Use of Genetic Algorithm and Fuzzy Logic in Optimizing Graph Coloring Problem

Tabiya Manzoor Beigh
Research Scholar
Dept. of Comp. Sci. & Appls.
KU, Kurukshetra, Haryana, India

Girdhar Gopal
Assistant Professor
Dept. of Comp. Sci. & Appls.
KU, Kurukshetra, Haryana, India

ABSTRACT
Minimum number of colors while coloring the vertices of a graph is a massive apprehension of research scholars in the area of soft computing. Method such as Genetic Algorithm (GA) is highly preferred to solve the Graph Coloring problem by the researchers for many years. In this paper, an optimization technique based on Genetic Algorithm and Fuzzy Logic approach is applied for solving Graph Coloring Problem. The selection operator used in the optimization technique has based on Fuzzy logic. The proposed algorithm is tested on standard DIMACS instances. 11 problems from DIMACS dataset are picked and results are compared with known chromatic numbers. It has found that proposed algorithm has solved nearly all of the problem instances at very good efficiency rate.

General Terms
Fuzzy Logic, Genetic Algorithm, Graph Coloring.

Keywords
Alpha Cut, Fuzzy Logic, Genetic Algorithm, Graph Coloring Problem, Selection

1. INTRODUCTION
A graph can be defined as a set of vertices and edges. Edges will connect one of vertex to other vertex. These vertices can be considered as points or nodes in graph. The edges can be treated as lines or arcs. There are many problems associated with graph theory, as in past it has been observed that graph is a better technique to represent a problem i.e. Konigsberg Bridge Problem etc. Graph Coloring Problem (GCP) is also a well known problem in graph theory; it is a NP-Complete problem [1] [2]. For every graph the chromatic number is defined as smallest number of colors for which the coloring of all the vertices exists in the graph such that adjacent vertices receive different colors. GCP is a problem that also belongs to Constraint Satisfaction Problem (CSP). There are many practical applications of GCP; some of them are listed below:

- Map coloring
- Register Allocation
- Scheduling
- Pattern Matching
- Sudoku
- Radio Frequency Assignment

Genetic Algorithms (GAs) are powerful metaheuristics techniques mostly used in many real-world applications. It is based on natural selection process [3] [4]. GA starts from a random population and evolve the population generation-wise. Better and better solutions are evolved and hence at termination the population consists of best possible solutions. In this paper GA is used to solve the GCP while strictly adhering to the usage of no more than the number of colors equal to the chromatic index to color the various test graphs.

This paper is organized in following sections. Section 2 presents the related work for solutions of graph coloring problem. The proposed solution and algorithm is presented in section 3. Section 4 provides the implementation and results of the algorithms and conclusion is provided in section 5.

2. RELATED WORK
Lot of work has been done in past to illustrate the theory of GCP in terms of its generalization as a CSP [5]. A hybrid method that uses GA and the wisdom of artificial crowds is applied to GCP [6]. Normally Genetic Algorithm uses one parent chromosome to perform mutation operator. Here more than one parent is selected and mutation operator is applied to the best solution.

Different applications and its solutions have been applied to GCP [2]. Evolutionary and Genetic Algorithms have been applied to GCP and its applications [1] [7-9]. Constructive Gene GA is described. The work was direct on schemas and evaluating good schemas that survive for the long time duration. Evolution parameter used with population preserves best schemas [10]. In 1979, the k-coloring problem has been shown as NP-Complete problem by Garey et al. Further, they showed that finding the chromatic number of a graph is NP-Hard, which is at least as hard as NP-Complete problems [11].

3. PROPOSED WORK
Genetic Algorithm is used to optimize the GCP in this study. The basic steps of GA are described as: -

- Population initialization
- Fitness evaluation
- Reproduction
    - Selection of mating pool
    - Crossover
    - Mutation
- Replacement
- Test for termination
- Loop

In the Initialization step, N chromosomes are generated randomly [4]. In Fitness evaluation, the fitness value of each chromosome in a population is calculated. Fitness is assigned as the number of adjacent vertices with same color is treated as penalties for each chromosome. In the generation of new population the operator applied is Selection operator. In Selection operator, chromosomes having high fitness values have more chances to be selected for mating. After Selection operator, Crossover operator is applied. In Crossover, the chromosomes with fittest values will mate to produce more fit children. A pair of parents produces a pair of children. Mutation operator is applied after crossover operator. By applying Mutation operator, mutants are produced to complete the process of evolution in nature. After applying mutation, the resultant children are accepted. The newly generated children will replace old population. If terminating condition is met, then current population is produced as output otherwise process repeats from Fitness function [3-4].

In the proposed Genetic Algorithm, following parameters have been used for experiment:-

- Initial population: Random Generation.
- Selection: Alpha-cut based Selection.
- Crossover: 1-point crossover with probability 0.8.
- Mutation: Swapping with probability 0.01.
- Replacement: Simple Replacement.
- Population Size: 50 to 500.
- Number of Generations: 100 to 2000.

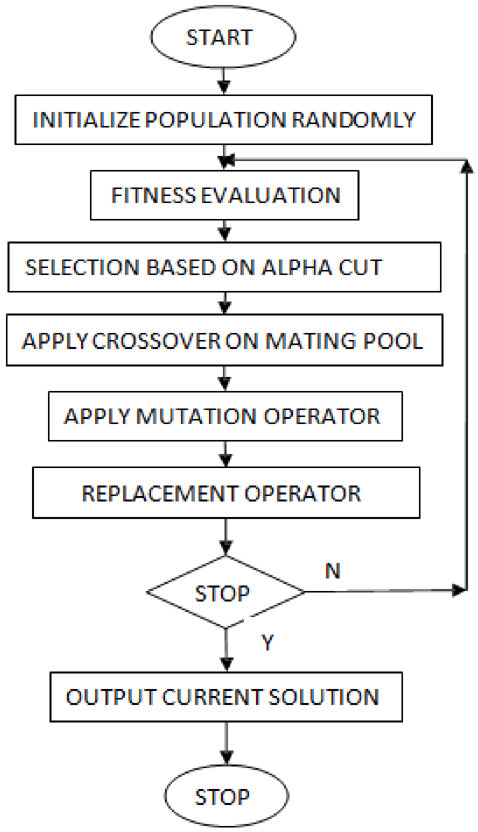

Figure 1: Proposed Genetic Algorithm

In this work, selection operator is used based on alpha cut from Fuzzy Logic. Fuzzy Logic is proposed by Lotfi Zadeh in 1965 to remove the problems of bi-valued logic. Fuzziness of real world can be simulated in theory also with the help of membership value which overcomes the limitations of bi-value logic and provide facilities for multi- valued logic. Alpha cut in fuzzy logic is a de-fuzzification operator. It computes an alpha value for each member in the set, and ignores all the members whose alpha value is less than a threshold value [12]. Proposed GA is explained in figure 1. Proposed selection is implemented with other operators in Genetic Algorithm like Crossover and Mutation. The results of Proposed Genetic Algorithm are compared with Simple Genetic Algorithm keeping all other parameters as constant. The details of results and problem instances are discussed in next section.

4. RESULTS AND DISCUSSIONS

The Proposed Genetic Algorithm is developed in MATLAB 2014a. Performance plots were generated using MATLAB 2014a. Experiments were performed on a server machine with Intel Xeon Quad core 2.7 GHz processor with 60GB RAM. The data set used in these experiments is taken from DIMACS data set. DIMACS is a popular collection of benchmark GCP instances. It is the center for Discrete Mathematics and Theoretical Computer Science. It is a part of Rutgers University (The State University of New Jersey Rutgers, et al. 2011). The files in DIMACS set have a predefined format. All files have a .col extension with a header indicating the number of vertices and edges in particular problem instance i.e.

p edge 64 728

After the header there is a line for each edge indicating the two vertex numbers adjacent with that edge i.e.

e 1 55

11 files were chosen for this experiment from the DIMACS collection. Each of the chosen instance vary in terms of edge count, vertex count and other complexities. Various instances cover a broad range of computational complexity. They differ from each other in terms of vertices range which is 11 (minimum) to 120 (maximum). Also the edges in problems are 20 (minimum) to 2112 (maximum). The chromatic number is already approximated for these problems, so it is easy to compare the performance and to validate the proposed algorithm.

The following files were used in this approach: (queen5_5.col, queen6_6.col, queen7_ 7.col, queen8_8.col, queen9_9, myciel3.col, myciel4.col, myciel5.col, jean.col, huck.col, games120.col). Further statistics of each instance are shown in Table 1 with following columns:

1. The name of the file.
2. Total count of Vertices |V|.
3. Total count of Edges |E|.
4. The chromatic number χ(G) as per literature.
5. The chromatic number χ(G) by proposed Genetic Algorithm.

Three different values of alpha cut have been used. The values used include 0.3, 0.4 and 0.5. Table 1 describes the results with the value of alpha cut of 0.3. Table 2 describes the statistics for each file with the alpha cut value of 0.4.

Table 1: Results Of Proposed Ga On 11 Gcp Instances With The Alpha Cut Value Of 0.3

File	|V|	|E|	Expected χ(G)	Results by proposed GA
myciel13.col	11	20	4	4
myciel14.col	23	71	5	6
queen5_5.col	25	160	5	5

queen6_6.col	36	290	7	7
myciel15.col	47	236	6	6
queen7_7.col	49	476	7	7
queen8_8.col	64	728	9	9
huck.col	74	301	11	13
jean.col	80	254	10	10
queen9_9.col	81	2112	10	10
games120.col	120	639	9	9

Table 2: Results Of Proposed Ga On 11 Gcp Instances With The Alpha Cut Value Of 0.4

File	|V|	|E|	Expected $\chi(G)$	Results by proposed GA
myciel13.col	11	20	4	4
myciel14.col	23	71	5	5
queen5_5.col	25	160	5	5
queen6_6.col	36	290	7	7
myciel15.col	47	236	6	6
queen7_7.col	49	476	7	7
queen8_8.col	64	728	9	9
huck.col	74	301	11	11
jean.col	80	254	10	10
queen9_9.col	81	2112	10	11
games120.col	120	639	9	9

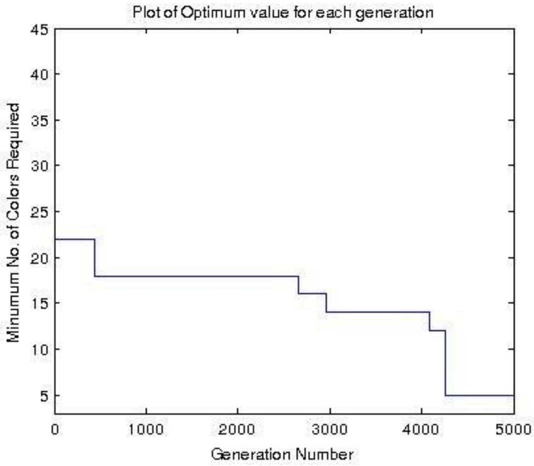

Figure 2: Performance plot for Algorithm with 5,000 generations

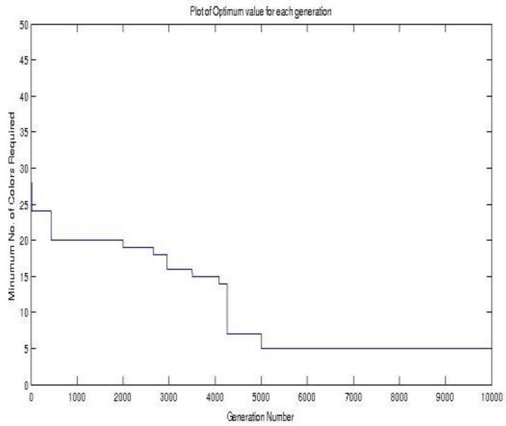

Figure 3: Performance plot for Algorithm with 10,000 generations.

The x-axis in the above graphs shows the number of generations and the y-axis shows optimum value of each generation. The graph shows improvements in the solution quality as the algorithm proceeds. The algorithm converges with the optimum value in both the plots. The main aim during this experiment was to improve the performance of simple GA. The proposed approach was run on various population sizes. The fitness of each chromosome was calculated by finding the number of adjacent vertices of same color. The chromosomes having more number of adjacent vertices of same color are considered as less fit and chromosomes having less number of adjacent vertices of same color are considered fit. Chromosome with highest fitness value is considered as the king of the generation. In proposed GA, the selection operator is implemented based on alpha cut. Other operators are implemented as in SGA. Results of both SGA and proposed GA are compared.

5. CONCLUSION

GCP is the determination of smallest number of colors for which the coloring of the vertices exists such that adjacent vertices receive different colors. GCP is a class of NP-Complete problem. GA is a process of selection of the individuals in a population satisfying the criterion of the survival of fittest and evolving them. Selection operator based on alpha value is used in GA which is based on alpha cut of fuzzy logic. The proposed GA is coded in MATLAB 2014a. 11 problem instances were taken from DIMACS benchmarking graph collection. Results revealed that proposed GA is better than SGA. Experiment was with three different values of alpha (0.3, 0.4 & 0.5), out of which the alpha value of 0.4 shows most promising results. Other techniques like local search can also be incorporated in the proposed solution which is under consideration.

6. REFERENCES

[1] Ali, F. F.-W. (1999). An evolutionary approach for graph coloring. In Proceedings of The International Conference on Systems, Man, and Cybernetics. (pp. 527-532). IEEE.

[2] Porumbel, & Cosmin., D. (2009). Heuristic Algorithms and Learning Techniques: Applications to the Graph Coloring Problem,. Département Informatique, Université d'Angers.

[3] Holland J. , "Adaptation in natural and artificial systems", University of Michigan Press, Ann Arbor, 1975.

[4] Goldberg D. E. , "Genetic algorithms in search, optimization, and machine learning", Addison Wesley Longman, Inc. , ISBN 0-201- 15767-5, 1989.

[5] Diaz, Mendez, I., & Zabala, P. (1999). A generalization of the Graph Coloring Problem. Departamento de Computacion, .

[6] Musa, M., & Roman, V. (2012). Genetic Algorithm applied to Graph Coloring Problem. Proceedings of the Twenty third Midwest Artificial Intelligence and Cognitive Science. University of Cincinnati Cincinnati, Ohio.

[7] Tagawa, K. K. (1999). Distance based hybrid genetic algorithm: an application for the graph coloring problem. In Proceedings of Congress on Evolutionary 2332. Washington.

[8] Croitoru, C. L. (2002). A New Genetic Graph Coloring Heuristic. In Proceedings of The Computational Symposium on Graph Coloring and its Generalizations, (pp. 63-74). Ithaca, New York, USA.

[9] Biman Ray, A. J. (2010). An Efficient GA with Multipoint Guided Mutation for Graph Coloring Problems. International Journal of Signal Processing, Image Processing and Pattern Recognition , 3 (2).

[10] Lorena, L. A. (1997). "Constructive genetic algorithm for graph coloring." Submetido para. Pesquisa Operacional.

[11] Garey, M. J. (1979). Computers and Intractability: A Guide to the Theory of NP-Completeness. New York, Nantes, France.: Freeman and Company.

[12] Rakesh Kumar, Gopal Girdhar (2013). Alpha Cut Based novel selection for Genetic Algorithm. International Journal Of Computer Applications , 13-17.

[13] Back, T., Hammel, U., & Schwefel, H. P. (1997). Evolutionary computation: comments on the history and current state. IEEE Transactions on Evolutionary Computation1, (pp. 3-17).