OPTIMIZING INVENTORY IN A MULTI-ECHELON MULTI-ITEM SUPPLY CHAIN WITH TIME-BASED CUSTOMER SERVICE LEVEL AGREEMENTS

Kathryn E. Caggiano
University of Wisconsin, Madison, WI 53706, kcaggiano@bus.wisc.edu

Peter L. Jackson
Cornell University, Ithaca, NY 14853, pj16@cornell.edu

John A. Muckstadt
Cornell University, Ithaca, NY 14853, jack@orie.cornell.edu

James A. Rappold
University of Wisconsin, Madison, WI 53706, jrappold@bus.wisc.edu

Abstract: *We present a continuous-review inventory model for tactical planning in a multi-item, multi-echelon service parts distribution system with time-based service level requirements. Our goal is to determine base stock levels for all items at all locations so that the service level agreements are met at minimum investment. We present exact time-based fill rate expressions for each item and efficient methods for their computation. We develop a greedy algorithm to find near-optimal solutions to large-scale problems quickly and a Lagrangian-based approach that provides near-optimal solutions and good lower bounds with increased computational effort.*

Keywords: *Inventory and service optimization; Distribution resource planning; Multi-echelon inventory; Service parts management; Time-based fill rates; Modeling methodologies.*

1 Introduction

In the realm of service parts management, relationships between suppliers and customers are often established through service agreements that extend over months or years. These agreements typically apply to the equipment the customer has purchased, and specify the type and timing of the service that will be provided. Service agreement details vary in nature and in complexity, often involving specific time-based guarantees and covering multiple pieces of equipment across multiple customer locations. The decisions faced by supply managers in positioning inventory to satisfy these service agreements at minimum investment have become exceptionally difficult. Many of the decision support models currently available are not adequate to the task.

Among the shortcomings of these models is their inability to accurately capture the time-based aspect of the service agreements. Another is the assumption that "service levels" are synonymous with "item fill rates". Since service agreements are written from the customer's perspective and *the customer's concern is the maintenance of the equipment*, not the maintenance of the individual component parts, these models are inconsistent with business practice.

In this paper, we consider a multi-item, multi-echelon distribution system with time-based service level requirements. Locations at the lowest level, or echelon, of the distribution network experience demand for parts on a continual basis. Our objective is to determine base stock levels for each item at each location so that all service level requirements stipulated by the collection of agreements are satisfied while minimizing the system inventory investment. To this end, we provide an exact characterization of what we call *channel fill rates* for each item at a demand location. Channel fill rates are the building blocks needed to represent time-based service level requirements. For instance, for a critical piece of equipment at a customer location, we can require the service level to be 90% instantaneous, 95% within 8 hours, and 98% within 2 days.

Our work has yielded several managerial insights regarding how service level agreements impact the procurement and positioning of inventory throughout the supply chain. First, since suppliers sometimes promise high levels of service to low-demand rate customers, the placement of stocking locations and the assignment of customers to stocking locations can greatly impact the cost associated with satisfying service level agreements. Second, for any demand location in a three-echelon network, the longest-horizon channel fill rate typically will be very high, even with a minimal amount of stock in the channel. Hence, it is the instantaneous and intermediate-horizon service level constraints that drive the inventory investment. Finally, it is sometimes cost-effective to stock items at demand locations even if there are no instantaneous service level constraints at these locations. This important and counterintuitive result stems from the fact that stock held at a demand location is *dedicated* to satisfying demand at that location, while stock held at a higher-level location may be used to satisfy demand or replenishment orders from many different demand locations.

The remainder of the paper is organized as follows. In Section 2, we review the relevant literature. In Section 3, we describe our modeling framework and state the optimization problem. In Section 4, we present exact expressions for the channel fill rates and describe methods for their efficient computation. In Section 5, we describe three solution approaches to the problem. In Section 6 we evaluate the performance of these three algorithms on example problems.

2 Literature review

There are relatively few papers in the research literature that consider inventory optimization in a multi-echelon system subject to time-based service level constraints. The two papers that are most closely related to ours are Cohen et al. (1986) and Ettl et al. (2000). In Cohen et al. (1986), the authors consider the problem of setting base stock levels in a single-item, multi-echelon distribution system subject to a single time-based fill rate constraint. Their periodic model assumes that the system will be "reset" to a starting condition at the end of the time period through regular replenishment shipments. The objective is to minimize total holding costs, regular replenishment shipment costs, and emergency shipment costs. Emergency shipments are made to satisfy demand shortages, subject to sharing rules. A weighted average time-based fill rate over all locations is used to measure the service level. Their solution procedure is a recursive branch and bound algorithm that solves the problem one echelon at a time, with the service level constraint evaluated laboriously once a lowest-echelon solution is reached.

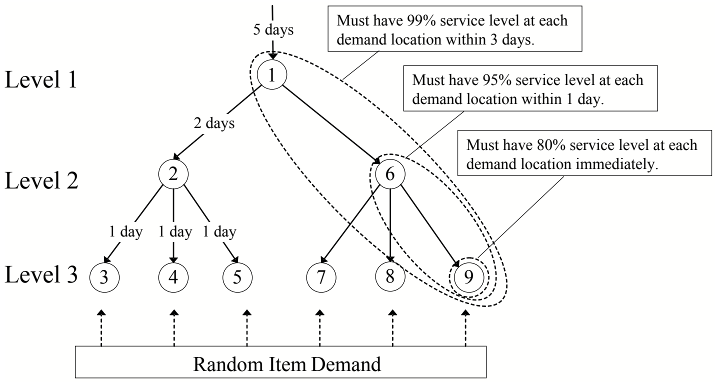
Ettl et al. (2000) examine a multi-echelon, uncapacitated supply chain with both distribution and assembly structures. A queueing-based approximation is used to capture replenishment delays due to stockouts, which are incorporated into the computation of actual lead times, with simplifications made to deal with the assembly structures. Our work is different in several ways. First, our model allows for a richer set of service level requirements, since we do not require service levels to be associated with individual end-items. That is, service level requirements may depend on the time-based fill rates of multiple items at multiple demand locations. Second, our fill rate expressions are exact; we have not assumed independence of replenishment delays between echelons. Moreover, our computationally efficient approximation method has been validated via a simulation study. Finally, our solution approaches are markedly different, and they are effective on large-scale problems. Ettl et al. (2000) do not address scalability issues.

3 The model

We consider a multi-item, multi-echelon distribution system with the following properties:

- The network has a tree-like structure, as in Figure 1, where each location is replenished from a unique parent location at the next-higher level. The sole location at the top level (hereafter called location 1) is replenished over a constant lead time for each item.
- Demand occurs at the leaf locations of the network. Without loss of generality, all such *demand locations* are assumed to be on the same level of the distribution network.

- The demand processes for all items at all demand locations are mutually independent Poisson processes with known demand rates, and unsatisfied demand is backordered.
- All items are replenished on a one-for-one, first-come, first-serve basis at all locations.
- Transport times for each item between adjacent network locations are constant.



Must have 99% service level at each demand location within 3 days.

Must have 95% service level at each demand location within 1 day.

Must have 80% service level at each demand location immediately.

Figure 1: An Example Distribution Network with Service Level Requirements

For ease of exposition, we will use the following notation in the remainder of the paper:

Distribution Network Parameters

I = the set of items, indexed by i.

J = the set of locations, indexed by j.

J^v = the set of locations at level $v = 1, 2, ..., n$. $\bigcup_{v=1}^{n} J^v = J$, $J^{v_1} \bigcap J^{v_2} = \emptyset, v_1 \neq v_2$.

P_j = the set of locations in the unique path from demand location $j \in J^n$ to the top level location in the distribution network, inclusive. This is called the *channel* associated with location j.

$P_j(v)$ = the unique location in the channel P_j at level v.

$p(j)$ = the parent location of location j in the distribution network, $j \neq 1$.

T_{ij} = the transport time for item i from location $p(j)$ to location j.

τ_{ij} = the expected replenishment time for item i from location $p(j)$ to location j.

c_i = the unit investment cost of item i.

Service Level Requirement and Demand Parameters

K = the set of service level constraints, indexed by k.

F_k = the target service level of constraint $k \in K$. For all $k \in K$, $F_k < 1$.

λ_{ij} = the rate at which orders for item i arrive at location j.

λ_{ijk} = the rate at which orders for item i that are associated with service level constraint k arrive at location j.

λ_k = the total rate at which orders for service parts associated with service level constraint k are placed. That is, $\lambda_k = \sum_{i \in I, j \in J^n} \lambda_{ijk}$.

w_{ijk} = λ_{ijk}/λ_k, the fraction of orders for service parts associated with service level constraint k that are for item i at location j.

v_k = the level of the distribution network with which service level constraint k is concerned, $v_k \in \{1, 2, ..., n\}$.

Stock Levels and Fill Rates

s_{ij} = the stock level of item i at location j.

$\mathbf{s^v}$ = the vector of stock levels of all items $i \in I$ at all network locations $j \in J^v$.

$\mathbf{s_{iP_j}}$ = the vector of stock levels of item i at the locations in the channel P_j.

$f_{ij}^v(\mathbf{s_{iP_j}})$ = the probability that an incoming order for item i at location $j \in J^n$ can be filled within the transport time from location $P_j(v)$.

Given the defined notation, we state the *Service Level Satisfaction* problem, or (SLS) as:

$$(SLS) \qquad \min_{s_{ij} \geq 0, \text{ integer}} \quad \sum_{i \in I} \sum_{j \in J} c_i s_{ij} \tag{3.1}$$

$$\text{subject to} \qquad \sum_{i \in I} \sum_{j \in J^n} w_{ijk} f_{ij}^{v_k}(\mathbf{s_{iP_j}}) \quad \geq \quad F_k \quad \forall k \in K. \tag{3.2}$$

The objective of the *SLS* model is to minimize the total base-stock investment. There are two sources of complexity in the service level constraints (3.2). First, each channel fill rate f_{ij}^v may appear in multiple service level constraints in combination with other channel fill rates, so the constraint set may not be separable by item or by location. Second, the channel fill rate functions themselves are complicated. For a given item i and demand location j, each channel fill rate f_{ij}^v, $v = 1, ..., n$, depends in a nonlinear way on the n stock levels in the channel P_j.

4 Channel fill rate functions

In this section we present exact expressions for channel fill rates in terms of the probability distributions of outstanding orders. The complete derivation can be found in Caggiano et al. (2005). Because of the computational effort required to calculate these expressions, they are not suitable for optimization of large-scale problems. Hence, using key facts from our analysis, we propose an approximation for calculating channel fill rates that is computationally efficient and yields very small estimation errors. We also define expressions for the channel fill rate gradients which are used in our optimization procedures.

Consider a particular item i in the channel composed of locations 1, 2, and 3 in the distribution network, as shown in Figure 2. Location 3 is the demand location for which we will present the probability expressions for the channel fill rates. Let location a represent all locations that are replenished by location 1 *except* for location 2, and let location b represent all locations replenished by location 2 *except* for location 3. For clarity, we suppress the i subscript on all variables and parameters and define:

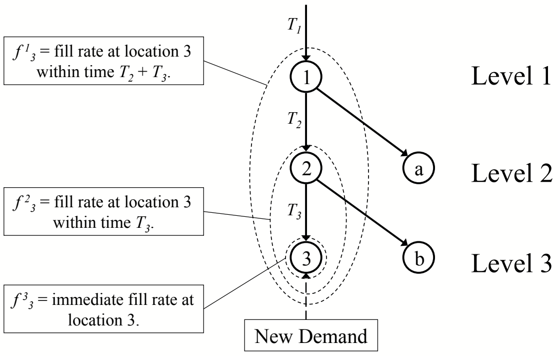

f^1_3 = fill rate at location 3 within time $T_2 + T_3$.
f^2_3 = fill rate at location 3 within time T_3.
f^3_3 = immediate fill rate at location 3.
T_1
T_2
T_3
1
2
3
a
b
Level 1
Level 2
Level 3
New Demand

Figure 2: Item distribution network

Y_j = the number of units on order at location j, $j = 1, 2, 3, a, b$.
N_j = $[Y_j - s_j]^+$, the number of units backordered at location j, $j = 1, 2, 3, a, b$.
E_j = the number of units en route from location $p(j)$ to location j, $j = 2, 3, a, b$.
Z_j = $(Y_j - E_j)$, the number of units on order at location j that are backordered at location $p(j)$, $j = 2, 3, a, b$. This represents the number of units currently on order at location j that will not arrive at location j within T_j units of time.

N_{12} $= [Z_2 - s_2]^+$, the number of units backordered at location 2 that are backordered at location 1. This represents the number of units currently backordered at location 2 that will not arrive at location 2 within T_2 units of time.

W_j = the number of units on order at location j that are backordered at location 2 and at location 1, $j = 3, b$ (i.e., the portion of N_{12} that is owed to location j). This represents the number of units currently on order at location j that will not arrive at location j within $T_2 + T_j$ units of time.

Given these definitions, exact expressions for the channel fill rates at location 3 may now be expressed using the probability distributions of Y_1, Y_2, and Y_3:

$$f_3^3(s_3, s_2, s_1) = \quad \Pr[Y_3 < s_3], \tag{4.1}$$

$$f_3^2(s_3, s_2, s_1) = \begin{cases} \Pr[Y_2 < s_2], & \text{if } s_3 = 0, \\ \Pr[Z_3 < s_3] = \Pr[Y_2 < s_2 + s_3] + \sum_{x=0}^{s_3-1} h_2(s_3, x), & \text{if } s_3 > 0, \end{cases} \tag{4.2}$$

$$f_3^1(s_3, s_2, s_1) = \begin{cases} \Pr[Y_1 < s_1], & \text{if } s_3 = s_2 = 0, \\[2ex] \Pr[Z_2 < s_2] & \\ \quad = \Pr[Y_1 < s_1 + s_2] + \sum_{x=0}^{s_2-1} h_1(s_2, x), & \text{if } s_3 = 0, s_2 > 0, \\[2ex] \Pr[W_3 < s_3] & \\ \quad = 1 - \left[\sum_{x=s_3}^{\infty} \sum_{y=x}^{\infty} \binom{y}{x} \left(\frac{\lambda_3}{\lambda_2}\right)^x \left(1 - \frac{\lambda_3}{\lambda_2}\right)^{y-x} h_1(s_2 + y, s_2 + y)\right], & \\ & \text{if } s_3 > 0, \end{cases} \tag{4.3}$$

where

$$h_l(u, x) = \sum_{z=u}^{\infty} \binom{z}{x} \left(\frac{\lambda_{l+1}}{\lambda_l}\right)^x \left(1 - \frac{\lambda_{l+1}}{\lambda_l}\right)^{z-x} \Pr[Y_l = s_l + z], \quad \text{for } l = 1, 2,$$

denotes the probability that *there are at least u backorders at location l and exactly x of these are owed to location $l+1$*. For a general n-level channel with location n representing the demand location, the channel fill rates at all levels $v = 1, ..., n$, are given by:

$$f_n^v(s_n, \ldots, s_1) = 1 - \left[\sum_{x_v = \mathcal{S}_v}^{\infty} \sum_{x_{v+1} = \mathcal{S}_{v+1}}^{x_v - s_v} \cdots \sum_{x_n = \mathcal{S}_n}^{x_{n-1} - s_{n-1}} B(x_v - s_v) B(x_{v+1} - s_{v+1}) \ldots \right. \\ \left. B(x_{n-1} - s_{n-1}) \Pr[Y_v = x_v]\right] \tag{4.4}$$

where $\mathcal{S}_l = \sum_{j=l}^{n} s_j$ is the total installation stock at and below network level l in the channel P_n, and $B(x_l - s_l) = \binom{x_l - s_l}{x_{l+1}} \left(\frac{\lambda_{l+1}}{\lambda_l}\right)^{x_{l+1}} \left(1 - \frac{\lambda_{l+1}}{\lambda_l}\right)^{(x_l - s_l) - x_{l+1}}$ is the binomial probability that x_{l+1} of the $(x_l - s_l)$ backorders at location l are owed to location $l + 1$.

We can compute the channel fill rate f_3^v for any level $v = 1, 2$, or 3, using (4.4) if the distribution of Y_v is available. Since $Y_1 \sim \text{Poisson}(\lambda_1 T_1)$, the fill rate f_3^1 can be computed exactly using equation (4.4). The probability distributions of Y_2 and Y_3, however, are difficult to characterize in general. Graves (1985) shows how to calculate the mean and variance of Y_2 for a given s_1, and transport times, T_1 and T_2. Namely,

$$E[Y_2] = \lambda_2 T_2 + \frac{\lambda_2}{\lambda_1} E[N_1], \text{ and} \tag{4.5}$$

$$Var[Y_2] = \lambda_2 T_2 + \frac{\lambda_2}{\lambda_1}\left(1 - \frac{\lambda_2}{\lambda_1}\right) E[N_1] + \left(\frac{\lambda_2}{\lambda_1}\right)^2 Var[N_1]. \tag{4.6}$$

We assume that Y_2 has a *negative binomial* distribution with mean and variance given by (4.5) and (4.6). We recursively compute moments for Y_3 and approximate its distribution as negative binomial as well. While the moment calculations for Y_2 are exact, those for Y_3 are approximate. In general, the moments of Y_v are based on $v-1$ approximations, so we may expect the accuracy of f_n^v to degrade as v increases.

Given the distribution of Y_1 and the approximations for Y_2 and Y_3, we can compute channel fill rates using (4.4). This approach is the *direct approximation method* since we directly compute the distributions of Z_3, Z_2, and W_3 for the fill rates f_3^2 and f_3^1. These computations are time-consuming and require the recursive convolution of the conditional distributions of Z_3 and Z_2 with those of N_2, N_1, and N_{12}. If we are interested only in *evaluating* the service levels achieved in the distribution system with *known* base stock levels, then this method is a reasonable approach that yields highly accurate values. However, for purposes of *optimizing* large-scale systems, the method is not viable because of the computational effort required.

The *indirect approximation method* exploits the fact that the *moments* of Z_3, Z_2, and W_3 are easy to compute, since they rely only on the *moments* of N_2, N_1, and N_{12}, respectively. We extend the negative binomial assumption to the distributions of Z_3, Z_2, and W_3. For optimization purposes, Caggiano et al. (2005) validate that this method is far more computationally efficient than the direct method and does not compromise fill rate accuracy significantly.

The computational efficiency of the optimization procedures also depends on estimating the incremental *changes* to the channel fill rates quickly, without actually having to update any stock levels. The nine channel fill rate gradients in our three-level system are:

$$\begin{aligned}
\frac{\Delta f_3^v}{\Delta s_3} &\equiv f_3^v(s_3+1, s_2, s_1) - f_3^v(s_3, s_2, s_1), \quad v = 1, 2, 3,\\
\frac{\Delta f_3^v}{\Delta s_2} &\equiv f_3^v(s_3, s_2+1, s_1) - f_3^v(s_3, s_2, s_1), \quad v = 1, 2, 3, \qquad (4.7)\\
\frac{\Delta f_3^v}{\Delta s_1} &\equiv f_3^v(s_3, s_2, s_1+1) - f_3^v(s_3, s_2, s_1), \quad v = 1, 2, 3.
\end{aligned}$$

In general, there are n^2 channel fill rate gradients in an n-level system. Using the channel fill rate expressions, we are able to estimate their gradients using a well-known Poisson approximation technique in conjunction with the chain rule.

We close this section with two observations exploited by our solution procedures. First, from (4.1), (4.2) and (4.3), all channel fill rates can be made arbitrarily close to 100% by increasing the demand location stock level s_3, regardless of the stock levels s_2 and s_1. Thus, a feasible solution can always be found for *SLS*. Second, for given stock levels $s_1 \geq \lfloor \lambda_1 T_1 \rfloor$ and $s_2 \geq \lfloor \lambda_2 \tau_2 \rfloor$, the channel fill rates are *concave* functions in s_3 for $s_3 \geq \lfloor \lambda_3 \tau_3 \rfloor$.

5 Solution procedures

In this section we describe two procedures for solving problem *SLS* approximately, *FastIncrement* and *PrimalDual*, as well as a third *Naive* approach, used for comparison purposes. The *Naive* approach is a greedy marginal analysis technique that increments stock levels *at the demand locations only* until all service level constraints are satisfied. The *FastIncrement* approach is a marginal analysis technique that increments stock levels *throughout the entire network* until all service level constraints are satisfied. Finally, the *PrimalDual* approach is a Lagrangian-based procedure which yields both near-optimal solutions and good lower bounds with increased computational effort. For notational convenience, we assume the distribution network has three levels. We also assume that for each item i and each location j, nonnegative integer upper and lower bounds ($\overline{s_{ij}}$ and $\underline{s_{ij}}$, respectively) have been established for s_{ij}.

5.1 The Naive procedure

The *Naive* solution approach is a three-step process that exploits the fact that a feasible solution for problem *SLS* can always be found by adjusting the demand location stock level vector, $\mathbf{s}^3$, regardless of how we set the upper level stock vectors $\mathbf{s}^1$ and $\mathbf{s}^2$:

- First, permanently fix the stock levels for all items at all *non-demand locations* to values given by the *lower bound* vectors, $\underline{\mathbf{s}^1}$ and $\underline{\mathbf{s}^2}$.
- Second, for item i at demand location j, set $s_{ij} = \max\{\underline{s_{ij}}, \lfloor \lambda_{ij}\tau_{ij} \rfloor\}$ where τ_{ij} is a function of the stock levels of item i at the non-demand locations in P_j. This typically guarantees that the channel fill rate functions f^v_{ij}, $v = 1, 2, 3$, will be concave in s_{ij}.
- Third, increment stock levels *at the demand locations only* until all service level constraints are satisfied using the following greedy marginal analysis algorithm:

Increment-Demand-Locations-Until-Feasible

```
INPUT: An instance of problem $SLS$; $\mathbf{s}$, an integral solution to $SLS$ (not necessarily feasible).
OUTPUT: $\mathbf{s}$, a feasible integral solution to $SLS$.

1. Determine $\bar{K} \subseteq K$, the set of all unsatisfied service level constraints with respect to the current
   stock level vector $\mathbf{s}$, and for each constraint $k \in \bar{K}$, compute the current feasibility gap,
       $g_k = F_k - \sum_{i \in I} \sum_{j \in J^3} w_{ijk} f^{v_k}_{ij}(s_{ij}, s_{ip(j)}, s_{i1}).$
2. If $\bar{K} = \emptyset$, then STOP and return $\mathbf{s}$.
3. For all $i \in I, j \in J^3$, compute $\Delta_{ij} = \sum_{k \in \bar{K}} \Delta_{ijk}$ , where $\Delta_{ijk} = \min\{w_{ijk} \frac{\Delta f^{v_k}_{ij}}{\Delta s_{ij}}, g_k\}$ is the
   incremental contribution of $s_{ij}$ towards the satisfaction of constraint $k$.
4. Find the pair $(i^*, j^*)$ such that: $(i^*, j^*) = \arg\max_{(i,j)} \frac{\Delta_{ij}}{c_i}$.
5. $s_{i^*j^*} \leftarrow s_{i^*j^*} + 1$.
6. Go to step 1.
```

The solution returned by *Increment-Demand-Locations-Until-Feasible* will not (in general) be optimal for *SLS*, or even optimal with respect to the fixed stock vectors $\underline{\mathbf{s}^1}$ and $\underline{\mathbf{s}^2}$. However, *Naive* is very fast computationally. Because we increment stock levels at demand locations only, each iteration requires the updating of values related to exactly three channel fill rates, and the only service level constraints affected are those that include one of these fill rates.

5.2 The FastIncrement procedure

The *FastIncrement* procedure greedily increments stock levels until all service level constraints are satisfied using a multi-step process that exploits the speed of the *Increment-Demand-Locations-Until-Feasible* algorithm while allowing for a broader (and usually better) range of solutions. The procedure works as follows:

- First, temporarily set the stock levels for all items at all *non-demand locations* to values given by the *upper bound* vectors, $\overline{\mathbf{s}^1}$ and $\overline{\mathbf{s}^2}$.
- Second, for item i at demand location j, set $s_{ij} = \underline{s_{ij}}$.
- Third, increment stock levels at the *demand locations only* until all service level constraints are satisfied using *Increment-Demand-Locations-Until-Feasible.* With the non-demand location stock levels fixed at their upper bounds, this step establishes a good *lower bound* solution at the demand locations in the sense that the resulting $\mathbf{s}^3$ vector is likely to be near-optimal in the absence of replenishment delays.

- Fourth, reset the stock levels for all items at *all non-demand locations* to the *lower bounds* given by $\underline{\mathbf{s}}^1$ and $\underline{\mathbf{s}}^2$.
- Fifth, increment stock levels at *all network locations* using the greedy algorithm *Increment-All-Locations-Until-Feasible.* This algorithm is identical to *Increment-Demand-Locations-Until-Feasible*, except that step 3 is replaced with:

3. For all $i \in I, j \in J$, compute $\Delta_{ij} = \sum_{k \in \bar{K}} \Delta_{ijk}$, where $\Delta_{ijk} = \min\{\sum_{j' \in J^3 : j \in P_{j'}} w_{ij'k} \frac{\Delta f_{ij'}^{v_k}}{\Delta s_{ij}}, g_k\}$ is the *total* incremental contribution of s_{ij} towards the satisfaction of constraint k.

In step 3 of *Increment-All-Locations-Until-Feasible*, items at *all locations* in the network are candidates for incrementing, not just items at the demand locations. This is important since when an item at a non-demand location j is incremented, the channel fill rates associated with *every demand location that has j in its channel* are affected. Moreover, to determine the marginal contribution of incrementing item i at location j, we must compute the gradients to the channel fill rates at *all* demand locations j' that have j in their channels. Unlike *Naive*, the running time of *FastIncrement* is hyperlinear in the number of non-demand locations. For searching, however, *FastIncrement* is more versatile than the *Naive* procedure and, for most problems instances, produces better solutions.

5.3 The PrimalDual procedure

Given a set of multiplier values $\Theta = \{\theta_k : k \in K\}$ for the service level constraints, the *PrimalDual* procedure uses Lagrangian relaxation to decompose *SLS* into $|I|$ single item subproblems. These subproblems are solved using a semi-enumerative approach. The result is a solution to *SLS* that is almost always infeasible, but whose objective function value is a *guaranteed lower bound* on the optimal objective function value. The multiplier values Θ are adjusted and the entire process repeated until a user-specified limit has been reached (either number of iterations or optimality gap). In any iteration, the solution providing the lower bound can be made feasible using *Increment-Demand-Locations-Until-Feasible* or *Increment-All-Locations-Until-Feasible.* Empirically, this results in excellent feasible solutions to *SLS.* See Caggiano et al. (2005) for a full discussion.

Although the running time of *PrimalDual* increases linearly in the number of items due to the decomposition, the semi-enumerative search used within each item subproblem makes this procedure far more computationally intensive than the other two. That is, the running time of *PrimalDual* grows hyperlinearly with the size of the stock level search ranges which, in turn, are driven by the *volume of item demand.* In order to keep this approach computationally viable for large-scale problems, some form of scaling can be undertaken with respect to the stock levels in the exhaustive search.

6 Example problems

We consider two example *SLS* problem instances. Table 1 summarizes the network structure and parameter values for the Small and Large problems, as well as the item costs and demand rates by location for the Small problem.

We use the Small problem to illustrate some fundamental differences between the solutions generated by the *Naive*, *FastIncrement*, and *PrimalDual* procedures and to highlight the advantages and disadvantages of each approach. The Large problem was provided by an industrial client and is more indicative of the problems found in practice. We provide a summary of computational results and discuss the quality and efficiency of each solution procedure.

The structure of the distribution system for the Small problem was shown earlier in Figure 1. All customers at a demand location are assumed to have the same service level requirements and

Problem Instance	Small	Large
Level 1 locations:	1	1
Level 2 locations:	2	4
Level 3 locations:	6	150
No. Items:	4	175
Max Daily Demand:	4.00	15.639672
Min Daily Demand:	0.25	0.000054
Total Daily Demand:	22.50	4762.26
Max Item Cost:	$10,000	$1,891
Min Item Cost:	$30	$0.01
T_1:	5 days	10 days
T_2:	2 days	1 day
T_3:	1 day	1 day

SMALL Problem

Item	Unit Cost	Average Daily Demand by Demand Location						Total
		3	4	5	7	8	9	
1	$10,000	2.00	0.50	0.50	0.50	0.50	0.50	4.50
2	$2,000	1.00	0.25	0.25	0.25	0.25	0.25	2.25
3	$500	3.00	0.75	0.75	0.75	0.75	0.75	6.75
4	$30	4.00	1.00	1.00	1.00	1.00	1.00	9.00
Total		10.00	2.50	2.50	2.50	2.50	2.50	22.50

Table 1: Problem Instance Summary and Small Problem Data

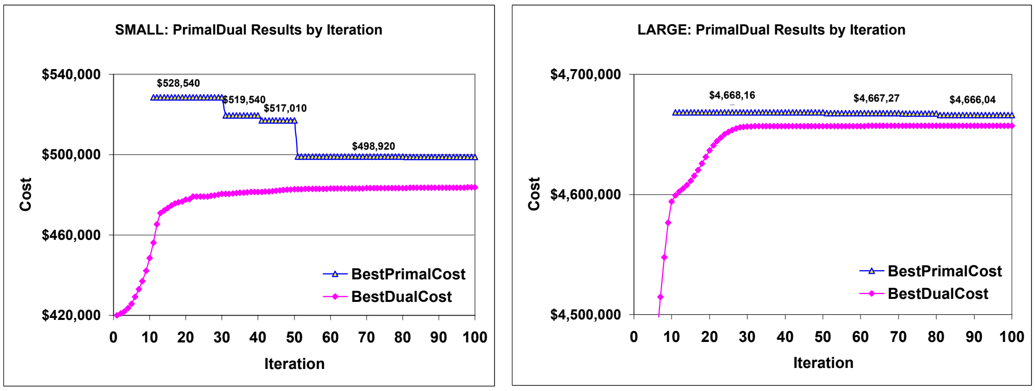
the same relative demand rates across items. Demand locations 3, 4, 5, and 9 each have service level requirements (across all four items) that specify an 80% instantaneous service level, a 95% service level within one day, and a 99% service level within three days. Demand locations 7 and 8 have the same 1-day and 3-day service level requirements, but there are no instantaneous service level requirements at these locations. The Small problem exhibits a number of characteristics that are likely to magnify the cost differentials between optimal and suboptimal solutions: a small number of items, a small number of demand locations for each intermediate level location, low demand rates, short lead times, and huge differences in the item unit costs. In this respect, the Small problem represents a worst-case scenario for the *Naive* and *FastIncrement* procedures.

SMALL Problem

	Solution Approach		
	Naïve	FastIncrement	PrimalDual*
Cost:	$548,420	$524,570	$498,920
Solution Time (sec):	<1	<1	57
Best Lower Bound:			$483,640
Optimality Gap:	13.39%	8.46%	3.16%

LARGE Problem

	Solution Approach		
	Naïve	FastIncrement	PrimalDual*
Cost:	$5,038,561	$4,668,031	$4,666,040
Solution Time (sec):	83	333	75114
Best Lower Bound:			$4,657,013
Optimality Gap:	8.19%	0.24%	0.19%



* Results shown for PrimalDual are based on 100 iterations, with feasible solutions created from the current best dual solution every 10 iterations.

Figure 3: Small and Large Problem Solution Summary

Figure 3 shows the computational results summary for the Small and Large problems. For the Small problem, the *Naive* solution holds all additional stock (above the required minimum amounts) at the demand locations. Although the majority of this additional stock is held in the lower cost, higher demand rate items, all six demand locations still must hold more units of items 1 and 2 then would otherwise be necessary to meet the service level requirements (i.e., more than if we were allowed to hold additional units of items 1 and 2 at the intermediate locations). Note that due to the 80% instantaneous service level requirements at locations 3, 4, 5, and 9, note that positive quantities of items 1, 3, and 4 must be held at these locations. The result is a solution whose total cost is at least 10% above the minimum total cost.

By contrast, the *PrimalDual* solution holds virtually all additional stock for the expensive items 1 and 2 at the intermediate locations instead of at the demand locations. For items 3 and 4, additional stock is held liberally throughout the network as needed to meet the service level constraints. The result is a more cost-effective solution that is provably near-optimal. Although a duality gap of 3.16% remains after 100 iterations, for small-scale problems such as this one, the fact that the stock levels must be integral combined with the fact that the channel fill rate functions are not jointly concave in their arguments makes it highly unlikely that the lower bound will ever become tight.

The *FastIncrement* solution to the Small problem shares properties with both the *Naive* and the *PrimalDual* solutions. For the subtree rooted at location 6, *FastIncrement*, like *PrimalDual*, holds additional stock of the expensive items 1 and 2 at the intermediate location instead of at the demand locations. For the subtree rooted at location 2, however, the *FastIncrement* solution resembles the *Naive* solution, only it is *worse*.

Why did this happen? Because the channel fill rate functions are not jointly concave in their arguments, the cost-benefit ratios that drive the sequence in which stock levels are incremented by *FastIncrement* do not always reflect good choices in the global sense. The *Naive* procedure shares this disadvantage. *FastIncrement* tends to put too much stock at the demand locations of subtrees that have many instantaneous service level constraints but only a few demand locations. It is in these cases that an incremental unit of stock at a demand location will tend, at first, to have a higher cost-benefit ratio than an incremental unit of stock at the corresponding intermediate location.

The Large problem tests the scalability of our solution procedures with 27,125 item-location combinations. The service level requirements at each demand location consisted of an 80% instantaneous service level, a 95% service level within one day, and a 99% service level within two days. Because it places all additional stock at demand locations, the *Naive* approach performs relatively poorly on the Large problem. By contrast, the *FastIncrement* procedure performs extremely well, achieving the best tradeoff between solution quality and computational effort. The *PrimalDual* procedure, while achieving the lowest cost solution for both problems, becomes computationally unattractive for the Large problem since its running time is hyperlinear in the volume of item demand. We note, however, that the running times listed for *PrimalDual* are for 100 iterations of the algorithm, and near-optimal solutions are found after only 10 iterations.

In summary, we have developed a continuous-review inventory model for a multi-item, multi-echelon distribution system with time-based service level requirements. We presented exact channel fill rate expressions, devised methods for computing channel fill rates and their gradients quickly, and developed and compared effective solution procedures for determining base stock levels that meet all service level requirements at minimum investment. The examples presented demonstrate that our *FastIncrement* and *PrimalDual* procedures, taken together, permit efficient inventory optimization in both small- and large-scale multi-echelon systems with time-based service level requirements.

References

Caggiano, K. E., Jackson, P. L., Muckstadt, J. A., and Rappold, J. A. (2005), “A Multi-Echelon, Multi-Item Inventory Model for Service Parts Management with Time-Based Service Level Constraints,” *Working paper, School of Business, University of Wisconsin-Madison, http://instruction.bus.wisc.edu/kcaggiano/papers/CR_model.pdf.*

Cohen, M., Kleindorfer, P., and Lee, H. L. (1986), "Optimal Stocking Policies for Low Usage Items in Multi-Echelon Inventory Systems," *Naval Research Logistics Quarterly*, 33, 17–38.

Ettl, M., Feigin, G. E., Lin, G. Y., and Yao, D. D. (2000), "A Supply Network Model with Base-Stock Control and Service Requirements," *Operations Research*, 48, 216–232.

Graves, S. C. (1985), "A Multi-Echelon Inventory Model for a Repairable Item with One-for-One Replenishment," *Management Science*, 31, 1247–1256.

7 Biographies

KATHRYN CAGGIANO is an assistant professor in the School of Business at the University of Wisconsin-Madison where she teaches graduate courses in operations management and data analysis. She holds a B.S. in Mathematics from the College of William and Mary, and M.S. and Ph.D. degrees in Operations Research from Cornell University. PETER JACKSON is a Professor in the School of Operations Research and Industrial Engineering and Director of the Systems Engineering Program at Cornell University. He holds a B.A. in Economics with Mathematics from University of Western Ontario, and M.Sc. and Ph.D. degrees in Operations Research from Stanford University. JOHN MUCKSTADT is the Acheson-Laibe Professor of Engineering in the School of Operations Research and Industrial Engineering at Cornell University. He served as the School's Director for nine years. His interests are in the area of supply chain system design and analysis. He holds a A.B. in Mathematics from the University of Rochester, and an M.A. in Mathematics, and M.S. and Ph.D. in Industrial Engineering from the University of Michigan. JAMES RAPPOLD is an assistant professor in the School of Business at the University of Wisconsin-Madison where he teaches graduate and executive education courses in supply chain management. He holds a B.S. in Industrial Management from Carnegie Mellon, and M.S. and Ph.D. degrees in Operations Research from Cornell University.