Optimizing Strategic Safety Stock Placement in Supply Chains

Stephen C. Graves
Sean P. Willems
Leaders for Manufacturing Program and
A. P. Sloan School of Management
Massachusetts Institute of Technology
Cambridge MA 02139-4307

January 1998, Revised August 1998

Information for Corresponding Author:

Stephen C. Graves
Leaders for Manufacturing Program and A. P. Sloan School of Management
Massachusetts Institute of Technology
Room E40-439
77 Massachusetts Ave.
Cambridge, MA 02143
617-253-6602 (phone)
617-253-1462 (fax)
sgraves@mit.edu

Optimizing Strategic Safety Stock Placement in Supply Chains

January 1998

Manufacturing managers face increasing pressure to reduce inventories across the supply chain. However, in complex supply chains it is not always obvious where to hold safety stock to minimize inventory costs and provide a high level of service to the final customer. In this paper we develop a framework for modeling safety stock in a supply chain that is subject to demand or forecast uncertainty. Key assumptions are that we can model the supply chain as a network, that each stage in the supply chain operates with a periodic–review base-stock policy, that demand is bounded and that there is a guaranteed service time between every stage and its customers. We develop an optimization algorithm for the placement of strategic safety stock for supply chains that can be modeled as spanning trees. As a partial validation of the model, we describe its successful application by product flow teams at Eastman Kodak. We discuss how the model has been used to reduce finished goods inventory, target cycle time reduction efforts and rationally size component inventories. We conclude with a list of needs to enhance the utility of the model.

1. Introduction

Manufacturing firms are subject to pressure to do everything faster, cheaper and better. Firms are expected to continue to improve customer service by increasing on-time deliveries and reducing delivery lead-times. At the same time, they must provide this service more cheaply and by utilizing fewer assets. And increasingly, firms need to do this for a global marketplace.

This pressure to improve forces companies to look at their operations from a supply-chain perspective and to seek improvements from better coordination and communication across the supply chain. A supply-chain perspective is essential to avoid some of the local sub-optimization that occurs when each step in a process operates independently with its own metrics and rewards. Using a supply chain as a focusing mechanism challenges an organization to examine cross-functional solutions to address some of the barriers that inhibit improvements.

The primary intent of this research is to develop a tactical tool to help cross-functional teams in their efforts to model and improve a supply chain. In particular we provide a framework for modeling a supply chain and develop an approach, within the framework, to optimize the inventory in a supply chain. More specifically, we provide an optimization algorithm for finding the optimal placement of safety stock in a supply chain that can be modeled as a spanning tree that is subject to uncertain demand. Key assumptions for the optimization are that each stage of the supply chain operates with a periodic-review, base-stock policy, that each stage quotes a guaranteed service time to its customers, and that demand is bounded.

Given our intent to develop a useful tactical tool for industry, we have made an enormous investment in time and energy to build a commercial-quality software application. With the software, one can build and optimize a network model of a supply chain, as described in this paper. This has been necessary in order for us to have an opportunity to test the research and validate the model as being applicable to industry. We do not discuss in the paper in any detail this software implementation of the model; however, the software is available to download from our web site, http://web.mit.edu/lfmrg3/www/. We do report in the paper on our experience applying the model to a supply chain at Kodak.

In the remainder of this section we briefly discuss related literature. In section 2, we present our framework for modeling a supply chain by describing the key assumptions. In section 3, we discuss and defend the assumptions. We introduce the model for a single stage in the supply chain in section 4; this serves as the building block for the multi-stage model described in section 5. In section 6 we develop the optimization algorithm for determining the safety stock placement in a supply chain modeled as a spanning tree. We discuss in section 7 how one can use the spanning tree algorithm to get near-optimal solutions for more general networks. We present an overview of our application experience with the model in section 8, and conclude in section 9 with thoughts on how to improve the tool.

Related Literature: There is an extensive literature on inventory models for multi-stage or multi-echelon systems with uncertain demand; much of this literature is applicable to supply chains as now defined. We refer the reader to the survey articles by Axsater (1993), Federgruen (1993), Inderfurth (1994), van Houtum et al. (1996) and Diks et al. (1996). Within this vast literature, we mention two sets of papers that are most related to our work.

First, we note the work by Simpson (1958) who determined optimal safety stocks for a supply chain modeled as a serial network. Our work is based on similar assumptions about the demand process and about the internal control policies for the supply chain. Our work is also closely related to that of Inderfurth (1991, 1993), Inderfurth and Minner(1995) and Minner (1995) who also build off of the framework proposed by Simpson for optimizing safety stocks in a supply chain. We extend the work of Simpson and of Inderfurth and Minner by treating a more general network, namely spanning trees. We also provide a different, and we believe richer, interpretation of the framework and its applicability to practice. And we provide new results in the appendix on the form of the optimal policies when we relax a constraint on the internal control policy for the supply chain.

Second, our work is closely related in intent to Lee and Billington (1993), Glasserman and Tayur (1995) and Ettl et al. (1996). Each of these papers examines the determination of the optimal base-stock levels in a multi-stage supply chain, and tries to do so in a way that is applicable to practice. Glasserman and Tayur (1995) show how to use simulation and infinitesimal perturbation analysis to find the optimal base-stock levels for capacitated multi-stage systems. Both Lee and Billington and Ettl et al. (1996) develop performance evaluation models of a multi-stage inventory system, where the key challenge is how to approximate the replenishment lead-times within the supply chain. They then formulate and solve a nonlinear optimization problem that minimizes the supply chain's inventory costs subject to user-specified customer service level requirements. Our work is similar in that we also assume base-stock policies and focus on minimizing the inventory requirements in a supply chain. The resulting models and algorithms are much different, though, due to different assumptions about the demand process and different constraints on service levels within the supply chain.

2. Assumptions

Multi-Stage Network: We model a supply chain as a network where nodes are stages in the supply chain and arcs denote that an upstream stage supplies a downstream stage. A stage represents a major processing function in the supply chain. A stage might represent the procurement of a raw material, or the production of a component, or the manufacture of a subassembly, or the assembly and test of a finished good, or the transportation of a finished product from a central distribution center to a regional warehouse.

An arc signifies that the component produced at the upstream stage is a required input to the process at the downstream stage. If a stage is connected to several upstream stages, then the production activity at the stage is an assembly requiring inputs from each of the upstream stages. A stage that is connected to multiple downstream stages is either a distribution node or a production activity that produces a common component for multiple internal customers. We can associate with each arc a scalar ϕ_{ij} to indicate how many units of the upstream component i are required per downstream unit j.

Each stage is a potential location for holding a safety-stock inventory of the item processed at the stage.

Production Lead-Times: For each stage, we assume a known deterministic production lead-time, call it T_i. When a stage reorders, the production lead-time is the time from when all of the inputs are available until production is completed and available to serve demand. The production lead-time includes both the waiting and processing time at the stage, plus any transportation time required to put the item into inventory. For instance, suppose stage k requires inputs from stage i and j, and has a three-day production lead-time; if we make a production request on stage k in time period t, then stage k completes the production at time t+3, provided that there are adequate supplies of i and j at time t.

We assume that the production lead-time is not impacted by the size of the order; hence, in effect, we assume that there are no capacity constraints that limit production at a stage.

Periodic-Review Base-Stock Replenishment Policy: We assume that all stages operate with a periodic-review policy with a common review period. Each period each stage observes demand either from an external customer or from its downstream stages, and places orders on its suppliers so as to replenish the observed demand. In effect, each stage operates with a one-for-one or base-stock replenishment policy. There is no time delay in ordering; hence, in each period the ordering policy passes the external customer demand back up the supply chain so that all stages see the customer demand.

External Demand: Without loss of generality, we assume that external demand occurs only at nodes that have no successors, which we term demand nodes or stages. For each demand node j, we assume that the end-item demand comes from a stationary process for which the average demand per time period is μ_j. We denote the demand observed at stage j in period t as $d_j(t)$. We assume that the demand process for end item j is bounded by the function $D_j(\tau)$, for $\tau = 1, 2, 3, \ldots M_j$, where M_j is the maximum replenishment time for the item[1]. That is, $D_j(\tau) \geq d_j(t-\tau+1) + d_j(t-\tau+2) + \ldots + d_j(t)$ for all t and for $\tau = 1, 2, 3, \ldots M_j$. We define $D_j(0) = 0$ and assume that $D_j(\tau)$ is

[1] The maximum replenishment time for node j is defined as $M_j = T_j + \max \{M_i \mid i:(i,j) \in A\}$.

increasing and concave on $\tau = 1, 2, 3, \ldots M_j$; thus, $D_j(\tau) - D_j(\tau\text{-}1)$ is nonnegative and decreases as τ increases.

Internal Demand: We term an internal stage to be one with internal customers or successors. For an internal stage, the demand at time t is the sum of the orders placed by the immediate successors. Since each stage orders according to a base-stock policy, the demand at internal stage i is given by:

$$d_i(t) = \sum_{(i,\, j) \in A} \phi_{ij}\, d_j(t)$$

where A is the arc set for the network representation of the supply chain. For both demand nodes and internal stages, stage j will order an amount $\phi_{ij}\, d_j(t)$ from upstream stage i, for all i that directly supply stage j ($\phi_{ij} > 0$).

We assume that the demand at each internal node of the supply chain is stationary and bounded. The average demand rate for component i is:

$$\mu_i = \sum_{(i,\, j) \in A} \phi_{ij}\, \mu_j .$$

We assume that demand for the component i is bounded by the function $D_i(\tau)$, for $\tau = 1, 2, 3, \ldots M_i$, where M_i is the maximum replenishment time for the item. This bound may be a given input or it may be derived from the demand bounds for the downstream, or customer, stages for stage i. We discuss in the next section how this might be done.

Guaranteed Service Times for End Items: We assume that each demand node j promises a guaranteed service time S_j by which the stage j will satisfy customer demand. For instance, if S_j

= 0, then the stage provides immediate service from inventory to the final customer; if $S_j > 0$, then the customer demand at time t, $d_j(t)$, must be filled by time $t + S_j$. Furthermore, we assume that stage j provides 100% service for the specified service time: stage j delivers exactly $d_j(t)$ to the customer at time $t + S_j$.

<u>Guaranteed Service Times for Internal Stages</u>: An internal stage i quotes and guarantees a service time S_{ij} for each downstream stage j, $(i, j) \in A$. Given the assumption of a base-stock policy, stage j places an order equal to $\phi_{ij}\, d_j(t)$ on stage i at time t; then stage i delivers exactly this amount to stage j at time $t + S_{ij}$. For instance, if $S_{ij} = 3$, then stage i will fulfill at time t + 3 an order placed at time t by stage j.

For the initial development of the model, we assume that stage i quotes the same service time to all of its downstream customers; that is, we assume that $S_{ij} = S_i$ for each downstream stage j, $(i, j) \in A$. We describe in Appendix II how to extend the model to permit customer-specific service times. In brief, if there is more than one downstream customer, we can insert zero-cost, zero production lead-time dummy nodes between a stage and its customers to enable the stage to quote different service times to each of its customers. The stage quotes the same service time to the dummy nodes and each dummy node is free to quote any valid service time to its customer stage.

The service times for both the end items and the internal stages are decision variables for the optimization model, as will be seen in section 5. However, as a model input, we may impose bounds on the service times for each stage. In particular, we assume that for each end item we are given a maximum service time as an input.

3. Discussion of Assumptions

The assumptions of bounded demand and of guaranteed service times are the most controversial. We need to frame the discussion of these assumptions in the context of the intent of the research. We desire to provide tactical guidance for where to position safety stock in a supply chain. In light of this, we pose the problem as one of finding the safety stock necessary to provide 100% service for both external and internal customers for a bounded demand process.

<u>Bounded Demand</u>: We do not require any assumptions about the distribution of demand. We do presume, though, that it is possible to establish a meaningful upper bound on demand over varying horizons for each end item. By meaningful, we mean in the context of setting safety stock policies: the safety stock should be set to cover all demand realizations that fall within the upper bounds. If demand were to exceed the upper bounds, then the safety stock, by design, will not be adequate. In such extraordinary cases, a manager would resort to other devices or tactics to handle the excess demand. For instance, a manager might use expediting, subcontracting, premium freight transportation, and/or overtime to accommodate the windfall of demand. In specifying the demand bounds, a manager indicates explicitly his or her preference for how demand variation should be handled -- what range is covered by safety stock and what range should be dealt with by other actions or responses.

As an example, consider a typical assumption where demand for end item j is normally distributed each period and i.i.d., with mean μ and standard deviation σ. Then, for the purposes of positioning safety stock, a manager might specify the demand bounds at the demand node by:

$$D_j(\tau) = \tau\mu + k\sigma\sqrt{\tau} \qquad (1)$$

where k is set to assure that the safety stock covers the demand variation some percentage of time. In this example, the choice of k indicates how frequently the manager is willing to resort to other tactics to cover demand variability.

In some contexts there may be natural bounds on the demand for an end item. For instance, suppose the end item is a component or subassembly for a manufacturing process whose production is limited by capacity constraints or by a frozen master schedule. An example would be a supply chain that supplies components to an automobile assembly line or an OEM subassembly to a system integrator. In these cases, bounded demand for the component corresponds to the maximum usage by the manufacturing process for the component over various time horizons.

For each internal stage we assume that we can also establish meaningful demand bounds. If internal stage i has a single successor, say stage j, then $D_i(\tau) = \phi_{ij} D_j(\tau)$ for all relevant τ. For internal stages with more than one successor, we require some judgment for deciding how to combine the demand bounds for the successors to obtain a relevant demand bound for the internal stage for the purposes of positioning the safety stock. One possibility is just to sum the demand bounds for the successors; however, this approach assumes that there is no risk pooling from combining the demand of multiple end items. An alternative approach is to assume that there will be some relative reduction in variability as we combine demand streams, i.e., some risk pooling. For instance, we might infer the demand bounds for internal stages by means of an expression like

$$D_i(\tau) = \tau \mu_i + \sqrt[p]{\sum_{(i,j)\in A} \left\{\phi_{ij} \left[D_j(\tau) - \tau \mu_j\right]\right\}^p} \qquad (2)$$

where $p \geq 1$ is a given constant. Larger values of p correspond to more risk pooling. Setting p = 1 models the case of no risk pooling. If one were to think of the end-item demand bounds analogous to (1), then combining demand bounds could be viewed as similar to combining standard deviations; indeed, this is what (2) will do when p = 2.

We do not attempt to model what happens when demand exceeds some maximal level. When demand might be regarded as being extraordinary, we assume that the operation would respond with an equally extraordinary measure, as noted above. We regard this as beyond the scope of the model, given the stated intention to provide tactical decision support. See Kimball (1988), Simpson (1958) and Graves (1988) for further discussion of this assumption.

<u>Guaranteed Service Times</u>: We assume that we can express the service at both external and internal stages by means of guaranteed service times. Furthermore, we assume perfect or 100% service; within the context of the model there are never any shortages nor any violations of the guaranteed service times. As such, we do not explicitly model a tradeoff between possible shortage costs and the costs for holding inventory. Rather, we pose the problem as being how to place safety stocks across the supply chain to provide 100% service for the assumed bounded demand with the least inventory holding cost.

In defense of these assumptions, we note that it is often very difficult in practice to assess shortage costs for an external customer. Similarly, when we have asked managers for their desired service level, more often than not the response is that there should be no stockouts for external customers. We have found that managers seem more comfortable with the notion of 100% service for some range of demand; they accept the fact that if demand exceeds this range they will have shortages unless they can somehow expand the response capability of their supply chain. The assumptions for the model presented herein are consistent with this perspective.

For an internal customer, guaranteed service times need not be optimal in terms of least inventory costs. Indeed we show in Appendix I how to relax this assumption for a serial network, and report the cost impact of this assumption for a set of 36 test problems: the safety stock holding cost is 26% higher on average, while the total inventory cost is 4% higher on average. However, guaranteed service times are very practical in contexts where there is the need to coordinate replenishments. For instance, any assembly or subassembly stage requires the concurrent availability of multiple components, not all of which might be explicitly included in the model. When we assume guaranteed service times, we make the challenge of coordinating the availability of these components much easier.

4. Single-Stage Model

In this section we present a model for the inventory at a single stage; the single-stage model serves as the building block for modeling a multi-stage supply chain.

Associated with each stage j is an inbound service time, call it SI_j. The inbound service time is the time it takes for the stage to get supplies from its immediate suppliers. In period t stage j observes demand $d_j(t)$ and places an order equal to $\phi_{ij}\, d_j(t)$ on each upstream stage i for which $\phi_{ij} > 0$. The inbound service time is the time for all of these orders to be delivered to stage j, so that stage j can then initiate the process to replenish $d_j(t)$. If stage j has a single upstream stage, say stage i, then $SI_j = S_i$. If production at stage j requires inputs from more than one upstream stage, then $SI_j = \max_{(i,j)\in A} \{S_i\}$. That is, stage j cannot commence production to replenish the demand observed in period t, $d_j(t)$, until all inputs have been received, where all inputs are available by $t + SI_j$ by definition.

The service time for stage j, S_j, is the outbound service time, namely the time allowed for stage j to satisfy demand; that is demand in period t is filled in period $t + S_j$, where we assume that stage j provides the same service time for all downstream customers.

<u>Inventory Model</u>: We define $I_j(t)$ to be the finished inventory at stage j at the end of period t, where we assume the inventory system starts at time t=0. Under the assumptions of perfect service and a base-stock replenishment policy, we can express $I_j(t)$ as

$$I_j(t) = B_j - d_j(t - SI_j - T_j,\ t - S_j) \qquad (3)$$

where $B_j = I_j(0) \geq 0$ denotes the base stock and where $d_j(a, b)$ denotes the demand at stage j over the time interval (a, b]. [see Kimball 1988, Simpson 1958 or Graves 1988] Since we assume a periodic-review replenishment policy, then without loss of generality we express all time parameters as integer units of the underlying time period. Hence, we understand $d_j(a, b)$, the demand at stage j over the time interval (a, b], to be given by

$$d_j(a, b) = d_j(a+1) + d_j(a+2) + \ldots + d_j(b)$$

for $a<b$ and $d_j(t)$ being the demand observed at stage j in time period t. When $a \geq b$, we define $d_j(a, b) = 0$. And for (3) to be true for small t, we define $d_j(a, b) = d_j(0, b)$ for $a<0$.

To explain (3), we observe that the replenishment time for the inventory at stage j is $SI_j + T_j$. Thus, in time period t stage j completes the replenishment of the demand observed in time period $t - SI_j - T_j$. Hence, at the end of time period t, the cumulative replenishment to the inventory at stage j equals $d_j(0, t - SI_j - T_j)$. For a given service time S_j, in time period t stage j fills the demand observed in time period $t - S_j$ from its inventory. By the end of time period t the cumulative shipments from the inventory at stage j equal $d_j(0, t - S_j)$. The difference between the cumulative replenishment and the cumulative shipments is the inventory shortfall, $d_j(t - SI_j - T_j, t - S_j)$. The on-hand inventory at stage j is the initial inventory or base stock minus the inventory shortfall, as given by (3).

<u>Determination of Base Stock</u>: We require that $I_j(t) \geq 0$ with probability 1 in order for the stage to provide 100% service to its customers. From (3) we see that 100% service requires that

$$B_j \geq d_j(t - SI_j - T_j, t - S_j) \qquad \text{with probability 1.}$$

Since we assume demand is bounded, we can satisfy the above requirement with the least inventory by setting the base stock as follows:

$$B_j = D_j(\tau) \qquad \text{where } \tau = \max\,[0, SI_j + T_j - S_j]. \tag{4}$$

By assumption, any smaller value for the base stock can not assure that $I_j(t) \geq 0$ with probability 1, and thus cannot guarantee 100% service.

In words, we set the base stock equal to the maximum possible demand over the *net* replenishment time for the stage. The replenishment time for stage j is the time to get the inputs (SI_j) plus the production time at stage j (T_j). The *net* replenishment time for stage j is the replenishment time minus the service time (S_j) quoted by the stage. The demand over the net replenishment time is demand that has been filled but that has not yet been replenished. The base stock must cover this time interval of exposure; thus the base stock is set to the maximum demand over this time interval.

It is possible that the promised service time is longer than the replenishment time, i.e., $SI_j + T_j < S_j$, and thus the net replenishment time is negative. For example, it may take five days for the stage to replenish its inventory, but the promised service time is eight days. In this case, we see from (4) that there is no need for a finished goods inventory; we can set the base stock B_j to zero and still provide 100% service. Indeed, in such a case, the stage would delay each order on its suppliers by S_j - SI_j - T_j periods, so that the supplies arrive when needed.

With no loss of generality, we can redefine the inbound service time so that the net replenishment time is nonnegative. In particular we redefine SI_j to be the smallest value that satisfies the following constraints:

$$SI_j \geq S_i \qquad \text{for all } (i, j) \in A \qquad \text{and}$$

$$SI_j + T_j \geq S_j \quad .$$

If the inbound service time is such that $SI_j > S_i$ for some $(i, j) \in A$, then stage j delays orders from stage i by SI_j - S_i periods.

Safety Stock Model: We use (3) and (4) to find the expected inventory level $E[I_j]$:

$$\begin{aligned} E[I_j] &= B_j - E[d_j(t - SI_j - T_j, t - S_j)] \\ &= D_j(SI_j + T_j - S_j) - (SI_j + T_j - S_j)\,\mu_j \qquad (5) \end{aligned}$$

for $SI_j + T_j - S_j \geq 0$. The expected inventory represents the safety stock held at stage j. The safety stock is a function of the net replenishment time and the bound on the demand process. As an example, suppose the demand bound is given by (1); then the safety stock is

$$E\left[I_j\right] = k\sigma\sqrt{SI_j + T_j - S_j} \; .$$

Pipeline Inventory: In addition to the safety stock, we may want to account for the in-process or pipeline stock at the stage. Following the argument for the development for equation (3), we observe that the work-in-process inventory at time t is given by

$$W_j(t) = d_j(t - SI_j - T_j, t - SI_j) .$$

That is, the work-in-process corresponds to T_j periods of demand given the assumption of a deterministic production lead-time for the stage.

We see that the expected work-in-process depends only on the lead-time at stage j and is not a function of the service times:

$$E[W_j] = T_j \mu_j .$$

Hence, in posing an optimization problem in the next section, we ignore the pipeline inventory and only model the safety stock. This is not to say that the work-in-process is not a significant part of the inventory in a supply chain. But for the purposes of this work, we assume that the lead-time of a stage, as well as the demand rate, are input parameters and thus the pipeline stock is predetermined. Nevertheless, in any application, we account for both the safety stock and the pipeline stock as both will contribute to the total supply chain inventory.

5. Multi-Stage Model

We can now use the single-stage model as a building block to model the inventory in a multi-stage system or supply chain. In particular, we just use (5) for every stage, but where the inbound service time is a function of the outbound service times for the upstream stages; to wit, the model for stage j is

$$E[I_j] = D_j(SI_j + T_j - S_j) - (SI_j + T_j - S_j)\,\mu_j \tag{6}$$

$$SI_j + T_j - S_j \geq 0 \tag{7}$$

$$SI_j - S_i \geq 0 \qquad \text{for all } (i, j) \in A \tag{8}$$

The first equation expresses the expected safety stock as a function of the net replenishment time. The second equation assures that the net replenishment time is nonnegative. The third equation constrains the inbound service time to equal or exceed the service times for the upstream stages. In (8), if stage j has no upstream supplier, then we require the inbound service time to be nonnegative.

We see from (6)-(8) that the expected inventory or safety stock in the supply chain is a function of the demand process, the production lead-times and the service times. We assume that the production lead-times, the means and bounds of the demand processes, and the maximum service times for the demand nodes are known input parameters. The service times are the decision variables. This suggests the following optimization problem **P** for finding the optimal service times:

$$\textbf{P} \qquad \min \sum_{j=1}^{N} h_j \left\{ D_j\left(SI_j + T_j - S_j\right) - \left(SI_j + T_j - S_j\right)\mu_j \right\}$$

$$\text{s. t.} \qquad S_j - SI_j \leq T_j \qquad \text{for } j = 1, 2 \ldots. N$$

$SI_j - S_i \geq 0$ for all $(i, j) \in A$

$S_j \leq s_j$ for all demand nodes j

$S_j, SI_j \geq 0$ and integer for $j = 1, 2 \ldots N$

where h_j denotes the per-unit holding cost for inventory at stage j and s_j is the maximum service time for demand node j. Thus, the objective of problem **P** is to minimize the holding cost for the safety stock in the supply chain. The constraints assure that the net replenishment time for each stage is nonnegative, that the inbound service time is at least as large as the maximum supplier service time, and that the end-item stages satisfy their service guarantee. The decision variables are the service times.

Problem **P** is a non-linear optimization problem. One can show that the objective function is a concave function, provided that the demand bound $D_j(\,)$ is a concave function for each stage j. Hence, in problem **P** we minimize a concave function over a set of linear constraints. Although the feasible region is not necessarily bounded, one can show that the optimal service times need not exceed the sum of the production lead-times, provided that the demand bound $D_j(\,)$ is a non-decreasing function for each stage j. Thus, problem **P** is to minimize a concave function over a closed, bounded convex set. An optimum for such problems is at an extreme point of the feasible region (e.g., Luenberger, 1973).

We have not been able to exploit this result to develop a general-purpose algorithm for **P** for any supply chain. However, there are algorithms for special cases, based on the network structure of the supply chain.

Simpson (1958) considered a serial-line supply chain, where he assumed that the guaranteed service time for the external customer is zero. Simpson showed that there is an optimal extreme point solution for **P** for which $S_i = 0$ or $S_i = S_{i+1} + T_i$, where stage i+1 supplies stage i (i.e., $SI_i = S_{i+1}$). Thus, there is an "all or nothing" optimal solution; either a stage has no safety stock ($S_i = S_{i+1} + T_i$) or the stage has sufficient safety stock ($S_i = 0$) to de-couple it from its downstream stage. Gallego and Zipkin (1994) provide supporting evidence that "all or nothing" policies can be near optimal in serial systems under more traditional assumptions where demand is not bounded.

Graves (1988) observed that the optimization for the serial line case is equivalent to a shortest path problem over N nodes. In a series of papers, Inderfurth (1991), (1993), Inderfurth and Minner (1995), and Minner (1995) show how to solve problem **P** by dynamic programming when the supply chain is an assembly network or a distribution network. Graves and Willems (1996) developed similar results for assembly and distribution networks. In the next section we present a dynamic programming algorithm for the more general case of a spanning tree.

6. Algorithm for Spanning Tree

We describe in this section how to solve P by dynamic programming when the underlying network for the supply chain is a spanning tree, like in the figure below.

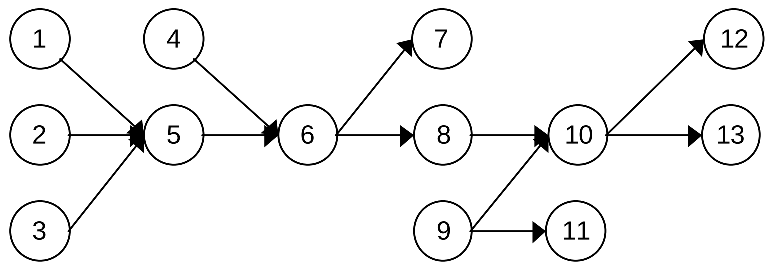

Figure 1: Spanning Tree

In the terminology of dynamic programming, we will solve **P** by decomposing the problem into N stages where there is a dynamic-programming stage for each node in the spanning tree. For a spanning tree, there is not a readily apparent ordering of the nodes by which the algorithm would proceed; indeed, we desire to sequence or number the nodes so that the algorithm is most efficient. We will enumerate the nodes in a spanning tree (and thus sequence the algorithm) so that there will be a single state variable for the dynamic programming recursion. However, the state variable for the dynamic program will be either the inbound service time at a stage or its outbound service time, where the determination depends on the topology of the network.

We first present the algorithm for numbering the nodes. Next we will present the functional equations for the dynamic programming recursions, and then state the algorithm. In section 7 we discuss solution strategies for more complex supply chains.

<u>Labeling the Nodes</u>: The algorithm for labeling or re-numbering the nodes is as follows:

1. Start with all nodes in the unlabeled set, U.

2. Set k := 1

   3. Find a node i $\in$ U such that node i is adjacent to at most one other node in U. That is, the degree of node i is 0 or 1 in the sub-graph with node set U and arc set A defined on U.

   4. Remove node i from set U and insert into the labeled set L; label node i with index k.

5. Stop if U is empty; otherwise set k:= k+1 and repeat steps 3 – 4.

For a spanning tree, it is easy to show that there will always be an unlabeled node in step 3 that is adjacent to at most one other unlabeled node. As a consequence, the algorithm will eventually label all of the nodes in N iterations. Indeed, we can show that each node labeled in the first N-1 steps is adjacent to exactly one other node in set U. That is, the nodes with labels 1, 2, … N-1 each have one adjacent node with a higher label; we define p(k) to be the node with higher label that is adjacent to node k, for k = 1, 2, … N-1. The node with label N obviously has no adjacent nodes with larger labels.

We assume in the following that the nodes in the spanning tree have been re-numbered according to this algorithm. For instance, we have used the algorithm to re-number the nodes in Figure 1 to produce Figure 2. Note that the labeling is not unique as there may be multiple choices for node i in step 3.

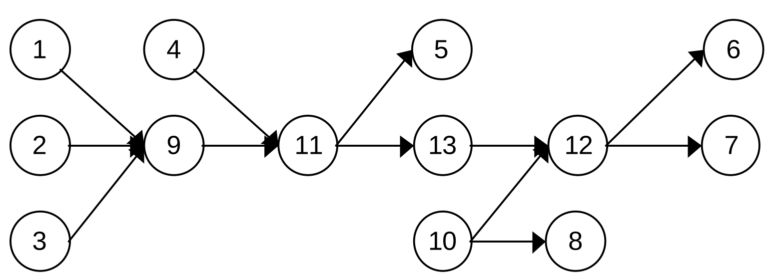

Figure 2: Renumbered Spanning Tree

For each node k we define N_k to be the subset of nodes {1, 2, ... k} that are connected to k on the sub-graph consisting of nodes {1, 2, … k}. We will use N_k to explain the dynamic programming recursion. We can determine N_k by the following equation:

$$N_k = \{k\} + \bigcup_{i<k,(i,k)\in A} N_i + \bigcup_{j<k,(k,j)\in A} N_j \ .$$

For instance, in Figure 2 N_k is {3} for k=3, {1, 2, 3, 9} for k=9, {1, 2, 3, 4, 5, 9, 11} for k=11 and {6, 7, 8, 10, 12} for k=12. We can compute N_k as part of the algorithm for re-numbering the nodes.

Dynamic Programming Recursion: We can solve the mathematical program **P** for the case of a spanning tree by dynamic programming. We first need to label or re-number the nodes as described in the previous section. Then the dynamic program evaluates a functional equation for each node, in the order of the node labels. There are two forms for the functional equation. One form determines the function $f_k(S)$, which is defined to be the minimum holding cost for safety stock in a sub-network with node set N_k, assuming that the outbound service time for stage k is S. The second form determines the function $g_k(SI)$, which is defined to be the minimum holding cost for safety stock in a sub-network with node set N_k, assuming that the inbound service time for stage k is SI.

At node k (or stage k) for $1 \le k \le N-1$, the dynamic programming algorithm will determine either $f_k(S)$ or $g_k(SI)$, depending upon the location of the node with higher label that is adjacent to k. If p(k) is downstream of node k, then the algorithm evaluates $f_k(S)$. If p(k) is upstream of node k, then the algorithm evaluates $g_k(SI)$. For node N, as will be seen, either functional equation can be evaluated.

To develop the functional equations we first define the minimum inventory holding cost for the sub-network with node set N_k as a function of both the inbound service time and the outbound service time at node k:

$$c_k(S,\ SI) = h_k\left\{D_k(SI + T_k - S) - (SI + T_k - S)\mu_k\right\} + \sum_{\substack{(i,k)\in A \\ i<k}} f_i(SI) + \sum_{\substack{(k,j)\in A \\ j<k}} g_j(S) \ .$$

The first term is the holding cost for the safety stock at node k as a function of S and SI.

The second term corresponds to the nodes in N_k that are upstream of k. For each node i that supplies node k, we include the minimum inventory holding costs for the sub-network with node set N_i, as a function of SI. The argument SI represents the inbound service time to node k, and thus, an upper bound for the outbound service time for node i. We can show that $f_i()$, the inventory holding costs for the sub-network with node set N_i, is non-increasing in the service time at node i. Hence, we equate the outbound service time at i to the inbound service time at k without loss of generality.

The third term corresponds to the nodes in N_k that are downstream of k. For each node j that is a customer of node k, we include the minimum inventory holding costs for the sub-network with node set N_j, as a function of S. The argument S represents the outbound service time for node k, and thus a lower bound for the inbound service time for node j. We can show that $g_j()$, the inventory holding costs for the sub-network with node set N_j, is non-decreasing in the inbound service time to node j, and thus, we equate the inbound service time at j to the outbound service time at k without loss of generality.

We now use the minimum inventory holding cost for the sub-network with node set N_k to develop the functional equation for $f_k(S)$:

$$f_k(S) = \min_{SI} \{ c_k(S, SI) \}$$

where the minimization is over the feasible set of inbound service times. From **P**, we see that SI $\geq \max(0, S - T_k)$. We can also bound SI by M_k-T_k, where M_k is the maximum replenishment time for node k. Thus, the minimization is subject to $\max(0, S - T_k) \leq SI \leq M_k - T_k$, and SI integer. The minimization can be done by enumeration.

The functional equation is evaluated for all possible integer outbound service times for node k, S = 0, 1, 2, … M_k.

The functional equation for $g_k(SI)$ is very similar in structure:

$$g_k(SI) = \min_{S} \{ c_k(S, SI) \}.$$

The minimization is over the feasible set of outbound service times. If k is an internal stage, then the feasible set is S = 0, 1, … SI + T_k; if k corresponds to a demand node, then the feasible set is S = 0, 1, ….s_k. The minimization can be done by enumeration.

The functional equation is evaluated for all possible integer inbound service times for node k, SI = 0, 1, 2, … M_k-T_k where M_k is the maximum replenishment time for node k.

The dynamic programming algorithm is now as follows:

1. For k:= 1 to N-1

    2. If p(k) is downstream of k, evaluate $f_k(S)$ for S = 0, 1, … M_k.

    3. If p(k) is upstream of k, evaluate $g_k(SI)$ for SI = 0, 1, … M_k-T_k.

4. For k:=N evaluate $g_k(SI)$ for SI = 0, 1, … M_k-T_k.

5. Minimize $g_N(SI)$ for SI = 0, 1, … M_N-T_N to obtain the optimal objective function value.

This procedure finds the optimal objective function value; to find an optimal set of service times entails the standard backtracking procedure for a dynamic program.

To summarize, at each stage of the dynamic program, we find the minimum inventory holding costs for the sub-network with node set N_k, as a function of a state variable. The state variable depends on the direction of the arc that connects the sub-network N_k to the rest of the network. When the connecting arc originates in N_k, then the state variable is the outbound service time (step 2); otherwise, the state variable is the inbound service time (step 3). We number the nodes so that the functions required to evaluate either $f_k(S)$ or $g_k(SI)$ have been determined prior to stage k in the dynamic program. At stage N (step 4), we determine the inventory costs for the entire network as a function of the inbound service time to node N. At step 5, we optimize over the inbound service time to find the optimal inventory cost.

The computational complexity of the algorithm is of order NM^2 where M is the maximum service time, which is bounded by the sum of the production lead-times $\sum_{j=1}^{N} T_j$. We have implemented the algorithm for a PC in the C++ programming language. The run times for real problems with 25 to 30 nodes are effectively instantaneous on a Pentium PC with a 100 megahertz Intel processor.

7. Extension to General Networks

We present in this section one possible approach to solving **P** for general networks. We describe how one can develop a lower bound to **P**, as well as a good feasible solution, by application of the spanning-tree algorithm.

We develop a lower bound by solving a Lagrangian relaxation to **P**. Suppose we have a general supply chain that we model as a connected network with N nodes and with N+K arcs. Suppose we select K+1 arcs from the arc set A such that the remaining N-1 arcs form a spanning tree; let A* denote the set of K+1 arcs. In **P**, for each arc (i, j) ∈ A we have a constraint of the form

$$SI_j - S_i \geq 0,$$

which relates the outbound service time at node i to the inbound service time at node j. We define a nonnegative Lagrange multiplier λ_{ij} for each constraint corresponding to the arca (i, j) ∈ A* , and then use the multiplier to remove the constraint and bring it to the objective function. We then have a relaxed version of **P**:

$$\textbf{P}\,(\lambda) \quad \min \sum_{j=1}^{N} h_j \left\{ D_j\left(SI_j + T_j - S_j\right) - \left(SI_j + T_j - S_j\right)\mu_j \right\} - \sum_{(i,j)\in A^*} \lambda_{ij}\left(SI_j - S_i\right)$$

s. t.	$S_j - SI_j \leq T_j$	for j= 1, 2 …. N
	$SI_j - S_i \geq 0$	for all (i, j) ∈ A – A*
	$S_j \leq s_j$	for all demand nodes j
	$S_j, SI_j \geq 0$ and integer	for j = 1, 2 ….N

The relaxed problem **P (λ)** can be solved with the spanning-tree algorithm, and yields a lower bound on **P**. We can improve the lower bound by solving the dual problem:

P(D) max **P(λ)**

s. t. $\lambda_{ij} \geq 0$ for (i, j) $\in$ A* .

We can solve the dual problem by a subgradient optimization or a dual ascent procedure.

To find an upper bound to **P**, we generate a feasible solution to **P**. One approach is again to focus on the K+1 constraints defined by the arc set A*. For each arc (i, j)$\in$A*, we somehow specify a nonnegative target, call it τ_{ij}. We then require for (i, j)$\in$A*, that SI_j equals or exceeds the target, and the target equals or exceeds S_i. We now replace the constraints in **P** for the arc set A* with this requirement, and define the new problem:

P(τ)

$$\min \sum_{j=1}^{N} h_j \left\{ D_j\left(SI_j + T_j - S_j\right) - \left(SI_j + T_j - S_j\right)\mu_j \right\}$$

s. t. $S_j - SI_j \leq T_j$	for j= 1, 2 …. N
$SI_j - S_i \geq 0$	for all (i, j) $\in$ A - A*
$SI_j \geq \tau_{ij} \geq S_i$	for all (i, j) $\in$ A*
$S_j \leq s_j$	for all demand nodes j
S_j, $SI_j \geq 0$ and integer	for j = 1, 2 ….N

We can solve **P(τ)** by the spanning-tree algorithm, with a slight modification to permit simple bounds on the service times. Since any feasible solution to **P(τ)** is also feasible to **P**, solving

P(τ) provides a heuristic solution to **P**, and an upper bound on its objective value. Clearly the quality of the heuristic depends on the choice of set A* and the targets for each arc in A*.

We cannot report on any systematic study of these approaches and leave that for further research. Nevertheless, from some limited exploratory work, we expect that these approaches will work reasonably well when K is small, say between 0 and 5 for a twenty-node network (N=20). Indeed, when K is small, we have been able to craft by hand optimal or near-optimal solutions by using the spanning-tree algorithm to iterate between relaxed and over-constrained versions of the problem. As K gets larger, though, this will not be possible. The challenge then will be to develop systematic procedures for selecting the arc set A*, for improving the lower bound by solution of the dual problem, and for improving the upper bound by setting the targets.

8. Application

This section presents an application of the model at the Eastman Kodak Company. Starting in 1995, Kodak has applied the model to eleven finished products across two of its assembly sites within its equipment division. We first present the model's application to a specific product flow. Then we provide a summary of Kodak's financial results to date from using the model.

Product Background: Kodak has applied the model to the internal supply chain for a high-end digital camera[2]. The key subassemblies for the digital camera are a traditional 35mm camera, an imager and a circuit-board assembly. The 35mm camera is procured from an outside vendor. The imager (a charge-coupled device) and the circuit-board assembly are produced internally. The 35mm camera supplies the lens, shutter and focus functions for the digital camera. The imager captures and digitizes the picture, and the circuit-board assembly processes and stores the image. To produce the digital camera, the back of the 35mm camera is removed and replaced with a housing containing the imager and circuit board. The camera is then tested to make sure that there are no defects in the imager. Once the camera passes the quality tests, the product is shipped to the distribution center. From the distribution center, the camera is shipped to the final customers, which for our purposes are high-end photography shops and computer superstores.

In Figure 3, we provide a high-level depiction of this supply chain. In addition to the three key subassemblies, we include the remaining parts in order to accurately represent the product's cost structure; since there are nearly one hundred additional parts in a camera, modeling them in any level of detail would greatly expand the size of the model. Hence, we

[2] The data presented in this section has been altered to protect proprietary information. However, the resulting qualitative relationships and insights drawn from this example are the same as would be from using the actual data.

group these parts into two aggregate stages of the supply chain, where one stage represents all of the parts with long procurement lead-times (greater than 60 days) and the other stage represents the short lead-time parts (less than 60 days).

We also aggregate certain operations. As seen in Figure 3, we combine the build operation for a camera with the test operation and the packing operation. The imager stage and circuit board stage are also aggregates as each represents the flow through a separate department. In the case of the circuit-board stage, this entails circuit board assembly and test. The imager stage consists of a semiconductor operation to produce wafers of imagers, followed by packaging and testing of the semiconductors, followed by an assembly operation. As we will discuss later, we subsequently expanded the model to capture some of these details.

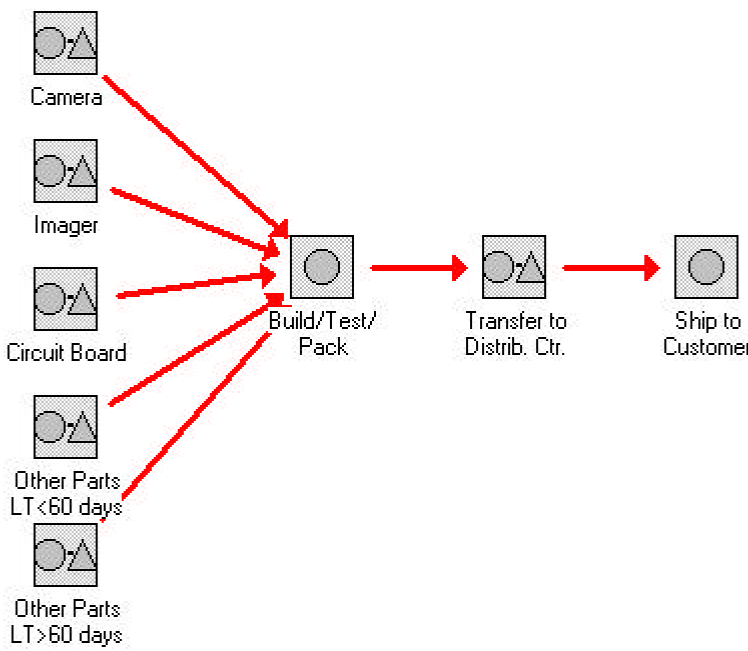

Figure 3: Implemented Safety Stock Policy for Digital Camera. All stages have a circle that denotes the processing activity at the stage. A triangle denotes that the stage holds a safety stock of its finished goods.

<u>Implementation Approach</u>: The product's supply chain crosses several functional boundaries within Kodak. Functional areas like circuit-board assembly and imager assembly are separate

departments and act as suppliers to an assembly group that performs final assembly and test. Distribution is a separate organization and owns the product once it leaves the final assembly area. To improve coordination across the departments, the equipment division at Kodak has set up product flow teams with the general charge to optimize their supply chains.

The product flow team for the digital camera relied on the model to identify opportunities for better coordination and improved asset utilization. The implementation strategy was to start simple and get experience with the model; once there was some evidence of the utility of the model, the team extended the application in increments to capture more and more of the supply chain.

The team implemented the model in phases. In the first phase, the goal was to optimize the safety stock placement in the final assembly area. In the second phase, a key internal supplier was integrated into the model. The third phase involved refining the coordination between the assembly group and the internal supplier, and incorporating other internal and external suppliers into the modeling effort.

Phase One: When the project first began, the goal was to optimize the safety stock levels for the stages that were under the direct control of the final assembly area. The decision to start with the final assembly area was based on the product's high material cost and its relatively simple supply chain structure, as described above. The cost and production lead-time for each stage are (the numbers have been disguised, but are illustrative):

Stage Name	**Production Lead-Time**	**Cost Added**
Camera	60	750
Imager	60	950
Circuit Board	40	650
Other Parts LT<60 days	60	150
Other Parts LT>60 days	150	200

Build/Test/Pack	6	250
Transfer to DC	2	50
Ship to Customer	3	0

Table 1 : Phase One Digital Camera Information

The demand bound was estimated by equation (1) where $\mu = 11$, $\sigma = 7$ and $k = 1.645$. From looking at historical demand and future demand estimates, Kodak felt that this function realistically captured the range of demand for which they wanted to use safety stock.

This characterization of demand excluded large one-time orders that come from the government and some large corporations. These orders are typically for 200-300 units with delivery scheduled less than a month from the time when the order is placed. However, since there is some advanced warning about these orders and they are independent of the other demand for the product, we developed a separate anticipatory stock policy to deal with these large, infrequent orders.

Marketing determined that the maximum service time to the final customer is five days.

Finally, the assembly group imposed the constraint that a safety stock of imagers must be held on site at final assembly. Therefore, we set the service time for the imager stage to be zero; the effect of this constraint increased the total safety stock cost by 8.7%

In the optimal solution the subassembly stages, the aggregate parts stages and the build/test/package stages hold safety stocks and quote zero service times. The ship to distribution and ship to customer stages each quote their maximum feasible service times, two and five days, respectively. The annual holding cost for the safety stock is $78,000. Thus, the optimal solution holds an inventory of components, subassemblies and completed cameras at the manufacturing site, but holds no inventory in the distribution center. In effect, the distribution center would act only as a processing center to receive orders and then to immediately ship out cameras once it received them. This is feasible since the maximum service time to the customer is five days, and it is possible to get the product from the assembly area through the distribution center and to the final customer within this five-day window.

The product flow team decided to explore some near-optimal solutions because they felt that there were some additional organizational constraints not captured in the model; in particular, distribution would want to hold safety stock on-site. To ameliorate the situation, the team suggested that both manufacturing and distribution hold safety stock and quote zero service times. However, the model showed that the cost for the safety stock would increase to $89,000. The team also investigated a policy in which the distribution center holds safety stock but the manufacturing site does not. The safety stock cost for this policy was $81,000, which was deemed to be acceptable as it was quite close to the unconstrained optimum and satisfied distribution's desire to hold inventory. This policy, as shown in Figure 3, was implemented at the end of phase one of the application.

<u>Phase Two</u>: After the initial phase of the project was completed, the product flow team expanded the model to incorporate the internal supply chain for the imager. The resulting supply chain is shown in Figure 4:

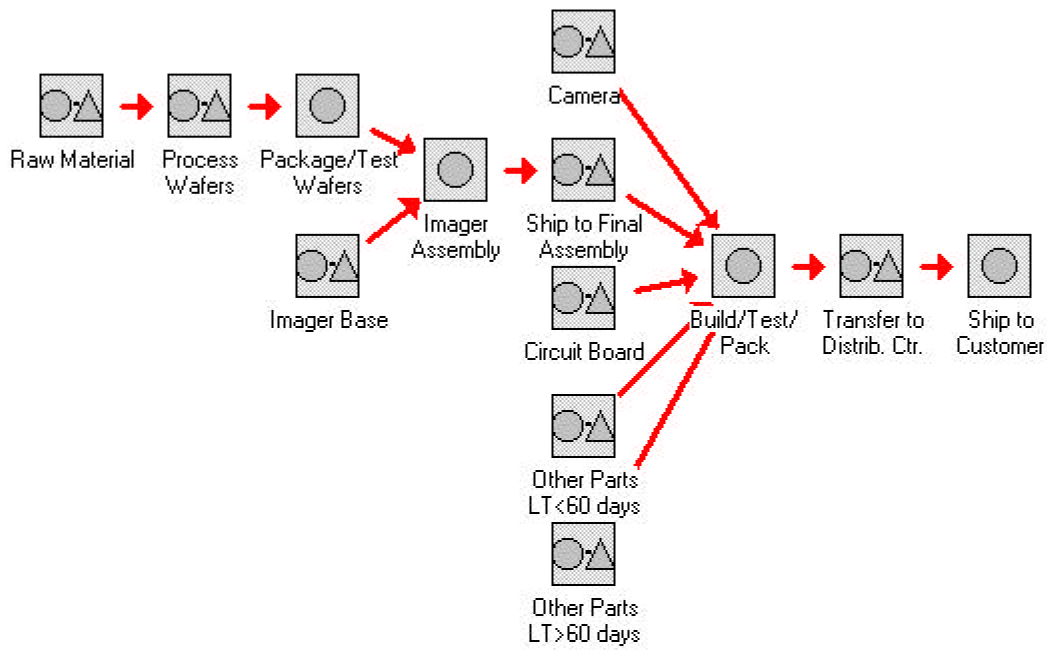

Figure 4 : Digital Camera Supply Chain

Prior to this study, safety stocks of (in-process) imagers had been held at each stage of the supply chain. By application of the model, the product flow team decided to remove safety stocks from two stages in the supply chain for the imagers, as shown in the figure. This required some increase in the downstream safety stocks of finished imagers, but overall the amount of imagers held in the supply chain as safety stock (measured in terms of finished imagers) was more than halved.

Now that the model has been successfully piloted with an internal supplier, the product flow team is in the process of extending the model to incorporate other key internal and external suppliers.

<u>Financial Results</u>: Table 2 contains the financial summary for two assembly sites that use the model. Site A has applied the model to each of its eight products and Site B has applied the model to each of its three product families. The sales volume has remained relatively constant over the three years.

Assembly Site A	Y/E 95	Y/E 96	Y/E 97
Worldwide FGI	$6.7m	$3.3m	$3.6m
Raw Material & WIP	$5.7m	$5.6m	$2.9m
Delivery Performance	80%	94%	97%
Manufacturing Operation	MTS	RTO	RTO
Assembly Site B			
Worldwide FGI	$4.0m	$4.0m	$3.2m
Raw Material & WIP	$4.5m	$1.6m	$2.5m
Delivery Performance	Unavailable	78%	94%
Manufacturing Operation	MTS	RTO	RTO

Table 2: Financial Summary for Assembly Sites A and B.

At the start of 1996, the sites moved from a make-to-schedule (MTS) to a replenish-to-order (RTO) system. The modeling effort began at the end of 1995 and was used to help guide the transition to replenish-to-order. The increase in worldwide finished goods inventory for 1997 is due to a marketing promotion that was underway in Europe. By our estimate, this promotion has increased the finished goods inventory by as much as $.5 million. In the first year of the project, the emphasis was on reducing the areas directly under the control of final assembly. Over the past year, the effort has been on reducing the raw material costs and WIP in the manufacturing supply chain. The total value of the inventory for these products has been reduced by over one third over the two years.

Beyond the immediate use to guide inventory decisions, Kodak's product flow teams have also used the model for a variety of other purposes. Some products have tens of components with long procurement lead-times. The model has helped to prioritize the suppliers with whom to work to reduce these lead-times. The teams have used the model to determine the cost effectiveness of lead-time reduction efforts in manufacturing. One can compare the investment required to reduce a lead-time versus the cost savings from the reductions in pipeline and safety stock cost. Finally, manufacturing and marketing personnel have used the model to help quantify the cost of quoting a specific maximum service time to the final customer. With the model, the supply-chain team can accurately estimate the costs of a one-day, one-week or two-week guaranteed service time to the customer, and weigh the costs of the policy against the marketing benefits of the policy.

Another benefit of the model is that it provides a common, objective framework with which a cross-functional supply-chain team can work. In particular, we note that it provides a standard terminology and set of assumptions for these teams to use as they work together to improve or optimize a supply chain. As such the model has been a very effective communication vehicle or platform.

9. Conclusion

In this paper we introduce and develop a model for positioning safety stock in a supply chain. We model the supply chain as a network, where the nodes of the network are the stages of a supply chain. We assume that each stage uses a base-stock policy to control its inventory. We also assume that each stage quotes a service time to its customers, both internal and external, and that each stage provides 100% service for these quoted service times. Finally we assume that external customer demand is bounded.

We show how to evaluate the inventory requirements at each stage as a function of the service times. For supply chains that can be modeled as spanning trees, we develop an optimization algorithm for finding the service times that minimize the holding cost for the safety stock in the supply chain. We describe how this optimization might extend to more complex networks but have not explored this in any depth.

As a form of validation, we describe an application of the model at Kodak to an internal supply chain for a digital camera. This application helped Kodak to re-position its inventories in this supply chain so as to reduce its inventory and increase its service performance. In particular, Kodak realized the benefit from creating a few strategic locations to hold safety stocks, rather than spreading the safety stock across the entire supply chain. We have also applied the model to a number of other related products at Kodak and at two other companies (Black 1998, Coughlin 1998 and Felch 1997).

As with any research undertaking, we end with a number of unresolved issues, as well as new questions. We discuss these in the relative order of importance, based on our experience in applying the research to date.

Stochastic Lead-times: We assume that associated with each stage is a known, deterministic lead-time. In practice, this is often not true. Indeed, procurement times for some components are often highly uncertain. It will be of value to capture this in the model. We know how to extend the model in an approximate way for stages that procure raw materials or components from an outside vendor. In effect, for such a stage we just need to build an approximation for the inventory requirements at the stage as a function of the outbound service time quoted by the stage and the stochastic procurement time. But it is less clear how to extend the model, either exactly or approximately, to permit stochastic lead-times at stages whose function is not procurement.

Non-stationary Demand: We assume that the demand processes for end items are stationary. Yet virtually all of the products with which we have worked have short life times, over which demand is never really stationary. In practice, one runs the model under various (stationary) scenarios to see how sensitive the safety stock is to the demand characterization (Coughlin 1998). Fortunately, we have found empirically that where the model locates safety stock in the supply chain is fairly insensitive to the demand. The size of the safety stock, though, does depend directly on the demand characterization. We currently are conducting research to understand better these observations, and then to use them to extend the model to treat non-stationary demand.

Different Review Periods: We assume that each stage operates with a periodic-review, base-stock policy with a common review period. In many supply chains different stages will operate with different reorder frequencies. That is, whereas one stage may place replenishment orders on a daily basis, another stage may do this weekly. In other cases, a stage may operate with a continuous-review policy so that the time between orders varies. We can extend the model to evaluate nested periodic-review base-stock polices, in which whenever one stage reorders, all stages downstream also reorder. That is, the review period for an upstream stage is an integer multiple of the review period of its immediate customers. However, we have not yet built the software to implement this extension, as it will be a major programming undertaking and it may only be a partial fix to the issue.

Capacity Constraints: In the model we ignore capacity constraints. For certain stages in a supply chain, the consideration of a capacity limit may be necessary in order to get a credible model for determining safety stock requirements. At this time, we do not have good ideas for how to add this to the model.

General Networks: We have developed and implemented an optimization algorithm for supply chains that can be modeled as spanning trees. We have described earlier how this algorithm might extend to more general networks. But more research is needed to test and refine these ideas as well as to uncover better approaches.

Appendix I

In this appendix we examine the question as to how costly is our assumption that each internal stage quotes a guaranteed service time for each of its customers. To get some insight into this issue, we consider a serial system, for which we can determine the optimal policy when we relax the assumption of guaranteed service times for internal customers. We then compare the inventory holding costs for the optimal policies with and without this assumption for a small set of test problems.

Consider a serial supply chain with N stages where stage 1 is the demand node and stage i supplies stage i –1 for i = 2, … N. The same assumptions hold as in the original model, except that we do not require that each internal stage guarantees a fixed service time to its customer. There are no restrictions on the service level that stage i provides to its customer, stage i –1 for i = 2, … N; rather, these internal service levels will depend on the base stocks, which will be chosen to minimize the inventory holding costs for the entire supply chain. We do assume that stage 1 provides a 100% service level to the external customer; and, without loss of generality, we assume that the service time quoted to the external customer is zero.

For ease of presentation, we assume that one unit of end-item demand translates into one unit of demand at each of the internal stages; that is, $\phi_{i,i-1} = 1$ for i = 2, … N. We let d(t) denote the end-item demand in period t, d(a, b) denote the end-item demand over the time interval (a, b], and $D(\tau)$ denote the maximum possible end-item demand over a time interval of τ periods.

For each stage i, we define B_i to be the base stock, $I_i(t)$ to be the on hand inventory at time t, and $Q_i(t)$ to be the shortfall or backlog at time t. The backlog at a stage is the amount that has been ordered by the stage's customer but not yet delivered. We assume at t=0, $I_i(t) = B_i \ge 0$ and $Q_i(t) = 0$ for all stages.

We can show for i = 1, 2, … N that the inventory on-hand and backlog at time t are given by:

$$\begin{aligned} I_i(t) &= \left[B_i - d(t - T_i, t) - Q_{i+1}(t - T_i)\right]^+ \\ Q_i(t) &= \left[d(t - T_i, t) + Q_{i+1}(t - T_i) - B_i\right]^+ \end{aligned} , \qquad \text{(A1)}$$

where $[x]^+ = \max(0, x)$, T_i is the production lead-time for stage i, and $Q_{N+1}(t) = 0$ by definition. The argument to show (A1) requires that each stage has a deterministic lead-time and that each stage follows a base-stock policy in which each period each stage observes end-item demand and places a replenishment order for this amount. The essence of the argument is to note that the net inventory on hand at a stage equals the stage's base stock minus the inventory on order. For stage i, the inventory on order at time t is the backlog as of time $t-T_i$, plus all of the demand over the interval $(t- T_i, t]$.

From (A1) we can show by induction that for i = 1, 2, … N

$$\begin{aligned} Q_i(t) = \max[0,\ & d(t - T_i, t) - B_i,\ d(t - T_i - T_{i+1}, t) - B_i - B_{i+1}, \ldots \\ & \ldots\ d(t - T_i - T_{i+1} - \ldots - T_N, t) - B_i - B_{i+1} - \ldots - B_N] \end{aligned} . \qquad \text{(A2)}$$

In order for the supply chain to provide 100% service to the external customer, we must select base stocks so that $Q_1(t) = 0$ for all t. That is, we will never have a backlog at stage 1. From (A2) we see that $Q_1(t) = 0$ is assured if the base stocks satisfy the following constraints:

$$B_1 + B_2 + \ldots + B_i \ge D(T_1 + T_2 + \ldots + T_i) \qquad \text{for } i = 1, 2, \ldots N . \qquad \text{(A3)}$$

Thus, if the base stocks satisfy (A3), then all of the terms on the right-hand-side of (A2) are guaranteed to be non-positive; there will never be a shortfall at stage 1 and end-item demand will be satisfied with 100% service. We can also see that if it is possible for the demand bounds to be realized, then the constraint set (A3) provides not just sufficient but also necessary conditions for assuring 100% service for end-item demand.

We now wish to select the base stocks to satisfy (A3) and to minimize the inventory holding costs for the supply chain. To develop an expression for the inventory holding costs, we note from (A1) that the net inventory on hand for i = 1, 2, … N is given by:

$$I_i(t) - Q_i(t) = B_i - d(t - T_i, t) - Q_{i+1}(t - T_i). \tag{A4}$$

From (A4) we can write the inventory holding costs for the supply chain as:

$$\sum_{i=1}^{N} h_i E[I_i(t)] = \sum_{i=1}^{N} h_i \{B_i - \mu T_i + E[Q_i(t)] - E[Q_{i+1}(t - T_i)]\} \tag{A5}$$

where h_i is the holding cost at stage i, μ is the expected demand rate, and E[] denotes expectation.

We now can pose an optimization problem to select the base stocks, namely we minimize (A5) subject to (A3) and non-negativity constraints on the base stocks. After dropping constant terms in the objective and noting that $Q_1(t) = 0$ for any feasible solution, we can write the optimization as

P*

$$\begin{aligned}
&\min \sum_{i=1}^{N} h_i B_i - \sum_{i=2}^{N} e_{i-1} E[Q_i] \\
&\text{s.t.} \\
&B_1 + B_2 + \ldots + B_i \geq D(T_1 + T_2 + \ldots + T_i) \quad \text{for } i = 1, 2, \ldots N \\
&B_i \geq 0 \quad \text{for } i = 1, 2, \ldots N
\end{aligned}$$

where $e_i = h_i - h_{i+1}$ is the echelon holding cost. We note from (A2) that $E[Q_i]$ is a non-linear function of $B_i, \dots B_N$ for $i = 1, 2 \dots N$.

Our main result is that there is an optimal solution to **P*** in which all of the constraints in (A3) are binding. More formally we state the following:

Result: If the echelon holding costs are nonnegative and if D( ) is a non-decreasing function, then an optimal solution to **P*** is given by

$$B_1 = D(T_1)$$

$$B_i = D(T_1 + \dots + T_i) - D(T_1 + \dots + T_{i-1}) \qquad \text{for } i = 2, \dots N. \qquad \text{(A6)}$$

Proof: We note that the solution given by (A6) is nonnegative and satisfies the constraints in (A3) as equalities; thus it is a feasible solution to **P***. To prove that this is also an optimal solution, we will argue that there must be an optimal solution in which the constraints in (A3) are binding.

Suppose we have a solution $B_1^*, \dots B_N^*$ such that (A3) holds as a strict inequality for one or more constraints. Suppose the k^{th} constraint is the first constraint that is not binding and that k<N; we will treat the case when k=N later. Thus, we assume

$$B_1^* + B_2^* + \dots + B_i^* = D\left(T_1 + T_2 + \dots + T_i\right) \text{ for } i = 1, 2, \dots k-1 \text{ and}$$
$$B_1^* + B_2^* + \dots + B_k^* > D\left(T_1 + T_2 + \dots + T_k\right).$$

We now define a new solution $B_1^{**}, \dots B_N^{**}$ in which the k^{th} constraint is satisfied as an equality, and show that its objective value is no worse than that for $B_1^*, \dots B_N^*$:

$$B_i^{**} = B_i^* \text{ for } i = 1, \dots N \text{ and } i \neq k, k+1$$

$B_k^{**} = B_k^{*} - \Delta$

$B_{k+1}^{**} = B_{k+1}^{*} + \Delta$

where $\Delta = B_k^{*} - D\left(\sum_{i=1}^{k} T_i\right) + D\left(\sum_{i=1}^{k-1} T_i\right)$

We first observe that $\Delta>0$ due to the supposition that the solution $B_1^{*}, \ldots B_N^{*}$ satisfies the k^{th} constraint in (A3) as a strict inequality. Thus, we have $B_{k+1}^{**} \geq 0$. We also see that $B_k^{**} \geq 0$ since D( ) is non-decreasing. Hence the new solution $B_1^{**}, \ldots B_N^{**}$ is nonnegative. By construction, the new solution satisfies the k^{th} constraint as an equality, and there are no changes in the remaining constraints. Thus, the new solution $B_1^{**}, \ldots B_N^{**}$ is a feasible solution.

To see the change in objective function for the new solution, we will decompose it into two parts. The change to the first part of the objective function is seen to be

$$\sum_{i=1}^{N} h_i B_i^{**} = \left(-h_k + h_{k+1}\right)\Delta + \sum_{i=1}^{N} h_i B_i^{*} = -e_k \Delta + \sum_{i=1}^{N} h_i B_i^{*}. \qquad \text{(A7)}$$

For the second part of the objective function, let $E[Q_i]^{*}$ and $E[Q_i]^{**}$ denote the expected backlog at stage i for the first and second solution. Then we find from (A2) that

$E[Q_i]^{**} = E[Q_i]^{*}$ for $i>k+1$,

$E[Q_i]^{*} \leq E[Q_i]^{**} \leq E[Q_i]^{*} + \Delta$ for $i<k+1$, and

$E[Q_{k+1}]^{*} \geq E[Q_{k+1}]^{**} \geq E[Q_{k+1}]^{*} - \Delta$.

Thus, for nonnegative echelon holding costs, we can bound the change to the second part of the objective function as follows:

$$-\sum_{i=2}^{N} e_{i-1} E\left[Q_i\right]^{**} \le -\sum_{i=2}^{N} e_{i-1} E\left[Q_i\right]^{*} + e_k \Delta \ . \qquad \text{(A8)}$$

By combining (A7) and (A8), we see that the objective function for the second solution is no greater than the objective for the first. Thus, we have found a new solution in which the first k constraints in (A3) are binding and whose objective value is no worse than that for the first solution. This argument can be continued in this fashion to construct a solution in which the first N-1 constraints in (A3) are binding and whose objective value is no worse than that for the solution B_1^*, … B_N^*. The argument for the case when k=N is similar in structure but easier; we just have to reduce the base stock for stage N until the N^{th} constraint is binding, which can be done with no penalty to the objective function.

Hence, there is a feasible solution that satisfies all of the constraints in (A3) as equalities and that has an objective value no higher than that for the solution B_1^*, … B_N^*. Furthermore, this new solution must be given by (A6), as it is easy to see that it is the unique binding solution to (A3). Finally we conclude that (A6) must be an optimal solution, as its objective value equals or is less than that for any interior solution B_1^*, … B_N^*. This completes the proof.

We note that the optimal base-stock policy can be determined directly from the demand bound, and does not depend at all on the holding costs. All we need to know is that the holding costs do not decrease as we move down the supply chain, closer to the customer. We also note that this result generalizes to assembly systems by means of the transformation given by Rosling (1989); namely we can transform an assembly system into an equivalent serial system, and the result applies.

We use this result to compare the performance of the base stock policies with and without the assumption of guaranteed service times for internal customers. The test problems were all for a 3-stage serial system; the problems differed according to their demand process, their production lead-times and their holding costs.

For the demand process, we start with a Poisson demand distribution with mean λ and with a specified percentile α to truncate the demand. For each time window of length τ, we set the demand bound $D(\tau)$ as the smallest integer such that the cumulative probability for the Poisson random variable with mean $\lambda\tau$ exceeds α. We then normalize the demand distribution over the truncated range. We consider four possible demand processes: $\lambda=10$, $\alpha=0.90$; $\lambda=10$, $\alpha=0.98$; $\lambda=50$, $\alpha=0.90$; $\lambda=50$, $\alpha=0.98$.

We permit three settings for the production lead-times and three settings for the holding costs, as follows:

$(T_1, T_2, T_3) =$ (4, 4, 4); (1, 3, 8); (8, 3, 1).

$(h_1, h_2, h_3) =$ (1, 0.5, 0.2); (1, 0.66, 0.33); (1, 0.8, 0.5).

By evaluating all combinations we have a total of 36 test problems. For each test problem we determine the optimal policy for the model with guaranteed internal service times and the optimal policy (given by the result above) for the model without this requirement. For each instance, we evaluate the base stocks, the safety-stock holding cost and the total inventory holding cost. The safety stock holding cost is given by the objective function of **P** for the model with guaranteed internal service times and by (A5) for the model without this requirement. The total inventory holding cost is the sum of the safety-stock holding cost plus the pipeline-stock holding cost. The expected pipeline stock at stage i is μT_i, for μ being the mean demand rate; we assume that the holding cost for the pipeline stock at stage i is $(h_i + h_{i+1})/2$.

For the 36 test problems we find that the safety-stock holding cost for the model with guaranteed internal service times is on average 26% higher than that for the model without this requirement; the range is between 7% and 43%. The size of the gap is insensitive to the choice of demand process. However, the gap becomes larger as the production lead-time at stage 1 increases and as the echelon holding cost at stage 1 increases.

The impact on the total inventory holding cost is less dramatic. The difference in holding costs is 4% on average, with a range from less than 1% to 14%. The gap increases as the holding cost of the pipeline stock decreases, namely as the production lead-time at stage 1 decreases and as the demand rate decreases.

From the limited computational study we see that there can be a significant increase in safety stock due to the assumption of guaranteed internal service times. Relative to the total inventory, this increase does not look as bad. Nevertheless, there is a cost in terms of higher inventories from the requirement of guaranteed internal service times. This cost needs to be considered in light of the practical benefits, as discussed in the body of the paper, from imposing this requirement. Based on our observations from industrial projects, this requirement, and the resulting increase in safety stock, has not been an issue as the assumption of guaranteed internal service times is already engrained in practice.

Appendix II

In this appendix, we show by example how to transform the network model of a supply chain to permit a stage to quote different service times to its downstream customers. We do this by augmenting the network with dummy nodes, where the dummy nodes are added so that in the modified network each stage will still quote the same service time to its downstream stages. But the effective service times to the original downstream stages can now differ.

Consider the simple three-stage network where stage 1 supplies both stage 2 and 3 as shown in the figure below:

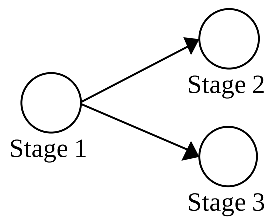

Figure A1: Original Network Formulation

Suppose we wish to allow stage 1 to quote different service times to stage 2 and 3. To do this, we modify the network by inserting dummy nodes between stage 1 and each of the downstream stages, as shown in the next figure.

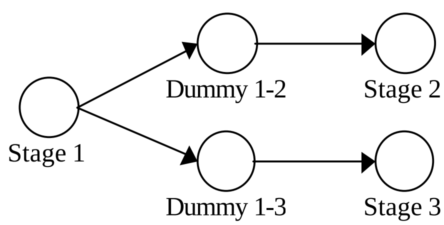

Figure A2: Modified Network Formulation

The production lead-time at each dummy node is zero. We assume that stage 1 quotes the same service time to each of the dummy nodes. But the service times quoted by the dummy nodes can differ. The effective service time from stage 1 to stage 2 is the service time from “dummy 1-2” to stage 2. Hence, in this way, we can allow the service time to stage 2 to differ from that to stage 3. But for the purposes of the spanning-tree algorithm, we still have a single service time being quoted by each node, albeit a greater number of nodes.

The cost for holding inventory at each dummy node is the same as at stage 1. Indeed, safety stock held at a dummy stage represents inventory held at stage 1. This inventory, though, is now dedicated to the dummy stage's downstream customer. Thus, in this example there are now three non-exclusive options for holding safety stock at stage 1. Safety stock at the "stage1" node is pooled and protects both stage 2 and stage 3, whereas safety stock at either of the dummy nodes is committed to a single downstream stage. The spanning-tree algorithm will determine the service times for stage 1 (S_1) and for the two dummy stages (S_{12} and S_{13}) so as to minimize the safety stock costs.

As an example, suppose the replenishment time for stage 1 is T_1 days. Then, the feasible set for the service times is given by the constraints $0 \leq S_{12} \leq T_1$ and $0 \leq S_{13} \leq T_1$. In the following discussion we consider the possible specifications for S_{12} and S_{13} to illustrate how the modified network would model different service times and the safety stock implications:

a. $S_{12} = S_{13} = T_1$. There is no need for any safety stock, pooled or dedicated, at stage 1, since the service time equals the replenishment time for both stages; we set $S_1 = T_1$.

b. $0 \leq S_{12} < S_{13} = T_1$. There is no need to hold safety stock, either pooled or dedicated, to protect demand at stage 3 since the service time equals the replenishment time; thus, we have $S_1 = T_1$. But there is a need to have a dedicated safety stock at stage 1 for demand at stage 2, the size of which depends upon the maximum demand at stage 2 over the net replenishment time of S_1 - S_{12}. In the modified network, this stock appears at the node "dummy 1-2."

c. $0 \le S_{12} \le S_{13} < T_1$. Stage 1 needs to hold a safety stock for demand for both stage 2 and stage 3. To benefit from pooling, in this model representation, we would find that $S_1 = S_{13}$. Thus, the pooled safety stock at stage 1 would be the maximum demand for stage 2 and 3 over the net replenishment time of T_1 - S_{13}. There would be no dedicated safety stock for stage 3 as $S_1 = S_{13}$. But there again is a need to have a dedicated safety stock at stage 1 for demand at stage 2, the size of which depends upon the demand at stage 2 over the net replenishment time of $S_1 - S_{12} = S_{13} - S_{12}$. In the modified network, this stock appears at the node "dummy 1-2."

Other cases when $S_{12} > S_{13}$ can be mapped into (b) or (c).

When there are more than two downstream stages, this modified network would again have one dummy node for each downstream stage. But optimization of the resulting modified network provides only an approximation. To explain the nature of the approximation, suppose there are three downstream stages, stages 2, 3, and 4. Then the modified network permits a pooled safety stock at stage 1 that is available to all of the downstream stages, and allows for a dedicated safety stock for each of the three downstream stages. But it may be that the best safety stock policy would hold safety stock at stage 1 for two of the downstream stages, say stage 2 and 3,and no safety stock for stage 4. In this case there would be a pooling benefit for holding safety stock at stage 1 for stages 2 and 3 that is not correctly modeled in the current network model.

Thus, the proposed approach would provide an upper bound on the actual inventory requirements when there is more than two downstream stages.

To extend this approach for the case of three or more downstream stages, we need to add additional dummy nodes to represent each “type” of safety stock that might be held at stage 1. In the example, we would need to add a dummy node for holding a pooled safety stock for stage 2 and 3; this dummy node would be supplied by stage 1, and then would supply the dummy nodes for the dedicated safety stocks for stage 2 and for stage 3. Considering all combinations we would need a total of six dummy nodes when there are three downstream stages, and in general, $2^n - 2$ for n downstream stages; and the resulting network would no longer be a spanning tree. Hence we do not regard this extension as being practical.

We conclude this appendix by discussing under what circumstances one might want to consider quoting different service times to downstream stages. This could be useful in terms of guiding where to modify the network by adding dummy nodes. Whereas the model could completely relax the assumption of a common service time for all downstream stages, this would approximately double the number of nodes in the network. We expect that in practice one would only permit this possibility where it could make a difference in holding costs.

One instance is when the downstream stages represent distinctly different market channels. In this case, the downstream stages could differ in terms of the stability of their demand streams, their production lead-times, and the service expectations for their markets.

For instance, one can envision two different channels of distribution, an OEM channel that has a steady volume dictated by annual purchase contracts and a consumer market facing a very volatile demand process. This case can be a candidate for creating different stockpiles of inventory, and thus different service times, for the different demand streams.

A more common example involves multiple market segments with different maximum service time requirements. For example, a product may be sold through superstores and small retail outlets. The retail outlets may hold no inventory in the store, only a demonstration unit to show customers. For these customers, product has to be shipped out within 24 hours of the order placement. The superstores carry an inventory of the product on-site and typically place orders a month in advance. The guaranteed service time to the retail segment is one day but is thirty days for the superstore segment. From the perspective of the supply chain, the retail segment must be treated as a make-to-order operation but the superstore segment can be treated like a make-to-stock operation. In the framework of the model, quoting the retail segment a low service time is a necessity but quoting the superstore segment the same service time would be needlessly costly since they do not require delivery for weeks.

Finally, we note that the quoting of different service times becomes more prevalent when there is less benefit from pooling demand at the upstream stage. For instance, if the downstream demand streams are positively correlated, then there may be limited benefit from demand pooling. In this case, the cost penalty of having dedicated safety stocks for the downstream demand, rather than a pooled safety stock, may be low.

Acknowledgement

This research has been supported in part by the Eastman Kodak Company, by the MIT Leaders for Manufacturing Program, a partnership between MIT and major US manufacturing firms, and by the MIT Integrated Supply Chain Management consortium.

References

Axsäter S., "Continuous Review Policies for Multi-Level Inventory Systems with Stochastic Demand," in: S. C. Graves, A. H. Rinnooy Kan and P. H. Zipkin (eds.), Handbooks in Operations Research and Management Science, Vol 4., Logistics of Production and Inventory, North-Holland, Amsterdam, 1993, chapter 4.

Black, B. E., "Utilizing the Principles and Implications of the Base Stock Model to Improve Supply Chain Performance," S.M. Thesis, Leaders for Manufacturing Program, MIT, Cambridge, MA., June 1998.

Coughlin, R. L., "Optimization and Measurement of a World-Wide Supply Chain," S.M. Thesis, Leaders for Manufacturing Program, MIT, Cambridge, MA., June 1998.

Diks, E. B., A. G. de Kok and A. G. Lagodimos, "Multi-echelon Systems: A service Measure *European Journal of Operations Research*, 95 (1996) 241-263.

Ettl, M., G. E. Feigin, G. Y. Lin and D. D. Yao, "A Supply Network Model with Base-Stock Control and Service Requirements," IBM Technical Report (RC 20473), 43 pages, 1996. Submitted to *Operations Research*.

Federgruen, A., "Centralized Planning Models for Multi-Echelon Inventory Systems Under Uncertainty," in: S. C. Graves, A. H. Rinnooy Kan and P. H. Zipkin (eds.), Handbooks in Operations Research and Management Science, Vol 4., Logistics of Production and Inventory, North-Holland, Amsterdam, 1993, chapter 3.

Felch, J. A., "Supply Chain Modeling for Inventory Analysis," S.M. Thesis, Leaders for Manufacturing Program, MIT, Cambridge, MA., June 1997.

Gallego, G., and P. Zipkin, "Qualitative Analysis of Multi-Stage Production-Transportation Systems: Stock Positioning and Performance Estimation," working paper, Columbia University, New York NY, July 1994.

Glasserman, P., and S. Tayur, "Sensitivity Analysis for Base-stock levels in Multiechelon Production-inventory Systems, *Management Science*, 41 (1995), 263-281.

Graves, S. C., "Safety Stocks in Manufacturing Systems," *Journal of Manufacturing and Operations Management*, 1 (1988), 67-101.

Graves, S. C., D. B. Kletter and W. B. Hetzel, "A Dynamic Model for Requirements Planning with Application to Supply Chain Optimization," *Operations Research*, 46 (1998) S35-S49.

Graves, S. C., and S. P. Willems, "Strategic Safety Stock Placement in Supply Chains," in *Proceedings of the 1996 MSOM Conference*, Hanover, NH, 1996.

Inderfurth, K., “Safety Stock Optimization in Multi-stage Inventory Systems,” *International Journal of Production Economics,* 24 (1991), 103-113.

Inderfurth, K., “Valuation of Leadtime Reduction in Multi-Stage Production Systems,” in: G. Fandel, T. Gulledge and A. Jones (eds.), Operations Research in Production Planning and Inventory Control, Springer, Berlin, 1993, 413-427.

Inderfurth, K., “Safety Stocks in Multistage Divergent Inventory Systems: A Survey,” *International Journal of Production Economics,* 35 (1994), 321-329.

Inderfurth, K., and S. Minner, “Safety Stocks in Multi-Stage Inventory Systems under Different Service Levels,” Working Paper No. 11, Otto-von-Guericke-Universitat Magdeburg, Fakultat fur Wirtschaftswissenschaft, Magdeburg, Germany, 1995.

Kimball, G. E., “General Principles of Inventory Control,” *Journal of Manufacturing and Operations Management*, 1 (1988), 119-130.

Lee, H. L., and C. Billington, “Material Management in Decentralized Supply Chains,” *Operations Research*, 41 (1993), 835-847.

Luenberger, D. G., Introduction to Linear and Nonlinear Programming, Addison Wesley, Reading, MA, 1973.

Minner, S., “Dynamic Programming Algorithms for Multi-Stage Safety Stock Optimization,” Working Paper No. 12, Otto-von-Guericke-Universitat Magdeburg, Fakultat fur Wirtschaftswissenschaft, Magdeburg, Germany, 1995.

Rosling, K., “Optimal Inventory Policies for Assembly Systems under Random Demands,” *Operations Research*, 37 (1989), 565-579.

Simpson, K. F., “In-process Inventories,” *Operations Research*, 6 (1958), 863-873.

Van Houtum, G. J., K. Inderfurth and W. H. J. Zijm, “Materials Coordination in Stochastic Multi-echelon Systems,” *European Journal of Operations Research*, 95 (1996), 1-23.