

Winner Determination in Geometrical Combinatorial Auctions

Bart Vangerven^{a,*}, Dries R. Goossens^b, Frits C.R. Spieksma^a

^a*Operations Research and Statistics, Faculty of Economics and Business, KU Leuven,
Naamsestraat 69, BE-3000 Leuven, Belgium*

^b*Faculty of Economics and Business Administration, Ghent University, Tweeckerkenstraat 2,
BE-9000 Ghent, Belgium*

Abstract

We consider auctions of items that can be arranged in rows. Examples of such a setting appear in allocating pieces of land for real estate development, or seats in a theater or stadium. The objective is, given bids on subsets of items, to find a subset of bids that maximizes auction revenue (often referred to as the *winner determination problem*). We describe a dynamic programming algorithm which, for a k -row problem with *connected* and *gap-free* bids, solves the winner determination problem in polynomial time. We study the complexity for bids in a grid, complementing known results in literature. Additionally, we study variants of the geometrical winner determination setting. We provide a NP-hardness proof for the 2-row setting with gap-free bids. Finally, we extend this dynamic programming algorithm to solve the case where bidders submit connected, but not necessarily gap-free bids in a 2-row and a 3-row problem.

Keywords: Auctions, dynamic programming, winner determination problem, complexity, rows

1. Introduction

In combinatorial auctions, bidders can place bids on combinations of items, called packages or bundles. Clearly, combinatorial auctions allow bidders to better express their preferences compared to the traditional auction formats, where bidders place bids on individual items. In particular, it makes sense to use a combinatorial auction when complementarities or substitution effects exist between different items.

Research on combinatorial auctions was triggered by applications such as the FCC spectrum auction (Jackson, 1976) and auctions for airport time slots

*Corresponding author

Email addresses: `bart.vangerven@kuleuven.be` (Bart Vangerven),
`dries.goossens@ugent.be` (Dries R. Goossens), `frits.spieksma@kuleuven.be` (Frits C.R. Spieksma)

(Rassenti et al., 1982). For an introduction to combinatorial auctions, we refer to the book edited by Cramton et al. (2006); for a survey of the literature, we refer to Abrache et al. (2007) and de Vries and Vohra (2003).

One important challenge within this domain is, given the bids, to decide which items should be allocated to which bidder, i.e., which bids to accept. In general, this *winner determination problem* is NP-hard (Van Hoesel and Müller, 2001), and does not allow good approximation results (Sandholm, 2002).

We discuss a combinatorial auction in a restricted topology. In this setting, an item corresponds to a rectangle, and all items are arranged in (a limited number of) rows, see Figure 1 for an example. Notice that the individual items

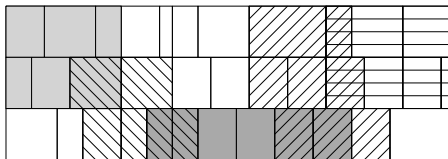


Figure 1: An example of an instance with 3 rows and 5 bids.

(or rectangles) need not have the same size. A bid consists of a set of items satisfying some restrictions (see Section 2 for a precise problem definition), together with a value. The objective is to select a set of bids that maximizes the sum of the expressed values, while making sure that each item is present at most once in a selected bid.

There are several situations in practice that motivate this specific geometric setting. We mention the following:

- Real estate. Goossens et al. (2014) describe how space in a newly erected building, to be used for housing and commercial purposes, is allocated using a combinatorial auction. The geometric structure of each of the levels of the building features the properties described here. Quan (1994) reports on empirical studies in real estate auctions. Several of these studies have focused on verifying and quantifying the *afternoon effect*. This afternoon effect describes similar items consistently selling for significantly less in later rounds in multi-object sequential auctions. Quan (1994) even reports on finding this effect in a large real estate auction (122 lots) of vacant lots that are geographically similar. The lots were formed in 23 groups based on their geographical proximity. In 20 out of the 23 groups of properties, the afternoon effect was present with the last bidder paying on average one-third less than the first bidder for geographically similar lots. A combinatorial auction, by selling all items simultaneously, can mitigate this effect.
- Mineral rights. Imagine a region that is partitioned into lots, with the lots organized in rows. For sale is the right to extract minerals, oil or gas found on or below the surface of the lot. Clearly, having adjacent lots allows for exploration and production efficiencies, a complementarity. For

more about this particular setting, we refer to Cramton (2007). Figure 2 shows an example of oil and gas leases neatly arranged in rows.

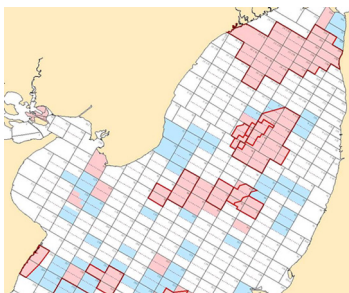


Figure 2: Oil and Gas Leases managed by the Texas General Land Office. Taken from: <http://www.glo.texas.gov/GLO/agency-administration/gis/gis-data.html>.

- Seats in a grandstand, theater or stadium. In some of these cases, one can even assume that a grid, consisting of rows and columns, is given where each cell represents a seat. Typically, demand exists for sets of adjacent seats - think of a family of four going to a ball game, or a group of friends visiting a concert. The complementarities that people perceive from adjacent seats offer possibilities for combinatorial auctions. Although tickets are usually sold at a fixed price, there are occasions where sports teams have auctioned off (part of) their seat licenses¹. Another, not unrealistic, example is the selling of airline tickets².
- Laboratory experiments. Scheffel et al. (2011) provide results of laboratory experiments testing different auction formats in five different value models. Their third value model has six pieces of land arranged in two rows on a shoreline. Bidders are interested in bundles that contain at least one lot at the shore. Their fourth value model has nine pieces of land arranged in three rows. In Scheffel et al. (2012) a local synergy value model is used in which 18 items are arranged rectangularly in three rows with bidders interested in adjacent items. Kazumori (2010) ran experiments using 16 items arranged rectangularly in four rows. Each agent has a base value for each item and a varying level of additional interest for adjacent items. These laboratory experiments required solving very small instances of the winner determination problem. In case one were to increase the number of pieces of land, or one wants to run a continuous

¹For instance, the New York Jets (NFL) have earned over \$16 million in an online auction for seat licenses. See <http://www.nfl.com/news/story/09000d5d80c071a4/article/jets-earn-more-than-16-million-in-online-psl-auction>.

²For instance, the article found at the following URL describes how some carriers require persons whose weight exceeds a given number to buy two (adjacent) tickets: <http://www.cheapair.com/blog/travel-tips/airline-policies-for-overweight-passengers-traveling-this-summer/>.

auction, or one wants to give bidders all sorts of feedback, an efficient algorithm for the winner determination becomes a necessity.

In all these cases, it is clear that complementarities between adjacent items exist; a combinatorial auction is best-placed to take these effects into account.

The main goal of this paper is to show how the specific geometric setting described above can be used to efficiently solve the winner determination problem (which is hard in general), using dynamic programming procedures. Additionally, we settle the complexity of the winner determination problem for bidding in a grid. This paper does not address mechanism design or bidding strategy issues.

Goossens et al. (2014) show that when a constraint is imposed stating that a bidder can have at most one winning bid, the winner determination problem is NP-hard even if all items are arranged on a single row. Hence, to have any prospect of coming up with a positive result, we allow bidders to win multiple bids. Notice however that, under some conditions on the bids, an optimal solution where each bidder has at most one winning bid is guaranteed to exist. This is the case, for instance, if the bids placed by each bidder satisfy at least one of the following conditions:

- every pair of bids of a bidder has a non-empty intersection
- all bids from the same bidder are *super-additive*, i.e. for any two disjoint sets S and T it should hold that the bid expressed on $S \cup T$ is at least as large as the sum of the expressed bids on S and T .

The first condition is satisfied if bidders place only one bid. Bids coming from (truthful) *single-minded* bidders, who are only interested in a specific set of items or a superset of these items, also satisfy the first condition. Indeed, more formally, single-minded bidders have a set of items S^* and a value v^* such that their valuation $v(S) = v^*$ for all $S \supseteq S^*$, and $v(S) = 0$ for all other S (see Nisan et al. (2007)). The second condition corresponds to the bids that can be expressed using a bidding language consisting of OR-bids (see Nisan (2000)). Summarizing, in these cases, our dynamic program will result in an optimal solution where each bidder has at most one winning bid.

1.1. Literature

Our problem is a special case of finding a maximum-weight independent set in a geometric intersection graph. In such a graph, there is a node for each bid (in our case: a (connected) set of rectangles), and two nodes are connected if and only if the corresponding bids overlap. Finding a maximum-weight independent set in a geometric intersection graph is a well-studied problem for several types of intersection graphs. For instance, in the work of Rothkopf et al. (1998), it is shown that if all items are arranged in a single row, and bids are only allowed for subsets of consecutive items, the resulting winner determination problem is polynomially solvable. These results follow from the equivalence of this problem to finding a maximum-weight independent set in an interval graph. For

an overview on results for more general intersection graphs we refer to Chan and Har-Peled (2012). Depending upon particular properties of the geometric figures, different complexity results are known. We restrict ourselves here to mentioning that for fat objects (like squares and disks) polynomial time approximation schemes are known (see Erlebach et al. (2001), Hochbaum and Maass (1985)). The important special case of finding a maximum-weight independent set in a rectangle intersection graph is considered in Chalermsook and Chuzhoy (2009).

In the context of auctions, Babaioff and Blumrosen (2008) and Christodoulou et al. (2010) study mechanism designs for the setting where geometric figures in the plane are the objects for sale. They sketch applications in advertising, renting land for exhibitions and licenses for location-based services. They show how to guarantee a certain fraction of the optimal welfare for certain shapes of geometric objects (e.g. convex figures). The geometric setting considered here is different; also we do not devise payment schemes for the bidders. This paper addresses the question of how to solve the winner determination problem, assuming bidders have placed bids for subsets of items.

Our problem is also somewhat related to rectangle packing. Given a set of rectangles, the rectangle packing problem is to find a bounding box (i.e. an enclosing rectangle) of minimum area that will contain the given rectangles without overlap. The optimization problem is NP-hard, while the problem of deciding whether a set of rectangles can be packed in a given bounding box is NP-complete (Leung et al., 1990). This resembles a setting where bidders want to acquire a set of seats in a theater, of given number and shape (e.g. four seats next to each other, a 3×2 block of seats, etc.), anywhere in the theater. This can be casted in our framework by having a bid for each possible set of seats. In general however, this problem is fundamentally different from our problem: in rectangle packing, given rectangles can be placed anywhere in the bounding box, whereas in our problem the position of the items are fixed and the decision to be made is whether or not to select a particular bid.

If we allow bidders to express multiple bids, the problem is NP-hard even in a setting where all items are arranged in a single row. Indeed, this follows immediately from the fact that the Job Interval Selection Problem (JISP) is MAX SNP-hard (Spieksma, 1999). In the JISP n pairs of intervals on the real line are given, and the objective is to select as many intervals as possible such that no two selected intervals intersect and at most one interval is selected from each pair.

1.2. Our Results

For the setting where items are arranged in rows, we show the following:

- For connected and gap-free bids (see Section 2 for precise definitions), the winner determination problem is easy when the number of rows is fixed (see Section 3.1). We provide a general polynomial time dynamic programming algorithm.

- For the setting where the bid space is a grid and both the number of rows and columns are a part of the input, we show that even when bids are constrained to be row bids or column bids, the resulting winner determination problem is NP-hard (see Section 3.2),
- For gap-free bids, the winner determination problem is NP-hard, even on two rows (see Section 4.1).
- For connected bids, the winner determination problem is easy on three rows or fewer (see Sections 4.2 and 4.3). We show this by adapting and expanding upon the general dynamic programming algorithm discussed in Section 3.1.

We point out that the complexity of the winner determination problem with connected bids on a fixed number of rows k , with $k \geq 4$, is still an open problem. If the number of rows is part of the input, a result in Rothkopf et al. (1998) implies the problem is NP-hard. An overview can be found in Table 1.

Rows	Connected and gap-free bids	Connected bids	Gap-free bids
1	$O(m + n)$		
2	$O(m^2 + nm)$		NP-hard (Viallette, 2004)
3	$O(m^3 + nm^2)$	$O(n^2 m^3)$	
$k : k \geq 3$	$O(m^k + nm^{k-1})$	Open problem	

Table 1: Overview of results if the number of rows k is not part of the input, m is the number of items and n is the number of bids.

2. Preliminaries

The geometric setting that we consider can be described as follows. Given are k rows. Each row contains an (ordered) set of items (or rectangles). If, on some row, an item a lies to the left of item b , then we write $a \prec b$. We use $X_j = \{0, 1, \dots, m_j\}$ to denote the set of items in row j , $j = 1, \dots, k$. The set of items that can be bid on is $\bigcup_{j=1}^k X_j \setminus \{0\}$; item 0 cannot be part of any bid, and is only present for notational convenience. We assume that item ℓ lies directly to the left of item $\ell + 1$, for each $\ell \in X_j \setminus \{m_j\}$, $j = 1, \dots, k$.

Definition 1. We say that a pair of items are adjacent if and only if they share a border with non-zero length.

Clearly, items ℓ and $\ell + 1$ are adjacent. However, items on different (but consecutive) rows can be adjacent as well. We use m to denote the number of items in the instance, i.e., $m = \sum_{j=1}^k m_j$. Figure 3 visualizes this.

We investigate the following problem, called the winner determination problem (WDP). Given is a set of bids \mathcal{B} on subsets of items, with $v(b)$ denoting the value of bid b , for each $b \in \mathcal{B}$. We set $n = |\mathcal{B}|$, i.e. there are n bids; specifying a

Row 1	1	2				m_1
Row 2	1	2				m_2
Row 3	1	2				m_3

Figure 3: An example of an instance with $k = 3$ (i.e. 3 rows) and $m_1 = 6$, $m_2 = 8$, $m_3 = 7$.

bid implies specifying a set of items, as well as a value $v(b) > 0$. The problem is to find an allocation that maximizes the sum of the values of the accepted bids, ensuring that each item is allocated at most once.

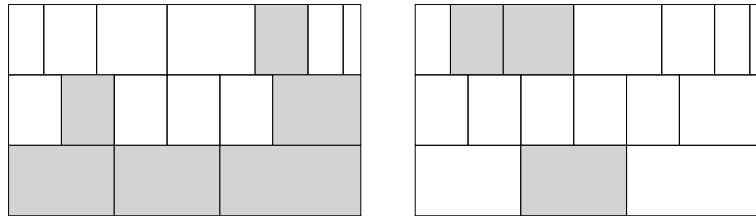
Given a bid b , consider the *item graph*, $H(b)$, which has a node for each item in bid b , and there is an edge between a pair of nodes in $H(b)$ if and only if the corresponding items are adjacent. There are two main restrictions on the bids that we consider. We define the concept of a *connected* bid.

Definition 2. We say that bid b is *connected* if the subgraph $H(b)$ induced by the items of bid b is connected. If bid b is not connected, we say that it is *disconnected*.

Further, let us define the concept of a bid that is *gap-free*. A formal definition of a bid having no gaps (i.e. being gap-free) is formulated as follows.

Definition 3. We say that bid b is *gap-free* if no three items $u \prec v \prec w$ on a single row exist for which $u \in b, v \notin b, w \in b$.

A bid that is not gap-free has at least one gap. Notice that it is easy to exhibit examples of connected bids that are not gap-free (see Figure 4a), and gap-free bids that are not connected (see Figure 4b). It is also easy to see that in the case of a single row, i.e. $k = 1$, connectedness of a bid is equivalent to a bid being gap-free.



(a) A bid that is connected and not gap-free. (b) A bid that is disconnected and gap-free.

Figure 4: Examples illustrating the concepts of a connectedness and gap-freeness.

Finally, it is important to see that bids on identical sets of items but with different values need not all be considered. Indeed, one need only consider the bid with the highest value. If more than one bid has the highest value, one

could use the bid entry time as a tie-breaker. Thus, all but the highest value bid on a specific set of items can be eliminated and bids will be unique in the sense that they are all for different sets of items.

3. Connected and gap-free bids

In this section we assume that bids are connected and gap-free. In Section 3.1 we describe a dynamic programming algorithm for the winner determination problem, tailored to our geometric setting described in Section 2, for the general case of k rows. This algorithm has a polynomial running time if we assume that k is not part of the input, or in other words, if we focus on problem instances with a specific number of rows. Notice that this assumption is reasonable for practical applications, as the auctioneer will typically be interested in a setting with one particular number of rows, namely that number resulting from the specific geometric structure underlying the items for sale.

In Section 3.2, we abandon this assumption, and study a setting where problem instances can have any number of rows (i.e. the number of rows is part of the input). We discuss a setting where items are arranged in a grid, and show that this problem is difficult, even when bids can cover only items in one row or one column.

3.1. A dynamic program for winner determination for case of k rows

This section is divided as follows: first we describe the dynamic programming algorithm for the case of k rows, then we proceed to a numerical example, after which we will discuss the proof of correctness.

The dynamic program for k rows.

Here we describe a dynamic programming approach for the case of k rows and bids that are both connected and gap-free. We show how the winner determination problem for this setting can be solved as a shortest path problem on a graph $G = (V, A)$, which is constructed as follows. There is a node in V for each element in the Cartesian product of the sets X_1, X_2, \dots, X_k . We write $V = \prod_{i=1}^k X_i$. Nodes in V are k -tuples. We consider the k -tuple $\mathbf{x} = \langle x_1, x_2, \dots, x_k \rangle$, where $x_1 \in X_1, x_2 \in X_2, \dots$ and $x_k \in X_k$. This k -tuple represents a state, i.e. a collection of assigned items. More specifically, the k -tuple \mathbf{x} represents a state where irrevocable decisions concerning the items $\{0, \dots, x_1\} \cup \{0, \dots, x_2\} \cup \dots \cup \{0, \dots, x_k\}$ have been made, i.e. for each row i all items from left to right up to and including x_i . As there is a node in V for every k -tuple, this leads to $O(m^k)$ nodes.

The arc set A includes two types of arcs: the zero arcs and bid arcs. The zero arcs have a weight of 0, and are used to handle items not included in the set of winning bids. Consider some node $\mathbf{x} = \langle x_1, x_2, \dots, x_i, \dots, x_k \rangle \in V$, with $1 \leq i \leq k$ and $x_i \neq m_i$. A zero arc goes from node \mathbf{x} to node $\langle x_1, \dots, x_i + 1, \dots, x_k \rangle \in V$, for each $1 \leq i \leq k$. Thus, up to k zero arcs emanate node $\mathbf{x} \in V$, giving rise to $O(m^k)$ zero arcs in the graph G .

The bid arcs correspond to actual bids and have a weight equal to the value of the bid $v(b)$. We represent a bid by listing k pairs of elements; each pair represents the first element, and the last element present in a bid on a particular row. For a bid b that contains elements from each of the k rows, we write: $b = \{(x_1^b, y_1^b), (x_2^b, y_2^b), \dots, (x_k^b, y_k^b)\}$, where the element $x_j^b \in X_j$ ($1 \leq j \leq k$) refers to the leftmost element of X_j present in bid b , and the element $y_j^b \in X_j$ ($1 \leq j \leq k$) refers to the rightmost element of X_j present in bid b . We use the symbol (\emptyset, \emptyset) to denote that a bid does not include items from that row. Thus, as an example, when we write $b = \{(\emptyset, \emptyset), (x_2^b, y_2^b), (x_3^b, y_3^b), (\emptyset, \emptyset)\}$ this means that the bid b does not include any items on the first row, it includes items x_2 up to and including y_2 on the second row, it includes items x_3 up to and including y_3 on the third row, and it does not include any items on the fourth row.

The bid arcs can be described as follows. Let us, for convenience, first assume that bid b contains elements from each of the k rows. To represent bid b in the graph G , we draw an arc from node $\langle x_1^b - 1, x_2^b - 1, \dots, x_k^b - 1 \rangle$ to node $\langle y_1^b, y_2^b, \dots, y_k^b \rangle$ with weight $v(b)$. Consider now a bid b such that there are rows with no elements in b . Observe that, due to connectedness of b , these rows can only have indices $1, 2, \dots, s(b)$ and $f(b), f(b) + 1, \dots, k$ with $0 \leq s(b) < f(b) \leq k + 1$. Note that if a bid b is present on the row 1 then $s(b) = 0$. Similarly, if a bid b is present on row k then $f(b) = k + 1$. Now, to represent bid b , for each $x_1 \in X_1, x_2 \in X_2, \dots, x_{s(b)} \in X_{s(b)}, x_{f(b)} \in X_{f(b)}, x_{f(b)+1} \in X_{f(b)+1}, \dots, x_k \in X_k$ there is an arc from node $\langle x_1, x_2, \dots, x_{s(b)}, x_{s(b)+1}^b - 1, \dots, x_{f(b)-1}^b - 1, x_{f(b)}, \dots, x_k \rangle$ to node $\langle x_1, x_2, \dots, x_{s(b)}, y_{s(b)+1}^b, \dots, y_{f(b)-1}^b, x_{f(b)}, \dots, x_k \rangle$ with weight $v(b)$. Notice that there are $O(nm^{k-1})$ bid arcs (of course it is conceivable that the number of bid arcs will be far less).

We now compute a longest path from node $\mathbf{0} = \langle 0, \dots, 0 \rangle$ to node $\mathbf{m} = \langle m_1, \dots, m_k \rangle$. The length of this path corresponds to the optimal revenue of the auction, and the winning bids can be derived from the arcs in the path. Notice that $G = (V, A)$ is acyclic by construction and consists of $O(m^k)$ nodes and $O(m^{k-1}(n + m))$ arcs. Hence, a longest path can be found efficiently by solving a shortest path problem in $G = (V, A)$ with edge weights multiplied by -1. Since Ahuja et al. (1993) show that shortest path problems in directed acyclic graphs with p nodes and q arcs can be solved in $O(p + q)$ time, our dynamic program requires $O(m^k + nm^{k-1})$ time.

Once a longest path is found, it is easy to see which bids are accepted. Every arc that is not a zero arc in $G = (V, A)$ corresponds to exactly one bid. To find the set of winning bids, for every non-zero arc in the longest path simply accept the bid corresponding to that arc.

Note that when the shortest path problem in $G = (V, A)$ is solved, it is easy to get *reduced costs* (i.e. *shadow prices*) for all arcs. The minimum of the reduced costs of all arcs corresponding to a single bid, is the amount by which the bid needs to be improved (i.e. increased) to be winning, if all other bids remain the same. This amount by which a bid needs to be increased, ceteris paribus, to become a winning bid has been termed the *winning level* of a bid

(Adomavicius and Gupta, 2005). Thus, in other words, the winning levels of currently non-winning bids are easy to compute by using our approach. This fact can be useful for providing feedback to bidders, helping bidders to better evaluate whether they should revise previous bids (Adomavicius et al., 2012).

A numerical example.

In this section we will use the layout found in Figure 5a to illustrate how the graph $G = (V, A)$ is created. As can be seen in Figure 5a, the example has two rows. In the first row there are three items and in the second row there are four items. Five bids are submitted, see Figures 5b-5f. Bid 1, b_1 with $v(b_1) = 12$, is on the first item in row 1 and on the first item in row 2. Bid 2, b_2 with $v(b_2) = 14$, is on item 2 in row 1 and items 2 and 3 in row 2. Bid 3, b_3 with $v(b_3) = 13$, is on the first item in row 1 and the first two items in row 2. Bid 4, b_4 with $v(b_4) = 9$, is on the last two items in row 1. Bid 5, b_5 with $v(b_5) = 4$, is on the last item in row 2. Given these bids, now the winner determination problem needs to be solved.

Now let us construct the graph $G = (V, A)$. First we construct the nodes. There is a node in V for each element in $X_1 \times X_2$. In our example: $X_1 = \{0, 1, 2, 3\}$ and $X_2 = \{0, 1, 2, 3, 4\}$. The resulting nodes are 2-tuples or pairs, each of which represent a state: the first component denotes which items have already been handled on the first row, the second component denotes which items have already been assigned on the second row. For example, in the pair $\langle 1, 3 \rangle$ we have already made decision regarding item 1 in row 1 and items 1, 2 and 3 in row 2. The appropriate nodes for the numerical example can be found in Figure 5g.

Bid 1 can be represented as $b_1 = \{(1, 1), (1, 1)\}$. That means that there is a single arc corresponding to bid 1 from $\langle 0, 0 \rangle$ to $\langle 1, 1 \rangle$. Bid 2 can be represented as $b_2 = \{(2, 2), (2, 3)\}$. That means that there is an arc corresponding to bid 2 from $\langle 1, 1 \rangle$ to $\langle 2, 3 \rangle$. Bid 3 can be represented as $b_3 = \{(1, 1), (1, 2)\}$. That means that there is an arc corresponding to bid 3 from $\langle 0, 0 \rangle$ to $\langle 1, 2 \rangle$. Bid 4 can be represented as $b_4 = \{(2, 3), (\emptyset, \emptyset)\}$. This leads to multiple arcs: from $\langle 1, 0 \rangle$ to $\langle 3, 0 \rangle$, from $\langle 1, 1 \rangle$ to $\langle 3, 1 \rangle$, from $\langle 1, 2 \rangle$ to $\langle 3, 2 \rangle$, from $\langle 1, 3 \rangle$ to $\langle 3, 3 \rangle$, and from $\langle 1, 4 \rangle$ to $\langle 3, 4 \rangle$. Bid 5 can be represented as $b_5 = \{(\emptyset, \emptyset), (4, 4)\}$. This also leads to multiple arcs: from $\langle 0, 3 \rangle$ to $\langle 0, 4 \rangle$, from $\langle 1, 3 \rangle$ to $\langle 1, 4 \rangle$, from $\langle 2, 3 \rangle$ to $\langle 2, 4 \rangle$ and from $\langle 3, 3 \rangle$ to $\langle 3, 4 \rangle$.

The longest path in the resulting graph from node $\langle 0, 0 \rangle$ to $\langle 3, 4 \rangle$ has a value of 30 and goes along the following arcs. First it goes from $\langle 0, 0 \rangle$ to $\langle 1, 1 \rangle$, which means bid 1 is selected. Next it goes from $\langle 1, 1 \rangle$ to $\langle 2, 3 \rangle$, meaning bid 2 is selected. Then there are two alternative but equally good paths. The first alternative is to go from $\langle 2, 3 \rangle$ to $\langle 2, 4 \rangle$, thus accepting bid 5 and then go from $\langle 2, 4 \rangle$ to $\langle 3, 4 \rangle$ by using the zero arc. The other alternative is to go from $\langle 2, 3 \rangle$ to $\langle 3, 3 \rangle$ by using the zero arc and then going from $\langle 3, 3 \rangle$ to $\langle 3, 4 \rangle$, thus accepting bid 5. The optimal solution of the winner determination problem in this instance is thus to accept bids 1, 2 and 5. The corresponding value is 30.

Proof of correctness.

1	2	3	
1	2	3	4

(a) Numerical example layout.

1	2		3
1	2	3	4

(c) Bid 2, b_2 , with $v(b_2) = 14$.

1	2	3	
1	2	3	4

(e) Bid 4, b_4 , with $v(b_4) = 9$.

1	2	3	
1	2	3	4

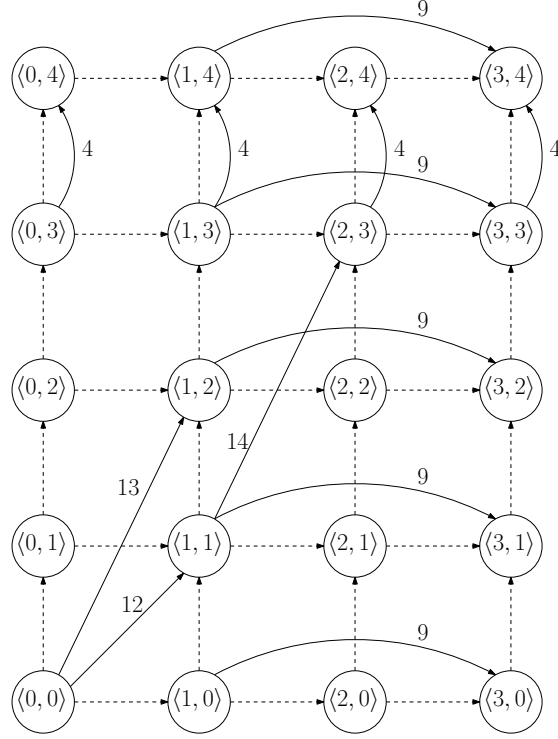
(b) Bid 1, b_1 , with $v(b_1) = 12$.

1	2	3	
1	2	3	4

(d) Bid 3, b_3 , with $v(b_3) = 13$.

1		2		3	
1	2	3	4		

(f) Bid 5, b_5 , with $v(b_5) = 4$.



(g) The resulting graph. The dashed arcs represent the zero-bids and as such have a value of zero. The other arcs are bid arcs and have a weight corresponding to the value of the bid.

Figure 5: The layout of the items, five bids, and the resulting graph.

Theorem 1. *The WDP with n connected and gap-free bids on m items that are arranged in k rows, can be solved by solving a shortest path problem in an acyclic graph $G = (V, A)$ with $O(m^k)$ nodes and $O(m^{k-1}(n + m))$ arcs for each $k \geq 1$.*

Proof. We show that the dynamic program for k rows, as described in 3.1, solves the winner determination problem to optimality, i.e. a longest path in G corresponds to a solution to the winner determination problem with the same value, and vice versa. We prove this by establishing a one-to-one correspondence between a feasible set of bids and a path in the graph G from $\mathbf{0} = \langle 0, \dots, 0 \rangle$ to $\mathbf{m} = \langle m_1, \dots, m_k \rangle$.

First, we show how a given set of non-overlapping bids corresponds to a path in G . We order the given bids in a sequence such that a bid containing item $x \in X_j$ comes before a bid containing item $y \in X_j$ when $x \prec y$ (for each $j \in \{1, \dots, k\}$). Notice that the bids being gap-free and connected implies that at least one such a sequence exists. The path in G corresponding to this sequence of bids consists of a single bid arc for each bid in the sequence, and zero arcs in between the bid arcs. Let us assume a partial path in G starting at $\mathbf{0}$ going to node $\mathbf{y} = \langle y_1, \dots, y_k \rangle$ has been found that corresponds to the first u bids in the sequence. Thus, the first u bids have allocated items up to y_1 in row 1, \dots , and up to y_k in row k . We show how to extend this partial path to incorporate the $(u + 1)$ -th bid.

Let b be the $(u + 1)$ -th bid. By definition $x_{s(b)+1}^b, \dots, x_{f(b)-1}^b$ are the leftmost items in the rows $s(b) + 1, \dots, f(b) - 1$ where bid b is present. By construction of the sequence we have that $y_i \prec x_i^b$ for $i = s(b) + 1, \dots, f(b) - 1$. Thus, we can use zero arcs starting in \mathbf{y} to bring us to node $\mathbf{y}' = \langle y_1, y_2, \dots, y_{s(b)}, x_{s(b)+1}^b - 1, \dots, x_{f(b)-1}^b - 1, y_{f(b)}, \dots, y_k \rangle$. Now we select the bid arc corresponding to bid b that leaves node \mathbf{y}' .

Second, given a path in G from $\mathbf{0}$ to \mathbf{m} , it is obvious how a feasible set of bids is chosen: simply take the bids corresponding to the bid arcs in the path. There can be no overlap between any pair of these bids, since there are no arcs in G from nodes \mathbf{x} to \mathbf{y} for which any of the components of \mathbf{x} succeeds a component of \mathbf{y} . The value of the set of bids coincides with the length of the path.

Finally, it is not difficult to verify that the graph G is acyclic, and hence a longest path can be found efficiently by solving a shortest path problem in G with edge weights multiplied by -1 . \square

3.2. The complexity of winner determination for bids in a grid

In this section we assume that a $k \times q$ grid is given, with $m = k \times q$ items (for instance representing seats in a grandstand, or a theater), and that connected bids are given. Naturally, the dynamic program for k rows, as presented in Section 3.1, can also be used to solve instances of a grid setting, as this is a special case of our geometric setting described in Section 2. However, if we consider instances for the grid setting with any number of rows (i.e. we consider k as part of the input), the dynamic program of Section 3.1 can no longer

guarantee a polynomial running time as the graph consists of $O(m^k)$ nodes and $O(m^{k-1}(n+m))$ arcs. As mentioned in Section 1, Rothkopf et al. (1998) find that if bids are allowed only on singletons, full rows, and full columns, the problem is easy to solve.

Probably the simplest bids that use multiple rows and columns are 2×2 bids (a 2×2 bid is a bid on cells $(i, j), (i, j+1), (i+1, j), (i+1, j+1)$) for some row i and column j . However, when each bid is a 2×2 bid, the complexity of the winner determination problem follows directly from the *tile salvage problem*. In this problem, an $k \times k$ grid is given, together with a set of unit squares that have been removed from this grid. The tile salvage problem is to find the maximum number of non-overlapping $x \times y$ tiled rectangles. Berman et al. (1990) show that the tile salvage problem is NP-complete, even for 2×2 tiles. Hence, our problem is hard even if only bids on 2×2 rectangles are allowed.

For connected bids in a $k \times q$ grid, the only setting whose complexity is open is a setting where each bid is either a *row* bid or a *column* bid. We say that a bid is a row bid (column bid) when it consists of consecutive items on some single row (column). Note that in this setting bids need not be on an entire row/column, but can be on a part of a row or a column as well. Obviously, if, in a grid of size $k \times q$, all bids are row bids (or all bids are column bids), the problem decomposes into k (q) independent single row (column) problems; however, if the instance contains both row bids and column bids, the complexity follows from the following observation.

Theorem 2. *The winner determination problem in a grid where each bid is a row bid or a column bid, is NP-hard.*

Proof. The following question is known to be NP-complete (Rendl and Woeginger, 1993). Given $2n$ distinct points in the plane, do there exist n axis-aligned, non-overlapping line segments each connecting a pair of points such that each point is connected to exactly one other point? A segment is called *axis-aligned* when the two points it connects either share an x -coordinate, or share a y -coordinate. Like Rendl and Woeginger (1993), we will call this problem RDOS (reconstruction of sets of disjoint orthogonal segments).

Let us now build an instance of the winner determination problem in a grid. For each distinct y -coordinate in the instance of RDOS there is a row in our problem, and for each distinct x -coordinate there is a column in our problem. This specifies the grid. Every cell of the grid corresponds to an item. An example can be seen in Figure 6a and 6b. For each pair of points sharing a y -coordinate (x -coordinate), there is a row (column) bid with value 1, containing all items in between the two points sharing the y -coordinate (x -coordinate). This completely specifies an instance of the winner determination problem in the grid. An example can be seen in Figure 6c and 6d. Now, if total revenue of the corresponding auction has a value of n , then apparently there are n row and column bids that do not overlap. These n bids correspond to n axis-aligned segments, and the answer to the question is yes. Finally, if the answer to the question is yes, there exist n non-overlapping row and column bids. \square

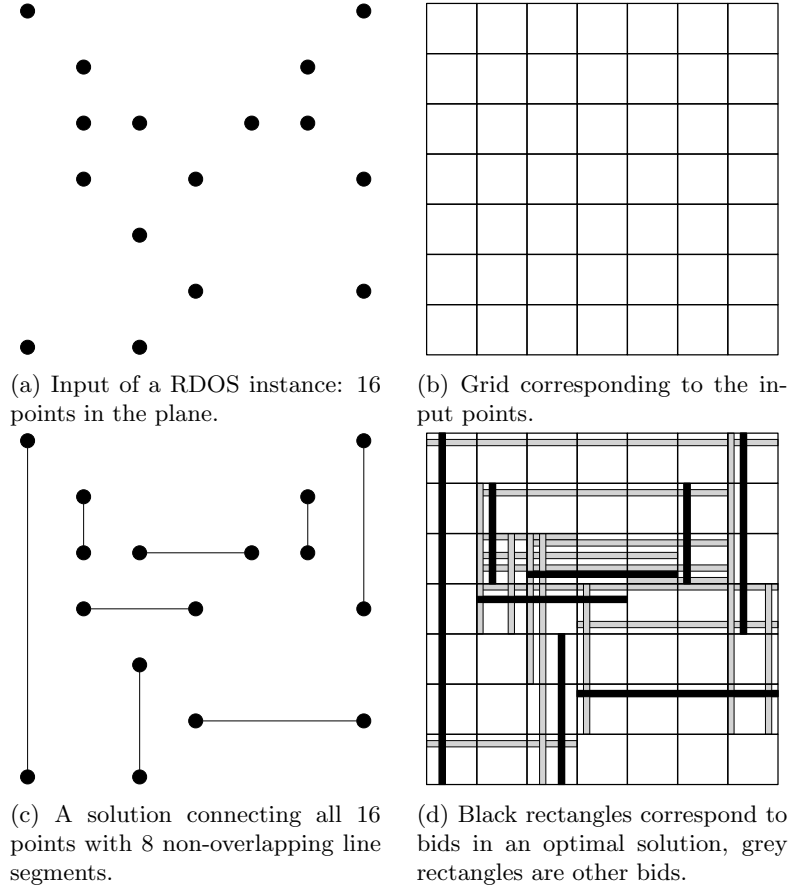


Figure 6: Transformation of a RDOS instance to an instance of the winner determination problem in a grid with row and column bids.

Notice that there is an easy 2-approximation algorithm for this setting. The approximation goes as follows. First, consider only row bids and solve the corresponding winner determination problem by solving the problem for each row. Next, perform a similar procedure for the sets of column bids. Finally, we take the best result of these two feasible solutions. It is easy to see that this is in fact a 2-approximation. Recall that solving the winner determination problem for connected bids on a single row is polynomially solvable.

4. Variants

In this section, we take a more detailed look at the case of two rows, showing the impact of each of the two assumptions (connected and gap-free) on the computational complexity of the WDP. Finally, we show how the dynamic program can be generalized to treat the case of three rows and connected bids.

4.1. The case of two rows and gap-free bids

We relax here the condition of connectedness; we only assume that bids are gap-free (we have two rows however). We claim that, in this case, the WDP becomes a special case of the problem of finding a maximum-weight independent set in a graph that is the edge-union of two interval graphs. Indeed, observe that since a bid is gap-free we can see each bid as the union of a set of consecutive items in row 1 and a set of consecutive items in row 2. By concatenating row 1 and row 2 into a single row, one can view each bid as consisting of two intervals, a left and a right interval. The resulting intersection graph has a node for each bid, and two nodes are connected if either their left intervals, or their right intervals (or both) overlap; in other words, the resulting intersection graph is a 2-union graph. It is shown in Bar-Yehuda et al. (2006), that the maximum-weight independent set problem is NP-hard on 2-union graphs, see also van Bevern et al. (2015). Note however that the intersection graph resulting from the 2-row problem we investigate is a special case of 2-union graphs. Indeed, in our special case all left intervals are to the left of all right intervals, which is not necessarily the case in a 2-union graph. However in the context of computational biology this precise special case has been studied by Vialette (2004).

Lemma 1. *The WDP with n gap-free bids on m items that are arranged in two rows, is NP-hard.*

Proof. See the proof of Proposition 7 in Vialette (2004). \square

4.2. The case of two rows and connected bids

Consider the case where bids are still connected, but not necessarily gap-free. Figure 7 shows an example of a bid that has 2 gaps. Given a bid b , the

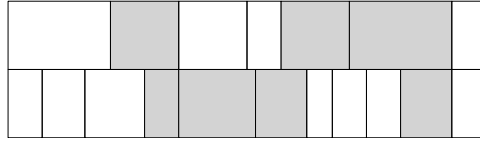


Figure 7: A bid with 2 gaps in a 2-row problem.

set of items that are in gap(s) of this bid b is given by $G(b) = \{x \notin b : \exists u, v \in b \text{ with } u \prec x \prec v\}$. In case $G(b)$ is empty, b is gap-free; otherwise $G(b)$ consists of, say $p(b)$ ($p(b) < m$), connected itemsets, each representing a single gap. More precisely, let $H(G(b))$ be the item graph corresponding to the items in $G(b)$; each of the $p(b)$ connected components of $H(G(b))$ corresponds to items making up a single gap. We write $G(b) = \bigcup_{\ell=1}^{p(b)} G^\ell(b)$, where $G^\ell(b)$ represents the items present in the ℓ -th gap of bid b where $1 \leq \ell \leq p(b)$.

Theorem 3. *The WDP with n connected bids on m items that are arranged in two rows, can be solved in polynomial time.*

Proof. Observe that, for each $\ell = 1, \dots, p(b)$ and $b \in B$, the itemset $G^\ell(b)$ consists of items on a single row (otherwise b would be disconnected). Let us now consider an instance defined by itemset $G^\ell(b)$, and by all bids $b' \in B$ that are contained in this itemset. Since each b' is connected (by assumption) and since $G^\ell(b)$ consists of items on a single row (see above), we can easily compute the value of this instance (denoted by $v(G^\ell(b))$) by using Theorem 1 with $k = 1$. Given a bid b , we do this for each $\ell = 1, \dots, p(b)$ finding the values $v(G^\ell(b))$ by applying Theorem 1 for $k = 1$ $O(m)$ times.

Finally, given an instance, we build a new instance where we replace each bid b that is not gap-free by a combined bid on the itemset $b \cup G(b)$, with a value $v(b) + \sum_{\ell=1}^p v(G^\ell(b))$. The resulting instance is created in polynomial time, is gap-free, and thus we can use Theorem 1 to solve it. \square

4.3. The case of three rows and connected bids

Here, we show how the winner determination problem for the setting with 3 rows and connected bids can be solved as a shortest path problem, using a generalization of the approach described in Section 3.1 that can handle bids with open gaps. The main challenge in this case is how to deal with gaps that may be present in a bid.

Let us first define the concept of an instance graph H . The instance graph H has a node for each item $x \in (X_1 \setminus \{0\}) \cup (X_2 \setminus \{0\}) \cup (X_3 \setminus \{0\})$. Two nodes corresponding to items that are adjacent are connected; moreover, there is a node s in the graph which is connected to the first item in each of the three rows, and there is a node t connected to the last item in each of the three rows (see Figure 8 for an example).

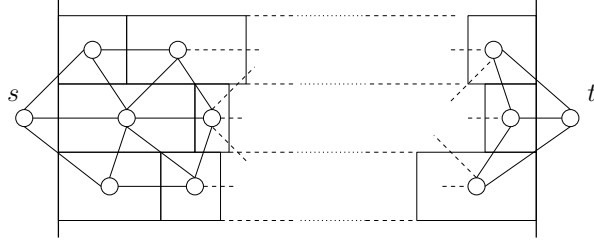


Figure 8: The instance graph H .

4.3.1. Types of gaps and bids

We distinguish two kind of gaps. To that end, consider the instance graph H , and a connected bid b , and suppose that bid b is not gap-free. Thus, each of the p gaps in bid b is represented by itemset $G^\ell(b)$, $\ell = 1, \dots, p$.

Definition 4. We call a gap $G^\ell(b)$ an *open gap* if, in the graph $H \setminus H(b)$, there is a path from each $x \in G^\ell(b)$ to either node s or node t . Each gap that is not an open gap is called a *closed gap*.

Because there are only 3 rows, a closed gap contains items in at most 2 rows (since a closed gap on 3 rows corresponds to a bid that is not connected). Also notice that an open gap only has items in the second row.

In Figure 9 there are 3 examples. In the top example, there is one gap with items on the first and second row. However, since there is no path in $H \setminus H(b)$ from any of the $x \in G^1(b)$ to either s or t , it is a closed gap. In the middle example, there is a gap with items on the second row. There exists a path in $H \setminus H(b)$ from each $x \in G^1(b)$ to s . Therefore, the gap in this bid is an open gap. In the bottom example of Figure 9 there are 4 gaps. The first gap, $G^1(b)$ on the left has one item on the first row; the second gap, $G^2(b)$, has one item on the third row; the third gap, $G^3(b)$, has one item on the first row and one item on the second row. There is no path in $H \setminus H(b)$ from any of the $x \in G^\ell(b)$ to either s or t for $\ell = 1, 2, 3$. This means that these three gaps are closed gaps. Finally, the fourth gap $G^4(b)$, is on the right and has one item on the second row. There is a path in $H \setminus H(b)$ from the item in $G^4(b)$ to t , making this gap an open gap.

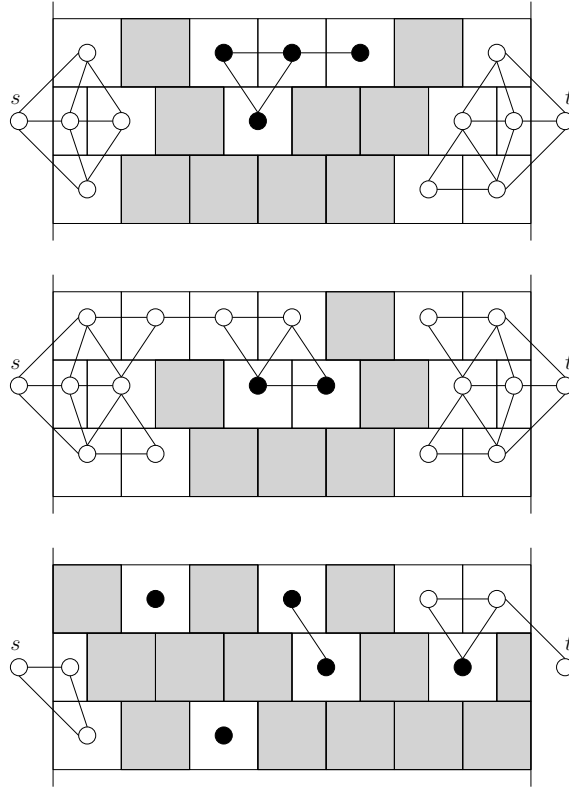


Figure 9: Examples of graphs $H \setminus H(b)$. Black nodes correspond to items in a gap.

We now partition the class of connected bids in two disjoint subclasses according to the following definition.

Definition 5. If a bid b has at least one open gap, it belongs to the subclass called *open bids*. The set of open bids is $\mathcal{B}_{\text{open}}$. If b has no open gaps, it belongs to the subclass of *closed bids*. The set of closed bids is $\mathcal{B}_{\text{closed}}$.

Consider a bid and its closed gaps. For each such closed gap, we solve the corresponding instance, yielding a value v . We then replace the bid with a combined bid that has the closed gaps filled and its original value increased by v . From Theorem 3 it follows that this operation can be done in polynomial time (recall that a closed gap contains items on at most two rows).

After this preprocessing of the bids, all closed gaps in all bids are ‘filled’ optimally. See for example Figures 10a, 10b and 10c. In Figure 10a there is a gap spanning the top and middle row, which is then filled optimally by solving a subproblem. In this case, the entire gap has been covered by other bids, but this is not necessarily always true. In Figure 10b, there is a gap only on the middle row. It is filled optimally by solving a subproblem which in this case only covers half the space in the gap. In Figure 10c, there is both a closed gap and an open gap. The closed gap has been filled optimally by solving a subproblem. After this preprocessing step we can ignore the gaps in closed bids, because

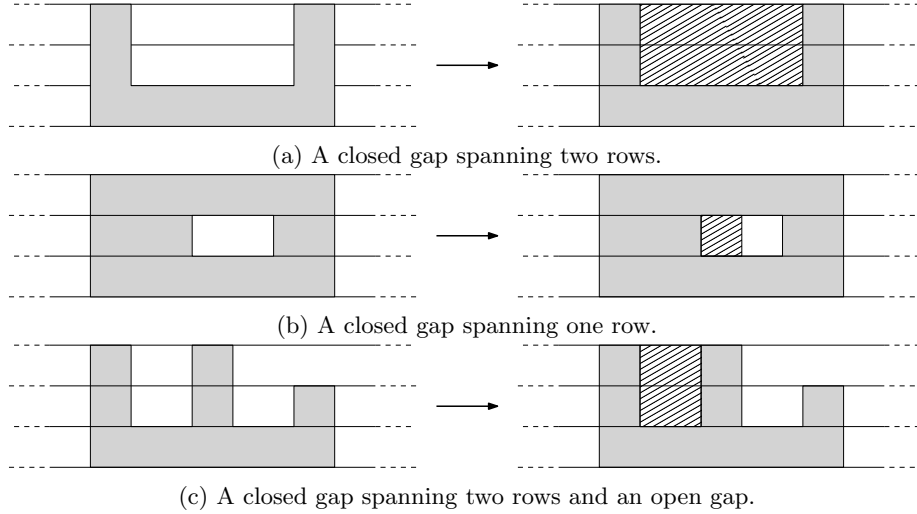


Figure 10: Filling closed gaps optimally.

they are all filled optimally (replacing the closed bid with a combined bid). In an open bid, all closed gaps have been filled as well, however, there is always at least one open gap remaining. Note that of course $\mathcal{B}_{\text{open}} \cup \mathcal{B}_{\text{closed}} = \mathcal{B}$ and $\mathcal{B}_{\text{open}} \cap \mathcal{B}_{\text{closed}} = \emptyset$ hold.

4.3.2. A polynomial-time algorithm

We show how the winner determination problem for the setting with 3 rows and connected bids can be solved as a shortest path problem. We construct the graph $G = (V, A)$ as follows.

The nodes. We define $\langle x_1, x_2, x_3, b \rangle$ (quadruples), where $x_1 \in X_1$, $x_2 \in X_2$, $x_3 \in X_3$, $b \in (\mathcal{B}_{\text{open}} \cup \emptyset)$. Every such quadruple corresponds to a node in V where items $\{0, \dots, x_1\} \cup \{0, \dots, x_2\} \cup \{0, \dots, x_3\}$ have been allocated. If $b = \emptyset$, we are in a *bid-independent state* (which corresponds to the states described in Section 3.1). If $b \neq \emptyset$, then we are in a *bid-dependent state* where we have to take into account one or more open gaps and have also assigned items $\{x'_b, \dots, x''_b\}$ in the middle row with $x_2 \prec x'_b$. Specifically, each open bid b is characterized by x'_b and x''_b , where x'_b is the leftmost item of the last contiguous set of items on the second row included in b and x''_b is the rightmost item of the last contiguous set of items on the second row included in b . Note that it is possible that $x'_b = x''_b$.

The arcs. There are 2 types of arcs. The first type of arcs are zero arcs, which are used to handle items not included in the set of winning bids. These arcs are not associated with any bid and thus have length 0. We distinguish 3 different types of zero arcs:

- Arcs between two bid-independent nodes:
 - from $\langle x_1, x, y, \emptyset \rangle$ to $\langle x_1 + 1, x, y, \emptyset \rangle$, $\forall x_1 \in X_1 \setminus \{m_1\}$, $x \in X_2$, $y \in X_3$,
 - from $\langle x, x_2, y, \emptyset \rangle$ to $\langle x, x_2 + 1, y, \emptyset \rangle$, $\forall x \in X_1$, $x_2 \in X_2 \setminus \{m_2\}$, $y \in X_3$, and
 - from $\langle x, y, x_3, \emptyset \rangle$ to $\langle x, y, x_3 + 1, \emptyset \rangle$, $\forall x \in X_1$, $y \in X_2$, $x_3 \in X_3 \setminus \{m_3\}$.
- Arcs between two bid-dependent nodes: from $\langle x, x_2 - 1, y, b \rangle$ to $\langle x, x_2, y, b \rangle$, $\forall x \in X_1$, $x_2 \in X_2 \setminus \{0\} : x_2 \prec x'_b$, $y \in X_3$, $b \in \mathcal{B}_{\text{open}}$.
- Arcs between bid-dependent and bid-independent nodes: from $\langle x, x'_b - 1, y, b \rangle$ to $\langle x, x''_b, y, \emptyset \rangle$, $\forall x \in X_1$, $y \in X_3$, $b \in \mathcal{B}_{\text{open}}$.

The second type of arcs are those which are associated with actual bids. The lengths of these are equal to the value of the corresponding (combined) bid. Note that there may be multiple arcs corresponding to the same bid. We now describe the 4 components of a node $\langle x_1, x_2, x_3, b' \rangle$ that make up a starting node for an arc that corresponds to a connected bid $b \neq b'$:

- First tuple component x_1 : if $b \cap X_1 \neq \emptyset$, then $x_1 = x - 1$, where x is the leftmost item in row 1 included in b . Otherwise all elements in X_1 are possible values for x_1 , i.e. multiple arcs will need to be constructed.
- Second tuple component x_2 : if $b \cap X_2 \neq \emptyset$, then $x_2 = x - 1$, where x is the leftmost item in row 2 included in b . Otherwise all elements in X_2 are possible values for x_2 , i.e. multiple arcs will need to be constructed.
- Third tuple component x_3 : if $b \cap X_3 \neq \emptyset$, then $x_3 = x - 1$, where x is the leftmost item in row 3 included in b . Otherwise all elements in X_3 are possible values for x_3 , i.e. multiple arcs will need to be constructed.

- Fourth tuple component b' : $b' = \emptyset$ or $b' \in \mathcal{B}_{\text{open}}$ for which the following holds:
 - $b \cap b' = \emptyset$ (no overlap) and
 - $\exists x \in b \cap X_2, \exists x' \text{ and } x'' \in b' \cap X_2 : x' \prec x \prec x''$ (b has an item in an open gap of b') and
 - $\nexists x \in b \cap X_2, x' \in b' \cap X_2 : x \prec x'$ (b does not have an item to the left of b' in the second row)

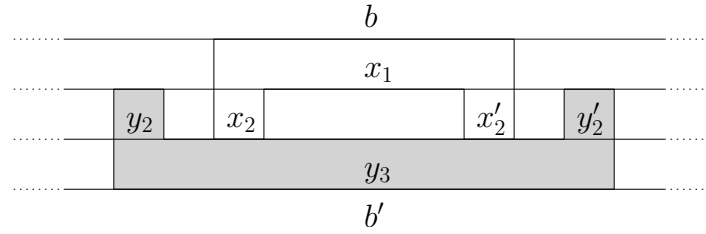
Now that we have determined all possible starting nodes for every connected bid, we have to determine the end nodes. End nodes for arcs depend on two things: the starting node and whether the bid b to which the arc corresponds is a closed or an open bid. We distinguish four cases:

- case 1: bid-independent starting node, closed bid
- case 2: bid-independent starting node, open bid
- case 3: bid-dependent starting node, closed bid
- case 4: bid-dependent starting node, open bid

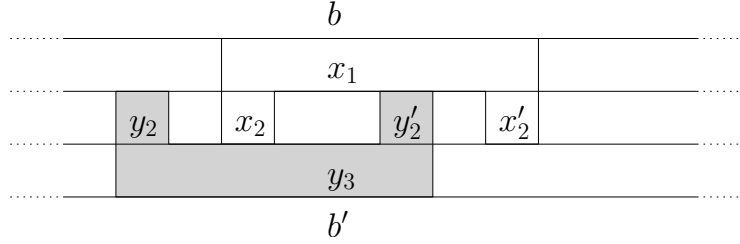
We will now discuss how the end node is constructed from the starting node in each case.

- Case 1: for every row for which there is an item $x \in b$, change the corresponding tuple component to the rightmost item included in b in that row, otherwise keep the value of the starting node. The fourth tuple component remains the same as the starting node, which is \emptyset .
- Case 2: for the first and third row for which there is an item $x \in b$, change the corresponding tuple component to the rightmost item of that row included in b , otherwise keep the value of the starting node. For the second row: find the leftmost contiguous interval included in b and change the second tuple component to the rightmost item in that interval. The fourth tuple component will change to b .
- Case 3: for every row for which there is an item $x \in b$, change the corresponding tuple component to the rightmost item included in b in that row, otherwise keep the value of the starting node. The fourth tuple component remains the same as the starting node.
- Case 4: let b' be an open bid with its leftmost item in row 2 to the left of leftmost item of open bid b in row 2. Observe that the itemset $b' \cup b$ may contain a closed gap: indeed there are two basic cases depending on whether the rightmost item in row 2 in b precedes (Figure 11a) or succeeds (Figure 11b) the rightmost item in row 2 in b' . In both cases, the value of the arc will be increased with the optimal value of a subproblem on the second row limited to the shaded area. In other words, we

construct a combined bid. In the case of Figure 11a, the first three tuple components are changed according to the rightmost item included in b in that row, otherwise keeping the value of the starting node. The fourth tuple component will remain the same. In the example, the arc would go from $\langle x_1 - 1, x_2 - 1, y_3, b' \rangle$ to $\langle x_1, x'_2, y_3, b' \rangle$. In the case of Figure 11b, the first three tuple components are changed according to the rightmost item included in b in that row, otherwise keep the value of the starting node. However, the fourth tuple component will be changed to b . In the example, the arc would go from $\langle x_1 - 1, x_2 - 1, y_3, b' \rangle$ to $\langle x_1, y'_2, y_3, b \rangle$.



(a) Two open bids creating an extra gap: option 1.



(b) Two open bids creating an extra gap: option 2.

Figure 11: Case 4 arcs.

Shortest path. We now compute a longest path from node $\langle 0, 0, 0, \emptyset \rangle$ to node $\langle m_1, m_2, m_3, \emptyset \rangle$. The length of this path corresponds to the optimal revenue of the auction, and the winning bids can be derived from the arcs in the path. Notice that $G = (V, A)$ is acyclic by construction and consists of $O(nm^3)$ nodes and $O(n^2m^3)$ arcs. Hence, a longest path can be found efficiently by solving a shortest path problem in $G = (V, A)$ with edge weights multiplied by -1. In the next section, we prove the correctness of this algorithm.

4.3.3. Proof of correctness

In order to prove the correctness of the algorithm described in Section 4.3.2, we show that (1) each path from node $\langle 0, 0, 0, \emptyset \rangle$ going to node $\langle m_1, m_2, m_3, \emptyset \rangle$ corresponds to a feasible allocation for the auction, and (2) vice-versa. Recall that after the preprocessing step described in Section 4.3.1, all closed gaps have been filled, resulting in combined bids and corresponding arcs in G . As the

individual bids corresponding to these arcs can easily be traced, we will ignore closed gaps in the remainder of this proof.

(1) Intuitively, consider a path starting from node $\langle 0, 0, 0, \emptyset \rangle$ and going to node $\langle m_1, m_2, m_3, \emptyset \rangle$. This path corresponds to an allocation for the auction by accepting the bids corresponding to the arcs associated with bids (the zero arcs can obviously be ignored). In order to show that this allocation is feasible, we need to argue that no pair of bids has overlap. Notice that for all arcs in the graph, when comparing the end node with the start node, none of the first 3 tuple components decreases, and at least one increases. This means that each arc represents moving to the right on at least one row, and that moving (back) to the left is not possible. For arcs with a bid-independent starting node, the start node corresponds to the leftmost items on each row included in the bid. Hence, overlap between the bid corresponding to this arc and any of the bids corresponding to previous arcs in the path is not possible. For arcs with bid-dependent starting nodes, more care is needed. Consider an arc whose starting node has b' as the fourth tuple component. By construction, for each such arc corresponding to a bid b , there is no overlap between b' and b . Furthermore, as the endpoint of this arc determines to what extent the open gap(s) of b' has been filled by b , we also avoid overlap between b and the corresponding bid of a possible next arc with value b' in the fourth tuple component of its starting node. The only way to move from a bid-dependent to a bid-independent node is through a zero arc, which ensures that the value for the second tuple corresponds with the rightmost item on the second row of the open bid b' , thereby excluding overlap between this bid and bids corresponding to subsequent arcs.

(2) Consider a feasible solution for the winner determination problem (i.e. no pair of bids in the allocation overlaps). By construction, for each (combined) bid in the allocation at least one arc in the graph exists. We show how to identify a path in G that corresponds with the bids in the allocation. We order the bids in a sequence such that bid p comes before bid q if p contains an item $x \in X_2$ for which $x \prec y$ for all $y \in X_2$ contained in bid q . In other words, we order the bids based on their leftmost item on the second row. Since we assume that bids are connected, bids that do not contain items on the second row have all items either on the first row or on the third. These bids should be inserted in the order such that a bid containing item $x \in X_j$ comes before a bid containing item $y \in X_j$ when $x \prec y$ (for each $j \in \{1, 3\}$). Recall that these bids are not used to fill closed gaps, as we handled this in the preprocessing step. Hence, at least one such sequence exists.

The path in G corresponding to this sequence of bids has a single bid arc for each (combined) bid in the sequence, and zero arcs in between the bid arcs. Let us assume a partial path in G starting at $\langle 0, 0, 0, \emptyset \rangle$ going to node $\langle u_1, v_1, w_1, b' \rangle$ has been found that corresponds to the first k bids in the sequence. We show how to extend this partial path to incorporate the $(k + 1)$ -th bid, say bid b . We discern 3 situations:

- $b' = \emptyset$: if b has items on each row, we select the corresponding arc starting at node $\langle u_2 - 1, v_2 - 1, w_2 - 1, \emptyset \rangle$, where u_2, v_2 , and w_2 are the leftmost items in b . If b has no items on one or more rows, we select the arc starting from the node with tuple value u_1, v_1 and/or w_1 for the respective row(s) on which b has no items.
- $b' \neq \emptyset$ and b is a closed bid: we select the corresponding arc starting at node $\langle x, y, z, b' \rangle$, where $x = u_1$ ($y = v_1, z = w_1$) if b does not include an item on the first (second, third) row, and $x = u_2 - 1$ ($y = v_2 - 1, z = w_2 - 1$) otherwise (where u_2, v_2 , and w_2 are the leftmost items in b).
- $b' \neq \emptyset$ and b is an open bid: we select the same arc as in the previous case, however, in this case one or more bids on items in row 2 may be enclosed between bids b and b' (see Figures 11a and 11b). As the value of these bids is included in the weight of the arc corresponding to b , we can remove these bids from the sequence (observe that these bids succeed b).

Notice that from all bid arcs that correspond to this bid b , we select one, and that we can always reach the selected arc from $\langle u_1, v_1, w_1, b' \rangle$ using zero arcs. Next, we iteratively select the next bid in the order, and proceed analogously. After having treated the last bid in the order, if the end node of the corresponding arc is not $\langle m_1, m_2, m_3, \emptyset \rangle$, we connect to this node using zero arcs. The following result is now apparent.

Theorem 4. *The WDP with n connected bids on m items that are arranged in three rows can be solved by solving a shortest path problem in a graph with $O(nm^3)$ nodes and $O(n^2m^3)$ arcs.*

5. Conclusion

We study the winner determination problem for a combinatorial auction with a specific geometric structure. We argue that this structure is relevant, as it occurs in real estate, plots of land, mineral rights, and theaters and stadium seats. The complementarities present in these situations offer great potential for combinatorial auctions. We point out that the items need not be rectangular but can be of any shape. In fact, if the itemset can be partitioned into k ordered subsets (rows), such that the adjacency relations between pairs of items on consecutive rows are consistent with the ordering of the items in each row, our framework applies.

With our dynamic programming algorithm, we present auctioneers a tool that enables them, under some reasonable assumptions on the bids and with a fixed number of rows, to efficiently compute the winning bids. Next, we complement existing results by showing that bidding in a grid is difficult, even when only row and column bids are allowed, if the number of rows is part of the input. We further investigate the precise impact of our assumptions.

Solving the winner determination problem efficiently is an essential component of mechanism design. As our paper assumes the bids are given, future

research that focusses on determining accompanying auction rules, and studies their impact on bidding strategy, efficiency and revenue would be valuable. Finally, our results may also prove useful for experimental research: our dynamic program will allow researchers to study bidder behavior in larger settings, involving more items and bidders than considered so far.

Acknowledgement

This research is supported by the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office. The authors wish to thank James B. Orlin for an interesting and stimulating conversation.

References

- Abrache, J., Crainic, T. G., Gendreau, M., Rekik, M., 2007. Combinatorial auctions. *Annals of Operations Research* 153 (1), 131–164.
- Adomavicius, G., Curley, S. P., Gupta, A., Sanyal, P., 2012. Effect of Information Feedback on Bidder Behavior in Continuous Combinatorial Auctions. *Management Science* 58 (4), 811–830.
- Adomavicius, G., Gupta, A., 2005. Toward Comprehensive Real-Time Bidder Support in Iterative Combinatorial Auctions. *Information Systems Research* 16 (2), 169–185.
- Ahuja, R. K., Magnanti, T. L., Orlin, J. B., 1993. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
- Babaioff, M., Blumrosen, L., 2008. Computationally-feasible truthful auctions for convex bundles. *Games and Economic Behavior* 63 (2), 588–620.
- Bar-Yehuda, R., Halldórsson, M. M., Naor, J., Shachnai, H., Shapira, I., 2006. Scheduling Split Intervals. *SIAM Journal on Computing* 36 (1), 1–15.
- Berman, F., Johnson, D., Leighton, T., Shor, P. W., Snyder, L., 1990. Generalized Planar Matching. *Journal of Algorithms* 11 (2), 153–184.
- Chalermsook, P., Chuzhoy, J., 2009. Maximum Independent Set of Rectangles. In: *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*. pp. 892–901.
- Chan, T. M., Har-Peled, S., 2012. Approximation Algorithms for Maximum Independent Set of Pseudo-Disks. *Discrete Computational Geometry* 48, 373–392.
- Christodoulou, G., Elbassioni, K., Fouz, M., 2010. Truthful Mechanisms for Exhibitions. In: *Internet and Network Economics*. Vol. 6484. Springer Berlin Heidelberg, pp. 170–181.

- Cramton, P., 2007. How Best to Auction Oil Rights. In: Humphreys, M., Sachs, J., Stiglitz, J. E. (Eds.), *Escaping the resource curse*. Cambridge Univ Press, Ch. 5, pp. 114–151.
- Cramton, P., Shoham, Y., Steinberg, R., 2006. *Combinatorial auctions*. MIT Press.
- de Vries, S., Vohra, R. V., 2003. Combinatorial Auctions: A Survey. *INFORMS Journal on Computing* 15 (3), 284–309.
- Erlebach, T., Jansen, K., Seidel, E., 2001. Polynomial-time Approximation Schemes for Geometric Graphs. In: *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms*. pp. 671–679.
- Goossens, D. R., Onderstal, S., Pijnacker, J., Spieksma, F. C. R., 2014. Solids: A Combinatorial Auction for Real Estate. *Interfaces* 44 (4), 351–363.
- Hochbaum, D. S., Maass, W., 1985. Approximation Schemes for Covering and Packing Problems in Image Processing and VLSI. *The Journal of the ACM* 32 (1), 130–136.
- Jackson, C., 1976. *Technology for Spectrum Markets*. Ph.D. thesis, Department of Electrical Engineering, School of Engineering, MIT.
- Kazumori, E., 2010. *Core-Selecting Auctions: An Experimental Study*. CARF F-Series CARF-F-226, Center for Advanced Research in Finance, Faculty of Economics, The University of Tokyo.
- Leung, J. Y. T., Tam, T. W., Wong, C., Young, G. H., Chin, F. Y., 1990. Packing squares into a square. *Journal of Parallel and Distributed Computing* 10 (3), 271–275.
- Nisan, N., 2000. Bidding and Allocation in Combinatorial Auctions. In: *Proceedings of the 2nd ACM Conference on Electronic Commerce*. EC '00. pp. 1–12.
- Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V. V., 2007. *Algorithmic game theory*. Vol. 1. Cambridge University Press Cambridge.
- Quan, D., 1994. Real Estate Auctions: A Survey of Theory and Practice. *Journal of Real Estate Finance and Economics* 9 (1), 23–49.
- 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.
- Rendl, F., Woeginger, G., 1993. Reconstructing sets of orthogonal line segments in the plane. *Discrete Mathematics* 119 (1-3), 167–174.
- Rothkopf, M. H., Pekeč, A., Harstad, R. M., 1998. Computationally Manageable Combinational 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.
- Scheffel, T., Pikovsky, A., Bichler, M., Guler, K., 2011. An Experimental Comparison of Linear and Nonlinear Price Combinatorial Auctions. *Information Systems Research* 22 (2), 346–368.
- Scheffel, T., Ziegler, G., Bichler, M., 2012. On the impact of package selection in combinatorial auctions: an experimental study in the context of spectrum auction design. *Experimental Economics* 15 (4), 667–692.
- Spieksma, F. C. R., 1999. On the approximability of an interval scheduling problem. *Journal of Scheduling* 2, 215–227.
- van Bevern, R., Mnich, M., Niedermeier, R., Weller, M., 2015. Interval scheduling and colorful independent sets. *Journal of Scheduling* 18, 449–469.
- Van Hoesel, S., Müller, R., 2001. Optimization in electronic markets: examples in combinatorial auctions. *Netnomics* 3 (1), 23–33.
- Viallette, S., 2004. On the computational complexity of 2-interval pattern matching problems. *Theoretical Computer Science* 312 (23), 223–249.