Optimizing Auctioneer's Revenues in Expanding Multi-Unit Auctions

Onn Shehory
IBM Haifa Research Lab
Haifa University Campus
Haifa 31905, Israel
onn@il.ibm.com

Eran Dror
Technion - Israel Institute of Technology
Haifa 32000, Israel

Categories and Subject Descriptors:
I.2.m [**Artificial Intelligence**]: Miscellaneous.

General Terms: Algorithms, Design, Economics.

Keywords: Expanding auction, Electronic commerce.

1. INTRODUCTION

In this study we provide a strategy that maximizes the expected revenues of the auctioneer in an expanding multi-unit auction. We model the auction process as a state graph in which nodes are auction states and edges are transitions. With this model, finding the optimal strategy is equivalent to solving a search problem on the state graph. We prove that the search problem to be solved, although seemingly exponentially complex, is actually linearly bounded. Based on this result, we introduce an informed strategy that optimizes the auctioneer's revenue.

In an expanding multi-unit auction, one item of a good is initially offered to the bidders. The bidders are free to raise their bid and the auctioneer is free to raise the number of units offered by one each time a new bid is received. The most significant factor affecting the auctioneer's revenue in an expanding multi-unit auction is the decision in which context to raise the number of units offered and in which to preserve it. Naïve auctioneer strategies for increasing the number of items are sub-optimal.

A simple but naive strategy for addressing this issue is the *step function strategy*. Such a strategy is used by, e.g., the auction site Olsale [1]. The strategy is based on a step function set by the auctioneer; the step function is composed of <price level, number of units> pairs that determine the number of units to be offered when the current bid reaches a certain level. Figure 1 depicts this strategy. When the current bid reaches p_1 the auctioneer offers one more unit for sale raising the total number of units offered to 2. If the price exceeds p_2 the auctioneer offers an additional unit raising the number of units offered to 3, and so on.

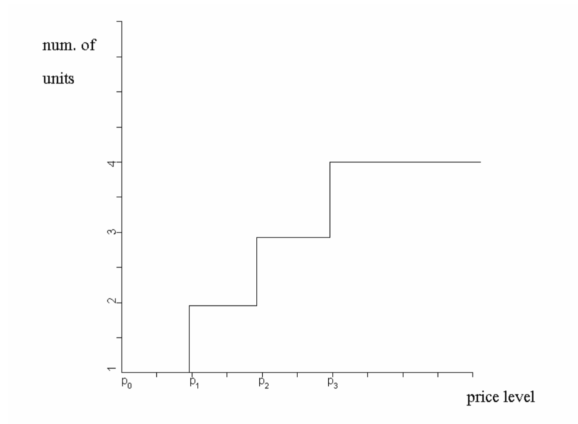

Figure 1. Step function strategy

The step function strategy is simple to implement, however it is not trivial to configure its parameters to maximize the auctioneer's revenues. The same step function may yield higher revenues in one context of execution and lower revenues in others.

2. THE INFORMED STRATEGY

We model the problem using a state graph, the nodes describing the states of the auction and edges describe the transitions between states. Each state S has the following properties:

- The price a bidder has to pay to join the winner list denoted by l_d where d is bid sequence. For instance, l_2 denotes the price a bidder has to pay to join the winner list after 2 bids have already been submitted and accepted by the auctioneer. Note that all states that may be formed after the auctioneer has received d bids share the same price level l_d.
- The number of units the auctioneer is willing to sell at price l_d denoted by n_S.
- A list of maximal bidding values[1] (MBV) of the bidders that current participate in the auction. These bidders may part of the winner list. The list contains bidders, which are currently in the winner list, or bidders that are willing to pay more than l_d and therefore may join the winner list in a later stage of the auction execution. We denote the list by B_S.

[1] The maximal bidding value of each bidder is the maximal bid he/she is willing to offer during the auction, this value may differ from his/her private value depending on the strategy adapted by him/her.

AAMAS'05, July 25-29, 2005, Utrecht, Netherlands.

- A list of MBVs of the bidders in the winner list. We denote this list by B_s^w. Clearly, $B_s^w \subseteq B_s$.

Figure 2 illustrates an auction execution state graph. Each node S_i in the graph consists of its bidder list B_i. The initial number of units offered is n_0 and the reservation price set initially by the auctioneer is l_0. The auctioneer decision to raise the number of units is reflected in the graph by traversing to the right (a *raise* transition). The descendent state created is characterized by an additional unit offered and a price level that is higher by Δ. When the auctioneer wishes to preserve the number of units the graph branches to the left (a *preserve* transition) forming a state that is characterized by the same number of units offered and a price level that is higher by Δ.

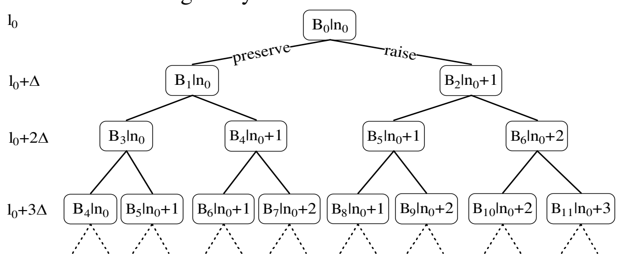

Figure 2. An auction execution state graph

In developing a strategy that is based on the state tree described, we face two problems. Firstly, the MBVs of bidders entering the auction are unknown. Secondly, the search for the most beneficial strategy may be computationally expensive.

The first problem can be addressed by assuming a probability distribution function f_i of maximal bidding value for each bidder b_i. Such distributions can be learned from historical biddings. Using this data, we devise an informed strategy. The strategy is a simulation-based. Given the current state of the auction, it attempts to determine the auctioneer's action that would yield the best results. It does this by generating MBV vectors of bidders participating in an auction and the current state of the auction. To find the auctioneer's best choice, a search problem that takes into account all possible scenarios should be solved for each MBV vector generated. A solution determines whether the auctioneer should raise the number of units sold by one, or preserve it.

Note that since the branching factor of the tree is 2, given the reservation price l_0 and the maximal bidding value of the bidders participating l_{max}, in the worst case the number of bid levels will be produced when only one unit is offered during the lifetime of the auction. Thus, the number of bid levels is bounded by $(l_{max} - l_0)/\Delta$ and the number of states in the state tree is bounded by $O(2^{(l_{\max}-l_0)/\Delta})$. Therefore, computing the complete state tree is too expensive. In an extended version of this paper we prove that the complexity of the search problem can be reduced to linear.

The algorithm below describes the implementation of the informed strategy within the auction execution process. The time allocated by the auctioneer for computing the optimal strategy is denoted by t_p, and the time in which a new bid that raises the price of the good sold is received is denoted by t_0:

1. $t_0 \leftarrow now, nRaise \leftarrow 0, n\Pr eserve \leftarrow 0$
2. Generate an MBV vector v_{sample} based on the probability distribution functions of participating bidders.
3. Using the informed strategy compute the optimal action to be taken based on vector v_{sample}. If raising the number of units sold by one is the best course of action then $nRaise \leftarrow nRaise + 1$, otherwise $n\Pr eserve \leftarrow n\Pr eserve + 1$.
4. If $now < t_0 + t_p$ go to step 2.
5. If $nRaise > n\Pr eserve$ raise the number of units sold by one, if $nRaise < n\Pr eserve$ preserve the number of units sold. Otherwise choose one of the operators randomly.

For a given state S_i we define its preserve branch denoted by *preserve_branch*(S_i) as the set of states formed by iteratively traversing left starting at state S_i until we reach a terminating state. In a similar manner we define *raise_branch*(S_i).

Theorem 1. Let T_0 be a state tree whose root is S_0. We show that the set of terminating states in T_0 is fully covered by a subset of terminating states. The subset, $S_{T_0}{}^*$, is defined as the set of terminating states reached by traversing on *raise_branch*(S_i) for each state S_i in *preserve_branch*(S_0). Each traversal from state S_i to its terminating state in *raise_branch*(S_i) is bounded by $O((l_{max} - l_0)/\Delta)$ states. In the worst case the size of *preserve_branch*(S_0) is $(l_{max} - l_0)/\Delta$ therefore the analysis of all states in *preserve_branch*(S_0) is bounded by $O((l_{\max} - l_0/\Delta)^2)$.

We define $S_{T_0}{}^*$ formally as follows. Let $P_{T_0} = preserve\_branch(s_0)$ and then $R_{T_0} = \bigcup_{s \in P_{T_0}} raise\_branch(s)$

$S_{T_0}{}^* = \{s \in R_{T_0} \cup P_{T_0} \mid \deg(s) = 0\}$. In this theorem we claim that $S_{T_0} = S_{T_0}{}^*$. That is, $S_{T_0}{}^*$ covers all the terminating states in the state tree produced by the auction execution. The definitions of Theorem 1 are illustrated in Figure 3. ■

The proof is omitted for space reasons. From Theorem 1 it follows that one needs to search only $O((l_{\max} - l_0/\Delta)^2)$ nodes to find the best action the auctioneer should take. The proof is omitted for space reasons. Based on theorem 1, one can further reduce the complexity to linear. This makes the suggested strategy feasible for real expanding auctions. Empirical results have shown that under diverse probability distribution functions the informed strategy we introduced produces significant gains.

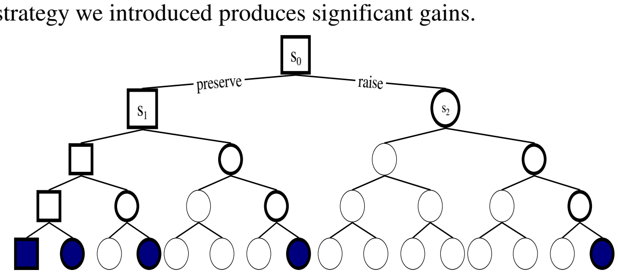

Figure 3. $S_{T_0}{}^*, R_{T_0}, P_{T_0}$ definitions: P_{T_0} states are rectangle shaped, R_{T_0} states are marked with a thick border style, $S_{T_0}{}^*$ are filled shapes.

3. REFERENCES

[1] www.olsale.co.il/olsale/rules.aspx?type=1