International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-2, July 2019

An Algorithm Based on Heap of Binary Search Tree to Solve Graph Coloring Problem

Ajay Narayan Shukla, Vishal Bharti, M.L. Garg

***Abstract** Graph coloring problem is one of most frequent studied problem in the graph theory due to its uses in different area of applications like simulation of biological networks, communication networks, register allocation and many more. This problem involves the coloring of the vertices of the given a graph G (V, E) with number of available colors in such a manner that adjacent vertices must assign colors different with each other. In this paper we present a hybrid approach to assign the colors to vertices of the given graph that is based on adjacency matrix and search tree data structure. The coloring process for a particular vertex in the graph will done by getting the feasible colors available in the color list. The feasible colors that may be assigned to a vertex, retrieved from the vertex-color binary search tree generated initially for available colors. The proposed solution for the graph coloring problem is efficient in terms of its running time complexity and it will work without affecting its complexity for any kind of graph.*

***Index Terms**: Adjacency matrix, Graph Coloring, hybrid approach, vertex-color binary search tree.*

I. INTRODUCTION

Graph coloring problem is studied mostly due its lot of applications in different area of research. Mathematically a graph G may be defined as a pair of vertices & edges (V, E). The graph coloring problem may be sated as to assign the color to each vertices in the given graph G in a manner that adjacent vertices in the graph must have different colors. During the coloring process of vertices in the graph it is also required to use the available colors in such a manner that total number of colors used is minimum. If the colors that is used in coloring process of a given graph is minimum, it is called chromatic number of the graph. Graph coloring problem is a NP class optimization problem which is mostly studied for improving its running time complexity. This problem is also considered for its solution due its different applications in many real world problems where researchers encode original problem into graph coloring problem and then try to find the efficient solution for the problem. Few applications in which graph coloring used for getting the solution for the problems are crew scheduling [1] in which the researchers have given the solution for the problem based on graph coloring model.

Revised Manuscript Received on 30 July 2019.
***** Correspondence Author
Ajay Narayan Shukla, CSE, DIT University, Dehradun, India.
Dr. Vishal Bharti, CSE, DIT University, Dehradun, India.
Dr. M L Garg, CSE, DIT University, Dehradun, India.

In another problem found in literature [2] the researchers have address the problem of satellite range scheduling and proposed the solution by applying certain heuristics and methodology based on graph coloring problem. In the literature few more problems and solutions based on graph coloring problem related to time tabling reported [3, 4] in which researchers have addressed the problem by mapping the problem domain in the form of graphs & networks and given efficient solution. Similarly the problems of allocation of variables to the registers [5], assignment of frequency in communication networks [6], pattern matching and applications in parallel computing[7]. Algorithms used for solving graph coloring problem may be categorized into two different category on the basis of types of approach used for finding the solution of the problem. These include exact algorithms [8-12] and approximate algorithms that uses construction method [13], local search [14], population based evolutionary algorithm [15-16], hybrid and algorithm based on heuristic [17-18]. The algorithms available in literature are mostly approximate in nature due to its NP hardness and in many applications where transformation of applications into graph coloring problem generates large number of instances requires to be analyzed at intermediate stage of solution and thus requires more effort to get the final solution. Due to this reason there are very few algorithms for solving the graph coloring problem that are exact in nature. The constraints in the graph coloring problem says that no two adjacent vertices can assign the same color but in the algorithms those explore the vertices for the coloring keeping the given constraints only find other implicit constraints and then it requires to handle these newly generated constraints in intermediate stages explicitly and that increases the complexity of the algorithms. In this paper we propose a hybrid approach based on the adjacency matrix representation of the given graph that uses k number of available colors to color the vertices of the graph. This coloring process involves the use of a heap of binary search trees that we are using to find the appropriate color corresponding to the vertices in the graph. The entire coloring process is free from any additional constraints to be handled during the coloring of the vertices and thus reduces the running time of the algorithm. The propose algorithm may be used in any kind of real world application which requires to solve using graph coloring problem without affecting working and running time of the algorithm.

II. LITERATURE REVIEW

There are various approaches available in literature to solve graph coloring problem depending upon the type of applications and type of graphs generated in application problems [19-20]. In one approach that is based on the conflict driven clause learning with backtracking [21] and researchers have given the solution of the k-clorability problem with conflict driven clause learning.

In their proposed methodology researchers have given an exact algorithm for the graph coloring problem with learning by using propositional logic they searched any implicit constraints generated during coloring of the vertices in the graph. Researchers have given the solution of the problem based on genetic algorithm [22] in which they have defined population based parameter in their proposed algorithm and obtained the result after comparing fitness value used. Recently another technique reported in literature in which researchers have given the exact solution for the graph coloring problem [23]. The proposed algorithm uses adjacency list representation of the given graph & researchers created an additional color adjacency list corresponding to every vertices in the given graph. Algorithm searches the color adjacency list that finds those colors for a vertex that cannot assign to vertex and stored those colors in stack. Finally except those colors which is in stack can assign to explored vertex. Researchers have also produces the solution of vertex coloring of the graphs by using Coupled relaxation oscillator based dynamical system that exploits insulator metal transition in Vanadium Dioxide [24]. A DNA based computing model for the vertex coloring problem has reported recently in literature [25]. In this proposed model researchers have tried to address the solution for the graph coloring problem where the theme of proposed algorithm is based on polymerase chain reaction technology. The outlines of their proposed model are as under.

Algorithm 1

Step 1. The splitting of the given graph into subgraphs by identification of common vertices for each subgraphs.

Step 2. Find the vertex order & set of possible colors for each vertex in the subgraph and determination of probes in subgraphs.

Step 3. Construction of initial solution for each subgraphs.

Step 4. Delete all the false solutions for each subgrpahs

Step 5. Combine the solution of all subgraphs and removal of false solution set using polymerase chain reaction.

Step 6. Final coloring solution for the given graph after sequencing the DNA molecules.

The above approach deals the problem by encoding the problem in DNA sequences and by performing the operations mentioned above in efficiently due to parallelism capabilities of DNA computing. In another approach researchers have address the university time table problem [26]. The solution of the problem has find by transforming the problem into graph coloring problem and then solve the problem using genetic algorithm based approach. Mostly the solution for the graph coloring problem in literature are approximate in nature & few of them are exact due to lot of instances generated at intermediate stage in solution and its handling cost incurred using optimization techniques available in literature so researchers most of time prefers to get approximate solution. In recent the researchers have tried to find the exact solution by using adjacency list representation of the given graph and creating the available color pool in the form color adjacency list. The working of their proposed algorithm may be described in the following manner.

Algorithm 2

Step 1. Create an empty stack to store color information about colored vertices in the given graph.

Step 2. Search the colors that are previously assigned to adjacent vertices of a node for which want to assign a color.

Step 3. Add the colors of adjacent vertices of searched node on to the stack.

Step 4. Compare the stored colors on to the stack with the set of available colors and those colors which are not in the stack may be assign to searched node for coloring in the graph.

Step 5. Exit.

The above suggested algorithm by researchers will terminates after assigning the available colors that will always minimum to each vertex in the graph. But the limitation of the suggested technique is that it uses extra running cost consumption in the removal of color information from stack and its comparison with set of available colors for getting the new color which may assign to newly searched vertex. Graph coloring problem is mostly studied in the literature due to its various applications in different area of real world applications. Vertex coloring of graph based local antimagic labeling concept is also reported in literature [27]. Researchers have also focused to evolve the solution for graph coloring problem based on the specific type of graphs which may be generated in specific area of applications in which the solution of application problem can be find by transforming the problem into graph coloring problem. One such kind of specific type of graph & its coloring solution reported for partition graph coloring problem [28]. Several more techniques based on the type graph & its uses in application area have discussed and found suitable for getting the graph coloring based solutions for specific class of problems [29-30].

III. MOTIVATION OF OUR APPROACH

Most of the time researchers focused to solve the graph coloring problem based on optimization techniques either using heuristics available in the literature or genetic algorithm based techniques. Researchers have also given the solution of the problem based on DNA computing model. But in most of the techniques the procedure adopted to find the solution for the graph coloring problem by keeping the constraint for the problem that says during coloring of the vertices in graph no adjacent vertices can assign the same color. But this is not only the constraints in problem that required to be handled in coloring process because it is not always possible that non adjacent vertices in graph may assigned the same color and this implicit constraint must be handled carefully. Thus the handling cost of these additional constraints generated at the intermediate stages of algorithms increases the running cost of the algorithms.

In our approach we have devise a new technique based on defined data structures for the problem that uses only the given constraint for the problem without generating any additional constraints that requires to be taken care and thus it reduces extra efforts needed to handle additional constraints. The propose algorithm will always give the exact solution for the problem with the given minimum number of colors that will need to color the vertices of the given graph.

The procedure uses the given graph represented in the form of adjacency matrix where the entry in the matrix corresponding to adjacent vertices is 1 otherwise it will 0. So if the given graph is having n number of vertices then it will be stored in an nXn square matrix to represent the graph. The available colors that needed to assign among the vertices for the given graph will stored in a one dimensional array and finally an additional one dimensional array is used with size of the array is the number of vertices in the given graph to store the assigned color information of the vertices.

IV. DATA STRUCTURE USED FOR THE PROBLEM

As per the discussion in previous section we present the proposed algorithm that uses the given graph in the form of adjacency matrix along with set of available colors needed to color the graph and a one dimensional array for storing final color values corresponding to each vertex in the graph. Suppose a 3-colorable graph as given in Fig 1 and let these three available colors that will use to color the vertices of the given graph is Blue, Red, Green colors.

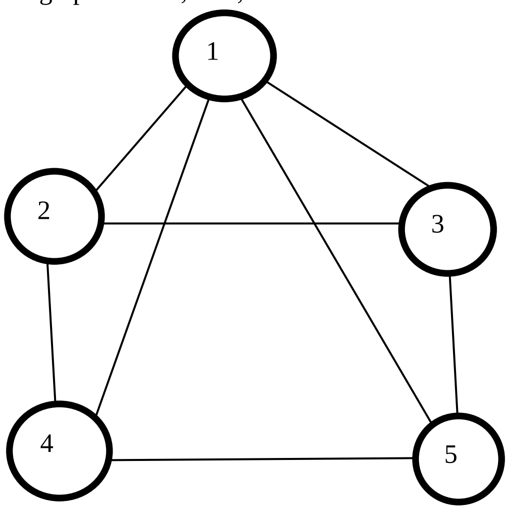

Fig 1: 3-colorable graph

We are using the adjacency matrix correspond to the given the graph for the processing to get the color values that can assign to each vertices in the given graph. The adjacency matrix for the graph given in figure 1 may be represented as in the given Table I.

Table I: Adjacency matrix representation of Fig. 1

	1	2	3	4	5
1	0	1	1	1	1
2	1	0	1	1	0
3	1	1	0	0	1
4	1	1	0	0	1
5	1	0	1	1	0

We have additionally created a vertex-color heap of binary search tree in which the root node of every tree contains vertex information of the graph and children of each tree contains all available colors. The color that can assign to the vertices of the graph will done on the basis of this defined auxiliary data structure. The general structure of vertex-color heap of binary search tree for the graph given in Fig 1 is shown in Fig 2.

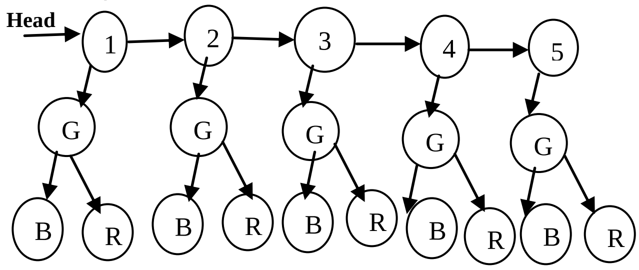

Fig 2: vertex-color binary search tree heap.

In the above Fig 2 the root of each tree will store the name of the vertex in the graph. The root of each tree is connected with its previous root of the tree as well as the next root of the tree. In case of last tree next field of root node will nil. Similarly the first root of the tree is connected with head pointer node. The root node of trees is connected in the form of linear link list. The children of each tree will initially constructed by adding each available colors in such a manner so that it will always create a height balanced search tree except possibly for lost node. This property can achieve easily by sorting the given color in a specific order & median color may be selected as the first color that can act as first child of the root node. The first child of each node may be added as the left or right child of its root node. In our case it is added as left child of root node and in this case root node will have only one child node. Rest of colors will be added as per constraints of the search tree and may act as left or right child of its parent. The final color information that is assigned to each vertex of the given graph will stored into one dimensional array. We have created a 1-D array color_list [ ] having size will be the number of vertices in the given graph.

V. PROPOSED ALGORITHMS

In the propose algorithm we are using adjacency matrix A[ ][ ] for graph, heap of binary search tree that stores available colors in the tree nodes other than root node of each tree and finally assigned color_list[ ] array as global.

Algorithm 3 assign_color (int n)

{
for i=1 to n
{
C= find_color(Head, tree i)

```
color_list[i] = C
for j=i+1 to n
{
if (A[i][j]= = 1)
delete_assigncolor_from_adjacent_vertex (Head, tree j, C)
}
}
} // end of Alogorithm3
```

In the above algorithm argument n is number of vertices in the graph. The feasible color that can assign to a particular vertex i will obtain from heap of binary search tree by calling the procedure find_color( ) and then the color obtained will stored in color_list [ ] array.

Algorithm 4 find_color(Head , tree i)

```
{
P=Head.next
while (P!= Nil)
{
if (P.info = = i)
{if (P.left or P.right)
return color as either( P.left )or (P.right)
else
{
print( vertex i cannot assigned a color)
exit()
}
P= P.next
}
}// end of Algorithm 4
```

Above algorithm4 runs to find the appropriate color from heap of binary search tree of colors and feasible color corresponding to calling vertex will returned.

Algorithm 5

```
del_assigncolor_adjacent_vertex (Head, tree j, C1)

{
Q= Head.next
while (Q1= Nil)
{
if ( Q.info= = j)
search & remove color C1 from tree j(if available in tree)
Q1=Q1.next
}
return;
}// end of Algorithm 5
```

Initially algorithm 3 will call algorithm4 to assign the color to vertices of the graph .Once color has assigned to a particular vertex in the graph, it calls algorithm 5 to update the color nodes in the heap of color trees of the adjacent vertices of the colored vertex by removing assigned color.

VI. WORKING OF PROPOSED ALGORITHMS & ITS ANALYSIS

Before starting of the algorithm we are assuming that the given graph has stored in the form of adjacency matrix A[][] where entry if A[i][j] is 1 then vertex j is adjacent vertex to i otherwise if it is 0 j is a non-adjacent vertex to i. We have created initially a heap of binary search tree in which each tree of the heap will contains all available colors in the form tree node except root of trees. Root of every tree contains the name of the vertex in the graph. The above proposed algorithm will always work efficiently for any kind of graph & always uses minimum number of colors corresponding to any given graph. Suppose the algorithm runs on the given graph in figure 1 after the assigning the color to the vertex 1 as green and 2 as blue the structure of binary search tree heap (Fig 3) and color assignment array (Fig 4) will as follows.

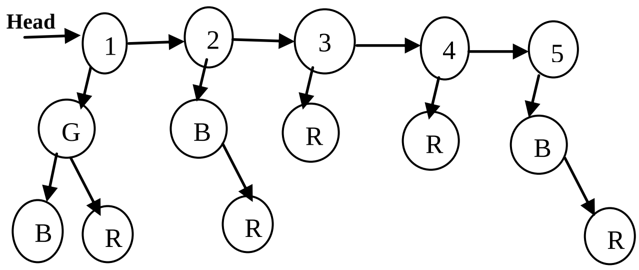

Fig 3: content of vertex-color binary search tree heap.

G	B			

Fig 4: color_list[ ] array after assigning colors to vertex 1,2 in the graph.

A. *Analysis of proposed algorithm*

The formal analysis of the above algorithms consists of analyzing the working of algorithm 3, algorithm 4 and algorithm 5 which is straightforward & simple. The proposed algorithm may implemented in in C language on any platform without its complexity. Suppose the number of vertices in the given graph is n and the minimum number of available colors that can use to color the vertices of the graph is k. The running time of the proposed algorithms may be described in the following manner.

i) The outer loop in procedure assign_color( ) will call find_color procedure n times to assign colors among n number of vertices in the given graph.

ii) The procedure find_color will search the heap to find the vertex for coloring which is the root node of tree and will take maximum n number of comparisons and its child will returned the color(constant time) that can assigned to the vertex . So the instructions executed maximum n^2 times.

iii) The instructions in del_assgncolor_adjacent_vertex ( ) procedure will executed maximum n*lg k times for single call of inner for loop. So the maximum number of times instructions executed in this procedure will be (n*(n-1)*n*lg k).

Finally the maximum number of instructions executed in the proposed algorithm will be $n^2 + (n^2(n-1)\lg k)$.

Therefore the total running complexity of the above proposed algorithm will be $O(n^2(n-1)\lg k)$.

VII. CONCLUSION & FUTURE SCOPE

In the proposed research work we tried to give the exact solution for the graph coloring problem with the help of a newly created data structure in the form of heap of binary search tree in which each tree

node except root node of every tree will contains initially all available colors and root of each
tree in heap will contain the information about a vertex in the graph. There are various solutions for the problem is existing in literature in the form of exact or approximation algorithm. But few of them works satisfactorily in case of dense graph but fails to work efficiently for sparse graph or it's vice versa. Our proposed algorithm will work for any kind of graph and mentioned running time of proposed algorithms is better in case of sparse graph. Since the graph coloring problem is NP class problem and therefore there will always scope for the improvement of running time algorithm proposed. In our case this may be improved modifying the created data structure that reduces the searching time of adjacent nodes in algorithm 5.

REFERENCES

1. Gamache M, Hertz A,Ouellet JO. "A graph coloring model for a feasibility problem in monthly crew scheduling with preferential bidding". *In Comput Oper Res 2007*;34(8): 2384–95.
2. Zufferey N, Amstutz P, Giaccari P. "Graph coloring approaches for a satellite range scheduling problem". *In J Schedul 2008*;11(4):263–77.
3. Werra DD. "An introduction to time tabling". *In Eur J Oper Res1985*;19:151–62.
4. Burke EK, McCollum B, Meisels A, Petrovic S, Qu R. "A graph based hyper heuristic for time tabling problems". *In EurJOperRes2007*; 176: 177–92.
5. Werra D D, Eisenbeis C, Lelait S, Marmol B. "On a graph theoretical model for cyclic register allocation". *In Discret Appl Math1999*;93(2–3):191–203.
6. Smith DH, Hurley S, Thiel SU. "Improving heuristics for the frequency assignment problem". *In Eur JOper Res1998*;107(1):76–86.
7. Hao etl. Algorithms for balanced graph coloring with applications in parallel computing IEEE Transactions on Parallel & Distributed Systems 2017 28(1):
8. Méndez Díaz I, Zabala P. "A cutting plane algorithm for graph coloring". *In Discret Appl Math 2008*;156(2):159–79.
9. Lucet C, Mendes F, Moukrim A. "An exact method for graph coloring". *In Comput Oper Res 2006*;33(8):2189–207.
10. Méndez-Díaz I, Zabala P. "A branch-and cut algorithm for graph coloring". *In Discret Appl Math 2006*;154:826–47.
11. Malaguti E, Monaci M, Toth P. "An exact approach for the Vertex Coloring Problem". *In Discrete Optim 2011*;8(2):174–90.
12. Segundo PS. "A new DSATUR based algorithm for exact vertex coloring". *In Comput Oper Res 2012*;39:1724–33.
13. [13] Chaitin GJ. "Register allocation and spilling via graph coloring". *In SIGPLAN Not 2004*;39(4):66–74..
14. Caramia M, DellOlmo P. "Coloring graphs by iterated local search traversing feasible and infeasible solutions". *In Discret Appl Math 2008*;156(2):201–17.
15. Porumbel DC,HaoJK,Kuntz P. Diversity control and multi parent recombination for evolutionary graph coloring algorithms.In:9th European conference on evolutionary computation in combinatorial optimization (EVOCOP2009). Tbingen, Germany; 2009.p.121–32.
16. Dowsland KA,Thompson JM. An improved ant colony optimization heuristic for graph coloring DiscretApplMath2008;156(3):313–24.
17. PrestwichSD. Generalized graph coloring by a hybrid of local search and constraint programming. Discret Appl Math 2008, 156(2):148-58.
18. Dowsland KA,Thompson JM. An improved ant colony optimization heuristic for graph coloring DiscretApplMath2008;156(3):313–24.
19. Marc Demange, Jerome Monnot, Petrica Pop, Bernard Ries "On The complexity of the selective graph coloring problem in some special classes of graphs". In Theoretical Computer Science 540-541(2014)89-102.
20. Marc Demange, Tinaz Ekim, Bernard Ries, Cerasela Tanasescu " On some applications of the selective graph coloring problem" in European Journal of Operation Research 240(2015) 307-314.
21. Zhaoyang Zhou, Chu-Min Li, Chong Huang, Ruchu Xu "An exact algorithm with learning for the graph coloring problem" Computers & Operation Research 51(2014)282-301.
22. Tabiya Manzoor Beigh,Girdhar Gopal " Use of Genetic Algorithm and Fuzzy Logic in Optimizing Graph Coloring Problem" IJCA Proceedings on Recent Innovations in Computer Science and Information Technology RICSIT 2016(1):1-4
23. Shukla, Ajay Narayan and Garg, M.L. and Misra, Rajiv, "An Approach to Solve Graph Coloring Problem Using Linked List". International Journal of Advanced Studies of Scientific Research, Vol. 4, No. 2, 2019.
24. Abhinav Parihar et al. "Vertex coloring of graphs via phase dynamics of coupled oscillatory networks". Scientific Reports, 7:11, DOI: 10.1038/s41598-017-00825-1.
25. Jin Xu, Xiaoli Qiang, Kai Zhang, Cheng Zhang, Jing Yang" A DNA Computing Model for the Graph Vertex Coloring Problem Based on Probe Graph" Engineering 4 (2018) 61-77.
26. Maram Assi, Bahia Halawi, Ramzi A. Haraty " Genetic Algorithm Analysis using the Graph Coloring Method for Solving the University Time Table Problem" Procedia Computer Science 126(2018) 899-906.
27. Arumugam, S., Premalatha, K., Bača, M. et al."Local Antimagic Vertex Coloring of a Graph". Graphs and Combinatorics (2017) 33: 275.
28. Stefka Fidanova, Petrica Pop. An improved hybrid ant-local search algorithm for the partition graph coloring problem. Journal of Computational and Applied Mathematics, 2016, (293): 55-61.
29. Rezapoor Mirsaleh, M. & Meybodi.,"A new memetic algorithm based on cellular learning automata for solving the vertex coloring problem" . M.R. Memetic Comp. (2016) 8(3):211-22.
30. Xudong Zhu ,Linglong Dai, Zhaocheng Wang, Xiaodang Wang. "Weighted-Graph-Coloring-Based Pilot Decontamination for Multicell Massive MIMO Systems". .IEEE Trans on Vehicular Technology 2017, 66(3):2829-2834.

AUTHORS PROFILE

Ajay Narayan Shukla received the B.E. degree in Computer Science & Engineering in 2002 and M.Tech in Computer Science & Engineering from HBTI Kanpur, India. He is having more than 16 years teaching and research experience in the various technical institutions. He is currently working as Assistant Professor in the Department of Computer Science and Engineering, DIT University Dehradun, India. He is having seven research papers in conferences and journals of repute in the field of networking, algorithms and Internet of Things. His research interest is in the area of Data Structures, Computer Networks and Algorithms based on optimization techniques.

Dr. Vishal Bharti is working as Professor & Head in Department CSE in DIT University, Dehradun. Previously he worked as Professor and Head in AIT-CSE at Chandigarh University, Mohali, Punjab, Associate Prof. & Head, Department of CSE at Dronacharya College of Engineering Gurgaon. He completed his Ph.D. in 2016. He did his M.Tech.and B.E. from Birla Institute of Technology, Mesra. He also holds Doctorate in Management Studies in addition to MBA in IT and Executive-MBA in S&M. He is having a mixed bag of experience of 13+ Years in both Academia and IT Industry He have five filed patents, twenty seven copyrights and 13 Govt. Grants(SERB, DST, NSTEDB, EDI etc.) to his credit. He published fifty six research papers at both National & International domain like IEEE, Springer, Taylor & Francis and was awarded with Best Faculty Award.

Madan Garg received the B.E. (Hons) degree in Electrical Engineering in 1974 from Thapar Institute of Engineering & Technology, Patiala, India and the M.Sc. Engineering degree in Electrical Power Systems in 1979 from Punjab Engineering College, Chandigarh. He received the Ph.D. degree in Computer Science & Engineering in 1992 from Thapar Institute of Engineering & Technology, Patiala, India by carrying out the research work at Indian Institute of Technology, Delhi, India. He was Assistant Professor at Philadelphia University, Amman, Jordan from 1995 to 1998. In 2002, he joined University Tenaga Nasional, Kajang, Malaysia. Presently, he is Professor in the Dept. of Computer Science & Engineering, DIT University, Dehradun. His current research interests include Knowledge representation and applications, Fuzzy Logic and Artificial Neural Networks.