Genetic Algorithms Based Solution To Maximum Clique Problem

Harsh Bhasin
computergrad.com
Faridabad, India
i_harsh_bhasin@yahoo.com

Rohan Mahajan
Lingaya's University
Faridabad, India
mahajanr28@gmail.com

***Abstract*- Maximum Clique Problem (MCP) is an NP Complete problem which finds its application in diverse fields. The work suggests the solution of above problem with the help of Genetic Algorithms (GAs). The work also takes into consideration, the various attempts that have been made to solve this problem and other such problems. The intend is to develop a generic methodology to solve all NP Complete problems via GAs thus exploring their ability to find out the optimal solution from amongst huge set of solutions. The work has been implemented and analyzed with satisfactory results.**

Keywords- Max Clique; NP Complete; Genetic Algorithm; Heuristic Search

I. INTRODUCTION

Clique Problem requires finding out all the fully connected subgraphs of a particular graph. Maximum Clique is a type of clique problem in which maximum clique is to be found. The problem is a NP Complete problem. Applying brute force algorithm is not possible as computation complexity is large. Since the number of possibility is too large and the cases can be arranged in order of their fitness therefore it becomes an apt case of applying GAs. In the work, GAs based solution has been proposed for maximum clique problem. The algorithm has been analyzed for many graphs with satisfactory results. The algorithm presented is much efficient as compared to randomization algorithms.

The paper has been organized as follows. It contains literature review followed by NP hard and explanation of GAs. The work proposed has been explained and analyzed in paper. The reason why maximum clique is an NP Complete problem has been discussed in the sections that follow. It is our sturdy believe that the work will open a window of GAs to the graph problem which is one of the most interesting problems and is even used to find out the largest subset of people knowing each other in a social networking site.

II. LITERATURE REVIEW

The work proposes a generic way to solution of maximum clique problem. So as to get an idea of the previous solutions, many research papers have been studied and analyzed. In one of the work, a branch and bound algorithm uses clique finding algorithm and coloring heuristics to determine the lower and upper bound [1]. The heuristics is used to determine the bound using the concept of graph coloring. The algorithm is evaluated against the edge probability for each size.

In another work, local search solution called k opt local search is used to find the maximum clique using a sequence of add and drop moves [2]. The work has been evaluated against DIM ACH benchmark. So as to limit the time variable depth search has been applied.

Resolution based rules have been used to encode the problem in one of the works. This is helpful in determining the lower bound. The work first of all achieves conjunctive normal form (CNF) and then defines resolution followed by interface based resolution rules [3].

It has been an object of great inquisitiveness as per as computer scientist are concerned. The concept of GAs has been touched in one of the thesis in the handbook of combinatorial optimization [4]. In other work solution using graph coloring is given by Patric R. J. Ostergard [5]. Pivot selection algorithm has been explained in one of the work [6].

The work is a part of larger endeavor to solve all such problem which is NP complete or NP hard using GAs. In this, few problems have already been solved. The problems that have been solved include Post Correspondence Problem [7], Vertex Cover Problem [8], Traveling Salesperson Person Problem [9] and Subset Sum Problem [10]. Currently another work concerning N puzzle problem is also been carried out. The present work is an essential component of bigger task.

III. NP COMPLETE

Algorithms can be divided into classes P & NP. P type problems are solvable in polynomial time [11].They are defined as problems whose complexity is O(n^k), k being a constant. The problems NP are verifiable in Polynomial time. These means that we can find whether a given solution is correct or not in a polynomial time, if given a solution [12].

A problem P is also NP since a problem that can be solved in polynomial time can also be verified in Polynomial time. It is generally agreed that P is a subset of NP. The case is depicted in Figure1. A problem for which no algorithm has yet been discovered, as yet, of polynomial time but it has not been proved that such algorithm does not exist, is called NP Complete problem [12, 13].

A problem is NP hard if it is as hard as any NP complete problem. NP hard problem can be a decision problem, search problem or an optimization problem [12, 13]. Maximum Clique problem is an NP hard problem which is proved in the next section.

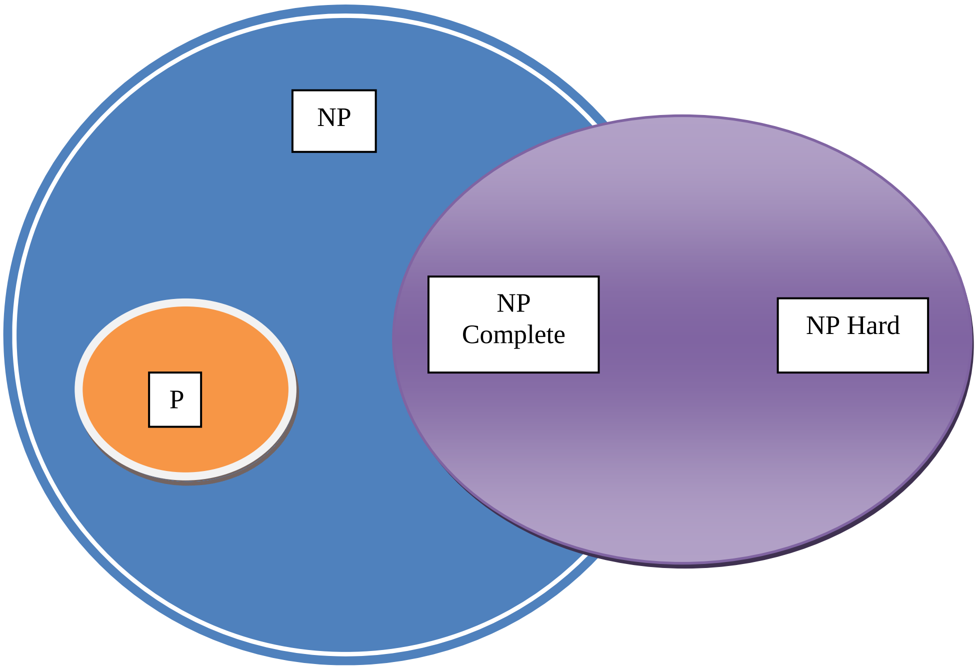

Figure 1 Relation between P and NP

IV. GENETIC ALGORITHM

The present problem is to find out a clique with maximum cardinality. Common understanding is that it is an NP Complete problem. The point can be proved with the following example. For example If there are 50 vertices in a graph and the number of fully connected sub graph are to be found then the total number of such graph will be
${}^{n}C_2 + {}^{n}C_2 + {}^{n}C_4 + {}^{n}C_5 + {}^{n}C_6 + {}^{n}C_7 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ {}^{n}C_{47} + {}^{n}C_{48} + {}^{n}C_{49}$
Which is equal to $2^{50} - 1 \sim 2^{50}$
Now, a very fast computer if processes 10^6 instructions in 1 sec then it will take 35.7 years to elucidate all the sub graphs and find out the maximum cliques. If the graph contains more vertices then the complexity will increase as per the problem. In such cases GAs comes to our rescue. They are known to perform efficiently if sample space is huge.

Genetic algorithms imitate the process of natural selection. A population is generated which consist of chromosomes. Chromosomes are further made up of cells. In our case the cells are binary that is 0 or 1. In the implementation the pseudo random number generator of the language generates a number upto 100. If the number is less than 50 then the cell of that chromosome become 0 and if number is greater than 50 then the cell becomes 1. The numbers of cells in a chromosome depend on the problem at hand. In our case the number of cells in a chromosome is equal to number of vertices in a graph. The number of chromosome can be an optimal number large enough to contain a feasible solution.

The attribute of chromosome are intermingled using the process of crossover. In our implementation 2 random numbers are generated by modulo n, and corresponding chromosome are taken. A crossover point is generated using the number of cells in chromosome. Two new chromosomes are formed as shown in figure 2.

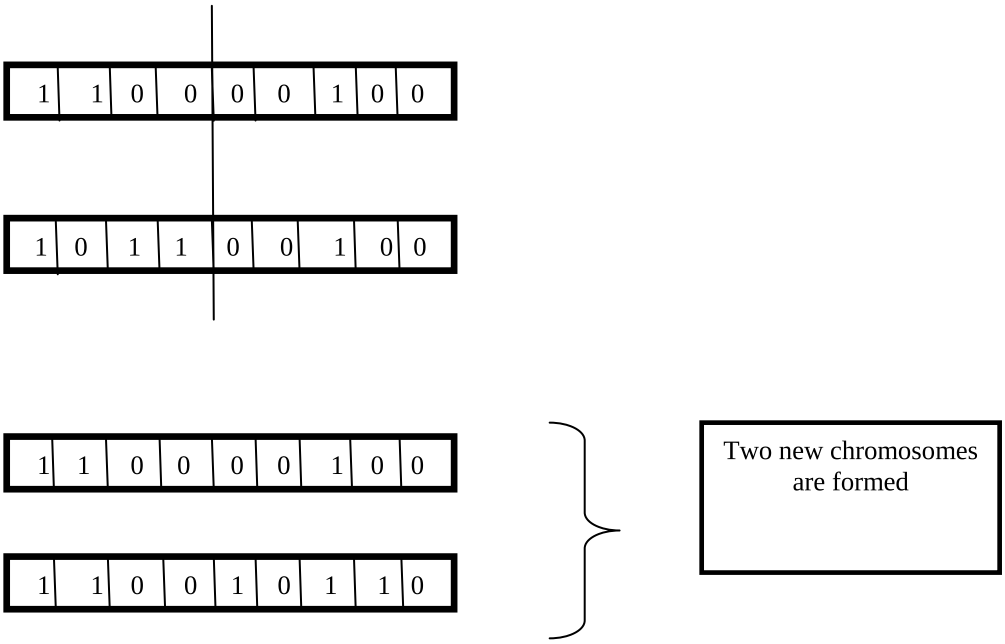

Figure 2 Single Point Crossover

The number of crossover is given by the formula
number of crossover = (number of cells in a chromosome * number of chromosomes * crossover rate) / 100
(1)
In the formula the crossover rate can be anywhere between 2-5%. In our implementation it is taken as 2.5%. So as to break the local maxima mutation is performed. In this process one chromosome is selected and one of its random bits is flipped as shown in the following figure 3.

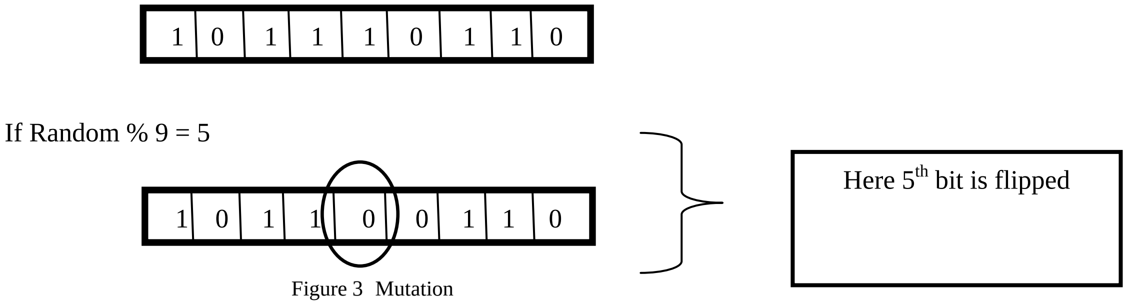

Figure 3 Mutation

The number of mutation is given by the formula
number of mutation = (number of cells in a chromosome * number of chromosome * mutation rate) / 100 (2)
In our implementation mutation rate is taken as 0.5%. The process does not increase the number of chromosomes. In our implementation the need of replication by roulette wheel selection is not there because generating the same set of chromosome can’t give a new sub graph.
The fitness calculation is done by the fitness formula

Fitness chromosome = $\frac{1}{1+e^{\lambda}}$ (3)

where λ denotes the fitness of chromosome. Fitness of a chromosome is defined as proportional to the number of chromosome it contain. More the number of vertices more is the fitness value.

V. PROPOSED WORK

Step 1 The population of n chromosomes is crafted. Each chromosome consists of number of cells equal to number of vertices in a graph. The population is generated using the following algorithm.

```
for each chromosome
        for each cell
        If (random( ) % 100 > 50)
        {
                Cell=1
        }
        else
        {
                Cell=0
        }
        end
end
```

Step 2 In each chromosome 1 are seen and using requisite positions subgraph is selected.

Suppose a graph consist of 6 vertices

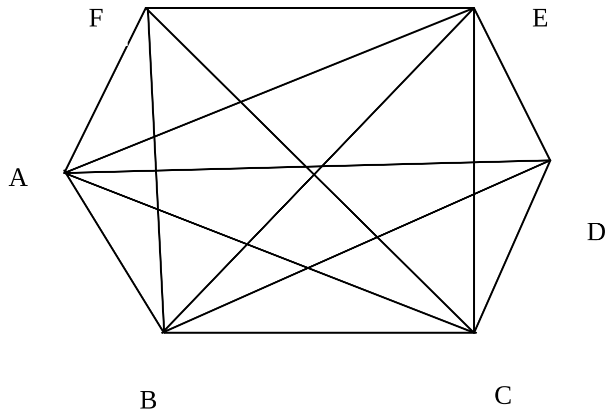

Figure 4 Graph with 6vetices

Now if a chromosome is 110101 then A, B, D, F is selected. If ABDF is fully connected it is inserted into data structure called feasible.

Step 3 for each feasible structure count the number of vertices and assign value equal to =1/(1+ e^{λ}), where where $\lambda = k_1 * n$ (4)

The fitness value is taken as k2/value. The values of K1 and K2 can be determined experimentaly.

Step 4 If solution is not found in step 3, take any 2 chromosome by modulo number of chromosome and then perform crossover as explained in Genetic Algorithm section above.

Step 5 If solution is not found even after crossover then perform mutation which is explained in Genetic Algorithm section.

Step 6 Number of crossover and mutation is given by the formula “1” and “2”

Crossover value is generally 2-5% and mutation is 0.5%. In our experiment crossover is 2.5% and mutation is 0.5%. The process is explained by the following figure 4.

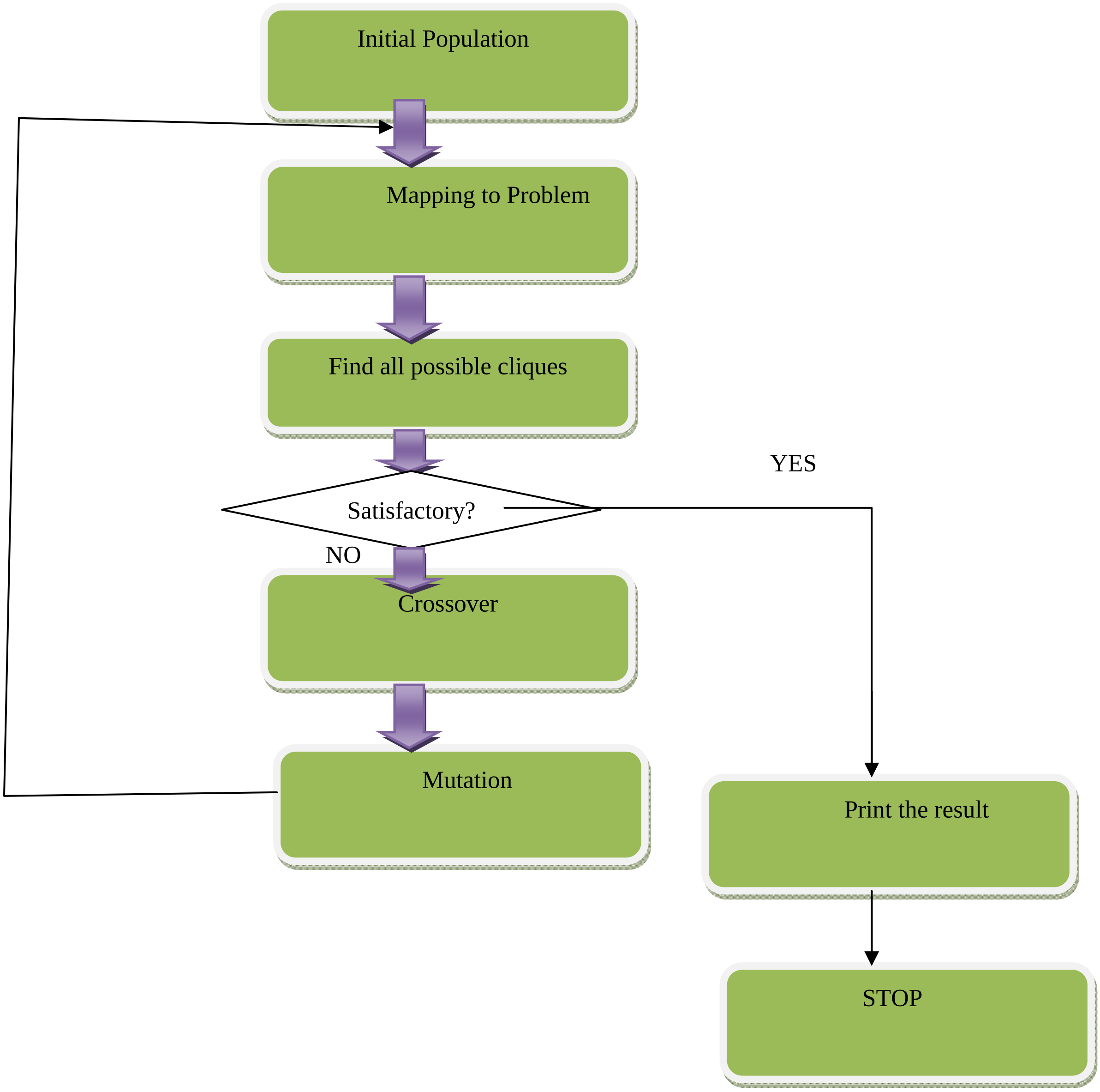

Figure 5 Steps: Maximum Clique

VI. Conclusion And Future Scope

In the analysis, 27 graphs were considered and were randomly generated such that the edges are not known. Analysis was done for various values of n, the number of vertices in a graph. For small values of n, i.e. less than 7, the solution is exact. Genetic process was applied and results were analyzed even for a graph with large number of vertices. The results have been compared with the standard results.

If the value of n is taken as 70 or greater in this experiment, then the results were not satisfactory. On analysis, it was found that this is due to fact that the number of one's in the right half are more. So as to handle this situation, a process of moderation was applied. The results obtained were better.

The rate of replication was taken as 3% and Roulette Wheel Selection was applied which resulted in the replication of favorable data, thus making the population fitter. The overall results are encouraging. It must also be remembered that it is not always that a solution to the problem can be found. Moreover GAs give best solutions but are not guaranteed to do so always.

References

[1] An Algorithm for Finding a Maximum Clique in a Graph: David R. Wood Operations Research Letters 21 (5), 211-217

[2] An Effective Local Search for the Maximum Clique Problem: Kengo Katayama Information Processing Letters Volume 95 Issue 5, 15 September 2005 Pages 503-511

[3] A Max-SAT Inference-Based Pre-processing for Max-Clique: Federico Heras and Javier Larrosa SAT'08 Proceedings of the 11th international conference on Theory and applications of satisfiability testing Pages 139-152 Springer-Verlag Berlin, Heidelberg ©2008 ISBN:3-540-79718-1 978-3-540-79718-0

[4] The Maximum Clique Problem: Immanuel M Bomze Handbook of combinatorial Optimization Supplement Volume A ISBN-0-7923-5924-0

[5] A fast algorithm for the maximum clique problem: Patric R. J. Ostergard Discrete Applied Mathematics 120 (2002) 197–207

[6] A note on the problem of reporting maximal cliques: F. Cazals and C. Karande

[7] Randomized algorithm approach for solving PCP: Harsh Bhasin, Nishant Gupta IJCSE, Volume 4, Issue 1, Jan 2012, e-ISSN : 0975-3397, Print ISSN : 2229-5631

[8] Harnessing Genetic Algorithm for Vertex Cover Problem: Harsh Bhasin, Gitanjali Ahuja JCSE, Volume 4, Issue 2, Jan 2012, e-ISSN : 0975-3397, Print ISSN : 2229-5631

[9] Harnessing Cellular Automata and Genetic Algorithms to solve Traveling Salesman Problem
ICICT, To be held on 1st Apr,2012

[10] Modified Genetic Algorithms Based Solution To Subset Sum Problem IJARAI Volume 1 No 1 April 2012

[11] D.Kumlander,"Applying AI and Incomplete Solution Principles to Solve NP-hard Problems in the Real-Time Systems", Proceedings of the 10th WSEAS International Conference on Computers, Vouliagmeni, Athens, Greece, July 13-15, 2006.

[12] T.H. Cormen, C.E. Leiserson, and R.L. Rivest, "Introduction to Algorithms", the MIT Press, Cambridge, Massachusetts, USA, 2009.

[13] Michael R. Garey and David S. Johnson (1979), "Computers and Intractability: A Guide to the Theory of NP-Completeness". W.H. Freeman.

AUTHORS PROFILE

Harsh Bhasin has completed his B. Tech in Computer Science and Engineering, M. Tech (C. E.), currently perusing PhD .He is proprietor of Sahib Soft and a member of International Association of Computer Science and Information Technology. He has published more than 24 papers in the domain of Random Number generation and NP Hard Problem by Cellular Automata; Cryptography, Machine Learning, Equation Solving and NP Hard Problems by Genetic Algorithm; conversion of south Asian languages by Natural Language Processing. His various papers have been published in IJCA online, IJCSIT, IJCST, IJCSE, IJCNWC, IJIEE and IEEE and National and International conferences.

Rohan Mahajan is an Engineering student, Lingaya's University, Faridabad.