A Two-Stage Push-Pull Production Planning Model

Feng Cheng
IBM T.J. Watson Research Center
Yorktown Heights, NY 10598
E-mail: fcheng@us.ibm.com

Markus Ettl
IBM T.J. Watson Research Center
Yorktown Heights, NY 10598
E-mail: msettl@us.ibm.com

Yingdong Lu
IBM T.J. Watson Research Center
Yorktown Heights, NY 10598
E-mail: yingdong@us.ibm.com

David D. Yao
IEOR Department, 302 Mudd Building
Columbia University, New York, NY 10027
E-mail: yao@columbia.edu

A Two-Stage Push-Pull Production Planning Model

Abstract

We study a hybrid push-pull manufacturing system that the IBM Systems Group implemented as a response to a complex configuration environment and increasing pressure for responsive order fulfillment lead times. This is a two-stage (fabrication and fulfillment center) manufacturing process, which builds and stocks tested subassemblies for just- in-time configuration of the final product when a specific customer order is received. The first production stage (fabrication) is a push process where parts are replenished, tested, and assembled into subassemblies according to product-level build plans. The subassembly inventory is kept in stock ready for the final assembly of the end products. The second production stage (fulfillment) is a pull-based assemble-to-order process where the final assembly process is initiated when a customer order is received and no finished goods inventory is kept for end products. One important planning issue is to find an optimal trade-off between capacity utilization and inventory cost reduction that strives to meet the quarter-end peak demand. We present a nonlinear optimization model to minimize the total inventory cost subject to the service level constraints and the production capacity constraints. This results in a convex program with linear constraints. An efficient algorithm using decomposition is developed for solving the nonlinear optimization problem. Other numerical methods including a piece-wise linearization method and a general purpose nonlinear solver are also tested along with the decomposition method. Several variations of the model are formulated to incorporate additional features in practice. Numerical results are presented to show the performance improvement generated by the optimal solutions over a production-smoothing strategy.

Keywords: push-pull production, configure-to-order, production smoothing, capacitated inventory model.

1. Introduction

Server computer manufacturing is highly capital-intensive and characterized by high customer expectations, short product life cycles, proliferating product variety, unpredictable demand, long and variable manufacturing cycle times, and considerable supply chain complexity. On the one hand, manufacturing firms need to improve the manufacturing efficiency by maximizing the utilization of production capacity and by minimizing the inventory costs. One the other hand, they need to deal with increasingly complex configuration requirements and tremendous pressure for responsive order fulfillment.

The emphasis of this paper is to provide decision support for supply chain operations in a hybrid push-pull system with multiple products and complex bills-of-materials. Products are assembled in two stages called fabrication and fulfillment. We formulate a multiple-period production-inventory optimization problem with service level constraints defined at the product level for each period. The goal of the optimization is to generate a build plan that minimizes the inventory cost while meeting customer demand with a given service level in a capacity- constrained environment. Capacity limitations are modeled in the form of machine and labor capacity constraints. Labor capacity can be shared among different manufacturing tasks. The total labor capacity can be adjusted according to demand by using overtime work force at a premium cost. Machine capacity can be shared only among tasks that involve manufacturing or testing of similar components.

Below in the rest of this section, we begin with a review of related literature, followed by a description of the industrial problem that motivated our study, and an overview of the contributions of this paper. In Section 2, we define a product-based optimization model where configured systems are built in a squared set, that is, all components in the bill-of-materials of a system need to be built and replenished at the same time. In Section 3, we introduce a component-based optimization model where the squared set restriction is removed to allow additional flexibility when optimizing the build plan. In Section 4 we develop a linear approximation approach and a decomposition algorithm to provide efficient solutions. Numerical results are presented in Section 5, and several extensions of the model are presented in Section 6. The paper concludes with a summary and directions for future research.

1.1 Literature Review

Although there is a large body of literature on pure push systems and pure pull systems, research on hybrid systems is relatively sketchy. Some cases of hybrid systems have been studied in specialized contexts. For example, Aviv and Federgruen (2001) provide an analysis on capacitated multi-item inventory systems where the items are produced in two stages. They investigate the benefits of various delayed product differ-

entiation (postponement) strategies and other related issues in a two-stage production system. The typical setting of two-stage production systems discussed in the postponement literature usually involves a common intermediate product in the first stage and a simple localization operation in the second stage.

Federgruen and Zipkin (1986a, 1986b) establish the optimality of the modified base-stock policy in a discrete-time single item, single location finite capacitated production system under an average cost criterion and a discounted cost criterion. Aviv and Federgruen (1997) allow the values of parameters such as demand distributions, capacity levels, and costs, to vary in a periodic pattern. They prove that a periodic modified base- stock policy is optimal, and show that the problem may be solved using a value-iteration method. Kapuscinski and Tayur (1996) also consider a periodic model; they compute inventory levels using infinitesimal perturbation analysis, a simulation-based method of estimating derivatives discussed by Glasserman and Tayur (1995).

A detailed survey of the state-of-the-art research on assemble-to-order (ATO) systems can be found in Song and Zipkin (2003). Also refer to the volume on supply chain structures edited by Song and Yao (2001). Several authors focus on performance evaluation and optimization techniques for the base-stock policy, assuming that demand in each period has a multivariate normal distribution. Hausman et al. (1998) examine the problem of maximizing the probability filling all demand in a period within a time window subject to a linear budget constraint, and develop heuristic methods. The model of Agrawal and Cohen (2001) minimizes the total expected component inventory costs, subject to constraints on the order fill rates using an allocation policy characterized by partial FCFS and fair-share allocation. Kaminsky and Swaminathan (2004) develop heuristics that utilize forecast evolution information to make production decisions in a capacitated manufacturing environment, taking into account production costs, holding costs, salvage costs and stock-out costs. Cheng et al. (2002) study the problem of minimizing average component inventory holding cost subject to product-family dependent fill rate constraints in a configure-to-order (CTO) system and provide an exact algorithm and a greedy heuristic for computing the optimal inventory policy. Song and Yao (2002) investigate the optimal inventory-service tradeoff in a single-product ATO system, focusing on fill-rate performance. The results are extended to multi-product models in Lu, Song and Yao (2003, 2005), where issues such as backorder minimization, leadtime variability and advance demand information are also studied.

Dietrich et al. (2005) discuss the use of implosion technology throughout IBM's supply chain for a variety of applications that require a rapid assessment of capability to respond to changes in demand, supply, or capacity. The paper describes the application of implosion to available-to-promise generation within a complex configured value chain characterized by configure-to-order product structures. Beyer and Ward (2002) describe an inventory management study conducted at HP's Network Server Division. The authors compute

inventory policies that reduce inventory and transportation costs in a two-tier supply network with a single manufacturing site and multiple fulfillment centers. Feitzinger and Lee (1997) show that postponement provides an effective means for firms to implement mass customization without incurring large operating costs associated with managing proliferating product variety.

1.2 Motivation and Contributions

Our work is motivated by a hybrid push-pull manufacturing strategy implemented at IBM's Systems Group (SG). As one of the largest providers of server computers, IBM is facing a significant challenge in managing one of the most vertically integrated supply chains in the industry. The supply chain is under constant pressure to move towards the assembly of components from a vast array of outside suppliers, and to sell components to the marketplace. Since IBM manufactures most major assemblies of its server computers in house, the integrated supply chain is even more complex and difficult to manage. Clearly, such an environment makes managing the extended supply chain critical to the companys success. To deal with these complexities, SG introduced the so-called fab/fulfillment model to their high-end server computer supply chain. The fab/fulfillment process combines mass production with small-batch production to achieve mass customization. In this model, the fulfillment center pulls fab tested parts consumptively, and builds complete systems to order. The goal of the fulfillment center is rapid, customized response to customer orders.

The fab/fulfillment concept was an original design point of IBM's integrated supply chain reengineering efforts that aimed at reaching world-class performance on responsiveness (customer serviceability) combined with complex configurations and options offered in a server manufacturing environment. The first stage (fabrication) is a push process where parts are replenished, tested, and assembled into subassemblies according to product-level build plans. The second stage (fulfillment) is a pull-based assemble-to-order process where the final assembly process is initiated when a customer order is received. The assemble-to-order paradigm offers higher product variety and hence often results in broader market coverage and increased demand volume. Since the most time consuming manufacturing tasks are completed in the first stage of production, cycle times in the second stage are relatively short which allows the fulfillment center to respond rapidly to customer orders. Postponing the final assembly provides additional flexibility in terms of product variety and achieves resource pooling in terms of maximizing the usage of manufacturing capacities and parts inventories.

The reconciliation of sales, manufacturing, and financial plans starts with the Sales and Operations Planning (S&OP) process and continues through the Top-Input decision to the MRP Master Scheduling process. The S&OP process is designed to help the company maximize customer satisfaction and meet profitability goals by achieving the optimal balance of demand and supply. Decisions on what to make

and when and how to deliver finished products are made based on scenarios that best meet business unit performance goals, factoring in service levels, sales forecasts, revenue plans, and pre-tax income analysis which balances inventory against the risk of stock outs. The S&OP process establishes an integrated set of plans that sales, marketing, operations, engineering and financial stakeholders are committed to support. The focus is at the product family level (such a zSeries z900 servers), and most emphasis is placed on quarterly total volumes.

The translation into volumes for specific system configurations in a product family (called Machine-Type-Model part numbers, or MTMs) is made by manufacturing. This process called Top-Input decision process is designed to make a final build decision such that the totals match the S&OP sales plan, factoring in the expected mix at the MTM level and feature ratio forecasts that drive the demand for lower level parts. Feature ratio forecasts are updated monthly, usually using the average of the last several months of actuals and manually overridden for new products or to incorporate market intelligence. The Top-Input decision controls how these sources influence the build plan and determines the customer supply commitments. The committed levels of supply are exploded through the bills-of-materials in the MRP Master Scheduling process which translates them into supply requests for components, sourced both from IBM manufacturing plants and externally procured.

The fab/fulfillment concept as practiced by IBM SG includes building tested parts and sub-assemblies to a build plan. Because of limited production capacity, the fab typically overproduces versus demand early in the quarter to meet peak demands towards the end of the quarter. The build plan is specified by a quarterly total and a spread, which determines how much to build in each month of the quarter. Typical spreads are 30% in month one, 40% in month two, and 30% in month three (denoted as 30/40/30). The goals of an evenly spread build plan are capacity utilization, plant and supplier requirements stability, and ramp up for end-of-quarter peak demands, ensuring adequate inventory and even some fab capacity used for fulfillment at quarter-end. An important implication of this business model is that inventory builds up during the quarter, and is at its lowest point at the very end of a quarter. A related implication is that safety stock is almost entirely determined by two factors: the decision of how much to build for the quarter overall, and the spreading (e.g., 30/40/30) of the build plans. One consequence of this spreading is that the server supply chain is much less sensitive (and less amenable) to traditional improvement actions. While such practices simplify planning and decision making by dealing with capacity limitations at an aggregated level, they fail to address the issue of managing inventory-driven costs including financing, inventory write-downs and inventory write-offs (obsolescence) which are tremendous cost drivers.

The contributions of the paper are as follows. First, we formulate an optimization problem that is applicable to a large class of real-world supply chains that deal with a hybrid push-pull manufacturing strategy,

complex configured products, limited manufacturing and labor capacity, and customer service level agreements. Second, we present two variants of the problem that both have distinct advantages over the common practice of applying a quarter-to-month spread ratio to derive the build plan (production smoothing). In the product-based model, we define the weekly build quantities of fully configured systems as decision variables and demonstrate inventory cost reductions over the production smoothing strategy. In the component-based model, we derive optimal component- level build quantities to achieve further cost reductions by relaxing the squared set requirement of the product-based model and exploiting parts commonality. Third, we exploit the structure of the problem formulation to develop efficient numerical algorithms, in particular a backward-recursion algorithm that solves the problem optimally through decomposition. And fourth, we demonstrate the efficacy of the approach by numerical experiments with realistic production data. This produces several insights into the build plan optimization problem, specifically regarding the effects of component-based manufacturing.

2. A Product-Based Model

We formulate an nonlinear optimization problem to model the manufacturing planning problem under study. Instead of using a quarter-to-month spread ratio to derive the build plan as with the heuristic approach, we define the weekly build quantities of MTMs as the decision variables to capture potentially greater benefits from the use of analytic approach.

2.1 Formulation

We use the superscript K to index the MTM types, and the subscript i to index components (items or subassemblies). Let $\mathcal{K}$ denote the set of all MTM types, and $\mathcal{I} = \{1, ..., m\}$ denote the set of all components. We further partition $\mathcal{I}$ into disjoint subsets, $\mathcal{I} = \cup_{j=1}^{n} A_j$, with each A_j representing a collection of similar component, e.g. CPU, memory, and so forth.

Each MTM type is characterized by a subset of components, i.e., $K \subseteq \mathcal{I}$, along with the usage counts: u_i^K, $i \in K$, i.e., the number of units required from each component i.

Let $t = 1, ..., T$ index the time periods. For instance, when each period is a week, $T = 12$ represents a quarter.

Assume the following are given data:

- $C_{j,t}^M$, $j = 1, ..., n$, the machine capacity limit, in terms of the maximum number of components $i \in A_j$ that can be processed in period t. Note that this capacity is shared among all components in A_j, and only those components.

- h_t^K, $K \in \mathcal{K}$, inventory cost for each unit of type K MTM left over at the end of period t (after supplying demand). For instance, $h_t^K = \sum_{i \in K} c_i u_i^K$, with c_i being the cost (for raw materials, processing and labor) associated with each unit of component i. (Note that since the final assembly is only built to order, the "MTM left over at the end of period t" really refers to the ensemble of its components.)

- π_t, penalty cost for over-time labor, when the capacity limit C_t^L is exceeded.

In addition, assume for each MTM type K, we know its demand for each period, D_t^K. Suppose

$$D_t^K = \mu_t^K + \sigma_t^K Z_t,$$

with Z_t denoting the standard normal variate; assume Z_t's are i.i.d. over time. Denote

$$D^K(1,t) := \sum_{s=1}^{t} D_s^K;$$

and similarly denote

$$\mu^K(1,t) := \sum_{s=1}^{t} \mu_s^K,$$

and

$$\sigma^K(1,t) := [\sum_{s=1}^{t} (\sigma_s^K)^2]^{1/2}.$$

The decision variables are: x_t^K, the build quantity for each MTM type K in each period t. Note, again, that the MTMs will *not* be actually "built" as they are assembled to orders only. The MTM build quantities are surrogates for component build quantities. Once x_t^K's are decided, so will the build quantity for each component: for component i, this is equal to $\sum_{K \ni i} u_i^K x_t^K$. Denote:

$$x^K(1,t) := \sum_{s=1}^{t} x_s^K.$$

The objective is to minimize the total inventory cost and penalty cost for over-time labor:

$$\min \sum_{t=1}^{T} \{ \sum_{K \in \mathcal{K}} h_t^K \mathsf{E}[x^K(1,t) - D^K(1,t)]^+ . + \pi_t [\sum_{K \in \mathcal{K}} x_t^K - C_t^L]^+ \}. \tag{1}$$

The constraints are:

$$\sum_{i \in A_j} \sum_{K \ni i} u_i^K x_t^K \le C_j^M, \qquad j = 1, ..., n, \ t = 1, ..., T; \tag{2}$$

$$x^K(1,t) \ge \mu^K(1,t) + \sigma^K(1,t) \cdot \epsilon_{\alpha^K}, \qquad K \in \mathcal{K}, \ t = 1, ..., T. \tag{3}$$

The last constraint above is equivalent to

$$\mathsf{P}[x^K(1,t) \geq D^K(1,t)] \geq \alpha^K,$$

which enforces the service levels, represented by α^K, for product K, with ϵ_{α^K} denoting the point corresponding to the α^K percentile of the standard normal distribution; e.g., $\epsilon_{\alpha^K} = 1.64$ when $\alpha^K = 95\%$.

Note in the objective function in (1), the inventory part assumes full backlog of demand. The expectation term in (1) can be derived as follows:

$$\begin{aligned} & \mathsf{E}[x^K(1,t) - D^K(1,t)]^+ \\ = \; & \mathsf{E}[x^K(1,t) - \mu^K(1,t) - \sigma^K(1,t) \cdot Z]^+ \\ = \; & \sigma^K(1,t)\mathsf{E}\Big[\frac{x^K(1,t) - \mu^K(1,t)}{\sigma^K(1,t)} - Z\Big]^+ \\ = \; & \sigma^K(1,t)H\Big(\frac{x^K(1,t) - \mu^K(1,t)}{\sigma^K(1,t)}\Big), \end{aligned} \tag{4}$$

where

$$H(x) := \mathsf{E}[x - Z]^+ = \int_{-\infty}^{x} (x - z)\phi(z)dz = x\Phi(x) + \phi(x), \tag{5}$$

with Φ and ϕ being the distribution and the density functions of Z.

Since $H(x)$ is both increasing and convex in x, the optimization problem has an increasing and convex objective function, and a set of linear constraints.

Proposition 1 *(Feasibility condition) An optimal solution exists to the problem defined by (1)-(3) if the following conditions are satisfied.*

$$\sum_{i \in A_j} \sum_{K \ni i} u_i^K(\mu^K(1,t) + \sigma^K(1,t) \cdot \epsilon_{\alpha^K}) \leq C_j^M * t, \qquad j = 1, ..., n, t = 1, ..., T. \tag{6}$$

Discussions

1. What's referred to as MTMs here is really at the level of final products. That is, each MTM has a *fixed* configuration (bill of materials) in terms of components and their usage counts.

2. Decisions (build quantities) are made at the level of MTM (as defined above), by type and by period. Since each MTM has a fixed bill of materials, the decisions are readily translated into build quantities of components and sub-assemblies.

3. Inventory cost is charged to the *end* inventory of each period.

4. The definition of $C_{j,t}^M$ relates readily to more primitive data as follows. For instance in the case of constant capacity, suppose there are m machines for testing the subset of components in A_j, and suppose each machine handles a batch of B units at a time, and the testing time is w weeks on average. Then, $C_j^M = mB/w$ per week.

3. A Component-based Model

The product-based model presented above implicitly requires the components as defined by the bill of materials for an MTM be built in a "square" set, meaning that all the components required to assembly a particular MTM needs to be built or replenished at the same time. This is simply because the variables defined in the product-based model present the MTM-level build quantities. To convert the MTM-level build plan to a component-level build plan, one would simply apply the usage rates as defined in the bill of materials to get the build quantities of the components. With decision variables defined at the component-level, the "square" set restriction can be removed to allow additional flexibility when optimizing the build plan with the capacity constraints.

3.1 Component-based Model Formulation

To model the planning of build quantities at the component level, we introduce a set of variables to represent the build quantity of each component. Let w_t^i be the build quantity for component i in period t, $i \in \mathcal{I}$ and $1 \le t \le T$. We also define the accumulative build quantity for component i from period 1 to period t as

$$w^i(1,t) = \sum_{s=1}^{t} w_s^i.$$

The objective is to minimize the total component inventory cost and the overtime labor cost.

$$\min \quad \sum_{t=1}^{T} \{ \sum_{i \in \mathcal{I}} h_t^i \mathsf{E}[w^i(1,t) - \sum_{K \in \mathcal{K}} \sum_{K \ni i} u_i^K D^K(1,t)]^+ + \pi_t [\sum_{K \in \mathcal{K}} x_t^K - C_t^L]^+ \}. \tag{7}$$

The constraints are:

$$\sum_{i \in A_j} w_t^i \le C_j^M, \qquad j = 1, ..., n, \ t = 1, ..., T; \tag{8}$$

$$x^K(1,t) \ge \mu^K(1,t) + \sigma^K(1,t) \cdot z_{\alpha^K}, \qquad K \in \mathcal{K}, \ t = 1, ..., T; \tag{9}$$

$$w^i(1,t) \ge \sum_{K \in \mathcal{K}} \sum_{K \ni i} u_i^K x^K(1,t), \qquad i \in \mathcal{I}, \ t = 1, ..., T. \tag{10}$$

The last constraint above ensures that the MTM-level build plan is always feasible given the tested parts available at any given time. In fact we can combine (9) and (10) to get the service level constraints defined

at the component level as the following

$$w^i(1,t) \geq \sum_{K \in \mathcal{K}} \sum_{K \ni i} u_i^K \left(\mu^K(1,t) + \sigma^K(1,t) \cdot z_{\alpha^K} \right), \ t = 1, ..., T,$$

and eliminate the $x^K(1,t)$ variables from the formulation. Also note that h_t^i is now the inventory holding cost for component i in period t. Furthermore, this formulation implies that the labor requirement is determined by the MTM-level production.

Proposition 2 *The minimimal cost of the component model is no greater than the minimimal cost of the corresponding MTM model.*

Remarks

- The benefits of the component-based model over the product-based model are achieved by relaxing the requirement for building "square sets" at the component level. The difference between the product-based model and the component-based model reflects the cost reduction due to this factor. Since the capacity constraints are defined at the component level, it is clear that better capacity utilization can be achieved through optimizing the component level build plan. Especially when the capacity is tight for some parts in certain periods, one has the flexibility to adjust the build schedules for the parts to avoid capacity constraints in those periods. On the other hand, the product-based model would be more restrictive in this case since parts are to be built in "square sets", therefore, less flexible in making adjustments of the build plan.

- The another potential factor that could lead to additional cost savings is the risk-pooling effect on the demand variability when we switch from the product-based model to a component-based model. This is based on the fact that parts commonality exists among different MTMs. Since the demand variability is reduced through demand aggregation at the component-level, the safety stock required to maintain the same level of service should be less than what would be required if the safety stock was kept at the MTM level.

- However, the cost savings due to the demand variability reduction is not reflected in the current component-based model because the demand variability reduction is not captured in the model. To take the advantage of the risk-pooling effect, we would need to model the demand variability reduction that could be achieved by forecasting the component-level demand directly. Furthermore, a component-based (or build block-based) planning methodology should be adopted such that the common components are indeed shared by the MTMs that require them rather than individually allocated to each of those MTMs.

4. Two Solution Approaches

Both the product-based model and the component-based model have nonlinear objective functions, more specially convex functions, with linear constraints. Today's high-power nonlinear solvers are probably capable of solving this type of problems. However, there are certain structures of the problems that can be explored and utilized to facilitate the computation involved in the optimization. We have developed two computation approaches that demonstrated improved computation performance through numerical results. One approach is to approximate the nonlinear objective function through linearization so that a near-optimal solution can be obtained by using a linear program solver. The second approach is to use a backward-recursion algorithm to solve the problem through decomposition.

4.1 Piece-wise Linearization

First notice that the penalty part of the objective function in (1) can be replaced by $\pi_t u_t^K$, with

$$\sum_{K\in\mathcal{K}} x_t^K - C_t^L = v_t - w_t, \tag{11}$$

and $v_t, w_t \geq 0$ being additional slack variables. Furthermore, since

$$x_t^K = x^K(1,t) - x^K(1,t-1),$$

we can let

$$y_t^K := x^K(1,t), \qquad t = 1,...,T;\ K \in \mathcal{K}$$

be the new variables (and denote $y_0^K := 0$ for all K). This way, the optimization problem becomes:

$$\min \sum_{t=1}^{T} \left\{ \pi_t v_t + \sum_{K\in\mathcal{K}} h_t^K \sigma^K(1,t) H\Big(\frac{y_t^K - \mu^K(1,t)}{\sigma^K(1,t)}\Big) \right\}, \tag{12}$$

subject to the following constraints:

$$\sum_{i\in A_j} \sum_{K\ni i} u_i^K [y_t^K - y_{t-1}^K] \leq C_j^M, \qquad j = 1,...,n,\ t = 1,...,T; \tag{13}$$

$$\sum_{K\in\mathcal{K}} [y_t^K - y_{t-1}^K] - C_t^L = v_t - w_t, \qquad t = 1,...,T; \tag{14}$$

$$y_t^K \geq \mu^K(1,t) + \sigma^K(1,t)\cdot \epsilon_{\alpha^K}, \qquad K \in \mathcal{K},\ t = 1,...,T; \tag{15}$$

$$y_T^K \geq y_{T-1}^K \geq \cdots \geq y_1^K \geq 0, \qquad K \in \mathcal{K} \tag{16}$$

$$v_t \geq 0, \quad w_t \geq 0, \qquad K \in \mathcal{K},\ t = 1,...,T. \tag{17}$$

Clearly, the above is a separable convex programming problem.

From (5), it is readily verified that

$$H(x) \approx x, \quad \text{for} \quad x \geq 3,$$

and

$$H(x) \approx 0, \quad \text{for} \quad x \leq -3.$$

Hence, the only non-linearity of the function is within the interval $[-3, 3]$. (And, over this interval it is convex).

Hence, we should be able to approximate the H function by piecewise linear functions with a relatively small number of pieces. Suppose we preselect the following points on the x-axis:

$$-3 := z_0 < z_1 < \cdots < z_p < z_{p+1} := 3. \tag{18}$$

For any x, suppose $x \in [z_\ell, z_{\ell+1}]$, for some $\ell \leq p$. We can write

$$x = z_\ell + \lambda(z_{\ell+1} - z_\ell),$$

for some $\lambda \in [0, 1]$. We can then approximate $H(x)$ as follows:

$$H(x) \approx H(z_\ell) + \lambda[H(z_{\ell+1}) - H(z_\ell)],$$

which is nothing more than linear interpolation. In general, we can write any $x \in (-3, +3)$ as

$$x = \sum_{\ell=0}^{p} \lambda_{\ell+1}(z_{\ell+1} - z_\ell),$$

and

$$H(x) \approx \sum_{\ell=0}^{p} \lambda_{\ell+1}[H(z_{\ell+1}) - H(z_\ell)].$$

These, along with $H(x) = x$ for $x \geq 3$ and $H(x) = 0$ for $x \leq -3$, approximate the H function by $p + 2$ linear pieces.

We now apply this linearization procedure to the H function in (12). Thanks to the constraint in (15), we can start at

$$z_0 = \min_{K \in \mathcal{K}} \{\epsilon_{\alpha^K}\}, \tag{19}$$

instead of $z_0 = -3$ as in (18). (The other preselected z_ℓ's are the same as in (18).)

Write

$$y_t^K = \mu^K(1, t) + \sigma^K(1, t) \sum_{\ell=0}^{p} \lambda_{\ell+1}^{K,t}(z_{\ell+1} - z_\ell). \tag{20}$$

Then, we have

$$H\Big(\frac{y_t^K - \mu^K(1,t)}{\sigma^K(1,t)}\Big) = H\Big(\sum_{\ell=0}^{p} \lambda_{\ell+1}^{K,t}(z_{\ell+1} - z_\ell)\Big) = \sum_{\ell=0}^{p} \lambda_{\ell+1}^{K,t}[H(z_{\ell+1}) - H(z_\ell)].$$

Consequently, the original optimization problem becomes:

$$\min_{\lambda} \sum_{t=1}^{T} \quad \left\{ \pi_t v_t + \sum_{K\in\mathcal{K}} h_t^K \sum_{\ell=0}^{p} \lambda_{\ell+1}^{K,t}[H(z_{\ell+1}) - H(z_\ell)] \right\};$$

with the following constraints:

$$\sum_{i\in A_j} \sum_{K\ni i} u_i^K \sum_{\ell=0}^{p} (\lambda_{\ell+1}^{K,t} - \lambda_{\ell+1}^{K,t-1})(z_{\ell+1} - z_\ell) \le C_j^M, \qquad j = 1, ..., n,\ t = 1, ..., T; \tag{21}$$

$$\sum_{\ell=0}^{p} \lambda_{\ell+1}^{K,t}(z_{\ell+1} - z_\ell) \ge \epsilon_{\alpha^K}, \qquad K \in \mathcal{K},\ t = 1, ..., T; \tag{22}$$

$$\sum_{K\in\mathcal{K}} \sum_{\ell=0}^{p} (\lambda_{\ell+1}^{K,t} - \lambda_{\ell+1}^{K,t-1})(z_{\ell+1} - z_\ell) - C_t^L = v_t - w_t, \qquad t = 1, ..., T; \tag{23}$$

$$\sum_{\ell=0}^{p} \lambda_{\ell+1}^{K,t}(z_{\ell+1} - z_\ell) \ge \sum_{\ell=0}^{p} \lambda_{\ell+1}^{K,t-1}(z_{\ell+1} - z_\ell), \qquad K \in \mathcal{K},\ t = 1, ..., T; \tag{24}$$

$$0 \le \lambda_\ell^{K,t} \le 1, \qquad \forall K, t, \ell. \tag{25}$$

The above is a linear programming with $\lambda_i^{K,t}$ as decision variables. Once the optimal solution to $\lambda_i^{K,t}$ is obtained, the optimal solution to y_t^K follows from (20).

The modeling results in a convex programming problem, i.e. minimizing a convex function over a polyhedron defined by linear constraints, and standard convex programming solvers can be employed to produce a solution. The objective function is regular both analytically and geometrically in a local sense, that is, it is locally smooth and flat (close to a linear function). Although the technique of using linear piecewise function approximation to solve convex programming is usually not practical due to its computational complexity, the form of objective function in this problem allows us to use a linear approximation with a small number of segments. The unique form of the objective function in our formulation is shared by a large class of problems in inventory and supply chain management. Therefore, the proposed methods can be adapted to solve many related problems.

4.2 Decomposition

A special case of the component-based model is when the overtime labor cost is not considered for build plan decisions. Then we can rewrite the component-based model (7-10) without the penalty part of the objective function as follows.

$$\min \sum_{t=1}^{T} \{ \sum_{i\in\mathcal{I}} h_t^i \mathsf{E}[w^i(1,t) - d^i(1,t)]^+ \}, \tag{26}$$

Subject to:

$$\sum_{i \in A_j} w_t^i \leq C_j^M, \qquad \forall j, t; \tag{27}$$

$$w^i(1,t) \geq B^i(1,t), \qquad \forall i, t. \tag{28}$$

where

$$d^i(1,t) = \sum_{K \in \mathcal{K}} \sum_{K \ni i} u_i^K D^K(1,t),\ t = 1, ..., T,$$

and

$$B^i(1,t) = \sum_{K \in \mathcal{K}} \sum_{K \ni i} u_i^K \left(\mu^K(1,t) + \sigma^K(1,t) \cdot z_{\alpha^K} \right),\ t = 1, ..., T.$$

Notice that the new problem is separable in j. Therefore, we can decompose the problem by j and solve a number of smaller problems each with simple constraints. We can rewrite the constraint (27) as

$$\sum_{i \in A_j} w^i(1,t) \geq \bar{C}_j(1,t),$$

where $\bar{C}_j(1,t)$ is given by

$$\bar{C}_j(1,t) = \sum_{i \in A_j} w^i(1,t+1) - C_j^M.$$

In fact this new formulation is also separable in t. Therefore, we can further simplify the sub-problem for a given j, as defined in (26- 28), by decomposing it for each $t = T, ..., 1$, i.e.,

$$\min \sum_{i \in A_j} h_t^i \mathsf{E}[w^i(1,t) - d^i(1,t)]^+, \tag{29}$$

Subject to:

$$\sum_{i \in A_j} w^i(1,t) \geq \bar{C}_j(1,t), \tag{30}$$

$$\bar{B}^i(1,t) \leq w^i(1,t) \leq w^i(1,t+1), \qquad i \in A_j. \tag{31}$$

where

$$\bar{B}^i(1,t) = B^i(1,t) + w^i(1,t+1) - \bar{B}^i(1,t+1)$$

with $\bar{B}^i(1,T+1) := 0$. Note that $w^i(1,t+1)$ is a constant given by the solution for period $t+1$ and $w^i(1,T+1) := 0$. The structure revealed in this simplified formulation allows us to decompose the original problem by j and by t into a sequence of smaller problems, thus reducing the complexity of the problem and hence the total computation time required.

The objective function of the sub-problem given j and t is given by

$$g(j,t|\mathbf{w}(j,t)) = \sum_{i \in A_j} h_t^i \mathsf{E}[w^i(1,t) - d^i(1,t)]^+,$$

where $\mathbf{w}(j,t)$ denotes the vector $w^i(1,t) : i \in A_j$. The value function is then defined as

$$v(j,t) = \min \sum_{k=t}^{T} g(j,k|\mathbf{w}(j,k)).$$

The new formulation of the problem with the decomposition can be presented as follows.

$$v(j,t) = \min_{\mathbf{w}(j,t)\in\mathcal{U}_t} \left(v(j,t+1|\mathbf{w}(j,t) + g(j,t+1|\mathbf{w}(j,t+1))\right),$$

where $t = 1, ..., T, j = 1, ..., n$ and $v(j,T) = 0, \forall j$. $\mathcal{U}_t$ is defined by (30) and (31).

Given the structure of the sub-problem for given t and j, it can be observed that the optimal solution $\mathbf{w}(j,t)$ has the following properties

Proposition 3 *The problem defined by* (29-31) *is feasible if*

$$\sum_{i\in A_j} w^i(1,t+1) \geq \bar{C}_j(1,t). \tag{32}$$

The optimal solution $\mathbf{w}(j,t)$ *is always at the boundaries defined by (30) and (31).*

Proof. If (32) does not hold, then clearly (30) cannot be true. Furthermore, since the objective function of the sub-problem g is convex and non-decreasing in $\mathbf{w}(j,t)$, the optimal value must be attained at the boundaries.

There are two cases that an optimal solution can be attained.

- If constraint (30) is not binding, then the optimal solution is directly given by the lower bounds defined by (31);
- If constraint (30) is binding, the optimal solution lies on the line defined by (30) and between the two intersects of the line with the bounds defined by (31).

The resulting problem with $i \in A_j$ for a given j can be solved using a backward recursion algorithm outlined below. Note that each stage of the recursion involves solving a nonlinear optimization problem with a convex objective and linear constraints. The solution to this problem (29-31) can be easily obtained by performing a greedy search.

- Step 0: Initialize $w_i(1,t) = \lceil B^i(1,t) \rceil,\ t = 1, ..., T$ ($\lceil x \rceil$ = the least integer greater than or equal to x), and let the step size be Δw.
- Step 1: If for $t = T$, $\sum_{i\in A_j} w^i(1,t) > C_j^M(1,t)$, then the problem is infeasible. Exit. Otherwise continue to Step 2.

- Step 2: For $t = T-1, ..., 2$,
  If for some $t < T$, $\sum_{i \in A_j} w^i(1,t) > C_j^M(1,t)$, find $i^* = \min\{h_t^i \Phi((w^i(1,t) - \mu^i(1,t))/\sigma^i(1,t))\}$. Let $w^{i^*}(1,t) := w^{i^*}(1,t) - \Delta w$ and $w^{i^*}(1,t-1) := w^{i^*}(1,t-1) + \Delta w$. Repeat this step until $\sum_{i \in A_j} w^i(1,t) \leq C_j^M(1,t)$.

One added benefit of the backward recursion algorithm is that we can easily restrict the solutions to be integers by letting $\Delta w = 1$. Adding the integer constraints in the original formulation would increase the computational complexity substantially and most likely would render the problem intractable when a standard NLP solver is used.

5. Numerical Results

This section contains a representative sample of results obtained from implementing the optimization models described in the previous sections. The data set resembles a typical real-world problem compiled from actual IBM server data. It consists of 12 machine types, i.e., Model 1 – Model 12. Each machine type configuration is assembled from components that fall into the four commodity groups, i.e., MCM (multi-chip modules), power supplies, memory and logic. Each configuration shipped is a potentially unique configuration of top-level components selected from the configuration menu of the particular machine type being ordered. The assembly process is modeled as a fulfillment site where the selected components are merged with the appropriate chassis for shipment to end customers. The precise makeup of a final configuration is not known until an order is placed because of the ability of customers to personalize their orders. As a result, the bill-of-materials of a machine type entails fractional usage rates (also called feature ratios) that indicate the expected usage of a specific component in a particular machine type. The bills-of-materials and feature ratios as well as the unit component costs are depicted in Figure 1.

The customer demand for end products is modeled as a twelve-week outlook consisting of a sales volume projection for each week and machine type as shown in Figure 2. In the numerical study we assume that the forecast for each period is normally distributed with a standard deviation of 20% of the sales volume projection. Notice that the forecast is heavily skewed towards the later periods in the twelve-week outlook. In practice, we observed that often up to 60% of customer orders are shipped in the last 3 weeks of a quarter.

Because of limited production capacity, component subassemblies and tests are scheduled ahead of time early in the quarter to meet peak demands towards the end of the quarter. To optimize production resources, the fab capacity is spread evenly across the quarter and the parts intake into the line is driven by the fabrication schedule. Typical spreads are 30% in month one, 40% in month two, and 30% in month three, denoted as 30/40/30. To demonstrate the advantages of the product-based and component-based models over

	Unit Cost	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	Model 8	Model 9	Model 10	Model 11	Model 12
MCM 1	$1,280	-	-	-	-	-	0.5	-	-	-	-	-	-
MCM 2	$10,780	-	-	-	-	2.8	-	-	-	-	-	-	2.8
MCM 3	$5,850	2.0	3.2	8.8	-	-	-	-	-	-	8.0	4.0	-
MCM 4	$131	-	-	-	14.5	18.6	12.5	24.0	48.0	25.7	-	-	6.0
MCM 5	$10,780	8.0	2.0	8.8	-	-	-	-	-	-	8.0	2.0	-
Power 1	$21	7.0	8.0	4.0	3.2	-	2.0	-	1.0	1.0	7.0	8.0	3.0
Power 2	$94	2.5	3.3	3.0	-	2.0	-	3.0	-	2.0	2.5	3.3	-
Power 3	$44	2.0	2.0	5.0	2.0	-	-	-	1.0	-	2.0	2.0	1.0
Memory 1	$244	1.0	2.8	1.5	-	-	-	-	-	-	1.0	2.8	-
Memory 2	$233	0.1	0.4	-	-	-	-	-	-	-	0.1	0.4	-
Memory 3	$619	-	-	-	0.2	0.8	1.3	1.0	0.1	1.0	-	-	0.8
Memory 4	$316	-	-	-	0.4	0.9	1.8	-	0.5	1.4	-	-	1.0
Logic 1	$522	-	-	1.0	0.8	0.8	0.8	0.8	0.8	0.8	-	-	0.8
Logic 2	$131	1.0	1.0	-	1.4	2.5	2.7	2.3	1.2	2.1	1.0	1.0	1.0
Logic 3	$88	-	-	-	3.0	3.0	5.0	5.1	2.4	1.8	-	-	1.5
Logic 4	$44	2.5	2.5	2.2	3.5	5.6	6.7	5.4	2.6	5.0	2.5	2.5	-

Figure 1: Unit component costs and bills-of-materials of sample data set.

	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	Model 8	Model 9	Model 10	Model 11	Model 12
Period 1	6	2	3	6	3	5	6	6	2	5	3	2
Period 2	9	3	5	7	5	8	8	5	4	4	6	4
Period 3	10	2	6	10	5	6	7	6	3	6	7	4
Period 4	16	5	13	14	12	13	15	16	8	10	16	6
Period 5	19	4	11	9	9	11	17	10	8	6	19	6
Period 6	9	4	16	16	10	12	20	14	8	11	19	7
Period 7	18	5	12	17	14	14	18	10	7	15	12	6
Period 8	23	9	23	23	15	26	34	27	12	23	33	11
Period 9	23	6	15	27	13	23	34	20	13	21	31	12
Period 10	25	11	27	38	23	31	40	33	14	28	28	13
Period 11	56	14	42	42	37	39	54	48	26	32	46	19
Period 12	54	20	38	54	30	46	64	60	31	48	68	22
Total	268	85	211	263	176	234	317	255	136	209	288	112

Figure 2: Demand forecast for the 12 machine types.

such a production smoothing strategy, we present Figure 3. The figure compares the cumulative parts intake (in millions of dollars) of the different production strategies. The (un-capacitated) forecast constitutes a lower bound and is included for reference.

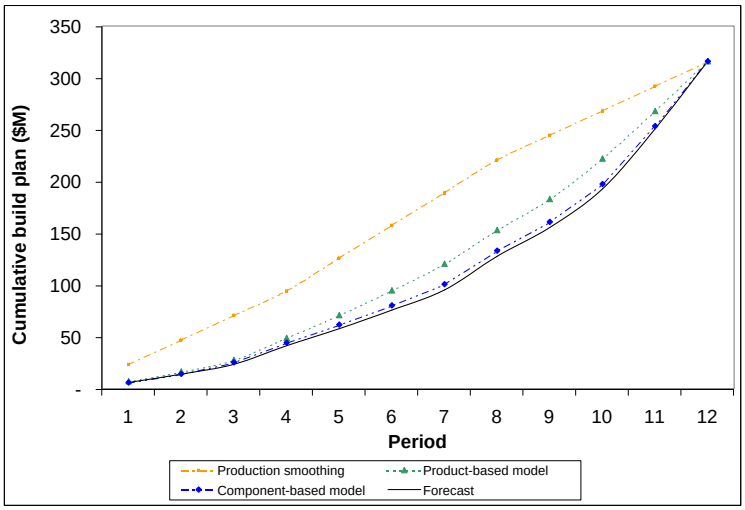

Figure 3: Cost comparisons among different manufacturing strategies.

To provide a fair comparison, we computed safety stock requirements to achieve a customer service level target of 90% in all scenarios. The production capacity specified separately for each of the four commodity groups was limited to 160% of the average weekly commodity run rate over the entire planning horizon. In the product-based model, we computed the cost of each machine type configuration as the sum of the costs of the components used in its assembly. Once we obtained the optimal build quantities for each machine type, we applied the feature ratios to compute the component build quantities which were then used to calculate the cumulative parts intake.

As a general observation, it is evident that the inventory build-up during the quarter is much higher under a production smoothing approach than it is under an optimal strategy. For example, the production smoothing approach indicates that about $190M of component inventory should be built by week 7, whereas the product-based model and the component-based model recommend building $121M and $101M respectively. Since the product-based model assumes that all components that make up a configured system are manufactured as a squared set, it is less flexible in its build plan recommendation and yields slightly higher inventory costs than the component-based model.

For the same scenario, Figures 4 and 5 illustrate the relative amount of inventory produced early as a

percentage of the cumulative forecast for a low-dollar and a high-dollar MCM part number.

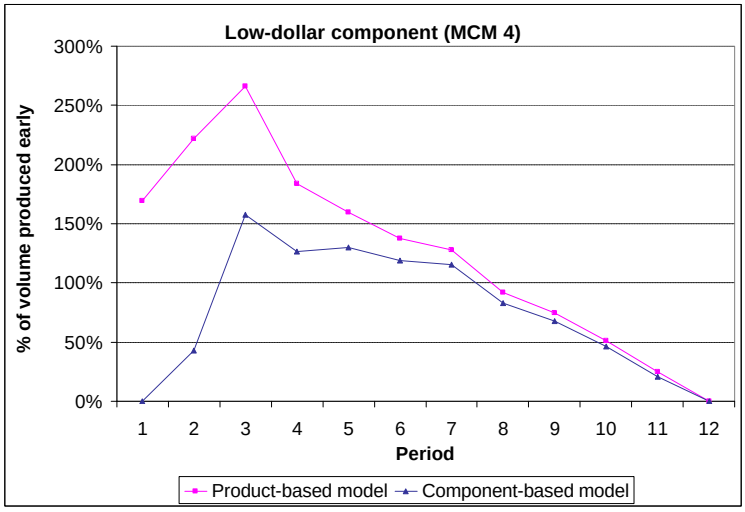

Figure 4: Relative amount of inventory build ahead of time for a low-dollar MCM component.

Notice that the component-based model results in lower build-ahead volumes than the product-based model in both instances. And more importantly, the component-based model avoids inventory build-ahead altogether for the high-dollar part as shown in Figure 5. In other words, the intake of the high-dollar part number into fabrication is scheduled just-in-time to satisfy customer orders. The reason is that the high-dollar part MCM5 drives about 58% of the total cost of components assembled into systems that are shipped to customers over the entire quarter. Since the squared set restriction is removed in the component-based model, it attempts to schedule low-dollar parts early in the quarter in order to reserve enough production capacity to schedule the high-dollar parts as close to the end customer demand as possible.

Figures 6 and 7 show a similar comparison for memory part numbers. Notice that in this case the component-based model produces a larger build-ahead of low-dollar components than the product-based model. But on the other hand, it also achieves significantly lower levels of inventory for the high-dollar part number as illustrated in Figure 7. Postponing the manufacturing of high-dollar parts is particularly desirable because it reduces the high risk and expense associated with building up component inventories in advance of receiving actual customer orders.

Next we investigate the impact of manufacturing capacity on the optimal build plan. As in the base case, we assume that the production capacity for each commodity group is 160% of the average weekly demand over the 12-week planning horizon. Figure 8 illustrates the capacity utilization of one of the commodity

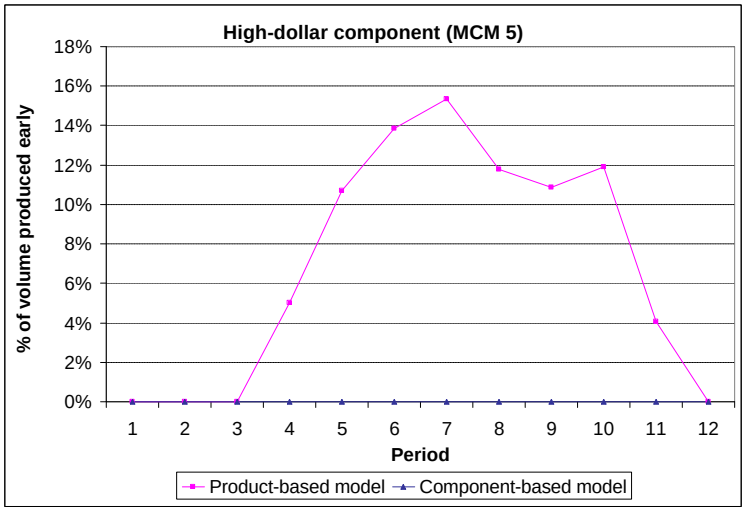

Figure 5: Relative amount of inventory build ahead of time for a high-dollar MCM component.

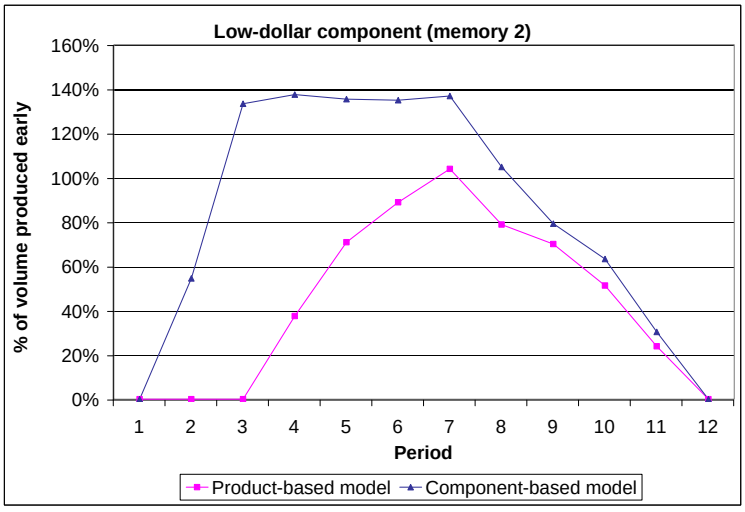

Figure 6: Relative amount of inventory build ahead of time for a low-dollar memory component.

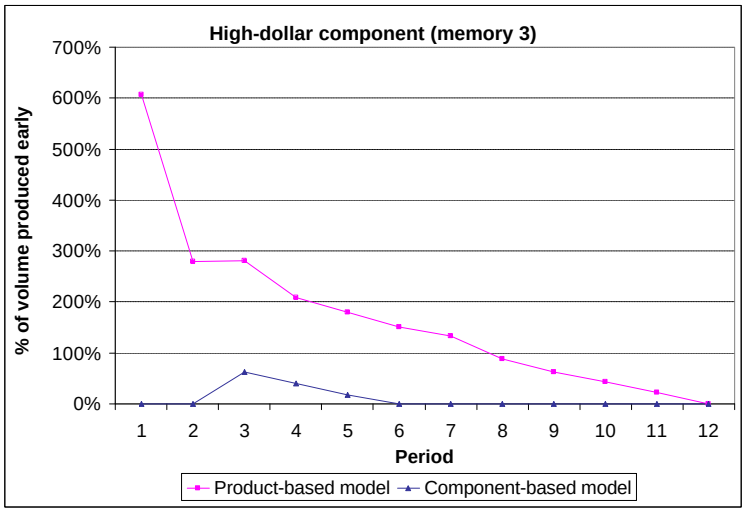

Figure 7: Relative amount of inventory build ahead of time for a high-dollar memory component.

groups (MCMs) for the build plans generated by the optimization models. The available capacity (7,800 units per week) can accommodate about 62% of the peak demand which is shown in the chart for reference.

As expected, inventory in the product-based model starts to ramp up earlier than in the component-based model, leaving some manufacturing capacity underutilized in weeks 9 and 10. In the component-based model, the inventory build-up is postponed until week 7 where the manufacturing capacity becomes fully utilized for the remaining six weeks of the planning horizon. This reiterates the advantages of utilizing the component- based model when optimizing build plans. Figure 9 shows a similar comparison when the MCM capacity is further reduced to 5,900 units per week.

The discussion thus far has focused on some qualitative characteristics of the two proposed manufacturing strategies. We now turn our attention to the supply chain's financial outcome as measured by the total inventory cost given in eqns. (1) and (7). Figures 10 and 11 summarize the total inventory cost (in millions) pertaining to the three manufacturing strategies for different customer service levels and capacity limitations. The rows labeled "production smoothing" represent the (30/40/30) quarter-to- month spread to determine the build quantities. The columns labeled "% cost reduction" report the relative inventory cost reduction of the product-based and component-based model over the production smoothing strategy. The tables report the financial outcomes for the case of unlimited manufacturing capacity as well as two additional scenarios where manufacturing capacity is moderately and tightly constrained.

The results attest to the significant savings that can be achieved when implementing the models discussed

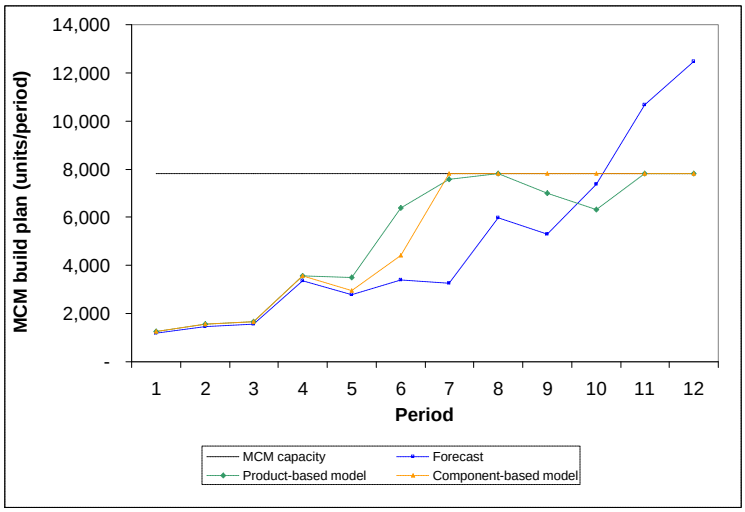

Figure 8: Comparison between product-based and component-based build plans for MCMs under moderate capacity constraints.

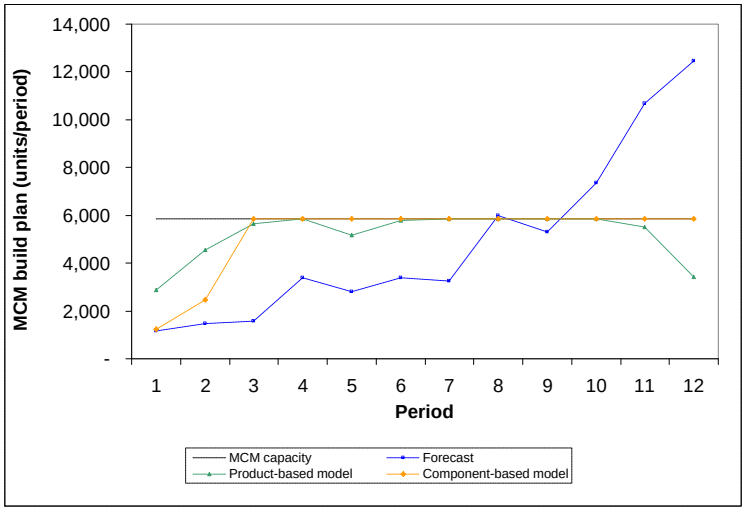

Figure 9: Comparison between product-based and component-based build plans for MCMs under tight capacity constraints.

90% service level						
	Unlimited capacity	% cost reduction	Light capacity constraint	% cost reduction	Tight capacity constraint	% cost reduction
Production smoothing	923	-	923	-	923	-
Product-based model	271	70.6%	306	66.8%	398	56.9%
Component-based model	271	70.6%	281	69.6%	301	67.4%

Figure 10: Total cost comparisons among different manufacturing strategies for 90% service level target.

95% service level						
	Unlimited capacity	% cost reduction	Light capacity constraint	% cost reduction	Tight capacity constraint	% cost reduction
Production smoothing	1,030	-	1,030	-	1,030	-
Product-based model	340	67.0%	391	62.0%	510	50.5%
Component-based model	340	67.0%	352	65.8%	377	63.4%

Figure 11: Total cost comparisons among different manufacturing strategies for 95% service level target.

in this paper. Overall inventory cost can be reduced by 50% or more if the solution of the product-based model is implemented. Furthermore, optimizing the build quantities at the component level can generate as much as 30% additional savings over the product-based model. When capacity becomes more constrained the demand in later periods exceeds the available capacity and components need to be built ahead of time when demand volumes are low. In the product-based model, the squared-set requirements impose additional restrictions that force the optimization to build components even earlier, whereas the component-based model can often delay the assembly of high-dollar components to later periods as the squared-set restriction is being removed.

6. Extensions

We describe briefly below a number of extensions that incorporate additional considerations in real-world scenarios.

6.1 A Modified Formulation

In reality, some of the test capacities are not specific to a certain type of components (called commodity type). For example, MCM parts and memory parts may go through similar tests requiring the same capacity. Here we define the capacity constraints in terms of the type of tests to be performed (called driver type). Denote the subset of $\mathcal{K}$ with components requiring the test capacity of driver type j by V_j. Note that V_j may not be disjoint since some components are tested with more than one driver type. We further define that C_j^V be the test capacity of driver type j, $j = 1, ..., n_v$ (n_v is the number of different driver types), and v_{ij} be the

usage rate by component i when driver type j is used. The new constraints are:

$$\sum_{i \in V_j} v_{ij} w_t^i \leq C_j^V, \qquad j = 1, ..., n_v; \tag{33}$$

$$\tag{34}$$

To deal with temporal dependency requirements, we introduce a new variable representing the number of the components that have completed the first test each time that such a dependency occurs. Let $\hat{\mathcal{I}}$ be the set of components going through two tests with a dependency requirement. For each $i \in \hat{\mathcal{I}}$ and $t = 1, ..., T$, we define a new variable $w^{i'}(1,t)$ and a new constraint

$$w^{i'}(1,t) \geq w^i(1, t + \tau_i),$$

where τ_i is the lead time (rounded to weeks) of the first test. The cases where the dependency is defined for a component in more than two tests can be treated in a similar fashion.

6.2 Dynamic Capacity

To deal with peak demands, capacity can be added dynamically using a special type of test machines called "WIP drivers", which are built to increase the production capacity during the peak demand period. These WIP drivers can be disassembled after the peak demand period and the components are used to fulfill demand.

Let $\mathcal{K}_0$ denote the special class of the WIP test machines. We have $\mathcal{K}_0 = \{K_1, K_2, \cdots, K_n\}$, where K_j represents the MTM of the WIP test machines for component type j, $j = 1, \cdots, n$. Two more classes of variables need to be added to the original problem,

- $x^{K_j}(1,t)$ and $x_t^{K_j}$ are the cumulative amount of the testing machines for component type j produced up to time t and increment at time t respectively;
- $y^{K_j}(1,t)$ and $y_t^{K_j}$ denote the cumulative amount of the amount of the test machines disassembled up to time t and increment at time t respectively.

Another change is that the machine capacity is no longer constant C_j^M any more, but

$$C_{j,t}^M = C_{j,0}^M + v_j[x^{K_j}(1,t) - y^{K_j}(1,t)],$$

where v_j denote the contribution of WIP testing machine K_j to the test capacity for commodity type j, and $C_{j,0}^M$ denotes the initial testing capacity.

An additional feasibility constraint is

$$y^{K_j}(1,t) \leq x^{K_j}(1,t).$$

The labor cost in the objective function needs to be changed as follows

$$\pi_t[\sum_{K\in\mathcal{K}\cup\mathcal{K}_0} x_t^K + \sum_{K\in\mathcal{K}_0} y_t^K - C_t^L]^+.$$

To reflect the parts inventory of the disassembled WIP drivers, we need to introduce a set of variables to represent the build quantity (including the disassembled parts) of each component. Let w_t^i be the build quantity for component i in period t, $i \in \mathcal{I}$ and $1 \le t \le T$. Similarly, let

$$w^i(1,t) = \sum_{\tau=1}^{t} w_\tau^i.$$

The objective is to minimize the component inventory cost and the overtime labor cost.

$$\begin{aligned} \min \quad & \sum_{t=1}^{T}\{\sum_{i\in\mathcal{I}} h_t^i \mathsf{E}[w^i(1,t) - \sum_{K\in\mathcal{K}}\sum_{K\ni i} u_i^K D^K(1,t)]^+ \\ & +\pi_t[\sum_{K\in\mathcal{K}\cup\mathcal{K}_0} x_t^K + \sum_{K\in\mathcal{K}_0} y_t^K - C_t^L]^+\}. \end{aligned} \tag{35}$$

The constraints are:

$$\sum_{i\in A_j}[w_t^i - \sum_{Kin\mathcal{K}_0}\sum_{K\ni i} u_i^K y_t^K] \le C_{j,0}^M + v_j[x^{K_j}(1,t) - y^{K_j}(1,t)], \qquad j = 1,...,n; \tag{36}$$

$$y^{K_j}(1,t) \le x^{K_j}(1,t), \qquad j = 1,...,n; \tag{37}$$

$$x^K(1,t) \ge \mu^K(1,t) + \sigma^K(1,t)\cdot\epsilon_{\alpha^K}, \quad K\in\mathcal{K},\ t = 1,...,T; \tag{38}$$

$$w^i(1,t) \ge \sum_{K\in\mathcal{K}\cup\mathcal{K}_0}\sum_{K\ni i} u_i^K x^K(1,t), \qquad t = 1,...,T. \tag{39}$$

Since this formulation has the same structure as the original problem (i.e., a convex program with linear constraints), the solution methods discussed in Section 4 still apply.

7. Concluding Remarks

In this paper we presented two variants of a production planning problem in hybrid push-pull systems. We exploited the structure of the problem formulation to develop efficient numerical algorithms, in particular a backward-recursion algorithm that solves the problem optimally through decomposition. Both variants of the model have distinct advantages over a commonly implemented production smoothing strategy. Numerical experimentation with realistic production data produced several business insights, in particular regarding the effects of component-based manufacturing. Specifically we demonstrated that the component-based model achieves much lower levels of inventory by postponing the manufacturing of high-dollar parts, thus reducing the risk and expense associated with building up component inventories in advance of receiving

actual customer orders. We described several extensions to incorporate additional real-world features and demonstrated that the resulting models can also be solved using the solution methods developed in this paper.

References

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

Aviv, Y. and Federgruen, A. (1997). "Stochastic Inventory Models with Limited Production Capacity and Periodically Varying Parameters," Probability in the Engineering and Informational Sciences, 11, 107-135.

Aviv, Y. and Federgruen, A. (2001). "Capacitated Multi-item Inventory Systems with Random and Seasonally Fluctuating Demands: Implications for Postponement Strategies," Management Sciences, 47(4), 512-531.

Beyer, D. and Ward, J. (2002). Network Server Supply Chain at HP: A Case Study. In: Song, J. and Yao, D. (eds.) *Supply Chain Structures: Coordination, Information and Optimization*. Kluwer.

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

Dietrich, B. et al. (2005). Applications of Implosion in Manufacturing. To appear in: An, C. and Fromm, H. (eds.) *Advances in Supply Chain Management*, Springer.

Federgruen, A. and Zipkin, P. (1986a). "An Inventory Model with Limited Production Capacity and Uncertain Demands I. The Average-Cost Criterion.," Mathematics of Operations Research, 11(2), 193-207.

Federgruen, A. and Zipkin, P. (1986b). "An Inventory Model with Limited Production Capacity and Uncertain Demands II. The Discounted-Cost Criterion", Mathematics of Operations Research, 11(2), 208-215.

Feitzinger, E. and Lee, H. (1997). Mass Customization at Hewlett Packard: The Power of Postponement. *Harvard Business Review*, 75, 1, 116-121.

Glasserman, P. and Tayur, S. (1995). "Sensitivity Analysis for Base-Stock Levels in Multi-Echelon Production-Inventory Systems," Management Science, 42(5), 263-281.

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

Kaminsky, P. and Swaminathan, J.M. (2004). "Effective Heuristics for Capacitated Production Planning with Multiperiod Production and Demand with Forecast Band Refinement", Manufacturing and Service Operations Management 6, 184-194.

Kapuscinski, R. and Tayur, S. (1998). "A Capacitated Production-Inventory Model with Periodic Demand," Operations Research, 46(6), 899-911.

Lu, Y., Song, J.S. and Yao, D.D. (2003). "Order Fill Rate, Leadtime Variability, and Advance Demand Information in an Assemble-to-Order System". *Operations Research,* 51, 292-308.

Lu, Y., Song, J.S. and Yao, D.D. (2005). Backorder Minimization in Multiproduct Assemble-to-Order Systems. To appear in *IIE Transactions.*

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

Song, J.S. and Yao, D.D. (eds.) (2002). *Supply Chain Structures: Coordination, Information, and Optimization,* Kluwer.

Song, J.S. and Zipkin, P. (2003). Assemble-To-Order Systems. In: de Kok A.G. and Graves, S. (eds.) *Handbooks in Operations Research and Management Science, Vol. 11. Supply Chain Management: Design, Coordination and Operation*. Elsevier.

Zipkin, P. (2000). *Foundations of Inventory Management*, McGraw-Hill.