International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:12, 2007

Combinatorial Approach to Reliability Evaluation of Network with Unreliable Nodes and Unreliable Edges

Y. Shpungin

***Abstract*—**Estimating the reliability of a computer network has been a subject of great interest. It is a well known fact that this problem is NP-hard. In this paper we present a very efficient combinatorial approach for Monte Carlo reliability estimation of a network with unreliable nodes and unreliable edges. Its core is the computation of some network combinatorial invariants. These invariants, once computed, directly provide pure and simple framework for computation of network reliability. As a specific case of this approach we obtain tight lower and upper bounds for distributed network reliability (the so called residual connectedness reliability). We also present some simulation results.

***Keywords*—**Combinatorial invariants, Monte Carlo simulation, reliability, unreliable nodes and unreliable edges.

I. Introduction

ESTIMATING network reliability has been drawing a lot of scientific attention in the field. In this short paper we skip a detailed survey of the previous work and only point out several main research directions:

- algorithms for reliability computation [1]-[5]
- reliability estimation by means of simulation [6]-[17]
- constructing tractable lower and upper bounds on the network reliability [18]-[20]

Network reliability estimation problems are usually NP-hard [2] and this explains why Monte Carlo (MC) methods are so popular in solving reliability problems for large networks. The essence of most MC applications is a so called Crude Monte Carlo (CMC). The main drawback of CMC is that they are very inefficient in two extreme cases: highly reliable and highly unreliable networks (the so called rare event phenomenon). The inefficiency expresses itself in the unbounded growth of the relative error resulting from the reliability increase (or decrease). Note that the investigation of reliable networks most important in applications. We present a very promising approach, based on some new combinatorial ideas, which is especially effective for highly reliable networks. This approach incorporates both simulation and analytic methods and eliminates in principle the rare event phenomenon.

Y. Shpungin is with Negev Monte Carlo Research Center (NMCRC) and Department of Software Engineering, Sami Shamoon College of Engineering (SCE), Beer Sheva, 84100, Israel (e-mail: yosefs@sce.ac.il).

To the best of our knowledge, most of the existing algorithms for reliability computing address either unreliable nodes or unreliable edges, but not both.

In this paper, we present a Monte Carlo scheme for a more complicated and realistic case when both nodes and edges are unreliable and also address the problem of network's reliability estimation when only the nodes are unreliable. In section 1, we give some basic definitions and explain the nature of main difficulties arising in CMC and its modifications. In section 2, we propose our combinatorial approach, describe the appropriate simulation schemes and show how they can be applied to evaluate the reliability of a distributed network. In section 3, we present some numerical results.

A. Basic Notions and Definitions

All networks consist of *vertices* (*nodes*) and *edges*. There are many types of networks varying in their performance definitions and therefore with different concepts of reliability. For our purposes, we define a network in the following manner. By network $N=(V,E,T)$ we denote an undirected graph with a node-set $V, |V|=n$, an edge-set $E, |E|=m$, and a set $T \subseteq V$ of special nodes called *terminals*. Further, by *node* we mean a node, which is not a terminal. Nodes and edges could fail (become non-operational). If an element (either node or edge) fails, we say that it is *down*, otherwise we say it is *up*. Each node $v \in V$ is associated with a probability p_v of being *up* and a probability $q_v = 1 - p_v$ of being *down.* For the edges the appropriate probabilities will be p_e and q_e. We say that nodes are identical if they all have the same probability of being *up*, that is for each $v_1, v_2 \in V$, we have $p_{v_1} = p_{v_2} = p_v$. Identical edges are defined in a similar way. Let us consider three important types of networks:

a) Edges are reliable and nodes are not reliable, i.e. edges never fail and a node $v \in V \setminus T$ can fail.
b) Nodes are reliable and edges are not reliable, i.e. an edge $e \in E$ can fail.
c) Both edges and nodes are not reliable.

For convenience we suppose that terminals do not fail. In a given network N, the *state* of N, by definition, is a network induced by all its elements (nodes and edges) which are in the *up state.* We say that the state is *Good* if the network is

operational according to some operational criterion and *Bad* otherwise. *B* denotes a class of *Bad* states while *G* denotes a class of *Good* states, respectively. In this paper, we deal with two types of the network's operational criteria.

- *Terminal connectivity*: the state is *Good* if any pair of terminals is connected by the elements in the *up* state. In the case of $T \subset V, |T| = k$, we will use the term *"k-connectivity*" and in the case of $T = V$, we will use the term "*overall connectivity*".
- *Residual connectivity*: the state is *Good* if all the nodes which are *up* are connected by the edges in the *up* state.

The *terminal connectivity* criterion is quite traditional. It has the property of being monotone: each subset of a *Bad* state is a *Bad* state and each superset of a *Good* state is a *G*ood state. The *residual* connectivity criterion is highly applicable for the distributed computer network performance, but this criterion does not satisfy the *monotony* property.

We define the network reliability $R(N)$ as the probability that the network is in the *Good* state. Let us illustrate some defined notions. In Fig. 1.a we see a network *N*. Fig. 1.b shows a state $\{a,b,c,e,1,2,6\}$ of network *N*. Suppose that the operational criterion is terminal connectivity and nodes *a* and *e* are the terminals. Then we see that the state in Fig. 1.b is a *Good* state. Now let us suppose that the criterion is the residual connectedness. We see that the node *c* is disconnected from other nodes in the *up*-state. So, for the residual connectedness criterion, the same state in Fig. 1.b is a *Bad* state.

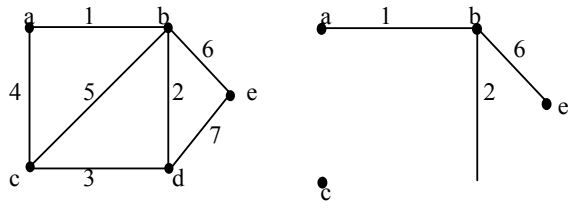

Fig. 1 (a) The Network *N*; (b) The possible state of *N*

B. Reliability and Monte Carlo

To explain the advantages of the combinatorial approach, let us look at the Crude Monte Carlo (CMC) from a more general point of view. Consider an *urn U* with a large number of balls *b* in it. Suppose that each ball *b* is marked with some value $z(b)$ and we want to calculate the sum of $z(b)$ over *b* in *U*:

$$Z = \sum_{b \in U} z(b) \qquad (1)$$

This completely matches the computation of network reliability. In this case, the balls *b* are the states, and $z(b)$ are defined as 0 for any *Bad* state and as the probability of the state if it is good. Therefore, *Z* becomes the reliability of the network. Since the number of balls in *U* is large, the whole sum cannot be computed precisely, so we are forced to estimate *Z*. To convert the above expression into an MC scheme, we introduce the probability distribution $p(b)$ on *U*. Ball *b* will be drawn with probability $p(b)$. Then one may express the sum *Z* in the following form:

$$Z = \sum_{b \in U} p(b) \cdot \frac{z(b)}{p(b)} = E[Y(b)], \qquad (2)$$

where $Y(b) = \frac{z(b)}{p(b)}$. (Note that values of $Y(b)$ are 0 or 1 for the *Bad* and *Good* states, respectively). The main idea of this representation is that the sum *Z* is expressed as an *expectation* of some random variable. Now the CMC scheme for evaluating *Z* can be stated as follows:

- draw ball *b M* times from the urn *U*, with probability $p(b_i)$;
- calculate $\hat{Y} = \frac{1}{M} \cdot \sum_{i=1}^{M} Y(b_i)$, which is an *unbiased* estimator of *Z*.

It is clear from the CMC simulation scheme that when the network is highly reliable, we will almost never observe the *Bad* states. This means that the estimated value of reliability might be 1. In Reliability theory such effect is called "*the rare event phenomenon*". It is easy to get the general expression for the relative error: $\delta_{CMC} = \frac{1}{\sqrt{M}} \cdot \sqrt{\frac{R}{1-R}}$. We see the unbounded growth of the relative error for highly reliable networks (i.e. for $R \to 1$). We would like to emphasize once again that it is the main deficiency of the CMC and various versions of it, reducing this unbounded growth but *not eliminating* it. The reason for this lies in the above *urn* scheme. Namely, each ball, which represents the *state*, is drawn from the *urn* with the *probability of this state*. Therefore the growth of the relative error is caused by a bad definition of the probability measure $p(b)$ on network states.

As it was mentioned earlier, a very promising approach for evaluating reliability of network with unreliable edges, was suggested and developed in the papers [8], [9], [12], [14]-[17]. What is essential for this approach is that the simulation schemes are *homogeneous*. In plain words, a MC scheme is homogeneous, if the *balls* are drawn from the *urn* with probability which *does not depend* on the probabilities of the states. (More on homogeneous schemes see in [9]). One important feature of the homogeneous schemes is that the relative error is bounded. (A basic example of a non-homogeneous scheme is CMC).In this paper, we suggest a homogeneous Monte Carlo scheme for reliability problems of a network with unreliable nodes and unreliable edges.

II. Description of the Method

A. Combinatorial Approach to Reliability Evaluation

In this section we deal with the *k*-connectivity criterion. We start by introducing some very useful network

combinatorial notions. Let Π_V be a set of all node permutations of nodes in V and let Π_E be a set of all edge permutations in E. Define the following Cartesian Product: $\Pi = \Pi_{V\setminus T} \times \Pi_E$. As a result every permutation $\pi \in \Pi$ is a pair $(\pi_{V\setminus T}, \pi_E)$, where $\pi_V \in \Pi_{V\setminus T}, \pi_E \in \Pi_E$. By sub-permutation $\pi(i, j)$ of π we mean a sequence constructed of i first nodes from π_V and j first edges from π_E. For each sub-permutation $\pi(i, j)$ define a network state $N(\pi(i, j))$ where all the nodes and edges in $\pi(i, j)$ are *up* and all other nodes and edges are *down.* For our further exposition, the following definition of so called *anchor* is very important.

Definition. Associate with each permutation $\pi \in \Pi$ a pair of indexes $(r := r(\pi), s := s(\pi))$ so that the state $N(\pi(r, s))$ is *Good*, but for all $i < r$ and $j \le m$, $N(\pi(i, j))$ is *Bad* (remind that $m = |E|$). We call this state $N(\pi(r, s))$ an *anchor* of the permutation π.

Note that for all $i > r$ and $j > s$, the states $N(\pi(i, j))$ are *Good* due to the monotony of the *k-connectivity* criterion. Let us illustrate this definition by an example.

Example. Suppose we have a network with unreliable nodes and edges shown in Fig. 1.a. The nodes *a* and *e* are terminals and therefore they are reliable. Then for the permutation $\pi = ((c, b, d), (5,1,2,6,7,4,3))$ we have $r = 2, s = 4$ since $N(\pi(2,4))$ is a *Good* state, while for all $i < 2$ and $j \le 7$ the states $N(\pi(i, j))$ are *Bad*. Clearly, for all $i > 2$ and $j > 4$, the appropriate states are *Good* states. An *anchor* is not necessarily the minimal path set. For example, consider some other state for the same network: {b,1,4,6}. Clearly, it is not a minimal path set, but it is an anchor of the permutation ((*b*),(*1,4,6*)).

Now we prove the following simple claims which provide the Monte Carlo schemes for the network types in question.

Claim 1. Let *N* be a network with identical nodes and identical edges and with the *k-connectivity* criterion. Then its reliability may be expressed in the following form:

$$R(N) = \sum_{\pi \in \Pi} f(\pi), \qquad (3)$$

where $f(\pi) = \sum_{i=r(\pi)}^{n} \frac{1}{i!\cdot(n-i)!} p_v^i q_v^{n-i} \sum_{j=s(\pi)}^{m} \frac{1}{j!\cdot(m-j)!} p_e^j \cdot q_e^{m-j}$, (4)

where p_v and p_e are node and edge *up* probabilities, correspondingly.

Proof. Network reliability $R(N)$ equals the sum of all *Good* states probabilities. Consider a *Good* state where i nodes and j edges are *up*, and the rest of the nodes and edges are *down*. It is easy to see that this state is $N(\pi(i, j))$ for some permutation π, and it will remain the same as $N(\pi(i, j))$ for exactly $i!\cdot(n-i)!\cdot j!\cdot(m-j)!$ permutations. This is because while we have the partition to *i* and *n-i* nodes and also *j* and *m-j* edges, the state remains the unchanged and it stays that way for all possible permutations within the partition. As a result, we get the denominators in (4). The criterion monotony results in the following. For all $k > i, l > j$ corresponding states $N(\pi(k,l))$ are *Good,* and from this the assertion can be made.

The monotony property is essential for claim 1. As a result, we cannot use formula (4) for the residual connectedness criterion. Nevertheless, for this case we can compute lower and upper bounds for reliability, based on the formula (4). We have made some calculations of these bounds for a network with both unreliable nodes and edges. However in this brief paper we present the results for the "one-dimensional" network type (a network with unreliable nodes and reliable edges). This type is a special case of the "two-dimensional" type (a network with both unreliable nodes and edges). It is very useful in applications.

Claim 2. Let *N* be a network with identical and unreliable nodes and with reliable edges. Suppose that the criterion is the residual connectedness. Then we have the following bounds for network reliability:

$$\sum_{\pi \in \Pi_V} f_{low}(\pi) \le R(N) \le \sum_{\pi \in \Pi_V} f_{up}(\pi), \qquad (5)$$

where

$$f_{low}(\pi) = \frac{1}{r(\pi)!(n-r(\pi))!} \cdot p^{r(\pi)} \cdot q^{n-r(\pi)} \qquad (6)$$

and

$$f_{up}(\pi) = \sum_{i=r(\pi)}^{n} \frac{1}{i!(n-i)!} \cdot p^i \cdot q^{n-i}. \qquad (7)$$

Proof. As it was mentioned above, the residual connectedness criterion does not provide monotony property. Therefore, for $i > r(\pi)$, all states $N(\pi(i))$ will be *Bad* in the worst case, and *Good* in the best case.

Remark. It is easy to see that $\lim_{p\to 1}(f_{up}(\pi) - f_{low}(\pi)) = 0$.

B. Monte Carlo Scheme for Evaluating Network Reliability Number

To get the expression (4) in a more appropriate form for Monte Carlo computations, let us introduce some combinatorial invariant of a network for any reliability criterion.

Definition. Denote by $x_{i,j}$ the number of *Good* states where *i* nodes and *j* edges are *up* and such that $N(\pi(i, j))$ is the *anchor* of some permutation π, i.e., $i = r(\pi), j = s(\pi)$. We say that the set $\{\{x_{i,j}\}, 1 \le i \le n, 1 \le j \le m\}$ is a *combinatorial spectrum* of the network.

Remark. Notion of spectrum in another form and for the network with unreliable edges was also introduced in [17].

Now, changing the order for summation in (4), and multiplying and dividing by the same expression $n!\cdot m!$, we get:

$$R(N) = \frac{1}{n! \cdot m!} \sum_{r=1}^{n} \sum_{s=1}^{m} x_{r,s} \cdot f(r,s) \cdot n! \cdot m!, \qquad (8)$$

where

$$f(r,s) = \sum_{i=r}^{n} \frac{1}{i! \cdot (n-i)!} p_v^i \cdot q_v^{n-i} \cdot \sum_{j=s}^{m} \frac{1}{j! \cdot (m-j)!} p_e^j \cdot q_e^{m-j}. \quad (9)$$

The difference between $f(\pi)$ in (4) and $f(r,s)$ in (9) is the following. The formula for $f(\pi)$ expresses the probability of all *Good* states "growing up" from the state $N(\pi(r,s))$, defined by a certain permutation π. In contrast $f(r,s)$ is similar probability, but for all permutations having the same associated values *r* and *s*. Note that the values $x_{i,j}$ are topological invariants of the network in the sense that they don't depend on the nodes and edges probabilities. This means that, once computed, they may be used for reliability calculation for any probabilities of nodes and edges. We would like to emphasize that the expression (8) provides the homogeneous MC scheme, as we simulate the permutations with the probabilities $\frac{1}{n! \cdot m!}$. The latter don't depend on the network states probabilities.

The following Monte Carlo Scheme is based on the formula (8).

Simulation Scheme

Step1. Initiate all $x_{i,j}$ to be 0.

Step2. Simulate the permutation $\pi \in \Pi$.

Step3. Find $r = r(\pi)$ - the minimal index of node in π so that the state $N(\pi(r,m))$ is *Good*.

Step4. Find $s = s(\pi)$ - the minimal index of edge in π so that the state $N(\pi(r,s))$ is *Good.*

Step5. $x_{r,s} := x_{r,s} + 1$.

Step6. Repeat steps 2-5 *M* times.

Step7. For each $x_{i,j} \neq 0$ $(1 \le i \le n, 1 \le j \le m)$ compute the value of $f(i,j)$ by (9).

Step8. Compute the average $\frac{\sum_{i=1}^{n} \sum_{j=1}^{m} x_{i,j} f(i,j)}{M}$. It is an unbiased estimator for $R(N)$**.**

Note that the computation scheme is very simple. After steps from 1 to 6 we get the estimates of $x_{i,j}$. The computations of all $f(i,j)$ are straightforward and not time-consuming, since the most values of $x_{i,j}$ are zeros.

Example. In this example we show one run of the above scheme for the network in Fig. 1. Suppose that the simulated permutation π is ((d,b,c),(2,5,4,1,7,6,3)). We see that *r*=2 is the minimal index for the state $N(\pi(r,7))$ to be Good. We can also see that the minimal *j* for the state $N(\pi(2,j))$ to be good is j=5. Now we let $x_{r,s} := x_{r,s} + 1$ and proceed to the next run.

The simulation scheme for bounds computation is based on the claim 2. The scheme is similar to the above scheme for reliability evaluation , and similarly makes use of the combinatorial spectrum (analogous to the spectrum in the "two-dimensional" case). We omit the description of the scheme in this paper and only present the computational results in Table III.

Remark. The results we described are valid for networks with unreliable and identical nodes and unreliable and identical edges. It is possible to use our method for different (non-identical) nodes and edges if the difference between the maximal and minimal probabilities is small enough (as is the case of many applications). By using the same ideas we can compute the lower and upper bounds for the reliability.

III. Numerical Examples

In this section, we present simulation results obtained by the MC schemes described in section 2. For these simulations we choose two different networks. One is a *hypercube* H_6 with $2^6 = 64$ nodes and $2^5 \cdot 6 = 192$ edges. The hypercube is very helpful for distributed systems, since it is very reliable, has a symmetric and clear topology and allows effective implementation of different algorithms. The other network is *grid* with 100 nodes and 176 edges. Tables I and II present the estimates of reliability for these networks with two-terminal criterion. Table III presents the estimates of lower and upper bounds for reliability. These estimates are carried out for hypercube with unreliable nodes and for residual connectedness criterion. All simulation results are based on $M = 10^5$ replications.

Remark. By the relative error δ of estimating probability Φ we mean the absolute error of Φ divided by $\min\{\Phi, 1-\Phi\}$. Thus, when *R* is estimated to be 0.9605 with the 1.03% relative error (Table I), this percentage is to be taken from the complementary 0.0395.

TABLE I
Simulation Results for Hypercube H_6
(2-Terminal Connectivity)

p_v	0.7	0.8	0.9	0.95
p_e	0.7	0.8	0.9	0.95
$\hat{R}$	0.2061	0.3871	0.6482	0.8136
δ	0.0053	0.0036	0.0039	0.0059
p_v	0.95	0.95	0.95	0.99
p_e	0.96	0.97	0.98	0.99
$\hat{R}$	0.8494	0.8818	0.9227	0.9605
δ	0.0060	0.0076	0.0101	0.0103

TABLE II
SIMULATION RESULTS FOR GRID
(2-TERMINAL CONNECTIVITY)

p_v	0.7	0.8	0.9	0.95
p_e	0.7	0.8	0.9	0.95
$\widehat{R}$	0.0155	0.1395	0.4796	0.7159
δ	0.0194	0.0057	0.0029	0.0042
p_v	0.95	0.95	0.95	0.99
p_e	0.96	0.97	0.98	0.99
$\widehat{R}$	0.7664	0.8193	0.8749	0.9384
δ	0.0051	0.0055	0.0064	0.0097

TABLE III
SIMULATION RESULTS FOR HYPERCUBE H_6
(RESIDUAL CONNECTEDNESS)

p	0.5	0.6	0.7	0.8	0.9
$\widehat{R}_{low}$	0.71503	0.85112	0.96783	0.99650	0.99992
$\widehat{R}_{up}$	0.94946	0.97291	0.99218	0.99885	0.99996

IV. CONCLUSION

1) The suggested MC scheme is very efficient and can easily be implemented for evaluating reliability for the network with unreliable and identical nodes and unreliable and identical edges, for the case of k-terminal connectivity criterion.
2) For the residual connectedness criterion the same approach provides tight lower and upper bounds for reliability.
3) One of the main advantages of the scheme is eliminating the rare event phenomenon. This fact results in bounding the relative error, so the method it is especially efficient for highly reliable networks.
4) One of the substantial advantages of the scheme is that, once computed, *combinatorial spectrum* serves for as many values of nodes and edges probabilities as needed.
5) For the more complicated case of non-identical nodes and non-identical edges, when the probabilities are close enough, the tight lower and upper bounds for the reliability may be computed.

REFERENCES

[1] Agraval, R. E. Barlow, "A survey of network reliability and domination theory", *Operation Research*, vol. 32, 1984, pp. 478-492.
[2] M.O. Ball, "Complexity of network reliability computation", *Networks*, 10, 1980, 153-165.
[3] R. E. Barlow, F. Proschan, *Statistical Theory of Reliability and Life Testing*, 1975. Holt, Rinehart & Winston.
[4] C. J. Colbourn, *The Combinatorics of Network Reliabilty*. Oxford Univ. Press.
[5] C.J. Colbourn, A. Satyanarayana, C. Suffel, K. Sutner, "Computing residual connectedness reliability for restricted networks", *Discrete Applied Mathematics*, 44, 1993, 221-232.
[6] Lucia I. P. Resende, "Implementation of factoring algorithm for reliability evaluation of undirected networks", *IEEE,Trans. Reliability*, vol 37, 1988, pp. 462-468.
[7] M. C. Easton, C. K. Wong, "Sequential destruction method for Monte Carlo evaluation of system reliability", *IEEE Trans. Reliaility*, vol R-29, 1980, pp. 27-32.
[8] T. Elperin, I. Gertsbakh, M. Lomonosov, "Estimation of network reliability using graph evolution models", *IEEE Transactions on Reliability*, 40(5), 1991, 572-581.
[9] T. Elperin, I. Gertsbakh, M. Lomonosov, "An evolution model for Monte Carlo estimation of equilibrium network renewal parameters", *Probability in the Engineering and Informational Sciences*, 6, 1992, 457-469.
[10] G. S. Fishman, "A Monte Carlo sampling plan for estimating network reliability", *Operations Research*, vol 34, 1986, pp. 581-592.
[11] G. S. Fishman, "A Monte Carlo sampling plan for estimating reliability parameters and related functions", *Networks*, vol. 17, 1987, pp 169-186.
[12] K-P. Hui, N. Bean, M. Kraetzl, and D P. Kroese, "The Tree Cut and Merge Algorithm for Estimation of Network Reliability", *Probability in the Engineering and Informational Sciences*, 17(1):24-25, 2003.
[13] 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, pp 376-380.
[14] M. Lomonosov, "On Monte Carlo estimates in network reliability", *Probability in the Engineering and Information Sciences, 8*, 1994, 245-64.
[15] M Lomonosov, Y. Shpungin, "Combinatorics of Reliability Monte Carlo", *Random Structures & Algorithms*, 14,1999, No. 4, 329-343.
[16] Y.Shpungin, "Combinatorial and Computational Aspects of Monte Carlo Estimation of Network Reliability", Ph. D. Dissertation, Dept. of Mathematics and Computer Science, Ben Gurion University of the Negev, 1996.
[17] Gertsbakh and Y. Shpungin, "Combinatorial approaches to Monte Carlo estimation of network lifetime distribution", *Appl. Stochastic Models Bus. Ind.*, 2004, 20:49-57.
[18] C. J. Colbourn, D. D. Harms, "Bounding all-terminal reliability in computer networks", *Networks*, 1988, pp. 1-12.
[19] Z. He, Y. Tian and Y. Chen, "Simulating the Reliability of Distributed Systems with Unreliable Nodes", *I. J. of Simulation*, vol 3, 1-2, pp. 68-78.
[20] K-P. Hui, "Reliability Estimation", Ph D. dissertation, Faculty of Engineering, Computer and Mathematical Sciences, University of Adelaide, 2005.