Submodular Flows

*András Frank**

Research Institute for Telecommunication,
Budapest, Hungary

The submodular flow model, due to J. Edmonds and R. Giles, is a common generalization of network flows, polymatroid intersections, and directed cut coverings. Here we outline a combinatorial method, developed in earlier papers, for solving the submodular flow optimization problem. Some applications and theoretical consequences are also discussed.

1. Introduction

In the last couple of years a large number of papers have appeared on sub- and supermodular functions. It turned out that these functions play a unifying role in combinatorial optimization. In [17] a beautiful survey can be found about the connections between the various models.

One of the most general frameworks is due to Edmonds and Giles [4]. Their model includes the minimum cost flow, polymatroid intersection, directed cut covering (Lucchesi-Younger), and orientation (Nash-Williams) problems. From the algorithmical point of view, Grötschel, Lovász, and Schrijver [11] have discovered a good algorithm for the Edmonds-Giles problem, based on the ellipsoid method. But it is desirable to have a purely combinatorial algorithm that only consists of steps like making an auxiliary digraph, finding augmenting paths, etc. Such kinds of methods were known for the minimum cost flow [5] for the matroid intersection [2,6] and for the Lucchesi-Younger problem [9] among the above special cases. The polymatroid intersection problem was solved only for the case of (0,1) objectives [14,16].

As far as the general Edmonds-Giles problem is concerned, a polynomial time combinatorial algorithm was developed in [7] for the case when the variables are bounded by 0 and 1. [8] contains a method for finding a feasible solution when the bounds on the variables are

*Research partly supported by Sonderforschungsbereich 21 /DFG/, Institut für Operations Research, Universität Bonn.

ISBN 0-12-566780-9

arbitrary numbers. Making use of these algorithms along with a scaling technique, quite recently we were able to develop a polynomial time algorithmn for the general Edmonds-Giles problem [1]. That algorithm looks important even in the special case of polymatroid intersections since before that, as mentioned, efficient combinatorial algorithms were known only for the case of (0,1) weights.

The purpose of the present paper is to summarize the method, its theoretical consequences, and some applications. However, the present approach is different from the one in [1] where the concept of restricted Edmonds-Giles problems was used. Here we work throughout on the original problem so the algorithm and the proof of its validity can be discussed directly.

It should be emphasized that the algorithm needs an oracle which can, roughly, minimize a submodular function. In the applications we exhibit in Section 9 this oracle is indeed available via a combinatorial algorithm.

2. Preliminaries

Throughout the paper we work with a finite ground set V of n elements. If $A \subseteq V$, then $\bar{A}$ denotes $V-A$. Sets A,B are *intersecting* if none of $A \cap B$, $A-B$, $B-A$ is empty. If, in addition, $A \cup B \neq V$ then A,B are *crossing*. A family $\mathbf{B}$ of subsets of V is *intersecting (crossing)* if $A \cap B$, $A \cup B \in \mathbf{B}$ for all intersecting (crossing) sets $A,B \in \mathbf{B}$. $\mathbf{B}$ is called a *ring* family if it is closed under taking union and intersection. For intersecting (crossing) families we can assume without loss of generality that $\emptyset \notin \mathbf{B}$ ($\emptyset, V \notin \mathbf{B}$). A family of subsets is *laminar* if it does not contain two intersecting sets.

A set function b is *submodular* on A,B if $b(A)+b(B) \geq b(A \cap B)+b(A \cup B)$. If the reverse inequality holds, b is called *supermodular* if equality holds, b is *modular* on A,B. Sometimes we refer to a pair $(b, \mathbf{B})$ as an *intersecting (crossing)* submodular function if $\mathbf{B}$ is an intersecting (crossing) family and b is a function on $\mathbf{B}$ submodular on intersecting (crossing) pairs. An intersecting submodular function $(b, \mathbf{B})$ and an intersecting supermodular function $(p, \mathbf{R})$ are said to be *compliant* if $B \in \mathbf{B}$ $P \in \mathbf{P}$, $B-P \neq \emptyset$, $P-B \neq \emptyset$ imply that $B-P \in \mathbf{B}$, $P-B \in \mathbf{P}$ and $b(B)-p(P) \geq b(B-P)-p(P-B)$.

A set A is called a $u\bar{v}$-set if $u \in A$, $v \notin A$. Let $G=(V,E)$ be a directed graph with node set V and arrow set E. (For directed edges we use the term arrow while an edge means an undirected edge.) Multiple arrows are allowed but loops not. An arrow uv *leaves (enters)* $B \subset V$ if B is a $u\bar{v}$-set ($v\bar{u}$-set). For a vector $x \in \mathbb{R}^E$, and $B \subset V$, $\rho_x(B)$ denotes $\sum(x(e): e$ enters $B)$ and $\partial_x(B)=\rho_x(\bar{B})$. $\lambda_x(B)$ denotes $\rho_x(B)-\partial_x(B)$. It is easy to check that λ_x is modular on pairs A,B of subsets of V.

Let $\mathbf{B}$ be a family of subsets of V. The *arrow-incidence* matrix B of $\mathbf{B}$ is a $(0, \pm 1)$ matrix with rows corresponding to the members of $\mathbf{B}$ and with columns corresponding to the arrows of G. An entry $b_{B,e}$ is $+1$ (-1) if e enters (leaves) B and O otherwise.

For an integer vector d by $\lfloor \frac{d}{2} \rfloor$ we mean a vector d' with components $d'(e)=\lfloor \frac{d(e)}{2} \rfloor$. We assume that addition, substraction, and comparison of two real numbers are one computational step each.

The following useful concept is due to Hoffman and Edmonds-Giles [13,4]. A linear system $Ax \leq b$ is called *totally dual integral* (TDI) if for any integral vector d the dual linear program min yb subject to $y \geq 0$, $yA=d$ has an integral optimal solution if it has an optimal solution. The basic feature of TDI systems is given by the following theorem.

THEOREM A TDI linear system defines a polyhedron spanned by its integer points provided that b is integral-valued.

3. Submodular Flows

Let $(b', \mathbf{B}')$ be a crossing submodular function and let $G=(V,E)$ be a directed graph. Let $f \in (R \cup \{-\infty\})^E$, $g \in (R \cup \{+\infty\})^E$ be capacities and $d \in R^E$ a weighting on the arrows. Let B' denote the arrow-incidence matrix of $\mathbf{B}'$ and consider the following dual pair of linear programs.

(1′) max dx subject to
$B'x \leq b'$ (or, equivalently $\lambda_x(B) \leq b'(B)$ for every $B \in \mathbf{B}'$)
$f \leq x \leq g$

(2′) min $b'y + gz - fw$ subject to

$$(y,z,w)\begin{bmatrix} B' \\ I \\ -I \end{bmatrix} = d$$

$$(y,z,w) \geq 0$$

where the components of y correspond to the members of $\mathbf{B}'$ and the components of both z and w correspond to the elements of E so that $w(e)=0$ if $f(e)=-\infty$ and $z(e)=0$ if $g(e)=+\infty$. ([I] is the identity matrix of appropriate size.) These linear programming problems were introduced by Edmonds and Giles [4]. We call a linear programming problem of form (1′) a *submodular flow* problem (or sometimes an Edmonds-Giles problem) and a solution to it is said to be a *submodular* *flow.*

THEOREM 1. [4] The linear system (1′) is TDI. Consequently, if d is integer valued and (2′) has an optimal solution, then (2′) has an integral optimal solution. If b', and (the finite components of) f and g are integral valued and (1′) has an optimal solution, then (1′) has an integral optimal solution.

The algorithm will find these optima. First, we shall be dealing with a simplified version of the submodular flow problem when the bound imposed on λ_x is an intersecting submodular function denoted by $(b, \mathbf{B})$. In Section 7 we shall indicate how the general crossing case can be reduced to this version. In the intersecting case we shall refer to the linear programs (1′) and (2′) as (1) and (2), respectively.

Henceforth we assume d to be integer valued. A rational d can be replaced by $D \cdot d$ where D is the common denominator of the components of d. For irrational d the algorithm does not work. Thus the algorithm in [7] for the 0−1 case has a slight and mostly theoretical advantage, namely, it does work well if the components of d are irrational. For example they may be of the form $a\sqrt{2} + b$ (a,b integers).

For simplicity, we require that no arrow $e = ab$ exists with $f(e) = -\infty$. In the contrary case, if $g(e) = +\infty$, replace e by $e_1 = ab$ and $e_2 = ba$ with $f(e_1) = f(e_2) = 0$, $g(e_1) = g(e_2) = +\infty$ and $d(e_1) = d(e)$, $d(e_2) = -d(e)$; if $g(e) < +\infty$, replace e by $e_1 = ba$ with $f(e_1) = -g(e)$, $g(e_1) = +\infty$ and $d(e_1) = -d(e)$.

The complementary slackness conditions are as follows.

(3a′) $z(e) > 0 \Rightarrow x(e) = g(e) \quad (<\infty)$ for $e \in E$.

(3b′) $w(e) > 0 \Rightarrow x(e) = f(e) \quad (>-\infty)$ for $e \in E$.

(3c) $y(B) > 0 \Rightarrow \lambda_x(B) = b(B) \quad$ for $B \in \mathbf{B}$.

Denote yb_e by $y(e)$ where b_e is the column vector of B corresponding to e. Suppose we have a feasible solution x to (1′) and a vector y for which

(3a) $y(e) > d(e) \Rightarrow x(e) = f(e) \; (> -\infty)$

(3b) $y(e) < d(e) \Rightarrow x(e) = g(e) \; (< +\infty)$.

By letting $w(e) = y(e) - d(e)$ for arrows satisfying (3a) and $z(e) = d(e) - y(e)$ for arrows satisfying (3b) and each other component of w and z is 0, the vector x and (y,z,w) satisfy (3a′b′c). Therefore our purpose is to determine algorithmically the vectors x,y satisfying (3abc).

4. Tight Sets and Potentials

Assume x is a solution to (1). A set $B \in \mathbf{B}$ is called *b-tight* or briefly *tight* with respect to x if $\lambda_x(B) = b(B)$. The following lemmas are taken from [7].

LEMMA 2. The intersection of two tight sets is tight. If a family of tight sets forms a connected hypergraph, its union is also tight.

Denote by $B_x(v)$ the intersection of tight sets containing v. By the lemma $B_x(v)$ is tight.

A fundamental feature of the method is that the dual variables associated with the members of $\mathbf{B}$ are not used during the algorithm. Instead, we work with potentials which are vectors in Z^V. At the end of the algorithm the optimal dual solution can be reconstructed from the final potential. To be more precise, assume we have a solution x to (1) and a potential Π such that

$$uv \in E,\ \Pi(v) - \Pi(u) > d(e) \Rightarrow x(e) = f(e) \qquad (4a)$$

$$uv \in E,\ \Pi(v) - \Pi(u) < d(e) \Rightarrow x(e) = g(e) \qquad (4b)$$

$$u \in B_x(v) \qquad \Rightarrow \Pi(u) \geq \Pi(v). \qquad (4c)$$

With the help of this potential Π we are going to define a vector y that, along with x, will satisfy (3abc).

Let the distinct values of Π be $\Pi_0 < \Pi_1 < \cdots < \Pi_k$ and $V_i = \{u\colon \Pi(u) \geq \Pi_i\}, i = 1,\ldots,k$.

LEMMA 3. (4c) is equivalent to the fact that each V_i partitions into tight sets.

Namely, the partition is formed by the components of the hypergraph $\{B_x(v)\colon v \in V_i\}$. Denote this family of components by $\mathbf{K}_x(V_i)$.

For $B \in \mathbf{B}$ define $y(B) = \sum(\Pi_i - \Pi_{i-1})$ where the summation extends over those subscripts i for which $B \in \mathbf{K}_x(V_i)$ (the empty sum is defined to be zero).

LEMMA 4. For each $e = uv \in E$, $\Pi(v) - \Pi(u) = y(e)$.

These lemmas imply that x and y satisfy (3abc) and y is integer if Π is. From algorithmical point of view, in order to get y we have to be able to determine $B_x(v)$. (Creating the components of a hypergraph is easy).

5. Strategy of the Algorithm

Our remaining purpose is to find a solution x to (1) and a potential Π which satisfy the optimality criteria (4abc). The algorithm starts with an arbitrary submodular flow which is found by an algorithm described in [8].

We shall need the scaling technique. It was introduced by Edmonds and Karp [3] for solving the minimum cost flow problem. They scaled the capacities. In our case however, this does not seem to work since difficulties arise from the fact that $\lfloor \frac{b}{2} \rfloor$ is not submodular. This is why we are going to scale the cost function d. To this end suppose that $d(e)$ is given in base 2 and that the biggest non-zero digit is 2^{K-1}, i.e. the number $\max_{e \in E} |d(e)|$ consists of K digits.

The basic idea is as follows. If one already has a solution x and a potential Π satisfying (4abc) with respect to a cost function d' then it is possible to determine another x' and Π' satisfying (4abc) with respect to a new cost function d'' where d'' differs from d' in one component by one. This will be done by the *Inner Algorithm*.

The *Level Procedure*, starting with x, Π satisfying (4abc) with respect to $d'' = \lfloor \frac{d'}{2} \rfloor$, finds a solution x' and a potential Π' satisfying (4abc) with respect to d'. If the Inner Algorithm is available, this is simple since x and 2Π satisfy (4abc) with respect to $2d''$ and $2d''$ differs from d' in any component by at most one. Therefore the Level Procedure is nothing but a series of applications of the Inner Algorithm, at most $|E|$ times, yielding the required x' and Π'.

Let $d_0 = d$ and $d_i = \lfloor \frac{d_{i-1}}{2} \rfloor$, $i = 1,2,\ldots,K$. Obviously $d_K(e) = 0$ if $d(e) \geq 0$ and $d_K(e) = -1$ if $d(e) < 0$. Any solution x' and $\Pi = 0$ satisfy (4abc) with respect to $d = 0$. Applying the Inner Algorithm at most $|E|$ times we obtain a vector x_K and a potential Π_K satisfying (4abc) with respect to d_K. We call this part of the algorithm the *Beginning Phase*. Then apply the Level Procedure K times: first to d_K, then to $d_{K-1}, \ldots,$ finally to d_1. This is called the *Level Phase*. The final x and Π satisfy (4abc) with respect to the original cost function d. One can see that the Beginning Phase needs at most $|E|$, while the Level Phase at most $K|E|$ applications of the Inner Algorithm.

In Section 8 we shall give a combinatorial good characterization of dual infeasibility. At this point it is important to know that, as can be proved, none of the intermediate problems are dual infeasible unless the original problem is dual infeasible.

In the next section we concentrate on the Inner Algorithm.

6. Inner Algorithm

The Inner Algorithm works with the following input and output.

Input

x: feasible solution to (1)

Π: potential

$e' = ab$: arrow in E

d': cost function

χ: $+1$ or -1

such that (4abc) holds.

Output

x': feasible solution to (1)

Π': potential

such that (4abc) holds with respect to the modified d_1 where $d_1(e) = d'(e)$ if e ae e' and $d_1(e') = d(e') + \chi$.

In the Beginning Phase, apply the Inner Algorithm with $\chi = -1$. The input x and Π almost satisfy (4abc) with respect to d_1. Only (4 a) can be violated by e'. In the Level Phase $\chi = 1$ and the starting x and Π almost satisfy (4abc) with respect to d_1. Only (4b) can be violated by e'. We shall be dealing only with this latter case. The algorithm is quite analogous when $\chi = -1$.

Suppose now that $e' = ab$ violates (4b) with respect to x, Π, and d_1, i.e. $d_1(ab) = \Pi(b) - \Pi(a) + 1$. Denote $d_1(uv) - \Pi(v) + \Pi(u)$ by $\bar{d}_1(uv)$. Define an auxiliary digraph H_x on V in which three kinds of arrows may exist having the following capacities.

1. $e_1 = uv$ is a (so-called *forward*) arrow if $uv \in E$, $x(uv) < g(uv)$ and $d_1(uv) = 0$. Its capacity is $c(e_1) = g(uv) - x(uv)$.

2. $e_2 = vu$ is a *(backward)* arrow if uv E, $x(uv) > f(uv)$ and $d_1(uv) = 0$. Its capacity is $c(e_2) = x(uv) - g(uv)$.

3. $e_3 = uv$ is a *(jumping)* arrow if there is no tight $u\bar{v}$-set and $\Pi(u) = \Pi(v)$. Its capacity if $c(e_3) = \min(b(B) - \lambda_x(B)$: $B \in \mathbf{B}$, B is a $u\bar{v}$-set).

(The minimum on the empty set is defined to be $+\infty$.)

One can see that all capacities are positive. Try to find a directed path in H_x from b to a. There may be two cases.

CASE 1. No path exists, i.e. $a \notin T = \{v$: v can be reached from b in $H_x\}$. Revise the potential as follows. $\Pi'(u) = \Pi(u) + 1$ if $u \in T$

and $= \Pi(u)$ otherwise. The next claim is a straightforward consequence of the optimality criteria (4abc) and the definition of H_x.

CLAIM $x' := x$ and Π' satisfy (4abc) with respect to d_1.

CASE 2. In H_x there exists a path from b to a.

Let A be such a path with a minimum number of arrows. Denote by Δ the least capacity of the arrows on $A + e'$. (Δ is called the capacity of the augmentation along A.) It can be shown that if $\Delta = +\infty$, the problem is dual infeasible. Therefore we can suppose that $\Delta < +\infty$.

Define a new vector x':

$$x'(uv) = \begin{cases} x(uv) + \Delta & \text{if } uv \in E \text{ is on } A \text{ or } uv = e'. \\ x(uv) - \Delta & \text{if } uv \in E \text{ and } vu \text{ is on } A. \\ x(v) & \textit{otherwise.} \end{cases}$$

We call this change an augmentation (of Δ amount). Call an arrow on the augmenting path *critical* if its capacity is Δ.

It is easy to see that:

LEMMA 5.[8] For each $B \in \mathbf{B}$, $\lambda_{x'}(B) = \lambda_x(B) + \Delta\,(\partial^J(B) - \partial^J(\bar{B}))$ where $\partial^J(B)$ stands for the number of jumping arrows on P leaving B.

The next lemma is crucial to the algorithm.

LEMMA 6. x' is solution to (1).

PROOF Obviously $f \le x' \le g$. Set $\epsilon(B) = b(B) - \lambda_x(B)$ for $B \in \mathbf{B}$. Then $\epsilon(B)$ is submodular on intersecting pairs. We are going to prove that $\partial^J(B)\cdot\Delta \le \epsilon(B)$ for each $B \in \mathbf{B}$. By Lemma 5 this already implies that $\lambda_{x'}(B) \le b(B)$, i.e. x' is a solution to (1).

Proceed by induction on the value $\partial^J(B)$. The case $\partial^J(B) = 0$ is trivial. Let $\partial^J(B) > 0$ and let uv be a jumping arrow on P leaving B such that $\Pi(v)$ $(=\Pi(u))$ is as large as possible. If there are more such arrows let uv be the first one on P (starting from b).

CLAIM. $\partial^J(B \cup B_x(u)) = \partial^J(B) - 1$.

PROOF. Since no jumping arrows leaves $B_x(u)$ and uv does not leave $B \cup B_x(u)$ we have $\partial^J(B \cup B_x(u)) \le \partial^J(B) - 1$. On the other hand if qr is another jumping arrow on P leaving B then we claim that $r \notin B_x(u)$ (and so qr leaves $B \cup B_x(u)$ too): in the contrary case $\Pi(r) \ge \Pi(u)$ by (4c) and therefore, by the maximal choice of uv, $\Pi(r) = \Pi(v) = \Pi(u)$. Hence ur is a jumping arrow in H_x. By the assumption on uv, uv precedes qr on P and so ur is a shortcut arrow to P contradicting the minimality of P.

Now we have

$$\begin{aligned} \epsilon(B) &= \epsilon(B) + \epsilon(B_x(u)) \\ &\ge \epsilon(B \cap B_x(u)) + \epsilon(B \cup B_x(u)) \\ &\ge \Delta + \Delta\cdot(\partial^J(B) - 1) \\ &= \partial^J(B) \end{aligned}$$

as required. Here we made use of the induction hypothesis for $B \cup B_x(u)$ and the previous claim. □

LEMMA 7. (4abc) holds again with respect to x', Π and d_1 with the only possible exception that e' still violates (4b). (This is the case exactly when $\Delta < g(e') - x(e')$.)

PROOF. The statement for (4a) and (4b) follows directly from the definition of H_x. We prove (4c). By Lemma 3, V_i is the union of disjoint sets $X_1, \ldots, X_r$ where each X_i is tight with respect to x. Since no jumping arrow leaves any tight set and no jumping arrow enters V_i we have $\partial^J(\bar{X}_i) = \partial^J(X_i) = 0$. Thus each X_i is tight with respect to x'. Apply again Lemma 3. □

Like the classical maximum flow algorithm, the Inner Algorithm consists of iterating the augmentation procedure. More precisely, in every loop of the iteration we apply the augmentation to an input x, Π which was the output x', Π of the previous augmentation, that is $x := x'$ and Π is unchanged. The Inner Algorithm terminates when either Case 1 occurs (and then we perform the potential change described there) or e' stops violating (4b) (since the current $\Delta = g(e') - x(e')$). Note that the potential Π remains unchanged during the whole Inner Algorithm except, possibly, at the very end if Case 1 occurs.

To justify the algorithm we have to prove that the number of subsequent augmentations can be bounded by a polynomial of $|V|$. To this end we always choose a shortest augmenting path. This kind of selection was proposed by Edmonds and Karp [3] in order to get a polynomial bound for the maximum flow algorithm. In addition, among the various shortest augmenting paths in a given stage we break ties by a lexicographic ordering. This technique was devised by Schönsleben [16] and Lawler and Martel [14] to obtain a vector of maximum component-sum in the intersection of two polymatroids.

Assume that the nodes of H_x have fixed (distinct) indices. For notational convenience we do not distinguish between the name and the index of a node. That is, for two nodes u, v, $u > v$ means that the index of u is bigger than that of v.

By a *shortest path* from b to a we mean one with a minimum number of arrows and this number is the *length* of the path.

$\sigma_x(u)$ $(\tau_x(t))$ stands for the length of a shortest path from b to u (u to a) in H_x. Call an arrow uv in H_x *admissible* if $\sigma_x(u) + \tau_x(v) + 1 = \sigma_x(a)$. Obviously, a shortest path from b to a consists of admissible arrows.

Let us define $i_x(v)$ as the minimum index u for which uv is admissible. If no such a u exists then $i_x(u) = \infty$. The nodes of the augmenting path P we will use are (in reverse order) $a, i_x(a), i_x(i_x(a)), \ldots, b$. None of these indices is ∞.

Let J_x denote the set of jumping arrows in H_x.

LEMMA 8. Suppose that $\sigma_x(v) > \sigma_x(u)$ and uv is a new jumping arrow in $H_{x'}$, that is $uv \notin J_x$, $uv \in J_{x'}$. There exists an arrow v_1u_1 on P such that $v_1u_1, v_1v, uu_1 \in J_x$ and $\sigma_x(u) = \sigma_x(v_1) = \sigma_x(v) - 1 = \sigma_x(u_1) - 1$.

PROOF. Since uv is a new jumping arrow in H_x, therefore $v \notin B_x(u)$ and $B_x(u)$ is not tight with respect to x'. Set $J(u) = \{w: \Pi(w) = \Pi(u), w \in B_x(u)\}$. Let B be a maximal set in B satisfying the following properties

$a.$ B is tight with respect to x,

$b.$ $w \in B$ implies that $\Pi(w) \geq \Pi(u)$,

$c.$ $w \in P$, $w \in B - J(u)$, $\Pi(w) = \Pi(u)$ imply that $\sigma_x(w) < \sigma_x(u)$, (5)

$d.$ B is a $u\bar{v}$-set.

This definition does make sense since $B_x(u)$ satisfies (5).

Since $uv \in J_x'$, B cannot be tight with respect to x'. Thus, by Lemma 5, $\partial^J(\bar{B}) > 0$. Let $v_1u_1 \in J_x$ be an arrow on P entering B. We are going to prove that v_1u_1 satisfies the requirements of the lemma. By Lemma 2, $B' = B \cup B_x(v_1)$ is tight with respect to x. If $w \in B_x(v_1)$ then $\Pi(w) \geq \Pi(v_1) = \Pi(u_1) \geq \Pi(u)$ therefore (5b) holds for B'. We show that (5c) is also true for B'. To this end let $w \in (P \cap B_x(v_1)) - B$ such that $\Pi(w) = \Pi(u)$. Then $\Pi(w) \geq \Pi(v_1) = \Pi(u_1) \geq \Pi(u)$ from which $\Pi(w) = \Pi(v_1)$. Thus either $v_1w \in J_x$ or $v_1 = w$. Since $w \neq u_1$ and $w \in P$ we have $\sigma_x(w) \leq \sigma_x(v_1) = \sigma_x(u_1) - 1 \leq \sigma_x(u)$. Consequently, B' satisfies (5abc). Being B maximal B' cannot be a $u\bar{v}$-set, that is $v \in B_x(v_1)$. Then $\Pi(v) \geq \Pi(v_1) = \Pi(u_1) \geq \Pi(u) = \Pi(v)$ therefore equality holds everywhere. Thus (i) $\sigma_x(v) \leq \sigma_x(v_1) + 1$. We claim that $u_1 \in J(u)$ and so $\sigma_x(u_1) \leq \sigma_x(u) + 1$. For otherwise, using (5c) for $w = u_1$ and (i), $\sigma_x(v) \leq \sigma_x(v_1) + 1 = \sigma_x(u_1) < \sigma_x(u)$, a contradiction.

Now we have $\sigma_x(u) + 1 \leq \sigma_x(v) \leq \sigma_x(v_1) + 1 =$

$\sigma_x(u_1) \leq \sigma_x(u) + 1$ from which equality follows everywhere. Furthermore u, v, u_1, v_1 are distinct nodes and $v_1v, uu_1, v_1u_1 \in J_x$. □

LEMMA 9. For $w \in V$ $\sigma_x(w)$ and $\tau_x(w)$ are non-decreasing.

PROOF. We prove the lemma for $\sigma_x(w)$. If uv is a new arrow in H_x, for which $\sigma_x(u) < \sigma_x(v)$ then $uv \in J_{x'}$. By Lemma 8 $\sigma_x(u) = \sigma_x(v) - 1$ therefore $\sigma_x(w)$ cannot decrease. □

By a *phase* we mean a maximal sequence of subsequent augmentations in which $\sigma_x(a)$ is unchanged. Obviously the number of phase is at most n.

LEMMA 10. In one phase $i_x(v)$ does not decrease.

PROOF. The only possibility for decreasing $i_x(v)$ would be a new jumping arrow uv arising by performing an augmentation. Apply Lemma 8 and consider those nodes v_1, u_1 of P. Then u_1v, uu_1, v_1u_1 are all admissible arrows in H_x. Thus $i_x(v) \leq v_1 = i_x(u_1) \leq u$, i.e. the new jumping arow uv does not reduce $i_x(v)$. □

LEMMA 11. After making an augmentation, a critical arrow disappears from the auxiliary digraph.

PROOF. The lemma is obvious if uv is a critical forward or backward arrow. Assume that $uv \in J_x$. We are going to prove the existence of a $u\bar{v}$-set B tight with respect to x'. Since uv is critical there exists a $u\bar{v}$-set B_1 for which $\Delta = b(B_1) - \lambda_x(B_1)$ and choose B_1 to be minimal.

CLAIM. $B_1 \subseteq B_x(u)$.

PROOF.
$\Delta = \epsilon(B_1) = \epsilon(B_1) + \epsilon(B_x(u)) \geq \epsilon(B_1 \cap B_x(u)) + \epsilon(B_1 \cup B_x(u)) \geq \Delta + 0$ from which $\epsilon(B_1 \cap B_x(u)) = \Delta$. By the minimality of B_1, $B_1 = B_1 \cap B_x(u)$, i.e. $B_1 \subseteq B_x(u)$.

Let B be a maximal set in B for which

a. B is a $u\bar{v}$-set,

b. $\epsilon(B) = \Delta$

(6) c. $w \in B$ implies $\Pi(w) \geq \Pi(u)$,

d. $\Pi(w) = \Pi(u)$, $w \in B \cap P$ imply that $\sigma_x(w) \leq \sigma_x(u)$.

The definition of B does make sense because B_1 satisfies (6).

CLAIM. There is no jumping arrow st on P entering B.

PROOF. Suppose on the contrary that st exists. Let $B' = B \cup B_x(s)$. We are going to prove that B' satisfies (6) which will contradict the maximal choice of B.

(6a): $v \in B_x(s)$ would imply $\Pi(v) \geq \Pi(s) = \Pi(t) \geq \Pi(u) = \Pi(v)$ whence $\Pi(s) = \Pi(v)$ and sv would be a jumping arrow in H_x and then $\sigma_x(v) \leq \sigma_x(s) + 1$. But this is impossible since

$\sigma_x(s) + 1 = \sigma_x(t) \le \sigma_x(u) = \sigma_x(v) - 1$, i.e. sv would be a shortcut arrow to P. Thus B' is a $u\bar{v}$-set.

(6b): $\Delta + 0 \ge \epsilon(B) + \epsilon(B_x(s)) \ge \epsilon(B') + \epsilon(B \cap B_x(s)) \ge \Delta + 0$ from which $\epsilon(B') = \Delta$.

(6c): If $w \in B_x(s)$ then $\Pi(w) \ge \Pi(s) = \Pi(t) \ge \Pi(u)$.

(6d): Let $w \in (B_x(s) \cap P) - B$ such that $\Pi(w) = \Pi(u)$. Then, because of $\Pi(w) \ge \Pi(s) = \Pi(t) \ge \Pi(u)$, we have $\Pi(w) = \Pi(s)$ and $sw \in J_x$. Since w ae t, either $w = s$ or w precedes s on P. Thus $\sigma_x(w) \le \sigma_x(s) = \sigma_x(t) - 1 \le \sigma_x(u) - 1$ and (6d) is true for B'.

In other words the claim says that $\partial^J(\bar{B}) = 0$. Hence $b(B) \ge \lambda_{x'}(B) = \lambda_x(B) - \Delta \quad (\partial^J(\bar{B}) - \partial^J(B)) = b(B) - \Delta - 0 + \Delta \cdot \partial^J(B) \ge b(B) - \Delta + \Delta$ from which $b(B) = \lambda_{x'}(B)$ follows and the proof of the lemma is now complete. □

LEMMA 12. If uv is a critical jumping arrow on an augmentation path P_1 then uv will no longer be a jumping admissible arrow during the whole phase.

PROOF. By Lemma 11 after augmenting along P_1 the arrow uv disappears from the auxiliary digraph. That time we had $i_{x_1}(v) = u$, thus, by Lemma 10 $i_x(v) \ge u$ during the whole phase. Assume now indirectly that later in the same phase we are making an augmentation of the current x along an augmentating path P so that uv becomes again a jumping admissible arrow. Applying Lemma 8 we have $u \le i_x(v) \le v_1 = i_x(u_1) \le u$ whence $u = v_1$ that is uv was a jumping arrow already in H_x, a contradiction. □

Summing up, by now we have proved that within one phase an arrow may be critical at most once. Since in H_x there may be three parallel arrows from u to v the number of subsequent augmentations is at most $3n^2$ in one phase and thus the overall number of augmentations is at most $3n^2$. Furthermore if the input data b, f, g are all integral then all the arithmetic is integral and the final submodular flow is also integral.

In order to be able to apply the Inner Algorithm we need an oracle to

(*) compute the min value of $b(B) - \lambda_x(B)$ over the $u\bar{v}$-members of **B**.

With the help of this oracle we can determine the auxiliary digraph H_x as well as the capacities of jumping arrows in H_x. Assume this oracle is available with complexity h. One augmenting path and the new H_x, with the capacities can be computed in $0(n^2h)$ steps. Thus the overall complexity of the Inner Algorithm is $0(n^5h)$. We have seen that the Inner Algorithm needs to be applied at most $(K+1)$ e times where

$e = |E|$. Consequently, an optimal solution to (1) and a potential satisfying (4abc) can be obtained in at most $0(n^5ehK)$ steps.

Finally we briefly remark that the present algorithm can be considered as a generalization of the method for finding a feasible solution to (1) [8], consequently the present algorithm can be applied to find a starting feasible solution. To this end adjoin a new node r to the graph along with arrows from v to r for each $v \in V$. For a new arrow vr set $f(vr) = 0$, $g(vr) = \infty$ $d(vr) = -1$ and for an old arrow uv set $d(uv) = 0$. The original submodular problem has a feasible solution if and only if the new one has a solution of 0 cost. Thus we can apply the optimal submodular flow algorithm to this new problem. Notice that a starting feasible solution to the new probem is easily available by taking $x(uv) = 0$ for $uv \in E$ and $x(vr) = \max(0, -\min_{v \in B} b(B))$ for $v \in V$.

Incidently this trick gives rise to a feasibility criterion. See Section 8.

7. Crossing Families

In this section we indicate how the methods developed for the intersecting case lead to a solution for the general crossing problem. The following lemma was proved in [7].

LEMMA 13. For a crossing submodular function $(b', \mathbf{B}')$ define a function b on as follows. Set $\mathbf{B} = \{X: X \text{ ae } \emptyset, X = \cap X_i, X_i \in \mathbf{B}', \bar{X}_i \cap \bar{X}_j = \emptyset\} \cup \{V\}$ and $b(X) = \min(\sum b'(X_i): X = \cap X_i, X_i \in \mathbf{B}', \bar{X}_i \cap \bar{X}_j = \emptyset)$ for $X \in \mathbf{B} - \{V\}$ and $b(V) = 0$. Then $(b, \mathbf{B})$ is an intersecting submodular function. Moreover the submodular polyhedron P' defined by (1′) is exactly the submodular flow polyhedron P defined by (1).

This lemma makes it possible to apply the algorithm developed for intersecting submodular functions. As far as the oracle (*) is concerned the content of the next lemma is that an oracle for b' gives the right answer with respect to b as well. [8]

LEMMA 14. For $x \in P' (=P)$, $u, v \in V$ we have
$\min(b(B) - \lambda_x(B): B \in \mathbf{B}, B \text{ is a } U\bar{V}\text{-set})$
$= \min(b'(B) - \lambda_x(B): B \in \mathbf{B}', B \text{ is a } u\bar{v}\text{-set})$.

These lemmas show that in order to determine the optimal x and Π case can be used without any change for crossing submodular functions providing that a starting solution is available. The problem of finding a starting solution can also be reduced to the intersecting case but a bit more sophisticated trick is needed. See [8].

To construct the optimal dual solution needs some more work. In [7] a simple combintorial procedure was shown, given x and Π

satisfying (4abc), for computing the optimal solution to the dual of (1'), which will in addition be integer-valued if Π is.

8. Feasibility and Optimality

The next results are taken from [1,7,8].

THEOREM 15. The linear system (1') has a solution if and only if

$$\rho_f(\cup B_i) - \partial_g(\cup B_j) \leq b'(B_{ij})$$

for disjoint non-empty sets $B_1, B_2, \ldots, B_k$ (possibly not in B') where each B_i is the intersection of pairwise co-disjoint members B_{ij} of B' ($j = 1,2,\ldots,k_i$). Moreover, if b', f, g are integral-valued and the conditon holds, then (1') has an integral valued solution.

It should be noted that the condition becomes much simpler if B' is a ring-family and b' is submodular on every pair. In this case it is necessary and sufficient for (1') to have a solution that $\rho_f(B) - \partial_g(B) \leq b'(B)$ for $B \in B'$ If B' consists of all subsets of V and b' is identically zero, we get back Hoffman's circulation theorem [5].

In order to formulate dual feasibility conditions let us define a digraph $H = (V,F)$ and a cost function d' on its arrows as follows.

$e = uv \in F$ if $uv \in E$ and $g(uv) + \infty$. Set $d'(e) = -d(e)$.

$e = vu \in F$ if $uv \in E$ and $f(uv) = -\infty$. Set $d'(e) = d(e)$.

$e = uv \in F$ if there is no $u\bar{v}$-set in **B**. Set $d'(e) = 0$.

THEOREM 16. The linear programming dual to (1') has a solution if and only if H does not possess a directed circuit of negative cost.

The following theorem gives a criterion for a submodular flow to be optimal. Let x be a solution to (1'). Let us define a digraph

$G_x = (V,E_x)$ and a cost function d' on its arrows.

$e = uv \in E_x$ if $uv \in E$ and $x(uv) < g(uv)$. Set $d'(e) = -d(e)$.

$e = vu \in E_x$ if $uv \in E$ and $x(uv) > f(uv)$. Set $d'(e) = d(e)$.

$e = uv \in E_x$ if there is no b-tight $u\bar{v}$-set in **B**. Set $d'(e) = 0$.

THEOREM 17. A submodular flow x is an optimal solution to (1') if and only if there is no negative directed circuit in G_x.

An interesting consequence of Theorem 15 was derived in [7,8].

Discrete Separation Theorem.

Let (b, **B**) and (p, **P**) be intersecting submodular and supermodular functions, respectively. There exists a modular function m: $2^V \to R$ such that $b(B) \leq m(B)$ for $B \in$ **B** and $m(P) \geq p(P)$ for $P \in$ **P** if and only if $\sum p(P_i) \leq \sum b(B_j)$ holds for every disjoint members P_i of **P** and disjoint members B_j of **B** such that $\cup P_i = \cup B_j$. Moreover, if b and p are integer-valued, then m can be chosen to be integer-valued.

For a ring family the condition is simpler.

THEOREM 18. Let **K** be a ring-family and b and p integer-valued sub- and supermodular functions, respectively, on **K** If $p \leq b$, there exists an integer-valued modular function m for which $p \leq m \leq b$.

It is an open problem to find a characterization for the existence of an integer-valued intersecting modular function on **F** such that, given integer-valued intersecting sub- and supermodular functions, respectively, on **F**, $b(F) \geq m(F) \geq P(F)$ for $F \in$ **F**.

9. Applications

In the introduction it was mentioned that the min cost flow, the (poly-) matroid intersection and the Lucchesi-Younger problem are special cases of the Edmonds-Giles model. See also [6,9]. Here we discuss further applications.

I. Orientations.

A directed graph is called *k-strongly arrow connected* or briefly *k-connected* if the number of entering arrows is at least k for any non-empty proper subset. The following theorem is due to Nash-Williams [15].

THEOREM 19. An undirected graph has a k-connected orientation if and only if there exists $2k$ edges between every subset and it complement.

Here we consider a generalization of this problem. Suppose we are given a mixed graph $G = (V, A \cup E)$ (i.e. a graph with arrows and edges). The problem is to find an orientation of the edges so that the resulting digraph should be k-connected. Another problem is to find a minimum cost k-connected orientation if the two possible orientations of each edge have different costs. This problem can be reduced to (1') as follows. First, give an arbitrary orientation to the edges in E. Denote by $\rho(B)$ the number of new and original arrows entering B. Consider the following linear system.

$$\rho_x(B) - \partial_x(B) \le \rho(B) - k \qquad (7)$$
$$0 \le x \le 1$$

It is easy to see that $b'(B) = \rho(B) - k$ is a crossing submodular function on $\mathbf{B}' = 2^V - \{\emptyset, V\}$. Thus (7) is a problem of form (1'). On the other hand there is a one-to-one correspondence between the integer-valued solutions x to (7) and the k-connected orientations of G. Namely, reorient those arrows among the elements of E for which $x(e) = 1$. Therefore an algorithm for the submodular flow problem can be applied to get a minimum cost k-connected orientation. The oracle needed by the algorithm in this case is to minimize $\rho(B) - k - \rho_x(B) + \partial_x(B)$ over $u\bar{v}$-sets. This is equivalent to minimizing $\rho'(B)$ over $u\bar{v}$-sets where ρ' denotes the in-degree function in the reoriented digraph defined by x. This minimization problem is done by a max flow min cut computation.

PROOF OF THEOREM 19. The hypothesis of the Theorem means that the vector consisting of components 1/2 is a solution to (7). But then there exists an integer-valued solution to (7) and such a 0–1 vector corresponds to a k-connected orientation. □

II. Kernel systems

Let $H = (U,A)$ be a digraph, $(p,\mathbf{P})$ an intersecting supermodular function ($\mathbf{P} \subset 2^U$). Morover, at each node v an intersecting submodular function $(b_v, \mathbf{B}_v)$ is given where $\mathbf{B}_v$ consists of some subsets of the arrows entering v. Consider the following linear system.

$$\rho_x(P) \ge p(P) \text{ for } P \in \mathbf{P}$$
$$x(B) \le b_v(B) \text{ for } v \in V, B \in \mathbf{B}_v, \qquad (8)$$
$$0 \le x$$

THEOREM 20. The linear system (8) is TDI.

This theorem was proved in [10] in the special case when no submodular constraints were imposed at the nodes. The proof of this theorem is by showing that (8) can be reduced to (1') by an elementary construction. Namely, let a digraph $G = (V,E,)$ be defined by $V = A' \cup A''$, $E = \{a''a' : \text{for } a \in A\}$, where A' and A'' are disjoint copies of A. Let us define $(b',\mathbf{B}')$ as follows.

$B' \in \mathbf{B}'$ and $b'(B') = b_v(B)$ provided that $B \in \mathbf{B}_v$ for some $v \in U$.

$V - X \in \mathbf{B}'$ and $b'(V - X) = -p(Z)$ provided that there is a $Z \in \mathbf{P}$ such that $X = X_1 \cup X_2$ and X_1 consists of all arrows corresponding to arrows in H induced by Z.

It is not hard to see that $(b',\mathbf{B}')$ is a crossing submodular function and the submodular polyhedron defined by $(b',\mathbf{B}')$ is exactly the solution set of (8).

Note that if one imposed submodular functions at the nodes on the leaving arrows rather than the entering arrows, the resulting linear system would involve the Hamiltonian path problem so the corresponding TDI-ness theorem would not be true.

Here we list some problems which can be transformed into form (8). See also [10].

A. Extend a digraph by adjoining arrows of minimum weight so as to have a flow of value k from a source to a sink. [5]

B. Extend a digraph by adjoining arrows of minimum weight so as to have a flow of value k from a fixed source to each other node.

C. How many arrows can be covered by k spanning arborescences rooted at a fixed node ?

D. When can a digraph be covered by k branchings?

E. Given a digraph and a matroid on its arrow set, find k arrow-disjoint arborescences rooted at a fixed node so that the k arrows entering each node should be independent in the matroid.

III. Generalized polymatroids, semimodular flows

Let $(b, \mathbf{B})$ and $(p, \mathbf{P})$ be intersecting sub- and supermodular functions, respectively, which are compliant. In [11] we called the polyhedron $Q = \{x: x(B) \le b(B) \text{ for } B \in \mathbf{B} \text{ and } x(P) \ge p(P) \text{ for } P \in \mathbf{P}\}$ a *generalized polymatroid* or g-polymatroid and showed that the polymatroid intersection theorem holds for g-polymatroids as well. Namely, if Q_i is a g-polymatroid defined by $(b_i, \mathbf{B}_i, p_i, \mathbf{P}_i)$, $i = 1,2$, the linear system

$$x(B_i) \ge b(B_i) \text{ for } B_i \in \mathbf{B}_i$$
$$i = 1,2 \qquad (9)$$
$$x(P_i) \le p(P_i) \text{ for } P_i \in \mathbf{P}_i$$

is totally dual integral.

In [11] it was also shown that the submodular flow polyhedron arises by projecting the intersection of two g-polymatroids. On the other hand the intersection problem (9) can be formulated as a submodular flow problem therefore the algorithm described in the previous sections applies. To this end take two copies S' and S'' of the groundset S and lead an arrow from each s'' to s'. Let $V = S'' \cup S'$

and let $(b',\mathbf{B}')$ be defined as follows.

$$X \in \mathbf{B}_1 \qquad X' \in \mathbf{B}', \ b'(X') = b_1(X)$$

$$X \in \mathbf{P}_1 \qquad V - X' \in \mathbf{B}', \ b'(V - X') = -p_1(X)$$

For let

$$X \in \mathbf{B}_2 \qquad V - X'' \in \mathbf{B}', \ b'(V - X'') = -b_2(X)$$

$$X \in \mathbf{P}_2 \qquad X'' \in \mathbf{B}', \ b'(X'') = -p(X).$$

It is easy to see that $(b',\mathbf{B}')$ is a crossing submodular function and the submodular flow polyhedron defined by this system is exactly the solution set of (9).

Next we show a symmetric version of the Edmonds-Giles problem in which both lower and upper bounds are imposed on $\lambda_x(B) = \rho_x(B) - \partial_x(B)$. Suppose that $G = (V,E)$ is a digraph and $(b,\mathbf{B})$, $(p,\mathbf{P})$ are intersecting sub- and supermodular functions, respectively, which are compliant.

THEOREM 21. The linear system

$$\lambda_x(B) \le b(B) \text{ for } B \in \mathbf{B}$$

$$\lambda_x(P) \ge p(P) \text{ for } P \in \mathbf{P} \qquad (10)$$

$$f \le x \le g$$

is totally dual integral.

A solution to (10) is called a *semimodular flow*. To reduce (10) to a submodular flow problem, adjoin a new node r to the graph. Let $X \in \mathbf{B}'$ and $b'(X) = b(X)$ if $X \in \mathbf{B}$ and $= -p(X)$ if $V + r - X \in \mathbf{P}$. Now $(b',\mathbf{B}')$ is a crossing submodular function and the submodular flow polyhedron defined by it is the solution set of (10).

References

[1] W. Cunningham-A. Frank, A primal-dual algorithm for submodular flows, *Mathematics of Operations Research*, to appear.

[2] J. Edmonds, Matroid intersection, *Discrete Optimization, Annals of Discrete Math.*, Vol. 4, North-Holland, Amsterdam 1979.

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

[4] J. Edmonds-R. Giles, A min-max relation for submodular functions on graphs, *Studies in Integer Programming (Proc. Workshop on Programming*, Bonn, 1975, P.L. Hammer, E.L. Johnson, B.H. Korte, eds.) *Annals of Discrete Mathematics* 1 (1977), 185-204.

[5] L.R. Ford-D.R. Fulkerson, *Flows in Networks*, Princeton Univ. Press, Princeton, New Jersey, 1962.

[6] A. Frank, A Weighted Matroid Intersection Algorithm, *J. Algorithms* 2, 328-336 (1981).

[7] A. Frank, An algorithm for submodular functions on graphs, *Annals of Discrete Mathematics* 16 (1982) 97-120.

[8] A. Frank, Finding Feasible Vectors of Edmonds-Giles Polyhedra, *J. Combinatorial Theory Ser. B*, (submitted).

[9] A. Frank, How to make a digraph strongly connected, *Combinatorica*, 1 (2) 1981.

[10] A. Frank, Kernel systems of directed graphs, *Acta Sci. Math.* 41 (1-2) 1979.

[11] A. Frank, Generalized polymatroids, *Proceedings of the Sixth Hungarian Combinatorial Colloquium*, held in Eger, 1981 to appear.

[12] M. Grötschel, L. Lovász, A. Schrijver, The ellipsoid method and its consequences in combinatorial optimization, *Combinatorica* 1 (2) 1981.

[13] A. Hoffman, A generalization of max flow min cut, *Mathematical Programming* 6 (1974) 352-359.

[14] E.L. Lawler-C.U. Martel, Computing maximal polymatroidal network flows, Memorandum No. VCB/ERL M80/52, December 22, 1980, Electronics Research Laboratory, College of Engineering, Univ. of California, Berkeley, 94720.

[15] C. St. J. A. Nash-Williams, Well Balanced Orientations of Finite Graphs and Unobtrusive odd-vertex-pairings, in: *Recent Progress in Combinatorics*, Academic Press, New York, 1969.

[16] P. Schönsleben, Ganzzahlige Polymatroid-Intersektions-Algorithmen, Ph. D. dissertation, Eidgenössische Techn. Hochschule Zürich 1980.

[17] A. Schrijver, Total dual integrality from directed graphs, crossing families and sub- and supermodular functions, in this Proceedings.