Non-linear anonymous pricing in combinatorial auctions

Andreas Drexl, Kurt Jørnsten, Diether Knof

Institut für Betriebswirtschaftslehre, Christian-Albrechts-Universität, Kiel, Germany, andreas.drexl@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

Working Paper, August 2005

Abstract

In combinatorial auctions the pricing problem is of main concern since it is the means by which the auctioneer signals the result of the auction to the participants. In order for the auction to be regarded as fair among the various participants the price signals should be such that a participant that has won a subset of items knows why his bid was a winning bid and that agents that have not acquired any item easily can detect why they lost. The problem in the combinatorial auction setting is that the winner determination problem is a hard integer programming problem and hence in general there does not exist a linear pricing scheme supporting the optimal allocation. This means that single item prices, that support the optimal allocation can in general not be found. In this article we present an alternative.

From integer programming duality theory we know that there exist non-linear anonymous price functions that support the optimal allocation. In this paper we will provide a means to obtain a simple form of a non-linear anonymous price system that supports the optimal allocation. Our method relies on the fact that we separate the solution of the winner determination problem and the pricing problem. This separation yields a non-linear price function of a much simpler form compared to when the two problems are solved simultaneously. The pure pricing problem is formulated as a mixed-integer program. If solving this program is too demanding computationally a heuristic can be used which essentially requires us to solve a sequence of linear programming relaxations of a new mixed-integer programming formulation of the pricing problem. The procedure is computationally tested using instances of the combinatorial auctions test suite [17].

Keywords: Combinatorial auctions, set packing, duality theory, non-linear anonymous prices

1 Introduction

Auctions have been used for a long term to generate an efficient market mechanism to trade goods and services. The most common auction format is a single unit auction in which the goods are auctioned off in some predetermined sequence using either an English or a Dutch type auction (see, e.g., Klemperer 2002, McAfee and McMillan 1987, Milgrom 1989, Wolfstetter 1999). However, in many auctions/markets a participants valuation of an object depends significantly on which other objects the participant acquires. Objects can be substitutes or complements and the valuation of a particular bundle of items may not equal the sum of the individual items in the bundle. In order to design an efficient auction in such situations a non sequential auction format is required. Auctions in which agents are allowed to bid on bundles of items and the auctioneer sells the whole set of items in one single auction are named combinatorial auctions.

Recently, the interest in the design of combinatorial auction mechanisms has been booming. The design of such mechanisms requires among others to address a couple of issues: (a) settlement of the auction rules and who is in charge of controlling them, (b) which agents are allowed to take part in the auction, (c) which bidding formats are allowed, (d) how are the winners to be determined, (e) how much are the winners to be charged, (f) how much information is provided to the participants, (g) is the auction format of a single round or an iterative, multiple round type? In this paper we will focus on issue (f), that is, on the price information given by the auctioneer to the participating agents.

The pricing problem in combinatorial auctions has two important aspects: (i) In an iterative combinatorial auction the prices presented to the agents should provide means for an agent to revise his bid properly knowing that the price information given by the auctioneer in each round contains information on the potential winner valuation in the current round and information that the agent can use in order to decide whether to rise/lower his bid on a certain bundle or withdraw from the auction. (ii) When the auction is terminated the prices provided by the auctioneer should be such that it is easy for the agents having obtained winning bids to understand why their bid won and how much they will be charged. For the losers in the auction the prices provided should be such that it is easy to detect why the bid was not high enough to obtain the particular bundle and provide means for the agents to determine that the auction was fair.

Since the winner determination problem in a combinatorial auction is an integer program we know that in general there do not exist linear prices on the single items that clear the market, i.e. support the optimal allocation of bundles to winning agents. In the literature this problem of non-existence of linear market clearing prices has been resolved in various ways. The most common solution is to generate so-called pseudo-dual prices which are in some sense a best possible approximation to linear prices that can be obtained. Other means of tackling the pricing problem in combinatorial auctions are to use non-anonymous/discriminatory prices or non-linear anonymous prices.

Subsequently we will present a way to generate a non-linear extension of linear prices for combinatorial auctions that makes it easy for the agents to analyse their current bids and to understand why they have won or why they have lost in the auction. The way that the mixed-integer pricing program is solved can be interpreted as constructing sets of restricted auctions from the original auction each of which has the integrality property. Since the algorithm is constructing linear prices such that the union of these sets of restricted auctions spans the original auction we have constructed a set of extreme linear prices and each agent should be capable of handling all prices that can be generated as a convex combination of the extreme prices.

The paper is organized as follows: In section 2 we present the mathematical programming formulation of the winner determination problem and discuss the pricing problem briefly. In section 3 we provide a linear programming formulation for the determination of pseudo-dual prices and discuss their properties. A review of related work is given in section 4. Section 5 is dedicated to the new pricing model. In section 6 we discuss the relationship between the pricing model and so-called reduced combinatorial auctions. In section 7 we present computational results based on auctions generated from the combinatorial auctions test suite [17]. A couple of research opportunities are discussed in section 8. Section 9 concludes the paper.

2 Winner determination problem

Let us assume for simplicity that only one unit of each object is available. Let $I = \{1, \ldots, m\}$ denote the set of items, and let $J = \{1, \ldots, n\}$ be the set of bids. Then the model reads as follows:

$$\left.\begin{array}{rll} z = \max & \sum\limits_{j \in J} b_j x_j & \\ \text{s.t.} & \sum\limits_{j \in J} a_{ij} x_j \leq 1 & \forall\, i \in I \\ & x_j \in \{0,1\} & \forall\, j \in J \end{array}\right\} \tag{1}$$

The parameter b_j is the bid price for bundle j from bidder i. The binary parameter a_{ij} equals 1, if item i is contained in bid j (0, otherwise). The variable x_j indicates whether bid j is accepted $(x_j = 1)$ or not $(x_j = 0)$.

The winner determination problem (1) is an integer programming problem. In general solving the linear programming relaxation of the winner determination problem will result in a solution in which some of the variables have non-integral values. In such cases where the integer programming problem has a duality gap which is strictly greater than zero, we know from theory (see, e.g., Nemhauser and Wolsey 1988) that there does not exist a linear price function that supports the optimal allocation of winning bundles.

It is obvious that if bidders submit their true values on the various bundles, the solution to the winner determination problem gives an efficient allocation of indivisible objects in an exchange economy. The formulation above is valid for the winner determination problem in the case of subadditive and superadditive bids, however, in the latter case it is of special interest. If items are substitutes a more general winner determination problem based on so-called XOR bids is needed (see, e.g., Xia et al. 2004).

Model (1) is the most widely studied single-unit (each item is unique and there is only one unit for sale each), single-sided (one seller and multiple buyers) case. It is the set packing problem, a well-known NP-complete optimization problem (Garey and Johnson 1979). Exact and heuristic algorithms for solving the set packing problem have been developed by, e.g., Borndörfer (1998), Delorme et al. (2004), Harche and Thompson (1994), Hoffmann and Padberg (1993) and Sandholm et al. (2005). Special cases of the set packing problem, which can be solved in polynomial time, have been studied in, e.g., Rothkopf et al. (1998) and van Hoesel and Müller (2001).

A recent survey of combinatorial auctions is provided by de Vries and Vohra (2003). Combinatorial auctions can be useful in many environments and have been considered for problems including selling spectrum rights (McMillan 1994, Milgrom 2000), airport take-off & landing time slot allocation (Rassenti et al. 1982), railroad segments (Brewer 1999), and delivery routes (Caplice and Sheffi 2003). Other applications are surveyed in, for instance, Kwon et al. (2005).

If the linear programming relaxation of the winner determination problem has variables that are fractional in the optimal solution the dual prices if used as information will overcharge the agents and hence might lead to that some agents withdraw from the auction too early leading to an inferior outcome. In accordance with this observation several authors, starting from the seminal work of Rassenti et al. (1982), have suggested the use of approximate pseudo-dual prices which can be thought of as prices that are approximately fulfilling the requirements of dual feasibility, primal complementary slackness and dual complementary slackness given the optimal, and thus feasible, solution to the winner determination problem. The pseudo-dual prices are anonymous but do not fulfill the requirement that the bid on a non-winning bundle is less than the sum of the prices of the individual items in the bundle.

3 Basic properties and definitions

First, we define the linear programming relaxation of the winner determination problem, that is, the problem

$$\left.\begin{array}{rll} \bar{z} = \text{max} & \sum\limits_{j \in J} b_j x_j & \\ \text{s.t.} & \sum\limits_{j \in J} a_{ij} x_j \leq 1 & \forall\, i \in I \\ & x_j \geq 0 & \forall\, j \in J \end{array}\right\} \tag{2}$$

and the corresponding dual

$$\left.\begin{array}{rll} \bar{z} = \text{min} & \sum\limits_{i \in I} u_i & \\ \text{s.t.} & \sum\limits_{i \in I} a_{ij} u_i \geq b_j & \forall\, j \in J \\ & u_i \geq 0 & \forall\, i \in I \end{array}\right\} \tag{3}$$

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

For the linear programming relaxation of (1) we know that an optimal primal solution $\bar{\mathbf{x}}^* = (\bar{x}_j^*)$ of (2) and the corresponding optimal dual solution $\mathbf{u}^* = (u_i^*)$ of (3) have the properties provided below.

Property 1 *(primal feasibility)*
An optimal primal solution $\bar{\mathbf{x}}^* = (\bar{x}_j^*)$ *satisfies the constraints*

$$\begin{array}{ll} \sum\limits_{j \in J} a_{ij} \bar{x}_j^* \leq 1 & \forall\, i \in I \\ \bar{x}_j^* \geq 0 & \forall\, j \in J \end{array}$$

and is said to be primal feasible.

Property 2 *(dual feasibility)*

An optimal dual solution $\bar{\mathbf{u}}^* = (\bar{u}_i^*)$ *satisfies the constraints*

$$\begin{aligned} &\sum_{i \in I} a_{ij} u_i^* \geq b_j && \forall\, j \in J \\ &u_i^* \geq 0 && \forall\, i \in I \end{aligned}$$

and is said to be dual feasible.

Property 3 *(primal complementary slackness)*

If an optimal primal solution $(\bar{x}_j^*)$ *and the corresponding optimal dual solution* (u_i^*) *satisfy the constraints*

$$\bar{x}_j^* \left(\sum_{i \in I} a_{ij} u_i^* - b_j \right) = 0 \quad \forall\, j \in J,$$

then the primal complementary slackness condition is assured.

Property 4 *(dual complementary slackness)*

If an optimal primal solution $(\bar{x}_j^*)$ *and the corresponding optimal dual solution* (u_i^*) *satisfy the constraints*

$$u_i^* \left(\sum_{j \in J} a_{ij} \bar{x}_j^* - 1 \right) = 0 \quad \forall\, i \in I,$$

then the dual complementary slackness condition is assured.

Finally, we define the meaning of anonymous vs. non-anonymous prices.

Definition 1 *(anonymous/non-anonymous prices)*

A price is called anonymous if all agents face the same price. If agents face different prices we have a non-anonymous price system.

4 Related work

Over time several suggestions have been made to address the problem of finding interpretable dual prices for integer and mixed-integer programming problems. Two streams of research can be distinguished. First, research related to duality theory for general purpose integer programming problems. Second, work dedicated specifically to the set packing problem.

Integer programming We review this branch of research too, because our approach in some sense relies on integer programming duality theory.

For a primal integer programming problem

$$\left. \begin{aligned} Z = \max \quad & \sum_{j=1}^{n} c_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^{n} \mathbf{a}_j x_j \leq \mathbf{b} && \mathbf{a}_j, \mathbf{b} \in \mathbb{R}^m \\ & x_j \geq 0 \text{ and integer} && j = 1, \ldots, n \end{aligned} \right\} \tag{4}$$

where we assume that $\{\mathbf{a}_j\}_{j=1}^n$ and $\mathbf{b}$ are integer vectors, there exists a dual

$$\left.\begin{aligned} W = \text{min} \quad & F(\mathbf{b}) \\ \text{s.t.} \quad & F(\mathbf{a}_j) \geq c_j \quad j = 1, \dots, n \\ & F \in \mathbf{F} \end{aligned}\right\} \tag{5}$$

where $\mathbf{F} = \{F \in F_+^m : F$ is superadditive and $F(0) \geq 0\}$ and F_+^m denotes the set of nondecreasing functions $F : \mathbb{R}^m \to \mathbb{R}^* = \mathbb{R} \cup \{-\infty, +\infty\}$. The set $\mathbf{F}$ is the set of dual price functions. We adopt the standard convention that $Z = -\infty$ if (4) is infeasible and $Z = +\infty$ if (4) has feasible solutions of arbitrarily large value and a similar convention for (5).

The primal dual pair (4) and (5) have the same properties as in standard linear programming duality, hence given an optimal primal dual pair $(\mathbf{x}^*, F^*)$ the solutions are primal and dual feasible, respectively, and primal complementary slackness is satisfied.

In the decisive work of Gomory and Baumol (1960) dual prices and their relationship to the marginal values of scarce resources have been discussed. Alcaly and Klevorick (1966) address the problem encountered with the approach of Gomory and Baumol that “free goods” might have nonzero prices. Wolsey (1981) gives a concise description of this theory, and shows that in the integer programming case we need to expand our view of prices to price-functions in order to achieve interpretable and computable duals (see also Scarf 1990, Williams 1996 and Sturmfels 2004).

Wolsey (1981) shows that different algorithmic approaches for solving the primal integer program lead to different characterisations of the optimal price functions. Specifically the cutting plane approach shows that there is always an optimal price function of the form

$$F^*(\mathbf{b}) = \sum_{i=1}^{m} \pi_i b_i + \sum_{i=1}^{r} \pi_{m+i} F^i(\mathbf{b}) \tag{6}$$

where

$$F^t(\mathbf{b}) = \left\lfloor \sum_{i=1}^{m} \lambda_i^{t-1} b_i + \sum_{i=1}^{t-1} \lambda_{m+i}^{t-1} F^i(\mathbf{b}) \right\rfloor \tag{7}$$

with $\pi = (\pi_1, \dots, \pi_{m+r}) \geq \mathbf{0}$, $\lambda^{t-1} = (\lambda_1^{t-1}, \dots, \lambda_{m+t-1}^{t-1}) \geq \mathbf{0}$, $t = 1, \dots, r$, where r is the number of cutting planes, and $\lfloor \gamma \rfloor$ denotes the largest integer less than or equal to γ.

If a linear programming based branch and bound algorithm terminates on problem (4), and (4) has a finite optimal value, then (5) has an optimal solution of the form

$$F^*(\mathbf{b}) = \max_{t=1}^{r} \left[\alpha(t) + \mathbf{u}^t \mathbf{b}\right] \tag{8}$$

with $\mathbf{u}^t = (u_1^t, \dots, u_m^t) \geq \mathbf{0}, t = 1, \dots, r$, where t indexes the terminal fathomed nodes for some finite value of r. $(\mathbf{u}^t, \underline{\mathbf{u}}^t, \bar{\mathbf{u}}^t) \geq \mathbf{0}$ is the dual feasible solution associated with node t and $\alpha(t) = -\underline{\mathbf{u}}^t g^t + \bar{\mathbf{u}}^t h^t$ reflects the bounds $g_j^t \leq x_j \leq h_j^t$ on variables x_j obtained through branching.

In order to calculate reduced cost we only have to evaluate each column $F^*(\mathbf{a}_j) - c_j \geq 0$.

Apparently, the dual price function given in the dual above yields a non-linear anonymous price function for every combinatorial auction with the winner determination problem (1). Obviously, the problem with this approach is that the derivation of the price function is very complicated. Recently, Klabjan (2003) has developed an algorithm for computing the subadditive dual function which seems to be practical for the set partitioning problem.

Set packing Another 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 (see properties 1 to 4).

In a combinatorial auction the auctioneer is trying to get a good and hopefully optimal solution to the winner determination problem. Assume that the optimal integer solution $\mathbf{x}^* = (x_j^*)$ to the winner determination problem (1) has been found and that the linear programming relaxation (2) does not have the integrality property. Then we know that there does not exist 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 solution $\mathbf{x}^* = (x_j^*)$ 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) Primal 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) Dual 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 loosing bids for a particular bundle of items 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 et al. (1982) and by DeMartini et al. (1999), among others.

Assume that the winner determination problem (1) has been solved to optimality and that (x_j^*) is the corresponding optimal integer solution. Let $J_0 := \{j \in J : x_j^* = 0\}$ and $J_1 := \{j \in J : x_j^* = 1\}$ denote the set of loosing and winning bids, respectively. Apparently, we have $J_0 \cap J_1 = \emptyset$ and $J_0 \cup J_1 = J$.

In the following we will sketch the approach by DeMartini et al. (1999) and Kwasnica et al. (2005) (other approaches can be found in, e.g., Parkes 2001, Bikhchandani and Ostroy 2002 and Xia et al. 2004). The main component is to solve the linear program (9).

$$\begin{aligned}
\min \quad & w && && \text{(9a)}\\
\text{s.t.} \quad & \sum_{j\in J} a_{ij}u_i + y_j \geq b_j && \forall\, j \in J_0 && \text{(9b)}\\
& \sum_{j\in J} a_{ij}u_i = b_j && \forall\, j \in J_1 && \text{(9c)}\\
& w \geq y_j && \forall\, j \in J_0 && \text{(9d)}\\
& u_i \geq 0 && \forall\, i \in I && \text{(9e)}\\
& y_j \geq 0 && \forall\, j \in J_0 && \text{(9f)}
\end{aligned}$$

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 (9a) has been designed to minimize the number of such bids. 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 (9a) are not unique a sequence of iterations each of which requires to solve the linear program (9) is performed (for details see DeMartini et al. 1999 and Kwasnica et al. 2005).

Recently, Dunford et al. (2003) have shown that pseudo-dual, linear pricing algorithms produce non-monotonity between rounds, something that is expected to disturb bidders in iterative combinatorial auctions. The non-linear anonymous pricing methodology presented in the following section gives the chance to overcome this deficiency.

5 Pricing model

Since we have separated the solution of the winner determination problem and the pricing problem we know the optimal primal solution. Let (x_j^*) denote an optimal integer solution of (1) and let $J_0 := \{j \in J : x_j^* = 0\}$ and $J_1 := \{j \in J : x_j^* = 1\}$ be the corresponding set of loosing and winning bids, respectively. Then the constraints of the pricing problem can be written as follows:

$$Y = \left\{ (\mathbf{y}, \mathbf{u}) \in \mathbb{R}^{|J_0|\times m} : \begin{array}{lll}
\sum_{i\in I} a_{ij}u_i - b_j y_j \geq 0 & \forall\, j \in J_0 & \text{(10a)}\\
\sum_{i\in I} a_{ij}u_i = b_j & \forall\, j \in J_1 & \text{(10b)}\\
y_j \in \{0,1\} & \forall\, j \in J_0 & \text{(10c)}\\
\left.\begin{array}{ll} u_i = 0, & \text{if } \sum_{j\in J} a_{ij}x_j^* < 1 \\ u_i \geq 0, & \text{otherwise}\end{array}\right\} & \forall\, i \in I & \text{(10d)}
\end{array} \right\}$$

Constraints (10a) in conjunction with the second branch of (10d) assure dual feasibility. Constraints (10b) assure primal complementary slackness. Constraint (10c) requires the variables corresponding to the loosing bids to be binary. Finally, the first branch of (10d) addresses dual complementary slackness. Hence, overall the constraints assure that any feasible solution of the mixed-integer pricing model has the properties 1 to 4.

Note that both constraints (10a) and (10b) originate from the same (in-)equality. In particular, (10b) is the result of fixing y_j^* to 1 for all bids $j \in J$ with $x_j^* = 1$, that is, the original winning bids are also winning bids in the pricing problem. The equality sign assures that these bids have reduced cost of zero.

Now we are ready to formulate the pricing model as follows:

$$\max \left\{ \sum_{j \in J_0} y_j : (\mathbf{y}, \mathbf{u}) \in Y \right\} \tag{11}$$

The objective function of (11) aims at maximizing the number of loosing bids being covered in the optimal solution. We will show now that this objective is the primary choice (see section 8, too). An important property of dual prices is to assure that as many loosing bids $j \in J_0$ as possible are priced out, that is, have non-positive reduced cost $\bar{b}_j := b_j - \sum_{i \in I} a_{ij} u_i \leq 0$. The following corollary states that the objective of our pricing model takes care of this characteristic.

Corollary 1 *The mixed-integer pricing model prices out as many loosing bids as possible.*

Remark 1 *The objective of (11) guarantees that in the case that the winner determination problem has the integrality property only one price will be generated, that is optimal linear programming shadow prices.*

Apparently, solving model (11) produces one price vector which has the desired properties. But the question is how to compute other prices to which the max operator can be applied. This can be achieved easily by adding cover cuts to (11) as follows. Assume that (11) has been solved to optimality and let (y_j^*) denote the optimal binary variables. Let $K_0 = \{j \in J_0 : y_j^* = 0\}$ and $K_1 = \{j \in J_0 : y_j^* = 1\}$ denote the subset of loosing and winning bids, respectively. If we define

$$Y' = \left\{ (\mathbf{y}, \mathbf{u}) \in Y : \sum_{j \in K_0} y_j \geq 1 \right\}$$

and solve max $\{\sum_{j \in J_0} y_j : (\mathbf{y}, \mathbf{u}) \in Y'\}$ instead of (11) we get another price vector. Iterating this way produces a sequence of up to $|J_0| + 1$ prices. The following corollary highlights the fact that the procedure generates a non-linear anonymous price system.

Corollary 2 *The price system has the form*

$$F^*(\mathbf{d}) = \max_{t=1}^{|J_0|+1} [\mathbf{u}^t \mathbf{d}].$$

In case of $\mathbf{d} = \mathbf{1}$ we get the optimal objective function value, in case of $\mathbf{d} = \mathbf{a}_j$ we get the reduced cost for column j by evaluating $F^*(\mathbf{a}_j) - b_j \geq 0$.

In practice $|J_0|+1$ might be too large to be handled by the agents and, hence, we propose to proceed as follows. Again assume that (11) has been solved to optimality and let (y_j^*) denote the optimal binary variables. Let

$$K_0^\mu = K_0^{\mu-1} \setminus \{j \in K_0^{\mu-1} : y_j^* = 1\}$$

and

$$K_1^\mu = K_1^{\mu-1} \cup \{j \in K_0^{\mu-1} : y_j^* = 1\}$$

indicate that we move bids that currently are priced out. Initially, we have $K_0^1 = \{j \in J_0 : y_j^* = 0\}$ and $K_1^1 = \{j \in J_0 : y_j^* = 1\}$. Then we solve in iteration $\mu = 2, 3, \ldots$ the optimization problem

$$\max\left\{\epsilon \sum_{j \in K_1^{\mu-1}} y_j + \sum_{j \in K_0^{\mu-1}} y_j : (\mathbf{y}, \mathbf{u}) \in Y\right\} \tag{12}$$

where $\epsilon = \frac{1}{|J_0|}$. The objective function of (12) aims at letting as many as possible so far loosing bids win. This objective, in particular the choice of ϵ, lexicographically search for alternative optimal solutions in which bids loosing in the previous iteration become winning bids in the current one. The iteration terminates once we have $|K_1^\mu| = |K_1^{\mu-1}|$ yielding in total $q \leq |J_0| + 1$ prices.

In the following section we will illustrate by means of a computational study that this approach is effective and efficient. Before doing so we would like to notice that the non-linear anonymous price system is fairly simple. Whether this price system is efficient in practical use remains to be tested experimentally.

An example with 6 items and 21 bids taken from Parkes (2001) illustrates the idea. The bid prices (b_j) and the coeficient matrix (a_{ij}) are provided in Table 1. Note that the bids 4, 12, 14 and 20 are superadditive.

j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
b_j	60	50	50	200	100	110	250	50	60	50	110	200	100	255	50	50	75	100	125	200	250
a_{1j}	1			1		1	1	1			1		1	1	1			1		1	1
a_{2j}		1		1	1		1		1		1	1		1		1		1	1		1
a_{3j}			1		1	1	1			1		1	1	1			1		1	1	1
a_{4j}	1	1	1	1	1	1	1														
a_{5j}								1	1	1	1	1	1	1							
a_{6j}															1	1	1	1	1	1	1

Table 1: Instance – Parkes (2001)

integer program (1)	linear program (2)
$x_4^* = x_{17}^* = 1$	$\bar{x}_4^* = \bar{x}_{12}^* = \bar{x}_{20}^* = 0.5$
OFV = 275	OFV = 300

Table 2: Instance – results

h	$\mathbf{y}^h$	$\mathbf{u}^h$
1	$y_{12} = 0$	(125,75, 75, 0, 0, 0)
2	$y_{20} = 0$	(60, 140, 75, 0, 0, 0)

Table 3: Bids selected and price system

Table 2 provides the solution (x_j^*) of the integer program (1) and the solution $(\bar{x}_j^*)$ of the linear programming relaxation (2) for this instance. Variables not given there have value 0.

OFV abbreviates optimal objective function value. Our pricing model generates the vectors $\mathbf{y}^h$ and $\mathbf{u}^h$ $(h = 1, 2)$ displayed in table 3 (variables y_j not given explicitely have value 1).

One might ask why items 4, 5 and 6 have price 0 in all three cases. The reason is as follows. Consider, for instance, bids 4 and 17. Bid 17 complements bid 4 in the sense that two of the three items not contained in bid 4 are chosen. Bid 7, on the other hand, contains item 3, which is not contained in bid 4, at the bid price 250. If we look now at bids 3 and 10 we can easily detect that their bids price of 50 is attributed to item 3 only. As a consequence, the price of items 4 and 5 equals 0.

bid	reduced cost		
4	max (200, 200) – 200	=	0
12	max (215, 135) – 200	=	15
20	max (135, 215) – 200	=	15

Table 4: Reduced cost

In a planned experimental study we will try to find out how agents behave when facing these non-linear prices as compared with alternatives such as pseudo-dual prices or discriminatory prices. However, before leaving the example we would like to point out, that alternative sets of prices exist, for instance the two extreme prices (60, 140, 75, 0, 0, 0) and (140, 60, 75, 0, 0, 0). The calculation of the reduced cost for the bids 4, 12 and 20 is displayed in table 4, indicating that for bids 12 and 20 a rise of more than 15 in bid price must be done by the agents. For our method to generate this price system we have to revise the objective function as outlined in section 8.

6 Pricing model and reduced combinatorial auctions

In the following we establish relations between the mixed-integer pricing model and what we call reduced combinatorial auction.

Definition 2 *(reduced combinatorial auction)*

A reduced combinatorial auction is an auction where some bids have been eliminated such that the integrality property holds and, hence, a linear price system exists.

Recall that (x_j^*) is an optimal integer solution of (1) and that $J_1 := \{j \in J : x_j^* = 1\}$ and $J_0 := J \setminus J_1$ is the corresponding set of winning and loosing bids, respectively. Moreover, assume that (11) has been solved to optimality and let (y_j^*) denote the optimal binary variables. Furthermore, let $K_1 = \{j \in J_0 : y_j^* = 1\}$ denote the subset of winning bids.

The following corollary establishes a close relationship between the pricing model and reduced combinatorial auctions.

Corollary 3 *Given a solution of the pricing model (11) only containing bids $J_1 \cup K_1$ the corresponding reduced auction restricted to the bids $J_1 \cup K_1$ has the integrality property. Moreover, the optimal solution of the reduced auction with objective function value z is an optimal solution of the original winner determination problem (1), too.*

It is easy to see that this is true also for the subsequent solutions to the pricing model (12). Hence, the sequential solution to the pricing model corresponds to the construction of a set of reduced auctions each of which has the integrality property.

For illustrative purposes look at the example of Parkes (2001) given above. The two reduced auctions are one auction with all bids but bid 12 and one auction with all bids but bid 20.

7 Computational results

In the following we will present numerical results which have been obtained with model (12) 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 [17]) which provides a set of distributions for modeling realistic bidding behavior. In particular, we have generated instances using the built-in distributions `arbitrary`, `matching`, `paths`, `regions`, and `scheduling`, respectively.

The methods described earlier have been imlemented using the CPLEX callable library (version 9.0) on an AMD Athlon with 1 GB RAM and 2.1 Ghz clockpulse.

The results of the numerical experiments are given in tables 5 to 9. Each of the tables is structured as follows:

- Column 1 displays the number of items m.
- In column 2 the number of bids $n = m \cdot \rho$ with $\rho \in \{2, 3, \ldots, 10, 20, \ldots, 100\}$ is given. For each combination of m and n we have generated 10 instances.
- Columns 3 to 5 provide the minimum (Min), average (Avg) and maximum (Max) number of prices q calculated.
- Columns 6 to 8 show the minimum, average and maximum CPU time in seconds required for solving the sequence of mixed-integer programs (12).[1]
- Column 9 displays the number of instances (# LP) for which the linear programming relaxation of the set packing problem (1) had the integrality property, that is, the number of cases in which there was no need to solve our mixed-integer pricing model (12).
- Finally, column 10 displays the number of instances (# MIP) for which we had to solve our mixed-integer pricing model (12).

Note that columns 9 and 10 complement each other, that is, sum up to the number 10 of instances generated for each combination of m and n. Furthermore, the number of prices q covers only the subset of instances for which the integrality property was not observed. Otherwise the minimum number of prices generated is, of course, one (in this case columns 3 to 8 show "–").

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

[1] Notice that the time needed in order to solve the original combinatorial auction (1) is not included.

		q (# of prices)			time (pricing, sec)				
m	n	Min	Avg	Max	Min	Avg	Max	# LP	# MIP
20	40	3.00	3.12	4.00	0.11	0.26	0.49	2	8
	60	3.00	3.10	4.00	0.20	6.29	37.07	0	10
	80	3.00	3.20	4.00	0.34	634.47	6162.70	0	10
	100	3.00	3.60	4.00	3.94	1387.08	5324.35	0	10
	120	3.00	3.20	4.00	0.20	352.75	1939.47	0	10
	140	3.00	3.40	4.00	0.12	5465.71	51556.56	0	10
	160	3.00	3.40	4.00	0.20	6998.41	46212.63	0	10
	180	3.00	3.10	4.00	0.48	3030.19	28071.66	0	10
	200	3.00	3.30	4.00	0.50	3009.56	16238.22	0	10
30	60	3.00	3.50	4.00	0.34	22.36	97.24	0	10
	90	3.00	3.60	4.00	4.60	5604.50	24664.21	0	10
	120	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.
40	80	3.00	3.70	5.00	5.08	2140.82	9784.77	0	10
	120	3.00	4.00	5.00	1102.67	30277.41	59452.15	0	2
	160	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.

Table 5: Distribution type `arbitrary`

		q (# of prices)			time (pricing, sec)				
m	n	Min	Avg	Max	Min	Avg	Max	# LP	# MIP
20	40	3.00	3.00	3.00	0.01	0.01	0.01	9	1
	60	–	–	–	–	–	–	10	0
	80	–	–	–	–	–	–	10	0
	100	3.00	3.00	3.00	0.01	0.01	0.02	7	3
	120	3.00	3.00	3.00	0.02	0.02	0.02	8	2
	140	–	–	–	–	–	–	10	0
	160	3.00	3.00	3.00	0.04	0.04	0.04	9	1
	180	–	–	–	–	–	–	10	0
	200	3.00	3.00	3.00	0.02	0.03	0.05	8	2
	400	–	–	–	–	–	–	10	0
	600	3.00	3.00	3.00	0.12	0.12	0.12	9	1
	800	–	–	–	–	–	–	10	0
	1000	–	–	–	–	–	–	10	0
	1200	3.00	3.00	3.00	0.24	0.24	0.24	9	1
	1400	–	–	–	–	–	–	10	0
		and so on for $n \in \{1600, 1800, 2000\}$							
40	80	2.00	2.20	3.00	0.01	0.01	0.02	5	5
	120	2.00	2.00	2.00	0.01	0.01	0.01	8	2
	160	2.00	2.00	2.00	0.01	0.01	0.01	8	2
	200	2.00	2.33	3.00	0.01	0.02	0.05	7	3
	240	2.00	2.00	2.00	0.01	0.01	0.01	9	1
	280	–	–	–	–	–	–	10	0
		and so on for $n \in \{320, 360, 400\}$							
	800	3.00	3.00	3.00	0.18	0.18	0.18	9	1
	1200	–	–	–	–	–	–	10	0
		and so on for $n \in \{1600, \ldots, 4000\}$							

Table 6: Distribution type `matching`

m	n	q (# of prices) Min	Avg	Max	time (pricing, sec) Min	Avg	Max	# LP	# MIP
20	40	2.00	2.33	3.00	0.00	0.00	0.01	4	6
	60	2.00	2.00	2.00	0.00	0.00	0.01	5	5
	80	2.00	2.40	3.00	0.00	0.01	0.02	5	5
	100	2.00	2.00	2.00	0.00	0.00	0.01	5	5
	120	2.00	2.00	2.00	0.00	0.00	0.01	3	7
	140	2.00	2.14	3.00	0.01	0.01	0.02	3	7
	160	2.00	2.00	2.00	0.01	0.01	0.02	4	6
	180	2.00	2.00	2.00	0.00	0.00	0.01	2	8
	200	2.00	2.00	2.00	0.01	0.01	0.02	1	9
	400	2.00	2.00	2.00	0.02	0.02	0.03	0	10
	600	2.00	2.00	2.00	0.04	0.04	0.05	0	10
	800	2.00	2.00	2.00	0.05	0.06	0.08	2	8
	1000	2.00	2.00	2.00	0.08	0.09	0.11	0	10
	1200	2.00	2.00	2.00	0.11	0.12	0.15	0	10
	1400	2.00	2.00	2.00	0.14	0.17	0.22	0	10
	1600	2.00	2.00	2.00	0.17	0.20	0.25	0	10
	1800	2.00	2.00	2.00	0.17	0.23	0.30	0	10
	2000	2.00	2.00	2.00	0.23	0.27	0.32	0	10
30	60	2.00	2.16	3.00	0.00	0.00	0.01	4	6
	90	2.00	2.20	3.00	0.01	0.01	0.02	5	5
	120	2.00	2.10	3.00	0.01	0.01	0.03	0	10
	150	2.00	2.12	3.00	0.00	0.01	0.02	2	8
	180	2.00	2.16	3.00	0.01	0.01	0.02	4	6
	210	2.00	2.00	2.00	0.01	0.01	0.02	1	9
	240	2.00	2.00	2.00	0.01	0.01	0.02	1	9
	270	2.00	2.11	3.00	0.01	0.01	0.04	1	9
	300	2.00	2.00	2.00	0.01	0.02	0.03	1	9
	600	2.00	2.00	2.00	0.04	0.04	0.06	0	10
	900	2.00	2.00	2.00	0.07	0.07	0.09	0	10
	1200	2.00	2.00	2.00	0.10	0.12	0.14	0	10
	1500	2.00	2.00	2.00	0.17	0.18	0.20	0	10
	1800	2.00	2.00	2.00	0.23	0.24	0.27	0	10
	2100	2.00	2.00	2.00	0.28	0.32	0.39	0	10
	2400	2.00	2.00	2.00	0.34	0.39	0.44	0	10
	2700	2.00	2.00	2.00	0.43	0.48	0.53	0	10
	3000	2.00	2.00	2.00	0.52	0.59	0.66	0	10
40	80	2.00	2.20	3.00	0.00	0.01	0.02	5	5
	120	2.00	2.20	3.00	0.00	0.01	0.03	0	10
	160	2.00	2.12	3.00	0.01	0.01	0.03	2	8
	200	2.00	2.37	3.00	0.00	0.01	0.05	2	8
	240	2.00	2.50	3.00	0.01	0.03	0.05	2	8
	280	2.00	2.20	3.00	0.01	0.02	0.07	0	10
	320	2.00	2.12	3.00	0.02	0.02	0.05	2	8
	360	2.00	2.20	3.00	0.02	0.03	0.07	0	10
	400	2.00	2.37	3.00	0.03	0.04	0.08	2	8
	800	2.00	2.00	2.00	0.07	0.08	0.10	2	8
	1200	2.00	2.00	2.00	0.13	0.14	0.15	0	10
	1600	2.00	2.00	2.00	0.21	0.22	0.24	0	10
	2000	2.00	2.00	2.00	0.29	0.32	0.36	0	10
	2400	2.00	2.00	2.00	0.38	0.45	0.50	0	10
	2800	2.00	2.00	2.00	0.55	0.62	0.74	0	10
	3200	2.00	2.00	2.00	0.62	0.73	0.84	0	10
	3600	2.00	2.00	2.00	0.82	0.90	1.00	0	10
	4000	2.00	2.00	2.00	0.95	1.06	1.21	0	10

Table 7: Distribution type paths

		q (# of prices)			time (pricing, sec)				
m	n	Min	Avg	Max	Min	Avg	Max	# LP	# MIP
20	40	3.00	3.00	3.00	0.02	0.03	0.05	8	2
	60	3.00	3.00	3.00	0.06	0.06	0.06	9	1
	80	3.00	3.00	3.00	0.04	0.06	0.10	5	5
	100	3.00	3.00	3.00	0.09	0.12	0.17	5	5
	120	3.00	3.00	3.00	0.03	0.18	0.60	1	9
	140	3.00	3.00	3.00	0.09	0.26	0.57	4	6
	160	3.00	3.00	3.00	0.07	0.45	1.18	6	4
	180	3.00	3.33	4.00	0.16	7.10	20.88	7	3
	200	3.00	3.00	3.00	0.13	0.77	1.64	5	5
	400	3.00	3.00	3.00	0.17	0.62	1.69	3	7
	600	3.00	3.00	3.00	0.28	0.80	2.25	6	4
	800	3.00	3.00	3.00	0.45	0.92	1.95	4	6
	1000	3.00	3.00	3.00	0.50	1.56	4.66	3	7
	1200	3.00	3.00	3.00	0.93	4.01	7.10	8	2
	1400	3.00	3.00	3.00	1.75	2.59	4.23	7	3
	1600	3.00	3.00	3.00	0.86	0.86	0.86	9	1
	1800	3.00	3.00	3.00	0.72	1.79	4.92	5	5
	2000	3.00	3.00	3.00	1.40	1.40	1.40	9	1
30	60	3.00	3.00	3.00	0.06	0.19	0.44	5	5
	90	3.00	3.00	3.00	0.09	0.70	1.85	4	6
	120	3.00	3.00	3.00	0.09	1.53	4.15	3	7
	150	3.00	3.00	3.00	0.14	41.99	198.52	4	6
	180	3.00	3.14	4.00	0.06	6.35	33.57	3	7
	210	3.00	3.00	3.00	0.22	14.92	83.06	4	6
	240	3.00	3.11	4.00	0.09	13.58	91.91	1	9
	270	3.00	3.10	4.00	0.10	48.82	434.61	0	10
	300	3.00	3.00	3.00	0.26	36.77	244.83	0	10
	600	3.00	3.00	3.00	0.75	96.38	572.69	1	9
	900	3.00	3.00	3.00	1.60	4.49	12.10	3	7
	1200	3.00	3.00	3.00	0.94	1.78	2.21	5	5
	1500	3.00	3.00	3.00	1.55	5.18	12.82	5	5
	1800	3.00	3.00	3.00	2.11	3.60	6.41	3	7
	2100	3.00	3.00	3.00	1.54	7.98	15.43	6	4
	2400	3.00	3.00	3.00	2.09	4.46	7.92	6	4
	2700	3.00	3.00	3.00	2.09	5.33	10.27	5	5
	3000	3.00	3.00	3.00	1.90	1.90	1.90	9	1
40	80	3.00	3.00	3.00	0.21	11.83	46.38	4	6
	120	3.00	3.00	3.00	0.09	1.69	4.36	2	8
	160	3.00	3.20	4.00	0.17	1375.57	7357.25	0	10
	200	3.00	3.10	4.00	2.36	2468.21	9385.06	0	10
	240	3.00	3.00	3.00	0.18	7178.26	59318.16	0	10
	280	3.00	3.00	3.00	0.45	5640.98	48675.48	1	9
	320	3.00	3.00	3.00	2.63	3007.33	24661.61	1	9
	360	3.00	3.11	4.00	2.42	6948.36	57075.97	1	9
	400	3.00	3.00	3.00	6.67	3443.83	14722.23	2	8
	800	3.00	3.11	4.00	1.87	31.09	162.88	1	9
	1200	3.00	3.00	3.00	3.06	21.25	68.06	2	8
	1600	3.00	3.00	3.00	1.32	30.97	224.24	0	10
	2000	3.00	3.00	3.00	1.95	14.07	55.11	3	7
	2400	3.00	3.00	3.00	2.59	12.51	19.44	3	7
	2800	3.00	3.00	3.00	1.94	20.66	63.03	6	4
	3200	3.00	3.00	3.00	22.15	25.36	29.46	6	4
	3600	3.00	3.00	3.00	5.40	13.88	33.20	5	5
	4000	3.00	3.00	3.00	3.16	13.14	26.30	6	4

Table 8: Distribution type `regions`

		q (# of prices)			time (pricing, sec)				
m	n	Min	Avg	Max	Min	Avg	Max	# LP	# MIP
20	40	2.00	2.20	3.00	0.00	0.00	0.02	5	5
	60	2.00	2.66	3.00	0.00	0.03	0.10	4	6
	80	2.00	2.50	3.00	0.00	0.01	0.02	6	4
	100	2.00	2.50	3.00	0.00	0.04	0.15	4	6
	120	2.00	2.75	3.00	0.01	0.04	0.09	6	4
	140	2.00	2.87	3.00	0.02	0.37	2.18	2	8
	160	2.00	2.00	2.00	0.01	0.01	0.01	9	1
	180	2.00	3.00	4.00	0.00	0.05	0.20	5	5
	200	2.00	2.60	3.00	0.00	0.03	0.07	5	5
	400	2.00	2.33	3.00	0.00	0.05	0.12	7	3
	600	2.00	2.50	3.00	0.04	0.17	0.30	8	2
	800	2.00	2.33	3.00	0.05	0.15	0.35	7	3
	1000	3.00	3.00	3.00	0.28	0.28	0.29	8	2
	1200	–	–	–	–	–	–	10	0
	1400	2.00	2.50	3.00	0.05	0.47	0.89	8	2
	1600	2.00	2.00	2.00	0.07	0.10	0.14	8	2
	1800	2.00	2.50	3.00	0.12	0.13	0.15	8	2
	2000	–	–	–	–	–	–	10	0
30	60	2.00	2.00	2.00	0.00	0.00	0.01	2	8
	90	2.00	2.37	3.00	0.00	0.05	0.22	2	8
	120	2.00	2.60	3.00	0.01	0.13	0.52	0	10
	150	2.00	2.88	3.00	0.01	0.47	1.51	1	9
	180	2.00	2.71	3.00	0.00	0.41	1.14	3	7
	210	2.00	2.85	3.00	0.02	1.96	8.05	3	7
	240	2.00	2.60	3.00	0.01	0.66	3.06	5	5
	270	3.00	3.00	3.00	0.08	2.18	10.61	4	6
	300	2.00	2.80	3.00	0.03	0.10	0.22	5	5
	600	2.00	2.62	3.00	0.04	0.63	3.21	2	8
	900	2.00	2.83	4.00	0.02	1.08	3.69	4	6
	1200	2.00	2.88	3.00	0.08	19.31	165.30	1	9
	1500	2.00	2.66	4.00	0.04	0.22	0.92	4	6
	1800	2.00	2.80	3.00	0.06	0.66	1.75	5	5
	2100	3.00	3.16	4.00	0.11	0.98	3.42	4	6
	2400	2.00	2.50	3.00	0.09	0.95	2.20	4	6
	2700	2.00	2.25	3.00	0.11	0.71	2.30	6	4
	3000	2.00	3.00	4.00	0.19	0.76	2.91	5	5
40	80	2.00	2.14	3.00	0.00	0.02	0.14	3	7
	120	2.00	2.28	3.00	0.00	0.02	0.09	3	7
	160	2.00	2.55	3.00	0.00	21.19	182.54	1	9
	200	2.00	2.42	3.00	0.02	168.52	646.28	3	7
	240	2.00	2.75	3.00	0.01	78.24	583.91	2	8
	280	2.00	2.88	4.00	0.01	4941.79	44388.71	1	9
	320	2.00	2.90	3.00	0.02	78.16	715.99	0	10
	360	2.00	2.66	3.00	0.02	302.71	2330.65	1	9
	400	2.00	2.66	3.00	0.03	0.86	4.27	4	6
	800	2.00	2.88	3.00	0.03	0.55	2.51	1	9
	1200	2.00	2.85	3.00	0.05	10.22	54.68	3	7
	1600	2.00	2.57	3.00	0.13	0.44	0.91	3	7
	2000	2.00	2.83	3.00	0.19	270.47	1609.03	4	6
	2400	2.00	2.66	3.00	0.08	3.98	21.15	4	6
	2800	2.00	3.00	4.00	0.15	2.89	13.90	3	7
	3200	2.00	2.57	4.00	0.14	1319.30	9230.70	3	7
	3600	2.00	2.71	3.00	0.34	48.14	323.08	3	7
	4000	2.00	2.66	3.00	0.17	4.45	15.60	4	6

Table 9: Distribution type `scheduling`

- The instances of the distribution type `arbitrary` are hard in the sense that the integrality property shows up rarely and that the mixed-integer pricing problems are difficult to solve. This is the reason why we did not consider the full range of m and n values in table 5 and why we have solved only 2 instances with 40 items and 120 bids. "n.a." means not available.

- As can be observed from table 6 the instances of the distribution type `matching`[2] are easy. On the one hand most of the instances have the integrality property. On the other hand the mixed-integer pricing problems can be solved quickly. Most important, the number of prices generated upon termination of our algorithm never exceeds three.

- Instances of the distribution type `paths` are also easily accessible by our methodology (see table 7). Although the integrality property cannot be observed as oftenly as for the previous type the pricing problems can be soved quickly. The number of prices which have to be generated is very small, too.

- For the distribution type `regions` there is much greater variety (see table 8). In case of 40 items and 240 to 400 bids some mixed-integer programs are challenging from a computational point of view (in each case only one instances requires excessive CPU time). Fortunately, however, the number of prices generated remains small.

- For the distribution type `scheduling` the picture is very much the same as for the previous type (see table 9). Again, for 40 items and 280 bids one instances requires excessive CPU time.

Note that excessive computation times for some instances are due to the objective function chosen in the pricing model (12). Restricted experiments with a variant of the objective cut down computation times drastically. However, our impression was that the prices then did not contain that much information as for the case reported above.

Summarizing the computational study has shown that the methodology developed is effective and efficient in calculating a non-linear, anonymous price system containing only a few prices. Despite the distribution type `arbitrary` (which is supposed to be not that important from a practical point of view) the pricing problem in general can be solved quickly.

8 Research opportunities

In this paper we have provided a means to obtain a simple form of a non-linear anonymous price system that supports the optimal allocation of bids to bidders in a combinatorial auction. Our method essentially requires to solve a few mixed-integer programs. The procedure has been shown to be effective and efficient by means of experiments using instances of the combinatorial auctions test suite [17]. The methodology developed in this paper opens up many research opportunities which will be sketched in the following.

Algorithms For large-size instances it might be too demanding from a computational point of view to solve the mixed-integer programs (11) and (12), that is, fast and reliable heuristics

[2]Note that due to some reasons for $m = 30$ no instance can be generated at all.

are needed. The following observations might serve as starting point for the development of special purpose heuristics.

- If the original combinatorial auction (1) has the integrality property then the optimal solution of the linear programming relaxation of the mixed-integer program (11) has an optimal solution $(\mathbf{y}^*, \mathbf{u}^*)$ with all variables $j \in J_0$ equal to one and thus yielding the shadow prices to the linear program.

- If the original combinatorial auction (1) does not have the integrality property, i.e. $x_j^* \neq \bar{x}_j^*$ for some j, the optimal objective function value for the mixed-integer pricing problem (11) will be less than the cardinality of J_0. Hence, some of the binary variables $y_j, j \in J_0$, have fractional values.

Due to the structure of the mixed-integer pricing problem (11) we can round down all fractional variables y_j to zero and produce a feasible solution to the mixed-integer program. Then we have produced a price vector with the required property to price out all non-winning bids except the bids the corresponding binary variables of which have been rounded down to zero.

This gives rice to a nice heuristic for finding the non-linear price function we are looking for which is based on the successive solution of various linear programming relaxations of the mixed-integer pricing problem (11). More precisely, the linear prices that are determined when solving the mixed-integer pricing problem are the partial prices needed to build up a non-linear anonymous price system for a combinatorial auction. In order to get a non-linear price system that is as simple as possible for a given problem the goal is to minimize the number of prices that need to be generated from the mixed-integer pricing problem in order to price out all loosing bids. A sketch of the algorithm can be given as follows:

1. Solve the linear programming relaxation of the pricing problem (11). Round down all binary variables that are fractional and resolve the linear program with these variables fixed to zero.

2. Based on the prices (u_i) derived in step 1 cheque by scanning the bids greedily if one or more of the variables that are rounded down can be added with value 1 without changing the dual prices. If so add these variables.

3. For the remaining, say k, variables generate a new linear programming relaxation in which all these k variables are required to take on value 1. If this problem is feasible (which usually is not the case) we have found a linear price system that clears the market consisting of the prices derived in steps 1 and 3. The price system consists of two linear prices on which the max operator can be applied in order to produce reduced cost.

4. If the relaxation in step 3 is infeasible create an alternative relaxation by enforcing that at least one of the remaining k variables must take the value 1 and resolve the linear program. Using the prices derived check if some of the remaining $k-1$ variables can be added with value 1 without changing the price system. If so do it. Repeat step 4 until the stack of variables that are not priced out is eliminated. Overall in this step k linear programs have to be solved.

5. The result is a price function consisting of $q+1$ linear prices if the number of repetitions in step 4 is q. The non-linear price function sought for is obtained by applying the max operator to these $q+1$ prices.

Note that it is beyond the scope of this paper to implement this algorithm and to evaluate the results.

Other objective functions First of all one might ask whether the objective function of model (11) is a good or not? Three alternatives which come into mind quickly are as follows:

$$\max \left\{ \sum_{j \in J_0} b_j y_j : (\mathbf{y}, \mathbf{u}) \in Y \right\} \tag{13}$$

$$\max \left\{ \sum_{j \in J_0} \left(\sum_{i \in I} a_{ij} \right) y_j : (\mathbf{y}, \mathbf{u}) \in Y \right\} \tag{14}$$

$$\min \left\{ \sum_{i \in I} \left(\sum_{j \in J_0} a_{ij} \right) u_i - \sum_{j \in J_0} b_j y_j : (\mathbf{y}, \mathbf{u}) \in Y \right\} \tag{15}$$

Compared to (11) the objective (13) takes into account the value of the loosing bids to be included. Objective (14) looks at the number of items covered by the loosing bids. Finally, objective (15) aims at minimizing the sum of the "distances" of the item prices from their bid price. A generalization of the objective of our pricing model (15) is as follows:

$$\min \left\{ d \left(\left(\sum_{i \in I} a_{ij} u_i \right)_{j \in J_0}, \left(b_j \, y_j \right)_{j \in J_0} \right) : (\mathbf{y}, \mathbf{u}) \in Y \right\} \tag{16}$$

(16) is a general metric which contains the l_1-norm of (15) as special case. Note that both (15) and (16) are related to the question which we raised at the end of section 5.

Other application domains In the short run we will taylor the approach to models with binary variables and semi-assignment constraints such as, for instance, the uncapacitated facility location problem and the generalized assignment problem, respectively. Combinatorial exchange auctions with multiple buyers and sellers are on the agenda as well (see Parkes et al. 2005, Xia et al. 2005). In the long run, of course, the aim is to generalize the methodology to general purpose binary or mixed-integer programming problems.

Experimental economics In order to make the methodology useful for practice experiments have to be conducted. Such experiments should, for instance, show how many prices, probably supported with their convex combinations, can be handled by agents. Presumably such experiments first should be done by means of a computational study (using, e.g., BidPots, the intelligent-agent auction simulator of Dunford et al. 2003) and later on with "real" agents (such as, e.g., students in a lab).

Miscellaneous The following constraint forces the sum of prices to make-up the optimal objective function value z of the winner determination problem (1).

$$\sum_{i \in I} u_i = z \tag{17}$$

This can be looked upon as an aggregate of the constraints (10b). If the constraints (10b) are deleted from the pricing model and replaced by constraint (17) we get more freedom in the selection of item prices at the expense that winning bids might have reduced cost different from zero. If this is an attractive alternative needs to be tested. Note that the pricing problem so obtained corresponds to an auction in which all winning bids have been aggregated into one single bid; see Drexl and Jørnsten (2005).

9 Conclusions

In this paper we have provided a means to obtain a simple form of a non-linear anonymous price system that supports the optimal allocation of bids to bidders in a combinatorial auction. The computational tests show that the number of prices needed does not grow with increasing problem size. Our method essentially requires to solve a sequence of mixed-integer programming formulations of the pricing problem.

Interestingly, Dunford et al. (2003) have shown that dual-based linear pricing algorithms produce non-monotonity between rounds – something that bidders might find disturbing. There is serious hope that the non-linear anonymous price system provided by our methodology will overcome this deficiency.

References

[1] ALCALY, R.E., KLEVORICK, A.K. (1966), A note on the dual prices of integer programs, Econometrica, Vol. 34, pp. 206–214

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

[3] BORNDÖRFER, R. (1998), *Aspects of set packing, partitioning, and covering*, PhD Dissertation, Technical University of Berlin

[4] BREWER, P.J. (1999), Decentralized computation procurement and computational robustness in a smart market, Economic Theory, Vol. 13, pp. 41–92

[5] CAPLICE, C.G., SHEFFI, Y. (2003), Optimization-based procurement for transportation services, Journal of Business Logistics, Vol. 24, No. 2, pp. 109–128

[6] DELORME, X., GANDIBLEUX, X., RODRIGUEZ, J. (2004), GRASP for set packing problems, European Journal of Operational Research, Vol. 153, pp. 564–580

[7] 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

[8] DE VRIES, S., VOHRA, R. (2003), Combinatorial auctions: a survey, INFORMS Journal on Computing, Vol. 15, pp. 284–309

[9] DREXL, A., JØRNSTEN, K. (2005), Reflections about pseudo-dual prices in combinatorial auctions, Working Paper No. 590, University of Kiel

[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

[11] GAREY, M.R., JOHNSON, D.S. (1979), *Computers and intractability: a guide to the theory of NP-completeness*, Freeman, San Francisco, CA

[12] GOMORY, R.E., BAUMOL, W.J. (1960), Integer programming and pricing, Econometrica, Vol. 28, pp. 521–550

[13] 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

[14] HARCHE, F., THOMPSON, G.L. (1994), The column substraction algorithm: an exact method for solving weighted set covering, packing and partitioning problems, Computers & Operations Research, Vol. 21, pp. 689–705

[15] VAN HOESEL, S., MÜLLER, R. (2001), Optimization in electronic markets: examples in combinatorial auctions, Netnomics, Vol. 3, pp.23–33

[16] HOFFMANN, K., PADBERG, M.W. (1993), Solving airline crew scheduling problems by branch and cut, Management Science, Vol. 39, pp. 657–672

[17] `http://robotics.stanford.edu/CATS`

[18] KLABJAN, D. (2003), A practical algorithm for computing a subadditive dual function for set partitioning, Technical Report, University of Illinois at Urbana-Champain

[19] KLEMPERER, P. (2002), What really matters in auction design, Journal of Economic Perspectives, Vol. 16, pp. 169–189

[20] 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

[21] KWON, R.H., ANANDALINGAM, G., UNGAR, L.H. (2005), Iterative combinatorial auctions with bidder-determined combinations, Management Science, Vol. 51, pp. 407–418

[22] MCAFEE, R.P., MCMILLAN, J. (1987), Auctions and bidding, Journal of Economic Literature, Vol. 25, pp. 699–738

[23] MCMILLAN, J. (1994), Selling spectrum rights, Journal of Economic Perspectives, Vol. 8, pp. 145–162

[24] MILGROM, P. (1989), Auctions and bidding: a primer, Journal of Economic Perspectives, Vol. 3, pp. 3-22

[25] MILGROM, P. (2000), Putting auction theory to work: the simultaneous ascending auction, Journal of Political Economy, Vol. 108, pp. 245–272

[26] NEMHAUSER, G.L., WOLSEY, L.A. (1988), *Integer and combinatorial optimization*, Wiley, New York

[27] PARKES, D.C. (2001), Iterative combinatorial auctions: achieving economic and computational efficiency, PhD Dissertation, Department of Computer and Information Science, University of Pennsylvania

[28] PARKES, D.C., CAVALLO, R., ELPRIN, N. et al. (2005), ICE: An iterative combinatorial exchange, Proceedings EC'05

[29] 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

[30] ROTHKOPF, M.H., PEKEČ, A., HARSTAD, R.M. (1998), Computationally manageable combinational auctions, Management Science, Vol. 44, pp. 1131–1147

[31] 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

[32] SCARF, H.E. (1990), Mathematical programming and economic theory, Operations Research, Vol. 38, pp. 377–385

[33] STURMFELS, B. (2004), Algebraic recipes for integer programming, *Proceedings of Symposia in Applied Mathematics*, Vol. 61, pp. 99–113

[34] WILLIAMS, H.P. (1996), Duality in mathematics and linear and integer programming, Journal of Optimization Theory and Applications, Vol. 90, pp. 257–278

[35] WOLFSTETTER, E. (1999), *Topics in microeconomics: auctions, incentives and industrial organization*, Cambridge University Press, Cambridge

[36] WOLSEY, L.A. (1981), Integer programming duality: price functions and sensitivity analysis, Mathematical Programming, Vol. 20, pp. 173–195

[37] XIA, M., KOEHLER, G.J., WHINSTON, A.B. (2004), Pricing combinatorial auctions, European Journal of Operational Research, Vol. 154, pp. 251–270

[38] XIA, M., STALLAERT, J., WHINSTON, A.B. (2005), Solving the combinatorial double auction problem, European Journal of Operational Research, Vol. 164, pp. 239–251