

D. R. Goossens, R. Müller, F. C. R. Spieksma

Matrix bids in combinatorial auctions:  
expressiveness and micro-economic properties

RM/07/016



Maastricht research school of **E**conomics  
of **T**echnology and **O**rganizations

Universiteit Maastricht  
Faculty of Economics and Business Administration  
P.O. Box 616  
NL - 6200 MD Maastricht

phone : ++31 43 388 3830  
fax : ++31 43 388 4873



# Matrix bids in combinatorial auctions: expressiveness and micro-economic properties\*

D.R. Goossens <sup>(1)</sup>, R. Müller <sup>(2)</sup>, F.C.R. Spieksma <sup>(1)</sup>

(1) ORSTAT, FETEW, Katholieke Universiteit Leuven, Belgium

(2) Quantitative Economics, Maastricht University, The Netherlands

## Abstract

A combinatorial auction is an auction where multiple items are for sale simultaneously to a set of buyers. Furthermore, buyers are allowed to place bids on subsets of the available items. This paper focusses on a combinatorial auction where a bidder can express his preferences by means of a so-called *ordered matrix bid*. Ordered matrix bids are a bidding language that allows a compact representation of a bidder's preferences, and was developed by Day (2004). We give an overview of how a combinatorial auction with matrix bids works. We elaborate on the relevance of the matrix bid auction and we develop methods to verify whether a given matrix bid satisfies properties related to micro-economic theory as free disposal, subadditivity, submodularity and the gross substitutes property. Finally, we investigate how a collection of arbitrary bids can be represented as a matrix bid.

**Keywords:** combinatorial auction, matrix bids, free disposal, subadditivity, submodularity, gross substitutes, expressiveness

## 1 Introduction

A combinatorial auction is an auction where multiple items are for sale simultaneously to a set of buyers. In a combinatorial auction a buyer is allowed to place bids on subsets of the items. These subsets are sometimes called *bundles*. The auctioneer decides – after one or more rounds or after a certain amount of time depending upon the design – to accept some of the bids and to allocate the items accordingly to the bidders. The main advantage of a combinatorial auction is that it allows a bidder to express his<sup>1</sup> preferences to a greater extent. Indeed, it may occur that a bidder values a set of items higher than the sum of his valuations for the individual items of this set. If this is the case, we say that these items are complements to this bidder; if the converse is true we say these items are substitutes. Notice that these complementarity or substitution-effects may be bidder-specific. A combinatorial auction offers a way to make use of complementarity or substitution-effects.

---

\*This research was partially supported by FWO Grant No. G.0114.03.

<sup>1</sup>his can be replaced by her (and he by she)

The popularity of combinatorial auctions (and corresponding research) has increased in recent years. As a result, there are many examples where combinatorial auctions prove to be a successful way to market items. The airline sector offers several possibilities for combinatorial auctions: landing slots (Ball, Donohue & Hoffman 2005, Rassenti, Smith & Bulfin 1982), seats (Eso 2001) and flights (Bleichwitz & Klier 2005). We also find applications of combinatorial auctions in truckload transportation (Caplice & Sheffi 2005, Ledyard, Olson, Porter, Swanson & Torma 2002, Sheffi 2004) and allocating bus routes (Cantillon & Pesendorfer 2005). We conclude this non-exhaustive overview by mentioning Epstein, Henríquez, Catalán, Weintraub & Martínez (2002), who use a combinatorial auction to assign catering contracts for meals in Chilean schools.

In a combinatorial auction in its most general form, bidders can bid whatever amount they please on any subset of items in which they are interested. This, however, immediately poses a problem when it comes to representing such a valuation:  $2^m - 1$  numbers might be needed (where  $m$  is the number of items). A way to overcome this difficulty is to use a bidding language. However, whereas a bidding language may allow a bidder to compactly represent his valuation, the set of possible bids that can be represented (i.e., the expressiveness) will be restricted. We refer to Nisan (2005) for an in-depth review of this issue.

In this paper, we investigate so-called *ordered matrix bids*, originally proposed by Day (2004). Matrix bids allow a compact representation of a bidder's valuation, and allow a bid on any subset of the items. However, matrix bids also restrict what a bidder can bid for these sets. This paper is organized as follows. In section 2, we give an overview of how an auction with ordered matrix bids works. We elaborate on the relevance of matrix bids in section 3. In section 4, we verify to what extent matrix bids are related to free disposal, complement freeness, decreasing marginal valuations and the gross substitutes property. Finally, in section 5, we investigate how a collection of arbitrary bids can be represented as a matrix bid.

## 2 The matrix bid auction

The matrix bid auction is a multi-item, single-unit combinatorial auction. This means that for each item that is auctioned, only one unit of this item is available. In the matrix bid auction, each bidder must submit a strict ordering (or ranking) of the items in which he is interested. Furthermore, we assume that for each bidder, the extra value an item adds to a set is determined only by the number of higher ranked items in that set, according to the ranking of that bidder.

Let  $G$  be the set of items, indexed by  $i$  and  $B$  the set of bidders, indexed by  $j$ . The ordering of the items is denoted by  $r_{ij}$ , which is item  $i$ 's position in bidder  $j$ 's ranking, for each  $i \in G$  and  $j \in B$ . This ordering should be strict in the sense that for each bidder  $j$ ,  $r_{i_1j} \neq r_{i_2j}$  for any pair of distinct items  $i_1$  and  $i_2$ . For instance, if  $r_{ij} = 2$ , item  $i$  is bidder  $j$ 's second highest ranked item.

Furthermore, each bidder  $j$  specifies values  $b_{ijk}$ , which correspond to the value the bidder is prepared to pay for item  $i$  given that it is the  $k$ -th highest ranked item in the set that bidder  $j$  is awarded. The  $b_{ijk}$  values allow to determine the value bidder  $j$  attributes to any set  $S \subseteq G$ . Indeed, bidder  $j$ 's bid on a set  $S$  is denoted as  $b_j(S)$  and can be computed as:

$$b_j(S) = \sum_{i \in S} b_{i,j,k(i,j,S)} \quad (1)$$

where  $k(i, j, S)$  is the ranking of item  $i$  amongst the items in the set  $S$ , according to bidder  $j$ 's ranking. Notice that equation (1) assumes that no externalities are involved, i.e. a bidder's valuation depends only on the items he wins, and not for instance on the identity of the bidders to whom the other items are allocated. The winner determination problem is, given the bids  $b_j(S)$  for each set  $S$  and each bidder  $j$ , to determine which bidder is to receive which items, such that the total winning bid value is maximized. Notice that we assume that each bidder pays what he bids for the subsets he wins.

Observe that the value for index  $k$  of item  $i$  in bidder  $j$ 's bid can never be higher than the rank  $r_{ij}$ . This allows us to arrange the values  $b_{ijk}$  as a lower triangular matrix for each bidder  $j$ , where the rows correspond to the items, ordered by decreasing rank and the columns correspond to values for  $k$ . Hence the name matrix bid (with order). Notice also that bidder  $j$ 's ranking  $r_{ij}$  does not necessarily reflect a preference order of the items. If an item is highly ranked, this merely means that its added value to a set depends on less items than the added value of a lower ranked item. Furthermore, we make no assumption regarding the  $b_{ijk}$  values. Indeed, these values may be negative, e.g. to reflect the disposal cost of an unwanted item. Specifying a sufficiently large negative value can also keep the bidder from winning this item in the first place. We do assume that bids are normalized, in the sense that any bidder's bid on the empty set is zero.

As a frivolous example, we consider the following matrix bid, where a bidder expresses his preferences for an ice cream. There are two flavors of ice cream (vanilla and banana), and also hot chocolate and strawberry sauce are available.

vanilla ice	4			
banana ice	5	2		
hot chocolate	-5	0	3	
strawberry sauce	-5	0	3	-1

Consider now the value this bidder  $j$  attributes to vanilla ice with hot chocolate. Observe that for this choice of  $S$ , vanilla ice is the highest ranked item (that is,  $k(\text{vanilla ice}, j, S) = 1$ ), and hot chocolate is the second highest ranked item (that is,  $k(\text{hot chocolate}, j, S) = 2$ ). We find using (1):

$$\begin{aligned} b_j(S) &= b_{\text{vanilla ice},j,k(\text{vanilla ice},j,S)} + b_{\text{hot chocolate},j,k(\text{hot chocolate},j,S)} \\ &= b_{1,j,1} + b_{3,j,2} \\ &= 4 + 0 = 4. \end{aligned}$$

Thus, this matrix bid can be interpreted as follows: bidder  $j$  feels that he needs

at least one scoop of ice cream of one of the two available flavors, although he prefers banana. Indeed, no combination without ice cream will result in a positive valuation, because the bidder charges a (disposal) cost of 5 if he gets one or both toppings without ice cream. Furthermore, the bidder is not willing to pay as much for the second scoop of ice cream as for the first. The highest bid this bidder places is 9, for the combination of vanilla and banana ice with any one of the two toppings.

### 3 Motivation

There are three reasons for investigating a combinatorial auction with matrix bids. A main reason, as outlined in the introduction, is to have a compact way of representing preferences. Two other reasons are capturing structure and allowing faster computation. We now explore these arguments in more detail.

First, the matrix bid auction also offers a compact way of representing preferences. Indeed, each bidder only needs to communicate an ordered list of  $m$  items and  $\frac{m(m+1)}{2}$  matrix bid entries, which is far less than bids for each of the  $2^m - 1$  possible sets of items in a general combinatorial auction. We do recognize that choosing a ranking of the items and filling the matrix bid with appropriate values might not be a trivial task for the bidder. However, in section 5.1, we develop a procedure that recognizes whether a given collection of bids can be translated into a matrix bid. If this is not the case, the algorithm in section 5.2 presents a way to approximate this collection of bids by a matrix bid, in a way that does not expose the bidder to paying more than he stated for any set of items.

Second, bids in any practical combinatorial auction are likely to possess some structure. In literature, we find references of both theoretical structures (see e.g. Rothkopf, Pekeç & Harstad (1998), Nisan (2000), and Leyton-Brown & Shoham (2005)) and structures in practice (see e.g. Bleischwitz & Kliwer (2005) and Goossens, Maas, Spieksma & van de Klundert (2007)). Capturing and understanding this structure is important, since it improves our understanding of various properties of an auction. Furthermore, it allows us to develop algorithms that can be more efficient than algorithms for a general combinatorial auction. One of the most challenging problems related to combinatorial auctions is to decide which bidders should get what items. This problem is called the winner determination problem. Sandholm (2002) shows that the winner determination problem cannot be approximated within a ratio of  $\max(K^{\epsilon-1}, m^{\epsilon-1/2})$  in polynomial time for any fixed  $\epsilon > 0$  (unless  $P=ZPP$ ), where  $K$  is the number of bundles on which a bid has been made, and  $m$  is the number of items. However, if bid functions are submodular (see section 4.3), a greedy algorithm produces a  $1/2$ -approximation for this problem (Lehmann, Lehmann & Nisan 2006). The matrix bid auction, where the incremental value an item adds to a bid on a set is determined only by the number of higher ranked items in that set, also imposes a structure. Thus, the matrix bid auction offers a way of capturing structure that may be present in combinatorial auctions. Day (2004) illustrates that this structure encompasses the following six types of bidders.

**Additive preference bidder:** For every item  $i$ , the bidder has a price  $p_i$ . The bidder's valuation for a set  $S$  is then  $\sum_{i \in S} p_i$ .

**Single-minded bidder:** This bidder is interested in one particular set  $S$  for which he is willing to pay a price  $p$ . These single minded bids  $(S, p)$  are also known as flat bids or atomic bids (Nisan 2000).

**Nested flat bidder:** This bidder is a generalization of the single-minded bidder and expresses a chain of  $q$  exclusive single-minded bids  $(S_1, p_1), (S_2, p_2), \dots (S_q, p_q)$  such that  $S_1 \subset S_2 \subset \dots \subset S_q$ .

**Nested k-of bidder:** The  $k$ -of bid function consists of a bid  $(k, S, p)$ , which is a bid of  $p$  on any subset of  $S$  of at least size  $k$ . Multiple  $k$ -of bids  $(k_1, S, p_1), (k_2, S, p_2), \dots, (k_q, S, p_q)$  on the same set  $S$  can be represented in a single matrix bid, provided that all  $k$ -values are different. This bid function is also known as the general symmetric bid function, in which the bidder specifies prices  $p_1, p_2, \dots, p_m$  where  $p_k$  is the price the bidder is willing to pay for the  $k$ -th item won (see Nisan (2000)). The bidder's valuation for a set  $S$  is then  $\sum_{i=1}^{|S|} p_i$ .

**Partition bidder:** This bidder partitions the items into a number of groups of substitutes. The bidder gives a ranking of the groups and prices he is willing to pay for receiving exactly one item from each group, given that exactly one item from each higher ranked group has been received. This bid function can be generalized to accommodate an arbitrary given demand for each group of substitutes.

**Add-on bidder:** This bid function consists of a bid for an essential item, and extra prices the bidder is willing to pay for each number of items from a set of add-on items in which the bidder is interested.

Any auction whose bidders are from these types is a combinatorial auction with matrix bids.

Finally, matrix bid auctions allow for a faster computation due to the restriction on the preferences that is assumed. Despite this restriction, the winner determination problem of the matrix bid auction is  $NP$ -hard (Day 2004). Even more, there exists no polynomial-time approximation scheme for the winner determination problem for the matrix bid auction even when all bidders have an identical ranking of the items, unless  $P = NP$  (Goossens & Spieksma 2006). Nevertheless, Day & Raghavan (2006) solve the winner determination problem of the matrix bid auction using a formulation based on the assignment problem, and compare this with solving the winner determination problem of a general combinatorial auction using a formulation based on the set packing problem. The authors conclude that the computation time for the general combinatorial auction is higher and grows much faster than for the matrix bid auction. Moreover, they manage to solve the winner determination problem for matrix bid auctions with 72 items, 75 bidders and over  $10^{23}$  bids, whereas for the general combinatorial auction, the largest instances that can be solved have 16 items, 25 bidders, and less than  $10^9$  bids. Although this comparison is somewhat distorted

since it does not use a state-of-the-art method (e.g. CABOB, see Sandholm, Suri, Gilpin & Levine (2005)) to solve the winner determination problem for the general auction, it does give an indication of the size of the matrix bid auctions that can be solved in practice. We refer to Goossens & Spieksma (2006) for an extensive discussion on computational issues related to the matrix bid auction.

## 4 Recognizing matrix bid properties

In this section, we discuss the relationship between the bid function implied by a matrix bid and micro-economic concepts as free disposal (4.1), complement freeness (4.2), decreasing marginal valuations (4.3), and the gross substitutes property (4.4). In particular, we show how to verify efficiently whether a given matrix bid satisfies each of these properties. Since this section deals only with the bid function of a single bidder, the index  $j$  that is used to indicate the bidder will be dropped. In literature, many of the economic concepts discussed in this section are in terms of a valuation function. Although for some auctions (e.g. the VCG auction), it has been shown that it is in the bidder's best interest to bid his true valuation, in general, a bidder's bid and his valuation need not be identical. Indeed, strategic considerations may motivate a bidder to express bids that differ considerably from his valuation. Nevertheless, in this section, we ignore this issue and assume equivalence between the notions bid function and valuation function. This is common practice in studies on bidding languages (see e.g. Nisan (2005)). We refer to Gul & Stacchetti (1999) and Cramton, Steinberg & Shoham (2005) for the definitions used in this section.

### 4.1 Free disposal

In micro-economics, it is often assumed that agents prefer more to less. In the context of an auction, this means that a bidder is always willing to receive one or more items for free. Consequently, a seller will never get stuck with unsold items and can therefore dispose of any number of items at no cost. Free disposal is a very common assumption in literature on combinatorial auctions (Lehmann et al. 2006, Nisan 2000) and can be defined as follows.

**Definition 1.** *A bid function  $b$  satisfies the free disposal property if*

$$b(S) \leq b(T) \quad \forall S \subseteq T \subseteq G. \quad (2)$$

Notice that this definition is equivalent with the definition of a monotone non-decreasing function. Alternatively stated, this definition implies that disposing an item from a set cannot increase the value of the resulting set.

In combinatorial auctions where a bidder does not communicate bids on every possible subset of items, but rather only on a limited number of subsets of his liking, allocating all items may be problematic for the auctioneer. In its strictest sense, the lack of free disposal would mean that buyers do not accept anything extra beyond what they bid on. Using this interpretation, even finding a solution to the winner determination problem where all items are allocated is *NP*-complete (Sandholm, Suri, Gilpin & Levine 2002). However, many



other approaches allow the auctioneer to allocate all items, without disposal cost. Nisan (2005) assumes that bids of each bidder satisfy the free disposal property. Moreover, if a bidder did not express a bid on a set  $S$ , the auctioneer can construct a new bid  $b(S)$  equal to the highest bid over all subsets of  $S$ . Obviously, the newly created bids also satisfy the free disposal property. A similar approach is followed by Leyton-Brown, Shoham & Tennenholtz (2000), since they allow the auctioneer to create additional bids with value zero for any subset of items, which can then be combined with any of the bids expressed by the bidders (which also satisfy the free disposal property).

The concept of free disposal is, however, also relevant in a combinatorial auction where bidders do express bids on every possible subset, as in the matrix bid auction. In this case, assuming bid functions that satisfy free disposal guarantees the existence of a total winning bid value maximizing allocation in which all items are awarded to some bidder.

Obviously, not every matrix bid satisfies the free disposal property. For instance, the matrix bid that was used in section 2 does not allow free disposal, since  $b(\text{hot chocolate}) < b(\emptyset)$ . The matrix framework indeed allows the bidder to take into account a disposal cost which can vary across the items and may lead to one or more sets with a negative valuation. However, imposing that each entry in the matrix bid is non-negative is not sufficient to attain the free disposal property. This is illustrated by the matrix bid below, as  $b(y) = 3 > 2 = b(x, y)$ .

$$\begin{array}{c|cc} \text{item } x & 1 & \\ \text{item } y & 3 & 1 \end{array}$$

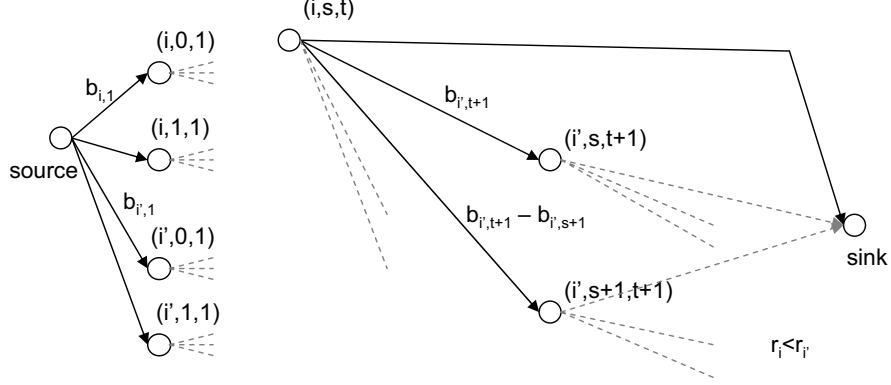
Verifying free disposal can be done in polynomial time for a given matrix bid, as witnessed by the following theorem.

**Theorem 1.** *Verifying whether a matrix bid  $b$  satisfies the free disposal property can be done in polynomial time.*

**Proof.** We will show that solving a shortest path problem on an acyclic graph involving  $O(m^3)$  nodes and  $O(m^4)$  arcs determines whether a matrix bid  $b$  satisfies the free disposal property (2). The graph can be described as follows.

The graph contains a source and a sink, and nodes indexed by  $(i, s, t)$ . The index  $i$  refers to item  $i$  and ranges from 1 to  $m$ . The index  $t$  ranges from 1 to  $r_i$ , while  $s$  ranges from 0 to  $t$ . There are arcs from each node  $(i, s, t)$  to  $(i', s, t+1)$  and to  $(i', s+1, t+1)$ , for all items  $i'$  ranked lower than  $i$  (recall that we consider a single bidder). Furthermore, there is an arc from the source to each node  $(i, 0, 1)$  and  $(i, 1, 1)$ , and there is an arc from each node (except the source) to the sink. Let the cost on the arc from  $(i, s, t)$  to  $(i', s, t+1)$  be equal to  $b_{i', t+1}$  and let the cost on the arc from  $(i, s, t)$  to  $(i', s+1, t+1)$  be equal to  $b_{i', t+1} - b_{i', s+1}$ . Analogously, the arcs from the source to each node  $(i, 0, 1)$  and  $(i, 1, 1)$  have a cost equal to  $b_{i, 1}$  and zero respectively. Arcs to the sink have a cost equal to zero. This completes the description of the graph. Notice that this graph is acyclic and contains a number of nodes and arcs that is polynomial in the number of items ( $O(m^3)$  and  $O(m^4)$  respectively). Figure 1

illustrates this graph; arcs with no indication of their cost have a cost equal to 0.



**Figure 1:** Graph used to verify free disposal

The graph described above should be interpreted as follows. Each node  $(i, s, t)$  corresponds to a state where  $s$  and  $t$  items ranked at least as high as item  $i$  are present in set  $S$  and set  $T \supseteq S$  respectively. Selecting an arc from  $(i, s, t)$  to  $(i', s, t+1)$  corresponds to adding item  $i'$  to set  $T$  as the  $(t+1)$ -th best item, but not to  $S$ , whereas an arc from  $(i, s, t)$  to  $(i', s+1, t+1)$  corresponds to adding item  $i'$  to both set  $S$  and set  $T$ , as the  $(s+1)$ -th and  $(t+1)$ -th best item respectively. In this way, each path from source to sink determines sets  $S$  and  $T$ , and, vice versa, there is a path from the source to the sink for each possible  $S$  and  $T$ . Notice that the arcs are such that  $S$  will always be a subset of  $T$ .

We now show that the length of a path from source to sink in this graph equals  $b(T) - b(S)$ . Each path from source to sink consists of two types of arcs: those arcs that add items  $i$  to the set  $T$  (and not to  $S$ ), and those arcs that add items  $i$  to both  $S$  and  $T$ . The former arcs give rise to the term  $\sum_{i \in T \setminus S} b_{i,k(i,T)}$ , whereas the latter arcs give rise to the term  $\sum_{i \in S \subseteq T} (b_{i,k(i,T)} - b_{i,k(i,S)})$ . Recall from section 2 that  $k(i, A)$  denotes the rank of item  $i$  in the set  $A$ . Thus, the length of the path equals

$$\begin{aligned} \sum_{i \in T \setminus S} b_{i,k(i,T)} + \sum_{i \in S \subseteq T} (b_{i,k(i,T)} - b_{i,k(i,S)}) &= \sum_{i \in T} b_{i,k(i,T)} - \sum_{i \in S} b_{i,k(i,S)} \\ &= b(T) - b(S). \end{aligned}$$

Thus, verifying the free disposal property (see definition 1) for a given matrix bid can be done by solving a shortest path problem in this graph, which takes polynomial time. Concluding, a shortest path in our graph with non-negative length is equivalent to free disposal.  $\square$

## 4.2 Complement freeness

Although the difficulty to deal with complementarity or substitution effects in a bidder's valuation in a classic sequential auction is a major motivation for researching combinatorial auctions in the first place, assuming the absence of complementarities is quite common in micro-economic theory. Lehmann et al. (2006) state that “in most of micro-economic theory, the consumers are assumed to exhibit diminishing marginal utilities”. In their work, they assume that the valuation of a union of disjoint sets is never higher than the sum of the valuations of the individual sets. This notion can be formalized as follows and is also known as subadditivity.

**Definition 2.** A bid function  $b$  is complement free (or subadditive) if

$$b(S \cup T) \leq b(S) + b(T) \quad \forall S, T : S \cap T = \emptyset. \quad (3)$$

Although the winner determination problem for bidders with a complement free bid function remains  $NP$ -hard (Lehmann et al. 2006), there exists a polynomial time algorithm that finds a  $O(\frac{1}{\log m})$ -approximation if, given external prices for all items, a bidder can determine in polynomial time for which set his valuation  $b$  exceeds the sum of the prices of the items in that set the most (Dobzinski, Nisan & Schapira 2005). If a bidder can only determine his valuation for a given set in polynomial time, then the approximation ratio decreases to  $O(m^{-1/2})$  (Dobzinski et al. 2005).

A sufficient condition to have a complement free matrix bid is that the  $b_{ik}$  values are non-increasing in the rows (i.e.  $b_{ik} \leq b_{i,k+1} \forall i, k$ ). Indeed, for each item  $i$ , the valuation of the union of two sets will only make use of  $b_{ik}$  values with a value of index  $k$  at least as high as the value used in the valuation of the individual sets. Having non-increasing  $b_{ik}$  values is not a necessary condition though, as is illustrated by the following example of a complement free bid function.

item $x$	0		
item $y$	2	1	
item $z$	2	1	2

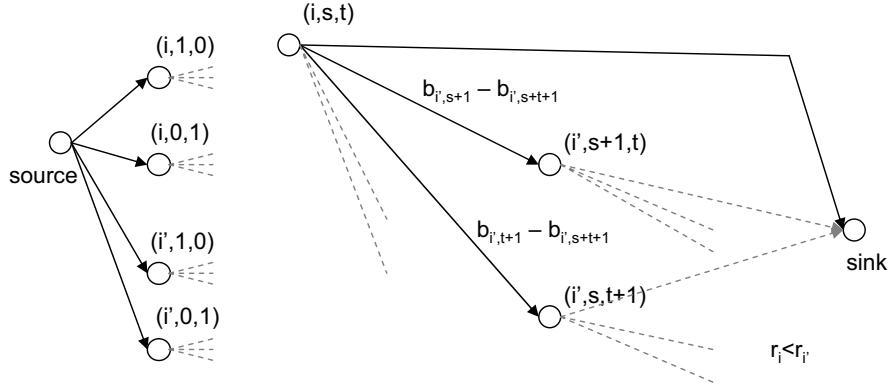
We now show how we can verify in polynomial time whether a matrix bid is complement free.

**Theorem 2.** Verifying whether a matrix bid  $b$  satisfies the complement free property can be done in polynomial time.

**Proof.** Given definition 2, it suffices to establish the existence of sets of items  $S$  and  $T$ , such that  $S \cap T = \emptyset$  and  $b(S) + b(T) - b(S \cup T) < 0$ , to find out whether a matrix bid  $b$  is not complement free. We show that this can be done by solving a shortest path problem, by adapting the construction described in Theorem 1.

Consider a graph containing a source, a sink, and nodes, indexed by  $(i, s, t)$ . The index  $i$  refers to item  $i$ , with  $1 \leq i \leq m$ . Both indices  $s$  and  $t$  range from 0 to  $r_i$ , insofar  $0 < s + t \leq r_i$ . There are arcs from each node  $(i, s, t)$  to  $(i', s + 1, t)$

and to  $(i', s, t + 1)$ , for all items  $i'$  ranked lower than  $i$ . Furthermore, there is an arc from the source to each node  $(i, 1, 0)$  and  $(i, 0, 1)$ , and there is an arc from each node (except the source) to the sink. Let the cost of the arc from  $(i, s, t)$  to  $(i', s + 1, t)$  be equal to  $b_{i', s+1} - b_{i', s+t+1}$  and let the cost of the arc to  $(i', s, t + 1)$  be equal to  $b_{i', t+1} - b_{i', s+t+1}$ . The arcs from the source to each node  $(i, 1, 0)$  and  $(i, 0, 1)$ , and also all arcs to the sink have a cost equal to zero. Notice that this graph is acyclic. Figure 2 illustrates this graph; arcs with no indication of their cost have a cost equal to 0.



**Figure 2:** Graph used to verify complement freeness

Each node  $(i, s, t)$  in the graph represents the state where  $s$  and  $t$  items ranked at least as high as item  $i$  are present in sets  $S$  and  $T$  respectively. Each path from source to sink determines what items are to be added to sets  $S$  and  $T$ , and the arcs are such that these sets will always be disjoint. If such a path contains an arc from  $(i, s, t)$  to  $(i', s + 1, t)$ , this corresponds to item  $i'$  being present in set  $S$  as the  $(s + 1)$ -th best item. If an arc from  $(i, s, t)$  to  $(i', s, t + 1)$  is present in the path, this means that item  $i'$  is in set  $T$  as the  $(t + 1)$ -th best item. We now show that the length of a path from source to sink in this graph equals  $b(S) + b(T) - b(S \cup T)$ . Each path from source to sink consists of two types of arcs: those arcs that add item an item  $i$  to the set  $S$ , and those arcs that add an item  $i$  to  $T$ . Given the choice of the costs of the arcs, the length of a path equals

$$\begin{aligned}
& \sum_{i \in S} (b_{i,k(i,S)} - b_{i,k(i,S \cup T)}) + \sum_{i \in T} (b_{i,k(i,T)} - b_{i,k(i,S \cup T)}) \\
&= \sum_{i \in S} b_{i,k(i,S)} + \sum_{i \in T} b_{i,k(i,T)} - \sum_{i \in S \cup T} b_{i,k(i,S \cup T)} \\
&= b(S) + b(T) - b(S \cup T).
\end{aligned}$$

Thus, if a shortest path in this graph has a non-negative length, then the matrix bid  $b$  is complement free, and vice versa. Since the graph is acyclic and contains

a number of nodes and arcs that is polynomial in the number of items ( $O(m^3)$  and  $O(m^4)$  respectively), verifying whether a matrix bid is complement free can be done in polynomial time.  $\square$

Complement free valuations find their natural counterpart in substitute free valuations, for which  $b(S \cup T) \geq b(S) + b(T)$  for all disjoint sets  $S$  and  $T$ . This property is also known as superadditivity. In this case, there is no need to prevent a bidder from winning multiple bids. Furthermore, the auctioneer only needs to take into account the highest bid for each set.

Notice that having  $b_{ik}$  values that are non-decreasing in the rows is a sufficient condition for a matrix bid to be superadditive. Non-decreasing  $b_{ik}$  values in the rows is, however, not a necessary condition, which is illustrated by the matrix bid below.

item $x$	0		
item $y$	1	2	
item $z$	2	3	2

Verifying whether a matrix bid satisfies the substitute free property can also be done in polynomial time. Using the same graph as in theorem 2, a longest path with non-positive length implies superadditivity of the given matrix bid. Since this graph has no cycles, its longest path can be found in polynomial time.

**Corollary 1.** *Verifying whether a matrix bid  $b$  satisfies the substitute free property can be done in polynomial time.*

### 4.3 Decreasing marginal valuations

In many practical applications, and also in most of micro-economic theory, it is assumed that the more items an agent has, the less he values an extra item. This concept is called decreasing marginal valuations.

**Definition 3.** *A bid function  $b$  has decreasing marginal valuations if*

$$b(T \cup \{x\}) - b(T) \leq b(S \cup \{x\}) - b(S) \quad \forall S \subseteq T, x \in G. \quad (4)$$

Moulin (1988) showed that a (bid) function has decreasing marginal valuations if and only if it is submodular.

**Definition 4.** *A bid function  $b$  is submodular if*

$$b(S \cup T) + b(S \cap T) \leq b(S) + b(T) \quad \forall S, T \subseteq G. \quad (5)$$

Lehmann et al. (2006) show that a valuation where the items have decreasing marginal valuations is also complement free (assuming that this valuation function satisfies free disposal and normalization). The authors also provide an example that illustrates that the converse is not true. Indeed, a complement free valuation function may still have so-called hidden complementarities. When we consider a bidder's valuation for a set of items  $A$ , given that this bidder already

acquired some set of items  $W$  ( $W \cap A = \emptyset$ ), complementarities may still arise. In other words, even if a bid function  $b$  is complement free, this is not necessarily the case for marginal bids  $b'$ , defined by  $b'(A) = b(A|W) = b(A \cup W) - b(W)$ . If we want to enforce that the marginal bids are complement free as well, the bid function  $b$  is required to be submodular (Lehmann et al. 2006). This phenomenon may play a role in an auction with multiple rounds.

Lehmann et al. (2006) also show that the winner determination problem for bidders with a submodular bid function remains *NP*-hard, but that a greedy algorithm produces a  $1/2$ -approximation. This algorithm simply assigns the items one by one (in no particular order) to the bidder with the highest marginal value for that item, given the other items that bidder already acquired. The following example with two items ( $x$  and  $y$ ) and two bidders ( $A$  and  $B$ ) shows that this approximation is tight, even for submodular matrix bids.

bidder $A$			bidder $B$		
item $x$		1	item $x$		1
item $y$		1 0	item $y$		0 0

The optimal total winning bid value for the auctioneer in this auction is 2, by allocating item  $x$  to bidder  $B$  and item  $y$  to bidder  $A$ . However, the greedy algorithm can generate a total winning bid value of 1, by starting with allocating item  $x$  arbitrarily to bidder  $A$  and ending up with marginal bids of zero for item  $y$ . Khot, Lipton, Markakis & Mehta (2005) show that if each bidder can determine his valuation for a given set in polynomial time and if this valuation function is submodular, it is *NP*-hard to approximate the optimal solution by a factor better than  $\frac{e-1}{e}$ . This result assumes that, given external prices for all items, bidders cannot determine in polynomial time (in the number of items and bidders) for which set their valuation exceeds the sum of the prices of the items in that set the most.

Day (2004) suggests that a matrix bid with  $b_{ik}$  values that are non-increasing in both the rows and the columns represents a bid function with decreasing marginal valuations. Notice from the example below that this is not necessarily the case (with  $S = \{x, z\}$  and  $T = \{y, z\}$ ).

item $x$		7		
item $y$		6	5	
item $z$		5	1	0

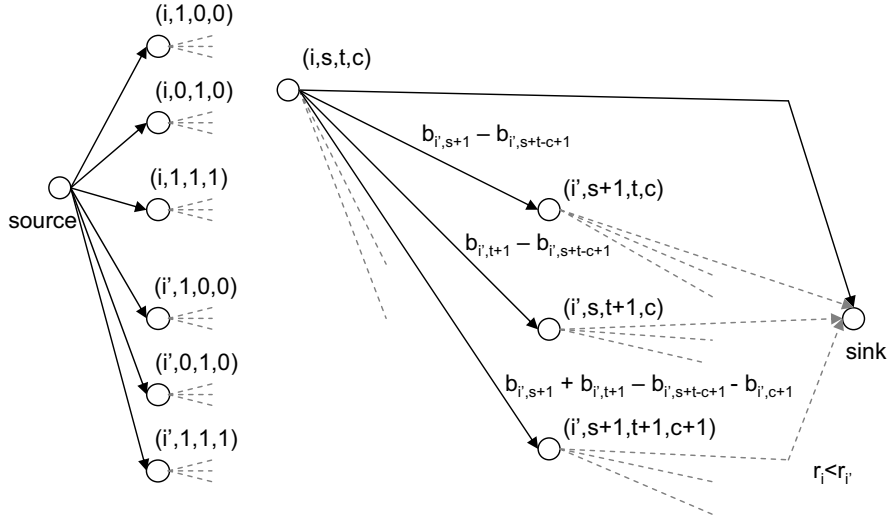
Furthermore, not all bid functions with decreasing marginal valuations can be represented as a matrix bid, as can be easily verified for the submodular bid function  $b$  that produces the following bids on each subset of the item set  $\{x, y, z\}$ :  $b(\{\}) = 0$ ,  $b(\{x\}) = 1$ ,  $b(\{y\}) = 2$ ,  $b(\{z\}) = 3$ ,  $b(\{x, y\}) = 3$ ,  $b(\{y, z\}) = 3$ ,  $b(\{x, z\}) = 3$  and  $b(\{x, y, z\}) = 3$ .

We can, however, verify whether a matrix bid represents a valuation function with decreasing marginal valuations in polynomial time.

**Theorem 3.** *Verifying whether a matrix bid  $b$  has decreasing marginal valuations can be done in polynomial time.*

**Proof.** Using the equivalence result by Moulin (1988), it is sufficient to establish the existence of sets of items  $S$  and  $T$ , for which  $b(S) + b(T) - b(S \cup T) - b(S \cap T) < 0$ , in order to find out whether a matrix bid is not submodular. We show that the existence of such sets can be verified by solving a shortest path problem, again by adapting the construction described in Theorem 1.

Consider a graph containing a source, a sink, and nodes, indexed by  $(i, s, t, c)$ . The index  $i$  refers to item  $i$ , with  $1 \leq i \leq m$ . The indices  $s, t$  and  $c$  range from 0 to  $r_i$ , though no nodes are needed if both  $s$  and  $t$  are 0. There are arcs from each node  $(i, s, t, c)$  to  $(i', s+1, t, c)$ ,  $(i', s, t+1, c)$  and  $(i', s+1, t+1, c+1)$ , for all items  $i'$  ranked lower than  $i$ . Furthermore, there are arcs from the source to each node  $(i, 1, 0, 0)$ ,  $(i, 0, 1, 0)$  and  $(i, 1, 1, 1)$ , and there are arcs from each node but the source to the sink. Let the cost on the arc from  $(i, s, t, c)$  to  $(i', s+1, t, c)$  be equal to  $b_{i', s+1} - b_{i', s+t-c+1}$  and let the cost on the arc to  $(i', s, t+1, c)$  be equal to  $b_{i', t+1} - b_{i', s+t-c+1}$ . The arcs from  $(i, s, t, c)$  to  $(i', s+1, t+1, c+1)$  have a cost equal to  $b_{i', s+1} + b_{i', t+1} - b_{i', s+t-c+1} - b_{i', c+1}$ . The arcs from the source to each node  $(i, 1, 0, 0)$ ,  $(i, 0, 1, 0)$  and  $(i, 1, 1, 1)$ , and also all arcs to the sink have a cost equal to zero. Figure 3 illustrates this acyclic graph; arcs with no indication of their cost have a cost equal to 0.



**Figure 3:** Graph used to verify decreasing marginal valuations

The graph should be interpreted as follows. Each node  $(i, s, t, c)$  stands for a state where  $s, t$ , and  $c$  items ranked at least as high as item  $i$  that are in sets  $S, T$ , and  $S \cap T$  respectively. Each path from source to sink determines what items are to be added to sets  $S$  and  $T$ , and there is a path from source to sink for each possible  $S$  and  $T$ . If the arc from  $(i, s, t, c)$  to  $(i', s+1, t, c)$  is included in the path, this means that item  $i'$  is in set  $S$  as the  $(s+1)$ -th best item, whereas the arc to  $(i', s, t+1, c)$  corresponds to item  $i'$  being in set  $T$ , as the  $(t+1)$ -th best item. The arc from  $(i, s, t, c)$  to  $(i', s+1, t+1, c+1)$  corresponds to adding

item  $i'$  to both  $S$  and  $T$ , where  $c + 1$  is the number of items in  $S \cap T$  ranked at least as high as  $i'$ .

We now show that the length of a path from source to sink corresponds to  $b(S) + b(T) - b(S \cup T) - b(S \cap T)$ . Each path consists of three types of arcs: those arcs that add an item  $i$  to  $S$  and not to  $T$ , those that add an item  $i$  to  $T$  and not to  $S$ , and those that add an item  $i$  to both  $S$  and  $T$ . Given the choice of the costs of these arcs, the length of the path equals

$$\begin{aligned}
& \sum_{i \in S \setminus T} (b_{i,k(i,S)} - b_{i,k(i,S \cup T)}) + \sum_{i \in T \setminus S} (b_{i,k(i,T)} - b_{i,k(i,S \cup T)}) \\
& + \sum_{i \in S \cap T} (b_{i,k(i,S)} + b_{i,k(i,T)} - b_{i,k(i,S \cup T)} - b_{i,k(i,S \cap T)}) \\
& = \sum_{i \in S} b_{i,k(i,S)} + \sum_{i \in T} b_{i,k(i,T)} - \sum_{i \in S \cup T} b_{i,k(i,S \cup T)} - \sum_{i \in S \cap T} b_{i,k(i,S \cap T)} \\
& = b(S) + b(T) - b(S \cup T) - b(S \cap T).
\end{aligned}$$

Thus, if the shortest path in this graph has a non-negative length, then the matrix bid has decreasing marginal valuations, and vice versa. Since the graph contains a number of nodes and arcs that is polynomial in the number of items ( $O(m^4)$  and  $O(m^5)$  respectively), it is clear that verifying whether a matrix bid  $b$  has decreasing marginal valuations can be done in polynomial time.  $\square$

If a bid function  $b$  satisfies the property that  $b(S \cup T) + b(S \cap T) \geq b(S) + b(T)$  for all sets  $S$  and  $T$ , we call  $b$  supermodular, or, equivalently,  $b$  is said to have increasing marginal valuations. It is pointed out by de Vries & Vohra (2003) that if there are only two bid functions a bidder can have, both of them non-decreasing, integer valued and supermodular, then the corresponding winner determination problem can be solved in polynomial time.

Notice that having  $b_{ik}$  values that are non-decreasing in the rows is no sufficient condition for a matrix bid to be supermodular. Choosing  $S = \{x, z\}$  and  $T = \{y, z\}$  in the matrix bid below illustrates this.

item $x$	1		
item $y$	1	2	
item $z$	0	5	6

The same graph as described in theorem 3 can be used to verify whether a matrix bid is supermodular. Indeed, a longest path from source to sink with a non-positive length implies that the matrix bid is supermodular. Since the graph is acyclic, its longest path can be found in polynomial time.

**Corollary 2.** *Verifying whether a matrix bid  $b$  has increasing marginal valuations can be done in polynomial time.*



## 4.4 Gross substitutes property

The gross substitutes property was introduced by Kelso & Crawford (1982) in the context of a labor market, and applied to auctions by e.g. Bevia, Quinzii & Silva (1999), and Bikhchandani & Mamer (1997). The gross substitutes property departs from a price vector  $p$  containing prices that are to be paid for each item  $i$ . Given a valuation function  $b$ , we can define the demand set  $D(p)$  of the corresponding bidder given the current price vector  $p$  as

$$D(p) = \{\operatorname{argmax}_{S \subseteq G} (b(S) - \sum_{i \in S} p_i)\}. \quad (6)$$

The gross substitutes property requires that a bidder will continue to demand items for which the price did not rise, when other items have become more expensive. This condition can be defined more formally as follows.

**Definition 5.** *A bid function  $b$  satisfies the gross substitutes property if for all price vectors  $p \leq q$  (according to a point-wise comparison) and all sets  $S \in D(p)$ , there exists a set  $T \in D(q)$  such that  $\{i \in S : p_i = q_i\} \subseteq T$ .*

The gross substitutes property is stronger than submodularity, since Gul & Stacchetti (1999) show that each bid function that satisfies the gross substitutes condition is also submodular, whereas the converse is not true. Gul & Stacchetti (1999) also prove that in an auction where each bidder has a bid function that satisfies the gross substitutes property, there exists a price vector and an allocation such that every bidder receives a set of items that is in his demand set given these prices. This situation is known as a Walrasian equilibrium. Kelso & Crawford (1982) develop a fully polynomial approximation scheme for finding this Walrasian equilibrium (see also Nisan & Segal (2006)).

Assuming that a bid function  $b$  has the gross substitutes property leads to a number of interesting results. Indeed, the LP-relaxation of a set packing formulation for the winner determination problem of a combinatorial auction where all bidders have bid functions that satisfy the gross substitutes condition has an integral solution (Kelso & Crawford 1982, Bikhchandani, de Vries, Schummer & Vohra 2002). Furthermore, Murota & Tamura (2003) and Fujishige & Yang (2003) show that given a valuation function that satisfies the gross substitutes property and a price vector, the bidder's demand set can be computed in polynomial time. Using the equivalence of separation and optimization (Grötschel, Lovász & Schrijver 1981), it follows that in this setting, the winner determination problem can be solved in polynomial time. Ausubel & Milgrom (2005) show that if every bidder has gross substitutes bid functions, the Vickrey-Clarke-Groves auction does not suffer from a number weaknesses as e.g. vulnerability to the use of multiple bidding identities by a single bidder.

The fact that the definition of the gross substitutes property is based on prices that should be paid for each item is somewhat awkward, since a matrix bid specifies bids, and the price for a set of items is simply the winning bid for that set. However, Reijnierse, Potters & van Gellekom (2002) formulated the following equivalent characterization of the gross substitutes property, which is

independent of prices. A bid function  $b$  satisfies the gross substitutes property if for all  $S \subseteq G$  and  $x, y, z \in G$  the following conditions hold:

$$b(S \cup \{x, y\}) - b(S \cup \{x\}) \leq b(S \cup \{y\}) - b(S), \text{ and} \quad (7)$$

$$\begin{aligned} & b(S \cup \{x, y\}) + b(S \cup \{z\}) \\ & \leq \max(b(S \cup \{x, z\}) + b(S \cup \{y\}), b(S \cup \{y, z\}) + b(S \cup \{x\})). \end{aligned} \quad (8)$$

Reijnierse et al. (2002) also show that it can be checked whether a bid function satisfies the gross substitutes property in a time complexity of  $O(K \log^3(K))$  where  $K$  is the number of subsets of  $G$ , which equals  $2^m$ . For matrix bids, however, this can be done in a time which is polynomial in the number of items, as witnessed by the following theorem.

**Theorem 4.** *Verifying whether a matrix bid  $b$  satisfies the gross substitutes property can be done in polynomial time.*

**Proof.** From the equivalence result by Reijnierse et al. (2002) it follows that a matrix bid  $b$  has the gross substitutes property if and only if conditions (7) and (8) are satisfied. Moulin (1988) shows that condition (7) is in fact equivalent with submodularity. According to theorem 3, checking whether a matrix bid  $b$  is submodular can be done in polynomial time. As for condition (8), it suffices to consider the setting where  $x, y, z \notin S$ . Indeed, the matrix bid auction is a single-unit auction, implying  $b(S \cup \{x\}) = b(S)$  if  $x \in S$ . Using this, it is trivial to see that condition (8) where any subset of  $\{x, y, z\}$  is in  $S$ , is satisfied for any submodular matrix bid. In the remainder of this proof, we will show that solving a shortest path problem on an acyclic graph suffices to check whether a matrix bid  $b$  satisfies condition (8) with  $x, y, z \notin S$ . We will first assume that  $r_x < r_y < r_z$  (i.e. item  $x$  is ranked higher than item  $y$ , which is ranked higher than item  $z$ ) and afterwards we will discuss the other settings.

First, we show that for the setting with  $r_x < r_y < r_z$ , the following is true:

$$b(S \cup \{x, z\}) + b(S \cup \{y\}) = b(S \cup \{y, z\}) + b(S \cup \{x\}). \quad (9)$$

Indeed, all items  $i$  in  $S$  that are ranked higher than item  $x$  contribute  $b(i, k(i, S))$  to each of the four terms in equation (9), where  $k(i, S)$  is the rank  $i$  has amongst the items in  $S$ . Item  $x$  adds  $b(x, k(x, S \cup \{x, z\}))$  to the left-hand side of equation (9), which equals  $b(x, k(x, S \cup \{x\}))$ , added to the right-hand side. Items  $i \in S$  that are ranked between  $x$  and  $y$  contribute  $b(i, k(i, S)) + b(i, k(i, S) + 1)$  to both sides of the equation. Items  $i \in S$  ranked between  $y$  and  $z$  contribute  $2b(i, k(i, S) + 1)$  to both the left-hand and right-hand side of equation (9). Furthermore, also item  $y$  adds equal amounts to both sides of the equation, namely  $b(y, k(y, S \cup \{y\}))$  and  $b(y, k(y, S \cup \{y, z\}))$ . The same goes for item  $z$ , adding the equal terms  $b(z, k(z, S \cup \{x, z\}))$  and  $b(z, k(z, S \cup \{y, z\}))$  to the left-hand and the right-hand side respectively. Finally, items  $i \in S$  ranked lower than  $z$  add  $b(i, k(i, S) + 2) + b(i, k(i, S) + 1)$  to both sides of the equation. Using this result, condition (8) can be reformulated as

$$b(S \cup \{x, y\}) + b(S \cup \{z\}) \leq b(S \cup \{x, z\}) + b(S \cup \{y\}). \quad (10)$$

**Table 1:** Cost on the arcs from  $(i, s, q)$  to  $(i', s + 1, q)$  and to  $(i', s, q + 1)$ , depending on the value for  $q$ .

	$(i, s, q) \rightarrow (i', s + 1, q)$	$(i, s, q) \rightarrow (i', s, q + 1)$
$q = 0$	$b_{i',s+1} + b_{i',s+1} - b_{i',s+1} - b_{i',s+1} = 0$	$b_{i',s+1} - b_{i',s+1} = 0$
$q = 1$	$b_{i',s+2} + b_{i',s+1} - b_{i',s+2} - b_{i',s+1} = 0$	$b_{i',s+2} - b_{i',s+1}$
$q = 2$	$b_{i',s+3} + b_{i',s+1} - b_{i',s+2} - b_{i',s+2}$	$b_{i',s+1} - b_{i',s+2}$
$q = 3$	$b_{i',s+3} + b_{i',s+2} - b_{i',s+3} - b_{i',s+2} = 0$	(no such arc exists)

Consider a graph containing a source, a sink and nodes, indexed by  $(i, s, q)$ . The index  $i$  refers to item  $i$  and ranges from 1 to  $m$ , whereas  $s$  ranges from 0 to  $r_i$  and  $q$  ranges from 0 to 3. There are arcs from each node  $(i, s, q)$  to  $(i', s + 1, q)$  and to  $(i', s, q + 1)$ , for all items  $i'$  ranked lower than  $i$  and insofar  $q + 1 \leq 3$ . Furthermore, there are arcs from the source to each node  $(i, 1, 0)$  and  $(i, 0, 1)$ , and there are arcs from each node  $(i, s, 3)$  to the sink. Depending on the value for  $q$ , the arc from  $(i, s, q)$  to  $(i', s + 1, q)$  and the arc from  $(i, s, q)$  to  $(i', s, q + 1)$  have costs as presented in table 1. The arcs from the source to each node  $(i, 1, 0)$ , and  $(i, 0, 1)$ , and also all arcs to the sink have a cost equal to zero.

The graph can be interpreted as follows. Each node  $(i, s, q)$  represents a state where  $s$  items ranked at least as high as item  $i$  are in set  $S$ . The index  $q$  keeps track of how many of the items  $x, y$ , and  $z$  are ranked at least as high as item  $i$ , and should be understood as follows.

$$\begin{aligned}
q = 0 : & \quad r_x > r_i \\
q = 1 : & \quad r_y > r_i > r_x \\
q = 2 : & \quad r_z > r_i > r_y \\
q = 3 : & \quad r_i > r_z
\end{aligned}$$

Each path from source to sink determines which items are to be added to set  $S$  and which items are selected to play the role of  $x, y$  and  $z$ . If the path contains an arc from  $(i, s, q)$  or from the source to  $(i', s + 1, q)$ , this indicates that item  $i'$  is added to the set  $S$ , as the  $(s + 1)$ -th highest ranked item. An arc from  $(i, s, q)$  or from the source to  $(i', s, q + 1)$  indicates that item  $i'$  is selected as item  $x, y$  or  $z$ , for  $q = 0, q = 1$ , or  $q = 2$  respectively. The fact that there are only arcs from nodes  $(i, s, 3)$  to the sink, guarantees that items  $x, y$  and  $z$  are selected in each path from the source to the sink.

We now show that the length of a path from source to sink in this graph equals  $b(S \cup \{x, y\}) + b(S \cup \{z\}) - b(S \cup \{x, z\}) - b(S \cup \{y\})$ . Each path from source to sink has exactly one arc that selects an item  $x$  (namely the arc where  $q$  increases from 0 to 1), one arc that selects an item  $y$  ( $q$  increases from 1 to 2), and one arc that selects an item  $z$  ( $q$  increases from 2 to 3). The other arcs in the path can be divided into four types: those arcs that add an item  $i$  ranked higher than  $x$  to the set  $S$ , those arcs that add an item  $i$  ranked between  $x$  and  $y$  to  $S$ , those arcs that add an item  $i$  ranked between  $y$  and  $z$  to  $S$ , and those arcs that add an item  $i$  ranked lower than  $z$ . From the costs in table 1, it follows that the length of a path from source to sink equals

$$\begin{aligned}
& \sum_{i \in S: r_x > r_i} (b_{i,k(i,S \cup \{x,y\})} + b_{i,k(i,S \cup \{z\})} - b_{i,k(i,S \cup \{x,z\})} - b_{i,k(i,S \cup \{y\})}) \\
& + \sum_{i \in S: r_y > r_i > r_x} (b_{i,k(i,S \cup \{x,y\})} + b_{i,k(i,S \cup \{z\})} - b_{i,k(i,S \cup \{x,z\})} - b_{i,k(i,S \cup \{y\})}) \\
& + \sum_{i \in S: r_z > r_i > r_y} (b_{i,k(i,S \cup \{x,y\})} + b_{i,k(i,S \cup \{z\})} - b_{i,k(i,S \cup \{x,z\})} - b_{i,k(i,S \cup \{y\})}) \\
& + \sum_{i \in S: r_i > r_z} (b_{i,k(i,S \cup \{x,y\})} + b_{i,k(i,S \cup \{z\})} - b_{i,k(i,S \cup \{x,z\})} - b_{i,k(i,S \cup \{y\})}) \\
& + b_{x,k(x,S \cup \{x,y\})} - b_{x,k(x,S \cup \{x,z\})} + b_{y,k(y,S \cup \{x,y\})} - b_{y,k(y,S \cup \{y\})} \\
& + b_{z,k(z,S \cup \{z\})} - b_{z,k(z,S \cup \{x,z\})} \\
& = \sum_{i \in S \cup \{x,y\}} b_{i,k(i,S \cup \{x,y\})} + \sum_{i \in S \cup \{z\}} b_{i,k(i,S \cup \{z\})} \\
& - \sum_{i \in S \cup \{x,z\}} b_{i,k(i,S \cup \{x,z\})} - \sum_{i \in S \cup \{y\}} b_{i,k(i,S \cup \{y\})} \\
& = b(S \cup \{x,y\}) + b(S \cup \{z\}) - b(S \cup \{x,z\}) - b(S \cup \{y\}).
\end{aligned}$$

Thus, if this graph has a shortest path with non-negative length, then condition (8) is satisfied for every  $S$  and every  $x, y, z \notin S$  such that  $r_x < r_y < r_z$ . Moreover, a similar reasoning can be used to prove that this result is also valid if  $r_y < r_x < r_z$ .

We now show that for the setting with  $r_x < r_z < r_y$ , condition (8) is satisfied for any matrix bid, since

$$b(S \cup \{x,y\}) + b(S \cup \{z\}) = b(S \cup \{y,z\}) + b(S \cup \{x\}). \quad (11)$$

Observe that all items  $i$  in  $S$  that are ranked higher than item  $x$  contribute  $b(i, k(i, S))$  to each of the terms in equation (11). Item  $x$  adds  $b(x, k(x, S \cup \{x,y\}))$  to the left-hand side of equation (11), which equals  $b(x, k(x, S \cup \{x\}))$ , added to the right-hand side. Items  $i \in S$  that are ranked between  $x$  and  $z$  contribute  $b(i, k(i, S)) + b(i, k(i, S) + 1)$  to both sides of the equation. Also item  $z$  adds an equal term to both sides of the equation, namely  $b(z, k(z, S \cup \{z\}))$  and  $b(z, k(z, S \cup \{y,z\}))$  to the left-hand and the right-hand side respectively. Items  $i \in S$  ranked between  $z$  and  $y$  contribute  $2b(i, k(i, S) + 1)$  to both the left-hand and right-hand side of equation (11). Furthermore, also item  $y$  adds equal amounts to both sides of the equation, namely  $b(y, k(y, S \cup \{x,y\}))$  and  $b(y, k(y, S \cup \{y,z\}))$ . Finally, items  $i \in S$  ranked lower than  $y$  add  $b(i, k(i, S) + 2) + b(i, k(i, S) + 1)$  to both sides of the equation. A similar reasoning can be used to show that equality (11) is also valid for a setting with  $r_z < r_x < r_y$ . Obviously, condition (8) is satisfied for any matrix bid that satisfies equality (11).

An analogous proof can be developed to show that condition (8) is also always satisfied for the settings with  $r_y < r_z < r_x$  or  $r_z < r_y < r_x$ , since

$$b(S \cup \{x, y\}) + b(S \cup \{z\}) = b(S \cup \{x, z\}) + b(S \cup \{y\}). \quad (12)$$

We can conclude that it can be verified in polynomial time whether a matrix bid satisfies conditions (7) and (8) by solving a shortest path problem on two graphs, with a number of nodes and arcs that is polynomial in the number of items ( $O(m^4)$  and  $O(m^5)$ , and  $O(m^2)$  and  $O(m^3)$  respectively). If the shortest paths in both graphs have a non-negative length, then the matrix bid has the gross substitutes property. Otherwise, if one of the shortest paths has a negative length, the matrix bid does not have the gross substitutes property.  $\square$

## 5 Expressing arbitrary bids as a matrix bid

The matrix bid auction fits in the scope of the restricted preference approach, which accepts that bidders cannot fully express their preferences in order to solve the winner determination problem more efficiently. In section 5.1, we describe an algorithm that answers the question whether a given collection of bids can be represented in a matrix bid. If the answer is no, we need to find a matrix bid that offers a good approximation of the bids. This is what is done in section 5.2, by generalizing the algorithm of section 5.1.

### 5.1 Exact expression of a bid function using a matrix bid

From the description in section 2, it is clear that not every bid function can be represented by a matrix bid. This is due to the fact that there is in general only one entry available to express the added value of an item to many sets. Indeed, whereas there is only one set corresponding to the entries on the first column and to those on the diagonal, this is not the case for the other entries. Day (2004) shows that for each entry  $b_{ik}$ , the number of sets containing  $k - 1$  items ranked higher than  $i$  corresponds to Pascal's triangle<sup>2</sup>, which implies that the entries towards the middle and the bottom depend on the largest number of sets (notice that we drop the index  $j$  again, since this section deals only with the bid function of a single bidder).

As mentioned in section 2, an arbitrary bid function can be represented as a matrix bid if and only if the incremental value an item adds to a set is determined only by the number of higher ranked items in that set. Although a list of bidder types that can be represented in a matrix bid (see section 3) gives some insight in the expressiveness of the matrix bid language, a test to determine whether a given collection of bids can fit into a matrix bid would be useful as well. Indeed, in practice, bidders do not necessarily bid according to a well-known structure, but often simply express a number of bids on various sets of their interest.

Consider a given set of items  $G$ , with  $|G| = m$ , and bids  $v(S)$  with  $S \subseteq G$ . A naive way of verifying whether the given bids can fit into a matrix bid involves checking every possible ranking of the items; this has a time complexity

<sup>2</sup>see <http://mathworld.wolfram.com/PascalsTriangle.html>

of  $O(m!2^m)$ . The following procedure  $\text{FIT}(p, G, v)$  tries to fill the first  $p$  rows of the matrix bid with items in  $G$ , such that the bids that follow from this matrix bid correspond with the given bids  $v$ . In other words, if such a matrix bid exists,  $\text{FIT}(m, G, v)$  returns YES and produces the ranking of the items and entries  $b_{ik}$ . Otherwise,  $\text{FIT}(m, G, v)$  returns NO, indicating that the given bids cannot be represented in a matrix bid.

**Theorem 5.**  *$\text{FIT}(m, G, v)$  recognizes whether a bid function  $b$ , defined on subsets of a set  $G$ , containing  $m$  items, can be expressed as a matrix bid.*

**Proof.** The algorithm  $\text{FIT}(p, G, v)$  searches for items in  $G$  that can be placed on the  $p$ -th row of the matrix bid. To keep track of these items, the algorithm uses the set  $A$ , which is initialized to the empty set in step 1. Furthermore, the algorithm initializes the set  $L$  of items that still need to be assigned to a row  $p$ . In step 2, an item is picked from  $L$ . In step 3, we verify whether a value can be found for an entry  $k$  on row  $p$  that satisfies the requirements that follow from the given bids  $v$ . By repeating step 3 for every entry of row  $p$ , we find out whether item  $i$  can be placed on row  $p$ ; if this is the case, we add it to  $A$ . From the way a bid is computed from a matrix bid (see 1), it follows that the key condition is indeed:

$$\exists b_{ik} \in \mathbb{R} : v(S \cup \{i\}) - v(S) = b_{ik}, \forall S \subseteq G \setminus \{i\} : |S| = k - 1, \forall k \in \{1, \dots, p\}?$$

This condition is checked for all items that have not yet been placed on a row of the matrix bid. If we find a non-empty set  $A$  in step 4, we can use the items in  $A$  to fill rows  $p - |A| + 1$  to  $p$ , with the  $b_{ik}$  values that are obtained in step 3. Indeed, if an item can be placed on row  $p$ , it can also be placed on a higher ranked row (i.e.  $p - 1, p - 2$ , etc.) since the sets for which the condition in step 3 needs to be valid on such a row are a subset of the sets that need to satisfy the condition for row  $p$ . Notice that the choice of which item of  $A$  should be placed on which row from  $p - |A| + 1$  to  $p$  has no effect on choices made regarding rows lower than  $p$  or higher than  $p - |A| + 1$ . At this point, we perform the FIT algorithm again, in order to place the remaining  $G \setminus A$  items on the remaining  $p - |A|$  rows. If we can continue the algorithm until all rows have been filled with an item, then clearly the given bids can be represented by a matrix bid.

If at some stage in step 4, the set  $A$  is empty, then we conclude that there is no way to represent the bids in a matrix bid. Indeed, this means that we found at some row  $q$  a set  $Q$  of items (with  $|Q| = q$ ) that cannot be placed at that row  $q$ . We also know that none of these items can be placed at a row ranked lower than  $q$  (i.e.  $q + 1, q + 2$ , etc.), since otherwise this would have been clear at an earlier stage of the algorithm. Therefore, we can conclude that if the algorithm returns NO, no matrix bid exists that represents the given bids.  $\square$

---

**Algorithm 1** FIT( $p, G, v$ )

---

STEP 1:

$L \leftarrow G; A \leftarrow \{\};$   
Go to STEP 2;

STEP 2:

Pick an item  $i$  from  $L$ ;  
 $L \leftarrow L \setminus \{i\}; k \leftarrow 1;$   
Go to STEP 3;

STEP 3:

$\exists b_{ik} \in \mathbb{R} : v(S \cup \{i\}) - v(S) = b_{ik}$  for all sets  $S \subseteq G \setminus \{i\}$  with  $|S| = k - 1$ ?

**if YES then**

**if** ( $k < p$ ) **then**

$k \leftarrow k + 1;$   
        Go to STEP 3;

**else**

$A \leftarrow A \cup \{i\};$

**if** ( $L \neq \{\}$ ) **then**

            Go to STEP 2;

**else**

            Go to STEP 4;

**end if**

**end if**

**end if**

**if NO then**

**if** ( $L \neq \{\}$ ) **then**

        Go to STEP 2;

**else**

        Go to STEP 4;

**end if**

**end if**

STEP 4:

**if** ( $A = \{\}$ ) **then**

    return NO;

**else**

**for** ( $i \in A$ ) **do**

$r_i = p;$

        Fill row  $p$  with  $b_{ik}$  with  $1 \leq k \leq p;$

$p \leftarrow p - 1;$

**end for**

**if** ( $p = 0$ ) **then**

        return YES;

**else**

        FIT( $p, G \setminus A, v$ );

**end if**

**end if**

---

In order to verify whether there exists an entry for item  $i$  on column  $k$  of row  $p$ , we need to check all sets of size  $k-1$  that do not contain item  $i$ . Thus, verifying whether an item fits on a row requires processing at most  $2^m$  sets. As a result, this algorithm has a time complexity of  $O(m^2 2^m)$ , which is exponential in the number of items, but polynomial in the input, being the number of bids, i.e.  $2^m$ .

## 5.2 Approximate expression of a bid function using a matrix bid

We now focus on the question how we can approximate a given collection of bids by a matrix bid. Day (2004) points out that such an approximation should ideally be safe and effective. An approximation is safe if it does not expose the bidder to paying more than he is willing to for any set. An approximation consisting of entries  $a_{ik}$  is effective if there exists no other safe approximation with entries  $b_{ik}$  such that  $b_{ik} \geq a_{ik}$  for all  $i \in G$  and all  $k \in \{1, \dots, r_i\}$  and  $b_{ik} > a_{ik}$  for at least one  $i \in G$  and  $k \in \{1, \dots, r_i\}$ . Day (2004) also points out that the entries in the first column and on the diagonal can always be filled in exactly, since they correspond to only one set. However, no advice is given on how to choose one of the  $m!$  possible rankings of the items.

The following algorithm APPROX( $m, G, v$ ) is an adaptation of the FIT algorithm, such that if no item is found that fits on some row of the matrix bid, an item  $q$  is selected and a safe approximation is constructed. Indeed, if no item fits on some row, there is for each item at least one entry on that row for which there is no single value that represents the marginal value of that item for all relevant sets. The selected item  $q$  is that item for which the sum of absolute values of the difference between the highest and the lowest marginal value is minimal. In case there are multiple such items, the algorithm makes an arbitrary choice among these items.

---

### Algorithm 2 APPROX( $m, G, v$ )

---

```

for ( $p = m$  downto 1) do
  for ( $i \in G$ ) do
    for ( $k = 1$  to  $p$ ) do
       $high(i, k) \leftarrow \max_{S \subseteq G \setminus \{i\}: |S|=k-1} (v(S \cup \{i\}) - v(S));$ 
       $low(i, k) \leftarrow \min_{S \subseteq G \setminus \{i\}: |S|=k-1} (v(S \cup \{i\}) - v(S));$ 
    end for
     $d(i) \leftarrow \sum_{k=1}^p |high(i, k) - low(i, k)|;$ 
  end for
   $Q \leftarrow \operatorname{argmin}_i d(i);$ 
  Pick an arbitrary  $q \in Q;$ 
   $G \leftarrow G \setminus \{q\};$ 
   $r_q = p;$ 
  for ( $k = 1$  to  $p$ ) do
     $b_{qk} \leftarrow low(q, k);$ 
  end for
end for

```

---



Notice that the notion of a safe approximation is especially important when the matrix bid is expressed by the bidder and communicated to the auctioneer. However, it may also be the case that the bidder does not communicate his bids as a matrix bid, but that the auctioneer translates them to a matrix bid. The auctioneer might do this in order to benefit from the fact that the winner determination problem of a matrix bid auction can be solved more efficiently than that of a general auction (see Day & Raghavan (2006)). If this is the case, the approximation need not necessarily be safe, since the auctioneer could use it only to determine the allocation, and use the original bids to determine the actual prices. In this case, the choice for the average of the highest and the lowest marginal value (or even the highest marginal value) as an approximate entry  $b_{qk}$  in the matrix bid could be motivated as well.

## 6 Conclusion

In this paper, we focussed on the matrix bid auction, which is a combinatorial auction where a restriction is imposed on the preferences that can be expressed by a bidder. We investigated the structure on the bids implied by this auction and revealed the relationship between a matrix bid and concepts like free disposal, complement freeness, decreasing marginal valuations, and gross substitutes. Finally, we developed tools that should facilitate the use of the matrix bid auction in practice. Given a number of bids, we found a way to determine whether these bids can be expressed as a matrix bid. Finally, we also show how to approximate these bids in a matrix bid if an exact representation is not possible.

## References

- Ausubel, L. M. & Milgrom, P. (2005). The lovely but lonely Vickrey auction, in P. Cramton, R. Steinberg & Y. Shoham (eds), *Combinatorial auctions*, MIT Press, pp. 17–40.
- Ball, M. O., Donohue, G. L. & Hoffman, K. (2005). Auctions for the safe, efficient and equitable allocation of airspace system resources, in P. Cramton, R. Steinberg & Y. Shoham (eds), *Combinatorial auctions*, MIT Press, pp. 507–538.
- Bevia, C., Quinzii, M. & Silva, J. A. (1999). Buying several indivisible goods, *Mathematical Social Sciences* **37**(1): 1–23.
- Bikhchandani, S., de Vries, S., Schummer, J. & Vohra, R. (2002). Linear programming and Vickrey auctions, in B. Dietrich & R. Vohra (eds), *Mathematics of the Internet: e-auction and markets*, Springer, pp. 75–116.
- Bikhchandani, S. & Mamer, J. (1997). Competitive equilibrium in an exchange economy with indivisibilities, *Journal of Economic Theory* **74**: 385–413.

- Bleichwitz, Y. & Klierer, G. (2005). Accelerating Vickrey payment computation in combinatorial auctions for an airline alliance., in *WEA '05: Proceedings of the 4th international workshop on experimental and efficient algorithms, Santorini Island, Greece*, Springer, pp. 228–239.
- Cantillon, E. & Pesendorfer, M. (2005). Auctioning bus routes: the London experience, in P. Cramton, R. Steinberg & Y. Shoham (eds), *Combinatorial auctions*, MIT Press, pp. 573–591.
- Caplice, C. & Sheffi, Y. (2005). Combinatorial auctions for truckload transportation, in P. Cramton, R. Steinberg & Y. Shoham (eds), *Combinatorial auctions*, MIT Press, pp. 539–571.
- Cramton, P., Steinberg, R. & Shoham, Y. (2005). *Combinatorial auctions*, MIT Press.
- Day, R. W. (2004). *Expressing preferences with price-vector agents in combinatorial auctions*, PhD thesis, University of Maryland.
- Day, R. W. & Raghavan, S. (2006). Matrix bidding in combinatorial auctions. Manuscript.
- de Vries, S. & Vohra, R. V. (2003). Combinatorial auctions: a survey, *INFORMS Journal on Computing* (3): 284–309.
- Dobzinski, S., Nisan, N. & Schapira, M. (2005). Approximation algorithms for combinatorial auctions with complement-free bidders, in *STOC '05: Proceedings of the 37th annual ACM symposium on theory of computing, Baltimore, USA*, ACM Press, pp. 610–618.
- Epstein, R., Henríquez, L., Catalán, J., Weintraub, G. Y. & Martínez, C. (2002). A combinational auction improves school meals in Chile, *Interfaces* **32**(6): 1–14.
- Eso, M. (2001). An iterative online auction for airline seats, in B. Dietrich & R. Vohra (eds), *Mathematics of the Internet: e-auction and markets*, Springer-Verlag, pp. 45–58.
- Fujishige, S. & Yang, Z. (2003). A note on Kelso and Crawford’s gross substitutes condition, *Mathematics of Operations Research* **28**(3): 463–469.
- Goossens, D., Maas, A. J. T., Spieksma, F. C. R. & van de Klundert, J. J. (2007). Exact algorithms for procurement problems under a total quantity discount structure, *European Journal of Operational Research* **178**(2): 603–626.
- Goossens, D. & Spieksma, F. C. R. (2006). Exact algorithms for the matrix bid auction, *FETEW Research Report KBI-0620*, K.U.Leuven.
- Grötschel, M., Lovász, L. & Schrijver, A. (1981). The ellipsoid method and its consequences in combinatorial optimization, *Combinatorica* **1**(2): 169–197.
- Gul, F. & Stacchetti, E. (1999). Walrasian equilibrium with gross substitutes, *Journal of Economic Theory* **87**(1): 95–124.

- Kelso, Alexander S. J. & Crawford, V. P. (1982). Job matching, coalition formation, and gross substitutes, *Econometrica* **50**(6): 1483–1504.
- Khot, S., Lipton, R. J., Markakis, E. & Mehta, A. (2005). Inapproximability results for combinatorial auctions with submodular utility functions, in *WINE '05: Proceedings of the 1st international workshop on Internet and network economics, Hong Kong, China*, Springer, pp. 92–101.
- Ledyard, J. O., Olson, M., Porter, D., Swanson, J. A. & Torma, D. P. (2002). The first use of a combined-value auction for transportation services, *Interfaces* **32**(5): 4–12.
- Lehmann, B., Lehmann, D. & Nisan, N. (2006). Combinatorial auctions with decreasing marginal utilities, *Games and Economic Behavior* **55**(2): 270–296.
- Leyton-Brown, K. & Shoham, Y. (2005). A test suite for combinatorial auctions, in P. Cramton, R. Steinberg & Y. Shoham (eds), *Combinatorial auctions*, MIT Press, pp. 451–478.
- Leyton-Brown, K., Shoham, Y. & Tennenholtz, M. (2000). An algorithm for multi-unit combinatorial auctions, in *AAAI/IAAI '00: Proceedings of the 17th national conference on artificial intelligence and 12th conference on innovative applications of artificial intelligence, Austin, USA*, AAAI Press / The MIT Press, pp. 56–61.
- Moulin, H. (1988). *Axioms of cooperative decision making*, Cambridge University Press.
- Murota, K. & Tamura, A. (2003). New characterizations of m-convex functions and their applications to economic equilibrium models with indivisibilities, *Discrete Applied Mathematics* **131**(2): 495–512.
- Nisan, N. (2000). Bidding and allocation in combinatorial auctions, in *EC '00: Proceedings of the 2nd ACM conference on electronic commerce, Minneapolis, USA*, ACM Press, pp. 1–12.
- Nisan, N. (2005). Bidding languages for combinatorial auctions, in P. Cramton, R. Steinberg & Y. Shoham (eds), *Combinatorial auctions*, MIT Press, pp. 215–231.
- Nisan, N. & Segal, I. (2006). The communication requirements of efficient allocations and supporting prices, *Journal of Economic Theory* **129**(1): 192–224.
- Rassenti, S. J., Smith, V. L. & Bulfin, R. L. (1982). A combinatorial mechanism for airport time slot allocation, *Bell Journal of Economics* **13**: 402–417.
- Reijnierse, H., Potters, J. A. & van Gellekom, A. (2002). Verifying gross substitutability, *Economic Theory* **20**(4): 767–776.
- Rothkopf, M., Pekeç, A. & Harstad, R. M. (1998). Computationally manageable combinatorial auctions, *Management Science* **44**(8): 1131–1147.
- Sandholm, T. (2002). Algorithm for optimal winner determination in combinatorial auctions, *Artificial Intelligence* **135**(1-2): 1–54.

- Sandholm, T., Suri, S., Gilpin, A. & Levine, D. (2002). Winner determination in combinatorial auction generalizations, in *AAMAS '02: Proceedings of the 1st international joint conference on autonomous agents and multiagent systems, Bologna, Italy*, ACM Press, pp. 69–76.
- Sandholm, T., Suri, S., Gilpin, A. & Levine, D. (2005). CABOB: a fast optimal algorithm for combinatorial auctions, *Management Science* **51**: 374–390.
- Sheffi, Y. (2004). Combinatorial auctions in the procurement of transportation services, *Interfaces* **34**(4): 245–252.