Obtaining and Using Cumulative Bounds of Network Reliability

Alexey S. Rodionov and Denis A. Migov

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.72182

Abstract

In this chapter, we study the task of obtaining and using the exact cumulative bounds of various network reliability indices. A network is modeled by a non-directed random graph with reliable nodes and unreliable edges that fail independently. The approach based on cumulative updating of the network reliability bounds was introduced by Won and Karray in 2010. Using this method, we can find out whether the network is reliable enough with respect to a given threshold. The cumulative updating continues until either the lower reliability bound becomes greater than the threshold or the threshold becomes greater than the upper reliability bound. In the first case, we decide that a network is reliable enough; in the second case, we decide that a network is unreliable. We show how to speed up cumulative bounds obtaining by using partial sums and how to update bounds when applying different methods of reduction and decomposition. Various reliability indices are considered: k-terminal probabilistic connectivity, diameter constrained reliability, average pairwise connectivity, and the expected size of a subnetwork that contains a special node. Expected values can be used for unambiguous decision-making about network reliability, development of evolutionary algorithms for network topology optimization, and obtaining approximate reliability values.

Keywords: network reliability, factoring method, network connectivity, random graph, diameter constraint, probabilistic connectivity, pairwise connectivity, network topology optimization, estimation, cumulative updating

1. Introduction

The network reliability analysis is one of the primary tasks in the course of the network topology design and optimization. Usually, random graphs are used when modeling networks with unreliable elements [1–7]; therefore, the network reliability is defined as a connectivity measure in a random graph. There are many reliability indices of networks, for example, probabilistic connectivity [4], average pairwise reliability [5, 6], and diameter constrained reliability [7].

The most widespread network reliability index is the probabilistic connectivity of the corresponding random graph. Depending on the number of terminals (selected nodes that must be connected), there are three types of this measure such as two-terminal, all-terminal (ATR) and k-terminal reliabilities. The average pairwise reliability (APR) describes the network reliability from the point of connection between every two nodes, while a network may be still disconnected. The reliability of a network with a diameter constraint (DCR) is defined as the probability of each pair of nodes connectivity by paths that travel through a limited number of communication channels. This index is more useful, especially for P2P networks [8], wavelength division multiplexing networks, wireless sensor networks, and so on.

Another reliability index we consider in this chapter is the mathematical expectation of the size of a connected subgraph that contains a special node (MENC) [2], which describes the quality of monitoring area coverage.

Despite the fact that problems of exact reliability computations are NP-hard [9, 10], it is possible to conduct the exact calculation for networks of practical interest dimension, taking into account some special features of network topologies. For large-scale networks, various methods of reliability evaluation are widely used [11, 12]. Won and Karray suggested the cumulative updating of reliability bounds for uniqueness of decision on network feasibility [13]. Originally, this approach was presented for ATR. In this chapter, we present our results on cumulative updating of other reliability indices, and some improvements for ATR bounds cumulative updating. In addition, the main ideas of constructing the cumulative bounds for random graph characteristics are discussed along with examples of their usage in obtaining better evaluation of reliability indices and improving bionic optimization algorithms.

2. The basic notations

We simulate a network with perfectly reliable nodes and unreliable links by an undirected random graph $G = (V,E)$ with given presence probabilities $0 \le e_i \le 1$ of the edges. The notations which are used in this chapter are given below. Most of them coincide with the notations from [14].

G—Undirected probabilistic network;

V—Set of n nodes;

E—Set of m edges;

e_i; e_{ij} – i-th edge or edge that connects i-th and j-th nodes, depending on context;

p_j – Operating probability of j-th edge;

w_i – Weight of node i, $WT = (w_1,\ldots,w_n)$;

$W(G)$ – Total weight of nodes of G;

$R(G)$ – All-terminal reliability of G, that is, probability that every two nodes are connected;

K—Set of terminal nodes;

$R_K^d(G)$—Diameter constrained reliability of G, that is, the probability that every two nodes from K are connected by a path that travel through not more than d edges;

LB, UB—Lower and upper bounds of G reliability. In case of ATR, we use the original notation form [13]: *RL, RU;*

R_0—Predefined threshold for the network reliability value;

$N(G)$ $(M(G))$—Mathematical expectation of the number of disconnected (or connected) pairs of nodes in G;

CS(G; s)—Mathematical expectation of a number of nodes in a connected subgraph that contains a node s; if $s = 1$, then CS(G);

$\overline{R}$—Average pairwise reliability of G;

C—Edge chain composed of edges $e1,\dots,ek$;

G/C (G/e)—Network G with a contracted chain C (edge e);

$G\backslash C$ $(G\backslash e)$—Network G without chain C (edge e); and

G_e^*—Network G with an absolutely reliable edge e;

3. Factoring method and cumulative updating of network reliability bounds

The most common exact method for calculating various network reliability measures is the factoring method. According to this method, we divide the probability space, which consists of all particular network realizations, into two sets based on the presence or absence of a network element. Further on we refer to such a network element as a factored element, or a pivot element. As a result, for a given network G and a factored element e, we will obtain the two networks: G_e^*, where e is absolutely reliable, and $G\backslash e$, where e is absolutely unreliable, so we could remove it. The probability of G_e^* is equal to the pivot element reliability, and the probability of $G\backslash e$ is equal to the pivot element failure probability. For a reliability index *Rel* of the initial network, the following expression holds (the total probability law):

$$Rel(G) = p_e Rel\left(G_e^*\right) + \left(1 - p_e\right) Rel(G\backslash e) \tag{1}$$

Then the obtained networks are subject to the same factoring procedure. Recursions continue until either an unreliable network is obtained (0 is returned) or an absolutely reliable network is obtained (1 is returned). For ATR (**Figure 1**), expression (1) turns to:

$$R(G) = p_e R(G/e) + \left(1 - p_e\right) R(G\backslash e). \tag{2}$$

We can speed up the factoring process by calculating the intermediate networks reliabilities directly, that is, without further factorization. For ATR calculation, the five-vertex graph reliability formula can be used [13]. Another way to accelerate the reliability computing by using

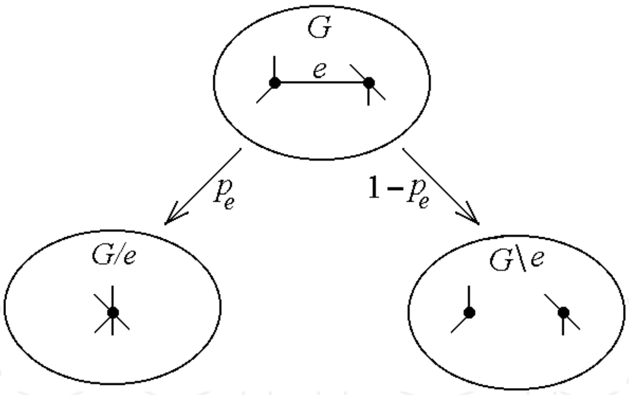

Figure 1. The factoring method.

various reduction [4] and decomposition [17] methods, such as serial-parallel transformation on each recursive call of the factoring procedure, biconnected decomposition, and other methods.

The idea of the cumulative updating method is an incremental updating of exact lower (*LB*) and upper (*UB*) reliability bounds and comparing them with a given reliability threshold R_0. If $LB > R_0$, then the network is reliable. If $UB < R_0$, then the network is unreliable. The *LB* must necessarily be non-decreasing, and the *UB* must necessarily be non-increasing. Another obligatory condition is the equality of *LB*, *UB* at the last step and exact reliability value *R*. One possible way of bounds updating is the usage of the factoring method. When a final graph or a disconnected graph is obtained, we can update bounds [13].

As a rule, any algorithm of calculation of mathematical expectation for a non-negative function μ of a random graph *G* practically comes to summarizing non-negative values. According to the definition of mathematical expectation in the discrete case [14]:

$$E[\mu(G)] = \sum_{H \in \Gamma} P(H) \cdot \mu(H), \tag{3}$$

where Γ is a set of all possible final realizations of *G* that are obtained during factorization.

If some realizations Γ_0 are obtained along with their probabilities, and a function μ is obtained for all these realizations, then the *LB* is greater or equal to the corresponding partial sum.

Now let us assume that for μ, its possible minimal (μ_m) and maximal (μ_M) values are known. In this case, we easily obtain the following bounds:

$$LB = \sum_{H \in \Gamma_0} P(H) \cdot \mu(H) + \mu_m \left(1 - \sum_{H \in \Gamma_0} P(H)\right); \quad UB = \sum_{H \in \Gamma_0} P(H) \cdot \mu(H) + \mu_M \left(1 - \sum_{H \in \Gamma_0} P(H)\right). \tag{4}$$

Finally, both bounds will obviously come to an exact value.

From (4), we easily obtain the equations for improving bounds when the new (*i*-th) realization H_i is obtained and $\mu(H_i)$ is calculated:

$$LB_i = LB_{i-1} + P(H_i) \cdot \left[\mu(H_i) - \mu_m\right]; \quad UB_i = UB_{i-1} - P(H_i) \cdot \left[\mu_M - \mu(H)\right]. \tag{5}$$

For ATR, $\mu_m = 0$ and $\mu_M = 1$, so we arrive to the equations presented by Won and Karray in [13], while for MENC $\mu_m = 1$ and $\mu_M = n$, where n is a total number of nodes in a graph. Initial values of bounds are equal to these values.

4. Improvements for ATR bounds cumulative updating

4.1. The chain branching

Our methods named chain branching (CB) and chain reduction (CR) [15] could be used [16] .as a basis for two kinds of the ATR bound updating (UAB) algorithms named UAB based on CB (UAB_CB), and UAB based on CR (UAB_CR). These methods can be used if a network contains a chain, that is, a sequence of adjacent two-degree nodes. The CB reduces the calculation of ATR for calculating the ATRs of networks obtained from G by removing a pivot chain, or removing it and contracting its terminal nodes, while CR reduces it for calculating the ATR of the network G_R, in which, this chain is substituted by a single edge. Let a chain C consist of k edges, its terminal nodes are s and t, and let an edge e_{st} has a reliability p_{st}, then [16]

$$R(G) = \left[\prod_{i=1}^{k} p_i + p_{st} \sum_{i=1}^{k} (1-p_i) \prod_{j \neq i} p_j\right] \cdot R(G/C) + (1-p_{st}) \sum_{i=1}^{k} (1-p_i) \prod_{j \neq i} p_j \cdot R(G \backslash C). \tag{6}$$

The best choice is the usage of the longest chain as a pivot. However, it requires finding of all the chains. We may accelerate the calculation by choosing a chain (or an edge) incidental to a node with a minimal degree for obtaining a new chain.

4.2. The usage of Cutnodes

Let a network G have a cutnode u that divides G into two subnetworks G_1 and G_2, therefore, $R(G) = R(G_1)R(G_2)$. $RL^{(1)}$ and $RU^{(1)}$ are lower and upper bounds for $R(G_1)$, respectively. Like wise, $RL^{(2)}$ and $RU^{(2)}$ are the lower and upper bounds for $R(G_2)$, respectively. Then $RL^{(1)} \cdot RL^{(2)} \leq R(G) \leq RU^{(1)} \cdot RU^{(2)}$.

Let us estimate the ATR of subnetworks G_1 and G_2 in turn starting from G_1. While estimating the ATR of G_1, we have no information about the ATR of G_2, thus $RL^{(2)} = 0$ and $RU^{(2)} = 1$, and RL for $R(G)$ is 0, while RU for $R(G)$ is $RU(G_1)$. After some steps of estimating $R(G_1)$, we switch to estimating $RL^{(2)}$ and $RU^{(2)}$, thus improving RL and RU for G, and so on. Therefore, we calculate the bounds for $R(G_1)$ and $R(G_2)$ separately (possibly in parallel). Now let us continue with applying this scheme to calculating the bounds for $R(G_1)$ and $R(G_2)$ by the factoring method.

Similar to [15], we suppose that some networks H_1, H_1, …, H_I are obtained from G_1, and networks $K_1, K_2, \ldots, K_J$ from G_2 by factorization (1). Let us denote $x_i = P(H_i)$, $y_i = R(H_i)$, $1 \leq i \leq I$, and $s_i = P(K_i)$, $t_i = R(K_i)$, $1 \leq i \leq J$. Then [16]:

$$R(G_1) = \sum_{i=1}^{I} x_i y_i = 1 - \sum_{i=1}^{I} x_i (1 - y_i); \quad R(G_2) = \sum_{i=1}^{J} s_i t_i = 1 - \sum_{i=1}^{J} s_i (1 - t_i). \tag{7}$$

Statement 1 *For all a,b such that 1 ≤ a ≤ I, 1 ≤ b ≤ J, the following inequalities hold:*

$$\sum_{i=1}^{a} x_i y_i \sum_{i=1}^{b} s_i t_i \leq R(G) \leq \left[1 - \sum_{i=1}^{a} x_i (1 - y_i)\right] \left[1 - \sum_{i=1}^{b} s_i (1 - t_i)\right]. \tag{8}$$

Proof of the statement can be found in [16], as well as the algorithm with the use of cutnodes (UAB_C).

4.3. The usage of two-node cuts

Now let us describe the UAB modification that uses two-node cuts. Let the network G has a node cut u,v that divides it into two subnetworks G_1 and G_2 (**Figure 2**). For any network H that contains the nodes u and v we will denote a network that is obtained by contracting these nodes as H'.

For the first time, the approach based on two-node cut was presented by Kevin Wood [17] in 1989. It describes how to perform the reliability-preserving triconnected decomposition of a network for calculating its k-terminal reliability. Later in 2006, we have introduced a general method which allows using an arbitrary node cut for calculating all-terminal reliability [18, 19]. This method makes possible to reduce the reliability calculation to the calculation of reliabilities of a group of networks each one with a smaller dimension. The total number of these networks is 2*B_W, where B_W is the Bell number and W is the number of nodes in the cut. A particular case of two-node cut along with numerical experiments to demonstrate the efficiency for the ATR calculation was presented in [16, 20], and in [21]—for the k-terminal reliability calculation. Despite the fact that the method from [17] presents a graph transformation for further factoring, and the method from [20] presents an expression for the network reliability, both of them practically lead to similar calculations of the same complexity. Another result was with the ATR calculation using special kind of node cuts [22]: the so-called

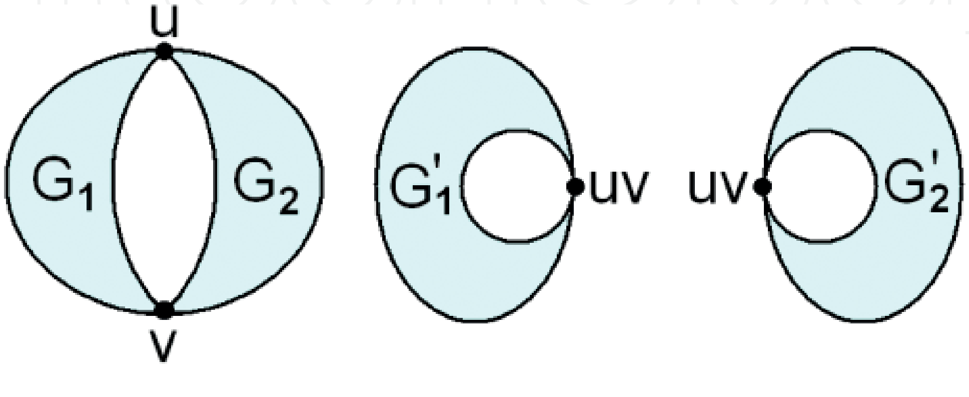

Figure 2. The network G with two-node cut and auxiliary networks.

longitudinal and cycle cuts. All the listed results, which we have obtained, are described in detail in [19]. Following is the corresponding Eq. [20] for the ATR without proof:

Theorem 1 *The following equation holds:*

$$R(G) = R(G_1)\left[R\left(G_2'(u,v)\right) - R(G_2)\right] + R(G_2)\left[R\left(G_1'(u,v)\right) - R(G_1)\right] + R(G_1)R(G_2). \tag{9}$$

The equation for three-node cut is sufficiently more complicated [18, 19]:

$$\begin{aligned} R(G) = \frac{1}{2}\Big[&R\left(G_1^{1|23}\right)\left(R\left(G_2^{12|3}\right) + R\left(G_2^{13|2}\right) - R\left(G_2^{1|23}\right)\right) \\ &+R\left(G_1^{12|3}\right)\left(R\left(G_2^{1|23}\right) + R\left(G_2^{13|2}\right) - R\left(G_2^{12|3}\right)\right) + R\left(G_1^{13|2}\right)\left(R\left(G_2^{12|3}\right) + R\left(G_2^{1|23}\right) - R\left(G_2^{13|2}\right)\right) \\ &-R(G_1)\left(R\left(G_2^{12|3}\right) + R\left(G_2^{13|2}\right) + R\left(G_2^{1|23}\right)\right) - R(G_2)\left(R\left(G_1^{12|3}\right) + R\left(G_1^{13|2}\right) + R\left(G_1^{1|23}\right)\right) \\ &+R(G_1)R(G_2)\Big] + R(G_1)R\left(G_2^{123}\right) + R\left(G_1^{123}\right)R(G_2), \end{aligned} \tag{10}$$

where nodes 1, 2, and 3 composes the threee-node cut, $G_i^{x|yz}$ is a graph that is obtained by merging y and z nodes in G_i, and G_i^{xyz} is obtained by merging x, y, and z nodes. The expression for four-node cut has also been obtained [19], but we do not present it due to its huge size.

However, our results on using an arbitrary node cut for all-terminal reliability calculation were published only in Russian [18, 19], and only the two-node cut case was presented in English [16, 20]. Maybe, this is the reason that these results are not so widely known, and therefore, the same ideas were proposed again by Juan Manuel Burgos and Franco Robledo Amoza [23, 24] in 2016. As we have done it in [16–20], in [23, 24], the authors show how to calculate the all-terminal reliability of a network with a node cut via reliabilities of a group of smaller networks, but in a somewhat different way. As a result, Eqs. (9–10) were presented.

Let us describe how we can use expression (9) and update the ATR bounds separately for both subgraphs G_1 and G_2 (**Figure 2**) for obtaining the ATR bounds of the whole graph G.

Suppose that some networks $H_1, \dots, H_I$ are obtained from G_1 during the factoring process. If this factoring does not involve edges incidental to u or v, then parallel factoring in $G_1'(u,v)$ leads to obtain $H_1'(u,v), \dots, H_I'(u,v)$. Let us introduce the following denotations: $x_i = P(H_i)$, $y_i = R(H_i)$, $x_i' = P\left(H_i'(u,v)\right)$, and $y_i' = R\left(H_i'(u,v)\right)$, for $1 \le i \le I$. As in both G_1 and $G_1'(u,v)$, the factoring is executed by the same pivot edges, we have $x_i' = x_i$. Obviously $y_i' \ge y_i$ because $H_1'(u,v)$ differs from H_i by the edge (u,v) that has reliability 1 in $H_1'(u,v)$. Similarly, by the factoring in G_2 by edges that are not incidental to x or y, we obtain the networks $K_1, \dots, K_j$, and by the factoring in $G_2'(u,v)$, we obtain $K_1', \dots, K_J'$. Let us denote $s_i P(K_i)$, $t_i = R(K_i)$, $s_i' = P\left(K_i'(u,v)\right)$, and $t_i' = R\left(K_i'(u,v)\right)$ for $1 \le i \le J$. Similar to the above case, we have $s_i' = s_i$, $t_i' \ge t_i$. Let $L = \max\{I, J\}$, and let $x_i = y_i = y_i' = 0$ for $I + 1 \le i \le L$, $s_i = t_i = t_i' = 0$ for $J + 1 \le i \le L$.

Let us obtain $RL_{1,i}$, $RU_{1,i}$, $RL_{1,i}'$, $RU_{1,i}'$, $RL_{2,i}$, $RU_{2,i}$, $RL_{2,i}'$, $RU_{2,i}'$ for $1 \le i \le L$:

$$RL_i^1 = \sum_{j=1}^{i} x_j y_{j'}, RU_i^1 = 1 - \sum_{j=1}^{i} x_j \left(1 - y_j\right), RL'^1_i = \sum_{j=1}^{i} x_j y_{j'}, RU'^1_i = 1 - \sum_{j=1}^{i} x_j \left(1 - y_{j'}\right),$$
$$RL_i^2 = \sum_{j=1}^{i} s_j t_j, RU_i^2 = 1 - \sum_{j=1}^{i} s_j (1 - t_j), RL'_i{}^2 = \sum_{j=1}^{i} s_j t_{j'}, RU'^2_i = 1 - \sum_{j=1}^{i} s_j \left(1 - t_{j'}\right).$$

As is shown in [13], these values are the upper and lower bounds for reliabilities of the corresponding networks; these bounds are exact for $i = L$. In the case $i = 0$, the lower bounds are 0 and the upper bounds are 1.

Note that $RL'_{1,i} \geq RL_{1,i}$, $RU'_{1,i} \geq RU_{1,i}$, as $y'_i \geq y_i$; $RL'_{2,i} \geq RL_{2,i}$, $RU'_{2,i} \geq RU_{2,i}$, as $t'_i \geq t_i$.

Now we obtain RL_j, and RU_j for odd and even j separately. For $j = 2i$, $0 \leq i \leq L$:

$$\begin{aligned} RU_{2i} &= RU_i^1 RU'^2_i + RU'^1_i RU_i^2 - RU_i^1 RU_i^2, \\ RL_{2i} &= RL_i^1 \left(RL'^2_i - RL_i^2\right) + RL_i^2 \left(RL'^1_i - RL_i^1\right) + RL_i^1 RL_i^2. \end{aligned} \tag{11}$$

For $j = 2i + 1, 0 \leq i \leq L - 1$:

$$\begin{aligned} RU_{2i+1} &= RU^1_{i+1} RU'^2_i + RU'^1_{i+1} RU_i^2 - RU^1_{i+1} RU_i^2, RL_{2i+1} \\ &= L^1_{i+1} \left(RL'^2_i - RL_i^2\right) + RL_i^2 \left(RL'^1_{i+1} - RL^1_{i+1}\right) + RL^1_{i+1} RL_i^2. \end{aligned} \tag{12}$$

Statement 2 *For any* $1 \leq j \leq 2\ L$, *we have*

$$RU_{j-1} \geq RU_j \geq R(G) \geq RL_j \geq RL_{j-1}. \tag{13}$$

If $j = 2\ L$, then the second and the third inequalities become equalities. The proof of the statement can be found in [16], as well as the algorithm (UAB_2NC) with two-node cuts.

Note that both CB and CR can be used for speeding up the 2NC algorithm under the condition that the nodes u and v do not belong to a pivot chain. In the case of CR $RU_j^1 {:=} RU_j^1 - x_j(1 - p)$, $RU'^1_j {:=} RU'^1_j - x_j(1 - p)$, where p is a multiplier by which the reliability of a reduced network must be multiplied in CR [4].

4.4. Case studies

We have compared the algorithms presented with node cuts UAB_C and UAB_2NC to the algorithm CR from [13] that does not take possible cutnodes or two-node cuts into account. The results [16] of the comparison are presented in **Tables 1–3**. The CR was used while factoring for better performance, and the five-node networks were considered as simple ones. The PC with Intel Core Duo 2.93 GHz was used for testing.

We have chosen the two networks G_1 and G_2 for testing UAB_C, and the two networks G_3 and G_4 for testing UAB_2NC.

Network	R_0	Algorithm	Time	Recursions
G_1	0.9	UAB_CR	39 s	5,245,491
G_1	0.9	UAB_C	<1 ms	2504
G_1	0.94	UAB_CR	34 s	639,562
G_1	0.94	UAB_C	<1 ms	2643
G_2	0.5	UAB_CR	2 s	259,879
G_2	0.5	UAB_C	60 ms	4322
G_2	0.901	UAB_CR	3 min 22 s	24,765,569
G_2	0.901	UAB_C	45 ms	3233

Table 1. Comparison of UAB_CR and UAB_C.

R_0	*UAB_CR*		*UAB_2NC*	
	Time (s)	Recursions	Time (s)	Recursions
0.5	2.2	257,084	0.08	8696
0.6	43.2	5,055,553	0.18	19,399
$R(G_3)$	—	—	0.28	27,180
0.92	13	1,532,293	0.08	8725
0.922	2.5	281,565	0.08	8696

Table 2. Results of the numerical experiments for the network G_3.

R_0	*UAB_CR*		*UAB_2NC*	
	Time (s)	Recursions	Time (s)	Recursions
0.2	0.12	16,814	0.1	10,963
0.3	27	3,630,494	0.87	80,756
$R(G_4)$	—	—	8,8	805,358
0.996	113	14,878,088	0.02	1747
0.998	0.36	45,374	< 1 ms	35

Table 3. Results of the numerical experiments for the network G_4.

The network G_1 consists of two 9-vertex complete graphs which are connected by a cutnode. Thus, it contains 17 nodes and 72 edges.

The network G_2 consists of two 5×5 lattices which are connected by a cutnode. As a cutnode, we choose the corner. The resulting network contains 49 nodes and 80 edges. **Table 1** shows the results of numerical experiments.

The network G_3 (48 nodes and 79 edges) is composed of two 5×5 lattices which are connected by two nodes (**Figure 3**).

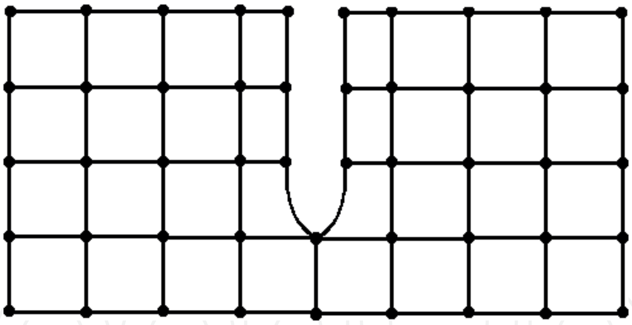

Figure 3. The union of two 5 × 5 lattices by two nodes.

The network G_4 (22 nodes and 108 edges) is composed of two 11-vertex complete graphs which are connected by two nodes without edge connecting them. A similar network that is composed of two 5-vertex complete graphs is presented in **Figure 4**.

The edges G_1 and G_4 are equally reliable with p = 0.5, while the edges G_2 and G_3 are equally reliable with p = 0.9. $R(G_1) = 0.9307194$, $R(G_2) = 0.883248$, $R(G_3) = 0.903168801959$, and $R(G_4) = 0.982472649148$.

Tables 2 and **3** present the results for G_3 and G_4, respectively. The third row contains the results when the threshold value coincides with the exact reliability value. The results for the factoring algorithm with chain reduction are shown in the second column. The results for the UAB_2NC are shown in the third column.

As can be seen, the approach proposed has a great advantage. The efficiency of UAB as it is depends on the closeness of the threshold value R_0 to the reliability value $R(G)$.

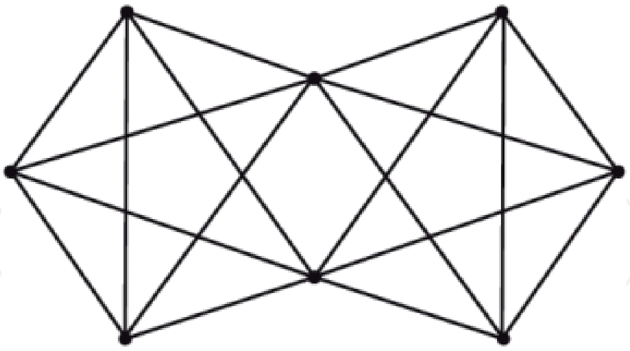

Figure 4. The union of two complete five-node networks by two nodes without connecting edge.

5. Cumulative updating of diameter constrained network reliability

For the DCR, expression (1) takes the following form:

$$R_K^d(G) = p_e R_K^d\left(G_e^*\right) + \left(1 - p_e\right) R_K^d(G \backslash e). \tag{14}$$

We denote this method as a simple factoring method (SFM). Cancela and Petingi have proposed a modified factoring method for calculating DCR [7], which is much faster than the already described classical factoring method in the diameter constrained case (14). The Cancela and Petingi factoring method (CPFM) operates with a list of paths instead of graphs. For any terminals s, t, the list $P_{st}(d)$ of all the paths with a limited length between s, t is generated. By P_d the union of $P_{st}(d)$ for all the pairs of terminals is denoted. $P(e)$ is composed of the paths of P_d which include the edge e. The list of CPFM arguments are given below [7]:

- np_{st}: the number of paths of length at most d between s and t in G;
- $links_p$: the number of non-perfect edges (edges e such that $p(e) < 1$) in the path p, for every $p \in P_d$.
- $feasible_p$: this is a flag, which has the value False when the path is no longer feasible, that is, it includes an edge which failed; and True otherwise;
- $connected_{st}$: this is a flag, which has the value True when s and t are connected by a perfect path of length at most d and False otherwise;
- *connectedPairs*: the number of pairs of terminals which are connected by a perfect path of length at most d.

Pseudocode of the method proposed for the DCR bounds cumulative updating [27], which is based on CPFM, is presented below:

```
Input: $G = (V;E)$, $d$, $P_d$, $P(e)$, $np(s; t)$, $links(p)$, $feasible(p)$, $connected(s,t)$, $connectedPairs$, $RL = 0$, $RU = 1$, $Pl = 1$.
Function FACTO ($np(s,t)$, $links(p)$, $feasible(p)$, $connected(s,t)$,$connectedPairs$, $Pr$)
if $RL > R_0$ or $RU < R_0$ then
   $e$ is an arbitrary edge: $0 < p_e < 1$
   contractEdge ($np(s; t)$, $links(p)$, $feasible(p)$, $connected(s,t)$,$connectedPairs$, $Pr$)
   deleteEdge ($np(s; t)$, $links(p)$, $feasible(p)$, $connected(s,t)$,$connectedPairs$, $Pr$)
end FACTO

Function contractEdge($np(s; t)$, $links(p)$, $feasible(p)$, $connected(s,t)$,$connectedPairs$, $Pr$)
$\mathrm{Pr} = \mathrm{Pr} * \mathrm{p_e}$
for each $p = (s,..,t)$ in $P(e)$ such that $feasible(p) = true$ do
   $links(p) = links(p) - 1$
   if $connected(s,t) = false$ and $links(p) = 0$ then
            $connected(s,t)$ true
            $connectedPairs = connectedPairs + 1$
            if $connectedPairs = |\mathrm{K}|*(|\mathrm{K}|-1)/2$ then $RL = RL + Pr$
FACTO ($np(s,t)$, $links(p)$, $feasible(p)$, $connected(s,t)$, $connectedPairs$, $Pr$)
end contractEdge

Function deleteEdge($np(s,t)$, $links(p)$, $feasible(p)$, $connected(s,t)$, $connectedPairs$, $Pr$)
$Pr = Pr *(1-p_e)$
For each $p = (s,\ldots,t)$ in $P(e)$ such that $feasible(p) = true$ do
```

```
    $feasible(p) = false$
    $np(s,t) = np(s,t) - 1$
    if $np(s,t) = 0$ then $RU = RU - Pr$
FACTO ($np(s,t)$, $links(p)$, $feasible(p)$, $connected(s,t)$, $connectedPairs$, $Pr$)
end deleteEdge
```

For the speeding up of DCR calculation, it is possible to apply methods of reduction and decomposition. In our previous studies [25–27], we have obtained such methods which can calculate the DCR faster. These methods are the analogue of the well-known series-parallel transformation for CPFM, and the pivot edge selection strategy. Also, we have obtained decomposition methods for calculating DCR in the case with two terminals. The methods obtained allow us to significantly reduce the number of recursive calls in CPFM and the complexity of DCR computation. In the above algorithm, we assume that the series-parallel reduction has been already performed and the pivot edge selection strategy is used.

Below we show how *RL* and *RU* are changing during the procedure proposed for the Intellinet network topology (**Figure 5**). The edge reliability is equal to 0.7 for each edge and the diameter value is equal to 12. The threshold value was equal to the exact DCR value, which was previously

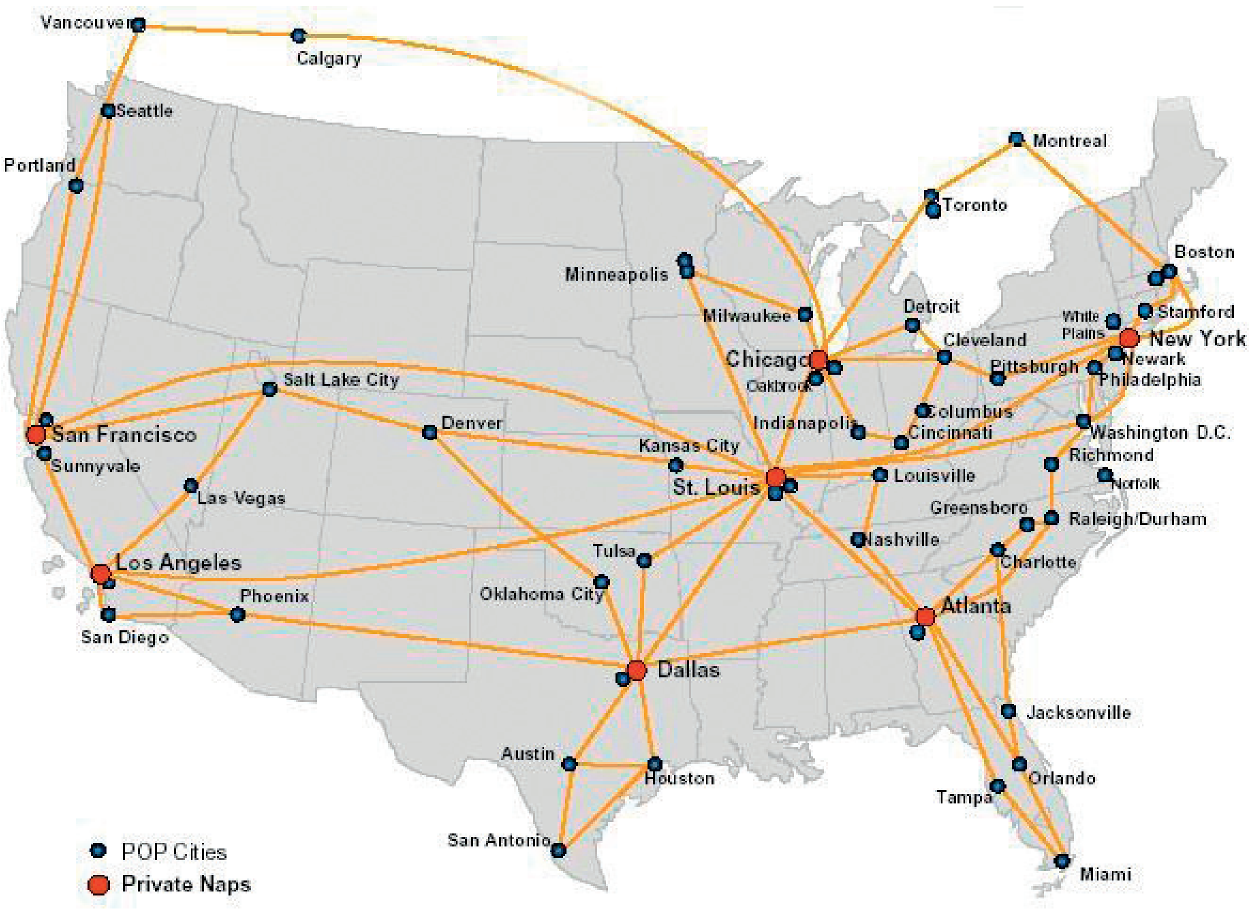

Figure 5. The Intellinet network.

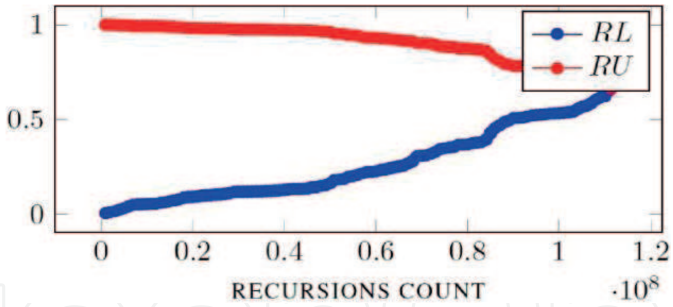

Figure 6. The behavior of DCR bounds for the Intellinet network.

calculated. The calculation time was about 12 min. Experiments were performed on Intel Xeon E31240 3.3 GHz, 8 cores (**Figure 6**).

6. Special problems of updating reliability bounds

When special methods of graph reduction or decomposition are applied for the speeding up of calculations, the corresponding equations for updating the bounds must be used. Derivation of such equations for different functions μ(G) is based on taking into account all ″inner″ changes of bounds that are concealed inside intermediate steps thus leading to final results of such a reduction or a decomposition. For example, when "branching by chain" [15] is applied, consequent factoring by edges of a chain is used for derivation, and each factoring may require a change of bounds.

Now let us turn to an example of a simple reduction, which is removing a dangling node. Let the graph G have some attached trees in its structure is shown in **Figure 7**.

In the case of the ATR, we have the trivial equality $R(G) = p_e R(G \backslash e)$, where e is the edge that connects the dangling node to the graph. As removing e with probability $1 - p_e$ leads to graph

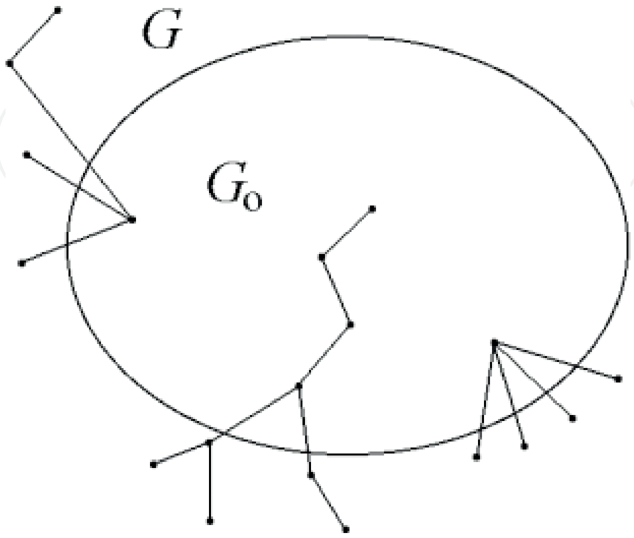

Figure 7. The graph with attached trees.

disconnection, the upper bound must be decreased by $1 - p_e$. The lower bound rests unchanged, but probabilities of all further realizations of $G\backslash e$ must be multiplied by p_e. If there are some attached trees as shown in **Figure 7**, then the ATR of the graph G is equal to the ATR of this graph without all attached trees (G_0) multiplied by the product of reliabilities of all the edges in these trees (let us denote it as *Pr*): $R(G) = \Pr \cdot R(G_0)$. Thus, we continue with the graph G_0, *LB* does not change, *Pr* must be included as a multiplier in probabilities of further realizations, and *UB* is reduced by probability of at least one edge of attached trees, that is, 1-*Pr*. If the reduction takes place at the initial step, then the upper bound starts from *Pr*.

The case of APR is not so simple. It is known that the task of obtaining this reliability index is equivalent to the one of obtaining mathematical expectation of the number of disconnected pairs of nodes in a random graph (EDP, see [28, 29]). Indeed, the following equations are valid:

$$\overline{R}(G) = \frac{C_n^2 - N(G)}{C_n^2}; N(G) = C_n^2\left(1 - \overline{R}(G)\right). \tag{15}$$

From these equations, bounds are easily obtained:

$$LB_{APR} = \frac{C_n^2 - UB_{EDP}}{C_n^2}, UB_{APR} = \frac{C_n^2 - LB_{EDP}}{C_n^2}. \tag{16}$$

In [14], the following equation was derived with deleting the dangling node t that is incidental to the node s:

$$N(G) = N(G_0) + \left(1 - p_{st}\right)w_t W(G\backslash e_{st}). \tag{17}$$

Here w_t is the weight of the node t, initially 1. This weight shows an expected number of nodes that are merged in this node during the graph transformations: when using the factoring method, for example, one should remember the number of nodes that are merged if we assume that an edge between them is reliable. This weight is used for calculating the number of disconnected pairs of nodes when the graph is divided. Let $W(G)$ be the total weight of the all nodes (initially, it is equal to the number of nodes) in G. G_0 equals G with the only difference: the weight of the node s becomes equal to $w_s + p_{st}w_t$.

From this, we obtain the following changes in the bounds: LB_{EDP} is increased by $\left(1 - p_{st}\right)$ $w_t[W(G) - w_t]$ and the UB_{EDP} is decreased by $p_{st}w_t[W(G) - w_t]$. Contrary to case of the ATR, the result of removing attached trees highly depends on their structures, so dangling nodes must be removed one at a time. The only known exception with the derived equation is the case of the attached chain [6].

7. Using cumulative bounds for network reliability approximation

A simple approximation of the function μ(*G*) through its bounds is their average: μ(*G*) = (*LB* + *UB*)/2. However, the bounds tend to the exact solution with different rates. It seems to be reasonable assuming that cross of lines, obtained by the linear approximation of curves for *LB* and *UB*, may be a better approximation of μ(*G*). Let us have the bounds LB_i and uB_i at a step $i > 0$ and let $\mu_i(G)$ be the corresponding approximation. The following proportion takes place:

$$\left(\mu_M - UB_i\right) : \left(LB_i - \mu_m\right) = \left(UB_i - \widehat{\mu}_i(G)\right) : \left(\widehat{\mu}_i(G) - LB_i\right). \tag{18}$$

Hence, we have

$$\widehat{\mu}_i(G) = \frac{\mu_M \cdot LB_i - \mu_m \cdot UB_i}{\mu_M - UB_i - \mu_m + LB_i}. \tag{19}$$

Let us name this approximation as the approximation based on trends (ABT).

In the case of ATR, $\mu_m = 0$ and $LB = UB = \mu(G)$, so

$$\widehat{R}_i(G) = \frac{LB_i}{1 - UB_i + LB_i}, \tag{20}$$

while in the case of MENC, $\mu_m = 1$ and $\mu_M = C_n^2$, so

$$\widehat{CS}_i(G) = \frac{C_n^2 \cdot LB_i - UB_i}{C_n^2 - UB_i - 1 + LB_i}. \tag{21}$$

At the last step, $LB = UB = \mu(G)$, and from (19) we have that the proposed approximation also equals this value.

In **Figures 8–10**, the behavior of bounds for the EDP of 4x4 lattice ($p = 0.7$), for the MENC of the same lattice with *c*-node 1, and for the probability of the flow transmission (PFT) between two

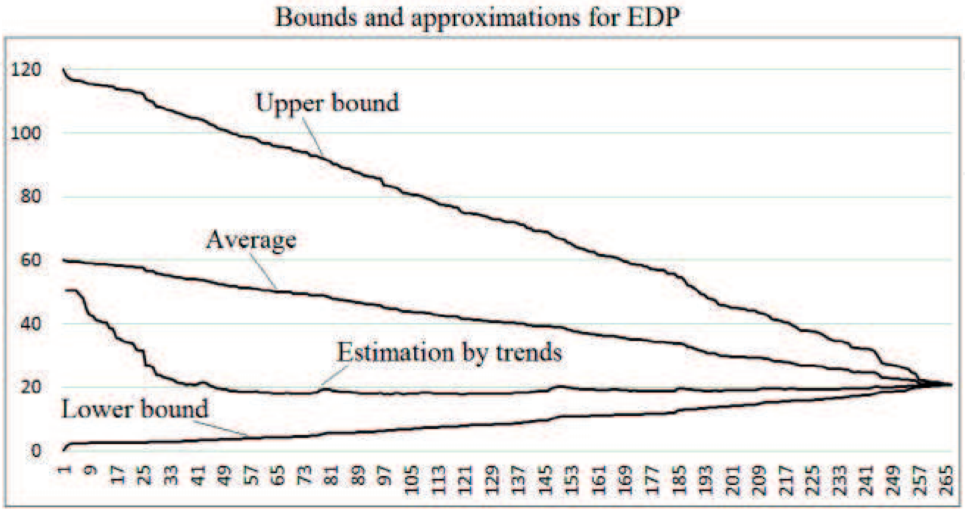

Figure 8. The behavior of bounds and approximations of EDP, the exact value is 20.915633.

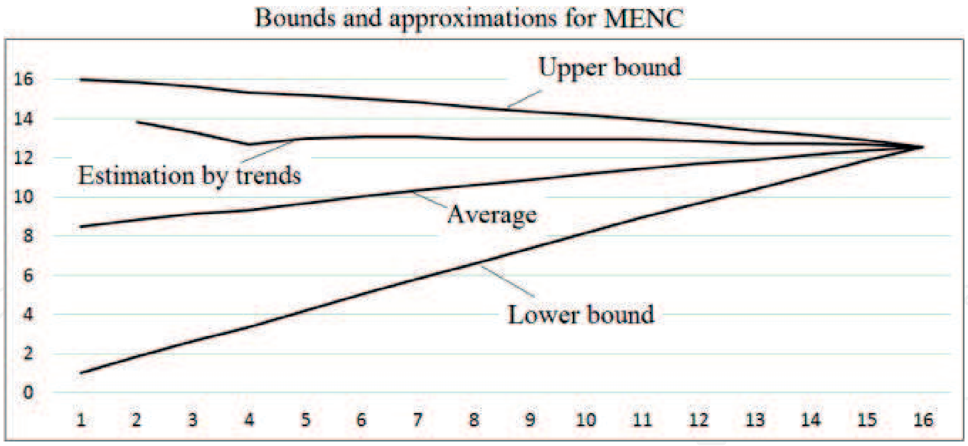

Figure 9. The behavior of bounds and approximations of MENC, the exact value is 12.562672.

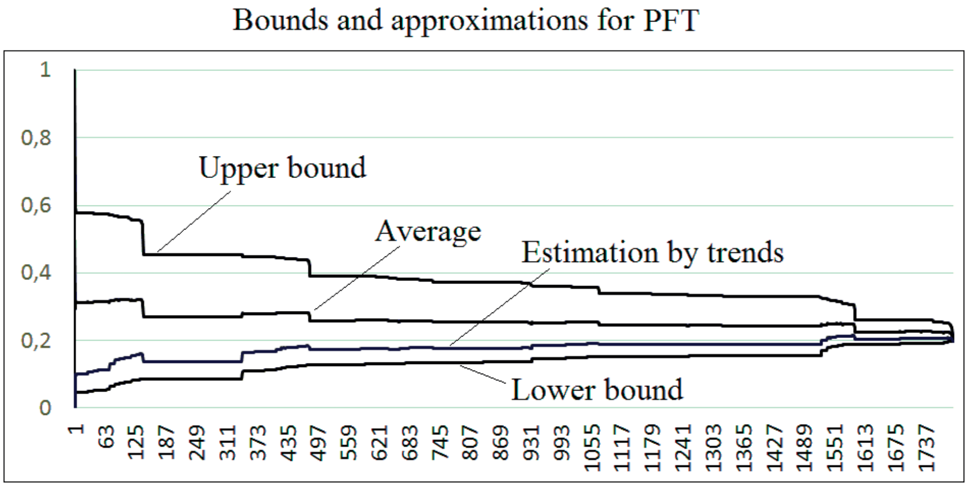

Figure 10. The behavior of bounds and approximations of PFT, the exact value is 0.196600.

diagonal corners of this lattice with throughput of all edges 1 and the edges reliability uniformly distributed between 0.5 and 1, are presented, respectively.

As we can see the ABT has become better than the average of *LB* and *UB* from the first step or very fast.

We consider the practical usage of cumulative bounds and ABT in the following section.

8. Using cumulative bounds in network topology optimization

When optimizing the network topology by a certain criteria, a fitness function (FF) must be calculated for each alternative. In our case (k-terminal reliability, MENC, and APR) this means using NP-hard algorithms for its obtaining. Note that using approximate algorithms may lead to wrong decisions when structures with close values of FF are compared.

We propose using the cumulative bounds for decision-making about whether a new solution obtained by crossover or mutation deserves including into population.

The ideas are simple (without the loss of generality, we assume that a task is maximizing some function of an unreliable network):

1. A wittingly inappropriate solution may be rejected before carrying out the exhaustive calculation of the corresponding FF. If the upper bound of the FF becomes smaller than the FF value of the worst solution in the current population, then this potential solution is rejected.
2. At initial stages, a new solution may be included into the population based on the LB value. If it exceeds the FF value of the worst solution in the current population, then this new solution substitutes the worst one with assigning ABT as an approximate value of the FF.

Note that inevitable narrowing of bounds of FF in population leads to seldom updating of the population and to a better approximation of FF.

The experiments [30, 31] with network topology optimization by the criteria of the ATR and the DCR show that using the cumulative bounds allow the speeding up of calculations up to two times without loss of precision or with negligible loss (about 10^{-7}).

9. Conclusion

The general approach of obtaining the cumulative bounds of random graph functions is presented. We have shown the possibility of computing the cumulative bounds by partial sums and updating the cumulative bounds in case of applying different methods of reduction and decomposition. For example, various indices of the network reliability were considered: the all-terminal reliability, the diameter constrained reliability, the average pairwise connectivity, and the expected size of a subnetwork that contains a special node. Also, we have described how the cumulative bounds approach can be used for network reliability evaluation and development of evolutionary algorithms for the network topology optimization. Future works can include methods of computing the cumulative bounds for other reliability indices along with methods of reduction and decomposition, further improvement of evolutionary algorithms for the network topology optimization, and parallel algorithms for the network reliability analysis.

Acknowledgements

This research was supported by the Russian Foundation for Basic Research under grants 17-07-00775, 17-47-540977.

Author details

Alexey S. Rodionov and Denis A. Migov*

*Address all correspondence to: mdinka@rav.sscc.ru

Institute of Computational Mathematics and Mathematical Geophysics of SB RAS,
Novosibirsk, Russia

References

[1] Colbourn CJ. The Combinatorics of Network Reliability. New York, NY, USA: Oxford University Press; 1987. 160 p

[2] Jereb L. Network reliability: Models, measure and analysis. In: 6th IFIP Workshop on Performance Modeling and Evaluation of ATM Networks; 1998. p. T02/1–T02/10

[3] Moore EF, Shannon CE. Reliable circuits using less reliable relays. Journal of The Franklin Institute. 1956;**262**(4b):191-208

[4] Shooman AM. Algorithms for network reliability and connection availability analysis. In: Electro/International 1995, Boston, MA, USA. 1995, pp. 309-333

[5] Sun F, Shayman MA. On pairwise connectivity of wireless multihop networks. International Journal of Security and Networks. 2007;**2**(1/2):37-49

[6] Rodionov AS, Rodionova OK. Exact bounds for average pairwise network reliability. In: The 7th International Conference on Ubiquitous Information Management and Communication; Malaysia. ACM; 2013. Article No. 45

[7] Cancela H, Petingi L. Diameter constrained network reliability: Exact evaluation by factorization and bounds. In: Int. Conf. on Industrial Logistics; Okinawa, Japan. 2001. p. 359-356

[8] Pandurangan G, Raghavan P, Upfal E. Building low-diameter peer-to-peer networks. IEEE Journal on Selected Areas in Communications. 2003;**21**(6):995-1002

[9] Valiant LG. The complexity of enumeration and reliability problems. SIAM Journal on Computing. 1979;**8**(3):410-421

[10] Bodlaender HL, Wolle T. A note on the complexity of network reliability problems. IEEE Transactions on Information Theory. 2004;**47**:1971-1988

[11] Deuermeyer LA. New approach for network reliability analysis. IEEE Transactions on Reliability. 1982;**31**(4):350-354

[12] Goyal NK, Misra RB, Chaturvedi SK. Snem: A new approach to evaluate terminal pair reliability of communication networks. Journal of Quality in Maintenance Engineering. 2005;**11**(3):239-253

[13] Won JM, Karray F. Cumulative update of all-terminal reliability for faster feasibility decision. IEEE Transactions on Reliability. 2010;**59**(3):551-562

[14] Rodionov AS. Cumulative estimated values of structural Network's reliability indices and their usage. In: Dynamics of Systems, Mechanisms and Machines; Omsk, Russia. IEEE Press; 2016. p. 1-4

[15] Rodionova OK, Rodionov AS, Choo H. Network probabilistic connectivity: Exact calculation with use of chains. In: International Conference on Computational Science and Its Application. Lecture Notes in Computer Science, vol. 3036; Heidelberg: Springer; 2004. p. 565-568

[16] Rodionov AS, Migov DA, Rodionova OK. Improvements in the efficiency of cumulative updating of all-terminal network reliability. IEEE Transactions on Reliability. 2012;**61**(2): 460-465

[17] Wood RK. Triconnected decomposition for computing K–terminal network reliability. Networks. 1989;**19**:203-220

[18] Migov DA. Using node cuts for exact calculation of network probabilistic connectivity. In: Aryn EM, editor. Proc. Int. Conf. on Computational Technologies in Science, Engineering and Education, vol. 2; 20–22–09-2006; Pavlodar, Kazakhstan. 2006. p. 51-58. (in Russian)

[19] Migov DA Random Graph Probabilistic Connectivity Calculation Using Node Cuts [Dissertation] ICM&MG SB RAS, Novosibirsk, Russia: 2008. 97 p. (in Russian) Abstract available from: http://www.sscc.ru/Diss_sov/Migov.pdf

[20] Migov DA, Rodionov AS, Rodionova OK, Choo H. Network probabilistic connectivity: Using node cuts. In: EUC Workshop. LNCS, vol. 4097; Springer; 2006. p. 702-709

[21] Migov DA. Calculation of K-terminal network probabilistic connectivity using 2-node cuts. Bull. Inst. of Comp. Math. Geoph. Ser. Computer Science. Proc. Int. Conf. on Problems of Operation of Information Networks; 7–11-08-2006; Berdsk, Russia. 2006;**6**:134-138. (in Russian)

[22] Migov DA Network probabilistic connectivity computation with use of special kind vertex cuts. In: Proc. Russian Conf. of Young Scientists on Mathematical Modeling and Information Technologies; 1–03–11-2004; Novosibirsk, Russia. ICT SB RAS:2004. (in Russian) Available from: http://www.ict.nsc.ru/ws/YM2004/8566/Paper.htm

[23] Burgos JM, Amoza FR. Factorization of network reliability with perfect nodes I: Introduction and statements. Discrete Applied Mathematics. 2016;**198**:82-90

[24] Burgos JM. Factorization of network reliability with perfect nodes II: Connectivity matrix. Discrete Applied Mathematics. 2016;**198**:91-100

[25] Migov DA. Computing diameter constrained reliability of a network with junction points. Automation and Remote Control. 2011;**72**(7):1415-1419

[26] Migov DA, Rodionov AS. Decomposing graph with 2-node cuts for diameter constrained network reliability calculation. In: 7th Int. Conference on Ubiquitous Information Management and Communication; Malaysia. New York: ACM; 2013. Article No. 39

[27] Migov DA, Nesterov SN. Methods of speeding up of diameter constrained network reliability calculation. In: International Conference on Computational Science and its Application. LNCS, Vol. 9156(2); Springer. 2015. pp. 121-133

[28] Rodionov AS, Rodionova OK. Network probabilistic connectivity: Expectation of a number of disconnected pairs of nodes. In: High Performance Computing and Communications. Lecture Notes in Computer Science, vol. 4208; Berlin: Springer; 2006. p. 101-109

[29] Rodionov AS, Rodionova OK, Choo H. On the expected value of a number of disconnected pairs of nodes in unreliable network. In: Computational Science and its Applications. Lecture Notes in Computer Science, vol. 4707; Berlin: Springer; 2007. p. 534-543

[30] Nechunaeva K, Migov D. Speeding up of genetic algorithm for network topology optimization with use of cumulative updating of network reliability. In: Conference on Ubiquitous Information Management and Communication; Bali, Indonesia. New York: ACM; 2015. p. Article No. 42

[31] Migov DA, Nechunaeva KA, Nesterov SN, Rodionov AS. Cumulative updating of network reliability with diameter constraint and network topology optimization. In: Computational Science and its Applications. Lecture Notes in Computer Science, vol. 9786; Berlin. Springer; 2016. p. 141-152