NEW RESEARCH LINES FOR MAX-SAT
Exploiting the Recent Resolution Rule for Max-SAT

Federico Heras
Barcelona, Spain

Keywords: Max-SAT, Combinatorial optimization, Inference, Local search, Systematic search, Logic programming.

Abstract: This paper presents the current state-of-the-art techniques for Max-SAT solving and points out new research lines in order to exploit the benefits of the novel resolution rule for Max-SAT.

1 INTRODUCTION

Given a propositional formula in conjunctive normal form (*CNF*), determining whether there exists a variable assignment such that the formula evaluates to true is called the Boolean Satisfiability Problem, commonly abbreviated as SAT.

Max-SAT is the optimization variant of SAT: Given a propositional formula , the *unweighted Max-SAT* problem then is to find a variable assignment that maximizes the number of satisfied clauses. In *weighted Max-SAT*, each clause has an associated *weight* and the goal is to maximize the total weight of the satisfied clauses.

In this paper we focus on (weighted) Max-SAT which is a well-known *NP Hard* problem. It is considered one of the fundamental combinatorial optimization problems and many important problems can be naturally expressed as Max-SAT. They include academic problems such as *Max-Cut* or *Max-Clique*, as well as real problems in domains like *routing*, *bioinformatics*, *electronic markets*, etc.

Search methods for Max-SAT solving are usually classified in two categories: *systematic* and *local search*. For a given Max-SAT instance, a systematic search algorithm traverse the whole search space and can either find the optimal solution or proves that no solution exists. This is why they are considered *complete*. On the other hand, a local search algorithm initially selects a point of the search space, and moves from the current solution to a neighbor candidate. Then, it stops when a solution is found or when a limit is reached such as a time limit or a maximum number of steps. Typically, local search solvers are *incomplete*, because they do not assure to find a solution neither to prove its optimality.

An alternative way to solve problems is *inference*. The aim of inference is to make explicit some implicit information from the problem instance. Algorithms purely based on inference are complete as systematic search. However, they are prohibitive because of their high memory requirements (Rish and Dechter, 2000). In practice, limited forms of inference are applied inside search algorithms in order to reduce the number of search steps. This small amount of inference is usually called *incomplete (or limited) inference*. Recently, a general inference mechanism for Max-SAT was introduced in (Larrosa et al., 2008) which extends classical resolution. It is called the resolution rule for Max-SAT. It is a *sound* and *complete* (Larrosa et al., 2008; Bonet et al., 2007) rule.

Modern Max-SAT solvers based on systematic search follow the well-know *branch and bound* scheme and they apply a limited number of Max-SAT resolution steps during search in order to simplify the problem (Larrosa et al., 2008; Heras et al., 2008; Li et al., 2007). As a consequence, the search process is boosted several orders of magnitude. The limited resolution process affects to mutually inconsistent subsets of clauses that are detected during search by patterns or via *unit propagation* (Larrosa et al., 2008; Li et al., 2007; Heras et al., 2008). When we analyze current works on Max-SAT we observe they are refinements to improve the pattern detection mechanisms or the effectiveness of unit propagation and to obtain better results (Lin et al., 2008).

The aim of this paper is to propose new solving techniques based on the recent resolution rule for Max-SAT. First, we study the viability of a complete inference-based algorithm. Second, we propose new

Heras F. (2010).
NEW RESEARCH LINES FOR MAX-SAT - Exploiting the Recent Resolution Rule for Max-SAT.
In *Proceedings of the 2nd International Conference on Agents and Artificial Intelligence - Artificial Intelligence*, pages 648-651
DOI: 10.5220/0002761706480651

methods of limited inference. Finally, we propose to use inference in order to improve the performance of local search algorithms.

2 PRELIMINARIES

In the sequel $X = \{x_1, x_2, \ldots, x_n\}$ is the set of boolean variables. A *literal* is either a variable x_i or its negation $\bar{x_i}$. The variable to which literal l refers is noted $var(l)$. Given a literal l, its negation $\bar{l}$ is $\bar{x_i}$ if l is x_i and is x_i if l is $\bar{x_i}$. A *clause* C is a disjunction of literals. In the following, possibly subscripted capital letters A, B, C, D, and E will always represent clauses. The *size* of a clause is the number of literals that it has. An *assignment* is a set of literals not containing a variable and its negation. Assignments of maximal size n are called *complete*, otherwise they are called *partial*. An assignment *satisfies* a literal iff it belongs to the assignment, it satisfies a clause iff it satisfies one or more of its literals and it *falsifies* (or *unsatisfies*) a clause iff it contains the negation of all its literals. In the latter case we say that the clause is *conflicting* as it always happens with the empty clause, noted $\square$.

A *weighted* clause is a pair (C, w), where C is a clause and w is the cost of its falsification, also called its *weight*. Many real problems contain clauses that *must* be satisfied. We call such clauses *mandatory* or *hard* and associate with them a special weight $\top$. Non-mandatory clauses are also called *soft* clauses. A *weighted formula* in *conjunctive normal form* (WCNF) $\mathcal{F}$ is a multiset of weighted clauses. A *model* is a complete assignment that satisfies all mandatory clauses. The *cost of an assignment* is the sum of weights of the clauses that it falsifies. Given a WCNF formula, *Weighted* Max-SAT is the problem of finding a model of minimum cost. Note that if a formula contains only mandatory clauses, weighted Max-SAT is equivalent to classical SAT.

Following (Larrosa et al., 2008; Bonet et al., 2007), the resolution rule can be extended from SAT to Max-SAT as,

$$\{(x \vee A, u), (\bar{x} \vee B, w)\} \equiv \left\{ \begin{array}{l} (A \vee B, m), \\ (x \vee A, u - m), \\ (\bar{x} \vee B, w - m), \\ (x \vee A \vee \bar{B}, m), \\ (\bar{x} \vee \bar{A} \vee B, m) \end{array} \right\}$$

where $m = \min\{u, w\}$.

The identification of mandatory clauses with $\top$ allows to extend some well-known simplification rules from SAT to Max-SAT such as *unit propagation*. Unit propagation for Max-SAT (Larrosa et al., 2008) can be applied as in SAT (Silva and Sakallah, 1996) only when unit hard clauses exist on the formula. Given a unit hard clause $(l, \top)$, literal l is *propagated* which means that it is assigned accordingly to satisfy the clause. Such an assignment can falsify literals in other clauses that may become also hard unit clauses. The procedure is repeated until no more hard unit clauses are generated or until a *conflict* is detected, that is, a hard clause is falsified.

We say that a weighted formula $\mathcal{F}'$ is a *relaxation* $\mathcal{F}$ (noted $\mathcal{F}' \sqsubseteq \mathcal{F}$) if the optimal cost of $\mathcal{F}'$ is less than or equal to the optimal cost in $\mathcal{F}$ (non-models are considered to have cost infinity). We say that two weighted formulas $\mathcal{F}'$ and $\mathcal{F}$ are *equivalent* (noted $\mathcal{F}' \equiv \mathcal{F}$) if $\mathcal{F}' \sqsubseteq \mathcal{F}$ and $\mathcal{F} \sqsubseteq \mathcal{F}'$.

Given a SAT or Max-SAT formula, a subset of (weighted) clauses is *inconsistent* if all them cannot be simultaneusly satisfied by any variable assignment. The clauses involved in an inconsistent subset of clauses can be determined with several steps of resolution. This process is called *refutation*. If all the clauses involved in a refutation are used at most once, it is called a *tree-like refutation*, otherwise it is considered a *general refutation*.

Given an unsatisfiable SAT formula (i.e. a Max-SAT formula that only contains hard clauses), a subset of clauses whose conjunction is still unsatisfiable is called an *unsatisfiable core* of the original formula. Unsatisfiability cores of SAT problem instances are currently provided with general refutations, that is, existing procedures do not assure to use each clause just once at most (Liffiton and Sakallah, 2008).

3 NEW RESEARCH LINES

In this Section we propose different research venues that exploit the novel resolution rule for Max-SAT. Most recent works for Max-SAT present refinements to improve the performance of the works in (Chu Min Li and Planes, 2005; Li et al., 2007; Larrosa et al., 2008; Heras et al., 2008). In what follows, we propose new methods based on complete inference, incomplete inference and local search.

3.1 Complete Inference

It would be interesting to investigate the effectiveness of a complete inference-based algorithm for Max-SAT and how it compares with local and systematic search. It is well-known that this approach has severe memory limitations but it may be effective for problem instances with a very small *induced width*. The induced width is a property associated to each

problem instance. Its computation implies an NP-Hard problem but accurate approximations are known (Rish and Dechter, 2000). Given the information provided by the induced width we can decide whether a problem instance can be solved only with inference or not.

3.2 Incomplete Inference

To start with, we can limit the complete method introduced in Section 3.1 using the value of the induced width. Then, we can use it as a pre-processing or during systematic search when the induced width of the problem resulting of some variable assignments is small enough. An initial work in this field can be found Max-SAT in (Argelich et al., 2008). However, from previous approaches in other frameworks, it seems that this approach is not very competitive in general (Rish and Dechter, 2000).

Current approaches for solving the SAT problems apply some kind of incomplete (or limited) inference. For example, modern SAT solvers apply unit propagation in order to detect conflicting clauses and then to *learn* new clauses in order to boost the search procedure (Silva and Sakallah, 1996). Also, different preprocessing methods based on limited inference are applied for local (Cha and Iwama, 1996) and systematic search (Eén and Biere, 2005).

Recently some techniques from the SAT comunity have been extended to Max-SAT. First, unit propagation and clause learning can be applied directly to Max-SAT when the clauses involved are hard (Larrosa et al., 2008; Heras et al., 2008) (i.e, all clauses involved are like $(C, \top)$). But, unit propagation is used also in Max-SAT to produce as many empty clauses as possible (i.e., $(\square, m)$). As consequence, the problem is simplified and search is boosted. In (Chu Min Li and Planes, 2005) a method for detecting inconsistent subsets of clauses via unit propagation is presented and then in (Heras et al., 2008) such subset is transformed to an equivalent but simpler subset by applying the resolution rule for Max-SAT. Note that the clauses involved in the inconsistent subset are detected via a tree-like refutation.

Observe that one of the most important differences between the classical resolution rule for SAT and the novel resolution rule for Max-SAT is that clauses in SAT can be used once and again during a resolution process returning an equivalent formula. Differently, in the Max-SAT context clauses may disappear after a resolution step and they should be used only once or at most as many times as their weight is greater than 0 to assure a resulting equivalent formula.

Based on the same idea of using unit propagation to produce new weighted empty clauses, we propose to use techniques of detecting unsatisfiable cores to produce empty clauses. The idea is to apply SAT techniques in order to detect unsatisfiable cores and then apply Max-SAT resolution if and only if the refutation is tree-like to assure the correctness of the transformation. In particular, the Max-SAT problem is solved assuming all clauses are hard with a SAT solver like (Silva and Sakallah, 1996) and the information about the unsatisfiable core is recorded during search. Once the SAT solver has proved the unsatisfiability, a refutation is built using the information of the unsatisfiable core. Then, Max-SAT resolution is applied as dictated by the refutation if and only if it is tree-like. However, we think this is unlikely to happen in general and time can be prohibitive (each detection of an unsatisfiable core requires solving a SAT instance).

A more specific recent work on 2-SAT (i.e., all clauses have size two) (Buresh-Oppenheim and Mitchell, 2006) shows how minimum tree-like refutations can be built for 2-SAT in *polynomial* time. Observe that the same process can be applied for Max-SAT but in order to simplify the formula as much as possible. This may be specially powerful for problems with lots of weighted clauses of arity 2 such as the general *Max-2-SAT problem*, the *Max-CUT* problem or the *spin-glass* problem.

Finally, we propose to study new forms of limited inference for problems with a very specific structure. In (Heras and Larrosa, 2008) we presented preprocessing that exploits the synergy of two resolution-based rules in order to simplify problems by a set of hard clauses of size 2 and a set of soft clauses of size 1. Observe that prominent optimization problems can be encoded in this way such as the *Maximum Clique Problem*, *Minimum Vertex Covering*, *Maximum Indepedent Set*, etc. Results indicated that a systematic search solver is boosted several orders of magnitude after the pre-processing for some problem instances.

3.3 Local Search

The first successful local search solver for SAT was GSAT (Selman et al., 1992) that is a simple best-improvement search algorithm. Performance improvements were achieved by the usage of techniques to avoid falling in *local minima*, that is, areas where no improvements can be reached. Major improvements were obtained with the development of the WALKSAT architecture (Selman et al., 1993). In each search step, WALKSAT algorithms first choose a currently unsatisfied clause and then *flip* a variable occurring in this clause, that is, the value of the variable is changed from true to false or viceversa. Current

research is focused on proposing new schemes for selecting the variable to be *flipped* and mechanisms to avoid local minima. The use of inference in local search is restricted to the classical resolution rule and the SAT problem (Cha and Iwama, 1996).

We propose to apply the recent resolution rule to simplify the problem in local search solvers so that they can obtain better solutions in less search steps or time. In (Heras, 2009) we have recently extended the pre-processing in (Heras and Larrosa, 2008) and the limited inference of (Heras et al., 2008) has also been considered as a pre-processing. Preliminary results indicate that feeding a local search with the pre-processed instance usually improve its performance (Heras, 2009). In particular, algorithms based on the WALKSAT architecture obtain substantially better solutions. Furthermore, the hard instances presented in (Xu et al., 2007) [1] were all solved for the first time.

4 CONCLUSIONS

In this paper we have presented several ways to benefit from the novel resolution rule for Max-SAT (Larrosa et al., 2008). We have pointed out that complete inference-based algorithms may be effective on problems with small induced width. Regarding incomplete inference, we have presented a general method based on detecting unsatisfiable cores but restricted to the case in which a tree-like refutation can be built. We have also proposed incomplete inference techniques based on building minimum tree-like refutations restricted to Max-2-SAT as introduced in (Buresh-Oppenheim and Mitchell, 2006). We have suggested that specific limited inference for problems with a very particular structure can be very powerful in practice (Heras and Larrosa, 2008). Finally, we have argued that local search can also benefit from inference (Heras, 2009).

ACKNOWLEDGEMENTS

The author is grateful to Rafael Gimenez for useful comments.

REFERENCES

Argelich, J., Li, C. M., and Manyà, F. (2008). A preprocessor for max-sat solvers. In *SAT*, pages 15–20.

Bonet, M. L., Levy, J., and Manyà, F. (2007). Resolution for max-sat. *Artif. Intell.*, 171(8-9):606–618.

Buresh-Oppenheim, J. and Mitchell, D. G. (2006). Minimum witnesses for unsatisfiable 2cnfs. In *SAT*, pages 42–47.

Cha, B. and Iwama, K. (1996). Adding new clauses for faster local search. In *Proc. of the 13thAAAI*, pages 332–337, Portland, OR.

Chu Min Li, F. M. and Planes, J. (2005). Exploiting unit propagation to compute lower bounds in branch and bound max-sat solvers. In *Proc. of the 11th CP*, Sitges, Spain.

Eén, N. and Biere, A. (2005). Effective preprocessing in sat through variable and clause elimination. In *SAT*, pages 61–75.

Heras, F. (2009). Max-sat resolution-based pre-processings and their effect on local search solvers. *Submitted*.

Heras, F. and Larrosa, J. (2008). A max-sat inference-based pre-processing for max-clique. In *SAT*, pages 139–152.

Heras, F., Larrosa, J., and Oliveras, A. (2008). Minimaxsat: An efficient weighted max-sat solver. *J. Artif. Intell. Res. (JAIR)*, 31:1–32.

Larrosa, J., Heras, F., and de Givry, S. (2008). A logical approach to efficient max-sat solving. *Artificial Intelligence*, 172(2-3):204–233.

Li, C. M., Manyà, F., and Planes, J. (2007). New inference rules for max-sat. *J. Artif. Intell. Res. (JAIR)*, 30:321–359.

Liffiton, M. H. and Sakallah, K. A. (2008). Algorithms for computing minimal unsatisfiable subsets of constraints. *J. Autom. Reasoning*, 40(1):1–33.

Lin, H., Su, K., and Li, C. M. (2008). Within-problem learning for efficient lower bound computation in max-sat solving. In *AAAI*, pages 351–356.

Rish, I. and Dechter, R. (2000). Resolution versus search: Two strategies for sat. *J. Autom. Reasoning*, 24(1/2).

Selman, B., Kautz, H. A., and Cohen, B. (1993). Local search strategies for satisfiability testing. In *Proceedings of the second DIMACS Challenges on Cliques, Coloring and Satisfiability*.

Selman, B., Levesque, H. J., and Mitchell, D. G. (1992). A new method for solving hard satisfiability problems. In *AAAI*, pages 440–446.

Silva, J. P. M. and Sakallah, K. A. (1996). Grasp - a new search algorithm for satisfiability. In *ICCAD*, pages 220–227.

Xu, K., Boussemart, F., Hemery, F., and Lecoutre, C. (2007). Random constraint satisfaction: Easy generation of hard (satisfiable) instances. *Artif. Intell.*, 171(8-9):514–534.

[1] Available at http://www.nlsde.buaa.edu.cn/ kexu/