# From Non-Clausal to Clausal MinSAT

Chu-Min LI, [a]Felip MANYÀ, [b] Joan Ramon SOLER, [b] and Amanda VIDAL [b]

[a] *MIS, Université de Picardie, France*
[b] *Artificial Intelligence Research Institute, IIIA-CSIC, Spain*

**Abstract.** We tackle the problem of solving MinSAT for multisets of propositional formulas that are not necessarily in clausal form. Our approach reduces non-clausal to clausal MinSAT, since this allows us to rely on the much developed clause-based MinSAT solvers. The main contribution of this paper is the definition of several transformations of multisets of propositional formulas into multisets of clauses so that the maximum number of unsatisfied clauses in both multisets is preserved.

**Keywords.** Minimum satisfiability, clausal form, cost-preserving transformation.

## 1. Introduction

SAT is the problem of deciding whether there exists an assignment that satisfies a given (multi)set of propositional formulas. On the other hand, MaxSAT and MinSAT are optimization versions of SAT whose goal is to find an assignment that minimizes or maximizes the number of unsatisfied formulas, respectively. These problems are significant because many practical questions can be solved by first encoding them as a SAT, MaxSAT or MinSAT problems, and then finding a solution by solving the resulting encoding with a SAT, MaxSAT or MinSAT solver. While SAT is used to solve decision problems, MaxSAT and MinSAT are used to solve optimization problems [2,5].

We can distinguish between clausal SAT, MaxSAT, and MinSAT and non-clausal SAT, MaxSAT, and MinSAT, respectively. In the clausal case, the input multiset only contains clauses (i.e., disjunctions of literals). In the non-clausal case, the input multiset contains propositional formulas that are not necessarily in clausal form.

Many combinatorial problems admit more natural and compact encodings when represented in non-clausal form. However, the fastest and most robust SAT/MaxSAT/MinSAT solvers require their input in clausal form. Thus, some kind of clausal form transformation is needed to solve them. In SAT, there are several algorithms that transform a multiset of arbitrary propositional formulas into a satisfiability equivalent multiset of clauses [7,9]. Unfortunately, usual clausal form transformations used in SAT are not valid neither in MaxSAT nor MinSAT. The reason is that they are not cost-preserving; i.e., they do not preserve the minimum/maximum number of unsatisfied formulas between the input and the transformed multiset. It is therefore important to analyze how the existing SAT clausal form transformations behave for MaxSAT and MinSAT, as well as to investigate new cost-preserving transformations.

A few approaches to solve the non-clausal MaxSAT problem have been reported in [1,3,4], including one based on clausal form transformations [3]. To the best of our knowledge, no clausal form transformations have been defined for MinSAT. The

only existing approach to solve Non-clausal MinSAT is via its reduction to Non-clausal MaxSAT [8]. Thus, in this paper, we focus on cost-preserving transformations from non-clausal to clausal MinSAT. More specifically, we define four transformations that, in practice, might produce different computational behavior.

## 2. Preliminaries

Given a set of propositional variables $\{x_1, \ldots, x_n\}$, a *literal* $\ell$ is a variable $x_i$ or its negation $\neg x_i$, and a *clause* is a disjunction of literals. A *weighted clause* is a pair $(c, w)$, where $c$ is a clause and $w$, its weight, is a positive integer or infinity. If its weight is infinity, it is called a *hard clause* (we omit infinity weights for simplicity); otherwise, it is a *soft clause*. A *Weighted Partial MinSAT* instance is a finite multiset of weighted clauses. We represent MinSAT instances using multisets instead of sets because repeated clauses cannot be collapsed into one of such clauses as in SAT. Considering sets might affect the preservation of the minimum number of unsatisfied clauses.

Let the binary operations $\wedge, \vee$ be defined between (multi) sets of formulas as usual, namely for two multisets $A, B$ and operation $\star$ either $\vee$ or $\wedge$, we let $A \star B := \{a \star b \colon a \in A, b \in B\}$. This naturally implies that the operations are associative and distributive over multisets. Moreover, for a multiset $A$ and a single formula $\varphi$, we will use the convention that $A \star \varphi := A \star \{\varphi\}$. Lastly, $\cup$ or $\bigcup$ will denote the union of sets while $\sqcup$ and $\bigsqcup$ will denote the union of multisets, so that $\{a\} \cup \{a\} = \{a\}$ and $\{a\} \sqcup \{a\} = \{a, a\}$.

A *truth assignment* or evaluation is a mapping from the variables into $\{0, 1\}$. We say it satisfies literal $x$ ($\neg x$) if $x$ evaluates to 1 (0), weighted clause $(c, w)$ if it satisfies a literal of $c$, and a multiset of clauses if it satisfies all its clauses. The weight $w$ is interpreted as the penalty of violating clause $c$. When all clauses have the same weight, their weights will be omitted.

The Weighted Partial MaxSAT problem, or WPMaxSAT, for an instance $\phi$, consists in finding an assignment that satisfies the hard clauses and minimizes the sum of the weights of the unsatisfied soft clauses. The Weighted Partial MinSAT problem, or WPMinSAT, is to find an assignment that satisfies the hard clauses and maximizes the sum of the weights of the unsatisfied soft clauses. The most common subproblems of WPMaxSAT are the following: Weighted MaxSAT (WMaxSAT), which is WPMaxSAT without hard clauses; Partial MaxSAT (PMaxSAT), which is WPMaxSAT when all the soft clauses have the same weight, and MaxSAT, which is PMaxSAT without hard clauses. Similarly, WMinSAT is WPMinSAT without hard clauses; PMinSAT is WPMinSAT when all the soft clauses have the same weight, and MinSAT is PMinSAT without hard clauses.

On the other hand, arbitrary propositional formulas are built in the usual way from a set of variables by using the binary connectives $\wedge, \vee$ and the unary connective $\neg$. Implication and bi-implication are binary connectives defined from the previous ones, by letting $\varphi \rightarrow \psi := \neg\varphi \vee \psi$ and $\varphi \leftrightarrow \psi := (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$. Propositional formulas are evaluated by considering the standard semantics of the connectives. When WPMaxSAT (WPMinSAT), as well as its simpler cases, take as input a multiset of propositional formulas, we refer to this problem as Non-clausal WPMaxSAT (WPMinSAT).

For what concerns this work, given a multiset of formulas $A$, $MinSAT(A)$ will formally denote the maximum number of unsatisfiable formulas in $A$.

Any propositional formula can be translated into Conjunctive Normal Form (CNF) through the following rules, which preserve logical equivalence: double negation elimination, De Morgan's laws and distributivity of $\vee$ over $\wedge$. Further simplifications[1] are also applied. This produces a formula in CNF, namely, a conjunction of clauses. In the sequel, $CNF(\varphi)$ denotes the formula equivalent to $\varphi$ in CNF resulting from the previous equivalences, so naturally, for any truth-assignment $e$ it holds that $e(\varphi) = e(CNF(\varphi))$.

When inquiring about the satisfiability of a set of propositional formulas $A = \{\varphi_1, \ldots, \varphi_n\}$, each of the $CNF(\varphi_i)$ formulas can be split into a set of clauses (simply removing the conjunctions) to get a set of clauses that will serve as input to a SAT solver. We will refer to the union of these sets by $CNF_{SAT}(A)$. The satisfiability of $CNF_{SAT}(A)$ coincides with that of $A$, i.e., $SAT(A)$ if and only if $SAT(CNF_{SAT}(A))$. As we show below, this approach does not serve to solve non-clausal MinSAT or MaxSAT.

**Example 2.1.** *Let $A = \{(\neg x \leftrightarrow x) \wedge (\neg y \leftrightarrow y), x \vee y\}$ be a multiset of propositional formulas. Applying the transformation defined above, we get $CNF_{SAT}(A) = \{x, \neg x, y, \neg y, x \vee y\}$. The evaluation $e(x) = e(y) = 0$ violates two formulas of $A$ and three clauses of $CNF_{SAT}(A)$, despite being an optimal MinSAT solution for both $A$ and $CNF_{SAT}(A)$. Thus, that transformation is not cost preserving and $MinSAT(A) \neq MinSAT(CNF_{SAT}(A))$. Similarly, the evaluation $e(x) = e(y) = 1$ is an optimal MaxSAT solution for both $A$ and $CNF_{SAT}(A)$, but violates one formula of $A$ and two clauses of $CNF_{SAT}(A)$.*

Example 2.1 shows that the main problem arises when the CNF is split into clauses. This operation can generate additional clauses that violate the preservation of the maximum number of unsatisfied formulas. To overcome this drawback, we propose several new cost-preserving transformations for MinSAT in the next two sections.

## 3. Transformation $CNF_{minSAT_d}$

We first define a cost-preserving transformation for MinSAT, called $CNF_{minSAT_d}$, which does not add new variables. It has the advantage that it does not expand the solution space.

We define the partial mapping $^* : Fm \rightarrow MS$, where $Fm$ stands for arbitrary formulas (but, as seen below, $*$ will be defined only over two particular families of formulas) and $MS$ denotes the set of all multisets of clauses, as follows:

$$(c_1 \wedge c_2 \wedge \ldots \wedge c_n)^* := \{c_1, (\neg c_1)^* \vee c_2, \ldots, (\neg c_1)^* \vee (\neg c_2)^* \vee \ldots \vee (\neg c_{n-1})^* \vee c_n\}$$

$$(\neg(\ell_1 \vee \ell_2 \vee \ldots \ell_n))^* := \{\neg \ell_1, \ell_1 \vee \neg \ell_2, \ldots, \ell_1 \vee \ell_2 \vee \ldots \vee \ell_{n-1} \vee \neg \ell_n\}$$

for $c_i$ denoting clauses and $\ell_i$ denoting literals.

**Definition 3.1** (Transformation $CNF_{minSAT_d}$)**.** *Let $A = A_{NC} \sqcup A_C$ be a multiset of formulas of which $A_{NC}$ are not clauses and $A_C$ are clauses. $CNF_{minSAT_d}(A)$ is the multiset of clauses*

$$\bigsqcup_{\varphi \in A_{NC}} CNF(\varphi)^* \sqcup A_C.$$

---

[1] e.g., removing clauses with a literal and its negation, or repeated clauses.

**Example 3.2.** *Given the multiset $A = \{\neg(\neg x_1 \wedge \neg x_2) \wedge (x_3 \vee x_4)\}$ formed by a single formula $\varphi = \neg(\neg x_1 \wedge \neg x_2) \wedge (x_3 \vee x_4)$, we have that $CNF(\varphi) = (x_1 \vee x_2) \wedge (x_3 \vee x_4)$. We convert it to a cost-preserving multiset of clauses for MinSAT as follows:*

$$CNF_{minSAT_d}(A) = ((x_1 \vee x_2) \wedge (x_3 \vee x_4))^* = \{x_1 \vee x_2, (\neg(x_1 \vee x_2))^* \vee (x_3 \vee x_4)\} =$$
$$\{x_1 \vee x_2, \{\neg x_1, x_1 \vee \neg x_2\} \vee (x_3 \vee x_4)\} =$$
$$\{x_1 \vee x_2, \neg x_1 \vee x_3 \vee x_4, \neg x_2 \vee x_3 \vee x_4\}.$$

To prove that transformation $CNF_{minSAT_d}$ is cost-preserving for MinSAT, we first study how translation $*$ behaves. Since it is, in an intuitive way, working from the inner-most level to the outer-most, let us first show the most internal level, and later, how the full translation works.

**Lemma 3.3.** *For a clause $c = \ell_1 \vee \ldots \vee \ell_n$ and a truth assignment $e$, $e(c) = 1$ if and only if all clauses $d \in (\neg c)^*$ are evaluated to 1 except for one.*

*Proof.* Since $e(c) = 1$ there is a minimum index $1 \le i_0 \le n$ for which $e(\ell_{i_0}) = 1$. For any $j < i_0$, since $e(\ell_j) = 0$, we have that $e(\ell_1 \vee \ell_2 \vee \ldots \vee \neg \ell_j) \ge e(\neg \ell_j) = 1$. Similarly, for any $j > i_0$, $e(\ell_1 \vee \ell_2 \vee \ldots \vee \neg \ell_j) \ge e(\ell_{i_0}) = 1$. □

**Lemma 3.4.** *For any formula $\phi$ and any truth-assignment $e$, it holds that $e(CNF(\phi)) = 0$ if and only if there is exactly one clause $c$ in $CNF(\phi)^*$ such that $e(c) = 0$. Moreover, for any $e$, either $e$ satisfies all clauses from $CNF(\phi)^*$ or it satisfies all but one clause.*

*Proof.* Let $CNF(\phi) = c_1 \wedge \ldots \wedge c_n$. First, assume that $e(CNF(\phi)) = 0$, and we will see there is exactly one falsified clause in $CNF(\phi)^*$. $e(c_1 \wedge \ldots \wedge c_n) = 0$ implies $i_0 = min\{i \in \{1, \ldots, n\}: e(c_i) = 0\}$ exists. Then, clearly for any $j < i_0$ and any $c \in (\neg c_1)^* \vee \ldots \vee (\neg c_{j-1})^* \vee c_j$, it holds that $e(c) = 1$, since by definition of disjunction of multisets, any such $c$ is of the form $\ldots \vee c_j$.

On the other hand, for $j > i_0$, any clause $c \in (\neg c_1)^* \vee \ldots \vee (\neg c_{j-1})^* \vee c_j$ has a sub-clause that belongs to $(\neg c_{i_0})^*$. We prove now that for any such $d \in (\neg c_{i_0})^*$ it also holds that $e(d) = 1$, thus implying $e(c) = 1$ too. Indeed, $c_{i_0} = l_1 \vee \ldots \vee l_s$ for some $s$, so $e(c_{i_0}) = 0$ implies that $e(l_i) = 0$ for all $1 \le i \le s$. Since every clause $d$ in $\neg(c_{i_0})^*$ is by definition of the form $\ldots \vee \neg l_j$ for $1 \le j \le s$, necessarily $e(d) \ge e(\neg l_j) = 1$.

Now, from Lemma 3.3, we know that all clauses from $(\neg c_1)^* \vee \ldots \vee (\neg c_{i_0-1})^* \vee c_{i_0}$ are satisfied by $e$ except for $d_1 \vee \ldots \vee d_{i_0-1} \vee c_{i_0}$ from $CNF(\phi)^*$, which is falsified.

On the other hand, to check that if one clause from $CNF(\phi)^*$ is falsified under $e$ then also $CNF(\phi)$ is falsified, we can reason by contraposition. Assume $e(CNF(\phi)) = 1$, so $e(c_i) = 1$ for each $1 \le i \le n$. Since any clause $d$ from $CNF(\phi)^*$ is of the form $\ldots \vee c_j$ for some $1 \le j \le n$, it is immediate that all clauses from $CNF(\phi)^*$ are satisfied.

Now, since any evaluation $e$ can either satisfy or falsify $CNF(\phi)$, from the cases above we know $e$ can either satisfy all clauses in $CNF(\phi)^*$, or satisfy all but one clause from $CNF(\phi)^*$. This concludes the proof. □

The previous lemma directly implies that, for any set of formulas $A$ and truth assignment $e$, it holds that $|\{\phi \in A: e(\phi) = 0\}| = |\{\phi \in CNF_{minSAT_d}(A): e(\phi) = 0\}|$.

**Corollary 3.5.** $MinSAT(A) = MinSAT(CNF_{minSAT_d}(A))$.

*Proof.* To check that $MinSAT(A) \leq MinSAT(CNF_{minSAT_d}(A))$, suppose that $MinSAT(A) = n$. Then, there is some truth assignment $e$ such that $|\{\phi \in A: e(\phi) = 0\}| = n$. Thus, also $|\{\phi \in CNF_{minSAT_d}(A): e(\phi) = 0\}| = n$, so the maximum number of unsatisfied formulas in $CNF_{minSAT_d}(A)$ is necessarily greater or equal than $n$.

The analogous reasoning in the other direction proofs $MinSAT(CNF_{minSAT_d}(A)) \leq MinSAT(A)$. $\square$

**Example 3.6.** *Transformation $CNF_{minSAT_d}$, applied to the multiset of formulas $A = \{\neg(\neg x_1 \wedge \neg x_2) \wedge (x_3 \vee x_4)\}$ from Example 3.2, derived the multiset of clauses $\Phi = \{x_1 \vee x_2, \neg x_1 \vee x_3 \vee x_4, x_1 \vee \neg x_2 \vee x_3 \vee x_4\}$. Corollary 3.5 guarantees that the maximum number of unsatisfied formulas in $A$ and $\Phi$ is preserved. In fact, for every evaluation $e$, the number of formulas unsatisfied by $e$ in $A$ and $\Phi$ is the same.*

## 4. Extending the Language and Relying on Partial MinSAT

We now define three cost-preservation transformations for MinSAT that add fresh variables to the language and rely on the partial MinSAT formalism to express some hard equivalences between the new variables and the non-clausal formulas of the input multiset $A$. We add the new variables $y_\psi$ for some (sub)formulas $\psi$ appearing in $A$ or in $CNF_{SAT}(A)$, which will be specified for each translation. The difference among the three transformations lies in the formulas that receive a new variable and the way we encode the hard and soft constraints. For a given non-clausal MinSAT problem, the proposed transformations can generate multisets of clauses whose size ranges from polynomial to exponential in the length of the input formula. Moreover, the number of fresh variables can also be substantially different.

In what follows, we will use the following convention. For a set of formulas $A$ and an arbitrary truth assignment $e$, we will let $e'$ to be the modified truth assignment defined by letting $e'(p) = e(p)$ for all variables $p$ in $A$, and $e'(y_\psi) = e(\psi)$ in all other cases.

### 4.1. Transformation $CNF_{minSAT_e}$

**Definition 4.1** (Transformation $CNF_{minSAT_e}$). *Let $A = A_{NC} \sqcup A_C$ be a multiset of formulas of which $A_{NC}$ are not a clauses and $A_C$ are clauses. Let $\{y_\varphi: \varphi \in A_{NC}\}$ be a set of fresh variables not appearing in $A$. We call $CNF_{minSAT_e}(A)$ to the partial MinSAT instance given by*

$$\textbf{Hard clauses (HC)} := \bigcup_{\varphi \in A_{NC}} CNF_{SAT}(\neg \varphi \vee y_\varphi), \quad \textbf{Soft clauses (SC)} := \bigsqcup_{\varphi \in A_{NC}} \{y_\varphi\} \cup A_C$$

Notice that the formulas $\varphi \in A_{NC}$ occur negated in $CNF_{SAT}(\neg \varphi \vee y_\varphi)$. This is relevant because the formulas $\varphi$ occur with positive polarity in transformation $CNF_{minSAT_d}$. This implies that we could avoid the combinatorial explosion due to the application of distributivity of $\vee$ over $\wedge$ if the most appropriate transformation is chosen for each formula.

**Example 4.2.** *For the multiset $A = \{\neg(\neg x_1 \wedge \neg x_2) \wedge (x_3 \vee x_4)\}$ of Example 3.2, we have that $\neg \varphi \vee y_\varphi = \neg(\neg(\neg x_1 \wedge \neg x_2) \wedge (x_3 \vee x_4)) \vee y_\varphi$. The application of Transformation $CNF_{minSAT_e}$ derives the following partial MinSAT instance:*

Hard clauses: $\quad \neg x_1 \vee \neg x_3 \vee y_\varphi \quad \neg x_1 \vee \neg x_4 \vee y_\varphi \quad \neg x_2 \vee \neg x_3 \vee y_\varphi \quad \neg x_2 \vee \neg x_4 \vee y_\varphi$
Soft clauses: $\quad y_\varphi$

In the proofs below for checking that the previous translation is cost preserving, with the objective of lightening the notation, we will assume $A_C$ is empty, since it is clear it does not affect the calculations.

**Lemma 4.3.** *Let A be a multiset of formulas and let e be a truth assignment such that $e(c) = 1$ for all $c \in \mathbf{HC}(CNF_{minSAT_e}(A))$. Then,*

$$|\{c \in \mathbf{SC}(CNF_{minSAT_e}(A)) \colon e(c) = 0\}| \leq |\{\varphi \in A \colon e(\varphi) = 0\}|.$$

*Proof.* By assumption $e(c) = 1$ for all $c \in CNF_{SAT}(\neg\varphi \vee y_\varphi)$ for each $\varphi \in A$. Since for an arbitrary formula $\psi$ the conjunction of all clauses $CNF_{SAT}(\psi)$ equals $\psi$, it is clear that $e(\neg\varphi \vee y_\varphi) = 1$ for all $\varphi \in A$, implying that $e(\varphi) \leq e(y_\varphi)$. Then, for each $\varphi \in A$ such that $e(y_\varphi) = 0$, necessarily $e(\varphi) = 0$. □

On the other hand, for any set of formulas $A$ and evaluation $e$, we can prove that the evaluation $e'$ defined at the start of Section 4 is such that the number of falsified formulas of $A$ under $e$ and that of falsified soft clauses in $CNF_{minSAT_e}(A)$ under $e'$ is the same.

**Lemma 4.4.** *Let A be a multiset of formulas and let e be an arbitrary truth assignment. Then, $e'$ satisfies all hard clauses in $CNF_{minSAT_e}(A)$ and*

$$|\{\phi \in A \colon e(\phi) = 0\}| = |\{c \in \mathbf{SC}(CNF_{minSAT_e}(A)) \colon e'(c) = 0\}|.$$

*Proof.* Observe that the truth assignment $e'$ is defined in such a way that it preserves evaluation for all variables in $A$ (and so, for all formulas in $A$), and where $e'(y_\varphi) = e(\varphi)$. Then, for any $\varphi \in A$ we have that $e'(\neg\varphi \vee y_\varphi) = e'(\neg\varphi) \vee e'(y_\varphi) = e(\neg\varphi) \vee e(\varphi) = 1$. Thus, all hard clauses in $CNF_{minSAT_e}(A)$ hold.

For what concerns the second statement of the lemma, we only need to check that $|\{\phi \in A \colon e(\phi) = 0\}| = |\{y_\phi \in \mathbf{SC}(CNF_{minSAT_e}(A)) \colon e'(y_\phi) = 0\}|$. This is immediate, since $e'(y_\phi) = e(\phi)$ by definition. □

**Corollary 4.5.** $MinSAT(A) = MinSAT(CNF_{minSAT_e}(A))$.

*Proof.* To prove $\leq$ observe that $MinSAT(A) = n$ implies there is some truth assignment $e$ for which $|\{\phi \in A \colon e(\phi) = 0\}| = n$. Then, Lemma 4.4 implies that $|\{c \in \mathbf{SC}(CNF_{minSAT_e}(A)) \colon e'(c) = 0\}| = n$ too. Since $MinSAT(CNF_{minSAT_e}(A))$ is the maximum number of falsifiable clauses, we get that, in particular $MinSAT(CNF_{minSAT_e}(A)) \geq n$ too, so $MinSAT(A) \leq MinSAT(CNF_{minSAT_e}(A))$.

Symmetrically, to prove $\geq$ observe that, by definition, $MinSAT(CNF_{minSAT_e}(A)) = |\{c \in \mathbf{SC}(CNF_{minSAT_e}(A)) \colon g(c) = 0\}|$ for a certain truth assignment $g$ that moreover satisfies all hard clauses in $CNF_{minSAT_e}(A)$. Applying Lemma 4.3 we know that, for such $g$, it holds that $|\{c \in \mathbf{SC}(CNF_{minSAT_e}(A)) \colon g(c) = 0\}| \leq |\{\varphi \in A \colon g(\varphi) = 0\}|$. Thus, $|\{\varphi \in A \colon g(\varphi) = 0\}| \geq n$. Again, since $MinSAT(A)$ is the maximum of falsifiable formulas from $A$, in particular $MinSAT(A) \geq n$ too, meaning that $MinSAT(A) \geq MinSAT(CNF_{minSAT_e}(A))$, concluding the proof. □

## 4.2. Transformation $CNF_{minSAT_i}$

**Definition 4.6** (Transformation $CNF_{minSAT_i}$). *Let $A = A_{NC} \sqcup A_C$ be a multiset of formulas of which $A_{NC}$ are not clauses and $A_C$ are clauses. Let $\{y_c \colon \varphi \in A_{NC}, c \in CNF_{SAT}(\varphi)\}$ be a set of fresh variables not appearing in A.*

*We call $CNF_{minSAT_i}(A)$ to the partial MinSAT instance given by*

$$\textbf{\textit{Hard clauses (HC)}} \coloneqq \bigcup_{\varphi \in A_{NC}} \bigcup_{c \in CNF_{SAT}(\varphi)} \{\neg c \vee y_c\}$$

$$\textbf{\textit{Soft clauses (SC)}} \coloneqq \bigsqcup_{\varphi \in A_{NC}} (\bigwedge_{c \in CNF_{SAT}(\varphi)} y_c)^*$$

**Example 4.7.** *For the multiset $A = \{(\neg x_1 \to x_2) \wedge (x_1 \to \neg x_2)\}$, transformation $CNF_{minSAT_i}(A)$ derives the following partial MinSAT instance:*

| | | | | |
|---|---|---|---|---|
| Hard clauses: | $\neg x_1 \vee y_{c_1}$ | $\neg x_2 \vee y_{c_1}$ | $x_1 \vee y_{c_2}$ | $x_2 \vee y_{c_2}$ |
| Soft clauses: | $y_{c_1}$ | $\neg y_{c_1} \vee y_{c_2}$ | | |

*Observe that $CNF_{minSAT_i}(A)$ associates one fresh variable with every clause of $CNF_{SAT}(\varphi)$. If only one fresh variable is associated with each CNF, we would have as hard clauses $\neg x_1 \vee y_{c_1}, \neg x_2 \vee y_{c_1}, x_1 \vee y_{c_1}, x_2 \vee y_{c_1}$, and we would get an infeasible solution if $y_{c_1}$ is unsatisfied because we would detect a contradiction in the hard part.*

*If we consider the multiset $A' = \{(\neg x_1 \to x_2), (x_1 \to \neg x_2)\}$ containing two formulas, transformation $CNF_{minSAT_i}(A)$ derives the following partial MinSAT instance:*

| | | | | |
|---|---|---|---|---|
| Hard clauses: | $\neg x_1 \vee y_{c_1}$ | $\neg x_2 \vee y_{c_1}$ | $x_1 \vee y_{c_2}$ | $x_2 \vee y_{c_2}$ |
| Soft clauses: | $y_{c_1}$ | $y_{c_2}$ | | |

*Despite the similarity between A and A', the maximum number of unsatisfied clauses is 1 in A and 2 in A'. They have the same hard part, but the soft clauses are different.*

Similarly to what we did in the previous subsection, we will prove the cost preservation assuming $A_C = \emptyset$. We prove an analogous version of Lemma 4.3, but resorting to Lemma 3.4 to keep track of the behavior of the new soft clauses.

**Lemma 4.8.** *Let A be a multiset of formulas and let e be a truth assignment such that $e(c) = 1$ for all $c \in \textbf{HC}(CNF_{minSAT_i}(A))$. Then,*

$$|\{c \in \textbf{SC}(CNF_{minSAT_i}(A)) \colon e(c) = 0\}| \leq |\{\varphi \in A \colon e(\varphi) = 0\}|.$$

*Proof.* By assumption $e(\neg c \vee y_c)$ for all $c \in CNF_{SAT}(\varphi)$ with $\varphi \in A$. Thus, $e(c) \leq e(y_c)$ for all such $c$ and $y_c$. Moreover, observe that each formula of the form $\bigwedge_{c \in CNF_{SAT}(\varphi)} y_c$ is already in conjunctive normal form. Thus, by Lemma 3.4, $e(\bigwedge_{c \in CNF_{SAT}(\varphi)} y_c) = 0$ if and only if exactly one clause from $(\bigwedge_{c \in CNF_{SAT}(\varphi)} y_c)^*$ is falsified, and otherwise all clauses in $(\bigwedge_{c \in CNF_{SAT}(\varphi)} y_c)^*$ are satisfied by $e$.

By definition,

$$|\{c \in \mathbf{SC}(CNF_{minSAT_i}(A)) \colon e(c) = 0\}| = \sum_{\varphi \in A} |\{d \in (\bigwedge_{c \in CNF_{SAT}(\varphi)} y_c)^* \colon e(d) = 0\}|.$$

The previous observation implies that $|\{d \in (\bigwedge_{c \in CNF_{SAT}(\varphi)} y_c)^* \colon e(d) = 0\}| \leq 1$. Moreover, $|\{d \in (\bigwedge_{c \in CNF_{SAT}(\varphi)} y_c)^* \colon e(d) = 0\}| = 1$ if and only if $e(\bigwedge_{c \in CNF_{SAT}(\varphi)} y_c) = 0$, and so, there is at least some $c \in CNF_{SAT}(\varphi)$ for which $e(y_c) = 0$ too. Since we saw above that $e(c) \leq e(y_c)$, we have that $e(c) = 0$, making $e(\varphi) = 0$ too. $\square$

We will see in the next lemma the analogous to Lemma 4.4. Indeed, we can easily check that the number of unsatisfied formulas of $A$ under $e$ coincides with the number of unsatisfied soft clauses in $CNF_{minSAT_i}(A)$ under $e'$ (for $e'$ as defined in the beginning of Section 4). It will only be necessary to rely on Lemma 3.4 when necessary.

**Lemma 4.9.** *Let $A$ be a multiset of formulas and let $e$ be a truth assignment. Then, $e'$ satisfies all hard clauses in $CNF_{minSAT_i}(A)$ and $|\{\phi \in A \colon e(\phi) = 0\}| = |\{c \in \mathbf{SC}(CNF_{minSAT_i}(A)) \colon e'(c) = 0\}|$.*

*Proof.* Observe that the truth assignment $e'$ is defined in such a way that it preserves evaluation for all variables in $A$ (and so, for all formulas in $A$), and where $e'(y_c) = e(c)$ for all $c \in CNF_{SAT}(\varphi)$ for $\varphi \in A$. Then, for any $\varphi \in A$ we have that $e'(\neg c \vee y_\varphi) = e'(\neg c) \vee e'(y_c) = e(\neg c) \vee e(c) = 1$. Thus, all hard clauses in $CNF_{minSAT_i}(A)$ hold.

For what concerns the second statement of the lemma, we only need to check that

$$|\{\phi \in A \colon e(\phi) = 0\}| = |\{d \in (\bigwedge_{c \in CNF_{SAT}(\phi)} y_c)^* \colon \phi \in A, e'(d) = 0\}|.$$

By Lemma 3.4, for each $\phi \in A$ there are two cases:

- $|\{d \in (\bigwedge_{c \in CNF_{SAT}(\phi)} y_c)^* \colon e'(d) = 0\}| = 1$, which happens if and only if $e'(\bigwedge_{c \in CNF_{SAT}(\phi)} y_c) = 0$, and so, since $e'(y_c) = e(c)$ for each such $c$, if and only if $e(\phi) = 0$ too, or
- $|\{d \in (\bigwedge_{c \in CNF_{SAT}(\phi)} y_c)^* \colon e'(d) = 0\}| = 0$, which implies, following the same reasoning, that $e(\phi) = 1$. $\square$

**Corollary 4.10.** $MinSAT(A) = MinSAT(CNF_{minSAT_e}(A))$

*Proof.* To prove $\leq$ observe that $MinSAT(A) = n$ implies there is some truth assignment $e$ for which $|\{\phi \in A \colon e(\phi) = 0\}| = n$. Then Lemma 4.4 implies that $|\{c \in \mathbf{SC}(CNF_{minSAT_e}(A)) \colon e'(c) = 0\}| = n$ too. Since $MinSAT(CNF_{minSAT_e}(A))$ is the maximum number of falsifiable clauses, we get that, in particular, $MinSAT(CNF_{minSAT_e}(A)) \geq n$ too, so $MinSAT(A) \leq MinSAT(CNF_{minSAT_e}(A))$.

Symmetrically, to prove $\geq$ observe that, by definition, $MinSAT(CNF_{minSAT_e}(A)) = |\{c \in \mathbf{SC}(CNF_{minSAT_e}(A)) \colon e(c) = 0\}|$ for a certain truth assignment $e$ that moreover satisfies all hard clauses in $CNF_{minSAT_e}(A)$. Applying Lemma 4.3 we know that, for such $g$, it holds that $|\{c \in \mathbf{SC}(CNF_{minSAT_e}(A)) \colon g(c) = 0\}| \leq |\{\varphi \in A \colon g(\varphi) = 0\}|$. Thus, $|\{\varphi \in A \colon g(\varphi) = 0\}| \geq n$. Again, since $MinSAT(A)$ is the maximum of falsifiable formulas from $A$, in particular $MinSAT(A) \geq n$ too, meaning that $MinSAT(A) \geq MinSAT(CNF_{minSAT_e}(A))$, concluding the proof. $\square$

### 4.3. Transformation $CNF_{minSAT_t}$

To avoid the generation of a number of clauses that can be exponential in the number of input formulas due to the application of distributivity when deriving the CNF, we can adapt the Tseitin transformation to MinSAT. We begin by recalling the Tseitin transformation for SAT, and then point out how it is adapted to MinSAT.

For each formula $\varphi$, let us denote by $SFm(\varphi)$ the set of subformulas of $\varphi$. Then, for each $\psi \in SFm(\varphi)$ consider a new variable $y_\psi$, and for each such formula $\psi$ we define the set of clauses $Def(\psi)$ by[2]

$$Def(p) := \emptyset \qquad \qquad \text{for } p \text{ propositonal variable,}$$
$$Def(\psi \star \chi) := CNF_{SAT}(y_{\psi \star \chi} \leftrightarrow y_\psi \star y_\chi) \qquad \text{for } \star \in \{\vee, \wedge\},$$
$$Def(\neg \psi) := CNF_{SAT}(y_{\neg \psi} \leftrightarrow \neg y_\psi)$$

Recall that all connectives different from $\wedge, \vee, \neg$ are simply wrapping some expression involving these three, so any formula $\varphi$ is written in fact in the previous language and so, $Def$ is correctly defined for all formulas. It is clear that the above definitions generate clauses with at most 3 literals each. Then, for a formula $\varphi$, its Tseitin SAT transformation $T(\varphi)$ is the set of clauses $T(\varphi) := \{y_\varphi\} \cup \bigcup_{\psi \in SFm(\varphi)} Def(\psi)$.

It is rather standard to check that $SAT(\varphi)$ if and only if $SAT(T(\varphi))$, and thus, also for a set of formulas $A$, we have that $SAT(A)$ if and only if $SAT(T(A))$, where, as usual, by $T(A)$ we denote the set $\bigcup_{\varphi \in A} T(\varphi)$.

To use this transformation in order to preserve *MinSAT*, it is only necessary to do a slight modification to the previous transformation and rely on partial *MinSAT*. Let us consider the transformation $T^-(\varphi)$ given by

$$T^-(\varphi) := \bigcup_{\psi \in SFm(\varphi)} Def(\psi)$$

Observe this is simply removing from the resulting set of clauses the outermost variable, which is the one that, at SAT, is imposing that the corresponding formula is satisfied. Similarly to $CNF_{minSAT_e}$, we define the transformation $CNF_{minSAT_t}$.

**Definition 4.11** (Transformation $CNF_{minSAT_t}$). *Let $A = A_{NC} \sqcup A_C$ be a multiset of formulas of which $A_{NC}$ are not clauses and $A_C$ are clauses. We call $CNF_{minSAT_t}(A)$ to the partial MinSAT instance given by*

$$\textbf{Hard clauses (HC)} := \bigcup_{\varphi \in A_{NC}} T^-(\varphi), \qquad \textbf{Soft clauses (SC)} := \bigsqcup_{\varphi \in A_{NC}} \{y_\varphi\} \sqcup A_C$$

**Example 4.12.** *Given the multiset of formulas $A = \{x_1 \wedge x_2, x_3 \wedge x_4\}$, $CNF_{minSAT_t}(A)$ derives the following partial MinSAT instance:*

Hard clauses: $\neg y_1 \vee x_1 \quad \neg y_1 \vee x_2 \quad y_1 \vee \neg x_1 \vee \neg x_2 \quad \neg y_2 \vee x_3 \quad \neg y_2 \vee x_4 \quad y_2 \vee \neg x_3 \vee \neg x_4$
Soft clauses: $y_1 \quad y_2$

---

[2]To simplify the notation, we do not distinguish between propositional variables and more complex formulas. Thus, a variable $p$ will receive a fresh variable $y_p$ in the new language, and $p$ will no longer appear in the translation. The definition of $Def$ over propositional variables is also included to lighten notation later on.

**Lemma 4.13.** $MinSAT(A) = MinSAT(CNF_{minSAT_t}(A))$

*Proof.* To prove $\leq$ observe that $MinSAT(A) = n$ implies there is some truth assignment $e$ for which $|\{\phi \in A : e(\phi) = 0\}| = n$. As we did in Lemma 4.4 for $CNF_{minSAT_e}$, we can easily check that $|\{c \in \mathbf{SC}(CNF_{minSAT_t}(A)) : e'(c) = 0\}| = n$ too. Indeed, since $e'(y_\psi) = e(\psi)$ for all subformula $\psi$ of formulas in $A$, clearly all hard clauses are satisfied under $e'$. Moreover, since the multiset of soft clauses is formed exactly by the singletons $y_\phi$ for each $\phi \in A$ (as a multiset), then necessarily $|\{c \in \mathbf{SC}(CNF_{minSAT_t}(A)) : e'(c) = 0\}| = n \leq MinSAT(CNF_{minSAT_t}(A))$, since MinSAT is the maximum number of falsifiable formulas. Thus, $MinSAT(A) \leq MinSAT(CNF_{minSAT_t}(A))$.

On the other hand, similarly again to the case for $CNF_{minSAT_e}$, to prove $\geq$ observe that, by definition, $MinSAT(CNF_{minSAT_t}(A)) = |\{c \in \mathbf{SC}(CNF_{minSAT_t}(A)) : g(c) = 0\}|$ for a certain truth assignment $g$ that moreover satisfies all hard clauses in $CNF_{minSAT_t}(A)$. Let us modify $g$ to cope with the original variables in $A$, simply by letting $g'(p) = g(y_p)$ for all propositional variables $p$ appearing in $A$. Then, as it happens in the usual Tseitin transformation, is immediate that $g'(\phi) = g(y_\phi)$ for all $\phi \in A$. Thus, we have that $|\{c \in \mathbf{SC}(CNF_{minSAT_t}(A)) : g(c) = 0\}| = |\{\phi \in A : g'(\phi) = 0\}| \leq MinSAT(A)$.  $\square$

## 5. Conclusions and Future Work

We proposed the first approach to solve non-clausal MinSAT via its reduction to clausal MinSAT. We defined four clausal transformations ($CNF_{minSAT_d}$, $CNF_{minSAT_e}$, $CNF_{minSAT_i}$ and $CNF_{minSAT_t}$) and proved its correctness. The most immediate future work is to conduct an empirical comparison and extend the results to finite-domain variables [6].

## References

[1]  G. Fiorino. New tableau characterizations for non-clausal MaxSAT problem. *Logic Journal of the IGPL*, 2021.

[2]  C. M. Li and F. Manyà. MaxSAT, hard and soft constraints. In A. Biere, H. van Maaren, and T. Walsh, editors, *Handbook of Satisfiability*, pages 613–631. IOS Press, 2009.

[3]  C. M. Li, F. Manyà, and J. R. Soler. Clausal form transformation in MaxSAT. In *Proceedings of the 49th IEEE International Symposium on Multiple-Valued Logic, ISMVL*, pages 132–137, 2019.

[4]  C. M. Li, F. Manyà, and J. R. Soler. A tableau calculus for non-clausal maximum satisfiability. In *Proceedings of the 28th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods, TABLEAUX*, pages 58–73, 2019.

[5]  C. M. Li, Z. Zhu, F. Manyà, and L. Simon. Optimizing with minimum satisfiability. *Artificial Intelligence*, 190:32–44, 2012.

[6]  F. Manyà. The 2-SAT problem in signed CNF formulas. *Multiple-Valued Logic. An International Journal*, 5(4):307–325, 2000.

[7]  D. A. Plaisted and S. Greenbaum. A structure-preserving clause form translation. *Journal of Symbolic Computation*, 2:293–304, 1986.

[8]  J. R. Soler. *New Solving Techniques for Maximum and Minimum Satisfiability*. PhD thesis, UAB, 2021.

[9]  G. Tseitin. *Studies in Constructive Mathematics and Mathematical Logic, Part II*, chapter On the Complexity of Derivations in the Propositional Calculus, pages 115–125. Steklov Mathematical Inst., 1968.