Estimation of Network Reliability Using Graph Evolution Models

T. Elperin
Ben-Gurion University of the Negev, Beer Sheva
I. Gertsbakh
Ben-Gurion University of the Negev, Beer Sheva
M. Lomonosov
Ben-Gurion University of the Negev, Beer Sheva

Key Words — **Terminal reliability, Spanning tree, Lifetime simulation, Kruskal algorithm, Graph evolution, Monte Carlo method**

Reader Aids —
Purpose: Widen and advance state of the art.
Special math needed for explanations: Basic graph theory, probability and reliability theory.
Results useful to: System reliability analysts.

Abstract — **Monte Carlo techniques for estimating various network reliability characteristics, including terminal connectivity, are developed by assuming that edges are subject to failures with arbitrary probabilities and nodes are absolutely reliable. The core of our approach is introducing network time-evolution processes and using certain graph-theoretic machinery resulting in a considerable increase in accuracy for Monte Carlo estimates, especially for highly reliable networks. Simulation strategies and numerical results are presented and discussed.**

1. INTRODUCTION

Reliability of networks with randomly failing edges is a subject of extensive research. Several directions prevail in this research:

- algorithms for reliability computation [1–3,21,22]
- reliability estimation by means of simulation [6,8–11,14,16,17]
- constructing tractable lower and upper bounds on the network reliability [2–4,18–20].

This paper describes an approach to network reliability simulation based on an artificial time-evolution formulation of network failure reliability characteristics. This approach incorporates both simulation and analytic methods and has no difficulty when the edge failure probabilities are distinct. The aim of this work is to develop efficient simulation procedures. In particular, we are interested in reducing the relative error of network failure estimation for highly reliable networks.

A network $\mathfrak{N}$ is an undirected graph $G=(V,E)$, with node-set V, $|V|=n$, and edge-set E, $|E|=m$, whose spanning subgraphs (V,F) are classified as up (operational) and down (non-operational), subject to the reasonable monotonicity condition: If (V,F) is up then all subgraphs (V,F') are up, where $F \subseteq F'$.

The usual network reliability problem is: Suppose that edges of G are s-independently erased with probabilities $q(e)$, $e \in E$. Let F denote the set of non-erased edges; the subgraph (V,F) of G then appears with probability:

$$\Pr\{F\} = \Pi_{e \in F}\, \bar{q}(e)\, \Pi_{e \in E-F}\, q(e) \tag{1-1}$$

In practice this static model describes: i) systems without edge renewal, or ii) stationary regimes of renewable systems. In case ii, $q(e)$ is the equilibrium probability that a renewable edge e is down.

Notation

$R(\mathfrak{N},q)$	reliability of $\mathfrak{N}$
$\boldsymbol{q}$	the m-vector with coordinates $q(e) \in E$.
T	a fixed subset of V; the members of T are called terminals

Other, standard notation is given in "Information for Readers & Authors" at the rear of each issue.

The reliability of $\mathfrak{N}$ is defined as the probability $R(\mathfrak{N},q)$ that the random subgraph (V,F) is operating. As a principal example, we consider terminal reliability defined by: (V,F) is up if T lies in one component of (V,F). This network is denoted by (G,T).

We consider Monte Carlo simulation of network reliability based on time evolution modification of the static model. This modification leads to an important acceleration (variance reduction) of the Monte Carlo procedures, and guarantees boundedness of the relative error, irrespective of the value of graph unreliability (claim 6.2).

Section 2 presents a general statistical framework for reliability evaluation. Section 3 expresses the network reliability $R(\mathfrak{N},\boldsymbol{q})$ in terms of two different Markov processes: i) a destruction process (DP), and ii) a creation process (CP). Section 4 presents combinatorics for analyzing DP for the network (G,T). The central role belongs to the notion of maximal spanning tree, the Kruskal algorithm, and lemma 4.1. In the case of equal edge-failure probabilities, the DP leads to an efficient Monte Carlo sampling scheme studied earlier by Fishman [10]. Sections 5 & 6 describe a modification of CP based on the graph-theoretical notion of closure. Section 7 presents numerical results for a family of networks, with a comparison of several Monte Carlo approaches.

2. MONTE CARLO SAMPLING SCHEME

By the Monte Carlo method for evaluating a sum,

$$Z = \sum_{u \in U} z(u) \tag{2-1}$$

0018-9529/91/1200-0572$01.00

over a very large set U of "outcomes", we mean the following method. Introduce the probability distribution $p(u)$ on U and consider U as an *urn* from which a *ball* u can be drawn with probability $p(u)$. Also let Z denote the mean value of random variable $Y(u) = z(u)/p(u)$ by:

$$Z = \sum_{u \in U} p(u) \cdot \frac{z(u)}{p(u)} = \mathrm{E}\{Y\}. \tag{2-2}$$

The variance and coefficient of variation of Y are:

$$\mathrm{Var}\{Y\} = \sum_{u \in U} p(u) Y^2(u) - (\mathrm{E}\{Y\})^2 \tag{2-3}$$

$$\delta_Y = \left(\sum_{u \in U} p(u) Y^2(u) / \mathrm{E}\{Y\}^2 - 1 \right)^{1/2}. \tag{2-4}$$

From basic statistics we have —

Claim 2.1. Let $S = (u_1, u_2, \ldots, u_N)$ be the result of N s-independent choices from U, with probabilities $p(u)$. Then —

$$\hat{Y}_s = N^{-1} \sum_{i=1}^{N} Y(u_i)$$

is an unbiased estimate of $\mathrm{E}\{Y\}$, with variance and coefficient of variation equal:

$$\mathrm{Var}\{\hat{Y}_s\} = N^{-1} \cdot \mathrm{Var}\{Y\}$$

$$\delta_S = \delta_Y / \sqrt{N}. \qquad \square$$

Crude Monte Carlo

Consider a network $\mathfrak{N}$ with a graph $G = (V,E)$ and some operational (up) criterion. Realization of the above scheme for the set of subsets of E as the urn U, with $p(F)$, $F \subseteq E$, given by (1-1), and $Y(F) = 1$ when (V,F) is up, and 0 otherwise, are referred to as crude Monte Carlo (CMC) for evaluating $R(\mathfrak{N},\boldsymbol{q})$, or equivalently, for evaluating $Q(\mathfrak{N},\boldsymbol{q}) = 1 - R(\mathfrak{N},\boldsymbol{q})$. The variance of CMC is:

$$\mathrm{Var}_{\mathrm{CMC}}\{R\} = \mathrm{Var}_{\mathrm{CMC}}\{Q\} = R \cdot Q$$

and the relative error in evaluating Q on the basis of N s-independent experiments is:

$$\delta_{\mathrm{CMC}} = \sqrt{\frac{R}{Q}} \cdot \frac{1}{\sqrt{N}}.$$

The main deficiency of CMC is the unbounded growth of δ_{CMC} as Q approaches 0 (viz, for highly reliable networks). Various improvements of CMC have been suggested in order to reduce or eliminate this effect [6,8–11,14,17].

In this paper we offer another *urn* scheme for evaluating network reliability which guarantees finite relative error. The *balls* u in this scheme are the trajectories of a certain Markov process on the state space 2^E or its proper reduction, and the value $Y(u)$ of the random variable is the conditional probability of the up state for a given trajectory. Except for a special choice of the *urn* we suggest no changes to the above basic sampling scheme. This, however, provides performance of the Monte Carlo which is in general comparable with the existing advanced sampling techniques. In certain cases, such as highly reliable networks and dense graphs, the suggested method is definitely better.

3. GRAPH DESTRUCTION AND CREATION PROCESSES

Introduce an artificial time t and let $F(t)$ denote the set of edges existing at the instant t. Consider two types of graph evolution processes $G(t) = (V,F(t))$, $F(t) \subseteq E$, $t \geq 0$.

Destruction Process (DP)

Initially, at $t=0$, all edges are up: $F(0) = E$. Edges leave the set $F(t)$ s-independently, at random moments $\tau(e)$, with the Cdf, $\Pr\{\tau(e) \leq t\} = 1 - \exp(-\lambda(e)t)$. Let $\tau(\mathfrak{N})$ denote the random moment when $G(t)$ goes down. The Cdf of $\tau(\mathfrak{N})$ is $\Pr\{\tau(\mathfrak{N}) \leq t\} = \bar{R}(\mathfrak{N},\boldsymbol{q})$, where $\boldsymbol{q}$ is an m-vector with the components $q(e) = 1 - \exp(-\lambda(e)t)$, $e \in E$. The static model in the Introduction agrees with DP when the edge failure rates $\lambda(e)$ are chosen so that —

$$\lambda(e) = -\log(1 - q(e)), \tag{3-1}$$

and is realized at $t=1$.

Creation Process (CP)

Initially, at $t=0$, all edges are down: $F(0) = \emptyset$. The edges of G join $F(t)$ s-independently, at random moments $\tau(e)$, with the Cdf, $\Pr\{\tau(e) \leq t\} = 1 - \exp(-\lambda(e)t)$, $e \in E$, and operate forever. Let $\xi(\mathfrak{N})$ denote the moment when $G(t)$ goes up. The Cdf of $\xi(\mathfrak{N})$ is $\Pr\{\xi(\mathfrak{N}) \leq t\} = R(\mathfrak{N},\boldsymbol{q})$, with $q(e) = \exp(-\lambda(e)t)$. The static model agrees with CP when the edge birth rates $\lambda(e)$ are chosen so that,

$$\lambda(e) = -\log(q(e)), \tag{3-2}$$

and is realised at $t=1$.

For each of these processes consider an ordering (*permutation*) $w = (e_1, \ldots, e_m)$ of E specifying the order in which the edges are erased (in DP) or created (in CP). The probability of w is given by the well-known expression [7,17]:

$$\Pr\{w\} = \prod_{j=1}^{m} \frac{\lambda(e_j)}{\lambda(E_{j-1})}, \tag{3-3}$$

where $E_0 = E$, $E_i = E - e_1 - \ldots - e_i$, $1 \leq i \leq m-1$, and $\lambda(E_i) = \Sigma_{e \in E_i} \lambda(e)$.

For a given w, an edge e is called DP-*critical* if erasing it causes $G(t)$ to go down, and CP-*critical* if its creation causes

$G(t)$ to go up. The ordinal number of the critical edge in w is called the *critical number of* w and denoted by $[w]$, so that

$$[w] = \begin{cases} \min\{i: G-e_1-\ldots-e_i \text{ is down}\}, & \text{in DP} \\ \min\{i: (V, (e_1,e_2,\ldots,e_i)) \text{ is up}\}, & \text{in CP.} \end{cases} \quad (3\text{-}4)$$

Put $P(t|w) = \Pr\{\tau(\mathfrak{N}) \le t|w\}$ for DP, and $\Pr\{\xi(\mathfrak{N}) \le t|w\}$ for CP. By a well-known property of Markov processes [7], $P(t|w)$ is a convolution of exponential r.v.'s:

$$P(t|w) = \mathrm{Conv}_{1\le i\le [w]}\{1-\exp[-\lambda(E_i)t]\}. \quad (3\text{-}5)$$

$P(t|w)$ does not depend on the order of $e_{[w]+1},\ldots,e_m$ in the permutation w. The following notions are therefore reasonable.

Trajectories and Tails

An ordered subset $x = (e_1, e_2,\ldots,e_r)$ of E is called a *trajectory of* DP if $G-e_1-e_2-\ldots-e_i$ is up for $i < r$ and down for $i = r$; x is called a *trajectory of* CP if $(V,\{e_1,\ldots,e_i\})$ is down for $i < r$ and up for $i=r$. An ordered subset $y = (e_r, e_{r+1},\ldots,e_m)$ of E is called a *tail of* DP if $(V, \{e_i,e_{i+1},\ldots,e_m\})$ is down for $i > r$ and up for $i = r$; y is called a *tail of* CP if $G-e_i-e_{i+1}-\ldots-e_m$ is up for $i > r$ and down for $i=r$. □

The *critical number* $r=[w]$ divides a permutation $w=(e_1,\ldots,e_m)$ into the *trajectory* $\mathrm{tr}(w)=(e_1,e_2,\ldots,e_r)$ and the *tail* $\mathrm{tl}(w)=(e_r,e_{r+1},\ldots,e_m)$. A trajectory x can be identified with the set (bundle) of permutations w satisfying $\mathrm{tr}(w)=x$, thus —

$$P(t|x) = P(t|w) \text{ for } w \text{ satisfying } \mathrm{tr}(w) = x,$$

$$\Pr\{x\} = \sum_{w:\mathrm{tr}(w)=x} \Pr\{w\} = \prod_{i=1}^{r} \frac{\lambda(e_i)}{\lambda(E-e_1-\ldots-e_{i-1})}. \quad (3\text{-}6)$$

By the total probability formula,

$$P(t) \equiv \sum_x P(t|x)\Pr\{x\} = \begin{cases} \Pr\{\tau(\mathfrak{N}) \le t\} = \bar{R}(\mathfrak{N},\boldsymbol{q}), & \text{for DP} \\ \Pr\{\xi(\mathfrak{N}) \le t\} = R(\mathfrak{N},\boldsymbol{q}), & \text{for CP} \end{cases} \quad (3\text{-}7)$$

$\boldsymbol{q} = \{q(e); e \in E\}$ is the vector of edge failure probabilities,

$$q(e) = \begin{cases} 1-\exp[-\lambda(e)t], & \text{for DP} \\ \exp[-\lambda(e)t], & \text{for CP.} \end{cases}$$

The sum at the left in (3-7) is over all trajectories of the corresponding process.

The Monte Carlo scheme based on generating trajectories x and exactly computing $P(t|x)$ is characterised by the variance:

$$\mathrm{Var}_P = \sum_x \Pr\{x\}\, P^2(t|x) - P(t)^2. \quad (3\text{-}8)$$

The following obvious variance decomposition reveals the gain in accuracy provided by DP or CP with respect to CMC (at the expense of more complex computations).

$$\mathrm{Var}_{\mathrm{CMC}} = \mathrm{Var}_P + \sum_x \Pr\{x\} \cdot P(t|x)\, \bar{P}(t|x). \quad (3\text{-}9)$$

Similarly, a tail y can be identified with the bundle of permutations w satisfying $\mathrm{tl}(w)=y$. Its probability is:

$$\Pr\{y\} = \left(\prod_{i=r}^{m} \frac{\lambda(e_i)}{\lambda(e_i)+\ldots+\lambda(e_m)}\right) \cdot \int_0^\infty \exp[-t] \prod_{i=1}^{r-1} (1-\exp[t\cdot\lambda(e_i)/\lambda(y)])dt. \quad (3\text{-}10)$$

The Cdf of the critical moment, given y, is:

$$P(t|y) = \frac{\int_0^{\lambda(y)\cdot t} \Psi(s)ds}{\int_0^\infty \Psi(s)ds} \quad (3\text{-}10')$$

$$\Psi(s) \equiv \exp[-s] \prod_{e\notin y} (1-\exp[-s\lambda(e)/\lambda(y)])$$

$$\lambda(y) = \lambda(e_r) + \lambda(e_{r+1}) + \ldots + \lambda(e_m).$$

In the important case of equal edge-failure probabilities ($\lambda(e) = \lambda$ for all $e \in E$) we have —

$$\Pr\{w\} = \frac{1}{m!}, \quad \Pr\{x\} = \frac{(m-r)!}{m!}, \quad \Pr\{y\} = \frac{(r-1)!}{m!}. \quad (3\text{-}11)$$

Now we describe generating permutations, trajectories and tails with their "natural" probabilities, as they appear in the corresponding process.

Generating permutations

For each edge e generate a value $b(e)$ of r.v. $\tau(e)$, the lifetime of e. Then the desired permutation $w=(e_1,e_2,\ldots,e_m)$ is induced by the inequalities:

$$b(e_1) < b(e_2) < \ldots < b(e_m). \quad (3\text{-}12)$$

This method is equivalent to drawing a permutation w from the *urn* (see section 2) of all $m!$ possible permutations of E with probability $p(w)$ given by (3-3). □

Generating trajectories

A trajectory $x=(e_1,e_2,\ldots,e_r)$ is generated by sequentially choosing e_1 from E with probability $\lambda(e_1)/\lambda(E)$, e_2 from $E-e_1$ with probability $\lambda(e_2)/\lambda(E-e_1)$, etc, until the critical

edge is generated. This is equivalent to drawing a trajectory from an *urn* with probability (3-6).

Sequentially generating tails in the reverse order e_m, $e_{m-1},\ldots$, with natural probabilities (3-10) is practically intractable, except for the case of equal edge-failure probabilities. In the latter case, tails are generated in the same sequential manner as trajectories. □

4. IDENTIFYING THE CRITICAL EDGE OF DP FOR $\mathfrak{N}=(G,T)$

Notation

G_k	component of $G-e_1-e_2-\ldots-e_{k-1}$ that contains T
D	some spanning tree of G_k
$D(T)$	the minimal subtree of D that contains T
D_w	the lexicographically maximal spanning tree with respect to w
$e(w,T)$	the junior edge of the subtree $D_w(T)$.

Consider generating a trajectory $x=(e_1,e_2,\ldots)$ of DP. After a current edge e_k is erased, we need only to check if the terminal-set T is connected by $G-e_1-e_2-\ldots-e_k$. Surely, the choice of e_k can always be restricted to the edge-set of G_k. The possibilities are:

i. $e_k \notin D$. Then $G_{k+1}=G-e_k$, and obviously G_{k+1} contains T; D might be preserved.

ii. otherwise. Find an edge e of G_k-e_k connecting the components of $D-e_k$. If e exists, then $G_{k+1}=G-e_k$, G_{k+1} connects T, and we can put $D:=D-e_k+e$; otherwise (e does not exist), consider the components of $D-e_k$. If one of them, say D' contains the entire T, then $G_{k+1}=G_k(D')$ which is the subgraph of G_k induced by the vertices of D', G_{k+1} connects T, and we can put $D:=D'$; otherwise T is disconnected in G_k-e_k, so that the critical number is k, and $(e_1,e_2,\ldots,e_k)$ is the desired trajectory. □

An important fact is that when an entire permutation w is available, the critical edge can be determined by using exactly one special spanning tree D, as shown in lemma 4.1 below.

For a spanning tree D, $D(T)$ is the union of the chains of D between all pairs of terminals. An edge permutation w induces the following lexicographic order among the spanning trees: $D'>D''$ when the senior edge of $D'-D''$ is greater than that of $D''-D'$. Then D_w is exactly the tree constructed by the famous Kruskal algorithm [15], with the input w.

Lemma 4.1. Let $b(e)$ be the lifetime of edge e in DP, and $w=(e_1,e_2,\ldots,e_m)$ be the edge permutation induced by inequalities $b(e_1)<b(e_2)<\ldots<b(e_m)$. Then (G,T) fails when $e(w,T)$ fails, so that $e(w,T)$ is the critical edge of w.

Proof. Let k be the ordinal number of $e(w,T)$ in w, and put $F_i=E-e_1-\ldots-e_i$, $i=1,\ldots,k$. For $i<k$, (V,F_i) contains $D_w(T)$ and thus connects T. We show that T is disconnected in (V,F_k). Consider the components D', D'' of D_w-e_k. By definition of e_k, both D', D'' contain terminals. When e_k fails, the subset C of E connecting D' & D'' becomes empty. If it did not, then there would exist e_j in C, with $j>k$, so that the spanning tree $D_w-e_k+e_j$ is lexicographically greater than D_w—which is a contradiction. □

Example

Figure 1a presents the graph called *dodecahedron* [12]. It has 20 nodes and 30 edges. The double-circled nodes 1,3,17 are the terminals. The numbers near edges specify the edge lifetimes, $b(e)$. The corresponding tree D_w is shown on figure 1b; its bold part is $D_w(T)$. The network lifetime is 19, while the overall connectivity is lost earlier, at $t=6$.

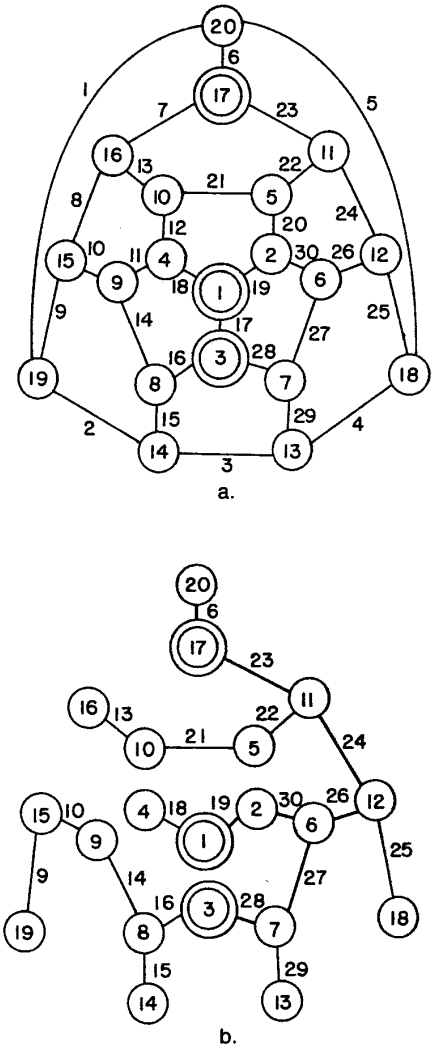


Figure 1. The Dodecahedron and Its Maximal Spanning Tree

Simulation Strategy

The following Monte Carlo simulation algorithm is suggested by lemma 4.1.

1. Generate N permutations w_j $(j=1,\ldots,N)$ as described in (3-12).

2. For each $j=1,\ldots,N$, determine $[w_j]$ using the Kruskal algorithm; the trajectory $\mathrm{tr}(w_j)$ is thus identified.

3. Compute $P(t|w_j)$ by (3-5).

4. Compute the estimate of network failure probability as the sample average of $P(t|w_j)$ $(j=1,\ldots,N)$, see (2-2); compute the corresponding sample variance. □

The Kruskal algorithm used in the above simulation starts from e_m and terminates with e_r, ie, it deals only with the tail of the given permutation. In principle, this property could be exploited by sequentially generating the tail of a permutation w_j, applying (3-10) & (3-10′). Unfortunately, however, these formulas are too complicated for straightforward calculations. The situation is, however, much easier in the important case of equal edge-failure probabilities, as in (3-11). The reliability simulation of $\mathfrak{N}=(G,T)$ with equal edge-failure probabilities is considered by Fishman [8,10]. In [10], the simulation method is based on generating tails, and lemma 4.1 was used, but without being formulated explicitly.

The case of equal edge-failure probabilities is extremely favorable for the network Monte Carlo. Indeed, in this case all permutations have the same probability, $1/(m!)$. On the other hand, assuming, without loss of generality, $\lambda(e)=1$ for all $e \in E$, the convolution (3-5) can be considerably simplified. Indeed, in the theory of order statistics [5] it is well-known that $1-\exp(-(m-i+1)t)$ is the Cdf of the r.v. $V_i=\tau_{(i)}-\tau_{(i-1)}$, where $\tau_{(i)}$ is order statistic i for the sample of m i.i.d. r.v.'s $\tau_i \sim \exp(1)$, $\tau_0=0$. Moreover, the r.v.'s V_i, $i=1,\ldots,m$, are s-independent. Thus the r.h.s. of (3-5) is:

$$\mathrm{Conv}_{1\le i\le [w]}\{1-\exp[-(m-i+1)t]\} = \Pr\{\tau_{[w]}\le t\}. \quad (4\text{-}1)$$

The Cdf $H_r(t)$ of the r-th order statistic is given by a well-known formula

$$H_r(t) = \Pr\{\tau_{(r)}\le t\} = \mathrm{binfc}(r;1-e^{-t},m) \quad (4\text{-}2)$$

Thus (3-7) acquires the form:

$$\bar{R}(\mathfrak{N},q) = \sum_{r=1}^{m} A(r)H_r(t), \quad (4\text{-}3)$$

$$A(r) = (\text{number of permutations with } [w] = r)/m!.$$

This presentation is used in [10] as a basis for a Monte Carlo sampling scheme. A remarkable property of (4-3) is that the relevant combinatorics of $\mathfrak{N}=(G,T)$, expressed by the numbers $\{A(r), r=1,\ldots,m\}$, are totally separated from the probabilities contained in the functions $H_r(t)$. These functions are standard and always available for any value of t. So, the Monte Carlo simulation efforts should be turned to obtaining the distribution $\{A(r), r=1,\ldots,m\}$. It is reasonable to call it the *internal distribution* (ID) of the network (G,T). As an illustration, tables 1 & 2 present the IDs of several complete graphs with $T=V$, and for the dodecahedredon with various T's. obtained by simulation.

TABLE 1
Simulated ID's of Complete Graphs, $N = 10^6$

	$A(r)$'s for					
r	K_5	K_6	K_7	K_8	K_9	K_{11}
4	0.023810					
5	0.095238	0.002050				
6	0.285714	0.010155	0.000127			
7	0.595238	0.029820	0.000761	0.000006		
8		0.071937	0.002688	0.000043	0.000002	
9		0.155212	0.007294	0.000165	0.000002	
10		0.298834	0.016465	0.000529	0.000016	0.000000
11		0.431992	0.033110	0.001451	0.000029	0.000000
12			0.061712	0.003070	0.000087	0.000000
13			0.108629	0.006247	0.000259	0.000000
14			0.181284	0.011650	0.000549	0.000000
15			0.277702	0.020218	0.001019	0.000000
16			0.310228	0.034246	0.001869	0.000000
17				0.055661	0.003360	0.000006
18				0.086821	0.005759	0.000005
19				0.130692	0.009628	0.000013
20				0.186388	0.014864	0.000041
21				0.241062	0.023444	0.000065
22				0.221751	0.035028	0.000117
23					0.051427	0.000216
24					0.073990	0.000342
25					0.103387	0.000499
26					0.138312	0.000757
27					0.177942	0.001205
28					0.201261	0.001785
29					0.157776	0.002634
30						0.003678
31						0.005431
32						0.007685
33						0.010677
34						0.014367
35						0.019461
36						0.026335
37						0.035125
38						0.045961
39						0.059512
40						0.075767
41						0.093734
42						0.112722
43						0.130512
44						0.140335
45						0.130172
46						0.080841

5. CREATION AND MERGING PROCESSES FOR $\mathfrak{N}=(G,T)$

A closer look at the performance of the Kruskal algorithm reveals that on each step of constructing a maximal spanning tree, there can be identified a set of *irrelevant* edges whose future appearance does not affect the time $\xi(\mathfrak{N})$. These are exactly the edges complementing the existing part of the tree to its graph-theoretical closure.

The *closure* of a subset F of E consists of F and all edges of G whose ends lie in the same component of the spanning subgraph (V,F). A subset F is *closed* if it coincides with its closure.

TABLE 2
Simulated ID's for the Dodecahedron, $N = 10^5$

	$A(r)$		
r	$T = V$	$T = \{1, 7, 8, 11, 16\}$	$T = \{1, 20\}$
3	0.00476	0.00130	0.00039
4	0.01637	0.00429	0.00170
5	0.03366	0.00959	0.00374
6	0.05991	0.01928	0.00752
7	0.09559	0.03486	0.01464
8	0.13872	0.05867	0.02600
9	0.17746	0.09259	0.04515
10	0.20031	0.13270	0.07215
11	0.17794	0.17232	0.10635
12	0.09528	0.17733	0.13207
13		0.13625	0.13642
14		0.08407	0.12568
15		0.04464	0.10235
16		0.02017	0.07963
17		0.00784	0.05561
18		0.00284	0.03731
19		0.00091	0.02361
20		0.00028	0.01428
21		0.00005	0.00836
22		0.00002	0.00401
23			0.00183
24			0.00089
25			0.00029
26			0.00002

For example, closing the set of bold edges of the dodecahedron shown in figure 1b adds the edge $e=(1,3)$.

The closure operation enables us to deal simultaneously with thicker bundles of permutations than in the original DP or CP. A serious obstacle for using this approach in the DP for the network (G,T) is the stochastic properties of tails, expressed by (3-10), (3-10′). This obstacle never appears for (G,T) in CP, as is shown below [3,19].

In what follows, the notion of *regular partition* of V plays the central role. Given a graph $G=(V,E)$, a partition $g=\{X_1,X_2,\ldots,X_r\}$ of V, where $X_i \cap X_j = \emptyset$ for $i \neq j$, and $\cup_{i=1}^{r} X_i = V$, is called *regular* (with respect to G) if each induced subgraph $G(X_i)$ is connected. Arbitrary set F of edges generates a regular partition $\langle F\rangle = \{X_1,X_2,\ldots,X_r\}$ where X_i are the components of the spanning subgraph (V,F) (including isolated nodes, if any). Subsets F' and F'' are *equivalent* if $\langle F'\rangle = \langle F''\rangle$, and identify every regular partition g with the class of subsets F of E satisfying $\langle F\rangle = g$. Clearly, each such class is the collection of subsets of edges with a common closure. For every regular g, let its components be referred to as *super-nodes* and $E(g)$ denote the set of *external* edges (the edges between distinct super-nodes). Put $\lambda(g)=\Sigma_{e\in E(g)}\lambda(e)$.

Consider the set $L(G)$ of all regular partitions of V, partially ordered by the relation: $g' < g''$ when g'' is obtained by merging components of g'.

Suppose that a state $F(t)$ of CP (see section 3) belongs to an equivalence class $g(t) = g$. Clearly, the time $F(t)$ spends in g has the Cdf $1-\exp[-\lambda(g)\cdot t]$. On leaving g, $F(t)$ jumps in one of direct successors of g, say g', obtained by merging exactly two super-nodes of g, and chosen with probability $(\lambda(g) - \lambda(g'))/\lambda(g)$. The above is summarized in the following.

Claim 5.1.

i. $g(t) = \langle F(t)\rangle$ is a Markov process on $L(G)$;
ii. the time spent by $g(t)$ in a state g is distributed as $\exp(\lambda(g))$;
iii. the transition $g \to g'$ has the probability $\Pr\{g'|g\} = (\lambda(g) - \lambda(g'))/\lambda(g)$, when g' is a direct successor of g, and 0 otherwise.

For additional details see [3,19]. In the following, $g(t)$ is referred to as Merging Process, MP.

Example

Figure 2 presents $L(K_4)$, the set of all regular partitions g of the complete 4-node graph, "naturally" stratified into 4 levels according to the number of super-nodes in g. The arrows show the direct successions in $L(K_4)$, thus forming the transition graph of the Markov process $g(t)$. The members of $L(K_4)$ are represented by circles; the corresponding closed spanning subgraph is drawn in each circle. Let $T = \{2,3\}$. The double circles correspond to the partitions for which T lies in one super-node.

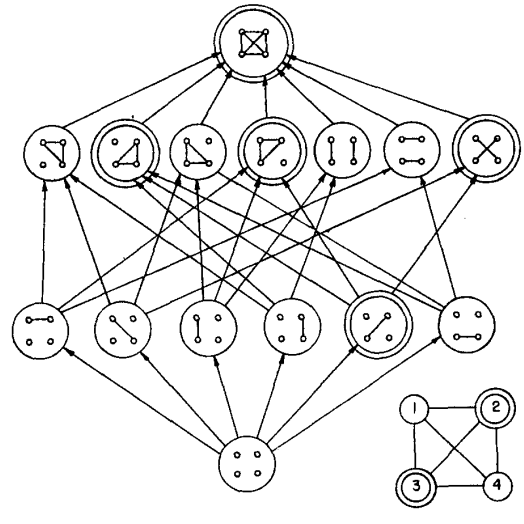


Figure 2. Transition Diagram for a Markov Process Whose States Are All Regular Partitions of a Complete 4-Node Graph

Given a network (G,T), we say that $g \in L(G)$ is *up* if all terminals lie in one super-node of g. A *trajectory* of $g(t)$ is a sequence $u=(g_0,g_1,\ldots,g_r)$ of regular partitions where g_0 is the trivial partition into singletons; g_i is a direct successor of g_{i-1} for $i=1,\ldots,r$; and r is the first i such that g_i is up. In general, trajectories have distinct lengths, so that r depends on u. The probability of u is:

$$\Pr\{u\} = \prod_{i=0}^{r-1} \frac{\lambda(g_i)-\lambda(g_{i+1})}{\lambda(g_i)}. \quad (5\text{-}1)$$

The conditional distribution function of $\xi(\mathfrak{A})$ along u is

$$P(t|u) = \mathrm{Conv}_{0\le i\le r-1}\{1-\exp[-\lambda(g_i)t]\}. \quad (5\text{-}2)$$

Finally, the Cdf of $\xi(\mathfrak{A})$ is

$$\Pr\{\xi(\mathfrak{A})\le t\} = \sum_{u\in U} \Pr\{u\}P(t|u), \quad (5\text{-}3)$$

where U is the set of all trajectories of $g(t)$.

Returning to the initial creation process $F(t) \subseteq E$, we see that a trajectory $x=(e_1,e_2,\dots)$ of CP produces a uniquely defined trajectory of MP which we denote by $\langle x\rangle$. We say that trajectories x', x'' of CP are equivalent if $\langle x'\rangle = \langle x''\rangle$. Thus, a trajectory u of MP represents the class of trajectories of CP satisfying $\langle x\rangle = u$; we write it as $x \in u$.

For $x \in u$ one has —

$$\Pr\{x|u\} = \Pr\{x\}/\Pr\{u\}$$

$$P(t|u) = \sum_{x\in u} \Pr\{x|u\} \cdot P(t|x).$$

The Monte Carlo scheme based on generating trajectories of MP and exactly computing $P(t|u)$ using (5-2) has

$$\mathrm{Var}_{\mathrm{MP}} = \sum_{u\in U} \Pr\{u\}P(t|u)^2 - P(t)^2 \quad (5\text{-}4)$$

$$P(t) \equiv \Pr\{\xi(\mathfrak{A}) \le t\}.$$

Comparison with CMC and CP is based on the expansions:

$$\mathrm{Var}_{\mathrm{CMC}} = \mathrm{Var}_{\mathrm{MP}} + \sum_{u} \Pr\{u\}P(t|u)\ \bar{P}(t|u), \quad (5\text{-}5)$$

$$\mathrm{Var}_{\mathrm{CP}} = \mathrm{Var}_{\mathrm{MP}} + \sum_{u} \Pr\{u\}\left(\sum_{x\in u} \Pr\{x|u\}\ P(t|x)^2 - P(t|x)^2\right). \quad (5\text{-}6)$$

The second term in the r.h.s. of (5-6) is the part of $\mathrm{Var}_{\mathrm{CP}}$ eliminated by the state space reduction when CP was transformed into MP.

In the example of figure 2 the trajectories of $g(t)$ are the paths starting in $g_{\bar{0}}$ and terminating in doubled circles.

Simulation strategy

For estimating the sum (5-3), the sampling scheme of section 2 is applied, with "natural" probabilities $p(u) = \Pr\{u\}$ given by (5-1).

Generating a trajectory of $g(t)$

Start from $g_o = \{X_1,\dots,X_n\}$, $|X_i| = 1$. At step r one has a sequence $g_0,g_1,g_2,\dots,g_r$, with $g_r = \{X_1,\dots,X_{n-r}\}$, and a list E_r of the edges between distinct components X_i. If g_r is up, then stop; otherwise draw an edge from E_r and form g_{r+1} by merging the two components connected by this edge.

After a trajectory $u=(g_0,g_1,\dots,g_r)$ is formed, the conditional Cdf $P(t|u)$ is computed as the convolution of functions $(1-\exp[-\Lambda_i t])$ where $\Lambda_i = \Sigma_{e\in E_r} \lambda(e)$, $i=0,\dots,r-1$.

Since all Λ_i are distinct (in fact, $\Lambda_0 > \Lambda_1 > \dots \Lambda_{r-1}$), the above convolution is a linear combination of the exponents $\exp[-\Lambda_i t]$, $i=0,\dots,r-1$, whose coefficients are homogenous functions of Λ_i, of order 0. The following recurrent procedure has the complexity $O(r^2)$:

Assume $\mathrm{Conv}_{r-k\le i\le r-1}\{1-\exp[-\Lambda_i t]\}$

$$= 1 - \sum_{i=1}^{k} A_{k,i} \exp[-\Lambda_{r-i}t]$$

with $\sum_{i=1}^{k} A_{k,i} = 1.$

(To start with, we have $A_{1,1} = 1$).

Then —

$$A_{k+1,i} = A_{k,i} \cdot \frac{\Lambda_{r-k-1}}{\Lambda_{r-k-1}-\Lambda_{r-i}},\ i=1,\dots,k;$$

$$A_{k+1,k+1} = 1 - \sum_{i=1}^{k} A_{k+1,i}.$$

6. COMPLEXITY OF THE MP-MONTE CARLO

For evaluating the computational complexity of the Monte Carlo scheme based on MP consider two problems.

A. Estimating $Q = \bar{R}(\mathfrak{A},\boldsymbol{q})$ for particular value of the vector $\boldsymbol{q}$ with a given mean relative error δ.

B. Estimating $Q = \bar{R}(\mathfrak{A},\boldsymbol{q})$ for a 1-*parameter* family q_t of the form $q_t(e) = q_1(e)^t = \exp[-\lambda(e)t]$, $t \in (t_1,t_2,\dots,t_k)$ with a given mean relative error δ.

The following statements can be easily established.

Claim 6.1. The complexity of one simulation run for A is $O(n^2)$. For B, the values of $Q(t_1|u),\dots,Q(t_k|u)$ are available in one simulation run: its complexity is $O(n^2) + O(k\cdot n)$.

Let $\delta_{\mathrm{MP}}(Q)$ denote the coefficient of variation in the MP Monte Carlo scheme:

$$\delta^2_{\mathrm{MP}}(Q) = \frac{\mathrm{Var}_{\mathrm{MP}}}{Q^2} = \frac{\Sigma_{u\in U} \Pr\{u\}Q(t|u)^2}{Q(t)^2} - 1.$$

Then the complexity of the MP Monte Carlo for both A and B is:

$$\frac{\delta^2_{\mathrm{MP}}(Q)}{\delta^2} \cdot O(n^2). \qquad \square$$

A pleasant feature of MP is given by the following statement.

Claim 6.2. For a given n and a given operational criterion the coefficient of variation $\delta^2_{MP}(Q)$ is bounded uniformly for all values of $\lambda(e)$, $e \in E$ and $0 \le t \le \infty$.

Proof. Since the λ's appear only in the products $\lambda(e)\cdot t$ and in 0-homogeneous form, it is sufficient to prove that $\lim_{t\to\infty} \delta^2_{MP}(Q)$ exists and is bounded in the unit ball $\Sigma_e \lambda^2(e) \le 1$.

Consider the lattice L of all partitions of the node-set V (states), and let UP and DN denote the sets of up and down states respectively. A trajectory of MP is a sequence $u = (g_0,g_1,\dots,g_r)$, where $g_0,\dots,g_{r-1} \in$ DN, $g_r \in$ UP (in this formulation trajectories with zero probabilities are permitted). Then —

$$Q(t|u) = 1-\mathrm{Conv}_{0\le i\le r-1}\{1-\exp[-\lambda(g_i)t]\}$$

$$= A(u)\exp[-\lambda(g_{r-1})t]\ (1+o(1))$$

as $t \to \infty$ (since $\lambda(g_o) > \dots > \lambda(g_{r-1})$). Define $\mu(\mathfrak{N}) = \min\{\lambda(g) : g \in \text{DN with a direct successor in UP}\}$ and $U_0 = \{u = (g_0,\dots,g_{r-1},g_r) \in U : \lambda(g_{r-1}) = \mu(\mathfrak{N})\}$. Then —

$$\delta^2_{MP}(Q) \le \frac{\max_{u\in U_0} A(u)\cdot \exp[-\mu(\mathfrak{N})t]}{Q(t)} - 1$$

[for t large enough]

$$\to \frac{\max_{u\in U_0} A(u)}{A(\mathfrak{N})} - 1, \text{ as } t \to \infty,$$

$$A(\mathfrak{N}) = \sum_{u\in U_0} \Pr\{u\}\, A(u).$$

This limit is a continuous function of $\lambda(e)$, $e \in E$, and the assertion follows. □

In the particular case of complete graph $G=K_n$, with equal edge-failure rates, and all-terminal connectivity as an operational criterion, it can be shown that $\delta^2_{MP}(Q) \le 1$ for all n. The following seemingly non-trivial question is then reasonable and important.

Question. Is there a universal constant A such that $\delta_{MP}(Q) \le A$ for all n, all possible λ's and $0 \le t < \infty$?

7. SIMULATION RESULTS

In order to evaluate the performance of the Monte Carlo schemes based on edge destruction & creation processes, a series of experiments has been done for several networks. Network failure probability $Q=\bar{R}\{\mathfrak{N},q\}$ was estimated along the lines of the simulation strategies for DP & MP described in sections 4 & 5. As a performance measure of the simulation method we take:

$$W = (\text{Variance}) \times (\text{CPU-time in sec for 1000 replications}). \quad (7\text{-}1)$$

The values W_{DP} & W_{MP} for DP & MP, respectively, were compared with the corresponding W_{CMC} for the crude Monte Carlo (CMC) and for some methods in [6,9,14,16].

The CMC was based on erasing edge e with probability $q(e)$, $e \in E$, and on checking the terminal connectivity of the resulting subnetwork. When no edges fail, the CMC simulation program skips the terminal connectivity check [6]. The set union algorithm of Hopcroft & Ullman [13] was applied for the connectivity check. For both DP & CP, the ratios W_{CMC}/W_{DP} & W_{CMC}/W_{MP} were computed.

From the accuracy point of view, the principal parameter is the relative error:

$$\delta = (\text{Variance})^{1/2}/(\text{Network failure probability}). \quad (7\text{-}2)$$

The variance reduction factor with respect to CMC was computed for MP as, $\eta_{MP} = \delta^2_{CMC}/\delta^2_{MP}$.

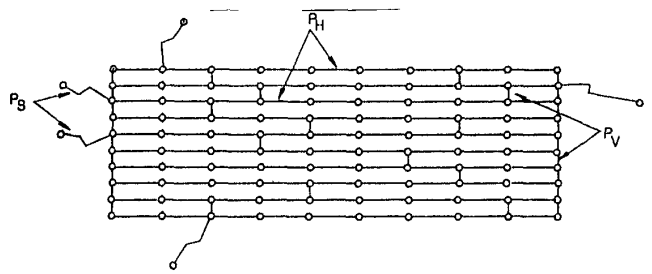
The simulation results are based on $N=10^4$ replications. The following networks were chosen for numerical experiments:

1. The Easton-Wong [6] communication network with 105 nodes and 127 edges, with the all-terminal connectivity as the operational criterion.
2. The dodecahedron (figure 1a) with the s-t connectivity as the operational criterion for $s=1$ and $t=20$ [9].
3. The dodecahedron with the all-terminal connectivity as the operational criterion.
4. A family of complete graphs K_{10}, K_{15}, K_{20}, K_{25}, K_{30} with the all-terminal connectivity as the operational criterion.

For networks #1 & #2, the performance of our methods is compared with that of alternative methods [6,9]. Table 3 presents simulation results for the Easton-Wong network, table 4 for the dodecahedrons, and table 5 for complete graphs. The following conclusions can be drawn from analyzing tables 3 - 5.

1. DP is not competitive with MP, in terms of both variance reduction factor and in the W performance measure. Relative to CMC, DP has good performance parameters for non-dense and very reliable graphs, as shown in table 4. The reasonable application field of DP is networks with equal edge-failure probabilities. In that case, one simulation run results in estimating the ID of (G,T) and serves for any q value; see section 5.
2. MP is very efficient for highly reliable networks and dense graphs; see lines 4,5 and 9,10 in table 4, and lines 3,4,5 in table 5. The performance of MP increases when the network reliability approaches 1.
3. The suggested MP algorithm needs no extra modifications to include the cases of distinct edge failure probabilities and various partition-dependent operational criteria.
4. The complexity of MP-evaluation of $Q(t)$ for several t_i $(i=1,\dots,k)$ is essentially the same as for one value of t; see section 6. In particular, all 5 values of Q in table 4 can be obtained in a single simulation experiment of 10^4 replications; this would increase the W-performance ratio by a factor of 5.

TABLE 3
Simulation Results of the Easton — Wong Network [6]
(all-terminal connectivity)



Edge failure probabilities			Network failure probability						
P_H	P_V	P_S	$Q^{*)}$	$\delta^{\%}_{DP}$	$\frac{W_{CMC}}{W_{DP}}$	$\delta^{\%}_{MP}$	η_{MP}	$\frac{W_{CMC}}{W_{MP}}$	$\frac{W_{CMC}^{**}}{W_{SD}}$
0.02	0.01	0.001	0.0438	0.9	<1	0.6	67	3.4	2.4
0.005	0.01	0.0005	0.00538	1.6	1.5	1.2	135	5.4	7.6

*)Estimated by MP
**)W_{CMC}/W_{SD} is the performance ratio of the CMC versus the Easton-Wong's sequential destruction method [6].

TABLE 4
Simulation Results for the Dodecahedron
($s-t$ connectivity; s = "1", t = "20", see Fig. 1,a.)

$q(e)$	$Q^{*)}$	$\delta^{\%}_{DP}$	$\frac{W_{CMC}}{W_{DP}}$	$\delta^{\%}_{MP}$	η_{MP}	$\frac{W_{CMC}}{W_{MP}}$	$\frac{W_{CMC}}{W_{DAG}}$ {1}*	$\frac{W_{CMC}}{W_{SC}}$ {2}*	$\frac{W_{CMC}}{W_B}$ {3}*	$\frac{W_{CMC}}{W_{FS}}$ {4}*
0.5	0.71023	—	<1	0.38	2.8	0.67	1.56	0.68	0.56	0.05
0.2	0.0358	—	<1	1.8	8.3	2.0	—	—	—	—
0.1	0.00282	—	<1	3.1	37.7	8.8	1.81	1.40	12.3	70.3
0.05	0.000288	34	1.3	3.8	246	55.7	1.91	2.71	136	3714.4
0.02	0.0000167	43	8.2	4.1	3472	495	—	—	—	—
					All-terminal connectivity					
0.2	0.1876	—	<1	0.43	23	5.4	—	—	—	—
0.1	0.0226	1.2	4.0	0.72	85	20	—	—	—	—
0.05	0.002650	1.5	10.2	0.86	504	108	—	—	—	—
0.02	0.000158	1.6	23.0	0.95	6960	1227	—	—	—	—
0.01	0.00002	1.7	109	0.98	51590	7635	—	—	—	—

*) Computed by MP.
{1}* Dagger sampling [16], source [9].
{2}* Sequential Construction [6], source [9].
{3}* Method of Bounds [9].
{4}* Methods of failure sets [14], source [9]. The failure set method produced two estimates, the table presents the better one, with larger value of W_{CMC}/W_{FS}.

5. In our experiments, sparse networks were represented by the dodecahedron. Based on the results in tables 3 & 4, we suggest using MP for $0 \leq Q < 0.05$.

6. The comparison with the results from the literature reveals that MP (for a particular value of the q-vector) is competitive with the sequential destruction method [16] applied to a very sparse network; see table 3. Based on data in [9], MP Monte Carlo considerable outperforms the dagger method and the sequential construction method when applied to the dodecahedron, for $Q < 0.05$; see table 4. For the same example, the MP is inferior to the Fishman method of bounds [9] by a factor of 1.5–2.5.

TABLE 5
Simulation Results for Complete Graphs
(all-terminal connectivity). Edge failure probability $q = 0.55$

The graph	Network failure probability Q^*)	$\frac{W_{CMC}}{W_{DP}}$	$\delta^{\%}_{MP}$	η_{MP}	$\frac{W_{CMC}}{W_{MP}}$
K_{10}	$0.456 \cdot 10^{-1}$	<1	0.54	70	21
K_{15}	$0.346 \cdot 10^{-2}$	<1	0.57	887	169
K_{20}	$0.232 \cdot 10^{-3}$	<1	0.53	15160	3280
K_{25}	$0.147 \cdot 10^{-4}$	<1	0.50	$0.27 \cdot 10^{6}$	47220
K_{30}	$0.889 \cdot 10^{-5}$	<1	0.47	$5.08 \cdot 10^{7}$	$0.73 \cdot 10^{7}$

*) Estimated by MP.

The dodecahedron with terminals $s=1$ and $t=20$ (figure 1a) has a specific feature with respect to the method of bounds. Namely, the bound $1-A$ in terms of edge-disjoint cuts [9, p 149; 11, p 463] asymptotically coincides—in this particular case—with the true network unreliability value, since the collection of cuts chosen for A contains all minimum size cuts between s and t (which is far from being so, for general G, s, t).

The MP is considerable inferior to the Karp-Luby method of failure sets; see table 4. The Karp-Luby method requires extra effort for computing failure sets, in terms of computer time and computer memory; see comments on this issue in [9, p 53].

ACKNOWLEDGMENT

We thank the referees for their constructive and helpful comments.

REFERENCES

[1] A. Agraval, R. E. Barlow, "A survey of network reliability and domination theory", *Operations Research*, vol 32, 1984, pp 478–492.
[2] R. E. Barlow, F. Proschan, *Statistical Theory of Reliability and Life Testing*, 1975; Holt, Rinehart & Winston.
[3] C. J. Colbourn, *The Combinatorics of Network Reliability*, 1987; Oxford Univ. Press.
[4] C. J. Colbourn, D. D. Harms, "Bounding all-terminal reliability in computer networks", *Networks*, 1988, pp 1–12.
[5] H. A. David, *Order Statistics*, 1981, 2nd ed; John Wiley & Sons.
[6] M. C. Easton, C. K. Wong, "Sequential destruction method for Monte Carlo evaluation of system reliability", *IEEE Trans. Reliability*, vol R-29, 1980 Apr, pp 27–32.
[7] W. Feller, *An Introduction to Probability Theory and Its Applications*, vol II, 1966; John Wiley & Sons.
[8] G. S. Fishman, "A Monte Carlo sampling plan for estimating network reliability", *Operations Research*, vol 34, 1986, pp 581–592.
[9] G. S. Fishman, "A comparison of four Monte Carlo methods for estimating the probability of *s-t* connectedness", *IEEE Trans. Reliability*, vol R-35, 1986, pp 145–154.
[10] G. S. Fishman, "A Monte Carlo sampling plan for estimating reliability parameters and related functions", *Networks*, vol 17, 1987, pp 169–186.
[11] G. S. Fishman, "Estimating the *s-t* reliability function using importance and stratified sampling", *Operations Research*, vol 37, 1989, pp 462–473.
[12] F. Harary, *Graph Theory*, 1969; Addison Wesley.
[13] J. E. Hopcroft, J. D. Ullman, "Set merging algorithms", *SIAM J. Computing*, vol 2, 1973, pp 296–303.
[14] R. Karp, M. G. Luby, "A new Monte Carlo method for estimating the failure probability of an *n*-component system", 1983; Computer Science Div., Univ. of California.
[15] J. B. Kruskal, "On the shortest spanning tree of a graph and the travelling salesman problem", *Proc. Amer. Mathematical Soc.*, vol 7, 1956, pp 48–50.
[16] H. Kumamoto, et al, "Dagger sampling Monte Carlo for system unavailability evaluation", *IEEE Trans. Reliability*, vol R-29, 1980 Jun, pp 122–125.
[17] H. Kumamoto, T. Kazuo, I. Koichi, E. J. Henley, "State transition Monte Carlo for evaluating large, repairable systems", *IEEE Trans. Reliability*, vol R-29, 1980 Dec, pp 376–380.
[18] M. V. Lomonosov, V. P. Polesskii, "On maximum of probability of connectivity", *Problems of Information Transmission*, vol 8, num 4, 1972.
[19] M. V. Lomonosov, "Bernoulli scheme with closure", *Problems of Information Transmission*, vol 10, num 1, 1974.
[20] M. V. Lomonosov, "Tender-spot of a reliable network", *Discrete Applied Mathematics*, (to appear).
[21] Lucia I. P. Resende, "Implementation of factoring algorithm for reliability evaluation of undirected networks", *IEEE Trans. Reliability*, vol 37, 1988 Dec, pp 462–468.
[22] A. Satanarayana, R. K. Wood, "A linear time algorithm for *k*-terminal reliability in series-parallel networks", *SIAM J. Computing*, 1985, pp 818–832.

AUTHORS

Dr. Tov I. Elperin; The Pearlstone Center for Aeronautical Engineering Studies; Dept. of Mechanical Engineering; Ben Gurion University of the Negev; POBox 653; Beer-Sheva 84105 ISRAEL.

Tov. I. Elperin: For biography, see *IEEE Trans. Reliability*, vol 39, 1990 Jun, p 208.

Dr. Ilya B. Gertsbakh; Dept. of Mathematics and Computer Science; Ben Gurion University of the Negev; POBox 653; Beer-Sheva 84105 ISRAEL.

Ilya B. Gertsbakh: For biography, see *IEEE Trans. Reliability*, vol 39, 1990 Jun, p 208.

Dr. Michael V. Lomonosov; Dept. of Mathematics and Computer Science; Ben Gurion University of the Negev; POBox 653; Beer-Sheva 84105 ISRAEL.

Michael V. Lomonosov is an Associate Professor in the Dept. of Mathematics and Computer Science at the Ben Gurion University of the Negev. His research interests include network reliability and network flow theory.

Manuscript TR89-113 received 1989 December 12; revised 1990 September 22; revised 1991 April 28.

IEEE Log Number 01573 ◄TR►