EFFICIENT ALGORITHMS FOR FINDING OPTIMAL POWER-OF-TWO POLICIES FOR PRODUCTION/DISTRIBUTION SYSTEMS WITH GENERAL JOINT SETUP COSTS

AWI FEDERGRUEN
Columbia University, New York, New York

YU-SHENG ZHENG
University of Pennsylvania, Philadelphia, Pennsylvania

(Received July 1991; revisions received September 1992, June 1993; accepted December 1993)

We consider a production/distribution system represented by a general directed acyclic network. Each node is associated with a specific "product" at a given location and/or production stage. An arc (i, j) indicates that item i is used to "produce" item j. External demands may occur at any of the network's nodes. These demands occur continuously at item-specific constant rates. Components may be assembled in any given proportions. The cost structure consists of inventory carrying, variable, and fixed production/distribution costs. The latter depend, at any given replenishment epoch, on the specific set of items being replenished, according to an arbitrary set function merely assumed to be monotone and submodular. It has been shown that a simply structured, so-called power-of-two policy is guaranteed to come within 2% of a lower bound for the minimum cost. In this paper, we derive efficient algorithms for the computation of an optimal power-of-two policy, possibly in combination with this lower bound. These consist of a limited number of polymatroidal maximum flow calculations in networks closely associated with the original production/distribution network.

Work by Maxwell and Muckstadt (1985), Queyranne (1985), Roundy (1986), and Federgruen, Queyranne and Zheng (1992) addressed the problem of determining replenishment strategies for a general production/distribution system represented by a general directed acyclic network. Each node is associated with a specific "product" at a given location and/or production stage. External demands may occur continuously at any node and at item-specific constant rates. An arc (i, j) indicates that product i is used to produce product j. Components may be assembled in any given proportions. Orders are delivered instantaneously and no backlogging is allowed. The cost structure consists of inventory carrying, variable, and fixed production/distribution costs. The above network representation underlies many popular commercial planning systems, in particular, Material Requirements Planning (MRP) and Distribution Requirements Planning (DRP) systems.

Federgruen, Queyranne and Zheng showed that when minimizing long-run average costs, a simply structured, so-called *power-of-two* policy is guaranteed to come within 2% of being optimal for the general network model with standard inventory carrying and variable replenishment costs and a general joint setup cost structure assumed to be monotone and reflecting economies of scale in the sense of submodularity. (Submodularity means that the marginal increase of the setup cost due to the addition of a new product to a given collection of jointly replenished items, is no larger than if the same product were added to a subset of this collection.) This result generalizes those discovered by Roundy's pioneering papers (1985, 1986) for the case of *separable* setup costs or joint setup costs specified by a "family model" (see Section 4).

This paper develops solution procedures for the general model discussed in Federgruen, Queyranne and Zheng. We present efficient algorithms for finding optimal power-of-two policies. These consist of a limited number of maximum flow computations in subgraphs of a so-called route-product network whose topology is induced by the topology of the product network and with capacity constraints imposed only on the arcs emanating from the source and those terminating in the sink. The capacities on the former set of arcs are determined by the demand and inventory cost parameters. The capacity constraints with respect to the latter reflect the setup cost structure. If the latter is inseparable, upper bounds prevail on the total flow in *all* subsets of the arcs that point to the sink node. These upper bounds constitute a submodular set function closely related to the setup cost structure.

An earlier approach for the general model considered here is due to Queyranne. The approach uses Maxwell and Muckstadt's divide-and-conquer method and requires up to $(2N - 1)$ calls to an expensive oracle for minimizing a general submodular set function (e.g., by the ellipsoid method, see Grötschel, Lovasz and Schryver 1981). Our algorithms consist of a limited number

Subject classifications: Inventory/production: Multiater echelon, stage. Inventory/production: approximations and heuristics.
Area of review: Manufacturing, Operations and Scheduling.

0030-364X/95/4303-0458 $01.25

of maximum flow computations, and the only oracle required is to check whether a given vector of setup costs is a proper allocation of the joint cost structure (i.e., is a member of the so-called setup cost polyhedron defined below). In contrast with Queyranne's approach, this oracle can be tailored to the specific submodular setup cost function being used (as discussed in Section 4).

The algorithms presented here may be viewed as generalizations of those developed for joint replenishment models with joint setup costs and general networks with a separable cost structure (see Zheng 1987 and Federgruen and Zheng 1992, 1993) which, in turn, are related to Maxwell and Muckstadt's *divide and conquer* algorithm. Compared to the separable cost case, a moderate increase in complexity is required for the most general model, while, in some important special cases, the *same order* of complexity is maintained. See Muckstadt and Roundy (1993) for a survey of numerous papers on algorithms for special network topologies and cost structures.

The remainder of this paper is organized as follows. Section 1 specifies the model and a number of preliminary results. Section 2 develops two general algorithms, with specific advantages for each. The general complexity of these algorithms, specified as an upper bound on the number of elementary operations and calls to the above mentioned oracle (or a related oracle), is discussed in Section 3. Section 4 discusses the computational complexity for specific types of cost structures.

1. THE MODEL

Let the production/distribution system be represented by a general directed acyclic network (N, A), with node set N and arc set A. We use N and A both to represent the node and arc set as well as their cardinalities. For each node $i \in N$, let $P(i)$ denote the set of its immediate predecessors in the network, i.e., $P(i) = \{l \in N: (l, i) \in A\}$. External demands occur at node $i \in N$ at constant rate d_i'; for any arc $(i, j) \in A$, λ_{ij} represents the number of units of product i required to produce one unit of product j. Orders are delivered instantaneously. Variable order costs are proportional to order volumes. Let h_i' denote the cost per unit of time for carrying one unit of product i in inventory. The incremental holding cost rate for product i is given by $h_i = h_i' - \sum_{j \in P(i)} \lambda_{ji} h_j'$ and is assumed to be nonnegative.

The general joint setup cost structure is represented by a general set function K: $2^N \to R^+$ which specifies for any subset of products $S \subset N$ a setup cost $K(S)$ to be incurred whenever this specific collection of "products" is replenished together. The function $K(\cdot)$ is assumed to satisfy these structural properties:

i. (**Monotonicity**) $K(S) \leqslant K(T)$ if $S \subset T$;
ii. (**Submodularity**) $K(S \cup \{i\}) - K(S) \geqslant K(T \cup \{i\}) - K(T)$ if $S \subset T$, $i \notin T$;
iii. (**Nontriviality**) $K(\{i\}) > 0$, $i \in N$.

A power-of-two policy replenishes a product only when its inventory is down to zero and prescribes for each product i a constant replenishment interval t_i, such that a replenishment for this product occurs at times 0, t_i, $2t_i$, $3t_i$, ... Moreover, all product replenishment intervals are chosen as power-of-two multiples of a common base planning period T_L. A route is any directed path in G ending at a product with external demand. A route is specified by a sequence of nodes $r = (i_1, i_2, \ldots, i_m)$. Let R be the collection of all routes in G. (As with N and A, we use R both to denote the set and its cardinality.) For any $r \in R$, we say that product $i \in r$, if route r passes through node i. For each route $(i_1, \ldots, i_m) \in R$, let $H_r = \frac{1}{2} h_{i_1} d_r$, where $d_r = \lambda_{i_1 i_2}, \lambda_{i_2 i_3}, \ldots, \lambda_{i_{m-1} i_m} d_{i_m}$ is the rate at which units of product i_1 are requested to follow route r. Let

$$\mathbf{K} = \left\{\mathbf{k} \in R^N: \sum_{i \in S} k_i \leqslant K(S) \text{ for all } S \subseteq N;\ \mathbf{k} \geqslant 0\right\}.$$

We note that because the set function $K(\cdot)$ is monotone and submodular the polyhedron $\mathbf{K}$ is a so-called *polymatroid* which we refer to as the *setup cost polyhedron*. The long-run average cost of any given power-of-two policy $\mathbf{t}$ is given by

$$c(\mathbf{t}) = \max_{\mathbf{k} \in \mathbf{K}} \sum_{i \in N} k_i / t_i + \sum_{r \in R} H_r \max_{i \in r} t_i, \tag{1}$$

where the first term is the total setup cost (see Federgruen and Zheng 1992) and the second term is the total holding cost (see Roundy 1986).

Definition 1. For any partition $\{N_l, l = 1, \ldots, M\}$ of N, let $E_l \equiv \cup_{j=1}^{l} N_j$; $R_l \equiv \{r \in R: r \subseteq E_l, r \not\subseteq E_{l-1}\}$. For $S \subseteq N_l$, let:

$$H_l(S) \equiv \sum \{H_r: r \subseteq E_{l-1} \cup S,\ r \not\subseteq E_{l-1}\};$$
$$K_l(S) \equiv K(E_{l-1} \cup S) - K(E_{l-1}).$$

Lemma 1. *For any power-of-two vector $\mathbf{t} = (t_i, i \in N)$, suppose that $\{N_l, l = 1, \ldots, M\}$ is a partition of N and $t(1) \leqslant t(2) \leqslant t(M)$ a set of values such that $t_i = t(l)$ for all $i \in N_l$ and all $l = 1, \ldots, M$. Then,*

$$c(\mathbf{t}) = \sum_{l=1}^{M} C_l(t(l)),$$

where $C_l(t) = K_l(N_l)/t + H_l(N_l)t$.

Proof. The fact that the total setup cost can be written as $\sum_{l=1}^{M} K_l(N_l)$ was shown in Lemma A2.3 in Federgruen and Zheng (1992); that the total holding cost is $\sum_{l=1}^{M} H_l(N_l)T(l)$ is due to:

$$\sum_{r \in R} H_r \max_{i \in r} t_i = \sum_{l=1}^{M} \sum_{\substack{r \subseteq E_l \\ r \not\subseteq E_{l-1}}} H_r \max_{i \in r} t_i$$
$$= \sum_{l=1}^{M} \sum_{\substack{r \subseteq E_l \\ r \not\subseteq E_{l-1}}} H_r t(l) = \sum_{l=1}^{M} H_l(N_l) t(l),$$

where the second equality follows again from $t(1) \leq \cdots \leq t(M)$.

Let $\mathbf{T} \stackrel{\text{def}}{=} \{\mathbf{t} > 0: t_i = 2^{m_i} T_L,\ m_i \text{ integer}\}$ denote the set of all power-of-two vectors. (T_L is sometimes predetermined, but may be varied continuously in other settings.) The problem of determining an optimal power-of-two policy can thus be written as follows.

Problem P

$$c^* = \min_{\mathbf{t}\in\mathbf{T}} c(\mathbf{t})$$

and its continuous relaxation is as follows.

Problem RP

$$c_* = \min_{\mathbf{t}>0} c(\mathbf{t}).$$

It follows from (1) that

$$\begin{aligned} c_* &= \min_{\mathbf{t}>0} \max_{\mathbf{k}\in\mathbf{K}} \Big\{ \sum_{i\in N} k_i/t_i + \sum_{r\in R} H_r \max_{i\in r} t_i \Big\} \\ &\geq \max_{\mathbf{k}\in\mathbf{K}} \min_{\mathbf{t}>0} \Big\{ \sum_{i\in N} k_i/t_i + \sum_{r\in R} H_r \max_{i\in r} t_i \Big\} \stackrel{\text{def}}{=} c_D. \end{aligned} \tag{2}$$

(The inequality is easy to verify and applies to *any* min-max problem; it is referred to as weak duality.) We refer to the problem to the right of the inequality as the *dual* problem.

Federgruen, Queyranne and Zheng (1992, Lemma 3) show that this *dual* problem may be written in the following form.

Problem D

$$c_D = D(\mathbf{k}, \mathbf{x}, \mathbf{v}) = \max \sum_i 2(k_i v_i)^{1/2} \tag{3}$$

subject to

$$\mathbf{k} \in \mathbf{K} \tag{4}$$

$$\sum_{i\in N} x_{ri} = H_r \tag{5}$$

$$\sum_{r\in R} x_{ri} = v_i \tag{6}$$

$$\mathbf{x} \geq 0 \quad v \geq 0. \tag{7}$$

The dual program **D** also suggests that **x** can be viewed as a flow in a bipartite network with node set $\mathbf{N} = R \cup N$ and arc set $\mathbf{A} = \{(r, i): r \in R, i \in r\}$. Such a network was introduced in Federgruen and Zheng (1990), where it is referred to as the route-product network. The latter paper shows, for the *separable* cost case, that the relaxed problem **D** can be solved via a limited number of maximum flow computations in subnetworks of the route-product network with arc capacities specified by the demand and cost parameters. We show for the general model treated here that **RP** continues to be solvable via a limited number of maximum flow computations in similar networks. The essential difference is that to represent the joint-cost structure, *joint* capacity bounds need to be imposed on collections of arcs rather than on individual arcs only. Networks with such (monotone, submodular) upper bounds on *sets* of arcs that emanate from or point to a common node are referred to as polymatroidal network flow models.

We first define a class of networks with polymatroidal capacity constraints.

Definition 2. Let $\{N_l: l = 1, 2, \ldots, M\}$ be a partition of N and $\{R_l: l = 1, \ldots, M\}$ be the corresponding partition of R; see Definition 1. For any $\nu > 0$, the associated network $G(\nu, N_l)$ is the bipartite network with node set $\{s\} \cup R_l \cup N_l \cup \{t\}$ and arc set

$$A_l \stackrel{\text{def}}{=} \{(s, r): r \in R_l\} \cup \{(r, i): r \in R_l,\ i \in N_l\} \cup \{(i, t): i \in N_l\}.$$

Joint capacity constraints are imposed with respect to the arcs $\{(i, t): i \in N_l\}$ only; these are specified by the submodular and monotone set function $K_l(\cdot)/\nu$. Each arc $(s, r)(r \in R_l)$ has capacity H_r; all other arcs have *infinite* capacity.

Note that the above defined flow network $G(\nu, N_l)$ is a special case of the so-called polymatroidal networks introduced by Lawler and Martel (1982) and Hassin (1982). Many fundamental properties of ordinary flow networks carry over to these polymatroidal networks. Our derivation below is based on the well-known max flow-min cut theorem.

Let (R_l^-, R_l^+) and (N_l^-, N_l^+) be partitions of R_l and N_l, respectively. We define the capacity of a cut (S, T) with $S = \{s\} \cup R_l^- \cup N_l^-$ and $T = R_l^+ \cup N_l^+ \cup \{t\}$ by

$$C^*(R_l^- \cup N_l^-, R_l^+ \cup N_l^+) = \begin{cases} \sum_{r\in R_l^+} H_r + K_l(N_l^-)/\nu, & \text{if there are arcs } (r, i) \text{ with } r \in R_l^-, i \in N_l^+ \\ \infty, & \text{otherwise.} \end{cases} \tag{8}$$

Let

$$C(N_l^-, N_l^+) \stackrel{\text{def}}{=} \min\{C^*(R_l^- \cup N_l^-, R_l^+ \cup N_l^+): R_l^-, R_l^+ \text{ is a partition of } R_l\}.$$

It clearly follows from (8) that the above minimum is achieved by $R_l^- = \{r \subseteq E_{l-1} \cup N_l^-,\ r \not\subseteq E_{l-1}\}$ and $R_l^+ = \{r \subseteq E_l,\ r \not\subseteq E_{l-1} \cup N_l^-\}$, and hence

$$C(N_l^-, N_l^+) = H_l(N_l) - H_l(N_l^-) + K_l(N_l^-)/\nu. \tag{9}$$

A node partition (N_l^-, N_l^+) is said to be a min-cut of $G(\nu, N_l)$ if $C(N_l^-, N_l^+) \leq C(N_l^{--}, N_l^{++})$ for all partitions (N_l^{--}, N_l^{++}) of N_l.

A flow from s to t in $G(\nu, N_l)$ is the sum of the total outflow from the source node s, or the total inflow to the sink node t.

Lemma 2. $\text{Min}_{N_l^-, N_l^+} C(N_l^-, N_l^+) =$ *Max flow.*

Proof. This identity follows from the fact that $\text{Min}_{N_l^-, N_l^+} C(N_l^-, N_l^+)$ equals the minimum capacity of any arc partition cut (see Lawler and Martel, or Zheng for details).

2. GENERAL SOLUTION METHODS

In this section we describe two solution methods. The first is a *two-stage method*, which first determines an optimal solution to **RP** and then transforms this solution by simple rounding to an optimal solution of **P**. Our second algorithm generates an optimal solution of **P** *directly*; its complexity is $0(N)$ smaller than that of the two-stage method. The advantage of the former is that the rounding procedure, similar to those used in Maxwell and Muckstadt and Roundy, can be designed to generate a power-of-two policy whose cost comes within 2% of being optimal, as opposed to a 6% bound for the policy generated by the *direct* method. A second advantage is the fact that c_*, the optimum value of **RP** obtained as an intermediate result, is a *lower bound* for the minimum cost (see Federgruen, Queyranne and Zheng).

We first derive a characterization theorem that establishes necessary and sufficient conditions for an optimal solution of the relaxed problem **RP**. The proof applies the max-flow min-cut theorem to the polymatroidal networks $G(\nu, N_l)$ defined in the previous section.

Theorem 1. (Characterization theorem for **RP**) *For* **t** > 0, *with component values* $t(1) \le t(2) \le \cdots \le t(M)$, *let* $\{N_l, l = 1, \ldots, M\}$ *be a partition of* N, *such that* $N_l = \{i \in N\colon t_i = t(l)\}(l = 1, \ldots, M)$; **t** *is an optimal solution of* **RP** *if the following conditions are satisfied:* For $l = 1, \ldots, M$:

i. $t(l) = [K_l(N_l)/H_l(N_l)]^{1/2}$

ii. $K_l(S)/H_l(S) \ge t^2(l)$ for all $S \subseteq N_l$.

Also, if conditions i *and* ii *are satisfied, then* $c^* = c_D$.

Proof. To prove the sufficiency part, we have by Lemma 1, that

$$c(\mathbf{t}) = \sum_{l=1}^{M} [K_l(N_l)/t(l) + H(N_l)t(l)] = \sum_{l=1}^{M} 2(K_l(N_l)H(N_l))^{1/2},$$

where the second equality is due to condition i. In view of (2) it suffices to construct a triple $(\mathbf{k}^*, \mathbf{x}^*, \mathbf{v}^*)$, which is feasible for **D** and with

$$D(\mathbf{k}^*, \mathbf{x}^*, \mathbf{v}^*) = \sum_{l=1}^{M} 2(K_l(N_l)H(N_l))^{1/2}.$$

For each $l = 1, 2, \ldots, M$ consider the associated polymatroidal network $G(t^2(l), N_l)$; condition ii implies that for any $S \subseteq N_l$, $K_l(S)/t^2(l) - H_l(S) \ge 0$ or

$$K_l(S)/t^2(l) + [H_l(N_l) - H_l(S)] \ge H_l(N_l) = K_l(N_l)/t^2(l)$$

that is, (N_l, ϕ) is a minimum cut in the associated network $G(t^2(l), N_l)$. By Lemma 2 there exists a maximum flow $(\mathbf{x}^l, \mathbf{v}^l)$ in the network with

$$\sum_{i \in N_l} v_i = K_l(N_l)/t(l)^2, \quad l = 1, \ldots, M. \tag{10}$$

For all $l = 1, \ldots, M$ let

$$\mathbf{k}^l = t^2(l)\mathbf{v}^l. \tag{11}$$

Let $\mathbf{k}^* = (\mathbf{k}^1, \ldots, \mathbf{k}^M)$, $\mathbf{v}^* = (\mathbf{v}^1, \ldots, \mathbf{v}^M)$, and construct $\mathbf{x}^*$ as follows: For all $l = 1, \ldots, M$ and $i \in N_l$ let $x^*_{ri} = x^l_{ri}$, if $r \in R_l$ and $x^*_{ri} = 0$ otherwise. Since the pair $(\mathbf{x}^l, \mathbf{v}^l)$ is a feasible flow in the polymatroidal network $G(t^2(l), N_l)$ $(l = 1, 2, \ldots, M)$, it follows that $(\mathbf{x}^*, \mathbf{v}^*)$ satisfies (3)–(5). Also, for all $l = 1, \ldots, M$ and $S \subseteq N_l$, $v^l(S) \le K_l(S)/t^2(l)$, i.e., $k^*(S) \le K_l(S)$. It follows from Lemma 1 in Federgruen and Zheng (1992) that $k^*(S) \le K(S)$ for all $S \subseteq N$, i.e., $k^* \in K$. Finally,

$$\sum_{i \in N} 2(k_i^* v_i^*)^{1/2} = \sum_{l=1}^{M} \sum_{i \in N_l} 2(k_i^* v_i^*)^{1/2} = \sum_{l=1}^{M} 2t(l) \sum_{i \in N_l} v_i^* = \sum_{l=1}^{M} 2K_l(N_l)/t(l) = \sum_{l=1}^{M} 2(K_l(N_l)H_l(N_l))^{1/2}$$

where the second and the third equality follows from (10) and (11) and the last equality follows from condition i. Since the necessity part of this theorem is not directly used for the development of our algorithms, we refer the interested reader to Zheng for a proof.

For a given partition $(N_1, \ldots, N_M)$ of N, let the *associated* vector **t** be defined by:

$$t_i = [K_l(N_l)/H_l(N_l)]^{1/2} \quad i \in N_l, \quad l = 1, \ldots, M.$$

A partition will be referred to as *optimal* if the associated **t**-vector is an optimal solution for **RP**.

The following lemma, whose proof follows easily from the definition of a minimum cut, is useful in generating an optimal partition.

Lemma 3. *Let* $\{N_l\colon l = 1, \ldots, M\}$ *be a partition of* N *and fix* $l = 1, \ldots, M$. *Let* $(\underline{N}, \bar{N})$ *be a min-cut of an associated polymatroidal network* $G(\nu, N_l)$ *for some* $\nu > 0$. *Then*

i. $$\frac{K_l(\underline{N})}{H_l(\underline{N})} \le \nu \le \frac{K_l(N_l) - K_l(\underline{N})}{H_l(N_l) - H_l(\underline{N})}.$$

ii. *If* $(\underline{N}_1, \underline{N}_2)$ *is a partition of* $\underline{N}$, *then*

$$\frac{K_l(\underline{N}) - K_l(\underline{N}_1)}{H_l(\underline{N}) - H_l(\underline{N}_1)} \le \nu.$$

Similarly, if $(\bar{N}_1, \bar{N}_2)$ *is a partition of* $\bar{N}$, *then*

$$\nu \le \frac{K_l(\underline{N} \cup \bar{N}_1) - K_l(\underline{N})}{H_l(\underline{N} \cup \bar{N}_1) - H_l(\underline{N})}.$$

Theorem 1 and Lemma 3 suggest the following algorithm for solving **RP**.

Algorithm RP

STEP 0. $M := 1$, $l := 1$, $N_1 := N$.

STEP 1. (Let $H_l(\cdot)$ and $K_l(\cdot)$ be defined as in Definition 1.) Set $\nu = K_l(N_l)/H_l(N_l)$. Find a maximum flow in the associated polymatroidal network $G(\nu, N_l)$; if (N_l, ϕ) is a min-cut, go to Step 2. Otherwise, we have a nontrivial min-cut $(\underline{N}_l, \bar{N}_l)$ of $G(\nu, N_l)$. Renumber $(\underline{N}_l, \bar{N}_l, N_{l+1}, \ldots, N_M)$ as $(N_l, N_{l+1}, \ldots, N_{M+1})$; $M := M + 1$ and repeat Step 1.

STEP 2. If $l = M$, stop, $(N_1, N_2, \ldots, N_M)$ is the desired partition; otherwise $l := l + 1$ and go back to Step 1.

Algorithm **RP** results in an optimal partition, i.e., the associated vector **t** is an optimal solution of **RP**. The vector **t** can be rounded off to obtain optimal power-of-two policies by using procedures similar to those used by Maxwell and Muckstadt and Roundy. See Federgruen and Zheng (1991) for a proof.

Note that after the first execution of Step 1, each subsequent execution follows an increase of either M or l by one unit. Since M represents the number of sets in the final partition $\{N_1, \ldots, N_M\}$, we must have $M \leq N$. Thus, at most $(2N - 1)$ maximum flow computations are needed in associated polymatroidal networks. The computational complexity of the algorithm is thus given by $0(\Gamma N)$, with Γ a bound for the complexity of determining a maximum flow in one of the polymatroidal networks $G(\nu, N_l)(\nu > 0, N_l \subset N)$. The magnitude of Γ will be discussed in detail in Section 3.

When the base planning period T_L is predetermined, the two-stage algorithm can be reduced to the following *integrated* algorithm which generates an optimal power-of-two policy directly.

Algorithm P

Let $\nu^0 = 2^{2j-1}T_L^2$, with j the unique integer such that

$$\nu^0 \leq K(N)/H(N) < 4\nu^0\Big(H(N) = \sum_{r \in R} H_r\Big).$$

STEP 0. $\nu := \nu^0$. Find a min cut (N_1, N_2) of $G(\nu, N)$. $M := 2$;

STEP 1. If $N_1 := \phi$, then **begin** $\nu := 4\nu^0$; go to Step 2 **end**

$\nu := \nu/4$. Find a min cut $(\underline{N}_1, \bar{N}_1)$ of $G(\nu, N_1)$.

Rename $(\underline{N}_1, \bar{N}_1, N_2, \ldots, N_M)$ as $(N_1, N_2, \ldots, N_{M+1})$;

$M := M + 1$; $t_i^* := \sqrt{2\nu}$, $i \in N_2$; go back to Step 1.

STEP 2. If $N_M = \phi$, then stop.

Find a min cut $(\underline{N}_M, \bar{N}_M)$ of $G(\nu, N_M)$.

Rename $\underline{N}_M$, $\bar{N}_M$ as N_M, N_{M+1}; $t_i^* = \sqrt{\nu/2}$, $i \in N_M$

$M := M + 1$; $\nu := 4\nu$; go back to Step 2.

The power-of-two vector $\mathbf{t}^*$ generated by algorithm **P** specifies an optimal power-of-two policy. This is easy to verify by the characterization theorem below. Clearly, the computational complexity of algorithm **P** is $(M + 1)$ $0(\Gamma)$, where M is the number of distinct power-of-two values in the vector $\mathbf{t}^*$. In practice, M would be a small number, say, no more than 10. We conclude that the overall complexity of algorithm **P** is $O(\Gamma)$, as opposed to algorithm **RP** with complexity bound $O(N\Gamma)$.

Algorithm **P** is based on the following characterization theorem for the integer program **P**.

Theorem 2. *For* $\mathbf{t} \in \mathbf{T}$, *suppose that* $\{N_l: l = 1, \ldots, M\}$ *is a partition of* N *such that* $N_l = \{i \in N: t_i = t(l)\}(l = 1, \ldots, M)$, *and* $t(1) < t(2) < \cdots < t(M)$. *Let* $K_l(\cdot)$, $H_l(\cdot)$ *and* E_l *be defined as in Definition* 1, $(l = 1, \ldots, M)$. *Then* **t** *is optimal for* **P** *if the following conditions hold for all* $l = 1, \ldots, M$:

i. $\frac{1}{\sqrt{2}}[K_l(N_l)/H_l(N_l)]^{1/2} \leq t(l) \leq \sqrt{2}\,[K_l(N_l)/H_l(N_l)]^{1/2}$;

ii. $\frac{1}{\sqrt{2}}t(l) \leq (K_l(S)/H_l(S))^{1/2}$ for all $S \subseteq N_l$;

$$([K_l(N_l) - K_l(S)]/[H_l(N_l) - H_l(S)])^{1/2} \leq \sqrt{2}\,t(l) \quad \text{for all } S \subseteq N_l.$$

We refer to Federgruen and Zheng (1991) for proofs of Theorem 2, and the claims that the algorithms generate optimal power-of-two policies. Readers familiar with Federgruen and Zheng 1990, 1992 may observe similarities in the development of parallel results for special cases.

3. COMPUTATIONAL COMPLEXITY OF THE ALGORITHMS

Algorithms **RP** and **P** have complexity $0(N\Gamma)$ and $0(\Gamma)$, respectively, with Γ a bound for the complexity of determining a minimum cut (or maximum flow) in one of the associated polymatroidal networks $G(\nu, N_l)$ ($\nu > 0$ and $N_l \subset N$). In this section we discuss efficient implementations of these maximum flow problems tailored to the specific structure of the associated polymatroidal networks. We refer to Tardos, Tovey and Trick (1986) for a review of augmenting path algorithms for *general* polymatroidal network flow problems.

As pointed out in the Introduction, the polymatroidal flow network $G(\nu, N_l)$ is of a rather special type: Only the flows on the arcs pointing toward the sink are constrained by general polymatroidal capacity constraints; all other arcs are constrained by individual capacity bounds only. We exploit this special structure and the general results in Tardos, Tovey and Trick to design a maximum flow algorithm whose complexity bound is $0(RN^3d)$.

We first review some of the basic results in Lawler and Martel, and Tardos, Tovey and Trick. In the terminology of the latter, a class of augmenting path methods for a given polymatroidal network flow problem in a graph (N, A) with source s and sink t is specified as follows:

1. For a given feasible flow $\mathbf{y}$, let $AUG(\mathbf{y})$ be the set of all augmenting paths. An augmenting path is an undirected path in the network starting at the source s and terminating at the sink t; each augmenting path has a capacity $\delta(> 0)$ which denotes the maximum amount by which the flows on the forward arcs in the path may be increased (and the backward arcs decreased).
2. Each augmenting path method in the considered class has the following structure:

STEP 0. **(Initialization)** Set $\mathbf{y} := \mathbf{y}^0$, an initial feasible flow.

STEP 1. If $AUG(\mathbf{y}) = \phi$, stop; otherwise, select an augmenting path in $AUG(\mathbf{y})$ and determine its capacity δ.

STEP 2. Augment the flow on all forward arcs by δ and decrease the flow on the backward arcs by δ. Return to Step 1.

Thus, within a given class of augmenting path methods only the choice of a specific augmenting path in $AUG(\mathbf{y})$ and the computational procedure employed for its identification may vary. An auxiliary network associated with a given graph (N, A) and a given class of augmenting path methods is a graph $(\bar{N}, \bar{A})$ with node set $\bar{N}$ (containing again a source s and a sink t) and arc set $\bar{A}$ with the following properties:

i. associated with each feasible flow $\mathbf{y}$ in (N, A) is a nonnegative capacity $\delta_{\mathbf{y}}(e)$ for each arc $e \in \bar{A}$ and a set of s-t augmenting paths defined as all *directed* paths from s to t on which all capacities $\delta_{\mathbf{y}}(e)$ are *positive*;
ii. for each feasible flow $\mathbf{y}$ in (N, A), there exists a one-to-one correspondence between $AUG(\mathbf{y})$ and the set of s-t augmenting paths in $(\bar{N}, \bar{A})$.

Let d denote the time required to evaluate one of the capacities $\{\delta_{\mathbf{y}}(e), e \in \bar{A}\}$ for any given flow $\mathbf{y}$, n the maximum length of an augmenting path, and Δ the total amount of work required to compute the $\delta_{\mathbf{y}}(\cdot)$ numbers for any subset of n arcs.

Tardos, Tovey and Trick show that a specific augmenting path method (within the given class of methods) may be designed with complexity $0(n|\bar{A}|\Delta)$ provided that three properties are satisfied. Before stating these, we first need the following definitions.

For any feasible flow $\mathbf{y}$ in (N, A), let $\sigma_{\mathbf{y}}(i)[\tau_{\mathbf{y}}(i)]$ be the length of the shortest $s - i$ $[i - t]$ augmenting path in the auxiliary network. (If no such path exists let $\sigma_{\mathbf{y}}(i) = \infty[\tau_{\mathbf{y}}(i) = \infty]$.) Consider a specific indexing of the nodes in $\bar{N}$. Given two paths P and P' in $(\bar{N}, \bar{A})$ with the same number of arcs, we say that P is lexicographically smaller than P' if the vector of node indices in P, placed in reverse order, is lexicographically smaller than the vector of node indices in P' (also placed in reverse order).

Property 1. If the augmenting paths in Step 1 are chosen to correspond with shortest s-t augmenting paths in $(\bar{N}, \bar{A})$, then $\sigma_{\mathbf{y}}(i)$ and $\tau_{\mathbf{y}}(i)$ are nondecreasing for all i.

In view of Property 1 the algorithm may be partitioned into phases according to the value of $\sigma_{\mathbf{y}}(t)$.

Property 2. If all augmentations correspond with minimum length s-t augmenting paths in $(\bar{N}, \bar{A})$, then after an augmentation changing $\mathbf{y}$ to $\mathbf{y}'$, the first arc on the path with $\delta_{\mathbf{y}}(e)$ minimal will have $\delta_{\mathbf{y}'}(e) = 0$.

Property 3. In each phase, apply a specific indexing of the nodes in $\bar{N}$. If augmentations are chosen to correspond with lexicographically minimal shortest paths in $(\bar{N}, \bar{A})$ with respect to this indexing, then the paths that realize $\sigma_{\mathbf{x}}(i)$ are lexicographically nondecreasing during this phase for each node i.

Lemma 4. (see Theorem 4.3 in Tardos, Tovey and Trick) *Consider a graph* (N, A), *a class of augmenting path algorithms and an auxiliary network* $(\bar{N}, \bar{A})$ *satisfying Properties* 1, 2 *and* 3. *There exists an augmenting path method in the considered class with complexity* $0(n|\bar{A}|\Delta)$.

Now consider an associated polymatroidal network flow problem in $G(\nu, N_l)$ for some $\nu > 0$ and $N_l \subset N$; see Definition 2. Note that all networks $G(\nu, N_l)$ have a similar bipartite structure and that the associated capacity set functions $K_l(\cdot)/\nu$, are simple transformations of the basic set function $K(\cdot)$. The complexity of the proposed maximum flow algorithms, as applied to the networks $G(\nu, N_l)$ is thus bounded by their performance with respect to the full network $G(\nu, N)$. Therefore, in the remainder of this section we focus on the latter. If $S \subseteq N$ and $u \in R^N$, we write $u(S)$ instead of $\sum_{i \in S} u_i$.

Given a feasible flow $\mathbf{y} = (\mathbf{x}, \mathbf{v})$, a set $S \subset N$ is said to be *tight* if $\sum_{i \in S} y(i, t) = \sum_{i \in S} v_i = K(S)/\nu$ and an arc (i, t) is said to be *saturated* if there is some tight set containing i. It follows from Lemma 2.2 in Lawler and Martel that if arc (i, t) is saturated there exists a *minimal* tight set containing i, which we call $U(i)$. In view of the special structure of the associated polymatroidal network $G(\nu, N)$, Lawler and Martel's definition of the set of augmenting paths $AUG(\mathbf{y})$ for a given feasible flow $\mathbf{y}$ amounts to the collection of undirected paths from s to t with the properties:

a. each backward arc e on the path is nonvoid, i.e., $y(e) > 0$;
b. for a forward arc e on the path which does not point to the sink, the flow $y(e)$ is smaller than the arc's capacity;
c. if a forward arc (i, t) is saturated $(i \in N)$, then the following arc on the path is a backward arc (t, j) with $j \in U(i)$.

Observe that the flow on an arc (i, t) which is saturated for a given feasible flow $\mathbf{y}$, may nevertheless sometimes be increased by an equal reduction of the flow on some arc (j, t) $(j \neq i)$. The amount by which the flow on (i, t) may be increased is thus dependent on the specific backward arc (t, j) which is chosen as the successor arc of (i, t). This suggests the following auxiliary network $\bar{G} = (\bar{N}, \bar{A})$.

Note first that only arcs which start at a product node are incident to the sink t. Thus, let

$$\bar{N} = \{N \cup \{s\} \cup \{t\} \cup \{t_i\colon i \in N\}\}$$

with cardinality $O(R + N)$ and

$$\bar{A} = \{\mathbf{A} \cup \mathbf{A}^{-1}\} \cup \{(s, r), (i, t), (t_i, i)\colon r \in R, i \in N\} \cup \{(i, t_j)\colon i \neq j, i, j \in N\}$$

with cardinality $O(RN)$. For any arc $e \in \bar{A}$ and feasible flow $\mathbf{y}$ define

$$\delta_y(e) = \begin{cases} H_r - x_{sr} & e = (s, r) \\ \infty & e = (r, i) \in \mathbf{A} \\ y(e^{-1}) & e \in \mathbf{A}^{-1} \\ \min\{K(S)/\nu - v(S) | i \in S \subseteq N\{j\}\} & e = (i, t_j) \\ y((i, t)) = v_i & e = (t_i, i) \\ \min\{K(S)/\nu - v(S) | i \in S \subset N\} & e = (i, t). \end{cases} \quad (12)$$

We first show that for any feasible flow $\mathbf{y}$, there is a one-to-one correspondence between $AUG(\mathbf{y})$ and the collection of s-t augmenting paths in $\bar{G}$: Any arc (s, r) with $r \in R$ and $e \in \mathbf{A} \cup \mathbf{A}^{-1}$ corresponds to the "same" arc in $\bar{A}$. The same applies to any arc (i, t) $(i \in N)$ when it is used as the *last* arc on the path. If (i, t) is followed by a backward arc (t, j) $(j \in N)$ the corresponding arc in the auxiliary network is (i, t_j) and the arc corresponding to (t, j) is (t_j, j). With this one-to-one correspondence, we have for any feasible flow $\mathbf{y}$ that an undirected path from s to t in $G(\nu, N)$ belongs to $AUG(\mathbf{y})$ if and only if the corresponding path in the auxiliary network consists of arcs with $\delta_{\mathbf{y}}(e) > 0$.

Lemma 5. *For any $\nu > 0$ consider an associated polymatroidal network $G(\nu, N)$. The above defined class of augmenting path methods and the auxiliary network $\bar{G}(\bar{N}, \bar{A})$ with the capacity function $\{\delta_{\mathbf{y}}$: $\mathbf{y}$ feasible flow in $\bar{G}(\bar{N}, \bar{A})\}$ satisfy Properties* 1, 2, *and* 3.

Proof. Property 1 follows immediately from Lemma 11.1 in Lawler and Martel and the observation that for any feasible flow $\mathbf{y}$, each augmenting path in $AUG(\mathbf{y})$ and its corresponding s-t augmenting path in the auxiliary network have the same number of arcs. Lawler and Martel define in the network G for any feasible flow $\mathbf{y}$ and each arc pair $(e, \bar{e})$, such that e and $\bar{e}$ are incident to a common node, a capacity number $\delta'_y(e, \bar{e})$ (see Section 9 in Lawler and Martel). For arcs e in $\bar{A}$ which are not incident to one of the nodes $\{t_i\colon i \in N\}$, it is easy to verify that $\delta_{\mathbf{y}}(e) = \delta'_{\mathbf{y}}(e', \bar{e})$, where e' is the corresponding arc in $G(\nu, N)$ and $\bar{e}$ is any arc that shares a node with e'. Likewise, if $e = \{i, t_j\}$ for $i, j \in N$, then $\delta_{\mathbf{y}}(e) = \delta'_{\mathbf{y}}((i, t), (t, j))$ and if $e = (t_i, i)$, then $\delta_y(e) = \delta'_y((t, i), \bar{e})$ for any arc $\bar{e}$ in $G(\nu, N)$ with node i as its tail or head. These identities and Theorem 9.1 in Lawler and Martel establish Property 2.

Finally, to prove Property 3 recall first from Theorem 8.1 in Tardos, Tovey and Trick that when all augmenting paths are chosen to be of minimal length in a *given phase* of the algorithm, no arc in $G(\nu, N)$ may be used simultaneously as a forward and backward arc. In a given phase, assign an index $i(w)$ for each node w in $\bar{N}$ as follows: $i(t) = 0$, $i(t_l) = l$, $i(l) = 2l$ for $l \in N$; the nodes $r \in R$ are arbitrarily indexed from $2N + 1$ to $2N + R$, and $i(s) = 2N + R + 1$. Let $\hat{A}$ be the collection of directed arcs in the original network which are used in some augmenting path during this phase. Assign an index $i(e)$ to each arc e in $\hat{A}$ so that for $e^1 = (w^1, w^2)$ and $e^2 = (w^3, w^4)(e^1 \neq e^2)$, $i(e^1) < i(e^2)$ if $i(w^2) < i(w^4)$ or $i(w^2) = i(w^4)$ and $i(w^1) < i(w^3)$.

Lemma 11.3 in Lawler and Martel establishes that for *any* indexing of the arcs in $\hat{A}$ (and thus, in particular, for the above constructed indexing) Property 3 is satisfied with respect to the vector of arc indices. Property 3 now follows from this result and the following observation: For any node $i \in N$ consider a pair of equal length paths (in $G(\nu, N)$)P_1, P_2 from s to i using arcs in $\hat{A}$ and let $\bar{P}_1$, $\bar{P}_2$ be the corresponding pair of paths from s to i in the auxiliary network. If the vector of arc indices of P_1 placed in reverse order is lexicographically smaller than the vector of arc indices of P_2 (also placed in reverse order), then the vector of node indices in $\bar{P}_1$ is lexicographically smaller than the vector of node indices in $\bar{P}_2$ (when both vectors are again placed in reverse order).

Observe that the maximum length of a *shortest* augmenting path in this network is bounded by $2N + 1$. This follows because, with the exception of the first two nodes, every other node along the path must be one of the product nodes of which there are at most N. By Lemma 4, with $n \leq 2N + 1$, $|\bar{A}| = 0(RN)$ and $\Delta \leq (2N + 1)d$, we conclude that $\Gamma = 0(RN^3 d)$.

4. COMPLEXITY BOUNDS FOR SPECIFIC TYPES OF COST STRUCTURES

The complexity bound $\Gamma = 0(RN^3 d)$ for algorithm **P** applies to all (submodular) joint cost structures (and all network topologies). Only the magnitude of d depends on the specific cost structures employed. In this section, we discuss the magnitude of d for the most important special structures (see subsection 4.1).

For some cost structures, it is possible to replace polymatroidal network flow problems by equivalent (classical) maximal flow problems on an expanded graph, i.e., with capacity restrictions on individual arcs only, as in the case of separable cost structures. (Such networks are often referred to as *modular networks*.) The resulting increase in the number of nodes and arcs is usually vastly compensated for by the fact that large efficiency improvements can be obtained for *ordinary* (modular) maximum flow problems. The latter have the additional advantage of permitting the use of generally accessible maximum flow routines. In subsection 4.2 we discuss some of the cost structures which allow for a transformation into ordinary maximum flow problems.

In the discussion below, we assume that a single type of joint cost structure applies to all nodes in N. The transformations and optimization routines are easy to extend to settings, where the nodes in N are partitioned into several classes, each with its own type of cost structure.

In assessing the order of magnitude of d, note from (12) that the computation of $\delta_{\mathbf{y}}(e)$ is trivial $(0(1))$ for all cases except when $e = (i, t_j)$ or $e = (i, t)$ with $i, j \in N$. If $e = (i, t_j)$ define $\hat{v} \in R^N$ by $\hat{v}_i = X$ (X is large), $\hat{v}_j = 0$ and $\hat{v}_l = v_l$ for all $l \neq i, j$. If $e = (i, t)$ define $\hat{v}$ by $\hat{v}_i = X$ (X is large); $\hat{v}_l = v_l l \neq i$. Note that in both cases

$$\delta_y(e) = X - v_i - \hat{v}(N) + \min\{K(S)/\nu + \hat{v}(N\backslash S);\ S \subset N\}$$
$$= \min\{K(S)/\nu + \hat{v}(N\backslash S):\ S \subset N\} - \begin{cases} \sum_{l \neq j} v_l, & \text{if } e = (i, t_j) \\ \sum_{l=1}^{N} v_l, & \text{if } e = (i, t). \end{cases}$$

The first term to the right of this equation is a special instance of the so-called *minimum oracle* problem for polymatroids.

Minimum Oracle Problem

For a given vector $\mathbf{u} \in R^N$ and submodular set function $K(\cdot)$ defined on 2^N (the subsets of N) determine a set S^* achieving

$$\min\Big\{K(S) + \sum_{i \in N\backslash S} u_i:\ S \subseteq N\Big\}. \tag{13}$$

It is well known (see Edmonds 1970) that a solution to the minimum oracle problem may be obtained by solving the "dual" *maximum element* problem.

Maximum Element Problem

For a given vector $\mathbf{u} \in R_+^N$ and submodular set function $K(\cdot)$, determine a solution $\mathbf{x}^* \in R^N$, achieving

$$\max \sum_{i \in N} x_i \tag{14a}$$

subject to

$$\sum_{i \in S} x_i \leqq K(S), \quad S \subseteq N \tag{14b}$$

$$x_i \leqq u_i, \quad i \in N \tag{14c}$$

$$\mathbf{x} \geqslant 0 \tag{14d}$$

and select S^* as the largest set S for which (14b) is binding when $\mathbf{x} = \mathbf{x}^*$.

(A largest binding set exists because if (14b) is binding for $S_1, S_2 \subseteq E$, then it is binding for $S = S_1 \cup S_2$ as well.) In particular, we have the following lemma.

Lemma 6. *Max*$\{x(N)$: *x satisfies* (14b)–(14d)$\}$ = *min*$\{K(S) + \sum_{i \in N\backslash S} u_i: S \subseteq N\}$.

Another related problem is the so-called *membership test* problem, which consists of verifying whether a given vector $\mathbf{x} \in R^N$ satisfies (14b) and (14d). Grötschel, Lovasz and Schryver developed polynomial general solution methods for the above three problems (the minimum oracle, maximum element, and membership test problems) which apply to *general* submodular set functions (and the corresponding polymatroids). These methods, however, are based on repeated use of the ellipsoid method and therefore appear cumbersome. (A pseudopolynomial, combinatorial procedure is due to Cunningham (1986); see also Bixby, Cunningham and Topkis (1985).)

4.1. Special Cost Structures: Efficient Implementations of the Polymatroidal Maximum Flow Algorithm

In this subsection, we discuss efficient implementations of the general polymatroidal algorithm of Section 3 for a number of important special cost structures.

The First-Order Interaction Model

$$K(S) = \begin{cases} K_0 + \sum_{i \in S} K_i & \text{if } S \neq \phi \\ 0 & \text{if } S = \phi. \end{cases}$$

The first-order interaction model is the most frequently used cost structure in models with joint costs. The literature on the so-called joint replenishment problem confines itself almost exclusively to this structure; see Federgruen and Zheng (1992) and the many references and survey articles cited therein.

Generalized Symmetric Structure

As in the first model, each node $i \in N$ is characterized by a single value $K_i \geqq 0$ and $K(S) = f(\sum_{i \in S} K_i)$ with f an arbitrary, nondecreasing concave function. The first-order interaction structure represents a special case with $f(x) = K_0 + x$ if $x > 0$ and $f(0) = 0$. It is easy to verify that $K(\cdot)$ is monotone and submodular. The term "generalized symmetric" was first introduced in Federgruen and Groenevelt (1988). (This structure generalizes the "symmetric" case discussed in subsection 4.2, where all $K_i = 1$, i.e., where $K(S) = f(|S|)$; see Topkis (1984) and Lawler and Martel (1982).)

Generalized Symmetric Structures of Order *m*

This structure provides a further generalization of the second model. There are $m \geqq 2$ distinct attributes $\alpha_i^1, \alpha_i^2, \ldots, \alpha_i^m$ associated with each node $i \in N$. Let f:

$R^m \to R$ be continuously differentiable everywhere and twice differentiable almost everywhere, with ∇f the gradient and $\nabla^2 f$ the Hessian. Assume that $f(0) = 0$, $\nabla f \geqq 0$, and $\nabla^2 f \leqq 0$ (almost everywhere). If

$$K(S) = f\Big(\sum_{i\in S} \alpha_i^1, \sum_{i\in S} \alpha_i^2, \dots, \sum_{i\in S} \alpha_i^m\Big) \quad S \subset N, \quad (15)$$

the structure is called generalized symmetric of order m.

For example, when $m = 2$, α_i^1 may be a measure for the production setup and α_i^2 for the transportation setup involved when item i is included in the production batch. A special case arises when a generalized symmetric structure (of order 1) is modified by a quantity discount which is a function of the number of items included in the replenishment batch, i.e.,

$$K(S) = g\Big(\sum_{i\in S} K_i\Big) - h(|S|)$$

with $g(\cdot)$ concave and $h(\cdot)$ convex. (Let $m = 2$, $\alpha_i^1 = K_i$ and $\alpha_i^2 = 1$, $i \in N$; $f(x, y) = g(x) - h(y)$.) Andres and Emmons (1975) consider a special case of this structure with $h(l) = 1$ if $l = N$ and $h(l) = 0$ for $l < N$.

More generally, generalized symmetric functions may be used to *approximate* more general cost structures, in a similar way that polynomials or rational functions are used to approximate general nonlinear functions. Observe that a generalized symmetric function is specified by Nm numbers as opposed to 2^N values required to specify a general set function. The α-numbers may be estimated by standard regression techniques, using indicator variables to represent set membership. This approximation technique is similar in spirit to that employed in Herer and Roundy (1990) to approximate (submodular) vehicle routing costs by a family structure; see subsection 4.2.3.

The following lemma, establishing that $K(\cdot)$ is monotone and submodular, is due to Federgruen and Groenevelt. (See the Appendix for its proof.)

Lemma 7. *Let $f(\cdot)$ be continuously differentiable everywhere, and twice differentiable almost everywhere. Assume that $f(0) = 0$, $\nabla f \geqslant 0$, and $\nabla^2 f \leqslant 0$ (almost everywhere). The set function $f(\cdot)$ specified by (15) is monotone and submodular.*

If $\mathbf{x}^*$ is an optimal solution to (14), let $\text{sat}(\mathbf{x}^*)$ denote its saturated set, i.e., the largest set S for which (14b) is binding when $\mathbf{x} = \mathbf{x}^*$.

4.1.1. The First-Order Interaction Structure

Let $N_1 = \{i \in N: u_i \geqslant K_i\}$. We distinguish between two cases:

a. $(u(N_1) \geqslant \sum_{i\in N_1} K_i + K_0)$ In this case, $S^* = N_1$ is a minimizing set. To verify this, note that for any $S \subset N$ with $S \neq \emptyset$,

$$\begin{aligned} K(S) - u(S) &= K_0 + \sum_{i\in S} K_i - u(S) \\ &= \Big(K_0 + \sum_{i\in S\cap N_1} k_i - \sum_{i\in S\cap N} u_i\Big) \\ &\quad + \sum_{i\in S\setminus N_1} (K_i - u_i) \geqq K_0 + \sum_{i\in N_1} K_i - \sum_{i\in N_1} u_i \\ &= K(N_1) - u(N_1); \end{aligned}$$

also

$$K(N_1) - u(N_1) \geqq 0 = K(\phi) - u(\phi).$$

b. $(u(N_1) < \sum_{i\in N_1} K_i + K_0)$ In this case, $S^* = \phi$ is a unique minimizing set since for any $S \neq \phi$,

$$\begin{aligned} K(S) - u(S) &= \Big(K_0 + \sum_{i\in S\cap N_1} K_i - \sum_{i\in S\cap N_1} u_i\Big) \\ &\quad + \sum_{i\in S\setminus N_1} (K_i - u_i) > 0. \end{aligned}$$

For both a and b, the minimum oracle problem thus reduces to N comparisons only.

4.1.2. Generalized Symmetric Structures of Order 1

The maximal element problem can be solved by the following procedure developed by Federgruen and Groenevelt (1988).

STEP 1. Find a permutation $(\alpha_1, \alpha_2, \dots, \alpha_N)$ of N for which

$$u_{\alpha_1}/K_{\alpha_1} \geqq u_{\alpha_2}/K_{\alpha_2} \geqq \cdots \geqq u_{\alpha_N}/K_{\alpha_N}.$$

Let $x_{\alpha_1} := \min\,(f(K_{\alpha_1}), u_{\alpha_1})$

STEP 2. **For** $l = 2$ to N **do**
if $u_{\alpha_l} < f(\sum_{i=1}^{l} K_{\alpha_i}) - \sum_{i=1}^{l-1} x_{\alpha_i}$
then $x_{\alpha_l} := u_{\alpha_l}$
else $j := l$; $x_{\alpha_l} := f(\sum_{i=1}^{l} K_{\alpha_i}) - \sum_{i=1}^{l-1} x_{\alpha_i}$.

The procedure ends up with a maximal $\mathbf{x}$ in (14) and $\text{sat}(\mathbf{x}) = \{\alpha_1, \dots, \alpha_j\}$. See Federgruen and Groenevelt (1988) for a proof. The complexity of this procedure is obviously $0(N \log N)$. It is easy to verify that only *one* ranking of the nodes in N, i.e., one execution of Step 1 is required to compute capacity numbers $\{\delta_{\mathbf{y}}(e)\}$ for a given feasible flow $\mathbf{y}$ and a complete *collection* of arcs in the auxiliary network. Thus, invoking the complexity bound in Lemma 3 we have $\Delta = 0(N^2)$, so that $\Gamma = 0(RN^4)$.

4.1.3. Generalized Symmetric Structure of Higher Order

We restrict ourselves to structures of order 2, i.e.,

$$K(S) = f\Big(\sum_{i\in S} K_i, \sum_{i\in S} M_i\Big),$$

where $f(0) = 0$. Then $\nabla f \geqq 0$ and $\nabla^2 f \leqq 0$, and K_i, $M_i \geqq 0$ for all $i \in N$.

An efficient solution procedure for the maximum element problem was obtained by Federgruen and Groenevelt under the additional assumption that f is concave. This procedure employs an $0(N^2 \log N)$ membership test described in the Appendix.

Minimum Oracle Procedure for Generalized Symmetric Structures of Order 2 (GS2)

STEP 1. $T := \phi$;

STEP 2. **For** $j = 1$ **to** N **do**
begin find $\bar{x} = \min\{K(S \cup \{j\}) - \sum_{l \in S} x_l$: $T \subset S \subset \{1, \ldots, j-1\}$,
and let S' be the largest subset for which this minimum is achieved
if $\bar{x} > u_j$, then $x_j := u_j$
else $x_j := \bar{x}$; $T := S' \cup \{j\}$.
end.

Lemma 8. *The* **GS2** *procedure terminates with* **x** *representing a maximal element in* (14) *and* $T = \text{sat}(\mathbf{x})$ *solves the minimum oracle problem.*

Proof. It is well known from Edmonds that $F^* = (\mathbf{x}$: $\mathbf{x}$ satisfies (14b), (14c)} is a polymatroid. A maximal element **x** may thus be obtained by the greedy procedure: first x_1 is set at its maximal value; in the jth iteration, x_j is set at the highest feasible value given the values for $x_1, \ldots, x_{j-1}$. The above procedure represents an implementation of this greedy procedure in which sat(**x**) is simultaneously constructed.

In the jth iteration of the procedure, given fixed values for $x_1, \ldots, x_{j-1}$ the maximum feasible value for x_j is $\min\{u_j, \hat{x}\}$, where

$$\hat{x} = \min\left\{K(S \cup \{j\}) - \sum_{l \in S} x_l \colon S \subset \{1, \ldots, j-1\}\right\}.$$

Let $\hat{S}$ be a set which attains this minimum. Note by induction that at the start of the jth iteration of Step 2, $T \subset \{1, \ldots, j-1\}$. Also (14b) is a binding constraint for the set T and the vector $(x_1, \ldots, x_{j-1}, t, 0, \ldots, 0)$ for any $t > 0$. We refer to such a set as a *tight* set. Thus, both $\hat{S}$ and T are tight sets in F^* for $x = (x_1, \ldots, x_{j-1}, \hat{x}, 0, \ldots, 0)$ and, hence, $S^0 = \hat{S} \cup T$ is tight as well. Thus, $\hat{x} = f(S^0 \cup \{h\}) - \sum_{l \in S^0} x_l$ and $\{1, \ldots, j\} \supseteq S^0 \supseteq T$. So $\hat{x} = \bar{x}$. If $\bar{x} > u_j$, (14b) is redundant for every set $S \subset N$ with $j \in S$ and $x = (x_1, \ldots, x_j, 0, \ldots, 0)$; thus, T remains the largest tight subset of $\{1, \ldots, j\}$. Alternatively if $\bar{x} \leqq u_j$, $S' \cup \{j\}$ is the largest tight subset of $\{1, \ldots, j\}$.

It is easy to verify that in Step 2 of **GS2** $\bar{x}$ may be found by the membership procedure for generalized symmetric polymatroids of order 2, restricting oneself to permutations of $\{1, \ldots, j-1\}$ in which the elements of T precede those in $\{1, \ldots, j-1\} \backslash T$. Since N membership tests need to be performed in a single execution of the procedure, the complexity of **GS2** is $0(N^3 \log N)$.

4.2. Special Cost Structures: Efficient Solution Via Equivalent Ordinary Maximum Flow Problems

In this subsection, we describe a number of cost structures for which the polymatroidal maximum flow problems can be solved efficiently via *transformation* into *ordinary* maximum flow problems.

4.2.1. Symmetric Structure

The setup cost structure $K(\cdot)$ is said to be *symmetric* if

$$K(S) = f(|S|), \quad S \subset N$$

with f a concave nondecreasing function and $f(0) = 0$. (As mentioned above this is a special case of the generalized symmetric structure.) See Federgruen and Groenevelt (1987) for a description of a transformation of $G(\nu, N)$ into an equivalent ordinary network flow problem, with at most N additional nodes and N^2 additional arcs. By a generalization of the arguments in Gusfield, Martel and Fernandez-Baca (1985) one verifies that a maximum flow in $G(\nu, N)$ is computable in $0(RN^2)$ operations; see also Ahuja et al. (1988).

4.2.2. Tree Structured Partially Dedicated Machines

In some settings, the production plant consists of several (say M) machines, each of which is suitable for a specific set of products. Let N_m denote the set of products which can be manufactured on machine $m (m = 1, \ldots, M)$ and assume that the machines represent nested levels of specialization or flexibility, i.e., for any pair of machines m, m' we have either $N_m \subset N_{m'}$ (m' is more flexible than m); $N_m \supset N_{m'}$ (m is more flexible than m') or $N_m \cap N_{m'} = \phi$ (m and m' are dedicated to disjoint collections of items). Let K_m denote the setup cost incurred when starting a production batch on machine $m (m = 1, \ldots, M)$, regardless of the specific composition of the batch. Let C_i denote an additional setup cost incurred whenever product $i (i = 1, \ldots, N)$ is included in a batch. The setup cost set function $K(\cdot)$ is thus given by:

$$K(S) = \sum_{i \in S} C_i + \min\left\{\sum_{m \in J} K_m \colon \left(\bigcup_{m \in J} N_m\right) \supseteq S\right\}.$$

Federgruen and Groenevelt (1987a) show that this structure is submodular and that it may be represented by a modular one via the addition of at most $2N$ nodes and $2N$ arcs. We observe that the transformed network is still multipartite. By a generalization of the arguments in Gusfield, Martel and Fernandez-Baca, one concludes that maximum flow algorithms exist with $\Gamma = 0(RN^2)$; see also Ahuja et al.

The above structure may be generalized to settings where for every pair of machines (m, m') either $N_m \cap N_{m'} = \phi$ or there exist machines m'' and m''' (not necessarily identical to m and m') such that $N_{m''} = N_m \cap N_{m'}$ and $N_{m'''} = N_m \cup N_{m'}$. To ensure submodularity, the machine setup cost numbers now need to satisfy the inequalities $K_{m''} + K_{m'''} \leqslant K_m + K_{m'}$. This case is referred to as an intersecting family structure. No transformation into modular networks is known for this generalized structure, but it can be handled efficiently with the polymatroidal algorithm of Section 3 with $d = 0((N + M) \log(N + M) + MN)$, solving minimum oracle problems via a bottom-up algorithm, the details of which we omit.

4.2.3. The Family Model (see Roundy 1986)

A list of item families F is given. The setup cost K_f associated with a family $f \in F$ is incurred whenever at least one of the items in f is replenished in a joint replenishment. Thus,

$$K(S) = \left\{\sum K_f: f \in F, f \cap S \neq \phi\right\}, \quad S \subset N.$$

It is easy to show that $K(\cdot)$ is monotone and submodular. To obtain a simple, rather than a polymatroidal network flow representation, remove arcs $\{(i, t), i \in N\}$. Add a node for each $f \in F$. Add an infinite capacity arc (i, f) for each node $i \in f$; connect each node $f \in F$ with the sink t by an arc with capacity K_f. Roundy (1986) described a transformation into an ordinary network model. Our transformation uses a route-product network that is different from his. As a result, the transformed network is a tripartite network. By a generalization of the arguments in Gusfield, Martel and Fernandez-Baca, it can be shown that $\Gamma = 0((R + F)N^2)$, where F denotes $|F|$; see also Ahuja et al. The first-order interaction model may be viewed as an important special case with $F = \{N, \{1\}, \{2\}, \dots, \{N\}\}$. Approaching the first-interaction model this way implies adding $(N + 1)$ additional nodes to the original network, instead of a *single* additional node in the approach described above.

APPENDIX

Proof of Lemma 7. Define $\alpha^i(S) = \sum_{j\in S} \alpha^i_j$, $i = 1, \dots, m$, and $\alpha(S) = (\alpha^1(S), \dots, \alpha^m(S))$; then $K(S) = f(\alpha(S))$. Let $T \subset S \subset E$. Let

$$\Phi(\lambda) \overset{\text{def}}{=} f(\alpha(T) + \lambda[\alpha(S) - \alpha(T)]), \quad \lambda \geq 0.$$

Here $\Phi(\cdot)$ is differentiable in view of the differentiability of f (invoking the chain rule); $\Phi(1) = f(\alpha(S))$ and $\Phi(0) = f(\alpha(T))$. Thus,

$$f(S) = f(\alpha(S)) = \Phi(1) = \Phi(0) + \int_0^1 \Phi'(\lambda)\, d\lambda$$

$$= f(\alpha(T)) + \int_0^1 \sum_{i=1}^m \{(\alpha^i(S) - \alpha^i(T)) \frac{\partial f(\alpha(T) + \lambda[\alpha(S) - \alpha(T)])\}}{\partial x_i}\, d\lambda$$

$$\geq f(\alpha(T)) = K(T)$$

since the integrand is nonnegative. (Note that $\alpha^i(S) \geq \alpha^i(T)$ and $\partial f/\partial x_i \geq 0$ for all $i = 1, \dots, m$.) This proves monotonicity.

To verify submodularity, let $j \notin S$. By the above derivation,

$$K(S \cup \{j\}) - K(S) = \int_0^1 \sum_{i=1}^m \left\{\alpha^i(\{j\}) \frac{\partial f(\alpha(S) + \lambda\alpha(\{j\}))}{\partial x_i}\right\} d\lambda$$

and

$$K(S \cup \{j\}) - K(T) = \int_0^1 \sum_{i=1}^m \left\{\alpha^i(\{j\}) \frac{\partial f(\alpha(T) + \lambda\alpha(\{j\}))}{\partial x_i}\right\} d\lambda.$$

For any $0 \leq \lambda \leq 1$ and $i = 1, \dots, m$, let

$$\Psi_i(\lambda, y) \overset{\text{def}}{=} \frac{\partial f(y + \lambda\alpha_j)}{\partial x_i}, \quad y \geq 0.$$

Note that $\Psi_i(\lambda, \cdot)$ is *continuous* everywhere because ∇f is continuous everywhere; moreover, by assumption it is differentiable with

$$\frac{\partial \Psi_i(\lambda, y)}{\partial y} = \sum_{l=1}^m \frac{\partial^2 f(y + \lambda\alpha_j)}{\partial x_i\, \partial x_l}$$

almost everywhere. Thus,

$$[K(S \cup \{j\}) - K(S)] - [K(T \cup \{j\}) - K(T)]$$

$$= \int_0^1 \sum_{i=1}^m \alpha^i_j[\Psi_i(\lambda, \alpha(S)) - \Psi_i(\lambda, \alpha(T))]\, d\lambda$$

$$= \int_0^1 \sum_{i=1}^m \alpha^i_j \int_{\alpha(T)}^{\alpha(S)} \frac{\partial \Psi_i(\lambda, y)}{\partial y}\, dy\, d\lambda$$

$$= \int_0^1 \sum_{i=1}^m \alpha^i_j \int_{\alpha(T)}^{\alpha(S)} \sum_{l=1}^m \frac{\partial^2 f(y + \lambda\alpha_j)}{\partial x_i\, \partial x_l}\, dy\, d\lambda \leq 0$$

because $\nabla^2 f \leq 0$ (almost everywhere), $\alpha(T) \leq \alpha(S)$ and $\alpha^i_j \geq 0$ for all $i = 1, \dots, m$.

Lemma A.1. (Membership test for generalized symmetric polymatroids of order 2). *Let* $K(S) = f(\sum_{i\in S} K_i, \sum_{i\in S} M_i)$ *be a submodular function on N, where f is concave with* $f(0) = 0$, $\nabla f \geq 0$, $\nabla^2 f \leq 0$. *Then* $x \in F \overset{\text{def}}{=} \{x$: *x satisfies* (14b) *and* (14d)$\}$, *if and only if for every* $0 \leq t \leq 1$

$$\sum_{l=1}^j x_{\alpha_l} \leq K(\{\alpha_1, \dots, \alpha_j\}) = f\left(\sum_{l=1}^j K_{\alpha_l}, \sum_{l=1}^j M_{\alpha_l}\right)$$

$$j = 1, \dots, n \quad \text{(A.2)}$$

holds for some permutation $(\alpha_1, \dots, \alpha_n)$ *of N that satisfies*

$$x_{\alpha_l}/[tK_{\alpha_l} + (1-t)M_{\alpha_l}] \geq x_{\alpha_{l+1}}/[tK_{\alpha_{l+1}} + (1-t)M_{\alpha_{l+1}}].$$

$$l = 1, \dots, n-1. \quad \text{(A.3)}$$

Proof. Consider the collection

$$Z = \left\{\left(\sum_{i\in S} K_i, \sum_{i\in S} M_i, \sum_{i\in S} x_i\right): S \subset N\right\}$$

of points in R^3_+. Obviously, $x \in F$ if and only if the region $\Omega = \{(a, b, \zeta): f(a, b) \geq \zeta\}$ contains Z. Let $\hat{Z}$ be the convex hull of Z. In view of the concavity of f, it suffices to verify that all extreme points of $\hat{Z}$ lie in Ω. Each extreme point is the (unique) optimal solution of the linear program:

$\mathbf{LP}_{\lambda,\mu,\nu} = \max - \lambda a - \mu b + \nu\zeta$

subject to

$(a, b, \zeta) \in \hat{Z}$.

for some λ, μ, ν. Recall that $f(0) = 0$ and $f(\cdot)$ nondecreasing. Thus, if $x \in F$, then there exists an extreme point of $\hat{Z}$ outside Ω which is the optimal solution of $\mathbf{LP}_{\lambda,\mu,\nu}$ for some $\lambda > 0$, $\mu > 0$ and $\nu = 1$. (Such an extreme point is incident to a facet of $\hat{Z}$, where the outward normal (n_1, n_2, n_3) satisfies $n_1 \leq 0$, $n_2 \leq 0$ and $n_3 \geq 0$.) Set $\omega = \lambda + \mu$ and $t = \lambda/(\lambda + \mu)$. Thus, if $x \in F$ there exists an extreme point of $\hat{Z}$ outside Ω which is the optimal solution of $\mathbf{LP}_{\omega t, \omega(1-t), 1}$ with $\omega > 0$ and $0 \leq t \leq 1$. Note that for fixed $\omega > 0$ and $0 \leq t \leq 1$, $\mathbf{LP}_{\omega t, \omega(1-t), 1}$ may be solved via the parametric programming problem

$LP'_{\omega,t} = \max \sum z_i(x_i - \omega[tK_i + (1-t)M_i])$

subject to

$0 \leq z_i \leq 1, \quad i \in E$.

Next, fix t, $0 \leq t \leq 1$ and let $(\alpha_1, \ldots, \alpha_n)$ be a permutation of E which satisfies (8). Note that for $\omega \geq 0$, the largest solution $z(\omega)$ of $\mathbf{LP}'_{\omega,t}$ is given by $z(\omega)_i = 1$ if $x_i \geq \omega(tK_i + (1-t)M_i)$ and $z(\omega)_i = 0$ otherwise. Define

$$[K, M, x](\omega) = \sum_{i \in E} z(\omega)_i(K_i, M_i, x_i)$$

and note that

$$Z' = \{[K, M, x](\omega): \omega \geq 0\} = \left\{\left(\sum_{l=1}^{j} K_{\alpha_l}, \sum_{l=1}^{j} M_{\alpha_l}, \sum_{l=1}^{j} x_{\alpha_l}\right): j = 1, \ldots, n\right\}.$$

It thus suffices to test whether the n points in Z' lie in Ω. This test corresponds with (8).

Lemma A1 implies that there are at most $0(n^2)$ different permutations to be tested: When t is gradually increased from 0 to 1 a new permutation is encountered only at those values of t, $0 \leq t \leq 1$, for which

$$[tK_l + (1-t)M_l]/x_l = [tK_m + (1-t)M_m]/x_m$$

for some pair (m, l), $m < l$. This occurs for at most $n(n-1)/2$ critical values of t. These critical values determine the endpoints of at most $n(n-1)/2 + 1$ subintervals of [0, 1] such that on each subinterval a given permutation satisfies (A.3).

Determining and sorting these critical values can be done in $0(n^2 \log n)$ time. Let $\alpha = (\alpha_1, \ldots, \alpha_n)$ be the permutation satisfying (A.2) on a given subinterval and let

$$x^{(j)} = \sum_{l=1}^{j} x_{\alpha_l}, \; K^{(j)} = \sum_{l=1}^{j} K_{\alpha_l}$$

and $M^{(j)} = \sum_{l=1}^{j} M_{\alpha_l}$ for $j = 1, \ldots, n$. The permutation $\underline{\alpha}'$ which prevails on the next subinterval differs from α by just one interchange of two neighboring components, say the jth and $j+1$st one. Thus, all partial sums $x^{(r)}$, $K^{(r)}$, $M^{(r)}$ with $r \neq j$ remain unaltered, and $x^{(j)}$, $K^{(j)}$ and $M^{(j)}$ may be updated by three additions and three subtractions only:

$$x^{(j)} := x^{(j)} + x_{\alpha_{j+1}} - x_{\alpha_j}; \; K^{(j)} = K^{(j)} + K_{\alpha_{j+1}} - K_{\alpha_j};$$

$$M^{(j)} := M^{(j)} + M_{\alpha_{j+1}} - M_{\alpha_j}.$$

Thus, if x satisfies (A.2) for the permutation $\underline{\alpha}$, it does so for permutation $\underline{\alpha}'$ as well provided the jth inequality in (A.2) holds. To test the latter one updates $x^{(j)}$, $K^{(j)}$ and $M^{(j)}$ and performs the single test $f(K^{(j)}, M^{(j)}) \geq x^{(j)}$. Thus, the incremental amount of work on each new subinterval consists of 0(1) operations and a single evaluation of the function f.

We conclude that the entire membership test for generalized symmetric polymatroids of order 2 takes a total of $0(n^2 \log n)$ operations.

REFERENCES

AHUJA, R., J. ORLIN, C. STEIN AND R. TARJAN. 1988. Improved Algorithms for Bipartite Network Flow Problems (to appear).

ANDRES, F., AND H. EMMONS. 1975. A Multiproduct Inventory System With Interactive Setup Cost. *Mgmt. Sci.*, 1055–1063.

→ BIXBY, R. E., W. H. CUNNINGHAM, AND D. M. TOPKIS. 1985. The Partial Order of a Polymatroid Extreme Point. *Math. O. R.* **10,** 367–378.

CUNNINGHAM, W. H. 1986. On Submodular Function Minimization. *Combinatorica* **5**(3), 185–192.

EDMONDS, J. 1970. Submodular Functions, Matroids and Certain Polyhedra. In *Combinatorial Structures and Their Applications*, R. Guy et al. (eds.). Gordon and Breach, New York, 69–87.

FEDERGRUEN, A., AND H. GROENEVELT. 1987. Polymatroidal Flow Network Models with Multiple Sinks. *Networks* **18,** 285–302.

→ FEDERGRUEN, A., AND H. GROENEVELT. 1988. Characterization and Control of Achievable Performance in General Queueing Systems. *Opns. Res.* **36,** 733–741.

FEDERGRUEN, A., AND Y.-S. ZHENG. 1991. Efficient Algorithms for Finding Optimal Power-of-Two Policies for Production/Distribution Systems With General Joint Setup Costs (unabridged version of this paper). Working Papers, Graduate School of Business, Columbia University, New York.

→ FEDERGRUEN, A., AND Y.-S. ZHENG. 1992. The Joint Replenishment Problem With General Joint Cost Structures: General Solution Methods and Performance Bounds. *Opns. Res.* **40,** 384–404.

→ FEDERGRUEN, A., AND Y.-S. ZHENG. 1993. Optimal Power-of-Two Replenishment Strategies in Capacitated General Production/Distribution Networks. *Mgmt. Sci.* **39,** 710–727.

→ FEDERGRUEN, A., M. QUEYRANNE AND Y.-S. ZHENG. 1992. Simple Power of Two Policies Are Close to Optimal in a General Class of Production/Distribution Networks

With General Joint Setup Costs. *Math. O. R.* **17,** 951–963.

GRÖTSCHEL, M., L. LOVASZ AND A. SCHRYVER. 1981. The Ellipsoid Method and Its Consequences in Combinatorial Optimization. *Combinatorica* **1,** 169–197.

GUSFIELD, D., C. MARTEL AND D. FERNANDEZ-BACA. 1985. Fast Algorithms for Bipartite Network Flow. Working Paper, Department of Computer Science, Yale University, New Haven, Conn.

HASSIN, R. 1982. Minimum Cost Flow in the Set-Constraints. *Networks* **12,** 1–21.

HERER, Y., AND R. ROUNDY. 1990. Heuristics for a One Warehouse Multi-retailer Distribution Problem With Performance Bounds. Technical Report No. 916, School of Operations Research, Cornell University, Ithaca, N.Y.

JACKSON, P. L., W. L. MAXWELL AND J. A. MUCKSTADT. 1988. Determining Optimal Intervals in Capacitated Production/Distribution Systems. *Mgmt. Sci.* **35,** 938–958.

LAWLER, E. L., AND C. U. MARTEL. 1982. Computing Maximal 'Polymatroidal' Network Flows. *Math. O. R.* **7,** 334–346.

MAXWELL, W. L., AND J. A. MUCKSTADT. 1985. Establishing Consistent and Realistic Reorder Intervals in Production-Distribution Systems. *Opns. Res.* **33,** 1316–1341.

MUCKSTADT, J., AND R. ROUNDY. 1993. Analysis of Multi-Stage Production Systems. In *Handbooks in Operations Research*, Vol. 4, S. Graves, A. Rinnooy Kan and P. Zipkin (eds.). North–Holland, Amsterdam.

QUEYRANNE, M. 1985. A Polynomial-Time Submodular Extension to Roundy's 98% Effective Heuristic for Production/Inventory Systems. Working Paper no. 1136, University of British Columbia, Vancouver, B. C., Canada.

ROUNDY, R. 1985. 98%-Effective Integer-Ratio Lot-Sizing for One-Warehouse Multi-Retailer Systems. *Mgmt. Sci.* **31,** 1416–1430.

ROUNDY, R. 1986. A 98% Effective Lot-Sizing Rule for a Multi-Product, Multi-Stage Production Inventory Systems. *Math. O. R.* **11,** 699–727.

TARDOS, E., C. TOVEY AND M. TRICK. 1986. Layered Augmented Path Algorithms. *Math. O. R.* **11,** 362–370.

TOPKIS, D. M. 1984. Adjacency on Polymatroids. *Math. Prog.* **30,** 229–237.

ZHENG, Y.-S. 1987. Replenishment Strategies for Production/Distribution Networks with General Joint Setup Costs. Ph.D. Dissertation, Columbia University, New York.