Indian Institute of Management Calcutta

Working Paper Series

WPS No. 749
July 2014

Determining the Deadness Levels of Packages in Online Multi-unit Combinatorial Auctions

Anup K Sen
Professor, Indian Institute of Management Calcutta
D. H. Road, Joka P.O.Kolkata 700 104
sen@iimcal.ac.in

Amitava Bagchi
Professor, Heritage Institute of Technology
East Kolkata Township, Anandapur, Kolkata 700107
amitava.bagchi@heritageit.edu

Determining the Deadness Levels of Packages in Online Multi-unit Combinatorial Auctions

Anup K Sen
Indian Institute of Management Calcutta

Amitava Bagchi
Heritage Institute of Technology Kolkata

Abstract

In online iterative combinatorial auctions, the deadness level (DL) of a package serves as a tight lower bound on a fresh bid that can be meaningfully placed on the package. Computational methods exist for determining the DL values of packages in the single-unit case. But when there are multiple identical units of items, these levels are hard to determine, and no closed form expression or computational method has been proposed as yet. This note examines the properties of package DLs in the multi-unit case; it provides theoretical results with supporting illustrative examples and presents for the first time an exact method for computing DL values. This could help to promote more widespread use in industry of such online auctions.

Keywords: Multi-unit combinatorial auctions, online combinatorial auctions, deadness levels (DLs)**,** intelligent agent-based systems

1. INTRODUCTION

When complementary items are put on auction, bidders must be permitted to bid on packages (*i.e.*, bundles) of items. In such combinatorial auctions (CAs), bidders need guidance from the seller to estimate package valuations correctly. When only one unit of each item is on auction, we can make use of the notion of 'quotes' (Sandholm 2002) to define a lower bound (*Deadness Level DL*) and an upper bound (*Winning Level WL*) for each package. These bounds can be readily computed, enabling bidders to place bids in the intermediate region between them. But when multiple identical units of items are on auction, no method is currently available for computing the exact value of the DL of a package.

The best known example of the single-unit CA is the FCC spectrum auction conducted by the U.S. government. This auction is multi-round, and information about its current state is provided to bidders in the inter-round breaks (Brunner *et al*., 2010; Kagel *et al*., 2010). Our interest, however, is in online (*i.e.*, continuous), iterative (Parkes, 2006), multi-unit eBay-like auctions, in which, in addition to the maximization of the seller's revenue, an important objective is bidder satisfaction and high bidder participation. A bidder must be allowed to join and leave the auction at any time, which compels the seller to provide information feedback after each bid in the form of a lower bound (*Multi-unit Deadness Level MDL*) and upper bound (*Multi-unit Winning Level MWL*) on packages of interest. Such auctions have a pre-announced duration, and winners are determined at termination with the help of a WDP (*Winner Determination Problem)* algorithm (Sandholm, 2006).

Multi-unit CAs differ from single-unit CAs in that a package *p* consists of a *multi-set* of items, and more than one bid on *p* can get included in a winning combination. This forces a more careful interpretation of the meanings of the deadness and winning levels. Significant cost reductions are possible in the procurement of goods and materials if multi-unit CAs can be efficiently implemented. The major difficulty lies in determining an exact value of the deadness level *MDL*(*p*) of a package *p*, and this problem has remained unsolved for some years. In this paper we propose a method for the first time for computing *MDL*(*p*) accurately; this will make it possible for a bidder to restrict a bid B on a package p to the "safe" range between $MDL(p)$ and $MWL(p)$, thereby keeping (p,B) in contention for inclusion in winning combinations in future.

The objectives of this paper are twofold:

- To establish the properties of the Deadness and Winning Levels (*MDL*s and *MWL*s) of packages in the multi-unit case; these properties are often interesting and sometimes differ from the ones in the single-unit case;
- To describe a scheme for the exact computation of the MDL of a package at any given instant during an online multi-unit CA; we provide a programming procedure and give some computational results.

2. BOUNDS ON MEANINGFUL BIDS

Let $S = \{ X^{\alpha}, Y^{\beta}, \ldots \}$ be a multi-set of items on auction; here S consists of α identical units of item X, β identical units of item Y, and so on. The auction starts at time instant 1 and ends at some time instant $T > 1$. Bids $b(q_1,1), b(q_2,2),\ldots, b(q_T,T)$, having non-negative integer values, are placed by bidders on packages $q_1, q_2,\ldots, q_T$ at successive integer time instants 1, 2,..., T, where for $1 \leq t \leq T$, q_t is a multi-set that is a subset of S. The packages $q_1, q_2,\ldots, q_T$ are not necessarily distinct. (Bids arrive at irregular time intervals; the instants are numbered 1, 2, 3, … , just for convenience.)

Definition 1: *Maxfit*(p,t): Let p be any package (*i.e.*, p is any subset of S). We view p as a container and insert the packages q_k, $1 \leq k \leq t$, into p in a non-overlapping manner, no bid being used more than once, ensuring for each item that the number of units put into p does not exceed the capacity of p. *Maxfit*(p,t) is the maximum sum obtainable of the corresponding bid values.

While no bid is used more than once when filling up p, the same package (occurring as two different q_k's in the bid sequence) might get included more than once in p, which cannot happen in the single-unit case. Thus more than one copy of a package can be simultaneously *active* with its own bid value, and these copies can be considered independently for inclusion in the (provisional) winning combinations. Other copies of the package might be inactive and not play any further role in the auction (see Example 1 below).

Definition 2: a) $MWL(p,t)$: The *Winning Level* of a package p at time instant t is the smallest (non-negative) bid B on a copy of p at instant t+1 that makes the pair (p,B) a member of a provisional winning combination of packages at instant t+1.

The definition is the same as in the single-unit case (Adomavicius and Gupta, 2005), except that more than one copy of p might be simultaneously active.

b) $MDL(p,t)$: The *Deadness Level* of a package p at time instant t is the smallest bid B on a copy of p at instant t+1 such that there exists a (hypothetical, perhaps empty) sequence of bids that puts the pair (p,B) in a provisional winning combination of packages at some instant $t_1 \geq t+1$.

A bid of value $B = MWL(p,t)$ on p at instant t+1 ensures that the pair (p,B) is in a winning combination at instant t+1, while a bid of value $B' = MDL(p,t)$ on p at instant t+1 implies that there exists a continuation of the auction such that the pair (p,B') is in a winning combination at a future instant. When we say '*(p,B) is in a winning combination*' we imply that any ties in value are resolved in favor of (p,B); in order for (p,B) to be in a winning combination unequivocally, B must exceed $MWL(p,t)$ (or $MDL(p,t)$ as the case may be) by a small amount (say the bid increment).

Example 1: Let the multi-set S of items consist of 8 units of X and 8 units of Y, and suppose that $T > 5$. For ease of notation we write $S = X^8Y^8$. Let the given sequence of bids be as follows: $b(X^2Y^2,1) = 20$, $b(X^2Y,2) = 25$, $b(X,3) = 30$, $b(X^3Y^5,4) = 100$, $b(XY^2,5) = 35$. This says the first bid of value 20 has been placed on a package consisting of two units of X and two units of Y, the second bid of value 25 has been placed on a package consisting of two units of X and one unit of Y, and so on. Let $p = X^2Y^2$; then $S\backslash p = X^6Y^6$ (see Figure 1). The values of $MWL(p)$ and $MDL(p)$ after each bid are shown in Table 1.

$p = X^2Y^2$	$S\backslash p = X^6Y^6$

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

At the end of instant 1 there is only one package and it fits into p as well as into $S\backslash p$. Suppose a bid of 0 is placed on a copy of p at instant 2. This copy of p fits into S in addition to the existing package X^2Y^2, and the two together have a total bid value of 20, so $MWL(p,1) = 0$. Consequently, $MDL(p,1) = 0$ as well. At instant 2, a bid of 25 is placed on the package X^2Y. $MWL\ (p,2)$ remains 0 because a copy of p with bid 0 will fit into S along with the two packages X^2Y^2 and X^2Y, and the three packages together will have a total bid value of 45.

$S = X^8Y^8$, $p = X^2Y^2$, $S\backslash p = X^6Y^6$							
Instant	**Package**	**Bid**	***Maxfit*(*S*)**	***Maxfit*(*p*)**	***Maxfit*(*S\p*)**	***MWL*(*p*)**	***MDL*(*p*)**
1	X^2Y^2	20	20	20	20	0	0
2	X^2Y	25	45	25	45	0	0
3	X	30	75	30	75	0	0
4	X^3Y^5	100	175	30	155	20	20
5	XY^2	35	190	65	155	35	25

Table 1: Values of *MWL*(*p*) and *MDL*(*p*) in Example 1

At instant 3 a bid of value 30 is placed on the package X. $Maxfit(S,3)$ and $Maxfit(S\backslash p,3)$ both become 75, and $Maxfit(p,3)$ becomes 30. If we now place a bid of 0 on a copy of p at instant 4, this will fit into S, so $MWL(p,3)$ and $MDL(p,3)$ both remain 0. At instant 4, the bid of value 100 on X^3Y^5 increases $Maxfit(S)$ to 175, but $Maxfit(S\backslash p)$ is only 155 because the four packages do not all fit into $S\backslash p$. So $MWL(p,4) = 20$ since a bid of 20 on p now makes $(p,20)$ a part of the winning combination $\{\ (p,20),\ (X^2Y,25),\ (X,30),\ (X^3Y^5,100)\ \}$. $MDL(p,4)$ is also 20 because a bid of $(X^2Y^2,20)$ at instant 5 makes it a part of winning combination.

At instant 5, the bid of 35 on the package XY^2 changes $Maxfit(S)$ to 190. $MWL(p,5)$ increases to 35, but $MDL(p,5)$ becomes 25 because a bid of 25 on p at instant 5 followed by a bid of 100 on X^4Y^4 does not change $Maxfit(S)$ and yields the winning combination $\{\ (X^2Y^2,\ 25),\ (X,30)\ ,(XY^2,35),\ (X^4Y^4,100)\ \}$. Here two copies of p, namely, $(X^2Y^2,\ 25)$ and $\{\ (X,30)\ ,(XY^2,35)\ \}$, are part of the winning combination. We note that: i) the original bid of 20 on $p = X^2Y^2$ remains active until instant 4, but at instant 5 when $MDL(p)$ increases to 25, the bid $(X^2Y^2,20)$ becomes inactive and plays no further role in the auction; ii) at instant 5 there are two bids that together fit into p with a total value of 65. □

2.1 BASIC PROPERTIES

We now list some basic theoretical properties of online multi-unit CAs. In the single-unit case, the following results are known to hold (Adomavicius and Gupta, 2005):

i) $Maxfit(p,t) + Maxfit(S\backslash p,t) \leq Maxfit(S,t)$;
ii) $WL(p,t) = Maxfit(S,t) - Maxfit(S\backslash p,t)$;
iii) $DL(p,t) = Maxfit(p,t)$;
iv) $DL(p,t) \leq WL(p,t)$;
v) $DL(p,t)$ is non-decreasing in t.

These results do not all generalize to the multi-unit case. For example, (i) and (iii) are no longer valid. In Table 1, at all five instants, $MDL(p,t) < Maxfit(p,t)$ and therefore $MDL(p,t)$ cannot be directly determined by computing $Maxfit(p,t)$. In addition, $Maxfit(p,t) + Maxfit(S\backslash p,t) > Maxfit(S,t)$, because some of the packages on which bids have been placed fit into both p and $S\backslash p$. But (ii), (iv) and (v) continue to hold as shown below.

Claim 1: $MDL(p,t) \leq MWL(p,t) = Maxfit(S,t) - Maxfit(S\backslash p,t)$.
Proof: The claim follows from the definitions of $MDL(p,t)$ and $MWL(p,t)$. Clearly, $MDL(p,t)$ can never exceed $MWL(p,t)$. If a bid $B \geq (Maxfit(S,t) - Maxfit(S\backslash p,t))$ is placed on p at instant t+1, then (p,B) becomes a part of the provisional winning combination. □

Claim 2: If a bid of value $B < MDL(p,t)$ is placed on p at instant $t+1$, then the pair (p,B) becomes inactive and plays no further role in the auction.
Proof: By the definition of $MDL(p,t)$, the bid (p,B) cannot form a part of any winning combination in future. □

Claim 3: For any package p, $MDL(p,t)$ is non-decreasing in t.
Proof: Consider any instant k, and suppose that $MDL(p,k) = A$ and $MDL(p,k+1) = B$. We have to show that $A \leq B$. We argue as follows. All the bids that have been placed at all instants $\leq k$ remain in consideration at instant k+1; however, one additional bid $b(q_{k+1},k+1)$ is placed at instant k+1. By definition, A is the *smallest* bid on p at instant k+1 for which there exists a sequence of bids that puts (p,A) in the (provisional) winning combination at a future instant. When computing A, all possible future sequences of bids placed at instants $\geq k$+1 are taken into consideration. The bid $b(q_{k+1},k+1)$ that actually gets placed at instant k+1 is only a particular case, and it imposes a constraint on the value of B but not on that of A, so $A \leq B$. □

3. COMPUTATION OF MDL VALUES

No systematic procedure for computing the *MDL* value of a package p at instant t in a multi-unit CA has been suggested as yet. Here we propose for the first time a method for determining $MDL(p,t)$ exactly. It works as follows: From the set $\Delta_t = \{\ b(q_k,k),\ 1 \leq k \leq t\ \}$ of bids that have been placed in the auction at instants $\leq t$, create all possible subsets of bids that fit into $S\backslash p$. Let these subsets be $p_1, p_2, ..., p_j$ (viewed as packages) in arbitrary order. Every p_k has an associated value $B_1(k)$ which is the sum of the bid values of its constituent bids. Suppose there is a bid B on p and a bid $B_2(k)$ on package $q_k = (S\backslash p)\backslash p_k$. Then $p + p_k + q_k = S$, and we try to achieve $B + B_1(k) + B_2(k) = Maxfit(S,t)$ (see Figure 2). To get a lower bound on the value of B, we eliminate from the set Δ_t those packages that have been used in computing $B_1(k)$, then pack into p as many as possible of the remaining packages in Δ_t so as to maximize the total value of the corresponding bids. $MDL(p,t)$ is obtained by varying p_k to find the smallest value of B that satisfies the above conditions (see Function Compute_MDL and Example 2 below).

(p,B)	$(p_k,B_1(k))$	(q_k,B_2)

Figure 2: Partition of *S* into Three Parts

// The function Compute_MDL computes $MDL(p,t)$ from the set of bids $\Delta_t = \{ b(q_k,k), 1 \le k \le t \}$

function Compute_MDL(*package p, instant t, set of bids* Δ_t)
{ let $MDL(p,t) = \infty$;
  compute $M = Maxfit(S)$ from the set of bids Δ_t;
  for every possible package r that fits into $S\backslash p$ and consists of a subset of packages from Δ_t
    { let $s = (S\backslash p)\backslash r$; // note that $p + r + s = S$
      let B_1 = sum of the bid values of the packages in r;
      place a hypothetical bid $(s, M - B_1)$;
      compute $newM = Maxfit(S)$ from the set $\Delta_t \ U \ \{ (s, M - B_1) \}$;
      compute $MDL(p,t) = \min \{ MDL(p,t), newM – M \}$;
    }
  return $MDL(p,t)$;
}

3.1 COMPUTATIONAL RESULTS

We now give some computational results that illustrate interesting aspects of Compute_MDL().

Example 2: Suppose $S = X^9Y^9$, $p = X^3Y^3$ and $T \geq 10$; then $S\backslash p = X^6Y^6$. Let $b(X^3,1) = 45$, $b(Y^2,2) = 50$ and $b(X^2Y^2,3) = 75$ be the first three bids. Table 2 shows the other bids. Values of r, s, B_1, B_2, M and $newM$ are also shown. Consider instant 4. A bid of 50 on p at instant 5 makes p a part of the provisional winning combination because we can place a (hypothetical) bid of 70 on package XY that will make $Maxfit(S) = 255$. Thus $MDL(p,4) = 255 – 205 = 50$. $MDL(p,t)$ values at other instants are similarly obtained. □

The data in Table 2 suggest the following observations:

- $MDL(p,t)$ is non-decreasing in t, but *not* $MWL(p,t)$.
- $MDL(p)$ changes even when bids are placed on packages that do not fit into p; *e.g.*, bids on X^4Y^3 or X^2Y^4 change $MDL(p,t)$. This is not possible in the single-unit case.
- At instant 6, $MDL(p,t)$ is 80, but no subset of existing bid values add up to 80, the closest such value being 75. This curious situation cannot arise in the single-unit case.
- There is no apparent pattern in the manner in which the package r changes with time. For example, $r = X^5Y^5$ at instant 4, but $r = Y^5$ at instants 5 and 6, and $r = X^4Y$ at instant 7. How can we explain these sudden changes in r?
- The bid (s,B_2) can never decrease $Maxfit(S)$, so $newM \geq M$ always. In fact, $newM(t+1) \geq M(t+1) + MDL(p,t)$.
- In the example of Table 2, the package p is kept fixed and its MDL values are computed at different instants. However, at step 4 of the algorithm, before bidding on Y^3, the bidder might want to know the value of $MDL(Y^3)$. In Table 3, for illustration, we determine the $MDL(q,t)$ and $MDL(q,t+1)$ values of the package q on which the bid is placed at instant t. S is kept unchanged at X^9Y^9, but q and $S\backslash q$ change with time. It is found that $MDL(q,t) \leq MDL(q,t+1)$, but the difference $MDL(q,t+1) - MDL(q,t)$ is sometimes large and sometimes small.

- While Compute_MDL() runs fast on the examples we have tried, in the worst case the running time is exponential in the number of bids. One way to speed it up would be to try and develop an incremental version that would determine, given a package *p*, the value of *MDL*(*p*,*t*+1) from the value of *MDL*(*p*,*t*) with a minimum of computation.

$S = X^9Y^9$ $p = X^3Y^3$ $S\backslash p = X^6Y^6$									
Instant	*Package*	*Bid*	*Maxfit*		*MWL*(*p*)	*r and* B_1	*s* and B_2	*newM*	*MDL*(*p*)
			S	$S\backslash p$					
4	Y^3	85	255	205	50	X^5Y^5 = (1) + (3) + (4) = 205	XY = 70	305	50
5	X^4Y^3	125	330	210	120	Y^5 = (2) + (4) = 135	X^6Y = 120	405	75
6	X^2Y^4	130	350	225	125	Y^5 = (2) + (4) = 135	X^6Y = 135	430	80
7	XY	55	360	260	100	X^4Y =(1) + (7) =100	X^2Y^5 = 175	445	85
8	X^2	30	370	265	105	X^6Y =(1) + (7) + (8) =130	Y^5 = 145	465	95
9	X^2Y	85	400	275	125	X^3Y^5 =(4) + (7) + (9)=225	X^3Y = 70	505	105
10	XY^2	100	450	320	130	X^6Y^2 =(1) + (7) + (9) =185	Y^4 = 135	580	130

Table 2: Values of *MDL(p)* in Example 2

Instant t	*Package q*	*Bid*	*MDL(q)*	
			Before bid	*After bid*
4	Y^3	85	0	0
5	X^4Y^3	125	50	95
6	X^2Y^4	130	75	130
7	XY	55	0	10
8	X^2	30	0	0
9	X^2Y	85	20	30
10	XY^2	100	25	50

Table 3: *MDL(q)* in Example 2

3.2 THEORETICAL RESULTS

Before we present any theoretical results related to Compute_*MDL*, we establish a more basic claim, namely that in any CA, if package *q* is a subset of package *r* then *MDL*(*q*,*t*) $\leq$ *MDL*(*r*,*t*) at every instant *t*. This proof does not depend on the computation procedure described above but it makes use of a similar construction and argument. Claims 3 and 4 taken together imply that *MDL*(*p*,*t*) is in a certain sense well-behaved, since it is non-decreasing with time and subsets of a package *p* have *MDL* values no larger than that of *p*.

Claim 4: Let *q* and *r* be packages where *q* is a subset of *r*. Then *MDL*(*q*,*t*) $\leq$ *MDL*(*r*,*t*) at every time instant *t*.

Proof: Suppose *MDL*(*q*,*t*) = *A* and *MDL*(*r*,*t*) = *B*. We want to show that *A* $\leq$ *B*. We divide *S* into three regions, where the first region holds package *r* exactly, the second r_1 consists of a subset of the packages on which bids have been placed at instants $\leq$ *t*, and the third r_2 equals $(S\backslash r)\backslash r_1$. The upper portion of Figure 3 shows the packages and the corresponding bids. Here, B_1 is the sum of the bid values on the packages that constitute r_1. We choose the value of B_2 in such a way that $B + B_1 + B_2 = Maxfit(S,t)$; this cannot occur for a smaller value since $MDL(r,t) = B$. Some of the packages on which bids have been placed in the auction so far have gone into the second region. Of the remaining ones, some will fit into p, but the total of their bid values cannot exceed B.

(r,B)	(r_1,B_1)	(r_2,B_2)

(q,B)	(q_1,B_1)	(q_2,B_2)

Figure 3: Proof of Claim 4

To show that $A \leq B$, we now divide S into three parts, where the first part holds package q exactly, the second part q_1 equals r_1, and the third part $q_2 = (S \backslash q) \backslash q_1$ as shown in the lower portion of Figure 3. The bids are the same as in the upper portion. Here too $B + B_1 + B_2 = Maxfit(S,t)$. No difficulties arise since q is a subset of r, and it follows that $A = MDL(q,t) \leq B$. This argument is valid only when q is a subset of r. □

At two instants t_1 and t_2, where $t_1 < t_2$, let the package q be q_1 and q_2 respectively. Suppose q_1 is a subset of q_2. Then by Claims 3 and 4 we must have $MDL(q_1, t_1) \leq MDL(q_2, t_2)$. It can be verified that Table 3 satisfies this condition.

We now show that Compute_MDL does indeed determine $MDL(p,t)$ exactly. We recall that $MDL(p,t) = B$ if there is a bid (p,B) at instant t+1 for which there is a sequence of bids that ensures (p,B) is in the winning combination at a future instant. We first establish that if this is the case then for an appropriate choice of a hypothetical bid, p will be in the provisional winning combination at instant t+2.

Claim 5: Suppose $b(q_1,k_1)$, $b(q_2,k_2)$, $b(q_3,k_3)$, …, $b(q_l,k_l)$ is the hypothetical sequence of bids at instants $k_1, k_2, \ldots, k_l$ needed to ensure that $(p, MDL(p,t))$ is in the winning sequence at some instant $\geq k_l$. Here the k_j's are $> t+1$ and in increasing order. Then a bid $(p,MDL(p,t))$ at instant t+1 followed by a bid on package $(q_1 + q_2 + \ldots + q_l)$ of value $(b(q_1,k_1) + b(q_2,k_2) + \ldots + b(q_l,k_l))$ at instant t+2 ensures that $(p, MDL(p,t))$ is in the winning combination at instant t+2.
Proof: Immediate. □

Claim 6: The procedure Compute_MDL yields the correct value of $MDL(p,t)$.
Proof: Suppose $MDL(p,t) = B$. Then by Claim 5, at instant t+2 there must exist packages (p,B), (r,B_1), (s,B_2) as in Figure 3 such that $p + r + s = S$ and $B + B_1 + B_2 = M = Maxfit(S,t)$. Here r consists of packages from Δ_t, *i.e.*, r is composed of packages on which bids have been placed at instants $\leq t$, B_1 is the sum of the corresponding bid values, and $s = (S \backslash p) \backslash r$. By definition, B must be the smallest bid value on p for which the above conditions are true. There are many possible choices for r. Compute_MDL takes each such (r,B_1) pair and determines appropriate pairs (p,B) and (s,B_2). Since $Maxfit(S,t)$ has already been determined, given B_1 we know the value of $B + B_2$, but B and B_2 are not individually known to us. To overcome this problem, we place a hypothetical bid $(s, M - B_1)$ at instant t+1 and compute $Maxfit(S)$ afresh to get $newM$, which must be $\geq M$ since an additional bid has been placed. We now work backwards. The quantity $(newM – M)$ is subtracted from $M - B_1$ to give us B_2, and a new bid (p,B) is now placed, where $B = newM – M$. The algorithm varies r and gets the minimum value of B; the construction ensures that there cannot exist any combination of packages in $\Delta_t \backslash r$ with values that add up to a quantity $> B$. □

4. CONCLUSION

In some areas of business activity, such as in the procurement and sale of goods and materials, online multi-unit CAs can find useful application. But such auction schemes still have no convenient implementations, partly because satisfactory solutions to some problems faced by bidders have been unavailable. The number of possible packages is large, making it hard for bidders to form good estimates of package valuations. In the single-unit case, two package parameters, the deadness level DL and the winning level WL, can be computed by the seller and communicated to bidders, and these help to guide bidders in placing the next bid. How can we extend the ideas to the multi-unit case, in which more than one copy of a package can be active simultaneously? The winning level MWL of a package can be determined as before, but not its deadness level MDL. Here we propose an exact method for computing MDL values to help bidders to avoid placing inactive (*i.e.*, dead) bids that play no role in the formation of winning combinations. It remains to develop a more efficient, and if possible, an incremental, algorithm for computing *MDL* values. It also remains to extend the notion of deadness level to more generalized versions of Combinatorial Auctions (see Fionda and Greco, 2013),

REFERENCES

[Adomavicius and Gupta, 2005] Adomavicius, G. and Gupta, A. (2005), Toward Comprehensive Real-Time Bidder Support in Iterative Combinatorial Auctions, *Information Systems Research*, 16(2): 169-185.

[Brunner *et al.*, 2010] Brunner, C., Goeree, J., Holt, C. and Ledyard, J. (2010), An Experimental Test of Flexible Combinatorial Spectrum Auction Formats, *American Economic Journal: Microeconomics*, 2(1): 39–57.

[Cramton *et al.*, 2006] Cramton, P., Shoham, Y. and Steinberg, R. (Eds) (2006), *Combinatorial Auctions*, The MIT Press, Cambridge, MA, USA.

[Fionda and Greco, 2013] Fionda, V. and Greco, G. (2013), The complexity of mixed multi-unit combinatorial auctions: Tractability under structural and qualitative restrictions, *Artificial Intelligence* 196, 2013, pp. 1-25.

[Kagel *et al.*, 2010] Kagel, J., Lien, Y. and Milgrom, P. (2010), Ascending Prices and Package Bidding: A Theoretical and Experimental Analysis, *American Economic Journal: Micro-economics*, 2: 160–185.

[Parkes, 2006] Parkes David C., Iterative Combinatorial Auctions, Chapter 2 in [Cramton *et al.*, 2006].

[Sandholm, 2002] Sandholm, T., Algorithms for Optimal Winner Determination in Combinatorial Auctions, *Artificial Intelligence* 135, 2002, 1-54.

[Sandholm, 2006] Sandholm, T., Optimal Winner Determination Algorithms. Chapter 14 in [Cramton *et al.*, 2006]