Proceedings of the 2005 IEEE
International Conference on Automation Science and Engineering
Edmonton, Canada, August 1 & 2, 2005

A Lagrangian Heuristic for Bid Evaluation in e-Procurement Auctions

S. Kameshwaran
INRIA Lorraine
ISGMP Bat: A, Ile du Saulcy
Metz 57000
Email: kamesh@loria.fr

Y. Narahari
Department of Computer Science and Automation
Indian Institute of Science
Bangalore 560012
Email: hari@csa.iisc.ernet.in

***Abstract*— e-Procurement is an Internet based business process for sourcing direct or indirect materials. This paper considers a procurement scenario of a buyer procuring large quantities of a single good. The suppliers submit nonconvex piecewise linear supply curves as their bids. Such bids enable the suppliers to effectively express their economies of scale and transportation constraints. The buyer imposes a business rule of limiting the winning suppliers within a preferred range. The bid evaluation problem faced by the buyer is to select the winning suppliers and their trading quantities, such that the cost of procurement is minimized while satisfying the supply, demand, and business constraints. The bid evaluation problem is $\mathcal{NP}$-hard even for a simple special case. In this paper, the problem is formulated as a mixed integer linear programming problem and we propose a Lagrangian relaxation based heuristic to find a near optimal solution. The computational experiments performed on representative data sets show that the proposed heuristic produces a feasible solution with negligible optimality gap.**

I. INTRODUCTION

e-Procurement is an Internet-based business process for obtaining material and services and managing their inflow into the organization. e-Procurement is popularly implemented using privately owned e-marketplaces rather than using public e-markets. The private marketplace is owned by an organization or group of organizations to conduct the required procurement activities with access limited to certain suppliers. Once a procurement opportunity is identified, a request-for-quotation is distributed to the pre-selected potential suppliers. Initially, Internet technologies were used to create e-catalog systems, which supported fixed price trades. Auction mechanisms that enable dynamic pricing have become popular recently. With the Internet technologies enabling an interactive front end for human interaction and back end computers that can support complex business processes, the research in e-procurement is focused on bidding languages and bid evaluation techniques to make the process computationally and economically efficient.

In this paper we consider the procurement of multiple units of a single good, like a raw material. For industrial procurement, the traded quantity is large for a single supplier to provide the entire demand. Even otherwise, the buyer would prefer sourcing from multiple suppliers to reduce the high risk of failure on the single supplier. But too many suppliers would increase the cost of supply operations and hence the buyer would impose a business constraint of limiting the suppliers in a preferred range. Trading of large quantities also require a bidding language that can express the economies of scales and production and/or transportation constraints faced by the suppliers. We use a generic nonconvex piecewise linear cost curve as the bidding language. The bid evaluation problem faced by the buyer to select the winning bidders and their trading quantities is formulated as a mixed integer linear programming problem. Optimization techniques like linear programming and constraint programming are used in commercial bid analysis products from companies like Emptoris (www.emptoris.com), Rapt (www.rapt.com), and Mindflow (www.mindflow.com). The main advantage of optimization techniques is that one can easily incorporate various business rules and purchasing logic (like limiting the number of winning suppliers) as side constraints. However, with expressive bidding languages like piecewise linear supply curves, the bid evaluation problems are computationally challenging. If the auction is a multiple round auction, then the bid evaluation problem needs to be solved fast repeatedly with time constraints. In this paper, we propose a Lagrangian relaxation based heuristic to solve the bid evaluation problem in the procurement scenario discussed above.

II. PROCUREMENT OF MULTIPLE UNITS OF A SINGLE GOOD

The buyer is interested in procuring $\underline{b}$ units of a single good. The bid submitted by a supplier is a piecewise linear cost function over the quantity, shown in Fig. 1. The qualitative properties of the good are assumed to be well defined and identical across the different suppliers. The buyer imposes a business constraint that restricts the number of winning suppliers to be in the range $[\underline{W}, \overline{W}]$. This is an important

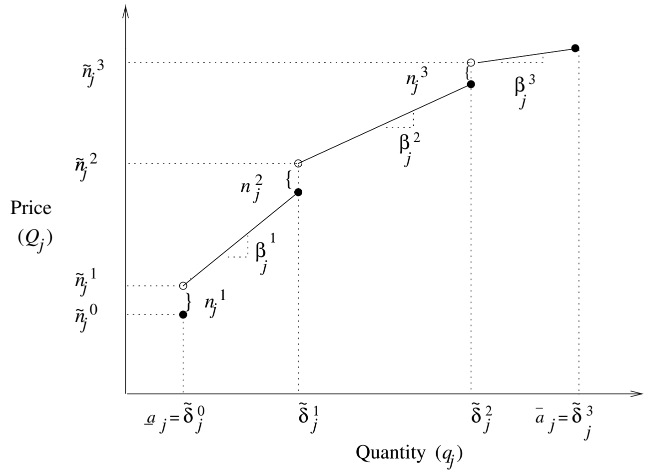

Fig. 1. Piecewise linear cost function Q_j

business constraint in industrial procurement settings to avoid too few or too many suppliers.

A. Piecewise Linear Supply Curves

In the procurement scenario discussed in this paper, the bid submitted by a supplier is a piecewise linear cost function over the quantity, shown in Fig. 1. The notations of the supply curve are explained below:

$[\underline{a}_j, \overline{a}_j]$	$\underline{a}_j$ is the minimum quantity that can be supplied and $\overline{a}_j$ is the upper limit
Q_j	Cost function for bid j defined over $[\underline{a}_j, \overline{a}_j]$
l_j	Number of piecewise linear segments in Q_j
β_j^s	Slope of Q_j on $(\tilde{\delta}_j^{s-1}, \tilde{\delta}_j^s)$
$\tilde{n}_j^s$	$\lim_{q_j \to \tilde{\delta}_j^{s+}} Q_j(q_j)$
n_j^s	Fixed cost associated with segment s of bid j

The cost function Q_j can be compactly represented by tuples of break points, slopes, and costs at break points: $Q_j \equiv ((\underline{a}_j = \tilde{\delta}_j^0, \ldots, \overline{a}_j = \tilde{\delta}_j^{l_j}), (\beta_j^1, \ldots, \beta_j^{l_j}), (\tilde{n}_j^0, \ldots, \tilde{n}_j^{l_j}))$. For notational convenience, define $\delta_j^s \equiv \tilde{\delta}_j^s - \tilde{\delta}_j^{s-1}$ and n_j^s as the fixed cost associated with segment s. Note that, by this definition, $n_j^0 = \tilde{n}_j^0$. The function is assumed to be non-decreasing, but the slopes β_j^s need not be decreasing as shown in the figure. The assumed cost structure is generic enough to include various special cases: concave, convex, continuous, and $\underline{a}_j = 0$. The cost structure enables the suppliers to express their *volume discount* strategy or *economies of scales* and/or the production and logistics constraints. The volume discount strategy, which is *buy more and pay less* can be expressed with marginally decreasing cost functions. The discontinuities in the cost structure can capture the production and transportation constraints.

Procurement auctions with piecewise linear cost curves are common in industry for long-term strategic sourcing [1]. A procurement scenario with piecewise linear cost function was considered in [2] and approximation algorithms based on dynamic programming were developed. The nonconvex piecewise linear supply curves were also used in multi-unit procurement of heterogeneous goods [3] and multiattribute procurement [4].

B. Bid Evaluation Problem

The bid evaluation problem faced by the buyer is to *minimize the total cost* of procurement by making the following decisions: (1) select a set of winning bidders $J' \subseteq J$ and (2) determine the trading quantity q_j for each winning bid $j \in J'$. The above decisions are to be made subject to the following constraints:

1) **Supply Constraint**: For every winning bid $j \in J'$, $q_j \in [\underline{a}_j, \overline{a}_j]$, and for losing bids, $q_j = 0$.
2) **Demand Constraint**: The total quantity procured should satisfy the demand of the buyer: $\sum_{j \in J'} q_j \geq \underline{b}$.
3) **Business Constraint**: The number of winning bids should be in the given range: $|J'| \in [\underline{W}, \overline{W}]$.

The cost function Q_j of Fig. 1 is nonlinear but due to the piecewise linear nature, the bid evaluation problem can be modeled as the following mixed integer linear programming problem.

$$\text{(PROC)}: \ \min \sum_j \left(n_j^0 d_j^0 + \sum_{s=1}^{l_j} \left(n_j^s d_j^s + \beta_j^s \delta_j^s x_j^s \right) \right) \quad (1)$$

subject to

$$d_j^1 \leq d_j^0 \qquad \forall j \in J \quad (2)$$

$$x_j^s \leq d_j^s \qquad \forall j \in J;\ 1 \leq s \leq l_j \quad (3)$$

$$x_j^s \geq d_j^{s+1} \qquad \forall j \in J;\ 1 \leq s < l_j \quad (4)$$

$$\sum_{j \in J} \left(\underline{a}_j d_j^0 + \sum_{s=1}^{l_j} \delta_j^s x_j^s \right) \geq \underline{b} \quad (5)$$

$$\underline{W} \leq \sum_{j \in J} d_j^0 \leq \overline{W} \quad (6)$$

$$d_j^s \in \{0, 1\} \qquad \forall j \in J;\ 0 \leq s \leq l_j$$

$$x_j^s \in [0, 1] \qquad \forall j \in J;\ 1 \leq s \leq l_j$$

The decision variables x_j^s denote the fraction of goods chosen from the linear segment s of bid j. For this setup to make sense, whenever $x_j^s > 0$ then $x_j^{s-1} = 1$, for all s. To enable this, binary decision variables d_j^s for each segment are used, which denote the selection/rejection of segment s of bid j. The winning quantity for bid j is $\underline{a}_j d_j^0 + \sum_{s=1}^{l_j} \delta_j^s x_j^s$ with cost $n_j^0 d_j^0 + \sum_{s=1}^{l_j} (n_j^s d_j^s + \delta_j^s \beta_j^s x_j^s)$. The binary decision variable d_j^0 is also used as an indicator variable for selecting or rejecting a bid, as $d_j^0 = 0$ implies that no quantity is selected for trading from bid j. This is used for the business constraint (6). The bid evaluation problem was shown to be $\mathcal{NP}$-hard upon reduction from the knapsack problem in [5]. In the following, a Lagrangian relaxation based heuristic is proposed to find a near-optimal solution for the bid evaluation problem.

III. Lagrangian Based Heuristic

The Lagrangian problem for PROC by dualizing the demand constraint is:

$$(\mathrm{PR}_\pi): \quad \min \sum_j \left(n_j^0 d_j^0 + \sum_{s=1}^{l_j} \left(n_j^s d_j^s + \beta_j^s \delta_j^s x_j^s \right) \right) + \pi \left(\underline{b} - \sum_j \left(\underline{a}_j d_j^0 + \sum_{s=1}^{l_j} \delta_j^s x_j^s \right) \right) \quad (7)$$

subject to the constraints (2), (3), (4), and (6).

A. Solving the Lagrangian Problem

It can be easily seen that the x_j^s are redundant in PR_π, as there will always exist an optimal solution with $x_j^s = d_j^s$. Thus the PR_π can be reformulated as:

$$(\mathrm{PR}_\pi): \qquad \min \sum_{j \in J} \sum_{s=0}^{l_j} (\hat{\beta}_j^s - \pi) \delta_j^s \hat{x}_j^s + \pi \underline{b} \qquad (8)$$

subject to

$$\hat{x}_j^s \geq \hat{x}_j^{s+1} \qquad \forall j \in J;\ 0 \leq s < l_j$$
$$\underline{W} \leq \sum_{j \in J} \hat{x}_j^0 \leq \overline{W}$$
$$\hat{x}_j^s \in \{0,1\} \qquad \forall j \in J;\ 0 \leq s \leq l_j$$

where $\hat{\beta}_j^0 = n_j^0 / \underline{a}_j$, $\hat{\beta}_j^s = (n_j^s + \beta_j^s \delta_j^s)/\delta_j^s$ for $s > 0$, and $\delta_j^0 = \underline{a}_j$. By assigning $d_j^0 = \hat{x}_j^0$ and $d_j^s = x_j^s = \hat{x}_j^s$, one can easily verify that the above two formulations are the same and yield the same optimal value. Let $N = |J|$ be the total number of bids submitted.

Algorithm 1: Algorithm to solve the Lagrangian problem PR_π for a given $\pi \geq 0$

1) (Initialize)
   $\hat{x}_j^s = 0,\ 0 \leq s \leq l_j,\ \forall j \in J$;
   $J'' = J$; $J' = \emptyset$; $c_\pi = \pi \underline{b}$;
2) $\forall j \in J$ **do**:
   a) $r_j(\pi) = \arg \min_{1 \leq t \leq l_j} \{ \sum_{s=0}^{t} (\hat{\beta}_j^s - \pi) \delta_j^s \}$;
   b) $p_j = \sum_{s=0}^{r_j(\pi)} (\hat{\beta}_j^s - \pi) \delta_j^s$;

   od;
3) **while** ($|J''| > N - \underline{W}$) **do:**
   a) $k = \arg\min \{ p_j : j \in J'' \}$;
   b) $x_k^s \leftarrow 1,\ 0 \leq s \leq r_k(\pi)$;
   c) $c_\pi \leftarrow c_\pi + p_k$; $J'' \leftarrow J'' \setminus \{k\}$; $J' \leftarrow J' \cup \{k\}$;

   od;
4) **while** ($|J''| > N - \overline{W}$) **do:**
   a) $k = \arg\min \{ p_j : j \in J'' \}$;
   b) **if** ($p_k \leq 0$)
      then $x_k^s \leftarrow 1,\ 0 \leq s \leq r_k(\pi)$;
      $c_\pi \leftarrow c_\pi + p_k$; $J'' \leftarrow J'' \setminus \{k\}$;
      $J' \leftarrow J' \cup \{k\}$;
      else break;
      fi;

   od;

In the case of multiple solutions in Step 2(a), the maximum subscript is chosen to accommodate as much quantity as possible. The optimality of the algorithm can be verified as follows. For a given $\pi \geq 0$, the *potential* contribution p_j to the final solution from bid j is calculated in Step 2. Note that the minimum is taken over the segments $1 \leq s \leq l_j$ and the indivisible segment is considered as mandatory. This is required because if bid j is a winning bid then the $\underline{a}_j$ from the indivisible segment is the minimum acceptable amount. In Step 3, the first $\underline{W}$ bids with least p_j are selected. This satisfies the business constraint for the lower limit on the winning bids. Step 4 accepts bids with non-positive contribution till the upper limit is reached. Thus the objective value can only decrease in Step 4, while maintaining feasibility. The winning bids are J' and the objective value $Z(\mathrm{PR}_\pi) = c_\pi$. Steps 3 and 4 together require first $\overline{W}$ bids to be in sorted order. The possible algorithms for implementing this are:

- The algorithm *median of medians* [6] that finds the kth-order statistics in $O(N)$. To find the first $\overline{W}$ sorted values, the complexity is $O(\overline{W}N)$.
- Construct a *binary heap* [7] with N p_j's in $O(N)$ time and remove the first $\overline{W}$ minimum elements from it. The complexity is $O(N + \overline{W} \log N)$ as the complexity of insertion is $O(\log N)$.
- Perform *insertion sort* [6] for $\overline{W}$ iterations and the complexity is $O(\overline{W}N)$.

The insertion sort is the easiest of the three to implement. Let $L = \sum_{j \in J} lj$ be the total number of linear segments. Step 2 takes $O(L)$ time and since L and $\overline{W}$ are $O(N)$, the worst case complexity of the algorithm is $O(N \log N)$.

B. Solving the Lagrangian Dual

The Lagrangian dual problem that determines the lower bound to PROC is

$$(\mathrm{LD}:\mathrm{PROC}): \qquad \max_{\pi \geq 0} Z(\mathrm{PR}_\pi)$$

The Lagrangian dual problem is a convex optimization problem as $Z(\mathrm{PR}_\pi)$ is a piecewise linear concave function of π. The prominent solution technique for solving the Lagrangian dual problem is *subgradient optimization* algorithm [8], which does not guarantee an optimal solution in finite iterations. An exact algorithm is proposed below for solving the Lagrangian dual by explicitly enumerating the break points π at which $Z(\mathrm{PR}_\pi)$ changes its slope.

Let $\hat{b}(\pi)$ denote the total quantity of good procured and let W_π denote the set of winning bids for a given π. Then $\hat{b}(\pi) = \sum_{j \in W_\pi} \sum_{s=0}^{s=r_j(\pi)} \delta_j^s$. For a given $\pi \geq 0$, the $Z(\mathrm{PR}_\pi)$

can be re-written as

$$Z(\mathrm{PR}_\pi) = \sum_{j \in W_\pi} \sum_{s=0}^{r_j(\pi)} \hat{\beta}_j^s \delta_j^s + \pi(\underline{b} - \hat{b}(\pi)) \tag{9}$$

Let π^1 be the current value of π. Then the slope of $Z(\mathrm{PR}_\pi)$ will change for $\pi \geq \pi^1$ whenever $\hat{b}(\pi)$ or W_π changes. The $\hat{b}(\pi)$ may change due to any of the following reasons: (a) The $r_j(\pi)$ is changed for $j \in W_{\pi^1}$, (b) A losing bid becomes a winning bid, and (c) A losing bid replaces a winning bid. The cases (b) and (c) change the set of winning bids W_{π^1}: case (b) happens if $|W_{\pi^1}| < \overline{W}$ and case (c) is possible if $|W_{\pi^1}| = \overline{W}$. The conditions under which the above changes will occur are identified first and then the algorithm is proposed based on those conditions.

Consider a bid j, irrespective of it being a winning or a losing bid. One can easily verify that for any bid j, $r_j(\pi)$ is a non-decreasing function of π. Let at exactly $\pi = \pi^2$, $r_j(\pi^1)$ be increased to $r_j(\pi^2)$. Thus the cost added from the segments $r_j(\pi^2) - r_j(\pi^1)$ is minimal among the cost added from the segments $t - r_j(\pi^1)$, for, $r_j(\pi^1) < t \leq l_j$.

$$r_j(\pi^2) = \arg\min\{ \sum_{s=r_j(\pi^1)+1}^{t} (\hat{\beta}_j^s - \pi^2)\delta_j^s : \ r_j(\pi^1) < t \leq l_j\}$$

If there are multiple solutions, then the segment with the maximum index is chosen to accommodate as much quantity as possible. The added cost should be exactly zero for the $r_j(\pi^1)$ to change. If it is greater than zero then $r_j(\pi^1)$ cannot change and if it is less than zero it would have already changed. Thus, the π^2 is:

$$\pi^2 = \min\left\{ \frac{\sum_{s=r_j(\pi^1)+1}^{t} \hat{\beta}_j^s \delta_j^s}{\sum_{s=r_j(\pi^1)+1}^{t} \delta_j^s} : \ r_j(\pi^1) < t \leq l_j \right\}$$

Thus for a given π, one can easily find the current $r_j(\pi)$ and the next π at which $r_j(\pi)$ will increase.

Now consider the cases (b) and (c), where the set of winning bids W_π changes. Let at π^1, $W_{\pi^1} < \overline{W}$. Then a losing bid j has contribution $\sum_{s=0}^{r_j(\pi^1)} (\hat{\beta}_j^s - \pi^1)\delta_j^s > 0$. Whenever the contribution becomes zero, then it can become a winning bid. Thus a losing bid j can become a winning bid (assuming no change in $r_j(\pi)$) if there exists a $\pi'' > 0$ such that

$$\pi'' = \frac{\sum_{s=0}^{r_j(\pi^1)} (\hat{\beta}_j^s - \pi^1)\delta_j^s}{\sum_{s=0}^{r_j(\pi^1)} \delta_j^s}$$

Clearly at $\pi^1 + \pi''$ the contribution is zero and hence the bid j can become a winning bid (if at $\pi^1 + \pi''$ the set of winning bids is less than $\overline{W}$).

For case (c), the set of winning bids is equal to $\overline{W}$. Whenever a losing bid replaces a winning bid, there will possibly be a change in the total winning quantity. This is shown in Fig. 2. Consider the four bids (A), (B), (C), and (D) shown in the figure. Assume that there can be only two winning bids. Then, at π^0, (A) and (B) are the winning bids and (C) and (D) are the losing bids. The y-axis shows the contribution $p_j(\pi)$ of bid j to the objective value. Also assume that there will be no change in the $r_j(\pi)$ for any of these bids in the region $[\pi^0, \pi^4]$. Then, at π^1, (D) replaces (B) as a winning bid. Till π^2, (A) and (D) are the winning bids, but at again π^2, (B) replaces (A) as the winning bid. At π^3, (B) is replaced by (C). Let q_j denote the winning quantity of bid j for $\pi \in [\pi^0, \pi^4]$. Note that by virtue of the assumption of no change in $r_j(\pi)$ for $\pi \in [\pi^0, \pi^4]$, the q_j is a constant in the region of interest. Hence, $p_j(\pi)$ will be a linear function in this region (as shown in the Fig. 2) with q_j as the slope. Consider bids (B) and (D). At $\pi = \pi^0$, their contribution is $p_{(B)}(\pi^0)$ and $p_{(D)}(\pi^0)$, respectively and $p_{(B)}(\pi^0) < p_{(D)}(\pi^0)$. But at $\pi = \pi^1$, their contribution is the same. The $\pi^1 = \pi^0 + \pi''$ can be found from

$$\pi'' = \frac{p_{(D)}(\pi^0) - p_{(B)}(\pi^0)}{q_{(D)} - q_{(B)}}$$

The (D) can replace (B) as winning bid at $\pi = \pi^1$ only if $q_{(D)} > q_{(B)}$ (that is $\pi'' = 0$). Therefore at π^1, $q_{(D)} - q_{(B)}$ is added to the winning quantity and hence the slope of PR_π changes. Thus, whenever a winning bid is replaced by a losing bid, the winning quantity increases.

Let $\pi_j^{(a)}$, $\pi_j^{(b)}$, and $\pi_j^{(c)}$ denote the next π's at which, the cases (a), (b), and (c), respectively, are realized for bid j. If any of them are negative, then they can be removed from consideration. The next π at which the $Z(\mathrm{PR}_\pi)$ would possibly change is:

$$\pi = \min_j\{\min\{\pi_j^{(a)}, \pi_j^{(b)}, \pi_j^{(c)}\}\}$$

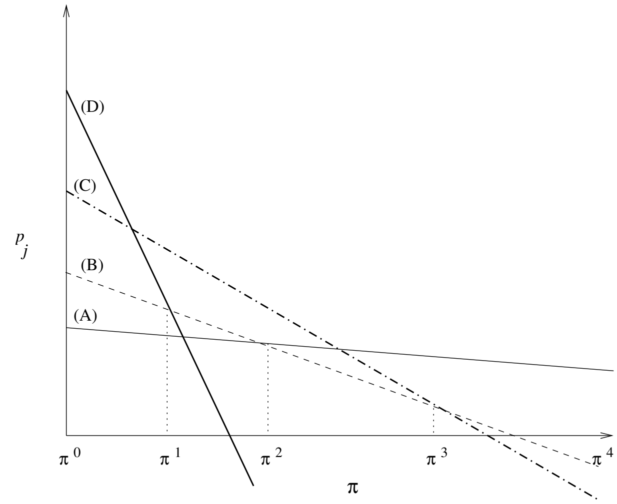

Fig. 2. Change in the set of winning bids

TABLE I

FUNCTIONS USED FOR SOLVING THE LAGRANGIAN DUAL OF PROC

Function	Input	Output	Description
`lag_pr`	π, R	WB, b	Using the Algorithm 1, this function finds the winning set of bids WB and the total winning quantity for a given input π and R.
`next_pi_a`	π, R	π, R	Given the current π and R, this function finds the next $\tilde{\pi}_j$ for all $j \in J$, which is the next π at which new segments are added to the solution. The next π at which the objective function changes slope is $\min_j \tilde{\pi}_j$. The new π and R are suitably updated and returned as outputs.
`next_pi_bc`	π, R, WB	π	This function finds the next π at which the set of winning bids WB changes (due to cases (b) and (c)) for the given R.

Proposition 1: $\hat{b}(\pi)$ is a non-decreasing function of π and optimal $\pi^* = \arg\min_{\pi \geq 0}\{\hat{b}(\pi) \geq \underline{b}\}$.

Proof: The $\hat{b}(\pi)$ changes due to change in $r_j(\pi)$ or W_π and in both cases $\hat{b}(\pi)$ increases. The $Z(\mathrm{PR}_\pi)$ is a piecewise linear concave function and the slope changes whenever $\hat{b}(\pi)$ changes. Let there be no change in $\hat{b}(\pi)$ for $\pi \in [\pi^1, \pi^2]$. Then for $Z(\mathrm{PR}_\pi)$ (given by (9)) in range $[\pi^1, \pi^2]$, $\underline{b} - \hat{b}(\pi)$ is the slope. The function is increasing if $\underline{b} > \hat{b}(\pi)$ and decreasing if $\underline{b} < \hat{b}(\pi)$. Thus at $\pi = \pi^*$, when $\hat{b}(\pi) \geq \underline{b}$, the $Z(\mathrm{PR}_\pi)$ changes direction from increasing to decreasing, which is the optimal objective value of the Lagrangian dual. ∎

An algorithm for solving the Lagrangian dual is proposed below. Let $R = \{(j, r_j, p_j) : \forall j \in J\}$ denote the set that contains information about the current r_j segment and the cost p_j of bid $j \in J$. The functions used in the algorithm are described in Table I.

Algorithm 2: Algorithm to solve the Lagrangian Dual of PROC

1) (Initialize) $\pi = \pi' = 0$; $b = b' = 0$; $WB = \emptyset$; $R = R' = \{(j, 0, n_j^0) : \forall j \in J\}$
2) `lag_pr`(π, R, WB, b);
3) **while** ($b < \underline{b}$) **do**:
   a) $\pi' \leftarrow \pi$; $b' \leftarrow b$; $R' \leftarrow R$;
   b) `next_pi_a`(π, R);
   c) `lag_pr`(π, WB, b);

   od;
4) **while** ($b' < \underline{b}$) **do**:
   a) `next_pi_bc`(π', R', WB);
   b) **if** ($\pi' \geq \pi$) **or** ($\pi' = $ *null*) **break fi**;
   c) `lag_pr`(π', WB, b');

   od;

The algorithm starts with $\pi = 0$ and $r_j = 0$ for $j \in J$ (as the indivisible segment is mandatory). Step 3 finds the π at which new segments are added to the winning bids till the demand $\underline{b}$ is satisfied. However, this need not be optimal, as the winning quantity could have been added due to the change in winning bids. This is handled in Step 4. However, if no more change is possible ($\pi' = $ *null*) or the next π value exceeds the already known best π, the search in Step 4 is terminated. The set of winning bids is given in WB. The algorithm does not trace all the πs at which $Z(\mathrm{PR}_\pi)$ changes its slope. In Step 3 it only identifies the break points due to change in $r_j(\pi)$ (due to case (a)). This identifies the segment $[\pi', \pi]$ with quantities $[b', b]$ such that $b' < \underline{b} \leq b$. The optimal π^*, which is in the range $[\pi', \pi]$, is identified in Step 4 by looking for changes in slope due to cases (b) and (c). This would considerably reduce the computational time, as the function `lag_pr` is not evaluated for all break points in the range $[0, \pi^*]$.

C. Heuristic Solution and Optimality Gap

A feasible solution to PROC is obtained from the optimal solution of the Lagrangian dual by assigning $d_j^0 = \hat{x}_j^0$ and $d_j^s = x_j^s = \hat{x}_j^s$. This solution is feasible to PROC as the allocation $\hat{b}(\pi^*)$ for the optimal π^* satisfies the demand ($\hat{b}(\pi^*) \geq \underline{b}$). The above solution is optimal to PROC whenever $\hat{b}(\pi^*) = \underline{b}$:

$$\begin{aligned} Z(\mathrm{LD}:\mathrm{PROC}) &\leq Z(\mathrm{PROC}) \\ \sum_{j \in W_{\pi^*}} \sum_{s=0}^{r_j(\pi^*)} \hat{\beta}_j^s \delta_j^s + \pi^*(\underline{b} - \hat{b}(\pi^*)) &\leq Z(\mathrm{PROC}) \\ \sum_{j \in W_{\pi^*}} \sum_{s=0}^{r_j(\pi^*)} \hat{\beta}_j^s \delta_j^s - Z(\mathrm{PROC}) &\leq \pi^*(\hat{b}(\pi^*) - \underline{b}) \end{aligned}$$

The heuristic solution is improved if $\hat{b}(\pi^*) > \underline{b}$, by removing the excess units $\hat{b}(\pi^*) - \underline{b}$ from the $r_j(\pi^*)$ if it is a divisible segment ($r_j(\pi^*) > 0$). This is implemented by using a greedy strategy: pick the segment $r_j(\pi^*) > 0$ with highest $\beta_j^{r_j(\pi^*)}$ or $\hat{\beta}_j^{r_j(\pi^*)}$ and remove the excess units. This is repeated till the demand is met or no more reduction is possible.

IV. COMPUTATIONAL EXPERIMENTS

The proposed heuristic was tested on three different random problem sets to study the computational time and the

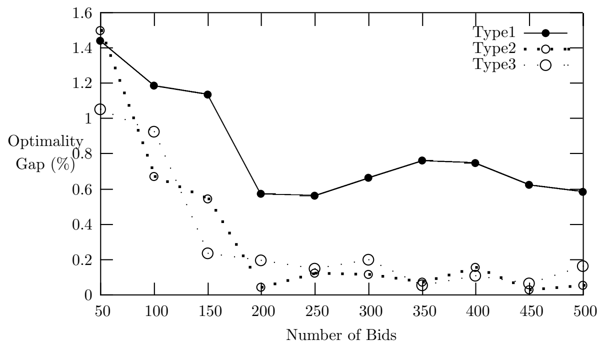

Fig. 3. Optimality gap for the LR based Heuristic

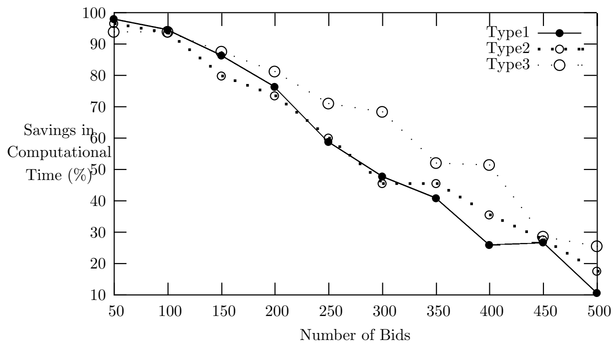

Fig. 4. Savings in time for the LR based Heuristic

optimality gap. The **Type 1** problems had decreasing slopes for the linear segments in the cost function and the function parameters were highly correlated across the items. This resembles bids in procurement auctions where the cost of the item does not vary much across the suppliers. The **Type 2** problems had similar cost functions but the parameters were uncorrelated and the **Type 3** problems had uncorrelated and arbitrary piecewise linear functions. The number of segments was chosen randomly from 3 to 5 for **Type 1** problems and from 3 to 10 for **Type 2** and **Type 3** problems. To determine the optimality gap and the savings in computational time for the heuristic, the problem was solved to optimality using ILOG Concert Technology of CPLEX 9.1 [9]. The experiments were carried out on a Linux based PC equipped with a 3GHz Intel Xeon processor with 4GB RAM and the algorithms were coded in Java. Figures 3 and 4 show the average optimality gap and savings in time taken over 20 problem instances. The optimality gap and the savings in solution time are computed against the direct solution by ILOG CPLEX. The average optimality gap for **Type1** problems was around 0.7% and for **Type2** and **Type3**, it was less than 0.2%. The saving in time steadily declines with the number of bids. This is due to increase in the number of `lag_pr` computations, which involves the sorting of the bids. The sorting procedure was implemented using insertion sort, but use of binary heap data structure [7] would give better performance for problems with a large number of bids. However, the number of `lag_pr` computations remains the same and therefore for scalable performance, an intelligent technique to reduce the number of evaluations of π's is required.

V. CONCLUSIONS

This paper considered the procurement of multiple units of a single good, where the suppliers submit a nonconvex piecewise linear cost function as the bid. The buyer's business constraint of restricting the number of winning suppliers in a preferred range was included as a side constraint in the mixed integer linear programming formulation of the bid evaluation problem. A heuristic based on Lagrangian relaxation was proposed to find a feasible solution to the problem. Computational experiments performed on representative data sets showed that the proposed heuristic produced near optimal solutions, with negligible loss in optimality. By decreasing the number of evaluations of the Lagrangian problem, one can expect further improvement in computational time. Use of intelligent search techniques along with search techniques (for single variable) such as binary, binomial, and Fibonacci will lead to scalable algorithms. Computational results showed that the optimality gap of the feasible solution is very low, which can be used in conjunction with exact enumeration techniques like branch-and-bound for solving the problem to optimality.

REFERENCES

[1] A. Davenport and J. Kalagnanam, "Price negotiations for procurement of direct inputs," IBM Research, Yorktown Heights, NJ, USA, Research Report RC 22078, 2001.
[2] A. Kothari, D. Parkes, and S. Suri, "Approximately strategy proof and tractable multi-unit auctions," in *Proceedings of ACM Conference on Electronic Commerce (EC-03)*, 2003.
[3] M. Eso, S. Ghosh, J. Kalagnanam, and L. Ladanyi, "Bid evaluation in procurement auctions with piece-wise linear supply curves," IBM Research, Yorktown Heights, NJ, USA, Research Report RC 22219, 2001.
[4] S. Kameshwaran and Y. Narahari, "Trade determination in multiattribute exchanges," in *Proceedings of IEEE Conference on E-Commerce CEC' 03*. IEEE Computer Society, 2003, pp. 173–180.
[5] S. Kameshwaran, "Algorithms for piecewise linear knapsack problems with applications in electronic commerce," PhD Thesis, Department of Computer Science and Automation, Indian Institute of Science, Bangalore, 2004.
[6] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *Data Structures and Algorithms*. Addison Wesley, 1983.
[7] T. Cormen, C. Leiserson, and R. Rivest, *Introduction to Algorithms*. Cambridge: MIT Press, 1990.
[8] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*. New York: John Wiley and Sons, 1988.
[9] *ILOG Concert Technology 1.1 User's Manual*, ILOG CPLEX, February 2001.