A decision support system for winner determination in multi-objective combinatorial auctions

K. Balaji
IBM India Research Laboratory,
Block 1, Indian Institute of Technology,
Hauz Khas, New Delhi - 110 016, INDIA.
kbalaji@in.ibm.com

Abhinanda Sarkar *
John F. Welsch Technology Center,
GE Corporate Research and Development,
Bangalore, INDIA.
Abhinanda.Sarkar@geind.ge.com

Abstract

Deciding the winners in a combinatorial auction with multiple objectives is a notoriously hard decision problem. A support system which would assist the auctioneer in making such complex decisions, is a highly desirable tool for her in that scenario. This work addresses the design of such a tool. Using the decision support system the auctioneer can set preference on the various objectives and evaluate the multiple winner sets computed by the system to determine the winner set for an auction. For computing multiple winner sets the system uses an algorithm based on the Metropolis algorithm. Through simulation experiments, the proposed system is validated to be, (i) consistent - the response of the system matches the preferences set by the auctioneer, (ii) convergent - converges to a set of pareto-optimal solution (winner set) in a reasonable amount of time, and (iii) robust - scalable for large number of bids, bidders and items in the auction.

1 Introduction

Electronic marketplaces are becoming popular, as they facilitate the businesses to trade over a common platform using multitude exchange mechanisms. The exchange of items using auction is gaining momentum in these marketplaces, as they enable the participants to discover the true values of the items which are of interest to them. Combinatorial auctions facilitate the bidders to bid on combination of multiple items by specifying a value for the combination. This mechanism allows the bidders to express complementarities - the bid for the combination of items is worth more than the sum of the separate items, and

*This work was done when this author was with IBM India Research Laboratory, New Delhi.

substitutabilities - the bid for the combination is less than the sum of the separate items, as opposed to other multi-item exchange mechanisms like sequential auctions and simultaneous auctions. This provides a great motivation for the researchers to realize this auction type in practice [5][7][9] [13]. The increased interest in combinatorial auctions could also be accredited to the natural occurrence of this auction type in government privatization and rights allocation (such as FCC spectrum rights auctions [2], frequency band rights in geographically adjacent areas, etc. [12]), business to business (B2B) auctions where the values of the items may also turn out to be a function of other items obtained due to the internal pricing models of each business and in other applications [14].

The winner determination of single objective combinatorial auctions had been studied extensively in the literature [1][2][9] [10][11][12][14][15][16]. In this work, we study the winner determination of multi-objective combinatorial auctions which hasn′t been addressed in the literature as yet. More precisely, this work presents a decision support system for the auctioneer to determine the winner set. Using the system the auctioneer can set preference on the various objectives and evaluate the multiple winner sets computed by the system to determine the winner set for an auction. For computing multiple winner sets the system uses an algorithm based on the Metropolis algorithm. Simulation study on the benchmarks of [3], reveal that the system is, (i) consistent - the response of the system matches the preferences set by the auctioneer, (ii) convergent - converges to a set of pareto-optimal solution (winner set) in a reasonable amount of time, and (iii) robust - scalable for large number of bids, bidders and items in the auction. In this paper, eventhough the proposed algorithm is experimented only for linear combination of the multiple objectives, we note that the algorithm is applicable even when the combination of the objectives is a nonlinear function.

The paper is organized as follows. In Section 2, we define the problem and then present a support system and an optimization model for the same. An algorithm based on the Metropolis algorithm is presented in Section 3. Simulation experiments validating various qualitative parameters (consistency, convergence and robustness) of the system is presented in Section 4. We conclude in Section 5.

2 Problem definition and system model

In this section, we define the winner determination problem of multi-objective combinatorial auctions and then present an optimisation model and a decision support system for the same.

For a combinatorial auction the multi-objective winner determination problem is defined as follows. The auctioneer announces a set of items on combinatorial auction. The bidders bid on combination of these items quoting a single bid value for the combination. The multi-objective combinatorial auctions winner determination problem is to allocate the goods to the bidders optimising multiple objectives such as revenue, profit, number of bidders in the winning set, etc., based on the weights associated by the system to these objectives according to the preferences set by the auctioneer.

In this paper, we consider the following optimisation model for our analysis. Consider an auction with m items. Let $B_i = ((b_1^i, b_2^i, \ldots, b_m^i), v_i)$, $1 \leq i \leq n$, be the n bids of the auction. The term $b_j^i = 1$, if the i^{th} bid bids for the j^{th} item and $b_j^i = 0$, otherwise. Let v_i and p_i be the value and the profit[1] correponding to the i^{th} bid.

[1]The profit corresponding to the bid i, is defined as, $p_i = v_i - \sum_{j=1}^{m} c_j b_j^i$ where c_j is the cost price of item j.

The winner determination problem with two objectives, namely, profit and revenue, with their respective weights as w_1 and w_2, could be modeled as the following optimisation problem,

$$Maximize\, w_1 \sum_{i=1}^{n} p_i x_i + w_2 \sum_{i=1}^{n} v_i x_i, \text{ such that,}$$

$$\sum_{i=1}^{n} b_j^i x_i \leq 1 \text{ for } j = 1, 2, \ldots, m, x_i = 1 \text{ or } 0, \text{ for } i = 1, 2, \ldots, n$$

The process flow of the decision support system is as follows . The auctioneer sets the preferences to the multiple objectives of a multi-objective combinatorial auction. The system associates weights to the multiple objectives according to the auctioneer$'$s preference, fetches the bids of the auction and models the winner determination problem as described earlier in this section. After modeling, it computes optimal or nearly-optimal winner sets using the algorithm which is to be proposed in Section 4. The system after computing the winner sets, ranks and presents them to the auctioneer. The system ranks the winner sets by sorting them in non-increasing order of their objective value[2]. The winner sets presented to the auctioneer are feasible solutions[3] which are pareto optimal[4] or nearly-pareto[5] optimal solutions. The nearly-pareto optimal (dominated) solutions could also be preferred by the auctioneer if they have some non-quantifiable preferences among the bidders, such as, bidder A could be preferred over bidder B because A is more reliable to B in terms of delivering the items within the specified deadline.

3 The Algorithm

In this section, we present an algorithm for the winner determination problem of a combinatorial auction with multiple objectives. The proposed algorithm is based on the Metropolis algorithm [4][6][8].

The multi-objective combinatorial auction winner determination algorithm proceeds as follows. Initially, the algorithm by random sampling chooses a set of solutions M from the solution space, Ω. On these sampled solutions the algorithm iterates for T steps. For every iteration, the algorithm probabilistically decides either to stay in the same state that it is examining, or migrate to any of the neighbors of the examining state. After the T steps the algorithm outputs a set of solutions, which are pareto-optimal or nearly pareto-optimal solutions for the optimisation problem.

In the algorithm, the state space (solution space), the neighbors of a state (solution), x, and the transitions are defined as follows. The state space $\Omega = \{x \in \{0,1\}^n\}$, is the set of all allocations (feasible and infeasible) of bids to the goods. For a solution $x = (x_1, x_2, \ldots, x_n) \in \Omega$, its neighbors are $y = x$ and $y = (x_1, \ldots, \bar{x}_i, \ldots, x_n) \in \Omega$, where $1 \leq i \leq n$ and $\bar{x}_i = 0$ or 1, if $x_i = 1$ or 0, respectively. Transition from a state $x \in \Omega$ to its neighboring state $y \in \Omega$ is defined as,

Transition A : with probability $\frac{1}{2}$ stay in the same state x, that is $y = x$; otherwise,

[2]The solutions which have same objective value are sorted based on the preferences of the auctioneer.

[3]A solution to an optimisation problem is called a feasible solution if it satisfies all the constraints of the problem.

[4]In a multi-objective optimization problem, a feasible solution x is called a pareto optimal solution if it dominates the other feasible solutions in at least one objective (that is optimal in terms of at least one objective). The set of pareto optimal solutions is called the pareto optimal frontier.

[5]Solutions that are close to the pareto optimal frontier.

Transition B : with probability $min\{1, e^{f(y)-f(x)}\}$, go to $y \neq x$.

In Transition B, the function $f(x)$ is defined as the objective value of x, if x is a feasible solution. If x is infeasible, it is defined as the objective value of x', where x' is the feasible solution output by the following Greedy Algorithm for the input x.

`Greedy Algorithm :`

Input : An infeasible allocation, x.

I. Initialise, the winner set, $W = \emptyset$.

II. Sort the bids of x in decreasing (or non-increasing) order of their bid values. Let the sorted list be $x_1, x_2, \ldots, x_k$.

III. Traverse the sorted list from x_1 to x_k and include x_i in the winner set, if all the goods requested by the bid x_i can be allocated to it. Mark the allocated goods to x_i so that they can't be allocated to any other bid.

IV. x' is set to W.

Output : The solution x'.

The following algorithm outputs a set of pareto-optimal or nearly pareto-optimal solutions for the winner determination problem.

`Multi-Objective Combinatorial Auction Winner Determination Algorithm :`

1. Fix the number of iterations, T, for simulation.

2. A set of M solutions are choosen at random from the solution space Ω. These are the initial solutions for the simulation.

3. With respect to each of the M initial solutions, do Steps 4 to 6.

4. Set x to be the initial solution.

5. For each of the T iterations, update x to the solution obtained by migrating from x to one of its neighbors, using Transition A or B.

6. At the end of T iterations output, x, if x is feasible, otherwise, output x', the solution output by the Greedy Algorithm for the input x.

4 Simulation Experiments

The algorithm proposed in Section 4 was experimented on the combinatorial auctions data sets : Random, weighted random, uniform, and decay [3] [15], which are described below.

(a) `Random :` For each bid, the number of items is picked randomly from $1, 2, \ldots, m$. That many items are chosen without replacement. The price is picked randomly from [0,1].

(b) `Weighted random :` As above, except for that the price is picked randomly between 0 and the number of items in the bid.

(c) `Uniform :` The items are chosen randomly. But, same number of items are chosen in every draw. The prices are picked from [0,1].

(d) `Decay` **:** Initially, one random item is chosen for the bid. Then, with probability α a new random item is added, until, it is not possible to add the chosen item or no item is left for choosing.

In our study, we classified the data as the auctions with small number of bids and large number of bids. The auctions with small number of bids consist of 50, 100, 150 and 200 bids. The number of bids were 500, 600, 700, 750, 800, 850, 900, 950 and 1000, for auctions with large number of bids. In some experiments we further classifed the data set as sparse and dense bids. Sparse bids bid on small number of items and dense bids bid on large number of items. We denote the subclass of small bids with sparse and dense properties as small-sparse and small-dense bids, respectively. Similarly, large-sparse and large-dense bids are the large bids with sparse and dense properties, respectively.

4.1 Consistency

For this study, in all the experiments, the number of iterations (T) were set to 10^6. The number of initial samples chosen for the data sets with small and large number of bids were 50 and 10, respectively. In both the data sets, the algorithm behaves as expected for the various preferential values set by the auctioneer (Figures 1 to 6). The plots shown in Figures 1 to 6 are for the cases where the small and large number of bids are 50

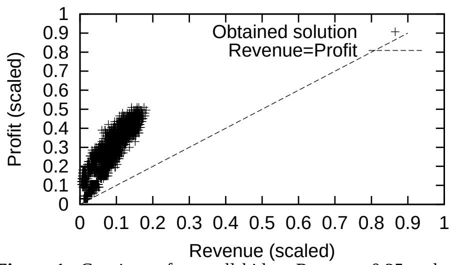

Figure 1: Consistent for small bids : Revenue=0.25 and Profit=0.75

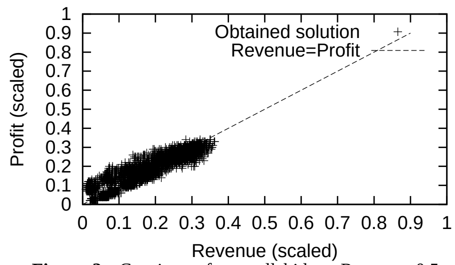

Figure 2: Consistent for small bids : Revenue=0.5 and Profit=0.5

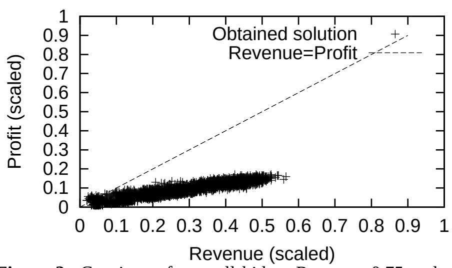

Figure 3: Consistent for small bids : Revenue=0.75 and Profit=0.25

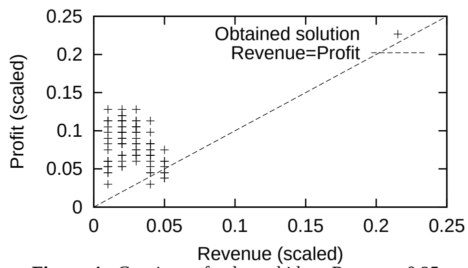

Figure 4: Consistent for large bids : Revenue=0.25 and Profit=0.75

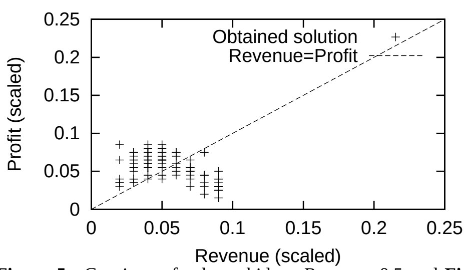

Figure 5: Consistent for large bids : Revenue=0.5 and Profit=0.5

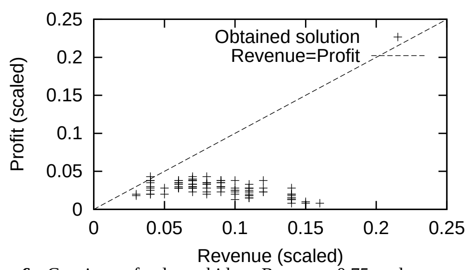

Figure 6: Consistent for large bids : Revenue=0.75 and Profit=0.25

and 1000, respectively. We observed similar behaviour for the remaining cases too depending on their class labels. For the case with small number of bidders, from Figures 1 to 3 it can seen that, (i) all the solutions are in the region $Profit > Revenue$, when Weightage for revenue=0.25 and Weightage for profit=0.75, (ii) the solutions are equally distributed with respect to the line $Revenue = Profit$, when both Weightage for revenue and Weightage for Profit are 0.5, and (iii) all the solutions are in the region $Profit < Revenue$, when Weightage for revenue=0.75 and Weightage for Profit=0.25. These imply that the solutions output by the algorithm are according to the weights set by the auctioneer on the objectives revenue and profit. The Figures 4 to 6 show that, the same is true for the case of auctions with large number of bids too.

4.2 Convergence

The study of convergence of the proposed algorithm were done on the bid classes small-sparse, small-dense, large-sparse and large-dense.

For small-sparse and small-dense structures we ran the algorithm upto 10^7 iterations. For the experiments, the number of bids chosen were 50,100,150 and 200 for small-dense class and 50,100 and 150 for small-sparse class. The number of initial samples chosen were 10, for this study. After every 10^i iterations $i = 2$ to 7, the best of the 10 solutions was compared against the optimum value obtained using the LP-Solve[6] and the ratio of Metropolis solution to LP-Solve solution is plotted[7] in the Figures 7-8. It can be seen from the plots that after 10^6 iterations that the proposed algorithm attains the optimum, for the cases of 50 bids and 100 bids in small-dense structure and 50 bids in small-sparse structure. It is due to the dense neighborhood structure formed in the solution space the algorithm attains optimum for most of the cases in small-dense structure. In the case of auctions with small number of bids from the plots it can be inferred that after 10^6 iterations the best solution obtained is about 95 % of the optimal value.

Similar study was done for the case of large-dense and large-sparse structures. For this, the algorithm was run on 10^6 iterations. In this case the optimal value obtained by the algorithm is compared with optimal value obtained by LP-Solve for the relaxed integer linear program rather than the integer version. In these cases, the number of initial samples chosen was 50. The intution for choosing more initial samples than in the case of small structures is that the number of solutions in the solution space is very high in this case as compared to the small structures. The plots (Figure 9 and 10) show that the algorithm would converge to the optimum after a reasonable number of iterations.

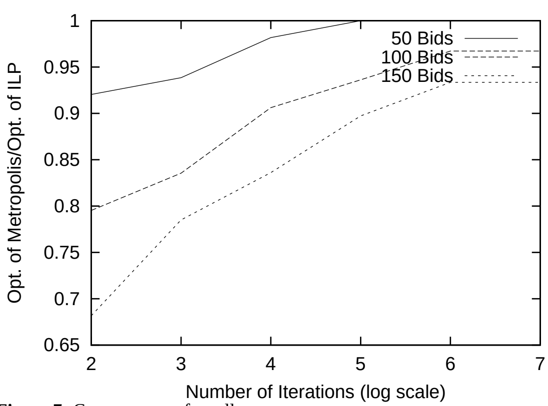

Figure 7: Convergence of small-sparse structure

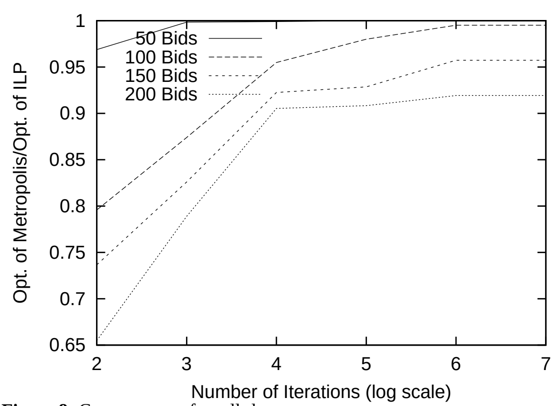

Figure 8: Convergence of small-dense structure

[6]Optimisation Library available for download at $ftp://ftp.es.elu.tue.nl/pub/lp_solve$. This is a non-commercial version.

[7]The x-axis in these plots is in log-scale.

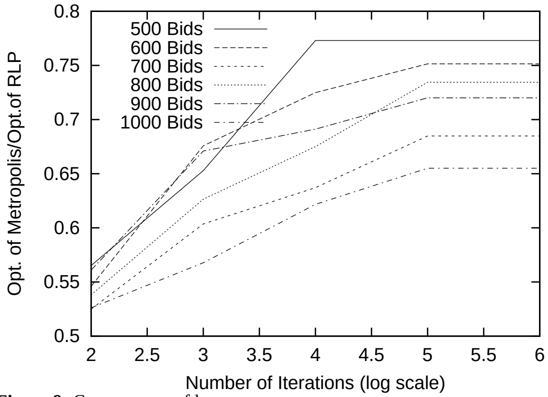

Figure 9: Convergence of large-sparse structure

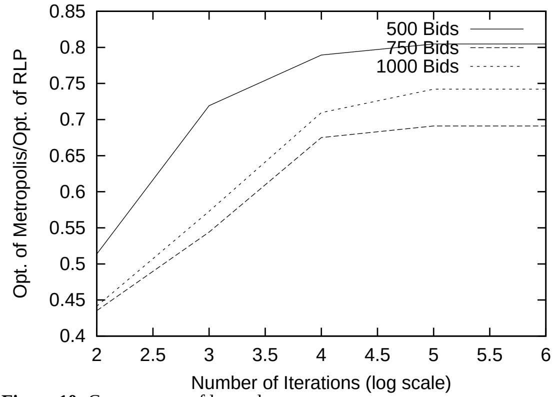

Figure 10: Convergence of large-dense structure

4.3 Robustness

In the simulations for convergence we observed that the convergence of the algorithm strongly depends on the number of iterations rather the parameters like number of bids, number of bidders and number of items in auction. Hence, the system would be scalable and be stable with respect to these parameters.

5 Conclusion

In this paper we presented a decision support system for the winner determination problem in multi-objective combinatorial auctions. Using this, the auctioneer can study the sensitivity of the various objectives and decide the winning set. A randomized algorithm based on the Metropolis algorithm was presented. The presented algorithm computes multiple winner sets which would be optimal or nearly-optimal with respect to the various objectives. By simulation experiments, the proposed system was shown to be (i) consistent - the response of the system matches the preferences set by the auctioneer, (ii) convergent - converges to a set of pareto-optimal solution (winner set) in a reasonable amount of time, and (iii) robust - scalable for large number of bids, bidders and items in the auction.

Acknowledgments

The authors would like to thank R.Ananthapadmanabha, K.L.Karthik, and K.N.G.Krishnaram for their contribution in the initial stages of this work. We thank Sven de Vries and Rakesh Vohra for providing us the test data. The first author would like to thank Ravi Kothari and Arthi for their valuable suggestions in presenting the work.

References

[1] Balaji, K., Barik, R., Kumar, A., & Kundu, A. (2001) An allocation algorithm for generalized bids in combinatorial auctions. the Proceedings of 9th International Conference on Advanced Computing and Communications held at Bhubaneshwar, India, December 16-19.

[2] Cramtom, P.C. (1997) The FCC spectrum auction: an early assessment. Journal of Economics and Management Strategy 6(3):431-495.

[3] de Vries, S., & Vohra, R. (2000) Combinatorial Auctions : A survey. Technical Report (discussion paper) no. 1296, The Center for Mathematical Studies in Economics and Management Science, Northwestern University.

[4] Diaconis, P., & Saloff-Coste, L. (1998) What do we know about the Metropolis algorithm?. J. Comp. System Sci., 57:20-36.

[5] Fujishama, Y., Leyton-Brown, K., & Shoham, Y. (1999) Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. Proceedings of IJCAI'99, Stockholm, Sweden. Morgan Kaufmann.

[6] Gilks, W.R., Richardson, S., & Spiegelhalter, D.J. eds. (1996), Markov Chain Monte Carlo in Practice. London : Chapman and Hall.

[7] Gonen, R., & Lehmann, D. (2000). Optimal solutions for multi-unit combinatorial auctions : branch and bound heuristics. Proceeding of ACM EC-00, Minneapolis, Minnesota, October 2000, pp. 13-20.

[8] Jerrum, M., & Sinclair, A. (1996) The Markov chain Monte Carlo method : an approach to approximate counting and integration. In: Approximantion Algorithms for NP-hard Problems, (Dorit Hochbaum, ed.), PWS.

[9] Lehmann, D., O'Callaghan, L.I., & Shoham, Y. (1999) Truth Revelation in Approximately Efficient Combinatorial Auctions. Proceedings of ACM EC-99, Denver, Colorado. pp.96-102.

[10] Leyton-Brown, K., Shoham, Y., & Tennenholtz, M. (2000) An algorithm for multi-unit combinatorial auctions. Proceedings of Seventh National Conference on Artificial Intelligence (AAAI-2000). pp. 66-76.

[11] Nisan, N. (2000) Bidding and allocation in combinatorial auctions. Proceeding of ACM EC-00, Minneapolis, Minnesota, October 2000. pp. 1-12.

[12] Rothkhof, M.H., Pekec, A., & Harstad, R.M. (1998) Computationally manageable combinatorial auctions. Management Science 44(8):1131-1147.

[13] Sakurai, Y., Yokoo, M., & Kamei, K. (2000) An efficient approximate algorithm for winner determination in combinatorial auctions. Proceeding of ACM EC-00, Minneapolis, Minnesota, October 2000. pp. 30-37.

[14] Sandholm, T. (1993) An implementation of contract net protocol based on marginal cost calculations. Proceedings of AAAI, 1993, August 1993.

[15] Sandholm, T. (1999) An algorithm for optimal winner determination in combinatorial auctions. IJCAI-99.

[16] Sandholm, T. (1999) eMediator: A next generation electronic commerce server, AAAI Workshop Technical Report WS-99-01. pp. 46-55.

[17] Zurel, E., & Nisan, N. (2001) An Efficient Approximate Allocation Algorithm for Combinatorial Auctions. Proceedings of EC'01, October 14-17, 2001, Tampa, Florida, USA.