Global Journal of Pure and Applied Mathematics.
ISSN 0973-1768 Volume 11, Number 4 (2015), pp. 2487-2494

http://www.ripublication.com

Solving Graph Coloring Problem for Large Graphs

Raja Marappan
Department of Computer Applications, School of Computing,
SASTRA University, Thanjavur - 613401, India
e-mail: raja_csmath@cse.sastra.edu

Gopalakrishnan Sethumadhavan
Department of Computer Applications, School of Computing,
SASTRA University, Thanjavur - 613401, India
e-mail: sgk@mca.sastra.edu

Abstract

Most of the engineering applications warrant the solution of Graph Coloring Problem, an NP-hard combinatorial optimization problem. A simple graph G of large sizes can also be solved using soft computing techniques. This paper explores new genetic method using (μ, λ) evolutionary strategy in which μ parents produce λ offspring (λ > μ), and the best μ offspring are chosen to replace the parent population. This genetic method is implemented on some of the large benchmark graphs (n ≥ 500) to obtain the near optimal solution. The (μ, λ) evolutionary strategy is implemented with Multi Parent Conflict Gene Crossover and Conflict Gene Mutation operators with Conflict Gene Removal constraints to minimize the search space, the average number of genetic generations ($\bar{g}$), average crossovers ($\bar{c}$) and mutations ($\bar{m}$) and also to increase the percentage of successful runs. The proposed genetic method is implemented in Java to obtain the near optimal solution for some of the large benchmark graphs and the results are presented.

Keywords: approximation methods; evolutionary approach; genetic algorithm; graph coloring; NP-hard

I. INTRODUCTION

The simple graph G = (V, E) consists of n vertices and m edges having a vertex set $V(G) = \{v_1, v_2, v_3, \ldots, v_n\}$ and an edge set $E(G) = \{e_1, e_2, e_3, \ldots, e_m\}$ such that each edge e_k is uniquely incident with an unordered pair of end vertices v_i and v_j such that $(v_i, v_j) = e_k \in E(G)$ for some i, j, k. A(G), the adjacency matrix of G is an $n \times n$ symmetric binary matrix where $A_{ij} = 1$ if $(v_i, v_j) \in E(G)$; and $A_{ij} = 0$, otherwise. The minimum number of colors $\chi(G)$ required to color V(G) so that no two adjacent vertices have the same color is called the chromatic number of G. Finding the value for $\chi(G)$, is called a Graph Coloring Problem (GCP) and it is used in solving bandwidth assignment, register allocation, scheduling and noise reduction in circuits [1], [2], [3], [20]. The optimal coloring of G occupies the solution space of size n!. The complexity for searching increases with n in the solution space. Hence the approximation methods are to be designed to minimize the search space while maximize the percentage of successful runs. $\chi(G)$ cannot be computed in polynomial time because it is a NP-hard combinatorial optimization problem [1], [4], [10], [11], [12], [13]. Hence effective stochastic algorithm must be designed to find the value of $\chi(G)$. The branch-and-bound, backtracking and branch-and-cut methods are also used to solve GCP [4], [5], [6]. Other methods to solve GCP are Ant Colony Optimization (ACO) [24], Particle Swarm Optimization (PSO) [17], [18], heuristic and local search [14], [15], [25], genetic algorithm with symmetry breaking [24].

A new genetic algorithm to find the near optimal solution to $\chi(G)$ has been devised using Single Parent Conflict Gene crossover (SPCGX) and conflict edge mutation operators [22]. A genetic algorithm using SPCGX and mutation with Conflict Gene Removal (CGR) procedure has been designed in [23]. This paper presents a new genetic method to find the near optimal solution to $\chi(G)$ by applying (μ, λ) evolutionary strategy with Multi Parent Conflict Gene crossover (MPCGX) and Conflict Gene Mutation (CGM) with Conflict Gene Removal (CGR) constraints.

Section II explains the proposed genetic method to solve GCP. Experimental results of this method are presented in Section III. Conclusions are drawn in Section IV.

II. THE PROPOSED GENETIC METHOD

The proposed genetic method used the following notations:

Notation 1: Gene Sequence
The color values of V(G) is represented as $(g_1, g_2, \ldots, g_n)$ where g_i $(1 \leq i \leq n)$ is a positive integer.

Notation 2: Crossover & Mutation Probability, Population Size
The crossover and mutation probabilities are defined as p_c and p_m. The random collection of gene sequences at generation g is called the population and it is denoted

as P_g. The population size N is the number of distinct genes sequences at generation g. That is, $N = |P_g|$.

Notation 3: Objective Function of $\chi(G)$-coloring
f(G), the general fitness function of G is the number of distinct integers present in the gene sequence, and is also positive. It is determined using the survival of the fittest strategy in each generation. The objective function of χ(G)-coloring is to minimize f(G) such that $f(G)-\chi(G) = 0$.

Notation 4: Conflicting Genes
The genes g_a and g_b corresponding to the vertices v_a and v_b are conflicting in a given gene sequence $(g_1, g_2, \ldots, g_n)$ iff $g_a=g_b$ where $(v_a, v_b) \in E(G)$. Then $(g_1, g_2, \ldots, g_n)$ is a conflicting gene sequence. The vertices v_a and v_b are also termed as conflicting vertices of that edge.

The flow chart for the proposed genetic method is shown in Fig.1 and procedure is presented below:

Step 1: Population Initialization
Initialize generation number g = 0. Randomly initialize P_0 in $[1, \chi(G)]$.

Step 2: Fitness Evaluation and Selection
Select a better and worst gene sequence in each generation g using the fitness proportionate selection defined in [23].

Step 3: Crossover Operation
Perform the crossover operation for the selected gene sequences using MPCGX.

Step 4: Mutation Operation
Perform the mutation operation for the generated offspring using the CGM defined in [23].

Step 5: Population updating and termination of genetic operations
Find $F_g(i'')$ for the updated gene sequence i''. If $F_g(i'')$ is 0 then the solution is obtained and stop further generations. Otherwise perform g = g + 1 and update P_g; update the parent population using (μ, λ) evolutionary strategy and go to Step 2.

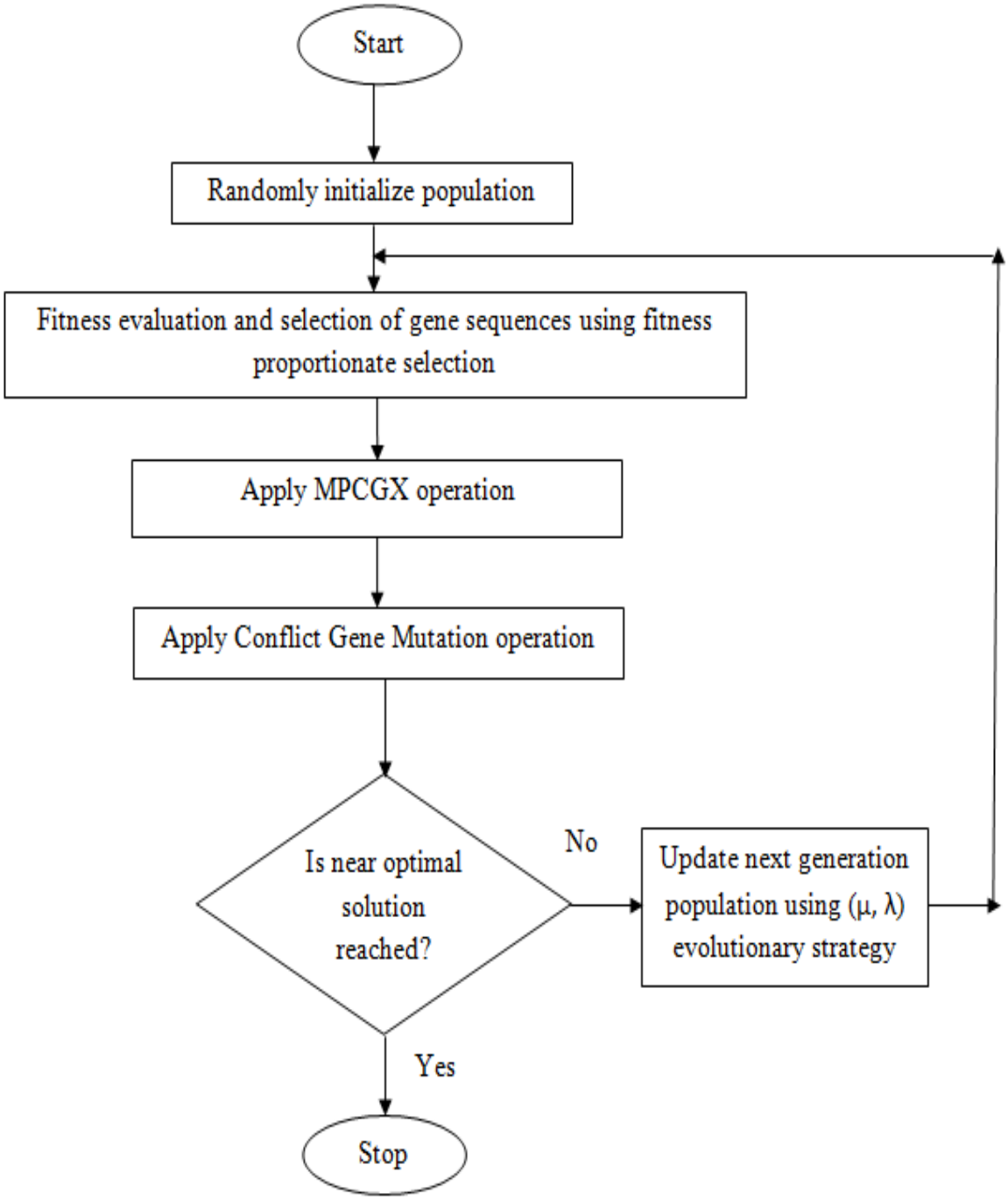

Fig.1. Flowchart of the genetic algorithm to solve GCP

III. EXPERIMENTAL ANALYSIS ON LARGE BENCHMAK GRAPHS

The experiments are conducted to obtain the near optimal solution on some of the large benchmark graphs using Intel Xeon processor with 256 GB DDR3 in Windows 8 Professional OS under JDK 1.8.0 environment. The computation of near optimal solution on some of the large benchmark graphs are shown in Table I. The minimum number of colors, $\bar{g}$ and the percentage of successful runs are obtained using the proposed genetic algorithm and are shown in Table I.

Table I COMPUTATION OF NEAR OPTIMAL SOLUTION ON LARGE BENCHMARK GRAPHS

Graph No	Graph (G)			Minimum color obtained in the proposed method	$\bar{g}$ obtained in the proposed method	Percentage of successful runs of the proposed method
	Graph Type	Instances	$\chi(G)$			
1	homer.col	n=561; m=1629	13	13	11060	80.0%
2	inithx.i.1.col	n=864; m=18707	54	54	71	48.0%
3	inithx.i.2.col	n=645; m=13979	31	31	33	20.0%
4	inithx.i.3.col	n=621; m=13969	31	31	63	52.0%
5	latin_square_10.col	n=900; m=307350	*	415	798	60.0%

* $\chi(G)$ not yet found [19]

The devised method is compared with some of the existing methods and the inferences are given below:

1. The method stochastically converges with smaller N to obtain the optimal solution for inithx.i.1.col unlike the near-equal convergence in [21].
2. For homer.col graph both $\bar{g}$ and the percentage of successful runs are increased.
3. The method obtains the near optimal color of latin_square_10.col graph (n=900; m=307350) is 415.

$\bar{c}$ & $\bar{m}$ performed to obtain the near optimal solution for these large graphs using the devised method is shown in Fig. 2. The average computing time to obtain the near optimal solution for these graphs is presented in Fig. 3.

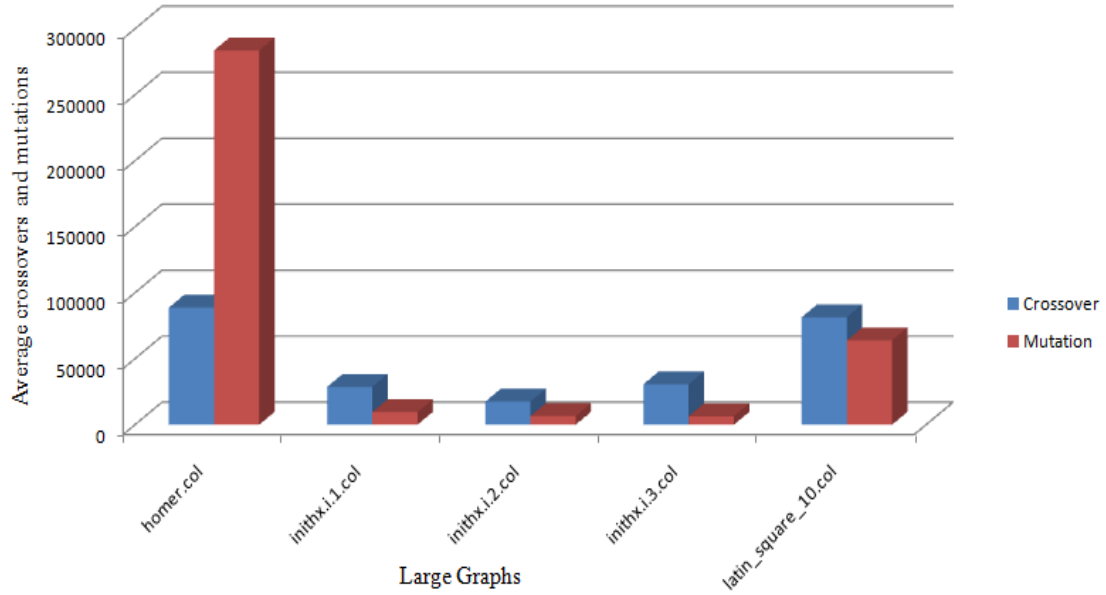

Fig. 2 $\bar{c}$ & $\bar{m}$ performed on large graphs

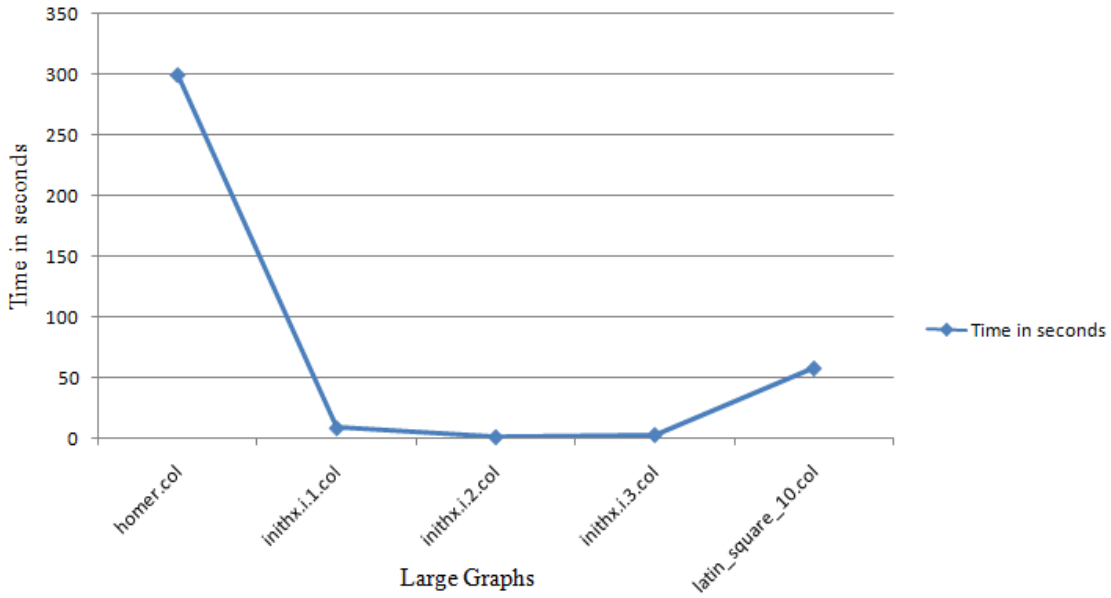

Fig. 3 Computing time (in seconds) to find the near optimal solution on some of the large graphs

IV. CONCLUSIONS

New method is devised to solve graph coloring using (μ, λ) evolutionary strategy and the experiment is conducted on some of the large graphs to obtain the near optimal

solution. The proposed crossover and mutation is applied with (μ, λ) evolutionary strategy to obtain the solution. The devised method is working well even for minimal population size N (≤ 15). The devised operators monotonically decrease the value of the fitness function and minimize the average number of genetic generations, crossovers and mutations to obtain the stochastic convergence. In future, we intend to refine the devised operators further to minimize the computational complexity.

ACKNOWLEDGMENT

The authors would like to acknowledge the support rendered by the management of SASTRA University by the way of providing necessary infrastructure and financial support for this research. The authors would like to thank Gary Lewandowski, and Michael Trick for uploading the graph repository in world wide web consortium [19].

REFERENCES

1. Garey, M. R. and Johnson, D. S., Computers and Interactability: A Guide to the Theory of NP-Completeness, Freeman, San Francisco, 1979.
2. Tommy R. Jensen, Bjarne Toft, Graph Coloring Problems, Wiley-Interscience, 1995.
3. Noise Reduction in VLSI Circuits using Modified GA Based Graph Coloring - International Journal of Control and Automation, Vol. 3, No. 2, June, 2010.
4. Anuj Mehrotra and Michael A. Trick. A column generation approach for graph coloring. INFORMS Jounal on Computing, 8(4):344–354, 1996.
5. Isabel Méndez-Diaz and Paula Zabala. A branch-and-cut algorithm for graph coloring. Discrete Applied Mathematics, 154(5):826–847, 2006.
6. R. Monasson. On the analysis of backtrack procedures for the coloring of random graphs. In E. Ben-Naim, H. Frauenfelder, and Z. Toroczkai, editors, Complex Networks, pages 235–254. Springer, 2004.
7. Lixia Han and Zhanli Han, A Novel Bi-objective Genetic Algorithm for the Graph Coloring Problem, 2010 Second International Conference on Computer Modeling and Simulation.
8. G. Rudolph. "Finite Markov chain results in evolutionary computation: A tour Horizon", Fundamenta informaticae, vol. 35, no.2, pp. 67-89.
9. Back T. "Evolutionary algorithms in theory and practice", New York Oxford University Press, pp. 21-28, 1996.
10. Philippe Galinier and Alain Hertz. A survey of local search methods for graph coloring. Comput. Oper. Res., 33(9):2547–2562, 2006.
11. D. S. Johnson, C. R. Aragon, L. A. McGeoch, and C. Schevon. Optimization by simulated annealing: An experimental evaluation; part II, graph coloring and number partitioning. Operations Research, 39(3):378–406, 1991.
12. David S. Johnson and Michael A. Trick. Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge, volume 26. American Mathematical Society, 1993.

13. Kazunori Mizuno and Seiichi Nishihara. Constructive generation of very hard 3-colorability instances. Discrete Appl. Math., 156(2):218–229, 2008.
14. Steven Prestwich, Generalised graph colouring by a hybrid of local search and constraint programming, Elsevier, Discrete Applied Mathematics 156 (2008), pages 148-158.
15. Igor Dukanovic, Franz Rendl, A semidefinite programming-based heuristic for graph coloring, Elsevier, Discrete Applied Mathematics 156 (2008), pages 180-189.
16. Thang N. Bui, ThanhVu H. Nguyen, Chirag M. Patel, Kim-Anh T. Phan, An ant-based algorithm for coloring graphs, Elsevier, Discrete Applied Mathematics 156 (2008), pages 190-200.
17. Ling-Yuan Hsu, Shi-Jinn Horng, Pingzhi Fan, Muhammad Khurram Khan, Yuh-Rau Wang, Ray-Shine Run, Jui-Lin Lai, and Rong-Jian Chen, MTPSO algorithm for solving planar graph coloring problem, Expert Systems with Applications 38 (2011), pages 5525-5531.
18. Cui, G., Qin, L., Liu, S., Wang, Y., Zhang, X., & Cao, X. (2008). Modified PSO algorithm for solving planar graph coloring problem. Progress in Natural Science, 18, 353–357.
19. The Graph Coloring instances, http://mat.gsia.cmu.edu/COLOR/instances.html.
20. Junichi Yoshino, Isao Ohtomo, Study on efficient channel assignment method using the genetic algorithm for mobile communication systems, Soft Computing, Springer 2004.
21. Soma Saha, Rajeev Kumar, Gyan Baboo, Characterization of graph properties for improved Pareto fronts using heuristics and EA for bi-objective graph coloring problem, Applied Soft Computing, ASOC-1644, 2012.
22. Raja Marappan, and Gopalakrishnan Sethumadhavan, A new genetic algorithm for graph coloring, In CIMSim2013, 5th International Conference on Computational Intelligence, Modelling and Simulation, Seoul, South Korea, pages 49-54, 2013.
23. Gopalakrishnan Sethumadhavan, Raja Marappan, A Genetic Algorithm for Graph Coloring using Single Parent Conflict Gene Crossover and Mutation with Conflict Gene Removal Procedure, 2013 IEEE International Conference on Computational Intelligence and Computing Research, India, pages 350-355, 26-28 December 2013.
24. Raja Marappan, Gopalakrishnan Sethumadhavan, Solution to Graph Coloring Problem using Evolutionary Optimization through Symmetry-Breaking Approach, International Journal of Applied Engineering Research, 2015, Research India Publications, Volume 10, Number 10 (2015) pp. 26573-26580.
25. Raja Marappan, Gopalakrishnan Sethumadhavan, Solution to Graph Coloring Problem using Heuristics and Recursive Backtracking, International Journal of Applied Engineering Research, 2015, Research India Publications, ISSN 0973-4562, Volume 10, Number 10 (2015) pp. 25939-25944.