

Network Reliability Computation by using Different Binary Decision Diagrams

Manoj Singhal¹, R. K. Chauhan², Girish Sharma³

1 Asso. Prof., Deptt. Of MCA, Academy of Business and Engineering Sciences, 19th K.M. stone, Delhi-Hapur by pass Road, NH-24, Vijay Nagar, Ghaziabad, (U.P.), India, manoj_singhal@rediffmail.com

2 Prof. & Chairman, Deptt. Of Computer Science and Applications, Kurukshetra University, Kurukshetra (Haryana), India, rkc.dcsa@gmail.com

3 Asso. Prof., Deptt. Of MCA, Bhai Parmanand Institute Of Business Studies, Delhi, India, gkps123@gmail.com

ABSTRACT

In this Paper we consider three different types of variable ordering; namely optimal ordering, good ordering and bad ordering for constructing the BDD of a given network by applying three different heuristics. This classification is based on the size of the BDD, because the size of the BDD strongly depends on the ordering of variables. After that we find the reliability of the given network by these different BDD. It is observed experimentally that the results (Reliability) of applying Classical Inclusion-exclusion principle are the same as obtained by applying BDD for calculating reliability of a given network in each case. However the complexity of the BDD increases in bad ordering case.

Key Words: *Binary Decision Diagrams (BDD), Directed Acyclic Graph (DAG), Computer communication Network (CCN), Ordered Binary Decision Diagrams (OBDD), Modified Binary Decision Diagrams (MBDD).*

1. INTRODUCTION

Network reliability analysis receives considerable attention for the design, validation, and maintenance of many real world systems, such as computer, communication, or power networks. The components of a network are subject to random failures, as more and more enterprises become dependent upon CCN or networked computing applications. Failure of a single component may directly affect the functioning of a network. So the probability of each component of a CCN is a crucial consideration while considering the reliability of a network. Hence the reliability consideration is an important factor in CCN. The IEEE 90 standard defines the reliability as “**The ability of a system or component to perform its required functions under stated conditions for a specified period of time.**” There are so many exact methods for computation of network reliability [1]. The network model is a directed stochastic graph $G = (V, E)$, where V is the vertex set, and E is the set of directed edges. An incidence relation which associates with each edge of G a pair of nodes of G , called its end vertices. The edges represent components that can fail with known probability. In real problems, these probabilities are usually computed from statistical data.

The problem related with connection function is NP-hard [2]. The same thing is observed for planar graphs [3]. In the exact method there are two classes for the computation of the network

reliability. The first class deals with the enumeration of all the minimum paths or cuts. A path is a subset of components (edges and/or vertices), that guarantees the source and the sink to be connected if all the components of this subset are functioning. A path is a minimal if a subset of elements in the path does not exist that is also a path. A cut is a subset of components (edges and/or vertices), whose failure disconnect the source and sink. A cut is a minimal if the subset of elements in the cut does not exist that is also a cut. The probabilistic evaluation uses the inclusion-exclusion or sum of disjoint products methods because this enumeration provides non-disjoint events. Numerous works about this kind of methods have been presented in literature [4, 5, 6].

In the second class, the algorithms are based on graph topology. In the first process we reduce the size of the graph by removing some structures. These structures are as polygon-to-chain [7] and delta-to-star reductions [8]. By this we will be able to compute the reliability in linear time and the reduction will result in a single edge. The idea is to decompose the problem in to one failed and another functioning. The same was confirmed by Theologou & Carlier [9] for dense networks. Satyanarayana & Chang [10] and Wood [11] have shown that the factoring algorithms with reductions are more efficient at solving this problem than the classical path or cut enumeration methods.

2. BINARY DECISION DIAGRAMS

Akers [12] first introduced BDD to represent Boolean functions i.e. a BDD is a data structure used to represent a Boolean Function. Bryant [13] popularized the use of BDD by introducing a set of algorithms for efficient construction and manipulation of BDD structure. The BDD structure provides compact representations of Boolean expressions. A BDD is a directed acyclic graph (DAG) based on the Shannon decomposition. The Shannon decomposition for a Boolean function is defined as follows:

$$f = x \cdot f_{x=1} + \bar{x} \cdot f_{x=0}$$

where x is one of the decision variables, and f is the Boolean function evaluated at $x = i$. By using Shannon's decomposition, any Boolean expression can be transformed in to binary tree. BDD are used to work out the terminal reliability of the links. Madre and coudert [14] found BDD usefulness in reliability analysis which was further extended by Rauzy [15, 16]. They are specially used to assess fault trees in system analysis. In the network reliability framework, Sekine & Imai [17], and Trivedi [18] have shown how to functionally construct the corresponding BDD.

Figure 1 shows the truth table of a Boolean function f and its corresponding Shannon tree. Sink nodes are labelled either with 0, or with 1, representing the two corresponding constant expressions. Each internal node u is labelled with a Boolean variable $\text{var}(u)$, and has two out-edges called 0-edge, and 1-edge. The node linked by the 1-edge represents the Boolean expression when $x_i = 1$, i.e. $f_{xi=1}$; while the node linked by the 0-edge represents the Boolean expression when $x_i = 0$, i.e. $f_{xi=0}$. The two outgoing edges are given by two functions $\text{low}(u)$ and $\text{high}(u)$. The diagram is given below.

| Truth Table | | | | Decision Tree |
|-------------|-------|-------|-----|---------------|
| x_1 | x_2 | x_3 | f | |
| 0 | 0 | 0 | 1 | |
| 0 | 0 | 1 | 1 | |
| 0 | 1 | 0 | 0 | |
| 0 | 1 | 1 | 1 | |
| 1 | 0 | 0 | 0 | |
| 1 | 0 | 1 | 1 | |
| 1 | 1 | 0 | 0 | |
| 1 | 1 | 1 | 1 | |

Figure 1. Truth Table of a Boolean Function f and its Corresponding Decision Tree

Note : A dashed (solid) tree branch denotes the case where the decision variable is 0 (1).

Indeed, such representation is space consuming. It is possible to shrink by using following three postulates.

1. **Remove Duplicate Terminals:** Delete all but one terminal vertex with a given label, and redirect all arcs into the deleted vertices to the remaining one.
2. **Delete Redundant Non Terminals:** If non terminal vertices u , and v have $\text{var}(u) = \text{var}(v)$, $\text{low}(u) = \text{low}(v)$, and $\text{high}(u) = \text{high}(v)$, then delete one of the two vertices, and redirect all incoming arcs to the other vertex.
3. **Delete Duplicate tests:** If non terminal vertex v has $\text{low}(v) = \text{high}(v)$, then delete v , and redirect all incoming arcs to $\text{low}(v)$.

If we apply all these three rules then the above decision tree can be reduced in to the diagrams given below in figure 2.

2.1 Ordered Binary Decision Diagrams

For an ordered BDD (OBDD), we impose a total ordering \prec over the set of variables and require that for any vertex u , and either non terminal child v , their respective variables must be ordered [19].

2.2 Modified Binary Decision Diagrams

Another BDD having the same size or less size of a given BDD is called the modified binary decision diagram (MBDD) [20].The size of the BDD means the total number of non-terminal nodes and number of nodes in a particular level [21].

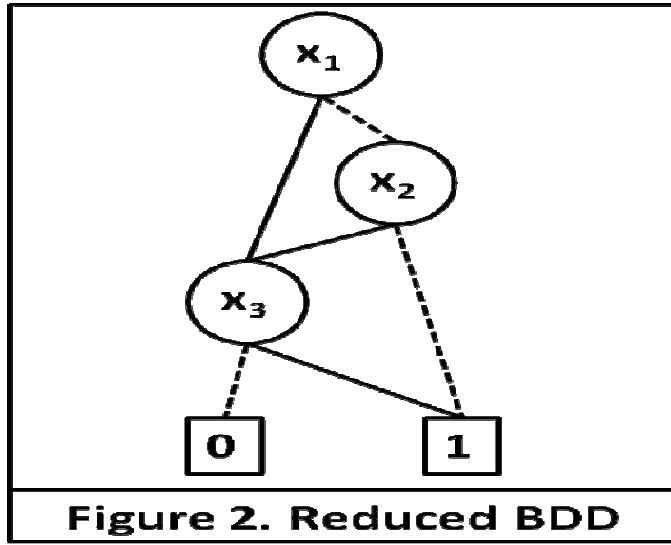


Figure 2. Reduced BDD

3. Network Reliability

The reliability of a network G is the probability that G supports a given operation. We distinguish three kinds of operation and hence three kind of reliability.

Two Terminal Reliability: It is the probability that two given vertices, called the source and the sink, can communicate. It is also called the terminal-pair reliability [22].

K Terminal Reliability: When the operation requires only a few vertices, a subset k of $N(G)$, to communicate with each other, this is K terminal reliability.

All Terminal Reliability: When the operation requires that each pair of vertices is able to communicate via at least one operational path, this is all terminal reliability. We can see that 2-terminal reliability and all terminal reliability are the particular case of K -terminal reliability.

Let us take an example of a directed network. The network is represented in the form of a graph (Figure 3) $G = (V, E)$, where V is the set of vertices (or nodes) and E the set of edges (or arcs). In order to study the reliability of a network we identify a single node as a source node (node 1) and a single node as a sink node (node 6) [22]. The analysis may be either qualitative or quantitative. Here we use path enumeration method to find the reliability of the example network.

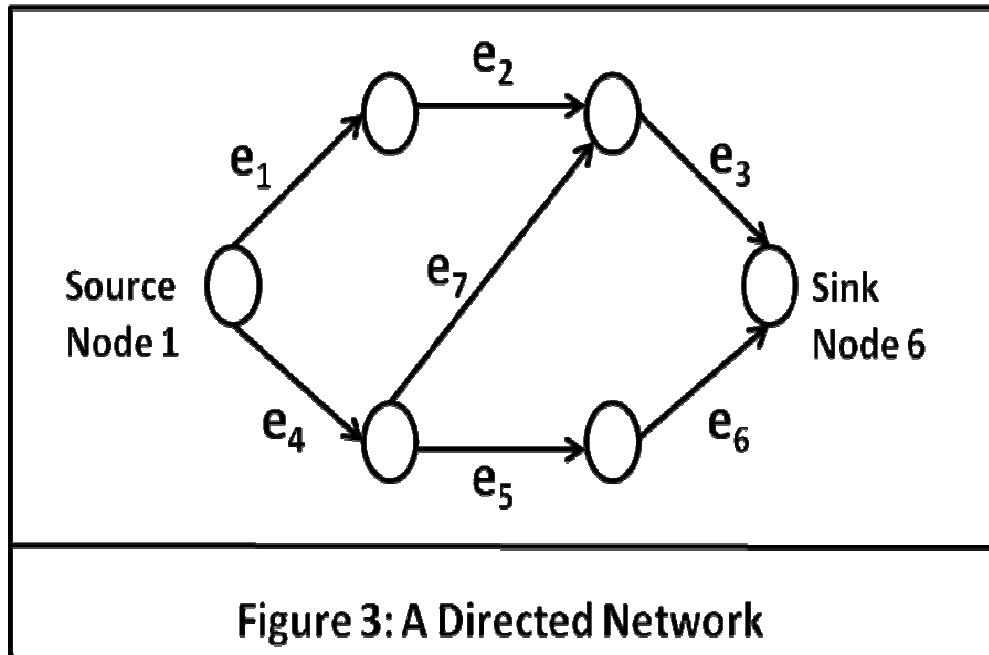
The network possess three min-paths. These are $H_1 = \{e_1, e_2, e_3\}$, $H_2 = \{e_4, e_5, e_6\}$ and $H_3 = \{e_4, e_7, e_3\}$.

Let H_1, H_2, \dots, H_n be the n min-paths then the network connectivity C can be represented as a logical OR of its min-paths.

$$C = H_1 \cup H_2 \cup \dots \cup H_n$$

So the point to point reliability is:

$$R_s = \Pr\{C\} = \Pr\{H_1 \cup H_2 \cup \dots \cup H_n\} \quad \text{----(1)}$$



The network connectivity of our network can be expressed as

$$C_{1-6} = e_1 e_2 e_3 \cup e_4 e_5 e_6 \cup e_4 e_7 e_3 \quad \dots (2)$$

The probability of the union of non-disjoint events, as in Formula (1), can be computed by several techniques. Here we use the inclusion-exclusion principle.

Inclusion-exclusion Formula: One method of transforming a Boolean expression $\Phi(G)$ into a probability expression is to use Poincare's theorem, also called inclusion-exclusion method.

Let us consider an example with two minimal paths H_1 and H_2 and the Boolean expression $\Phi(G) = H_1 + H_2$, then the probability expression $E(\Phi(G))$ can be expressed as follows:

$$E(H_1 + H_2) = E(H_1) + E(H_2) - E(H_1 \cdot H_2)$$

Poincare's formula for m min-paths :

$$E(\Phi(G)) = \sum_{1 \leq i \leq m} E(H_i) - \sum_{1 \leq j \leq k \leq m} E(H_j \cdot H_k) + \dots + (-1)^{m+1} E(H_1 \cdot H_2 \cdot H_3 \dots H_m)$$

Let P_i denote the probability of edge e_i of being working, by applying the Classical inclusion-exclusion formula for calculating the probability of given network (figure 3), we get

$$\begin{aligned} R_{1-6} = Pr\{C_{1-6}\} &= p_1 p_2 p_3 + p_4 p_5 p_6 + p_3 p_4 p_7 - p_1 p_2 p_3 p_4 p_5 p_6 - p_1 p_2 p_3 p_4 p_7 - p_3 p_4 p_5 p_6 p_7 \\ &\quad + p_1 p_2 p_3 p_4 p_5 p_6 p_7 \end{aligned} \quad \dots (3)$$

4. Generation of BDD and Reliability Computation

The BDD based network reliability analysis involves three main steps. These are as follows:

1. Order the link by applying a good heuristic approach.
2. Generate the BDD from the probabilistic graph of the given directed network.
3. Evaluate network reliability by using Shannon's decomposition.

A heuristic is good in the sense that it generates the minimum size BDD by taking an optimal variable ordering. A particular sequence of variables is known as a variable ordering. An ordering is said to be optimal if it generates the minimum size BDD. The size of the BDD means the total number of non-terminal nodes and number of nodes in a particular level. It has been observed that the size of the BDD strongly depends on the ordering of variables [23, 24, 25, 26].

In this paper we take three type of variable orderings. Let E be the total number of links (edges) in the network and N be the number of non-terminal nodes in the OBDD or MBDD.

4.1 Optimal Ordering: If the number of non-terminal nodes in the OBDD or MBDD is equal to the number of links in the network then the ordering is said to be optimal ordering ($N = E$).

4.2 Good Ordering: If $N \leq [E + (\lceil E/4 \rceil)]$, then the ordering is said to be good ordering where $\lceil E/4 \rceil$ is called the ceiling of E, denotes the least integer that is not less than E.

4.3 Bad Ordering: If $N > [E + (\lceil E/4 \rceil)]$, then the ordering is said to be bad ordering.

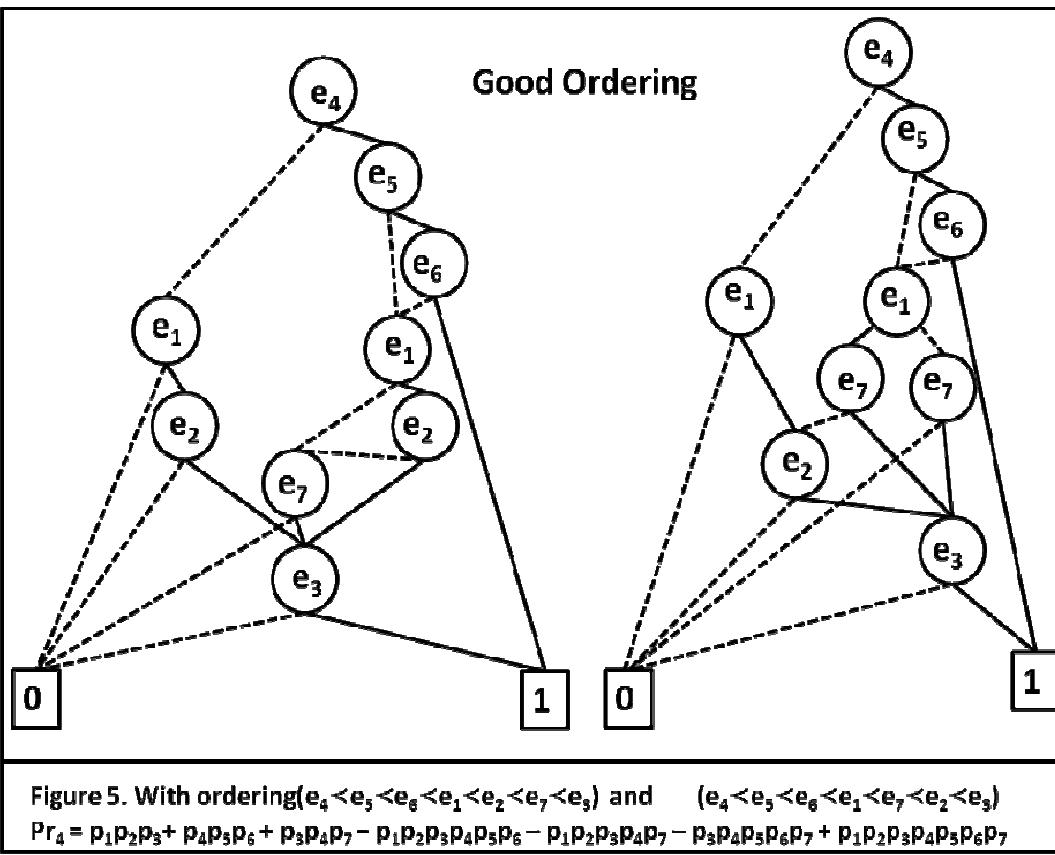
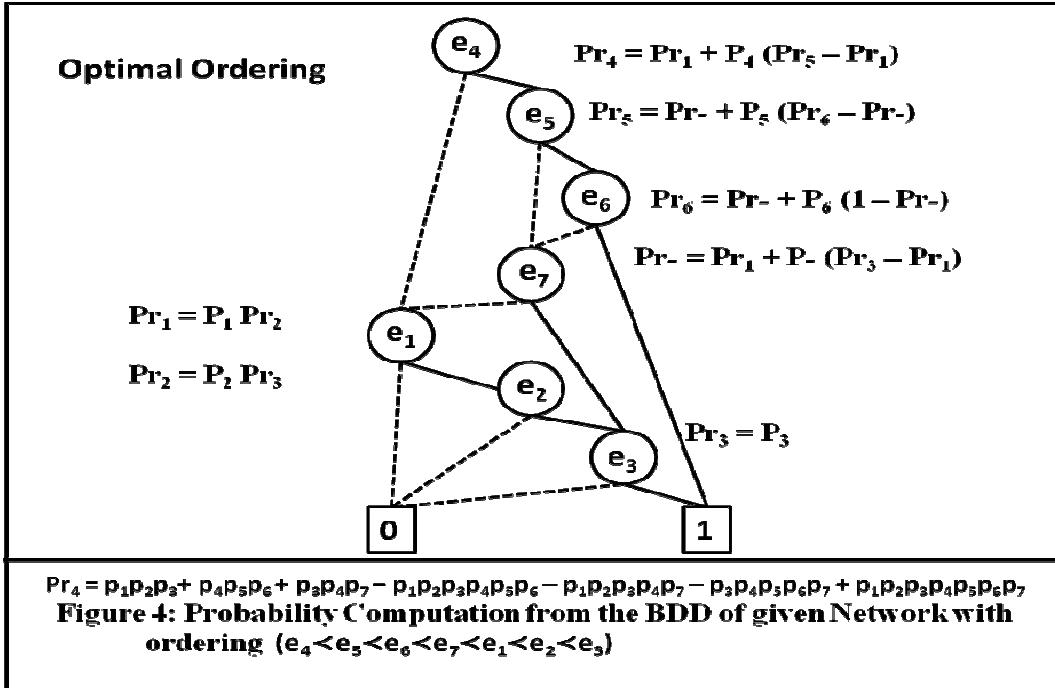
Now we apply the Shannon's decomposition to the Boolean connectivity function of the directed network expressed as the union of the min-paths in Formula (2). The Boolean connectivity expression (2) is built in figure 4, 5 and 6 by taking five different variables ordering. The computation of the probability of the BDD can again proceed recursively by resorting to the Shannon's decomposition.

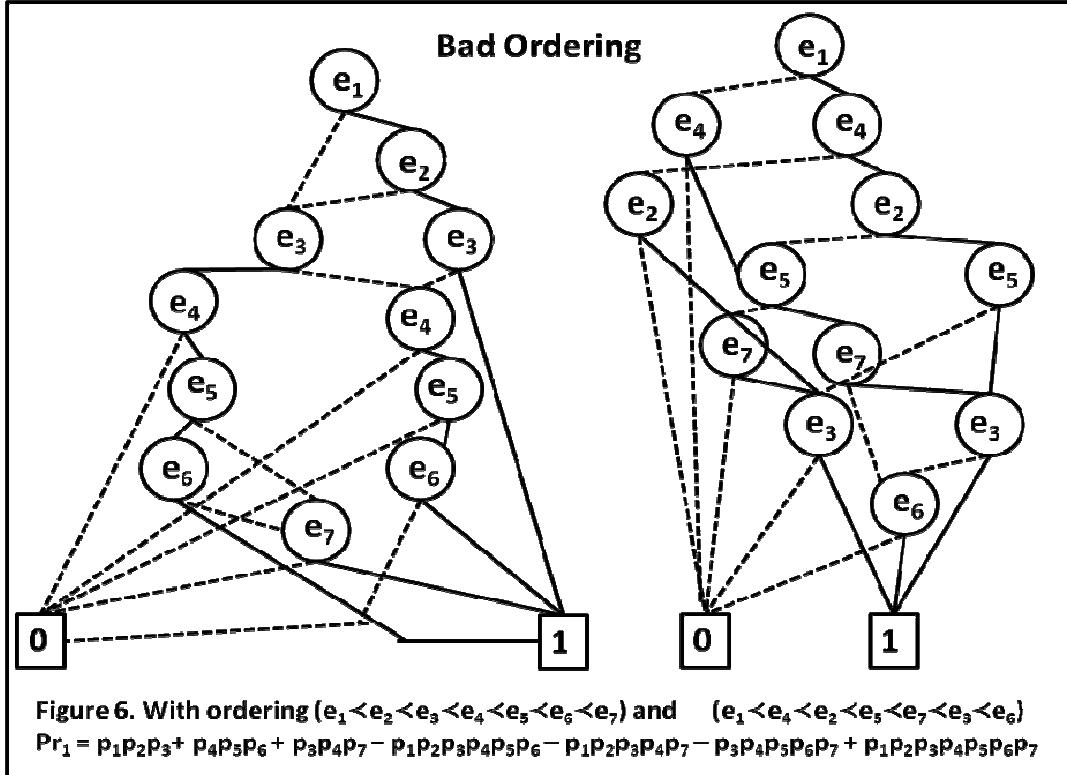
$$\Pr\{F\} = p_1 \Pr\{Fx_1 = 1\} + (1 - p_1) \Pr\{Fx_1 = 0\} = \Pr\{Fx_1 = 0\} + p_1(\Pr\{Fx_1 = 1\} - \Pr\{Fx_1 = 0\}) \quad (4)$$

where p_1 is the probability of the Boolean variable x_1 to be true and $(1 - p_1)$ is the probability of the Boolean variable x_1 to be false. Recursive application of Equation (4) is pictorially shown in Figure 4, 5 and 6.

Here we will show that the network reliability, which is obtained by Poincare theorem is equal to the network reliability, which is obtained recursively by different BDD (different ordering variables) of the same network (Figure 3).

Since the size of the BDD heavily depends on the variable ordering. The size of BDD means the total number of nodes in the BDD and number of nodes in a particular level. There are several variables ordering are possible for constructing the different BDD of the given CCN (figure 3). We have constructed only five different BDD of the given CCN and compute the reliability of the given CCN by using these different BDD. We found that the reliability obtained in each case by using BDD is same as the reliability obtained by inclusion-exclusion formula. But the size of the BDD is different in all the cases. Our program is written in the C language and computations are done by using a Pentium 4 processor with 512 MB of RAM. The computation speed heavily depends on the variables ordering because the size of the BDD heavily depends on the variable ordering.





For the ordering $e_4 < e_5 < e_6 < e_7 < e_1 < e_2 < e_3$ the size of the BDD is minimum; because it contains only seven non-terminal nodes and there are only seven edges (number of variables in connectivity function) in the given network. So the ordering $e_4 < e_5 < e_6 < e_7 < e_1 < e_2 < e_3$ is the optimal ordering for generating the BDD of the given network. The diagram is shown above in figure 4.

For the ordering $e_4 < e_5 < e_6 < e_1 < e_2 < e_7 < e_3$ and $e_4 < e_5 < e_6 < e_1 < e_7 < e_2 < e_3$, the BDD contain 9 non-terminal vertices. While for the ordering $e_1 < e_2 < e_3 < e_4 < e_5 < e_6 < e_7$ and $e_1 < e_4 < e_2 < e_5 < e_7 < e_3 < e_6$, the BDD have 11and 12 non-terminal vertices. So the size of the BDD is not minimum for these ordering and the complexity of the BDD in terms of size is more than the optimal ordering.

Thus the orderings $e_4 < e_5 < e_6 < e_1 < e_2 < e_7 < e_3$ and $e_4 < e_5 < e_6 < e_1 < e_7 < e_2 < e_3$, are the good ordering as per the definition. The ordering $e_1 < e_2 < e_3 < e_4 < e_5 < e_6 < e_7$ and $e_1 < e_4 < e_2 < e_5 < e_7 < e_3 < e_6$, are bad orderings. The BDD of good orderings and bad orderings are shown in figure 5 and figure 6 respectively.

5. Conclusion

A method for evaluating the reliability via BDD by taking three different type of variable ordering has been proposed in this paper. We found that the results (reliability) are same by applying the different variable ordering. We also found that the size of the BDD (i.e. the number of non-terminal nodes) depends on the variable ordering. But we can say that at least one optimal ordering may exist for finding the reliability of a given CCN by using BDD. So we use optimal ordering for finding the reliability of a CCN by using BDD or OBDD or MBDD. Because optimal ordering generates minimum size BDD, which can use less memory to store and less complexity. We might also use good ordering for finding the reliability of the CCN. In future we

will try to develop a new algorithm for finding all optimal ordering on a given network. This algorithm will work on both the networks (Directed and Undirected networks).

REFERENCES

- [1] C. Lucet and J.-F. Manouvrier, “Exact methods to compute network reliability”, in *Statistical and Probabilistic Models in Reliability*, D. C. Ionescu and N. Limnios, Eds. Birkhauser Boston, pp. 279–294, 1999.
- [2] M. O. Ball, “Computational complexity of network reliability analysis: An overview”, *IEEE Trans. Reliability*, vol. R-35, no. 3, pp. 230–239, 1986.
- [3] J. S. Provan, “The complexity of reliability computations on planar and acyclic graphs”, *SIAM J. Computing*, vol. 15, no. 3, pp. 694–702, 1986.
- [4] M. O. Locks, “A minimizing algorithm for sum of disjoint products”, *IEEE Trans. Reliability*, vol. R-36, no. 4, pp. 436–445, 1987.
- [5] S. Hariri and C. S. Raghavendra, “SYREL: A symbolic reliability algorithm based on path and cut set methods”, *IEEE Trans. Computers*, vol. C-36, no. 10, pp. 1224–1232, 1987.
- [6] S. H. Ahmad, “Simple enumeration of minimal cut sets of acyclic directed graph”, *IEEE Trans. Reliability*, vol. R-27, no. 5, pp. 484–487, 1988.
- [7] M. S. Choi and C. H. Jun, “Some variant of polygon-to-chain reductions in evaluating reliability of undirected network”, *Microelectron. Reliab.*, vol. 35, no. 1, pp. 1–11, 1985.
- [8] J. P. Gadani, “System effectiveness evaluation using star and delta transformations”, *IEEE Trans. Reliability*, vol. R-30, no. 1, pp. 43–47, 1981.
- [9] O. Theologou and J. Carlier, “Factoring and reductions for networks with imperfect vertices”, *IEEE Trans. Reliability*, vol. R-40, pp. 210–217, 1991.
- [10] A. Satyanarayana and M. K. Chang, “Network reliability and the factoring theorem”, *Networks*, vol. 13, pp. 107–120, 1983.
- [11] R. K. Wood, “A factoring algorithm using polygon-to-chain reductions for computing K-terminal network reliability”, *Networks*, vol. 15, pp. 173–190, 1985.
- [12] B. Akers, “Binary decision diagrams”, *IEEE Trans. Computers*, vol. C-27, pp. 509–516, 1978.
- [13] R. E. Bryant, “Symbolic Boolean manipulation with ordered binary-decision diagrams”, *ACM Computing Surveys*, vol. 24, no. 3, pp. 293–318, 1992.
- [14] O. Coudert and J. C. Madre, “Implicit and incremental computation of primes and essential primes of Boolean functions”, in *Proc. of the 29th ACM/IEEE Design Automation Conference*, 1992, pp. 36–39.
- [15] A. Rauzy, “New algorithms for fault tolerant trees analysis”, *Reliability Engineering and System Safety*, vol. 5, no. 59, pp. 203–211, 1993.
- [16] A. Rauzy, “A new methodology to handle Boolean models with loops”, *IEEE Trans. Reliability*, vol. R-52, no. 1, pp. 96–105, 2003.
- [17] H. Imai, K. Sekine, and K. Imai, “Computational investigations of all terminal network reliability via BDD”, *IEICE Transactions on Fundamentals*, vol. E82-A, no. 5, pp. 714–721, 1999.
- [18] X. Zang, H. Sun, and K. S. Trivedi, “A BDD-based algorithm for reliability Graph Analysis”, *Technical Report* [Online]. Available: <http://www.ee.duke.edu/~hairong/workinduke/relgrap>.
- [19] F. Yeh, S. Lu, and S. Kuo, “OBDD-based evaluation of k-terminal network reliability”, *IEEE Trans. Reliability*, vol. R-51, no. 4, pp. 443–451, 2002.
- [20] Manoj Singhal, R. K. Chauhan, Girish Sharma, “Use of Modified Binary Decision Diagrams in Reliability Evaluation of a Directed Computer Communication Network”, *The Icfai University Journal of Computer Sciences*, Vol. III No. 3, pp. 22–30, July 2009.

- [21] Manoj Singhal, R. K. Chauhan, Girish Sharma, "A New Optimal Approach for evaluating the size of BDD (Binary Decision Diagram) for calculating the Reliability of a CCN (Computer Communication Network)", International Journal of Advanced Networking and Applications, Vol. 1, issue 4, pp. 230-235, Jan-Feb 2010.
- [22] S. Kuo, S. Lu, F. Yeh, "Determining terminal pair reliability based on edge expansion diagrams using OBDD", IEEE Trans. Reliability, Vol. 48, no. 3, pp. 234-246, 1999.
- [23] Manoj Singhal, R. K. Chauhan, Girish Sharma, "Effects of Variable Ordering on Binary Decision Diagrams for Computation of Reliability of a Computer Communication Network" Journal of Computer Science (accepted).
- [24] S. J. Friedman and K. J. Supowit, "Finding the optimal variable ordering for binary decision diagrams", in *Proc. 24th ACM/IEEE Conference on Design Automation*, 1987, pp.348-356.
- [25] S. J. Friedman and K. J. Supowit, "Finding an optimal variable ordering for binary decision diagrams", *IEEE Trans. Computers*, vol. C-39, no. 5, pp. 710-713, 1990.
- [26] Manoj Singhal, R. K. Chauhan, Girish Sharma, "An alternate approach to compute the reliability of a computer communication network using Binary Decision Diagrams", Communications in Computer and Information Science, pp. 160-170, IC3 2010, Springer-Verlag Berlin Heidelberg 2010.

Mr. Manoj Singhal is currently working as an Associate Professor, Department of Computer Application (MCA) in ABES Engineering College, Ghaziabad (U.P.).



Prof. (Dr.) R. K. Chauhan is currently working as a Chairman, Department of Computer Science and Application in Kurukshetra University, Kurukshetra. He has published more than 70 research papers in International/National Journal/Conference. He has guided 5 Ph.D. His research interest includes Data Mining, DBMS and Algorithm.



Dr. Girish Sharma is currently working as an Asso. Professor, Department of Computer Application in Bhai Parmanand Institute of Business Studies, New Delhi. He has published more than 15 research papers in International/National journal / Conference. His research interest includes Mathematical Modelling, Distributed Systems and Parallel Architecture.

