To appear in “Intelligent Information Agents”, Chapter 10, Springer, 1999, eds.
Matthias Klusch

# Dynamic Supply Chain Structuring for Electronic Commerce Among Agents*

Daniel Dajun Zeng
Katia Sycara
Graduate School of Industrial Administration
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA

November 12, 1998

**Abstract**

Electronic commerce and the vast amounts of real-time information available through means of EDI and the Internet are reshaping the way enterprises conduct business. A new computational infrastructure and models are needed for a business to gain a competitive edge through effective use of this information base. One of the key issues in competing in the electronic marketplace is product/service differentiation. Currently there are no computational models for multi-issue decision making in electronic commerce.

We develop a model of inter-organizational electronic commerce that explores various new choices and opportunities that the electronic marketplace offers. The particular motivating applications of our work

* This research has been sponsored in part by ONR Grant No. N00014-96-1-1222, and by NSF grant #IRI-9612131.

are supply chain management. Two major performance measures of supply chain activities are cost and leadtime. In our model, we explicitly address these two issues in a unified fashion for a variety of supply chain activities, such as outsourcing, supplier selection, production capacity, transportation mode selection, and inventory positioning. We model different business entities as autonomous software agents interconnected via the Internet. The main research focus of our efforts is how to coordinate software agents in supply chains dynamically and flexibly such that goods and services can be delivered at the right time in a cost-effective manner.

The supply chain structure is modeled by an AND/OR network. We develop an efficient algorithm for software agents in supply chains to evaluate the alternatives that offer different leadtime and cost parameters. We have coupled this model with operational level decision making such as stochastic inventory management. Experimental results show that our model results in significant improvement in solution quality as compared to traditional models.

# 1 Introduction

Current networking technology and the ready availability of vast amounts of real-time data and information on the Internet-based Infosphere bring to business decision makers more abundant and accurate information. Many online businesses specialize in delivering electronic catalog services and performing other intermediary functions such as business/product yellowpage and matchmaking. Emerging computing paradigms such as Internet-based software agents are also making locating and accessing information increasingly easier.

Electronic commerce is reshaping both consumer market and inter-organizational business. How to compete in the electronic marketplace effectively poses significant challenges to practitioners and researchers. In this paper, we focus on inter-organizational electronic commerce in the context of supply chain management. Researchers and practitioners have observed that the nature of competition in electronic commerce does not resemble undifferentiated Bertrand competition. This suggests that price alone is not the only decision criterion. Other decision criteria related with product/service differentiation need to be considered. Two of the prominent performance measures of a supply chain are cost and leadtime. By leadtime, we mean the amount of time that elapses from the instant that an order (or service request) is placed until it arrives. By cost, we mean the sum of the costs of all activities required to satisfy the order (or deliver the service). Some examples of these supply chain activities and decisions are:

- **Supplier selection**. Procurement management is playing an increasingly important role nowadays with the globalization of manufacturing and advances in network information infrastructure. There usually exists a rich set of suppliers offering raw materials of varying quality, cost, and delivery leadtime. Decisions regarding supplier selection have to be made after a careful evaluation of the impact of raw material cost and delivery time responsiveness on the supply chain as a whole.

- bf Subcontracting. In manufacturing, managers face "make or buy" decisions—the choice between making components/products in house or subcontracting them to outside sources. When making in house is not an option or is clearly suboptimal, management has to select the appropriate subcontractors from a pool of potential subcontractors that offer different levels of service under different prices. These decisions are critical in today's highly competitive and dynamic business environment.

- **Transportation mode selection**. Typically, multiple transportation modes are available to supply chain managers, offering a wide range of cost/time options. Decisions regarding which mode is best suited for the current order are dependent on how urgently the order needs to be filled, how expensive these modes are, and where this transportation activity is located in the supply chain network.

- **Assembly/subassembly**. Assembly/subassembly operations cannot start until all the components/subcomponents/raw materials required become available. Production managers need to make sure that all these materials are accessible for use at the right place and the right time.

- **Production rate decision**. Production rate decisions correspond with the choice between using faster, more expensive, high-capacity production facilities versus slower but cheaper facilities. To evaluate the tradeoffs between these options is not trivial considering all the upstream and downstream activities in the supply chain.

In this paper, we present a model of inter-organizational electronic commerce that explicitly addresses the time and cost issues in a unified fashion. We model different business entities as autonomous software agents interconnected via the Internet. These agents act on behalf of their human users/organizations in order to perform laborious information gathering tasks, such as locating and accessing information from various on-line information sources, filter away irrelevant or unwanted information, and provide decision support. Section 1.1 discusses some of the related literature. Section 2 presents a brief description of our supply chain model, called LCT (Leadtime-Cost-Tradeoff). The LCT model needs to be integrated with other operational level decision making models such as inventory management to enable intelligent agents to make the full range of supply chain decisions. Section 3 presents how to integrate LCT into the EOQ model where the demand rate is assumed to be constant. We present in Section 4 our model and analysis of stochastic inventory management given the leadtime/cost choices. Experimental results show that our model results in significant improvement in solution quality as compared to traditional models. Computing optimal policies for the resulting model proves to be computationally difficult. Section 5 presents a computational study of making inventory decisions that take advantage of leadtime/cost options. We conclude the paper in Section 6 by summarizing the results and pointing out other extensions to our agent-based multi-issue supply chain model.

## 1.1 Related Literature

Effective use of the Internet by individual users, organizations, or decision support machine systems has been hampered by some dominant characteristics of the Infosphere. Information available from the net is unorganized, multi-modal, and distributed on server sites all over the world. The availability, type and reliability of information services are constantly changing. In addition, information is ambiguous and possibly erroneous due to the dynamic nature of the information sources and potential information updating and maintenance problems. The notion of Intelligent Software Agents (e.g., [WJ95, SZ96]) has been proposed to address this challenge. In this paper, we model a supply chain as a multi-agent system where different business entities interact with one another through intelligent software agents that act on their behalf. In general, multi-agent systems can compartmentalize specialized task knowledge, organize themselves to avoid processing bottlenecks, and can be built expressly to deal with dynamic changes in the agent and information-source landscape. In addition, Multiple Intelligent Agents are ideally suited to the predominant characteristics of the Infosphere (and in particular supply chain management), such as the heterogeneity of the information sources, the diversity of information gathering and decision support tasks that the gathered information supports, and the presence of multiple users/organizations with related information and decision aiding needs.

In order for autonomous software agents to make sensible decisions in any nontrivial domain such as supply chain management, they need to have access to domain-specific decision making models and related computational mechanisms[LS97]. We briefly survey some of these models in literature that are most relevant to multi-issue (time and cost) supply chain management.

The first models that consider the possibility of purchasing shorter leadtimes at a premium cost appeared in [Bul64] and [Fuk64], among others. The main objective of these papers is to find the optimal ordering policy that minimizes ordering, holding and penalty costs when subject to random demand. Structural results regarding the optimal replenishment policy were established when there are only two options and the leadtimes of the two options differ by one time unit (in periodic review situations).

Kaplan in [Kap70] analyzed optimal policies for a dynamic inventory problem when the leadtime is a discrete random variable with known distribution. Assuming that outstanding orders do not cross in time, Kaplan derived the structure of optimal policies which is shown to be similar to those obtained with deterministic leadtimes. Although [Kap70] is not concerned with different leadtime options, it gives a good survey for the technical difficulties that we also encountered.

Song and others in [Son94] studied the impact of stochastic *leadtimes* on the optimal inventory decisions and the optimal cost in a base-stock inventory model. The focus there is to evaluate the impact of the variability of leadtimes but not to derive an inventory policy which makes use of the availability of multiple leadtime/cost options. In [LZ93] Lau and Zhao considered the order splitting between two suppliers that offer different leadtime with uncertainty. The authors assumed a constant splitting ratio among two suppliers and developed computational methods to compute the optimal ratio, ordering quantities and reordering point in a continuous review inventory setting. Several papers (e.g., [BDR94]) deal with situations where leadtimes is one of the decision variables. Their assumption is that by paying "leadtime crashing cost" leadtime reduction can be achieved. The goal of these papers is to find the single best leadtime option under single sourcing.

# 2 The LCT Supply Chain Model

We have developed a supply chain model, called LCT, based on an AND/OR network representation[HZ97]. This model is capable of capturing a variety of supply chain activities and decisions.

In LCT, a supply chain is modeled as a directed acyclic graph with parallel arcs. The model follows an activity-on-arc representation where each arc corresponds to a particular supply chain activity (production, transportation, subcontracting, etc.). Note that each activity/arc has two performance measures: leadtime and cost. In this supply chain network, nodes represent completion of activities and may be used to establish precedent constraints among activities. The graph is directed towards one particular "root node". The root node corresponds to the retailer of the product that the supply chain produces. End customers interact with the root node only. We define two types of nodes that each specifies conditions for satisfying prior activities: *conjunction* and *disjunction* nodes. Conjunction nodes or AND nodes are nodes for which *all* the activities that correspond to the incoming arcs must be accomplished before the outgoing activities can begin; whereas disjunction nodes or OR nodes requires that *at least one* of the incoming activities must be finished before the outgoing activities can begin.

Based on LCT we have developed efficient computation methods to identify the entire efficient frontier between leadtime and cost in supply chains. This efficient frontier at the "root node", i.e., the retailer point, compactly represents all the undominated, feasible combinations of supply chain activities. By a feasible combination of supply chain activities, we mean the set of activities that guarantee the availability of goods or services at the root node. We say a combination dominates the other when the former offers cheaper cost and shorter leadtime than the latter. Given the efficient frontier coupled with the market demand profile and pricing strategy at the root node, management can converge on the optimal tradeoff point specifying a particular supply chain configuration.

One of the limitations of LCT is that the model doesn't explicitly consider inventory. Without inventory, the solution concept based on the leadtime cost efficient frontier applies to "one-shot" scenarios in which single period demand is considered at the root node (e.g., make-to-order). If demand for the end product is repetitive, holding inventory at one or more places in the supply chain clearly has the potential of improving the performance of the whole system. The rest of the paper focuses on integrating LCT with inventory management—out first step to extend LCT to address multi-period demand.

In this paper, we assume that inventory can be held only at the root node[1]. In other words, we are concerned with integrating one stage inventory management within the context of LCT. Since we only add the inventory capacity at the root node, the entire efficient frontier between leadtime and cost in the supply chain network remains the same. We are interested in ways through which the end product retailer can take advantage of the availability of multiple options with varying leadtime and cost parameters. Despite the restrictive assumption made in this model as to the inventory location, this model captures the fundamental characteristics of a variety of supply chain management situations. For instance, the model is readily applicable for retailers who may get goods/services from various manufacturers that quote different unit price and delivery leadtime. In another example, a manufacturing firm is structuring its international sourcing base. Suppose that this firm adopts a make-to-stock policy. Our model can be applied to make sourcing decisions based on the current inventory stock level.

# 3 LCT in Inventory Models with Constant Demand Rates

In this section, we demonstrate how the LCT model can be integrated into inventory models that assume constant demand rates.

[1]The extension of LCT which allows inventory at arbitrary nodes in the supply chain network is beyond the scope of this paper.

Let $l$ denote the leadtime, $UP(l)$ the cheapest unit ordering cost for goods for which the order fulfillment takes at most $l$. The leadtime cost efficient frontier computed in LCT takes the form of the function $UP(l)$. When the leadtime measure can be properly discretized (e.g., in units of days), $UP(l)$ is a step function:

$$UP(l) = c_i \quad \text{if } i \leq l < i+1 \quad \text{for } i = 0, 1, \ldots, M$$

where $M$ is the maximum leadtime from all possible alternatives and $c_i$ is the minimum unit ordering cost if the target leadtime is expected to be strictly less than $i+1$. Without loss of generality, we assume that $c_i$ is nonincreasing with respect to $i$.

Let $SK(l)$ denote the setup cost associated with selecting the cheapest supply chain configuration that achieves leadtime $l$.

$$SK(l) = K_i \quad \text{if } i \leq l < i+1 \quad \text{for } i = 0, 1, \ldots, M$$

The total ordering cost $TC(x, l)$ for $x$ units of product with the leadtime requirement $l$ is given by

$$TC(x, l) = K_i + c_i x \quad \text{if } i \leq l < i+1 \quad \text{for } i = 0, 1, \ldots, M$$

We follow the standard assumptions of the EOQ inventory model: The demand rate $\lambda$ is constant; no stockout or backlogging is allowed. Consider the following situation which is a special case of our model. There is only one alternative available that offers leadtime $k$, setup cost $K$, and unit ordering price $c$. The optimal ordering policy in this case is well known. It follows the $(Q, R)$ policy, where $Q$ is the standard EOQ quantity, $R$ is the reorder point which is equal to $k\lambda$. (We assume $k \leq Q/\lambda$ for simplicity.)

Let's consider the general case where more than one alternatives are available. We only consider the cycle inventory which is the amount of inventory physically on hand at any point in time. The optimal ordering policy involves using the alternative with leadtime $i^*$ which is defined as follows:

$$\lambda c_{i^*} + \sqrt{2K_{i^*}\lambda h} \equiv \min_i (\lambda c_i + \sqrt{2K_i \lambda h})$$

This implies a single sourcing policy will be optimal. Search for $i^*$ can be easily done by enumerating all available modes. It is clear that when

$K_i$ is nonincreasing with respect to $i$, the least expensive alternative that also offers the longest leadtime is always the mode of choice. After the alternative $i^*$ has been chosen, the classical $(Q, R)$ policy can be used to determine the order amount and reorder point. Obviously, other extensions such as finite production capacity can be done in the same fashion without causing additional technical problems.

# 4 LCT in Periodic Review Stochastic Inventory Model

In the previous section, we show that integrating LCT with inventory models with constant demand rates can be easily done. When we consider inventory management with uncertain demand, however, the situation changes dramatically. In this section, we first present a formal formulation of the problem and then proves some formal properties. We demonstrate the technical difficulties of finding optimal policies and motivate our computational work (presented in Section 5) in finding effective suboptimal policies.

We study an $N$-period stochastic inventory problem in which there are $m$ different ordering options. These options represent different leadtime and cost tradeoffs which can be computed using the LCT model given the network topology of a supply chain and time/cost information for the supply chain activities. We ignore the setup cost in this study[2].

We use the following notation in our study. Most of the notation follows the standard one used in $N$-period single-stage stochastic inventory modeling:

$N =$ the number of periods in the planning horizon

$m =$ the number of delivery/production options. We assume that $m < N$.

$\lambda_i =$ the leadtime associated with option $i$, $i = 1, 2, \ldots, m$. We assume that these leadtimes are deterministic. Without loss of generality, we assume that $\lambda_i < \lambda_j$ when $i < j$.

$\tau =$ the maximum leadtime from all possible options. $\tau = \lambda_m$

[2]It is not entirely arbitrary since electronic commerce contributes to the setup cost reduction.

$c_i =$ the unit ordering cost with option $i$, $i = 1, 2, \ldots, m$. We assume that $c_i > c_j$ when $i < j$[3].

$t =$ the demand for the item during each period. We assume that the demand is stationary.

$e(x^+) =$ the salvage value of having $x^+ = max(x, 0)$ units of inventory on hand at the end of the period $N$. We assume $e(x^+)$ is convex.

$\alpha =$ the one-period discount factor

$x_1 =$ the current stock level

$x_2, x_3, \ldots, x_\tau =$ the outstanding orders such that $x_2$ is due at the start of the next period, $x_3$ is to be delivered two periods hence, etc.

$z_i =$ the amount of goods to be ordered at the start of the present period using option $i$, $i = 1, 2, \ldots, m$. These are the inventory decision variables.

$L(x) =$ the expected operational costs during the period, exclusive of ordering costs, w.r.t. the stock on hand at the beginning of the period:

$$L(x) = \begin{cases} \int_0^x h(x-t)f(t)dt + \int_x^\infty p(t-x)f(t)dt & x > 0, \\ \int_0^\infty p(t-x)f(t)dt & x \leq 0 \end{cases} \quad (1)$$

We assume that the holding cost $h(\cdot)$ and penalty cost $p(\cdot)$ are non-decreasing and convex. The unfulfilled ordered are backlogged. Since integration preserves the convexity, the convexity of $L(x)$ is easily seen.

$C_n(x_1, x_2, \cdots, x_\tau) =$ the minimum expected cost following an optimal policy, given that only $n$ future periods are to be taken into account, where $(x_1, x_2, \ldots, x_\tau)$ represents all the information about the current stock level as well as the amounts of goods whose orders have been submitted and are to be delivered during the following $\tau - 1$ periods.

To simplify the notation, we also use the following vector-based representations:

[3]This is not a restriction. See the discussion in Section 4.3

$\underline{x}$ : the row vector of $(x_1, x_2, \ldots, x_\tau)$

$\underline{z}$ : the row vector of $(z_1, z_2, \ldots, z_m)$

The functional equation for $C_n(\cdot)$ is easily seen to be

$$C_n(x_1, x_2, \ldots, x_\tau) = \min_{z_i \geq 0, \text{for } 1 \leq i \leq m} \Big\{ \sum_{i=1}^{m} c_i z_i + L(x_1 + y_0) + \tag{2}$$

$$+\alpha \int_0^\infty C_{n-1}(x_1 + y_0 - t + x_2, x_2 + y_1, \cdots, x_\tau + y_{\tau-1}, y_\tau) f(t) dt \Big\} \tag{3}$$

where, $y_i$ is defined as follows:

$$y_i = \begin{cases} z_{\lambda_j} & \text{if } i = \lambda_j, \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Under these assumptions, we can prove the convexity of the value function.

**Theorem 1** *$C_n(\underline{x})$ is convex.*

**Proof.** We prove Theorem 1 by induction. Given the cost structure of the salvage value,

$$C_0(\underline{x}) = p(x_1^-) + e(x_1^+) \tag{5}$$

$C_0(\underline{x})$ is convex.

By induction on $C_{n-1}$, we will prove that $C_n$ is also convex. We first introduce some auxiliary vectors to simplify the notation. Define

$$g(\underline{x}, \underline{z}, t) \equiv C_{n-1}(A \begin{bmatrix} \underline{x} \\ \underline{z} \\ t \end{bmatrix}) \tag{6}$$

where $A$ is a $\tau \times (\tau + m + 1)$ matrix. Each element of $A$, $A_{ij}$, is given as follows:

$$A_{ij} = \begin{cases} 1 & \text{if } i = 1 \text{ and } j = 1, \\ -1 & \text{if } i = 1 \text{ and } j = \tau + m + 1, \\ 1 & \text{if } j = i + 1 \text{ for all } i \in [1, \tau - 1], \\ 1 & \text{if } \lambda_j = i \text{ for all } j \in [\tau + 1, \tau + m], \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

It can be easily verified that the functional equation (2) can be rewritten as:

$$C_n(\underline{x}) = \min_{\underline{z}}\Big\{\sum_{i=1}^{m} c_i z_i + L(x+y_0) \qquad + \alpha \int_0^{\infty} g(\underline{x},\underline{z},t) f(t) dt\Big\} \qquad (8)$$

Given that $C_{n-1}$ is convex and that $A$ is a full-rank linear transformation, $g(\underline{x},\underline{z},t)$ is convex due to [Roc70] Theorem 5.7 Part A.

Define

$$q(\underline{x},\underline{z}) = \int_0^{\infty} f(t) g(\underline{x},\underline{z},t) dt \qquad (9)$$

We know that $q(\underline{x},\underline{z})$ is convex since $f(t) \geq 0$ due to [Ash72].

Since the operating cost $L$ is convex, and $q$ is convex, we conclude that the summation $\sum_{i=1}^{m} c_i z_i + L(x+y_0) + q(\underline{x},\underline{z})$ is convex. Due to [Roc70] Theorem 5.7 Part B, we know that $C_n$ is convex.

## 4.1 Optimal Inventory Control Policy

Karlin and Scarf in [AKS58] studied inventory models in the presence of a time lag. Their models can be viewed as a special case of ours since they assumed that there is only one leadtime option available. Based on the convexity of the objective function, they proved that the optimal policy follows an order-up-to policy. Simply put in our notation, if $m = 1$, $z_1^* = (S - (x_1 + x_2 + ... + x_{\lambda_1}))^+$, $S$ being the order-up-to level to be determined. Fukuda in [Fuk64] extended this result to deal with 2-mode cases. Again, based on the convexity of the objective function, he proved that the optimal control policy is very similar to an order-up-to policy except for an additional stock level up to which it is desired to order using the quicker and more expensive option. The intuition is that to use quicker option to handle large, unexpected demand while the steady portion of the demand flow is handled by the slower and less expensive option. In both cases, the optimal inventory policies are not difficult to compute.

One might think that the similar intuition may be extended to the general $m$ option case by having $m$ order-up-to levels for each leadtime option. Unfortunately, this is not the case. By solving a very simple 2-mode ($\lambda_1 = 0, \lambda_2 = 2$) problem, we found that no simple order-up-to or order-up-to like structures exists. The complexity of the problem is coming from the fact that although the convexity of the objective function holds, the optimal controls are functions of all the $x_i$ for $i = 1, 2, \ldots, \tau$ rather than functions of $\sum_{i=1}^{\tau} x_i$.

Using the value iteration approaches in dynamic programming, we can compute the optimal control policies regardless of whether they follow the order-up-to structure or not. However, these value iterations approaches are almost impossible to scale up since the size of state space itself is exponential with respect to the maximum leadtime. For practical purposes, we need to find other more efficient algorithms.

The same technical difficulties have been identified in different inventory management and dynamic programming settings. (e.g.,[Kap70]). In order to get analytically appealing results, the standard way of avoiding these difficulties in inventory management is to assume that at any certain moment, there is only one outstanding order. This is clearly not our option, since what we are interested is precisely using multiple options at the same time.

To address these computational issues, we have developed several sub-optimal polices that are easy to compute. In Section 5, we reported these policies and an experimental evaluation of their performances. To illustrate the significance of making use of multiple leadtime options, in Section 4.2 we use a numerical example to demonstrate that using multiple leadtime options can result in significant improvement in solution quality. As a side result, we establish in Section 4.3 a simple dominance relationship among leadtime options that can be used to eliminate certain options from consideration without compromising the solution quality. Admittedly, this dominance relationship does not help in worse case scenarios where no leadtime options are dominated by others. It could, however, save some computation by throwing out the dominated options.

## 4.2 An Numerical Example: The Value of Having Multiple Leadtime Options

In this section, we use a numerical example to demonstrate that having multiple leadtime options can result in significant improvement as compared to having one (Karlin and Scarf's model) or two leadtime options where the leadtime difference is 1 time unit (Fukuda's model).

Consider the following scenario. We assume that one period demand is discretely distributed according to the following probability mass function:

| $d$ | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| $p(d)$ | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 |

To construct a comparison baseline, we first start off with using one leadtime option only. We are interested in minimizing the infinite horizon average cost per stage since we want to explore the average performance of the system in steady states. Suppose that the unit holding cost $h = 1.00$ and the penalty cost $p = 30.00$. When there is only one leadtime option available, say, one with the ordering cost $c_1 = 15$ and the delivery leadtime $\lambda_1 = 0$ (instantaneous delivery), we know that the order-up-to policy is optimal. We compute the optimal order-up-to levels and find that the average cost per stage is 33.62. In effect, based on Karlin and Scarf's theorem[AKS58], we can computer the optimal order-up-to level and the average cost for any (one) option with a positive delivery leadtime. Using 33.62 as the target cost, we are interested in the ordering cost of other leadtime options that lead to the same 33.62 average cost. The isocost curve is given in Figure 4.2. The $x$-axis represents the delivery leadtime. The $y$-axis represents the ordering price required to achieve the target cost.

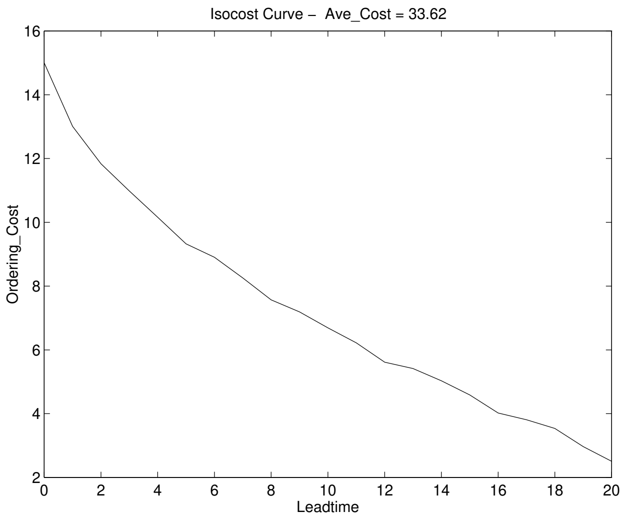


Figure 1: An Isocost Curve

For simplicity, let us consider the first three points on this isocost curve. If we can use only one option, we are indifferent among option 1 (leadtime 0; ordering cost 15.00), option 2 (leadtime 1; ordering cost 13.01), or option 3 (leadtime 2; ordering cost 11.84). Suppose that now we have two options—Option 1 and Option 2—available at the same time. How should we take advantage of this additional choice? Recall that this is exactly what Fukuda's model addresses (see Section 4.1). We compute the optimal control and find the average cost decreases from 33.62 to 30.46. Using option 2 and option 3 results in a average cost of 30.55, computed similarly.

How about using these three options together? Since the size of the problem is very small, we can afford to enumerate all the states in the state space and compute the optimal control for each state. We use a linear program coded in AMPL to compute the minimal steady state average cost. The result is 27.18. Percentage wise, it represents a 19.16% decrease in operating cost from 33.62 (using one option only), which is quite significant.

This example reinforces our intuitive notion that by dynamically combining these multiple leadtime options contingent on the current inventory position and outstanding orders, the system can achieve lower cost. Due to the complex interactions between these leadtime options, we could not find an analytically elegant solution. Nevertheless, this model presents an abstraction of realistic supply chain management scenarios and could potentially offer significant benefits. This has motivated our work in developing computational methods to construct suboptimal yet effective policies, reported in Section 5.

## 4.3 A Simple Dominance Relationship

As we discussed in Section 4.1, considering all leadtime options poses serious computational challenges. In this section, we establish a simple dominance relationship which, when applicable, helps reduce computational efforts by excluding certain options—the dominated ones—without compromising the solution quality.

The intuition behind this dominance relationship is quite obvious: for any option $i$, if there is another option $j$ that offers either quicker delivery under same ordering cost, or offers lower price while ensuring same delivery, or offers both lower price and quicker delivery, option $i$ will never be used in the optimal controls and therefore can be ignored. In this case, we say $i$ is dominated by $j$. The following lemma formalizes this notion.

**Lemma 1** *An option $i$ with leadtime $\lambda_i$ and ordering cost $c_i$ is dominated by another option $j$ with leadtime $\lambda_j$ and ordering cost $c_j$ when $c_i \geq c_j$ and $\lambda_i \geq \lambda_j$. There always exist an optimal control policy which does not use option $i$.*

The proof is straightforward. Consider the set of all the control policies that use option $i$, denoted by $\omega_i$. For each control $u_i \in \omega_i$, substitute option $j$ for option $i$ as follows: if $\lambda_i = \lambda_j$, simply use option $j$ whenever $i$ is used. If $\lambda_i > \lambda_j$, use option $j$ whenever $i$ is used but delay the orderings by $\lambda_i - \lambda_j$. It is clear that the resulted control policy after substitution costs equal to or less than the original control policy $u_i$ that uses option $i$. Since this is true for any arbitrary $u_i$, the lemma immediately follows.

# 5 Computing Inventory Policies with Multiple Leadtime Options

For computational purposes, we assume that demand is a discrete random variable. For problems that have small maximum leadtime from all options (say, the maximum leadtime $\tau < 4$) and do not require fine-granularity demand discretization, the optimal policy can be found either by policy iteration through linear programming or value iteration through dynamic programming [Ber95]. Both linear programming and dynamic programming require the explicit storage of the state space. Since the size of the state space grows exponentially with the maximum leadtime, neither linear programming nor dynamic programming can be used to solve large-sized problems (say, with more than 5 options). To give an example how quickly the size of the state space becomes unmanageable, consider the following scenario: Suppose that we have 5 leadtime options whose delivery leadtimes are 1, 2, 3, 4, 5, respectively. Furthermore, suppose that one-period demand is a discrete random variable that takes on values from zero up to 20. The minimum number of the states required by value iteration using dynamic programming is in the order of magnitude of $10^{7}$[4]. To maintain such a large

[4] We do not consider the complications at the boundaries of the state space when we compute this minimum number of states required. In practice, larger state space is necessary to ensure that value iteration returns reliable results.

state space and compute optimal control for each and every state repeatedly till the value function converges is not practical. Neither can policy iteration handle computation of this size.

In this section, we propose three suboptimal control policies that are easier to compute than the optimal policies. The performance of these policies is evaluated via simulation.

## 5.1 The Dynamic Switching Policy

By the dynamic switching policy, we mean the policy that uses only one leadtime option at each ordering point but does not require the same option to be used at different ordering points. This type of policy has a nature mapping in practice: "do not split orders between suppliers at each ordering point; but switch among suppliers in different time periods if needed". Using our notation, this means:

$$\sum_{i=1}^{m} z_i = z_j \quad \text{for some } j\colon\ 1 \le j \le m \tag{10}$$

In order to search for the optimal policies within the set of dynamic switching policies, we still need to explicitly represent the state space and therefore we are limited to solving small-sized problem instances. However, to compute the optimal controls (how much to order using which option) for each state for the dynamic switching policy is much easier as compared to finding the true optimal controls for each state in general due to the fact that choices are much limited. In addition to its favorable computational properties, we are interested in dynamic switching polices because of its managerial implications.

## 5.2 The Echelon Order-up-to Policy

It is clear that we cannot afford to explicitly carry around the state space in order to solve large-sized problems. One way of avoiding representing the state space is to apply the idea of *parameterize the control policies.* We consider a class of controls that can be characterized by a small number of parameters. Based on these parameters, we can easily deduce the corresponding control for any given arbitrary state. For instance, consider an one-option inventory model (See Section 4.1), an order-up-to policy is characterized by one parameter order-up-to level $S$. Given any state $x$ (the current inventory position), the control (order amount) is $(S-x)^+$. This way, the original problem of finding optimal control policies is transformed into finding the best parameters $S^*$ that minimizes the overall system cost. Note that in this case, we do not need to explicitly record and visit each state in the state space.

Note that for our general $m$-option inventory problems, the order-up-to policy or its variants have been proved nonoptimal. The simplicity of the order-up-to policy and its wide acceptance in practice, however, have rendered itself as a good candidate suboptimal policy in which we are interested. In our study, we consider the following policy fashioned after the order-up-to policy.

$$z_i = (S_i - \sum_{j=1}^{\lambda_i} x_j - \sum_{j=1}^{i-1} z_j)^+ \quad \text{for } i = 1, \dots, m \tag{11}$$

We call this the "echelon order-up-to policy" due to the similarity between this policy and the policies developed in multi-echelon inventory research [CS60]. Intuitively, we can imagine that for each leadtime option, there is an "order-up-to" level that guarantees that the sum of future arrivals and committed orders using quicker options reaches a predetermined level. The rationale behind this is similar to that behind Fukuda's policy for two adjacent (in terms of leadtimes) options. As long as we have enough inventory on hand and on order, we use slow options. The costly quick options are used to handle situations where shortage occurs.

## 5.3 The Separating Planes Policy

As we will report in Section 5.4, although the echelon order-up-to policy is easy to compute and can be used to solve large-sized problems, the resulting solution quality measured by the average per stage cost is not entirely satisfactory. To achieve lower and closer-to-optimal costs, we develop the following policy which can be viewed as a generalization of the echelon order-up-to policy.

$$z_i = (\beta_i - \sum_{j=1}^{\tau} \alpha_{ij} x_j)^+ \quad \text{for } i = 1, \dots, m \tag{12}$$

Recall that when we follow an echelon order-up-to policy, we weigh all the $x_j$ and $z_j$ equally. We analyzed a number of the optimal policies (for small-sized problems) produced by policy iteration or value iteration and observed that equally weighing $x_j$ and $z_j$ clearly violates optimality. In the meantime, to a large extent, the points in the high-dimensional space of $(x_1, x_2, \ldots, x_m, z_j)$ for $j = 1, 2, \ldots, m$ can be separated by hyperplanes. These hyperplanes are not necessarily in parallel. This motivates the development of the separating plane policy. This policy, like the echelon order-up-to policy, does not require explicit representation of the state space and therefore can be used in solving large-sized problem instances. The separating plane policy involves more parameters than the echelon order-up-to policy and therefore are more computationally intensive. Using the separating plane policy, however, results in lower average per state system cost, as demonstrated in Section 5.4.

## 5.4 Experimental Comparisons

To evaluate the effectiveness of the proposed control policies, we performed the following experiments. We generated 396 problem instances by varying the number of leadtime options available, demand profiles and cost parameters summarized below:

- **Leadtime options:** we considered the following groups of leadtime options: two leadtime options with leadtimes 0 and 2; three leadtime options with leadtimes 0, 1, and 2; three leadtime options with leadtimes 1, 2, and 3. Different combinations of the unit ordering costs for these options are tested[5]. The cost values tested are shown in the following table:

| leadtime | ordering costs tested |
|---|---|
| 0 | 10, 9, 8 |
| 1 | 8, 7, 6 |
| 2 | 6, 5, 4 |
| 3 | 4, 3 |

[5] Not all combinations are tested.

- **Demand distributions:** we considered the following two groups of discrete demand distributions. The first group includes discretized truncated normals with negative mass moved to zero and upper-tail mass moved to 10. The mean of these distributions was set to 5 and the following standard deviations were tested: .5, 1, 2, 5, 10. The second group is consisted of 10 distributions with one period demand $t$ taking on values from 0 to 3. The tested probability mass functions are given in the following table.

| | probability mass function | | | |
|---|---|---|---|---|
| | $p(t=0)$ | $p(t=1)$ | $p(t=2)$ | $p(t=3)$ |
| 1 | 0.25 | 0.25 | 0.25 | 0.25 |
| 2 | 0.8 | 0.05 | 0.05 | 0.1 |
| 3 | 0.1 | 0.05 | 0.05 | 0.8 |
| 4 | 0.1 | 0.8 | 0.05 | 0.05 |
| 5 | 0.05 | 0.05 | 0.8 | 0.1 |
| 6 | 0.4 | 0.35 | 0.05 | 0.2 |

- **Cost parameters**: the holding cost per period is always set to 1. We tested two values of the penalty cost per period: 15 and 30.

We developed the simulation testbed in the C programming language. All experiments were performed on a Sun Ultra-1 machine. Table 1 represents the summarized results. Results of the policies are stated as percentage excess over the optimal cost which was obtained using value iterations. Table 2 records the average amount of time in CPU minutes that it took to compute the policies for one problem instance. Little attempt was made to make the coding efficient, so these time figures should be viewed with caution.

From these experimental findings, we make the following observations: the dynamic switching policy is easy to compute (for small-sized problems) but the solution quality measured by the average per stage cost is not satisfactory (16.2% above the optimal cost on grand average). The echelon order-up-to policy takes a moderate computational effort to compute and yields reasonable results (11.9% above the optimal). The separating plane policy takes relatively intensive computational efforts to compute but yields very good results (5.8% above the optimal cost).

| Leadtime Options | Dynamic Switching | Echelon Order-up-to | Separating Plane |
|---|---|---|---|
| 2 modes (0, 2) | 15.3 | 11.6 | 5.4 |
| 3 modes (0, 1, 2) | 16.2 | 12.2 | 6.2 |
| 3 modes (1, 2, 3) | 18.7 | 12.1 | 6.3 |

Table 1: Average Performance of Policies Considered (% Error)

| Leadtime Options | Optimal Policy | Dynamic Switching | Echelon Order-up-to | Separating Plane |
|---|---|---|---|---|
| 2 modes (0, 2) | 12.2 | 0.3 | 0.9 | 1.9 |
| 3 modes (0, 1, 2) | 12.4 | 0.3 | 4.8 | 11.4 |
| 3 modes (1, 2, 3) | 200.0 | 3.2 | 4.9 | 12.8 |

Table 2: Average CPU time of Computing Policies Considered (in minutes)

We have also conducted experiments for problem instances of larger size (maximum leadtimes ranging from 4 to 6) using the echelon order-up-to and the separating plane policies. For these problems, optimal policies are not known since neither value iteration nor policy iteration can be applied because of the size of the problem. In addition, the dynamic switching policy cannot be used due to the size of the state space. Initial experimental results suggest that the echelon order-up-to and the separating plane policies are capable of handling large-sized problems and the separating place policy outperforms the echelon order-up-to policy by roughly the same percentage which we observed for the small-sized problems. Further investigation is needed to determine how effective these policies are, possibly through establishing lower bounds on the average cost.

# 6 Concluding Remarks

Inter-organizational electronic commerce is reshaping the way enterprises conduct business. In this paper, we present a model of supply chain that explicitly addresses time and cost issues. We have coupled this model with inventory management and performed a computational study to find effective control policies. These models and computational mechanisms are essential for software agents to take advantage of abundant choices that come with the increasingly accessible worldwide information infrastructure, and to gain competitive edge in highly dynamic business environments.

We conclude this paper by presenting some of the extensions to our agent-based supply chain model.

- Other performance measures of supply chains such as product quality, service levels, etc., are not addressed in our current model. We plan to enrich our model to address these additional measures.

- Our current model assumes that cost/leadtime information associated with each supply chain activity is known and deterministic. In practice, more often than not uncertainty will arise, especially in leadtime. Models that deal with the stochastic leadtime is highly desirable. We have proved that finding stochastic leadtime/cost efficient frontier is intractable if using the first order stochastic dominance alone. We are currently working on identifying other stochastic dominance relationships (possibly of the heuristic nature) that cut down computation.

- In this paper, we discussed how to integrate inventory decisions into the LCT model. We assumed that inventory is held at the root of the supply chain. We are currently working on integrating LCT with multi-echelon production/inventory models. Current multi-echelon inventory literature typically is not concerned with leadtime issues. To evaluate the impact of leadtime options in a multi-echelon setting is significant. The main research issues are: how to measure and evaluate the overall performance of such a complex supply chain, how to make inventory positioning decisions, how to compute efficiently multi-echelon inventory policies with leadtime options, etc.

# References


[AKS58] Kenneth J. Arrow, Samuel Karlin, and Herbert Scarf. *Studies in the Mathematical Theory of Inventory and Production.* Stanford University Press, 1958.

[Ash72] Robert B. Ash. *Real Analysis and Probability.* Probability and Mathematical Statistics. Academic press, 1972.

[BDR94] M. Ben-Daya and Abdul Raouf. Inventory models involving lead times as a decision variable. *Journal of Operational Research Society*, 45(5):579–582, 1994.

[Ber95] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control.* Athena Scientific, 1995.

[Bul64] E. Bulinskaya. Some results concerning optimum inventory policies. *Theory of Probability and its Applications*, 9(3):389–403, 1964.

[CS60] A. Clark and H. Scarf. Optimal policies for a multi-echelon inventory problem. *Management Science*, 6(4):475–490, 1960.

[Fuk64] Y. Fukuda. Optimal policies for the inventory problem with negotiable leadtime. *Management Science*, 10(4):690–708, 1964.

[HZ97] Arthur Hsu and Dajun Zeng. Finding ordering policies given multiple leadtime options. Graduate School of Industrial Administration, working paper, Carnegie Mellon University, 1997.

[Kap70] Robert S. Kaplan. A dynamic inventory model with stochastic lead times. *Management Science*, 16(7):491–507, 1970.

[LS97] J.S. Liu and K. Sycara. Coordination of multiple agents for production management. *Annals of Operations Research*, 75:235–289, 1997.

[LZ93] Hon-Shiang Lau and Long-Geng Zhao. Optimal ordering policies with two suppliers when lead times and demands are all stochastic. *European Journal of Operational Research*, 68:120–133, 1993.

[Roc70] R. Tyrrell Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[Son94] Jing-Sheng Song. The effect of leadtime uncertainty in a simple stochastic inventory model. *Management Science*, 40(5):603–613, May 1994.

[SZ96] Katia Sycara and Dajun Zeng. Coordination of multiple intelligent software agents. *International Journal of Cooperative Information Systems*, 5(2 & 3):181–211, 1996.

[WJ95] M. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995.