Received: August 26, 2023; **Accepted:** January 30, 2024.
DOI. 10.30473/coam.2023.68880.1244
Winter-Spring (2024) Vol. 9, No. 1, (185-194)
Research Article

Open Access

Control and Optimization in Applied Mathematics - COAM

A Hybrid Floyd-Warshall and Graph Coloring Algorithm for Finding the Smallest Number of Colors Needed for a Distance Coloring of Graphs

Hanifa Mosawi[1], Mostafa Tavakoli[1], Khatere Ghorbani-Moghadam[2]

[1]Department of Applied Mathematics, Faculty of Mathematical Sciences, Ferdowsi University of Mashhad, P.O. Box 1159, Mashhad 91775, Iran.
[2]Mosaheb Institute of Mathematics, Kharazmi University, Tehran, Iran.

Correspondence:
Mostafa Tavakoli
E-mail:
m_tavakoli@um.ac.ir

How to Cite
Mosawi, H., Tavakoli, M., Ghorbani-Moghadam, Kh. (2024). "A Hybrid floyd–warshall and graph coloring algorithm for finding the smallest number of colors needed for a distance coloring of graphs", Control and Optimization in Applied Mathematics, 9(1): 185-194.

Abstract. Graph coloring is a crucial area of research in graph theory, with numerous algorithms proposed for various types of graph coloring, particularly graph p-distance coloring. In this study, we employ a recently introduced graph coloring algorithm to develop a hybrid algorithm approximating the chromatic number p-distance, where p represents a positive integer number. We apply our algorithm to molecular graphs as practical applications of our findings.

Keywords. p-distance coloring, p-distance chromatic number, Graph adjacency matrix, Hybrid algorithm.

MSC. 05C15, 05C76, 05C38.

https://mathco.journals.pnu.ac.ir

1 Introduction

All graphs considered in this paper are simple. Let $G = (V, E)$ be a graph. The distance between two vertices u and v, denoted by $d(u, v)$, is the length of a shortest connecting path between u and v. If there is no path between u and v, we set $d(u, v) = \infty$. The diameter of G, denoted by $D(G)$, is defined as $\max\{d(u, v) : u, v \in V(G)\}$. The graph adjacency matrix for the Floyd–Warshall algorithm is made as below:

$$A(i,j) = \begin{cases} 1 & \text{If there is an edge between } v_i \text{ and } vj, \\ \infty & \text{If there is no edge between } v_i \text{ and } vj, \\ 0 & \text{if } i = j. \end{cases}$$

Different types of graph coloring have emerged in response to various applications, including normal, dominant, recessive, star, p-distance, and others. Graph coloring in a graph $G = (V, E)$ is a function that maps the set V to a set $C = \{1, 2, \ldots, k\}$ of colors, ensuring that adjacent vertices receive different colors. The chromatic number of a graph G, denoted $\chi(G)$ (refer to [21]), is the minimum number of distinct colors required to color G. Graph distance coloring was first introduced in 1969 [16, 17]. The p-distance coloring of a graph $G = (V, E)$ is a mapping from V to a set of colors, where vertices with a distance of at most p receive different colors. The p-distance chromatic number, denoted $\gamma(p, G)$, represents the minimum number of distinct colors needed for this coloring. For a positive integer p, p is the power of G if $G^p = (V, E_p)$ is a graph with the vertex set $V(G)$ and the edge set $\{uv : u, v \in V(G) \text{ and } d_G(u, v) \leq p\}$. It is evident that $\gamma(p, G) = \chi(G^p)$.

The p-distance coloring has various applications, including solving frequency assignment problems such as radio channel allocation [14, 19]. This problem arises when multiple radio transmitters, such as (mobile phones, operate in the same area and share the same or nearby transmitter channels. To prevent wave interference, the problem of allocating frequencies to different transmitters can be reduced to a graph coloring problem, which can be solved by p-distance coloring of the network. In [8], Fertin et al. simulated the network graph when the transmitters are regularly broadcast on the plane and solved the problem by p-distance coloring of the network.

The p-distance problem has been studied by various researchers since the seventies, including Kramer [15], Speranza [22], and Antonucci [2]. In the eighties, it was also studied by Gionfriddo [12] and Gionfriddo and Milici [13] in the nineties. Recent articles have reviewed these topics [1, 5, 6, 17]. The distance coloring parameters of graphs have been researched in general [18], and the 2-distance chromatic number of some graph products has been investigated in [11]. Additionally, in [5], 2-distance coloring of distance graphs has been studied. To find the p-distance color for a given graph, we use the Floyd–Warshall algorithm [9] with the GCA graph coloring algorithm proposed in the [20], which we refer to as $GDCA$.

The continuation of the pper is arranged as follows: Section 2 introduces the $GDCA$ algorithm and provides an example to illustrate its process. Section 3 presents a table of algorithm results on several benchmark graphs found in the $DIAMCS$ library [7] is presented. Subsequently, in Section 4, we apply our algorithm to calculate the chromatic number of two molecular graphs from [17].

2 Graph p-Distance Coloring Algorithm

To present our algorithm, we need to recall the Floyd–Warshall and graph coloring algorithms (GCA). The Floyd–Warshall algorithm receives the adjacency matrix A constructed according to the definition and returns the distance matrix.

The GCA colors simple graphs without loops and multiple edges, undirected, connected or non-connected, and finite, using the graph adjacency matrix. The graph adjacency matrix for the GCA

Algorithm 8 Floyd–warshall algorithm

Input The graph adjacency matrix (A), the number of rows (n) of matrix A.
Output: Return D^n.
step-1 $D^0 = A$

1-1: **While** $k < n$ **do**:

1-1-1: Put $D^{(k)} = (d_{ij}^{(k)})$, let a new matrix be $n \times n$.
While $i < n$ **do**:
While $j < n$ **do**:
$(d_{ij}^{(k)}) = \min((d_{ij}^{(k-1)}), (d_{ik}^{(k-1)}) + (d_{kj}^{(k-1)}))$.

algorithm is made as below:

$$A(i,j) = \begin{cases} 1 & \text{If there is an edge between } v_i \text{ and } v_j, \\ 0 & \text{otherwise.} \end{cases}$$

Algorithm 9 Graph Coloring Algorithm (GCA)

Input A is adjacency matrix of graph $G = (V, E)$.
Output: Return $x(k)$, where it is sets of separation and k is number of sets.
Step-1 Put $k = 0, n = A.rows, V = 1, \dots, n, x = 1, \dots, n$.
Step-2 **While** $V \neq \phi$ **Do**:

2-1 **While** $sum(sum(A'))' > 0$ **DO**:

2-1-1 Put i as the smallest row which the sum of it, is not zero.

2-1-2 Put i neighbors in the w.

2-1-3 Put $t = \cup(i, w)$ and order t.

2-1-4 Remove rows and columns contain array t from the largest to the smallest from A and V.

2-2 Put $k = k + 1$; put union (i)'s and the only vertices in t; sort; and the members of this set are the index of the members of the set x, so take the corresponding numbers of this set from the largest to the smallest index of x and put them in x_k.

2-3 Put c equals to the neighbors of (i)'s.

2-4 **If** $|c| = 1$, **Then** $k = k + 1$ and $x(k) = x$ and the algorithm terminates.

2-5 **If** $|c| > 1$, **Then** set the B as the adjacency matrix of the induced subgraph $G[c]$. The matrix B is constructed in such a way that the algorithm deletes the rows and columns of the matrix A according to the ordered set t, from the largest to the smallest, and the matrix B exists.

2-6 **If** $sum(sum(B'))' = 0$, **Then** $k = k+1$ and $x(k) = x$ and the algorithm terminates, **otherwise** $A = B$ and $V = 1, \dots, length(c)$, **Then** go to **Step 2**.

To better comprehend the (GCA) algorithm, it is necessary to explain the concepts of $sum(sum(A'))'$ and $sum(sum(B'))'$. To determine whether the matrix A is zero or not, the algorithm calculates the transpose of the matrix A, denoted as A'. Furthermore, $sum(A')$ represents a vector obtained by summing each row of A, and also, $sum(sum(A'))'$ denotes the sum of the elements in the vector $sum(A')$. If $sum(sum(A'))' > 0$ is greater than zero, the algorithm proceeds with the subsequent commands.

Similarly, to determine the zero nature of matrix B, the algorithm calculates the transpose of matrix B, denoted as B'. Additionally, $sum(B')$ represents a vector obtained by summing each row of B, and $sum(sum(B'))'$ denotes the sum of the elements in the vector $sum(B')$. If $sum(sum(B'))'$ equals zero, it indicates that matrix B is indeed a zero matrix. The $GDCA$ utilizes the Floyd–Warshall algorithm and the GCA graph coloring algorithm. Initially, we construct the graph's adjacency matrix, denoted as A, according to the given definition. The Floyd–Warshall algorithm takes this adjacency matrix as an input and generates the distance matrix, denoted as D. To create the adjacency matrix G^p using the distance matrix, we perform as follow:

$$D(i,j) = \begin{cases} 1, & \text{If } 1 < D(i,j) \leq p, \\ 0, & \text{otherwise.} \end{cases}$$

Now, the constructed adjacency matrix G^p is obtained using the coloring algorithm GCA. The output of this algorithm represents the coloring of the graph G^p, thereby providing the p-distance coloring for the graph G. The algorithm for p-distance graph coloring is as follows:

Algorithm 10 Graph p-distance coloring algorithm $(GDCA)$ to obtain sets of separation

Input Graph adjacency matrix for Floyd–Warshall algorithm, (A) and p.

Output: Return x_k, where it is a set of separations and k is number of sets while the graph G is colored p-distance.

Step 1. $D = Floyd - Warshall(A)$, Obtain the distance matrix using the Floyd–Warshall algorithm.

Step 2. Build the adjacency matrix G^p as follow:

If $1 < D(i,j) \leq p$, **Then** $D(i,j) = 1$, **Otherwise** $D(i,j) = 0$.

Step 3. Set $A := D$.

Step 4. Execute the algorithm GCA on A.

Instead of utilizing the GCA coloring algorithm, proposed in [20], our proposed algorithm, $GDCA$ can incorporate other coloring algorithms mentioned in [3].

In step 2, the adjacency matrix of the graph G^p is created, and by applying graph coloring to this graph, the p-distance coloring for the original graph G is achieved. In this process, we have reduced the p-distance graph coloring problem to the graph coloring problem. Since the graph coloring problem is known to be *NP-hard* [10], it follows that the p-distance graph is also *NP-hard*.

2.1 Illustrative Example

This example includes the colorings of four graphs $G^1 = (V, E_1), G^2 = (V, E_2), G^3 = (V, E_3), G^4 = (V, E_4)$, which have been obtained from graph $G = (V, E)$ with $|V| = 7$ and $|E| = 5$, by the GDCA algorithm. The graphs of G^p, where $1 \leqslant p \leqslant 4$, obtained by the algorithm are shown in Figure 1.

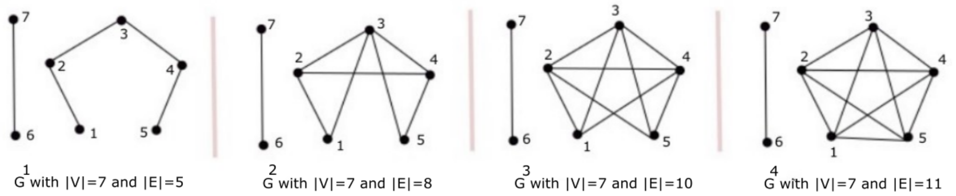

Figure 1: $G = (V, E)$ with $|V| = 7$ and $|E_p| = 5, 8, 10, 11$ for $p = 1, 2, 3, 4$.

In step 0, the adjacency matrix of $G = (V, E)$ with $|V| = 7$ and $|E| = 5$, for p in which $1 \leqslant p \leqslant 4$ is given to the algorithm as follows:

$$A = \begin{bmatrix} 0 & 1 & \infty & \infty & \infty & \infty & \infty \\ 1 & 0 & 1 & \infty & \infty & \infty & \infty \\ \infty & 1 & 0 & 1 & \infty & \infty & \infty \\ \infty & \infty & 1 & 0 & 1 & \infty & \infty \\ \infty & \infty & \infty & 1 & 0 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & 0 & 1 \\ \infty & \infty & \infty & \infty & \infty & 1 & 0 \end{bmatrix}$$

In step 1, the distance matrix for the graph $G = (V, E)$ with $|V| = 7$ and $|E| = 5$ is obtained by the Floyd–Warshall algorithm. In step 2, the adjacency matrix G^p is created. In step 3, we set $A = D$, and in step 4, the coloring algorithm GCA receives and colors the adjacency matrix A and the output of this algorithm. The sets of separation are x_k in the graph G, which is a separated p-distance and k represents the number of obtained sets or coloring number. The value of k is zero at the beginning. At each step, when the coloring of graph is finished with one color, the value of k is incremented by one. The set x_k is colored with the kth color, and the largest k in x_k is the number of different colors that is used to color the graph.

For 1-distance coloring, the GDCA algorithm receives A and $p = 1$. It generates the matrix D and using the matrix D, it creates the adjacency matrix G^1 and puts it in A. Then the GCA algorithm receives A and x_1 and x_2 produces as follows: At this stage, the number of edges of G^1 is equal to 5.

$$D = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & \infty & \infty \\ 1 & 0 & 1 & 2 & 3 & \infty & \infty \\ 2 & 1 & 0 & 1 & 2 & \infty & \infty \\ 3 & 2 & 1 & 0 & 1 & \infty & \infty \\ 4 & 3 & 2 & 1 & 0 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & 0 & 1 \\ \infty & \infty & \infty & \infty & \infty & 1 & 0 \end{bmatrix}, \qquad A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

$$|E_1| = 5, \quad x_1 = [1, 3, 5, 6], \quad x_2 = [2, 4, 7].$$

For 2-distance coloring, the GDCA algorithm receives A and $p = 2$. It generates the matrix D. Using the matrix D, it creates the adjacency matrix G^2 and puts it in A. Then the GCA algorithm receives A and x_1, x_2, and x_3 produces as follows: At this stage, the number of edges of G^2 is equal to 8.

$$D = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & \infty & \infty \\ 1 & 0 & 1 & 2 & 3 & \infty & \infty \\ 2 & 1 & 0 & 1 & 2 & \infty & \infty \\ 3 & 2 & 1 & 0 & 1 & \infty & \infty \\ 4 & 3 & 2 & 1 & 0 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & 0 & 1 \\ \infty & \infty & \infty & \infty & \infty & 1 & 0 \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

$$|E_2| = 8, \quad x_1 = [1, 4, 6], \quad x_2 = [2, 5, 7], \quad x_3 = 3.$$

For 3-distance coloring, the GDCA algorithm receives A and $p = 3$. It generates the matrix D. Using the matrix D, it creates the adjacency matrix G^3 and puts it in A. Then GCA algorithm receives A and x_1, x_2, x_3, and x_4 produces as follows: At this stage, the number of edges of G^3 is equal to 10.

$$D = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & \infty & \infty \\ 1 & 0 & 1 & 2 & 3 & \infty & \infty \\ 2 & 1 & 0 & 1 & 2 & \infty & \infty \\ 3 & 2 & 1 & 0 & 1 & \infty & \infty \\ 4 & 3 & 2 & 1 & 0 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & 0 & 1 \\ \infty & \infty & \infty & \infty & \infty & 1 & 0 \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

$$|E_3| = 10, \quad x_1 = [1, 5, 6], \quad x_2 = [3, 7], \quad x_3 = 4, \quad x_4 = 2.$$

For 4-distance coloring, the GDCA algorithm receives A and $p = 4$. It generates the matrix D. Using the matrix D, it creates the adjacency matrix G^4 and puts it in A. Then the GCA algorithm receives A and x_1, x_2, x_3, x_4, and x_5 produces as follows: At this stage, the number of edges of G^4 is equal to 11.

$$D = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & \infty & \infty \\ 1 & 0 & 1 & 2 & 3 & \infty & \infty \\ 2 & 1 & 0 & 1 & 2 & \infty & \infty \\ 3 & 2 & 1 & 0 & 1 & \infty & \infty \\ 4 & 3 & 2 & 1 & 0 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & 0 & 1 \\ \infty & \infty & \infty & \infty & \infty & 1 & 0 \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

$$|E_4| = 11, \quad x_1 = [3, 6], \quad x_2 = [1, 7], \quad x_3 = 2, \quad x_4 = 4, \quad x_5 = 5.$$

3 Numerical Results

Here, we present some numerical results obtained by applying MATLAB 80 9.3. All experiments were run on a PC with CPU Intel Core (TM) i7-7700K CPU at 4.20GHz, 32G bytes of SDRAM memory, and Windows 10 operating system. Here, we show the numerical results of the $GDCA$ algorithm, which has been tested on some benchmark graphs located in the $DIAMCS$ library, in Table 1. In Table 1, the first column entitled Graph shows the name of benchmarks graph; the column entitled V shows the number of vertices; the column entitled E shows the number of edges; the column entitled Den displays the density of edges obtained from the relation $Den = 2E/v(v-1)$; and the $best$ solution or $\chi(G)$ is for 1-distance coloring, the chromatic number or the best number ever known. In the rest of the columns, the results of the algorithm in terms of calculation time T and coloring number R for G^p graphs, where $p = \{1, 2, 3, 4, 5, 6, 7\}$, and also the number of edges of G^p, E, is given. If the number of edges of G^p and G^{p-1} are equal, then the graph is saturated. Experiments have been performed on a 12-core system and MATLAB software. The GCA algorithm simply colors the graph G. Since $G = G^1$, then the coloring of p-distance can be obtained directly with the GCA algorithm. As a result, it requires less calculation time for coloring. The calculation time T for coloring of a p-distance after obtaining the adjacency matrix of the graph G^p is calculated from step 3 onwards in Table 1. With the increase of p, the density of edges of the graph G^p increases strongly and the coloring number also increases. Also, the number of iterations of the GCA coloring algorithm increases as the coloring number. As a result, the calculation time is also greatly increased.

Table 1: Results R and calculation times (in the seconds) T.

Graph	V	E	Den.	Best/$\chi(G)$	G^1		G^2		G^3		G^4		G^5		G^6		G^7	
					R,E	***T***	***R,E***	***T***	***R,E***	***T***	***R,E***	***T***	***R,E***	***T***	***R,E***	***T***	***R,E***	***T***
DSJC125-1	125	736	0.09	5	7-736	0.0293	42-5508	0.2543	123-7748	0.5634	125-7750	0.5691	-	-	-	-	-	-
inithx-i3	621	13969	0.07	31	31-13969	0.6250	558-155960	25.0136	559-155961	25.2172	559-1555961	25.1700	-	-	-	-	-	-
le450-5c	450	9803	0.10	5	6-9803	0.2357	237-98014	4.2451	450-101025	10.7515	450-101025	10.7925	-	-	-	-	-	-
mycie17	191	2360	0.13	8	8-2360	0.0296	191-18145	1.2253	191-18145	1.2252	-	-	-	-	-	-	-	-
queen10-10	100	1470	0.59	11	13-1470	0.0408	100-4950	0.4023	100-4950	0.4002	-	-	-	-	-	-	-	-
DSJR500-1	500	3555	0.03	12	14-3555	0.4462	31-10177	0.9224	55-19785	1.3808	86-31709	1.9638	126-44851	2.6388	173-58368	3.4875	221-71793	4.4179

To the best of our knowledge, no existing algorithm exists for determining the distance coloring of graphs. In this study, we present an algorithm specifically designed for this purpose. As mentioned previously, our algorithm is versatile and can utilize different types of graph coloring algorithms. Therefore , we compare the graph coloring algorithm (GCA) utilized in Graph Distance Coloring Algorithm $(GDCA)$ with the best existing graph coloring algorithms. In [3], the FF, LDO, WP, IDO, $DSATUR$ algorithms, and the RLF were tested on benchmark graphs provided by $DIMACS$ [7]. The GCA algorithm was also tested on the same benchmark graphs, and the results were included in the last two columns of the tables in [3]. The benchmark graphs, used for testing the algorithms include $Mycielski$, SGB, $david$, $jean$, $anna$, $homer$, $huck$, $miles$, and $game$. Additionally, the number of vertices V, the number of edges E shows the number of edges, and the density of edges Den were recorded. The density is calculated using the formula $Den = 2E/v(v-1)$ and displays either the chromatic number $\chi(G)$ or the best number known $Best$. The number of colors R obtained by the algorithms, and the calculation time in seconds T are also provided. Table 2 presents one of the comparison tables for graph coloring.

Table 2: The results and computation times for Register Allocation graphs.

Graph	V	E	Den.	Best/$\chi(G)$	RLF		DSATUR		WP		LDO		IDO		FF		GCA	
					R	***T***	***R***	***T***	***R***	***T***	***R***	***T***	***R***	***T***	***R***	***T***	***R***	***T***
fpso12-i1	496	11654	0.09	65	65	0.9869	65	3.1791	65	0.0044	65	0.0646	65	1.8096	65	0.0552	65	0.3890
mulsol-i1	197	3925	0.20	49	49	0.1299	49	0.6347	49	0.0021	49	0.0153	49	0.2924	49	0.0137	49	0.1182
mulsol-i2	188	3885	0,22	31	31	0.1171	31	0.6423	31	0.0015	31	0.0145	31	0.2899	31	0.0133	31	0.0773
mulsol-i3	184	3916	0.23	31	31	0.1164	31	0.6189	31	0.0015	31	0.0143	31	0.2805	31	0.0134	31	0.0772
mulsol-i4	185	3946	0.23	31	31	0.1243	31	0.6328	31	0.0015	31	0.0145	31	0.2994	31	0.0130	31	0.0777
mulsol-i5	186	3973	0.23	31	31	0.1253	31	0.6286	31	0.0015	31	0.0145	31	0.2900	31	0.0128	31	0.0761
inithx-i1	864	18707	0.05	54	54	2.7427	54	6.7614	54	0.0066	54	0.1337	54	4.2802	54	0.1266	54	1.5789
inithx-i2	645	13979	0.07	31	31	1.4014	31	4.2319	31	0.0037	31	0.0839	31	2.5214	31	0.0800	31	0.6985
inithx-i3	621	13969	0.07	31	31	1.3034	31	4.1724	31	0.0035	31	0.0819	31	2.5577	31	0.0780	31	0.6160
zeroin-i1	211	4100	0.18	49	49	0.1427	49	0.6636	49	0.0020	49	0.0157	49	0.3188	49	0.0139	49	0.1197
zeroin-i2	211	3541	0.16	30	30	0.1062	30	0.5390	30	0.0015	30	0.0136	30	0.2504	30	0.0124	30	0.0753
zeroin-i3	206	3540	0.17	30	30	0.1150	30	0.5439	30	0.0014	30	0.0134	30	0.2530	30	0.0123	30	0.0681

4 Applications

A molecular graph serves as a representation of a chemical compound's structural formula based on graph theory. Specifically, a molecular graph is a labeled graph where the vertices represent atoms of the compound and edges correspond to chemical bonds. Therefore, molecular graphs can be described as graphs with a maximum vertex degree of 4. The chromatic number has been utilized in [4] to classify certain molecules (molecular graphs). In this study, we applied our algorithm to compute the chromatic number of two specific molecular graphs, namely $F_{5,12}$ and the truncated cube.

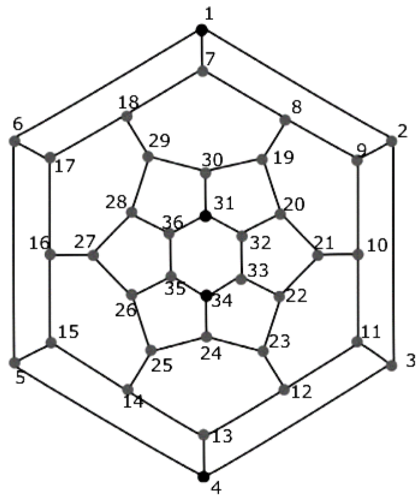

Figure 2: $G = (V, E)$ with $|V| = 36$ and $|E| = 54$.

$|E_1| = 54,$
$x_1 = [1, 3, 5, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30],$
$x_2 = [2, 4, 6, 7, 11, 15, 19, 21, 23, 25, 27, 29, 31, 33, 35],$
$x_3 = [9, 13, 17, 32, 34, 36].$

$|E_2| = 162,$
$x_1 = [1, 4, 10, 16, 19, 25, 33, 36],$
$x_2 = [2, 5, 12, 18, 21, 27, 31, 34],$
$x_3 = [3, 6, 8, 14, 23, 29, 32, 35],$
$x_4 = [7, 11, 15, 20, 24, 28],$
$x_5 = [9, 13, 17, 22, 26, 30].$

$|E_1| = 36,$
$x_1 = [1, 3, 5, 7, 10, 12, 14, 21],$
$x_2 = [2, 4, 6, 8, 9, 11, 13, 15],$
$x_3 = [16, 17, 18, 19, 20],$
$x_4 = [22, 23, 24].$

$|E_2| = 84,$
$x_1 = [1, 4, 13, 16, 20, 22],$
$x_2 = [2, 7, 11, 14, 19, 21],$
$x_3 = [3, 6, 10, 15, 17, 23],$
$x_4 = [5, 8, 9, 12, 18, 24].$

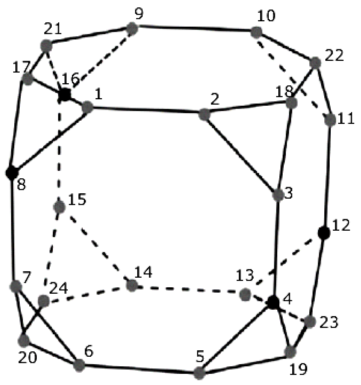

Figure 3: $G = (V, E)$ with $|V| = 24$ and $|E| = 36$.

Declarations

Availability of supporting data
All data generated or analyzed during this study are included in this published paper.

Funding
No funds, grants, or other support were received for conducting this study.

Competing interests
The authors have no competing interests to declare that are relevant to the content of this paper.

Authors' contributions
The main manuscript text is written collectively by the authors.

References

[1] Anitha, K., Selvam, B., Thirusangu, K. (2016). "Distance-2 chromatic number for the duplicate graph of the Mycielskian graphs", International Journal of Mathematics of Trends and Technology, 37, 67-72.

[2] Antonucci, S. (1978). "Generalizzazioni del concetto di cromatismo d'un grafo", Bollettino dell'Unione Matematica Italiana, 15-B, 20-31.

[3] Aslan, M., Baykan, N.A. (2016). "A performance comparison of graph coloring algorithms", International Journal of Intelligent Systems and Applications in Engineering, 4, 1-7.

[4] Bakaein, S., Tavakoli, M., Ashrafi, A.R., Ori, O. (2018). "Coloring of fullerenes", Fullerenes, Nanotubes and Carbon Nanostructures, 26, 705-708.

[5] Benmedjdoub, B., Bouchemakh, I. (2019). “2-distance colorings of integer distance graphs”, Discussiones Mathematicae Graph Theory, 39, 589-603.

[6] Bousquet, N., Esperet, L., Harutyunyan, A. (2019). “Exact distance colouring in trees”, Combinatorics, Probability and Computing, 28, 177-186.

[7] DIMACS graph coloring instances, (2016). “Instances homepage on CMU”, [online]. Available: http:mat.gsia.cmu.edu/COLOR/instances.html.

[8] Fertin, G., Godard, E., Raspaud, A. (2003). “Acyclic and k-distance coloring of the grid”, Information Processing Letters, 87(1), 51-58.

[9] Floyd, R.W. (1962). “Algorithm 97: Shortest path”, Communications of the ACM, 5, 345.

[10] Garey, M.R., Johnson, D.S. (1979). “Computers and intractability: A guide to the theory of NP-completeness”, W.H. Freeman and Company, New York, Nantes, France.

[11] Ghazi, Gh., Rahbarnia, F., Tavakoli, M. (2020). “2-Distance chromatic number of some graph products”, Discrete Mathematics, Algorithms and Applications, 12, 2050021.

[12] Gionfriddo, M. (1978). “Sulle colorazioni Ls d’un grafo finito”, Capsula. Un. Stuoia. Ital. (5). v15-A, 444-454.

[13] Gionfriddo, M., Milici, S. (1988). “On the parameter $v2(h) \leq 6h$ for L2-coloured graphs”, Discrete Mathematics, 68, 107-110.

[14] Jacko, P., Jendrol, S. (2005). “Distance coloring of the hexagonal lattice”, Discussiones Mathematicae Graph Theory, 25(1-2), 151-166.

[15] Kramer, F. (1972). “Brève communication. Sur le nombre chromatique des graphes”, Revue française d’automatique informatique recherche opérationnelle, Mathématique, 6(R1), 67-70.

[16] Kramer, F., Kramer, H. (1969). “Un probleme de coloration des sommets d’un graphe”, CR Académie des sciences Paris A, 268(7), 46-48.

[17] Kramer, F., Kramer, H. (2008). “A survey on the distance-colouring of graphs”, Discrete Mathematics, 308(2-3), 422-426.

[18] Liu, D.D.F. (2008). “From rainbow to the lonely runner: A survey on coloring parameters of distance graphs”, Taiwanese Journal of Mathematics, 12, 851-871.

[19] Molloy, M., Salavatipour, M.R. (2002). “Frequency channel assignment on planar networks”, In Algorithms—ESA 2002: 10th Annual European Symposium Rome, Italy, September 17–21, 2002 Proceedings 10 (pp. 736-747). Springer Berlin Heidelberg.

[20] Mosawi, H., Tavakoli, M., Ghorbani-Moghadam, Kh. (2023). “Solving graph coloring problem using adjacency matrix algorithm”, Accepted by Mathematics Interdisciplinary Research.

[21] S. Sohrabi Hesan, F. Rahbarnia, M. Tavakoli, “The smallest number of colors needed for a coloring of the square of the Cartesian product of certain graphs”, Control and Optimization in Applied Mathematics, 8 (2023) 83-93.

[22] Speranza, F. (1975). “Colorazioni di specie superiore d’un grafo”, Bollettino dell’Unione Matematica Italiana (4) 12, 53–62.