Dynamic Procurement Subject to Temporal and Capacity Constraints

Jiong Sun[1] and Norman Sadeh[2]
[1]*Graduate School of Industrial Administration,* [2]*School of Computer Science*
Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213, USA
Email: [1]*jiongs@andrew.cmu.edu,* [2]*sadeh@cs.cmu.edu*

Abstract

Reverse auctions offer the prospect of more efficiently matching suppliers and producers in the face of changing market conditions. Prior research has ignored the temporal and finite capacity constraints under which reverse auctioneers typically operate. In this paper, we consider the problem faced by a manufacturer that can procure key components from a number of possible suppliers through multi-attribute reverse auctions. Bids submitted by prospective suppliers include a price and a delivery date. The manufacturer has to select a combination of supplier bids that will maximize its overall profit, taking into account its own finite capacity and the prices and delivery dates offered by different suppliers for the same components. The manufacturer's profit is determined by the revenue generated by the products it sells, the costs of the components it purchases as well as late delivery penalties it incurs if it fails to deliver products in time to its own customers. We provide a formal model of this important class of problems, discuss its complexity and introduce rules that can be used to efficiently prune the resulting search space. We also introduce a branch-and-bound algorithm and an efficient heuristic search procedure. Computational results show that our heuristic procedure typically yields solutions that are within 10 percent of the optimum. They also indicate that taking into account finite capacity considerations can significantly improve the manufacturer's bottom line.

1. Introduction

Today's global economy is characterized by fast changing market demands, short product lifecycles and increasing pressures to offer high degrees of customization, while keeping costs and lead times to a minimum. In this context, the competitiveness of both manufacturing and service companies will increasingly be tied to their ability to identify promising supply chain partners in response to changing market conditions. Today, however dynamic supply chain practices are confined to relatively simple scenarios such as those found in the context of MRO (Maintenance, Repair and Operations) procurement. The slow adoption of these practices and the failure of many early electronic marketplaces can in part be attributed to the one-dimensional nature of early solutions that forced suppliers to compete solely on the basis of price.

Similarly, research in the area has generally ignored key temporal and capacity constraints under which most reverse auctioneers operate. For instance, a PC manufacturer can only assemble so many PCs at once and not all PCs are due at the same time. Such considerations can be used to help the PC manufacturer select among bids from competing suppliers. In this paper, we summarize research aimed at exploiting these temporal and capacity constraints to help a reverse auctioneer select among competing multi-attribute procurement bids that differ in prices and delivery dates. We refer to this problem as the *Finite Capacity Multi-Attribute Procurement* (FCMAP) problem. It is representative of a broad range of practical reverse auctions, whether in the manufacturing or service industry. We start by providing a formal definition of the FCMAP problem, discuss its complexity and introduce several rules that can be used to prune its search space. We then present a branch-and-bound algorithm and a heuristic search procedure along with computational results showing that accounting for the reverse auctioneer's finite capacity can significantly improve its bottom line.

The balance of this paper is organized as follows. Section 2 provides a brief review of the literature. In section 3, we introduce a formal model of the FCMAP problem. Section 4 identifies three rules that can help the reverse auctioneer eliminate non-competitive bids or bid combinations. Section 5 introduces a branch-and-bound algorithm for the

FCMAP problem. Section 6 presents a heuristic search procedure for the FCMAP problem that takes advantage of a property identified in Section 4. Computational results are presented in Section 7, comparing variations of our search heuristic under different bid distributions, measuring distance from optimum and evaluating the impact of taking finite capacity considerations into account. Section 8 provides some concluding remarks and discusses future extensions of this research.

2. Literature review

Few researchers have studied supply chain formation problems in the context of capacity-constrained environments. A notable exception is the work of Gallien and Wein who have proposed a reverse auction mechanism that takes into account supplier capacity constraints [1]. Babaioff and Nisan have designed information exchange protocols that enhance supply chain responsiveness in the face of surges or drops in demand and supply [2]. Their work however assumes infinite production capacity, where an increase in the production volume of one product does not impact the ability of the manufacturer to possibly increase or maintain production levels for other products. Other relevant work includes that of Walsh and Wellman [3], though here again capacity constraints are ignored. Sadeh *et al.,* discuss MASCOT, an agent-based supply chain decision support tool that supports finite capacity models [4]. Their work to date has focused on the empirical study of real-time available-to-promise and profitable-to-promise functionality and on scheduling coordination across static supply chains. Another significant effort in this area is the work carried out by the team of Collins and Gini in the context of MAGNET [5].

3. The finite capacity multi-attribute procurement problem

The Finite Capacity Multi-Attribute Procurement (FCMAP) problem revolves around a reverse auctioneer – referred below as the "manufacturer", though it could also be a service provider. The manufacturer has to satisfy a set of customer commitments or orders O_i, $i\in M=\{1,2,\ldots,m\}$. Each order i needs to be completed by a due date dd_i, and requires one or more components (or services), which the manufacturer can obtain from a number of possible suppliers. The manufacturer has to wait for all the components before it can start processing the order (e.g., waiting for all the components required to assemble a given PC). For the sake of simplicity, we assume that the processing required by the manufacturer to complete work on customer order O_i has a fixed duration du_i, and that the manufacturer can only process one order at a time ("capacity constraint"). It should be noted however that the model and techniques presented in this paper can easily be generalized to accommodate setups or situations where the manufacturer can process multiple orders at the same time.

Formally, for each order O_i and each component $comp_{ij}$, $j\in N_i=\{1,2,\ldots,n_i\}$, the manufacturer organizes a reverse auction for which it receives a set of multi-attribute bids $\beta_{ij}=\{B_1^{ij},\ldots,B_{n_{ij}}^{ij}\}$ from prospective suppliers. Each bid B_k^{ij} includes a bid price bp_k^{ij} and a proposed delivery date dl_k^{ij}. Below we use the notation $B_k^{ij}=(dl_k^{ij},bp_k^{ij})$.

Failure by the manufacturer to meet an order O_i's due date results in a penalty $tard_i\times T_i$, where T_i is the time by which delivery of the product or service is late, and $tard_i$ is the marginal penalty for missing the delivery date. Such penalties, which are commonly used to model manufacturing scheduling problems, reflect actual contractual terms, loss of customer goodwill, interests on lost profits or a combination of the above [6].

A solution to the FCMAP problem consists of:

- a selection of bids:
  $\{Bid\_Comb_i,\ldots,Bid\_Comb_m\}$,
  where Bid_Comb_i ($i\in M$) is a combination of n_i bids - one for each of the components required by order O_i, and
- a collection of start times: $ST=\{st_1,\ldots,st_m\}$, where st_i is the time when the manufacturer is scheduled to start processing order O_i, and $st_i\geq dl^{ij},\forall j=1,..,n_i$, since orders cannot be processed before all the components they require have been delivered by suppliers.

Given a solution (Bid_Comb,ST), the profit of the manufacturer is the difference between the revenue generated by its customer orders (once they have been completed) and the sum of its procurement costs and tardiness penalties. This is denoted:

$$prof(Bid\_Comb,ST) = \sum_{i\in M} rev_i - \sum_{i\in M}\sum_{j\in N_i} bp^{ij} - \sum_i tard_i\times T_i \qquad (1)$$

where,

- rev_i is the revenue generated by the completion of order O_i (i.e., the amount paid by the customer),
- bp^{ij} is the price of component $comp_{ij}$ in $Bid\_Comb,$

- $T_i = Max(0,\ st_i + du_i - dd_i)$ with st_i being the start time of order O_i in ST.

Note that because we assume a given set of orders, the term $\sum_{i \in M} rev_i$ is the same across all solutions. Accordingly, maximizing profit in Equation (1) is equivalent to minimizing the sum of procurement and tardiness costs: $\sum_{i \in M} \sum_{j \in N_i} bp^{ij} + \sum_{i \in M} tard_i \times T_i$.

It is worth noting that the above model contrasts with earlier research in dynamic supply chain formation [7, 5, 3], which has generally assumed manufacturers with infinite capacity or fixed lead times and ignored delivery dates and tardiness penalties.

From a complexity standpoint, it can easily be seen that the FCMAP problem is strongly NP-hard, since the special situation where all components are free and available at time zero reduces to the single machine total weighted tardiness problem, itself a well known NP-hard problem [8].

An example of an exact procedure to solve FCMAP problems involves looking at all possible procurement bid combinations and, for each such combination, solving to optimality a single machine weighted tardiness problem with release dates (e.g., using a branch-and-bound algorithm). A release date is a date before which a given order is not allowed to be processed. Given a combination of procurement bids Bid_Comb_i, an order O_i has a release date:

$$r_i = \underset{j \in N_i}{Max}[dl^{ij}] \tag{2}$$

where dl^{ij} denotes the delivery date of component $comp_{ij}$ in Bid_Comb_i. In other words, the component that arrives the latest determines the order's release date.

Clearly, with the exception of fairly small problems, the requirements of the above procedure are computationally prohibitive. Instead, we identify below a number of rules that can be used to efficiently prune the search space associated with FCMAP problems.

4. Pruning the search space

Pruning Rule 1: Eliminating expensive bids with late delivery dates

Consider an FCMAP problem P with an order O_i requiring a component $comp_{ij}$ for which the manufacturer has received a set of bids $\beta_{ij} = \{B_1^{ij}, \ldots, B_{n_{ij}}^{ij}\}$ from prospective suppliers. Let $B_k^{ij} = (dl_k^{ij}, bp_k^{ij})$ and $B_l^{ij} = (dl_l^{ij}, bp_l^{ij})$ be two bids in β_{ij} such that:

$$dl_l^{ij} \geq dl_k^{ij} \text{ and } bp_l^{ij} \geq bp_k^{ij}.$$

Then problem P' with *$\beta'_{ij} = \beta_{ij} \setminus \{B_l^{ij}\}$ admits optimal solutions with the exact same profit as problem P.*

The correctness of this rule should be obvious. Its application is illustrated in Figure 1, where an order requires two components: component 1 and component 2. The manufacturer has received bids for each component. Using Rule 1, it can be determined, for instance, that bid_{14} is not competitive given that it is more expensive than bid_{13} and arrives later. Similarly, bid_{22} and bid_{24} can also be pruned.

Pruning Rule 2: Eliminating expensive bids with unnecessarily early delivery dates

Consider an FCMAP problem P with an order O_i requiring a set of components $comp_{ij}$, $j \in N_i = \{1,2,\ldots,n_i\}$. Let $\beta_{ij} = \{B_1^{ij}, \ldots, B_{n_{ij}}^{ij}\}$ be the set of bids received by the manufacturer for each component $comp_{ij}$ with $B_k^{ij} = (dl_k^{ij}, bp_k^{ij})$. We define $r_i^{earliest}$ as the earliest possible release date for order O_i. It can be computed as:

$$r_i^{earliest} = \underset{1 \le j \le n_i}{Max}\ \underset{1 \le k \le n_{ij}}{Min}\ dl_k^{ij}$$

Let B_k^{ij} and B_l^{ij} be two bids for component $comp_{ij}$ such that:

$$bp_l^{ij} \geq bp_k^{ij} \text{ and } dl_l^{ij} \leq dl_k^{ij} \leq r_i^{earliest}.$$

Then problem P' with $\beta'_{ij} = \beta_{ij} \setminus \{B_l^{ij}\}$ admits the exact same set of optimal solutions as problem P.

The correctness of this rule should be obvious. Application of this rule is also illustrated in Figure 1, where it results in the pruning of bid_{11}. This is because both bid_{11} and bid_{12} arrive before the order's earliest release date, $r^{earliest}$, and bid_{11} is more expensive than bid_{12}.

Pruning Rule 3: Eliminating expensive bid combinations with unnecessarily early delivery dates

Consider an FCMAP problem P whose search space has already been pruned using Rule 1. In other words, given two bids $B_k^{ij} = (dl_k^{ij}, bp_k^{ij})$ and $B_l^{ij} = (dl_l^{ij}, bp_l^{ij})$, $l \neq k$, for the same component $comp_{ij}$, if $dl_l^{ij} > dl_k^{ij}$, then $bp_l^{ij} < bp_k^{ij}$.

Let $Bid\_Comb_i^a = \{B_a^{i1}, \ldots, B_a^{in_i}\}$ *be a combination of bids for the* n_i *components required by order* O_i. *Suppose also that there exist two bids* $B_a^{ik} = (dl_a^{ik}, bp_a^{ik}) \in Bid\_Comb_i^a$ *and* $B_b^{ik} = (dl_b^{ik}, bp_b^{ik})$ *such that:*

$$dl_a^{ik} < dl_b^{ik} \le dl_a^{il},$$

then $Bid\_Comb_i^a$ *is dominated by* $Bid\_Comb_i^b$, *where* $Bid\_Comb_i^b = (Bid\_Comb_i^a \setminus \{B_a^{ik}\}) \cup \{B_b^{ik}\}$. *By "dominated" we mean that, for every solution to problem P involving* $Bid\_Sel_i^a$, *there is a better solution where* $Bid\_Comb_i^a$ *is replaced by* $Bid\_Comb_i^b$.

Given that $Bid\_Comb_i^a$ includes a bid for a second component $comp_{ij}$ that gets delivered at time $dl_a^{il} \ge dl_b^{ik} > dl_a^{ik}$, replacing bid B_a^{ik} with bid B_b^{ik} will not delay the start of order O_i and can only help reduce the cost of its components since $bp_a^{ik} > bp_b^{ik}$ (as indicated earlier, we assume that Rule 1 has already been applied to prune bids). It is easy to build a formal proof based on the above observation. Note that Rule 3 subsumes Rule 2.

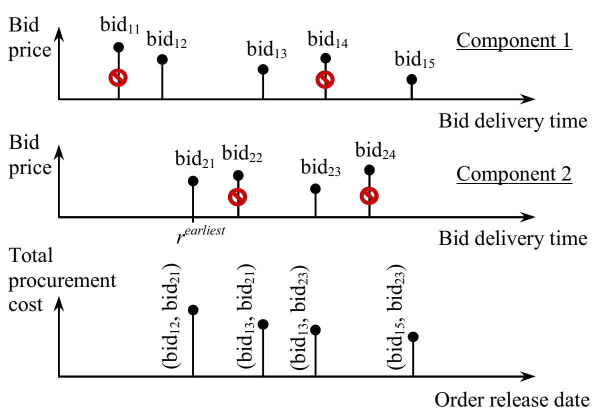

Figure 1. From 20 bid combinations to 4 non-dominated ones

The three pruning rules we just identified can be used to prune the set of bids to be considered. This is illustrated in Figure 1, where the combination of the three rules brings the number of bid combinations to be considered from 20 to just 4 non-dominated combinations. In particular, the application of Rule 3 helps us prune bid combination (bid_{12}, bid_{23}). This is because this combination is dominated by (bid_{13}, bid_{23}), which results in the same release date but is cheaper. Another bid combination pruned using Rule 3 is (bid_{15}, bid_{21}). It should be clear that, for each order, Rules 1 and 2 can be applied in $O(c \cdot b \cdot \log b)$ time, where b is an upper-bound on the number of bids received for a given component and c an upper-bound on the number of components required by a given order. It can also be shown that, for a given order O_i, Rule 3 can be applied in $O(tb \cdot \log tb)$ time, where tb is the total number of bids received for order O_i across all the components it requires [9].

Consider the non-dominated bid combinations resulting from the application of our three pruning rules to an FCMAP problem. Let the non-dominated bid combinations of order O_i be denoted

$$Bid\_Comb_i^* = \{Bid\_Comb_i^1 = (r_{i1}, pc_{i1}), \ldots, Bid\_Comb_i^{m_i} = (r_{im_i}, pc_{im_i})\},$$

where r_{ik} is the release date of bid combination $Bid\_Comb_i^k$, as defined in Equation (2), and pc_{ik} is its total procurement cost, defined as the sum of its component bid prices. It follows that:

Property 1: *For each order* O_i, $i \in M$, *it must hold that, if* $r_{ia} < r_{ib}$, *then* $pc_{ia} > pc_{ib}$, $\forall a, b \in \{1, \ldots, m_i\}$, $a \ne b$. *In other words, the total procurement costs of non-dominated bid combinations strictly decrease as their release dates increase*

<u>*Proof*</u>: We have already shown that, following the application of Rule 1, the bids that remain for a given component have prices that strictly decrease as their delivery dates increase.

Let $Bid\_Comb_i^a$ be a non-dominated bid combination for order O_i - following the application of Rules 1 through 3. Let its release date r_{ia} be determined by the delivery date of component *j*, namely $r_{ia} = dl_a^{ij}$. Note that, by definition, the release date of a bid combination is always determined by one or more of its components. Given that Rule 3 has already been applied, the delivery date dl_a^{ik} of any component *k* must be the latest delivery date among those bids for component *k* that satisfy $dl_a^{ik} \le dl_a^{ij}$.

Consider another non-dominated bid combination $Bid\_Comb_i^b$ for order O_i such that $r_{ib} > r_{ia}$. Let *l* be the index of one of the components determining the release date of bid combination $Bid\_Comb_i^b$, namely $r_{ib} = dl_b^{il} > r_{ia} = dl_a^{ij}$. Just as for bid combination $Bid\_Comb_i^a$, the fact that Rule 3 has been applied implies that the delivery date dl_b^{ik} of any component k in $Bid\_Comb_i^b$ must be the latest delivery date among those bids for component k that satisfy $dl_b^{ik} \leq dl_b^{il}$. Given that $r_{ib} = dl_b^{il} > r_{ia} = dl_a^{ij}$, it also follows that, for any component k, we have $dl_b^{ik} \geq dl_a^{ik}$ with a strict inequality for at least one component, namely component l. Given that Rule 1 has been applied, it also follows that $bp_a^{ik} \leq bp_b^{ik}$ with a strict inequality for at least one component (component l). Hence,

$$pc_{ia} = \sum_{1 \leq k \leq n_i} bp_a^{ik} > pc_{ib} = \sum_{1 \leq k \leq n_i} bp_b^{ik}. \quad \square$$

Property 1 is illustrated in Figure 2, where we have two bid combinations $Bid\_Comb_i^a$ and $Bid\_Comb_i^b$ for an order O_i that requires three components. In this particular example, r_{ib} is determined by the delivery date of component 3, while r_{ia} is determined by that of component 2. The two bid combinations share the same delivery dates for two out of three of the components required by order O_i: components 1 and 2. The difference in procurement cost comes from the higher price associated with the later delivery of component 3 in bid combination $Bid\_Comb_i^b$ (namely, dl_b^{i3} = $r_{ib} > dl_a^{i3}$).

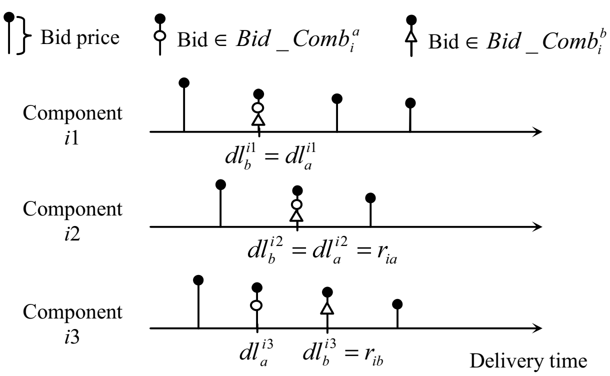

Figure 2. Illustration of Property 1

In the following sections, we introduce a branch- and -bound algorithm to solve the FCMAP problem along with a (significantly faster) heuristic search procedure that takes advantage of Property 1. Both procedures take advantage of the pruning rules we just introduced.

5. A branch-and-bound algorithm

Following the application of the pruning rules introduced in the previous section, optimal solutions to the FCMAP problem can be obtained using a branch-and-bound procedure. Branching is done over the sequence in which orders are processed by the manufacturer and over the release dates of non-dominated bid combinations of each order. Specifically, the algorithm first picks an order to be processed by the manufacturer then tries all the release dates (of non-dominated bid combinations) available for this order. Note that, as orders are sequenced in this fashion, some of their available release dates become dominated, given prior sequencing decisions. For instance, consider two orders O_1 and O_2, with O_2 having two release dates $r_{21}<r_{22}$ - following the application of Pruning Rules 1 through 3. Suppose that, at the current node, O_1 is sequenced before O_2 and that O_1's earliest completion date is greater than r_{22}. It follows that release date r_{21} is strictly dominated by release date r_{22} at this particular node. Release dates that become dominated as a result of prior assignments can be pruned on the fly, thereby further speeding up the search procedure. Given a node n in the search tree, namely a partial sequence of orders and a selection of release dates for each of the orders already sequenced, it is possible to compute an upper-bound for the profit of all complete solutions (i.e. leaf nodes) compatible with this node:

$$UB_n = \sum_{i \in M} rev_i - \sum_{i \in OS_n} (pc_i + tard_i \times T_i)$$

$$- \sum_{i \notin OS_n} [tard_i \times \max(0, cd_{OS_n} + du_i - dd_i) + mpc_i],$$

where,

- OS_n is the set of orders sequenced at node n;
- pc_i is the total procurement cost associated with the non-dominated release date (or bid combination) assigned to order $O_i \in OS_n$ and T_i is its tardiness. Note that each order is scheduled to start as early as possible, given prior sequencing decisions and the release date assigned to it: there are no benefits to starting later;
- cd_{OS_n} is the completion date of the last order in OS_n;
- mpc_i is the minimum possible procurement cost of order O_i - this cost is node-independent.

If the upper bound of a node n is lower than the best feasible solution found so far, the node n and all its descendants are pruned.

6. An Early/Tardy search heuristic

Property 1 tells us that, following the application of the pruning rules, the procurement costs of non-dominated bid combinations strictly decrease as release dates increase. Figure 3 plots the total procurement cost and tardiness cost of an order for different possible start times. While tardiness costs increase linearly for start times that miss the order's due date, procurement costs vary according to a decreasing step function. Specifically, the circles in Figure 3 represent the order's non-dominated bid combinations. For instance, if the order starts at time t, its procurement cost is pc_i, namely, the procurement cost of the latest non-dominated bid combination Bid_Comb^i compatible with this start time. Its tardiness cost is equal to $tard \times Max(0, t+du-dd)$, where $tard$ is its marginal tardiness penalty, dd its due date and du its duration (or processing time). The end result is an early/tardy scheduling problem with non-linear earliness costs.

Ow and Morton have introduced an early/tardy dispatch rule for one-machine scheduling problems subject to linear earliness and tardiness costs [10]. Because our earliness costs are not linear, this heuristic can not readily be applied. Below, we briefly review some of its key elements and discuss how we have adapted it to produce a family of heuristic search procedures for the FCMAP problem.

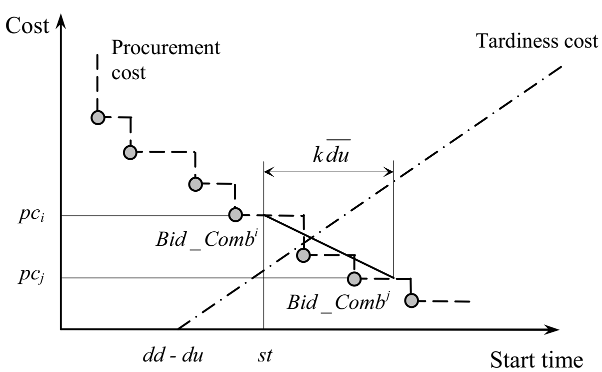

Figure 3. An order's tardiness and procurement costs

Ow and Morton's rule essentially interpolates between two extreme cases. The first extreme situation is one where all orders are assumed to have plenty of time and where only earliness costs need to be minimized. The second situation is one where all orders are assumed to be late and where only tardiness needs to be minimized. In the former case, it can be shown that an optimal solution can be built by sequencing orders according to a *Weighted Longest Processing Time* dispatch rule of the form:

$$P_i(S_i) = -earl_i / du_i ,$$

where $P_i(S_i)$ is the priority of order O_i, S_i is its slack at time t (defined as $dd_i - du_i - t$), du_i its processing time and $earl_i$ its marginal earliness cost – namely the penalty incurred for every unit of time the order finishes before its due date. Conversely, in the latter case, when all jobs are assumed to be late, it can be shown that an optimal solution can be built by sequencing orders according to the *Weighted Shortest Processing Time* dispatch rule, where each order receives a priority:

$$P_i(S_i) = tard_i / du_i .$$

Ow and Morton's dispatch rule interpolates between these two cases by assigning to each order an early/tardy priority that varies with its slack:

$$P_i(S_i) = \begin{cases} \dfrac{tard_i}{du_i} & \text{if } S_i \le 0, \\ \dfrac{tard_i}{du_i}\exp\!\left(-\dfrac{tard_i + earl_i}{earl_i}\cdot\dfrac{S_i}{k\cdot\overline{du}}\right) & \text{if } 0 < S_i \le \dfrac{tard_i}{tard_i + earl_i}k\overline{du}, \\ (earl_i)^{-2}\left(\dfrac{tard_i}{du_i} - \dfrac{tard_i + earl_i}{du_i}\cdot\dfrac{S_i}{k\cdot\overline{du}}\right)^3 & \text{if } \dfrac{tard_i}{tard_i + earl_i}k\overline{du} < S_i \le k\overline{du}, \\ -\dfrac{earl_i}{du_i} & \text{otherwise,} \end{cases} \quad (3)$$

where $\overline{du}$ is the average processing time of an order and k is a look-ahead parameter. This parameter can intuitively be thought of as the average number of orders that will typically get processed ahead of an order queueing in front of the machine. The above formula can easily be seen to reduce to the Weighted Shortest Processing Time dispatch rule when slack $S_i=0$ and to the Weighted Longest Processing Time dispatch rule when $S_i \to \infty$. The value of the look-ahead parameter k controls the transition between these two extremes, with higher values of k making the transition start earlier.

In the FCMAP problem however, orders cannot start before their earliest possible release date (e.g. see Pruning Rule 2). In addition, earliness costs vary according to a step function. A marginal earliness cost can however be obtained through regression, whether locally or globally. Specifically, we distinguish between the following two approaches to computing marginal earliness costs for an order in the FCMAP problem:

1) *Local Earliness Weight*: At time t, the local marginal earliness cost associated with an order O (see Figure 3) can be approximated as the

difference in procurement costs associated with the latest non-dominated bid combinations compatible with processing the order at respectively time t (namely Bid_Comb^i) and time $t + k \cdot \overline{du}$ (namely Bid_Comb^j):

$$earl^L = \frac{pc_i - pc_j}{k\overline{du}},$$

2) *Global Earliness Weight:* An alternative involves computing a single global marginal earliness cost for each order. This can be done using a Least Square Regression:

$$earl^G = \frac{\sum pc \cdot rd - n \cdot \overline{pc} \cdot \overline{rd}}{\sum rd^2 - n \cdot \overline{rd}^2},$$

where $\overline{pc}$ is the average procurement cost of non-dominated bid combinations for the order, and rd is their average release date.

The simplest possible release policy for the FCMAP problem involves releasing each order at its earliest possible release date, namely $r_i^{earliest}$ (see Pruning Rule 2 – it should be clear that this release date is never pruned by Rule 3). We refer to this policy as an *Immediate Release Policy.* It might sometime result in releasing some orders too early and hence yield unnecessarily high procurement costs. An alternative is to use an *Intrinsic Release Policy* that releases orders when their early/tardy priority $P_i(S_i)$ becomes positive. $P_i(S_i)$ can be viewed as the marginal cost incurred for delaying the start of order O_i at time t. As long as this cost is negative, there is no benefit to releasing the order. The tipping point, where $P_i(S_i) = 0$, is the order's intrinsic release date:

$$\hat{r}_i = dd_i - du_i - \frac{tard_i}{tard_i + earl_i} k\overline{du}.$$

Here again, one can use either the local or global earliness cost associated with an order. Intuitively, one would expect the global earliness cost to be more appropriate for the computation of an order's release date and its local earliness cost to be better suited for the computation of its priority at a particular point in time. This has generally been confirmed in our experiments. In Section 7, we only present results where priorities are computed using local earliness costs. We do however report results, where release dates are computed with both local and global earliness costs, as we have not found any significant differences between these two policies.

Rather than limiting ourselves to deterministic adaptations of Ow and Morton's dispatch rule, we have also experimented with randomized versions, where order release dates and priorities are modified by small stochastic disturbances. This enables our procedure to make up for the way in which it approximates procurement costs and sample the search space in the vicinity of its deterministic solution. The resulting early/tardy search heuristic operates by looping through the following procedure for a pre-specified amount of time. As it iterates, the procedure alternates between the immediate release policy and the intrinsic release policy and successively tries a number of different values for the heuristic's look-ahead parameter k. The following outlines one iteration – namely for one particular release policy and one particular value of the look-ahead parameter.

1. For each order O_k, $k \in M=\{1,2,\ldots,m\}$, compute the order's release date. When using the immediate release policy, this simply amounts to setting the order's release date $RD_k = r_k^{earliest}$. When using the intrinsic release policy, the order's release date is computed as $RD_k = Max\{r_k^{earliest}, (1+\alpha)\times\hat{r}_k\}$, where α is randomly drawn from the uniform distribution $[-dev_1, +dev_1]$ (dev_1 is a parameter that controls how widely the procedure samples the search space);
2. Dispatch the orders, namely let $t_0 = \underset{k \in M}{Min}\, RD_k$
   1) For all those orders O_k that have not yet been scheduled and whose release dates are before t_0, compute the order's priority at time t_0 as:
   $$PR_k(t_0) = (1+\beta) \cdot P_k(dd_k - du_k - t_0),$$
   where P_k is the early/tardy priority defined in (3) and where β is randomly drawn from the uniform distribution $[-dev_2, +dev_2]$ (dev_2 is a parameter that controls how widely the procedure samples the search space);
   2) Let order O_i be the order with the highest priority. Schedule O_i to start at time t_0;
   3) If all orders have been scheduled, then Stop. Else, let $t_1 = t_0 + du_i$ and t_2 be the earliest release date among those orders that have not yet been scheduled. Set $t_0 = Max\{t_1, t_2\}$ and repeat Steps 1-3.
   4) Compute the profit of the resulting solution. If it is higher than the best solution obtained so far, make this the new best solution.

A deterministic version of this procedure simply amounts to setting dev_1 and dev_2 to zero.

7. Computational evaluation

A number of experiments have been run to evaluate the impact of our pruning rules and the performance of our heuristic search procedure. Below we summarize results of two sets of experiments aimed at evaluating the impact of ignoring the manufacturer's capacity constraints and at gauging the overall quality of the solutions produced by our search heuristic.

7.1. Empirical setup

Problems were randomly generated to cover a broad range of conditions by varying the distribution of bid prices and bid delivery dates as well as the overall load faced by the manufacturer. Results are reported for 2 groups of problems:

1. Problems with 10 orders, 5 required components per order and 20 supplier bids per component. These problems were kept small enough so that they could be solved with our branch-and-bound algorithm. Key parameter values were drawn from the following uniform distributions:
   - Order processing time: U[5,25]
   - Order marginal tardiness cost: U[1,10]
   - Order due dates: 2 distributions:
     a. Medium Load (***ml***) problems: U[100,300]
     b. Heavy Load (***hl***) problems: U[100,200]
   - Component bid deliveries: 2 distributions:
     a. Narrow distribution (***nd***): U[0,50]
     b. Wide distribution (***wd***): U[0,100]
   - Component bid prices: 2 distributions:
     a. Narrow bid price distribution (***np***): U[5,35]
     b. Wide bid price distribution (***wp***): U[5,65]

A total of 20 problems were generated in each category (ml/hl, nd/wd, np/wp), yielding a total of 160 problems.

2. Problems with 50 orders, 5 required components per order and 20 supplier bids per component. On these larger problems, we were only able to compare our search heuristic with a procedure that reflects traditional procurement practices by ignoring capacity constraints – referred below as an "infinite capacity" procedure. Key parameter values were drawn from the following uniform distributions:
   - Order processing time: U[10,50]
   - Order marginal tardiness cost: U[1,10]
   - Order due dates: U[500,1500]
   - Component bid deliveries: same 2 distributions as 10-order problems (***nd/wd***)
   - Component bid prices: same 2 distributions as 10-order problems (***np/wp***)

A total of 20 problems were generated in each category for a total of 80 problems.

Note that the above distributions were selected to be representative of scenarios where the manufacturer's capacity is not sufficient to deliver all orders in time and where therefore finding the right tradeoffs between minimizing tardiness costs and procurement costs is most critical. Note also that order revenues are irrelevant, since the orders to be produced are fixed. In other words, all solutions admit the same overall revenue and overall profit is solely determined by the sum of tardiness and procurement costs associated with a given solution. Accordingly, we report overall costs rather than overall profits.

7.2. Impact of ignoring the manufacturer's capacity

Results presented in Figure 4 compare the performance of our Early/Tardy search heuristic (***ET***) with that of an "infinite capacity" procurement policy. This policy first selects the cheapest bids compatible with each order's due date, then schedules these orders using Vepsalainen and Morton's Apparent Tardiness Cost dispatch rule [11], a well-regarded heuristic to minimize tardiness costs – note that there are no earliness costs here, since procurement decisions have already been taken care of in the first step. This policy is representative of reverse auctions, where the manufacturer ignores its internal capacity constraints when selecting among competing bids. The results in Figure 4 represent 95% confidence intervals for the total cost per order – procurement cost plus tardiness cost, computed in all four categories of 50-order problems. The version of our ET search heuristic used in these experiments relied on global earliness weights for release date computations and on local earliness weights for dispatching computations. Deviation parameters were set as dev_1=dev_2=0.3.

It can be seen that accounting for capacity constraints in selecting procurement bids significantly improves the manufacturer's bottom line across all four problem sets. The most significant improvements are observed on the more difficult problem set, namely np/wd and wp/wd, where bid delivery dates can vary widely. A breakdown of average tardiness and procurement costs is provided in Table 1 for the most difficult problem set, wp/wd. It can be seen that our ET search heuristic yields solutions that have one third of the cost of an infinite capacity procurement policy. More specifically, in contrast to the infinite capacity policy, our ET heuristic is capable of sacrificing procurement costs to decrease overall costs through reductions in tardiness penalties. In this particular case, a relatively minor increase in procurement costs of a couple of percent results in a major reduction in tardiness costs (of nearly 80%). These results strongly validate the FCMAP model advocated in this paper.

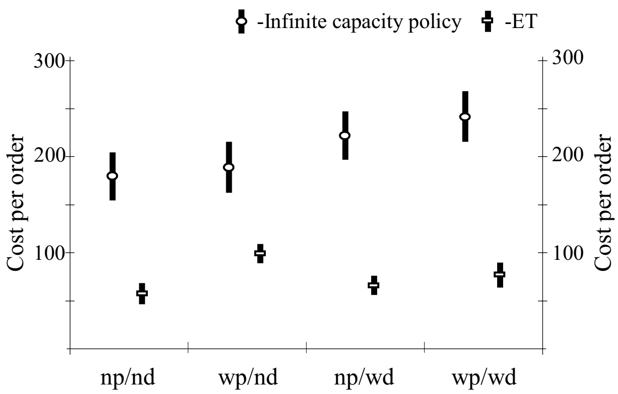

Figure 4. Impact of ignoring the manufacturer's capacity

Table 1. Impact of ignoring capacity - standard deviations are between parentheses

wp/wd	Proc. Cost	Tard. Cost	Total Cost per Order
Infinite Capacity Policy	**37.35** (0.91)	**206.79** (58.71)	**244.14** (59.08)
ET (Global – Local)	**39.11** (1.14)	**41.53** (25.80)	**80.64** (25.79)

7.3. Distance from the optimum

Experiments have been conducted to measure distance from the optimum, using the branch-and-bound procedure introduced in Section 5. Just like our ET heuristic, branch-and-bound was used following the application of the pruning rules introduced in Section 4. Clearly, even with the pruning rules, this procedure remains very slow and is only practical on relatively small problems. Tables 1 and 2 report distance from the optimum (obtained with branch-and-bound) for several variations of our early/tardy search heuristics (ET) as well as the infinite capacity procurement policy introduced earlier in this Section. This distance was computed as: [*cost*(*solution*)–*cost*(*optimal_solution*)] /*cost*(*optimal_solution*). The variations of our ET heuristics included both a deterministic version where dev_1=dev_2=0 and stochastic version where dev_1= dev_2=0.3. While dispatching decisions were made using local earliness costs, we report results obtained with release policies using both local and global earliness costs. Standard deviations are between parentheses. As can be seen, our ET heuristic is generally within 10 percent of the optimum. Even the deterministic version yields solutions that are within 15% of the optimum across all 8 problems sets, suggesting that the insight given by Property 1 and the adaptation of Ow and Morton's dispatch rule are rather effective.

Table 2. Distance to optimum – medium load

	Infinite Capacity	ET Search Heuristic			
		Global RD / Local Priority		Local RD / Local Priority	
		Determ	Stoch	Determ	Stoch
np/nd	49.07 (32.79)	6.36 (2.97)	6.31 (2.96)	6.08 (3.23)	5.62 (3.31)
wp/nd	32.51 (28.29)	13.63 (6.35)	12.04 (5.01)	13.63 (6.35)	10.73 (5.05)
np/wd	112.29 (100.06)	9.31 (2.67)	9.20 (2.68)	9.33 (2.69)	6.97 (2.95)
wp/wd	60.66 (64.89)	15.08 (6.10)	13.18 (5.85)	14.68 (6.74)	12.95 (4.75)

Table 3. Distance to optimum – heavy load

	Infinite Capacity	ET Search Heuristic			
		Global RD / Local Priority		Local RD / Local Priority	
		Determ	Stoch	Determ	Stoch
np/nd	51.96 (44.51)	6.71 (2.91)	6.47 (2.99)	7.13 (3.19)	6.58 (2.94)
wp/nd	46.08 (36.01)	13.07 (5.40)	11.70 (4.54)	13.07 (5.40)	11.60 (4.54)
np/wd	176.67 (117.53)	9.64 (3.39)	8.75 (3.89)	9.97 (3.39)	8.81 (3.81)
wp/wd	134.35 (98.64)	13.96 (7.01)	13.14 (6.64)	13.96 (7.01)	13.30 (6.85)

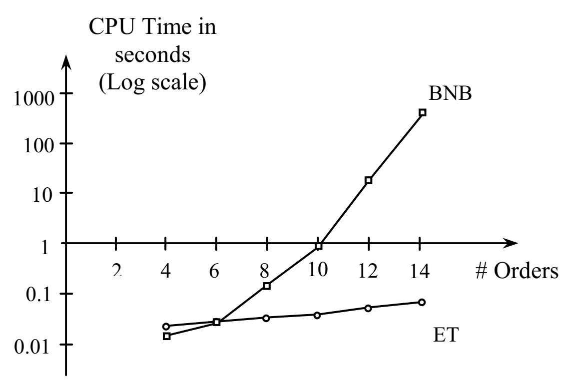

Figure 5. CPU time

Finally, it should be noted that while the branch-and-bound algorithm is computationally prohibitive, our ET search heuristic scales very nicely as depicted in Figure 5, where we report CPU time as the number of orders increases.

8. Concluding remarks

Prior work on dynamic supply chain formation has generally ignored capacity and delivery date considerations. In this paper, we have introduced a model for finite capacity multi-attribute procurement problems faced by manufacturers who have to select among supplier bids that differ in terms of prices and delivery dates. We have identified several dominance criteria that enable the manufacturer to quickly eliminate uncompetitive combinations of bids. A branch-and-bound algorithm and a heuristic search procedure have been introduced to help the manufacturer select a combination of bids that maximizes its overall profit, taking into account its finite capacity as well as the prices and delivery dates associated with different supplier bids. We have shown that this heuristic greatly improves over simpler infinite capacity bid selection models. Comparison with optimum solutions obtained using branch-and-bound, suggest that our search procedure generally yields solutions that are within 10% of the optimum.

9. Acknowledgements

The research reported in this paper has been conduced as part of the MASCHINE project, a joint research effort between Carnegie Mellon University and the University of Michigan funded by the National Science Foundation under ITR Grant 0205435.

10. References

[1] J. Gallien and L.M. Wein. A Smart Market for Industrial Procurement with Capacity Constraints. Submitted to *Management Science*, 2002.

[2] M. Babaioff and N. Nisan. Concurrent Auctions Across the Supply Chain. *Proceedings of ACM Conference on Electronic Commerce* (*ACM-EC'*01), pages 1-10, Tampa, October 2001.

[3] W.E. Walsh and M.P. Wellman. A Market Protocol for Decentralized Task Allocation. *Third International Conference on Multi-Agent Systems*, pages 325-332, 1998.

[4] N.M. Sadeh, D.W. Hildum, D. Kjenstad and A. Tseng. MASCOT: An Agent-Based Architecture for Dynamic Supply Chain Creation and Coordination in the Internet Economy. *Journal of Production Planning and Control*, 12(3):21-223, 2001.

[5] J. Collins, C. Bilot, M.L. Gini, and B. Mobasher. Decision Processes in Agent-Based Automated Contracting. *IEEE Internet Computing*, pages 61-72, March 2001.

[6] M. Pinedo. *Scheduling: Theory, Algorithms, and Systems*, Prentice Hall, NJ, 1995.

[7] P. Faratin and M. Klein. Automated Contract Negotiation as a System of Constraints. *In Proceedings of the Workshop on Distributed Constraint Reasoning, IJCAI*-01, pages 33-45, Seattle, August 2001.

[8] J. Du and J.Y.T. Leung. Minimizing Total Tardiness on One Processor is NP-Hard. *Mathematics of Operations Research*, 15:483-495, 1990.

[9] J. Sun and N.M. Sadeh. Coordinating Multi-Attribute Reverse Auctions Subject to Finite Capacity Considerations. Carnegie Mellon University, eSupply Chain Management Lab. Technical Report CMU-ISRI-03-105, August 2003.

[10] P.S. Ow and T.E. Morton, The Single Machine Early/Tardy Problem, *Management Science*, 35(2), 177-191, 1989.

[11] A. Vepsalainen and T.E. Morton. Priority Rules and Lead Time Estimation for Job Shop Scheduling with Weighted Tardiness Costs. *Management Science*, 33:1036-1047, 1987.