On the Computational Feasibility of Combinatorial Auctions on all Subsets

Sven de Vries* & Rakesh Vohra†

June 20, 2000

Abstract

The FCC has announced in the auction of licenses in the 747-762 and 777-792 band (auction No. 31) that bids on packages of licenses be allowed. Under the current proposal (see the Public Nortice concerning Auction 31 issued on May 18th, 2000) the set of allowable packages on which bids may be submitted, is restricted in a particular way. One of the arguments made in favor of this restriction is that it makes the problem of determining a provisionally winning set of bids easy. Here we examine the computational difficulty of determining a provisionally winning set of bids when bids on all possible packages are allowed. We argue that there is no computational obstacle to allowing bids on all possible packages.

*Zentrum Mathematik, TU München, D-80290 München, Germany.
e-mail: devries@ma.tum.de

†Department of Managerial Economics and Decision Sciences, Kellogg School of Management, Northwestern University, Evanston IL 60208. e-mail: r-vohra@nwu.edu

Summary

Auction 31 will involve 12 licences and thus $2^{12} - 1$ possible packages on which bids could be placed. Given a collection of bids on different packages, the problem of determining a feasible collection of bids that will maximize revenue belongs to a class of well studied integer optimization problems called set packing problems.

To get a sense of just how hard a set packing problem of the kind implied by Auction 31 would be, we generated some artificial instances. Here we report on those instances that we believe are the hardest, in a computational sense.[1]

Let $M = \{1, 2, \ldots, 12\}$ be the set of licenses and fix a parameter $\epsilon \geq 0$. For each $i \in M$ we assign a number v_i drawn uniformly from $U[90, 110]$. For each $S \subset M$ we draw an α_S from $U[-\epsilon, \epsilon]$. The highest bid, $v(S)$, on a subset S of licenses was chosen to be

$$(1 + \alpha_S) * \sum_{i \in S} v_i.$$

50 instances for various values of ϵ of an optimization problem were generated in this way. Thus our test problems involve *every* possible subset of 12 licenses giving rise to an optimization problem with $2^{12} = 4096$ variables.

The construction produces an optimization problem with lots of near optimal solutions. This forces an algorithm for solving such problems to spend its time checking and eliminating many solutions before certifying a particular solution as optimal.

Of course, one could try to model first the valuations of various bidders and then take the maximum of their valuations to obtain the profile of maximum bids. If in this profile the average price per item in large sets was substantially larger than for small sets, then it is apparent that an optimal solution can only involve those large sets. Therefore determining the collection of provisionally winning bids is difficult, if the average price per item is almost the same for all sets except for noise.

The resulting optimization problems were solved using the branch-and-bound solver[2] of the CPLEX 6.5 callable library (see ILOG (1997)). We wrote the (straight forward) program in C++. The running-times we report were obtained on an SGI O2 computer with one MIPS R10000 processor at 175MHz with 256MB of main memory (though less memory was necessary) running IRIX 6.5.5m. No finetuning of the software was done.

[1]Our thanks to Robert Weber for suggesting this class.

[2]In fact at all nodes, clique-cuts might be added.

Every instance generated was solved in under 20 seconds. The average time was about 10 seconds (see the section below for more details). For this reason we believe that the problem of determining the provisionally winning set of bids, when bids on all subsets are allowed, in Auction 31, can be routinely and quickly solved by utilizing commercially available, robust software.[3] This claim also has implications for a number of the schemes that have been proposed for computing minimum bids. Furthermore, as there is an extensive literature devoted to set packing problems, it is very likely, that by using those methods and finetuning the runtimes can be reduced and auction involving even more sets (for more items) can be solved rapidly.

We are aware of only two other arguments against allowing bids on all possible packages of licences. The first is that the row and package restriction makes the strategic problems faced by bidders more transparent. Yes, but so is packaging all licenses into a single bundle and auctioning it off. An overly elaborate auction may leave 'money on the table' as will an overly simple one. The issue is the trade-off between the two which as far as we know has not been addressed explicitly.

The second argument relies on the proposers inability to conceive of economically interesting packages that do not not follow the row and column restriction. We can think of one. A bidder interested in a package of 4 licenses. Under the current restrictions such a bidder would have to submit 4 single item bids or a bid for any allowable superset of her 4 licences.

The Data

The first figure (below) conveys information about the solution times.

The branch-and-bound algorithm used in CPLEX involves the solution of a number of sub-problems identified with branch-and-bound nodes. The figure below summarizes the number of such sub-problems that needed to be solved.

The remaining figures provide more detailed information about means, modes and medians.

Figure 1 shows the average running time for various values of ϵ. Figure 2 shows the average number of branch-and-bound nodes the solution took. Figures 3, 4 provide more details of the distribution of running times and necessary nodes for our experiments.

[3] Results concerning other generation schemes can be found in de Vries and Vohra 2000.

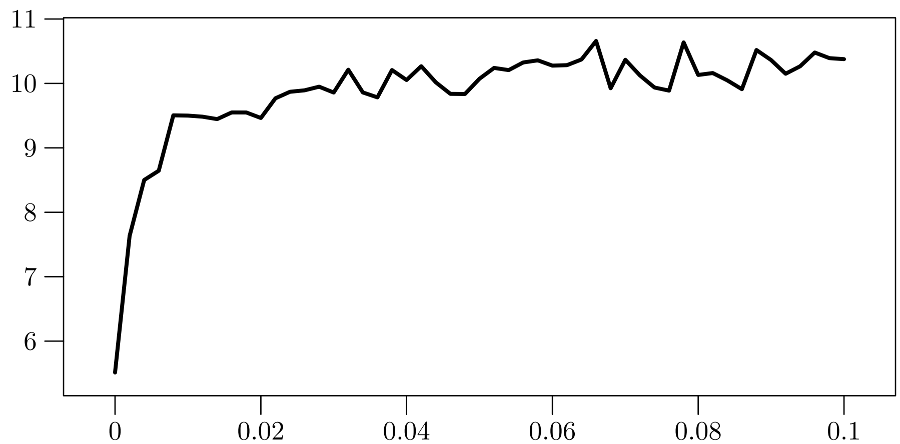

Figure 1: The abcissa depicts the size of the parameter ϵ and the ordinate depicts the average running time (in seconds) (out of 50 repetitions) necessary for solving the auction problem on 12 items and 4096 bidsets.

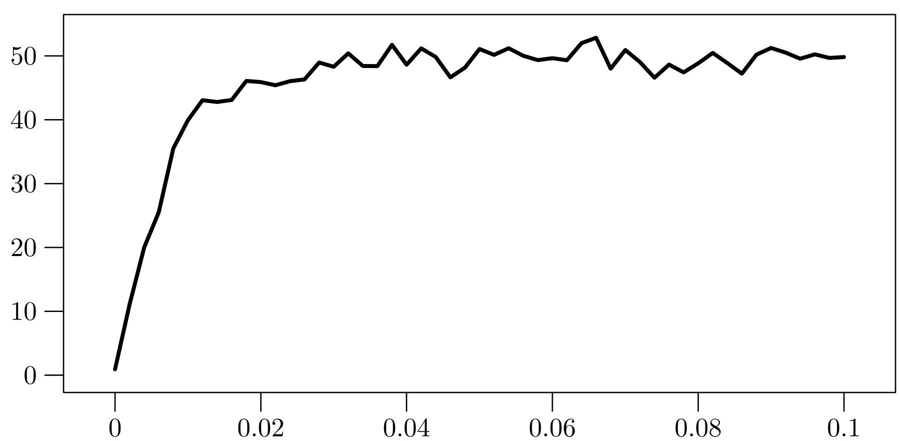

Figure 2: The abcissa depicts the size of the parameter ϵ and the ordinate depicts the average number (out of 50 repetitions) of branch-and-bound nodes necessary for solving the auction problem on 12 items and 4096 bidsets.

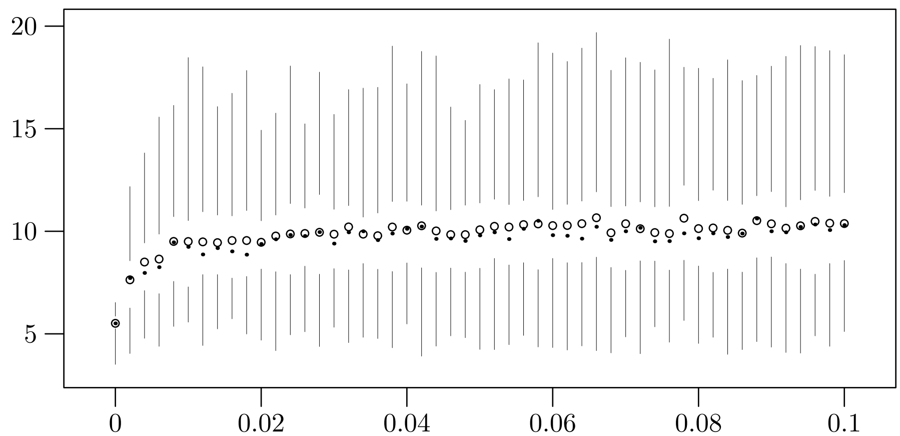

Figure 3: The abcissa depicts the size of the parameter ϵ and the ordinate depicts the statistics about the running time (in seconds) (out of 50 repetitions) necessary for solving the auction problem on 12 items and 4096 bidsets. The vertical lines indicate (from top to bottom) maximal value, upper quartile, lower quartile and minimum value. The dot (.) marks the median, the hollow circle ($\circ$) marks the mean.

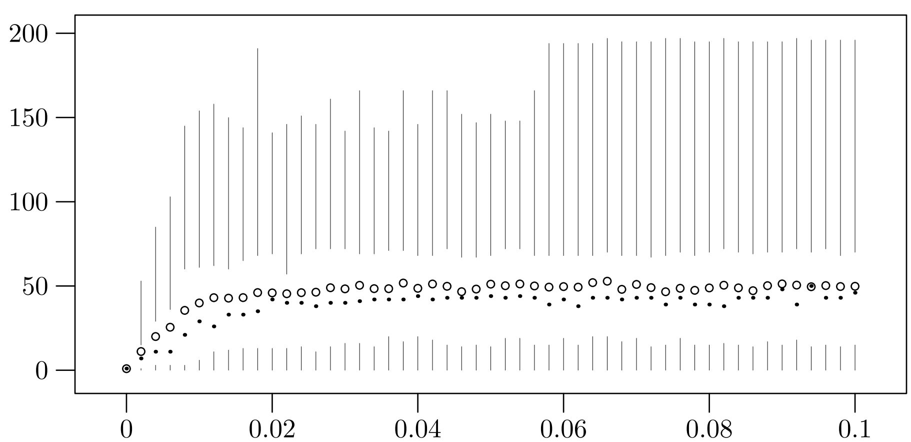

Figure 4: The abcissa depicts the size of the parameter ϵ and the ordinate depicts the statistics about the number of branch-and-bound nodes (out of 50 repetitions) necessary for solving the auction problem on 12 items and 4096 bidsets. The vertical lines indicate (from top to bottom) maximal value, upper quartile, lower quartile and minimum value. The dot (.) marks the median, the hollow circle ($\circ$) marks the mean.

REFERENCES

1. ILOG Inc. CPLEX Division, 930 Tahoe Blvd. # 802-279, Incline Village, NV 89451-9436, USA, "Using the CPLEX Callable Library" (information available at URL `http://www.cplex.com`) 1997.

2. Sven de Vries and Rakesh V. Vohra, "Combinatori al Auctions: A Survey", manuscript, 2000.