Manufacturing & Service Operations Management

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

# Inventory-Service Optimization in Configure-to-Order Systems

Feng Cheng, Markus Ettl, Grace Lin, David D. Yao,

To cite this article:
Feng Cheng, Markus Ettl, Grace Lin, David D. Yao, (2002) Inventory-Service Optimization in Configure-to-Order Systems. Manufacturing & Service Operations Management 4(2):114-132. http://dx.doi.org/10.1287/msom.4.2.114.282

# Inventory-Service Optimization in Configure-to-Order Systems

Feng Cheng • Markus Ettl • Grace Lin • David D. Yao*
*IBM Research Division, T.J. Watson Research Center, Yorktown Heights, New York 10598*
*IBM Research Division, T.J. Watson Research Center, Yorktown Heights, New York 10598*
*IBM Research Division, T.J. Watson Research Center, Yorktown Heights, New York 10598*
*IEOR Department, 302 Mudd Building, Columbia University, New York, New York 10027*
*fcheng@us.ibm.com • msettl@us.ibm.com • gracelin@us.ibm.com • yao@ieor.columbia.edu*

This study is motivated by a process-reengineering problem in personal computer (PC) manufacturing, i.e., to move from a build-to-stock operation that is centered around end-product inventory towards a configure-to-order (CTO) operation that eliminates end-product inventory. In fact, CTO has made irrelevant the notion of preconfigured machine types and focuses instead on maintaining the right amount of inventory at the components. CTO appears to be the ideal operational model that provides both mass customization and a quick response time to order fulfillment. To quantify the inventory-service trade-off in the CTO environment, we develop a nonlinear optimization model with multiple constraints, reflecting the service levels offered to different market segments. To solve the optimization problem, we develop an exact algorithm for the important case of demand in each market segment having (at least) one unique component, and a greedy heuristic for the general (nonunique component) case. Furthermore, we show how to use sensitivity analysis, along with simulation, to fine-tune the solutions. The performance of the model and the solution approach is examined by extensive numerical studies on realistic problem data. We present the major findings in applying our model to study the inventory-service impacts in the reengineering of a PC manufacturing process.
(*Supply Chain Optimization; Inventory/Production; Assembly Systems; Configure-to-Order; Postponement; Stochastic Models*)

## 1. Introduction

A configure-to-order (CTO) system is a hybrid of make-to-stock and make-to-order operations: A set of components (subassemblies) are built to stock, whereas the end products are assembled to order. This hybrid model is most suitable in an environment where the time it takes to assemble the end product is negligible, while the production/replenishment leadtime for each component is much more substantial. By keeping inventory at the component level, customer orders can be filled quickly. On the other hand, postponing the final assembly until order arrival provides a high level of flexibility in terms of product variety, and also achieves resource pooling in terms of maximizing the usage of component inventory. Therefore, the CTO system is an ideal business process model that provides both mass customization and a quick response time to order fulfillment.

Such a hybrid model is often referred to as an assemble-to-order (ATO) system in the research literature, e.g., Glasserman and Wang (1998) and Song et al. (1999). In a standard ATO system, usually there is a (small) set of preconfigured end products from

*Research undertaken while an academic visitor at IBM Research Division, T.J. Watson Research Center.

Manufacturing & Service Operations Management © 2002 INFORMS
Vol. 4, No. 2, Spring 2002, pp. 114–132
1523-4614/02/0402/0114$05.00
1526-5498 electronic ISSN

which customers must choose. Unlike an ATO system, a CTO system allows each customer to configure an end product by selecting a personalized subset of components which may be ordered in any arbitrary multiples. The manufacturing process of personal computers (PCs) is a good example of such an environment: Each end product can be configured with a hard drive (size to be determined), processor (speed to be determined), system memory (amount to be determined), display (panel size and resolution to be determined), optical drive (DVD/CD-RW, CD-RW, etc.), and operating system. This means that each end product will comprise a different set of multiples of the same components combined to satisfy customer requirements.

Aside from some consistency check on the product so configured, there is no prespecified set of end products that limits the customer's choice, and so studying a CTO system from an end-product perspective is neither useful nor practical. For instance, suppose any subset of $n$ components is a possible end-product configuration. In terms of data collection and forecasting, one would then have to deal with the intractable task of forecasting $2^n$ demand streams. The CTO model we develop here formalizes what has become prevailing industry practice, i.e., to forecast aggregate demands (say, demands for particular market segments) along with the component usages associated with such demands. When there are $m$ market segments, forecasting amounts to an effort of order $O(mn)$, including the forecasts of component usages.

The study reported here is part of a larger project that aims at helping IBM's Personal Computing Division (PCD) to migrate from its existing operation to a CTO operation. The manufacturing at PCD is traditionally a build-to-stock process, a process that is internally referred to as the MTM (''machine-type/model'') operation. It is based on a set of end products, or MTMs. Demand forecasts are generated for each MTM over a future planning horizon. A material requirements planning (MRP) type of explosion technique is then used to determine the component requirements over the planning horizon, based on the bill-of-materials (BOM) structure of each MTM. Because of the random variation involved in the demand forecasts, finished goods inventory is held for each MTM in order to meet a desirable customer service level. However, holding finished goods inventory over any period of time is very costly in the PC business because product life cycles are measured in months, and price reductions take place almost every other week.

To move from this business process to a Web-based CTO operation where customer orders are taken from the Internet, finished goods inventory will be eliminated altogether, and emphasis will be shifted to the components. Because of their long leadtimes, the components will be powered off a forecast and executed off a replenishment process. The focus of our study is on the inventory-service trade-off of the new system, and on the performance gains in terms of reduced inventory investment and increased service level. There are other benefits as well: The CTO operation can achieve better forecast accuracy through demand aggregation. Customer demand is expected to increase, as orders will no longer be confined within a restricted set of preconfigured MTMs. The optimization model we develop below provides an analytical tool to quantify these performance impacts.

A brief review of the related literature is in order. There are many studies of ATO systems that differ quite substantially in the detailed modeling assumptions and approaches. For example, Hausman et al. (1998) and Zhang (1997) study periodic-review (discrete-time) models with multivariate normal demand and constant component replenishment leadtimes. Song (forthcoming, 1998) studies continuous-review models with multivariate compound Poisson demand and deterministic leadtimes. Song et al. (1999), Glasserman and Wang (1998), and Wang (1988) also consider multivariate (compound) Poisson demand, but the supply process for each component is capacitated and modeled as a single-server queue. Gallien and Wein (2001) consider uncapacitated leadtimes, focusing on a single demand stream and assuming order synchronization. Cheung and Hausman (1995) also assume uncapacitated leadtimes in the context of a repair shop. They use a combination of order synchronization and disaggregation in their analysis.

In terms of approaches, Song (forthcoming, 1998) and Song et al. (1999) focus on developing exact and approximate performance evaluation procedures that are computationally efficient. Xu (2001) explores the dependence structure in several ATO systems using stochastic comparison techniques, and develops bounds on key performance measures. Glasserman and Wang (1998) study the inventory-leadtime tradeoffs using asymptotics of tail distributions, and derive a linear relationship between leadtime and inventory in the limiting sense of high fill rates; see also Glasserman (1999). Wang (1988) further applies this asymptotic result in an optimization problem to minimize average inventory holding cost with a constraint on the order fill rate; also refer to Wang (2001). Swaminathan and Tayur (1999) use stochastic programming models to study three different strategies at the assembly stage: utilizing component commonality, postponement (the ''vanilla box approach''), and integrating assembly task design and operations. Other related recent works, not necessarily in the ATO setting, include Aviv and Federgruen (1999), Dong and Lee (2001), Garg and Lee (1999), Li (1992), Mahajan and van Ryzin (1999), Vandaele and Lambrecht (2001), and Zipkin (2000).

Whereas most studies in the literature focus on certain segments of the supply chain, modeled as simple stand-alone queues, the recent work of Ettl et al. (2000) aims at modeling large-scale, end-to-end enterprise supply chains, such as those in the PC industry. In contrast, the model we develop in this paper is simpler in the network configuration—there are only two levels of BOM: the components and the end products; but more demanding in service requirement—the fill rate of each product is essentially the off-the-shelf availability of all its components. To deal with this more stringent service requirement, we use a lower bound as a surrogate for the analytically intractable fill rate. This treatment is essentially in the same spirit as the approach in Connors and Yao (1996), Song et al. (1999), and Song and Yao (forthcoming). The novelty of our approach here is to exploit the simpler BOM structure to come up with algorithms that solve the optimization problem in a more efficient manner than the gradient search in Ettl et al. (2000).

The rest of the paper is organized as follows. In the next three sections we present details of our model: the given data required as input to the model (§2), the base-stock control policy that we focus on (§3), and the modeling of the customer service requirement (§4). The optimization problem and the algorithms that solve the problem are presented in §5. These are followed by §6, where the performance of the model and the solution technique is examined in detail. In §7, we present the major findings in applying our model to study the inventory/service impacts in reengineering PCD's business process. Brief concluding remarks are presented in §8.

## 2. The Model and Given Data

We consider a hybrid model, by which each end product is assembled to order from a set of components, which in turn are built to stock. In other words, no finished goods inventory is kept for any end product, whereas each component (''building block'') has its own inventory, replenished from a supplier following a base-stock policy.

Each component inventory is indexed by $i$, $i \in \mathcal{S}$, where $\mathcal{S}$ denotes the set of all components. Associated with each component is a ''store,'' where the inventory is kept.

In the CTO environment, there is no prespecified product ''menu''; in principle, every order can require a distinct set of components. Let $\mathcal{M}$ denote the set of product/demand families that use the same set of components. For instance, $\mathcal{M}$ = {low-end machines, high-end machines, servers}; or $\mathcal{M}$ = {individuals, small business, corporations}. (Also, refer to Figure 1 for a numerical example considered in §6.)

Time is discrete, indexed by $t$, with each time unit called a period. Let $D_m(t)$ denote the demand associated with product family $m$ in period $t$. Each order of type $m$ requires a random number of units from component $i$, denoted as $X_{mi}$, which takes on nonnegative integer values. Denote:

$$\mathcal{S}_m := \mathcal{S} - \{i\colon X_{mi} \equiv 0\}, \qquad \mathcal{M}_i := \mathcal{M} - \{m\colon X_{mi} \equiv 0\}.$$

That is, $\mathcal{S}_m$ denotes the set of components used in type $m$ products, whereas $\mathcal{M}_i$ denotes all the product fam-

ilies that use component ***i***. (Here, $X_{mi} \equiv 0$ means $P(X_{mi} = 0) = 1$.)

There are two kinds of leadtimes: those associated with the components, and those associated with the end products:

- $L_i^{\text{in}}$, $i \in \mathcal{S}$: the inbound leadtime—the time for the supplier of component $i$ to replenish to store $i$ once an order is placed. Assume this leadtime is known through a given distribution, for instance, a normal distribution with mean and variance given.
- $L_m^{\text{out}}$, $m \in \mathcal{M}$: the outbound leadtime—the time to supply a customer demand of type $m$, provided there is no stockout of any component $i \in \mathcal{S}_m$. This time includes the order processing time, the assembly/reconfiguration time, and the transportation time to deliver the order. The distribution of this leadtime is also assumed known.

The first step in our analysis is to translate the end-product demand into demand for each component $i$:

$$D_i(t) = \sum_{m \in \mathcal{M}_i} \sum_{k=1}^{D_m(t+L_m^{\text{out}})} X_{mi}(k), \tag{1}$$

where $X_{mi}(k)$, for $k = 1, 2, \ldots$, are i.i.d. copies of $X_{mi}$, and the inclusion of $L_m^{\text{out}}$ in the end-product demand is a standard MRP type of demand-leadtime offset. Assume the mean and the variance of $X_{mi}$ are known. For instance, these can be derived from empirical demand data. Applying Wald's identity (refer to, e.g., Ross 1998) and conditioning on $L_m^{\text{out}}$, we can then derive the mean and the variance of $D_i(t)$:

$$\begin{aligned} \mathrm{E}[D_i(t)] &= \sum_{m \in \mathcal{M}_i} \mathrm{E}(X_{mi})\mathrm{E}[D_m(t + L_m^{\text{out}})] \\ &= \sum_{m \in \mathcal{M}_i} \mathrm{E}(X_{mi}) \sum_l \mathrm{E}[D_m(t + l)]\mathrm{P}[L_m^{\text{out}} = l], \end{aligned} \tag{2}$$

and

$$\begin{aligned} \mathrm{Var}[D_i(t)] &= \sum_{m \in \mathcal{M}_i} \{\mathrm{E}[D_m(t + L_m^{\text{out}})]\mathrm{Var}(X_{mi}) \\ &\qquad + \mathrm{Var}[D_m(t + L_m^{\text{out}})]\mathrm{E}^2(X_{mi})\} \\ &= \sum_{m \in \mathcal{M}_i} \mathrm{Var}(X_{mi}) \sum_l \mathrm{E}[D_m(t + l)]\mathrm{P}[L_m^{\text{out}} = l] \\ &\quad + \sum_{m \in \mathcal{M}_i} \mathrm{E}^2(X_{mi}) \sum_l \mathrm{E}[D_m^2(t + l)]\mathrm{P}[L_m^{\text{out}} = l] \\ &\quad - \sum_{m \in \mathcal{M}_i} \mathrm{E}^2(X_{mi}) \left( \sum_l \mathrm{E}[D_m(t + l)]\mathrm{P}[L_m^{\text{out}} = l] \right)^2. \end{aligned} \tag{3}$$

The variance calculation above assumes the independence of the demands across product families. If these are correlated, the above derivation can be modified without essential difficulties by incorporating the covariance terms. Also note that in applying Wald's identity to (1) certain dependence between $X_{mi}$ and $D_m$ is allowed.

## 3. Base-Stock Control

As mentioned earlier, each store $i$ follows a base-stock policy. Let $R_i(t)$ denote the base-stock level at store $i$ in period $t$, i.e., on-hand inventory plus any outstanding orders (on-order position). Let $I_i(t)$ and $B_i(t)$ denote the inventory and backorder levels, respectively, in period $t$. A combination of standard inventory theory (e.g., Zipkin 2000) and queueing analysis (Ettl et al. 2000) yields the following results. Denote:

$$D_i(t, t + l) := D_i(t) + D_i(t + 1) + \cdots + D_i(t + l),$$

and

$$\mu_i(t, t + l) = \sum_{s=t}^{t+l} \mathrm{E}[D_i(s)], \tag{4}$$

$$\sigma_i^2(t, t + l) = \sum_{s=t}^{t+l} \mathrm{Var}[D_i(s)]. \tag{5}$$

Assuming that demands follow normal distributions, we write

$$D_i(t, t + l) = \mu_i(t, t + l) + Z \cdot \sigma_i(t, t + l),$$

where $Z$ denotes the standard normal variate; and write

$$R_i(t) = \mu_i(t, t + l_i^{\text{in}}) + k_i(t) \cdot \sigma_i(t, t + l_i^{\text{in}}), \tag{6}$$

where $l_i^{\text{in}} := \mathrm{E}[L_i^{\text{in}}]$ is the expected inbound leadtime, and $k_i(t)$ is often referred to as the *safety factor*. Then,

$$I_i(t) = [R_i(t) - D_i(t, t + l_i^{\text{in}})]^+ \quad \text{and}$$

$$B_i(t) = [D_i(t, t + l_i^{\text{in}}) - R_i(t)]^+. \tag{7}$$

(Note that because time is discrete, we shall round *up* any real-valued $l_i^{\text{in}}$ to the next integer.) Furthermore, recall the following standard function in inventory theory (e.g., Zipkin 2000):

$$G(x) := \mathrm{E}[Z - x]^+ = \int_x^\infty (z - x)\phi(z)\,dz$$
$$= \phi(x) - x\bar{\Phi}(x), \tag{8}$$

with $Z$ denoting the standard normal variate, $\phi$ and $\Phi$ denoting, respectively, the density function and the distribution function of $Z$, and $\bar{\Phi}(x) := 1 - \Phi(x)$. A related function is:

$$H(x) := \mathrm{E}[x - Z]^+ = x + G(x) = \phi(x) + x\Phi(x). \tag{9}$$

We can then derive:

$$\mathrm{E}[I_i(t)] = \mathrm{E}[R_i(t) - D_i(t, t + l_i^{\mathrm{in}})]^+$$
$$= \sigma_i(t, t + l_i^{\mathrm{in}})H(k_i(t)); \tag{10}$$
$$\mathrm{E}[B_i(t)] = \mathrm{E}[R_i(t) - D_i(t, t + l_i^{\mathrm{in}})]^+$$
$$= \sigma_i(t, t + l_i^{\mathrm{in}})G(k_i(t)); \quad \text{and} \tag{11}$$
$$\mathrm{P}[I(t) = 0] = \mathrm{P}[D_i(t, t + l_i^{\mathrm{in}}) \geq R_i(t)]$$
$$= \mathrm{P}[Z \geq k_i(t)] = \bar{\Phi}(k_i(t)). \tag{12}$$

To facilitate implementation, it is often desirable to translate $R_i(t)$ into "days of supply" (DOS), or more precisely, periods of supply. To do so, note that the $\mu_i$ part of $R_i(t)$ simply translates into $l_i^{\mathrm{in}}$ periods (up to $t$) of demand. In addition, we can turn the safety-stock part of $R_i(t)$ into

$$\frac{k_i(t)\sigma_i(t, t + l_i^{\mathrm{in}})}{\mu_i(t, t + l_i^{\mathrm{in}})/l_i^{\mathrm{in}}}$$

periods of demand. Hence, we can express $R_i(t)$ in terms of periods of demand, or DOS, as follows:

$$\mathrm{DOS}_i(t) = l_i^{\mathrm{in}}\left[1 + k_i(t)\frac{\sigma_i(t, t + l_i^{\mathrm{in}})}{\mu_i(t, t + l_i^{\mathrm{in}})}\right]. \tag{13}$$

Note the intuitively appealing form of (13), in particular the safety-stock (or rather, *safety time*) part, which is equal to the product of the safety factor and the coefficient of variation (i.e., the ratio of standard deviation to mean) of the component demand over the (inbound) leadtime.

Next, suppose demand is stationary, i.e., for each product family $m$, $D_m(t)$ is invariant in distribution over time. Then, (4) and (5) reduce to the following (omitting the time arguments):

$$\mu_i = l_i^{\mathrm{in}}\mathrm{E}[D_i] \quad \text{and} \quad \sigma_i^2 = l_i^{\mathrm{in}}\mathrm{Var}[D_i]. \tag{14}$$

We can then write

$$R_i = l_i^{\mathrm{in}}\mathrm{E}[D_i] + k_i\sqrt{l_i^{\mathrm{in}}}\,\mathrm{sd}[D_i]; \tag{15}$$

and hence,

$$\mathrm{DOS}_i = \frac{R_i}{\mathrm{E}[D_i]} = l_i^{\mathrm{in}} + k_i\theta_i\sqrt{l_i^{\mathrm{in}}} = l_i^{\mathrm{in}}\left[1 + k_i\frac{\theta_i}{\sqrt{l_i^{\mathrm{in}}}}\right], \tag{16}$$

where $\theta_i := \mathrm{sd}[D_i]/\mathrm{E}[D_i]$ is the coefficient of variation of the demand *per period* for component *i*. Hence $\theta_i/\sqrt{l_i^{\mathrm{in}}}$ is the coefficient of variation of the demand over the leadtime $l_i^{\mathrm{in}}$, which is consistent with the general formula in (13).

Sometimes it is more appropriate to adjust the demand distribution to account for nonnegativity. Specifically, instead of $D = \mu + \sigma Z$, where $Z$ is the standard normal variate, we should have $\tilde{D} = [\mu + \sigma Z]^+$. The adjusted mean follows from (8):

$$\mathrm{E}[\tilde{D}] = \sigma\mathrm{E}\left[Z - \left(-\frac{\mu}{\sigma}\right)\right]^+ = \sigma G\left(-\frac{\mu}{\sigma}\right). \tag{17}$$

To derive the adjusted variance, note the following:

$$\mathrm{E}\{[(Z - x)^+]^2\} = \int_x^\infty (z - x)^2\phi(z)\,dz$$
$$= x\phi(x) + \bar{\Phi}(x) - 2x\phi(x) + x^2\bar{\Phi}(x)$$
$$= \bar{\Phi}(x) - xG(x),$$

where the last equation makes use of (8). Hence,

$$\mathrm{Var}[\tilde{D}] = \sigma^2\mathrm{Var}\left\{\left[Z - \left(-\frac{\mu}{\sigma}\right)\right]^+\right\}$$
$$= \sigma^2\mathrm{E}\left\{\left[\left(Z - \left(-\frac{\mu}{\sigma}\right)\right)^+\right]^2\right\} - [\mathrm{E}(\tilde{D})]^2$$
$$= \sigma^2\left[\bar{\Phi}\left(-\frac{\mu}{\sigma}\right) + \frac{\mu}{\sigma}G\left(-\frac{\mu}{\sigma}\right) - G^2\left(-\frac{\mu}{\sigma}\right)\right]. \tag{18}$$

For moderately large $x$ (say, $x \geq 2$), from (8), we have $G(-x) \cong x$, and hence

$$\mathrm{E}[\tilde{D}] \cong \mathrm{E}[D], \qquad \mathrm{Var}\,[\tilde{D}] \cong \mathrm{Var}[D],$$

from (17) and (18). Therefore, the above adjustment is only needed when the coefficient of variation of the demand, $\sigma/\mu$, is relatively large, say, 0.5 or above. (In

the numerical examples of §6 we indeed implement this adjustment in the relevant cases.)

From (6), (15), and (16), it is clear that to identify the base-stock policy is tantamount to specifying the safety factor $k_i$ for each component inventory. In the following sections, we discuss how to set the safety factor values so as to achieve the best inventory-service performance as specified in the optimization problems below.

For ease of exposition, we shall focus on stationary demand. For nonstationary demand, we can simply solve the optimization problems below period by period.

# 4. Service Requirement

## 4.1. Off-the-Shelf Availability

To start with, consider the special case in which each order of type $m$ requires exactly one unit of component $i \in \mathcal{S}_m$. Let $\alpha$ be the required service level, defined here as the immediate (i.e., off-the-shelf) availability of all the components required to assemble a unit of type $m$ product, for any $m$. Let $E_i$ denote the event that component $i$ is out of stock. Then, we require

$$\mathrm{P}\left[\bigcup_{i\in\mathcal{S}_m} E_i\right] \le 1 - \alpha.$$

Making use of the well-known inclusion-exclusion formula (e.g., Ross 1998):

$$\mathrm{P}\left[\bigcup_{i\in\mathcal{S}_m} E_i\right] = \sum_i \mathrm{P}(E_i) - \sum_{i<j} \mathrm{P}(E_i \cap E_j) + \sum_{i<j<k} \mathrm{P}(E_i \cap E_j \cap E_k) - \cdots,$$

where the indices $i$, $j$, $k$ on the right-hand side all belong to $\mathcal{S}_m$, we have, as an approximation,

$$\mathrm{P}\left[\bigcup_{i\in\mathcal{S}_m} E_i\right] \cong \sum_{i\in\mathcal{S}_m} \mathrm{P}(E_i) = \sum_{i\in\mathcal{S}_m} \bar{\Phi}(k_i) \le 1 - \alpha. \quad (19)$$

Note the essence of the above approximation is to ignore the probability of simultaneous stockout of two or more components.

There is another way to arrive at the above inequality. Suppose we express the service requirement as follows:

$$\prod_{i\in\mathcal{S}_m} \Phi(k_i) \ge \alpha. \quad (20)$$

Note that the left-hand side in (20) is, in fact, a *lower bound* of the availability (no-stockout probability) of the set of components in $\mathcal{S}_m$ that is required to assemble the end product $m$, i.e., it is a lower bound of the desired immediate availability. This claim (of a lower bound) can be argued by using stochastic comparison techniques involving the notion of *association.* (Refer to, e.g., Ross 1998 for background materials.) Intuitively, since the component inventories are driven by a common demand stream $\{D_m(t)\}$, and hence positively correlated, the chance of missing one or several components must be less than when the component inventories are independent, which is what is assumed by the product on the left-hand side of (20).

Since

$$\prod_{i\in\mathcal{S}_m} \Phi(k_i) = \prod_{i\in\mathcal{S}_m} [1 - \bar{\Phi}(k_i)] \cong 1 - \sum_{i\in\mathcal{S}_m} \bar{\Phi}(k_i), \quad (21)$$

combining the above and (20), we arrive at the same inequality in (19).

In the general setting, consider demand of product family $m$. Let $A \subseteq \mathcal{S}_m$ denote a certain configuration which occurs in this demand stream with probability $\mathrm{P}(A)$. Then the no-stockout probability, $\Pi_{i\in A}\,\Phi(k_i)$, should be weighted by $\mathrm{P}(A)$. Hence, the service requirement in (20) should be changed to

$$\begin{aligned}\alpha &\le \sum_{A\subseteq\mathcal{S}_m} \mathrm{P}(A) \prod_{i\in A} \Phi(k_i)\\ &\approx \sum_{A\subseteq\mathcal{S}_m} \mathrm{P}(A)\left[1 - \sum_{i\in A} \bar{\Phi}(k_i)\right]\\ &= 1 - \sum_{A\subseteq\mathcal{S}_m} \mathrm{P}(A) \sum_{i\in A} \bar{\Phi}(k_i)\\ &= 1 - \sum_{i\in\mathcal{S}_m} \left(\sum_{A\ni i} \mathrm{P}(A)\right)\bar{\Phi}(k_i).\end{aligned}$$

Since

$$\sum_{A\ni i} \mathrm{P}(A) = \mathrm{P}(X_{mi} > 0) := r_{mi}, \quad (22)$$

the service requirement in (19) can be extended to the following:

$$\sum_{i \in \mathcal{S}_m} r_{mi} \bar{\Phi}(k_i) \leq 1 - \alpha, \tag{23}$$

where $r_{mi}$ follows (22).

Note that when the batch size $X_{mi}$ is large, the stockout at component $i$ should occur more often than $\bar{\Phi}(k_i)$ [cf. Lee 1996]. To see this, first consider the case of unit (demand) arrivals, i.e., $X_{mi} \equiv 1$. Then, the stockout probability is $\mathrm{P}[D_i > R_i]$, or $\mathrm{P}[D_i + 1 \geq R_i]$.

In the general case of batch arrivals, the stockout probability is

$$\mathrm{P}[D_i + X_{mi} \geq R_i] = \mathrm{P}\left[Z \geq k_i - \frac{X_{mi}}{\sigma_i}\right] = \bar{\Phi}\left(k_i - \frac{X_{mi}}{\sigma_i}\right),$$

which is larger than $\bar{\Phi}(k_i)$. But this gap should be insignificant since

$$\frac{X_{mi}}{\sigma_i} = \frac{X_{mi}}{\theta_i \mu_i},$$

where $\theta_i := \sigma_i / \mu_i$, with $\mu_i$ and $\sigma_i$ following (14). The batch size of an incoming demand is usually orders of magnitude smaller when compared against $\mu_i$, which is the mean of demand summed over all product types $m \in \mathcal{M}_i$ and over the leadtime. (Large-size orders will likely be processed via separate contracts/channels rather than in a CTO environment.) Hence, below we shall simply use $\bar{\Phi}(k_i)$ as the stockout probability. Also note that this underestimation of the stockout probability is compensated for by the overestimation involved in (20), because the latter is a lower bound of the no-stockout probability.

### 4.2. Response-Time Serviceability

We now relate the immediate availability discussed above to another type of customer service requirement, which is expressed in terms of order response time, $W_m$—the delay (waiting time) between the time the order is placed and the time it is received by the customer.

Suppose the required service level of type $m$ demand is:

$$\mathrm{P}[W_m \leq w_m] \geq \alpha, \tag{24}$$

where $w_m$ and $\alpha$ are parameters that specify the given requirement, e.g., fulfill the order within $w_m = 5$ days with $\alpha = 99\%$ probability.

We have the following two cases:

(i) When there is no stockout at any component $i \in \mathcal{S}_m$—denoting the associated probability as $\pi_{0m}(t)$—the delay is simply $L_m^{\text{out}}$, the outbound leadtime.

(ii) Suppose there is a stockout at a component $i \in \mathcal{S}_m$. Denote the associated probability as $\pi_{im}(t)$. Then, the delay becomes $L_m^{\text{out}} + \tau_i$, where $\tau_i$ is the additional delay before the stocked-out component becomes available.

Hence, we can write

$$\begin{aligned}\mathrm{P}[W_m \leq w_m] &\approx \pi_{0m}(t)\mathrm{P}[L_m^{\text{out}} \leq w_m] \\ &\quad + \sum_{i \in \mathcal{S}_m} \pi_{im}(t)\mathrm{P}[L_m^{\text{out}} + \tau_i \leq w_m] \\ &= \left[\prod_{i \in \mathcal{S}_m} r_{mi}\Phi(k_i)\right]\mathrm{P}[L_m^{\text{out}} \leq w_m] \\ &\quad + \sum_{i \in \mathcal{S}_m} r_{mi}\bar{\Phi}(k_i)\mathrm{P}[L_m^{\text{out}} + \tau_i \leq w_m]. \end{aligned} \tag{25}$$

Note that in the above approximation, we have again ignored the probability of two or more components stocking out at the same time.

In most applications, it is reasonable to assume $L_m^{\text{out}} \leq w_m$. For instance, this is the case when the outbound leadtime $L_m^{\text{out}}$ is nearly deterministic and the delay limit $w_m$ is set to be "safely" larger than $L_m^{\text{out}}$.

On the other hand, $\tau_i$, which is quite intractable, can be approximated as follows (refer to Ettl et al. 2000, Equation (7)):

$$\tau_i \approx l_i^{\text{in}} \cdot \frac{\mathrm{E}(B_i)}{\bar{\Phi}(k_i)(R_i + 1)}.$$

From (11), we have

$$\mathrm{E}(B_i) = \sigma_i G(k_i) = \sigma_i[\phi(k_i) - k_i\bar{\Phi}(k_i)].$$

Note that the following holds:

$$\frac{\phi(x) - x\bar{\Phi}(x)}{\bar{\Phi}(x)} \sim \frac{\bar{\Phi}(x)}{\phi(x)} \sim \frac{1}{x},$$

for moderately large $x$ (e.g., $x \geq 3$). Hence, making use of the above, along with (15), we have

$$\begin{aligned}\tau_i &\approx l_i^{\text{in}} \cdot \frac{\sigma_i}{k_i(R_i + 1)} \\ &\le l_i^{\text{in}} \cdot \frac{\sigma_i}{k_i R_i} \\ &= \frac{l_i^{\text{in}}\sqrt{l_i^{\text{in}}}\,\text{sd}(D_i)}{k_i[l_i^{\text{in}}\text{E}(D_i) + k_i\sqrt{l_i^{\text{in}}}\,\text{sd}(D_i)]} \\ &= \frac{l_i^{\text{in}}}{k_i\sqrt{l_i^{\text{in}}}\,\text{E}(D_i)/\text{sd}(D_i) + k_i^2}. \end{aligned} \tag{26}$$

The above allows us to set $w_m$ such that $w_m \ge L_i^{\text{out}} + \tau_i$, for all $i \in \mathcal{S}_m$; then the response-time serviceability in (25) can be met at almost 100% (i.e., modulo the approximation involved in estimating $\tau_i$). In other words, aiming for a high immediate availability (say, 95%) will enable us to set a reasonable response-time target ($w_m$) and to achieve a near-100% service level.

## 5. Inventory-Service Optimization

Our objective is to minimize the expected inventory investment (or, more precisely, inventory capital), subject to meeting the service requirement for each product family as expressed in (23). The problem can be presented as follows:

$$\begin{aligned} \min \quad & \sum_{i\in\mathcal{S}} c_i\sigma_i H(k_i) \\ \text{s.t.} \quad & \sum_{i\in\mathcal{S}_m} r_{mi}\bar{\Phi}(k_i) \le \bar{\alpha}_m, \qquad m \in \mathcal{M}, \end{aligned}$$

where $r_{mi}$ is the probability defined in (22), $c_i$ is the unit cost of the on-hand inventory of component $i$, and $\bar{\alpha}_m = 1 - \alpha_m$ with $\alpha_m$ being the required service level for product family $m$. Recall that $\sigma_i H(k_i)$ is the expected on-hand inventory of component $i$; refer to (10), and $\sigma_i$ follows the specification in (14).

To solve the above optimization problem, we first rewrite the constraints as follows:

$$\sum_{i\in\mathcal{S}_m} r_{mi}\Phi(k_i) \ge \sum_{i\in\mathcal{S}_m} r_{mi} - \hat{\alpha}_m,$$

and (abusing notation) let

$$r_{mi} \leftarrow \frac{r_{mi}}{\sum_{i\in\mathcal{S}_m} r_{mi} - \bar{\alpha}_m}.$$

The optimization problem then becomes:

$$\begin{aligned} \min \quad & \sum_{i\in\mathcal{S}} c_i\sigma_i H(k_i) && (27) \\ \text{s.t.} \quad & \sum_{i\in\mathcal{S}_m} r_{mi}\Phi(k_i) \ge 1, \qquad m \in \mathcal{M}. && (28) \end{aligned}$$

Several remarks are in order:

Remark 1. Note that $H(\cdot)$ is an increasing and convex function, as is evident from the first equality in (9), while $\Phi(\cdot)$ is an increasing function.

Remark 2. For any two product types, $m$ and $m'$, if $\mathcal{S}_{m'} \subset \mathcal{S}_m$, and $r_{m'i} \le r_{mi}$ for $i \in \mathcal{S}_{m'}$, then the constraint corresponding to $m'$ becomes superfluous. We assume in the above formulation that all such superfluous constraints have already been removed through preprocessing.

Remark 3. Suppose a product family $m$ involves a *unique* component, denoted $i_m$, i.e., $i_m \notin \mathcal{S}_{m'}$ when $m' \ne m$. Then, the corresponding constraint must be binding. For otherwise, we can always decrease the value of $k_{i_m}$ until the constraint becomes binding, without affecting the other constraints, while decreasing the objective value (because $H(\cdot)$ is an increasing function as explained in Remark 1).

The Lagrangian corresponding to the above optimization problem is:

$$L = \sum_{i\in\mathcal{S}} c_i\sigma_i H(k_i) - \sum_{m\in\mathcal{M}} \lambda_m\left(\sum_{i\in\mathcal{S}_m} r_{mi}\Phi(k_i) - 1\right), \tag{29}$$

where $\lambda_m \ge 0$, $m \in \mathcal{M}$, are the Lagrangian multipliers. Hence, taking derivatives and setting them to zero, taking into account

$$H'(x) = -x\phi(x) + \Phi(x) + x\phi(x) = \Phi(x), \tag{30}$$

we obtain the following system of nonlinear equations that characterizes the optimal solution:

$$\sum_{m\in\mathcal{M}_i} r_{mi}\lambda_m = c_i\sigma_i\frac{\Phi(k_i)}{\phi(k_i)}, \qquad i \in \mathcal{S}; \tag{31}$$

$$\sum_{i\in\mathcal{S}_m} r_{mi}\,\Phi(k_i) = 1, \qquad m \in \mathcal{M} \quad \text{and} \quad \lambda_m > 0. \tag{32}$$

### 5.1. Unique Components

While solving the above system of nonlinear equations is quite intractable in general, in the following important case we do have an efficient algorithm that

solves the nonlinear equations and generates the optimal solution.

Suppose every product family $m$ uses a unique component $i_m$ that is not used in any other product families. Then, as pointed out in the above Remark 3, all the constraints in (28) must be binding, i.e., the equations in (32) hold for all $m \in \mathcal{M}$, because all the Lagrangian multipliers are positive.

Focusing on the unique components in (31), $i = i_m$, we have $\mathcal{M}_{i_m} = \{m\}$, a singleton set, and hence,

$$\lambda_m = \frac{c_{i_m}\sigma_{i_m}}{r_{mi_m}} \cdot \frac{\Phi(k_{i_m})}{\phi(k_{i_m})}, \qquad m \in \mathcal{M}. \tag{33}$$

Suppose we have derived the variables for all the unique components, $k_{i_m}$, $m \in \mathcal{M}$. Then, all the Lagrangian multipliers, $\lambda_m$, $m \in \mathcal{M}$, follow from (33). We can then derive $k_i$, for all $i$ that is not a unique component, from the remaining equations in (31). That is,

$$\frac{\Phi(k_i)}{\phi(k_i)} = \frac{1}{c_i\sigma_i} \sum_{m \in \mathcal{M}_i} r_{mi}\lambda_m, \qquad i \in \mathcal{S}, \qquad i \neq i_m. \tag{34}$$

To close this loop, we still have to derive the variables $k_{i_m}$ that correspond to the unique components. But these are readily derived from the equations in (32), each involving exactly one such variable; all the other variables corresponding to the nonunique components have already been derived.

In particular, direct verification establishes that $\Phi(k)/\phi(k)$ is increasing in $k$. Hence, we know the variables corresponding to the nonunique components, $(k_i)_{i \in \mathcal{S}, i \neq i_m}$, are increasing in those of the unique components, $(k_{i_m})_{m \in \mathcal{M}}$. Hence, denoting $y_m := \Phi(k_{i_m})$ and $y := (y_m)_{m \in \mathcal{M}}$, we can write (32) as

$$r_{mi_m}y_m + \sum_{i \neq i_m} r_{mi}h_{mi}(y) = 1, \qquad m \in \mathcal{M}, \tag{35}$$

where $h_{mi}(y)$ are increasing functions. Therefore, we can solve this system of equations using a bisection-like algorithm detailed below:

ALGORITHM 1.

1. *For each $m \in \mathcal{M}$, set $k_{i_m} = 0$; set $y_m^L = 0$ and $y_m^H = 1$. Set $\epsilon = 10^{-6}$ (or any other desired accuracy).*
2. *For each $m \in \mathcal{M}$, compute $\lambda_m$ following (33).*
3. *For each nonunique component, $i \neq i_m$, compute $k_i$ from (34). (This can be done by gradually incrementing $k_i$ until the ratio on the left reaches the same value as the right-hand side. Or, use bisection.)*
4. *For each $m \in \mathcal{M}$, set $y_m = \Phi(k_{i_m})$, and compute the left-hand side of (32), denoted $LHS_m$:*
   - *If $LHS_m > 1 + \epsilon$, set $y_m^H \leftarrow y_m$ and $y_m \leftarrow (y_m + y_m^L)/2$;*
   - *if $LHS_m < 1 + \epsilon$, set $y_m^L \leftarrow y_m$ and $y_m \leftarrow (y_m + y_m^H)/2$;*
   - *derive $k_{i_m} = \Phi^{-1}(y_m)$.*

*If $\max_{m \in \mathcal{M}} |y_m - y_m'| \leq \epsilon$ (where $y_m'$ denotes the $y_m$ value of the previous iteration), stop; else, go to 2.*

While the above algorithm is guaranteed to converge, thanks to the bisection procedure, it may not converge to the optimal solution. This can happen when at convergence the left-hand sides of (35) are not all equal to one. This has been observed in our numerical studies, but the gap to optimality is negligible.

On the other hand, convergence to the optimal solution is guaranteed if for each $m$, the increase or decrease in the left-hand side of (35) is dominated by the first term, i.e., an increase (respectively, decrease) in $y_m$ results in an increase (respectively, decrease) in $r_{mi_m}y_m + \Sigma_{i \neq i_m} r_{mi}h_{mi}(y)$, regardless of the increase or decrease in the other components of $y$. This appears to be the case observed in many of the numerical examples we have run.

### 5.2. A Greedy Heuristic

Without the unique components, the main difficulty in solving the optimization problem in (27) and (28) has to do with the combinatorial nature of the equations in (32): We have to consider all possibilities of the Lagrangian multipliers being zero or positive. Hence, we propose a greedy heuristic as follows. We gradually decrease the left-hand side of all the constraints to one; at each step, we identify a variable $k_i$ such that incrementing its value will yield the smallest increase in the objective value.

At each step in the algorithm, let $\beta_m$ denote the value of the left-hand side of the constraint $m$ in (32), and let $\delta := \max_{m \in \mathcal{M}} \beta_m$. Denote

$$A := \{i : \beta_m < \delta, \forall m \in \mathcal{M}_i\}.$$

That is, the set $A$ collects the indices of all variables

that can be increased without increasing $\delta$. We want to identify

$$i^* = \underset{i \in A}{\operatorname{argmin}} \{c_i \sigma_i [H(k_i + \delta_i) - H(k_i)]\},$$

where

$$\delta_i := \min_{m \in \mathcal{M}_i} \left\{ \frac{\delta - \beta_m}{r_{mi}} \right\} \tag{36}$$

specifies how much $k_i$ can be increased (without exceeding the current value of the constraint $\delta$). Since $\delta_i$ is a small increment, we can approximate the above difference by the derivative (refer to (30)),

$$H'(k_i + \delta_i/2) \cdot \delta_i = \Phi(k_i + \delta_i/2) \cdot \delta_i.$$

To summarize, here is the algorithm:

ALGORITHM 2.

1. *For each $i \in \mathcal{S}$, set $k_i = 0$. For each $m \in \mathcal{M}$, set $\beta_m = 0$. Set $\delta = \Delta$.*

2. *Identify the set $A := \{i : \beta_m < \delta, \forall m \in \mathcal{M}_i\}$. If $\delta = 1$ and $A = \emptyset$, stop; else, continue.*

3. *If $A = \emptyset$ and $\delta < 1$, set $\delta \leftarrow \delta + \Delta$. Find $i^* = \arg\min_{i \in A} \{c_i \sigma_i \delta_i \Phi(k_i + \delta_i/2)\}$, where $\delta_i$ follows (36). Set $k_{i^*} \leftarrow k_{i^*} + \delta_{i^*}$.*

4. *For $m \in \mathcal{M}_{i^*}$, set $\beta_m \leftarrow \beta_m + r_{mi^*} k_{i^*}$. Go to 2.*

### 5.3. Sensitivity and Inventory-Service Trade-off

As is well known, the Lagrangian multipliers in the optimization model discussed above have the interpretation of ''shadow prices.'' Specifically, suppose the right-hand side of (28) is changed from 1 to $1 - \epsilon$, where $\epsilon$ is a small positive quantity. Then the change in the objective value, around optimality, is approximately $-\epsilon \sum_{m \in \mathcal{M}} \lambda_m$, and this is evident from (29).

On the other hand, from the formulation of the optimization problem we can verify that the change of the right-hand side of (28) from 1 to $1 - \epsilon$ corresponds to increasing $\bar{\alpha}_m$ (the no-fill rate) by an amount

$$\epsilon \left( \sum_{i \in \mathcal{S}_m} r_{mi} - \bar{\alpha}_m \right).$$

Hence, the Lagrangian multipliers returned by the algorithms developed earlier can be used to conduct sensitivity analysis. Specifically, we know that a reduction of the total inventory investment by an amount $d$ can be achieved by reducing the service requirement of product family $m$, provided $\lambda > 0$, from $\alpha_m$ to

$$\alpha_m - \epsilon \left[ \sum_{i \in \mathcal{S}_m} r_{mi} - (1 - \alpha_m) \right],$$

with $\epsilon = d/\lambda_m$.

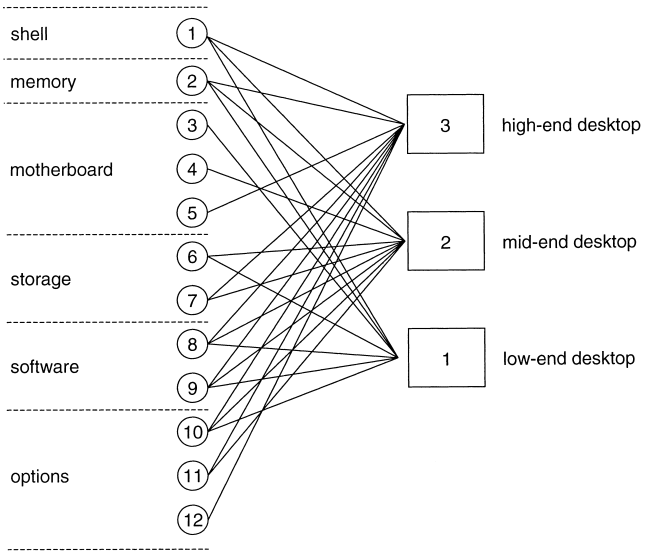
This sensitivity analysis can also be used in another way. Since the service level used in the optimization problem is a lower bound of the true value, the optimal solution is conservative in that it returns a service level that is higher than what is required. Our numerical experience shows this typically results in an objective value that is 10–15% higher (than if the true service levels were used). Therefore, after solving the optimization problem we can use the above sensitivity analysis to find the reductions on the $\alpha_m$ values (required service levels) that correspond to reducing the objective value by say, 10%, and then resolve the optimization problem using the reduced $\alpha_m$ values. (Refer to the discussions on the results in Tables 4 and 5 of the next section.)

## 6. Numerical Results

For our numerical studies, we consider a family of desktop computers which are assembled from a set of 12 different components $i = 1, 2, \ldots, 12$ as illustrated in Figure 1.

All components used in the assembly of an end product are purchased from external suppliers. The supplier leadtimes are deterministic with $E(L_i^{in})$ time units, which represent the time required to manufacture the component and ship it from a supplier warehouse to the PC assembly plant. The lead times, unit costs, and demand configurations are summarized in Table 1. These are based on actual data from a PC assembly system, with the necessary alterations to protect proprietary information. The modified values, however, still provide a realistic product representation. As the final assembly process for end products takes no more than a few hours, which is an order of magnitude shorter than the component leadtime, the

**Figure 1 A Configure-to-Order System with Three Product Families**




system fits well within the framework discussed in §4.

We consider three customer segments, i.e., $m = 1, 2, 3$, representing low-end, mid-range, and high-end demands, respectively. Orders for end products differ by customer segment in terms of the possible selection of components that constitute the end product. We assume that each order requires no more than one unit from component $i$. The random variables $X_{mi}$ defined in §2 thus take on values of zero or one. The marginal distribution of $X_{mi}$ is displayed in the last three columns of Table 1 for each customer segment. It determines the possible combination of components that can be selected by orders from each customer segment. The distribution indicates the proportion of orders that request a specific component. For instance, for high-end customers, 100% of the orders select a 13GB disk drive. For mid-range customers, 40% of orders request a 7GB disk drive, and 60% request a 13GB hard drive. Each order requests exactly one component from each component category. The only exception is the ''options'' category, of which more than one component can be selected.

We represent the customer orders per time unit as i.i.d. normal random variables with mean $E(D_m) = 100$, and coefficient of variation $CV(D_m) = 0.25$ and 0.50, for all $m = 1, 2, 3$.

Notice that end products from any given customer segment use a motherboard that is unique to that customer segment. Therefore, we can use the algorithm described in §5.1 to compute the optimal base-stock policies.

### 6.1. Validation of the Analytical Model

Here we examine two aspects of the analytical model developed in the previous sections: the solution algorithm (of §5.1) and the lower bound (for the fill rate) involved in the service-level constraints.

For the first issue, recall that the bisection proce-

**Table 1 Bill-of-Materials Structure for Example Configure-to-Order System**

| $i$ | Category | Component | Leadtime E ($L_i^{in}$) | Unit Cost $c_i$ | P ($X_{mi} > 0$) | | |
|---|---|---|---|---|---|---|---|
| | | | | | Low-End | Mid-Range | High-End |
| 1 | Shell | Base unit | 5 | 215 | 1.0 | 1.0 | 1.0 |
| 2 | Memory | 128MB card | 15 | 232 | 1.0 | 1.0 | 1.0 |
| 3 | Motherboard | 450 MHz board | 12 | 246 | 1.0 | — | — |
| 4 | Motherboard | 500 MHz board | 12 | 316 | — | 1.0 | — |
| 5 | Motherboard | 600 MHz board | 12 | 639 | — | — | 1.0 |
| 6 | Storage | 7GB disk drive | 18 | 215 | 1.0 | 0.4 | — |
| 7 | Storage | 13GB disk drive | 18 | 250 | — | 0.6 | 1.0 |
| 8 | Software | Preload A | 4 | 90 | 0.7 | 0.5 | 0.3 |
| 9 | Software | Preload B | 4 | 90 | 0.3 | 0.5 | 0.7 |
| 10 | Options | CD ROM | 10 | 126 | 1.0 | 1.0 | 1.0 |
| 11 | Options | Video graphics card | 6 | 90 | — | 0.3 | 0.6 |
| 12 | Options | Ethernet card | 10 | 90 | — | 0.2 | 0.5 |

**Table 2 Quality of the Algorithm: Identical Service Target for All Customer Segments**

| | $CV(D_m) = 0.25$ | | | | $CV(D_m) = 0.50$ | | | |
|---|---|---|---|---|---|---|---|---|
| $\alpha$ | $\alpha^*_{analy}$ | $z^*_{analy}$ | $z^*_{search}$ | rel.dev. | $\alpha^*_{analy}$ | $z^*_{analy}$ | $z^*_{search}$ | rel.dev. |
| 0.800 | (0.800, 0.800, 0.800) | 437,637 | — | — | (0.800, 0.800, 0.800) | 875,273 | — | — |
| 0.820 | (0.820, 0.819, 0.818) | 450,311 | 451,121 | 0.2% | (0.820, 0.819, 0.818) | 900,621 | 902,243 | 0.2% |
| 0.840 | (0.839, 0.837, 0.840) | 462,308 | 463,088 | 0.2% | (0.839, 0.837, 0.840) | 924,616 | 926,176 | 0.2% |
| 0.860 | (0.860, 0.860, 0.860) | 477,489 | — | — | (0.860, 0.860, 0.860) | 954,978 | — | — |
| 0.880 | (0.880, 0.880, 0.880) | 494,050 | — | — | (0.880, 0.880, 0.880) | 988,100 | — | — |
| 0.900 | (0.901, 0.900, 0.900) | 513,383 | 512,050 | −0.3% | (0.901, 0.900, 0.900) | 1,026,766 | 1,024,199 | −0.3% |
| 0.920 | (0.920, 0.920, 0.920) | 536,004 | — | — | (0.920, 0.920, 0.920) | 1,072,007 | — | — |
| 0.940 | (0.940, 0.940, 0.940) | 564,446 | — | — | (0.940, 0.940, 0.940) | 1,128,892 | — | — |
| 0.960 | (0.960, 0.960, 0.960) | 602,862 | — | — | (0.960, 0.960, 0.960) | 1,205,723 | — | — |
| 0.980 | (0.980, 0.980, 0.980) | 664,478 | — | — | (0.980, 0.980, 0.980) | 1,328,956 | — | — |

dure is guaranteed to converge, but it may converge to a suboptimal solution. More specifically, the convergent point (solution) is optimal if and only if all the service constraints are binding. Hence, in Table 2 we present the solutions for two different demand scenarios (in terms of the coefficients of variation) and several different levels of required service, ranging from 80% to 98% (but uniform across customer segments). The column labeled $\alpha^*_{analy}$ reports the fill rates corresponding to the solution returned by the algorithm, with the objective values reported in the column labeled $z^*_{analy}$.

There are three solutions in each demand scenario that are suboptimal, reflected by the gaps between $\alpha$ and $\alpha^*_{analy}$. For each of these solutions, we perform a random search of 5,000,000 points around the neighborhood of the point that the bisection procedure converges to, and report under $z^*_{search}$ the best objective value among all solutions feasible to (28). We can see that the objective values match very well. In particular, in the two cases where the algorithm overachieves the required service level, the objective value is only slightly above the best value returned by the search procedure.

Table 3 extends the same study to allow different service requirements for different customer segments. The various combinations of service requirements are displayed in the first three columns. For all the above examples, the number of iterations until convergence is between 15 and 23, for a tolerance of $\epsilon = 10^{-6}$.

To address the second issue, the quality of involving the lower bound on the fill rate (off-the-shelf availability) in the service constraints, for every combination of the service requirements listed in Table 3, we determine the ''true'' fill rate for each customer segment by using simulation to evaluate the solution generated by the analytical method. Figures 2 to 4 show the comparisons between the analytical and simulated values. (The coefficient of variation of demand was 0.50.) In each figure, we varied the service requirements of one customer segment and held the other two constant. For example, the service requirements (targets) for the high-end and mid-range customer segments in Figure 4 were 0.98 and 0.95, respectively, while the service target for the low-end segment varies between 0.90 and 0.99. (The simulation point estimates are obtained by the batch-mean method, dividing every run into 10 batches of 5,000 arrivals per customer segment.) Clearly, the analytical values, being a lower bound of the true fill rate, underestimate the achieved service level (i.e., overachieve the service requirements). Overall, however, the analytical curves track the simulated curves quite closely, and the gap decreases as the service requirement increases.

Next, we continue examining the second issue raised above, but from a different angle: Since the lower bounds involved in the service constraints overachieve the service requirements, what is the true optimal solution that exactly meets the service requirements? To study this, we first apply simulation to the cases in Table 2 and report the results in Table 4,

**Table 3 Quality of the Algorithm: Different Service Targets for Different Customer Segments**

| | | | $CV(D_m) = 0.50$ | | | |
|---|---|---|---|---|---|---|
| $\alpha_1$ | $\alpha_2$ | $\alpha_3$ | $\alpha^*_{analy}$ | $z^*_{analy}$ | $z^*_{search}$ | rel.dev. |
| 0.920 | 0.950 | 0.900 | (0.921, 0.950, 0.900) | 1,083,953 | 1,085,977 | 0.2% |
| 0.920 | 0.950 | 0.920 | (0.920, 0.950, 0.920) | 1,102,866 | — | — |
| 0.920 | 0.950 | 0.940 | (0.919, 0.950, 0.940) | 1,128,973 | 1,131,144 | 0.2% |
| 0.920 | 0.950 | 0.960 | (0.919, 0.950, 0.960) | 1,168,104 | 1,172,183 | 0.3% |
| 0.920 | 0.950 | 0.980 | (0.918, 0.948, 0.980) | 1,239,232 | 1,244,627 | 0.4% |
| 0.920 | 0.900 | 0.980 | (0.918, 0.908, 0.980) | 1,215,895 | 1,217,521 | 0.1% |
| 0.920 | 0.920 | 0.980 | (0.919, 0.925, 0.980) | 1,224,201 | 1,226,235 | 0.2% |
| 0.920 | 0.940 | 0.980 | (0.916, 0.943, 0.980) | 1,234,501 | 1,237,110 | 0.2% |
| 0.920 | 0.960 | 0.980 | (0.920, 0.964, 0.980) | 1,256,906 | 1,255,050 | −0.1% |
| 0.920 | 0.980 | 0.980 | (0.923, 0.980, 0.980) | 1,292,961 | 1,290,429 | −0.2% |
| 0.900 | 0.950 | 0.980 | (0.904, 0.947, 0.980) | 1,233,337 | 1,238,372 | 0.4% |
| 0.920 | 0.950 | 0.980 | (0.918, 0.948, 0.980) | 1,239,232 | 1,244,627 | 0.4% |
| 0.940 | 0.950 | 0.980 | (0.942, 0.949, 0.980) | 1,251,034 | 1,254,381 | 0.3% |
| 0.960 | 0.950 | 0.980 | (0.958, 0.950, 0.980) | 1,263,566 | 1,267,927 | 0.3% |
| 0.980 | 0.950 | 0.980 | (0.980, 0.955, 0.980) | 1,298,052 | 1,297,527 | 0.0% |

**Figure 2 Comparison Between Analytical and Simulated Fill Rates**

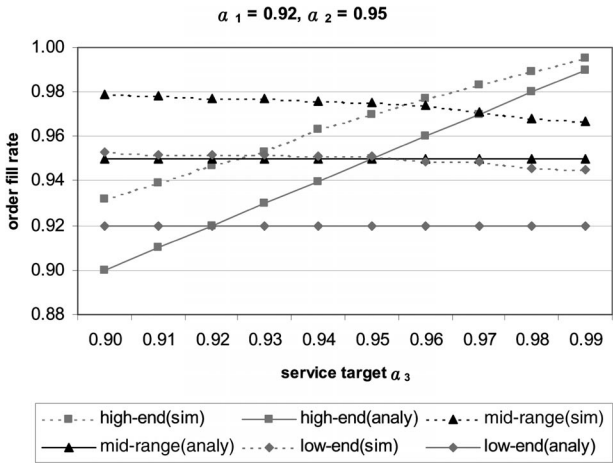
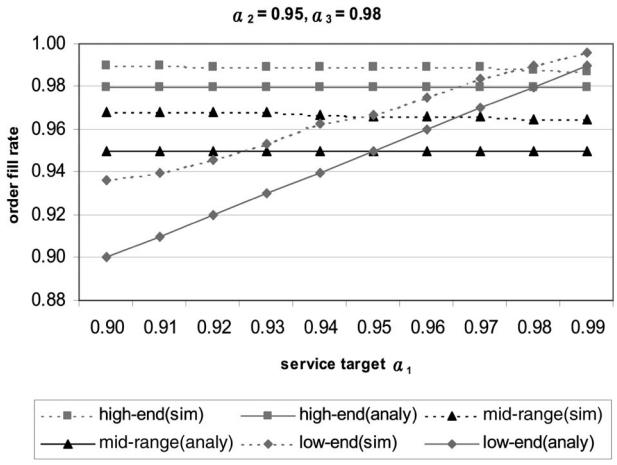
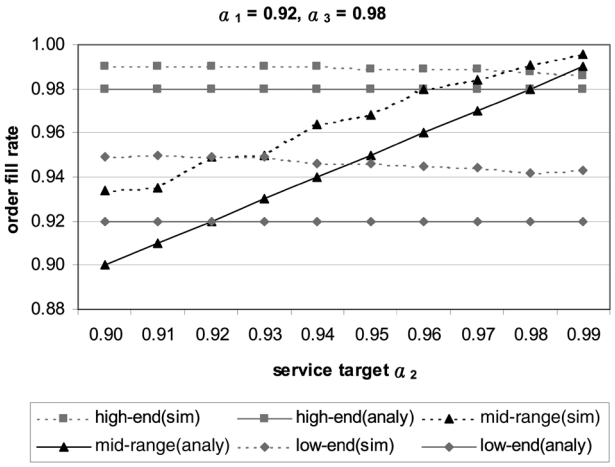



**Figure 3 Comparison Between Analytical and Simulated Fill Rates**




where the column under $\alpha^*_{analy}$ lists the actual (i.e., *simulated*) service levels achieved by the analytical solutions, and the column under $z^*_{analy}$ lists the corresponding objective values (which are the same as in Table 2). Next, we use the sensitivity analysis in §5.3, combined with simulation, to search for the best solution that exactly meets the service requirements (as verified by simulation). Specifically, we gradually decrease the service requirements in the optimization model (i.e., the $\alpha_m$ values) using the sensitivity analysis as a guideline, solve the new optimization problem, and then use simulation to find the achieved service levels. This procedure is repeated until the analytical solution returns a set of simulated service levels that are within a predefined tolerance of the original service requirements. When this procedure terminates, we generate 10,000 more points randomly around the neighborhood of the identified solution. The best solution found in this procedure (i.e., the one with the lowest expected inventory investment) is re-

**Figure 4 Comparison Between Analytical and Simulated Fill Rates**




ported under the columns labeled $\alpha^*_{\text{search}}$ and $z^*_{\text{search}}$. The relative difference between $z^*_{\text{analy}}$ and $z^*_{\text{search}}$ is reported in the column labeled $\Delta_z$. We applied the same study to the cases in Table 3, with the results summarized in Table 5.

As the numbers in Tables 4 and 5 illustrate, the objective values corresponding to the analytical solution are about 8–15% suboptimal. This is due to the lower bounds used in the service constraints. In practice, the analytical method can be used to quickly generate a starting solution. It can also be combined with simulation and sensitivity analysis, as outlined above, to fine-tune the starting solution until optimality.

### 6.2. Effect of Risk Pooling

In the examples presented above, the inventory of each common component forms a common buffer from which the demand in all customer segments can draw (if the component is needed). Hence, demand for each component is an aggregate of the demand over all customer segments. To the extent that the safety stock of the component inventory is determined, among other things, by the variability of this aggregate demand (on the component), the required inventory level of this component to achieve a given service target can be lowered by this aggregation, or *risk pooling*. In general, the impact of risk pooling becomes more significant when the number of product configurations is large and the correlation among demands is small (e.g., Aviv and Federgruen 1999, Brown et al. 2000, Lee 1996).

To understand how risk pooling influences the performance of a CTO system, we compare some examples studied above with their no-risk-pooling counterparts. Specifically, we consider the cases in Table 4, with the coefficient of variation of demand being 0.5. For the no-risk-pooling scenario, we apply the optimization to each customer segment separately, and then sum up the expected inventory investments over all three customer segments. The comparison against the risk-pooling results is plotted in Figure 5.

In the figure, the service levels are the actually realized values obtained by simulating the analytical solutions. We observe that risk pooling results in a

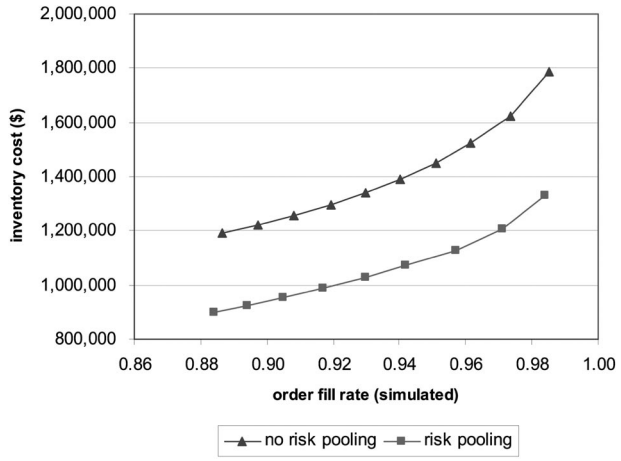
**Table 4 Comparisons Between the Analytical and the Search Solutions: Identical Service Target for All Customer Segments**

| | $CV(D_m) = 0.25$ | | | | | $CV(D_m) = 0.50$ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $\alpha$ | $\alpha^*_{\text{analy}}$ | $z^*_{\text{analy}}$ | $\alpha^*_{\text{search}}$ | $z^*_{\text{search}}$ | $\Delta_z$ | $\alpha^*_{\text{analy}}$ | $z^*_{\text{analy}}$ | $\alpha^*_{\text{search}}$ | $z^*_{\text{search}}$ | $\Delta_z$ |
| 0.800 | 0.890 | 437,636 | 0.805 | 372,116 | 15.0% | 0.893 | 875,272 | 0.813 | 744,232 | 15.0% |
| 0.820 | 0.896 | 450,310 | 0.839 | 385,133 | 14.5% | 0.902 | 900,620 | 0.828 | 767,392 | 14.8% |
| 0.840 | 0.908 | 462,307 | 0.852 | 398,653 | 13.8% | 0.911 | 924,614 | 0.844 | 792,946 | 14.2% |
| 0.860 | 0.919 | 477,492 | 0.877 | 414,776 | 13.1% | 0.919 | 954,983 | 0.865 | 823,529 | 13.8% |
| 0.880 | 0.928 | 494,050 | 0.893 | 428,233 | 13.3% | 0.928 | 988,099 | 0.895 | 856,465 | 13.3% |
| 0.900 | 0.939 | 513,383 | 0.915 | 452,212 | 11.9% | 0.939 | 1,026,766 | 0.915 | 904,428 | 11.9% |
| 0.920 | 0.950 | 536,003 | 0.934 | 477,074 | 11.0% | 0.949 | 1,072,007 | 0.931 | 954,148 | 11.0% |
| 0.940 | 0.962 | 564,446 | 0.947 | 503,809 | 10.7% | 0.961 | 1,128,892 | 0.950 | 1,011,283 | 10.4% |
| 0.960 | 0.974 | 602,861 | 0.969 | 553,908 | 8.1% | 0.973 | 1,205,723 | 0.968 | 1,107,816 | 8.1% |
| 0.980 | 0.985 | 664,477 | 0.982 | 610,014 | 8.2% | 0.985 | 1,328,955 | 0.982 | 1,220,027 | 8.2% |

**Table 5 Comparisons Between the Analytical and the Search Solutions: Different Service Targets for Different Customer Segments**

| $\alpha_1$ | $\alpha_2$ | $\alpha_3$ | $\alpha^*_{analy}$ | $z^*_{analy}$ | $\alpha^*_{search}$ | $z^*_{search}$ | $\Delta_z$ |
|---|---|---|---|---|---|---|---|
| 0.920 | 0.950 | 0.900 | (0.947, 0.969, 0.930) | 1,083,953 | (0.925, 0.959, 0.901) | 922,025 | 14.9% |
| 0.920 | 0.950 | 0.920 | (0.945, 0.968, 0.945) | 1,102,866 | (0.925, 0.961, 0.920) | 948,462 | 14.0% |
| 0.920 | 0.950 | 0.940 | (0.945, 0.966, 0.962) | 1,128,973 | (0.932, 0.951, 0.941) | 978,332 | 13.3% |
| 0.920 | 0.950 | 0.960 | (0.942, 0.964, 0.977) | 1,168,104 | (0.926, 0.955, 0.966) | 1,013,516 | 13.2% |
| 0.920 | 0.950 | 0.980 | (0.940, 0.958, 0.989) | 1,239,232 | (0.927, 0.953, 0.982) | 1,077,292 | 13.1% |
| 0.920 | 0.900 | 0.980 | (0.944, 0.919, 0.991) | 1,215,895 | (0.924, 0.902, 0.980) | 1,046,427 | 13.9% |
| 0.920 | 0.920 | 0.980 | (0.943, 0.936, 0.990) | 1,224,201 | (0.924, 0.921, 0.986) | 1,112,248 | 9.1% |
| 0.920 | 0.940 | 0.980 | (0.940, 0.953, 0.989) | 1,234,501 | (0.922, 0.940, 0.983) | 1,123,316 | 9.0% |
| 0.920 | 0.960 | 0.980 | (0.941, 0.972, 0.988) | 1,256,906 | (0.920, 0.960, 0.981) | 1,094,290 | 12.9% |
| 0.920 | 0.980 | 0.980 | (0.938, 0.986, 0.986) | 1,290,429 | (0.926, 0.981, 0.981) | 1,114,633 | 13.6% |
| 0.900 | 0.950 | 0.980 | (0.930, 0.958, 0.990) | 1,233,337 | (0.901, 0.951, 0.981) | 1,071,165 | 13.1% |
| 0.920 | 0.950 | 0.980 | (0.940, 0.958, 0.989) | 1,239,232 | (0.927, 0.953, 0.982) | 1,077,292 | 13.1% |
| 0.940 | 0.950 | 0.980 | (0.959, 0.956, 0.989) | 1,251,034 | (0.941, 0.955, 0.981) | 1,118,018 | 10.6% |
| 0.960 | 0.950 | 0.980 | (0.972, 0.955, 0.988) | 1,263,566 | (0.962, 0.958, 0.981) | 1,141,255 | 9.7% |
| 0.980 | 0.950 | 0.980 | (0.987, 0.956, 0.987) | 1,297,527 | (0.985, 0.956, 0.980) | 1,129,761 | 12.9% |

**Figure 5 Effect of Risk Pooling**




significant reduction of inventory investment: For the same service target, the expected inventory investment is 20–25% lower with risk pooling, with the largest gap appearing at the high end: 25.8% when $\alpha = 0.98$. Our other experiments indicate that the gap also increases with demand variability. For instance, when the coefficient of variation of demand is reduced from 0.5 to 0.25, the relative difference at a service target of 0.98 is 18.4%.

Table 6 shows the optimal component safety-stock levels expressed in days of supply (as defined in (16)) for both risk pooling and no risk pooling. As expected, risk pooling requires a lower days of supply value. Further, the amount of safety stock needed tends to be lower when the commonality of a component is higher. For example, in the risk-pooling scenario, the 128MB memory card is used in all three customer segments and requires 2.4 days of supply, whereas

**Table 6 Comparison Between Risk-Pooling and No-Risk-Pooling Scenarios, Service Target $\alpha = 0.90$ for All Customer Segments**

| | | $DOS_i$ (risk pooling) | $DOS_i$ (no risk pooling) | | |
|---|---|---|---|---|---|
| | | All | | | |
| $i$ | Component | Segments | Low-End | Mid-Range | High-End |
| 1 | Base unit | 1.6 | 2.5 | 2.6 | 2.7 |
| 2 | 128MB card | 2.4 | 3.8 | 3.9 | 4.2 |
| 3 | 450 MHz board | 3.0 | 3.4 | — | — |
| 4 | 500 MHz board | 3.0 | — | 3.3 | — |
| 5 | 600 MHz board | 2.9 | — | — | 2.9 |
| 6 | 7GB disk drive | 3.2 | 4.1 | 4.7 | — |
| 7 | 13GB disk drive | 3.2 | — | 3.7 | 4.4 |
| 8 | Preload A | 1.7 | 2.3 | 2.7 | 3.1 |
| 9 | Preload B | 1.7 | 3.0 | 2.7 | 2.5 |
| 10 | CD ROM | 2.3 | 3.7 | 3.8 | 4.0 |
| 11 | Video graphics card | 2.5 | — | 3.6 | 3.1 |
| 12 | Ethernet card | 3.4 | — | 4.8 | 4.2 |

the Ethernet card, which is used in the high-end and mid-range segments only, requires 3.4 days of supply.

## 7. A Process Reengineering Application

Here we describe the study mentioned in the Introduction, which was part of a larger project aimed at the reengineering of IBM PCD's business process—from a build-to-stock operation to a configure-to-order operation where safety-stock inventory is held only at component level.

To carry out the study, we have developed two basic models: the ''as-is'' model, which is a reflection of PCD's present operation, and the ''to-be'' model, which is based on the optimization model described in the previous sections—in particular, with the component inventory levels generated by the algorithm in §5. For both models, we aggregate PCD's production-inventory system into two stages, the first stage consists of the components, and the second stage includes the assembly and the order fulfillment. Three factors have been identified as focal points of our study:

(i) manufacturing strategy—the ''as-is'' operation versus the ''to-be'' model;

(ii) the accuracy of demand forecast at the end-product level versus at the component level;

(iii) the effect of mass customization as a result of, for instance, direct sales over the Internet.

To study the first factor we select a major product family at PCD, which consists of 18 end products assembled from a total of 17 components. We use PCD's existing data—including BOM, unit costs, assembly and procurement leadtimes—to run a detailed simulation model. The demand for each end product is statistically generated, based on historical data. The days-of-supply targets are set to meet a required service level of 95% (for all end products). Following PCD's current practice, these targets are determined using a combination of simple heuristic rules and judgement calls from product managers, and verified by simulation (via trial and error).

We then feed the same data, including the same statistically generated demand streams, into the optimization model, which eliminates the finished goods inventory at the end-product level and optimally sets the base-stock level for each component inventory. The optimization model minimizes the overall inventory investment while meeting the same service level of 95% for all end products. We take the optimal base-stock levels and rerun the simulation to verify the analytical results.

**Figure 6** **Comparison Between Build-to-Stock (''as-is'') and Assemble-to-Order (''to-be'') System**

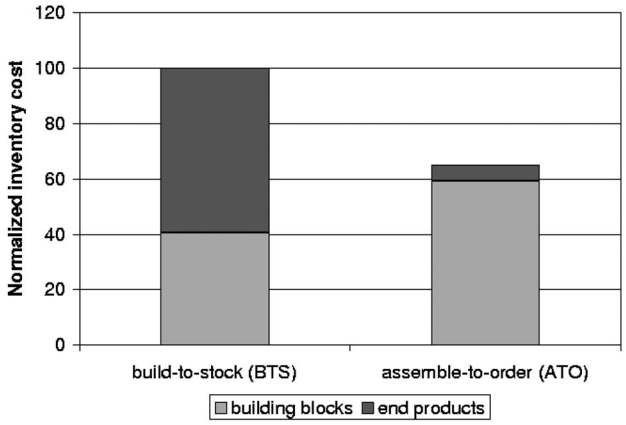



Figure 6 shows the comparison between the ''as-is'' and the ''to-be'' models in terms of the overall inventory investment. To protect proprietary information, the vertical axis in all figures is normalized with respect to the inventory investment of the ''as-is'' model, which is 100. As expected, the inventory investment at the end-product level is eliminated in the ''to-be'' model. (The cost shown is due to WIP; the cost due to finished goods is nil.) In contrast, the ''as-is'' model keeps a significant amount of end-product inventory. On the other hand, the amount of component inventory is higher in the ''to-be'' model, which is again expected because the required service level of 95% is common to both models. Overall, the ''to-be'' model reduces the overall inventory investment by about 30%.

Note that in the above study, both models use the same demand forecast at the end-product level. The ''to-be'' model, however, can easily switch to forecasting demand directly at the component level. This will result in improved forecast accuracy because we

**Figure 7 Effect of Improving Forecast Accuracy**

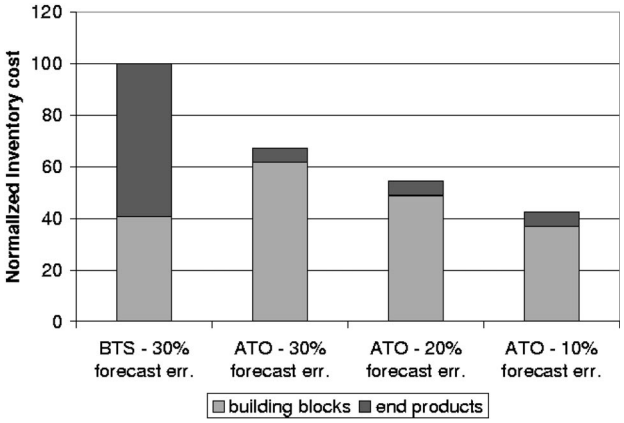
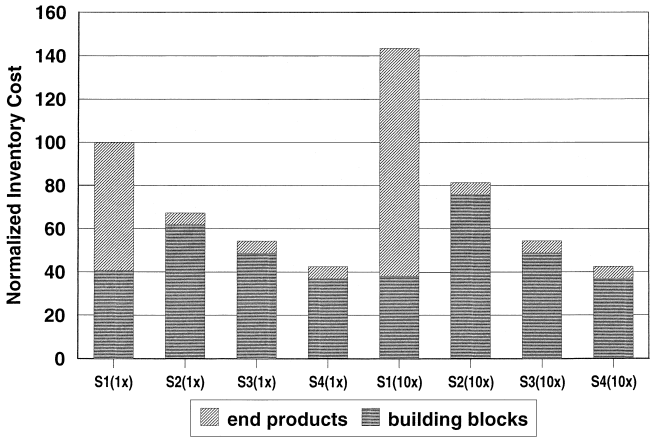



**Figure 8 Effect of Product Variety on Inventory**




can take advantage of parts commonality, as each component is generally used in several end products. Hence, in our study of the second factor we evaluate the effect of forecast accuracy through a sensitivity analysis. Figure 7 shows the overall inventory investment associated with three different levels of forecast accuracy. The first two columns repeat the comparison in the last figure, i.e., both ''as-is'' and ''to-be'' models assume 30% forecast error (i.e., the coefficient of variation equals 0.3) at the end-product level. The next two columns represent improved forecast errors, at 20 and 30%, achieved by the ''to-be'' model through forecasting at the component level.

Our study of the third factor aims at analyzing the impact on inventory when the system supports a richer product set, in terms of product variety. The motivation is to support mass customization: In an Internet-based direct-sales environment, for instance, the number of different product configurations that customers want to order can be significantly larger than what is currently supported in the build-to-stock environment. Figure 8 shows the inventory investments: The four columns on the left correspond to the current product set (1x), with the first scenario (S1) being the ''as-is'' model, and the other three being the ''to-be'' model at the current (S2) and improved (S3, S4) forecast accuracy levels, respectively. The four columns on the right repeat these scenarios with a product set that is 10 times as large in variety (10x), while maintaining the overall volume. (Also refer to Table 7 for a summary of all the different scenarios.)

**Table 7 Summary of the Scenarios Used to Study the Effect of Product Variety**

| Scenario | Description | 1× Cases | 10× Cases |
|---|---|---|---|
| S1 | ''as-is'' | original product set, 30% forecast error, 90% service | 10 times larger product set, 30% × $\sqrt{10}$ forecast error at MTM level |
| S2 | ''to-be'' | forecast at MTM level, 30% forecast error, 90% service | 10 times larger product set, forecast error as in S2(1×) |
| S3 | ''to-be'' | forecast at BB level, 20% forecast error, 9% service | 10 times larger product set, forecast error as in S3(1×) |
| S4 | ''to-be'' | forecast at BB level, 10% forecast error, 90% service | 10 times larger product set, forecast error as in S4(1×) |

Observe that as the product variety increases, a significantly higher level of inventory is required in the ''as-is'' model. This is because forecast accuracy will deteriorate when the end products proliferate (i.e., larger varieties at smaller volumes). On the other hand, in the ''to-be'' environment, the increase in inventory investment is very modest. This is because the proliferation of end products will have minimal effect on the forecast accuracy at the building-block level due to parts commonality. This

strongly supports the fact that the building-block model is the right process to support a direct-sales operation.

## 8. Concluding Remarks

We have developed an analytical model for the configure-to-order operation, which has become an important new business model in Internet commerce. We use the model to study the optimal inventory-service trade-off, formulated as a nonlinear programming problem with a set of constraints reflecting the service levels offered to different market segments. To solve the optimization problem, we have developed an exact algorithm for the important case of demand in each market segment having a unique component, and a greedy heuristic for the nonunique component case. We have also shown how to use the sensitivity analysis, along with simulation, to fine-tune the solutions. On the qualitative side, we have highlighted several key benefits of the CTO operation in terms of risk pooling. As part of a larger project aimed at the reengineering of IBM PCD's business process, we have applied our model to study three factors in terms of their impact on reducing inventory capital and enhancing customer service: (i) manufacturing strategy—the machine-type-model-based operation versus the building-block-based operation, (ii) the accuracy of demand forecast at the end-product (machine configurations) level versus at the market segment level; (iii) the effect of mass customization as a result of direct sales over the Internet.

### Acknowledgments

The authors thank the referees and the consulting Senior Editor, Professor Lawrance Wein, for their comments and suggestions that greatly improved this paper. They also thank Maike Schwarz for her assistance in obtaining the CTO simulation results, and Lisa Koenig of the IBM Personal Computing Division for her support and for providing them with the opportunity to work on this project.

### References

Aviv, Y., A. Federgruen. 1999. The benefits of design for postponement. S. Tayur, R. Ganeshan, and M. Magazine, eds. *Quantitative Models for Supply Chain Management.* Kluwer Academic Publishers, Norwell, MA, 553–584.

Brown, A., H. Lee, R. Petrakian. 2000. Xilinx improves its semiconductor supply chain using product and process postponement. *Interfaces* **30**(4) 65–80.

Cheung, K.L., W. Hausman. 1995. Multiple failures in a multi-item spare inventory model. *IIE Trans.* **27** 171–180.

Connors, D.P., D.D. Yao. 1996. Methods for job configuration in semiconductor manufacturing. *IEEE Trans. Semiconductor Manufacturing* **9** 401–411.

Dong, L., H.L. Lee. 2001. Efficient supply chain structures for personal computers. J.S. Song and D.D. Yao, eds. *Supply Chain Structures: Coordination, Information and Optimization.* Kluwer Academic Publishers, Norwell, MA, 7–44.

Ettl, M., G.E. Feigin, G.Y. Lin, D.D. Yao. 2000. A supply network model with base-stock control and service requirements. *Oper. Res.* **48** 216–232.

Gallien, J., L. Wein. 2001. A simple and effective component procurement policy for stochastic assembly systems. *Queueing Systems* **38** 221–248.

Garg, A., H.L. Lee. 1999. Managing product variety: An operations perspective. S. Tayur, R. Ganeshan, and M. Magazine, eds. *Quantitative Models for Supply Chain Management.* Kluwer Academic Publishers, Norwell, MA, 467–490.

Glasserman, P. 1999. Service levels and tail probabilities in multistage capacitated production-inventory systems. S. Tayur, R. Ganeshan, and M. Magazine, eds. *Quantitative Models for Supply Chain Management.* Kluwer Academic Publishers, Norwell, MA, 41–70.

——, Y. Wang. 1998. Leadtime-inventory tradeoffs in assemble-to-order systems. *Oper. Res.* **46** 858–871.

Hausman, W.H., H.L. Lee, A.X. Zhang. 1998. Order response time reliability in a multi-item inventory system. *Eur. J. Oper. Res.* **109** 646–659.

Lee, H. 1996. Effective inventory and service management through product and process redesign. *Oper. Res.* **44** 151–159.

Li, L. 1992. The role of inventory in delivery-time competition. *Management Sci.* **38** 182–197.

Mahajan, S., G.J. van Ryzin. 1999. Retail inventories and consumer choice. S. Tayur, R. Ganeshan, and M. Magazine, eds. *Quantitative Models for Supply Chain Management.* Kluwer Academic Publishers, Norwell, MA, 491–552.

Ross, S.M. 1998. *Stochastic Processes,* 2nd ed. Wiley, New York.

Song, J.S. 1998. On the order fill rate in a multi-item, base-stock inventory system. *Oper. Res.* **46** 831–845.

——. Evaluation of order-based backorders. Forthcoming, *Management Sci.*

——, D.D. Yao. Performance analysis and optimization in assemble-to-order systems with random leadtimes. Forthcoming, *Oper. Res.*

——, S. Xu, B. Liu. 1999. Order fulfillment performance measures in an assembly-to-order system with stochastic leadtimes. *Oper. Res.* **47** 131–149.

Swaminathan, J.M., S.R. Tayur. 1999. Stochastic programming mod-

els for managing product variety. S. Tayur, R. Ganeshan, and M. Magazine, eds. *Quantitative Models for Supply Chain Management.* Kluwer Academic Publishers, Norwell, MA, 585–624.

Vandaele, N.J., M.R. Lambrecht. 2001. Planning and scheduling in an assembly-to-order environment: Spicer-off-highway products division. J.S. Song and D.D. Yao, eds. *Supply Chain Structures: Coordination, Information and Optimization.* Kluwer Academic Publishers, Norwell, MA, 207–255.

Wang, Y. 1988. Service levels in production-inventory networks: Bottlenecks, tradeoffs, and optimization. Ph.D. dissertation, Columbia University, New York.

——. 2001. Leadtime, inventory, and service level in assemble-to-order systems. J.S. Song and D.D. Yao, eds. *Supply Chain Structures: Coordination, Information and Optimization.* Kluwer Academic Publishers, Norwell, MA, 311–357.

Xu, S.H. 2001. Dependence analysis of assemble-to-order systems. J.S. Song and D.D. Yao, eds. *Supply Chain Structures: Coordination, Information and Optimization.* Kluwer Academic Publishers, Norwell, MA, 359–414.

Zhang, A.X. 1997. Demand fulfillment rates in an assemble-to-order system with multiple products and dependent demands. *Production and Oper. Management* **6** 309–324.

Zipkin, P. 2000. *Foundations of Inventory Management.* Irwin/McGraw-Hill, New York.

*The consulting Senior Editor for this manuscript was Lawrence Wein. This manuscript was received on December 26, 2001, and was with the authors 27 days for 2 revisions. The average review cycle time was 14.5 days.*