Coordinated Selection of Procurement Bids in Finite Capacity Environments

Jiong Sun and Norman M. Sadeh

November 2006
CMU-ISRI-06-118

Tepper School of Business
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213-3890

Email Addresses: jiongs@andrew.cmu.edu, sadeh@cs.cmu.edu

The research reported in this paper has been funded by the National Science Foundation under ITR Grant 0205435.

Keywords: Supply Chain Management, Procurement, Bid Selection/Winner Determination, Finite Capacity Scheduling

Abstract

Research on the evaluation and selection of procurement bids ("winner determination") has traditionally ignored the temporal and finite capacity constraints under which manufacturers and service providers often operate. We consider the problem faced by a firm that procures multiple key components or services from a number of possible suppliers. Bids submitted by suppliers include a price and a delivery date. The firm has to select a combination of supplier bids that will maximize its overall profit. Profit is determined by the revenue generated by the products (or services) sold by the firm, the costs of the components (or services) it acquires as well as late delivery penalties it incurs if it fails to deliver its products/services 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 proceed to show that our model can be characterized as a pseudo-early/tardy scheduling problem and use this observation to build an efficient heuristic search procedure. Computational results show that our heuristic procedure typically yields solutions that are within a few percent from the optimum. They further indicate that taking into account the manufacturer/service provider's capacity can significantly improve its bottom line.

1. INTRODUCTION

Today's global economy is characterized by fast changing market demands, short product lifecycles and increasing pressure 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 dynamically select among multiple possible supply chain partners in response to changing market conditions. In this paper, we consider an environment where a firm needs to meet customer delivery commitments while procuring a combination of key components or services from multiple possible suppliers. At any point in time, components or services offered by different suppliers may vary both in terms of prices and delivery dates. Such a situation arises in a number of different contexts. This includes manufacturers with long-term relationships with more than one supplier (possibly independently managed plants owned by the same firm) as well as manufacturers or service providers dynamically selecting prospective suppliers in response to changing market demands. These latter scenarios arise in the context of capacity subcontracting in manufacturing and logistics [2] as well as in a wide range of other sectors (e.g. call center capacity, dynamic procurement of programming services [38], translation services [29], and a growing number of other services [30]. These dynamic practices are increasingly facilitated by the emergence of e-business standards, such as ebXML [16], SOAP [49], UDDI [34] and WSDL [50].

Prior research on bid selection ("winner determination") has generally ignored temporal and capacity constraints under which companies operate (e.g. due dates by which different orders need to be delivered to customers as well as the limited capacity available to assemble components/services obtained from suppliers). The work presented herein shows that taking such constraints into account can help companies make more judicious decisions when it comes to selecting among multiple supply alternatives.

Specifically, we present techniques aimed at exploiting temporal and capacity constraints to help a firm select among supply alternatives that differ in prices and delivery dates. We refer to this problem as the *Finite Capacity Multi-Component*

Procurement (FCMCP) problem. This article provides a formal definition of the FCMCP problem, discusses its complexity and introduces several rules that can be used to prune its search space. It also presents a branch-and-bound algorithm, a simulated annealing procedure and an efficient pseudo-early/tardy heuristic search procedure that all take advantage of these pruning rules. Computational results show that accounting for the firm's finite capacity can significantly improve its bottom line, confirming the important role played by finite capacity considerations in procurement problems. Results are also presented that compare the performance of our heuristic search procedures both in terms of solution quality and computational requirements under different supply profile (or "bid profile") assumptions. These results suggest that our pseudo-early/tardy procedure is generally capable of generating solutions that are within just a few percent from the optimum and that it scales nicely as problem size increases.

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 FCMCP problem. Section 4 identifies three rules that can help a firm (manufacturer or service provider) eliminate non-competitive procurement bids or bid combinations. Section 5 introduces a branch-and-bound algorithm that takes advantage of our pruning rules. This is followed by the presentation of two heuristic search procedures that also take advantage of these pruning rules. In particular, Section 6 introduces a heuristic search procedure that exploits a property of pruned FCMCP problems introduced in section 4 to solve the resulting problem as a pseudo-early/tardy scheduling problem. In Section 7, a second heuristic search procedure is presented that combines Simulated Annealing (SA) search with a cost estimator based on the well-known "Apparent Tardy Cost" rule first introduced by Vepsalainen and Morton [48]. Section 8 presents a post-processing procedure that can further improve the quality of a solution. An extensive set of computational results are presented and discussed in Section 9. Section 10 discusses extensions of our techniques where we relax the lot-for-lot assumption made earlier and where we also account for inventory costs. Section 11 provides some concluding remarks and discusses future extensions of this research.

2. LITERATURE OVERVIEW

2.1 Sourcing and Procurement Strategies

A number of different studies have examined tradeoffs associated with different sourcing and procurement strategies, going back to work comparing Japanese and US sourcing and procurement models in the automotive industry in the late eighties [51]. More recent work includes that of Pyke and Johnson [39] who provide qualitative guidelines for selecting between five types of supplier relationships, from full ownership of the supplier to short-term, market-based competition among multiple suppliers. They argue that different types of sourcing strategies are better suited for different situations and that companies should generally consider a mix of short-term and long-term relationships. They further argue that critical, high-value added components or components with complex interfaces are often better handled through strategic partnerships, whereas commoditized components available from multiple sources can more effectively be handled through dynamic e-procurement. The authors also suggest that "firms that decide to pursue strategic alliances should strongly consider introducing competition into the relationship, while firms that buy over the Internet should consider building longer-term relationships". Peleg et al. [36] compare three procurement strategies: the above two plus a mixed strategy combining both short-term and long-term elements. They show that the superiority of one strategy over the others depends on contract terms. Bensaou [5] reports on a study that debunks the myth that Japanese car manufacturers rely solely on long-term strategic partnerships with suppliers and advocates the management of portfolio of buyer-supplier relationships covering a wide spectrum of possible arrangements. de Boer et al. [14] consider the decision faced by a purchaser that has to decide how many supplier tenders to invite for a given purchase. A review of models for constructing short-term and long-term contracts in business-to-business markets has been conducted by Kleindorfer and Wu [27]. Elmaghraby [17] also provides an excellent review of research done in the fields of economics and operations research on tradeoffs between different sourcing strategies. Collectively, this body of research indicates that many environments warrant considering dynamic sourcing and procurement strategies, where one can dynamically select between offers from

multiple possible suppliers. As already indicated in Section 1, these scenarios are not limited to the manufacturing sector. They also extend to the service industry.

2.2 Bid Selection/Winner Determination

Reverse auctions are commonly used for procurement in large enterprises. Simple formats such as first-price sealed bid auctions and English auctions have become popular in maintenance, repair, and operations ("MRO") procurement. More complex formats involving combinatorial auctions are also being introduced in strategic sourcing contexts [25, 43]. Reverse multi-unit auctions with volume discounts have also been studied by Davenport et al. [12], using a set covering formulation and an iterative descending price auction that yields competitive equilibria.

There is also a growing realization that for reverse auctions to be practical in settings that go beyond simple MRO procurement environments, they have to be able to accommodate non-price attributes such as quality or leadtimes. Beil and Wein [4] consider the problem faced by a manufacturer who uses a reverse auction to award a contract to a single supplier based on bids that include a price and a set of non-price attributes. Using a multi-round, open-ascending auction mechanism, they suggest an inverse-optimization approach that, subject to some assumptions, allows the buyer to learn the suppliers' cost functions and then determine a scoring rule that maximizes its own utility. Instead of maximizing the buyer's utility, Milgrom [31] shows that true costs can be revealed and efficiency is achievable if the auctioneer announces his true utility function as the scoring rule in a Vickrey auction. Che [7] shows that to maximize utility, the optimal scoring rule may not be identical to the buyer's true value function.

The past few years have also seen some initial work on capacity-constrained allocation mechanisms. This research so far has primarily focused on mechanisms to accommodate supplier capacity constraints. In particular, Gallien and Wein [19] design and analyze a multi-item, multi-round, pay-as-you-bid procurement auction mechanism, considering the capacity constraints of suppliers.

Other relevant work includes that of Chen et al. [8] who consider a multi-item, Vickrey procurement auction that incorporates transportation costs. To determine the

optimal selection of winning bids, optimization algorithms are proposed for the resulting allocation problems. This line of work goes back to research by Stanley et al. [46], who employed linear programming techniques for a static, single-attribute, single-item procurement auction, and includes more recent extensions looking at multi-round, multi-attribute, and multi-unit variations of this problem (e.g., [6], [20], [33], [40]).

The work we present is not restricted to procurement auctions but rather extends to any situation where a manufacturer/service provider has to select among multiple procurement options that differ in price and delivery date. This includes but is not limited to one-shot, sealed-bid procurement auction scenarios. The manufacturer/service provider explicitly computes the profits generated by different (non-dominated) bid combinations and attempts to identify one that maximizes its overall profit. This is done taking into account the synchronization requirements associated with the procurement of multiple components (or services) required by a given end product (or service) as well as the costs of different procurement bids, the limited capacity of the manufacturer (e.g., to assemble finished products) and potential late delivery penalties.

2.3 Coordinating Procurement and Planning

Another related area of research deals with coordinating procurement and production planning. Most work in this area has assumed stochastic models in which capacity is either ignored or modeled at a relatively coarse level. This includes the work of Bassok and Akella [3] who explore a single-period, single-machine model that integrates production and raw material ordering decisions in a manufacturing facility with a single type of raw material and one or more finished products with stochastic demand. Raw material delivery is assumed to be stochastic: the manufacturer typically receives just a fraction of what it orders. The authors focus on determining the quantities in which products are released into the system, taking into account the system's capacity and expectations about the fraction of ordered raw materials likely to arrive. The objective is to minimize the sum of backlog costs, production costs, ordering costs, as well as raw material and finished goods holding costs. Gurnani et

al. [22] study a similar problem, where a manufacturer faces stochastic demand for a single finished product that requires two critical components. Other relevant work in this area include that of Song et al. [45], Gurnani et al. [21], Gallien and Wein [19], Yano [52], Kumar [28], Hopp and Spearman [23], Shore [44], and Chu et al. [9] to name just a few. Ciarallo et al. [10] studied a multi-period aggregated production planning problem with a single-product single-stage manufacturer, and Jain and Silver [24] considered a similar model where the manufacturer can pay a premium to purchase dedicated capacity from a supplier. Karmarkar and Lin [26] study a multi-period production planning problem in which demand and yield are random. These are just a few of the many articles published in this broad area.

In contrast to the above, the FCMCP model introduced in this paper is deterministic. It explicitly captures the finite capacity of the manufacturer/assembler (or service provider) and allows for environments with multiple finished products, each requiring a possibly different set of components. Demand is modeled with each order having its own delivery date and its own marginal penalty for not meeting that date. By differentiating between different orders, their individual due dates, tardiness costs and component requirements, it becomes possible to develop solutions that capture the finer tradeoffs entailed by these requirements and the finite capacity of the manufacturer/service provider.

3. THE FINITE CAPACITY MUTI-COMPONENT PROCUREMENT PROBLEM

The Finite Capacity Muti-component Procurement (FCMCP) problem revolves a manufacturer or service provider (later referred to as the "manufacturer") that has to satisfy a set of customer commitments or orders O_i, $i \in M = \{1,...,m\}$ (see Figure 1). Each order i needs to be completed by a due date dd_i, and requires one or more components or services (later referred to as "components" or "supplies"), 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 product or waiting for different tasks to be completed before being able to deliver a service). 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").

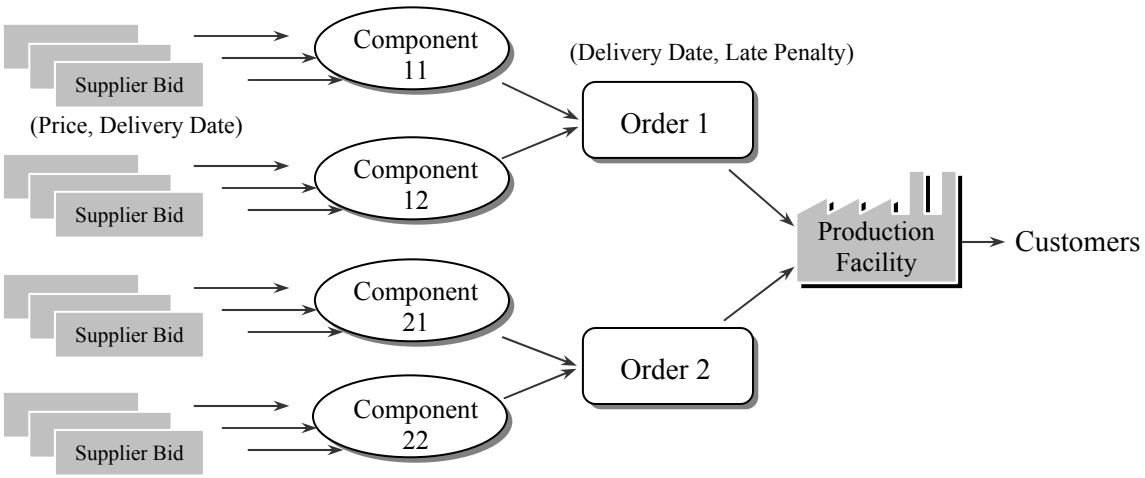


Figure 1. Finite capacity muti-component procurement problem

Formally, for each order O_i and each component $comp_{ij}$, $j \in N_i = \{1,...,n_i\}$, the manufacturer can select from a set of multi-attribute bids $\beta_{ij} = \{B_{ij}^1,...,B_{ij}^{n_{ij}}\}$ from prospective suppliers, where n_{ij} is the total number of procurement bids for component $comp_{ij}$. Each bid B_{ij}^k includes a bid price bp_{ij}^k and a proposed delivery date dl_{ij}^k. Below we use the notation $B_{ij}^k = (dl_{ij}^k, bp_{ij}^k)$.

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 [37].

A solution to the FCMCP problem consists of:

- a selection of bids: $Bid\_Comb=\{Bid\_Comb_1,\ldots,Bid\_Comb_m\}$, 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:

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

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*, and
- $T_i = Max(0, st_i + du_i - dd_i)$ 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: $cost(Bid\_Comb,ST) = \sum_{i \in M} \sum_{j \in N_i} bp_{ij} + \sum_{i \in M} tard_i \times T_i$.

It is worth noting that the above model contrasts with earlier research in dynamic supply chain formation, which has generally assumed manufacturers with infinite capacity or fixed lead times and ignored delivery dates and tardiness penalties [11, 13, 18]).

From a complexity standpoint, it can easily be seen that the FCMCP 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 [15].

An example of an exact procedure to solve FCMCP 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. Below, we identify a number of rules that can be used to efficiently prune the search space associated with FCMCP problems.

4. PRUNING THE SEARCH SPACE

Pruning Rule 1: Eliminating Expensive Bids with Late Delivery Dates

Consider an FCMCP problem P with an order O_i requiring a component $comp_{ij}$ for which the manufacturer has received a set of bids $\beta_{ij} = \{B_{ij}^1, ..., B_{ij}^{n_{ij}}\}$ from possible suppliers. Let $B_{ij}^k = (dl_{ij}^k, bp_{ij}^k)$ and $B_{ij}^l = (dl_{ij}^l, bp_{ij}^l)$ be two bids in β_{ij} such that:

$$dl_{ij}^l \geq dl_{ij}^k \text{ and } bp_{ij}^l \geq bp_{ij}^k .$$

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

The correctness of this rule should be obvious. Its application is illustrated in Figure 2, 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 late. Similarly, bid_{22} and bid_{24} can also be pruned.

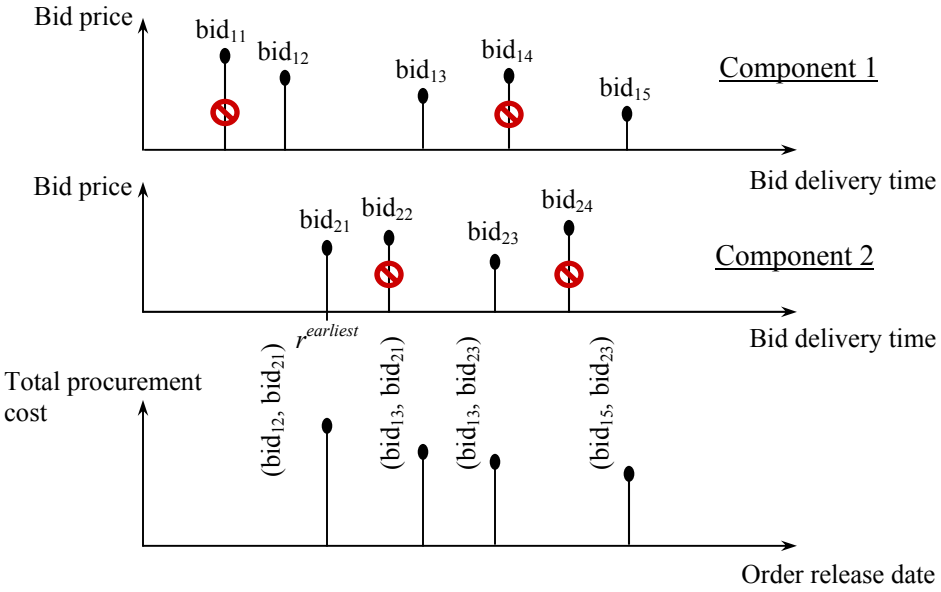


Figure 2. From 20 bid combinations to 4 non-dominated ones

Pruning Rule 2: Eliminating Expensive Bids with Unnecessarily Early Delivery Dates

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

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

Let B_{ij}^k and B_{ij}^l be two bids for component $comp_{ij}$ such that:

$$bp_{ij}^l \ge bp_{ij}^k \text{ and } dl_{ij}^l \le dl_{ij}^k \le r_i^{earliest}.$$

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

An intuitive explanation should suffice to convince the reader. While bid B_{ij}^{l} has an earlier delivery date than bid B_{ij}^{k}, this earlier date is not worth paying more for: it does not add any scheduling flexibility to the manufacturer since the start of order O_i remains constrained by $r_i^{earliest} \geq dl_{ij}^{l}$. A formal proof can easily be built based on this observation.

Note that, in general, it is not possible to prune bid B_{ij}^{k}. This is because other bids for component $comp_{ij}$ may have delivery dates that are after $r_i^{earliest}$, which would reduce the number of available scheduling options possibly leading to lower quality solutions. Application of this rule is also illustrated in Figure 2, 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 FCMCP problem P whose search space has already been pruned using Rule 1. In other words, given two bids $B_{ij}^{k} = (dl_{ij}^{k}, bp_{ij}^{k})$ and $B_{ij}^{l} = (dl_{ij}^{l}, bp_{ij}^{l})$, $k \neq l$, for the same component $comp_{ij}$, if $dl_{ij}^{l} > dl_{ij}^{k}$, then $bp_{ij}^{l} < bp_{ij}^{k}$.

Let $Bid\_Comb_i^a = \{B_{i1}^a, ..., B_{in_i}^a\}$ be a combination of bids for the n_i components required by order O_i. Suppose also that there exist two bids $B_{ik}^a = (dl_{ik}^a, bp_{ik}^a) \in Bid\_Comb_i^a$ and $B_{ik}^b = (dl_{ik}^b, bp_{ik}^b) \in Bid\_Comb_i^b$, and a component $comp_{il}$, $l \neq k$ such that $dl_{ik}^a < dl_{ik}^b \leq dl_{il}^a$, then $Bid\_Comb_i^a$ is dominated by $Bid\_Comb_i^b$, where $Bid\_Comb_i^b = (Bid\_Comb_i^a \setminus \{B_{ik}^a\}) \cup \{B_{ik}^b\}$. By "dominated" we mean that, for every solution to problem P involving $Bid\_Comb_i^a$, 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 $comp_{il}$ that gets delivered at time $dl_{il}^a \geq dl_{ik}^b > dl_{ik}^a$, replacing bid B_{ik}^a with bid B_{ik}^b will not delay the

start of order O_i and can only help reduce the cost of its components since $bp_{ik}^a > bp_{ik}^b$ (as indicated earlier, we assume that Rule 1 has already been applied to prune bids). It is straightforward to build a formal proof based on the above observation. Note also that Rule 3 actually subsumes Rule 2 – Rule 2 is easier to visualize and also introduces the notion of earliest possible release date, which we use later in this article.

The three pruning rules we just identified can be used to prune the set of bids to be considered. This is illustrated in Figure 2, 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. This is done as follows:

1. For each component $comp_{ij}$, create a sorted list $\lambda_{ij} =< B_{ij}^1,..,B_{ij}^{n_{ij}} >$ such that $dl_{ij}^k < dl_{ij}^{k+1}$. Create an overall list of delivery dates for all the bids received for O_i (i.e., for all the components required by the order) and sort the delivery dates in increasing order. Let Λ_i be this sorted list
2. For each date r_i in Λ_i, keep only those non-dominated combinations of bids that are compatible with having r_i as order O_i's release date. Note that such bid combinations are of the form $\{B_{i1}^a,...,B_{in_i}^a\}$ where $dl^{ij} \leq r_i, \forall j$, and there is no other bid B_{ij}^b such that $dl_{ij}^a < dl_{ij}^b \leq r_i$. In other words, for each component $comp_{ij}$, B_{ij}^a is the latest bid compatible with release date r_i (and

hence also the cheapest such bid). Finding such bids requires very little time, given the sorted bid lists λ_{ij} created in step 1.

As a parenthesis, it is worth noting that the three pruning rules we just introduced apply to scenarios with more complex constraints, as they only take advantage of the release constraint that requires each order to have all its components before it can be processed. For instance, this includes problems where the manufacturer is modeled as a job shop, the capacity of some machines is greater than one and there are sequence dependent setup times. It can also be shown that the pruning rules can be extended to accommodate problems with inventory holding costs, as long as orders are not allowed to be shipped before their due dates – this assumption corresponds to having finished goods inventory and is representative of many supply chain situations. This latter extension will be revisited in Section 10.

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

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

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 $pc_{ia} > pc_{ib}$, $\forall a,b \in \{1,...,m_i\}, a \neq b$. In other words, the total procurement costs of non-dominated bid combinations strictly decrease as their release dates increase.*

Proof:

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 $r_{ia} = dl_{ij}^a$. 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_{ik}^{a} of any component k must be the latest delivery date among those bids for component k that satisfy $dl_{ik}^{a} \le dl_{ij}^{a}$.

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 $r_{ib} = dl_{il}^{b} > r_{ia} = dl_{ij}^{a}$. Just as for bid combination $Bid\_Comb_i^a$, the fact that Rule 3 has been applied implies that the delivery date dl_{ik}^{b} of any component k in $Bid\_Comb_i^b$ must be the latest delivery date among those bids for component k that satisfy $dl_{ik}^{b} \le dl_{il}^{b}$. Given that $r_{ib} = dl_{il}^{b} > r_{ia} = dl_{ij}^{a}$, it also follows that, for any component k, we have $dl_{ik}^{b} \ge dl_{ik}^{a}$ with a strict inequality for at least one component, namely component l. Given that Rule 1 has been applied, it also follows that, for any component k, $bp_{ik}^{b} \le bp_{ik}^{a}$ with a strict inequality for at least one component (component l). Hence, $pc_{ia} = \sum_{1 \le k \le n_i} bp_{ik}^{a} > pc_{ib} = \sum_{1 \le k \le n_i} bp_{ik}^{b}$. □

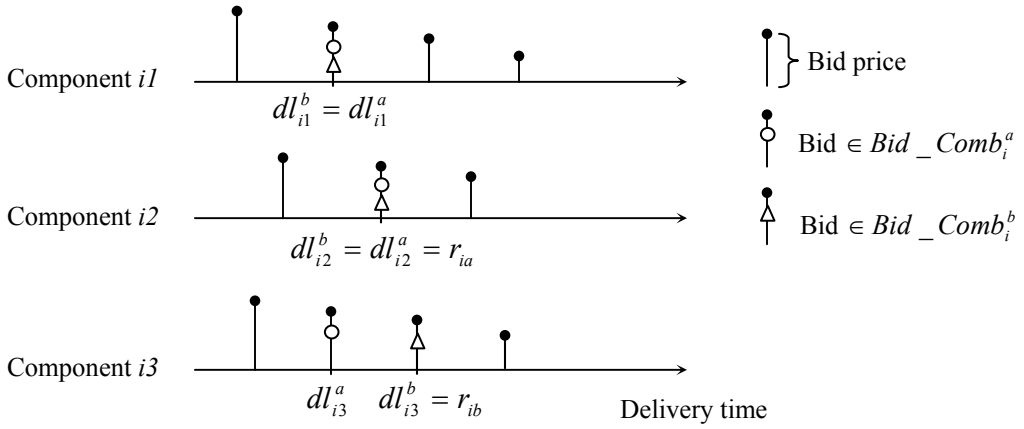


Figure 3. Illustration of Property 1

Property 1 is illustrated in Figure 3, 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_{ia} is determined by the delivery date of component 2, while r_{ib} is

determined by that of component 3. 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 lower price associated with the later delivery of component 3 in bid combination $Bid\_Comb_i^b$ (namely, $dl_{i3}^b = r_{ib} > dl_{i3}^b$).

Note also that, if after application of the pruning rules there exist two non-dominated bid combinations, $Bid\_Comb_i^a$ and $Bid\_Comb_i^b$, such that $r_{ia} = r_{ib}$, it must hold that $pc_{ia} = pc_{ib}$.

In the following sections, we introduce a branch-and-bound algorithm to solve the FCMCP problem along with two (significantly faster) heuristic search procedures. All three procedures take advantage of the pruning rules we just introduced. One of the two heuristic procedures also takes advantage of Property 1.

5. A BRANCH-AND-BOUND ALGORITHM

Following the application of the pruning rules introduced in the previous section, optimal solutions to the FCMCP 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} and r_{22} with $r_{21} < r_{22}$ - 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 a lower-bound for the profit of all complete solutions (i.e., leaf nodes) compatible with this node:

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

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

6. AN EARLY/TARDY 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 4 plots the total procurement cost and tardiness cost of an order for different possible start times. While tardiness costs increase linearly with start times that miss the order's due date, procurement costs vary according to a decreasing step-wise function. Specifically, the circles in Figure 4 represent the order's non-dominated bid combinations. For instance, if the order starts at time *st*, its procurement cost is pc_i, namely the procurement cost of the latest non-dominated bid

combination compatible with this start time (Bid_Comb^i). Its tardiness cost is equal to $tard \times \max(0, st + du - dd)$, where *tard* is its marginal tardiness penalty, *dd* its due date and *du* its duration (or processing time). The resulting problem can be viewed as a pseudo early/tardy scheduling problem. It bears a lot of similarity to a traditional early/tardy scheduling problem (e.g., [35]) but is also slightly different because procurement costs associated with different bid combinations lead to:

1) Step-wise earliness costs – in contrast to linear earliness costs found in a traditional early/tardy problem; and
2) Potential savings for completing the order past its due date, to the extent that there are late bid combinations that are so cheap that it is worth finishing the order late. Again this is different from a traditional early/tardy problem, where finishing an order on time always leads to the lowest cost for that order.

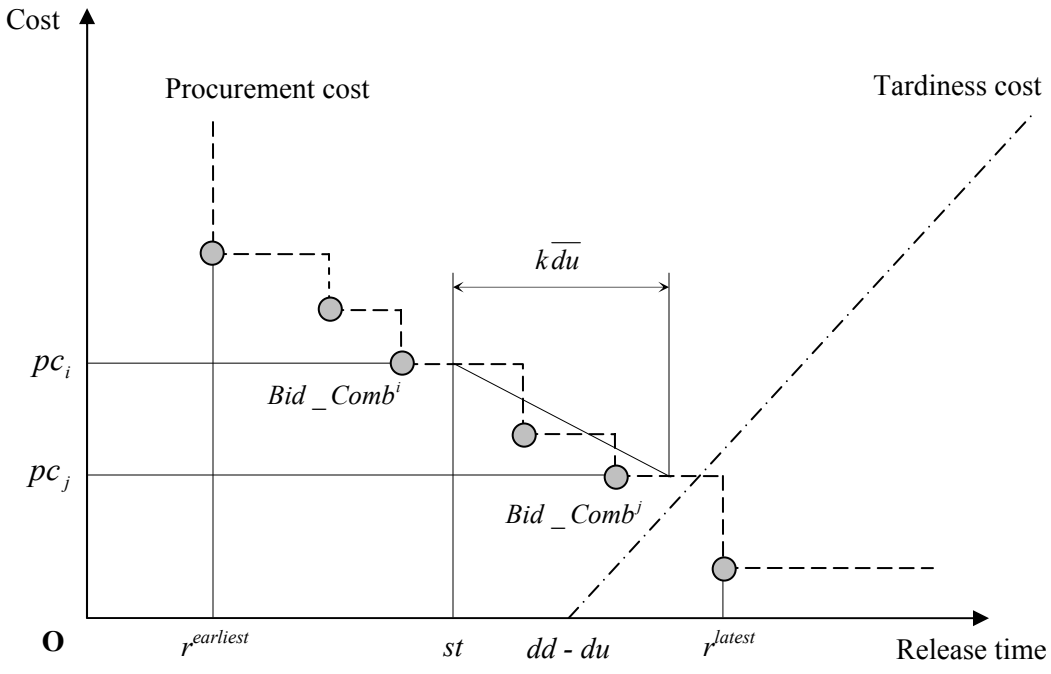


Figure 4. An order's tardiness and procurement costs

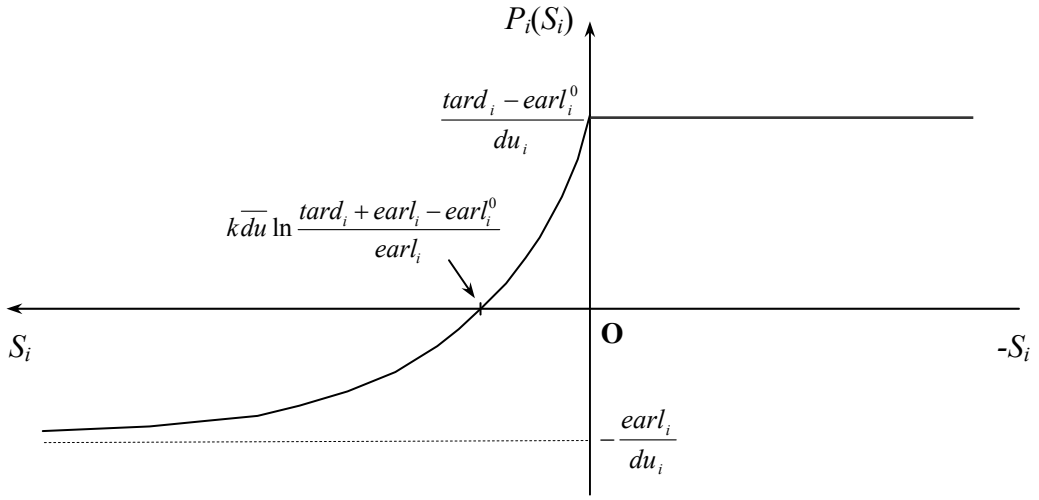


Figure 5. Priority function for Pseudo-Early/Tardy heuristic

Ow and Morton [35] have introduced an early/tardy dispatch rule for one-machine scheduling problems subject to linear earliness and tardiness costs. 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 FCMCP problem.

Ow and Morton's dispatch rule interpolates between two extreme cases. The first situation is one where all orders are assumed to have plenty of time and where only earliness costs need to be minimized. The second case 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, where each order receives a priority:

$$P_i = -earl_i / du_i \ ,$$

where P_i is the priority of order O_i, du_i is its processing time and $earl_i$ is 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 tardy, it can be shown that an optimal solution can be built by sequencing orders according to a *Weighted Shortest Processing Time* dispatch rule of the form:

$$P_i = tard_i / du_i ,$$

where $tard_i$ is its marginal tardiness penalty.

Like Ow and Morton's, our early/tardy heuristic interpolates between two extreme cases: one where all orders have plenty of time and one where all orders are late. The priority associated with this latter situation is different however from the one in Ow and Morton's rule. This is because later start times for orders that are late may still result in reductions in procurement costs. Accordingly, the priority associated with this latter situation is:

$$P_i = (tard_i - earl_i^0) / du_i ,$$

where $earl_i^0$ is the earliness weight at $st = dd_i - du_i$.

The resulting early/tardy heuristic assigns each order a priority that varies with its slack S_i:

$$P_i(S_i) = -\frac{earl_i}{du_i} + \frac{tard_i + earl_i - earl_i^0}{du_i} \times \exp[-\frac{(S_i)^+}{k \cdot \overline{du}}] \qquad (3)$$

where $\overline{du}$ is the average processing time of an order, k is a look-ahead parameter. $(X)^+$ denotes Max (X, 0), and slack S_i at time t is defined as:

$$S_i = dd_i - du_i - t .$$

The above formula can easily be seen to reduce to the *Weighted Shortest Processing Time* dispatch rule with a marginal tardiness cost of $tard_i - earl_i^0$ when slack $S_i \le 0$ and to the *Weighted Longest Processing Time* dispatch rule when $S_i \to \infty$. The look-ahead parameter k can intuitively be thought of as the average number of orders that would be tardy if order O_i is selected to be scheduled next. The value of k basically controls the transition between the two extreme scenarios between which this rule interpolates. Higher values of k make the transition start earlier. This can be interpreted as being more sensitive to tardiness when a larger number of orders stand to be late.

In the FCMCP problem, an order O_i cannot start before its earliest possible release date $r_i^{earliest}$ (see Pruning Rule 2 – it should be clear that this release date is never pruned by 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 FCMCP problem:

1) *Local Earliness Weight*: At time *t*, the local marginal earliness cost associated with an order *O* (see Figure 4) 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 FCMCP problem involves releasing each order O_i at its earliest possible release date, namely $r_i^{earliest}$. 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*, which 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:

$$\hat{r}_i = dd_i - du_i + k\overline{du} \cdot \ln \frac{earl_i}{tard_i + earl_i - earl_i^0}. \quad (4)$$

Here again, one can use either the local or global earliness weights associated with an order. Intuitively, one would expect the global earliness weight to be more appropriate for the computation of an order's release date and its local earliness weight 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 9, we only present results where priorities are computed using local earliness weights. We do however report results, where release dates are computed with both local and global earliness weights, and we found our heuristic performs better with global earliness weight.

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 perturbations. This enables our procedure to make up for the way in which it approximates procurement costs, sampling the search space in the vicinity of its deterministic solution. The resulting pseudo-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 and intrinsic release policies discussed earlier and successively tries a number of different values for the heuristic's look-ahead parameter *k*. The following outlines one iteration – i.e. with one particular release policy and one particular value of the look-ahead parameter.

1. For each order O_i, $i \in M = \{1,2,\ldots,m\}$, compute the order's release date. When using the immediate release policy, this simply amounts to setting the order's release date $RD_i = r_i^{earliest}$. When using the intrinsic release policy, the order's release date is computed as $RD_i = Max\{r_i^{earliest}, (1+\alpha) \times \hat{r}_i\}$, 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 $t_0 = \underset{i \in M}{Min}\, RD_i$

    1) For all those orders O_i that have not yet been scheduled and whose release dates are before t_0, compute the order's priority at time t_0 as:

$$PR_i(t_0) = (1+\beta)\cdot P_i(dd_i - du_i - t_0),$$

where P_i is the pseudo-early/tardy priority defined in (3) and 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. A SIMULATED ANNEALING SEARCH PROCEDURE

A second heuristic search procedure for the FCMCP problem involves using Simulated Annealing (SA) to explore different combinations of bids. Given a selection of non-dominated bid combinations $Bid\_Comb = \{Bid\_Comb_1, ..., Bid\_Comb_m\}$ – one combination per order, the procedure computes the release date r_i of each order O_i and sequences the orders, using the Apparent Tardiness Cost (ATC) dispatch rule first introduced in [48]. ATC is known to generally yield high quality schedules for the one-machine total weighted tardiness problem and has a $O(m\cdot\log m)$ complexity. As such it is an excellent estimator for the best solution compatible with a given selection of bid combinations. The following further details the SA procedure:

Step 1 – Initialization:

Set an initial temperature $Temp=Temp_0$, and an initial bid selection $Bid\_Comb^1 = \{Bid\_Comb_1^1,...,Bid\_Comb_m^1\}$;

Use the ATC dispatch rule to build a schedule. Let $cost^1=cost(Bid\_Comb^1, ST^1)$, where $ST^1 = \{st_1^1,...,st_m^1\}$ is the set of start times assigned by ATC to orders O_1 through O_m. Set $Bid\_Comb^{opt}=Bid\_Comb^1$ and $cost^{opt}=cost^1$.

Step 2 – Search:

Perform the following step N times:

Select $Bid\_Comb=neighbor(Bid\_Comb^1)$ (randomly or through some heuristic), and compute $cost=cost(Bid\_Comb, ST)$, where ST is the set of order start times assigned by the ATC dispatch rule;

If $cost^1 \geq cost \geq cost^{opt}$, set $Bid\_Comb^1=Bid\_Comb$;

Else if $cost>cost^1$ and $rand() \leq exp((cost^1-cost)/Temp)$, set $Bid\_Comb^1 = Bid\_Comb$;

Else if $cost<cost^{opt}$, set $Bid\_Comb^{opt}=Bid\_Comb^1=Bid\_Comb$.

If Bid_Comb^{opt} was not modified in the last N iterations, decrease the temperature $Temp = Temp \cdot \alpha$. Go to Step 3.

Step 3 – Termination Condition:

If Bid_Comb^{opt} has not been improved over the past K steps, then STOP and return (Bid_Comb^{opt}, ST^{opt}) as the best solution found by the procedure, otherwise go to Step 2.

The initial bid combination Bid_Comb^1 is randomly generated. Note also that the ATC dispatch rule is itself a parametric dispatch rule with a look-ahead parameter [41]. In our experiments, we systematically run ATC with values of the look-ahead parameter equal to 0.5, 1.0, 1.5, …, 6.0 and pick the best of the 12 solutions we have generated.

We have studied variations of this procedure that rely on different movesets. In particular, we have considered a one-bid moveset variation, where we modify the selection of a single bid (for a given component), and a two-bid moveset variation, where two bid selections (for two different components) are modified at once. For

both types of movesets, we have also experimented with two ways of selecting moves:

- a random mechanism, where a move in the moveset is randomly selected, and
- an organized mechanism that replaces the bid(s) that reduce most the profit of the current solution (Bid_Comb^1, ST^1), namely, those bids for which $bp^{ij} + tard_i \cdot T_i$ is the greatest.

The experiments presented in Section 9 use the organized mechanism, as we found it to generally yield higher quality solutions than the random one. We have not found a great difference between variations of our procedure using one-bid movesets and two bid movesets.

8. RIGHT-SHIFTING SOLUTION IMPROVEMENT PROCEDURE

Since the total cost function of each order is the sum of a linearly increasing tardiness cost function and a step-wise non-increasing earliness cost function, the total cost function has multiple local minimum points, as shown in Figure 6. Meanwhile, the above pseudo-early/tardy heuristic dispatches orders as soon as the machine becomes available. Hence, the solution produced by the pseudo-early/tardy heuristic can sometimes be further improved by right-shifting orders to lower local minimum points without changing the order processing sequence. Let <1,2,…,m> denote an order processing sequence produced by the above algorithm, st_i denote the start time of order O_i, $i \in \{1,2,\ldots,m\}$. As shown in Figure 6, the local minimum points have a one-to-one correspondence with the non-dominated bid combinations. The improvement method processes each order i from m to 1 as follows:

1. Right-shift early orders: If $st_i < dd_i - du_i$ and $st_i < st_{i+1} - du_i$, right-shift order O_i until $Min\{dd_i - du_i,\ st_{i+1} - du_i\}$, i.e., let $st_i = Min\{dd_i - du_i,\ st_{i+1} - du_i\}$. Even if its cost remains the same, right-shifting O_i creates more room to right-shift earlier orders, and therefore may help decrease the costs of the other orders. Go to Step 2.

2. Under any of the following situations, stop right-shifting:
   - $st_i = st_{i+1} - du_i$;
   - There is no local minimum point between st_i and $st_{i+1} - du_i$; or
   - All the local minimum points between st_i and $st_{i+1} - du_i$ have higher costs than the current cost, tc_i, of order O_i.

   Otherwise, let j^* be a local minimum point, the cost of which is the lowest among all the points between st_i and $st_{i+1} - du_i$. Right-shift order O_i to the local minimum point j^*, i.e., let $st_i = st_{j^*} - du_i$.

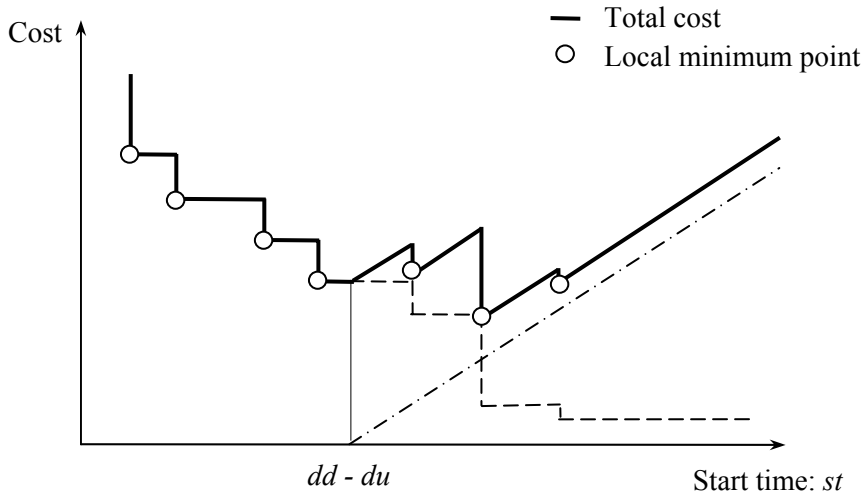


Figure 6. An order's total cost

9. COMPUTATIONAL EVALUATION

A number of experiments have been run to evaluate the impact of our pruning rules, the performance of our heuristic search procedures, and the benefits of our FCMCP model over one-dimensional bid selection models that ignore the manufacturer's finite capacity. These experiments are further detailed below.

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. The parameters used to generate these problems and the ratios between these parameters are consistent with those used in prior scheduling work (e.g. [32, 41]), as further detailed below. Specifically, 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 to optimality with our branch-and-bound algorithm. Two sets of problems were designed. Key parameter values of the first set 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:
        i. Medium Load (***ml***) problems: U[100,300]
        ii. Heavy Load (***hl***) problems: U[100,200]
    - Component bid deliveries: 2 distributions:
        i. Narrow bid delivery distribution (***nd***): U[0,50]
        ii. Wide bid delivery distribution (***wd***): U[0,100]
    - Component bid prices: 2 distributions:
        i. Narrow bid price distribution (***np***): U[5,35]
        ii. Wide bid price distribution (***wp***): U[5,65]

    The other set of problems has all the same distributions except for the component bid deliveries distributions:

        i. Narrow bid delivery distribution (***nd***): U[0,150]
        ii. Wide bid delivery distribution (***wd***): U[0,200]

    The first set of problems represents relatively easy problem instances, and in almost all cases, it holds that $dd - du > r^{latest}$, where r^{latest} is the latest release date determined by the cheapest non-dominant bid combination. Namely, the cost function is quasi-convex and has no local minimum point after $dd - du$.

We call this set of problems *Early-Bid* problems. The other set of problems, which we call *Mixed-Bid* problems, represents relatively hard problem instances, where, in many situations, $dd - du < r^{latest}$, i.e., the cost function has multiple local minimum points after *dd* – *du* (see Figure 6). A total of 20 problems were generated in each category (ml/hl, nd/wd, np/wp), yielding a total of 320 problems.

2. **Problems with 500 orders, 5 required components per order and 20 supplier bids per component**: While these problems were too large to be solved with branch-and-bound (even with our pruning rules), they were used to validate results obtained on the smaller sets of problems. This includes, determining how our heuristic search procedures scale up and evaluating the benefits of our FCMCP model over one-dimensional bid selection models and policies that ignore the manufacturer's finite capacity – the latter being simply referred to below as "infinite capacity" policies. Key parameter values were drawn from the following uniform distributions:
    - Order processing time: U[1,5]
    - Order marginal tardiness cost: U[1,10]
    - Order due dates:
        i. Medium Load (***ml***) problems: U[500,1500]
        ii. Heavy Load (***hl***) problems: U[500,1000]
    - Component bid deliveries: 2 distributions:
        i. Narrow bid delivery distribution (***nd***): U[0,800]
        ii. Wide bid delivery distribution (***wd***): U[0,1000]
    - 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 160 problems.

As indicated earlier, these parameter values were chosen to sample a broad range of conditions. They are also consistent with parameter values used in earlier scheduling studies. In particular, they approximately correspond to slack factor values ranging between 0.66 and 2.5, which is consistent with parameter values reported in [32].

Also, if one approximates the value of each order as the sum of the prices paid for its components, marginal tardiness penalties per day are on average roughly between 3% and 5% of order values with some orders having significantly higher ratios. Again, this is in line with values assumed in [32] and reflects the observation by Morton et al. that many due dates in a firm's order book are relatively soft, but a few will be critical with large losses of goodwill and business penalties if they are missed.

Note also that in measuring performance of our heuristics, 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 – see equation (1). Accordingly, we report overall costs rather than overall profits.

Distance from the Optimum

Tables 1 and 2 summarize results obtained on 10-order/*Early-Bid* problems. Tables 3 and 4 summarize 10-order/*Mixed-Bid* problems. These tables provide the average distance from the optimum of solutions obtained with different variations of our search heuristics across 16 problem sets (four medium load, *Early-Bid* problem sets in Table 1, four heavy load, *Early-Bid* problem sets in Table 2, four medium load, *Mixed-Bid* problem sets in Table 3, and four heavy load, *Mixed-Bid* problem sets in Table 4) with each problem set including a total of 20 problems. This distance from the optimum was computed as:

$$[cost(solution) - cost(optimal\_solution)] / cost(optimal\_solution).$$

Standard deviations are provided between parentheses. Optimal solutions were obtained using the branch-and-bound procedure introduced in Section 5. Results are reported for the following techniques:

- **Infinite capacity**: This is a technique that reflects traditional, one-dimensional bid selection models, where the manufacturer's capacity is ignored. Specifically, for each order and each component, the manufacturer selects the cheapest bid compatible with the order's due date. Orders are then scheduled according to the

ATC dispatch rule, namely the same rule used in our Simulated Annealing procedure (See Section 7).

- **Finite capacity**: Results are reported for a number of variations of our search heuristics:
    - Simulated Annealing (SA) procedure: This is the procedure introduced in Section 7 with $Temp_0$=300, α=0.95, N=60 and K=40. The procedure was run five times on each problem and we report both average performance and best performance over 5 runs.
    - Pseudo-Early/Tardy (PET) Procedure: this is the pseudo-early/tardy heuristic introduced in Section 6. Here again we report results for several variations of this heuristic:
        - *G-L* uses global earliness weights in its release policy and local earliness weights in its priority computations,
        - *L-L* uses local earliness weights for both release date and priority computations,
        - *Det* is a deterministic variation of the pseudo-early/tardy heuristic described in Section 6, namely $dev_1 = dev_2 = 0$,
        - *Rand* is a stochastic variation of the same heuristic with dev_1 = 0.3 and dev_2 = 0.3. For comparison sake, the CPU time given to this heuristic was the same CPU time required by an average SA run in the same problem category,
        - *Rand-RS* improves the solutions produced by *Rand* using the post-processing procedure described in Section 8,
        - *Hybrid* is a heuristic that runs SA once, PET/L-L/Rand-RS once, PET/G-L/Rand-RS once and takes the best of the resulting solutions.

Table 1. Percentage deviation from the optimum – 10-order/early-bid/medium-load problems (Standard deviations are provided between parentheses)

	Inf. Cap.	Finite Capacity								
		SA		PET						Hyb.
				L-L			G-L			
		Avg. of 5 Runs	Best of 5 Runs	Det	Rand	Rand-RS	Det	Rand.	Rand-RS	
np/nd	35.04 (36.84)	26.68 (24.04)	22.39 (21.29)	1.82 (333)	0.29 (0.64)	0.19 (0.57)	1.89 (4.69)	0.03 (0.12)	0.00 (0.00)	0.00 (0.00)
wp/nd	37.57 (26.42)	30.87 (20.14)	23.04 (14.21)	2.06 (4.36)	0.03 (0.12)	0.03 (0.12)	2.71 (5.28)	0.04 (0.18)	0.03 (0.18)	0.00 (0.00)
np/wd	92.25 (84.21)	42.43 (38.62)	25.76 (24.71)	7.31 (5.25)	3.45 (4.12)	1.88 (2.42)	3.01 (2.86)	0.48 (0.81)	0.46 (0.79)	0.37 (0.61)
wp/wd	76.58 (72.95)	39.01 (28.34)	24.79 (18.88)	9.74 (9.20)	4.08 (5.70)	3.19 (4.97)	4.75 (5.52)	1.14 (2.52)	1.03 (2.51)	0.80 (2.45)

Table 2. Percentage deviation from the optimum – 10-order/early-bid/heavy-load problems (Standard deviations are provided between parentheses)

	Inf. Cap.	Finite Capacity								
		SA		PET						Hyb.
				L-L			G-L			
		Avg. of 5 Runs	Best of 5 Runs	Det	Rand	Rand-RS	Det	Rand	Rand-RS	
np/nd	37.16 (34.79)	25.40 (22.56)	20.12 (18.84)	7.85 (5.76)	2.31 (2.71)	2.15 (2.55)	8.21 (6.62)	2.65 (3.91)	2.39 (3.67)	1.24 (1.64)
wp/nd	47.83 (41.49)	37.97 (31.14)	33.00 (26.83)	14.64 (12.11)	3.59 (6.32)	3.51 (6.28)	13.75 (12.34)	3.13 (4.46)	3.07 (4.31)	1.93 (2.92)
np/wd	131.11 (80.67)	56.17 (25.44)	36.53 (19.36)	14.51 (7.84)	6.50 (4.03)	4.92 (3.11)	10.23 (8.74)	3.09 (4.04)	3.02 (4.09)	2.28 (2.16)
wp/wd	165.93 (100.54)	77.20 (36.86)	52.08 (23.56)	20.65 (8.90)	10.34 (6.33)	9.47 (6.04)	18.74 (8.33)	6.13 (5.12)	5.75 (4.36)	5.23 (3.63)

Table 3. Percentage deviation from the optimum – 10-order/mixed-bid/medium-load problems (Standard deviations are provided between parentheses)

	Inf. Cap.	Finite Capacity								
		SA		PET						Hyb.
				L-L			G-L			
		Avg. of 5 Runs	Best of 5 Runs	Det	Rand	Rand-RS	Det	Rand	Rand-RS	
np/nd	89.73 (78.07)	52.86 (40.77)	30.24 (22.09)	16.92 (7.87)	8.75 (6.81)	6.30 (5.02)	3.90 (4.25)	0.85 (1.25)	0.62 (1.13)	0.57 (1.14)
wp/nd	59.71 (52.40)	36.41 (26.04)	24.23 (16.98)	17.78 (9.38)	11.06 (8.85)	7.30 (6.15)	4.72 (5.03)	2.04 (3.10)	1.94 (3.08)	1.41 (2.01)
np/wd	86.09 (84.32)	55.73 (58.56)	35.44 (42.25)	17.80 (10.17)	10.40 (7.29)	7.92 (6.82)	5.66 (3.36)	2.29 (2.41)	1.65 (1.80)	1.62 (1.82)
wp/wd	106.61 (79.84)	81.53 (65.82)	61.93 (59.13)	22.60 (10.25)	15.24 (10.17)	11.29 (7.82)	8.46 (5.58)	4.30 (3.61)	3.44 (2.80)	3.22 (2.80)

Table 4. Percentage deviation from the optimum – 10-order/mixed-bid/heavy-load problems (Standard deviations are provided between parentheses)

	Inf. Cap.	Finite Capacity								
		SA		PET						Hyb.
		Avg. of 5 Runs	Best of 5 Runs	L-L			G-L			
				Det	Rand	Rand-RS	Det	Rand	Rand-RS	
np/nd	187.43 (112.91)	79.69 (43.21)	41.71 (17.80)	19.04 (6.10)	13.82 (5.94)	13.29 (6.17)	14.13 (8.44)	4.17 (4.02)	4.10 (3.87)	3.98 (3.62)
wp/nd	126.71 (74.62)	71.04 (53.18)	40.29 (30.19)	21.37 (9.47)	12.45 (5.93)	10.74 (5.36)	15.18 (7.68)	6.32 (5.01)	6.06 (5.01)	5.68 (4.46)
np/wd	253.74 (170.31)	146.23 (127.82)	76.65 (82.99)	17.91 (5.77)	12.68 (5.29)	12.37 (5.31)	11.65 (6.52)	5.08 (4.69)	4.93 (4.55)	4.41 (3.60)
wp/wd	126.73 (104.02)	93.21 (77.45)	72.92 (71.58)	28.07 (10.77)	16.67 (9.13)	15.84 (9.24)	15.74 (8.71)	6.47 (5.11)	6.41 (5.09)	6.41 (5.09)

Tables 1-4 yield a number of observations:

- **Importance of the FCMCP Model:** All finite capacity heuristics yield solutions with significantly lower costs than the infinite capacity one, thereby confirming the importance of the FCMCP model. Taking into account finite capacity considerations and tightly coordinating the procurement of the multiple components required by each order subject to these finite capacity considerations significantly improve the manufacturer's bottom line. The results are generally most impressive on problem categories np/wd and wp/wd, where our hybrid heuristic respectively reduces total costs by at least 70% on *Early-Bid* problems and 80% on *Mixed-Bid* problems.
- **Distance from the Optimum:** Our hybrid heuristics yields solutions that are respectively less than 3.3% from the optimum on medium-load/*Mixed-Bid* problems and 6.5% from the optimum on heavy-load/*Mixed-Bid* problems. Also, our solutions are respectively less than 0.8% from the optimum on medium-load/*Early-Bid* problems and 5.3% from the optimum on heavy-load/*Early-Bid* problems. In particularly, PET-GL-Rand-RS gets the optimal solutions on all 20 randomly generated *Early-Bid* ml/np/nd problems.
- **Effectiveness of Property 1:** Even deterministic versions of the PET heuristic using G-L yields solutions that are respectively within 4.8% from the optimum on medium-load/*Early-Bid* problems and 18.8% from the optimum on heavy-

load/*Early-Bid* problems. Even without right-shifting improvement, our stochastic version of the PET heuristic using G-L yields solutions that are respectively within 1.2% from the optimum on medium-load/*Early-Bid* problems and 6.2% from the optimum on heavy-load/*Early-Bid* problems. This strongly suggests that Property 1 and the way in which our PET heuristic approximates earliness costs are rather effective. Note also that this deterministic version of our heuristic takes only a tiny fraction of a second on these problems.

- **PET heuristic versus SA heuristic:** Given the same amount of CPU time, the PET heuristic performs significantly better than the SA search procedure.
- **Effectiveness of Right-Shifting to Improve PET:** Even for the *Early-Bid* problems, on average, **PET-Rand-RS** respectively improves the solutions given by **PET-Rand** by 0.47% using L-L and by 0.12% using G-L. A look at the results on *Mixed-Bid* problems confirms the effectiveness of **PET-Rand-RS** in the cases where $dd - du > r^{latest}$, and the average improvements are respectively 2.00% using L-L and 0.29% using G-L.
- **Global versus Local Earliness Weights:** The G-L variation of the PET heuristic generally performs much better than L-L on both medium and heavy load problems, particularly on category wp/wd: by as much as 3.7% better on *Early-Bid* problems and by 9.4% on *Mixed-Bid* problems (L-L performs only slightly better than G-L on category *Early-Bid*/hl/np/nd). Additional results not reported here show however that local earliness weights yield significantly better results than global earliness weights when it comes to priority computations. These results confirm our intuition that local earliness weight computations better capture the changing profiles of non-dominated bid combinations, and are better suited for the computation of local priorities, whereas global earliness weights are more appropriate for the computation of order release dates - which require a more global perspective.

Impact of Ignoring the Manufacturer's Capacity on Larger Problems

Figure 7 and Table 5 summarize results evaluating the impact of ignoring the manufacturer's finite capacity on larger problems with 500 orders in medium load

situations. Similar results for heavy load situations are provided in Figure 8 and Table 6. It can be seen that the PET-Rand-RS systematically yields much better results than the infinite capacity policy. A look at the cost breakdowns provided in Table 5 and 6 indicates that PET is capable of selectively sacrificing procurement costs to yield significant reductions in tardiness costs. On some problems, PET reduces overall costs by more than 90%. These results further validate the benefits of the FCMCP model advocated in this paper and the way in which our PET heuristic leverages Property 1.

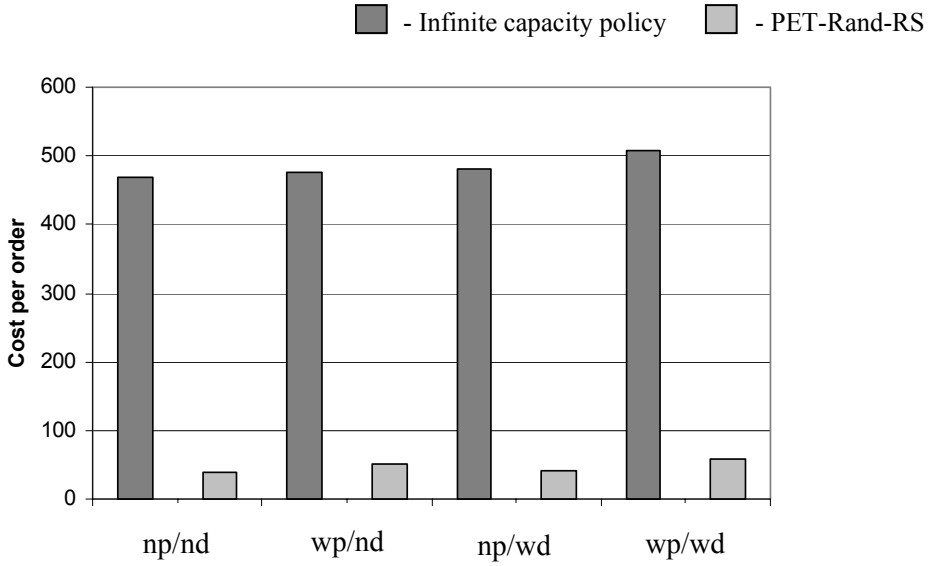


Figure 7. Infinite capacity policy vs. PET-Rand-RS heuristic: average overall cost per order (500-order/medium-load)

Table 5. Cost breakdown (500-order/medium-load)

ML		**Total Cost per Order**	Procurement Cost	Tardy Cost
np/nd	Inf. Cap.	469.2 (42.9)	30.6 (0.1)	438.6 (42.8)
	PET-Rand-RS	39.6 (3.0)	36.6 (0.4)	2.9 (3.0)
wp/nd	Inf. Cap.	475.9 (43.8)	38.2 (0.3)	437.7 (43.7)
	PET-Rand-RS	52.1 (2.6)	49.6 (1.0)	2.5 (2.3)
np/wd	Inf. Cap.	482.1 (40.9)	31.4 (0.2)	450.7 (41.0)
	PET-Rand-RS	41.2 (2.9)	38.5 (0.5)	2.8 (2.3)
wp/wd	Inf. Cap.	508.2 (43.0)	40.0 (0.5)	468.2 (43.1)
	PET-Rand-RS	57.7 (2.9)	54.0 (1.0)	3.6 (2.6)

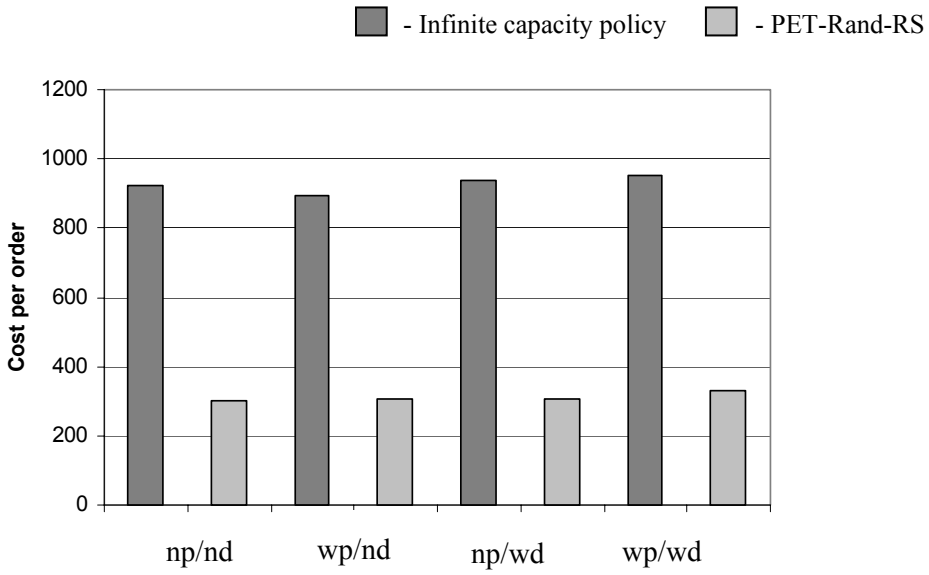


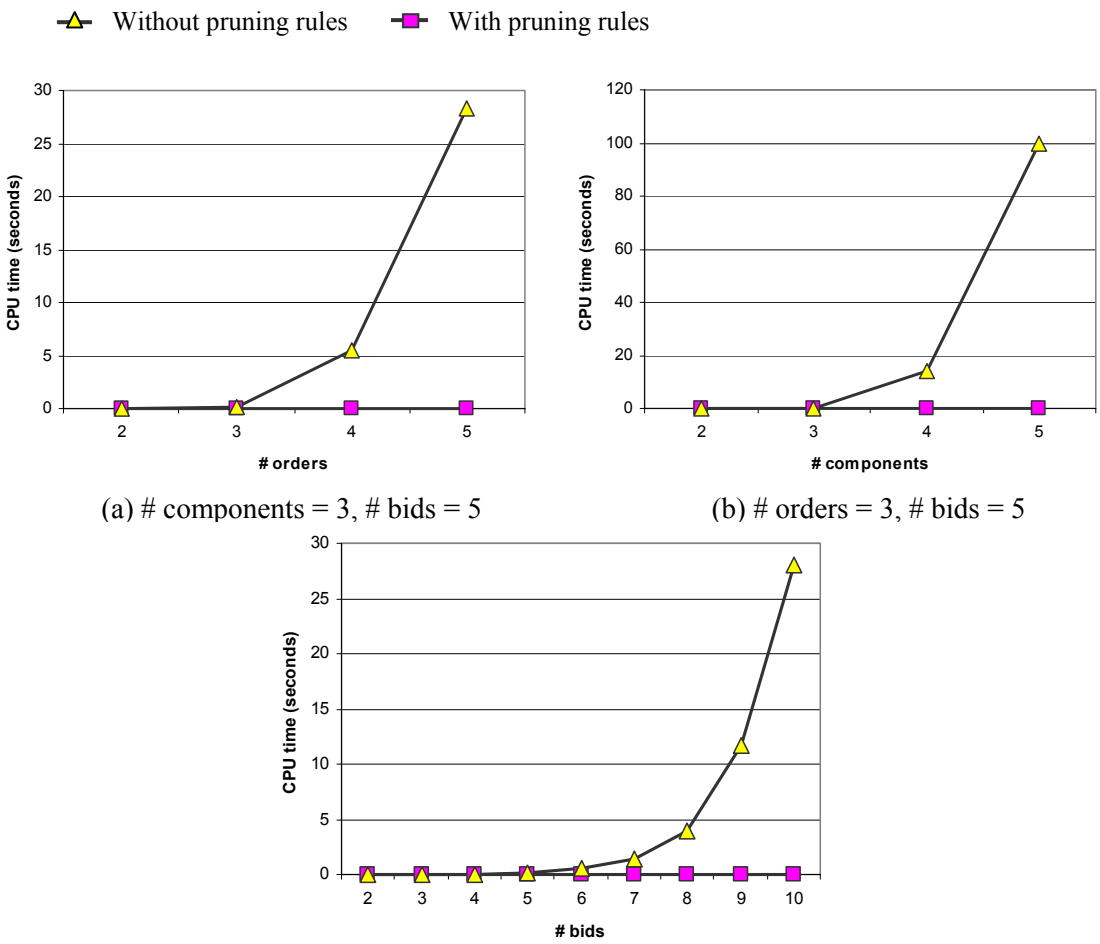
Figure 8. Infinite capacity policy vs. PET-Rand-RS heuristic: average overall cost per order (500-order/heavy-load)

Table 6. Cost breakdown (500-order/heavy-load)

ML		Total Cost per Order	Procurement Cost	Tardy Cost
np/nd	Inf. Cap.	922.8 (81.6)	31.1 (0.1)	891.6 (81.6)
	PET-Rand-RS	301.3 (36.8)	37.0 (0.7)	264.3 (36.6)
wp/nd	Inf. Cap.	895.8 (62.2)	39.2 (0.3)	856.6 (62.2)
	PET-Rand-RS	307.5 (31.1)	49.1 (0.8)	258.5 (31.2)
np/wd	Inf. Cap.	936.4 (60.5)	32.7 (0.2)	903.7 (60.5)
	PET-Rand-RS	304.8 (30.0)	39.2 (0.7)	265.6 (30.1)
wp/wd	Inf. Cap.	953.2 (78.4)	42.3 (0.4)	910.9 (78.4)
	PET-Rand-RS	328.0 (33.4)	53.5 (0.8)	274.4 (33.4)

Effectiveness of Pruning Rules

The CPU time required to find an optimal solution with our branch-and-bound algorithm is plotted in Figure 9 with and without the three pruning rules. Without the rules, CPU time increases exponentially with the number of orders, the number of components per order and the number of bids per component. In contrast, when using the three pruning rules, CPU time increases much more slowly with problem size, and is only around 0.1 second on these problems. This confirms the effectiveness of our three pruning rules in reducing the search space. All CPU times were obtained using a 1GHz Pentium-III computer.



(a) # components = 3, # bids = 5

(b) # orders = 3, # bids = 5

(c) # orders = 3, # components = 3

Figure 9. Effectiveness of pruning rules

Computational Requirements

Our computational results suggest that the CPU time required by the SA procedure increases almost linearly with problem size, while that of branch-and-bound grows exponentially (see Figure 10). By design, the CPU time allocated to the PET-Rand heuristic was set to be equal to that of the SA procedure. CPU times of Infinite Capacity Policy, PET-Rand and PET-Rand-RS on large problems of 50 to 500 orders are reported in Figure 11. As expected, the infinite capacity heuristics is the fastest, though PET does not require more than 12 seconds on these large problems. It can also be seen that the additional time required by the RS solution improvement procedure is negligible .

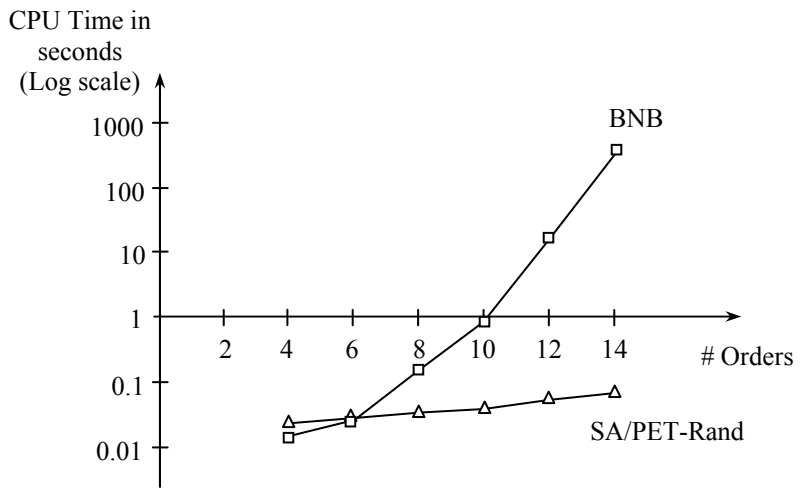


Figure 10. CPU time on small problems

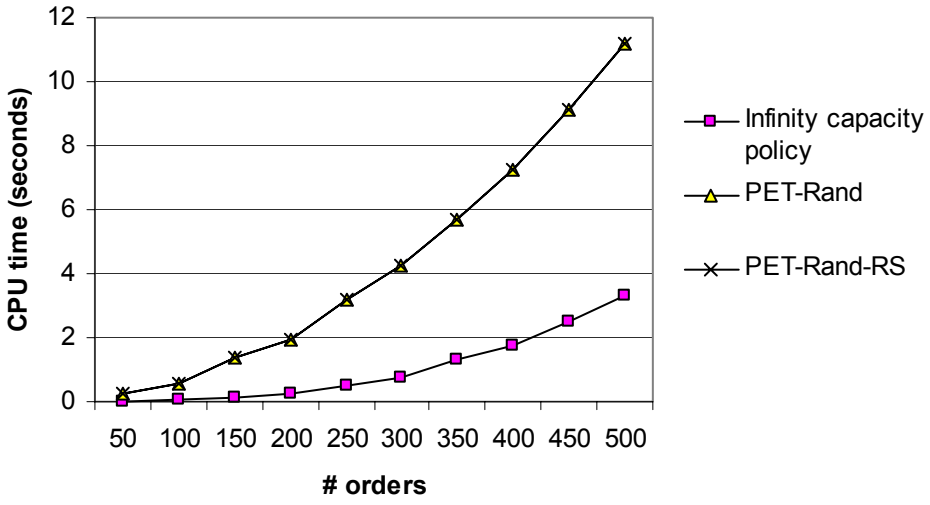


Figure 11. CPU time on large problems

10. RELAXING THE LOT-FOR-LOT ASSUMPTION

To take advantage of price discounts or to reduce fixed ordering costs, the manufacturer may want to consolidate its procurement orders for the components required for different customer orders. More generally, customer order quantities and quantities in supply bids may not match. This section outlines how the pruning techniques and pseudo-ET heuristic introduced earlier can be extended to deal with

this more general situation. This includes dealing with inventory costs as well as with customer orders for products with overlapping BOMs. Additional details can be found in Appendix A.

The key insight behind extending the techniques presented earlier is to combine supply bids for a given component into bundles large enough to satisfy a customer order's need (for that component), organize these bundles by the dates by which they can be fully delivered, and for each possible delivery date (and each customer order) to identify the cheapest available bundle ("dominant bundle"). More specifically, one can build a procedure that repeatedly cycles through the following three steps:

1. It identifies dominant component bundles for the components needed by each customer order – taking into account estimates of the cost of buying more than is immediately needed;
2. The procedure treats these bundles as virtual supply bids that can be pruned using the pruning rules introduced in Section 4; and
3. It uses the pseudo-ET heuristic to decide in which sequence to dispatch orders.

The costs associated with different component bundles reflect the purchasing price of the individual bids they comprise as well as the possible inventory costs incurred if the total quantity of an individual bid is not immediately consumed. This latter cost is approximated using an expected component consumption rate (e.g. based on orders that have not yet been dispatched or, more generally, based on forecasting data). Because we allow for customer orders with overlapping BOMs, each time an order is dispatched, component bundles need to be recomputed for the remaining orders. This is further detailed in Appendix A. These extensions have been implemented and tested in the context of the Supply Chain Trading Agent Competition ("TAC-SCM") [1, 42], a simulated, multi-period environment in which a number of different software agents acting on behalf of competing firms need each day to decide which customer request for quote to bid on and which component supply bid to accept. Since its inception in 2003, this annual tournament has each year attracted somewhere between 20 and 30 teams from around the world.

11. CONCLUDING REMARKS

Prior work on dynamic supply chain formation has generally ignored capacity and delivery date considerations. In this paper, we have introduced a deterministic model for finite capacity muti-component procurement problems faced by firms that 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 (or service provider) to quickly eliminate uncompetitive combinations of bids and have shown that the resulting problem can be modeled as a pseudo-early/tardy problem with stepwise earliness costs. A branch-and-bound algorithm, a randomized pseudo-early/tardy search heuristic and a Simulated Annealing 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 these procedures greatly improve over simpler infinite capacity bid selection models. Comparison with optimum solutions obtained using branch-and-bound, suggest that a hybrid heuristic that combines our PET and SA procedures generally yields solutions that are within a few percent of the optimum.

It should also be noted that the model and techniques presented in this paper can easily be generalized to accommodate situations where the manufacturer can process multiple orders at the same time (non-unary capacity) or where the manufacturer incurs setup times for switching production between different product families. This is true for the pruning rules we introduced as well as the branch-and-bound procedure and two heuristic search procedures. At the same time, we have not attempted to evaluate our techniques on these problems and hence do not know, for instance, how far our heuristic search procedures would be from the optimum. It is also worth noting that our pruning rules also apply to situations where the manufacturer is modeled as a more complex job shop environment, where each order has to flow through a (possibly different) succession of machining (or service) centers. Future work will aim to refine our model in support of dynamic profitable-to-promise functionality, where the manufacturer needs to determine how to respond to requests

for bids from prospective customers while also selecting among procurement bids from prospective suppliers [1, 42].

ACKNOWLEDGEMENTS

The research reported in this paper has been funded by the National Science Foundation under ITR Grant 0205435.

REFERENCES

[1] R. Arunachalam and N.M. Sadeh. The supply chain trading agent competition, *Electronic Commerce Research and Applications* 4 (2004), 63–81.

[2] A. Atamturk and D.S. Hochbaum. Capacity acquisition, subcontracting and lot sizing, *Management Sci* 47(8) (2001), 1081-1100.

[3] Y. Bassok and R. Akella, Ordering and production decisions with supply quality and demand uncertainty, *Management Sci* 37(12) (1991), 1556-1574.

[4] D.R. Beil and L.W. Wein, An inverse optimization-based auction mechanism to support a multinattribute RFQ process, *Management Sci., Vol.* 49*, No. 11* (2003), 1529-1545.

[5] M. Bensaou, Portfolios of buyer-supplier relationships, *Sloan Management Review* 40(4) (1999), 35-44.

[6] M. Bichler and J. Kalagnanam, Configurable offers and winner determination in multi-attribute auctions, *European Journal of Operational Res* 160(2) (2005), 380-394

[7] Y.-K. Che, Design competition through multidimensional auctions, *RAND J. Econom* 24 (1993), 668-680.

[8] R.R. Chen, R.O. Roundy, R.Q. Zhang and G. Janakiraman, Efficient auction mechanisms for supply chain procurement, *Management Sci* 51(3) (2005), 467-482.

[9] C. Chu, J.M. Proth, Y. Wardi and X. Xie, Supply management in assembly systems: the case of random lead times. *Lecture Notes in Control and Information Sciences*

199: 11th International Conference on Analysis and Optimization of Systems: Discrete Event Systems (1994), 443-448.

[10] F.W. Ciarallo, R. Akella and T.E. Morton, A periodic review production planning model with uncertain capacity and uncertain demand optimality of extended myopic policies, *Management Sci* 40(3) (1994), 320-332.

[11] J. Collins, C. Bilot, M.L. Gini and B. Mobasher, Decision processes in agent-based automated contracting, *IEEE Internet Computing* 5 (2001), 61-72.

[12] A. Davenport, A. Hohner and J. Kalagnanam, Iterative Reverse Combinatorial Auctions with Side Constraints, IBM Research Report, 2002.

[13] R. Davis and R.G. Smith, Negotiation as a metaphor for distributed problem solving, *Artificial Intelligence* 20 (1983), 63-109.

[14] L. de Boer, G. Van Dijkhuizen and J. Telgen, Basis for modeling the costs of supplier selection: The economic tender quantity, *Journal of the Operational Research Society* 51(10) (2000), 1128-1135.

[15] J. Du and J.Y.T. Leung, Minimizing total tardiness on one processor is NP-hard, *Mathematics of Oper Res* 15 (1990), 483-495.

[16] ebXML Technical Architecture Project Team, ebXML Technical Architecture Specification v.1.0.4, Available at http://www.ebxml.org/specs/ebTA.pdf.

[17] W.J. Elmaghraby, Supply contract competition and sourcing policies, *Manufacturing and Services Operations Management* 2(4) (2000), 350-371.

[18] P. Faratin and M. Klein, Automated contract negotiation as a system of constraints, *Proceedings of the Workshop on Distributed Constraint Reasoning, IJCAI-01,* Seattle, WA, August 2001, 33-45.

[19] J. Gallien and L.M. Wein, A simple effective component procurement policy for stochastic assembly systems, *Queueing Systems* 38(2) (2001), 221-248..

[20] R. Gonen and D. Lehmann, Optimal solutions for multiunit combinatorial auctions: Branch and bound heuristics, *Proc. ACM Conf. Electronic Commerce (ACM-EC)*, Minneapolis, MN (2000), ACM, New York, 13-20.

[21] H. Gurnani, R. Akella and J. Lehoczky, Optimal order policies in assembly systems with random demand and random supplier delivery, *IIE Trans* 28 (1996), 865-878.

[22] H. Gurnani, R. Akella and J. Lehoczky, Supply management in assembly systems with random yield and random demand, *IIE Trans* 32 (2000), 701-714.

[23] W. Hopp and M. Spearman, Setting safety lead times for purchased components in assembly systems, *IIE Trans* 25 (1993), 2-11.

[24] K. Jain and E.A. Silver, The single period procurement problem where dedicated supplier capacity can be reserved, *Naval Res Logist* 42 (1995), 915-934.

[25] J. Kalagnanam and D.C. Parkes, Auctions, bidding and exchange design, In D. Simchi-Levi et al., editors, *Handbook of Quantitative Supply Chain Analysis: Modeling in the E-Business Era*. Kluwer, 2004.

[26] U. Karmarkar and S. Lin, Production planning with uncertain yield and demand, Working Paper, William E. Simon Graduate School of Business Administration, University of Rochester (1986).

[27] P.R. Kleindorfer and D.J. Wu, Integrating long- and short-term contracting via business-to-business exchanges for capital-intensive industries, *Management Science* 49(11) (2003), 1597-1615.

[28] A. Kumar, Component inventory costs in an assembly problem with uncertain supplier lead-times, *IIE Trans* 21 (1989), 112-121.

[29] Language123.com. http://language123.com.

[30] T.M. Malone, *The Future of Work*, Harvard Business School Press, 2004.

[31] P.R. Milgrom, An economist's vision of the B-to-B marketplace, Executive white paper, www.perfect.com, 2000.

[32] T.E. Morton, S.R. Lawrence, S. Rajagopolan and S. Kekre. SCHED-STAR: A price-based shop scheduling module, *Journal of Manufacturing and Operations Management* 1(1) (1988), 131-181.

[33] N. Nisan, Bidding and allocation in combinatorial auctions. *Proc. ACM Conf. Electronic Commerce (ACM-EC)*, Minneapolis, MN (2000), ACM, New York, 1-12.

[34] OASIS: UDDI: Executive Overview: Enabling Service Oriented Architecture, Available at http://uddi.org/pubs/uddi-exec-wp.pdf, 2004.

[35] P.S. Ow and T.E. Morton, The single machine early/tardy problem, *Management Sci* 35(2) (1989), 177-191.

[36] B. Peleg, H.L. Lee and W.H. Hausman, Short-term e-procurement strategies versus long-term contracts, *Production and Operations Management* 11(4) (2002), 458-479.

[37] M. Pinedo, *Scheduling: Theory, Algorithms, and Systems*, Prentice Hall, Upper Saddle River NJ, 1995.

[38] Programmingbids.com, http://www.programmingbids.com.

[39] D.F. Pyke and M.E. Johnson, Sourcing strategy and supplier relationships: Alliances vs. eProcurement, In C. Billington, H. Lee, J. Neale, T. Harrison (editors), *The Practice of Supply Chain Management,* Kluwer Publishers (2003), pp. 77-89.

[40] M.H. Rothkopf, A. Pekeč and R.M. Harstad, Computationally manageable combinational auctions, *Management Sci* 44(8) (1998), 1131-1147.

[41] N.M. Sadeh, Micro-opportunistic scheduling: The Micro-Boss factory scheduler, In B. Zweben and M. Fox editors, *Intelligent Scheduling*, Morgan Kaufmann Publishers, 1998.

[42] N.M. Sadeh, R. Arunachalam, R. Aurell, J. Eriksson, N. Finne and S. Janson, TAC'03: a supply chain trading competition, *AI Magazine* 24 (2003), 92-94.

[43] T. Sandholm, S. Suri, A. Gilpin and D. Levine, CABOB: A Fast Optimal Algorithm for Winner Determination in Combinatorial Auctions, *Management Sci* 51(3) (2005), 374-390.

[44] H. Shore, Setting safety lead-times for purchased components in assembly systems: A general solution procedure, *IIE Trans* 27 (1995), 634-637.

[45] J. Song, C.A. Yano and P. Lerssrisuriya, Contract assembly: dealing with combined supply lead time and demand quantity uncertainty, *Manufacturing and Service Oper Management* 2(3) (2000), 287-296.

[46] E.D. Stanley, D.P. Honig and L. Gainen, Linear programming in bid evaluation, *Naval Res Logist* 1 (1954), 48-52.

[47] J.A. Van Mieghem, Coordinating investment, production, and subcontracting, *Management Sci* 45(7) (1999), 954-971.

[48] A. Vepsalainen and T.E. Morton, Priority rules and lead time estimation for job shop scheduling with weighted tardiness costs, *Management Sci* 33(8) (1987), 1035-1047.

[49] W3C, SOAP Version 1.2 Part 1: Messaging Framework, W3C Recommendation, June 2003, Available at http://www.w3.org/TR/soap12-part1/.

[50] W3C, Web Services Description Language (WSDL) 1.1, Note 15 March 2001, Available at http://www.w3.org/TR/wsdl.

[51] J.P. Womack, D.T. Jones and D. Roos. *The Marchine That Changed the World: The Story of Lean Production*, Harper Perennial, 1991.

[52] C. Yano, Stochastic lead times in two-level assembly systems. *IIE Trans* 19 (1987), 371-378.

APPENDIX A: DETAILS ON RELAXING THE LOT-FOR-LOT ASSUMPTION

As indicated in Section 10, relaxing the lot-for-lot assumption made in this paper can be done using a procedure that repeatedly cycles through the following three steps:

1. It identifies dominant component bundles for the components needed by each customer order – taking into account estimates of the cost of buying more than is immediately needed;
2. The procedure treats these bundles as virtual supply bids that can be pruned using the pruning rules introduced in Section 4; and
3. It uses the pseudo-ET heuristic to decide in which sequence to dispatch orders.

This is further detailed below.

1. Bundling

Under this extended model, because supply bids are no longer assumed to match customer order quantities, it may become necessary to combine multiple supply bids to satisfy the component requirements of a given order. Let oq_i be the product quantity required by customer order *i*. We continue to denote that order's due date as dd_i and its processing time (or duration) as du_i. Let bom_{ij} denote the number of type *j* components required for one product unit of order *i*.

We further define $S_j(dl) = \{B_j^1,...,B_j^{n_j(dl)}\}$ as the set of all supply bids for component *j* that arrive by some delivery date *dl*. In other words, each bid B_j^k's delivery date dl_j^k is such that $dl_j^k \le dl$. Let q_j^k and bp_j^k respectively denote that bid's quantity and unit bid price, i.e., $B_j^k = (dl_j^k, bp_j^k, q_j^k)$. Each delivery date dl_j^k can potentially be the basis for a bid bundle for a component *j* required by a given order *i* to the extent that, by itself or in combination with other supply bids set to arrive by dl_j^k, it can provide $bom_{ij} \cdot oq_i$ type *j* components. Accordingly, given a component *j* required by an order *i* and a possible delivery date *dl*, we can define one or more bid bundles of the form $BB_{ij}^l(dl) = \{\hat{B}_{ij}^{1l},...,\hat{B}_{ij}^{n_j(dl)l}\}$, where each bundle is a collection of fully or partially

consumed supply bids for component *j*. Each fully or partially consumed bid $\hat{B}_{ij}^{kl} \in BB_{ij}^{l}(dl)$ is defined in relation to an actual bid $B_j^k \in S_j(dl)$, with $\hat{B}_{ij}^{kl} = (dl_j^k, bp_j^k, \hat{q}_{ij}^{kl})$ where $\hat{q}_{ij}^{kl} \in [0, q_j^k]$ is the number of units consumed from this particular bid and $\sum_k \hat{q}_{ij}^{kl} = bom_{ij} \cdot oq_i$. Clearly, for a bid bundle to be viable, any of its component bids $\hat{B}_{ij}^{kl} \in BB_{ij}^{l}(dl)$ for which $\hat{q}_{ij}^{kl} > 0$ has to be fully purchased, even if it is only partially consumed.

Ideally, for each component *j* required by an order *i* and each possible delivery date *dl*, one would want to find the cheapest available bid bundle(s). These bundles could be obtained by solving the following problem:

Min

$$\sum_{k=1,n_j(dl)} \left\{ q_j^k \cdot bp_j^k \cdot \alpha_{ij}^k + \hat{q}_{ij}^k \cdot inv_j \cdot \max\{0, dd_i - du_i - dl_j^k\} + LO\_INV_{ij}^k \right\} + ADD\_PROC_{ij}(dl)$$

s.t.

$$\hat{q}_{ij}^k \le q_j^k,\ \hat{q}_{ij}^k \ge 0 \text{ and integral}, \forall k$$

$$\sum_k \hat{q}_{ij}^k \ge bom_{ij} \cdot oq_i$$

$$\alpha_{ij}^k \in \{0,1\} \text{ and } \hat{q}_{ij}^k \le \alpha_{ij}^k \cdot M,\ \forall k \ (M \text{ is some large number})$$

where,

- inv_j is the unit holding cost of component *j* per unit of time, and $inv_j \cdot \max\{0, dd_i - du_i - dl_j^k\}$ is thus the added inventory cost incurred for possibly holding a unit of component *j* between bid B_j^k's delivery date dl_j^k and the time when that unit is really needed for order *i*, namely $dd_i - du_i$
- α_{ij}^k is a binary variable indicating whether bid B_j^k is being used to satisfy order *i*'s need for type *j* components.
- $LO\_INV_{ij}^k$ is the "leftover" inventory cost associated with $q_j^k - \hat{q}_{ij}^k$, namely the part of bid B_j^k that is not used to satisfy order *i*.

- $ADD\_PROC_{ij}(dl)$ is the cost one can expect to incur for procuring additional type *j* components for orders other than order *i*, taking into account any surplus resulting from the collection of selected bids (α_{ij}^k). Such a term can be estimated by using an average per unit cost of type *j* components and by taking into account all the type *j* component requirements of orders that have not yet been dispatched – this quantity has to be pro-rated based on *dl*, the delivery date for which bid bundles are currently identified. This term enables us to take into account savings that can possibly be obtained by procuring quantities of type *j* components that exceed the requirements of order *i*, to the extent that these quantities are required for other orders. In other words, a given solution includes acquiring $\sum_k q_j^k$ but consuming only $bom_{ij} \cdot oq_i$ of these components for order *i*, leaving the difference for other possible orders that also require this component. If there are no such orders, the difference results in a net loss (since we pay $\sum_{k=1,n_j(dl)} \left\{ q_j^k \cdot bp_j^k \cdot \alpha_{ij}^k \right\}$). However, if such orders exist and the selected bids are cheap, the difference may possibly result in a saving, which should be taken into account when evaluating different possible bid bundles. $ADD\_PROC_{ij}(dl)$ is zero if $\sum_k q_j^k$ exceeds the total remaining requirements for type *j* components (including those of order *i*). Otherwise it is simply the cost of acquiring those type *j* components that are still needed beyond the $\sum_k q_j^k$ components provided by this bid bundle (using an average unit cost).

Different possible approximations can be made to estimate $LO\_INV_{ij}^k$ and $ADD\_PROC_{ij}(dl)$. Below, we briefly describe an approximation we have implemented in the context of the Supply Chain Trading Agent Competition ("TAC-SCM") [1, 42]. This is a simulated, multi-period environment in which a number of different software agents acting on behalf of competing firms need to each day decide which customer request for quote to bid on and which component supply offer to

accept. Since 2003, the competition has been organized as an annual event, attracting in excess of 30 teams from around the world each of the past two years. In such a multi-period environment (the competition simulates the operation of a supply chain over a full year with one-day periods), it is reasonable to assume that, within a limit, excess component quantities acquired on a given day will eventually be consumed in later periods. When this is the case, finding the cheapest available bid bundle(s) for a given order *i*, a given component type *j* and a given delivery date *dl*, can be reformulated as a follows:

Min

$$\sum_{k=1,n_j(dl)} \left\{ q_j^k \cdot (bp_j^k - ap_j) \cdot \alpha_{ij}^k + \hat{q}_{ij}^k \cdot inv_j \cdot \max\{0, dd_i - du_i - dl_j^k\} \right\} + LO\_INV_{ij}$$

s.t.

$$\hat{q}_{ij}^k \le q_j^k,\ \hat{q}_{ij}^k \ge 0 \text{ and integral}, \forall k$$

$$\sum_k \hat{q}_{ij}^k \ge bom_{ij} \cdot oq_i$$

$$\alpha_{ij}^k \in \{0,1\} \text{ and } \hat{q}_{ij}^k \le \alpha_{ij}^k \cdot M,\ \forall k \text{ } (M \text{ is some large number})$$

where:

- ap_j is the average unit price of type *j* components (e.g. based on historical data)
- LO_INV_{ij}, the "leftover" inventory cost, is approximated by assuming a steady consumption rate cr_j (expressed as a number of components consumed per unit of time), namely:

$$LO\_INV_{ij} = inv_j \frac{\left(\sum_k \alpha_{ij}^k \cdot q_j^k - bom_{ij} \cdot oq_i \right)^2}{2 \cdot cr_j}$$

Solving this quadratic problem would be too time-consuming to be practical. Instead, we further restrict ourselves to solutions that have at most one bid B_j^k for which $\alpha_{ij}^k = 1$ and $\hat{q}_{ij}^k < q_j^k$. While this assumption limits our ability to accept large,

significantly discounted supply bids, this can in part be remedied with efficient post-processing procedures that attempt to substitute accepted supply bids with cheaper ones (while respecting the schedule's procurement constraints). Under this new assumption, good supply bundles can efficiently be identified using a greedy procedure, where each available supply bid B_j^k for component *j* is ranked according to the following priority:

$$\rho_{ij}^k = -\{q_j^k(bp_j^k - avg\_price_j) + q_{ij}^k \cdot inv_j \cdot \max(0, dd_i - du_i - dl_j^k) + inv_j \frac{\left(\max(0, Q + q_j^k - bom_{ij} \cdot oq_i\right)^2}{2 \cdot cr_j}\},$$

where *Q* denotes the total quantity of type *j* components already selected for order *i* with the delivery date currently under consideration. The greedy procedure continues adding bids, always selecting the one with the highest priority (or lowest marginal cost) until $\sum_k q_{ij}^k \geq bom_{ij} \cdot oq_i$.

2. Pruning

Based on the above, we can also estimate the price of a bid bundle $BB_{ij}^l(dl) = \{\hat{B}_{ij}^{1l}, ..., \hat{B}_{ij}^{n_j(dl)l}\}$ as:

$$BBP[BB_{ij}^l(dl)] = \sum_{k=1,n_j(dl)} \left\{q_j^k \cdot (bp_j^k - ap_j) \cdot \alpha_{ij}^k + \hat{q}_{ij}^k \cdot inv_j \cdot \max\{0, dd_i - du_i - dl_j^k\}\right\} + LO\_INV_{ij}$$

Using these costs, we can prune bid bundles, using the same procedures introduced to prune bids in Section 4 – every bid bundle $BB_{ij}^l(dl)$ has a bid price $BBP[BB_{ij}^l(dl)]$ and an aggregate delivery date *dl*. Note that this model takes into account the inventory costs.

3. Dispatching

Finally, we can also use the pseudo-ET heuristic introduced in Section 6 to decide on the sequence in which to process orders. Each time an order is dispatched, we now need to update the list of dominant bid bundles for each remaining order (since orders

can have overlapping BOMs). In particular, any excess in components resulting from an accepted supply bid can be modeled as one or more artificial bids with price equal to *0* (since the cost of acquiring these bids has already been factored in). As already indicated earlier, this model extension has been successfully implemented in an entry to the TAC-SCM 2004 competition where supply chain trading agents need to make bidding and procurement decisions in simulated 15-second days [1, 42].