Practical Notes on Obtaining Reliability Polynomials

ALEXEY S. RODIONOV
ICM&MG SB RAS
Laboratory "Modeling dynamical processes in information nets"
6 Prospekt Lavrentieva, 630090 Novosibirsk
RUSSIAN FEDERATION
alrod@sscc.ru

OLGA RODIONOVA
Novosibirsk State University
Higher College of Informatics
35 Russkaya, 630058 Novosibirsk
RUSSIAN FEDERATION
rolcon@mail.ru

Abstract: General technical problems of obtaining coefficients of reliability polynomials for different reliability indices, such as all-terminal reliability, average pairwise reliability and average size of a connected sub-graph containing a special node are discussed in the paper. It is shown that one of possible forms of a reliability polynomial presentation helps in considerable speeding up of its obtaining by using an intentional meaning of coefficients. Procedure of calculations in this case can be reduced to summarizing parts of vectors of binomial coefficients.

Key–Words: Network reliability, Reliability polynomials

1 Introduction

This paper is generalization and extension of authors' results in obtaining and usage reliability polynomials of random graphs [14, 15, 11, 12].

Reliability polynomial of random graphs with unreliable elements are investigated for a long time and by many authors [2, 6, 7, 1, 9, 8], but most papers concerns theoretical aspects only but not algorithms of their obtaining, mainly in connection with Tutte polynomials [4].

We use mostly common and well-explored model of a random multi-graph with n reliable nodes and m unreliable edges that may fail independently with a probability $q = 1 - p$, where p is named edge's *reliability*. Full denotation for such graph is $G(n, m, p)$, but for short we use G or $G(n, m)$ if rest parameters are clear from a context.

Most often polynomials for all-terminal reliability (ATR) of a random graph with unreliable edge are considered. In [9] reliability polynomials are used for estimating an "edge's impotence" in a graph's structure: edge, whose removal leads to a maximal decreasing of a polynomial's plot, is considered as contributed maximal income into graph's reliability. At the same time it is known that for some indices reliability polynomials for graphs with the same number of nodes and edges may intersect (see [11, 6, 7], for example). Thus as choice of optimal structure, as finding edges' "significance" depends on edges' reliability. In [2] different forms of presentation of these polynomials are presented and usage of coefficients for graph's properties analysis is discussed. In [12] one of the authors of current paper had shown how the meaning of coefficients of a reliability polynomial for ATR in one of its forms can be used for considerable speeding up of their obtaining. Hereafter we show how meaning of coefficients can be practically used for their faster obtaining in the case of some other reliability indices.

In the current paper we consider common and special technical problems of obtaining polynomials for all-terminal reliability (ATR-polynomial, $R(G, p)$), mathematical expectation of a number of disconnected pairs of nodes (EDP-polynomial, $N(G, p)$), and mathematical expectation of a number of nodes in a connected subgraph that contains some special node (control center in an unreliable network, hereafter referred as c-node) (MENC-polynomial, $C(G, p)$).

Note, that we consider EDP-polynomials because the task of obtaining this index is equivalent to the task of obtaining average pairwise connectivity (APC, $\bar{R}(G)$). Indeed, the following equations are clear:

$$\bar{R}(G) = \frac{C_n^2 - N(G)}{C_n^2}, \quad (1)$$

$$N(G) = C_n^2 \left(1 - \bar{R}(G)\right). \quad (2)$$

From this we have that if EDP-polynomial is

$$N(G, p) = \sum_{i=0}^{m} n_i (1-p)^i p^{m-i},$$

then APC-polynomial is

$$\bar{R}(G, p) = \sum_{i=0}^{m} \frac{C_n^2 - n_i}{C_n^2} (1-p)^i p^{m-i}. \quad (3)$$

EDP was examined in [10, 5].

Most of proposed solutions may be efficiently used for obtaining polynomials for other indices of random graphs.

The rest of the paper is organized as follows. In Section 2 we present main denotations and assumptions used in the paper. In Section 3 the factoring method is described in connection with polynomial's representation. Sections from 4 to 9 are devoted to treating different structural particularities of graphs. Section 10 shows how to obtain some coefficients by analyzing initial graph's structure without complex calculations. In Section 11 we show how to obtain coefficients of different reliability polynomials on the example of small graph with chains in its structure, while the last Section is a brief conclusion.

2 Main denotations and assumptions

For further derivations we need the following denotations:

$\underline{k}$ — set $(1,\ldots,k)$;

$\overline{i,j}$ — set $(i,\ldots,j),\ i \leq j$;

$G(n,m) = (V,U,\Lambda,WT)$ — non-oriented multigraph with a set of nodes V, set of edges U, matrix of edges multiplicities Λ and vector of nodes weights WT. An edge is usually denoted as e_{ij}, that means that it connects nodes v_i and v_j, but sometimes it is more convenient denote it as e_i (i-th edge). Corresponding multiplicity may be denoted as λ_{ij} or λ_i;

$\mathbb{T}(n,\Lambda), \mathbb{C}(n,\Lambda)$ — n-nodes tree or cycle of multi-edges;

$n = |V|$**,** $m = |U|$ — number of nodes and edges, respectively;

$w_i = w(v_i)$ — weight of a node v_i, $WT = w_1,\ldots,w_n$;

$W(G)$ — total weight of all nodes of G;

$W^*(G)$ — total sum of pairwise productions of weights of all nodes of G (if all $w_i = 1$, then $W^*(G) = C_n^2$);

k-**multi-edge** — multi-edge with multiplicity k;

p — probability of an edge being existent (being in a working state, edge's *reliability*), $q = 1 - p$;

Ch_k — chain, composed of k edges $e_1,\ldots,e_k$;

$G/Ch\,(G/e)$ — network, obtained from G by contracting pair of nodes by a chain Ch (edge e);

$G\backslash C\,(G\backslash e)$ — subnetwork of G, obtained by deleting chain Ch (edge e);

$S(U)$ — sum of multiplicities of edges with numbers from set U;

$L(\vec{S})$ — number of elements in a vector $\vec{S}$ (its dimension).

Some polynomials are frequently used in our derivations so we give them special denotations:

$$Q_s(p) = (1-p)^s; \quad (4)$$

$$M_s(p) = 1 - Q_s(p) = I(s) - (1-p)^s = \sum_{i=0}^{s-1} C_s^i p^{s-i}(1-p)^i; \quad (5)$$

$$D(\vec{S},p) = \prod_{i=1}^{L(\vec{S})} M_{s_i}(p); \quad (6)$$

$$Z(\vec{S},p) = \sum_{i=1}^{L(\vec{S})} Q_{s_i}(p) \prod_{j=1,\,j\neq i}^{L(\vec{S})} M_{s_i}(p). \quad (7)$$

$Q_s(p)$ corresponds to the probability that multi-edge with the multiplicity s fails completely, while $M_s(p)$ shows the probability that al least one of its edges is in working state. Note that $Q_{k+l}(p) = Q_k(p)Q_l(p)$.

Note also that $D(\vec{S},p)$ is the RP of a tree while $D(\vec{S},p) + Z(\vec{S},p)$ is the RP of a cycle with vector of edges' multiplicities $\vec{S}$ (these cases are discussed in Section 6 in details).

For short, if it does not lead to variant reading, then we use "edge" for "multi-edge" in the paper. If it does matter, we use "multi-edge" or "single edge".

In some of following equations we need use powers of polynomials, let us denote power of a polynomial $Pol(p)$ as $\uparrow Pol(p)$.

For shortening some expressions we assume that if $j > k$, then $\prod_{i=j}^{k} p_i = 1$.

Though a node's weight is a polynomial or fractional polynomial in general case, we usually skip argument for shorten expressions.

3 Factorization Method and Polynomial Representation

As it is noted in [2], there are various ways of a reliability polynomial representations. In some cases coefficients have intensional meaning. In [12] the presentation

$$R(G,p)=\sum_{i=0}^{m} a_i(1-p)^i p^{m-i}, \quad (8)$$

had been efficiently used for speeding up obtaining coefficients of ATR-polynomial. It is known that a_i is equal to a number of connected sugraphs (subgraphs on complete set of nodes), that may be obtained by removing exactly i edges. Coefficients of polynomial in its classic presentation

$$R(G,p)=\sum_{i=0}^{m} b_i p^i \quad (9)$$

are connected with those in (8) by the following equations:

$$\begin{aligned} &b_0 = a_m;\\ &b_{m-i} = \sum_{j=i}^{m}(-1)^{i+j}C_j^i a_j, i=0,\ldots,m-1; \quad (10)\\ &a_m = b_0;\\ &a_i = b_{m-i}+\sum_{j=i+1}^{m}(-1)^{i+j-1}C_j^i a_j,\ i{=}m{-}1,\ldots 0. \quad (11)\end{aligned}$$

Note that the meaning of coefficients in (8) usually allows obtaining some coefficients directly and may be used for derivation of finite expressions for graphs of small dimension (2-5 nodes) or of some special kind (trees, cycles) only. At the same time this presentation well corresponds for the factorization method [14, 16]:

$$R(G,p)=pR(G/e_{ij})+(1-p)R(G\backslash\{e_{ij}\}). \quad (12)$$

We use this method as basic one hereafter.

3.1 Supporting right kind of a polynomial

In (8) all summands have the same total power of p and $(1-p)$, while during possible decompositions and reductions of intermediate graphs, obtained during calculations, we can obtain polynomials with different powers between and inside them. For equalizing powers, polynomials or summands with lesser power are multiplied by the special polynomial of the kind (8) that is identically equal to one (its power is equal to a difference of powers):

$$\begin{aligned} I(n) &\equiv 1 \equiv 1^n \equiv (1-p+p)^n\\ &= \sum_{i=0}^{n} C_n^i p^i (1-p)^{n-i},\\ I(0) &= 1. \quad (13)\end{aligned}$$

For example,

$$\begin{aligned} &p(1-p)+p(1-p)^3 = I(2)p(1-p)+p(1-p)^3\\ &= [p^2+2p(1-p)+(1-p)^2]p(1-p)+p(1-p)^3\\ &= p^3(1-p)+2p^2(1-p)^2+2p(1-p)^3.\end{aligned}$$

Most important is that obtaining coefficients of some polynomial

$$P(p)=\sum_{i=1}^{k} a_i I(s_i) p^{u_i}(1-p)^{m-s_i-u_i}$$

we can reduce to summarizing shifted vectors $a_i Bin(s_i)$, where $Bin(n)$ is a vector of binomial coefficients $C_n^i,\ i=1,\ldots,n$. Next highly important point is that we need not calculate those coefficients, that can be obtained beforehand by their meaning, thus reducing number of operations when summarizing vectors: parts of vectors that corresponds to unknown coefficients must be treated only. High efficiency of this approach is shown in [12] on example of ATR-polynomial.

Note. Shift of a vector of binomial coefficients depends on power of $(1-p)$ only, thus only this power must be traced in the calculation process.

General scheme of the factorization method

General recursive scheme of the method is as follows:

1. Check if a graph allows direct obtaining of coefficients. If YES, then calculate them and exit, else go to the next step.

2. Check if graph's reduction is possible. If YES, then do it and go back to the step 1, else go to the next step.

3. Choose a pivot element and execute factorization, that is prepare pair of graphs and make recursive calls of the procedure.

Note that in some cases reduction leads to obtaining more than one graph of smaller dimension and, consequently, several calls of basic procedure are needed.

One can see that the factoring method is more effective if:

1. Factoring is terminated on the highest possible level;

2. Dimension of a graph under consideration can be reduced.

Thus the general scheme of construction a factoring algorithm for obtaining polynomial for some new reliability (and not only reliability) index includes the following steps:

1. Obtaining polynomials for graphs of small dimensions (usually up to 5 nodes).

2. Obtaining polynomials for graphs of special kinds (trees, cycles, ladders, etc.).

3. Finding ways for reducing graph's dimension (removing dangling nodes, replacing chains by edges, etc.).

4. Finding ways for graph's decomposition by using graph's particularities, such as cutnodes, bridges and node cuts.

5. Design of effective algorithms for program realization of mentioned reduction and decomposition, if ways are found.

6. Examination of coefficients' meaning and finding values and expressions for some of them thus allow their obtaining directly before executing main procedure.

4 Handling dangling nodes

Dangling nodes (for short we will name it as d-node further on) are common in a structure of real networks. In most cases such nodes can be deleted from a graph before calculations.

4.1 ATR-polynomial

This case is simplest and obvious one: for all-terminal connectivity an edge (e) that is incidental to a d-node must exist. If not, then a graph is disconnected. Thus, from (12) we have that

$$R(G,p) = pR(G/e). \tag{14}$$

If d-node is connected with the rest of a graph by a multi-edge E of multiplicity k, then

$$R(G,p) = [I(k) - (1-p)^k]R(G/E). \tag{15}$$

Note, that for ATR G/e (G/E) means that d-node and its adjacent node are contracted.

4.2 EDP-polynomial

As it is shown in [10], we can remove a dangling node (v_t) by contracting it with its adjacent node (v_s) using the following rule:

$$N(G,p) = N(G^*,p) + Q_{\lambda_{st}}(p)w_t, \tag{16}$$

where G^* has the structure of G/e_{st} and *weight* of a node v_s equal to $w_s^* = w_s + p_{st}w_t$. Weight of a node means expected value of a number of nodes, contracted into it during factoring and reduction processes. Initially all w_i are equal to one, but then they are polynomials $w_i(p)$, in general case. In our case e_{st} may be a multi-edge E of multiplicity k, thus

$$\begin{aligned} &N(G,p) = N(G^*,p) + (1-p)^k w_t(p) \\ &= N(G^*,p)I(\max(0, \uparrow N(G^*,p) - \uparrow w_t(p) - k)) + \\ &(1-p)^k w_t(p)I(\max(0, \uparrow w_t(p) + k - \uparrow N(G^*,p)), \end{aligned} \tag{17}$$

and

$$\begin{aligned} &w_s^*(p) = w_s(p)I(\max(0, \uparrow w_t(p) + k - \uparrow w_s(p)) + \\ &\sum_{i=1}^{k} C_k^i p^{k-i}(1-p)^i w_t(p) \times \\ &\quad I(\max(0, \uparrow w_s(p) - k - \uparrow w_t(p)). \end{aligned} \tag{18}$$

Note. More effective way of utilizing independent of G summands is their separate accumulation and addition of final sum to the result on the last stage. In this case we need not spend efforts on equalizing powers of $N(G^*,p)$ and $(1-p)^k w_t(p)$.

4.3 MENC-polynomial

Similar to the previous index, MENC requires weighted nodes. There are several ways of obtaining this polynomial. Most obvious is through obtaining all polynomials for two-terminal probabilistic connectivity $R_{1i}(G,p)$:

$$\mathbb{C}(G,p) = w_1 + \sum_{i=2}^{n} w_i R_{1i}(G,p). \tag{19}$$

It is clear that if some node v_t is a d-node, then $R_{st}(p)=R_{sx}(p)M_{xt}(p)$, where v_x is adjacent to v_t. From this and remembering about nodes' weights, we have that any dangling node, if it is not a c-node, may be deleted while its adjacent node increases its weight by $M_{\lambda_{st}}(p)w_t$. If v_t is a c-node, then it is deleted, and v_s not only changes its weight, but became a new c-node in a reduced graph.

5 Handling trees

If a graph is a tree of single edges, then it is usually an obvious case. More complex it became if multi-edges are allowed.

5.1 ATR-polynomial

Equation for a tree of multi-edges is presented in [12]. In denotations of the current paper it looks as

$$R(\mathbb{T},p) = D(\vec{\Lambda},p) = \sum_{i=0}^{m} C_m^i p^i (1-p)^{m-i} + \sum_{U \subset \underline{n-1}} (-1)^{|U|} \sum_{j=0}^{S(U)} C_{m-W(U)}^j p^j (1-p)^{m-j}. \quad (20)$$

5.2 EDP-polynomial

$$N(\mathbb{T},p) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} w_i w_j \Big(1 - \prod_{e_{st} \in Pt_{ij}} M_{\lambda_{st}}(p)\Big), \quad (21)$$

where Pt_{ij} is an unique path between v_i and v_j. thus, if no multi-edges, then

$$N(\mathbb{T}(n),p) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} w_i w_j \Big(1 - p^{d_{ij}}\Big), \quad (22)$$

where d_{ij} is a number of edges between v_i and v_j. In partial cases we have simpler equations.

5.2.1 Chain

Let us have a chain $Ch(n,m)$ with n nodes and $n-1$ multi-edges e_i with multiplicities λ_i. There are $k-1$ pairs of nodes with distance 1, $k-2$ pairs with distance 2, etc. Thus

$$N(Ch(n,m),p) = C_n^2 - \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} w_i w_j \Big[1 - \prod_{s=i}^{j-1} M_{\lambda_s}(p)\Big]. \quad (23)$$

If no multi-edges, then

$$N(Ch(n,n-1),p) = C_k^2 - \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} w_i w_j (1-p^{j-i}), \quad (24)$$

and, if all weights are unit, then

$$N(Ch(n,n-1),p) = C_n^2 - \sum_{i=1}^{n-1} i(1-p^{n-i}). \quad (25)$$

5.2.2 Star

Let us have a star-like graph $S(n,m)$, in wich $n-1$ nodes are adjacent to one (center). Thus we have a node v_1 with weight w_1, that is adjacent to $n-1$ d-nodes v_i, $i = 2,\ldots,n$ with weights w_i by edges e_{1i}. Using (21) we have:

$$N(S(n,m),p) = \sum_{i=2}^{n} w_1 w_i [1 - M_{\lambda_i}(p)] + \sum_{i=2}^{n-1} \sum_{j=i+1}^{n} w_i w_j [1 - M_{\lambda_i}(p) M_{\lambda_j}(p)]. \quad (26)$$

In the simplest case of no multi-edges and unit weights we obtain

$$N(S(n,m),p) = C_n^2 - (n-1)p + C_{n-1}^2 p^2. \quad (27)$$

5.3 MENC-polynomial

This case differs from the previous one by considering only pathes from a c-node (let it be v_1) to all rest nodes, and by considering connection, not disconnection of nodes. Thus

$$C(\mathbb{T}(n)) = w_1 + \sum_{i=2}^{n} w_i \prod_{e_{st} \in Pt_{1i}} M_{\lambda_{st}}(p), \quad (28)$$

and, if no multi-edges, then

$$C(\mathbb{T}(n),p) = w_1 + \sum_{i=1}^{n} w_j p^{d_{1j}}. \quad (29)$$

Equations for chain and star depends on placement of a c-node.

5.3.1 Chain

If c-node has number 1, then

$$C(Ch,p) = w_1 I(m) + \sum_{i=2}^{n} w_i \prod_{j=1}^{i-1} I(m-\lambda_i) M_{\lambda_i}(p). \quad (30)$$

In the case of no multi-edges we have

$$C(Ch,p) = \sum_{i=1}^{n} w_i I(n-i) p^{i-1}. \quad (31)$$

Now let a c-node have number $1 < s < k$. Without loss of generality we assume that $s \le n-s$. Omitting simple reasoning we have:

$$C(Ch,p) = \sum_{i=1}^{k} w_i \prod_{j=\min(i,s)}^{\max(s-1,i-s+2)} M_{\lambda_j}(p) \times \left(1 - \prod_{j=1}^{s-1} M_{\lambda_j}(p) \prod_{j=i}^{k-1} M_{\lambda_j}(p)\right). \quad (32)$$

If no multi-edges, then

$$C(Ch,p) = \sum_{i=1}^{k} w_i p^{|s-i|}\left(1 - p^{|k-s+i-1|}\right), \quad (33)$$

or, considering ratio between s and k,

$$C(Ch,p) = w_s + \sum_{i=1}^{s-1}(w_i + w_{2s-i})p^{s-i} + \sum_{i=s+1}^{k} w_i p^{i-s} - \left(\sum_{i=1}^{k} w_i\right) p^k. \quad (34)$$

5.3.2 Star

There are two possible ways of a c-node placement.

Case 1: c-node is a central one, let it be v_0. For simple we denote edges (v_0, v_i), $i = 1, \ldots, k$ as e_i their multiplicities and weights as λ_i and w_i, correspondingly. From this we have

$$C(G) = w_0 I(m) + \sum_{i=1}^{k} w_i M_{\lambda_i}(p) I(m - \lambda_i). \quad (35)$$

If no multi-edges, then

$$C(G) = w_0 I(1) + p\sum_{i=1}^{k} w_i = w_0(1-p) + p\sum_{i=0}^{k} w_i. \quad (36)$$

Case 2: c-node is a leaf. Let it be v_0 and let central node be v_1. According to (19) and using (35) we obtain

$$C(G) = w_0 I(n-1) + p[w_1 I(n-2) + \sum_{i=2}^{k} w_i I(m-\lambda_i) M_{\lambda_i}(p)]. \quad (37)$$

If no multi-edges, then

$$C(G) = w_0 I(2) + I(1)pw_1 + p^2\sum_{i=1}^{k} w_i = w_0(1-p)^2 + (2w_0 + w_1)p(1-p) + \left(\sum_{i=0}^{k} w_i\right)p^2. \quad (38)$$

6 Handling Cycle-shaped graphs

6.1 ATR-polynomial

Equation for obtaining ATR-polynomial for a cycle of multi-edges is presented in [12]. Let us cite it adopting to denotations of current paper:

$$R(\mathbb{C}(n,\vec{\Lambda}) = \prod_{i=1}^{n} M_{\lambda_i}(p) + \sum_{i=1}^{n} Q_{\lambda_i}(p) \prod_{\substack{j=1\\ j\neq i}}^{n} M_{\lambda_j}(p) = Z(\vec{\Lambda},p) = I(m) + \sum_{U\subset \underline{n}} (|U|-1)(-1)^{|U|+1}(1-p)^{S(U)} I(S(\underline{n}\backslash U)). \quad (39)$$

6.2 EDP-polynomial

Equation for this case may be found in [5]. In denotations of current paper it looks as:

$$N(\mathbb{C}(n,m),p) = \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} w_i w_j \left[1 - \prod_{s=i}^{j-1} M_{\lambda_s}(p)\right] \times \left[1 - \prod_{s=j}^{n} M_{\lambda_s}(p) \prod_{s=1}^{i-1} M_{\lambda_s}(p)\right]. \quad (40)$$

In the case of no multi-edges we obtain

$$N(\mathbb{C}(n,n),p) = \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} w_i w_j I(n) - I(n-j+i)p^{j-i} - I(j-i)p^{n-j+i} + p^n \quad (41)$$

6.3 MENC-polynomial

Let v_1 be a c-node. There are two pathes from it to any other node of a cycle, thus

$$\mathbb{C}(G,p) = w_1 + \sum_{i=2}^{n}\left(\prod_{s=2}^{i-1} M_{\lambda_s}(p) + \prod_{s=i}^{n} M_{\lambda_s}(p)\right) - (n-1)\prod_{s=2}^{n} M_{\lambda_s}(p). \quad (42)$$

7 Handling cutnodes

Let a graph G have a cutnode v_x that connects two its components: G_1 and G_2. Without loss of generality we can assume that nodes from v_1 to v_{x-1} belongs to G_1, from v_{x+1} to v_n – to G_2, while v_x belongs to both.

7.1 ATR-polynomial

Quite obviously $R(G,p) = R(G_1,p)R(G_2,p)$.

7.2 EDP-polynomial

Handling cutnodes for speeding up calculation of EDP in case of different edges' reliabilities have been discussed in [11]. In the case of equal reliabilities we have:

$$N(G,p) = N(G_1,p) + N(G_2,p) + \sum_{i=1}^{x-1}\sum_{j=x+1}^{n} w_i w_j \left[1 - R_{ix}(G_1,p)R_{xj}(G_2,p)\right]. \quad (43)$$

7.3 MENC-polynomial

As usual, we consider v_1 as a c-node. It is clear that

$$\mathbb{C}(G,p){=}\mathbb{C}(G_1,p){+}R_{1x}(G_1,p)\sum_{i=x+1}^{n} R_{x,i}(G_2,p)w_i. \quad (44)$$

8 Handling bridges

If there is a bridge in a graph, then we make factoring by it, thus obtaining a disconnected 2-component graph (let components be G_1 and G_2) and a graph with a cutnode. Last case have been discussed above, handling a disconnected graph depends on index under consideration. Without loss of generality we can assume that nodes in G_1 are numbered from 1 to s and in G_2 — from $t = s + 1$ to n, and that e_{st} is a bridge.

8.1 ATR-polynomial

Quite obviously

$$R(G,p) = M_{\lambda_{st}}(p)R(G_1,p)R(G_2,p).$$

8.2 EDP-polynomial

Note that in a 2-component graph all pairs in which nodes belongs to different components are disconnected. Thus

$$N(G,p) = M_{\lambda_{st}}(p)\left\{ N(G_1,p) + N(G_2,p) + \sum_{i=1}^{s}\sum_{j=t}^{n} w_i w_j \left[1 - R_{is}(G_1,p)R_{tj}(G_2,p)\right] \right\} + Q_{\lambda_{st}}(p)W(G_1)W(G_2). \quad (45)$$

8.3 MENC-polynomial

As usual, we consider v_1 as a c-node. It is clear that

$$C(G,p) = C(G_1,p) + R_{1s}(G_1,p)M_{\lambda_{st}}(p)\left\{ \sum_{i=s+1}^{n} R_{x,i}(G_2,p)w_i \right\} \quad (46)$$

9 Handling chains

While substituting chains by edges is very effective when calculating k-terminal reliability (probabilistic connectivity), it is not so when obtaining reliability polynomials, partially due to occurrence of intermediate fractional polynomials. Best choice in this case is trying "branching by chain".

9.1 ATR-polynomial

Main way of a graph's dimension reduction is "branching by chain" [13]. Let us cite the following theorem from [12]:

Theorem 1 *Let graph G have a chain Ch, that consists of k multi-edges e_1, e_2, …, e_k with multiplicities λ_1, λ_2, …, λ_k, correspondingly, that connects nodes v_s and v_t. Then*

$$R(G,p){=}\Big[I(\lambda_{st},p)\prod_{i=1}^{m} M_{\lambda_i}(p){+}M(\lambda_{st},p) \times \sum_{i=1}^{k} N(\lambda_i,p)\prod_{j\neq i} M_{\lambda_i}(p)\Big] R(G/Ch,p) + N(\lambda_{st},p)\sum_{i=1}^{k} N(\lambda_i,p)\prod_{j\neq i} M_{\lambda_i}(p)R(G\backslash Ch,p), \quad (47)$$

where λ_{st} – multiplicity of edge that connects ending nodes of Ch directly (zero if no such edge).

Proof of the theorem is based on consequent factoring by edges of a pivot chain. After deleting of an edge a graph with attached chain that ends by a d-node is obtained, and after contracting pair of nodes — a graph with shorter chain. If there exists an edge directly connecting terminal nodes of a chain, then first factoring is made by it and a graph is obtained with cutnode between a cycle and a graph, in which a chain is deleted and its terminal nodes are contracted. After consequent using equations for deleting d-nodes and handling cutnodes, and making final collecting of terms we obtain what required.

The same scheme is used for obtaining equations for branching by a chain in the case of other reliability indices.

9.2 EDP-polynomial

Because graphs, obtained after removing d-nodes during factoring by chain's edges, are similar in structure but have different weights of chain's terminal nodes [11, 5], equations are rather complex and here we consider the case of a 2-edge chain (e_{sx}, e_{xt}) and no e_{st} only. First we cite the following theorem from [10]:

Theorem 2 *If a connected random graph has a simple chain $Ch = (e_{sx}, e_{xt})$ connecting nodes s and t through a node v_x with degree 2, then the following equation is true:*

$$N(G) = \big[p_{st}(1-p_{sx}p_{xt}) + p_{sx}p_{xt}\big]N(G^*/Ch) + \big(1-p_{st}-p_{sx}p_{xt}\big)N(G^o\backslash Ch) - a_{st}\frac{(1-p_{st})p_{sx}p_{xt}(1-p_{sx})(1-p_{xt})}{1-p_{sx}p_{xt}}w_x^2 + (1-p_{sx})(1-p_{xt})\big(W(G)-w_x\big)w_x. \quad (48)$$

where a_{st} is a probability of v_s and v_t being disconnected in $G\backslash Ch\backslash e_{st}$, G^*/Ch differs from G/Ch by a weight of joint node and $G^o\backslash Ch$ differs from $G\backslash Ch$ by weights of terminal nodes of a chain:

$$WT(G^*, sxt) = w_s + w_t + \frac{(p_{sx}p_{st} + p_{sx}p_{xt} + p_{st}p_{xt} - 2p_{sx}p_{st}p_{xt})}{p_{st} + p_{sx}p_{xt} - p_{sx}p_{st}p_{xt}}w_x. \quad (49)$$

$$\begin{aligned} WT_s(G^o) &= w_s + w_x\frac{p_{sx}(1-p_{xt})}{1-p_{sx}p_{xt}}; \quad (50)\\ WT_t(G^o) &= w_t + w_x\frac{(1-p_{sx})p_{xt}}{1-p_{sx}p_{xt}}. \end{aligned}$$

From this we easily obtain

$$N(G,p) = \Big[1-(1-p)^{\lambda_{st}}Z(p)\Big]\,N(G^*,p)+ \quad (51)$$

$$(1-p)^{\lambda_{st}}Z(p)N(G^o,p) - a_{st}(p)\frac{(1-p)^u\,[1-Z(p)]}{Z(p)}w_x^2 + [1-Z(p)]\,w_x\,[W(G)-w_x]\,; \quad (52)$$

$$WT(G^*, sxt) = w_s + w_t + \frac{WP_1(p)}{WP_2(p)}w_x, \quad (53)$$

where

$$Z(p) = M_{\lambda_{sx}}(p)M_{\lambda_{xt}}(p); \quad (54)$$

$$WP_1(p) = M_{\lambda_{st}}(p)M_{\lambda_{sx}}(p) + M_{\lambda_{st}}(p)M_{\lambda_{xt}}(p) + Z(p) - 2M_{\lambda_{st}}(p)M_{\lambda_{sx}}(p)M_{\lambda_{xt}}(p),$$

$$WP_2(p) = M_{\lambda_{st}}(p) + Z(p) - M_{\lambda_{st}}(p)M_{\lambda_{sx}}(p)M_{\lambda_{xt}}(p)(p);$$

$$WT_s(G^o) = w_s + w_x\frac{M_{\lambda_{sx}}(p)Q_{\lambda_{xt}(p)}}{1-Z(p)}; \quad (55)$$

$$WT_t(G^o) = w_t + w_x\frac{M_{\lambda_{xt}}(p)Q_{\lambda_{st}(p)}}{1-Z(p)}. \quad (56)$$

Note. If we make not branching by a chain, but make consequent factoring by its edges, then we do not obtain fractional polynomials. So we can mark the input point into branching and make reduction of fractions after return from recursions.

9.3 MENC-polynomial

There are several ways of obtaining this polynomial. Most obvious is through obtaining all polynomials for two-terminal probabilistic connectivity $R_{1i}(G, p)$:

$$\mathbb{C}(G, p) = w_1 + \sum_{i=2}^{n} w_i R_{1i}(G, p). \quad (57)$$

Serial-parallel reduction of a graph when obtaining $R_{1i}(G, p)$ by using factoring method is highly efficient and well-known technique [3] and is used when terminal nodes do not considered as inner nodes of chains.

Let v_i be an inner node of some chain with terminal nodes v_s and v_t. Without loss of generality we can assume that nodes of a chain have numbers from s to t in increasing order and $s < i < t$. Let edges of a Ch be $e_s = (v_s, v_{s+1}), \ldots, e_{s+k-1} = (v_{s+k-1}, v_t)$. Let A be an event that nodes v_1 and v_i are connected, B and E — that pairs of nodes $v_1 - v_s$, and $v_1 - v_t$ are connected in $G\backslash Ch$. We can obtain a probability of A as:

$$P(A) = P(BE)P(A|BE) + P(B\bar{E})P(A|B\bar{E}) + P(\bar{B}E)P(A|\bar{B}E). \quad (58)$$

It is easy to see that $P(B\cup E) = R_{1,(s,t)}(G/Ch)$, where (s, t) is a number of a node that is obtained by contracting nodes v_s and v_t. From this and using meaning of events B and E, we obtain that

$$\begin{aligned} P(B\bar{E}) &= R_{1,(s,t)}(G/Ch) - R_{1t}(G\backslash Ch);\\ P(\bar{B}E) &= R_{1,(s,t)}(G/Ch) - R_{1s}(G\backslash Ch);\\ P(BE) &= R_{1t}(G\backslash Ch) + R_{1s}(G\backslash Ch) - R_{1,(s,t)}(G/Ch). \end{aligned}$$

Thus we have that

$$R_{1i} = \Big[R_{1t}(G\backslash Ch,p){+}R_{1s}(G\backslash Ch,p){-}$$
$$R_{1,(s,t)}(G/Ch,p)\Big]\left(\prod_{j=s}^{i-1} M_{\lambda_j}(p){+}\right.$$
$$\left.\prod_{j=i}^{t-1} M_{\lambda_j}(p){-}\prod_{j=s}^{t-1} M_{\lambda_j}(p)\right){+}$$
$$\Big[R_{1,(s,t)}(G/Ch,p){-}R_{1t}(G\backslash Ch,p)\Big]\prod_{j=i}^{t-1} M_{\lambda_j}(p){+}$$
$$\Big[R_{1,(s,t)}(G/Ch,p){-}R_{1s}(G\backslash Ch,p)\Big]\prod_{j=s}^{i-1} M_{\lambda_j}(p)$$
$$= R_{1s}(G\backslash Ch,p)\prod_{j=i}^{t-1} M_{\lambda_j}(p) +$$
$$R_{1t}(G\backslash Ch,p)\prod_{j=s}^{i-1} M_{\lambda_j}(p) -$$
$$\Big[R_{1t}(G\backslash Ch,p) + R_{1s}(G\backslash Ch,p) -$$
$$R_{1,(s,t)}(G/Ch,p)\Big]\prod_{j=s}^{t-1} M_{\lambda_j}(p) = Y_i(G,p). \qquad (59)$$

Thus, for obtaining total income into $C(G,p)$ of all pairs $v_1 - v_i$ where v_i is a node of a chain we need to obtain $R_{1s}(G\backslash Ch,p)$, $R_{1t}(G\backslash Ch,p)$, and $R_{1,(s,t)}(G/Ch,p)$ only:

$$S(G,Ch,p) =$$
$$R_{1t}(G\backslash Ch,p)\sum_{i=s}^{t} w_i\left[\prod_{j=s}^{i-1} M_{\lambda_j}(p) - \prod_{j=s}^{t-1} M_{\lambda_j}(p)\right] +$$
$$R_{1s}(G\backslash Ch,p)\sum_{i=s}^{t-1} w_i\left[\prod_{j=i}^{t-1} M_{\lambda_j}(p) - \prod_{j=s}^{t-1} M_{\lambda_j}(p)\right] +$$
$$R_{1,(s,t)}(G/Ch,p)\prod_{j=s}^{t-1} M_{\lambda_j}(p)\sum_{i=s}^{t} w_i. \qquad (60)$$

Now let us consider the case when a c-node is an inner node of some chain. Let nodes of this chain be numbered from 1 to k and corresponding edges from 1 to $k-1$ and let v_s, $1 < s < k$ be a c-node.

For obtaining reliability of connection between c-node and some other node of a chain (terminal nodes included) we consider pseudo-cycle that consists of a chain Ch and pseudo-edge e_{1k} whose reliability polynomial is $R_{1k}(G\backslash Ch,p)$. Thus total income of all pairs $v_1 - v_i$, $i \in \overline{1,k}\backslash s$ into $C(G,p)$ is

$$X(G,Ch,p) = \sum_{i\in\overline{1,k}\backslash s} w_i\left[\prod_{j=i}^{s-1} M_{\lambda_j}(p) +\right.$$
$$\prod_{j=1}^{i-1} M_{\lambda_j}(p)\prod_{j=s}^{k-1} M_{\lambda_j}(p)R_{1k}(G\backslash Ch,p) -$$
$$\left.\prod_{j=1}^{k-1} M_{\lambda_j}(p)R_{1k}(G\backslash Ch,p)\right]. \qquad (61)$$

10 Obtaining some coefficients by analyzing initial graph's structure

10.1 ATR-polynomial

This case is well-known one. All $a_i = 0$ for $i > m-n+1$, a_{m-n+1} is a number of covering trees, and if there are at least k edge-independent paths between any pair of edges, then $\forall i{\in}[0,\ldots,k{-}1]\ a_i = C_m^i$. In particular, if there are no bridges in the graph, then $a_1 = m$.

10.2 EDP-polynomial

Let all edges fail. This means that all connections are broken and their total number is $n_m = C_n^2 = n(n-1)/2$.

If only one edge remains, then we have m variants of such event. In all these events only one connection exists and $C_n^2 - 1$ are broken, thus

$$n_{m-1} = m(C_n^2 - 1) = m[n(n-1)/2 - 1]. \qquad (62)$$

In the case of two remaining edges we need knowledge about nodes' degrees in G. Let D_2 be a set of nodes with degrees exceeding one. Then total number of chains with length two is

$$K_2 = \sum_{v\in D_2} C^2_{deg(v)}, \qquad (63)$$

and for each such case we have $C_{n-3}^2 + 3(n-3)$ broken pairwise connections, while for non-adjacent pair of edges — $C_{n-4}^2 + 4(n-4) + 4$. From this we obtain

$$n_{m-2}{=}K_2\Big[C_{n-3}^2{+}3(n{-}3)\Big]{+}$$
$$\big(C_m^2{-}K_2\big)\Big[C_{n-4}^2{+}4(n{-}4){+}4\Big]$$
$${=}K_2\big(C_n^2{-}3\big){+}$$
$$\big(C_m^2{-}K_2\big)\big(C_n^2{-}2\big) = C_m^2\big(C_n^2{-}2\big){-}K_2. \qquad (64)$$

Derivation of the next from the end coefficient (n_{m-3}) requires additional knowledge about structure of graph G, that means about existence and number of triangles (their finding has square complicity). Complete analysis of variants requires too much space, let us present final equations.

Let D_3 be a set of nodes with degrees not less than three. Let us denote a set of edges that connects nodes from D_2 as U^+ and consider a number of triangles T as known. In this case total number of chains with length three is equal to

$$K_3^{\sim}=\sum_{e_{ij}\in U^+}\big[(deg(v_i)-1)(deg(v_j)-1)\big]-3T=S_1-3T. \quad (65)$$

Using these information and considering cases of non-adjacent remaining edges and of chain with length two and separate edge, we obtain final

$$n_{m-3}=C_m^3(C_n^2-3)-K_2(m-2)+6T-S_1\ . \quad (66)$$

10.3 MENC-polynomial

If no edges fail, then all nodes are connected with a c-node, thus $C_0=n$.

If all edges fail, then a c-node is isolated and $C_m=1$.

If all edges but one fail, then a connected subgraph with 2 nodes occurs in the case of an edge adjacent to a c-node only (let v_1 be a c-node), and in $m-deg(v_1)$ cases c-node remains isolated. Thus $C_{m-1}=m-deg(v_1)+2deg(v_1)$.

If pair of edges remains only, then 3 nodes are in a connected subgraph in $C^2_{deg(v_1)}$ cases for sure, in $C^2_{m-deg(v_1)}$ cases c-node remains isolated, while for obtaining rest cases for 3 nodes and, correspondingly, number of 2-node connected subgraphs, we need find number of all 2-edge chains for which v_1 is a terminal node. This number may be easily found using square of an adjacency matrix. Thus obtaining C_{m-2} has complexity $O(n^2)$.

11 Examples

Let us consider graph in the Fig.1. Exhaustive search requires consideration of $2^9=512$ possible sugraphs.

11.1 ATR-polynomial

The case is simple one. Our preliminary knowledge is that r_0 and r_1 are 1 and 8, and that for $i>2$ all r_i are zeros. Obtaining r_2 (number of covering trees) using Kirchhoff theorem is possible also but is not necessary for such a small graph. Thus exhaustive search

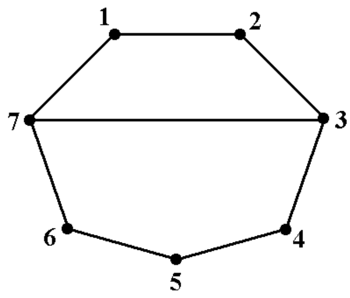

Figure 1: Test 7-node 8-edge graph

reduces to consideration of $C_8^2=28$ sugraphs. Using our technique gives two possible ways of calculations:

1. make factoring by the edge e_{25} and thus obtain with a probability $1-p$ a cycle of length 7, for which equation (39) may be applied and, with a probability p, 2 cycles of length 3 and 4, that are connected through a cutnode, and thus equation (43) is used;
2. make branching by one of two chains (3-4-5-6-7 or 7-1-2-3) using (47) thus reducing task to obtaining polynomials for a chain and cycle.

Both ways are similar in laboriousness. The resulting polynomial is

$$R(G,p)=p^8+8p^7(1-p)+19p^6(1-p).$$

11.2 EDP-polynomial

By analyzing graph's structure we can obtain that n_0 and n_1 are zeros, $n_8=C_7^2=21$, $n_7=8(C_7^2-1)=160$. For obtaining n_7 we find that all nodes have degrees 2 or more (2,2,3,2,2,2, and 3), thus $K_2=1+1+3+1+1+1+3=11$ and $S_1=15$, and $n_6=C_8^2(C_7^2-2)-11=521$. No triangles are in the graph's structure, thus

$$n_5=C_8^3(C_6^2-3)-11(8-2)-15=927.$$

Thus exhaustive search reduces to consideration of $\sum_{i=2}^4 C_8^i=154$ sugraphs. Let us use branching by a chain and choose chain (1-2-3) as pivot one. First we need obtain the probability of v_1 and v_3 been disconnected in $G\backslash Gh$. Several steps of serial-parallel reduction and using (13) gives

$$a_{13}=p^5(1-p)+9p^4(1-p)^2+16p^3(1-p)^3+14p^2(1-p)^4+6p(1-p)^5+(1-p)^6.$$

According to (51) we need to obtain EDP-polynomials for a 5-node cycle-shaped graph connected with the d-node v_1 and cycle-shaped graph with one double edge. Both graphs have fractional-polynomial weights of part of nodes.

1. *Cycle with a d-node (pivot chain is deleted).*

We need obtain new weights for terminal nodes of the chain (v_2^o and v_5^o). According to (55) and (56) we have that

$$w_1^o(p) = w_3^o(p) = 1 + \frac{p(1-p)}{1-p^2} = 1 + \frac{p}{1+p}.$$

Then, using (16) we obtain a cycle $G_1^o = (2-3-4-5-2)$ with new weight of v_7 equal to

$$w_7' = 1 + pw_1^o(p) = 1 + p + \frac{p^2}{1+p}.$$

While obtaining this cycle we obtain independent summand

$$\Delta = (1-p)(4+w_3^o(p))w_1^o(p) = \frac{5+11p-4p^2-12p^3}{(1+p)^2}.$$

By using (41) for the cycle we obtain EDP-polynomial:

$$N(G_1^o,p) = (1-p)^2(10p^5 + 31p^4 + 51p^3 + 51p^2 + 33p + 10)/(1+p), \quad (67)$$

thus total polynomial for a cycle with a d-node is

$$\frac{10p^8+21p^7+10p^6-21p^5-38p^4-35p^3+4p^2+34p+15}{(p+1)^2}.$$

2. *Cycle with a double edge (2 nodes are contracted by the pivot chain)*

First we obtain new weight of a joint node (let it keep number 3). According to (53), because of no e_{13} we have simply $w_3^*(p) = 3$. Now nodes v_3 and v_7 are connected by double edge with total reliability $2p-p^2$. Then we use (40) for obtaining EDP-polynomial of this cycle:

$$N(G^*,p) = 18 - 12p - 10p^2 - 10p^2 - 10p^3 - 10p^4 + 42p^5 - 18p^6. \quad (68)$$

By substituting all these equations into (51) and using (13) we obtain final

$$N(G,p) = 72p^6(1-p)^2+495p^5(1-p)^3+ 941p^4(1-p)^4+927p^3(1-p)^5+ 521p^2(1-p)^6+160p(1-p)^7+21(1-p)^8.$$

11.3 MENC-polynomial

For obtaining this polynomial we need or obtain all polynomials $R_{1,j}$, $i = 2,\ldots,7$, that means solving several similar tasks on initial graph, or use results of Section 9.3. Really, we can simplify obtaining sum of all polynomials after considering graphs $G\backslash Ch_1$, $G\backslash Ch_2$ and G/Ch_2, where Ch_1 is (3-2-1-7) and Ch_2 is (4-4-5-6-7). All needed polynomials are obtained trivially:

$$R_{37}(G\backslash Ch_1,p) = p^5+5p^4(1-p)+6p^3(1-p)^2+ 4p^2(1-p)^3 + p(1-p)^4; \quad (69)$$

$$R_{13}(G\backslash Ch_2,p) = p^4+4p^3(1-p)+2p^2(1-p)^2; \quad (70)$$

$$R_{17}(G\backslash Ch_2,p) = p^4+4p^3(1-p)+3p^2(1-p)^2 + p(1-p)^3; \quad (71)$$

$$R_{1,(37)}(G/Ch_2,p) = p^3(1-p)+2p^2(1-p) + p(1-p)^2. \quad (72)$$

Then, using (60) and (61) we obtain

$$X(G,Ch_1,p) = 2p+2p^2+2p^3-3p^4+p^5+ p^6-5p^7+3p^8; \quad (73)$$

$$S(G,Ch_2,p) = 2p^2+3p^3+4p^4+p^5-4p^6-8p^7+6p^8.$$

As all R_{1j} are included in these sums, we simply summarize them adding 1 as weight of v_1, and use (8) for equalizing powers of summands. Thus

$$C(G,p) = 7p^8+56p^7(1-p)+174p^6(1-p)^2+ 242p^5(1-p)^3+211p^4(1-p)^4+121p^3(1-p)^5+ 45p^2(1-p)^6+10p(1-p)^7+(1-p)^8.$$

Numerical experiments are out of scope of current paper. We can only mention that in [12] it was shown that our methodology allows more than 1000 times speeding up in obtaining ATR-polynomial in comparison with Maple 11.

Conclusion

In this paper we show that the same methodology may be applied to constructing algorithms for obtaining polynomials for different reliability indices of random graphs. Thorough examination of structural particularities may help in significant speeding up of calculations. Our future works concerns parallel realizations of our algorithms and consideration of reliability of multi-layer networks.

Acknowledgements: The authors wish to thank their colleagues and students for fruitful discussions.

References:

[1] Pietro Camarda. Bounds evaluation of coefficients in the reliability polynomial. *Microelectronics Reliability*, 30(6):1099 – 1110, 1990.

[2] Manoj Chari and Charles J. Colbourn. Reliability Polynomials: A Survey. *Journal of Combinatorics, Information & System Sciences*, 22:177–192, 1997.

[3] Charles J. Colbourn. *The Combinatorics of Network Reliability*. Oxford University Press, Inc., New York, NY, USA, 1987.

[4] Joanna A. Ellis-Monaghan and Criel Merino. Graph polynomials and their applications i: The tutte polynomial. In Matthias Dehmer, editor, *Structural Analysis of Complex Networks*, pages 219–255. Birkhüser Boston, 2011.

[5] Olga Gadyatskaya, Alexey S. Rodionov, and Olga K. Rodionova. Using edp-polynomials for network structure optimization. In Osvaldo Gervasi, Beniamino Murgante, Antonio Laganà, David Taniar, Youngsong Mun, and Marina L. Gavrilova, editors, *Computational Science and Its Applications - ICCSA 2008, International Conference, Perugia, Italy, June 30 - July 3, 2008, Proceedings, Part II*, volume 5073 of *Lecture Notes in Computer Science*, pages 1061–1076. Springer, 2008.

[6] Alexander K. Kelmans. Crossing properties of graph reliability functions. Technical Report 98–39, Rutgers University, New Brunswick, New Jersey, 1999.

[7] Alexander K. Kelmans. Crossing properties of graph reliability functions. *Journal of Combinatorics, Information & System Sciences*, 35(3):206–221, 2000.

[8] James Oxley and Dominic Welsh. Chromatic, flow and reliability polynomials: The complexity of their coefficients. *Comb. Probab. Comput.*, 11(4):403–426, July 2002.

[9] L.B. Page and J.E. Perry. A practical implementation of the factoring theorem for network reliability. *Reliability, IEEE Transactions on*, 37(3):259–267, Aug 1988.

[10] Alexey S. Rodionov and Olga K. Rodionova. Network probabilistic connectivity: Expectation of a number of disconnected pairs of nodes. In Michael Gerndt and Dieter Kranzlmüller, editors, *High Performance Computing and Communications*, volume 4208 of *Lecture Notes in Computer Science*, pages 101–109. Springer Berlin Heidelberg, 2006.

[11] Alexey S. Rodionov, Olga K. Rodionova, and Hyunseung Choo. On the expected value of a number of disconnected pairs of nodes in unreliable network. In Osvaldo Gervasi and MarinaL. Gavrilova, editors, *Computational Science and Its Applications – ICCSA 2007*, volume 4707 of *Lecture Notes in Computer Science*, pages 534–543. Springer Berlin Heidelberg, 2007.

[12] A.S. Rodionov. Speeding up computation of the reliability polynomial coefficients for a random graph. *Automation and Remote Control*, 72(7):1474–1486, 2011.

[13] Olga K. Rodionova, Alexey S. Rodionov, and Hyunseung Choo. Network probabilistic connectivity: Exact calculation with use of chains. In Marian Bubak, Geert Dick van Albada, Peter M. A. Sloot, and Jack Dongarra, editors, *Computational Science – ICCS 2004, 4th International Conference, Part I*, volume 3036 of *Lecture Notes in Computer Science*, pages 565–568. Springer Berlin Heidelberg, 2004.

[14] Olga K. Rodionova, Alexey S. Rodionov, and Hyunseung Choo. Network probabilistic connectivity: Optimal structures. In Antonio Laganà, Marina L. Gavrilova, Vipin Kumar, Youngsong Mun, C.J.Kenneth Tan, and Osvaldo Gervasi, editors, *Computational Science and Its Applications – ICCSA 2004*, volume 3046 of *Lecture Notes in Computer Science*, pages 431–440. Springer Berlin Heidelberg, 2004.

[15] Olga K. Rodionova, Alexey S. Rodionov, and Hyunseung Choo. Network probabilistic connectivity: Reliability polynomial. In *Proceedings of the 8th International Conference "Problems of Operation of Information Networks", Issyk-Kul, Kyrgys Republic, August 22–29, 2004. Proceedings, Part I*, pages 321–327, 2004.

[16] A. Satyanarayana and Mark K. Chang. Network reliability and the factoring theorem. *Networks*, 13(1):107–120, 1983.