Minimum Cost Flow With Set-Constraints

Refael Hassin
Department of Statistics, Tel-Aviv University, Ramat-Aviv, Tel-Aviv, Israel

The minimum cost network flow problem with set-constraints is a generalization of the well-known minimum cost network flow problem, in which bounds on the sum of flows through sets of arcs exist. This paper investigates some variations of this problem, including the polymatroid intersection problem, where for each node two polymatroids are given; one polymatroid constrains flows entering the node, and the other constrains flows leaving it.

I. INTRODUCTION

The minimum cost network flow problem with set-constraints is a generalization of the well-known minimum cost network flow problem, in which bounds on the sum of flows through sets of arcs exist. This paper investigates some variations of this problem where each node has two polymatroids, one constrains flows entering the node, and the other constrains flows leaving it. Notation and fundamental concepts are defined in Sec. II. The algorithm is first presented and motivated for a limited case in Sec. III. Sec. IV describes how the dual problem can be decomposed into trivial subproblems. Sec. V proves some properties of submodular and supermodular functions and uses them to prove an existence theorem for flows with set-constraints. In Sec. VI, this theorem is used to generalize the algorithm of Sec. III. Finally, in Sec. VII, we set a condition for the existence of a trivial solution to a polymatroid intersection problem in which the costs are not restricted in sign.

II. NOTATION AND TERMINOLOGY

Throughout this paper n will be a positive integer, E will be the set $\{1, \ldots, n\}$, and $R_+^n = \{x : x \in R^n \text{ and } x \geqslant 0\}$. A real valued function r whose domain is all of the subsets of E is said to be *submodular* if

$$r(S) + r(T) \geqslant r(S \cup T) + r(S \cap T) \qquad \forall S, T \subseteq E,$$

and *supermodular* if

$$r(S) + r(T) \leqslant r(S \cup T) + r(S \cap T) \qquad \forall S, T \subseteq E.$$

NETWORKS, Vol. 12 (1982) 1–21

CCC 0028-3045/82/010001-21$02.10

The function is said to be *nondecreasing* if

$$r(S) \leqslant r(T) \qquad \forall S \subseteq T \subseteq E.$$

A non-negative nondecreasing function $r \not\equiv 0$ which is submodular is called a β_0 *function.* If it is supermodular we call it a γ_0 *function.*

For the following linear program, there exists an immediate solution called the *greedy algorithm* [2, 3].

$$\text{Minimize} \sum_S r(S)\, u(S) \tag{1}$$

subject to

$$\sum_{S:\, j \in S} u(S) \geqslant c_j \qquad \forall j \in E,$$

where $c_1 \geqslant c_2 \geqslant \cdots \geqslant c_k > 0 \geqslant c_{k+1} \geqslant \cdots \geqslant c_n$, and r is a β_0 function defined on the subsets of E. The solution is

$$\begin{aligned} u(\{1,\ldots,t\}) &= c_t - c_{t+1}, && \text{for } t = 1,\ldots,k-1; \\ u(\{1,\ldots,k\}) &= c_k; && \\ u(S) &= 0, && \text{for all other } S \subseteq E. \end{aligned} \tag{2}$$

A *polymatroid* is a set $P \subseteq R_+^n$ such that

(i) $0 \leqslant x^0 \leqslant x^1$ and $x^1 \in P$ imply $x^0 \in P$;

(ii) for $\alpha \in R_+^n$, every $x \in P$ such that $x \leqslant \alpha$ and no $x' \in P$ such that $x' > x$ exists, has the same component sum $\Sigma_{j=1}^n x_j$ called the *rank* of α, $r(\alpha)$.

A polymatroid is called *integral* if (ii) holds also when α and x are restricted to be integer valued. It is a *matroid* when the vectors are restricted to be 0-1 valued.

It has been shown [2] that the rank function $r(\alpha)$ for any polymatroid is a β_0 function, and that if r is a β_0 function on 2^E then $\{x \in R_+^n\colon \Sigma_{j \in S}\, x_j \leqslant r(S)\ \forall S \subseteq E\}$ is a polymatroid.

Problem (3) is the dual linear program of problem (1):

$$\text{Maximize} \sum_{j \in E} c_j x_j \tag{3}$$

subject to

$$\sum_{j \in S} x_j \leqslant r(S) \qquad \forall S \subseteq E,$$

$$x_j \geqslant 0 \qquad \forall j \in E.$$

Since r is a β_0 function, this problem is to find a maximum weighted vector in the polymatroid defined by r.

Let r and r' be two β_0 functions defined on 2^E. The *polymatroid intersection* problem is to find the maximum weighted vector which belongs to both polymatroids defined by r and r', i.e.,

$$\text{maximize} \sum_{j \in E} c_j x_j \tag{4}$$

subject to

$$\sum_{j \in S} x_j \leqslant r(S) \qquad \forall S \subseteq E,$$

$$\sum_{j \in S} x_j \leqslant r'(S) \qquad \forall S \subseteq E,$$

$$x_j \geqslant 0 \qquad \forall j \in E.$$

Certain problems of matching, job sequencing, experimental design, network synthesis and information theory can be formulated in this form [6, 11]. Algorithms which solve problem 4 and its variations are described in [4, 5, 9, 10].

In the following, (N, A) will denote a directed network with a set of nodes N and a set of arcs $A \subseteq N \times N$. The flow assigned to arc $(i, j) \in A$ will be denoted by x_{ij}, and c_{ij} will denote its unit cost. We also use the notation $x(i, S) = \Sigma_{j \in S}\, x_{ij}$ and $x(S, i) = \Sigma_{j \in S}\, x_{ji}$.

The minimum cost flow problem is a well-known problem in network theory. The problem is to

$$\text{minimize} \sum_{(i,j) \in A} c_{ij} x_{ij} \tag{5}$$

subject to

$$x(i, N - i) - x(N - i, i) = 0 \qquad \forall\, i \in N,$$

$$0 \leqslant x_{ij} \leqslant r_{ij} \qquad \forall\, (i, j) \in A,$$

where r_{ij} $(i, j) \in A$ is a set of given capacities of the arcs of the network.

Problems (3) and (4) and problem (5) are of theoretical interest and have useful applications. It would therefore seem that following a combination of the above, formulated in (6) below, could also be of much interest.

$$\text{Maximize}^* \sum_{(i,j) \in A} c_{ij} x_{ij} \tag{6}$$

*Network flow theory usually deals with "minimization" problems, while matroid intersection is usually defined as a "maximization" problem. We chose to define the problem in terms of maximization. Clearly this is just a matter of notation.

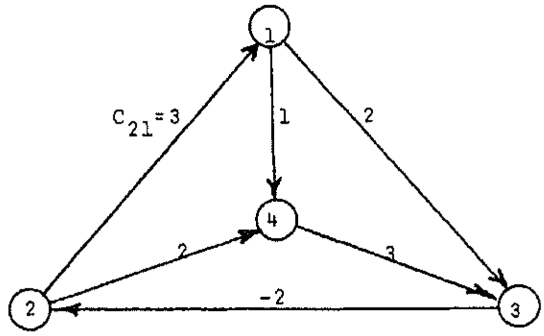


FIG. 1.

subject to

$$
\begin{aligned}
&x(i,N-i)-x(N-i,i)=0 && \forall i\in N,\\
&x(i,F)\leqslant r^+(i,F) && \forall i\in N\ F\subseteq N-i,\\
&x(F,i)\leqslant r^-(F,i) && \forall i\in N\ F\subseteq N-i,\\
&x_{ij}\geqslant 0 && \forall (i,j)\in A,
\end{aligned}
$$

where $r^+(i,\cdot)$ and $r^-(\cdot,i)$ are given β_0 functions defined on the subsets of $N-i$. Thus for each node two polymatroids are defined, one constraining the flows entering the node and the other constraining the flows leaving it.

We note that the polymatroid intersection problem (4) is a special case of (6). In this case the network consists of nodes $E\cup\{s,t\}$ and arcs $(t,s),(s,1),\ldots,(s,n),(1,t),\ldots,(n,t)$. The upper bounds on the flows are $r^+(s,S)=r(S)$ and $r^-(S,t)=r'(S)$ for every $S\subseteq E$, and $r^+(t,\{s\})=r^-(\{t\},s)=\infty$.

The following example illustrates our problem and will be used later to clarify the algorithm which we develop.

Example. (N,A) is the network presented in Figure 1 where the numbers written on the arcs are the unit flow costs, and the flow is constrained as follows:

(i) Bounds on flows leaving the nodes:
$x_{13}\leqslant 3, x_{14}\leqslant 5, x_{13}+x_{14}\leqslant 6;$
$x_{21}\leqslant 8, x_{24}\leqslant 4, x_{21}+x_{24}\leqslant 9;$
$x_{32}\leqslant 8;$
$x_{43}\leqslant 5.$

(ii) Bounds on flows entering the nodes:
$x_{21}\leqslant 8;$
$x_{32}\leqslant 8;$
$x_{13}\leqslant 3, x_{43}\leqslant 5, x_{13}+x_{43}\leqslant 9;$
$x_{14}\leqslant 5, x_{24}\leqslant 4, x_{14}+x_{24}\leqslant 6.$

II. POLYMATROID INTERSECTION

In this section we present an algorithm for the polymatroid intersection problem (4). We present it now in order to motivate the general algorithm for problem (6) which we develop later, and therefore the details and proofs are omitted. The main idea underlying the algorithm is that if x is a vector which maximizes both

$$\sum_{i \in E} v_i x_i \tag{7a}$$

subject to

$$\sum_{i \in S} x_i \leqslant r(S) \qquad \forall S \subseteq E,\ x \geqslant 0$$

and

$$\sum_{i \in E} v_i' x_i \tag{7b}$$

subject to

$$\sum_{i \in S} x_i \leqslant r'(S) \qquad \forall S \subseteq E,\ x \geqslant 0$$

for some pair of vectors v and v' such that $v + v' = c$, then x is also an optimal solution to (4). Note that each of the above problems is easily solved by the greedy algorithm.

Polymatroid Intersection Algorithm

Step 1: Start with any pair of vectors v and v' such that $v + v' = c$.

Step 2: Try to find a vector x which solves both (7a) and (7b). If one exists it is the optimal solution. Else there exists $M \subseteq E$ such that $\Sigma_{i \in M} x_i$ is strictly greater in any solution of one problem [say (7a)] than in any solution of the other problem.

Step 3: Equally decrease v_i and increase v_i' for every $i \in M$ until for some $i \in M$, v_i (or v_i') becomes either zero or equal to some $v_j(v_j')$, $j \notin M$. Return to Step 2.

We apply now the algorithm to an example of matroid intersection taken from [10]. Let $r(S)$ and $r'(S)$ be the number of arcs in a spanning tree of the subgraph induced by the arcs of S in the graphs G and G' in Figure 2, respectively. Let the costs be $c_1 = 3$, $c_2 = 5$, $c_3 = 6$, $c_4 = 10$, and $c_5 = 8$.

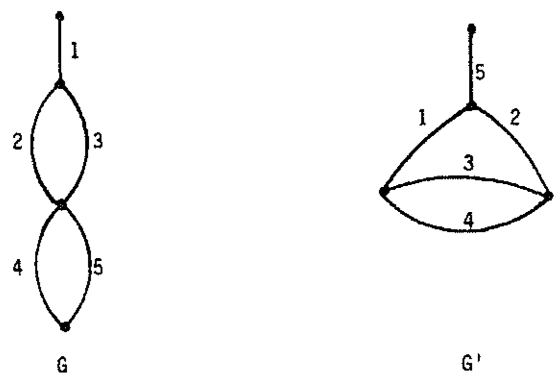


FIG. 2.

TABLE I

Iteration	1		2		3	
i	v	v'	v	v'	v	v'
1	3	0	3	0	2	1
2	5	0	5	0	4	1
3	6	0	5	1	4	2
4	10	0	9	1	8	2
5	8	0	8	0	8	0

We start with $v = c$ and $v' = 0$. The greedy algorithm applied to (7a) gives $x = (1, 0, 1, 1, 0)$ but (7b) requires $x_3 + x_4 \leqslant 1$, thus we reach Step 3 with $M = \{3, 4\}$. We decrease v_3 and v_4 and increase v'_3 and v'_4 until $v_2 = v_3$. The greedy algorithm now yields for (7a) $x = (1, x_2, x_3, 1, 0)$ with $x_3 = 1 - x_2$ and $x_2 \in \{0, 1\}$. However in (7b) $x_1 + x_2 + x_3 + x_4 \leqslant 2$ is required and thus $M = \{1, 2, 3, 4\}$. Another change in v and v' yields the solution $x = (1, 0, 1, 0, 1)$ which solves both (7a) and (7b) and is therefore optimal. The Table I summarizes the calculations.

IV. THE DUAL RESTRICTED PROBLEM

The dual problem of (6) is to

$$\text{Minimize} \sum_{i \in N} \sum_{F \subseteq N-i} [r^+(i, F)\, u^+(i, F) + r^-(F, i)\, u^-(F, i)] \tag{8}$$

subject to

$$u_i - u_j + \sum_{\substack{F \subseteq N-i \\ j \in F}} u^+(i, F) + \sum_{\substack{F \subseteq N-j \\ i \in F}} u^-(F, j) \geqslant c_{ij} \qquad \forall (i, j) \in A,$$

$$u^+(i, F), \;\; u^-(F, i) \geqslant 0 \qquad \forall i \in N, F \subseteq N - i.$$

To allow a decomposition of the dual problem we first transform every arc $(i, j) \in A$ into two arcs (i, m) and (m, j), where m is a slack node, and assign to these arcs any costs c_{im} and c_{mj} such that their sum $c_{im} + c_{mj}$ equals the original cost of the arc, c_{ij}. Figure 3 shows the transformed network for the example presented in Figure 1.

The nodes of the new network consist of the original set of nodes N with a set of slack nodes N_0, and the arcs of the new network consist of two sets of arcs $A_1 \subseteq N \times N_0$ and $A_2 \subseteq N_0 \times N$. The dual problem (8) is now reformulated:

$$\text{Minimize} \sum_{i \in N} \sum_{F \subseteq N-i} [r^+(i, F)\, u^+(i, F) + r^-(F, i)\, u^-(F, i)] \tag{9}$$

subject to

$$u_i - u_j + \sum_{\substack{F \subseteq N-i \\ j \in F}} u^+(i, F) \geqslant c_{ij} \qquad \forall (i, j) \in A_1,$$

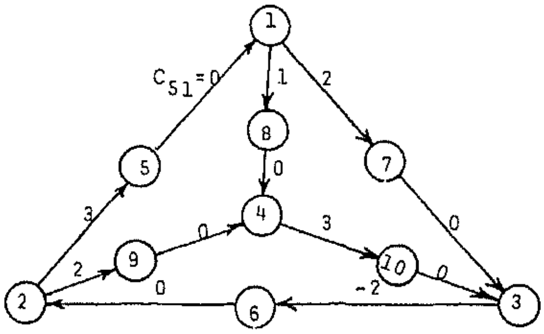


FIG. 3.

$$u_i - u_j + \sum_{\substack{F \subseteq N-j \\ i \in F}} u^-(F, j) \geqslant c_{ij} \qquad \forall (i, j) \in A_2,$$

$$u^+(i, F), \;\; u^-(F, i) \geqslant 0 \qquad \forall i \in N, F \subseteq N - i.$$

Suppose that we fix the values of the node variables $u_i\, i \in N$, thus forming a "restricted" dual problem, and let $v_{ij} = c_{ij} - u_i + u_j$. The problem reduces to $2|N|$ independent subproblems as follows:

$$(P_i^+) \text{ Minimize } z_i^+ = \sum_{F \subseteq N-i} r^+(i, F)\, u^+(i, F)$$

subject to

$$\sum_{\substack{F \subseteq N-i \\ j \in F}} u^+(i, F) \geqslant v_{ij} \qquad \forall (i, j) \in A_1,$$

$$u^+(i, F) \geqslant 0 \qquad \forall F \subseteq N - i.$$

$$(P_i^-) \text{ Minimize } z_i^- = \sum_{F \subseteq N-i} r^-(F, i)\, u^-(F, i)$$

subject to

$$\sum_{\substack{F \subseteq N-i \\ j \in F}} u^-(F, i) \geqslant v_{ji} \qquad \forall (j, i) \in A_2,$$

$$u^-(F, i) \geqslant 0 \qquad \forall F \subseteq N - i.$$

Each of these problems is solved by the "greedy algorithm." To solve (P_i^+) let $j(1)$, $j(2), \ldots,$ be an ordering of $j \in N - i$ such that $v_{i,j(1)} \geqslant v_{i,j(2)} \geqslant \cdots \geqslant v_{i,j(k)} > 0 \geqslant v_{i,j(k+1)} \geqslant \cdots$. Let $G_t = \{j(1), j(2), \ldots, j(t)\}$ for $1 \leqslant t \leqslant k$. Then the following solution is optimal:

$$u^+(i, G_t) = v_{i,j(t)} - v_{i,j(t+1)} \qquad \text{for } t = 1, \ldots, k-1, \tag{10a}$$

$$u^+(i, G_k) = v_{i,j(k)}, \tag{10b}$$

$$u^+(i, F) = 0 \qquad \text{for all other } F \subseteq N - i. \tag{10c}$$

Similar solutions can be obtained for the problems (P_i^-). In the following until we solve the example, we simplify the discussion by assuming that $r^-(F, i) = \infty$ for all $i \in N$ and $F \subseteq N - i$, so that the problems (P_i^-) can be ignored and the transformation of the network is unnecessary. We also simplify the notation by omitting the + sign from $r^+(i, F)$, $u^+(i, F)$, and z_i^+.

For a given set u_i $i \in N$ we obtained from (P_i^+) and (10)

$$z_i = \sum_{t=1}^{k-1} r(i, G_t)\,(v_{i,j(t)} - v_{i,j(t+1)}) + r(i, G_k)\, v_{i,j}(k). \tag{11}$$

Suppose that either $t = 1$ or $v_{i,j(t)} < v_{i,j(t-1)}$. Let $a_{i,j(t)}$ be the increase in z_i per unit increase in $v_{i,j(t)}$. Then

$$a_{i,j(t)} = \begin{cases} r(i, G_t) - r(i, G_{t-1}) & \text{if } v_{i,j(t)} \geqslant 0 \text{ and } t \neq 1, \\ r(i, G_1) & \text{if } v_{i,j(t)} \geqslant 0 \text{ and } t = 1, \\ 0 & \text{if } v_{i,j(t)} < 0. \end{cases}$$

Suppose that either $t = |N| - 1$ or $v_{i,j(t)} > v_{i,j(t+1)}$. Let $b_{i,j(t)}$ be the decrease in the objective per unit decrease in $v_{i,j(t)}$. Then,

$$b_{i,j(t)} = \begin{cases} r(i, G_t) - r(i, G_{t-1}) & \text{if } v_{i,j(t)} > 0 \text{ and } t \neq 1, \\ r(i, G_1) & \text{if } v_{i,j(t)} > 0 \text{ and } t = 1, \\ 0 & \text{if } v_{i,j(t)} \leqslant 0. \end{cases}$$

Note that if $v_{i,j(t+1)} < v_{i,j(t)} < v_{i,j(t-1)}$ then $a_{i,j(t)} = b_{i,j(t)}$ and otherwise $a_{i,j(t)} \geqslant b_{i,j(t)}$.

Suppose that $v_{ij} = v$ for all $j \in S \subseteq N - i$. A simultaneous increase in all u_j $j \in S$ will result in a similar increase in v_{ij} for all $j \in S$. Let $B = \{j \in N - i\colon v_{ij} > v\}$, then from (11) we obtain that the increase in z_i per unit increase in u_j $j \in S$ is

$$a(i, S) = r(i, B \cup S) - r(i, B). \tag{12}$$

Similarly, let $D = \{j \in N - i\colon v_{ij} \geqslant v\}$, then the decrease in z_i per unit decrease in u_j $j \in S$ is

$$b(i, S) = r(i, D) - r(i, D - S). \tag{13}$$

Note that $a(i, S)$ and $b(i, S)$ are the values obtained while considering the arcs (i, j) $j \in S$ as a single arc.

Example. Let $N = \{0, 1, 2, 3, 4, 5\}$ and suppose $v_{01} > v_{02} = v_{03} = v_{04} = v > v_{05} \geqslant 0$, then z_0 is equal to

$$\zeta = r(0, \{1\})\,(v_{01} - v) + r(0, \{1, 2, 3, 4\})\,(v - v_{05}).$$

If v_{02} and v_{03} are increased to $v' \leqslant v_{01}$ then the new value of z_0 is

$$\zeta' = r(0, \{1\})(v_{01} - v') + r(0, \{1, 2, 3\})(v' - v) + r(0, \{1, 2, 3, 4\})(v - v_{05}).$$

Thus

$$\zeta' - \zeta = (v' - v)\,[r(0, \{1, 2, 3\}) - r(0, \{1\})]$$

and

$$a(0, \{2, 3\}) = r(0, \{1, 2, 3\}) - r(0, \{1\}).$$

On the other hand, if v_{02} and v_{03} are decreased to $v'' \geqslant v_5$, then the new value of z_0 is

$$\zeta'' = r(0, \{1\})(v_{01} - v) + r(0, \{1, 4\})(v - v'') + r(0, \{1, 2, 3, 4\}\ (v'' - v_{05}),$$

$$\zeta - \zeta'' = (v - v'')\,(r(0, \{1, 2, 3, 4\}) - r(0, \{1, 4\})$$

and thus

$$b(0, \{2, 3\}) = r(0, \{1, 2, 3, 4\}) - r(0, \{1, 4\}).$$

V. AN EXISTENCE THEOREM

The existence theorem which we prove in this section underlies the algorithm developed in Sec. VI.

Lemma 1. For every $i \in N$, $a(i, S)$ is submodular and $b(i, S)$ is supermodular.

Proof: For simplicity we drop the index i. Let $S \subseteq N$, and $T \subseteq N$. Since the effect of changing v_{ij}'s with distinct values is additive, we just have to consider the case in which $v_{ij} = v$ for all $j \in S \cup T$. Let $B = \{j : v_{ij} > v\}$ and let $D = \{j : v_{ij} \geqslant v\}$. Using the submodularity of $r(\cdot)$ and eqs. (12)-(13), we obtain

$$\begin{aligned} a(S) + a(T) &= r(B \cup S) + r(B \cup T) - 2r(B) \\ &\geqslant r((B \cup S) \cup (B \cup T)) + r((B \cup S) \cap (B \cup T)) - 2r(B) \\ &= r(B \cup S \cup T) + r(B \cup (S \cap T)) - 2r(B) = a(S \cup T) + a(S \cap T), \\ b(S) + b(T) &= 2r(D) - [r(D - S) + r(D - T)] \\ &\leqslant 2r(D) - [r((D - S) \cup (D - T)) + r((D - S) \cap (D - T))] \\ &= 2r(D) - [r(D - (S \cap T)) + r(D - (S \cup T))] = b(S \cap T) + b(S \cup T). \end{aligned}$$

We define the *sum* $S + T$, of the sets $S = \{s_1, \ldots, s_p\}$ and $T = \{t_1, \ldots, t_q\}$ as $S + T = \{s_1, \ldots, s_p, t_1, \ldots, t_p\}$ (we do not require $S \cap T = \phi$).

Lemma 2. Let r be a real function on 2^E. Suppose that sets $H_p \subseteq E$, $p = 1, \ldots, \bar{p}$ are given. Let $S_q = \{e \in E$: e is included in at least q of the sets H_p, $p = 1, \ldots, \bar{p}\}$.

(a) If r is submodular, then $\Sigma_p\, r(H_p) \geqslant \Sigma_q\, r(S_q)$.

(b) If r is supermodular, then $\Sigma_p\, r(H_p) \leqslant \Sigma_q\, r(S_q)$.

Proof: Note that $\Sigma_q S_q = \Sigma_p H_p$. Construct the sets S_q from the sets H_p by successively replacing pairs of sets H_i, H_j such that $H_i \not\subseteq H_j$ and $H_j \not\subseteq H_i$, by $H_i \cup H_j$ and $H_i \cap H_j$. This operation leaves $\Sigma_p H_p$ unchanged and stops when the sets H_p are nested and hence constitute a permutation of the sets S_q.

If r is submodular, this procedure decreases $\Sigma_p r(H_p)$ since $r(H_i) + r(H_j) \geqslant r(H_i \cup H_j) + r(H_i \cap H_j)$. If r is supermodular, this sum is increased.

Example. Suppose r is submodular, then

$$2r(\{1, 2, 3\}) + r(\{1\}) \leqslant \begin{cases} r(\{1, 2\}) + r(\{2, 3\}) + r(\{1, 3\}) + r(1), \\ 3r(\{1\}) + 2r(\{2\}) + 2r(\{3\}), \\ 2r(\{1, 2\} + r(\{1, 3\}) + r(\{3\}). \end{cases}$$

Lemma 3. For every node $i \in N$ and sets $S, T \subseteq N - i$,

$$a(i, S) - a(i, S - T) \geqslant b(i, T) - b(i, T - S).$$

Proof: For simplicity we drop the index i. Set $S_v = \{j \in S\colon v_j = v\}$ and $T_v = \{j \in T\colon v_j = v\}$, then

$$a(S) = \sum_v a(S_v), \quad a(S - T) = \sum_v a(S_v - T_v)$$

$$b(T) = \sum_v b(T_v - S_v), \quad b(T - S) = \sum_v b(T_v - S_v)$$

where the sums are taken over all distinct values of v_j. Thus

$$[a(S) - a(S - T)] - [b(T) - b(T - S)] = \sum_v \{[a(S_v) - a(S_v - T_v)] - [b(T_v) - b(T_v - S_v)]\}.$$

Let $B_v = \{j\colon v_j > v\}$ and let $D_v \{j\colon v_j \geqslant v\}$. From (12) and (13) we obtain

$$a(S_v) = r(B_v \cup S_v) - r(B_v),$$
$$a(S_v - T_v) = r(B_v \cup (S_v - T_v)) - r(B_v),$$
$$b(T_v) = r(D_v) - r(D_v - T_v),$$
$$b(T_v - S_v) = r(D_v) - r(D_v - (T_v - S_v)).$$

Therefore

$$[a(S) - a(S - T)] - [b(T) - b(T - S)] = \sum_v \{[r(B_v \cup S_v) + r(D_v - T_v)] - [r(D_v - (T_v - S_v)) + r(B_v \cup (S_v - T_v))]\}$$

Since the set in the third term is the union of the sets in the first two terms, while the set in the last term is their intersection, and since r is submodular, the whole term is non-negative.

Lemma 4. Let r and r' be two real functions on 2^E. Assume that r is submodular, r' is supermodular, and $r(S) - r(S - T) \geqslant r'(T) - r'(T - S)$ for every $S \subseteq E$ and $T \subseteq E$. Let $H_p \subseteq E$ $p = 1, \ldots, \bar{p}$, and $G_m \subseteq E$ $m = 1, \ldots, \bar{m}$ be given sets. For $q = 1, 2, \ldots$, define $S_q = \{e \in E$: e is included in the sets H_p $p = 1, \ldots, \bar{p}$, at least q times *more* than in the sets G_m $m = 1, \ldots, \bar{m}\}$, and $S'_q = \{e \in E$: e is included in the sets H_p $p = 1, \ldots, \bar{p}$, at least q times *less* than in the sets G_m $m = 1, \ldots, \bar{m}\}$. Then

$$\sum_p r(H_p) - \sum_m r'(G_m) \geqslant \sum_q r(S_q) - \sum_q r'(S'_q).$$

Proof: Suppose $H_p \cap G_m = \phi$ for $p = 1, \ldots, \bar{p}$ and $m = 1, \ldots, \bar{m}$. By Lemma 2 $\Sigma_p\, r(H_p) \geqslant \Sigma_q\, r(S_q)$ and $\Sigma_m r'(G_m) \leqslant \Sigma_q\, r'(S'_q)$. The subtraction of the last expression from the first yields the result for this case.

Suppose $H_i \cap G_j \neq \phi$. Since $r(H_i) - r'(G_j) \geqslant r(H_i - G_j) - r'(G_j - H_i)$, replacing H_i by $H_i - G_j$, and G_j by $G_j - H_i$, decreases the left-hand side of the inequality. This procedure may be repeated until $H_p \cap G_m = \phi$ for $p = 1, \ldots, \bar{p}$ and $m = 1, \ldots, \bar{m}$, at which point Lemma 2 applies.

Example. Let $H_1 = \{1, 2, 3\}$, $H_2 = \{1, 3\}$, $H_3 = \{2, 3, 5\}$, $G_1 = \{1, 2, 5\}$, $G_2 = \{4, 5\}$, $G_3 = \{2, 4\}$, and $G_4 = \{5\}$. Then, $\Sigma_p\, H_p = \{1, 1, 2, 2, 3, 3, 3, 5\}$ and $\Sigma_m\, G_m = \{1, 2, 2, 4, 4, 5, 5, 5\}$. Therefore $S_1 = \{1, 3\}$, $S_2 = S_3 = \{3\}$, $S'_1 = S'_2 = \{4, 5\}$. If r and r' satisfy the conditions required by Lemma 4, then

$$r(\{1, 2, 3\}) + r(\{1, 3\}) + r(\{2, 3, 5\}) - r(\{1, 2, 5\}) - r(\{4, 5\}) - r(\{2, 4\}) - r(\{5\}) \geqslant r(\{1, 3\}) + 2r(\{3\}) - 2r(\{4, 5\}).$$

The following theorem is a generalization of Hoffman's existence theorem for circulations [8, 11].

Theorem 1. For every $i \in N$ let $k(i, \cdot)$ be a submodular function and $d(i, \cdot)$ be a supermodular function such that $k(i, S) - k(i, S - T) \geqslant d(i, T) - d(i, T - S)$ for any S, $T \subseteq N - i$. A necessary and sufficient condition for the existence of $x \geqslant 0$ such that $x(i, N - i) = x(N - i, i)$ and $d(i, S) \leqslant x(i, S) \leqslant k(i, S)$ for every $i \in N$ and $S \subseteq N$ is that

$$\sum_{i \in M} k(i, N - M) \geqslant \sum_{i \in N-M} d(i, M) \qquad \forall M \subseteq N.$$

Proof: (a) The condition is necessary since for a feasible circulation

$$\sum_{i \in N-M} d(i, M) \leqslant \sum_{i \in N-M} x(i, M) = \sum_{i \in M} x(i, N - M) \leqslant \sum_{i \in M} k(i, N - M).$$

(b) Suppose that no feasible circulation exists. Then there exists a circulation y with $y(i, S) \leqslant k(i, S)$ for every $S \subseteq A$, a node $m \in N$ and a set $Q \subseteq N - m$, such that $y(m, Q) < d(m, Q)$ and the solution to the auxiliary problem,

$$\text{maximize } x(m, Q)$$

subject to

$$\begin{aligned} &x(i, N-i) - x(N-i, i) = 0 && \forall i \in N, \\ &x(i, S) \leqslant 0 && \text{if } y(i, S) = k(i, S), \\ &x(i, S) \geqslant 0 && \text{if } y(i, S) \leqslant d(i, S), \end{aligned}$$

is $x(m, Q) = 0$.

Thus there exist an integer L and a non-negative integral solution to the following dual system:

$$u_i - u_j + \sum_{S_j^+} u(i, S) - \sum_{S_j^-} u(i, S) = N_{ij} \qquad \forall (i, j) \in A$$

where

$$S_j^+ = \{S \subseteq N - i : j \in S, y(i, S) = k(i, S)$$

$$S_j^- = \{S \subset N - i : j \in S, y(i, S) \leqslant d(i, S)\}$$

$$N_{ij} = \begin{cases} L & i = m \text{ and } j \in Q \\ 0 & \text{otherwise.} \end{cases}$$

For $q = 0, 1, 2, \ldots$, let $T_q = \{i \in N : u_i \geqslant q\}$, then $(T_q, N - T_q) = \{(i, j) \in A : u_i \geqslant q, u_j < q\}$, and $(N - T_q, T_q) = \{(i, j) \in A : u_i < q, u_j \geqslant q\}$. Hence,

$$\begin{aligned} 0 &= \sum_q y(N - T_q, T_q) - \sum_q y(T_q, N - T_q) \\ &= \sum_{(i,j) \in A} (u_j - u_i)^+ y_{ij} - \sum_{(i,j) \in A} (u_i - u_j)^+ y_{ij} \\ &= \sum_{(i,j) \in A} (u_j - u_i) y_{ij} \\ &= \sum_{(i,j) \in A} \sum_{S_j^+} u(i, S) y_{ij} - \sum_{(i,j) \in A} \sum_{S_j^-} u(i, S) y_{ij} - \sum_{j \in Q} L y_{mj} \\ &= \sum_{i \in N} \sum_{S : y(i,S) = k(i,S)} u(i, S) y(i, S) - \sum_{i \in N} \sum_{S : y(i,S) \leqslant d(i,S)} u(i, S) y(i, S) \\ &\quad - L y(m, Q) > \sum_{i \in N} \sum_{S : y(i,S) = k(i,S)} u(i, S) k(i, S) \\ &\quad - \sum_{i \in N} \sum_{S : y(i,S) \leqslant d(i,S)} u(i, S) d(i, S) - L d(m, Q) \end{aligned}$$

(by Lemma 4)

$$\geqslant \sum_{i\in N}\sum_{q} k\left(i,\left\{j: \sum_{S_j^+} u(i,S) - \sum_{S_j^-} u(i,S) - N_{ij} \geqslant q\right\}\right)$$
$$- \sum_{i\in N}\sum_{q} d\left(i,\left\{j: \sum_{S_j^-} u(i,S) - \sum_{S_j^+} u(i,S) + N_{ij} \geqslant q\right\}\right)$$
$$= \sum_{i\in N}\sum_{q} k(i, \{j: u_j - u_i \geqslant q\}) - \sum_{i\in N}\sum_{q} d(i, \{j: u_i - u_j \geqslant q\})$$
$$= \sum_{q}\left[\sum_{i\in N: u_i < q} k(i, \{j: u_j \geqslant q\}) - \sum_{i\in N: u_i \geqslant q} d(i, \{j: u_j < q\})\right]$$
$$= \sum_{q}\left[\sum_{i\in N-T_q} k(i, T_q) - \sum_{i\in T_q} d(i, N - T_q)\right].$$

Thus, there exists q such that the condition of the theorem does not hold for the set T_q.

VI. AN ALGORITHM

An *improving set* is a set $M \subseteq N$ such that a simultaneous increase in all u_i $i \in M$ decreases the dual objective $\Sigma_{i\in N} z_i$. An increase in all u_i $i \in M$, equally increases v_{ij} for all $(i,j) \in (N - M, M)$ and decreases v_{ij} for all $(i,j) \in (M, N - M)$. Therefore a set $M \subseteq N$ is an improving set if and only if $\Sigma_{i\in N-M}\, a(i,M) < \Sigma_{i\in M}\, b(i, N - M)$.

Theorem 2. A dual solution u_i $i \in N$ is optimal if and only if no improving sets exist, i.e., $\Sigma_{i\in N-M}\, a(i,M) \geqslant \Sigma_{i\in M}\, b(i, N - M)$ for all $M \subseteq N$.

Proof: The condition is trivially necessary. To prove sufficiency we use Theorem 1. Let $a(i, S)$ and $b(i, S)$ be the marginal costs associated with a feasible dual solution, as defined in Sec. III. By Lemmas 1 and 3 and the assumption of this theorem, they satisfy the conditions stated in Theorem 1 for $k(i, S)$ and $d(i, S)$, respectively. Hence there exists a circulation x such that $b(i, S) \leqslant x(i, S) \leqslant a(i, S)$ for all $i \in N$, and $S \subseteq N - i$. By the complementary slackness theorem of linear programming, the solution u_i, $i \in N$ is optimal if $x(i, S) = r(i, S)$ whenever $u(i, S) > 0$. However, $u(i, S) > 0$ is possible in the solution to (P_i) only if $S = G_t$ and $v_{j(t)} > v_{j(t+1)}$. In such a case, $a(i, G_t) = b(i, G_t) = r(G_t)$. Thus $b(i, G_t) \leqslant x(i, G_t) \leqslant a(i, G_t)$ implies that $x(i, G_t) = r(i, G_t)$ as required. Therefore the complementary condition holds and u_i, $i \in N$ is an optimal solution for the dual problem (while the circulation x is optimal for the primal problem).

We are now in a position to describe, in general, the class of dual algorithms which solve our problem. Then we describe in more detail a primal dual variation of this class. Both the general and specific algorithms are modifications of algorithms which are described in [7] for the minimum cost flow problem.

A General Dual Algorithm

Step 1: Try to find an improving set $M \subseteq N$. If none exists, the solution is optimal, else, proceed to Step 2.

Step 2: Increase u_i, $i \in M$, until for some $(i, j) \in A$ either v_{ij} becomes zero or it becomes equal to v_{im} for $m \in N - M_u$, and proceed to Step 1. If this does not happen then the dual problem is unbounded and the primal is feasible.

While Step 2 of the general algorithm is common to all the dual algorithms [7] (including the so-called primal-dual) Step 1 is implemented in different ways.

We now describe a procedure which implements Step 1 of the general algorithm using both primal and dual variables. This procedure tries to construct a circulation satisfying

$$b(i, S) \leqslant x(i, S) \leqslant a(i, S) \qquad \forall i \in N, S \subseteq N - i. \tag{14}$$

By Lemma 1, Lemma 3, and Theorem 1, such a circulation exists if and only if $\Sigma_{i \in M} a(i, N - M) \geqslant \Sigma_{i \in N-M} b(i, M)$, i.e., if and only if no improving set exists, and by Theorem 2 an optimal solution has been reached.

A *simple cycle* in (N, A) is a sequence of $k (k \geqslant 2)$ distinct arcs $\alpha_m \in A, m = 1, \ldots, k$, and k distinct nodes $i_m \in N$, $m = 1, \ldots, k$, such that either $\alpha_m = (i_m, i_{m+1})$ or $\alpha_m = (i_{m+1}, i_m)$ for $m = 1, \ldots, k$ and $i_{k+1} = i_1$. Arc α_m in this cycle is *positively oriented* if $\alpha_m = (i_m, i_{m+1})$ *negatively oriented* if $\alpha_m = (i_{m+1}, i_m)$. If arc (i, j) has cost c_{ij}, then the *cost* of this cycle is the sum of the costs of its positively oriented arcs less the sum of the costs of its negatively oriented arcs. It is easy to see that taking $v_{ij} = c_{ij} - u_i + u_j$ as the cost of arc (i, j) rather than c_{ij}, does not change the cost of any simple cycle in (N, A).

One way to apply the dual algorithm is by perturbing the costs of the arcs so that no simple cycle has zero cost. For example, every c_{ij} can be increased by a small quantity ϵ_{ij} as described in [1], while these quantities are removed from the final solution. Costs need not be perturbed in advance, but only when necessary, as demonstrated below.

When no zero cost cycles exist with respect to the cost v_{ij}, it is possible to search for a circulation satisfying (14) by assigning flow values to the arcs according to

$$x(i, N - i) = x(N - i, i) \qquad \forall i \in N, \tag{15}$$

$$x(i, S) = a(i, S) \qquad \forall S \subseteq N - i \text{ such that } a(i, S) = b(i, S). \tag{16}$$

We show now that the process terminates with either a circulation satisfying (14) or with an improving set. Let $G \subseteq A$ be the set of arcs for which flow has not yet been defined, and let $(i, n) \in G$ be an arc with a nonzero cost; then x_{in} can be determined by (16), or there exists another arc $(i, m) \in G$ such that $v_{im} = v_{in}$. Note that nodes m and n cannot be connected by arcs with zero cost since this means that a simple cycle with zero cost exists. We leave m and advance as much as possible along arcs of G with zero costs. Let (p, q) be the last arc traversed, then every other arc of G incident with node q has nonzero cost. Therefore $x(q, N - q)$ is known from (16) and either x_{pq}

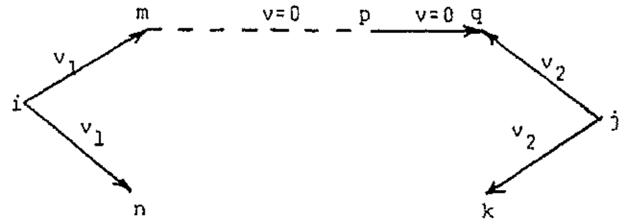


FIG. 4.

can be determined from (15) or there is a (nonzero cost) arc $x_{jq} \in G$ (see Fig. 4). If x_{jq} cannot be determined from (16) then there exists an arc $(j, k) \in G$ with $v_{jk} = v_{jq}$. Nodes k and n cannot be connected by zero cost arcs since this will mean the existence of a zero cost cycle. Thus we continue the search from k. Since $|A|$ is finite, the search terminates with an arc of G to which flow can be assigned. Repeating the search we finally obtain a circulation satisfying (14) if one exists, or we discover a set of nodes $M \subseteq N$ for which (15) and (16) are inconsistent.

The *Primal-Dual Algorithm* (see [12] for the minimum cost flow problem) locates a set $M \subseteq N$ for which

$$\sum_{i \in N-M} a(i, M) - \sum_{i \in N} b(i, N-M) \tag{17}$$

is most negative. We call such a set a *maximum improving set*. If the set with the smallest cardinality among these sets is chosen it can be proved (cf. [7] for the minimum cost flow problem) that if M_i is the maximum improving set found in iteration i, then either (17) is more negative for M_i in iteration i than for M_{i+1} in iteration $i + 1$, or it has the same value and $M_i \subset M_{i+1}$ (we assume that for every $(i, j) \in A$, $r(i, \{j\}) > 0$). Therefore the same value of (17) cannot be repeated more than $|N|$ times and the total number of iterations must be smaller than $|N|$ times the value of (17) in the first iteration. For the polymatroid intersection problem (4) with $r(E) \leqslant r'(E)$ a bound of $n \cdot r(E)$ iterations is obtained.

When no zero cost cycles exist, maximum improving sets (with smallest cardinality) can be found by assigning flows according to (15) and (16) and finally taking the union of the sets thus found, for which the total flow which must leave the set is (strictly) greater than the flow which must enter it. (An exact derivation of this algorithm for the minimum cost problem can be found in [7].)

The algorithm may become clearer in the following example presented in Figures 1 and 3. Figure 5(a) describes the initial solution $u_i = 0$, $\forall i \in N$. Thus $v_{ij} = c_{ij}$ $\forall (i, j) \in A$. The values of v_{ij}, a_{ij}, and b_{ij} are shown. Whenever there exists a set $S \subseteq A - i$ such that v_{ij} has a common value for every $j \in S$, the values of $a(i, S)$ and $b(i, S)$ are also shown.

We now search for the maximum improving set. We start by letting $x_{ij} = a_{ij}$ for arcs with $a_{ij} = b_{ij}$. For example $x_{17} = 3$ and $x_{18} = 3$. This implies $x_{51} = 6$, and x_{25} must be equal to 6. Since $a_{25} = b_{25} = 8$ we found a set of nodes such that the total flow which must enter it is greater than the flow which must leave it. By backtracking we see that this set is (1, 5). We conclude that its complement $\{2, 3, 4, 6, 7, 8, 9, 10\}$ is

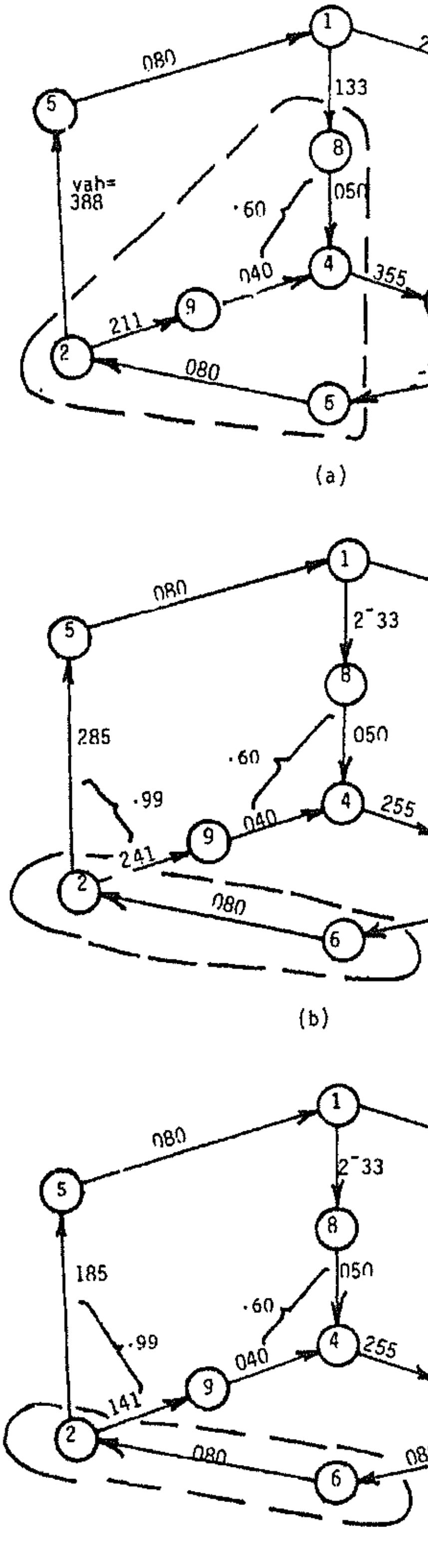
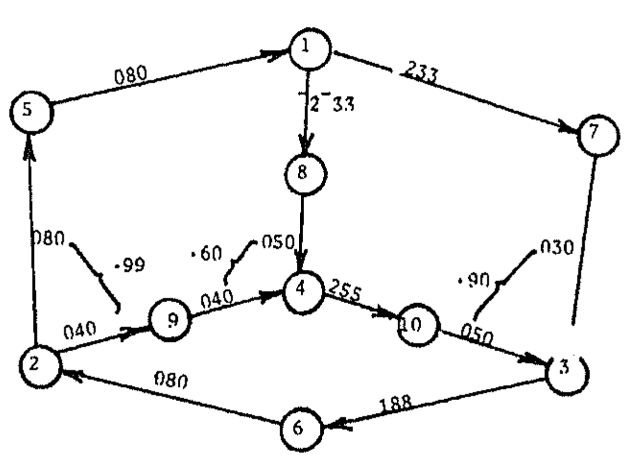
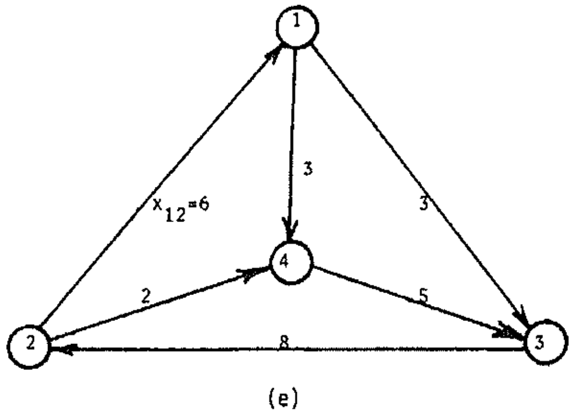

1
5
7
8
4
9
10
2
3
6
080
233
133
vah=
388
.60
050
.90
080
040
355
211
050
080
-200
(a)
080
233
2-33
285
.60
050
030
.90
.99
040
255
241
050
080
-100
(b)
080
233
2-33
185
050
030
.60
.90
.99
040
255
141
050
080
080
(c)

FIG. 5.



(d)



(e)

FIG. 5. (*Continued*)

an improving set, and we can perform Step 2 of the general dual algorithm. However if we want to find the maximum improving set we just ignore the set $\{1, 5\}$ and continue the search. For example we continue by assigning $x_{25} = 8$ and $x_{29} = 1$. This implies $x_{62} = 9$, and x_{36} must be equal to 9. Since $a_{36} = b_{36} = 0$, we found a set, $\{2, 6\}$, which is an improving set. Since $x_{29} = 1$ and $x_{18} = 3$ we assign $x_{94} = 1$ and $x_{84} = 3$ so that $x_{4,10}$ must be equal to 4, but $a_{4,10} = b_{4,10} = 5$ and thus $\{4, 8, 9\}$ is an improving set. We continue with $x_{4,10} = 5, x_{17} = 3, x_{10,3} = 5$, and $x_{73} = 3$, thus x_{36} must be equal to 8. Since $a_{36} = b_{36} = 0$, the set $\{3, 7, 10\}$ is not an improving set.

With all $x_{ij}(i, j) \in A$ determined, the maximum improving set is the union of all the improving sets which were found, i.e., the set $I = \{2, 4, 6, 8, 9\}$. The dual variables u_i, $i \in I$ are increased by the maximum possible increase, which in this case is one unit when v_{25} becomes equal to v_{29} and v_{18} becomes equal to v_{17}. To avoid the possibility of a zero cost cycle we perturb v_{18} so that it is assumed to be smaller than v_{17}. This is shown in Figure 5(b) by marking $v_{17} = 2^-$.

The sequence of networks which results from successive increase of the dual variables of the maximum improving sets is shown in Figure 5(b)-5(d) where the maximum improving sets are marked by broken lines.

VII. POLYMATROID INTERSECTION: GREEDY AND GENEROUS ALGORITHMS

In Theorem 1, the following condition which determines the relation between the upper and lower bounds, and whose meaning has not been explained thus far, is required:

$$k(S) - k(S - T) \geqslant d(T) - d(T - S) \qquad \forall S, T \subseteq E. \tag{18}$$

In this section we show that when (18) holds, instant solutions exist for certain problems of polymatroid intersection. Throughout this section we assume that weights $c_1 \geqslant c_2 \geqslant \cdots \geqslant c_n$ are given to the elements of E. We consider the following linear program:

$$\text{Maximize} \sum_{i \in E} c_i x_i \tag{19}$$

subject to

$$d(S) \leqslant x(S) \leqslant k(S) \qquad \forall S \subseteq E,$$

where $x(S) = \Sigma_{i \in S}\, x_i$, k is a β_0 function and d is a γ_0 function.

Let

$$x_t^{GR} = \begin{cases} k(F_1) & t = 1, \\ k(F_t) - k(F_{t-1}) & t = 2, \ldots, n, \end{cases}$$

where $F_t = \{1, \ldots, t\}$. This is the *greedy solution* which, as we have already mentioned, solves (19) if $c_n \geqslant 0$ and $d \equiv 0$.

Let

$$x_t^{GE} = \begin{cases} d(G_t) - d(G_{t+1}) & t = 1, \ldots, n-1 \\ d(G_n) & t = n. \end{cases}$$

where $G_t = \{t, \ldots, n\}$. This is the *generous solution* which as we prove in the next theorem and corollary, solves (19) if $c_1 \leqslant 0$ and $k \equiv \infty$.

Theorem 3. If d is nondecreasing and supermodular and if $c_n \geqslant 0$, x^{GE} solves the following linear program:

$$\text{Minimize} \sum_{i \in E} c_i x_i$$

subject to

$$d(S) \leqslant x(S) \qquad \forall S \subseteq E.$$

Proof: (i) We first show that the generous solution is feasible. Let $S = \{s(1), s(2), \ldots, s(m)\} \subseteq E$, then

$$x^{GE}(S) - d(S) = [d(G_{s(1)}) + \cdots + d(G_{s(m)})] - [d(G_{s(1)+1}) + \cdots + d(G_{s(m)+1}) + d(S)]$$

(where if $s(j) = n$ then $d(G_{s(j)+1})$ is omitted). From part (b) of Lemma 2 we conclude that the expression is non-negative and thus x^{GE} is feasible.

(ii) Suppose x^{GE} is not optimal. Let y denote the optimal solution, and let t be the smallest $i \in E$ such that $y_i > d(G_i) - d(G_{i+1})$, i.e., the smallest $i \in E$ such that $y(G_i) > d(G_i)$. Since x^{GE} is feasible, $y(S) > d(S)$ must be satisfied for all $t \in S \subseteq G_t$.

(iii) We show now that a better solution can be obtained by decreasing y_t and equally increasing some y_i $i < t$. Suppose this is impossible. Then there exist sets H_p $p = 1, \ldots, \bar{p}$ satisfying $y(H_p) = d(H_p)$ and $t \in \cap_{k=1}^{\bar{p}} H_p \subseteq G_t$. (i.e., if y_t is decreased, some y_i $i \geq t$ must be increased.) From Lemma 2 with $S_q = \{e \in E\colon e$ is included in at least q of the sets $H_p\}$ we obtain

$$\sum_q y(S_q) = \sum_p y(H_p) = \sum_p d(H_p) \leq \sum_q d(S_q) < \sum_q y(S_q),$$

where the last inequality follows from $t \in S_{\bar{p}} = \cap_{k=1}^{\bar{p}} H_p \subseteq G_t$ and part (ii) of this proof. Thus the assumption that x^{GE} is not optimal has led into a contradiction.

Corollary. If d is nonincreasing and supermodular and if $c_1 \leq 0$ then x^{GE} solves the following problem:

$$\text{Maximize} \sum_{i \in E} c_i x_i$$

subject to $d(S) \leq x(S)$.

Returning to problem (17), with both lower and upper bounds on $x(S)$, we observe that if $c \geq 0$ and $x^{GR}(S) \geq d(S)$ for all $S \subseteq E$, then x^{GR} is an optimal solution. If $c \geq 0$ and $x^{GE}(S) \leq k(S)$, then x^{GE} is optimal.

Theorem 4. A ~~necessary and~~ sufficient condition for x^{GE} and x^{GR} to be feasible for any cost vector c is that $k(S) - k(S - T) \geq d(T) - d(T - S)$ for all T, $S \subseteq E$ ~~and for all $S \subseteq T \subseteq E$.~~

Proof: ~~(a)~~ Suppose $S \subseteq T \subseteq E$ and $k(S) < d(T) - d(T - S)$. Then there exists a cost vector such that $x^{GE}(S) = d(T) - d(T - S) > k(S)$. Similarly, if $T \subseteq S \subseteq E$ and $k(S) - k(S - T) < d(T)$, then there exists a cost vector such that $x^{GR}(S) = k(S) - k(S - T) < d(T)$. Therefore the condition is necessary.

(b) Suppose the condition holds. Then

$$d(S) \leq x^{GE}(S) = \sum_{t \in S} [d(\{t, \ldots, n\}) - d(\{t + 1, \ldots, n\})]$$

$$\leq \sum_{t \in S} [k(\{1, \ldots, t\}) - k(\{1, \ldots, t - 1\})] = x^{GR}(S) \leq k(S),$$

where the first and last inequalities hold by the definitions of x^{GE} and x^{GR} and the second holds by the condition of the lemma with $S = \{1, \ldots, t\}$ and $T = \{t, \ldots, n\}$. Therefore, both solutions are feasible.

Theorem 5. Assume $\{i : c_i \geqslant 0\} = F_q$ and the condition of Theorem 4 holds. Then

$$\bar{x}_t = \begin{cases} k(F_t) - k(F_{t-1}) & t \leqslant q, \\ d(G_t) - d(G_{t+1}) & t > q, \end{cases}$$

is an optimal solution to (1¶).

Proof: To see that $\bar{x}$ is feasible note that

$$\begin{aligned} d(S) \leqslant x^{GE}(S) &= \sum_{t \in S} [d(G_t) - d(G_{t+1})] \\ &\leqslant \sum_{\substack{t \in S \\ t \leqslant q}} [k(F_t) - k(F_{t-1}) + \sum_{\substack{t \in S \\ t > q}} [d(G_t) - d(G_{t+1})] \\ &\leqslant \sum_{t \in S} [k(F_t) - k(F_{t-1})] = x^{GR}(S) \leqslant k(S), \end{aligned}$$

where the expression in the middle equals $\bar{x}(S)$.

To see that no better feasible solution exists, let z, z_1, z_2 be defined by the following three problems:

$$z = \max \sum_{i=1}^{n} c_i x_i \text{ s.t. } d(S) \leqslant x(S) \leqslant k(S) \qquad S \subseteq \{1, \ldots, n\},$$

$$z_1 = \max \sum_{i=1}^{q} c_i x_i \text{ s.t. } 0 \leqslant x(S) \leqslant k(S) \qquad S \subseteq \{1, \ldots, q\},$$

$$z_2 = \max \sum_{i=q+1}^{n} c_i x_i \text{ s.t. } d(S) \leqslant x(S) \qquad S \subset \{q+1, \ldots, n\}.$$

The second and third problems are defined on disjoint sets of variables and solved by $(\bar{x}_i, i \leqslant q)$ and $(\bar{x}_i, i > q)$, respectively. The first problem consists of their sum together with additional constraints. Therefore $z \leqslant z_1 + z_2$. However, equality is obtained for $\bar{x}$ and since it is feasible it is also optimal.

Corollary. If the condition of Theorem 3 holds, it can be used to apply the algorithm of Sec. V to the more general problem in which both upper and lower bounds on $x(S)$ exist.

My thanks to Professor A. J. Hoffman for his support and guidance and for suggesting the subject of this paper.

References

[1] G. B. Dantzig, *Linear Programming and Extensions*. Princeton U. P., Princeton, NJ (1963).

[2] J. Edmonds, Submodular functions, matroids, and certain polyhedra. *Combinatorial Structures and Their Applications*. Proceedings of the Calgary International Conference. Gordon and Breach, New York (1970) 67–87.

[3] J. Edmonds, Matroids and the greedy algorithm. *Math. Program.* 1 (1971) 127–136.

[4] S. Fujishige, A primal approach to the independent assignment problem. *J. Oper. Res. Soc. Jpn.* 20 (1977) 1.

[5] S. Fujishige, An algorithm for finding an optimal independent linkage. *J. Oper. Res. Soc. Jpn.* 20 (1977) 1.

[6] S. Fujishige, Polymatroidal dependence structure of a set of random variables. *Inform. Control* 39 (1978) 55–72.

[7] R. Hassin, The minimum cost flow problem: A unifying approach to existing algorithms and a new tree search algorithm. Submitted for publication; part of "On Network Flows," Ph.D. Dissertation, Yale University, 1978.

[8] A. J. Hoffman, Some recent applications of the theory of linear inequalities to extremal combinatorial analysis. *Proc. Symp. Appl. Math.* 10 (1960) 113–127.

[9] M. Iri and N. Tomizawa, An algorithm for finding an optimal independent assignment. *J. Oper. Res. Soc. Jpn.* 19 (1976) 32–57.

[10] E. L. Lawler, Matroid intersection algorithms. *Math. Program.* 9 (1975) 31–56.

[11] E. L. Lawler, *Combinatorial Optimization; Networks and Matroids*. Holt, Rinehart and Winston, New York (1976).

[12] J. F. Shapiro, A note on the primal-dual and out-of-kilter algorithms for network optimization problems. *Networks* 7 (1977) 81–88.

Received July 1978
Accepted February 1981