5.1 Analysis of the proposed algorithms

The proposed algorithm started with creation of creation of liner link list containing n number of nodes along with the creation of an adjacency square matrix of order nXn. The running time analysis of proposed algorithms involved to find the maximum number of times instructions executed in proposed algorithm 3 & algorithms 4.

Suppose the graph on which the presented algorithms applied for coloring consists of n number of vertices and the number of colors used to color the graph is k.

Let the number of vertices in the given graph is n and the minimum number of available colors that can use to color the vertices of the graph is k. The execution time of the presented procedures can be computed formally using following computational technique available in algorithms.

(1). The list creation executed in O(n) times in assign_color ( ) procedure.

(2). The adjacency matrix formation required either O(n2) time or it may involve by reading the files from disk which is one time.

(3). The procedure assign_color( ) uses to compute the actual color value for a vertex and it requires O(n) running time.

(4). delete_node( ) procedure will executed with constant running time.

(5). update_list ( ) procedure invoked n times and the instructions for finding the adjacent vertices for currently assigned color node and updating the corresponding color field in the list will require maximum $O(n^2)$ execution time. Although in every step it will be reduced.

(6). Since the update_list( ) invoked n times so the total running time for this procedure will be $O(n^3)$.

After the analysis of all the above steps the estimated execution time by leaving the additive less significant terms for the algorithms proposed will be $O(n^3)$.

6. EXPERIMENTAL RESULTS AND DISCUSSION

The proposed algorithms in section 5 is tested on few simple graphs generated by us initially for its working. Generally, graph coloring algorithms available in literature tested against DIMACS graph coloring instances. We also tested the proposed algorithms on several graph coloring instances of DIMACS and find that in most of the cases where the graph is a kind of undirected graph, our algorithm will always comparable with the best known results.

We used Python 3.6 that reads the DIMACS graph coloring instances and converted the graph into equivalent adjacency matrix stored in the text file. The graphs that were used to test the algorithms proposed, taken from DIMACS benchmark graphs from the source [35]. This source contains different class of graphs in the form of vertex-edge list. Several graphs were taken from the above source for testing for which the description is given in the Table 2.

All above files is in form of (V, E) for a particular file. For example, myciel3.col (11, 20) file consists of eleven vertices and twenty edges in the graph. We have used Python that read the files in the above format and converted the file in the form of adjacency matrix, which stored in a text file. The converted text file for myciel4.col (23, 71) is shown in the form of adjacency matrix in Table 3.

Table 2. Test cases from source [35]

Graph Name	Description
myciel3.col (11,20)	Triangle free graphs based on Mycielski transformation which cannot solve easily with property that coloring increase in problem size.
myciel4.col (23,71)	
myciel5.col (47,236)	
myciel6.col (95,755)	
myciel7.col (191,2360)	
2-Insertions_3.col (37,720	Generalization of myciel graphs by adding more vertices for increasing the graph size but at the same without changing the density of graph.
3-Insertions_3.col (56,110)	
queen5_5.col (25,320)	For the given graph the maximum clique is not more than 5
david.col (87,812)	Graphs translated from Stanford graphbase files
huck.col (74,602)	
jean.col (80,508)	
mugg88 1.col (88,146)	Very hard 3-coloribilty instances from Kazunori Mizuno.
mugg88 25.col (88,146)	
4-Insertions_3.col (79,156)	4-Insertions graph of order 3%
1-Insertions_4.col (67,232)	1-Insertions graph of order 4%
1-FullIns_3.col (30,100)	1-FullIns graph of order 3
2-FullIns_3.col (52,201)	2-FullIns graph of order 3

Table 3. Adjacency matrix for myciel4.col (23, 71) graph

0	1	0	1	0	0	1	0	1	0	0	0	1	0	1	0	0	1	0	1	0	0	0
0	0	1	0	0	1	0	1	0	0	0	1	0	1	0	0	1	0	1	0	0	0	0
0	0	0	0	1	0	1	0	0	1	0	0	1	0	0	1	0	1	0	0	1	0	0
0	0	0	0	1	1	0	0	0	1	0	1	0	0	0	1	1	0	0	0	1	0	0
0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

We implemented the algorithm in C language for the test cases of DIMACS graph coloring instances. After converting the graph into text file we used retrieved the adjacency matrix from text file and stored the content of the matrix into a two dimensional array a[ ][ ]. The command used for retrieving the content of text file is:

FILE *in = fopen("myciel4.txt", "r");

In the above statement "in" is s file pointer and "myciel4m.txt" is the name of the file and the file opened in read mode for getting the content from the file. We created a link list corresponding to each graph with fields as shown in figure 2 with n number of nodes where n is number of vertices in the selected graph for testing. The vertex information stored in the form integer values varied from 0 to n-1. Similarly, color value for each node initialized with 0. The result obtained after running the proposed algorithms for the test cases given in Table 2 and its comparison with existing techniques for same set graph coloring instances is shown in Table 4. We used Turbo C3.2.2.0 compiler over windows7 platform (32 bit) with hardware configuration Intel Pentium D CPU clock speed 2.8 GHZ and 4 GB RAM.

We compared with MCOA technique [28] to solve the graph coloring problem and found that for most of the above mentioned test cases our algorithm works satisfactorily and produced optimal coloring solution for the tested graphs.

The table below contains the results obtained after implementing the proposed algorithms in terms of number of colors required to color a particular graph. The first column of the table contains the name of graph along with number of vertices and number of edges in the graph. The values in the second column in the table is best known number of colors corresponding to graph in the first column. The entry in the third column is our result that achieved after running the proposed algorithms on the above given data set. Finlay the last column entries are the results of technique used in Modified cuckoo optimization algorithm [28].

The number of colors for myciel graphs myciel3, myciel4, myciel5, myciel6, myciel7, based on Mycielski transformation is same in our case to best known for all the above compared graphs. It is also similar and comparable with the techniques used in [28] for graph coloring.

Similarly, we tested graph queen5_5, david, huck, jean, from Stanford GraphBase File and get result which is equal to best known color value for the graph. The results after running the proposed algorithm for the graphs mugg88.1 and mugg88.25 graphs from Kuzunori Mizuno, the number of colors required to color above graphs and obtained number of color values with our algorithm is same.

When we tested the graph 1-Insertions-4, graph of order 4 % our algorithm uses to five colors to color the entire graph however the best known optimal color value for this graph is four. But our result is similar to result in the case of MOCA technique. After testing the graph 4-Insertions-3, graph of order 3 %, our algorithm requires four colors to color the whole graph for which the best optimal color value is three. However, our result is similar to the result obtained in case of MOCA technique applied to solve the problem.

When we run our proposed algorithm on the 1-FullIns_3 graph of order 3 and 2-FullIns_ 3 graph of order 3 found that the number of colors used for the coloring of graph 1-Fullins_3 is eight and for the graph 2-Fullins3 it was ten. However, in case of MOCA [28] technique for graph coloring for these graphs, the optimal colors used is four colors and five colors, respectively.

Table 4. Comparison of proposed algorithms with MOCA [28]

Graph Name	Best Known [35]	Our Case	MOCA [28]
myciel3.col (11,20)	4	**4**	4
myciel4.col (23,71)	5	**5**	5
myciel5.col (47,236)	6	**6**	6
myciel6.col (95,755)	7	**7**	7
myciel7.col (191,2360)	8	**8**	8
2-Insertions_3.col (37,720	4	**4**	4
3-Insertions_3.col (56,110)	4	**4**	4
queen5_5.col (25,320)	5	**5**	5
david.col (87,812)	11	**11**	11
huck.col (74,602)	11	**11**	11
jean.col (80,508)	10	**10**	10
mugg88 1.col (88,146)	4	**4**	4
mugg88 25.col (88,146)	4	**4**	4
4-Insertions_3.col (79,156)	3	4	4
1-Insertions_4.col (67,232)	4	5	5
1-FullIns_3.col (30,100)	-	8	4
2-FullIns_3.col (52,201)	-	10	5

Probably for last four cases our algorithms terminated with considerable number of more colors as compared to existing one is due to approach that we are using to decide a color for a particular vertex in the graph. In our algorithm when a color is given to a vertex, all the adjacent vertices of that that node assigned temporarily one higher integral color value. Due to this there may be the vertices which are non-adjacent vertex to previous assigned colored vertex but since the color field for current vertex is already having different than its non-adjacent vertices. Therefore, the feasible color value for current vertex remains unchanged and thus it increases coloring values for successive vertices and their corresponding adjacent verities in the graph.

In spite of except for last two graphs our algorithms work efficiently and successfully find either best color values for the graphs or comparable with existing known color values. We used existing data structure link list and array for finding the solution of the problem by focusing only constraint associated with graph coloring problem. The proposed algorithm is efficient in its working because we removing every vertex from the list once it assigned a particular and therefore next vertex for coloring is always available as the first node in the list. Similarly, at the time of searching adjacent vertices for currently assigned color and modifications of their color values is also be done in optimize manner.

7. CONCLUSION AND FUTURE SCOPE

The proposed research work intends to devise a new approach to address the graph coloring problem that must able to find the exact solution for the problem. In the proposed approach we tried to give the exact solution for graph coloring problem. The proposed technique uses the link list to vertex information for coloring and adjacency matrix for graph representation. We used mentioned data structures for finding the solution because these data structures are simple and efficient in terms of required operation that performed in find the solution of the problem. Although for few cases the proposed algorithms fail work efficiently but there is no any single technique in the literature which able to find the optimal solution for each kind of graphs. The best known solutions for the different category of graphs are the result of several techniques. The proposed approach will work for any kind of graphs however for some cases like discussed in previous section if graph falls in that mentioned category may not works satisfactorily.

As the graph coloring problem is a NP-class problem so there is always hope for the optimizing the time complexity of the algorithm for solving the problem. In our proposed algorithms after incorporating some techniques for selecting the feasible colors among the previously assigned colors with minimum colour index may able to produce better results for those class of graphs which still needs to find optimal coloring solution.

REFERENCES

[1] Zufferey, N., Amstutz, P., Giaccari, P. (2008). Graph coloring approaches for a satellite range scheduling problem. Journal of Scheduling, 11(4): 263-277. https://doi.org/10.1007/s10951-008-0066-8

[2] de Werra, D., Eisenbeis, C., Lelait, S., Marmol, B. (1999). On a graph theoretical model for cyclic register allocation. Discrete Applied Mathematics, 93(2-3): 191-203. https://doi.org/10.1016/S0166-218X(99)00105-5

[3] Tilley, J.A. (2017). The *a*-graph coloring problem. Discrete Applied Mathematics, 217(2): 304-317. https://doi.org/10.1016/j.dam.2016.09.011

[4] Demange, M., Monnot, J., Pop, P., Ries, B. (2014). On the complexity of the selective graph coloring problem in some special classes of graphs. Theoretical Computer Science, 540-541: 89-102. https://doi.org/10.1016/j.tcs.2013.04.018

[5] Lu, H., Halappanavar, M., Chavarría-Miranda, D., Gebremedhin, A.H., Panyala A., Kalyanaraman, A. (2017). Algorithms for balanced graph colorings with applications in parallel computing. IEEE Transactions on Parallel and Distributed Systems, 28(5): 1240-1256. https://doi.org/10.1109/TPDS.2016.2620142

[6] Fidanova, S., Pop, P. (2016). An improved hybrid ant-local search algorithm for the partition graph coloring problem. Journal of Computational and Applied Mathematics, 293: 55-61. https://doi.org/10.1016/j.cam.2015.04.030

[7] Galinier, P., Hamiez, J.P., Hao, J.K., Porumbel, D. (2013). Recent advances in graph vertex coloring. In: Zelinka I., Snášel V., Abraham A. (eds) Handbook of Optimization. Intelligent Systems Reference Library, vol 38. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-30504-7_20

[8] Hussin, B., Basari, A.S.H., Shibghatullah, A.S., Asmai, S.A., Othman, N.S. (2011). Exam timetabling using graph colouring approach. 2011 IEEE Conference on Open Systems, Langkawi, pp. 133-138. https://doi.org/10.1109/ICOS.2011.6079274

[9] Gaceb, D., Eglin, V., Lebourgeois, F., Emptoz, H. (2009). Robust approach of address block localization in business mail by graph coloring. The International Arab Journal of Information Technology, 6(3): 221-230.

[10] Khasawneh, M.A., Malkawi, M.I., Hayajneh, T.S. (2009). A graph-coloring-based navigational algorithm for personnel safety in nuclear applications. 2009 6th International Symposium on Mechatronics and its Applications, Sharjah, pp. 1-7. https://doi.org/10.1109/ISMA.2009.5164808

[11] Selemani, M.A, Mujuni, E., Mushi, A. (2013). An examination scheduling algorithm using graph colouring- the case of Sokoine University of agriculture. International Journal of Computer Engineering & Applications, II(I/III): 116-127.

[12] Zhu, X.D., Dai, L.L., Wang, Z.C., Wang, X.D. (2017). Weighted-graph-coloring-based pilot decontamination for multicell massive MIMO systems. IEEE Transactions on Vehicular Technology, 66(3): 2829-2834. https://doi.org/10.1109/TVT.2016.2572203

[13] Méndez Díaz, I., Zabala, P. (2008). A cutting plane algorithm for graph coloring. Discrete Applied Mathematics, 156(2): 159-179. https://doi.org/10.1016/j.dam.2006.07.010

[14] Lucet, C., Mendes, F., Moukrim, A. (2006). An exact method for graph coloring. Computers & Operations Research, 33(8): 2189-207. https://doi.org/10.1016/j.cor.2005.01.008

[15] Méndez-Díaz, I., Zabala, P. (2006). A branch-and cut algorithm for graph coloring. Discrete Applied Mathematics, 154(5): 826-847. https://doi.org/10.1016/j.dam.2005.05.022

[16] Malaguti, E., Monaci, M., Toth, P. (2011). An exact approach for the Vertex Coloring Problem. Discrete Optimization 8(2): 174-190. https://doi.org/10.1016/j.disopt.2010.07.005

[17] Segundo, P.S. (2012). A new DSATUR-based algorithm for exact vertex coloring. Computer & Operations Research, 39(7): 1724-1733. https://doi.org/10.1016/j.cor.2011.10.008

[18] Caramia, M., Dell'Olmo, P. (2008). Coloring graphs by iterated local search traversing feasible and infeasible solutions. Discrete Applied Mathematics, 156(2): 201-217. https://doi.org/10.1016/j.dam.2006.07.013

[19] Porumbel, D.C., Hao, J.K., Kuntz, P. (2009). Diversity control and multi parent recombination for evolutionary graph coloring algorithms. 9th European Conference on Evolutionary Computation in Combinatorial Optimization (EVOCOP2009). Tbingen, Germany, pp. 121-132. https://doi.org/10.1007/978-3-642-01009-5_11

[20] Dowsland, K.A., Thompson, J.M. (2008). An improved ant colony optimization heuristic for graph coloring Discrete Applied Mathematics, 156(3): 313-324. https://doi.org/10.1016/j.dam.2007.03.025

[21] Prestwich, S.D. (2008). Generalized graph coloring by a hybrid of local search and constraint programming. Discrete Applied Mathematics, 156(2): 148-158. https://doi.org/10.1016/j.dam.2006.07.011
[22] Zhou, Z.Y., Li, C.M., Huang, C., Xu, R.C. (2014). An exact algorithm with learning for the graph coloring problem. Computers & Operation Research, 51: 282-301. https://doi.org/10.1016/j.cor.2014.05.017
[23] Xu, J., Qiang, X.L., Zhang, K., Zhang, C., Yang, J. (2018). A DNA computing model for the graph vertex coloring problem based on probe graph. Engineering, 4(1): 61-77. https://doi.org/10.1016/j.eng.2018.02.011
[24] Shukla, A.N., Garg, M.L., Misra, R. (2019). An approach to solve graph coloring problem using linked list. International Journal of Advanced Studies of Scientific Research, 4(2).
[25] Shukl, A.N., Garg, M.L. (2018). A list based approach to solve graph coloring problem. 2018 International Conference on System Modeling & Advancement in Research Trends (SMART), Moradabad, India, pp. 265-267. https://doi.org/10.1109/SYSMART.2018.8746966
[26] Shukla, A.N., Garg, M.L. (2019). An approach to solve graph coloring problem using adjacency matrix. Oryzae. Biosc. Biotech. Res. Comm., 12(2). http://dx.doi.org/10.21786/bbrc/12.2/33
[27] Shukla, A.N., Bharti, V., Garg, M.L. (2019). An algorithm based on heap of binary search tree to solve graph coloring problem. International Journal of Recent Technology and Engineering (IJRTE), 8(2): 3920-3924.
[28] Mahmoudi, S., Lotfi, S. (2015). Modified cuckoo optimization algorithm (MCOA) to solve graph coloring problem. Applied Soft Computing, 33: 48-64. https://doi.org/10.1016/j.asoc.2015.04.020
[29] Parihar, A., Shukla, N., Jerry, M., Datta, S., Raychowdhury, A. (2017). Vertex coloring of graphs via phase dynamics of coupled oscillatory networks. Scientific Reports, 7: 11. https://doi.org/10.1038/s41598-017-00825-1
[30] Arumugam, S., Premalatha, K., Bača, M., Semaničová-Feňovčíková, A. (2017). Local antimagic vertex coloring of a graph. Graphs and Combinatorics, 33(2): 275-285. https://doi.org/10.1007/s00373-017-1758-7
[31] Rezapoor Mirsaleh, M., Meybodi, M.R. (2016). A new memetic algorithm based on cellular learning automata for solving the vertex coloring problem. Memetic Computing, 8(3): 211-222. https://doi.org/10.1007/s12293-016-0183-4
[32] Labed, S., Kout, A., Chikhi, S. (2018). Solving the graph b-coloring problem with hybrid genetic algorithm. 2018 3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS), Tebessa, pp. 1-7. https://doi.org/10.1109/PAIS.2018.8598525
[33] Aragón Artacho, F.J., Campoy, R. (2018). Solving graph coloring problems with the Douglas-Rachford algorithm. Set-Valued and Variational Analysis, 26(2): 277-304. https://doi.org/10.1007/s11228-017-0461-4
[34] Bahiense, L., Frota, Y., Noronha, T.F., Ribeiro, C.C. (2014). A branch-and-cut algorithm for the equitable coloring problem using a Formulation by representatives. Discrete Applied Mathematics, 164: 34-46. https://doi.org/10.1016/j.dam.2011.10.008
[35] https://turing.cs.hbg.psu.edu/txn131/graphcoloring.html #XXCAR, accessed on 10 March 2019.