SPECTRAL APPROACH TO RELIABILITY EVALUATION OF FLOW NETWORKS

Ilya Gertsbakh[(a)], Yoseph Shpungin[(b)]

[(a)] Department of Mathematics, Ben-Gurion University,
Beer-Sheva, Israel.
[(b)] Software Engineering Department,
Shamoon College of Engineering, Beer Sheva, Israel.

[(a)] elyager@bezeqint.net, [(b)] yosefs@sce.ac.il

ABSTRACT

In this paper we consider flow networks, which is an important class of networks, and which includes, for example, communication networks, transportation and supply networks, oil and power supply systems, etc. In our model, the edges are subject to failure, which may be caused by "enemy attack", earthquakes, disruption of communication channels etc. Each edge is characterized by its failure probability and flow capacity. The network reliability is defined as the probability that the flow between the source node and sink node is not less than some given threshold. Our approach to flow network reliability evaluation is based on estimating by means of an efficient Monte Carlo simulation, the network *topological invariant* called network destruction *spectrum* (D-spectrum). We consider also a design problem on flow network, namely its edge reinforcement in order to increase in an "optimal" way the network reliability.

Keywords: flow network, Monte Carlo simulation, D-spectra, network reliability design

1. INTRODUCTION

The maximum flow problem is a standard problem in operations research first solved by Ford and Fulkerson (Ford and Fulkerson 1962). They assumed that edges (and nodes) have some given nonrandom flow capacities. In stochastic flow network, it is assumed that edge (and/or node) capacities are *random*, which greatly complicates the problem and its solution.

There is a vast literature on stochastic flow networks, see (Lin 2004, 2001, Ramirez-Marquez and Coit 2005, Younes and Hassan 2011) and references there. Typically, the reliability of a stochastic flow network is measured by the probability $P(M \geq \Phi)$ that the maximal flow M which can be delivered from source *s* to sink *t* will be no less than some critical value Φ.

When network edges may be in two states (*up*/*down*), and $\Phi = 0$, the reliability of the network reduces to so-called *s-t* connectivity which, contrary to the classical maximum flow problem, is already NP-complete since its solution is based on enumeration of all *s-t* paths. Quite sophisticated methods have been developed to solve this problem by applying Monte Carlo methodology, see e.g. (Elperin, Gertsbakh and Lomonosov 1991).

Further development in network reliability studies has been made by assuming that edges capacity is an integer-valued random variable (Lin 2001). The proposed solution method is based on finding all boundary points (i.e. all such path sets) which allow to deliver the minimal demand flow Φ, and on applying the inclusion-exclusion techniques to compute the desired reliability. The applicability of this method is limited to rather small networks. The work (Ramirez-Marquez and Coit 2005) deals with a similar multistate model and introduces Monte Carlo (MC) approach. The MC is based on comparing already identified elements of the set of all multistate minimal cut vectors with randomly generated system state vectors.

Also genetic algorithms have been applied for reliability evaluation of stochastic flow networks, see e.g. (Younes and Hassan 2011).

The purpose of this paper is to demonstrate how a new methodology based on so-called D-spectra and BIM-spectra can be used for analysis and design of flow network reliability.

2. BASIC NOTIONS AND DEFINITIONS

2.1. Flow Network

We define *flow network* ***N*** as a pair (*V*, *E*), where *V* is a node-set and E is a set of directed edges. In our model, nodes can never fail, while edges can. If an edge fails, we say it is *down*; otherwise it is *up*. For each edge *e*, the probabilities $p(e)$ of being *up* and $q(e) = 1 - p(e)$ of being *down* are defined. Edges are assumed to be stochastically independent. In addition, for each edge *e*=(*a*,*b*) directed from the node *a* to the node *b*, we define the maximal flow $c(e)$ which can be delivered from *a* to *b* along this edge.

Also let $|V|=n, |E|=m.$ By state of a network we call a binary vector $(x_1, x_2, ..., x_m),$ where $x_i=1,$ if an edge is *up* and $x_i=0,$ otherwise. We say that the network state is *UP* if the maximal flow from source to sink is not less than some given value Φ, and the state is *DOWN*, otherwise.

Example 1. Figure 1 below represents very simple flow network with 4 nodes and 4 edges.

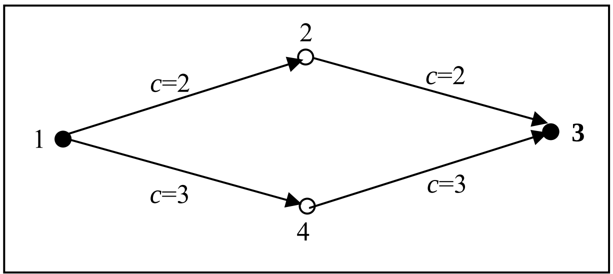

Figure 1: Flow Network with 4 Nodes and 4 Edges

It is easy to check that the maximal flow from source s=1 to sink t= 3 equals 5. For example, it may be obtained by the following flows $w(i,j)$ ((i,j) is an edge defined by the nodes *i* and *j*): $w(1,2)=2, w(1,4)=3, w(2,3)=2, w(4,3)=3.$

Suppose that we define the *UP* state for this network as a state with maximal flow no less than 3. Then, if for example the edge (1,4) is *down*, the maximal flow equals 2, and the network is in the *DOWN* state.

2.2. D-Spectrum

Let us now introduce the so-called destruction spectrum (D-spectrum), which will play a central role in our further network flow analysis.

It is important to stress that the D-spectrum is a purely topological characteristic of the network which depends only on its structure and network *DOWN* state definition. D-spectrum is completely separated from any information regarding the real stochastic mechanism which governs system failure appearance.

Definition 1. (Gertsbakh and Shpungin 2009) *Let $\pi=(e_{i_1},...,e_{i_m})$ be a permutation of network edges. Suppose that initially all edges are up. Start turning them from up to down by moving along π from left to right. Fix the first element e_{i_r} when the network state becomes DOWN. The ordinal number of this edge in the permutation is called the anchor of π and denoted $r(\pi)$.*

Consider now the set of all $m!$ permutations and assign to each permutation probability $1/m!$. Define the probability of the event $A(i)=\{r(\pi)=i\}$ as

$$f_i = P(A(i)) = \frac{\#\text{ of permutations with } r(\pi)=i}{m!} \quad (1)$$

Definition 2. (Gertsbakh and Shpungin 2009) *The discrete density function $\{f_i\}, i=1,2,...,m,$ is called the system destruction spectrum (D-spectrum). $F(x)=\sum_{i=1}^{x} f_i, x=1,...,m$ is called the cumulative D-spectrum.*

Example 1 (continued). Let us demonstrate the notion of *D*-spectrum on a network given in Figure 1. Suppose that $\Phi=3.$ The total number of permutations of 4 edges in the network is 24. Let $\pi=((1,2),(2,3),(4,3),(1,4)).$ We see that the first index such that the network state becomes *DOWN* (that is the maximal flow is less than 3) is 3. Therefore $r(\pi)=3$ is the anchor of this permutation. After going over all permutations we arrive at the following D-spectrum of the given network: $f_1=\frac{1}{2}, f_2=\frac{1}{3}, f_3=\frac{1}{6}.$ The cumulative D-spectrum is therefore: $F(1)=\frac{1}{2}, F(2)=\frac{5}{6}, F(3)=F(4)=1.$

Theorem 1. (Gertsbakh and Shpungin 2009) *Suppose that all network edges have equal down probabilities, i.e. $q_i \equiv q.$ Then the probability that network is in the DOWN state is given by the following formula:*

$$P(DOWN)=\sum_{i=1}^{m} F(i) q^i p^{m-i} \frac{m!}{i!(m-i)!} \quad (2)$$

Rather surprising relationship (2) established in this theorem follows from the fact that the number of network failure sets $C(x)$ of size $x, x=1,...,m$ can be expressed via the D-spectrum $F(x)$ by means of the following simple combinatorial relationship:

$$F(x)=\frac{C(x)}{m!/(x!(m-x)!)} \quad (3)$$

Formula (3) says that $F(x)$ is the ratio of the number of failure sets of size *x* among all possible sets of size *x* constructed from *m* different elements. This fact, in turn, follows from the definition of the cumulative D-spectrum $F(x)$.

It follows from (3) and independence of network edges that the probability associated with failure sets of size x equals $C(x)\cdot q^x \cdot (1-q)^{m-x}$. Now (2) follows from the fact that the network is *DOWN* if and only if it is in one of its failure states.

Example 1 (continued). Returning to our example 1, we calculate from (3) that there are *C*(1)=2 failure sets of size 1, *C*(2)=5 failure sets of size 2, *C*(3)=4 failure sets of size 3 and one failure set of size 4. For example, the failure sets of size 2 are all pairs of edges except of the pair ((1,2), (2,3)). We have by the theorem: $P(DOWN) = 2qp^3 + 5q^2p^2 + 4q^3p + q^4$.

Now we will introduce so-called Birnbaum Importance Measure (BIM) (Barlow and Proschan 1975) for system components. In simple words, BIM of edge *j* (denoted BIM_j) is the gain of network reliability obtained by replacing *down* edge *j* by an absolutely reliable one. Formally, BIM_j is defined as follows.

Definition 3.
$\text{BIM}_j = G(p_1,...0_j,..., p_m) - G(p_1,...,1_j,..., p_m)$, *where* $G(p_1,...0_j,..., p_m)$ *is the probability that the network is DOWN when edge j is down, and* $G(p_1,...,1_j,..., p_m)$ *is the probability that the network is DOWN when the edge j is up.*

The important role played by BIM_j follows from the fact that BIM_j equals the partial derivative of system reliability function $R(p_1,..., p_m) = 1 - G(p_1,..., p_m)$ with respect to p_j, see (Barlow and Proschan 1975).

The knowledge of edge BIMs is the key element in finding the optimal network reinforcement strategy. The use of BIM in reliability practice was very limited since typically the system reliability function $R(p_1,..., p_m)$ is not available in explicit form.

It turns out that in the case of equal component reliability there is a surprising connection between the BIMs and the network D-spectrum and its modification called BIM-spectrum which allows estimating and ranking the component BIMs without knowing the analytic form of system reliability function.

Definition 4. (Gertsbakh and Shpungin 2009)
Let $N(x;0_j)$ *be the number of permutations satisfying the following two conditions:*

(i) *If the first x edges in the permutation are down, then the network is DOWN*;
(ii) *Edge j is among the first x elements of the permutation*.

The collection
$\{z(x, j) = N(x;0_j) \cdot x!(m-x)!/m!\}$
for a fixed j and $x = 1, 2,..., m$ *is called the* BIM_j *– spectrum of edge j.*

The collection of all $\{z(x; j), x = 1, 2,..., m\}$ *for* $j = 1,..., m$ *is called the network* BIM-*spectrum.*

Let $N(x)$ be the number of permutations satisfying (i) only. Denote by $N(x;1_j) = N(x) - N(x;0_j)$.

Theorem 2. (Gertsbakh and Shpungin 2009) *Let* $p_i \equiv p, q = 1 - p$. *Then*

$$BIM_j = \sum_{x=1}^{m} \frac{1}{x!(m-x)!} \cdot (N(x;0_j)q^{x-1}(1-q)^{m-x} - N(x;1_j)q^x(1-q)^{m-x-1}) \quad (4)$$

The hint to the proof of this theorem is the following : the first sum in (4) equals the first term $G(p_1,...0_j,..., p_m)$ in the expression of BIM_j (**Definition** 3), and the second sum in (4) – to the second term in BIM_j in the same **Definition** 3.

Theorem 3. (Gertsbakh and Shpungin 2009)
If for all $1 \le x \le m$ *the inequality* $z(x,i) \ge z(x, j)$ *holds then* $BIM_i \ge BIM_j$, *no matter what the values of q are.*

Suppose that the previous condition does not take place. Than let the k be the maximal index such that $z(x,i) \ne z(x, j)$. *Suppose that* $z(k,i) > z(k, j)$.
Then there exists some value p_0 *such that for all* $p \ge p_0$ *the inequality* $BIM_i > BIM_j$ *holds.*

Example 1 (continued). Let us take the edge (1,2) from the network in Figure 1 and compute $z_{2,(1,2)}$. It is easy to see that there are 8 permutations π such that the network is *DOWN* when the two first edges of π are *down*, and the edge (1,2) is one of them. So $z_{2,(1,2)}$=8/24. We have for this edge the following BIM-spectrum:

$$z_{1,(1,2)} = 0, z_{2,(1,2)} = \frac{8}{24}, z_{3,(1,2)} = \frac{18}{24}, z_{4,(1,2)} = 1.$$

For edge (2,3) we have the same BIM-spectrum, and for edges (1,4) and (4,3) the BIM-spectrum is the following (*x* stands for the edge (1,4) or the edge (4,3)).

$$z_{1,x} = \frac{6}{24}, z_{2,x} = \frac{12}{24}, z_{3,x} = \frac{18}{24}, z_{4,x} = 1.$$

We see from this example that by Theorem 3, the BIMs of the edges (1,4) and (4,3) are greater than those of the edges (1,2) and (2,3) for all values of *q*.

3. MONTE CARLO FOR D-SPECTRA AND BIM-SPECTRA

Exact computation of D-spectra and BIM-spectra is an NP-hard problem. The practical way to calculate the spectra is approximating them using Monte Carlo (MC) methodology. The books (Gertsbakh and Shpungin 2009, 2011a) contain a series of efficient MC algorithms and examples of spectra calculation.

We give here a non-formal explanation of MC algorithms adopted to our purpose.

To estimate *F*(x), simulate *M* random permutations $\pi = (i_1, i_2,..., i_m)$ of edge numbers and imitate a sequential destruction of edges by moving along a permutation from left to right and by remembering the number N_i of such permutations that the system went

DOWN on the *i*-th step of the destruction process. Afterwards, as an MC estimate of $F(x)$ take the ratio

$$\hat{F}(x) = (N_1 + ... + N_x) / M.$$

Note that in order to check the state of the network on certain step of the destruction process, we use the Ford-Fulkerson algorithm (Cormen, Rivest, Leicerson, and Stein 2009) for calculating the maximal flow. If it turns out that the maximal flow is less than Φ, we say that the network is *DOWN*.

An important fact is that there is *no need* to check the network state on *each* step of the destruction process. The position of the anchor in a given permutation $\pi = (e_{i_1}, ..., e_{i_m})$ can be efficiently found by applying *bisection* search algorithm, which works as follows. Erase the first $\left\lfloor \frac{m}{2} \right\rfloor$ edges of the permutation. Check the state of the network using Ford-Fulkerson algorithm. If the network is already *DOWN*, the anchor should be among the first $\left\lfloor \frac{m}{2} \right\rfloor$ positions. If the network is *UP*, the anchor is among the remaining part of the permutation. Proceed in a similar way by bisecting the relevant part of the permutation until the position of the anchor is located. On the average, the number of flow checks is of magnitude $O(\log_2(m))$.

To approximate BIM-spectra, modify the above procedure and count the number of permutations $M(x; 0_j)$ equal to the number of permutations such that the system went *DOWN* during the first *x* failures *and* edge *j* was among these *x* components.

Example 1 (continued). Let us illustrate the D-spectrum calculation on the network in Figure 1. Take the number of permutations *M*=5. Denote the network edges: (1,2)=1, (1,4)=2, (2,3)=3, (4,3)=4. Suppose that the generated permutations are: $\pi_1 = (1,2,3,4)$, $\pi_2 = (4,3,2,1), \pi_3 = (1,3,2,4), \pi_4 = (2,1,3,4)$, $\pi_5 = (3,1,2,4)$. We see that in these five permutations the network went *DOWN* twice on the first step, once on the second step, and twice on the third step. So the estimators for $F(x)$ are the following: $\hat{F}(1) = \frac{2}{5}, \hat{F}(2) = \frac{3}{5}, \hat{F}(3) = \hat{F}(4) = 1$. Naturally, these values are far from the exact values (calculated above), because the number *M* of replications is too small.

Remark. Suppose that edge *up* probability *p* is not known exactly (as it usually takes place in practice) and lies in the interval $p \in [p_{\min}, p_{\max}]$. Since network reliability is a monotone function of its component reliability, we have the following bounds on $R(p)$: $R(p_{\min}) \le R(p) \le R(p_{\max})$. These bounds may be quite valuable in case of "fuzzy" information about the *q* values.

4. NUMERICAL EXAMPLES

4.1. Network Reliability as a Function of *p*

Let us consider the network from Figure 2. This network has 15 nodes (two of which are terminals, 1 and 2) and 35 edges.

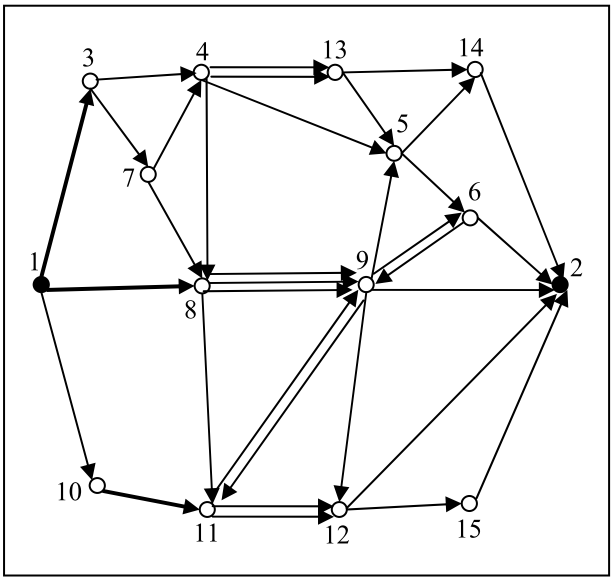

Figure 2: Flow Network with 15 Nodes and 35 Edges

The edge capacities $c(a,b)$ are given by the Table 1. In the case two nodes, say *x* and *y* are connected by some parallel edges, we denote them by $(x,y)_1$, $(x,y)_2$ and so on. We consider also edges (a,b) with flow going from a to b and in opposite direction. For example, we see in the table the edges (4,8) and (8,4).

The initial maximal flow for the given capacities equals 22.

Table 1: Edge Capacities

a,b	*c*(*a,b*)	*a,b*	*c*(*a,b*
1,3	8	$(8,9)_3$	3
1,8	9	8,11	4
1,10	8	9,5	6
3,4	6	9,6	4
3,7	6	9,2	5
4,5	6	9,11	4
4,8	6	9,12	5
$(4,13)_1$	5	10,11	5
$(4,13)_2$	2	11,9	4
5,6	6	$(11,12)_1$	4
5,14	5	$(11,12)_2$	2
6,2	6	12,2	6
6,9	3	12,15	5
7,4	5	13,14	5
7,8	4	13,5	5
8,4	4	14,2	5
$(8,9)_1$	5	15,2	5
$(8,9)_2$	4		

The most important characteristic of the network is its reliability $R(p) = 1 - P(DOWN; q)$ as a function of p, $p = 1 - q$. Let us demonstrate how this characteristic is computed using D-spectrum, for two values of the threshold: Φ =10 and Φ =12. Table 2 presents the edge cumulative spectra obtained by means of Monte Carlo simulation of $5 \cdot 10^4$ edge permutations. For example, the row for x=6 gives the probabilities 0.12924 and 0.20202 that the anchor for Φ =10 and Φ =12, respectively, is on one of the first six positions in random edge permutations.

Table 2: Edge Cumulative Spectrum

x	$F(x)$ Φ =10	$F(x)$ Φ =12	x	$F(x)$ Φ =10	$F(x)$ Φ =12
1	0	0	14	.81732	.93196
2	.00874	.01144	15	.88206	.96366
3	.02532	.0363	16	.92814	.98202
4	.05094	.0759	17	.96014	.992
5	.0851	.13134	18	.9795	.99692
6	.12924	.20202	19	.9903	.99916
7	.18566	.28702	20	.99608	.99968
8	.25376	.387	21	.9984	.99998
9	.33694	.49746	22	.99938	1
10	.43354	.61082	23	.99976	1
11	.53428	.7172	24	.9999	1
12	.63944	.81052	25	1	1
13	.735	.88198	26	1	1

With probability $\approx$ 0.92 the anchor is greater than 5, for $\Phi = 10$. So, with probability close to 0.92 network failure takes place after 5 edges have failed. For $\Phi = 12$, with probability close to 0.92 network failure occurs after 4 edges have failed.

Table 3 presents network *UP* state probabilities for various values of *p*. The calculations were performed using formula (2).

Table 3: *R*(*p*)

p	*R*(*p*) Φ =10	*R*(*p*) Φ =12
0.1	0	0
0.2	.000032	.000002
0.3	.001121	.000159
0.4	.0129	.0034
0.5	.0725	.0286
0.6	.2375	.1309
0.7	.5095	.3669
0.8	.7819	.6826
0.85	.8809	.8206
0.9	.9485	.9223
0.95	.9872	.9816
0.975	.9968	.9956

We see from the table that if, for example, we want to guarantee that the network is *UP* with probability greater than 0.95, it is enough to demand that edges be *up* with *p*>0.9, for Φ =10.

4.2. Edge Reinforcement Problem

By reinforcing an edge we mean replacing it by a more reliable one. This operation can be applied to a given number k of edges. The problem is to achieve the maximal network reliability by "the best possible" choice of the candidates for this replacement. In the case of equal edge probabilities we suggest the following method (Gertsbakh and Shpungin 2009, 2011a).

1. Estimate the BIM-spectra for all edges.
2. Range the edges by their BIM's spectra.
3. Take the first k edges with the highest BIM values and replace them by more reliable ones.

Note that this method is based on Theorem 3 from the Section 2.

Let us consider the network from Figure 2 and let $\Phi = 10$. Suppose that we can reinforce 3 edges. We skip here the intermediate results of the calculations. The final results are the following. The edges (1,3), (1,8), and (10,11) must be reinforced (they are marked bold in Figure 2). This conclusion may seem to be intuitively obvious, but for larger and more complicated networks similar conclusions are not so clear.

For illustration, the following Table 4 gives estimated values $\hat{z}(x;(a,b))$ of BIM-spectrum for one of the most important edges - (10,11), and a less important edge (13,14), for even x values.

Table 4: The BIM's Spectra

x	$\hat{z}(x;(10,11))$	$\hat{z}(x;(13,14))$
2	.0037	0
4	.0207	.0034
6	.0524	.0157
8	.1008	.0474
10	.1700	.1106
12	.2592	.2056
14	.3534	.3183
16	.4386	.4217
18	.5104	.5036
20	.5702	.5695
22	.6294	.6278
24	.6838	.6830
26	.7423	.7419
28	.8008	.7997
30	.8568	.8564
32	.9152	.9140
34	.9720	.9708

We see that the values of BIM-spectrum for edge (10,11) are consistently greater than those of edge (13,14). Remind that this means that (10,11) is more important than (13,14), no matter what the values of *q* are.

Remark. As it was noted, the BIM spectrum is a topological invariant for most reliability criteria, but in the case of flow networks the edge capacities may affect the spectrum. Nevertheless also in this case the topological features of the edge prevail on its capacity value influence.

To illustrate the remark, let us take from the network on Figure 2 two edges: (1,10) with capacity $c(1,10)=8$, and (10,11) with $c(10,11)=5$. In spite of the difference in capacities, all values of BIM-spectrum of edge (10,11) are consistently greater of those of edge (1,10).

For the second example take three parallel edges connecting nodes 8 and 9. We see from table 1 that $c((8,9)_1) > c((8,9)_2) > c((8,9)_3)$. Clearly that these three edges have the same topological features, but here the edge capacities affect the BIM-spectra, and we have $z(x,(8,9)_1) > z(x,(8,9)_2) > z(x,(8,9)_3)$.

Note that introducing parallel edges is a way of having edges with more than two states. For example, two independent edges connecting nodes 4 and 13 may be viewed as one edge with four possible capacities 0, 2, 5 and 7 with probabilities $q^2, q(1-q),\ q(1-q)$, and $(1-q)^2$, respectively.

5. POSSIBLE EXTENSIONS

To the best of our knowledge, there are no publications devoted to using network D-spectra technique to the study of network flow behavior in networks with unreliable edges. The model considered in this paper can be extended in several directions.

First, our method may be easily extended to the case of unreliable nodes.

Second, we can introduce several flow sources and several sinks.

Finally, introducing cost for edge reinforcement (for the purpose of increasing their reliability and/or flow capacity) will bring us to the search for the "best" predisaster design, similar to the study made by (Gertsbakh and Shpungin 2009, 2011b, Levitin, Gertsbakh, and Shpungin 2010, Peeta, Salman, Gunnec, and Kannan 2010).

REFERENCES

Barlow, R.E. and Proschan, F., 1975. *Statistical Theory of Reliability and Life Testing: Probability Models.* Holt, Rinehart and Winston, Inc.

Cormen, T.H., Rivest, R.L., Leicerson, C.E., Stein, C., 2009. *Introduction to Algorithms.* 3rd ed. MIT Press.

Elperin, T., Gertsbakh, I., Lomonosov, M., 1991. Estimation of network reliability using graph evolution models. *IEEE Transactions on Reliability,* 40(5), 572-581.

Ford, L.R. and Fulkerson, D.R., 1962. *Flows in Networks.* Princeton University Press.

Gertsbakh, I. and Shpungin, Y., 2009. *Models of Network Reliability: Analysis, Combinatorics, and Monte Carlo*. CRC Press.

Gertsbakh, I. and Shpungin, Y., 2011a. *Network Reliability and Resilience.* Springer.

Gertsbakh, I. and Shpungin, Y., 2011b. Predisaster Design of Transportation Network. *Transport and Telecommunication,* 12(1), 4-11.

Levitin, G., Gertsbakh, I., and Shpungin, Y., 2010. Evaluating the damage associated with intentional network disintegration. *Reliability Engineering and System Safety*, 96(4), 433–439.

Lin, Yi-Kuei, 2001. A simple algorithm for reliability evaluation of a stochastic-flow network with node failure. *Computers & Operations Research*, 28, 1277-1285.

Lin, Yi-Kuei, 2004. Reliability of stochastic-flow network with unreliable branches & nodes, under budget constraints, *IEEE Transactions on Reliability*, 53(3), 381-387.

Peeta, S., Salman,, S.F., Gunnec, D., and Kannan, V., 2010. Predisaster investment decisions for strengthening a highway network. *Computers and Operations Research*, 37, 1708–1719.

Ramirez-Marquez, J.E. and Coit D.W., 2005. A Monte Carlo simulation approach for Approximating multi-state two-terminal reliability, *Reliability Engineering and System Safety*, 87, 253-264.

Younes, A. and Hassan, M.R., 2011. A Genetic algorithm for reliability evaluation of stochastic flow network with node Failures. *International Journal of Computer Science and Security*, 4(6), 528-536.

AUTHORS BIOGRAPHY

Ilya B. Gertsbakh is Professor Emeritus in the Department of Mathematics at Ben Gurion University, Beer Sheva, Israel. He received his Ph.D. degree in applied probability and statistics from Latvian Academy of Sciences, Riga. He is author or coauthor of six books. He has published some 70 papers in international scientific journals.

Yoseph Shpungin is Head of the Software Engineering Department, at Shamoon College of Engineering, Beer Sheva, Israel. He received his Ph.D. degree in mathematics from Ben Gurion University, Beer Sheva, Israel. He is author of two books (with Ilya Gertsbakh) and many publications in international scientific journals.