

Drexl, Andreas; Jørnsten, Kurt; Knof, Diether

Working Paper — Digitized Version

Non-linear anonymous pricing combinatorial auctions

Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 625

Provided in Cooperation with:

Christian-Albrechts-University of Kiel, Institute of Business Administration

Suggested Citation: Drexl, Andreas; Jørnsten, Kurt; Knof, Diether (2007) : Non-linear anonymous pricing combinatorial auctions, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 625, Universität Kiel, Institut für Betriebswirtschaftslehre, Kiel

This Version is available at:

<https://hdl.handle.net/10419/147678>

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.

Manuskripte
aus den
Instituten für Betriebswirtschaftslehre
der Universität Kiel

No. 625

**Non-linear anonymous 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.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

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 a linear pricing scheme supporting the optimal allocation might not exist.

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.

The procedure is computationally tested using difficult instances of the combinatorial auctions test suite [16]. The results indicate that the number of extreme prices forming the non-linear anonymous price system is small.

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

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 [18], McAfee and McMillan [21], Milgrom [24]). However, in many auctions/markets a participant's 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) After each iteration 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. These prices are compatible with integer programming duality theory; they truly signal the bidders how to bid in subsequent rounds as opposed to an approximate linear price system generated from using a pseudo-dual approach. Since our computational experience shows that the complexity of the price system does not grow with problem size it is not too complicated for bidders to adjust bids according to our price system as shown by means of an instance below.

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. A review of related work is given in section 3. Section 4 is dedicated to the new pricing model. In section 5 we present computational results based on auctions generated from the combinatorial auctions test suite [16]. Section 6 concludes the paper.

2 Winner determination

2.1 Model

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

$$\left. \begin{aligned} z = \max \quad & \sum_{j \in J} b_j x_j \\ \text{s.t.} \quad & \sum_{j \in J} a_{ij} x_j \leq 1 \quad \forall i \in I \\ & x_j \in \{0, 1\} \quad \forall j \in J \end{aligned} \right\} \quad (1)$$

The parameter b_j is the bid price for bundle j . 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 an integrality gap which is strictly greater than zero, we know from theory (see, e.g., Nemhauser and Wolsey [25]) 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. [32]).

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 [12]). Exact and heuristic algorithms for solving the set packing problem have been developed by, e.g., Borndörfer [3], Delorme et al. [7], Günlük et al. [14] and Sandholm et al. [29]. Special cases of the set packing problem, which can be solved in polynomial time, have been studied in, e.g., Rothkopf et al. [28] and van Hoesel and Müller [15].

A recent survey of combinatorial auctions is provided by de Vries and Vohra [9]. Combinatorial auctions can be useful in many environments and have been considered for problems including selling spectrum rights (McMillan [22], Milgrom [24]), airport take-off & landing time slot allocation (Rassenti et al. [27]), railroad segments (Brewer [4]), and delivery routes (Caplice and Sheffi [5]). Other applications are surveyed in, for instance, Kwon et al. [20]; see also Part V of the book edited by Cramton et al. [6].

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. [27], 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. In the following subsection we will define these requirements formally.

2.2 Basic properties and definitions

First, we define the linear programming relaxation of (1), that is, the problem

$$\left. \begin{aligned} \bar{z} = \max \quad & \sum_{j \in J} b_j x_j \\ \text{s.t.} \quad & \sum_{j \in J} a_{ij} x_j \leq 1 \quad \forall i \in I \\ & x_j \geq 0 \quad \forall j \in J \end{aligned} \right\} \quad (2)$$

and the corresponding dual

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

where $\mathbf{u} = (u_i)$ is the vector of 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 some fundamental properties.¹

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

3 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 specifically dedicated to the set packing problem.

¹Throughout this paper $*$ always denotes an optimal solution.

3.1 Integer programming duality theory

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

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

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

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

where $\mathbf{F} = \{F \in F_+^m : F \text{ is superadditive and } F(0) \geq 0\}$ and F_+^m denotes the set of nondecreasing functions $F : \mathbb{R}^m \rightarrow \mathbb{R}^* = \mathbb{R} \cup \{-\infty, +\infty\}$. 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 [13] dual prices and their relationship to the marginal values of scarce resources have been discussed. Alcaly and Klevorick [1] address the problem encountered with the approach of Gomory and Baumol that “free goods” might have nonzero prices. Wolsey [30] 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.

In particular Wolsey [30] shows that different algorithmic approaches for solving the primal integer program lead to different characterisations of the optimal price functions. Specifically 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

$$F^*(\mathbf{b}) = \max_{t=1}^r [\alpha(t) + \mathbf{u}^t \mathbf{b}] \quad (6)$$

with $\mathbf{u}^t = (u_1^t, \dots, u_m^t) \geq 0, t = 1, \dots, r$, where t indexes the terminal fathomed nodes for some finite value of r . $(\mathbf{u}^t, \underline{\mathbf{u}}^t, \bar{\mathbf{u}}^t) \geq 0$ is the dual feasible solution associated with node t and $\alpha(t) = -\underline{\mathbf{u}}^t \mathbf{g}^t + \bar{\mathbf{u}}^t \mathbf{h}^t$ 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 [17] has developed an algorithm for computing the subadditive dual function which seems to be practical for the set partitioning problem.

3.2 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.

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 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 et al. [27] and by DeMartini et al. [8], among others.

In the following first we sketch the approach by DeMartini et al. [8] and Kwasnica et al. [19] which is one of the most recent models stipulating the use of pseudo-dual prices (see also Bikhchandani and Ostroy [2], Drexl and Jørnsten [10], Parkes [26], Wurmann and Wellman [31] and Xia et al. [32]). Second we outline a recently developed linear programming model the solution of which is "almost always" integral.

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. The main component of the approach by DeMartini et al. [8] and Kwasnica et al. [19] is to solve the linear program (7).

$$\min w \quad (7a)$$

$$\text{s.t. } \sum_{j \in J} a_{ij} u_i + y_j \geq b_j \quad \forall j \in J_0 \quad (7b)$$

$$\sum_{j \in J} a_{ij} u_i = b_j \quad \forall j \in J_1 \quad (7c)$$

$$w \geq y_j \quad \forall j \in J_0 \quad (7d)$$

$$u_i \geq 0 \quad \forall i \in I \quad (7e)$$

$$y_j \geq 0 \quad \forall j \in J_0 \quad (7f)$$

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 (7a) 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 (7a) are not unique a sequence of iterations each of which requires to solve the linear program (7) is performed (for details see DeMartini et al. [8] and Kwasnica et al. [19]).

Recently, Drexl, Jørnsten and Knof [11] developed linear programming models the solution of which is "almost always" is integral. The formulation of the models can be accomplished in two steps. First, again we assume that the integer program (1) has been solved. Then we eliminate all variables corresponding to the chosen items and introduce instead an artificial bid by means of column aggregation. 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 the parameter $\bar{a}_i = \sum_{j \in J_1} a_{ij}$ equals 1; otherwise if $\bar{a}_i = 0$ it is not contained. For the sake of notational brevity we denote the set of items not contained in the aggregated bid with $I_{<}$, that is,

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

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 (8).

$$\max \sum_{j \in J_0} b_j x_j + \bar{b} \cdot z \quad (8a)$$

$$\text{s.t. } \sum_{j \in J_0} a_{ij} x_j + \bar{a}_i z = 1 \quad \forall i \in I \setminus I_{<} \quad (8b)$$

$$\sum_{j \in J_0} a_{ij} x_j = 0 \quad \forall i \in I_{<} \quad (8c)$$

$$x_j \geq 0 \quad \forall j \in J_0 \quad (8d)$$

$$z \geq 0 \quad (8e)$$

The objective function (8a) counts the contribution of the variables contained in J_0 and of the variable z associated with the aggregated bid. Constraint (8b) assures that items $i \in I \setminus I_{<}$

are contained in a winning bundle exactly once. To the contrary, constraint (8c) assures that items $i \in I_<$ are never contained. Finally, (8d) and (8e) define the decision variables to be nonnegative.

In an in-depth computational study model (8) has been shown to very effective in producing integral solutions. For those instances where the outcome is fractional we propose in the following a non-linear pricing scheme.

4 Non-linear pricing

In this section first we provide a mixed-integer programming formulation of the pricing problem. Then we present an algorithm for the computation of a non-linear price function, supported by a small example. Subsequently, we show by means of a tiny example that a non-linear price system does not always exist.

4.1 Model

Since we have separated the solution of the winner determination problem and the pricing problem we know the optimal primal solution (x_j^*) and, hence, the sets J_0 and J_1 . In the following model we try to find a dual solution that fulfills both kinds of complementarities and as many of the dual feasibility constraints as possible by fixing variable $y_j = 1$ with $j \in J_0$ if and only if dual feasibility constraint number j is fulfilled. The constraints of the pricing problem are provided in (9).

$$Y = \left\{ (y, u) \in \mathbb{R}^{|J_0| \times m} : \sum_{i \in I} a_{ij} u_i - b_j y_j \geq 0 \quad \forall j \in J_0 \right. \quad (9a)$$

$$\left. \sum_{i \in I} a_{ij} u_i = b_j \quad \forall j \in J_1 \right\} \quad (9b)$$

$$y_j \in \{0, 1\} \quad \forall j \in J_0 \quad (9c)$$

$$\left. \begin{array}{l} u_i = 0, \text{ if } i \in I_< \\ u_i \geq 0, \text{ otherwise} \end{array} \right\} \quad \forall i \in I \quad (9d)$$

Constraints (9a) in conjunction with the second branch of (9d) address dual feasibility. Constraints (9b) assure dual complementary slackness. Constraint (9c) requires the variables corresponding to the losing bids to be binary. Finally, the first branch of (9d) addresses primal complementary slackness. Hence, overall the constraints assure that any feasible solution of the mixed-integer pricing model has the properties defined in section 2.2.

Note that both constraints (9a) and (9b) originate from the same (in-)equality. In particular, (9b) 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:

$$\max \left\{ \sum_{j \in J_0} y_j : (y, u) \in Y \right\} \quad (10)$$

The objective function of (10) aims at maximizing the number of losing bids being covered in the optimal solution. We will show now that this objective is the primary choice. An important property of dual prices is to assure that as many losing 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 proposition states that the objective of our pricing model takes care of this characteristic.

Proposition 1 *The mixed-integer pricing model prices out as many losing bids as possible.*

Remark 1 *The objective of (10) 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.*

4.2 Algorithms

Solving model (10) produces one price vector that assures that the prices for the winning bids are equal to the prices bid by those bidders. We will provide a means to compute an alternative price vector that maximizes the number of losing bids that satisfy dual feasibility, and primal complementary slackness. This process of calculating price vectors that maximize the number of losing bids not yet satisfied in previous iterations will be continued until we have a complete set of price vectors that price out correctly (i.e. satisfy both dual feasibility and primal complementary slackness) of all non-losing bids. Two algorithms which serve this purpose will be described in the sequel.

First, other prices can be computed by adding cover cuts. Assume we know the optimal binary variables (y_j^*) . Let $K_0 := \{j \in J_0 : y_j^* = 0\}$ denote the subset of losing bids. If we define

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

and solve $\max\{\sum_{j \in J_0} y_j : (y, u) \in Y'\}$ instead of (10) we get another vector y (because the cut $\sum_{j \in K_0} y_j \geq 1$ excludes the previous one) and another price vector u . Afterwards we update the sets K_0 and $Y' := \{(y, u) \in Y' : \sum_{j \in K_0} y_j \geq 1\}$ and reoptimize $\max\{\sum_{j \in J_0} y_j : (y, u) \in Y'\}$ (taking the new set Y') accordingly, and again get some other vectors (y, u) . Apparently, iterating this way produces a sequence of up to ρ prices, where ρ may be exponentially large.

The following proposition defines equation (6) for the special case considered here highlighting the fact that the procedure generates a non-linear anonymous price system.

Proposition 2 *The price system has the form*

$$F^*(d) = \max_{t=1}^{\rho} [u^t d].$$

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

In practice agents will only be able to handle a small number of prices and, hence, we propose a second algorithm which iterates as follows. Let $t = 2, 3, \dots$ denote the iteration counter. Initialize sets $K_0^1 = \{j \in J_0 : y_j^* = 0\}$ and $K_1^1 = \{j \in J_0 : y_j^* = 1\}$ in $t = 1$. Now the update

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

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

performed in iteration t indicates that we move bids that are priced out in iteration $t - 1$ from the set K_0^{t-1} to the set K_1^t . In iteration t we solve the optimization problem

$$\max \left\{ \epsilon \sum_{j \in K_1^t} y_j + \sum_{j \in K_0^t} y_j : (y, u) \in Y \right\} \quad (11)$$

where ϵ must be "small", that is, e.g., $\frac{1}{|J_0|}$. The objective function of (11) lexicographically searches for alternative optimal solutions in which bids losing in the previous iteration become winning bids in the current one. The iteration terminates once we have $|K_1^t| = |K_1^{t-1}|$.

We will explain the algorithm, in particular the objective of (11), in more detail. Let denote $\alpha := \epsilon \sum_{j \in K_1^{t-1}} y_j$ the first part and $\beta := \sum_{j \in K_0^{t-1}} y_j$ the second part of the objective, respectively. By definition the cardinality of K_0^1 initially is less than or equal to the cardinality of J_0 . Moreover, we transfer winning bids from the set K_0^{t-1} to the set K_1^t . While bids covered by K_1^{t-1} are weighted with ϵ bids contained in K_0^{t-1} are weighted with 1. Hence, α is always less than or equal to β and α tends to increase during the course of the algorithm but it never exceeds 1 due to the fact that ϵ is chosen small enough. The following proposition sheds light on the convergence properties of the algorithm.

Proposition 3 *The algorithm generates a finite number $q \leq |J_0| + 1$ of prices.*

Proof. Without loss of generality consider iteration t . First, we analyze the two extreme cases that either (a) $y_j^* = 1$ for all $j \in K_0^{t-1}$ or that (b) $y_j^* = 0$ for all $j \in K_0^{t-1}$. In case (a) all bids $j \in K_0^{t-1}$ are transferred to K_1^t and, hence, the algorithm terminates in iteration $t + 1$ due to $K_1^t = K_1^{t+1} = \emptyset$. In case (b) the algorithm terminates immediately due to $|K_1^{t-1}| = |K_1^t|$. In intermediate cases at least one bid $j \in K_0^{t-1}$ is selected, transferred to K_1^t and, hence, the cardinality of K_1^t increases. Apparently, the transfer of bids from K_0^t to K_1^t can be achieved at most $|J_0|$ times and, hence, $q \leq |J_0| + 1$ prices will be generated including the initial price generated when solving (10). \square

The proposition verifies that the algorithm does not cycle, although variables at 1 are not forced to stay 1 in the next iteration. Moreover, the number of prices does not grow exponentially. More important from a practical point of view, we will show by means of a computational study in section 5 that the price system is fairly small, that is, $q \ll |J_0|$.

4.3 Example

An example with 6 items and 21 bids taken from Parkes [26] illustrates the idea. Obviously, here we have a case with three agents each of which bids on every combination of items. The

| | | | | | | | | | | | | | | | | | | | | | |
|----------|----|----|----|-----|-----|-----|-----|----|----|----|-----|-----|-----|-----|----|----|----|-----|-----|-----|-----|
| 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 |

Table 1: Instance – Parkes [26]

| 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

| t | y^t | u^t |
|-----|--------------|------------------------|
| 1 | $y_{12} = 0$ | (60, 140, 75, 0, 0, 0) |
| 2 | $y_{20} = 0$ | (140, 60, 75, 0, 0, 0) |

Table 3: Bids selected and price system

bid prices (b_j) and the coefficient matrix (a_{ij}) are provided in Table 1. Note that the bids 4, 12, 14 and 20 are superadditive.

Table 2 provides the solution (x_j^*) of the integer program (1) and the solution (\bar{x}_j^*) of the linear programming relaxation (2). Variables not given there have value 0. OFV abbreviates optimal objective function value. Our pricing model generates the $q = 2$ vectors y^t and u^t ($t = 1, 2$) displayed in table 3 (variables y_j not given explicitly have value 1).

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 in order to let these bids potentially win in the next round of an iterative combinatorial auction.

| 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

The reduced cost obtained for bids 12 and 20 show the amount that these bidders have to raise their bids in order to generate alternative optimal solutions. This means that if we increase b_{12} to $215 + \epsilon$ then we get the optimal solution $x_1 = x_{12} = 1$ with OFV $275 + \epsilon$. Alternatively if we instead increase b_{20} to $215 + \epsilon$ we get $x_9 = x_{20} = 1$ with the same OFV. This result is due to the fact that our price system is based on integer programming duality theory and it could never have been achieved by using approximate linear prices such as the ones generated by using pseudo-duals.

4.4 Does a non-linear price system always exist?

In section 5 we will show by means of a computational study that our algorithm is effective and efficient in computing a non-linear price system. From a theoretical point of view, however, the question is whether a non-linear price system does always exist or not. A tiny counterexample with 3 items and 5 bids provided in table 5 sheds light on this issue.

| | | | | | | | | | | | | | | | |
|------|----|-------|---|-------|-------|-------|----|-------|---|----|-------|---|-------------------|-------|----------------|
| max | 10 | x_1 | + | 10 | x_2 | + | 10 | x_3 | + | 25 | x_4 | + | 25 | x_5 | |
| s.t. | | x_1 | | | | | | | + | | x_4 | | | | ≤ 1 |
| | | | | x_2 | | | | | + | | x_4 | + | | x_5 | ≤ 1 |
| | | | | | | x_3 | | | | | | + | | x_5 | ≤ 1 |
| | | | | | | | | | | | | | x_1, \dots, x_5 | | $\in \{0, 1\}$ |

Table 5: Counterexample

The optimal integer solution is $(x_j) = (1, 1, 1, 0, 0)$ with OFV 30. The optimal solution of the linear programming relaxation is $(\bar{x}_j) = (\frac{1}{2}, 0, \frac{1}{2}, \frac{1}{2}, \frac{1}{2})$ with OFV 35. Now in order to have dual complementary slackness the price vector must be $(u_i) = (10, 10, 10)$ and the variables x_4 and x_5 can not be priced out.

The instance shows that there is no guarantee that a non-linear price system always exists. The computational results presented in the following section, however, will show that larger more realistic instances do have enough dual degeneracy such that a non-linear price system exists.

5 Computational results

The methods described earlier have been implemented using the Cplex callable library (version 9.0) on an AMD Athlon with 1 GB RAM and 2.1 Ghz clockpulse running under the operating system Linux.

The primary purpose of the computational experiments is to show that the price system covers all losing bids. 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 [16]) 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.

We present numerical results only for the subset of instances which have recently been shown to be intractable by the linear pricing model (8); see Drexler, Jørnsten and Knof [11]. Fortunately, for all instances of the distributions types arbitrary, matching, paths and regions the linear pricing model (8) produced an integral solution. Although we let run our non-linear pricing algorithms also on these instances we restrain from detailing the results here. For the sake of shortness detailed results are shown only for those difficult instances of the distribution type scheduling, for which model (8) did not produce an integral solution.

The computational results are displayed in table 6. Column 1 shows the number of items m , column 2 the number of bids n and column 3 the unique identifier "id" of the 5 particular difficult pricing instances examined.² Column 4 provides the number of prices q calculated. Column 5 shows the CPU time in milliseconds required for solving the mixed-integer programs (10) and (11).³ The last column shows whether all losing bids are covered in at least one of the price vectors ("Y") or not ("N").

²The instances and the corresponding solutions of the pricing model can be obtained from the authors upon request.

³The time needed in order to solve the original combinatorial auction (1) is not included.

| m | n | id | q | msec | all losing bids covered |
|-----|-----|-----|-----|------|-------------------------|
| 30 | 120 | 034 | 3 | 100 | Y |
| 40 | 200 | 001 | 2 | 40 | Y |
| 40 | 200 | 007 | 2 | 40 | Y |
| 40 | 200 | 052 | 2 | 40 | Y |
| 60 | 240 | 038 | 3 | 3790 | Y |

Table 6: Results

The results show that all losing bids contribute to the computation of the price system. Furthermore, CPU times are very small.

The results obtained for all the instances of the various distribution types can be summarized as follows:⁴

- The instances of the distribution type *arbitrary* are hard in the sense that the mixed-integer pricing problems are difficult to solve. The minimum number of prices is 3 and the maximum number of prices is 5.
- Instances of the distribution type *matching* are easy because the mixed-integer pricing problems can be solved quickly. More 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. The number of prices which have to be generated is very small, too.
- For the distribution type *regions* there is greater variety. Some mixed-integer programs are challenging from a computational point of view. 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. Again, one instance requires excessive CPU time.

Summarizing, the computational tests show that the number of prices needed does not grow with increasing problem size.

6 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. By separating the pricing problem from the winner determination problem we are able to generate a fairly simple non-linear anonymous price system compatible with integer programming duality theory. 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.

⁴Detailed results can be obtained from the authors upon request, too.

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] CRAMTON, P., SHOHAM, Y., STEINBERG, R., eds., (2006), *Combinatorial auctions*, The MIT Press, Cambridge
- [7] DELORME, X., GANDIBLEUX, X., RODRIGUEZ, J. (2004), GRASP for set packing problems, *European Journal of Operational Research*, Vol. 153, pp. 564–580
- [8] 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
- [9] DE VRIES, S., VOHRA, R. (2003), Combinatorial auctions: a survey, *INFORMS Journal on Computing*, Vol. 15, pp. 284–309
- [10] DREXL, A., JØRNSTEN, K. (2005), Reflections about pseudo-dual prices in combinatorial auctions, *Journal of the Operational Research Society* (to appear)
- [11] DREXL, A., JØRNSTEN, K., KNOF, D. (2007), Column aggregation-based pricing combinatorial auctions, Working Paper, University of Kiel
- [12] GAREY, M.R., JOHNSON, D.S. (1979), *Computers and intractability: a guide to the theory of NP-completeness*, Freeman, San Francisco, CA
- [13] GOMORY, R.E., BAUMOL, W.J. (1960), Integer programming and pricing, *Econometrica*, Vol. 28, pp. 521–550
- [14] 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
- [15] VAN HOESEL, S., MÜLLER, R. (2001), Optimization in electronic markets: examples in combinatorial auctions, *Netnomics*, Vol. 3, pp.23–33
- [16] <http://robotics.stanford.edu/CATS>
- [17] KLABJAN, D. (2003), A practical algorithm for computing a subadditive dual function for set partitioning, Technical Report, University of Illinois at Urbana-Champaign

- [18] KLEMPERER, P. (2002), What really matters in auction design, *Journal of Economic Perspectives*, Vol. 16, pp. 169–189
- [19] 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
- [20] KWON, R.H., ANANDALINGAM, G., UNGAR, L.H. (2005), Iterative combinatorial auctions with bidder-determined combinations, *Management Science*, Vol. 51, pp. 407–418
- [21] MCAFEE, R.P., MCMILLAN, J. (1987), Auctions and bidding, *Journal of Economic Literature*, Vol. 25, pp. 699–738
- [22] MCMILLAN, J. (1994), Selling spectrum rights, *Journal of Economic Perspectives*, Vol. 8, pp. 145–162
- [23] MILGROM, P. (1989), Auctions and bidding: a primer, *Journal of Economic Perspectives*, Vol. 3, pp. 3–22
- [24] MILGROM, P. (2000), Putting auction theory to work: the simultaneous ascending auction, *Journal of Political Economy*, Vol. 108, pp. 245–272
- [25] NEMHAUSER, G.L., WOLSEY, L.A. (1988), *Integer and combinatorial optimization*, Wiley, New York
- [26] PARKES, D.C. (2001), Iterative combinatorial auctions: achieving economic and computational efficiency, PhD Dissertation, Department of Computer and Information Science, University of Pennsylvania
- [27] 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
- [28] ROTHKOPF, M.H., PEKEČ, A., HARSTAD, R.M. (1998), Computationally manageable combinational auctions, *Management Science*, Vol. 44, pp. 1131–1147
- [29] 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
- [30] WOLSEY, L.A. (1981), Integer programming duality: price functions and sensitivity analysis, *Mathematical Programming*, Vol. 20, pp. 173–195
- [31] WURMANN, P.R., WELLMAN, M.P. (2000), AkBA: A progressive, anonymous-price combinatorial auction. In *Second Conference on Electronic Commerce*, pp. 21–29
- [32] XIA, M., KOEHLER, G.J., WHINSTON, A.B. (2004), Pricing combinatorial auctions, *European Journal of Operational Research*, Vol. 154, pp. 251–270