Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)

An Exact Inference Scheme for MinSAT *

Chu-Min Li
Huazhong Univ. of Science and Technology
MIS, Université de Picardie Jules Verne

Felip Manyà
Artificial Intelligence Research Inst. (IIIA)
Spanish National Research Council (CSIC)

Abstract

We describe an exact inference-based algorithm for the MinSAT problem. Given a multiset of clauses ϕ, the algorithm derives as many empty clauses as the maximum number of clauses that can be falsified in ϕ by applying finitely many times an inference rule, and returns an optimal assignment. We prove the correctness of the algorithm, describe how it can be extended to deal with weighted MinSAT and weighted partial MinSAT instances, analyze the differences between the MaxSAT and MinSAT inference schemes, and define and empirically evaluate the MinSAT Pure Literal Rule.

1 Introduction

MinSAT is the problem of finding an assignment that maximizes the number of falsified clauses (or equivalently, minimizes the number of satisfied clauses) in a multiset of clauses, and MaxSAT is the problem of finding an assignment that minimizes the number of falsified clauses (or equivalently, maximizes the number of satisfied clauses). When weights are added to clauses, we refer to these problems as weighted MaxSAT/MinSAT. When there are clauses considered to be hard and clauses considered to be soft, we refer to them as (weighted) partial MaxSAT/MinSAT.

The development of MaxSAT solvers incorporating novel and powerful solving techniques [Li and Manyà, 2009], as well as the annual celebration of an evaluation of MaxSAT solvers [Argelich *et al.*, 2008], have been decisive to consolidate MaxSAT-based problem solving as a competitive alternative to solve challenging combinatorial optimization problems [Argelich *et al.*, 2011; Morgado *et al.*, 2013].

Given the success of MaxSAT, a number of researchers have started to look into MinSAT in the last five years. At first sight, one could think that the solving techniques and encodings to be used in MinSAT are very similar to the ones used in MaxSAT and, therefore, that there is no need of investigating MinSAT from a problem solving perspective. However, most of the research conducted so far indicates that they may be quite different, as well as that the performance profile of MaxSAT and MinSAT is also different for several optimization problems represented into these formalisms [Argelich *et al.*, 2014; Ignatiev *et al.*, 2014; Li *et al.*, 2012]. All these findings suggest that it is worth to investigate MinSAT, and find out whether it can be used as a generic problem solving approach for optimization problems either separately or in combination with MaxSAT. It is also worth mentioning that solving MinSAT is meaningful for both satisfiable and unsatisfiable instances, whereas solving MaxSAT is only meaningful for unsatisfiable instances.

Let us mention two examples that illustrate some differences between MaxSAT and MinSAT. The first example is about solvers: In branch-and-bound MinSAT solvers one has to compute, at every node of the search tree, an upper bound on the maximum number of clauses that can be falsified, and in branch-and-bound MaxSAT solvers one has to compute a lower bound on the minimum number of clauses that can be falsified. While the upper bound is computed using graph-based techniques, the lower bound is computed by applying unit propagation. As a result, for example in [Li *et al.*, 2012], we have that MinSAT greatly outperforms MaxSAT on combinatorial auctions and MaxClique instances using solvers sharing many implementation details like MinSatz [Li *et al.*, 2012] and MaxSatz [Li *et al.*, 2007; 2010a], despite of the fact that the encodings of such problems are almost identical.

The second example is about encodings: Given a Constraint Satisfaction Problem (CSP), one can represent the problem of determining the maximum number of constraints that can be satisfied with both MaxSAT and MinSAT encodings. Using the MaxSAT direct encoding [Argelich *et al.*, 2012], we must add one clause for every no-good, while using the MinSAT direct encoding [Argelich *et al.*, 2013], we must instead add one clause for every good. This implies, for instance, that for representing the constraint $X = Y$, we need a number of clauses linear in the domain size in MinSAT, and a quadratic number of clauses in MaxSAT. We are in the opposite situation if we want to represent the constraint $X \neq Y$, So, it seems that MaxSAT and MinSAT could be complementary in some scenarios [Argelich *et al.*, 2013].

*Research partially supported by the Generalitat de Catalunya grant AGAUR 2014-SGR-118, CSIC Intramural Project 201450E045, and the Ministerio de Economía y Competitividad project CO-PRIVACY TIN2011-27076-C03-03. The second author was supported by Mobility Grant PRX14/00195 of the Ministerio de Educación, Cultura y Deporte.

In this paper we focus on inference schemes for MaxSAT and MinSAT, and present original contributions to MinSAT. There exists a MaxSAT resolution rule (c.f. Section 4) that can also be applied to MinSAT because it preserves the number of falsified clauses when the premises of the rule are replaced by its conclusions. Besides, there exists also an exact inference algorithm for MaxSAT [Bonet *et al.*, 2007] that derives, by applying the MaxSAT resolution rule a finite number of times following a certain strategy, as many empty clauses as the minimum number of clauses that can be falsified in a multiset of clauses. Nevertheless, it is an open problem to figure out how an inference rule may be used to compute a MinSAT solution; i.e., to derive as many empty clauses as the maximum number of clauses that can be falsified.

The main objective of this work is to solve that open problem by devising an exact inference algorithm for MinSAT that relies on the existing inference algorithm for MaxSAT but contains significant differences despite of the fact that both algorithms apply the MaxSAT resolution rule. In particular, we describe the algorithm and prove its correctness, show how it can be extended to deal with weighted MinSAT and weighted partial MinSAT instances, analyze the differences between the MaxSAT and MinSAT inference schemes, and define and empirically evaluate the MinSAT Pure Literal Rule.

The paper is organized as follows: Section 2 gives basic definitions. Section 3 summarizes the related work. Section 4 presents the existing inference scheme for MaxSAT. Section 5 describes the inference algorithm for MinSAT, and proves its correctness. Section 6 compares the MaxSAT and MinSAT inference schemes, and presents new insights into MinSAT. Finally, Section 7 contains the concluding remarks.

2 Preliminaries

A literal is a propositional variable or a negated propositional variable. A clause is a disjunction of literals. A weighted clause is a pair (c, w), where c is a is a disjunction of literals and w, its weight, is a natural number or infinity. A clause is hard if its weight is infinity; otherwise it is soft. A weighted partial MinSAT (MaxSAT) instance is a multiset of weighted clauses $\phi = \{(h_1, \infty), \ldots, (h_k, \infty), (c_1, w_1), \ldots, (c_m, w_m)\}$, where the first k clauses are hard and the last m clauses are soft. For simplicity, in what follows, we omit infinity weights, and write $\phi = \{h_1, \ldots, h_k, (c_1, w_1), \ldots, (c_m, w_m)\}$. Notice that a soft clause (c, w) is equivalent to having w copies of the clause $(c, 1)$, and that $\{(c, w_1), (c, w_2)\}$ is equivalent to $(c, w_1 + w_2)$. A truth assignment assigns to each propositional variable either 0 or 1.

Weighted Partial MinSAT (MaxSAT), or WPMinSAT (WPMaxSAT), for an instance ϕ is the problem of finding an assignment in which the sum of the weights of the satisfied (falsified) soft clauses is minimal, and all the hard clauses are satisfied. The Weighted MinSAT (MaxSAT) problem, or WMinSAT (WMaxSAT), is the WPMinSAT (WPMaxSAT) problem when there are no hard clauses. The Partial MinSAT (MaxSAT) problem, or PMinSAT (PMaxSAT), is the WPMinSAT (WPMaxSAT) problem when all the soft clauses have the same weight. The (Unweighted) MinSAT (MaxSAT) problem is the Partial MinSAT (MaxSAT) problem when there are no hard clauses.

3 Related Work

The work on MinSAT can be traced back to the mid-90s in the area of approximation algorithms [Kohli *et al.*, 1994; Marathe and Ravi, 1996], but it was not until 2010 that MinSAT started to be studied from a problem solving perspective. The main results of this recent work on MinSAT may be succinctly summarized as follows:

I) Definition of transformation between MinSAT and MaxSAT: Reductions from MinSAT to PMaxSAT were defined in [Li *et al.*, 2010b], but they do not generalize to WPMinSAT. This drawback was overcome with the definition of the natural encoding [Kügel, 2012], which was improved in [Zhu *et al.*, 2012]. Reductions of WPMinSAT to Group MaxSAT were evaluated in [Heras *et al.*, 2012].

II) Development of branch-and-bound solvers: The only existing WPMinSAT solver, MinSatz [Li *et al.*, 2011; 2012], is based on MaxSatz [Li *et al.*, 2007], and implements upper bounds that exploit clique partition algorithms and MaxSAT technology.

III) Development of SAT-based solvers: There exist two WPMinSAT solvers of this class [Ansótegui *et al.*, 2012; Heras *et al.*, 2012]. The main difference with SAT-based MaxSAT solvers lies in the way of relaxing soft clauses.

IV) Definition and evaluation of genuine MinSAT encodings of relevant problems such as weighted MaxCSP [Argelich *et al.*, 2013; 2014], and graph problems [Ignatiev *et al.*, 2014; 2013].

4 Inference in MaxSAT

The classical resolution rule $x \vee A, \overline{x} \vee B \vdash A \vee B$ (where x is a variable, and A and B are disjunctions of literals) is unsound for MaxSAT. It works by adding the conclusion of the rule to the premises, and this may increase the number of falsified clauses in the derived instance. In contrast, the MaxSAT resolution rule works by replacing the premises by its conclusions in such a way that the number of falsified clauses is preserved for every assignment.

The MaxSAT resolution rule is defined as follows [Bonet *et al.*, 2007; Larrosa *et al.*, 2008]:

$$\begin{array}{l} x \vee a_1 \vee \ldots \vee a_s \\ \overline{x} \vee b_1 \vee \ldots \vee b_t \\ \hline\hline a_1 \vee \ldots \vee a_s \vee b_1 \vee \ldots \vee b_t \\ x \vee a_1 \vee \ldots \vee a_s \vee \overline{b_1} \\ x \vee a_1 \vee \ldots \vee a_s \vee b_1 \vee \overline{b_2} \\ \cdots \\ x \vee a_1 \vee \ldots \vee a_s \vee b_1 \vee \ldots \vee b_{t-1} \vee \overline{b_t} \\ \overline{x} \vee b_1 \vee \ldots \vee b_t \vee \overline{a_1} \\ \overline{x} \vee b_1 \vee \ldots \vee b_t \vee a_1 \vee \overline{a_2} \\ \cdots \\ \overline{x} \vee b_1 \vee \ldots \vee b_t \vee a_1 \vee \ldots \vee a_{s-1} \vee \overline{a_s} \end{array}$$

input: C
$C_0 := C$
for $i := 1$ **to** n
    $C := saturation(C_{i-1}, x_i)$
    $\langle C_i, D_i \rangle := partition(C, x_i)$
endfor
$m := |C_n|$
$I := \emptyset$
for $i := n$ **downto** 1
    $I := I \cup [x_i \mapsto max\_extension(x_i, I, D_i)]$
output: m, I

Figure 1: An exact inference algorithm for MaxSAT

The tautologies concluded are removed, and repeated literals in a clause are collapsed into one. We say that the rule *cuts* the variable x.

[Bonet *et al.*, 2006; 2007] proved that the MaxSAT resolution rule is sound and complete. It is sound because the number of clauses that every assignment falsifies in the premises is the same as the number of clauses that it falsifies in the conclusions. It is complete because, for every multiset of clauses C whose minimum number of falsified clauses is m, the application of the rule (finitely many times) following a strategy explained below derives a multiset formed by exactly m empty clauses.

Before describing the exact inference algorithm for MaxSAT of [Bonet *et al.*, 2007], we need two definitions:

Definition 1 *We write $\mathcal{C} \vdash \mathcal{D}$ when the multiset of clauses $\mathcal{D}$ can be obtained from the multiset $\mathcal{C}$ by applying the MaxSAT resolution rule a finite number of times. We write $\mathcal{C} \vdash_x \mathcal{D}$ when this sequence of applications only cuts the variable x.*

Definition 2 *A multiset of clauses $\mathcal{C}$ is said to be saturated w.r.t. the variable x if, for every pair of clauses $c_1 = x \vee A$ and $c_2 = \overline{x} \vee B$ of $\mathcal{C}$, after applying the MaxSAT resolution rule, there is a literal l such that l is in A and $\bar{l}$ is in B. A multiset of clauses $\mathcal{C}'$ is a* saturation of *$\mathcal{C}$ w.r.t. x if $\mathcal{C}'$ is saturated w.r.t. x and $\mathcal{C} \vdash_x \mathcal{C}'$; i.e., $\mathcal{C}'$ can be obtained from $\mathcal{C}$ by applying the inference rule cutting x finitely many times.*

Trivially, by the previous definition, a multiset of clauses $\mathcal{C}$ is saturated w.r.t. x iff every possible application of the inference rule cutting x only introduces clauses containing x due to the fact that the first clause in the conclusions is a tautology and, therefore, is eliminated.

Although every multiset of clauses is saturable, its saturation is not unique. For instance, the multiset $\{x_1,\ \overline{x}_1 \vee x_2,\ \overline{x}_1 \vee x_3\}$ has two possibles saturations w.r.t. variable x_1: the multiset $\{x_2,\ \overline{x}_2 \vee x_3,\ x_1 \vee \overline{x}_2 \vee \overline{x}_3,\ \overline{x}_1 \vee x_2 \vee x_3\}$, and the multiset $\{x_3,\ x_2 \vee \overline{x}_3,\ x_1 \vee \overline{x}_2 \vee \overline{x}_3,\ \overline{x}_1 \vee x_2 \vee x_3\}$. Nevertheless, the algorithm described below is correct independently of the saturation selected.

Figure 1 gives the pseudo-code of the exact inference algorithm for MaxSAT. Given an input multiset of clauses C with n different variables, the algorithm returns the minimum number m of clauses of C that can be falsified, and an optimal MaxSAT assignment I.

Function $saturation(C, x)$ computes a saturation of C w.r.t. x. The order in which the algorithm computes the saturation of the variables can be freely chosen; i.e., the sequence $x_1, \ldots x_n$ can be any enumeration of the variables.

Function $partition(C, x_i)$ computes a partition of C into the multiset C_i of clauses without occurrences of the variable x_i, and the multiset D_i of clauses with occurrences of x_i.

Function $max\_extension(x_i, I, D_i)$ computes a truth assignment for x_i as follows: if I satisfies all the clauses in D_i, including the case in which $D_i = \{\}$, then the function returns false (x_i is set to false); otherwise, either all the clauses of the form $x_i \vee A$ are satisfied or all the clauses of the form $\overline{x}_i \vee B$ are satisfied. In this case, x_i is set in such a way that all the clauses in D_i become satisfied.

The algorithm has two parts. In the first part, the algorithm successively saturates w.r.t. all the variables occurring in the input multiset. Once the current multiset is saturated w.r.t. the variable under consideration, say x_i, it partitions the resulting multiset into two multisets: C_i and D_i. C_i contains the clauses without occurrences of x_i, and D_i contains the clauses with occurrences of x_i. The algorithm continues saturating C_i w.r.t. one of the remaining variables, and ignores D_i. This process continues until all the variables are eliminated. At the end, C_n contains no variables, and the number of empty clauses in C_n is the returned minimum number of falsified clauses. In the second part, the algorithm builds an optimal assignment taking into account the information in D_i. Every optimal assignment must satisfy all the derived D_i's.

Example 1 *Let $\phi = \{\overline{x}_1, x_1 \vee x_2, x_1 \vee x_3, \overline{x}_3\}$. Resolving the first two clauses, we get $\{x_2, \overline{x}_1 \vee \overline{x}_2, x_1 \vee x_3, \overline{x}_3\}$. Resolving the second and third clause, we get a saturation of ϕ w.r.t. x_1: $\{x_2, \overline{x}_2 \vee x_3, \overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3, x_1 \vee x_2 \vee x_3, \overline{x}_3\}$. Hence, $C_1 = \{x_2, \overline{x}_2 \vee x_3, \overline{x}_3\}$, and $D_1 = \{\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3, x_1 \vee x_2 \vee x_3\}$.*

Resolving the first two clauses of C_1, we get $\{x_3, x_2 \vee \overline{x}_3, \overline{x}_3\}$, which is a saturation of C_1 w.r.t. x_2. Hence, $C_2 = \{x_3, \overline{x}_3\}$, and $D_2 = \{x_2 \vee \overline{x}_3\}$.

Resolving $\{x_3, \overline{x}_3\}$, we get the empty clause. Hence, $C_3 = \{\Box\}$, and $D_3 = \{\}$. So, the minimum number of falsified clauses is 1, and $x_3 \mapsto false, x_2 \mapsto false, x_1 \mapsto true$ is an optimal assignment.

5 Inference in MinSAT

The goal of an exact MinSAT algorithm is to compute an assignment that falsifies the maximum number of clauses in a MinSAT instance.

In this section we define an algorithm that derives, by applying the MaxSAT resolution rule, as many empty clauses as the maximum number of clauses that can be falsified. First of all, we should note that the MaxSAT resolution rule is also a sound inference rule for MinSAT because its application preserves the number of falsified clauses. Nevertheless, to get completeness we need a different strategy of applying the rule, and a new way of deriving an optimal assignment.

In our approach, given a variable x_i occurring in the MinSAT instance ϕ under consideration, we start by computing a saturation w.r.t. x_i. By the soundness of resolution, the number of clauses that an assignment falsifies in ϕ and in the computed saturation is the same. We then partition the saturation into two multisets as in MaxSAT: the multiset C_i of clauses without occurrences of the variable x_i, and the multiset D_i of clauses with occurrences of x_i. In the next step, in contrast to MaxSAT where a saturation of C_i w.r.t. x_{i+1} is computed, we now compute a saturation of the MinSAT instance ϕ' formed by both C_i and the multiset of clauses, say F_i, resulting of eliminating all the occurrences of the literals x_i and $\overline{x}_i$ in D_i; i.e., $F_i := \{A \,|\, x_i \vee A \in D_i\} \cup \{B \,|\, \overline{x}_i \vee B \in D_i\}$. As we show in the following lemma, the MinSAT problem for ϕ can be reduced to the MinSAT problem for $\phi' = C_i \cup F_i$.

Lemma 1 *Let ϕ be a MinSAT instance, and let ϕ' be a MinSAT instance derived from ϕ by first computing a saturation $C_i \cup D_i$ of ϕ w.r.t. a variable x_i occurring in ϕ, and then removing from D_i all the occurrences of both x_i and $\overline{x}_i$. We write $\phi' = C_i \cup F_i = C_i \cup \{A \,|\, x_i \vee A \in D_i\} \cup \{B \,|\, \overline{x}_i \vee B \in D_i\}$. If I is an optimal assignment of ϕ, then I is also an optimal assignment of ϕ' and falsifies the same number of clauses in ϕ and ϕ'.*

Proof Since every assignment falsifies the same number of clauses in ϕ and $C_i \cup D_i$ due to the soundness of resolution, we will prove that an optimal assignment I of $C_i \cup D_i$ is an optimal assignment of ϕ' and falsifies the same number of clauses in $C_i \cup D_i$ and ϕ'. We first prove that I falsifies the same number of clauses in both multisets: it suffices to prove that I falsifies a clause of the form $x_i \vee A \in D_i$ ($\overline{x}_i \vee B \in D_i$) iff I falsifies A (B), because the rest of clauses in F_i are identical in D_i.

Assume that I sets x_i to true. Then, we have three cases:
i) If I falsifies $\bar{x}_i \vee B$, then I falsifies B because it is a subclause of $\bar{x}_i \vee B$.
ii) If I fasifies B, then I falsifies $\bar{x}_i \vee B$ because we assumed that I sets x_i to true.
iii) I does not falsify neither $x_i \vee A$ nor A. I does not falsify $x_i \vee A$ because I sets x_i to true. If I falsifies A, then I satisfies all the clauses in D_i: it satisfies every clause of the form $\bar{x}_i \vee B$ because, by the definition of saturation, B contains a literal whose negation appears in A; and it satisfies every clause of the form $x_i \vee A$ because I sets x_i to true. This is in contradiction with I being optimal because the assignment obtained from I by setting x to false falsifies at least one additional clause ($x \vee A$ in D_i). So, I satisfies A.

The argument above works similarly if we assume that I sets x_i to false.

I is also an optimal assignment of ϕ' because there exists no assignment I' of ϕ' that falsifies more clauses of ϕ' than I does. If so, I' could be easily extended to an optimal assignment of ϕ by setting adequately the truth value of x_i due to the fact that no interpretation can falsify simultaneously a clause of the form A and a clause of the form B. □

Figure 2 gives the pseudo-code of the new exact inference algorithm for MinSAT proposed in this paper. Given an input multiset of clauses C with n different variables, the algorithm returns the maximum number m of clauses of C that can be falsified, and an optimal MinSAT assignment I.

Functions $saturation(C, x)$ and $partition(C, x_i)$ are defined as in the MaxSAT case. Function $min\_extension(x_i, I, D_i)$ computes a truth assignment for x_i as follows: if I falsifies a clause of D_i by setting x_i to false, then the function returns false; otherwise it returns true.

```
input: C
for i := 1 to n
    C := saturation(C, x_i)
    ⟨C_i, D_i⟩ := partition(C, x_i)
    F_i := {A | x ∨ A ∈ D_i} ∪ {B | x̄ ∨ B ∈ D_i}
    C := C_i ∪ F_i
endfor
m := |C|
I := ∅
for i := n downto 1
    I := I ∪ [x_i ↦ min_extension(x_i, I, D_i)]
output: m, I
```

Figure 2: An exact inference algorithm for MinSAT

There are two crucial differences with the exact MaxSAT algorithm of Section 4: The first one is that, after saturating w.r.t. the variable under consideration, the algorithm continues saturating using the multiset $C_i \cup F_i$ instead of the multiset C_i. The second one is the way of computing an optimal assignment.

Theorem 1 *Given an input multiset of clauses C with n different variables, the inference algorithm for MinSAT returns the maximum number m of clauses of C that can be falsified, and an optimal assignment I.*

Proof The algorithm constructs a finite sequence of multisets of clauses $\phi_1 = C_1 \cup F_1, \ldots, \phi_n = C_n \cup F_n$ from the input multiset C. By Lemma 1, the maximum number of clauses that can be falsified in each multiset $\phi_1, \ldots, \phi_n$ is the same as in the input multiset C. Since ϕ_k contains one less variable than ϕ_{k-1}, for $k = 2, \ldots, n$, we have that ϕ_n contains no variables; actually, it only contains empty clauses unless the input multiset is the empty multiset. So, the number of empty clauses in ϕ_n is the maximum number of clauses that can be falsified in ϕ_n, and also in $\phi_1, \ldots, \phi_{n-1}$ and C.

For deriving an optimal assignment we derive a sequence of assignments $I_1, \ldots, I_n$ in such a way that (i) I_k is an assignment of the variables $x_n, x_{n-1}, \ldots, x_{n-k+1}$; and (ii) I_k is derived from I_{k-1} by setting $I_k(x_j) = I_{k-1}(x_j)$ for $j = n, \ldots, n-k+2$, and $I_k(x_{n-k+1}) = false$ if this setting leads I_k to falsify a clause of D_{n-k+1}; otherwise, $I_k(x_{n-k+1}) = true$. We will prove, by induction on k, that I_k is an optimal assignment of $C_{n-k+1} \cup D_{n-k+1}$ for $k = 1, \ldots, n$, and in particular that I_n is an optimal assignment of $C_1 \cup D_1$ and, therefore, of the input multiset C.

When $k = 1$, we have that C_n is empty or only contains empty clauses, and D_n is empty or contains unit clauses either of the form x_n or of the form $\overline{x}_n$, but it cannot contain occurrences of both x_n and $\overline{x}_n$. By setting x_n in such a way that the clauses in D_n are falsified, we get an assignment that falsifies the maximum number of clauses in $C_n \cup D_n$. If D_n is empty, the value of x_n is not relevant, and we set x_n to true. So, I_1 is an optimal assignment of $C_n \cup D_n$ and ϕ_n.

Assume that I_k is an optimal assignment of $C_{n-k+1} \cup D_{n-k+1}$. We must prove that I_{k+1} is an optimal assignment of $C_{n-k} \cup D_{n-k}$. Since $C_{n-k+1} \cup D_{n-k+1}$ is a saturation of ϕ_{n-k} w.r.t. the variable x_{n-k+1} and the resolution rule is sound, I_k is also an optimal assignment of $\phi_{n-k} = C_{n-k} \cup F_{n-k}$. Since ϕ_{n-k} is obtained from $C_{n-k} \cup D_{n-k}$ by removing the literals containing x_{n-k} in D_{n-k}, an assignment cannot falsify more clauses in $C_{n-k} \cup D_{n-k}$ than in ϕ_{n-k}. Recall that an assignment cannot simultaneously falsify in F_{n-k} clauses of the form A and B, where A (B) is a clause obtained from a clause of the form $x_{n-k} \vee A$ ($\overline{x}_{n-k} \vee B$) of D_{n-k}. By forcing I_{k+1} to set x_{n-k} in such a way that the number of clauses falsified in D_{n-k} is maximized, we have that the number of falsified clauses in $C_{n-k} \cup D_{n-k}$ and ϕ_{n-k} is the same. So, I_{k+1} is an optimal assignment of $C_{n-k} \cup D_{n-k}$.

□

Example 2 *Let $\phi = \{\overline{x}_1, x_1 \vee x_2, x_1 \vee x_3, \overline{x}_3\}$ as in Example 1. Resolving the first two clauses, we get $\{x_2, \overline{x}_1 \vee \overline{x}_2, x_1 \vee x_3, \overline{x}_3\}$. Resolving the second and third clause, we get a saturation of ϕ w.r.t. x_1: $\{x_2, \overline{x}_2 \vee x_3, \overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3, x_1 \vee x_2 \vee x_3, \overline{x}_3\}$. Hence, $C_1 = \{x_2, \overline{x}_2 \vee x_3, \overline{x}_3\}$, $D_1 = \{\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3, x_1 \vee x_2 \vee x_3\}$, and $F_1 = \{\overline{x}_2 \vee \overline{x}_3, x_2 \vee x_3\}$. So, the problem reduces to find the maximum number of falsified clauses in $C_1 \cup F_1 = \{x_2, \overline{x}_2 \vee x_3, \overline{x}_2 \vee \overline{x}_3, x_2 \vee x_3, \overline{x}_3\}$.*

We now resolve the second and fourth clause, and get $\{x_2, x_3, \overline{x}_2 \vee \overline{x}_3, \overline{x}_3\}$. We resolve the first and third clause, and get a saturation of $C_1 \cup F_1$ w.r.t. x_2: $\{\overline{x}_3, x_2 \vee x_3, x_3, \overline{x}_3\}$. Hence, $C_2 = \{\overline{x}_3, x_3, \overline{x}_3\}$, $D_2 = \{x_2 \vee x_3\}$, and $F_2 = \{x_3\}$ So, the problem reduces to find the maximum number of falsified clauses in $C_2 \cup F_2 = \{\overline{x}_3, x_3, x_3, \overline{x}_3\}$.

Resolving the two first and the two last clauses of $\{\overline{x}_3, x_3, x_3, \overline{x}_3\}$, we get two empty clauses. Hence, $C_3 = \{\square, \square\}$, $D_3 = \{\}$, and $F_3 = \{\}$. So, the maximum number of falsified clauses is 2, and $x_3 \mapsto true, x_2 \mapsto true, x_1 \mapsto true$ is an optimal assignment.

6 Comparing the Inference Schemes for MaxSAT and MinSAT

A key component of the previous MaxSAT and MinSAT algorithms is the concept of saturation, and an essential difference between the two algorithms lies in the way of exploiting its properties. Saturation divides the clauses with occurrences of a variable x_i into a group of clauses of the form $x_i \vee A$ and another group of clauses of the form $\overline{x}_i \vee B$, ensuring that every assignment satisfies at least all the subclauses A or all the subclauses B due to the fact that, for every pair of clauses (A, B), there is a literal l in A such that $\bar{l}$ is in B.

In the algorithm for MaxSAT, by successively computing a saturation w.r.t. all the variables and due to the soundness of the MaxSAT resolution rule, we have that the number of clauses that an assignment falsifies in the input multiset C is the same as the number of clauses that it falsifies in $C_n \cup \bigcup_{i=1}^{n} D_i$, where C_n is either the empty multiset or only contains empty clauses because when C_n is derived all the variables have been eliminated. Since $\bigcup_{i=1}^{n} D_i$ is satisfiable, it turns out that C_n contains as many empty clauses as the minimum number of clauses that can be falsified in C, and every assignment satisfying $\bigcup_{i=1}^{n} D_i$ is an optimal MaxSAT assignment. Observe that, for finding a satisfying assignment of $\bigcup_{i=1}^{n} D_i$, we just need to set every x_i to true (false) if some subclause A (B) is falsified, because the other group is satisfied independently of the value of x_i. So, we force the algorithm to satisfy the group of falsified subclauses to get an optimal MaxSAT assignment. We refer the reader to [Bonet *et al.*, 2007] for a formal proof of the previous results.

In the algorithm for MinSAT, we exploit the set D_i obtained after computing a saturation w.r.t. the variable x_i in a different way. Since at most one of the groups of subclauses A and B is falsified, and C_i contains no occurrences of x_i, the MinSAT problem for $C_i \cup D_i$ can be safely reduced to the MinSAT problem for $C_i \cup F_i$. By setting x_i to false (true) if there is a subclause A (B) falsified, we have that every optimal MinSAT assignment of $C_i \cup F_i$ can be extended to an optimal MinSAT assignment of $C_i \cup D_i$. So, we now force the algorithm to falsify, instead of satisfy, the group of falsified subclauses to get an optimal MinSAT assignment.

Observe that reducing a multiset C to $C_i \cup D_i$ is stronger than reducing C to $C_i \cup F_i$ because C and $C_i \cup D_i$ have the same number of falsified clauses for each assignment, while C and $C_i \cup F_i$ only preserve the maximum number of falsified clauses. It can happen, for instance, that C is satisfiable and $C_i \cup F_i$ is unsatisfiable as the following example shows: Let C be the satisfiable multiset $\{x_1 \vee x_2, x_1 \vee \overline{x}_2, x_2 \vee x_3\}$. If we saturate C w.r.t. the variable x_1, we have that $C_1 \cup F_1$ is the unsatisfiable multiset $\{x_2, \overline{x}_2, x_2 \vee x_3\}$.

The SAT Pure Literal Rule (PLR) and MaxSAT PLR are identical: If all the occurrences of a propositional variable x_i in a multiset have the same polarity, then all the clauses containing the variable x_i can be removed. Nevertheless, this rule is not valid in MinSAT as the following counterexample shows: Let C be again the multiset $\{x_1 \vee x_2, x_1 \vee \overline{x}_2, x_2 \vee x_3\}$, which has a maximum of two falsified clauses by assigning x_1, x_2 and x_3 to false. By applying the previous PLR w.r.t. the variable x_1, we derive the multiset $\{x_2 \vee x_3\}$, and by applying the PLR w.r.t. the variable x_2, we derive the empty multiset that has no falsified clauses.

Lemma 2 *Let C be a MinSAT instance where all the occurrences of the propositional variable x_i have the same polarity, and let C' be the MinSAT instance resulting of eliminating all the occurrences of the variable x_i in C. Then, C and C' have the same maximum number of falsified clauses.*

Proof Since x_i occurs only with one polarity, saturating C w.r.t. the variable x_i, we get $C_i \cup D_i = C$ because in this case D_i is formed by all the clauses of C with occurrences of x_i, and C_i is formed by the rest of clauses. So, by removing all the occurrences of x_i in D_i, we get $C_i \cup F_i = C'$. By Lemma 1, C and C' have the same maximum number of falsified clauses. □

Therefore, the MINSAT PLR states that if all the occurrences of a propositional variable x_i in a MinSAT instance have the same polarity, then all the occurrences of the variable x_i can be removed.

Figure 3 compares the mean time, in seconds, needed by MinSatz to solve random Min-2SAT instances for different values of the ratio of the number of clauses to the number of variables (r) with a version of MinSatz that applies the MinSAT PLR (W), and a version of MinSatz that does not ap-

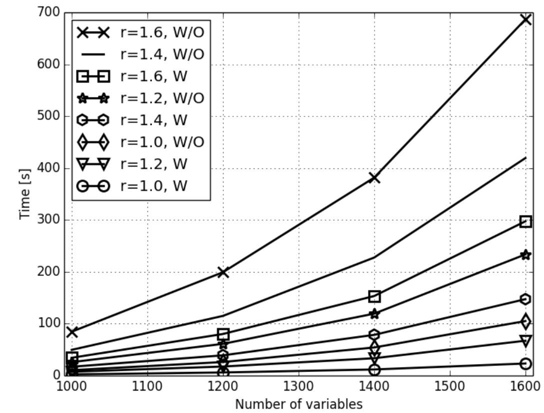

Figure 3: Impact on time of the MinSAT PLR

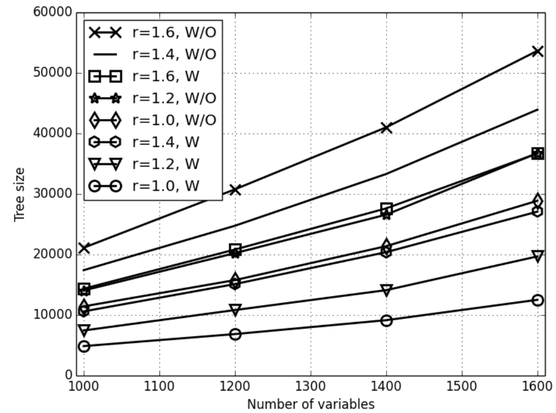

Figure 5: Impact on proof tree size of the MinSAT PLR

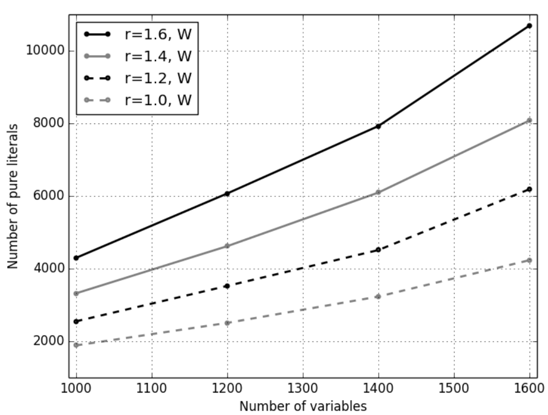

Figure 4: Number of applications of the MinSAT PLR

ply the MinSAT PLR (W/O). A total of 100 instances were solved at each point of the plots. Figure 4 shows the mean number of application of the MinSAT PLR when solving the instances of Figure 3. Figure 5 compares the mean number of nodes (y-axis) of the proof trees derived by MinSatz with and without the application of the MinSAT PLR. In all the figures the x-axis represents the number of variables.

We observe that the application of the MinSAT PLR produces significant speed-ups and reductions of the proof tree size on the tested instances, as well as that rule is applied a remarkable number of times. Interestingly, the speed-ups are negligible when the same instances are solved with MaxSatz.

Even though the worst-case complexity of the exact algorithms for MinSAT and MaxSAT coincides, in practice MinSAT may require more inference steps than MaxSAT because the number of empty clauses to be derived for a multiset C in MinSAT is always greater than or equal to the number of empty clauses to be derived in MaxSAT for C. Moreover, the MinSAT algorithm has to derive resolvents from $C_i \cup F_i$ at each step of the algorithm whereas the MaxSAT algorithm only has to derive resolvents from C_i, ignoring D_i in the next step. However, this may be an advantage in some cases because the clauses of D_i are shortened, and the availability of unit and binary clauses may produce shorter refutations.

Many practical optimization problems admit more compact and natural MaxSAT and MinSAT encodings if they are encoded using weighted clauses instead of unweighted ones, as well as considering hard and soft clauses. To keep the description as simple as possible, we have presented our results for unweighted MinSAT, but the proposed inference scheme can be extended to both WMinSAT and WPMinSAT.

In the case of WMinSAT, we just need to use the weighted version of the resolution rule [Bonet *et al.*, 2007; Larrosa *et al.*, 2008]. From a conceptual point of view, a weighted clause (c, w) is equivalent to have w copies of the unweighted clause c, and the application of the weighted MaxSAT resolution rule to two clauses $(x \vee D_1, w_1), (\overline{x} \vee D_2, w_2)$ is equivalent to apply the resolution rule $min(w_1, w_2)$ times to the unweighted clauses $x \vee D_1, \overline{x} \vee D_2$. So, using the weighted MaxSAT resolution rule, ϕ_n will be a multiset of weighted empty clauses $\{(\Box, w_1), \ldots, (\Box, w_k)\}$, and $w_1 + \cdots + w_k$ will be the optimal cost of the input multiset.

In the case of WPMinSAT, we must first derive an equivalent WMinSAT instance as explained below, and then solve the derived instance with the weighted version of our algorithm. We will assume that there is an assignment that satisfies all the hard clauses, since otherwise no solution exists.

Given a WPMinSAT instance ϕ whose number of hard clauses is $\#hard$ and whose sum of the weights of all its soft clauses is w, we derive a WMinSAT instance ϕ' by adding (i) all the soft clauses in ϕ, and (ii) the soft clauses $(\bar{l}_1, w{+}1)$, $(l_1 \vee \bar{l}_2, w+1), \ldots, (l_1 \vee l_2 \vee \cdots \vee \bar{l}_k, w+1)$ for each hard clause $h_i = l_1 \vee l_2 \vee \cdots \vee l_k$ in ϕ. These clauses are obtained by negating h_i and adding the weight $w + 1$ to each clause.

Observe that an assignment I satisfies h_i iff I falsifies exactly one clause among $\bar{l}_1, l_1 \vee \bar{l}_2, \ldots, l_1 \vee l_2 \vee \cdots \vee \bar{l}_k$; or equivalently, I falsifies h_i iff I satisfies the clauses $\bar{l}_1, l_1 \vee \bar{l}_2, \ldots, l_1 \vee l_2 \vee \cdots \vee \bar{l}_k$. Since the clauses derived from hard clauses have weight $w + 1$ and we assumed the hard part of

ϕ is satisfiable, every optimal solution of ϕ' falsifies exactly one clause derived from a hard clause, and is also an optimal solution of ϕ. Besides, if the maximum sum of the weights of the falsified clauses in ϕ' is m, then the maximum sum of the weights of the falsified clauses in ϕ is $m - \#hard \times (w + 1)$. Notice that the way of dealing with hard clauses in MinSAT is different than in MaxSAT, where hard clauses are not negated.

7 Conclusions

We described an exact inference algorithm for MinSAT and proved its correctness, extended it to WMinSAT and WPMinSAT, analyzed the differences between the MaxSAT and MinSAT inference schemes, and defined and empirically evaluated the MinSAT PLR. All these results are, to the best of our knowledge, the first contributions to the study of inference schemes for MinSAT, and provide further evidence that the solving techniques applicable to MaxSAT and MinSAT are often different. Hence, we believe that it is important to continue investigating on MinSAT.

As future work we plan to empirically compare the inference algorithms for MaxSAT and MinSAT, extend them to finite-domain variables [Ansótegui *et al.*, 2013], and identify optimization problem instances that are particularly well-suited for MaxSAT and MinSAT inference-based methods.

References

[Ansótegui *et al.*, 2012] Carlos Ansótegui, Chu Min Li, Felip Manyà, and Zhu Zhu. A SAT-based approach to MinSAT. In *Proc. of CCIA-2012*, pages 185–189, 2012.

[Ansótegui *et al.*, 2013] Carlos Ansótegui, Maria Luisa Bonet, Jordi Levy, and Felip Manyà. Resolution procedures for multiple-valued optimization. *Information Sciences*, 227:43–59, 2013.

[Argelich *et al.*, 2008] Josep Argelich, Chu Min Li, Felip Manyà, and Jordi Planes. The first and second Max-SAT evaluations. *Journal on Satisfiability, Boolean Modeling and Computation*, 4:251–278, 2008.

[Argelich *et al.*, 2011] Josep Argelich, Chu Min Li, Felip Manyà, and Jordi Planes. Analyzing the instances of the maxsat evaluation. In *Proc. of SAT-2011*, pages 360–361, 2011.

[Argelich *et al.*, 2012] Josep Argelich, Alba Cabiscol, Inês Lynce, and Felip Manyà. Efficient encodings from CSP into SAT, and from MaxCSP into MaxSAT. *Multiple-Valued Logic and Soft Computing*, 19(1-3):3–23, 2012.

[Argelich *et al.*, 2013] Josep Argelich, Chu Min Li, Felip Manyà, and Zhu Zhu. MinSAT versus MaxSAT for optimization problems. In *Proc. of CP 2013*, pages 133–142, 2013.

[Argelich *et al.*, 2014] Josep Argelich, Chu Min Li, Felip Manyà, and Zhu Zhu. Many-valued MinSAT solving. In *Proc. of ISMVL 2014*, pages 32–37, 2014.

[Bonet *et al.*, 2006] María Luisa Bonet, Jordi Levy, and Felip Manyà. A complete calculus for Max-SAT. In *Proc. of SAT-2006*, pages 240–251, 2006.

[Bonet *et al.*, 2007] María Luisa Bonet, Jordi Levy, and Felip Manyà. Resolution for Max-SAT. *Artificial Intelligence*, 171(8–9):240–251, 2007.

[Heras *et al.*, 2012] Federico Heras, António Morgado, Jordi Planes, and Joao Marques-Silva. Iterative SAT solving for minimum satisfiability. In *Proc. of ICTAI 2012*, pages 922–927, 2012.

[Ignatiev *et al.*, 2013] Alexey Ignatiev, António Morgado, Jordi Planes, and João Marques-Silva. Maximal falsifiability - definitions, algorithms, and applications. In *Proc. of LPAR 2013*, pages 439–456, 2013.

[Ignatiev *et al.*, 2014] Alexey Ignatiev, António Morgado, and João Marques-Silva. On reducing maximum independent set to minimum satisfiability. In *Proc. of SAT 2014*, pages 103–120, 2014.

[Kohli *et al.*, 1994] Rajeev Kohli, Ramesh Krishnamurti, and Prakash Mirchandani. The minimum satisfiability problem. *SIAM J. Discrete Mathematics*, 7(2):275–283, 1994.

[Kügel, 2012] Adrian Kügel. Natural Max-SAT encoding of Min-SAT. In *Proc. of LION 6*, 2012.

[Larrosa *et al.*, 2008] Javier Larrosa, Federico Heras, and Simon de Givry. A logical approach to efficient Max-SAT solving. *Artificial Intelligence*, 172(2–3):204–233, 2008.

[Li and Manyà, 2009] Chu Min Li and F. Manyà. MaxSAT, hard and soft constraints. In Armin Biere, Hans van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability*, pages 613–631. IOS Press, 2009.

[Li *et al.*, 2007] Chu Min Li, Felip Manyà, and Jordi Planes. New inference rules for Max-SAT. *Journal of Artificial Intelligence Research*, 30:321–359, 2007.

[Li *et al.*, 2010a] Chu Min Li, Felip Manyà, Nouredine Ould Mohamedou, and Jordi Planes. Resolution-based lower bounds in MaxSAT. *Constraints*, 15(4):456–484, 2010.

[Li *et al.*, 2010b] Chu Min Li, Felip Manyà, Zhe Quan, and Zhu Zhu. Exact MinSAT solving. In *Proc. of SAT-2010*, pages 363–368, 2010.

[Li *et al.*, 2011] Chu Min Li, Zhu Zhu, Felip Manyà, and Laurent Simon. Minimum satisfiability and its applications. In *Proc. of IJCAI-2011*, pages 605–610, 2011.

[Li *et al.*, 2012] Chu Min Li, Zhu Zhu, Felip Manyà, and Laurent Simon. Optimizing with minimum satisfiability. *Artificial Intelligence*, 190:32–44, 2012.

[Marathe and Ravi, 1996] M. V. Marathe and S. S. Ravi. On approximation algorithms for the minimum satisfiability problem. *Information Processing Letters*, 58:23–29, 1996.

[Morgado *et al.*, 2013] António Morgado, Federico Heras, Mark H. Liffiton, Jordi Planes, and João Marques-Silva. Iterative and core-guided MaxSAT solving: A survey and assessment. *Constraints*, 18(4):478–534, 2013.

[Zhu *et al.*, 2012] Zhu Zhu, Chu Min Li, Felip Manyà, and Josep Argelich. A new encoding from MinSAT into MaxSAT. In *Proc. of CP 2012*, pages 455–463, 2012.