

A New Approach for Finding the various Optimal Variable Ordering to Generate the Binary Decision Diagrams (BDD) of a Computer Communication Network

Manoj Singhal

Asso. Prof., Deptt. of IT
AIMT, 49 Knowledge Park III
Greater Noida (U.P.), India

Dr. Girish Sharma

Asso. Prof., Deptt. of MCA
Bhai Parmanand Institute of
Business Studies
New Delhi, India

Dr. R. K. Chauhan

Prof. Deptt. of Computer
Science & Application
Kurukshetra University,
Kurukshetra, Haryana, India

ABSTRACT

In this paper we have adopted a new approach for finding the various optimal ordering to generate the binary decision diagrams of a computer communication network. We have shown that these binary decision diagrams are of minimum size and take same time to generate. If two binary decision diagrams have the same size and representing the same Boolean function, then these binary decision diagrams are known as dual binary decision diagrams, because they are dual of each other. We have also shown that the reliability obtained from these dual binary decision diagrams is equal by applying Shannon's decomposition.

Key Words Binary Decision Diagrams (BDD), Directed Acyclic Graph (DAG), Computer communication Network (CNN), Dual Binary Decision Diagrams (DBDD), Ordered Binary Decision Diagrams (OBDD).

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 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] have shown how to functionally construct the corresponding BDD. An alternate approach was given in [18] to compute BDD based network reliability.

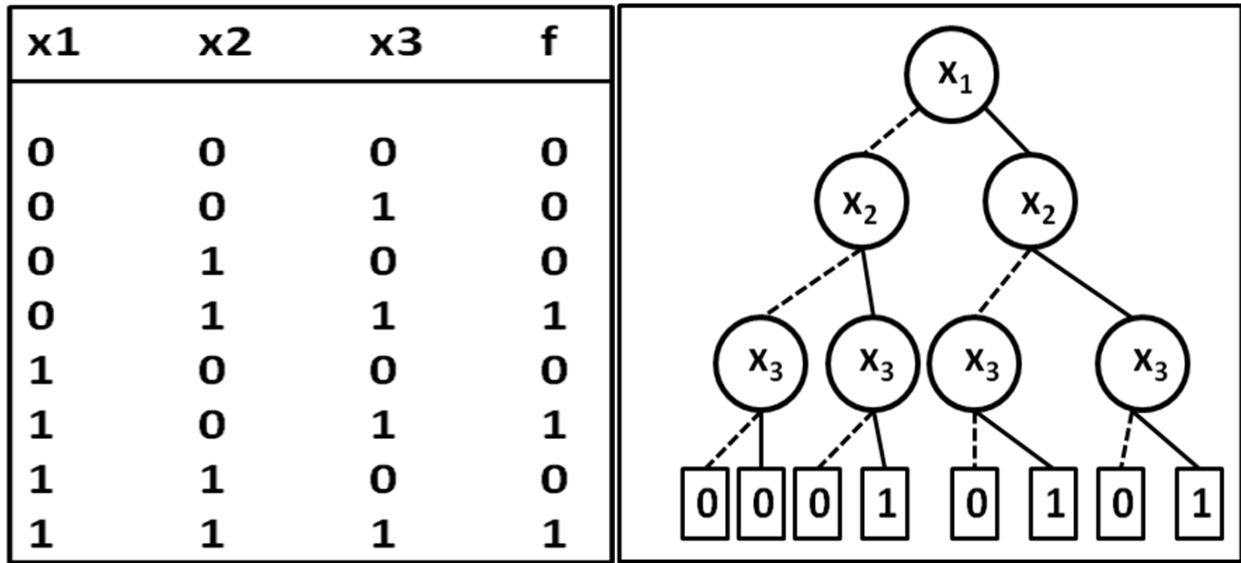


Fig 1: Truth table of a Boolean function f and its corresponding decision tree

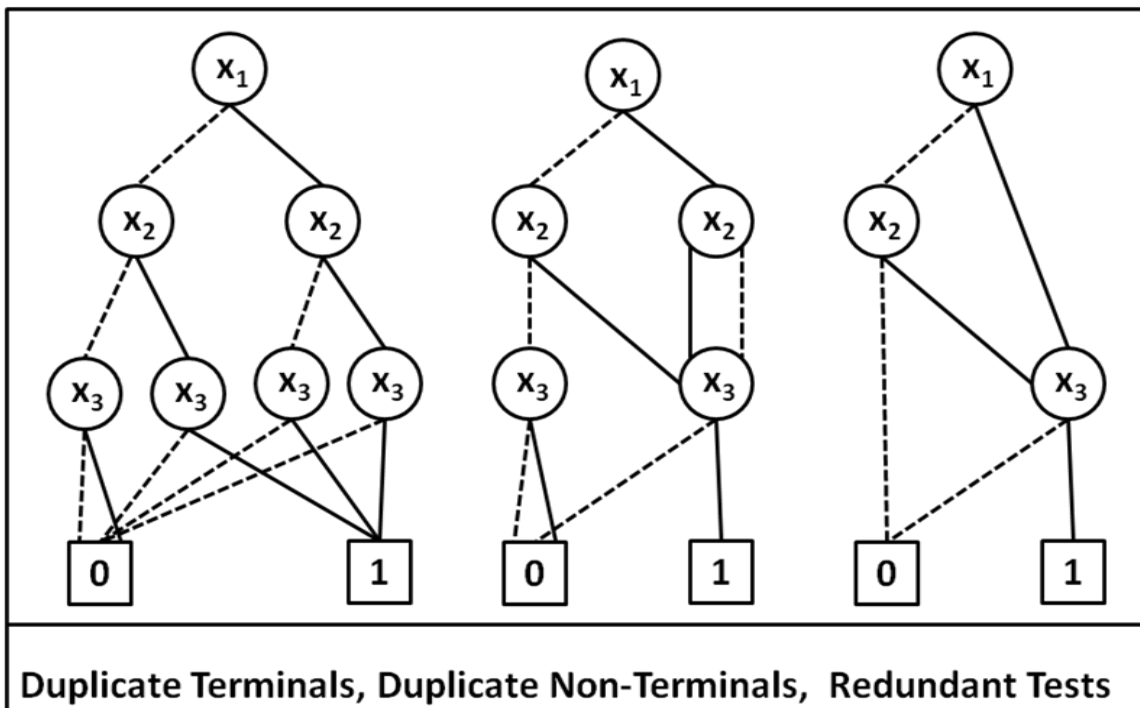


Fig 2: Reduction Process of the decision tree

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 $var(u)$, 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. $f_{x_i=1}$; while the node linked by the 0-edge represents the Boolean expression when $x_i = 0$, i.e. $f_{x_i=0}$. The two outgoing edges are given by two functions $low(u)$ and $high(u)$.

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 $var(u) = var(v)$, $low(u) = low(v)$, and $high(u) = high(v)$, 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 $low(v) = high(v)$, then delete v, and redirect all incoming arcs to $low(v)$.

The shrinking process is shown in figure 2.

The final reduced diagram of the decision tree is shown in figure 3.

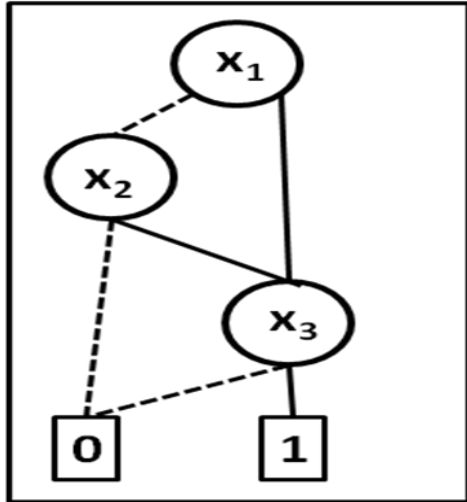


Fig 3: Reduced BDD

2.1 Ordered Binary Decision Diagrams

For an ordered BDD (OBDD), we impose a total ordering $<$ 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].

3. EFFECTS OF VARIABLE ORDERING

A particular sequence of variables is known as a variable ordering. It has been observed that the size of the BDD strongly depends on the ordering of variables [20]. The size of BDD means the total number of non-terminal nodes in the BDD and number of nodes in a particular level [21]. An ordering is said to be optimal if it generates the minimum size BDD [22]. The given function f has three different variables, so its BDD must contain at least three non terminal vertices. We have generated the different BDD of the given function by taking the ordering $x_2 < x_1 < x_3$, $x_3 < x_1 < x_2$, $x_3 < x_2 < x_1$, $x_1 < x_3 < x_2$ and $x_2 < x_3 < x_1$. The reduced ordered binary decision diagrams are shown in figure 4(a), 4(b), 4(c), 4(d) and 4(e) respectively.

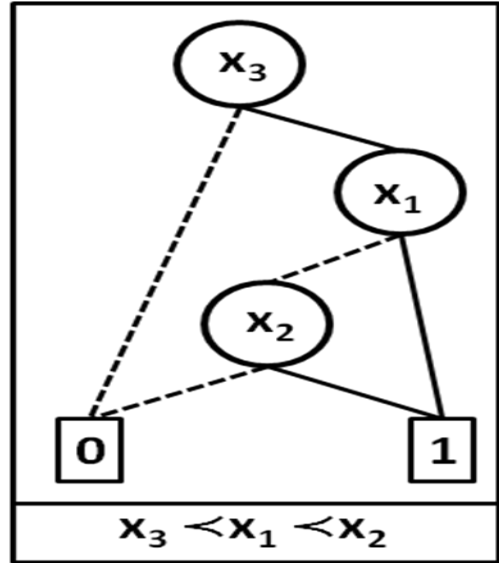


Fig 4(b): Ordered BDD

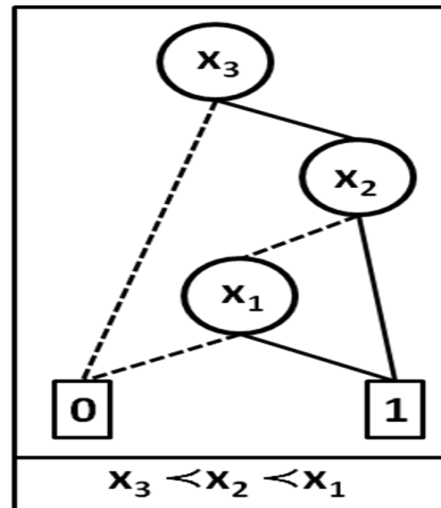


Fig 4(c): Ordered BDD

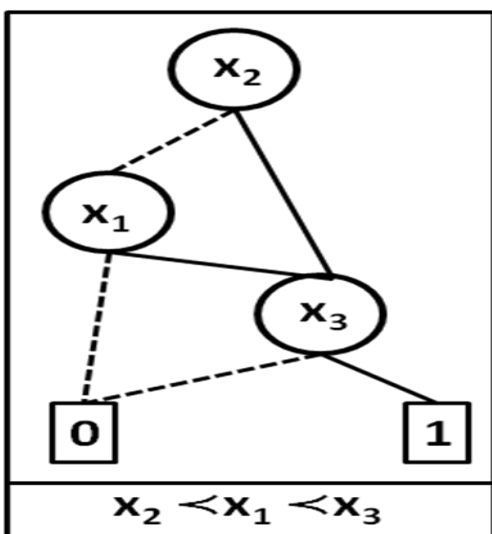


Fig 4(a): Ordered BDD

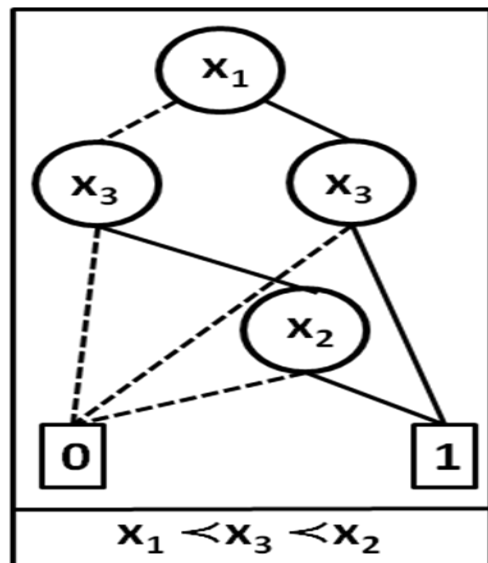


Fig 4(d): Ordered BDD

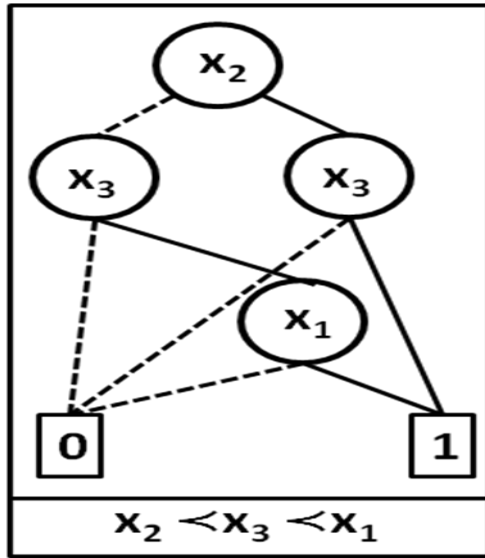


Fig 4(e): Ordered BDD

We have constructed different BDD of the given function by taking all possible different variable ordering. There are three types of variable ordering (Optimal, Good and Bad) to generate different BDD of the given network [23]. We have found that for the variable ordering $x_1 < x_2 < x_3$, $x_2 < x_1 < x_3$, $x_3 < x_1 < x_2$, and $x_3 < x_2 < x_1$, the number of non-terminal vertices in the BDD is three. Since the given Boolean function has three different variables therefore its BDD must contain at least three non terminal vertices. So we can say that the size of the BDD is minimal for these variable ordering. These orderings are known as optimal ordering and these binary decision diagrams are known as Dual Binary Decision Diagrams or modified binary decision diagram, as they represent the same Boolean function and having the same size [24].

We have also found that for the ordering $x_1 < x_3 < x_2$ and $x_2 < x_3 < x_1$ the size of the BDD is maximal. These BDD contains four non-terminal vertices, but they also represent the same Boolean function.

Thus we can conclude that more than one optimal ordering is possible for finding the minimal size BDD. We have applied the same concept to find various optimal ordering to generate the BDD of a CCN.

4. 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 [25].

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 terminal reliability and all terminal reliability are the particular case of K-terminal reliability [26].

5. EXAMPLE NETWORK

Let us take an example of a directed network G (V, E) with single source S and single sink T as shown in figure 5.

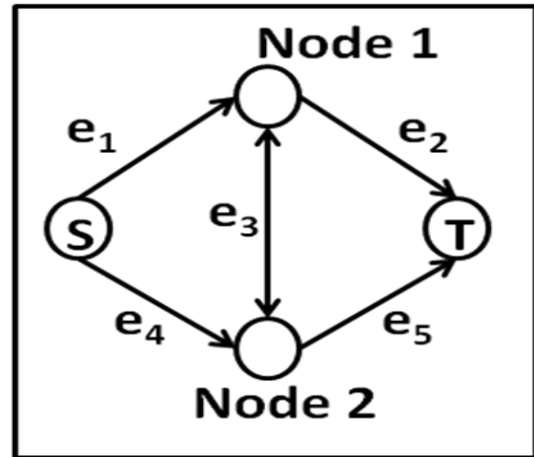


Fig 5: A Directed Network

The network has four nodes and five edges. The middle edge of the network is bi-directional; therefore the given network is an example of bi-directional network. Bi-directional networks are those networks in which at least one edge is bi-directional.

The network has four min-paths. These are

$$H_1 = \{e_1, e_2\}, H_2 = \{e_1, e_3, e_5\}, H_3 = \{e_4, e_5\} \text{ and } H_4 = \{e_4, e_3, e_2\}$$

Let H_1, H_2, \dots, H_n be the n minpaths from source to sink in a network then the network connectivity function C can be represented as a logical OR of its min-paths.

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

So the point to point reliability is:

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

So the network connectivity of our network can be expressed as

$$C_{1-4} = e_1 e_2 \cup e_1 e_3 e_5 \cup e_4 e_5 \cup e_4 e_3 e_2 \quad (2)$$

X. Zang [27] has found the minimal BDD of the given directed network. This minimal BDD is shown in figure 6. He has shown that the minimal BDD of the given network consists of 8 non-terminal nodes. The diagram of minimal BDD and its probability computation is shown in figure 6.

Here we have adopted a new approach to find out various other minimal BDD of the given network by changing its variable ordering. The BDD based network reliability computation involves three main steps.

- (i) Ordering the network edges by using an optimal variable ordering heuristic.
- (ii) Generate BDD from the probabilistic graph of the network with the help of network connectivity function.
- (iii) Evaluate the network reliability recursively from the BDD by applying Shannon's decomposition.

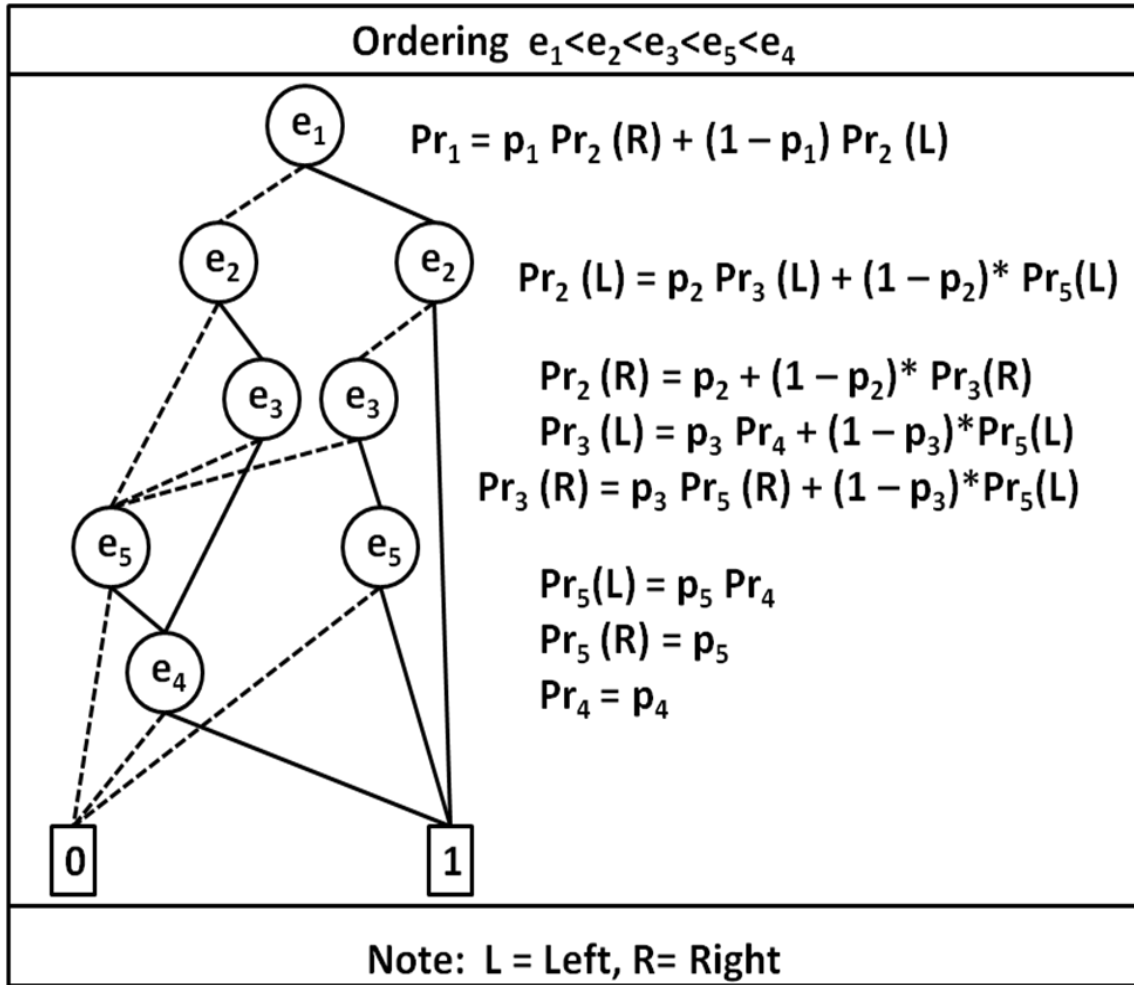


Fig 6: BDD and its Probability Computation

5.1 Heuristic Approach:

The heuristic approach is given below:

1. Traverse the graph from source S to sink T. Find all the min-paths from source to sink.

These are $H_1 = \{e_1, e_2\}$, $H_2 = \{e_4, e_5\}$, $H_3 = \{e_1, e_3, e_5\}$ and $H_4 = \{e_4, e_3, e_2\}$.

2. Check whether these paths are disjoint or not. If all the paths are disjoint then we can select any one of the disjoint paths. Then select the second, third and so on.

3. If all min-paths are not disjoint then find only those min-paths which are disjoint. We have found that the min-paths H_1 and H_2 are disjoint. Now we can move from source S via min-path H_1 or H_2 . To choose either H_1 or H_2 , we analyse these min-paths. If we select min-path H_1 , then the next node is node 1. If we select min-path H_2 , then the next node is node 2.

Since the degree of the node 1 and node 2 are same then we can move from source S via min-path H_1 or H_2 .

To choose either H_1 or H_2 , we will give preference to min-path H_2 then middle edge e_3 and then min-path H_1 , we can take the ordering either $e_1 < e_2$ or $e_2 < e_1$.

After applying the above heuristic approach, we have found that the various other variable orderings are $e_4 < e_5 < e_3 < e_2 < e_1$, $e_4 < e_5 < e_3 < e_1 < e_2$ and $e_1 < e_2 < e_3 < e_4 < e_5$.

5.2 Connectivity Function:

We have already developed the connectivity function of the given network as shown in equation (2).

5.3 Shannon’s Decomposition:

The Shannon’s decomposition is defined as follows:

$$Pr\{F\} = p_1 Pr\{F_{x_1=1}\} + (1 - p_1) Pr\{F_{x_1=0}\}$$

$$= Pr\{F_{x_1=0}\} + p_1 (Pr\{F_{x_1=1}\} - Pr\{F_{x_1=0}\})$$

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.

The BDD and its probability computation for the ordering $e_4 < e_5 < e_3 < e_2 < e_1$, $e_4 < e_5 < e_3 < e_1 < e_2$ and $e_1 < e_2 < e_3 < e_4 < e_5$ are shown in figure 7, figure 8 and figure 9 respectively.

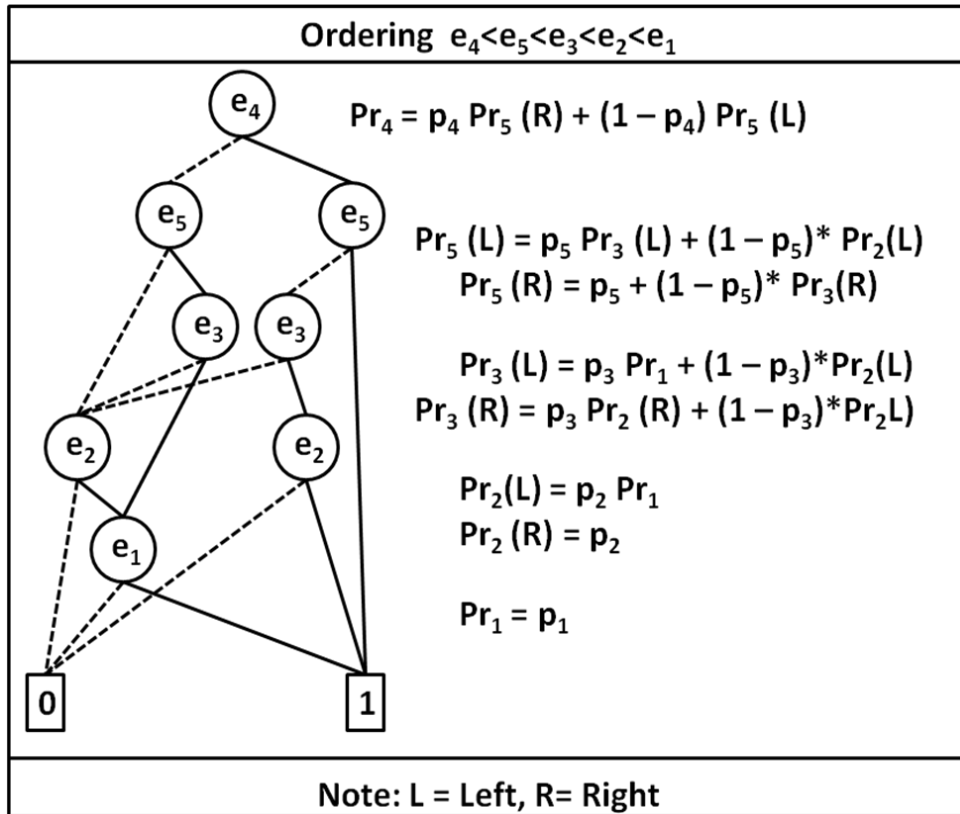


Fig 7: Minimal BDD and its Probability Computation

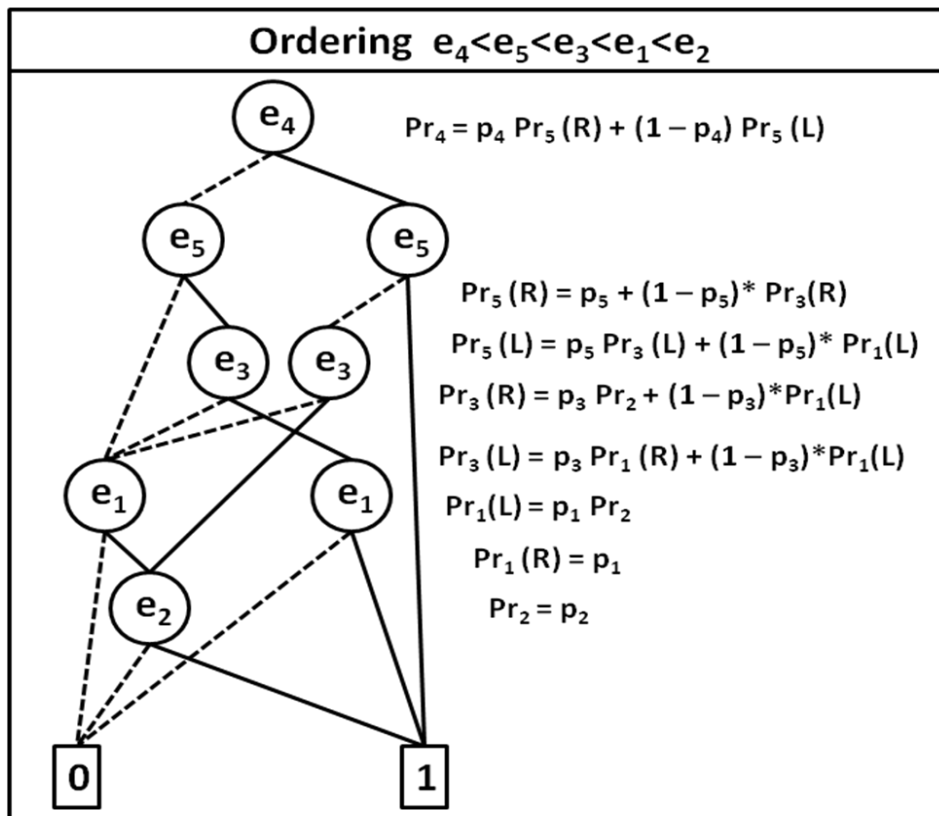


Fig 8: Minimal BDD and its Probability Computation

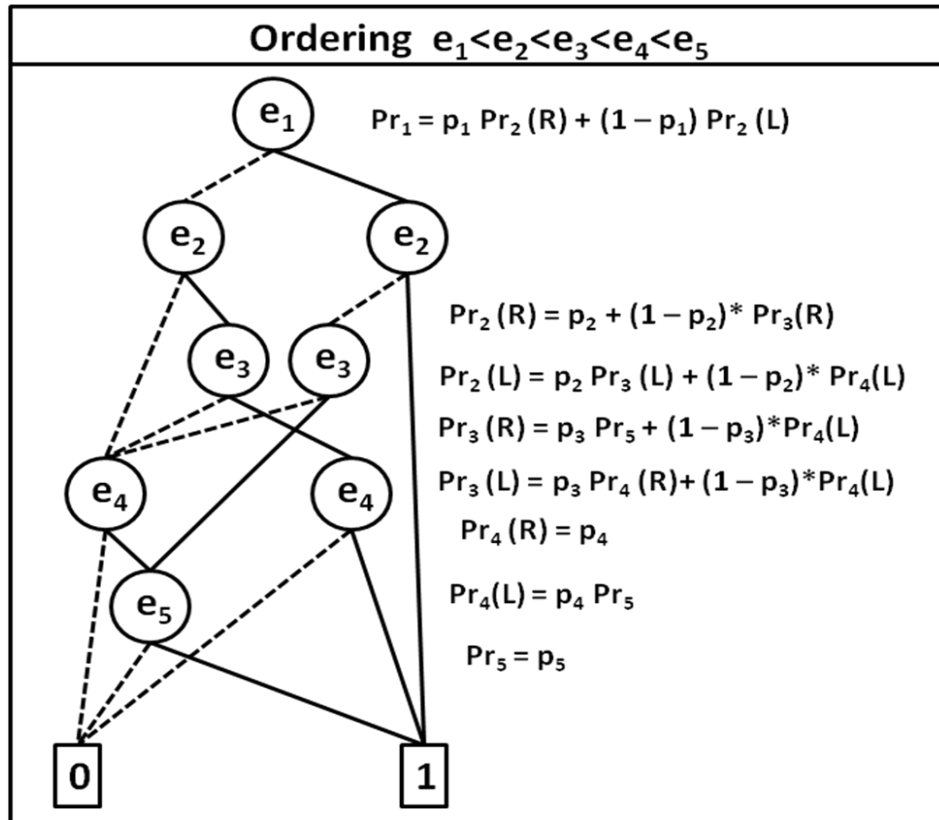


Fig 9: Minimal BDD and its Probability Computation

Here we have found that the BDD (figure 7, 8 and 9) of the given network also has 8 non-terminal nodes. Therefore these minimal BDD are known as dual binary decision diagram (DBDD) or modified binary decision diagrams because they have the same size as shown by Zang in [27].

Therefore the various other optimal variable orderings are $e_4 \prec e_5 \prec e_3 \prec e_2 \prec e_1$, $e_4 \prec e_5 \prec e_3 \prec e_1 \prec e_2$ and $e_1 \prec e_2 \prec e_3 \prec e_4 \prec e_5$.

Here we have also found that

$$Pr_1 \text{ (for figure 6)} = Pr_1 \text{ (for figure 9)}$$

$$Pr_4 \text{ (for figure 7)} = Pr_4 \text{ (for figure 8)}$$

$$\text{and } Pr_1 \text{ (for figure 6 and 9)} = Pr_4 \text{ (for figure 7 and 8)}$$

6. EXPERIMENTAL RESULT

Our program is written in the C language and computations are done by using a Pentium 4 processor with 1 GB of RAM. The computation speed heavily depends on the variables ordering because 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. We have constructed only few DBDD of the given CCN and compute the reliability of the given CCN by using these DBDD. We have found that the reliability obtained in each case by using DBDD is same. We have also found that the size of all DBDD is same. The time taken to generate the above BDD is same. We have found that the reliability of the given CCN is same in

all the cases. We also found that the size of the BDD is minimal for several orderings called the optimal ordering.

7. CONCLUSIONS

A method for generating the DBDD of a computer communication network has been proposed in this paper. We have also evaluated the reliability via these DBDD by taking different variables ordering. We found that the result (reliability) is same by all the DBDD. We have also found that the size of the BDD (i.e. the total number of nodes and number of nodes in a particular level) are same in all cases. Our future work will focus on computing other kinds of reliability and reusing the BDD structure in order to optimize design of network topology with augmentation.

8. 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. Carrier : 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 BDDs : IEICE Transactions on Fundamentals, vol. E82-A, no. 5, pp. 714–721, 1999.
- [18] 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.
- [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 : Effects of Variable Ordering on Binary Decision Diagrams for Computation of Reliability of a Computer Communication Network : Journal of Computer Science, Vol. 4, issue 6, pp. 1837-1845, Sep-Oct 2010.
- [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. 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.
- [23] Manoj Singhal, R. K. Chauhan, Girish Sharma : Network Reliability Computation using Different Binary Decision Diagrams: International Journal of Distributed and Parallel Systems, Vol. 1, No. 1, pp. 82-91, September 2010.
- [24] 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.
- [25] 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.
- [26] G. Hardy, C. Lucet, and N. Limnios : Computing all-terminal reliability of stochastic networks with binary decision diagrams : in Proc. 11th International Symposium on Applied Stochastic Models and Data Analysis, 2005, pp. 1468-1473.
- [27] 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>.

9. AUTHORS PROFILE

Mr. Manoj Singhal is currently working as an Associate Professor, Department of Information Technology in Accurate Institute of Management and Technology, Greater Noida (U.P.), India.

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 23 research papers in International/National journal / Conference. His research interest includes Mathematical Modelling, Distributed Systems and Parallel Architecture.

Prof. (Dr.) R. K. Chauhan is currently working as a Professor, 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 7 Ph.D. His research interest includes Data Mining, DBMS and Algorithm.