Artificial Intelligence Research and Development
A. Cortés et al. (Eds.)

doi:10.3233/FAIA220311

A Tableau Calculus for MaxSAT Based on Resolution

Shoulin LI, [a] Jordi COLL, [a] Djamal HABET, [a] Chu-Min LI [a,b] and Felip MANYÀ [c]

[a] *Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France*
[b] *MIS, Université de Picardie, Amiens, France*
[c] *Artificial Intelligence Research Institute (IIIA), CSIC, Bellaterra, Spain*

Abstract. We define a new MaxSAT tableau calculus based on resolution. Given a multiset of propositional clauses ϕ, we prove that the calculus is sound in the sense that if the minimum number of contradictions derived among the branches of a completed tableau for ϕ is m, then the minimum number of unsatisfied clauses in ϕ is m. We also prove that it is complete in the sense that if the minimum number of unsatisfied clauses in ϕ is m, then the minimum number of contradictions among the branches of any completed tableau for ϕ is m. Moreover, we describe how to extend the proposed calculus to solve Weighted Partial MaxSAT.

Keywords. Maximum satisfiability, semantic tableaux, completeness.

1. Introduction

The Satisfiability problem (SAT) is the problem of deciding if there exists a truth assignment for a given propositional formula in conjunctive normal form (CNF) that evaluates the formula to true. An important optimization variant of SAT is Maximum Satisfiability (MaxSAT), which is the problem of finding a truth assignment that minimizes the number of unsatisfied clauses in a multiset of clauses [23]. Note that minimizing the number of unsatisfied clauses is equivalent to maximizing the number of satisfied clauses.

The inference rules applied in SAT are sound if they preserve satisfiability. Nevertheless, such rules are not applicable in MaxSAT because they are usually unsound. Sound MaxSAT inference rules must preserve the minimum number of unsatisfied clauses between the premises and the conclusions. As a consequence, new complete inference systems for MaxSAT have had to be defined [20]. They are MaxSAT extensions of either the resolution rule [33] or semantic tableaux [11,15,34].

This paper presents a new tableau calculus for MaxSAT based on resolution, proves its completeness and defines its extension to Weighted Partial MaxSAT, the case in which some clauses can be declared as hard and soft clauses have an associated weight. The advantage of our calculus is that it can produce shorter proofs than other related approaches in some cases.

Although this work is mainly theoretical, it is worth mentioning that MaxSAT offers a competitive generic problem solving formalism for combinatorial optimization. For example, MaxSAT has been applied to solve optimization problems in real-world domains as diverse as combinatorial testing [1], community detection in complex networks [17], diagnosis [12], group testing [10], planning [35], routing [31], scheduling [5] and team formation [32], among others. Furthermore, there exist efficient branch-and-bound [21] and SAT-based MaxSAT solvers [4].

The paper is structured as follows. Section 2 defines basic concepts. Section 3 reviews the related work. Section 4 defines a novel MaxSAT tableau calculus based on resolution and proves its completeness. Section 5 defines an extension of the proposed calculus to Weighted Partial MaxSAT. Section 6 gives the conclusions.

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 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. The infinity weight is denoted by $\top$. A weighted partial MaxSAT instance is a multiset of weighted clauses $\phi = \{(h_1,\top),\ldots,(h_k,\top),(c_1,w_1),\ldots,(c_m,w_m)\}$, where the first k clauses are hard and the last m clauses are soft. A soft clause (c,w) is equivalent to having w copies of the clause $(c,1)$, and $\{(c,w_1),(c,w_2)\}$ is equivalent to (c,w_1+w_2). For simplicity, in what follows, we omit weights when all the soft clauses have the same weight.

A truth assignment assigns to each propositional variable either 0 (false) or 1 (true). Weighted Partial MaxSAT, or WPMaxSAT, for an instance ϕ is the problem of finding an assignment that satisfies all the hard clauses and minimizes the sum of the weights of the unsatisfied soft clauses; such an assignment is said to be an optimal assignment.

The Weighted MaxSAT problem, or WMaxSAT, is WPMaxSAT when there are no hard clauses. The Partial MaxSAT problem, or PMaxSAT, is WPMaxSAT when all the soft clauses have the same weight. The (Unweighted) MaxSAT problem is PMaxSAT when there are no hard clauses. The SAT problem, or SAT, is PMaxSAT when there are no soft clauses.

Minimum Satisfiability (MinSAT) is the dual problem of MaxSAT and its goal is to find an assignment that maximizes the number of unsatisfied clauses. The most general extension of MinSAT is Weighted Partial MinSAT, or WPMinSAT, whose goal is to find an assignment that satisfies all the hard clauses and maximizes the sum of the weights of the unsatisfied soft clauses.

3. Related Work

The fact that unit propagation could not be used to simplify CNF formulas in branch-and-bound MaXSAT solvers led to the definition of incomplete resolution-based inference rules for MaxSAT [18,24,25] and of a complete MaxSAT resolution rule [7,8,16]. More recently, the proof complexity community has drawn the attention to MaxSAT resolution with the aim of defining a stronger proof system than SAT resolution. For example, MaxSAT resolution with the split rule (replace clause C with $\neg x \vee C$ and $x \vee C$) produces polynomial-size proofs of the pigeon hole principle, and this does not happen if MaxSAT resolution is replaced with SAT resolution [6,19]. MaxSAT resolution has also been used in a MinSAT branch-and-bound solver [30] and a variable elimination algorithm for MinSAT has been defined in [22,29].

The first tableau calculus for MaxSAT was defined in [27] and then it was extended to non-clausal MaxSAT [13,28]. These works inspired the creation of a complete natural deduction calculus for MaxSAT [9] and tableau calculi for MinSAT [2,3,14,26].

Compared with existing tableau calculi, the calculus of this paper does not need to expand all the clauses in a branch to detect all the possible contradictions. It only needs to expand those clauses containing a complementary literal in another clause of the branch.

4. A MaxSAT tableau calculus based on resolution

We define a MaxSAT tableau calculus and prove its soundness and completeness.

Definition 4.1. *A tableau is a tree with a finite number of branches whose nodes are labelled by either a clause or a box ($\Box$). A box in a tableau denotes a contradiction. A branch is a maximal path in a tree, and we assume that branches have a finite number of nodes.*

Definition 4.2. *Let $\phi = \{\phi_1, \ldots, \phi_m\}$ be a multiset of clauses, l a literal, and D and D' disjunction of literals. A tableau for ϕ is constructed by a sequence of applications of the following rules:*

Initialize *A tree with a single branch with m nodes such that each node is labelled with a clause of ϕ is a tableau for ϕ. Such a tableau is called the initial tableau and its clauses are declared to be active.*

Given a tableau T for ϕ and a branch b of T,

- *Res*-**rule** *If b contains two active clauses with complementary literals, $l \vee D$ and $\neg l \vee D'$, the tableau obtained by appending a new left branch with two nodes below b labelled with $\neg l$ and D and a new right branch with two nodes below b labelled with l and D' is a tableau for ϕ. Clauses $l \vee D$ and $\neg l \vee D'$ become inactive in b and the added clauses are declared to be active.*
- *Unit*-**rule** *If b contains an active unit clause l and an active non-unit clause $\neg l \vee D$, the tableau obtained by appending a new left node below b labelled with $\Box$ and a new right node with two nodes below b labelled with l and D is a tableau for ϕ. Clauses l and $\neg l \vee D$ become inactive in b and the added non-empty clauses are declared to be active.*
- $\Box$-**rule** *If b contains two active and complementary unit clauses, l and $\neg l$, the tableau obtained by appending a node below b labelled with $\Box$ is a tableau for ϕ. Clauses l and $\neg l$ become inactive in b.*

The expansion rules of the previous definition are summarized in Figure 1. Note that all the rules preserve the number of premises falsified by an assignment I in at least one branch and do not decrease that number in the other branch (if any).

Definition 4.3. *Let T be a tableau for a multiset of propositional clauses ϕ. A branch b of T is saturated when no further expansion rules can be applied on b, and T is completed when all its branches are saturated. The cost of a saturated branch is the number of boxes on the branch. The cost of a completed tableau is the minimum cost among all its branches.*

$$
\begin{array}{c|c} \multicolumn{2}{c}{l \vee D} \\ \multicolumn{2}{c}{\neg l \vee D'} \\ \hline \neg l & l \\ D & D' \\ \multicolumn{2}{c}{\textit{Res}\text{-rule}} \end{array}
\qquad
\begin{array}{c|c} \multicolumn{2}{c}{l} \\ \multicolumn{2}{c}{\neg l \vee D} \\ \hline \Box & l \\ & D \\ \multicolumn{2}{c}{\textit{Unit}\text{-rule}} \end{array}
\qquad
\begin{array}{c} l \\ \neg l \\ \hline \Box \\ \Box\text{-rule} \end{array}
$$

Figure 1. Tableau expansion rules for non-clausal MaxSAT

Notice that a branch becomes saturated when it does not contain active clauses with complementary literals. We show below that the minimum number of clauses that can be unsatisfied in a multiset of propositional clauses ϕ is m iff the cost of a completed tableau for ϕ is m. Thus, the systematic construction of a completed tableau for ϕ provides an exact method for MaxSAT.

Example 4.4. *Figure 2 shows how we can create a tableau, with the previous calculus, to prove that the minimum number of unsatisfied clauses in the multiset of clauses* $\{x_1, x_2, \neg x_1 \vee x_3, \neg x_1 \vee \neg x_2 \vee \neg x_3\}$ *is one. We first create the initial tableau (the leftmost tableau) and then apply the Res-rule to the clauses* $\neg x_1 \vee x_3$ *and* $\neg x_1 \vee \neg x_2 \vee \neg x_3$ *(resolving variable* x_3*), getting as a result the second tableau in the figure. We apply the* $\Box$*-rule to* x_1 *and* $\neg x_1$ *on the leftmost branch and obtain the third tableau. That branch is now saturated because its current active clauses (*x_2 *and* $\neg x_3$*) do not contain complementary literals. Then, we apply the* $Unit$*-rule to the clauses* x_1 *and* $\neg x_1 \vee \neg x_2$ *(resolving variable* x_1*) on the rightmost branch, getting as a result the fourth tableau whose middle branch is saturated (current active clauses:* x_2 *and* x_3*). Finally, we apply the* $\Box$*-rule to* x_2 *and* $\neg x_2$ *on the rightmost branch and this branch becomes also saturated (current active clauses:* x_1 *and* x_3*). Since the minimum number of boxes among the branches of the last tableau is one, the minimum number of clauses that can be unsatisfied in* ϕ *is also one.*

The advantage of the defined calculus with respect to other MaxSAT tableau calculi [13,27,28] is that it does not need to expand all the active clauses to saturate a branch. It only needs to expand those active clauses containing a complementary literal in another active clause of the branch. This implies, in some cases, that the resulting tableaux have fewer nodes. For instance, the other calculi need to double the number of branches to solve the multiset of clauses of Example 4.4.

4.1. Soundness and completeness

We prove the soundness and completeness of the proposed tableau calculus for MaxSAT. Before presenting the completeness theorem, we prove termination and the soundness of the expansion rules.

Proposition 4.5. *A tableau for a multiset of propositional clauses* ϕ *is completed in a finite number of steps.*

Proof. We first create an initial tableau and then apply expansion rules in the newly created branches until they become saturated. The *Res*- and $Unit$-rule reduce the number

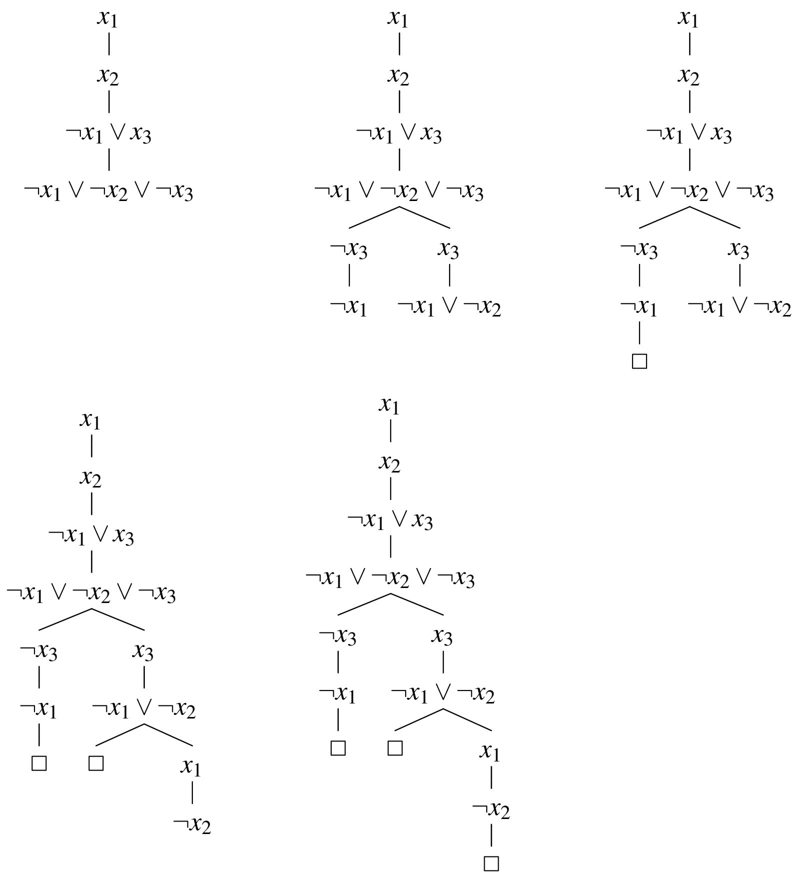

Figure 2. A tableaux for the non-clausal MaxSAT instance $\{x_1, x_2, \neg x_1 \vee x_3, \neg x_1 \vee \neg x_2 \vee \neg x_3\}$.

of connectives. Since we began with a finite number of connectives, these rules can only be applied a finite number of times. The $\Box$-rule inactivates two literals and adds a box. Since we began with a finite number of literals and boxes cannot be premises of any expansion rule, this rule can only be applied a finite number of times. Hence, the construction of any completed tableau terminates in a finite number of steps. $\Box$

Proposition 4.6. *An assignment I falsifies k premises of a Res-, Unit-, and $\Box$-rule iff assignment I falsifies k clauses in one branch of the conclusions of the rule and at least k clauses in the other branch (if any).*

Proof. We prove the result for each rule:

- *Res*-rule: An assignment I satisfies both $l \vee D$ and $\neg l \vee D'$ iff I satisfies either l and D' or $\neg l$ and D. In this case, I satisfies the clauses of one branch and falsifies at least one clause of the other branch. In any other case, I falsifies either $l \vee D$ or $\neg l \vee D'$. If I falsifies $l \vee D$, it falsifies exactly one clause (D) of the left branch and at least one clause of the right branch. If I falsifies $\neg l \vee D'$, it falsifies exactly one clause (D') of the right branch and at least one clause of the left branch. Hence, the number of unsatisfied clauses is preserved in at least one branch of the rule.

- *Unit*-rule: An assignment I satisfies both l and $\neg l \vee D$ iff I satisfies l and D, and in this case I satisfies the two clauses of the right branch. In any other case, I falsifies either l or $\neg l \vee D$. So, I falsifies the left branch and at least one clause of the right branch. Hence, the number of unsatisfied clauses is preserved in at least one branch of the rule.
- $\Box$-rule: An assignment I either falsifies l or $\neg l$, and satisfies the complementary literal of the unsatisfied literal. Since the single conclusion is a box and denotes a contradiction, I falsifies the same number of clauses in the premises and the conclusion.

$\Box$

Theorem 4.7. Soundness & completeness. *The cost of a completed tableau for a multiset of clauses ϕ is m iff the minimum number of unsatisfied clauses in ϕ is m.*

Proof. (*Soundness:*) T was derived by creating a sequence of tableaux $T_0,\ldots,T_n$ ($n \geq 0$) such that T_0 is an initial tableau for ϕ, $T_n = T$, and T_i was obtained by a single application of the *Res*-, *Unit*- or $\Box$-rule on an branch of T_{i-1} for $i = 1,\ldots,n$. By Proposition 4.5, we know that such a sequence is finite. Since T has cost m, T_n contains one branch b with exactly m boxes and the rest of branches contain at least m boxes. Moreover, the active clauses in every branch of T_n do not contain complementary literals; otherwise, we could yet apply expansion rules and T_n could not be completed. The assignment that sets to true the literals occurring in the active clauses of an optimal branch only falsifies the m boxes and there cannot be any assignment satisfying less than m clauses in a branch of T_n because each branch contains at least m boxes. Therefore, the minimum number of active clauses than can be unsatisfied among the branches of T_n is m.

Proposition 4.6 guarantees that the minimum number of unsatisfied active clauses is preserved in the sequence of tableaux $T_0,\ldots,T_n$. Thus, the minimum number of unsatisfied active clauses in T_0 is also m. Since T_0 is formed by a single branch that only contains the clauses in ϕ and all these clauses are active, the minimum number of clauses that can be unsatisfied in ϕ is m.

(*Completeness:*) Assume that there is a completed tableau T for ϕ that does not have cost m. We distinguish two cases:

(i) T has a branch b of cost k, where $k < m$. Then, T has a branch with k boxes and a satisfiable multiset of active clauses because T is completed. This implies that the minimum number of unsatisfied active clauses among the branches of T is at most k. By Proposition 4.6, this also holds for T_0, but this is in contradiction with m being the minimum number of unsatisfied clauses in ϕ because $k < m$. Thus, any branch of T has at least cost m.

(ii) T has no branch of cost m. This is in contradiction with m being the minimum number of unsatisfied clauses in ϕ. Since the tableau expansion rules preserve the minimum number of unsatisfied clauses and the branches of any completed tableau only contain active clauses that are boxes or clauses without complementary literals, T must have a saturated branch with m boxes. Thus, T has a branch of cost m.

Hence, each completed tableau T for a multiset of clauses ϕ has cost m if the minimum number of clauses that can be unsatisfied in ϕ is m. $\Box$

$$\begin{array}{c} (l \vee D, \top) \\ \hline \begin{array}{c|c} (l, \top) & (D, \top) \end{array} \\ \beta\text{-rule} \end{array} \qquad \begin{array}{c} (l \vee D, w_1) \\ (\neg l \vee D', w_2) \\ \hline \begin{array}{c|c} (l \vee D, w_1 - w) & (l \vee D, w_1 - w) \\ (\neg l \vee D', w_2 - w) & (\neg l \vee D', w_2 - w) \\ (\neg l, w) & (l, w) \\ (D, w) & (D', w) \end{array} \\ \text{where } \; w = \min(w_1, w_2) \\ Res\text{-rule} \end{array}$$

$$\begin{array}{c} (l, \top) \\ (\neg l \vee D, w) \\ \hline (D, w) \end{array} \qquad \begin{array}{c} (l, w_1) \\ (\neg l \vee D, w_2) \\ \hline \begin{array}{c|c} (l, w_1 - w) & (l, w_1 - w) \\ (\neg l \vee D, w_2 - w) & (\neg l \vee D, w_2 - w) \\ (\Box, w) & (l, w) \\ & (D, w) \end{array} \\ \text{where } \; w = \min(w_1, w_2) \end{array}$$

$$Unit\text{-rule}$$

$$\begin{array}{c} (l, \top) \\ (\neg l, \top) \\ \hline \blacksquare \end{array} \qquad \begin{array}{c} (l, \top) \\ (\neg l, w) \\ \hline (\Box, w) \end{array} \qquad \begin{array}{c} (l, w_1) \\ (\neg l, w_2) \\ \hline (\Box, w) \\ (l, w_1 - w) \\ (\neg l, w_2 - w) \\ \text{where } \; w = \min(w_1, w_2) \end{array}$$

$$\Box\text{-rule}$$

Figure 3. Tableau expansion rules for Weighted Partial MaxSAT

5. A Tableau Calculus for Weighted Partial MaxSAT based on Resolution

Dealing with weighted soft clauses can be understood as collapsing several unweighted MaxSAT inferences into a single inference, because a weighted clause (C, w) can be replaced by w copies of the unweighted clause C. If there are two premises (C_1, w_1) and (C_2, w_2) with different weights ($w_1 \neq w_2$), (C_1, w_1) and (C_2, w_2) become inactive but $(C_1, w_1 - w)$ and $(C_2, w_2 - w)$, where $w = \min(w_1, w_2)$, are added as active clauses (clauses with weight 0 are not added). Then, the conclusions of the inference have weight w. For example, from $(x_1, 1)$ and $(\neg x_1, 3)$ we derive $(\Box, 1)$ and $(\neg x_1, 2)$.

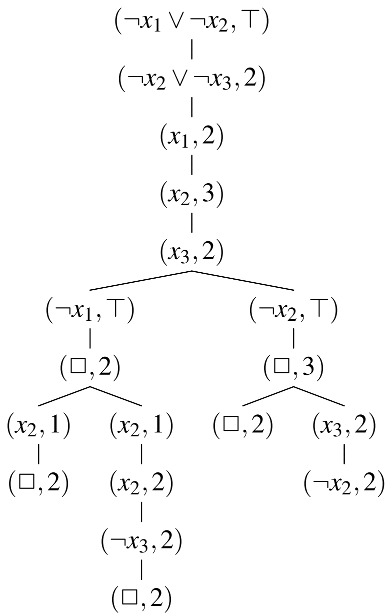

Figure 4. A tableau for the WPMaxSAT instance $\{(\neg x_1 \vee \neg x_2, \top), (\neg x_2 \vee \neg x_3, 2), (x_1, 2), (x_2, 3), (x_3, 2)\}$

When dealing with hard clauses, the inference applied in SAT remains valid in MaxSAT when the premises are hard. Moreover, the detection of a contradiction between two hard clauses implies that we have identified an infeasible solution. In this case, the contradiction is represented by $\blacksquare$ and the branch containing that contradiction can be pruned.

Figure 3 displays the expansion rules of a complete tableau calculus for Weighted Partial MaxSAT. It is formed by the extensions of the *Res*-, *Unit*-, and $\Box$-rules when their premises contain unit hard clauses or weighted soft clauses. In the case in which we have a non-unit hard clause, we can use the β-rule. In our calculus, unit hard clauses are always active while non-unit hard premises become inactive after applying an inference rule. Notice that other inference rules could be used to deal with hard premises but we used the β-rule because it produces a simple and complete calculus.

Example 5.1. *Figure 4 displays a tableau for the WPMaxSAT instance* $\{(\neg x_1 \vee \neg x_2, \top), (\neg x_2 \vee \neg x_3, 2), (x_1, 2), (x_2, 3), (x_3, 2)\}$. *Firstly, we apply the* β*-rule to* $(\neg x_1 \vee \neg x_2, \top)$. *Secondly, we apply the* $\Box$*-rule to* $(\neg x_1, \top)$ *and* $(x_1, 2)$ *in the left branch. Thirdly, we apply the* $\Box$*-rule to* $(\neg x_2, \top)$ *and* $(x_2, 3)$ *in the right branch. Fourthly, we apply the Unit-rule to* $(\neg x_2 \vee \neg x_3, 2)$ *and* $(x_2, 3)$ *in the left branch. Fifthly, we apply the* $\Box$*-rule to* $(\neg x_3, 2)$ *and* $(x_3, 2)$ *in the second leftmost branch. Sixthly, we apply the Unit-rule to* $(\neg x_2 \vee \neg x_3, 2)$ *and* $(x_3, 2)$ *in the right branch. Since the minimum cost among all the branches is 3, the minimum sum of weights of the unsatisfied soft clauses while satisfying the hard clauses is 3.*

Taking into account that a soft clause (c,w) is equivalent to having w copies of clause $(c,1)$, and $\{(c,w_1),(c,w_2)\}$ is equivalent to (c,w_1+w_2), we can prove that the previous calculus is complete for WPMaxSAT.

6. Conclusions

We presented a new tableau calculus for MaxSAT based on resolution, proved its completeness and defined its extension to WPMaxSAT. The proposed calculus has the advantage of producing shorter proofs in some cases. Moreover, this work is a step forward to better understanding the logic of MaxSAT. In future work, we plan to extend the calculus to non-clausal MaxSAT, MinSAT and first-order logic.

Acknowledgements: This work has been supported by the French Agence Nationale de la Recherche, reference ANR-19-CHIA-0013-01, and grant PID2019-111544GB-C21 funded by MCIN/AEI/10.13039/501100011033. The last author was supported by mobility grant PRX21/00488 of the Spanish *Ministerio de Universidades*.

References

[1] C. Ansótegui, F. Manyà, J. Ojeda, J. M. Salvia, and E. Torres. Incomplete MaxSAT approaches for combinatorial testing. *Journal of Heuristics*, (in press), 2022.

[2] J. Argelich, C. M. Li, F. Manyà, and J. R. Soler. Clause branching in MaxSAT and MinSAT. In *Proceedings of the 21st International Conference of the Catalan Association for Artificial Intelligence, Roses, Spain*, volume 308 of *Frontiers in Artificial Intelligence and Applications*, pages 17–26. IOS Press, 2018.

[3] J. Argelich, C. M. Li, F. Manyà, and J. R. Soler. Clause tableaux for maximum and minimum satisfiability. *Logic Journal of the IGPL*, 29(1):7–27, 2021.

[4] F. Bacchus, M. Järvisalo, and M. Ruben. Maximum satisfiability. In A. Biere, M. Heule, H. van Maaren, and T. Walsh, editors, *Handbook of Satisfiability*, pages 929–991. IOS Press, 2021.

[5] M. Bofill, J. Coll, M. Garcia, J. Giráldez-Cru, G. Pesant, J. Suy, and M. Villaret. Constraint solving approaches to the business-to-business meeting scheduling problem. *Journal of Artificial Intelligence Research*, 74:263–301, 2022.

[6] M. L. Bonet and J. Levy. Equivalence between systems stronger than resolution. In *Proceedings of the 23rd International Conference on Theory and Applications of Satisfiability Testing, SAT-2020, Alghero, Italy*, pages 166–181. Springer LNCS 12178, 2020.

[7] M. L. Bonet, J. Levy, and F. Manyà. A complete calculus for Max-SAT. In *Proceedings of the 9th International Conference on Theory and Applications of Satisfiability Testing, SAT-2006, Seattle, USA*, pages 240–251. Springer LNCS 4121, 2006.

[8] M. L. Bonet, J. Levy, and F. Manyà. Resolution for Max-SAT. *Artificial Intelligence*, 171:240–251, 2007.

[9] J. Casas-Roma, A. Huertas, and F. Manyà. Solving MaxSAT with natural deduction. In *Proceedings of the 20th International Conference of the Catalan Association for Artificial Intelligence, Deltebre, Spain*, volume 300 of *Frontiers in Artificial Intelligence and Applications*, pages 186–195. IOS Press, 2017.

[10] L. Ciampiconi, B. Ghosh, J. Scarlett, and K. S. Meel. A MaxSAT-based framework for group testing. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 10144–10152, 2020.

[11] M. D'Agostino. Tableaux methods for classical propositional logic. In M. D'Agostino, D. Gabbay, R. Hähnle, and J. Posegga, editors, *Handbook of Tableau Methods*, pages 45–123. Kluwer, 1999.

[12] D. D'Almeida and É. Grégoire. Model-based diagnosis with default information implemented through MAX-SAT technology. In *Proceedings of the IEEE 13th International Conference on Information Reuse & Integration, IRI, Las Vegas, NV, USA*, pages 33–36, 2012.

[13] G. Fiorino. New tableau characterizations for non-clausal MaxSAT problem. *Logic Journal of the IGPL*, 2021. doi:10.1093/jigpal/jzab012.

[14] G. Fiorino. A non-clausal tableau calculus for MinSAT. *Information Processing Letters*, 173:106167, 2022.

[15] R. Hähnle. Tableaux and related methods. In J. A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, pages 100–178. Elsevier and MIT Press, 2001.

[16] F. Heras and J. Larrosa. New inference rules for efficient Max-SAT solving. In *Proceedings of the National Conference on Artificial Intelligence, AAAI-2006, Boston/MA, USA*, pages 68–73, 2006.

[17] S. Jabbour, N. Mhadhbi, B. Raddaoui, and L. Sais. A SAT-based framework for overlapping community detection in networks. In *Proceedings of the 21st Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, Part II, PAKDD, Jeju, South Korea*, pages 786–798, 2017.

[18] J. Larrosa and F. Heras. Resolution in Max-SAT and its relation to local consistency in weighted CSPs. In *Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI-2005, Edinburgh, Scotland*, pages 193–198. Morgan Kaufmann, 2005.

[19] J. Larrosa and E. Rollon. Towards a better understanding of (partial weighted) MaxSAT proof systems. In *Proceedings of the 23rd International Conference on Theory and Applications of Satisfiability Testing, SAT-2020, Alghero, Italy*, pages 218–232. Springer LNCS 12178, 2020.

[20] C. Li and F. Manyà. Inference in MaxSAT and MinSAT. In *The Logic of Software*, volume 13360 of *LNCS*, pages 350–369. Springer, 2022.

[21] C. Li, Z. Xu, J. Coll, F. Manyà, D. Habet, and K. He. Combining clause learning and branch and bound for MaxSAT. In *Proceedings of the 27th International Conference on Principles and Practice of Constraint Programming, CP, Montpellier, France*, volume 210 of *LIPIcs*, pages 38:1–38:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[22] C. M. Li and F. Manyà. An exact inference scheme for MinSAT. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence, IJCAI, Buenos Aires, Argentina*, pages 1959–1965, 2015.

[23] C. M. Li and F. Manyà. MaxSAT, hard and soft constraints. In A. Biere, M. Heule, H. van Maaren, and T. Walsh, editors, *Handbook of Satisfiability*, pages 903–927. IOS Press, 2021.

[24] C. M. Li, F. Manyà, N. O. Mohamedou, and J. Planes. Resolution-based lower bounds in MaxSAT. *Constraints*, 15(4):456–484, 2010.

[25] C. M. Li, F. Manyà, and J. Planes. New inference rules for Max-SAT. *Journal of Artificial Intelligence Research*, 30:321–359, 2007.

[26] C. M. Li, F. Manyà, and J. R. Soler. A clause tableau calculus for MinSAT. In *Proceedings of the 19th International Conference of the Catalan Association for Artificial Intelligence, CCIA, Barcelona, Spain*, volume 288 of *Frontiers in Artificial Intelligence and Applications*, pages 88–97. IOS Press, 2016.

[27] C. M. Li, F. Manyà, and J. R. Soler. A clause tableaux calculus for MaxSAT. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence, IJCAI, New York, USA*, pages 766–772, 2016.

[28] 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, London, UK*, pages 58–73. Springer LNCS 11714, 2019.

[29] C. M. Li, F. Xiao, and F. Manyà. A resolution calculus for MinSAT. *Logic Journal of the IGPL*, 29(1):28–44, 2021.

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

[31] Y. Li, S. Lin, S. Nishizawa, and H. Onodera. Mcell: Multi-row cell layout synthesis with resource constrained MAX-SAT based detailed routing. In *IEEE/ACM International Conference On Computer Aided Design*, pages 157:1–157:8, 2020.

[32] F. Manyà, S. Negrete, C. Roig, and J. R. Soler. Solving the team composition problem in a classroom. *Fundamamenta Informaticae*, 174(1):83–101, 2020.

[33] J. A. Robinson. A machine oriented logic based on the resolution principle. *Journal of the ACM*, 12(1):23–41, 1965.

[34] R. Smullyan. *First-Order Logic*. Dover Publications, New York, second corrected edition, 1995. First published 1968 by Springer-Verlag.

[35] L. Zhang and F. Bacchus. MAXSAT heuristics for cost optimal planning. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence, Toronto, Ontario, Canada*, pages 1846–1852, 2012.