ON THE SHORTEST ARBORESCENCE OF A DIRECTED GRAPH

CHU YOENG-JIN (朱永津) AND LIU THENG-HONG (劉振宏)
(Institute of Mathematics, Academia Sinica)

Abstract
Kruskal and Lott have given the algorithm for finding the shortest spanning subtree of a graph. In many practical problems, we should consider not only the "line-segment" but also the "directional line-segment" for example, we may prepare a scheme of routes to the canal under certain conditions.
In this paper, an algorithm for finding the shortest arborescence of a directed graph is established by induction.

I. The Problem
Given a directed graph $G = (X; U)$, where $X = \{x_0, x_1, \cdots, x_n\}$ is a set of vertices and $U \subseteq X \times X$ is a set of arcs. An arborescence of a directed graph is defined as follows:

Definition.[1] An arborescence of a directed graph $G = (X; U)$ is a subgraph $H = (X; V)$ of $G$ which contains no cycle such that
(a) there is a particular vertex called the root, which is the terminal vertex of every arc in $V$;
(b) for any other vertex $x_i$, there is one and only one arc in $V$, whose terminal vertex is $x_i$.

When there is an arborescence $H = (X; V)$, for any vertex $x$, we associate a non-negative integral number $h(x)$, called the generational number of $x$, relating to $H$, where $h(x)$ is defined as follows:
- If $x_0$ is the root of $H$, let $h(x_0) = 0$.
- If there is an arc $(x_i, x_j) \in V$ and the generational number of $x_i$ has been defined, then define $h(x_j) = h(x_i) + 1$.

Since an arborescence is a tree, all the properties of the tree are true to the arborescence. In particular, when any arc $(x_i, x_j) \in V$ is added to $V$, there will appear a cycle and only a cycle designated by $\mu$.

Set
$$h(x_i) = \min_{\mu \in \tau} h(x_j).$$
If $x_0 = x_i$, then $\mu$ is a loop; otherwise, in $H$ there exists a path $P(x_0, x_i)$ from $x_0$ to $x_i$. This means that in $H$ there exists an arc $(x_0, x_i) \in P(x_0, x_i) \subset \mu$. This is an important property for the proof of our algorithm.

II. Algorithm of the Problem
We show the algorithm through the following example:
Given a directed graph $G = (X; U)$, as indicated in Fig. 1, where $|X| = 9$ and the numbers in brackets are the lengths of the arcs.
Set
$$U'(x_i) = \{(x_i, x_j) | (x_i, x_j) \in U, j = 1, 2, \cdots, n\}.$$

Step 1. For any vertex $x_i$, whose $U'(x_i) \neq \phi$, take the arc $u_i$ such that
$$l(u_i) = \min_{u \in U'(x_i)} l(u)$$
(if such arcs are more than one, take one of them arbitrarily). The set of these arcs is designated by $W_i$. In our concrete case $W_i = \{(x_a, x_b), (x_b, x_c), (x_c, x_d), (x_d, x_e), (x_e, x_f), (x_f, x_g), (x_g, x_h), (x_h, x_i)\}$.

Step 2. If $|W_i| < n - 1$, then the process stops; and there is no arborescence of $G_i$ if $|W_i| > n - 1$, then we choose $n - 1$ arcs in $W_i$, and the set of these $n - 1$ arcs is designated by $Y_i$ such that
$$\max l(u) = \min_{u \in W_i-Y_i} l(u).$$

The choice may not be unique. In our concrete case $V_0 = W_i - (x_a, x_b)$, $(x_a, x_b)$ is the longest arc in $W_i$. ($l_u = 7$).

Step 3. If there is no loop in $V_i$, then the process stops and $H_i = (X; V_i)$ is the shortest arborescence of $G_i$ (the proof is trivial, so it is omitted); otherwise, there exist some loops in $V_i$, for example, $C_1, C_2, \cdots, C_{k_{\Phi}}$, where
$$S = \{u_1, u_2, \cdots, u_{r_0}\}, r_0 = 1, 2, \cdots, k_{\Phi}.$$

We retract $C_i$ into a single vertex $x_i$ and the new graph obtained from the graph $G_i$ by such retraction is designated by $G_i = (X; U_i)$. In our concrete case, $h_0 = 2$. $G_i$ is shown by Fig. 2.

$C_1 = \{(x_b, x_c), (x_c, x_d), (x_d, x_e)\}$
$C_2 = \{(x_f, x_g), (x_g, x_h), (x_h, x_i)\}$. 

\footnote{Received Nov. 7, 1964.}
The length of the arc $u$ of $G_3$ is redefined as follows:

$$l(u) = \begin{cases} 
  l(u), & \text{if the terminal vertex of } u \text{ is not a retractive vertex,} \\
  l(u) + d_i, & \text{if the terminal vertex of } u \text{ is a retractive vertex } \gamma_i, \\
  l(u_i) + d_i, & \text{if the terminal vertex of } u_i \text{ is the arc of } C \gamma_i \text{ which has the same terminal vertex as } u_i \text{ (in graph } G),
\end{cases}$$

where $d_i = \max l(u_i)$.

In Fig. 2, the numbers in brackets are obtained by this formula.

We repeat Steps 1, 2, and 3, until the process stops (it must be stopped at any step). A graph $G_p = (X_p; U_p)$ is obtained. If the process stops at Step 2, the arborescence of $G_p$ (and hence $G$) will not exist; otherwise (i.e., the process stops at Step 3), we obtain the shortest arborescence $H_p = \{X_p; V_p\}$ of $G_p$. In our concrete case, $p = 1$, $H_p$ is shown in Fig. 2 by heavy lines. Then we turn to the following step.

Step 4. We extend $H_p$ to an arborescence $H_{p+1}$ of $G_{p+1}$ (note that $G_0 = G$) in the following manner:

Let $\gamma_i$ be a retractive vertex in $G_p$. The corresponding loop in $G_{p+1}$ is $C_i^{p+1}$. Define $D_i^{p+1} = C_i^{p+1} - \{\gamma_i\}$, $t = 1, 2, \ldots, h_{p+1}$.

When $\gamma_i$ is the root of $H_p$, $u_i$ is one of the longest arcs in $C_i^{p+1}$; otherwise, in $G_p$, there is an arc $u_i \in V_p$ whose terminal vertex is $\gamma_i$, and $u_i$ is an arc in $G_{p+1} (u_i \notin C_i^{p+1})$ also. Suppose its terminal vertex is $x(x \in C_i^{p+1})$ in $G_{p+1}$. Then $u_i$ is the arc in $C_i^{p+1}$ whose terminal vertex is $x$.

For convenience, set

$$X_t = X^{p+1}, \quad t = 0, 1, 2, \ldots, p \quad (\text{where } X_0 = X),$$

Then define

$$V^{p+1} = V^p + \sum_{t=1}^{h_{p+1}} D_t^{p+1},$$

$$H_{p+1} = \{X^{p+1}; V^{p+1}\}.$$

Obviously $H_{p+1}$ is an arborescence of $G_{p+1}$ (note that $V^{p+1}$ is different from $V_{p+1}$). We shall prove that $H_{p+1}$ is the shortest arborescence of $G_{p+1}$ in Section III.

The process is repeated until the shortest arborescence of $G$ (note $G_0 = G$) is obtained. In our concrete case, we obtain the shortest arborescence $H$ by one step only, and $H$ is shown in Fig. 3 by heavy lines.

III. Proof of the Algorithm

Let $HP$ be an arbitrary arborescence of $G_p$ (not necessarily the shortest) and $H_{p+1}$ be obtained from $HP$ by Step 4. We set

$$H_{p+1} = \phi(HP),$$

$$\Phi_{p+1} = \phi(HP) \mid (HP \text{ being any arborescence of } G_p),$$

and denote the set of all shortest arborescences of $G_{p+1}$ by $S$. Then we have the following Lemma. The intersection of $\Phi_{p+1}$ and $S$ is not empty.

Proof. We take a particular arborescence $H_{p+1} = \{X^{p+1}; V^{p+1}\}$ in $S$ such that

$$|V^{p+1} \cap V_{p+1}| = \max_{H \in \Phi_{p+1}} |V \cap V_{p+1}|.$$
and from the formula of Step 3 we obtain
\[ P'(H') = P^*(H'^{-1}) + K_{p-1} \quad (H'^{-1} = \phi(H')) , \]
where
\[ K_{p-1} = \sum_{k=1}^{p-1} [p^{-1}(C_k^{-1}) - d_k^{-1}] \]
is a constant. Hence
\[ P'(H') = \min_{m \in A_p} [P'(H') \leftrightarrow P^*(H'^{-1})] = \min_{m \in A_p} P^*(H') = \min_{m \in A_p} P^*(H) . \]
The proof is completed.

(Note that \( P'(u) = l(u) \) throughout the paper.)
The algorithm can be proved inductively from this theorem.

When we wish to find the shortest arborescence of \( G \) which takes a given vertex \( x \) as root, we must change \( U \) to \( U = U'(x) \) only.

From the definition of arborescence in §1, by changing the phrase “a particular vertex” to “a particular vertices”, we may define the definition of \( k \)-arborescence. The algorithm to find the shortest \( k \)-arborescence is analogous, while the difference is not essential, and so omitted.

For a class of directed graphs (including symmetrical graphs as its special case), the algorithm may be simpler, and Solin’s algorithm may be contained for the shortest tree as a special case.

The consideration of \( k \)-arborescence is first proposed by Ma Chung-fan.

REFERENCES


THERMO-ELASTIC STRESS AND DEFORMATION CAUSED BY A SPHERICAL INCLUSION IN AN INFINITE ELASTIC SLAB*

LIU HEREN-CHIN (刘先智)
(South China Institute of Technology, Tsinan)

ABSTRACT

In this paper are studied the thermal stress and deformation caused by a spherical inclusion in an infinite elastic plate, the inclusion, and the plate having the same elastic behaviour but different coefficients of thermal expansion.

By means of the potential theory, the thermo-elastic displacement function is first found for both the interior and the exterior of the buried sphere; thereupon the pure elasticity boundary value problem concerned is solved in order to restore the free surface condition of the plate.

In the solution of the problem the integrals are left uninterpreted, for it seems to the author that they can be handled conveniently only with numerical treatment.

I. Introduction

The thermal stress problem concerning a semi-infinite elastic solid with a spherical inclusion is solved by using the Love function\(^{(1)}\) and the Galerkin stress function\(^{(2)}\), wherein it was also assumed that the two materials concerned possess the same elastic property but different coefficients of thermal expansion.

The present work is to extend the same investigation to the infinite elastic plate through the Fourier transformation.

II. The Thermal Elastic Displacement Potential in an Infinite Elastic Solid with a Spherical Inclusion

As indicated by Fig. 1, within an infinite elastic plate \(|z| \ll b\), a sphere is buried without initial stress on the boundary surface between the sphere and the plate when the composite body is kept at zero temperature. The radius of the sphere is \(a\), the \(z\)-axis of the coordinate system goes through the centre of the sphere, and \(c\) is the distance from the sphere centre to the coordinate origin. We have to set \(b - c > a\) in order to let the sphere be totally concealed. The two materials of the composite body were assumed to have the same elastic property, but different coefficients of thermal expansion. For brevity, we introduce \(\eta = a_1 - a_2\), where \(a_1\) and \(a_2\) denote the coefficients of thermal expansion of the sphere and the plate respectively.

* Received Dec. 10, 1964.