



# 计算机科学

COMPUTER SCIENCE

## 均衡加权图着色问题与启发式算法

欧开明, 江华

引用本文

欧开明, 江华. 均衡加权图着色问题与启发式算法[J]. 计算机科学, 2024, 51(11A): 231200103-7.

OU Kaiming, JIANG Hua. [Balanced Weighted Graph Coloring Problem and Its Heuristic Algorithms](#)[J].

Computer Science, 2024, 51(11A): 231200103-7.

---

## 相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

**Similar articles recommended (Please use Firefox or IE to view the article)**

[基于Grover算法的图着色问题求解](#)

Solving Graph Coloring Problem Based on Grover Algorithm

计算机科学, 2023, 50(6): 351-357. <https://doi.org/10.11896/jsjcx.220400051>

[图着色问题禁忌搜索改进算法](#)

Improved Algorithm for Tabu Search of Graph Coloring Problems

计算机科学, 2022, 49(11A): 211000128-5. <https://doi.org/10.11896/jsjcx.211000128>

[基于U-net++网络的弱光图像增强方法](#)

Low-light Image Enhancement Method Based on U-net++ Network

计算机科学, 2021, 48(11A): 278-282. <https://doi.org/10.11896/jsjcx.210300111>

[基于弱约束指派的DSP寄存器偶对分配算法研究](#)

Research on DSP Register Pairs Allocation Algorithm with Weak Assigning Constraints

计算机科学, 2021, 48(6A): 587-595. <https://doi.org/10.11896/jsjcx.200600061>

[面向机器学习系统的需求建模与决策选择](#)

Requirements Modeling and Decision-making for Machine Learning Systems

计算机科学, 2020, 47(12): 42-49. <https://doi.org/10.11896/jsjcx.201200021>

# 均衡加权图着色问题与启发式算法

欧开明 江华

云南大学软件学院 昆明 650504

(750778507@qq.com)

**摘要** 给定一个无向图  $G$  和一个颜色数  $k$ , 图的  $k$  着色问题(GCP)指给  $G$  中的每个顶点分配  $k$  种颜色中的一种, 使得任意相邻的两个顶点获得不同的颜色。均衡资源分配是将资源尽可能均匀地分配给各个参与者, 旨在实现资源的公平利用和任务的合理分担。针对传统的图着色问题无法解决均衡资源分配的情况, 提出了图着色问题的一个新变种——均衡加权图着色问题, 其目标是寻找合法的着色使得每种颜色类的权值和的标准差最小。提出了一种将两种局部搜索结合到进化算法中的 HEA-TLS 算法来寻找该问题的最优解。基于新颖性的局部搜索的目的是寻找到一个合法解。改善解均衡性的局部搜索的目的是在合法解的基础上, 改善解的均衡性。进化算法中设计了一个均衡权值交叉, 可以根据父代颜色类权值的变化自适应地选择传递给子代的颜色类, 通过种群的遗传进化来产生更加均衡的着色解。在 DIMACS 图上使用通用求解器 CPLEX 进行对比评估, HEA-TLS 在所有测试中取得了几乎最优的结果, 验证了所提方法的有效性。

**关键词:** 图着色; 均衡加权图着色问题; 局部搜索; 混合进化算法

**中图分类号** TP301

## Balanced Weighted Graph Coloring Problem and Its Heuristic Algorithms

OU Kaiming and JIANG Hua

School of Software, Yunnan University, Kunming 650504, China

**Abstract** Given an undirected graph  $G$  and a number of colors  $k$ , the graph  $k$ -coloring problem(GCP) refers to assigning one of  $k$  colors to each vertex in  $G$  such that any two adjacent vertices receive different colors. Balanced resource allocation is to distribute resources as evenly as possible to each participant, aiming to achieve fair utilization of resources and reasonable sharing of tasks. Aiming at the situation that the traditional graph coloring problem cannot solve the balanced resource allocation, a new variant of the graph coloring problem, the balanced weighted graph coloring problem, is proposed, whose goal is to find a legal coloring that minimizes the standard deviation of the sum of the weights of each color class. A HEA-TLS algorithm that combines two local searches into an evolutionary algorithm is proposed to find an optimal solution to this problem. The novelty-based local search aims to find a legitimate solution. The purpose of the local search for improving the equilibrium of the solution is to improve the equilibrium of the solution based on the legitimate solution. A balanced weight crossover is designed in the evolutionary algorithm, which can adaptively select the color classes to be passed to the offspring according to the changes in the weights of the parent color classes, to produce a more balanced coloring solution through genetic evolution of the population. In a comparative evaluation using the generalized solver CPLEX on the DIMACS graph, HEA-TLS achieves almost optimal results in all tests, validating the effectiveness of the proposed method.

**Keywords** Graph coloring, Balanced weighted graph coloring problem, Local search, Hybrid evolutionary algorithm

## 1 引言

给定无向图  $G=(V, E)$ , 图着色问题是给  $G$  中的每个顶点分配颜色, 使任意两个相邻(即有边连接)的顶点获得不同颜色。使得图能够成功着色的最少颜色数称为图的着色数。给定颜色数  $k$ , 寻找  $G$  的合法着色, 称为  $k$  着色问题。

GCP 问题有多种变种问题, 最常见的一个变体是加权图着色问题(WGCP)<sup>[1]</sup>。给定一个顶点加权的无向图  $G=(V, E, \omega)$  和颜色数  $k$ , 每个顶点的权值由权值函数  $\omega(\cdot)$  定义。WGCP 指给图中的每个顶点分配  $k$  种颜色中的一种, 目标是

最小化每个颜色类的权值之和, 其中颜色类的权值定义为染该颜色的顶点的最大权值。

GCP 问题和 WGCP 问题不仅仅是数学理论上的一个挑战, 它们还在实际应用中发挥着重要作用。为了解决大规模无人机集群中网络资源有限的问题, Gong 等<sup>[2]</sup> 针对无人机收发节点构成的通信网络, 提出了一种基于图着色的三维网络资源分配算法。停机位分配是地面机场系统运作的一个重要任务, Han 等<sup>[3]</sup> 将复杂的机场停机位分配建立图着色模型, 并采用贪心算法对该问题进行求解。为了降低弹性光网络的频谱碎片化程度和阻塞率, Ge 等<sup>[4]</sup> 提出一种基于加权图着色模

基金项目: 国家自然科学基金(62162066)

This work was supported by the National Natural Science Foundation of China(62162066).

通信作者: 江华(huajiang@ynu.edu.cn)

型的弹性光网络频谱分配算法,更好地分配了频谱资源并降低了链路阻塞率。Deogratis 等<sup>[5]</sup>使用图着色模型安排学校的课时表,并设计一种基于图着色的时间表调度算法。

在资源分配和任务调度这一类应用中,除了满足基本的约束条件外,往往还需要一些额外的目标,如公平性和均衡性。例如飞行员排班问题中,每个航班都需要分配给一名飞行员<sup>[6]</sup>。为确保分配的公平性,使得每个飞行员有充足的休息时间,要求分配给每位飞行员的航班时间应尽可能地相等。飞行员排班可以被建模为经典的图着色问题,其中每个飞行员被建模为一种颜色,每个航班被建模为一个顶点,当且仅当两个顶点(航班)由于时间限制而不能被一个飞行员执行时,这两个顶点(航班)才是相邻的。合法的顶点着色表示满足飞行时间限制的分配。飞行员排班任务是确保分配的公平性,而经典的图着色问题是寻找最小染色数  $k$ , 无法实现任务分配的公平性目标,因此并不能直接用于解决该问题。

公平性或均衡性的目标在类似的任务分配的任务中经常出现。经典的图着色模型虽然可以解决一般的资源分配任务,但是并不能解决以均衡性为优化目标的资源分配任务。基于这一观察,本文提出了一种新的图着色问题变体,即均衡加权图着色问题(Balanced Weighted Graph Coloring Problem, BWGCP)。BWGCP 的目标是找到一种可行的着色方案,优化目标为每种颜色类的权值和之间的标准差最小。

针对新提出的均衡加权图着色问题,本文提出了一种结合局部搜索和进化算法的混合进化算法。在 DIMACS 基准图上的实验结果表明,所提出的算法能够找到比通用求解器更高质量的均衡着色解,可以有效求解均衡加权图着色问题。第 2 节给出均衡加权图着色问题的定义和性质;第 3 节介绍 BWGCP 算法的总体框架和主要组成部分;第 4 节为实验及分析;最后总结全文。

## 2 均衡加权图着色问题

给定一个加权图  $G=(V, E, \omega)$ , 其中  $V=\{v_1, \dots, v_n\}$  是顶点集合,  $E=\{e_1, \dots, e_m\}$  是边集合,  $\omega()$  是权值函数。给定一个颜色数  $k$ ,  $G$  的合法着色表示为  $S=\{V_1, \dots, V_k\}$ , 其中每个  $V_i$  中的任意一对顶点都不相邻,  $V=V_1 \cup V_2 \cup \dots \cup V_k$ , 且对于所有  $i \neq j$  满足  $V_i \cap V_j = \emptyset$ 。  $V_i \in S$  在本文中称为一个颜色类。经典的图着色问题是找到能给图  $G$  正确着色的最小颜色数  $k$ 。

给定一个图  $G=(V, E, \omega)$  和一个颜色数  $k$ , 颜色类的平均权值  $\alpha$  定义如下:

$$\alpha = \frac{\sum_{v_i \in V} \omega(v_i)}{k} \quad (1)$$

颜色类  $V_j$  的权值  $\omega(V_j)$  定义如下:

$$\omega(V_j) = \sum_{v_i \in V_j} \omega(v_i) \quad (2)$$

给定  $G$  的一个合法着色  $S=\{V_1, \dots, V_k\}$ ,  $S$  的权值标准差  $F(S)$  定义为:

$$F(S) = \frac{\sum_{V_j \in V} |\omega(V_j) - \alpha|}{k * \alpha} \quad (3)$$

均衡加权图着色问题就是在给定图  $G=(V, E, \omega)$  和颜色数  $k$  的情况下,找到权值标准差最小的合法着色  $S$ 。

为了对后文提出的算法进行更全面的评估,本文将 BWGCP 建模为混合整数规划(MIP),并使用 CPLEX 进行

求解。MIP 计算过程的优化目标如式(4)所示,计算完成后再除以  $k * \alpha$ , 以便与式(3)保持一致。本文将  $x_{i,j}$  定义为 0-1 变量,当顶点  $v_i$  染颜色  $j$  时,  $x_{i,j}$  取 1, 否则取 0。

$$\min \sum_{V_j \in S} \left| \sum_{v_i \in V_j} \omega(v_i) * x_{i,j} - \alpha \right| \quad (4)$$

BWGCP 的约束规则如下:

1) 每个顶点只能有一种颜色:

$$\sum_{V_j \in S} x_{i,j} = 1, \forall v_i \in V \quad (5)$$

2) 相邻顶点不能有相同颜色。对于任意相邻顶点  $(v_m, v_n) \in E$ :

$$x_{m,j} + x_{n,j} \leq 1, \forall V_j \in S \quad (6)$$

BWGCP 的提出受到了任务分配等实际应用的启发。在这些应用中,不仅要正确分配所有任务,还要考虑分配给每个人的任务量的均衡性。本文以飞行员分配问题<sup>[6]</sup>为例说明 BWGCP 的应用。给定一组航班  $L=\{l_1, l_2, \dots, l_n\}$  和一组飞行员  $P=\{p_1, p_2, \dots, p_k\}$ , 每个航班  $l_i$  的飞行时间为  $d(l_i)$ 。飞行员分配问题的目标是将每个航班  $l_i \in L$  分配给一个飞行员  $p_j \in P$ , 并尽可能使每个飞行员获得相等的飞行时长。可行的分配必须满足某些约束条件,例如,一名飞行员不能同时执飞两个航班。

为了解决飞行员分配问题,可以将其建模为均衡加权图着色问题。首先定义一个顶点集  $V=\{v_1, \dots, v_n\}$ , 每一个顶点  $v_i$  表示一个航班  $l_i \in L$ , 边集  $E=\{(v_i, v_j) \mid v_i, v_j \in V\}$  定义为:若航班  $l_i$  和  $l_j$  不能同时交给一名飞行员执飞,则  $(v_i, v_j) \in E$ 。对于每个顶点  $v_i \in V$ , 其权值定义为  $\omega(v_i) = d(l_i)$ 。由此得到一个加权图  $G=(V, E, \omega)$ 。设  $k=|P|$ , 在  $G$  上的一个合法着色,就是一个合法的任务分配, BWGCP 的最优解即是考虑任务分配公平性的最优解。

图的  $k$  着色问题是一个 NP 难问题, BWGCP 的解首先要求是一个合法的 GCP 解,因此 BWGCP 也是 NP 难问题。下一节将介绍针对 BWGCP 的启发式算法。

## 3 求解均衡加权图着色问题的启发式算法

### 3.1 相关算法

由于 GCP 的 NP 难性质,几乎没有高效的精确算法。现有的 GCP 算法都是基于启发式搜索。GCP 的启发式算法一般可分为两类:局部搜索算法和混合算法。局部搜索算法,如禁忌搜索算法<sup>[7-8]</sup>、蚁群算法<sup>[9]</sup>、模拟退火算法<sup>[10]</sup>, 通常能快速找到次优解,但容易陷入局部最优。混合算法,如 Ma-col<sup>[11]</sup>、Memetic<sup>[12]</sup>、量子退火算法<sup>[13-14]</sup>、H2col<sup>[15]</sup>、COMA<sup>[16]</sup> 等,通常将局部搜索算法与其他算法相结合,为 GCP 提供高质量的着色解。

### 3.2 求解 BWGCP 的混合算法

为了寻找 BWGCP 问题的最优解,本文提出一种将两种局部搜索结合到进化算法中的混合算法,称为基于两种局部搜索的混合进化算法(Hybrid Evolutionary Algorithm with Two Local Search, HEA-TLS)。该算法由 5 个部分组成:初始种群生成器,基于新颖性的局部搜索算法,改进解均衡性的局部搜索算法,以及均衡权值交叉和种群更新。

给定一个图  $G=(V, E)$  和一个颜色数  $k$ , 在初始的种群生成阶段,使用随机着色的方法生成  $p$  个初始解  $\{S_1, \dots, S_p\}$ 。初始解通常是不合法的,即解中存在相邻顶点被分配了相同颜色的情况, HEA-TLS 首先使用基于新颖性的局部搜索

算法来迭代改进这个不合法解,直至其合法为止。然后采用改进解均衡性的局部搜索算法降低目标函数  $F(S)$ 。之后对种群使用均衡权值交叉生成一个子代解  $S_0$ ,再通过两个局部搜索后得到一个均衡优化的合法子代  $S_0$ 。种群更新用于判断是否将  $S_0$  插入种群  $p$  以更新种群。算法框架如算法 1 所示。各组成部分的实现详见后文。

### 算法 1 均衡加权进化算法

输入:  $G(V, E), k, \text{population size } p$

输出: The best  $k$ -coloring  $S^*$

1.  $P = \{S_1, \dots, S_p\} \leftarrow \text{Initial Population}()$
2. for  $i=1$  to  $p$  do
3.  $S_i \leftarrow \text{Finding Valid Colorings}() / * 3.3 * /$
4.  $S_i \leftarrow \text{Improving the Balance of Solution}() / * 3.4 * /$
5. end for
6.  $S^* = \text{argmin}\{F(S_i), i=1, \dots, p\}$
7. repeat
8. Randomly choose  $m$  individuals  $\{S_{i1}, \dots, S_{im}\}$  from  $P(2 \leq m \leq p)$
9.  $S_0 \leftarrow \text{BWC}\{S_1, \dots, S_{im}\} / * 3.5 * /$
10.  $S_0 \leftarrow \text{Finding Valid Colorings}() / * 3.3 * /$
11.  $S_0 \leftarrow \text{Improving the Balance of Solution}() / * 3.4 * /$
12. if  $F(S_0) < F(S^*)$  then
13.  $S^* = S_0$
14.  $\{S_1, \dots, S_p\} \leftarrow \text{Pool Updating}\{S_0, S_1, \dots, S_p\} / * 3.6 * /$
15. until Stop condition met

### 3.3 基于新颖性的局部搜索

初始种群  $\{S_1, \dots, S_p\}$  的着色方案通常是不合法的。针对经典寻找合法着色的禁忌搜索算法存在搜索速度慢,不能很好地满足实际应用中时效性的问题,本文提出了一种基于新颖性的局部搜索算法。定义  $move(v, i, j)$  表示顶点  $v$  从颜色  $i$  变为颜色  $j$  的一次移动。移动发生时的迭代次数  $iter$  定义为该移动时间。算法维护一个冲突表  $conflictedgestable$ , 用来保存当前解中的所有冲突边。移动得分表  $movescoretable$ , 用来记录每个移动  $move(v, i, j)$  的得分,其定义为: 顶点  $v$  从当前颜色  $i$  改变为颜色  $j$  后冲突边数量的变化。移动得分为正表示该移动可以减少冲突边的数量。我们将得分正的移动放入下降行动表  $descentactiontable$  中,代表可以减小冲突边的移动。当移动得分为负时,表明该移动会增加冲突边的数量。

算法首先检查下降行动表是否为空。如果不为空,算法会从表中选择移动时间最早(迭代次数  $iter$  最小的)的移动来执行。如果为空,算法会随机选择一条冲突边,以 5% 的概率随机选择这条边的一个端点并改变其颜色,而以 95% 的概率根据以下选择策略选择一个移动: 算法在所选边的两个顶点的所有可行移动中,根据得分找出一个最佳移动和一个次佳移动。最佳移动和次佳移动是指移动得分最高的两个移动,在得分相同时选移动时间较早的,以减小搜索陷入局部最优的可能性。如果最佳移动的移动时间不是所有移动中最晚的,则选择该移动执行。如果最佳移动是最晚发生的移动,为了避免算法陷入局部最优,以 45% 的概率选最佳移动,以 55% 的概率选择次佳移动。当迭代次数超过最大迭代次数且着色解仍不合法时,算法返回 null,表示未找到合法着色解。上述被称为“新颖性”的移动选择策略最初是针对布尔可满足

性问题的局部搜索提出的<sup>[17]</sup>。基于新颖性的局部搜索如算法 2 所示。

### 算法 2 基于新颖性的局部搜索

输入: An initial solution  $S, k, \text{max\_iter}$

输出: A feasible solution  $S^*$

1.  $S^* = S, \text{iter} = 1$
2. Initialize conflict edges table and move score table and descent actions table
3. Let  $cf(S)$  be the number of conflicting edges of  $S$
4. while  $cf(S) > 0$  and  $\text{iter} < \text{max\_iter}$  do
5. if descent actions table  $\neq \emptyset$  Then
6. Execute  $move(v, i, j)$  with the earliest time from descent actions table
7. else
8. Randomly select a conflicting edge
9. if  $\text{rand}(0, 100) < 5$  Then
10. Randomly change the color of one vertex of edge
11. else
12. Determine two best moves from all moves of the edge
13. if most recent move  $\neq$  best move Then
14. Execute the best  $move(v, i, j)$
15. else if  $\text{rand}(0, 100) < 45$  Then
16. Execute the best  $move(v1, i, j)$
17. else
18. Execute the second best  $move(v2, m, n)$
19. Update conflict edges table and move score table and descent actions table
20. if  $cf(S) < cf(S^*)$  Then
21.  $S^* = S$
22.  $\text{iter} = \text{iter} + 1$
23. end while
24. if  $(cf(S) = 0)$
25. return  $S^*$
26. else
27. return null

### 3.4 改进解均衡性的局部搜索

在基于新颖性的局部搜索算法获得一个合法着色解后,本文设计了一个改善解均衡性的局部搜索算法。算法在合法着色的基础上,在不改变解合法性的情况下对这个着色解进行均衡优化操作,使得分配的颜色类权值和均衡。在这里,移动得分  $move\_score(v, i, j)$  定义为发生  $move(v, i, j)$  前后  $F(S)$  的差值。假设当前解为  $S, S'$  为执行  $move(v, i, j)$  后的解,移动得分的计算如式(7)所示。移动得分为负,表示该移动可以降低  $F(S)$ 。

$$move\_score(v, i, j) = F(S') - F(S) \quad (7)$$

算法首先查找当前解中有无合法移动。遍历所有可行的移动,寻找得分最低(小于 0)且不在禁忌期的合法移动去执行。发生移动后不会产生新的冲突边的移动称为合法移动。禁忌期表示一旦执行了  $move(v, i, j)$ , 顶点  $v$  在接下来的  $l$  次迭代中禁止移回颜色  $i$ , 本文中  $l$  参考文献[16]设定为  $l = \text{iter} + \text{rand}(1, 7)$ 。

如果不存在这样的移动,算法会寻找交换得分最低(小于 0)且不在禁忌期的合法交换移动去执行。假设  $v_m$  的颜色是  $i, v_n$  的颜色是  $j, (v_m, v_n) \in E$ 。如果执行  $move(m, i, j)$  和  $move(n, j, i)$  后,  $S$  仍是合法解,则称这两个移动为合法交换移动。

交换移动得分  $exchange\_score(m, n, i, j)$  的计算如式(8)所示。

$$exchange\_score(m, n, i, j) = F(S') - F(S) \quad (8)$$

如果上述两种移动都不存在,则随机执行一个合法移动  $move(v, i, j)$ ,并更新禁忌期  $tabu$  和  $F(S)$ 。如果  $F(S)$  小于最优解  $F(S^*)$ ,则将  $F(S^*)$  更新为  $F(S)$ ,同时将当前步数  $current\_steps$  置零;否则将当前步数加 1 并继续循环。当没有合法移动或当前步数大于最大步数  $max\_step$  时,算法结束。改进解均衡性的局部搜索如算法 3 所示。

**算法 3** 改进解均衡性的局部搜索

输入: A feasible solution S, Max\_Step, iter

输出: A balanced optimized feasible  $S^*$

1. current\_step=0,  $S^* = S$
2. Initialize  $tabu[v][c]$  to 0 for all v and c
3. while current\_step < Max\_Step do
4. Iterate over all feasible moves;
5. if  $\exists$  a legal move(v,i,j) && move\_score < 0 &&  $tabu[v][j] < iter$  Then
6. Execute a move(v,i,j) with the lowest score
7. else if  $\exists$  a legal exchange move(m,n,i,j) && score < 0 &&  $tabu[m][j] < iter, tabu[n][i] < iter$  Then
8. Execute a move(m,n,i,j) with the lowest score
9. else if  $\exists$  legal move
10. Randomly execute a legal move
11. else
12. return  $S^*$
13. Update F(S) and tabu
14. if  $F(S) < F(S^*)$  Then
15.  $F(S^*) = F(S), current\_step = 0$
16. else
17. current\_step++, iter++
18. end while
19. return  $S^*$

**3.5 均衡权值交叉**

初始种群  $\{S_1, \dots, S_p\}$  中的个体,在经过两个局部搜索算法后引入到进化算法中,让种群通过交叉遗传的方式进化,以产生更加均衡的后代  $S_0$ 。本文使用的均衡权值交叉(Balanced Weight Crossover, BWC)是 Macol<sup>[11]</sup> 中提出的 AMPaX 交叉的改进版。BWC 使用多个父代生成子代,在每一步

交叉操作中,它都会根据父代颜色类权值的变化,自适应地选择传递给子代的父代颜色类。BWC 与 AMPaX 的主要区别在于:AMPaX 的目标是找到获得合法着色所需的最少颜色数  $k$ ,因此在交叉过程中它更倾向于将合法颜色类中顶点较多的颜色类传递给子代;而 BWC 的目标是改善解的均衡性,因此在交叉过程中优先将权值接近平均值  $\alpha$  的颜色类传给子代  $S_0$ 。

算法首先从种群中随机选择  $m$  个个体作为父代,父代  $S_i$  定义为  $S_i = \{V_{i1}, \dots, V_{ik}\}$ ,  $V_{ij}$  表示父代  $i$  的第  $j$  个颜色类。BWC 产生的子代  $S_0$  定义为  $S_0 = \{V_{01}, \dots, V_{0k}\}$ 。在交叉过程中,权值接近  $\alpha$  的颜色类被认为是优质颜色类,  $S_0$  中这种颜色类越多,则  $F(S)$  越小。如算法 4 第 3 行, BWC 更倾向于将这类颜色进化给后代。BWC 会逐步选取父代中的颜色类去构建  $S_0$ , 当一个颜色类构建完后,该颜色类的顶点将从所有父代中移除。为了使子代更加多样化,避免从单一父代上选取颜色,为父代的选择设置了一个禁忌期  $t$ , 定义为  $t = m/2$ , 表示在选择父代的一个颜色类构建子代的颜色后,在  $t$  步内不能选择该父代中的颜色类。当  $S_0$  被分配了  $k$  个颜色类后,顶点集  $V$  中未分配颜色的顶点会随机分配一个颜色。均衡权值交叉如算法 4 所示。

**算法 4** 均衡权值交叉

输入: m parent individuals  $\{S_1, \dots, S_m\}$  ( $m \geq 2$ )

输出: An infeasible solution  $S_0 = \{V_{01}, \dots, V_{0k}\}$

1. Initialize  $t(S_j)$  to 0 for all  $j, j = \{1, \dots, m\}$
2. for  $i = 1$  to  $k$  do
3. Select a parent  $S_{best}$  from  $\{S_1, \dots, S_m\}$  with color class  $V^*$  closest to  $\alpha$  &&  $t(S_{best}) = 0$
4.  $V_{0i} \leftarrow$  vertices in color class  $V^*$
5. Remove vertices in  $V^*$  from  $\{S_1, \dots, S_m\}$
6.  $t(S_{best}) \leftarrow m/2, t(S_j) = t(S_j) - 1, j = \{1, \dots, m\} \setminus \{best\}$
7. end for
8. Randomly assign a color class for each unassigned vertex in  $S_0$

图 1 给出了均衡权值交叉的过程。示例包括 4 个父代  $\{S_1, \dots, S_4\}$  和 9 个顶点  $\{v_1, \dots, v_9\}, k = 3$ 。首先选择权值最接近  $\alpha$  的  $V_{12}$  作为子代的第一个颜色类,然后从所有父代中删除  $V_{12}$  中的顶点  $\{v_4, v_5, v_6\}$ ,之后依次选择  $V_{33}$  和  $V_{21}$  作为  $S_0$  的颜色类。经过  $k$  次后,给剩余顶点  $v_1$  随机分配一个颜色。

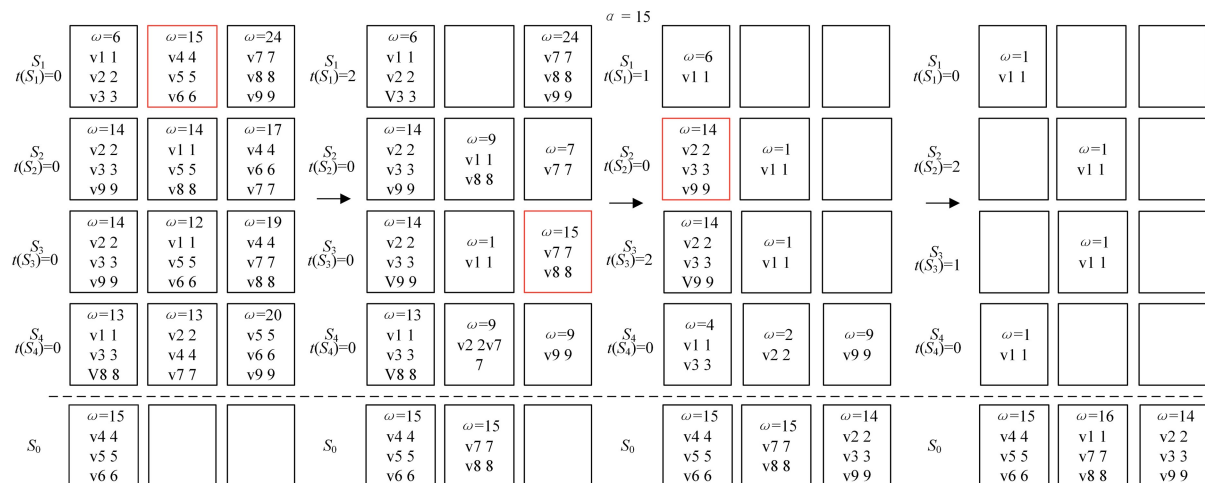


图 1 均衡权值交叉

Fig. 1 Balanced weight crossover

### 3.6 种群更新

BWC得到的不合法子代 $S_0$ ,再次经过两个局部搜索算法变成一个均衡的合法着色后,计算 $F(S_0)$ 。 $F(S_w)$ 表示种群 $p$ 中 $F(S)$ 的最大值,如果 $F(S_0) < F(S_w)$ ,则 $S_0$ 直接取代 $S_w$ 。当 $F(S_0) > F(S_w)$ ,即 $S_0$ 的表现不如最差的父代中时,以20%的概率让 $S_0$ 替换 $S_w$ ,来增加种群的多样性<sup>[11]</sup>。

## 4 实验结果

### 4.1 数据集介绍

本文用64张DIMACS基准图进行实验测试。为了评估本文提出的HEA-TLS算法求解BWGCP的有效性,对该数据集进行了预处理操作,目的是使得解空间中存在一个均衡着色解。具体如下:给定图 $G=(V,E)$ 和颜色数 $k$ ,首先使用基于新颖性的局部搜索算法寻找一个合法着色解,并为每个顶点随机分配1到100之间的权值,之后调整顶点权值的使得 $F(S)=0$ 。本文主要针对均衡加权图着色问题,而不是寻找最小染色数 $k$ ,如果按照最小着色数,则基本不存在均衡权值的空间。因此文中对颜色数进行了适当放宽,在大多数图颜色数增大10,部分较难的图增大了15。

### 4.2 实验设置

实验环境为:Linux系统,Intel(R) Xeon(R) CPU E5-2680 v4@2.40GHz,算法均采用Java编写。实验一是在未预处理的DIMACS图中,评估基于新颖性的局部搜索算法解决

传统图着色问题的能力,使用禁忌搜索作为对比算法,着色数 $k$ 采用Wang<sup>[18]</sup>的设定。每组实验执行算法20次,把成功率、平均迭代时间、平均迭代次数整理成表。

实验二是在预处理后的DIMACS基准图中,使用式(3)的 $F(S)$ 作为评估指标,评估改进解均衡性的局部搜索算法和HEA-TLS在BWGCP中的能力。由于传统图着色问题的优化目标是寻找到合法着色所需的最小染色数 $k$ ,而不是寻找均衡权值的着色解,因此3.1节求解图着色问题的相关算法不适合作为对比。为了对本文的算法进行更全面的评估,将BWGCP问题转化为MIP,使用CPLEX进行求解比较,求解器版本为20.1.0.0。

参考Maco<sup>[11]</sup>,HEA-TLS的种群规模 $p$ 设为20,父代个数 $m$ 设为4,调整这两个参数未发现对结果有明显影响。实验二中,运行4种算法各4次,每次2h。基于新颖性的算法和改善解均衡性的局部搜索算法执行一次后,计算 $F(S)$ 并更新最优解 $F(S^*)$ ,重复这一过程直到时间结束。在表中,符号“-”表示没找到合法解,表中加粗的字体表示算法找到的最优解 $\text{Min } F(S)$ 最小。

### 4.3 实验一结果分析

表1列出了禁忌搜索算法和基于新颖性的局部搜索算法求解传统图着色问题时的部分实验结果。其中, $k$ 表示颜色数; $Succ$ 表示成功率,反映了算法的健壮性; $Time$ 表示执行成功的算例花费的平均时间; $Iteration$ 是执行成功算例的平均迭代次数,次数越小算法效率越高。

表1 基于新颖性的局部搜索算法结果对比  
Table 1 Results comparison of novelty-based local search algorithms

Instance	$k$	禁忌搜索算法			基于新颖性的局部搜索算法		
		$Succ$	$Time/s$	$Iteration$	$Succ$	$Time/s$	$Iteration$
5-FullIns_4	9	70	4.4	$8.09 \times 10^5$	100	0.01	$1.09 \times 10^3$
abb313GPIA	9	20	110	$1.04 \times 10^7$	100	0.28	$7.71 \times 10^4$
flat1000_60_0	60	100	1912	$2.40 \times 10^6$	100	51	$2.77 \times 10^6$
flat1000_76_0	89	25	483	$1.18 \times 10^7$	90	73	$6.47 \times 10^6$
DSJC250.5	28	100	22	$2.04 \times 10^6$	90	23	$6.70 \times 10^6$
DSJC250.9	72	100	10	$5.41 \times 10^5$	100	1.92	$2.27 \times 10^5$
DSJC500.5	49	20	237	$1.10 \times 10^7$	5	26	$4.33 \times 10^6$
DSJC500.9	127	100	193	$4.85 \times 10^6$	95	96	$7.57 \times 10^6$
DSJC1000.1	21	95	31	$2.95 \times 10^{56}$	100	0.47	$9.37 \times 10^4$
DSJC1000.5	90	15	442	$1.12 \times 10^7$	90	87	$6.97 \times 10^6$
DSJC1000.9	229	100	558	$5.81 \times 10^6$	100	113	$5.25 \times 10^6$
le450_15c	15	—	—	—	90	11	$4.81 \times 10^6$
le450_15d	15	—	—	—	20	17	$6.94 \times 10^6$
le450_25c	26	90	2.04	$2.46 \times 10^5$	100	0.20	$5.55 \times 10^4$
DSJR500.5	133	60	8.58	$2.99 \times 10^5$	100	0.14	$7.26 \times 10^3$
fpsol2.i.3	75	—	—	—	100	0	$1.44 \times 10^2$
inithx.i.2	77	—	—	—	100	0.01	$2.59 \times 10^2$
inithx.i.3	77	—	—	—	100	0.01	$2.39 \times 10^2$
miles750	31	70	0.03	$6.10 \times 10^3$	100	0.03	$1.14 \times 10^3$
miles1000	42	65	0.05	$6.16 \times 10^3$	100	0.02	$1.17 \times 10^3$
miles1500	73	100	0.02	$1.20 \times 10^3$	100	0.01	$5.85 \times 10^2$
mulsol.i.4	37	—	—	—	100	0	$2.93 \times 10^2$
mulsol.i.5	37	—	—	—	100	0	$2.91 \times 10^2$
r125.5	36	10	5.17	$8.90 \times 10^5$	100	0.10	$1.54 \times 10^4$
r250.1c	66	—	—	—	100	0.02	$1.463 \times 10^3$
r250.5	67	—	—	—	100	0.31	$5.11 \times 10^4$
r1000.1c	98	—	—	—	100	12	$5.68 \times 10^5$
r1000.5	249	—	—	—	90	177	$9.90 \times 10^6$
wap03a	45	—	—	—	90	39	$7.77 \times 10^6$
wap08a	42	5	449	$1.99 \times 10^7$	100	25	$5.63 \times 10^6$
zeroin.i.3	39	—	—	—	100	0	$1.84 \times 10^2$

首先观察成功率,除了 DSJC250.5 和 DSJC500.5 外,基于新颖性的局部搜索算法的成功率均大于禁忌搜索算法。在 le450\_15c 等 22 张图中,新颖性的局部搜索算法找到了禁忌搜索算法未找到的合法解,且成功率基本为 100%。然后观察求解时间,除 DSJC250.5 等 5 张图中,新颖性局部搜索算法的求解时间略大于禁忌搜索算法外,其余所有图中基于新颖性的局部搜索算法的求解时间均远小于禁忌搜索算法。最后观察迭代次数,除了 flat1000\_60\_0 等 8 张图外,新颖性局部搜索算法的迭代次数均远小于禁忌搜索算法。

实验结果表明,引入了新颖性策略的局部搜索算法表现出了比较好的性能,相对于传统的禁忌搜索算法,其算法健壮性更强,求解速度更快,算法效率更高。所以在改善解均衡性

的局部搜索和 HEA-TLS 中,采用基于新颖性的局部搜索算法来寻找合法着色解。

#### 4.4 实验二结果分析

##### 4.4.1 改善解均衡性的局部搜索算法结果分析

表 2 是 4 种算法在预处理后的 DIMACS 图中求解 BWGCP 问题的部分性能统计表。可以看出,改善解均衡性的局部搜索算法的目标函数值远小于基于新颖性的局部搜索算法,说明该算法有效改进了合法解的均衡性。在 r1000.5 图中,改善解均衡性的局部搜索算法的  $MinF(S)=0.0109$ ,比基于新颖性的局部搜索算法的  $MinF(S)=0.2259$  小了一个数量级,算法的求解时间增大为 130s,说明算法在改进着色的均衡性上花费了较多的时间,但仍在可接受的时间内。

表 2 DIMACS 数据集中算法求解质量统计

Table 2 Statistics of algorithmic solution quality in DIMACS dataset

Instance	$k$	基于新颖性的局部搜索		改善解均衡性的局部搜索		HEA-TLS	CPLEX
		$Min F(S)$	time/ms	$Min F(S)$	time/ms	$Min F(S)$	$Min F(S)$
5-FullIns_4	19	0.1943	14	<b>0.0000</b>	76	<b>0.0000</b>	<b>0.0000</b>
abb313GPIA	19	0.0401	113	0.0000	<b>196</b>	0.0000	0.0749
flat1000_60_0	92	0.1133	1764	0.1069	<b>2044</b>	0.0995	—
flat1000_76_0	99	0.1157	<b>434</b>	0.0766	706	0.0784	—
DSJC250.5	38	0.1071	5	0.0392	17	<b>0.0263</b>	—
DSJC250.9	82	<b>0.1837</b>	11	0.1527	82	0.1279	—
DSJC500.5	59	0.1062	45	0.0718	103	<b>0.0569</b>	—
DSJC500.9	137	0.1767	145	0.1719	258	<b>0.1521</b>	—
DSJC1000.1	31	0.0563	68	0.0011	31	<b>0.0004</b>	0.1304
DSJC1000.5	100	0.1246	437	0.0953	706	<b>0.0844</b>	—
DSJC1000.9	240	0.1921	2903	0.1858	3808	<b>0.1764</b>	—
le450_15c	25	0.0631	8	0.0043	23	<b>0.0023</b>	0.0132
le450_15d	25	0.0474	12	0.0041	24	<b>0.0029</b>	0.0173
le450_25c	35	0.1009	13	0.0031	34	<b>0.0020</b>	0.0180
le450_25d	35	0.0934	10	0.0032	36	<b>0.0022</b>	—
DSJR500.5	143	0.2062	69	0.0085	2744	<b>0.0079</b>	—
fpsol2.i.3	85	0.2968	2	0.0062	83	0.0038	<b>0.0036</b>
inithx.i.2	87	0.2588	6	0.0032	258	<b>0.0021</b>	0.0112
inithx.i.3	87	0.2815	6	0.0040	182	<b>0.0023</b>	<b>0.0023</b>
miles750	41	0.2012	0	0.0188	2	<b>0.0102</b>	<b>0.0102</b>
miles1000	52	0.2510	0	0.0400	3	<b>0.0259</b>	0.0342
miles1500	83	0.2073	1	0.0212	4	0.0108	<b>0.0000</b>
multsol.i.4	47	0.2096	0	0.0068	24	<b>0.0036</b>	0.0104
multsol.i.5	47	0.2123	1	0.0068	22	<b>0.0042</b>	0.0194
r125.5	46	0.1838	0	0.0088	23	<b>0.0075</b>	0.0170
r250.1c	76	0.2070	6	0.0691	20	<b>0.0246</b>	—
r250.5	77	0.1824	24	0.0106	178	<b>0.0094</b>	0.1922
r1000.1c	108	0.2045	673	0.0818	565	<b>0.0650</b>	—
r1000.5	264	0.2259	316	0.0109	130060	<b>0.0098</b>	—
wap03a	55	0.1060	2646	<b>0.0000</b>	3349	0.0000	—
wap08a	52	0.0972	0	0.0002	721	<b>0.0001</b>	0.1634
zeroin.i.3	49	0.2250	0	0.0065	21	<b>0.0040</b>	0.0080

在 64 张 DIMACS 图中,改善解均衡性的局部搜索算法在 42 张图中取得了比 CPLEX 更均衡的着色解,在 5-FullIns\_4 等 10 张图中两种算法求解质量相等。

在 fpsol2.i.3 等 12 张简单图中,CPLEX 取得了更好的结果,但是对于 DSJC1000.9 等 18 张复杂图,CPLEX 在给定的时间内未找到合法着色解,表明了 CPLEX 在处理高复杂度问题时的局限性。值得注意的是,即使在 inithx.i.2 这种 CPLEX 的求解质量更高的图中,两者的差距也仅为 0.0017。在 DIMACS 上的实验表明,CPLEX 在简单图中的求解质量较高。但在复杂图中,由于寻找合法着色解较为困难,因此其均衡权值的效果并不理想。无论是简单图还是复杂图,改善解均衡性的局部搜索算法基本都能找到一个比 CPLEX 更

高质量或者相近质量的均衡着色解,表明该算法可以有效地求解不同规模的均衡图着色问题。

##### 4.4.2 HEA-TLS 算法结果分析

如表 2 所列,HEA-TLS 和 CPLEX 在类似 5-FullIns\_4 的简单图中,都求出  $MinF(S)=0$  的最优解。CPLEX 在 fpsol2.i.3 等 7 张简单图中表现更加出色,而 HEA-TLS 的求解结果与其非常接近,例如 fpsol2.i.3 中两者仅相差 0.0002。进一步分析 DSJC125.5 这类中等难度的图。在这些图中 CPLEX 的性能均不如 HEA-TLS,表明 HEA-TLS 在应对中等规模的均衡加权图着色问题时具有更好的求解能力。当面对 JC1000.9 等 18 张复杂图时,CPLEX 未能求出合法的着色解。

对比 CPLEX, HEA-TLS 在 44 张图中均取得了更好的着色解,展现了它在处理中高复杂度均衡加权图着色问题时的强大能力和较高鲁棒性。通过对比分析,可以看出 HEA-TLS 在不同复杂度的 DIMACS 基准图上具有广泛的适用性,都能提供高质量的着色解,尤其是在处理复杂图时,相比 CPLEX 展现出了更加出色的性能。

接着分析 HEA-TLS 与改善解均衡性的局部搜索算法,在处理如 5-FullIns\_4 这样的简单图时,这两种算法均能够找到最优解,显示出它们在处理低复杂度问题上的有效性。除了在 flat1000\_76\_0 等 4 张图中稍显不足外,HEA-TLS 在 48 张图中找到了比改善解均衡性的局部搜索算法更好的着色解,特别是在 r1000.1c 这类图中,HEA-TLS 成功将目标函数值从 0.0818 优化至 0.0650,体现了其在处理复杂问题时的均衡优化能力。

DIMACS 基准图的实验结果表明,HEA-TLS 在提高着色解的均衡性方面具有的显著优势。该算法利用遗传进化机制,通过引入的均衡交叉操作有效地改进了颜色类的均衡性,从而产生了更优质且均衡的子代解。与改善解均衡性的局部搜索算法相比,HEA-TLS 在着色解的均衡性上有了明显的提升。

**结束语** 本文提出了均衡加权图着色问题,并设计了一种混合算法 HEA-TLS 来解决该问题。该算法结合局部搜索和进化算法来寻找该问题的最优解。在 DIMACS 基准图上的实验结果表明,该算法可以有效地解决 BWGCP 问题。同时我们发现改进解均衡性的局部搜索算法和进化算法中的交叉策略还有待改进和提高,后续研究其他的策略来改进均衡性,同时进一步探究均衡加权图着色问题在其他现实场景中的实际问题。

## 参考文献

- [1] FURINI F, MALAGUTI E. Exact weighted vertex coloring via branch-and-price[J]. *Discrete Optimization*, 2012, 9(2): 130-136.
- [2] GONG G W, XIE T, ZHAO H T et al. A graph coloring-based resource allocation algorithm for large-scale UAV swarm 3D network[J]. *Signal Processing*, 2022, 38(8): 1693-1702.
- [3] HAN G L, DU W. Research on the application of graph coloring model in the parking space allocation problem[J]. *Telecommunications Information*, 2019(7): 27-33.
- [4] GE R L, JIANG L, CHEN M Y et al. Spectrum allocation algorithm for elastic optical networks based on graph coloring model[J]. *Optical Communication Technology*, 2023, 47(2): 59-63.
- [5] DEGRATIAS E. Using Graph Coloring for Effective Timetable Scheduling at Ordinary Secondary Level[J]. *International Journal of Curriculum and Instruction*, 2022, 14(2): 1166-1188.
- [6] LI Z, WANG S, LI W D, et al. A VRP-based Approach for the Airline Crew Pairing Problem[C] // 2022 10th International Conference on Information Systems and Computing Technology (ISCTech). IEEE, 2022: 309-315.
- [7] PRAJAPATI V K, JAIN M, CHOUHAN L. Tabu search algorithm(TSA): A comprehensive survey[C] // 2020 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE). IEEE, 2020: 1-8.
- [8] WEI Z, HAO J K. Kernel based tabu search for the Set-union Knapsack Problem[J]. *Expert Systems with Applications*, 2021, 165: 113802.
- [9] MIRJALLI S, DONG J S, LEWIS A. Ant colony optimizer: Theory, literature review, and application in AUV path planning: Methods and applications[J]. *Studies in Computational Intelligence*, 2020, 811: 7-21.
- [10] KOSE A, SONMEZ B A, BALABAN M. Simulated annealing algorithm for graph coloring[J]. *arXiv:1712.00709*, 2017.
- [11] LÜ Z, HAO J K. A memetic algorithm for graph coloring[J]. *European Journal of Operational Research*, 2010, 203(1): 241-250.
- [12] HAO J K. Memetic algorithms in discrete optimization [M] // *Handbook of Memetic Algorithms*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012: 73-94.
- [13] TITLOYE O, CRISPIN A. Graph coloring with a distributed hybrid quantum annealing algorithm[C] // *Agent and Multi-Agent Systems: Technologies and Applications: 5th KES International Conference (KES-AMSTA 2011)*. Manchester, UK, Springer Berlin Heidelberg, 2011: 553-562.
- [14] ARDELEAN S M, UDRESCU M. Graph coloring using the reduced quantum genetic algorithm[J]. *PeerJ Computer Science*, 2022, 8: e836.
- [15] MOALIC L, GONDRAN A. Variations on memetic algorithms for graph coloring problems[J]. *Journal of Heuristics*, 2018, 24: 1-24.
- [16] WU J, CHANG Z, YUAN L, et al. A memetic algorithm for resource allocation problem based on node-weighted graphs [application notes][J]. *IEEE Computational Intelligence Magazine*, 2014, 9(2): 58-69.
- [17] LEHMAN J, STANLEY K O. Novelty search and the problem with objectives[C] // *Genetic Programming Theory and Practice IX*. Berlin: Springer, 2011: 37-56.
- [18] WANG J C, WANG S, LI Z, et al. Improved algorithm for forbidden search in graph coloring problem[J]. *Computer Science*, 2022, 49(S2): 94-98.



**OU Kaiming**, born in 1997, postgraduate. His main research interest is optimization problems.



**JIANG Hua**, born in 1978, Ph.D, associate professor, is a professional member of CCF(No. D7308M). His main research interests include practical algorithm solving for combinatorial optimization problems, algorithms for social network analysis and

graph theory.