Manufacturing & Service Operations Management

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

Requirements Planning with Substitutions: Exploiting Bill-of-Materials Flexibility in Production Planning

Anantaram Balakrishnan, Joseph Geunes,

Requirements Planning with Substitutions: Exploiting Bill-of-Materials Flexibility in Production Planning

Anantaram Balakrishnan • Joseph Geunes
Smeal College of Business Administration, Penn State University, University Park, Pennsylvania 16802
anantb@psu.edu • geunes@ise.ufl.edu
Department of Industrial and Systems Engineering, The University of Florida, Gainesville, Florida 32611

Designing product lines with substitutable components and subassemblies permits companies to offer a broader variety of products while continuing to exploit economies of scale in production and inventory costs. Past research on models incorporating component substitutions focuses on the benefits from reduced safety-stock requirements. This paper addresses a dynamic requirements-planning problem for two-stage multiproduct manufacturing systems with bill-of-materials flexibility, i.e., with options to use substitute components or subassemblies produced by an upstream stage to meet demand in each period at the downstream stage. We model the problem as an integer program, and describe a dynamic-programming solution method to find the production and substitution quantities that satisfy given multiperiod downstream demands at minimum total setup, production, conversion, and holding cost. This methodology can serve as a module in requirements-planning systems to plan opportunistic component substitutions based on relative future demands and production costs. Computational results using real data from an aluminum-tube manufacturer show that substitution can save, on average, 8.7% of manufacturing cost. We also apply the model to random problems with a simple product structure to develop insights regarding substitution behavior and impacts.
(*Inventory-Production: Applications; Inventory-Production: Scale-Economies-Lot Sizing; Inventory-Production: Deterministic; Network-Graphs: Applications*)

1. Introduction

Advances in manufacturing and product technologies and improvements in product design have greatly increased not only process flexibility and product variety but also what we might call "bill-of-materials (BOM) flexibility"—the ability to opportunistically use different raw materials, parts, or subassemblies in the manufacture of an end-item. Consider, for instance, today's personal computer systems. Their modular architecture permits replacing a component or card with an alternate component or card without reducing the functionality of the end-item. So, if the standard card needed for a particular system configuration is not readily available, the manufacturer can still meet demand by substituting the missing card with one that provides equal or greater functionality. Similar substitution opportunities arise in other contexts. For instance, in the manufacture of aluminum tubes, each finished tube has a preferred extrusion size to use as input to the tube drawing process, but can also be produced from other available sizes, but at higher cost (Balakrishnan and Brown 1996). Likewise, we can substitute higher-strength alloys for lower-grade ones, or use higher-grade integrated circuits (that meet, say,

1523-4614/00/0202/0166$05.00
1526-5498 electronic ISSN

more stringent military specifications) for less demanding applications (Rosolen 1987, Bitran and Dasu 1992). This paper addresses requirements planning in a multiproduct manufacturing environment with component substitution options in the BOM. We use the word "component" to refer to any raw material, part, or subassembly used to produce an end-item.

As the above examples illustrate, BOM flexibility arises naturally in some industries (e.g., continuous material-deformation processes) but has also become common in assembly operations as companies introduce greater modularity in their product designs (Ulrich et al. 1993). Modular product design, a key element of mass customization (Pine 1993), permits manufacturers to exploit economies of scale by producing just a few basic module types, and yet cover many different market segments by combining the modules into numerous product variants. The consumer-electronics industry provides many examples of the successful use of this strategy. The product-design literature stresses cost savings due to component commonality for modular products; however, modularity also provides opportunistic substitution options that, as we emphasize in this paper, can provide even further savings.

As process and product flexibility have increased, so has recent research on models to plan and effectively exploit this flexibility. For instance, the literature on *process flexibility* addresses a wide spectrum of issues ranging from strategic capacity planning decisions (e.g., Jordan and Graves 1995, Fine and Freund 1990) to detailed operations-scheduling issues (e.g., Tang and Denardo 1988a, 1988b). Researchers have also demonstrated the risk-pooling benefits of component *commonality* under various service-level constraints, assuming a fixed BOM structure (e.g., Baker et al. 1986 and Gerchak et al. 1988). However, understanding the benefits of *BOM flexibility* has not received as much attention in the modeling literature. This paper attempts to partially fill this gap by developing and testing, using real and simulated data, a dynamic requirements-planning model that incorporates component substitution options.

We focus on planned component substitutions in manufacturing or assembly, i.e., the deliberate decisions of a planner to use substitute components well ahead of actual production. What circumstances might lead a planner to substitute a preferred part with an alternate component despite having to incur additional conversion costs? Substitutability can provide manufacturing cost savings through reduced setup and inventory costs. For instance, if the demand for a product is highly seasonal, using a substitute component during low-demand periods might be more economical than producing and holding the preferred component. Here, we exploit "demand pooling", i.e., aggregating the demand over several products to reduce unit production costs. Substitutability provides other benefits as well. For new products (e.g., new automobile models), using an existing component in initial production runs until a new replacement component is developed can considerably speed up time-to-market. Conversely, we might use a new component in an older product to avoid producing and stocking the older component.

To support production-planning decisions in these contexts, we consider the following *Requirements Planning with Substitutions (RPS)* problem: Given the flexible bill-of-materials for each product (or end-item) and its projected (deterministic) demand during every period of the planning horizon, determine when and how much of each component to produce and hold in inventory, and when to use substitute components in order to meet product demands at minimum total cost. We consider setup and variable production costs, holding costs, and substitution or *conversion* costs for each component. The per-unit conversion cost represents any additional processing effort or cost incurred when we substitute a preferred component with an alternate component. Our model can incorporate both *complete* or *product-specific* substitutions, i.e., we permit a particular part to substitute for another component in either all the products that contain the second component or only a specified subset of these products.

Our RPS model differs from two other types of substitution phenomena addressed in the inventory-management literature—*demand* substitution and *reactive* substitution. Demand substitution refers to the consumer's choice of an alternate product, at the time of purchase, if the preferred product is not available. Product assortment-planning models (e.g., Agrawal and Smith 1996), based on assumptions about consumer choice, incorporate this "customer flexibility" to decide optimal retailer stocking policies. A second stream of literature considers reactive or online substitutions—when a plant runs out of stock for a preferred component, it uses a substitute in order to maintain desired service levels and meet delivery commitments for end-items. Ignall and Veinott (1969), McGillivray and Silver (1978), and Bassok et al. (1996) address inventory management in this context. Similarly, Bitran and Dasu (1992), Ou and Wein (1995), and Bitran and Gilbert (1994) discuss stochastic models incorporating substitutions in the presence of random yields, with applications to semiconductor manufacturing and hotel room assignment. To maintain tractability, all of these models make certain simplifying assumptions, e.g., they assume independent product demands or special substitution structures such as full downward substitution, and do not incorporate setup costs or other economies of scale in production. Our RPS solution method can serve as a module in existing hierarchical-planning (e.g., Hax and Meal 1975) or MRP systems to plan the production and use of substitute components. Currently, a few systems contain heuristic rules to evaluate reactive substitution options (Rosolen 1987), but none of the commercially-available MRP systems appear to incorporate the ability to plan substitutions based on production and inventory cost tradeoffs.

The RPS problem requires simultaneously considering the production plans for multiple items. As in much of the prior literature on commonality and substitutability, we focus on the interactions between two successive manufacturing stages. This "single-level" version is applicable to contexts such as the production of printed circuit boards for final assembly of make-to-order personal computer systems, or the production of aluminum extrusions to be used for manufacturing drawn tubes (see §3.2), both of which entail substitutions only at the final make-to-order stage. The single-level problem is also well-suited to generate insights regarding substitution behavior. We propose a dynamic-programming solution method that generalizes the classical single-item lot-sizing algorithm (Wagner and Whitin 1958). A convenient shortest-path interpretation over an appropriately defined network serves to intuitively explain the algorithm. The multilevel problem with substitutions generalizes the multilevel lot-sizing model, which itself is very difficult to solve optimally (e.g., Afentakis et al. 1984, Afentakis and Gavish 1986). In the concluding section of this paper, we describe opportunities to extend the single-level method and its underlying principles to the multilevel setting.

We implemented and applied the (single-level) RPS algorithm to two sets of test problems: one based on real data from aluminum-tube manufacturing to verify the algorithm's practical viability and savings potential, and the second using simulated data for a simple product structure to understand substitution drivers and patterns under various cost and demand scenarios. For the tube-manufacturing application, our results indicate potential savings in total tube fabrication costs of 8.7% on average. Our tests on random problems reveal some interesting phenomena. For instance, under certain conditions, substitution can actually increase when the cost parameters for the substitute component increases.

The remainder of this paper is organized as follows. Section 2 formulates the RPS problem as a mixed-integer program, represents it as a generalized network-flow problem with concave costs, identifies key properties of optimal solutions, and develops the dynamic-programming solution approach via an equivalent shortest-path representation. We also outline modifications needed to handle certain model extensions. Section 3 presents our computational results, and discusses their implications. Section 4 briefly discusses extensions to multistage planning.

2. Problem Definition and Formulation

Let N denote the number of products or end-items, with flexible bills-of-materials, produced by a manufacturer. Given each product's projected demand in every period of a T-period planning horizon, and the costs for producing, holding, and converting components, the Requirements Planning with Substitutions (RPS) model decides how much of each component to produce and hold in every period and when to use substitute components in order to meet all demands at minimum total cost. Our solution method assumes that all components have zero initial inventories, and all costs are nonnegative. To keep the discussion simple, we initially assume that each product requires only one component type, and backlogging is not permitted.

2.1. Notation and Formulation

Let $I = \{1, 2, \ldots, N\}$ be the index set of *products*, d_{it} the projected *demand* for product $i \in I$ in period $t = 1, 2, \ldots, T$, and $J = \{1, 2, \ldots, M\}$ the set of all *component types* needed to manufacture the products in I. We assume that end-items are produced in each period to meet demand in that period. Producing component j in period t incurs a fixed *setup* cost s_{jt} and a variable *production* cost of p_{jt} per unit. Let h_{jt} denote the cost of holding one unit of component j in inventory during period t, assessed for convenience against ending inventory in each period. We assume zero production lead times, although positive deterministic lead times are easy to incorporate in the model. Our initial RPS model assumes that every product needs only one input component type (i.e., no assembly). Let $J(i) \subseteq J$ be the subset of alternative component types needed for each product i. In general, the model permits product-specific substitutions through appropriate choice of $J(i)$. For the special case of complete substitutions, if component j can always substitute for component j', then $j \in J(i)$ whenever $j' \in J(i)$. For each $j \in J(i)$, let a_{ij} denote the number of units of component j needed to produce one unit of product i, and let $\alpha_{ij} = 1/a_{ij}$. Using component $j \in J(i)$ to manufacture product i in period t incurs a nonnegative *conversion* cost of c_{ijt} per unit of the component. This conversion cost might represent, for instance, the incremental processing cost if we use component j instead of the product's preferred component. In the aluminum-tube manufacturing setting, where products represent drawn tubes of various sizes and components are extrusions, the conversion cost c_{ijt} captures the cost to reduce an extrusion of size j to a finished tube of size i (see §3.2). The larger the extrusion size, the greater the cost to produce a particular (smaller) finished tube. Likewise, in electronics assembly, using a substitute component or card might require special processing.

To formulate the RPS problem we define the following decision variables. For all $j \in J$ and $t = 1, 2, \ldots T$, let:

X_{jt} equals number of units of component j produced in period t;

Y_{jt} equals 1 if we produce component j in period t, and 0 otherwise;

H_{jt} equals inventory of component j at the end of period t; and

W_{ijt} equals number of units of component j used to produce product i in period t, for all $i \in I(j)$;

where $I(j) = \{i \in I: j \in J(i)\}$ is the set of all products for which component j can be used. The RPS problem then has the following mixed-integer programming formulation:

$$\text{[RPS] Minimize } \sum_{t=1}^{T} \sum_{j \in J} \{s_{jt}Y_{jt} + p_{jt}X_{jt} + h_{jt}H_{jt}\} + \sum_{t=1}^{T} \sum_{j \in J} \sum_{i \in I(j)} c_{ijt}W_{ijt} \tag{1}$$

subject to:

$$\textit{Inventory balance: } H_{j,t-1} + X_{jt} = \sum_{i \in I(j)} W_{ijt} + H_{jt} \quad \text{for all } j \in J,\ t = 1, \ldots, T, \tag{2}$$

$$\textit{Zero Initial Inventory: } H_{j0} = 0 \text{ for all } j \in J, \tag{3}$$

$$\textit{Component usage: } \sum_{j \in J(i)} \alpha_{ij}W_{ijt} = d_{it} \quad \text{for all } i \in I,\ t = 1, \ldots, T, \tag{4}$$

$$\textit{Setup forcing: } X_{jt} \le M_{jt}Y_{jt} \quad \text{for all } j \in J,\ t = 1, \ldots, T, \tag{5}$$

$$\begin{aligned} &\textit{Nonnegativity: } X_{jt}, H_{jt}, W_{ijt} \ge 0 \\ &\textit{Integrality: } Y_{jt} \in \{0, 1\} \\ &\qquad \text{for all } i \in I,\ j \in J,\ t = 1, \ldots, T. \end{aligned} \tag{6}$$

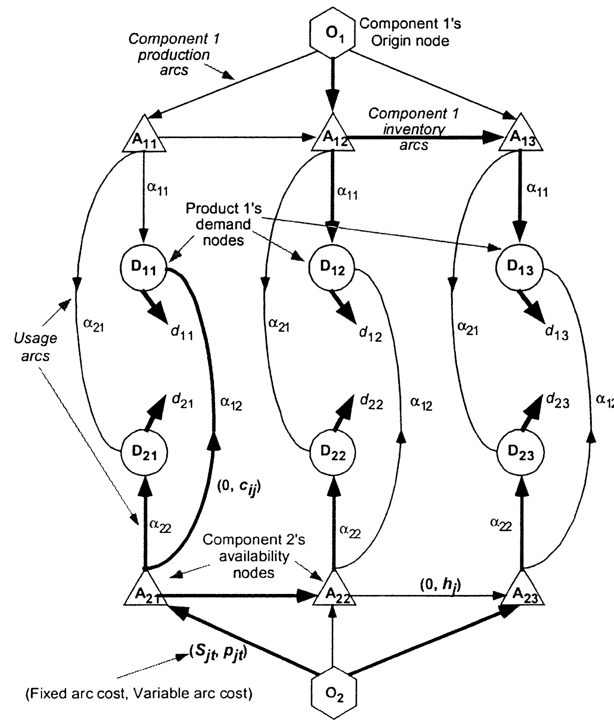
The Objective Function (1) minimizes the total T-period setup, production, holding, and conversion costs for components. Constraints (2) and (3) represent the inventory balance identities for components. Constraints (4) state that, in every period, the total usage of all component types $j \in J(i)$ that can produce product i equals the demand for that product. The setup forcing Constraint (5) ensures that we incur the setup cost (by setting $Y_{jt} = 1$) if we produce component j in period t (i.e., if $X_{jt} > 0$). The constant M_{jt} in this constraint is a large number, say, equal to the total number of components needed to meet the demand for all products $i \in I(j)$ in all periods from t to T. Because costs are nonnegative, the problem has an optimal solution that sets all ending inventories to zero. The model can, however, easily incorporate any desired positive ending inventory levels.

2.2. Generalized Network-Flow Representation

Our solution algorithm relies on certain key properties of optimal RPS solutions. To establish these properties, we first view the RPS problem as the following generalized network-flow problem with concave costs, defined over a directed network G containing three types of nodes: *origin* nodes O_j, component *availability* nodes A_{jt}, and product demand nodes D_{it} for each product i, component j, and period t. Three types of arcs interconnect these nodes: *production* arcs from each origin node O_j to each of component j's availability nodes A_{jt}, *inventory* arcs from $A_{j,t-1}$ to A_{jt} for all components j and all periods $t > 1$, and *usage* arcs from A_{jt} to D_{it} for every product $i \in I(j)$ and all periods t. Each demand node D_{it} has a demand of d_{it} that must be supplied from the origin nodes. For convenience, we add a supersource with unlimited supply, connected to each of the origin nodes. Figure 1 illustrates this network construction for a simple two-product, two-component example with a planning horizon of three periods. In this example, each product requires one unit of either component type.

By interpreting the flows on the production arcs (O_j, A_{jt}), inventory arcs ($A_{j,t-1}$, A_{jt}), and usage arcs (A_{jt}, D_{it}) as the values of the X_{jt}, $H_{j,t-1}$, and W_{ijt} variables, respectively, we observe that flow-conservation requirements at the component availability nodes A_{jt} correspond to the inventory-balance Constraints (2) in formulation [RPS]. The usage Constraints (4) translate to "generalized" flow-conservation equations (see, for instance, Ahuja et al. 1993) at the demand nodes, with a weight of α_{ij} assigned to the flow on each incoming arc (A_{jt}, D_{it}) at node D_{it}. Hence, any feasible solution (satisfying all demands) to the generalized network-flow problem defined on G corresponds to a feasible plan for producing, stocking, and using components in the RPS problem. The dark lines on Figure 1 show a sample solution to our two-product example. This solution produces Component 2 in Period 1 to meet the demand for Products 1 and 2 in Period 1, and for Product 2 in Period 2. In Period 2, we setup and produce

Figure 1 **Generalized Network Flow Representation of Two-Product Example**



Bold arcs indicate sample solution to two-product example.

enough of Component 1 to meet the demand for Product 1 in Periods 2 and 3. Finally, in Period 3 we again setup and produce Component 2 to meet Product 2's demand in that period.

If the arc cost structure consists of a fixed charge for selecting the arc, and a per-unit flow cost for routing flows on the arc, then the problem of finding the minimum cost generalized network flow on G resembles the well-known uncapacitated, fixed-charge network-design model (e.g., Magnanti and Wong 1984) but with generalized flow-conservation requirements at all destination nodes. In particular, we can model the fixed and variable cost structure of formulation [RPS] by assigning a *fixed charge* of s_{jt} and a per-unit *flow cost* of p_{jt} to each production arc (O_{jt}, A_{jt}), and per-unit flow costs of $h_{j,t-1}$ and c_{ijt}, respectively, to the inventory arcs ($A_{j,t-1}$, A_{jt}) and the usage arcs (A_{jt}, D_{jt}). For this special case, the problem reduces further to a facility-location model in which facilities correspond to component-period pairs, and customers to product-period pairs.

2.3. Properties of Optimal RPS Solutions

Using the generalized network-flow interpretation, we establish the required properties of optimal RPS solutions for the general case when the cost on each arc (production, holding, or conversion costs) can be any nonnegative, concave function of the flow on that arc.

PROPOSITION 1. ROOTED TREE SOLUTION. *With concave arc costs, the equivalent generalized network-flow problem defined on G has an optimal rooted-tree solution, i.e., the set of arcs carrying positive flow in this solution form a directed tree rooted at the supersource.*

PROOF. See Appendix A.

This rooted-tree property resembles the spanning tree characterization of optimal (extreme point) solutions to the standard minimum (concave) cost network-flow problem, without gains or losses, i.e., with $\alpha_{ij} = 1$ on all arcs. To our knowledge, the property has not been previously extended to the case when generalized flow conservation holds at the destination nodes.

Proposition 1 implies the following two properties of optimal RPS solutions. Because the rooted tree has at most one arc entering any availability node A_{jt}, we have:

Zero Inventory-Production (ZIP) Property. The RPS problem has an optimal solution in which, for any component $j \in J$ and period $t = 2, \ldots, T$, the solution either produces component j in period t or carries inventory of this component from period $t - 1$ to t (or neither), but not both.

Likewise, since each demand node has only one incident arc carrying positive flow, we obtain:

Homogeneous Product Lots (HPL) Property. The RPS problem has an optimal solution that, for every product $i \in I$ and period $t = 1, \ldots, T$, produces all d_{it} units of product i demanded in period t using only one component type $j \in J(i)$.

The ZIP property is well known for the single-item lot-sizing problem and some of its generalizations (Wagner and Whitin 1958). The HPL property, equivalent to the single-source assignment property for uncapacitated facility location (e.g., Cornuejols et al. 1990), reduces the decision on how much of each component to use for product i in period t to one of selecting a single component j from $J(i)$ to meet all of product i's demand d_{it} in that period.

Using the HPL property, suppose the rooted-tree solution produces all units of product i in period t using one component type $j \in J(i)$, and suppose the most recent batch of component j was produced in period t' $\leq t$. Then, all of the type j components used for product i in period t must have been produced in period t'. Otherwise, the solution carries inventory of component j from $(t' - 1)$ to t' while simultaneously producing this component at t', violating the ZIP property. Hence, the HPL and ZIP properties have the following important corollary:

Most Recent Usage (MRU) Property. The RPS problem has an optimal solution that produces all d_{it} units of any product $i \in I$ demanded in a period t using only components from the most recent batch (at or before period t) of component type j, for some $j \in J(i)$.

Thus far, we have assumed only that the production, holding, and conversion costs are concave functions. If we further assume that production (setup plus variable) costs do not increase over time (and, assuming for simplicity, zero production lead times) the following property holds:

Immediate Usage (IU) Property. Under nonincreasing setup and variable production costs, the RPS problem has an optimal solution such that, if component j is produced in period t, this component type is used to satisfy the immediate (same period) demand for at least one product $i \in I(j)$, i.e., the solution never produces components solely for stock.

PROOF. Because production costs do not increase with time, and holding costs are nonnegative, given any solution that produces a component solely for stock, we can construct another feasible solution with equal or lower total cost by deferring the production of that component until the first subsequent period when it is used. □

Assuming that conversion costs do not increase with time, the IU property also applies to more general multilevel settings, implying "nestedness" of production setups. That is, whenever we set up to produce component j, we must also set up and produce in the same period (or after a lag equal to component j's production lead time) at least one downstream component k that uses component j. In turn, component k must be used immediately by one of its downstream stages, and so on. Thus, component j must be used in at least one end-item that is shipped in that period. Love (1972) established this property for serial production systems without substitutions under nonincreasing production costs.

The solution method for single-level RPS problems that we present next relies primarily on the ZIP, HPL, and MRU properties. As we note later, the IU property can be valuable for solving multilevel problems.

2.4. Shortest-Path Solution Method

For single-level RPS problems with fixed plus variable costs, as reflected in formulation (RPS), this section develops a dynamic-programming solution method by viewing (RPS) as a shortest-path problem over an appropriate network.

We define a *setup vector* $\mathbf{U}$ as an M-vector with integer elements $u_j \in \{0, 1, \ldots, T\}$, for all $j \in J$, where each u_j represents the index of a time period in which component j is produced (with $u_i = 0$ if component j has not been produced thus far). Define $t(\mathbf{U}) = \max \{u_j: j \in J\}$ as the *reference period* corresponding to setup vector $\mathbf{U}$, and let

$$F(\mathbf{U}) = \sum_{j:u_j = t(\mathbf{U})} s_{j,t(\mathbf{U})}. \tag{7}$$

$F(\mathbf{U})$ gives the *setup cost* associated with $\mathbf{U}$. For a given production plan, we say that $\mathbf{U}$ is the *most recent* setup vector at period $t \geq t(\mathbf{U})$ if, for every component $j \in J$, the given plan produces component j in period u_j but not in any of the periods $u_j + 1, \ldots, t$. By the MRU property, this information on most recent setups is adequate to compute the marginal cost of meeting each product's demand in period t. In particular, for any $i \in I$, the production, holding, and conversion cost of using component $j \in J(i)$ to produce each unit of product i in period t, given that this component was last produced in period u_j, is $(p_{j,u_j} + \sum_{i=u_j}^{t-1} h_{ji} + c_{ijt})a_{ij}$. The best component to use is the one that minimizes this cost. Hence, the variable cost of meeting product i's demand in period t, given that $\mathbf{U}$ is the most recent setup vector, is

$$C_{it}(\mathbf{U}) = \min_{j \in J(i)} \left(p_{j,u_j} + \sum_{i=u_j}^{t-1} h_{ji} + c_{ijt}\right) a_{ij} d_{it}. \tag{8}$$

So, if we know the sequence of setup periods for each component, and hence the corresponding setup vectors $\mathbf{U}$ "visited" by the production plan, we can compute the total cost of this plan by adding the associated setup costs $F(\mathbf{U})$ to the total variable cost of meeting each product's demand in every period.

To find the best plan, we consider a shortest-path problem defined on the following directed, layered network G'. The network contains one node for each possible setup vector $\mathbf{U}$; for convenience, we use $\mathbf{U}$ to denote both the setup vector and its corresponding node in G'. For $l = 1, \ldots, T$, we include in layer l all nodes $\mathbf{U}$ with $t(\mathbf{U}) = l$. Note that Layer 1 contains only nodes $\mathbf{U}$, with $u_j = 0$ or 1 for all $j \in J$, such that the subset of component types produced, $J' = \{j \in J: u_j = 1\}$, can cover all products. Arcs of G' will represent valid setup transitions. That is, from any node $\mathbf{U}$, G' contains arcs $(\mathbf{U}, \mathbf{V})$ only to nodes $\mathbf{V}$ in higher-indexed layers (i.e., with $t(\mathbf{V}) > t(\mathbf{U})$) that have $v_j = u_j$ or $t(\mathbf{V})$ for all $j \in J$. In this case, we set the *length* of arc $(\mathbf{U}, \mathbf{V})$ to be:

$$d(\mathbf{U}, \mathbf{V}) = F(\mathbf{U}) + E(\mathbf{U}, \mathbf{V}), \tag{9}$$

where

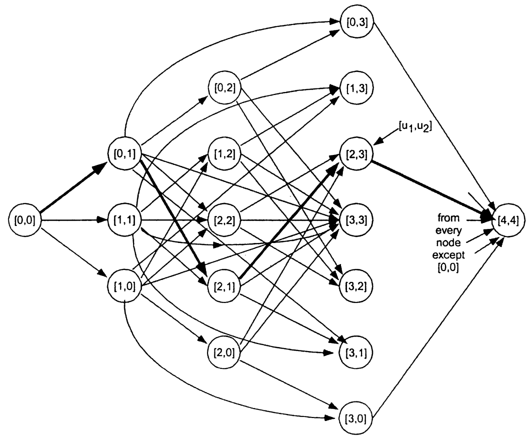
$$E(\mathbf{U}, \mathbf{V}) = \sum_{i \in I} \sum_{i=t(\mathbf{U})}^{t(\mathbf{V})-1} C_{it}(\mathbf{U}) \tag{10}$$

is the total variable (production, holding, and conversion) cost to meet all end-item demands in periods $t(\mathbf{U})$ to $t(\mathbf{V}) - 1$. The network G' also contains two additional nodes: a source node $\mathbf{0}$ and a destination node $\mathbf{T}'$, associated respectively with layers 0 and $(T + 1)$. Node $\mathbf{0}$ is connected via zero length arcs to all the nodes in Layer 1. Every node (except the source) is connected to the destination $\mathbf{T}'$; the length of arc $(\mathbf{U}, \mathbf{T}')$ is computed using (9) and (10), with $t(\mathbf{T}') = T + 1$. Figure 2 shows the shortest path network G' for our previous two-product example.

Clearly, every path from $\mathbf{0}$ to $\mathbf{T}'$ in G' corresponds to a valid component production plan, i.e., the nodes visited by this path represent the unique sequence of setup vectors that characterize the plan. For instance, the bold arcs in Figure 2 represent the $\mathbf{0}$-to-$\mathbf{T}'$ path corresponding to the sample production plan shown in Figure 1.

Moreover, the length of this path exactly equals the

Figure 2 **Equivalent Shortest-Path Network for Two-Product Example**



In the node labels $[u_1,u_2]$, u_1 and u_2 represent the most recent setup periods for components 1 and 2. Path defined by the bold arcs corresponds to sample solution to two-product example.

total setup, production, holding, and conversion cost incurred by the corresponding production plan. Hence, the length of the shortest path from 0 to $\mathbf{T}'$ in G' equals the optimal value of (RPS). The optimal solution is easy to reconstruct from this shortest path. For every node $\mathbf{U}$ that the path visits, set up and produce each component j in period u_j. Given these setups, find the best component to use for each product in every period using Equation (8), and based on these usage quantities determine the production quantity for each component batch.

Network G' contains $O(T^M)$ nodes and $O(T^{M+1})$ arcs, where M is the number of component types in J. So, although shortest-path algorithms require only polynomial time in the number of nodes or arcs, the method's worst-case performance is exponential in the number of component types when applied to the RPS problem.

Algorithmic Improvements. By exploiting the special structure of network G' and its arc-length parameters, we can reduce the actual (not worst-case) computational effort needed to find the shortest path. For instance, the network has a layered, acyclic structure that shortest-path methods might exploit. Moreover, its arc lengths $d(\mathbf{U}, \mathbf{V})$ depend only on the starting node $\mathbf{U}$ and the reference period $t(\mathbf{V})$ for the ending node $\mathbf{V}$. Hence, for any $t > t(\mathbf{U})$, $d(\mathbf{U}, \mathbf{V})$ is the same for all of node $\mathbf{U}$'s neighbors $\mathbf{V}$ in layer t. Moreover, certain quick tests can help eliminate some of these computations. For instance, if $pc^i_{\min} \equiv \min_{j\in J(i)} [\min\{(p_{jt} + c_{ijt})a_{ij}\colon t = 1, \ldots, T\}]$ denotes the minimum variable production plus conversion cost among all components for product i, then,

$$E(\mathbf{U}, \mathbf{V}) \geq \sum_{i\in I} \sum_{i=t(\mathbf{U})}^{t(\mathbf{V})-1} pc^i_{\min} d_{it}. \tag{11}$$

Adding $F(\mathbf{U})$ to the right-hand side of Inequality (11) gives an easily computed lower bound on the true arc length $d(\mathbf{U}, \mathbf{V})$ (more sophisticated bounds are also possible, but require more computations or storage). If the shortest-path length to node $\mathbf{U}$ plus this lower bound on $d(\mathbf{U}, \mathbf{V})$ exceeds the length of a current shortest-path to node $\mathbf{V}$, then arc $(\mathbf{U}, \mathbf{V})$ cannot belong to the optimal solution, and so we need not compute its length.

Special cost structures can lead to additional simplifications. Suppose the production costs do not increase with time, and the ranking of products according to their holding costs does not change with time. In this case, Balakrishnan and Geunes (1998) establish a property called the *Contiguous Usage* property that can reduce the number of comparisons needed for arc-length computations. Finally, let $L(\mathbf{U})$ denote the length of the shortest path from node 0 to node $\mathbf{U}$ in network G'. We can show that, for any two setup vectors $\mathbf{U}$ and $\mathbf{V}$ with $\mathbf{V} \geq \mathbf{U}$ and $\mathbf{V} \neq \mathbf{U}$, if $L(\mathbf{U}) + F(\mathbf{U}) \geq L(\mathbf{V}) + F(\mathbf{V})$ then G' contains a shortest $\mathbf{0}$-to-$\mathbf{T}'$ path that does not pass through node $\mathbf{U}$. This property holds because, under the specified conditions, in reaching node $\mathbf{V}$ we have met all of the product demands covered until node $\mathbf{U}$ at equal or lower cost. Furthermore, because of nonincreasing costs, we can meet demand in future periods from the setups corresponding to $\mathbf{V}$ less expensively than supplying these demands from the setups corresponding to $\mathbf{U}$. Hence, we can disregard node $\mathbf{U}$ in future computations.

2.5. Model Extensions

We now consider extensions of the shortest-path method to handle backlogging or single-level assembly.

2.5.1. Backlogging. Instead of producing end-items in each period, suppose we are permitted to delay the delivery of products, but incur a *backlogging cost* of b_{it} per unit of backlogged demand for product i in period t. We assume that each product requires only one input component type, and that the demand backlogging cost is higher than any decrease in the variable production plus conversion costs from one period to the next, i.e., $(p_{jt} + c_{ijt}) - (p_{j,t+1} + c_{ij,t+1}) \le b_{it}$ for all $j \in J(i)$. The latter assumption ensures that product demands, if backlogged, are met from the next production batch of the appropriate component. Under these conditions, we can show that the following additional property holds:

All-or-Nothing Backlogging Property. The RPS problem with backlogging has an optimal solution that either backlogs the entire demand d_{it} of product i in period t to some future period $\bar{t}$ or meets all of the demand in period t itself. If demand is backlogged and component j is used to meet this demand, then $\bar{t}$ must be the next setup period, after t, for component j.

We exploit this property to adapt our shortest-path algorithm by associating nodes of the network with setup vector pairs $\{\mathbf{U}, \hat{\mathbf{U}}\}$, where the elements $\hat{u}_j$ of $\hat{\mathbf{U}} > \mathbf{U}$ represent the next setup period after u_j for each component j. Including the node $\{\mathbf{U}, \hat{\mathbf{U}}\}$ in an origin-to-destination path corresponds to selecting a production plan that produces component j in period u_j and next produces this component in period $\hat{u}_j$, for all $j \in J$. The shortest-path network contains an arc from $\{\mathbf{U}, \hat{\mathbf{U}}\}$ to $\{\mathbf{V}, \hat{\mathbf{V}}\}$ only if $t(\mathbf{V}) > t(\mathbf{U})$ and, for each $j \in J$, either $v_j = u_j$ and $\hat{v}_j = \hat{u}_j > t(\mathbf{V})$, or $v_j = \hat{\boldsymbol{u}}_{\boldsymbol{j}} = t(\mathbf{V})$ and $\hat{v}_j > v_j$. As before, the cost of this arc equals the fixed setup cost $F(\mathbf{U})$ for all components produced in period $t(\mathbf{U})$, plus the minimum variable cost to meet the demand for all products in the time interval from $t(\mathbf{U})$ to $t(\mathbf{V}) - 1$. The variable cost component must now include the possibility of backlogging in addition to the production, holding, and conversion costs. Accordingly, we define $C_{it}(\mathbf{U}, \hat{\mathbf{U}})$, the minimum (variable) cost of meeting product i's demand in period t, as follows:

$$C_{it}(\mathbf{U}, \hat{\mathbf{U}}) = \min_{j \in J(i)} \left\{ \min\left[\left(p_{j,\hat{u}_j} + \sum_{\bar{t}=t}^{\hat{u}_j - 1} b_{i\bar{t}} + c_{ij\hat{u}_j} \right), \left(p_{j,u_j} + \sum_{\bar{t}=u_j}^{t-1} h_{j\bar{t}} + c_{ijt} \right) \right] \right\} a_{ij} d_{it}, \tag{12}$$

where the two inner terms represent, respectively, the costs of using component j to meet product i's demand in period t with and without backlogging. Adding this cost for all products $i \in I$ and all periods $t = t(\mathbf{U}), \ldots, t(\mathbf{V}) - 1$ gives the variable cost $E(\{\mathbf{U}, \hat{\mathbf{U}}\}, \{\mathbf{V}, \hat{\mathbf{V}}\})$. We set the arc cost $d(\{\mathbf{U}, \hat{\mathbf{U}}\}, \{\mathbf{V}, \hat{\mathbf{V}}\})$ equal to $F(\mathbf{U}) + E\{\mathbf{U}, \hat{\mathbf{U}}\}, \{\mathbf{V}, \hat{\mathbf{V}}\})$.

The length of the shortest origin-to-destination path in this expanded network, containing $O(T^{2M})$ nodes and $O(T^{2M+1})$ arcs, is the optimal value of the RPS problem with backlogging.

2.5.2. Single-Level Assembly. A minor adaptation of the shortest-path method permits handling single-level assembly situations (without backlogging) when end-items require multiple component types that are substitutable. As before, we permit product-specific substitutions, but now we must consider two additional possibilities: whether or not the substitution options for a product depend on which other component types are chosen.

Under *independent substitutions,* a component j can substitute for component j' in product i regardless of the other component types chosen for the product. Suppose, for instance, product i requires two input components, one each from component subsets $J_1(i)$ and $J_2(i)$. With independent substitutions, we can independently select which component to use from each group. In this case, we can equivalently replace the single product i with two products, i_1 and i_2, and set $d_{i_1 t} = d_{i_2 t} = d_{it}$. We refer to this transformation as *product replication.* Because each product in the replicated problem requires only a single component type, our shortest-path method of §2.4 solves this problem.

Interacting substitutions is more restrictive: The choice of component in one group impacts the available choices in other groups. In its most general form, we need to consider the different possible product *configurations,* i.e., permissible combinations of component types needed for each product. All of the properties of optimal solutions discussed in §2.3 continue to hold for this model. In particular, the HPL property implies that, for all $i \in I$ and $t = 1, \ldots T$, all units of product i shipped in period t have the same configuration. So, the shortest-path representation and solution algorithm applies, except that we now choose the least-cost configuration among all possible configurations for product i in order to compute the cost parameter $C_{it}(\mathbf{U})$. Let k_i denote the number of possible configurations for product i, each with a unique bill of materials, and let $C_{ikt}(\mathbf{U})$ denote the total cost of producing d_{it} units of product i in period t using configuration k. This cost consists of the variable production, holding (since the last setup period u_j), and conversion costs for every component j in the bill of materials corresponding to configuration k. Then, $C_{it}(\mathbf{U}) = \min_{k=1,\ldots,k_i} C_{ikt}(\mathbf{U})$. Using Equations (9) and (10), we determine the arc lengths $d(\mathbf{U},\mathbf{V})$ in network G'. The shortest $\mathbf{0}$-to-$\mathbf{T}'$ path with these new arc lengths gives the optimal solution to the RPS problem with single-level assembly and interacting substitutions. If substitutions are independent for some products and component groups but interacting in others, we can adopt a hybrid approach combining product replication and configuration selection.

Finally, we note that combining backlogging and assembly makes the problem quite complex because we need to keep track of more than one previous and next setup period for each component. As discussed in §4, similar complexities arise for multilevel problems. Section 4 outlines some algorithmic strategies to approximately solve multilevel problems using the single-level solution method.

3. Computations

Our goals in developing the RPS model and solution method were both to validate its practical viability and potential savings, and to understand situations under which substitutions can be effective. Accordingly, we implemented our shortest-path algorithm for the single-level RPS problem, and performed two sets of computational tests. The first set of experiments, presented in §3.1, uses randomly generated demand for a simple product structure to illustrate substitution behavior under a variety of cost and demand parameters. The second set, presented in §3.2, uses real data from aluminum-tube manufacturing—a context that permits substitutions at the final stage of manufacturing—to validate the RPS model and evaluate its economic impact.

3.1. Simple Random Problems to Study Substitution Behavior

The extent to which production plans exploit substitution options and the consequent savings depend on the cost and demand parameters of the RPS model. In order to identify the most influential among these parameters and understand substitution patterns, we applied the shortest-path algorithm to a class of randomly generated test problems having a simple structure, with two products and two components. Product P1 can use one unit of either Component C1 or Component C2, with C1 being the preferred component; Product P2 requires one unit of Component C2. This restricted setting provides the advantage of limiting the set of parameters to vary while capturing the dominant tradeoffs in the model.

Substitution behavior and effectiveness might be measured both by the *frequency* of substitutions and the cost *savings* compared to ignoring substitution opportunities. We applied the RPS algorithm to over 250 problem instances with different demand patterns and cost values. Results for a selected set of experiments with nonstationary demands are discussed next.

Demand Scenarios. Substitutions are likely to be most attractive when product demands exhibit marked variations over time—during periods of low demand for a product, substitution is more economical than producing and holding that product's preferred component. This type of demand variation can occur in two practical contexts—when a new product is introduced, and when products face seasonal demands, especially if the peak demands for the products are out of phase. Accordingly, we consider two nonstationary demand scenarios, labeled *new product introduction* and *complementary seasonality.*

In the *new product introduction* (NPI) scenario, Product P1 is a new product that will replace Product P2, but P1 can use the older-generation Component C2 in place of its preferred Component C1. Product P1's demand increases gradually over time, as P2's demand declines. In the initial periods when P1's demand is low, using C2 to produce P1 not only eliminates the need to set up and hold C1, but also permits introducing P1 to the market before C1 is fully developed and tested. We do not address here the economic benefits of reducing product introduction time (see, for instance, Ulrich et al. 1993). To simulate the product demands over a 20-period planning horizon, we generate 20 uniformly distributed random numbers between 100 and 400 and sort them in increasing (for P1) or decreasing (for P2) order.

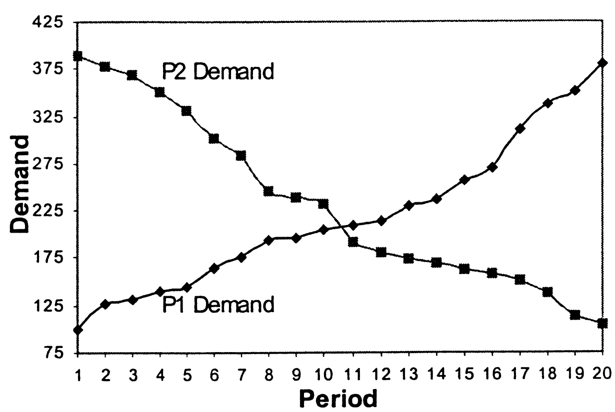
With *complementary seasonality* (CS), both products have seasonal demands, but Product P1's lean season coincides with Product P2's peak season, and vice versa. For our experiments, we divided the 20-period planning horizon into five seasons of four periods each, alternating between seasons of increasing and decreasing demand. Again, product demands were obtained by sorting uniformly distributed random numbers in increasing or decreasing order. Figures 3a and 3b show typical demand patterns for the two demand scenarios.

Cost Assumptions. We assume, without loss of generality, that both components have the same variable production cost (so we can ignore this cost in our computations) but permit different holding costs. The conversion cost c_{21} can capture the difference in the total variable (production + substitution) costs between using Components C2 and C1 in Product P1. Increasing the conversion cost of a substitute component reduces the incentive to substitute, but as setup costs increase relative to holding costs, substitution can become attractive as a means to reduce total production cost via demand pooling. In order to limit the number of cost parameters to vary, we assume that all costs are time-invariant (and so we omit the subscript t when specifying the parameters), and both components have the same setup cost S.

Selected Results. Our initial experiments with many problem instances having extreme (high or low) values for setup, holding, and conversion costs for each demand scenario showed that, when setup costs are high relative to conversion cost, substitutions occur frequently and can provide considerable savings. We then parametrically varied the setup costs for both components, and the holding cost h_2 for the substitute Component C2, keeping the conversion cost c_{21} and the holding cost h_1 for Component C1 fixed at 1. Figures 4 and 5 summarize the results of these tests. The statistics reported in these figures correspond to averages

Figure 3 Nonstationary Product Demand Patterns

(a) New Product Introduction



(b) Complementary Seasonality

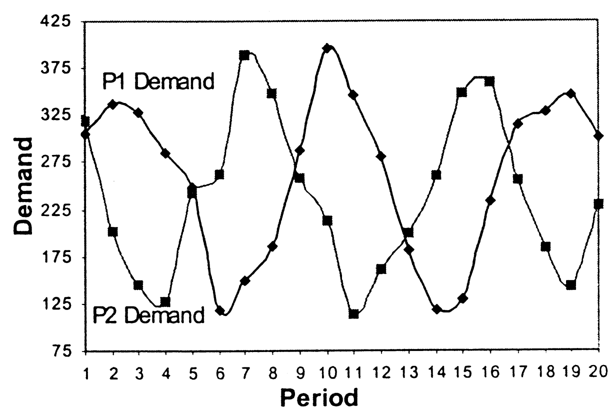
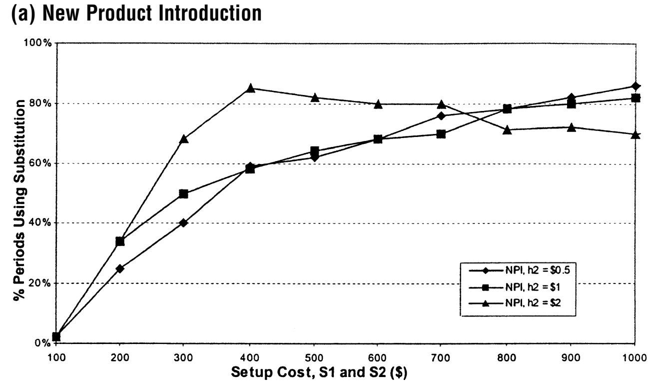


Figure 4 Substitution Frequency for Random Test Problems

(a) New Product Introduction



(b) Complementary Seasonality

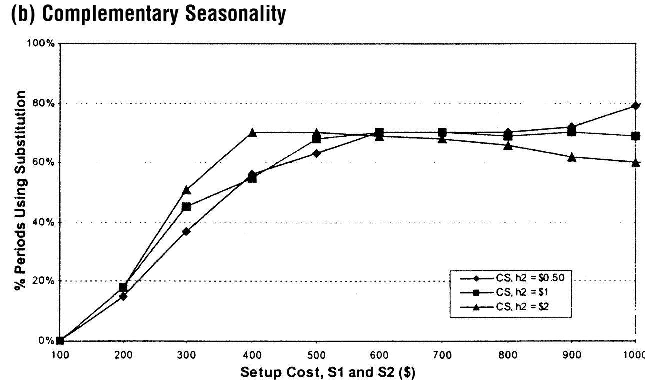
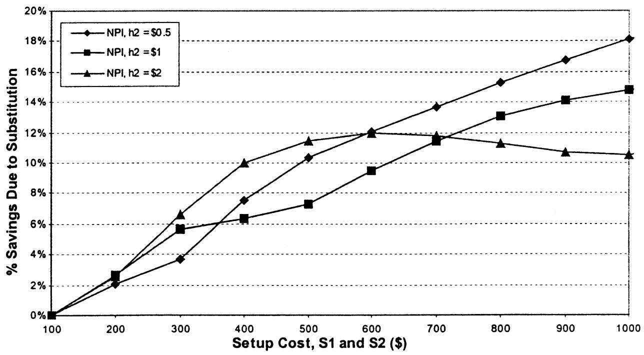
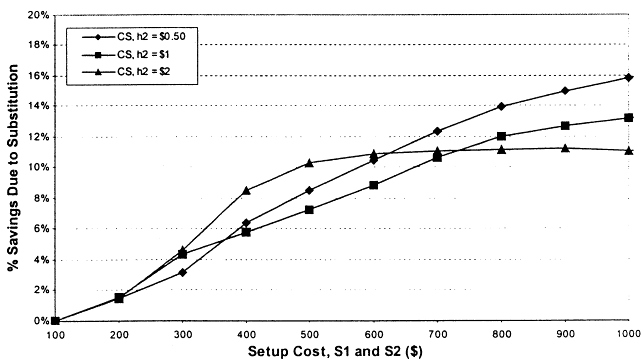


Figure 5 Cost Savings Due to Substitution for Random Test Problems
(a) New Product Introduction



(b) Complementary Seasonality

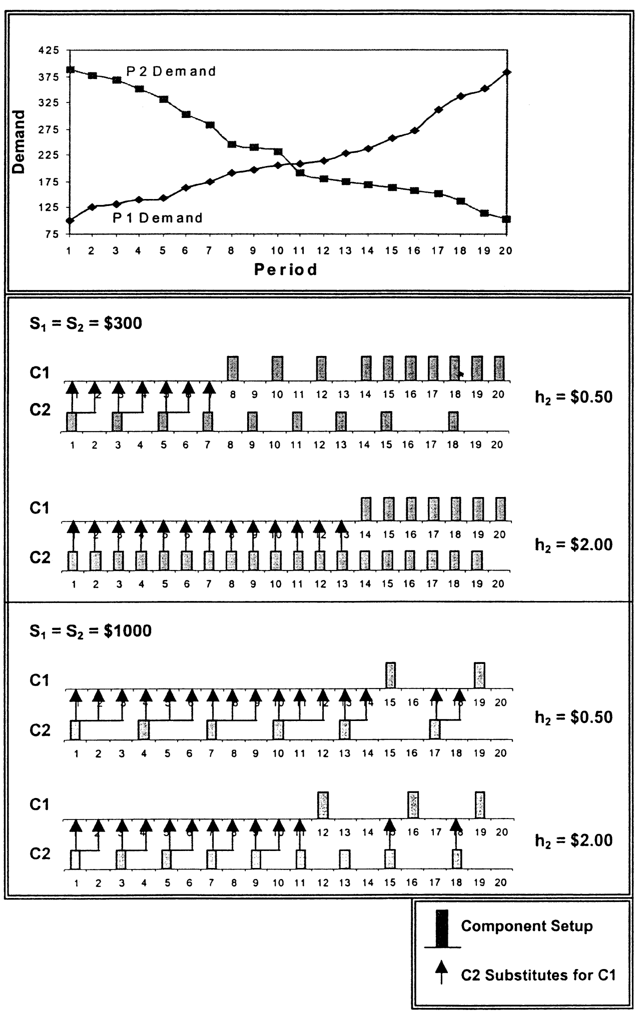


over five demand instances from the NPI and CS demand scenarios. Figure 4 shows how substitution frequency, measured as the percentage of periods in which the optimal solution uses the substitute Component C2 for Product P1, varies as a function of S. Figure 5 illustrates the relationship between setup cost and the savings due to substitutions, expressed as a percentage of the optimal total cost without substitutions. We obtain this latter cost by solving two single-item lot-sizing problems, one for each component, assuming Product P1 always uses Component C1. We note that these savings statistics must be interpreted with caution since they depend heavily on the absolute magnitudes of the cost parameter values we choose.

Interpretation. The charts in Figures 4 and 5 illustrate several interesting phenomena. First, as a function of S, substitution frequency appears to be neither concave nor convex. Indeed, for some values of h_2 (e.g., $h_2 = 2$), substitution frequency is not even monotonically increasing: After an initial increase, it decreases with S in a certain range before increasing again. For other h_2 values, substitution frequency increases at a noticeably slower rate for intermediate values of S. The setup-cost values around which this "oscillating" behavior occurs depends on h_2. These charts indicate a complex interplay between setup costs, holding costs, and demand values. Intuitively, as the setup cost of Component C1 increases, we expect substitutions to increase, whereas increasing C2's setup cost increases the component's average cost and would, therefore, reduce substitutions. When the setup costs for both components increase simultaneously, as in our experiments, each of these effects appear to dominate in turn as S increases. When S is close to zero, setting up both components in every period is economical, and so no substitution occurs. For extremely high setup costs, the solution will set-up only Component C2 and use 100% substitutions. As we vary S from low to high values, initially the solution can sustain frequent setups for Component C2 (while setting up C1 less frequently) by exploiting demand pooling through substitutions; so substitution frequency increases. Beyond a certain threshold value of S (that depends on h_2), producing Component C2 in every period becomes uneconomical. In this case, the need to incur an additional holding cost for C2 makes substitutions unattractive during periods in which C2 is not produced, so the number of periods in which substitutions occur might decrease. Finally, when setup costs rise high enough, substitutions are cheaper than incurring setups for C1, despite the cost of holding C2. Finally, note from Figure 5 that as substitution frequency increases, percentage savings also increases even though the denominator (total cost without substitutions) increases with S.

Second, and more surprising than the previous trend, is the behavior of substitutions as the holding cost h_2 for the substitute Component C2 increases, for a fixed value of S. In particular, for certain ranges of setup-cost values, substitution frequency (and savings) actually *increases* with h_2. Intuitively, we might expect that as the cost of the substitute component increases, substitutions must decrease. Indeed, from Figure 4a we note that when $S \geq 800$ for the NPI scenario this hypothesis holds true, i.e., increasing h_2 decreases substitutions. However, the same figure also shows that for $S = 300$, substitution frequency and % savings are higher for $h_2 = 2$ than for $h_2 = 1$ or 0.5. Figure 6, showing the optimal setup and substitution patterns for one NPI problem instance at two values of S and h_2, helps explain this phenomenon. For $S = 300$, increasing h_2 from 0.5 to 2 increases the setup frequency for Component C2 from every alternate period to every period (in the first 15 periods). The solution exploits these more frequent setups by substituting C2 for C1 in each of the first 13 periods (when demand for P1 is low or moderate) without incurring any holding cost for C2, whereas previously (with $h_2 = 0.5$) producing and holding C1 was cheaper from Period 8 onwards. We note here that setting up and producing C2 in every period became economical not only because

Figure 6 Selected RPS Solutions for NPI Demand Scenario



h_2 increased, but also due to pooling Product P1's demand with P2's demand, i.e., substitutions and setup frequency are interdependent. In contrast, for $S = 1{,}000$, although increasing h_2 from 0.5 to 2.0 again increases the setup frequency for Component C2, the higher holding cost for C2 (compared to the marginal production cost of C1) leads to fewer substitutions.

Figures 4 and 5 also reveal other notable patterns of substitution behavior. For instance, comparing the results for the NPI versus CS scenarios, we observe that for any given value of S, both substitution frequency and savings are higher for the NPI scenario. We attribute this difference to the greater disparity in the magnitudes of the two product's demands over an extended period of time. That is, substitution behavior depends not only on the magnitude of difference in product demands, but also on the duration of time during which this difference in volume continues. Examination of the optimal RPS solutions (as in Figure 6) for various scenarios also showed that substitutions tend to occur in "runs" or contiguous periods—during the early periods for the NPI scenario when the new Product P1 has low demand, and around the demand troughs for Product P1 in the CS scenario.

In summary, substitution behavior is influenced by an intricate combination of factors, making it difficult to predict, especially in settings with nonstationary demand. Our RPS-solution method has provided a way to study this behavior empirically, albeit in a simple setting, permitting us to better understand the impact of various parameters and conditions under which substitutions are likely to be most effective.

3.2. Real-World Application: Aluminum-Tube Manufacturing

Seamless aluminum-drawn tubes have many applications in aircraft, construction, transportation, and other industries. The manufacture of these tubes provides a natural application setting for the single-level RPS model. In this context, finished *tubes* made to customer-specified dimensions are the end products, and aluminum extrusions or *blooms* are the (single) input components used to produce finished tubes. The tube-fabrication process consists of three consecutive stages: ingot casting, extrusion, and drawing. Balakrishnan and Brown (1996) describe this process flow in detail. Because aluminum ingots are typically made to stock or purchased, we focus on production planning for the final two stages of tube manufacturing—bloom extrusion and tube drawing.

A *bloom* is an intermediate-sized hollow cylinder formed by forcing a solid aluminum ingot at elevated temperature through a set of concentric dies on an extrusion press. Extrusion is a batch process; each batch consists of blooms of the same alloy and physical dimensions. Before a batch of blooms can be produced, the extrusion press must be set up by loading the appropriate dies, adjusting the machine settings, and preheating the ingots. The actual processing time to extrude a bloom varies approximately inversely with the cross-sectional area (CSA) of the bloom. Because extrusion tooling is expensive, blooms are only produced in certain standard sizes, specified by their outer diameter (OD) and wall thickness (WT).

Customers order finished *tubes* in a variety of sizes (OD, WT, length) and material specifications (e.g., alloy, temper). Producing a finished tube from a bloom entails reducing both the OD and WT (while elongating the workpiece) by repeatedly "drawing" the bloom (i.e., pulling it through a die) at room temperature, using a draw bench. This cold-forming process permits precise control of dimensions, and also is necessary to meet strength and hardness specifications. The tube-drawing process is very flexible, i.e., it can produce the same finished size from several alternate bloom sizes (having larger OD and WT than the finished size, but subject to some upper limits). However, process constraints limit the amount of CSA reduction achievable in each drawing pass (Balakrishnan and Brown 1996). Hence, reducing a larger bloom to a given finished tube size requires more drawing passes. Moreover, if the required bloom-to-tube CSA reduction exceeds certain thresholds, intermediate annealing steps may also be necessary.

As this discussion suggests, production planners in the tube-manufacturing facility have considerable leeway in deciding which bloom to use for each ordered batch of tubes, i.e., the BOM is flexible because the tube-drawing process is flexible. Since the lead time for tube drawing is less than delivery lead time, planners can incorporate actual demand information for, say, the next four weeks in planning bloom-extrusion operations. Thus, the single-level RPS model is quite well-suited for this context. Current practice does not systematically exploit the available BOM flexibility. Planners make ad hoc decisions regarding how much of each bloom to produce and what bloom to use for each tube order, largely based on past history. Our goal is to estimate the savings possible from optimal planning and coordination of the two stages.

In the RPS terminology, each standard bloom size is a component type and each finished tube size is a product or end-item. The setup and variable costs of extruding blooms represent the component production costs. The conversion cost to use component j for product i is the tube-drawing cost to reduce the bloom size j to finished tube size i. Given the set of standard bloom sizes and the demand for each finished tube size over the planning horizon, the RPS model determines the bloom production plan and bloom-to-tube assignments that minimize the total extrusion setup and production cost, bloom holding cost, and incremental bloom-to-tube conversion cost (relative to using the preferred bloom).

To test the RPS model for this application context, we used demand and process data from an aluminum-tube manufacturer. We focused on one of the plant's most important product families, consisting of finished tubes of a particular alloy with OD values in a certain range. The data covered 24 weeks of demand. Eight standard bloom sizes accounted for 84% of total bloom production (by weight) for this product family. During the 24 weeks, 95% (by weight) of this bloom production was used to produce 109 different finished tube sizes. Hence, our test problems contained $m = 8$ components and $n = 109$ products. We divided the 24 weeks into six demand sets of four periods each, giving us six test problems each with a four-week planning horizon. To ensure consistency in units of measurement at the extrusion and tube-drawing stages, we translated the weekly demand for each product from number of pieces ordered to total pounds ordered (which depends on the number of pieces, tube dimensions, and material density).

Data preparation consisted of: (i) determining the component production cost parameters by estimating the setup and extrusion time for each bloom size; (ii) developing the flexible BOM for each tube, i.e., identifying which among the eight blooms can produce each product; and (iii), computing the cost to convert each feasible bloom to the finished tube size. Bloom production costs were assumed proportional to the time required to set up and produce a batch of blooms on an extrusion press, and conversion costs proportional to the draw bench time needed to reduce the bloom to the finished size.

Bloom Production Effort. Bloom production costs consist of the fixed cost to set up the extrusion press for each batch of blooms, and variable (volume-dependent) costs representing extrusion press usage to actually produce the blooms from ingots. Typical extrusion press *setup times* are around 45 minutes; we assume setup times are the same for all bloom sizes. To compute bloom production effort and hence the *variable costs,* we used past data on actual extrusion time to statistically estimate the extrusion speed, expressed in pounds of bloom extruded per minute, as a function of bloom OD and WT (see Balakrishnan et al. 1995). This model confirmed that blooms with larger cross-sectional areas have higher extrusion speed. The reciprocal of extrusion speed gives the extrusion press time needed per pound of production for each bloom size.

Substitution Options and Conversion Effort. For each finished tube size, we applied the process constraints in use at the tube plant to determine which among the eight blooms could produce that tube. Developing this flexible BOM entailed constructing the complete process plan (e.g., deciding the number of drawing passes, the percentage reduction per draw, the intermediate workpiece sizes, and so on) for every feasible bloom-to-tube assignment. These plans were based on the actual processing rules used by the planners, including limits on the maximum CSA reduction per draw, drawing angle, bulb clearance, and so on (see Balakrishnan and Brown 1996). Out of the 872 total possibilities, 818 bloom-to-tube assignments were feasible. As a by-product of the process planning exercise, we were able to quantify the tube-drawing effort, expressed as total minutes of draw-bench time needed per tube, for each feasible assignment. Draw-bench time consists of the time to load each tube onto the machine, plus the actual drawing time.

Managers felt that our extrusion and tube-drawing "effort" models were realistic, and the processing times that these models predicted were reasonable. However, we were not able to obtain precise estimates for the cost per hour of extrusion and draw-bench time, which were needed to convert the effort calculations into monetary values. In part, this uncertainty in cost estimates stemmed from inadequate accounting systems, differences over what "relevant" costs to include (e.g., costs of direct and indirect labor, consumables, equipment depreciation, opportunity costs, etc.), and variation in these costs among different facilities. For our *base case,* we valued extrusion time at \$1,000 per hour, but we also tested higher and lower values. Managers generally felt that since the extrusion press is much more capital intensive than a draw bench and requires more operators, extrusion time is three to four times as valuable as draw-bench time. The higher this ratio, the greater the conversion cost. Our base case assumes a 3-to-1 ratio of extrusion press cost per hour to draw-bench cost per hour, and 45 minutes setup time per batch of blooms. Again, we performed sensitivity analysis with respect to these parameters.

To determine bloom holding costs, we computed the value per pound of each bloom size by adding the raw material costs (which typically account for over 50% of total bloom costs) to the (variable) cost of extruding the bloom. Raw material costs were based on the current per pound value of aluminum alloy on the London Metal Exchange. Bloom holding costs were obtained by applying a 20% annual holding rate to the total bloom cost.

We define the *preferred* component type for a product as the bloom that requires the lowest variable production (extrusion) plus conversion (tube-drawing) cost to produce the corresponding tube. An alternative definition of preferred bloom (one that tube production planners might prefer, and that appeared to more closely approximate actual practice) selects the bloom that requires the lowest conversion cost. As we discuss later, this latter definition considerably increases the cost benchmark (total system cost without substitutions) that we use to compute savings due to substitutions. So, our definition of preferred bloom provides a conservative estimate of savings.

In summary, using the setup and processing times at the extrusion press as the drivers for component production costs, and tube-drawing time as the driver for conversion cost, we determined the flexible BOM and computed the necessary cost parameters for the RPS model. Our data set provides six 4-week demand scenarios involving eight component types and 109 products. For each demand scenario, we considered three values of extrusion-press setup time per batch (30, 45, and 60 minutes), three values of extrusion-press cost per hour ($500, $1,000, and $1,500), and two values of the ratio of extrusion-press cost to draw-bench cost (3-to-1 and 4-to-1), resulting in 108 problem instances.

Solving the RPS Model. For this application, the shortest-path network G for the RPS-solution procedure contains nearly 400,000 nodes and over 18 million arcs. To solve these large problems, we implemented Pape's (1980) shortest-path algorithm which uses a modified label-correcting scheme (see Ahuja et al. 1993) by maintaining a double-ended queue to keep track of nodes whose successors are candidates for label updates. At each step, instead of searching for the minimum temporary label (as in other label-correcting schemes such as Dijkstra's algorithm), this method gives priority to nodes whose labels have changed most recently. Although Pape's algorithm is only pseudopolynomial (in the worst case), empirical studies have shown that it examines fewer nodes than most label-correcting schemes, and is very efficient in practice (Ahuja et al. 1993). Our experience confirms these prior observations.

Our overall implementation of the RPS solution procedure, programmed in C to run on a personal computer, consists of problem preprocessing, network generation, and shortest-path optimization. Before creating the shortest-path network G, we computed the optimal total cost without using substitutions, obtained by solving single-item lot-sizing problems for each of the eight blooms, assuming that every tube is assigned to its preferred bloom. This cost is an upper bound on the optimal value of the RPS problem; hence, any arc whose length exceeds this value cannot belong to the shortest path. This preprocessing step allowed us to eliminate numerous arcs from the network (on the order of several hundred thousand). The network G was generated and stored using a forward star representation (Ahuja et al. 1993). Because the cost of an arc depends only on the setup vector corresponding to its start node and the layer (period) containing the arc's end node, we only need to calculate the cost to go from each node to every higher layer.

Results. Shortest-path solution times for our 390,626-node problems averaged approximately 9 minutes on a personal computer with a Pentium II 400 MHz processor and 64 MB of memory. Table 1 presents the results of our computations. The table reports averages, over the six demand sets, of three performance measures—savings, number of blooms used, and substitution frequency—for different values of extrusion setup time and extrusion cost per hour, assuming a 3-to-1 ratio of extrusion cost per hour to drawing cost per hour. The *percentage cost savings* due to substitutions is the difference between the optimal cost without substitutions and the optimal value of the RPS problem, expressed as a percentage of the former cost. To avoid overstating the percentage savings due to substitution, we include the tube-drawing cost in the total cost. Recall that, in general, the RPS objective

Table 1 Computational Results for Aluminum-Tube Planning Problems

Extrusion Setup Time (min.)	Extrusion Cost ($/hr)	Percentage Cost Savings[2] Avg.[1]	Min.	Max.	Number of Blooms Used[1]	Substitution Rate[1,3]
	500	**7.74%**	3.21%	13.82%	**4**	**75.7%**
30	1000	**7.75%**	2.78%	14.54%	**3.5**	**78.0%**
	1500	**7.64%**	2.78%	14.24%	**3.5**	**76.8%**
	500	**8.70%**	4.73%	17.56%	**3.33**	**75.6%**
45	1000	**8.73%**	4.32%	18.27%	**3.33**	**76.8%**
	1500	**8.60%**	4.32%	17.99%	**3.33**	**76.4%**
	500	**9.63%**	4.95%	20.92%	**3**	**76.8%**
60	1000	**9.65%**	5.00%	21.61%	**3**	**76.9%**
	1500	**9.50%**	4.8%	21.33%	**3**	**76.6%**

[1]Average values over 6 demand sets.

[2]Savings in total cost due to substitutions as a percentage of total cost without substitutions.

[3]Percentage of products that use a substitute bloom in any period with nonzero demand.

function only requires the incremental bloom-to-tube conversion cost (relative to producing each tube from its preferred bloom). By adding the cost of drawing each tube from its preferred bloom, the denominator includes all costs incurred at the extrusion and tube-drawing stages; the $ saving in the numerator remains unaffected by this addition. Table 1 also reports the average *number of bloom sizes* (out of the eight available bloom sizes) that the RPS solution actually used in each planning interval. The *substitution frequency* denotes the percentage of demand occurrences (product-period combinations) for which the RPS solution used a substitute instead of the preferred bloom.

The results in Table 1 show that incorporating substitution options and applying the RPS model can provide significant production-cost savings. On average, the model achieves 8.7% savings in extrusion, holding, and conversion costs for the base case. To test the robustness of this savings and our choice of planning interval, we also determined the optimal cost without substitutions, assuming that the product demands for all 24 weeks are known in advance. The sum of the optimal RPS values for the six 4-week demand sets was still 7.2% lower than this optimal 24-week cost without substitutions, suggesting that limiting the RPS model's planning horizon to four weeks does not significantly erode savings. We also studied the effect of using the alternate definition of preferred bloom for each tube, namely, the bloom that has the lowest conversion cost to produce that tube. For the base case, the percentage savings for the RPS solution over the new optimal total cost without substitutions (assuming each tube is assigned to its new preferred bloom) increased to 10.49%. Note that with this new definition of preferred blooms, the denominator of the savings statistic also increases, but the greater absolute savings more than offset this increase. A similar computation using the blooms that the planners had actually used for each product (instead of the preferred blooms) produced savings of 10.95% due to substitutions. Indeed, using either of these latter benchmarks instead of the preferred bloom as originally defined, produced additional savings of around 2% (above those reported in Table 1) for each of the various parameter combinations reported in Table 1. These results indicate that the proper choice of a preferred component should take into account not only the conversion cost, but also the variable production cost.

From Table 1, we note that the RPS solution typically uses only around three or four blooms in each 4-week period, while the number of preferred blooms needed to meet demand in each planning interval was over five, on average. Thus, with substitutions, the facility can avoid producing at least one bloom size in each planning interval. Table 1 also indicates that the RPS solution uses a high substitution frequency to achieve the savings. Our set of eight blooms contained certain "compromise" blooms that were not preferred for many products, yet could produce many different tubes at a modest (variable + conversion) cost premium. Incorporating the appropriate tradeoffs, the RPS model was able to exploit the availability of these blooms to realize savings in overall costs. Finally, we note the three performance metrics do not vary substantially when we change either the extrusion setup time or the cost per hour parameters. Increasing the setup time does increase the savings from 7.8% to 9.7%, accompanied by a modest increase in the substitution frequency, as anticipated. However, changing the cost per hour does not produce a systematic pattern of behavior. Increasing this cost parameter increases all three types of costs—setup, processing, and conversion costs. The effects of these simultaneous increases are unpredictable because increasing, say, the setup cost tends to increase substitutions, while increasing the conversion cost makes substitution less attractive. The results in Table 1 indicate that these effects seem to cancel out; therefore, not knowing the exact cost per hour does not significantly affect the results. Changing the ratio of extrusion-press cost to draw-bench cost, however, has a clear and significant impact on the results. When this ratio was changed from 3-to-1 to 4-to-1, the percentage cost savings increased by approximately 5% for all of the cases tested, while the substitution frequency increased by 6 to 10%. This result is not surprising, since increasing this cost ratio reduces relative draw-bench costs, and hence conversion costs.

To summarize, aluminum-tube manufacturing serves as a good context to illustrate and apply the RPS model. Our results indicate that the shortest-path solution method is viable for this context, and produces tangible savings.

4. Conclusions

Both continuous and batch-manufacturing contexts provide opportunities to use alternate raw materials and components to produce a set of end products. These opportunities will likely increase as companies adopt modular product designs and expand their product offerings. Exploiting substitution opportunities in medium and short-term production plans can provide significant savings in manufacturing costs. Requirements planning with substitutions is, therefore, an important practical problem. In this paper, we proposed an optimization model and developed a solution procedure to optimally plan production and substitutions for single-level problems, assuming dynamic, deterministic demand. The algorithm exploits certain special properties of optimal RPS solutions, and extends the classical dynamic-programming approach for single-item lot sizing.

We validated the method by applying it to a practical problem in aluminum-tube manufacturing, and also used it to understand substitution behavior using a simple product structure. Our computational experiments demonstrate that incorporating component substitutability at the product-design stage, and exploiting this flexibility during production planning can reduce manufacturing costs even under deterministic demand. Substitutions can be particularly advantageous when products have seasonal or other forms of nonstationary demand, conversion costs are low, and component setup costs are high. Standard MRP systems generally assume a fixed, inflexible bill of materials. The requirements-planning method described in this paper can extend the capabilities of standard MRP systems to exploit substitution options. It can also serve as a tool to estimate the potential benefits of designing for substitutability when a firm is contemplating product design changes. Besides production lot sizing, our model might also apply to other contexts such as purchasing. For instance, if a firm has two qualified vendors who can both supply all the necessary components, albeit at different costs, using a single source during periods of low demand might reduce total procurement costs.

This paper focused on lot sizing and planned substitutions for two consecutive stages of production—the periodic assembly or fabrication of end-items with known demand and the required component production at its immediately preceding stage. As we have seen, this single-level model is itself relevant to manufacturing contexts such as final assembly of computers or production of metal tubes, plates, or sheets. The model formulation readily extends to the multistage setting, as do the ZIP, HPL, and IU properties of optimal RPS solutions. Developing effective solution methods for this setting can further extend the practical applicability of the RPS model. However, the multistage production planning problem is extremely difficult to solve optimally, even without substitutions. The difficulty stems from having to keep track of the timing of several previous setups for each upstream component, unlike the single-level model which only requires information on the most recent setup for each component. For instance, to decide which component to use to meet product i's demand in period t, we need to know the production and holding cost for each of its alternative input components $j \in J(i)$. In turn, these costs depend on which among component j's alternative input parts $j' \in J'(j)$ was used when component j was last produced (say, at period $t' \leq t$), and how long the chosen part j' was held in inventory. So, if we knew the last setup period $t'(j)$ (on or before period t) for every component $j \in J(i)$, and the last setup period $t''(j')$, on or before $t'(j)$, for every part $j' \in J'(j)$, then we can determine the best BOM to use to meet product i's demand in period t. Because each upstream part might be used in multiple products, we see that as we go further upstream we need to keep track of more setup information. Properties such as the IU property can reduce the number of setup combinations to consider (e.g., consider only combinations that permit immediately using each part that is produced by at least one downstream component). Nevertheless, the state space expands dramatically for multilevel problems, making a direct extension of the dynamic-programming approach impractical.

Instead, we might consider ways to exploit our ability to solve single-level problems quickly and optimally by considering decomposition or heuristic approaches to the multilevel problem. For instance, by dualizing the constraints that link the production and inventory decisions at one level to the substitution (and usage) decisions at the next, the multistage RPS model decomposes into several single-level problems. Thus, a Lagrangian-relaxation approach might well prove effective, especially if the formulation is strengthened using some of the properties of optimal RPS solutions. Alternatively, we might consider iterative heuristics that sequentially determine lot-sizing decisions at each level—from level 0 to upstream levels, using approximations for the upstream components' production costs at each iteration, and from the upstream to downstream stages, using approximations to downstream demand at each iteration. Balakrishnan and Geunes (1998) and Geunes (1999) elaborate on these decomposition and heuristic strategies. Developing and testing effective solution methods for requirements planning in multistage, multiproduct systems with substitutions is a promising research direction to pursue in the future.[1]

Appendix A: Rooted-Tree Characterization of Optimal RPS Solutions

PROPOSITION 1: ROOTED TREE SOLUTION *With concave arc costs, the equivalent generalized network flow problem defined on G has an optimal rooted-tree solution, i.e., the set of arcs carrying positive flow in this solution form a directed tree rooted at the supersource.*

PROOF. Suppose we have an optimal flow solution that does not satisfy the rooted-tree property, i.e., at some node l, at least two incoming arcs carry positive flow. For convenience, choose l as the node that is closest to the source among all nodes with two incoming flows. Let ϕ_{ij} denote the current flow on each arc (i, j) of the network. Let $(i1, l)$ and $(i2, l)$ be two of these arcs, and let β_1 and β_2 denote their respective flow weights. If node l is a demand node, then the flow weights are the multipliers α_{il} and α_{jl} on the incoming arcs; otherwise, if node l is an availability node, both weights are one. Let P_1 and P_2 be the two corresponding paths (with $(i1, l)$ and $(i2, l)$ as their terminal arcs) from the source node to node l, and let $f_1, f_2 > 0$ denote the current flow on these paths. By our choice of node l (as the one receiving multiple inputs and is closest to the source), $f_1 = \phi_{i1,l} = \min_{(i,j)\in P_1} \phi_{ij}$ and $f_2 = \phi_{i2,i} = \min_{(i,j)\in P_2} \phi_{ij}$. For $g = 1, 2$, let $\Delta_g^+(\nu)$ denote the total marginal *cost* of sending ν more units of flow to path P_g, which is the sum of the marginal costs of increasing the current flow by ν units on each arc of path P_g. That is, if $\delta_{ij}^+(\nu)$ is the cost of increasing the current flow on arc (i, j) by ν units, then $\Delta_g^+(\nu) = \Sigma_{(i,j)\in P_g} \delta_{ij}^+(\nu)$. Similarly, for $0 \le \nu \le f_g$, let $\Delta_g^-(\nu) = \Sigma_{(i,j)\in P_g} \delta_{ij}^-(\nu)$ denote the marginal *savings* if we withdraw ν units of flow from path P_g, where $\delta_{ij}^-(\nu)$ is the savings if we withdraw ν units of current flow from arc (i, j).

Because the arc costs are concave, $\delta_{ij}^+(\nu)/\nu \le \delta_{ij}^-(\nu')/\nu'$ for any arc (i, j) and $\nu, \nu' > 0$ with $\nu' \le \phi_{ij}$. Summing these inequalities over all arcs belonging to path P_g, we get $\Delta_g^+(\nu)/\nu \le \Delta_g^-(\nu')/\nu'$, i.e., the per-unit savings of withdrawing flow equals or exceeds the per-unit cost of adding flow. Hence,

$$\frac{\Delta_1^+(\beta_2 f_2/\beta_1)}{\beta_2 f_2/\beta_1} \le \frac{\Delta_1^-(f_1)}{f_1} \text{ and } \frac{\Delta_2^+(\beta_1 f_1/\beta_2)}{\beta_1 f_1/\beta_2} \le \frac{\Delta_2^-(f_2)}{f_2}.$$

LEMMA: *Either* $\Delta_1^-(f_1) - \Delta_2^+(\beta_1 f_1/\beta_2) \ge 0$ or $\Delta_2^-(f_2) - \Delta_1^+(\beta_2 f_2/\beta_1) \ge 0$.

PROOF OF LEMMA. Suppose $\Delta_2^-(f_2) - \Delta_1^+(\beta_2 f_2/\beta_1) < 0$. Then, since $\Delta_2^+(\beta_1 f_1/\beta_2) f_2/\beta_1 f_1/\beta_2 \le \Delta_2^-(f_2)$ and $-\Delta_1^-(f_1)\beta_2 f_2/\beta_1/f_1 \le -\Delta_1^+(\beta_2 f_2/\beta_1)$, we have $\Delta_2^+(\beta_1 f_1/\beta_2) f_2/\beta_1 f_1/\beta_2 - \Delta_1^-(f_1)\beta_2 f_2/\beta_1/f_1 < 0$, i.e., $\Delta_1^-(f_1) - \Delta_2^+(\beta_1 f_1/\beta_2) > 0$. Similarly, we can prove that, if $\Delta_1^-(f_1) - \Delta_2^+(\beta_1 f_1/\beta_2) < 0$, then $\Delta_2^-(f_2) - \Delta_1^+(\beta_2 f_2/\beta_1) > 0$. □

Without loss of generality, assume that $\Delta_1^-(f_1) - \Delta_2^+(\beta_1 f_1/\beta_2) \ge 0$. In this case, we can withdraw the f_1 units that currently flow on path P_1, and instead supply an equivalent amount to node l by sending $\beta_1 f_1/\beta_2$ additional units on path P_2. The cost condition $\Delta_1^-(f_1) - \Delta_2^+(\beta_1 f_1/\beta_2) \ge 0$ implies that redirecting the flows in this way does not increase the total cost of the solution. Moreover, this flow-redirecting process has reduced the number of flow-carrying arcs entering node l by one. Repeating this process at all nodes that have multiple incoming arcs with positive flow, we obtain an equal or lower cost flow solution that satisfies the rooted-tree condition. □

[1]We thank two anonymous reviewers and the senior editor for their detailed comments and helpful suggestions that have led to several improvements in the paper.

References

Afentakis, P., B. Gavish, U. Karmarkar. 1984. Computationally efficient optimal solutions to the lot-sizing problem in multistage assembly systems. *Management Sci.* **30** 222–239.

——, ——. 1986. Optimal lot-sizing algorithms for complex product structures. *Oper. Res.* **34** 237–249.

Agrawal, N., S. Smith. 1996. Management of multi-item retail inventory systems with demand substitution. Proceedings of the 1996 MSOM Conference, Dartmouth College, Hanover, NH.

Ahuja, R., T. Magnanti, J. Orlin. 1993. *Network Flows: Theory, Algorithms, and Applications.* Prentice Hall, Englewood Cliffs, NJ.

Baker, K., M. Magazine, H. Nuttle. 1986. The effect of commonality on safety stock in a simple inventory model. *Management Sci.* **32** 982–988.

Balakrishnan, A., S. Brown, D. Dunlap, R. Pahl. 1995. Interdisciplinary industry-university collaboration: Lessons from an operations improvement project. *Interfaces* **25** 12–41.

——, ——. 1996. Process planning for aluminum tubes: An engineering-operations perspective. *Oper. Res.* **44** 7–20.

——, J. Geunes. 1998. Requirements planning for multi-product systems with component substitutability. Working Paper, Penn State University, University Park, PA.

Bassok, Y., R. Anupindi, R. Akella. 1996. Multiproduct inventory models with substitution. Working Paper, Northwestern University, Evanston, IL.

Bitran, G., S. Dasu. 1992. Ordering policies in an environment of stochastic yields and substitutable demands. *Oper. Res.* **40** 999–1017.

——, S. Gilbert. 1994. Co-production processes with random yields in the semiconductor industry. *Oper. Res.* **42** 476–491.

Cornuejols, G., G. Nemhauser, L. Wolsey. 1990. The uncapacitated facility location problem. P. Mirchandani, R. Francis, eds. *Discrete Location Theory.* Wiley & Sons, New York. Chapter 3.

Fine, C., R. Freund. 1990. Optimal investment in product-flexible manufacturing capacity. *Management Sci.* **36** 449–466.

Gerchak, Y., M. Magazine, A. Gamble. 1988. Component commonality with service level requirements. *Management Sci.* **34** 753–760.

Geunes, J. 1999. *Models To Optimize Multi-Stage Linkages In Manufacturing And Distribution Operations.* Ph.D. Thesis, Penn State University, University Park, PA.

Hax, A., H. Meal. 1975. Hierarchical integration of production planning and scheduling. M. Geisler ed. *Studies in Management Sciences. Vol. I, Logistics.* North Holland-American Elsevier, Amsterdam, Holland.

Ignall, E., A. Veinott, Jr. 1969. Optimality of myopic inventory policies for several substitute products. *Management Sci.* **15** 284–304.

Jordan, W., S. Graves. 1995. Principles on the benefits of manufacturing process flexibility. *Management Sci.* **41** 577–594.

Love, S. 1972. A facilities in series inventory model with nested schedules. *Management Sci.* **18** 327–338.

Magnanti, T., R. Wong. 1984. Network design and transportation planning: Models and algorithms. *Transportation Sci.* **18** 1–55.

McGillivray, A., E. Silver. 1978. Some concepts for inventory control under substitutable demand. *INFOR.* **16** 47–63.

Ou, J., L. Wein. 1995. Dynamic scheduling of a production/inventory system with by-products and random yield. *Management Sci.* **41** 1000–1017.

Pape, U. 1980. Shortest path lengths. *ACM Trans. on Math. Software* **3** 450–455.

Pine, B. 1993. *Mass Customization: The New Frontier in Business Competition.* Harvard Business School Press, Boston, MA.

Rosolen, N. 1987. Component part number substitution in MRP. *Prod. and Inv. Management* 3rd Q. **28** 22–26.

Tang, C., E. Denardo. 1988a. Models arising from a flexible manufacturing machine, part I: Minimization of the number of tool switches. *Oper. Res.* **36** 767–777.

——, ——. 1988b. Models arising from a flexible manufacturing machine, part II: Minimization of the number of switching instants. *Oper. Res.* **36** 778–784.

Ulrich, K., D. Sartorius, S. Pearson, M. Jakiela. 1993. Including the value of time in design-for-manufacturing decision making. *Management Sci.* **39** 429–447.

Wagner, H., T. Whitin. 1958. Dynamic version of the economic lot size model. *Management Sci.* **5** 89–96.

The consulting Senior Editor for this manuscript was Ken Baker. This manuscript was received January 1, 1998, and was with the authors 11 1/2 months for 3 revisions. The average review cycle time was 56 days.