An Elementary Tree Transformation Based Approach for Reliability Estimation of Interconnection Networks

Pradyumna Kumar Tripathy
Department of Computer Science and Engineering
Silicon Institute of Technology, Bhubaneswar , Odisha, India

Ipsit Hota
DBPMU Komna, NHM
Nuapada, Odisha, India

Ranjan Kumar Dash
Dept. of Information Technology,
College of Engineering and Technology, Bhubaneswar, Odisha

Chitta Ranjan Tripathy
Vice Chancellor,
Sambalpur University, Burla, Odisha

Abstract- Interconnection network plays an important role in Computer networks, distributed access networks, Communication networks etc. The factors that affect the design of ICN are Reliability, Cost, Flow, Power etc. Reliability evaluation of an Interconnection network is a NP-hard problem. A small change in layout requires the repetition of the complete procedure. We proposed a simple and efficient algorithm which exactly estimates the network reliability of an Interconnection network. The algorithm is suitable for estimating network reliability of both regular and general networks with homogeneous or non-homogeneous link capacity. The algorithm generates all the disjoint terms while enumerating the spanning trees of the Interconnection Networks. The overhead time for disjointing process can be eliminated and the non-redundant disjoint terms are produced which leads to generate the reliability expression.

Keywords – Interconnection networks, Reliability, Elementary Tree Transformation, Computation Tree, Spanning Tree

I. NOTATIONS

G	*Equivalent Probabilistic Graph of the input network.*
T	*A spanning tree of the graph G*
ST	*Set for holding the intermediate generated spanning trees of graph G*
Chord_set	*Set containing all the intermediate generated chords of a graph G*
Edge_set	*Set of edges that involves in a cycle formed by adding a chord to a spanning tree of the graph.*
Node_set	*Set of nodes that involves in a cycle formed by adding a chord to a spanning tree of the graph.*
C_j	*Chord at* j^{th} *position in Chord_set*
E	*Consists of edges which are not included in any spanning tree* ST_i *or in the spanning trees derived from it.*
G'	*Graph generated by adding a chord* C_j *to a spanning tree T*
N_k	*An edge that is present at* k^{th} *position in Edge_set*
S	$ST_i \cup E_i$
NR	*Network Reliability*
|Chord_Set|	*Number of elements in Chort_set*
|ST|	*Number of elements in ST set *

|Node_set| *Number of elements in Node_set*
Add_nonredundant(ST,T) *Adds the spanning tree T in ST set if it is not present in ST set earlier.*
Find_chord(G, ST_i, E_i) *Find all the edges present in G but not in S.*
Check_cycle(G', C_j) *Returns the edges except the chord C_j that creates a cycle in G'.*
Join(ST_i, C_j) *Adds the link C_j to the spanning tree ST_i*
Make_set(E_i, N_k) *Adds the edge E_i with N_k*

II. INTRODUCTION

Modeling is used to validate the designs and evaluating the performance of many real-world complex systems, such as computer communication system [1], human system [2], power transmission and distribution system [3], transportation system and traffic management [4]. In an interconnection network, reliability is an important index. Network reliability is defined as the probability that all the nodes of a network are connected. The correct evaluation of reliability is a NP-hard problem. Therefore it has attracted many researchers for estimation of the exact reliability interconnection networks.

The theory of network Reliability originates from a series of lectures given by Von Neumann in 1952[5]. In general reliability evaluation involves three NP hard problems. They are the problems of searching for all minimal path or minimal cuts; the problems of searching for all lower boundary points (LBPs) and upper boundary points(UBPs).or namely d-mp /d-mc and the problem of calculating union probability. Searching for MP is popular in the recent literature: symbolic augmentation-based [6], expression-based [7], and direct search-based [8] algorithms. The symbolic expression-based algorithms define the symbolic terms and operations, and develop their algebraic manipulation to produce the MP. However, sometime it generates a large number of intermediate terms which are not necessarily all to be the final MP. Such approvals normally causes inefficient search in both computational storage and time. The augmentation based algorithm is a method to generate the MP from a known simple two terminal network. Adding arc by arc, the new generated MP's incurred by the augmented arc are collected until completing of the target network. When the size of the target network becomes large, it is quiet impractical to do such an augmentation. Colbourn's algorithm [9] has been reported having a good efficiency of $0(n\pi)$, where π is the no. of min-path. Because of most of the direct search based algorithm have very computational overheads in duplicated verification of min-path; the most recent chen's algorithm [10] has relieved much of the overhead by means of backtracking. There are several approaches and methodologies [11] used for computing two-terminal reliability (2TR), Assuming two states(operational and failed) for the system and its components .In 2TR However, there are a number of realistic systems with their components having more than two states, and thus the binary state approaches cannot properly compute the system reliability [12-13]. Hence, the multi-state two-terminal reliability (M2TR) problem has recently attracted the significant attention [14-15]. In 2TR analyses, These are the three important measures to access the performance of systems represented by probabilistic graph: g-terminal reliability,2-terminal reliability, and k-terminal reliability. The g-terminal reliability is the probability that every node in the network is able to communicate with each other. The 2-terminal reliability is the probability that a communication exist between a specified pair of nodes in network. The k-terminal reliability ensures that a specified set of k-nodes of the network are able to communicate with each other. it is of interest to calculate the reliability associated with the existence of connecting paths between two specific network nodes, usually known as the source and the sink. Thus, in more general terms, one is interested in obtaining the probability so that the source and the sink can communicate. There are many operational networks that follow the two-terminal rationale, for example: hard-wired and wireless telecommunications, computer networks, electric power distribution, and circuits with in electronic devices, to name just a few. For these networked systems, reliability occupies an important role since it can be used as a tool to assess performance, to make maintenance decisions, and to prioritize system improvements (Ramirez-Marquez and Coit, 2005)[16] during both the development and the operational phase of the network. Minimal cut sets (MCSs) have been developed from elementary modes (EMs) [17, 18], a metabolic pathway analysis (MPA) [19] method that uses convex analysis [20] to identify all possible and feasible metabolic routes for a given network at steady state. A review of the history of EMs can be seen in [21]. This review focuses on MCSs which, together with EMs, form dual representations of metabolic networks with both being able to be converted into each other [22]. MCSs can be considered the smallest "failure modes" in a system; they were first introduced in 2004 by S. Klamt and Gilles [23], motivated by their desire to gain deeper insight into the functionality and capability of an organism by further analyzing the structure of its metabolic network. In particular, they looked at how potential failure modes in a metabolic network could render the network structurally incapable of performing certain functions. Chen and Yuang access two categories of partition techniques for computing 2-terminal pair reliability (path-based, and cut-based algorithm), and concluded that for most benchmark networks, cut-based algorithm are superior to the path-set based algorithms with respect to computation time. A new model that estimates the reliability by summing the linear and quadratic unreliability of each minimal cut-set was proposed in [24]. Recently Goyal et al. [25] proposed a new approach known as source node exclusion method (SNEM). However, it does not produce a reliability expression. Path-set enumeration (a prerequisite to SDP) techniques have been devised by many researchers such as Jasmon & Foong[26] proposed an approach directly provides minimal cut-sets without the use of minimal path-sets or inversion techniques. But they could not eliminate the generation of redundant terms, and therefore require additional calculation to generate a minimal solution. Recently Gerbe et al. [27] developed a technique, based on a predecessor matrix needs the network to have directed edges, and its size increases with the number of links in the network. The MVI-SDP based method was first proposed in [28]. Further improvement and simplification of this approach was Soh & Rai. Besides, the impact of pre-processing i.e., ordering of path-sets/cut-sets to obtain a compact reliability expressions and results on various ordering schemes. However, When the sub algorithms are implemented together to develop a general algorithm the computational effort to generate the minimal cut-sets in complex networks is significantly reduced.

A new approach is proposed in this paper which not only generates all the spanning trees of a network but also performs a disjoint operation simultaneously to generate all reliability terms and its corresponding self complementary unreliability terms. The process does not suffer from any additional overhead of disjointing process. The proposed method guarantees for non-redundant enumeration of disjoint terms. The proposed method is suitable both for regular and general networks.

The section-III contains the proposed algorithm. The proposed algorithm is supported by an illustration in section IV. Section V describes the applicability of the proposed method on various bench mark interconnection networks. Section VI concludes the paper with a scope for future work.

III. PROPOSED ALGORITHM

Definitations:

1. *Network Reliability:* Network reliability is defined as the probability that all the nodes of a network are connected.
2. *Spanning Tree:* A spanning tree is a acyclic, connected subset of Graph G, which has all the vertices covered with minimum possible number of edges.
3. *Computation Tree:* It is a propositional branching time logic tree, permitting explicit quantification over all possible futures.

The algorithm uses a search tree technique to construct a computation tree. The computation tree can be used to output all spanning trees by relative changes between spanning trees rather than the entire spanning trees themselves. The algorithm starts by taking the initial spanning tree. The process is based on elementary tree transformation. The non-tree edges also called as chords are exchanged with the branches of the initial spanning tree to produce all spanning trees of the graph. This computation can be represented by a computation tree with the initial spanning tree as its root. To generate all the spanning trees the children of the root node is expanded as like as root. To overcome the repetition of same spanning tree each spanning tree has a set of edges that will not include in that spanning tree as well as in its children spanning trees. The spanning trees so generated are self disjoint and the eliminated edge sets gives the unreliability terms from which the reliability expression for the given Interconnection network can be computed. The algorithm to evaluate the reliability of the Interconnection Network is presented as follows.

Estimate_NetworkReliability ( )
Input: G
Output: NR
Begin

ST $\leftarrow \Phi$

Generate an initial spanning tree T

Add_nonredundant(ST , T)

$E_1 \leftarrow \Phi$

For each $ST_i \quad \forall i = 1$ *to* $|ST|$

```
{    Chord_set ← Find_chord(G , S )
     For each Cj ∈ Chord_set    ∀ j=1 to| Chord_set|
      {
       G' ← Join (STi , Cj )
       Edge_set ← Check_cycle( G' , Cj )
       For each Nk ∈ Edge_set   ∀ k=1 to | Node_set|
         {
         T ← G' ~ Nk
         Add_nonredundant(ST, T )
         E ← Make_set( Ei , Nk )
         }
       }
  }
End
```

IV. ILLUSTRATION

Consider the network shown in Fig-I having 4 number of nodes and 5 number of links. G is the equivalent probabilistic graph of the inputted Interconnection Network. The following steps illustrate the enumeration of all the self disjoint spanning trees using the proposed algorithm.

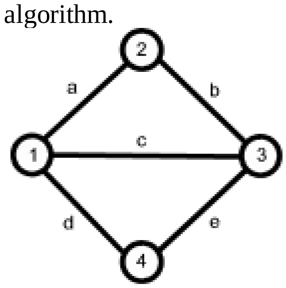


Fig: I- An Example Interconnection Network

Let the initial spanning tree of graph G is generated as “abd” as shown in Fig- II. Now T= “abd”.

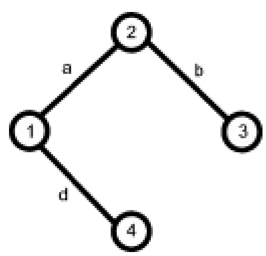


Fig: II Initial spanning tree G

Add_nonredundant (ST, T) function checks whether the spanning tree passed in the argument is present in ST set or not. If it is not present then it adds the current spanning tree into ST set. Here in our case ST set is empty, so it adds T to ST and the E_1 is initialized to Φ.

Now |ST| is 1, ST_1 = “abd” and E_1= “Φ”.

Find_chord(G,ST₁,E₁) will find out all the edges that is present in G but not present in S (i.e $ST_i \cup E_i$) and stores them in Chord_set.

Now the Chord_set = [c, e].

Here C_1 = ‘c’ and it calls *Join* (ST_1,C_1) function. This function simply adds the chord C_1 to the spanning tree ST_1.. The resultant graph is G’. In this case G’ is shown in Fig-III.

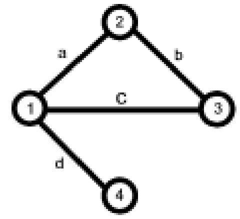


Fig:III. Graph obtained by adding the chord ‘c’ to the spanning tree T

Adding chord ‘c’ to the spanning tree ‘T’ generates a cycle in the resulting graph G’. The function *Check_cycle (G’,C_1)* returns the edges except the edge present in C_1 that creates a cycle in the graph G’. After the *Check_cycle( )* function is executed the Edge_set=[a, b], here ***a*** and b edges contributes for formation of a cycle.

After getting the edges that contributes for formation of a cycle, a new spanning tree can be generated by deleting an edge.

G’ ~ N_k produces the edges that are present in G’ but not present in N_k.

In the example G’ = {a, b, c, d} and N_1 = {a} .

So T ← G’ ~ N_1 produces {b, c, d} which leads to form a new spanning tree T.

The *Add_nonredundant (ST,T)* checks whether the generated spanning tree T is present in ST , if it is not then it adds T to ST set. Here the new spanning tree T will be added to ST and then it’s corresponding E set is updated using the

Make_set (E_1, N_1) function. *Make_set ( )* simply add the edges present in its two arguments.

Here the E set will be updated as E_2 = *Make_set(E_1 , N_1)* which leads to E_2={a} .

In the similar way all the spanning trees ST_i of G can be generated. The Fig :IV represents how 6 number of spanning trees of the inputted graph G (Fig : I) have been enumerated from the initial spanning tree T.

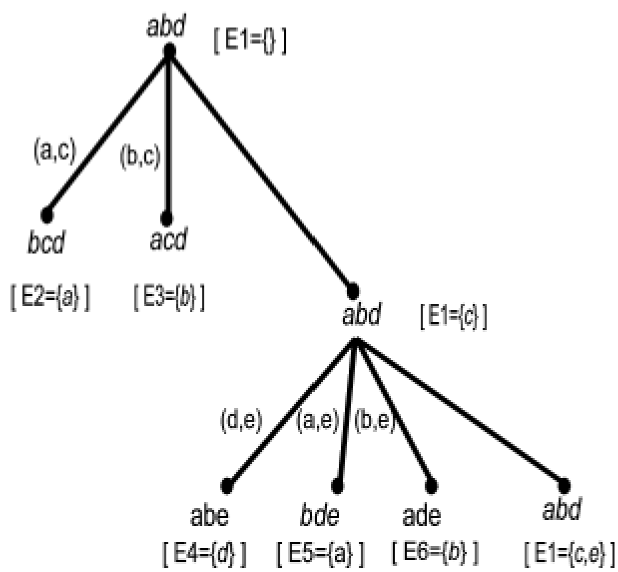


Fig:IV Intermidiate stage of ST enumeration

Here the spanning tree at the last node is E_1={*c* ,*e*}indicates that there is no chord remains for *abd.* Then the control goes to the second spanning tree present in ST set and assuming it as the root node with E2={a}, it starts enumerating spanning trees from it. In the mean while if it gets any duplicate spanning tree then it is discarded and updates its E set. There are 8 number of spanning trees of the graph G are enumerated successfully and are shown in Table-1 .

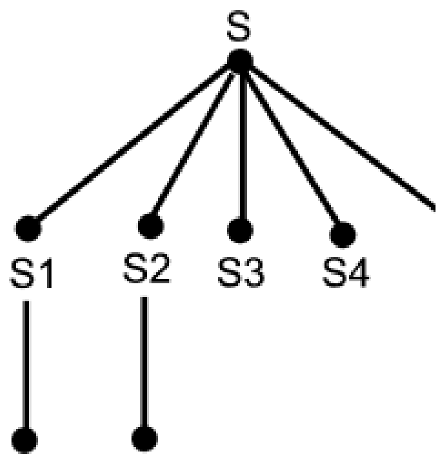


Fig:V Computation Tree where each node is a ST

While computing the spanning trees it generate two sets

ST_i and E_i which contains all the spanning trees as well as the edges that is not included in that ST . From the ST_i and E_i the reliability expression can be formulated. The ST_i and E_i for the given network is shown in Table-I

.Table I The generated ST_i and E_i set of G

i	ST_i	E_i
1	abd	Null
2	bcd	a
3	acd	b
4	abe	dc
5	bde	ac
6	ade	bc
7	bce	ad
8	ace	bd

The ST_i represents the reliability terms (*p* terms) and the E_i

represents the unreliability terms (*q* terms) of the reliability expression of the given Interconnection Network.

The reliability expression generated for the example network is $NR= p^3+ p^3q+ p^3q+ p^3q^2+ p^3q^2+ p^3q^2+ p^3q^2+ p^3q^2$

Considering the p=0.9, the NR of G is computed to be

0.91125.

V. EXPERIMENTAL RESULTS

In order to discuss the diversity of use of proposed method, it is applied to three different network types corresponding to following four distinguished cases:

1. Case- I – Irregular networks with homogeneous link capacities

Under this case, the irregular networks considered are shown in Fig. Vi. The no. of spanning trees generated along with the computed reliability are presented in Table-II. The CPU times as mentioned in Column 5 of this table are quite reasonable as well as acceptable.

2. Case-II- Regular networks with homogeneous link capacities

The regular networks of interest are Mesh, Crossed cube, Twisted pair cubes, Hypercube. The link capacities are considered as homogeneous here, since the networks are regular. The reliability computed for each of these networks as well as the CPU time are tabulated (Table-II). From the simulated result, the above mentioned networks can be arranged in following decreasing order of their computed reliability:

Crossed Cube, Twisted Pair HC, Mesh, Hyper Cube

3. Case-III- Irregular networks with heterogeneous link capacities

The irregular networks normally have links with heterogeneous capacities. So while evaluating reliability of these networks, both homogenous and heterogeneous link capacities have taken into consideration. The candidate irregular networks whose reliability are evaluated are shown in Fig. VI. Table IV presents the simulated results showing the no of generated spanning tress, the reliability values as well as CPU time.

Table II : Reliability of some irregular networks presented in Figure-VI with homogeneous link capacity.

SL	N,L	No. of ST generated	Reliability	TIME(Sec)
1	6,8	24	0.7528747	0.452401
2	8,12	288	0.9050142	18.365182
3	8,13	480	0.9051061	29.786515
4	9,12	153	0.6287834	6.176550
5	9,14	951	0.8804156	42.137510
6	10,14	576	0.8449175	31.000991

Table III: Reliability of some regular networks with homogeneous link capacity.

Network	No. of ST generated	Reliability	Time (Sec)
2D Mesh	4	0.947700	0.021
3D Mesh	142	0.916086	0.150
3D Crossed Cube	216	0.964967	0.331
Twisted Pair HC	253	0.949850	0.375
Hyper Cube	256	0.900381	0.336

Table IV: Reliability of some irregular networks presented in Figure-VII with heterogeneous link capacity.

SL	N,L	No. of ST generated	Reliability	Time(sec)
1	6,8	24	0.73113861	0.368414
2	6,9	55	0.78089292	1.563162
3	6,11	209	0.88239094	30.515458
4	7,10	81	0.74152988	3.692818
5	7,11	141	0.83764404	16.895961
6	8,12	204	0.68919088	28.857445
7	8,13	360	0.68940920	39.154381
8	9,12	153	0.57435398	19.829799
9	9,15	987	0.86649008	63.483222
10	10,14	576	0.88619346	44.988189

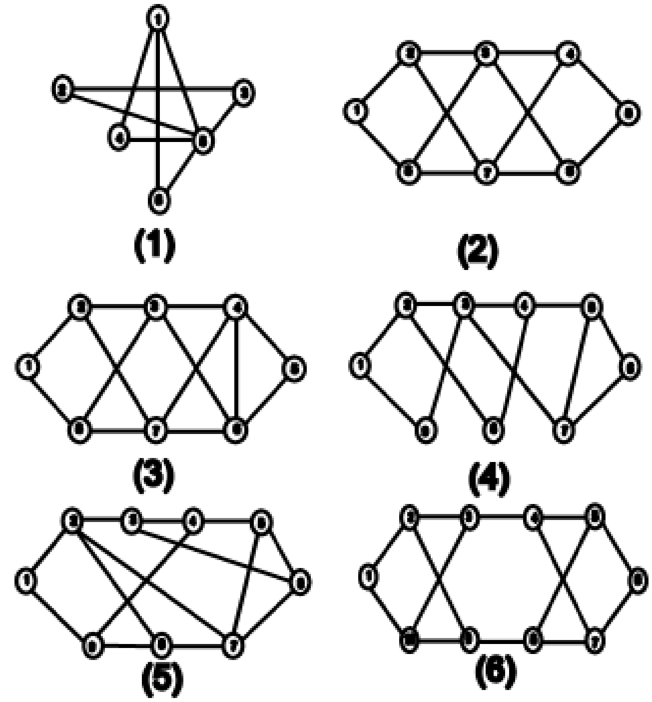
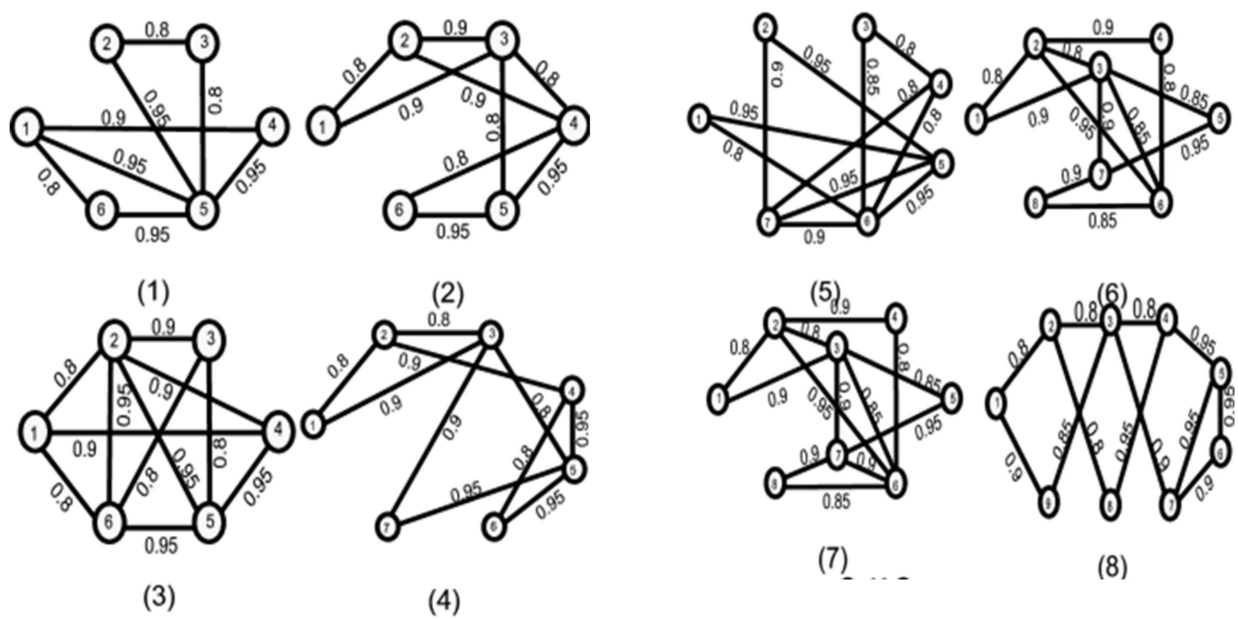
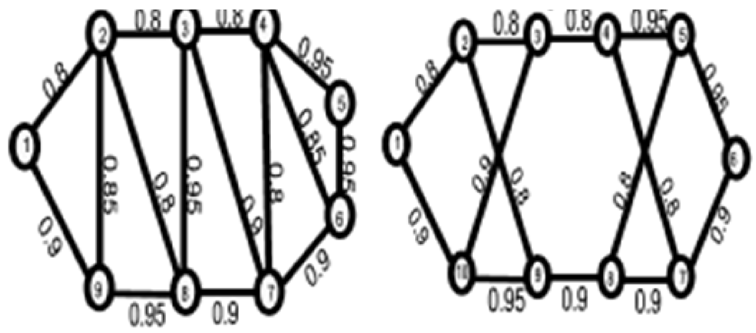


Fig VI: Some sample Interconnection Networks with homogeneous link capacity (p=0.9)





(9) **(10)**

FigVII: Some sample Interconnection Networks with heterogeneous link capacity

VI. CONCLUSIONS

The proposed approach presented in this paper is meant for exact estimation of networks reliability even for complex networks. The proposed approach not only efficiently enumerates all the spanning trees of the given network but also disjoint them implicitly to form the reliability expression. It ensures non-redundant generation of spanning trees of a given network .The proposed approach can be applicable to regular networks, general networks, and also for networks with homogenous and non-homogenous link capacity. The present work can be further extended by implementing different heuristic approaches to get near optimal solution for very large size networks.

REFERENCES

[1] H. Pham, “Special issue on critical reliability challenges and practices, ”*IEEE Transaction Systems, Man, Cybern. A, Syst., Humans,* vol. 37, no. 2, pp. 141–142, 2007.

[2] A. G. Sutcliffe and A. Gregoriades, “Automating scenario analysis human and system reliability,” *IEEE Transaction. Systems, Man, Cybern.A, Syst., Humans*, vol. 37, no. 2, pp. 249–261, 2007.

[3] Y.-S. Dai, M. Xie, and X. Wang, “A heuristic algorithm for reliability modeling and analysis of grid systems,” *IEEE Transaction.System., Man, Cybern.A, Syst., Humans*, vol. 37, no. 2, pp. 189–200, 2007.

[4] T. Aven, “Availability evaluation of oil/gas production and transportation system”, *Reliability, Eng.*vol.18, pp.35–44,1987.

[5] J. Von Neumann, C. E. Shannon and J. McCarthy, Eds. Princeton, Nj USA: Princeton Univ. Press” Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components”, Automata Studies, pp.43–98, 1956.

[6] W. C. Yeh, “A simple heuristic algorithm for generating all minimal paths,”*IEEETransaction.Reliability*, vol. 56, no. 3, pp. 488 – 494, 2007.

[7] W.-C. Yeh, “A simple universal generating function method to search for all minimal paths in networks,”*IEEE Transactions on .Systems*, Man, and Cybernetics Part A: Systems and Humans, vol.39, no.6, pp.1247–1254, 2009.

[8] S. G. Chen, “Search for all minimal paths in a general directed flow network with unreliable nodes,” International Journal of Reliability and Quality Performance, vol.2, no.2, pp.63–70, 2011.

[9] C. J. Colbourn, TheCombinatorics of Network Reliability. , NY: Oxford University Press, 1987.

[10] S. G. Chen, “Search for all minimal paths in a general directed flow network with unreliable nodes,” International Journal of Reliability and Quality Performance, vol.2, no.2, .63–70, 2011.

[11] Systems ”Proc. *IEEE Transaction* on PAS, Vol.Pas-98, PP - 2181-2189, 1979.

[12] R. A. Boedigheimer and K. C. Kapur, “Customer-driven reliability models for multistate coherent systems,”*IEEE Transaction. Reliability*, vol. 43, no.1, pp. 46–50, 1994.

[13] R. Billinton and W. Zhang, “State extension for adequacy evaluation of composite power systems - applications,” *IEEE Transaction Power Syst*em, vol. 15, no. 1, pp. 427–432, 2000.

[14] J. Xue, “On multistate system analysis”, *IEEE Transaction. Reliability, vol. R-34,* no. 4, pp. 329–337, Dec. 1985.

[15] C.-C. Jane and Y.-W.Laih, “A practical algorithm for computing multi-state two-terminal reliability,” *IEEE Transaction.Reliability”,* vol. 57, no. 2, pp. 295–302, 2008.

[16] J. E. Ramirez-Marquez and D. Coit, “Composite importance measures for multistate systems with multistate components,” *IEEE* Transactions on Reliability, vol. 53, no. 3, pp. 517–529, 2005.

[17] Gagneur, J.Klamt, S. Computation of elementary modes: A unifying framework and the new binary approach. BMC Bioinformatics 2004.

[18] Trinh, C.T.; Wlaschin, A.; Srienc, F. Elementary mode analysis: A useful metabolic pathway analysis tool for characterizing cellular metabolism. Appli.Microbiol.Biotechnol, vol. 81, pp. 813–826, 2009.

[19] Fell, D.A.; Wagner, A. The small world of metabolism. Nat. Biotechnol, vol. 18, pp. 1121–1122, 2000.

[20] Schilling, C.H.; Schuster, S.; Palsson, B.O.; Heinrich, R. Metabolic pathway analysis: Basic concepts and scientific applications in the post-genomic era. Biotechnol.Prog, vol. 15, pp.296–303, 1999.
[21] Papin, J.A. Comparison of network-based pathway analysis methods. Trends Biotechnol, vol. 22, pp. 400–405, 2004.
[22] Klamt, S. Generalized concept of minimal cut sets in biochemical networks. Biosystems, vol. 83, pp. 233–247, 2006.
[23] Klamt, S.; Gilles, E. Minimal cut sets in biochemical reaction networks. Bioinformatics, vol. 20, pp. 226–234, 2004.
[24] T. Jin and D.W. Coit, "Approximating network reliability estimates using linear and quadratic unreliability of minimal cuts," *Reliability Engineering and systems safety,vol.82,pp.41-83*, 2003.
[25] N. K. Goyal, R. B. Misra, and S.K. Chaturvedi, "SNEM: A new approach to evaluate terminal pair reliability of communication network, "*Journal of Quality in maintenance Engineering*,vol.11, pp.239-253, 2005.
[26] G.B.Jasmon and K.W. Foong, "A method for evaluating all the minimal cuts of a graph, " *IEEE Treansaction on reliability,* vol. R-36, no. 5, pp. 239-245,1987.
[27] B.A. Gebre and J. E. Raimer-marquiz." Element substitution algorithm for two terminal network reliability analysis," *IIE Transactions on operation engineering*,vol.39,pp.265-275, 2007.
[28] S. Soh and S. Rai, "CAREL: Computer aided reliability evaluator for distributed computing networks," *IEEE transaction on parallel and distributed systems,* vol.2, no. 2,pp. 199-213,1991.