Generating k-Best Solutions to Auction Winner Determination Problems

TERENCE KELLY and ANDREW BYDE
Hewlett-Packard Laboratories

Auction participants cannot always articulate their requirements and preferences. Sometimes, for instance, the buyer in a procurement auction cannot quantify the value of non-price solution attributes or delineate between hard and soft constraints. This precludes formulating the winner determination problem (WDP) as an optimization problem. Existing decision-support aids for such situations extend an optimization framework. We present an approach that frames the decision problem as one of *exploration* rather than optimization. Our method relies on an algorithm that generates k-best solutions to auction WDPs. Our algorithm can incorporate hard constraints into the generation process and can scale to practical procurement auctions. We show how to extract useful guidance from k-best WDP solutions, and we evaluate our method using real bids submitted by real suppliers in an HP material parts procurement auction.

Categories and Subject Descriptors: G.2.2 [**Graph Theory**]: Path and Circuit Problems

General Terms: Algorithms, Economics, Experimentation, Performance, Theory

Additional Key Words and Phrases: auctions, decision support, knapsack problems, k-shortest paths, preference elicitation, procurement

1. INTRODUCTION

Winner determination problems (WDPs) in auctions are *computationally* difficult for many interesting auction types, e.g., combinatorial auctions (CAs) [Rothkopf et al. 1998]. WDPs can also pose *cognitive* challenges: Agents may be unsure of the value of non-price solution attributes, or of whether to express considerations in the objective function or constraints of an optimization problem. These issues are vexing even in single-agent decision problems such as that faced by a bid-taking buyer in a reverse (procurement) auction. If the buyer "knows a good solution when she sees one" but cannot articulate its *properties*, we cannot formulate the WDP as an optimization problem.

Two existing techniques are used in such cases. In *scenario navigation*, the buyer runs a price-optimizing WDP solver with different constraints until finding an attractive solution. More sophisticated *preference elicitation* methods allow the expression of imprecise preferences over non-price solution features and/or query

Author addresses: `terence.p.kelly@hp.com` `andrew.byde@hp.com`
Document version: `$Id: bgk.tex,v 1.81 2006/04/27 18:40:48 kterence Exp $`

the buyer to refine a model of her latent utility function [Boutilier et al. 2004]. Preference elicitation offers advantages over ad hoc scenario navigation, but has several shortcomings: It restricts the functional form of preferences, it can require an excessive number of queries, and revealed preferences can be intransitive or otherwise problematic [Conen and Sandholm 2001; Sandholm and Boutilier 2006].

This paper introduces a different approach to decision support in auctions. Our method is based on an algorithm that generates k-best solutions to an auction WDP in descending order of objective function quality. In the most general case, it can generate solutions that maximize gains from trade in a combinatorial exchange, albeit with limited scalability. When specialized for reverse auctions our algorithm can scale to practical problem sizes. Hard constraints can be incorporated to prevent unacceptable solutions from being generated. Bids may express volume discounts/surcharges, and multi-sourcing is supported.

The k-cheapest solutions to a procurement auction can be post-processed in several ways to aid the decision-maker. If the buyer defines ordinal preferences over solution features, the Pareto frontier of undominated solutions may be computed. More importantly, k-best solutions define prices on bundles of constraints, which can focus the buyer's attention on the WDP's most pressing tradeoffs. Finally, we show how the k-cheapest solutions admit a variety of informative visualizations.

In summary, our main contribution is an algorithm that allows us to cast the procurement decision problem as one of *exploration* rather than *optimization*. Scenario navigation and preference elicitation are grounded in optimization: They use samples of the buyer's constraints or estimates of her preferences and they seek to compute an optimal solution. By contrast, we require only seller bids and we generate a large set of candidates from the most promising region of the solution space: solutions that entail minimal expense. The k-best solutions invite a wide range of analyses including clustering, visualization, and dominance pruning. In addition to revealing satisfactory WDP solutions, such exploration can yield valuable qualitative as well as quantitative insight into the cost of constraints and the nature of the competitive landscape. "Mining" candidate solutions is an increasingly popular paradigm in automated design [Yukish 2004], and our algorithm opens the possibility of applying similar strategies to auction decision-making.

An extended version of this paper contains additional material omitted here due to space limitations [Kelly and Byde 2006].

2. GENERAL APPROACH

We can generate k-best solutions to any combinatorial auction or exchange by linking the following observations:

1. The WDP in sealed-bid CAs is a generalized knapsack problem [Kelly 2004].
2. Dynamic programming can solve such problems [Kellerer et al. 2004].
3. Dynamic programs are longest-/shortest-paths problems [Ahuja et al. 1993].
4. We can generate the k shortest paths in a graph [Eppstein 1998].

We express the dynamic program corresponding to an auction WDP as an optimal-path problem on a graph whose path lengths represent the objective function (e.g., total surplus for exchanges or buyer expenditure for procurement). We then compute k-shortest paths to generate k-best solutions to the WDP.

The k-best WDP solutions invite a wide range of interesting analyses: If ordinal preferences are defined over solution attributes, dominance pruning yields solutions on the Pareto frontier. Furthermore, the k-best solutions *define prices on bundles of constraints*: The price of any bundle of constraints satisfied by a generated solution is the difference in objective function value between the best satisfying solution and the best unconstrained solution. No restrictions on the mathematical form of constraints are necessary; arbitrary non-linearities pose no special difficulties.

This general approach has limited scalability for arbitrary CAs; computing even the *first*-best solution to a CA is NP-hard [Rothkopf et al. 1998]. Practical pseudo-polynomial solvers are available if the number of types of goods in a multi-unit CA is small [Kelly 2004], but this approach does not scale in the number of good types. The remainder of this paper considers procurement auctions and presents an algorithm that can scale to real-world problems with many types of goods.

3. PROCUREMENT AUCTIONS

Businesses increasingly obtain goods through procurement auctions. Such auctions account for roughly $21 billion of HP's expenditures over the past four years, and US firms will spend hundreds of billions of dollars via procurement auctions in 2006 [Beckett 2005]. In practice, buyer preferences typically encompass non-price solution attributes and side constraints, e.g.,

1. a desire to have 2–4 suppliers for each type of good;
2. "XOR" constraints on winners, e.g., supplier B must be excluded if A is chosen;
3. constraints on the total number of winning sellers;
4. constraints on the distribution of expenditure across sellers.

Many constraints are "soft" in the sense that the buyer would relax them in exchange for sufficiently large savings.

This section presents a k-best-solutions algorithm for procurement auctions with multiple items (types of goods) available in multiple units. Seller bids may encode volume discounts and volume surcharges. The buyer may obtain units of an item from multiple suppliers (multi-sourcing), and she may constrain multi-sourcing on individual items before k-best solutions are generated. Section 4 expands our algorithm to include *global* constraints, and Section 5 applies the algorithm to real bids from an actual material-parts procurement auction.

3.1 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. Granularity parameter Q specifies the number of *quantiles* (shares of an item) that bids offer to supply. E.g., if $Q = 4$ bids offer to supply 25%, 50%, 75%, or 100% of the demand for each item. Let S denote the number of sellers. For each item i, seller s submits a bid B_{is} that is a list of (q, p) pairs, where q is a quantity in the range $1..Q$ and p is the payment that the seller requires for supplying q/Q of the buyer's demand for item i. (The seller implicitly receives zero payment if it supplies $Q = 0$ quantiles.) An *acceptable* solution to the auction WDP gives the buyer Q units of each item. The practical problems that motivate

	item i_1		item i_2		item i_3	
seller	50%	100%	50%	100%	50%	100%
s_A	\$3	\$6	\$4	\$7	\$5	\$11
s_B	\$2	\$7	\$5	\$8	\$4	\$10

Table I. Bids in example problem.

auction i_1 i_2 i_3	$\sum p$ (\$)	auction i_1 i_2 i_3	$\sum p$ (\$)	auction i_1 i_2 i_3	$\sum p$ (\$)
AA AA AA	24	AB AA AA	23	BB AA AA	25
AA AA AB	22	AB AA AB	21	BB AA AB	23
AA AA BB	23	AB AA BB	22	BB AA BB	24
AA AB AA	26	AB AB AA	25	BB AB AA	27
AA AB AB	24	AB AB AB	23	BB AB AB	25
AA AB BB	25	AB AB BB	24	BB AB BB	26
AA BB AA	25	AB BB AA	24	BB BB AA	26
AA BB AB	23	AB BB AB	22	BB BB AB	24
AA BB BB	24	AB BB BB	23	BB BB BB	25

Table II. Solutions in example problem.

our work involve well under a dozen sellers but can include scores of items. Small values of Q, e.g., $Q = 4$ quantiles, permit sufficiently expressive bids.

Example Consider a procurement auction with $I = 3$ items (i_1, i_2, and i_3) and $S = 2$ sellers (s_A and s_B). $Q = 2$, so sellers may supply 0%, 50%, or 100% of each item. Bids are shown in Table I. Seller s_A offers a volume discount on item i_2 and imposes a volume surcharge on i_3. Seller s_B imposes a surcharge on items i_1 and i_3 but offers a discount on i_2. Each sub-auction has three acceptable solutions: The buyer may acquire 100% from s_A, 100% from s_B, or 50% from each. Therefore there are $3^3 = 27$ acceptable solutions to the overall WDP. Table II lists the solutions and the total payment associated with each. Sub-auction solutions are encoded as two-character strings: "AA" means s_A supplied 100% of the item, "AB" means each seller supplied 50%, etc. The best solution costs \$21. However, if we require that seller s_B must supply at least 50% of each item, then the best solution costs \$22 and we say that the constraint costs \$1.

3.2 Generating Individual-Item Outcomes

The number of acceptable solutions to each sub-auction is the number of ways Q indistinguishable balls (representing quantiles) can be placed in S distinguishable boxes (representing sellers). The number of solutions is given by the multiset formula (see Feller [1970]): $R(S,Q) = \frac{(Q+S-1)!}{Q!(S-1)!}$. For problem sizes of practical interest, $R(S,Q)$ is remarkably small, e.g., $R(12,10) = 352,716$. We generate all acceptable solutions to individual-item sub-auctions using a recursive algorithm described in the extended version of this paper [Kelly and Byde 2006]. The asymptotic time requirement is $O(R(S,Q))$, which is acceptable in practice.

Table III summarizes the notation developed so far and used in the remainder of this section.

3.3 Generating k Best Overall Solutions

The number of acceptable solutions to our *overall* procurement auction WDP is $R(S,Q)^I$, which is far too large to generate for problems of practical interest. We therefore *selectively* generate overall solutions in ascending order of buyer expense.

notation	meaning
I	number of items (types of goods)
S	number of sellers
Q	number of quantiles (shares)
k	desired number of solutions
R	shorthand for $R(S, Q)$

Table III. Notation.

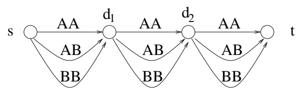

Fig. 1. Solutions graph for example of Section 3.1.

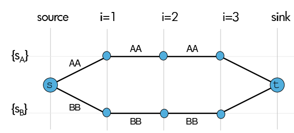

Fig. 2. Constrained solutions graph G_1: $S = 2$, $I = 3$, $Q = 2$.

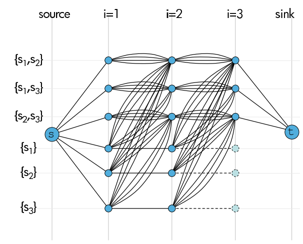

Fig. 3. Constrained solutions graph G_2: $S = I = Q = 3$.

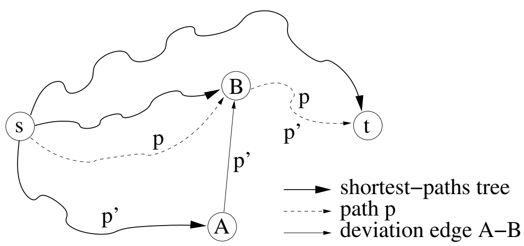

Fig. 4. Illustration of Definition 3.1. The tree of shortest paths from source node s is shown in bold, and an arbitrary path p from s to destination t is shown with a dashed line. Path p', a deviation of p, consists of the shortest path from s to node A followed by deviation edge $A \to B$ and the remainder of p from B to t. Deviations of p' have deviation arcs incident to the path between s and A only; see Perko [1986].

We construct a *solutions graph* in which paths correspond to acceptable overall WDP solutions; Figure 1 shows the graph for our example. Edges between a node pair represent acceptable solutions to individual-item sub-auctions and are labeled as in Table II. Edge weights (not shown) represent the cost of the corresponding sub-auction solution. Paths from s to t correspond to acceptable solutions to the overall WDP, and path lengths represent total buyer expenditure. k-cheapest acceptable solutions to the procurement auction WDP are k-shortest paths in the solutions graph.

Our k-shortest paths algorithm uses a memory-efficient representation of solution graph paths in a length-ordered heap. This representation is based on the concept of *deviations*. First, compute a tree of shortest paths from the origin.

Definition 3.1 Hoffman and Pavley [1959]. *A **deviation** from a path p is a path p', having the same origin and destination as p, which is initially part of the shortest-paths tree, which then contains exactly one link, called the deviation link, which is not a link of p, but whose terminal node is the terminal node of a link of p. The final portion of p' coincides with p. (See Figure 4.)*

Edge weights are non-negative, so a deviation p' is no shorter than its "parent" path p, and a path is uniquely defined by its parent and deviation edge; see Perko [1986] for important technical details. We can therefore represent all solution graph paths as a heap-ordered tree with the shortest path as the root; the children of each node are its deviations. This tree can be constructed *incrementally* by generating paths in ascending order of length. Figure 5 sketches our algorithm, SKSP, discussed in detail in Kelly and Byde [2006].

```
1: compute tree of shortest paths
2: insert shortest path into PQ
3: while fewer than k paths have been generated
4:     p = extract-min(PQ)  /* next shortest path */
5:     if p is NULL
6:         terminate  /* fewer than k paths exist */
7:     output p
8:     for each deviation d of path p
9:         if PQ contains fewer than k paths
10:            insert d into PQ
11:        else  /* PQ full, w/ exactly k paths */
12:            if d is longer than longest path in PQ
13:                discard d
14:            else
15:                remove & discard longest path in PQ
16:                insert d into PQ
```

Fig. 5. The SKSP algorithm. Priority queue `PQ` holds paths, ordered by length. `PQ` must support extraction of both longest and shortest paths, so it is implemented as a pair of binary heaps. SKSP is memory efficient because lines 12–16 maintain the invariant that `PQ` contains at most k paths.

In the worst case, SKSP requires $O(k \log(k) IR)$ time and $O(k + IR)$ memory where R is the $R(S, Q)$ of the multiset formula; see Kelly and Byde [2006] for details. In the special case where $Q = 1$ (i.e., no multi-sourcing), SKSP requires $O(k + IS)$ memory and $O(k \log(k) IS)$ time, because $R(S, Q) = S$ in this case. Our ongoing research is developing improved algorithms with reduced asymptotic time and memory requirements.

4. GLOBAL CONSTRAINTS

Global constraints on a full solution are those whose satisfying global solutions are not the product of restricted sets of individual-item outcomes. This section describes an approach for modifying the simple graph representation of Section 3.3 to incorporate certain types of hard constraints, in the sense that solutions that violate the constraints are not generated when the k-shortest paths algorithm operates on the modified graph. We call the expanded graph that encodes global constraints a *constrained solutions graph*, to distinguish it from the simple linear solutions graph depicted in Figure 1. We show that several useful global constraints have efficient representations; Kelly and Byde [2006] contains further examples.

4.1 Constrained Number of Winners

In this section we will demonstrate how to expand the solutions graph to form a constrained solutions graph G_f whose paths correspond to outcomes with a fixed number S_f of sellers receiving non-zero allocations.

Define $\sigma_{inc}(o_i)$ to be the set of sellers that receive non-zero quantiles (that are *included*) in some auction given the outcome o_i, let $\sigma_{inc}(o_1, \ldots, o_i)$ likewise be the set of sellers that are included in the outcome $(o_1, \ldots, o_i)$ of a collection of auctions.

Let Σ_f be the set of sets of S_f or fewer sellers. The set of nodes $V(G_f)$ is $V(G_f) = \{\mathtt{s}, \mathtt{t}\} \cup (\{1, \ldots, I\} \times \Sigma_f)$. The nodes $\mathtt{s}$ and $\mathtt{t}$ are a convenient source and sink for the k-shortest paths algorithm. For each vertex (i, σ) and outcome o_{i+1} of the individual-item auction $i+1$ $(1 \le i < I)$ such that $\sigma \cup \sigma_{inc}(o_{i+1}) \in \Sigma_f$ we add a directed edge from (i, σ) to $(i+1, \sigma \cup \sigma_{inc}(o_{i+1}))$, labeled with o_{i+1}, and with length $c(o_{i+1})$ equal to the cost of outcome o_{i+1}. In addition, for each outcome o_1 to the first auction we connect the source node $\mathtt{s}$ to $(1, \sigma_{inc}(o_1))$ with an edge of length $c(o_1)$ labeled with o_1, and we connect every node of the form (I, σ) for which $|\sigma| = S_f$ to the sink node $\mathtt{t}$ via an edge of length zero.

PROPOSITION 4.1. *The labeling of edges establishes a 1–1 mapping from paths in G_f connecting* s *to* t*, to global outcomes o satisfying the constraint $|\sigma_{inc}(o)| = S_f$.*

Proof. See [Kelly and Byde 2006]. □

Since the length of a path in G_f is the cost of the corresponding global outcome, the k-shortest paths of G_f correspond to the k-cheapest solutions to the winner determination problem. Figure 2 shows G_1 for the example in Section 3.1 ($I = 3$, $S = 2$, $Q = 2$). A more complicated example, G_2 given $I = S = Q = 3$, is shown in Figure 3. Note that G_f may contain vertices and edges that are not on any path from s to t—for example (I, σ) with $|\sigma| < S_f$, as shown in Figure 3.

The complexity of G_f relative to the unconstrained solutions graph depends on the size of Σ_f. For S_f fixed, but S varying, $|\Sigma_f| = O(S^{S_f})$.

4.2 Incremental Function Representation

In general, what we are doing in Section 4.1 is evaluating a function at each step whose value (set of sellers included in first i auctions) depends on the value at the previous step (set of sellers included in first $i-1$ auctions) and the outcome in the i^{th} auction. Any chain of functions of this form such that the suitability of a global outcome can be determined by examining the output of the last function provides a way of restricting the k-best algorithm to generate only those global outcomes (paths) that satisfy the global constraint. In the following definition, O_i is the set of outcomes to the i^{th} single-item auction, and $O = \prod_i O_i$ is the set of global outcomes.

DEFINITION 4.1. *Suppose that $O_{cons} \subseteq O$ is a global constraint represented as a subset of the space of global outcomes (those that are acceptable). An* ***incremental representation*** *of O_{cons} is defined as a sequence of sets $X_i, i = 0, \ldots, I$ with $X_0 = \{*\}$, functions $f_i : X_{i-1} \times O_i \to X_i, i = 1, \ldots, I$, and a subset $X_{cons} \subseteq X_I$, such that the function $\mathcal{F} : O \to X_I$ defined by*

$$\mathcal{F}(o_1, \ldots, o_I) = f_I(f_{I-1}(\ldots f_2(f_1(*, o_1), o_2), \ldots o_{I-1}), o_I)$$

satisfies $\mathcal{F}^{-1}(X_{cons}) = O_{cons}$. Given such a representation, the sets X_i will be referred to as the ***partial values****, the functions f_i are the* ***incremental functions****, and X_{cons} the* ***final values****.*

In these terms, the constraint in Section 4.1 is represented by partial values equal to the set of subsets of sellers, $X_i = \Sigma_f$; the incremental function is the union of the partial value at step $i-1$ with the set of sellers included in step i, $f_1(*, o_1) = \sigma_{inc}(o_1)$, $f_i(x, o_i) = x \cup \sigma_{inc}(o_i)$, $i > 1$; and the final values are $X_{cons} = \{x \in \Sigma_f : |x| = S_f\}$.

4.2.1 *Constrained Solutions Graph.* For any incremental representation we can construct a corresponding constrained solutions graph G as in Section 4.1. We let the nodes of G correspond to the partial values, with a source and sink node added,

$$V(G) = \{\mathtt{s}, \mathtt{t}\} \cup \bigcup_{i=1}^{I} \{i\} \times X_i.$$

We will identify the source node s with $(0, *)$ so as not to make special cases for f_1. For each $i < I$, $x_i \in X_i$ and $o_{i+1} \in O_{i+1}$ we add an edge to G to represent the

data set	# items I	# quantiles Q	vol discount?	% total cost	# undominated	# winners	# subsets	$\overline{H}$ range
$B_{50,1}$	50	1	N	99	33	5	1	0.461–0.545
$B_{25,1}$	25	1	N	90	30	3,4,5	4	0.304–0.691
$B_{25,4}$	25	4	N	90	35	4,5	2	0.417–0.548
$B_{25,4,D}$	25	4	Y	90	26	3,4,5	5	0.307–0.674
$B_{15,4,D}$	15	4	Y	75	46	2,3,4,5	8	0.096–0.756

Table IV. Summary of experiments and results.

transition that the outcome o_{i+1} induces from the "state" x_i to the new state $x_{i+1} = f_{i+1}(x_i, o_{i+1})$. This edge, which goes from the node (i, x_i) to $(i+1, f_{i+1}(x_i, o_{i+1}))$ is labeled with o_{i+1} and has length $c(o_{i+1})$. The reasoning behind Prop. 4.1 gives

PROPOSITION 4.2. *The labeling of edges establishes a 1–1 mapping from paths in G connecting* $\mathtt{s}$ *to* $\mathtt{t}$ *to global outcomes o satisfying the constraint $o \in O_{const}$.* □

4.3 Complexity

The complexity analysis of Section 3.3 and [Kelly and Byde 2006] extends naturally to constrained solutions graphs. The number of nodes in the graph is now $2 + \sum_{i=1}^{I} |X_i|$, and the number of edges $\sum_{i=1}^{I} |O_i| \times |X_{i-1}| + X_{cons}$. Introduce the constant $X_{max} = \max_i |X_i|$. Clearly the number of nodes is $O(IX_{max})$ and edges $O(IRX_{max})$, where $R = R(S, Q)$ is the multiset formula. Following through the same logic as in Section 3.3, we get that the storage requirements are $O(IRX_{max} + k)$ and time requirements $O(k \log(k) IRX_{max})$. In other words, the problem has become more complex by a factor of at most X_{max}.

5. EXPERIMENTS

We use our SKSP algorithm to compute k-best solutions from actual bids submitted to a material-parts procurement auction in which HP spent roughly $3.7 million. Our data includes, for each of several dozen items, the total number of units that HP wished to acquire and per-unit prices from each of six sellers. We rank items in descending order of each item's *minimal procurement cost*, i.e., the lowest possible cost of satisfying HP's total demand for the item.

Our experiments use five bid subsets, summarized in Table IV. Names encode size and other properties, e.g., $B_{50,1}$ includes the top 50 items in the auction and sets Q to 1 (i.e., no multi-sourcing). The top 50 items account for 99% of the buyer's total cost if every item is acquired at its minimal procurement cost. Bid set $B_{25,4}$ includes only the top 25 items, which account for 90% of the buyer's minimal total expenditure; it sets $Q = 4$, so a seller may supply 0%, 25%, 50%, or 100% of any item. Bid sets $B_{25,4,D}$ and $B_{15,4,D}$ include randomly-generated volume discounts for quantities $q = 1, \ldots, Q-1$ in these bids.

Our first remarkable result is that *the top k solutions entail nearly equal expenditure.* Figure 6 plots total buyer expenditure as a function of solution rank k for the top 100,000 solutions based on the $B_{50,1}$ bid set. Seller expenditure increases very gradually with k (note the logarithmic x axis); the 100,000th-best solution is only

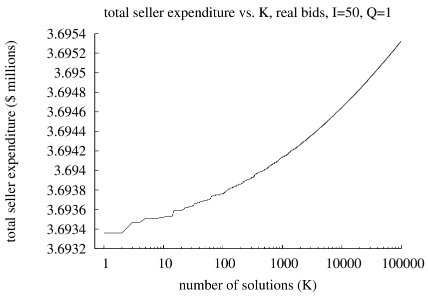

Fig. 6. Expenditure vs. solution rank.

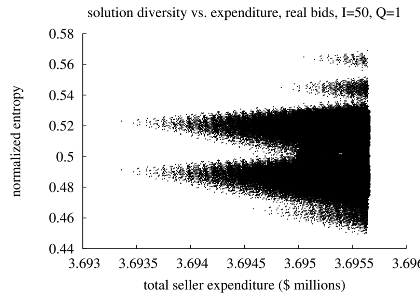

Fig. 7. $\overline{H}$ vs. expenditure.

0.054% more expensive than the cheapest solution. This result is robust across a wide range of our experiments.

But are the top k solutions *diverse*, i.e., do they differ in "interesting" ways? We address this question by defining summary measures of WDP solutions and describing how they vary among the top $k = 25,000$ solutions for each of our five bid sets. Our first measure summarizes the way a WDP solution distributes the buyer's money across sellers. Consider an *expenditure vector*, $\vec{x} = (x_1, x_2, \ldots, x_S)$, where x_s is the fraction of total expenditure given to seller s. We treat $\vec{x}$ as a probability vector and compute its information-theoretic entropy [Cover and Thomas 1991], normalized to lie in $[0, 1]$: $\overline{H} = \sum_{s=1}^{S} -x_s \log_2 x_s / M$, where M is a normalization constant. $\overline{H} = 1$ when all sellers receive the same amount of money; $\overline{H} = 0$ when a single seller supplies all goods. In addition to $\overline{H}$, we also consider the number of sellers included in a WDP solution and the specific subset of sellers included.

The rightmost three columns of Table IV present three views of solution diversity among the top $k = 25,000$ solutions for our bid sets. Column 7 shows the numbers of winners. For instance, *all* of the top 25,000 solutions based on the $B_{50,1}$ bids involve exactly five sellers, but the $B_{25,1}$ solutions involve three, four, or five sellers. Since $S = 6$, one of $2^6 - 1 = 63$ subsets of sellers win in a solution. Column 8 shows the number of such subsets that actually occur among the top 25,000 solutions. All solutions to $B_{50,1}$ involve the *same* subset of sellers, but four subsets occur among the solutions to $B_{25,1}$. The last column in Table IV shows the range of normalized entropy $\overline{H}$ across the top 25,000 solutions.

Our three diversity measures suggest similar conclusions: Comparing $B_{50,1}$ with $B_{25,1}$, and comparing $B_{25,4,D}$ with $B_{15,4,D}$, we see that *including fewer top items increases solution diversity.* An analogy to road networks illustrates why: The second-shortest path between two cities is likely to involve a brief and trivial detour from the shortest path if the network includes minor streets. We can ensure that paths differ substantially by including only major highways (or, by analogy, only items with high minimal procurement cost).

A comparison of $B_{25,1}$ with $B_{25,4}$ suggests that multi-sourcing items by increasing Q may decrease solution diversity. However the $B_{25,4,D}$ results suggest that volume discounts can restore the diversity lost by increasing Q. Finally, $B_{15,4,D}$ shows that the combined effect of volume discounts and a greatly reduced item count dramatically increases diversity.

$\overline{H}$ range	number of sellers in solution 2	3	4	5
[0.0, 0.1)	0.819%			
[0.1, 0.2)	0.772%	0.276%		
[0.2, 0.3)	0.594%	0.098%	0.512%	
[0.3, 0.4)	0.655%	0%	0.290%	0.883%
[0.4, 0.5)		0.053%	0.112%	0.638%
[0.5, 0.6)		0.172%	0.112%	0.416%
[0.6, 0.7)			0.284%	0.416%
[0.7, 0.8)				0.527%

Table V. Percent premium vs. $\overline{H}$ and number of winners.

How can we exploit k-best solutions when the WDP is viewed as a multi-criteria optimization problem? For instance, what if we prefer greater uniformity of expenditure across sellers (all else being equal), but we can not quantify the dollar value of increasing $\overline{H}$? Figure 7 is a scatterplot of $\overline{H}$ versus total expenditure for over 250,000 of the top $B_{50,1}$ solutions. The Pareto frontier of undominated solutions appears remarkably small. Column 6 in Table IV shows that this is true for all of our data sets: For the bi-criteria problem involving expenditure and normalized entropy, the top 25,000 solutions contain fewer than 50 undominated solutions. Experiments with other criteria lead to the same conclusion: *For practical problems, the Pareto frontier is conveniently small.*

Note that scenario navigation and preference elicitation methods based on integer programming will face difficulty if preferences or constraints involve normalized entropy, because the logarithmic terms in $\overline{H}$ turn the WDP into a difficult *non-linear* mixed integer program. This illustrates an important general advantage of our method: It places no restrictions on the form of preferences or constraints, and therefore it does not force a tradeoff between convenience of implementation and fidelity in modeling the problem domain. E.g., it does not force us to shoe-horn fundamentally non-linear preferences or constraints into a linear framework merely to suit the limitations of integer linear program solvers.

We now consider pricing bundles of constraints. Suppose that preferences involve the number of sellers in a solution as well as uniformity of expenditure ($\overline{H}$). Table V summarizes the 25,000 best solutions from the $B_{15,4,D}$ bids. For each number of sellers and each $\overline{H}$ range, the table shows the additional cost of the best solution expressed as a percentage of the cheapest solution. For example, the cheapest solution with five sellers and $\overline{H}$ in $[0.7, 0.8)$ costs 0.527% more than the cheapest unconstrained solution. The prices of bundles of constraints can be read directly from Table V. For example, if the decision-maker insists on a solution involving five sellers and $\overline{H} \geq 0.4$, the cheapest solution satisfying this bundle of constraints costs 0.416% more than the cheapest unconstrained solution.

6. RELATED WORK

Preference elicitation techniques for single-agent decision problems have been applied to auctions in order to preserve privacy and shorten bids [Lahaie and Parkes 2004]. Boutilier et al. [2004] explore preference elicitation to aid uncertain decision-makers in auctions. Sandholm and Boutilier [2006] review preference elicitation in combinatorial auctions. Most approaches place strong restrictions on preferences and require exponentially many queries in the worst case.

The relationship between combinatorial auction WDPs and generalized knapsack problems has attracted little attention until recently. Kellerer et al. [2004] and Kelly [2004] provide the first detailed discussions of the auction-knapsack connection.

Eppstein [1998] surveys k-shortest paths problems and algorithms. Techniques for encoding constraints in graphs so that a k-shortest paths algorithm generates only paths that satisfy the constraints have been explored. For example, Villeneuve and Desaulniers [2005] describe an approach based on string-matching algorithms; as noted above, this method is not suitable for our problem. Coutinho-Rodrigues et al. [1999] employ k-shortest paths computations with interactive elicitation queries to explore the Pareto frontier in bi-criteria optimization problems.

To the best of our knowledge, ours is the first systematic method of generating k-best solutions to auction WDPs that has been proposed.

7. CONCLUSIONS

This paper has introduced a general method for computing k-best solutions to auction WDPs. Applied to procurement auctions, it supports multi-sourcing and volume discounts/surcharges, and it scales to practical problem sizes: In the real-world procurement auctions that motivate our research, the number of items may be large but the number of sellers is not large and multi-sourcing at fine granularity is unnecessary. The time and memory requirements of our k-best solutions algorithm scale *linearly* in the number of items, and scale with $k \log k$. The constrained outcome graph can furthermore accommodate many useful global hard constraints with only a modest increase in computational complexity.

Our approach complements existing preference elicitation techniques and sidesteps several of their shortcomings, e.g., the need for excessively numerous or excessively vexing queries and restrictions on the functional form of constraints and utility functions. Because our approach does not involve the elicitation of preferences, the possibility of intransitive revealed preferences is absent. An important general advantage of our method compared to existing approaches is that arbitrary non-linearities in constraints or in preferences over solution attributes pose no special difficulties. This is important because non-linear solution attributes are natural in the procurement domain (e.g., our normalized entropy measure of expenditure uniformity).

Empirical results based on real procurement auction bids demonstrate that our algorithm can generate a large number of solutions to practical WDPs. Furthermore, the analysis of k-best solutions can give the decision-maker valuable qualitative as well as quantitative insight into the solution space. For example, our approach directly yields prices for all bundles of constraints satisfied by one or more of the k-best solutions; such prices are particularly helpful for the soft constraints of Section 3. Of course, the existing technique of scenario navigation based on integer program solvers can compute the price of one set of constraints at a time. By contrast, our approach *simultaneously* computes the prices of *many* bundles of constraints using a *lightweight* algorithm.

More generally, the set of k-best solutions provides rich opportunities to leverage a wide range of existing data-analysis techniques because it allows us to cast the auction decision problem as one of data exploration rather than optimization. Our empirical results show that enabling this data exploration perspective is useful because the k-best solutions are *similar* in terms of overall buyer expenditure yet are *diverse* in terms of several important characteristics (e.g., number of winning sellers and uniformity of buyer expenditure across sellers). We found that multi-sourcing can reduce diversity, but that diversity can be increased by volume discounts and by including only items whose minimal procurement cost is large. Finally, we considered the problem of *condensing* a large and diverse set of solutions to a manageable size via dominance pruning, and we found that the Pareto frontier of undominated solutions is conveniently small.

ACKNOWLEDGMENTS

We thank Kemal Guler for reviewing numerous drafts of this paper. Donald Knuth provided valuable insights into the combinatorial generation problem of listing sub-auction solutions. Alex Zhang explained the implications of our non-linear measure of expenditure uniformity. Michael Wellman, Daniel Reeves, and Claudio Bartolini offered detailed feedback on a late draft. Finally, we thank the anonymous SIGecom Exchanges reviewers for their detailed comments.

REFERENCES

Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. 1993. *Network Flows.* Prentice Hall.

Beckett, J. 2005. The business of bidding: Reinventing auctions for better results. `http://www.hpl.hp.com/news/2005/jul-sep/auctions.html`.

Boutilier, C., Sandholm, T., and Shields, R. 2004. Eliciting bid-taker non-price preferences in (combinatorial) auctions. In *Proc. AAAI.*

Conen, W. and Sandholm, T. 2001. Preference elicitation in combinatorial auctions [extended abstract]. In *Proc. ACM E-Commerce Conf.*

Coutinho-Rodrigues, J., Climaco, J., and Current, J. 1999. An interactive bi-objective shortest path approach. *Computers & Operations Research 26*, 789–798.

Cover, T. M. and Thomas, J. A. 1991. *Elements of Information Theory.* Wiley.

Eppstein, D. 1998. Finding the k shortest paths. *SIAM Journal on Computing 28,* 2, 652–673.

Feller, W. 1970. *An Introduction to Probability Theory and Its Applications*, Third ed. Vol. 1. John Wiley & Sons.

Hoffman, W. and Pavley, R. 1959. A method for the solution of the nth best path problem. *Journal of the ACM 6,* 4 (Oct.), 506–514.

Kellerer, H., Pferschy, U., and Pisinger, D. 2004. *Knapsack Problems.* Springer.

Kelly, T. 2004. Generalized knapsack solvers for multi-unit combinatorial auctions. In *AMEC VI.* `http://ana.lcs.mit.edu/peyman/amec-vi-accepted.htm`.

Kelly, T. and Byde, A. 2006. Generating k best solutions to winner determination problems. Tech. Rep. HPL-2006-40, HP Labs. Mar.

Lahaie, S. M. and Parkes, D. C. 2004. Applying learning algorithms to preference elicitation. In *Proc. ACM E-Commerce Conf.* 180–188.

Perko, A. 1986. Implementation of algorithms for k-shortest loopless paths. *Networks 16,* 2, 149–160.

Rothkopf, M. H., Pekec, A., and Harstad, R. M. 1998. Computationally manageable combinatorial auctions. *Management Science 44,* 8 (Aug.), 1131–1147.

Sandholm, T. and Boutilier, C. 2006. *Combinatorial Auctions.* MIT Press, Chapter 10.

Villeneuve, D. and Desaulniers, G. 2005. The shortest path problem with forbidden paths. *European Journal of Operations Research 165*, 97–107.

Yukish, M. A. 2004. Algorithms to identify Pareto points in multi-dimensional data sets. Ph.D. thesis, U. Penn. Dept. of Mech. Eng.