Association for Information Systems AIS Electronic Library (AISeL)

AMCIS 2012 Proceedings

Proceedings

Providing Information Feedback to Bidders in Online Multi-unit Combinatorial Auctions

Anup Sen Management Information Systems, Indian Institute of Management Calcutta, Kolkata, West Bengal, India., sen@iimcal.ac.in

Amitava Bagchi Heritage Institute of Technology, Kolkata 700107, India., bagchi@iimcal.ac.in

Follow this and additional works at: http://aisel.aisnet.org/amcis2012

Recommended Citation

Sen, Anup and Bagchi, Amitava, "Providing Information Feedback to Bidders in Online Multi-unit Combinatorial Auctions" (2012). AMCIS 2012 Proceedings. 5. http://aisel.aisnet.org/amcis2012/proceedings/DecisionSupport/5

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2012 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Providing Information Feedback to Bidders in Online Multi-unit Combinatorial Auctions

Anup K Sen Indian Institute of Management Calcutta sen@iimcal.ac.in

Amitava Bagchi Institute of Technology, Kolk

Heritage Institute of Technology, Kolkata, India <u>bagchi@iimcal.ac.in</u>

ABSTRACT

Bidders in online multi-unit combinatorial auctions face the acute problem of estimating the valuations of an immense number of packages. Can the seller guide the bidders to avoid placing bids that are too high or too low? In the single unit case, fast methods are now available for incrementally computing, for each package at each time instant, the recommended lower bound (Deadness Level) and upper bound (Winning Level) on the next bid. But when there are multiple units of items, it becomes difficult to compute the Deadness Level of a package accurately. An upper bound on this quantity can be derived however, and a bid that stays within this bound and the Winning Level is "safe", in the sense that it is not wasted and has the potential to become a winning bid. What is now needed is an incremental procedure for speeding up the computation of this bound.

Keywords: E-Commerce, Multi-User Systems, Software Agents

INTRODUCTION

Two items are said to be *complementary* when their combined valuation is higher than the sum of their individual valuations. When such items are sold together by auction, buyers can bid for combinations (packages, bundles) of items. Auctions that permit bids to be placed on packages are called *combinatorial auctions* (CAs). Consider the FCC spectrum auctions that sell licenses to wireless telephone operators for selected frequencies and geographical regions. A buyer is usually interested in acquiring a combination of licenses, and is willing to pay more for it than the sum of the prices of the individual licenses. Other examples can be cited from business domains such as logistics, transportation and travel, and supply chain management. In these single-unit CAs, only one unit of each item is available for sale.

In the last few years, the increasing popularity of online (continuous) auctions, such as the eBay C2C auctions, has spurred rapid growth in E-commerce. But such auctions still remain largely confined to single items. It is now agreed that when complementary items are present, it makes business sense to allow bids on packages. But CAs pose serious technical challenges. Additional complications arise in the online case, in which, owing to the extended time duration, a bidder must be allowed to join and leave the auction at any time. This compels the seller to provide information feedback to the bidder on the current state of the auction to enable meaningful bids to be placed. Some schemes for the efficient implementation of multiround and online single-unit CAs have been proposed in recent years. But multi-unit CAs, in which multiple units of items are auctioned simultaneously, have not yet received much attention. Online CAs belong to the class of multi-agent systems; there is a central webserver that runs the auction. Each bidder independently connects to it via the Internet and makes use of bidding software (*proxy agent*) to communicate with it.

In multi-unit CAs the number of possible packages is extremely large, and it becomes unbearably burdensome for an online bidder to determine how much to bid on a package. Suppose an online multi-unit CA has been organized to sell three items X, Y and Z, all of which are needed for assembling a PC. A PC assembly plant, to supplement its current inventory of items, would like to bid for a package consisting of 150 units of X, 200 units of Y and 100 units of Z. It can participate in the auction only if it is able to form a satisfactory estimate of the valuation of this package and the large number of other similar packages. In the single unit case, the seller can lighten the bidder's load by supplying information feedback on two numeric parameters that characterize the current status of a package p, namely the Deadness Level DL(p) and the Winning Level WL(p). When placing the next bid on p, DL(p) and WL(p) serve the bidder as lower and upper bounds on the bid value, and help to ensure that the bid is neither too low nor too high. A very low bid would be wasted, and a very high bid might result in overpayment for a package. Both WL(p) and DL(p) can be computed using dynamic programming procedures. An important property of DL(p) is that it is non-decreasing in time, so that a bid that lies below the current Deadness Level remains inactive at all subsequent instants.

There is a major point of difference between single-unit and multi-unit CAs. In the former, at most one bid on a package p is active at any time instant, corresponding to the highest bid that has been placed on p by any bidder up to that instant; other bids that have been placed on p are inactive and do not play any further role in the auction. In the multi-unit case, however, a

multiset (Knuth, 1973) of items is on auction, since an item can have multiple identical units. There can be more than one active bid on p at any instant because it might be possible to accommodate more than one copy of p within the multiset. This complicating factor forces a careful reinterpretation of the meaning of the Deadness and Winning Levels. How should we redefine these two parameters if their basic function is to remain the same as before, namely, to guide a bidder to bid intelligently? Significant cost reductions are anticipated in the procurement of goods and materials if multi-unit CAs can be made efficient by providing the seller a software agent to recommend upper and lower bounds on bid values of packages. Here we make an attempt to generalize ideas taken from the single-unit case. However, we find that though the Winning Level can be readily determined, an exact expression for the Deadness Level is hard to derive. In view of this difficulty, we try to compute a reasonable upper bound on the Deadness Level. A bid that stays within this bound and the Winning Level is "safe", in the sense that it is not wasted and has the potential to become a winning bid. A major issue that remains to be resolved is to determine the upper bound on the Deadness Level in an incremental manner.

In summary, this paper has the following broad objectives:

- To distinguish clearly between the bidding processes in single-unit and multi-unit CAs with the help of illustrative examples;
- To summarize the properties of the Deadness and Winning Levels of a package in the single-unit case, and then explain why it is difficult to determine the Deadness Level in the multi-unit case;
- To suggest methods for computing the two levels in multi-unit CAs.

LITERATURE REVIEW

Combinatorial Auctions (CAs), which permit bidders to bid on packages of items (Crampton, Shoham and Steinberg, 2006), received widespread publicity with the auction of frequency spectrum by the FCC (Goerce, Holt and Ledyard, 2006). The objective in all auctions, including CAs, is to maximize the seller's revenue. To allocate packages to the highest bidders, the *Winner Determination Problem* (WDP) must be solved. This is hard to do efficiently when the number of items is large (Sandholm, 2002; Sandholm and Suri, 2003), or when there are multiple units of items (Brammert and Endriss, 2008). CAs can be single-round, multi-round or continuous (on-line). The first price sealed bid CA is a single-round auction. The other single round CA is the VCG mechanism which integrates Vickrey's seminal ideas (Krishna, 2002) with the Clarke-Groves design (Clarke, 1971).

Multi-round CAs allow bidders to raise their bids in each round on the basis of the outcomes of the previous rounds. The Simultaneous Multiple Round (SMR) auction has been used by the FCC (1994-2003) to allocate frequency spectrum. Ascending Proxy auctions and the Clock-Proxy auction (Crampton et al., 2006) assume there is a single unit of each item, and allow multi-round package bidding using specific rules for fixing minimum prices on items and bundles at the start of each round. Decision support tools for progressive multi-unit combinatorial procurement auctions have been proposed in (Ervasti and Leskela, 2010; Leskela, Teich, Wallenius and Wallenius, 2007).

In on-line CAs, bidders can join the auction at any time and thus need information feedback after every bid. This issue can be addressed by providing bidders with the values of the Deadness and Winning Levels (DLs, WLs) of each package. The method described in (Adomavicius and Gupta, 2005; Adomavicius, Curley, Gupta and Sanyal, 2011) is efficient in the sense that non-competitive bids are eliminated, so no time is wasted on such bids. It makes use of a dynamic programming formulation to solve the WDP in an incremental manner. An improvement suggested by (Chakraborty, Sen and Bagchi, 2008) helps to scale up the method to a larger number of items. No results have been reported as yet on the DLs and WLs of packages in multi-unit CAs.

PRELIMINARIES: THE SINGLE-UNIT CASE

To get a better understanding of online CAs in the multi-unit case, we summarize the single-unit case first. We use time instants in our definitions to reflect the online situation. Let *N* be the set of items on auction. We assume that the online auction starts at time instant 1 and ends at time instant T > 1. Bids $b(q_1,1)$, $b(q_2,2)$, $b(q_3,3)$, ..., $b(q_T,T)$, having non-negative integer values, are placed by bidders on packages $q_1, q_2, q_3, ..., q_T$ at successive integer time instants 1, 2, 3, ..., *T*, where for $1 \le t \le T$, q_t is a subset of *N*. (The packages $q_1, q_2, q_3, ..., q_T$ are not necessarily distinct. Bids arrive in sequence at irregular time intervals; the instants are numbered 1, 2, 3, ..., just for convenience.).

Definition 1: Maxfit(p,t): Bids $b(q_r,r)$, $1 \le r \le t$, have already been placed in the auction by time instant *t*. Let *p* be any package (*i.e.*, *p* is any subset of *N*). We now try to fit the packages q_r into *p* in a non-overlapping manner, no q_r being used more than once, so as to maximize the sum of the corresponding bid values $b(q_r,r)$. This maximum sum is Maxfit(p,t).

Thus Maxfit(N,T) is the maximum revenue the seller can earn at the end of the auction. The set of packages whose bid values determine Maxfit(N,T) form the winning combination of packages, and the corresponding set of bidders is the winning set of bidders. (Ties are resolved arbitrarily.) Similarly, Maxfit(N,t) is the maximum revenue the seller can earn if the auction is viewed as terminating at time *t*; this gives a provisional winning combination of packages and a provisional set of winning bidders at time *t*.

We are now in a position to define the two non-negative parameters WL(p,t) and DL(p,t).

Definition 2: a) WL(p,t): The Winning Level of package p at time instant t is the minimum amount that must be bid on p at instant t+1 to ensure that p becomes a member of the provisional winning combination of packages at instant t+1.

b) DL(p,t): The *Deadness Level* of package p at time instant t is the smallest bid that can be placed on p at instant t+1 for which there exists a (hypothetical, perhaps empty) sequence of bids that puts p in the provisional winning combination of packages at some instant $t_1 \ge t+1$.

Definitions 2(a)-(b), though stated differently from the corresponding ones in (Adomavicius and Gupta 2005), are essentially equivalent to them. Both WL(p,t) and DL(p,t) can be determined using dynamic programming procedures.

Instant	Package	Bid	Maxfit(N)	Maxfit(p)	Maxfit(N\p)	WL(p)	DL(p)
1	{X, Y}	20	20	0	0	20	0
2	<i>{X}</i>	10	20	0	10	10	0
3	{Y}	15	25	15	10	15	15
4	{X, Y}	50	50	15	10	40	15

Table 1: Example of Single-unit CA

Example 1: Let $N = \{X, Y\}$ and T > 4. Let the given sequence of bids be as follows: $b(\{X,Y\},1) = 20$, $b(\{X\},2) = 10$, $b(\{Y\},3) = 15$, $b(\{X,Y\},4) = 50$. For the package $p = \{Y\}$, the values of WL(p) and DL(p) at the ends of successive instants are shown in Table 1. The package $N p = \{X\}$ is the complement of p with respect to N.

At time instant 1 there is only one bid and that is on package *N*. Since Maxfit(N) = 20 but Maxfit(Np) = 0, a bid value of 20 gets *p* included in the winning combination of packages at instant 2, so WL(p,1) = 20; DL(p,1) = 0 since we can place a bid of 0 on *p* at instant 2 and a (hypothetical) bid of 20 on *Np* at instant 3, which will result in *p* with a bid value of 0 getting included in the provisional winning combination of packages. It is easily seen that WL(p,3) = 15; DL(p,3) = 15 also, since only one copy of *p* can be accommodated in *N*, and the current bid value of *p* is already 15. The seller's revenue at time instant 4 is 50.

We now summarize a few basic properties of online single-unit CAs and determine expressions for the two levels. Some of the properties have been stated in (Adomavicius and Gupta 2005). We use our definitions above to prove these properties.

Claim 1: $Maxfit(p,t) + Maxfit(N \mid p,t) \leq Maxfit(N,t)$.

Proof: Each item has only one unit, so all active packages are distinct, and the same package cannot take part in the computations of both Maxfit(p,t) and Maxfit(N p,t). Since $N = p \cup N p$, Maxfit(N,t) cannot be less than Maxfit(p,t)+Maxfit(N p,t); however, Maxfit(p,t)+Maxfit(N p,t) can be less than Maxfit(n,t) because it might not be possible to fit the packages in Maxfit(N,t) into the two separate regions p and Np without straddling the boundary between them.

Claim 2: $DL(p,t) \leq WL(p,t)$.

Proof: Clear from the definitions of the two terms.

Claim 3:
$$DL(p,t) = Maxfit(p,t)$$
.

Proof: Any package that contributes to Maxfit(p,t) is a subset, proper or improper, of p. So a new bid placed on p, in order to be active, must have a value > Maxfit(p,t), otherwise it will have no impact on the auction outcome. If the bid on p at time t+1 is Maxfit(p,t), then the Maxfit values of p, Np and N do not change at instant t+1, so a (hypothetical) bid of Maxfit(N,t+1) - Maxfit(p,t+1) on Np ensures that DL(p,t) = Maxfit(p,t+1) = Maxfit(p,t).

Claim 4: $WL(p,t) = Maxfit(N,t) - Maxfit(N \mid p,t)$.

Proof: From the definition of WL(p,t), a (hypothetical) bid of Maxfit(N,t) - Maxfit(N p,t) on p at instant t+1 puts p in the provisional winning set of packages at instant t+1. \Box

Claim 5: For any package p, DL(p,t) is non-decreasing in t.

Proof: Maxfit(p,t) is non-decreasing in *t* because any bids placed after time instant *t* can only increase the *Maxfit* value of *p*, so this claim is implied by Claim 3. \Box

Claim 5 states a very desirable property of DL(p,t) that has great practical significance. The seller supplies to the bidder the value of DL(p,t) as a lower bound on the next bid on p, so this value should not decrease with time; otherwise a bid on p that was not considered earlier by a bidder because it was too low might later form part of a winning combination.

Example 1 shows that Claim 5 does not hold for WL(p,t). Some of the above claims are either not valid in the multi-unit case or are valid only in a weaker form (see below).

GENERALIZATION: THE MULTI-UNIT CASE

In multi-unit CAs, items have multiple units so instead of a set N of items on auction there is a multiset S of items; a package p is a subset of S (with both p and S being viewed as multisets). Definitions 1-2 get modified as given below. The symbols MWL(p,t) and MDL(p,t) are used in place of WL(p,t) and DL(p,t) to make a distinction between the single and multi-unit cases.

Definition 3: Maxfit(p,t): Let p be any package (*i.e.*, p is any subset of S). We fit the packages q_r , $1 \le r \le t$, into p in a nonoverlapping manner, no q_r being used more than once, ensuring that the number of units of an item put into p does not exceed the total number allowable in p. Maxfit(p,t) is the maximum sum obtainable of the corresponding bid values.

Note that no q_r is used more than once when filling up p, but the same package (occurring as two different q_r 's in the bid sequence) might be used more than once, which cannot happen in the single-unit case. Otherwise this definition is identical to Definition 1.

Definition 4: a) MWL(p,t): The Winning Level of package p at time instant t is the minimum amount that must be bid on a copy of p at instant t+1 to make that copy of p a member of the provisional winning combination of packages at instant t+1.

The reference to "a copy of p" underscores the point that more than one copy of p might be simultaneously active in the auction. Otherwise the definition is the same as in the single-unit case.

b) MDL(p,t): Let b(t) be the smallest bid that can be placed on a copy of p at instant t+1 for which there exists a (hypothetical, perhaps empty) sequence of bids that puts this copy of p in the provisional winning combination of packages at some instant $t_1 \ge t+1$. The *Deadness Level* of p at time instant t is the minimum of the b(r) values for all $r, t \le r < T$.

Remark 1: (a) The b(r) values for all $r, t \le r < T$, will be non-decreasing in the single-unit case and thus in the Definition 2(b), DL(p,t) is defined to be equal to b(t). If a bid is inactive (dead) ($\le DL(p,t)$) at an instant t, it should remain dead at all subsequent instants (with non-decreasing DL(p,t)). However, in the multi-unit case, the b(r) values for all $r, t \le r < T$, may not be non-decreasing. Hence straightforward generalization of 2(b) is not acceptable because then a dead bid value at an instant may become active at some subsequent instants.

(b) The generalization made in 4(b) is therefore not straightforward though the definition reduces to Definition 2(b) in the single-unit case. The definition ensures that MDL(p,t) is non-decreasing in t. But the value of MDL(p,t) is dependent on the future course of the auction. As a result, MDL(p,t) can become hard to determine in some cases. Therefore, no exact expression for MDL(p,t), similar to DL(p,t) in Claim 3, can be derived. \Box

$p = XY^2 \qquad \qquad S \langle p = X^* Y^* \rangle$	$p = XY^2 \qquad \qquad S \backslash p = X^4 Y^6$
--	---

Figure 1:	Regions p	and S\p	in S	$=X^5Y^8$
-----------	-----------	---------	------	-----------

Example 2: Let the set of items be $N = \{X, Y\}$ and let T > 7. There are 5 units of X and 8 units of Y, so the multiset S of items on auction is $\{X, X, X, X, X, Y, Y, Y, Y, Y, Y, Y, Y\}$. For ease of notation we write $S = X^5Y^8$. Let the given sequence of bids beas follows: $b(XY^2, 1) = 140$, $b(X^4, 2) = 170$, $b(XY^2, 3) = 190$, $b(X^4, 4) = 300$, $b(XY^2, 5) = 150$, $b(XY^2, 6) = 5$, $b(XY^2, 7) = 20$. This says the first bid has been placed on a package consisting of one unit of X and two units of Y, and so on for the other bids. We observe that in this auction all the bids were placed on the packages XY^2 and X^4 . For the package $p = XY^2$, the values of MWL(p) and MDL(p) after each bid are shown in Table 2; here the package $S \setminus p = X^4Y^6$ (see Figure 1).

At the end of instant 1 there is only one package and it fits into *p* as well as into *S**p*. Suppose a bid of 0 is placed on a new unit of *p* at instant 2. This unit of *p* will fit into *S* in addition to the existing unit of *p*, and the two units of *p* together will still have the maximum value of 140, so MWL(p,1) = 0, and consequently MDL(p,1) = 0. MWL(p,2) = 140 because a new unit of *p* will not get considered in the computation of Maxfit(S,2) unless its bid value is at least 140, in which case it will displace the existing copy of *p* from *S*. At instant 3, with the bid of 190 on a new copy of *p*, Maxfit(S,3) becomes 360; at the same time, $Maxfit(S \mid p,3)$ becomes 330 because the two copies of *p* together fit into *S**p*. This implies MWL(p,3) = 30, as a third copy of *p* will fit into *S* as well and it would suffice if its bid value is 30, displacing the existing package X^4 from *S*. At instant

Instant	Package	Bid	Maxfit(S)	Maxfit(p)	Maxfit(S\p)	MWL(p)	MDL(p)
1	XY^2	140	140	140	140	0	0
2	X^4	170	310	140	170	140	0
3	XY^2	190	360	190	330	30	0
4	X^4	300	490	190	330	160	0
5	XY^2	150	490	190	480	10	0
6	XY^2	5	490	190	480	10	5
7	XY^2	20	500	190	480	20	20

Table 2: Example of Multi-unit CA

4, the new unit of X^4 increases Maxfit(S) to 490, but the value of $Maxfit(S \p)$ does not change, so MWL(p,4) becomes 160. At instant 5, the bid of 150 on a third unit of p changes $Maxfit(S \p)$ to 480, but Maxfit(S) remains unaffected, so MWL(p,5) decreases to 10. DL(p,5) = 0 because there exists a hypothetical sequence of bids, namely a bid of 0 on p and then a bid of 10 on the package $\{X\}$, that yields a total value of 490. This shows that DL(p) = 0 at instants 2-4 also based on the future values. Bid 6 does not affect Maxfit(S) and $Maxfit(S \p)$, and MWL(p) remains unchanged. MDL(p,6) = 5 because more than four units of p will not fit into S so a new bid on p must have a value of at least 5 in order to displace one of the existing units of p from S. Bid 7 increases Maxfit(S) but not $Maxfit(S \p)$, so Maxfit(S), MWL(p) and MDL(p) all increase. The seller's revenue at Instant 7 is Maxfit(S,7) = 500. \Box

Remark 2: Example 2 shows that Claim 1 fails to generalize to the multi-unit case; in Table 2, Maxfit(p) + Maxfit(S | p) > Maxfit(S) at all instants except Instant 2, because some of the packages on which bids have been placed fit into both *p* and *S**p*. Claims 2 and 4 generalize readily and so does Claim 5 (see below). A weaker form of Claim 3 has been claimed (see Claim 8). In all the claims, t, t_1, t_2, \ldots , are instants during an auction, $1 \le t, t_1, t_2 \le T$.

Claim 6: a) $Maxfit(p,t)+Maxfit(S \mid p,t)$ can be less than, equal to or greater than Maxfit(S,t);

b) $MDL(p,t) \leq MWL(p,t);$

c) $MWL(p,t) = Maxfit(S,t) - Maxfit(S \setminus p,t)$.

Proof: (a), see Examples 1 and 2, and Remark 2. Parts (b) and (c) are direct consequences of the definitions of the terms.

Claim 7: For any package p, MDL(p,t) is non-decreasing in t.

Proof: Follows from the definition of MDL(p,t). See Remark 1. \Box

UPPER BOUND ON MDL

ł

3

By Claim 6(b), the value of MWL(p,t) can be determined just like the value of WL(p,t) in single-unit case, and a dynamic programming procedure exists for it. The function $UB_MDL(p,t)$ given below computes an upper bound on MDL(p,t).

int UB_MDL(package p, int t)

// given: a multi-unit CA, a package *p* that is a subset of *S*, a time instant *t*, $1 \le t < T$ // this function returns an upper bound on the value of MDL(p,t) $D_t = 0$; for $(1 \le x \le t)$ if $UB(p, x) > D_t$ then $D_t = UB(p, x)$; $D_t = \min \{ D_t, MWL(p,t) \}$; // no upper bound on MDL(p,t) should exceed MWL(p,t)return D_t ; ł

int *UB*(package *p*, int *x*) // *x* is a time instant, $1 \le x \le T$

// the bids placed so far are $b(q_1,1), b(q_2,2), \dots, b(q_x,x)$ on packages q_1, q_2, \dots, q_x

// let $Q_x = \{ q_1, q_2, ..., q_x \}; Q_x$ is to be viewed as a multiset

determine $P_x = \{ q_r : q_r \text{ is a subset, proper or improper, of } p, 1 \le r \le x \}; // P_x \text{ is a multiset}$

determine the subset R_x of packages from Q_x that fit into S p with bid values adding up to Maxfit(S p, x); $//R_x$ is a multiset fit into package n the packages in the difference set $P \setminus R_x$ (viewed as a multiset) so as to maximize the sum y_x of their bid

fit into package p the packages in the difference set $P_x \setminus R_x$ (viewed as a multiset) so as to maximize the sum y_x of their bid values;

// N.B. if there is a tie and more than one set R_x can be found that determine Maxfit(S|p,x),

// then take the minimum of the corresponding y_x values

return y_x ;

}

Example 4: Consider the multi-unit CA: $S = X^8 Y^8$, T = 5, $p = X^2 Y^2$, $S \setminus p = X^6 Y^6$. The bids are: $b(X^2, 1) = 20$, $b(Y^2, 2) = 20$, $b(X^2Y, 3) = 30$, $b(X^4Y^5, 4) = 100$, $b(X^4Y^6, 5) = 115$. It is easily checked that $UB_MDL(p, 5) = \max \{0, 0, 0, 40, 30\} = 40$, which is an upper bound on MDL(p,5) since $MDL(p,5) \le 30$ by the definition of MDL(p,t). \Box

Claim 8: For any package *p* at any instant *t*, $1 \le t < T$, $MDL(p,t) \le UB_MDL(p,t) \le Maxfit(p,t)$.

Proof: On examining function $UB_MDL(p,t)$, it is clear that MDL(p,t) cannot exceed UB(p,t) which in turn cannot exceed $UB_MDL(p,t)$. $UB_MDL(p,t)$, on the other hand, cannot exceed Maxfit(p,t) due to the computational procedure. It might happen that owing to a tie there is another set R_t^* that also determines $Maxfit(S \mid p,t)$; in that case UB(p,t) will return the minimum of the two values. \Box

Remark 3: MDL(p,t) seems to be a natural generalization of the concept of Deadness Level to the multi-unit case. But it is difficult to compute it directly, and we have only provided an upper bound on it. Although this value is not accurate, the seller can supply it to the bidder as a tentative lower bound on the next bid on *p*. As $MDL(p,t) \le UB_MDL(p,t)$, the bid will be a "safe" one and will not be wasted, since it is certain not to dip below the threshold level of MDL(p,t). When there is only one unit of each item, $UB_MDL(p,t)$ equals MDL(p,t), so the computational procedure described above is consistent with the results given in single-unit case. However, it is difficult to compute $UB_MDL(p,t)$ incrementally.

The difficulty in computing MDL(p,t) is essentially caused by the fact that p and $S \ p$ have items in common. If the two packages have no common items then we would always have MDL(p,t) = Maxfit(p,t) as in single-unit case. But in practice it would be too drastic to impose the restriction that p and $S \ p$ should have no items in common. The equality of the two terms would also hold under the weaker condition that no package in Q_t is a subset of both p and $S \ p$. The general case gives rise to two interesting open problems:

Open Problem 1: Formulate a computational procedure that will determine $UB_MDL(p,t)$ in an incremental manner for all values of t, $1 \le t < T$.

Open Problem 2: Characterize the multi-unit CAs for which MDL(p,t) = Maxfit(p,t) for all values of $t, 1 \le t < T$.

CONCLUSION

In some areas of business activity, such as in the procurement and sale of goods and materials, online multi-unit CAs are likely to find useful application. But such auction schemes still have no convenient implementations, partly because no satisfactory solutions have yet been found to the bidding problems faced by bidders. The number of packages is exceedingly large, making it hard for bidders to form good estimates of package valuations. In the single-unit case, two package parameters that can be computed by the seller, called the Deadness Level and the Winning Level, help to guide the bidder in placing the next bid. The computational procedures are incremental in nature and therefore quite fast. This paper tries to generalize the ideas to the multi-unit case, in which more than one unit of a package can be active simultaneously. While the Winning Level of a package can be determined in the same manner as before, the notion of Deadness Level must be reinterpreted carefully. A method for computing an upper bound on the Deadness Level has been described here. No procedure for computing the exact value has yet been formulated. Moreover, the Deadness Level needs to be computed in an incremental manner; otherwise, online multi-unit CAs would run too slowly for use in real life situations.

REFERENCES

1. Adomavicius, G., Curley, S., Gupta, A. and Sanyal, P. (2011). Effect of Information Feedback on Bidder Behavior in Continuous Combinatorial Auctions. *Management Science*, Articles in Advance, 1 - 20.

- 2. Adomavicius, G. and Gupta, A. (2005), Toward Comprehensive Real-Time Bidder Support in Iterative Combinatorial Auctions, *Information Systems Research*, 16(2), 169-185.
- 3. Brammert, O. and Endriss, U. (2008), Comparing Winner Determination Algorithms for Mixed Multi-Unit Combinatorial Auctions, *AAMAS*, Estoril, Portugal, 1601-1604.
- 4. Chakraborty S, Sen, Anup K. and Bagchi, A. (2008), Reducing Operational Overheads of Seller and Bidder in Online Combinatorial Auctions, 18th Workshop on Information Technologies and Systems (WITS-2008), 200-205.
- 5. Clarke, E. H. (1971), Multipart Pricing of Public Goods. Public Choice, 11: 17 33.
- 6. Cramton, P., Shoham, Y. and Steinberg, R. (Eds) (2006), *Combinatorial Auctions*, The MIT Press, Cambridge, MA, USA.
- 7. Ervasti, E and Leskela, R. (2010), Allocative Efficiency in Simulated Multiple-unit Combinatorial Auctions with Quantity Support, *European Journal of Operations Research*, 203, 251-260.
- 8. Goeree J.O., Holt, A.C. and Ledyard, J. O. (2006), An Experimental Comparison of the FCC's Combinatorial and Non-Combinatorial Simultaneous Multiple Round Auctions. *Report for Wireless Telecommunications Bureau of the Federal Communications Commission (FCC).*
- 9. Knuth, D. E. (1973), The Art of Computer Programming (vol 3), Sorting & Searching, Addison-Wesley.
- 10. Krishna, V. (2002), Auction Theory, London, Academic Press.
- 11. Leskela R, Teich, J., Wallenius, H. and Wallenius, J. (2007), Decision Support for Multi-Unit Combinatorial Bundle Auctions, *Decision Support Systems* 43: 420 434.
- 12. Sandholm, T. (2002), Algorithms for Optimal Winner Determination in Combinatorial Auctions, *Artificial Intelligence Journal*, vol 135, 1-54.
- 13. Sandholm, T. and Suri, S. (2003), BOB: Improved Winner Determination in Combinatorial Auctions and Generalizations, *Artificial Intelligence Journal*, vol 145, 33 58.