Multi-Dimensional Bid Improvement Algorithm for Simultaneous Auctions

Teddy Candale and **Sandip Sen**
University of Tulsa, USA
{tdi,sandip}@utulsa.edu

Abstract

Bidding for multi-items in simultaneous auctions raises challenging problems. In multi-auction settings, the determination of optimal bids by potential buyers requires combinatorial calculations. While an optimal bidding strategy is known when bidding in sequential auctions, only suboptimal strategies are available when bidding for items being sold in simultaneous auctions. We investigate a multi-dimensional bid improvement scheme, motivated by optimization techniques, to derive optimal bids for item bundles in simultaneous auctions. Given a vector of initial bids, the proposed scheme systematically improves bids for each item. Such multi-dimensional improvements result in locally optimal bid vectors. Globally optimal bid vectors are guaranteed in the limit for infinite restarts. For ease of presentation we use two-item scenarios to explain the working of the algorithm. Experimental results show polynomial complexity of variants of this algorithm under different types of bidder valuations for item bundles.

1 Introduction

Auction theory has received significant attention from agent researchers following the development of electronic auctions on the Internet. Numerous sites like eBay, uBid offer different types of auctions to facilitate the trading of goods. Also sites like eMediator implements a large variety of combinatorial auctions and exchanges [Sandholm, 2000]. Researchers are interested both in designing auctions with desirable properties [Parkes, 2001; Sen *et al.*, 2005] and developing automated agents to represent interests of human users [Greenwald and Stone, 2001; Greenwald and Boyan, 2004; 2005; Stone *et al.*, 2003; Stone and Greenwald, 2005].

The problem of computing optimal bids is 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. Two settings are generally studied: *combinatorial auction* and *multi-auction*. A combinatorial auction offers multiple items. Bidders can submit the amount they are willing to pay to get bundles. In strategy-proof combinatorial auction settings, a rational bidder bids according to its true preferences and the auctioneer incurs the computational cost of optimal bundle allocation based on bids submitted. The winner determination problem for combinatorial auction is NP-Hard. This complexity may discourage certain sellers. Besides, an individual may not find all desired items in the same combinatorial auction.

In a multi-auction setting, multiple single-item auctions are held concurrently or sequentially. A potential bidder needs to estimate closing prices of such auctions to compute optimal bids. In contrast to combinatorial auctions, in multi-auction settings, a bidder incurs the computational cost of calculating optimal bids given its valuation for items and the expected closing prices. As in a single-auction setting, auctioneer in each of these multiple, independent auctions have a straightforward computational task of selecting a winner from bids submitted to a single auction.

In multi-auction settings, auctions can be *sequential* or *simultaneous*. Sequential auctions close at a predetermined known order. A bidder can bid in the auction that will close next and then wait for the outcome before computing its next bid. Optimal bidding policies have been developed for sequential auctions [Stone *et al.*, 2003; Greenwald and Boyan, 2004; 2005]. Simultaneous auctions, however, run concurrently and bidders do not know in advance their precise closing times. Bids to be submitted to all auctions have to be computed at the same time. Developing efficient optimal policies for simultaneous auctions is an open research problem.

We propose a novel bidding algorithm for simultaneous auctions motivated by optimization techniques. We sequentially improve components of the current bid vector to maximize the bidder's expected utility given the closing price distributions of the auctions. We reason with continuous closing price distributions which allows us to address a more general setting and enables the application of powerful methods tailored for continuous spaces. Our technique can be adapted to find optimal bids given discrete closing price distributions. We graphically illustrate the working of our bid improvement scheme. We then experimentally evaluate variants of our algorithm for different problem sizes and under substitutable, complementary, and unrelated item preferences. Our main theoretical result is that our proposed bidding scheme is optimal given infinite number of random restarts. Experimental results demonstrate that effective approximations to optimal results can be produced in polynomial time.

2 Related work

Combinatorial auctions can be designed to provide desirable social outcomes [Parkes, 2001; Sandholm, 2002]. However, it is not feasible to have a combinational auction which offers all items an individual may desire. This may be due to new needs that no seller has foreseen. Besides, items which are not directly related are unlikely to be found in the same auction. For example, a car company which wants to design a car with an on-board computer is unlikely to find pieces needed to build the engine in the auction it can find components to build the on-board computer.

Stone, Greenwald, and fellow researchers investigated the bundle bidding problem for multi-auctions in the context of the Trading Agent Competition (TAC) [Greenwald and Boyan, 2004; 2003; Stone *et al.*, 2003; Stone and Greenwald, 2005]. They assign valuations to individual items that correspond to expected marginal utilities of the items. The marginal utility of an item i is the extra-profit generated by the acquisition of i at zero price, and corresponds to the amount one is willing to pay for an item calculated as the extra benefit of getting it. Stone *et al.* approximated it by price sampling for sequential auctions since the exact calculation is of exponential complexity [Stone *et al.*, 2003].

Though Stone *et al.* presented the marginal utility based bidding scheme, $\overline{MU}$, in the context of sequential auctions, they used it for simultaneous auctions [Stone *et al.*, 2003]. Greenwald and Boyan [Greenwald and Boyan, 2004] showed that $\overline{MU}$ is optimal for sequential auctions but can be suboptimal for simultaneous auctions. Greenwald and Boyan proposed a variant of $\overline{MU}$, expected marginal utility bidding (EVMU), for bidding in simultaneous auctions where the bidder first computes an optimal set of items to bid for. It is the bundle with the best profit assuming actual closing prices in future auctions are equal to the expected closing prices. After determining the optimal bundle, the bidder bids expected marginal utility for each item in the bundle. Greenwald and Boyan proved that this method is optimal when prices are deterministic [Greenwald and Boyan, 2004]. This method, however, is still suboptimal in general. A downside of each of these methods is their exponential computational complexity as calculating the marginal utility of one item required the knowledge of the expected profit generated by the acquisition of each possible bundle.

Byde *et al.* [Byde *et al.*, 2002] present a heuristic approach to bidding in sequential and simultaneous auctions. Their approach does not have any theoretical guarantees of expected utility and cannot be implemented as presented in the paper as necessary heuristics are not included.

3 Multi-auction Model

We consider multiple sealed-bid auctions offering items from the set $\mathcal{I}$. A valuation function ϑ expresses the bidder's preferences for bundles or subsets of items from the set $\mathcal{I}$, i.e., the bidder is willing to pay up to $\vartheta(I)$ for a bundle of items $I \subseteq \mathcal{I}$. Each item i is available only in the single-item single-unit auction a_i. We do not specify the particular auction type but make the *exogenous price* assumption: the bids of our bidder do not influence the auction closing prices. An auction is modeled by the probability distribution F_i of the closing prices of the item being offered in that particular auction. We assume these distributions to be continuous, independent, and known by the bidder [Greenwald and Boyan, 2004; 2003; Stone *et al.*, 2003; Stone and Greenwald, 2005]. In practice approximate price distributions can be learned from observing electronic markets. When an auction closes, a closing price $p_i \in [\underline{p_i}, \overline{p_i}]$ is drawn from the distribution F_i. The bidder gets the item if it has placed a bid b_i greater than or equal to the closing price, i.e., if $p_i \leq b_i$, and the winning payment is equal to the closing price p_i. All auctions run in parallel and their closing times are not known by the bidder. The bidder place bids represented by $B = (b_1, \ldots, b_N) \in \mathcal{B}$ where $\mathcal{B}$ is the bid domain for all auctions. Replacing a bid is not allowed in this model.

Once all the auctions close, the bidder can compute its utility $\alpha(B, P)$ where $P = (p_1, \ldots, p_N)$ represents the closing prices of all auctions. The set of acquired items $\mathcal{I}_{ac}(B, P)$ is calculated as

$$\mathcal{I}_{ac}(B, P) = \{i \in \mathcal{I} \; s.t. \; p_i \leq b_i\}$$

and the corresponding utility received by the bidder is

$$\alpha(B, P) = \vartheta(\mathcal{I}_{ac}(B, P)) - \sum_{i \in \mathcal{I}_{ac}(B, P)} p_i.$$

The expected utility is then $\bar{\alpha}(B) = E_P[\alpha(B, P)]$ which can be calculated as specified in Proposition 1:

Proposition 1 (Expected utility)

$$\bar{\alpha}(B) = \sum_{I \subseteq \mathcal{I}} \left\{ \left(\prod_{i \in I} F_i(b_i) \right) \left(\prod_{j \notin I} (1 - F_j(b_j)) \right) \vartheta(I) \right\} - \sum_{i=1}^{N} \int_{\underline{p_i}}^{b_i} p_i \, f_i(p_i) \, dp_i,$$

where $F_i(b_i) = Pr\{p_i \leq b_i\} = \int_{\underline{p_i}}^{b_i} f_i(p_i) \, \mathrm{d}p_i$ and f_i is the pdf of F_i. Our research objective is to find a bid vector B^* which maximizes the expected utility $\bar{\alpha}$: $B^* = \underset{B \in \mathcal{B}}{\operatorname{argmax}} \, \bar{\alpha}(B)$.

4 Multi-Dimensional Bid Improvement

Our proposal uses an optimization technique to find optimal bids for sequential auctions. Assume that the bidder has decided by some means to bid the vector B. Is it possible to do better than bidding B, i.e., is there a bid vector B' such that $\bar{\alpha}(B') > \bar{\alpha}(B)$? Assuming B is sub-optimal, there is at least one item whose bid can be improved, i.e., there exist i and δ_i such that $\bar{\alpha}(B) < \bar{\alpha}((b_i + \delta_i) \vee B_{-i})$ where $B_{-i} = (b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_N)$ and $b'_i \vee B_{-i} = (b_1, \ldots, b_{i-1}, b'_i, b_{i+1}, \ldots, b_N)$. By repeating this process, we can realize the best improvement possible for the item i, which is equivalent to maximizing the function $b_i \mapsto \bar{\alpha}(b_i \vee B_{-i})$. The following formalizes the idea.

Definition 1 (Optimal bid for item i**)** *$\beta_i(B_{-i})$ is the optimal bid for item i given bids for item $j \neq i$ is fixed: $\beta_i(B_{-i}) = \underset{b_i \in [\underline{p_i}, \overline{p_i}]}{argmax} \; \bar{\alpha}(b_i \vee B_{-i})$.*

Proposition 2 (Optimal bid for item i)

$$\beta_i(B_{-i}) = \sum_{\substack{I \subseteq \mathcal{I} \\ i \in I}} \prod_{\substack{j \in I \\ j \neq i}} F_j(b_j) \prod_{l \notin I} (1 - F_l(b_l))\vartheta(I) - \sum_{\substack{J \subseteq \mathcal{I} \\ i \notin J}} \prod_{j \in J} F_j(b_j) \prod_{\substack{l \notin J \\ l \neq i}} (1 - F_l(b_l))\vartheta(J)$$

Sketch of proof

Space constraints preclude the presentation of the full proof. We note, however, that $\beta_i(B_{-i})$ is such that $\frac{\partial \bar{\alpha}}{\partial b_i}(\beta_i(B_{-i}) \vee B_{-i}) = 0$. Besides, let consider $\bar{\alpha}_i : b_i \mapsto \bar{\alpha}(b_i \vee B_{-i})$. We can show that

$$\begin{cases} 0 < \bar{\alpha}_i'(b_i) & \text{if } \beta_i(B_{-i}) > b_i \\ 0 = \bar{\alpha}_i'(b_i) & \text{if } \beta_i(B_{-i}) = b_i \\ 0 > \bar{\alpha}_i'(b_i) & \text{if } \beta_i(B_{-i}) < b_i \end{cases}$$

$$\text{where } \bar{\alpha}_i' = \frac{\mathrm{d}\,\bar{\alpha}_i}{\mathrm{d}\,b_i} = \frac{\partial \bar{\alpha}}{\partial b_i}$$

Consequently,

- $\bar{\alpha}_i$ is increasing when $\beta_i(B_{-i}) > b_i$.
- $\bar{\alpha}_i$ is decreasing when $\beta_i(B_{-i}) < b_i$.
- $b_i = \beta_i(B_{-i})$ is the unique local maximum, and thus the unique global maximum.

□

4.1 Best bid hyper-surfaces

The equation $b_i = \beta(B_{-i})$ defines a hyper-surface in the N-dimension space $\mathcal{B}$[1]. This hyper-surface divides $\mathcal{B}$ into two zones: one zone where, given a point B, b_i has to be increased to increase $\bar{\alpha}$, and the other zone where b_i has to be decreased to improve $\bar{\alpha}$. Hence, the i^{th} component of the gradient is always directed to the hyper-surface $b_i = \beta(B_{-i})$ and is equal to zero on the surface. An intersection of the N hyper-surfaces, one for each item, defines a bid vector whose gradient is equal to zero (when this intersection is inside the bid domain). This point, though a potential candidate, is not necessarily a local maximum since a gradient equal to zero does not guarantee the existence of a local optimum. Optimal bids can be found at the intersection of the N hyper-surfaces either inside the bid domain or at the boundary of the bid domain. The optimum can then be discovered by sequentially moving from one hyper-surface to another until no further improvements can be made.

We now outline our bid selection scheme. An initial bid vector B is chosen. Then, we repeatedly improve the N components of the bid vector B in any predetermined order. Improving the bid for item i involves replacing b_i by $\beta_i(B_{-i})$: $B \leftarrow \beta_i(B_{-i}) \vee B_{-i}$. We will refer to this improvement as *single improvement* and the sequence of N improvements as *N-sequential improvement*. In the bid domain $\mathcal{B}$, the single improvement can be regarded as going from B to the hyper-surface $b_i = \beta(B_{-i})$ by moving parallel to the b_i-axes. The process is stopped when no further improvement can be made. We refer to this sequential bid-improvement process as a Multi-Dimensional Bid Improvement (MDBI) scheme.

[1] $b_i = \beta(B_{-i})$ is a curve when $N = 2$, a surface when $N = 3$.

Figures 1, 2, and 3 provide a visual description of the above discussion. Each figure represents the bid domain $\mathcal{B}$ where $N = 2$ and hence we draw the two hyper-surfaces. We chose uniform probability distributions to represent the closing price distributions as a result of which the hyper-surfaces are lines. Arrows give the direction to follow to improve the expected utility in the zones delimited by the hyper-surfaces. Paths leading to local maxima are drawn as dashed lines with arrows representing the direction to follow.

Figures 1, 2, and 3 allow us to present some conclusions. For all types of item valuations, it is preferable to bid the individual valuation of an item when bidding only for this item. For substitutable items, β_i is decreasing and the maximum value taken by β_i is $\beta_i(\underline{p_i}) = v_i$ where $v_i = \vartheta(\{i\})$, and the bidder should always bid less than its individual valuation for each item when bidding for a bundle. On the contrary, for complementary items, β_i is increasing and the minimum value taken by β_i is $\beta_i(\underline{p_i}) = v_i$, and the bid for an item should always be more than the individual valuation for that item when bidding for an item bundle.

In the examples presented in these figures, we have unique optima. Given the nature of the closing price distributions, however, the N hyper-surfaces may have several intersections resulting in as many local optima. Consequently, we can only guarantee reaching a local maxima using the *N-sequential* MDBI process. This problem can be circumvented by restarting the improvement process, e.g., by choosing a random bid vector, when no further improvement is possible. Assuming an infinite number of restarts, this method will discover the optimal bid vector. As this is not feasible in practice, only a finite number of restarts can be considered.

We observe that for non-related items, and irrespective of the number of items, there exists a unique optimum. Indeed, in that case, $\beta_i(b_i) = v_i$. Consequently, the N hyper-surfaces are orthogonal (see Figure 3) and therefore have a unique intersection $\hat{B} = (v_1, \ldots, v_N)$ which corresponds to B^*. Starting from any bid vector, we will arrive at this solution using the MDBI scheme. As a result, the MDBI process will output the optimal bid vector irrespective of of initial bids when items are non-related.

4.2 Bidding Algorithm

We now describe the steps of our bid improvement algorithm in more detail. An initial bid vector B_0 is chosen. Let B_t be the bid after t iterations. B_{t+1} is obtained by improving sequentially bids for each of the N items in B_t, keeping the bids for other items constant. The process is stopped when $||B_{t+1} - B_t|| < \varepsilon$ where ε is a user-chosen positive constant and $||\cdot||$ is any vectorial norm. To improve a bid, we need to calculate β_i. Though the formula presented in Proposition 2 requires exponential computation, it is possible to approximate β_i in polynomial time via price sampling. Algorithm 1

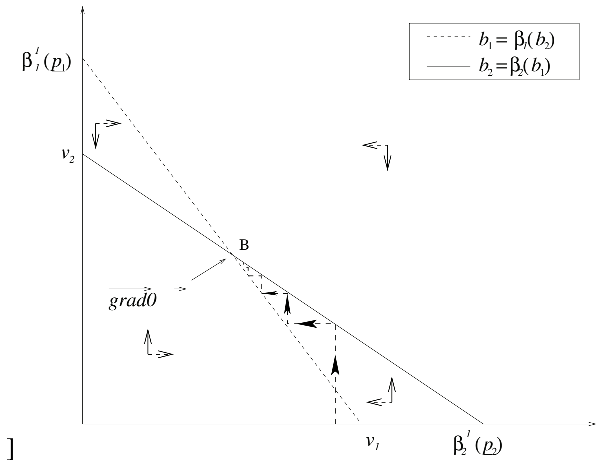

Figure 1: Hyper-surfaces when $N = 2$, closing price distributions are uniform, and items are substitutable.

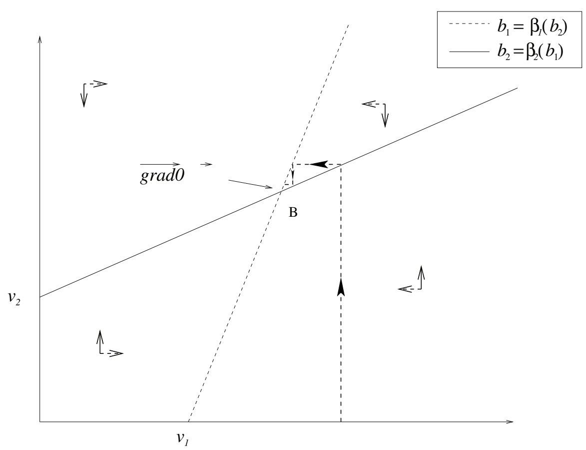

Figure 2: Hyper-surfaces when $N = 2$, closing price distributions are uniform, and items are complementary.

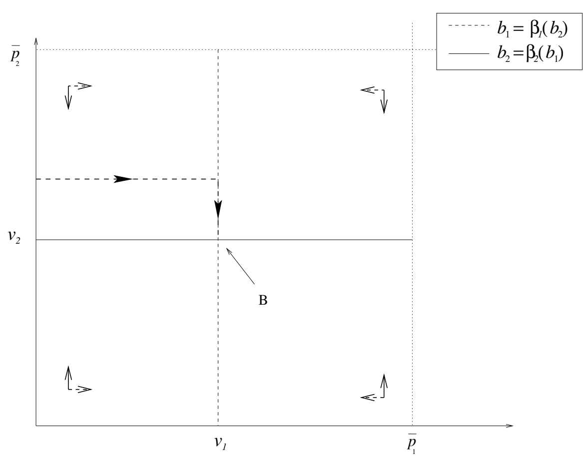

Figure 3: Hyper-surfaces when $N = 2$, closing price distributions are uniform, and items are non-related.

presents the pseudo-code to approximate β_i. Algorithm 2 presents the pseudo-code for the MDBI algorithm.

Algorithm 1 Approximation of $\beta_i(B_{-i})$ by price sampling

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

Algorithm 2 MDBI

$B \leftarrow initializeBids()$
// Improvement loop.
repeat
  $B' = B$
  for $i = 1..N$ **do**
    $b_i \leftarrow \beta_{approx}(i,\, B_{-i},\, F,\, \vartheta,\, K)$
  end for
until $||B - B'|| < \varepsilon$
return B

The computational complexity of approximating $\beta_i(B_{-i})$, $\beta_{approx}(i,\, B_{-i},\, F,\, \vartheta,\, K)$, is linear given the number of samples, K. More precisely, it is equal to $\theta(K)$ additions, $\theta(K)$ valuation calculations and $\theta(K)$ price sample generations. In MDBI, each N-sequential improvement requires N approximations of $\beta_i(B_{-i})$. The number of N-sequential improvement depends on the number of iterations of the improvement loop, $C(\varepsilon,\, N)$. We have $(C(\varepsilon,\, N) * N)$ N-sequential improvement and $C(\varepsilon,\, N)$ calculations of norms. Generally, vectorial norms can be computed in linear time given the dimension of the space (N in our case). Thus, the complexity of MDBI is $O(K\,N\,C(\varepsilon,\, N))$ additions, $O(K\,N\,C(\varepsilon,\, N))$ valuation calculations and $O(K\,N\,C(\varepsilon,\, N))$ price sample generations. The value of $C(\varepsilon,\, N)$ is difficult to judge *a priori* but experimental results presented in the next section show that $C(\varepsilon,\, N)$ increases only very gradually with N and thus the computational complexity of MDBI is approximately linear. This, in particular, is a significant improvement over current schemes [Greenwald and Boyan, 2004; Stone *et al.*, 2003] which have exponential complexity.

5 Experimental results

We ran experiments with the following objectives: to show that our algorithm performs well when probability distributions are discrete[2], both in terms of the quality of the solution provided and the time efficiency[3]. We compare variants of

[2]Our study was made assuming continuous distributions. In real-life environments, agents are likely to use discrete probability distributions.

[3]Though we provide formal analysis of the complexity of our algorithm, the expression of MDBI complexity contains a function ($C(\varepsilon,\, N)$) whose value is unknown. The goal of our experiments is to see whether $C(\varepsilon,\, N)$ scales minimally with N.

our algorithms with an exhaustive search or brute force algorithm (BF) that provides optimal results at high search costs. The BF approach consists of calculating the expected profit generated by each possible bid combinations and then bidding the best combination. BF is optimal, but its complexity is exponential. We now present three variants of our MDBI algorithm that we have experimented with:

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 starts with a randomly chosen bid vector and restarts the hill-climbing process $n-1$ times and outputs the bid vector with the highest expected utility.

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

5.1 Experimental settings

In the first set of experiments, we used the above-mentioned variants of the MDBI algorithm in an environment containing four single-item single-unit auctions and the focus is on evaluating their success in finding optimal solutions, as checked by the BF scheme, under different valuation functions. Subsequently we present results with larger number of items where the focus is on studying the time efficiency and scale-up properties of the algorithms. 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 four simulations, one for each of the bidders RSMDBI, RSMDBIWRn with n=5 or 10, VSMDBI, BF. 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) RVI, where valuations for bundles are random. Considering three set of items I, J, K such that $J \cup K = I$ and $J \cap K = \emptyset$. Items are substitutable if $\vartheta(I) \leq \vartheta(J) + \vartheta(K)$ for all such item sets. Items are complementary if $\vartheta(I) \geq \vartheta(J)+\vartheta(K)$ for all such item sets. Items are non-related if $\vartheta(I) = \vartheta(J) + \vartheta(K)$ for all such item sets. If no one of the above conditions are satisfied for all item sets, we consider the items to have random valuation. 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 auctions are repeated in each simulation, i.e., the same closing price distributions are utilized for each simulation. We have eight predefined probability distributions. Four of them produce price ranges from 10 to 90 and four of them from 60 to 140 with discrete values in increments of 10. For each range, one distribution is uniform (UP), one outputs the highest prices with higher probability (HP), one outputs the lowest 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.

5.2 Quality of solutions provided by algorithms

To compare our algorithms, we display the cumulative average profit generated by the bidders using each strategy in Table 1. Algorithms are ordered by the net utility obtained. Horizontal lines in the table groups the algorithms into blocks. Performance of algorithms in the same block are statistically indistinguishable according to the Wilcoxon test.

We highlight the following observations:

1. VSMDBI performs similar to RSMDBI except for RVI valuations where performances are worse.
2. With a small number of random restarts our algorithm always has performance similar to BF which is asymptotically optimal.
3. RSMDBIWRn with $n = 5, 10$ has better performances than RSMDBI except for non-related items where performances are equal. As we have observed before, restarts are unnecessary for non-related items.
4. Every algorithm is optimal when items are non-related.

Remark 1 shows that choosing B_{ϑ^1} as a starting point is a good heuristic for some but not all cases. Remark 2 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 3), we can say that the bid domain has, in general, some local maxima and/or ridges. However, since few restarts permit to find optimal bids, the number of local maxima and ridges is not very high. Consequently, RSMDBIWRn with reasonably small values of n can be considered as an approximately optimal method. From Remark 4, we find that almost every algorithm is optimal for NRI. Though this result is not that significant as we know the optimal bid vector in that case, it confirms our previous analysis which claimed that the optimal bids are found by the MDBI algorithm after the first N-sequential improvement irrespective of the initial bid vector.

5.3 Time efficiency and scale up properties

To evaluate the run time of our algorithm for larger problems, we run experiments as described in the last section but by changing the number of items from 4 to 15 to evaluate the values of $C(\varepsilon, N)$. Those experiments were similar to the previous ones except that we output the values of $C(\varepsilon, N)$ and average them. We highlight the following observations:

1. Except for NRI, 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.

We can conclude that $C(\varepsilon, N)$ is almost constant. Consequently, the computational complexity of our algorithms is almost linear given the number of items and hence has desirable time efficiency and scale up properties. The third remark

Strategy	Score
RSMDBIWR10	1940.2
BF	1909.4
RSMDBIWR5	1907.2
VSMDBI	1859.6
RSMDBI	1856.8

(a) SI

Strategy	Score
BF	7693.4
RSMDBIWR5	7693.3
RSMDBIWR10	7674.7
VSMDBI	7513.8
RSMDBI	7434.8

(b) CI

Strategy	Score
RSMDBI	3748.4
VSMDBI	3731.5
RSMDBIWR10	3725.2
RSMDBIWR5	3724.7
BF	3658.3

(c) NRI

Strategy	Score
RSMDBIWR10	6898.9
BF	6871.0
RSMDBIWR5	6828.6
RSMDBI	6239.8
VSMDBI	5946.3

(d) RVI

Table 1: Profits from MDBI variants with different valuation functions.

can be explained by the fact that VSMDBI will try to improve the initial bid $B_0 = B_{\vartheta^1}$ and will realize that there is no better bid vector. RSMDBI will improve its initial bid vector and reach the optimal one $B^* = B_{\vartheta^1}$ in one N-sequential improvements. Then, it behaves like VSMDBI.

In a companion paper [Candale and Sen, 2006] we show that MDBI produces equal or better solutions in much less time compared to MU and EVMU algorithms for larger number of items than discussed here. We cannot verify the optimality of those results as it was computationally prohibitive to run the BF algorithm for those larger problem sizes. The experimental results clearly demonstrate the effectiveness and scalability of the MDBI schemes.

6 Conclusions

We presented a new method for bidding in simultaneous auctions which is asymptotically optimal. We used a novel optimization based technique that improves any arbitrary bid vector. While we developed the bidding approach assuming continuous closing price probability distributions, we showed that our technique is applicable when closing prices are discrete. The MDBI algorithm performs optimally with few restarts and also scales up well with the number of items.

The MDBI algorithm produces optimal bid vectors with infinite random restarts. As this is infeasible in practice, we evaluate variants of the scheme with finite number of restarts and carefully chosen starting bid vectors. A surprising result is that good solutions can be achieved without restarts when items are substitutable, complementary, or are non-related. In general, small number of restarts can be used to approximate optimal expected utility in all cases.

A desirable property of our algorithm, in contrast to existing schemes, is its approximately linear time complexity. Experimental results show that the MDBI scheme scales up effectively with larger number of items.

We plan to study situations where a bidder can bid in both single-item and combinatorial auctions to acquire bundles. We also intend to adapt MDBI for generating bids for multi-unit auctions.

Acknowledgment US National Science Foundation award IIS-0209208 partially supported this work.

References

[Byde *et al.*, 2002] Andrew Byde, Chris Preist, and Nicholas R. Jennings. Decision procedures for multiple auctions. In *Proceedings of the 1st Joint International Conference on Autonomous Agents and Multi-Agent Systems*, pages 613–620. ACM, 2002.

[Candale and Sen, 2006] Teddy Candale and Sandip Sen. A comparison of bidding strategies for simultaneous auctions. *SIGecomm Exchange*, 5(5):41–48, January 2006.

[Greenwald and Boyan, 2003] Amy Greenwald and Justin Boyan. Bidding marginal utility in simultaneous auctions. In *Proceedings of the 18^{th} International Joint Conference on Artificial Intelligence*, pages 1463–1464, August 2003.

[Greenwald and Boyan, 2004] Amy Greenwald and Justin Boyan. Bidding under uncertainty: theory and experiments. In *AUAI '04: Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 209–216. AUAI Press, 2004.

[Greenwald and Boyan, 2005] Amy Greenwald and Justin Boyan. Bidding algorithms for simultaneous auctions: A case study. *Journal of Autonomous Agents and Multiagent Systems*, 10(1):67–89, 2005.

[Greenwald and Stone, 2001] Amy Greenwald and Peter Stone. Autonomous bidding agents in the trading agent competition. *IEEE Internet Computing*, 5(2):52–60, 2001.

[Parkes, 2001] David C. Parkes. *Iterative Combinatorial Auctions: Achieving Economic and Computational Efficiency*. PhD thesis, University of Pennsylvania, May 2001.

[Sandholm, 2000] Tuomas Sandholm. eMediator: a next generation electronic commerce server. In *AGENTS'00: Proceedings of the Fourth International Conference on Autonomous Agents*, pages 341–348. ACM Press, 2000.

[Sandholm, 2002] Tuomas Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135(1):1–54, 2002.

[Sen *et al.*, 2005] Sandip Sen, Teddy Candale, and Susnata Basak. Profit sharing auction. In *Proceedings of the Twentieth International Conference on Artificial Intelligence*, pages 170–175, July 2005.

[Stone and Greenwald, 2005] Peter Stone and Amy Greenwald. The first international trading agent competition: Autonomous bidding agents. *Electronic Commerce Research*, 5(2):229–65, April 2005.

[Stone *et al.*, 2003] P. Stone, R. E. Schapire, M. L. Littman, J. A. Csirik, and D. McAllester. Decision-theoretic bidding based on learned density models in simultaneous, interacting auctions. *Journal of Artificial Intelligence Research*, 19:209–242, 2003.