Journal of Reliability and Statistical Studies; ISSN (Print): 0974-8024, (Online): 2229-5666
Vol. 10, Issue 1 (2017): 15-26

AN IMPROVED ALGORITHM FOR K-TERMINAL PROBABILISTIC NETWORK RELIABILITY ANALYSIS

S. Chatterjee, Venkata Ramana B.*, Gajendra K. Vishwakarma and Aman Verma
Department of Applied Mathematics
Indian Institute of Technology (Indian School of Mines), Dhanbad
Jharkhand, INDIA
E Mail: chatterjee_subhashis@rediffmail.com, ramanabv09@gmail.com*,
vishwagk@rediffmail.com, amanverma@am.ism.ac.in

Received January 11, 2017
Modified March 03, 2017
Accepted April 03, 2017

Abstract

An optimal variable ordering technique for Reduced Ordered Binary Decision Diagram (ROBDD) has been proposed to solve the network reliability analysis problem for complex communication networks. Several approaches have been proposed in the literature using the static and dynamic variable ordering techniques to solve the terminal network reliability problems. In this paper, the Minimal path set enumeration of the networks using Breadth first search traversal and ROBDD based Sift-reordering technique on manipulating the reliability evaluation is presented. The experimental results are compared with the previous approaches in computational time and number of ROBDD nodes to evaluate the *K*-terminal network reliability analysis.

Key Words: Minimal Path Set, Binary Decision Diagram, Ordered Binary Decision Diagram, Reduced Ordered Binary Decision Diagram, Variable Ordering Heuristics, Breadth First Search Traversal.

1. Introduction

Network reliability has always been a core problem domain for design engineers to enhance their ability for designing reliable networks. Network models can represent varieties of applied problems like transportation networks, computer architectures, data and voice communication networks and electrical networks etc. Earlier attempts were concentrated mostly on systems whose failure could cause massive damage or loss of property or human life. But, recently, it has been recognized that highly reliable systems make economic sense in a wide range of industries. Thus in present work reliability analysis and design of communication networks is carried out using graph theoretic approach for evaluating the different reliability measures. In the case of large and complex communication networks, it is quite difficult to solve the reliability measure. Recent literature describes that this problem has been solved by introducing efficient algorithm to minimize the execution time and memory requirements. This paper concentrates on the ROBDD approach for solving the *K*-terminal network reliability problem for large scale networks.

The general network reliability problem contains a graph G = (*V, E*) with a set of vertices (nodes) $|V|$ and the set of edges (links) $|E|$. The edges can fail randomly and independently with known probability. A computational measure of the network reliability problem has been classified into two major categories: approximation and exact methods. Approximation methods are a good solution for large networks. Efficient algorithms have been proposed to solve the network reliability measures. Colbourn (1987) has proved that the exact method of solving the problem of network reliability analysis is NP-hard. The exact method of solving the network reliability measure involves two kinds of approaches, one is Path/Cut set method and the second method is based on the topology of the network. A path/cut is defined as a specified set of nodes that can communicate/not communicate with each other and leads to the functioning/non-functioning of the network. A path is a minimal path if it has no proper subpaths. The reliability value is computed based on the formula of inclusion-exclusion principle as done by Al-Ghanim (1999). Yeh (2008) proposed branch and bound algorithm for multi-state networks. Soh and Rai (1993) used the sum of disjoint products technique for path/cut set approach and it has been improved by Yeh (2007). In general, the proposed algorithms based on the path/cut set methods by Kobayashi and Yamamoto (1999), Shen (1995) and Mishra and Chaturvedi (2009) for solving the two-terminal network reliability problem are not recommended for large-scale networks due to exponential complexity nature. Hui (2007) have estimated the reliability ranking for edge relocated networks and Gertsbakh and Shungin (2008) have proposed to solve the network reliability importance measures using Monte Carlo simulation technique. On the other hand, to overcome the limitations of the path/cut set methods, state enumeration methods, topological methods and decomposition methods have been used by Satyanarayana (1983), Li and He (2002) and Kim and Kang (2013) and also some of the transformation methods have been developed. Binary Decision Diagram (BDD) is used to manipulate the Boolean expressions and is quite efficient in time and memory management particularly in the case of large-scale networks.

BDD-based methods have been studied by many authors, including Zang et al. (1999), Kuo et al. (1999) and Rauzy (2003) for 2-terminal reliability graph analysis, later, it has been extended to *K*-terminal as well as for all-terminal networks by Yeh et al. (2002), Hardy et al. (2007), Kuo et al. (2007) and Sekine (1999) with the extended BDDs. An efficient ROBDD based technique using cutsets has been developed by Xing (2008) for 2-terminal network reliability and sensitivity analysis. In this paper, the authors have tried to enhance the ROBDD based analysis of *K*-terminal network reliability analysis and compared the efficiency with Yeh et al. (2002) and Hardy et al. (2007).

The rest of the paper is organized as follows: Section-2 describes the problem statement and its assumptions about the network. Section-3 explores the fundamentals of the ROBDD with variable ordering techniques. The proposed methodology and algorithms have been discussed in the Section-4. Further, the evaluation of the *K*-terminal network reliability analysis and experimental results have been discussed in Sections 5 and 6. Finally, Section-7 concludes the paper.

2. Problem description

The main objective of the paper is to compute the *K*-terminal reliability of a given network via the minimal path by using the ROBDD method. Consider a network graph $G = (V, \text{E}, \text{P})$ as shown in Figure 1, where *V* - denotes the set of nodes, *E* - is the set of edges or arcs and *P* - is the probability assigned to each edge. Major assumptions about the network G are: each node or edge is subject to fail randomly and independently with known probability value. In this paper, the probability value of the each edge is fixed at 0.9. For efficient manipulation of the network reliability, Breadth First Search Traversal Method is used for enumerating minimal path sets by traversing the network. Further, the reliability of the network is computed by using traditional reliability algorithm from the bottom-up approach using the canonical ROBDD.

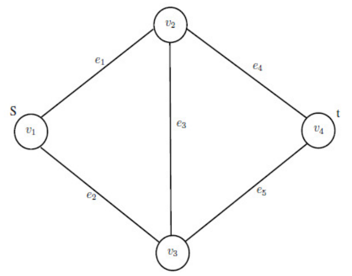


Figure 1: Sample Network

3. Binary Decision Diagram

Binary decision diagrams were discovered to represent the switching circuits and the basic definitions are given by Lee (1959) to represent the switching circuits. Later the research has been extended to VLSI design systems by Akers (1978). Efficient algorithms have been developed by introducing the Ordered BDDs and Reduced Ordered BDDs by Bryant (1986) to extend their application into various fields of research like reliability, computer science, data mining and graph theory. BDDs were first adapted to the network reliability analysis of binary state systems with Zang et al. (1999), in which the both system and its components are assumed to be in two states: operational or failed.

3.1 Preliminaries

BDDs are primarily defined as graph representation of boolean functions. Let $B = \{0,1\}$ and $n \in N$. The set of variables is denoted by $X_n = \{x_1, x_2, \cdots, x_n\}$.

Definition 3.1: Let $m, n \in N$, $f : B^n \to B^m$ is called a boolean function, in which *f* is a single output function for $m = 1$, otherwise multi-output function. A boolean variable x_i and $\overline{x_i}$ can be interpreted as

$$x_i : B^n \to B;\ (y_1, y_2, \cdots, y_i, \cdots, y_n) \to y_i$$

$$\overline{x}_i : B^n \to B;\ (y_1, y_2, \cdots, y_i, \cdots, y_n) \to \overline{y}_i$$

Definition 3.2: BDD is a graph based data-structure and is defined as a rooted directed acyclic graph (DAG) with two terminal nodes and a set of non-terminal nodes. The two terminal nodes are labeled by logical values *0* and *1*. Each non-terminal node associated with an input variable *x* of the boolean function and has two outgoing edges, termed as *0*-edge and *1*-edge. Each node in the BDD represents the Shannon decomposition on the boolean function: $f : B^n \rightarrow B^m, \forall\ x_i \in X_n :\ f = x_i \cdot f_{x_i=1} + \overline{x}_i \cdot f_{x_i=0}$

Definition 3.3: An Ordered Binary Decision Diagram (OBDD) is a BDD where the order of the input variables is fixed in all the paths of the graph and no variable appears more than once in a path. OBDD is a pair of variable ordering and finite directed acyclic graph (*G*) with a set of vertices (*V*) and edges (*E*) with exactly one root node is derived from the graph (*G*) as shown in Figure 2.

Definition 3.4: Any OBDD can be reduced to a Reduced Ordered Binary Decision Diagram (ROBDD) in linear time and linear space. ROBDD is the canonical form of boolean functions that can be derived by applying the following two reduction rules on ordered BDDs when the order of the variables is fixed.

- *Elimination rule*: Eliminate all the redundant nodes of the OBDD whose two edge point to the same node (Figure 2(b)).
- *Merging rule*: Share all the isomorphic sub-graphs as one sub-graph (OBDD) (Figure 2(b)).

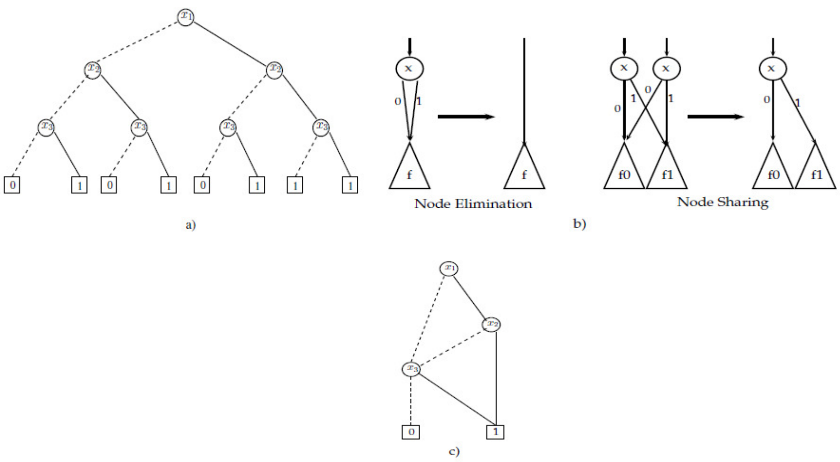
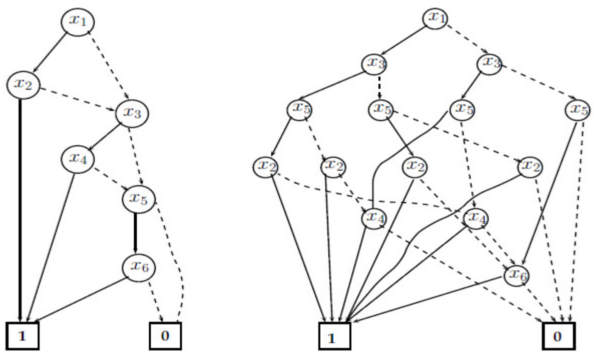


Figure 2: BDD construction: a) Original Decision Graph of the function $f = x_1x_2 + \overline{x_1}x_3 + x_1\overline{x_2}x_3$ with the variable order $x_1 < x_2 < x_3$ b) ROBDD reduction rules c) Reduced Ordered BDD

4. Variable Ordering

Computation of the optimal variable ordering for OBDD is NP-complete. Bollig and Wegener (1996) proved that the ROBDDs yield a minimized OBDD-representation for a given network, provided the variable ordering is appeared to be fixed. Based on the analysis of boolean function manipulation, the variable ordering heuristics can be classified as static and dynamic variable ordering techniques. Static variable ordering allows establishing the particular variable order prior to the construction of the BDD. These methods are deterministic and are based on the analysis of the formula of the network. Dynamic variable ordering addresses the two limitations of static variable ordering selection: (1) it is constant across all functions and (2) it is constant over all time of the BDD construction. Since the static heuristics are quite effective in selecting a good variable order for the single function, it is difficult to solve the optimal variable order for multiple functions. The size of the ROBDD is heavily dependent on the order of the input variables and keeping the same variable ordering throughout the sequence of operations will affect the size of the BDD. To overcome the drawbacks of static variable ordering techniques, various minimization algorithm have been proposed in the literature. A dynamic variable ordering for OBDDs has been introduced by Rudell (1993) with an efficient Swap algorithm which involves in exchanging all BDD nodes from level *i* to *i*+1 and vice versa. Bollig and Ingo (1996) have investigated on other exchange operations and bounds on swap algorithm to improve the time complexity and succeed in obtaining the optimal solutions.

In the present paper, the authors have implemented a C++ program which uses the dynamic Sift_reordering algorithm, that the variable ordering is systematically improved by swapping the adjacent variables sequentially. The algorithm turned out to be most efficient and popular because of its ability to select the variable to any position in the order in a short time. Its efficiency is completely based on the swap adjacent variable. The CUDD library has been used to manipulate the ROBDD operations designed by Somenzi (2012).



a) Variable order $x_1, x_2, x_3, x_4, x_5, x_6$ b) Variable order $x_1, x_3, x_5, x_2, x_4, x_6$

Figure 3: Variable ordering of the function $x_1x_2 + x_3x_4 + x_5x_6$

Further, the reliability is computed based on the ROBDD from the bottom-up approach. Figure 3 describes the two different kinds of variable ordering of the function $x_1x_2 + x_3x_4 + x_5x_6$. The linear variable ordering is presented in Figure 3 (a) and Figure 3(b) shows that exponential variable ordering for the same boolean function.

5. Proposed Approach

5.1 Algorithm

The Sift re-ordering technique has been implemented to enumerate the ROBDD number of nodes, in which the variables are moved in groups instead of a single variable. This section represents the algorithms which explain the proposed method for calculation of various measures of network reliability analysis. The pseudo code of the algorithm 1 explains the K - a terminal Network reliability evaluation procedure using ROBDD. The construction of the linear variable ordering ROBDD is shown in the algorithm 2. Finally, using the pseudocode given in the algorithm 3, it evaluates the reliability of the input network and combines the overall reliability in case of $|K| = |V|/2$ and $|K| = |V|$. The procedure for evaluation of the different measures of network reliability is discussed in the following sub-section.

Algorithm 1. *K*-terminal Network Reliability

```
Network _Reliability(Graph)
K =  no. of terminals
Terminal [N] are terminal values
Final_ROBDD = 1
for  i = 2 to n
Final_ROBDD = (Final_ROBDD) and
  (Create_ROBDD_2Terminals(Graph, Terminal[1], Terminal[i]))
Reliability = Computation_Reliability(Final_ROBDD)
return Reliability
```

Algorithm 2. ROBDD Construction

```
T_ROBDD = 0
Initialize start_node
Let e_i be an edge in the edge set E do
Next_node == sink_node then
Subpath_ROBDD = e_i_ROBDD
else if next_node is already  in this path then
continue;
else
Subpath_ROBDD = ROBDD gen(next_node) and e_i_ROBDD
T_ROBDD = T_ROBDD or subpath_ROBDD
end if
end for
clear start_node in this path
return T_ROBDD
```

Algorithm 3. Reliability Evaluation

```
Compute_Reliability (Final_ROBDD)
Require: Final_ROBDD
float Rk, Prob_1, Prob_0
if(Final_ROBDD == One)
return 1
if(Final_ROBDD == Zero)
return 0
if (Rk  = Computed  (Final_ROBDD))
return Rk
Prob_1 = Compute_Reliabilty (Sub ROBDD at one  edge of final_ROBDD)
Prob_0 = Compute_Reliabilty (Sub ROBDD at zero  edge of final_ROBDD)
InsertComputed(Final_ROBDD, Rk)
return Rk
```

5.2 Estimation of Network Reliability

In this section, evaluation of the *K*-terminal network reliability of the benchmark graphs given in Figure 5, using the efficient ROBDD method has been presented. The size of ROBDD is mainly dependent upon the variable order and not on the size of the network. Optimizing the size of ROBDD is recognized as an NP-complete problem. Initially, the minimal path sets are being evaluated by traversing the network using breadth-first search and a state of art technique is adopted for efficient manipulation of the network. The resulting computation process of enumeration *1*-paths of subgraphs of *G* is then combined to form the total number of minimal paths to *K*-terminal subgraphs of *G*. By sharing these subgraphs of the binary decision diagram, the given network can be modified as a reduced ordered directed acyclic graph. Based on these disjoint property of BDD graph analysis the *K*-terminal network reliability can be evaluated. The probability of the edges of the given network are denoted by p_k, thus the measure of the *K*-terminal reliability function *f* is evaluated recursively obtain by the formula: $\Pr(f) = \Pr(x_i) \cdot \Pr(f_{x_i=1}) + [1 - \Pr(x_i)] \cdot \Pr(f_{x_i=0})$.

The two-terminal network reliability using ROBDD has been explained graphically in Figure 4. Further, the reliability function for an undirected graph given in Figure 1 of *K*-terminal network can be derived by connecting all the (*K*-1) 2-terminal networks if this terminal-pair set can cover all the nodes. Therefore, the reliability function for an undirected *K*-terminal network can be denoted as $\mathrm{Re}\,l_k(G) = \prod_{k-1} \mathrm{Re}\,l_2(G_{s,t})$.

The reliability function can be derived from the paths:

$$\mathrm{Re}\,l_k(G) = \mathrm{Re}\,l_k(2,3,4) = e_3e_5 + e_3e_4 + e_1e_2e_3e_4 + e_1e_2e_5 + e_4e_5$$

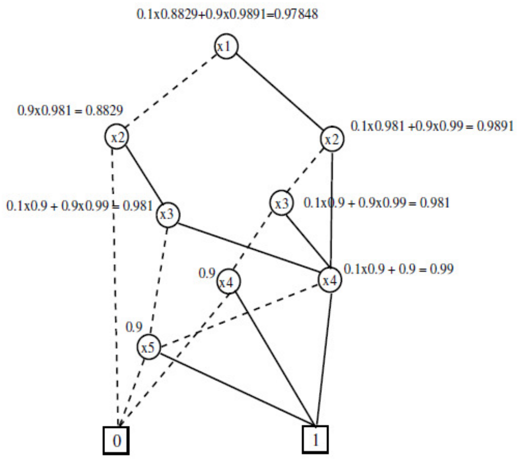


Figure 4: Reliability evaluation using ROBDD path function

6. Experimental Results

The proposed approach for determining the *K*-terminal reliability of the benchmark networks has been implemented on the Xeon E5 Workstation with 4GB memory. The algorithm is implemented using a C++ programming language and the program files were compiled with GNU g++ 4.8.4-2 library on Ubuntu operating system with kernel version 3.19.0-61-generic. The efficient manipulation of ROBDD operations was organized by CUDD library. Figure 5 shows the benchmark networks used to enumerate the *K*-terminal, 2-terminal and All-terminal network reliability using ROBDDs. Table 1 explores the simulation results developed for the benchmark networks imposed from Yeh et al. (2002). Important network reliability measures of which $K = 2$, $|K| = |V|/2$ and $|K| = |V|$ are being evaluated for each network shown in Figure 5 and the experimental results are being compared with the previous works. In the Table 1, the *EED_BFS* column denotes the method of Edge Expansion Diagram, proposed by Yeh et al. (2002) and the column *Hardy* explores the results of the efficient algorithm proposed by Hardy et al. (2007). The columns ROBDD are the results obtained by the optimal variable ordering technique used in the present paper.

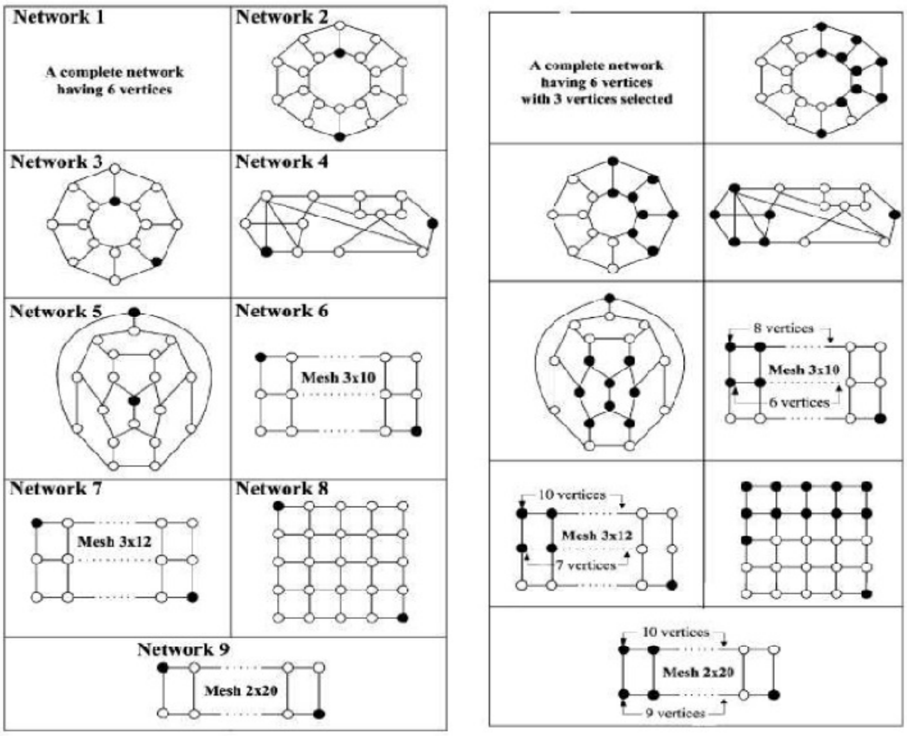


Figure 5: Benchmark Networks

Network			|K|=2				|K| = |V|/2			
Type	**|V|**	**|E|**	**EED _BFS**	**Hardy**	**ROB DD**	**Reliability**	**EED_ BFS**	**Hardy**	**ROB DD**	**Relia bility**
1	6	15	148	154	121	0.99998	176	145	162	0.99997
2	20	30	423	65665	260	0.994395	518	424450	530	0.984681
3	16	24	302	8452	184	0.995553	356	31659	362	0.988922
4	14	23	292	1810	154	0.98739	294	2639	199	0.98520
5	20	30	3591	5370	1204	0.99712	5697	6995	1728	0.989092
6	30	47	257	356	235	0.964474	310	415	259	0.954935
7	36	57	317	440	295	0.96173	383	376	330	0.95108
8	25	40	1149	2425	1269	0.975557	1633	1794	2139	0.95742
9	40	58	115	152	113	0.784482	136	989	133	0.765248

Network			|K|=|V|			
Type	**|V|**	**|E|**	**EED_BFS**	**Hardy**	**ROBDD**	**Reliability**
1	6	15	173	174	158	0.99994
2	20	30	221	45307	295	0.973503
3	16	24	295	5961	221	0.979658
4	14	23	232	2567	102	0.980765
5	20	30	1903	4813	1511	0.960069
6	30	47	184	236	183	0.924055
7	36	57	226	290	289	0.9173
8	25	40	697	1397	879	0.93981
9	40	58	97	1541	1503	0.7453

Table 1: Results of Network Reliability measures

Network			|K|=2			|K| = |V|/2		
Type	**|V|**	**|E|**	**EED_BFS**	**Hardy**	**ROBDD**	**EED_BFS**	**Hardy**	**ROBDD**
1	6	15	0.02	0.03	0.00	0.77	0.03	0.00
2	20	30	0.08	0.35	0.07	1.03	1.50	0.55
3	16	24	0.02	0.11	0.00	0.87	0.50	0.03
4	14	23	0.02	0.06	0.00	0.76	0.07	0.00
5	20	30	0.33	0.09	0.13	2.18	0.12	0.29
6	30	47	0.55	0.06	2.01	2.50	0.40	6.22
7	36	57	2.88	0.05	5.26	7.80	0.02	9.22
8	25	40	0.43	0.05	0.56	2.50	0.08	3.05
9	40	58	0.02	0.08	29.32	1.33	0.07	29.80

Network			|K|=|V|		
Type	**|V|**	**|E|**	**EED_BFS**	**Hardy**	**ROBDD**
1	6	15	0.82	0.04	0.00
2	20	30	0.95	0.20	1.32
3	16	24	1.08	0.06	0.09
4	14	23	0.83	0.07	0.02
5	20	30	2.50	0.12	0.65
6	30	47	2.40	0.08	18.90
7	36	57	7.73	0.04	25.46
8	25	40	2.72	0.07	7.87
9	40	58	1.55	0.05	171.19

Table 2: Execution time for estimating the ROBDD nodes and Reliability

The proposed method shows comparatively good results than previous approaches for the various measures of network reliability analysis. The computation time for the enumeration of ROBDD nodes to represent the minimal number of paths of the input network and reliability evaluation is shown in Table 2. But in the case of complete networks, sift-reordering technique performance is efficient in ROBDD number of nodes but the computation time increases with the size of the network.

7. Conclusion

In this paper, the network reliability analysis problem has been solved to enhance the effort of optimal variable ordering of the ROBDD with the sift-reordering technique. The results have been improved using this technique. Execution time to enumerate the ROBDD number of nodes and the reliability value are compared with the recent approaches with same benchmark networks given in the literature. Breadth First Traversal Method is used to find the minimal paths of the given graph and reliability is evaluated using efficient ROBDD approach. The execution time to enumerate the ROBDD number of nodes and reliability analysis is less than other algorithms.

Acknowledgements

The authors are very indebted to the anonymous referees, the Executive Editor Professor S.B. Singh and the Editor-in-Chief Professor Vinod Kumar for their valuable suggestions regarding improvement of the paper.

References

1. Akers, S.B. (1978). Binary decision diagrams, IEEE Transactions on computers, 100(6), p. 509-516.
2. Al-Ghanim, A.M. (1999). A heuristic technique for generating minimal path and cutsets of a 24 general network, Computers & Industrial Engineering, 36 (1), p. 45-55.

3. Bollig, B. and Wegener, I. (1996). Improving the variable ordering of OBDDs is NP-complete, IEEE Transactions on computers, 45 (9), p. 993-1002.
4. Bryant, R.E. (1986). Graph-based algorithms for boolean function manipulation, IEEE Transactions on Computers, 100 (8), p. 677-691.
5. Colbourn, C. (1987). The Combinatorics of Network Reliability, Oxford University Press, New York, USA.
6. Gertsbakh, I. and Shpungin, Y. (2008). Network reliability importance measures: combinatorics and monte carlo based computations, WSEAS Transactions on Computers, 7 (4), p. 216-227.
7. Hardy, G., Lucet, C. and Limnios, N. (2007). K-terminal network reliability measures with binary decision diagrams, IEEE Transactions on Reliability, 56 (3), p. 506-515.
8. Hui, K.P. (2007). Monte Carlo network reliability ranking estimation, IEEE Transactions on Reliability, 56 (1), p. 50-57.
9. Kim, Y. and Kang, W.H. (2013). Network reliability analysis of complex systems using a non-simulation-based method, Reliability Engineering & System Safety, 110, p. 80-88.
10. Kobayashi, K. and Yamamoto, H. (1999). A new algorithm in enumerating all minimal paths in a sparse network, Reliability Engineering & System Safety, 65 (1), p. 11-15.
11. Kuo, S.Y., Lu, S.K. and Yeh, F.M. (1999). Determining terminal-pair reliability based on edge expansion diagrams using OBDD, IEEE Transactions on Reliability, 48 (3), p. 234-246.
12. Kuo, S.Y., Yeh, F.M. and Lin, H.Y. (2007). Efficient and exact reliability evaluation for networks with imperfect vertices, IEEE Transactions on Reliability, 56 (2), p. 288-300.
13. Lee, C.Y. (1959). Representation of Switching Circuits by Binary-Decision Programs, Bell Labs Technical Journal, 38 (4), p. 985-999.
14. Li, J. and He, J. (2002). A recursive decomposition algorithm for network seismic reliability evaluation, Earthquake Engineering & Structural Dynamics, 31 (8), p. 1525-1539.
15. Mishra, R. and Chaturvedi, S.K. (2009). A cutsets-based unified framework to evaluate network reliability measures, IEEE Transactions on Reliability, 58 (4), p. 658-666.
16. Rauzy, A. (2003). A new methodology to handle Boolean models with loops, IEEE Transactions on Reliability, 52 (1), p. 96-105.
17. Rudell, R. (1993). Dynamic variable ordering for ordered binary decision diagrams, Proceeding of the IEEE/ACM International conference on Computer-aided design, p. 42-47.
18. Satyanarayana, A. and Chang, M.K. (1983). Network reliability and the factoring theorem, Networks, 13 (1), p. 107-120.

19. Sekine, K. (1999). Computational investigations of all-terminal network reliability via BDDs, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 82 (5), p. 714-721.
20. Shen, Y. (1995). A new simple algorithm for enumerating all minimal paths and cuts of a graph, Microelectronics Reliability, 35 (6), p. 973-976.
21. Soh, S. and Rai, S. (1993). Experimental results on preprocessing of path/cut terms in sum of disjoint products technique, IEEE Transactions on Reliability, 42 (1), p. 24-33.
22. Somenzi, F. (2012). CUDD-2.5.0, http://vlsi.colorado.edu.
23. Xing, L. (2008). An efficient binary decision diagram based approach for network reliability and sensitivity analysis, IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 38 (1), p. 105-115.
24. Yeh, F.M., Lu, S.K. and Kuo, S.Y. (2002). OBDD-based evaluation of k-terminal network reliability, IEEE Transactions on Reliability, 51 (4), p. 443-451.
25. Yeh, W.C. (2007). An improved sum of disjoint products technique for the symbolic network reliability analysis with known minimal paths, Reliability Engineering & System Safety, 92 (2), p. 260-268.
26. Yeh, W.C. (2008). A greedy branch and bound inclusion-exclusion algorithm for calculating the exact multi-state network reliability, IEEE Transactions on Reliability, 57 (1), p. 88-93.
27. Zang, X. Sun, N. and Trivedi, K.S. (1999). A BDD-based algorithm for reliability analysis of phased-mission systems, IEEE Transactions on Reliability, 48 (1), p. 50-60.