Efficiently Generating k-Best Solutions for Procurement Auctions

Andrew Byde, Terence Kelly
Digital Media Systems Laboratory
HP Laboratories Bristol
HPL-2006-145
October 19, 2006*

auctions, combinatorial optimization, decision support, e-commerce, graphs, algorithms, procurement

This paper considers the problem of generating *k* cheapest solutions to a class of procurement auction winner determination problems. A computationally efficient solution to this problem would enable a fundamentally new approach to decision support for procurement, based on "mining" the *k* cheapest solutions. However, previous methods do not scale in crucial problem-size parameters, e.g., the number of sellers. Our novel algorithm achieves an exponential performance improvement over previous methods, and scales polynomially in *all* natural measures of problem size. By efficiently computing *k*-cheapest solutions, our algorithm qualitatively expands the practical applicability of the data-exploration approach to procurement decision support.

* Internal Accession Date Only
Approved for External Publication

Efficiently Generating k-Best Solutions for Procurement Auctions

Andrew Byde, Terence Kelly

Abstract

This paper considers the problem of generating k cheapest solutions to a class of procurement auction winner determination problems. A computationally efficient solution to this problem would enable a fundamentally new approach to decision support for procurement, based on "mining" the k cheapest solutions. However, previous methods do not scale in crucial problem-size parameters, e.g., the number of sellers. Our novel algorithm achieves an exponential performance improvement over previous methods, and scales polynomially in *all* natural measures of problem size. By efficiently computing k-cheapest solutions, our algorithm qualitatively expands the practical applicability of the data-exploration approach to procurement decision support.

1 Introduction

Solving the winner determination problem (WDP) in a large procurement auction is non-trivial, both because of the computational complexity of deciding which is the cheapest subset of bids offered, but also more importantly because the cheapest is not always the best: For business reasons, procurement executives have preferences over assignments of contracts to suppliers that are not easy to state and that sometimes involve non-linear WDP solution attributes. The procurement decision problem therefore cannot be easily incorporated into *any* optimization procedure.

Conventional approaches to decision support for such situations extend optimization frameworks. Recently, however, a fundamentally different approach has been proposed that casts the procurement decision problem as one of *data exploration* rather than optimization [Kelly and Byde, 2006a]. This method transforms the procurement auction WDP into a shortest-paths problem, and employs k-shortest paths algorithms to generate k-cheapest solutions to the WDP. The purpose of this exercise is to illuminate the competitive landscape and to help the procurement executive understand the tradeoffs inherent in her constraints and preferences and in supplier bids. The k-cheapest solutions can be mined to reveal the implicit costs of constraints on WDP solutions and can support visualizations detailing the dependence of cost on other solution attributes. If the decision-maker has ordinal preferences over solution attributes, the k-cheapest solutions can also be condensed via dominance pruning to obtain a compact Pareto frontier of undominated WDP solutions.

The main problem with previous work on this approach is scalability. The k-cheapest solutions algorithm of [Kelly and Byde, 2006a] scales well in the number of items (distinct types of goods) in a procurement auction, but it requires time and memory exponential in the number of sellers and in the number of fractions of each item that a seller can supply. This paper solves both problems. Our new algorithm achieves polynomial time and memory complexity via two fundamental improvements: we transform the auction WDP into a far more compact graph than the one used in prior work, and we exploit state-of-the-art k-shortest paths algorithms. Like [Kelly and Byde, 2006a], we can encode global constraints on the WDP solution into our graph, ensuring that the k-cheapest solutions we generate all satisfy the constraints. In summary, our method offers all of the advantages of the earlier approach but suffers none of its computational drawbacks.

The new graph construction is relatively straightforward in the case where there are no constraints other than feasibility imposed on the solution, so we describe this construction first, in Section 3. Section 4 presents the more complicated case of a global constraint on auction WDP solutions. Potentially *any* constraint could be encoded into the structure of the graph, but some kinds of constraints yield impractically large graphs. Therefore to demonstrate the utility of the method, we give a few examples of useful constraints that are compatible with our method at low computational cost.

2 Procurement Auctions

Auctions play an increasingly important role in procurement and bring several benefits. For example, over 20% of Sun Microsystems' expenditure in recent years has occurred via procurement auctions, and Sun claims to have saved \$300 million as a result [Hannon, 2004]. Less tangible yet equally important benefits include speed and agility: auctions can reduce procurement decision times from months to days.

For a large buyer, procurement is a *strategic* situation. The cheapest way of acquiring needed goods is almost never the best, and complex business considerations typically affect procurement decisions. For example, the buyer might want to restrict the number of suppliers awarded contracts, and it might want to spread its expenditure evenly among chosen suppliers. However such desiderata are rarely hard constraints; the buyer might waive her own guidelines in exchange for large savings. Sophisticated decision support is required because procurement executives typically cannot articulate their strategic business concerns as a formal constrained optimization problem. The decision-maker must understand the cost implications of her preferences and the competitive landscape inherent in supplier bids.

Our formal framework, described in this section, supports important aspects of real-world procurement: Most procurement occurs through *multi-item* auctions, in which several types of goods are acquired simultaneously. *Multi-sourcing* means that several sellers can supply fractions of the buyer's demand for a single type of good. Furthermore, suppliers' production capacities and inventory levels can lead to *volume discounts* and *volume surcharges* in bids.

2.1 Definitions and Notation

Let I denote the number of *items* (distinct types of goods) that the buyer wishes to acquire; the overall procurement auction consists of I sub-auctions that are cleared simultaneously. Global granularity parameter Q specifies the number of *quantiles* (shares of an item) that bids offer to supply. If $Q=4$, for instance, then bids offer to supply 25%, 50%, 75%, or 100% of the total number of demanded units of each item. Let S denote the number of sellers; we assume that $S \geq 2$. Let $B_{is}(q)$ be the amount of money that seller s demands for q quantiles of item i; implicitly, $B_{is}(0)=0$.

For a given procurement auction we will construct a weighted directed acyclic graph G with special source and sink vertices $\mathtt{s}$ and $\mathtt{t}$ such that the k-shortest paths from $\mathtt{s}$ to $\mathtt{t}$ correspond to the k cheapest solutions to the WDP. We will describe this graph as a *compact outcome graph*, because its paths represent outcomes to the WDP, and because, relative to the outcome graphs specified in [Kelly and Byde, 2006a], it has relatively few vertices and edges.

Published work such as [Eppstein, 1998] demonstrates that given a graph G with n vertices and m edges, the k shortest paths can be calculated implicitly in time $O(m+n\log n+k)$. The $n\log n$ term comes from Dijkstra's algorithm [Dijkstra, 1959] for constructing the tree of shortest paths from $\mathtt{s}$ to each other vertex; if such a tree has already been constructed Eppstein's algorithm takes time $O(m+n+k)$. Happily, since our graph is directed and acyclic, it is possible to construct the shortest-path tree in time $O(m)$, so that our graph's k shortest paths can be found implicitly in time $O(m+n+k)$.[1]

Graph construction is significantly simpler in the unconstrained case, so we first discuss this case; a full discussion of the constrained case is presented in Section 4.

3 Unconstrained Graph Construction

The construction is via a sequence of sub-graphs G_i with source $\mathtt{s}_i$ and $\mathtt{t}_i$, such that each path from $\mathtt{s}_i$ to $\mathtt{t}_i$ corresponds uniquely to a solution to the i^{th} single-item auction. We will describe the construction of G_i in the next section, but note that if we have such a sequence of graphs then we can form a suitable full graph G by chaining the G_i together. To be precise, for each $i<I$ we replace each vertex $\mathtt{t}_i$ with $\mathtt{s}_{i+1}$. The source of G is $\mathtt{s}=\mathtt{s}_1$ and its sink is $\mathtt{t}=\mathtt{t}_I$. Any path from $\mathtt{s}$ to $\mathtt{t}$ in G must have I sub-paths, passing from $\mathtt{s}_i$ to $\mathtt{t}_i$ for each i, and thus must correspond to an outcome in each single-item auction; clearly the length of the full path is the sum of the sub-path lengths, which replicates the cost-relationship of full outcomes and single-item outcomes. This gives the relationship between paths and auction outcomes that we need.

3.1 Sub-graph Construction

Let the vertices of G_i be the set of pairs (s,q) (for $s=1,\dots,S$ and $q=0,\dots,Q$) with the special source and sink vertices $\mathtt{s}_i$ and $\mathtt{t}_i$ added.

Edges represent assignments of a number of quantiles to a seller: for each quantile $q_1=0,\dots,Q$ add an edge from $\mathtt{s}_i$ to $(1,q_1)$ with label q_1 and length $B_{i1}(q_1)$. Likewise for each seller $s=2,\dots,S-1$, quantiles $q\leq Q$ and $q_s\leq Q-q$, connect the vertex $(s-1,q)$ to the vertex $(s,q+q_s)$ via an edge with label q_s and length $B_{is}(q_s)$. Finally, connect each vertex $(S-1,q)$ to $\mathtt{t}_i$ via an edge with label $Q-q$ and length $B_{iS}(Q-q)$. See Figure 1 for an example in the case $I=S=Q=3$.

Each path in G_i from $\mathtt{s}_i$ to $\mathtt{t}_i$ has a sequence of labels $q_1,q_2,\dots,q_S$. An important fact about these label sequences is summarized in the following proposition:

Proposition 3.1 *Each path in G_i from $\mathtt{s}_i$ to $\mathtt{t}_i$ corresponds, via the edge labeling, to a non-negative integer solution* $\mathbf{q}$ *of the equation*

$$\sum_{s=1}^{S} q_s = Q, \tag{1}$$

and vice-versa. Furthermore the length of this path is exactly the cost to the buyer of the outcome $\mathbf{q}$.

First we show that the labels in each path satisfy the equation; then we show that for any solution $q_1,\dots,q_S$ to (1) there is a path with labels q_s. Suppose then that we have a path from $\mathtt{s}_i$ to $\mathtt{t}_i$ with labels q_s. The first edge starts at $\mathtt{s}_i$ and therefore must end at $(1,q_1)$. In general if the j^{th} edge ends at (j,ξ_j), then the next vertex must be $(j+1,\xi_j+q_{j+1})$. It is easy to see from this that the sequence of vertices must be $\mathtt{s}_i$, $(1,q_1)$, $(2,q_1+q_2)$, $(3,q_1+q_2+q_3)$ and so on up to the penultimate vertex $(S-1,\sum_{s=1}^{S-1}q_s)$, which is connected to $\mathtt{t}_i$. A vertex of the form $(S-1,q)$ is connected to $\mathtt{t}_i$ only if it has label $q_S=Q-q$. It follows that a path from $\mathtt{s}_i$ to $\mathtt{t}_i$ has labels $q_1,q_2,\dots,q_S$ only if

$$\begin{aligned} Q-\textstyle\sum_{s=1}^{S-1}q_s &= q_S \\ \Rightarrow \qquad \textstyle\sum_{s=1}^{S}q_s &= Q. \end{aligned}$$

Suppose on the other hand that $q_1,\dots,q_S$ is a solution to (1). Since the q_s sum to Q it is clear that $q_s\leq Q-\sum_{j<s}q_j$, which implies that the sequence of vertices $\mathtt{s}_i$, $(1,q_1)$, $(2,q_1+q_2)$, $\dots$, $(s,\sum_{j<s}q_j)$, $\dots$, $(S-1,\sum_{j<S}q_j)$ form a path in G_i with edge labels $q_1,\dots,q_{S-1}$. The final

[1] To extract explicit representations takes additional time proportional to the number of edges in each path, for which we will derive good bounds in Section 3.3.

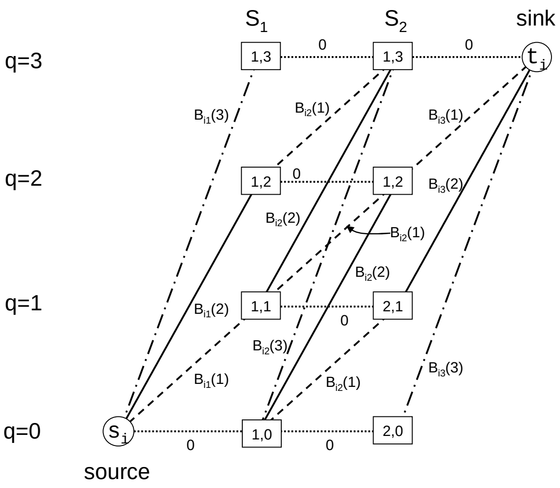

Figure 1: Individual-item solutions graph for $I = S = Q = 3$, as described in Section 3.2. Each edge is directed, left to right. Finely dashed edges have label $q = 0$; roughly dashed edges have label $q = 1$; solid edges have label $q = 2$ and dash-cut edges have label $q = 3$. Edge lengths are shown next to each edge.

edge can be added because, as required, $q_S = Q - \sum_{j<S} q_j$, as a direct consequence of (1).

Since the length of the edge with label q_s is $B_{is}(q_s)$ it is clear that the length of the path corresponding to the outcome $\mathbf{q} = (q_1, \ldots, q_S)$ is $\sum_s B_{is}(q_s) = \mathrm{cost}_i(\mathbf{q})$, i.e., the cost of the outcome to the buyer. □

3.2 Example

The graph for $S = Q = I = 3$ is shown in Figure 1. Each edge corresponds to assigning 0, 1, 2 or 3 of the 3 available quantiles to a particular seller; the "length" of each edge is the corresponding bid $B_{is}(q)$. The reader can verify that there are exactly ten paths from $\mathtt{s}_i$ to $\mathtt{t}_i$ (edges are directed, left to right), corresponding to the ten ways of allocating 3 quantiles among 3 sellers.

3.3 Complexity

In [Kelly and Byde, 2006a] a similar construction based on sub-graphs for each single-item auction was undertaken, in which the sub-graph consisted of the source and sink nodes, with an edge for each possible outcome to the i^{th} single-item auction. It is clearly only necessary to use the k shortest such edges, so the previous best bound on the number of edges in the i^{th} sub-graph was $\min(R(S,Q),k)$, where $R(S,Q) = (Q+S-1)!/Q!(S-1)!$—the number of solutions to an individual-item sub-auction—scales exponentially with respect to S and Q.

In our new construction there are $(Q+1)(S-1)+2$ vertices in each sub-graph G_i, and the number of edges is

$$(S-2)\left(\frac{(Q+1)(Q+2)}{2}\right) + 2(Q+1).$$

In both cases the important fact is the degree of the polynomial bound. When aggregated over all items the number of vertices is $O(ISQ)$, and the number of edges is $O(ISQ^2)$. This is a significant improvement on the previous exponential bound, at least in the case where k is large compared to S and Q.

To extract explicit representations takes additional time proportional to the number of edges in each path [Eppstein, 1998]. Each path in the compact outcome graph has exactly $(S+1)I$ edges by construction, so the full time complexity of explicitly listing the k shortest paths in a compact outcome graph is $O(IS(Q^2+k))$, in which the second extraction-related term is the dominant one for practical cases.

3.4 Alternative Construction

An alternative to constructing the graph by linking sub-graphs together is to define it all at once. This approach is less intuitive, but closer to the way in which the compact outcome graph is defined for the general constrained case, so we present it here to familiarize the reader with the approach.

The compact outcome graph is defined directly to have a vertex for each triple (i,s,q), with the caveat that when $s = S$ we must have $q = Q$. We also have an extra source vertex $\mathtt{s} = (0,S,Q)$ and identify (I,S,Q) as the sink node $\mathtt{t}$.

1. We connect (i,s,q) to $(i,s+1,q+q_s)$ for each $s < S-1$ and $q_s \leq Q-q$. These are the majority of edges, and correspond to the assignment of quantity q_s of item i to seller s.
2. We connect $(i,S-1,q)$ to (i,S,Q) whenever $0 < i \leq I$. These edges are the assignments to the last seller S, of an appropriate quantity to ensure that all of item i is sold.
3. We connect (i,S,Q) to $(i+1,1,q_1)$ for each $0 \leq i < I$ and $q_1 \leq Q$. These edges correspond to the assignment of quantity q_1 of item $i+1$ to the first seller, and are equivalent to the "linking" of subgraphs for items i and $i+1$.

The labels and lengths of edges follow the obvious pattern: an edge from (i,s,q) to $(i,s+1,q+q_s)$ is labeled with quantity q_s and has length $B_{is}(q_s)$; edges from (i,S,Q) to $(i+1,1,q_1)$ have label q_1 and length $B_{i+11}(q_1)$.

4 Constrained Case

In this section we consider the case where only some sub-collection of the set of all global outcomes is acceptable. Usually this set will be expressed in the form of a rule, such as "outcomes including at most 3 sellers." As in Section 3.4 we construct a graph with vertices indexed by item-seller-quantity triples, but now with an additional index, the "state" of the vertex, which represents the state of the solution so far constructed. By restricting those edges that are added to G on the basis of their state we can exclude paths that are bound to violate the constraint. To be precise, suppose X is any finite set with a special element $\{*\}$ (for example, think of $\{*,1,2,\ldots,N\}$). X will represent the intermediate states in the evaluation of the acceptability of an outcome as the outcome is constructed by assigning quantities to suppliers as in Section 3.

A directed graph G is a *constrained compact outcome graph* if the following conditions hold:

1. The set of vertices of G is the set of quadruples (i,s,q,x) with the constraint that only $q = Q$ is allowed when $s = S$, together with the special vertices $\mathtt{s} = (0,S,Q,*)$, and $\mathtt{t} = (I+1,1,0,*)$.
2. Every edge in G is labeled with a quantity $0 \le q_s \le Q$.
3. Every edge in G starting at (i,s,q,x) with label q_s either goes to $(i,s+1,q+q_s,x')$ if $s < S$, or goes to $(i+1,1,q_s,x')$ if $s = S$, where x' is some element of X.
4. Every edge ending at vertex (i,s,q,x), $i \le I$ with label q_s has length $B_{is}(q_s)$; every edge ending at vertex $\mathtt{t}$ has length 0.
5. There is at most one edge starting at (i,s,q,x) with each label q_s.

If G is a constrained compact outcome graph then paths from $\mathtt{s}$ to $\mathtt{t}$ correspond uniquely to outcomes to the global auction by identification of the edge labels with assignments of quantities of suitable items to sellers. The truth of this statement follows from similar reasoning to the proof of Proposition 3.1: first of all from (3) the sequence of intermediate vertices in any path from $\mathtt{s}$ to $\mathtt{t}$ must have i and s components of the form $(1,1)$, $(1,2)$, ..., $(1,S)$, $(2,1)$, $(2,2)$, ..., $(2,S)$, ..., (I,S). If $\xi_{i,s}$ is the corresponding sequence of quantile components then since $\xi_{i,S} = Q$, from (1), the differences

$$q_{i,s} = \begin{cases} \xi_{i,s} - \xi_{i,s-1} & \text{if } s > 1 \\ \xi_{i,1} & \text{if } s = 1 \end{cases}$$

must sum over s to Q for each i. From (3) these quantile differences must be equal to the edge labels. Since the quantities for each item sum to Q, they form an outcome to the global auction: $o_i(s) = q_{i,s}$. Furthermore since, from (5) there is at most one edge with each label emerging from each vertex, there is at most one path corresponding in this way to a given outcome. By (4) the length of a path from $\mathtt{s}$ to $\mathtt{t}$ is precisely the cost of the corresponding outcome.

A constrained compact outcome graph G *represents* a global constraint if the only outcomes that correspond to paths in G from $\mathtt{s}$ to $\mathtt{t}$ obey the constraint, and if every outcome obeying the constraint corresponds to a path in G from $\mathtt{s}$ to $\mathtt{t}$. By making the set X large enough we can in fact represent every constraint by a suitable constrained compact outcome graph; the challenge is not merely to construct constrained compact outcome graphs representing a constraint, but to do so efficiently.

4.1 Complexity

As in Section 3.3 it is easy to see from (5) that the out-degree of each vertex in a constrained outcome graph is at most $Q+1$, although the in-degree of a vertex could be larger. In general it is clear that the size of X determines the complexity of the graph. There are no more than $IS(Q+1)|X|$ vertices, and no more than $IS(Q+1)^2|X|$ edges, so as before the complexity of computing the k shortest paths is $O(ISQ^2)$. The in-degree of a particular vertex is at most $(Q+1)|X|$.

The complexity here is the same as before, except for a factor of $|X|$, giving a bound for explicit extraction of the k shortest paths of $O(IS|X|(Q^2+k))$.

4.2 Example Constrained Compact Outcome Graphs

Prior work in [Kelly and Byde, 2006b] describes three types of constraints that can be efficiently represented using constrained outcome graphs:

1. Constraints based on the number or set of sellers *included* in the global solution, in the sense that they supply a non-zero quantity of one or more item.
2. Constraints based on the total number of quantiles bought from one or more sellers.
3. Constraints based on the total value of the allocation to one or more sellers. In this case the constrained compact outcome graph formalism greatly reduces the number generated, but usually cannot guarantee that *only* outcomes satisfying the constraint are generated.

Since the third class of examples is essentially equivalent to the second, we shall demonstrate that the first two have representations using *constrained* compact outcome graphs.

Included Sellers

An *included sellers* global constraint is one in which a global outcome o is acceptable if the set of included sellers $\sigma_{inc}(o)$ is in some collection $\mathcal{S}$ of sets of sellers. Such a constraint has a representation in which the set X is the set of subsets of sets of sellers in $\mathcal{S}$:

$$X = \{x : x \subseteq \sigma' \text{and } \sigma' \in \mathcal{S}\}$$

From the definition of a constrained compact outcome graph, every vertex of G $(i,s,\xi_{i,s},x)$ for which $s < S$ has a collection of edges going to vertices of the form $(i,s+1,\xi_{i,s}+q_{s+1},x')$, at most one for each $q_{s+1} \le Q - \xi_{i,s}$. We construct G by choosing $x' = x$ if $q_{s+1} = 0$, and $x' = x \cup \{s+1\}$ otherwise. We add exactly those edges to G for which x' is thus in X. Similarly for the "linking" edges going from (i,S,Q,x) to $(i+1,1,q_1,x')$ (where $i < I$) we choose $x' = x$ if $q_1 = 0$ and $x' = x \cup \{1\}$ otherwise, adding exactly the edges for which $x' \in X$. Finally, we connect the vertices (I,S,Q,x) such that $x \in \mathcal{S}$ to $\mathtt{t}$.

x maintains a record of the set of sellers included so far in the outcome corresponding to a path through G. Since only those vertices for which $x \in \mathcal{S}$ are connected to $\mathtt{t}$, only outcomes for which the set of included sellers in the full solution is in $\mathcal{S}$ will correspond to such paths.

The complexity of the k cheapest algorithm is slower by a factor of $|X|$ when a constraint is integrated, so as a specific example, we can ensure that outcomes have no more than R sellers by letting $\mathcal{S}$ be the collection of sets of R sellers, of which there are $S!/R!(S-R)!$. The set of states has size

$$\sum_{r=1}^{R} \frac{S!}{r!(S-r)!}$$

For fixed R with variable S this expression is polynomial of order R.

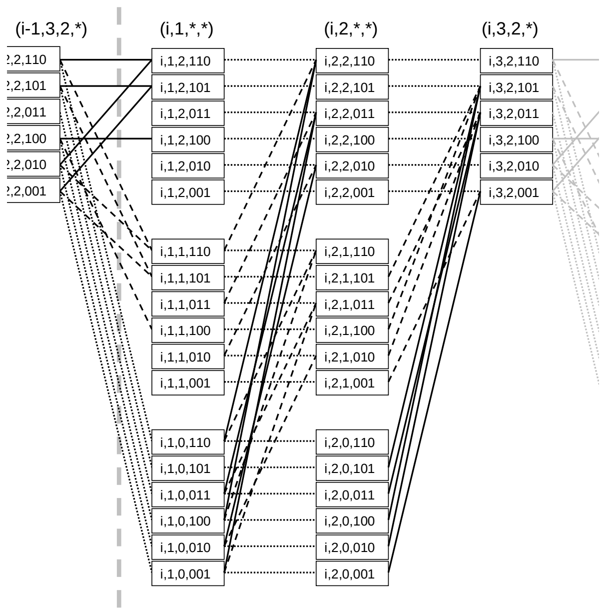

Figure 2: Section of the constrained solutions graph for $|\sigma_{inc}(o)| = 2$, in the case $Q = 2$, $S = 3$ and arbitrary I. The set of sellers x is represented in binary form. Edges are directed, left to right. Solid edges are labeled with $q = 2$; roughly dashed edges are labeled with $q = 1$ and finely dotted edges are labeled with $q = 0$.

Included Sellers Example

The full constrained solutions graph for $|\sigma_{inc}(o)| = 2$, in the case $Q = 2$, $S = 3$ and arbitrary I is shown in Figure 2. Sets of the 3 sellers are represented by binary strings: 110 represents sellers 1 and 2. The set of vertex states is

$$X = \{100, 010, 001, 110, 101, 011\}.$$

A vertex with state 110 can only be connected to another vertex with the same state, since once included a seller can not be removed; a vertex with state 010 could be connected to a vertex with state 010 or 110 representing either the assignment of an additional quantity to seller 2, or the assignment of this quantity to the previously unused seller 1.

Quantile Thresholds

As for included sellers, a global constraint evaluated on the total number of quantiles assigned to one or more sellers is computable at the level of an individual seller rather than an entire item-auction outcome, so there must be a constrained representation equivalent to the non-constrained representation.

We choose the set X to be a product of sets of the form $\{0, \ldots, T_s\}$, one for each seller s under consideration, where T_s is some threshold beyond which we are not interested in counting. Thus $x = (x_{s_1}, \ldots, x_{s_n})$, where each vertex state x_s counts the number of quantiles so far assigned to seller s. Clearly the complexity scale factor of integrating this constraint is $\prod_{s=1}^{S} T_s$.

To implement a *minimum* we use $x'_s = \min(T_s, x_s + q_s)$, and connect those vertices (I, S, Q, x) for which $x_s = T_s$ for each $s = s_1, \ldots, s_n$ to the sink `t`. To implement a *maximum* we use $x'_s = x_s + q_s$, omitting edges for which this is greater than T_s. We connect all vertices (I, S, Q, x) to `t`.

5 Related Work

Auctions are an increasingly important medium of exchange, and the theory of auctions is a large and active research area; Klemperer [1999; 2004] provides detailed reviews of the theory and practice of auctions. Elmaghraby [2000] reviews procurement auctions in the broader context of supply contract competition and sourcing problems. Dasgupta & Spulber [1989] consider the abstract problem of designing optimal procurement auctions (from the buyer's point of view) under both sole-sourcing and multi-sourcing. Recently, Tunca & Wu [2006] have explored practical and implementable procurement mechanisms that allow multi-sourcing.

Whereas the Economics literature typically assumes that auction participants can explicitly articulate their preferences and that profit/utility/surplus maximization is the goal of auction design, a complementary body of AI/E-Commerce research considers cases where these assumptions do not hold. Sandholm & Suri [2001] describe a range of side constraints and non-price attributes of allocations that can be important in real auctions. The literature on preference elicitation in auctions proceeds from the observation that agents cannot always explicitly state their side constraints and their preferences over non-price solution attributes. Sandholm & Boutilier [2006] review elicitation techniques aimed at helping agents in such situations. Preference elicitation methods typically make strong assumptions about the functional form of agent preferences and constraints, and the algorithms require an exponential number of interactive queries in the worst case. Parkes [2005] explicitly models the cost of elicitation in an auction design problem.

Preference elicitation extends an optimization framework. In a procurement auction, for instance, elicitation methods attempt to interactively refine a model of the buyer's latent utility function. This utility function can then be used in a conventional mathematical optimization solver to compute an optimal auction outcome. By contrast, Kelly & Byde [2006a] introduce a fundamentally different approach that casts the buyer's decision problem as one of *data exploration* rather than optimization. The authors observe that 1) the WDP in sealed-bid auctions is a generalized knapsack problem, 2) knapsack problems admit solution via dynamic programming, 3) dynamic programs correspond to shortest path problems, and 4) efficient k-shortest paths algorithms exist; therefore it is straightforward in principle to compute k-best solutions to any sealed-bid auction. The authors apply their method to bids from a real procurement auction and report that the k-cheapest solutions can be post-processed to aid the buyer in a variety of ways.

While the Kelly & Byde approach is workable in principle, its practical scalability is severely limited by two major drawbacks: First, it employs only simple "first-generation" k-shortest paths algorithms developed during the 1950s–1970s, ignoring asymptotically superior algorithms developed during the 1990s. Second and more importantly, the Kelly &

Byde method requires that solutions to individual-item sub-auctions be *generated exhaustively* and held in memory before the *k*-shortest paths algorithm is invoked. Because the number of sub-auction solutions is exponential in both the number of buyers and the granularity of multi-sourcing (S and Q in our notation), the time and memory requirements of the method are exponential in these crucial problem-size parameters. Our work corrects the deficiencies of [Kelly and Byde, 2006a] by applying modern *k*-shortest paths algorithms [Eppstein, 1998] and by defining a far more compact sub-auction solutions graph (Figure 1) to obtain a *k*-best solutions algorithm whose time and memory requirements are polynomial in *all* problem-size parameters.

6 Conclusions and Future Work

In this paper we have described an efficient algorithm for generating *k*-cheapest solutions to a range of procurement auction winner determination problems. It is useful for several purposes, most notably: to "explore the competitive landscape" by data-mining a large number of solutions; to simultaneously compute prices for many bundles of constraints; and to price non-linear constraints that cannot be incorporated into a standard integer linear program. Our approach can also efficiently incorporate certain classes of common hard constraints into the solution-generation process. If we are faced with constraints that do not admit efficient expression in a constrained solutions graph, we may simply generate solutions until one obeying the constraint is found. Whereas previous algorithms for generating *k*-cheapest solutions scale poorly in the number of sellers and the number of quantiles, our algorithm scales well in *all* problem size parameters. Previous work introduced the *k*-cheapest-solutions approach to procurement auction clearing and explained its benefits; our work makes this approach practical.

Our ongoing work involves large-scale tests of our algorithm using randomly generated bid data as well as bids from an actual procurement auction, to further understand what benefits can be gained by examining a large number of solutions. In particular we are exploring post-processing techniques, including dominance pruning and clustering, that summarize and condense the *k*-best solutions for the decision-maker. Finally, we are investigating more efficient ways of incorporating bundle- or XOR-bidding into our algorithms.

References

[Dasgupta and Spulber, 1989] Sudipto Dasgupta and Daniel F. Spulber. Managing procurement auctions. *Information Economics and Policy*, 4(1):5–29, 1989.

[Dijkstra, 1959] E. W. Dijkstra. A note on two problems in connection with graphs. *Numerische Mathematik*, 1:83–89, 1959.

[Elmaghraby, 2000] Wedad J. Elmaghraby. Supply contract competition and sourcing policies. *Manufacturing & Service Operations Management*, 2(4):350–371, 2000.

[Eppstein, 1998] David Eppstein. Finding the k shortest paths. *SIAM Journal on Computing*, 28(2):652–673, 1998.

[Hannon, 2004] David Hannon. Sun shines by combining two supplier strategies. *Purchasing*, May 2004. `http://www.purchasing.com/article/CA416161.html`.

[Kelly and Byde, 2006a] Terence Kelly and Andrew Byde. Generating k-best solutions to auction winner determination problems. *ACM SIGecom Exchanges*, 6(1), April 2006.

[Kelly and Byde, 2006b] Terence Kelly and Andrew Byde. Generating k best solutions to winner determination problems: Algorithms & application to procurement. Technical Report HPL-2006-40, Hewlett-Packard Laboratories, March 2006. `http://www.hpl.hp.com/techreports/2006/HPL-2006-40.html`.

[Klemperer, 1999] Paul Klemperer. Auction theory: A guide to the literature. *Journal of Economic Surveys*, 13(3):227–260, July 1999.

[Klemperer, 2004] Paul Klemperer. *Auctions: Theory and Practice*. Princeton University Press, 2004. ISBN 0-691-11426-9.

[Parkes, 2005] David C. Parkes. Auction design with costly preference elicitation. *Annals of Mathematics and Artificial Intelligence*, 44(3):269–302, July 2005.

[Sandholm and Boutilier, 2006] Tuomas Sandholm and Craig Boutilier. *Combinatorial Auctions*, chapter 10. MIT Press, January 2006.

[Sandholm and Suri, 2001] Tuomas Sandholm and Subhash Suri. Side constraints and non-price attributes in markets. In *IJCAI 2001 Workshop on Distributed Constraint Reasoning*, August 2001.

[Tunca and Wu, 2006] Tunay I. Tunca and Qiong Wu. Multiple sourcing and procurement process selection with bidding events, 2006. Working Paper, Stanford University Graduate School of Business.