Journal of Engineering Science and Technology Review 14 (6) (2021) 107 - 115

JOURNAL OF
Engineering Science and Technology Review

www.jestr.org

Research Article

A Novel Algorithm for Vertex Sum Distinguishing Total Coloring Based on Multiobjective Optimization

Zhao Huanping[1,*], Xue Dangqin[2] and Shi Huojie[3]

[1]*School of Computer and Software, Nanyang Institute of Technology, Nanyang 473004, China*
[2]*School of Intelligent Manufactuing, Nanyang Institute of Technology, Nanyang 473004, China*
[3]*Department of Biological Systems Engineering, Washington State University, Pullman, MI 98101, United States*

Received 29 July 2021; Accepted 13 November 2021

Abstract

Graph coloring is an important research field in graph theory. In vertex sum distinguishing total coloring, the constraint condition, that is, the color sum of each vertex is different from that of its associated edges is added on the basis of the normal total coloring. A novel algorithm was designed to increase the vertex sum distinguishing total coloring effciency. Subobjective functions were designed according to the vertex constraint, edge constraint, vertex-edge constraint, and vertex-edge sum constraint to improve the vertex sum distinguishing total coloring efficiency of random graphs. In each subobjective function, iterative swap was performed step by step using the coloring matrix and color set complement, in which the conflict was handled following the idea of impressed variation, thereby realizing the vertex sum distinguishing total coloring of random graphs. In the end, the lower bound of chromatic number in the vertex sum distinguishing total coloring was proven through the theoretical analysis and experimental comparison. Results demonstrate that when the order of a graph is smaller than 1000 and the edge density is smaller than 0.1, the conflict handling function can be converged at a high rate. When the order of the graph is greater than 1000, the convergence rate of the algorithm declines rapidly, and the accurate coloring result can be obtained only by increasing the coloring quantity, indicating that the vertex sum distinguishing total chromatic number of a connected graph with the order of not smaller than 3 ranged from $\mu_{\Sigma t}(G)$ to $\mu_{\Sigma t}(G)+3$. This finding is consistent with the algorithm analysis result. On this basis, this algorithm can effectively complete the vertex sum distinguishing total coloring of random graphs, and its time complexity does not exceed $O(n^3)$. The proposed algorithm provides evidence for exploring complex networks, transportation, and network security.

Keywords: Multiobjective optimization, Vertex sum distinguishing total coloring, Vertex sum distinguishing total chromatic number, Objective function

1. Introduction

Graph coloring is a very classical problem in the field of graph theory. Competition arrangement, network communication problem, curriculum schedule arrangement problem, material storage problem, printed circuit board (PCB) design, and index register design can be converted into the graph coloring problem for solving [1-6]. Thus, the research on graph coloring has great theoretical and realistic significance. Ordinary intelligent algorithms are suitable for the graph coloring problem based on single-objective optimization. However, they have deficiencies and limitations when used to solve the graph coloring problem under multiple restrictive conditions. Therefore, rapidly and efficiently obtaining multicondition constrained coloring results is a key problem to solve.

The present studies on graph coloring mainly include theoretical and algorithm studies. In recent years, many scholars all over the world have carried out extensive theoretical studies on graph coloring. Various concepts and conjectures, such as adjacent vertex distinguishing edge coloring, vertex distinguishing edge coloring, adjacent vertex distinguishing total coloring, and vertex distinguishing total coloring, are successively proposed [7-12]. However, the graph coloring problem is an NP-complete problem. At present, public intelligent algorithms [13-15], such as genetic algorithm (GA), ant colony algorithm, and simulated annealing algorithm, show very good execution efficiency when solving the graph coloring problem with small scale and low complexity. However, they can be easily caught in local optimum, accompanied by a high calculation cost. GA is initially used to generate initial solutions, followed by the variable-field search through tabu search algorithm to update the vertex coloring given their characteristics [16], thereby accelerating the algorithm search. Nevertheless, these coloring algorithms have very great limitations when faced with the graph coloring problem under multiple constraint conditions, such as vertex sum distinguishing total coloring. Multiobjective optimization aims to solve the optimization problem containing multiple objective functions and constraint conditions. The multiobjective optimization was applied to vertex sum distinguishing total coloring of graphs. This problem was divided into multiple subproblems corresponding to respective coloring conditions. A subobjective function vector and a decision space were set for each subproblem. Moreover, each subobjective function gradually obtained the optimal solution in the color exchange process, finally enabling the total objective

*E-mail address: 34875716@qq.com
ISSN: 1791-2377 © 2021 School of Science, IHU. All rights reserved.
doi:10.25103/jestr.146.12

function to satisfy the requirements for vertex sum distinguishing total coloring of graphs.

2. State of the art

Graph coloring is a very active research subject in graph theory. Since Burris [17] proposed the concept of edge distinguishing edge coloring of graphs, it has been extensively investigated by a number of domestic and foreign scholars[18-21]. Flandrin [12] et al. studied the adjacent sum distinguishing edge coloring problem of graphs and proposed a conjecture for a connected graph G(≠G5) containing at least three vertexes, i.e., $\Delta(G) \le \chi\sum(G) \le \Delta(G)+2$. Pilsnisk [22] et al. put forward the concept and conjecture of adjacent sum distinguishing total coloring, namely, $\chi(G) \le \Delta(G)+3$ held true for a simple graph with the orders of at least three. They proved that this conclusion was also true in complete graphs, circles, and bipartite graphs. Dong and Wang [23] proved that the conjecture of adjacent sum distinguishing total coloring was also true for sparse graphs. Cui F X, Yang C, and Ye H B [24] et al. studied the adjacent vertex sum distinguishing total coloring of joint graphs and obtained the precise values of the adjacent vertex sum distinguishing total chromatic numbers in three types of joint graphs, such as road–road, road–circle, and circle–circle. Tian S L and Yang H [25] et al. obtained the precise values of the adjacent sumdistinguishing edge chromatic numbers for the lexicographic product P_n[H] of a path P_n and a connected simple graph H. The graph coloring problem is an extensively studied combinational optimization problem, which receives high attention from a number of researchers. The methods used to solve the graph coloring problem are mainly divided into two types, namely, heuristic method and accurate computation method, where the former mainly includes greedy method, local search method, and hybrid evolution method. For instance, Basmassi and Benameur [26]et al. combined a heuristic algorithm with an intelligent algorithm, used the greedy sequential algorithm to solve the genetic operator and correct the unrealizable solution after crossover and mutation, upgraded the population by improving the coloring quality of chromosomes, and improved the algorithm convergence rate. This hybrid algorithm achieved the expected coloring result in solving the vertex coloring and edge coloring problems of graphs. Sun W and Hao J K [27] et al. combined the coarsening procedure with the refinement procedure to improve the coarse graph through a refinement program, ensuring that the graph could be continuously improved in the coarsening phase. In addition, the designed weight tabu coloring algorithm showed excellent operating efficiency in weight graphs especially sparse graphs. However, when the graph density was increased, its operating time presented an exponential increasing trend. Artacho and Campoy [28] improved the Douglas–Rachford algorithm and used the overall programing formula to calculate the binary index variables and effectively solve the problem, that is, the algorithm could be easily caught in a limit cycle. This algorithm showed good performance in solving the graph precoloring problem, but its operating efficiency was related to the selection of starting point. If the starting point was inappropriately selected, then the algorithm convergence could be slow, with long execution time. Arindam [29] et al. applied the improved GA to the total coloring of a graph, used a new coding scheme to express the vertex and edge of the graph, and determined the total chromatic number of the graph through the greedy algorithm as the fitness value of chromosomes to accelerate the algorithm convergence. This algorithm could acquire the coloring results of standard graphs rapidly. The accurate computation method is mainly represented by the backtracking method and branch and bound method. The branch and bound algorithm based on the heuristic algorithm DSATUR is effective in accurately solving vertex coloring, but its lower bound can be calculated only once without updating. To solve this problem, Furini and Ternier [30] et al. introduced reduced graph to calculate the lower bound of the branch node. For high-density vertex coloring, these bound methods shorten the operating time and reduce the number of nodes. On the contrary, for medium and low-density vertex coloring, the bound technique can only effectively reduce the number of explored nodes. However, it fails to effectively shorten the operating time. The studies on graph coloring problem are mainly concentrated on vertex coloring and edge coloring. Other coloring problems have also been investigated. For example, Li J W [31-32] et al. studied vertex distinguishing total coloring of graphs and effectively solved the vertex distinguishing total chromatic number of a random graph with the given number of vertexes. The time complexity of this algorithm did not exceed $O(n^3)$.

The domestic and foreign studies on the vertex sum distinguishing total coloring of graphs are still in the initial stage. The vertex sum distinguishing total coloring algorithms for graphs have been rarely involved. The constraint conditions for the vertex sum distinguishing total coloring of graphs are much more complicated than vertex coloring and edge coloring. However, the traditional intelligent algorithms are suitable to solve vertex coloring and edge coloring with low complexity, but cannot solve the vertex sum distinguishing total coloring problem of graphs. The research idea of the multiobjective optimization was combined to convert the coloring conditions of the algorithm into multiple subobjective functions. Subsequently, the conflict of coloring conditions was handled using the idea of pre-exchange and mutation. Subsequently, the solution space was searched by means of stepwise optimization, thereby accelerating the algorithm convergence.

The remainder of this study is organized as follows: the concept of vertex sum distinguishing total coloring, the subobjective functions of vertex sum distinguishing total coloring algorithm, and the relevant algorithm design are described in Section 3. In Section 4, the experimental results and algorithm analysis are presented. In the last section, the entire paper is summarized, and conclusions are drawn.

3. Methodology

3.1 Related definitions

For any undirected graph $G(V,E)$, $V(G)$ stands for the vertex set of graph G, $E(G)$ is the edge set of graph G, $C(u)$ represents the color set used by the vertex u and its associated edges in graph G, and $\overline{C}(u)$ denotes the complement for the color set of the vertex u. On this basis, the related definitions in the coloring of graph G are as follows:

Definition 1 [12]: For graph $G(V,E)$, the mapping f: $E(G)\rightarrow\{1,2,\dots,k\}$enables $f(e) \neq f(e')$ any adjacent edges e and e', and then f is called a k-normal edge coloring of G, abbreviated as *k-PEC*.

Definition 2 [12]: For graph $G(V,E)$,the mapping f: $E(G)\cup V(G)\to\{1,2,...,k\}$ satisfies the following conditions:

1) $\forall uv,uw\in E(G)$, $v\neq w$ and $f(uv)\neq f(uw)$;

2) $\forall uv\in E(G)$, $f(u)\neq f(v), f(v)\neq f(uv), f(u)\neq f(uv)$

then f is referred to as a k-normal total coloring of G, abbreviated as *k-PTC*. $\chi_T(G)$ =min $\{k|$ *k-PTC* for $G\}$ is called the total chromatic number of G.

Definition 3 [12]: For a simple connected graph $G(V,E)$ with the orders of not less than 2, if the mapping f: $V(G)\cup E(G)\to\{1,2,\ldots,k\}$ satisfies the following conditions:

1) $\forall uv,uw\in E(G)$, $v\neq w$, and $f(uv)\neq f(uw)$;

2) $\forall uv\in E(G)$, $S(u)\neq S(v)$ and $S(u)=\sum_{uv\in E(G)} f(uv)$

then f is regarded as k-adjacent vertex sum distinguishing edge coloring of G, abbreviated as *k-NSDEC*. $\chi_{n\Sigma e}$= min $\{k|$ *K-NSDEC* for $G\}$ is considered the adjacent vertex sum distinguishing edge chromatic number of G.

Conjecture 1 [12]: For a connected graph with the orders of at least three but not reaching five, the following is true:

$$\chi_{n\Sigma e}\leq\Delta(G)+2 \tag{1}$$

where △(G) is the maximum number of degrees of the graph.

Definition 4 [27]: For a simple connected graph G(V,E) with the orders of not less than two, if the mapping f: $V(G)\cup E(G)\to\{1,2,\ldots,k\}$ satisfies the following:

1) $\forall uv,uw\in E(G)$, $v\neq w$, and $f(uv)\neq f(uw)$;

2) $\forall uv\in E(G)$, $f(u)\neq f(v)$, $f(v)\neq f(uw)$, $f(u)\neq f(uw)$

3) $\forall uv\in E(G)$, $S(u)\neq S(v)$, $S(u)=\sum_{uv\in E(G)} f(uv)+f(u)$

then f is called the k-adjacent vertex sum distinguishing total coloring of G, abbreviated as *k-NSDTC*. $\chi_{n\Sigma T}$= min $\{k|$ *K-NSDTC* for $G\}$ is considered the adjacent vertex sum distinguishing total chromatic number of G.

Definition 5: For a simple connected graph $G(V,E)$ with the orders of not less than two, if the mapping f: $E(G)\cup V(G)\to\{1,2,...,k\}$ satisfies the following:

1) $\forall uv,uw\in E(G)$, $v\neq w$, and $f(uv)\neq f(uw)$;

2) $\forall uv\in E(G)$, $f(u)\neq f(v)$, $f(v)\neq f(uw)$, $f(u)\neq f(uw)$

3) $\forall u,v\in V(G)$, $S(u)\neq S(v)$, $S(u)=\sum_{uv\in E(G)} f(uv)$

then f is referred to as k-vertex sum distinguishing total coloring of G, abbreviated as *k-VSDTC*. $\chi_{\Sigma T}$= min $\{k|$ *K-VSDTC* for $G\}$ is the vertex sum distinguishing total chromatic number of G.

Conjecture 2: For a random connected graph G with the orders of not less than three, the following condition is satisfied:

$$\chi_{\Sigma T}(G)\geq\max(\chi^1_{\Sigma T}(G),\chi^2_{\Sigma T}(G)) \tag{2}$$

where $\chi^1_{\Sigma T}(G)=\max\{\left\lceil\frac{n_j+d_j^2+2d_j}{d_j+1}\right\rceil | j=1,2,...,l\}$ and

$$\chi^2_{\Sigma T}(G)=\max\{\left\lceil\frac{\sum_{i=1}^{j}n_i+(d_j+1)\times d_j/2+d_1\times(d_1+1)/2-1}{d_j+1}\right\rceil | j=1,2,...,l\}$$

$d_1<d_2<...<d_l$ represents the different degrees of vertexes in the graph G, and n_j is the number of vertexes with the degree of d_j.

3.2 Establishment of constraint functions

The definition of vertex sum distinguishing total coloring graphs indicates that this algorithm must satisfy the following constraint conditions: (a) color difference between adjacent edges; (b) color difference between adjacent vertexes; (c) color difference between vertex and its associated edges; (d) the color sum of all vertexes in the graph is unequal. According to these constraint conditions, the constraint functions are defined as follows:

3.2.1 Edge constraint function

The mapping f: $E(G)\to\{1,2,\ldots,k\}$ exists for graph $G(V,E)$, and the edge constraint function is defined as follows: $e_i,e_j\in E(G)$ is set, and e_i and e_j are adjacent edges; moreover,

$$f_1(e_i,e_j)=\begin{cases}1 & f(e_i)=f(e_j)\\ 0 & otherwise\end{cases} \tag{3}$$

Then:

$$F_1(e_i,e_j)=\sum_{e_i,e_j\in E(G)} f_1(e_i,e_j) \tag{4}$$

$F_1(e_i,e_j)$ denotes the number of edges that do not satisfy the constraint condition (a), which is satisfied when and only when $F_1(e_i,e_j)$=0.

3.2.2 Vertex constraint function

The mapping f: $E(G)\to\{1,2,\ldots,k\}$exists for graph $G(V,E)$, and the vertex constraint function is defined as follows, where $uv\in E(G)$ is set:

$$f_2(u,v)=\begin{cases}1 & f(u)=f(v)\\ 0 & otherwise\end{cases} \tag{5}$$

Then:

$$F_2(u,v)=\sum_{uv\in E(G)} f_2(u,v) \tag{6}$$

$F_2(u,v)$ stands for the number of vertexes that do not satisfy the constraint condition (b), which is satisfied when and only when $F_2(u,v)$=0.

3.2.3 Vertex–edge constraint function

The mapping f: $\{V(G),E(G)\}\to\{1,2,\ldots,k\}$ exists for graph $G(V,E)$, and the vertex–edge constraint condition is defined as follows, where $uv\in E(G)$ is set:

$$f_3(v,e)=\begin{cases}1 & f(uv)=f(u)\quad or\ f(uv)=f(v)\\ 0 & otherwise\end{cases} \tag{7}$$

Then:

$$F_3(v,e)=\sum_{v\in V(G),e\in E(G)} f_3(v,e) \tag{8}$$

$F_3(v,e)$ represents the number of vertexes that do not satisfy the constraint condition (c), which is satisfied when and only when $F_3(v,e)=0$.

3.2.4 Vertex sum constraint function

The mapping f: $\{V(G),E(G)\}\rightarrow\{1,2,\ldots,k\}$ exists for graph $G(V,E)$, and $S(u)=\sum_{u\in V(G)} f(u)$ is set as the color sum of the associated edges of the vertex u. The vertex sum constraint function is defined as follows:

For any $u,v\in V(G)$, the following are set:

$$f_4(u,v)=\begin{cases}1 & S(v)=S(u)\\ 0 & otherwise\end{cases} \tag{9}$$

Then:

$$F_4(u,v)=\sum_{u,v\in V(G)} f_4(u,v) \tag{10}$$

$F_4(u,v)$ denotes the number of vertexes that do not satisfy the constraint condition (d), which is satisfied when and only when $F_4(u,v)=0$.

3.2.5 Total objective function

$$F_{\Sigma T}=F_1+F_2+F_3+F_4 \tag{11}$$

$F_{\Sigma T}$ stands for the number of vertexes that do not satisfy the four coloring conditions, and the coloring succeeds only when $F_{\Sigma T}=0$.

3.3 Algorithm design and description

Algorithm 1: generating a simple random connected graph

Input: non-isomorphic simple connected subgraphs with minimum edges

Output: adjacency matrix of graph G

Step 1) The number of vertexes (n) of graph G is input, as well as the matrix C_{n*n} with n vertexes constituting an underlying graph; it refers to all non-isomorphic connected graphs containing n-1 edges and n vertexes.

Step 2) The set N[]is used to number the positions with the value of 0 above the main diagonal of *color*[][]. Without considering the situation of isomorphic graphs, $sum=2_{m-1}$ n-vertex random graphs can be generated by this underlying graph, and each graph corresponds to one element in the set $T=\{0,1,\ldots,sum\}$.

Step 3) The symmetric numbers are determined by considering the counter-diagonal matrix as the axis of symmetry and then save them in the array *sysN*[], which is inverted and spliced to the tail part of *sysM*[], where the number 2 is saved in *sysN*[1].

Step 4) The value 1 is set at the corresponding position in the matrix, thereby generating a random graph based on the present underlying graph. The random graph C_{n*n} based on this underlying graph can be obtained by such a cycle.

Algorithm 2: edge coloring algorithm

Input: adjacency matrix C_{n*n} of the graph

Output: normal edge coloring matrix

Step 1) Random coloring of the adjacency matrix in graph G is performed. The complement *comset*[]and sorted collection *manyc*[]are obtained according to the C after coloring, where the former stores the colors that do not appear at the edges associated with the vertex v_i, and the vertexes are saved in *manyc*[] in a descending order according to the color set complement.

Step 2) $color[0][0]=\sum_{i=1}^{n}(comset[i][0]-color[i][0])$ is set, and whether the value of *color*[0][0] is 0 is determined; if yes, then the subobjective function is $F_1=0$, and the algorithm ends; otherwise, proceed to **Step 3)**.

Step 3) vertex u is removed from *manyc*[] in a certain order and compared it with the vertex v. When an edge exists between the two vertexes, $color[u][v]\neq0$, $comset[u][]\cap comset[][v]=\{a_1,a_2,\ldots,a_i\}$, where a_i represents the color i (i is an integer from 1 to k). The edge uv is colored into the color a_1, and *comset*[u][] and *comset*[][v] are simultaneously altered. When each vertex is compared with all the vertexes, except the last vertex in *manyc*[], the one round of transformation is completed.

Step 4) The value of *color*[0][0] after one round of transportation is recalculated. If the value is 0, then the algorithm ends; otherwise, a new sorted collection *manyc*[]is regenerated according to the updated color set complement *comset*[][], proceed to **Step3)**, and the next round of adjustment is performed.

Algorithm 3: detect and handle vertex-edge color sum conflict

Input: adjacency matrix C_{n*n} of normal edge coloring of graph G

Output: the adjacency matrix after the color sum conflict of graph G is handled

Step 1) The color sum *colorSum*[]of associated edges of each vertex is calculated, whether *colorSum*[i]=*colorSum*[j] exists is judged; if yes, then proceed to **Step 2)**; otherwise, no color sum conflict is found in the present coloring result, and then the algorithm ends.

Step 2) The vertexes are pre-exchanged with those not adjacent to the vertexes with different color sums once, the conflict pairs after the exchange are calculated, the exchange vertexes with the minimum conflict pair (*cf_i*) are determined and marked, and the concrete exchange process is described as follows:

Step 2.1) The $v_p(p\neq i,j)$ that satisfies the conditions for the vertex v_i is determined, and $colorSum[v_i]\neq colorSum[v_p]$ and $color[v_i][v_p]\neq 0$ are required.

Step 2.2) If the color set complements $\overline{C_{vi}}\cap\overline{C_{vp}}=\phi$, then the pre-exchange cannot be performed, and $cf\_i = n*n$.

Step 2.3) If the color set complements $\overline{C_{vi}}\cap\overline{C_{vp}}\neq\phi$, then all the color sum conflict pairs *cf_i* are calculated after the common colors in the color set complement *comset*[*vp*][*mm*] are exchanged with those in the edge color set *comset*[v_p][v_i], and the present vertex is marked as v_p.

Step 2.4) All the vertexes that satisfy the conditions are pre-exchanged, and then the minimum color sum conflict pair *cf_i* is obtained.

Step 3) The vertex v_j is pre-exchanged with the other adjacent vertexes with different edge color sums except for v_i, and the conflict pair *cf_j* is calculated after the exchange.

Step 4) The total color sum conflict pairs *cf__i* and *cf__j* are used to judge after the exchange, and the locally optimal exchange is implemented, specifically as follows:

Step 4.1) If *cf_i*=*cf_j*=$n*n$ and $\overline{C_{vi}}=\overline{C_{vj}}=\{a_1,a_2,\ldots,a_m\}$, then exchange v_i and v_j, *i.e.*, $color[v_i][v_j]=color[v_j][v_i]=a_1$, and then simultaneously alter $\overline{C_{vi}}$ and $\overline{C_{vj}}$.

Step 4.2) If *cf_i*=*cf_j*=$n*n$, and $\overline{C_{vi}}$ is φ or $\overline{C_{vj}}$ is φ, then the exchange follows the idea of impressed variation;

Step 4.3) If *cf_i*=*cf_j*=$n*n$ and $\overline{C_{vi}}=\overline{C_{vj}}(v_iv_j\notin E(G)$, then the present vertexes with color sum conflict is skipped, and the next pair of vertexes with color sum conflict is determined;

Step 4.4) If *cf_i*≤*cf_j*and it is unequal to $n*n$, then the exchange in Step (2) is completed;

Step 4.5) If $cf_j \le cf_i$and it is unequal to $n*n$, then the exchange in **Step 3)** is completed;

Return to **Step 1)** and retest whether any edge color sum conflict exists.

Algorithm 4: vertex sum distinguishing total coloring algorithm

Input: orders n of the graph G

Output: vertex sum distinguishing total coloring matrix of the graph G

Step 1) An initial adjacency matrix C_{n*n} of the graph G is generated using Algorithm 1 according to the orders n;

Step 2) The edge coloring of the graph G is completed according to Algorithm 2, and **Step 3)** is performed after the coloring succeeds;

Step 3) The color sum of the associated edges of each vertex is calculated, and the vertex–edge color sum conflict is adjusted using Algorithm 3;

Step 4) The vertexes are colored using each color set complement;

Step 5) The color sum of each vertex and its associated edges are calculated, and the vertex total color sum conflict is adjusted through the Algorithm 3.

4 Result analysis and discussion

4.1 Algorithm test

According to these algorithm steps, the 10-vertex graph G is considered to provide the test results.

Step 1) A random graph is generated

According to the Algorithm 1, a underlying graph is selected from the underlying graphs of 10-vertex . On this basis, a random graph is generated as the original graph for algorithm testing, and the initial matrix of 10-vertex simple connected graph is shown in Fig.1.

$$\begin{array}{cc} & \begin{array}{cccccccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 & v_9 & v_{10} \end{array} \\ \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \\ v_9 \\ v_{10} \end{array} & \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

Fig. 1. Initial Matrix of Graph G

The degree sequence of each vertex is calculated, as shown in Table 1:

Table 1. Number of Vertexes under Each Degree in Graph G

Degree	9	8	7	6	5	4	3	2	1
Number of vertices	2	1	3	3	1	0	0	0	0

Step 2) Edge coloring of graph G

Fig.1 and Table 1 show that over two vertexes with the maximum degrees are adjacent in graph G. Thus, the minimum color number required is color_n= △ +2=11. *color_n*++ is executed because the color sum is equal among multiple vertexes during the final total color sum adjustment, followed by the edge coloring of graph G using *color_n*=12colors according to the coloring rule of the Algorithm 2. In addition, the conflict is handled by invoking rules until F_1=0. The normal edge coloring results are displayed in Fig.2 and Table 2.

$$\begin{array}{cc} & \begin{array}{cccccccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 & v_9 & v_{10} \end{array} \\ \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \\ v_9 \\ v_{10} \end{array} & \begin{pmatrix} \ddots & 6 & 1 & 9 & 10 & 3 & 2 & 8 & 11 & 4 \\ 6 & \ddots & 10 & 3 & 0 & 7 & 0 & 0 & 1 & 5 \\ 1 & 10 & \ddots & 4 & 8 & 12 & 11 & 5 & 9 & 3 \\ 9 & 7 & 4 & \ddots & 2 & 1 & 3 & 12 & 0 & 0 \\ 10 & 0 & 8 & 2 & \ddots & 0 & 0 & 7 & 4 & 1 \\ 3 & 4 & 12 & 1 & 0 & \ddots & 10 & 9 & 8 & 0 \\ 2 & 0 & 11 & 3 & 0 & 10 & \ddots & 4 & 5 & 0 \\ 8 & 0 & 5 & 12 & 7 & 9 & 4 & \ddots & 3 & 2 \\ 11 & 1 & 9 & 0 & 4 & 8 & 5 & 3 & \ddots & 0 \\ 4 & 5 & 3 & 0 & 1 & 0 & 0 & 2 & 0 & \ddots \end{pmatrix} \end{array} \quad \begin{array}{c} \overline{C_i} \\ \begin{pmatrix} 5 & 7 & 12 & & & & \\ 2 & 4 & 8 & 9 & 11 & 12 & \\ 2 & 6 & 7 & & & & \\ 5 & 6 & 8 & 10 & 11 & & \\ 3 & 5 & 6 & 9 & 11 & 12 & \\ 2 & 5 & 6 & 7 & 11 & & \\ 1 & 6 & 7 & 8 & 9 & 12 & \\ 1 & 6 & 10 & 11 & & & \\ 2 & 6 & 7 & 10 & 12 & & \\ 6 & 7 & 8 & 9 & 10 & 11 & 12 \end{pmatrix} \end{array}$$

Fig. 2. Coloring Results after Edge Coloring of Graph G

Table 2. Vertex Color Sum after Edge Coloring

vertex	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
Sum of colors	54	32	53	34	32	40	35	50	41	15

Step 3) Edge color sum conflict is adjusted according to Algorithm 3 (edge color sum conflict adjustment algorithm)

All vertexes in graph G are reranked in a descending order according to the number of elements in the color set complement, and then the sorted collection *manyc*[]={v_{10},v_2,v_5,v_7,v_4,v_6,v_9,v_8,v_1,v_3}is obtained. The color sum conflict pairs are sought from v_1; the first conflict pair (v_2,v_5) is obtained, v_2 is pre-exchanged with the other vertexes to obtain the minimum conflict pair cf_i =1, and the exchange point is v_6. Similarly, v_5 is exchanged with the other vertexes to obtain the minimum conflict pair cf_i=2, and the exchane point is v_8. As v_6 and v_8 share a common color 6, the edge colors of v_6 and v_8 are exchanged, whether edge color sum conflict exists is tested. The adjusted coloring results are presented in Fig.3.

$$\begin{array}{cc} & \begin{array}{cccccccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 & v_9 & v_{10} \end{array} \\ \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \\ v_9 \\ v_{10} \end{array} & \begin{pmatrix} \ddots & 6 & 1 & 9 & 10 & 3 & 2 & 8 & 11 & 4 \\ 6 & \ddots & 10 & 3 & 0 & 4 & 0 & 0 & 1 & 5 \\ 1 & 10 & \ddots & 4 & 8 & 12 & 11 & 5 & 9 & 3 \\ 9 & 3 & 4 & \ddots & 2 & 1 & 7 & 12 & 0 & 0 \\ 10 & 0 & 8 & 2 & \ddots & 0 & 0 & 7 & 4 & 1 \\ 3 & 4 & 12 & 1 & 0 & \ddots & 10 & 9 & 8 & 0 \\ 2 & 0 & 11 & 7 & 0 & 10 & \ddots & 4 & 5 & 0 \\ 8 & 0 & 5 & 12 & 7 & 9 & 4 & \ddots & 3 & 2 \\ 11 & 1 & 9 & 0 & 4 & 8 & 5 & 3 & \ddots & 0 \\ 4 & 5 & 3 & 0 & 1 & 0 & 0 & 2 & 0 & \ddots \end{pmatrix} \end{array}$$

Fig. 3. Coloring Results after Edge Color Sum Conflict Adjustment in Graph G

Table 3. Vertex–Edge Color Sum after Edge Color Sum Conflict Adjustment in Graph G

vertex	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
Sum of colors	54	29	53	38	32	37	39	50	41	15

As shown in Table 3, the edge color sum varies from vertex to vertex, and the adjustment of edge color sum conflict is completed.

Step 4) Vertex coloring

The color set complement of each vertex is shown in Table 4.

Table 4. Statitical Table of Color Set Complements after Normal Edge Coloring of Graph G

vertex	current element number of the color complement	$\overline{C_{vi}}$: color complement
v_1	3	5 7 12
v_2	6	2 7 8 9 11 12
v_3	3	6 7 12
v_4	5	5 6 8 10 11
v_5	6	3 5 6 9 11 12
v_6	5	5 6 7 11 12
v_7	6	1 3 6 8 9 12
v_8	4	1 6 10 11
v_9	5	2 6 7 10 12
v_{10}	7	6 7 8 9 10 11 12

The vertexes in *manyc*[]are resorted in an ascending order of the complements to regenerate a sorted collection: *manyc*[] ={ v_1, v_3, v_8, v_4, v_6, v_9, v_2,v_5,v_7, v_{10} }, and the vertex sequence for coloring is the vertex sequence in *manyc*[]. The vertex v_1 is colored. The vertex v_1 is colored with the header element 5 in its complement because no coloring conflict exists between adjacent vertexes in this case. If the header element 6 is selected for the vertex v_3, then no color conflict between adjacent vertexes is observed. Vertex v_3 is then colored with the color 6. Similarly, vertex v_8 is colored with color 10. When vertex v_4 is colored, the header element color 5 and second element color 6 in its color set complement are already selected by the adjacent vertexes v_1 and v_3. Thus, v_4 is colored with the third element color 8 in its color set complement. The remainder can be deduced by analogy, namely, the other vertexes are colored with the following colors: 7 (v_6), 2 (v_9), 9 (v_2), 3 (v_5), 3 (v_7), and 7 (v_{10}). Under this circumstance, the colors of any two adjacent vertexes in the graph are different. Therefore, the vertex coloring is completed. In other words, the color for each vertex in $F_3(v,e)=0$ is selected from the color set complement after the normal edge coloring is completed for the vertex. Hence, $F_4(u,v)=0$. The coloring results in this case are as shown in Fig. 4.

$$
\begin{array}{c|cccccccccc}
 & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 & v_9 & v_{10} \\
\hline
v_1 & 5 & 6 & 1 & 9 & 10 & 3 & 2 & 8 & 11 & 4 \\
v_2 & 6 & 9 & 10 & 3 & 0 & 4 & 0 & 0 & 1 & 5 \\
v_3 & 1 & 10 & 6 & 4 & 8 & 12 & 11 & 5 & 9 & 3 \\
v_4 & 9 & 3 & 4 & 8 & 2 & 1 & 7 & 12 & 0 & 0 \\
v_5 & 10 & 0 & 8 & 2 & 3 & 0 & 0 & 7 & 4 & 1 \\
v_6 & 3 & 4 & 12 & 1 & 0 & 7 & 10 & 9 & 8 & 0 \\
v_7 & 2 & 0 & 11 & 7 & 0 & 10 & 3 & 4 & 5 & 0 \\
v_8 & 8 & 0 & 5 & 12 & 7 & 9 & 4 & 10 & 3 & 2 \\
v_9 & 11 & 1 & 9 & 0 & 4 & 8 & 5 & 3 & 2 & 0 \\
v_{10} & 4 & 5 & 3 & 0 & 1 & 0 & 0 & 2 & 0 & 7
\end{array}
$$

Fig. 4. Coloring Results after Vertex Coloring of Graph *G*

The statistical results of the color sum of each vertex after the vertex coloring is completed are listed in Table 5.

Table 5. Vertex–Edge Color Sum after the Adjustment of Edge Color Sum Conflict in Graph *G*

Vertex	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
Sum of colors	59	38	59	46	35	44	42	60	43	17

Table 5 shows that the total color sum of vertex v_1 is equal to that of v_3, and the total color sum conflict The vertex sum distinguishing total chromatic number $\chi_{\Sigma T}(G) \le \Delta(G)+3$ was obtained by analyzing the collection of random graphs within eight vertexes, and the coloring results are listed in Table 7.

adjustment is implemented according to the related algorithm.

Step 5) Adjustment of vertex total color sum conflict

The vertex coloring remains unchanged in the adjustment of total color sum conflict, and all the exchange processes are performed on the precondition that any edge coloring conflict does not occur, given that no new coloring conflict exists between adjacent vertexes. When the vertex total color sum conflict is handled, the vertex sum-distinguishing total coloring is completed.

The total color sum conflict is adjusted through Algorithm 3. In the living example, the following adjustments are involved in the vertex total color sum conflict adjustment. $v_1 \leftrightarrow v_2$ is adjusted, and the common color 12 of the complements is exchanged; $v_4 \leftrightarrow v_5$ is adjusted, and the common color 11 of the complements is exchanged; $v_5 \leftrightarrow v_9$ adjust, and the common color 12 of the complements is exchanged; the post-adjustment coloring results are shown in Fig.5 and Table 6.

$$
\begin{array}{c|cccccccccc}
 & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 & v_9 & v_{10} \\
\hline
v_1 & 5 & 12 & 1 & 9 & 10 & 3 & 2 & 8 & 11 & 4 \\
v_2 & 12 & 9 & 10 & 3 & 0 & 4 & 0 & 0 & 1 & 5 \\
v_3 & 1 & 10 & 6 & 4 & 8 & 12 & 11 & 5 & 9 & 3 \\
v_4 & 9 & 3 & 4 & 8 & 11 & 1 & 7 & 12 & 0 & 0 \\
v_5 & 10 & 0 & 8 & 11 & 3 & 0 & 0 & 7 & 12 & 1 \\
v_6 & 3 & 4 & 12 & 1 & 0 & 7 & 10 & 9 & 8 & 0 \\
v_7 & 2 & 0 & 11 & 7 & 0 & 10 & 3 & 4 & 5 & 0 \\
v_8 & 8 & 0 & 5 & 12 & 7 & 9 & 4 & 10 & 3 & 2 \\
v_9 & 11 & 1 & 9 & 0 & 12 & 8 & 5 & 3 & 2 & 0 \\
v_{10} & 4 & 5 & 3 & 0 & 1 & 0 & 0 & 2 & 0 & 7
\end{array}
$$

Fig. 5. Vertex Sum Distinguishing Total Coloring Results of Graph *G*

Table 6. Statistical Table of Vertex Color Sum after Total Color Sum Conflict Adjustment in Graph *G*

Vertex	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
Sum of colors	65	44	69	46	55	54	42	60	51	22

Step 6) In this case, $F_{\sum T}=F_1+F_2+F_3+F_4$, the coloring is completed, and the number of colors used is $color\_n=\mu_t(G)+1$, conforming to the content of the Assertion 10. Therefore, the coloring is successful.

4.2 Experimental results

The algorithm was used to test the relationships among the vertex number, color number, and edge density in some graphs under the environment of VC6.0 Windows 64-bit system with 2 GB memory and 500 G hard disk. The relation maps of vertex number–edge density–color number obtained through the substantive test are as shown in Fig.6 and 7.

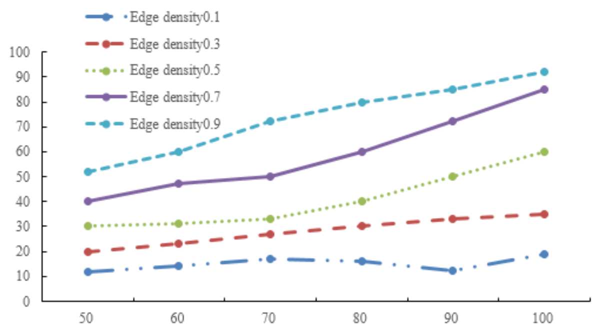

Fig. 6. Vertex number–edge density–color number within 100 vertexes

The vertex sum distinguishing total chromatic number $\chi_{\Sigma T}(G) \le \Delta(G)+3$ was obtained by analyzing the collection of random graphs within eight vertexes, and the coloring results are listed in Table 7.

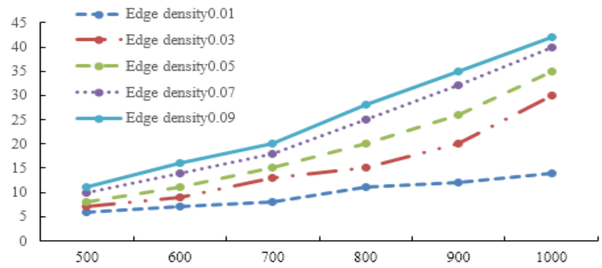

Fig. 7. Vertex number–edge density–color number within 1000 vertexes

Table 7. Vertex Sum Distinguishing Total Coloring Results of Graph G within Eight Vertexes

Vertex	Total number of graph	$\chi_{\Sigma T}(G) \le \Delta(G)+3$
3	2	2
4	6	6
5	21	21
6	113	113
7	853	853
8	11117	11117

4.3 Algorithm analysis

4.3.1 Algorithm accuracy

This algorithm decomposed the vertex sum distinguishing total coloring problem into multiple subproblems, for each of which an objective function was set. Subsequently, each problem was solved successively according to the corresponding rules. First, the edges in the graph were pre-colored randomly. Second, whether any conflict existed in the edge coloring, *i.e*., whether colored edges with the same color exists, was checked. If yes, then the color iteration and exchange should be performed according to the rules specified by the algorithm until the edge coloring satisfied the requirements, namely, F_1=0. Subsequently, whether the color sum sets of adjacent vertexes were equal was judged. If yes, then a conflict existed. Subsequently, it was handled by invoking the vertex–edge color sum conflict adjustment algorithm until F_3=0. The value range of vertex coloring was the color complement of the present vertex. If no colors were usable, then exchange operation was performed on the preconditions of F_1=0and F_3=0 until F_2=0 . Finally, the vertex total color sum conflict was adjusted. If the color set complement of the present vertex was the same and non-empty, then the present color set complement was reduced into the complement without the addition of the colors given to other adjacent vertexes, followed by the exchange operation. During the adjustment process, the vertex–edge coloring conflict would not be caused even if the vertex coloring was unchanged, and finally F_4=0 was ensured. On this basis, each subobjective function satisfied the requirements, and the vertex sum distinguishing total coloring was completed. This algorithm achieved the total objective $F_{\Sigma T}=F_1+F_2+F_3+F_4$ by gradually optimizing all subobjectives; thus, this algorithm was accurate.

4.3.2 Algorithm convergence

All the colors used by the algorithm to handle the coloring conflict were obtained from the color set complement *comset*[][], ensuring that the value of F_3 did not increase. The colors used in the vertex coloring were also obtained from *comset*[][]. Moreover, F_2 and F_3 did not increase because unchanging the values of F_1 and F_3 was considered the precondition in the vertex conflict adjustment. The vertex–edge sum conflict was adjusted when the values of F_1, F_2, and F_3 are unchanged. Therefore, the total objective function presented monotonic non-increasing trend in the algorithm execution process.

Among the four constraint functions, the value of edge constraint function, vertex constraint function, vertex–edge constraint function, and vertex sum constraint function was $0 \le F_1 \le (n-1)(n-2)/2$, $0 \le F_2 \le n(n-1)/2$, $F_3=0$ and $0 \le F_4 \le n(n-1)/2$,respectively.Thus, $0 \le F_{\Sigma T}=F_1+F_2+F_3+F_4 \le (n-1)(3n-2)/2$. Therefore, limits existed in the objective function. Meanwhile, as the function was monotonic, the algorithm was convergent.

4.3.3 Time complexity of the algorithm

According to the algorithm description, the algorithm mainly had the following steps:

a) Generating an $n*n$ adjacency matrix, with the time complexity of $T_1(n)=\sum_{i=1}^{n}\sum_{j=1}^{n} Random = O(n^2)$.

b) Edge precoloring. The worst time complexity of this step is equal to the time complexity of edge precoloring when G is a complete graph with n vertexes, i.e., $T_2(n)=\sum_{i=1}^{n}\sum_{j=1}^{n} pre\_c = O(n^2)$.

c) Handling adjacent vertex–edge coloring conflicts. For graph G containing n vertexes, only the conflict set is adjusted in the algorithm, and the worst time complexity is $T_3(n)=\sum^{\max}\sum_{i=1}^{n}\sum_{j=1}^{n} exchange = O(n^3)$.

d) Adjusting color sum conflicts. When G is a complete graph and the color sums of all vertexes are equal, the number of conflicts is the greatest. Each vertex should be comparatively judged with the other n-1 vertexes. In each adjustment, the colors of the two edges and the color set complements of the two vertexes should be altered; the time complexity is $T_4(n)=O(n^2)$.

e) Vertex coloring. The vertex is colored by selecting a color from the color set complement, with the time complexity of $T_5(n)=O(n)$.

Through a comparative analysis, the algorithm operating time mainly depends on the graph G as well as the vertex number n and edge number m. In the worst case, the time complexity of this algorithm is $T(n)=O(n^3)$.

5 Conclusion

The vertex sum distinguishing total coloring is relatively complicated in graph coloring. Specifically, th.e constraint conditions are added on the basis of the vertex distinguishing total coloring. The proposed vertex sum distinguishing total coloring algorithm performed iterative exchange and gradual optimization by setting multiple subobjective functions and using color set complements based on the research idea of multiobjective optimization. Thus, the total objective function reached the optimal value, and the global optimum was achieved from the local optimum. The following conclusions are drawn:

(1) The upper bound of vertex sum distinguishing total chromatic number $\Delta(G)+3$ is obtained by analyzing the constraint conditions for the vertex sum distinguishing the total coloring of the graph.

(2) The coloring efficiency can be improved through color sum conflict adjustment by means of pre-exchange. The color sum vertexes that satisfy the conditions are pre-exchanged, and the color sum conflict pairs are calculated. Subsequently, the locally optimal exchange is realized through the exchange following the idea of impressed variation.

(3) The color sets are differentiated using the color set

complements of each vertex. The number of color set complements should be convenient for generating a sorted collection to handle the color conflicts. In case of any conflict, heuristic coloring can be implemented by taking elements from the color set complements, thereby improving the coloring efficiency.

The proposed algorithm was specific to the characteristics of vertex sum distinguishing total coloring of graphs. According to the constraint conditions, the multiobjective constraint subfunctions were generated, the vertex coloring was conducted based on the normal edge coloring and the vertex sum distinguishing edge coloring, and then the coloring of random graphs was completed within a short time. The constraint conditions of the algorithm can be reduced to obtain the other coloring results of random graphs, such as adjacent vertex sum-distinguishing edge coloring, adjacent vertex sum distinguishing total coloring and vertex sum distinguishing edge coloring. However, the algorithm execution efficiency is low in the case of a large-scale graph. Therefore, the large-scale graph can be segmented into small-scale subgraphs through the graph segmentation algorithm, or the algorithm execution efficiency can be improved through the multithreading parallel computation to allow the algorithm to realize accurate coloring within shorter time.

Acknowledgements

The study was supported by the Science and Technology Research Project of Henan Province (Grant No. 142102210554).

This is an Open Access article distributed under the terms of the Creative Commons Attribution License.

References

1. Han, G. L., Du, W., “The application research of parking space allocation based on graph coloring model”. *Telecom Express*, (7), 2019, pp.27-32.
2. Liu, W., Liao, R. B., “PCI automatic allocation method based on improved graph cololring algorithm”. *Computer engineering and design*, 39(2), 2018, pp.352-357.
3. Pratap, A., Misra, R., Gupta, U., “Randomized Graph Coloring Algorithm for Physical Cell ID Assignment in LTE-A Femtocellular Networks”. *Wireless Personal Communications*, 91(3), 2016, pp,1-23.
4. Oylum, S., Tinaz, E., Caner, T. Z., “A decomposition approach to solve the selective graph coloring problem in some perfect graph families”. *Networks*, 73(2), 2019, pp,145-169.
5. Ren, Y. L., Wang, L., Liang, D. Y., ”Research on Railway Shunting Problem Based on Graph Coloring”. *Advances in Applied Mathematics,* 10(2), 2021, pp.402-415.
6. Xu, F. M., Qian, H., Cao, H. Y., ”Seni-dynamic pilot allocation based on graph colorng in massive MIMO systems”. *Telecommunications Science*, 36(11), 2020, pp.39-46.
7. Balister, N., Riordan, M., Schelp, R.H.,”Vertex-distinguishing ProperEdge- Colorings”, *J.GraphTheory,* 42(2), 2003, pp.95-109.
8. Bo, Y. H., Wang, L. X., “2-distance Coloring of Sparse Planar Garphs”, *Advances in Mathematics*, 48(2), 2019, pp.145-155.
9. Dong, X. Y., Ma, D. J., “Strong edge coloring of Goldberg snark”. *Journal of Northeast Normal University (Natural Science Edition)*, 50(4), 2018, pp.16-19.
10. Li, T., Chen, X. E., Wang, Z. W., “General Vertex-Distinguishing Total coloring of Corona graph of 2K2VK1”. *Journal of Jilin University(Science Edetion)*, 57(3), 2019, pp.544-552.
11. Chen, X. E., Zhang, S., Gu, Z. P., “Vertex distinguishing IE-total coloring of K2,4,p”. *Journal of Electronics & Information Technology*,42(12),2020,pp.2999-3004.
12. Flandrin, E., Marczyk, A., Przybylo, J., ”Neighbor Sum Distinguishing Index”. *Graphs Combin*, 29(5), 2013, pp.1329-1336.
13. Wei, J., “Application of Graph Coloring Algorithm Based on Ant Colony Optimization in Intelligent Community”. *Journal of Jishou University (Natural Science Edition)*, 40(2), 2019, pp. 27-30,33.
14. Zheng, T. Y., Zhang, H., Zhou, Jiang., M. J., ”D2D Resource Allocation and Power Control Based on Improved Graph Coloring”.*Telecommunication Engineering*, 384(11), 2020, pp.82-87.
15. Ran, C. S., Zhang, T., “Improved coloring algorithm based on ant colony for application in intelligence community”, *Computer Engineering and Design*, 37(1), 2016, pp. 152-153.
16. Li, S. Z., He, S. F., ”Hybrid o timization algorithm based on graph coloring problem”, *Application Research of Computers*, (1), 2016, pp.90-100.
17. Burris, A. C., ”Vertex-Distinguishing Edge-Colorings”,*PhD thesis Memphis State University*, 12(7), 1993, pp. 118 -129.
18. Tian, S. L., Dong, X. F., Liu, R. L., “Adjacent vertex-distinguishing coloring of the semistrong product of graphs”. *Operations Research Transactions*, 21(3), 2017, pp.119-125.
19. Hocquard,H.,Montassier. M., “Adjacent“vertex-distinguishing edge coloring of graphs with maximum degree at least five”,*Electronic Notes in Discrete Mathematics*, 38, 2011, pp.457-462.
20. Hocquard, H., Montassier, M., “Adjacent vertex distinguishing edge coloring of graphs with maximum degree”. *Delta .J.Comb.Optim*, 26(1), 2013, pp.152-160.
21. Caro, Y., Hansberg, A, Montejano, A., “Unavoidable chromatic patterns in 2 - colorings of the complete graph”. *Journal of Graph Theory*, 97 (4), 2020,pp,1-24.
22. Pilsniak, M., Woniak, M., ”On the total-neighbor-distinguishing index by sums”. *Graphs and Combinatorics*, 31(3), 2015, pp.771-782.
23. Dong, A. J., Wang, G. H., ”Neighbor sum distinguishing total colorings of Graphs With Bounded Maximum Average Degree”. *Acta Mathematica Sinica*, 30(4), 2014, pp.703-709.
24. Cui, F. X., Yang, C., Ye, H. B., Yao, B., ”Neighbor full sum distinguishing total coloring of join graphs”. *Journal of Jilin University(Science Edition)*, 59(6), 2021, pp.1-9.
25. Tian, S. L., Yang, H., SuoL, W. Q., Yang, Q., ”Neighbor sum distinguishing edge coloring of the lexicographic product of paths”. *Operations Research Transactions*, 24(1), 2020, pp.140-146.
26. Basmassi, M. A., Benameur, L., Chentoufi, J. A., “ A novel greedy genetic algorithm to solve combinatorial optimization problem”. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Virtual Safranbolu*, Turkey: ISPRS, 2020, pp. 117-120.
27. Sun, W., Hao, J. K., Zang, Y., “A solution-driven multilevel approach for graph coloring”. *Applied Soft Computing*, (2), 2021, pp.107174-107199.
28. Artacho, F., Campoy, R., Elser, V., ”An enhanced formulation for solving graph coloring problems with the Douglas-Rachford algorithm”. *Journal of Global Optimization*, 77(2), 2018, pp. 383-403.
29. Dey, A., Agarwal, A., Dixit, P., “A genetic algorithm for total graph coloring”. .*Journal of Intelligent & Fuzzy Systems,* 37(6), 2019, pp.7831-7838.
30. Furini, F., Gabrel, V., Ternier, I. C., ”An improved DSATUR-based branch and bound algorithm for the vertex coloring problem”. .*Networks*, 69(1), 2017,pp.124-141.
31. Li, J. W., Li, X. H., Dong, W., “Algorithm for vertex distinguishing total coloring of random graphs”. *Application Research of Computers*, 32(6), 2015, pp.1707-1710.

32. Cao, D. T., Li, J. W., Jiang, D. D., "Adjacent vertex-distinguishing equitable V-total coloring algorithm of graph based on multi-objective optimization". *Journal of Computer Application*, 37(2), 2017, pp.457-462.