Journal of the Operations Research
Society of Japan
Vol. 21, No. 4, December, 1978

A NEW METHOD FINDING THE K–TH BEST PATH IN A GRAPH

Hiroaki Ishii
Railway Technical Research Institute

(Received March 4, 1978; Revised June 8, 1978)

Abstract This paper presents a new algorithm determining the K-th best path without any circuit between two specified vertices in a connected, simple and nonoriented graph. The method presented here is based on the well-known fact that the minimum set of ring sum of several Euler graphs and a special path between two vertices consists of all paths between the vertices. Lastly, an illustrative example is given and the efficiency of the algorithm is estimated approximately.

1. Introduction

For the K-th best path problem there exist several available algorithms which approach to the problem by making use of the minimum tree [2], the shortest path algorithm [6], DP[1] or the path algebra [5]. The review of these are given in [4] and [6]. The purpose of this paper is to present a new algorithm which differs from any of them at the point that the edges in a graph are used positively, while in previous techniques the vertices are used more directly.

The algorithm presented here is based on the fact that a minimum set of ring sum of Euler graphs and an arbitrary path between two vertices is a collection of all paths between the vertices. In this paper the K-th best path is defined as the path without any circuit between two given vertices that has the K-th best length, where K≧2 and the K-th best length is allowed to be equal to the (k - 1)th best length. Since this definition is the same as the case assumed the positive length to each edge in a graph in the algorithm of Yen's [6], we try to estimate to show it is practical as well as that of Yen's.

2. Notation

Consider a connected and simple linear graph consisting of n vertices and of nonoriented m edges and assume that to each edge a positive number (length) is assigned. Let P be a path between specified vertices s and t, C be a circuit and E be an Euler graph, all of which are subgraphs of the graph. Note that a path considered in this paper is a subgraph which has not any circuit. Let us use set operations $\bigcup, \bigcap$, - and $\oplus$ by considering a subgraph as a set of edges, for example, $E_1 \oplus E_2 = (E_1 \bigcup E_2) - (E_1 \bigcap E_2)$.

For P, C and E the following results are well known [3].

(1) A collection {E} of all possible Euler graphs is an Abelian group under the ring sum $\oplus$

(2) Let {P} be the collection of all possible paths between s and t. Then

$$\{P\} = \min \{P_1 \oplus E \ ; E \varepsilon \{E\}\}$$

where $P_1 \ \varepsilon\{P\}$ and min A is the subcollection such that if $\alpha, \beta \ \varepsilon$ A, then $\alpha\varepsilon$ min A means $\beta \not\subset \alpha$ as long as $\beta \neq \phi$.

As a set of generators of {E}, we consider a set of fundamental circuits

$$C_1, C_2, \ldots , C_f \ , f = m - n + 1 .$$

Let S be a subgraph. We define the length of S, denoted by L(S), as the sum of lenghs of all edges in it. Assume that by some efficient shortest path algorithm the minimum tree from the origin vertex s have been already obtained, where the minimum tree is a set of edges which determines the shortest path from s to all other vertices in the graph [2]. Each edge in the minimum tree is called a branch and each edge not in it called chord. Let us construct each fundamental circuit c_i $(1 \leq i \leq f)$ such that C_i contains exactly one chord. Let $e_1, e_2, \ldots , e_f$ be chords and let C_i be the fundamental circuit which contains e

We summarize the symbols used in the following section.

T[s,x]	:	The shortest path between s and x.
P	:	The path between s and t.
P_1	:	The shortest path between s and t. P_1 = T[s,t].
P[a,b]	:	The part of P between a and b.

3. Algorithm

Let R_k $(1 \leq k)$ be the collection of the ring sums of P_1 and an Euler graph and also let us denote a collection by { }.

Step 0. Determine R_1 , Q_1 , and E_1 as follows.

$R_1 = \{P_1\}$

$Q_1 = P_1$

$E_1 = P_1 \oplus Q_1 = \phi$

Step 1. For $k \geq 2$, determine R_k as follows.

$$R_k = (R_{k-1} - \{Q_{k-1}\}) \bigcup \{Q_{k-1} + C_i\} ,$$

where $e_i \bigcap E_{k-1} = \phi$

and $Q_{k-1} \oplus C_i \neq Q_\ell$, $(\ell = 1, \ldots, k = 2)$

Step 2. Pick out any one of the subgraphs of minimum length in R_k and set it to Q_k, so that

$$L(Q_k) = \min L(S),\ S \in R_k.$$

Determine E_k by

$$E_k = P_1 + Q_k.$$

Step 3. See if the Q_k is a path or not.

To get the K-th best path, repeat the step 1 through the step 3 until K paths are obtained.

4. Analysis of The Algorithm

Let Q_k be a path P, and let it contains the chords $e_{\alpha 1}, e_{\alpha 2}, \ldots, e_{\alpha_\lambda}$, in this order, between s and t. Let $e_{\alpha_i} = P[u,v]$. Then we have the following Lemma 1.

Lemma 1.

$$T[s,v] \oplus P[s,v] = C_{\alpha_1} \oplus C_{\alpha_2} \oplus \cdots \oplus C_{\alpha_i}$$

$$T[s,v] \oplus P[v,t] = P_1 \oplus C_{\alpha_{i+1}} \oplus C_{\alpha_{i+2}} \oplus \cdots \oplus C_{\alpha_\lambda}$$

Proof.

Since $T[s,v] \oplus P[s,v]$ is an Euler graph and the circuits are uniquely determined by the chords, we have the former part of the Lemma. About the latter half,

$$T[s,v] \oplus P[v,t] = T[s,v] \oplus P[s,v] \oplus P$$
$$= C_{\alpha_1} \oplus C_{\alpha_2} \oplus \cdots \oplus C_{\alpha_i} \oplus P$$
$$= P_1 \oplus C_{\alpha_{i+1}} \oplus C_{\alpha_{i+2}} \oplus \cdots \oplus C_{\alpha_\lambda} . \text{ QED}$$

Let $E_i = C_{\alpha_{i+1}} \oplus C_{\alpha_{i+2}} \oplus \cdots \oplus C_{\alpha_\lambda}$. Then we have Lemma 2.

Lemma 2.

$$L(P) \geq L(P_1 \oplus E_i) \quad (0 \leq i \leq \lambda - 1).$$

Proof.

$$L(P) = L(P[s,v]) + L(P[v,t])$$
$$\geq L(T[s,v]) + L(P[v,t])$$
$$\geq L(T[s,v] \oplus P[v,t])$$
$$= L(P_1 \oplus E_i) .$$

QED

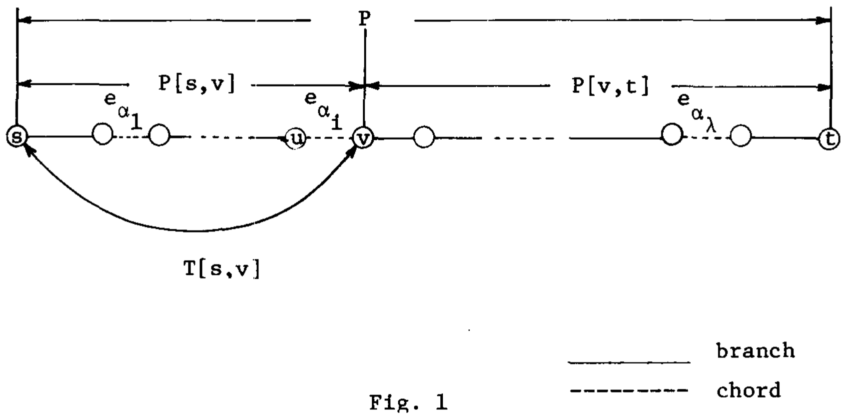

Fig. 1

Theorem

Let P' be a path such that $L(P') < L(P)$. Then there exists ℓ such that $Q_\ell = P'$, $(\ell < k)$.

Proof. Let P' has the chords e_{β_1} , e_{β_2} , ... , e_{β_μ} , in this order, between s and t and let $E'_j = C_{\beta_{j+1}} \oplus C_{\beta_{j+2}} \oplus \cdots \oplus C_{\beta_\mu}$.

Then, by virtue of Lemma 2, we have

$$(4.1) \qquad L(P) > L(P') \geq L(P_1 \oplus E'_j) .$$

If for some j, $(1 \leq j \leq \mu-1)$, there exists k', $(k' < k)$, such that $Q_{k'} = P_1 \oplus E'_j$, since immediately after R_k, the algorithm generates $R_{k'+1}$, $Q_{k'} \oplus C_{\beta_j} = P_1 \oplus E'_{j-1}$ must be contained in $R_{k'+1}$ as long as it has not already been picked out. Then by (4.1) there must exist k", $(k'' < k)$, such that $Q_{k''} = P_1 \oplus E'_{j-1}$ and hence there exists ℓ $(\ell<k)$ such that $Q_\ell = P_1 \oplus E'_0 = P'$.

Since $P_1 \oplus E'_{\mu-1} = P_1 \oplus C_{\beta_\mu} \in R_2$, there exists k', $(k'<k)$, such that $Q_{k'} = P_1 \oplus E'_{\mu-1}$ and hence we have the theorem. QED

5. Efficiency for A Complete Graph

For a complete graph

$$m = n(n-1)/2$$
$$f = m - n + 1 = (n-1)(n-2)/2 \quad .$$

In step 1, $\{Q_k + C_i\}$ requires at most nf additions. Suppose that (K + K') subgraphs are needed to get the K-th best path, where K' is the number of times when the subgraph picked out in step 2 is not a path. Since the step 1 increases the subgraphs by at most f at a time, to get the K-th best path we need approximately

$$(K + K')nf = (K + K')n(n-1)(n-2)/2 \text{ additions and}$$
$$f + 2f + \ldots + (K + K' - 1)f$$
$$= (K + K' - 1)(K + K')(n-1)(n-2)/2 \text{ comparisons.}$$

Though it is difficult to estimate K', in our experiments using the railway network it was observed that $0 \leq K' \leq 20K$.

By J. Yen [6], his algorithm requires approximately $qKn^3/6$ additions and $qKn^3/3$ comparisons, where $0<q\leq 1$. In this point of view, the algorithm proposed here seems to be as good as that of Yen's which is one of the most efficient.

6. Illustrative Example

Consider the graph in Fig. 2, whose fundamental circuits are shown in Fig. 3. We want to find the K-th best path between s(=1) and t(=7). In Fig. 2 and 3, dotted lines represent chords and solid lines represent branches in the minimum tree of s. The results after applying the algorithm in section 3 are summarized in Table 1.

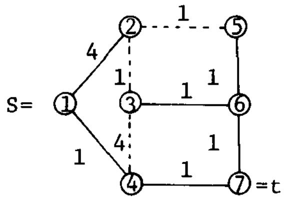

Fig. 2

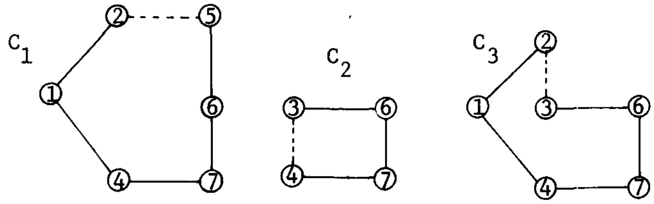

Fig. 3

Table 1

R	Ring Sum	L	Q or P
R_1	P_1	2	$Q_1 = P_1$
R_2	$P_1 \oplus C_1$	7	$Q_2 = P_2$
	$P_1 \oplus C_2$	7	
	$P_1 \oplus C_3$	7	
R_3	$P_1 \oplus C_2$	7	
	$P_1 \oplus C_3$	7	
	$P_2 \oplus C_2$	12	
	$P_2 \oplus C_3$	6	Q_3
R_4	$P_1 \oplus C_2$	7	$Q_4 = P_3$
	$P_1 \oplus C_3$	7	
	$P_2 \oplus C_2$	12	
	$Q_3 \oplus C_2$	9	
R_5	$P_1 \oplus C_3$	7	$Q_5 = P_4$
	$P_2 \oplus C_2$	12	
	$Q_3 \oplus C_2$	9	
	$P_3 \oplus C_3$	10	
R_6	$P_2 \oplus C_2$	12	
	$Q_3 \oplus C_2$	9	$Q_6 = P_5$
	$P_3 \oplus C_3$	10	
R_7	$P_2 \oplus C_2$	12	
	$P_3 \oplus C_3$	10	$Q_7 = P_6$
R_8	$P_2 \oplus C_2$	12	$Q_8 = P_7$

References

1. Bellman, R. and Kalaba, R.: On kth Best Policies. *J. of SIAM*, Vol. 8 (1960) No. 4, pp. 582-588.
2. Hoffman, W. and Pavley, R.: A Method for the Solutions of the Nth Best Path Problem. *JACM*, Vol. 6 (1959), pp. 506-514.
3. Mayeda, W.: Graphy Theory (from Chapter 1 to 3). *Wiley-Interscience*, John Wiley & Sons, Inc., 1972.
4. Pollack, M.: Solution of the kth Best Route Through a Network -- a Review. *J. Math. Anal. Appl.*, Vol. 3 (1961), pp. 547-559.
5. Shier, D.: Iterative Methods for Determining the K Shortest Paths in a Network. *Networks*, Vol. 6(3) (1976), pp. 205-230.
6. Yen, J.: Finding the K Shortest Loopless Paths in a Network. *Man. Sci.*, Vol. 17 (1971), pp. 712-716.

Hiroaki Ishii: Railway Technical Research Institute
Hikaricho 2-8-38, Kokubunji
Tokyo, Japan