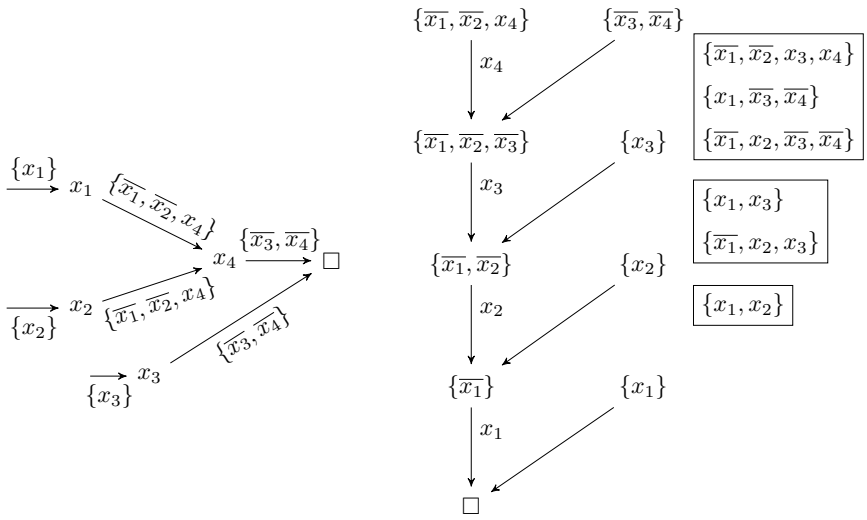


Figure 1: Implication graph corresponding to a propagation sequence of ψ and its transformation by Max-SAT resolution, where compensation clauses for each step are represented in boxes

2.2.4. Learning Schemes

One of the major challenges for BnB solvers for Max-SAT is to perform efficient Max-SAT resolution transformations of ISs. There are two main learning schemes. The first scheme learns a transformation only if all the intermediary resolvents are smaller than four [9] whereas the second learns transformations matching particular patterns. These patterns are mainly described by inference rules that can be deduced from Max-SAT resolution. They can be specific cases of Max-SAT resolution or a combination of several Max-SAT resolution steps. We give below three major patterns used in state of the art solvers:

$$\frac{\{l_1, l_2\}, \{l_1, \overline{l_2}\}}{\{l_1\}} \ (P_1)$$

$$\frac{\{l_1, l_2\}, \{l_1, l_3\}, \{\overline{l_2}, \overline{l_3}\}}{\{l_1\}, \{l_1, l_2, l_3\}, \{\overline{l_1}, \overline{l_2}, \overline{l_3}\}} \ (P_2)$$

$$\frac{\{l_1\}, \{\overline{l_1}, l_2\}, \{\overline{l_2}, l_3\}, ..., \{\overline{l_k}, l_{k+1}\}, \{\overline{l_{k+1}}\}}{\square, \{l_1, \overline{l_2}\}, \{l_2, \overline{l_3}\}, ..., \{l_k, \overline{l_{k+1}}\}} \ (P_3)$$

P_1 is a particular case of Rule 1 in [12] while P_2 is related to Rule 5 and P_3

corresponds to Rule 4. In their weighted form, the pattern (P_1) corresponds to a particular case of chain resolution rule [20] whereas patterns ($P2$) and ($P3$) are specific cases of the cycle resolution rule [20]. A pattern can cover an IS entirely as in pattern ($P3$), or partially as in patterns ($P1$) and ($P2$). These patterns present several advantages: they can be easily identified and they don't increase the number of clauses in the formula. Furthermore, patterns ($P1$) and ($P2$) produce a unit resolvent clause which enables the solver to detect more ISs using SUP.

Recently, new patterns called Unit Clause Subsets (UCS) [16], were introduced and empirically studied in order to extend the learning mechanisms in BnB Max-SAT solvers. Hereafter, we give the formal definition of these patterns and we explain how they can be easily detected using the implication graph of an IS.

Definition 4 (Unit Clause Subset)**.** *Let Φ be a CNF formula and k a natural number. A k-Unit Clause Subset where $k \geq 2$, denoted k-UCS, is a set of clauses $\{c_1, ..., c_k\} \subseteq \Phi$ such that there exists a sequence of Max-SAT resolution steps on $c_1, ..., c_k$ that produces a unit clause reslovent. In particular, if $\forall i \in \{1, ..., k\}$ we have $|c_i| = 2$, it is a binary k-UCS, denoted k^b-UCS.*

Example 2. *The patterns ($P1$) and ($P2$) which are learned in state of the art BnB solvers, correspond respectively to a 2^b-UCS and a 3^b-UCS.*

Definition 5 (First Unique Implication Point [21])**.** *Let G be an implication graph. A Unique Implication Point (UIP) is any node in G such that any path from the literals propagated by unit clauses to the conflict node must pass through it. The First UIP (FUIP) is the UIP closest to the conflict node.*

UCS patterns have a high apparition frequency (in more than 57% of the detected ISs [16]). Furthermore, certain k-UCS patterns are easily detectable by analyzing the implication graph of the obtained IS. Indeed, as outlined in Example 3, the clauses which are between the conflict and the FUIP produce a unit resolvent clause if they are transformed by Max-SAT resolution in the reverse propagation order.

Example 3. *We consider the IS* $\psi = \{\{x_1\}, \{x_2\}, \{\overline{x_1}, \overline{x_2}, x_3\}, \{\overline{x_3}, x_4\}, \{\overline{x_3}, x_5\}, \{\overline{x_4}, \overline{x_5}\}\}$ *detected by the sequence of unit propagations represented in the form of the implication graph* G *on the left in Figure 2. Clearly, the node* x_3 *is the FUIP of* G*. The Max-SAT resolution transformation of* ψ *with respect to the variable sequence* $S = \langle x_5, x_4, x_3 \rangle$ *(representing the reverse order of propagation until the FUIP is encountered) is given on the right in Figure 2. This transformation produces the unit resolvent* $\{\overline{x_3}\}$*. Therefore, the set of clauses* $\psi' = \{\{\overline{x_3}, x_4\}, \{\overline{x_3}, x_5\}, \{\overline{x_4}, \overline{x_5}\}\} \subset \psi$ *is a* 3*-UCS. More specifically, since all the clauses in* ψ' *are binary, it is a* 3^b*-UCS.*

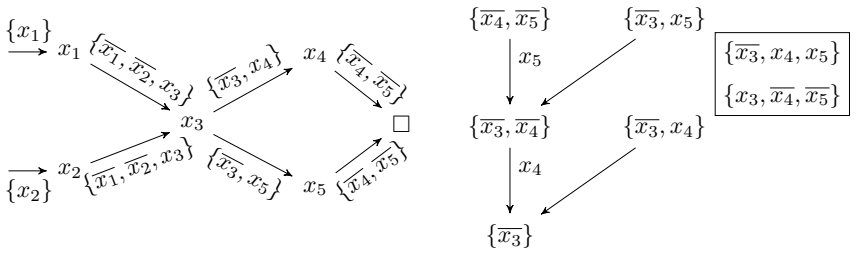


Figure 2: Implication graph corresponding to a propagation sequence of ψ and its transformation by Max-SAT resolution, where compensation clauses for each step are represented in boxes

3. Motivation

Two reasons are generally invoked to explain the efficiency of the learning schemes: they limit the growing of the formula size and produce small compensation clauses which are more likely to be used by SUP. However, experimental analysis shows that increasing learning slightly may deteriorate significantly the performance of the solver [16]. The reasons cited above cannot explain this behavior. Furthermore, the empirical study conducted in [16] shows a correlation between the decrease of the number of propagations, the decrease of the number of detected ISs and the increase of the number of decisions. More specifically, Max-SAT resolution transformations may result in a "fragmentation" of the information contained in the original clauses of the formula. This phenomenon,

showcased in Example 4, occurs when clauses are fragmented into two (or more) clauses after transformation by Max-SAT resolution which may obstruct their exploitation by the SUP mechanism. In this case, the original information can only be retrieved by applying Max-SAT resolution between the fragmented clauses.

Example 4. *We consider the IS* $\psi = \{\{x_1\}, \{x_2\}, \{x_3\}, \{\overline{x_3}, \overline{x_4}\}, \{\overline{x_1}, \overline{x_2}, x_4\}\}$*, introduced in Example 1, detected by the implication graph* G *in Figure 1 and whose Max-SAT resolution transformation with respect to the reverse order of propagation, corresponding to the sequence* $S = \langle x_4, x_3, x_2, x_1\rangle$*, is represented on the right in the same figure. If the unique neighbor of* x_1 *in the implication graph is set to true in the transformed IS, we obtain* $\Theta(\psi, S)|_{\{x_4\}} = \{\{x_1, \overline{x_3}\}, \{x_1, x_3\}, \{x_1, x_2\}, \{\overline{x_1}, x_2, \overline{x_3}\}, \{\overline{x_1}, x_2, x_3\}, \square\}$*. Clearly, the literal* x_1 *can't be propagated in* $\Theta(\psi, S)|_{\{x_4\}}$*. We can produce the resolvent* x_1 *if we perform a Max-SAT resolution step between the clauses* $\{x_1, x_3\}$ *and* $\{x_1, \overline{x_3}\}$ *but the SUP mechanism alone cannot ensure the propagation of this literal in the transformed IS even with respect to its neighborhood in the implication graph. We say that the information leading to the propagation of* x_1 *was fragmented into several compensation clauses.*

When such transformations are learned, it may affect the capability of the SUP mechanism to detect ISs in the lower nodes of the search tree. Thus, the LB estimation may be less accurate and the solvers may explore more nodes.

4. A New Characterization of Max-SAT resolution Transformations

In this section, we introduce the notion of UP-resilience of a transformation and we explain how this property quantifies the impact of Max-SAT resolution on the SUP mechanism and, thus, on the detection of Inconsistent Subsets. We also prove that the main patterns (P1), (P2) and (P3) used in state-of-the-art BnB solvers for Max-SAT verify this property. Finally, we show that the application order of Max-SAT resolution has an impact on the UP-resilience of a transformation.

4.1. UP-Resilience

When fragmentation occurs, the compensation clauses which may propagate a literal of the constructed implication graph can contain additional literals which are not in its initial neighborhood. More precisely, the neighborhood of a literal l in the implication graph contain literals that are in direct interaction with it and which should enable, once propagated in the detected IS, the propagation of l. However, if after the transformation of the IS, the fragmentation occurs, new literals that interact directly with l are introduced and may thus obstruct the propagation of l by SUP even considering all the literals appearing in its neighborhood in the implication graph. Clearly, the power of SUP is depleted in such cases. Thus, to detect if a transformation is not affected by the fragmentation phenomenon, we can rely on the capability of SUP to propagate the literals of the constructed implication graph when their neighborhood literals are set to true. From here on, we will say that a literal l can be propagated in a formula Φ if the unit clause $\{l\}$ can be inferred from Φ by unit propagation.

Definition 6 (UP-Resilience in an Implication Graph)**.** *Let Φ be a CNF formula, ψ an IS detected by the propagation steps described by the implication graph $G = (V, A)$ and S a sequence of variables appearing in ψ. We say that the transformation $\Theta(\psi, S)$ is UP-resilient for $l \in V$ in G iff $\square \in neigh(l)$ or l can be propagated in $\Theta(\psi, S)|_{neigh(l)}$. We say that $\Theta(\psi, S)$ is UP-resilient for $L \subseteq V$ in G iff it is UP-resilient $\forall l \in L$ in G and that it is UP-resilient in G iff it is UP-resilient for V in G.*

Remark. *Note that the neighborhoods which include the special node $\square$ are not valid assignments. All transformations are considered UP-resilient for literals with such neighborhoods.*

Example 5. *We consider the IS $\psi = \{\{x_1\}, \{x_2\}, \{x_3\}, \{\overline{x_3}, \overline{x_4}\}, \{\overline{x_1}, \overline{x_2}, x_4\}\}$, introduced in Example 1, detected by the implication graph G in Figure 1. We showed in Example 4 that the fragmentation phenomenon occurs since the literal x_1 can't be propagated in $\Theta(\psi, S)|_{neigh(x_1)}$. Thus, the described transformation is not UP-resilient in the implication graph G.*

The previous definition of UP-resilience depends on the neighborhoods of the literals in the implication graph. However, the same IS can be detected by several sequences of propagation steps which can be described by distinct implication graphs.

Example 6. *We consider the IS* $\psi = \{\{x_1\}, \{x_2\}, \{x_3\}, \{\overline{x_3}, \overline{x_4}\}, \{\overline{x_1}, \overline{x_2}, x_4\}\}$*, introduced in Example 1. In addition to the implication graph presented in the original example represented on the left in Figure 3, another sequence of propagations, corresponding to the implication graph represented on the right in the same figure, can lead to the detection of the IS* ψ*. The propagation sequences corresponding to the same implication graph are considered as equivalent.*

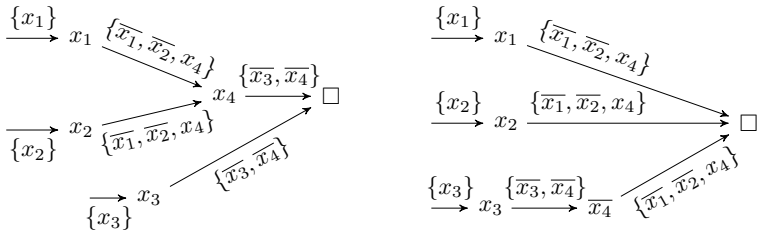


Figure 3: Implication graphs corresponding to the possible propagation sequences of ψ

In the previous definition, the UP-resilience of a transformation may therefore vary depending on how the IS has been detected since it depends on the constructed implication graph. To overcome this limitation, we propose to consider all the possible ways (i.e. sequences of propagation steps) to detect the IS. To this end, we introduce the set of the possible neighborhoods of a literal appearing in an IS.

Definition 7 (Possible Neighborhoods)**.** *Let* Φ *be a CNF formula and* ψ *an IS of* Φ*. For a literal* l *appearing in* ψ*, we define its possible neighborhoods as* $pneigh(l) = \{neigh_G(l) | G = (V, A)$ *implication graph of* ψ *s.t.* $l \in V\}$*. We extend this definition on any set of literals* L *appearing in* ψ *as* $pneigh(L) = \{\bigcup_{l \in L} neigh_G(l) | G = (V, A)$ *implication graph of* ψ *s.t.* $L \subseteq V\}$*.*

Next, we can give a general definition of UP-resilience which does not depend anymore on the propagation steps which have led to the IS discovery.

Definition 8 (UP-Resilience)**.** *Let Φ be a CNF formula, ψ an IS of Φ and S a sequence of variables appearing in ψ. The transformation $\Theta(\psi, S)$ is UP-resilient for a literal l appearing in ψ iff $\forall N \in pneigh(l)$: $\square \in N$ or l can be propagated in $\Theta(\psi, S)|_N$. We say that $\Theta(\psi, S)$ is UP-resilient if it is UP-resilient for all the literals appearing in ψ.*

Example 7. *We consider the IS $\psi = \{\{x_1\}, \{x_2\}, \{x_3\}, \{\overline{x_3}, \overline{x_4}\}, \{\overline{x_1}, \overline{x_2}, x_4\}\}$, introduced in Example 1, whose distinct implication graphs corresponding to the possible propagation sequences are represented in Figure 3. We have $pneigh(x_1) = \{\{x_4\}, \{\square\}\}$. And, as explained in Example 1, for $N = \{x_4\}$, the literal x_1 can't be propagated in $\Theta(\psi, S)|_N = \{\{x_1, \overline{x_3}\}, \{x_1, x_3\}, \{x_1, x_2\}, \{\overline{x_1}, x_2, \overline{x_3}\}, \{\overline{x_1}, x_2, x_3\}, \square\}$. Therefore, the transformation is not UP-resilient for x_1 and, consequently, it is not UP-resilient.*

Similarly, we can further quantify the impact of a Max-SAT resolution transformation of an IS ψ on SUP by considering its percentage of UP-resilience, i.e, the percentage of couples (l, N) s.t. $\square \in N$ or l can be propagated in $\Theta(\psi, S)|_N$. Finally, notice that in the case of (weighted) partial formulas, the presence of hard clauses may have a direct impact on the UP-resilience of the detected ISs. Indeed, as hard clauses are maintained after the application of weighted Max-SAT resolution, a literal propagated by a hard clause can always be propagated with respect to its neighborhood in the corresponding implication graph after the transformation of the IS.

4.2. Impact on IS Detection

One of the most interesting properties of UP-resilient transformations is the capability to retrieve the propagations which are not necessary anymore to an IS. We have seen that if a transformation is UP-resilient for a literal l, then l can be propagated in the transformed formula when the literals of each possible neighborhood, not containing the empty clause, are set to true. We now show

that this property can be extended to sets of literals, i.e., if a transformation is UP-resilient for a set of literals L, then the literals of L can be propagated in the transformed formula with respect to the possible neighborhoods of L, not containing the empty clause. This is stated more formally in the following property:

Property 1. *Let Φ be a CNF formula, ψ an IS of Φ and S a sequence of variables appearing in ψ. For any set of literals L appearing in ψ, if the transformation $\Theta(\psi, S)$ is UP-resilient for L then $\forall N \in pneigh(L) : \square \in N$ or $\forall l \in L, l$ can be propagated in $\Theta(\psi, S)|_{N \setminus \{l\}}$.*

Proof. We prove this property by induction on $|L| = n$:

- If $n = 1$, then $L = \{l\}$ and the property is verified.
- Suppose the property is true for every set of size n. Let L be of size $n+1$ and l a literal in L. We set $L' = L \setminus \{l\}$ and let $N \in pneigh(L)$. Clearly, $N = N_1 \cup N_2$ where $N_1 \in pneigh(L')$ and $N_2 \in pneigh(l)$. Moreover, since $|L'| = n$, we know by induction that $\forall N \in pneigh(L') : \square \in N$ or every literal l' in L' can be propagated in $\Theta(\psi, S)|_{N \setminus \{l'\}}$. In particular, $\square \in N_1$ or every literal l' in L' can be propagated in $\Theta(\psi, S)|_{N_1 \setminus \{l'\}}$. Also, The transformation $\Theta(\psi, S)$ is UP-resilient for L and particularly for l and thus, we have $\forall N \in pneigh(l) : \square \in N$ or l can be propagated in $\Theta(\psi, S)|_N$. In particular, $\square \in N_2$ or l can be propagated in $\Theta(\psi, S)|_{N_2}$. Thus, We have the following cases:
  - If $\square \in N_1$ or $\square \in N_2$ then $\square \in N$
  - Else, every literal l' in L' and l can be propagated respectively in $\Theta(\psi, S)|_{N_1 \setminus \{l'\}}$ and $\Theta(\psi, S)|_{N_2}$. Therefore, the clauses that ensure the propagation of every literal l' in L' in $\Theta(\psi, S)|_{N_1 \setminus \{l'\}}$ also ensure their propagation in $\Theta(\psi, S)|_{(N_1 \cup N_2) \setminus \{l'\}}$ and, similarly, the clauses that ensure the propagation of l in $\Theta(\psi, S)|_{N_2}$ also ensure its propagation in $\Theta(\psi, S)|_{(N_1 \cup N_2) \setminus \{l\}}$.

We deduce that $\forall N \in pneigh(L) : \square \in N$ or every literal l in L can be propagated in $\Theta(\psi, S)|_{N \setminus \{l\}}$.

■

When a subset ψ' of an IS ψ is not necessary anymore (for instance when ψ is not minimal), this property ensures that SUP can perform the same propagations in the transformed formula as in the original clauses of ψ'. More specifically, the UP-resilience of a set of literals L with respect to its neighborhood $neigh(L)$ is maintained individually on every literal in L.

4.3. UP-Resilience of Main Patterns

Empirical results have shown that the existing learning schemes are efficient, but until now there was no theoretical explanation to these experimental facts. Hereafter, we show that the main patterns $(P1)$, $(P2)$ and $(P3)$ used in state of the BnB solvers for Max-SAT are UP-resilient which contributes to explain theoretically the efficiency of these patterns.

Proposition 1. *Let Φ be a CNF formula, ψ an IS and $\psi' \subset \psi$ such that ψ' matches the premises of pattern (P_1). Then, the Max-SAT resolution transformation described in (P_1) is UP-resilient.*

Proof. $\psi' = \{\{l_1, l_2\}, \{l_1, \overline{l_2}\}\}$. Therefore, there are two possible propagation sequences whose implication graphs are represented in Figure 4. Since all possible neighborhoods of literals $\overline{l_1}$, l_2 and $\overline{l_2}$ contain the empty clause, the transformation of ψ' as in (P_1) with respect to the only possible variable sequence $S = < var(l_2) >$ is UP-resilient.

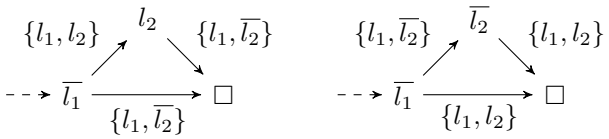


Figure 4: Implication graphs corresponding to the possible propagation sequences for an IS containing the premises of pattern (P_1)

■

Proposition 2. *Let Φ be a CNF formula, ψ an IS and $\psi' \subset \psi$ such that ψ' matches the premises of pattern (P_2). Then, the Max-SAT resolution transformation described in (P_2) is UP-resilient.*

Proof. $\psi' = \{\{l_1, l_2\}, \{l_1, l_3\}, \{\overline{l_2}, \overline{l_3}\}\}$. Therefore, there are two possible propagation sequences whose implication graphs are represented in Figure 5. There are two Max-SAT resolution application orders $S_1 = < var(l_2), var(l_3) >$ and $S_2 = < var(l_3), var(l_2) >$ that produce the same transformation described by pattern (P_2). Since all possible neighborhoods of l_2 and $\overline{l_2}$ contain the empty clause, the transformation of ψ by Max-SAT resolution is UP-resilient for l_2 and $\overline{l_2}$. We have $pneigh(\overline{l_1}) = \{\{l_3, \square\} \cup pred(\overline{l_1}), \{l_2, l_3\} \cup pred(\overline{l_1})\}$, where $pred(\overline{l_1}))$ denotes the predecessors of $\overline{l_1}$, and clearly the clause $c = \{\overline{l_1}, \overline{l_2}, \overline{l_3}\}$ propagates $\overline{l_1}$ when the literals l_2, l_3 in its second neighborhood are set to true. Also, $pneigh(l_3) = \{\{\overline{l_1}, \square\}, \{\overline{l_1}, \overline{l_2}\}\}$ and similarly the clause $c' = \{l_1, l_2, l_3\}$ propagates l_3 when the literals in its neighborhood $\{\overline{l_1}, \overline{l_2}\}$ are set to true.

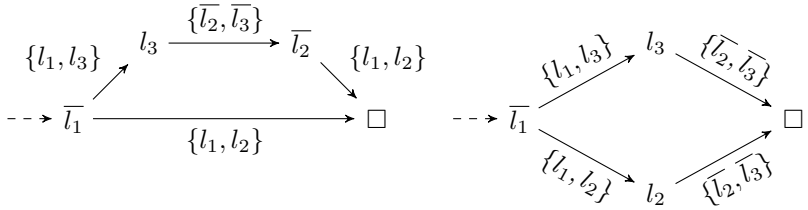


Figure 5: Implication graphs corresponding to the possible propagation sequences for an IS containing the premises of pattern (P_2)

■

Proposition 3. *Let Φ be a CNF formula, ψ an IS that matches the premises of pattern (P_3). Then, the Max-SAT resolution transformation described in (P_3) is UP-resilient.*

Proof. $\psi = \{\{l_1\}, \{\overline{l_1}, l_2\}, \{\overline{l_2}, l_3\}, ..., \{\overline{l_k}, l_{k+1}\}, \{\overline{l_{k+1}}\}\}$. Therefore, there are k possible propagation sequences whose implication graphs are represented in Figure 6. In each graph a different clause of ψ, containing the literal l_i where $1 < i < k+1$ is falsified. When $i = 1$ (resp. $i = k+1$), the unit clause $\{l_i\}$ (resp. $\{\overline{l_{k+1}}\}$) is falsified. We have $pneigh(l_i) = \{\{l_{i-1}, l_{i+1}\}, \{l_{i-1}, \square\}\}$ for $1 < i < k$ and, clearly, the clause $\{l_i, \overline{l_{i+1}}\}$, obtained after the transformation, ensures

the propagation of the literal l_i with respect to its neighborhood $\{l_{i-1}, l_{i+1}\}$. Also, for literals l_1 and l_{k+1}, we have respectively $pneigh(l_1) = \{\{l_2\}, \{\square\}\}$ and $pneigh(l_{k+1}) = \{\{l_k, \square\}\}$ and, clearly, the clause $\{l_1, \overline{l_2}\}$ ensures the propagation of the literal l_1 with respect to its neighborhood $\{l_2\}$. The same arguments ensure UP-resilience for literals $\overline{l_i}$ for $1 \leq i \leq k+1$. It is important to note that this proof on the UP-resilience of ISs matching the premises of pattern $(P3)$ is clearly valid for every Max-SAT resolution application order consistent with $(P3)$.

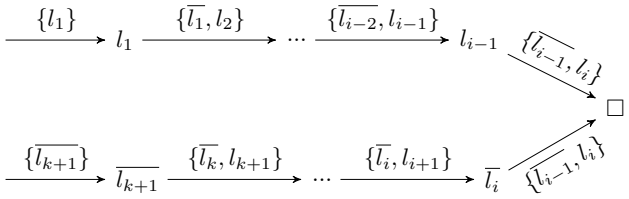


Figure 6: Implication graphs corresponding to the possible propagation sequences for an IS matching the premise of pattern $(P3)$

■

4.4. UP-Resilience and Max-SAT resolution Application Orders

An important factor that has a direct impact on the UP-resilience property is the Max-SAT resolution application order. Two orders are used in BnB solvers for Max-SAT:

- The Reverse Propagation Order (RPO): applies Max-SAT resolution steps in the reverse order of propagation and is used in most state of the art solvers.
- The Smallest Intermediary Resolvent (SIR)[22] applies the Max-SAT resolution steps according to the size of the resolvents between clauses, favoring the smallest ones.

In Example 8, we show that application orders can have a direct impact not only on the UP-resilience of certain literals appearing in the detected IS, but also on the UP-resilience of the Max-SAT resolution transformation of the IS.

Furthermore, we empirically evaluate the impact of the current orders of application of Max-SAT resolution on the UP-resilience of the transformation in Section 6.

Example 8. *We consider the IS* $\psi = \{\{x_1\}, \{x_2\}, \{x_3\}, \{\overline{x_3}, \overline{x_4}\}, \{\overline{x_1}, \overline{x_2}, x_4\}\}$*, introduced in Example 1, whose distinct implication graphs corresponding to the possible propagation sequences are represented in Figure 3. In Example 7, we showed that the transformation of* ψ *with respect to PRO is not UP-resilient since, in particular, it is not UP-resilient for literal* x_1*. Now, we consider the Max-SAT resolution transformation of* ψ *with respect to the variable sequence* $S = \langle x_3, x_4, x_1, x_2 \rangle$ *corresponding to SIR which is given in Figure 7. We have* $\Theta(\psi, S) = \{\{x_1, x_2\}, \{x_1, x_4\}, \{x_3, x_4\}, \{\overline{x_1}, x_2, x_4\}, \square\}$*. The empty clause appears in the possible neighborhoods of the literals* x_4 *and* $\overline{x_4}$*. The propagation of the literal* x_3 *with respect to its possible neighborhood* $\{\overline{x_4}\}$*, not containing the empty clause, is ensured by the clause* $\{x_3, x_4\}$*. Furthermore, the propagation of literals* x_1 *and* x_2 *with respect to their possible neighborhood* $\{x_4\}$*, not containing the empty clause, is ensured by the clauses* $\{x_1, \overline{x_4}\}$ *and* $\{\overline{x_1}, x_2, \overline{x_4}\}$*. Therefore, the transformation* $\Theta(\psi, S)$ *is UP-resilient.*

In the next section, we characterize the Max-SAT resolution transformations of UCS patterns by UP-resilience. In particular, we introduce new Max-SAT resolution application orders that ensure the UP-resilience of UCS patterns and we also show that the current orders can't ensure UP-resilience for these patterns.

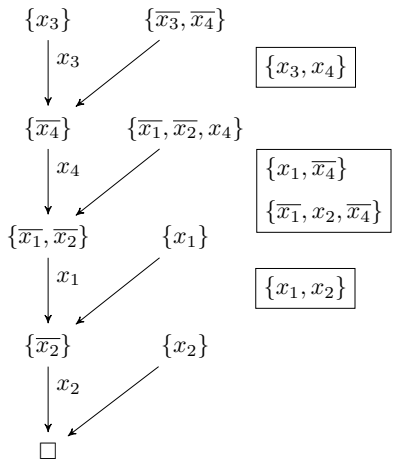


Figure 7: Implication graph corresponding to the Max-SAT resolution transformation of ψ with respect to SIR

5. On the UP-Resilience of UCSs

In this section, we prove that binary k-UCSs are UP-resilient by providing two different orders that ensure the UP-resilience of their transformation by Max-SAT resolution. We also show that unlike the given orders, the current used mechanisms can't ensure UP-resilience for these patterns which provides an explanation to the empirical results in [16] and shows that our results can help extend the current used patterns in state of the art solvers. Furthermore, we generalize our result on the resilience of k^b-UCSs to k-UCSs where all clauses are binary except one of any size that is involved in the conflict.

5.1. Implication Graphs of UCS Patterns

In this section, we establish a characterization of implication graphs of the studied k-UCSs. More specifically, we study the portion of the graph representing the propagation sequence of the detected k-UCS which is delimited by the FIUP and the empty clause. This characterization is established by the following lemma:

Lemma 1. *Let $k \geq 2$ and ψ be a k-UCS whose clauses are binary except for the conflict clause of size $s \geq 2$, recognized by the FUIP l in an implication graph G of an IS such that $|succ(l)| = s$. Then, there exists exactly s disjoint paths from l to $\square$ in G.*

Proof. Since l is a UIP, all the paths from the literals propagated by unit clauses to the conflict node in G pass through it. We have $|succ(l)| = s$. Therefore, there are at least s different paths from l to $\square$ in G. Let p_1,...,p_s be those paths. Suppose we have a different path p_{s+1} from l to $\square$. We have two possible cases:

- $|pred(\square)| \neq s$. This is absurd since the conflict clause is of size s and thus $|pred(\square)| = s$.
- Else, since $|pred(\square)| = s$, there exists $l' \neq l \in p_{s+1}$ and $i \in \{1, ..., s\}$ such that $l' \in p_i$ and $|pred(l')| > 1$. This is absurd since all clauses of ψ except the conflict clause are binary.

We deduce that there are exactly s different paths from l to $\square$ in G. The same argument of the second case ensures that these paths are disjoint.

■

As explained in Section 2.2.4, when a UCS is detected, we know that the reverse propagation order ensures the production of a unit resolvent clause after the transformation. However, in general, this is not necessarily true for all application orders. Since this is the main feature of UCS patterns, we must ensure that the introduced orders produce a unit resolvent clause. It is important to note that the condition on the successors of the FUIP in Lemma 1 ensures the production of such clause for all possible orders. In the next section, we prove the UP-resilience of binary k-UCSs. To this end, we show in the next proposition that the condition on the FUIP successors in Lemma 1 is always verified for binary k-UCSs. Later, when we generalize our result, we only consider the graphs described by Lemma 1, i.e., which verify the condition on the successors of the FUIP.

Proposition 4. *Let $k \geq 2$ and ψ be a k^b-UCS recognized by the FUIP l in an implication graph G of an IS. Then, $|succ(l)| = 2$.*

Proof. Suppose that $|succ(l)| \neq 2$. We have two possible cases:

- if $|succ(l)| > 2$ then, since $|succ(\square)| = 2$, there exists a literal with two predecessors. This is absurd since all the clauses are binary.
- if $|succ(l)| = 1$ then l is not the FUIP which is absurd.

■

5.2. On the UP-Resilience of Binary UCSs

In Section 4.3, we proved the UP-resilience of the main patterns. The next corollary is an immediate consequence of this result:

Corollary 1. *For $k \in \{2, 3\}$, k^b-UCSs are UP-resilient.*

Proof. 2^b-UCSs and 3^b-UCSs are all of the respective forms $\psi_{2^b} = \{\{l_1, l_2\}, \{l_1, \overline{l_2}\}\}$ and $\psi_{3^b} = \{\{l_1, l_2\}, \{l_1, l_3\}, \{\overline{l_2}, \overline{l_3}\}\}$ which correspond to the premises of patterns (P_1) and (P_2). Thus, we obtain the wanted result using Propositions 1 and 2. ■

In this section, we want to generalize the result of Corollary 1 to all binary k-UCSs. To this end, we introduce new orders of application of Max-SAT resolution using the characterization established in Lemma 1. We prove the resilience of k^b-UCSs with respect to these orders and we explain theoretically the inefficiency of the current orders for these patterns.

Definition 9 (Path Resolvent Order)**.** *Let $p_1 = \langle l, l_1^{p_1}, ..., l_{n_1}^{p_1}, \square\rangle (n_1 \geq 0)$ and $p_2 = \langle l, l_1^{p_2}, ..., l_{n_2}^{p_2}, \square\rangle (n_2 \geq 0)$ denote two disjoint paths from l to $\square$. The Path Resolvent Order (PRO) of p_1 and p_2 is defined as $PRO(p_1, p_2) = \langle var(l_1^{p_1}), ..., var(l_{n_1}^{p_1}), var(l_1^{p_2}), ..., var(l_{n_2}^{p_2})\rangle$.*

Theorem 1. *For any $k \geq 2$, the transformation of k^b-UCSs with respect to PRO is UP-resilient.*

Proof. Let $k \geq 2$ and ψ be a k^b-UCS recognized by the FUIP l in the implication graph G of an IS. By Lemma 1 and Proposition 4, we know that there are 2 disjoint paths from l to $\square$ in G. Let $p_1 = \langle l, l_1^{p_1}, ..., l_{n_1}^{p_1}, \square \langle (n_1 \geq 0)$ and $p_2 = \langle l, l_1^{p_2}, ..., l_{n_2}^{p_2}, \square \rangle (n_2 \geq 0)$ denote these paths in G where $n_1 + n_2 = k - 1$. And, suppose w.l.o.g that $l_{n_1}^{p_1} = l'$ is the conflict literal, i.e., the last propagated literal. We have two possible propagation sequences whose implication graphs are G and G' represented in Figure 8.

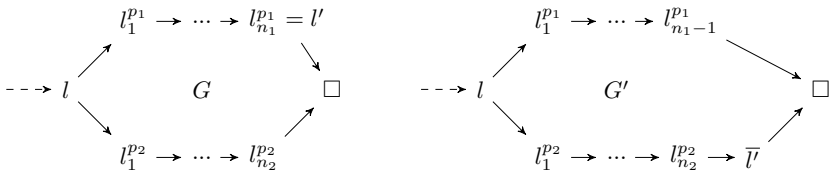


Figure 8: Implication graphs corresponding to the possible propagation sequences for k^b-UCSs.

We prove that the Max-SAT resolution transformation relatively to the order $O = PRO(p_1, p_2)$ is UP-resilient:

- The clause propagating l is not deleted after the transformation by Max-SAT resolution relatively to the order O so it clearly propagates l if its predecessors are set to true and thus the transformation by Max-SAT resolution relatively to the order O is UP-resilient for l. This argument also applies for the literals that were involved in the propagation of l.

- All possible neighborhoods of literals $l_{n_1}^{p_1} = l'$ and $\overline{l'}$ contain the empty clause. Therefore, the transformation by Max-SAT resolution relatively to the order O is UP-resilient for l' and $\overline{l'}$.

- For $i \in \{1, 2\}$, we set $l_0^{p_i} = l$. Every literal $l_j^{p_i}$ such that $1 \leq j < n_i$ admits exactly one neighborhood $neigh(l_j^{p_1}) = \{l_{j-1}^{p_i}, l_{j+1}^{p_i}\}$ that doesn't contain the empty clause. Similarly, for $l_{n_2}^{p_2}$, we have $neigh(l_{n_2}^{p_2}) = \{l_{n_2-1}^{p_i}, \overline{l'}\}$. The Max-SAT resolution step on $var(l_j^{p_1})(1 \leq j < n_i)$ is of the form:

$$\frac{\{\bar{l}, l_j^{p_i}\}, \{\overline{l_j^{p_i}}, l_{j+1}^{p_i}\}}{\{\bar{l}, l_{j+1}^{p_i}\}, \{\bar{l}, l_j^{p_i}, \overline{l_{j+1}^{p_i}}\}, \{l, \overline{l_j^{p_i}}, l_{j+1}^{p_i}\}}$$

The clause $c = \{l, \overline{l_j^{p_i}}, l_{j+1}^{p_i}\}$ clearly ensures the propagation of literal $l_{j+1}^{p_i}$ if $l_j^{p_i} \in neigh(l_{j+1}^{p_i})$ is set to true since $\bar{l}$ is propagated by the unit resolvent clause $\{\bar{l}\}$. Also, for $j = 1$, the clause $c' = \{\bar{l}, l_1^{p_1}, \overline{l_2^{p_1}}\}$ ensures the propagation of $l_1^{p_1}$ if $l, l_2^{p_1} \in neigh(l_1^{p_1})$ are set to true. Thus, We deduce that the transformation is UP-resilient for $l_j^{p_i}$ where $1 \leq j \leq n_i$ $(j \neq n_1)$.

We conclude that the transformation of ψ by Max-SAT resolution relatively to the order O is UP-resilient.

■

Definition 10 (Path Resolvent Circular Order)**.** *Let* $p_1 = \langle l, l_1^{p_1}, ..., l_{n_1}^{p_1}, \square\rangle$ $(n_1 \geq 0)$ *and* $p_2 = \langle l, l_1^{p_2}, ..., l_{n_2}^{p_2}, \square\rangle (n_2 \geq 0)$ *denote two disjoint paths from* l *to* $\square$*. The Path Resolvent Circular Order (PRCO) of* p_1 *and* p_2 *is defined as* $PRCO(p_1, p_2) = \langle var(l_1^{p_1}), ..., var(l_{n_1}^{p_1}), var(l_{n_2}^{p_2}), ..., var(l_1^{p_2})\rangle$.

Theorem 2. *For any* $k \geq 2$*, the transformation of* k^b*-UCSs with respect to PRCO is UP-resilient.*

Proof. Let $k \geq 2$ and ψ be a k^b-UCS recognized by the FUIP l in the implication graph G of an IS. By Lemma 1 and Proposition 4, let $p_1 = \langle l, l_1^{p_1}, ..., l_{n_1}^{p_1}, \square\rangle (n_1 \geq 0)$ and $p_2 = \langle l, l_1^{p_2}, ..., l_{n_2}^{p_2}, \square\rangle (n_2 \geq 0)$ denote the two disjoint paths from l to $\square$ in G where $n_1 + n_2 = k - 1$. And, suppose w.l.o.g that $l_{n_1}^{p_1} = l'$ is the conflict literal. We have two possible propagation sequences whose implication graphs are G and G' represented in Figure 8. We prove that the Max-SAT resolution transformation relatively to the order $O = PRCO(p_1, p_2)$ is UP-resilient:

- The same arguments in the proof of Theorem 1 ensure the UP-resilience of the transformation respectively to O for $l_j^{p_1} (1 \leq j \leq n_1)$ and $\overline{l'}$ as well as l and all the literals involved in its propagation.

- Every literal $l_j^{p_2}$ such that $1 \leq j \leq n_2$ admits exactly one neighborhood $neigh(l_j^{p_2}) = \{l_{j-1}^{p_2}, l_{j+1}^{p_2}\}$ that doesn't contain the empty clause (we set $l_0^{p_2} = l$ and $l_{n_2+1}^{p_2} = \overline{l'}$). The Max-SAT resolution step on $var(l_j^{p_2})$ $(j \neq 1)$ is of the form:

$$\frac{\{\bar{l}, \overline{l_j^{p_2}}\}, \{l_j^{p_2}, \overline{l_{j-1}^{p_2}}\}}{\{\bar{l}, \overline{l_{j-1}^{p_2}}\}, \{\bar{l}, \overline{l_j^{p_2}}, l_{j-1}^{p_2}\}, \{l, l_j^{p_2}, \overline{l_{j-1}^{p_2}}\}}$$

The clause $c = \{l, l_j^{p_2}, \overline{l_{j-1}^{p_2}}\}$ clearly ensures the propagation of literal $l_j^{p_2}$ when $l_{j-1}^{p_2} \in neigh(l_j^{p_2})$ is set to true since $\bar{l}$ is propagated by the unit resolvent clause $\{\bar{l}\}$. Also, the clause $c' = \{\bar{l}, \overline{l_2^{p_2}}, l_1^{p_2}\}$, generated by the Max-SAT resolution step on $var(l_2^{p_2})$, clearly ensures the propagation of $l_1^{p_2}$ when its neighbors $l, l_2^{p_2} \in neigh(l_1^{p_2})$ are set to true. Thus, the transformation is UP-resilient for $l_j^{p_2}$ where $1 \leq j \leq n_2$.

We conclude that the transformation by Max-SAT resolution relatively to the order O is UP-resilient.

■

There is a major difference between the orders we introduced. Indeed, PRCO ensures a linear input resolution transformation, i.e., at each intermediary Max-SAT resolution step we use the resolvent obtained in the previous step and a clause from the detected k^b-UCS. This is not always the case for PRO. The following result is an immediate consequence of either Theorem 1 or 2.

Corollary 2. *For any $k \geq 2$, k^b-UCSs are UP-resilient.*

Empirical results show that 2^b-UCSs and 3^b-UCSs, which correspond respectively to the patterns $(P1)$ and $(P2)$ have a positive impact on the performance of BnB solvers for Max-SAT [16, 11]. The result in corollary 1 obtained through properties 1 and 2 prove that 2^b-UCSs and 3^b-UCSs are UP-resilient for any given order of application of Max-SAT resolution which explains why learning them has a positive impact regardless of the chosen order. This is not the case for k^b-UCSs when $k > 3$. Empirical studies on the AHMAXSAT solver in [16] show that learning 4^b-UCSs and 5^b-UCSs has a major negative impact on its performance. This can be explained by the inadequacy of the Max-SAT resolution application orders used in state of the art BnB solvers for k^b-UCSs when $k > 3$. More specifically, the SIR heuristic becomes practically unusable since all the intermediary resolvents have the same size (binary) as shown in the

proofs of Theorems 1 and 2, whereas the Reverse Propagation Order doesn't always ensure the UP-resilience of the transformation as shown in the following example on a 4^b-UCS which can be easily extended to any k^b-UCS for $k > 4$.

Example 9. *We consider the IS* $\psi = \{\{l\}, \{\bar{l}, l_1\}, \{\bar{l}, l_2\}, \{\overline{l_1}, l_3\}, \{\overline{l_2}, \overline{l_3}\}\}$ *detected by one of the possible implication graphs represented on the left in Figure 9 after the respective propagation of literals* l_1, l_2 *and* l_3 *(or* $\overline{l_3}$*). Clearly, the subset* $\psi' = \{\{\bar{l}, l_1\}, \{\bar{l}, l_2\}, \{\overline{l_1}, l_3\}, \{\overline{l_2}, \overline{l_3}\}\} \subset \psi$ *is a* 4^b*-UCS recognized by the FUIP* l*. The Max-SAT resolution transformation of* ψ' *with respect to RPO which corresponds to the variable sequence* $S = \langle var(l_3), var(l_2), var(l_1)\rangle$ *is represented on the right in Figure 9.*

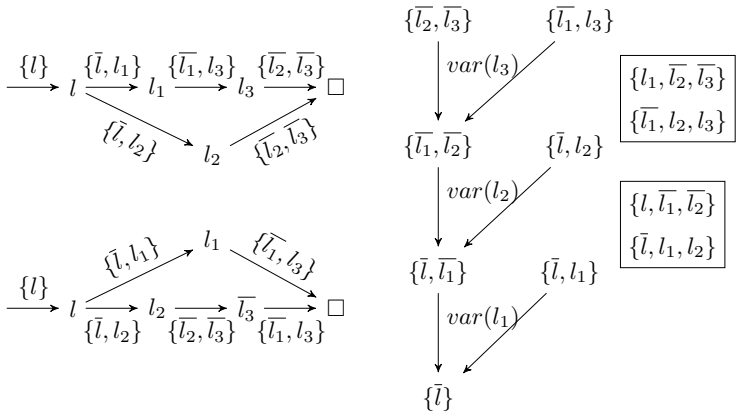


Figure 9: Implication graphs corresponding to the possible propagation sequences of ψ in Example 9 and the application of Max-SAT resolution steps relatively to RPO

The literal l_1 *has one neighborhood* $neigh(l_1) = \{l, l_3\}$ *that doesn't contain the empty clause. Clearly, the literal* l_1 *can't be propagated in* $\Theta(\psi, S)|_{neigh(l_1)} = \{\{l_1, \overline{l_2}\}, \{l_1, l_2\}\}$*. Similarly, the fragmentation phenomenon also occurs for* l_2 *and we conclude that the transformation of* ψ' *relatively to RPO is not UP-resilient.*

5.3. Generalization to UCSs

In this section, we generalize our result to k-UCSs where all clauses are binary except one of any size that is involved in the conflict when the implication

graph corresponds to the characterization established in Lemma 1. A clause involved in the conflict is either the falsified clause or contains the conflict literal, i.e., the last propagated literal. Unfortunately, although PRCO has the advantage of ensuring a linear input transformation, we couldn't generalize it to obtain the wanted result. Nevertheless, we managed to prove our result using a generalization of PRO to a multitude of paths.

Definition 11 (Multiple Path Resolvent Order)**.** *Let* $s \geq 2$ *and* $p_1 = \langle l, l_1^{p_1}, ..., l_{n_1}^{p_1}, \square \rangle, ..., p_s = \langle l, l_1^{p_2}, ..., l_{n_s}^{p_s}, \square \rangle$ *denote* s *disjoint paths from* l *to* $\square$*. The Multiple Path Resolvent Order (MPRO) of* $p_1, ..., p_s$ *is defined inductively on* s *as follows:*

- *If* $s = 2$, $MPRO(p_1, p_2) = PRO(p_1, p_2)$
- *Else* $MPRO(p_1, ..., p_s) = PRO(\langle l, MPRO(p_1, ..., p_{s-1}), \square \rangle, p_s)$.

Theorem 3. *Let* $k \geq 2$ *and* ψ *be a* k*-UCS whose clauses are binary except for the conflict clause* c *of size* $|c| = s \geq 3$*, recognized by the FUIP* l *in the implication graph* G *of an IS such that* $|succ(l)| = s$*. The transformation of* ψ *with respect to MPRO is UP-resilient.*

Proof. We suppose w.l.o.g that $c = \{\overline{l_1}, ..., \overline{l_s}\}$. By Lemma 1, there are exactly s disjoint paths $p_1 = \langle l, l_1^{p_1}, ..., l_{n_1}^{p_1}, \square \rangle, ..., p_s = \langle l, l_1^{p_s}, ..., l_{n_s}^{p_s}, \square \rangle$ from l to $\square$ in the implication graph G, represented in Figure 10, such that $\sum_{i=1}^{s} n_i = k - 1$ and $l_{n_i}^{p_i} = l_i$ for $i \in \{1, ..., s\}$. Other than G, there are exactly $\binom{s-1}{s} = s$ possible implication graphs all similar to the graph G' represented in Figure 10.

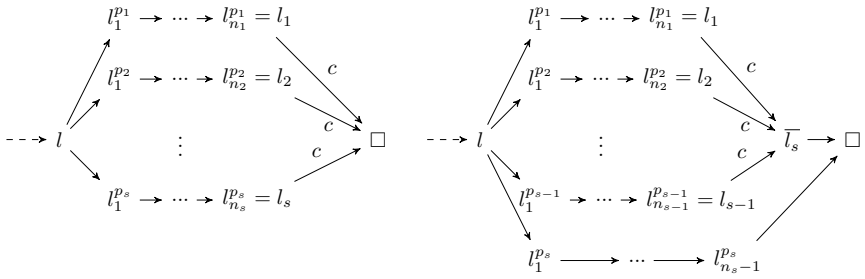


Figure 10: Implication graphs corresponding to the possible propagation sequences for k-UCSs with binary clauses except for the conflict clause

We prove that the Max-SAT resolution transformation relatively to the order $O = MPRO(p_1, ..., p_s)$ is UP-resilient:

- The same arguments in the proof of Theorem 1 ensure the UP-resilience of the transformation respectively to O for $l_j^{p_i}$ where $1 \leq i \leq s$ and $1 \leq j < n_i$ as well as l and all the literals involved in its propagation. Furthermore, all the neighborhoods of literals $\overline{l_1}, ..., \overline{l_s}$ contain the empty clause.

- For $i \in \{1, ..., s\}$, $\forall N \in pneigh(l_{n_i}^{p_i})$ $(n_i > 1)$ s.t $\square \notin N$, we have $l_{n_i-1}^{p_i} \in N$ (exists since $n_i > 1$). Clearly, the clause $c = \{l, \overline{l_{n_i-1}^{p_i}}, l_{n_i}^{p_i}\}$ obtained by the application of Max-SAT resolution on $var(l_{n_i-1}^{p_i})$ ensures the propagation of $l_{n_i}^{p_i}$ in any of these neighborhoods when $l_{n_i-1}^{p_i}$ is set to true since $\bar{l}$ is propagated by the unit resolvent clause $\{\bar{l}\}$. We deduce that the transformation relatively to the order O is UP-resilient for $l_{n_i}^{p_i}$ where $1 \leq i \leq s$ and $n_i > 1$.

- We still need to prove the UP-resilience of the transformation for literals $l_{n_i}^{p_i} = l_i$ when $n_i = 1$, with respect to their possible neighborhoods $\{l, \overline{l_j}\}$ for $j \in \{1, .., s\} \setminus \{i\}$ not containing the empty clause. For this end, we prove by induction on $|c| \geq 3$ that the compensation clauses produced by the Max-SAT resolution steps on $var(l_1), ..., var(l_s)$ ensure the propagation of each literal l_i if we consider the neighborhoods as mentioned above. For simplification, in the first Max-SAT resolution step, we replace c by the clause $c' = \{\bar{l}, \overline{l_1}, ..., \overline{l_s}\}$. This doesn't affect our result since we only omit a single clause containing the literal l:

  - If $|c| = 3$, $c = \{\overline{l_1}, \overline{l_2}, \overline{l_3}\}$. The Max-SAT resolution steps are represented on the left in Figure 11 and we can easily check that the compensation clauses ensure the propagation of the literals l_i, for $1 \leq i \leq 3$, if we consider the neighborhoods mentioned above.

  - Suppose the property is true for any clause of size $s \geq 3$. Let $c = \{l_1, ..., l_{s+1}\}$ of size $s + 1$. The first Max-SAT resolution step is represented on the right in Figure 11. The resolvent clause is

$\{\bar{l}, l_2, ..., l_{s+1}\}$ and if we consider $c' = \{l_2, ..., l_{s+1}\}$ of size s we ensure by induction the propagation of any literal l_i where $2 \leq i \leq s+1$ with respect to the neighborhoods $\{l, \overline{l_j}\}$ for $j \in \{2, .., s+1\} \setminus \{i\}$. Thus, each compensation clause $cc_k = \{\bar{l}, l_1, \overline{l_2}, ..., \overline{l_k}, l_{k+1}\}$ for $k \in \{1, ..., s\}$ ensures the propagation of literal l_1 with respect to the neighborhood $\{l, \overline{l_{k+1}}\}$ since by induction the propagation of literals $l_2, ..., l_k$ is ensured in the same neighborhood. Now, we prove by induction on $k \in \{1, ..., s\}$ that the clause cc_k ensures the propagation of l_{k+1} with respect to the neighborhood $\{l, \overline{l_1}\}$:

* If $k = 1$, $cc_1 = \{\bar{l}, l_1, l_2\}$ clearly ensures the propagation of l_2 with respect to the neighborhood $\{l, \overline{l_1}\}$.
* Suppose for $1 \leq k' < k \leq s$, $cc_{k'}$ ensures the propagation of $l_{k'+1}$ with respect to the neighborhood $\{l, \overline{l_1}\}$. $cc_k = \{\bar{l}, l_1, \overline{l_2}, ..., \overline{l_k}, l_{k+1}\}$ clearly ensures the propagation of literal l_{k+1} with respect to the neighborhood $\{l, \overline{l_1}\}$ since by induction the propagation of $l_2, ..., l_k$ is ensured in the same neighborhood by the clauses $cc_1, ..., cc_{k-1}$.

We conclude that the transformation by Max-SAT resolution relatively to the order O is UP-resilient.

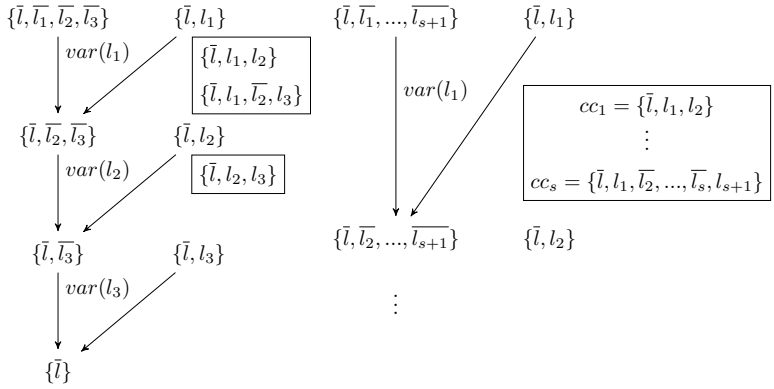


Figure 11: Application of Max-SAT resolution steps on the variables of the non binary clause c by induction on its size

■

Corollary 3. *Let $k \geq 2$ and ψ be a k-UCS whose clauses are binary except for a single clause c of size $|c| = s \geq 3$ involved in the conflict, recognized by the FUIP l in the implication graph G of an IS such that $succ(l) = s$. There exists a UP-resilient transformation of ψ.*

Proof. If c is the conflict clause then we obtain the result by Theorem 3. Else, c contains the conflict literal and the detected implication graph G has the same form as the second graph represented in Figure 10. Clearly, there is a propagation sequence where c is falsified, i.e., corresponding to an implication graph G' similar to the first graph represented in Figure 10. Thus, we deduce the UP-resilience of the transformation with respect to MPRO through the same arguments in the proof of Theorem 3.

■

The SIR order is defined relatively to the size of the intermediary resolvents. Thus, it may theoretically simulate any order when the sizes of the resolvents are the same or many different orders when many resolvents share the same size which is the case of the studied UCSs. That's why this heuristic remains practically unusable even in the generalized case. Furthermore, RPO doesn't necessarily ensure the UP-resilience of k-UCSs described in the previous corollary. We finish this section by an example that highlights this fact. This example where the non binary clause is tertiary can be easily extended to any size $s > 3$.

Example 10. *We consider the IS $\psi = \{\{l\}, \{\bar{l}, l_1\}, \{\bar{l}, l_2\}, \{\bar{l}, l_3\}, \{\overline{l_1}, l_4\}, \{\overline{l_2}, \overline{l_3}, \overline{l_4}\}\}$ (we name the tertiary clause c) detected by the first implication graph represented on the left in Figure 12 after the respective propagation of literals l_1, l_2, l_3 and l_4. In the second graph on the left in the same figure, we represent another possible propagation sequence which outlines the possible neighborhood of l_4, $neigh(l_4) = \{l_1, \overline{l_3}\}$ not containing the empty clause. Clearly, the subset $\psi' = \psi \setminus \{\{l\}\}$ is a 5-UCS recognized by the FUIP l such that c participates in the conflict and $|succ(l)| = |c| = 3$. The Max-SAT resolution transformation of ψ' with respect to RPO which corresponds to the variable sequence $S = \langle var(l_4), var(l_3), var(l_2), var(l_1)\rangle$ is represented on the right in*

Figure 12. Clearly, the literal l_4 can't be propagated in $\Theta(\psi, S)|_{neigh(l_4)} = \{\{\bar{l}\}, \{\bar{l}, \overline{l_2}\}, \{l_2, l_4\}, \{\overline{l_2}, l_4\}\}$. We conclude that the transformation of ψ' relatively to RPO is not UP-resilient.

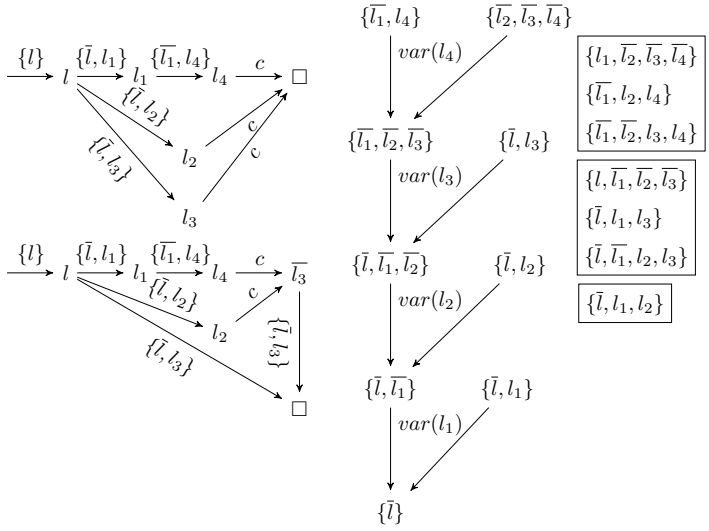


Figure 12: Implication graphs corresponding to the possible propagation sequences of ψ in Example 10 and the application of Max-SAT resolution steps relatively to RPO

6. Empirical Study on the Relevance of UP-Resilience

In this section, we present an experimental study whose aim is to provide empirical evidence on the relevance of UP-resilience. We also study the impact of the order factor and, more specifically, the current orders used in state of the art BnB solvers on the UP-resilience of Max-SAT resolution transformations.

We have implemented a simple procedure to compute the percentage of UP-resilience of the transformations. We generate all the possible neighborhoods of the literals and then we check if each literal can be propagated when each of its neighborhoods is satisfied. This naive implementation is time-consuming and increases the solving time of 25% in average. Its purpose is to evaluate the transformation of ISs as well as the learning mechanisms, not to be competitive. The implementation is performed in the solver ahmaxsat [8], which was ranked first in several categories during the 2014, 2015 and 2016 Max-

SAT Evaluations[1]. Tests are performed on unweighted and weighted, random and crafted instances of previous Max-SAT Evaluations. Instances are divided into three categories: unweighted Max-SAT (denoted ms), partial Max-SAT (denoted pms) and weighted partial Max-SAT (denoted wpms). BnB Max-SAT solvers are not competitive on industrial instances, which is why they were omitted. We have made 143856 separate runs of the solver which, in our opinion, is sufficient to show the impact of the studied components on the solver behavior. All the experiments are performed on machines equipped with Intel Xeon 2.4 Ghz processors and 24 Gb of RAM and running under a GNU/Linux operating system. The cutoff time is fixed to 1800 seconds per instance.

6.1. Learning Schemes and UP-Resilience

In the first set of experiments, we evaluate the impact of different learning schemes on the behavior of BnB Max-SAT solvers. We consider the following variants:

- ahmaxsat^{IRS}: uses the learning scheme of MiniMAXSAT [9]. A transformation is learned if all the intermediary resolvents contain less than four literals.
- ahmaxsat^{PAT}: learns the transformations when the ISs match the three main patterns presented in Section 2.2.4.
- ahmaxsat^{PAT+}: learns the transformations when the ISs match the main patterns or the extended set of patterns including UCSs, presented in Section 2.2.4.
- ahmaxsat^{UPR}: learns only UP-resilient transformations of ISs.

Table 1 shows the obtained results. For each variant, the columns $S(T)$, D and L represent respectively the number of solved instances with the average solving time, the average number of decisions and the percentage of learned

[1] https://maxsat-evaluations.github.io/2019/history.html

transformations. Columns marked with a star consider only the instances solved by all solver variants.

We first observe that the IRS learning scheme behaves very differently from PAT and PAT+. It learns in average 73,9% of the transformations while PAT and PAT+ learn less than 25% of them. However, The average percentage of UP-resilience of the transformations (not represented in the table) learned by the IRS learning scheme is relatively low (79%) compared to PAT and PAT+ (resp. 100% and 98%). Furthermore, the solver explores more than three times (resp. four times) more nodes of the search tree with the IRS learning scheme than with PAT (resp. PAT+). This is clearly due to the fact that SUP detects less ISs with the IRS learning scheme and, thus, provides a worse estimation of LB compared to PAT and PAT+ which leads to less cuts in the search tree. Consequently, ahmaxsat^{IRS} solves much less instances (at least 367 less instances than the other variants) with a higher average solving time for shared instances (at least 138 seconds more than the other variants). From these results, we can conclude that the IRS learning scheme, as it is implemented in ahmaxsat^{IRS}, does not control efficiently the impact of the transformations on the SUP mechanism.

Secondly, the comparison of the PAT and PAT+ learning schemes shows a slight increase of the percentage of learned transformations while keeping a high percentage of UP-resilience (resp. 100% and 98%). Consequently, the average number of decisions as well as the average solving time are slightly reduced. It is also interesting to observe that the percentage of transformations learned with the PAT+ learning scheme is lower than the one obtained with the UPR learning scheme. This clearly shows that the current patterns do not allow the detection of all UP-resilient transformations. This is also coherent with our theoretical study in Section 5. Finally, the UPR learning scheme outperforms all the other schemes (1407 instances solved in 58.27 seconds on average) when we don't consider the computation time required for checking the UP-resilience of the transformations which provides empirical evidence on the relevance of this property.

Instance classes			#	ahmaxsat$_{JRS}$			ahmaxsat$_{PAT}$			ahmaxsat$_{PAT+}$			ahmaxsat$_{UPR}$		
				S(T*)	D*	L*	S(T*)	D*	L*	S(T*)	D*	L*	S(T*)	D*	L*
ms	crafted	bipartite	100	1(1678,2)	2476529	76,3%	100(25,82)	7111	6,3%	100(24,32)	7159	6,2%	100(23,47)	6873	6,3%
		maxcut	67	55(100,14)	542060	90,3%	56(78,56)	244495	28,5%	56(50,07)	173361	36,3%	56(45,57)	162782	41,5%
		set-covering	10	0(-)	-	-	0(-)	-	-	0(-)	-	-	0(-)	-	-
	random	highgirth	82	3(1376,89)	5821506	72,3%	6(1154,65)	4015493	27,1%	6(1146,41)	4116346	28,2%	6(1140,21)	4120315	31,1%
		max2sat	100	16(732,36)	1384544	76,7%	100(8,37)	3321	11,9%	100(7,9)	3363	13,1%	100(7,73)	3150	12,3%
		max3sat	100	66(543,87)	1928894	85,9%	98(139,06)	147000	13,2%	100(111,21)	134846	17,5%	100(114,31)	134644	16,9%
		min2sat	96	87(109,92)	191822	76,1%	96(1,21)	414	17,0%	96(1,13)	410	17,0%	96(1,2)	415	16,9%
pms	crafted	frb	25	5(165,63)	639395	86,4%	5(430,65)	1364465	39,9%	5(181,58)	585949	49,4%	5(156,59)	521263	52,3%
		job-shop	3	0(-)	-	-	0(-)	-	-	0(-)	-	-	0(-)	-	-
		maxclique	158	132(24,58)	43769	94,4%	133(25,93)	24506	61,5%	132(28,69)	19382	66,9%	134(17,97)	17486	67,3%
		maxone	140	107(26,45)	157230	73,3%	107(38,6)	148070	12,6%	107(38,19)	133747	12,8%	107(23,75)	120250	27,3%
		min-enc/kbtree	42	15(529,73)	533014	53,2%	30(128,35)	49425	3,8%	30(142,41)	52920	4,0%	32(109,88)	46916	7,3%
		pseudo/miplib	4	2(0,08)	2528	94,9%	2(0,03)	268	25,4%	2(0,03)	288	31,1%	2(0,04)	465	39,5%
		reversi	44	7(93,3)	4619	29,2%	7(95,73)	4210	12,4%	7(99,57)	4067	14,3%	7(87,02)	4053	17,3%
		scheduling	5	0(-)	-	-	0(-)	-	-	0(-)	-	-	0(-)	-	-
	random	min2sat	60	24(441,42)	112557	63,7%	56(13,06)	1241	5,4%	55(14,94)	1241	5,4%	56(13,2)	1247	5,4%
		min3sat	60	21(670,79)	710507	72,7%	58(51,87)	16959	13,6%	58(52,07)	15453	13,6%	58(47,97)	15455	14,0%
		pmax2sat	60	59(126,18)	101463	69,7%	60(4,16)	987	18,5%	60(4,13)	962	20,4%	60(4,15)	975	19,7%
		pmax3sat/hi	30	29(291,09)	958427	79,3%	30(72,14)	74046	11,6%	30(67,64)	69971	13,9%	30(68,68)	72809	13,8%
wpms	crafted	auction	40	39(216,7)	296973	82,4%	40(145,56)	478389	35,2%	40(166,79)	316119	42,1%	40(142,71)	271838	58,9%
		CSG	10	1(710,27)	85274	99,7%	4(773,04)	84860	25,4%	4(859,04)	84860	25,4%	4(578,74)	82248	32,4%
		frb	34	14(55,59)	231672	89,7%	14(145,88)	492276	44,3%	14(101,81)	212369	53,0%	14(61,69)	188998	55,8%
		min-enc	74	51(136,27)	20317	29,7%	54(99,2)	19591	6,1%	54(109,78)	18615	6,2%	54(89,21)	63529	13,8%
		pseudo/miplib	12	1(4,97)	43	-	3(1,47)	37	-	3(1,47)	37	-	3(1,26)	35	-
		ramsey	15	4(145,77)	164184	81,3%	4(65,92)	158529	53,2%	4(69,42)	157559	54,6%	4(65,62)	162982	56,9%
	random	random-net	32	2(627,37)	1018377	74,2%	1(1686,64)	1253990	3,4%	3(374,2)	335161	9,1%	3(656,23)	1354993	25,1%
		set-covering	45	10(976,49)	4695	9,7%	10(954,54)	3655	4,2%	10(1013,31)	3655	4,2%	10(936,04)	3655	4,2%
		wmaxcut	48	42(52,4)	42579	88,5%	46(103,42)	58039	23,2%	46(56,47)	29387	36,7%	46(44,16)	25809	42,3%
	random	wmax2sat	120	81(492,51)	97322	63,0%	120(24,82)	1902	9,0%	120(23,89)	1736	10,5%	120(23,61)	1816	9,4%
		wmax3sat	40	39(253,31)	151542	81,7%	40(146,39)	38306	7,7%	40(131,51)	34237	11,7%	40(128,51)	34635	12,0%
		wpmax2sat	90	90(63,35)	4023	60,0%	90(7,17)	575	16,7%	90(7,23)	575	18,1%	90(6,7)	577	17,8%
		wpmax3sat/hi	30	30(237,83)	110777	64,8%	30(125,09)	31591	5,2%	30(123,23)	30451	6,4%	30(121,34)	30906	7,2%
	Total/Average		1776	1033(210,69)	327864	73,9%	1400(72,51)	95222	21,5%	1402(65,94)	72683	24,5%	1407(58,27)	71279	27,6%

Table 1: Impact of learning schemes on the efficiency of BnB Max-Sat solvers

6.2. Orders and UP-Resilience

In the second set of experiments, we evaluate the impact of the current orders of application of Max-SAT resolution on the UP-resilience of the transformation. We compare two variants:

- ahmaxsat^{RPO}: applies the Max-SAT resolution steps in reverse propagation order (RPO)
- ahmaxsat^{SIR}: uses the Smallest Intermediary Resolvent (SIR) heuristic described in [22].

The results are presented in Table 2 where UPR* refers to the average percentage of UP-resilience of the Max-SAT resolution transformations of the detected ISs for instances solved by both variants. We observe that the average percentage of UP-resilience is significantly higher with the SIR heuristic (64.8% compared to 57.1% for RPO). Consequently, less decisions are necessary to solve the instances (102,088 compared to 123,028 on average for RPO) and the solving time is reduced (by 27 seconds on average). This shows the importance of the Max-SAT resolution application order factor which has a direct impact on the UP-resilience of the transformations and, thus, on the solver performance. In our theoretical study in Section 5, we also provide an explanation for the seemingly low average percentage of UP-resilience (less than 65% for both orders) by showing that they don't necessarily ensure UP-resilience of certain patterns.

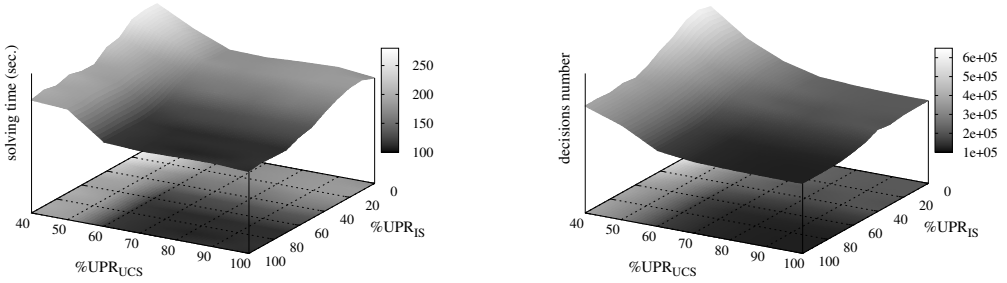
6.3. Impact of UP-Resilience

In the last set of experiments, we have implemented a new learning scheme based on the minimum percentage of UP-resilience allowed in the learned transformations of ISs (%UPR$_{\text{IS}}$) and UCSs (noted %UPR$_{\text{UCS}}$). We have tested this new learning scheme with %UPR$_{\text{IS}}$ and %UPR$_{\text{UCS}}$ ranging respectively from 0 to 100 and from 40 to 100. Results are presented in Fig. 13. We observe that the best performance is achieved with %UPR$_{\text{IS}}$ and %UPR$_{\text{UCS}}$ ranging from 60% to 100%. On these ranges of values, the average percentage of UP-resilience of the learned transformations is always higher than 90%. With values lower than

Instance classes			#	ahmaxsat^{SIR}			ahmaxsat^{RPO}		
				S(T*)	D*	UPR*	S(T*)	D*	UPR*
ms	crafted	bipartite	100	100(117,58)	34915	45,2%	100(136,44)	40608	41,6%
		maxcut	67	56(55,82)	171380	79,4%	56(64,55)	208687	73,1%
		set-covering	10	0(-)	-	-	0(-)	-	-
	random	highgirth	82	6(1017,82)	3745740	68,8%	2(1349,10)	4198100	57,9%
		max2sat	100	100(115,48)	45444	54,4%	100(157,59)	60277	48,3%
		max3sat	100	100(290,07)	328030	64,2%	99(347,74)	408764	54,5%
		min2sat	96	96(3,27)	1046	55,3%	96(3,93)	1188	51,6%
pms	crafted	frb	25	5(181,58)	585949	84,5%	5(184,82)	609626	79,7%
		job-shop	3	0(-)	-	-	0(-)	-	-
		maxclique	158	132(29,21)	19493	85,9%	133(26,15)	18686	83,0%
		maxone	140	107(38,19)	133747	86,1%	107(43,45)	131269	75,3%
		min-enc/kbtree	42	30(191,23)	57989	70,7%	20(661,99)	164731	58,8%
		pseudo/miplib	4	2(0,03)	288	83,8%	2(0,03)	271	77,9%
		reversi	44	7(2,12)	56	59,8%	5(6,78)	69	45,8%
		scheduling	5	0(-)	-	-	0(-)	-	-
	random	min2sat	60	55(115,72)	6200	40,9%	50(171,87)	9245	33,2%
		min3sat	60	58(324,52)	85114	52,7%	57(407,59)	107054	44,1%
		pmax2sat	60	60(4,42)	1021	61,4%	60(6,09)	1349	53,8%
		pmax3sat/hi	30	30(83,47)	84944	58,9%	30(100,55)	99455	47,7%
wpms	crafted	auction	40	40(169,35)	310021	92,4%	40(204,42)	300832	80,2%
		CSG	10	4(589,24)	54813	76,1%	4(547,29)	54569	50,1%
		frb	34	14(101,81)	212369	85,2%	14(74,36)	220618	81,3%
		min-enc	74	53(54,73)	18863	68,1%	50(151,67)	8055	51,3%
		pseudo/miplib	12	2(0,08)	52	70,6%	2(1,16)	52	37,7%
		ramsey	15	4(69,42)	157559	82,9%	4(74,05)	160686	76,9%
		random-net	32	3(499,20)	391871	92,1%	2(915,15)	823476	77,0%
		set-covering	45	10(-)	-	-	0(-)	-	-
		wmaxcut	48	42(56,47)	29387	79,9%	46(60,52)	34640	73,0%
	random	wmax2sat	120	120(61,64)	3993	53,5%	120(80,77)	4794	47,5%
		wmax3sat	40	40(142,42)	36836	61,9%	40(175,07)	46786	50,4%
		wpmax2sat	90	89(7,31)	579	59,2%	90(10,59)	643	51,6%
		wpmax3sat/hi	30	30(123,23)	30451	54,3%	30(162,61)	37451	40,0%
Total/Average			1776	1395(97,02)	80036	64,8%	1364(128,36)	93175	57,1%

Table 2: Impact of current Max-SAT resolution application orders on the UP-resilience of transformations

60%, the average number of decisions increases importantly and so does the average solving time. These results show that UP-resilience quantifies accurately the impact of the transformations on the efficiency of SUP and provide further incentive for our study in Section 5.



(a) Average solving time

(b) Average number of decisions

Figure 13: Impact of the minimum percentage of UP-resilience allowed for learning the transformations of ISs and UCSs

7. Conclusion and Future Work

We have introduced in this paper the notion of UP-resilience which quantifies the impact of Max-SAT resolution transformations on the SUP mechanism. We have shown that, according to the UP-resilience criterion, the main patterns do not have a negative impact on SUP. It contributes to explain theoretically the efficiency of the most used learning scheme which was proved only empirically until now. We also proved that k^b-UCSs are UP-resilient with respect to two different orders PRO and PRCO. Then, we generalized this result to k-UCSs where all clauses are binary except one of any size involved in the conflict. We showed that unlike our orders, the current mechanisms don't necessarily ensure UP-resilience for these patterns. Thus, our orders can help extend the current patterns used in state of the art BnB solvers. Finally, we presented an experimental study providing empirical evidence on the relevance of UP-resilience.

To our best knowledge, this is the first work in which UP-resilience is used

to characterize the transformations by Max-SAT resolution in order to decide the relevance of their application. Indeed, this can be a starting point of a new approach to extend Max-SAT resolution patterns. In our case, we chose UCS patterns because they present several advantages: the introduction of unit clauses as well as the high frequency of their apparition. We also showed the limits of the current orders of application of Max-SAT resolution. In fact, this is the first work in which the proposed orders are introduced relatively to the structure of the implication graphs representing the possible propagation sequences of an IS.

The prospects of our research include the extension of our studies to k-UCSs in general. It also opens new perspectives for finding orders of application of Max-SAT resolution that ensure UP-resilience or maximize its percentage by thoroughly studying the implication graphs corresponding to the propagation sequences of certain ISs. Another interesting prospect is extending our work on more sophisticated mechanisms that are used to detect disjoint ISs such as Generalized Unit Propagation (GUP) [23] which combines SUP whith Failed Literals (FL) [24]. For instance, if the reason of unsatisfiablity of the detected IS after applying GUP can be represented in the form of an implication graph, our results would be applicable to GUP.

Finally, BnB solvers perform poorly on structured instances such as industrial ones. In our opinion, this is mainly due to their inability to consider the structural properties of these instances in the exploration of the search space. An extended learning mechanism as described in our work may allow to guide the exploration of the search tree by using the learned information as it is done in modern SAT solvers. Thus, the UP-resilience property presented in this paper may be a significant step towards the improvement of BnB Max-SAT solvers performance on industrial instances. Also, increasing knowledge about Max-SAT resolution can be useful for SAT-based solvers, which are mainly efficient on industrial instances, as some solvers, such as EVA [6], already exploit Max-SAT resolution to transform cores returned by SAT solvers.

References

[1] A. Abramé, D. Habet, On the resiliency of unit propagation to max-resolution, in: Q. Yang, M. Wooldridge (Eds.), Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI 2015), AAAI Press, 2015, pp. 268–274.

[2] M. S. Cherif, D. Habet, Towards the characterization of max-resolution transformations of ucss by up-resilience, in: T. Schiex, S. de Givry (Eds.), Principles and Practice of Constraint Programming, Springer International Publishing, Cham, 2019, pp. 91–107.

[3] A. M. Alexey Ignatiev, J. Marques-Silva, RC2: an efficient maxsat solver, MaxSAT Evaluation 2018. JSAT 11. (2019) 53–64.

[4] J. Davies, F. Bacchus, Solving maxsat by solving a sequence of simpler sat instances, in: J. Lee (Ed.), International conference on principles and practice of constraint programming, Springer, 2011, pp. 225–239.

[5] R. Martins, V. Manquinho, I. Lynce, Open-wbo: A modular maxsat solver,, in: C. Sinz, U. Egly (Eds.), Theory and Applications of Satisfiability Testing – SAT 2014, Springer International Publishing, Cham, 2014, pp. 438–445.

[6] N. Narodytska, F. Bacchus, Maximum satisfiability using core-guided maxsat resolution, in: Proceedings of the Twenty-Eighth Conference on Artificial Intelligence (AAAI-14), AAAI Press, 2014, pp. 2717–2723.

[7] C. Ansótegui, M. L. Bonet, J. Levy, Solving (weighted) partial maxsat through satisfiability testing, in: O. Kullmann (Ed.), Theory and Applications of Satisfiability Testing - SAT 2009, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 427–440.

[8] A. Abramé, D. Habet, ahmaxsat: Description and Evaluation of a Branch and Bound Max-SAT Solver, Journal on Satisfiability, Boolean Modeling and Computation 9 (2015) 89–128.

[9] F. Heras, J. Larrosa, A. Oliveras, MiniMaxSAT: An efficient weighted max-sat solver, Journal of Artificial Intelligence Research 31 (2008) 1–32.

[10] A. Küegel, Improved exact solver for the weighted max-sat problem, in: D. L. Berre (Ed.), POS-10. Pragmatics of SAT, Vol. 8 of EPiC Series in Computing, EasyChair, 2012, pp. 15–27.

[11] C. M. Li, F. Manyà, N. O. Mohamedou, J. Planes, Resolution-based lower bounds in maxsat, Constraints 15 (2010) 456–484.

[12] C. M. Li, F. Manyà, J. Planes, New inference rules for max-sat, Journal of Artificial Intelligence Research 30 (2007) 321–359.

[13] C. M. Li, F. Manyà, J. Planes, Exploiting unit propagation to compute lower bounds in branch and bound max-sat solvers, in: P. van Beek (Ed.), Principles and Practice of Constraint Programming - CP 2005, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 403–414.

[14] M. L. Bonet, J. Levy, F. Manyà, Resolution for max-sat, Artificial Intelligence 171 (8) (2007) 606–618.

[15] F. Heras, J. Larrosa, New inference rules for efficient Max-SAT solving, in: A. Cohn (Ed.), Proceedings of the 21st National Conference on Artificial Intelligence (AAAI 2006), AAAI Press, 2006, pp. 68–73.

[16] A. Abramé, D. Habet, On the extension of learning for Max-SAT, in: U. Endriss, J. Leite (Eds.), Proceedings of the 7th European Starting AI Researcher Symposium (STAIRS 2014), Vol. 241 of Frontiers in Artificial Intelligence and Applications, IOS Press, 2014, pp. 1–10.

[17] R. Niedermeier, P. Rossmanith, New upper bounds for maximum satisfiability, Journal of Algorithms 36 (1) (2000) 63–88. doi:https://doi.org/10.1006/jagm.2000.1075.

[18] J. P. Marques-Silva, K. A. Sakallah, Grasp: A search algorithm for propositional satisfiability, Computers, IEEE Transactions on 48 (1999) 506–521.

[19] C. M. Li, F. Manyà, An exact inference scheme for minsat, in: IJCAI, 2015, pp. 1959–1965.

[20] J. Larrosa, F. Heras, S. De Givry, A logical approach to efficient Max-SAT solving, Artificial Intelligence 172 (2-3) (2008) 204–233.

[21] L. Zhang, C. F. Madigan, M. H. Moskewicz, S. Malik, Efficient conflict driven learning in a boolean satisfiability solver, in: Proceedings of the 2001 IEEE/ACM International Conference on Computer-Aided Design, ICCAD 01, IEEE Press, 2001, pp. 279–285.

[22] A. Abramé, D. Habet, Efficient application of max-sat resolution on inconsistent subsets, in: B. O'Sullivan (Ed.), Principles and Practice of Constraint Programming, Springer International Publishing, Cham, 2014, pp. 92–107.

[23] A. Kuegel, Improved exact solver for the weighted max-sat problem, in: D. L. Berre (Ed.), POS-10. Pragmatics of SAT, Vol. 8 of EPiC Series in Computing, EasyChair, 2012, pp. 15–27. doi:10.29007/38lm.

[24] C.-M. Li, F. Manyà, J. Planes, Detecting Disjoint Inconsistent Subformulas for Computing Lower Bounds for Max-SAT, in: Proceedings of the 21st National Conference on Artificial Intelligence (AAAI-06), Vol. 1, AAAI Press, 2006, pp. 86–91, event-place: Boston, Massachusetts.