ISSN XXXX XXXX © 2017 IJESC

Research Article Volume 7 Issue No.8

Migrating Birds Optimization (MBO) Algorithm to Solve Graph Coloring Problem

Vahit Tongur[1], Erkan Ülker[2]
PhD Student[1], Associate Professor[2]
Department of Computer Engineering
Necmettin Erbakan University, Faculty of Engineering and Architecture, Konya, Turkey[1]
Selcuk University, Faculty of Engineering, Konya, Turkey[2]

Abstract:
Graph Coloring Problem is a way of coloring the vertices of a graph such that no two adjacent vertices share the same color. Its aim is to use the minimum number of color for the vertices of a graph. This problem is a member of NP-hard combinatorial optimization problems. Therefore, this problem has often been tried to be solved by meta-heuristic methods in the literature. Migrating Birds Optimization algorithm is a novel meta-heuristic algorithm to solve discrete problems. This algorithm is inspired by V formation during the migration of migratory birds. In this paper, the performance of Migrating Birds Optimization algorithm is tested on the Graph Coloring Problem data sets exist in the literature. Obtained results are compared with optimal results of related data sets.

Keywords: Migrating Birds Optimization, Graph Coloring Problem, Meta-heuristic Optimization.

I.INTRODUCTION

The Graph Coloring Problem (GCP) is a typically combinatorial optimization problem. Given a graph $G = (V, E)$ with vertex set $V= \{v_1, v_2, ...,v_n\}$ and edge set $E = \{e_1, e_2, ... e_m\}$. Also given an integer *k*which is used minimum number of different color for coloring the graph *G* with *k* such that no two adjacent vertices have the same color. This *k* is called chromatic entropy and demonstrated $\chi(G)$. Adjacency matrix $A(G)$ of *G* is an $n \times n$ symmetric binary matrix. If there is an edge between the vertices v_i and v_j, v_i and v_j are called adjacent vertices and $A_{ij} = 1$, $A_{ij} = 0$ otherwise.

Objective function of GCP can be expressed as follow:

$$min f(C) = \sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}\, p \quad (1)$$

$$p = \begin{cases} 1\ if\ (A_{ij} = 1)\ and\ (c_i = c_j) \\ 0\ if\ (A_{ij} = 0)\ or\ (c_i \neq c_j) \end{cases}$$

$$i \in \{1, \dots, n\}, j \in \{1, \dots, n\}, A_{ij} \in \{0,1\}, p \in \{0,1\}, f(C) \geq 0$$

where*n* is number of vertices, A_{ij} is a cell value of adjacency matrix $A(G)$, *p* is an integer value, c_i is color of *ith* vertex, c_j is color of *j th* vertex. According to equation 1, if $f(C) = 0$, graph *G* is painted with *k* color that no two adjacent vertices have the same color. If $f(C) = 1$, only one pair vertices (adjacent vertices) of the graph *G* are painted the same color. GCP is a member of NP-hard combinatorial optimization problems. Therefore, this problem is solved also with meta-heuristic methods. Avanthay et al. proposed an adaptation of variable neighborhood search method to the graph coloring problem [1]. Durrett et al. have presented a genetic algorithm to solving the problem of chromatic entropy, a combinatorial optimization problem related to graph coloring [2]. Eiben et al. have presented adaptive evolutionary algorithm to solve graph coloring problem [3].Fleurent and Ferland have proposed a genetic and hybrid algorithms for graph coloring [4]. Galinier and Jin-KaoHao have presented hybrid evolutionary algorithms for graph coloring [5]. Marappan and Sethumadhavan have proposed a new genetic algorithm for graph coloring [6]. Marappan et al. have used to new approximation algorithms for solving graph coloring problem [7]. Porumbel et al. have presented evaluation functions for the graph K-coloring problem [8]. Shen has solved graph coloring problem using genetic programming [9]. Hindi and Yampolskiy have proposed a genetic algorithm to solve the graph coloring problem [10]. The GCP is used to solve the real life problems such as scheduling [11], pattern matching, register allocation [12], radio frequency assignment [13], map coloring [14] and noise reduction [15] etc. In this study, we are proposed Migrating Birds Optimization (MBO) algorithm that is a novel meta-heuristic algorithm. The performance of the MBO algorithm is tested on selected some instances from the DIMACS [16] benchmark graph collection. Obtained results showed that MBO algorithm can be proposed for GCP.

II. MIGRATING BIRDS OPTIMIZATION (MBO) ALGORITHM

The MBO algorithm that is used to solve Quadratic Assignment Problem (QAP) is proposed by Duman et al. [17]. The QAP is a member of discrete problem class. The MBO algorithm is inspired by migrating birds' story. The MBO algorithm is inspired from V flight formation that during migration of migratory birds. In the real life, air turbulence generated by leader bird of the flock in the V flight formation affects birds on the left and right side of the leader bird. In this way, birds in the flock can fly longer distance with less power. Also, leader bird is most tired bird in the flock. Therefore, leader bird goes to the end of the flock for relaxing. Therefore, leader bird goes to the end of the flock for relaxing and one of the birds following it takes his place. The MBO algorithm has been inspired by this behavior of migratory birds. In the MBO algorithm, each bird in the flock represents a solution in the search space. This algorithm starts with initial solutions that are generated randomly and each solution is tried to improve with using a neighborhood structure. Number of neighbor solutions are determined according to equation 2 and 4.

$$r = 3,5,7,\dots,t;\ \forall t \in (2w+1) \quad \text{where } w \in N^{+} \tag{2}$$

$$x = 1,2,\dots,(r-1)/2 \tag{3}$$

$$u = r - x \tag{4}$$

wherer is number of neighbor solutions of leader bird, *w* is an integer value, *x* is number of shared neighbor solutions and *u* is number of neighbor solutions of the other birds except the leader bird. For each bird, these own neighbor solutions are sorted from best to worst and first (best) neighbor solution is compared with itself for improve own solution. If first neighbor solution better than own solution, this neighbor solution take the place of own solution. Remaining *x* neighbor solutions are shared with following own. Neighbor sharing for leader bird is applied both left and right lines. In other words, *x* neighbor solutions are shared left side. After, remaining *x* neighbor solutions are shared right side. Therefore, $r \geq 3$should be. For example, $r = 3$ and $x = 1$. In this case, for the leader bird, first neighbor solution is used for comparison. Second neighbor solution is shared left side and third neighbor solution is shared right side. For other birds except leader bird are generated *u(r-x)* neighbor solutions. Afterward, sharing solutions coming from in front of its are added into the own neighbor solutions. Thus, these birds have *r* neighbor solutions. These *r* neighbor solutions are sorted from best to worst and first (best) neighbor solution are used for comparison. Remaining *x* neighbor solutions are shared with solution following it. Last remaining *x* neighbor solutions are discarded. This sharing process is applied to right and left side of the flock. This sharing mechanism is a unique feature that the MBO algorithm distinguishes from other meta-heuristic algorithms. In the MBO algorithm, generated neighbor solutions by the leader bird are shared to both sides of the flock. Neighbor solutions of the other birds (except the leader bird) only are shared to own side. If the best solution of the flock is not on the leader position, its neighbor solutions are not shared to both sides. Therefore, best solution must have been at leader position for transfer to the other side. For this reason, leader bird is changed periodically. Firstly, leader bird is sent to the left side of the flock and following it on the left side takes the leader position. At the second change, leader bird is sent to the right side of the flock. This process is continued like this and it is called leader replacing. Iteration limit is determined as fitness evaluation = (problem size)^3 by Duman et al.

III. IMPLEMENTATION TO GRAPH COLORING PROBLEM (GCP) OF THE MBO ALGORITHM

Since the MBO algorithm is designed for discrete problems, the structure of the MBO is not changed to solve the GCP problem.

```
Generate random z initial population and place V formation
repeat
repeat
        Generate r neighbors for leader
        Generate r-x neighbors for other birds, except leader
        Share unused best x neighbors to foloowing birds
        Compare current solution and best neighbor solution
        for each birds
    Untilnumber of flap (f)
    Replace leader
  Untiliteration limit or find best result
Return best solution in the flock
```

Algorithm 1. Pseudocode code of the MBO algorithm

The initial population is created by assigned a randomly chosen color code to the permutation cell of each individual.Neighbor generating is performed by changing the color code of a randomly selected cell.Pseudocode code of the MBO algorithm is given in Algorithm 1.

A. *Initial Population*

As in many meta-heuristic algorithm, initial population of the MBO algorithm is generated randomly and solutions are tried to be developed each iteration. Each individual (bird) in the MBO algorithm represents a solution in the search space. These individuals have a permutation array and length of the permutation is the number of vertices in the given graph. Each cell in the permutation is assigned a color code. These codes are numbered so the number of different colors. In Figure 1, a graph that has 5 vertices is painted with 3 different colors. Vertices number and color code are given in Figure 1.

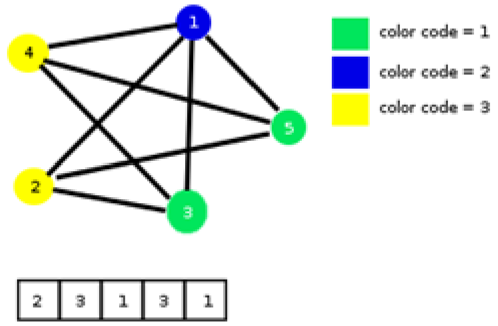

Figure.1. A sample graph

According to Figure 1 adjacency matrix *A(G)* is as follow;

Vertices	**1**	**2**	**3**	**4**	**5**
1	0	1	1	1	1
2	1	0	1	0	1
3	1	1	0	1	0
4	1	0	1	0	1
5	1	1	0	1	0

In the GCP solution with the MBO, the *A(G)* is not considered when the initial population is generated. Color codes are assigned to the permutation cells (vertex) randomly. According to Figure 1 three individuals are can be created as in Table 1;

Table.1. The permutation array and fitness of the individuals.

Individual	**Permutation**	**According to Equation 1 Fitness**
1	2 1 2 3 2	2
2	2 3 1 3 1	0
3	1 1 3 2 2	2

According to Table 1, individual 1 has two vertices that are painted the same color and are also adjacent (connected vertex). Therefore, the fitness of individual 1 is 2. The same situation is valid for individual 3. Nevertheless however, the individual 2 has a permutation that obtains an optimal result. Permutation array of the individual 2 is as such in Figure 1.

Also, an initial population that is generated "V" formation is demonstrated in Figure 2.

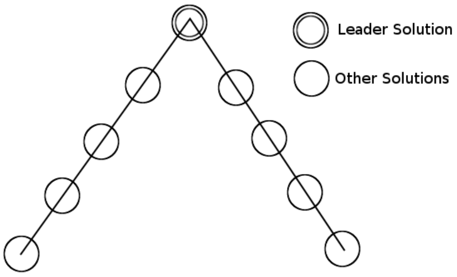

Figure.2.Initial population

B. Neighborhood Strategy

Neighborhood is used to improve a solution. For generating a neighbor (candidate) solution in the meta-heuristic algorithms that are designed to solve discrete problem are used usually swap and insertion method. In this paper, neither of these methods was used. Instead, completely random new neighbor solutions are generated. The neighbor solution is generated by assigned a randomly determined color code to a randomly determined permutation cell. Neighbor generating process for individual 3 is demonstrated in Table 2.

Table.2. Neighbor generation

Current sequence	**Current fitness**	**Randomly selected position**
1-1-3-2-**2**	2	5
Randomly selected color	**New sequence**	**New fitness**
1	1-1-3-2-**1**	3

As seen in Table 2, randomly selected cell 5 from the current permutation sequence is painted with randomly chosen color 1. The fitness value of the new individual is calculated according to Equation 1. This process is repeated r times for leader solution. Other solutions (except the leader solution) are repeated *u* times.

C. Sharing Mechanism

The sharing mechanism is the most important feature that distinguishes the MBO algorithm from other meta-heuristic algorithms. In this way, the neighbor solutions of an individual in the population can be transferred to another. The number of neighbor solutions to be shared is determined by the x parameter. In this mechanism, the neighbor solutions of the leader solution are shared to the left and right sides of the flock. Neighbor solutions of other solutions are shared only with the side on which they are located. Once the sharing process is over, each individual's neighbor solutions are sorted from best to worst and the best neighbor solutions are used to develop current solutions. The sharing mechanism is shown in Figure 3.

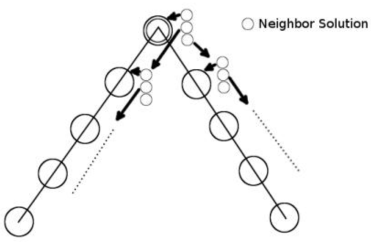

Figure.3.Sharing Mechanism

D. Leader Replacing

If the flock is always in the same order, the neighbor solution produced on any side can not be transferred to the other side. To overcome this problem, the leader solution is replaced at certain intervals. Replacing the leader solution depends on the flapping parameter (f). This parameter allows the flock to remain in the same order for a certain period of time. Leader replacing is primarily performed on the left side of the flock. The next leader replacing takes place this time on the right side of the flock. This operation is performed on the left and right sides of the flock, respectively, until the MBO algorithm is terminated. The leader replacing is shown in Figure 4.

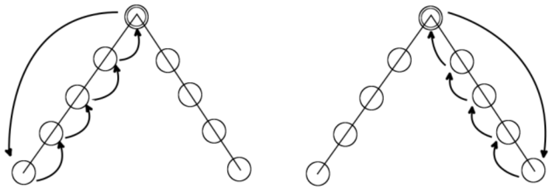

Figure.4. Leader Replacing

IV. EXPERIMENTAL RESULTS

To test the performance of the MBO algorithm were used data from the DIMACS benchmarking graph collection. In the benchmark files, (p) is the number of vertices, and edge is the number of edges:

p edge 191 2360

Also, the lines starting with (e) indicates which vertices the edges are connected with:

e 4 100

MBO have been performed on a Intel(R) Core(TM) i5-3330 CPU @ 3.00GHz processor, 4 GB RAM and Linux Ubuntu 14.04 (64-bit) operating system and the MBO is coded with QT Creator 3.0.1 gcc compiler and C++ language.

Table.3.Parameters of the MBO

Parameter	**Value**
Bird*(z)*	51
Neighbor*(r)*	3
Flap*(f)*	30
SharingNeighbor*(x)*	1

In the parameter tests of this new meta-heuristic algorithm proposed for the solution of the QAP problems by Duman et al., the parameters that number of birds (*z*), number of flapping (*f*), number of neighboring (*r*) and number of sharing (*x*) have found as 51, 10, 3, and 1, respectively [17]. In this paper, the parameters in [17] are tested for the GCP problem. The results show that the parameters in [17] can be used for the GCP problem. Unlike [17], the flap parameter is set to 30 in order to more sharing in the same sequence of the flock. Finally, iteration limit is set to 500 or it is terminated when it finds the best result. The parameters of the MBO for GCP are given in Table 3. The vertex number, edge number and chromatic entropy values of the benchmark instances used in this study are given in Table 4. Each instance was run 30 times independent and obtained results were given in Table 4.

Table.4.20 GCP benchmark instances from DIMACS and obtained best results from the MBO

Graph No	Graph Type	Instances	χ(G)	MBO	*opt/run*
1	queen5_5.col	n=25; m=320	5	5	**30/30**
2	queen6_6.col	n=36; m=580	7	7	**30/30**
3	queen7_7.col	n=49; m=952	7	7	**30/30**
4	queen8_8.col	n=64; m=1456	9	9	**30/30**
5	queen8_12.col	n=96; m=2736	12	12	**30/30**
6	queen9_9.col	n=81; m=2112	10	10	**30/30**
7	myciel3.col	n=11; m=20	4	4	**30/30**
8	myciel4.col	n=23; m=71	5	5	**30/30**
9	myciel5.col	n=47; m=236	6	6	**30/30**
10	myciel6.col	n=95; m=755	7	7	**30/30**
11	huck.col	n=74; m=602	11	11	**30/30**
12	jean.col	n=80; m=508	10	10	**30/30**
13	david.col	n=87;m=812	11	11	28/30
14	myciel7.col	n=191; m=2360	8	8	29/30
15	games120.col	n=120; m=1276	9	9	**30/30**
16	miles250.col	n=128; m=774	8	8	**30/30**
17	anna.col	n=138; m=986	11	11	**30/30**
18	miles500.col	n=128; m=2340	20	20	**30/30**
19	miles750.col	n=128; m=4226	31	31	**30/30**
20	miles1000.col	n=128; m=6432	42	42	**30/30**

Table 4 shows that various data that vertex number is from 11 to 191 and edge number is from 20 to 6432 are used. The graph type column that second column in Table 4, shows the file names of the benchmark problems used. The instances column shows the number of vertices (n) and the number of edges (m) of the problem. χ(G) is chromatic entropy (number of different color). The MBO column in the Table 4 shows the results of the best chromatic entropy obtained with the MBO algorithm. The 'opt/run' column shows the number of optimal solutions (obtained from MBO algorithm)/number of runs. As seen in Table 4, the MBO algorithm has reached the optimal result in all 30 runs except problems 13 and 14. The MBO algorithm has found chromatic entropy values as 12 and 9 for problems 13 and 14, respectively. The results from the MBO algorithm are visualized in figures 5, 6 and 7. In terms of visibility and ease of control, small problems (myciel3.col, myciel4.col and myciel5.col) are visualized.

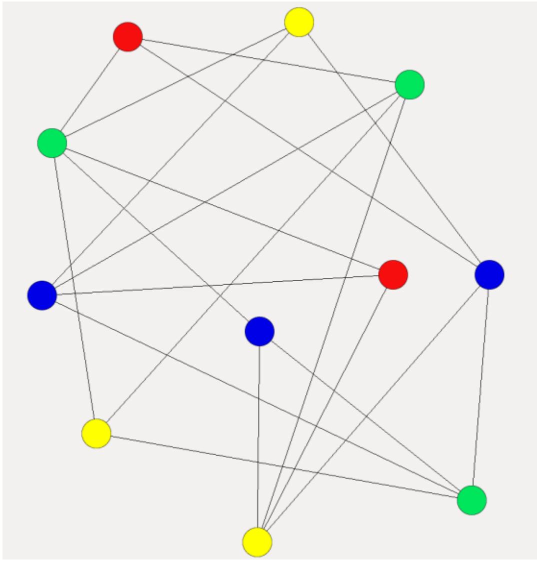

Figure.5. GCP Solution for myciel3.col

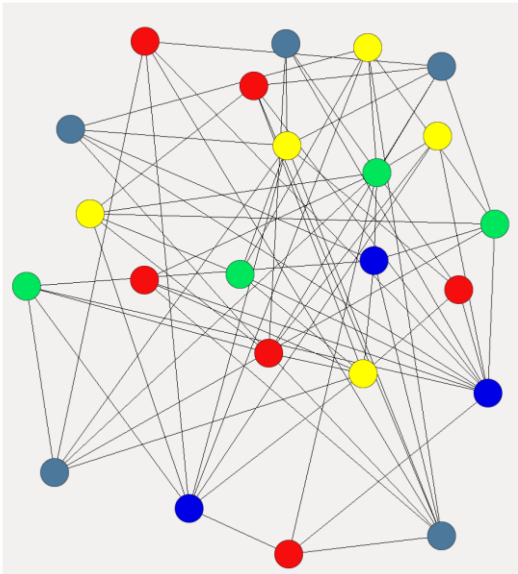

Figure.6.GCP Solution for myciel4.col

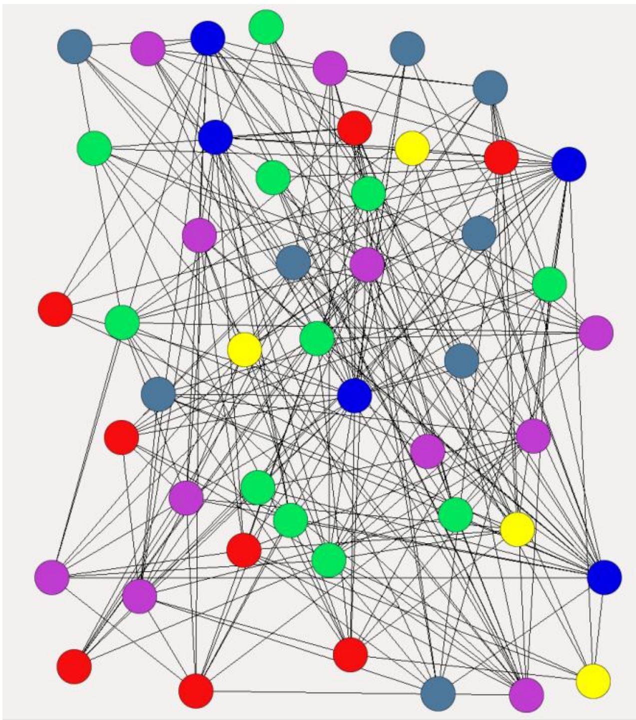

Figure.7. GCP Solution for myciel5.col

In Table 5, obtained results (χ(G)) from the MBO are compared with the GA, HPGAGCP, EM, M1 and M2 (EM, M1 and M2 are three methods used in [7]).

Table.5.Comparison Of Experimental Results Of The Mbo

Graph No	MBO	GA[10]	HPGAGCP[10]	EM[7]	M1[7]	M2[7]
1	5	5	5	5	5	5
2	7	7	8	7	7	7
3	7	7	8	7	7	7
4	9	9	10	9	9	9
5	12	NA	NA	12	15	12
6	10	NA	NA	10	11	10
7	4	4	4	4	4	4
8	5	5	5	5	5	5
9	6	6	6	6	6	6
10	7	NA	NA	7	7	7
11	11	11	11	11	11	11
12	10	10	10	10	10	10
13	11	11	11	11	11	11
14	8	NA	NA	8	8	8
15	9	9	9	9	9	9
16	8	8	8	8	8	8
17	11	11	11	11	11	11
18	20	NA	NA	20	22	20
19	31	NA	NA	31	32	31
20	42	42	42	42	48	43

As seen in Table 5, the MBO algorithm and EM mothod have reached the optimal result in all benchmark problems. NA indicates that there is no solution to the problem in the related paper. It is seen that HPGAGCP method could not reach optimal value on the problems 2, 3 and 4. Also, the M1 method could not obtain optimal results for the problems 5, 6, 18, 19 and 20. It is seen that M2 method could not reach optimal value only on the problem 20.

V. CONCLUSION

In this paper, the performance of the MBO algorithm has been tested on the Graph Coloring Problem data sets exist in the DIMACS benchmarking graph collection. Twenty different problems that different size have been selected in order to test the performance of the MBO algorithm.Each problem has been run independently 30 times.The obtained results show that the MBO algorithm has achieved the best known results on all test problems. In addition, the MBO algorithm has reached the optimum result for all 30 runs on the 18 out of the 20 problems.These results show that the neighborhood strategy used on the basic MBO algorithm is quite successful to solve the GCP.As a result, the MBO algorithm designed for QAP problems can be proposed to solve the GCP.

VI. REFERENCES

[1].Avanthay, Cédric, Alain Hertz, and Nicolas Zufferey. "A variable neighborhood search for graph coloring." European Journal of Operational Research, Vol.151,no. 2, 2003, pp. 379-388.

[2].Durrett, Greg, Muriel Médard, and Una-May O'Reilly. "A Genetic Algorithm to Minimize Chromatic Entropy." Evo COP., 2010, pp. 59-70.

[3].Eiben, Ágoston E., Jan K. Van Der Hauw, and Jano I. van Hemert. "Graph coloring with adaptive evolutionary algorithms." Journal of Heuristics,Vol. 4, no.1, 1998, pp. 25-46.

[4].Fleurent, Charles, and Jacques A. Ferland. "Genetic and hybrid algorithms for graph coloring." Annals of Operations Research, Vol. 63, no. 3, 1996, pp. 437-461.

[5].Galinier, Philippe, and Jin-Kao Hao. "Hybrid evolutionary algorithms for graph coloring." Journal of combinatorial optimization, Vol. 3, no. 4, 1999, pp. 379-397.

[6].Marappan, Raja, and Gopalakrishnan Sethumadhavan. "A new genetic algorithm for graph coloring." Computational Intelligence, Modelling and Simulation (CIMSim), 2013 Fifth International Conference on. IEEE, 2013, pp. 49-54.

[7].Marappan, Raja, Gopalakrishnan Sethumadhavan, and R. K. Srihari. "New approximation algorithms for solving graph coloring problem–An experimental approach.", Perspectives in Science 8, 2016, pp. 384-387.

[8].Porumbel, Daniel Cosmin, Jin-Kao Hao, and Pascale Kuntz. "A study of evaluation functions for the graph k-coloring problem." International Conference on Artificial Evolution (Evolution Artificielle). Springer, Berlin, Heidel berg, 2007, pp. 124-135.

[9].Shen, J. "Solving the graph coloring problem using genetic programming.", 2003.

[10].Hindi, Musa, and Roman V. Yampolskiy. "Genetic Algorithm Applied to the Graph Coloring Problem." MAICS., 2012, pp. 60-66.

[11].Marx, Dániel. "Graph colouring problems and their applications in scheduling." Periodica Polytechnica Electrical Engineering,Vol. 48.no. 1-2, 2004,pp. 11-16.

[12].Wu, Shengning, and Sikun Li. "Extending traditional graph-coloring register allocation exploiting meta-heuristics for embedded systems." Natural Computation, 2007. ICNC 2007. Third International Conference on. Vol. 4. IEEE, 2007, pp. 324-329.

[13].Singha, Satrughna, Tanmay Bhattacharya, and SR Bhadra Chaudhuri. "An approach for reducing crosstalk in restricted channel routing using graph coloring problem and genetic algorithm." Computer and Electrical Engineering, 2008. ICCEE 2008. International Conference on. IEEE, 2008, pp. 807-811.

[14].Gwee, Bah-Hwee, Meng-Hiot Lim, and Jiun-Sien Ho. "Solving four-colouring map problem using genetic algorithm." Artificial Neural Networks and Expert Systems, 1993. Proceedings. First New Zealand International Two-Stream Conference on. IEEE, 1993, pp. 332-333.

[15].Maitra, Timir, et al. "Noise reduction in VLSI circuits using modified GA based graph coloring." International Journal of Control and Automation, Vol. 3, no .2, 2010,pp. 37-44.

[16].Rutgers, The State University of New Jersey, Research, AT&T Labs, Jersey, Cancer Institute of New, University, Princeton, Labs, Alcatel-Lucent Bell, America, NEC Laboratories, and Sciences, Applied Communication. 2011. Center for Discrete Mathematics and Theoretical Computer Science (DIMACS)., 2011, Nov. 1, http://dimacs.rutgers.edu/

[17].Duman, Ekrem, Mitat Uysal, and Ali Fuat Alkaya. "Migrating Birds Optimization: A new metaheuristic approach and its performance on quadratic assignment problem." Information Sciences, Vol. 217, 2012, pp. 65-77.