Reliability of a Network with Heterogeneous Components

Ilya B. Gertsbakh, Yoseph Shpungin and Radislav Vaisman

Abstract We investigate reliability of network-type systems under the assumption that the network has $K > 1$ types of i.i.d. components. Our method is an extension the D-spectra method to K dimensions. It is based on Monte Carlo simulation for estimating the number of system failure sets having k_i components of i-th type, $i = 1, 2, \ldots, K$. We demonstrate our approach on a Barabasi-Albert network with 68 edges and 34 nodes and terminal connectivity as an operational criterion, for $K = 2$ types of nodes or edges as the components subject to failure.

Keywords Network terminal reliability · Several types of components · Two-dimensional spectrum · Monte Carlo simulation · Two-dimensional quantile

1 Introduction

Networks play a major role as critical infrastructures underpinning our societies and economies. Very often networks function in the presence of various disruptions from hacker attacks, natural disasters like earthquakes and natural degradations, as well as unforseen military and terrorist strikes [2, 5, 8, 11, 15, 17]. All these circumstances create growing interest to the problems of network robustness, reliability, and pre-disaster management [2, 4, 5, 15]. Reliability and resilience of network-type structures attracted major attention in the framework of general network theory, see e.g.

I.B. Gertsbakh (✉)
Department of Mathematics, Ben-Gurion University, Beer-Sheva, Israel
e-mail: elyager@bezeqint.net

Y. Shpungin
Software Engineering Department, Sami Shamoon College of Engineering, Beer-Sheva, Israel
e-mail: yosefs@sce.ac.il

R. Vaisman
School of Mathematics and Physics, The University of Queensland, Brisbane, Australia
e-mail: r.vaisman@uq.edu.au

A. Lisnianski et al. (eds.), *Recent Advances in Multi-state Systems Reliability*, Springer Series in Reliability Engineering, DOI 10.1007/978-3-319-63423-4_1

[2, 5, 9, 13, 17]. Typically, the basic model of a network functioning in the presence of random "attacks" on its nodes or edges assumed random structure of the network itself, like Poisson or Barabasi-Albert [1, 3, 9] and focused on random and independent removal of nodes and edges. The components subject to failure were assumed to be identical and independent and the network failure criterion was network disintegration or disappearance of the so-called giant component [2, 9, 11]. The research in this direction was successfully advanced in [1] by using the results of percolation theory which provided the threshold value of network components to be removed to cause network failure. The limitation of this approach, however, is that it is not applicable to some other network failure criteria, like loss of terminal connectivity, decrease of the largest network component below some critical size (for finite networks), and network disintegration into critical number of isolated clusters [5, 6].

A very promising direction in the reliability study of network-type structures is the use of so-called signatures, first suggested by Samaniego [12–14]. The essential feature of this approach is that it is based on system structural invariant which depends only on system structure function and does not depend of probabilistic properties (like lifetime distribution) of system components. Despite its elegance and universality with respect to system failure criteria, it has been efficiently applied only to systems consisting of one type i.i.d. or exchangeable components.

The main purpose of the present work is to extend the signature (or so-called D-spectra) approach [5, 6] to network systems consisting of several groups of i.i.d. components. As a principal example to illustrate our approach and its abilities we consider a transportation (or supply) network of realistic size (34 nodes, 68 edges), having as the operational criterion the terminal connectivity. We consider the case when the nodes or the edges are subject to failure. In both cases, the components subjected to failure consist of two different groups of i.i.d. components.

The exposition in the paper is the following. Our approach is an extension of the D-spectra methodology to the case of heterogeneous network. Therefore, we start with a short overview of the D-spectra approach to the systems consisting of one-type components. In this case, the D-spectrum or signature allows to count the number $C(k)$ of failure sets having k failed components. With the knowledge of $C(k)$, system *DOWN* probability can be expressed automatically. Since the case of $K > 2$ groups of independent components is a rather straightforward generalization of the case of $K = 2$ groups of components, we devote the main part of Sect. 3 to the description of our approach to the $K = 2$ case.

When the system has two types of components, the key to the reliability analysis is estimation of the number $C(k, r)$ of so-called (k, r)-failure sets which have k and r failed components of the first and the second type, respectively. $C(k, r)$ are system structural invariants. Similar to the one-dimensional case, the estimation of $C(k, r)$ is made via the so-called *two-dimensional spectrum* which estimates the frequencies of the (k, r)-failure sets in a sample of simulated random permutations. We present an efficient Monte Carlo algorithm for estimating the two-dimensional spectrum.

In Sect. 3, we demonstrate how our approach works for a realistic example of a transportation/supply network with 34 nodes and 68 edges. The network was designed by using Barabasi-Albert preferential attraction method [1]. We consider the case of edge failures and two versions of node failures. We demonstrate how relocation of so-called strong nodes can change network reliability.

Section 4 is devoted to the analysis of the network failure state under a random attack on network nodes by a two-type shocks process. Our analysis allows to define two-dimensional quantile area for the random location of the "hitting point" of network failure. Finally, in the last Sect. 5 we present the formulas generalizing our approach for $K > 2$ types of components and some concluding remarks.

2 The Principal Model: Two Types of Components

2.1 Network Description

Our basic model is a network $N = (V, E, T)$ where V is a set of vertices (nodes), $|V| = n + k$, E is a set of edges (links), $|E| = m$, and T is a set of special nodes called *terminals*, $|T| = k$, $T \subset V$. Components subject to failures are either the links or the nonterminal nodes. Edge failure means that this edge is erased, nonterminal node failure means that all edges incident to this node are erased. In this paper we consider only one form of network *DOWN* state-so-called loss of terminal connectivity which means the network is *DOWN* if not all its terminal nodes are mutually connected.

In this section we consider the case when components subject to failure (nodes or edges) consist of two independent groups of i.i.d. components having lifetime CDF $H_1(t)$ and $H_2(t)$. So, if the edges fail, m_i edges have lifetime CDF $H_i(t)$, $i = 1, 2, \ldots$ and $m_1 + m_2 = m$, and nodes remain absolutely reliable. If the nodes fail, then n_i nodes have i.i.d. lifetimes $H_i(t)$, $i = 1, 2, \ldots$ and $n_1 + n_2 = n$, and edges remain absolutely reliable.

To simplify the exposition, we consider in detail the case of two groups of components in the network. Extension to $K > 2$ groups is straightforward and is left for Sect. 5.

2.2 One Type of Components

Since the case of two-type of component network is almost a straightforward generalization of our method of dealing with the standard one-type case, we remind shortly the basic definitions and principal steps for the "standard" situation where all components have i.i.d. lifetimes with CDF $H(t)$.

Let $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ be the network component state vector. $x_i = 1/0$ if the i-th component is *up*/*down* respectively.

Network state is determined via a binary function $\varphi(\mathbf{x})$ which is 1 or 0 if the network is *UP* or *DOWN*, respectively. If $\varphi(\mathbf{x}^*) = 0$, $\mathbf{x}^*$ is called a failure vector. If we ignore the order of *up /down* components in this vector, then $\mathbf{x}^*$ determines a *failure set*, i.e. a set of j *down* components and $n-j$ *up* components. For simplicity, we call $\mathbf{x}^*$ *failure set*.

Now define D-spectrum or signature for our network. Let us consider a random permutation of component numbers

$$\pi = (i_1, i_2, \ldots, i_n).$$

Suppose that all components are *up* and, moving from left to right, we turn them *down*. The network state is controlled on each step of this destruction process.

Definition 1.1 The ordinal number in the permutation π of the component whose turning down causes network state change from *UP* to *DOWN* is called the *anchor* of this permutation.

Assume that the permutations π are taken randomly and independently from the set of all $n!$ permutations. Then the anchor becomes a discrete random variable with support $\{1, 2, \ldots, n\}$.

Definition 1.2 The distribution $\mathbf{f} = (f_1, f_2, \ldots, f_n)$ of the anchor is called *D-spectrum* or *signature*, (where "D" stands for *destruction* process of anchor discovery).

Remark 1.1 Historically, the signature was first introduced by Samaniego [11] in a form equivalent to Definition 1.2. Independently, it was described 6 years later in [3] under the term *Internal Distribution*. The authors of [4–6] used the term D-spectra.

Definition 1.3 Denote by Y the discrete random variable with density $\mathbf{f}$. Its *cumulative* distribution function

$$F_0(k) = \sum_{i=1}^{k} f_i$$

is called *cumulative D-spectrum* or cumulative signature.

For networks having more than $n = 7$–8 components the calculation of D-spectra is made by means of an efficient Monte Carlo algorithm, see for example [5, 6]. This algorithm generates a sample of M permutations and estimates the frequency $\widehat{f(k)}$ of anchor appearance on the k-th position.

Denote by $C(k),\ k = 1, \ldots, n$, the number of failure sets which have k components *down* and $(n-k)$ remaining components *up*. $C(k)$ is a *combinatorial invariant* of the system. Knowing $C(k)$ and the *up/down* probabilities p and $q = 1-p$ of network components, we are able to compute system *DOWN* probability as

$$P(DOWN) = \sum_{k=1}^{n} C(k) q^k p^{(n-k)}. \tag{1}$$

Let $H(t)$ be CDF of component lifetime τ: $P(\tau \leq t) = H(t)$. Denote by p the probability that the component is *up* at time t_0. Then

$$p = 1 - H(t_0), q = H(t_0).$$

Therefore, (1) gives the probability that the network is *DOWN* at time t_0. Thus, the probability that system lifetime τ_{sys} does not exceed t_0 is

$$P(\tau_{sys} \leq t_0) = \sum_{k=1}^{n} C(k)[H(t_0)]^k [1 - H(t_0)]^{(n-k)}. \tag{2}$$

The crucial fact in obtaining Eqs. (1) or (2) is the following formula connecting $C(k)$ and $F_0(k)$:

$$C(k) = F_0(k) \frac{n!}{k!(n-k)!}.$$

It can be proved analytically using the formulas of order statistics for random variables with CDF $H(t)$, see [4], or using combinatorial arguments, see for example [5].

2.3 Two Types of Components

Now we turn to the network which has components of two types, namely there are n_1 components of type 1 and n_2 components of type 2. For sake of brevity, we call them x-type and y-type components, respectively, $n_1 + n_2 = n$. These x and y-type components have i.i.d. lifetimes, with CDFs $H_1(t)$ and $H_2(t)$, respectively.

The key to the principal formula (1) is the knowledge of $C(k)$, the number of failure sets with k components *down*. Now, when we have two types of components, we need to know the values of $C(k, r)$, the numbers of failure sets which have k *down* components of x-type and r *down* components of y-type, (the remaining $(n_1 - k)$ and $(n_2 - r)$ components are *up*). Then, the *DOWN* probability for network with two types of components equals

$$P^{\star}(DOWN) = \sum_{0 \leq k \leq n_1} \sum_{0 \leq r \leq n_2} C(k, r) q_1^k p_1^{(n_1-k)} q_2^r p_2^{(n_2-r)},$$

where q_1, q_2 and $p_1 = 1 - q_1, p_2 = 1 - q_2$ are *down* and *up* probabilities for x-type and y-type components, respectively.

Similar to the one-type component systems, $C(k, r)$ are *invariants* depending on system structure function and not depending on component lifetime distributions [5, 14].

2.4 Counting (k, r)-failure Vectors

In case of two types of components, we have to modify the notation for system state vector $x^\star$. Now it will be an ordered sequence of n_1 pairs (x_i, I) for components of x-type and n_2 pairs (y_j, I) for y-type components, where $x_1, \dots, x_{n_1}$ are the names (numbers) of x-components and $y_1, \dots, y_{n_2}$ are the names of y-components. Indicator I will be 1 or 0, if the corresponding component is *up* or *down*, respectively.

Example 1.1 Consider the network shown on Fig. 1. Components subject to failure are the edges, nodes are reliable. The network fails if there no connection between terminals S and T. The network has $n_1 = 3$ components of x-type $x_1 = (b, c)$, $x_2 = (b, T)$, $x_3 = (c, T)$, and two components of y-type—$y_1 = (S, a), y_2 = (a, b)$. Consider, for example, a vector $\mathbf{x}^* = (x_1, 1), (x_2, 0), (x_3, 1), (y_1, 0), (y_2, 0)$. Obviously $\phi(\mathbf{x}^*) = 0$. This failure vector contains one x-edge down and two y-edges down.

Sequential destruction of a random permutation. Consider a random permutation $\pi^\star$ of n_1 x-type pairs mixed randomly with n_2 pairs of y-type. Set $I = 1$ in all pairs, i.e. initially set all components in *up*. Start turning *down* component after component by moving along the permutation from left to right. Check system state on each step and locate the first component (the anchor) when the system goes *DOWN*. Let the first observed failure set has (u, v) components of type 1 and type 2, respectively. Continue turning *down* sequentially all remaining $(n_1 + n_2) - (u + v)$ components in the permutation. Note that on each step appears a new failure set.

Definition 1.4 Random permutation is called of (u, v)-*anchor type* if its anchor produces failure set of type (u, v).

Definition 1.5 Random permutation is called a (k, r)-*generator* if among the failure sets revealed during the destruction process *after* the anchor has been revealed, there is a (k, r)-failure set.

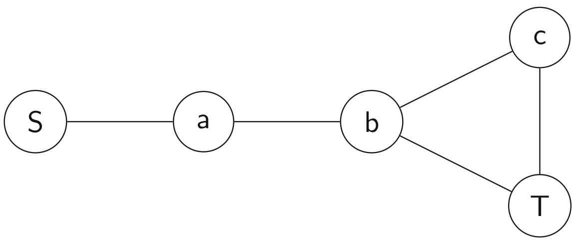

Fig. 1 Network with 5 components. It is *UP* if there is an $S - T$ connection. Edges $(S, a), (a, b)$ are of y-type, the remaining edges are of x-type

Example 1.1 **continued**. Suppose we have the following random permutation before the destruction process starts: $\pi^\star = [(x_1, 1), (x_3, 1)(y_1, 1), (x_2, 1), (y_2, 1)]$. Below are 5 stages of the sequential destruction:

$$\begin{aligned}
\mathbf{1}:&\ [(x_1, 0), (x_3, 1), (y_1, 1), (x_2, 1), (y_2, 1)].\\
\mathbf{2}:&\ [(x_1, 0), (x_3, 0), (y_1, 1), (x_2, 1), (y_2, 1)].\\
\mathbf{3}:&\ [(x_1, 0), (x_3, 0), (\mathbf{y}_1, 0), (x_2, 1), (y_2, 1)].\\
\mathbf{4}:&\ [(x_1, 0), (x_3, 0), (y_1, 0), (x_2, 0), (y_2, 1)].\\
\mathbf{5}:&\ [(x_1, 0), (x_3, 0), (y_1, 0), (x_2, 0), (y_2, 0)].
\end{aligned}$$

The anchor is observed on the third step and therefore $\pi^\star$ is of (2,1) anchor-type. Analysing steps 4 and 5, it is seen that $\pi^\star$ is also a (3,1) and (3,2) generator.

Definition 1.6 Denote by $F(k, r)$ the probability that a random permutation is of (k, r) anchor-type **or** is a (k, r)-type generator. Obviously,

$$F(k, r) = \frac{N(k, r)}{(n_1 + n_2)!}, \tag{3}$$

where $N(k, r)$ is the number of permutations which are of (k, r)-anchor type *or* (k, r)-generators. We call the matrix $\|F(k, r)\|_{(n_1+1)\times(n_2+1)}$ the *two-dimensional or 2D-spectrum.*

Definition 1.7 Let $g(k, r)$ be the probability that a random permutation is of (k, r)-anchor type. Obviously,

$$g(k, r) = \frac{A(k, r)}{(n_1 + n_2)!},$$

where $A(k, r)$ is the number of permutations which are of (k, r)-anchor type.

Example 1.1 **continued** Let us determine $N(2, 1)$. All permutations of three x-es and two y-s of type $(x_i, x_j, y_l, x_s, y_z)$ with one y_l on third position and two x-es among the first three positions, produce failure sets of type (2,1). By permuting the first three elements and the remaining two elements, and also by replacing y_1 by y_2 among first three elements, we will have 24 permutations for a fixed pair of x_i, x_j. Since we can choose this pair in three ways, there is a total of $N(2, 1) = 72$ permutations. Among them, there are 8 anchor-type (2,1)-permutations. These permutations must have y_j on the third position, and two x-es on the first two positions, like $\pi = (x_1, x_3, y_1, x_2, y_2)$. There are two ways to exchange the positions of x_1 and x_3, two ways to exchange y_1 by y_2 on the third position, and two ways to exchange components on the fourth and fifth positions. Therefore, for our network, $F(2, 1) = 72/5! = 0.6$ and $a(2, 1) = 8/120 = 0.0666$.

In Table 1 we present the $\|F(k, r)\|$ and $\|g(k, r)\|$ matrices for system shown on Fig. 1.

Table 1 $\|F(k,r)\|$ and $\|g(k,r)\|$ matrices

r	$k=0$	$k=1$	$k=2$	$k=3$	$r=0$	$k=0$	$k=1$	$k=2$	$k=3$
0	0.0	0.0	0.2	0.1	0	0.0	0.0	0.2	0.0333
1	0.4	0.6	0.6	0.4	1	0.4	0.3	0.0666	0.0
2	0.1	0.3	0.6	1.0	2	0	0	0	0

2.5 Counting the Number $C(k,r)$ of (k,r)-failure Sets

Here the main role is played by the following Theorem.

Theorem 1.1

$$C(k,r) = F(k,r)\frac{(n_1+n_2)!}{(k+r)!(n_1+n_2-k-r)!}. \tag{4}$$

Proof From the description of the sequential destruction of random permutation, follows that a (k,r)-failure set is a "compact" block of $(k+r)$ components located at the first $(k+r)$ positions of the permutation (the anchor-type or generated failure set). It is also obvious that one permutation can produce not more than a single (k,r) failure set. Permutations between the members of one such set produce $(k+r)!$ copies of it, and each copy is a failure set. In addition, there are $(n_1+n_2-k-r)!$ permutations of the remaining components. Therefore $N(k,r)$ permutations produce

$$\frac{N(k,r)}{(n+k)!(n_1+n_2-k-r)!}$$

original (k,r) failure sets. Remembering (3), we arrive at the desired formula (4). □

The following Corollary establishes the connection between the cumulative one-dimensional D-spectrum $F_0(k)$ (see Definition 1.3) and the 2D-spectrum.

Corollary 1.1

$$F_0(w) = \sum_{k=0}^{\min(n_1,w)} F(k, w-k).$$

Proof Suppose that we declare n_2 components of y-type to be identical to the components of x-type. Then each (k,r)-failure set becomes a $(k+r)$-failure set in the system having n_1+n_2 identical components. Therefore,

$$\sum_{k=0}^{\min(n_1,w)} C(k, w-k) = C(w),$$

or

$$\sum_{k=0}^{w} F(k, w-k)\frac{(n_1+n_2)!}{w!(n_1+n_2-w)!} = F_0(w)\frac{(n_1+n_2)!}{w!(n_1+n_2-w)!},$$

which proves the Corollary. □

Example 1.1 **continued**. Let us verify $C(2, 1)$. By (4), $C(2, 1) = 0.6 \cdot 5!/(3!2!) = 6$. Indeed, there are 6 failure sets having two x-type and one y-type component: $[x_1, x_2, y_1,], [x_1, x_3, y_1], [x_2, x_3, y_1], [x_1, x_2, y_2,], [x_1, x_3, y_2], [x_2, x_3, y_2]$.

2.6 Simulation Algorithm for Estimating $F(k, r)$

Algorithm 1 2D-Spectra

Input: n_1 and n_2—the number of x-type and y-type components, respectively. N-number of replications.

Output: $\hat{G}$ and $\hat{F}$ -the estimators of $\|g(k, r)\|$ and $\|F(k, r)\|$, respectively.

1: Set $t = 1$ and let $M_1[i, j]$ and $M_2[i, j]$ be two matrices with $n_1 + 1$ rows and $n_2 + 1$ columns. Put all elements of these matrices to be zero.
2: Generate $\prod_t = (\prod_1^{(t)}, \ldots, \prod_{n_1+n_2}^{(t)})$ - a random component permutation.
3: Find the anchor J_t of $\prod_t$.
4: Set K_t and R_t be the number of x-type and y-type components in the first J_t elements of $\prod_t$. Set $M_1[i = K_t + 1, j = R_t + 1] = M_1[i = K_t + 1, j = R_t + 1] + 1$ and $M_2[i = K_t + 1, j = R_t + 1] = M_2[i = K_t + 1, j = R_t + 1] + 1$.
5: Set: $T_1 = K_t$ and $T_2 = R_t$.
6: **for** $i = J_t + 1$ `to` $n_1 + n_2$ **do**
7: **if** $\prod_{i+1}^{(t)}$ is x-type component **then** set $T_1 := T_1 + 1$,
8: **else** $T_2 := T_2 + 1$.
9: **end if**
10: $M_2[T_1 + 1, T_2 + 1] = M_2[T_1 + 1, T_2 + 1] + 1$.
11: **end for**
12: If $t < N$ set $t = t + 1$ and go to **Step 2**.
13: **return**: $\hat{G} = \|M_1\|/N$, $\hat{F} = \|M_2\|/N$.

Exact calculation of $F(k, r)$, like it was done in Example 1.1, becomes impractical already for n exceeding 6–8. We suggest using a Monte Carlo simulation algorithm for estimation of the $F(k, r)$ probabilities. This algorithm is based on simulating a relatively large (say 1,000,000) random permutations and extracting from them information about the number of failure sets. The algorithm below allows rather efficient and accurate estimation for networks with 50–70 components. Note that each random permutation of size n which has a (k, r)-anchor, produces also $n - (k + r)$ generated failure sets.

This algorithm has been applied to a network with 34 nodes and 68 edges, see Sect. 3. Quite accurate estimates of the G and F matrices were obtained by using $N = 10^6$ replications. The CPU time did not exceed 16 s.

3 Reliability of a Transportation Network

3.1 Description of the Network. Reliable Nodes, Unreliable Edges

The network is shown on Fig. 2. This is a hypothetical geographically oriented road network. It is designed as Barabasi-Albert system [1] with 34 nodes and 68 edges. Centrally located node 31 represents the capital city.

Important strategic objects (e.g. hospitals, supply centers, etc.) are located in terminal nodes 2, 5, 9, 33, 34. Thirteen edges are more reliable roads.

$$(14,33),(31,33),(33,23),(22,23),(5,22),(5,20),(29,34),$$
$$(34,14),(15,14),(34,31),(5,31),(20,31),(20,29).$$

They form a ring around the capital and also contain several radial roads. These edges in our notation are the "strong" x-type edges. The remaining $68 - 13 = 55$ edges are the y-type edges. We remind that network failure means the loss of terminal connectivity: the network is *DOWN* if at least one of the terminals gets separated from other terminals. Edges can fail as a result of an enemy "attack", natural disaster or heavy road accidents, see [5, 7, 10, 15].

Table 2 presents $P(DOWN)$ calculated by (3) and Algorithm "2D-spectra" for $F(k,r)$ estimation, on the basis of generating $N = 10^6$ random permutations. The results were checked by crude Monte Carlo simulation, based also on 10^6 replications, see P_{cmc}. As it is seen from the table, the relative error is quite small which means that the estimation by our algorithm is very accurate. We see from the table that in order to provide $P(DOWN) \le 0.05$ it is necessary to have $p_1 \ge 0.7$ for type y and about 0.8–0.9 for strong edges. Very interesting is the fact that increasing strong edge reliability from 0.9 to 0.99 has relatively little effect on $P(DOWN)$.

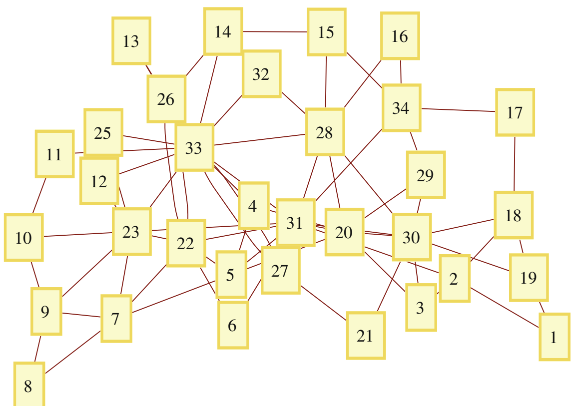

Fig. 2 Transport network with 34 nodes and 68 edges

Table 2 $P(DOWN)$ for edge failure: estimated and simulated values, $N = 1{,}000{,}000$

p_2	p_1	$P(DOWN)$	P_{cmc}	Rel.err. %
0.5	0.6	0.39536	0.39526	0.10
0.5	0.7	0.34776	0.34749	0.13
0.5	0.8	0.32061	0.31854	0.14
0.5	0.9	0.30799	0.30835	0.14
0.5	0.99	0.30615	0.31641	0.15
0.6	0.7	0.16621	0.16551	0.20
0.6	0.8	0.14808	0.14592	0.25
0.6	0.9	0.13862	0.13725	0.25
0.6	0.99	0.13754	0.13415	0.25
0.7	0.8	0.04932	0.04906	0.45
0.7	0.9	0.04551	0.04450	0.45
0.7	0.99	0.04442	0.04400	0.46
0.8	0.9	0.00877	0.00862	1.10
0.8	0.99	0.00844	0.00834	1.10
0.9	0.99	0.00053	0.00049	0.40

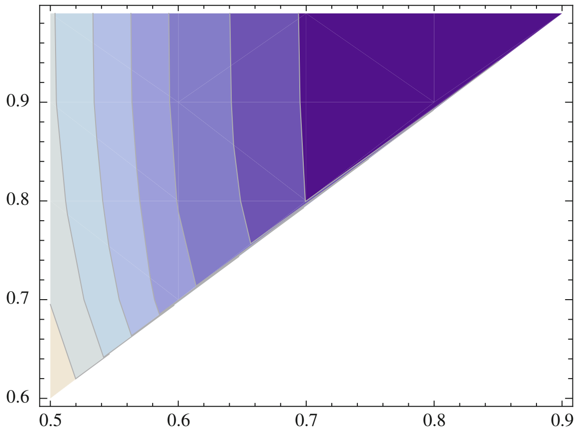

Fig. 3 Contour plot for data of Table 2 (edge failures)

Interesting information is provided by the contour plot on Fig. 3. Area with $P(DOWN) < 0.05$ is shown by deep blue color. The adjacent blue area corresponds to *DOWN* probabilities in the interval [0.05–0.1].

We also investigated the situation with edges are deteriorating in time. It as assumed that strong edge reliability $p_2(t)$ depends on time as $p_1(t) = e^{-t}$, and the remaining edges have $p_2(t) = e^{-2t}$. The numerical results are presented in Table 3.

Table 3 $P(DOWN)$ as a function of time (edge failures)

t	$p_2 = e^{-2t}$	$p_1 = e^{-t}$	$P(DOWN)$
0.1	0.819	0.905	0.0058
0.2	0.801	0.895	0.0086
0.3	0.779	0.882	0.0132
0.4	0.751	0.867	0.0214
0.5	0.716	0.846	0.0371
0.6	0.670	0.819	0.0692
0.7	0.606	0.779	0.1410
0.8	0.513	0.717	0.3142
0.9	0.368	0.607	0.7010
1.0	0.135	0.368	0.9990

3.2 Unreliable Nodes

We also have studied the network reliability when the nodes are subject to failure. Six nodes 20, 22, 23, 28, 30 and 31 are declared to be the x-type1.4 presents the results of the numerical investigation of network reliability. Again it is seen that our algorithm provides quite accurate results with a small relative error. Figure 4 (right) shows the area of parameters (p_1, p_2) where the *DOWN* probability is smaller than 0.05 (shown by deep blue).

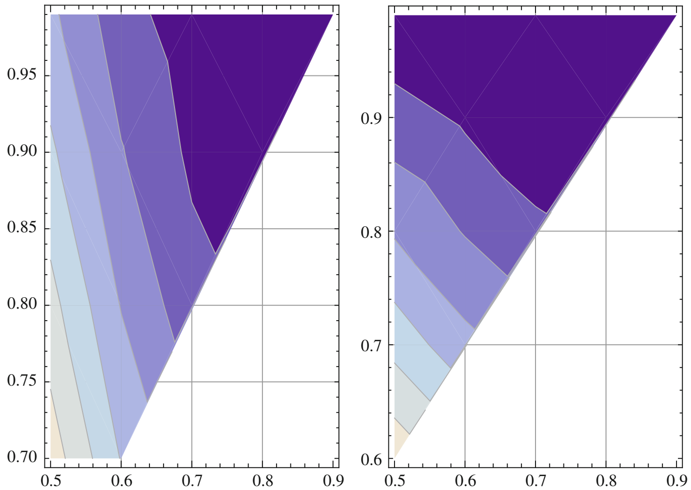

Fig. 4 Contour plots for data of Table 4. Nodes relocated (*left*), original (*right*)

Table 4 $P(DOWN)$ for node failure: estimated and simulated values, N = 1,000,000 runs

p_2	p_1	$P(DOWN)$	P_{cmc}	Rel. err. %	$P(DOWN)^{\star}$
0.5	0.6	0.33666	0.33633	0.14	–
0.5	0.7	0.23340	0.23360	0.18	–
0.5	0.8	0.14375	0.14359	0.24	0.26697
0.5	0.9	0.06719	0.06660	0.30	0.20931
0.5	0.99	0.00647	0.00634	0.12	0.16008
0.6	0.7	0.16476	0.16541	0.22	0.19729
0.6	0.8	0.09654	0.09634	0.30	0.14738
0.6	0.9	0.04240	0.04259	0.47	0.10304
0.6	0.99	0.00394	0.00393	1.60	0.06956
0.7	0.8	0.05727	0.05686	0.40	0.06905
0.7	0.9	0.02374	0.02373	0.60	0.04076
0.7	0.99	0.00220	0.00211	2.12	0.02234
0.8	0.9	0.01030	0.01046	0.95	0.01214
0.8	0.99	0.00083	0.00089	3.40	0.00435
0.9	0.99	0.00024	0.00021	6.40	0.00029

In order to see how influential is the location of the strong nodes, we relocated these nodes to periphery. Now nodes $14, 15, 18, 11, 10, 7$ are declared to be strong nodes of x-type. As it could be expected, the network with relocated strong nodes is less reliable, as it is seen from last column $P(DOWN)^{\star}$ of Table 4, and the contour surface plot on Fig. 4, on the left. Deep blue area shows low $P(DOWN)$ values, and is, therefore, the area of high reliability. It is considerably larger for the original location of the strong nodes (the plot on the right).

4 $||g(k,r)||$ Matrix and "Shock Process" Trajectories

Suppose that the network is subject to a two-dimensional "shock process" which is a random sequence of type "x"-shocks which hit randomly the strong components (strong nodes or strong edges), permuted randomly with type "y"-shocks which hit the weak components.

This process stops when the network fails. As it follows from the definition of the permutation destruction process, the networks fails at the "stopping point" determined by the permutation anchor. The distribution of the location $(x = V, y = U)$ of the "stopping point" is shown on Fig. 5 by means of surface contour plot. In this example the "shocks" kill strong and weak nodes.

The exact probabilistic meaning of this plot is the following. The elements of matrix $G = ||g(k, r)||$ present the conditional probabilities that the shock process stops at coordinate $(V = k, U = r)$, given that network is $DOWN$:

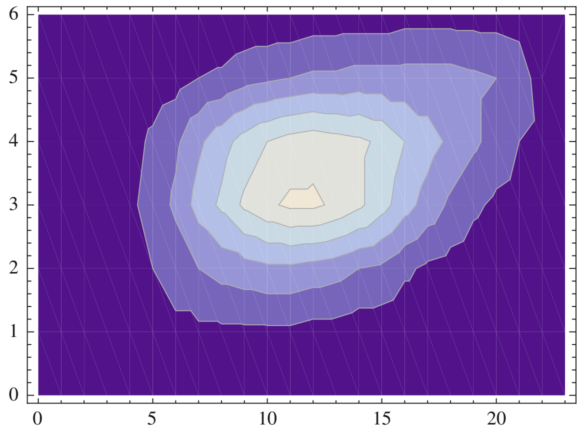

Fig. 5 Contour plot for $\|g(k, r)\|$ matrix

$$g(k, r) = P((V = k, U = r)|DOWN),$$

where V, U are the numbers of strong and weak destroyed components at the stopping point, respectively.

Let us examine the plot on Fig. 5. The horizontal axis is for weak nodes, vertical axis—for strong. By deep blue is shown the area where the trajectory does not stop. Here the trajectory does not stop at all. The adjacent area (light blue) shows points having stopping probability between 0.005 and 0.01. Next area closer to the center shows the points having probabilities between 0.01 and 0.015, and so on. So,the point $g(V = 2, U = 7)$ lies in the probability interval $[0.010, 0.015]$.

The $\|g(k, r)\|$ matrix is a valuable structural characteristic of the network. Let us demonstrate its use by investigating so-called "quantile areas".

Contrary to the definition of a quantile for one-dimensional case, for more dimensions there are many ways to determine the area which has probabilistic mass q, see e.g. [16]. Let us consider here the triangular areas of type $U + V \le D$. Omitting the routine calculations, we present the following results for $D = 4, 5, 6, 7, 8, 9$:

$$P(U + V \le 4) = 0.0068, P(U + V \le 5) = 0.015, P(U + V \le 6) = 0.030,$$

$$P(U + V \le 7) = 0.052, P(U + V \le 8) = 0.084, P(U + V \le 9) = 0.128.$$

So, for example, the network fails with probability 0.128 if the total number of failed nodes is not more than 9.

Comparing the size of equal quantile areas may serve as an instrument to compare the reliability of alternative structures. For example, structure A is more reliable than structure B if the two dimensional 0.1-quantile area $D_A(q = 0.1)$ for A is larger than the similar area $D_B(q = 0.1)$ for structure B.

5 More Than Two Types of Components—Concluding Remarks

Suppose that the network has $K > 2$ different groups of i.i.d. components. Then the expression for network *DOWN* probability will be a natural extension of (3) to more variables. Denote by n_i the number of i-th type components, $n_1 + n_2 + \cdots + n_K = n$, and let $C(x_1, \ldots, x_K)$ be the number of failure sets having x_i components of i-th type *down*, $i = 1, .., K$. Then

$$P(DOWN) = \sum_{0 \le x_i \le n_i, i=1,\ldots,K} C(x_1, x_2, \ldots, x_K) \prod_{i=1}^{K} q_i^{x_i} \prod_{i=1}^{K} p_i^{n_i - x_i}.$$

The main problem remains estimation of $C(x_1, \ldots, x_K)$, the numbers of failure sets. This can be done in the framework of the above described Algorithm, with obvious modifications. Now the random permutation will have K types of symbols for denoting components of K groups, and now the failure sets of anchor-type and of generated type will have x_i components of i-th type, $i = 1, \ldots, K$ For $K = 3$, for example, the F-matrix will become a three-dimensional cubic matrix.

There are several important issues left outside the scope of the present paper. Let us mention on the first place the investigation of component importance, see e.g. [6]. Similar to the networks with one type of components, for several types of components, importance issues are the key to optimal network design and to the "nomination" of the components to be the "strong" ones.

Very interesting would be also to compare several competing network structures by analyzing their q-quantile "areas", as it was briefly discussed in Sect. 4. We leave these issues for the future research.

Acknowledgements The work of Radislav Vaisman was supported by the Australian Research Council Centre of Excellence for Mathematical & Statistical Frontiers, under grant number CE140100049.

References

1. Barabasi A, Albert R (1999) Emergence of scaling in random networks. Science 286:509–551
2. Daqing L, Qiong Z, Zio E, Havlin S, Kang R (2015) Network reliability analysis based on percolation theory. Eng Reliab Syst Saf 142:1–15
3. ElperinT Gertsbakh I, Lomonosov M (1991) Estimation of network reliability using graph evolution models. IEEE Trans Reliab 40:572–581
4. Gertsbakh I, Shpungin Y (2010) Models of network reliability: analysis, combinatorics and Monte Carlo. CRC Press, Boca Raton
5. Gertsbakh I, Shpungin Y (2011) Network reliability and resilience. Springer briefs in electrical and computer engineering. Springer, Berlin

6. Gertsbakh I, Shpungin Y (2012) Using D-spectra in Monte Carlo estimation of system reliability and component importance. In: FrenkelI LA (ed) Recent advances in system reliability. Springer, London, pp 49–61
7. Hudges J, Healy K (2014) Measuring the resilience of transport infrastructure. New Zealand Transport Agency, Report No. 546
8. Nagurney A, Qlang Q (2007) Robustness of transportation networks subject to degradable links. Lett J Explor Front Phys
9. Newman M (2010) Networks. Oxford University Press
10. Peeta S, Salman S, Gunnec D, Kannan V (2010) Predisaster investment decisions for strengthening a highway network. Comput Oper Res 37:1708–1719
11. Schneider C, Yazdani N, Araujo N, Havlin S, Herrmann H (2013) Toward designing robust coupled networks. Scientific reports
12. Samaniego F (1985) On closure of the IFR under formation of coherent systems. IEEE Trans Reliab 34:69–72
13. Samaniego F (2007) System signatures and their applications in engineering reliability. Springer
14. Samaniego F, Navarro J, Balakrishnan N (2011) Signature-based representation of the reliability of systems with heterogeneous components. J Appl Probab 48:856–867
15. Sanchez-Silva M, Daniels M, Lieras G, Patino D (2005) A transport network reliability model for the efficient assignment of resources. Transp Res Part B: Methodol 39(1):47
16. Sering R (2005) Quantile functions for multivariate approaches and applications. Stat Neerl 56(part B):214–232
17. Sterbenz J, Hutchinson D, Çetinkayaa E, Jabbara A, Rohrera J, Schöllerc M, Smith P (2010) Resilience and survivability in communication networks: strategies, principles and survey of discipline. Comput Netw 54(8):1245–1265