A NEW SHORTEST PATH UNDATING ALGORITHM

by

S. Goto and A. Sangiovanni-Vincentelli

Memorandum No. ERL-M588

28 May 1976

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

A New Shortest Path Undating Algorithm[†]

S. Goto[††] and A. Sangiovanni-Vincentelli[†††]

ABSTRACT

A new algorithm for updating shortest paths from all vertices to a set of vertices following a decreasing-length-modification of some arcs, is presented. The algorithm is based on a formula for inverting algebraic analogy with the well-known Householder's formula for inverting modified matrices. The number of operations (i.e., additions and comparisons) required for solving the modified shortest path is estimated as $O(n^2)$, where n is the overall number of vertices.

The algorithm proposed here is particularly powerful for solving the large-scale networks with sparse structure.

[†]Research sponsored by the National Science Foundation Grant ENG72-03783.

[††]Dept. of Electrical Engineering and Computer Science, University of California, Berkeley, California 94720, currently on leave from Nippon Electric Co., Ltd., Japan.

[†††]Instituto di Elettrotecnica ed Elettronica, Politecnico di Milano, 20133 Milano, Italy.

I. Introduction

The shortest path problem is one of the fundamental problems in the area of network programming. In some application it is necessary to compute shortest paths in many networks each different from the other only for some slight changes in costs or for the addition (or deletion) of a few vertices and arcs. While the shortest path problem has been very deeply investigated and many algorithms have been proposed[1~6], a little attention has been devoted to the updating of shortest paths when some changes occur in the original networks. The key in devising an efficient updating algorithm is to take into account as much as possible the results of the original shortest path computation. Spira and Pan[7] gave lower bounds of the computation needed to update shortest paths from one specified vertex to all the other vertices. Hsieh and Kershenbaum proposed an algorithm[8] based on Bellman's method. In this paper, an algorithm for updating shortest paths is proposed. Its structure has been derived by looking at the formal analogy between the shortest path problem and the problem of solving linear algebraic equations introduced by Carrè[4] and Iri-Nakamori[5]. In particular, the algorithm is based on a theorem which can be considered the analogies of the well-known Householder's formula[9] for inverting modified matrices. Its structure is essentially the same of an algorithm for computing the solution of a modified system of linear algebraic equations given in [10]. It assumes that the original shortest paths have been computed by means of a shortest path algorithm analogous to the Crout method[6] for solving the linear algebraic equations. This characteristics allows the effective application to the case of sparse structure. The complexity of computation is very close to the lower bound given in [7]. Its restriction consists in the fact that only decreasing-length-modification can be taken into account. However, some important applications [11] are characterized just by decreasing-length-modifications.

The paper is organized as follows: in Section II some preliminary remarks and definitions are given. In Section III, the main theorem and the modification algorithm are presented and the complexity of computation is evaluated. In Section IV an example is described and in Section V some concluding remarks are given.

II Preliminalies

A <u>diagraph</u> $G = (V,E)$ consists of a set V of n elements, together with a subset E of ordered pairs (u,v) of elements taken from V. The elements of V are called <u>vertices</u> and the members of E are called <u>arcs</u>. Let R be the set of real numbers and $\omega: E \to R$ be a function called <u>cost function</u> which associates a real number called <u>cost</u> or <u>length</u>. A <u>network</u> $\tilde{G} = (V,E,\omega)$ consists of a graph and a cost function ω. We assume that $\forall\, v \in V,\ (v,v) \notin E$, i.e., the network does not admit any <u>selfloops</u>.

A finite order sequence $P = \{v_1,v_2,\ldots,v_\ell\}$ of distinct vertices is called a <u>path</u> from v_1 to v_ℓ if $(v_i,v_{i+1}) \in E$ for $i = 1,2,\ldots,\ell-1$, and

$$\omega(P) = \sum_{i=1}^{\ell-1} \omega(e_i) \ ; \ e_i = (v_i,v_{i+1}) \tag{1}$$

is called the <u>length</u> of P. For an ordered pair (s,t), the set of paths from s to t is denoted by $P(s,t)$. If

$$\forall\, P' \in P(s,t),\ \omega(\bar{P}) \leq \omega(P') \tag{2}$$

the $\bar{P}$ and $\omega(\bar{P})$ are called the <u>shortest path</u> and the <u>distance</u> (i.e., the length of the shortest path) from s to t, respectively.

Let $V = \{v_1,v_2,\ldots,v_n\}$ be the set of vertices of $G = (V,E)$, then the lengths of arcs, are represented by an adjaceny $n \times n$ matrix $A = \{a_{ij}\}$, called <u>measure matrix</u> whose entries are defined as follows:

$$a_{ij} = \begin{cases} \omega(e_{ij}) & ; \ i \neq j \text{ and } e_{ij} = (v_i,v_j) \in E \\ \infty & ; \ i = j \text{ or } (v_i,v_j) \notin E \end{cases} \tag{3}$$

Let x_{ij} be the distance from vertex v_i to v_j for each $i, j = 1,2,\ldots,n$. Then the $n \times n$ matrix $X = \{x_{ij}\}$ is called the distance matrix of the network, where the diagonal elements x_{ii}; $i = 1,2,\ldots,n$, are considered to be 0 in a natural sense.

The problem of determining X or a part of it has been solved in many ways.[1-6] One of these approaches[4][5] is based on an analogy between the shortest path problem and the problem of solving a set of linear algebraic equations. In order to state this analogy Carrè introduced a particular algebra. He considered a semiring $(S, \oplus, \otimes)$, i.e., a set S with two binary operations, $\oplus$: generalized additions and $\otimes$: generalized multiplications, closed on S and obeying the commutative, associative, and distributive laws. In the shortest path problem, S is given by $R \cup \{\infty\}$ and $\oplus$ and $\otimes$ are defined as follows:

$\forall x, y \in S$

$$x \oplus y = \min\{x,y\} \tag{4}$$

$$x \otimes y = x + y \tag{5}$$

The unit element in the algebra is 0 and the null element is ∞. Moreoever, we define a generalized addition and a generalized multiplication of matrices with elements in S as follows. Given two $n \times m$ matrices $X = \{x_{ij}\}$ and $Y = \{y_{ij}\}$, $Z = X \oplus Y$ is the $n \times m$ matrix with elements $z_{ij} = x_{ij} \oplus y_{ij}$. Given an $n \times p$ matrix $X = \{x_{ij}\}$ and a $p \times m$ matrix, $z = X \otimes Y$ is an $n \times m$ matrix with elements $z_{ij} = \sum_{k=1}^{p} x_{ik} \otimes y_{kj}$, where symbol $\sum$ denotes generalized summation.

Since the distance x_{ij} from v_i to v_j satisfies the relation

$$x_{ij} = \begin{cases} \min\limits_{k=1,2,\ldots,n} \{x_{ik} + a_{kj}\} \; ; & i \neq j \\ 0 \; ; & i = j \end{cases} \tag{6}$$

According to Eq. (4), (5) and to the definition of generalized addition and multiplication of matrices, Eq. (6) can be written in a matrix form.

$$X = A \otimes X \oplus I_n \tag{7}$$

where I_n is a square matrix of order n with o's in the main diagonal positions and with ∞'s in the off-diagonal positions. It is considered to be the <u>unit matrix</u> in the shortest path problem.

If the network $\tilde{G}$ does not admit any cycles with negative length, the solution $\bar{X}$ of Eq. (7) can be obtained as

$$\bar{X} = A^* \otimes I_n \tag{8}$$

where A^* is an n x n matrix obtained via the following equation:

$$A^* = I_n \oplus A \oplus A^2 \oplus \ldots . \oplus A^{n-1} \tag{9}$$[1]

Generalizing Eq. (8), in [4] it is shown that any equation of the form

$$Y = A \otimes Y \oplus Z \tag{10}$$

where A is an n x n matrix with elements in S, Z is any n x p matrix with elements in S and $\bar{Y}$ is an n x p unknown matrix, has the solution

$$\bar{Y} = A^* \otimes Z \tag{11}$$

Carrè has shown that the solution of Eq. (8) and (10) can be also obtained by algorithms formally identical to the algorithm used in the solution of a system of linear algebraic equations. In particular, one of the most efficient method is the analogous one to the reduced Crout algorithm.[6] As in linear algebraic equation case,[12] this algorithm is quite efficient when large scale shortest path problems have to be solved, being the measure matrix A sparse. The Crout algorithm for the shortest path problem has the following procedure. Let $L = \{\ell_{ij}\}$ be an n x n lower triangular matrix and $U = \{u_{ij}\}$ an n x n upper triangular matrix with elements

$$\ell_{ii} = 0 \qquad (i = 1,2,\dots,n)$$

$$\ell_{i1} = a_{i1} \qquad (i = 2,3,\dots,n-1)$$

$$u_{1j} = a_{1j} \qquad (j = 2,3,\dots,n-1)$$

$$\ell_{ij} = a_{ij} \oplus \sum_{k=1}^{j-1} \ell_{ik} \otimes u_{kj} \tag{12}$$

$$(i = j+1,\ j+2,\dots,n;\ j=2,3,\dots,n-1)$$

$$u_{ij} = a_{ij} \oplus \sum_{k=1}^{i-1} \ell_{ik} \otimes u_{kj}$$

$$(j = i+1,\ i+2,\dots,n;\ j = 2,\ 3,\dots,n-1)$$

Let $L^{(k)}$ $(k = 2,3,\dots,n)$ be a lower triangular matrix defined as follows:

$$\ell_{ii}^{(k)} = 0 \qquad (i = 1,2,\dots,n)$$

$$\ell_{kj}^{(k)} = \ell_{kj} \qquad (j = 1,2,\dots,k-1) \tag{13}$$

$$\ell_{ij}^{(k)} = \infty \qquad (i = k,\ j = k+1,\dots,n \text{ or } i \neq k)$$

and $\cup^{(k)}$ $(k = 1,2,\ldots,n-1)$ a upper triangular matrix defined as follows:

$$\begin{aligned} u_{ii}^{(k)} &= 0 && (i = 1,2,\ldots,n) \\ u_{kj}^{(k)} &= u_{kj} && (j = k+1, k+2,\ldots,n) \\ u_{ij}^{(k)} &= \infty && (i = k, j = 1,\ldots,k-1 \text{ or } i \neq k) \end{aligned} \tag{14}$$

Then,

$$\bar{X} = \prod_{k=1}^{n-1} \cup^{(k)} \otimes \prod_{k=1}^{n-1} L^{(n-k+1)} \otimes I_n \tag{15}$$

or

$$\bar{Y} = \prod_{k=1}^{n-1} \cup^{(k)} \otimes \prod_{k=1}^{n-1} L^{(n-k+1)} \otimes Z \tag{16}$$

where $\prod_{k=1}^{n-1}$ is intended as a generalized product of $n - 1$ matrices.

Let $\bar{\underset{\sim}{x}}^{(h)}$ and $\underset{\sim}{b}^{(h)}$ $(h = 1,2,\ldots,n)$ be the h-th column of $\bar{X}$ and I_n, respectively. Then, by Eq. (15), each column of $\bar{X}$ is given by

$$\bar{\underset{\sim}{x}}^{(h)} = \prod_{k=1}^{n-1} \cup^{(k)} \otimes \underset{\sim}{t}^{(h)} \tag{17}$$

where

$$\underset{\sim}{t}^{(h)} = \prod_{k=1}^{n-1} L^{(n-k+1)} \otimes \underset{\sim}{b}^{(h)} \tag{18}$$

Let $\bar{\underset{\sim}{y}}^{(P)}$ and $\underset{\sim}{z}^{(P)}$ $(P = 1,2,\ldots,m)$ be the p-th column of $\bar{Y}$ and Z, respectively. Then, by Eq. (16), each column of $\bar{Y}$ is given by

$$\bar{\underset{\sim}{y}}^{(P)} = \prod_{k=1}^{n-1} \cup^{(k)} \otimes \underset{\sim}{t}^{(P)} \tag{19}$$

where

$$\underset{\sim}{t}^{(P)} = \prod_{k=1}^{n-1} L^{(n-k+1)} \otimes \underset{\sim}{z}^{(P)} \tag{20}$$

In analogy with the linear algebraic equations, the procedures for L and U through Eq. (12) may be called <u>Triangular Factorization</u>, the procedures for evaluating $\underset{\sim}{t}^{(h)}$ in Eq. (18) and $\underset{\sim}{t}^{(P)}$ in Eq. (20) <u>Forward Substitution</u> and the procedures for $\underset{\sim}{\bar{x}}^{(h)}$ in Eq. (17) and $\underset{\sim}{\bar{y}}^{(P)}$ in Eq. (19) <u>Backward Substitution</u>.

The expression Eq. (15) can be viewed as a shortest path version of the <u>eliminate form of inverse</u>.[12]

The complexity of the algorithm has to be evaluated by taking into account of both generalized additions (comparisons) and generalized multiplications (additions), since for most of the available computers both of them require the same amount of computation time. When A is a full matrix, i.e., when all the off-diagonal elements are non-infinity, the number of operations (comparisons and additions) required to evaluate $\bar{x}^{(h)}$ in Eq. (17) is estimated by $\frac{n^3}{3}$ in Triangular Factorization which is performed once, $(n-h)^2/2$ in Forward Substitution for each h and $n^2/2$ in Backward Substitution for each h; this results in the total value

$$\frac{n^3}{3} + \frac{n^2}{2}q + \frac{q^3}{6}$$

where q is the number of the specified columns of $\bar{X}$ to be evaluated.[2] Particularly, we need n^3 operations for q = n (all shortest path problem).

The Crout type algorithm mentioned above enables us to estimate the number of operations in terms of network structure (independent of numerical values of arc-lengths). Furthermore, it should be noted that when A is sparse, computation time can be saved to a great amount if the program is implemented so as to execute only non-trivial operations.[3]

III. A modification algorithm

In some applications, a shortest path problem has to be solved with very slight modifications of the involved network $\tilde{G}$. A modification may occur when some vertices or arcs are deleted or added or when some arc-lengths are decreased or increased.

In this case, it is expected that less computation should be required to evaluate shortest paths in the modified networks by using the previous shortest paths. Recently, Spira and Pan[7] showed the lower bound for updating shortest paths on particular cases. While efficient algorithms are well-known and often used for inverting modified matrices[9] or solving a system of linear algebraic equations with modified coefficient.[10,13,14]

In this section, the analogy between shortest path problem and linear algebraic equation problem presented in Section II is explored in order to devise a new updating algorithm.

In the linear algebraic case, if B is the modified coefficient matrix, it is always possible to decompose B as

$$B = A + C \tag{21}$$

where C is the modification matrix. Moreover if C is a rank m ($m < n$) matrix, it is possible to decompose C as

$$C = H\,K^T \tag{22}$$

where H and K are (n x m) matrices and superscript T denotes the transposition of a matrix. Then,

$$B = A + H\,K^T \tag{23}$$

and Eq. (23) plays an important role in the development of modification algorithms. In order to apply the analogy, we have to represent the modification to the measure matrix A in the same way.

Let B be the measure matrix of the modified network, B can be decomposed as in Eq. (21) if and only if

$$\forall_{i,j}\ ,\ b_{ij} \leq a_{ij} \tag{24}$$

since the generalized addition in the shortest path problem is a min operation. Then, it is possible to write

$$B = A \oplus C \tag{25}$$

where C is defined as follows:

$$C_{ij} = \begin{cases} b_{ij} & \text{if } b_{ij} < a_{ij} \\ \infty & \text{if } b_{ij} = a_{ij} \end{cases} \tag{26}$$

Then, we have to assume that only decreasing-length-modifications are performed on a given network if we want to apply the analogy with the linear algebraic equation case.[4] We can pick up some of the columns and rows which together contain all the finite elements of C. Let $a_1, a_2, \ldots, a_\alpha$ and $b_1, b_2, \ldots, b_\beta$ be the column and row numbers which are taken, respectively. Let m[5] be the sum of α and β and $H = \{h_{ij}\}$ and $K = \{k_{ij}\}$ be two (n x m) matrices defined as follows:

for $1 \leq j \leq \alpha$

$$h_{ij} = C_{ia_j} \tag{27}$$

$$k_{ij} = \begin{cases} 0 \; ; \; i = a_j \\ \infty \; ; \; i \neq a_j \end{cases} \tag{28}$$

$$(i = 1,2,\ldots,n)$$

for $\alpha + 1 \leq j \leq m$

$$h_{ij} = \begin{cases} 0 \; ; \; i = b_{j-\alpha} \\ \infty \; ; \; i \neq b_{j-\alpha} \end{cases} \tag{29}$$

$$k_{ij} = C_{b_{j-\alpha} i} \tag{30}$$

$$(i = 1,2,\ldots,n)$$

where, $m = \alpha + \beta$

According to Eq. (27) ~ (30), H and K can be decomposed as

$$H = [C_{n\alpha} \vdots I_{n\beta}]$$

$$K = [I_{n\alpha} \vdots C_{n\beta}]$$

where

$$C_{n\alpha} = \underbrace{\begin{bmatrix} C_{1a_1} & C_{1a_2} & \cdot\cdot & C_{1a_\alpha} \\ C_{2a_1} & C_{2a_2} & \cdot\cdot & C_{2_\alpha} \\ \vdots & \vdots & & \vdots \\ C_{na_1} & C_{na_2} & \cdot\cdot & C_{na_\alpha} \end{bmatrix}}_{\alpha} \Big\} n \tag{31}$$

$$
C_{n\beta} = \underbrace{\begin{bmatrix} C_{b_1 1} & C_{b_2 1} & \cdots & C_{b_\beta 1} \\ C_{b_1 2} & C_{b_2 2} & \cdots & C_{b_\beta 2} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ C_{b_1 n} & C_{b\,n} & & C_{b_\beta n} \end{bmatrix}}_{\beta} \left.\vphantom{\begin{matrix} 1\\1\\1\\1\\1\\1 \end{matrix}}\right\} n \tag{32}
$$

$$
I_{n\alpha} = \underbrace{\begin{bmatrix} \infty & \infty & \infty \\ \cdot & \cdot & \cdot \\ 0 & \infty & \cdot \\ \infty & 0 & \infty \\ \cdot & \infty & \cdot \\ \infty & \infty & 0 \\ \infty & \infty & \infty \end{bmatrix}}_{\alpha} \begin{matrix} \\ \\ a_1 \\ a_2 \\ \cdot \\ a_\alpha \\ \\ \end{matrix} \left.\vphantom{\begin{matrix} 1\\1\\1\\1\\1\\1\\1 \end{matrix}}\right\} n \tag{33}
$$

$$
I_{n\beta} = \underbrace{\begin{bmatrix} \infty & \infty & \infty \\ \cdot & \cdot & \cdot \\ \infty & \cdot & \cdot \\ 0 & \infty & \cdot \\ \infty & 0 & \infty \\ \cdot & \infty & \\ \infty & \infty & 0 \\ \infty & \infty & \infty \end{bmatrix}}_{\beta} \begin{matrix} \\ \\ \\ b_1 \\ b_2 \\ \cdot \\ b_\beta \\ \\ \end{matrix} \left.\vphantom{\begin{matrix} 1\\1\\1\\1\\1\\1\\1\\1 \end{matrix}}\right\} n \tag{34}
$$

Therefore, the following Lemma holds.

Lemma 1.

The matrix C can be decomposed as

$$
C = H \otimes K^T \tag{35}
$$

where, superscript T denotes the transposition of a matrix.

⟨Proof⟩

$$(H \otimes K^T)ij = \sum_{\ell=1}^{n} (h_{i\ell} \otimes k_{\ell j})$$

$$= \sum_{\ell=1}^{\alpha} (h_{i\ell} \otimes k_{j\ell}) \oplus \sum_{\ell=\alpha+1}^{m} (h_{i\ell} \otimes k_{j\ell})$$

$$= \sum_{\ell=1}^{\alpha} (C_{ia_\ell} \otimes k_{j\ell}) \oplus \sum_{\ell=\alpha+1}^{m} (h_{i\ell} \otimes C_{b_{\ell-\alpha}j})$$

$$= C_{ij} \oplus C_{ij}$$

$$= C_{ij}$$

Q.E.D.

Then, the equation of the shortest path for the modified networks can be written as

$$\hat{X} = (A \oplus H \otimes K^T) \otimes \hat{X} \oplus I_n \tag{36}$$

Now, we can relate A* to $(A \oplus H \otimes K^T)*$ with a formula which is the exact analogous to the well-known Householder's formula.[6]

Theorem 1

$$(A \oplus H \otimes K^T)^* = A^* \oplus A^*H(K^TA^*H)^* K^TA^* \tag{37}$$

⟨Proof⟩

According to Eq. (8), the solution of Eq. (36), $\bar{\hat{X}}$, is equal to the left hand side of Eq. (37). Then, in order to state that Eq. (37) holds, it is sufficient to prove that the right hand side of Eq. (37) is a solution for Eq. (36).

$$(A \oplus H K^T) \otimes \bar{\hat{X}} \oplus I_n$$

$$= (A \oplus H K^T) (A^* \oplus A^* H(K^T A^* H)^* K^T A^*) \oplus I_n$$

$$= A A^* \oplus I_n$$

$$\oplus A A^* H(K^T A^* H) K^T A^*$$

$$\oplus H K^T A^* \oplus H K^T A^* H(K^T A^* H)^* K^T A^*$$

$$= A^* \oplus A A^* H(K^T A^* H) K^T A^*$$

$$\oplus H(I_n \oplus K^T A^* H(K^T A^* H)^*) K^T A^*$$

$$= A^* \oplus A A^* H(K^T A^* H) K^T A^* \oplus H(K^T A^* H)^* K^T A^*$$

$$= A^* \oplus (A A^* \oplus I_n) H(K^T A^* H)^* K^T A^*$$

$$= A^* \oplus A^* H(K^T A^* H)^* K^T A^*$$

$$= \bar{X}$$

Q.E.D.

Theorem 1 suggests an efficient algorithm to compute the solution $\bar{\hat{X}}$ of Eq. (7), i.e., $\bar{X}$ or A^*. We assume that A^* has been computed by means of Triangular Factorization, Forward and Backward Substitutions.

Modification algorithm -----MOD

STEP 1: Compute the solution $\bar{X}'$ of $X' = A \otimes X' \oplus H$ by evaluating $A^* \otimes H$.

STEP 2: Compute $K^T \otimes \bar{X}$

STEP 3: Compute $K^T \otimes \bar{X}'$

STEP 4: Perform Triangular Factorization of the matrix $K^T \otimes \bar{X}'$

STEP 5: Compute the solution $\bar{W}$ by Forward and Backward Substitutions for $W = (K^T \otimes \bar{X}') \otimes W \oplus (K^T \otimes \bar{X})$

STEP 6: Compute $\bar{\hat{X}} = \bar{X} \oplus (\bar{X}' \otimes \bar{W})$

Theorem 2

$\bar{\hat{X}}$ given by MOD is the solution of Eq. (36).

⟨Proof⟩

The solution $\bar{X}'$ of $X' = A \otimes X' \oplus H$ is represented by $A^* \otimes H$. The solution $\bar{W}$ of $W = (K^T \otimes X') \otimes W \oplus K^T \otimes \bar{X}$ is given by $(K^T \otimes X')^* \otimes K^T \otimes \bar{X}$. Substituting the solutions $\bar{X}$, $\bar{X}'$ and $\bar{W}$ into $\bar{X} \oplus \bar{X}' \otimes \bar{W}$, we have $A^* \oplus A^*H(K^TA^*H)^*K^TA^*$. Theorem 1 guarantees that this expression is the solution of Eq. (36).

Q.E.D.

The complexity of MOD is now evaluated. When A is a full matrix, by taking account of the particular structure of $H = [C_{n\alpha} \vdots I_{n\beta}]$, STEP 1 requires

$$\alpha\, n^2 \tag{38}$$

non-trivial additions and comparisons and STEP 2

$$\beta \ n^2 \tag{39}$$

non-trivial additions and comparisons by considering the particular structure of $K = [I_{n\alpha} \vdots C_{n\beta}]$

STEP 3 requires obviously

$$\beta \ m \ n \tag{40}$$

non-trivial additions and comparisons and STEP 4 and 5 requires

$$1/3 \ m^3 \tag{41}$$

$$m^2 n \tag{42}$$

non-trivial operations, respectively and STEP 6 requires

$$m \ n^2 \tag{43}$$

additions and

$$(m+1)n^2 \tag{44}$$

comparisons.

If not all the modified shortest paths are required, but only the shortest paths from all the vertices to some specified vertices (whose number is denoted by q)[7], then only q columns corresponding to those specified vertices have to be computed. Then, STEP 2 requires

$$q \ \beta \ n \tag{45}$$

non-trivial additions and comparisons by performing the operations for only q specified columns of $\bar{X}$ but not for all columns. STEP 5 requires

$$q\ m^2 \tag{46}$$

non-trivial additons and comparisons and STEP 6 requires

$$m\ q\ n \tag{47}$$

additions and

$$(m+1)q\ n \tag{48}$$

comparisons.

Then, the total number of non-trivial operations required to update the shortest paths from each vertex to q specified vertices is estimated as

$$\text{additions:}\quad \alpha\ n^2 + ((q+\beta)m + q\beta)n + qm^2 + 1/3\ m^3 \tag{49}$$

$$\simeq \alpha\ n^2 + ((q+\beta)m + q\beta)n \quad (n >> q,m) \tag{50}$$

$$\text{comparisons:}\ \alpha\ n^2 + ((q+\beta)m + q(\beta+1))n+qm^2 + 1/3\ m^3 \tag{51}$$

$$\simeq \alpha\ n^2 + ((q+\beta)m + q(\beta+1))n \quad (n >> q,m) \tag{52}$$

Note

[1] Here, we assume that the solution $\bar{X}(A^*)$ of Eq. (7) is given. If Triangular Factorization of A (i.e., $L = \{\ell_{ij}\}$ and $U = \{U_{ij}\}$) is assumed to be given instead of A^*, $m\ n^2$ non-trivial additions and comparisons are required instead of $\alpha\ n^2$ to carry out Forward and Backward Substitutions. This assumption is considered to be a reasonable

one in the case of updating some specified shortest paths but not all shortest paths.

[2] In almost all cases, A is very sparse. In this case, STEP 1 can be more conveniently carried out. Let γ_i and ζ_i be the number of non infinity elements in the i-th row of L and the i-th column of U except for the main diagonal, to carry out Forward and Backward Substitutions in STEP 1,

$$m \sum_{i=1}^{n-1} (\gamma_i + \zeta_i) \tag{53}$$

non-trivial additions and comparisons are required.

Spira and Pan showed that if a new vertex is added at least 1/2(n-1)(n-2) comparisons are required to updata shortest paths from one specified vertex to all the vertices. According to the Modification algorithm proposed here, we need exactly $(n^2 + 6n + 4)$ comparisons if A^* is given and $(2n^2 + 6n + 4)$ comparisons if Triangular Factorization of A is given. $(q = 1, \alpha = \beta = 1, m = 2)$. Anyhow, if shortest paths from q specified vertices to all the vertices have to be obtained in modified networks, MOD requires a number of operations whose leading term (n^2) is independent of q but only depends on modified elements A, i.e., α or m.

Updating algorithms[7,8] based on a Dijkstra's procedure or Bellman's procedure may require q as the coefficient of the leading term (n^2). Therefore, it requires $0(n^3)$ operations for updating all shortest paths, however, MOD requires $0(n^2)$ operations.

The main features of MOD are then:

a) the capability of exploiting sparsity

b) its complexity measure very close to the lower bound computed in [7] and independent of the number of specified vertices in the leading term.

VI. EXAMPLE

We shall show an example of updating shortest path based on Modification Algorithm.

In the network shown in Fig. 1, the measure matrix is given by

$$A = \begin{bmatrix} \infty & 9 & 2 & 5 \\ 8 & \infty & 7 & \infty \\ 8 & 6 & \infty & 12 \\ 4 & \infty & \infty & \infty \end{bmatrix} \tag{54}$$

By Triangular Factorization, we have the following two matrices $L = \{\ell_{ij}\}$ and $U = \{u_{ij}\}$, according to the procedure in Eq. (12).

$$L = \begin{bmatrix} 0 & & \infty & \\ 8 & 0 & & \\ 8 & 6 & 0 & \\ 4 & 13 & 6 & 0 \end{bmatrix} \tag{55}$$

$$U = \begin{bmatrix} 0 & 9 & 2 & 5 \\ & 0 & 7 & 13 \\ & \infty & 0 & 12 \\ & & & 0 \end{bmatrix} \tag{56}$$

The distance matrix calculated by performing Forward and Backward Substitions in the following way results in

$$
X = \begin{bmatrix} 0 & 9 & 2 & 5 \\ & 0 & \infty & \infty \\ \infty & & 0 & \infty \\ & & & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & \infty & \infty & \infty \\ & 0 & 7 & 13 \\ \infty & & 0 & \infty \\ & & & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & \infty & \infty & \infty \\ & 0 & \infty & \infty \\ \infty & & 0 & 12 \\ & & & 0 \end{bmatrix}
$$

$$
\otimes \begin{bmatrix} 0 & & & \\ \infty & 0 & \infty & \\ \infty & \infty & 0 & \\ 4 & 13 & 6 & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & & & \\ \infty & 0 & \infty & \\ 8 & 6 & 0 & \\ \infty & \infty & \infty & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & & & \\ 8 & 0 & \infty & \\ \infty & \infty & 0 & \\ \infty & \infty & \infty & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & & & \\ & 0 & \infty & \\ & \infty & 0 & \\ & & & 0 \end{bmatrix} \tag{57}
$$

Thus, we have

$$
\bar{X} = \begin{bmatrix} 0 & 8 & 2 & 5 \\ 8 & 0 & 7 & 13 \\ 8 & 6 & 0 & 12 \\ 4 & 12 & 6 & 0 \end{bmatrix} \tag{58}
$$

Let the lengths of arcs be changed, as shown in Fig. 2. The new measure matrix is given by

$$
B = \begin{bmatrix} \infty & 3 & 2 & 5 \\ 2 & \infty & 7 & \infty \\ 3 & 4 & \infty & 5 \\ 1 & \infty & \infty & \infty \end{bmatrix} \tag{59}
$$

and may be written in the form:

$$
B = A \oplus C \tag{60}
$$

where,

$$
C = \begin{bmatrix} \infty & 3 & \infty & \infty \\ 2 & \infty & \infty & \infty \\ 3 & 4 & \infty & 5 \\ 1 & \infty & \infty & \infty \end{bmatrix} \tag{61}
$$

Matrix C is decomposed into:

$$C = H \otimes K^T \tag{62}$$

where,

$$H = \begin{bmatrix} \infty & 3 & \infty \\ 2 & \infty & \infty \\ 3 & 4 & 0 \\ 1 & \infty & \infty \end{bmatrix} \tag{63}$$

$$K = \begin{bmatrix} 0 & \infty & 3 \\ \infty & 0 & 4 \\ \infty & \infty & \infty \\ \infty & \infty & 5 \end{bmatrix} \tag{64}$$

The steps of Modification Algorithm are performed in the following way:

STEP 1:

$$\bar{X}' = \begin{bmatrix} 0 & 9 & 2 & 5 \\ & 0 & \infty & \infty \\ \infty & & 0 & \infty \\ & & & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & \infty & \infty & \infty \\ & 0 & 7 & 13 \\ \infty & & 0 & \infty \\ & & & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & \infty & \infty & \infty \\ & 0 & \infty & \infty \\ \infty & & 0 & 12 \\ & & & 0 \end{bmatrix}$$

$$\otimes \begin{bmatrix} 0 & & & \\ \infty & 0 & \infty & \\ \infty & \infty & 0 & \\ 4 & 13 & 6 & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & & & \\ \infty & 0 & \infty & \\ 8 & 6 & 0 & \\ \infty & \infty & \infty & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & & & \\ 8 & 0 & \infty & \\ \infty & \infty & 0 & \\ \infty & \infty & \infty & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & 3 & \infty \\ 2 & 0 & \infty \\ 3 & 4 & 0 \\ 1 & \infty & \infty \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 3 & 2 \\ 2 & 0 & 7 \\ 3 & 4 & 0 \\ 1 & 7 & 6 \end{bmatrix} \tag{65}$$

STEP 2:

$$K^T \otimes \bar{X} = \begin{bmatrix} 0 & \infty & \infty & \infty \\ \infty & 0 & \infty & \infty \\ 3 & 4 & 0 & 5 \end{bmatrix} \otimes \begin{bmatrix} 0 & 8 & 2 & 5 \\ 8 & 0 & 7 & 13 \\ 8 & 6 & 0 & 12 \\ 4 & 12 & 6 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 8 & 2 & 5 \\ 8 & 0 & 7 & 13 \\ 3 & 4 & 0 & 5 \end{bmatrix} \tag{66}$$

STEP 3:

$$K^T \otimes \bar{X}' = \begin{bmatrix} 0 & \infty & \infty & \infty \\ \infty & 0 & \infty & \infty \\ 3 & 4 & 0 & 5 \end{bmatrix} \otimes \begin{bmatrix} 0 & 3 & 2 \\ 2 & 0 & 7 \\ 3 & 4 & 0 \\ 1 & 7 & 6 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 3 & 2 \\ 2 & 0 & 7 \\ 3 & 4 & 0 \end{bmatrix} \tag{67}$$

STEP 4:

Triangular Factorization of $K^T \otimes \bar{X}'$ is shown by

$$\begin{bmatrix} 0 & 3 & 2 \\ 2 & 0 & 4 \\ 3 & 4 & 0 \end{bmatrix}$$

STEP 5:

$$W = \begin{bmatrix} 0 & 3 & 2 \\ & 0 & \infty \\ \infty & & \\ & & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & \infty & \infty \\ & 0 & 4 \\ \infty & & \\ & & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & & \\ & & \infty \\ \infty & 0 & \\ 3 & 4 & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & 8 & 2 & 5 \\ 8 & 0 & 7 & 13 \\ 3 & 4 & 0 & 5 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 3 & 2 & 5 \\ 2 & 0 & 4 & 7 \\ 3 & 4 & 0 & 5 \end{bmatrix}$$

STEP 6:

$$\hat{\bar{X}} = \begin{bmatrix} 0 & 8 & 2 & 5 \\ 8 & 0 & 7 & 13 \\ 8 & 6 & 0 & 12 \\ 4 & 12 & 6 & 0 \end{bmatrix} \oplus \begin{bmatrix} 0 & 3 & 2 \\ 2 & 0 & 7 \\ 3 & 4 & 0 \\ 1 & 7 & 6 \end{bmatrix} \otimes \begin{bmatrix} 0 & 3 & 2 & 5 \\ 2 & 0 & 4 & 7 \\ 3 & 4 & 0 & 5 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 3 & 2 & 5 \\ 2 & 0 & 4 & 7 \\ 3 & 4 & 0 & 5 \\ 1 & 4 & 3 & 0 \end{bmatrix} \tag{69}$$

V. CONCLUDING REMARKS

The problem of updating shortest paths from all vertices to a set of vertices following decreasing-length-modification has been discussed. In order to solve the problem effectively, the analogy between shortest path problem and the problem of finding the solution of a system of linear algebraic equations has been exploited. The analogy has been used to derive a formula which bears formal analogy with the well-known Householder's formula for inverting modified matrices.

An efficient algorithm based on this formula has been proposed and its complexity of computation is evaluated. The complexity has been estimated as $0(n^2)$ and is very close to the lower bound computed in [7]. Furthermore, the complexity in its leading term is independent of the number of specified vertices (number of end vertices to get the shortest paths), though other algorithms based on Dijkstra and Bellman method depent on it. Moreover, since its structure is mainly based on the reduced Crout algorithm for shortest path problems, it can be very efficient when the given network is sparse.

The problem of updating the shortest paths for the cases when some of arc -lengths could have increased, is not dealt with in this paper. Further investigation is required to cover the remaining alternatives.

As a final remark, it has to be noted that efficient decomposition algorithms analogous to the tearing algorithms developed in the linear algebraic system problem can be devised by exploiting the modification algorithms proposed in this paper. The decomposition algorithms will be thoroughly discussed in a forthcoming paper.

ACKNOWLEDGMENTS

The authors are greatly indebted with I. Cederbaum for his valuable comments on an earlier draft of this paper, with I. Frisch, A. Kershenbaum and W. Hsieh for their pointing out this problem and with E. S. Kuh and R. Karp for their helpful suggestions.

REFERENCES

1. S. E. Dreyfus: An appraisal of some shortest-path algorithms, Oper. Res., 19, 3, p. 395 (1969-5).

2. R. E. Bellman: On a routing problem, Quart. Appl. Math., 16, pp. 87-90 (1958).

3. E. W. Dijkstra: A note on two problems in connection with graphs, Numerishe Mathematik, 1, P. 269 (1959).

4. B. A. Carre: An algebra for network routing problems, J. Inst. Maths. Applics, Vol. 7, pp. 273-294 (1971).

5. M. Iri and M. Nakmori: Path-sets, operator semi-groups and shortest path algorithms on a network, RAAG Research Notes, Third Series, No. 185 (1972-10).

6. S. Goto, T. Ohtsuki and T. Yoshimura: A shortest path calculation program based on code generation technique, the 13th Annual Allterton Conference on Circuit and System Theory (1975-10). Also submitted in IEEE Trans. on CAS.

7. P. M. Spira and A. Pan: On finding and updating spanning trees and shortest paths, SIAM J. Comput. Vol. 4, No. 3 (1975-9).

8. W. Hsieh and A. Kershenbaum: A diverse shortest path procedure, Symposium on new directions and recent results in algorithms and complexity, (1976-4).

9. A. S. Householder: Principles of numerical analysis, p. 79, New York, McGraw-Hill Book Co. (1953).

10. G. Guardabassi and A. Sangiovanni-Vincentelli: A two levels algorithm for tearing, Memo ERL-M569, University of California, Berkeley, (1975-12).
Also to appear in IEEE Trans. on CAS, (1976-9).

11. W. Hsieh and A. Kershenbaum: Shortest path tree updating procedures, Working memorandum LL3-9, Network Analysis Corporation (1975-11).

12. R. P. Tewarson: Sparse Matrices, Academic Press, New York (1973).

13. J. M. Bennet: Triangular factors of modified matrices, Numer. Math., Vol. 7, pp. 217-221 (1965).

14. T. Fujisawa, E. S. Kuh and T. Ohsuki: A sparse matrix method for analysis of piece-wise-linear resistive networks, IEEE Trans. on Circuit Theory, Vol. CT-19, Nov. 1972.

15. J. E. Hopcroft and R. M. Karp: An $n^{\frac{5}{2}}$ algorithm for maximum matchings in bipartite graphs, SIAM J. Comput., Vol. 2, No. 4, (1973-12).

FOOTNOTES

1. A^i is defined as $\underbrace{A \otimes A \otimes \cdots \otimes A}_{i}$

2. Here, the first q columns of X are assumed to be evaluated.

3. The term 'non-trivial operations' here refer to operations of the form $a \oplus b$ or $a \otimes b$ with $a, b \neq 0$ or ∞.

4. When some new arcs or vertices are added to network $\tilde{G}$, the analogy can be applied. Since, the corresponding entries in B are finite while those were ∞'s in A.

5. The number m depends on what column numbers or row numbers are chosen to contain all finite elements of C. Finding the minimum number of m is reduced to calculating the term rank of the matrix, where changed elements are represented as "1" and unchanged ones are as "0". Hopcroft and Karp proposed an $O(n^{\frac{5}{2}})$ algorithm to find the term rank.[15]

6. The Householder's formula[9] in inverting modified matrices can be written as

$$(A + HK^T)^I = A^I - A^I H(I_n + K^T A^I H)^I K^T A^I$$

where, the superscript I indicates inverse.

7. In order to evaluate the shortest paths from some specified vertices to all vertices, consider the transpased matrix of A and apply the same technique.

CAPTIONS

Fig. 1 Network

(i) : vertex

(i) : weight of arc

Fig. 2. Network with decreasing-length-modifications

(i) : vertex

(i) : length of arc

* denotes the arc-length which is different from one in Fig. 1.

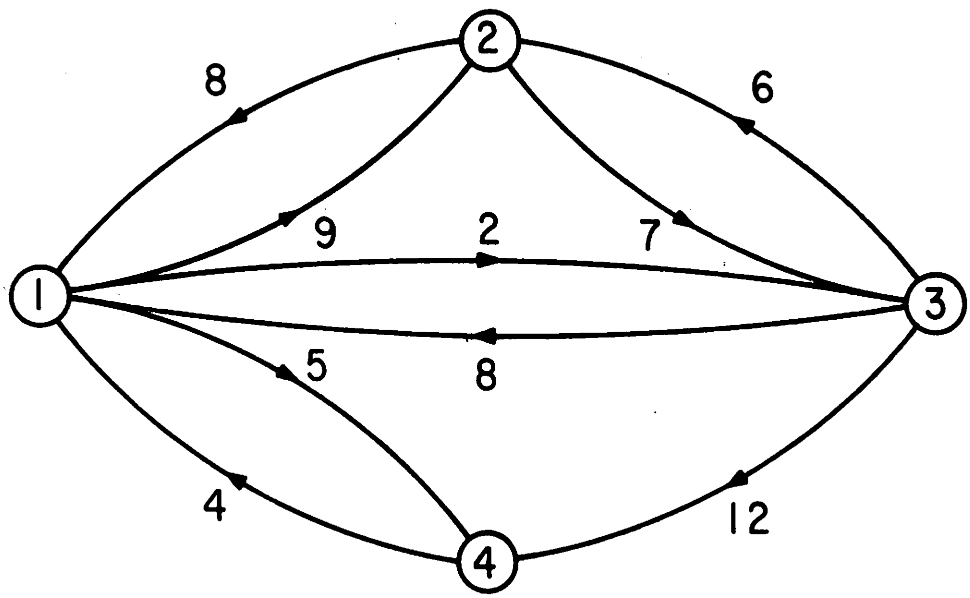

Fig. 1

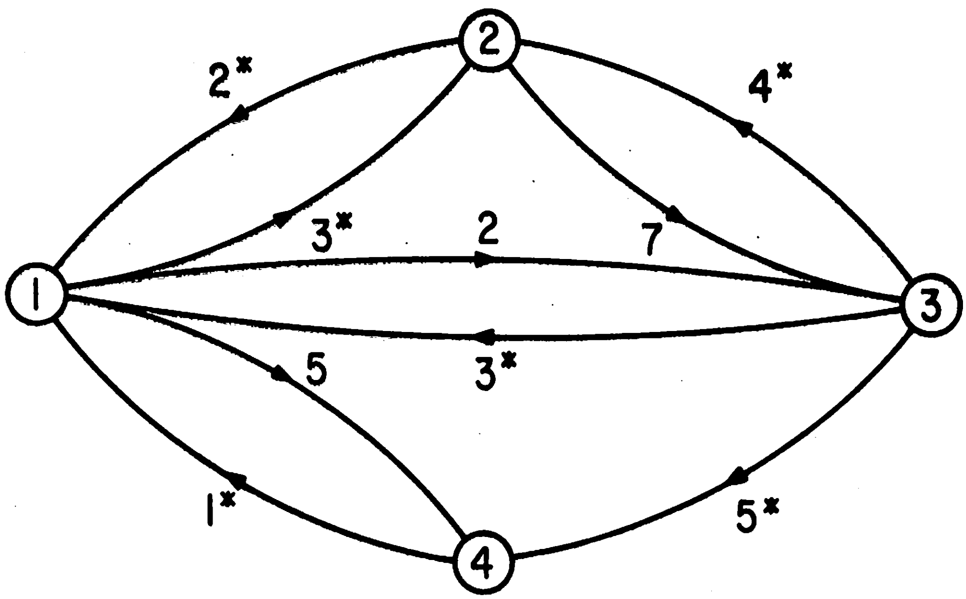

Fig. 2