A Comparison of Bidding Strategies for Simultaneous Auctions

Teddy Candale
and
Sandip Sen
University of Tulsa

Bidding for multiple items or bundles on online auctions raise challenging problems. We assume that an agent has a valuation function that returns its valuation for an arbitrary bundle. In the real world all or most of the items of interest to an agent is not present in a single combinatorial auction. We focus on bidding for multiple items in a set of auctions, each of which sell only a single unit of a particular item. Hence an agent has to bid in multiple auctions to obtain item bundles. While an optimal bidding strategy is known when bidding in sequential auctions, only suboptimal strategies are available when bidding for items sold in auctions running simultaneously. We investigate a hill-climbing bidding strategy, which is optimal given an infinite number of restarts, to decide on an agent's bid for simultaneous auctions. We provide a comparison of this algorithm with existing ones, both in terms of utilities generated and computation time, along with a discussion of the strengths and weaknesses of these strategies.

Categories and Subject Descriptors: I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Coherence and coordination, Multiagent systems, Intelligent agents*

General Terms: Performance, Experimentation

Additional Key Words and Phrases: auctions, bidding, multi-dimensional bid improvement

1. INTRODUCTION

This paper deals with bidding strategies for purchasing multiple items or bundles from simultaneous electronic auctions. Auction theory has received significant attention from agent researchers following the development of electronic auctions on the Internet. Researchers are interested both in designing auctions with desirable properties [Parkes 2001; Sandholm 2000; Sen et al. 2005; Wurman et al. 1998] and designing automated agents to represent interests of human users [Greenwald and Boyan 2004; Stone and Greenwald 2003]. Bidding strategies for single-item auctions are well-known. For strategy-proof auctions, a rational bidder will bid its true valuation for the item. The problem of computing optimal bids is more complex when bidding for multiple items. The valuation function of a potential buyer expresses the maximum amount it is willing to pay to acquire each bundle of items. In a multi-auction setting, multiple single-item auctions are run concurrently or sequentially. A potential bidder needs to estimate closing prices of such auctions to compute optimal bids. In multi-auction settings, auctions can be *sequential* or *simultaneous*. Sequential auctions close at a predetermined known order. Simultaneous auctions run in parallel. Bids to be submitted to all auctions have to be computed simultaneously.

Bidding in simultaneous auctions has received a lot of attention partly due to the Trading Agent Competition (TAC, http://www.sics.se/tac/). A TAC agent represents clients for which it has to organize travel packages. Travel goods consist of airline tickets, hotel reservations, tickets for entertainment events, etc. The agent has to obtain those items in single-item auctions as inexpensively as possible to maximized profits while offering attractive bundles to its clients.

Bidding in multiple auctions have also been studied in other contexts. For example, Boutilier et al. have studied the bundle bidding problem for allocating resources [Boutilier et al. 1999]. In their model, agents share a pool of resources and need particular resources to perform specific tasks. Resources are auctioned sequentially and price distributions are learned online. Bids are computed using a Markov Decision Process and the optimal policy is determined using value iteration.

Bid computation often assumes discrete closing price distributions as minimum bid increments in some real-life auctions lead to discrete closing prices. Even with discrete bids, however, calculating optimal bids using brute force methods leads to exponential complexity. If N is the number of items and n_b is the maximum number of possible bids for any item, deciding optimal bids requires $O(n_b^N)$ expected utility calculations. To reduce this complexity, the marginal utility bidding approach has been used [Greenwald and Boyan 2004; 2005; Stone et al. 2003]. The marginal utility of an item is the additional profit earned by acquiring that item. Profits are calculated by estimating future closing prices of items to be acquired. While this method is optimal for sequential auctions, it is suboptimal for simultaneous auctions. We believe that part of this problem is due to the loss of information about the bidder's preferences. For example, if a bidder has the same valuation for two items A and B, i.e., $\vartheta(\{A\}) = \vartheta(\{B\})$, and these two items have the same closing price distribution, the bidder will always bid the same value for A and B. When those items are substitutable, i.e. $\vartheta(\{A\}) + \vartheta(\{B\}) > \vartheta(\{A, B\})$, this may not be optimal since it may be preferable to have a high bid for one and a low bid for the other. In sequential auctions, the extra information received from the outcome of completed auctions helps avoid this problem.

We have recently developed a multi-dimensional bid improvement technique to determine bids that maximize the expected utility function. We reason with continuous closing price distributions as this allows us to address a more general setting and working with a continuous space allows us to apply powerful optimization methods not available for discrete spaces. Our technique can also be applied to find optimal bids given discrete closing price distributions. While we present the motivation, characteristics and other technical details in an auxilliary paper (under review), the focus of this paper is an experimental comparative evaluation of our proposed algorithm with existing ones in terms of time complexity and the quality of the solutions generated. We experiment with a variety of valuation functions and closing price distributions to gauge both the absolute performance, in terms of optimality, of our approach as well as relative solution quality and computational efficiency of our approach when compared to existing approaches.

We provide a more detailed description of bidding algorithms for two top-scoring agents [Greenwald and Boyan 2004; 2005; Stone et al. 2003; Stone and Greenwald 2003] in the Trading Agent Competition (TAC) in Section 3 and offer a comparison of the effectiveness of their algorithms and ours in Section 4.

2. SIMULTANEOUS AUCTION MODEL

We consider a bidder which plans to acquire items from the set $\mathcal{I} = \{1, \ldots, N\}$. A valuation function ϑ represents the bidder's preference by assigning a value the bidder is willing to pay for each bundle. Item i is available in only one single-item single-unit auction a_i. An auction is modeled by the cumulative probability distribution F_i of the closing prices in the range $[\underline{p_i}, \overline{p_i}]$. We assume these distributions to be continuous, independent and known to the bidder. When a_i closes, the bidder gets the item if it has placed a bid b_i greater than or equal to the closing price p_i ($b_i \geq p_i$) and the payment is equal to the closing price. All auctions are run in parallel and their closing times are not known by the bidder. $B = (b_1, \ldots, b_N)$ represents the bids placed simultaneously in all auctions by the bidder and $P = (p_1, \ldots, p_N)$ are the closing prices. We assume bidders to be rational, i.e., it wants to maximize its expected utility. Hence it tries to find a bid vector B^* such that $B^* = \underset{B \in \mathcal{B}}{\operatorname{argmax}}\, \bar{\alpha}(B)$ where $\mathcal{B}$ is the bid space and $\bar{\alpha}(B)$ is the expected utility of bidding B.

3. BIDDING STRATEGIES

We present below the bundle bidding strategies that we subsequently evaluate.

3.1 Marginal utility bidding ($\overline{MU}$)

We first present the optimal bidding strategy for bundles in sequential auctions [Greenwald and Boyan 2004; Stone et al. 2003]. In the remainder of this paper, we will refer to the method as *expected marginal utility bidding* or $\overline{MU}$. When bidding for the i^{th} item, the bidder places the bid $b_i = \bar{\mu}(i, I_h, I_r)$ in auction a_i. I_h contains items held by the bidder, I_r contains items to be auctioned, $\bar{\mu}(i, I_h, I_r)$ is the expected marginal utility of item i; it can be viewed has the extra-profit due to the acquisition of i at zero-cost.

For simultaneous auctions, a bidder has to compute all bids simultaneously. In that case, $\mathcal{I}_h = \emptyset$. The bidder bids for each item as if it is the first item to be auctioned. In other words, the bidder places $b_i = \bar{\mu}(i, \emptyset, \mathcal{I} \setminus \{i\},)$ in auction a_i.

$\overline{MU}$ is optimal when bidding in sequential auctions as shown by Greenwald in [Greenwald and Boyan 2004]. $\overline{MU}$ is suboptimal for simultaneous auctions. In particular, when items are substitutable, the bidder may acquire two items it may not desire to acquire together. Another downside of $\overline{MU}$ is its complexity. The calculation of the marginal utility is exponential since it requires the knowledge of the profit generated by each possible bundle.

3.2 Expected Value Marginal Utility bidding (EVMU)

In this section, we present a variant of $\overline{MU}$ introduced by Greenwald [Greenwald and Boyan 2004]. This variant tries to overcome the issue raised by leaving out of consideration some bidder preferences. To prevent the bidder from obtaining two "similar" items, a set of preferred items is precomputed. We refer to this set as the *acquisition set* $\mathcal{I}_{evmu}$. The acquisition set is the optimal set of items to be obtained assuming that items are auctioned at deterministic prices equal to the expected closing prices corresponding to these items. The bidder places a bid for all items in $\mathcal{I}_{evmu}$ and the bids are equal to the expected marginal utility of the items. Like

$\overline{MU}$, the computation of marginal utility in EVMU is exponential.

3.3 Multi-dimensional Bid Improvement (MDBI)

Now we present an approach to bidding in simultaneous auctions based on an incremental optimization technique [Candale 2005], a multi-dimensional bid improvement algorithm (MDBI). An initial bid vector B_0 is chosen. Let B_t be the bid after t iterations. B_{t+1} is obtained by sequentially improving bids for each of the N items in turn. While considering the improvement of the bid for an item, we keep the bids for other items constant in B_t. We call this $N - step$ perturbation an $N - sequential$ improvement. The process is stopped when $||B_{t+1} - B_t|| < \varepsilon$ where ε is a positive constant defined by the user and $|| \cdot ||$ is any vectorial norm.

To improve the bid for item i, we replace the bid for the i^{th} item with the optimum bid for that item holding the bids for other items constant. More formally, we use $\beta_i(B_{-i}) = \underset{b_i \in [\underline{p_i}, \overline{p_i}]}{\operatorname{argmax}} \bar{\alpha}(b_i \vee B_{-i})$. The following presents the pseudo-code to approximate β_i, where K is the number of price samples generated from the closing price distributions for all the items.

$\beta \leftarrow 0$
for $k = 1..K$ **do**
  $P \leftarrow \text{generatePriceSamples}(F_1, \ldots, F_N)$
  $\beta \leftarrow \beta + \left(\vartheta\left(\mathcal{I}_{ac}(\overline{p_i} \vee B_{-i}, P)\right) - \vartheta\left(\mathcal{I}_{ac}(\underline{p_i} \vee B_{-i}, P)\right)\right)$
end for
return $\dfrac{\beta}{K}$

We present three variants of MDBI:

Random Start Bid Improver (RSMDBI):. RSMDBI starts with a randomly chosen bid vector and does not use restarts.

Random Start Bid Improver With Restart (RSMDBIWRn):. RSMDBIWRn restarts the hill-climbing process with random bid vectors $n - 1$ times and outputs the bid vector with the highest expected utility.

Valuation Start Bid Improver (VSMDBI):. VSMDBI starts with the bid $B_{\vartheta^1} = (v_1, \ldots, v_N)$, where $v_i = \vartheta(\{i\})$, and does not use restarts[1].

The complexity of the approximation of $\beta_i(B_{-i})$ is linear given K the number of samples. In MDBI, each N-sequential improvement requires N approximations of $\beta_i(B_{-i})$. The number of N-sequential improvement, $C(\varepsilon, N)$, corresponds to the number of iterations of the improvement loop. Thus, the complexity of MDBI is $O(K\,N\,C(\varepsilon, N))$. Experimental results presented in Section 4 shows that the complexity of MDBI is satisfactory.

3.4 Brute force algorithm (BF)

In this section, we discuss an exhaustive search-based bidding algorithm: we assume that the bidder has a finite set of bids it can place for every item[2]. The brute force algorithm (BF) estimates the expected utility of bidding for every possible bid

[1] Restarts are used to avoid local maxima.
[2] If prices are discrete, the set of bids can be the set of prices.

combination. The bid vector with the best expected utility is used to bid in all the auctions. Computing one expected utility requires an exponential number of computation. However, this value can be approximated by price sampling. In that case, BF is asymptotically optimal. When the number of samples tends to infinity, the error in the estimation of expected utility tends to zero.

BF is asymptotically optimal, but its complexity is exponential. Indeed, the expected utility generated by each possible bundle is calculated.

4. EXPERIMENTAL RESULTS

4.1 Experimental settings

Our experimentation goal is to compare the efficiency of different variants of the MDBI scheme with variants of MU bidding both in terms of the quality of the solution generated and time efficiency.

We ran our experiments in an environment containing four single-item single-unit auctions. All closing prices are drawn from discrete closing price distributions. A simulation consists of one bidder with the knowledge of all closing price distributions. This bidder can place one bid in all four auctions at each iteration. At the end of each iteration, the bidder knows which items it won and the payment it has to make for those items. At the end of each simulation, the average profit of the bidder is calculated. A run of our experiment consists of seven simulations, one for each of the bidders RSMDBI, RSMDBIWRn with n=5 or 10, VSMDBI, BF, $\overline{MU}$, EVMU. For a run, bidders in each simulation share the same valuation function. We generate four kinds of valuation functions: (a) SI where items are substitutable, (b) CI where items are complementary, (c) NRI where items are non-related, and (d) RI where valuations for bundles are random. Valuation for single-item bundle is drawn from the range [0, 100], i.e., $\vartheta(\{i\}) \in [0, 100]\ \forall i \in \mathcal{I}$. In each run, the same closing price distributions are used for each simulation. We have used eight predefined closing distributions: four of them produce price ranges from 10 to 90 and four of them from 60 to 140 with increment of 10. For each range, one distribution is uniform (UP), one outputs higher prices with higher probability (HP), one outputs lower prices with higher probability (LP), and one outputs price in the middle of the range with higher probability (MP). For each run, distributions for the four auctions are chosen randomly.

4.2 Quality of solutions generated by algorithms

For performance comparison, we present the cumulative average profit made by the bidders using each algorithm in Table I. The algorithms are ordered by the profits they generate. Algorithms with no statistical performance difference are grouped together. To obtain such groupings, we consider the average profits output by our algorithms as random variable and use the Wilcoxon test to identify the significance of the difference in the performances.

We highlight the following observations from Table I:

(1) RSMDBI performs similar to or better than previously known algorithms.

(2) VSMDBI performs similar to RSMDBI except for RI valuations where performances are worse.

(3) With few random restarts our algorithm always performs similar to BF which is asymptotically optimal.

(4) RSMDBIWRn, $n = 5, 10$ has better performances than RSMDBI except for non-related items where performances are equal.

(5) Every algorithm is optimal when items are non-related except EVMU,

(6) EVMU performs better than $\overline{MU}$ except when items are non-related when its performance is worse.

Remark 1 above shows that RSMDBI is preferable to $\overline{MU}$ and EVMU. In fact, RSMDBI has performances equal to or better than MU variants. Remark 2 shows that choosing B_{ϑ^1} as a starting point may be a good heuristic for particular situations but is not a good idea in general. Remark 3 shows that with reasonable number of restarts (5 in our experiments) the optimal bid vector is always found. Since RSMDBI is, in general, not optimal (Remark 4), we can say that the bid domain has, in general, local maxima. However, since few restarts permit to find optimal bids, the number of local maxima is not very high. Consequently, RSMDBIWRn with reasonably small values of n can be considered to be an approximately optimal method.

Remark 5 shows that almost every algorithm is optimal for NRI valuations. Though we know the optimal bid vector in this case, this result confirms previous analysis. The $\overline{MU}$ optimality can be explained by the fact that bidding for non-related items can be done independently from one another. Regarding the different variants of our algorithms, previous analysis showed that optimal bids are found without restarts from any initial bid vector.

4.3 Time efficiency of MDBI variants

We discuss the complexity of MDBI in Section 3.3. The expression of this complexity contains an unknown function $C(\varepsilon, N)$. We wanted to find out the complexity of MDBI variants in practice. We run experiments using the RSMDBI and VSMDBI algorithms as described in Section 4.1 but by varying the number of items. We did not include RSMDBIWRn since the time needed to output the solution is $n\,T$ where T is the time needed by RSMDBI to terminate.

The time complexity of VSMDBI and RSMDBI appears to be linear in N. Since the complexity of MDBI is equal to $O(K\,N\,C(\varepsilon, N))$, $C(\varepsilon, N)$ is constant for VSMDBI given N. We also collected the average value of $C(\varepsilon, N)$ from our experiments and can highlight the following observations:

(1) Except for non-related items, the number of N-sequential improvements ($C(\varepsilon, N)$) increases very slowly. The largest range is [2, 4].

(2) The number of N-sequential improvements is always better for VSMDBI.

(3) $C(\varepsilon, N) = 1$ for VSMDBI and $C(\varepsilon, N) = 2$ for RSMDBI when items are non-related.

Remark 3 is explained by the fact that VSMDBI realizes that the initial bid $B_0 = B_{\vartheta^1}$ cannot be improved locally. RSMDBI will improve its initial bid vector and reach the optimal one $B^* = B_{\vartheta^1}$ in one N-sequential improvements.

Strategy	Score
RSMDBIWR10	1940.29
BF	1909.42
RSMDBIWR5	1907.25
VSMDBI	1859.65
RSMDBI	1856.85
EVMU	1708.18
$\overline{MU}$	1600.59

(a) SI

Strategy	Score
BF	7693.41
RSMDBIWR5	7693.32
RSMDBIWR10	7674.74
VSMDBI	7513.83
RSMDBI	7434.81
EVMU	7413.88
$\overline{MU}$	7202.51

(b) CI

Strategy	Score
RSMDBI	3748.42
VSMDBI	3731.57
RSMDBIWR10	3725.22
RSMDBIWR5	3724.76
$\overline{MU}$	3715.95
BF	3658.38
EVMU	3522.67

(c) NRI

Strategy	Score
RSMDBIWR10	6898.97
BF	6871.01
RSMDBIWR5	6828.67
RSMDBI	6239.84
EVMU	6224.37
VSMDBI	5946.33
$\overline{MU}$	5184.76

(d) RI

Table I. Cumulative profits of different algorithms.

5. CONCLUSION

The goal of this paper was to evaluate the relative performance of some existing algorithms based on marginal utility calculations and variants of a more recent algorithm on mulitidimensional bid improvement, MDBI, on a range of scenarios involving bidding for bundles of items in simultaneous auctions. The scenarios investigated include substitutable, complementary, non-related and random valuations. We observe that the MDBI algorithm performs optimally with only a few random restarts. In particular, it outperforms the marginal utility based algorithms for substitutable and complementary valuations. For the MDBI algorithm, in most cases it does not matter whether the initial bid vector is chosen randomly or based on valuation of individual items.

The experiments in this paper were conducted on a small set of items. Part of the reason for doing this is the exponential computational complexity of expected utility based bidding schemes. The MDBI algorithm, on the other hand has only linear complexity, a great advantage for its use in larger problems.

ACKNOWLEDGMENTS

This work has been supported in part by an NSF award IIS-0209208.

REFERENCES

Boutilier, C., Goldszmidt, M., and Sabata, B. 1999. Sequential auctions for the allocation of resources with complementarities. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence*. 527–534.

CANDALE, T. 2005. Hill-climbing approach to bidding for bundles in simultaneous auctions. M.S. thesis, University of Tulsa.

GREENWALD, A. AND BOYAN, J. 2004. Bidding under uncertainty: theory and experiments. In *AUAI '04: Proceedings of the 20th conference on Uncertainty in artificial intelligence*. AUAI Press, 209–216.

GREENWALD, A. AND BOYAN, J. 2005. Bidding algorithms for simultaneous auctions: A case study. *Journal of Autonomous Agents and Multiagent Systems 10,* 1, 67–89.

PARKES, D. C. 2001. Iterative combinatorial auctions: Achieving economic and computational efficiency. Ph.D. thesis, University of Pennsylvania.

SANDHOLM, T. 2000. emediator: a next generation electronic commerce server. In *AGENTS '00: Proceedings of the fourth international conference on Autonomous agents*. ACM Press, 341–348.

SEN, S., CANDALE, T., AND BASAK, S. 2005. Profit sharing auction. In *Proceedings of the Twentieth International Conference on Artificial Intelligence (AAAI-2005)*.

STONE, P. AND GREENWALD, A. 2003. The first international trading agent competition: Autonomous bidding agents. *Journal of Electronic Commerce Research*.

STONE, P., SCHAPIRE, R. E., LITTMAN, M. L., CSIRIK, J. A., AND MCALLESTER, D. 2003. Decision-theoretic bidding based on learned density models in simultaneous, interacting auctions. *Journal of Artificial Intelligence Research 19*, 209–242.

WURMAN, P. R., WELLMAN, M. P., AND WALSH, W. E. 1998. The Michigan Internet AuctionBot: A configurable auction server for human and software agents. In *Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98)*, K. P. Sycara and M. Wooldridge, Eds. ACM Press, New York, 301–308.

Received Janary 2006; accepted January 2006