Effects Of Variable Ordering On Binary Decision Diagrams For Computation Of Reliability Of A Computer Communication Network

[1]Manoj Singhal, [2]R. K. Chauhan, [3]Girish Sharma

Abstract

In this paper we compute the reliability of a computer communication network (CCN) by using different Binary Decision Diagrams (BDD). Here we take several orderings to generate different BDD of the given CCN and then compute network reliability by these different BDD. It is observed experimentally that the results (Reliability) of applying Classical Inclusion-exclusion principle on the given network are the same as obtained by applying Shannon's decomposition on different BDD.

Keywords : Binary Decision Diagrams (BDD), Directed Acyclic Graph (DAG), Computer communication Network (CNN)

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." [1]. There are so many exact methods for computation of network reliability. 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 who associates with each edge of G a pair of vertices 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 [14]. The same thing is observed for planar graphs [13]. 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-

[1]Asso. Prof., Dept. of MCA, Academy of Business and Engineering Sciences, Ghaziabad. Email : manoj_singhal@rediffmail.com

[2]Prof. & Chairman, Dept. Of Computer Science and Applications, Kurukshetra University, Kurukshetra. Email : rkc.dcsa@gmail.com

[3]Asstt. Prof., Dept. of MCA, Bhai Parmanand Institute Of Business Studies, Delhi. Email : gkps123@gmail.com

disjoint events. Numerous works about this kind of methods have been presented in literature, [22, 15, and 21]. 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 [16] and delta-to-star reductions [12]. 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 [18] for dense networks. Satyanarayana & Chang [5] and Wood [20] have shown that the factoring algorithms with reductions are more efficient at solving this problem than the classical path or cut enumeration methods.

This paper is organized as follows. First we will illustrate the preliminaries of BDD in Section II. In Section III, we define the network reliability and constructing the BDD of the given network with different variable ordering. Finally we draw some conclusion.

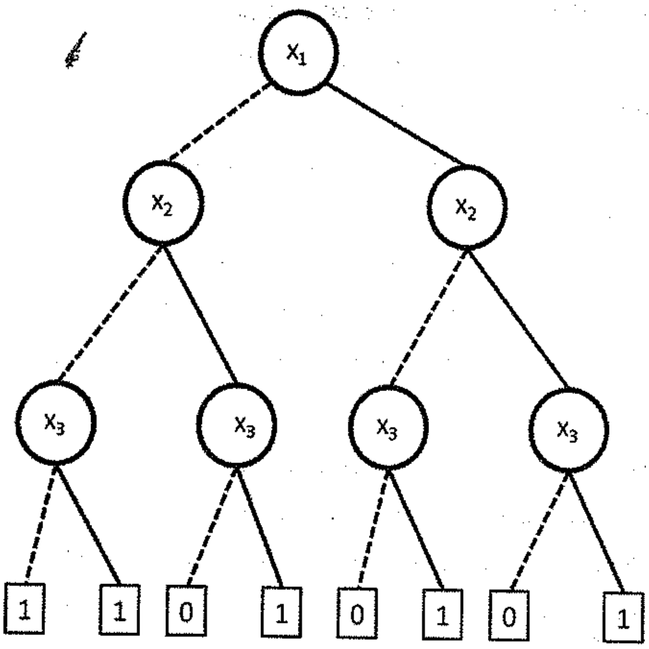
2. Binary Decision Diagram

Akers [6] first introduced BDD to represent Boolean functions i.e a BDD is a data structure used to represent a Boolean Function. Bryant 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 [6, 19].The Shannon decomposition for a Boolean function is defined as follows:

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

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 [17] found BDD usefulness in reliability analysis which was further extended by Rauzy [3, 4]. They are specially used to assess fault trees in system analysis. In the network reliability framework, Sekine & Imai [10], and Trivedi [26] have shown how to functionally construct the corresponding BDD. Sink nodes are labeled either with 0, or with 1, representing the two corresponding constant expressions. Each internal node u is labeled 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. $f_{xi=1}$; while the node linked by the 0-edge represents the Boolean expression when xi = 0, i.e. fxi=0. The two outgoing edges are given by two functions low (u) and high (u). Figure 1 shows the truth table of a Boolean function f and its corresponding Shannon tree.

x1	x2	x3	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



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

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

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

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.

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.

Delete Duplicate Tests : If non terminal vertex v has low(v) = high(v), then delete v, and redirect all incoming arcs to low(v).

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

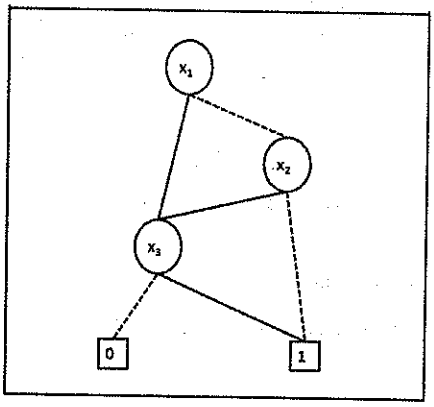


Figure 2 : Reduced BDD of the aboved decision tree

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 [2, 11].

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 each other, this is K terminal reliability [8].

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 [9].

Let us take an example of a directed network G(V, E) with single source and single sink as shown below Here we use path enumeration method to find the reliability of the example network.

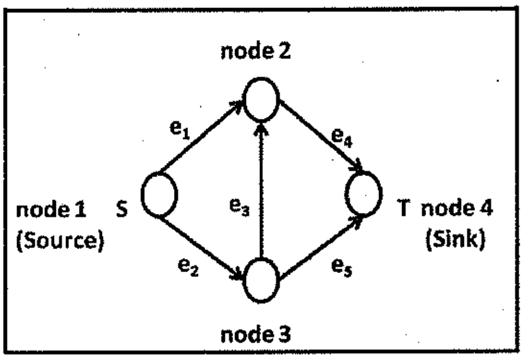


Figure 3 : A Directed Network

The network has three minpaths.

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

Let $H_1, H_2, \cdots H_n$ 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 minpaths.

$$C = H_1 U H_2 \cdots U H_i \cdots U H_n$$

So the point to point reliability is:

$$R_s = Pr\{C\} = Pr\{ H_1 U H_2 \cdots U H_i \cdots U H_n\} \qquad -(1)$$

and the network connectivity of our network can be expressed as

$$C_{1-4} = e_1 e_4 U e_2 e_3 e_4 U e_2 e_5 \qquad \_\_(2)$$

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 Φ(G) into a probability expression is to use Poincare's theorem, also called inclusion-exclusion method[27]. 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(Φ(G)) can be expressed as follows:

$$E(H_1 + H_2) = E(H_1) + E(H_2) - E(H_1 . H_2)$$

Poincare's formula for m minpaths :

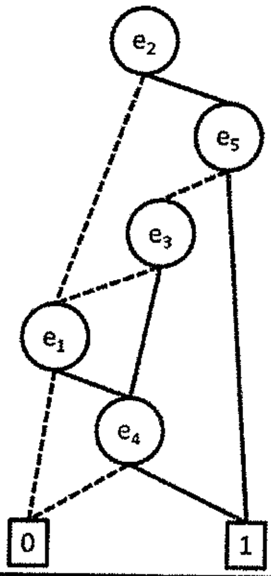
$$E(\Phi(G)) = \sum_{1 \le i \le m} E(H_i) - \sum_{1 \le j \le k \le m} E(H_j . H_k) + \cdots + (-1)^{m+1} E((H_1 . H_2 . H_3 \ldots H_m)$$

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

$$R_{1-4} = Pr\{C_{1-4}\} = p_1p_4 + p_2p_3p_4 + p_2p_5 - p_1p_2p_3p_4 - p_2p_3p_4p_5 - p_1p_2p_4p_5 + p_1p_2p_3p_4p_5 \qquad --- (3)$$

In this paper, we take several ordering to generate different BDD of the given network and 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).

We apply the Shannon's decomposition to the Boolean connectivity function of the directed network expressed as the union of the minpaths in Formula(2).



$$pr_2 = pr_1 + p_2 (pr_5 - pr_1)$$

$$pr_5 = pr_3 + p_5 (1 - pr_3)$$

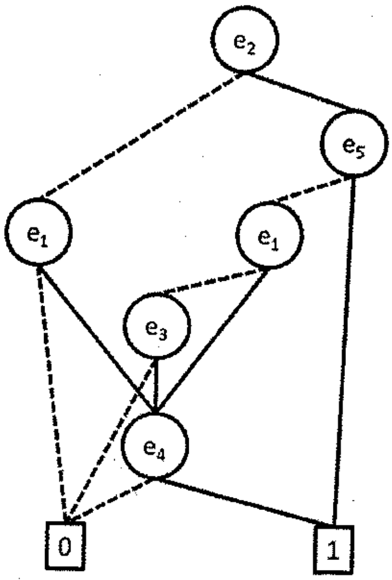
$$pr_3 = pr_1 + p_3 (p_4 - pr_1)$$

$$pr_1 = 0 + p_1 p_4$$

$$pr_4 = p_4$$

$$Pr_2 = p_1p_4 + p_2p_3p_4 + p_2p_5 - p_1p_2p_3p_4 - p_2p_3p_4p_5 - p_1p_2p_4p_5 + p_1p_2p_3p_4p_5$$

Figure 4 (A) : BDD representation of the given Network with ordering $e_2 < e_5 < e_3 < e_1 < e_4$ and its Reliability computation



$$pr_2 = pr_1 + p_2 (pr_5 - pr_1)$$

$$pr_5 = pr_1 + p_5 (1 - pr_1)$$

$$pr_1 = pr_3 + p_1 (pr_4 - pr_3)$$

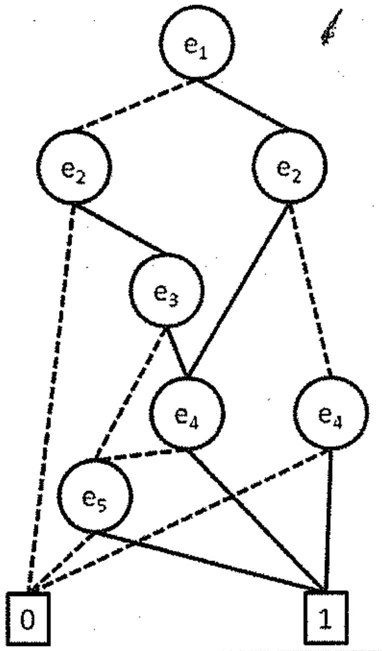
$$pr_1 = 0 + p_1 (pr_4 - 0)$$

$$pr_3 = 0 + p_3 (p_4 - 0)$$

$$pr_4 = p_4$$

$$Pr_2 = p_1p_4 + p_2p_3p_4 + p_2p_5 - p_1p_2p_3p_4 - p_2p_3p_4p_5 - p_1p_2p_4p_5 + p_1p_2p_3p_4p_5$$

Figure 4 (B) : BDD representation of the given Network with ordering $e_2 < e_5 < e_1 < e_3 < e_4$ and its Reliability computation



$pr_1 = pr_2 + p_1 (pr_2 - pr_2)$

$pr_2 = 0 + p_2 (pr_3 - 0)$

$pr_2 = pr_4 + p_2 (pr_4 - pr_4)$

$pr_3 = pr_5 + p_3 (pr_4 - pr_5)$

$Pr_4 = pr_5 + p_4 (1 - pr_5)$

$pr_4 = p_4$

$pr_5 = p_5$

$$Pr_1 = p_1p_4 + p_2p_3p_4 + p_2p_5 - p_1p_2p_3p_4 - p_2p_3p_4p_5 - p_1p_2p_4p_5 + p_1p_2p_3p_4p_5$$

Figure 4 (C) : BDD representation of the given Network with ordering $e_1 < e_2 < e_3 < e_4 < e_5$ and its Reliability computation

The computation of the probability of the BDD of figure 3 can be calculated recursively by resorting to the Shannon decomposition.

$Pr\{F\} = p_l Pr\{Fx_l = 1\} + (1 “ p1) Pr\{Fx_l = 0\} = Pr\{Fx_l = 0\} + p_l(Pr\{Fx_l = 1\} ” Pr\{Fx_l = 0\})$ ---- (4)

where p_l is the probability of the Boolean variable x_l to be true and $(1" p_l)$ is the probability of the Boolean variable x_l to be false.

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 [10, 22, 30]. The Boolean connectivity expression (2) and the computation of the probability of the BDD are shown in figure 4 (A), 4 (B) and 4 (C).

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. The size of BDD means the total number of nodes in the BDD and number of nodes in a particular level. There are more than 100 variables ordering are possible for constructing the different BDD of the given CCN. We have constructed several 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. Out of these several BDD we have shown only three different BDD by taking three variables ordering (as shown in the above figures 4 (A), 4 (B) & 4 (C)) for computing the reliability of the given CCN with these three different BDD. We have found that the reliability of the given CCN is same in all the cases and is equal to the reliability of the given CCN obtained by inclusion-exclusion formula. We also found that the size of the BDD is minimum only for a particular ordering called the optimal ordering or good ordering (as shown in figure 4 (A)). Other orderings are called the bad orderings (as shown in figure 4 (B) and 4 (C)) but reliability results are same. The time taken to construct the BDD and compute the reliability is also minimum in optimal ordering case in compare to the bad ordering case.

Hence $R_{1\text{-}4} = p_1p_4 + p_2p_3p_4 + p_2p_5$ “ $p_1p_2p_3p_4$ “ $p_2p_3p_4p_5$ “ $p_1p_2p_4p_5 + p_1p_2p_3p_4p_5$

4. Conclusion

A method for evaluating the reliability via BDD by taking three different variables ordering has been proposed in this paper. We find 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 nodes) heavily depend on the variables ordering. Our future work will focus on computing other kinds of reliability and reusing the BDD structure in order to optimize design of network topology.

References

[1] Institute of Electrical and Electronics Engineers, *"IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries"*, New York, NY, 1990.

[2] B. Akers, *"Binary decision diagrams"*, IEEE Trans. Computers, Vol. C-27, PP. 509–516, Jun. 1978.

[3] R. E. Bryant, *"Symbolic Boolean manipulation with ordered binary-decision diagrams"*, *ACM Computing Surveys*, Vol. 24, No. 3, PP. 293–318, 1992.

[4] M. O. Ball, *"Complexity of network reliability computations Networks"*, Vol. 10, PP. 153–165, 1980.

[5] M. O. Ball, *"Computational complexity of network reliability analysis: An overview"*, *IEEE Trans. Reliability*, Vol. R-35, No. 3, PP. 230–239, Aug. 1986.

[6] H. Andersen, *"An introduction to binary decision diagrams"*, Oct. 1997, Lecture Notes for 49285 Advanced Algorithm E97.

[7] A. Rauzy, *"New algorithms for fault tolerant trees analysis, Reliability Engineering and System Safety"*, Vol. 5, No. 59, PP.203–211, 1993.

[8] 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,PP. 36–39, IEEE Computer Society Press, 1992.

[9] O.Coudert and J.C.Madre, *"A new method to compute prime and essential prime implicants of Boolean functions"*, in Proc. of MIT VLSI Conference, March 1992.

[10] 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.

[11] J.S.Provan, *"The complexity of reliability computations on planar and acyclic graphs"*, SIAM J. Computing, Vol. 15, No. 3, PP. 694–702, 1986.

[12] 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, May 2005.

[13] 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, May 1999.

[14] S.H.Ahmad, *"Simple enumeration of minimal cutsets of acyclic directed graph"*, IEEE Trans. Reliability, Vol. R-27, No. 5, PP. 484–487, Dec. 1988.

[15] M.O.Locks, *"A minimizing algorithm for sum of disjoint products"*, IEEE Trans. Reliability, Vol. R.36, No.4, PP.436–445, Oct. 1987.

[16] S.Hariri and C.S. Raghavendra, *"SYREL: A symbolic reliability algorithm based on path and cutset methods"*, IEEE Trans. Computers, Vol. C-36, No.10, PP.1224–1232, Oct. 1987.

[17] R. K.Wood, *"A factoring algorithm using polygon-to-chain reductions for computing K-terminal network reliability Networks"*, Vol. 15, PP. 173–190, 1985.

[18] A. Satyanarayana and M. K. Chang, *"Network reliability and the factoring theorem Networks"*, Vol. 13, PP. 107–120, 1983.

[19] O. Theologou and J. Carlier, *"Factoring and reductions for networks with imperfect vertices"*, IEEE Trans. Reliability, Vol. R-40, PP. 210–217, 1991.

[20] J. Carlier and C. Lucet, *"A decomposition algorithm for network reliability evaluation"*, Discrete Applied Mathematics, Vol. 65, PP. 141–156, 1996.

[21] F.M.Yeh, S.K.Lu and S.Y. Kuo, *"OBDD-based evaluation of k-terminal network reliability"*, IEEE Trans. Reliability, Vol. R-51, No. 4, 2002.

[22] 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, PP. 348–356, 1987.

[23] 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, 1995.

[24] J.P.Gadani, *"System effectiveness evaluation using star and delta transformations"*, IEEE Trans. Reliability, Vol. R-30, No. 1, PP. 43–47.

[25] X.Zang, H.Sun and K.Trivedi, *"A BDD-based algorithm for reliability evaluation of phased mission systems"*, IEEE Trans. Reliability, Vol. R-48, No. 1, PP. 50–60, 1999.

[26] K. Odeh, *"New Algorithms for Probabilistic and Logic Analysis of Fault Trees"*, (in in French) PhD-thesis, Université de Technologie de Compiègne, French, 1995.

[27] 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.

[28] A. Rauzy, *"A new methodology to handle Boolean models with loops"*, IEEE Trans. Reliability, Vol. R-52, No.1, PP.96–105, 2003.

[29] Andrea Bobbio,Caterina Ferraris and Roberta Terruggia, *"New Challenges in Network Reliability Analysis"*, Technical Report, TR-INF-2006-11-04-UNIPMN.

[30] 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.

Author's Biography

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 Asstt. 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.