Backorder Minimization in Multiproduct Assemble-to-Order Systems

Yingdong Lu*
IBM T.J. Watson Research Center
Yorktown Height, NY 10598
Tel. (914) 945-3738, e-mail: yingdong@us.ibm.com

Jing-Sheng Song†
Graduate School of Management
University of California, Irvine, CA 92697
Tel. (949) 824-2482, e-mail: jssong@uci.edu

David D. Yao‡
Dept. of Industrial Engineering and Operations Research
Columbia University, New York, NY 10027
Tel. (212) 854-2934, e-mail: yao@columbia.edu

February 2003

Abstract

We consider a multiproduct assemble-to-order system. Components are built to stock with inventory controlled by base-stock rules, but final products are assembled to customer orders only. There are multiple types of orders, which arrive at the system following batch Poisson processes. The leadtimes for replenishing component inventory are stochastic. We study the optimal allocation of a given budget among component inventories so as to minimize a weighted average of backorders over product types (or, alternatively, the average response time among orders). We derive several easy-to-compute bounds and approximations for the expected number of backorders, and use them to formulate surrogate optimization problems. Efficient algorithms are developed to solve these problems, and numerical examples illustrate the effectiveness of the bounds and approximations. Our study also generates new insights on several design issues, both analytically and numerically.

*Part of this author's research was conducted at the Graduate School of Management, University of California, Irvine, as a post-doctoral researcher, and supported in part by NSF grants DMI-9896339 and DMI-0084922.

†Supported in part by NSF grants DMI-9896339 and DMI- 0084922.

‡Supported in part by NSF grant DMI-0085124, and by HK/RGC grant CUHK4376/99E.

1 Introduction

Realizing the competitive advantages of being flexible, responsive as well as efficient, and enabled by advanced technologies, more and more manufacturing companies are reengineering their product design and delivery processes to move toward mass customization. That is, to produce customized products in a cost-effective way that is comparable to mass production (see, e.g., Pine 1993). This requires modularizing the production process so that the product can be quickly assembled from standardized components and modules in different configurations, based on what customers individually request. As a result, a new kind of production-inventory system has emerged and is becoming increasingly popular. This is the assemble-to-order (ATO) system: inventories are kept only at the component level, and the final products are assembled only after customer orders are received.

This new system, in turn, presents challenging operational and system design issues to managers. As each customer order typically involves several components, the stockout of any component will cause a delay in fulfilling the order. So, the optimal stock level of one component should be determined in conjunction with those of other components to ensure the simultaneous availability. However, holding inventory of a large number of components requires significant financial commitment, making it impossible to maintain a high level of inventory across all components. Standard single-item inventory planning tools, while suitable for the mass-production make-to-stock environment, are no longer applicable. Therefore, new planning tools are needed to strike the optimal inventory-service tradeoff in ATO systems.

In this paper we consider an ATO system supporting multiple types of demand, which arrive at the system following Poisson processes. The inventory of each component is controlled by a base-stock policy; and the replenishment leadtimes among the components are independent, identically distributed (i.i.d.) random variables. The system performance measure we focus on is the expected backorders for each product. Our objective is to minimize a weighted average of backorders over all product types, subject to a budget constraint on the component inventory. Through Little's law (Wolff 1989), this objective relates directly to the response time performance in fulfilling customer orders. (Also refer to Song 2002).

Another objective of our study is to use both analysis and numerical examples to sharpen our understanding on the key performance drivers of the ATO system. For instance, the single-item inventory theory suggests that we should stock enough of each component to cover the leadtime demand plus some safety stock to protect against variability (of both demand and leadtime). Let s be the stock level of a component. Then

$$s = \mu + k\sigma,$$

where μ is the average leadtime demand for the component; σ, the corresponding standard deviation; and k, the safety factor, a positive parameter indicative of the level of protection against uncertainty or variability. What is often followed in practice is the so-called *equal fractile* rule – having the same k value across all components. Should this be the case here? What is the impact of demand or leadtime variability? What is the cost effect of component inventory? What role does the relative importance of the product play (e.g., high-margin versus low-margin products)? Both our analytical and numerical results provide insights to these issues. More broadly, the models and solution approaches developed here can be used to support system designs, such as evaluating design alternatives of product modules and configurations.

The work reported here extends the single-product ATO system studied in Song and Yao (2002). Like this previous work, we develop upper and lower bounds for the backorders, and use them as surrogates in the optimization problem. However, the bounds for the multi-product model here are more involved and exhibit different structures. For instance, the lower bound in the multi-product case leads to an optimization problem known as the min-max-sum resource allocation problem (see, e.g., Kouvelis *et al.* (2001)), for which known algorithms are limited to the branch-and-bound types. Here, we are able to identify a special property in our model, what we call a *stack structure*, and solve the problem using a much more efficient greedy algorithm. In terms of the upper bound, we show that a modification of the approach in Song and Yao (2002) produces not only a good *approximation* of the average backorder but also a surrogate optimization problem that bears the same stack structure as the one in the lower-bound approach.

In many ways, this paper can be viewed as a sequel to Lu, Song and Yao (2003a), which also studies a multi-product ATO system, but focuses on fill-rate performance. Simply put, fill rate takes the form of a probability of availability (or no-stockout) across all components; whereas backorder, the performance measure studied here, takes the form of the expectation of a maximum over backordered components. Mathematically, these two performance measures require very different approaches. Physically, fill rate has more of a system focus: how much of the exogenous demand the system can fill; backorder is more customer-oriented: as explained earlier, it is a proxy of the response time to fulfill a customer order.

A brief review of other related literature is in order. (A more detailed survey of the state-of-the-art research on ATO systems has recently appeared in Song and Zipkin 2001.) Several authors have worked on optimization models for ATO systems. These studies are quite different from ours in the detailed modelling assumptions and approaches. For example, Hausman *et al.* (1998) studies a periodic-review (discrete-time) model with multivariate normal demand and constant component replenishment leadtimes. The paper examines the problem of maximizing the fill rate, which is a multivariate normal probability, subject to a linear budget constraint. The paper develops heuristic methods. The best of these seems to be the equal k-rule mentioned above. (They refer to it as the *equal fractile heuristic*, which selects the base-stock levels to equalize the components' fill rates. Agrawal and Cohen (2001) assume the same discrete-time framework. Their model minimizes the total expected component inventory costs, subject to constraints on the order fill rates. Instead of FCFS allocation rule, they assume a combination of FCFS and a fair-share allocation policy. Their study leads to quite different solutions from the equal-fractile heuristic of Hausman et al, however. Cheng et al. (2002) also use the discrete-time formulation. They assume i.i.d. replenishment leadtimes and FCFS allocation rule. They study the problem of minimizing average component inventory holding cost subject to product-family dependent fill rate constraints. functions are linear functions of the item fill rates. An exact algorithm and a greedy heuristic are developed. Using a continuous-time framework, Gallien and Wein (2001) assume i.i.d. component leadtimes in a single-product assembly system. Demand is Poisson with rate λ. To keep the analysis tractable, the paper imposes a synchronization assumption, namely, that components are assembled in the same sequence they are ordered in. In the context of a repair shop, Cheung and Hausman (1995) assume i.i.d. leadtimes and multivariate Poisson demand model. This is a special case of ours (the unit demand case). They use a combination of order synchronization and disaggregation in their analysis, and their approximate backorder function appears to be quite different from ours. Wang (1999) considers multivariate compound Poisson demand but the supply process for each component is capacitated and modeled as a single-server queue. Applying the asymptotic results developed in Glasserman and Wang (1998), he examines the problem of minimizing average inventory cost subject to a fill-rate constraint.

The rest of the paper is organized as follows. We start with model description in §2, followed by establishing in §3 several properties of the backorder function. In §4, we develop the lower bounds to backorders, as well as solution approaches to the lower-bound problem. We then turn to considering upper bounds in §5. Extensions to batch demands are presented in §6, and numerical studies are summarized in §7.

2 Model Description

Let $\mathcal{I} := \{1, 2, \cdots, m\}$ index the set of m components. Let $\mathcal{K} := \{K \subset \mathcal{I}\}$ index the set of end-product types, where K denotes the subset of components (one unit each) required to assemble one unit of a certain end-product, referred to as a type K product. In §6 we relax the unit demand assumption and allow random batch size for each component in K. Note that $\mathcal{K}$ is not necessarily the set of *all* possible subsets of $\mathcal{I}$.

Associated with each product type K is an independent Poisson demand stream, with rate λ^K. For each component i, let $\lambda_i := \sum_{K \ni i} \lambda^K$. Then, clearly the demand for components also form Poisson processes, with rate λ_i for component i. Note, however, these Poisson processes are no longer independent. (Throughout the paper we shall use subscripts to index components and superscripts to index product/demand types.)

Upon the arrival of a demand for product type K, if there is on-hand inventory for each component $i \in K$, then the demand will be filled immediately. (In other words, the time to assemble the components into the end-product is assumed negligible.) On the other hand, if there is a stockout of one or more components in K, then the order will be backlogged. When there are multiple backorders, they are filled on a first-come-first-served (FCFS) basis.

The inventory of each component is controlled by a base-stock (or, "one-for-one" replenishment) policy, with s_i denoting the base-stock level. Specifically, the arrival of each demand K will trigger the replenishment of one unit of each component $i \in K$, regardless of whether the demand is filled or backlogged. This way, the inventory position (i.e., the on-hand inventory plus on-order units minus backorders) of component i is always maintained at a constant level, of s_i units. Assume that the replenishment leadtimes for each component are independent and identically distributed (i.i.d.) random variables, with a common distribution function denoted $G_i(\cdot)$; and the leadtimes are independent among all components. Denote $G_i^c := 1 - G_i$. Let L_i denote the generic random variable following the distribution G_i.

With the above model formulation, it is clear that for each component i, the number of outstanding orders at any time t, denoted $X_i(t)$, is equal to the number of jobs in service in an $M/G_i/\infty$ queue, with arrival rate λ_i, for $i = 1, ..., m$. However, since the arrival of any demand of type K may generate simultaneous arrivals at all the component queues, $i \in K$, these m queues are *not* independent. As shown in Lu et al. (2001), under mild regularity conditions, $(X_1(t), ..., X_m(t))$ has steady-state limit, denoted $(X_1, ..., X_m)$, which can be characterized by the following probability generating function:

$$\psi(z_1, ..., z_m) = \exp\left[\sum_{K\in\mathcal{K}} \lambda^K \int_0^\infty (\prod_{j\in K}(G_j(u) + z_j G_j^c(u)) - 1)du\right],$$

This is a multivariate Poisson distribution, with the marginal distribution of X_i, $i = 1, ..., m$, following a Poisson distribution with mean $\lambda_i \mathsf{E}[L_i]$.

Denote $x^+ := \max\{0, x\}$. We are interested in the following backorder-related performance measures:

$$\begin{aligned} B_i &= \text{number of backordered component } i = [X_i - s_i]^+, \\ B^K &= \text{number of backordered type-}K\text{ demand,} \end{aligned}$$

Furthermore, there is a breakdown of the backordered components B_i into the demand types to which they belong:

$$B_i^K := \sum_{k=1}^{B_i} 1_k(\frac{\lambda^K}{\lambda_i}) = \sum_{k=1}^{[X_i - s_i]^+} 1_k(\frac{\lambda^K}{\lambda_i}), \tag{1}$$

where 1_k's are i.i.d. Bernoulli random variables with parameter λ^K/λ_i (i.e., $1_k = 1$ with probability λ^K/λ_i, and $1_k = 0$ otherwise). Clearly, under the FCFS rule, we have

$$B^K = \max_{i \in K} B_i^K. \tag{2}$$

Let c_i be the unit cost of component i. Let $w^K \geq 0$ be a weighting factor for the average backorder for product K. It measures the relative importance of service provided to type K orders. Denote $\mathbf{s} := (s_1, ..., s_m)$. Although the decision variables s_i's are non-negative integers, below we shall treat these as non-negative real values, as they are involved in bounds and approximations. In most applications, s_i's are typically large values; hence, ignoring the integrality is appropriate.

We are interested in the following optimization problem:

$$(P) \quad \min_{\mathbf{s}} \sum_{K \in \mathcal{K}} w^K \mathsf{E}[B^K(\mathbf{s})] \qquad \text{s.t.} \quad c_1 s_1 + \cdots + c_m s_m \leq C, \quad \mathbf{s} \geq \mathbf{0}. \tag{3}$$

where $C > 0$ is given constant, representing the total inventory budget. In industrial applications, inventory budget often applies to safety stock only. Since inventory can be divided into WIP and safety stock, and the WIP part is independent of the base-stock level, the constraint in the above problem formulation is consistent with practice. (Also see Sherbrooke (1992).) As pointed out in the Introduction, the objective function in (3) relates to a delay objective via Little's law: Suppose we want to

$$\min_{\mathbf{s}} \sum_{K \in \mathcal{K}} v^K \mathsf{E}[W^K(\mathbf{s})],$$

where W^K denotes the delay (response time) in filling type K orders, and v^k is the associated cost. Then, we can write

$$v^K \mathsf{E}[W^K(\mathbf{s})] = \frac{v^K}{\lambda^K} \mathsf{E}[B^K(\mathbf{s})];$$

hence, we can set $w^K = v^K/\lambda^K$ in (3).

3 Structural Properties

We establish here several properties of the backorder functions. We first show that increasing the base-stock level of any item not only reduces the mean but also the variance of the backorders.

Proposition 1 The following properties hold for the back-order related performance measures.

- For each $i = 1, ..., m$, B_i is decreasing and convex in s_i, and so is B_i^K for all $K \ni i$.
- For each $K \in \mathcal{K}$, B^K is decreasing and convex in $\mathbf{s} = (s_1, ..., s_m)$, and hence so is $\sum_K w^K B^K$.

Consequently, for any increasing and convex function $f(x)$, the above decreasing convexity holds for $\mathsf{E}f(B_i)$, $\mathsf{E}f(B^K)$ and $\mathsf{E}f(\sum_K w^K B^K)$. In particular, the means — $\mathsf{E}(B_i)$, $\mathsf{E}(B^K)$ and $\mathsf{E}(\sum_K w^K B^K)$ — are decreasing and convex (in s_i and in $\mathbf{s}$).

Proof. That B_i is decreasing and convex in s_i follows from $B_i = [X_i - s_i]^+$, since $[x]^+$ is increasing and convex in x. From (1), the decreasing convexity of B_i^K follows from the decreasing convexity of B_i.

Since max and + both preserve monotonicity and convexity these properties (decreasingness and convexity) apply to B^K and $\sum_K w^K B^K$ as well. Similarly, since these properties are preserved by any increasing and convex function, the last statement follows. $\square$

Note that in the above results, the convexity of B_i is in the sense of strong stochastic convexity as defined in Shanthikumar and Yao (1991); whereas the convexity of B_i^K is in the sense of "sample path" (which is weaker than the "strong" version); refer to Shaked and Shanthikumar (1994); also see Chang *et al.* (1994), Example 5.3.8.

Proposition 2 For any convex function $f(x)$, $\mathsf{E}f[B_i^K - \mathsf{E}(B_i^K)]$ is decreasing in s_i. In particular, $\mathsf{Var}(B_i^K)$ is decreasing in s_i, for each i and each $K \ni i$; and $\mathsf{Var}(B_i)$ is decreasing in s_i, for all i.

Proof. We first show that $\mathsf{Var}(B_i)$ is decreasing in s_i. Recall, $B_i = [X_i - s_i]^+$. Hence, it suffices to show

$$\begin{aligned} & \mathsf{E}[(X_i - s_i)^+]^2 - \mathsf{E}[(X_i - s_i - 1)^+]^2 \\ \geq \; & [\mathsf{E}(X_i - s_i)^+]^2 - [\mathsf{E}(X_i - s_i - 1)^+]^2. \end{aligned} \tag{4}$$

We have,

$$\begin{aligned} (X_i - s_i)^+ - (X_i - s_i - 1)^+ &= \mathbf{1}(X_i \geq s_i + 1), \\ (X_i - s_i)^+ + (X_i - s_i - 1)^+ &= (2X_i - 2s_i + 1)\mathbf{1}(X_i \geq s_i + 1). \end{aligned}$$

(Note that we are treating X_i's, as well as s_i's as integer valued.) Hence,

$$\begin{aligned} & \mathsf{E}\{[(X_i - s_i)^+]^2 - [(X_i - s_i - 1)^+]^2\} \\ = \; & \mathsf{E}[(2X_i - 2s_i + 1)\mathbf{1}(X_i \geq s_i + 1)]; \end{aligned}$$

where as

$$
\begin{aligned}
& [\mathsf{E}(X_i - s_i)^+]^2 - [\mathsf{E}(X_i - s_i - 1)^+]^2 \\
= \ & [\mathsf{E}(X_i - s_i)^+ + \mathsf{E}(X_i - s_i - 1)^+][\mathsf{E}(X_i - s_i)^+ - \mathsf{E}(X_i - s_i - 1)^+] \\
= \ & \mathsf{E}[(2X_i - 2s_i + 1)\mathbf{1}(X_i \geq s_i + 1)]\mathsf{P}(X_i \geq s_i + 1).
\end{aligned}
$$

Hence, the inequality in (4) holds.

Next, write $B_{i,n}^K$ as B_i^K conditioning upon $X_i - s_i = n$, for $n = 0, 1, \ldots$. From (1), since X_i is independent of the Bernoulli variables involved, it suffices to show that $\mathsf{E}f[B_{i,n}^K - \mathsf{E}(B_{i,n}^K)]$ is increasing in n.

Denote $M_n := B_{i,n}^K - \mathsf{E}(B_{i,n}^K)$. From (1), clearly, $\{M_n\}$ forms a martingale (e.g., Ross 1996). Hence, the convexity of f implies that $\{f(M_n)\}$ is a submartingale, which, in turn, implies that $\mathsf{E}f(M_n)$ is increasing in n, and hence decreasing in s_i, which is what is desired.

Letting $f(x) = x^2$, we have $\mathsf{E}(M_n^2) = \mathsf{Var}(B_{i,n}^K)$. Note that we can write

$$
\mathsf{Var}(B_i^K) = \mathsf{E}[\mathsf{Var}(B_i^K | X_i)] + \mathsf{Var}[\mathsf{E}(B_i^K | X_i)]. \tag{5}
$$

The first term on the right hand side above is decreasing in s_i, since $\mathsf{Var}(B_{i,n}^K)$ is increasing in n given $X_i - s_i = n$ as argued above. The second term, following (1), is equal to $(\lambda^K/\lambda_i)\mathsf{Var}(B_i)$; hence, it is also decreasing in s_i, from what has already been shown for $\mathsf{Var}(B_i)$. Therefore, we can conclude that $\mathsf{Var}(B_i^K)$ is decreasing in s_i. $\square$

The next result says that, in terms of reducing the expected number of backorders, it pays to have a higher base-stock level for a component that has a longer mean leadtime and a lower unit cost, provided there is also a higher overall demand for this component.

Proposition 3 If for any two items i and j we have $\lambda_j \mathsf{E}[L_j] \geq \lambda_i \mathsf{E}[L_i]$, $\lambda_j \geq \lambda_i$, and $c_j \leq c_i$, then the optimal solution to (3), with equal weights in the objective function, must satisfy $s_j \geq s_i$.

Recall the outstanding orders for item i, X_i, has a Poisson distribution with mean $\lambda_i \mathsf{E}[L_i]$. Using the single-item inventory formula we have

$$
s_i = \lambda_i \mathsf{E}[L_i] + k_i \sqrt{\lambda_i \mathsf{E}[L_i]},
$$

where k_i is the safety factor. Now, suppose items i and j have exactly the same demand characteristics and the same leadtimes, i.e., $\lambda_i = \lambda_j$ and $\mathsf{E}[L_i] = \mathsf{E}[L_j]$. Assume, however, item i is less expensive, i.e., $c_i < c_j$. Then, applying the above proposition we should have $k_i > k_j$. That is, we would like to keep higher safety stock for the cheaper item. This shows that the commonly used equal fractile rule is not effective; the protection level k should not be a constant across all items.

After some reflection, the result is actually quite intuitive. Since both items have the same demand characteristics, they are equally important to the system responsiveness. But the inexpensive item consumes less money, so it is more economical.

Now, consider another interesting case: The two items have the same mean leadtime demand, $\lambda_i \mathsf{E}[L_i] = \lambda_j \mathsf{E}[L_j]$. Also, they have the same cost, $c_i = c_j$. Then, the proposition indicates that the item that has higher aggregate rate ($\lambda_j > \lambda_i$) should have larger safety factor (i.e., $k_j \geq k_i$). This, too, is intuitive. Since both items have the same cost, allocating one more unit to item j would result in bigger chance to fulfill an demand, and hence have bigger effect on improving system responsiveness.

The above proposition is an extension of a similar result in Song and Yao (2002) for assemble-to-order systems with a single product type. (With a single product, $\lambda_i = \lambda_j = \lambda$.) The proof also follows the approach there, with added complexity due to multiproduct types. The main idea is to use the joint likelihood ratio ordering of Shanthikumar and Yao (1991a), along with the rearrangement inequalities in Chang *et al* (1994). Details of the proof are available in a technical report; see Lu, Song and Yao (2003b).

4 The Lower-Bound Approach

Applying Jensen's inequality to (2), we have

$$\mathsf{E}[B^K] \geq \max_{i \in K} \mathsf{E}[B_i^K] = \max_{i \in K} \{ \frac{\lambda^K}{\lambda_i} \mathsf{E}[B_i] \}.$$

Hence,

$$\mathsf{E}[\sum_{K \in \mathcal{K}} w^K B^K] \geq \sum_{K \in \mathcal{K}} \max_{i \in K} \{ \frac{w^K \lambda^K}{\lambda_i} \mathsf{E}[B_i] \}. \tag{6}$$

We now use the lower bound in (6) as a surrogate for the objective function in (3):

$$\begin{array}{lll} (LB) & \min\limits_{\mathbf{s}} & \sum\limits_{K \in \mathcal{K}} \max\limits_{i \in K} \{ \frac{w^K \lambda^K}{\lambda_i} \mathsf{E}[B_i(s_i)] \} \\ & \text{s.t.} & c_1 s_1 + \cdots + c_m s_m \leq C; \qquad \mathbf{s} \geq \mathbf{0}. \end{array} \tag{7}$$

Although the objective function in (LB) is much easier to evaluate (involving the marginal distributions only), its inseparability still presents great challenge in optimization.

4.1 The Stack Structure

Now we further relax the integer requirement in (LB), and treat each s_i as a nonnegative real variable. Let $\mathcal{R}_+^m$ denote the m-dimensional nonnegative real vector space. The relaxed problem is then

$$(LB') \quad \min_{\mathbf{s}} \quad \sum_{K\in\mathcal{K}} \max_{i\in K}\{\frac{w^K\lambda^K}{\lambda_i}\mathsf{E}[B_i(s_i)]\} \qquad (8)$$
$$\text{s.t.} \quad c_1s_1+\cdots+c_ms_m \le C; \qquad \mathbf{s}\ge\mathbf{0}.$$

Obviously, the resulting objective value is a lower bound for (LB), which in turn still a lower bound on the original problem. In the rest of this section we discuss how to solve (LB') efficiently. Once we obtain a solution for (LB′), we round it to the nearest integer solution as an approximate solution for the original problem.

A change of variable will better reveal the structure of (LB'). Since

$$b_i(s_i) = \mathsf{E}[B_i(s_i)] = \mathsf{E}[X_i - s_i]^+$$

is strictly decreasing in $s_i \in \mathcal{R}^1_+$, its inverse function $b_i^{-1}(\cdot)$ is well defined, and is also decreasing and convex. Since $\lim_{s_i\to\infty} b_i(s_i) = 0$, we denote $b_i^{-1}(0) = +\infty$. Hence, for each $i = 1, 2, ..., m$, define

$$z_i := \frac{b_i(s_i)}{\lambda_i}, \qquad \text{or} \qquad s_i = b_i^{-1}(\lambda_i z_i) := h_i(z_i); \qquad (9)$$

and $\mathbf{z} := (z_1, ..., z_m)$. The problem (LB') is equivalent to

$$(LB') \quad \min_{\mathbf{z}} \quad \sum_{K\in\mathcal{K}} w^K\lambda^K \max_{i\in K} z_i \qquad (10)$$
$$\text{s.t.} \quad \sum_i c_ih_i(z_i) \le C; \qquad \mathbf{z}\in\mathcal{R}^m_+.$$

Suppose $\{z_i^*,\ i = 1, \cdots, m\}$ is the optimal solution to (10). Denote $u_K := \max_{i\in K} z_j^*$. Without loss of generality, rename the product types in ascending order of u_K:

$$u_{K_1} \le u_{K_2} \le \cdots.$$

We now argue that z_i^*, for $i \in K_1$, should all have the same value. Suppose this is not the case. That is, there exists $i_0 \in K_1$ such that

$$z_{i_0}^* < u_{K_1} := \max_{j\in K_1} z_j^*.$$

Then, increasing $z_{i_0}^*$ to u_{K_1} will not change the objective value, and at the same time will not violate the constraint because $h_{i_0}(\cdot)$ is a decreasing function.

Similarly, since $u_{K_2} := \max_{j\in K_2} z_j^*$ is the second smallest u_K values, for any $i \in K_2 \setminus K_1$, if $z_i^* < u_{K_2}$, we can increase its value to u_{K_2} without changing the objective value or violating the constraint.

Continuing this argument, we can conclude that the optimal solution $\mathbf{z}^*$ to (10) satisfies a stack-like structure that can be formalized as follows.

Suppose $\mathcal{K} = \{K_1, \ldots, K_n\}$, i.e., there are n product (or, demand) types. Note that we can write the set of all components as

$$\mathcal{I} = \cup_{\ell=1}^{n} K_\ell := \{1, ..., m\}.$$

Consider a particular order (permutation) of the n subsets, $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$. Form n new subsets inductively as follows:

$$K^*_{\pi_1} = K_{\pi_1}, \qquad K^*_{\pi_\ell} = K_{\pi_\ell} \backslash (K^*_{\pi_1} \cup \cdots \cup K^*_{\pi_{\ell-1}}), \quad \ell = 2, ..., n. \tag{11}$$

Then, clearly, $(K^*_{\pi_1}, \ldots, K^*_{\pi_n})$ is a partition of $\mathcal{I}$ (whereas $(K_{\pi_1}, \ldots, K_{\pi_n})$ is not).

Proposition 4 There exists an optimal solution $\mathbf{z}^*$ to (10) satisfying the following stack-like structure. Suppose $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ is a permutation such that $\max_{i \in K_{\pi_\ell}} z^*_j$ is increasing in ℓ. Denote $u_{\pi_\ell} := \max_{j \in K_{\pi_\ell}} z^*_j$, and hence

$$u_{K_{\pi_1}} \le u_{K_{\pi_2}} \le \cdots \le u_{K_{\pi_n}}.$$

Let $(K^*_{\pi_1}, \ldots, K^*_{\pi_n})$ be the partition of $\mathcal{I}$ specified in (11). Then, we have,

$$z^*_j = u_{K_{\pi_\ell}}, \qquad \forall\, j \in K^*_{\pi_\ell}$$

for all $\ell = 1, ..., n$.

4.2 The Optimization Algorithm

To simplify notation, denote, for each permutation $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$,

$$\lambda^{\pi_\ell} := \lambda^{K_{\pi_\ell}} \qquad w^{\pi_\ell} := w^{K_{\pi_\ell}} \qquad u_{\pi_\ell} := u_{K_{\pi_\ell}}.$$

Furthermore, based on Proposition 4.1, denote

$$g_\ell(z) := \sum_{i \in K^*_{\pi_\ell}} c_i h_i(z).$$

We can then rewrite the optimization problem in (10) as follows.

$$(LB') \quad \min_{\pi} \quad \min \sum_{\ell=1}^{n} w^{\pi_\ell} \lambda^{\pi_\ell} u_{\pi_\ell} \tag{12}$$

$$\text{s.t.} \quad \sum_{\ell=1}^{n} g_\ell(u_{\pi_\ell}) \le C, \tag{13}$$

$$u_{\pi_1} \le \cdots \le u_{\pi_n}, (u_1, \cdots, u_n) \in \mathcal{R}^n_+.$$

Hence, given the permutation π, denote the inner minimization part of the above problem, including the constraints, as (LB'_π). We can solve (LB'_π) for all π; the one that yields the smallest objective value is then the optimal solution to the original problem (LB').

For each π, (LB'_π) is greedily solvable, via a marginal allocation algorithm. To describe this algorithm, reformulate the problem with yet another transformation of variables:

$$y_1 := u_{K_{\pi_1}}; \qquad y_\ell := u_{K_{\pi_\ell}} - u_{K_{\pi_{\ell-1}}}, \quad \ell = 2, ..., n.$$

Hence,

$$u_{K_{\pi_\ell}} = y_1 + \cdots + y_\ell, \quad \ell = 2, ..., n.$$

Also denote $\mathbf{y} := (y_1, \ldots, y_n)$;

$$v_\ell := u_{K_{\pi_1}} + \cdots + u_{K_{\pi_\ell}} \qquad \ell = 2, ..., n.$$

Then, (LB'_π) becomes the following:

$$(LB'_\pi) \quad \min_{\mathbf{y}} \quad \sum_{\ell=1}^{n} v_\ell y_\ell \tag{14}$$
$$\text{s.t.} \quad \sum_{\ell=1}^{n} g_\ell(y_1 + \cdots + y_\ell) \leq C, \qquad \mathbf{y} \in \mathcal{R}^n_+.$$

The marginal allocation scheme works as follows: Start with $\mathbf{y} = \mathbf{0}$; hence, the zero objective value. In each step, allow an increase of $\Delta > 0$ of the objective value, with Δ a prespecified constant. This corresponds to allowing y_k to increase by an amount Δ/a_k, for each $k = 1, ..., n$, which, in turn, corresponds to the decrease of the left hand side of the constraint by the following amount:

$$\sum_{\ell=1}^{n} [g_\ell(y_1 + \cdots + y_k + \cdots + y_\ell) - g_\ell(y_1 + \cdots + y_k + \Delta/a_k + \cdots + y_\ell)].$$

(Recall, $g_\ell(\cdot)$ is a decreasing function, for all ℓ.) Therefore, select the k that yields the largest such decrease, and update y_k to $y_k + \Delta/a_k$. Repeat this procedure until the left hand side of the constraint is reduced to C.

4.3 Heuristics

Although for each π, (LB'_π) is greedily solvable, we still need to solve $n!$ such problems. One heuristic is to do pairwise improvement: Start from any permutation, say, $(1, \ldots, n)$. Compare this with all other permutations that result from interchanging 1 with $\ell \neq 1$. This involves solving n marginal allocation problems like (LB'_π) above. Suppose $(\pi_1, 2, ..., n)$ gives the smallest objective value. Next, fix π_1 at the first position, apply the same pairwise interchange to the remaining $n - 1$

elements of the permutation, and so forth. This heuristic requires solving $n(n-1)/2$ (as opposed to $n!$) marginal allocation problems.

An even simpler heuristic is to apply the marginal allocation scheme that solves (LB'_π) directly to the problem (LB') in (12). The details are as follows:

- Start with $\pi = (1, ..., n)$, and set $u_{\pi_\ell} = 0$ for all ℓ.
- Suppose the current solution is (u_{π_ℓ}), corresponding to some permutation $\pi = (\pi_\ell)$. For each $k = 1, ..., n$, do:

  (a) increase u_{π_k} to $u_{\pi_k} + \Delta/\lambda^{\pi_k}$; rearrange the u values, and let π^k denote the resulting permutation;

  (b) evaluate the left hand side of the constraint in (13)

  Select k^* as the one corresponding to the smallest value in (b) above (i.e., the one that yields the largest decrease in the left hand side of the constraint). Update $u_{\pi_{k^*}}$ to $u_{\pi_{k^*}} + \Delta/\lambda^{\pi_{k^*}}$; and π to π^{k^*}.
- Repeat the above until the left hand side of the constraint in (13) is reduced to C.

Remarks. Note that with any given permutation the original discrete lower bound surrogate problem in (7) is a separable discrete convex programming, which can be easily solved by a greedy algorithm. So the heuristic algorithm above can also be applied to solve the problem in (7).

5 The Upper-Bound Approach

We now switch to using the *upper* bound as a surrogate for the objective function in the backorder minimization problem in (3). This is based on a simple inequality: for a set of non-negative variables, $x_i \geq 0$, we have

$$\max_i \{x_i\} \leq a + \sum_i (x_i - a)^+, \tag{15}$$

which holds for any a, and can be directly verified. Hence, minimizing the right hand side above, with respect to a, yields an upper bound for $\max_i\{x_i\}$. Note, however, when $a < 0$, the right hand side becomes

$$a + \sum_i (x_i - a) \geq \sum_i x_i.$$

So, in minimizing the right hand side of (15), we only need to consider $a \geq 0$.

When (x_i) is a vector of non-negative random variables, the upper bound has the advantage of involving only the marginal distributions. Hence, applying this to (2), we have

$$B^K := \max\{B_i^K\} \le a^K + \sum_{i\in K}(B_i^K - a^K)^+, \tag{16}$$

where a^K is a parameter; and hence,

$$\sum_{K\in\mathcal{K}} w^K B^K \le \sum_{K\in\mathcal{K}} w^K [a^K + \sum_{i\in K}(B_i^K - a^K)^+].$$

Consequently, we have

$$\mathsf{E}[\sum_{K\in\mathcal{K}} w^K B^K] \le \sum_{K\in\mathcal{K}} w^K a^K + \sum_{K\in\mathcal{K}} w^K \sum_{i\in K} \mathsf{E}(B_i^K - a^K)^+.$$

Therefore, using the above upper bound as a surrogate objective, and writing $\mathbf{a} := (a^K)_{K\in\mathcal{K}}$ (and $\mathbf{s} := (s_i)_{i\in\mathcal{I}}$ as before), we have the following optimization problem:

$$(UB) \quad \min_{(\mathbf{a},\mathbf{s})} \quad \sum_{K\in\mathcal{K}} w^K a^K + \sum_{K\in\mathcal{K}} w^K \sum_{i\in K} \mathsf{E}(B_i^K - a^K)^+ \tag{17}$$

$$\text{s.t.} \quad c_1 s_1 + \cdots + c_m s_m \le C, \qquad \mathbf{s} \ge \mathbf{0}.$$

In the above problem, minimizing over $\mathbf{a}$ is not as important as minimizing over $\mathbf{s}$, as the former is meant to improve on the upper bound. (In fact, as we shall illustrate below, we could very well forgo the minimization over $\mathbf{a}$ by fixing it at a specific value.) Hence, we propose to use a common (scalar) a for all product type K, and then optimize a. This way, the problem becomes,

$$(UB) \quad \min_{(a,\mathbf{s})} \quad a \sum_{K\in\mathcal{K}} w^K + \sum_{K\in\mathcal{K}} w^K \sum_{i\in K} \mathsf{E}(B_i^K - a)^+ \tag{18}$$

$$\text{s.t.} \quad c_1 s_1 + \cdots + c_m s_m \le C, \qquad \mathbf{s} \ge \mathbf{0}.$$

For fixed a, (UB) is a separable convex programming, which is greedily solvable. We can then conduct a line search on a. Also note that taking derivative on the objective with respect to a, we have

$$\sum_{K\in\mathcal{K}} w^K - \sum_{K} w^K \sum_{i\in K} \mathsf{P}[X_i > s_i + a].$$

When the service level is high, which is the case in most applications, $\mathsf{P}[X_i > s_i + a]$ is low, and hence the above derivative will be positive; i.e., it is optimal to have $a = 0$.

Another approach is the following. Since the upper bound in (16) holds for any a^K, we can pick a set of such parameters, a_i, $i \in K$:

$$B^K \leq a_i + \sum_{i\in K}[B_i^K - a_i]^+, \qquad i \in K.$$

In particular, selecting $a_i = \mathsf{E}(B_i^K)$, we have:

$$B^K \leq \max_{i\in K} \mathsf{E}(B_i^K) + \sum_{i\in K}[B_i^K - \mathsf{E}(B_i^K)]^+.$$

Hence, we have the following problem:

$$(UB') \quad \min_{\mathbf{s}} \quad \sum_K w^K \Big\{ \max_{i\in K}[\frac{\lambda^K}{\lambda_i}\mathsf{E}B_i(s_i)] + \sum_{i\in K} \mathsf{E}[B_i^K(s_i) - \mathsf{E}B_i^K(s_i)]^+ \Big\} \tag{19}$$

$$\text{s.t.} \quad c_1 s_1 + \cdots + c_m s_m \leq C, \qquad \mathbf{s} \geq \mathbf{0}.$$

We can go one step further to change the sum over i in the second part of the objective in (19) to max over i. This might result in an objective that is no longer an upper bound; however, numerical results have demonstrated that this serves as a very good approximation of the objective value.

$$(UB'') \quad \min_{\mathbf{s}} \quad \sum_K w^K \max_{i\in K} \Big\{ \frac{\lambda^K}{\lambda_i}\mathsf{E}B_i(s_i)] + \mathsf{E}[B_i^K(s_i) - \mathsf{E}B_i^K(s_i)]^+ \Big\} \tag{20}$$

$$\text{s.t.} \quad c_1 s_1 + \cdots + c_m s_m \leq C, \qquad \mathbf{s} \geq \mathbf{0}.$$

Applying Proposition 2, we know, like $\mathsf{E}B_i(s_i)$, $\mathsf{E}[B_i^K(s_i) - \mathsf{E}B_i^K(s_i)]^+$ is also decreasing in s_i, since $[x]^+$ is a convex function. Hence, in (UB'') we have a problem of the same structure as the lower-bound problem in the last section, hence same optimization and heuristic algorithms can be applied.

6 Extensions to Batch Demand

We now extend our model to allow batch demand. Suppose each arrival of type K requires Q_i^K units of component i. In Lu *et al.* (2002) the probability generating function is derived for the joint distribution of outstanding orders in steady state, $(X_1, \ldots, X_m)$:

$$\psi(z_1, ..., z_m) := \mathsf{E}[\prod_{j=1}^m z_j^{X_j}]$$

$$= \exp\left[\sum_{K\in\mathcal{K}} \lambda^K \int_0^\infty (\psi^{Q^K}(G_1(u) + z_1 G_1^c(u), ..., G_m(u) + z_m G_m^c(u)) - 1)du\right], \tag{21}$$

where $\psi^{Q^K}(z_1, ..., z_m)$ denotes the generating function of $Q^K = (Q_i^K)_{i \in K}$.

Based on the above, the marginal distribution of X_i, for each component i, can be derived; and hence, the expected number of backordered component $\mathsf{E}(B_i) = \mathsf{E}[X_i - s_i]^+$. For instance, the first two moments of X_i are:

$$\mathsf{E}[X_j] = \frac{\partial}{\partial z_j}|_{z_j=1}\psi(z_1, ..., z_m) = \mathsf{E}[L_j] \sum_{K \in \mathcal{K}_j} \lambda^K \mathsf{E}[Q_j^K] \tag{22}$$

and

$$\begin{aligned} \mathsf{E}[X_j^2] &= \left(\frac{\partial^2}{\partial z_j^2} + \frac{\partial}{\partial z_j}\right)|_{z_j=1}\psi(z_1, ..., z_m) \\ &= \mathsf{E}[X_j] + (\mathsf{E}[X_j])^2 + \sum_{K \in \mathcal{K}_j} \lambda^K (\mathsf{E}[(Q_j^K)^2] - \mathsf{E}[Q_j^K]) \int_0^\infty [G_j^c(u)]^2 du. \end{aligned} \tag{23}$$

Recall from (1), B_i^K is the number of backlogged units of component i that are due to type K demand. In the case of single-unit demand, the number of backorders of type K is then simply $B^K = \max_{i \in K} B_i^K$, following (2). When demand arrives in batches, however, we must first translate B_i^K into *batches*:

$$B_i^{Q,K} := \inf\{n : \sum_{m=1}^{n} Q_i^K(m) \geq B_i^K\}, \tag{24}$$

where $Q_i^K(m)$, $m = 1, \ldots, n$, are independent copies of Q_i^K. Then,

$$B^K = \max_{i \in K} B_i^{Q,K}.$$

Here, B_i^K, the number of backlogged *units* of component i due to demand of type K, can be derived as follows:

$$B_i^K = \sum_{k=1}^{B_i} \mathbf{1}_k \left(\frac{\lambda^K \mathsf{E}[Q_i^K]}{\sum_{J \in \mathcal{K}_i} \lambda^J \mathsf{E}[Q_i^J]} \right),$$

where $\mathbf{1}_k(p)$'s are i.i.d. Bernoulli random variables with parameter p, same as in (1). Taking expectations on both sides, and applying Wald's identity (Ross 1996) to the summation on the right hand side, we have

$$\mathsf{E}[B_i^K] = \frac{\lambda^K \mathsf{E}[B_i] \mathsf{E}[Q_i^K]}{\sum_{J \in \mathcal{K}_i} \lambda^J \mathsf{E}[Q_i^J]}. \tag{25}$$

On the other hand, following (24), we have

$$\sum_{m=1}^{B_i^{Q,K}} Q_i^K(m) \geq B_i^K.$$

Applying Wald's identity again, we have

$$\mathsf{E}[Q_i^K]\mathsf{E}[B_i^{Q,K}] \geq \mathsf{E}[B_i^K]. \tag{26}$$

Hence, making use of Jensen's inequality along with (25) and (26), we have:

$$\begin{aligned} \mathsf{E}[B^K] &= \mathsf{E}[\max_{i \in K} B_i^{Q,K}] \geq \max_{i \in K} \mathsf{E}[B_i^{Q,K}] \\ &\geq \max_{i \in K} \frac{\mathsf{E}[B_i^K]}{\mathsf{E}[Q_i^K]} = \max_{i \in K} \left\{ \frac{\lambda^K \mathsf{E}[B_i]}{\sum_{J \in \mathcal{K}_i} \lambda^J \mathsf{E}[Q_i^J]} \right\}. \end{aligned} \tag{27}$$

Using the above lower bound we obtain the counterpart of the surrogate problem in (7), extending the latter to the case of batch demand:

$$\min_{\mathbf{s}} \quad \sum_{K \in \mathcal{K}} w^K \lambda^K \max_{i \in K} \{ \frac{\mathsf{E}[B_i(s_i)]}{\sum_{J \in \mathcal{K}_i} \lambda^J \mathsf{E}[Q_i^J]} \} \tag{28}$$

$$\text{s.t.} \qquad c_1 s_1 + \cdots + c_m s_m \leq C; \qquad \mathbf{s} \geq \mathbf{0}.$$

Note that the objective function in (28) has exactly the same form as the one in (7). Hence, the solution technique in §4 applies.

7 Numerical Studies

Consider an ATO system with six components and the following six product types,

$$\{2,5\},\ \{3,5\},\ \{1,2,5\},\ \{1,3,6\},\ \{1,3,4,5\},\ \{1,3,4,6\}.$$

The overall demand arrival rate is $\lambda = 4$, which splits into the six product types as follows:

$$q^{25} = 0.10,\ q^{35} = 0.40,\ q^{125} = 0.15,\ q^{136} = 0.10,\ q^{1345} = 0.20,\ q^{1346} = 0.05.$$

The component leadtimes follow independent exponential distributions with means

$$(\mathsf{E}[L_1], \mathsf{E}[L_2], ..., \mathsf{E}[L_6]) = (1,1,1,1,2,2).$$

Consider two sets of cost data: $\mathbf{c}^{(1)} = (1,1,1,1,1,1)$ and $\mathbf{c}^{(2)} = (2,2,3,2,1,1)$, with respective weights $\mathbf{w}^{(1)} = (1,1,1,1,1,1)$ and $\mathbf{w}^{(2)} = (1,1,1.2,1.5,1.2,1.5)$, and various inventory budget levels: $C = 20, 24, 32$ for $\mathbf{c}^{(1)}$; and $C = 30, 40, 50$ for $\mathbf{c}^{(2)}$.

In Tables 1 and 2, we summarize the solutions obtained by (i) an exhaustive search via simulation, (ii) the lower-bound approach along with the heuristic algorithm in §4.3, (iii) the upper-bound approach based on (UB), and (iv) the approximation approach based on (UB''). For each approach, we report the optimal base-stock levels, the corresponding objective value, and the resulting weighted average backorder. Note that while the last two measures are equal for the simulation based optimal solution, they are different for the other three approaches. For instance, the objective value corresponding to (ii) is necessarily a lower bound of the objective value of (i); whereas the weighted average backorder corresponding to (ii) is obtained from using the optimal base-stock levels derived from the lower-bound approach (to evaluate analytically the original objective).

In Tables 3 and 4, we repeat the same example with $\lambda = 8$, and modified budget levels: $C = 30, 36, 45$ for $\mathbf{c}^{(1)}$; and $C = 40, 50, 60$ for $\mathbf{c}^{(2)}$.

[INSERT TABLE 1-4 HERE]

The results indicate that solutions from the surrogate problems are very close to the true optimal solution in all cases.

In Table 5 we report the optimal safety factors associated with the optimal base-stock levels in the previous four tables. From these safety factors, we can derive the component fill rates from a normal distribution table. For instance, $k = 1.00$ corresponds to a fill rate of $\Phi(1) = 84.13\%$, where $\Phi(x)$ denotes the distribution function of the standard normal variate; $k = 2$ corresponds to a fill rate of $\Phi(2) = 97.72\%$, and so forth. As can be observed from the table, the optimal safety factors vary widely among components, confirming that the equal fractile rule in general cannot be optimal.

[INSERT TABLE 5 HERE]

Next, we consider a simplified version of the personal computer example studied in Cheng *et al.* (2002). Suppose there are six components that play a key role in differentiating demand types. These are: (1) Ethernet card, (2) standard hard drive (e.g., 7GB disk drive), (3) high profile hard drive (e.g., 13 GB disk drive), (4) video graphics card, (5) standard processor (e.g., 450 MHz board), (6) high profile processor (e.g., 600 MHz). The average component leadtimes and cost data are as follows:

$$(\mathsf{E}[L_i])_{i=1}^6 = (1.7, 3, 3, 1, 2, 2), \qquad (c_i)_{i=1}^6 = (1.0, 2.4, 2.8, 1.4, 2.7, 7.1).$$

Suppose there are four product types, configured as follows:

$$\{2, 5\},\ \{1, 2, 5\},\ \{1, 3, 6\},\ \{1, 3, 4, 6\}.$$

Note that the first two are "low-end" machines, while the last two are "high-end" machines. Consider $\lambda = 4$.

Suppose, over a certain period of time, due to price reduction on the high-end machines, part of the demand on the low-end machines shifts to the high-end machines, resulting in the following four cases:

$$\begin{aligned}
\text{Case1}: &\quad q^{25} = 0.30,\ q^{125} = 0.25,\ q^{136} = 0.30,\ q^{1346} = 0.15; \\
\text{Case2}: &\quad q^{25} = 0.30,\ q^{125} = 0.15,\ q^{136} = 0.30,\ q^{1346} = 0.25; \\
\text{Case3}: &\quad q^{25} = 0.20,\ q^{125} = 0.15,\ q^{136} = 0.35,\ q^{1346} = 0.30; \\
\text{Case4}: &\quad q^{25} = 0.10,\ q^{125} = 0.10,\ q^{136} = 0.40,\ q^{1346} = 0.40.
\end{aligned}$$

In Table 6, we report the optimal base-stock levels corresponding to the above cases for various levels of inventory budget, $80, 90, 100$ and 110. We can observe that the base-stock levels for components 3 and 6, hard drive and processor used in the high-end machines, steadily increase over the period; while the base-stock levels for components 2 and 4, which are associated with low-end machines, steadily decrease. On the other hand, the base-stock level for component 1, which is used in all but one product type, has very little change over the period. These results support the importance of judicious inventory planning when demand shifts among product types. Also observe the ordering among base-stock levels as dictated in Proposition 3. For instance, consider components 2 and 3, which have the same mean leadtime. Since $c_2 < c_3$, and $\lambda_2 < \lambda_3$ in Cases 3 and 4 (where $(\lambda_2, \lambda_3) = \lambda(.35, .65)$ and $\lambda(.30, .80)$, respectively), we have $s_2 \leq s_3$ in these cases. The same holds for components 5 and 6, also in Cases 3 and 4.

[INSERT TABLE 1-4 HERE]

8 Conclusions

The focus of this study is on the inventory-service tradeoff in a multi-product assemble-to-order system — how to allocate a given budget among component inventories so as to optimize customer service, in terms of minimizing the average backorders (or, alternatively, the average order response times) among different product types. We have brought out the importance of backorder as a measurement for customer service, derived analytical expressions for the expected backorder, developed bounds and approximations for it, and used them as surrogate objectives. These result in several optimization problems that exhibit appealing structures and lend themselves to efficient algorithms. Through both analytical and numerical results, we have shown that the model developed here can be used as a tool to garner managerial and operational insights, as well as to support system design.

References

[1] Agrawal, m. and Cohen, M., Optimal Material Control and Performance Evaluation in an Assembly Environment with Component Commonality. *Naval Research Logistics* **48** (2001), 409-429.

[2] Chang, C.S., Shanthikumar, J.G., and Yao, D.D., Stochastic convexity and stochastic majorization, in *Stochastic Modeling and Analysis of Manufacturing Systems* (Chapter 5), D.D. Yao (ed.) Springer-Verlag, New York, 1994.

[3] Cheng, F., Ettl, M., Lin, G.Y., and Yao, D.D., Inventory-Service Optimization in Configure-to-Order Systems, *Manufacturing & Service Operations Management* **4** (2002), 114-132.

[4] Cheung, K.L. and Hausman, W., Multiple Failures in a Multi-Item Spare Inventory Model, *IIE Transactions*, **27** (1995), 171-180.

[5] Glasserman, P. and Wang, Y., Leadtime-Inventory tradeoffs in Assemble-to-Order Systems, *Operations Research*, **46** (1998), 858-871.

[6] Gallien, J. and Wein, L., A Simple and Effective Component Procurement Policy for Stochastic Assembly Systems, *Queueing Systems*, **38** (2001), 221-248.

[7] Hausman, W.H., Lee, H.L. and Zhang, A.X., Order Response Time Reliability in a Multi-Item Inventory System, *European J. of Operational Research*, **109** (1998), 646-659.

[8] Kouvelis, P., Karabati, S. and Yu, G. A Min-Max-Sum Resource Allocation Problem and its Applications, *Operations Research,* **49** (2001), 913-922.

[9] Lu, Y., Song, J.S. and Yao, D. D. Order Fill Rate, Leadtime Variability, and Advance Demand Information in an Assemble-to-Order System. *Operations Research* **51** (2003), March-April, to appear.

[10] Lu, Y., Song, J.S. and Yao, D. D. Backorder Minimization in Multiproduct Assemble-to-Order Systems. Preprint, Graduate School of Management, University of California, Irvine, (2003). Electronic version available at http://www.gsm.uci.edu/∼song/research.htm.

[11] Pine, B.J. *Mass Customization: The New Frontier in Business Competition.* Harvard Business School Press, Boston, 1993.

[12] Ross, S.M., *Stochastic Processes,* 2nd ed., Wiley, New York, 1996.

[13] Shaked, M. and Shanthikumar, J.G., *Stochastic Orders and Their Applications,* Academic Press, New York, 1994.

[14] Shanthikumar, J.G. and Yao, D.D., Strong Stochastic Convexity and Its Applications, *J. Appl. Prob.* **28** (1991), 131-145.

[15] Shanthikumar, J.G. and Yao, D.D., Bivariate Characterization of Some Stochastic Order Relations, *Adv. Appl. Prob.* **23** (1991a), 642-659.

[16] Sherbrooke, C.C., *Optimal Inventory Modeling of Systems*, Wiley, New York, 1992.

[17] Song, J.S., Order-Bases Backorders and Their Implications in Multi-Item Inventory Systems *Management Science* **48** (2002), 499-516.

[18] Song, J.S. and Yao, D.D., Performance Analysis and Optimization in Assemble-to-Order Systems with Random Leadtimes, *Operations Research* **50** (2002), 889-903.

[19] Song, J.S. and Zipkin, P. Supply Chain Operations: Assemble-to-Order Systems, Chapter 11 of *Supply Chain Management*, T. De Kok and S. Graves, eds., in *Handbooks in Operations Research and Management Science*, Vol. XXX, North-Holland. Forthcoming, 2001. Electronic version available at http://www.gsm.uci.edu/∼song/research.htm.

[20] Wang, Y. Near-Optimal Base-Stock Policies in Assemble-to-Order Systems under Service Levels Requirements, preprint, MIT Sloan School, 1999.

[21] Wolff, R., *Stochastic Modeling and the Theory of Queues,* Prentice Hall, Englewood Cliffs, New Jersey, 1989.

Coefficients	*Weight*	*Budget*	*Algorithms*	*s1*	*s2*	*s3*	*s4*	*s5*	*s6*	*Objective*	*Ave. BackOrder*	*Rel. Err. (%)*
(1,1,1,1,1,1)	(1,1,1,1,1,1)	C=20	Optimal	3	2	4	1	8	2	1.4312	1.4312	
			Lower Bound	3	2	3	2	8	2	0.8675	1.5052	5.17
			Upper Bound	3	2	4	1	8	2	2.7470	1.5052	0.00
			Approximation	3	2	4	1	8	2	1.7808	1.4312	0.00
		C=24	Optimal	3	2	5	2	10	2	0.7694	0.7694	
			Lower Bound	3	2	5	2	9	3	0.4097	0.7918	2.91
			Upper Bound	4	2	5	2	9	2	1.7035	0.7859	2.14
			Approximation	3	2	5	2	10	2	0.8620	0.7694	0.00
		C=32	Optimal	5	3	6	3	12	3	0.1857	0.1857	
			Lower Bound	5	3	6	3	11	4	0.0959	0.2037	9.69
			Upper Bound	5	3	6	3	12	3	0.6654	0.1857	0.00
			Approximation	5	3	6	3	12	3	0.2602	0.1857	0.00
(2,2,3,2,1,1)	(1,1,1,1,1,1)	C=30	Optimal	3	2	2	2	8	2	1.5052	1.5052	
			Lower Bound	3	2	2	2	8	2	0.9676	1.5052	0.00
			Upper Bound	3	2	2	2	8	2	3.1790	1.5052	0.00
			Approximation	3	2	2	2	8	2	2.3844	1.5052	0.00
		C=40	Optimal	3	2	4	2	11	3	0.7999	0.7999	
			Lower Bound	3	2	4	2	10	4	0.4357	0.8253	3.18
			Upper Bound	3	2	4	2	11	3	1.7740	0.7999	0.00
			Approximation	3	2	4	2	10	4	0.9352	0.8253	3.18
		C=50	Optimal	4	3	5	3	11	4	0.3188	0.3188	
			Lower Bound	4	3	5	3	11	4	0.1671	0.3188	0.00
			Upper Bound	4	3	5	3	11	4	0.8513	0.3188	0.00
			Approximation	4	3	5	3	11	4	0.3670	0.3188	0.00

Table 1: **System: Six product: q25=0.1,q35=0.4, q125=0.15, q136=0.1,q1345=0.2, q1346=0.05**

Six items: L=(1,1,1,1,2,2)

Overall Arrival rate=4

Coefficients	*Weights*	*Budget*	*Algorithms*	*s1*	*s2*	*s3*	*s4*	*s5*	*s6*	*Objective*	*Ave. BackOrder*	*Rel. Err. (%)*
(1,1,1,1,1,1)	(1,1,1.2,1.5,1.2,1.5)	C=20	Optimal	3	2	4	1	8	2	1.8138	1.8138	
			Lower Bound	3	2	3	2	8	2	0.8942	1.8507	2.03
			Upper Bound	3	2	4	1	8	2	3.1732	2.0296	11.90
			Approximation	3	2	4	1	8	2	2.0001	1.8138	0.00
		C=24	Optimal	4	2	4	2	9	3	0.9976	0.9976	
			Lower Bound	3	3	4	2	9	3	0.4333	1.0606	6.32
			Upper Bound	3	3	5	2	9	2	2.0304	1.0809	8.36
			Approximation	3	2	5	2	10	2	0.8620	1.0139	1.64
		C=32	Optimal	5	3	6	3	11	4	0.2236	0.2236	
			Lower Bound	5	3	6	3	11	4	0.0669	0.2236	0.00
			Upper Bound	5	3	6	3	12	3	0.8942	0.2455	9.83
			Approximation	5	3	6	3	12	3	0.2784	0.2455	9.83
(2,2,3,2,1,1)	(1,1,1.2,1.5,1.2,1.5)	C=30	Optimal	3	2	3	1	7	2	2.4270	2.4270	
			Lower Bound	2	3	3	1	7	2	1.1662	2.5966	6.99
			Upper Bound	2	3	2	2	8	2	3.2222	2.6642	9.77
			Approximation	3	2	2	2	8	2	2.6039	2.4699	1.77
		C=40	Optimal	3	2	5	2	9	2	1.1570	1.1570	
			Lower Bound	3	2	5	2	9	2	0.4507	1.1570	0.00
			Upper Bound	3	2	4	2	11	3	2.0991	1.2087	4.47
			Approximation	3	2	5	2	9	2	1.2384	1.1570	0.00
		C=50	Optimal	4	3	5	3	12	3	0.4097	0.4097	
			Lower Bound	4	4	5	2	12	3	0.1743	0.4634	13.10
			Upper Bound	5	2	5	3	12	3	0.8907	0.4432	8.16
			Approximation	4	3	5	3	12	3	0.4339	0.4097	0.00

Table 2: **System: Six product: q25=0.1,q35=0.4, q125=0.15, q136=0.1,q1345=0.2, q1346=0.05**

Six items: L=(1,1,1,1,2,2)

Overall Arrival rate=4

Coefficients	*Weight*	*Budget*	*Algorithms*	*s1*	*s2*	*s3*	*s4*	*s5*	*s6*	*Objective*	*Ave. BackOrder*	*Rel. Err. (%)*
(1,1,1,1,1,1)	(1,1,1,1,1,1)	C=30	Optimal	4	2	6	2	14	2	3.6992	3.6992	
			Lower Bound	4	2	5	2	13	4	2.1184	3.9141	5.81
			Upper Bound	4	2	6	2	14	2	6.1314	3.6992	0.00
			Approximation	4	2	6	2	14	2	5.2077	3.6992	0.00
		C=36	Optimal	5	3	7	3	15	3	2.1585	2.1585	
			Lower Bound	5	3	6	3	15	4	1.2252	2.2591	4.66
			Upper Bound	5	3	7	3	15	3	3.7673	2.1585	0.00
			Approximation	5	3	7	3	15	3	2.7572	2.1585	0.00
		C=45	Optimal	6	4	9	4	18	4	0.7906	0.7906	
			Lower Bound	6	4	8	4	18	5	0.4027	0.8076	2.15
			Upper Bound	6	4	9	4	18	4	1.7213	0.7906	0.00
			Approximation	6	4	8	4	18	5	0.8303	0.8076	2.15
(2,2,3,2,1,1)	(1,1,1,1,1,1)	C=40	Optimal	3	2	4	2	12	2	5.8115	5.8115	
			Lower Bound	3	2	4	2	12	2	3.1483	5.8115	0.00
			Upper Bound	3	1	3	1	16	3	8.3235	6.0408	3.95
			Approximation	3	2	4	2	12	2	5.9234	5.1901	0.00
		C=50	Optimal	4	2	5	2	16	3	3.6540	3.654	
			Lower Bound	4	2	6	2	13	3	1.8405	3.9222	7.34
			Upper Bound	4	2	5	2	16	3	5.3085	3.654	0.00
			Approximation	4	2	6	2	13	13	4.0001	3.9222	7.34
		C=60	Optimal	5	3	7	3	14	3	2.5787	2.5787	
			Lower Bound	5	3	7	3	15	4	1.0385	2.5787	0.00
			Upper Bound	5	3	6	3	17	3	3.5718	2.6443	2.54
			Approximation	5	3	7	3	14	3	2.5598	2.5787	0.00

Table 3: **System: Six product: q25=0.1,q35=0.4, q125=0.15, q136=0.1,q1345=0.2, q1346=0.05**

Six items: L=(1,1,1,1,2,2)

Overall Arrival rate=8

Coefficients	*Weights*	*Budget*	*Algorithms*	*s1*	*s2*	*s3*	*s4*	*s5*	*s6*	*Objective*	*Ave. BackOrder*	*Rel. Err. (%)*
(1,1,1,1,1,1)	(1,1,1.2,1.5,1.2,1.5)	C=30	Optimal	4	3	5	2	13	3	4.6977	4.6977	
			Lower Bound	3	4	5	2	13	3	2.7525	5.2219	11.16
			Upper Bound	4	3	5	2	14	2	7.1485	4.9309	4.96
			Approximation	3	4	5	2	13	3	5.5713	5.2219	11.16
		C=36	Optimal	4	4	7	3	15	4	2.6205	2.6205	
			Lower Bound	5	4	6	3	15	4	1.5072	2.6381	0.67
			Upper Bound	4	4	7	3	15	4	3.7673	2.6205	0.00
			Approximation	4	4	7	3	15	4	2.8011	2.6205	0.00
		C=45	Optimal	6	4	8	4	18	5	1.0377	1.0377	
			Lower Bound	6	5	8	4	18	4	0.4719	1.0931	5.33
			Upper Bound	6	5	8	4	18	4	1.8007	1.0931	5.33
			Approximation	6	4	8	4	18	5	1.0984	1.0377	0.00
(2,2,3,2,1,1)	(1,1,1.2,1.5,1.2,1.5)	C=40	Optimal	3	2	4	1	15	2	6.3911	6.3911	
			Lower Bound	3	3	4	1	12	3	4.0054	6.6257	3.67
			Upper Bound	3	2	4	2	12	3	9.7655	6.4639	1.14
			Approximation	3	2	4	1	15	2	6.7800	6.3911	0.00
		C=50	Optimal	4	2	5	2	16	3	4.2470	4.2470	
			Lower Bound	4	3	5	2	13	4	2.5485	4.4930	5.79
			Upper Bound	4	2	5	2	16	3	5.3085	4.2470	0.00
			Approximation	4	2	6	2	13	4	4.5020	4.4007	3.62
		C=60	Optimal	5	4	6	3	15	3	2.8692	2.8692	
			Lower Bound	5	4	6	3	15	3	1.4638	2.8692	0.00
			Upper Bound	5	2	6	3	18	4	3.8970	3.0192	5.23
			Approximation	6	3	6	3	15	3	2.9403	2.9007	1.10

Table 4: **System: Six product: q25=0.1,q35=0.4, q125=0.15, q136=0.1,q1345=0.2, q1346=0.05**

Six items: L=(1,1,1,1,2,2)

Overall Arrival rate=8

ArrivalRate	*Coefficients*	*Weight*	*Budget*	*k1*	*k2*	*k3*	*k4*	*k5*	*k6*
4	(1,1,1,1,1,1)	(1,1,1,1,1,1)	20	0.71	1.00	0.58	0.00	0.46	0.73
			24	0.71	1.00	1.15	1.00	1.23	0.73
			32	2.12	2.00	1.73	2.00	1.99	1.64
	(2,2,3,2,1,1)	(1,1,1,1,1,1)	30	0.71	1.00	-0.58	1.00	0.46	0.73
			40	0.71	1.00	0.58	1.00	1.61	2.56
			50	1.41	2.00	1.15	2.00	1.61	2.56
	(1,1,1,1,1,1)	(1,1,1.2,1.5,1.2,1.5)	20	0.71	1.00	0.58	0.00	0.46	0.73
			24	1.41	1.00	0.58	1.00	0.84	1.64
			32	2.12	2.00	1.73	2.00	1.61	2.56
	(2,2,3,2,1,1)	(1,1,1.2,1.5,1.2,1.5)	30	0.71	1.00	0.00	0.00	0.46	0.73
			40	0.71	1.00	0.58	1.00	0.84	1.64
			50	1.41	2.00	1.15	2.00	1.61	1.64
8	(1,1,1,1,1,1)	(1,1,1,1,1,1)	30	0.00	0.00	0.00	0.00	0.11	-0.26
			36	0.50	0.71	0.41	0.71	0.38	0.39
			45	1.00	1.41	1.22	1.41	1.19	1.03
	(2,2,3,2,1,1)	(1,1,1,1,1,1)	40	-0.50	0.00	-0.82	0.00	-0.43	0.39
			50	0.00	0.00	0.00	0.00	0.11	1.03
			60	0.50	0.71	0.41	0.71	0.38	1.03
	(1,1,1,1,1,1)	(1,1,1.2,1.5,1.2,1.5)	30	0.00	0.71	-0.41	0.00	-0.16	0.39
			36	0.00	1.41	0.41	0.71	0.38	1.03
			45	1.00	1.41	0.82	1.41	1.19	1.68
	(2,2,3,2,1,1)	(1,1,1.2,1.5,1.2,1.5)	40	-0.50	0.00	-0.82	-0.71	0.38	-0.26
			50	0.00	0.00	-0.41	0.00	0.65	0.39
			60	1.00	0.71	0.00	0.71	0.38	1.03

Table 5: Optimal safety factor

C=80							
Case	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**E[B]**
1	6	8	5	1	5	4	2.5162
2	5	6	6	2	5	4	2.8598
3	6	5	7	2	3	5	3.1933
4	5	3	8	2	2	6	3.9448
C=90							
Case	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**E[B]**
1	6	8	6	1	6	4	2.0028
2	6	6	7	2	5	5	2.2392
3	6	5	8	2	4	5	2.6248
4	6	3	9	2	2	6	3.2930
C=100							
Case	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**E[B]**
1	7	9	7	1	6	5	1.2669
2	7	7	8	2	5	5	1.6107
3	7	5	8	2	4	6	2.1991
4	7	3	10	2	2	7	2.5152
C=110							
Case	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**E[B}**
1	7	10	7	2	7	5	0.9597
2	7	8	8	2	6	6	1.1432
3	7	6	9	2	4	7	1.5371
4	7	4	11	3	3	8	1.5954

Table 6. Optimal base-stock level with changes in demand rates