Global Analysis and Discrete Mathematics
Volume 5, Issue 1, pp. 67–74
ISSN: 2476-5341

A Practical Algorithm for $[r, s, t]$-Coloring of Graph

Hajar Alimorad*

Received: 11 August 2019 / Accepted: 6 June 2020

Abstract Coloring graphs is one of important and frequently used topics in diverse sciences. In the majority of the articles, it is intended to find a proper bound for vertex coloring, edge coloring or total coloring in the graph. Although it is important to find a proper algorithm for graph coloring, it is hard and time-consuming too. In this paper, a new algorithm for vertex coloring, edge coloring and $[r, s, t]-$ coloring is presented. Then, this algorithm is used to solve the applied problems of eight-queens and $[r, s, t]-$ coloring. Here, there are numerical examples to study the efficiency of the method and to compare the results.

Keywords Adjacency matrix · Eight queen puzzle · Linear graph · Matching · $[r, s, t]$- Coloring

Mathematics Subject Classification (2010) 05C15 · 05C57 · 94C15

1 Introduction

Suppose that $G(V, E)$ is a graph with the set of vertices V and the set of edges E. Coloring the graph is a special state of the marking graph problems. Its general approach is to use matching some colors to the edges or the vertices while the coloring follows a special limitation. In the simplest state, a graph coloring in which there is no two adjacent edges with the same color is called a permissible painting. The minimum number of colors needed in a G- coloring is called the chromatic index and is shown as $\chi^{'}(G)$. Graphs coloring has many

*Corresponding author
H. Alimorad
Dept. of Mathematics, Jahrom University, P.O. Box: 74135-111, Iran.
E-mail: h.alimorad@jahromu.ac.ir

© 2020 Damghan University. All rights reserved. http://gadm.du.ac.ir/

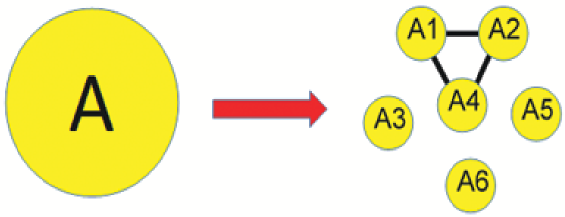

Fig. 1: divide "A" into six sub-duties from "A_1" to "A_6"

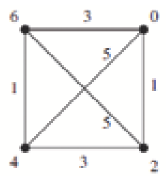

Fig. 2: $[2, 2, 1]$-coloring with seven colors of the complete graph K_4

usages in various practical and theoretical fields. In addition to the classical problems defined in this field, various problems with wide usages in industry and science are defined and solved, by designating different limitations on the graphs, the coloring method and even the number of colors for the elements of the graph. For example, the problem of radio channel allocation can be modelized by a special kind of coloring for the vertices of the graphs, called **λ-coloring** ([1]). Solving the scheduling problems is one of the applications for the edges-coloring of a graph ([2]).

As another example, suppose that the time for performing a duty like "A" is "T". We divide "A" into six sub-duties from "A_1" to "A_6" so that each takes $t = \frac{T}{6}$ time to reduce the performance time of the duty "A".

All the duties can not perform parallelly, the dependence between duties can be shown by drawing an edge between two corresponding vertices. By doing this, scheduling problem turns into a problem of graph coloring. The performing time of "A" is the product of multiplying the chromatic index of "G" into "t". Another type of graph-coloring is coloring the edges and the vertices at the same time.

Given non-negative r, s and t, an $[r, s, t]$-coloring of a graph G with vertex set $V(G)$ and edge set $E(G)$ is a mapping c from $V(G) \bigcup E(G)$ to the color set $\{0, 1, ..., k-1\}$ such that $|c(v_i)-c(v_j)| \geq r$ for every two adjacent vertices v_i, v_j, $|c(e_i) - c(e_j)| \geq s$ for every two adjacent edges e_i, e_j and $|c(v_i) - c(e_j)| \geq t$ for all pairs of incident vertices and edges, respectively. The $[r, s, t]$-Chromatic number $\chi_{r,s,t}(G)$ of G is defined to be the minimum k such that G admits an

$[r, s, t]$-coloring.
Obviously, a $[1, 0, 0]$-coloring is an ordinary vertex coloring, a $[0, 1, 0]$-coloring is an edge coloring, and a $[1, 1, 1]$-coloring is a total coloring ([3]).
Figure (2) shows as examples a $[2, 2, 1]$-coloring with seven colors of the complete graph K_4 ([3]).
In most of the articles, we aim to find a proper bound for vertex coloring, edge coloring or total coloring in the graph. It is important to find a proper algorithm for graph coloring, but it is also hard and time-consuming. Therefore, in this article, we will present a new algorithm for vertex coloring, edge coloring and $[r, s, t]$- coloring. Then, we will use this algorithm to solve the applied problems of eight-queens and $[r, s, t]$- coloring.

2 Introducing the algorithm

In this part, a new algorithm will be explained for graph coloring.

Definition 1 Adjacency matrix of graph: For a simple graph with vertex set V, the adjacency matrix is a square $|V|\times|V|$ matrix A such that its element A_{ij} is one when there is an edge from vertex i to vertex j, and zero when there is no edge ([4]).

Definition 2 The incidence matrix of graph: Any graph "G has a correspondent matrix of $|V| \times |E|$, which is called the incident matrix of G. In the incident matrix of $M(G) = [m_{ij}]$, in which m_{ij} is the number of times (0 , 1 or 2), the vertex v_i is vertical to e_j.

Definition 3 Linear graph $L(G)$: The linear graph $L(G)$ in the simple graph G, is a graph whose vertices are correspondent to the edges of G in a one-to-one manner. Two vertices in $L(G)$ are adjacent if and only if their correspondent edges in G are adjacent.

Definition 4 Suppose that G is a graph. The subset S from the vertices of G is called independent subset when no pair of vertices in S are adjacent.

2.1 The algorithm of edge coloring of a graph using adjacency matrix $L(G)$

With the help of adjacency matrix of a simple graph, we can easily detect the independent vertices set.
The sets of independent edges can be defined by adjacency matrix of $L(G)$.
In the graph theory, a matching or independent edge set is a set of edges without common vertices.
In this part, we will define an algorithm for finding the adjacency matrix of $L(G)$.
First, we will write an algorithm to define the matrix of $M(G)$ and then we will find the matrix of $L(G)$.
Pseudocode:

1. Given the matrix A,
2. n (the number of the rows) is equal to the number of the rows in the matrix A. Calculate m (the number of the columns) like $m = \frac{sum(sum(A))}{2}$. For $j > i$ and beginning with $i = 1$,
   If $A_{ij} = 1$ let $k = 1$:
   Let the element of k-th column in the matrix $M(G)$ in the row i and j be 1 and the other elements of the column be zero.
3. Put $k = k + A_{ij}$,
   Repeat the 3rd step.

The adjacency matrix of graph $L(G)$ is calculated in the following way,

$$L = M^T M - 2I$$

In which I is the identity matrix ([5]).
For the edge coloring of the graph, after specifying the independent sets of the edges, we will color all the elements of each set with one color, beginning with the first color, and the number of the independent sets of $\chi^{'}(G)$ (the chromatic index of edges).

2.2 The eight-queen puzzle and the edge coloring of a graph

The eight-queen puzzle is the problem of placing eight chess queens on an 8×8 chessboard so that no two queens threaten each other; thus, a solution requires that no two queens share the same row, column, or diagonal. The eight-queen puzzle is an example of the more general n-queen problem of placing n non-attacking queens on an $n \times n$ chessboard ([6]). It's not affordable at all to study all the possible ways to line the queens up for reaching the right way of setting. On the other hand, we do not have a special answer for each n. The number of all the positions is equal to $C(8^2, 8)$, which is a large number. In this part, we will solve this problem with edge coloring of a graph. We assume that we have a complete bipartite graph like $K_{8\times8}$. The edges with different colors in the graph show the edge or vertex intersection between the quarters. Here, we need an 8-matching of the edges with the same color. There is an additional term here because the queens do not have equal diagonals. In sets of eight edges with the same color, there is at most only one element picked from each set of S_1 and S_2.

$$S_1 = \{i = j | i = 1, 2, ..., 8, \qquad j = 1, 2, ..., 8\}$$
$$= \{(1, 1), (2, 2), (3, 3), (4, 4), (5, 5), (6, 6), (7, 7), (8, 8)\}$$

$$S_2 = \{i + j = 9 | i = 1, 2, ..., 8, \qquad j = 1, 2, ..., 8\}$$
$$= \{(1, 8), (2, 7), (3, 6), (4, 5), (5, 4), (6, 3), (7, 2), (8, 1)\}$$

That is how we find a set of 8-matching with the help of the $L(G)$ graph. 2-independent sets, is the number of zeros above the main diagonal of $L(G)$

because these zeros indicate the two independent edges.
To find 3-matching, we act as follows, if

$$L(i, j) = L(j, k) = L(k, i) = 0$$

for $i < j < k$, then $\{i, j, k\}$ specifies a triple set of independent edges.
In general, for finding k-independent sets ($k \leq n$), we use all of $(k-1)$-independent sets. In this case, we should find j_k such that

$$L(j_1, j_2) = L(j_2, j_3) = L(j_3, j_4) = ... = L(j_{k-1}, j_k) = 0$$

when $\{j_1, j_2, ..., j_{k-1}\}$ is an independent set.
In this way, we find a set of 8-matching with the help of the $L(G)$ and there's at most only one element picked from each set of S_1 and S_2. These sets determine the order of lining the queens on the chessboard. The number of all answers is 92. Some of the answers are in the following:

$$\{(1,7),(2,8),(3,5),(4,4),(5,3),(6,1),(7,2),(8,6)\}$$
$$\{(1,8),(2,6),(3,5),(4,7),(5,3),(6,4),(7,1),(8,2)\}$$
$$\{(1,8),(2,6),(3,5),(4,4),(5,2),(6,1),(7,3),(8,7)\}$$
$$\{(1,7),(2,8),(3,5),(4,1),(5,3),(6,2),(7,4),(8,6)\}$$
$$\{(1,7),(2,6),(3,8),(4,3),(5,1),(6,5),(7,2),(8,4)\}$$

2.3 $[r, s, t]$- Coloring of graph with the help of adjacency matrix of $A(G)$ and matrix of $M(G)$

As it was explained in the first part, we have the condition $|c(v_i) - c(v_j)| \geq r$, for the two adjacent vertices v_i, v_j, the condition $|c(e_i) - c(e_j)| \geq s$, for the adjacent edges e_i, e_j and the condition $|c(v_i) - c(e_j)| \geq t$, for the adjacent vertex and edge v_i, e_j. Therefore, we use the algorithm in the following way,
Pseudocode:

1. In the adjacency matrix of $A(G)$ for $j > i$,
   Beginning with vertex v_1, we define all the independent sets of vertices. All the vertices of the first set get the color zero and all the vertices in the i-th set respectively get color $0 + (i-1)r$ (The first condition).
2. Beginning with vertex v_1 and with the help of incidental matrix of $M(G)$, in the first row, the first entries define the adjacent edges in v_1. They get colored respectively like the following:

   $$t, t+s, t+2s, ...$$

   Also, for other vertices, we color the edges which are not colored yet with colors $t, t+s, t+2s, ...$, provided that the color of the edge in that vertex is not repetitive.

In article ([6]), the authors have found some bounds for $\chi_{r,s,t}(G)$.

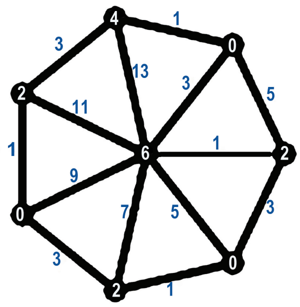

Fig. 3: $[2, 2, 1]$-coloring, $\chi_{r,s,t}(G) = 14$

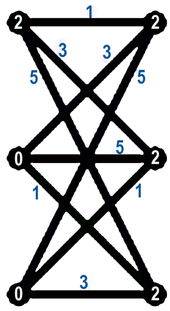

Fig. 4: $[2, 2, 1]$-coloring, $\chi(G) = 2$, $\chi'(G) = 3$ and $\chi_{r,s,t}(G) = 6$

Theorem 1

$max\{r(\chi(G)-1)+1, s(\chi'(G)-1)+1, t+1\} \leq \chi_{r,s,t}(G) \leq r(\chi(G)-1)+s(\chi'(G)-1)+t+1$

(see ([6]) for more details).

Theorem 2 *If* $m = min\{r, s, t\}$ *and* $M = max\{r, s, t\}$ *then*

$$m\Delta(G) + 1 \leq \chi_{r,s,t}(G) \leq M(\Delta(G) + c1) + 1$$

with a constant $c \geq 2$ *([6]).*

In this part, we will solve some examples of $[r, s, t]$- coloring with the help of the presented algorithm and will show the results for theorem (1) and (2). According to the first theorem, since $\chi(G) = 4, \chi'(G) = 7$, then $13 \leq \chi_{r,s,t}(G) \leq 20$. According to the second theorem, we have $8 \leq \chi_{r,s,t}(G) \leq 17$ for $c = 2$. In other examples, the reader can also compare the value of $\chi_{r,s,t}(G)$ by the represented algorithm, with the bounds given in theorem (1) and (2).

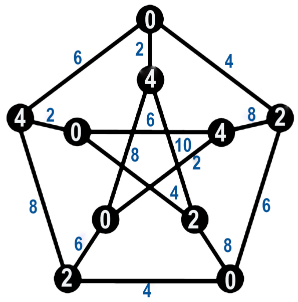

Fig. 5: $[2, 2, 2]$-coloring, $\chi(G) = 3$, $\chi'(G) = 4$ and $\chi_{r,s,t}(G) = 11$

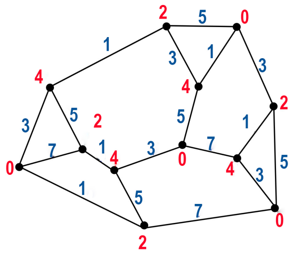

Fig. 6: $[2, 2, 1]$-coloring, $\chi(G) = 3$, $\chi'(G) = 3$ and $\chi_{r,s,t}(G) = 8$

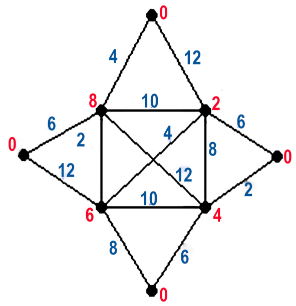

Fig. 7: $[2, 2, 2]$-coloring, $\chi(G) = 4$, $\chi'(G) = 5$ and $\chi_{r,s,t}(G) = 13$

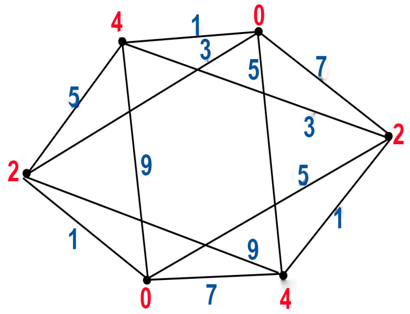

Fig. 8: [2, 2, 1]-coloring, $\chi(G) = 3$, $\chi'(G) = 5$ and $\chi_{r,s,t}(G) = 10$

3 Conclusion

In this article, there is an optimal algorithm represented for vertex and edge coloring of graphs, using the properties of the adjacency matrix of graphs and linear matrix of graphs. This algorithm is used for edge coloring, vertex coloring and $[r, s, t]$-coloring.

4 Suggestions for future research

In writing the code algorithm, for finding the independent vertex and edge sets, we used the "for-loop". Although the time for operating the program is short, the main problem is that the number of vertices and edges of a graph must already be defined, and the number of the "for-loop" for finding the matching depends on the number of vertices and edges. If we could write the computerized algorithm for finding the independent sets without the necessity for knowing the number of the vertices and the edges, this algorithm would be more optimal.

References

1. D. Kral and R. Shrekovski, A theorem about channel assignment problem, SIAM Journal on Discrete Mathematics, 16 (3): 426–437, (2003).
2. U. Schauz, The tournament scheduling problem with absences, European Journal of Operational Research, 254(3): 746–754, (2016).
3. A. Kemnitz and M. Marangio, $[r, s, t]$-Colorings of graphs, Discrete Mathematics, 307: 199–207, (2007).
4. N. Biggs, Algebraic Graph Theory, Cambridge Mathematical Library (2nd Ed), 2.1, p.7, Cambridge University Press, (1993).
5. S. Skiena, Line Graph, Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica, 128 and 135-139, Reading, MA: Addison-Wesley (1990).
6. E.J. Hoffman, J.C. Loessi, and R.C. Moore, Construction for the solution of the m queens problem, Mathematics Magazine, 42: 66–72, (1969).