COMBINATORICA
Bolyai Society – Springer-Verlag

COMBINATORICA **18** (4) (1998) 503–518

ALGORITHMS FOR MEASURING PERTURBABILITY IN MATROID OPTIMIZATION*

GREG N. FREDERICKSON† and ROBERTO SOLIS-OBA‡

Received June 13, 1997

The *perturbability function* of a matroid measures the maximum increase in the weight of its minimum weight bases that can be produced by increases of a given total cost on the weights of its elements. We present an algorithm for computing this function that runs in strongly polynomial time for matroids in which independence can be tested in strongly polynomial time. Furthermore, for the case of transversal matroids we are able to take advantage of their special structure to design a faster algorithm for computing the perturbability function.

1. Introduction

A fundamental problem in the study of dynamic systems is that of measuring how sensitive a problem is to perturbations in its input [17, 18, 22]. A significant limitation of the current methods for sensitivity analysis in combinatorial optimization is that they measure only changes in the solution of a problem produced by perturbations in the value of a single element in its input [17, 21, 27, 28]. In some situations it is necessary to determine the maximum effect that bounded changes in the whole input of a problem can have over the value of its solution, so that sensitivity analysis does not suffice [8, 10, 12, 18, 24]. In this paper we consider the important class of matroid optimization problems, and present efficient algorithms to compute the effect that this type of changes has over their solutions.

The *perturbability function* of a matroid measures the maximum increase in the weight of its minimum weight bases that can be produced by increases of a given total cost on the weights of its elements. We assume a cost model that for

Mathematics Subject Classification (1991): 05B35, 0504, 68R05

* A preliminary version of this paper appeared in the Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms, 1997, pp. 659–668.

† Research of this author partially supported by the National Science Foundation under grant CCR-9322501.

‡ Research of this author partially supported by NSF grant CCR-9322501, by CONACyT-Fulbright and by Universidad Autonoma Metropolitana. This work was done while this author was a graduate student at Prude University.

any element e of a matroid charges $\delta * c(e)$ to increase the weight of e by δ, where $c(e)$ is a non-negative value.

Matroid theory provides an elegant structure that captures the essence of a large and important class of problems [19, 20, 26, 29]. There are matroid optimization problems in computational biology [16], graph theory [8, 19], and electrical networks [20, 26] for which only estimates of the input values are available, or for which changes in the input values are expected. The perturbability functions for these problems can be used to assess the quality of their solutions. In this paper we present a general algorithm for computing the perturbability function for any matroid. Furthermore, for the case of transversal matroids we are able to take advantage of their special structure to design a faster algorithm for computing the perturbability function.

The concept of perturbability function has been considered before in the context of minimum spanning trees [8], shortest paths [12], and maximum flows in planar graphs [24]. Related problems have also been considered. Drangmeister et al. [6] give a constant-approximation algorithm for the problem of spending a fixed budget reducing the weights of the edges of a given graph to minimize the weight of its minimum spanning trees. Berman et al. [2] study the problem of shortening the weights of the edges of a given rooted tree to minimize the sum of the distances from the root to all of the other vertices in the tree.

Our algorithm for computing the perturbability function of a matroid runs in strongly polynomial time for any matroid with a strongly polynomial time independence test. There are two key ideas in the algorithm. The first is a reduction from the problem of computing the perturbability function of a weighted matroid to that of computing the perturbability function of some family F of minors of the matroid. Each minor in the family F is formed by elements of the same weight. The second key idea is a transformation from the latter problem to the membership problem on a matroid polyhedron [3]. Our algorithm computes the perturbability function of a weighted matroid in $O(m^5n^2+m^4n^4\tau)$ time, where m is the number of elements in the matroid, n is its rank, and τ is the time needed to test independence for a set of at most n elements. As we show, the perturbability function is piecewise linear and it has at most mn breakpoints. Our algorithm can compute all the breakpoints of the function within the time bound stated above.

For the case of transversal matroids, we give an algorithm that computes the perturbability function in $O(mn(m+n^2)|E|\log(m^2/|E|+2))$ time, where E is the set of edges in the bipartite graph that defines the transversal matroid (see e.g. [13]). We prove an extension of Hall's Theorem for the class of minors that constitute family F. This extended theorem allows us to solve the membership problem for the matroid polyhedron associated with each submatroid in F by performing a single minimum-cut computation over a bipartite graph.

The rest of the paper is organized as follows. In Section 2 we give some matroid terminology. In Section 3 we describe our general algorithm for computing the perturbability function of a matroid, and in Section 4 we present our more efficient algorithm for transversal matroids.

2. Matroid terminology

A *weighted matroid* $M=(E,\mathcal{I},w,c)$ consists of a finite set E of *elements* and a collection $\mathcal{I}$ of subsets of E satisfying well-known axioms (see e.g. [29]). Function w assigns a non-negative weight $w(e)$ to each element $e \in E$. We assume that the weight of an element is not a static quantity, but it can be increased at a certain cost. For any value $\delta \geq 0$, the cost of increasing the weight of some element e by δ is $\delta * c(e)$. Set E is called the *ground set* of M, and the subsets in $\mathcal{I}$ are called the *independent sets* of the matroid.

An independent set of maximum cardinality is a *base* of M. The number of elements in any base is the *rank* of the matroid. For any set $S \subseteq E$, $rank(S,M)$ denotes the size of the largest independent subset of S. We denote by n the rank of a matroid and by m the size of its ground set. A set $S \subseteq E$ not in $\mathcal{I}$ is called a *dependent set*. A minimal dependent set is a *circuit*.

The perturbability function of a matroid $M=(E,\mathcal{I},w,c)$ is denoted as F_M. For any vector $x \in \mathbb{R}^E$ let $r_M(x)$ denote the minimum weight of a base of M with respect to x as the weight function. It is interesting to note that this function is strongly related to the Lovász extension for submodular functions (see e.g. [11]). With the help of r_M we can define formally the perturbability function: for any value $b \geq 0$, $F_M(b) = \max\{r(w+x) - r(w) \mid cx \leq b\}$. It is easy to see that function r_M is concave, and thus that F_M is also concave.

For any set $T \subseteq E$, the *restriction* of M to T, denoted as $M|T$, is the matroid whose ground set is T and whose independent sets are $\{S \mid S \in \mathcal{I}$ and $S \subseteq T\}$. The *contraction* of T from M, denoted as M/T, is the matroid with ground set $E-T$ and independent sets $\{S \mid S \subseteq E-T,\ S \cup Y \in \mathcal{I}\}$, where Y is any base of $M|T$. Any matroid derived from M by repeated application of these two operations is called a *minor* of M.

Let $M_1=(E_1,\mathcal{I}_1,w_1,c_1)$ and $M_2=(E_2,\mathcal{I}_2,w_2,c_2)$ be matroids and $E_1 \cap E_2 = \emptyset$[1]. The *sum* of M_1 and M_2, denoted as $M_1 \oplus M_2$, is the matroid $(E_1 \cup E_2, \mathcal{I}_{1\cup 2}, w_1 \cup w_2, c_1 \cup c_2)$, where $\mathcal{I}_{1\cup 2} = \{I_1 \cup I_2 \mid I_1 \in \mathcal{I}_1, I_2 \in \mathcal{I}_2\}$.

The convex hull of incidence vectors of the independent subsets of a matroid M form a *matroid polyhedron* $\mathcal{P}(M)$. Let $\mathbb{R}_+$ denote the set of nonnegative real numbers. An explicit representation for the matroid polyhedron of a matroid M is given by $\mathcal{P}(M) = \{z \in \mathbb{R}_+^E \mid \sum_{e \in S} z(e) \leq rank(S,M)$ for all $S \subseteq E\}$. Given a vector $x \in \mathbb{R}_+^E$, a *base* of x is a maximal vector $y \in \mathcal{P}(M)$ such that $y(e) \leq x(e)$ for all $e \in E$.

3. The general algorithm

In this section we present an algorithm for computing the perturbability function F_M of a matroid $M=(E,\mathcal{I},w,c)$. The basic structure of our algorithm for arbitrary

[1] This condition is not necessary, we introduce it here for simplicity only.

matroids is similar to that in [8] for minimum spanning trees; however, the design of the key subroutine requires new ideas that go significantly beyond our work in [8].

Given a set $S \subseteq E$, we define *coverage*(S,M) as the minimum number of elements that any minimum weight base of M shares with S. We say that the weights of the elements in a set S are *lifted* when the weight of every element in S is increased by the same amount. Let *tolerance*(S,M) be the minimum amount by which the weights of the elements in S have to be lifted to decrease *coverage*(S,M). The *rate* of S in M, denoted as *rate*(S,M), is defined as $c(S)/$*coverage*(S,M), where $c(S)$ is the sum of the costs of the elements in S. Observe that if the weight of each element in S is increased by some value $\Delta \leq$ *tolerance*(S,M), then the value of *rate*$(S,M) * \Delta$ gives the cost for increasing the weight of every minimum weight base of M by at least Δ.

Our approach for computing $F_M(b)$ for a given value $b \geq 0$, consists in lifting the weights of the elements in a set S of smallest *rate* in M up to the point where *coverage*(S,M) decreases. Then a new set S of smallest *rate* is chosen and the process is repeated until the total cost of the weight increases reaches a desired value, b. The algorithm is the following.

Algorithm *uplift* (M,b)
balance $\leftarrow b$
orig_wgt $\leftarrow$ weight of a minimum weight base of M
while *balance* >0 **do**
  Find a set $S \subseteq E$ of smallest *rate* in M.
  Lift the weights of the elements in S by
  $$\Delta = \min\{\, tolerance(S,M), balance/c(S)\,\}.$$
  balance $\leftarrow$ *balance* $- \Delta * c(S)$
end while
new_wgt $\leftarrow$ weight of a minimum weight base of M
return $(new\_wgt - orig\_wgt)$

To illustrate how the algorithm works, consider the following simple example. Let $M = (E,\mathcal{I},w,c)$ be a matroid with ground set $E = \{e_1,e_2,e_3,e_4,e_5\}$ and in which a set $S \subseteq E$ is independent if and only if $|S| \leq 2$. The weights of the elements are $1,1,1,2$, and 4, respectively, and their costs are $1,4,1,1$, and 8. Let $b=27$. In the first iteration of the while-loop, *uplift* selects subset $S=\{e_1,e_3\}$ with *rate* 2 and *tolerance* 1, and lifts the weights of these elements to 2. The cost of these weight increases is 2, and so the remaining budget has value 25. The second set selected by *uplift* is $S=\{e_1,e_3,e_4\}$ with *rate* 3 and *tolerance* 2. Algorithm *uplift* spends 6 units of the budget lifting the weights of the elements to 4. In the following two iterations *uplift* selects set $\{e_2\}$ with *rate* 4 and *tolerance* 3, and set $\{e_1,e_2,e_3,e_4\}$ with *rate* 7 and *tolerance* ∞. The final weights of the elements are $5,5,5,5$, and 4, and $F_M(27)=(4+5)-(1+1)=7$.

3.1. Analysis of the algorithm

Fix a matroid $M=(E,\mathcal{I},w,c)$ and a budget value $b^*\geq 0$. Let increases of total cost b^* be made on the weights of the elements of E so as to maximize the minimum weight of any base of M. Let w^* be a function that gives these increased weights, and let $\mathcal{A}$ be an algorithm that given M and b^* correctly computes $F_M(b^*)$ and w^*. Without loss of generality we assume that $\mathcal{A}$ works in several stages. In each stage it selects some set of elements and lifts their weights by a certain amount, in such a way that at the end of all stages the weight of every element $e\in E$ is $w^*(e)$.

We show that the increase in the weight of the minimum weight bases of M produced by *uplift* is the same as that produced by $\mathcal{A}$. For this purpose only, it is convenient to imagine that whenever $\mathcal{A}$ or *uplift* increases the weight of an element e by some amount Δ, it does not just add Δ to $w(e)$, but gradually raises the weight from $w(e)$ to $w(e)+\Delta$. This allows us to consider the partial solutions built by the algorithms when any fraction b of the budget b^* has been spent by them increasing element weights. For any value b, $0\leq b\leq b^*$, let S_b be the set of elements whose weights are being lifted by *uplift* when the budget spent by it reaches value b, let w_b describe the weight function at that precise moment, and let $M_b=(E,\mathcal{I},w_b,c)$. Similarly, let S_b^* be the set of elements whose weights are being lifted by $\mathcal{A}$ when the budget spent by it reaches value b, let w_b^* be the weight function at that precise moment, and let $M_b^*=(E,\mathcal{I},w_b^*,c)$.

To show that *uplift* finds an optimal solution, it is sufficient to prove that $\mathit{rate}(S_b,M_b)=\mathit{rate}(S_b^*,M_b^*)$, for all $0\leq b\leq b^*$. Let $\mathit{sm\_eq}(x,M)$ be the subset of elements in M of weight no larger than x.

Lemma 3.1. *Let* $M=(E,\mathcal{I},w,c)$, $M'=(E,\mathcal{I},w',c)$ *be matroids and* $S\subseteq E$. *If for every element* $e\in S$, $\mathit{sm\_eq}\,(w'(e),M')\subseteq \mathit{sm\_eq}\,(w(e),M)$ *then* $\mathit{coverage}\,(S,M)\leq \mathit{coverage}\,(S,M')$.

Proof. Let B' be a minimum weight base of M' for which $|B'\cap S|=\mathit{coverage}\,(S,M')$. If B' is not a minimum weight base of M, it can be transformed into one by taking every element $e\in E-B'$, one at a time, including e in B' and removing the element of largest weight in the unique circuit that is formed.

Note that by including in B' any element $e\in S-B'$, we create a circuit C in which all elements $a\in C$ have weights $w'(a)\leq w'(e)$ because B' is a minimum weight base of M'. Moreover, since $a\in \mathit{sm\_eq}(w'(e),M')\subseteq \mathit{sm\_eq}(w(e),M)$ then $w(a)\leq w(e)$, and thus it is possible to construct a minimum weight base of M that does not include any element from $S-B'$. This implies that $\mathit{coverage}\,(S,M)\leq \mathit{coverage}\,(S,M')$. ∎

Lemma 3.1 suggests a way in which $\mathcal{A}$ can select the sets S_b^* that makes it easy to compare their rates with those of the sets S_b. We would like that during the construction of the optimal solution, $\mathcal{A}$ always chooses S_b^* to include only elements e for which $\mathit{sm\_eq}(w_b(e),M_b)\subseteq \mathit{sm\_eq}(w_b^*(e),M_b^*)$. If this is possible, then by

Lemma 3.1, $rate(S_b^*, M_b) \leq rate(S_b^*, M_b^*)$. Since *uplift* always chooses the set S_b with smallest *rate* in M_b, then it would follow that $rate(S_b, M_b) \leq rate(S_b^*, M_b^*)$.

We now specify how the sets S_b^* are selected. Fix a value b, $0 \leq b \leq b^*$. For any value $\Delta \geq 0$, let c_Δ be the total cost of bringing the weight of every element e from its initial value $w(e)$ up to $\min\{w^*(e), w_b(e)+\Delta\}$. Let Δ_b be such that $c_{\Delta_b} = b$ and let $w_b^*(e) = \min\{w^*(e), w_b(e) + \Delta_b\}$ for all $e \in E$.

Note that $w_b^*(e) \leq w^*(e)$ and $w_{b^*}^*(e) = w^*(e)$ for all $e \in E$. Select S_b^* to include all elements e for which $w_b^*(e) = w_b(e) + \Delta_b \leq w^*(e)$ and $w_{b-\varepsilon}^*(e) < w^*(e)$ for all $b \geq \varepsilon > 0$.

Theorem 3.1. *Algorithm uplift correctly computes $F_M(b^*)$ for any matroid $M = (E, \mathcal{I}, w, c)$ and any value $b^* \geq 0$.*

Proof. Consider any value $b, 0 \leq b \leq b^*$. Let $e \in S_b^*$. By definition of S_b^* and $w_b^*(e)$ it follows that $w_b^*(e) = w_b(e) + \Delta_b$ and $w_b^*(e') \leq w_b(e') + \Delta_b$ for all $e' \in E$. Hence if $w_b(e') \leq w_b(e)$ then $w_b^*(e') \leq w_b^*(e)$. Therefore, $sm\_eq(w_b(e), M_b) \subseteq sm\_eq(w_b^*(e), M_b^*)$, and by the above discussion $rate(S_b, M_b) \leq rate(S_b^*, M_b^*)$. ∎

Corollary 3.1. *The perturbability function F_M is piece-wise linear, concave and non-decreasing.*

Proof. The proof of Theorem 3.1 shows that any partial solution constructed by *uplift* is optimal for some budget value. Hence, *uplift* "marches" along the whole curve F_M when given as input an infinite budget value. In each iteration of the while-loop, algorithm *uplift* "traverses" a (portion of a) linear segment of F_M whose slope is equal to the inverse of the *rate* of the set S selected during that iteration. Since the slope of F_M can only change when algorithm *uplift* selects a different set, then F_M is piecewise linear.

We have already shown that F_M is concave, and it clearly is non-decreasing. ∎

3.2. Finding a set of smallest rate

In this section we describe an algorithm for finding a set of smallest *rate* in a matroid M. Our approach consists of two stages. In the first stage we form a family of submatroids of M, each formed by elements of the same weight. We show that at least one of these submatroids M_i is such that any subset of elements with smallest *rate* in M_i also has smallest *rate* in M. In the second stage we exploit the fact that all elements in M_i have the same weight to find a set of smallest *rate* in it.

Fix a matroid $M = (E, \mathcal{I}, w, c)$. Let $w_1, w_2, \ldots, w_p$ be the different element weights in M. For a given subset $S \subseteq E$, let $S_{<w_i}$, $S_{=w_i}$, $S_{\leq w_i}$, and $S_{>w_i}$ denote, respectively, the sets formed by all elements in S of weight smaller than w_i, equal to w_i, at most w_i, and larger than w_i.

It is known that the minimum weight bases of a matroid form a base family of a matroid (see e.g. [11, Theorem 3.15]):

Lemma 3.2. (Fujishige) *The minimum weight bases of matroid M coincide with the base family of the matroid*

$$M' = \bigoplus_{i=1}^{p} (M|E_{\leq w_i})/E_{<w_i}. \qquad \blacksquare$$

This lemma allows us to prove that a set of smallest *rate* in M can be found by considering only the submatroids $M_i=(M|E_{\leq w_i})/E_{<w_i}$.

Lemma 3.3. *Let $S \subseteq E$ be a set of smallest rate in matroid M and w_i be the weight of some element in S, then*

1. *$S_{=w_i}$ is a set of smallest rate in M, and*
2. *any set $T \subseteq E_{=w_i}$ of smallest rate in $M_i = (M|E_{\leq w_i})/E_{<w_i}$ has also smallest rate in M.*

Proof. Let $w_1, w_2, \ldots, w_r$ be the different weights of the elements in S. By Lemma 3.2, $rate(S) = \sum_{j=i}^{r} c(S_{=w_j}) / \sum_{j=1}^{r} coverage(S_{=w_j}) = \min\{c(S_{=w_j})/coverage(S_{=w_j}) \,|\, j = 1,\ldots,r\}$ since S has smallest *rate* in M. Therefore, $rate(S) = rate(S_{=w_j})$ for all $j=1,\ldots,r$.

If $T \subseteq E_{=w_i}$ is a set of smallest *rate* in M_i then by Lemma 3.2, $rate(T, M_i) = rate(T, M)$ and so T has smallest *rate* in M. $\blacksquare$

Lemma 3.3 states that we can find a set of smallest *rate* in a matroid M with arbitrary element weights by finding a set S_i of smallest *rate* in each submatroid $M_i=(M|E_{\leq w_i})/E_{<w_i}$, and selecting the set S_j for which the *rate* is minimum. All elements in the submatroid M_i have the same weight, and thus, every base of M_i has minimum weight. This means that for the purpose of computing a set of smallest *rate* in M_i we can consider that M_i is an unweighted matroid. This observation greatly simplifies the problem of computing a set of smallest *rate* in M_i.

Consider an unweighted matroid $M_i = (E_i, \mathcal{I}_i)$ with rank n_i and $m_i = |E_i|$. Define the function $g_i(\lambda) = \min\{ c(T) - \lambda * coverage(T, M_i) \,|\, T \subseteq E_i \}$, for every $\lambda \geq 0$. Let σ_i be the smallest *rate* of any subset of E_i. Then, it is easy to see that σ_i is equal to the largest value of λ for which $g_i(\lambda) = 0$. Note that the slope of the curve g_i at any given point λ is equal to $coverage(S, M_i)$ for some set $S \subseteq E_i$, and thus g_i is piecewise linear and has at most n_i breakpoints. Furthermore, function g_i is non-increasing and its leftmost linear piece has slope 0. The value of σ_i is, then, equal to the value λ of the first breakpoint of g_i.

Since $\sigma_i \leq c(E_i)/n_i$, we can compute the value of σ_i by starting at the point $\lambda = c(E_i)/n_i$ and moving backwards along the curve g_i using Newton's method. This procedure for computing the value of σ_i needs to solve at most n_i parametric problems of the form: $\min\{ c(T)/\lambda - coverage(T, M_i) \mid T \subseteq E_i \}$. Since M_i is an unweighted matroid, then for any set $S \subseteq E_i$, $\mathit{coverage}(S,M_i) = \mathit{rank}(E_i,M_i) - \mathit{rank}(E_i - S, M_i)$. Thus, the above parametric problem is equivalent to

$$\min\{c(T)/\lambda + \mathit{rank}(E_i - T, M_i) \mid T \subseteq E_i\}. \tag{1}$$

Cunningham [3], showed that problem (1) is equivalent to the problem of finding a base for the vector c/λ in the matroid polyhedron $\mathcal{P}(M_i)$. The fastest known algorithm for solving this latter problem is due to Narayanan [23].

Lemma 3.4. *A set of smallest rate in an unweighted matroid $M_i = (E_i, \mathcal{I}_i)$ can be computed in $O(m_i^4 n_i + m_i^3 n_i^3 \tau)$ time, where τ is the time required to test whether a set of at most n_i elements is independent in M_i.*

Proof. We use Narayanan's algorithm [23] to solve problem (1) in $O(m_i^4 + m_i^3 n_i^2 \tau)$ time. Since at most n_i problems of the form (1) have to be solved to find a set of smallest *rate* in M_i, the total time needed to solve the latter problem is $O(m_i^4 n_i + m_i^3 n_i^3 \tau)$. ∎

The process that we have described finds a set S of smallest *rate* in M such that all the elements in S have the same weight, say w_i. To compute $\mathit{tolerance}(S,M)$ we just find the smallest weight w_j to which the weights of the elements in S must be lifted in order to decrease $\mathit{coverage}(S,M)$. If such a weight w_j exists, then $\mathit{tolerance}(S,M) = w_j - w_i$, otherwise $\mathit{tolerance}(S,M) = \infty$. Note that since all elements in S have weight w_i, then $\mathit{coverage}(S,M) = \mathit{coverage}(S, (M|E_{\le w_i})/E_{<w_i})$. Also, since $M_i = (M|E_{\le w_i})/E_{<w_i}$ is an unweighted matroid, then $\mathit{coverage}(S,M_i) = \mathit{rank}(E_{=w_i}, M_i) - \mathit{rank}(E_{=w_i} - S, M_i)$. Thus, $\mathit{coverage}(S,M_i)$ can be computed by performing at most $|E_{=w_i}|$ independence tests in M_i. To test if a set $T \subseteq E_{=w_i}$ is independent in M_i, find a minimum weight base B of M, and test if $(B - B_{=w_i}) \cup T$ is independent in M. Since independence in M_i can be tested within the same time needed to test independence in M, then $\mathit{coverage}(S,M_i)$ can be computed in $O(|E|\tau)$ time, and $\mathit{tolerance}(S,M)$ in $O(|E|^2\tau)$ time.

Lemma 3.5. *Algorithm uplift performs at most $(m-1)n+1$ iterations.*

Proof. Let $w_1 < w_2 < \cdots < w_p$ be the different element weights in the input matroid M. Consider any iteration of the while-loop of *uplift*. Let S be the set of smallest *rate* selected by the algorithm on this iteration. We know from the the above discussion that all elements in S have the same weight. Suppose that in this iteration the algorithm increases the weights of the elements in S by $\Delta = \mathit{tolerance}(S,M)$. By definition of *tolerance* all elements in S have their weights increased to some value w_i, $i \in \{2,3,\ldots,p\}$. Since S intersects every minimum weight base of M, and every minimum weight base has the same number of elements of a given weight, then this iteration increases by at least one the number of elements of weight w_i in every minimum weight base of the matroid.

The number of elements of weight w_i in a base can be at most n, and hence at most n of the iterations can increase the number of elements of weight w_i in every minimum weight base. Since only in the last iteration the weights of the elements in some set S are increased by $balance/c(S)$, the algorithm performs at most $(m-1)n+1$ iterations. ∎

Theorem 3.2. *The perturbability function of a weighted matroid* $M=(E,\mathcal{I},w,c)$ *can be computed in* $O(m^5n^2+m^4n^4\tau)$ *time.*

Proof. A set of smallest *rate* in M is computed by finding sets of smallest *rate* in the submatroids $(M|E_{\leq w_i})/E_{<w_i}$, for all $1\leq i\leq p$. The total time required to solve these subproblems is $\sum_{i=1}^{p} O(|E_{=w_i}|^4 n+|E_{=w_i}|^3 n^3\tau)=O(m^4n+m^3n^3\tau)$. This time dominates each iteration of *uplift*, and since *uplift* performs at most mn iterations, the total time needed to compute the perturbability function is $O(m^5n^2+m^4n^4\tau)$. ∎

Corollary 3.2. *The set of all breakpoints of the perturbability function of a matroid* M *can be computed in* $O(m^5n^2+m^4n^4\tau)$ *time.* ∎

4. Transversal matroids

Given a bipartite graph $G=(D\cup D',E)$, a *transversal matroid* can be defined in terms of matchings in G. A *matching* of G is a subset of edges $S\subseteq E$ such that no two edges in S share a common endpoint. A transversal matroid $M=(D,\mathcal{I},w,c)$ has as ground set, D, the set of vertices in one side of the bipartite graph G. A subset of D is independent in M if it can be covered by a matching. Let $m=|D|$ and n be the rank of M.

We use again algorithm *uplift* to compute the perturbability function of a transversal matroid, but design a more efficient algorithm for solving problem (1) in the computation of a set with smallest *rate*.

Let $M_i=(D_i,\mathcal{I}_i)=(M|E_{\leq w_i})/E_{<w_i}$. Submatroid M_i is a *gammoid* [25]. For any set $S\subseteq D_i$ and vector $x\in\mathbb{R}^{D_i}$ we let $x(S)=\sum_{e\in S}x(e)$. From results in [3, 7], it follows that for any vector $x\in\mathbb{R}^{D_i}$, $\min\{\,x(S)+rank(D_i-S,M_i)\mid S\subseteq D_i\,\}=\max\{y(D_i)\,|\,y\in\mathcal{P}(M_i)\text{ and }y\leq x\}$, where $\mathcal{P}(M_i)$ is the matroid polyhedron for M_i. Problem (1) then is equivalent to

$$\max\{y(D_i)\mid y\in\mathcal{P}(M_i)\text{ and }y\leq c/\lambda\,\}. \tag{2}$$

We derive below an expression for $\mathcal{P}(M_i)$ that allows us to solve efficiently problem (2). Consider again the bipartite graph $G=(D\cup D',A)$ that defines transversal matroid M. For any set $S\subseteq D$, let $\mathcal{N}(S)=\{v\in D'\mid v\text{ is adjacent to some vertex of }S\}$. Hall's Theorem (see e.g. [29]) states that graph G has a matching covering some set of vertices $S\subseteq D$ if and only if $|\mathcal{N}(S')|\geq|S'|$ for all $S'\subseteq S$. Our expression for the matroid polyhedron $\mathcal{P}(M_i)$ will come from a description of the independent sets of the submatroids M_i of the same flavor as that provided by Hall's Theorem.

Let B be a minimum weight base of M, and let $\bar{B}_{=w_i} = B - B_{=w_i}$. For any set $T \subseteq D_i$, we define the function

$$f_{Bi}(T) = \min\{\, |\mathcal{N}(T \cup S)| - |S| : S \subseteq \bar{B}_{=w_i}\,\}. \tag{3}$$

Lemma 4.1. *A set $S \subseteq D_i$ is independent in M_i if and only if $|S \cap T| \leq f_{Bi}(T)$ for all $T \subseteq D_i$.*

Proof. Suppose that set $S \subseteq D_i$ is independent in M_i. Then, $S \cup \bar{B}_{=w_i}$ is an independent set of M, and by Hall's Theorem, $|N(S')| \geq |S'|$ for all $S' \subseteq S \cup \bar{B}_{=w_i}$. Let $T \subseteq D_i$ and $F \subseteq \bar{B}_{=w_i}$. Clearly $(S \cap T) \cup F \subseteq S \cup \bar{B}_{=w_i}$, and thus

$$|\mathcal{N}((S \cap T) \cup F)| \geq |(S \cap T) \cup F| = |S \cap T| + |F|.$$

Since $|\mathcal{N}(T \cup F)| \geq |\mathcal{N}((S \cap T) \cup F)|$, then $|S \cap T| \leq |\mathcal{N}(T \cup F)| - |F|$, and therefore, $|S \cap T| \leq f_{Bi}(T)$ for all $T \subseteq D_i$.

Now, consider the other direction of the implication. Suppose that for some set $S \subseteq D_i$, $|S \cap T| \leq f_{Bi}(T)$ for all $T \subseteq D_i$. This means that $|S \cap T| \leq |\mathcal{N}(T \cup F)| - |F|$ for all $T \subseteq D_i$ and $F \subseteq \bar{B}_{=w_i}$. In particular, by choosing $T \subseteq S$ we get $|\mathcal{N}(T \cup F)| \geq |T| + |F| = |T \cup F|$. Thus, $|\mathcal{N}(T \cup F)| \geq |T \cup F|$ for all $T \cup F \subseteq S \cup \bar{B}_{=w_i}$, and hence, by Hall's Theorem, $S \cup \bar{B}_{=w_i}$ is independent in M. ∎

A set function $h : 2^H \mapsto \mathbb{R}$ is *submodular* if $h(S) + h(T) \geq h(S \cup T) + h(S \cap T)$ holds for all $S, T \subseteq H$.

Lemma 4.2. *Function f_{Bi} is submodular.*

Proof. Let T and T' be subsets of D_i. Let $S \subseteq \bar{B}_{=w_i}$ and $S' \subseteq \bar{B}_{=w_i}$ be such that $f_{Bi}(T) = |\mathcal{N}(T \cup S)| - |S|$ and $f_{Bi}(T') = |\mathcal{N}(T' \cup S')| - |S'|$. Then,

$$f_{Bi}(T) + f_{Bi}(T') = |\mathcal{N}(T \cup S)| + |\mathcal{N}(T' \cup S')| - |S| - |S'|$$
$$= |\mathcal{N}(T \cup S)| + |\mathcal{N}(T' \cup S') - \mathcal{N}(T \cup S)| + |\mathcal{N}(T' \cup S') \cap \mathcal{N}(T \cup S)| - |S \cup S'| - |S \cap S'|$$
$$\geq |\mathcal{N}(T \cup S \cup T' \cup S')| + |\mathcal{N}(T \cap T') \cup \mathcal{N}(S \cap S')| - |S \cup S'| - |S \cap S'|$$
$$\geq f_{Bi}(T \cup T') + f_{Bi}(T \cap T'). \quad ∎$$

We use the following result by Edmonds [7, 4] to give explicit expressions for the independent sets of M_i and for its matroid polyhedron $\mathcal{P}(M_i)$.

Theorem 4.1. (Edmonds) *Let $h : 2^H \mapsto \mathbb{R}$ be a submodular function and $\mathcal{J} = \{\, S \mid S \subseteq H$ such that $|S \cap F'| \leq h(F')$ for all $F' \subseteq H\,\}$. Then $\mathcal{J}$ is the family of independent sets of a matroid on H, and the convex hull of incidence vectors of members of $\mathcal{J}$ is $\mathcal{C} = \{\, y = (y_1, y_2, \ldots, y_{|H|}) \mid 0 \leq y_i \leq 1$ for all $1 \leq i \leq |H|$ and $y(F') \leq h(F')$ for all $F' \subseteq H\,\}$.* ∎

From this Theorem and Lemma 4.2 we derive the following result.

Theorem 4.2. *The family of independent sets of the submatroid* $M_i = (D_i, \mathcal{I}_i) = (M|E_{\leq w_i})/E_{<w_i}$ *is* $\mathcal{I}_i = \{ S \mid S \subseteq D_i$ *and* $|S \cap F| \leq f_{Bi}(F)$ *for all* $F \subseteq D_i \}$ *and its matroid polyhedron is* $\mathcal{P}(M_i) = \{ z = (z_1, z_2, \ldots, z_{|D_i|}) \mid 0 \leq z_i \leq 1$ *for all* $1 \leq i \leq |D_i|$ *and* $z(F) \leq f_{Bi}(F)$ *for all* $F \subseteq D_i \}$.

Theorem 4.2 allows us to solve problem (2) using the polymatroid greedy algorithm (see e.g. [14, 5]). The algorithm is the following.

Algorithm *poly_greedy* $(M_i, f_{Bi}, c, \lambda)$
$y(e_\ell) \leftarrow 0$ for all $e_\ell \in D_i$
for each $e_\ell \in D_i$ **do**
  $\delta_\ell \leftarrow \min\{ f_{Bi}(T) - y(T) \mid e_\ell \in T \subseteq D_i \}$
  $y(e_\ell) \leftarrow \min\{1, \delta_\ell, c(e_\ell)/\lambda\}$
end for

Consider an iteration of the for-loop of algorithm *poly_greedy*. Using the definition of f_{Bi}, the value of δ_ℓ can be rewritten as $\min\{ |\mathcal{N}(T \cup F)| - |F| - y(T) \mid e_\ell \in T \subseteq D_i$ and $F \subseteq \bar{B}_{=w_i} \}$. This minimization problem can be solved by computing a minimum cut in an auxiliary graph $\hat{G}_\ell = (\{s,t\} \cup D_i \cup \bar{B}_{=w_i} \cup D', E_\ell)$ (see Figure 1). Graph $\hat{G}_\ell$ has an edge of infinite capacity from s to e_ℓ, and an edge of capacity $y(e_i)$ from s to each $e_i \in D_i - \{e_\ell\}$. There is an edge of capacity 1 from s to every vertex in $\bar{B}_{=w_i}$, and an edge of infinite capacity from $v \in D_i \cup \bar{B}_{=w_i}$ to $w \in D'$ whenever the edge (v, w) belongs to the bipartite graph G. Finally, $\hat{G}_\ell$ has an edge of capacity 1 from each vertex in D' to t.

Lemma 4.3. *Let* $(R, \bar{R})$ *be a minimum cut of* $\hat{G}_\ell$ *separating* s *from* t*, and let* $c(R, \bar{R})$ *be its capacity. Then,* $c(R, \bar{R}) - y(D_i) - |\bar{B}_{=w_i}| = \min\{ |\mathcal{N}(T \cup F)| - |F| - y(T) \mid e_\ell \in T \subseteq D_i$ *and* $F \subseteq \bar{B}_{=w_i} \}$.

Proof. Without loss of generality assume that $s \in R$. Clearly vertex e_ℓ has to be in R. Observe that for any $v \in R$, $\mathcal{N}(v) \subseteq R$ because otherwise the minimum cut would have infinite capacity, and from Figure 1 it is apparent that a minimum cut of $\hat{G}_\ell$ has finite capacity. Let $R_{D_i} = R \cap D_i$ and $R_{\bar{B}_{=w_i}} = R \cap \bar{B}_{=w_i}$. From Figure 1 we can see that $c(R, \bar{R}) = |\mathcal{N}(R_{D_i} \cup R_{\bar{B}_{=w_i}})| + \sum_{e \in (D_i - R_{D_i})} y(e) + |\bar{B}_{=w_i} - R_{\bar{B}_{=w_i}}|$.

Therefore,

$$c(R, \bar{R}) - \sum_{e \in D_i} y(e) - |\bar{B}_{=w_i}| = |\mathcal{N}(R_{D_i} \cup R_{\bar{B}_{=w_i}}) - \sum_{e \in R_{D_i}} y(e) - |R_{\bar{B}_{=w_i}}|$$

$$= \min\{ |\mathcal{N}(T \cup F)| - \sum_{e \in T} y(e) - |F| : e_\ell \in T \subseteq D_i \text{ and } F \subseteq \bar{B}_{=w_i} \}. \quad \blacksquare$$

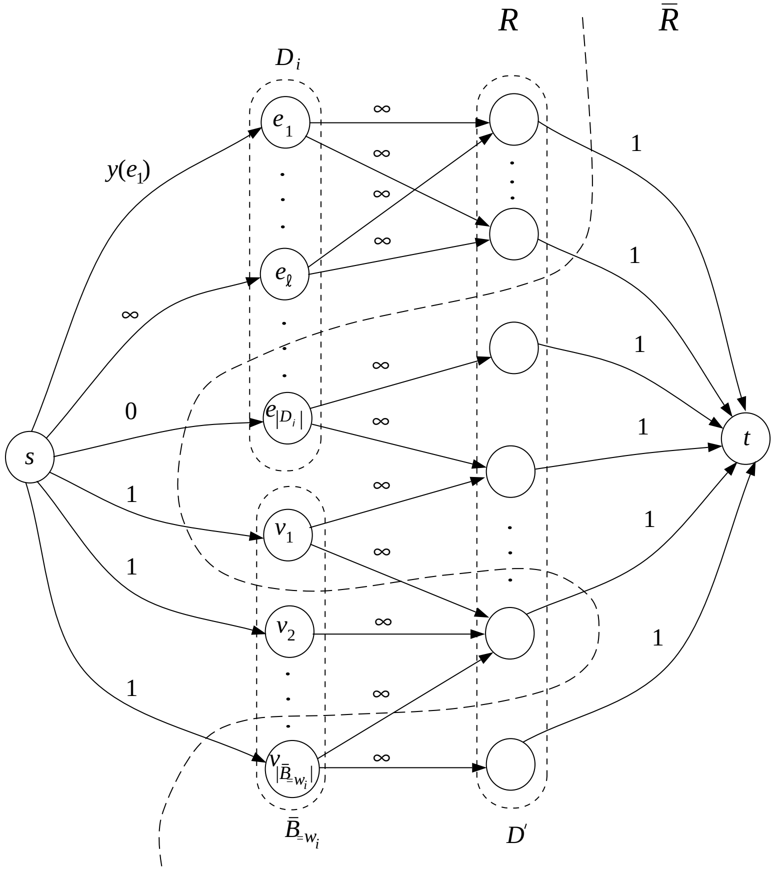

Figure 1. Auxiliary graph $\hat{G}_\ell$ showing a minimum cut $(R, \bar{R})$

Lemma 4.3 allows us to implement *poly_greedy* by performing $|D_i|$ maximum flow computations on auxiliary graphs $\hat{G}_\ell$. However, we can do better than that. Note that in each iteration of the for-loop of *poly_greedy* the value of δ_ℓ can be computed by finding a maximum flow of $\hat{G}_\ell$ that saturates every edge leaving s except the edge (s, e_ℓ) of infinite capacity. By Lemma 4.3, δ_ℓ is equal to the value of the flow going through edge (s, e_ℓ) in this maximum flow. If the value of the flow on edge (s, e_ℓ) is larger than $\min\{1, c(e_\ell)/\lambda\}$, then *poly_greedy* reduces it to $\min\{1, c(e_\ell)/\lambda\}$. Thus, since in each iteration of the for-loop *poly_greedy* essentially computes the maximum flow that can be sent through edge (s, e_ℓ), we can modify the auxiliary graphs so that all values δ_ℓ can be determined with a single maximum flow computation on a new auxiliary graph G_i.

This new auxiliary graph $G_i = (\{s, t\} \cup D_i \cup \bar{B}_{=w_i} \cup D', E_i)$ is built with the same topology and edge capacities as any of the graphs $\hat{G}_\ell$, except that each edge from s to $e_j \in D_i$ has capacity $\min\{1, c(e_j)/\lambda\}$ (there is no edge of infinite capacity incident to s). Let z be a maximum flow for G_i that saturates every edge from s to $\bar{B}_{=w_i}$. (Such a flow must exist since the elements in $\bar{B}_{=w_i}$ form an independent set of the transversal matroid M, and hence, there has to be a matching in G_i covering them.) Let z' be the vector giving the value of the flow z through the edges from s to vertices in D_i.

Lemma 4.4. *The vector z' is a maximizer for problem* (2).

Proof. It is clear that $z'(s,e_j) \le c(e_j)/\lambda$ for every $e_j \in D_i$. We need only prove that z' is a maximal vector in $\mathcal{P}(M_i)$ with this property. Let F and S be, respectively, subsets of D_i and $\bar{B}_{=w_i}$. Since z is a valid flow function for G_i, then it follows that $z(F \cup S) \le |\mathcal{N}(F \cup S)|$. Also, $z(F \cup S) = z(F) + |S|$ because z saturates every edge from s to $\bar{B}_{=w_i}$. Combining these two inequalities we get that $z'(F) \le |\mathcal{N}(F \cup S)| - |S|$ for any $F \subseteq D_i$ and $S \subseteq \bar{B}_{=w_i}$. Therefore, by Theorem 4.2, $z' \in \mathcal{P}(M_i)$.

Vector z' is maximal because the vector y computed by *poly_greedy* forms a flow function for the edges from s to D_i of value no larger than the flow z'. ∎

Lemma 4.5. *A set of smallest rate in the submatroid $M_i = (D_i, \mathcal{I}_i)$ can be computed in $O(|D_i \cup \bar{B}_{=w_i}| + |E_i| \log(|D_i \cup \bar{B}_{=w_i}|^2/|E_i| + 2))$ time.*

Proof. We compute a set of smallest *rate* in M_i using Newton's method, as described in the previous section. Note that each iteration of Newton's method decreases the value of the parameter λ in problem (1) and increases the value of vector c/λ that defines the capacities of the edges from s to D_i in G_i. Hence in two successive iterations of Newton's method, the only change in G_i is an increase in the capacities of some edges leaving the source vertex. Using parametric flow techniques, all iterations of Newton's method can be performed in $O(|D_i \cup \bar{B}_{=w_i}||E_i| \log(|D_i \cup \bar{B}_{=w_i}|^2/|E_i| + 2))$ time using the algorithm FIFO with dynamic trees in [1].

Let z' be the vector giving the value of the final flow through the edges from s to vertices in D_i. Let λ_i be the *rate* of a set of smallest *rate* in M_i. Let $S \subseteq D_i$ be the set formed by all elements $e \in D_i$ such that $z'(s,e) = c(e)/\lambda_i$. Since $\min\{\, c(T)/\lambda_i + \mathit{rank}(D_i - T, M_i) \mid T \subseteq D_i \,\} = \max\{y(D_i) \mid y \in \mathcal{P}(M_i)$ and $y \le c/\lambda_i \,\} = z'(D_i)$, then $z'(D_i) = c(S)/\lambda_i + \mathit{rank}(D_i - S, M_i)$, and therefore S is a set of smallest *rate* in M_i. ∎

Theorem 4.3. *The perturbability function of a transversal matroid M can be computed in $O(mn(m+n^2)|E| \log(m^2/|E|+2))$ time, where E is the set of edges in the bipartite graph that defines M.*

Proof. By the previous lemma, the total time required to find a set of smallest *rate* in M is $O(\sum_{i=1}^{p} |D_i \cup \bar{B}_{=w_i}||E_i| \log(|D_i \cup \bar{B}_{=w_i}|^2/|E_i| + 2)) = O(|E| \log(m^2/|E| + 2)$ $\sum_{i=1}^{p} |D_i \cup \bar{B}_{=w_i}|)$. Observe that the sets D_i and $\bar{B}_{=w_i}$ are disjoint and that $\sum_{i=1}^{p} |D_i| =$

$|D| = m$. Furthermore, note that at most n of the matroids M_i have to be considered when computing a set of smallest *rate* in M. To see why, consider a minimum weight base B of M. Let w_i be the weight of some element in M such that no element of B has weight w_i. Then, the only independent set in matroid $M_i = (M|E_{\leq w_i})/E_{<w_i}$ is the empty set. Therefore, $\sum_{i=1}^{p} |\bar{B}_{=w_i}| \leq n^2$, and the total time to compute a set of smallest rate in M is $O((m+n^2)|E|\log(m^2/|E|+2)$. Algorithm *uplift* adds an additional factor mn to this time. ∎

5. Conclusions

We have defined the concept of perturbability function of a matroid, which generalizes that of sensitivity analysis since it considers simultaneous changes in the weights of all the elements in the input. The perturbability function can be used to assess the quality of the solution of a problem when there are uncertainties or expected changes in its input.

We have presented an $O(m^5n^2 + m^4n^4\tau)$ general algorithm for computing the perturbability function of a matroid. We have also identified key properties of transversal matroids, and exploited them to design a perturbability algorithm that is more efficient than our general algorithm.

We have recently extended the techniques presented in this paper to design efficient algorithms for computing the perturbability function of other interesting classes of matroids. Specifically, we have designed efficient perturbability algorithms for scheduling and partition matroids. We have also considered the problem of evaluating the perturbability function at only a given point, instead of computing all the breakpoints. We have been able to design an optimal algorithm for this version of the problem on partition matroids. These new results will appear in a forthcoming paper [10].

It would be interesting to see if the techniques presented here can be used to design perturbability algorithms for non-matroid optimization problems.

Acknowledgments. We thank the anonymous referees for several helpful comments.

References

[1] R. K. Ahuja, J. B. Orlin, C. Stein, and R. E. Tarjan: Improved algorithms for bipartite network flow, *SIAM Journal on Computing*, **23** (1994), 906–933.

[2] O. Berman, D. I. Ingco, and A. R. Odoni: Improving the location of minisum facilities through network modification, *Annals of Operations Research,* **40** (1992), 1–16.

[3] W. H. Cunningham: Testing membership in matroid polyhedra, *Journal of Combinatorial Theory,* Series B, **36** (1984), 61–188.

[4] W. H. Cunningham: Minimum cuts, modular functions and matroid polyhedra, *Networks,* **15** (1985), 205–215.

[5] W. H. Cunningham: Optimal attack and reinforcement of a network, *Journal of the ACM,* **32** (1985), 549–561.

[6] K. U. Drangmeister, S. O. Krumke, M. V. Marathe, H. Noltemeier, and S. S. Ravi: Modifying edges of a network to obtain short subgraphs, *Theoretical Computer Science,* **203** (1998), 91–121.

[7] J. Edmonds: Submodular functions, matroids, and certain polyhedra, in *Combinatorial Structures,* (R. K. Guy et al. eds.), Gordon and Beach, New York 1970, pp. 69–87.

[8] G. N. Frederickson and R. Solis-Oba: Increasing the weight of minimum spanning trees, *Proceedings of the Seventh Annual ACM-SIAM Symposium on Discrete Algorithms,* 1996, 539–546.

[9] G. N. Frederickson and R. Solis-Oba: Efficient algorithms for robustness in matroid optimization, *Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms,* 1997, 659–668.

[10] G. N. Frederickson and R. Solis-Oba: Efficient algorithms for robustness in resource allocation and scheduling problems, manuscript, 1997.

[11] S. Fujishige: *Submodular functions and optimization*, Elsevier Science Publishers, Amsterdam, 1991.

[12] D. R. Fulkerson and G. C. Harding: Maximizing the minimum source-sink path subject to a budget constraint, *Mathematical Programming*, **13** (1977), 116–118.

[13] H. N. Gabow and R. E. Tarjan: Efficient algorithms for a family of matroid intersection problems, *Journal of Algorithms*, **5** (1984), 80–131.

[14] H. N. Gabow: Algorithms for graphic polymatroids and parametric s-sets, *Proceedings of the 6th Annual ACM-SIAM Symposium on Discrete Algorithms,* 1995, 88–97.

[15] G. Gallo, M. D. Grigoriadis, and R. E. Tarjan: A fast parametric maximum flow algorithm and applications, *SIAM Journal on Computing*, **18** (1989), 30–55.

[16] D. Gusfield, L. Wang, and P. Stelling: Graph traversals, genes and matroids: an efficient special case of the traveling salesman problem, *Tech. Rep. No. CSE-96-3,* University of California, Davis, 1996.

[17] D. Gusfield: Sensitivity analysis for combinatorial optimization, Ph.D. thesis, University of California, Berkeley, 1980.

[18] F. S. Hillier: *Introduction to operations research*, McGraw Hill, New York, 1995.

[19] E. L. Lawler: *Combinatorial optimization: networks and matroids*, Holt, Rinehart and Winston, 1976.

[20] J. Lee and J. Ryan: Matroid applications and algorithms, *ORSA J. Computing*, **4** (1992), 70–98.

[21] M. Libura: Sensitivity analysis for minimum weight base of a matroid, *Control and Cybernetics,* **20** (1991), 7–24.

[22] M. Mansour, S. Balemi, and W. Troul (editors): *Robustness of dynamic systems with parameter uncertainties*, Birkhauser Verlag, 1992.

[23] H. Narayanan: A rounding technique for the polymatroid membership problem, *Linear Algebra and its Applications,* **221** (1995), 41–57.

[24] C. A. Phillips: The network inhibition problem, *Proceedings of the 25th Annual ACM Symposium on Theory of Computing,* 1993, 776–785.

[25] J. S. Pym: The linking of sets in graphs, *J. London Math. Soc.*, **44** (1969), 542–550.

[26] A. Recski: *Matroid theory and its applications*, Springer Verlag, Berlin, 1989.

[27] D. Shier: Arc tolerances in shortest path and network flow problems, *Networks,* **10** (1980), 277-291.

[28] R. E. Tarjan: Sensitivity analysis of minimum spanning trees and shortest path trees, *Information Processing Letters,* **14** (1982), 30–33.

[29] D. J. A. Welsh: *Matroid Theory*, Academic Press, London 1976.

Greg N. Frederickson

Department of Computer Science,
Purdue University,
West Lafayette IN 47907
USA
gnf@cs.purdue.edu

Roberto Solis-Oba

Max Planck Institut für Informatik
66123 Saarbrücken,
Germany
solis@mpi-sb.mpg.de