AN ALGORITHM FOR MINIMUM PARTITIONING OF A TRANSVERSAL MATROID*

by

Susumu Sasabe

Memorandum No. UCB/ERL M78/36

23 June 1978

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

An Algorithm for Minimum Partitioning of a Transversal Matroid*

by

Susumu Sasabe

Nippon Electric Company, Ltd.

Kawasaki, Japan

ABSTRACT

This paper presents an algorithm for partitioning a transversal matroid, defined on a finite set E and a family Q of subsets of E, into the minimum number of independent subsets.

The worst case running time is $O(m^2n)$, where m and n are the number of elements in E and Q, respectively.

---

* This research was partially supported by the National Science Foundation Grant MCS 77-09906 and was done while the author was a visitor at the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley.

1. INTRODUCTION

We consider a minimum partitioning algorithm of a transversal matroid. By the minimum partitioning algorithm we mean the algorithm for finding as few as k mutually disjoint independent subsets of the matroid elements.

J. Edmonds gives a theorem which states the necessary and sufficient conditions of a general matroid [1]. In the proof of the theorem, he shows an algorithm for minimum partitioning.

For a transversal matroid, more detailed results are presented by J. Edmonds and D.R. Fulkerson [2]. In particular, conditions of the minimum partitioning of a transversal matroid are derived using network flows. However, their main attention is focused on a creation of a matroid theory and a generalization of various previously known results on partitioning in terms of a matroid, rather than algorithm efficiencies. The minimum partitioning problem of a transversal matroid has potentially many applications in the area of combinatorial optimization. So the computational complexity of the problem is of interest to us.

A transversal matroid is defined on a finite set E and a family Q of subsets of E. Let m and n be the number of elements in E and Q, respectively.

Although the algorithm in [1,2] can be applied to a partitioning problem of any type of matroids, it requires $O(m^3c(m,n))$ steps, where $c(m,n)$ is the number of steps required to test for independence in the transversal matroid [3]. The present algorithm requires only $O(m^2n)$ steps.

2. PARTITIONING OF A TRANSVERSAL MATROID

A TRANSVERSAL MATROID

Let $E = \{e_i \mid i=1,2,\ldots,m\}$ be a set of elements. Let $Q = \{q_j \mid j=1,2,\ldots n\}$ be a family of (not necessarily disjoint) subsets of E. The incidence relations between E and Q can be visualized in the form of a bipartite graph, $G = (E,Q,A)$, where the nodes in one part are members of E, the nodes in the other part are members of Q and the arc (e_i, q_j), which is directed from e_i to q_j, is in A if and only if $e_i \in q_j$.

A matching in a graph $G = (E,Q,A)$ is a set of arcs $B \subseteq A$ such that no two elements of B meet at a common node. A set of nodes $T \subseteq E$ is a partial transversal of Q if there is a matching in $G = (E,Q,A)$ which covers all nodes in T. For any family Q of subsets of E, $M = (E,F)$ is a transversal matroid, where F is the set of partial transversals of Q. A subset I in F is called an independent subset.

A PARTITIONING PROBLEM

A partition of a matroid $M = (E,F)$ separates the set E into k mutually disjoint independent subsets $I_1, I_2, \ldots, I_k$. We shall use network flows to produce a partition.

Consider the directed graph shown in Figure 1. In Figure 1 we have, in addition to the bipartite graph defined on E and Q, a source node s, a sink node t and arcs which are directed from s to e_i and from q_j to t. The flow capacities of arcs (s,e_i) are all unity. The flow capacities of arcs (q_j,t) are all equal to k.

The following theorem states a relation between partitioning of a transversal matroid and network flows.

Theorem 1 (Edmonds and Fulkerson [2]). There is an integral flow of value m from s to t if and only if there is a partition of a transversal matroid $M = (E,F)$ into k independent subsets.

Proof. The proof is constructive and based on the integrity theorem and max-flow min-cut theorem for network flows [4]. Let us take a path decomposition of the flow. For each e_i, there is exactly one path which includes the arc (s, e_i).

For each q_j, there are at most k paths which include the arc (q_j,t). We put labels, 1 through at most k, to these paths associated with q_j such that each path has a distinct label.

After the labeling, we classify all paths according to their labels. From the nature of labeling, a set of arcs, (e_i, q_j)'s, which are in the paths with a same label, is a matching in the subgraph $G = (E,Q,A)$. From the definition, the set of nodes, e_i's, in the matching is an independent subset of the transversal matroid.

Using a converse procedure, k mutually disjoint independent subsets of a transversal matroid yield an integral flow of value m from s to t.

By theorem 1, the minimum partitioning problem of a transversal matroid becomes a problem of finding the minimum value of k which admits a flow of value m in the flow network shown in Figure 1.

One of the important results in [2] is a derivation of a transversal theorem using network flows. This suggests computationally good algorithms.

3. A MINIMUM PARTITIONING ALGORITHM

In this section we present the minimum partitioning algorithm, i.e., an algorithm for finding the minimum value of k which admits a flow of value m in the flow network of Figure 1. We shall introduce some terminology which is used to describe the algorithm. Since the nature of the algorithm is similar to those of a maximal flow algorithm [4] and a bipartite matching algorithm [5], some same terminology will be used. However, their meanings are slightly changed so that they are appropriate for our problem.

An arc is said to be saturated if the amount of flow through the arc is equal to its flow capacity. With respect to a given flow, let S be a set of saturated arcs. A node e_i in E is said to be free if all arcs incident with the node are not in S. A node q_j in Q is said to be free if the arc from q_j to t is not in S.

By ignoring the directions of arcs, an alternating path is defined to be a path of arcs which are alternately in S and not in S. An alternating tree is defined to be a tree which satisfies the following two conditions. (1) The tree contains exactly one free node in E, which we call its root. (2) All paths between the root and any other node in the tree are alternating paths. An augmenting path is an alternating path between a free node in E and a free node in Q.

With respect to an augmenting path, there is an unique path called a s-t augmenting path, which consists of three components, i.e., the arc from s to the free node in E, the augmenting path and the arc from the free node in Q to t.

The s-t augmenting path corresponds to a flow augmenting path of maximal flow algorithm. If a s-t augmenting path exists, the flow can be

decreased by unity in each arc in S and can be increased by unity in each arc not in S. The result is the augmented flow. From the augmenting path theorem [4], it follows that a flow from s to t is maximal if and only if it admits no s-t augmenting path.

DESCRIPTION OF THE ALGORITHM

The computational procedure is begun with the zero flow and unity value of k. An alternating tree rooted to a free node in E is constructed by means of a labeling technique. Eventually, one or the other of two events must occur. Either a free node in Q is added to the tree, or else it is not possible to add more nodes and arcs to the tree. In the former case, we have an augmenting path and the value of the flow can be augmented by unity using a corresponding s-t augmenting path. Then the root becomes a non-free node and the tree-constructing procedure is repeated for another free node in E.

In the latter case, no augmenting path exists and the flow is maximal under the current value of k. Then we increase the value of k by unity and the tree-constructing procedure is repeated. The computation is terminated when no free node in E exists. The algorithm steps are summarized as follows.

Step 0 (Start) $i = 1$; $k = 1$; $S = \emptyset$.

Step 1 (Labeling)

(1.0) Give the label "f" to a free node e_i in E.

(1.1) If all labeled nodes have been scanned, go to Step 2. Otherwise, find a node v with an unscanned label. If $v \in E$, go to Step 1.2; if $v \in Q$, go to Step 1.3.

(1.2) Scan the label on node v ($v \in E$) as follows. For each arc (v, v') not in S, give node v' the label "v", unless node v' is already labeled. Return to Step 1.1.

(1.3) Scan the label on node v ($v \in Q$) as follows. If node v is free, go to Step 3. Otherwise, for each arc (v, v') in S, give node v' the label "v". Return to Step 1.1.

Step 2 (Increasing of k) No augmenting path has been found. $k = k + 1$; remove all labels; go to Step 1.

Step 3 (Augmenting) An augmenting path has been found. Augment the flow using a s-t augmenting path which corresponds to the augmenting path. Update S according to the new flow.

Step 4 If $i = m$, then halt. Otherwise, $i = i + 1$; remove all labels; go to Step 1.

The labels given in Step 1 serve to conduct a form of backtracking in Step 3. The nodes in the augmenting path are identified by the backtracking from the free node in Q. The initial node in the path is identified with the label "f".

CORRECTNESS OF THE ALGORITHM

Theorem 2. The algorithm finds the minimum value of k which admits a flow of value m from s to t.

Proof. Let k^* be the minimum value of k. Let k' be the value when the algorithm halts.

The number of augmentations is equal to m because one free node in E becomes a non-free node after each augmentation and the existence of the augmenting path is guaranteed by the increasing of k in Step 2. Hence, the algorithm produces a flow of value m. It follows that $k' \geq k^*$.

At the $(k - 1)$ st stage of the algorithm, there is at least one free node in E which has no augmenting path. This implies that the flow is maximal at that moment and its value is less than m. Hence $k' - 1 < k^*$. Since k is integral, $k' = k^*$.

By Theorems 1 and 2, the algorithm finds as few as k mutually disjoint independent subsets of transversal matroid elements. The path decomposition procedure described in the proof of Theorem 1 gives information on partition from the flow resulted by the algorithm.

COMPLEXITY OF ALGORITHM

We can evaluate the complexity of the algorithm as follows. The number of augmentations is equal to m and the increasing of k in Step 2 guarantees that the number of repetitions of Step 1 for each free node in E is at most 2. Therefore, the loop, Step 1 through Step 4, is executed $O(m)$ times.

In the inner loop of Step 1, Step 1.1 requires $O(m + n)$ steps. Since the time spent in each execution of Steps 1.2 and 1.3 is bounded by the number of arcs incident with node v and Steps 1.2 and 1.3 are executed only once for the given v, the total time spent in Steps 1.2 and 1.3 is $O(mn)$. Hence Step 1 requires $O(mn)$ steps.

Step 2 requires $O(m + n)$ steps because the number of arcs between q_j and t is equal to n and the number of labels is equal to $m + n$.

Step 3 requires $O(mn)$ steps because the number of steps required by the backtracking procedure and the updating S is bounded by the number of arcs in the flow network. And Step 4 requires $O(m + n)$ steps for removing labels.

Hence, the overall computational complexity of the algorithm is $O(m^2n)$.

4. CONCLUSION

An algorithm for minimum partitioning of a transversal matroid has been presented. The algorithm has a worst case running time of $O(m^2n)$. The minimum partitioning problem is closely related to the maximum packing problem; i.e., the problem of finding the maximum number of disjoint maximal independent subsets which are called bases. The present algorithm can be applied to the maximum packing problem of a transversal matroid with minor modifications and would yield $O(m^2n)$ running time.

ACKNOWLEDGEMENT

The author wishes to thank Richard M. Karp and Eugene L. Lawler for their helpful comments and suggestions.

REFERENCES

[1] J. Edmonds, "Minimum Partition of a Matroid into Independent Subsets," J. Res. NBS (Math. and Math Phys.), Vol. 69B, Nos. 1 and 2, p. 62-72 (1965).

[2] J. Edmonds and D.R. Fulkerson, "Transversals and Matroid Partition," J. Res. NBS (Math and Math Phys.), Vol. 69B, No. 3, p. 147-153 (1965).

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

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

[5] C. Berge, "Two Theorems in Graph Theory." Proc. Nat. Acad. Sci. USA, Vol. 43, p. 842-844 (1957).

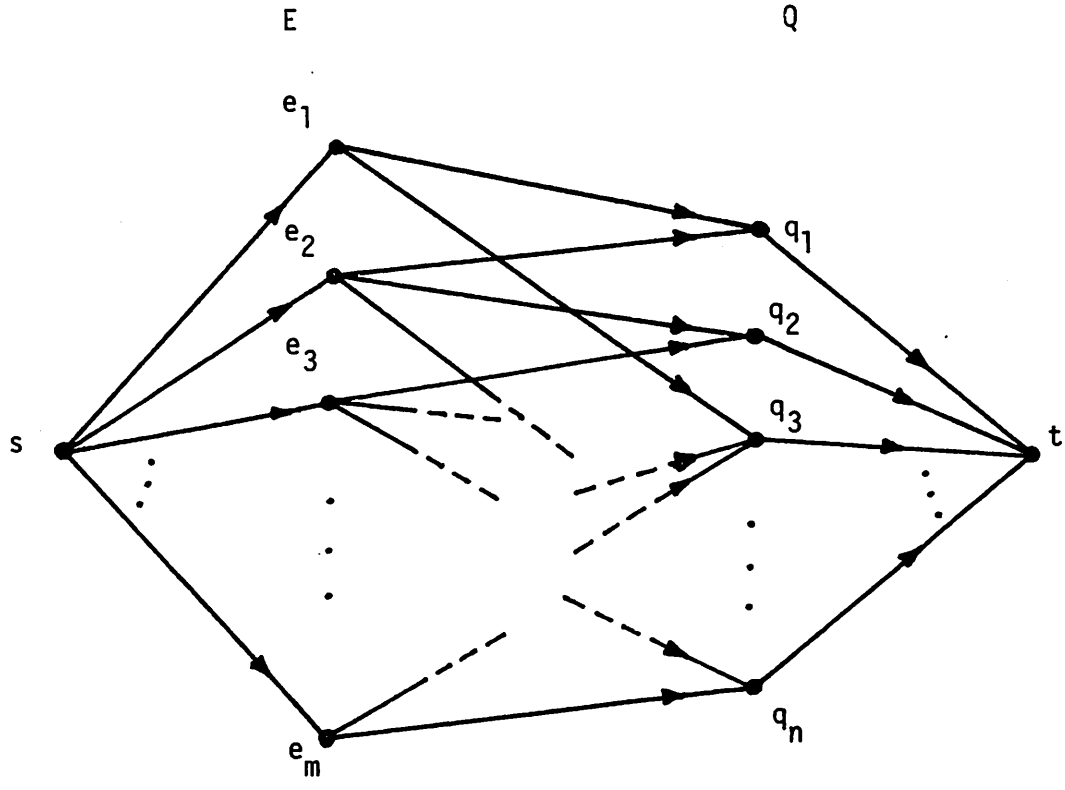

Figure 1. A flow network