COMPUTING MAXIMAL "POLYMATROIDAL" NETWORK FLOWS

by

E. L. Lawler and C. U. Martel

Memorandum No. UCB/ERL M80/52

22 December 1980

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

COMPUTING MAXIMAL "POLYMATROIDAL" NETWORK FLOWS

E. L. Lawler
Computer Science Division
University of California, Berkeley

C. U. Martel
Department of Electrical and Computer Engineering
University of California, Davis

Abstract

In the "classical" network flow model, flows are constrained by the capacities of individual arcs. In the "polymatroidal" network flow model introduced in this paper, 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 are all shown to yield to straightforward generalization. We describe a maximal flow algorithm which finds augmenting paths by labeling arcs instead of nodes, as in the case of the classical model. As a counterpart of a known result for the classical model, we prove that the number of augmentations required to achieve a maximal value flow is bounded by the cube of the number of arcs in the network, provided each successive augmentation is made along a shortest augmenting path, with ties between shortest paths broken by lexicography.

Keywords: matroid, polymatroid, flow network, algorithm, optimization, max-flow min-cut theorem.

This research was supported by the National Science Foundation Grant MCS 78-20054.

1. Introduction

In the "classical" network flow model, flows are constrained by the capacities of individual arcs. In the "polymatroidal" network flow model introduced in this paper, flows are constrained by the capacities of sets of arcs. Yet the essential features of the classical model are related: the augmenting path theorem, the integral flow theorem and the max-flow min-cut theorem are all shown to yield to straightforward generalization. We describe a maximal flow algorithm which finds augmenting paths by labeling arcs instead of nodes, as in the classical model. As a counterpart to a result of Edmonds and Karp [3] for the classical model, we prove that the number of augmentations required to achieve a maximal value flow is bounded by the cube of the number of arcs in the network, provided each successive augmentation is made along a shortest augmenting path, with ties between shortest paths broken by lexicography.

We believe that the theory of polymatroidal network flows provides a satisfying generalization and unification of both classical network flow theory and much of the theory of matroid optimization. In particular, various well known algorithms for (poly)matroid intersection [1,8,9,10,13], matroid partitioning [2,10] and other matroid optimization problems [5,7] can be shown to be equivalent, or very nearly equivalent to specializations of the maximal flow algorithm presented in this paper. These specializations are obtained by applying the maximal flow algorithm to appropriately formulated polymatroidal flow networks. Moreover, the application of the max-flow min-cut theorem to these networks yields simple proofs of known duality theorems for these matroid optimization problems. These observations, as well as generalizations of the polymatroidal flow model to provide for costs and lower bounds on arc flows, will be reported in papers in preparation by the authors.

The plan of this paper is as follows. Section 2 deals with preliminaries, in the form of some simple lemmas concerning polymatroidal rank functions. In Section 3 the polymatroidal network flow model is formulated and in Section 4 the concept of an augmenting path is introduced. Sections 5 through 7 present the maximal flow algorithm and provide proofs of the augmenting path theorem and max-flow min-cut theorem with reference to that algorithm. Sections 8 through 10 concern properties of shortcut-free augmenting paths and provide a proof of the integrality theorem. In Sections 11 and 12 the bound on the number of augmentations is proved, and various issues of computational complexity are discussed.

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[10]. 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* ρ: $2^E \to \mathbb{R}^+$ satisfying the properties

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

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

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

Inequalities (2.2) state that the rank function is monotone and inequalities (2.3) assert that it is submodular. If ρ is also 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.

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

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

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

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

<u>Lemma 2.1</u>. *If* Y *is a saturated set, then*

$$\rho(X) - f(X) \geq \rho(X \cap Y) - f(X \cap Y),$$
$$\rho(X) - f(X) \geq \rho(X \cup Y) - f(X \cup Y),$$

for all $X \subseteq E$.

<u>Proof</u>:

$\rho(X) + \rho(Y) \geq \rho(X \cup Y) + \rho(X \cap Y)$, by submodularity

$f(X) + f(Y) = f(X \cup Y) + f(X \cap Y)$, by (2.4).

Hence

$$\rho(X) - f(X) + \rho(Y) - f(Y) \geq \rho(X \cup Y) - f(X \cup Y) + \rho(X \cap Y) - f(X \cap Y),$$

and the result follows from the saturation of Y and the feasibility of f. □

Lemma 2.2. *If* X *and* Y *are saturated sets, then so are* $X \cap Y$ *and* $X \cup Y$.

Proof: Apply Lemma 2.1 with X a saturated set. □

Lemma 2.3. 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 2.2, $S(e) \cap S'(e)$ 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 $f\colon 2^E \to \mathbb{R}$, obtained as in (2.4), where E is the set of arcs. A flow f is *feasible* if

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

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

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

(We write f(e) in place of the more cumbersome f({e}).)

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, an 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 $H(e) = T(e) = \{e\}$.

Remark: We could choose to formulate the network flow model in terms of capacity functions which are simply nonnegative and submodular. All of the results in this paper would then follow, in virtually the same form. This is due to the fact that for every nonnegative submodular function ρ there is polymatroidal rank function $\bar{\rho}$ in the relation

$$\bar{\rho}(\emptyset) = 0,$$
$$\bar{\rho}(X) = \min\{\rho(Y) \mid X \subseteq Y \subseteq E\}, \ X \neq \emptyset,$$

where $\bar{\rho}$ has the same effect as a capacity function as ρ. We prefer to assume capacity functions are monotone, as well as submodular, because it slightly simplies definitions and seems somewhat more intuitively appealing. Lemma 2.3 indicates the effect of monotonicity.

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 (4.1) 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 $\mu(X)$ denote the number of forward arcs in X minus the number of backward arcs. Then we must have

$$f'(X) = f(X) + \delta\mu(X) \leq \alpha_j(X). \tag{4.3}$$

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 $\mu(X)$ 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 $T(e_i) \subseteq X$, $i = 1, \ldots, \ell$. By (4.2), each e_i must be paired with a distinct backward arc $e_i' \in T(e_i)$. It follows that $\mu(X) \leq 0$ if X is saturated, and the constraints (4.3) permit δ to be strictly positive. □

5. Maximal Flow Algorithm

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 instead of nodes. We present below an algorithm for computing maximum-value flows in which labeling is carried out in a "breadth-first" manner so that when an augmenting path is found it contains as few arcs as possible. We do not need shortest augmenting paths for our present purposes, but such paths will be useful later on.

The label applied to each arc has three components. The first component is either "+" or "-" indicating whether the arc is forward or backward. The second component is an arc name or index, for use in backtracing to find an augmenting path. The third component is a *level number* used to carry out labeling in a breadth-first manner. (The level of an arc is its position in a shortest path from s.) An arc is said to be *at* level ℓ if it is labeled and its level number is ℓ.

Maximal Flow Algorithm

Step 0. (Initialize flow) Let f be any feasible flow, possibly the zero flow.

Step 1. (Label arcs at level 1) To each nonvoid arc directed into s apply the label (-,*,1) and to each arc directed out from s whose tail is unsaturated apply the label (+,*,1). (No other arcs are labeled and no arcs are scanned.)

Set ℓ to zero.

Go to Step 3.

Step 2. (Label arcs at level ℓ+1) As long as there remains a labeled arc at level ℓ which is unscanned, choose such an arc e and scan it as follows:

(2.1) If e has a "+" label and its head is saturated, then apply the label (-,e,ℓ+1) to all unlabeled arcs in H(e).

(2.2) If e has a "-" label and its tail is saturated, then apply the label (+,e,ℓ+1) to all unlabeled arcs e' such that $e \in T(e')$.

(2.3) If e has a "+" label, its head is unsaturated, and e is directed into node j, or if e has a "-" label and e is directed out from node j, then apply the label $(+,e,\ell+1)$ to all unlabeled arcs directed out from $\bar{j}$ whose tails are unsaturated and the label $(-,e,\ell+1)$ to all unlabeled nonvoid arcs directed into j.

Step 3. (Check for augmenting path) If there is an arc e labeled "-" which is directed out from t or there is an arc labeled "+" which is directed into t and whose head is unsaturated, go to Step 5. (An augmenting path has been found.)

Step 4. (Check for maximality of flow) If there is no arc at level $\ell+1$, then stop. (There is no augmenting path and the flow is maximal.) Otherwise set ℓ to $\ell+1$ and return to Step 2.

Step 5. (Augment flow) Starting at arc e, identified in Step 3, find an augmenting path by backtracing. (If e has the label $(+,e',\ell+1)$ then e' is the second-to-last arc in the path, the label on arc e' indicates the third-to-last arc, and so on.) Determine the maximum amount δ by which the flow can be augmented. Augment the flow to obtain a new feasible flow f and return to Step 1. (We defer until later a discussion of how to find δ.)

6. Augmenting Path Theorem

We asserted in Step 4 of the maximal flow algorithm 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 6.1. (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 i such that either there is an arc directed from i with a "-" label or there is an arc directed into i 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.

Consider any node $i \in S$ and any arc $e \in U \cap A_i$. The tail of e is saturated, else a "+" label would have been applied to e, either in Step 1, if $i = s$, or in Step (2.3), if $i \neq s$. Moreover, we assert that $T(e) \subseteq U \cap A_i$. Suppose there were some arc $e' \in T(e)$, with $e' \notin U$. The following cases exhaust all possibilities, if e' is not in U:

(i) e' is unlabeled and directed into a node $j \in S$. This is not possible, because a "-" label would have been applied to e' in Step 1, if $j = s$, or in Step (2.3), if $j \neq s$, when some arc incident to j was scanned. (Note that e' is nonvoid, by Lemma 2.3.)

(ii) e' has a "-" label. This is not possible, because scanning e' would apply a "+" label to e in Step (2.2).

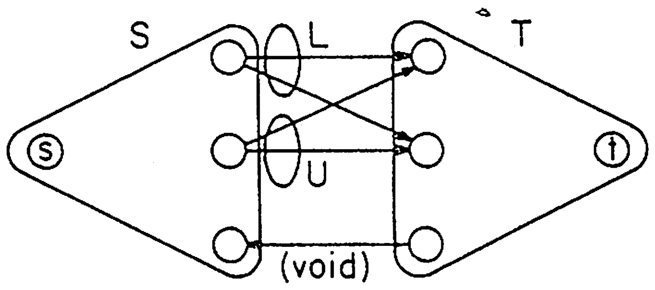

Fig. 1. Cut (S,T) with sets of forward arcs L,U.

(iii) e' has a "+" label. If so, e' could only have received its label in Step (2.2) when an arc $e'' \in T(e')$ was scanned. But $e'' \in T(e)$, else we would have $e' \in T(e) \cap T(e') \subsetneq T(e')$, in contradiction to Lemma 2.2. But if $e'' \in T(e)$, then e would have received a "+" label in Step (2.2) when e" was scanned. Hence this case cannot occur.

Since $T(e) \subseteq U \cap A_i$, for each $e \in U \cap A_i$, $U \cap A_i$ is the union of saturated sets and is itself a saturated set.

Now consider any node $j \in T$ and the set $L \cap B_j$. If any arc $e \in L \cap B_j$ has a "-" label, then this label could only have been applied in Step (2.1) when some arc e' with a "+" label was scanned and $e \in H(e')$. It follows that if $H(e) \subseteq L \cap B_j$, for each arc $e \in B_j$ with a "+" label, then $L \cap B_j$ is the union of saturated sets. So suppose $e \in B_j \cap L$, e has a "+" label, and there is an arc $e' \in H(e)$, with $e' \notin L$. The following cases exhaust all possibilities, if e' is not in L:

(i) e' is unlabeled. This is not possible, because a "-" label would have been applied to e' in Step (2.1) when e was scanned.

(ii) e' has a "-" label and is directed from a node $i \in T$. This is not possible, because $i \in S$ if e' is labeled "-".

(iii) e' has a "+" label and is directed from a node $i \in T$. The tail of e' is saturated, else e' was labeled in Step (2.3) with $i \in S$. Since the tail of e' is saturated, it could only have received its label in Step (2.2) when some arc $e'' \in T(e')$ with a "-" label was scanned. But then $i \in S$. Hence this case cannot occur.

It follows that $L \cap B_j$ is the union of saturated sets and is itself a saturated set.

We have shown that $f(U \cap A_i) = \alpha_i(U \cap A_i)$, for each $i \in S$, and $f(L \cap B_j) = \beta_j(L \cap B_j)$, for each $j \in T$. Since each arc (i,j), with $i \in T$, $j \in S$, is void, it follows that the net flow across the cut is

$$f(U) + f(L) = \sum_{i \in S} f(U \cap A_i) + \sum_{j \in T} f(L \cap A_j)$$

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

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

7. Max-Flow Min-Cut Theorem

The proof of Theorem 6.1 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

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

Theorem 7.1. (Max-Flow Min-Cut Theorem). *The maximum value of a flow is equal to the minimum capacity of an arc-partitioned cut.*

Proof: Let f be a maximal flow. (Such a flow clearly exists, since the flow problem is a linear programming problem with a nonempty bounded set of feasible solutions.) Apply the labeling procedure of the maximal flow algorithm and construct an arc-partitioned cut, as in the proof of Theorem 6.1. The capacity of this cut is equal to the value v of the flow f. The theorem now follows from (6.1). □

8. Shortcut-free Augmenting Paths

An augmenting path can be *shortcut* if some portion of it can be removed to obtain a shorter augmenting path. For example, suppose $P = (e_1, \ldots, e_p)$ contains a forward arc e_i and a backward arc e_k, with $i+1 < k$ and $e_k \in H(e_i)$. Then removing arcs $e_{i+1}, \ldots, e_{k-1}$ from P yields a shorter augmenting path $P' = (e_1, \ldots, e_i, e_k, \ldots, e_p)$. The labeling procedure of the maximal flow algorithm finds shortest augmenting paths and these are certainly shortcut-free.

In the next section we show that that the *capacity* of a shortcut-free augmenting path, i.e. the maximum amount δ by which the flow can be augmented along it, can be determined by computing the "capacities" of successive pairs of arcs in the path. The integrality theorem for polymatroidal flows then follows immediately. In order to obtain these results, we first provide a few more definitions.

An ordered pair of arcs $(e, \bar{e})$ is said to be *admissible* with respect to a given flow f if, when e is scanned in the labeling procedure, $\bar{e}$ is eligible to receive a label from e. More specifically, admissible pairs are of three kinds, corresponding to the ways in which e' might be labeled in Steps (2.1)-(2.3) of the maximal flow algorithm:

(i) A pair $(e, \bar{e})$ is a *saturated head pair* at node j if both e and $\bar{e}$ are directed into j, the heads of e and $\bar{e}$ are saturated and $\bar{e} \in H(e)$.

(ii) A pair $(e, \bar{e})$ is a *saturated tail pair* at node j if both e and $\bar{e}$ are directed from j, the tails of e and $\bar{e}$ are saturated and $e \in T(\bar{e})$.

(iii) A pair $(e,\bar{e})$ is an *unsaturated pair* at node j if either

(a) e and $\bar{e}$ are both directed into j, the head of e is unsaturated and $\bar{e}$ is nonvoid.

(b) e and $\bar{e}$ are both directed from j, e is nonvoid and the tail of $\bar{e}$ is unsaturated.

(c) e is directed into j, $\bar{e}$ is directed from j, and the head of e and the tail of $\bar{e}$ are unsaturated.

(d) e is directed from j, $\bar{e}$ is directed into j and both e and $\bar{e}$ are nonvoid.

Let $P = (e_1,\ldots,e_p)$ be an augmenting path with respect to f. Clearly each pair of arcs (e_i,e_{i+1}), $i = 1,\ldots,p-1$, is admissible with respect to f. In order for arcs e_1 and e_p each to be contained in two admissible pairs in the path (one pair for the head and one for the tail of each arc), we introduce two *virtual* arcs * and **, incident with nodes s and t respectively. Hereafter when we refer to the consecutive pairs of arcs in P we mean by convention to include the pairs $(*,e_1)$ and $(e_p,**)$ which are considered to be unsaturated. This convention enables us to avoid the statement of exceptions and special cases.

<u>Lemma 8.1</u>. *Let* P *be a shortcut-free augmenting path. With reference to any node* j *and the consecutive pairs of arcs in* P:

(i) P *contains at most one unsaturated pair at* j.

(ii) *All unsaturated head pairs at* j *precede the unsaturated pair, in any, followed by all saturated tail pairs.*

(iii) *If* P *contains saturated head pairs* $(e_1,\bar{e}_1),\ldots,(e_k,\bar{e}_k)$ *at node* j, *in that order, then the sets*

$$H(e_1) \cup \ldots \cup H(e_i), \quad 1 \le i \le k ,$$

remain saturated after augmentation of the flow along P.

(iv) *If* P *contains saturated tail pairs* $(e_1,\bar{e}_1),\ldots,(e_k,\bar{e}_k)$ *at node* j, *in that order, then the sets*

$$T(\bar{e}_i) \cup \ldots \cup T(\bar{e}_k), \quad 1 \leq i \leq k ,$$

remain saturated after augmentation of the flow along P.

<u>Proof</u>: (i) Suppose P were to contain two unsaturated pairs $(a,\bar{a})$, $(b,\bar{b})$ in that order. Then $(a,\bar{b})$ would be admissible and P would have a shortcut.

(ii) Suppose an unsaturated pair $(a,\bar{a})$ were to precede a saturated head pair $(b,\bar{b})$. Then $(a,\bar{b})$ would be admissible and P would have a shortcut. Similar analyses of other cases completes the proof of (ii).

(iii) If a saturated head pair $(a,\bar{a})$ is followed either by another saturated head pair $(b,\bar{b})$ or by an unsaturated pair $(b,\bar{b})$ in P, then $\bar{b} \notin H(a)$, else P would have a shortcut. It follows that for each i, $1 \leq i \leq k$,

$$\mu(H(e_1) \cup \ldots \cup H(e_i)) \geq 0,$$

where μ is defined as in the proof of Lemma 4.1. Therefore the set $H(e_1) \cup \ldots \cup H(e_i)$ must remain saturated after augmentation.

(iv) Proof is similar to that for (iii). □

9. <u>Augmenting Path Capacity and the Integrality Theorem</u>

With respect to a given flow f, let us define the value $\delta(e,\bar{e})$ for an admissible arc pair $(e,\bar{e})$ at node j as follows.

$$\delta(e,\bar{e}) = \min\{\delta_1(e,\bar{e}),\ \delta_2(e,\bar{e})\},$$

where

$$\delta_1(e,\bar{e}) = \begin{cases} f(e) & \text{if } e \in A_j \\ \min\{\beta_j(X)-f(X) \mid e \in X \subseteq B_j - \{\bar{e}\}\} & \text{if } e \in B_j. \end{cases}$$

$$\delta_2(e,\bar{e}) = \begin{cases} \min\{\alpha_j(X) - f(X) \mid \bar{e} \in X \subseteq B_j - \{e\}\} & \text{if } \bar{e} \in A_j. \\ f(\bar{e}) & \text{if } \bar{e} \in B_j \end{cases}$$

By definition, $\delta_1(*,\bar{e}) = \delta_2(e,**) = +\infty$.

Let us define $\delta(P)$ to be the minimum of the values $\delta(e_i,e_{i+1})$ over all consecutive pairs of arcs (e_i,e_{i+1}), $i = 0,\ldots,p$, in an augmenting path $P = (e_1,\ldots,e_p)$. (Here $e_0 = *$, $e_{p+1} = **$, as before.) An arc pair (e_i,e_{i+1}) is said to be *critical* if $\delta(P) = \delta(e_i,e_{i+1})$.

Theorem 9.1. *If* P *is a shortcut-free augmenting path with respect to flow* f, *the maximum possible augmentation of the flow along* P *is* $\delta(P)$. *Furthermore, after augmentation of the flow by* $\delta(P)$, *each critical pair in* P *is inadmissible with respect to the augmented flow.*

Proof: Let δ be the maximum amount of augmentation possible along P. We first show that $\delta \geq \delta(P)$.

If δ is determined by the flow through a backward arc in P, then clearly $\delta \geq \delta(P)$, from the definition of $\delta(P)$. So suppose δ is determined by an inequality of the form

$$f'(X) = f(X) + \delta\mu(X) \leq \alpha_j(X),$$

where $\mu(X)$ is defined as in the proof of Lemma 4.1. By that lemma, $\delta > 0$, hence we can assume that X is unsaturated by f and that $\mu(X) \geq 1$. Hence X contains at least one consecutive pair of arcs $(e,\bar{e})$ such that $\bar{e}$ is a forward arc, $\bar{e} \in X \cap A_j$ and $e \notin X$. Let $(e,\bar{e})$ be the first such arc pair in P. Let $(e_1,\bar{e}_1),\ldots,(e_k,\bar{e}_k)$ be the saturated tail pairs at node j in P.

If the tail of $\bar{e}$ is unsaturated, let

$$X' = X \cup T(\bar{e}_1) \cup \ldots \cup T(\bar{e}_k), \tag{9.1}$$

and if the tail of $\bar{e}$ is saturated, let

$$X' = (X \cap T(\bar{e}_i)) \cup T(\bar{e}_{i+1}) \cup \ldots \cup T(\bar{e}_k), \tag{9.2}$$

where $\bar{e}_i = \bar{e}$. By assumption, X is saturated by f', hence X' is also saturated, by Lemmas 2.2 and 8.1. X' has been constructed to have exactly one forward arc which is not paired with a backward arc, so $\mu(X') = 1$. It follows that

$$f'(X') = f(X') + \delta = \alpha_j(X').$$

Because $\bar{e} \in X' \subseteq A_j - \{e\}$, we have

$$\delta = \alpha_j(X') - f(X') \geq \delta(e,\bar{e}) \geq \delta(P).$$

We shall now show that augmentation by an amount $\delta(P)$ renders each critical pair inadmissible with respect to the augmented flow. (Consequently $\delta \leq \delta(P)$.) Let $(e,\bar{e})$ be a critical arc pair at node j. If e or $\bar{e}$ is a backward arc and $\delta(e,\bar{e}) = f(e)$ or $\delta(e,\bar{e}) = f(\bar{e})$, then $(e,\bar{e})$ obviously becomes inadmissible. So suppose $\bar{e}$ is a forward arc and

$$\delta(e,\bar{e}) = \alpha_j(X) - f(X),$$

for some X such that $\bar{e} \in X \subseteq A_j - \{e\}$. If $\mu(X) \geq 1$ we are done. So suppose $\mu(X) \leq 0$. From the definition of $\delta(e,\bar{e})$ it must be the case that $\bar{e}$ is a forward arc in P. Now construct the set X' as in (9.1) or (9.2), depending upon whether or not the tail of $\bar{e}$ is unsaturated. By applying Lemma 2.1, we see that

$$\alpha_j(X') - f(X') \leq \alpha_j(X) - f(X) = \delta(e,\bar{e}) = \delta(P).$$

Hence augmentation by an amount $\delta(P)$ renders $(e,\bar{e})$ inadmissible. $\square$

From the definition of $\delta(P)$, it is evident that if the capacity functions and the flow f are integer-valued, then $\delta(P)$ is also a positive integer. The theorem below then follows immediately.

Theorem 9.2. (Integrality Theorem) *If all capacity functions are integer-valued, then there is a maximal flow* f *which is integral. Moreover,* f *can be obtained by a finite number of augmentations along shortcut-free augmenting paths, beginning with the zero flow.*

10. The Splicing Lemma

We now prove a technical lemma needed in the following section. For the purpose of this lemma, (e,e), for any arc e, is by definition an admissible pair with respect to any flow. (Note also that if e is directed from i to j, there are actually two such admissible pairs, one at i and one at j.)

Lemma 10.1. (Splicing Lemma) *Suppose flow* f *is augmented along a shortcut-free path* P *to obtain flow* f'. *If* $(e,\bar{e})$ *is an admissible pair with respect to* f' *but not with respect to* f, *then* P *contains two arcs* x *and* $\bar{x}$, *with* x *preceding* $\bar{x}$ *(and possibly with* $x = \bar{e}$ *or* $\bar{x} = e$*), such that* $(e,\bar{x})$ *and* $(x,\bar{e})$ *are admissing pairs with respect to* f.

Proof:

Case 1. $(e,\bar{e})$ is a saturated head pair at j with respect to f'. In this case, $\bar{e} \neq **$ and $f(\bar{e}) > 0$.

(i) If the head of e is unsaturated by f, then $f(\bar{e}) = 0$, else $(e,\bar{e})$ would be admissible with respect to f. Because $f'(\bar{e}) > 0$, $\bar{e}$ is a forward arc in P. Let $\bar{x}$ be the arc immediately following $\bar{e}$. Then $(e,\bar{x})$ and $(\bar{x},\bar{e})$, where $x = \bar{e}$, are admissible pairs with respect to f.

(ii) If the head of e is saturated by f and $f(\bar{e}) = 0$, then the argument in (i) applies.

(iii) If the head of e is saturated by f and $f(\bar{e}) > 0$, then $\bar{e} \notin H(e)$, else $(e,\bar{e})$ would be admissible with respect to f. (Here and below H(e) and H'(e) denote head sets with respect to f and f'.) But H(e) is not saturated by f', else we would have $H'(e) \subseteq H(e)$ and $\bar{e} \notin H'(e)$. It follows that P contains at least one backward arc $\bar{x}$, where $\bar{x} \in H(e)$. Let $\bar{x}$ be the last such arc in P and let x be the immediately preceding arc. If $(x,\bar{x})$ is either an unsaturated head pair or a saturated head pair with $\bar{e} \in H(x)$ then we are done; $(e,\bar{x})$ and $(x,\bar{e})$ are admissible with respect to f. So suppose $(x,\bar{x})$ is a saturated head pair with $\bar{e} \notin H(x)$. By Lemma 8.1, P contains no unsaturated pairs at the same node, prior to $(x,\bar{x})$. Let $(x_1,\bar{x}_1),\ldots,(x_i,\bar{x}_i)$ be the saturated head pairs in P, up to and including $(x,\bar{x})$. By Lemmas 2.2 and 8.1, $X = H(e) \cup H(x_1) \cup \ldots \cup H(x_i)$ is saturated by f. By construction, $\mu(X) \geq 0$ so X remains saturated by f'. But $e \in X$, hence $H'(e) \subseteq X$ and $\bar{e} \in X$. Thus for some x_i, $1 \leq i \leq k-1$, $\bar{e} \in H(x_i)$. Now choose x to be x_i and $(x,\bar{e})$ is admissible with respect to f.

Case 2. $(e,\bar{e})$ is a saturated tail pair with respect to f'. The proof for this case is similar to that for Case 1.

Case 3. $(e,\bar{e})$ is an unsaturated pair with respect to f'. We consider the case that $(e,\bar{e})$ is an unsaturated pair at a node into which both e and $\bar{e}$ are directed; the proofs for other cases are similar. If the head of e is unsaturated by f or if $f(\bar{e}) = 0$, then the argument in Case 1 (i) applies. So suppose the head of e is saturated by f and $f(\bar{e}) > 0$. Then there is a backward arc $\bar{x}$ in P such that $\bar{x} \in H(e)$. (Possibly $\bar{x} = e$). Let x be the arc immediately preceding $\bar{x}$ in P. Clearly $(x,\bar{e})$ and $(e,\bar{x})$ are admissible pairs with respect to f. □

11. Bounding the Number of Augmentations

We now wish to obtain a polynomial bound on the number of augmentations required to compute a maximal flow, using the splicing lemma as a tool. This requires a bit more notation. With respect to a given feasible flow f, let $\sigma_j(e)$ denote the number of (nonvirtual) arcs in a shortest path from node s to node j, such that each successive pair of arcs in the path is admissible with respect to f and e is the last arc in the path (either $e \in A_j$ or $e \in B_j$). Similarly, let $\tau_j(e)$ denote the length of a shortest path from node j to node t which begins with arc e. Recall that by convention each augmenting path starts with virtual arc * and ends with **. Let $\sigma_s(*) = \tau_t(**) = 0$.

Lemma 11.1. *Suppose flow* f *is augmented along a shortest augmenting path* P *to obtain flow* f'. *For each node* j *and arc* e,

$$\sigma_j(e) \leq \sigma_j'(e), \quad \tau_j(e) \leq \tau_j'(e),$$

where σ', τ' and σ, τ denote path lengths with respect to f' and f.

Proof: Assume that for some j and e,

$$\sigma_j'(\bar{e}) < \sigma_j(\bar{e})$$

and let

$$\sigma_j'(\bar{e}) = \min_{i,x} \{\sigma_i'(x) | \sigma_i'(x) < \sigma_i(x)\} .$$

There is some arc e preceding $\bar{e}$ in a shortest path from s to j with respect to f' (else $\bar{e} = *$, $j = s$ and we would have $\sigma_s'(*) = 0 < \sigma_s(*) = 0$). Let k be the node which both e and $\bar{e}$ are incident to. By assumption, $\sigma_k'(e) \geq \sigma_k(e)$, so $(e,\bar{e})$ is not admissible with respect to f (else $\sigma_j'(e) \geq \sigma_j(e)$). Since $(e,\bar{e})$ is admissible with respect to f', we know by the splicing lemma that P contains arcs x, $\bar{x}$ such that $(e,\bar{x})$ and $(x,\bar{e})$ are admissible pairs at node k. It must be that $\sigma_k(x) \leq \sigma_k(e)$, else P would not be a shortest augmenting path with respect to f. But then

$$\sigma_j(\bar{e}) \leq \sigma_k(x) + 1 \leq \sigma_k(e) + 1 \leq \sigma_k'(e) + 1 = \sigma_j'(\bar{e}),$$

which contradicts our assumption that $\sigma_j(\bar{e}) > \sigma_j'(\bar{e})$. The proof that $\tau_j(e) \leq \tau_j'(e)$ is similar. □

Corollary 11.2. *If augmentations are made along shortest augmenting paths, then the number of arcs in successive augmenting paths is nondecreasing.*

Proof: The length of a shortest augmenting path is $\min_e\{\tau_s(e)\}$.

Apply Lemma 11.1 and note that

$$\min_e\{\tau_s'(e)\} \geq \min_e\{\tau_s(e)\}. \qquad \square$$

Suppose we carry out successive augmentations along shortest augmenting paths. The augmentations made along paths with the same number of arcs will be said to constitute a *phase* of the maximal flow computation. There can be no more than m phases, where m is the number of arcs in the network, since no path can contain more than m arcs.

In order to bound the number of augmentations within a phase we introduce a lexicographic ordering to break ties between shortest paths. The arcs are indexed arbitrarily. Given two paths P and P' with the same number of arcs, the rule to determine if P is *lexicographically smaller* than P', written $P \leq P'$, is as follows. If the index of the last arc in P is smaller than the index of the last arc in P', then $P \leq P'$. If these arcs are the same, then compare the indices of the next-to-last arcs, and so on. If all arcs are the same, then $P = P'$ and $P' \leq P$.

It is easy to modify the maximal flow algorithm so as to insure that each augmentation is made along a lexicographically minimal shortest path. Modify Step 2 so that the arcs at level ℓ are scanned in order of increasing index. Also modify the algorithm so that the arc e identified in Step 3 (as the last arc of an augmenting path) has the smallest possible index. These are the only changes necessary.

Lemma 11.3. *Suppose flow* f *is augmented along a lexicographically minimal shortest augmenting path* P *to obtain flow* f' *and that* $\sigma_j(e) = \sigma_j'(e)$*, for some given node* j *and arc* e. *If* Q, Q' *are lexicographically minimal paths realizing the values* $\sigma_j(e)$, $\sigma_j'(e)$ *respectively, then* $Q \leq Q'$.

Proof: Assume that for some j and $\bar{e}$, $\sigma_j(\bar{e}) = \sigma'_j(\bar{e})$ and that $Q \not\leq Q'$. Further assume, without loss of generality, that the next-to-last arcs in Q and Q' are different, these being e and e', respectively. Since $Q \not\leq Q'$ the index of e is greater than the index of e', therefore $(e',\bar{e})$ is not an admissible pair with respect to f. Since $(e',\bar{e})$ is admissible with respect to f' but not f, by the splicing lemma P contains a pair $(x,\bar{x})$ such that $(x,\bar{e})$, and $(e,\bar{x})$ are admissible with respect to f. The index of e' is greater than or equal to the index of x, else P is not a lexicographically minimal path. Hence the index of e is greater than the index of x, by our previous observation. But the index of e is less than or equal to the index of x, else Q is not lexicographically minimal. This contradicts our assumption that $Q \not\leq Q'$. □

Lemma 11.4. *If augmentations are made along lexicographically minimal shortest augmenting paths, then an arc pair* $(e,\bar{e})$ *which is critical in a path* P *cannot appear in any later path* P' *in the same phase of the computation.*

Proof: By contradiction. Assume that a path P with respect to flow f contains a critical pair $(e,\bar{e})$ which is contained in a later path P" with respect to flow f", in the same phase. Let k be the node to which both e and $\bar{e}$ are incident and let j be the other node to which $\bar{e}$ is incident. We know that $(e,\bar{e})$ is not admissible with respect to f', the flow existing after augmentation along P. The lexicographically minimal shortest path Q' realizing $\sigma'_j(\bar{e})$ contains some arc x as its next-to-last arc, with index x > greater than index e. But we must also have index e > index x, else either P" or Q is not lexicographically minimal. This yields the desired contradiction. □

Theorem 11.5. *If augmentations are made along lexicographically minimal shortest augmenting paths, then a maximal flow is achieved with at most* m^3 *augmentations, where* m *is the number of arcs in the network.*

Proof: There are at most m^2 possible arc pairs. At least one pair is critical in each augmenting path and cannot be contained in a later augmenting path in the same phase. Hence there are at most m^2 augmentations in each phase. There are at most m phases and so at most m^3 augmentations in all. □

This result can be compared with that of Edmonds and Karp [3] for the classical network flow model. In that case there are at most n phases, where n is the number of nodes, and at most m augmentations per phase.

12. Complexity of the Maximal Flow Algorithm

The maximal flow algorithm requires that certain computations be carried out with respect to the capacity functions and a given feasible flow f. Until this point, we have not discussed how these computations can and should be carried out. There are two types of problems we should like to be able to solve. In general polymatroidal terms these are as follows:

(i) *The saturation problem.* Given a polymatroid (E,ρ), a feasible function f and an element $e \in E$, is e saturated by f?

(ii) *The* δ *problem.* Given a polymatroid (E,ρ), a feasible function f and an element $e \in E$, what is the maximum value δ such that f' is feasible, where $f'(e) = f(e) + \delta$ and $f'(e') = f(e')$ for $e' \neq e$?

We must be able to solve the saturation problem in order to perform scanning and labeling in Steps 1 and 2 of the algorithm. Notice that any procedure solving the saturation problem in time c can be used to find $S(e)$, the unique minimal saturated set containing e, in time $c|E|$. Simply set $f(e') = 0$, for each $e' \neq e$, one at a time. Then $e' \in S(e)$ if and only if e becomes unsaturated.

If we can solve the δ problem for each capacity function α_j or β_j, then we can compute $\delta(e_i, e_{i+1})$ for each consecutive pair of arcs in an augmenting path $P = (e_1, \ldots, e_p)$, as required to determine the path capacity $\delta(P)$ in Step 5. For example, suppose (e_i, e_{i+1}) is a saturated head pair at node j. Then we can solve the δ problem for the polymatroid (B_j, β_j), $e_i \in B_j$, and $\bar{f}$, where $\bar{f}(e_{i+1}) = 0$ and $\bar{f}(e) = f(e)$ for $e \neq e_{i+1}$. The smaller of this value of δ and $f(e_{i+1})$ is then $\delta(e_i, e_{i+1})$. Notice also that any procedure solving the δ problem also solves the saturation problem. (An element $e \in E$ is saturated if and only if $\delta = 0$.)

If ρ is arbitrary and defined only by an explicit listing of values, i.e. $\rho(A)$ for each $A \subseteq E$, we cannot expect to be able to solve these problems efficiently, unless possibly $|E|$ is small. However, if ρ has some special structure, we may be more fortunate. For example, if ρ is a function whose value is determined by the cardinality of the set $A \subseteq E$. In this case there are numbers $s_1 \geq s_2 \geq \ldots > s_{|E|} \geq 0$ such that

$$\rho(A) = \sum_{i=1}^{|A|} s_i .$$

Both the saturation problem and the δ problem can be solved in $O(|E| \log |E|)$ time, with a sort of the values $f(e)$. This situation arises in the solution of the scheduling problem dealt with in [11,12].

In order to be able to state a general bound on the complexity of the maximal flow algorithm, we shall suppose there are black-box subroutines (or "oracles") to which we can pass on much of the burden of the computation. For example, we might suppose that there is a subroutine, operating in time d, to solve the δ problem for each polymatroid (A_j,α_j), (B_j,β_j), $j = 1,\ldots,n$. Then we have the following result.

Theorem 12.1. *Suppose for each capacity function there is a subroutine for solving the δ problem in time d. Then a maximal flow can be computed in time $O(m^5 d)$, where m is the number of arcs in the network.*

Proof: At most m^3 augmentations are needed. Each augmentation requires that each of the m arcs be scanned at most once. The scanning of an arc e requires that the saturation problem be solved, in time d, and (in the worst case) that H(e) or T(e) be found, in time $d|A_j|$ or $d|B_j|$. Scanning and labeling can thus be seen to require at most $O(m^2 d)$ time per augmentation. Determining $\delta(P)$ for each augmenting path P requires at most O(md) time. The time for this, and all other operations, is dominated by the time required for scanning and labeling. □

In the case of matroid optimization algorithms, it is common to assume the existence of a subroutine which determines whether or not a given subset of elements is independent in a matroid. A natural generalization of such a subroutine is one which tests a given function for feasibility with respect to ρ. Suppose there exists such a subroutine and that it runs in time c. Let us also suppose that f and ρ are integer-valued. We can then proceed as follows.

To solve the saturation problem, let f' be such that $f'(e) = f(e) + 1$ and $f'(e') = f(e')$, $e' \neq e$, and apply the feasibility test to f'. Element e is unsaturated if and only if f' is feasible. To solve the

δ problem, carry out a bisection search on values of δ in the interval $[f(e), \rho(e)]$. For each trial value of δ, test whether f' is feasible, where $f'(e) = f(e) + \delta$, $f'(e') = f(e)$, for $e' \neq e$. $O(c \log_2 \rho(e))$ time is thus sufficient to solve the δ problem. (Moreover, with proper modification of the procedure, this bound can be achieved, without prior knowledge of $\rho(e)$.) This yields the following theorem.

Theorem 12.2. *Suppose for each capacity function there is a subroutine for testing the feasibility of a flow in time c. If all capacity functions are integer-valued, then a maximal flow can be computed in time* $O(m^4 c(m+\log r))$, *where r is the maximum value of any capacity function for any single arc.*

Proof: By analysis similar to that in the proof of Theorem 12.1, given the observations preceding this theorem. □

13. Concluding Remarks

The polymatroidal network flow model as formulated here was suggested by research of one of the authors on a machine scheduling problem [11,12]. The authors wish to acknowledge that the same network flow model (generalized to accommodate lower bounds and costs on arc flows) was independently formulated in the doctoral thesis of Rafael Hassin [6], supervised by Alan Hoffman. Hassin's thesis concerns rather different questions than those dealt with in the present paper and we believe there is very little overlap. The polymatroidal network flow model also bears at least some similarity to a model of Edmonds and Giles [4] in which submodular set functions are assigned to nodes instead of arcs. However, there appears to be very little, if any, relationship between the network flow model studied in this paper and the notion of "flows" in matroids. (Except that both provide generalizations of the classical network flow model.) Cf. Welsh [14].

References

[1] M. Aigner, T. A. Dowling, "Matching Theory for Combinatorial Geometries," Trans. Amer. Math. Soc., 158 (1971) 231-245.

[2] J. Edmonds, "Minimum Partition of a Matroid into Independent Subsets," J. Res. Nat'l Bureau Standards, 69B (1965) 67-72.

[3] J. Edmonds and R. M. Karp, "Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems," J. ACM, 19 (1972) 248-264.

[4] J. Edmonds and R. Giles, "A Min-Max Relation for Submodular Functions on Graphs," in Studies in Integer Programming (Proc. Workshop on Programming Bonn, 1975, P. L. Hammer, E. L. Johnson and B. H. Korte, eds.), Annals Discrete Math, 1 (1977) 185-204.

[5] S. Fujishige, "Algorithms for Solving the Independent Flow Problem," J. Opns. Res. Soc. Japan, 21 (1978) 189-204.

[6] R. Hassin, "On Network Flows," PhD Dissertation, Yale University, 1978.

[7] M. Iri and N. Tomigawa, "An Algorithm for Finding an Optimal Independent Assignment Set," J. Opns. Res. Soc. Japan, 19 (1976) 32-57.

[8] S. Krogdahl, "A Combinatorial Proof of Lawler's Matroid Intersection Algorithm," unpublished. 1975.

[9] E. L. Lawler, "Matroid Intersection Algorithms," Math. Programming 9 (1975) 31-56.

[10] E. L. Lawler, Combinatorial Optimization: Networks and Matroids, Holt, Rinehart and Winston, 1976.

[11] C. U. Martel, "Generalized Network Flows with an Application to Multiprocessor Scheduling," PhD Thesis, University of California, Berkeley, 1980.

[12] C. U. Martel, "Scheduling Uniform Machines with Release Times, Deadlines and Due Times," to appear in J. ACM.

[13] P. Schönsleben, "Ganzzahlige Polymatroid-Intersektions - Algorithmen," thesis, ETH, Zürich, 1980.

[14] D. J. A. Welsh, Matroid Theory, Academic Press, 1976.