

A Service of



Leibniz-Informationszentrum Wirtschaft Leibniz Information Centre for Economics

Drexl, Andreas; Jørnsten, Kurt; Knof, Diether

Working Paper — Digitized Version Column aggregation-based pricing combinatorial auctions

Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 624

Provided in Cooperation with: Christian-Albrechts-University of Kiel, Institute of Business Administration

Suggested Citation: Drexl, Andreas; Jørnsten, Kurt; Knof, Diether (2007) : Column aggregation-based pricing combinatorial auctions, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 624, Universität Kiel, Institut für Betriebswirtschaftslehre, Kiel

This Version is available at: https://hdl.handle.net/10419/147677

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.



WWW.ECONSTOR.EU

Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel

No. 624

Column aggregation-based pricing combinatorial auctions

Andreas Drexl, Kurt Jørnsten, Diether Knof

June 2007

© Do not copy, publish or distribute without authors' permission.

Institut für Betriebswirtschaftslehre, Christian-Albrechts-Universität, Kiel, Germany, andreas.drex1@bwl.uni-kiel.de

Norwegian School of Economics and Business Administration, Department of Finance and Management Science, Bergen, Norway, kurt.jornsten@nhh.no

Institut für Mathematik, Christian-Albrechts-Universität, Kiel, Germany, dknof@computerlabor.math.uni-kiel.de

Abstract

Combinatorial auctions permitting bids on bundles of items have been developed to remedy the exposure problem associated with single-item auctions. Given winning bundle prices a set of item prices is called market clearing or equilibrium if all the winning bids are greater than or equal and if all the losing bids are less than or equal to the total price of the bundle items. However, the prices for individual items are not readily computed once the winner determination problem is solved. This is due to the duality gap of integer programming caused by the indivisibility of the items. In this paper we propose a family of linear programming models the optimal solution of which is integral "almost always", producing linear prices at the expense of having reduced cost zero for the aggregate winning bids only. We provide a computational proof of this conjecture by an in-depth experimental study of 18,000 instances from the combinatorial auction test suite (CATS; see [13]). Summarizing this analysis we have linear prices for all but five of the whole bunch of instances and, hence, there exists a linear price function that supports the optimal allocation of winning bundles.

Keywords: Combinatorial auctions, set packing, dual prices, linear programming

1 Introduction

Combinatorial auctions are auctions where single bids on multiple distinct items are allowed. Single item auctions have been the topic of intensive research for many years and particular incentive compatible efficient auctions have been developed. In some markets, however, a participant's valuation of an item depends significantly on which other items the participant acquires. Items can be substitutes or complements, and the valuation of a particular bundle of items may not be equal to the sum of the valuations of the individual items, that is, valuations are not additive. In this setting, economic efficiency is increased by allowing bidders to bid on combinations of items, which is exactly what a combinatorial auction does. Due to this increased economic efficiency combinatorial auctions have become the focus of extensive research in recent years.

One major obstacle in the design of combinatorial auctions is the solution of the winner determination problem. Winner determination is equivalent to the weighted set packing problem which belongs to the class of NP-hard integer programs (for a detailed exposition of this issue see Rothkopf, Pekeč and Harstad [21]). In practice combinatorial auctions usually are applied in a multi-round setting; see, e.g., Parkes [18]. During each round, bidders submit bids on packages and then the auctioneer determines a provisional allocation of bundles to bidders. In this case dual information may be useful for bidders as marginal values that enable bidders to bid more efficiently in subsequent rounds.

When the linear programming relaxation of the winner determination problem for a combinatorial auction does not possess an integer solution so far it is not possible to achieve a linear price function that supports the optimal allocation of winning bundles. In this paper, however, we propose a family of linear programming models the optimal solution of which is "almost always" integral and, hence, there exists a linear price function that supports the optimal allocation of winning bundles. These linear prices are derived at the expense of having reduced cost zero for the aggregate winning bids only.

This means that some of the winning bids might have positive, others negative reduced cost using these prices. However, it is questionable whether this is problematic because it gives new information to the winning agents since a winning agent with a negative reduced cost knows that there must exist complementary agents with positive reduced cost that sum up to the same amount.

The exposition of our work is as follows: In section 2 we introduce the winner determination problem. A review of related work is given in section 3. In section 4 we present some linear programming models the solution of which is "almost always" integral. Section 5 contains an illustrative instance. In section 6 we provide a computational proof of the "almost always" conjecture. In section 7 we briefly discuss our pricing scheme in an iterative setting. Section 8 concludes the paper.

2 Winner determination

One of the most important challanges with combinatorial auctions is solving the winner determination problem, a topic which has received most attention from researchers.

Let us assume for simplicity that only one unit of each item is available. Two general models for winner determination in combinatorial auctions are known from the literature. The first one (see Wurmann and Wellman [23]) assumes that every bidder submits a bid on every subset of items. Furthermore, the number of bundles a winner may win is limited to at most one. The second model (see, e.g., DeMartini, Kwasnica, Ledyard and Porter [6] and Kwasnica, Ledyard, Porter and DeMartini [14]) includes the first as a special case; it allows multiple bundles per winner. For notational simplicity let $I = \{1, \ldots, m\}$ denote the set of items, and let $J = \{1, \ldots, n\}$ be the set of (bundles) bids. Then the second model reads as shown in the set of equations (1).

$$z = \max \sum_{j \in I} b_j x_j \tag{1a}$$

s.t.
$$\sum_{j \in J} a_{ij} x_j \le 1 \qquad \forall i \in I$$
 (1b)

$$x_j \in \{0, 1\} \qquad \forall j \in J \tag{1c}$$

Here b_j is the bid price for bundle j. If several bidders bid for the same bundle b_j is the maximum bid for that bundle, taken over all bidders. x_j indicates whether bid j is accepted $(x_j = 1)$ or not $(x_j = 0)$. The parameter a_{ij} is equal to 1, if item i is contained in bid j or not $(a_{ij} = 0)$.

Model (1) is the most widely studied single-unit (each item is unique and there is only one unit for sale of each item), single-sided (one seller and multiple buyers) case and we will study it subsequently, too. It is the set packing problem, a well-known NP-complete optimization problem (Garey and Johnson [11]). Exact and heuristic algorithms for solving the set packing problem have been developed by, e.g., Borndörfer [2], Delorme, Gandibleux and Rodriguez [5], Günlük, Ladányi, de Vries [12] and Sandholm, Suri, Gilpin, Levine [22]). A recent survey of combinatorial auctions is provided by de Vries and Vohra [7]. Combinatorial auctions can be useful in many environments and have been considered for problems including selling spectrum rights (McMillan [16], Milgrom [17]), airport takeoff & landing time slot allocation (Rassenti, Smith and Bulfin [20]), railroad segments (Brewer [3]), and delivery routes (Caplice and Sheffi [4]). Other applications are surveyed in, for instance, Kwon, Anandalingam and Ungar [15].

3 Related work

Over time several suggestions have been made to address the problem of finding interpretable dual prices for the set packing problem (1). Most of this work uses linear programming duality theory.

If we relax the integrality constraint (1c) on x_j we obtain the linear programming relaxation (2) which will be used in section 6 for benchmarking purposes, too.

$$\bar{z} = \max \left\{ \begin{array}{l} \sum\limits_{j \in J} b_j x_j \\ \text{s.t.} \quad \sum\limits_{j \in J} a_{ij} x_j \le 1 \\ x_j \ge 0 \end{array} \right\}$$

$$(2)$$

The corresponding dual is

$$\bar{z} = \min \left\{ \begin{array}{ccc} \sum_{i \in I} u_i & & \\ \text{s.t.} & \sum_{i \in I} a_{ij} u_i \ge b_j & \forall j \in J \\ u_i \ge 0 & \forall i \in I \end{array} \right\}$$
(3)

where $\mathbf{u} = (u_i)$ is the vector uf dual variables.

For the linear programming relaxation (2) we know that an optimal primal solution $\bar{\mathbf{x}}^* = (\bar{x}_j^*)$ and the corresponding optimal dual solution $\mathbf{u}^* = (u_i^*)$ have some fundamental properties.

Apparently, an optimal primal solution is primal feasible and an optimal dual solution is dual feasible. If an optimal primal solution and the corresponding optimal dual solution satisfy the constraints $u_i^*\left(\sum_{j\in J}a_{ij}\bar{x}_j^*-1\right)=0$ $\forall i\in I$, then primal complementary slackness is assured. If an optimal primal solution and the corresponding optimal dual solution satisfy the constraints $\bar{x}_j^*\left(\sum_{i\in I}a_{ij}u_i^*-b_j\right)=0$ $\forall j\in J$, then dual complementary slackness is assured.

One stream of research frequently used in the combinatorial auction setting is to impute pseudo-dual prices, that is, prices that are in some sense close to the prices obtained for a pure linear program. The way these pseudo-dual prices are constructed is based on the following ideas: (i) The winning bundles should have reduced cost equal to zero. A standard requirement for a linear program based on linear programming duality theory is that a basic variables reduced cost should be equal to zero. (ii) For the non-winning bids

the item prices should ideally have the property that all non-winning bids are priced out, i.e. the reduced costs for these bids should be non-negative. However, in the general case when the linear programming relaxation does not yield an integral solution this is unachievable. The approximation made in these cases in order to obtain an approximate linear price function is to require that as many as possible of the non-winning bids are priced out or, alternatively, that the maximum deviation for a linear price to price out the non-winning bids is minimal. (iii) As in linear programming it is often required that prices for constraints that have slack in the optimal solution yield an item price of zero. All these requirements can be interpreted as requiring primal feasibility, primal complementary slackness, dual feasibility, and dual complementary slackness.

In a combinatorial auction the auctioneer is trying to get a good and hopefully optimal solution to the winner determination problem. Assume that we know an optimal integer solution to the winner determination problem (1) and that the linear programming relaxation (2) does not have the integrality property. Then we do not have a linear price system that can be interpreted as an equilibrium market clearing mechanism.

The underlying assumptions made when constructing a set of approximate pseudo-dual prices are: (a) The optimal solution (x_j^*) of (1) is primal feasible. (b) At least one of the properties dual feasibility, primal complementary slackness or dual complementary slackness must be relaxed.

The 'normal' approach taken in the procedures that have been developed to construct pseudo-dual prices is that: (i) Dual complementary slackness should be required. This means that we make sure that the winning bids for the different bundles of items all have reduced cost equal to zero. (ii) Primal complementary slackness should be required. This means that the price for an unsold item should be equal to zero.

Hence the 'normal' relaxation used is to relax the requirement of dual feasibility leading to the fact that some of the losing bids will have a negative reduced cost when faced with the pseudo-dual price system making the agents that have submitted these bids suspicious and wondering why their bid has not been successful. This is the approach taken by Rassenti, Smith and Bulfin [20] and DeMartini, Kwasnica, Ledyard, Porter [6], among others.

In the following we sketch the approach by DeMartini, Kwasnica, Ledyard, Porter [6] and Kwasnica, Ledyard, Porter, DeMartini [14] which is one of the most recent models stipulating the use of pseudo-dual prices; other approaches can be found in, e.g., Parkes [18], Bikhchandani and Ostroy [1] and Xia, Koehler and Whinston [24].

Let $J_0 := \{j \in J : x_j^* = 0\}$ and $J_1 := \{j \in J : x_j^* = 1\}$ denote the set of losing and winning bids, respectively. Apparently, we have $J_0 \cap J_1 = \emptyset$ and $J_0 \cup J_1 = J$. The main component is to solve the linear program (4).

s.t.
$$\sum_{j \in J} a_{ij} u_i + y_j \ge b_j \qquad \forall j \in J_0$$
(4b)

$$\sum_{j \in J} a_{ij} u_i = b_j \qquad \forall j \in J_1 \tag{4c}$$

(4a)

$$w \ge y_j \qquad \qquad \forall j \in J_0 \tag{4d}$$

$$u_i \ge 0 \qquad \qquad \forall i \in I$$
 (4e)

$$y_j \ge 0 \qquad \qquad \forall j \in J_0 \tag{4f}$$

At the prices (u_i) there may be some losing bids for which $\sum_{j\in J} a_{ij}u_i \leq b_j$, falsely signaling a possible winner, which is by virtue the nature of package bidding. Of course, such bids can be resubmitted if $(b_j - \sum_{j\in J} a_{ij}u_i)$ is 'large enough'. The objective (4a) has been designed to minimize the largest violation. If "ideal" prices exist, they will be the solution with $y_j = 0$ for all $j \in J_0$ and, hence w will be equal to zero. If the prices from (4a) are not unique a sequence of iterations each of which requires to solve the linear program (4) is performed (for details see DeMartini, Kwasnica, Ledyard, Porter [6] and Kwasnica, Ledyard, Porter, DeMartini [14]).

In the following section we will present some linear programs which do not require one of the relaxations described above in order to achieve interpretable prices.

4 Variable elimination/aggregation

The formulation of the family of linear programming models is accomplished in two steps. First, we solve the integer program (1) and, hence, we know the sets J_0 and J_1 . Then we eliminate all variables corresponding to the chosen items and introduce instead an artificional one by means of column aggregation.

More precisely, in step two we eliminate the bids (variables) contained in J_1 and introduce an aggregated bid instead. This bid contributes the amount

$$\bar{b} = \sum_{j \in J_1} b_j$$

to the optimal objective function value and it contains item i if

$$\bar{a}_i = \sum_{j \in J_1} a_{ij}$$

equals 1; if $\bar{a}_i = 0$, item *i* is not included. For the sake of notational brevity

$$I_{<} = \left\{ i \in I : \sum_{j \in J_1} a_{ij} < 1 \right\}$$

denotes the set of items not contained in the aggregated bid.

A linear program following the idea of variable elimination/aggregation, that is, which solely uses the losing bids J_0 and the aggregated bid, is provided in (5).

$$\max \sum_{j \in J_0} b_j x_j + \bar{b} \cdot z \tag{5a}$$

s.t.
$$\sum_{j \in J_0} a_{ij} x_j + \bar{a}_i z \le 1 \qquad \forall i \in I$$
(5b)

$$x_j \ge 0 \qquad \qquad \forall j \in J_0 \tag{5c}$$

$$z \ge 0 \tag{5d}$$

The objective function (5a) counts the contribution of the variables contained in J_0 and of the variable z associated with the aggregated bid. Constraint (5b) assures that item $i \in I$ is at most once contained in a winning bundle. Finally, (5c) and (5d) define the decision variables to be nonnegative.

Apparently, in terms of the optimal solution (x_j^*) in general we have positive slack in some of the constraints (5b). If we consider the constraints associated with the items $i \in I_{\leq}$ separately, we get the linear program (6).

$$\max \sum_{j \in J_0} b_j x_j + \bar{b} \cdot z \tag{6a}$$

s.t.
$$\sum_{i \in J_0} a_{ij} x_j + \bar{a}_i z \le 1 \qquad \forall i \in I \setminus I_{<}$$
(6b)

$$\sum_{j \in J_0} a_{ij} x_j = 0 \qquad \forall i \in I_{<} \tag{6c}$$

$$x_j \ge 0 \qquad \qquad \forall j \in J_0 \tag{6d}$$

$$\geq 0$$
 (6e)

Finally, if we replace the inequality (6b) through an equality we get the linear program (7).

 \boldsymbol{z}

$$\max \sum_{j \in J_0} b_j x_j + \bar{b} \cdot z \tag{7a}$$

s.t.
$$\sum_{j \in J_0} a_{ij} x_j + \tilde{a}_i z = 1 \qquad \forall i \in I \setminus I_{<}$$
(7b)

$$\sum_{j \in J_0} a_{ij} x_j = 0 \qquad \forall i \in I_{<}$$
(7c)

$$\begin{array}{ll} x_j \ge 0 & \forall j \in J_0 \\ z \ge 0 & (7e) \end{array}$$

Note that while in model (6) only the dual variables associated with equation (6c) are unrestricted in sign this is the case in model (7) for all the dual variables.

The linear program (5) has first been proposed by Drexl and Jørnsten [8]. There it has been shown by means of a small instance that aggregation might pay-off in order to get linear prices.

In section 6 we will show be means of a computational study how the models (5) to (7) behave in terms of integrality of optimal solutions and dual degeneracy. Before we illustrate the idea using an example.

5 Illustrative instance

An example with 8 items and 20 bids taken from the combinatorial auction test suite (CATS; see [13]) illustrates the models presented above. The bid prices (b_j) and the coefficient matrix (a_{ij}) of this instance are provided in Table 1.

solution of integer program (1)
$x_6^* = x_8^* = x_{19}^* = 1$ OFV = 6,469
solution of linear program (2)
$\bar{x}_2^* = \bar{x}_5^* = \bar{x}_7^* = \bar{x}_{11}^* = \bar{x}_{19}^* = 0.33, \bar{x}_4^* = \bar{x}_6^* = 0.67$ OFV = 7,203

Table	3:	Results ·	- Part	1
-------	----	-----------	--------	---

solution of linear program (5)
$x_2^{(5)} = x_4^{(5)} = 0.4, x_5^{(5)} = x_7^{(5)} = x_{10}^{(5)} = x_{11}^{(5)} = x_{17}^{(5)} = 2^{(5)} = 0.2$
OFV = 7,093.6
solution of linear programs (6) and (7)
$z^{(6)} = 1, z^{(7)} = 1$
OFV = 6,469

Table 4: Results – Part 2

Table 3 detailles the results of the solution of the integer program (1) and of the linear programming relaxation (2) of this instance. Variables not given explicitly have value 0. OFV abbreviates optimal objective function value.

If we apply the variable elimination/column aggregation technique described above we get the modified instance shown in Table 2, where column j = 21 corresponds to the aggregated bid.

Table 4 provides the results of the solution of the aggregate models (5), (6) and (7) for the instance; a superscript refers to each of the models. As can be seen model (5) does not have an integral solution which is the case both for model (6) and for model (7).

One set of dual prices associated with the integer solution of model (6) is

 $(u_i^{(6)}) = (2,014; 420; 693; 81; 2,838; 0; 2,560; 701)$

while one set of prices of model (7) is

 $(u_i^{(7)}) = (0; 2,446; 693; 2,806; 3,707; -177; 0; 701).$

The first price vector contains one dual variable with value 0 while the second contains two. Contrary to this particular observation the computational results presented below will show that dual degeneracy of model (7) in general is a bit less than that of model (6). Of course, the dual solution provided depends on the particular solver used (in our case Cplex).

	j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	b_j	693	4,924	81	420	2,791	761	2,806	858	2,763	4,040	3,876	5,852	3,14	7 2,27	9 5,15	8 1,598	3 4,798	701	4,850	2,788
<u>- 14</u>	a_{1j}					1		1	1			1	1	1	1	1		1			1
÷	a_{2j}				1					1	1		1	1	1		1	1		1	
ž	a_{3j}	1	1							1		1	1			1		1		1	1
tu n	a_{4j}			1				1			1	1	1			1	1	1		1	1
5	a_{5j}		1			1				1	1	1	1			1					
	a_{6j}		1				1			1	1		1			1		1			1
	a_{7j}		1					1					1	1		1	1	1		1	
	a_{8j}		1			1				1	1		1	1	1	1	1		1	1	
		-																			
	j	1	2		3	4	5	7	9	10	11	. 1	2 :	13	14	15	16	17	18	20	21
Tahle	b_j	693	3 4,92	4 {	31 4	20 2,	791	2,806	2,763	3 4,04	0 3,87	76 5,8	352 3,	147 2	2,279	5,158	1,598	4,798	701	2,788	6,469
ې -	a_{1j}						1	1		<u></u>				1	1	1	1	1		1	1
≦ 2	a_{2j}					1			1	1		-	1	1	1		1	1			1
	a_{3j}	1	1						1		1		1			1		1		1	1
₫	a_{4j}				1			1		1	1		1			1	1	1		1	1
Ista	a_{5j}		1				1		1	1	1	•	1			1					
nce	a_{6j}		1							1	1			1			1	1		1	1
-	a_{7j}		1					1					1	1		1	1	1			1
	a_{8j}		1				1		1	1			1	1	1	1	1		1		1

2 ē

ω

6 Computational results

The purpose of the computational experiments is to show that the solution of the aggregate linear programs is "almost always" integral. The numerical results presented have been obtained for a set of randomly generated instances. We decided to use the well motivated and universally accepted combinatorial auction generator CATS (combinatorial auction test suite; see [13]) which provides a set of distributions for modeling realistic bidding behavior. In particular, we have generated instances using the built-in distributions arbitrary, paths, regions, scheduling, and matching respectively.

The models described earlier have been imlemented using the Cplex callable library (version 10.0.1; all parameters with default values) on an AMD Athlon(tm) XP 3000+ with 512 MB RAM and 2.1 Ghz clockpulse running under the operating system Linux.

The results of the numerical experiments for the distribution types arbitrary, paths, regions, scheduling and matching are given in tables 5 to 9, respectively. Each table is structured as follows: Column 1 displays the number of items m.¹ In column 2 the number of bids $n = m \cdot \rho$ with $\rho \in \{2, 3, 4, 5\}$ is given. For each combination of m and n we have generated 100 instances.² The next eight columns provide for each of the four models (including the linear programming relaxation (2) of the set packing problem) first how often we observe on average the case that the solution of the aggregate linear program is integral (displayed in column LP=IP). Second, the column #DV0(%) displays the average number of dual variables equal to "0" in percent for the cases where LP=IP in order to have an indication of the average number of cases where we had LP=IP and the average dual degeneracy.

The results presented in tables 5 to 9 indicate the following:

- For the distribution types arbitrary and regions the linear programming relaxation (2) is integral rarely, if so in particular for smaller instances. For the distribution types paths, scheduling and matching solving the ordinary linear programming relaxation is a viable approach in many cases.
- For the distribution type arbitrary the aggregate model (5) is only slightly better than model (2) in terms of having LP=IP. The same holds true for the distribution type scheduling (where model (2) did already a pretty good job). For the distribution type regions the aggregate model (5) produces on average for every second instance an integral solution. For the other two distribution types the aggregate model (5) produces an integral solution in most cases.
- If we consider models (6) and (7) instead of model (5) the whole picture changes drastically, that is, both models produce almost always integer solutions. Model (6) is a bit weaker in this regard only for a few particular instances of the distribution types arbitrary and scheduling.

¹For the distribution type matching due to some reasons for $m \in \{30, 50, 70, 90\}$ no instance can be generated at all.

²Hence, overall we have $4 \cdot 4 \cdot 10 \cdot 100 + 4 \cdot 5 \cdot 100 = 18,000$ instances.

		model	(2)	model (5)		model (6)	model (7)		
m	n	LP=IP	#DV0(%)	LP=IP	#DV0(%)	LP=IP	#DV0(%)	LP=IP	#DV0(%)	
10	20	60	59.4	65	62.6	100	66.0	100	60.6	
	30	44	56.4	54	59.9	100	60.5	100	52.9	
	40	47	55.5	59	58.2	100	55.2	100	49.9	
	50	30	59.4	47	61.3	99	54.7	100	47.2	
20	40	20	58.7	24	60.2	100	71.9	100	68.5	
	60	1	51.7	3	49.4	100	66.2	100	62.2	
	80	5	55.6	12	58.7	100	62.4	100	56.8	
	100	3	56.0	4	56.7	100	60.3	100	53.9	
30	60	0	-	1	60.0	100	72.4	100	71.1	
	90	0	-	3	55.2	100	69.1	100	64.9	
	120	0	-	0	-	100	64.4	100	59.2	
	150	0	-	0	-	100	61.8	100	55.9	
40	80	0	-	0		100	72.5	100	70.9	
	120	0		0		100	69.8	100	66.9	
	160	0	-	0		100	66.8	100	62.4	
	200	0	-	0	-	100	65.3	100	59.3	
50	100	0	-	0	-	100	73.1	100	71.3	
	150	0		0	-	100	69.9	100	66.6	
	200	0	-	0	~	100	67.7	100	62.6	
	250	0	-	0	-	100	63.8	100	58.3	
60	120	0	-	0	-	100	72.8	100	71.2	
	180	0		0	-	100	69.8	100	66.6	
	240	0	_	0		100	68.4	100	63.7	
	300	0		0	~	100	65.2	100	59.2	
70	140	0		0	-	100	73.0	100	71.6	
	210	0	_	0		100	70.7	100	68.5	
	280	0	-	0	~	100	68.0	100	63.4	
	350	0	_	0		100	64.5	100	58.7	
80	160	0	_	0	-	100	73.5	100	72.0	
	240	0	_	0	-	100	70.2	100	68.1	
	320	0		0	-	100	68.5	100	63.5	
	400	0	-	0	-	100	64.1	100	58.5	
90	180	0	-	0	-	100	73.6	100	72.3	
	270	0	-	0	-	100	70.4	100	68.1	
	360	0	-	0	-	100	67.9	100	63.7	
	450	0	-	0	-	100	64.1	100	58.4	
100	200	0	_	0	-	100	73.5	100	71.9	
	300	0	-	0	-	100	70.2	100	68.2	
	400	0		0	. –	100	67.4	100	63.1	
	500	0		0	_	100	63.9	100	57.8	
ave	rage	5.2	56.6	6.8	58.2	99.9	67.3	100.0	63.2	

Table 5: Distribution type arbitrary

		model	(2)	model	(5)	model (6)	model ((7)
m	n	LP=IP	#DV0(%)	LP=IP	#DV0(%)	LP=IP	#DV0(%)	LP=IP	#DV0(%)
10	20	96	63.4	99	76.7	100	75.2	100	69.7
	30	97	67.1	100	76.2	100	73.2	100	70.0
	40	100	73.8	100	80.2	100	74.9	100	72.3
	50	99	77.9	100	81.8	100	74.4	100	72.4
20	40	05	50.7	100	69.0	100	64.6	100	62.6
20	40 60	95 04	46 1	100	66.0	100	60.3	100	59.3
	80	94	51 0	100	65.2	100	56.2	100	58.3
	100	99	56.9	100	66.7	100	53.5	100	57.5
20	60	90	55.0	100	70 1	100	66.8	100	66.1
30	00	09 00	55.0	100	60.8	100	62.5	100	65.5
	100	00	50.0	100	60.1	100	59.2	100	64.3
	120	95	52.0	100	68.0	100	56.1	100	62.4
	150	95	57.2	100	00.9	100	60.A	100	69.0
40	80	84	57.9	100	74.5	100	08.4	100	00.9 60.2
	120	72	55.1	100	72.7	100	05.3	100	09.3 60.6
	160	81	57.8	100	72.0	100	02.4	100	69.0
	200	85	61.7	100	73.0	100	00.1	100	00.4
50	100	68	60.2	100	76.4	100	70.5	100	71.8
	150	72	60.3	100	75.9	100	67.9	100	72.3
	200	69	60.2	100	75.1	100	65.4	100	73.4
	250	72	64.3	100	76.0	100	63.9	100	72.3
60	120	70	62.6	100	77.6	100	71.7	100	73.0
	180	61	62.6	100	77.6	100	70.0	100	75.4
	240	60	63.6	100	77.2	100	67.1	100	74.4
	300	69	66.6	100	77.8	100	66.9	100	75.2
70	140	60	64.8	100	79.8	100	72.5	100	74.9
10	210	41	63.9	100	78.3	100	70.8	100	76.1
	280	44	65.8	100	78.4	100	69.2	100	77.2
	350	54	67.4	100	79.0	100	68.2	100	76.6
<u>0</u> 0	160	11	66 7	100	79.8	100	73.6	100	75.7
00	240	52	66 6	100	79.8	100	72.1	100	77.9
	240	J2 13	68.4	100	80.2	100	70.7	100	79.2
	400	37	69.9	100	79.8	100	69.7	100	78.1
00	100	40	67 7	100	80.0	100	73.8	100	76.9
90	180	42	07.7	100	80.3	100	72.5	100	78.5
	270	20	07.5 60.0	100	80.6	100	71 5	100	79.0
	700 200	20 26	09.2 70.7	100	81.2	100	70.8	100	80.2
	450	30	10.1	100	01.2	100	75.0	100	77 A
100	200	41	68.6	100	81.0	100	15.2	100	70.6
	300	29	68.4	100	80.8	100	13.3	100	70 7
	400	22	69.3	100	0U.0	100	12.1 71 A	100	79.0
	500	30	(1.4	100	01.4 76.0	100	60 1	100 0	72 3
ave	rage	66.0	63.0	99.9	10.2	100.0	00.1	100.0	

Table 6: Distribution type paths

		model	(2)	model	(5)	model (6)	model ((7)
m	n	LP=IP	#DV0(%)	LP=IP	#DV0(%)	LP=IP	#DV0(%)	LP=IP	#DV0(%)
10	20	79	68.1	87	73.8	100	67.8	100	59.2
-•	30	73	67.8	89	73.4	100	64.4	100	54.8
	40	71	66.7	86	73.3	100	62.0	100	51.5
	50	79	68.5	92	74.6	100	59.4	100	52.2
20	40	67	73.4	85	77.9	100	72.2	100	67.8
20	60	64	69.4	93	75.4	100	71.2	100	64.9
	80	53	67.6	93	75.7	100	67.8	100	60.4
	100	47	67.1	94	75.5	100	63.6	100	55.3
30	60	42	67.0	72	72.6	100	74.5	100	69.7
	90	27	65.2	82	70.5	100	69.2	100	64.0
	120	27	64.6	86	71.2	100	68.9	100	60.0
	150	17	64.2	91	72.1	100	66.5	100	56.3
40	80	22	61.5	54	67.8	100	75.0	100	71.0
	120	14	61.8	53	66.4	100	69.6	100	63.6
	160	5	59.8	60	67.2	100	69.1	100	62.0
	200	9	60.4	69	68.0	100	64.1	100	55.9
50	100	8	60.4	19	65.1	100	74.5	100	70.8
	150	2	67.5	17	60.7	100	70.9	100	65.0
	200	1	59.3	13	61.9	100	67.7	100	60.5
	250	3	55.9	22	63.0	100	63.3	100	56.2
60	120	6	60.9	63	69.3	100	75.1	100	72.4
	180	1	62.6	79	66.9	100	71.1	100	67.6
	240	1	61.3	83	67.0	100	68.4	100	61.4
	300	1	65.2	91	68.1	100	63.2	100	56.1
70	140	4	56.2	22	63.2	100	74.8	100	71.5
	210	0		26	63.6	100	70.1	100	65.3
	280	0	-	36	63.2	100	66.7	100	61.2
	350	0	-	52	63.3	100	63.7	100	57.1
80	160	3	60.4	58	66.8	100	75.6	100	72.6
	240	0		77	65.7	100	70.7	100	68.4
	320	0	-	89	67.0	100	66.4	100	63.4
	400	0	-	93	67.9	100	63.2	100	59.1
90	180	1	56.8	15	62.4	100	74.8	100	71.4
	270	0	-	25	62.5	100	70.2	100	66.0
	360	0	- .	34	61.7	100	66.6	100	61.8
	450	0	-	51	64.1	100	62.6	100	57.0
100	200	0	-	1	58.3	100	74.8	100	71.6
	300	0	~	3	57.9	100	70.8	100	66.U
	400	0	~	2	62.0	100	65.6	100	59.5
	500	0		1	60.4	100	62.5	100	55.2
ave	erage	18.1	63.7	56.4	67.2	100.0	68.5	100.0	62.7

Table 7: Distribution type regions

•

		model	(2)	model	(5)	model (6)	model ((7)
m	n	LP=IP	#DV0(%)	LP=IP	#DV0(%)	LP=IP	#DV0(%)	LP=IP	#DV0(%)
10	20	83	65.3	86	66.9	100	70.4	100	59.6
	30	91	59.0	94	60.3	100	66.3	100	56.7
	40	88	57.0	91	58.5	100	62.5	100	52.4
	50	84	63.8	92	63.9	100	60.1	100	50.0
20	40	84	71.2	88	75.3	100	75.1	100	54.5
20	60	66	57.1	76	60.3	100	68.8	100	55.1
	80	49	43.5	66	46.6	100	65.1	100	55.6
	100	51	40.0	65	45.4	99	56.9	100	50.4
30	60	88	85.9	89	86.3	100	74.5	100	57.4
	90	74	70.0	83	69.3	99	68.8	100	52.2
	120	51	53.1	67	54.1	99	58.7	99	53.0
	150	45	34.8	66	37.6	99	47.7	100	40.2
40	80	97	91.9	97	91.9	100	73.4	100	52.6
	120	85	86.6	87	87.0	99	70.4	100	50.9
	160	72	77.2	78	75.8	100	62.9	100	48.3
	200	53	65.1	67	64.4	95	53.6	97	36.6
50	100	100	93.3	100	93.3	100	73.9	100	54.0
	150	91	90.1	93	89.6	100	69.6	100	51.2
	200	87	81.5	93	81.9	100	66.9	100	49.4
	250	77	80.7	84	80.9	99	61.0	100	46.6
60	120	98	94.6	98	94.6	100	71.0	100	48.6
	180	90	92.1	92	92.1	100	73.7	100	48.8
	240	86	89.5	89	89.2	98	68.7	99	49.3
	300	84	86.4	90	86.8	100	67.1	100	42.3
70	140	99	95.2	99	95.4	100	72.9	100	55.9
	210	97	93.6	98	93.6	100	68.3	100	47.7
	280	93	91.6	97	91.3	100	70.4	100	42.9
	350	91	89.3	95	89.6	100	68.6	100	45.2
80	160	100	95.8	100	95.9	100	70.1	100	53.4
	240	100	94.4	100	94.4	100	69.1	100	47.1
	320	98	92.0	99	92.3	100	69.4	100	47.6
	400	91	90.9	93	91.2	99	65.3	100	42.7
90	180	99	96.3	99	96.3	100	71.2	100	44.7
	270	99	94.8	99	94.9	99	68.5	100	48.1
	360	97	93.3	99	93.3	100	71.5	100	49.7
	450	97	92.1	98	92.2	99	67.0	100	41.9
100	200	100	96.7	100	96.7	100	72.2	100	48.0
	300	100	95.5	100	95.5	100	67.2	100	46.0
	400	98	94.3	98	94.4	100	70.0	100	46.3
	500	96	93.1	97	93.1	100	69.8	100	42.3
ave	rage	85.7	80.7	90.0	81.3	99.6	67.5	99.8	49.1

Table 8: Distribution type scheduling

		model	(2)	model (5)	model (6)	model (7)		
m	n	LP=IP	#DV0(%)	LP=IP	#DV0(%)	LP=IP	#DV0(%)	LP=IP	#DV0(%)	
	40	88	69.7	100	80.8	100	73.0	100	65.9	
	60	94	73.8	100	83.1	100	71.0	100	65.3	
	80	89	77.2	100	85.0	100	68.7	100	65.7	
	100	90	79.6	99	86.5	100	68.0	100	68.3	
40	80	99	69.6	100	80.1	100	72.6	100	65.1	
	120	94	73.0	100	81.8	100	69.8	100	66.3	
	160	97	75.8	100	84.1	100	70.1	100	67.7	
	200	99	77.7	100	85.9	100	67.6	100	67.7	
60	120	100	68.8	100	77.1	100	63.3	100	57.9	
	180	92	58.3	100	71.1	100	63.0	100	55.2	
	240	91	53.8	100	69.9	100	62.3	100	53.5	
	300	81	54.0	100	69.3	100	61.5	100	52.8	
80	160	98	70.7	100	77.2	100	60.5	100	56.8	
	240	93	54.6	100	67.4	100	60.5	100	54.1	
	320	73	47.5	100	62.2	100	59.2	100	52.7	
	400	69	45.5	100	61.7	100	58.4	100	51.7	
100	200	97	70.3	100	77.7	100	60.4	100	57.2	
	300	89	56.4	100	69.4	100	60.1	100	53.9	
	400	74	47.3	100	63.1	100	58.8	100	52.5	
	500	58	44.3	100	59.9	100	56.8	100	52.2	
ave	erage	88.2	63.4	99.9	74.7	100.0	64.3	100.0	59.1	

Table 9: Distribution type matching

- Comparing models (6) and (7) in terms of dual degeneracy the first interesting observation is that dual degeneracy of model (7) is on average for all distribution types but paths smaller than that produced by model (6). At first glance this is surprising because dual solutions being feasible and optimal for (6) are feasible and optimal for (7) as well. But due to the fact that for model (7) all variables can take also negative values there is much greater variety in fixing dual variables so as to comply with the primal solution and, hence, an entry of "0" appears less often.
- On average integral solutions produced for the distribution type scheduling by means of model (7) show by far lower dual degeneracy than that of the intergral solutions produced by the other models, also for the other distribution types except paths. On the other hand dual degeneracy is rather high for the distribution type path, irrespective of the model which produced an integral solution.

Summarizing, our linear programming models yield optimal integer solutions for all but five instances of the distribution type scheduling and, hence, we have computationally proven the "almost always" conjecture.

7 Discussion

Iterative combinatorial auctions are predominant in e-business. An iterative format allows bidders to learn about their rivals' valuations through the bidding process, allowing them to adjust their own valuations. On the downside, iterative procedures open up space for strategic and collusive behaviour. Furthermore, all of the pathologies from single-round formats get magnified in an iterative setting. Moreover, iterative formats have difficult design issues.

In the following we address some particular multi-round design issues arising of our linear pricing scheme: The aggregate linear pricing scheme prices out all losing bids almost always. Hence, for the agents with losing bids the prices can be used in the standard way to determine how much they have to increase their bid in order to be competitive. For the agents whose bids have been selected, i.e. the winning bids, the situation is a bit more complicated, since some of the winning bids might have positive, others negative reduced cost using these prices. However, it is questionable whether this is a problematic because it gives new information to the winning agents since a winning agent with a negative reduced cost that sum up to the same amount. It is assumed that winning agents pay as bid. In general this is not a good mechanism in iterative auctions. However, with the information produced by the aggregate prices agents know that their win depends on complementary bids. This might reduce the potential for strategic bidding. To investigate the incentive effects of using aggregate linear prices in an iterative auction is a research issue in its own right.

Other design issues are discussed in Pekeč and Rothkopf [19]; to mention a few: Dealing with the exposure problem and the expressiveness of the bidding language; the threshold problem and procedures for keeping the bidding moving; avoiding and resolving ties.

8 Summary

In this paper we have presented a family of linear programming models for the set packing problem the solution of which is "almost always" integral. Hence, in general dual prices for the set packing problem now are readily available. For the few more difficult instances we are going to develop non-linear pricing algorithms; see Drexl, Jørnsten and Knof [9].

Recently, Dunford, Hoffmann, Menon, Sultana and Wilson [10] have shown that pseudodual, linear pricing algorithms produce non-monotonity between rounds, something that is expected to disturb bidders in iterative combinatorial auctions. The fact that now interpretable prices are available in most cases could help to avoid such effects.

References

[1] BIKHCHANDANI, S., OSTROY, J. (2002), The package assignment model, Journal of Economic Theory, Vol. 107, pp. 377–406

- BORNDÖRFER, R. (1998), Aspects of set packing, partitioning, and covering, PhD Dissertation, Technical University of Berlin
- [3] BREWER, P.J. (1999), Decentralized computation procurement and computational robustness in a smart market, Economic Theory, Vol. 13, pp. 41–92
- [4] CAPLICE, C.G., SHEFFI, Y. (2003), Optimization-based procurement for transportation services, Journal of Business Logistics, Vol. 24, No. 2, pp. 109–128
- [5] DELORME, X., GANDIBLEUX, X., RODRIGUEZ, J. (2004), GRASP for set packing problems, European Journal of Operational Research, Vol. 153, pp. 564–580
- [6] DEMARTINI, C., KWASNICA, A.M., LEDYARD, J.O., PORTER, D. (1999), A new and improved design for multi-object iterative auctions. Working Paper, California Institute of Technology
- [7] DE VRIES, S., VOHRA, R. (2003), Combinatorial auctions: a survey, INFORMS Journal on Computing, Vol. 15, pp. 284–309
- [8] DREXL, A., JØRNSTEN, K. (2005), Reflections about pseudo-dual prices in combinatorial auctions, Journal of the Operational Research Society (to appear)
- [9] DREXL, A., JØRNSTEN, K., KNOF, D. (2007), Non-linear anonymous pricing combinatorial auctions, Working Paper (in preparation)
- [10] DUNFORD, M., HOFFMANN, K., MENON, D., SULTANA, R., WILSON, T. (2003), Testing linear pricing algorithms for use in ascending combinatorial auctions, Technical Report, George Mason University, Systems Engineering & Operations Research Department (revised)
- [11] GAREY, M.R., JOHNSON, D.S. (1979), Computers and intractability: a guide to the theory of NP-completeness, Freeman, San Francisco, CA
- [12] GÜNLÜK, O., LADÁNYI, L., DE VRIES, S. (2005), A branch-and-price algorithm and new test problems for spectrum auctions, Management Science, Vol. 51, pp. 391-404
- [13] http://robotics.stanford.edu/CATS
- [14] KWASNICA, A.M., LEDYARD, J.O., PORTER, D., DEMARTINI, C. (2005), A new and improved design for multi-object iterative auctions, Management Science, Vol. 51, pp. 419–434
- [15] KWON, R.H., ANANDALINGAM, G., UNGAR, L.H. (2005), Iterative combinatorial auctions with bidder-determined combinations, Management Science, Vol. 51, pp. 407–418
- [16] MCMILLAN, J. (1994), Selling spectrum rights, Journal of Economic Perspectives, Vol. 8, pp. 145–162
- [17] MILGROM, P. (2000), Putting auction theory to work: the simultaneous ascending auction, Journal of Political Economy, Vol. 108, pp. 245–272

- [18] PARKES, D.C. (2001), Iterative combinatorial auctions: achieving economic and computational efficiency, PhD Dissertation, Department of Computer and Information Science, University of Pennsylvania
- [19] PEKEČ, A., ROTHKOPF, M.H. (2003), Combinatorial auction design, Management Science, Vol. 49, pp. 1485–1503
- [20] RASSENTI, S.J., SMITH, V.J., BULFIN, R.L. (1982), A combinatorial auction mechanism for airport time slot allocation, Bell Journal of Economics, Vol. 13, pp. 402-417
- [21] ROTHKOPF, M.H., PEKEČ, A., HARSTAD, R.M. (1998), Computationally manageable combinational auctions, Management Science, Vol. 44, pp. 1131–1147
- [22] SANDHOLM, T., SURI, S., GILPIN, A., LEVINE, D. (2005), CABOB: a fast optimal algorithm for winner determination in combinatorial auctions, Management Science, Vol. 51, pp. 374–390
- [23] WURMANN, P.R., WELLMAN, M.P. (1999), Equilibrium prices in bundle auctions. Working Paper, Artificial Intelligence Laboratory, The University of Michigan
- [24] XIA, M., KOEHLER, G.J., WHINSTON, A.B. (2004), Pricing combinatorial auctions, European Journal of Operational Research, Vol. 154, pp. 251–270