Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 8531659, 13 pages
https://doi.org/10.1155/2022/8531659

Research Article

Application of Multiobjective Optimization Integrating Numerical and Scientific Computing in Graph Theory Coloring Algorithm

Shuxia Li [1,2]

[1] *College of Mathematical Sciences, Inner Mongolia Normal University, Hohhot 010020, Inner Mongolia, China*
[2] *Inner Mongolia Center for Applied Mathematics, Hohhot 010020, Inner Mongolia, China*

Correspondence should be addressed to Shuxia Li; 20051043@imnu.edu.cn

Received 28 April 2022; Accepted 10 June 2022; Published 9 July 2022

Academic Editor: Xiantao Jiang

In the current era of increasingly complex social life, as people's demand quality is getting higher and higher, the solution of the problem often has multiple indicators to achieve the optimum. This forces the graph coloring problem (GCP) to become complicated, and it is difficult to directly obtain the optimal solution, which brings new challenges to the solution of the problem. In response to this problem, the GCP field is of great research significance. With the in-depth study of GCP, the research on multiobjective optimization (MOO) in graph coloring algorithm (GCA) is gradually carried out. Its performance advantage is of great significance for solving multicondition constraint problems. This paper aims to study the application of Genetic Algorithm (GA) in GCA. Through the analysis and research of GA, and the fusion of numerical analysis and scientific computing, it can be applied to the construction of the neighbor distinguishable uniform V-full coloring algorithm (AVDEVTCA) to solve the AVDEVTC problem. This paper describes the basic theory of MOO and graph coloring. It conducts an experimental analysis of the algorithm performance and uses the relevant theoretical formulas to explain it. The results show that the algorithm takes 5754.142 S seconds to test 21325415 images and can color a large number of results that cannot be done manually in less time. It has greatly improved in terms of time and manpower saving and greatly improved practicability and work efficiency.

1. Introduction

The modernization process is constantly evolving, and various fields are constantly developing at the same time, and many numerical calculation problems are extended. Ordinary human computing has been unable to solve huge data or more accurate data. Numerical analysis and scientific computing are the use of computers to perform numerical calculations. Numerical analysis and scientific computing have successfully solved a series of difficult problems such as the optimal solution of parameters in all walks of life, with high accuracy. The scientific computing power of computers is still limited. For example, in numerical weather forecasting, only medium- and short-term forecasts can be made. In the aerodynamic design of the aircraft, it can only be carried out in parts, and in the oil exploration, only rough mathematical models can be processed. More powerful computers are required for long-term numerical weather forecasting, overall aircraft aerodynamic design, and processing of more accurate mathematical models in oil exploration.

In today's increasingly complex situation, how to quickly and efficiently obtain the optimal solution for problems with multiple constraints has far-reaching significance for the development of all walks of life. The concept of MOO is that when multiple objectives need to be achieved in a certain situation, due to the inherent conflict between objectives, the optimization of one objective is at the expense of the deterioration of other objectives, so it is difficult to obtain a unique optimal solution. Instead, there is coordination and compromise among them to make the overall goal as optimal as possible. The MOO idea has better effect on the multiconstraint problem to be solved and has less restrictions, so its application range is very wide. In recent years, some scholars have used scientific computing to solve graph coloring problems, but the application and research of MOO algorithms in this area are relatively few. Therefore, it is of great significance to apply genetic algorithm to the research of solving the problem of average total color of graphs.

2. Related Work

With the progress of society, more and more people have studied numerical analysis and scientific computing. Danaila et al. proposed to use numerical methods to solve problems in various fields of application and provided 12 computational projects [1]. This point of view has aroused the continuous attention of later scholars on the research of this problem, but the application in other fields needs to be expanded. Therefore, And Elman and Furnival applied multigrid to solve the stochastic steady-state diffusion problem [2]. Although his research has proved theoretically and experimentally that the convergence rate is not related to spatial discretization and random discretization. But the research to solve the disk simulation problem has not been carried out. Based on this, Wilber et al. described an algorithm applied to define a smooth function on the unit disk for numerical computation to solve-related problems [3]. However, although this study fills the gap of numerical analysis applied to the disk simulation problem, it lacks a strong practical example to prove it. Dlz et al. utilized boundary integral equations to efficiently solve partial differential equations of strong elliptic operators with constant coefficients and stochastic Dirichlet data. But the values it provides are not accurate enough [4]. Tonicello et al. explored some spectral analysis and typical fluid mechanics problems. Although there is a certain progress significance, but with the emergence of data, mining and data popular learning problems can no longer meet the needs [5]. Based on this, Druskin et al. proposed the application of graph Laplacian dimensionality reduction based on data spectral clustering [6]. While this is somewhat helpful for solving related problems, the computational cost is too high. Karaa et al. proposed the use of FEM to derive the solution of the fractional time diffusion equation over a bounded convex domain. Although there is some example data analysis, its accuracy is not enough [7].

After a period of research, it is found that MOO provides new ideas for solving new problems in numerical computing and scientific computing applications. Therefore, some scholars turn their attention to the research of MOO. Rashidi and Khorshidi developed a MOO method based on differential evolution algorithm and local unimodal sampling technique. He applied it to a biomass gasification system to calculate the optimal values of system parameters for multiple generations [8]. This is an important attempt on the application of MOO to problem solving. But it only exists as an auxiliary function and is not the main research object. Therefore, Zille et al. proposed a new method called a weighted optimization framework to solve MOO problems with a large number of decision variables [9]. This study takes MOO method as the main research object and fills the research gap. However, it was found in the study that this method relies on the grouping mechanism to be solved. Do et al. successfully solved the MOO problem of turning process through Taguchi combination method and MOORA technology. However, the authenticity of the experimental data needs to be verified [10]. Onler et al. proposed a method to determine the ideal process parameters for Co-Cr-Mo alloy binder jetting [11]. This method solves the problem of high-speed and low-cost manufacturing of high-volume defect-free products. But the range of conditions of application is very limited. Kazi et al. developed a hybrid powder blending EDM technology [12]. Although this technology improves the problem of increasing the corresponding amount of powder in mixed powder processing, it does not solve the problem in essence. Babaelahi et al. made parabolic fin (convex) heat sinks through MOO to maximize thermal efficiency and minimize entropy generation [13]. It applies multiproblem optimization to the field of LED lamps and broadens the application scope of this technology, but the control accuracy is not enough. Yadav et al. used a multiobjective genetic algorithm to optimize the rotational speed of the equipment. He determined optimal values of design parameters related to active magnetic bearing (AMB) geometry and electromagnetic actuators [14]. However, the experimental process is not rigorous enough, ignoring the influence of different environments. The above research attempts to apply MOO to the solution of numerical problems in various fields. However, the application research of fusion MOO in graph coloring algorithm needs to be supplemented.

In order to solve the application research problem of fusion MOO in graph coloring algorithm, this paper uses MOO to analyze the graph coloring algorithm. The algorithm performance test is simulated to achieve the highest accuracy and the lowest time loss. The innovation of this paper is (1) the theoretical knowledge of MOO and graph coloring algorithm are introduced. It also uses MOO and graph coloring algorithm to analyze how genetic algorithms and graph theory shading algorithms play a role in the application research of MOO integrating numerical analysis and scientific computing in graph coloring algorithm. (2) The performance of the proposed algorithm is described below. Experimental results show that the algorithm has excellent performance and practicability and significantly improves the coloring efficiency.

2.1. Methods. When solving graph coloring problems using common intelligent algorithms, there is a limitation that is limited to solving graph coloring problems with a single constraint, for example, ant colony algorithm, BP neural network algorithm, tree algorithm, etc. However, when faced with the problem of multiconstraint graph coloring, the effect of common algorithms is often unsatisfactory [15]. Through the investigation, it is found that there are very few introductions that combine scientific computing with graph coloring algorithms. So this paper proposes an algorithm to solve this multiconstraint problem. The algorithm designs a total objective function and four subobjective functions to accommodate the various constraints of AVDEVTC.

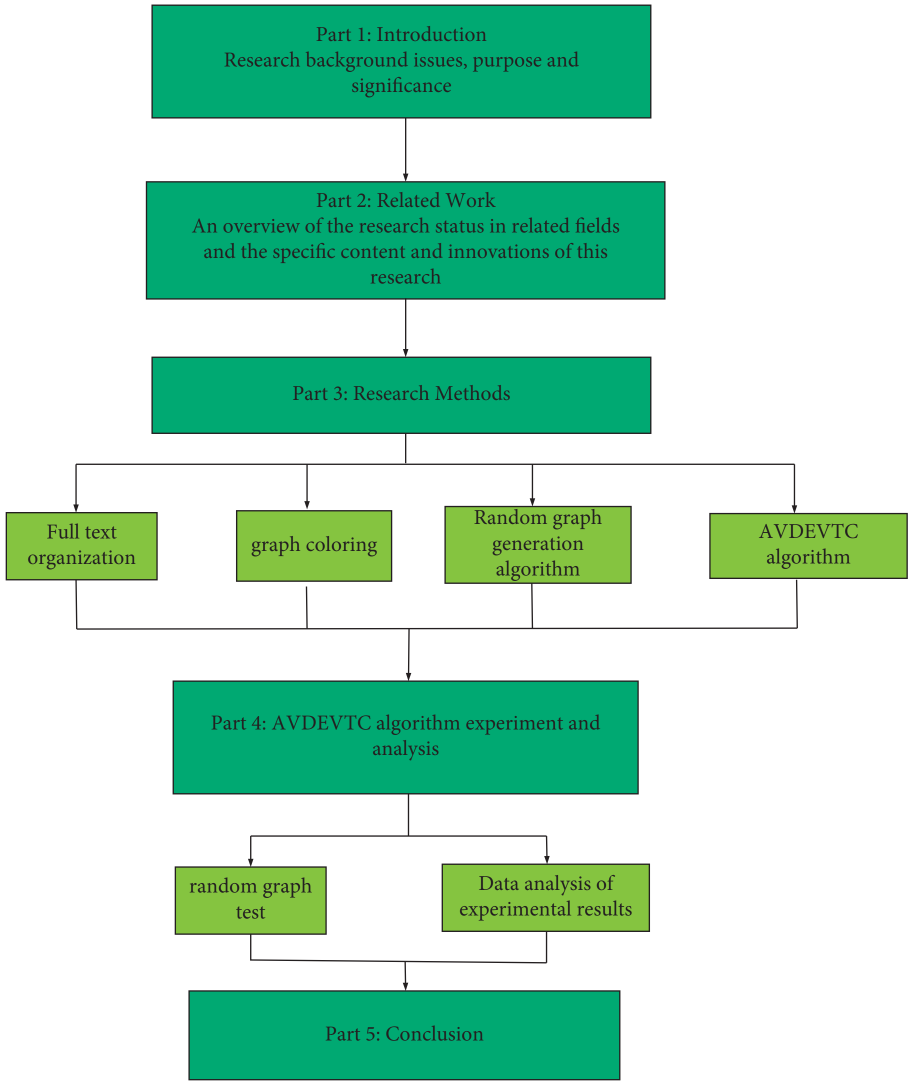


Figure 1: Full-text organizational chart.

Through the repeated exchange operation of the color set of each point on the coloring matrix, each subpurpose function is optimized, and then the overall objective function requirements are satisfied. Extensive experiments and analysis show that the algorithm obtains correct and less time-consuming results for AVDEVTCN and graph adjacency matrix. The organizational structure of this paper is shown in Figure 1:

As can be seen from Figure 1, the full text of this study consists of five parts. The first part mainly introduces the research background of numerical computation and MOO problems. It leads to the fact that conventional intelligent algorithms cannot solve multiconstraint problems well to illustrate the purpose and significance of this research. The second part makes a general analysis of the research status in the fields of numerical analysis, scientific computing and graph coloring, and explains the content and innovation of this paper. The third part describes the organization structure and method of this research and shows it through the structure diagram. It also introduces the related methods of graph coloring and MOO. It also provides an algorithmic description of the random graph generation algorithm and the proposed new algorithm. The fourth part obtains the result by simulating the random graph and testing a large number of graphs and draws the conclusion that the algorithm can improve the work efficiency after analyzing the result data. The fifth part concludes and reflects.

2.2. Figure Staining. This paper mainly focuses on the algorithm research for the solution of GCP. Graph coloring problem (GCP), also known as coloring problem, is one of the most famous NP-complete problems. Mathematical definition: given an undirected graph $G = (V, E)$, where V is the set of vertices and E is the set of edges. The graph coloring problem is to divide V into K color groups, each of which forms an independent set, that is, there are no adjacent vertices in it. Its optimized version is to hope to obtain the smallest K value. Therefore, the following describes the definition and introduction of the principle of graph coloring to prepare for the subsequent optimization algorithm.

2.2.1. Definition of Strongly Distinguishable Coloring. T (Q, W) is a simple connected graph with order greater than or equal to 2. The mapping of E (Q)∪R (W) to positive integer $Y = \{1, 2, \ldots, k\}$ is f, and for $\forall ui, uo \in W(T), i \neq o, f(ui) \neq f(uo)$, for $\forall ui \in W(T), u \neq i, f(u) \neq f(i), f(u) \neq f(ui), f(i) \equiv f(ui)$, for $\forall ui \in W(T), u \neq i, Y(u) \neq Y(i)$, it is expressed by formula (1):

$$f = k - ASDTC \cdot \text{of}(T). \tag{1}$$

Then the strongly distinguishable panchromatic number of adjacent points is defined as formula (2):

$$P_{ast}(T) = \min\{k | k - ASDTC \cdot \text{of}(T)\}. \tag{2}$$

Among them, P_{ast} (T) is the strongly discriminative panchromatic number of adjacent points in the graph T and $Y(u) = \{f(u)\} \cup \{f(i)\} | ui \in W(T)\}$.

If there is a connected graph whose order T is not less than 3, then there is formula (3):

$$P_{ast}(T) \geq \Delta(T) + 1. \tag{3}$$

When the graph T has more than two adjacent maximum degree vertices, it is shown in formula (4):

$$P_{ast}(T) \geq \Delta(T) + 2. \tag{4}$$

Suppose A_n is a complete graph of order n with $n \geq 3$, then there is formula (5):

$$P_{ast}(A_n) \geq n + [\log_2 n]. \tag{5}$$

Among them, $[\log_2 n]$ is the smallest integer not less than $\log_2 n$ in formula (5).

Then according to the above principles, it is guessed that there is a simple connected graph with order T of at least 3, and there is formula (6):

$$P_{ast}(T) \leq n + [\log_2 n] + 1. \tag{6}$$

For the plane graph T of order not less than 3, there is formula (7):

$$P_{ast}(T) \leq \Delta(T) + 3. \tag{7}$$

The combination degree of graph T is shown in formula (8):

$$\alpha_i(T) = \min\left\{\mathfrak{I} \middle| \left(\frac{\mathfrak{I}}{i+1}\right) \geq n_i, \beta \leq i \leq \Delta\right\}. \tag{8}$$

Among them, i is the number of points; n_{i} is the moderate degree of E (T); $\alpha_i(T)$ is the combination degree of graph T; and the P_{ast} of part of the graph T is shown in Table 1:

There is a simple connected graph whose order T is at least 3, then there is formula (9):

$$P_{vst}(T) \geq \Delta(T) + 1. \tag{9}$$

When the graph T has a maximum degree vertex of ≥2, then there is formula (10):

$$P_{vst}(T) \geq \Delta(T) + 2. \tag{10}$$

Assuming that there is a complete graph F when $n \geq 3$, then there is formula (11):

Table 1: P_{ast} of part of graph T.

Figure T	To meet the conditions	P_{ast}
Z_n	n is odd and $n \geq 3$	4
Z_n	n is even and $n \geq 3$	5
X_n	$n \equiv 0$ (mod2), $n \neq 4$, 10, $n \geq 3$	4
X_n	$n \equiv 1$ (mod2), $n = 4$, 10, $n \geq 3$	5
C_n	$n \geq 3$	$n+1$
V_n	$n = 2$, 3	5
V_5	⊗	6
V_n	$n \geq 5$	$n+1$
B_n	$n = 2^k$, $k = 2$, 3, 4, 5	$n+k$
B_n	$n \geq 3$, and ≠6, 14	$n + [\log_2 n]$
N	$\Delta(N) \geq 4$ and there is no maximum degree point neighbor	$\Delta(N) + 1$
N	$\Delta(N) \geq 4$and there is maximum degree point neighbor	$\Delta(N) + 2$

$$P_{vst}(F_n) \geq n + [\log_2 n]. \tag{11}$$

In formula (11), $[\log_2 n]$ is the smallest integer of $\geq \log_2 n$.

There is a simple connected graph of order T (Q, W) not less than 2, f is the mapping from $Q(T) \cup WW(T)$ to positive integer $Y = \{1, 2, \ldots k\}$. And for $\forall ui, uo \in W(T), i \neq o, f(ui) \neq f(uo)$, $\forall ui \in W(T), u \neq i, f(u) \neq f(i), f(u) \neq f(ui), f(i) \equiv f(ui)$, and $\forall ui \in W(T), u \neq i, Y(u) \neq Y(i)$, the point-intensity distinguishable complete colors of the graph are shown in formula (12):

$$f = k - VSDTC \cdot \text{of}(T). \tag{12}$$

Then the strongly distinguishable panchromatic number of adjacent points is defined as formula (13):

$$P_{ast}(T) = \min\{k | k - VSDTC \cdot \text{of}(T)\}. \tag{13}$$

Among them, P_{ast} (T) is the k-point strongly distinguishing panchromatic number of graph T, and $Y(u) = \{f(u)\} \cup \{f(i)\} | ui \in W(T) \cup f(ui) | uo \in W(T)\}$.

Assuming that there is a complete bipartite graph F_{nm}, there is formula (14):

$$n + 2 \leq P_{vst}(F_{nm}) = n + 3. \tag{14}$$

Among them, $2 \leq m \leq n$.

Assuming that there is a complete bipartite graph F_{nm}, there is formula (15):

$$n + 3 \leq P_{vst}(F_{nm}) = n + 4. \tag{15}$$

Among them, $n \geq 4$.

According to the above derivation, for the complete graph F_n ($n \geq 3$), there is formula (16):

$$P_{vst}(K_n) = P_{ast}(K_n). \tag{16}$$

The P_{vst} of part of the graph T is shown in Table 2.

3. MOO and Genetic Algorithm That Integrates Numerical Analysis and Scientific Computing

This paper combines the MOO idea with the graph coloring algorithm and combines numerical analysis and scientific computing to propose a new algorithm.

Table 2: P_{vst} of partial graph T.

Figure T	To meet the conditions	P_{vst}
M_3	$\otimes$	4
M_n	$4 \le n \le 11$	5
N_n	$5 \le n \le 16$	5
N_n	$15 \le n \le 48$	6
B_n	$n \ge 5$	$n+1$
B_n	$n = 3, 4$	6
V_n	$n \ge 5$	$n+1$
P_n	$n = 2^k$, $k = 2, 3, 4, 5, 6$	$n+k$
P_n	$n \ge 3$, and $\ne 6$, 14	$n + [\log_2 n]$

3.1. MOO. Multifunctional optimization (decision-making) problems often do not have a single globally optimal solution, but there are multiple optimal solutions [16]. The elements of the optimal solution set for a multipurpose problem cannot be compared with all subobjectives. In general, this is called the Pareto optimal solution set, and the elements of the solution set are called nongood and bad optimal solutions. Assuming that for any two solutions $S1$ and $S2$ for all objectives, $S1$ is better than $S2$, then $S1$ is said to dominate $S2$. If $S1$ is not dominated by other solutions, then $S1$ is called a nondominated solution (undominated solution), also known as a Pareto solution.

Objective functions and constraints in multifunctional optimization problems are generally functions with independent variables [17]. Its optimization definition is shown in formula (17):

$$\begin{aligned} M&: y = g(x) = g_1(x), g_2(x), \ldots, g_z(x) \\ S&: d(x) = \left(d_1(x), d_2(x), \ldots, d_c(x)\right) \le 0 \\ &W x = \left(x_1, x_2, \ldots, x_n\right) \in x, y = \left(y_1, y_2, \ldots, y_n\right) \in Y. \end{aligned} \tag{17}$$

Among them, X is the independent variable space; Y is the target space; the independent variable vector is x; and the target vector is y, and the feasible solution set is determined by $d(x) \le 0$.

A feasible set is a set of independent variable vectors that satisfy the constraints as shown in formula (18):

$$X_g = \{x \in X | d(x) \le 0\}. \tag{18}$$

Among them, in the feasible region of the target space, Y_g is the phase of X_g Numerical analysis can be used to study its error bounds, iterative scheme convergence, convergence order and speed, and computational complexity.

Numerical analysis is the study of mathematical analysis algorithms (as opposed to general symbolic operations) that use numerical approximations (as opposed to discrete mathematics).

Scientific computing uses advanced computing capabilities to understand and solve complex problems. Practical aspects of obtaining an exact solution to a multiobjective optimization problem can be obtained through scientific computing.

3.2. Genetic Algorithm. Genetic algorithm is an efficient optimization method that can be used to solve and deal with complex optimization problems [18]. Genetic algorithm is a stochastic intelligent optimization algorithm based on natural selection and natural genetic mechanism. In real life, when using the genetic algorithm to solve the optimization problem, the chromosome size of the population is first determined according to the feasible region. The encoded fitness value is then calculated by constructing a fitness function. Then, according to different fitness values, the chromosomes are selected, crossed, and mutated to generate more populations that meet the conditions and search for the best solution in a wider range of feasible space. In this paper, the idea of genetic algorithm is used to improve the full coloring algorithm. The basic process of genetic algorithm is shown in Figure 2:

As shown in Figure 2, the optimization problem is solved by a genetic algorithm. First, the size of the chromosome is determined, and second, the fitness of the code is calculated. It again performs regular operations on chromosomes based on the fitness value. Finally, a population that satisfies the conditions is generated to complete the entire algorithm and obtain the optimal solution of the problem.

Genetic algorithm has the advantages of self-organization, self-adaptation, self-learning, and intrinsic parallelism, so it can search multiple regions at the same time. But at the same time, the genetic algorithm also has shortcomings, such as when the scale of the graph increases, the chromosome subspace increases sharply, and the design of the crossover operator and mutation operator is blind. This greatly affects the optimization efficiency of the graph coloring problem. Genetic algorithm has good global search ability, but still has some shortcomings in local search ability.

4. Algorithm for Generating Random Graphs

The two random graph generation algorithms provided basic research data to demonstrate the test of the result set, statistics and related additions, and conjectures of a series of coloring algorithms, laying the foundation for subsequent research on coloring algorithms [19].

4.1. ER Model. In 1959, Hungarian mathematicians proposed a random network model that randomly connected points and edges with the same probability to form a random network, namely the ER model. When a number of fixed points are given, it is assumed that each vertex is connected by edges, and a random graph can be obtained by randomly selecting edges from these edges. The number of random graphs that can be generated is shown in formula (19):

$$L^h_{g(g-1)/2}. \tag{19}$$

Among them, g is the number of vertices; $g(g-1)/2$ is the number of edges; and h is the number of edges randomly selected from them.

4.2. Binomial Model. The probability of its existence is shown in formula (20):

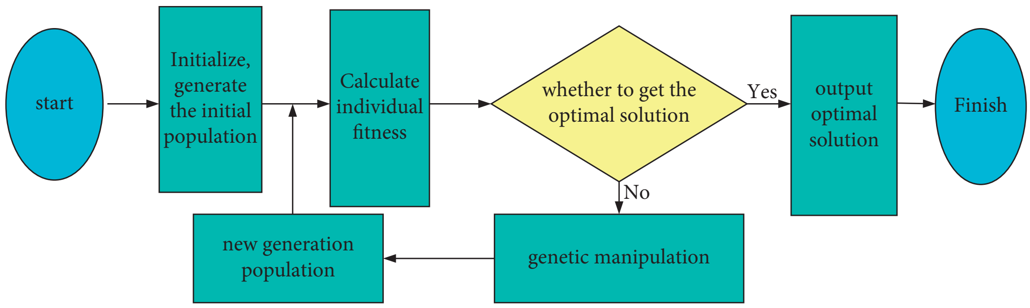


Figure 2: Basic flowchart of genetic algorithm.

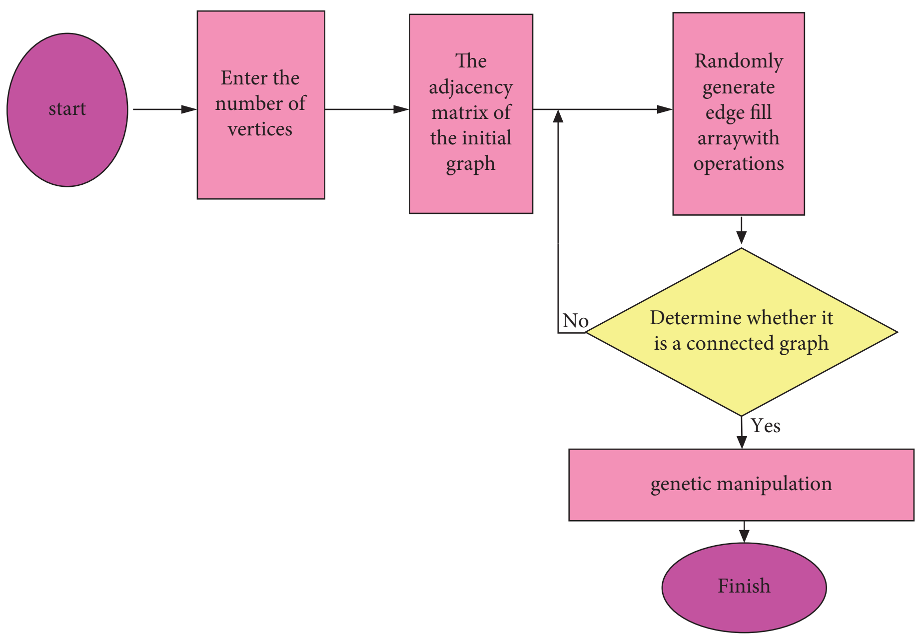


Figure 3: Flowchart of random graph algorithm.

$$G(i) = g^{B}(1-g)^{m\frac{(m-1)}{2-m}}. \tag{20}$$

Among them, m is the number of vertices; g is the connection probability; and B is the average value of all connections.

The random graph algorithm is designed, and the basic process of the random graph algorithm is obtained, as shown in Figure 3:

As can be seen from Figure 3, the random graph is generated by the random graph algorithm. First determine the number of vertices to get the random matrix of the adjacency graph, and then randomly generate the edge-filling array of the graph. It fills the edges according to the rules, 1 means there is an edge, and 0 means no edge. Next, judge whether it is a connected graph, if not, return to the previous step, if yes, output the adjacency matrix of the random graph, and the algorithm ends.

4.3. Spanning Tree Algorithm. In the mathematical field of graph theory, if a subgraph of a connected graph G is a tree containing all vertices of G, then the subgraph is called a spanning tree of G. A spanning tree is a minimal connected subgraph of a connected graph that contains all vertices in the graph. The spanning tree of a graph is not unique. By traversing from different vertices, different spanning trees can be obtained. This algorithm can realize all spanning trees within a limited number of points, which brings convenience to the problem of graph generation and graph coloring studied in this paper [20].

The weight $Q(S)$ of the tree is written as formula (21):

$$Q(S) = \sum_{(z,x)\in S\,D} \partial(z,x). \tag{21}$$

Among them, SD is the edge set of S; $Q(z,x)$ is the weight of the edge (z,x).

4.4. Solving Matrix Eigenvalue Algorithms. Although there are many ways to solve the eigenvalues of mathematical matrices, this paper uses the QR decomposition method to solve the corresponding eigenvalues and eigenvectors. The QR algorithm is an algorithm that uses a recursive method to solve the eigenvalues and eigendirections of a matrix. When the number of recursive rounds is appropriate, the QR algorithm will obtain an upper triangular matrix, and then

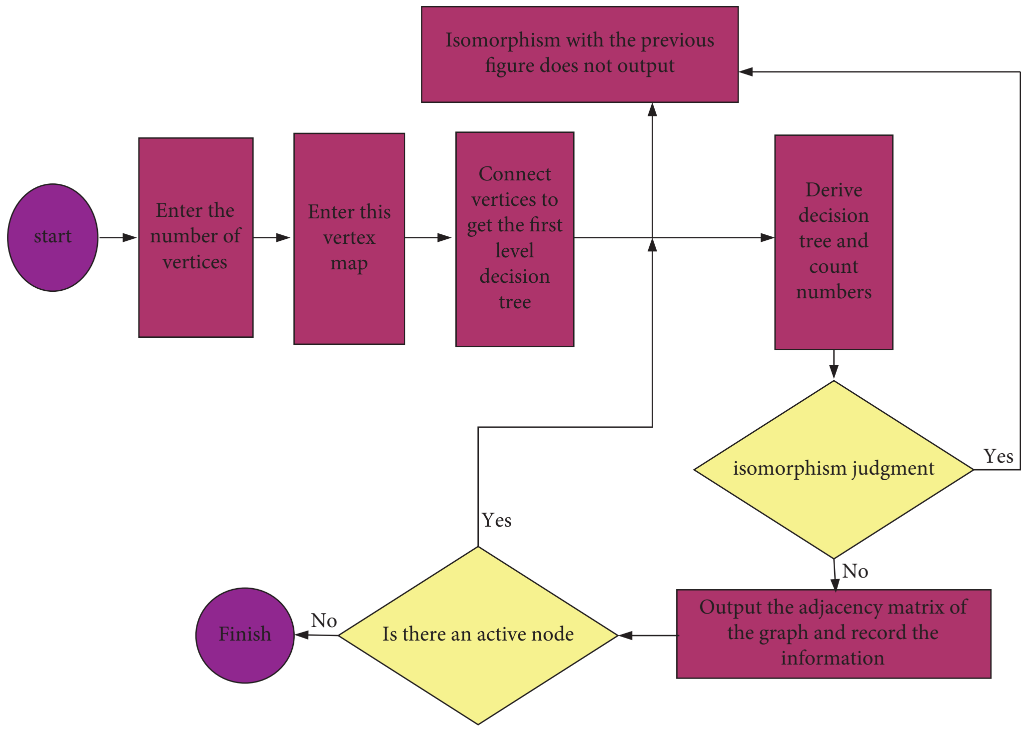


FIGURE 4: Flowchart of spanning tree algorithm.

obtain the eigenvalues. The basic idea is to generate the same repeating sequence as matrix J by orthogonal decomposition of the matrix, as in formula (22):

$$\begin{cases} J_x = Y_x P_x \\ J_{x+1} = P_x Q_x \end{cases} \quad (x = 1, 2, \ldots). \tag{22}$$

Among them, J is the matrix to be solved.

The basic flow of the spanning tree algorithm is shown in Figure 4:

As can be seen from Figure 4, all spanning trees for a given point are output using the spanning tree algorithm. First determine the number of vertices, then input the star map of the vertex to obtain the first layer of the decision tree, and then continue to generate the decision tree according to the rules, count the active node set and number, and then judge whether it is an isomorphic graph. If yes, return to the previous step without output, otherwise, output the adjacency matrix of the graph and record it. Then judge whether there is an active node, if yes, then return to continue to derive the decision tree, otherwise the algorithm ends.

4.5. Neighbor Distinguishable Uniform V-Full Coloring Algorithm of Graphs

4.5.1. Algorithm Description. First randomly color the vertex and then the same color on any two points. After the random vertices are colored, the related vertices are colored with the remaining colors. After half of the objective function, if it is not correct, continue to adjust until it fully meets the requirements after adjustment.

The flow of the MOO algorithm is shown in Figure 5.

It requires that the difference in the number of times every two colors is used is not more than 1. Adjacent vertices have different colors, adjacent edges have different colors, and vertices and associated edges have different colors. Then the multiobjective function satisfying these conditions is shown in formula (23):

$$F(G) = f_1(g_1) + f(g_2) + f(g_3) + f(g_4). \tag{23}$$

Among them, the decision vector is $G = (G_1, G_2, G_3, G_4)$.

4.5.2. Build the Objective Function. The edge coloring objective function is constructed as shown in formula (24):

$$\begin{aligned} \alpha_1(b_1, b_2) &= \begin{cases} 1, f(b_1) = f(b_2) \\ 0, \text{other} \end{cases} \\ f_1(g_1) &= \sum_{b_1 b_2 \in H(K)} \alpha_1(b_1, b_2). \end{aligned} \tag{24}$$

Among them, b_1 is any two adjacent edges and $\in H(G)$;

The point-edge objective function is constructed as shown in formula (25):

$$\begin{aligned} \alpha_2(m, i) &= \begin{cases} 1, f(mi) = f(m) \text{or} f(mi) = f(i) \\ 0, \text{other} \end{cases} \\ f_2(g) &= \sum_{mi \in} \alpha_2(m, i). \end{aligned} \tag{25}$$

Among them, mi is any two edges and $\in H(G)$. The number of edges for vertex color conflict is $f_2(g)$.

The objective function of constructing the color set is shown in formula (26):

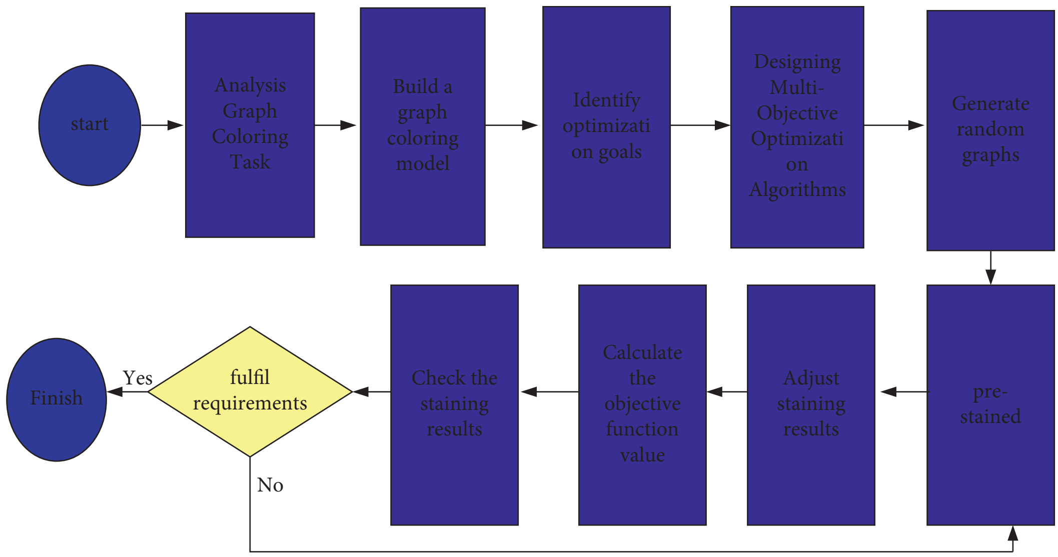


Figure 5: Flowchart of MOO algorithm.

$$f_3(g_3) = \sum_{mi\in H(G)} \alpha_3(m,i). \tag{26}$$

Among them, m, i are two adjacent points, $\in V(G)$. $\alpha_3(m,i) = \begin{cases} 1, & mi \in H(G) \text{ and } P(m) = P(i), \\ 0, & \text{other} \end{cases}$; $g3 = (P1, P2, \ldots, Pn)$. The number of color set collisions for neighbor pairs is $f_3(g_3)$.

The limitation of the unified objective function is that the difference in the number of uses of any two colors is less than 1. Its function is shown in formula (27):

$$f_4(g) = \sum_{mi\in H(G)} y_4(e,r). \tag{27}$$

The number that does not satisfy that the difference in the number of times of use of each two colors is not greater than 1 is $f_4(g)$.

According to the above subobjective functions, the total objective function is obtained as shown in formula (28):

$$Z = \min F(G). \tag{28}$$

Among them, $F(G) = f_1(g_1) + f_2(g_2) + f_3(g_3) + f_4(g_4)$. When $F(G) = 0$, the staining was successful.

5. Experimental Simulation and Analysis

In the random graph test, the algorithm is tested by choosing a random graph with 8 vertices. Initialize the random graph: first, an adjacency matrix of a random graph with 8 vertices is generated by using the random graph generation algorithm. Then count the degrees of each vertex and determine the initial chromatic number. The initial graph adjacency matrix is shown in Table 3:

It can be seen from Table 3 that the maximum degree of the graph is 7, and the chromatic number required for full dyeing can be obtained as $7 + 1 = 8$. The number of graphs obtained for each of the other vertices is 3 for 3 degrees, 2 for 4 degrees, and 2 for 5 degrees. Among them, $Q(V_i)$ is the degree of each vertex, and V_i represents each vertex of the graph.

Table 3: Adjacency matrix of initial graph.

V_i	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	$Q(v_n)$
V_1	1	0	1	1	0	1	0	0	4
V_2	1	1	1	0	1	1	1	1	7
V_3	0	0	1	0	1	1	0	0	3
V_4	1	0	1	1	0	0	1	1	5
V_5	0	0	1	0	1	1	1	0	4
V_6	0	0	1	1	0	1	0	0	3
V_7	1	1	0	0	1	0	1	1	5
V_8	0	0	1	0	1	0	1	0	3

Table 4: Initial staining matrix.

V_i	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	$Q(v_n)$
V_1	5	0	3	1	0	4	0	8	4
V_2	4	7	6	2	3	5	8	1	7
V_3	0	5	3	0	1	7	0	0	3
V_4	6	4	8	7	0	0	1	0	5
V_5	1	0	6	0	3	7	5	0	4
V_6	0	0	4	8	0	3	0	6	3
V_7	5	7	0	8	4	0	6	3	5
V_8	0	5	4	0	3	0	8	0	3

Table 5: Conflicting set lookup.

V_n	$Q(v_n)$	$S(v_n)$	$Q(v_n) + S(v_n)$	Is it normal?
V_1	4	3	7	Yes
V_2	7	0	7	Yes
V_3	3	5	8	No
V_4	5	2	7	Yes
V_5	4	4	8	No
V_6	3	6	9	No
V_7	5	2	7	Yes
V_8	3	4	7	Yes

Table 6: Nonuniform dyeing results.

V_i	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	$Q(v_n)$
V_1	5	3	0	1	0	6	0	8	4
V_2	7	4	8	2	1	5	6	3	7
V_3	8	1	3	0	0	7	0	0	3
V_4	5	2	8	7	0	0	1	3	5
V_5	7	0	3	0	8	5	1	0	4
V_6	5	0	8	0	0	4	0	2	3
V_7	5	7	0	8	4	0	6	3	5
V_8	0	5	4	0	3	0	8	0	3

Table 7: Final uniform dyeing results.

V_i	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	$Q(v_n)$
V_1	5	7	0	3	0	2	0	1	4
V_2	7	4	8	2	1	5	6	3	7
V_3	4	0	5	3	0	6	0	0	3
V_4	5	2	8	7	0	4	0	3	5
V_5	7	0	3	0	6	5	1	0	4
V_6	7	0	8	0	0	4	0	2	3
V_7	5	7	0	8	1	0	6	2	5
V_8	0	5	4	0	3	0	6	0	3

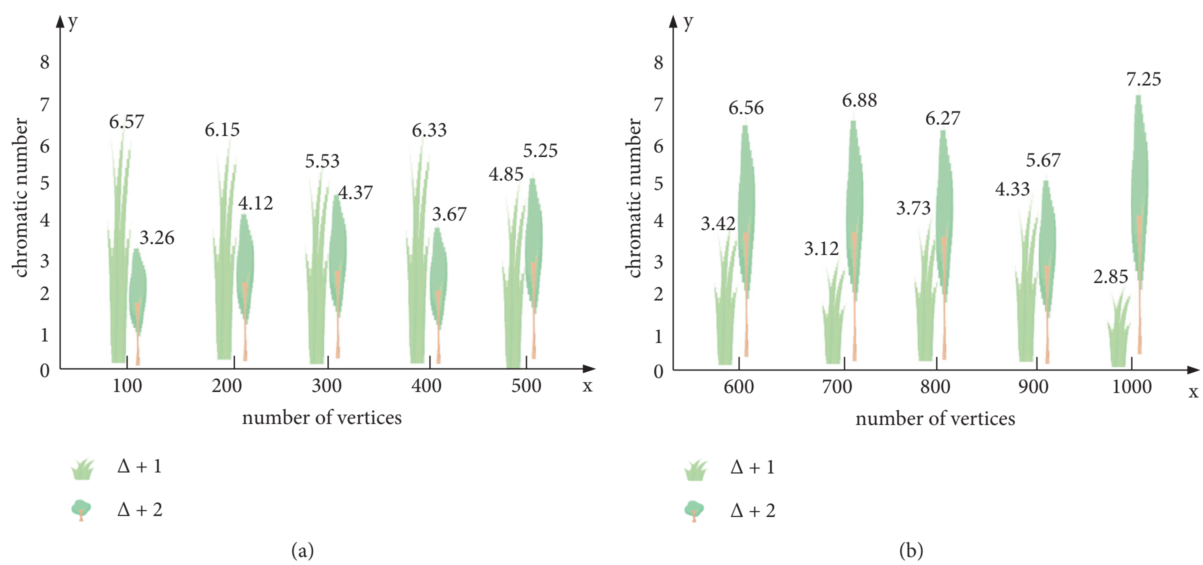


Figure 6: Graph test results. (a) The number of chromatic number graphs with vertices ranging from 100 to 500. (b) The number of chromatic graphs with vertices ranging from 600 to 1000.

Vertex coloring: color vertices are randomly selected from colors 1 to 8, and the vertex colors can be the same.

Edge prestaining: it does random coloring and vertex-associative edge coloring for edges after removing the color used for vertex coloring. The obtained coloring matrix is shown in Table 4:

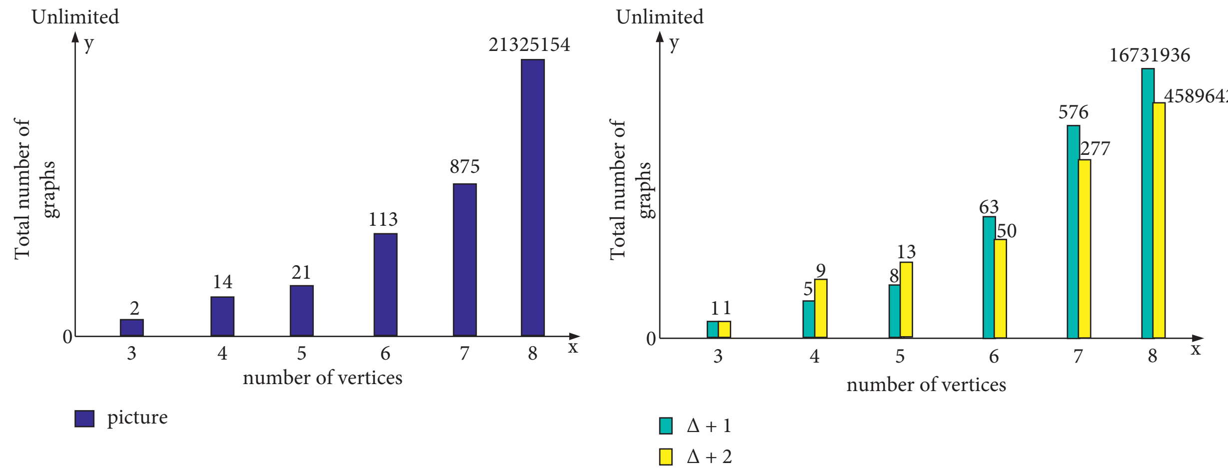
From Table 4, the color complement sets $S\ (V1)$~$S\ (V8)$ of each vertex can be obtained as follows: {2, 6, 7, 8}; {⊘}; {2, 4, 6, 8}; {2, 3, 5}; {2, 4, 8}; {1, 2, 5, 7}; {1, 2}; {1, 2, 6, 7}.

Find conflict sets: if there is a conflict, there is $Q(V_n) + |S(V_n)| \neq k - 1$, and the conflict situation is shown in Table 5.

It can be seen from Table 5 that when the color number k is 8, $Q\ (Vn) + S\ (Vn) = k - 1 = 7$. Among them, $V3$, $V5$, and $V6$ are in conflict, and the staining is abnormal.

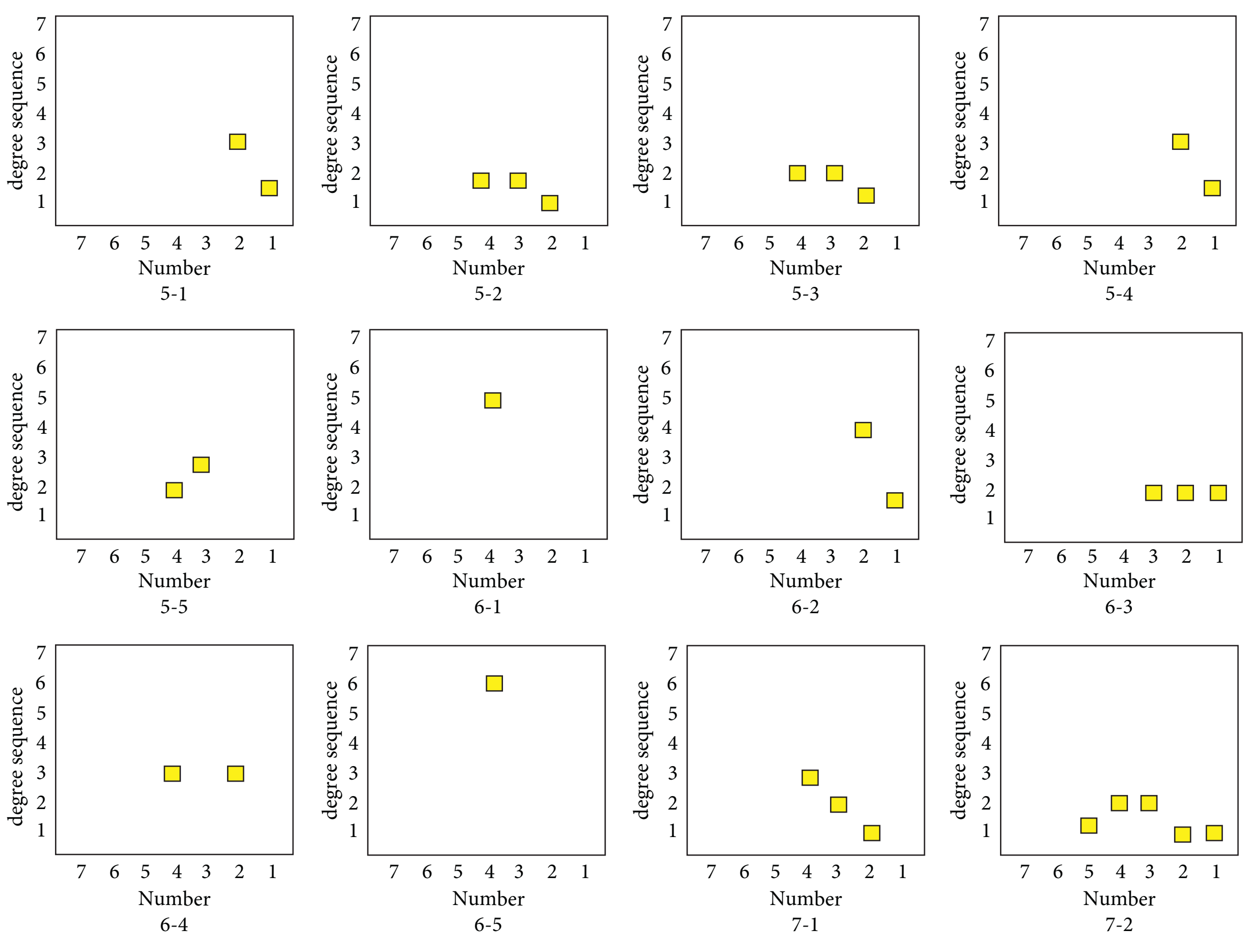
Adjust the coloring conflict: it swaps conflicting colors according to the rules, and then modifies the set of color complements. The new nonuniform dyeing results obtained are shown in Table 6:

As can be seen from Table 6, the staining conflict has been resolved. The color complement sets $S\ (V1)$~$S\ (V8)$ of each vertex are obtained as follows: {2, 4, 7}; {⊘}; {2, 4, 5, 6}; {4, 6}; {2, 4, 6}; {3, 6, 7}; {1, 2}; {1, 2, 6, 7}.



(a) (b)

FIGURE 7: Graph test within $3 \leq n \leq 8$. (a) Total number of graphs. (b) Comparison of the number of images with different chromatic numbers.



(a)

FIGURE 8: Continued.

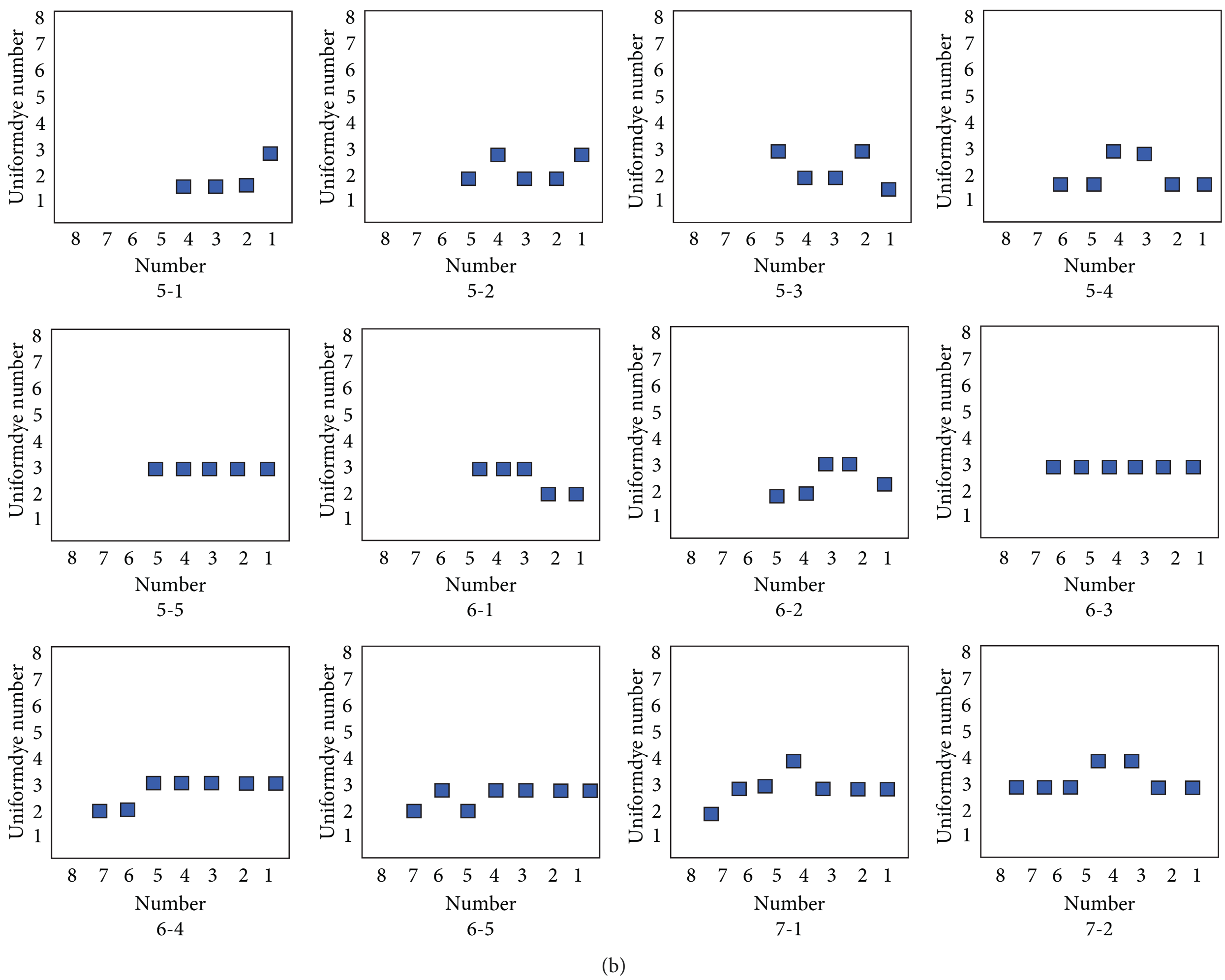


Figure 8: Degree sequence, chromatic number test results. (a) Partial graph degree sequence. (b) The number of uniform dyeing in part of the figure.

Check whether the chromatic number is uniform: as you can see from the table data, the use of color is not uniform. The use of color should continue to be adjusted. After adjusting the use of color according to the algorithm, the final uniform dyeing results are obtained, as shown in Table 7:

According to the data in Table 7, the difference between the number of used colors of any color is less than or equal to 1. It is already average and meets full staining conditions. The color complement sets $S\,(V1)$~$S\,(V8)$ of each vertex are obtained as follows: {4, 6, 8}; {⊘}; {1, 2, 7, 8}; {1, 6}; {2, 4, 8}; {1, 3, 5, 6}; {3, 4}; {1, 2, 7, 8}.

Experimental result data, it tests graphs with vertices ranging from 100 to 1000 and obtains graphs with chromatic numbers of $\triangle + 1$ and $\triangle + 2$, as shown in Figure 6:

As can be seen from Figure 6, the number of $\Delta + 1$ chromatic numbers with vertices ranging from 100 to 1000 is 6.51, 6.15, 5.53, 6.33, 4.85, 3.42, 3.12, 3.73, 4.33, and 2.85, respectively. The overall number of graphs with $\Delta + 1$ chromatic number decreases as the number of vertices increases. The number of $\Delta + 2$ chromatic numbers from 100 to 1000 for fixed-point numbers has risen from 3.26 to 7.25. The overall number of graphs with $\Delta + 2$ chromatic number tends to increase as the number of fixed points increases.

All graphs within $3 \leq n \leq 8$ were tested, as shown in Figure 7:

It can be seen from Figure 7 that there are 2, 14, 21, 113, and 853 nonisomorphic graphs with vertices 3 to 7, respectively, and the number of vertices 8 is all numbers. There are 21325188 isomorphic images in it.

Test the degree sequence with the chromatic number and the number of edges. Due to space limitations, the test results of some graphs are shown in Figure 8:

In Figure 8, the vertex degree distributions in the graph are represented as degree sequences. The number of uses of the color number is reflected in the number of uniform dyeing. Each of the small graphs represents a graph test result.

It tests the graph between 20 and 400 points. Due to the large number of samples, in order to obtain the change curve of color number and edge density, only the coloring results of a part of the chart are listed. The change curve is shown in Figure 9:

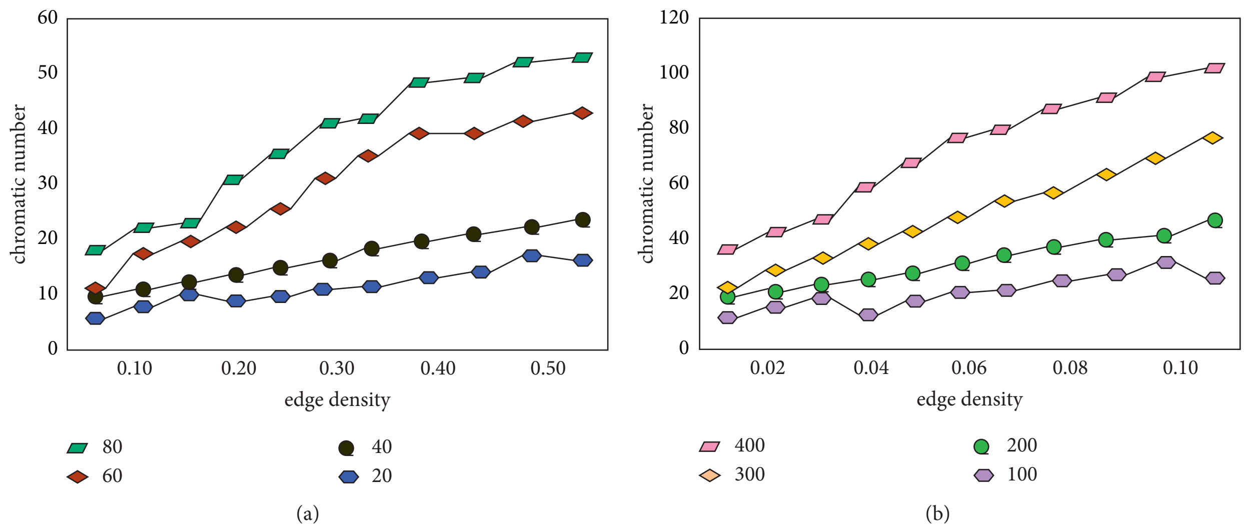


Figure 9: Chromatic number-edge density graph. (a) Change curve 1. (b) Change curve 2.

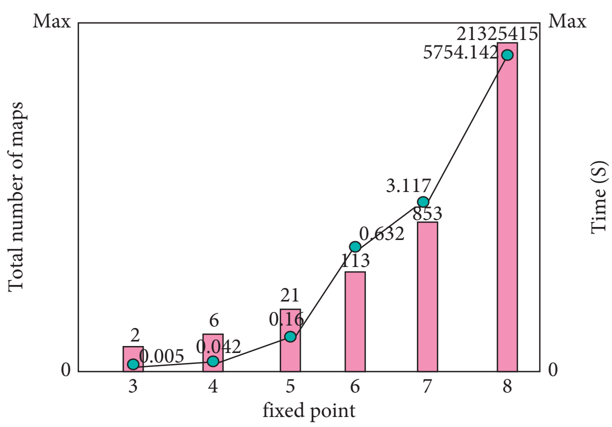


Figure 10: Operation time.

As can be seen from Figure 9, as the edge density and the number of points continue to increase, the structure of the connected graph becomes more complex, and more colors need to be used. So the number of colors increases with edge density and number of dots. Through the analysis of the test results, it can be seen that all simple connection graphs within 8 vertices have AVDEVTC.

Calculation of running time: all graphs of random graphs with 8 vertices were tested using this algorithm, and the running time is shown in Figure 10:

As can be seen from Figure 10, the corresponding running times for the 2, 6, 21, 113, 853, 21325415 graphs are 0.0005 S, 0.042 S, 0.16 S, 0.632 S, 3.117 S, 5754.142 S. As the number of graphs increases, the computation time also increases, and the computation speed tends to decrease. This is due to the reason that the data are to be written to the text file after the result. It can be seen that the algorithm can obtain the dyeing results in a relatively short time.

Through comprehensive experimental tests, it can be seen that the operation speed of the dyeing algorithm based on MOO is faster than that of the traditional method, and it can complete the test results that cannot be obtained manually in a short time. The algorithm can be used to obtain a large amount of research data. Of course, there may be some uncertain factors, such as the instability of the use environment, the difference of operators, and the use time and frequency. The results of this experiment are not completely accurate and reliable and have certain differences.

6. Conclusions

With the continuous upgrading of computer computing, people have higher requirements for the efficiency and accuracy of obtaining data. The development of the field of graph coloring problems is inseparable from the contribution of scientific computing. MOO methods have been widely used in many fields because of their multiconstraint advantages. This article first gives a general introduction to graph coloring and genetic algorithm, so that people can understand their functions and principles. It then analyzes their function using the relevant principle formulas. In the end, it was found that both have great advantages in function. In the experimental part, this paper tests the uniform dyeing algorithm based on MOO. The conclusion is that the algorithm can reduce labor and time consumption while performing uniform full dyeing. Compared with traditional dyeing methods, it is more practical and efficient. The dyeing algorithm proposed in this study based on MOO thinking provides a reference for solving the full dyeing problem with multiple conditions and has certain progress significance. However, the test conditions of this study are limited, and the multicondition constraint graph coloring problem will be more complicated in reality. This application of combining the MOO idea with the graph coloring algorithm will be more valuable and more difficult.

Data Availability

The data used to support the findings of this study are available from the author upon request.

Conflicts of Interest

The author declares no conflicts of interest.

Acknowledgments

This research study was sponsored by Science and Technology Research Project of Colleges and Universities in Inner Mongolia Autonomous Region (nos. NJZY22599 and NJZY22600). The authors thank the project for supporting this article.

References

[1] I. Danaila, P. Joly, S. M. Kaber, and M. Postel, "An introduction to scientific computing," *Texts in Computational Science & Engineering*, vol. 8, no. 2, pp. 246–267, 2017.

[2] H. Elman and D. Furnival, "Solving the stochastic steady-state diffusion problem using multigrid," *IMA Journal of Numerical Analysis*, vol. 27, no. 4, pp. 675–688, 2018.

[3] H. Wilber, A. Townsend, and G. B. Wright, "Computing with functions in spherical and polar geometries II. The disk," *SIAM Journal on Scientific Computing*, vol. 39, no. 3, pp. 238–262, 2017.

[4] J. Dlz, H. Harbrecht, and M. Peters, "h-matrix accelerated second moment analysis for potentials with rough correlation," *Journal of Scientific Computing*, vol. 65, no. 1, pp. 387–410, 2018.

[5] N. Tonicello, G. Lodato, and L. Vervisch, "A comparative study from spectral analyses of high-order methods with non-constant advection velocities," *Journal of Scientific Computing*, vol. 87, no. 3, pp. 1–38, 2021.

[6] V. Druskin, A. V. Mamonov, and M. Zaslavsky, "Distance preserving model order reduction of graph-laplacians and cluster Analysis," *Journal of Scientific Computing*, vol. 90, no. 1, pp. 1–30, 2022.

[7] S. Karaa, K. Mustapha, and A. K. Pani, "Optimal error analysis of a FEM for fractional diffusion problems by energy arguments," *Journal of Scientific Computing*, vol. 74, no. 1, pp. 519–535, 2018.

[8] H. Rashidi and J. Khorshidi, "Exergy analysis and multi-objective optimization of a biomass gasification based multigeneration system," *International Journal of Hydrogen Energy*, vol. 43, no. 5, pp. 2631–2644, 2018.

[9] H. Zille, H. Ishibuchi, S. Mostaghim, and Y. Nojima, "A framework for large-scale multiobjective optimization based on problem transformation," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 2, pp. 260–275, 2018.

[10] D. Do, Trung, V. Duong, Duc, and N. T. Nguyen, "Study on MOO of the turning process of en 10503 steel by combination of taguchi method and MOORA technique," *EUREKA Physics and Engineering*, vol. 7, no. 2, pp. 52–65, 2021.

[11] R. Onler, A. S. Koca, B. Kirim, and E. Soylemez, "MOO of binder jet additive manufacturing of Co-Cr-Mo using machine learning," *International Journal of Advanced Manufacturing Technology*, vol. 119, no. 1-2, pp. 1091–1108, 2021.

[12] F. Kazi, C. A. Waghmare, and M. S. Sohani, "Multi-objective optimization of machining parameters in hybrid powder-mixed EDM process by response surface methodology and normalized fuzzy logic algorithm," *International Journal on Interactive Design and Manufacturing*, vol. 15, no. 4, pp. 695–706, 2021.

[13] M. Babaelahi, S. M. M. Nejad, and M. Saadatfar, "Multi-objective optimization of thermal management system in LED bulbs based on entropy generation minimization concept," *High Temperature*, vol. 58, no. 6, pp. 875–883, 2021.

[14] V. K. Yadav, P. Kumar, and G. Bhushan, "Multi-objective optimization in geometric design of active magnetic bearing based on force slew rate, overall volume, and total losses considerations through genetic algorithms," *Journal of the Institution of Engineers: Series C*, vol. 102, no. 6, pp. 1473–1487, 2021.

[15] O. Cetinaslan, "Position-based simulation of elastic models on the GPU with energy aware gauss-seidel algorithm," *Computer Graphics Forum*, vol. 38, no. 8, pp. 41–52, 2019.

[16] F. Toufic and K. Alqhtani, "Study of nurses' roster by coloring algorithm parsus manually," *Filomat*, vol. 33, no. 2, pp. 561–569, 2019.

[17] S. Agrawal and V. Chhabra, "A novel exact heuristic graph coloring algorithm based on finding independent set," *International Journal of Computer Application*, vol. 179, no. 8, pp. 15–18, 2017.

[18] P. Micek and V. Wiechert, "An on-line competitive algorithm for coloring bipartite graphs without long induced paths," *Algorithmica*, vol. 77, no. 4, pp. 1060–1070, 2017.

[19] G. Kara and C. Özturan, "Algorithm 1002," *ACM Transactions on Mathematical Software*, vol. 45, no. 4, pp. 1–28, 2019.

[20] A. Dey, A. Agarwal, P. Dixit et al., "A genetic algorithm for total graph coloring," *Journal of Intelligent and Fuzzy Systems*, vol. 37, no. 6, pp. 7831–7838, 2019.