Reformulation based MaxSAT robustness (Extended abstract) *

Miquel Bofill[1†], Dídac Busquets[2‡], and Mateu Villaret[1†]

[1] Departament d'Informàtica, Matemàtica Aplicada i Estadística,
Universitat de Girona, Spain
{mbofill,villaret}@imae.udg.edu

[2] Department of Electrical and Electronic Engineering,
Imperial College London, UK
didac.busquets@imperial.ac.uk

Abstract. The presence of uncertainty in the real world makes robustness a desirable property of solutions to Constraint Satisfaction Problems (CSP). A solution is said to be robust if it can be easily repaired when unexpected events happen. This has already been addressed in the frameworks of Boolean satisfiability (SAT) and Constraint Programming (CP). In this paper we consider the unaddressed problem of robustness in weighted MaxSAT, by showing how robust solutions to weighted MaxSAT instances can be effectively obtained via reformulation into pseudo-Boolean formulae. Our encoding provides a reasonable balance between increase in size and performance. We also consider flexible robustness for problems having some unrepairable breakage, in other words, problems for which there does not exist a robust solution.

1 Introduction

Uncertainty is inherent to most real world problems. For instance, in job-shop scheduling, if a machine breaks down, a new solution must be computed. Such a new solution should be fast to compute and, ideally, should also be close to the initial one (e.g., in the job-shop problem, it is not desirable to reassign a large number of tasks). Thus, instead of looking for an optimal solution, which may be brittle and not comply with these two requirements, one could directly look for a solution that can be *easily* repaired (*easy* referring both to time and

*This is a summary of the paper: Miquel Bofill, Dídac Busquets, Víctor Muñoz, Mateu Villaret, "Reformulation based MaxSAT robustness", *Constraints*, April 2013, Volume 18, Issue 2, pp 202-235, doi link: 10.1007/s10601-012-9130-2. A preliminary version of this work was published in: Miquel Bofill, Dídac Busquets, Mateu Villaret, "A declarative approach to robust weighted Max-SAT", *Proceedings of the 12th international ACM SIGPLAN symposium on Principles and practice of declarative programming (PPDP 2010)*.

†Supported by the Spanish Ministry of Science and Innovation (project TIN2012-33042).

‡Supported by the UK EPSRC Grand Challenge Project Autonomic Power System.

number of repairs). Such a solution is said to be *robust*. Obviously, this robust solution may be suboptimal, compared to a non-robust one. This fact has been sometimes called *the price of robustness* [2] but, in many real world situations, it is worth sacrificing some optimality for a stronger solution.

In this paper we consider the unaddressed problem of seeking robust solutions in weighted MaxSAT. MaxSAT is becoming a competitive approach for solving combinatorial optimization problems [11] in the CP framework, as well as to deal with Max-Constraint Satisfaction Problems (Max-CSP) [1].

Existing works on robust solutions for (plain) SAT [4,13] and CP [6,7,8,9,10] are based on two main approaches: reformulation and search-based algorithms. The idea of reformulation is to extend the initial instance so that a solution to the extended instance is a robust solution to the initial one [4]. On the other hand, search-based algorithms look for robust solutions with backtracking, propagation and consistency techniques [6,7,8,9,10]. Previous works [5] claim that, at least for CP, the reformulation approach results in prohibitively large formulas.

In this paper we show that reformulation is still feasible in the setting of weighted MaxSAT. This has several advantages. First, the notion of robustness can be directly expressed in the original formulation of the problem with no need of changing the underlying solving method. Additionally, the reformulation can be easily adapted to interesting extensions of the notion of robustness such as adding dependencies between breakable and repairable variables, or introducing failure probabilities, among others. Contrarily, in search-based approaches, if the notion of robustness is modified, the algorithm must probably be modified too.

2 Weighted MaxSAT Robustness

The following definition generalizes the one of Ginsberg et al. [4] to partial weighted MaxSAT.

Definition 1. *Let F be a partial weighted MaxSAT formula and S_1, S_2 and S_3 be sets of variables occurring in F, such that $(S_1 \cup S_2) \cap S_3 = \emptyset$. A $(S_1^a, S_2^b, S_3, \beta)$-* supermodel *of F is a (minimal cost) model of F such that if we modify the values taken by the variables in a subset of S_1 of size at most* ***a*** *(breakage), then another model can be obtained by modifying the values of the variables in a disjoint subset of S_2 of size at most* ***b*** *(repair) and the values of any number of variables in S_3 (don't-care variables), and moreover the solution and all possible repaired solutions have a cost of at most β. When the set of don't-care variables S_3 is empty we simply talk of (S_1^a, S_2^b, β)-supermodels. Also, if the sets S_1, S_2 and S_3 are unrestricted, we talk of an (a, b, β)-supermodel.*

The idea behind S_3 is that sometimes a formula contains auxiliary or redundant variables, whose values are implied by others, and a change in their values should not be counted neither as a break nor as a repair.

In the following we assume that $F = C \wedge W$ is a partial weighted MaxSAT formula, where C denotes the set of mandatory clauses and W denotes the set of weighted, non-mandatory clauses. W.l.o.g., we assume that W consists only of unary clauses, i.e., $W = (l_1, w_1) \wedge \cdots \wedge (l_k, w_k)$, where l_i is a literal and w_i is a weight for all i in $1..k$. We define $B = \sum_{j \in 1..k} \bar{l}_j \cdot w_j$, which amounts to the cost of the unsatisfied clauses in W.

We now show how we can reformulate an initial formula F to an extended formula F_{SM}, whose solution is a robust solution to F. We achieve this by means of Boolean *cardinality constraints*, which allow us to state, for a given set of Boolean variables E and a given number p, that at most p variables of E can be true [12]. A formula which is only $\mathcal{O}(n^a)$ larger than F is obtained (where n is the number of variables). This is especially important, since it means that the complexity (in size) of our approach does not depend on the number of repairs, but only on the number of breakages, which is usually assumed to be low. This is an important improvement from the encoding in [4], whose size is $\mathcal{O}(n^{a+b})$.

The key idea of the encoding is the following: instead of encoding the different repairs by explicitly flipping (i.e., negating) the variables (as done in [4]), simply rename the variables and restrict the number of variables that can change their value by means of cardinality constraints. As we need a different repair for each possible breakage, a different renaming of the repair (and don't-care) variables is necessary for each possible breakage. To this end, each variable of a repair set R is tagged with the name of the breakage set S which is repairing.

Definition 2. (Variable renaming). *Let R and S be sets of variables. The function $\mathsf{ren}_{R,S} : \mathcal{X} \to \mathcal{X}$ is defined as $\mathsf{ren}_{R,S}(x) = x^S$ for every variable $x \in R$, where x^S is a new atom, and $\mathsf{ren}_{R,S}(x) = x$ if $x \notin R$.*

Definition 3. (Formula renaming). *Let F be a Boolean (or pseudo-Boolean) formula, and R and S be sets of variables. Then $F^{\mathsf{ren}_{R,S}}$ denotes the formula F where all occurrences of each variable x have been replaced by $\mathsf{ren}_{R,S}(x)$.*

Definition 4. (Difference cardinality). *Let R and S be sets of variables, and $\mathsf{ren}_{R,S}$ be a variable renaming function. Then we define the* difference cardinality formula *as* $\nabla^{\mathsf{ren}_{R,S}} = \sum_{x \in R}(x \neq \mathsf{ren}_{R,S}(x)) = \sum_{x \in R}(x \neq x^S)$.

The encoding of F_{SM}, which will give us a $(S_1^a, S_2^b, S_3, \beta)$-supermodel of F, is the following pseudo-Boolean Optimisation (PBO) instance:

$$
\begin{array}{lll}
F_{SM}^{\nabla} = & \textbf{Minimize} & B \\
 & \textbf{Subject To} & C \wedge (B \leq \beta) \wedge \\
 & \displaystyle\bigwedge_{S \subseteq S_1,\, 1 \leq |S| \leq a} & \left(C_{\overline{S}}^{\mathsf{ren}_{(S_2 \setminus S) \cup S_3,\, S}} \wedge B_{\overline{S}}^{\mathsf{ren}_{(S_2 \setminus S) \cup S_3,\, S}} \leq \beta \;\wedge \nabla^{\mathsf{ren}_{S_2 \setminus S,\, S}} \leq b \right)
\end{array}
$$

Roughly speaking, the meaning of the PBO instance F_{SM} is the following: we have first replaced the weighted clauses W of the original formula F by the objective function to be minimized, which corresponds to B, the sum of the weights of the falsified clauses. Then we have C, that is, the mandatory clauses of the original formula, and we add $B \leq \beta$ to bound the cost. Next, we need to be able to repair all possible breakages. The big *and* accounts for all possible (*breakage*) sets S of size smaller than or equal to a. For each of these breakages, we flip the broken variables in the original mandatory clauses and rename those allowed to change in $C_{\overline{S}}^{\mathsf{ren}_{(S_2 \setminus S) \cup S_3,\, S}}$, we bound the cost of the new solution with $B_{\overline{S}}^{\mathsf{ren}_{(S_2 \setminus S) \cup S_3,\, S}} \leq \beta$, and limit the number of repairs with $\nabla^{\mathsf{ren}_{S_2 \setminus S,\, S}} \leq b$. Note that $\mathsf{ren}_{(S_2 \setminus S) \cup S_3,\, S}$ is a renaming of the variables in $(S_2 \setminus S) \cup S_3$, by labeling them with S. Since a different renaming is needed for every considered subset S of S_1, we just choose that set S for the renaming, as it improves readability.

In [3] we also provide: the correctness proofs of the F_{SM}^{∇} encoding for finding supermodels of a partial weighted MaxSAT formula F; the generalizations of our definitions and encodings so that partially robust solutions can be obtained for problem instances lacking totally robust solutions, and an extensive benchmarking section considering resource allocation problem instances.

3 Conclusion

In this paper we have proposed a mechanism for finding robust solutions to weighted MaxSAT problems. We have extended the approach of Ginsberg et al. [4] to deal with cost constraints and don't-care variables. By using cardinality constraints, the reformulation results in a much smaller problem in the pseudo-Boolean framework. Moreover, with our approach, the solution to the extended instance provides not only the supermodel for the initial problem, but also a possible repair for each of the potential breakages.

(a, b, β)-super solutions do not exactly match (a, b)-super solutions of [7] because the cost β is not explicitly considered there. Notice that imposing a constraint on the cost of (a, b)-super solutions would only guarantee it for the solution found, but not on the possible repairs. Therefore, if we wanted such restriction to hold also for the repairs, we should modify the algorithm to find (a, b)-super solutions, whilst our approach guarantees a cost of β both for the initial solution and for every possible repair. Using state-of-the-art pseudo-Boolean solvers we have been able to find (a, b, β)-super solutions for combinations of a ranging from 1 to 2 breakages, b ranging from 1 to 8 repairs, and β ranging from a 60% to a 90% of optimality, for several resource allocation problems, most of the times in far less than 1000 seconds. This is quite successful, especially compared to previous works on robustness, which were restricted to (1,0)- and (1,1)-supermodels [4] for reformulation approaches, and to at most (1,3)-super solutions to CSP problems with search-based approaches [5].

Our approach can be seen as a generic framework for robustness through reformulation, since slight changes in the encoding allow to model other notions of robustness. For example, a variant could be to directly designate the potential breaks to handle: instead of using S_1^a, we could decide what (combinations of) breaks deserve being repaired, which would be always a subset of 2^{S_1}. This would be useful if we only want to consider those breaks having a non negligible probability of occurring, particularly in very large problems. We could also think of a robustness notion where each breakable variable has a corresponding set of associated repairable variables.

References

1. Josep Argelich, Alba Cabiscol, Inês Lynce, and Felip Manyà. Modelling Max-CSP as Partial Max-SAT. In *11th International Conference on Theory and Applications of Satisfiability Testing, SAT 2008*, volume 4996 of *LNCS*, pages 1–14. Springer, 2008.
2. D. Bertsimas and M. Sim. The Price of Robustness. *Operations Research*, 52(1):35–53, 2004.
3. Miquel Bofill, Dídac Busquets, Víctor Muñoz, and Mateu Villaret. Reformulation based MaxSAT robustness. *Constraints*, 18(2):202–235, 2013.
4. Matthew L. Ginsberg, Andrew J. Parkes, and Amitabha Roy. Supermodels and robustness. In *15th National Conference on Artificial Intelligence and 10th Innovative Applications of Artificial Intelligence Conference, AAAI/IAAI 1998*, pages 334–339. AAAI Press / The MIT Press, 1998.
5. E. Hebrard. *Robust Solutions for Constraint Satisfaction and Optimisation under Uncertainty*. PhD thesis, University of New South Wales, 2006.
6. Emmanuel Hebrard, Brahim Hnich, Barry O'Sullivan, and Toby Walsh. Finding Diverse and Similar Solutions in Constraint Programming. In *20th National Conference on Artificial Intelligence and 17th Innovative Applications of Artificial Intelligence Conference, AAAI/IAAI 2005*, pages 372–377. AAAI Press / The MIT Press, 2005.
7. Emmanuel Hebrard, Brahim Hnich, and Toby Walsh. Robust Solutions for Constraint Satisfaction and Optimization. In *16th Eureopean Conference on Artificial Intelligence, ECAI 2004*, pages 186–190. IOS Press, 2004.
8. Emmanuel Hebrard, Brahim Hnich, and Toby Walsh. Super Solutions in Constraint Programming. In *8th International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, CPAIOR 2004*, volume 3011 of *LNCS*, pages 157–172. Springer, 2004.
9. Emmanuel Hebrard, Brahim Hnich, and Toby Walsh. Improved Algorithm for Finding (a,b)-Super Solutions. In *Workshop on Constraint Programming for Planning and Scheduling*, pages 236–248, 2005.
10. Alan Holland and Barry O'Sullivan. Weighted Super Solutions for Constraint Programs. In *20th National Conference on Artificial Intelligence and 17th Innovative Applications of Artificial Intelligence Conference, AAAI/IAAI 2005*, pages 378–383. AAAI Press / The MIT Press, 2005.
11. Chu Min Li and Felip Manyà. *Handbook of Satisfiability*, chapter MaxSAT, Hard and Soft Constraints, pages 613–631. IOS Press, 2009.
12. Olivier Roussel and Vasco Manquinho. *Handbook of Satisfiability*, chapter Pseudo-Boolean and Cardinality Constraints, pages 695–734. IOS Press, 2009.
13. Amitabha Roy. Fault Tolerant Boolean Satisfiability. *Journal of Artificial Intelligence Research*, 25:503–527, 2006.