stichting
mathematisch
centrum

AFDELING MATHEMATISCHE BESLISKUNDE BW 129/80 AUGUSTUS
(DEPARTMENT OF OPERATIONS RESEARCH)

E.L. LAWLER

AN INTRODUCTION TO POLYMATROIDAL NETWORK FLOWS

Preprint

kruislaan 413 1098 SJ amsterdam

Printed at the Mathematical Centre, 413 Kruislaan, Amsterdam.

The Mathematical Centre, founded the 11-th of February 1946, is a non-profit institution aiming at the promotion of pure mathematics and its applications. It is sponsored by the Netherlands Government through the Netherlands Organization for the Advancement of Pure Research (Z.W.O.).

1980 Mathematics Subject Classification: 90B10, 05B35

AN INTRODUCTION TO POLYMATROIDAL NETWORK FLOWS

J

E.L. LAWLER

Computer Science Division, University of California, Berkeley, CA 94720, USA

ABSTRACT

In the "classical" network flow model, flows are constrained by the capacities of individual arcs. In the "polymatroidal" network flow model, flows are constrained by the capacities of *sets* of arcs. Yet the essential features of the classical model are retained: the augmenting path theorem, the integral flow theorem, and the max-flow min-cut theorem all yield to straightforward generalization. In this paper we provide an introduction to the theory of polymatroidal network flows, with the objective of showing that this theory provides a satisfying generalization and unification of both classical network flow theory and much of the theory of matroid optimization.

KEY WORDS & PHRASES: *matroid, polymatroid, network flow, algorithm, optimization, max-flow min-cut theorem.*

NOTE: This research was supported in part by NSF grant MCS78-20054, and in part by the Mathematisch Centrum, Amsterdam. This paper will be published in the Proceedings of the School on Analysis and Design of Algorithms in Combinatorial Optimization, Udine, Italy, September 10-21, 1979.

1. INTRODUCTION

In the "classical" network flow model, flows are constrained by the capacities of individual arcs. In the "polymatroidal" network flow model, flows are constrained by the capacities of *sets* of arcs. Yet the essential features of the classical model are retained: the augmenting path theorem, the integral flow theorem and the max-flow min-cut theorem all yield to straightforward generalization.

In this paper we provide an introduction to the theory of polymatroidal network flows. Our principal objective is to show that this theory provides a satisfying generalization and unification of both classical network flow theory and much of the theory of matroid optimization, including (poly)matroid intersection and matroid partitioning. We shall also indicate how the polymatroidal network flow model can be used to formulate and solve problems with no readily apparent polymatroidal structure.

The results presented here were obtained jointly with C.U. Martel [9], whose solution to a problem in multiprocessor scheduling suggested the formulation of the polymatroidal network flow model. It has come to our attention that the same model was formulated independently by Hassin [4]. A related model has also been investigated by Edmonds and Giles [3].

2. SOME POLYMATROIDAL PRELIMINARIES

We assume that the reader is familiar with the basic concepts of network flow theory and with at least some of the ideas of matroid optimization, as presented in [7]. In this section we present a few results concerning polymatroids which are needed in the remainder of the paper.

A *polymatroid* (E,ρ) is defined by a finite set of *elements* E and a *rank function* $\rho\colon 2^E \to \mathbb{R}^+$ satisfying the properties

$$\rho(\emptyset) = 0, \tag{2.1}$$

$$\rho(X) \leq \rho(Y) \quad (X \subseteq Y \subseteq E), \tag{2.2}$$

$$\rho(X\cup Y)+\rho(X\cap Y) \le \rho(X)+\rho(Y) \quad (X \subseteq E,\ Y \subseteq E). \tag{2.3}$$

Inequalities (2.2) state that the rank function is monotone and inequalities (2.3) assert that it is submodular. If also ρ is integer-valued and $\rho(\{e\}) = 0$ or 1 for all $e \in E$, then the polymatroid is a *matroid*.

We shall be dealing with polymatroids whose elements are arcs of a network. We shall assign values of "flow" to these arcs, which is equivalent to specifying a function $f\colon E \to \mathbb{R}$. This function can be extended to subsets of E in a natural way, i.e.

$$f(\emptyset) = 0,$$
$$f(X) = \textstyle\sum_{x\in X} f(x) \quad (\emptyset \ne X \subseteq E). \tag{2.4}$$

Such an extended flow function f will be said to be *feasible* with respect to the rank function ρ if for all $X \subseteq E$,

$$f(X) \le \rho(X). \tag{2.5}$$

A feasible function f *saturates* X if (2.5) holds with equality. An individual element e will be said to be *saturated* if there is some saturated set in which it is contained.

The following two lemmas apply with respect to any polymatroid (E,ρ) and any feasible function f.

LEMMA 2.1. *If* X *and* Y *are saturated sets, then so are* $X\cap Y$ *and* $X\cup Y$.

Proof. We have

$$\begin{aligned} f(X\cap Y) &\le \rho(X\cap Y), && \text{by feasibility}\\ &\le \rho(X)+\rho(Y)-\rho(X\cup Y), && \text{by submodularity}\\ &\le f(X)+f(Y)-f(X\cup Y), && \text{by } f(X\cup Y) \le \rho(X\cup Y) \text{ and saturation of } X,Y\\ &= f(X\cap Y), && \text{by (2.4).}\end{aligned}$$

Hence $f(X\cap Y) = \rho(X\cap Y)$ and $X\cap Y$ is saturated. The proof for $X\cup Y$ is similar. □

LEMMA 2.2. *If* $e \in E$ *is saturated, then there is a unique minimal saturated set* $S(e)$ *containing* e. *Moreover, for each* $e' \in S(e)$, $e' \neq e$, *it is the case that* $f(e') > 0$.

Proof. Suppose $S(e)$ and $S'(e)$ are distinct minimal saturated sets containing e. By Lemma 2.1, $S(e) \cap S'(e)$ is also a saturated set containing e, and neither $S(e)$ nor $S'(e)$ can be minimal.

Now suppose $S(e)$ is the unique minimal saturated set containing e and there is an element $e' \neq e$ in $S(e)$ such that $f(e') = 0$.

$$\begin{aligned} f(S(e)-\{e'\}) &\leq \rho(S(e)-\{e'\}), && \text{by feasibility} \\ &\leq \rho(S(e)), && \text{by monotonicity} \\ &= f(S(e)), && \text{by assumption} \\ &= f(S(e)-\{e'\}), && \text{since } f(e') = 0. \end{aligned}$$

It follows that $S(e)-\{e'\}$ is also saturated and $S(e)$ cannot be the minimal saturated set containing e. □

3. POLYMATROIDAL FLOW NETWORKS

We shall consider only the simplest type of flow network, namely one in which there is a single *source* s and a single *sink* t. Our objective will be to find a maximum-value flow from s to t.

For each node j of the network there are specified two *capacity functions* α_j and β_j. The function α_j (β_j) satisfies properties (2.1)-(2.3) with respect to the set of arcs A_j (B_j) directed out from (into) node j. Thus (A_j, α_j) and (B_j, β_j) are polymatroids. (Comment: We permit there to be multiple arcs from one node to another. Hence A_j and B_j may be arbitrarily large finite sets.)

A *flow* in the network is an assignment of real numbers to the arcs of the network. We let a flow be represented by a function $f: 2^E \to \mathbb{R}$, obtained as in (2.4). A flow f is *feasible* if

$$f(A_j) = f(B_j), \qquad j \neq s,t, \tag{3.1}$$

$$f \text{ is feasible for } \alpha_j,\beta_j, \qquad \text{for all nodes } j, \tag{3.2}$$

$$f(e) \geq 0, \qquad \text{for all arcs } e. \tag{3.3}$$

Equations (3.1) impose the customary flow conservation law at each node other than the source and sink. Property (3.2) indicates that capacity constraints are satisfied on sets of arcs, and (3.3) simply demands that the flow through each arc be nonnegative. Our objective is to find a feasible flow of maximum value, i.e. one which maximizes

$$v = f(A_s)-f(B_s) = f(B_t)-f(A_t). \tag{3.4}$$

If, for a given feasible flow f, the arc e = (i,j) is saturated with respect to α_i, we shall say that the *tail* of e is saturated and denote the minimal saturated set containing e by T(e), where $T(e) \subseteq A_i$. Similarly, if e is saturated with respect to β_j, we shall say that the *head* of e is saturated and denote the minimal saturated set containing e by H(e), where $H(e) \subseteq B_j$.

In the case of an ordinary flow network in which there is a specified capacity c_{ij} for each arc e = (i,j), we can define $\alpha_i(e) = \beta_j(e) = c_{ij}$, and then extend the functions α_i,β_i to sets as in (2.4). The resulting capacity functions are modular, i.e. satisfy (2.3) with equality. Note that in this special case the head of an arc e is saturated if and only if its tail is saturated, and H(e) = T(e) = {e}.

4. AUGMENTING PATHS

With respect to a given feasible flow f, an *augmenting path* is an undirected path of distinct arcs (but not necessarily distinct nodes) from s to t such that

(4.1) each backward arc e in the path is nonvoid, i.e. f(e) > 0, and

(4.2) if the head (tail) of a forward arc e in the path is saturated, then the following (preceding) arc in the path is a backward arc contained in H(e) (T(e)).

In an ordinary flow network the minimal saturated set containing a saturated arc e is simply {e}, and since repetitions of arcs are not allowed, (4.2) does not permit any forward arc to be saturated. Thus, in this specialization our definition almost exactly coincides with the accepted notion of an augmenting path, the only (inconsequential) difference being that we permit repetitions of nodes.

We shall want to use augmenting paths in the customary way. That is, for some strictly positive δ, we want to increase the flow through each forward arc by δ and decrease the flow through each backward arc by δ, and thereby obtain an augmented flow which is feasible. It is not readily apparent that this can be done in our generalization.

LEMMA 4.1. *For any augmenting path there exists a strictly positive value of* δ *by which the flow can be augmented.*

Proof. There are two types of constraints on δ. First, the flow through each backward arc must remain nonnegative, and (4.2) assures us that there is a strictly positive value of δ for which this is possible. Second, for each node j and each $X \subseteq A_j$ (and similarly for each $X \subseteq B_j$) the resulting flow f' must be such that

$$f'(X) \leq \alpha_j(X).$$

Let m(X) denote the number of forward arcs in X minus the number of backward arcs. Then we must have

$$f'(X) = f(X)+\delta m(X) \leq \alpha_j(X). \qquad (4.3)$$

The only way in which (4.3) could fail to permit δ to be strictly positive would be for X to be saturated by f and for m(X) to be strictly positive. But if X is saturated and contains forward arcs $e_1,e_2,\ldots,e_\ell$, then the tails of these forward arcs are saturated and $T(e_i) \subseteq X$, $i = 1,2,\ldots,\ell$. By (4.2), each e_i must be paired with a distinct backward arc $e_i' \in T(e_i)$. It follows that $m(X) \leq 0$, and the constraints (4.3) permit δ to be strictly positive. □

For many applications we need to be assured that there exists a maximal flow that is integer-valued. Hence we wish to obtain an integer version of Lemma 4.1.

An augmenting path can be *shortcut* if some portion of it can be removed to obtain a shorter augmenting path. For example, suppose an augmenting path contains two forward arcs e and e', both directed into the same node j, and that the heads of both of these arcs are unsaturated. If e occurs before e', then all the arcs following e up to and including e' can be removed from the augmenting path. A similar condition holds for two unsaturated forward arcs directed out from the same node.

The reader is invited to establish that an augmenting path which does not admit a shortcut has the following property: If the path passes through a given node j, the occurrences of j in the path are ordered as follows. First, there may be pairs of consecutive arcs of the form (e_h,e_h'), $h = 1,2,\ldots,k$, where each e_h is a forward arc directed into j whose head is saturated and $e_h' \in H(e_h)$. Second, there may be no more than one arc pair of the form (e,e'), where e is either a backward arc directed out from j or a forward arc directed into j whose head is unsaturated and e' is either a backward arc directed into j or a forward arc directed out from j whose tail is unsaturated. (If j = s(t), then there is only a single arc e' (e).) Third, there may be arc pairs of the form (e_i',e_i), $i = 1,2,\ldots,\ell$, where each e_i is a forward arc directed out from j whose tail is saturated and $e_i' \in T(e_i)$. Moreover, the sets $H(e_h)$, {e,e'} and $T(e_i)$ are disjoint.

From these observations we can conclude that an augmenting path which does not admit a shortcut contains at most one forward arc in A_j which is unsaturated with respect to α_j and at most one forward arc in B_j which is unsaturated with respect to β_j. And, moreover, each of the sets $H(e_h)$ and $T(e_i)$ remains saturated after augmentation.

LEMMA 4.2. *Suppose all capacity functions and the existing feasible flow are integer-valued. Then for any augmenting path which admits no shortcut there exists a strictly positive integer value of δ by which the flow can be augmented.*

Proof. Let the maximum permissible value of δ be determined as in the proof of the previous lemma. If δ is determined by the amount of flow in a backward arc, then δ is an integer. So suppose a constraint of the form (4.3) is binding. If $m(X) = 1$, then δ is an integer. So suppose $m(X) > 1$. After augmentation of the existing flow f by δ, the resulting flow f' saturates X. As before, let $e_1, e_2, \ldots, e_\ell$ denote the forward arcs in X whose tails are saturated by f. Then the sets $T(e_i)$, $i = 1,2,\ldots,\ell$ remain saturated after augmentation, and the set

$$X' = X \cup T(e_1) \cup \ldots \cup T(e_\ell)$$

is also saturated by f'. Hence

$$f'(X') = f(X') + \delta m(X') = \alpha_j(X'). \tag{4.4}$$

But there is at most one forward arc in X' whose tail is unsaturated by f. Hence $m(X') \leq 1$ and (4.4) indicates that δ is integer. □

5. A LABELING PROCEDURE

Augmenting paths can be found by means of a labeling procedure which is much like that employed for ordinary flow networks. The principal difference is that labels are applied to arcs rather than to nodes. A labeling procedure which constructs augmenting paths without shortcuts is as follows:

Step 0. Initially all arcs are unlabeled and unscanned.

Step 1. To each nonvoid arc directed into s apply the label $(-,*)$ and to each arc directed out from s whose tail is unsaturated apply the label $(+,*)$.

Step 2. If there is an arc labeled "-" which is directed out from t or an arc labeled "+" which is directed into t whose head is unsaturated, stop. (An augmenting path has been found. The arcs in this path can be determined by backtracing, using the second component of each arc label.)

Step 3. If there are no arcs which are labeled and unscanned, stop. (There is no augmenting path.) Otherwise, find such an arc e and scan it as follows:
Suppose either e has a "+" label and is directed into node j or e has a "-" label and is directed out from node j. If e has a "+" label and its head is saturated, then apply the label (-,e) to all unlabeled arcs in H(e). If e has a "+" label and its head is unsaturated or e has a "-" label then apply the label (-,e) to all nonvoid arcs directed into j and apply the label (+,e) to all unlabeled arcs directed out from j whose tails are unsaturated. In addition, if e has a "-" label and its tail is saturated, apply the label (+,e) to all arcs e' such that $e \in T(e')$. Return to Step 2.

We have asserted in Step 3 that if the labeling procedure fails to find an augmenting path, then no augmenting path exists. This fact is by no means evident. The alert reader may even suspect that the labeling procedure may be defective, in that it permits a given arc to be given only one type of label ("+" or "-"), whereas both types might be applicable. We shall now prove that if the procedure fails to find an augmenting path then not only is there no augmenting path, but the flow is in fact maximal.

THEOREM 5.1 (Augmenting Path Theorem). *A flow is maximal if and only if it admits no augmenting path.*

Proof. If there is an augmenting path then Lemma 4.1 shows that the flow cannot be maximal. So suppose that the labeling procedure fails to find an augmenting path and let us show that this implies that the flow is maximal. The discussion which follows is with reference to the labels existing at the termination of the procedure.

Let us partition the nodes of the network into two sets, S and T. S is to contain node s, together with all nodes j such that either there is an arc directed from j with a "-" label or there is an arc directed into j with a "+" label whose head is unsaturated. All other nodes (including necessarily t) are in the set T.

We have thus defined a cut (S,T). Each "backward" arc (i,j), where $i \in T$, $j \in S$, must be void, else it would have received a "-" label and i would be in S. Let us partition the forward arcs (i,j), where $i \in S$, $j \in T$ into two sets U and L. Set U is to contain all unlabeled forward arcs and L is to contain all forward arcs which are labeled (either "+" or "-"). We thus have the situation indicated in Figure 1.

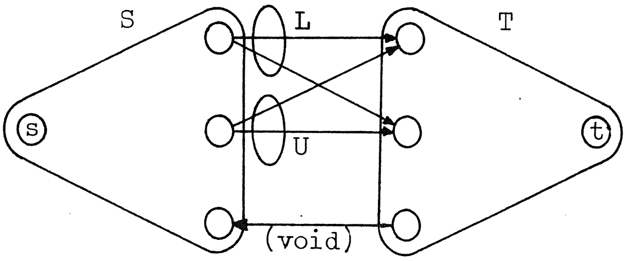

Figure 1. Cut (S,T,L,U).

Consider any node $i \in S$ and the set of arcs $U \cap A_i$. The tail of each arc $e \in U \cap A_i$ is saturated, else e could have a "+" label. Moreover, $T(e) \subseteq U \cap A_i$. For suppose there is some $e' \in T(e)$ such that $e' \notin U$. Such an arc e' cannot be unlabeled and directed to a node in S, else it could have received a "-" label. So e' must be labeled and directed to a node in T. If e' has a "-" label, then e could have received a "+" label from the scanning of e'. So e' must have a "+" label and this can be so only because there is some arc $e'' \in T(e')$ which has a "-" label. But if $e'' \in T(e)$ then e could have a "+" label. And if $e'' \notin T(e)$, we would have $e' \in T(e) \cap T(e') \neq T(e')$, a contradiction. Hence $e' \in U$ and $T(e) \subseteq U \cap A_i$. It follows that $U \cap A_i$ is the union of saturated sets and is itself a saturated set.

Now consider any node $j \in T$ and the set of arcs $L \cap B_j$. If an arc $e \in L \cap B_j$ has a "-" label, then there is a "+" labeled arc $e' \in L \cap B_j$ such that $e \in H(e')$. If an arc $e \in L \cap B_j$ has a "+" label, then its head is saturated (else $j \in S$) and $H(e) \subseteq L \cap B_j$, by the following reasoning. An arc $e' \in H(e)$ cannot be unlabeled because it could receive a "-" label from the scanning of e. If e' has a "-" label, then it must be directed from a node in S, by definition of S. If e' has a "+" label this label must have resulted from the scanning of an arc incident to a node in S. Hence $e' \in L$ and $H(e) \subseteq L \cap B_j$. It follows that $L \cap B_j$ is the union of saturated sets and is itself a saturated set.

We have shown that the net flow across the cut (S,T) is

$$\sum_{i \in S} f(U \cap A_i) + \sum_{j \in T} f(L \cap B_j) = \sum_{i \in S} \alpha_i(U \cap A_i) + \sum_{j \in T} \beta_j(L \cap B_j).$$

The flow is therefore maximal and there can be no augmenting path. □

From Theorem 5.1 and Lemma 4.2 we also obtain the following result in the case of integer capacities.

THEOREM 5.2 (Integral Flow Theorem). *If all capacity functions are integer-valued, then there is a maximal flow which is integral.*

6. MAX-FLOW MIN-CUT THEOREM

The proof of Theorem 5.1 clearly indicates the form of a max-flow min-cut theorem for polymatroidal network flows, which we now proceed to state.

An *arc-partitioned cut* (S,T,U,L) is defined by a partition of the nodes into two sets S and T, with $s \in S$, $t \in T$, and by a partition of the forward arcs across the cut into two sets U and L. The *capacity* of such an arc-partitioned cut is defined as

$$c(S,T,U,L) = \sum_{i \in S} \alpha_i(U \cap A_i) + \sum_{j \in T} \beta_j(L \cap B_j).$$

As in the case of ordinary flow networks, the value v of any feasible flow f is equal to the net flow across any cut, i.e.

$$v = f(U)+f(L)-f(B),$$

where B is the set of backward arcs, and clearly

$$v \le c(S,T,U,L). \tag{6.1}$$

THEOREM 6.1 (Max-Flow Min-Cut Theorem). *The maximum value of a flow is equal to the minimum capacity of an arc-partitioned cut.*

Proof. The proofs of Theorems 5.1 and 5.2, together with (6.1), are sufficient to establish the theorem for networks in which all capacity functions are integer (or rational) valued.

To complete the proof of the theorem, we must show that every network actually admits a maximal flow. (There is the possibility that a sequence of flow augmentations might fail to terminate with a well-defined maximal flow.) This question will be resolved in a later paper, in which questions of algorithmic efficiency will also be addressed. □

In the next several sections we shall indicate how the max-flow min-cut theorem specializes in various applications of the polymatroidal network flow model.

7. MATROID INTERSECTION

The (unweighted) matroid intersection problem is as follows. Given two matroids (E,ρ_1) and (E,ρ_2), find the largest possible set I which is independent in each of the matroids, i.e. such that $\rho_1(I) = \rho_2(I) = (I)$. This problem can be formulated and solved as a polymatroidal network flow problem, as shown in Figure 2. There are exactly two nodes, s and t, in the network and each arc from s to t corresponds to an element of E. The

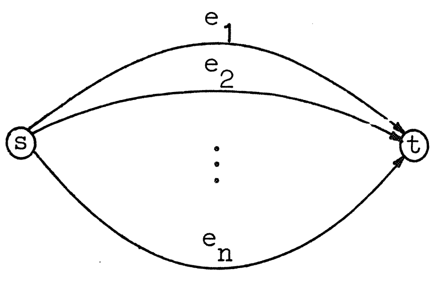

Figure 2. Flow Network for Matroid Intersection.

two capacity functions are determined by the two matroid rank functions: $\alpha_s = \rho_1$, $\beta_t = \rho_2$. Since these capacity functions are integer-valued, there exists a maximal flow which is integer. Any such integral maximal flow corresponds to a solution to the matroid intersection problem.

When the maximal flow algorithm suggested in Section 5 is applied to the network shown in Figure 2, it specializes precisely to the well-known matroid intersection algorithm [6,7]. An augmenting path without shortcuts corresponds to an "augmenting sequence". Minimal saturated sets T(e) and H(e) correspond to circuits $C_e^{(1)}$ and $C_e^{(2)}$ as defined in [6,7], and so forth.

It is also interesting to note that the max-flow min-cut theorem specializes exactly the well-known matroid intersection duality theorem. A partitioned cut (S,T,L,U) must have S = {s}, T = {t} and is obviously determined by a partition of the set E into subsets L and U. Thus we have the following

THEOREM 7.1 (Matroid Intersection Duality Theorem).

$$\max|I| = \min_{L \cup U = E} \{\rho_1(U)+\rho_2(L)\}.$$

8. MATROID PARTITIONING

Suppose we are given k matroids (E,ρ_i), i = 1,2,...,k, and we wish to determine whether or not there exists a partition of E into k sets I_i, i = 1,2,...,k, such that I_i is independent in (E,ρ_i). We can construct a flow network as shown in Figure 3. In this network each arc (s,e) has unit capacity, the flow into each node (E,ρ_i) is constrained by a capacity function $\beta_i = \rho_i$, and there are no other capacity constraints. If there exists an integral maximal flow of value |E| (which necessarily saturates each arc (s,e)), then there exists a partition of the desired type, otherwise not.

As it might be expected, when the maximal flow algorithm is applied to the network shown in Figure 3, it specializes to an algorithm very similar to that which has been proposed for solving the matroid partition problem [1,7].

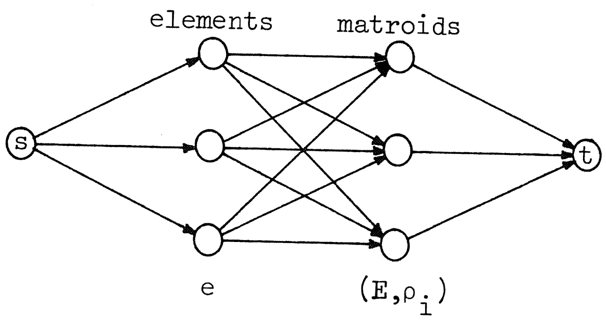

Figure 3. Flow Network for Matroid Partitioning.

A well-known set of necessary and sufficient conditions for the existence of a solution to the partitioning problem can be obtained quite easily from the max-flow min-cut theorem for polymatroidal network flows. There exists a solution to the matroid partitioning problem if and only if for the network of Figure 3 there does not exist an arc-partitioned cut with capacity strictly less than $|E|$. Any cut of finite capacity must be of of the form shown in Figure 4 where $S = A\cup\{s\}$, for some $A \subseteq E$. (A node (E,ρ_i) cannot be in S, else the cut would have unbounded capacity.) The capacity of such a cut is $\sum\rho_i(A) + |E-A|$, and if this is strictly less than $|E|$ we have $\sum\rho_i(A) < |A|$.

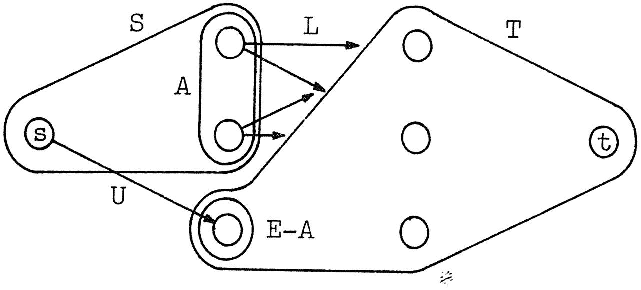

Figure 4. Cut in Proof of Theorem 8.1.

THEOREM 8.1 (Edmonds and Fulkerson [2]). *There exists a solution to the matroid partition problem if and only if for all* $A \subseteq E$,

$$|A| \leq \sum_i \rho_i(A).$$

9. A SCHEDULING PROBLEM

The polymatroidal network flow model can be applied to formulate and solve problems which have no readily apparent polymatroidal structure. The following problem in scheduling is such an example.

Suppose there are n *jobs*, $j = 1,2,\ldots,n$, each with a *release time* r_j, a *deadline* d_j, and a *processing requirement* p_j. It is desired to obtain a feasible preemptive schedule for these n jobs on m machines, where machine i has *speed* s_i, with $s_1 \geq s_2 \geq \ldots \geq s_m$. The usual conventions apply, i.e. a machine can work on only one job at a time and no job can be worked on by more than one machine at a time.

The set of 2n numbers $\{r_j\} \cup \{d_j\}$ defines at most 2n-1 distinct time intervals. We construct a flow network as shown in Figure 5, with a node for each job j and a node k for each of the time intervals. There is an arc (j,k) if and only if it is feasible to process job j in interval k. The arc flow f((j,k)) indicates the number of units of processing of job j to be done in interval k.

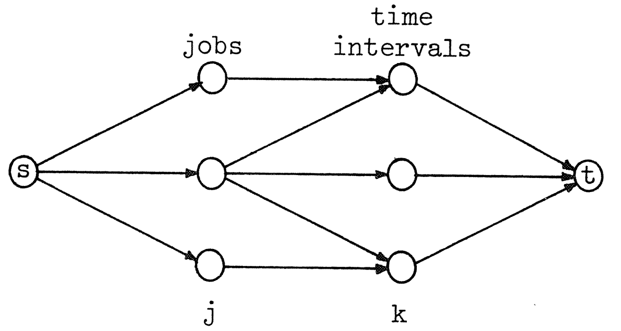

Figure 5. Flow Network for Scheduling Problem.

Suppose interval k has length t_k. It follows from a well-known result of scheduling theory that the arc flows into node k should be constrained by a capacity function of the form

$$\beta_k(X) = \begin{cases} (s_1+s_2+\ldots+s_\ell)t_k, & \text{if } |X| = \ell \le m-1, \\ (s_1+s_2+\ldots+s_m)t_k, & \text{if } |X| \ge m. \end{cases}$$

The function β_k is easily shown to satisfy properties (2.1)-(2.3).

Finally, a capacity of p_j is specified for each arc (s,j). There are no other capacity constraints in the network. We assert that there exists a feasible schedule if and only if there exists a flow whose value is equal to $\sum p_j$, i.e. a flow which saturates each arc (s,j).

If there does not exist a feasible schedule, the maximal flow algorithm identifies a subset of jobs for which it is easy to show that there is not enough available machine capacity. The details of the application of the max-flow min-cut theorem can be found in [9].

We comment that if the machines are *identical*, i.e. all have the same speed, the polymatroidal flow network for this problem specializes to an ordinary flow network [5]. If the machines are *unrelated*, i.e. have different speeds for different jobs, there appears to be no better alternative than solution by linear programming [8].

10. FURTHER EXTENSIONS

The polymatroidal network flow model can be extended and elaborated in many of the same ways as the classical model. Lower bounds on arc flow, in the form of supermodular set functions, can be applied. Convex cost functions can be specified for the flow through individual arcs, and a minimum-cost feasible flow can be computed. It appears that the polymatroidal model retains the desirable features of ordinary network flows under these extensions, and this will be one direction for future investigation.

REFERENCES

1. Edmonds, J., Minimum partition of a matroid into independent subsets, *J. Res. NBS*, 69B, 67-72, 1965.
2. Edmonds, J., Fulkerson, D.R., Transversals and matroid partition, *J. Res. NBS*, 69B, 147-153, 1965.
3. Edmonds, J., Giles, R., A min-max relation for submodular functions on graphs, *Annals Discrete Math.*, 1, 185-204,1977.
4. Hassin, R., On network flows, Ph.D. thesis, Yale University, May, 1978.
5. Horn, W., Some simple scheduling algorithms, *Naval Res. Logist. Quart.*, 21, 177-185, 1974.
6. Lawler, E.L., Matroid intersection algorithms, *Math. Programming*, 9, 31-56, 1975.
7. Lawler, E.L., *Combinatorial optimization: networks and matroids,* Holt, Rinehart and Winston, 1976.
8. Lawler, E.L., Labetoulle, J., On preemptive scheduling of unrelated parallel processors by linear programming, *J. ACM*, 25, 612-619, 1978.
9. Martel, C.U., Generalized network flows with an application to multiprocessor scheduling, Ph.D. thesis, University of California at Berkeley, May, 1980.