

Super Solutions for Combinatorial Auctions ^{*}

Alan Holland and Barry O’Sullivan

Cork Constraint Computation Centre
Department of Computer Science, University College Cork, Ireland
{a.holland,b.osullivan}@cs.ucc.ie

Abstract. Super solutions provide a framework for finding robust solutions to Constraint Satisfaction Problems [5, 3]. We present a novel application of super solutions to combinatorial auctions in which a bid may be disqualified or withdrawn after the winners are announced. We examine the effectiveness of super solutions in different auction scenarios that simulate economically motivated bidding patterns. We also analyze the drawbacks of this approach and motivate an extension to the framework that permits a more flexible and realistic approach for determining robust solutions.

1 Introduction

Many auctions involve the sale of various distinct items in which bidders perceive complementarity or substitutability between them. When auctioning such items, the auctioneer typically packages them in such a way as to maximize the complementarities amongst them and then puts these packages up for auction. When selling a farm, for example, it may be sold as a single item or divided into separate packages such as the farmhouse, outhouses, sites overlooking the beach, arable and non-arable land. When deciding on how to package the sale, it is impossible to know for sure what packages would maximize revenue. Bidders may view the complementarities between items differently and as the size of the auction grows it quickly becomes impossible for the auctioneer to know how the items should be packaged.

It is more economically efficient for the bidders to bid on combinations of items. Unfortunately the number of possible combinations of items of interest to each bidder grows exponentially as the number of items increases. The bids also need to be communicated to the auctioneer in a concise manner, which becomes increasingly difficult in large auctions.

The objective is typically to maximize revenue when selling items in a forward auction and to minimize cost when procuring items in a reverse auction. We consider the scenario where optimum revenue is non-essential but a *good robust* solution is essential. A *good* solution is one whose revenue is within a percentage of the optimum. A *good robust* solution is a good solution for which we know that even if winning bids are withdrawn another good solution can be found. Robust solutions are desirable in

^{*} This work has received funding from Science Foundation Ireland (Grant Number 00/Pl.1/C075) and from Enterprise Ireland, Research Innovation Fund (Grant Number RIF-2001-317).

scenarios where suppliers or customers are unreliable and a good solution needs to be found with minimal upset to other bidders.

This paper is organized as follows. Section 2 introduces combinatorial auctions and describes the Winner Determination Problem. Section 3 presents super solutions and briefly discusses how they can be applied to combinatorial auctions. Section 4 outlines the experimental results achieved when finding super solutions with respect to various objectives one may wish to consider. It also tackles the optimization problems of finding a super solution with maximal robustness and revenue. Section 5 describes some of the limitations of super solutions and proposes an extension to the framework to overcome these difficulties. Some concluding remarks are made in Section 6.

2 Combinatorial Auctions

Combinatorial auctions fall into two categories, forward and reverse. In a reverse auction the auctioneer is seeking to procure goods and therefore minimize cost. If it is possible to purchase more items than are strictly necessary, the problem is known as a Set Covering Problem. The buyer may choose to stipulate in the auction rules that no surplus items are to be purchased. With the introduction of this constraint this problem becomes a Set Partition Problem.

In the remainder of this paper we focus our attention on forward auctions, where items are being sold and the objective is to maximize revenue. However, we will present our motivation for our interest in combinatorial auctions briefly below before considering forward auctions in more detail.

2.1 Motivation

The popularity of online auctions has increased in recent years because the internet promises to promote competition, thereby increasing revenue for the auctioneer. In an auction, where items exhibit complementarities or substitutabilities there exists a phenomenon known as the *exposure problem* [8]. This occurs when bidders seek a certain set of items but do not want to end up with a subset of the items that they may find valueless. This encourages cautionary bidding tactics that result in depressed bidding. Combinatorial auctions may alleviate the exposure problem by permitting bids on an arbitrary combination of items that suits the bidders needs. In this manner such auctions improve efficiency where items exhibit complementarities/substitutabilities for the bidders.

Such auctions have been used in many real-world scenarios such as procurement for the Mars Corporation [6] and the sale of spectrum licences in America's Federal Communications Commission (FCC) auctions. The London Transport Authority also operated a combinatorial auction in their procurement of bus services from private operators [9, 8]. The Chilean government have also adopted combinatorial auctions for the supply of school meals to children. In the latter case, the quality of suppliers was considered as well as the bid amount in deciding the winner and the system also ensured there was no monopoly in any individual region. The reported supply costs have fallen by 22% since the adoption of the program [2].

In auctions where complementarities or substitutabilities are exhibited between items, there is a compelling argument for the introduction of combinatorial bidding to improve overall efficiency. Their application is spreading to other areas such as Supply Chain Management [12] that also demand robust solutions.

2.2 Forward Auctions: The Set Packing Problem

The auctioneer must determine the winner from all bids received. This is known as the *Winner Determination Problem* (WDP) and can be represented as a Set Packing Problem (SPP). In this case it is not necessary to sell all items in order to maximize revenue.

The SPP can be formulated as follows. Let I be the set of items to be auctioned and V is a collection of subsets of I . Let $x_j = 1$ if the j^{th} set in V is a winning bid and c_j signifies the amount of that bid. Also, let $a_{ij} = 1$ if the j^{th} set in V contains $i \in I$. The problem can then be stated as follows:

$$\begin{aligned} \max \quad & \sum_{j \in V} c_j x_j \\ \text{s.t.} \quad & \sum_{j \in V} a_{ij} x_j \leq 1 \quad \forall i \in I \end{aligned}$$

We have encoded the SPP as a CSP where variables represent bids and the domains have only two values, 0 and 1, representing failure and success respectively. We have assumed the notion of *free disposal*¹. Constraints between variables preclude the success of two bids containing the same item. The objective function is to maximize revenue and bid values are determined using a lookup table whose indices are variable-value pairs. An alternative formulation may include variables representing items whose domain consists of values representing the bids that include that item. Although when free disposal is assumed, not all items need to be assigned to bids so a null value is also included in the domain to indicate the item was unsold.

Example 1. Consider a simple example where an auctioneer is selling two items and there are four interested parties, bidders x_1, x_2, x_3 and x_4 (Table 1). Bidders x_1 and x_2 are interested in the first and second items respectively, bidding \$0.60million for the relevant item. Bidders x_3 and x_4 seek both items only and bid \$1.15million and \$1.10million for the pair but \$0 for each item individually². The revenue maximizing solution for the auctioneer is to sell the items separately to x_1 and x_2 thus securing \$1.2million. It is impossible for the auctioneer to know in advance whether combining the items in a single sale would be profitable or not. Combining both items in this case would have resulted in \$0.05million of lost revenue, since the winning bid would have been for \$1.15million rather than \$1.2million. Instead combinatorial auctions allow the bidders decide on parcels of items that suit their needs thereby improving overall efficiency.

¹ If the notion of *free disposal* is assumed, not all items need to be covered. In other words, there is no penalty for not selling some items.

² Such arbitrary complementarities amongst different bidders are often seen in property sales.

Table 1. Bids in Example 1

| Bidders | Items | | |
|---------|-------------|-------------|------|
| | A | B | AB |
| x_1 | 0.60 | 0.00 | 0.00 |
| x_2 | 0.00 | 0.60 | 0.00 |
| x_3 | 0.00 | 0.00 | 1.15 |
| x_4 | 0.00 | 0.00 | 1.10 |

Example 1 may be encoded as a CSP by taking the four bids as variables with domains containing 0 and 1. Bids x_1 and x_3 cannot both succeed, since they both require item A, so a constraint is added precluding the assignment in which both variables take the value 1. Similarly, bid combinations x_2 and x_3 , x_2 and x_4 , and x_3 and x_4 cannot win simultaneously. Therefore, in this example the set of CSP variables, V , would be as follows: x_1 , x_2 , x_3 and x_4 whose domains are all $\{0, 1\}$. The constraints are $x_1 + x_3 \leq 1$, $x_1 + x_4 \leq 1$, $x_2 + x_3 \leq 1$, $x_2 + x_4 \leq 1$ and $x_3 + x_4 \leq 1$. A lookup table, $a[i][j]$, is used to determine the amounts corresponding to variable values and the objective function, if we wish to optimize, is to maximize the sum of these amounts, $max \sum_{x_i \in V} a[i][x_i]$.

3 Super Solutions & Combinatorial Auctions

The purpose of finding super solutions is that if the solution changes slightly, another good solution may be found by changing a limited number of other variables. An (a,b) -super solution is one in which at most a variables may lose their values and a repair solution may be found by changing at most b variables [5]. Only a particular set of variables in the solution may be subject to change and these are said to be members of the *break-set*. For each variable in the break-set, a *repair-set* is required that comprises the set of variables whose values may change to provide another solution.

In an auction, for example, bids may be retracted so are included in the break set. However, if we also used variables to represent the items, these may not break if items may not be withdrawn from an auction after the winners are announced. Robust solutions are particularly desirable for combinatorial auctions, as opposed to single-unit auctions, because bid withdrawals/disqualifications can leave the auctioneer facing the exposure problem that is faced by bidders in single unit auctions. The exposure problem refers to the situation where bidders are left holding a set of items that are essentially valueless. Ideally this set of items would complement items associated with some of the successful bids so the solution may be repaired easily or the items are valued highly by another bidder. In the case of a single-unit auction the solution is inherently robust because if a bid is withdrawn then the second highest bid for that item is chosen as a repair.

Example 2. Let us consider a simple example with two variables X and $Y \in \{0, 1\}$. There are three possible solutions $\langle 1, 1 \rangle$, $\langle 0, 1 \rangle$ and $\langle 1, 0 \rangle$. $\langle 1, 1 \rangle$ could be considered

a (1,0)-super solution because if either of the variables breaks we are still left with a solution after making 0 changes. However, solutions $\langle 1, 0 \rangle$ and $\langle 0, 1 \rangle$ may be regarded as (1,1)-super solutions because we can always repair one variable if necessary to form another solution.

Similarly, some variables in the break set may not break depending on their value in the solution. Winning bids may be retracted, whilst retraction of losing bids is meaningless and has no effect on the solution. Super solutions need only be recorded for winning bids, hence assignments representing losing bids are said to have robust values.

Robust values aid the search for super solutions because we do not have to worry about finding repair solutions for the possible failure of those variables. In an auction scenario, solution failure is only likely to occur when one or more successful bids are withdrawn or disqualified. Consider Example 1, the optimal solution is $x_1 = 1, x_2 = 1, x_3 = 0$ and $x_4 = 0$. There are two winning bids involved in this solution so we only need to worry about repair solutions for variables x_1 and x_2 . If x_3 , for example, was withdrawn from the auction after the announcement of the winners, it would have no material effect on the solution. The WDP is an optimization problem that seeks to maximize revenue. Therefore, as well as being concerned with finding robust solutions, we also wish to find robust solutions with maximal revenue. The following example highlights this point.

Example 3. Consider the auction given in Example 1, with the additional constraint that we require a robust solution and we are willing to compromise on revenue but only by 20%. To be more precise, a (1,1)-super solution that matches the revenue constraint is required. This means that if any winning bid in the solution is withdrawn, a repair solution with revenue within 20% of optimum is necessary. In this case, if we were to choose the optimum solution $\langle 1, 1, 0, 0 \rangle$ we would be in trouble if either of the bids were withdrawn. Say x_1 was withdrawn or disqualified for some reason, the next best solution in terms of revenue is $\langle 0, 0, 1, 0 \rangle$. However, two variables would need to be changed in order to find a repair, therefore the optimum solution is not a (1,1)-super solution. However, $\langle 0, 0, 1, 0 \rangle$ is a (1,1)-super solution because a repair $\langle 0, 0, 0, 1 \rangle$ exists. This repair is in itself a super solution, but is dominated in terms of revenue by the previous solution therefore not chosen. Therefore, the best outcome in this situation is to select the (1,1)-super solution $\langle 0, 0, 1, 0 \rangle$, if we wish to find the maximal revenue for a robust solution within 20% of the optimum.

It is possible to reformulate the model so that the the only solutions are super solutions [5, 3]. However, we use search because [5] provides evidence of its superior performance over reformulation. Finding super solutions can be computationally expensive [4] so a pure CP approach to the WDP that has an exponential search space does not scale very well. A hybrid approach incorporating OR techniques is required for larger auctions. Section 4 concentrates upon a fixed size of problem (20 items and 100 bids) and shows how (1, b)-super solutions are achievable for tighter revenue constraints in auctions with shorter bids³. We refer the reader to [4] for a discussion on the difficulty of computing super solutions compared to straightforward CSPs. Some

³ Few items in each bid.

hybrid techniques that could aid the scalability of solving combinatorial auctions are briefly outlined also.

4 Experimental Results

In this section we consider the effect of considering robustness in combinatorial auctions. We consider a number of aspects: finding $(1,b)$ -super solutions, optimizing robustness and optimizing revenue. We used the Combinatorial Auction Test Suite (CATS) [7] to generate sample auction data. We generated sample auction problems in which there are 20 items for sale and 100 non-dominated bids⁴ that produce 100 CSP variables. We have examined smaller combinatorial auctions because a pure CP approach needs to be augmented with global constraints that incorporate OR techniques to increase pruning sufficiently so that thousands of bids may be examined. There are a number of ways to accelerate the search for super solutions in combinatorial auctions although this is not the focus of our work. Polynomial matching algorithms may be used in auctions whose bid length is very small, such as those for airport landing/take-off slots. Another technique may be to use an LP relaxation of the SPP to form an upper bound on potential revenue in sub-branches of the search tree. This is soluble in polynomial time and can therefore greatly improve performance. Such additional techniques, that are outlined in [10], can aid the scalability of a CP approach but our aim in these experiments is to focus upon the robustness of various auction distributions and consider the tradeoff between robustness and revenue. Our experiments use the EFC constraint solver [1] with additional super solution extensions developed by Hebrard et al [4, 5].

Twenty instances of each problem were used to generate average results. We used various distribution types that simulate different economically motivated auction scenarios. The `regions-npv` distribution is modelled on a scenario in which items are location dependent and complementarity is a function of the proximity of these items in 2-dimensional space, such as in a spectrum or real estate auction and valuations are distributed normally. In the `arbitrary-npv` distribution, complementarities may not be as universal as geographical adjacency with valuations being normally distributed again. Bidders view the complementarity of items slightly differently. The `scheduling` distribution simulates an auction for time slices on a resource in a distributed job-shop scheduling problem. The bids for this distribution type tend to be shorter, therefore there are a more combinations of possibly successful bids. This increases the difficulty of finding the optimal winner in the WDP but also increases the likelihood of being able to find a robust solution because of the increased availability of repair solutions.

4.1 Constraint Satisfaction

In this experiment we first solve the auction optimally. This can be done using an efficient IP solver such as ILOG's CPLEX package or CABOB[11]. We then stipulate a minimum percentage of optimum revenue that is acceptable and the maximum number of variables that can change, b , when forming a super solution. We then use the

⁴ The CATS flags included `int.prices` with the `bid.alpha` parameter set to 1000.

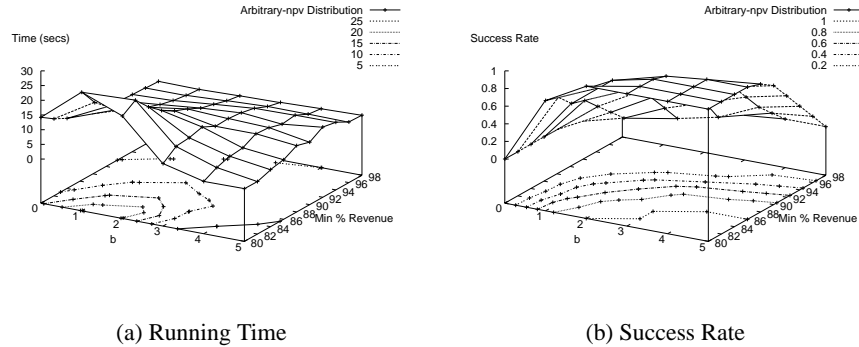


Fig. 1. arbitrary-npv distribution - finding a $(1,b)$ -super solution.

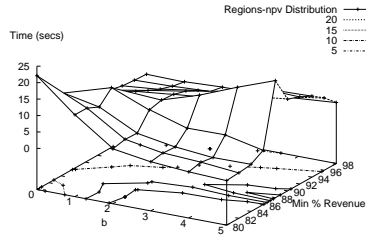
constraint-based solver to search for a satisfactory super solution. An IP approach to establishing such robust solutions would be extremely difficult to implement. The results are presented in Figures 1,2 and 3.

The contours on the horizontal plane of the graphs indicate the gradient of the surface in the graph. This helps illustrate the rates of fall-off in the running time and success rate for the different distributions. It is evident from these contours that the `scheduling` distribution reaches a very high success rate with $b = 1$ and acceptable revenue of at least 90%, for example (Figure 3). The contours also help show where the peak running times are encountered for the various distributions.

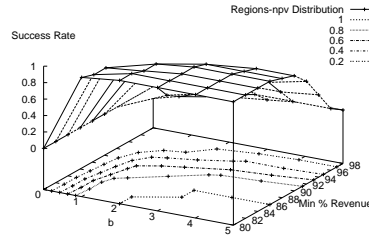
These figures indicate the increasing levels of complexity for the various distributions: the `arbitrary-npv` being the easiest and `scheduling` distribution being the hardest. We can surmise from these graphs that when the constraints are tight, (b is low and minimal revenue is high), that it is easy to find that there is no super solutions so running times are short. When constraints are very relaxed it is then easy to find a super solution. However, there is a phase transition area where the existence of a super solution is unpredictable and the running times are high. This peak in complexity is most clearly visible in Figure 3(a). The `arbitrary-npv` and `regions-npv` are evidently easier problems to solve. It is to be expected that running times increase with b because the search for a repair solution is longer. It can decrease in some instances when an increase in b results in many more solutions.

We can estimate from Figures 1,2 and 3 that the hardest satisfaction problems for the various distributions occur when the success rate is approximately 75%.

It is more difficult to solve auctions in which there are many *short* bids (involving a small number of items) optimally because there are fewer constraints between the bid variables and deeper traversal of the search tree is required. We are seeking a robust solution within a given percentage of optimal revenue such that if any successful bid is withdrawn, a repair solution that is also within the same percentage of the optimum can be found by changing at most b other variables. In Figures 1, 2 and 3 we varied b from 0

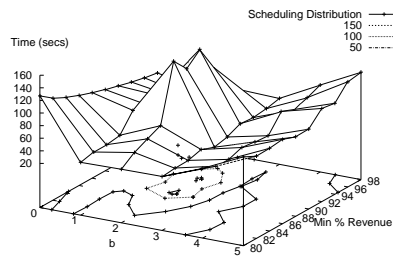


(a) Running Time

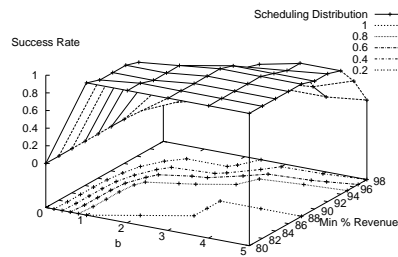


(b) Success Rate

Fig. 2. regions-npv distribution - finding a $(1,b)$ -super solution.



(a) Running Time



(b) Success Rate

Fig. 3. scheduling distribution - finding a $(1,b)$ -super solution.

to 5 and accepted the first super solution found. We did not consider values of b greater than 5 because most of the auction solutions contain 5 or less winning bids.

The arbitrary-npv distribution has long bids therefore few combinations of bids form valid solutions. This leads to fewer solutions and reduced time to solve. The regions-npv distribution has slightly shorter bids, therefore it is more difficult to solve than arbitrary-npv but easier than the scheduling distribution that has many short bids. The problem difficulty increases but the availability of robust solutions also increases because there are more possible repair solutions above the minimum threshold revenue. Figure 3(b) shows how the success rate for the scheduling distribution is better than for the arbitrary-npv and regions-npv distributions (see Figures 1(b) and 2(b) respectively). The increased availability of repair solutions accounts for the steeper contours towards 100% satisfiability in Figure 3(b).

4.2 Constraint Optimization

In a real-life scenario an auctioneer may seek a robust solution but would like the best robust solution in terms of either robustness or revenue. We may then employ an anytime algorithm that searches for the optimal super solution with maximum revenue being the objective function. We use a branch and bound algorithm that finds super solutions and optimizes on either reparability in the case of an over-constrained problem or revenue when there are many super solutions that satisfy the given constraints. This can be regarded as an anytime algorithm that finds the best possible robust solution in a given time-frame. Our analysis focuses on two forms of optimization:

1. **Optimizing Robustness:** Intended for use in an over-constrained scenario in which there are no super-solutions. We seek a solution that contains a minimal number of irreparable variables.
2. **Optimizing Revenue:** Intended for use in an under-constrained scenario in which there are many super solutions. We seek the super-solution with maximal revenue.

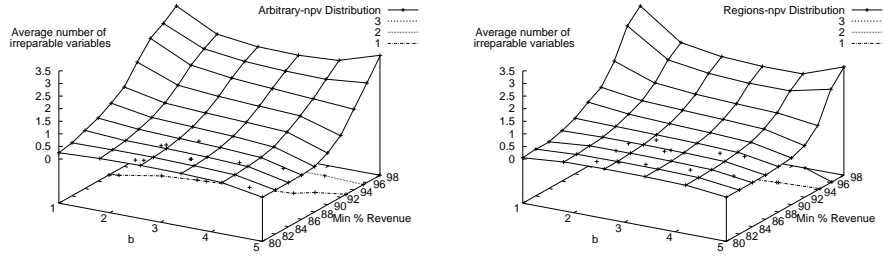
Optimizing Robustness A scenario may occur where there exists no $(1,b)$ -super solution that satisfies the minimal revenue criterion. We may seek a solution that minimizes the number of irreparable variables in a super solution thus compromising b but maintaining the revenue constraint. Hebrard et al. [4] have developed a $(1,b)$ -super Branch&Bound algorithm that will find a super solution with a minimal number of irreparable variables if no super solution satisfies the given value of b .

Figure 4 shows how many variables in the super-solution will not provide a repair solution with at most b variable values being changed to form a new solution that supports the revenue constraint. These results show that easier distributions to solve are less supportive of robust solutions⁵. When the revenue constraint on an auction with an `arbitrary-npv` distribution type is tight (95%) and few changes can be made to the solution ($b \leq 2$) then there are typically 2-3 bids in the auction that will not provide a repair solution given those constraints on revenue and b . The `regions-npv` and `scheduling` auctions have shorter bids and denser solution spaces so it is possible to find super-solutions that support such tight constraints.

Figure 5 compares the number of irreparable variables in the case of $b = 0$. Recall that a $(1,0)$ -super solution is a solution in which if any winning bid is withdrawn, then the remaining winning bids still form a valid solution. This is only possible when there are many small winning bids whose value is less than that of the tolerable loss in potential revenue. This is a very tight constraint that is in fact unsatisfied by any of the sample auctions in our test-set. However, we can attempt to find a solution that minimizes the number of winning bids that do not satisfy this constraint, or *irreparable* variables. Figure 5 shows the increase in the minimum number of such variables as the constraint on minimum revenue is tightened. As this constraint is tightened, there is a trade-off against robustness.

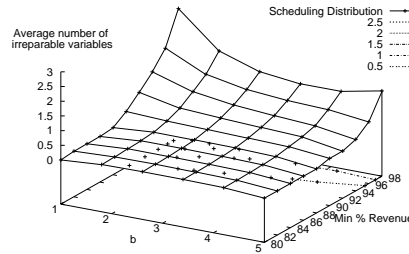
There are two principle factors governing the reparability of a solution, the number of winning bids and the number of possible repair solutions. The `arbitrary-npv`

⁵ Recall that auctions with long bids have fewer combinations of possible solutions so are easier to solve.



(a) arbitrary-npv distribution

(b) regions-npv distribution



(c) scheduling distribution

Fig. 4. Optimizing robustness.

distribution has fewer winning bids and fewer solutions so its reparability degrades rapidly. The regions-npv and scheduling auctions tend to have more available repair solutions therefore degrade more slowly as the revenue increases. The latter distribution has more winning bids participating in the solutions. This accounts for why more of those bids do not have repair solutions. Also, there are more solutions in this distribution hence the increase in minimal revenue leads to a lower slope in the number of irreparable variables.

Optimizing Revenue If there are many $(1,b)$ -super solutions satisfying the revenue constraints then it is desirable to find the revenue-maximizing super solution. This a much more difficult problem than finding *any* super solution given a constraint on revenue.

We have developed a branch and bound algorithm that returns the optimal super-solution in terms of an objective function. In our case we search for a super solution

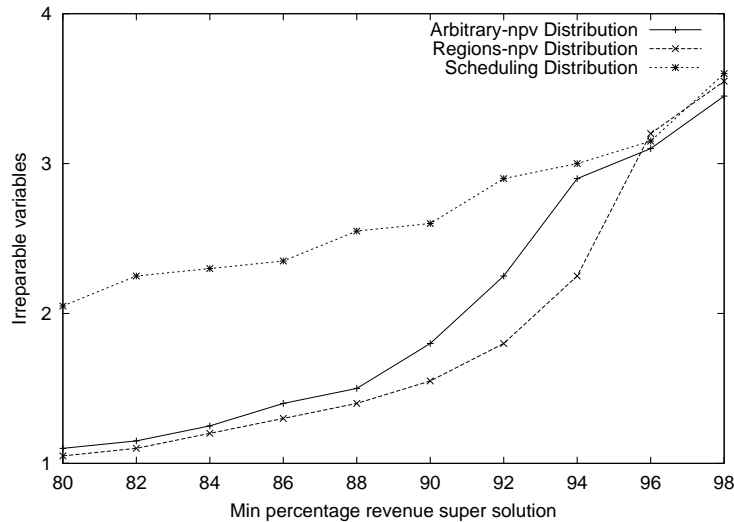


Fig. 5. Average number of irreparable variables in most robust solution (with $b=0$)

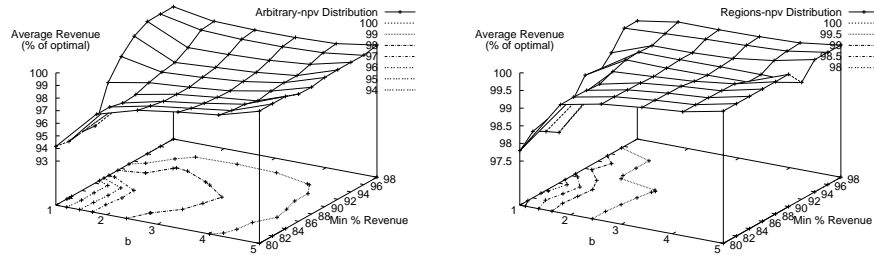
whose revenue is maximal whilst the constraints on the revenue for repair solutions remain the same⁶.

Figures 6(a), 6(b) and 6(c) show clearly that when we permit more variable changes, the expected increase in revenue of the optimal super solution increases significantly. Notice how optimization is far more difficult than satisfaction. This can clearly be seen by comparing the running times in Figure 7 with those in Figures 1, 2 and 3. We have restricted our analysis of the `scheduling` distribution to revenue greater than 90% because there are so many super solutions for revenue lower than this that optimization becomes extremely difficult. However, it is more difficult to see that as the constraint on acceptable revenue for repair solutions is tightened (minimum revenue for repair solutions increases) that in some cases this leads to a super solution of reduced optimal revenue. Note, however, that these graphs are averaged over those instances that proved satisfiable so as minimum revenue increases some problems became unsatisfiable therefore negating the decrease in revenue from satisfiable instances.

5 Extensions to Super Solutions

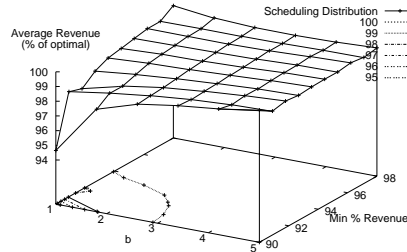
Whilst the super solution framework provides an excellent framework for finding robust solutions, it is somewhat inflexible in some respects that are important to real-life applications. We have discovered some limitations in the approach when applied to combinatorial auctions. Firstly there is an underlying assumption that when a repair solution is created, the incurred cost of changing each variable's value in the repair set is

⁶ An alternative approach may be to maximize the minimal revenue on the super solution and all repair solutions.



(a) arbitrary-npv distribution

(b) regions-npv distribution



(c) scheduling distribution

Fig. 6. Average optimal revenue of satisfiable instances.

the same. In a real-world scenario, informing a losing bidder that they have now won because of the withdrawal/disqualification of a winning bid would typically incur less cost than informing a winning bidder that they have now lost. This can be seen as a disadvantage of the super solution framework and militates against its deployment in real-life scenarios. Calculation of the cost associated with changing the losing/winning status of any bid is in reality a more complex issue that may depend on several other factors. Testing the legality of a repair solution by measuring the cardinality of the repair set may be overly restrictive in many application domains.

For example, the cost associated with changing the value of a variable in a solution may depend on its destination value. Consider a factory scheduling problem where variables represent machines and values correspond to states. The cost of changing the state of any machine depends on both the source and destination states.

The cost of changing a variable may also depend on the variable that caused the break. For example, if a particular agent withdraws a bid from an auction, the auctioneer may favour rejection of the agent's other successful bids rather than disturbing an

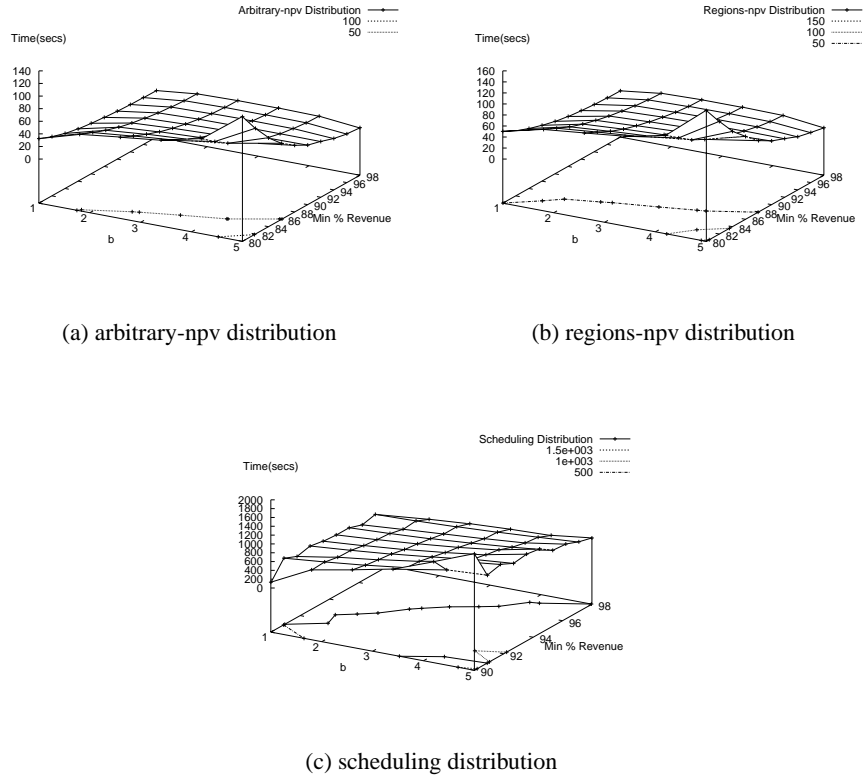


Fig. 7. Average time to find the super solution of maximal revenue.

innocent party. The cost of a repair solution should therefore depend on the destination value and the breakage variable(s). Furthermore the cost of repair may also be a function of time.

One possible approach is to extend the super solution framework to take account of the cost of repair. For example, we can use the concept of *inertia* as a metric for the cost of repair required to form an alternative solution. Previously the cardinality of the repair set was used to measure the cost of repair. We argue that changing some variables in a repair solution incurs less cost than others thereby motivating the introduction of a different metric for determining the legality of repair sets. We motivate this approach by using robust solutions to combinatorial auctions as an example application domain.

Hebrard et al [4] also described how some variables may fail (such as machines in a job-shop problem) and others may not. If we generalize this approach so that there is a probability of failure associated with each variable value, we can then alter the criteria for repair solutions according to each individual potential break. The *inertia* of each

value may be viewed as a measure of its aversion to change, therefore can be used to determine the cost of repair in the repair solution.

This extended feature of super solutions is motivated by the maintenance of robust solutions in combinatorial auctions. Agents' bids may be regarded as variables and Hebrard *et al's* (a,b) -super solution guarantees that an alternative solution may be found if a bids are retracted by changing at most b other variables. Maintaining a record of the reliability of assignments in the break set can help find repair solutions of lower *inertia* for values that are more likely to fail. All bids may not be viewed equally by the auctioneer and preferences may be shown for some agents. Eliminating a certain bid from the solution because of the retraction of another bid may incur varying costs for each bid, therefore the *inertia* associated with a value representing a winning bid reflects the unwillingness of the auctioneer to let that bid lose due to the actions of another.

Losing bidders typically would not mind being told that their bid was now accepted due to a winning bid's retraction so the *inertia* associated with such bids may be lower. This may be seen as a generalization of super solutions whose values of inertia are all 0 or 1. Furthermore, the *break-set* is regarded as the set of variables that may break, therefore the *inertia* of these variables values are all 1. Those variables not in the break-set have coefficients of inertia of 0 for all their values.

The overall cost of repair required to move from one solution to another is a function of the inertia on all values that need to be changed. An example of such a function could be the sum of the inertial values although not exclusively so. Taking this approach further repairs may be made that restrict the impact on particular variables. In an auction scenario, we may wish that a repair solution does not impact too unfairly on any particular agent when another agent retracts a bid. The development of this extension to super solutions is our current topic of research.

6 Conclusion

Combinatorial auctions are becoming and increasingly popular means of selling/procuring items because they provide enhanced economic efficiency over traditional single unit auctions. Super solutions offer a promising platform for developing robust solutions for such auctions that may leave the auctioneer with an exposure problem if bids are withdrawn or disqualified. We have demonstrated the computational feasibility of finding super solutions for economically-motivated auction problems and shown how an any-time algorithm may find the best possible super solution given a limited time frame. We outlined some hybrid techniques that may aid the scalability of super solutions for larger auctions.

We also presented some limitations of the approach and suggested an extension to the framework. This extension incorporates a more flexible metric, termed *inertia*, for determining the legality of repair solutions and associating probabilities of failure to different variables. The introduction of weighted robustness of values complements this approach by allowing the construction of repair solutions of lower inertia for unreliable values. The development of this extension to super solutions is our current topic of research.

Acknowledgements

We are very grateful to Emmanuel Hebrard and Brahim Hnich for their assistance.

References

1. Fahiem Bacchus and George Katsirelos. EFC constraint solver. <http://www.cs.toronto.edu/~gkatsi/efc/efc.html>.
2. Rafael Epstein, Lysette Henríquez, Jaime Catalán, Gabriel Y. Weintraub, and Cristián Martínez. A combinatorial auction improves school meals in Chile. *Interfaces*, 32(6):1–14, 2002.
3. Matthew L. Ginsberg, Andrew J. Parkes, and Amitabha Roy. Supermodels and robustness. In *AAAI/IAAI*, pages 334–339, 1998.
4. Emmanuel Hebrard, Brahim Hnich, and Toby Walsh. Robust solutions for constraint satisfaction and optimization. In *Proceedings of ECAI 2004*.
5. Emmanuel Hebrard, Brahim Hnich, and Toby Walsh. Super solutions in constraint programming. In *Proceedings of CP-AI-OR 2004*, 2004.
6. Gail Hohner, John Rich, Ed Ng, Grant Reid, Andrew J. Davenport, Jayant R. Kalagnanam, Ho Soo Lee, and Chae An. Combinatorial and quantity-discount procurement auctions benefit mars, incorporated and its suppliers. *Interfaces*, 33(1):23–35, 2003.
7. Kevin Leyton-Brown, Mark Pearson, and Yoav Shoham. Towards a universal test suite for combinatorial auction algorithms. In *ACM Conference on Electronic Commerce*, pages 66–76, 2000.
8. Paul Milgrom. *Putting Auction Theory to Work*. Cambridge, March 2004.
9. Martin Pesendorfer and Estelle Cantillon. Combination bidding in multi-unit auctions. Harvard Business School Working Draft, 2003.
10. Tuomas Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135(1-2):1–54, 2002.
11. Tuomas Sandholm, Subhash Suri, Andrew Gilpin, and David Levine. CABOB: A fast optimal algorithm for combinatorial auctions. In *IJCAI*, pages 1102–1108, 2001.
12. W.E. Walsh, M.P. Wellman, and F. Ygge. Combinatorial auctions for supply chain formation. In *ACM Conference on Electronic Commerce*, pages 260–269, 2000.