Testing BOI and BOB algorithms for solving the Winner Determination Problem in Radio Spectrum Auctions

Y. Saez [a]; A. Mochon [b]; J.L. Gomez-Barroso [b] and P. Isasi [a]

[a] *Artificial Intelligence Department of Carlos III University in Madrid, Spain.*
[b] *Department of Applied Economics, UNED, Madrid. Spain,*
yago.saez@uc3m.es;{amochon;jlgomez}@cee.uned.es;isasi@ia.uc3m.es

Abstract

Combinatorial auctions are a promising auction format for allocating radio spectrum, as well as other goods. An important handicap of combinatorial auctions is determining the winner bids among many options, that is, solving the winner determination problem (WDP). This paper tackles this computational problem using two approaches in a combinatorial first-price sealed bid auction. The first one, is an A* based on items (BOI). The second one, is an A* based on bids (BOB). These two techniques are tested in several scenarios for allocating radio spectrum licenses. The results obtained reveal that the search algorithm A* with the BOB formulation outperforms the other and always finds the optimal solution very quickly.

1. Introduction

The radio spectrum is a scarce and core resource in modern economies. Development of information and communication technology (ICT) markets will depend, to a great extent, on spectrum allocation policy. Hence, governments must assign licenses to use radio frequencies in an efficient way, in their search for global welfare. Traditionally, tenders have been the most widely-used method for allocating the spectrum, although in recent years auctions are being used in many countries. When auctions are conducted, the determination of the specific auction format is a key element in the process. The spectrum licenses have a special characteristic to consider before choosing the auction format, as the bidder's value of a license may depend on the number of licenses he wins. For bidders, licenses might be complements (synergies exist) as well as substitutes. In these circumstances, combinatorial or package auctions [1] (auctions that allow participants to bid both for complete packages of items as well as for individual items) are the most appropriate ones. Combinatorial auctions have already been used to allocate other goods with similar properties such as landing time slots in airports, bus routes, freight transportation services, etc. Nevertheless,

combinatorial auctions have an intrinsic computational problem: finding a feasible combination of bids which maximizes the auctioneer's revenue, that is, solving the winner determination problem (WDP). Finding the WDP is a NP-complete problem [2] and is inapproximable.

The aim of this work is to build an efficient simulator for combinatorial first-price sealed bid auctions. It will be applied in several scenarios that can occur in the allocation of radio spectrum licenses. According to auctioneer decisions about the management of the spectrum, the number of licenses to be auctioned and the number of bidders can be modified. Hence, the WDP must be solved in an appropriate space of time for all scenarios. To this end, several search algorithms have been developed and tested. Generally, there are two approaches for addressing the WDP: optimal and non-optimal algorithms. In this work we focus on the optimal approach. The brute-force based techniques are able to obtain the optimal solution, but these techniques are only practical when the size of the problem instance is small. In other cases, heuristic search techniques are preferable. First, an A* search branch on items (BOI) was implemented, but it was not fast enough. Afterwards, another A* based on a branch on bids (BOB) with a more efficient heuristic function was successfully developed.

The WDP has been dealt with by other authors in several ways. Some researches have restricted the set of combinations for which bidders are allowed to bid on, [3]. Others have design approximation algorithms that compute a sub-optimal allocation [4]. Nevertheless, these two approximations can lead to important economic inefficiencies. Hence, most researches have focused on the design of heuristic search algorithms, see [5], [6], [7], [8], among others.

The remainder of this article is structured in the following manner. The auction model and a detailed explanation of the WDP are described in section 2. Section 3 deals with the mechanisms implemented for solving the WDP. Section 4 describes the experimental scenarios tested and evaluates the results of the techniques developed for each of them. Finally, in section 5, the main conclusions and future work are presented.

2. The combinatorial first-price sealed bid auction and the WDP

The first country in the world to use auctions to allocate spectrum was New Zealand in 1989, [9]. In addition, the first combinatorial auction to assign spectrum was held in Nigeria in 2002, [10]. The Nigerian Communications Commission (NCC) adopted a single round, first-price sealed bid combinatorial auction format. This same auction format has been used by other regulators in other countries,such as the Office of Communications (Ofcom) in UK. In a combinatorial first-price single round sealed bid auction, bidders are allowed to submit in one single round as many bids as they wish for any combination of available lots. For example, if we are auctioning licenses A, B and C, bidders might submit sealed bids for the following combinations: A; B; C; AB; BC; AC; ABC. Then, the auctioneer determines the combination of feasible bids that maximizes his revenues, i.e., solves the WDP. After that, the winners have to pay what they bid for the awarded items.

An important rule to specify in a combinatorial auction is that of the bidding language. The most frequent ones are the *OR* and *XOR bidding languages*. If the *OR bidding language* is used, bidders can win several bids. Nevertheless, the *XOR bidding language* only allows bidders to win one bid. The selection of the appropriate bidding language depends on the properties of the items being auctioned. If items are complements (superadditive values) as well as substitutes (subadditive values), just as in the spectrum licenses, the *OR bidding language* is not the best one. For example, if bidder j bids according to Table 1, with the *OR bidding language* he can win license A for 5 and license C for 3, so he has to pay a total price of 8, although he was only interested in paying 6 for both of them. In these circumstances, where items exhibit substitutability, the *OR bidding language* does not fit the bidders' values. Thus, auctioning spectrum licenses, the most appropriate language is the *XOR bidding language*. With this bidding language, bidders submit exclusive-or (*XOR*) constraints among all their combinatorial bids, so they can only win one bid. This language enables bidders to express their preferences. The *XOR bidding language* is the one used in this paper, which was introduced by [2], [11].

With all the bids submitted, the auctioneer needs to solve

Table 1. Example of bidder *j* personal values for complements and substitutes licenses

Set of items	Personal Value
A	5
B	10
C	3
AB	30 (complementary licenses)
AC	6 (substitutive licenses)

the WDP, i.e., find a feasible combination of bids that maximizes the revenue of the auctioneer. Let $M=\{1, ... , m\}$ be the set of items to be auctioned and $B_{j,i}=\{B_{j,1}, ... , B_{j,n}\}$ the set of bids submitted by bidder j. A bid is a tuple $B_{j,i} = (S_{j,i},$

$p_{j,i})$, where $S_{j,i} \subseteq M$ is a set of items and $p_{j,i}$ the price bidder j is willing to pay for that package. A combination of bids is feasible if it allocates no item more than once.

The WDP is a complex optimization problem that searches the feasible combination of bids which yields a maximum revenue to the auctioneer. Table 2 illustrates an example of bids made by three bidders in a combinatorial sealed bid auction of licenses A, B and C (for simplicity sake the example does not include all possible combinations of bids). In this example, there are multiple feasible solutions to the problem: $\{B_{1,1}, B_{2,1}\}$; $\{B_{3,2}\}$; $\{B_{1,2}, B_{3,1}\}$; $\{B_{1,1}, B_{3,1}\}$ and $\{B_{2,2}, B_{3,1}\}$. Nevertheless, the feasible combination of bids that maximize the auctioneer's revenue is $\{B_{1,2}, B_{3,1}\}$, with revenue 100€ (80€+20€). As the number of bids, bidders and items increases, the problem becomes more complex. In fact, this problem is NP-complete and inapproximable, so advance computational techniques need to be developed in order to solve it.

To solve the WDP with *XOR*-constraints we have included a dummy item that enforces an exclusive-or relation so each bidder can only win one bid, see [7]. In the example presented in Table 2, bidder 1 bids would be: $B_{1,1} = (\{A, Z\}, 10€)$ and $B_{1,2} = (\{A,C, Z\}, 80€)$, where Z is the dummy item that makes both bids mutually exclusive.

The sealed bid auction with package (combinatorial) bidding is a simple, quick and low cost auction.

Table 2. Possible bids with 3 bidders and 3 licenses

Pack ages	Bids done by		
	Bidder j=1	Bidder j=2	Bidder j=3
A	$B_{1,1}=(\{A\},10€)$	-	-
B	-	-	$B_{3,1}= (\{B\},20€)$
BC	-	$B_{2,1}=(\{B,C\},40€)$	-
AC	$B_{1,2}=\{A,C\},80€)$	$B_{2,2}=(\{A,C\},60€)$	-
ABC	-	-	$B_{3,2}=(\{A,B,C\},30€)$

For simplification, the example does not include all possible combination of licenses.

Combinatorial first-price sealed bid auctions are especially appropriate when there are asymmetries between bidders. Moreover, this auction format prevents collusion among participants. Furthermore, in first-price sealed bid auctions, bidders tend to shade their bids; that is, they bid less than their true value. As bidders' revenues are calculated as the difference between their values and the price paid, they need to bid under their value to earn positive profits. The degree of shading depends on the number of bidders; as the number of competitors increases, it reduces the underbid degree, [12].

3. Solving the WDP with BOI and BOB algorithms

Solving the WDP is a necessary step for allocating goods,

such as spectrum licenses, by means of combinatorial auctions. Hence, developing an appropriate technique to work out this complex computation problem within an adequate time is required. When tackling the WDP with tree search two different types of formulation can be used: BOI and BOB [2], [5] and [6]. In this section we present the different heuristic search algorithms developed in this work to solve this problem for both formulations.

3.1. A* search branching on items (BOI)

The first informed search strategy that we have implemented is an A* search based on BOI. The A* algorithm is a well known best-first search algorithm which determines the least-cost path from a given initial node to one goal node. The BOI formulation uses a tree representation with the possible combination of items. Here each combination of items (a node) is represented with a vector of numbers and its length is the number of lots. In this vector the position of the digit corresponds to the item, and the number inside the vector indicates the bidder. The tree has as many child nodes as number of bidders plus one branch for zero (which means that a lot has not been awarded). A tree of two bidders and two lots (A, B) can be seen in Figure 1. The * represents the root of the tree. The leaves of the tree walk through all the possible bid combinations; for example, leaf 22 is translated to bid AB from bidder 2.
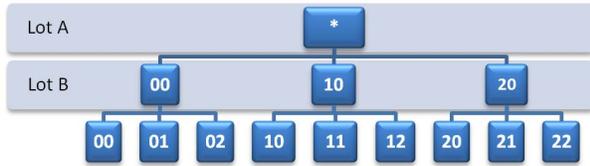

Figure 1. An example tree in BOI formulation

Each node of the tree contains two important values: one is the cost of the bid that it represents, and the other is a heuristic value used to estimate if the path is promising or not. The cost of the bid is obtained with $g(x)$, which calculates the bid, and the heuristic value is calculated with a heuristic function called $h(x)$. This admissible heuristic function follows two steps:

1. It searches the combination that yields the maximum revenue, given the provisional awarded bids.
2. It calculates the remaining possible bids for the rest of the bidders which have not been awarded yet.

For example, using the bids shown in Table 2, one leaf with "10" would represent the bid $B_{1,1} = (\{A\}, 10€)$. Therefore, the value of $g(x)$ for node "10" in that case would be 10€. The value of $h(x)$ would be the result of the sum of these values:

1. The maximum bid that the bidder 1 has combined with

the awarded item, which is $B_{1,2} = (\{A,C\}, 80€)$.
2. The maximum value for the not awarded items. In this example, as item B an C are not awarded yet, the maximum value is $B_{2,1} = (\{B,C\}, 40€)$.

Finally, the A* algorithm uses the result of the sum of $g(x)$ plus $h(x)$ for deciding which is the most convenient path to explore ($f(x)$).

The problem of this heuristic is the computational time needed for calculating remaining possible bids for the rest of the bidders, and it get worse when the number of bidders and bids grows. This problem is referred to in the literature as the that of identifying feasible bids with respect to a given partial solution B' and it has different solutions, such as the discrimination of bids into bins (static), or the use of counters for the corresponding bids (dynamic).

3.2. A* search branching on bids (BOB)

When working with BOB approaches, several differences in representation must be taken into account. Now, as can be seen in Figure 2, the nodes represent bids instead of item combinations. This tree has been generated starting from the example shown in Table 2.
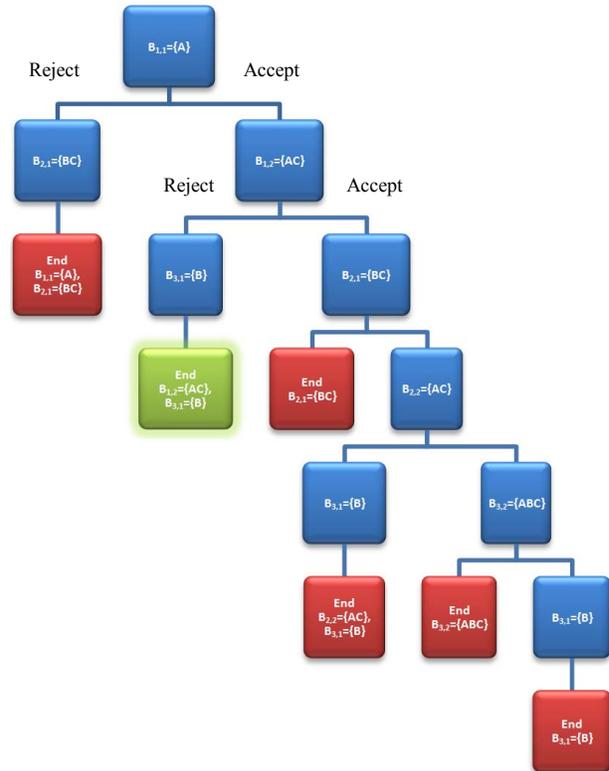

Figure 2. An example tree in BOB formulation

This representation is better than BOI because it allows reducing the search space down to only those bids submitted by the bidders. The search space would be the

same for BOI formulations only in the worst case scenario, when all participants send all possible combinations of bids. However, as can be presumed from Figure 2, the depth of the trees in BOB is usually bigger than in BOI formulations, and it always works with binary trees. The depth of the tree is not constant and it depends on the bids. Hence, a set of bid ordering heuristics that can speed up the search are usually considered [5]. The main idea is to construct many high-revenue allocations early. This measure allows the heuristic to avoid exploring unsuccessful paths. Three different ordering heuristics have been tested in this research with this formulation (all from max to min):

1. Ordering by price, $h_o(price)$.
2. Ordering by number of lots, $h_1(lots)$.
3. Ordering by the mean price per item (mean = p/|B|), $h_2(mean)$.

As the heuristic function developed for the A*(BOI) is not practical with BOB formulation, a new simpler one is proposed. First, the value of $g(x)$ is the cost of the bid, and if the bid is rejected then the value is zero. Second, the $h(x)$ is calculated by adding the higher remaining bids. The main difference between the heuristic used in A*(BOI) and this one is that the bids chosen with higher values do not need to be a feasible solution to the problem. With this measure the heuristic function does not need to be constantly calculating the compatibility between bids. Instead, a sorted list of remaining bids is built for each node.

4. Experimental scenarios and results

In this section, the scenarios tested and the results obtained for each of them are presented.

4.1. Experimental scenarios tested

When allocating spectrum, governments must take important decisions about spectrum packaging and auction design. For example, if an operator needs 24 MHz of spectrum (either contiguous or non-contiguous) to provide its service, the auctioneer can create one or more packages of 24 MHz or design a combinatorial auction that allows the bidder to ask for a combination of 3x8 MHz blocks. Then the auctioneer can face diverse scenarios for selling the spectrum where the number of lots and bidders differs. Therefore, the three scenarios described in Table 3 have been tested in this work.

Table 3. Experimental scenarios tested

Scenario	Nº of lots	Nº of bidders	Bids	Feasible solutions
I	5	5	75	3,125
II	12	6	330	279,936
III	10	7	367	282,475,249

In these experiments we have considered that operators are only interested in contiguous licenses.

All the scenarios have been run 100 times in the same machine: a dual Intel QuadCore® processor, with 8Gb of RAM memory and running Windows® 2003 Server edition. In order to avoid bias, all the bidders' valuations change in each execution following a Gaussian distribution between two limits. The application has been completely developed under .net and it has integration with Microsoft® Excel to import and export the results, see Figure 3.
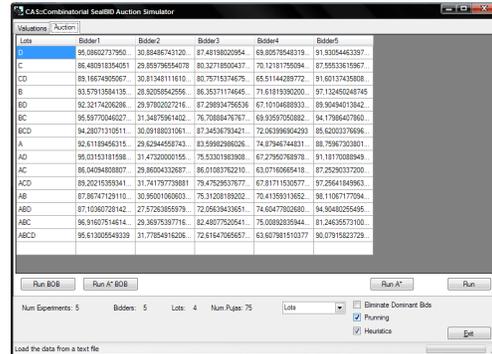
Figure 3. Application developed to solve the WDP

However, as measuring the execution time depends on the programming and the workload of the machine, we have decided to count the number of the search tree explored nodes to evaluate the success of the techniques developed.

4.2. Results obtained

With all the bids submitted, the auctioneer needs to compute the WDP. Solving the WDP means that the system must find the feasible combination of bids which guarantees the maximum revenue to the auctioneer.

The first approach made was the brute-force. It was only tested so as to have a point of reference for the incoming developments. Afterwards, an A* with BOI formulation was implemented. Its results in terms of visited nodes with respect to brute-force were obviously improved (see Table 4). However, the execution time was far from been useful for scenarios II and III (7 hours, and more than 48 hours respectively).

Table 4. Explored nodes for brute-force and A* BOI

| Scenario	Brute-Force	A* BOI	Percentage
I	3,125	1,555	49.76%
II	279,936	137,257	49,03%
III	282,475,249	n/a	n/a

Those results encouraged us to improve the system with the A* based on BOB formulation and with a more effective heuristic. The results of the improvements made by this approach can be found in Table 5 and Figure 4.

Table 5. Explored nodes reduction for A* BOB compared to total number of nodes

| Scenario	$h_o(price)$	$h_1(lots)$	$h_2(mean)$	Average
I	62,95%	64,42%	63,69%	63,69%
II	44,56%	51,14%	44,70%	46,80%
III	99,80%	99,66%	99,61%	**99,69%**
Avg	69,10%	**71,74%**	69,33%	

A reduction of 80% means that only the 20% of the total number of nodes (in BOB formulation) are needed to find the optimal solution.

Analyzing the results and attending to the number of visited nodes needed to find the solution, the best ordering heuristic for the A*(BOB) approach is (in average) the $h_1(lots)$. However, the results obtained by $h_o(price)$ and $h_2(mean)$ are very similar for scenarios I and II, but when the scenario turns complex, the $h_o(price)$ improves.

	I	II	III
■ Prices	1354	9464	46840
■ Lots	1300	8340	79133
■ Mean	1327	9440	89558
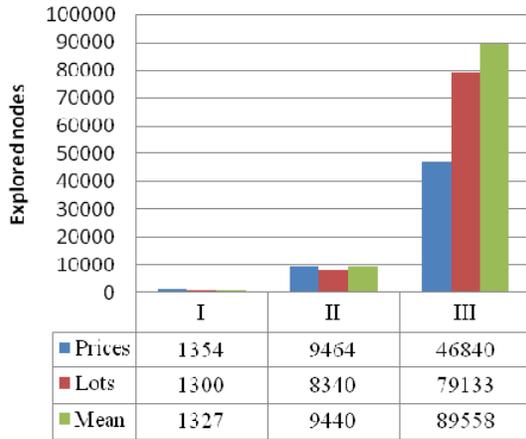
Figure 4. Average explored nodes for each scenario and ordering heuristic.

As can be seen in Figure 4 the absolute value of explored nodes (in average) is remarkably lower for the prices bid ordering heuristic in the third scenario (the most complex). If we take a look at the time consumption of each technique (Figure 5), the $h_o(price)$ heuristic visibly outperforms the rest. This is because the heuristic function designed for the A*(BOB) is based on maximum prices, and the application of other types of sorting means more computational effort to deal with this task.

	I	II	III
■ Prices	0,02	0.24	1
■ Lots	0,04	0.26	3,4
■ Mean	0,03	0.35	4,5
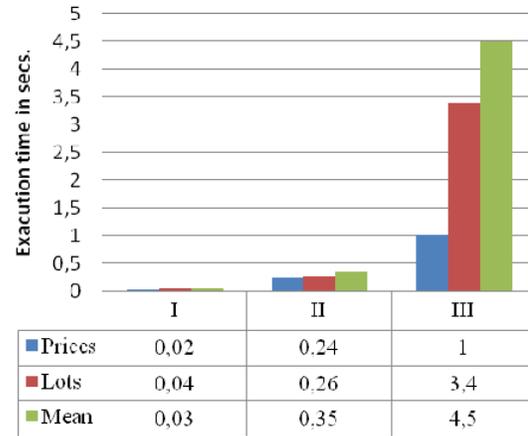
Figure 5. Average execution time for each scenario and ordering heuristic.

5. Conclusions and future work

This is an introductory work about how to solve the WDP. We have successfully developed a system which is capable of finding the exact optimal solution for the WDP for *XOR bidding language* in a reasonable execution time (1 second for the most difficult scenario). Therefore, this system can be used in any combinatorial auction where the WDP needs to be solved.

After having developed the brute-force, the A*(BOI) and the A*(BOB) approaches, we conclude that the key factors for developing an efficient algorithm to solve the WDP are:

1. Determining an appropriate priority of bids.
2. Developing an efficient technique for identifying feasible bids given a partial solution. (This is critical.).
3. Implementing techniques for detecting unpromising paths as soon as possible.

The results obtained revealed that the heuristic function proposed for the A* BOB is simple and very effective. In fact, the most complex scenario (III) needs one second on average to solve the optimum solution. Thus, in cases when not very large instances need to be solved, this can be an excellent approach. Nevertheless, if the reader needs more performance, the commercial software ILOG® CPLEX outperforms most of the existing optimal algorithms for the multi-unit case with respect to running time. Only CABOB performs comparably well [6], and it is not so easy to implement as A*.

As a future work, in order to validate the results, we want to make an exhaustive experimentation with instances generated with the Combinatorial Auction Test Suite (CATS) software, [13]. Furthermore, these tests will allow

us to compare and validate our results with other approaches found in the literature and test them with CATS instances.

6. References

[1] Cramton, P.; Shoham, Y. and Steinberg, R. (2006). Combinatorial auctions. The MIT Press (Cambridge).

[2] Sandholm, T. (2002a). Algorithm for Optimal Winner Determination in Combinatorial Auctions. Artificial Intelligence, 135, pp. 1-54.

[3] Rothkopf, M.H., Peckec, A. and Harstad R.M. (1998) Computationally Manageable Combinatorial Auctions. Management Science, 44 (8), pp. 1131-1147.

[4] Halldorsson, M. M. (2000) Approximations of weighted independent set and hereditary subset problems. Journal of Graph Algorithms and Applications, 4 (1), pp. 1-6.

[5] Sandholm, T. and Suri, S. (2003). BOB: Improved winner determination in combinatorial auctions and generalizations. Artificial Intelligence, 145, (1-2), pp 33-58.

[6] Sandholm, T., Suri, S. Gilpin, A. and Levine, D. (2005). CABOB: A fast optimal algorithm for winner determination in combinatorial auctions.

[7] Fujishima, Y. Leyton-Brown, K. and Shoham, Y. (1999) Taming the computational complexity of combinatorial auctions: optimal and approximate approaches, in Proceedings of the IJCAI'99, pp. 548-553.

[8] Mito, M. and Fujita, S. (2004). On heuristics for solving winner determination problem in combinatorial auctions. Journal of Heuristics, 10, pp. 507-523.

[9] Grünwald, A. (2001). Riding the US wave: spectrum auctions in the digital age. Telecommunications Policy, vol. 25 (10-11), pp.719-728.

[10] Kobodt, C., Maldoom, D. and Marsden, R., (2003), The First Combinatorial Spectrum Auction: Lessons from the Nigerian Auction of Fixed Wireless Access Licenses, DotEcon DP 03/01.

[11] Sandholm, T. (2002b). eMediator: A next generation electronic commerce server. Computational Intelligence, 18, pp. 656-676.

[12] Krishna, V. (2002). Auction Theory. San Diego: Academic Press.

[13] Leyton-Brown, K., Pearson, M., & Shoham, Y. (2000), Towards a Universal Test Suite for Combinatorial Auction Algorithms. In the Proceedings of ACM Conference on Electronic Commerce (EC-00).