Network reliability importance measures : combinatorics and Monte Carlo based computations

ILYA GERTSBAKH
Department of Mathematics
Ben Gurion University
P.O.B 653 Beer Sheva 84105
ISRAEL

YOSEPH SHPUNGIN
Department of Software Engineering
Negev Academic College of Engineering and NMCRC
P.O.B 950 Beer Sheva 84100
ISRAEL

Abstract: - In this paper we focus on computational aspects of network reliability importance measure evaluation. It is a well known fact that most network reliability problems are NP-hard and therefore there is a significant gap between theoretical analysis and the ability to compute different reliability parameters for large or even moderate networks. In this paper we present two very efficient combinatorial Monte Carlo models for evaluating network reliability importance measures.

Key-Words: - Network, Reliability, Importance Measure, Monte Carlo, Combinatorial Approach

1 Introduction

Computing network reliability is very important but it is not the only problem in reliability analysis. One of the purposes of network reliability analysis is to identify the weakness in a system and to quantify the impact of component failures on the network failure. The so called "reliability importance measures" are used for this purpose. The importance measures provide numerical indicator to determine which components are more important for network reliability improvement or more critical for system failure. Many different importance measures were proposed in literature. But there is a significant gap between theoretical analysis and the ability to compute these measures for moderate or large networks. Therefore using Monte Carlo (MC) methods in solving such problems is very popular. The essence of most MC applications is the so called Crude Monte Carlo (CMC). The main drawback of CMC is that it is very inefficient in two extreme cases: highly reliable and highly unreliable networks (the so called rare event phenomenon). Our purpose in this paper is to describe how two very efficient MC models can be used for evaluating network reliability importance measures. The common feature for these two models is that the appropriate simulation schemes are *homogeneous*. Let us explain the latter notion in plain words. 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 equals 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 very large, the whole sum cannot be computed precisely, and we are forced to *estimate Z* by some MC scheme. We say that 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 [1]). One important feature of homogeneous schemes is that the relative error is bounded. (A basic example of a non-homogeneous scheme is the so called Crude Monte Carlo and its variations). As it was mentioned above, the main purpose of this paper is to present efficient Monte Carlo methods applicable to large networks. This paper is organized as follows. In Section 2, we give some basic notions and definitions. In Section 3, we present a computationally efficient *model* [1,2] for evaluating reliability gradient vector, which can be used for computing Birnbaum Importance Measure in general case of networks with non-identical elements. For networks with identical elements we propose in Section 4 a highly efficient spectrum approach [3-5]. It's worth noting that this approach provides easily implemented computations and allows obtaining with minimal difficulties different topological features of the network. Section 5 presents a numerical example.

2 Basic notions and definitions

All networks have *vertices* (*nodes*) and *edges*. There are many types of networks varying in their performance definitions and therefore with different concepts of their reliability. Let *K*-network be an undirected graph $N = (V, E, K)$ with a node-set V, an edge-set E and a set $K \subseteq V$ of special nodes called *terminals*. Also let $|V| = m$ and $|E| = n$. In our model, nodes can never fail, while edges can. If an edge fails, we say that it is *down*; otherwise we say it is *up*. By state of a network we call a binary vector $(x_1,..,x_n)$, where each component $x_i = 1$ if e_i is *up* and $x_i = 0$ otherwise. A state of the network is defined as being *Good* if any two terminals are connected by a path consisting of edges in the *up* state. Otherwise it is *Bad*. The *terminal connectivity* criterion 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. There are two network reliability models: *static* and *dynamic*. In this paper we restrict our attention to static networks. Each edge e_i is associated with probability p_i of being *up* and a probability $q_i = 1 - p_i$ of being *down*. We say that edges are identical if they all have the same probability of being *up*, that is for each $i \neq j$ we have $p_i = p_j = p$. We define the network reliability $R = R(p_1,..., p_n)$ as the probability that the network is in a *Good* state.

The following are the main *component* reliability importance measures proposed in literature: Birnbaum Importance Measure (BIM) [6], Fussell-Vesely Importance Measure (FVIM) [7], Criticality Importance [8], Reliability Achievement Worth [9], Reliability Reduction Worth [9]. In our paper we focus on the two first of them. The BIM of element e_i is defined as

$$I_i^B = \frac{\partial R(p_1,..., p_n)}{\partial p_i} \qquad (2)$$

It expresses the rate of increase of the network reliability with respect to the element's reliability increase.

Remark. For equal $p_i \equiv p$, first the derivatives $\frac{\partial R}{\partial p_i}$ are computed and only afterwards all p_i set to be equal p.

The FVIM of element e_i is defined by

$$I_i^{FV} = 1 - \frac{R(p_1,..., p_{i-1}, 0, p_{i+1},..., p_n)}{R(p_1,..., p_n)} \qquad (3)$$

It quantifies the decrement in system reliability caused by a particular component failure.

3 Using Reliability Gradient for BIM evaluation

As it was mentioned in Introduction, the reliability gradient vector can be used for BIM evaluation. The purpose of this section is to describe a special form of the gradient vector which allows using a highly efficient Graph Evolution Model [2] for its computation. Similar form of the gradient was outlined in [1]. We will get here a special form of gradient for more general case of monotone systems. Let us consider a monotone system of *n* elements. Suppose that each element e_i may be in two states : *up* with probability p_i and *down* with probability q_i. The *state* of a system is defined as a binary vector $(x_1,..,x_n)$, where each component $x_i = 1$ if e_i is *up* and $x_i = 0$ otherwise. All 2^n binary states are divided into two classes: *Good* and *Bad.*

Definition 1. Reliability gradient vector ∇R is defined as $\nabla R = (\frac{\partial R}{\partial p_1},..., \frac{\partial R}{\partial p_n})$, i.e. component *i* of the reliability gradient vector is BIM of element e_i.

Definition 2. System state $w = (w_1,..., w_n) \in Bad$ is called *direct neighbor* or simply *neighbor* of state $v = (v_1,..., v_n) \in UP$ if *w* differs from *v* in exactly one position. The set of all neighbor states of *DOWN* is called *border set* and denoted as *DN**. Obviously, $DN^* \subseteq DOWN$.

It turns out that the reliability gradient vector is intimately related to border states. To reveal this connection, we introduce an *artificial evolution process* on system elements. At $t = 0$ all elements are *down*. Element e_i is "born" after random time $\tau_i \sim \exp(\lambda_i)$, where λ_i is chosen so that the following equality takes place: $p_i = P(\tau_i \le 1) = 1 - e^{-\lambda_i}$. After the "birth", element e_i remains *up* forever. Consider two system states $v = (v_1, v_2,..., v_{i-1}, 0, v_{i+1},...v_n)$ and

$w=(v_1,v_2,...,v_{i-1},1,v_{i+1}...,v_n)$. Suppose that at time *t* the system is in state *v*. We look for the probability that during a small time interval Δt the system moves from *v* to *w*. Obviously, it will happen iff the element e_i is born during this interval, and all other components which are in state 0 will not become alive during the same interval. The first event has probability $\lambda_i\cdot\Delta t+o(\Delta t)$, and the second event has probability $1-o(\Delta t)$. Then the probability that during $[t,t+\Delta t]$ there will be the transition $v\rightarrow w$ equals $\lambda_i\cdot\Delta t+o(\Delta t)$. Let v be a border state of system, i.e. $v\in DN^*$. Denote by $\Gamma(v)$ the sum of λ_i over the set of all indices *i* such that $v+(0,...,1_i,...0)\in Good$. Call $\Gamma(v)$ the *flow* from *v* into *Good*. Formally, $\Gamma(v)=\sum_{\{v\in DN^*,v+(0,...,0,1_i,0,...,0)\in UP\}}\lambda_i$. We need two other notations. Let $R(p_1(t),...,p_n(t))$ be the probability that the system is in *Good state* at the instant *t*. Let $P(v;t)$ be the probability that the system is in state *v* at time *t*. Now let us consider the event "the system is in *Good* state at time $t+\Delta t$". This event takes place if at time *t* the system was already in the *Good* set or at time *t* it was in one of its border states and went during this interval from a border state to *Good*. All other possibilities which involve more than one transition during $[t,t+\Delta t]$ have probability $o(\Delta t)$. Formally,

$$R(p_1(t+\Delta t),...,p_n(t+\Delta t))=R(p_1(t),...,p_n(t))+\sum_{v\in DN^*}P(v;t)\cdot\Gamma(v)\cdot\Delta t+o(\Delta t).$$

Transfer $R(p_1(t),...,p_n(t))$ to the left-hand side, divide both sides by Δt and set $\Delta t\rightarrow 0$. We arrive at the following relationship:

$$\frac{dR(p_1(t),...,p_n(t))}{dt}=\sum_{v\in DN^*}P(v;t)\Gamma(v) \quad (4)$$

Now, represent the left-hand side of (4) in an alternative form:

$$\frac{dR(p_1(t),...,p_n(t))}{dt}=\sum_{j=1}^{n}\frac{dR}{dp_j}\cdot\frac{dp_j(t)}{dt}=$$

$$(p_j(t)=1-e^{-\lambda_j t},q_j(t)=e^{-\lambda_j t})=\sum_{j=1}^{n}\frac{dR}{dp_j}q_j\cdot\lambda_j=$$

$$\nabla R\cdot\{q_1\lambda_1,...,q_n\lambda_n\} \quad (5)$$

Comparing (4) and (5) we arrive at the desired relationship between the gradient vector and the border state probabilities:

$$\nabla R\cdot\{q_1\lambda_1,...,q_n\lambda_n\}=\sum_{v\in DN^*}P(v;t)\Gamma(v) \quad (6)$$

From the latter formula we can get the expression for BIM of system elements in the following manner. It follows from the above proof that if instead of general vector $\{q_1\lambda_1,...,q_n\lambda_n\}$ we take specific vector $\{0,...,q_i\lambda_i,...,0\}$ then we get the following formula: $\frac{\partial R}{\partial p_i}\cdot q_i\lambda_i=\nabla R\cdot\{0,...,q_i\lambda_i,...,0\}=$

$$\sum_{\{v\in DN^*,v+(0,...,1_i,...,0)\in Good\}}P(v;1)\cdot\lambda_i \quad (7)$$

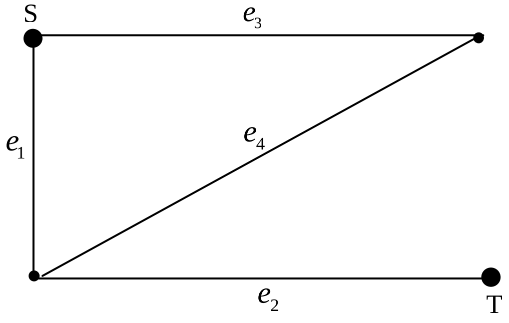
Example 1. Let us take the network given in Fig. 1 and compute the BIM for edge e_1. The appropriate border states *v* such that $v+(1,0,0,0)\in Good$ are: $S_1=(0,1,0,0)$, S_2=(0,1,1,0), $S_3=(0,1,0,1)$. Then we get by (7):

$$\frac{\partial R}{\partial p_1}\cdot q_1\lambda_1=\lambda_1\cdot(P(S_1)+P(S_2)+P(S_3))=$$

$$\lambda_1\cdot(p_2\cdot q_1\cdot q_3\cdot q_4+p_2\cdot p_3\cdot q_1\cdot q_4+p_2\cdot p_4\cdot q_1\cdot q_3).$$

Dividing the both sides of the latter expression by $q_1\lambda_1$, we arrive to the BIM of e_1.

Fig. 1



The above example demonstrates computations via formula (6). It is obvious that the main technical difficulty lies in identifying the border states and finding their probabilities. Computations similar to shown in the above example are difficult to carry out for large or moderate networks. There is however a powerful computational Monte Carlo technique based on introducing a special scheme called Evolution and Merging process, which allows efficient estimation of expressions of type (6). It was first suggested in the principal paper [2]. We will leave the relevant details outside this paper and present them and the corresponding numerical results in our forthcoming paper.

4 Spectral approach to computing network reliability importance measures

In this section we will derive the BIM and the FVIM for networks with identical elements by means of so called network combinatorial spectrum. This notion was introduced in [3] and [4,5] to estimate network lifetime distribution and /or its static reliability. For reader's convenience we remind shortly the principal idea of network combinatorial characteristic called spectrum. For simplicity we demonstrate the method for the case of reliable nodes and unreliable edges. It was shown in [4] that this approach is applicable also to the case of reliable edges and unreliable nodes, or – which is more complicated – to the case of both unreliable nodes and edges.

Let Π_E be the set of all edge permutations in *E*. Let π be a particular permutation. By sub-permutation $\pi(i)$ of π we denote a sequence constructed of the first *i* edges in π. For each sub-permutation $\pi(i)$ we define a network state $S(\pi(i))$, where all the edges in $\pi(i)$ are *up* and all other edges in π are *down*. For each edge e_j and permutation π denote by $\pi(e_j)$ the index of this edge in π.

Example 2. Let us take the network in Fig.1 and let our permutation be $\pi = (1,3,2,4)$. Then, for example, $\pi(3) = (1,3,2)$ and $S(\pi(3))$ is a state in which edges 1,3,2 are up, and edge 4 is down. We have also: $\pi(e_1) = 1$, $\pi(e_2) = 3$, $\pi(e_3) = 2$, $\pi(e_4) = 4$.

Next we define an *anchor*. This notion plays a central role in our reasoning.

Definition 3. Let $r = r(\pi)$ be the first index in permutation π so that $N(\pi(r))$ is *Good.* We say that $r(\pi)$ is the anchor of the permutation π.

Definition 4. Denote by x_i the number of all permutations π such that i is the *anchor* of π. We say that the set

$$SP = \{\{x_i\}, 1 \le i \le n\} \tag{8}$$

is the *combinatorial spectrum* of the network.

Example 3. We demonstrate these definitions on a network given in Fig. 1. The total number of permutations of 4 edges in the network is 24. Let $\pi = (3,1,2,4)$. We see that the first index such that the network state becomes *Good* is 3. Therefore $r(\pi) = r(3,1,2,4) = 3$ is the anchor of this permutation. After going over all permutations we arrive at the following combinatorial spectrum of the given network:

i	1	2	3	4
x_i	0	4	14	6

It was shown in [4] that given a network spectrum $SP = \{\{x_i\}, 1 \le i \le n\}$, the network reliability may be expressed in the following form:

$$R = \sum_{r=1}^{n} x_r \sum_{i=r}^{n} \frac{p^i \cdot q^{n-i}}{i! \cdot (n-i)!} \tag{9}$$

In our example, the network reliability is:

$$R = \sum_{r=1}^{4} x_r \sum_{i=r}^{4} \frac{p^i \cdot q^{n-i}}{i! \cdot (4-i)!} = p^4 + 3p^3 \cdot q + p^2 \cdot q^2 .$$

Remark. Sometimes it is more convenient to use the *cumulative* form of the spectrum:

$$SP^* = \{y_i : y_i = \sum_{k=1}^{i} x_i, 1 \le i \le n\} \tag{10}$$

The value y_i expresses the number of permutations π such that $r(\pi) \le i$, or, in other words, that $N(\pi(i))$ is *Good.*

Example 4. For the network on Fig. 1, we have from the above example that there are 14 permutations with anchor $r = 3$ and 4 permutations with anchor $r = 2$. So, we get $y_3 = 18$.

It is easy to check from (9) that in the case of the cumulative spectrum the network reliability is given by

$$R = \sum_{i=1}^{n} y_i \cdot \frac{p^i \cdot q^{n-i}}{i! \cdot (n-i)!} \tag{11}$$

Clearly, in the case of large or moderate networks we can not get the exact values of the spectrum. We can however try to estimate them by a Monte Carlo simulation [3,4]. It is worth to mention the main advantages of this combinatorial approach:

(a) eliminating the rare event phenomenon. This fact results in bounding the relative error, so the method is especially efficient for highly reliable networks.

(b) once computed, the *combinatorial spectrum* serves for as many values of nodes or edge failure probabilities as needed.

(c) possibility to use for solving different reliability problems in dynamic networks.

Definition 5. Denote by $z_{i,j}$ the number of all permutations π such that $S(\pi(i))$ is *Good* and $\pi(e_j) \le i$. We call the set $\{z_{i,j}, 1 \le i \le n, 1 \le j \le n\}$ - the BIM *spectrum.*

Definition 6. Denote by $v_{i,j}$ the number of all permutations π such that $S(\pi(i))$ is *Good* and $\pi(e_j) > i$ .We call the set $\{v_{i,j}, 1 \le i \le n, 1 \le j \le n\}$ - the FVIM *spectrum.*

We see from these definitions that $z_{i,j} + v_{i,j} = y_i$.

Table 1

i	y_1	$z_{i,1}$	$z_{i,2}$	$z_{i,3}$	$z_{i,4}$
1	0	0	0	0	0
2	4	4	4	0	0
3	18	12	18	12	12
4	24	24	24	24	24

Example 5. Let us take the edge e_1 from the network in Fig. 1 and let us compute $z_{3,1}$. It is easy to see that there are 6 permutations π such that $S(\pi(3))$ is *Good* and $\pi(e_1) > 3$. So we get $y_3 - z_{3,1} = 6$. From the previous example, $y_3 = 18$. Hence, we get that $z_{3,1} = 12$. The BIM spectrum for the network is given in Table 1.

Claim 1.

(a) The BIM for edge *j* is given by the following formula:

$$I_j^B = \sum_{i=1}^{n} \frac{z_{i,j}\cdot p^{i-1}\cdot q^{n-i} - (y_i - z_{i,j})\cdot p^{i}\cdot q^{n-i-1}}{i!\cdot(n-i)!} \qquad (12)$$

(b) The FVIM for edge *j* is given by the following formula:

$$I_j^{FV} = 1 - \frac{1}{R(p)}\sum_{i=1}^{n} \frac{v_{i,j}\cdot p^{i}\cdot q^{n-i-1}}{i!\cdot(n-i)!} \qquad (13)$$

Proof. (a) Remind that BIM for edge e_j equals $\frac{\partial R}{\partial p_j} = R(p_1,...1_j,...,p_n) - R(p_1,...,0_j,...,p_n)$.

The value $z_{i,j}$, by definition, is the number of permutations π such that $S(\pi(i))$ is *Good* and the edge e_j is *up*. For fixed permutation π the probability of an appropriate state with e_j being *up*, equals $p^{i-1}\cdot q^{n-i}$. Take into account that a specific state with *i* edges being *up* and *n-i* edges being *down* we obtain $i!\cdot(n-i)!$ times (from different permutations). Then the summary probability of all *Good* states with *i* edges being *up* and *n-i* edges being *down* equals $\frac{z_{i,j}\cdot p^{i-1}\cdot q^{n-i}}{i!\cdot(n-i)!}$. For the case of the edge e_j being *down* we get the expression of the appropriate probability as $\frac{(y_i - z_{i,j})\cdot p^{i}\cdot q^{n-i-1}}{i!\cdot(n-i)!}$, and (a) follows.

(b) Using (a), the definition of IS^{FV} and the above mentioned fact that $z_{i,j} + v_{i,j} = y_{i,j}$ we arrive at the desired expression.

In order to rank the elements according to their importance measure there is no need to compute the partial derivatives. The following simple claim takes place.

Claim 2. Let $\{z_{ij}, 1 \le i \le n\}$ and $\{z_{is}, 1 \le i \le n\}$ be the BIM spectrum elements for the edges e_j and e_s respectively. Then:

(a) If for all $1 \le i \le n$ the inequality $z_{ij} \ge z_{is}$ holds, then $\frac{\partial R}{\partial p_j} \ge \frac{\partial R}{\partial p_s}$. Moreover, if for at least one index it holds $z_{ij} > z_{is}$, then $\frac{\partial R}{\partial p_j} > \frac{\partial R}{\partial p_s}$.

(b) Suppose that the condition of (a) does not take place. Than let the k be the maximal index such that $z_{ij} \ne z_{is}$. Suppose that $z_{kj} > z_{ks}$.Then there exists some value p_0 such that for all $p \ge p_0$ the inequality $\frac{\partial R}{\partial p_j} > \frac{\partial R}{\partial p_s}$ holds.

Proof. (a) From (12) we have:

$$\frac{\partial R}{\partial p_j} - \frac{\partial R}{\partial p_s} = \sum_{i=1}^{n} \frac{z_{i,j}\cdot p^{i-1}\cdot q^{n-i} - (y_i - z_{i,j})\cdot p^{i}\cdot q^{n-i-1}}{i!\cdot(n-i)!} -$$

$$\sum_{i=1}^{n} \frac{(z_{i,j} - z_{i,s})\cdot p^{i-1}\cdot q^{n-i} - (z_{i,s} - z_{i,j})\cdot p^{i}\cdot q^{n-i-1}}{i!\cdot(n-i)!} =$$

$$\sum_{i=1}^{n} \frac{(z_{i,j} - z_{i,s})\cdot p^{i-1}\cdot q^{n-i-1}}{i!\cdot(n-i)!}$$ and (a) follows.

(b) From the definition of *k* and the latter expression we obtain:

$$\sum_{i=1}^{k} \frac{(z_{i,j} - z_{i,s})\cdot p^{i-1}\cdot q^{n-i-1}}{i!\cdot(n-i)!} =$$

$$p^{k-1}\cdot q^{n-k-1}\cdot\Big(\sum_{i=1}^{k} \frac{(z_{i,j} - z_{i,s})}{i!\cdot(n-i)!}\cdot\Big(\frac{q}{p}\Big)^{k-i}\Big)$$ and for $p \to 1$, the assertion follows.

We use the following Monte Carlo scheme to obtain unbiased estimates for the y_i, $z_{i,j}$ and $v_{i,j}$.

Simulation scheme.

Step1. Initialize all a_i, $b_{i,j}$ and $c_{i,j}$ to be 0.

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

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

Step4. Let $a_r := a_r + 1$.

Step5. For all j such that $\pi(e_j) \le r$ let $b_{r,j} := b_{r,j} + 1$.

Step6. For all s such that $\pi(e_s) > r$ let $c_{r,s} := c_{r,s} + 1$.

Step7. Let $r := r + 1$. If $r \le n$ Go to Step4.

Step8. Repeat steps 2-7 *M* times.

Computing $\hat{y}_i = \frac{a_i\cdot n!}{M}, \hat{z}_{i,j} = \frac{b_{i,j}\cdot n!}{M}, \hat{v}_{i,j} = \frac{c_{i,j}\cdot n!}{M}$ we can from (11), (12), (13) obtain the unbiased estimates for R, I^B and I^{FV} accordingly.

5 Numerical Example

In this section we present an example, which explains how we can rank edges in accordance to their BIM by using spectrum approach. We choose two-terminal hypercube H_4 - a network with 2^4 =16 nodes and 32 edges (numerical results for larger networks we will present in our forthcoming paper). This hypercube is shown on Fig. 2. Its terminals are

nodes 1 and 16. Edges will be denoted by (k,s) , where k and s are node numbers.
Let us note that hypercube configurations are widely used in computer networks [10].

Fig. 2

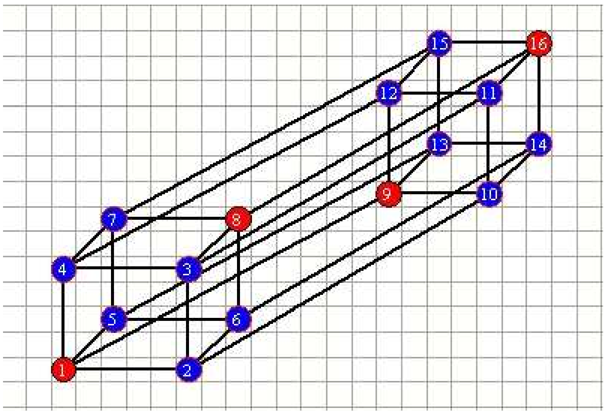


Table 2 presents a fragment of simulation results, based on 10000 replications. By a_i we marked the simulated value of spectrum and by $b_{i,(k,s)}$ - simulated value of BIM spectrum IS^B for edge (k,s). Remind that for edge ranking there is no need to compute their BIM's. We see from Table 2 that the values of IS^B for edges (1,9) and (8,16) are very close to each other. On the other side, these values are consistently greater than the BIM spectrum values for edge (1,2) and the latter – than those of (3,4). So, we rank the edges by their importance in the following order (read table 2 in horizontal direction) $(1,9)=(8,16)>(1,2)>(3,4)$.

Table 2

i	a_i	$b_{i,(1,9)}$	$b_{i,(8,16)}$	$b_{i,(1,2)}$	$b_{i,(3,4)}$
6	2	**2**	**1**	0	1
7	15	**14**	**14**	2	3
8	59	**51**	**50**	12	12
9	154	**129**	**128**	40	38
10	350	**286**	**267**	109	91
11	679	**501**	**492**	235	206
12	1333	**886**	**904**	525	438
13	2385	**1478**	**1492**	996	892
14	3723	**2187**	**2210**	1625	1547
15	5230	**3012**	**3042**	2502	2363

Note that from the whole data file one can infer that in the given network there are three, by their BIM's, different groups of edges. The first is the pair of edges (1,9) ad (8,16). The second is the group consisting of all other edges incident to the terminals 1 and 16, and the third – all other edges. This conclusion may seem to be intuitively obvious, but for the same hypercube with $k>2$ nonsymmetrical terminals or for other, nonsymmetrical networks similar conclusions are not so clear. Here the ranking is obvious in spite of random fluctuations of the $b_{i,(k,s)}$ values. In more involved cases there will be a need to use statistical tools for better discrimination between equally important edge groups.

4 Conclusions

(1) To the best of our knowledge, very few works were conducted on computational problems of reliability importance values evaluation for large or moderate networks.

(2) The proposed method for the general case of different edges is efficient, since it is based on Monte Carlo model with well-established efficiency. Note that one of the advantages of this method is that in one simulation session we can evaluate importance measures for different edges and for different given edge probabilities.

(3) The spectrum approach is highly efficient and has many advantages. One of them is that constructed spectrum does not depend on the edge probabilities and reflects the topological features of a network.

(4) In many practical situations, the ranking of edges by their importance measures may be done without computing appropriate probabilities, but only using the BIM spectrum.

(5) The two proposed methods may be easily implemented for networks with reliable edges and unreliable nodes, and also (this is technically more difficult) for networks with both unreliable nodes and edges.

(6) Our methods can be considered as the first step towards optimal network design.

References:

[1] 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*, Vol. 6, 1992, pp. 457-469.

[2] T. Elperin, I. Gertsbakh, M. Lomonosov, Estimation of network reliability using graph evolution models, *IEEE Transactions on Reliability*, Vol. 40, No.5, 1991, pp. 572-581.

[3] Gertsbakh and Y. Shpungin, Combinatorial approaches to Monte Carlo estimation of network

lifetime distribution, *Appl. Stochastic Models Bus. Ind.*, Vol 20, 2004, pp. 49-57.

[4] Y. Shpungin, Combinatorial Approach to Reliability Evaluation of Network with Unreliable Nodes and Unreliable Edges, *International Journal of Computer Science*, Vol.1, No.3, 2006, pp. 177-183.

[5] Y. Shpungin, Networks with unreliable nodes and edges: Monte Carlo lifetime Estimation, *International Journal of Applied Mathematics and Computer Sciences,* Vol. 4, No. 1, 2007, pp. 168-173.

[6] Z. W. Birnbaum, *On the importance of different components in a multicomponent system, Multivariate Analysis 2,* New York, Academic Press, 1969.

[7] J. Fussel, How to calculate system reliability and safety characteristics, *IEEE Transactions on Reliability,* Vol. 24, No.3, 1975, pp. 169-174.

[8] F. C. Meng, Comparing the importance of system elements by some structural characteristics, *IEEE Transctions on Reliability,* Vol 45, No 1, 1996, pp. 59-65.

[9] A. Gandini, Importance ad sensitivity analysis in assessing system reliability, *IEEE Transactions on Reliability*, Vol. 39, No.1,1990, pp. 61-70.

[10] M. Mitzenmacher, E. Upfal, *Probability and Computing . Randomized Algorithms and Probabilistic Analysis,* Cambridge University Press, 2005.