Auction Robustness through Satisfiability Modulo Theories*

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

[1] Departament d'Informàtica i Matemàtica Aplicada
Universitat de Girona
{miquel.bofill,mateu.villaret}@udg.edu
[2] Institut d'Informàtica i Aplicacions
Universitat de Girona
didac.busquets@udg.edu

Abstract. Solution robustness is a desirable feature when dealing with uncertainty. This issue has rarely been taken into account in the field of auctions, where the goal is to obtain optimal solutions (i.e., maximize the auctioneer's benefit). In this paper we define a notion of robustness for auctions where some resources may become unavailable once the auction has already been cleared. This notion of robustness balances the number of changes needed for repairing the solution and the possible loss of benefit for the auctioneer. In order to obtain robust solutions for auctions, we define a mechanism based on transporting the concept of supermodel to the setting of weighted Max-SAT. We show that finding a supermodel of a weighted Max-SAT formula amounts to find a model of an SMT (Satisfiability Modulo Theories) formula.

1 Introduction

Auctions are widely used as a resource allocation mechanism, since they enable an efficient distribution of resources amongst the agents requesting them [1]. Most auction mechanisms focus on maximizing the auctioneer's benefit, and assume that once the auction is cleared (i.e., a solution is found), the state of the world cannot change. However, in real applications, this is not the case. In the time between the clearance of the auction and the moment of the allocation of resources taking place, many things can happen. For instance, a machine that has been assigned to a given agent may break down before it starts using it, a winning agent may decide to withdraw its bid because it found a better deal elsewhere, etc. The consequence of such unexpected events is that the solution may then not yield the optimal benefits. Even worse, the solution may not be applicable at all. In such cases, a new solution must be found. To do so, the auction could be repeated, accounting for the new reality. However, the new solution could be completely different from the initial one, meaning that some bids that were

---

winners in the first solution are losers in the new one. Such behavior could be a nuisance for the participating bidders, especially for those that were told their bids were winners but become losers in the repetition of the auction. To avoid such situations, the auctioneer could try to find a solution that is prepared for unexpected events. Ideally, such a solution should be applicable no matter what these events are, although this is not always achievable. Thus, when this happens, the solution should be reparable with the minimum number of changes to the original solution. And, while the repaired solution will probably be sub-optimal, we are interested in still obtaining a high profit for the auctioneer.

The issue of bid withdrawal has already been addressed in [2]. But, as far as we know, the problem of resource unavailability has never been taken into account. Thus, in this paper we introduce a new mechanism for finding robust solutions to auctions with a bounded number of allowed breakages in resource availability, a bounded number of repairs in bid assignment, and a minimum guaranteed profit for the auctioneer:

> If a solution is found, we can guarantee that, for any breakage in resource availability involving at most a resources, the solution can be repaired with at most b changes in bid assignment. In addition, the initial solution and all repaired solutions have a profit of at least β. We call such a solution an *(a,b,β)-super solution*.

Our starting point is the standard modeling of an auction as a weighted Max-SAT problem [3, 4] with boolean variables to denote whether a bid is winner or loser. In addition, we also use boolean variables representing resource availability. The transformation used to obtain (a, b, β)-super solutions works in the spirit of the one of [5] to obtain (a, b)-supermodels for propositional logic. As in [5], we define a set of breakable variables and a set of repairable variables. Here, the breakage set corresponds to the variables denoting resource availability, and the repair set to those corresponding to bid assignment.

In fact, our mechanism can be applied to any weighted Max-SAT problem, auctions being a particular case. An interesting point of our result is that weighted Max-SAT turns out to be not expressive enough to deal with the profit constraints that we must add in the robust version of the problem. For this reason, we move to the setting of Satisfiability Modulo Theories (SMT) [6] and model the auction as a weighted Max-SMT formula. Hence, for our purposes the transformation of [5] must be adapted to this new setting. The SMT problem for a theory T is: given a formula F, determine whether there is a model of $T \cup \{F\}$. Hence, an SMT instance is a generalization of a boolean SAT instance in which some propositional variables have been replaced by predicates from the underlying theories. For example, if T denotes the theory of linear (integer or real) arithmetic, then a formula can contain clauses like, e.g., $p \lor q \lor x+2 \leq y \lor x = y+z$, providing a much richer modeling language than is possible with SAT formulas.

The rest of the paper is organized as follows. In Section 2 we review some related work concerning robustness and auctions. In Section 3 we formally define the kinds of auctions we consider. In Section 4 we describe an example to show

how we look for robust solutions. Section 5 is devoted to the mechanism for finding robust solutions. Finally, in Section 6 we conclude.

2 Related work

Here we briefly review some related work on robust solutions in general, and on robustness and the use of logics in the field of auctions.

2.1 Supermodels

The seminal work on robust solutions for propositional logic formulas is the one of [5], where the notion of *supermodel* is introduced. The complexity for finding such supermodels in several propositional logic fragments has been studied in [7].

Definition 1 (from [5]). *An (S_1^a, S_2^b)-supermodel of a boolean formula F is a 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).*

An (S_1^a, S_2^b)-supermodel in which the breakage and the repair set are unrestricted is denoted as an (a, b)-supermodel.

The task of finding (a, b)-supermodels is NP-complete. The main idea is to encode the supermodel requirements of a formula F as a new formula F_{SM} whose size is polynomially bounded by the size of F. This new formula F_{SM} has a model if and only if F has an (a, b)-supermodel.

Example 1. The formula $F = p \vee q$ has three models, $\{p, q\}$, $\{\neg p, q\}$ and $\{p, \neg q\}$, which are all $(1, 1)$-supermodels. The encoding F_{SM} for a $(1, 1)$-supermodel of F, according to [5], would be

$$F_{SM} = \overbrace{(p \vee q)}^{\text{original } F} \wedge \left(\overbrace{\overbrace{(\neg p \vee q)}^{\text{no repair}} \vee \overbrace{(\neg p \vee \neg q)}^{\text{repair } q}}^{\text{break in } p} \right) \wedge \left(\overbrace{\overbrace{(p \vee \neg q)}^{\text{no repair}} \vee \overbrace{(\neg p \vee \neg q)}^{\text{repair } p}}^{\text{break in } q} \right)$$

Note that, for instance, if the satisfying interpretation (model) chosen for F_{SM} is $\{\neg p, q\}$, i.e., $\{p = false, q = true\}$, then $p \vee q$ is satisfied and, moreover, if q switches to *false*, then a new model for $p \vee q$ can be obtained by switching p to *true*. That is, a break in q has a repair on p. The key idea is that the value of the subformula $\neg p \vee \neg q$ under the initial interpretation coincides with the value of $p \vee q$ under the repaired interpretation and, hence, F_{SM} has a model if and only if F has a supermodel. Note also that only the first model $\{p, q\}$ is a $(1, 0)$-supermodel.

The concept of (a, b)-supermodel for propositional logic has been generalized to that of (a, b)-*super solution* in the context of constraint programming in [8].

2.2 Robustness in auctions

There are several works that deal with robustness with respect to potential manipulations of the auction mechanism (such as false-name bids and other types of manipulations). However, this is not the concept of robustness we are interested in. As we have described in Section 1, we focus our research on robustness of the solution to the auction. Some works, such as [9, 10] add the concept of robustness (*fault tolerance*) to *mechanism desing* in order to deal with potential failures in the execution of tasks by the agents, although they use a probabilistic approach and do not consider repairing the solutions. As far as we know, the only work that deals with solution repair in auctions is that of [2]. This work addresses the problem of bid withdrawal (i.e. a bidder that withdraws a winning bid), and, in order to find robust solutions, uses the (α, β)-*weighted super solutions* [11], an extension of super solutions [8] that takes into account the breakage probability (α) and the cost of repair (β). Our work is similar, since our approach is also based on supermodels and we look for solutions with a bounded cost. However, we consider the problem of resource unavailability, which is not considered in [2]. Moreover, we are also interested in keeping the number of repairs low, which is only done indirectly (through the cost function) in [2]. In addition, our techniques are completely different because we use the logic framework of Satisfiability Modulo Theories, while [2] presents an ad-hoc search algorithm to find robust solutions.

Another closely related problem is that of robust knapsack [12] (i.e. a knapsack problem where the weights and/or values of the objects are imprecise). Given that many auction mechanisms can be modeled as a knapsack problem [13], it is reasonable to think that some of the robust approaches to this problem may yield robust solutions to auctions. However, the robustness concept used in the field of knapsack is somehow different to ours, since it does not consider the possibility of repairing a solution. Instead, a robust solution of a knapsack problem with imprecision is such that, on average, performs well regardless of what the actual weights or values of the objects are.

2.3 Auctions and Logics

Regarding the use of logics in the field of auctions, it has been mainly used to define different bidding languages [14]. There are also some works that use logics as the method for solving the winner determination problem. For instance, Baral et al [15] model the auction using SModels, and use their approach to solve combinatorial auctions and combinatorial exchanges. However, they do not deal with robustness issues. On the other hand, modeling an auction as a weighted Max-SAT formula is a standard problem in Max-SAT benchmarks [4].

3 Auction formalization

There are many types of auctions, depending on several factors, such as the number of items being offered, the number of units for each item, or the way

bidders may express their requests, among others [16]. In this work, we restrict our attention to Combinatorial Auctions [17] and, more precisely, to the winner determination problem (clearing algorithm) of the auction. Formally, a combinatorial auction is defined as follows. There is:

- a set of K agents,
- a set of N items or goods, and
- a set of M bids of the form (S_i, p_i), where $S_i \subseteq \{1..N\}$ is the subset of goods (i.e. bundle) requested in bid i, and $p_i \in I\!R^+$ is the value (price) of the bid. We denote by R_j the set of indexes of the bids sent by agent j, $j \in \{1..K\}$.

The goal is to select the winning bids, so that no good is allocated more than once, and the profit for the auctioneer is maximized:

$$\max \sum_{i=1}^{M} b_i \cdot p_i \quad \text{s.t.} \ \textstyle\sum_{i|j \in S_i} b_i \leq 1, \forall j \in \{1..N\}$$

where b_i is a boolean variable indicating whether bid i is winner or loser.

Additionally, one may introduce other constraints, such as forcing all items to be allocated (which would imply replacing the $\leq$ of the previous equation by an equality), guaranteeing that all agents win at least (or at most) a given number of bids ($\sum_{i \in R_j} b_i \geq Q$, $\forall j \in \{1..K\}$, where Q is the minimum number of winning bids per agent – or maximum if using $\leq Q$), etc.

Regarding the bidding language, in the rest of the paper we assume that bidders use an OR-language [14], that is, each bidder submits a list of bids (pairs of bundle and price), and it is interested in winning any number of the bids sent to the auctioneer. However, the approach we present may work with any other bidding language (XOR, OR-of-XOR, ...), as long as the needed restrictions (as mentioned in the previous paragraph) are added.

4   Running example

In order to illustrate our approach, we present an example that we will use in the forthcoming sections to explain each of the steps needed to find robust solutions to auctions. For the sake of simplicity, we use single-item bundles, that is, each bid requests only one item. Moreover, we impose the constraint that each agent must win at least one of the bids it sent.

Example 2. Assume we have 3 agents and 4 goods (or resources). In the following table we indicate the price of each single-item bid of each agent:

	1	2	3	4
1	10	15	-	-
2	-	5	10	20
3	15	-	-	10

Thus we have the following list of bids:

$$[\underbrace{(1,10),(2,15)}_{\text{first agent's bids}},\underbrace{(2,5),(3,10),(4,20)}_{\text{second agent's bids}},\underbrace{(1,15),(4,10)}_{\text{third agent's bids}}]$$

We define the boolean variables g_1, g_2, g_3 and g_4 to represent the availability of the corresponding goods, and the boolean variables $b_i, i \in \{1..7\}$ to indicate whether bid i is winner or loser. Then, assuming that the only possible source of breakages is resource availability, we define the breakage set as being $S_1 = \{g_1, g_2, g_3, g_4\}$. Then, the repair set is $S_2 = \{b_1, b_2, b_3, b_4, b_5, b_6, b_7\}$, since a break in a resource may imply that a winning bid becomes loser and, eventually, other assignments can be reconsidered in order to improve the auctioneer's profit under the new circumstances.

Assume that we look for *robust solutions* where one break may occur and each possible break must be repairable with at most four changes. Assume, moreover, that we want that whatever the break is, the profit of the initial solution and of the repaired solution is at least 30. This will correspond to a $(1, 4, 30)$-super solution (see Definition 2 of Subsection 5.2).

The optimal solution to this auction without considering robustness would be to set as winning bids the second, the fourth, the fifth and the sixth bids, i.e.,

$$b_1 = 0, b_2 = 1, b_3 = 0, b_4 = 1, b_5 = 1, b_6 = 1, b_7 = 0,$$

which means assigning good 2 to the first agent, goods 3 and 4 to the second agent and good 1 to the third agent. For the sake of readability we use the notation such as 2456 to indicate which are the winning bids of a solution. Then, with this solution, the auctioneer would have a profit of 60.

However, this optimal solution is not a $(1, 4, 30)$-super solution: if good 2 became unavailable (break), the only alternative for the first agent would be good 1, but this is already allocated to the third agent; this would imply finding also an alternative for the third agent, which would be good 4, but this good is allocated to the second agent. Thus, repairing the breakage of good 2 would imply modifying two winning bids (b_2 to b_1 and b_6 to b_7) and unassigning one winning bid (b_5), meaning five repairs (as shown in boldface):

$$\mathbf{b_1 = 1, b_2 = 0}, b_3 = 0, b_4 = 1, \mathbf{b_5 = 0, b_6 = 0, b_7 = 1},$$

which is more than the four allowed repairs. Note that for each bidder, choosing a new winning bid may imply two repairs (one to set the initially winning bid to 0, and in case he had no other winning bids, another one to set one of its losing bids to 1).

This auction has 9 feasible solutions (i.e., solutions satisfying the constraints on bid incompatibility and minimum number of winning bids per agent), which are the following: 1247, 137, 1347, 147, 246, 2456, 247, 2467 and 256. Within these solutions, only three of them are $(1, 4, 30)$-super solutions: 246, 2467 and 247. Next we go through the details of solution 2467, i.e.,

$$b_1 = 0, b_2 = 1, b_3 = 0, b_4 = 1, b_5 = 0, b_6 = 1, b_7 = 1$$

which has a profit of 50 units for the auctioneer. For this solution, the four possible breakages can be repaired as follows:

1. $g_1 = 0$. The repair is $b_6 = 0$, being the new solution 247, and the profit 35.
2. $g_2 = 0$. The repair is $b_1 = 1, b_2 = 0, b_6 = 0$, being the new solution 147 and the profit 30.
3. $g_3 = 0$. The repair is $b_4 = 0, b_5 = 1, b_7 = 0$, being the new solution 256 and the profit 50.
4. $g_4 = 0$. The repair is $b_7 = 0$, being the new solution 246 and the profit 40.

It can be seen that all repairs have a profit of at least 30 and the number of repairs is not greater than 4. Moreover, in the third case there is not even loss in the profit. We let the reader check that solutions 246 and 247 are also $(1, 4, 30)$-super solutions, with a profit of 40 and 35, respectively. However, since the profit of solution 2467 is higher, this would be the optimal $(1, 4, 30)$-super solution to this auction.

As for the rest of feasible solutions, some of them (147 and 1247) are $(1, 4, \_)$-super solutions, meaning that they can be repaired with at most 4 changes, but not $(1, 4, 30)$-super solutions, since they do not satisfy that the solution and its repairs have a profit of at least 30. In particular, solution 147 has a profit of 30, but one of its repairs has a profit of only 25 (when good 3 becomes unavailable, the second agent must be assigned good 2, which corresponds to a low value bid). Similarly, solution 1247 has a profit of 45, but it also fails in the profit of repairs, since one of them has again a profit of 25.

Finally, some solutions (137, 1347, 2456 and 256) are not even $(1, 4, \_)$-super solutions, since they do need more than 4 changes in order to repair some of the breakages. This is the case of the optimal solution without robustness (2456), as we have seen a few paragraphs above.

5 Mapping auctions to supermodels

5.1 Auctions as weighted Max-SAT problems

An auction A like defined in Section 3 can be modeled as a weighted Max-SAT formula F^A as follows. Let $g_1, \ldots, g_N$ be a set of boolean variables representing the availability of each of the N goods, and let $b_1, \ldots, b_M$ be a set of boolean variables representing whether each of the M bids is winner or loser.

1. *Resource availability.* For each bid j of the form (S_j, p_j), where $S_j = \{i_1, \ldots, i_{n_j}\} \subseteq \{1..N\}$, we state

$$b_j \rightarrow g_{i_1} \wedge \cdots \wedge g_{i_{n_j}}$$

to indicate that, whenever bid j is accepted, then all goods it requests must be available[3].

---

[3] These constraints are only needed to deal with resource unavailability, i.e., they do not appear in standard (non-robust) auctions.

2. *Bid incompatibility.* For each pair of bids i and j (with $i \neq j$) such that $S_i \cap S_j \neq \emptyset$, we state

$$\neg b_i \vee \neg b_j$$

to indicate incompatibility between bids requesting the same good.

3. *Minimum winning bids.* For each set $R_k = \{i_1, \dots, i_{n_k}\}$ corresponding to the bids of agent k, we state

$$b_{i_1} \vee \cdots \vee b_{i_{n_k}}$$

to indicate that at least one bid of each agent must be accepted. Note that this restriction can change, or even disappear, depending on the type of bidding language used in the auction (OR, XOR, OR-of-XOR, ...) and also on the constraints about the number of winning bids per agent.

4. *Bid's value.* For each bid i, we add a unit clause

$$(b_i, p_i)$$

indicating that if bid i is not accepted, then there is a loss of profit of p_i.

The conjunction of the previous constraints defines a weighted Max-SAT problem for the auction.

5.2 Robust auctions as weighted Max-SMT problems

Here we show how a weighted Max-SAT formula F^A defining an auction A can be transformed into a weighted Max-SMT formula defining a robust version of the former. In particular, we describe how to obtain a Max-SMT formula F_{SM}^A such that F_{SM}^A has a model if and only if A has an (a, b, β)-super solution. Some of the proofs are omitted due to lack of space.

Definition 2. *An (a, b, β)-super solution of an auction is a (maximal profit) solution for the auction such that, if a goods become unavailable (breakage), then another solution can be obtained by changing at most b bids from winner to loser or vice-versa (repair) and, moreover, the solution and all possible repaired solutions have a profit of at least β.*

The following definition generalizes the one of [5] to weighted Max-SAT.

Definition 3. *An (S_1^a, S_2^b, β)-supermodel of a weighted Max-SAT formula F is a 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), 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, moreover, the solution and all possible repaired solutions have a cost of at most β.*

Lemma 1. *An auction A with N goods and M bids has an (a, b, β)-super solution if and only if the weighted Max-SAT formula F^A has an (S_1^a, S_2^b, β')-supermodel where $S_1 = \{g_1, \dots, g_N\}$, $S_2 = \{b_1, \dots, b_M\}$ and cost (i.e. loss of profit) $\beta' = \left(\sum_{i=1}^{M} p_i\right) - \beta$.*

Now we show how to construct a weighted Max-SMT formula F_{SM} from a weighted Max-SAT formula F, such that F has an (S_1^a, S_2^b, β)-supermodel if and only if F_{SM} has a model.

The construction of F_{SM}. Let

$$F = C \wedge W$$

be a weighted Max-SAT formula, where C denotes the set of mandatory constraints and W denotes the set of weighted, non-mandatory constraints[4]. For the sake of simplicity, we will assume that W consists only of unary clauses of the form (b, w), where b is a boolean variable and w is a weight. Note that any Max-SAT formula can be transformed into an equisatisfiable one fulfilling this requirement by reification, i.e., by replacing any weighted constraint (F, w) such that F is not unary by $(F \leftrightarrow b) \wedge (b, w)$.

Now, assuming that

$$W = (b_1, w_1) \wedge \cdots \wedge (b_k, w_k)$$

we introduce a set of integer variables $i_1, \ldots, i_k$ and define

$$L = \bigwedge_{j \in 1..k} (b_j \rightarrow i_j = 1) \wedge (\neg b_j \rightarrow i_j = 0)$$

Let S^n denote the set of all (possibly empty) subsets of a set S whose size is at most n, and let S^{n+} denote the set of all non-empty subsets of a set S whose size is at most n. Moreover, let $F_{\overline{S}}$ denote a boolean formula F where all occurrences of variables in the set S have been flipped (i.e., negated).

We define

$$B_{\overline{S}} = \sum_{j \in 1..k} \begin{cases} i_j \cdot w_j & \text{if } b_j \in S \\ (1 - i_j) \cdot w_j & \text{if } b_j \notin S \end{cases}$$

where S is a set of boolean variables. We denote by B the particular case $B_{\overline{\emptyset}} = \sum_{j \in 1..k} (1 - i_j) \cdot w_j$, corresponding to the cost of the unsatisfied clauses in W.

Finally, we define

$$F_{SM} = C \wedge W \wedge L \wedge (B \leq \beta) \wedge \bigwedge_{S \in S_1^{a+}} \left(\bigvee_{T \in (S_2 \setminus S)^b} (C_{\overline{S \cup T}} \wedge (B_{\overline{S \cup T}} \leq \beta)) \right)$$

Note that, due to L and the constraints of the form $B \leq \beta$, this formula is not plain SAT: it falls into SAT modulo the quantifier-free fragment of the (first-order) linear arithmetic theory. The main lemma and theorem follow here.

Lemma 2. *A weighted Max-SAT formula F has an (S_1^a, S_2^b, β)-supermodel if and only if the weighted Max-SMT formula F_{SM} has a model.*

---

[4] We talk about constraints instead of clauses since our transformation does not require the formula to be in CNF format.

This lemma follows the spirit of the result of [5], but adding costs. Here, the key idea is that $B_{\overline{S \cup T}}$, gives us the cost of the unsatisfied clauses in W if the variables in $S \cup T$ were to change their value with respect to the initial solution. Note that the variables in S represent the breakage variables and the ones in T represent the repair variables and, hence, $S \cup T$ denotes the set of variables that are going to change their value. Notice also that the sets S and T are disjoint, since it makes no sense to repair a broken variable in this setting.

Theorem 1. *An auction A has an (a, b, β)-super solution if and only if the weighted Max-SMT formula F_{SM}^A has a model.*

Proof. Let $F^A = C \wedge W$ denote the weighted Max-SAT formula obtained from A as explained in Subsection 5.1, where C denotes the set of mandatory constraints introduced in points 1, 2, and 3, and W denotes the set of non-mandatory unit clauses of point 4. Let $S_1 = \{g_1, \ldots, g_N\}$, $S_2 = \{b_1, \ldots, b_M\}$ and $\beta' = \left(\sum_{i=1}^M p_i\right) - \beta$. By Lemma 2, F^A has an (S_1^a, S_2^b, β')-supermodel if and only if F_{SM}^A has a model. Therefore, by Lemma 1, A has an (a, b, β)-super solution if and only if F_{SM}^A has a model. □

Observe that, since a and b are constants, the increase of size of F_{SM}^A w.r.t. the size of F^A is polynomially bounded, namely, it is $\mathcal{O}(n^{a+b})$ larger than F^A, where n is the number of variables of F^A.

5.3 The whole story

Here we take the auction A of Example 2 and briefly describe how to model it as a robust auction. We begin by modeling the auction as a Max-SAT problem as explained in Subsection 5.1, which gives us $F^A = C \wedge W$ with

$$
\begin{aligned}
C = \ & (b_1 \rightarrow g_1) \wedge (b_2 \rightarrow g_2) \ \wedge (b_3 \rightarrow g_2) \wedge (b_4 \rightarrow g_3) \wedge (b_5 \rightarrow g_4) \ \wedge \\
& (b_6 \rightarrow g_1) \wedge (b_7 \rightarrow g_4) \ \wedge (\neg b_1 \vee \neg b_6) \wedge (\neg b_2 \vee \neg b_3) \wedge (\neg b_5 \vee \neg b_7) \wedge \\
& (b_1 \vee b_2) \wedge (b_3 \vee b_4 \vee b_5) \wedge (b_6 \vee b_7) \\
W = \ & (b_1, 10) \wedge (b_2, 15) \wedge (b_3, 5) \wedge (b_4, 10) \wedge (b_5, 20) \wedge (b_6, 15) \wedge (b_7, 10)
\end{aligned}
$$

Now observe that the sum of the costs of the non-mandatory weighted clauses is $10 + 15 + 5 + 10 + 20 + 15 + 10 = 85$. Then, as stated by Lemma 1, in order to A have a $(1, 4, 30)$-super solution, we must look for a $(S_1^1, S_2^4, 85 - 30)$-supermodel of F^A, i.e., a $(S_1^1, S_2^4, 55)$-supermodel of F^A, where $S_1 = \{g_1, g_2, g_3, g_4\}$ and $S_2 = \{b_1, b_2, b_3, b_4, b_5, b_6, b_7\}$. Finally, according to Lemma 2, this amounts to find a model of the weighted Max-SMT formula

$$
F_{SM}^A = C \wedge W \wedge L \wedge (B \leq 55) \wedge \bigwedge_{S \in S_1^{1+}} \left(\bigvee_{T \in (S_2 \setminus S)^4} (C_{\overline{S \cup T}} \wedge (B_{\overline{S \cup T}} \leq 55)) \right)
$$

as described in Subsection 5.2, where

$$
L = \bigwedge_{j \in 1..7} (b_j \rightarrow i_j = 1) \wedge (\neg b_j \rightarrow i_j = 0)
$$

$$B = (1 - i_1) \cdot 10 + (1 - i_2) \cdot 15 + (1 - i_3) \cdot 5 + (1 - i_4) \cdot 10+$$
$$(1 - i_5) \cdot 20 + (1 - i_6) \cdot 15 + (1 - i_7) \cdot 10$$

Note that S_1^{1+} denotes the non-empty subsets of S_1 with at most one element, i.e., the singletons $\{g_1\}, \{g_2\}, \{g_3\}$ and $\{g_4\}$. And, since S_1 and S_2 are disjoint, we have that $(S_2 \setminus S)^4 = S_2^4$, i.e., the (possibly empty) subsets of S_2 of size at most 4. Due to lack of space we do not develop each element $C_{\overline{S \cup T}}$ and $B_{\overline{S \cup T}}$.

6 Conclusions

In this paper we have presented a mechanism for obtaining robust solutions to auctions. This issue has rarely been considered in the field of auctions, with only one exception that we know of [2]. However, we think that robustness is a key issue when dealing with real world applications, where uncertainty is almost always present. In particular, we have focused on the possibility of some of the resources becoming unavailable once the auction has already been cleared. Thus, we provide a mechanism to proactively look for solutions that can be easily repaired when such unexpected events happen.

The notion of robustness that we have presented balances the number of allowed repairs when a break occurs and the loss of profit for the auctioneer. Thus, we have defined what we call (a, b, β)-super solutions. This allows the auctioneer to choose the more convenient values of each parameter, depending on how conservative or risk seeking its strategy is.

Our mechanism is based on the modeling of an auction as a weighted Max-SAT formula. However, since SAT does not allow to easily encode formulas with arithmetic operations, needed to achieve robustness, we have moved the problem to the richer logical framework of Satisfiability Modulo Theories.

Let us mention that state-of-the art SMT solvers have a rich input language, and it is not necessary (neither convenient) to translate any formula to CNF in order they can read it. In fact, we can feed an SMT solver directly with a formula such as F_{SM} (as described in Subsection 5.2) contrarily to what is done in [5], where all formulas are translated to CNF. It is worth noting that we have not considered the option of translating our transformed formula into a linear program since, on the one hand, the SMT solvers dealing with the theory of linear arithmetic already apply a (modified) simplex algorithm and, on the other hand, such a translation would imply a flattening of the structure of the problem (of which the SMT solvers can eventually take profit) and the addition of many new variables. Moreover, the richness of the SMT language allows us to easily add new constraints when needed.

Some experiments have been carried out with Yices [18], an SMT solver that, apart from pure SMT problems, is able to solve weighted Max-SMT problems. At present only toy examples have been checked. In the future we plan to test our transformation with realistic benchmarks from the Combinatorial Auctions Test Suite (CATS) [19].

Finally, we want to acknowledge the fruitful discussions and comments we've had with the rest of the SuRoS project team.

References

1. McMillan, J.: Selling spectrum rights. Journal of Economic Perspectives **8** (1994) 145–162
2. Holland, A., O'Sullivan, B.: Robust solutions for combinatorial auctions. In: ACM Conf. on Electronic Commerce. (2005)
3. Larrosa, J., Heras, F., de Givry, S.: A logical approach to efficient max-sat solving. Artif. Intell. **172** (2008) 204–233
4. Argelich, J., Li, C.M., Manyà, F., Planes, J.: The first and second Max-SAT evaluations. J. on Satisfiability, Boolean Modeling and Computation **4** (2008) 251–278
5. Ginsberg, M.L., Parkes, A.J., Roy, A.: Supermodels and robustness. In: Proc. of AAAI'98. (1998) 334–339
6. Sebastiani, R.: Lazy satisability modulo theories. J. on Satisfiability, Boolean Modeling and Computation **3** (2007) 141–224
7. Roy, A.: Fault tolerant boolean satisfiability. J. Artificial Intelligence Research **25** (2006) 503–527
8. Hebrard, E., Hnich, B., Walsh, T.: Super solutions in constraint programming. In: Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. Lecture Notes in Computer Science. Springer (2004) 157–172
9. Ramchurn, S.D., Mezzetti, C., Giovannucci, A., Rodriguez, J.A., Dash, R.K., Jennings, N.R.: Trust-based mechanisms for robust and efficient task allocation in the presence of execution uncertainty. Journal of Artificial Intelligence Research **35** (2009) 119–159
10. Porter, R., Ronen, A., Shoham, Y., Tennenholtz, M.: Fault tolerant mechanism design. Artificial Intelligence **172** (2008) 1783 – 1799
11. Holland, A., O'Sullivan, B.: Weighted super solutions for constraint programs. In: Proc. of AAAI'05. (2005) 378–383
12. Yu, G.: On the max-min 0-1 knapsack problem with robust optimization applications. Operations Research **44** (1996) 407–415
13. Kelly, T.: Generalized Knapsack Solvers for Multi-unit Combinatorial Auctions: Analysis and Application to Computational Resource Allocation. LNAI **3435** (2005) 73–86
14. Nisan, N.: Bidding and allocation in combinatorial auctions. In: Proc. of ACM Conference on Electronic Commerce. (2000) 1–12
15. Baral, C., Uyan, C.: Declarative specification and solution of combinatorial auctions using logic programming. In: Proc. of LPNMR'01, LNCS. Volume 2173. (2001) 186–199
16. Wurman, P.R., Wellman, M.P., Walsh, W.E.: A parametrization of the auction design space. Games and Economic Behavior **35** (2001) 304 – 338
17. Cramton, P., Shoham, Y., Steinberg, R., eds.: Combinatorial Auctions. MIT Press (2006)
18. Dutertre, B., de Moura, L.: The Yices SMT solver. Tool paper available at http://yices.csl.sri.com/tool-paper.pdf (2006)
19. Leyton-Brown, K., Pearson, M., Shoham, Y.: Towards a universal test suite for combinatorial auction algorithms. In: Proc. of ACM Conference on Electronic Commerce. (2000) 66–76