ELEKTROTEHNIŠKI VESTNIK 78(4): 193-197, 2011
ENGLISH EDITION

Time and space complexity of combinatorial auction in Telecommunication

Jernej Virant, Andrej Košir

University of Ljubljana, Faculty of Electrical Engineering, Tržaška 25, 1000 Ljubljana, Slovenia
E-mail: *andrej.kosir@ldos.fe.uni-lj.si, jernej.virant@gmail.com*

Keywords: combinatorial auction, optimization, operational research, telecommunication

Abstract

In this paper we evaluate time and space complexity of combinatorial auction solver. The application of combinatorial auction optimization problem in the area of telecommunication is presented. Using random problem instance generator we experimentally measured execution times and the amount of memory used for wide range of optimization problem sizes given as number of goods and number of bids. Results are reported as running times in seconds and memory used in kilobytes on a personal computer. We conclude that real world combinatorial auction problems can be solved on a personal computer.

1 INTRODUCTION

A combinatorial auction is an economic-based resource allocation mechanism where a user (buyer) can place bids for each of the possible combinations of resources instead of bidding for each resource separately. That kind of auction increases the economic efficiency and maximizes the revenue comparing to the classic auction.

Among other applications, simple combinatorial auctions have been used in estate auctions, transport routes and in the allocation of radio spectrum for wireless communications. In the radio spectrum allocation scheme multiple users bid for the required number of time slots and the allocations are done with the primary objective of maximizing the number of satisfied users in the system. In the second round of the auction the unallocated resources are allocated to users in the way that the system throughput is maximized. This kind of allocation mechanism avoids exposure problem where algorithms fail to satisfy the minimum slot requirements of the users due to substitutability and complementary requirements of user slots [1]. Combinatorial auction can also be used for the real time sub-carrier allocation scheme for OFDM transmission [2].

A computational problem is how to efficiently determine the goods (items, resources) allocation. Given a set of bids in a combinatorial auction, the goal is to find an allocation of items to bidders that maximize the auctioneer's revenue. This is called the winner determination problem and belongs to a group of NP complete problems. There are some special case auctions (with constraints on the set of bids) where a polynomial-time solution does exist [3].

The goal of this paper is to experimentally evaluate the time and space complexity of combinatorial auction solver algorithm applied to a selected group of optimization problems.

The rest of the paper is organized as follows: In Section 2 we describe the combinatorial auction problem in detail to the required extent. Combinatorial auction implementation in telecommunication is briefly presented in Section 3. Experimental results are given in Section 4, followed by conclusions in Section 5.

2 OPTIMIZATION TASK AND COMBINATORIAL AUCTION PROBLEM

In this section, we describe the combinatorial auction model and formulate it as an optimization problem. We also briefly describe the winner determination algorithm.

Optimization problem of the general combinatorial auction is to determine the winning bids (that is to solve the winner determination problem). The set of items (resources) is denoted by $M = \{1,2,\dots,m\}$. Each bid consists not of a single item but from a subset of items, denoted by $\boldsymbol{b}_j, j = 1 \dots n$. A price $p_j = p(\boldsymbol{b}_j)$ is assigned to each bid by the auctioneer. Note that this price can be or cannot be the sum prices of items contained in the bid $\boldsymbol{b}_j$. Typically this is not the case since different aspects of market regulation can be carried out through bid price assignments.

Input parameter for the winner determination algorithm is a set of bids with costs, where each element consists of a pair $(\boldsymbol{b}_j, p_j)$. Optimization task is to maximize the revenue where each resource can be allocated to one bid only [5],

Received October 24, 2011
Accepted November 7, 2011

$$max\ \sum_{j=1}^{n} p_j\, x_j\,,\quad s.t.\quad \max \sum_{j|i\in S_j} x_j \leq 1,\ x_j \in \{0,1\}.$$

Note that binary variables x_j are selector variables and the expression $\sum_{j=1}^{n} p_j\, x_j$ is the price of all sold bids, which is maximized subject to (s.t) constrain that each item can be sold at most once (that is x_j representing the same item can sum up to one). The solution algorithm output is a vector $\boldsymbol{x} = (x_1, \dots, x_m)$ indicating the selected bids.

This general problem of winner determination is computationally complex (NP complete) and is not approximable [6].

Optimal algorithms for solving the winner determination problem use search trees. The goal of search trees is to make a set of decisions for all bids (e.g. for each bid, deciding whether to accept or reject it). Once the search has finished, the optimal set of decisions is found and proven optimal. However, in practice, the space is much too large to be searched exhaustively. Algorithms use special techniques (e.g. branch and bound) that selectively search the space to find a solution. For special case combinatorial auctions (e.g. bound on the number of resources in bids) polynomial time approximation algorithms are being used [5].

There are also so called “any-time solvers” available where a quick stochastic optimization solver is run a large amount of times using smart random walks what can lead to a very good solution. An advantage of this approach is that the execution can be terminated any time (this is the origin of their name) and we still get an approximation to a solution. The drawback is that there is no indication whether the solution is optimal, close to optimal or even far from optimal.

3 COMBINATORIAL AUCTION IN TELECOMMUNICATION

The most common use of combinatorial auction in telecommunications is the allocation of carrier resources. Among others, known applications of combinatorial auction in real world are allocation of radio spectrum for wireless communications [1] and OFDM sub-carrier allocation [2].

The allocation of radio spectrum uses an algorithm based on reverse auction (NP complete problem) which searches through all of the possible combinations (search tree). OFDM sub-carrier allocation uses an algorithm that selectively searches the search tree.

Let us point out that that the main reason why combinatorial auction is chosen as a resource selling mechanism is the fact that it allows the implementation of required regulations to the telecommunication market. This is done through constrained resource selling accomplished by the bid formulation. It is well known that telecommunication market is highly complex since it involves several connected subsystems such as communication infrastructures, telecommunication services and users of essentially different interests and communities. Besides, the impact of telecommunication services to the economy and to the whole society is known to be important. Therefore, there is an interest to regulate the ownership of resources such as frequency spectrum and others.

4 EXPERIMENTAL RESULTS

Combinatorial auction solver algorithms are essentially independent of the nature of the problem represented by the underlying combinatorial auction. Therefore, a randomly generated test data can be used to evaluate time and space complexity of combinatorial auction solvers. We used random optimization problem instance generator CATS [6] since a high level of flexibility in evaluation experimental designs is enabled in this way.

Standard implementation of branch-and-bound combinatorial auction solver CASS [6] was used as a auction solver. It allows simple command line parameters based control suitable for script testing. Test suits CATS and CASS are widely used by the research community.

Since a huge number of runs are required by the testing designs, the only manageable option of performing experiments is to use test scripts. Such high number of runs is required due to the random generation of optimization problem instance and due to relatively large ranges of parameters of problem instances to be controlled during the experiment.

We tested time and space complexity of winner determination algorithm on a personal computer. We implemented test scripts in PowerShell environment which generate and solve combinatorial auction problems in loops according to the chosen experimental designs, and measured the elapsed time and the amount of memory used by the solver algorithm.

We made the following types of experiments reported later in this section:

- Time complexity of winner determination;
- Time complexity of winner determination for same size combinatorial auctions;
- Time complexity of winner determination for the same combinatorial auction;
- Space complexity of winner determination;
- Space complexity of winner determination for same size combinatorial auctions;
- Space complexity of winner determination for the same combinatorial auction.

Selected details on problem instances and experimental results are given in subsequent subsections.

4.1 Materials and methods

Tests were conducted on the personal computer HP Compaq nx6310 with the operating system Windows XP. Standard benchmarking approach using test scripts were applied. Generated combinatorial auctions with their optimal allocations and test reports were automatically traced in order to verify the regularity and the repeatability of experiments. Auctions were generated by CATS suite and solved by CASS suite [6]. CATS can generate different types of auctions; we used two of them in various tests as indicated below.

Combinatorial auctions were generated with different input parameters and were solved while measuring the elapsed time and memory usage. Input parameters were the number of bids and the number of goods (size of the problem). We incremented the size of input parameters until a certain computation time threshold (1000 seconds) has been reached. In practice, usually there is no need to provide real time or near real time solutions. Therefore we have chosen 1000 seconds as a reasonable solving time for relatively small problems.

A relevant parameter of a given combinatorial auction instance is also the probability distribution of bids and goods. We refer it as an internal distribution here. It can be selected by command line parameters supplied to the packet CATS. We have performed our tests using any distribution and L6 distribution. It has exponential goods distribution and normal price distribution. For further details on probability distribution L6 see [6].

4.2 *Script testing*

We used script testing because it is programmable, very useful for repetitive tests and very flexible. Once we wrote a script we change the input parameters via property files. We wrote two scripts; one for time complexity which measured the elapsed time and one for space complexity which measured memory usage through peak working set property of a process in Windows operating system.

4.3 *Time and space complexity experiments*

We have done several tests to examine time and space complexity of the winner determination solver. Our objective was to examine the influence of input parameters (number of goods and bids) on computation time and memory usage. We also examined the deviation of the winner determination computation time and memory usage of same size auctions and the same auction as a result of different randomly generated instances and other parameters. This is necessary to assure proper measurement characteristics.

In the following subchapters results of six experiments are presented.

4.3.1 *Experiment 1: time complexity*

We have discovered that not only the size of the optimization problem but also the probability distribution of prices and goods affects the time and space complexity. As shown in Figure 1 only relatively small combinatorial auctions with arbitrary price distribution can be solved.

It is difficult to make any claim on probability distributions involved in real world combinatorial auction problems in the area of telecommunication. However, real problems can involve large number of goods (items, resources) but not a large number of bids since there are only a limited number of bids that meet technical, legislation and commercial (market) constraints and are acceptable for the investor. Therefore, results shown at Figure 1 are not limiting to the high extent for real world applications. Besides, number of bids included in the optimization process can be pre-selected in practice by applying simple exclusion rules according to technical and legislation limitations. On the other hand, numbers shown at Figure 1 would not improve dramatically if only a stronger hardware platform would be used and no other improvement to the algorithm such as parallelization would be made.

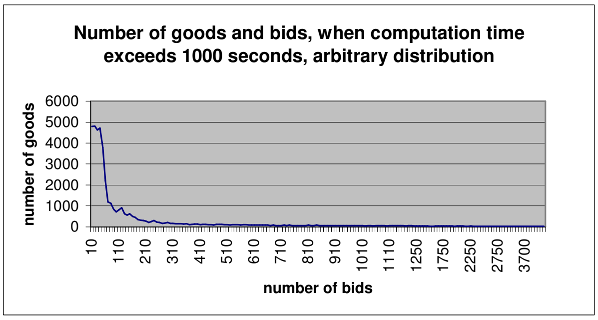

Figure 1. Combinatorial auction size when computation time exceeds 1000 seconds, arbitrary internal distribution.

Larger combinatorial auctions can be solved with the L6 distribution as shown in Figure 2. The number of goods is steadily decreasing by the number of bids; small variations are mainly due to the variations in randomly generated problem instances.

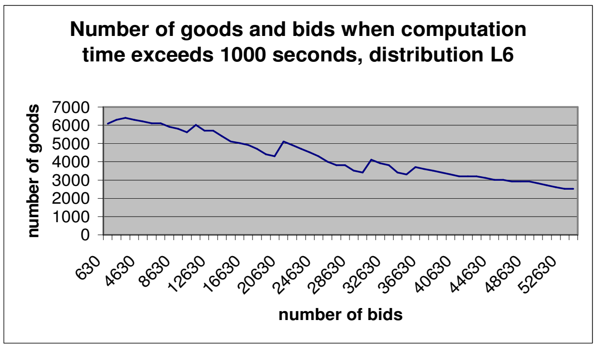

Figure 2. Combinatorial auction size when computation time exceeds 1000 seconds, L6 internal distribution.

Figures 3 and 4 show that computation time is non-linear with the increment of input parameters and exhibits exponential growth. That holds true for all distributions. In practice this means that using stronger hardware would not considerably improve figures.

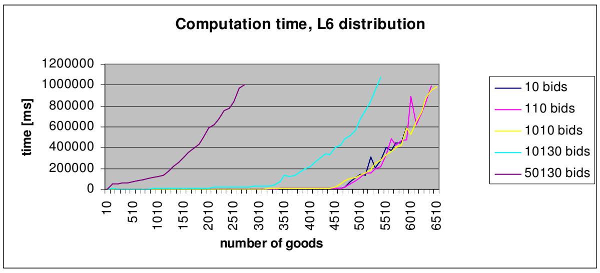

Figure 3. Combinatorial auction computation time with the increase of goods, internal distribution L6.

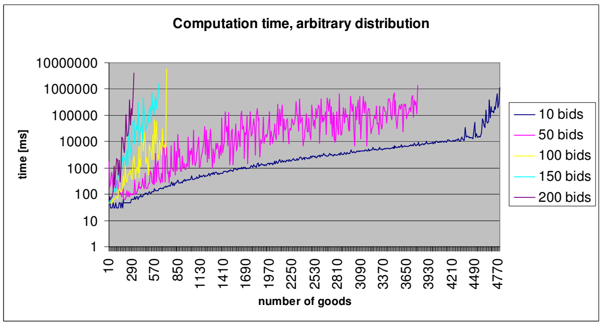

Figure 4. Winner determination computation time with the increase of goods, arbitrary internal distribution.

4.3.2 *Experiment 2: Time complexity of same size auctions*

We analyzed the computation time of same size auctions. Figure 5 shows that there are significant differences among same size auctions. That holds true for both distributions.

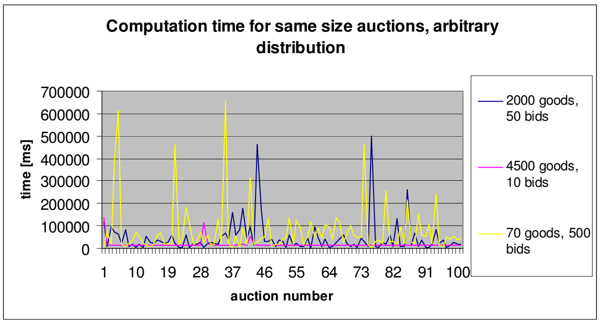

Figure 5. Winner determination computation time of same size auctions.

Different instances of the same size auctions differ greatly in search trees which results in different computation times.

4.3.3 *Experiment 3: Time complexity of the same auction*

We analyzed the computation time of the same auction. We discovered there are some differences because of other processes that ran on the computer. That holds true for both distributions. However, we believe that the size of exhibited variations does not compromise experimental measurements presented in this paper.

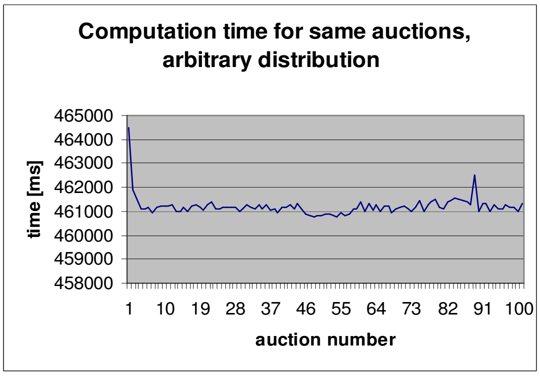

Figure 6. Winner determination computation time of the same auction

4.3.4 *Experiment 4: Space complexity*

We discovered that memory usage is linear with the increment of bids (Figure 7) and nonlinear with the increment of goods for both distributions (Figure 8).

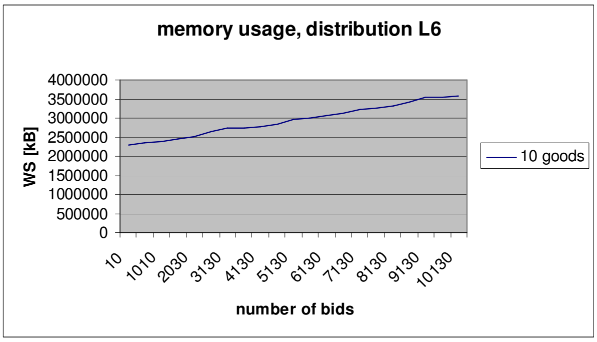

Figure 7. Memory usage of winner determination, *internal* distribution L6.

We also discovered that the memory usage is independent of distribution for auctions with same number of bids. Memory usage does depend on distribution for auctions with same number of goods.

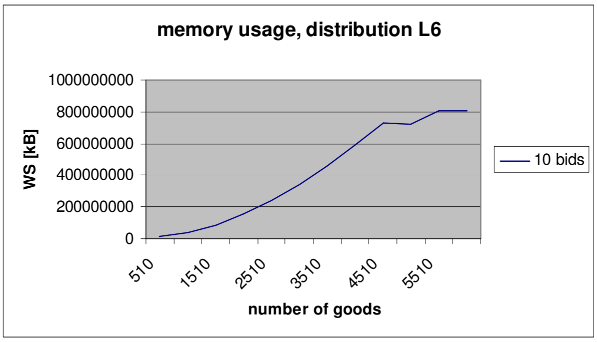

Figure 8. Memory usage of winner determination, *internal* distribution L6.

The increment of memory usage with number of bids and with the number of goods is steady. To be honest, we do not have any good explanation of course of the memory usage after the number of goods value 4500.

4.3.5 *Experiment 5: Space complexity of same size auctions*

We analyzed the memory usage for computation of same size auctions. Similar to the computation time, memory usage depends on search trees. Results are shown in Figure 9. That holds true for both distributions. However, the variations observed are lower comparing to one of computation time.

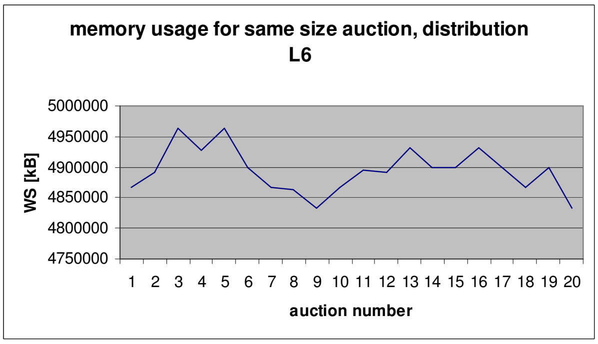

Figure 9. Memory usage of winner determination for same size auctions, internal distribution L6.

4.3.6 *Experiment 6: Space complexity of same size auction*

We analyzed the memory usage for the winner determination of the same auction. Results are shown on Figure 10. We did not discover any significant deviation. That holds true for both distributions.

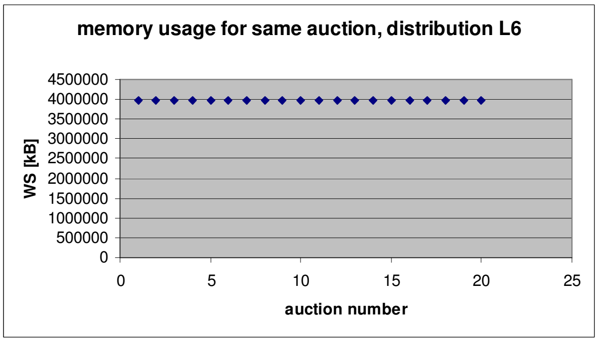

Figure 10. Memory usage of winner determination for the same auction, internal distribution L6.

4.4 *Discussion on results*

We discovered that the computation time and memory usage for winner determination of combinatorial auctions depends on the distribution of prices and bids in auctions. The computation time for winner determination significantly depends on the number of bids and goods, but the computation time for the distribution L6 depends more on the number of goods than bids. The number of goods in a combinatorial auction that can be solved on a personal computer is about the same for both distributions. The number of bids in a combinatorial auction that can be solved on personal computer also depends on the distribution. We also discovered high deviation in the winner determination problem solution of same size auctions.

The memory usage for winner determination of combinatorial auction depends on the number of goods and bids. It grows nonlinearly with the increase of goods and approximately linearly with the increase of bids. It grows with the increase of goods independently of distribution. There is also high deviation in the memory usage of winner determination of same size auction.

5 Conclusion and Further Work

We analyzed the computation time and memory usage for winner determination of combinatorial auctions for two different distributions. We also analyzed the deviation in computation time and memory usage for same size auctions and the same auction. We had some problems while testing. We could not use some of the more interesting distributions, because CATS encountered problems with the generation of combinatorial auctions. This remains our goal for the future. We would also like to determine if computation time depends more on the distribution of prices or the distribution of goods.

Literature

[1] Pal S, Kundu R, Chatterjee M, Das S.K, Combinatorial Reverse Auction-Based Scheduling in Multirate Wireless Systems. 2007 Available at: http://www.cs.ucf.edu/~mainak/papers/TC-auction.pdf (4.6, 2011).

[2] Wong C.Y, Tsui C.Y, Cheng R.S, Letaief K.B, A real time sub-carrier allocation scheme for multiple access downlink OFDM transmission 1986, Available at: http://repository.ust.hk/dspace/bitstream/1783.1/2553/1/real.pdf (4.6, 2011).

[3] Cramton P, ShohamY, Steinberg R, Introduction to combinatorial auctions, 2006, Available at: http://www.cramton.umd.edu/papers2005-2009/cramton-shoham-steinberg-combinatorial-auctions-introduction.pdf (4.6, 2011).

[4] Sandholm T, Optimal winner determination algorithms, 2006 Available at: http://www.cs.duke.edu/courses/cps173/spring10/optimal_wdp_chapter.pdf (4.6, 2011).

[5] Lehmann D, Muller R, Sandholm T, The winner determination problem, 2006 Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.63.8065&rep=rep1&type=pdf (4.6, 2011)

[6] Leyton-Brown, K, Downloads related to my work, http://www.cs.ubc.ca/~kevinlb/downloads.html (4.6, 2011).

Jernej Virant is with Generalna policijska uprava Slovena. His work includes the introduction of intelligent information systems into information telecommunication systems.

Andrej Košir is a professor at the University of Ljubljana Faculty of electrical engineering. His research interest includes operational research in telecommunication, user modeling, user adaptation and social signal processing.