Annals of Discrete Mathematics 16 (1982) 97–120

AN ALGORITHM FOR SUBMODULAR FUNCTIONS ON GRAPHS

András FRANK*

Research Institute for Telecommunication, Budapest, Hungary 1026

A constructive method is described for proving the Edmonds–Giles theorem which yields a good algorithm provided that a fast subroutine is available for minimizing a submodular set function.

The algorithm can be used for finding a maximum weight common independent set of two matroids, for finding a minimum weight covering of directed cuts of a digraph, and, as a new application, for finding a minimum cost k strongly connected orientation of an undirected graph.

As a theoretical consequence of the algorithm, we prove a combinatorial feasibility theorem for Edmonds–Giles polyhedron and then we derive a discrete separation theorem which says, roughly, an integer valued submodular function B and an integer valued supermodular function R can be separated by an integer valued modular function provided that $R \leqslant B$.

0. Introduction

In [2] Edmonds and Giles have proved a quite general min–max relation for submodular functions on graphs. This result includes such specializations as Hoffman's circulation theorem, Edmonds' polymatroid intersection theorem [1] and the Lucchesi–Younger theorem [15, 16] on directed cuts. Despite this generality, the proof is not too difficult to understand, but it is far from being constructive. One of the purposes of the present paper is to describe an algorithmic proof of the Edmonds–Giles theorem. This proof yields a polynomial bounded algorithm provided that a fast subroutine is available for minimizing a submodular set function. It should be noted that such subroutines indeed exist for the specializations mentioned above.

Recently, Grötschel, Lovász and Schrijver [11] developed a procedure for minimizing an arbitrary submodular function. Their algorithm, which uses the ideas of the ellipsoid method, is a good one. It also implies a rather surprising result, namely, the number of sets X whose value $b(X)$ is explicitly needed

* This paper was written, partly, while the author was visiting the University of Waterloo, Waterloo, Ontario, Canada N2L 3G1.

during the algorithm can be bounded above by a polynomial function of n, the cardinality of the ground set. I think it is a great challenge for combinatorial optimization to find a 'proper' combinatorial algorithm for minimizing a submodular function ('proper' means that the algorithm may use integer arithmetic only and no approximation procedure).

Actually, the method of Grötschel et al. is suitable for algorithmically solving the Edmonds–Giles problem itself. Hence in this sense the present algorithm is not the first one. However our method which operates with such classical combinatorial devices as augmenting path, labelling technique etc., also provides a proof for the Edmonds–Giles theorem while the method of Grötschel et al. does not lend itself to such a proof. In fact, their method essentially makes use of the theorem itself.

Since the Edmonds–Giles theorem implies Edmonds' matroid intersection theorem as well as Lucchesi-Younger theorem on the maximum number of edge-disjoint directed cuts, the specializations of our procedure obviously provide algorithms for these cases. These specializations are rather important for their own sake, so it seems to be worthwhile to work out the details and exploit the special advantages for these cases. See [7, 8]. As a further application of the method we shall show how to find the cheapest k-strongly connected orientation of a $2k$-edge-connected undirected graph if the two possible orientations of any edge may have different costs. (The existence of such an orientation was proved by Nash-Williams [17]. See also [5].)

A theoretical consequence of our algorithm is a combinatorial feasibility theorem from which a discrete separation theorem will be derived. This states, roughly, that the integer valued super- and submodular functions r and b can be separated by an integer valued modular function provided that $r \leqslant b$. This theorem can be considered as a counterpart of the well-known 'continuous' result that a concave and a convex function on a convex, compact set in R^n can be separated by a linear function if the concave function nowhere exceeds the convex one.

Another corollary gives a common generalization of the augmenting circuit theorem from network flow theory and its counterpart in matroid intersection theory [12, 13].

1. Preliminaries

Throughout the paper we work with a finite ground set V of n elements. If $A \subseteq V$, the complement of A is denoted by $\bar{A}$. Sets $A, B \subseteq V$ are *co-disjoint* if $\bar{A}$ and $\bar{B}$ are disjoint. Sets $A, B \subseteq V$ are *intersecting* if none of $A \cap B$, $A - B$, $B - A$ is empty. If, in addition, $A \cup B \neq V$, then A and B are *crossing*. A family $\mathscr{F}$ of subsets of V is *intersecting* (*crossing*) if $A \cap B$, $A \cup B \in \mathscr{F}$ for all intersecting (crossing) members A, B of $\mathscr{F}$. A set function b is *submodular* on A, B if $b(A) + b(B) \geqslant b(A \cap B) + b(A \cup B)$. If equality holds the function is *modular on* A, B. A function r is *supermodular* if $-r$ is submodular. 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 n vertices and m arrows. (We use the term 'arrow' rather than directed edge.) Multiple arrows are allowed but loops not. An arrow uv *enters* (*leaves*) $B \subset V$ if B is a $v\bar{u}$-set ($u\bar{v}$-set). For $H \subseteq E$, $\rho_H(B)$ stands for the number of arrows in H entering B.

Set $\rho(B) = \rho_E(B)$ and define $\delta_H(B) = \rho_H(\bar{B})$. For a single element set we use $\rho(v)$ instead of $\rho(\{v\})$.

Often we shall not distinguish between a subset H of E and its incidence vector x. For example, $\rho_x(B) = \rho_H(B)$.

Let $\mathscr{F}'$ be a crossing family of subsets of V and A' be a $(0, \pm 1)$ matrix the rows of which correspond to the members of $\mathscr{F}'$, the columns correspond to the elements of E and

$$a'_{F,e} = \begin{cases} -1 & \text{if } e \text{ leaves } F, \\ +1 & \text{if } e \text{ enters } F, \\ 0 & \text{otherwise}. \end{cases}$$

Let b' be an integer-valued function on $\mathscr{F}'$ submodular on crossing members of $\mathscr{F}'$. Without loss of generality we can assume that $V, \emptyset \notin \mathscr{F}'$. Let d be a nonnegative vector in R^E, that is, d is a weighting of the arrows. The theorem of Edmonds and Giles can be formulated as follows.

Theorem 1. *The linear programming problem*

$$\begin{aligned} &\max dx \\ &\text{s.t.} \quad 0 \leqslant x \leqslant 1, \ A'x \leqslant b', \end{aligned} \tag{1}$$

has an integral optimal solution provided that it has a feasible solution at all. If, in addition, d is integer-valued there exists an integral optimal solution to the dual linear programming problem.

Remark. Actually, Edmonds and Giles proved their theorem in a more general form. They allowed d to have negative components and the bounds for x were arbitrary, not necessarily 0 and 1. It should be noted however, that the three special cases mentioned earlier (Edmonds' matroid polyhedron theorem, the Lucchesi–Younger theorem and graph orientation) are consequences of this apparently weaker version. Moreover, if $d \not\geqslant 0$, the algorithm can simply be modified without losing the polynomial bound. If in (1) the more general constraint $f \leqslant x \leqslant g$ is prescribed (f, g integer-valued) the algorithm can also be extended to handle this case. That transformation assures finite termination and so provides a proof for the general form of the theorem. However in this case the polynomial bound may be destroyed as is the case in network flow theory where the out-of-kilter algorithm is not a good algorithm. In order to get a proper good algorithm for this general case some scaling technique seems to be needed [3]. Here we do not go into details in this direction but only mention how the general problem (in principle) can be converted into form (1).

(a) If $d(e)<0$, replace $e = uv$ by $e' = vu$. Set $f'(e') = -g(e)$ and $g'(e') = -f(e)$ and $d'(e') = -d(e)$. If $d(e) \geqslant 0$, set $f'(e) = f(e)$, $g'(e) = g(e)$, $d'(e) = d(e)$.

(b) Set $f''(e) = 0$ and $c(e) = g'(e) - f'(e)$ for each arrow e and set $b'(X) = b(X) - \Sigma\,(g'(e)$: e enters $X) + \Sigma\,(f'(e)$: e leaves $X)$ for $X \in \mathscr{F}'$.

(c) Replace any arrow e by $c(e)$ parallel arrows. The new problem is now equivalent to the original one and is of form (1).

First we shall prove Theorem 1 and give an algorithm for the even more special case when the family of subsets in question is intersecting and the function on the family is submodular on each intersecting pair. In Section 8 we show how this proof and algorithm extends to general crossing families.

2. Intersecting families

In order to avoid confusion, instead of $\mathscr{F}'$, let $\mathscr{F}$ denote an intersecting family of subsets of V and assume that $\emptyset \notin \mathscr{F}$, $V \in \mathscr{F}$. Let b be an integer-valued function on $\mathscr{F}$, submodular on intersecting sets and $b(V) = 0$. Let A be defined in the same way as matrix A'. Consider the dual pair of linear programs:

$$\begin{aligned} &\max dx \\ &\text{s.t.}\quad 0 \leqslant x \leqslant 1,\ Ax \leqslant b, \end{aligned} \tag{2}$$

and

$$\begin{aligned} &\min by + 1z \\ &\text{s.t.}\quad y, z \geqslant 0,\ (y, z)\begin{bmatrix} A \\ I \end{bmatrix} \geqslant d, \end{aligned} \tag{3}$$

where the components of y and z correspond to the elements of $\mathscr{F}$ and E, respectively, and I denotes the identity matrix (of appropriate size).

The complementary slackness conditions are

$$\begin{aligned} &x(e) > 0 \to ya_e + z(e) = d(e), \\ &z(e) > 0 \to x(e) = 1, \\ &y(F) > 0 \to a_F^* x = b(F), \end{aligned}$$

where a_e denotes the column vector of A belonging to e and a_F^* stands for the row vector of A belonging to F. One can see that y determines uniquely the optimal vector z, namely $z(e) = d(e) - ya_e$, if this value is positive and 0 otherwise. Thus henceforth we shall refer to the dual solution only by the vector y.

Since we are interested in 0–1 vectors x the optimality criteria are as follows:

$$\begin{aligned} &x(e) = 1 \to ya_e \leqslant d(e), \\ &x(e) = 0 \to ya_e \geqslant d(e), \\ &y(F) > 0 \to a_F^* x = b(F). \end{aligned} \tag{4}$$

Starting with any integral feasible solution to (2), the algorithm will construct vectors x and y satisfying (4). The method is based on two ideas. The first one makes it possible for the dual program to be handled with the help of a so-called potential which is a $|V|$-dimensional vector (unlike the usually much higher dimensional vector y). At the end of the process the optimal y will be simply reconstructed from the potential. The second idea is a generalization of the classical augmenting path method. Roughly speaking, we shall use an augmenting system of disjoint paths rather than one augmenting path for performing a single augmentation. This set of paths will be determined by introducing appropriate auxiliary arrows: a path in the extended graph will define disjoint paths in the original one.

Let us fix a feasible 0–1 solution x to (2) and denote $a_F^* x$ by $\sigma_x(F)$ or briefly $\sigma(F)$. This is a function on $\mathscr{F}$ depending on x and a simple counting argument shows that σ is modular, i.e., $\sigma(K) + \sigma(L) = \sigma(K \cap L) + \sigma(K \cup L)$. The feasibility of x means that $\sigma(F) \leqslant b(F)$ for each $F \in \mathscr{F}$.

A member F of $\mathscr{F}$ is called *b-strict* (with respect to x) or briefly *strict* when it is not ambiguous if $\sigma(F) = b(F)$. For example, V is always strict since $\sigma(V) = b(V) = 0$.

Lemma 2. *If K and L are intersecting strict members of $\mathscr{F}$, then $K \cap L$ and $K \cup L$ are also strict.*

Proof. $b(K) + b(L) = \sigma(K) + \sigma(L) = \sigma(K \cap L) + \sigma(K \cup L) \leqslant b(K \cap L) + b(K \cup L) \leqslant b(K) + b(L)$ from which $\sigma(K \cap L) = b(K \cap L)$ and $\sigma(K \cup L) = b(K \cup L)$ follow, as required. $\square$

Denote by $P(v)$ the intersection of all strict sets containing a vertex v of G. ($P(v)$ depends on x.)

Lemma 3. (a) *$P(v)$ is strict.*

(b) *If a family of strict sets forms a connected hypergraph, the union is again strict.*

Proof. Both statements are direct consequences of Lemma 2. □

3. Potentials

Assume, besides x, we have a vector p in R^V called a *potential* such that

$$x(uv)=1\to \bar{d}(uv)\geq 0\,, \tag{5a}$$

$$x(uv)=0\to \bar{d}(uv)\leq 0\,, \tag{5b}$$

$$u\in P(v)\to p(u)\geq p(v)\,, \tag{5c}$$

where $\bar{d}(uv)=d(uv)-p(v)+p(u)$.

Since adding a constant to each component of p does not affect (5) it can be assumed that the minimum component of p is 0. Let the different potential values be $0=p_0<p_1<\cdots<p_k$. If $k>0$ let $V_i=\{u: p(u)\geq p_i\}$, $i=1,2,\ldots,k$.

Lemma 4. (5c) *is equivalent to the fact that each V_i partitions into strict sets.*

Proof. For $v\in V_i$, (5c) implies $P(v)\subseteq V_i$, thus the components of hypergraph $\{P(v): v\in V_i\}$ partition V_i. Denote by $\mathscr{K}(V_i)$ the set of these components. Lemma 3(b) states that the members of $\mathscr{K}(V_i)$ are strict.

The reverse direction is obvious. □

The notation $\mathscr{K}(V_i)$ introduced in the proof will also be needed later.

For $F\in\mathscr{F}$, define $y(F)=\Sigma\,(p_i-p_{i-1})$, where the summation is taken over those indices for which $F\in\mathscr{K}(V_i)$. (Here the empty sum is defined to be zero.)

Claim. *For any arrow $e=uv$, $p(v)-p(u)=ya_e$.*

Proof. Let t_i (s_i) denote the number of sets in $\mathscr{K}(V_i)$ which are entered (left) by e. Obviously both t_i and s_i are 0 or 1. Now we have

$$\begin{aligned} ya_e &= \sum_F (y(F)\colon e \text{ enters } F)-\sum_F (y(F)\colon e \text{ leaves } F)\\ &=\sum_F\Big(\sum_i (p_i-p_{i-1}\colon F\in\mathscr{K}(V_i))\colon e \text{ enters } F\Big)\\ &\quad-\sum_F\Big(\sum_i (p_i-p_{i-1}\colon F\in\mathscr{K}(V_i))\colon e \text{ leaves } F\Big)\\ &=\sum_{i=1}^k t_i(p_i-p_{i-1})-\sum_{i=1}^k s_i(p_i-p_{i-1})\\ &=\sum_i (p_i-p_{i-1}\colon e \text{ enters } V_i)-\sum_i (p_i-p_{i-1}\colon e \text{ leaves } V_i)\\ &=\begin{cases} p(v)-p(u) & \text{if } p(v)>p(u)\,,\\ 0 & \text{if } p(v)=p(u)\,,\\ -(p(u)-p(v)) & \text{if } p(v)<p(u)\,.\end{cases}\end{aligned}$$

Here we made use of the fact that $t_i=s_i$ whenever $u,v\in V_i$ or $u,v\notin V_i$. Furthermore, if $p(v)>p(u)$ then the second sum is empty, while $p(u)>p(v)$ implies that the first sum is empty. □

By this claim and the definition of y we need a 0–1 feasible vector x and a potential p satisfying (5).

We shall refer to an arrow uv as a *1-arrow* (with respect to the given vector x) if $x(uv)=1$ while uv is a *0-arrow* if $x(uv)=0$.

The algorithm will maintain (5a) and (5c) and the number of arrows violating (5b) will gradually reduce.

4. Inner algorithm and proof

The core of our procedure is the following.

Inner algorithm

Input. x : 0–1 feasible solution to (2),
p : potential,
$e=ab$: 0-arrow,

so that (5a) and (5c) hold but e violates (5b).

Output. x': 0–1 feasible solution to (2),
p': potential,

so that (5a) and (5c) continue to hold, e does not violate (5b) and any arrow can violate (5b) only if it violates (5b) with respect to x and p.

Assume this algorithm is available. At the beginning let $p \equiv 0$ and x be an arbitrary 0–1 feasible solution to (2). Repeat the Inner Algorithm until there are no more arrows violating (5b). After no more than $|E|$ applications of this algorithm its output will satisfy all the three criteria in (5).

To describe the Inner Algorithm, define an auxiliary digraph $H=(V, A)$ (depending on the current x and p) as follows. Set $A = A_B \cup A_W \cup A_R$ where

$$A_B = \{uv\colon uv \text{ is a 1-arrow and } \bar{d}(uv) \leqslant 0\},$$
$$A_W = \{vu\colon uv \text{ is a 0-arrow and } \bar{d}(uv) \geqslant 0\},$$
$$A_R = \{uv\colon u \in P(v) \text{ and } p(u) = p(v)\}.$$

(Note that A may contain parallel arrows.) Refer to the elements of A_B, A_W and A_R as blue, white and red arrows, respectively.

Try to find a directed path in H from a to b. There may be two cases.

Case 1. $b \notin T = \{v\colon v \text{ can be reached from } a \text{ in } H\}$. Obviously,

(P) there is no arrow in H leaving T.

Revise the potential as follows:

$$p'(v) = \begin{cases} p(v) & \text{if } v \in T, \\ p(v) + \delta & \text{if } v \notin T, \end{cases}$$

where $\delta = \min(\delta_e, \delta_B, \delta_W, \delta_R)$, where

$$\delta_e = \bar{d}(ab),$$
$$\delta_B = \min\{\bar{d}(uv)\colon uv \text{ is a 1-arrow of } G \text{ leaving } T\},$$
$$\delta_W = \min\{-\bar{d}(uv)\colon uv \text{ is a 0-arrow of } G \text{ entering } T\}$$

and

$$\delta_R = \min\{p(u) - p(v)\colon v \notin T,\ u \in P(v) \cap T\}.$$

(Here the minimum is defined to be plus infinity when it is taken over the empty set.)

Claim. $\delta > 0$.

Proof. Since e violated (5b), $\delta_e > 0$. If $\bar{d}(uv) \leqslant 0$ for a 1-arrow uv leaving T, then uv would be a (blue) arrow in H leaving T, contradicting (P), therefore $\delta_B > 0$. If $\bar{d}(uv) \geqslant 0$ for a 0-arrow uv entering T, then vu would be a (white) arrow in H leaving T, contradicting (P), so $\delta_W > 0$. Finally, from (5c), $p(u) \geqslant p(v)$ whenever $u \in P(v)$, that is $\delta_R \geqslant 0$. If there were u and v with $v \notin T$, $u \in P(v) \cap T$ and $p(u) = p(v)$, then uv would be a (red) arrow in H leaving T, contradicting (P). □

The revised function $\bar{d}'(uv)$ is

$$\bar{d}'(uv) = \begin{cases} \bar{d}(uv) - \delta & \text{if } uv \text{ leaves } T, \\ \bar{d}(uv) + \delta & \text{if } uv \text{ enters } T, \\ \bar{d}(uv) & \text{otherwise}. \end{cases} \tag{6}$$

Claim. (5a) *continues to hold.*

Proof. For a 1-arrow uv, $\bar{d}(uv) \geqslant 0$. If, indirectly $\bar{d}'(uv) < 0$, then uv leaves T by (6). Now $\bar{d}(uv) \geqslant \delta_B \geqslant \delta$, that is, $\bar{d}'(uv) \geqslant 0$, a contradiction. □

Claim. *If* (5b) *was true for a* 0-*arrow* uv, *it continues to hold.*

Proof. Since $\bar{d}(uv) \leqslant 0$, the indirect assumption $\bar{d}'(uv) > 0$ and (6) would imply that uv enters T. Now $-\bar{d}(uv) \geqslant \delta_W \geqslant \delta$, that is, $\bar{d}'(uv) = \bar{d}(uv) + \delta \leqslant 0$, a contradiction. □

Claim. (5c) *continues to hold.*

Proof. Note that $P(v)$ does not depend on the potential change. Let $u \in P(v)$ and suppose indirectly that $p'(u) < p'(v)$. Then $v \notin T$, $u \in P(v) \cap T$, thus $p'(u) = p(u)$ and $p'(v) = p(v) + \delta$. Hence $p(u) - p(v) < \delta$. On the other hand $p(u) - p(v) \geqslant \delta_R \geqslant \delta$, a contradiction. □

If $\delta = \delta_e$, the arrow $e = ab$ satisfies (5b), and thus the solutions $x' = x$ and p' satisfy the requirements of the Inner Algorithm.

If $\delta < \delta_e$ then repeat the Inner Algorithm using, as inputs, the same x, the revised potential $p := p'$ and the same arrow $e = ab$ which still violates (5b). Observe that the arrow set induced by T in the new auxiliary digraph H' is the same as it was in H. Moreover, the definition of δ ensures that H' contains at least one arrow leaving T (which is blue, white or red according as δ is equal to δ_B, δ_W or δ_R). This implies that the set $T' = \{v\colon v \text{ can be reached from } a \text{ in } H'\}$ properly includes T. Consequently, after at most $|V| - 1$ iterations, either the equality $\delta = \delta_e$ will hold or vertex b will be reached from a. This is Case 2.

Case 2. There is a directed path from a to b in H. Let U be a shortest path. (Actually, we shall use only the fact that there is no red 'shortcut' arrow to U, that is, if the vertices of U in order are $a = v_0, v_1, \ldots, v_k = b$, then $v_i v_{i+j}$ $(j \geqslant 2)$ must not be a red arrow.)

Since ba is a white arrow in H, U and ba form a directed circuit in H. This may include blue, white and red arrows. Let E_1 be the set of arrows in G which correspond to the blue or white arrows of that circuit. Define a new vector x' as follows:

$$x'(e) = \begin{cases} 1 - x(e) & \text{if } e \in E_1, \\ x(e) & \text{otherwise}. \end{cases}$$

(That is, a 1-arrow in E_1 becomes a 0-arrow, while a 0-arrow will be a 1-arrow.) We shall prove that x' and $p' := p$ satisfy the requirements of the Inner Algorithm. For a member F of $\mathcal{F}$ let $\rho_r(F)$ ($\delta_r(F)$) stand for the number of red arrows of U entering (leaving) F.

Lemma 5. *x' is a feasible solution to* (2).

Proof. The proof consists of proving a number of claims.

Claim. $a^*_F x' = a^*_F x + \rho_r(F) - \delta_r(F)$.

Proof. This is quite easy when $\rho_r(F) = \delta_r(F) = 0$ and, in general, follows by a simple induction on $\rho_r(F) + \delta_r(F)$. □

We have to prove that $a^*_F x' \leq b(F)$. By the claim it suffices to prove that $\rho_r(F) \leq \varepsilon(F)$, where $\varepsilon(F) = b(F) - \sigma_x(F)$ (recall that $\sigma_x(F) = a^*_F x$). Now $\varepsilon(F)$ is submodular on intersecting members of $\mathcal{F}$.

Let uv be a red arrow of U entering F such that $p(u)$ ($= p(v)$) is as large as possible, and if there are more such arrows let uv be the last one on the path U (starting from a).

Claim. $\rho_r(F \cup P(v)) = \rho_r(F) - 1$.

Proof. Since no red arrow enters $P(v)$ and uv does not enter $F \cup P(v)$, $\rho_r(F \cup P(v)) \leq \rho_r(F) - 1$. On the other hand if st is another red arrow of U which enters F, then we claim that $s \notin P(v)$ (that is, st enters $F \cup P(v)$ as well): in the contrary case $p(s) \geq p(v)$ by (5c) thus the maximal choice of $p(v)$ implies $p(s) = p(t) = p(v)$. However, this implies that sv is a red arrow. Because of the choice of uv, st precedes uv on the path U (starting from a) thus sv is a red shortcut arrow to U, a contradiction. □

Claim. $\rho_r(F) \leq \varepsilon(F)$ *for any* $F \in \mathcal{F}$.

Proof. By induction on $\rho_r(F)$. Observe that "$\varepsilon(F) \geq 0$ for each $F \in \mathcal{F}$" is equivalent to "x is a feasible solution to (2)" and $\varepsilon(F) = 0$ if and only if F is strict (with respect to x). Let $\rho_r(F) > 0$ and let uv be defined in the same way as in the previous claim. Then

$$\varepsilon(F) = \varepsilon(F) + \varepsilon(P(v)) \geq \varepsilon(F \cap P(v)) + \varepsilon(F \cup P(v)) \geq 1 + \varepsilon(F \cup P(v))$$
$$\geq 1 + \rho_r(F \cup P(v)) = \rho_r(F).$$

Here we used the submodularity of ε, the induction hypothesis for $F \cup P(v)$ and the previous claim. □

This completes the proof of the lemma. □

After proving the lemma, let us investigate what happened to the optimality criteria. Since ab has become a 1-arrow it does not violate (5b). If uv is a new 1-arrow, then vu was a white arrow in H so $\bar{d}(uv) \geq 0$. If uv is a new 0-arrow then uv was a blue arrow in H thus $\bar{d}(uv) \leq 0$. That is, (5a) continues to hold and new 0-arrow violating (5b) has not arisen.

Claim. (5c) *holds with respect to x' and p'.*

Proof. From Lemma 4 we know that V_i is the union of disjoint strict sets $X_1, X_2, \ldots, X_r$ where each X_i is strict with respect to x. Since no red arrow leaves any strict set and no red arrow enters V_i (for a red arrow uv we had $p(u) = p(v)$) we have $\rho_r(X_i) = \delta_r(X_i) = 0$ whence $a^*_{X_i} x' = a^*_{X_i} x$, that is, each X_i is strict with respect to x'. Apply again Lemma 4. □

The current primal solution x is 0–1 valued throughout the algorithm regardless the integrality of the objective function d. If, in addition, d was integral, then the current potential p is also integral throughout the process, and hence so is the dual solution (y, z). These observations complete the proof of Theorem 1, when the set system $\mathcal{F}$ in question is intersecting. □

5. Steps of the algorithm

Before describing the algorithms in detail some remarks are needed about the steps and the number of steps of the algorithm. In order to apply the algorithm we have to be able to determine the set $P(v)$ for each vertex v, in any intermediate stage. To this end suppose we have an oracle which can

(O) decide, for any primal solution x and vertices u, v whether or not there exists a strict $v\bar{u}$-member of $\mathcal{F}$.

A simple argument shows that $P(v)$ consists of those vertices u for which the answer is no. This means that, in constructing the auxiliary digraph H belonging to a given stage of the algorithm, A_R can be defined as $A_R = \{uv: p(u) = p(v)$ and there is no strict $v\bar{u}$-set$\}$.

If oracle (O) is available and its run needs at most g steps, then $P(v)$ can be determined in at most gn steps for a fixed vertex v. For all v this means n^2g steps.

Another part of the algorithm tries to find a directed path from a to b in the auxiliary digraph H. This can be done with a well-known labelling technique. If this is accomplished by a breadth-first search then a shortcut free path will automatically be produced, if it exists. If no path exists from a to b in H, the set of labelled vertices will just be T. The labelling procedure needs at most n^2 steps. Moreover, if $\delta < \delta_e$ occurs during the algorithm and the Inner Algorithm is started again with the same x and $p := p'$, then the labels determined previously may be used again (recall that $T \subset T'$). In this case the new auxiliary digraph arises simply from the old one in such a way that some arrows from T to $\bar{T}$ are added while some arrows from $\bar{T}$ to T are deleted. Therefore the overall complexity of the Inner Algorithm can be bounded by $O(n^2 + n^2g)$.

The Inner Algorithm will be applied at most $|E|$ times. From the optimal primal solution x and potential p the optimal dual solution can be obtained in at most $O(n^3)$ steps since the components of the hypergraph $\{P(v): v \in V_i\}$ can be obtained in $O(n^2)$ steps and we have at most n different V_i's. Consequently, the optimal primal–dual solutions to linear programms (2) and (3) can be obtained in at most $O(mn^2g + n^3)$ steps provided that a starting 0–1 feasible solution to (2) and oracle (O) is available.

In order for (O) to be available we need a subroutine for minimizing a submodular function, namely minimize $\varepsilon(F)$ $(= b(F) - \sigma_x(F))$ over the $v\bar{u}$-members F of $\mathscr{F}$. If the minimum is negative, the current vector x is not feasible, if the minimum is zero, then there exists a strict $v\bar{u}$-set, if the minimum is positive, then $u \in P(v)$.

Algorithm for intersecting $\mathscr{F}$

Input. G: directed graph,
$\mathscr{F}$: intersecting family, $\mathscr{F} \subset 2^V$,
$b: \mathscr{F} \to Z$ integer-valued function, submodular on intersecting pairs,
$d: E \to R^+$ nonnegative objective function,
x: 0–1 feasible solution to (2).

Output. x: optimal 0–1 solution to (2),
y, z: optimal solution to (3), which is integral if d is.

Step 1.
1.0. Determine $P(v)$, for each $v \in V$.
1.1. If every 0-arrow satisfies (5b), the current x is optimal. Go to Step 4.
1.2. Select an arrow $e = ab$ violating (5b).
1.3. Form the auxiliary digraph $H = (V, A)$ and try to find a directed path from a to b by the labelling technique (making use of labels determined but not deleted previously). If a path U exists go to Step 3.

Step 2 (Change in potential).
2.0. Let T be the set of the labelled vertices. Count δ and set $p(v) := p(v) + \delta$ whenever $v \not\in T$.
2.1. If $\delta = \delta_e$ delete all the labels and go to 1.1.
2.2. Go to 1.3.

Step 3 (Change in x). Denoting by E_1 the set of arrows of G corresponding to the blue and white arrows of the circuit $U + ba$, set

$$x(e) := \begin{cases} 1 - x(e) & \text{if } e \in E_1, \\ x(e) & \text{otherwise}. \end{cases}$$

Go to 1.0.

Step 4 (Forming the optimal solution (y, z) to (3)).
4.0. Let the different values of p be $0 = p_0 < p_1 < \cdots < p_k$. Set $V_i = \{u: p(u) \geq p_i\}$ for $i = 1, 2, \ldots, k$.
4.1. For each i, determine the components of the hypergraph $\{P(u): u \in V_i\}$. Denote by $\mathscr{K}(V_i)$ the set of these components.
4.2. For $F \in \mathscr{F}$, set $y(F) = \Sigma\,(p_i - p_{i-1})$, where the summation is taken over those indices i for which $F \in \mathscr{K}(V_i)$. (The empty sum is zero.)
4.3. For a 1-arrow e, set $z(e) = d(e) - ya_e$, for a 0-arrow e set $z(e) = 0$. Halt.

6. Starting feasible solution

In this section we investigate the problem of finding a 0–1 feasible solution to (2). It is assumed again that $\mathscr{F}$ is an intersecting family and b is submodular on intersecting members of $\mathscr{F}$.

Feasibility Theorem. *There exists a* 0–1 *feasible solution to* (2) *if and only if*

$$\sum_i b(X_i) \geq -\delta(\bigcup X_i), \tag{7}$$

for disjoint members $X_1, X_2, \ldots, X_k$ *of* $\mathscr{F}$.

Proof. *Necessity.* For a feasible solution x we have $\Sigma\, b(X_i) \geq \Sigma\, \sigma(X_i) = \sigma(\bigcup X_i) \geq -\delta(\bigcup X_i)$.

Sufficiency. A simple trick due to Hoffman [18] will enable us to reduce the problem to that investigated before. Extend the graph $G = (V, E)$ by adding a new vertex r and $|E|$ new arrows as follows. For each vertex $v \in V$ join $\delta(v)$ parallel arrows from r to v. For each $F \in \mathcal{F}$ let $b'(F) = b(F) + \Sigma_{v\in F}\, \delta(v)$. Obviously b' is submodular on intersecting members of $\mathcal{F}$. Furthermore, since $\delta(F) \leq \Sigma_{v\in F}\, \delta(v)$ and $b(F) \geq -\delta(F)$ by (7), it follows that b' is nonnegative.

Let us consider the linear program (2) with respect to the extended graph G' and the new function b' whereas $\mathcal{F}$ remains the same. A simple argument shows that

(A) the original program has a feasible solution if and only if the new program has a feasible solution $x = (x_1, x_2)$ in which $x_2(e) = 1$ for each new arrow e.

Here the components of x_1 and x_2 correspond to the original and new arrows, respectively.

Let the new objective function be $d(e) = 1$ if e is a new arrow and $d(e) = 0$ if e is old. Since $b' \geq 0$ the identically zero vector is an appropriate starting feasible solution. Apply the algorithm with this starting solution. By (A) what we have to prove is that the value of the optimal solution to the new program is just $|E|$. The algorithm provides a primal solution x and a potential p which satisfy (5). Suppose indirectly that $x_2(ru) = 0$ for a new arrow ru. Observe that only the new arrows violated (5b) at the beginning of the algorithm, therefore $p(r) = 0$ throughout the algorithm. Furthermore, $x_2(ru) = 0$ and (5b) imply that $p(u) > 0$. Therefore the set $X = \{v: p(v) > 0\}$ is non-empty. From Lemma 4 X is a disjoint union of some strict sets X_i. That is $X = \bigcup X_i$ and $b'(X_i) = \sigma_x(X_i)$. Moreover, no original 1-arrow enters X and no original 0-arrow leaves X because of (5a) and (5b), respectively. Thus $\Sigma_i\, \sigma_{x_1}(X_i) = -\delta(X)$. Furthermore, since $x_2(ru) = 0$ we have $\sigma_{x_2}(X) < \Sigma\, (\delta(v): v \in X)$. Consequently, $\Sigma_i\, \sigma_x(X_i) = \Sigma_i\, \sigma_{x_1}(X_i) + \Sigma_i\, \sigma_{x_2}(X_i) < -\delta(X) + \Sigma\, (\delta(v): v \in X)$ from which $\Sigma_i\, b(X_i) + \Sigma\, (\delta(v): v \in X) = \Sigma\, b'(X_i) < \Sigma\, (\delta(v): v \in X) - \delta(X)$, that is, $\Sigma_i\, b(X_i) < -\delta(X)$, contradicting the hypothesis of the theorem. □

7. A discrete separation theorem

In this section we shall make use of the simple observation that, for (2) to have a feasible solution, it suffices to require (7) only for those families of disjoint members $X_1, X_2, \ldots, X_k$ of $\mathcal{F}$ where $X_i \cup X_j \in \mathcal{F}$ implies that $b(X_i) + b(X_j) < b(X_i \cup X_j)$ $(i \neq j)$. Indeed, if (7) were not true in general under this weaker restriction, then $\Sigma\, b(X_i) < -\delta(\bigcup X_i)$ for some family $\{X_1, X_2, \ldots, X_k\}$. Let k be as small as possible such that this inequality holds. Now, for some X_i and X_j, say X_1 and X_2, $X_1 \cup X_2 \in \mathcal{F}$ and $b(X_1) + b(X_2) \geq b(X_1 \cup X_2)$, whence $\{X_1 \cup X_2, X_3, \ldots, X_k\}$ would also violate (7). But this family consists of $k-1$ sets only, contradicting the minimality of k.

Let $\mathcal{K}$ be a family of subsets of S closed under union and intersection. Let R and B be two integer-valued functions on $\mathcal{K}$ which are super- and submodular on any two members of $\mathcal{K}$, respectively.

Discrete Separation Theorem. *If $R(X) \leq B(X)$ whenever $X \in \mathcal{K}$, there exists an integer-valued modular function m such that $R(X) \leq m(X) \leq B(X)$ for each $X \in \mathcal{K}$.*

Proof. We can suppose that $\bigcap(X: X \in \mathcal{K}) \neq \emptyset$. For otherwise join an extra vertex to each member of $\mathcal{K}$. Let $(S', \mathcal{K}')$ and $(S'', \mathcal{K}'')$ be two copies of $(S, \mathcal{K})$ and join k parallel arrows from any $s' \in S'$ to $s'' \in S''$ and from s'' to s', where k is a big number. Here 'big' means that the outdegree function δ satisfies:

$$-\delta(X') \leq B(X)\,, \tag{8a}$$

$$\delta(X'') \geq R(X)\,, \tag{8b}$$

$$\delta(X' \cup Y'') \geq R(Y) - B(X)\,, \tag{8c}$$

for any $X, Y \in \mathcal{K}$. Obviously, increasing k, $\delta(X'')$ increases since $X \neq \emptyset$ and so does $\delta(X' \cup Y'')$ whenever $X \neq Y$. Thus, for sufficiently large k, (8a), (8b) and (8c) (for $X \neq Y$) will hold. If $X = Y$ then $\delta(X' \cup X'') = 0$ for any k, however $0 \geq R(X) - B(X)$ follows from the hypothesis.

Denoting by E the set of arrows and by $V = S' \cup S''$ we have a directed graph $G = (V, E)$ and a family $\mathcal{F} = \mathcal{K}' \cup \mathcal{K}'' \cup \{V\}$ on its vertices. (Note that the fact $\emptyset \notin \mathcal{F}$ in (2) requires the assumption $\bigcap(X: X \in \mathcal{K}) \neq \emptyset$.) Furthermore set $b(X') = B(X)$ for $X' \in \mathcal{K}'$ and $b(X'') = -R(X)$ if $X'' \in \mathcal{K}''$ and $b(V) = 0$. Using the remark done at the beginning of this section, the Feasibility Theorem requires just the truth of (8a), (8b) and (8c). Therefore, by the Feasibility Theorem we have an integer-valued feasible solution x. Let us define $m(K) = \sigma_x(K') = \rho_x(K') - \delta_x(K')$ for $K \in \mathcal{K}$. Then m satisfies the requirements of the theorem. □

Remark. The main content of the Separation Theorem is that the separating modular function is integer-valued. Actually, the existence of a not necessarily integer-valued separating function follows simply from the classical real separation theorem since a submodular (supermodular) function on $\mathcal{K} \subset 2^S$ can be extended to a convex (concave) function on R^S so that the convex function nowhere exceeds the concave one.

In our treatment the Feasibility Theorem—and so the Separation Theorem—was a by-product of a more or less complicated algorithm. Of course there exist simpler proofs of them which do not use arrow-weights. In [6] we proved directly a theorem in terms of orientations of an undirected graph, which is equivalent to the Feasibility Theorem. However in that paper the Separation Theorem was not explicitly mentioned. In [9] we refine the proof of [6] by extending a method of Lawler and Martel [this volume, pp. 189–200] and prove a feasibility theorem for the general case (when $f \leq x \leq g$). Hence we have a good algorithm not depending on f and g and this allows us to obtain the separating modular function in polynomial time.

For an instance of applicability of the Discrete Separation Theorem we show how Edmonds' matroid intersection theorem [1] follows from it. An equivalent version of Edmonds' theorem states that two matroids M_1 and M_2 on S, with the same rank r, have a common base if and only if $b_1(X)+b_2(S-X) \geq r$ for any $X \subset S$, where b_i is the rank function of M_i, $i=1,2$. To see the sufficiency, let $\mathcal{K}$ consist of all subsets of S, set $B(X)=b_1(X)$ and $R(X)=r-b_2(S-X)$. Since $B(X) \geq R(X)$, by the Discrete Separation Theorem, an integer-valued modular function m separates $B(X)$ and $R(X)$. It is an easy exercise to check that m is 0–1 valued on the vertices and the set $X=\{x: m(x)=1\}$ is just a common base.

Another easy consequence of our separation result is a theorem on polymatroids due to Giles [10]. Let b_1 and b_2 be two submodular functions on all subsets of S such that $b_i(\emptyset)=0$ and b_i is monotone increasing, that is $b_i(X) \geq b_i(Y)$ for $X \supseteq Y$, $i=1,2$.

Theorem 7. *If $x \geq 0$ is an integer-valued vector ($x \in Z^V$) such that $x(T) \leq b_1(T)+b_2(T)$ for each $T \subseteq S$, then $x=x_1+x_2$ for some nonnegative integer-valued vectors x_1 and x_2 for which $x_i(T) \leq b_i(T)$ for each $T \subset S$ and $i=1,2$. (Here $x(T)$ stands for $\Sigma\,(x(s)\colon s \in T)$.)*

Proof. Apply the Discrete Separation Theorem to the functions $R(T)=x(T)-b_1'(T)$ and $B(T)=b_2(T)$ where $b_1'(T)=\min_{X \subseteq T}(b_1(X)+x(T-X))$. □

8. Crossing families

In this section we prove the Edmonds–Giles theorem for the more general case of crossing families and show how the algorithm of Section 5 can be extended. The idea behind this extension is that, with a crossing family $\mathcal{F}'$ and function b' on $\mathcal{F}'$ submodular on crossing members of $\mathcal{F}'$, one may associate an intersecting family $\mathcal{F}$ and a function b on $\mathcal{F}$ submodular on intersecting members of $\mathcal{F}$ so that the sets of feasible solutions to (1) and (2) coincide. Then we can apply the algorithm developed for solving (2).

We shall need a theorem due to Lovász [14].

Theorem 8. *Let $\mathcal{F}'' \subset 2^V$ be a crossing family ($\emptyset, V \notin \mathcal{F}''$), b'' be a function on $\mathcal{F}''$ submodular on any two crossing members of $\mathcal{F}''$. Define $\mathcal{F}=\{X: X=\bigcup X_i \neq V, X_i \in \mathcal{F}'', X_i \cap X_j=\emptyset\}$ and $b(X)=\min(\Sigma\, b''(X_i)\colon X_i \in \mathcal{F}', X=\bigcup X_i, X_i \cap X_j=\emptyset)$. If $X, Y \in \mathcal{F}$ and $X \cup Y \neq V$, then $X \cup Y$, $X \cap Y \in \mathcal{F}$ and $b(X)+b(Y) \geq b(X \cup Y)+b(X \cap Y)$. (Note that $\mathcal{F}'' \subseteq \mathcal{F}$ and $b''(X) \geq b(X)$ for $X \in \mathcal{F}''$.)*

Proof. Let $X, Y \in \mathcal{F}$ be such that $X \cup Y \neq V$. Then $X=\bigcup X_i$, $b(X)=\Sigma\, b''(X_i)$ for some disjoint members X_i of $\mathcal{F}''$ and $Y=\bigcup Y_j$, $b(Y)=\Sigma\, b''(Y_j)$ for some disjoint members Y_j of $\mathcal{F}''$.

If we have some members A_i of $\mathcal{F}''$ which form a connected hypergraph and their union is not V, then this union is in $\mathcal{F}''$. Therefore the components formed by the hypergraph $\{X_i\} \cup \{Y_j\}$ are disjoint members of $\mathcal{F}''$, thus $X \cup Y$ is in $\mathcal{F}$. Furthermore $X \cap Y=\bigcup(X_i \cap Y_j\colon X_i \cap Y_j \neq \emptyset)$ whence $X \cap Y \in \mathcal{F}$.

We need the following lemma.

Lemma 9. *Suppose that the members $A_1, A_2, \ldots, A_l$ and $B_1, B_2, \ldots, B_k$, $k, l \geq 1$, of $\mathcal{F}''$ partition A and B, respectively, $\{A_i\} \cup \{B_j\}$ forms a connected hypergraph and $A \cup B \neq V$. Then $A \cup B \in \mathcal{F}''$ and $\Sigma\, b''(A_i)+\Sigma\, b''(B_j) \geq b''(A \cup B)+b(A \cap B)$.*

Proof. The first part of the lemma is simple (for a similar observation see Lemma 3). To see the inequality we proceed by induction on $k+l$. The case $k+l=2$ is obvious so assume $k+l>2$. Deleting an appropriate edge of a hypergraph, say A_k, the resulting hypergraph remains connected. Now the induction hypothesis holds for $A'=A-A_k$ and B; thus

$$\sum_{i=1}^{k-1} b''(A_i)+\sum_{j=A}^{l} b''(B_j) \geq b''(A' \cup B)+b(A' \cap B).$$

Adding $b''(A_k)$ to both sides we get

$$\sum b''(A_i)+\sum b''(B_j) \geq b''(A' \cup B)+b''(A_k)+b(A' \cap B).$$

Since $A' \cup B$ and A_k are crossing members of $\mathcal{F}''$ we have

$$b''(A'\cup B)+b''(A_k)\geq b''(A'\cup B\cup A_k)+b''((A'\cup B)\cap A_k)$$
$$=b''(A\cup B)+b''(B\cap A_k).$$

Furthermore, $A'\cap B$ and $A_k\cap B$ are disjoint. Thus

$$b''(B\cap A_k)+b(A'\cap B)\geq b(A\cap B).$$

From the last three inequalities the lemma follows. □

Let $C_1, C_k, \ldots, C_m$ be the components of the hypergraph $\{X_i\}\cup\{Y_j\}$. Applying Lemma 9 to $A = X\cap C_h$ and $B = Y\cap C_h$, $h = 1, 2, \ldots, m$, we get

$$\begin{aligned} b(X)+b(Y) &= \sum_{h=1}^{m}\Big(\sum_i b''(X_i\cap C_h)+\sum_j b''(Y_j\cap C_h)\Big)\\ &\geq \sum_{h=1}^{m}(b''((X\cup Y)\cap C_h)+b(X\cap Y\cap C_h))\\ &\geq b(X\cup Y)+b(X\cap Y). \end{aligned}$$

This completes the proof of Theorem 8. □

In fact, what we need is the following version of Lovász's theorem.

Lemma 10. *Let $\mathcal{F}'\subset 2^V$ be a crossing family, b' be a function on $\mathcal{F}'$ submodular on any two crossing members of $\mathcal{F}'$. Define*

$$\mathcal{F}=\{X: X\neq\emptyset,\ X=\bigcap X_i,\ X_i\in\mathcal{F}',\ \bar{X}_i\cap\bar{X}_j=\emptyset\}\cup\{V\}.$$

In other words $\mathcal{F}-\{V\}$ consists of non-empty sets arising as the intersection of some pairwise co-disjoint members of $\mathcal{F}'$. Let $b(X)=\min(\Sigma\, b'(X_i): X=\bigcap X_i$, $x_i\in\mathcal{F}'$, $\bar{X}_i\cap\bar{X}_j=0)$ and $b(V)=0$.

Then $\mathcal{F}$ is an intersecting family and b is submodular on any two intersecting members of $\mathcal{F}$.

Proof. Apply Lovász's theorem for $\mathcal{F}''=\{X: \bar{X}\in\mathcal{F}'\}$ and $b''(X)=b'(\bar{X})$. □

Denote by P_1 and P_2 the polyhedra defined by (1) and (2) respectively. Now, for (1), b' is the given function on $\mathcal{F}'$ while b and $\mathcal{F}$ for (2) are defined as in Lemma 10.

Lemma 11. $P_1 = P_2$.

Proof. Since $\mathcal{F}'\subseteq\mathcal{F}$ and $b(F')\leq b'(F')$ for $F'\in\mathcal{F}'$ we have $P_1\supseteq P_2$. On the other hand, for a vector x in P_1 and for $F\in\mathcal{F}$ we have

$$b(F)=\sum b'(X_i)\geq\sum b(X_i)\geq\sum\sigma_x(X_i)=\sigma_x(F) \tag{9}$$

for some $X_i\in\mathcal{F}'$, where $F=\bigcap X_i$ and $\bar{X}_i\cap\bar{X}_j=\emptyset$. (9) shows that $x\in P_2$. □

Lemma 12. *For $F\in\mathcal{F}-\{V\}$ the following statements are equivalent:*
(a) *F is b-strict;*
(b) *F is the intersection of some b'-strict members of $\mathcal{F}'$;*
(c) *F is the intersection of some pairwise co-disjoint b'-strict members of $\mathcal{F}'$.*

Proof. (a)→(c) simply follows from (9). (c)→(b) is trivial. To see (b)→(a), let $F=\bigcap F_i$ where each F_i, $i=1,2,\ldots,t$, is a b'-strict member of $\mathcal{F}'$. If among these sets F_i there are two which cross, then they can be replaced by their intersection which is a b'-strict member of $\mathcal{F}'$. Thus if we assume t to be minimal, then the sets F_i are pairwise non-crossing and since their intersection F is non-empty they are pairwise co-disjoint. Thus $b(F)\leq\Sigma\, b'(F_i)=\Sigma\,\sigma_x(F_i)=\sigma_x(F)$. From this and (9), $b(F)=\sigma_x(F)$, as required. Note that this replacement operation yields a polynomial procedure. □

Taking into consideration Lemmas 10 and 11, in order to solve (1), it suffices to solve (2) with respect to b. The only difficulty arising from this approach is that of how one can work with the new function b when originally only b' is specified and from an algorithmic point of view the definition of b is rather complicated. Fortunately, we do not need the explicit values of b at all. We have seen that, in order to apply the algorithm of Section 5, only oracle (O) has to be available. The following lemma shows that this is indeed the case provided that the same oracle is available concerning the given $\mathcal{F}'$ and b'.

Lemma 13. *Given $x\in P_1$ $(=P_2)$ and $u, v\in V$, there exists a b-strict $v\bar{u}$-set in $\mathcal{F}$ if and only if there exists a b'-strict $v\bar{u}$-set in $\mathcal{F}'$.*

Proof. Let $F'\in\mathcal{F}'$ be a b'-strict $v\bar{u}$-set. Then $b(F')\leq b'(F')=\sigma_x(F')\leq b(F')$, i.e., F' is b-strict. Conversely, let $F\in\mathcal{F}$ be a b-strict $v\bar{u}$-set. By Lemma 12, F is the intersection of some b'-strict members of $\mathcal{F}'$. One of them is a $v\bar{u}$-set. □

Lemma 13 shows that, in order to get a primal solution x and a potential which satisfy all three optimality criteria, the algorithm developed for intersecting families can be applied without any change for a crossing family as well. The only difference occurs in Step 4 when the optimal dual solution is formed.

Performing Step 4 we shall have an optimal dual solution y to (3). From this we have to get an optimal solution to the dual of (1). For any $F \in \mathcal{F}$ with $y(F) > 0$, F is a b-strict member of $\mathcal{F}$. Let $y'(X_i) := y(F)$ for each X_i where the sets X_i are pairwise co-disjoint b'-strict members of $\mathcal{F}'$ whose intersection is F (see Lemma 12(c)).

It can immediately be seen that this vector y' is an optimal solution to the dual of (1). At this point the proof of Theorem 1 has been completed. In order to complete the algorithm we must be able to find algorithmically the sets X_i mentioned above. The remainder of this section is devoted to this purpose.

Lemma 14. *A b-strict set F, for which the hypergraph $\{P(u): u \in F\}$ is connected, can be obtained constructively as the intersection of pairwise co-disjoint b'-strict members of $\mathcal{F}'$.*

Proof. Let u be a vertex of F. By Lemma 13, $P(u) = \{v$: there is no b'-strict $u\bar{v}$-set$\}$. With the help of oracle (O) we can produce $P(u)$ as the intersection of some b'-strict members of $\mathcal{F}'$. By Lemma 12 we can algorithmically obtain $P(u)$ as the intersection of pairwise co-disjoint b'-strict members of $\mathcal{F}$.

Now suppose that X, Y are two crossing b-strict members of $\mathcal{F}$ and we have obtained co-disjoint b'-strict members X_i and Y_j of $\mathcal{F}'$ such that $X = \bigcap X_i$ and $Y = \bigcap Y_j$. Then $X \cup Y = \bigcap (Z: Z = X_i \cup Y_j$, X_i and Y_j are crossing). Here any set Z is b'-strict thus Lemma 11 applies again. That is, we can get $X \cup Y$ too as the intersection of pairwise co-disjoint b'-strict members of $\mathcal{F}'$. Now Lemma 14 follows since $\{P(u): u \in F\}$ is connected. □

Together with the potential p provided by the algorithm let V_i be defined as in Section 4. Recall that $\mathcal{K}(V_i)$ was the collection of components of the hypergraph $\{P(u): u \in V_i\}$. If $y(F) > 0$, then $F \in \bigcup \mathcal{K}(V_i)$ and apply Lemma 14.

Having finished the algorithmic proof of Theorem 1, we state the corresponding Feasibility Theorem for crossing families. The proof proceeds along the same line as that of the first Feasibility Theorem, so it is omitted.

Feasibility Theorem B. *There exists a* 0–1 *solution to* (1) *if and only if* $\sum b'(X_{ij}) \geq -\delta(\bigcup X_i)$ *for disjoint non-empty sets* $X_1, X_2, \ldots, X_k$ *where each* X_i *is the intersection of pairwise co-disjoint members* X_{ij} *of* $\mathcal{F}'$, $j = 1, 2, \ldots, k_i$.

9. Augmenting circuits

A basic result of network flow theory states that a feasible circulation is of minimum cost if and only if it admits no augmenting circuit with negative weight. In matroid theory a similar theorem, concerning two matroids on a weighted ground set, states that a common independent set of k elements is of maximum weight if and only if there is no augmenting circuit with negative weight in an appropriately defined auxiliary digraph (see [12, 13]). Here we show that these theorems are specializations of our more general result. For another general augmenting circuit result, see [19].

Let x be a feasible 0–1 solution to (1). Form a digraph $D = (V, \bar{A})$ depending on x as follows. Set $\bar{A} = \bar{A}_{\mathrm{B}} \cup \bar{A}_{\mathrm{W}} \cup \bar{A}_{\mathrm{R}}$ where

$$\bar{A}_{\mathrm{W}} = \{vu:\ uv \text{ is a 0-arrow}\},$$
$$\bar{A}_{\mathrm{W}} = \{vu:\ uv \text{ is a 0-arrow}\},$$
$$\bar{A}_{\mathrm{R}} = \{uv: \text{there is no } b'\text{-strict } v\bar{u}\text{-set in } \mathcal{F}'\}.$$

Let

$$d'(e) = \begin{cases} d(e) & \text{if } e \in \bar{A}_{\mathrm{B}}, \\ -d(e) & \text{if } e \in \bar{A}_{\mathrm{W}}, \\ 0 & \text{if } e \in \bar{A}_{\mathrm{R}}. \end{cases}$$

Augmenting Circuit Theorem. *An integer valued* 0–1 *solution to* (1) *is optimal if and only if there is no negative circuit in D with respect to the valuation d'.*

Proof. Let x be optimal. Starting with this x, apply the algorithm. We shall get a potential p such that x and p satisfy the optimality criteria. Let C be any circuit in D with vertices $x_1, x_2, \ldots, x_k$. The length $\lambda(C)$ of C is $\sum_{i=1}^{k} d'(x_i x_{i+1})$ (where $x_{k+1} = x_1$). If $x_i x_{i+1} \in \bar{A}_{\mathrm{B}}$, then $x_i x_{i+1}$ is a 1-arrow, thus $d'(x_i x_{i+1}) = d(x_i x_{i+1}) \geq p(x_{i+1}) - p(x_i)$. If $x_i x_{i+1} \in \bar{A}_{\mathrm{W}}$, then $x_{i+1} x_i$ is a 0-arrow, thus $d(x_{i+1} x_i) \leq p(x_i) - p(x_{i+1})$, that is, $d'(x_i x_{i+1}) \geq p(x_{i+1}) - p(x_i)$. Finally, if $x_i x_{i+1} \in \bar{A}_{\mathrm{R}}$, then $p(x_i) \leq p(x_{i+1})$, that is, $d'(x_i x_{i+1}) \geq p(x_{i+1}) - p(x_i)$. Now we have $\lambda(C) = \sum_{i=1}^{k} d'(x_i x_{i+1}) \geq \sum (p(x_{i+1}) - p(x_i)) = 0$.

Conversely, suppose x is not optimal. Again apply the algorithm starting with this x and the identically zero potential as inputs. Performing the algorithm, since x is not optimal, Case 2 will occur sometimes, say when the Inner Algorithm is applied for x, a potential p and a 0-arrow ab. There is a path from a to b in the auxiliary digraph H with vertices $a = x_1, x_2, \ldots, x_k = b$. If $x_i x_{i+1}$ is a blue arrow in H, then $x_i x_{i+1} \in \bar{A}_{\mathrm{B}}$ and $d'(x_i x_{i+1}) = d(x_i x_{i+1}) \leq p(x_{i+1}) - p(x_i)$. If $x_i x_{i+1}$ is a white arrow in H, then $x_{i+1} x_i$ is a 0-arrow. Thus $x_i x_{i+1} \in \bar{A}_{\mathrm{W}}$ and $d(x_{i+1} x_i) \geq p(x_i) - p(x_{i+1})$ whence $d'(x_i x_{i+1}) \leq p(x_{i+1}) - p(x_i)$. If

$x_ix_{i+1} \in A_R$, then $x_ix_{i+1} \in \bar{A}_R$ and $d'(x_ix_{i+1}) = 0$. Finally $ba \in \bar{A}_W$ and $d'(ba) < p(a) - p(b)$ (since ab violated (5b), that is $d(ab) > p(b) - p(a)$). Hence the length $\lambda(C)$ of the circuit $C = x_1, x_2, \ldots, x_k, x_1$ is $\sum_{i=1}^k d'(x_ix_{i+1}) < \sum_{i=1}^k p(x_{i+1}) - p(x_i) = 0$. □

If we consider the more general form of (1) when d is not restricted to be nonnegative and $f \leq x \leq g$ is required, then the same theorem is true provided that the auxiliary graph $D = (V, \bar{A})$ is defined as follows. $\bar{A} = \bar{A}_B \cup \bar{A}_W \cup \bar{A}_R$ where

$$\bar{A}_B = \{uv: x(uv) > f(uv)\},$$
$$\bar{A}_W = \{vu: x(uv) < g(uv)\},$$
$$\bar{A}_R = \{uv: \text{there is no } b'\text{-strict } v\bar{u}\text{-set } \mathcal{F}'\}$$

and the costs are

$$d'(e) = \begin{cases} d(e) & \text{if } e \in \bar{A}_B, \\ -d(e) & \text{if } e \in \bar{A}_W, \\ 0 & \text{if } e \in \bar{A}_R. \end{cases}$$

10. Orientations

In this last section we present a new application of Edmonds–Giles theorem which, somewhat surprisingly, concerns undirected graphs. Let $H = (V, A)$ be an undirected graph. The following theorem is due to Nash-Williams [17] (see also [5]).

Theorem 15. *H has a k-strongly connected orientation if and only if H is $2k$-edge connected.*

(A directed graph is *k-strongly connected* if $\rho(X) \geq k$ for $0 \subset X \subset V$.)

Suppose that the two possible orientations uv and vu of an edge may have different costs $c(uv)$ and $c(vu)$. We are interested in a minimum cost k-strongly connected orientation of H.

By means of $c(uv)$ define a directed graph $G = (V, E)$. Let E consist of those arrows uv for which $c(uv) > c(vu)$ and if $c(uv) = c(vu)$, then one of uv and vu (it does not matter which one) also belongs to E. Furthermore, let $d(uv) = c(uv) - c(vu)$. Then G is an orientation of H with nonnegative costs on its arrows.

Our purpose is to reverse some arrows of G so that the new digraph will be k-strongly connected and the total weight of reoriented arrows will be maximum. Such a reorientation can be described by means of a 0–1 vector x where $x(e)$ is 1 if e is to be reoriented and 0 otherwise. Set $\mathcal{F}' = \{X: 0 \subset X \subset V\}$ and $b'(X) = \rho(X) - k$ where $\rho(X)$ is the indegree function of G.

Consider the linear program (1) for this G, $\mathcal{F}'$ and b' and observe that a 0–1 vector x is a feasible solution to (1) if and only if it defines a k-strongly connected reorientation of G. Therefore our algorithm can be applied if we show that, in this case, oracle (O) is available. This is indeed the case since the oracle requires a subroutine to decide whether or not there exist $k+1$ arrow-disjoint paths from u to v in a directed graph which is a simple flow problem.

By means of a similar transformation we can algorithmically find a minimum cost k-strongly connected orientation of H which satisfies some additional constraints. For example, it can be required that the indegree of any vertex v should satisfy the inequality $f(v) \leq \rho(v) \leq g(v)$ where f and g are given in advance.

References

[1] J. Edmonds, Submodular functions, matroids and certain polyhedra, in: R. Guy, H. Hanani and J. Schönheim, eds., Combinatorial Structures and their Applications (Gordon and Breach, New York, 1970).

[2] J. Edmonds and R. Giles, A min–max relation for submodular functions on graphs, Annals of Discrete Mathematics 4 (North-Holland, Amsterdam, 1979).

[3] J. Edmonds and R.M. Karp, Theoretical improvements in algorithmic efficiency for network flow problems, J. ACM 19 (1972).

[4] R.L. Ford and D.R. Fulkerson, Flows in Networks (Princeton University Press, Princeton, 1962).

[5] A. Frank, On the orientation of graphs, J. Combin. Theory Ser. B 28(3) (1980) 251–261.

[6] A. Frank, On disjoint trees and arborescences, in: L. Lovász and V.T. Sós, eds., Proc. Algebraic Methods in Graph Theory (North-Holland, Amsterdam, 1981) pp. 159–169.

[7] A. Frank, How to make a directed graph strongly connected? Combinatorica 1(2) (1981).

[8] A. Frank, A weighted matroid intersection algorithm, J. Algorithms 2 (1981) 328–336.

[9] A. Frank, Finding feasible vectors of Edmonds-Giles polyhedra, J. Combin. Theory Ser. B, submitted.

[10] R. Giles, Submodular functions, graphs and integer polyhedra, D. Th., University of Waterloo, Waterloo, Ontario, 1975.

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

[12] S. Krogdahl, A combinatorial proof of Lawler's matroid intersection algorithm, unpublished manuscript, 1975.

[13] E.L. Lawler, Combinatorial Optimization: Networks and Matroids (Holt, Rinehart & Winston, New York, 1976).

[14] L. Lovász, Flats in matroids and geometric graphs, in: P.J. Cameron, ed., Combinatorial Surveys: Proc. 6th British Combinatorial Conference (Academic Press, New York, 1977).

[15] L. Lovász, On two minimax theorems in graph theory, J. Combin. Theory Ser. B 21(2) (1976) 96–103.
[16] C. Lucchesi and D. Younger, A minimax theorem for directed graphs, J. London Math. Soc. 17(2) (1978).
[17] C.St.J.A. Nash-Williams, Well-balanced orientations of finite graphs and unobstrusive odd-vertex-pairings, in: Recent Progress in Combinatorics (Academic Press, New York, 1969).
[18] A. Hoffman, Some recent applications of the theory of linear inequalities to extremal combinatorial analysis, Proc. Sympos. Appl. Math. 10 (1960).
[19] R. Burkard and H. Hamacher, Minimal cost flows in regular matroids, Math. Programming Stud. 14 (1981) 32–47.
[20] E.L. Lawler and C.U. Martel, Flow network formulations of polymatroid optimization problems, Annals of Discrete Mathematics 16 (1982) pp. 189–200.