Compact Bidding Languages and Supplier Selection for Markets with Economies of Scale and Scope

Martin Bichler, Stefan Schneider

Department of Informatics, TU München, Germany

Kemal Guler, Mehmet Sayal

HP Labs, Palo Alto, USA

Abstract

Combinatorial auctions have been used in procurement markets with economies of scope. Preference elicitation is already a problem in single-unit combinatorial auctions, but it becomes prohibitive even for small instances of multi-unit combinatorial auctions, as suppliers cannot be expected to enumerate a sufficient number of bids that would allow an auctioneer to find the efficient allocation. Auction design for markets with economies of scale and scope are much less well understood. They require more compact and yet expressive bidding languages, and the supplier selection typically is a hard computational problem. In this paper, we propose a compact bidding language to express the characteristics of a supplier's cost function in markets with economies of scale and scope. Bidders in these auctions can specify various discounts and markups on overall spend on all items or selected item sets, and specify complex conditions for these pricing rules. We propose an optimization formulation to solve the resulting supplier selection problem and provide an extensive experimental evaluation. We also discuss the impact of different language features on the computational effort, on total spend, and the knowledge representation of the bids. Interestingly, while in most settings volume discount bids can lead to significant cost savings, some types of volume discount bids can be worse than split-award auctions in simple settings.

Keywords: Decision support systems, auctions/bidding, e-commerce

1. Introduction

Economies of scale and scope describe key characteristics of a supplier's production function that influence allocations and prices on procurement markets. This paper is motivated by real world procurement negotiations of an industry partner, where procurement managers need to purchase large volumes of multiple items. This might be the yearly demand for different types of tires in the car industry or for different types of memory chips or hard disks in the PC industry. Often in these circumstances, economies of scale are jointly present with economies of scope. On the one hand suppliers that set up a finishing line for a certain product have high setup costs, but low marginal costs leading to a unit price degression. On the other hand economies of scale arise in shipping and handling a larger number of items to a customer, and in the joint procurement of raw materials. Unit prices can not only decrease, however. A supplier might increase unit prices, if the demand exceeds his capacity and he has to work shifts or purchase items from third-party suppliers.

Split-award auctions are regularly used for multi-item, multi-unit negotiations, where the best bidder gets the larger share of the volume for a particular quantity and the second best bidder gets a smaller share (e.g., a 70/30% split) (Anton and Yao 1992, Perry and Sakovics 2003, Anton et al. 2009). This is done to assure supply if one supplier is unable to fulfill his contract and a minimal supplier pool in the long run. With significant economies of scale, suppliers face a strategic problem in simple split-award auctions with only a single unit price. Since there is uncertainty about which quantity they will get awarded, they might speculate and bid less aggressively based on the unit cost for the smaller share. In other words, simple split award auctions do not allow suppliers to adequately express economies of scale.

In the recent years, driven by the new possibilities of the Internet, a growing literature is devoted to the design of optimization-based markets (aka. smart markets) (Gallien and Wein 2005), and in particular to combinatorial auctions, where bidders are allowed to submit bids on packages of discrete items (Cramton et al. 2006). The promise of these mechanisms is that by allowing market participants to reveal more comprehensive information about cost structures or utility functions, this can drastically increase allocative efficiency and lead to higher economic welfare. Unfortunately, the matching of complex preference profiles typically leads to hard optimization problems.

The literature in this field is typically focused on multi-item but single-unit negotiations and respective auction formats do not easily extend to multi-unit markets with economies of scale. While preference elicitation is already a fundamental problem for bidders in single-unit combinatorial auctions, it becomes prohibitive in multi-unit combinatorial auctions. Markets with significant economies of scale and scope require a fundamentally different bidding language that allows to specify discount rules rather than a huge number of multi-unit package bids.

1.1. Focus of this Research

In this paper, we will introduce a compact bidding language and the respective allocation problem for procurement markets with economies of scale and scope. This language combines logical spend conditions defined on volume and quantity with different types of discounts. Two central types of volume discounts discussed in the literature are incremental discounts and total quantity discounts. *Total quantity discounts* have been described as a discount policy, where the supplier has specified a number of quantity intervals (aka. discount intervals), and the price per unit for the entire quantity depends on the discount interval in which the total amount ordered lies (Goossens et al. 2007). In contrast, *incremental volume discounts* describe a discount policy, where the discounts apply only to the additional units above the threshold of the quantity interval. In business practice, such discount policies are often also defined on spend or on spend and quantity for one or more items. In addition, we will also allow for lump sum discounts, defining a one time reverse payment on overall spend or quantity.

So far, optimization formulations only exist for incremental or for total quantity discount bids, defined on quantity purchased. We focus on a bidding language, which provides considerably more flexibility in the discount policies used. Often increased expressiveness of a formal language comes at the cost of increased computational complexity (Papadimitriou 1993, Dantsin et al. 2001). It is important to understand, if increased expressiveness of this bidding language also leads to a higher computational burden during the supplier selection.

Procurement managers are not only interested in the cost-minimal solution to this optimization problem. They typically apply a number of side constraints in an interactive manner to find a good solution that considers a number of strategic and operational goals. Examples are upper and lower bounds on the number of suppliers or the volume awarded to one or a group of suppliers. These side constraints are not always known in advance exactly and whether they should be considered or not also depends on their impact on the total cost. For example, a purchasing manager wants to have no more than five suppliers, but he would also be willing to accept six or seven, if it reduces total cost significantly. This interactive exploration of different award scenarios based on allocation constraints and different types of volume discounts is also referred to as *scenario analysis*, and a number of decision support tools are provided by e-sourcing vendors in this area (Gartner 2008). Although, this type of decision support is vital for many companies and leading to multi-million dollar procurement decisions, there is surprisingly little academic literature in this field.

Scenario analysis poses tight time constraints to allow for an interactive exploration of different award scenarios. As we will show, the winner determination problem is an $\mathcal{NP}$-complete problem. A main research question in this paper is, for which problem sizes (number of bidders, bids, items, and discount intervals) a procurement manager can hope to solve respective instances optimally in acceptable time. Optimality of the solutions is desirable, as the allocation problem tries to minimize the cost of the buyer, and even a small decrease in procurement cost will directly impact company gain. In addition, we will analyze the impact of different discount policies on total cost. Some discount policies make it easier to approximate the underlying cost functions closely.

1.2. Contributions

In summary, the contributions in this paper are the following: We will *introduce a compact bidding language for markets with economies of scale and scope*, referred to as $\mathcal{L}_{ESS}$. Our bidding language allows for two different types of discounts, which have already been discussed in the literature: incremental and total quantity discount bids. Our approach allows to *handle both types of volume discounts*, and we have seen several applications, where different bidders submit different types of volume discount bids. In addition to previous approaches, $\mathcal{L}_{ESS}$ allows for *lump sum discounts on total spend to model economies of scope and various conditions on spend or quantity* for the different discount types. As a result, $\mathcal{L}_{ESS}$ is considerably more expressive than previous approaches and gives suppliers high flexibility in specifying their offerings. Apart from expressiveness, we introduce *description length* as an important criterion for bid languages, since bidders cannot be expected to submit arbitrarily many parameters or bids. We will see that there are considerable differences between bundle bids, $\mathcal{L}_{ESS}$ bids with total quantity or with incremental volume discounts.

In this work, we will investigate the buyer's problem, who needs to select quantities from suppliers providing bids in $\mathcal{L}_{ESS}$ such that his costs are minimized and his demand is satisfied. We will refer to this problem as *Supplier Quantity Selection (SQS)* problem and propose a respective *mixed integer program* (MIP). Modeling matters and there are considerable differences in the solution time depending on different model formulations. We will also discuss *additional allocation constraints* as they are typically used for scenario navigation.

Procurement managers need a clear understanding of which problem sizes they can analyze in an interactive manner during the scenario analysis or in dynamic auctions. Therefore, we will show that SQS is $\mathcal{NP}$-complete, and report on an extensive *evaluation of the empirical hardness* of the supplier quantity selection problem. Similar analyses have recently been performed for the winner determination problem in combinatorial auctions (Leyton-Brown et al. 2009). It is important that problem instances for the experimental evaluation mirror real-world characteristics. We introduce a *cost function* for markets with scale and scope economies and generate bids based on this cost function. $\mathcal{L}_{ESS}$ and the software framework used in this paper have already been used to support a number of high-stakes sourcing decisions with an industry partner. The synthetic bids matched the characteristics of those that we also found in the field. The experimental results show that realistic problem sizes of the SQS problem can be solved to optimality in a matter of minutes with IBM's CPLEX (version 12) and Gurobi 2.0. (All results reported are based on CPLEX.) We also find that the shape of the underlying cost function and the demand can have a significant impact on the runtime of the problems and empirical evaluations need to be interpreted with care.

Previous work has only focused on the computational complexity of the winner determination problem. The cost curves in our experimental evaluation allow us to *compare the total cost* achieved with $\mathcal{L}_{ESS}$ bids and different types of volume discounts and bids for split-award auctions. While we do not discuss mechanism design questions in our analysis, we assume a direct revelation mechanism where bidders submit bids that best reflect their cost curves. Even if bidding behavior in the lab or in the field is different, this result suggests that a richer bidding language can lead to considerably lower cost and more efficient results in markets with economies of scale and scope with $\mathcal{L}_{ESS}$. However, we also find that total quantity discounts with only a few intervals can lead to higher spend than simple split-award auctions.

In section 2, we provide an overview of related literature. Section 3 introduces $\mathcal{L}_{ESS}$, a bid language for markets with economies of scale and scope, and discusses relevant features of the language. In section 4 we formulate the winner determination problem as a mixed integer program, and propose various extensions in section 5. Section 6 describes the experimental design, while section 7 summarizes the main results of our experimental analysis. Finally, section 8 provides conclusions and an outlook on future research.

2. Related Literature

The literature on supplier selection and volume discounts includes studies of various discounting schemes, such as unit discounts (Silverson and Peterson 1979), inventory models with demand uncertainty and incremental quantity discounts and carload quantity discounts (Jucker and Rosenblatt 1985, Lee and Rosenblatt 1986). Munson and Rosenblatt (1998) provide a perspective on discounts used in practice, while Chaudhry et al. (1993) discuss a vendor selection model in the presence of price breaks.

Davenport and Kalagnanam (2000) were among the first authors to focus on auctions with incremental volume discount bids. An application thereof has been described in Hohner et al. (2003). Their bidding language requires suppliers to specify continuous supply curves for each item. Eso et al. (2001) further advance the ideas described in Davenport and Kalagnanam (2000) and allow for discontinuities and decreasing slopes in the bids. They use a branch-and-price approach to solve the winner determination problem.

There has also been some work on total quantity discounts, where the unit price starting at a particular quantity is charged for the entire quantity purchased, not only for the units above the threshold quantity. Katz et al. (1994) discuss a procurement decision support system and a respective mathematical program with total quantity discounts. Crama et al. (2004) investigate a problem, where a chemical company needs to purchase a number of ingredients from one or more suppliers with a total quantity discount. Here, only one discount rate is used for all ingredients. Crama et al. (2004) also need to decide how to use the purchased ingredients to manufacture the desired quantities of the endproducts, where there are alternative recipes, which is different to the problem analyzed in this paper.

In contrast, van de Klundert et al. (2005) describe a procurement problem in the telecom industry, where a telecom company needs to acquire capacity to accommodate its international calls. Carriers provide capacity and specify total quantity discounts. The general problem has been discussed in Goossens et al. (2007). They provide an interesting contribution to the supplier selection problem with total quantity discounts and proof that no polynomial-time approximation scheme with constant worst-case ratio exists for this supplier selection problem and that the decision version is strongly $\mathcal{NP}$-complete. They also show that the LP relaxation can be solved as a min-cost flow problem, and suggest a min-cost flow based branch-and-bound algorithm. This algorithm worked best for instances, where the number of suppliers does not exceed 20. For larger instances an LP-based branch-and-cut approach performed best. The paper also highlights the ”more for less” phenomenon in total quantity discounts, i.e., that it can be cheaper to purchase more due to the sawtooth form of the total cost function.

We have worked with a number of procurement managers in the past few years. While the above academic work covers important requirements, many real-world cases demand for a more comprehensive bidding language for practical applicability. In many applications, some suppliers provide incremental volume discount bids, others total quantity discount bids, or overall lump-sum discounts on total spend. As outlined, in this paper, we considerably extend the expressiveness of the bidding languages discussed in the literature and propose a mixed integer programming formulation to solve practically relevant problem sizes. In contrast to previous work, we suggest a parametric cost function to generate realistic bids and analyze different discount policies based on cost and description length.

While the academic literature is still in its infancy, a number of companies such as CombineNet, Emptoris, Iasta, and TradeExtension (Gartner 2008, Giunipero et al. 2009) provide decision support systems allowing for various types of discounts and complex bid types. These tools enable purchasing managers to explore different award scenarios based on various operational or strategic side constraints. Respective software vendors offer a wide variety of constraints among several dozens or even hundreds of constraint classes (Bichler et al. 2006). Unfortunately, little academic literature is published on the type and size of the optimization problems that can be solved with such packaged software tools, and the algorithmic approaches they use.

3. Bidding Language

In this section, we will introduce a bidding language allowing to describe a supplier's cost function. We focus on markets with economies of scale and scope, where bidders typically express discounts in order to reflect these economic characteristics. A language in computer science and logic assigns a semantic to a syntax. Bidding languages have been studied in the context of combinatorial auctions (Boutilier and Hoos 2001, Abrache et al. 2004, Nisan 2006). However, this research typically focuses on multi-item, but single-unit negotiations.

3.1. Description Length of Bidding Languages

Bidding languages should be expressive enough to allow for the description of different shapes that cost functions can assume including concave and convex shapes. At the same time, the bidding language should allow to describe bids in a compact way with only a few parameters. In general, expressivity of a bid language should increase efficiency of economic mechanisms, since bidders are able to better describe their preferences (Sandholm 2008, Benisch et al. 2008). Combinatorial auctions allow bidders to express all types of synergies across items, but the number of possible bids in large-scale combinatorial auctions is typically beyond what human bidders can express. There is a number of articles on the communication complexity of such auctions (Nisan and Segal 2001). As we have discussed in Section 1, this phenomenon becomes even worse with multi-unit combinatorial auctions. Compact bidding languages, which allow bidders to describe their preferences on multiple items and quantities as a function, can alleviate this problem.

Definition 1. *A* compact bidding language *allows to define the bid price as a function* $p_s : \mathbb{R}^{\mathcal{I}} \rightarrow \mathbb{R}$ *of quantity for one or more items* $i \in \mathcal{I}$.

We will also use the term *bid function* for p_s in this paper. Such a bid function has a particular parametric form. Clearly, the most compact format would be to reveal the parameters and the specification of the true total cost function in a direct revelation mechanism to an auctioneer. For example, we will use a cost function with seven parameters for the experimental evaluation in Section 6.1. The parametric shape of such cost functions might be non-linear and different among suppliers and industries, which is just one of the reasons, why the true specification of the function is typically not revealed in practice. It is rather common to specify volume discounts or markups for economies or diseconomies of scale. Such volume discounts can be formulated as piecewise linear functions. Therefore, such bids are typically an approximation of the underlying cost function, even if bidders are willing to reveal their costs truthfully. Let us now introduce *description length* as a measure for how compact bidders can describe information about their preferences or their underlying cost function.

Definition 2. *The* description length of a bid *consists of the bits to describe the parameters of the bid function* p_s *for a given maximum approximation error,* ϵ^{max}*, to the underlying utility or cost function.*

Bidding languages should allow for a close approximation of wide-spread types of cost functions, but the same time have a low description length. A bad approximation will make it difficult for the auctioneer to find an efficient allocation, even if suppliers bid truthfully. Multi-unit bundle bids in combinatorial auctions allow for close approximations, as they only specify discrete points but at the expense of a huge number of bids required to describe a bidder's costs.

3.2. The $\mathcal{L}_{ESS}$ Bidding Language

Bid languages we observed in practice exhibit substantial structural variation across bidders. Offers from suppliers can come in any combination of incremental or total quantity discounts, depending on multiple conditions. To accommodate the richness observed in practice one needs a language that allows for different discount types, denoted R_d.

In case of diseconomies of scale, for example, when the volume awarded is beyond the production capacity of a supplier, he might want to charge respective *markups* R_m to cover his increased per-unit costs. While markups are conceptually equivalent to volume discounts, we will use a separate notation R_m in the next section, as they need to be modeled differently in our optimization model.

In addition, we regularly observed *lump sum discounts* R_l, which describe refunds of part of the total price. For example, if the volume purchased exceeds a threshold, a supplier might be willing to reduce the overall payment by a fixed amount $R_l = \$10,000$ on the total price. These lump sum discounts R_l can also be defined on spend S_l or quantity Q_l and are often used to describe economies of scope.

In $\mathcal{L}_{ESS}$, bidders should be able to express such different types of discounts. Every supplier $s \in \mathcal{S}$ submits a base price $P_{i,s}$ for every item $i \in \mathcal{I}$ and the maximum quantity $E_{i,s}$ he is willing to supply. In addition, he specifies volume discounts $d \in \mathcal{D}$, lump sum discounts $l \in \mathcal{L}$, and markups $m \in \mathcal{M}$ to modify the base price based on certain *spend conditions*. The total bid price function $p_s : \mathbb{R}^{\mathcal{I}} \rightarrow \mathbb{R}$ of a set of items $\mathcal{I}$ can be written as

$$p_s(x_1, ..., x_{\mathcal{I}}) = \sum_{i \in \mathcal{I}} P_{i,s} x_{i,s} - \sum_{d \in \mathcal{D}} R_d y_d 1_{\{C_d\}} + \sum_{m \in \mathcal{M}} R_m y_m 1_{\{C_m\}} - \sum_{l \in \mathcal{L}} R_l 1_{\{C_l\}}$$

where R_d describes a *volume discount* per unit that is awarded on a quantity y_d if a *spend condition* C_d is true, e.g., after the quantity exceeds a certain lower bound on quantity, Q_d, or spend, S_d. Note that Q_d and S_d can be defined on a particular item provided by the supplier or also a set of items by this supplier. We will use the term "discount interval" and refer to spend conditions, which define a unit price for a particular quantity interval. Volume discounts of a specific bidder can be valid for the total quantity purchased (*total quantity discounts*) or for the amount exceeding a pre-specified threshold (*incremental (volume) discounts*). This also holds for markups R_m. Lump sum discounts R_l are defined on overall spend or quantity, not per unit.

Spend conditions (C) are an important language feature, which allow for much flexibility. By allowing conditional discounts and markups with possibly multiple conditions, we are able to formalize all features of bids that have been considered in the literature or we have encountered in practice. Spend conditions can be defined on a set of items and be based on spend (S) or volume (Q) purchased. For example, if spend on the items A and B is more than \$100,000, a supplier offers an lump sum discount of $R_l = \$4,000$. An elementary spend conditions C is treated as a literal in propositional logic such as $S_{A,B} > 100,000\$$ or $Q_A > 2000$. Composite spend conditions take the form of a conjunction of m elementary conditions. So in general, a discount rule takes the form of a Horn clause, with $C_1 \wedge C_2 ... \wedge C_k \implies R$. We limited discount rules to Horn clauses, in order to keep the corresponding supplier quantity selection problem of the auctioneer as concise as possible (see section 4). Modeling disjunctions and conjunctions in the condition of such a discount rule is possible as well, but rarely necessary as we found.

Definition 3 (Discount rule). *A discount rule F is a Horn clause of the form $C_1 \wedge C_2 ... \wedge C_k \implies R$, where*

- *C_k is a literal defined on spend levels or quantity levels for a set of items, with $k = \{1, .., \mathcal{K}\}$ being the set of respective spend conditions.*
- *R is a discount, i.e., a lump sum discount, an incremental volume discount, a total quantity discount, or a respective markup.*

The supplier is also able to specify disjunctive discount rules, i.e., two or more rules cannot be active at the same time. For example, in case the supplier purchases more than \$10,000 from item A and B, there is a discount of \$1.77. Alternatively, if the supplier buys more than 5,000 units from item A, the discount is \$1.02. Only one of these two discounts is eligible, and the auctioneer will choose the discount that minimizes his total cost.

Definition 4. *An $\mathcal{L}_{ESS}$ bid is a tuple $(\mathcal{P}, \mathcal{E}, \mathcal{H}, \mathcal{K})$, where*

- *$\mathcal{P}$ is a set of base unit prices for each item $i \in \mathcal{I}$,*
- *$\mathcal{E}$ is a set of maximum quantities $E_{i,s}$ that a supplier s can provide for each item $i \in \mathcal{I}$,*
- *$\mathcal{H}$ is a set of discount rules $F \in \mathcal{H}$, and*
- *$\mathcal{K}$ is a set of disjunctions specified on the set of rules in $\mathcal{H}$.*

$\mathcal{L}_{ESS}$ provides expressiveness at low description length by allowing to express step functions to describe the average unit costs of a supplier, or the respective piecewise linear functions describing the total cost function $c_s(x)$.

4. The Supplier Quantity Selection Problem

In the following, we will investigate a buyer's problem, who needs to select quantities from each supplier providing bids in $\mathcal{L}_{\mathcal{ESS}}$ such that his costs are minimized and his demand is satisfied. We will refer to this problem as *Supplier Quantity Selection (SQS)* problem and introduce a respective mixed integer program (MIP) in the following.

We will first introduce some necessary notation. We will use uppercase letters for parameters, lowercase letters for decision variables, and calligraphic fonts for sets. Sets indexed by a member of another set represent the subset of all elements that are relevant to the index. For example, $\mathcal{I}_d$ describes all items that are included in a discount rule $d \in \mathcal{D}$, and x_{i,s_d} describes the quantity purchased from supplier s on item i, which is part of the discount pricing rule $d \in \mathcal{D}$. Such discount rules can be defined on different items or sets of items per supplier.

$$
\min \sum_{i\in\mathcal{I}}\sum_{s\in\mathcal{S}} P_{i,s}x_{i,s} - \sum_{d\in\mathcal{D}} R_d y_d + \sum_{m\in\mathcal{M}} R_m y_m - \sum_{l\in\mathcal{L}} R_l c_l
$$

$s.t.$

$$
\begin{array}{rcllr}
\sum_{s\in\mathcal{S}} x_{i,s} & \geq & W_i & \forall i \in \mathcal{I} & (1) \\
x_{i,s} & \leq & E_{i,s} & \forall i \in \mathcal{I}, \forall s \in \mathcal{S} & (2) \\
\sum_{i\in\mathcal{I}_d} x_{i,s_d} - D_d c_d & \geq & y_d & \forall d \in \mathcal{D} & (3d) \\
\sum_{i\in\mathcal{I}_m} x_{i,s_m} + Bc_m & \leq & y_m + D_m + B & \forall m \in \mathcal{M} & (3m) \\
Bc_d & \geq & y_d & \forall d \in \mathcal{D} & (4d) \\
\sum_{n\in\mathcal{N}_d} j_n - \sum_{\bar{d}\in\bar{\mathcal{D}}_d} c_{\bar{d}} & \geq & |N_d|\, c_d & \forall d \in \mathcal{D} & (5d) \\
\sum_{n\in\mathcal{N}_l} j_n - \sum_{\bar{l}\in\bar{\mathcal{L}}_l} c_{\bar{l}} & \geq & |N_l|\, c_l & \forall l \in \mathcal{L} & (5l) \\
|N_m|^{-1} \left(\sum_{n\in\mathcal{N}_m} j_n + 1 \right) - \sum_{\bar{m}\in\bar{\mathcal{M}}_m} c_{\bar{m}} & \leq & c_m + 1 & \forall m \in \mathcal{M} & (5m) \\
\sum_{i\in\mathcal{I}_n} P_{i,s_n}x_{i,s_n} - \sum_{d\in\mathcal{D}_n} R_d y_d + \sum_{m\in\mathcal{M}_n} R_m y_m & \geq & S_n j_n & \forall n \in \mathcal{N} & (6l,d) \\
\sum_{i\in\mathcal{I}} x_{i,s_n} & \geq & Q_n j_n & \forall n \in \mathcal{N} & (7l,d) \\
\sum_{i\in\mathcal{I}} x_{i,s_n} - Q_n & < & Bj_n & \forall n \in \mathcal{N} & (8m) \\
x_{i,s}, y_d, y_m & \geq & 0 & \forall i \in \mathcal{I}, \forall s \in \mathcal{S}, & \\
 & & & \forall d \in \mathcal{D}, \forall m \in \mathcal{M} & \\
c_d, c_m, c_l, j_n & \in & \{0,1\} & \forall d \in \mathcal{D}, \forall l \in \mathcal{L}, & \\
 & & & \forall n \in \mathcal{N}, \forall m \in \mathcal{M} &
\end{array}
$$

The objective function minimizes the product of all base prices $P_{i,s}$ and quantities $x_{i,s}$ of item i purchased of supplier s, subtracts the sum of all discounts R_d and lump sum discounts R_l, and adds the markups R_m.

The first constraints (1) ensure that the demand W_i is fulfilled, and the second set of constraints (2) that the amount purchased from a product i does not exceed the maximum quantity $E_{i,s}$ provided by each supplier of each item. The constraint sets (3d) and (3m) determine the relevant volume y_d or y_m, for which the discount or markup resp. is defined, and B is a sufficiently large number. For example, if $D_d = 0$, then (3d) defines a total quantity discount, where $y_d = x_{i,s}$, otherwise, D_d is set to the threshold, after which the volume discount is valid, as such describing an incremental volume $y_d = x_{i,s} - D_d$. Typically, the discount intervals and markups hold for a single item, but they can also be defined on multiple items $i \in \mathcal{I}_d$.

For each discount rule, we introduce a binary variable c_d, c_l, and c_m. Such decision variables are determined based on spend conditions, which we define in constraint sets (6-8). Constraint (4d) makes sure that a discount is only provided ($y_{i,s} > 0$) if the respective binary variable for this discount, c_d, is true. Constraint sets (5d), (5l), and (5m) make sure that if a particular set of spend conditions is given ($j_n = 1$), which are a precondition for a discount, markup, or lump sum discount, then also the respective binary variable c_d, c_l, or c_m is true. $|N_d|$, $|N_m|$, and $|N_l|$ describe the number of conditions that need to be true for the respective binary variable to become true. These constraints also allow to specify sets of discount rules $\bar{\mathcal{D}} \subset \mathcal{D}$, $\bar{\mathcal{M}} \subset \mathcal{M}$, and $\bar{\mathcal{L}} \subset \mathcal{L}$, which cannot be active at the same time as the respective rule.

The final sets of constraints (6 - 8) model individual conditions on spend or quantity that need to be fulfilled for a particular discount rule in constraint sets (5). For example, constraint set (6 l,d) specifies a minimum spend condition for volume discounts and lump sum discounts. In words, if the total cost including markups and discounts (not considering other lump sum discounts) exceeds S_n, then an additional lump sum discount will be granted. Constraint set (7 l,d) determines a minimum quantity condition used in volume and lump sum discounts. Constraint set (8 m) defines a minimum quantity condition for a markup rule.

Goossens et al. (2007) describe the problem of selecting a set of suppliers that offer a variety of items using total quantity discounts. The problem is referred to as TQD. They have provided a polynomial reduction of 3-dimensional matching, a well-known strongly $\mathcal{NP}$-complete problem, to TQD. Showing that SQS is $\mathcal{NP}$-complete is straightforward, as it contains TQD as a special case.

Theorem 1. *The decision version of the SQS problem is strongly $\mathcal{NP}$-complete.*

Here, we refer to TQD' as the decision version of the more-for-less variant of TQD, and SQS' as the decision version of SQS. Any input for TQD' can be solved with SQS', and a solution to SQS' would solve TQD'. SQS' is obviously in $\mathcal{NP}$ since given a solution it suffices to check the constraints and the value of the solution.

While this shows that SQS is at least as hard as TQD to solve, it is interesting to understand how the increased expressiveness of $\mathcal{L}_{\mathcal{ESS}}$ impacts the empirical hardness of the problem. As is often the case, the formulation of the problem matters, and there are significant differences in runtime depending on the model formulation (Hooker 2009).

5. Scenario Analysis

We will focus on scenario analysis as a typical use case. During scenario analysis, procurement managers typically use additional side constraints to explore different award scenarios. For example, a purchasing manager might be interested in an optimal allocation with a maximum of 5 winners, or the optimal allocation, where the spend on an individual supplier is limited to 1 million dollars due to certain risk considerations. An ex-post analysis based on already submitted bids also allows to analyze the cost of a particular constraint by comparing the objective value of respective scenarios.

We will now discuss a number of side constraints that are important for procurement managers and used during the scenario analysis. Purchasing managers want to set a lower and an upper bound for the quantity a supplier can win overall (V_s^l, V_s^u) in (9), or on a particular item ($V_{i,s}^l$, $V_{i,s}^u$) in (10). Also, they want to limit the overall spend per winner (T_s^l, T_s^u) in (11). In constraint sets (12 & 13) the maximum number of winners L is restricted.

$$V_s^l \leq \sum_{l \in \mathcal{I}} x_{i,s} \leq V_s^u \quad \forall s \in \mathcal{S} \tag{9}$$

$$V_{i,s}^l \leq x_{i,s} \leq V_{i,s}^u \quad \forall i \in \mathcal{I}, \forall s \in \mathcal{S} \tag{10}$$

$$T_s^l \leq \sum_{i \in \mathcal{I}_s} P_{i,s} x_{i,s} - \sum_{d \in \mathcal{D}_s} R_d y_{i,d} + \sum_{m \in \mathcal{M}_s} R_m y_{i,m} - \sum_{l \in \mathcal{L}_s} R_l c_l \leq T_s^u \quad \forall s \in \mathcal{S} \tag{11}$$

$$\textstyle\sum_{i \in \mathcal{I}} x_{i,s} \leq B a_s \quad \forall s \in \mathcal{S} \tag{12}$$

$$\sum_{s \in \mathcal{S}} a_s \leq L \tag{13}$$

$$a_s \in \{0,1\} \quad \forall s \in \mathcal{S}$$

The number of scenarios can be huge, and it would be interesting for procurement managers to know, which side constraints have the biggest impact on total cost. For linear programs such information is provided by the dual variables of respective side constraints. Such dual information is not readily available for mixed integer programs and IP duality is a notoriously difficult topic (Williams 1996, Guzelsoy and Ralphs 2007). One can, however, fix the binary variables to their optimal values and resolve the MIP as a linear program. The resulting duals can then provide useful shadow prices for constraints such as (1), (9), (10), and (11). It can also be helpful to include the right-hand sides of certain constraints as variables, and ask whether there is an award scenario that improves the total cost by a certain percentage. This can be achieved by constraining the objective function value accordingly. Overall, sensitivity analysis can provide valuable feedback for procurement managers during scenario analysis.

6. Experimental Setup

Initial experimental analyses on the winner determination problem in combinatorial auctions have used simple value distributions, which have been criticized mainly for lacking economic justification (Andersson et al. 2000, de Vries and Vohra 2003). Leyton-Brown et al. (2009) put emphasis on the selection of realistic instance distributions for the analysis of a computational problem. Note, that the cost function imposes some structure on the bids generated, which has an impact on the computation time, as we will show. The underlying cost function allows for reasonable bids that resemble those found in real world settings.

6.1. *Cost Functions*

In the following, we will introduce a cost function, which will also be used to generate bids in our experiments. Similar cost functions have been used to estimate cost parameters of companies (Baumol 1987, Evans and Heckman 1984, Stewart 2009). There are many possible treatments for an experimental analysis of an expressive bidding language. The main difference in the bid data which we have collected from the field is the level of scale economies. Therefore, we will mainly focus on the analysis of volume discounts modeling economies of scale, which is reflected in the parametric form of the cost function, which we have chosen. The function allows us to generate different types of instances beyond those that we collected in the field and it allows for a systematic experimental evaluation of different returns to scale.

We will study an economy, where a set $\mathcal{S}$ of suppliers compete for a fixed quantity W_i of one or more items $i \in \mathcal{I}$. Suppliers share the same production technology formally represented by a function.

$$c_s(x_1, ..., x_I) = \quad A_s \lceil \sum_{i \in \mathcal{I}} x_i / W_{i,s} \rceil + \sum_{i \in \mathcal{I}} B_{i,s} \lceil x_i / z_i \rceil + \sum_{i \in \mathcal{I}} \beta_{i,s} (x_i / \gamma_{i,s})^{\rho}$$

The function allows to model very different shapes with convex and concave sections. A_s describes the fixed overhead cost of a supplier s and $W_{i,s}$ the quantity of item i a supplier s is awarded, while $B_{i,s}$ describes the item specific stepwise fixed cost for item i. z models the capacity bound, after which a new machine or plant needs to be used, adding an additional $B_{i,s}$ fixed costs. Note, that with stepwise fixed costs, the cost functions are not continuous any more. $\beta_{i,s}$ describes the slope of a variable cost function for product i, and the exponent ρ is the nonlinear element in the cost function, representing diseconomies of scale. γ is a parameter, which has an effect on the diseconomies of scale: it delays the effect. These parameters can now be systematically varied as will be described in our experimental design in the next section.

6.2. *Experimental Design*

We analyzed two main types of cost functions subsequently referred to as decreasing and u-shaped cost (u-cost). The difference between those types of functions lies in the parameter settings described in Table 1 of our experiments. We have drawn the parameters from a normal distribution, $N(\mu, \sigma^2)$ to generate different cost curves. Among those two types of cost functions, we analyze settings with stepwise fixed costs and without. Figure 1 illustrates random instances of decreasing and u-shaped cost curves with and without stepwise fixed costs. The main dependent variables are run time and total spend.

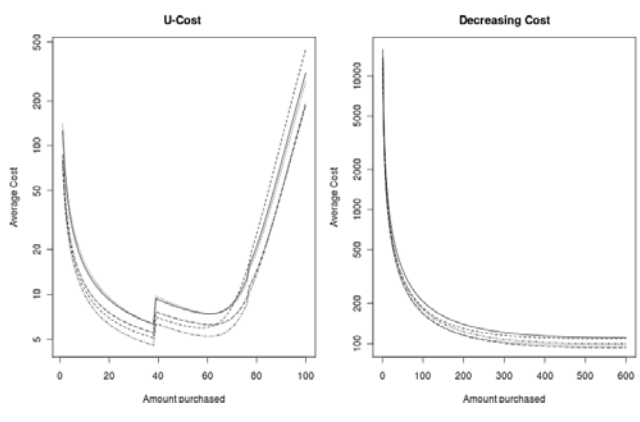

Figure 1: Generation of randomized cost curves for u-cost and a z_i of 30 % and decreasing cost with a z_i of 100 %

First, we have drawn the respective parameter for each item, and then added additional variation for each supplier taking the realization of the initial item-level random variable as the mean of a new supplier-specific random variable. For the capacity z_i, we assumed that either the capacity of a machine is around 30% with a standard deviation of 5% per item and an additional variation per supplier, or we assume there is only a single fixed cost block (100%), but no stepwise fixed costs. The demand of all items is drawn from a uniform distribution between 600 and 1000 items for the decreasing costs curves and 60 and 100 for u-shaped costs.

Variable	description	μ	σ_{item}	$\sigma_{supplier}$
Decreasing cost function				
A_s	supplier fixed overhead costs	0.0	-	0.0
$B_{i,s}$	per item (stepwise) fixed costs	10,000.0	5,000.0	2,500.0
z_i	capacity of production line	30% (100%)	5%	1%
ρ	power of the variable cost function	1.3	0.0	0.01
$\beta_{i,s}$	slope of the variable cost function	10.0	1.25	2.0
$\gamma_{i,s}$	slope delay of the variable cost function	1.0	0.0	0.0
U-shaped cost function				
A_s	supplier fixed overhead costs	0.0	-	0.0
$B_{i,s}$	per item (stepwise) fixed costs	50.0	10.0	30.0
z_i	capacity of production line	30% (100%)	5%	1%
ρ	power of the variable cost function	15	0.0	0.01
$\beta_{i,s}$	slope of the variable cost function	20.0	2.0	0.5
$\gamma_{i,s}$	slope delay of the variable cost function	60.0	0.0	3.0

Table 1: Parameters of the cost curve c_s

For the decreasing cost functions the item fixed costs $B_{i,s}$ are the dominating factor relative to the variable item cost, mainly determined by ρ, effecting the average cost to be stricly decreasing in most instances with a z_i of 100 %. We ran a large number of experiments and had to restrict ourselfes to the most interesting findings. For this reason, we only report experiments with supplier fixed overhead costs $A_s = 0$. Omission of the overhead costs allows for easier sensitivity analysis of the other parameters. For u-shaped cost funtions there are regions where average cost is decreasing and others where it is increasing. We tuned the slope $\beta_{i,s}$ and the slope delay $\gamma_{i,s}$ in the cost function to ensure a larger interval where the average cost is flat.

Based on the underlying total cost function, we generated bids for each supplier. Bid generation is the process of approximating the underlying total cost function with piecewise linear functions as defined for incremental volume discount bids and total quantity discount bids. Note that the piecewise linear functions for a total quantity discount bid need to go through the origin, as the prices are valid for the entire quantity. We assume a direct revelation mechanism, where bidders try to reveal bids reflecting the underlying total cost function truthfully. Note, that if the approximation error gets large and the bid language does not allow to describe the underlying cost function arbitrarily close, this might have an impact on the bidding strategy and bidders might have an incentive to speculate. With only a few predetermined quantity intervals, such approximation errors can become an issue (see supplementary material for details of the bid generation process).

We analyze two treatments: either the number of discounts per item and the corresponding quantity intervals are fixed, or the suppliers can supply an arbitrary number of intervals that is necessary to meet a predefined approximation error ϵ_s^{max}.

In the instances with decreasing cost functions the winner determination becomes trivial by purchasing the entire quantity from a single supplier, if there are no limits on supply. Typically, in such multi-sourcing events the purchasing manager does not allow to purchase the total demand of a single item from a single supplier, or suppliers are unable to do so. Therefore, in all experiments with a decrasing cost function, we included an additional side constraint that 70% of the demand per item could go to a single supplier at a maximum for these instances.

All experimental results are based on IBM's CPLEX 12.1 branch-and-cut solver and were conducted on an Intel Core Duo (3 GHz) with 4 GB of RAM running on 64-bit Suse Linux 11.2. CPLEX provides advanced presolving heuristics. We did, however, not find a significant positive or negative impact of presolving on the run time in general. All results are therefore reported with standard presolving applied.

7. Experimental Results

We will now summarize the results of our experiments with respect to computation time to solve the SQS problem and total spend.

7.1. Analysis of Computation Time for Interactive Scenario Analysis

Computation time is central if SQS is to be solved in scenario analysis, which is the main focus of this paper. For an interactive analysis of different award scenarios during the scenario analysis, the bid evaluation should not take more than a minute. This allows exploring different side constraints in an interactive manner.

In this section, we have therefore set a time limit of 60 seconds, if not stated explicitely, and analyzed the remaining MIP gap for different problem sizes. The integrality or MIP gap described is calculated based on the difference of the best integer solution and the best feasible LP relaxation of all open subproblems. Note that this MIP gap is a worst-case bound as it might turn out that the best feasible LP relaxation is much better than the best integer solution.

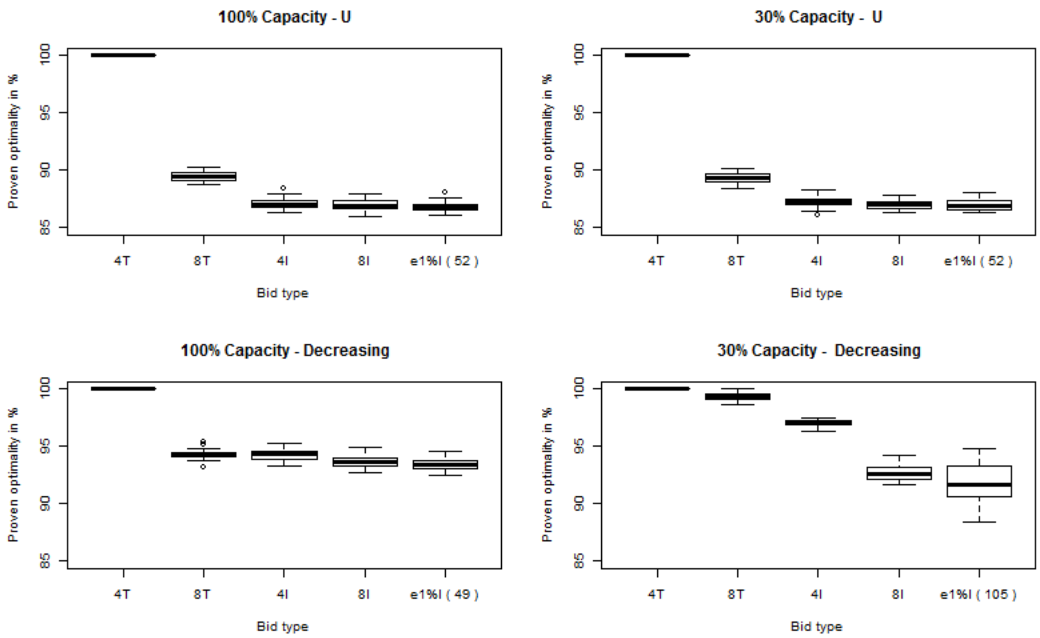

Figure 2: Optimality after 60 seconds for instances with 40 items and 15 suppliers

As a representative graph, Figure 2 illustrates the results for 40 items and 15 suppliers. Smaller instances showed a similar overall pattern. "100% Capacity - U" refers to cost curves without stepwise fixed costs and a u-shaped form. In contrast, "30% Capacity - Decreasing" describes instances where the underlying cost curve has a $\rho = 1.3$, and stepwise fixed costs at around 30% and 60% of the demand.

The average results of 30 experiments for all treatment combinations can be found in Tables **??** to **??** in the supplementary material. The columns 4T, 6T, and 8T summarize the results using total quantity discount bids with 4, 6, and 8 intervals. 4I, 6I, and 8I summarize the results of respective incremental volume discount bids. The string "e0.1%I" refers to incremental volume discount bids with an approximation error ϵ_s^{max} of 0.1% price for the total volume.

Overall, the results of experiments with a 60s time limit reveal that surprisingly large instances could be solved within a minute. We found many real-world problem sizes in high stakes procurement negotiations to have less than 40 items and 15 suppliers and interactive scenario analysis is actually possible with standard MIP solvers. Interestingly, the shape of the cost curve does have a significant impact on the computation time. In this set of experiments, bids based on u-shaped cost functions had a larger MIP gap.

We have then analyzed the runtime of larger problem instances based on u-shaped costs with 10 or 30 suppliers and 10, 30 or 50 items. Instances had either total quantity discounts or incremental discounts and 5 fixed intervals. The MIP gap after 5, 10, 20, 60, 120, and 500 seconds can be found in Table 2. All reported numbers are again average numbers for 30 instances solved. We have analyzed instances with a fixed set of 5 intervals, and instances, where the approximation error ϵ was limited to 1. A value of -1.000 means that there is no MIP gap reported by the solver.

As one can see the incremental volume discount instances with 5 intervals only were not solved to optimality, while the instances with total quantity discounts could be solved to optimality even for larger instances with 30 suppliers and 10 items in 2 minutes. So the MIP gap of the incremental volume discount bids was always higher, when the number of intervals was limited to five. With a limit on the approximation error ϵ to 1%, the number of intervals grows up to more than 80 intervals in case of total quantity discount bids, but only up to around 30 intervals in case of volume discount bids. Interestingly, the instances with only incremental volume discount bids are still harder to solve and the MIP gap is typically larger. So, the structure of the problems with total quantity discounts allows for larger problems to be solved.

			Optimality with time limit					
Suppliers	Items	Prices	5s	10s	20s	60s	120s	500s
Total quantity, 5 intervals								
10	10	5	1.000	1.000	1.000	1.000	1.000	1.000
10	30	5	1.000	1.000	1.000	1.000	1.000	1.000
10	50	5	0.997	0.998	0.999	0.999	1.000	1.000
30	10	5	1.000	1.000	1.000	1.000	1.000	1.000
30	30	5	0.993	0.996	0.997	0.998	0.998	0.998
30	50	5	0.989	0.991	0.993	0.996	0.996	0.997
Incremental, 5 intervals								
10	10	5	0.937	0.944	0.947	0.963	0.958	0.965
10	30	5	0.890	0.909	0.919	0.927	0.923	0.926
10	50	5	0.844	0.867	0.904	0.918	0.916	0.920
30	10	5	0.887	0.909	0.916	0.925	0.921	0.927
30	30	5	0.438	0.765	0.802	0.877	0.881	0.889
30	50	5	0.319	0.512	0.765	0.804	0.840	0.866
Total quantity $\epsilon < 1.0$								
10	10	80.41	0.636	0.680	0.725	0.775	0.812	0.868
10	30	77.36	-1.000	0.509	0.558	0.638	0.681	0.758
10	50	76.65	-1.000	-1.000	0.496	0.582	0.615	0.699
30	10	80.41	-1.000	0.515	0.584	0.646	0.693	0.760
30	30	77.36	-1.000	-1.000	-1.000	0.493	0.529	0.621
30	50	76.92	-1.000	-1.000	-1.000	0.000	0.441	0.549
Incremental $\epsilon < 1.0$								
10	10	28.20	0.595	0.811	0.880	0.937	0.911	0.946
10	30	25.28	-1.000	0.219	0.375	0.412	0.538	0.597
10	50	24.58	-1.000	-1.000	0.328	0.438	0.499	0.576
30	10	28.20	-1.000	-1.000	-1.000	-1.000	0.096	0.168
30	30	25.22	-1.000	-1.000	-1.000	-1.000	-1.000	0.108
30	50	24.52	-1.000	-1.000	-1.000	-1.000	-1.000	-1.000

Table 2: Optimality of the solutions for instances based on a u-shaped cost function (u-cost)

7.2. Analysis of Larger Instances without Time Constraints

Goossens et al. (2007) provided a tailor-made formulation for total quantity discount bids, as well as computational results on randomly generated test instances. In contrast, $\mathcal{L}_{ESS}$ provides bidders with more flexibility and allows for other types of volume discounts and various spend conditions. Typically, more flexibility and expressiveness comes at the cost of computational complexity. It is interesting to see, if the expressiveness of $\mathcal{L}_{ESS}$ comes at a large computational cost.

Therefore, in another set of experiments, we used the synthetic bids, which were kindly provided by Goossens et al. (2007) for a comparison. These experiments are limited to bids with total quantity discounts only. In their instances with 40 items, Goossens et al. (2007) generated an upperbound-increase from one interval to the next, which was a random number between 10,000 and 50,000, while for instances with 100 items, the upperbound increase was a random number between 10,000 and 100,000. The results of the formulation in Goossens et al. (2007) with a branch-and-cut approach and with the SQS formulation can be found in Tables **??** and **??**. We focus on the branch-and-cut results, since those provided the best results for larger instances. Note that we have used CPLEX version 12.1, while Goossens et al. (2007) used version 8.1.

Interestingly, much larger instances could be solved to optimality with this set of bids in seconds, and the differences in runtime between the results reported by Goossens et al. (2007) and the results of SQS were small. The results for the base case in were a slightly slower, while the results for the more-for-less scenario in Table were actually faster (see Table **??** and **??** in the supplementary material). The more-for-less scenario aimed for optimal solutions with free disposal of additional quantity, while the base case did not.

Obviously, the structure of the bids has a significant impact on the runtime. The bids generated based on our multi-product cost function were considerably harder to solve. In particular, those instances without stepwise fixed costs and a slope of $\rho = 1.3$. We assume that smooth cost functions generate a lot of solutions with similar objective function value, a type of symmetry problem. The structured instances in Goossens et al. (2007) include economies of scale, where intervals with more quantity have lower prices than intervals with less quantity, whereas the cost functions in our paper also include diseconomies of scale. Also the demand can make a difference. If the demand is increased, runtimes can increase, because there are more possible quantities to purchase and intervals.

In summary, the predictive quality of such runtime experiments depends on the structure of the bids and the respective scale economies in a market. However, our results show that instances of practically relevant size can be solved to optimality with very different types of bids.

7.3. Impact of Additional Side Constraints

Additional side constraints discussed in section 5 can have an impact on the runtime of the winner determination, and they are particularly important to scenario analysis. As an example, we will report on variations of the allocation constraint (10), in which a purchasing manager can specify bounds on the quantity per supplier and item.

We have used a problem with 20 items, 10 suppliers, a ρ of 1.3, no stepwise fixed costs, and varied the upper bounds $V^u_{i,s}$. In Figure 3 we report the time it took to solve the instances. An exhaustive sensitivity analysis of all possible side constraints and different right-hand sides is out of scope of this paper, but this example illustrates that if additional side constraints become binding, this can have a significant impact on computation time.

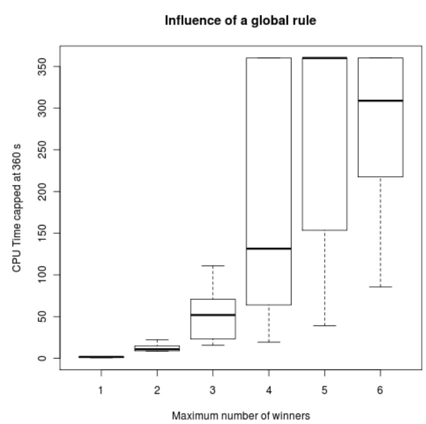

Figure 3: Sensitivity to side constraints on quantity per item $V^u_{i,s}$

7.4. Description Length

We have discussed in section 3.1 that apart from the expressiveness of a bid language, the description length matters. Figure 4 shows how the number of intervals in the SQS formulation increases with a decreasing maximal approximation error ϵ_s^{max}. The required number of discount intervals is much higher for total quantity bids. The large number of intervals needed to describe a cost function with low approximation error can make bidding impractical. Note that in an incentive compatible mechanism such as the Vickrey-Clarke-Groves mechanism, bidding truthfully might not be a dominant strategy, unless the bidding language allows describing the underlying cost function arbitrarily close.

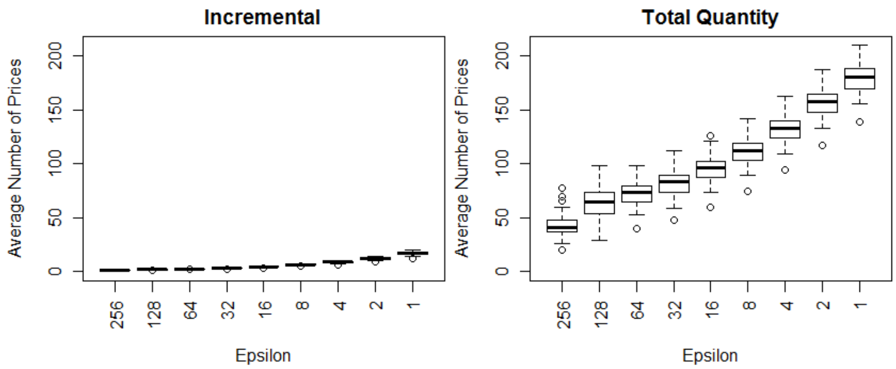

Figure 4: Description length of payment modifiers

7.5. Total Spend Comparison

Finally, we will discuss the impact of different volume discount types on the total cost of the purchasing organization. Again, we assume a direct revelation mechanism, and bidders submit their bid in a way that approximates their true costs as close as possible, either restricted by the number of intervals or a predefined approximation error. We had suppliers submit a single price quote for each item in a split-award auction. The same suppliers also submit bids in an auction with only total quantity discount bids, and in an auction with only incremental volume discount bids. For the latter two auction types, we also distinguished settings with a fixed set of 5 intervals, or an approximation error ϵ of 1.0%. All bids were determined based on the same underlying cost function. The results are summarized in Table 3. Each row summarizes the result of the split award auction in absolute numbers on the right, and the result of a particular volume discount auction relative to the cost of the split-award auction in percentage values after 5, 10, 20, 60, 120, and 500 seconds similar to the results presented on optimality in Table 2.

We will first look at the spend with a fixed number of 5 intervals. If the MIP gap is high, then also the spend with total quantity discount bids and incremental volume discount bids can be much worse than with simple split award auctions. Therefore, we will mainly look at the column with 500s, where the MIP gap was low. Interestingly, in the setting with total quantity bids and 5 intervals, the spend achieved was higher than that of split award auctions, although the smaller instances could be solved to optimality. The reason for this is the bad approximation. In contrast, incremental volume discount bids achieved a lower total cost compared to split award auctions for all problem sizes already after 120 seconds. This was the case, even though the problems could not be solved to optimality within 500 seconds.

In situations, where the approximation error ϵ was limited to 1%, total quantity discount bids led to cost savings compared to split-award auctions in small instances. Larger instances with 30 suppliers and 30 items led to a large MIP gap even after 500 seconds and the results were worse than those of a split award auction. Also in this setting, the incremental volume discounts led to lower total cost compared to split-award auctions for all problem sizes after 500 seconds. Also, after 500 seconds the total cost was always lower than with total quantity discount bids.

8. Summary and Conclusions

We have suggested a bidding language for markets with economies of scale and scope and a respective mixed integer program to solve the resulting supplier quantity selection (SQS) problem. The bidding language is considerably

		Spend in % of split awards with time limit						Split award
Suppliers	Items	5s	10s	20s	60s	120s	500s	(100%)
Total quantity 5 intervals								
10	10	101.13	100.94	100.94	100.02	100.94	100.94	5441
10	30	101.22	101.22	101.22	101.58	101.22	101.22	15965
10	50	102.28	102.20	102.19	102.45	102.19	102.19	26243
30	10	101.44	101.44	101.44	101.44	101.44	101.44	5110
30	30	102.75	102.59	102.57	103.80	102.56	102.56	14860
30	50	102.79	102.69	102.48	102.74	102.27	102.27	24688
Incremental 5 intervals								
10	10	93.65	93.47	93.49	93.27	93.45	93.46	5441
10	30	97.31	95.50	94.67	94.84	94.48	94.40	15965
10	50	103.54	100.72	96.70	95.87	95.62	95.46	26243
30	10	96.76	94.86	94.37	92.63	94.28	94.21	5110
30	30	109.45	109.09	105.02	97.81	95.97	95.62	14860
30	50	115.53	113.37	107.40	103.07	97.98	95.85	24688
Total quantity $\epsilon = 1.0$								
10	10	93.59	93.60	93.60	93.52	93.25	87.89	5441
10	30	29311.83	95.37	95.37	95.77	95.36	95.17	15965
10	50	31571.88	29585.34	96.65	97.44	96.64	96.64	26243
30	10	117653.66	97.36	97.34	96.49	97.36	96.96	5110
30	30	92869.85	92869.85	92869.85	102.60	102.30	102.29	14860
30	50	82807.65	83589.52	83589.52	92582.93	103.08	103.08	24688
Incremental $\epsilon = 1.0$								
10	10	87.72	87.73	86.35	86.09	86.27	86.15	5441
10	30	374.71	132.40	94.47	90.68	88.08	87.69	15965
10	50	450.39	335.86	123.02	95.08	89.94	89.19	26243
30	10	930.91	184.22	96.96	89.46	87.88	87.29	5110
30	30	1449.13	1277.17	1350.74	162.00	103.73	90.97	14860
30	50	1675.84	1433.83	1333.95	696.10	150.84	94.08	24688

Table 3: Comparison of spend using a u-shaped cost function

more expressive than what has been discussed in the literature so far and includes incremental volume discounts, total quantity discounts, lump sum discounts, and a variety of conditions defined on spend and quantity of selected items. While both, incremental volume discount bids and total quantity discount bids, have been described in the literature and are used in procurement practice, there has not been a thorough comparison among those discount policies as of yet. This is the first paper, to use different types of cost functions to generate bids, which allowed us to analyze not only computation times, but also total spend of different discount policies.

Our results show that that realistic problem sizes can be solved in a matter of minutes, but that problems with only incremental volume discount bids are harder to solve than those with only total quantity discount bids. If a supplier wants to approximate his true cost function with total quantity discount bids closely, this leads to a much larger and number of discount intervals. There were several situations, where the results of simple split-award auctions with simple price quotes lead to lower total cost than those of more advanced bidding languages. The reasons are either bad approximations of the cost curve or the inability to find the cost-minimal solution within acceptable response times. For example, if suppliers only used a few total quantity discount intervals, this led to much higher total cost for the buyer. However, we have also shown that significant savings can be achieved with compact bidding languages compared to split award auctions, if bidders are able to approximate their cost functions well.

In summary, a procurement manager needs to take care that the bidding language provides enough flexibility so that bidders can describe their cost structures arbitrarily close. At the same time bids should have low description length, such that suppliers are only forced to specify a few parameters and not hundreds of numbers. Also, a procurement manager should make sure that the number of bids in an application is such that he can expect to solve the problem instances to optimality. The results of our analysis should provide an better understanding under which circumstances compact bidding languages can be used in procurement practice. Mechanism design questions have been outside the scope of this paper, and remain fruitful questions for future research in this area.

References

Abrache, J., Bourbeau, B., Crainic, T. G., Gendreau, M., 2004. A new bidding framework for combinatorial e-auctions. Computers & Operations Research 31 (8), 1177 – 1203.

Andersson, A., Tenhunen, M., Ygge, F., 2000. Integer programming for combinatorial auction winner determination. In: Fourth International Conference on Multiagent Systems (ICMAS).

Anton, J., Brusco, S., Lopomo, G., March 2009. Coordination in split award auctions with uncertain scale economies: Theory and data. Tech. rep., Duke University.

Anton, J., Yao, D. A., 1992. Coordination in split award auctions. Quarterly Journal of Economics 107, 681–707.

Baumol, W. J., 1987. Microtheory: Applications and Origins, 1st Edition. MIT Press, Cambridge, MA, USA.

Benisch, M., Sadeh, N., Sandholm, T., 2008. A theory of expressiveness in mechanisms. In: 23rd AAAI Conference on Artificial Intelligence.

Bichler, M., Davenport, A., Hohner, G., Kalagnanam, J., 2006. Industrial procurement auctions. In: Cramton, P., Shoham, Y., Steinberg, R. (Eds.), Combinatorial Auctions. MIT Press.

Boutilier, C., Hoos, H. H., 2001. Bidding languages for combinatorial auctions. In: 17th International Joint Conference on Artificial Intelligence (IJCAI) 2001. Washington, USA, pp. 1211–1217.

Chaudhry, S., Forst, F., Zydiak, J. L., 1993. Vendor selection with price breaks. European Journal of Operaitonal Research 70, 52–66.

Crama, Y., Pascual, R., Torres, A., 2004. Optimal procurement decisions in the presence of total quantity discounts and alternative product recipes. European Journal of Operational Research 159, 364–378.

Cramton, P., Shoham, Y., Steinberg, R., 2006. Introduction to combinatorial auctions. In: Cramton, P., Shoham, Y., Steinberg, R. (Eds.), Combinatorial Auctions. MIT Press, Cambridge, MA.

Dantsin, E., Eiter, T., Gottlob, G., Voronkov, A., 2001. Complexity and expressive power of logic programming. ACM Computing Surveys 33, 374–425.

Davenport, A., Kalagnanam, J., 2000. Price negotiations for procurement of direct inputs. In: IMA ”Hot Topics” Workshop: Mathematics of the Internet: E-Auction and Markets. Vol. 127. Minneapolis, USA, pp. 27–44.

de Vries, S., Vohra, R., 2003. Combinatorial auctions: A survey. INFORMS Journal of Computing 15 (3), 284–309.

Eso, M., Ghosh, S., Kalagnanam, J., Ladanyi, L., 2001. Bid evaluation in procurement auctions with piece-wise linear supply-curves. Technical report rc22219, IBM T.J. Watson Research Center.

Evans, D., Heckman, J., 1984. A test for subadditivity of the cost function with an application to the bell system. American Economic Review 74 (4), 615–623.

Gallien, J., Wein, L., 2005. A smart market for industrial procurement with capacity constraints. Management Science 51, 76–91.

Gartner, May 2008. Magic quadrant for sourcing application suites. Tech. rep., Gartner Consulting.

Giunipero, L., Carter, P., Fearon, H., 2009. The role of optimization in strategic sourcing. Tech. rep., CAPS Research.

Goossens, D. R., Maas, A. J. T., Spieksma, F., van de Klundert, J. J., 2007. Exact algorithms for procurement problems under a total quantity discount structure. European Journal of Operational Research 178, 603–626.

Guzelsoy, M., Ralphs, T., 2007. Duality for mixed-integer linear programs. The International Journal of Operations Research 4, 118–137.

Hohner, G., Rich, J., Ng, E., Reid, G., Davenport, A., Kalagnanam, J., Lee, H., An, C., 2003. Combinatorial and quantity discount procurement auctions with mutual benefits at mars, incorporated. Interfaces 33 (1), 23–35.

Hooker, J. N., 2009. A principled approach to mixed integer/linear problem formulation. In: INFORMS Computing Society Conference. pp. 79–100.

Jucker, J. V., Rosenblatt, M. J., 1985. Single period inventory models with demand uncertainty and quantity discounts: Behavioral implications and a new solution procedure. Naval Research Logistics Quarterly 32 (4), 537–550.

Katz, P., Sadrian, A., Tendrick, P., 1994. Telephone companies analyze price quotations with bellcore's pdss software. Interfaces 24, 50–63.

Lee, H. L., Rosenblatt, M. J., 1986. A generalized quantity discount pricing model to increase supplier's profit. Management Science 32, 1177–1188.

Leyton-Brown, K., Nudelman, E., Shoham, Y., 2009. Empirical hardness models: Methodology and a case study on combinatorial auctions. Journal of the ACM 56, 1–52.

Munson, C. L., Rosenblatt, M. J., 1998. Theories and realities of quantity discounts: an exploratory study. Production and Operations Management 7 (4), 352–369.

Nisan, N., 2006. Bidding languages. In: Cramton, P., Shoham, Y., Steinberg, R. (Eds.), Combinatorial Auctions. MIT Press, Cambridge, MA.

Nisan, N., Segal, I., 2001. The communication complexity of efficient allocation problems. In: DIMACS workshop on Computational Issues in Game Theory and Mechanism Design. Minneapolis, MI.

Papadimitriou, C. H. (Ed.), 1993. Computational Complexity. Addison Wesley.

Perry, M., Sakovics, J., 2003. Auctions for split-award contracts. Journal of Industrial Economics 51, 215–242.

Sandholm, T., 2008. Expressiveness in mechanisms and its relation to efficiency: Our experience from $40 billion of combinatorial multi-attribute auctions, and recent theory. In: Proceedings of the 2nd International Workshop on Computational Social Choice.

Silverson, R., Peterson, E. A., 1979. Decision systems for inventory management and production planning. John Wiler and Sons, New York.

Stewart, K. G., 2009. Non-jointness and scope economies in the multiproduct symmetric generalized mcfadden cost function. Journal of Productivity Analysis 32 (3), 161–171.

van de Klundert, J., Kuipers, J., Spieksma, F., Winkels, M., 2005. Selecting telecommunication carriers to obtain volume discounts. Interfaces 35, 124–132.

Williams, H. P., 1996. Duality in mathematics and linear and integer programming. Journal of Optimization Theory and Applications 90, 257–278.