

ARBORESCENCE OPTIMIZATION PROBLEMS SOLVABLE BY EDMONDS' ALGORITHM

LEONIDAS GEORGIADIS*

Abstract. We consider a general class of optimization problems regarding spanning trees in directed graphs (arborescences). We present an algorithm for solving such problems, which can be considered as a generalization of Edmonds' algorithm for the solution of the minimum-cost arborescence problem. The considered class of optimization problems includes as special cases the standard minimum-cost arborescence problem, the bottleneck and the lexicographically optimal arborescence problem.

Key words. Minimum Cost Arborescence, Minimum Cost Directed Spanning Tree, Bottleneck Arborescence, Lexicographically Optimal Arborescence.

1. Introduction. Given a directed graph $G = (N, A)$, real number costs c_a for $a \in A$, and a root node r , the minimum-cost arborescence problem is to find the minimum-cost spanning tree in G directed out of r . Here, tree cost is the sum of the tree arc costs. An algorithm for solving this problem has been provided independently by Chu and Liu [3] and Edmonds [4], while Karp [7] provided a combinatorial optimality proof. Efficient implementations have been described by Tarjan [8], Camerini et. al. [2] and Gabow et. al. [5].

In this paper we consider the following optimization problem. We assume that arc costs take values in a set V endowed with a “less than” relation and an “addition” operation and we seek to find the directed spanning tree whose cost (“addition” of all tree arc costs) is minimal with respect to the “less than” operation. We provide an algorithm for solving this problem, which can be considered as generalization of Edmonds' algorithm. Special cases of this problem provide algorithms for the minimum-cost, bottleneck [1], [6] and lexicographically optimal spanning tree.

The paper is organized as follows. In the next section we provide the terminology and definitions used in the paper. In Section 3 we provide an algorithm and show its optimality. In Section 4 we discuss optimization problems that can be solved as special cases of the optimization problem considered in this paper.

2. Terminology and Definitions. Let V be a set endowed with a “less than” relation \preceq and an “addition” operation \oplus having the following properties.

- 1 Relation \preceq is defined for every pair of elements v_1, v_2 , of V . If $v_1 \preceq v_2$, v_1 is called “smaller than” v_2 and v_2 “larger than” v_1 .
- 2 Relation \preceq is transitive, i.e., $v_1 \preceq v_2$ and $v_2 \preceq v_3$ implies $v_1 \preceq v_3$.
- 3 The operation \oplus maps each pair of elements v_1, v_2 , of V to another element $v_1 \oplus v_2 \in V$ and satisfies the following properties
 - (a) commutativity, $v_1 \oplus v_2 = v_2 \oplus v_1$,
 - (b) associativity, $(v_1 \oplus v_2) \oplus v_3 = v_1 \oplus (v_2 \oplus v_3)$.
- 4 If $v_1 \preceq v_2$ and $v_3 \preceq v_4$ then $v_1 \oplus v_3 \preceq v_2 \oplus v_4$.

Note that relation \preceq would be an order relation if we included the antisymmetric property, i.e., $v_1 \preceq v_2$ and $v_2 \preceq v_1$ implies $v_1 = v_2$. However, for our purposes, the antisymmetric property is not needed.

Let $G = (N, A)$ be directed graph $G = (N, A)$ with node set N and arc set A . Denote by $A_+(n)$ the set of arcs in A emanating from node n and by $A_-(n)$ the set

*Aristotle University of Thessaloniki, School of Electrical and Computer Engineering, Telecommunications Dept., Thessaloniki 54006, GREECE. *e-mail:* leonid@eng.auth.gr

of arcs in A terminating at node n . With each arc $a \in A$ there is an associated cost $c_a \in V$. A subgraph $T = (N_T, A_T)$ of G is called an r -arborescence or directed tree out of r , if a) there is a directed path from node r of T to every other node of T using only the arcs in T and b) T has $|N_T| - 1$ arcs, where $|N_T|$ is the cardinality of the set N_T . Node r is called the root of the arborescence. It follows from the definition that for every node $n \neq r$ of T , there is exactly one arc of T terminating at n and there is no arc of T terminating at r . A set of node-disjoint arborescences with roots the set $R = \{r_1, r_2, \dots, r_k\}$ is called R -forest.

Let $G^s = (N^s, A^s)$ be a subgraph of $G = (N, A)$. Consider the cut $[N^s, N - N^s]$ and let A_+^s (A_-^s) be the set of forward (backward) arcs of the cut, i.e., the arcs emanating from (terminating in) N^s and terminating in (emanating from) $N - N^s$. Let N_+^s (N_-^s) be the set of nodes in $N - N^s$ that are endpoints of arcs in A_+^s (A_-^s). Define

$$\begin{aligned} G_e^s &= (N_e^s, A_e^s) \\ &= (N^s \cup N_-^s, A^s \cup A_-^s) \end{aligned}$$

and for $n \in N_-^s$, $A_+^s(n) = A_+(n) \cap A_-^s$.

The cost of G^s is the sum of its arc costs and is denoted by $C(G^s)$. That is,

$$C(G^s) = \sum_{a \in A^s} c_a.$$

where summation is considered with respect to the \oplus operation (the commutativity and associativity of the \oplus operation makes the order of summation irrelevant).

An r -arborescence (R -forest) in G spans G^s if the arborescence (forest) includes all nodes in G^s . If $G = G^s$, we simply say that the r -arborescence (R -forest) spans G .

The definitions above are illustrated in the following example. Consider the graph G on Figure 2.1. Let $N^s = \{3, 4, 5, 7, 8\}$ and let G^s be the subgraph of G induced by the nodes in N^s . Then,

$$\begin{aligned} A^s &= \{(3, 4), (4, 5), (5, 7), (7, 8), (8, 3), (5, 8)\}, \\ N_-^s &= \{1, 2, 6\}, \quad A_-^s = \{(1, 3), (2, 3), (2, 4), (6, 7)\}, \\ N_+^s &= \{6, 9\}, \quad A_+^s = \{(5, 6), (3, 9), (8, 9)\}, \end{aligned}$$

and G_e^s is the graph with node and arc set respectively,

$$\{1, 2, 6\} \cup N^s, \quad \{(1, 3), (2, 3), (2, 4), (6, 7)\} \cup A^s.$$

For node $2 \in N_-^s$, $A_+(2) = \{(2, 3), (2, 4), (2, 6)\}$, and $A_+^s(2) = \{(2, 3), (2, 4)\}$. The graphs

$$\begin{aligned} T_1 &= (\{2, 3, 4, 5\}, \{(2, 3), (3, 4), (4, 5)\}), \\ T_2 &= (\{6, 7, 8\}, \{(6, 7), (7, 8)\}), \end{aligned}$$

constitute an R -forest in G_e^s spanning G^s , where $R = \{2, 6\}$. The 2-arborescence in G consisting of the path

$$2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8,$$

is spanning G^s but is *not* a 2-arborescence in G_e^s since the arc $(5, 6)$ does not belong to A_e^s . In fact, it is important to note that of the nodes in G_e^s only nodes in G^s are terminating nodes for some arcs.

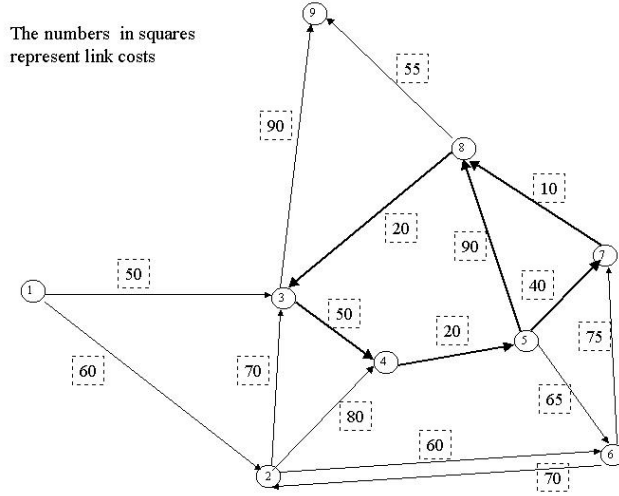


FIG. 2.1. Example Directed Graph G .

3. Optimization Problem and Optimal Algorithm . Let node r be given and assume that there is an r -arborescence spanning G . We are interested in finding the minimum-cost r -arborescence spanning G . More specifically, our objective is to find an r -arborescence spanning G , say T^* , such that if T is any r -arborescence spanning G then

$$C(T^*) \preceq C(T).$$

Since no r -arborescence contains links terminating at node r , we assume without loss of generality that $A_-(r) = \emptyset$.

Let $G^s = (N^s, A^s)$ be a subgraph of G that has the following properties.

- A:** Node r does not belong to N^s
- B:** For any $n \in N_-^s$, there is an n -arborescence in G_e^s spanning G^s .
- C:** For any $n \in N_-^s$, an n -arborescence in G_e^s spanning G^s has smaller cost than any R -forest in G_e^s spanning G^s , where $R \subseteq N_-^s$ and $n \in R$.

Note that the assumption that there is an r -arborescence spanning G together with Property A, imply that N_-^s is nonempty.

Let C_n^s be the minimum-cost n -arborescence in G_e^s spanning G^s . Due to Property B, C_n^s is well defined for any $n \in N_-^s$. Construct a network $\overline{G} = (\overline{N}, \overline{A})$ that replaces all nodes in N_s with a single new node n_s , as follows.

- $\overline{N} = N - N^s \cup \{n_s\}$.
- All arcs in A with endpoints in $N - N^s$ belong to \overline{A} . The cost of these arcs remains the same.
- The arcs in $A_-(n_s)$ are emanating from N_-^s . The arcs in $A_+(n_s)$ are terminating in N_+^s .
- The cost of the arc in $A_-(n_s)$ emanating from node $n \in N_-^s$ is C_n^s .
- The cost of the arc in $A_+(n_s)$ terminating at node $n \in N_+^s$ is

$$\min_{a=(i,n): i \in N^s} \{c_a\},$$

that is, the minimum (with respect to relation \preceq) of the arcs that are terminating at node n and are emanating from some node in N^s . The node

The numbers in squares represent link costs

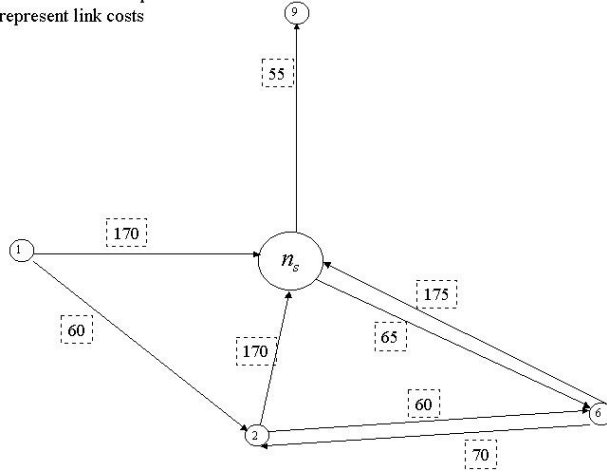


FIG. 3.1. The Contraction of Network G .

$t_n \in N^s$ for which the minimum is achieved will be called *associated* to node n .

We refer to \overline{G} as the *contraction* of network G . It can be shown that the subgraph G^s in Figure 2.1 with arc cost the real numbers shown next to each arc, satisfies properties A-C (see Theorem 3.2 below). Figure 3.1 shows the contraction of network G in Figure 2.1.

Since by assumption there is an r -arborescence spanning G , it is easy to see that there is an r -arborescence (in \overline{G}) spanning \overline{G} as well. Let \overline{T} be a minimum-cost r -arborescence spanning \overline{G} . Since $r \in N - N^s$, \overline{T} contains a unique link (\overline{n}, n_s) , $\overline{n} \in N_-^s$. Construct an r -arborescence T^* spanning G as follows.

- Replace node n_s with a minimum \overline{n} -arborescence in G_e^s spanning G^s .
- For each arc (n_s, n) of \overline{T} , include in T^* the link (t_n, n) , where t_n is the node in N^s associated to n .

The r -arborescence T^* thus constructed is called the *expansion* of \overline{T} .

Next we provide the main theorem on which the construction of the optimal algorithm is based.

THEOREM 3.1. *The r -arborescence T^* constructed with the above procedure is a minimum-cost r -arborescence spanning G .*

Proof. It is clear that T^* is an r -arborescence spanning G . Also, by construction

$$C(T^*) = C(\overline{T}). \quad (3.1)$$

Consider any other r -arborescence T^0 spanning G . Arborescence T^0 must be entering G^s through a subset R of the nodes in N_-^s . Moreover, the set $T^0 \cap G_e^s$ constitutes an R -forest in G_e^s spanning G^s . Let $n_0 \in R$. According to Properties B, C, there is an n_0 -arborescence in G_e^s spanning G^s , that has smaller cost than $T^0 \cap G_e^s$. Consider the r -arborescence T^1 spanning G that results by replacing the set of arcs of $T^0 \cap G_e^s$ by this n_0 -arborescence in G_e^s . Then,

$$C(T^1) \preceq C(T^0). \quad (3.2)$$

Consider now the r -arborescence T^2 spanning G that results by replacing a link of the form $(i, n) \in T^1 \cap A_+^s$, with the link (t_n, n) . Since by definition $c_{(t_n, n)} \preceq c_{(i, n)}$, $i \in A_+^s$,

$$C(T^2) \preceq C(T^1). \quad (3.3)$$

Next, consider the r -arborescence \bar{T}^2 spanning \bar{G} that results by contracting all the nodes in $T^2 \cap G^s$ to a single node n_s and by replacing the cost on link (n_0, n_s) with $C_{n_0}^s$. By construction we have

$$C(\bar{T}^2) = C(T^2). \quad (3.4)$$

Since \bar{T} is a minimum cost r -arborescence spanning \bar{G} , it holds,

$$C(\bar{T}) \preceq C(\bar{T}^2). \quad (3.5)$$

Relations (3.1)-(3.5) imply that

$$C(T^*) \preceq C(T^0).$$

Since T^0 is arbitrary, the results follows. \square

According to Theorem 3.1 if a subgraph G^s of G satisfying properties A-C can be found, the search for the optimal r -arborescence spanning G can be reduced to the search for the optimal r -arborescence spanning the contracted graph \bar{G} . It turns out that Properties A-C are satisfied by the cycles constructed during the course of Edmonds' algorithm. Specifically, let G^s be a subgraph of G with the following property.

Property D There is a directed cycle $(i_0 \rightarrow i_2 \rightarrow \dots \rightarrow i_{m-1})$ ($i_1 = i_{m-1}$ and no other node is repeated), $m \geq 2$, containing all nodes in N^s and such that the cost of arc (i_{k-1}, i_k) , $k = 0, \dots, m-1$, is the minimum of arc costs terminating at node i_k , that is,

$$c_{(i_{k-1}, i_k)} = \min_{a \in A_-(i_k)} \{c_a\}. \quad (3.6)$$

For a node $n \in N_-^s$, let P_n be the set of n -arborescences (directed paths in this case) in G_e^s spanning G^s of the form

$$(n \rightarrow i_k \rightarrow i_{k+1} \dots \rightarrow i_{k-1}).$$

Let T_n^* be a minimum cost path among the paths belonging to P_n . We have the following theorem.

THEOREM 3.2. *A graph satisfying property D also satisfies properties A -C, and a minimum cost n -arborescence in G_e^s spanning G^s is T_n^* .*

Proof. Property A is satisfied since $A_-(r) = \emptyset$ and therefore r cannot belong to a cycle. Property B holds since there is a directed cycle containing all nodes of G^s and there is at least one arc emanating from $n \in N_-^s$ and terminating at some node of G^s .

Consider now any R -forest F in G_e^s spanning G^e , where $R \subseteq N_-^s$ and $n \in R$. Assume without loss of generality that F contains arc (n, i_0) , $i_0 \in N^s$. If A_F is the

set of arcs of F , then taking into account that of the nodes in G_e^s only nodes in G^e are terminating nodes for some arcs, the cost of the forest is

$$c_{(n,i_0)} \oplus \sum_{k=1}^{m-1} \sum_{a \in A_-(i_k) \cap A_F} c_a \succeq c_{(n,i_0)} \oplus \sum_{k=1}^{m-1} c_{(i_{k-1},i_k)},$$

where the inequality is due to (3.6), the fact that there is exactly one arc in the set $A_-(i_k) \cap A_F$ and to Property 4 (see Section 2) that is satisfied by arc costs as elements of V . But the right hand side of this inequality is the cost of the directed path $(n \rightarrow i_0 \rightarrow i_1 \dots \rightarrow i_{m-1})$, which belongs to P_n and therefore has cost larger than T_n^* . Hence, Property C is also satisfied. \square

According to Theorems 3.1 and 3.2, if a cycle G^s in G satisfying Property D is found, then a minimum r -arborescence spanning G can be determined by finding a minimum r -arborescence in the contracted network \overline{G} which has fewer number of nodes. Hence we have the following algorithm for finding the minimum r -arborescence spanning G .

Algorithm A.

Contraction Phase

- 1 Discard all arcs $A_-(r)$. Let G_h be the resulting graph.
- 2 For each node $n \in G_h$ pick an arc with the minimum cost in $A_-(n)$. Let T_h be the graph consisting of the selected arcs and the associated arc endpoints.
- 3 If no cycle is formed, T_h is a minimal r -arborescence spanning G_h . Go to step 6. Else,
- 4 Determine a cycle G_h^s in T_h and form the contracted network \overline{G}_h of G_h .
- 5 Set $G_h \leftarrow \overline{G}_h$ and go to step 2.

Expansion Phase

- 6 Starting from the last r -arborescence T_h , form successively the expansions of the arborescences determined in steps 1-5.

As in [3], simple tests can be added to the previous algorithm to detect the case where no r -arborescence spanning G exists.

According to Theorem 3.2 and the construction of the contracted graph \overline{G} , the cost of arc (n, n_s) , $n \in N_-^s$ is

$$C_n^s = \min_{i_n: (n, i_n) \in A_+^s(n)} \left\{ c_{(n, i_n)} \oplus \sum_{k=n}^{n-2} c_{(i_k, i_{k+1})} \right\}. \quad (3.7)$$

Here, and through the rest of the paper, the addition operation with respect to cycle node subscripts refer to modulo- m operations.

4. Applications. If $V = \Re$, the set of real numbers with the standard order relation and addition operation, then

$$\begin{aligned} C_n^s &= \min_{i_n: (n, i_n) \in A_+^s(n)} \left\{ c_{(n, i_n)} + \sum_{k=n}^{n-2} c_{(i_k, i_{k+1})} \right\} \\ &= \min_{i_n: (n, i_n) \in A_+^s(n)} \{ c_{(n, i_n)} - c_{(i_{n-1}, i_n)} \} + \delta, \end{aligned}$$

where

$$\delta = \sum_{k=0}^{m-1} c_{(i_k, i_{k+1})}.$$

Hence, the algorithm of Section 3 is the same as Edmonds' and Chu and Liu's algorithm with the sole exception that in the latter,

$$\delta = \max_{k=0, \dots, m-1} \{c_{(i_k, i_{k+1})}\}.$$

However, as already observed in [5], this difference in the constants does not affect the resulting arborescence. To see this, note that in the contracted graph \overline{G} , the minimum cost r -arborescence spanning \overline{G} will contain only one of the links incoming to node n_s . Hence, if we add the same constant δ to all links incoming to node n_s in graph \overline{G} , the cost of all r -arborescences spanning \overline{G} will increase by this constant, and hence the minimum cost r -arborescence will not be affected. In fact, we get the same r -arborescence spanning \overline{G} if we set $\delta = 0$, i.e., if we set

$$C_n^s = \min_{i_n: (n, i_n) \in A_+^s(n)} \{c_{(n, i_n)} - c_{(i_{n-1}, i_n)}\}.$$

Consider next the bottleneck arborescence problem defined as follows.

Bottleneck Arborescence: Given a directed graph G , arc costs $c_a \in \mathfrak{R}$ and a root node $r \in N$, construct an r -arborescence T^* in G such that

$$\max_{a \in T^*} \{c_a\} \leq \max_{a \in T} \{c_a\}.$$

This is a special case of the problem described in Section 3 where $V = \mathfrak{R}$, \preceq is the standard order relation between real numbers and $v_1 \oplus v_2 = \max\{v_1, v_2\}$. Optimal algorithms for this problem have been provided previously by Camerini [1] and Gabow and Tarjan [6]. The algorithm that results from Algorithm A is different than either of these algorithms.

For the next problem identify V with the set of K -dimensional real vectors with ordered coordinates, i.e., if $\mathbf{c}_a \in V$ then,

$$\mathbf{c}_a = (c_{a1}, c_{a2}, \dots, c_{aK}),$$

where

$$c_{a1} \geq c_{a2} \geq \dots \geq c_{aK}.$$

Consider the lexicographic order relation, i.e., $\mathbf{c}_a \preceq_{lex} \mathbf{c}_b$ if either $\mathbf{c}_a = \mathbf{c}_b$, or there exists a number l , $1 \leq l \leq K$ such that $c_{ai} = c_{bi}$, for $1 \leq i \leq l-1$ and $c_{al} < c_{bl}$.

If $\mathbf{c}_a, \mathbf{c}_b \in V$, define $\mathbf{c}_a \oplus \mathbf{c}_b$ as the vector in V whose coordinates are the K largest coordinates of the vectors $\mathbf{c}_a, \mathbf{c}_b$, i.e., the K largest numbers in the set

$$\{c_{a1}, c_{a2}, \dots, c_{aK}, c_{b1}, c_{b2}, \dots, c_{bK}\}.$$

It can be verified that the lexicographic order and the \oplus operation thus defined satisfy Properties 1-4 in Section 2 and hence we have an algorithm for solving the following problem.

Lexicographically Optimal Arborescence: Given a directed graph G , arc costs $\mathbf{c}_a \in V$ and a root node $r \in N$, construct an r -arborescence T^* in G such that

$$C(T^*) \preceq_{lex} C(T),$$

where T is any r -arborescence spanning G .

To our knowledge, the resulting algorithm is new. Some special cases of the lexicographically optimal arborescence problem are worth mentioning.

- If $K = 1$, then the problem reduces to the bottleneck arborescence problem.
- If $K = |A|$ and $\mathbf{c}_a = (c_{a1}, 0, 0, \dots, 0)$, $c_{a1} \geq 0$, then we have the problem of finding the lexicographically optimal arborescence in a directed graph whose arc costs are real numbers (not vectors). This is a stricter optimization problem than the bottleneck spanning tree problem.
- If $1 < K < |A|$ and $\mathbf{c}_a = (c_{a1}, 0, 0, \dots, 0)$, $c_{a1} \geq 0$, then we have a problem that is stricter than the bottleneck spanning tree and weaker than lexicographic optimization.

The general case where the form of \mathbf{c}_a is other than $\mathbf{c}_a = (c_{a1}, 0, 0, \dots, 0)$, can also be useful in a situation where each arc actually represents a subnetwork. This is a situation that commonly appears in today's Internet. More specifically, an arc in G may represent a subnetwork belonging to a service provider. The endpoints of the arc represent the "access nodes" to the subnetwork of the service provider. The service provider may supply a vector \mathbf{c}_a that represents the "internal" cost of its subnetwork links when data is transferred between the two access points.

Note also that the general case appears during the course of the optimization algorithm, even if we start with costs of the form $\mathbf{c}_a = (c_{a1}, 0, 0, \dots, 0)$. Indeed, in this case the costs C_n^s of the contracted network will contain more than one nonzero coordinates.

REFERENCES

- [1] P. M. Camerini, "The Min-Max Spanning Tree Problem and Some Extensions", *Information Processing Letters*, v 7, no 1, 1978, pp 10-14.
- [2] P.M. Camerini, L. Fratta, and F. Maffioli, "A note on finding optimum branchings", *Networks*, v.9, 1979, pp.309-312.
- [3] Y. J. Chu and T. H. Liu, "On the shortest arborescence of a directed graph", *Science Sinica*, v.14, 1965, pp.1396-1400.
- [4] J. Edmonds, "Optimum branchings", *J. Research of the National Bureau of Standards*, 71B, 1967, 233-240.
- [5] H. N. Gabow, Z. Galil, T. Spencer and R. E. Tarjan, "Efficient Algorithms for finding minimum spanning trees in undirected and directed graphs," *Combinatorica*, 6 (2) (1986) 109-122.
- [6] H. N. Gabow, R. E. Tarjan, "Algorithms for Two Bottleneck Optimization Problems", *Journal of Algorithms*, 9, 411-417, 1988.
- [7] R. M. Karp, "A simple derivation of Edmonds' Algorithm for Optimum Branching," *Networks*, 1 (1971) 265-272.
- [8] R. E. Tarjan, "Finding Optimum Branchings", *Networks*, v.7, 1977, pp.25-35.