T. C. Hu
W. T. Torres

Shortcut in the Decomposition Algorithm for Shortest Paths in a Network*

Abstract: The problem considered is that of finding the shortest path between the two nodes of every pair in a large n-node network. A decomposition algorithm is proposed for use when the number of arcs is less than $n(n-1)$. The network is first decomposed into several overlapping subnetworks. Next, with each subnetwork treated separately, conditional shortest paths are obtained using triple operations. Finally, these conditional shortest paths are used to obtain the shortest paths between paired nodes in the original network by matrix mini-summation. This decomposition algorithm requires less computer storage and fewer arithmetic operations than other known algorithms.

Introduction

There are several shortest-path algorithms, all of which require the same number of arithmetic operations, e.g., those of Dantzig,[1] Floyd[2] and Murchland.[3] These algorithms treat the whole n-node network as a unit and require $n(n-1)(n-2)$ additions and the same number of comparisons. In practice, most networks have far fewer than $n(n-1)$ arcs (node connectors). In such cases decomposition can be used to reduce the amount of computation as well as the computer storage requirement. The first practical decomposition algorithm founded on matrix methods for finding shortest paths, due to Land and Stairs,[4] was based on the "cascade" algorithm and requires roughly twice the number of arithmetic operations in the decomposition algorithm of Hu.[5] The idea of the present algorithm is essentially that of Ref. 5, but it involves still fewer arithmetic operations.

We consider a network consisting of nodes N_j ($j = 1, \cdots, n$) with arcs leading from N_i to N_j. Each arc has associated with it a distance (or length) d_{ij}. These distances do not have to satisfy the triangular inequality, $d_{ij} + d_{jk} \geq d_{ik}$, and they need not be symmetric, i.e., $d_{ij} \neq d_{ji}$ in general. Furthermore, these distances can be negative provided that the sum of the arc lengths in any cycle (closed path) remains non-negative. If there is no arc leading from N_i to N_j, we define $d_{ij} = \infty$. Also, we define $d_{ii} = 0$ for all i. For an n-node network we construct an $n \times n$ matrix with entries d_{ij}; this matrix gives the complete description of the network. The distance of a path is defined to be the sum of the arc lengths in the path. To find the shortest path between the two nodes of every pair, we perform the triple operation (the symbol $\leftarrow$ means "is replaced by")

$$d_{ik} \leftarrow \min (d_{ik}, d_{ij} + d_{jk}), \tag{1}$$

for a fixed j and all i, $k \neq j$. The value of j is first fixed at 1, then 2, $\cdots$, n. The triple operations are completed when $j = n$ and all i, $k \neq n$. The final entry in the ith row and kth column is the shortest distance from N_i to N_k and is denoted by d_{ik}^*. There is also a calculation† associated with (1) that keeps track of the intermediate nodes in all the shortest paths. The number of additions and comparisons required to complete the triple operations (1) on an n-node network is $n(n-1)(n-2)$ each. In a sense this number of operations is a minimum for a complete network. If the network is large and its associated distance matrix has many entries equal to ∞, it is advantageous to use the decomposition algorithm described below to reduce both the amount of computation and the storage requirement.

As in Ref. 5 we take a subset of nodes in the network and identify it as the set A. Let X be another subset of nodes. The set X is called a cut set of A if X has the following properties: (a) The deletion of X together with its incident arcs will make the network disconnected and (b) all the nodes of A are in one component of the disconnected network and this component does not contain any

The authors are located at the Mathematics Research Center, University of Wisconsin, Madison, Wisconsin 53706.

* Sponsored by the United States Army under Contract DA-31-124-ARO-D-462 and by the National Science Foundation under Grant GP-8557.

† See the Appendix.

nodes not in A. The set X is said to be a minimum cut set if no proper subset of X has the properties (a) and (b). Let the network N be the union of these disjoint sets of nodes A, X and B, where X is a cut set of A. We use the same symbol to denote a subset of nodes as well as the subnetwork corresponding to the subset. The arcs of the subnetwork are those that connect nodes in the subset. We consider the network N as two overlapping networks, one with nodes in $A \cup X$, the other with nodes in $B \cup X$. We denote $A \cup X$ by $\bar{A}$ and $B \cup X$ by $\bar{B}$ and the cardinality of the sets A, B and X by $|A|$, $|B|$ and $|X|$, respectively.

The distance matrix of the network A is D_{AA}. The distance matrix from A to B is $D_{AB} = (d_{ab})$ where each entry d_{ab} is the distance from a node N_a in A to another node N_b in B. In the present case, because X is a cut set of A, D_{AB} has all entries equal to ∞. The associated distance matrix of the network N is shown in Fig. 1, where white blocks (D_{AB} and D_{BA}) indicate infinite-distance entries.

To construct a general distance matrix we proceed as follows: Label any subset of nodes as A and its minimum cut set as X_A. Let B be a cut set (not necessarily a minimum cut set) of $A \cup X_A$ and X_B be the minimum cut set of $A \cup X_A \cup B$. (Note that the minimum cut set of B is $X_A \cup X_B$.) Let C be a cut set of $A \cup X_A \cup B \cup X_B$ and X_C be the minimum cut set of $A \cup X_A \cup B \cup X_B \cup C$. Continue until decomposition is no longer advantageous. (Note that the subsets A, X_A, B, X_B, etc. need not be connected.) In Fig. 2 the original network is decomposed into four overlapping networks, $\bar{A} = A \cup X_A$, $\bar{B} = X_A \cup B \cup X_B$, $\bar{C} = X_B \cup C \cup X_C$ and $\bar{D} = X_C \cup D$.

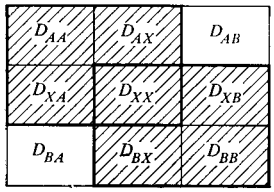


Figure 1 Distance matrix of two overlapping networks.

Figure 2 Distance matrix of four overlapping networks.

D_{AA}	D_{AX_A}	D_{AB}	D_{AX_B}	D_{AC}	D_{AX_C}	D_{AD}
D_{X_AA}	$D_{X_AX_A}$	D_{X_AB}	$D_{X_AX_B}$	D_{X_AC}	$D_{X_AX_C}$	D_{X_AD}
D_{BA}	D_{BX_A}	D_{BB}	D_{BX_B}	D_{BC}	D_{BX_C}	D_{BD}
D_{X_BA}	$D_{X_BX_A}$	D_{X_BB}	$D_{X_BX_B}$	D_{X_BC}	$D_{X_BX_C}$	D_{X_BD}
D_{CA}	D_{CX_A}	D_{CB}	D_{CX_B}	D_{CC}	D_{CX_C}	D_{CD}
D_{X_CA}	$D_{X_CX_A}$	D_{X_CB}	$D_{X_CX_B}$	D_{X_CC}	$D_{X_CX_C}$	D_{X_CD}
D_{DA}	D_{DX_A}	D_{DB}	D_{DX_B}	D_{DC}	D_{DX_C}	D_{DD}

The matrix of *shortest* distances between nodes in A is denoted by D^*_{AA}; similarly, we define D^*_{AB}, D^*_{XA}, etc. A conditional shortest path from N_i to N_j is a shortest path subject to the restriction that nodes in the path be in a certain subset of nodes of the network. The matrix of conditional shortest distances between pairs of nodes in A subject to the restriction that all nodes of the path be in A is denoted by $D^*_{AA}(A)$. Of course, we have $D^*_{AA}(N) \equiv D^*_{AA}$.

Now we state without proof the two theorems in Ref. 5:

• *Theorem 1*

Let $N = A \cup X \cup B$, where the removal of X will make the network disconnected. Then the shortest distances between nodes in the network $\bar{B}$ can be obtained by considering only the network $\bar{B}$, provided that the conditional shortest distances $D^*_{XX}(\bar{A})$ are known. (Note that $\bar{A} = N - B$.)

• *Theorem 2*

Let $N = A \cup X \cup B$, where again the removal of X will make the network disconnected. Then

$$D^*_{AB}(N) = \min_X [D^*_{AX}(N) + D^*_{XB}(N)] \tag{2}$$

and

$$D^*_{BA}(N) = \min_X [D^*_{BX}(N) + D^*_{XA}(N)]. \tag{3}$$

The operation (2) or (3) is called a matrix mini-summation because

$$d^*_{ik} = \min_j (d^*_{ij} + d^*_{jk}), \tag{4}$$

where $i \in A$, $j \in X$ and $k \in B$. The operation (4) is analogous to ordinary matrix multiplication with $+$ replacing $\times$ and min replacing summation. The number of additions (and comparisons) needed in (4) is the product $|A|\,|X|\,|B|$.

The algorithm

Consider now a network N that can be decomposed into m overlapping networks $\bar{A}, \bar{B}, \cdots, \bar{H}$ or $N = A \cup X_A \cup B \cup X_B \cup \cdots G \cup X_G \cup H$, where X_A, X_B, $\cdots$ are the minimum cut sets for A, $A \cup X_A \cup B$, $\cdots$, respectively. The general decomposition algorithm involves the following steps:

1. Perform the triple operation (1) on the $m - 1$ networks $\bar{A}, \bar{B}, \cdots, \bar{G}$ successively; the conditional shortest dis-

tances obtained in one network are used to replace the original distances in the succeeding network; i.e., $D^*_{X_A X_A}(\bar{A})$ replaces $D_{X_A X_A}$ before we perform the triple operation on $\bar{B} = X_A \cup B \cup X_B$.

2. Perform the triple operation on the m networks $\bar{H}$, $\bar{G}$, $\cdots$, $\bar{A}$ successively. The distances obtained in one network replace the distances in the succeeding network; i.e., $D^*_{X_G X_G}(N)$ replaces $D^*_{X_G X_G}(N - H)$.

3. Find the shortest distances between nodes that are not both in one of the sets $\bar{A}, \bar{B}, \cdots, \bar{H}$ by mini-summation (4).

This shortcut decomposition algorithm differs from the method in Ref. 5 in the way that the mini-summations (Step 3) are executed. We use the notation $A \oplus X_A \oplus B$ to denote the matrix mini-summation with $N_i \in A$, $N_i \in X_A$ and $N_k \in B$. Although both $A \oplus X_A \oplus B$ and $B \oplus X_A \oplus A$ must be calculated, we write only one of them. The order in which the matrix mini-summations should be executed is as follows:

$$A \oplus X_A \oplus B \cup X_B,$$
$$A \cup X_A \cup B \oplus X_B \oplus C \cup X_C,$$
$$A \cup X_A \cup B \cup X_B \cup C \oplus X_C \oplus D \cup X_D,$$
$$\vdots$$
$$A \cup X_A \cup \cdots \cup F \oplus X_F \oplus G \cup X_G \text{ and}$$
$$A \cup X_A \cup \cdots \cup G \oplus X_G \oplus H.$$

The $D^*_{AX_B}(N)$ obtained in the first matrix mini-summation are used in the second mini-summation.

To prove that the decomposition algorithm works, we use Theorems 1 and 2. In Step 1 of the algorithm, if we consider the sets A, $A \cup X_A \cup B$, $\cdots$ successively as the set A in Theorem 1 and $B \cup X_B \cup \cdots \cup H$, $C \cup X_C \cup \cdots \cup H$, $\cdots$ correspondingly as the set B in Theorem 1, we get the conditional shortest-distance matrices $D^*_{\bar{A}\bar{A}}(\bar{A})$, $\cdots$, $D^*_{\bar{G}\bar{G}}(\bar{A} \cup \bar{B} \cup \cdots \cup \bar{G})$ at the end of Step 1. In Step 2 of the algorithm, using Theorem 1 again, we obtain $D^*_{\bar{H}\bar{H}}(N)$, $\cdots$, $D^*_{\bar{A}\bar{A}}(N)$. In Step 3 of the algorithm we consider the subnetworks $A \cup X_A \cup B \cup X_B$, $A \cup X_A \cup B \cup X_B \cup C \cup X_C$, $\cdots$ with $X_A, X_B, \cdots$ as the set X in Theorem 2. Then we get the shortest distances between all pairs of nodes in the subnetworks. The last subnetwork is the network N.

Discussion

To reveal the idea in Theorem 1 more clearly, we first consider why the triple operation (1) works. (Various proofs on triple operations are available; see, for example, Refs. 1, 3, 5 and 6.) Basically, we create an arc between every pair of nodes such that the length of this arc is equal to the shortest distance between the nodes. To show that this shortest-path arc also results from the decomposition algorithm, we consider a network decomposed into four overlapping networks (Fig. 3) and show that, for a pair of nodes such as N_p and $N_q \in \bar{B}$, an arc of length equal to the shortest distance from N_p to N_q is created at the end of Step 2 of the decomposition algorithm. The solid lines arbitrarily represent this shortest path in Fig. 3. Because any subpath of a shortest path is itself a shortest path, the subpaths from N_1 to N_2, N_3 to N_4 and N_5 to N_6 are all shortest paths.

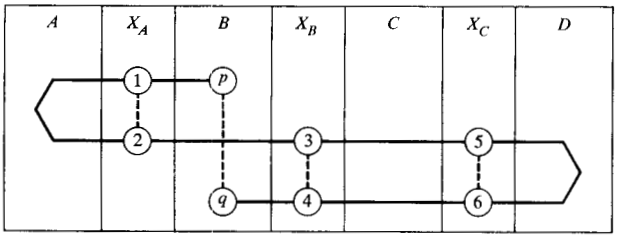


Figure 3 Development of a shortest-path arc in four overlapping networks.

In Step 1 of the decomposition algorithm, after the triple operation is performed on $\bar{A}$, an arc (dashed line) is created from N_1 to N_2 of length equal to that of the subpath. Now there exists a new shortest path (containing the newly created arc) from N_p to N_q which lies entirely in $N - A$. In Step 2, after the triple operation is performed on $\bar{D} = X_C \cup D$, an arc is created from N_5 to N_6 of length equal to that of the subpath. Thus there exists a new shortest path from N_3 to N_4 which lies entirely in $N - D$. Then in Step 2, after the triple operation is performed on $\bar{C}$, an arc is created from N_3 to N_4 of length equal to that of the shortest path from N_3 to N_4. Now there exists a shortest path from N_p to N_q which lies entirely in $N - A - (C \cup X_C \cup D)$. Finally in Step 2, after the triple operations are performed on $\bar{B}$, an arc is created from N_p to N_q of length equal to the shortest distance from N_p to N_q.

To approximate the number of arithmetic operations used in the decomposition algorithm, we assume that $|A| = |B| = \cdots = |H| = t$ and $|X_A| = |X_B| = \cdots = |X_G| = \delta$. We calculate only the number of additions; the number of comparisons needed is the same. In Step 1 the number of additions is $(t + \delta)^3 + (m - 2)(t + 2\delta)^3$; in Step 2 the number of additions is $2(t + \delta)^3 + (m - 2) \times (t + 2\delta)^3$; in Step 3 the number of additions is

$$\begin{aligned} &2\{t + (2t + \delta) + \cdots + [(m - 2)t + (m - 3)\delta]\} \\ &\quad \times \delta(t + \delta) + 2[(m - 1)t + (m - 2)\delta]\delta t \\ &= m(m - 1)t^2\delta + 2(m - 1)(m - 2)t\delta^2 \\ &\quad + (m - 2)(m - 3)\delta^3. \end{aligned}$$

The total number of additions is therefore

$$(2m - 1)t^3 + (m^2 + 11m - 15)t^2\delta + (2m^2 + 18m - 35)t\delta^2 + (m^2 + 11m - 23)\delta^3. \quad (5)$$

If we do not use the decomposition algorithm, but use the triple operation on the entire matrix, the number of additions is

$$[mt + (m - 1)\delta]^3. \tag{6}$$

For $t \geq \delta$ and $m \geq 2$, the number of additions in (5) is always less than that in (6). For large m the expression (5) approaches $m^2\delta(t + \delta)^2$ and (6) approaches $m^3(t + \delta)^3$. Thus the ratio of (5) to (6) approaches $[\delta/(t + \delta)]/m$ as m increases. In Ref. 5 the number of additions required is

$$\begin{aligned}(2m - 1)t^3 &+ (m^2 + 11m - 15)t^2\delta \\ &+ (m^2 + 21m - 37)t\delta^2 \\ &+ (m^3 - 3m^2 + 19m - 28)\delta^3,\end{aligned} \tag{7}$$

which is always larger than the value of expression (5).

Although the networks discussed have been composed of linearly overlapping sets as in Fig. 4a, the network decomposition described can also be applied to arbitrarily overlapping sets such as those shown in Fig. 4b. Using this figure as an example, we can decompose the network linearly by first letting $\bar{A}' = \bar{E}$, $\bar{B}' = \bar{A} \cup \bar{F}$, $\bar{C}' = \bar{B} \cup \bar{C} \cup \bar{D}$, $\bar{D}' = \bar{G}$ and $\bar{E}' = H$. The network $\bar{B}'$ can be decomposed further into two small networks $\bar{A}$ and $\bar{F}$ and the network $\bar{C}'$ into three small networks $\bar{B}$, $\bar{C}$ and $\bar{D}$. It is possible to decompose the network in Fig. 4b into the networks $\bar{A}$, $\bar{B}$, $\cdots$, $\bar{H}$ directly, but the number of operations needed is more than the number necessary to decompose the network linearly.

If we are interested in finding the shortest paths between some (but not all) pairs of nodes, we can simplify the problem by constructing a smaller network that is distance-equivalent† to the original network with respect to the pairs of nodes of interest.‡ Formulas similar to Wye-delta transformations[7] are used in conjunction with the decomposition algorithm.

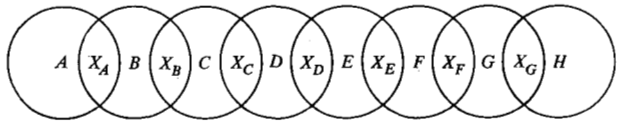
Appendix

To keep track of the arcs that make up the shortest paths, we use the following bookkeeping: In a table whose entries are calculated along with the triple operation (1), let the entry in the ith row and kth column indicate the first intermediate node on the shortest path from N_i to N_k and let k be the entry if there is no intermediate node. At the start all entries in the (i, k) positions are set equal to k. Then

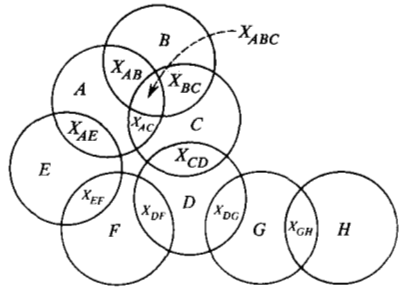
$$(i, k)\begin{cases} = (i, j) & \text{if } d_{ik} > d_{ij} + d_{jk} \text{ or} \\ \text{remains the same} & \text{if } d_{ik} \leq d_{ij} + d_{jk}. \end{cases}$$

† A p-node network is said to be distance-equivalent to an n-node network, $p \leq n$, if the distances of $p(p - 1)$ pairs of shortest paths in the two networks are the same.

‡ The idea of constructing a smaller-network maximal-flow-equivalent to the original network was developed by Akers.[7]



(a)



(b)

Figure 4 Network decomposition: (a) linearly overlapping sets and (b) nonlinearly overlapping sets.

References

1. G. B. Dantzig, "All Shortest Routes in a Graph," *Operations Research Technical Report 66-3,* Stanford University, November 1966.
2. R. W. Floyd, "Algorithm 97: Shortest Path," *Comm. ACM* **5,** 345 (1962).
3. J. D. Murchland, "A New Method for Finding All Elementary Paths in a Complete Directed Graph," *Report LSE-TNT-22,* London School of Economics, October 1965.
4. A. H. Land and S. W. Stairs, "The Extension of the Cascade Algorithm to Large Graphs," *Management Sci.; Theory (Series A)* **14,** 29 (1967).
5. T. C. Hu, "A Decomposition Algorithm for Shortest Paths in a Network," *Operations Res.* **16,** 91 (1968).
6. T. C. Hu, *Integer Programming and Network Flows,* Addison-Wesley Publishing Co., Inc., Cambridge, Mass. 1969.
7. J. B. Akers, Jr., "The Use of Wye-delta Transformations in Network Simplification," *Operations Res.* **8,** 311 (1960).

Received October 1, 1968