

A New Multi-Objective Evolutionary Approach to Graph Coloring and Channel Allocation Problems

S. Balakrishnan¹, Tamilarasi Suresh², Raja Marappan^{3,*}

¹Dr MGR Educational and Research Institute, Maduravoyal, Chennai, India.

²Department of Information Technology, St. Peter's Institute of Higher Education and Research, Avadi, Chennai, India.

³School of Computing, SASTRA Deemed University, Thanjavur, Tamil Nadu, India.

How to cite this paper: S. Balakrishnan, Tamilarasi Suresh, Raja Marappan. (2021) A New Multi-Objective Evolutionary Approach to Graph Coloring and Channel Allocation Problems. *Journal of Applied Mathematics and Computation*, 5(4), 252-263.
DOI: 10.26855/jamc.2021.12.003

Received: September 6, 2021

Accepted: September 30, 2021

Published: October 14, 2021

***Corresponding author:** Raja Marappan, School of Computing, SASTRA Deemed University, Thanjavur, Tamil Nadu, India.

Email: raja_csmath@cse.sastra.edu

Abstract

Recent years, a large amount of graph coloring algorithms have been applied in different disciplines and engineering. This paper exhibits a new multi-objective evolutionary algorithm using the new evolutionary operators with multi-objectives in finding the solution to graph coloring and channel allocation. Clique, a maximal connected complete subgraph of a given graph is obtained in solving channel allocation problem. The outcomes of the devised evolutionary method are compared with other recent approaches. Multiple better gene sequences are selected and are applying the subsequent crossover and mutation operations with multiple objectives. The devised operators significantly minimize the problem search space and average generations compared to the standard genetic algorithm. The proposed operators achieve better solution compared to some of the existing well known methods. The expected performance measures are also increased while minimizing the expected generations. The proposed operators can also be applied in solving special engineering applications of graph coloring.

Keywords

Evolutionary Algorithm, Graph Coloring, Channel Allocation, NP-hard

1. Introduction

A simple graph $G = (V, E)$ consists of a vertex set $V(G) = \{v_1, v_2, \dots, v_n\}$ with n vertices and edge set $E(G) = \{e_1, e_2, \dots, e_m\}$ with m edges such that each edge $e_i \in E(G)$ is uniquely incident with a vertex pair $(v_j, v_k) \in E(G)$, $1 \leq i \leq m$, $1 \leq j, k \leq n$ [1]. $A(G)$ is an $n \times n$ adjacency matrix of G such that A_{jk} assigns 1 if there exists an edge joining vertices v_k and v_j ; otherwise, A_{jk} assigns 0 [2]. Graph coloring is an assignment of color values to vertices of a G subject to certain constraints. It assigns integers to $V(G)$ such that no two adjacent vertices are assigned the same integer. Graph coloring is applied in different problems [34-37]. Graph Coloring Problem (GCP) is finding the value of a chromatic number, $\chi(G)$ that is the least number of integers required to assign $V(G)$ [1-3]. Different approximation and evolutionary methods are designed in solving GCP.

GCP is applied in different engineering applications. GCP can also be specifically applied in wireless networks where the cells are properly assigned to the corresponding channels based on its interference matrix called Channel Assignment Problem (CAP) [29, 38-41]. CAP allocates bandwidth and communication channels to base stations in cellular and wireless networks. The goal of CAP is to achieve maximum system spectral efficiency through channel reuse by avoiding the constraints of interference. The important criterion for next generation wireless systems is efficient spectrum utilization [53-54]. CAP defines the constraints: co-channel, adjacent channel and co-site constraints. Clique partitioning is applied in solving channel allocation schemes. Given the undirected graph G , its clique C is a subgraph which consists of the

vertex set V' in $V(G)$ such that there is an edge between every pair of vertices in V' . That is, C is the complete subgraph of G . The split of G into a finite number of maximum cliques such that each vertex belongs to exactly a single clique is called clique partition, an NP-complete problem. Some of the other methods in solving graph coloring and its application are Particle Swarm Optimization (PSO) [18-19, 25-26], Ant Colony Optimization (ACO) [24], Cuckoo search [20], evolutionary, approximation, backtracking, branch and cut, column generation and heuristic search [4-11, 13-17, 21-23]. These methods are having certain limitations and are listed in Table 1 [42-50].

Table 1. Limitations in some well known methods

Method	Limitations
Branch and cut	Selection of upper bound is required
Ant Colony Optimization	Recursive algorithm with more stack space is required
Column Generation	Computationally complex operations are performed
Semi-definite Programming	Near optimal solution is obtained for some graphs
Particle Swarm Optimization	More generations are required and applied for planar graphs
DSATUR	Near optimal solution
FROGSIM, FINOCCHI	Near optimal solution
Genetic operators	Near optimal solution
Evolutionary methods	More search space and expensive

GAs are heuristic and adaptive search methods which are based on “survival of fittest”. Recently genetic algorithms are applied in discovering communities in complex social networks and solving constraint optimization problems based upon the principles of natural evolution [55-56]. The proper design of new genetic operators can reduce the exploration of problem search space and expected generations. Fast stochastic convergence is obtained when minimizing the problem search space and reducing the expected generations. Hence this paper presents a new genetic method in solving GCP and CAP with clique partition strategy. New crossover and mutation operators with multiple objectives are introduced in this paper. These operators minimize the expected generations and also reduce the exploration of search space. These operators are also minimizing the minimum color for certain graphs compared to existing methods.

Section 2 formulates the mathematical model of CAP. The new genetic method with clique partitioning strategy is presented in Section 3. Section 4 analyses the theoretical results and stochastic convergence of the devised operators. The well known problems are experimented and its outcomes are compared with other methods and are presented in Section 5. Section 6 presents the conclusions.

2. Mathematical Formulation of CAP

In general, the wireless networks consist of m channels and n cells. The solution to CAP is defined in the vector space F which represents a symmetric binary matrix of size $n \times m$.

For each value of j and k ($1 \leq j \leq n$ and $1 \leq k \leq m$),

$F(j, k) = 1$ if cell j is assigned the k th channel; $F(j, k) = 0$ otherwise.

2.1 Demand Constraint

A total of d_i channels are required for cell i . This implies that the total number of ones in row i of F must be d_i . That is if the assignment to cell i violates the demand constraint, then

$$\sum_{q=1}^m F_{iq} - d_i \neq 0$$

2.2 Co-Site channel Constraint (CSC)

If channel p is within distance c_{ii} from an already assigned channel q in cell (i.e., $|p - q| < c_{ii}$), then channel p must not be assigned to cell i . That is if the assignment of channel p to cell i violates CSC, then

$$\sum_{\substack{q=p-(c_{ii}-1) \\ q \neq p \\ 1 \leq q \leq m}}^{p+(c_{ii}-1)} F_{iq} > 0$$

2.3 Co-Channel Constraint (CCC) & Adjacent Channel Constraint (ACC)

CCC requires that $c_{ij} = 1$, that is, cells i and j must use channels that maintain a minimum separation of 1 unit. ACC requires that $c_{ij} \geq 2$, that is, cells i and j must use channels separated by at least 2 units. Hence for CCC & ACC, if channel p in cell i is within distance from an already assigned channel q in cell j , where $c_{ij} > 0$ and $i \neq j$ (i.e., $|p - q| < c_{ij}$), then channel p must not be assigned to cell i . That is, if the assignment of channel p to cell i violates CCC and/or ACC, then

$$\sum_{\substack{j=1 \\ j \neq i \\ c_{ij} > 0}}^n \sum_{\substack{q=p-(c_{ij}-1) \\ 1 \leq q \leq n}}^{p+(c_{ij}-1)} F_{jq} > 0$$

Hence the objective function of CAP is derived in the following cost function:

$$C(F) = \sum_{i=1}^n \left(\sum_{q=1}^m F_{iq} - d_i \right) + \sum_{i=1}^n \sum_{p=1}^m \left(\sum_{\substack{q=p-(c_{ii}-1) \\ q \neq p \\ 1 \leq q \leq n}}^{p+(c_{ii}-1)} F_{iq} \right) F_{ip} + \sum_{i=1}^n \sum_{p=1}^m \left(\sum_{\substack{j=1 \\ j \neq i \\ c_{ij} > 0}}^n \sum_{\substack{q=p-(c_{ij}-1) \\ 1 \leq q \leq n}}^{p+(c_{ij}-1)} F_{jq} \right) F_{ip}$$

The objective function of CAP is to minimize $C(F)$; that is, find an F such that $C(F)$ reaches 0.

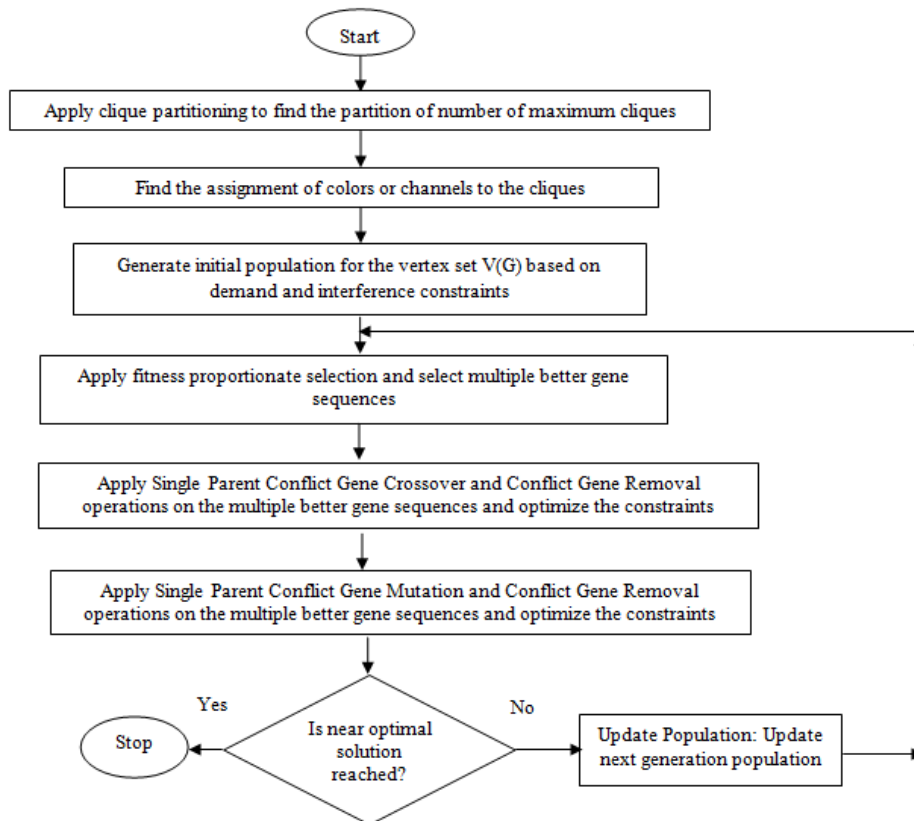


Figure 1. Flowchart of the proposed method in solving GCP & CAP.

3. Proposed Genetic Algorithm

The proposed method initially finds the number of maximum cliques for the given graph G . Then independent cliques are identified and solved based on the constraints. Starting with random initial population, fitness proportionate selection select multiple better gene sequences and are applied for crossover operation in the next stage. The new conflict based crossover and mutation operators are applied on more than one better individual with multiple objectives to produce the better offspring. This iterative procedure is continued until reaching the better solution.

The flowchart of this new method is presented in Figure 1. The clique partition strategy splits $V(G)$ into different cliques. The strategy splits G into a required number of maximum cliques to minimize the network interference in solving CAP. The independent cliques are identified and are colored or assigned the channels independently.

The general procedure to find the solution of CAP is given below:

- Partition $V(G)$ into number of maximum cliques using clique partition;
- Apply the proposed operators for the identified cliques;
- Find the independent cliques and assign valid channels based on interference constraints;

G' : (S, E') is constructed from G using clique partition. Every vertex s_i in S consists of a set of finite vertices v_i in V . A vertex s_i in S is a common neighbor of s_j and s_k in S if there exists edges (i, j) and (i, k) in E' . The maximum clique is obtained in the vertex set s_i in S using the following steps:

- Put each v_i in $V(G)$ in a distinct vertex s_i in S of G' .
- Identify the vertices s_{i1}, s_{i2} in S such that (s_{i1}, s_{i2}) in $E'(G)$ with large common adjacency.
- Merge the vertices into a single vertex s_{i1i2} which have s_{i1} and s_{i2} in all.
- Determine the intersection set which consists of the entire intersection neighbors of s_{i1} and s_{i2} .
- Remove all edges which are emerging from s_{i1} or s_{i2} in G' .
- Update new links from s_{i1i2} to all vertices.
- Perform the operations (a) to (f) repeatedly until G has 0 edges.

Procedure Clique Partition

// construct $G'(S, E')$

Initialize $S = \emptyset$ and $E' = \emptyset$;

For all $v_i \in V(G)$ do $s_i = \{v_i\}$; $S = S \cup \{s_i\}$;

For all $e_{i,j} \in E(G)$ do $E' = E' \cup \{e'_{i,j}\}$;

Repeat when $E' \neq \emptyset$:

$mc = -1$;

For all $e'_{i,j} \in E'$ do

$c_{i,j}$ = cardinality of common neighbor (G', s_i, s_j) ;

if $c_{i,j} > mc$

$mc = c_{i,j}$;

intersection set = intersection $(G', s_{index1}, s_{index2})$;

$E' = \text{remove}(E', s_{index1})$;

$E' = \text{remove}(E', s_{index2})$;

$s_{index1index2} = s_{index1} \cup s_{index2}$;

$S = S - s_{index1} - s_{index2}$;

$S = S \cup \{s_{index1index2}\}$;

Update E' for all $s_i \in$ intersection set

The new genetic operators are designed as follows:

3.1 Crossover Operation

The following are the steps in new recombination operator.

- Apply the following steps for the selected gene sequences.
- Choose p_{cr} .
- If $p_{cr} > p_c$ then apply the conflict color recombination and update offspring.
- Apply the removal of conflict colors for fixed times.
- Repeat this operation for fixed iterations to remove the conflicts.
- Apply crossover at single, multiple points on the offspring i' and j' to produce new offspring by optimizing the constraints.
- Repeat the above steps for finite iterations.

3.2 Mutation Operation

The mutation is performed when $p_m < p_{mr}$ fined as follows:

- Identify the conflicting individuals.
- Update new individuals with minimum change in gene values.
- Perform the gene updates to remove the total conflicts.
- Apply mutation at single, multiple points and update the better offspring by optimizing the constraints.
- Repeat the above steps for finite iterations.

4. Analysis and Stochastic Convergence

The new genetic operators are obtained better solution and achieve the stochastic convergence after reaching a fixed value of g . The proposed evolutionary operators achieve stochastic convergence in getting better near optimal solution.

Proof for the work carried to achieve stochastic convergence: Let S represents the finite search space. For arbitrarily chosen individuals $i, j \in S$, individuals should satisfy reachability condition.

Let p_c and p_m define the crossover and mutation probabilities for applying the proposed crossover and mutation. Then the probability of generating i'' is given by

Probability (i'') \geq

$$\text{Probability of crossover} \times p\{i' = \text{crossover}(i)\} \times \text{Probability of mutation} \times p\{i'' = \text{mutation}(i')\}$$

Clearly $p\{i' = \text{recombination}(i)\} > 0$ and $p\{i'' = \text{mutation}(i')\} > 0$.

Thus, the probability of generating i'' lies in 0 and 1 and hence i reaches to i'' [12].

The elitism operation replaces the worst individual with better one and P_g is updated. The devised operators reduce $F_g(j)$ & $F_g(i)$ during the generations and the following inferences are obtained:

Inference 1: Either $F_g(i)$ or $F_g(j)$ decreases to obtain the near solution.

$$F_0(i) \geq F_1(i) \geq F_2(i) \geq \dots \geq F_q(i) \geq F_{q+1}(i) = 0.$$

$$F_0(j) \geq F_1(j) \geq F_2(j) \geq \dots \geq F_q(j) \geq F_{q+1}(j) = 0.$$

Inference 2: $F_g(i)$ or $F_g(j)$ satisfies the following relationship for finite generations:

$$F_0(i) \geq F_1(i) \geq F_2(i) \geq \dots \geq F_q(i)$$

$$F_0(j) \geq F_1(j) \geq F_2(j) \geq \dots \geq F_q(j)$$

Then after finite generations the following relationship achieves the near optimal solution.

$$F_q(i) < F_{q+1}(i) < \dots < F_t(i)$$

$$F_q(j) < F_{q+1}(j) < \dots < F_t(j).$$

Thus the proposed operators prove the stochastic convergence in obtaining better near optimal solution.

5. Results and Analysis

The proposed method is experimented in solving some of the benchmark graphs and channel allocation problems using Intel Core i5-2450M 2.5GHz system using Java [27]. The graph instances n and m are used in evaluating the performance measurements of the method. The outcomes are shown in Tables 2, 3 and 4 with the following inferences:

- Stochastic convergence is reached even to the high density graphs, such as miles 1500.col.
- The expected generations \bar{g} is reduced compared to other genetic methods and the obtained values are shown in figures from Figure 2 to Figure 4.
- Solution is obtained with minimal complexity for queen, register allocation, Miles graphs.
- Better near solution is obtained for latin_square_10.col.

When comparing the results of this method with some of the existing methods, this new method offsets the problems such as improving solution, search space reduction, and achieve quick convergence for minimal N [25-58].

The important findings are as follows:

- Solution is obtained for different families of random graphs.
- Stochastic convergence is achieved for small $N < 10$ compared to other methods.
- Solution is obtained for queen, Mycielski graphs compared to near-optimal performance of existing methods.

4. Provides better solution than DSATUR, FINOCCHI and FROGSIM methods.

The comparison of results obtained with the other strategies is shown in Table 5. Table 6 represents the different CAP instances and the expected generations obtained are given in Figure 5. The convergence of near optimal solution is shown in Table 7. This new method significantly reduces the expected generations and achieves higher frequency of convergence for the benchmark and random instances.

Table 2. Computation of Near Optimal Solution on Small Graphs (n < 100)

Graph No	Graph Type	Graph (G)			Minimum color obtained in	
		Instances	$\chi(G)$	Density	Existing Method	Proposed method
1	queen5_5.col	n=25; m=320	5	1.07	5	5
2	queen6_6.col	n=36; m=580	7	0.92	7	7
3	queen7_7.col	n=49; m=952	7	0.81	7	7
4	queen8_8.col	n=64; m=1456	9	0.72	9	9
5	myciel5.col	n=47; m=236	6	0.22	6	6
6	myciel6.col	n=95; m=755	7	0.17	7	7
7	myciel4.col	n=23; m=71	5	0.28	5	5
8	myciel3.col	n=11; m=20	4	0.36	4	4
9	huck.col	n=74; m=301	11	0.11	11	11
10	jean.col	n=80; m=254	10	0.08	10	10
11	david.col	n=87; m=406	11	0.11	11	11
12	queen8_12.col	n=96; m=2736	12	0.60	12	12
13	queen9_9.col	n=81; m=2112	10	0.65	10	10
14	1-Insertions_4.col	n=67; m=232	4	0.10	4	4
15	2-Insertions_3.col	n=37; m=72	4	0.11	4	4
16	3-Insertions_3.col	n=56; m=110	4	0.07	4	4
17	4-Insertions_3.col	n=79; m=156	3	0.05	3	3

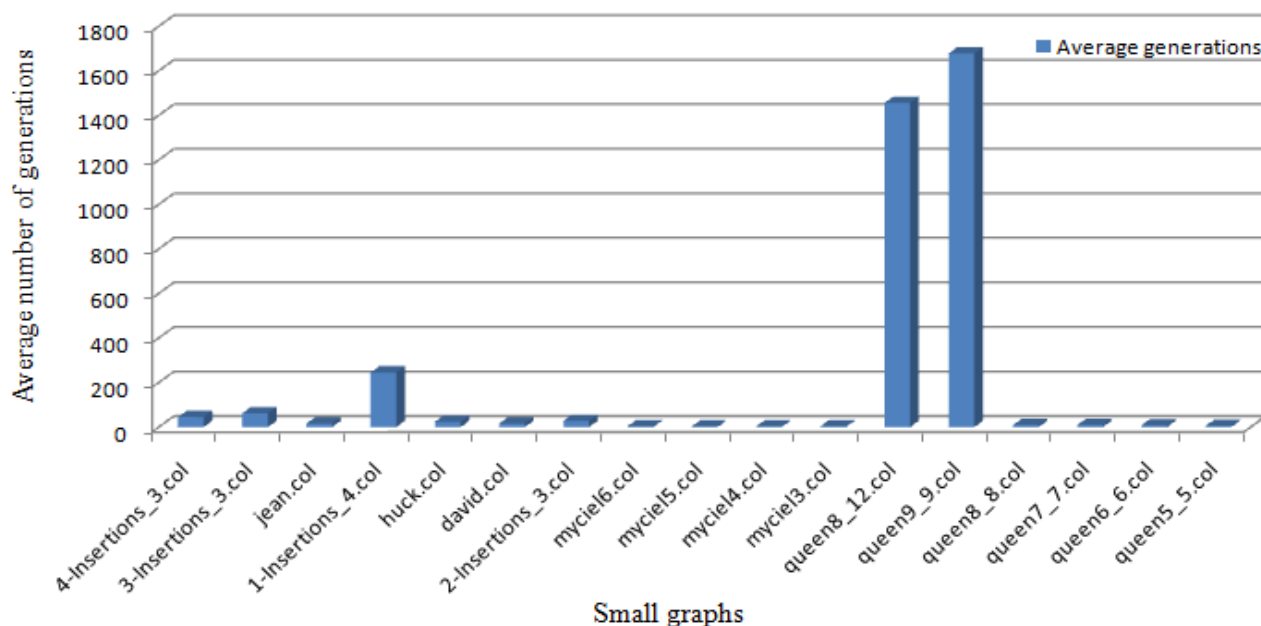


Figure 2. Expected generations obtained for small graphs.

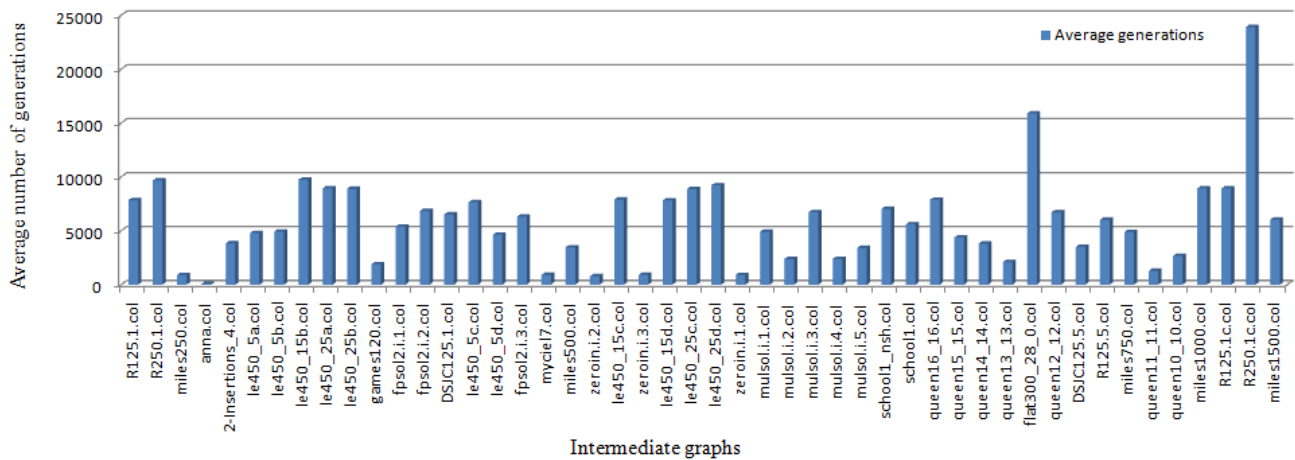


Figure 3. Expected generations obtained for intermediate graphs.

Table 3. Computation of Near Optimal Solution on Intermediate Graphs (100 ≤ n < 500)

Graph No.	Graph Type	Graph (G)			Minimum color obtained in	
		Instances	$\chi(G)$	Density	Existing method	Proposed method
18	myciel7.col	n=191; m=2360	8	0.13	8	8
19	games120.col	n=120; m=638	9	0.09	9	9
20	miles250.col	n=128; m=387	8	0.05	8	8
21	anna.col	n=138; m=493	11	0.05	11	11
22	queen10_10.col	n=100; m=2940	11	0.59	11	11
23	queen12_12.col	n=144; m=5192	12	0.50	12	12
24	queen14_14.col	n=196; m=8372	14	0.44	14	14
25	queen15_15.col	n=225; m=10360	15	0.41	15	15
26	queen16_16.col	n=256; m=12640	16	0.39	16	16
27	queen11_11.col	n=121; m=3960	11	0.55	11	11
28	queen13_13.col	n=169; m=6656	13	0.47	13	13
29	miles500.col	n=128; m=1170	20	0.14	20	20
30	miles750.col	n=128; m=4226	31	0.52	31	31
31	miles1000.col	n=128; m=6432	42	0.79	42	42
32	miles1500.col	n=128; m=10396	73	1.28	73	73
33	zeroin.i.1.col	n=211; m=4100	49	0.19	49	49
34	zeroin.i.2.col	n=211; m=3541	30	0.16	30	30
35	zeroin.i.3.col	n=206; m=3540	30	0.17	30	30
36	mulsol.i.1.col	n=197; m=3925	49	0.20	49	49
37	mulsol.i.2.col	n=188; m=3885	31	0.22	31	31
38	mulsol.i.3.col	n=184; m=3916	31	0.23	31	31
39	mulsol.i.4.col	n=185; m=3946	31	0.23	31	31
40	mulsol.i.5.col	n=186; m=3973	31	0.23	31	31
41	le450_5a.col	n=450; m=5714	5	0.06	5	5
42	le450_5b.col	n=450; m=5734	5	0.06	5	5
43	le450_5c.col	n=450; m=9803	5	0.10	5	5
44	le450_5d.col	n=450; m=9757	5	0.10	5	5
45	le450_15b.col	n=450; m=8169	15	0.08	15	15
46	le450_15c.col	n=450; m=16680	15	0.16	15	15
47	le450_15d.col	n=450; m=16750	15	0.17	15	15
48	le450_25a.col	n=450; m=8260	25	0.08	25	25
49	le450_25b.col	n=450; m=8263	25	0.08	25	25

50	le450_25c.col	n=450; m=17343	25	0.17	26	25
51	le450_25d.col	n=450; m=17425	25	0.17	26	25
52	school1.col	n=385; m=19095	14	0.26	14	14
53	school1_nsh.col	n=352; m=14612	14	0.24	14	14
54	fpsol2.i.1.col	n=496; m=11654	65	0.09	65	65
55	fpsol2.i.2.col	n=451; m=8691	30	0.09	30	30
56	fpsol2.i.3.col	n=425; m=8688	30	0.10	30	30
57	2-Insertions_4.col	n=149; m=541	4	0.05	4	4
58	DSJC125.1.col	n=125; m=736	5	0.09	5	5
59	DSJC125.5.col	n=125; m=3891	-	0.50	17	17
60	R125.1.col	n=125; m=209	-	0.03	5	5
61	R125.1c.col	n=125; m=7501	-	0.97	46	46
62	R125.5.col	n=125; m=3838	-	0.50	36	36
63	R250.1.col	n=250; m=867	-	0.03	8	8
64	R250.1c.col	n=250; m=30227	-	0.97	64	64
65	flat300_28_0.col	n=300; m=21695	28	0.48	31	28

Table 4. Computation of Near Optimal Solution on Large Graphs (n ≥ 500)

Graph No	Graph Type	Graph (G)			Minimum color obtained in	
		Instances	$\chi(G)$	Density	Existing method	Proposed method
66	latin_square_10.col	n=900; m=307350	-	0.76	97	97
67	homer.col	n=561; m=1629	13	0.01	13	13
68	inithx.i.1.col	n=864; m=18707	54	0.05	54	54
69	inithx.i.2.col	n=645; m=13979	31	0.07	31	31
70	inithx.i.3.col	n=621; m=13969	31	0.07	31	31
71	DSJC500.5.col	n=500; m=62624	-	0.50	48	48
72	DSJR500.1.col	n=500; m=3555	-	0.03	12	12
73	DSJR500.5.col	n=500; m=58862	122	0.47	122	122
74	DSJC1000.5.col	n=1000; m=249826	-	0.50	83	83
75	R1000.1.col	n=1000; m=14378	-	0.03	20	20
76	flat1000_50_0.col	n=1000; m=245000	50	0.49	50	50
77	flat1000_60_0.col	n=1000; m=245830	60	0.49	60	60
78	flat1000_76_0.col	n=1000; m=246708	76	0.49	82	80

Table 5. Comparison of proposed method with well known and recent methods [28, 30-31, 51-52]

Graph	$\chi(G)$	Near optimal color obtained obtained using					
		TS	HCA	RLF	Local search method	Hybrid and population-based method	Proposed method
le450_25c.col	25	26	26	27	25	25	25
le450_25d.col	25	26	26	28	25	25	25
flat300_28.col	28	32	31	37	28	29	28
flat1000_76_0.col	76	87	83	106	85	82	80
latin_square_10.col	-	130	127	122	99	97	97

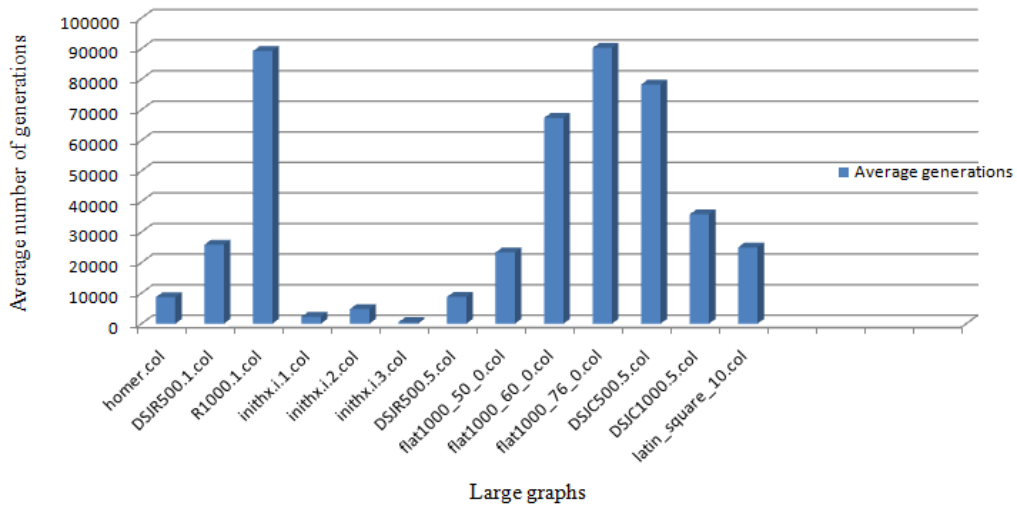


Figure 4. Expected generations obtained for large graphs.

Table 6. C(F) obtained for various benchmark instances

Problem	Number of Cells	Number of Channels	C(F)
Ex1	4	11	0
Ex2	5	17	0
HEX1	21	37	0
HEX2	21	91	0
HEX3	21	21	0
HEX4	21	56	0
KUNZ1	10	30	0
KUNZ2	15	44	0
KUNZ3	20	60	0
KUNZ4	25	73	0

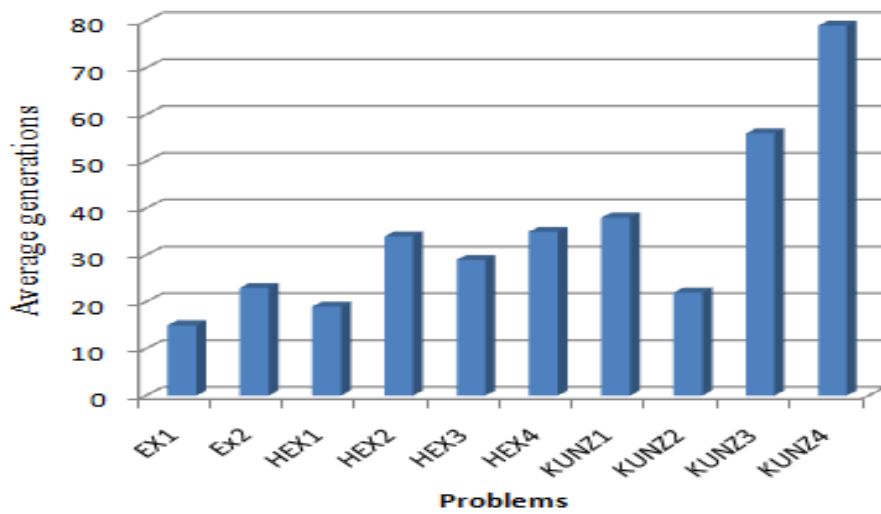


Figure 5. Expected generations obtained for CAP instances.

Table 7. Frequency of Convergence

Problem	No. of Cells (n)	No. of Channels (m)	Simulation Run Length	Frequency of Convergence
1	4	11	100	100%
2	25	73	20000	100%
3	21	385	20000	100%
4	500	1250	40000	100%
5	1000	1500	45000	100%
6	2000	1750	50000	99.6%
7	5000	2000	50000	98.7%
8	10000	2500	50000	98.9%
9	20000	5000	50000	97.9%
10	50000	7500	50000	95.2%

6. Conclusion

A new multi-objective evolutionary algorithm using new crossover and mutation operators with multi-objectives in finding the solution to graph coloring and its applications are devised in this research work. The outcomes of the devised evolutionary method are compared with other strategies. The method of selecting multiple better gene sequences and applying them in subsequent operations with multiple objectives using the devised operators minimized the searching significantly and also reduces the average generations compared to the standard genetic algorithm. The proposed operators achieve better solution compared to other methods. The expected performance measures are also increased while minimizing the expected generations. The proposed operators can also be applied in solving special engineering applications of graph coloring. The clique split strategy reduce the interference of CAP with minimal N and hence better performance is obtained compared to the well known methods. Stochastic convergence of higher frequency of convergence is attained for the standard and random CAPs of large sizes.

The scope for future work includes the following:

- Devising statistical based evolutionary operators for further state space reduction
- Improving the performance of the devised operators with proper searching strategy to minimize the generations
- Enhancing the results by devising a parallel evolutionary method
- Designing the new genetic programming in solving applications of coloring

References

- [1] Balakrishnan, R. and Ranganathan, K. (2000). A Textbook of Graph Theory. 1st Edition, Springer-Verlag, New York Publisher.
- [2] Michael R. Garey and David S. Johnson. (1979). Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman and Company, New York.
- [3] Timir Maitra, Anindya J. Pal, Debnath Bhattacharyya, and Tai-hoon Kim. (2010). Noise Reduction in VLSI Circuits using Modified GA Based Graph Coloring. International Journal of Control and Automation, 3(2), (2010).
- [4] Hertz, A. and Werra D. de. (1987). Using tabu search techniques for graph coloring. Computing, 39(4), 345-351.
- [5] Charles Fleurent and Jacques A. Ferland. (1996). Genetic and hybrid algorithms for graph coloring. Annals of Operations Research, 63(3), 437-461.
- [6] Christine, L. (2006). Mumford: New Order Based Crossovers for the Graph Coloring Problem. Parallel Problem Solving from Nature, 4193, 880-889.
- [7] Anuj Mehrotra and Michael A. Trick. (1995). A Column Generation Approach for Graph Coloring. INFORMS Journal on Computing, 8, 344-354.
- [8] Isabel Méndez-Díaz and Paula Zabala. A Branch and Cut algorithm for graph coloring. Discrete Applied Mathematics, 154, 826-847.
- [9] Remi Monasson. (2004). On the Analysis of Backtrack Procedures for the Coloring of Random Graphs. Lect.Notes Phys., 650, 235-254.

- [10] Lixia Han and Zhanli Han. (2010). A Novel Bi-objective Genetic Algorithm for the Graph Coloring Problem. 2nd International Conference on Computer Modeling and Simulation.
- [11] Tamás Szép and Zoltán Ádám Mann. (2010). Graph coloring: The more colors, the better? 11th International Symposium on Computational Intelligence and Informatics (CINTI).
- [12] Rudolph and Günter. (1998). Finite Markov Chain Results in Evolutionary Computation: A Tour d'Horizon. *Fundamenta Informaticae*, 35(1-4), 67-89.
- [13] A. E. Eiben, J. K. Van Der Hauw, and J. I. Van Hemert. (1998). Graph Coloring with Adaptive Evolutionary Algorithms. *Journal of Heuristics*, 4, 25-46.
- [14] Philippe Galinier and Alain Hertz. (2006). A survey of local search methods for graph coloring. *Computers & Operations Research*, 33(9), 2547-2562.
- [15] David S. Johnson, Cecilia R. Aragon, Lyle A. McGeoch, and Catherine Schevon. (1991). An Experimental Evaluation; Part II, Graph Coloring and Number Partitioning. *Operations Research*, 39(3).
- [16] David S. Johnson and Michael A. Trick. (1993). Cliques, Coloring, and Satisfiability. *American Mathematical Society*, 26.
- [17] Kazunori Mizuno and Seiichi Nishihara. (2008). Constructive generation of very hard 3-colorability instances. *Discrete Applied Mathematics*, 156, 218-229.
- [18] Blanca Cases, Carmen Hernandez, Manuel Grana, and Alicia D'Anjou. (2008). On the ability of Swarms to compute the 3-coloring of graphs. *Artificial Life*, (2008).
- [19] Manuel Graña, Blanca Cases, Carmen Hernandez, and Alicia D'Anjou. (2010). Further Results on Swarms Solving Graph Coloring. *ICCSA 2010: Computational Science and Its Applications*, 541-551.
- [20] Yongquan Zhou, Hongqing Zheng, Qifang Luo, and Jinzhao Wu. (2013). An improved Cuckoo Search Algorithm for Solving Planar Graph Coloring Problem. *Applied Mathematics & Information Sciences*, 7(2), 785-792.
- [21] Soma Saha, Rajeev Kumar, and Gyan Baboo. (2013). Characterization of graph properties for improved Pareto fronts using heuristics and EA for bi-objective graph coloring problem. *Applied Soft Computing*, 13(5), 2812-2822.
- [22] Steven Prestwich. (2008). Generalised graph colouring by a hybrid of local search and constraint programming. *Discrete Applied Mathematics*, 156, 148-158.
- [23] Igor Dukanovic and Franz Rendl. (2008). A semidefinite programming-based heuristic for graph coloring. *Discrete Applied Mathematics*, 156, 180-189.
- [24] Thang N. Bui, ThanhVu H. Nguyen, Chirag M. Patel, and Kim-Anh T. Phan. (2008). An ant-based algorithm for coloring graphs. *Discrete Applied Mathematics*, 156, 190-200.
- [25] Ling-Yuan Hsu, Shi-Jinn Horng, Pingzhi Fan, Muhammad Khurram Khan, Yuh-Rau Wang, Ray-Shine Run, Jui-Lin Lai, and Rong-Jian Chen. (2011). MTPSO algorithm for solving planar graph coloring problem. *Expert Systems with Applications*, 38, 5525-5531.
- [26] Guangzhao Cui, Limin Qin, Sha Liu, Yanfeng Wang, Xuncai Zhang, and Xianghong Cao. (2008). Modified PSO algorithm for solving planar graph coloring problem. *Progress in Natural Science*, 18, 353-357.
- [27] The Graph Coloring instances. <http://mat.gsia.cmu.edu/COLOR/instances.html>.
- [28] Hugo Hernández and Christian Blum. (2014). FrogSim: Distributed graph coloring in wireless ad hoc networks. *Telecommunication Systems*, 55(2), 211-223.
- [29] Junichi Yoshino and Isao Ohtomo. (2005). Study on efficient channel assignment method using the genetic algorithm for mobile communication systems. *Soft Computing*, 9(2), 143-148.
- [30] Pablo San Segundo. (2012). A new DSATUR-based algorithm for exact vertex coloring. *Computers & Operations Research*, 39(7), 1724-1733.
- [31] Philippe Galinier and Jin-Kao Hao. (1999). Hybrid Evolutionary Algorithms for Graph Coloring. *Journal of Combinatorial Optimization*, 3(4), 379-397.
- [32] Raja Marappan and Gopalakrishnan Sethumadhavan. (2013). A new genetic algorithm for graph coloring. 5th International Conference on Computational Intelligence, Modelling and Simulation, 49-54, (2013).
- [33] Gopalakrishnan Sethumadhavan and Raja Marappan. (2013). A Genetic Algorithm for Graph Coloring using Single Parent Conflict Gene Crossover and Mutation with Conflict Gene Removal Procedure. *IEEE International Conference on Computational Intelligence and Computing Research*, 350-355, (2013).
- [34] Xiang'en Chen, Yue Zu, Jin Xu, Zhiwen Wang, and Bing Yao. (2011). Vertex-Distinguishing E-Total Colorings of Graphs. *Arabian Journal for Science and Engineering*, 36(8), 1485-1500, (2011).
- [35] Lewis, R. M. R. (2016). *A Guide to Graph Coloring, Algorithms and Applications*. Springer International Publishing Switzerland (2016).
- [36] Marc Demange, Tinaz Ekim, Bernard Ries, and Cerasela Tanasescu. (2014). On some applications of the selective graph coloring problem. *European Journal of Operational Research*, (2014).

- [37] Marc Demange, Tinaz Ekim, and Bernard Ries. (2015). On the minimum and maximum selective graph coloring problems in some graph classes. *Discrete Applied Mathematics*, (2015).
- [38] Shih Heng Cheng and Ching Yao Huang. (2013). Coloring-Based Inter-WBAN Scheduling for Mobile Wireless Body Area Networks. *IEEE Transactions on Parallel and Distributed Systems*, 24(2), (2013).
- [39] Elmahdi Driouch and Wessam Ajib. (2013). Downlink Scheduling and Resource Allocation for Cognitive Radio MIMO Networks. *IEEE Transactions on Vehicular Technology*, 62(8), (2013).
- [40] Elmahdi Driouch and Wessam Ajib. (2012). Efficient Scheduling Algorithms for Multiantenna CDMA Systems. *IEEE Transactions on Vehicular Technology*, 61(2), (2012).
- [41] Matti Peltomäki, Juha-Matti Koljonen, Olav Tirkkonen, and Mikko Alava. (2012). Algorithms for Self-Organized Resource Allocation in Wireless Networks. *IEEE Transactions on Vehicular Technology*, 61(1), (2012).
- [42] Lynn Takeshita. (2016). *Coloring Signed Graphs*, (2016).
- [43] Edita Macajov, Andre Raspaud, and Martin Skoviera. (2016). The chromatic number of a signed graph.
- [44] Mohamed Abdelfattah and Ahmed Shawish. (2016). Automated Academic Schedule Builder for University's Faculties. *Proceedings of the World Congress on Engineering*, (2016).
- [45] Sandeep Saharan and Ravinder Kumar. (2016). Graph Coloring based Optimized Algorithm for Resource Utilization in Examination Scheduling. *Applied Mathematics & Information Sciences*, (2016).
- [46] Ferdous M. O. Tawfiq and Kholoud Khalid S. Al-qahtani. (2016). Graph Coloring Applied to Medical Doctors Schedule. *The 10th International Conference on Advanced Engineering Computing and Applications in Sciences*, (2016).
- [47] Simon Thevenin, Nicolas Zufferey, and Jean-Yves Potvin. (2016). Graph multi-coloring for a job scheduling application. *CIRRELT*, (2016).
- [48] Bing Zhou. (2016). On the Maximum Number of Dominating Classes in Graph Coloring. *Open Journal of Discrete Mathematics*, (2016).
- [49] Serge Gaspers and Edward J. Lee. (2016). Faster Graph Coloring in Polynomial Space.
- [50] Angelini, P., Bekos, M. A., De Luca, F., Didimo, W., Kaufmann, M., Kobourov, S., Montecchiani, F., Raftopoulou, C. N., Roselli, V., and Symvonis, A. (2017). Vertex-Coloring with Defects. *Journal of Graph Algorithms and Applications*, (2017).
- [51] Murat Aslan and Nurdan Akhan Baykan. (2016). A Performance Comparison of Graph Coloring Algorithms. *International Journal of Intelligent Systems and Applications in Engineering*, (2016).
- [52] Severino F. Galán. (2017). Simple decentralized graph coloring. *Computational Optimization and Applications*, (2017).
- [53] T. Veeramakali, S. Jayashri, and S. Prabu. (2017). Intelligent dynamic spectrum allocation with bandwidth flexibility in cognitive radio network. *Cluster Computing*, 20: 1575-1586, (2017).
- [54] I. Nelson, C. Annadurai, R. Kalidoss, and B. Partibane. (2017). Mitigation of co-channel interferences in cognitive multi-carrier code division multiple access system by singular value decomposition techniques. *Cluster Computing*. DOI: 10.1007/s10586-017-0997-y (2017).
- [55] Xuesong Yan, Hanmin Liu, Zhixin Zhu, and Qinghua Wu. (2017). Hybrid genetic algorithm for engineering design problems. *Cluster Computing*, 20: 263-275.
- [56] Xiangyu Meng, Liyan Dong, Yongli Li, and WilliamW. Guo. (2017). A genetic algorithm using K-path initialization for community detection in complex networks. *Cluster Computing*, 20: 311-320.
- [57] Marappan, R. and Sethumadhavan, G. (2018). Solution to graph coloring using genetic and tabu search procedures. *Arab J Sci Eng.*, 2018, 43, 525-542. Doi: <https://doi.org/10.1007/s13369-017-2686-9>.
- [58] Marappan, R. and Sethumadhavan, G. (2020). Complexity analysis and stochastic convergence of some well-known evolutionary operators for solving graph coloring problem. *Mathematics*, 2020, 8, 303. doi: <https://doi.org/10.3390/math8030303>.