JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS: Vol. 45, No. 3, MARCH 1985

TECHNICAL NOTE

Efficient Spanning Trees

H. W. CORLEY[1]

Communicated by C. T. Leondes

Abstract. The definition of a shortest spanning tree of a graph is generalized to that of an efficient spanning tree for graphs with vector weights, where the notion of optimality is of the Pareto type. An algorighm for obtaining all efficient spanning trees is presented.

Key Words. Networks, graphs, trees, Pareto optima, efficient points, algorithms.

1. Introduction

In recent work by Corley and Moon (Ref. 1), the notion of a Pareto shortest path was defined for graphs with vector weights associated with the arcs, and an algorithm for obtaining all such paths was developed. This brief note is essentially an addendum to Ref. 1. The concept of a shortest spanning tree of a graph with vector weights is extended through Pareto optimality to that of an efficient spanning tree (EST), and an algorithm for finding all EST's of a graph is presented.

The following definitions are needed; standard results and notation for graphs are summarized in Ref. 2. A point

$$v^* = (v_1^*, \ldots, v_m^*) \in V \subset R^m$$

is a Pareto minimum (or efficient point or vector minimum) of V if v^* is nondominated from below on V, i.e., if there does not exist $v = (v_1, \ldots, v_m) \in V$ for which

$$v_i \leq v_i^*, \qquad i = 1, \ldots, m,$$

[1] Professor, Department of Industrial Engineering, University of Texas at Arlington, Arlington, Texas.

and

$$v_j < v_j^*, \qquad \text{for some } j \in \{1, \ldots, m\}.$$

We write

$$v^* \in \text{vmin } V.$$

Consider now a connected nondirected graph

$$G = (X, A),$$

where $X = \{x_1, \ldots, x_n\}$ is the set of vertices, $A \subset X \times X$ is the set of arcs $(x_i, x_j) = (x_j, x_i)$, and associated with each arc $(x_i, x_j) \in A$ is an m-dimensional vector weight

$$c_{ij} = (c_{ij1}, \ldots, c_{ijm}).$$

Recall that a spanning tree, or simply tree,

$$T = (X, B)$$

of G is a connected subgraph of G with $n-1$ arcs. We present here an algorithm for finding all EST's of G, i.e., for obtaining all solutions $T = (X, B)$ to

$$\text{vmin}\left\{\sum_{(x_i, x_j) \in B} c_{ij} \colon T = (X, B) \in \tau\right\}, \tag{1}$$

where τ is the collection of all trees of G and the summation in (1) is a vector sum.

2. Algorithm

The following result provides the basis of the algorithm.

Result 2.1. Let $T_s = (X_s, B_s)$ be a subtree of an EST $T = (X, B)$ of $G = (X, A)$ such that $x_k \in X_s$, $x_l \notin X_s$. Then, (x_k, x_l) is an arc in an EST of G if and only if

$$c_{kl} \in \text{vmin}\{c_{ij} \colon x_i \in X_s,\ x_j \notin X_s\}. \tag{2}$$

Proof. Suppose first that

$$c_{kl} \notin \text{vmin}\{c_{ij} \colon x_i \in X_s,\ x_j \notin X_s\}.$$

Then, there exists an arc (x_p, x_q) for which $x_p \in X_s$, $x_q \notin X_s$, and c_{pq} dominates c_{kl}. Let

$$T_1 = (X, B_1)$$

be any tree of G for which $(x_k, x_l) \in B$. Construct a new tree T_2 from T_1 by forming

$$B_2 = [B_1 \backslash \{(x_k, x_l)\}] \cup \{(x_p, x_q)\}.$$

Then,

$$T_2 = (X, B_2)$$

is a tree of G such that

$$\sum_{(x_i, x_j) \in B_1} c_{ij} \text{ is dominated by } \sum_{(x_i, x_j) \in B_2} c_{ij}.$$

It follows that (x_k, x_l) cannot be an arc in any EST to establish the necessity of condition (2).

Suppose next that c_{kl} satisfies (2). There is nothing further to show if $(x_k, x_l) \in B$, so assume $(x_k, x_l) \notin B$. In this case, assume that, in forming T, arc $(x_p, x_q) \in B$ connects T_s to its complement subtree in T. The sufficiency of (2) is then established by constructing

$$B_3 = [B \backslash \{(x_p, x_q)\}] \cup \{(x_k, x_l)\}$$

and noting that

$$T_3 = (X, B_3)$$

is an EST as a consequence of (2). □

An algorithm for obtaining all EST's of G is next presented. The validity of the construction of the subtrees $(X_r(k), A_r(k))$ in the algorithm is an immediate consequence of the necessity and sufficiency of condition (2). Any procedure for determining vmin in Step 2 may be used; one such method is given as Algorithm 2 in Ref. 1. Step 6 below eliminates duplicate subtrees and therefore subsequent computational duplications. It should be noted that, in the case $m = 1$, the algorithm below is most closely related to that of Prim (Ref. 3).

Step 1. Set

$$X_1(1) = \{x_1\}, \qquad A_1(1) = \emptyset, \qquad r = 1, \qquad m_1 = 1, \qquad m_2 = \cdots = m_n = 0.$$

Step 2. Set

$$W_r(k) = \text{vmin}\{c_{ij}: x_i \in X_r(k); x_j \notin X_r(k); x_i, x_j \in A\}, \qquad k = 1, \ldots, m_r.$$

Step 3. Set $s = 1$.

Step 4. If $s = m_r + 1$, go to Step 9.

Step 5. Choose $c_{pq} \in W_r(s)$. Set

$$W_r(s) = W_r(s) \backslash \{c_{pq}\}, \qquad m_{r+1} = m_{r+1} + 1,$$

$$X_{r+1}(m_{r+1}) = X_r(s) \cup \{x_q\},$$

$$A_{r+1}(m_{r+1}) = A_r(s) \cup \{(x_p, x_q)\}.$$

Step 6. If $m_{r+1} = 1$, go to Step 7. If

$$(X_{r+1}(m_{r+1}), A_{r+1}(m_{r+1})) = (X_{r+1}(k), A_{r+1}(k)),$$

for some $k \in \{1, \ldots, m_{r+1} - 1\}$, set $m_{r+1} = m_{r+1} - 1$.

Step 7. If $W_r(s) \neq \emptyset$, go to Step 5.

Step 8. Set $s = s + 1$. Go to Step 4.

Step 9. Set $r = r + 1$. If $r < n$, go to Step 2. Otherwise, stop.

$T_k = (X, A_n(k)), \qquad k = 1, \ldots, m_n$, are the distinct EST's of (X, A).

3. Remarks

Because a graph may contain a large number of EST's, one may be interested in finding a single or only several EST's. It follows from results in, say, Geoffrion (Ref. 4) that one may obtain an EST by applying any standard shortest spanning tree algorithm (see Ref. 2) to the graph with vector weights

$$c_{ij} = (c_{ij1}, \ldots, c_{ijm})$$

replaced for each $(x_i, x_j) \in A$ by the scalars $\sum_{k=1}^{m} \lambda_k c_{ijk}$ for any nondegenerate set of nonnegative λ_k. In particular, finding such a shortest spanning tree with respect to any single component or to the sum of the components yields an EST.

One obvious application of an EST is in the construction of a physical system in which there are multiple incommensurable criteria. For example, a road network (the arcs) to connect a group of population centers (the vertices) might have associated with a potential road both a distance and an environmental damage factor. A road network interconnecting all centers could be chosen from the EST's.

Other possible applications might include the analysis of data structures in computer science and cluster analysis based on multiple factors (for

$m = 1$, see Refs. 5, 6). It is also conceivable that one might define Pareto traveling salesman and network flow problems and that the the notion of an EST might prove useful in their solution as in the standard problems (see Refs. 7, 8).

As a final remark, the application of Pareto optimality to graphs here and in Ref. 1 portends a raft of other similar extensions. One such nontrivial problem in the realm of graphical matching theory is to develop a Hungarian-type algorithm for determining all solutions to a Pareto assignment problem.

References

1. CORLEY, H. W., and MOON, I. D., *Shortest Paths in Networks with Vector Weights*, Journal of Optimization Theory and Applications (to appear).
2. CHRISTOFIDES, N., *Graph Theory: An Algorithmic Approach*, Academic Press, New York, New York, 1975.
3. PRIM, R. C., *Shortest Connection Networks and Some Generalizations*, Bell System Technical Journal, Vol. 36, pp. 1389–1401, 1957.
4. GEOFFRION, A. M., *Proper Efficiency and the Theory of Vector Maximization*, Journal of Mathematical Analysis and Applications, Vol. 22, pp. 618–630, 1968.
5. STANDISH, T. A., *Data Structure Techniques*, Addison-Wesley, Reading, Massachusetts, 1980.
6. ZAHN, C. T., *Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters*, IEEE Transactions on Computers, Vol. C-20, pp. 68–86, 1971.
7. HELD, M., and KARP, R. M., *The Traveling Salesman Problem and Minimum Spanning Trees*, Operations Research, Vol. 18, pp. 1138–1162, 1970.
8. GOMORY, R. E., and HU, R. C., *Multi-Terminal Network Flows*, SIAM Journal on Applied Mathematics, Vol. 9, pp. 551–571, 1961.