Part I

COMPUTER SCIENCE

AN ALGORITHM FOR FINDING A MATROID BASIS WHICH MAXIMIZES THE PRODUCT OF THE WEIGHTS OF THE ELEMENTS

T. I. FENNER and A. M. FRIEZE

Department of Computer Science, Birkbeck College, University of London, Malet Street, London WC1E 7HX, England

Department of Computer Science and Statistics, Queen Mary College, University of London, London E1 4NS, England

Abstract.

Consider the problem of finding a spanning tree in an edge-weighted connected graph that maximizes the product of its edge weights, where negative edge weights are allowed. We generalize this problem to matroids and give a polynomial time algorithm for its solution.

1. Introduction.

Given a connected graph G with vertex set V and edge set E, with integer edge weights, there are a number of very efficient algorithms for finding a maximum weight spanning tree, where the weight of a tree is the sum of the weights of the edges of the tree. These algorithms can be extended to solve problems in which we are interested in maximizing the ratio of two different tree weights – see Chandrasekaran [1] and Megiddo [7].

At the Silver Jubilee Conference on Combinatorics in Waterloo 1982, B. McKay raised the question of whether there exists a polynomial time algorithm for finding a maximum weight spanning tree when the weight of a tree is the product of the weights of the edges in the tree. If all the edge weights are non-negative then the problem can easily be solved using Kruskal's algorithm; however, in the general case, when we allow negative weights, it is no longer apparent that a polynomial time algorithm exists.

As Kruskal's algorithm is a special case of the Greedy algorithm for matroids, (see Edmonds [2], Gale [5], Rado [8], and Welsh [10]), it is not surprising that the problem referred to above can be solved in the context of matroids.

Here we use only the most elementary properties of matroids, which can be found for example in Lawler [6] or Welsh [11].

Received September 1984. Revised March 1985.

2. Matroid formulation of the problem.

We assume throughout that $M = (E, \mathscr{I})$ is a matroid, where E is the ground set of M and $\mathscr{I}$ is the set of independent sets of M. We denote the set of bases of M by $\mathscr{B}$.

Let $B \in \mathscr{B}$, then for $e \notin B$ and $f \in B$ we write $B+e$ for $B \cup \{e\}$ and $B+e-f$ for $(B \cup \{e\})-\{f\}$. In addition we denote the unique circuit contained in $B+e$ by $C(B,e)$.

Let Z denote the set of integers and Z^+ the set of non-negative integers. Let $w: E \to Z$, then for $B \in \mathscr{B}$ we define $w(B) = \prod_{e \in B} w(e)$.

Problem 1: Find B^* such that $w(B^*) = \max\{w(B): B \in \mathscr{B}\}$.

The algorithm to be described assumes that M is given in terms of an independence oracle, i.e. there is some procedure which for each $I \subseteq E$ answers the question "is I in $\mathscr{I}$?" in time bounded by a polynomial in $|I|$. Let $E^+ = \{e \in E: w(e) \geqslant 0\}$ and $E^- = E - E^+$. For convenience we refer to the members of E^+ as white elements and those of E^- as black elements. We define a partition of $\mathscr{B}$ by letting $\mathscr{B}_0 = \{B \in \mathscr{B}: |B \cap E^-| \text{ is even}\}$ and $\mathscr{B}_1 = \mathscr{B} - \mathscr{B}_0$. The members of $\mathscr{B}_0$ and $\mathscr{B}_1$ are called even and odd bases, respectively. Thus, if $w'(e) = |w(e)|$ for $e \in E$,

$$(1)\quad w(B) = \begin{cases} w'(B) & \text{for } B \in \mathscr{B}_0 \\ -w'(B) & \text{for } B \in \mathscr{B}_1 \end{cases}.$$

Hence we can solve Problem 1 in polynomial time if the following problem is solvable in polynomial time.

Problem 2: Let $w: E \to Z^+$ be a non-negative weight function and let $E = E^+ \cup E^-$ be a partition of E into white and black elements respectively.
Show $\mathscr{B}_0$ (defined as above) is empty or find $B_0^* \in \mathscr{B}_0$ where

$$(2)\qquad w(B_0^*) = \max\{w(B): B \in \mathscr{B}_0\}.$$

Given an algorithm for solving Problem 2, we can solve Problem 1 as follows: use the Greedy algorithm to find B^* where $w'(B^*) = \max\{w'(B): B \in \mathscr{B}\}$. If $B^* \in \mathscr{B}_0$ then B^* solves Problem 1; otherwise we solve Problem 2 for w'. If $\mathscr{B}_0$ is non-empty then B_0^* solves Problem 1; otherwise we use the Greedy algorithm to find B_1^* where $w'(B_1^*) = \min\{w'(B): B \in \mathscr{B}_1\}$, in which case B_1^* solves Problem 1.

It is therefore of interest to record the following simple result, although it is not needed in the sequel.

Theorem 1: *The following statements are equivalent*: (*a*) $\mathscr{B}_0 \neq \emptyset$ *and* $\mathscr{B}_1 \neq \emptyset$. (*b*) *There exists a circuit C which contains both black and white elements, i.e.* E^+ *does not separate* M.

Gabow and Tarjan [4] consider a problem closely related to Problem 2, that of finding a minimum weight basis with a specified number of black elements. It is, in fact, possible to use their algorithm repeatedly to solve Problem 2. However, our approach leads to a much simpler algorithm which is also more efficient.

3. The algorithm and its analysis.

We now present an algorithm for solving Problem 2.

Greedy – Exchange Algorithm (GEA)

STEP A: sort $E = \{e_1, \ldots, e_n\}$ so that $w(e_1) \leqslant w(e_2) \leqslant \ldots \leqslant w(e_n)$; using the Greedy Algorithm compute B_a where $w(B_a) = \max\{w(B) : B \in \mathscr{B}\}$; if $B_a \in \mathscr{B}_0$, terminate, with $B_0^* = B_a$.

STEP B: {find the best exchange of $f \in B_a$ for $e \notin B_a$ with e and f coloured differently}; let $X = \{(e, f) \colon e \notin B_a, f \in C(B_a, e)$ where e and f are coloured differently$\}$; if $X = \phi$ then $\mathscr{B}_0$ is empty, otherwise define $\bar{e}$, $\bar{f}$ and $B_b = B_a + \bar{e} - \bar{f}$ by $w(B_a + \bar{e} - \bar{f}) = \max\{w(B_a + e - f) \colon (e, f) \in X\}$; terminate with $B_0^* = B_b$.
End of GEA.

We next introduce some notation: for $B = \{e_{i_1}, e_{i_2}, \ldots, e_{i_m}\} \in \mathscr{B}$, where

$$i_1 < i_2 < \ldots < i_m, \text{ let } \ell(B) = (i_1, i_2, \ldots i_m) \in Z^m.$$

We define a total ordering $<$ on $\mathscr{B}$ by $B < B'$ if and only if $w(B) < w(B')$ or $w(B) = w(B')$ and $\ell(B)$ is lexicographically smaller than $\ell(B')$.

In addition we write $e < e'$ if $e = e_i$ and $e' = e_j$ where $i < j$.

It is well known that the basis B_a computed in Step A is the greatest element of $\mathscr{B}$ with respect to $<$.

We now prove the correctness of the algorithm.

LEMMA 1: *Suppose $\mathscr{B}_0$ and $\mathscr{B}_1$ are non-empty and $B_0 < B_1$ where $B_k = \max\{B \in \mathscr{B}_k\}$ for $k = 0,1$. For $e \notin B_0$, let $X(e) = \{f \in C(B_0, e) \colon f < e$ and f is coloured differently to $e\}$. Let $e^* = \max\{e \colon X(e) \neq \phi\}$ and $f^* = \min\{f \in X(e^*)\}$. Then e^* and f^* are well defined and $B_1 = B_0 + e^* - f^*$.*

PROOF: As $B_0 < B_1$ there exists $e \in B_1 - B_0$ and $f \in C(B_0, e)$ such that $B_0 < B_0 + e - f$. However, from the definition of B_0, $B_0 + e - f \in \mathscr{B}_1$ and hence $X(e) \neq \phi$, which implies that e^* and f^* are well defined.

Next let $B = B_0 + e^* - f^*$. The lemma will be proved if we can show that $B = B_1$. So let us assume that $B \neq B_1$. It follows that there exists $e \in B_1 - B$ and

$f \in C(B, e)$ such that $B < B' = B+e-f$. The definition of B_0 implies that $B' \in \mathscr{B}_1$ as $B_0 < B < B'$.

We note that

(3a) $f^* < e^*$; $\quad f^* \in C_1 = C(B_0, e^*)$; $\quad e^*$ and f^* are coloured differently.

(3b) $f < e$; $\quad f \in C_2 = C(B, e)$; $\quad e$ and f are the same colour.

If $f^* = e$ then $f \in X(e^*)$ and $f < f^*$, contradicting the definition of f^*. Thus $f^* \neq e$ which implies that $e \notin B_0$. Let $C_3 = C(B_0, e)$. Then, if $f \in C_3$ we have $B_0 < B_0+e-f \in \mathscr{B}_0$ which contradicts the definition of B_0; therefore

(4) $$f \notin C_3$$

and thus, by (3b),

(5) $$C_2 \neq C_3.$$

Thus $C_3 \nsubseteq B+e$, but $C_3 \subseteq B_0+e$, so

(6) $$f^* \in C_3.$$

By (5), $C_2 \nsubseteq B_0+e$, so $e^* \in C_2$. We shall complete the proof by showing that $f \notin C_1$; for then $f \in C_2 - C_1$ and $e^* \in C_1 \cap C_2$, so there exists a circuit C'' such that $f \in C'' \subseteq (C_1 \cup C_2) - \{e^*\} \subseteq B_0+e$. But then $C'' = C_3$ contradicting (4).

Proof that $f \notin C_1$.

Suppose e and f^* are the same colour; if $f^* < e$, (6) would imply $B_0 < B_0+e-f^* \in \mathscr{B}_0$, contradicting the definition of B_0. Thus $f < e < f^* < e^*$. Now if $f \in C_1$ then $f \in X(e^*)$, which would contradict our choice of f^*.

Suppose, on the other hand, that e and f^* are coloured differently; we show that $e^* > f$. If $f^* > e$ this follows directly from (3); if $f^* < e$ then (6) implies $f^* \in X(e)$, so $e < e^*$ by the definition of e^* and again $e^* > f$ by (3). If $f \in C_1$, we have $B_0 < B_0+e^*-f \in \mathscr{B}_0$, contradicting the definition of B_0. ■

It is now easy to prove.

THEOREM 2: *The Greedy-Exchange Algorithm solves Problem* 2.

PROOF: If the algorithm terminates in Step A with $B_0^* = B_a$ then clearly B_0^* satisfies (2). If this is not the case then $B_a = B_1$, where B_1 is as in Lemma 1. It follows from Lemma 1 that, if $\mathscr{B}_0 \neq \phi$, we can obtain B_0 from B_1 by exchanging one pair of elements. Thus Step B either correctly determines that $\mathscr{B}_0$ is empty or finds B_0^* satisfying (2). ■

The algorithm is clearly polynomial, the execution time being dominated by $O(n^2)$ calls to the oracle.

The spanning tree problem considered in Section 1, which is a special case of Problem 1, can be solved in time $O(|E|\log\beta(|E|,|V|))$ by using the algorithm of Fredman and Tarjan [3] with the improvement of Gabow, Galil and Spencer [3] to find the optimal weight spanning trees (in place of the Greedy Algorithm). Step B of the Greedy-Exchange Algorithm can be implemented in time $O(|E|\alpha(|E|,|V|))$ by using the method described in Tarjan [9] for finding the second best spanning tree. (Here $\beta(m,n)$ is defined to be $\min\{i|\log^{(i)}n \leqslant m/n\}$; we note that, for $m \geqslant n$, $\alpha(m,n) \leqslant \log\beta(m,n) \leqslant \log^* n$.)

Acknowledgement.

We thank the referee for pointing out the relevance of references [3] and [4], thus improving the time bound for Problem 1 to that above.

REFERENCES

1. R. Chandrasekaran, *Minimum ratio spanning trees*, Networks 7 (1977) 335–342.
2. J. Edmonds, *Matroids and the greedy algorithm*, Mathematical Programming 1 (1971) 127–136.
3. M. L. Fredman and R. E. Tarjan, *Fibonacci heaps and their uses in improved network optimization algorithms*, Proceedings of 25th Annual IEEE Symposium on Foundations of Computer Science (1984) 338–346.
4. H. N. Gabow and R. E. Tarjan, *Efficient algorithms for a family of matroid intersection problems*, Journal of Algorithms 5 (1984) 80–131.
5. D. Gale, *Optimal assignments in an ordered set: an application of matroid theory*, Journal of Combinatorial Theory 4 (1968) 176–180.
6. E. L. Lawler, *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, New York (1976).
7. N. Megiddo, *Combinatorial optimization with rational objective functions*, Mathematics of Operations Research 4 (1979) 414–424.
8. R. Rado, *Note on independence functions*, Proceedings of the London Mathematical Society 7 (1957) 300–320.
9. R. E. Tarjan, *Sensitivity analysis of minimum spanning trees and shortest path trees*, Information Processing Letters 14 (1982) 30–33.
10. D. J. A. Welsh, *Kruskal's algorithm for matroids*, Proceedings of the Cambridge Philosophical Society 64 (1968) 3–4.
11. D. J. A. Welsh, *Matroid Theory*, Academic Press, London (1976).