© 2018 IJSRSET | Volume 4 | Issue 10 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
Themed Section : Engineering and Technology

Solving Graph Coloring Problem Using Genetic Algorithm

A. Nivetha[1], T. Ramesh[2]

[1]MSc Mathematics, Dr SNS Rajalakshmi College of Arts and Science, Coimbatore, Tamil Nadu, India

[2]Assistant Professor, Dr SNS Rajalakshmi College of Arts and Science, Coimbatore, Tamil Nadu, India

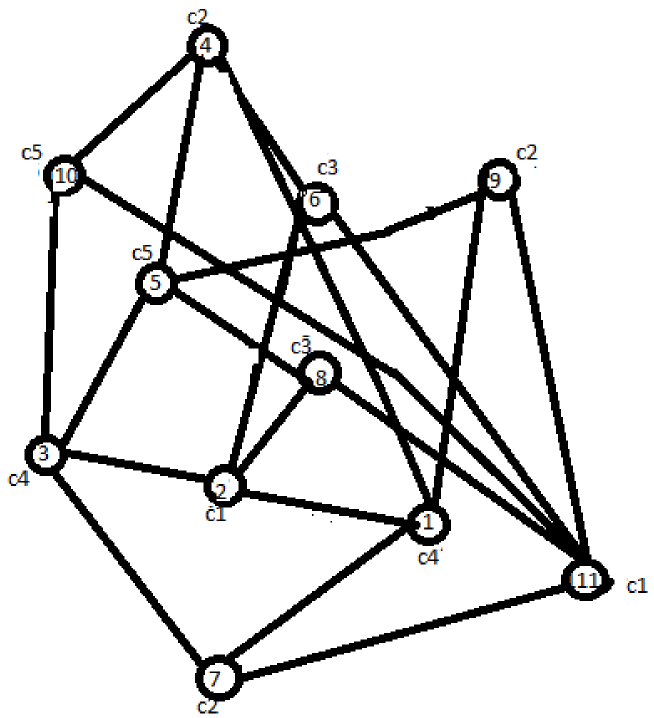
ABSTRACT

To Solve the Graph Coloring Problem we use the Genetic Algorithm (GA) that uses the classical crossover operation .Graph Coloring Problem is the interest researcher's area. We use the chromosomes as a binary operators. By Evolutionary Algorithm (EA) of Coloring is an assignment of colors to the vertices of a graph 'G' such that no two Adjacent Vertices have the same color. We reduce the chromatic number (maximum out degree +1) using the Genetic Algorithm till get the optimum solution.

Keywords : Genetic Algorithm, Binary Coding, Graph Coloring

I. INTRODUCTION

In Graph Coloring Problem is an assignment of colors to the vertices of a graph G, such no two adjacent vertices have the same color, using minimum number of colors. In the problem we illustrate with the simple graph in figure (a).



Assume that the graph G= (v,e) is to be colored with n number of colors. Then the minimum chromatic number is denoted by $\partial(G)$. Then the number n is equal to the minimum chromatic number $\partial(G)$. We Want to color all the vertices using n color, then n is equal to the $\partial(G)$ reached. Here we used the evolutionary algorithm are suitable for the optimization problem. In this paper we use Genetic Algorithm with binary coding, crossover, and fitness operator to the solution.

II. BINARY CODING

In Graph Coloring Problem we use the chromosome are the binary operators 1 and 0. The Fact, We use binary coding scheme for chromosome in our Genetic Algorithm, Which the implementation of the crossover operator easily and to improve the solution. For the simple undirected graph, we illustrated the binary coding in fig b. We use the two dimension array. The column represent the vertices and the row represent the colors respectively. The vertices colored then we used 1 and other entries are 0. Here every column must have one 1 and the row will have more than One 1 but not in the adjacent vertices. In the column has all the entries are 0 that the vertex is not colored. In the one column there are two 1's, then the coding in invalid. No same color assign to the adjacent vertices, then the coding is invalid. In any row all the entries are zero than the color is not used. The binary coding of the fig (a) is shown in fig (b).

	1	2	3	4	5	6	7	8	9	10	11
1	0	1	0	0	0	0	0	0	0	0	1
2	0	0	0	0	0	1	0	1	0	0	0
3	0	0	0	1	0	0	1	0	1	0	0
4	1	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0	0	1	0
6	0	0	1	0	0	0	0	0	0	0	0

Fig (b) (binary coding chromosome)

III. Genetic Algorithm

The Genetic Algorithm for Graph Coloring Problem is shown below.

i. Initialization
ii. Selection (Fitness)
iii. Crossover Point
iv. Mutation
v. Evaluate
vi. End

We repeat the process till we get the optimum solution

i. Initialization

The number of color depends on the Nature of the Problem, but typically contains several possible solutions. Often, the initial color is generated randomly, allowing the entire range of possible solution. Then the initialization table is shown in the fig(c).

	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11
s1	0	1	0	0	0	0	0	0	0	0	1
s2	0	0	0	0	0	1	0	1	0	0	0
s3	0	0	0	1	0	0	1	0	1	0	0
s4	1	0	0	0	0	0	0	0	0	0	0
s5	0	0	0	0	1	0	0	0	0	1	0
s6	0	0	1	0	0	0	0	0	0	0	0

Fig (c) (initialization chromosomes)

ii. Selection

During each successive generation, a portion of existing coloring is selected to breed of new generation. Individual Solutions are selected through a fitness-based process, where fitter solutions (as measured by a fitness function) are typically more likely to be selected. The fitness function is always problem dependent. It Graph Coloring Problem we minimize the number of colors. Fitness is calculated by the number of colored used. Certain selection methods rate fitness of each solutions and preferentially select the best solution. The fitness solution table shown in fig (d).

	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11
c1	0	0	0	1	0	0	1	0	1	0	0
c2	0	1	0	0	0	0	0	0	0	0	1
c3	0	0	0	0	0	1	0	1	0	0	0
c4	0	0	0	0	1	0	0	0	0	1	0
c5	1	0	0	0	0	0	0	0	0	0	0
c6	0	0	1	0	0	0	0	0	0	0	0

Fig (d) (selected chromosomes)

iii. Crossover

In Genetic Algorithms, crossover also called recombination, is a Genetic Operator used to combine the genetic Information of two parents to generate new offspring. Here we use the crossover point in the Graph Coloring Problem. There are three types of crossover points are there:

- Single Crossover Point
- Two Crossover Point
- Uniform Crossover Point

We use uniform crossover points in the problem. It shown in fig(e).

	v1	v2	v3	v4	v5	v6	v7	v8	v9	10v	v11
c1	0	0	0	1	0	0	0	0	0	0	1
c2	0	1	0	0	0	0	1	0	1	0	0
c3	0	0	0	0	0	1	0	0	0	0	0
c4	1	0	0	0	0	0	0	0	0	0	1
c5	1	0	1	0	0	0	0	0	0	0	0
c6	0	0	0	0	0	0	0	0	0	0	0

Fig (e) (crossover chromosomes)

iv. Mutation

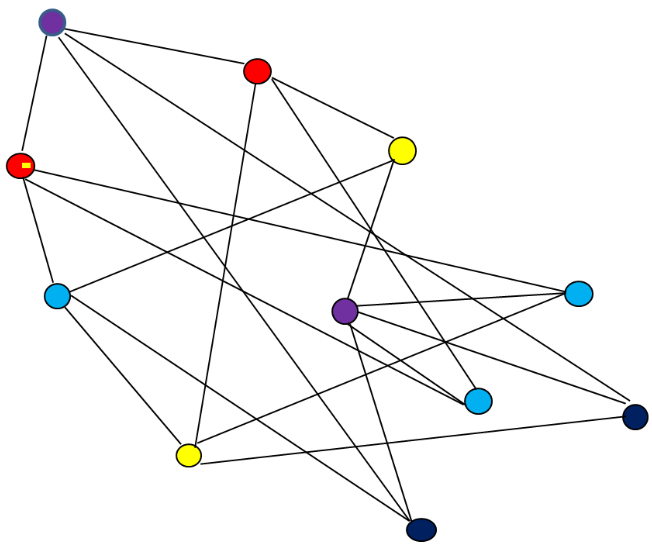
Mutation is a genetic operator used to maintain genetic diversity from one generation of a population of Genetic Algorithm chromosomes to the next level. We use Uniform Mutation in the Graph Coloring Problem. This operator replaces the value of the chosen 0 or 1 with a uniform random Value selected. User specified upper and lower bound to that 0 or 1. This Mutation Operator can only be used for Integer and float genes. The mutation table is given below in fig (f).

	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11
c1	0	1	0	0	0	0	0	0	0	0	1
c2	0	0	0	1	0	0	1	0	1	0	0
c3	0	0	0	0	0	1	0	1	0	0	0
c4	1	0	1	0	0	0	0	0	0	0	0
c5	0	0	0	0	1	0	0	0	0	1	0
c6	0	0	0	0	0	0	0	0	0	0	0

Fig (f) (mutation chromosomes)

v. Evaluate

Again we find the fitness of the mutation chromosomes. Then the solution is optimum we stop. It's not a optimum solution repeat the process till get the optimum solution. Now the solution optimum .We color all the vertices using the minimum color 5.



IV. CONCLUSION

In this paper, we propose a genetic algorithm (GA) for graph coloring problem (GCP). Unlike the previous works, we use a binary coding scheme for the first time for GCP. The main variation operator of our GA is the classical crossover operator of the genetic algorithm (GA). That means the population of the solutions is updated mainly using crossover operator. We select two parents randomly and apply the crossover operator with a high probability. Due to the nature of the encoding, the generated off springs may become invalid and in that case the off springs are corrected to valid solutions. Then a deterministic improvement technique is applied on the corrected off springs with low probability to locally improve the solution quality. The binary coding makes the local improvement procedure easy.

We start with $m + 1$ colors, where m is the maximum out-degree of the graph. The number $m + 1$ is the upper bound of chromatic number. That means, we start with upper bound of the chromatic number and the GA dynamically reduces the chromatic number to the possible minimum chromatic number in a single run.

V. REFERENCES

[1]. M. Kubale , "Graph colorings," American Mathematical Society, 2004.

[2]. C. W. K. Chen and D. Y. Y. Yun, "Unifying graph-matching problem with a practical solution," in Proc. International Conf. on Systems, Signals, Control, Computers, September 1998.

[3]. B. H. Gwee, M. H. Lim, and J. S. Ho, "Solving four-coloring map problem using genetic algorithm," in Proc. Artificial Neural Networks and Expert Systems.

[4]. B. Ray, A. J. Pal, D. Bhattacharyya, and T. Kim, "An efficient GA with multipoint guided mutation for graph coloring problems," International Journal of Signal Processing, Image Processing and Pattern Recognition, vol. 3, no. 2, pp. 51-58, 2010.

[5]. A. Lim, Y. Zhu, Q. Lou, and B. Rodrigues, "Heuristic methods for graph coloring problems," in Proc. Symposium on Applied Computing, 2005.