RC 21781 Revised (98962) 3 May 2001 Computer Science/Mathematics 33 pages

IBM Research Report

Inventory-Service Optimization in Configure-to-Order Systems

Feng Cheng, Markus Ettl, Grace Lin

IBM T.J. Watson Research Center
Yorktown Heights, NY 10598

David D. Yao

IEOR Department, Columbia University
New York, NY 10027

June, 2000; revision: March, 2001

Research Division
Almaden · Austin · Beijing · Haifa · T.J. Watson · Tokyo · Zurich

Inventory-Service Optimization in Configure-to-Order Systems

Feng Cheng, Markus Ettl, Grace Lin
IBM Research Division, T.J. Watson Research Center
Yorktown Heights, NY 10598
fcheng,msettl,gracelin@us.ibm.com

David D. Yao*
IEOR Department, 302 Mudd Building
Columbia University, New York, NY 10027
yao@ieor.columbia.edu

June, 2000; revision: March, 2001

Abstract

This study is motivated by a process-reengineering problem in PC manufacturing, i.e., to move from a build-to-stock operation that is centered around end-product (machine type model) inventory, towards a configure-to-order (CTO) operation that eliminates end-product inventory — in fact, CTO has made irrelevant the whole notion of pre-configured machine types — and focuses instead on maintaining the right amount of inventory at the components. Indeed, 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 tradeoff 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 non-unique 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 also demonstrate that the model can generate considerable new insights into the key benefits of the CTO operation, in particular the impact of risk pooling and improved forecast accuracy.

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

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. The manufacturing process of personal computers (PC's) is a good example of such an environment. 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 appears to be 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. [6, 12]). In an ATO system, usually there is a pre-configured set of end-product types from which customers must choose. In contrast, a CTO system takes the ATO concept one step further, in allowing each customer to configure a product in terms of selecting a personalized set of components that go into the product. Aside from some consistency check on the product so configured, there is no "menu" of product types that limits the customer's choice.

Our study reported here is part of a larger project that aims at helping IBM's Personal Systems Group (PSG) to migrate from its existing operation to a CTO system. The PC manufacturing at PSG is traditionally a build-to-plan (or build-to-forecast) process, a process that is internally referred to as the MTM ("machine-type model") operation. There is a set of end products, or MTM's. Demand forecasts over a future planning horizon are generated for each MTM, and updated periodically for each planning cycle (typically, a weekly cycle). An MRP-type ("material requirements planning") of explosion technique is then used to determine the requirements for the components over the planning horizon based on the bill-of-materials (BOM) structure of each end product. Because of the random variation involved in demand forecasts, safety stock is usually kept for each end product between the channel partners and IBM in order to meet a desirable customer service level. However, holding finished goods inventory for any length of time is very costly in the PC business, where product life cycle is measured in months and price reduction takes place almost every other week.

To move from this business process to a web-based CTO operation (in particular, customer orders will be taken from the internet), the finished-goods inventory will be eliminated, and the emphasis will be shifted to the components, or "building blocks" (BB). Because of their long leadtimes, the BB's 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 cost 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 pre-configured MTM's. The optimization model we develop below provides an analytical tool to quantify these performance impacts.

A brief review of related literature for the ATO system is in order. Song [11] and Song *et al.* [12] focus on developing exact and approximate performance evaluation procedures that are computationally efficient. Glasserman and Wang [6] study the leadtime-inventory tradeoff using a large deviations approach, and derive a linear relationship between leadtime and inventory in the limiting sense of high fill rates. Wang [15] further applies this asymptotic result in an optimization problem to minimize average inventory holding cost with a constraint on the order fill-rate. Swaminathan and Tayur [14] 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 [1], Garg and Lee [5], Li [8], Mahajan and van Ryzin [9], and Zipkin [16].

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.* [4] 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 [3, 12, 13]. 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 [4].

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), 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. The effect of risk pooling and the comparison between CTO and ATO are also studied. Brief concluding remarks are presented in §7.

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 pre-specified 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 non-negative integer values. Denote:

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

That is, $\mathcal{S}_m$ denotes the set of components used in type m products, whereas $\mathcal{M}_i$ denotes all the product families that use component i. (Here, $X_{mi} \equiv 0$ means $\mathsf{P}(X_{mi} = 0) = 1$.)

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

- $L_i^{\rm in}$, $i \in \mathcal{S}$: the in-bound 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.

- $L_m^{\rm out}$, $m \in \mathcal{M}$: the out-bound 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:

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

where $X_{mi}(k)$, for $k = 1, 2, ...$, are i.i.d. copies of X_{mi}, and the inclusion of L_m^{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., [10]) and conditioning on L_m^{out}, we can derive then derive the mean and the variance of $D_i(t)$:

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

and

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

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. Let $I_i(t)$ and $B_i(t)$ denote the inventory and backorder levels in period t. A combination of standard inventory theory (e.g., [16]) and queueing analysis ([4]) yields the following results. Denote:

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

and

$$\mu_i(t - \ell, t) = \sum_{s=t-\ell}^{t} \mathsf{E}[D_i(s)], \tag{4}$$

$$\sigma_i^2(t - \ell, t) = \sum_{s=t-\ell}^{t} \mathsf{Var}[D_i(s)]. \tag{5}$$

Assuming that demands follow normal distributions, we write

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

where Z denotes the standard normal variate; and write

$$R_i(t)=\mu_i(t-\ell_i^{\mathrm{in}},t)+k_i(t)\cdot\sigma_i(t-\ell_i^{\mathrm{in}},t), \tag{6}$$

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

$$I_i(t)=[R_i(t)-D_i(t-\ell_i^{\mathrm{in}},t)]^+. \qquad \text{and} \qquad B_i(t)=[D_i(t-\ell_i^{\mathrm{in}},t)-R_i(t)]^+, \tag{7}$$

(Note that since time is discrete, we shall round *up* any real-valued ℓ_i^{in} to the next integer.) Furthermore, recall the following standard function in inventory theory (e.g., [16]):

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

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

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

We can then derive:

$$\mathsf{E}[I_i(t)]=\mathsf{E}[R_i(t)-D_i(t-\ell_i^{\mathrm{in}},t)]^+=\sigma_i(t-\ell_i^{\mathrm{in}},t)H(k_i(t)), \tag{10}$$

$$\mathsf{E}[B_i(t)]=\mathsf{E}[R_i(t)-D_i(t-\ell_i^{\mathrm{in}},t)]^+=\sigma_i(t-\ell_i^{\mathrm{in}},t)G(k_i(t)); \tag{11}$$

and

$$\mathsf{P}[I(t)=0]=\mathsf{P}[D_i(t-\ell_i^{\mathrm{in}},t)\geq R_i(t)]=\mathsf{P}[Z\geq k_i(t)]=\bar{\Phi}(k_i(t)). \tag{12}$$

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 μ_i part of $R_i(t)$ simply translates into ℓ_i^{in} periods (up to t) of demand. In addition, we can turn the safety-stock part of $R_i(t)$ into

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

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

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

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 (in-bound) 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):

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

We can then write

$$R_i = \ell_i^{\rm in}\mathsf{E}[D_i] + k_i\sqrt{\ell_i^{\rm in}}\mathsf{sd}[D_i]; \tag{15}$$

and hence,

$$\text{DOS}_i = R_i/\mathsf{E}[D_i] = \ell_i^{\rm in} + k_i\theta_i\sqrt{\ell_i^{\rm in}} = \ell_i^{\rm in}[1 + k_i\frac{\theta_i}{\sqrt{\ell_i^{\rm in}}}], \tag{16}$$

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

Sometimes it is more appropriate to adjust the demand distribution to account for non-negativity. 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):

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

To derive the adjusted variance, note the following:

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

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

$$\begin{aligned} \mathsf{Var}[\tilde{D}] &= \sigma^2\mathsf{Var}\{[Z - (-\frac{\mu}{\sigma})]^+\} \\ &= \sigma^2\mathsf{E}\{[(Z - (-\frac{\mu}{\sigma}))^+]^2\} - [\mathsf{E}(\tilde{D})]^2 \\ &= \sigma^2[\bar{\Phi}(-\frac{\mu}{\sigma}) + \frac{\mu}{\sigma}G(-\frac{\mu}{\sigma}) - G^2(-\frac{\mu}{\sigma})]. \end{aligned} \tag{18}$$

For moderately large x (say, $x \geq 2$), from (8), we have $G(-x) \overset{\sim}{=} x$, and hence

$$\mathsf{E}[\tilde{D}] \overset{\sim}{=} \mathsf{E}[D], \qquad \mathsf{Var}[\tilde{D}] \overset{\sim}{=} \mathsf{Var}[D],$$

from (17) and (18). Therefore, the above adjustment is only needed when the coefficient of variation of the demand, σ/μ, 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 non-stationary 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 of each order of type m requires exactly one unit of component $i \in \mathcal{S}_m$. Let α 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

$$\mathsf{P}[\cup_{i\in\mathcal{S}_m} E_i] \leq 1 - \alpha.$$

Making use of the well-known inclusion-exclusion formula (e.g., [10]):

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

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

$$\mathsf{P}[\cup_{i\in\mathcal{S}_m} E_i] \overset{\sim}{=} \sum_{i\in\mathcal{S}_m} \mathsf{P}(E_i) = \sum_{i\in\mathcal{S}_m} \bar{\Phi}(k_i) \leq 1 - \alpha. \tag{19}$$

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:

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

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., [10] 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

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

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 $\mathsf{P}(A)$. Then the no-stockout probability, $\prod_{i\in A}\Phi(k_i)$, should be weighted by $\mathsf{P}(A)$. Hence, the service requirement in (20) should be changed to

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

Since

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

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

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

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. (12)]. To see this, first consider the case of unit (demand) arrivals, i.e., $X_{mi}\equiv 1$. Then, the stockout probability is $\mathsf{P}[D_i>R_i]$, or $\mathsf{P}[D_i+1\ge R_i]$. In the general case of batch arrivals, the stockout probability is

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

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 μ_i and σ_i following (14); and the batch size of an incoming demand is usually orders of magnitude smaller when compared against μ_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 by the overestimation involved in (20), since 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:

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

where w_m and α 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 $L_m^{\rm out}$, the out-bound 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 $L_m^{\rm out} + \tau_i$, where τ_i is the additional delay before the stocked-out component becomes available.

Hence, we can write

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

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

On the other hand, τ_i, which is quite intractable, can be approximated as follows (refer to [4], eqn (7)):

$$\tau_i \approx \ell_i^{\rm in} \cdot \frac{\mathsf{E}(B_i)}{\bar{\Phi}(k_i)(R_i+1)}.$$

From (11), we have

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

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

$$\begin{aligned} \tau_i &\approx \ell_i^{\rm in} \cdot \frac{\sigma_i}{k_i(R_i+1)} \\ &\leq \ell_i^{\rm in} \cdot \frac{\sigma_i}{k_i R_i} \\ &= \frac{\ell_i^{\rm in}\sqrt{\ell_i^{\rm in}}\mathsf{sd}(D_i)}{k_i[\ell_i^{\rm in}\mathsf{E}(D_i) + k_i\sqrt{\ell_i^{\rm in}}\mathsf{sd}(D_i)]} \\ &= \frac{\ell_i^{\rm in}}{k_i\sqrt{\ell_i^{\rm in}}\mathsf{E}(D_i)/\mathsf{sd}(D_i) + k_i^2}. \end{aligned} \qquad (26)$$

The above allows us to set w_m such that $w_m \geq L_i^{\rm out} + \tau_i$, 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 τ_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 cost, subject to meeting the service requirement for each product family as expressed in (23). The problem can be presented as follows:

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

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 α_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 σ_i follows the specification in (14).

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

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

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:

$$\min \sum_{i \in \mathcal{S}} c_i \sigma_i H(k_i) \tag{27}$$

$$\text{s.t.} \sum_{i \in \mathcal{S}_m} r_{mi} \Phi(k_i) \geq 1, \qquad m \in \mathcal{M}. \tag{28}$$

Several remarks are in order:

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

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

(iii) Suppose a product family m involves a *unique* component, denoted i_m, i.e., $i_m \notin \mathcal{S}_{m'}$ when $m' \neq m$. 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 (since $H(\cdot)$ is an increasing function as explained in (i)).

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 $\lambda_m \geq 0$, $m \in \mathcal{M}$, 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 non-linear equations that characterizes the optimal solution:

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

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

5.1 Unique Components

While solving the above system of non-linear equations is quite intractable in general, in the following important case, we do have an efficient algorithm that solves the non-linear equations and generate 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 (iii), all the constraints in (28) must be binding; i.e., the equations in (32) holds for all $m\in\mathcal{M}$, since 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 compoments, k_{i_m}, $m\in\mathcal{M}$. Then, all the Lagrangian multipliers, λ_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,

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

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 non-unique 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 non-unique 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.

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 λ_m following (33).

3. For each non-unique 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 $\mathrm{LHS}_m > 1 + \epsilon$, set $y_m^H \leftarrow y_m$ and $y_m \leftarrow (y_m + y_m^L)/2$;
   - if $\mathrm{LHS}_m < 1 + \epsilon$, 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 1. 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., a increase (resp. decrease) in y_m results in an increase (resp. decrease) in $r_{mi_m} y_m + \sum_{i \neq i_m} r_{mi} h_{mi}(y)$, 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

Although unique components is a prevailing case in CTO, it is still useful to examine the case when there is not necessarily a unique component in every product family. This will become useful specially for the ATO system, where there is a set or pre-configured end-products. We can treat the ATO system as a special case of the CTO by modeling each end-product as a customer segment. But then, it is not likely that each end-product will have a unique component. (Refer to the example in §6.3.)

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 increase the left hand side of all the constraints to 1; at each step, we identify a variable k_i such that incrementing its value will yield the largest decrease in the objective value.

At each step in the algorithm, let β_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 δ. We want to indentify

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

where

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

specifies how much x_i can be increased (without exceeding the current value of the constraint δ). Since δ_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:

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 $i^* = \arg\max_{i \in A} \{c_i \sigma_i \delta_i \Phi(k_i + \delta_i/2)\}$, where δ_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 Tradeoff

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 ϵ 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

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

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 cost by an amount d can be achieved by reducing the service requirement of product family m, provided $\lambda>0$, from α_m to

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

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 α_m values (required service levels) that correspond to reducing the objective value by say, 10%; and then re-solve the optimization problem using the reduced α_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.

FIGURE 1 ABOUT HERE

All components used in the assembly of an end product are purchased from external suppliers. The supplier lead times are deterministic with $\mathsf{E}(L_i^{\text{in}})$ 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 lead time, the system fits well within the framework discussed in §4.

					$\mathsf{P}(X_{mi} > 0)$		
i	category	component	lead time $\mathsf{E}(L_i^{\text{in}})$	unit cost c_i	low-end	mid-end	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	Pre-load A	4	90	0.7	0.5	0.3
9	software	Pre-load 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 1: Bill-of-materials structure for example configure-to-order system.

We consider 3 customer segments, i.e., $m = 1, 2, 3$, representing low-end, mid-end 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-end 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 $\mathsf{E}(D_m) = 100$, and coefficient of variation $\mathsf{cv}(D_m) = 0.25$ 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 the bisection procedure is guaranteed to converge, but it may converge to a sub-optimal solution. More specifically, the convergent point (solution) is optimal if and only if all the service constraints are binding. Hence, in Table 2 1, 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 α^*_{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 sub-optimal, reflected by the gaps between α and α^*_{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 the feasible points. We can see that the objective values match very well. In particular, in the two cases where the algorithm over-achieves the require service level, the objective value is only slightly above the best value returned by the search procedure.

	$\text{cv}(D_m) = 0.25$				$\text{cv}(D_m) = 0.50$			
α	α^*_{analy}	z^*_{analy}	z^*_{search}	rel.dev.	α^*_{analy}	z^*_{analy}	z^*_{search}	rel.dev.
.800	(.800,.800,.800)	437,637	–	–	(.800,.800,.800)	875,273	–	–
.820	(.820,.819,.818)	450,311	451,121	0.2%	(.820,.819,.818)	900,621	902,243	0.2%
.840	(.839,.837,.840)	462,308	463,088	0.2%	(.839,.837,.840)	924,616	926,176	0.2%
.860	(.860,.860,.860)	477,489	–	–	(.860,.860,.860)	954,978	–	–
.880	(.880,.880,.880)	494,050	–	–	(.880,.880,.880)	988,100	–	–
.900	(.901,.900,.900)	513,383	512,050	-0.3%	(.901,.900,.900)	1,026,766	1,024,199	-0.3%
.920	(.920,.920,.920)	536,004	–	–	(.920,.920,.920)	1,072,007	–	–
.940	(.940,.940,.940)	564,446	–	–	(.940,.940,.940)	1,128,892	–	–
.960	(.960,.960,.960)	602,862	–	–	(.960,.960,.960)	1,205,723	–	–
.980	(.980,.980,.980)	664,478	–	–	(.980,.980,.980)	1,328,956	–	–

Table 2: Quality of the algorithm: identical service target for all customer segments.

Table 3 extends the same study to allow different service requirements for different customer segments (one demand scenario only). 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 $\epsilon = 10^{-6}$.

			cv$(D_m) = 0.50$			
α_1	α_2	α_3	α^*_{analy}	z^*_{analy}	z^*_{search}	rel.dev.
.920	.950	.900	(.921,.950,.900)	1,083,953	1,085,977	0.2%
.920	.950	.920	(.920,.950,.920)	1,102,866	–	–
.920	.950	.940	(.919,.950,.940)	1,128,973	1,131,144	0.2%
.920	.950	.960	(.919,.950,.960)	1,168,104	1,172,183	0.3%
.920	.950	.980	(.918,.948,.980)	1,239,232	1,244,627	0.4%
.920	.900	.980	(.918,.908,.980)	1,215,895	1,217,521	0.1%
.920	.920	.980	(.919,.925,.980)	1,224,201	1,226,235	0.2%
.920	.940	.980	(.916,.943,.980)	1,234,501	1,237,110	0.2%
.920	.960	.980	(.920,.964,.980)	1,256,906	1,255,050	-0.1%
.920	.980	.980	(.923,.980,.980)	1,292,961	1,290,429	-0.2%
.900	.950	.980	(.904,.947,.980)	1,233,337	1,238,372	0.4%
.920	.950	.980	(.918,.948,.980)	1,239,232	1,244,627	0.4%
.940	.950	.980	(.942,.949,.980)	1,251,034	1,254,381	0.3%
.960	.950	.980	(.958,.950,.980)	1,263,566	1,267,927	0.3%
.980	.950	.980	(.980,.955,.980)	1,298,052	1,297,527	0.0%

Table 3: Quality of the algorithm: different service targets for different customer segments.

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-end 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., over-achieve the service requirements). Overall, however, the analytical curves track the simulated curves quite closely, and the gap decreases as the service requirement increases.

FIGURES 2-4 ABOUT HERE

Next, we continue examining the second issue raised above, but from a different angle: Since the lower bounds involved in the service constraints over-achieve 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, where the column under α^*_{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 α_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. Furthermore, 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 cost) is reported under the columns labelled α^*_{search} and z^*_{search}. The relative difference between z^*_{analy} and z^*_{search} is reported in the column labelled Δ_z. We applied the same study to the cases in Table 3, with the results summarized in Table 5.

	cv$(D_m) = 0.25$					cv$(D_m) = 0.50$				
α	α^*_{analy}	z^*_{analy}	α^*_{search}	z^*_{search}	Δ_z	α^*_{analy}	z^*_{analy}	α^*_{search}	z^*_{search}	Δ_z
.800	.890	437,636	.805	372,116	15.0%	.893	875,272	.813	744,232	15.0%
.820	.896	450,310	.839	385,133	14.5%	.902	900,620	.828	767,392	14.8%
.840	.908	462,307	.852	398,653	13.8%	.911	924,614	.844	792,946	14.2%
.860	.919	477,492	.877	414,776	13.1%	.919	954,983	.865	823,529	13.8%
.880	.928	494,050	.893	428,233	13.3%	.928	988,099	.895	856,465	13.3%
.900	.939	513,383	.915	452,212	11.9%	.939	1,026,766	.915	904,428	11.9%
.920	.950	536,003	.934	477,074	11.0%	.949	1,072,007	.931	954,148	11.0%
.940	.962	564,446	.947	503,809	10.7%	.961	1,128,892	.950	1,011,283	10.4%
.960	.974	602,861	.969	553,908	8.1%	.973	1,205,723	.968	1,107,816	8.1%
.980	.985	664,477	.982	610,014	8.2%	.985	1,328,955	.982	1,220,027	8.2%

Table 4: Comparisons between the analytical and the search solutions: identical service target for all customer segments.

As the numbers in the two Tables 4 and 5 illustrate, the objective values corresponding to the analytical solution are about 8-15% sub-optimal, 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 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 buffers from which the demand in all customer segments can draw upon (if the component is needed). Hence,

α_1	α_2	α_3	α^*_{analy}	z^*_{analy}	α^*_{search}	z^*_{search}	Δ_z
.920	.950	.900	(.947, .969, .930)	1,083,953	(.925, .959, .901)	922,025	14.9%
.920	.950	.920	(.945, .968, .945)	1,102,866	(.925, .961, .920)	948,462	14.0%
.920	.950	.940	(.945, .966, .962)	1,128,973	(.932, .951, .941)	978,332	13.3%
.920	.950	.960	(.942, .964, .977)	1,168,104	(.926, .955, .966)	1,013,516	13.2%
.920	.950	.980	(.940, .958, .989)	1,239,232	(.927, .953, .982)	1,077,292	13.1%
.920	.900	.980	(.944, .919, .991)	1,215,895	(.924, .902, .980)	1,046,427	13.9%
.920	.920	.980	(.943, .936, .990)	1,224,201	(.924, .921, .986)	1,112,248	9.1%
.920	.940	.980	(.940, .953, .989)	1,234,501	(.922, .940, .983)	1,123,316	9.0%
.920	.960	.980	(.941, .972, .988)	1,256,906	(.920, .960, .981)	1,094,290	12.9%
.920	.980	.980	(.938, .986, .986)	1,290,429	(.926, .981, .981)	1,114,633	13.6%
.900	.950	.980	(.930, .958, .990)	1,233,337	(.901, .951, .981)	1,071,165	13.1%
.920	.950	.980	(.940, .958, .989)	1,239,232	(.927, .953, .982)	1,077,292	13.1%
.940	.950	.980	(.959, .956, .989)	1,251,034	(.941, .955, .981)	1,118,018	10.6%
.960	.950	.980	(.972, .955, .988)	1,263,566	(.962, .958, .981)	1,141,255	9.7%
.980	.950	.980	(.987, .956, .987)	1,297,527	(.985, .956, .980)	1,129,761	12.9%

Table 5: Comparisons between the analytical and the search solutions: different service targets for different customer segments.

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 give service target can be lowered by this aggregation, or *risk pooling*. In general, the impact of risk pooling become more significant when the number of product configurations is large and the correlation among demands is small (e.g., Aviv and Federgruen [1], Brown *et al.* [2], Lee [7]).

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 costs over all three customer segments. The comparison against the risk-pooling results are plotted in Figure 5.

FIGURE 5 ABOUT HERE

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 significant reduction of inventory investment: For the same service target, the expected inventory cost 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%.

		days-of-supply (risk-pooling)	days-of-supply (no risk-pooling)		
i	component	all segments	low-end	mid-end	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	3.5	–	–
4	500 MHz board	3.0	–	3.3	–
5	600 MHz board	2.9	–	–	3.0
6	7GB disk drive	3.2	4.2	4.7	–
7	13GB disk drive	3.2	–	3.7	4.4
8	Pre-load A	2.3	2.3	2.7	3.1
9	Pre-load B	2.3	3.0	2.7	2.5
10	CD ROM	3.3	3.7	3.8	4.0
11	Video graphics card	2.5	–	3.6	3.2
12	Ethernet card	3.4	–	4.8	4.2

Table 6: Comparison between risk-pooling and no risk-pooling scenario, service target $\alpha = 0.90$ for all customer segments.

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, the 128MB memory card, which is used in all customer segments, requires 2.4 days-of-supply, whereas the Ethernet card, which is used in the high-end and mid-end segments, only requires 3.4 days-of-supply.

6.3 Comparison between CTO and ATO

As we mentioned in the introductory section, ATO differs from CTO in that it has a set of pre-configured end-products for customers to choose from. Conceptually, we can treat ATO as a special case of CTO, by letting each end-product represent a customer segment, associated with a degenerate distribution of $\{X_{mi}\}$ (i.e., with probability one for the designated configuration).

Hence, it would appear tempting to conclude that ATO can achieve the same level of service targets at a lower inventory costs. After all, ATO supports a smaller variety of product types, and it does not have the additional demand variation as CTO at the configuration level (i.e., those associted with $\{X_{mi}\}$). This argument, however, has ignored an important fact: In a CTO system, demand forecast is performed on customer segments, whereas in an ATO system demand forecast is done by end-product types. And it is widely known in industry that forecast by customer segments is generally more accurate, in terms of a smaller forecast error (typically measured by the coefficient of variation of the demand).

Therefore, in order to make a sensible comparison between the two systems, we need to first specify the relationship between their associated demand forecast errors.

Let D^{CTO} denote the demand per time unit of a product family (or customer segment) in the CTO system. Let D_c^{ATO} denote the demand for a pre-configured end-product c in the ATO system. Let c also be a member of the product family in the CTO system. Then, to make a fair comparison between the two systems, it is reasonable to assume the following relations:

$$\mathsf{E}\left(D^{\mathrm{CTO}}\right) = \mathsf{E}\left(\sum_c D_c^{\mathrm{ATO}}\right), \qquad \mathsf{Var}\left(D^{\mathrm{CTO}}\right) = \mathsf{Var}\left(\sum_c D_c^{\mathrm{ATO}}\right). \tag{37}$$

To satisfy the above relations, we can further assume:

$$\mathsf{E}\left(D_c^{\mathrm{ATO}}\right) = \pi_c \mathsf{E}\left(D^{\mathrm{CTO}}\right), \qquad \mathsf{Var}\left(D_c^{\mathrm{ATO}}\right) = \pi_c \mathsf{Var}\left(D^{\mathrm{CTO}}\right); \tag{38}$$

where

$$\pi_c > 0, \qquad \text{and} \qquad \sum_c \pi_c = 1.$$

From (38) it follows that the two coefficients of variation relate as follows:

$$\mathsf{cv}\left(D_c^{\mathrm{ATO}}\right) = \frac{1}{\sqrt{\pi_c}} \quad \mathsf{cv}\left(D^{\mathrm{CTO}}\right).$$

Since $\pi_c \leq 1$, the above does capture the fact that ATO results in a larger forecast error (for each end-product configuration) than CTO (for a customer segment).

Now, consider the demand for a certain component i over the leadtime, denoted D_i; and for simplicity, assume the leadtime is a constant, 1. Consider the CTO case, and suppose each order requires one (resp. zero) unit of i with probability p_i (resp. $1 - p_i$). Then,

$$\mathsf{Var}\left(D_i^{\mathrm{CTO}}\right) = \mathsf{E}\left(D^{\mathrm{CTO}}\right) p_i(1 - p_i) + \mathsf{Var}\left(D^{\mathrm{CTO}}\right) p_i^2.$$

On the other hand, in the ATO case, suppose each order of configuration c requires one unit of component i. Then, from (38), we have

$$\mathsf{Var}\left(D_i^{\mathrm{ATO}}\right) = \mathsf{Var}\left(D_c^{\mathrm{ATO}}\right) = \pi_c \mathsf{Var}\left(D^{\mathrm{CTO}}\right).$$

To enforce the relations in (37), let $p_i = \pi_c$. Then, we can verify from the above relations that

$$\mathsf{Var}\left(D_i^{\mathrm{CTO}}\right) \leq \mathsf{Var}\left(D_i^{\mathrm{ATO}}\right) \qquad \text{iff} \qquad \mathsf{E}\left(D^{\mathrm{CTO}}\right) \leq \mathsf{Var}\left(D^{\mathrm{CTO}}\right). \tag{39}$$

The above condition must hold in most cases. Exceptions will have to be some very special case in which D^{CTO} has zero or negligible forecast errors. The above also holds when the leadtime is random (just apply conditioning on the leadtime).

We use the above formalism to compare a CTO system with a corresponding ATO system. The CTO system is taken from the scenario of demand with a coefficient of variation of 0.5 in Table 2. For the comparable ATO system, we take two machine configurations from the low-end segment, and four configurations each from the mid-end and high-end segments. The π_c values are: 0.7 and 0.3 for the low-end machines; 0.1, 0.4, 0.2, and 0.3 for the mid-end machines; and 0.2, 0.3, 0.3, and 0.2 for the high-end machines. In addition, we normalize the costs so that the average total component cost per order is the same in both systems.

FIGURES 6-7 ABOUT HERE

In Figure 6, we report the expected safety stock in days-of-supply, and the service levels, both from the analytical model. To ensure a fair comparison, we use the greedy heuristic described in §5.2 to generate the solutions to both systems, since the ATO system does not have unique-components. (In this case, the solution generated by the heuristic for the CTO system is virtually identical to the optimal solution generated by the algorithm in §5.1.) We then simulate the actual service levels achieved by the analytical solutions, for both systems, and the results are reported in Figure 7. In both figures, the CTO system is superior to the ATO system, in terms of a better inventory-service tradeoff: at the same service level, it results in a lower inventory cost. (Note that the gap between the two systems is in fact wider in the simulated curves in Figure 7.) The key to this result is the lower component variability in the CTO system as explained in the inequality in (39).

7 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 tradeoff, 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 non-unique 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 new insights to some of the key benefits of the CTO operation, in terms of risk pooling and improved forecast accuracy.

As mentioned in the Introduction, our study was part of a larger project aimed at the reengineering of IBM/PSG's business process. Specifically, 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. Details of the study will be reported elsewhere.

Acknowledgements

We thank Maike Schwarz for her assistance in obtaining the CTO simulation results. We also thank Lisa Koenig of the IBM Personal Systems Group for providing us with the opportunity to work on this problem.

References

[1] AVIV, Y. AND FEDERGRUEN, A., The Benefits of Design for Postponement. In: *Quantitative Models for Supply Chain Management*, S. Tayur, R. Ganeshan and M. Magazine (eds.), Kluwer Academic Publishers, Norwell, 1999, 553-584.

[2] BROWN, A., LEE, H., AND PETRAKIAN, R., Xilinx improves its Semiconductor Supply Chain using Product and Process Postponement, *Interfaces*, 30 (4), 2000, 65-80.

[3] CONNORS, D.P. AND YAO, D.D., Methods for Job Configuration in Semiconductor Manufacturing, *IEEE Trans. Semiconductor Manufacturing*, **9** (1996), 401-411.

[4] ETTL, M., FEIGIN, G.E., LIN, G.Y., AND YAO, D.D., A Supply Network Model with Base-Stock Control and Service Requirements, *Operations Research*, **48** (2000), 216-232.

[5] GARG, A. AND H.L. LEE, Managing Product Variety: an Operations Perspective. In: *Quantitative Models for Supply Chain Management*, S. Tayur, R. Ganeshan and M. Magazine (eds.), Kluwer Academic Publishers, Norwell, 1999, 467-490.

[6] GLASSERMAN, P. AND WANG, Y., Leadtime-Inventory Tradeoffs in Assemble-to-Order Systems, *Operations Research*, **46** (1998), 858-871.

[7] LEE, H., Effective Inventory and Service Management Through Product and Process Redesign, *Operations Research,* **44** (1996), 151-159.

[8] LI, L., The Role of Inventory in Delivery-Time Competition. *Management Science,* **38** (1992), 182-197.

[9] MAHAJAN, S. AND G.J. VAN RYZIN, Retail Inventories and Consumer Choice. In: *Quantitative Models for Supply Chain Management*, S. Tayur, R. Ganeshan and M. Magazine (eds.), Kluwer Academic Publishers, Norwell, 1999, 491-552.

[10] ROSS, S.M., *Stochastic Processes,* 2nd ed., Wiley, New York, 1998.

[11] SONG, J.S., On the Order Fill Rate in a Multi-Item, Base-Stock Inventory System, *Operations Research,* **46** (1998), 831-845.

[12] SONG, J.S., XU, S. AND LIU, B. Order Fulfillment Performance Measures in an Assembly-to-Order System with Stochastic Leadtimes, *Operations Research*, **47** (1999), 131-149.

[13] SONG, J.S. AND D.D. YAO, Performance Analysis and Optimization in Assemble-to-Order Systems with Random Leadtimes, submitted for publication, 2000.

[14] SWAMINATHAN, J.M. AND S.R. TAYUR, Stochastic Programming Models for Managing Product Variety. In: *Quantitative Models for Supply Chain Management,* S. Tayur, R. Ganeshan and M. Magazine (eds.), Kluwer Academic Publishers, Norwell, 1999, 585-624.

[15] WANG, Y. *Service Levels in Production-Inventory Networks: Bottlenecks, tradeoffs, and Optimization*, Ph.D. Dissertation, Columbia University, 1988.

[16] ZIPKIN, P., *Foundations of Inventory Management,* Irwin/McGraw-Hill, New York, 2000.

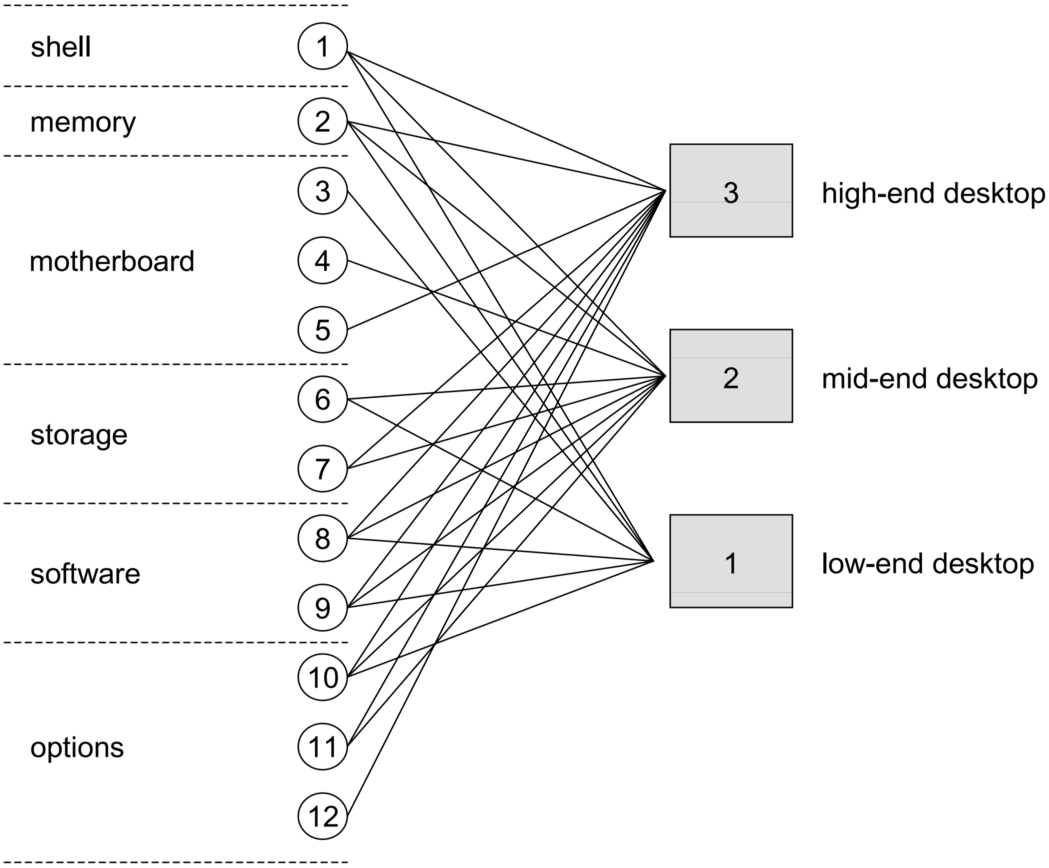

Figure 1: A configure-to-order system with three product families.

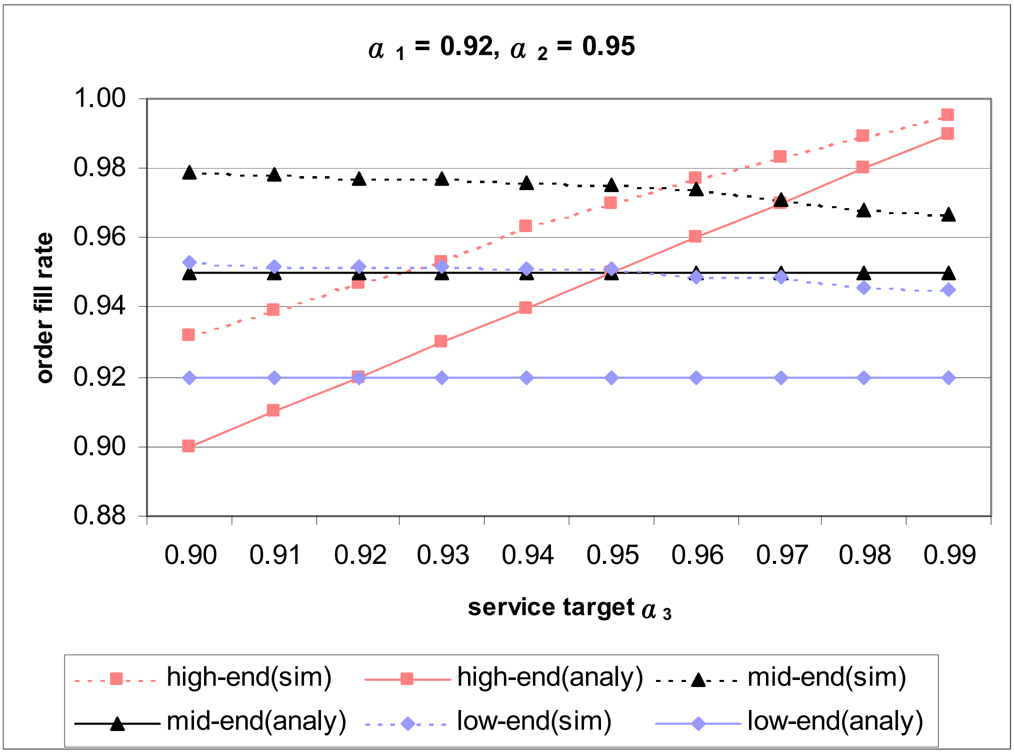

Figure 2: Comparison between analytical and simulated fill rates.

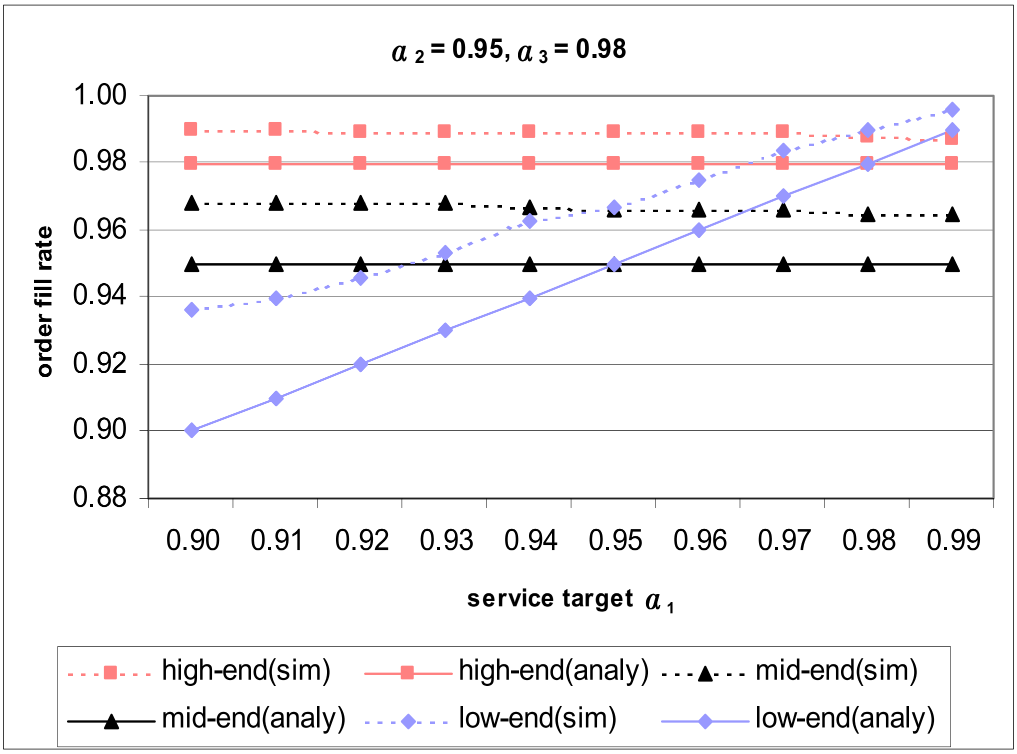

Figure 3: Comparison between analytical and simulated fill rates.

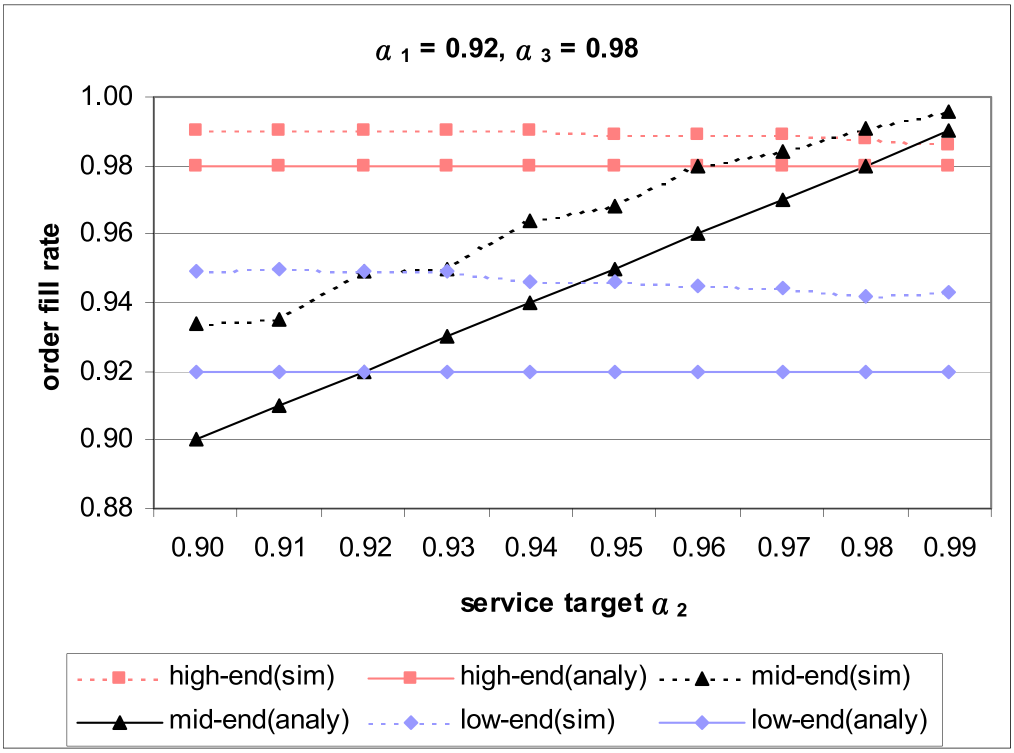

Figure 4: Comparison between analytical and simulated fill rates.

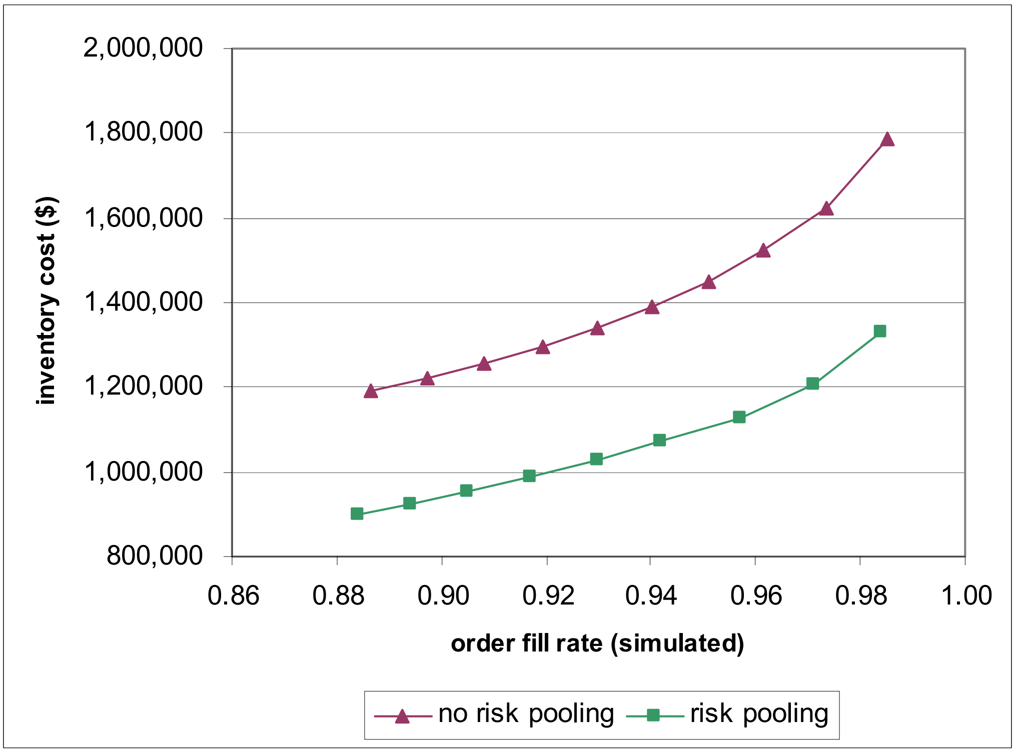

Figure 5: Effect of risk pooling.

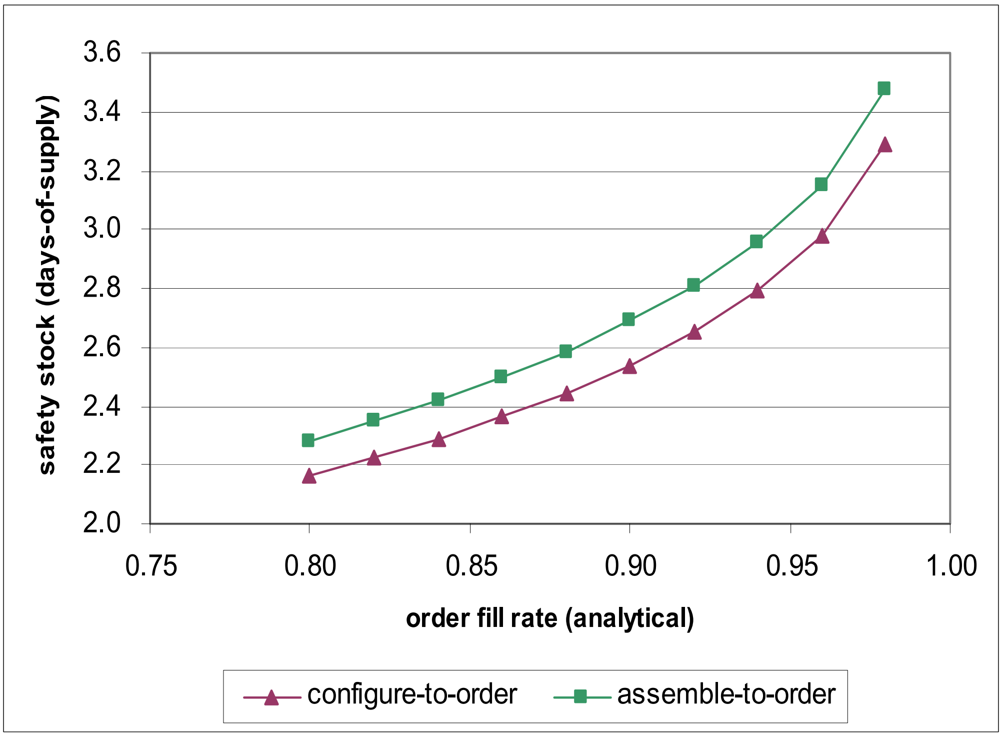

Figure 6: Comparison between CTO and ATO; analytical fill rates.

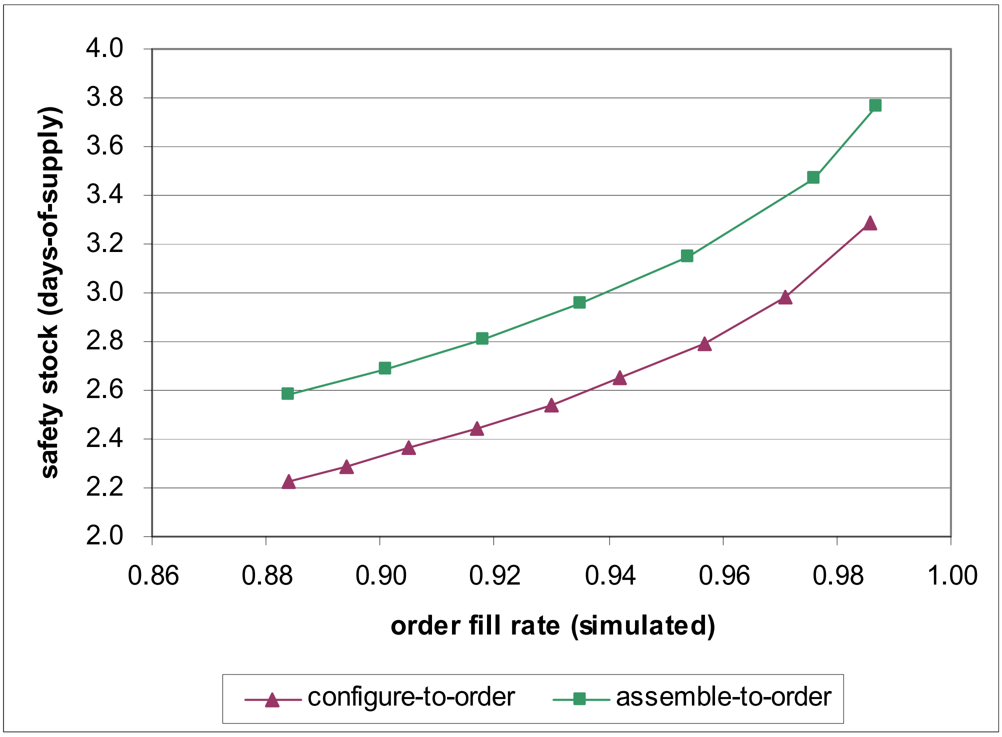

Figure 7: Comparison between CTO and ATO; simulated fill rates.