IBM Research Report RC 22478

BIDDING LANGUAGES AND WINNER DETERMINATION IN MULTI-ATTRIBUTE AUCTIONS

Martin Bichler and Jayant Kalagnanam

IBM T. J. Watson Research Center
P.O. Box 218, Yorktown Heights, NY 10533
{bichler, jayant}@us.ibm.com

ABSTRACT

The theory of procurement auctions traditionally assumes that the offered quantity and quality is fixed prior to source selection. Multi-attribute reverse auctions allow negotiation over price and qualitative attributes such as color, weight, or delivery time. They promise higher market efficiency through a more effective information exchange of buyer's preferences and supplier's offerings. This paper focuses on a number of winner determination problems in multi-attribute auctions. Previous work assumes that multi-attribute bids are described as attribute value pairs and that the entire demand is purchased from a single supplier. Our contribution is twofold: First, we will analyze the winner determination problem in case of multiple sourcing. Second, we will extend the concept of multi-attribute auctions to allow for configurable offers. Configurable offers enable suppliers to specify multiple values and price markups for each attribute. In addition, suppliers can define configuration and discount rules in form of propositional logic statements. These extensions provide suppliers with more flexibility in the specification of their bids and allow for an efficient information exchange among market participants. We will present MIP formulations for the resulting allocation problems and an implementation.

1 INTRODUCTION

Procurement auctions usually require the bid to specify several characteristics of the contract to be fulfilled. Previous models of procurement auctions have generally assumed that the qualitative attributes are fixed prior to competitive source selection - hence bidding competition is restricted to the price dimension (see [1] or [2]). While such an approach may be appropriate for auctions of homogeneous goods, most procurement includes heterogeneous offerings of suppliers. Traditionally, these types of negotiations are resolved through bilateral bargaining or sealed-bid tenders, where a buyer asks for bids in unstructured or semi-structured format and then the buyer selects one or more of these bids manually. A tendering procedure allows the sale to be determined by a variety of attributes involving not only price but quality, lead time, contract terms, and supplier reputation.

Recently, multi-attribute reverse auctions have become a popular means of automating this process further. The negotiable attributes are defined in advance, and suppliers can compete either in an open-cry or sealed-bid fashion on multiple attributes. This process allows more degrees of freedom for suppliers in formulating their bids, while at the same time it leverages the competitive forces of an auction to drive the negotiation to an equilibrium. Expected gains of multi-attribute auctions are increased speed of the negotiation, higher market transparency, as well as higher degrees of allocative efficiency.

Although the literature in this field is fairly young, a number of procurement departments and software vendors have embraced the idea[1]. Companies such as eBreviate or PurchasePro have implemented what is also called the "total cost approach". Here buyers specify monetary values (discounts and/or mark-ups) for attribute values, in order to be able to compare different offerings. Another approach uses decision analysis techniques [3] to assign weights and individual value functions to the relevant attributes, and calculate a value score. Bidders can then compete on this value score by improving one or more of the attributes. This approach is used by software vendors such as Clarus, IBM/DigitalUnion, Moai, Menerva, and Perfect. TIScover, an Austrian destination management system, is using this type of multi-attribute auction to match tourists and hoteliers on an accommodation market [4]. Most of these software packages have been developed during the past three years.

The existing game-theoretic literature [5-7] typically assumes quasi-linearity of buyers' scoring functions as well as suppliers' cost functions to analyze the strategic issues as well as efficiency of multi-attribute auctions. This generic format covers a broad variety of functional forms including linear additive functions, which have found its way into most commercial packages as a technique for modeling multi-attribute scoring functions. Two fundamental assumptions underlie the treatment of multi-attribute bids in the literature and in commercial software packages:

1. The bids are point bids and are specified as attribute value pairs for each of the attributes, and
2. The multi-attribute auctions usually assume that the contracts are awarded to a single bid – we call this the *sole souring* assumption.

The restriction to such simple multi-attribute bids in previous approaches is due to technical difficulties of specifying more complex offerings. The focus on sole sourcing is based on the argument that multi-attribute auctions are mainly used for contracts with high asset-specificity, such as Department of Defense (DoD) contracts, which were also the focus of initial economic analysis of multi-attribute auctions [5]. With the wide-spread use of multi-attribute auctions in part because of readily available software to support such formats, companies are using multi-attribute auctions also for the procurement of large

[1] Often also called mutivariate RFQ or multidimensional auction.

amounts of less specific goods (i.e. MRO procurement) where multiple sourcing becomes important. For example, in a recent procurement auction run by internal procurement at IBM for a large quantity of chairs for one of their office buildings multiple sourcing was entertained provided some conditions across the supply pool was satisfied.

In this paper we examine two extensions to the current formats used for multi-attribute auctions. First, we will analyze the winner determination problem in case of *multiple sourcing*. In this setting we examine the impact of several business rules that need to be imposed on the winner determination problem in order to obtain an acceptable supply from multiple suppliers. Second, we will extend multi-attribute auctions to allow for configurability in bids. In contrast to traditional multi-attribute offers, configurable offers enable suppliers to specify multiple values and price markups for each attribute. The facility of providing configurable offers introduces a problem of informational complexity since the price function (over discrete attributes) now needs to be specified over an exponential number of combinations.

We restrict our attention to a special case where the price dependence on a attribute is specified as a markup over a base price thereby restricting the price function to an additive form. This appears to be sufficient for many real world settings such as PC, logistics etc. In addition, we allow suppliers to specify constraints that restrict the set of available configurations or alternately allow the specification of discounts based on levels of multiple attributes. In general, in practice we encounter only a small number of such higher order terms and hence they can be managed quite effectively. Moreover, such configuration rules and discounts can be adequately represented using propositional logic. The advantage of such a restricted representation for configurable bids is that propositional logic can also be represented by linear inequalities which can then be added to the winner determination problems. The extension of these multi-attribute allocation problems has been motivated by our work on a large-scale procurement marketplace for the retail industry and our experience with internal procurement auctions.

In the next section we will provide a brief introduction into the literature of multi-attribute auctions. In section 3 we will describe the standard bid evaluation technique in multi-attribute auctions, namely the additive scoring function and elaborate on the issue of preference elicitation in multi-attribute auction. In section 4 we will analyze multi-attribute winner determination in the presence of multiple sourcing. Section 5 will introduce the concept of configurable offers and formulate the associated allocation problems. Finally, in section 5.6 we will summarize the article and provide an overview of a Java object framework implementing a variety of winner determination algorithms for electronic markets. The framework is currently used in a commercial implementation and can be used as a standard solver component in electronic procurement applications.

2 REVIEW OF THE LITERATURE

Only a small but steadily growing number of academic papers have considered multi-attribute auctions so far. A thorough analysis of the design of multi-attribute auctions has been provided by Che [5]. He derived a two-dimensional version of the revenue equivalence theorem [8]. Che also designs an optimal scoring rule based on the assumption that the buyer knows the probability distribution of the supplier's cost parameter. Branco's analysis is based on Che's *independent cost model* and derives an optimal auction mechanism for the case when the bidding firms' costs are correlated [6]. Bichler [9] and Bichler and Kaukal [10] show some first Internet-based implementations of the concept and discuss MAUT as an algorithm for bid evaluation in single-sourcing, multi-attribute reverse auctions. A variety of different multiple issue auction algorithms are suggested by Teich, Wallenius and Wallenius [11]. Multi-attribute English auctions have also been analyzed in the context of service allocation amongst artificial agents [12].

One direction in auction design is concerned with *efficient* mechanisms. Here, the primary objective of the planner is to maximize allocative efficiency. Milgrom provides proofs that allocative efficiency is achieved in a single-sourcing multi-attribute auction if the auctioneer announces his true utility function as the scoring rule, and conducts a Vickrey auction based on the resulting scores [7]. Some first laboratory experiments on the efficiency of multi-attribute auctions have been described in [13]. In these experiments and the efficiency of price-only and multi-attribute auctions did not show significant differences. In addition, the value scores in multi-attribute auctions were significantly higher than in price-only auctions. In recent paper, Parkes and Kalagnanam [14] provide an iterative multi-attribute auction design based on a primal-dual algorithm. They show that this design is incentive compatible for the sellers and Pareto efficient with truthful buyers.

Efficient auction design is concerned with how the surplus in an auction is divided among the bidders and the auctioneer, while *optimal auction design* concentrates on designs, which maximize the expected revenue of the bid taker. In multi-attribute auctions it is appropriate to speak about buyer *utility maximization* instead of revenue-maximization. As already suggested in Che's analysis, the optimal scoring rule in a multi-attribute reverse auction may not be identical to the buyer's true value function. Beil and Wein [15] focus on buyer utility-maximization in multi-attribute auctions, i.e. optimal auction design. The paper suggests an inverse-optimization based approach that allows the buyer via several changes in the announced scoring rule, to learn the suppliers' cost functions and then determine a scoring rule that maximizes the buyer's utility within an open-ascending auction format.

In this paper we focus on winner determination problems in multi-attribute auctions with multiple sourcing and configurable offers. The basic multi-attribute auction design on which our analysis is based on (see next section) can be found in practice, where it is often used to further automate existing RFQ processes. We do not make statements on the optimality of a scoring function, incentive compatibility of the payment schema or other mechanism design issues, but take the scoring function as given by the buyer and focus on the resulting allocation problems. The proposed optimization models are designed to maximize the buyer's utility.

3 SOLE SOURCING

A popular approach to implement multi-attribute auctions is based on traditional decision analysis techniques. Here bidders submit bids as attribute-value pairs, which are evaluated by a value or scoring function provided by the buyer. In this section we will briefly describe this multi-attribute auction format and discuss some of the limitations of this approach. We first restrict our attention to the single sourcing problem, where the entire demand is purchased from a single supplier.

3.1 The Standard Additive Scoring Function

In practical implementations, the elicitation of a buyer's preferences, and consequently the construction of an appropriate scoring function is of pivotal importance. A common approach is based on the use of established decision analysis techniques, such as MAUT [3], SMART [16] or AHP [17]. Although advanced versions of MAUT and AHP can model interactions among attributes, the basic techniques use a linear, weighted value function, which assumes *preferential independence* of all attributes. An attribute x is said to be preferentially independent of y if preferences for specific outcomes of x do not depend on the value of attribute y [18].

We next introduce some terminology and notation. Consider I bids (or offers) and J attributes. Each attribute $j \in J$ has an attribute space of K_j. A multi-attribute offer, received by the buyer, can then be described as an *n*-dimensional vector $\boldsymbol{v}_i. = (v_{i1} \ldots v_{iJ})$ where v_{ij} is the level of attribute *j*. In the case of an additive scoring function $S(\boldsymbol{v}_i)$ the buyer evaluates each relevant attribute v_{ij} through a scoring function $S_j(v_{ij})$. The overall value $S(\boldsymbol{v}_i)$ for a bid $\boldsymbol{v}_i$ is given by the sum of all individual scorings of the attributes. It is convenient to scale s_i and each of the single-attribute utility functions $S_j(.)$ from zero to one. That is, for a bid $\boldsymbol{v}_i$ and a scoring function that has weights $w_1 \ldots w_J$, the overall utility for a bid is given by

$$s_i = S(\mathbf{v}_i) = \sum_{j \in J} w_j S_j(v_{ij}) \ \ and \ \sum_{j \in J} w_j = 1 \tag{1}$$

The problem a buyer faces is to determine appropriate $S_j(.)$ functions and w_j weights. An *optimal auction* is allocating the deal to the suppliers in a way that maximizes the utility for the buyer, i.e. to the supplier providing the bid with the highest overall utility score for the buyer. The function *max* s_i with *1 £ i £ I* gives us the utility score of the winning bid and can be determined through open-cry or sealed-bid auction schemes.

3.2 Preference Elicitation

The assessment of appropriate weights w_j is key to MAUT and is an important aspect of a “good” preference model. Several techniques have been proposed in the traditional decision analysis literature to help users assign reasonable weights. One approach is called *pricing out* because it involves determining the value of one objective in terms of another (e.g. dollars). For example, one might say that 5 days faster delivery time is worth $400. The idea is to find the indifference point, i.e. determine the marginal rate of substitution between two attributes. Although this concept seems straightforward, it can be a difficult assessment to make.

Since many decision makers feel unable to provide exact weights, some of the more recent approaches only ask for uncertain estimates. For example, methods from fuzzy decision analysis use fuzzy sets for weights and individual scoring functions and fuzzy operators for the aggregation of those fuzzy sets [19]. AHP uses a different approach to weight determination. A principle used in AHP is that comparative judgments are applied to construct a symmetric matrix of pair-wise comparisons of all combinations of attributes. The method is based on the mathematical structure of consistent matrices and their associated right-eigenvector’s ability to generate true or approximate weights. The right eigenvector of the matrix results in the weights for the different objectives. More recent approaches try to estimate the buyers preferences based on comparisons of alternatives [20, 21]. These techniques assume weaker decision makers and do not ask for attribute-level utility assessments.

The previous discussion assumes linearity of the buyer’s value function. Interaction effects among attributes although relevant in many considerations are often neglected in real-world implementations. Preferential dependencies impact the shape of the utility function and require the modeling of non-linear utility functions. Two attributes may to some extent be substitutes or may complement each another. In certain applications, one might even argue that price is preferentially dependent on qualitative attributes. For example, having the choice between a luxury and low-budget cars, the importance of price might depend on the type of the car evaluated. In order to express these interdependencies among two attributes, an additive utility function can be extended towards a so-called multilinear expression [3]. In general, however, the notion of interaction among attributes is one of the most difficult concepts in multi-criteria decision making and is certainly an important issue to consider in the design of multi-attribute auctions.

4 MULTIPLE SOURCING

Initially multi-attribute auctions have been analyzed in the context of DoD purchasing activities with high asset specificity, such as weapon systems. With their more wide-spread use companies start using multi-attribute auctions even for less specific products such as MRO equipment, where quality plays a role. In these situations, buyers need to buy larger quantities and are willing to purchase from multiple suppliers.

4.1 Allocation to Multiple Suppliers

A computationally simple case assumes submission of divisible bids, i.e. that the bidders are willing to accept partial quantities of their bids for the same unit price. Similar to section 2, the auctioneer can use a scoring function to sort the bids by descending score per unit. Since bids can be divided into smaller quantities winning bids are the ones where $\Sigma_{i \text{Î} I}\, q_i \;\cdot D$ with q_i being the quantity provided in bid i and D being the demand. However, this is only applicable in a limited number of real-world scenarios.

A more realistic assumption is *indivisible* bids. We consider a computer vendor who wants to buy 2000 hard disks on a private exchange. Suppose we have decreasing average production costs, i.e. suppliers encounter economies of scale and the unit price of a hard disk is bound to the quantity sold. Again, the auctioneer tries to satisfy the buyer's quantity at the lowest cost. The indivisibility assumption turns the problem into a computationally hard problem, which cannot be solved by sorting of bid scores. To illustrate this consider the following simple example with 4 bids and a demand of 40.

Bid No.	Quantity	Score/Unit	Overall Score
1	10	31	310
2	40	25	1000
3	30	24	720
4	30	23	690

Table 1: Example with Indivisible Bids

There are several possible solutions sets to satisfy the buyer's demand: {2}, {1, 3}, or {1, 4}. The optimal solution (overall score = 1030) is provided by bid set {1, 3}. Selecting the optimal set of bids is related to the well known 0-1 knapsack problem, which is known to be NP-hard, but can be solved efficiently in practice by using dynamic programming [22]. Since, in practice, the combinations hardly ever sum up to exactly the demand specified by the buyer, we use the following integer program (IP) formulation with acceptable lower and upper bounds for the demand (D_{min} and D_{max}).

$$\max \sum_i \sum_l (q_{il} s_{il})\, r_{il} \tag{2}$$

subject to

$$D_{\min} \le \sum_i \sum_l q_{il}\, r_{il} \le D_{\max} \tag{3}$$

$$\sum_i \sum_l q_{il}\, p_{il}\, r_{il} \le C \tag{4}$$

$$\sum_i r_{il} \le 1 \;\; \forall l \in L \tag{5}$$

$$r_{il} \in \{0,1\} \;\; \forall i \in I, \forall l \in L \tag{6}$$

The objective in this optimization is to maximize the overall score (2), where s_{il} is the unit score of a bid i from supplier l 2 L and q_{il} is the quantity of bid i, so that the supplied quantity satisfies the lower and upper bound for the demand (3). The overall reservation price C is considered in (4) where p_{il} is the unit price of bid i from supplier l. Constraint (5) ensures that only one of the bids of a supplier is selected. We introduce the binary decision variables r_{il} to indicate the bids selected by the buyer (6).

In real-world settings there are several considerations besides cost minimization. These considerations are specified as a set of constraints that need to be satisfied while selecting a set of winning bids. We discuss two such rules, which have shown to be relevant in practical applications.

4.2 Number of Winning Bidders

An important multiple sourcing consideration is the number of winners. On the one hand, buyers want to make sure that the entire supply is not sourced from too few suppliers, since this creates a high exposure if some of them are not able to deliver on their promise. On the other hand, having too many suppliers creates a high overhead cost in terms of managing a large number of supplier relationships. These considerations introduce constraints on the minimum, L_{min}, and maximum, L_{max}, number of winning suppliers in the solution to the winner determination problem.

$$0.1 y_l \le \sum_i r_{il} \le e y_l \ \forall i \in I \tag{7}$$

$$L_{\min} \le \sum_l y_l \le L_{\max} \tag{8}$$

$$y_l \in \{0,1\} \ \forall l \in L \tag{9}$$

We introduce an indicator variable y_l for each supplier l, which takes the value 1 if the supplier has any winning bids and 0 otherwise. The first constraint (7) sets y_l to 1 if supplier l2L has any winning bids. Note that the constant multiplier C, the number of a supplier's bids, ensures that the right hand side is large enough when more than one bid of supplier l is selected.

4.3 Homogeneity of the Purchase

A basic problem, which arises from multi-attribute auctions in the case of multiple sourcing is the heterogeneity of goods purchased from different suppliers. The solution, which maximizes the overall score, might be one where the bids of all winning suppliers have different values in all attributes. While this is not necessarily a problem for all attributes, it can be important in certain applications to enforce homogeneity of a certain attribute in the set of winning bids. In order to capture such constraints we introduce an indicator variable z_{jk} that assumes the value 1 if any suppliers are chosen with a bid at level k for attribute j. T_{kj} is defined as the set of bids at level K for attribute J. This is formalized as follows:

$$0.1 z_{jk} \le \sum_{i,l \in T_{jk}} r_{il} \le \|T_{jk}\| z_{jk} \forall j \in J \tag{10}$$

$$\sum_k z_{jk} \le 1 \ \forall j \in J \tag{11}$$

$$z_{jk} \in \{0,1\} \ \forall j \in J, \forall k \in K \tag{12}$$

Notice that this constraint is for a given attribute of interest (such as color) at some level (say red). For each attribute, this formalism introduces as many constraints as there are levels and one extra constraint to enforce homogeneity.

4.4 Computational Issues

A dynamic programming approach can be used to solve the above problems. We found that commercial integer programming software using a branch-and-bound approach was able to solve problems of 200 bids and 20 attributes on the order of a few seconds. As expected, the consideration of additional constraints described in section 4.2 and 4.3 impacts the runtime of the program.

Figure 1 presents CPU time results (in seconds) for solving a randomly generated instance of the multi-attribute auction. This problem had fixed problem size of 10 attributes and 100 bids. We set the minimum number of winning suppliers to be equal to the maximum number of winning suppliers, which we varied between 1 and 20. The optimal allocation for this problem without any side constraints on the number of winning suppliers had five winning suppliers. For a very large number of winning suppliers the problem becomes easier to solve, since it may not be possible to find an allocation which satisfies the total demand of the buyer. A constraint on only the maximum number of winners did not have a significant impact on the runtime. So the major influencing factor for the exponential runtime was the constraint on the minimum number of winners.

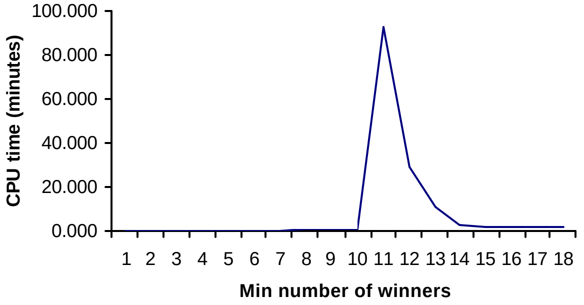

Figure 1: Experimental results for a multi-attribute auction, varying the number of allowed winning suppliers constraint

Figure 2 shows the CPU time results for solving a problem with a fixed problem size of 100 bids, 20 attributes and an increasing number of homogeneity constraints. The impact of adding these constraints on the runtime is close to linear. With low correlation of the bid data and a high number of possible attribute levels, this constraint can often not be satisfied and leads to infeasibility. We recommend enforcing the homogeneity constraint on as few attributes as possible in an auction with multi-attribute bids. Homogeneity constraints are very applicable to the evaluation of configurable offers discussed in the next section.

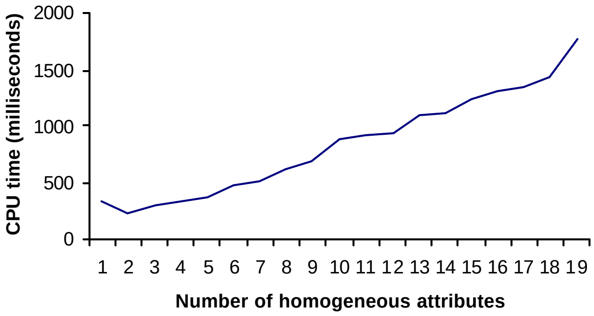

Figure 2: Experimental results for a multi-attribute auction, varying the number of homogeneity constraints

5 ALLOCATION OF CONFIGURABLE OFFERS

A basic assumption in our previous discussion has been that bids are described as *sets of attribute-value pairs*. In practice, however, many offers are specified as configurations, where each attribute can take a number of different attribute values. Lets assume a PC has only three attributes, namely processor speed, hard disk size, and price. A supplier could specify that there are three processors available {850MHz, 950 MHz, and 1GHz}, as well as two sizes of hard disks {10GB and 15GB}. The base configuration (850MHz, 10GB) prices for $1000. A configuration with a 1GHz computer is $100 more, and one with a 15GB hard disk costs an additional $200. Most services such as insurances or transportation can be considered as *configurable offers* in a similar way. The ability to express options in multi-attribute auctions is a crucial feature to further automate negotiations on complex goods and services.

Configurable multi-attribute offers exhibit combinatorial features. With only 10 attributes and 5 possible attribute values for each of them, there are already 5^{10} = *9.7 million* possible configurations. Clearly, finding the best configuration among all these is not an easy task. Enumerating all possible configurations is mostly not a viable alternative, since a large amount of individual bids would have to be generated and communicated to the buyer. Therefore, suppliers often restrict the number of configurations they offer to only a small selection. As a result, buyers might not find the best configuration and choose the offer of another supplier, i.e. the situation might lead to inefficient outcomes. It is in the interest of both, buyers and suppliers, to communicate offers in a compact that cover a large space of possible product configurations. In the following we will propose a procedure to describe configurable offers and determine the best individual configuration based on a buyers' scoring function.

5.1 Functional Description of Configurable Offers

There are many possible ways how multi-attribute offers can be made configurable. In this approach we allow suppliers to specify the possible values for each attribute in an offer in a functional format. In other words, we describe the possible configurations of a configurable offer as a function of price on quantity and qualitative attributes. Assuming additivity of the attributes, the total price p_i for a particular bid/offer i can be written as:

$$p_i = q_i p_{bi}(q_i) + q_i \sum_j f_{ij}(v_{ij}) \tag{13}$$

where q_i is quantity, $p_{bi}(q_i)$ is the base price per item as a function of quantity (i.e. specifies a volume discount), and $f_{ij}(v_{ij})$ is a functional specification for the impact of particular attribute values v_{ij} on the price of a product. The individual functions $p_{bi}(q_i)$ and $f_{ji}(v_{ij})$ can in general be nonlinear. For any attribute j, the difference $f_{ji}(v_{ij,}=v_{j,hi}) - f_{ji}(v_{ij} = v_{j,base})$ represents the price markup for level hi with respect to the *base* level. Examples would be discrete functions, which specify price markups for different types of CPUs (850 MHz, 900 MHz, 1GHz), as well as continuous functions, which specify the impact of decreasing lead time on price. This functional form can be sent to the auctioneer in an XML-based interchange format. For these purposes, we have designed CPML, an XML schema to describe configurable multi-attribute offers.

5.2 Allocation of Configurable Offers

In the following analysis we restrict ourselves to discrete price markups, where each attribute value v_{ij} of bid i (with indicator variable x_{ijk} for level k) has an individual markup m_{ijk}. In addition, each offer specifies a quantity q_i. The description of a configurable multi-attribute offer is now formulated as in (5) with p_i

being the unit price of a particular configuration $\mathbf{X}_i$. (Note that this is a 0-1 matrix that indicates the levels chosen for each attribute).

$$p_i = p(q_i, \mathbf{X}_i) = p_{bi} + \sum_j \sum_k m_{ijk} x_{ijk} \tag{14}$$

Assuming that there are no homogeneity constraints, we can solve the winner determination problem in a two-step procedure. In the first step we select the best possible configuration for each offer based on the buyer's scoring function. In a second step, the resulting "best configurations" are in the form of conventional multi-attribute offers and can be allocated as described in section (2) – (9).

The first step can be modeled as a variation of the *multiple-choice knapsack problem* [23]. We associate the binary decision variable x_{jk} to each attribute value k of attributes j. The objective maximizes the weighted score s_{jk} for each attribute value with weight being w_j. For ease of reading we omit the first subscript i for bids in the first part of this section. The objective used in (15) denotes the price attribute p of a selected configuration as an extra variable. Note, that we assume an *additive, quasi-linear utility function* with a linear (decreasing) scoring function on price, $s_p(p)$. Constraint (16) specifies that for each attribute exactly one value must be selected. Constraint (17) restricts the unit price to be smaller or equal to the buyer's unit reservation price C. In (18) we determine the value of the continuous variable p_i. This constraint could also be omitted, so that the price p_i needs to determined outside the optimization based on the selected attribute values.

$$\max \sum_j w_j \left(\sum_k s_{jk} x_{jk} \right) + w_p s_p(p) \tag{15}$$

subject to

$$\sum_k x_{jk} = 1 \; \forall j \in J \tag{16}$$

$$\sum_j \sum_k m_{ijk} x_{jk} + p_{bi} \le C \tag{17}$$

$$\sum_j \sum_k m_{ijk} x_{jk} + p_{bi} = p_i \tag{18}$$

$$x_{jk} \in \{0,1\} \; \forall j \in J, \forall k \in K \tag{19}$$

The procedure allows suppliers considerably more flexibility in specifying offers, while at the same time, the bids can be ranked and suppliers can compete in an open-cry manner.

5.3 Winner Determination with Homogeneity Constraints

The formulations in sections 5.2 work only under the assumption that there are no homogeneity constraints on the buyer's side. Being able to formulate homogeneity constraints is a very useful feature for the evaluation of configurable offers, however, at the expense of complexity of the winner determination. The following formulas (20) – (28) show the overall MIP model for the evaluation of configurable multi-attribute offers considering multiple sourcing and homogeneity constraints.

$$\max \sum_i q_i \left\{ \sum_j w_j \left(\sum_k s_{ijk} x_{ijk} \right) + w_p s_p \sum_j \sum_k m_{ijk} x_{ijk} + w_p (s_p p_{bi} + d) y_i \right\} \quad (20)$$

subject to

$$\sum_k x_{ijk} = y_i \quad \forall j \in J, \forall i \in I \quad (21)$$

$$\sum_i q_i \left(\sum_j \sum_k m_{ijk} x_{ijk} + p_{bi} y_i \right) \le C \quad (22)$$

$$D_{\min} \le \sum_i q_i y_i \le D_{\max} \quad (23)$$

$$L_{\min} \le \sum_i y_i \le L_{\max} \quad (24)$$

$$0.1 z_{jk} \le \sum_{i,l \in T_{jk}} x_{ijk} \le \left\| T_{jk} \right\| z_{jk} \forall j \in J \quad (25)$$

$$\sum_k z_{jk} = 1 \quad \forall j \in J \quad (26)$$

$$x_{ijk} \in \{0,1\} \quad \forall i \in I, \forall j \in J, \forall k \in K \quad (27)$$

$$y_i \in \{0,1\} \quad \forall i \in I \quad (28)$$

The objective maximizes the overall score of the selected configurations. As in (15) we assume an additive and quasi-linear scoring function with a linearly decreasing function on price. The variable s_{ijk} again denotes the score for a particular qualitative attribute value, whereas x_{ijk} is a binary indicator variable which indicates whether a particular attribute value has been chosen. The variable s_p describes the slope of the linear scoring function for price, whereas d is its intercept.

In (21) we select exactly one attribute value for each attribute in an offer, and introduce y_i as an indicator variable for a particular offer. The constraint in (22) specifies a reservation price C. (23) restricts the quantity to match an upper and lower bound (D_{min} and D_{max}) specified by the buyer, and (24) limits the number of winners. Finally, (25) and (26) specify homogeneity constraints. In (25) we introduce the indicator variable z_{jk} that assumes the value 1 if any suppliers are chosen with a bid at level k for attribute j. T_{kj} is defined as the set of bids at level K for attribute J. Compared to formula (2) – (6) we have dropped the index l, because we assume every bidder to submit only one configurable offer.

5.4 Treatment of Logical Configuration and Discount Rules

It is often essential for suppliers to express rules, which define constraints on the combination of attribute values, or discounts and markups based on some combination of attribute values. For instance, a *configuration rule* may include compatibility restrictions, saying attribute value x_{112} cannot be connected to attribute value x_{123}, or requirements like attribute value x_{112} and x_{132} needs attribute value x_{123}. Here, x_{ijk} is a binary indicator variable that takes a value of 1 if the level k of attribute j and offer i is chosen. In this section we are only considering configuration rules within an offer and, hence, omit the subscript i for the sake of readability. CPML provides the possibility to express these rules as logical implications. For example, the proposition

$$x_{23} => \neg\, x_{31} \quad (29)$$

describes the configuration rule that if a certain motherboard, x_{23} is selected by the user, then the buyer is restricted from using a certain type of CPU, x_{31}. Logical implication (=>) allows, that if any other kind of motherboard is selected, the particular CPU may or may not be chosen. Another type of rules, which is often found in practice are so called *discount rules*. For example,

$$x_{12} \wedge x_{31} \Leftrightarrow p^- \tag{30}$$

where x_{jk} again describe particular attribute values and p^- describes a certain discount (or markup) enforces a discount upon selection of these attribute values. The discount is only given, if and only if $x_{12} \wedge x_{31}$ is true. If $x_{12} \wedge x_{31}$ is false, then no discount will be granted. Therefore, we use the equivalence operator (⇔) for discount rules. We use x_{jk} to denote the *logical* as well as the *binary variable* in the MIP formulation. For ease of reading we omit the first subscript *i* for bids in the first part of this section.

For the evaluation of a configurable offering, these additional rules have to be considered in the IP formulation. In order to obtain an equivalent mathematical representation for any propositional logic expression, one must first consider basic logical operators to determine how each can be transformed into an equivalent representation in the form of an equation or inequality. Raman and Grossman [24] specify transformations, which can then be used to convert general logical expressions into an equivalent mathematical representation. Some of these transformations are described in Table 2.

Logical relation	Pure logical expression	Representation as linear inequalities
Logical "OR"	$x_{11} \vee x_{21} \vee_{\dots} \vee x_{n1}$	$x_{11} + x_{21} {}_{\dots} + x_{n1} \geqq 1$
Logical "AND"	$x_{11} \wedge x_{21} \wedge_{\dots} \wedge x_{11}$	$x_{11} \geqq 1;\ x_{21} \geqq 1;\ \dots;\ x_{n1} \geqq 1$
Implication (=>)	$\neg x_{11} \vee x_{21}$	$1 - x_{11} + x_{21} \geqq 1$
Equivalence (⇔)	$(\neg x_{11} \vee x_{21}) \wedge (\neg x_{21} \vee x_{11})$	$x_{11} - x_{21} \leqq 0;\ x_{21} - x_{11} \leqq 0$

Table 2: Representation of logical relations with linear inequalities

A common approach to convert a general logical expression into inequalities is to first transform it in its equivalent conjunctive normal form (CNF) representation. CNF involves the application of pure logical operations (and $\wedge$, or $\vee$, not $\neg$), and is a conjunction of clauses. A clause is defined as a set of basic literals separated by $\vee$-operators, such as

$$(x_{12} \vee x_{23}) \wedge (\neg x_{34} \vee x_{45}) \tag{31}$$

CNF can then be expressed as a set of linear inequality constraints, as shown in Table 2. We have chosen this approach to transform the configuration and discount rules in CPML into appropriate constraints in our IP formulation described in (20) – (28). Formulas (32) to (37) show how the proposition in (32) can be translated into linear constraints in our IP formulation.

$$x_{12} \wedge x_{23} \Leftrightarrow p^- \tag{32}$$

$$(\neg(x_{12} \wedge x_{23}) \vee p^-) \wedge (\text{Ø}p^- \vee (x_{12} \wedge x_{23})) \tag{33}$$

$$(\neg x_{12} \vee \neg x_{23} \vee p^-) \wedge (\text{Ø}p^- \vee x_{12}) \wedge (\neg p^- \vee x_{23}) \tag{34}$$

$$x_{12} + x_{23} - p^- \leqq 1 \tag{35}$$

$$x_{12} - p^- \geqq 0 \tag{36}$$

$$x_{23} - p^- \geqq 0 \tag{37}$$

In (33) the equivalence operator has been transformed into a proposition with pure logic operators. Using DeMorgan's Theorem the negation operator of the first term in brackets is moved inwards, so that we get CNF in (34). Finally, in (35) to (37) CNF is translated into inequalities, which can be added to the integer programming formulation. In addition, we have to introduce an additional binary indicator variable for p^- in our model, which indicates the discount if the rule takes effect.

Logical expression	Equivalent linear inequalities
$\wedge x_{ijk}$ => x_{irs} and $i \in I$, $j,r \in J$, $k \in K_j$, $s \in K_r$, $\forall\, j \neq r$	$\sum_{ijk \in R} (1 - x_{ijk}) + x_{irs} \geq 1 \ \forall i \in I$
$\vee x_{ijk}$ => x_{irs} and $i \in I$, $j,r \in J$, $k \in K_j$, $s \in K_r$, $\forall\, j \neq r$	$x_{irs} - x_{ijk} \geq 0 \ \ \forall ijk \in R, \forall i \in I$
$\wedge x_{ijk} \Leftrightarrow p^-$ and $i \in I$, $j \in J$, $k \in K_j$	$\sum_{ijk \in R} (1 - x_{ijk}) + p \geq 1 \ \forall i \in I$ $x_{ijk} - p \geq 0 \ \ \forall ijk \in R, \forall i \in I$
$\vee x_{ijk} \Leftrightarrow p^-$ and $i \in I$, $j \in J$, $k \in K_j$	$\sum_{ijk \in R} x_{ijk} - p \geq 0 \ \forall i \in I$ $p - x_{ijk} \geq 0 \ \ \forall ijk \in R, \forall i \in I$

Table 3: Translation of typical configuration and discount rules

The logical expressions in Table 3 describe common forms of configuration and discount rules with only conjunctions or only disjunctions in the antecedent and one literal in the consequent. We have used the notation with three subscripts so that the additional constraints can be added to the optimization formulation in (20) – (28). *R* is defined as the set of attribute values in the antecedent of a rule in an offer. Of course, the antecedent and the consequent of these rules can in general be any combination of conjunctions and disjunctions. In other words, with the relations given in Table 2 one can systematically model an arbitrary propositional logic expression as a set of linear equality and inequality constraints.

5.5 Computational Issues

From a computational point of view the allocation of configurable offers *without homogeneity* constraints is considerably easier to solve than the *problem with homogeneity constraints* described in section 5.3. Without homogeneity constraints, the overall winner determination can be split in several smaller problems (see section 5.2), in which the best possible configuration for each configurable offer is selected based on a buyer's scoring function. In our numerical simulations, the selection of the best configuration for an offer with four configuration rules, and ten attributes with four attribute values each could find the best configuration in the order of milliseconds using a commercial optimization package. The results of these individual selection problems are then used in the overall winner determination described in equations (2) – (9), the runtime of which has been analyzed in 4.4.

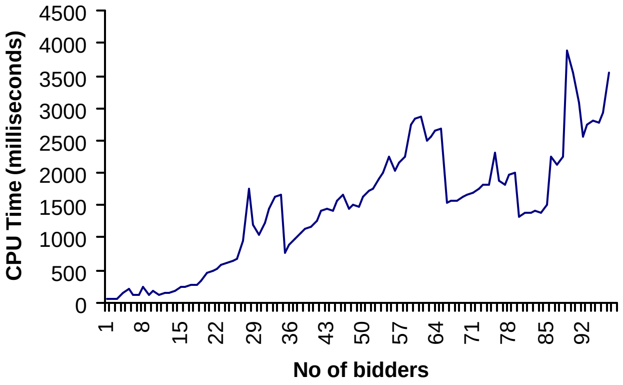

Figure 3: Experimental results for the allocation configurable offers with increasing numbers of bidders

The winner determination problem is considerably harder to solve in the presence of homogeneity constraints, because all bids have to be considered at the same time. Figure 3 shows the CPU times of a randomly generated problem instance with 30 attributes, a single homogeneity constraint on one of the attributes, and an increasing number of bidders. Figure 4 investigates the impact of the homogeneity constraints on the runtime of the winner determination. The problem size was constant with 60 bids and 20 attributes and no other side constraints were set for the experiment.

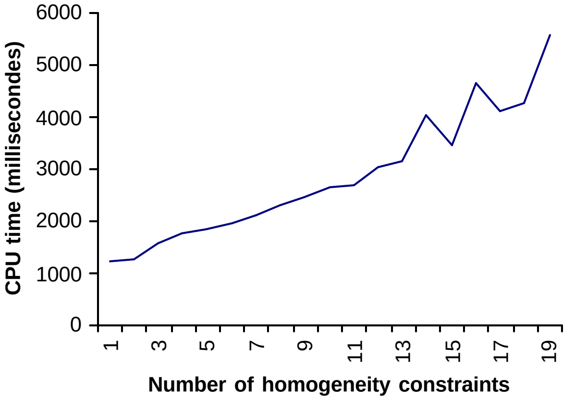

Figure 4: Experimental results for the allocation configurable offers with increasing numbers of homogeneity constraints

In our future research, we plan to extend the analysis towards configurable offers, which allow the specification of volume discounts. This aspect adds an additional degree of flexibility to suppliers, however, again at the expense of complexity in the winner determination.

5.6 Discussion of the Approach and Related Work

In reality, one might find more complex propositions or even completely new types of rules. For example, some companies use *eligibility rules* to differentiate their offerings across customers or customer groups. Rather than on product features, eligibility rules are based on customer characteristics, and can be used in a preprocessing step to filter the products, for which a particular customer is eligible. As we have shown, propositional logic is a useful and easy-to-understand form of knowledge representation for various types of business rules in the context of configurable offers.

Although the use of logic for multi-attribute and configurable offers is new, researchers have been looking into bidding languages in the context of combinatorial auctions, in which agents submit bundle bids [25, 26]. The winner determination problem in combinatorial auctions is essentially to find the set of bundle bids, which maximize the overall revenue or minimize cost in a reverse auction. Conceptually, this is equivalent to the weighted set packing problem. Often a buyer with a complex utility function will need to express multiple bundle bids in order to accurately reflect her utility function. Certain languages allow logical combinations of goods as a formulae, others allow logical combinations of bundle bids as a formulae, or both (aka *generalized logical bid*, GLB). The difference between logical bidding languages for combinatorial auctions and the work presented in this paper is the general way the agents' utilities and bids are described and the allocation problem is formulated. In GLB a propostion determines the overall price of a bundle, or combination of bundles. In CPML a proposition can restrict certain attribute combinations or determine markups or discounts on a subset of attribute values. The objective in a combinatorial auction is to maximize the auctioneer's revenue, or minimize the cost respectively. In the type of multi-attribute reverse auctions, which is considered in this paper, buyer preferences for various attributes of a product are described by a weighted scoring function, which needs to be considered in the winner determination.

There are a number of new initiatives to establish generic rule languages for business rules in electronic commerce applications. RuleML is based on situated courteous logic programs. Logic programs are based on predicate calculus and allow for variables, functions, and quantifiers in logic propositions. Although more complex, predicate calculus (aka first-order logic) provides considerably more expressiveness than propositonal logic. In contrast to predicate calculus, however, traditional logic programs are restricted to horn clauses. Situated courteous logic programs extend traditional logic programs with additional features such as prioritized conflict handling and procedural attachments [27]. RuleML is a relatively new approach, and first demo applications are currently being developed to illustrate the use of RuleML in a business context.

There are a number of strategic questions involved with the information revealed in the proposed bidding language. Of course, a bidder doesn't have to reveal his true pricing rules, nor does she have to reveal prices for all possible products. Bidders might not want to reval any pricing or configuration rules at all throughtout a negotiation. Nevertheless, this bidding language provides the possibility to do so and communicate pricing information about a large product space in an effective manner. In many industries, however, companies are already revealing these kinds of rules as a configuration system on the web (e.g. web-based PC configurators). But even without communicating any pricing rule, the language might be useful to describe the range of possible product configurations offered by a supplier.

6 CONCLUSIONS

In this paper we have discussed a number of winner determination problems in the context of multi-attribute auctions. During the past few years many advances have been made in the area of computational mechanism design. Besides multi-attribute auctions, many new auction mechanisms such as combinatorial auctions [28-30] or volume discount auctions [31] have been developed. The winner determination in these auctions is usually a computationally hard problem. This computational complexity has been a significant hurdle for the widespread use of these advanced auction models.

In an attempt to foster a more widespread (re-)use of multidimensional auction mechanisms, we have implemented the Multidimensional Auction Platform (MAP) [32], an object framework with a generic API to bid evaluation and allocation algorithms. MAP consists of a generic Java API for different allocation mechanisms, an XML schema to define various kinds of bids and asks, and a database schema to make these bids and asks persistent. Currently it implements winner determination algorithms for combinatorial and volume-discount auctions, as well as the multi-attribute allocation algorithms described in this paper. This framework enables application programmers to specify buyer preferences, allocation rules and supplier offerings in a declarative manner, and solve the allocation problems without having to re-implement the computationally complex algorithms. MAP is currently being used in a large-scale procurement marketplace for the retail industry.

7 REFERENCES

[1] W. Vickrey, "Counterspeculation, Auctions, and Competitive Sealed Tenders," *Journal of Finance*, pp. 8-37, 1961.

[2] S. Dasgupta and D. F. Spulber, "Managing procurement auctions," *Information Economics and Policy*, vol. 4, pp. 5-29, 1989.

[3] R. L. Keeny and H. Raiffa, *Decision Making with Multiple Objectives: Preferences and Value Tradeoffs*. Cambridge, UK: Cambridge University Press, 1993.

[4] M. Bichler, "BidTaker: An Application of Multi-Attribute Auction Markets in Tourism," presented at Wirtschaftsinformatik 2002, Augsburg, Germany, 2002.

[5] Y.-K. Che, "Design Competition through Multidimensional Auctions," *RAND Journal of Economics*, vol. 24, pp. 668-680, 1993.

[6] F. Branco, "The Design of Multidimensional Auctions," *RAND Journal of Economics*, vol. 28, pp. 63-81, 1997.

[7] P. Milgrom, "An Economist's Vision of the B-to-B Marketplace," Perfect.com, White Paper October 2000.

[8] J. G. Riley and J. G. Samuleson, "Optimal auctions," *American Economic Review*, vol. 71, pp. 381-392, 1981.

[9] M. Bichler, "Decision Analysis - A Critical Enabler for Multi-attribute Auctions," presented at 12th Electronic Commerce Conference, Bled, Slovenia, 1998.

[10] M. Bichler, M. Kaukal, and A. Segev, "Multi-attribute auctions for electronic procurement," presented at First IBM IAC Workshop on Internet Based Negotiation Technologies, Yorktown Heights, NY, USA, 1999.

[11] J. Teich, H. Wallenius, and J. Wallenius, "Multiple-issue auction and market algorithms for the world wide web," *Decision Support Systems*, vol. 26, pp. 49-66, 1999.

[12] N. Vulkan and N. R. Jennings, "Efficient mechanisms for the supply of services in multi-agent environments," *Decision Support Systems*, vol. 28, pp. 5-19, 2000.

[13] M. Bichler, "An Experimental Analysis of Multi-Attribute Auctions," *Decision Support Systems*, vol. 28, 2000.

[14] D. C. Parkes and J. Kalagnanam, "Multiattribute Reverse Auctions," presented at AAAI, 2002.

[15] D. R. Beil and L. M. Wein, "An Inverse-Optimization-Based Auction Mechanism To Support a Multi-Attribute RFQ Process," MIT, Boston, MA, USA, Research Report December 2001.

[16] W. Edwards, "How to use multiattribute utility measurement for social decisionmaking," *IEEE Transactions on Systems, Man, and Cybernetics SMC*, vol. 7, pp. 326-340, 1977.

[17] T. L. Saaty, *The Analytic Hierarchy Process*. New York, USA: McGraw Hill, 1980.

[18] D. L. Olson, *Decision Aids for Selection Problems*. New York, et al.: Springer, 1995.

[19] R. A. Ribeiro, "Fuzzy multiple attribut decision making: A review and new preference elicitation techniques," *Fuzzy Sets and Systems*, vol. 78, pp. 155-181, 1996.

[20] M. Bichler, J. Lee, H. S. Lee, and J.-Y. Chung, "ABSolute: An Intelligent Decision Making Framework for E-Sourcing," presented at 3rd International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems, San Jose, CA, 2001.

[21] V. S. Iyengar, J. Lee, and M. Campbell, "Q-Eval: Evaluating Multiple Attribute Items Using Queries," IBM Research, New York, Research Report 2001.

[22] W. L. Winston, *Operations Research - Applications and Algorithms*, 3 ed. Bemont, CA: Duxbury Press, 1994.

[23] S. Martello and P. Toth, *Knapsack Problems*. Chichester, New York: John Wiley & Sons, 1990.

[24] R. Raman and I. E. Grossmann, "Relation between MILP modelling and logical inference for process synthesis," *Computers and chemical engineering*, vol. 15, pp. 73-84, 1991.

[25] N. Nissan, "Bidding and allocations in combiatorial auctions," presented at ACM Conference on Electronic Commerce (EC-2000), Minneapolis, MI, 2000.

[26] C. Boutilier and H. H. Hoos, "Bidding Languages for Combinatorial Auctions," presented at 17th International Joint Conference on Artificial Intelligence (IJCAI) 2001, Washington, USA, 2001.

[27] B. Grosof, "Representing E-Business Rules for the Semantic Web: Situated Courteous Logic Programs in RuleML," presented at Workshop on Information Technologies and Systems (WITS '01), 2001.

[28] S. Rassenti, V. L. Smith, and R. L. Bulfin, "A Combinatorial Auction Mechanism for Airport Time Slot Allocations," *Bell Journal of Economics*, vol. 13, pp. 402-417, 1982.

[29] M. H. Rothkopf and A. Pekec, "Computationally Manageable Combinatorial Auctions," presented at Maryland Auction Conference, Maryland, USA, 1998.

[30] T. Sandholm, "Approaches to winner determination in combinatorial auctions," *Decision Support Systems*, vol. 28, pp. 165-176, 1999.

[31] A. Davenport and J. Kalagnanam, "Price Negotiations for Direct Procurement," IBM T.J. Watson Research, New York, Research Report RC 22078, February 2001.

[32] M. Bichler, J. Lee, H. S. Lee, and J. Kalagnanam, "Resource Allocation Algorithms for Electronic Auctions: A Framework Design," presented at 3rd International Conference on Electronic Commerce and Web Technologies (EC-Web), Aix-en-Provence, France, 2002.

APPENDIX: ABBREVIATIONS

C	unit reservation price
D_{min}	lower bound for a buyer's demand
D_{max}	upper bound for a buyer's demand
$f_{ij}(v_{ij})$	defines the price markup for a particular attribute value v_{ij} of offer I
I	set of offers
J	set of attributes
K_j	set of attribute values for attribute j
L	set of bidders
L_{min}	Minimum number of winning bidders
L_{max}	Maximum number of winning bidders
m_{ijk}	markup of level k of attribute j
$p_{bi}(q_i)$	price of a base configuration for an offer as a function of quantity
p_i / p_{il}	total price for a particular offer i (of bidder l)
p^-	a discount or markup specified in logical propositions about configuration rules
q_i / q_{il}	total quantity for a particular offer i (of bidder l)
s_i / s_{il}	score of offer i (of bidder l)
$S_j(.)$	scoring function for attribute j
$S(\mathbf{v}_i)$	set of attribute values for attribute j
$s_p(p_i)$	linear (decreasing) scoring function on price
T_{jk}	set of all bids at level K for attribute J
V_j	set of allowable values for attribute j
v_{ij}	value of attribute j of offer i
w_j	weight of attribute j
$\mathbf{X}_i$	0-1 matrix that indicates the levels chosen for each attribute in a configuration
x_{ijk}	a binary indicator variable for a particular attribute level v_{ij}
y_i	an indicator variable that one particular offer i has been chosen
y_l	an indicator variable that an offer of a particular supplier $l2L$ has been chosen
z_{jk}	indicator variable that assumes the value 1 if any suppliers are chosen with a bid at level k for attribute j
C	the number of a supplier's bids
r_{il}	best configuration from offer i based on a given scoring function