# Shortest Paths in Networks with Exponentially Distributed Arc Lengths

**V. G. Kulkarni**
*Curriculum in Operations Research and Systems Analysis, The University of North Carolina at Chapel Hill, Chapel Hill, North Carolina 27514*

This paper develops methods for the exact computation of the distribution of the length of the shortest path from a given source node $s$ to a given sink node $t$ in a directed network in which the arc lengths are independent and exponentially distributed random variables. A continuous time Markov chain with a single absorbing state is constructed from the original network such that the *time* until absorption into this absorbing state starting from the initial state is equal to the *length* of the shortest path in the original network. It is shown that the state space of this Markov chain is the set of all minimal $(s,t)$ cuts in the network and that its generator matrix is upper triangular. Algorithms are described for computing the distribution and moments of the length of the shortest path based on this Markov chain representation. Algorithms are also developed for computing the probability that a given $(s,t)$ path is the shortest path in the network and for computing the conditional distribution of the length of a path given that it is the shortest $(s,t)$ path in the network. All algorithms are numerically stable and are illustrated by several numerical examples.

## 1. INTRODUCTION

In this paper we develop analytical methods for the exact computation of the distribution of the length of the shortest path from a given source node $s$ to a given sink node $t$ in a directed network in which the lengths of the arcs are independent and exponentially distributed random variables. We also develop algorithms for the computation of the probability that a given path is the shortest $(s,t)$ path in the network and the conditional distribution of the length of a path given that it is the shortest $(s,t)$ path in the network.

In the case of deterministic arc lengths, there is considerable literature on the

This research was supported by the Air Force Office of Scientific Research under contract AFOSR-84-0140. Reproduction in whole or part is permitted for any purpose of the United States Government.

NETWORKS, Vol. 16 (1986) 255–274
 CCC 0028-3045/86/030255-20$04.00

problem of finding the shortest path between two nodes: see Deo and Pang [3] for a recent bibliography on the subject.

When the arc lengths are allowed to be random variables, the problem becomes considerably more difficult. Recently, attention has been given to the analysis of shortest paths in such "stochastic networks." The following questions have been addressed in the literature:

1. Distribution of the length of the shortest path.
2. Moments of the length of the shortest path.
3. Optimality index of a path: the probability that a given path is the shortest path in the network.

The analysis of some of these problems for general stochastic networks has been attempted by Pritsker [10] and Frank [5]. Martin [8] has analyzed the problem by using the unique arcs concept, and Sigal et al. [13] have used the uniformly directed cuts in their analysis of shortest paths. Each one of the above four papers represents the required probabilistic quantities as multiple integrals. Numerical evaluation of these integrals quickly gets out of hand, even for small networks. Mirchandani [9] uses an entirely different approach that avoids the evaluation of multiple integrals, but his approach works only when the arc lengths are discrete random variables.

Owing to the computational difficulties in the exact computation of the distribution of the length of the shortest path, recently attention has shifted to Monte Carlo simulation of stochastic networks to obtain estimates of the required distributions. Simulation methods for estimating some of the quantities listed above are developed by Sigal et al. [14] and Adlakha and Fishman [1].

In this paper we consider a special class of stochastic networks, namely networks with independent and exponentially distributed arc lengths. The results of this paper can be readily extended to stochastic networks whose arc lengths have phase-type distributions, and it is well known that phase-type distributions can be used to approximate nonexponential distributions. This extension, although straightforward, needs a lot of extra notation, and hence will be treated in a separate paper. If one goes beyond the phase-type distributions to the general distributions, the stochastic processes developed here remain theoretically valid but do not yield any numerically useful algorithms.

Section 2 introduces the relevant network terminology. We construct two alternative but equivalent continuous time Markov chains such that the *time* until absorption in the absorbing state starting from the initial state is equal to the *length* of the shortest path in the network. The state space of this Markov chain is shown to be the set of all minimal $(s,t)$ cuts in the network, augmented by an absorbing state. The generator matrix of the chain is shown to be upper triangular.

Using this structure, Section 3 develops algorithms to compute the distribution and moments of the length of the shortest path. Tight error bounds on the numerical evaluation of the distribution are given.

Section 4 describes the algorithms to compute the optimality indices of the paths in the network. Section 5 is devoted to the conditional analysis of the length of the shortest path.

All the algorithms described in Sections 3, 4, and 5 are simple, easy to implement on a computer, and computationally stable. Numerical experience with these algorithms is documented with the help of three networks in Section 6. Finally, the concluding Section 7 describes how the results of this paper can be used in undirected networks.

## 2. THE STOCHASTIC PROCESSES

Let $G = (V,A)$ be a directed network with node set $V$ and arc set $A$. Let $s$ and $t$ be two prespecified nodes in $V$ called the source and the sink, respectively. Let $L(u,v)$ be the length of the arc $(u,v) \in A$. In this section we construct a stochastic process such that the length of the shortest directed $(s,t)$ path in $G$ has the same distribution as that of a particular first passage time in this stochastic process.

To construct this stochastic process, it is convenient to visualize the network as a communication network with the nodes as stations capable of receiving and transmitting messages and arcs as one-way communication links connecting pairs of nodes. The messages are assumed to travel at a unit speed so that $L(u,v)$ denotes the travel time from node $u$ to $v$. As soon as a node receives a message over one of the incoming arcs, it transmits it along all the outgoing arcs and then disables itself (i.e., loses the ability to receive and transmit any future messages). This process continues until the message reaches the sink node $t$. Now, at any time there may be some nodes and arcs in the network that are "useless" for the progress of the message towards the sink node, i.e., even if the messages are received and transmitted by these nodes and carried by these arcs, the message can only reach disabled nodes. It is assumed that all such "useless" nodes are also disabled and the messages traveling on such arcs are aborted. For example, consider the example network shown in Figure 1. Suppose that the messages are currently traveling on arcs 1, 2, and 3. Suppose the message on arc 2 reaches node 3 before messages on arcs 1 and 3 reach their respective destinations. After node 3 receives the message, it transmits it along arcs 5 and 6 and disables itself. At this point, node 2 is also disabled, and the message traveling on arc 1 is aborted, since all messages from node 2 will eventually have to reach node 3, which is already disabled. Node 4 and the message on arc 3 however, are unaffected, since it is possible to go to the sink node 5 from node 4 without visiting a disabled node.

Now, let $X(t)$ be the set of all disabled nodes at time $t$. $X(t)$ is called the state

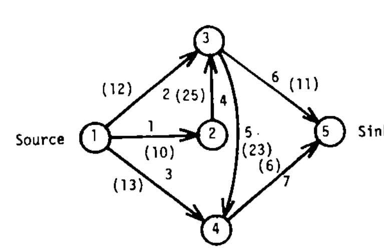


FIG. 1. Example network. The numbers on arcs are arc numbers. The numbers in brackets represent mean arc lengths.

of the network at time $t$. To describe the evolution of the $\{X(t),\ t \geqq 0\}$ process, we introduce the following notation. For any $X \subset V$ such that $s \in X$ and $t \in \overline{X} = V - X$, define

$R(X) = \{v \in V$: there is a path from $v$ to $t$ that does not visit any nodes in $X\}$

and

$$S(X) = V - R(X).$$

The process starts at time 0, with $X(0-) = \phi$ when the source node receives the message (from outside the network) to be transmitted to the sink node.

When a node $u \in V$ receives the message at time $t$, three events happen in zero time in the order described below:

E1. The message is transmitted from $u$ to all $v \in V$ such that $(u,v) \in A$.
E2. All nodes in $S(X(t-) \cup \{u\})$ are disabled.
E3. All messages heading for nodes in $S(X(t-) \cup \{u\})$ are aborted.

Note that once a message is transmitted, it keeps traveling even if the transmitting station is disabled. The message gets transmitted and retransmitted until the sink node receives it, at which point all nodes are disabled and the process terminates.

Now define

$$\Omega = \{X \subset V: s \in X,\ t \in \overline{X},\ X = S(X)\}$$

and

$$\Omega^* = \Omega \cup \{V\}.$$

The following theorem characterizes the state of the network at any time $t$.

**Theorem 1.** For $t \geqq 0$, $X(t) \in \Omega^*$.

*Proof.* From the description of events E1, E2, and E3, it is obvious that

1. $X(0+) = S(\{s\})$ and
2. If a node $u$ receives the message at time $t$, the state of the network changes from $X(t-)$ to $X(t+) = S(X(t-) \cup \{u\})$.

Now fix $t \geqq 0$, suppose that $X(t-) \in \Omega$, and consider two cases:

1. No node receives the message at time $t$. Then $X(t+) = X(t-)$, since the network state does not change at $t$.
2. A node $u$ receives the message at time $t$. Then $X(t+) = S(X(t-) \cup \{u\})$. Since $S(S(X)) = S(X)$ by the definition of $S(X)$, we have $S(X(t+)) = X(t+)$. If node $u$ is the sink node, $X(t+) = V$.

Thus, in both cases $X(t-) \in \Omega \Rightarrow X(t+) \in \Omega^*$ for $t \geqq 0$. Since $X(0+) = S(\{s\}) \in \Omega$, and $X(t) = V \Rightarrow X(t') = V$ for $t' \geqq t$, this implies that $X(t) \in \Omega^*$ for $t \geqq 0$. Hence the theorem follows. ■

Now let $X \subset V$ and define

$$C(X,\overline{X}) = \{(u,v) \in A: u \in X,\ v \in \overline{X}\}.$$

If $X = V$, $C(V,\bar{V})$ is defined to be empty, let

$$Y(t) = \{(u,v) \in A\text{: the arc } (u,v) \text{ is carrying the message at time } t\}.$$

An $(s,t)$ cut is called a minimal cut if no proper subset of it is an $(s,t)$ cut. It is shown in Provan and Ball [9] that there is a unique minimal cut contained in $C(X,\bar{X})$ if $X \in \Omega$. Denote this cut by $C(X)$. This cut is characterized as follows: $E$ is the minimal cut contained in $C(X,\bar{X})$ if and only if $E \subset C(X,\bar{X})$, and for every $(u,v) \in E$ there is a path from $s$ to $u$ that does not use any arcs of $C(X,\bar{X})$. With this characterization we are now able to prove the following:

**Theorem 2.** $X(t) \neq V \Rightarrow Y(t) = C(X(t))$, the minimal cut contained in $C(X(t), \overline{X(t)})$.

*Proof.* Suppose $(u,v) \in Y(t)$, i.e., the arc $(u,v)$ is carrying the message at time $t$. Thus the message must have reached node $u$ by time $t$ and must not have reached node $v$ by time $t$. Hence $u \in X(t)$, $v \in \overline{X(t)}$. Also, this message could not have visited any nodes in $\overline{X(t)}$; hence there must be a path from $s$ to $u$ that does not use any arcs of $C(X(t), \overline{X(t)})$. Hence $(u,v) \in C(X(t))$.

Conversely, suppose $(u,v) \in C(X(t))$. Hence $u \in X(t)$ and $v \in \overline{X(t)}$. Thus the message must have reached $u$ by time $t$. By the construction of the message transmission process, messages must be traveling on all arcs leaving $u$ and ending in any node in $\overline{X(t)}$; hence $(u,v)$ must be carrying the message. Hence $(u,v) \in Y(t)$.

This proves the theorem. ∎

**Remark.** It is clear that specifying $Y(t)$ instead of $X(t)$ is an alternative way of describing the network state. The state space for the process $\{Y(t),\ t \geqq 0\}$ is the union of the set of all minimal cuts in $G$ and $\{\phi\}$ since $X(t) = V \leftrightarrow Y(t) = \phi$.

We now make the following two assumptions:

A1: $L(u,v)$ $((u,v) \in A)$ are independent random variables.

A2: $L(u,v)$ is exponentially distributed with parameter $\mu(u,v) > 0$ $((u,v) \in A)$ (i.e., mean $= 1/\mu(u,v)$).

With these two assumptions we obtain the following theorem.

**Theorem 3.** Under assumptions A1 and A2, $\{X(t), t \geqq 0\}$ is a continuous time Markov chain (CTMC) with state space $\Omega^*$ and infinitesimal generator matrix $Q = [q(D,B)]$ $(D,B \in \Omega^*)$ given by

$$q(D,B) = \begin{cases} \sum\limits_{(u,v) \in C_v(D)} \mu(u,v) & \text{if } B = S(D \cup \{v\}) \text{ for some } v \in \bar{D}, \\ -\sum\limits_{(u,v) \in C(D)} \mu(u,v) & \text{if } B = D, \\ 0 & \text{otherwise.} \end{cases} \tag{2.1}$$

where $C_v(D) = \{(u,v) \in C(D)\}$ and $C(V) = \phi$.

TABLE I. State space for the example network.

| State $i$ | State Description for $X(t)$ (set of disabled nodes) | State Description for $Y(t)$ (set of arcs carrying messages) |
|---|---|---|
| 1 | {1} | {1,2,3} |
| 2 | {1,2} | {2,3,4} |
| 3 | {1,4} | {1,2,7} |
| 4 | {1,2,4} | {2,4,7} |
| 5 | {1,2,3} | {3,5,6} |
| 6 | {1,2,3,4} | {6,7} |
| 7 | {1,2,3,4,5} | $\phi$ |

*Proof.* $\{X(t), t \geqq 0\}$ is a CTMC as a consequence of A1 and A2. Theorem 1 implies that the state space of $\{X(t), t \geqq 0\}$ is $\Omega^*$. Suppose $X(t) = D \in \Omega$. Then all arcs in $C(D)$ are carrying the message. If the message on arc $(u,v)$ reaches node $v$, which happens with rate $\mu(u,v)$, the state of the network changes from $D$ to $B = S(D \cup \{v\})$. Hence $q(D,B) = \Sigma_{(u,v)\in C_v(D)}\, \mu(u,v)$. Once the message reaches the sink node at time $T$, $X(t) = V$ for $t \geqq T$. Hence $q(V,B) = 0$ for all $B \in \Omega^*$. $q(D,D)$ follows from the fact that row sums of $Q$ are zero. Hence the theorem follows. ∎

Whenever there is a transition in the $\{X(t), t \geqq 0\}$ process, the number of disabled nodes in the network increases by at least one. Hence the process cannot visit any state more than once. Now order the elements of $\Omega^*$ by nondecreasing cardinality, i.e., if $D$, $B \in \Omega^*$ and $|D| < |B|$, then $D$ appears before $B$ in this ordering. If $|D| = |B|$, then the ordering of $D$ and $B$ is immaterial. It is obvious that under this ordering the generator matrix $Q$ is upper triangular. Since $q(D,D) \neq 0$ for $D \in \Omega$, all states in $\Omega$ are transient. The state $V$, on the other hand, is absorbing, since $q(V,V) = 0$. Thus, the process $\{X(t), t \geqq 0\}$ is a finite state absorbing CTMC with an upper triangular generator matrix. This observation is of central importance in all the algorithms that are developed in this paper.

The material of this section is illustrated below by means of an example network

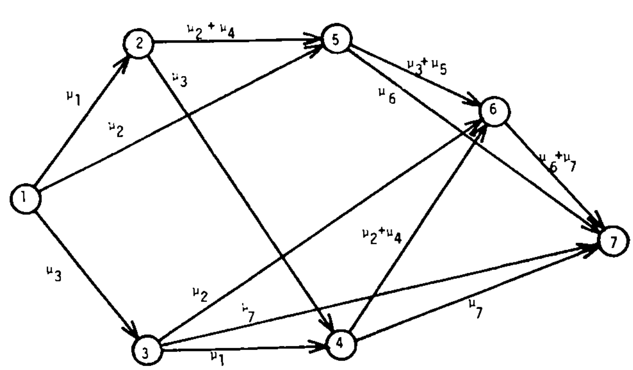


FIG. 2. Rate diagram for the example network.

shown in Figure 1. The network has five nodes and seven arcs. Node 1 is the source node, and node 5 is the sink node. We are interested in the shortest path from node 1 to 5. The CTMC $\{X(t),\ t \geqq 0\}$ for this network has seven states since there are six minimal cuts. The states are given in Table I. Columns two and three in the table describe the state space for the $X(t)$ and $Y(t)$ processes, respectively. Thus, the sets in column two and three are sets of nodes and arcs, respectively. The states are referred to by their number in column 1 throughout this paper. The rate diagram for the CTMC $\{X(t),\ t \geqq 0\}$ is shown in Figure 2. In this diagram, and in the rest of the paper, $\mu_i$ stands for the parameter of the exponential distribution associated with arc $i$. The upper triangular nature of the $Q$ matrix is reflected in the directed acyclic structure of the rate diagram. This example is used in the rest of this paper to illustrate the algorithms.

## 3. SHORTEST PATH ANALYSIS

In this section we develop algorithms for computing the distribution and moments of the length of the shortest path in stochastic networks satisfying assumptions A1 and A2. For each case it is possible to develop two algorithms, called backward and forward algorithms, borrowing terminology from the theory of CTMCs. The backward algorithms are illustrated by means of the example network.

From now on we assume that the states in $\Omega^*$ are numbered $1,2,\ldots,N = |\Omega^*|$ as suggested in the previous section so that the $Q$ matrix is upper triangular. State 1 is the initial state, namely $S(\{s\})$, and state $N$ is the final (absorbing) state, namely $V$.

### 3.1. The Distribution of the Length of the Shortest Path

Let $T$ be the length of the shortest path in the network. From the construction of the CTMC $\{X(t),\ t \geqq 0\}$ it is clear that

$$T = \min\{t \geqq 0\colon X(t) = N | X(0) = 1\}. \tag{3.1.1}$$

Thus, the *length* of the shortest path in the network is equal to the *time* until the CTMC $\{X(t),\ t \geqq 0\}$ gets absorbed in the final state starting from state 1. This is so because the messages travel at unit speed. We are interested in

$$F(t) = P\{T \leqq t\}. \tag{3.1.2}$$

We now describe two algorithms to compute $F(t)$.

**Backward Algorithm:** Define

$$p_i(t) = P\{X(t) = N | X(0) = i\}, \quad 1 \leqq i \leqq N. \tag{3.1.3}$$

Then $F(t) = p_1(t)$. The differential equations for $p_i(t)$ are given by

$$\left.\begin{aligned} p_i'(t) &= \sum_{j \geq i} q_{ij} p_j(t) \\ p_i(0) &= \delta_{iN} \end{aligned}\right\} \quad 1 \leqq i \leqq N \tag{3.1.4}$$

TABLE II. The generator matrix $Q$ for the example network.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| 1 | $-(\mu_1 + \mu_2 + \mu_3)$ | $\mu_1$ | $\mu_3$ | 0 | $\mu_2$ | 0 | 0 |
| 2 | 0 | $-(\mu_2 + \mu_3 + \mu_4)$ | 0 | $\mu_3$ | $\mu_2 + \mu_4$ | 0 | 0 |
| 3 | 0 | 0 | $-(\mu_1 + \mu_2 + \mu_7)$ | $\mu_1$ | 0 | $\mu_2$ | $\mu_7$ |
| 4 | 0 | 0 | 0 | $-(\mu_2 + \mu_4 + \mu_7)$ | 0 | $\mu_2 + \mu_4$ | $\mu_7$ |
| 5 | 0 | 0 | 0 | 0 | $-(\mu_3 + \mu_5 + \mu_6)$ | $\mu_3 + \mu_5$ | $\mu_6$ |
| 6 | 0 | 0 | 0 | 0 | 0 | $-(\mu_6 + \mu_7)$ | $\mu_6 + \mu_7$ |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

where $\delta_{ij} = 1$ if $i = j$ and 0 otherwise. Owing to the upper triangular nature of $Q$, Eqs. (3.1.4) can be easily solved in backward fashion, starting with $p_N(t) = 1$ for $t \geqq 0$, and computing $p_{N-1}(t), \ldots, p_1(t)$ recursively.

**Example:** For the example network, Eqs. (3.1.4) become

$$\begin{aligned}
p_7'(t) &= 0, \\
p_6'(t) &= -(\mu_6 + \mu_7)p_6(t) + (\mu_6 + \mu_7)p_7(t), \\
p_5'(t) &= -(\mu_3 + \mu_5 + \mu_6)p_5(t) + (\mu_3 + \mu_5)p_6(t) + \mu_6 p_7(t), \\
p_4'(t) &= -(\mu_2 + \mu_4 + \mu_7)p_4(t) + (\mu_2 + \mu_4)p_6(t) + \mu_7 p_7(t), \\
p_3'(t) &= -(\mu_1 + \mu_2 + \mu_7)p_3(t) + \mu_1 p_4(t) + \mu_2 p_6(t) + \mu_7 p_7(t), \\
p_2'(t) &= -(\mu_2 + \mu_3 + \mu_4)p_2(t) + \mu_3 p_4(t) + (\mu_2 + \mu_4)p_5(t), \\
p_1'(t) &= -(\mu_1 + \mu_2 + \mu_3)p_1(t) + \mu_1 p_2(t) + \mu_3 p_3(t) + \mu_2 p_5(t).
\end{aligned} \tag{3.1.5}$$

**Forward Algorithm:** Define

$$\bar{p}_j(t) = P\{X(t) = j | X(0) = 1\}, \quad 1 \leqq j \leqq N. \tag{3.1.6}$$

Then $F(t) = \bar{p}_N(t)$. The differential equations for $\bar{p}_j(t)$ are given by

$$\left.\begin{aligned}
\bar{p}_j'(t) &= \sum_{i \leqq j} \bar{p}_i(t) q_{ij}, \\
\bar{p}_j(0) &= \delta_{j1}.
\end{aligned}\right\} \quad 1 \leqq j \leqq N \tag{3.1.7}$$

Since $Q$ is upper triangular, the above equations can be solved in a forward manner, starting with $\bar{p}_1(t) = \exp(q_{11}t)$, $t \geqq 0$, and computing $\bar{p}_2(t), \ldots, \bar{p}_N(t)$ in that order.

**Numerical Evaluation of $F(t)$:** The numerical evaluation of $F(t)$ is carried out by using the randomization technique as described in Kulkarni and Adlakha [7]. A brief summary of the important results there is given below for completeness. Define

$$q = \max_{i \leqq i \leqq N} \{-q_{ii}\} \tag{3.1.8}$$

and let

$$q_{ij}^* = \delta_{ij} + q_{ij}/q \qquad i,j = 1,2, \ldots, N. \tag{3.1.9}$$

Then it is possible to write

$$F(t) = \sum_{n=0}^{\infty} \alpha_n e^{-qt}(qt)^n/n! \tag{3.1.10}$$

where $\alpha_n$ is the $(1,N)$th element in the $n$th power of $Q^* = [q_{ij}^*]$. It is computationally straightforward to compute $\alpha_n$'s recursively. Some consequences of the representation (3.1.10) are given below without proofs.

**Result 1.** $0 \leqq \alpha_n \leqq 1$ and $\alpha_n$ increases to 1 as $n \to \infty$.

**Result 2.** Define

$$q' = \min_{1 \leqq i < N} \{-q_{ii}\}. \tag{3.1.11}$$

If all $q_{ii}$ are distinct,

$$\lim_{n \to \infty} (1 - \alpha_{n+1})/(1 - \alpha_n) = (q - q')/q. \tag{3.1.12}$$

**Result 3.**

$$E(T^k) = \frac{k!}{q} \sum_{n=0}^{\infty} \binom{n + k - 1}{n} (1 - \alpha_n), \quad (k \geqq 1). \tag{3.1.13}$$

**Result 4.** Define

$$F_M(t) = 1 - \sum_{n=0}^{M} (1 - \alpha_n) e^{-qt} (qt)^n / n!, \tag{3.1.14}$$

$$\tilde{F}_M(t) = \alpha_M - \sum_{n=0}^{M} (\alpha_M - \alpha_n) e^{-qt} (qt)^n / n!. \tag{3.1.15}$$

Then

$$0 \leqq \tilde{F}_M(t) - F_M(t) \leqq 1 - \alpha_M \quad \text{for all } t \geqq 0 \tag{3.1.16}$$

and

$$F_M(t) \leqq F(t) \leqq \tilde{F}_M(t) \quad \text{for all } t \geqq 0. \tag{3.1.17}$$

This result is implied by results 4 and 5 in Section 4 of Kulkarni and Adlakha [7]. This is a very useful result in the numerical computation of $F(t)$: if it is desired to compute $F(t)$ within $\epsilon$, choose an $M$ so that $\alpha_M > 1 - \epsilon$. Then the resulting $F_M(t)$ and $\tilde{F}_M(t)$ bound $F(t)$ from below and above within $\epsilon$, respectively.

### 3.2. Moments of the Length of the Shortest Path

We now describe algorithms to compute the $k$th moment of the length of the shortest path without using Eq. (3.1.13).

**Forward Algorithm:** Define

$$T_i = \min\{t \geqq 0: X(t) = N | X(0) = i\}, \; 1 \leqq i \leqq N \tag{3.2.1}$$

and let

$$\tau_i(k) = E(T_i^k), \quad (k \geqq 0). \tag{3.2.2}$$

Since $\{X(t), t \geqq 0\}$ is an absorbing CTMC, it follows that $P\{T_i < \infty\} = 1$, and hence $\tau_i(0) = 1$ for all $1 \leqq i \leqq N$. It is also obvious that for $k \geqq 1$, $\tau_N(k) = 0$.

It can be proved by induction that $\tau_i(k)$ ($k \geqq 1$, $1 \leqq i < N$) satisfy the following equations:

$$\tau_i(k) = [k\tau_i(k-1) + \sum_{j>i} q_{ij}\tau_j(k)]/q_i \tag{3.2.3}$$

where $q_i = -q_{ii}$. We have $E(T^k) = \tau_1(k)$. Thus, to compute $E(T^k)$, one needs to compute $\tau_i(r)$ for $r = 1,2, \ldots ,k$; $i = N,N-1, \ldots ,1$ in that order.

**Example:** For the example network, Eqs. (3.2.3) become

$$\begin{aligned}
\tau_7(k) &= 0,\\
\tau_6(k) &= (k\tau_6(k-1) + (\mu_6 + \mu_7)\tau_7(k))/(\mu_6 + \mu_7),\\
\tau_5(k) &= (k\tau_5(k-1) + (\mu_3 + \mu_5)\tau_6(k) + \mu_6\tau_7(k))/(\mu_3 + \mu_5 + \mu_6),\\
\tau_4(k) &= (k\tau_4(k-1) + (\mu_2 + \mu_4)\tau_6(k) + \mu_7\tau_7(k))/(\mu_2 + \mu_4 + \mu_7),\\
\tau_3(k) &= (k\tau_3(k-1) + \mu_1\tau_4(k) + \mu_2\tau_6(k) + \mu_7\tau_7(k))/(\mu_1 + \mu_2 + \mu_7),\\
\tau_2(k) &= (k\tau_2(k-1) + \mu_3\tau_4(k) + (\mu_2 + \mu_4)\tau_5(k))/(\mu_2 + \mu_3 + \mu_4),\\
\tau_1(k) &= (k\tau_1(k-1) + \mu_1\tau_2(k) + \mu_3\tau_3(k) + \mu_2\tau_5(k))/(\mu_1 + \mu_2 + \mu_3).
\end{aligned} \tag{3.2.4}$$

**Forward Algorithm:** Define

$$\overline{T}_j = \inf\{t \geqq 0\colon X(t) = j | X(0) = 1\} \tag{3.2.5}$$

where the infimum over the empty set is defined to be $+\infty$. Let

$$\bar{\tau}_j(k) = E(\overline{T}_j^k), \quad (k \geqq 0). \tag{3.2.6}$$

Note that it is possible to have $\tau_j(0) = P\{T_j < \infty\} < 1$ for $1 \leqq j < N$. It can be shown that

$$\begin{aligned}
\bar{\tau}_j(k) &= \sum_{i<j} \left[ \sum_{r=0}^{k} (k!/r!)(\bar{\tau}_i(r)/q_i^{k-r}) \right] (q_{ij}/q_i),\\
\bar{\tau}(k) &= \delta_{k0}, \quad k \geqq 0,\ 1 \leqq j \leqq N.
\end{aligned} \tag{3.2.7}$$

The above equations can be solved in a forward manner to yield $\bar{\tau}_N(k) = E(T^k)$. Note that the backward algorithm requires less computer time and memory than the forward algorithm.

## 4. OPTIMALITY INDEX OF A PATH

Let $l$ be a prespecified $(s,t)$ path in the network. The optimality index of $l$, denoted by $R(l)$, is the probability that $l$ is the shortest $(s,t)$ path in the network. In this section we state a theorem that yields simple algorithms for computing $R(l)$ in networks with exponentially distributed arc lengths.

Suppose $v_1(= s)$, $v_2, \ldots ,v_k(= t)$ are the nodes visited by the path $l$. Let $l_i$ be the $(s,v_i)$ path that visits nodes $v_1,v_2, \ldots ,v_i$ $(i = 2, \ldots ,k)$. The following lemmas are used to construct algorithms to compute $R(l)$.

**Lemma 1.** Path $l$ is the shortest $(s,t)$ path in $G$ if and only if, for $i = 2, \ldots ,$ $k$, $l_i$ is the shortest $(s,v_i)$ path in $G$.

*Proof.* $(<=)$ Obvious, since $l_k$ is the same as $l$. $(=>)$ Suppose $p_1$ is a $(s,v_i)$ path that is shorter than $l_i$. Let $j$ be such that the path $p_1$ visits $v_j$ but not the nodes $v_{j+1}, \ldots ,v_k$. Now construct a path $p_2$ as follows. If $j = i$, then $p_2$ is the same as $p_1$. If $j > i$, then $p_2$ is equal to the $(s,v_j)$ segment of $p_1$. Then, obviously, $p_2$ is shorter than $l_j$ and $p_2$ does not visit any of the nodes $v_{j+1}, \ldots ,v_k$. Now construct a $(s,t)$ path $l'$ by appending the $(v_j,v_k)$ segment of $l$ onto $p_2$. Then it is clear that $l'$ is a simple $(s,t)$ path that is shorter than $l$. Hence $l$ is not the shortest path in $G$. ■

**Lemma 2.** Consider the communication model of the network as described in Section 2. The path $l$ is the shortest path in $G$ if and only if the message reaches the node $v_i$ along the arc $(v_{i-1},v_i)$ for all $i = 2, \ldots ,k$.

*Proof.* By using an argument similar to the one in the proof of Lemma 1, it can be seen that $l_i$ is the shortest $(s,v_i)$ path in $G$ if and only if the message reaches node $v_r$ along the arc $(v_{r-1},v_r)$ for all $r = 2, \ldots ,i$. The lemma now follows, since $l_k$ is the same as $l$.

Note that the above lemmas are true for all stochastic networks, whether or not they satisfy assumptions A1 and A2. If the network satisfies assumptions A1 and A2, one can use the theory of absorbing CTMCs to compute the optimality index of any given path as described below:

For $(u,v) \in A$ define

$$\mu_l(u,v) = \begin{cases} 0 & \text{if } l \text{ visits } v \text{ but not } u \\ \mu(u,v) & \text{otherwise.} \end{cases} \tag{4.5}$$

Construct a stochastic process $\{X_l(t),\ t \geqq 0\}$ on state space $\Omega^*$ with generator matrix defined below:

$$q_l(D,B) = \begin{cases} \displaystyle\sum_{(u,v) \in C_v(D)} \mu_l(u,v) & \text{if } B = S(D \cup \{v\}) \quad \text{for some } v \in \overline{D} \\ \displaystyle -\sum_{(u,v) \in C_v(D)} \mu(u,v) & \text{if } B = D. \end{cases} \tag{4.6}$$

If the states $D$ and $B$ have numbers $i$ and $j$, then $q_l(D,B)$ will be equivalently written as $q_{ij}(l)$. The generator matrix of $\{X_l(t),\ t \geqq 0\}$ is defined to be $Q(l) = [q_{ij}(l)]$.

It is important to note that the matrix $Q(l)$ is substochastic, i.e., its row sums are less than or equal to zero. The state $V \in \Omega^*$ is still an absorbing state for $X_l(t)$, except that due to the substochastic nature of $Q(l)$, the probability of absorption in $N$ is no longer 1. The physical meaning of this absorption probability is given in the following.

**Theorem 4.** $R(l) = P\{X_l(t) = N \text{ for some } t \geqq 0 | X_l(0) = 1\}$.

*Proof.* The generator matrix of $X_l(t)$ can be made stochastic (i.e., with zero row sums) by adding an extra absorbing state $a^*$ to the state space of $X_l(t)$. From the construction of $Q(l)$ it is clear that

$\{X_l(t) = N \text{ for some } t \geqq 0\} <=> \{$the message reaches node $v_i$ along the arc $(v_{i-1},v_i)$ for all $i = 2, \ldots ,k\}$
$\{X_l(t) = a^* \text{ for some } t \geqq 0\} <=> \{$the message reaches node $v_i$ along an arc other than $(v_{i-1},v_i)$ for some $i = 2, \ldots ,k\}$.

Theorem 4 now follows from Lemma 2. ∎

The above theorem provides a simple way of computing the optimality index of an $(s,t)$ path in the network. The algorithms to do so are given below.

**Backward Algorithm.** Define

$$v_i = P\{X_l(t) = N \text{ for some } t \geqq 0 | X_l(0) = i\}, \quad 1 \leqq i \leqq N. \tag{4.7}$$

The quantities $v_i$ satisfy the following equations

$$v_i = \sum_{j>i} v_j q_{ij}(l)/q_i(l) \qquad 1 \leqq i < N \tag{4.8}$$

where $q_i(l) = -q_{ii}(l)$. These equations can be solved in a backward fashion starting with $v_N = 1$ and computing $v_{N-1}, \ldots ,v_2,v_1$ in that order. Then we have $R(l) = v_1$.

**Example.** Consider the example network in Figure 1 and let $l$ be the path that visits nodes 1, 3, and 5. The generator matrix $Q(l)$ for the process $\{X_l(t),\ t \geqq 0\}$ is displayed in Table III. The Eqs. (4.8) for this case are as follows:

$$\begin{aligned}
v_7 &= 1,\\
v_6 &= \mu_6 v_7/(\mu_6 + \mu_7),\\
v_5 &= ((\mu_3 + \mu_5)v_6 + \mu_6 v_7)/(\mu_3 + \mu_5 + \mu_6),\\
v_4 &= \mu_2 v_6/(\mu_2 + \mu_4 + \mu_7),\\
v_3 &= (\mu_1 v_4 + \mu_2 v_6)/(\mu_1 + \mu_2 + \mu_7),\\
v_2 &= (\mu_3 v_3 + \mu_2 v_5)/(\mu_2 + \mu_3 + \mu_4),\\
v_1 &= (\mu_1 v_2 + \mu_3 v_3 + \mu_2 v_5)/(\mu_1 + \mu_2 + \mu_3).
\end{aligned} \tag{4.9}$$

The optimality index of $l$ is given by $R(l) = v_1$.

TABLE III. The generator matrix $Q(l)$ for the example network; The path $l$ visits nodes 1, 3 and 5.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| 1 | $-(\mu_1 + \mu_2 + \mu_3)$ | $\mu_1$ | $\mu_3$ | 0 | $\mu_2$ | o | 0 |
| 2 | 0 | $-(\mu_2 + \mu_3 + \mu_4)$ | $\mu_3$ | 0 | $\mu_2$ | o | 0 |
| 3 | 0 | 0 | $-(\mu_1 + \mu_2 + \mu_7)$ | $\mu_1$ | 0 | $\mu_2$ | 0 |
| 4 | 0 | 0 | 0 | $-(\mu_2 + \mu_4 + \mu_7)$ | 0 | $\mu_2$ | 0 |
| 5 | 0 | 0 | 0 | 0 | $-(\mu_3 + \mu_5 + \mu_6)$ | $\mu_3 + \mu_5$ | $\mu_6$ |
| 6 | 0 | 0 | 0 | 0 | 0 | $-(\mu_6 + \mu_7)$ | $\mu_6$ |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Forward Algorithm.** Define

$$\bar{v}_j = P\{X_l(t) = j \text{ for some } t \geqq 0 | X_l(0) = 1\},\ 1 \leqq j \leqq N. \tag{4.10}$$

Then $R(l) = \bar{v}_N$. The equations for $\bar{v}_j$ are given by

$$\bar{v}_j = \sum_{i<j} (q_{ij}(l)/q_i(l))\bar{v}_i. \tag{4.11}$$

The Eqs. (4.11) can be solved in a forward manner starting with $\bar{v}_1 = 1$ and computing $\bar{v}_2, \ldots, \bar{v}_N$ in that order.

## 5. CONDITIONAL ANALYSIS OF THE SHORTEST PATH

Let $l$ be an $(s,t)$ path in the network. Define

$$I(l) = \begin{cases} 1 \text{ if } l \text{ is the shortest } (s,t) \text{ path} \\ 0 \text{ otherwise.} \end{cases} \tag{5.1}$$

$I(l)$ is called the optimality indicator of the path $l$. It is obvious that $R(l) = P(I(l) = 1)$. Recall that $T$ is the length of the shortest $(s,t)$ path in the network. The distribution and moments of $T$ were obtained in Section 3. In this section we are interested in the following conditional quantities:

$$F_l(t) = P\{T \leqq t | I(l) = 1\}, \tag{5.2}$$

$$R_t(l) = P\{I(l) = 1 | T \leqq t\}, \tag{5.3}$$

$$\tau_k(l) = E(T^k | I(l) = 1). \tag{5.4}$$

To compute the above quantities, we first obtain

$$F(t,l) = P\{T \leqq t,\ I(l) = 1\}.$$

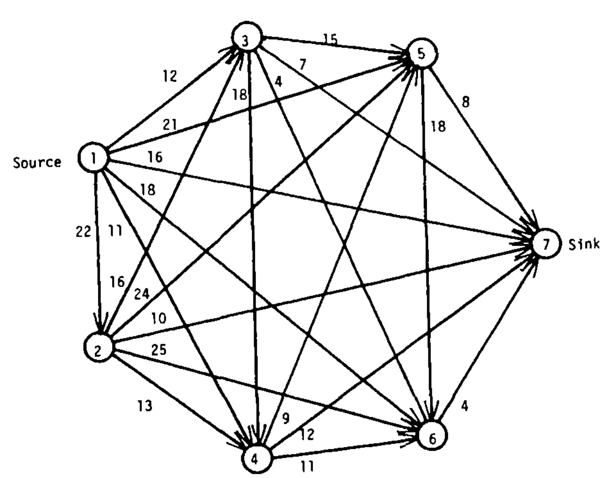


FIG. 3. Network 2. The numbers on arcs represent mean lengths.

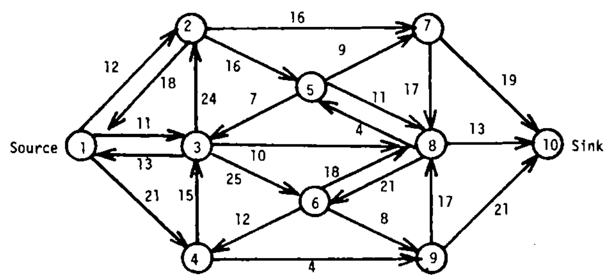


FIG. 4. Network 3. The numbers on arcs represent mean lengths.

Then we have

$$F_l(t) = F(t,l)/R(l), \tag{5.5}$$

$$R_t(l) = F(t,l)/F(t). \tag{5.6}$$

**Theorem 5.** Let $\{X_l(t),\ t \geqq 0\}$ be as defined in Section 4. We have

$$F(t,l) = P\{X_l(t) = N|X_l(0) = 1\}. \tag{5.7}$$

*Proof.* Let $X_l(0) = 1$. Equation (5.7) follows from the observation that the event $\{X_l(t) = N\}$ is equivalent to the event {path $l$ is the shortest *and* its length is less than or equal to $t$}. ∎

The above theorem implies that the algorithms of Subsection 4.1, when applied to the generator matrix $Q(l)$, yield $F(t,l)$. It is also clear that the algorithms of Subsection 4.2, when applied to $Q(l)$, yield $E(T^k,\ I(l) = 1)$. Hence conditional moments can be computed by using

$$\tau_k(l) = E(T^k;\ I(l) = 1)/R(l). \tag{5.8}$$

## 6. COMPUTATIONAL RESULTS

In this section we present computational results using the networks shown in Figures 1, 3, and 4 as examples. Backward algorithms of Subsections 3.1, 3.2, and 4 were implemented in Fortran 77.

The following information about the network size and its CTMC representation is given in Table IV for all the three networks:

1. Size of the state space
2. Number of positive elements in $Q$
3. The value of $q$ defined in Eq. (3.1.8)

TABLE IV. Network size descriptors.

| | Network 1 | Network 2 | Network 3 |
|---|---|---|---|
| (a) | 7 | 33 | 79 |
| (b) | 13 | 112 | 309 |
| (c) | 0.3500 | 1.0922 | 0.8972 |

TABLE V. Mean and standard deviation of the length of the shortest path.

| | Network 1 | Network 2 | Network 3 |
|---|---|---|---|
| $\mu$ | 11.9290 | 5.2487 | 19.2616 |
| $\sigma$ | 7.7930 | 3.6956 | 9.5970 |
| $M$ | 20 | 34 | 79 |
| $\alpha_M$ | 0.999997 | 0.999996 | 0.999995 |

The algorithms of Tsukiyama et al. [15] are used to generate the minimal cuts efficiently.

The means ($\mu$) and standard deviations ($\sigma$) of the lengths of the shortest paths in the three networks are displayed in Table V. In the numerical evaluation of the distribution, $F(t)$ is approximated by $F_M(t)$ of Eq. (3.1.14), where $M$ is chosen so that $\alpha_M \geqq 1 - 10^{-5}$. The values of $M$ and $\alpha_M$ are also shown in Table V for the three networks.

The distributions of the normalized random variables $(T - \mu)/\sigma$ are tabulated for the three networks in Table VI. The surprising feature of this table is that the normalized length of the shortest path seems to have more or less the same distribution for all the three networks. At this stage there is not sufficient data to see whether this is just a coincidence or is true in general.

The optimality indices for paths are tabulated in Tables VII, VIII, and IX. The paths are described by the sequence of nodes they visit. $R(l)$, $\tau(l)$, and $\sigma(l)$ give, respectively, the optimality index of $l$, the conditional mean length of $l$ given that it is the shortest path, and the conditional standard deviation of the length of $l$ given that it is the shortest path. Table VII contains the values of $R(l)$, $\tau(l)$, and $\sigma(l)$ for all paths in the network 1. Since networks 2 and 3 have a large number of paths, they are first arranged according to decreasing $R(l)$, and only the first 10 are displayed in Tables VIII and IX. Algorithms of Read and Tarjan [12] were used to enumerate all paths in the networks.

Table X contains the information about the CPU time needed to compute the distributions. It shows the time needed for "setup," i.e., setting up the rate matrix, and the time needed to actually compute the distribution for the three networks. It is clear that setting up the rate matrix, which involves generating all the cuts, is the most time-consuming step of the algorithms. Computation of mean, vari-

TABLE VI. $P((T - \mu)/\sigma \leqq x)$.

| $x$ | Network 1 | Network 2 | Network 3 |
|---|---|---|---|
| −3 | 0.0 | 0.0 | 0.0 |
| −2 | 0.0 | 0.0 | 0.0 |
| −1 | 0.1319 | 0.1390 | 0.1463 |
| 0 | 0.5825 | 0.5783 | 0.5608 |
| 1 | 0.8515 | 0.8507 | 0.8483 |
| 2 | 0.9548 | 0.9556 | 0.9588 |
| 3 | 0.9875 | 0.9877 | 0.9903 |

TABLE VII. Optimality indices and conditional analysis: network 1.

| Path $l$ | $R(l)$ | $\tau(l)$ | $\sigma(l)$ |
|---|---|---|---|
| 1 2 3 7 | 0.0597 | 14.9154 | 8.2827 |
| 1 2 3 4 7 | 0.0102 | 17.4507 | 8.7845 |
| 1 3 7 | 0.3515 | 11.6736 | 7.6681 |
| 1 3 4 7 | 0.0639 | 14.1202 | 8.1287 |
| 1 4 7 | 0.5147 | 11.3750 | 7.5866 |

ance, and distribution of $T$ takes a relatively small amount of time. These times do not include the compilation and linkage times. The times reported in Table X are for IBM 4341 system and varied a bit with the system load. Since there is no other known exact algorithm to evaluate these quantities, the speed of our algorithm cannot be compared. Work on comparing simulation algorithm is being carried out, and the comparative data shall be published later elsewhere.

## 7. CONCLUSIONS

In this paper we model the directed network with independent and exponentially distributed arc lengths by a continuous time Markov chain with upper triangular generator matrix. The state space of this Markov chain can be identified with the set of minimal cuts in the network. The *length* of the shortest path in the network is given by the *time* until absorption in this Markov chain. The special structure of this CTMC allows us to develop very simple algorithms for the exact shortest path analysis of the network. We have presented such algorithms for the distribution and moments of the length of the shortest path, optimality index of a path, and conditional analysis of the shortest path.

All the algorithms presented in this paper provide a new method of exact shortest path analysis of networks satisfying assumptions A1 and A2. They com-

TABLE VIII. Optimality indices and conditional analysis: network 2.

| Path $l$ | $R(l)$ | $\tau(l)$ | $\sigma(l)$ |
|---|---|---|---|
| 1 7 | 0.3280 | 3.9254 | 3.3584 |
| 1 6 7 | 0.1418 | 5.5023 | 3.5975 |
| 1 3 7 | 0.1282 | 5.4929 | 3.5673 |
| 1 4 7 | 0.0936 | 5.7349 | 3.6664 |
| 1 5 7 | 0.0754 | 5.7984 | 3.7011 |
| 1 2 7 | 0.0588 | 5.7572 | 3.6497 |
| 1 3 6 7 | 0.0566 | 6.2759 | 3.5695 |
| 1 4 6 7 | 0.0335 | 6.8621 | 3.8018 |
| 1 5 6 7 | 0.0110 | 6.9535 | 3.8536 |
| 1 3 5 7 | 0.0099 | 6.4700 | 3.5949 |

TABLE IX. Optimality indices and conditional analysis: network 3

| Path $l$ | $R(l)$ | $\tau(l)$ | $\sigma(l)$ |
|---|---|---|---|
| 1 3 8 10 | 0.2811 | 18.1189 | 9.3854 |
| 1 4 9 10 | 0.1968 | 18.0458 | 9.4583 |
| 1 2 7 10 | 0.1455 | 18.9292 | 9.5929 |
| 1 2 5 8 10 | 0.0534 | 20.4865 | 9.5199 |
| 1 4 9 8 10 | 0.0508 | 19.5888 | 9.2887 |
| 1 2 5 7 10 | 0.0448 | 20.9121 | 9.6614 |
| 1 3 6 9 10 | 0.0333 | 20.8472 | 9.6782 |
| 1 3 8 5 7 10 | 0.0326 | 21.9094 | 9.5756 |
| 1 2 7 8 10 | 0.0320 | 20.4643 | 9.5393 |
| 1 4 3 8 | 0.0225 | 19.2843 | 9.1734 |

plement the existing simulation and approximation techniques used in the shortest path analysis of stochastic networks. Thus, suppose one develops a simulation program to estimate the distribution of the length of the shortest path in a stochastic network with general arc length distributions. The validity of this program can be checked by running it for the special case of exponential arc lengths and comparing its results with the exact answers obtainable by the method described here. Also, goodness of any approximation can be tested by comparing its results with the exact ones obtained by the methods described here for networks with exponentially distributed arc lengths.

One limitation of the method described here is that the state space of the CTMC can grow exponentially with the network size. As the worst case example, for a complete network with $n$ nodes and $n(n-1)$ arcs, the size of the state space would be $2^{n-2}+1 (n \geqq 2)$. In practice, quite often, networks are sparse, in which case the size of the state space could still be manageable. One must also note that for very large networks *any* method of producing reasonably accurate answers will be prohibitively expensive.

It is possible to relax the independence assumption slightly. If there are two arcs $(u,v)$ and $(v,u)$ in the network, we need to assume only that the marginal distributions of their lengths are exponential; their lengths may be dependent. This is so since the message travels either on $(u,v)$ or $(v,u)$ but not on both. This implies that the methods of this paper are directly applicable to the undirected networks with independent and exponentially distributed arc lengths. One simply replaces every arc by two antiparallel arcs with the same length and uses the algorithms developed here.

TABLE X. CPU times for the three networks (in seconds)

| | Network 1 | Network 2 | Network 3 |
|---|---|---|---|
| Set Up | .19 | 0.96 | 1.97 |
| Computation of quantities in Tables V and VI | .05 | 0.13 | .26 |

## References


[1] V. G. Adlakha and G. S. Fishman. An improved conditional Monte Carlo technique for the stochastic shortest route problem. Technical Report No. 84-9, Curriculum in Operations Research and Systems Analysis, University of North Carolina, Chapel Hill, N.C. 27514 (1984).

[2] J. M. Burt and M. B. Garman. Conditional Monte Carlo: A simulation technique for stochastic network analysis. *Management Sci.* **18** (1971) 207–217.

[3] N. Deo and C. Pang. Shortest path algorithms: Taxonomy and annotation. *Networks* **14** (1984) 275–323.

[4] G. S. Fishman. Estimating critical path and arc probabilities in stochastic activity networks. TR-83/5, Curriculum in Operations Research and Systems Analysis, University of North Carolina, Chapel Hill, N.C. 27514 (1983).

[5] H. Frank. Shortest paths in probabilistic graphs. *Operations Res.* **17** (1969) 583–599.

[6] V. G. Kulkarni. A compact hash function for paths in PERT networks. *Operations Res. Lett.* **3**(3) (1984) 137–140.

[7] V. G. Kulkarni and V. G. Adlakha. Markov and Markov-regenerative PERT networks. *Operations Res.*, to appear.

[8] J. J. Martin. Distribution of the time through a directed acyclic network. *Operations Res.* **13** (1965) 46–66.

[9] P. B. Mirchandani. Shortest distance and reliability of probabilistic networks. *Comput. Operations Res.* **3** (1976) 347–355.

[10] A. A. B. Pritsker. Application of multi-channel queing results to the analysis of conveyor systems. *J. Indust. Eng.* **17** (1966) 14–21.

[11] J. S. Provan and M. O. Ball. Computing network reliability in time polynomial in the number of cuts. *Operations Res.* **32**(3) (1984) 516–526.

[12] R. C. Read and R. E. Tarjan. Bounds on backtrack algorithms for listing cycles, paths and spanning trees, *Networks* **5** (1975) 237–252.

[13] C. C. Sigal, A. A. B. Pritsker, and J. J. Solberg. The use of cutsets in Monte Carlo analysis of stochastic networks, *Math. Comp. Simulation* **21** (1979) 376–384.

[14] C. E. Sigal, A. A. B. Pritsker, and J. J. Solberg. The stochastic shortest route problem, *Operations Res.* **28** (1980) 1122–1129.

[15] S. Tsukiyama, I. Shirakawa, H. Ozaki, and H. Ariyoshi. An algorithm to enumerate all cutsets of a graph in linear time per cutset. *J. ACM* **27** (1980) 619–632.


Received July 1984
Accepted November 1985