SDP Algorithm for Network Reliability Evaluation

Petru Caşcaval, Sabina-Adriana Floria
Department of Computer Science and Engineering
"Gheorghe Asachi" Technical University of Iaşi
Dimitrie Mangeron Street, 27, 700050, Iaşi, Romania
cascaval@cs.tuiasi.ro, sabina.floria@cs.tuiasi.ro

***Abstract*—This paper addresses the issue of the two-terminal reliability evaluation of medium-to-large networks and proposes a new method to solve the problem of 'sum of disjoint products' (SDP) algebraically. This method uses a 'multiple variables inversion' (MVI) technique for transforming a structure function into a sum of disjoint products which has a one-to-one correspondence with the reliability expression. The results of our method for several network models are compared with those obtained by means of other well-known MVI techniques. For large-scale networks, our method offers a better solution.**

Keywords— two-terminal network reliability; SDP algorithm; minimal paths; minimal cuts; multiple variable inversion

I. INTRODUCTION

The network reliability theory is applied extensively in many real-word systems, which can be modeled as stochastic networks, such as computer and communications systems, distributed systems, networks of sensors, social networks etc. The reliability evaluation approaches use a variety of tools for system modeling and computation of reliability or availability indices that, in a certain way, describe the ability of a network to carry out a desired operation. Most tools are based on algorithms described in terms of minimal path set or minimal cut set (see, for example, [1]-[5]). Unfortunately, the problem of computing the network reliability based on the set of minimal paths or cuts is NP-hard [4], [6]. For this reason, in case of complex networks, other techniques for approximate reliability evaluation must be applied, such as those based on network decomposition or Monte Carlo simulation (see, for example, [7]-[10]).

In this work, we deal with the problem of exact evaluation of two-terminal reliability or availability indices in medium-to-large networks, and we propose an efficient SDP algorithm able to compute the two-terminal network reliability based on the minimal path set or the minimal cut set.

In order to reduce the computation burden, as in most such works, we assume that each node of the network is perfectly reliable. Because the failure of a node inhibits the work of all arcs connected to it, starting from the given network with unreliable nodes, an equivalent reliability model with perfect nodes but with links that have greater failure probabilities can be obtained [11, 12].

978-1-5090-5795-5/17/$31.00 ©2017 IEEE

For example, let us consider two adjacent nodes denoted by 1 and 2. Let p_1 and p_2 be the reliabilities of these nodes, and p_{12} be the reliability of the link between them. Fig. 1 presents two equivalent reliability models: one model with unreliable nodes, and another one with perfect nodes.

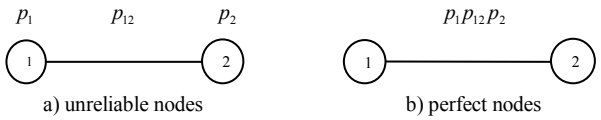


Fig. 1. Equivalent reliability models.

Moreover, to reduce the computation time for the minimal paths or cuts enumeration, once the source and destination nodes are known, the study can be simplified by artificially transforming the reliability graph into a directed one, as though only the communication from the source to the target node were of interest.

II. NOTATIONS AND PRELIMINARY CONSIDERATIONS

A. Nomenclature

a) *Reliability*. The two-terminal reliability of a stochastic network expresses the probability that there exists at least one path between any two specified nodes (let's say a source node and a target one), which operates successfully.

b) *Connected nodes*. Two nodes which can communicate with each other are connected, otherwise they are disconnected.

c) *Minimal path.* A minimal set of links (arcs) whose operation ensures that two given nodes are connected. For a minimal path, any proper subset is no longer a path.

d) *Minimal cut.* A set of links whose failure disconnects two given nodes. For a minimal cut, any proper subset is no longer a cut.

e) *Uniproduct.* A Boolean product composed only of distinct uncomplemented variables.

f) *Subproduct.* A part of a Boolean product which is a complemented or an uncomplemented uniproduct.

g) *Mixproduct.* A product of one uncomplemented subproduct and one or more complemented subproducts.

h) *Disjoint products.* A set of products expressing mutually exclusive states.

i) *Cube*. A vector of symbols representing a Boolean product.

B. Notations

a) $G(V, E)$ is a network model with node set $V = \{1, 2, \cdots, n\}$ and arc set $E = \{ x_1, x_2, \cdots, x_k \}$;

b) $s, t \in V, s \neq t$, are the source and target nodes;

c) p_x is the reliability of arc $x \in E$ and $q_x = 1 - p_x$;

d) R_{s-t} is the two-terminal reliability of network $G(V, E)$ with s and t the source and target nodes ($s-t$ network reliability).

C. Assumptions

a) Each node is perfect, as mentioned in section 1;

b) Each arc is either operational or failed, so a logical variable can be used to denote its state (the same notations $x_1, x_2, \cdots, x_k$ are used to denote these logical variables);

c) All failures that may affect the network under study are stochastically independent.

III. PROBLEM DESCRIPTION

Consider $G(V, E)$ the network under study and $s, t \in V, s \neq t$, the source and target nodes. For this model, consider the minimal path set $MPS = \{P_1, P_2, \ldots, P_m\}$. Note that, a minimal path $P_i \in MPS$ is expressed by a product of distinct logical variables associated with the arcs of this network, and the reliability of this path is given by the following equation:

$$Prob(P_i) = \prod_{c \in P_i} p_c \tag{1}$$

Starting from this minimal path set, a structure function $S = \bigcup_{i=1}^{m} P_i$ is defined, and the two-terminal network reliability of this model, R_{s-t}, is computed by applying the equation:

$$R_{s-t} = Prob(S) = Prob\left(\bigcup_{i=1}^{m} P_i \right). \tag{2}$$

When working with minimal cuts, let $MCS = \{C_1, C_2, \ldots, C_m\}$ be the cut set for the two-terminal network under study. The structure function in this case is $S = \bigcup_{i=1}^{m} C_i$, and the network reliability R_{s-t} is computed by applying the equation:

$$R_{s-t} = 1 - Prob(S) = 1 - Prob\left(\bigcup_{i=1}^{m} C_i \right). \tag{3}$$

Note that, for a minimal cut $C_i \in MCS$:

$$Prob(C_i) = \prod_{c \in P_i} q_c \ . \tag{4}$$

An efficient algorithm for enumerating all minimal paths and cuts of a graph is presented in [13]. Another method for the enumeration of minimal cuts in large network models and a comparative study for paths and cuts are presented in [14].

To compute the network reliability R_{s-t}, based on (2) or (3), the well-known rule of sum of disjoint products is recommended:

$$Prob\left(\bigcup_{i=1}^{m} A_i \right) = Prob(A_1) + Prob(\overline{A}_1 \cap A_2) + Prob(\overline{A}_1 \cap \overline{A}_2 \cap A_3) + \cdots + Prob(\overline{A}_1 \cap \overline{A}_2 \cap \cdots \cap \overline{A}_{m-1} \cap A_m) \ . \tag{5}$$

For this purpose, the structure function S must be transformed to an equivalent form S', composed only of disjoint products (DP), so that the network reliability R_{s-t} is computed by applying the equation:

$$R_{s-t} = Prob(S) = Prob\left(\bigcup_{i=1}^{m} P_i \right) =$$

$$= Prob(S') = Prob\left(\bigcup_{i=1}^{n} DP_i \right) = \sum_{i=1}^{n} Prob(DP_i). \tag{6}$$

Observe that (6) is much easier to compute than (2) or (3). In conclusion, the problem of computing the two-terminal network reliability essentially boils down to finding the disjoint products that define an equivalent structure function S'. Unfortunately, this task falls in the NP-hard category [4, 6, 12]. The space of solutions for this kind of functions is very large, so that the aim of an SDP algorithm is to identify the best possible solution for an equivalent function S'. That means that the number of disjoint products and the computing time for determining them should be as small as possible.

One of the best known SDP algorithms for expanding a structure function to another one, composed only of disjoint products, is given by Abraham [1]. Let us consider two undisjoint products, P and Q, and $X = \{x_1, x_2, \ldots, x_k\}$ be the set of logical variables included in P that do not belong to Q. Notice that P is an uniproduct whereas Q can also be a mixproduct. According to Abraham's theorem, the following logical expression can be written:

$$P + Q = P + \overline{x}_1 Q + x_1 \overline{x}_2 Q + x_1 x_2 \overline{x}_3 Q + \cdots + x_1 x_2 \cdots x_{k-1} \overline{x}_k Q \ . \tag{7}$$

To ensure that two products are disjoint, only a single complemented variable is added with each new term. Abraham's algorithm is a reference for the so-called 'single variable inversion' (SVI) algorithms. Remember that a product comprises a set of distinct variables, complemented or not. Two SVI products are disjoint if both contain at least one variable which reflects, in the two cases, complementary logical states (for example, x_1 and $\overline{x}_1$). Improved SVI methods are presented in [15] and [16].

To reduce the computation burden, other approaches based on the so-called MVI technique have been devised (see, for example, [2], [3], [17], and [18]). An excellent survey on MVI methods can be found in [12]. A new SDP algorithm based on an MVI technique is presented in the following section.

IV. A NEW MVI METHOD (NMVI)

A. Preliminaries

The method we propose uses the following laws from Boolean algebra:

- $x + \overline{x} = 1$ (Complementation law);
- $x + xy = x$ (Absorption law);
- $x + \overline{x}y = x + y$ (Idempotent law);
- $\overline{xy} = \overline{x} + x\overline{y} = \overline{x} + \overline{y}$ (De Morgan's law).

When an MVI technique is applied, a product may contain distinct logical variables (complemented or uncomplemented) but also one or more complemented groups of logical variables (complemented subproducts). For instance, a Boolean expression of six variables representing link 1 or 4 in the failed state, link 5 operational, link 6 in the failed state and links 2 and 3 in a don't care state is represented by $\overline{x_1 x_4} x_5 \overline{x_6}$. Note that in an SVI approach, the same network state is represented by the Boolean expression $x_1\overline{x_4}x_5\overline{x_6} + \overline{x_1}x_4x_5\overline{x_6}$. The advantage of an MVI approach is obvious.

To describe a product, we use a vector of length k (also called a cube) with the following meanings:

- an uncomplemented variable is indicated in cube by the symbol '1';
- a complemented variable is indicated by '-1';
- the absence of a variable is indicated by the symbol '–';
- a complemented subproduct is indicated by using a negative number smaller or equal to '-2', as illustrated in TABLE I.

To illustrate this new method, let us consider two undisjoint products, P and Q, for which the absorption law is not applicable (i.e., $P \not\subset Q$). Note that, to verify that $P \not\subset Q$ it is necessary to check that at least one variable from P (indicated in cube by '1') does not belong to Q in the same form (indicated in cube by any symbol $\neq$ '1'). Also, P and Q are two undisjoint products if there is no complemented subproduct in Q (indicated in cube by '-1', '-2', '-3', etc.) that also belongs to P in an uncomplemented form (indicated in cube by '1' in the same positions). Examples of two undisjoint products are presented in TABLE II. As illustrated in this table, three cases must be taken into account.

In order to expand a product Q in relation to a given uniproduct P, so that any new generated product to be disjoint with P, we propose the following two rules.

Rule 1. Let $X = \{x_1, x_2, \ldots, x_k\}$ be the set of logical variables included in P that do not belong to Q.

TABLE I. PRODUCT DESCRIPTION

Product	**Cube**										**Comment**
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	
$x_3x_5x_8$	–	–	1	–	1	–	–	1	–	–	Uniproduct
$\overline{x}_1x_3\overline{x}_5x_7x_9$	-1	–	1	–	-1	–	1	–	1	–	SVI product
$\overline{x_1}\,\overline{x_2x_4}x_5\overline{x_6}\,\overline{x_7x_{10}}$	-1	-2	-	-2	-2	1	-3	–	–	-3	MVI product

TABLE II. EXAMPLES OF UNDISJOINT PRODUCTS

Two undisjoint products (P and Q)	**Cube**										**Comment**
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	
$P = x_2x_5x_8$	–	1	–	–	1	–	–	1	–	–	Case 1
$Q = x_3x_4x_5x_7$	–	–	1	1	1	–	1	–	–	–	
$P = x_1x_9$	–	1	–	–	–	–	–	–	1	–	Case 2
$Q = \overline{x_2x_5}\;\overline{x_8x_9x_{10}}$	–	-2	–	–	-2	–	–	-3	-3	1	
$P = x_1x_3x_7x_8$	1	–	1	–	–	–	1	1	–	–	Case 3
$Q = \overline{x_2x_3x_5}\;\overline{x_6x_7x_8x_{10}}$	–	1	-2	–	-2	-3	-3	-3	–	1	

The following logical expression can be written based on the complementation law:

$$P+Q=P+Q(x_1x_2\cdots x_k+\overline{x_1x_2\cdots x_k})$$

$$=P+x_1x_2\cdots x_kQ+\overline{x_1x_2\cdots x_k}Q. \quad (8)$$

As follows, we will refer to the rule presented in (8) as the "type I expansion". When P and Q are both uniproducts (case 1 in TABLE II), the new product $x_1x_2\cdots x_k\,Q$ includes all the variables from P so that the absorption law is applicable. A reduced logical expression composed of two disjoint products is obtained:

$$P+Q=P+\overline{x_1x_2\cdots x_k}\,Q. \quad (9)$$

Otherwise, the new product $x_1x_2\cdots x_k\,Q$ must be expanded again until all the new generated products are disjoint with P and also between them.

Rule 2. Consider again two products, P and Q, in which $P=x_1x_2\cdots x_k\,R_1$, and $Q=\overline{x_1x_2\cdots x_kx_{k+1}\cdots x_m}R_2$. By applying the Boolean rule $\overline{xy}=\bar{x}+x\bar{y}$, the following logical expression can be written:

$$P+Q=P+\overline{x_1x_2\cdots x_kx_{k+1}\cdots x_m}R_2$$

$$=P+(\overline{x_1x_2\cdots x_k}+x_1x_2\cdots x_k\,\overline{x_{k+1}\cdots x_m})R_2$$

$$=P+\overline{x_1x_2\cdots x_k}R_2+x_1x_2\cdots x_k\,\overline{x_{k+1}\cdots x_m}R_2 \quad (10)$$

We will refer to the rule presented in (10) as the "type II expansion". Observe that, when $R_1\in R_2$ the product $x_1x_2\cdots x_k\,\overline{x_{k+1}\cdots x_m}R_2$ is absorbed by product P, so that a reduced logical expression composed of two disjoint products is obtained:

$$P+Q=P+\overline{x_1x_2\cdots x_k}R_2 \quad (11)$$

When $R_1\notin R_2$, the new mixproduct $x_1x_2\cdots x_k\,\overline{x_{k+1}\cdots x_m}R_2$ must be expanded again until all the new generated products are disjoint with P and also between them.

Remark 1. Based on (9) and (11), during the process of generating equivalent disjoint products, the absorption law is applicable in fewer cases. This is the main idea we have had in view for accelerating this time-consuming process.

Remark 2. When both type I and type II expansion rules are applicable (case 3 in TABLE II), we propose that the type II expansion rule be applied prior to the type I expansion.

The method we propose implies the following steps:

- The initial products (uniproducts) are sorted by length in ascending order;
- The first term is left unchanged;
- The second term is expanded with respect to the variables that appear only in the first one, by applying one or both rules previously defined (type I or type II expansion), as applicable.
- The operation is repeated for all other initial terms so that all the newly generated products are disjoint from each other and are also disjoint with all the terms already obtained in the previous steps.

For an efficient implementation of this method, we use three arrays, namely:

- the first one with the initial terms (uniproducts);
- a working array to store and process the terms generated by expanding a certain uniproduct;
- a collector array to store the final terms (disjoint products).

This method is illustrated by the following example. Take a set of uniproducts $MPS=\{adk, bfl, cek, adgl, cegl\}$. The process of expanding these terms by applying this method until all the products are disjoint from each other is presented in Fig. 2. Finally, the set of disjoint products obtained by this method is $DPS=\{adk, bfl\,\overline{adk}, cek\overline{ad}\,\overline{bfl}, adgl\,\bar{k}\,\overline{bf}, cegl\overline{ad}\,\bar{k}\,\overline{bf}\}$. Based on this set of disjoint products, by applying (6), the following two-terminal reliability expression is obtained:

$$R_{s-t}=p_ap_dp_k+p_bp_fp_l(1-p_ap_dp_k)+$$
$$p_cp_ep_k(1-p_ap_d)(1-p_bp_fp_l)+$$
$$p_ap_dp_gp_lq_k(1-p_bp_f)+$$
$$p_cp_ep_gp_l(1-p_ap_d)\,q_k(1-p_bp_f)\ . \quad (12)$$

V. COMPARATIVE STUDIES

A. Comparison with Abraham's method

To demonstrate the correctness of the new method we propose comparative results related to two-terminal network reliability evaluation obtained by applying NMVI and the SVI method given by Abraham [1] are presented in this section. On the other hand, the aim of this first comparative study is to illustrate the ability of the new MVI method to anticipate well enough the terms for which the absorption law could be applicable (see *Remark* 1), and to neglect them in order to reduce the number of terms that must be processed.

For this comparative study, four medium-to-large stochastic network models have been considered. To check the implementation of the two methods, the numerical results for the case with all the links having the same reliability (e.g., $p=0.95$) are computed. The comparative results are presented in TABLE III.

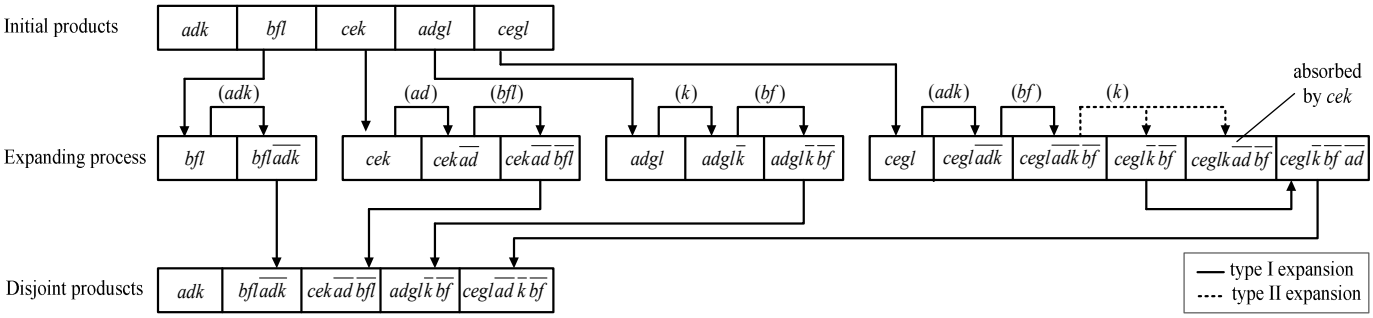
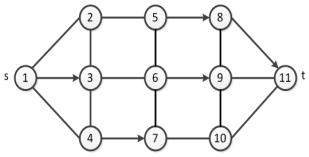
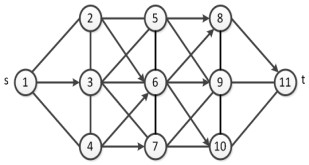
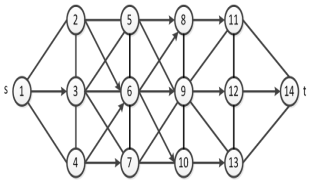
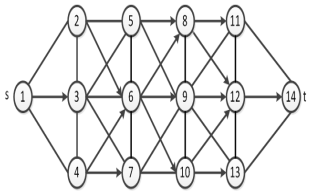


Fig. 2. Example of applying the method NMVI.

TABLE III. COMPARATIVE RESULTS: ABRAHAM'S METHOD VERSUS NMVI

Network model	Number of minimal paths	Applied method	Number of DPs	Processed terms	Absorption cases	R_{s-t} ($p = 0.95$)
	87	Abraham	2012	5472	675	0. 9994239
		NMVI	545	3458	546	
	993	Abraham	39395	174728	35687	0. 9997488
		NMVI	15502	136043	20384	
	5543	Abraham	897373	3961978	941589	0. 9997475
		NMVI	239101	2399387	456521	
	8716	Abraham	1835413	8057515	1817042	0. 9997488
		NMVI	574274	5318472	894473	

As shown in TABLE III, NMVI gives better solutions with fewer disjoint products. Also, the number of processed terms decreases significantly. Observe that the absorption rule is applicable in fewer cases (for the large networks, to less than half). This is an important merit for NMVI.

B. Comparison with other MVI methods

In this section our method is compared with other well-known MVI methods. For this comparative study the following MVI methods are considered:

- the method given by Veeraraghavan and Trivedi [3] (VT in this paper);

- CAREL method (PII option) given by Soh and Rai [2];
- KDH88 algorithm given by Heidtmann [17];
- The hybrid method (HM) given by Chaturvedi and Misra [19] that combines the best features of KDH88 and CAREL;
- NMVI.

The comparison criterion is the number of disjoint products generated, able to cover all the cases in which the two given nodes (*s* and *t*) are connected.

The network models we have considered for this comparative study are presented in Fig. 3. This selection includes network models typically used for comparing the efficiency of the methods dedicated to the problem of two-terminal network reliability evaluation. More exactly, we have considered those network models which are presented in at least two articles dedicated to the MVI techniques ([2], [3], [17] or [19]). Note that, another MVI algorithm dedicated to coherent-system reliability is presented in [18], but the paper does not report the number of disjoint products generated for the evaluated cases.

The comparative results are presented in TABLE IV. As shown in this table, compared to VT, CAREL or KDH88, NMVI gives better solutions with fewer disjoint products. When comparing NMVI with HM, their efficiency is found to be quite close.

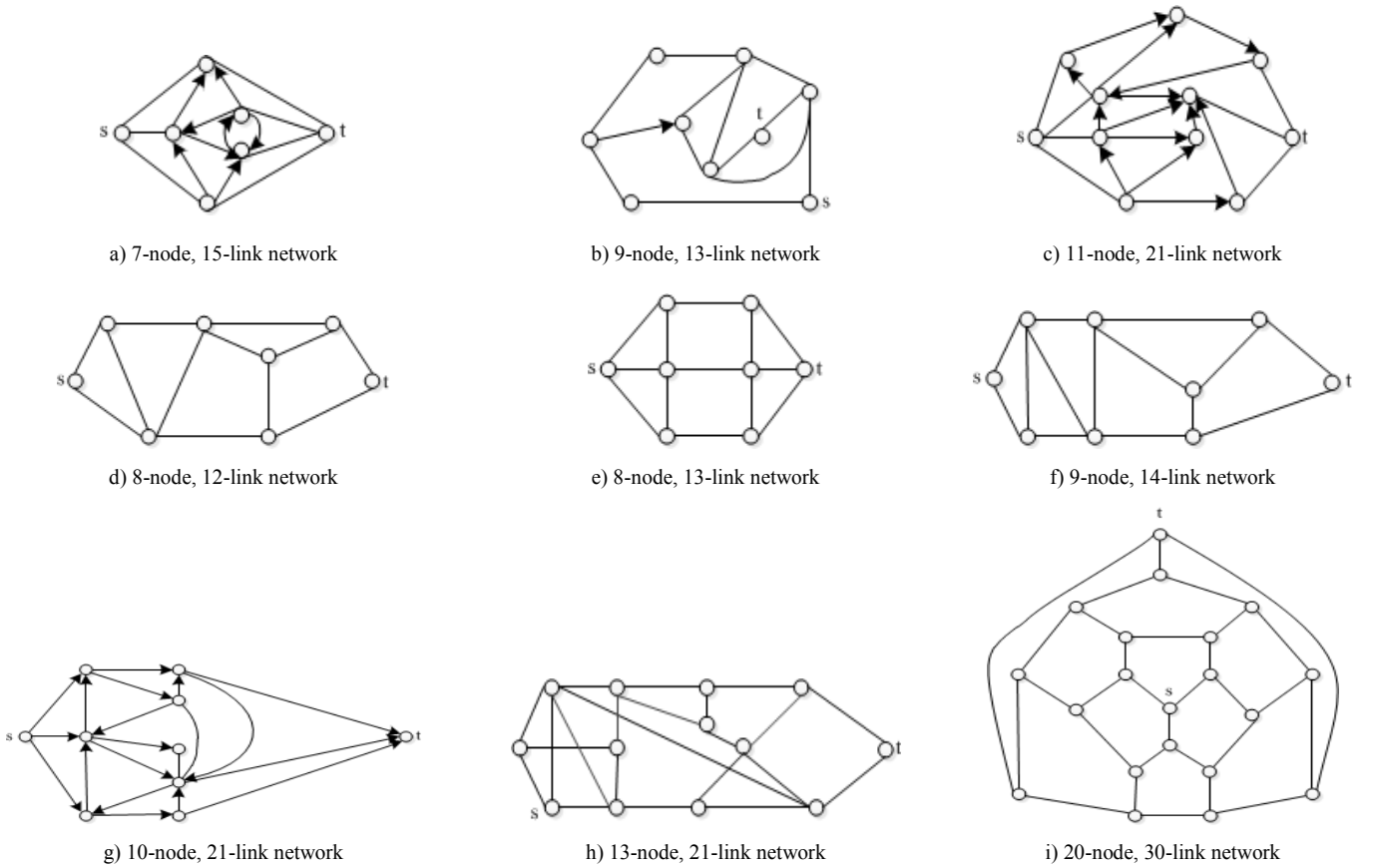
VI. CONCLUSIONS

In this paper, the process of algebraically extracting the disjoint products from a sum of products is discussed, and a new MVI method (NMVI) is proposed. In comparison with other well-known MVI methods (VT, CAREL or KDH88), NMVI gives better solutions with fewer disjoint products. At the same time, NMVI seems to be comparable with the hybrid method HM.

In case of complex networks, two or more such methods should be considered to obtain the best results. The hybrid methods that combine the best features of different MVI techniques must be taken into account.

Unfortunately, all the algorithms for exact reliability evaluation in network models fall in the NP-hard category, being difficult to apply for large networks, such as social networks. In these cases, other techniques for approximate evaluation can be applied, especially the Monte Carlo simulation (see, for example, [20]-[22]). Even so, the SDP algorithms for exact evaluation of two-terminal network reliability are still necessary for the validation of simulation programs.

Another approach for this SDP problem is based on binary decision diagrams (BDDs), as presented in [12], [23] [24]. This is a subject for a future work.



a) 7-node, 15-link network b) 9-node, 13-link network c) 11-node, 21-link network

d) 8-node, 12-link network e) 8-node, 13-link network f) 9-node, 14-link network

g) 10-node, 21-link network h) 13-node, 21-link network i) 20-node, 30-link network

Fig. 3. Network models for a comparative study of the MVI methods.

TABLE IV. COMPARATIVE RESULTS FOR SOME MVI METHODS

Network models presented in Fig. 3	Number of minimal paths	Number of disjoint products				
		VT [3]	CAREL [2]	KDH88 [17]	HM [19]	NMVI
a) 7-node, 15-link	14	–	23	27	–	23
b) 9-node, 13-link	18	27	30	31	–	25
c) 11-node, 21-link	18	82	94	101	–	82
d) 8-node, 12-link	24	41	39	41	38	38
e) 8-node, 13-link	29	77	76	75	77	77
f) 9-node, 14-link	44	90	82	87	80	82
g) 10-node, 21-link	64	305	309	–	–	298
h) 13-node, 21-link	281	–	2491	–	2302	2269
i) 20-node, 30-link	780	–	54032	–	46707	48696

REFERENCES

[1] J.A. Abraham, "An improved algorithm for network reliability", IEEE Trans. Reliability, vol. 28, pp. 58-61, April 1979.

[2] S. Soh, S. Rai, "Carel: Computer aided reliability evaluator for distributed computing networks", IEEE Trans. Parallel and Distributed Systems, vol 2, No. 2, pp. 199-213, April 1991.

[3] M. Veeraraghavan, K.S. Trivedi, "An improved algorithm for sysmbolic reliability analysis", IEEE Trans. Reliability, vol. 40, pp. 347-358, August 1991.

[4] J.S. Proven, M.O. Ball, "Computing network reliability in time polynomial in the number of cuts", Operations Research, vol. 32, pp. 516–526, 1984.

[5] M.O. Ball, J. S. Proven, "Disjoint products and efficient computation of reliability", Operations Research, vol. 36, pp. 703-715, 1988.

[6] K. S. Trivedi, Probability and Statistics with Reliability, Queueing and Computer Science Applications, John Wiley & Sons, New York, 2002.

[7] F. Beichelt, L. Spross, "An efficient method for reliability analysis of complex systems", J. Information Processing and Cybernetics EIK, vol. 23, pp. 227-235, 1987.

[8] P. Caşcaval, B.F. Romanescu, "Complementary Approaches for the Network Reliability Evaluation: Network Decomposition and Monte Carlo Simulation", Bul. Inst. Polit. Iaşi, Tomul L (LIV), Fasc. 1-4, Automatică şi Calculatoare, pp. 123-131, 2004.

[9] P. Caşcaval, A.R. Macovei, "Reliability Evaluation by Network Decomposition", Bul. Inst. Polit. Iaşi, Tomul XLIX (LIII), Fasc. 1-4, Automatică şi Calculatoare, pp. 56-65, 2003.

[10] P. Caşcaval, B.A. Botez, "Recursive Algorithm for Two-Terminal Network Reliability Evaluation", Bul. Inst. Polit. Iasi, LI (LV), Fasc.1-4, Automatică şi Calculatoare, pp. 137-146, 2005.

[11] M. Shooman, Reliability of computer systems and networks, John Wiley & Sons, New York, 2002, pp. 297-301.

[12] S.K. Chaturvedi, Network Reliability: Measures and Evaluation, Scrivener Publishing - Wiley, New Jersey, 2016.

[13] Y. Shen, "A new simple algorithm for enumerating all minimal paths and cuts of a graph", Microelectronics Reliability, vol. 35 (6), pp. 973–976, 1995.

[14] R. Mishra, M.A. Saifi, S.K. Chaturvedi, "Enumeration of minimal cutsets for directed networks with comparative reliability study for paths or cuts", Quality and Reliability Engineering International, vol. 32, pp. 555–565, February 2016.

[15] F. Beichelt, L. Spross, "An improved Abraham-method for generating disjoint products", IEEE Trans. Reliability, vol. 36, pp. 70-74, April 1987.

[16] M. O. Locks, "A minimizing algorithm for sum of disjoint products", IEEE Trans. Reliability, vol. 36, pp. 445-453, October 1987.

[17] K. Heidtmann, "Smaller sums of disjoint products by subproduct inversion", IEEE Trans. Reliability, vol. 38, No. 3, pp. 305-311, August 1989.

[18] T. Luo, K.S. Trivedi, "An improved algorithm for coherent-system reliability", IEEE Trans. Reliability, vol. 47, pp. 73-78, March 1998.

[19] S.K. Chaturvedi, K.B. Misra, "A hybrid method to evaluate reliability of complex networks", International Journal of Quality & Reliability Management, vol. 19, pp. 1098–1112, December 2002.

[20] K.F. Tee, L.R. Khan, H. Li, "Application of subset simulation in reliability estimation of underground pipelines", Reliability Engineering & System Safety, vol. 130, pp. 125–131, October 2014.

[21] K. Zuev, S. Wu, J. Beck, "General network reliability problem and its efficient solution by subset simulation", Probabilistic Engineering Mechanics, vol. 40, pp. 25–35, April 2015.

[22] H.-S. Li, Y.-Z. Ma, Z. Cao, "A generalized subset simulation approach for estimating small failure probabilities of multiple stochastic responses", Computers & Structures, vol. 153, pp. 239–251, June 2015.

[23] X. Zang, H.-R. Sun, and K.S. Trivedi, "A BDD approach to dependability analysis of distributed computer systems with imperfect coverage", Dependable Network Computing, D. R. Avresky (ed.), Kluwer Academic Publishers, Amsterdam, December 1999, pp. 167-190.

[24] A. Birolini, Reliability engineering. Theory and practice, Springer-Verlag, Berlin Heidelberg, 2014, pp. 57-60.