

一种高效的求解最小负载着色问题的局部搜索算法^{*}

田新亮^{1,2}, 欧阳丹彤^{1,2}, 周慧思^{1,2}, 蒋璐宇^{1,2}, 太 然^{1,2}, 张立明^{1,2}

¹(吉林大学 计算机科学与技术学院, 吉林 长春 130012)

²(符号计算与知识工程教育部重点实验室 (吉林大学), 吉林 长春 130012)

通信作者: 张立明, E-mail: limingzhang@jlu.edu.cn



摘 要: 最小负载着色问题 (minimum load coloring problem, MLCP) 源于构建光通信网络的波分复用 (wavelength division multiplexing, WDM) 技术, 是一个被证明的 NP 完全问题. 由于 NP 完全问题有着随问题规模呈指数增长的解空间, 因此启发式算法常被用来解决这类问题. 在对国内外相关工作的深入分析基础上得知, 现有的多类求解 MLCP 问题的启发式算法中局部搜索算法表现是最好的. 研究针对当前求解 MLCP 问题的局部搜索算法在数据预处理和邻域空间搜索上的不足, 提出了两点相应的优化策略: 一是在数据的预处理阶段, 提出一度顶点规则来约简数据的规模, 进而减小 MLCP 问题的搜索空间; 二是在算法的邻域空间搜索阶段, 提出两阶段多重选择策略 (two-stage best from multiple selections, TSBMS) 来帮助局部搜索算法在面对不同规模的邻域空间时可以高效地选择一个高质量的邻居解, 它有效地提高了局部搜索算法在处理不同规模数据时的求解表现. 将这个优化后的局部搜索算法命名为 IRLTS. 采用 74 个经典的测试用例来验证 IRLTS 算法的有效性. 实验结果表明, 无论最优解还是平均解, IRLTS 算法在大多数测试用例上都明显优于当前表现最好的 3 个局部搜索算法. 此外, 还通过实验验证了所提策略的有效性以及分析了关键参数对算法的影响.

关键词: 最小负载着色问题; 启发式算法; 局部搜索算法

中图法分类号: TP301

中文引用格式: 田新亮, 欧阳丹彤, 周慧思, 蒋璐宇, 太然, 张立明. 一种高效的求解最小负载着色问题的局部搜索算法. 软件学报, 2025, 36(8): 3677–3692. <http://www.jos.org.cn/1000-9825/7241.htm>

英文引用格式: Tian XL, Ouyang DT, Zhou HS, Jiang LY, Tai R, Zhang LM. Efficient Local Search Algorithm for Solving Minimum Load Coloring Problem. Ruan Jian Xue Bao/Journal of Software, 2025, 36(8): 3677–3692 (in Chinese). <http://www.jos.org.cn/1000-9825/7241.htm>

Efficient Local Search Algorithm for Solving Minimum Load Coloring Problem

TIAN Xin-Liang^{1,2}, OUYANG Dan-Tong^{1,2}, ZHOU Hui-Si^{1,2}, JIANG Lu-Yu^{1,2}, TAI Ran^{1,2}, ZHANG Li-Ming^{1,2}

¹(College of Computer Science and Technology, Jilin University, Changchun 130012, China)

²(Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education (Jilin University), Changchun 130012, China)

Abstract: The minimum load coloring problem (MLCP) is an important NP-complete problem arising from wavelength division multiplexing (WDM), a technology used for building optical communication networks. The solutions to NP-complete problems grow exponentially as the size of the problems expands, so heuristic algorithms are often used to solve such problems. Analysis of research at home and abroad shows that among the existing heuristic algorithms for solving the MLCP, local search algorithms exhibit the best performance. This study proposes two optimization strategies to overcome the limitations of existing local search algorithms in data preprocessing and neighborhood space search. First, during data preprocessing, a one-degree vertex rule is proposed to reduce the size of data and thus reduce the search space of the MLCP. Second, in the search phase of the algorithm, a strategy termed two-stage best from

* 基金项目: 国家自然科学基金 (62076108, 61872159)

收稿时间: 2023-01-07; 修改时间: 2024-01-30, 2024-04-17; 采用时间: 2024-06-18; jos 在线出版时间: 2024-12-31

CNKI 网络首发时间: 2025-01-02

multiple selections (TSBMS) is proposed to help local search algorithms efficiently select a high-quality neighborhood solution for neighborhood space with different sizes, which effectively improves the performance of local search algorithms for processing data of different sizes. This optimized local search algorithm is named IRLTS. Seventy-four classic test instances are adopted to validate the effectiveness of the IRLTS algorithm. Experimental results demonstrate that the IRLTS algorithm outperforms the three best local search algorithms on most test instances in terms of both optimal and average solutions. Furthermore, the effectiveness of the proposed strategy is validated through experiments, and the influence of key parameters on the IRLTS algorithm is analyzed.

Key words: minimum load coloring problem (MLCP); heuristic algorithm; local search algorithm

给定一个无向图 $G = (V, E)$, 其中 V 表示由 n 个顶点构成的集合, E 表示由 m 条边构成的集合. 最小负载着色问题 (minimum load coloring problem, MLCP) 旨在将集合 V 中的顶点划分为两个不相交的子集 V_r 和 V_b , 同时寻求两个子集的内部边总数中较小的值最大. 在这里, 集合的内部边是指边的两个端点属于同一个集合的边. 划分完成后, 集合 V_r 中的顶点被标记为红色, 集合 V_b 中的顶点被标记为蓝色^[1].

最小负载着色问题源于波分复用 (wavelength division multiplexing, WDM) 技术, 它在构建复杂的通信网络和大型的电网网络中有着广泛的应用^[2,3]. 下面我们通过一个简单的例子来解释 MLCP 问题在波分复用网络中的应用. 图 1 是一个光广播波分复用网络, 它由双通道无源星形耦合器连接 5 个节点构成. 网络中的每个节点都配备了一个可调发射机, 它可以利用两通道中的任何一个来传输消息. 此外, 每个节点还配备了一个固定调谐接收机, 它只能接收在前缀信道中发送的消息. 波分复用网络是一个 2-均匀超图 (two-uniform hypergraph), 网络中的节点需要使用一个或两个通道向另外两个节点发送消息, 该消息可以被连接到该通道的所有节点接收. 怎样将信道分配给这些节点的接收器而使得两个信道上的最大负载最小化, 这个问题可以转化为 MLCP 问题, 其中边对应于波分复用网络中的节点, 边 e_i 的两个端点对应于波分复用网络中节点 i 传输消息的两个节点. MLCP 问题寻求将图中的顶点着色为两种颜色, 每种颜色对应于波分复用网络中的一个通道. MLCP 问题所寻求的将两个子集内部边总数中较小的值最大等同于将两个通道上的最大负载最小化. 关于 MLCP 问题在波分复用网络中的更多的应用细节可以从文献 [4] 获知.

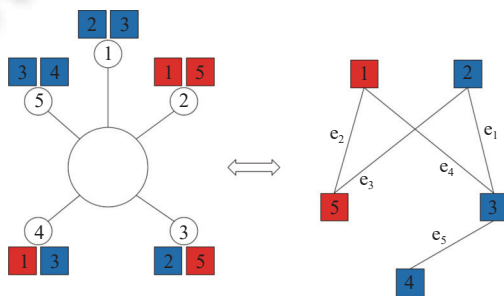


图 1 广播波分复用网络及其对应的图

鉴于 MLCP 问题的重要性和广泛应用性, 现有文献中存在很多关于 MLCP 问题的研究. 这些研究大致可以分为两类: 一类是致力于寻求 MLCP 问题最优解的近似算法和多项式时间算法. 这类算法的优点是可以确保所找到的解为最优解, 但是限于 MLCP 问题的复杂性, 在面对大规模的图测试用例时, 这类算法会耗费大量的运算时间. 因此现有的关于这类算法的研究多限于一些规模较小的图测试用例以及一些特殊的图测试用例, 比如树以及一些不相连的团构成的图. Ahuja 等人^[2]针对树形数据设计了一个复杂度为 $O(n^3)$ 的多项式时间算法, 并且证明了该类数据的最优解不超过 $1/2 + (\Delta/m)\log_2 n$, 其中 Δ 表示顶点度的最大值. Gutin 等人^[5]证明了宽度为 t 的树的最优解可以在时间 $O^*(2^t)$ 内被找到. Barbero 等人^[6]针对 MLCP 的扩展版本问题最大 c 负载着色 (max c -load coloring) 问题设计了相应的近似算法.

另一类求解 MLCP 问题的算法是启发式算法. 不同于近似算法和多项式算法, 启发式算法寻求在限定时间内尽可能地去找到一个高质量的解. 在面对大规模的测试用例时, 启发式算法是常被采用的一类算法并且有着出色的求解表现^[7,8]. Tang 等人^[9]利用人工蜂群算法来求解最小负载着色问题, 并设计了模拟退火算法作为对比算法.

实验结果显示 Tang 等人改进的人工蜂群算法在求解 MLCP 问题上有着良好的表现. Ye 等人^[10]针对 MLCP 问题设计了一个高效的局部搜索算法, 实验结果显示该算法远远超过当时其他求解 MLCP 问题的启发式算法. Zhang 等人^[11]结合局部搜索和进化算法设计了一个高效的文化基因算法, 实验结果显示该算法在部分标准测试数据上提高了已知的最优解. 近几年, Sun 等人^[1]和 Herrán 等人^[12]分别设计了两个不同版本的局部搜索算法, 是当前求解 MLCP 问题表现最好的启发式算法. Sun 等人设计的局部搜索算法在搜索过程中利用强化学习机制生成新的初始解来为局部搜索算法开辟新搜索区域. Alberto 等人设计了多个不同的邻域空间来帮助局部搜索算法在不同的区域搜索最优解, 这大大提高了算法的搜索表现.

虽然当前存在多种求解 MLCP 问题表现良好的启发式算法, 但现有的启发式算法多是针对较小规模的测试用例设计的, 所求解测试用例的顶点数目只有几百个. 随着科技的进步, 各种通信网络的规模越来越大, 所对应测试用例的规模也越来越大. 现有的启发式算法在面对大规模顶点数目所造成的巨大搜索空间时表现乏力, 因此, 设计高效的启发式算法去解决 MLCP 问题中的大规模图测试用例是亟待解决的一个问题. 本文在当前表现最好的一个启发式算法, 即 Sun 等人设计的高效的局部搜索算法 RLTS (reinforcement learning based tabu search) 的基础上, 提出了两点针对大规模测试用例的优化策略, 具体如下:

(1) 当前的启发式算法处理大规模图测试用例所面临的一个挑战在于随图顶点数目呈指数增长的解空间. 为此我们提出一度顶点规则对图测试用例进行约简. 一度顶点规则旨在通过约简大规模图测试用例的顶点数目来达到缩减解空间规模的目的, 它有效地减少了图测试用例的顶点数目, 从而为算法后期的搜索提供一个较小的搜索空间. 在第 3.1 节中将详细介绍一度顶点规则, 并对该规则的有效性进行理论性分析.

(2) 局部搜索算法通过不断迭代地从当前解移动到邻居解的方式来搜索最优解, 当前表现最优的局部搜索算法 RLTS 在搜索合适的邻居解时存在着过于耗时的缺陷, 在面对有较多顶点数的图测试用例所形成的大规模领域空间时, 这一缺陷尤为明显. 为此我们提出了两阶段多重选择策略 (two-stage multiple selection strategy, TSBMS) 来进一步优化当前最优的局部搜索算法 RLTS. TSBMS 策略改进于 Cai^[13]提出的多重选择策略 (multiple selection strategy, BMS), BMS 策略被广泛用于解决各类大规模图测试用例的组合优化问题^[14-16], 但是它存在着处理小规模图测试用例表现不佳的缺陷. 为此我们进一步优化 BMS 策略得到了 TSBMS 策略. TSBMS 策略可以高效地帮助 RLTS 算法选择合适的邻居解, 提高 RLTS 算法的适用性, 使其在面对不同规模的图测试用例时都有优秀的表现.

我们将优化后的 RLTS 算法命名为 IRLTS (improved reinforcement learning based tabu search). 在实验部分, 我们采用 74 个不同规模的经典测试用例来验证 IRLTS 算法的有效性. 实验结果表明, 无论最优解还是平均解, IRLTS 算法在大多数测试用例上都明显超过了当前最优的 3 个启发式算法.

本文第 1 节介绍 MLCP 问题的具体定义. 第 2 节介绍当前表现最优的 RLTS 算法. 第 3 节介绍我们提出的一度顶点规则和 TSBMS 策略, 以及优化 RLTS 算法后所形成的 IRLTS 算法. 在第 4 节, 我们通过对比实验来验证 IRLTS 算法和所提策略的有效性, 以及分析关键参数对 IRLTS 算法的影响. 第 5 节总结全文.

1 MLCP 问题定义

现有文献中存在多种对 MLCP 问题的具体定义. 在本文的研究中, 我们采用文献 [1] 中关于 MLCP 问题的定义. 给定一个无向图 $G = (V, E)$, 其中 V 表示由 n 个顶点构成的集合 ($V = \{v_1, v_2, v_3, \dots, v_n\}$), E 表示由 m 条边构成的集合 ($E = \{e_1, e_2, e_3, \dots, e_m\}$). 边是由两个顶点构成的无序二元组, 一般表示为 $e = \{u, v\}$, 简写为 e_{uv} , 其中 u 和 v 表示边 e 的两个端点. 最小负载着色问题旨在将集合 V 中的顶点划分为两个不相交的子集 V_r 和 V_b , 同时寻求两个子集合的 $|E(V_r)|$ 和 $|E(V_b)|$ 中较小的值最大, 其中 $E(V_r)$ 表示两端顶点都属于集合 V_r 的边的集合, $|E(V_r)|$ 表示两端顶点都属于集合 V_r 的边的数目. 同理, $E(V_b)$ 表示两端顶点都属于集合 V_b 的边的集合, $|E(V_b)|$ 表示两端顶点都属于集合 V_b 的边的数目. 划分完成后, 集合 V_r 中的顶点被标记为红色, 集合 V_b 中的顶点被标记为蓝色.

我们用 $e_{ij} = 1$ 表示顶点 v_i 和 v_j 之间存在边连接, $e_{ij} = 0$ 表示顶点 v_i 和 v_j 之间不存在边连接. $x_i = 1$ 表示顶点 v_i 属于集合 V_r , $x_i = 0$ 表示顶点 v_i 属于集合 V_b . $r_{ij} = 1$ 表示顶点 v_i 和 v_j 同时属于集合 V_r , 如果顶点 v_i 和 v_j 不同时属于

集合 V_r 则用 $r_{ij} = 0$ 表示. $b_{ij} = 1$ 表示顶点 v_i 和 v_j 同时属于集合 V_b , 如果顶点 v_i 和 v_j 不同时属于集合 V_b 则用 $b_{ij} = 0$ 表示. z 表示两端顶点都属于集合 V_r 的边的数目和两端顶点都属于集合 V_b 的边的数目中较小的数值.

$$\begin{aligned} & \max z & (1) \\ & \text{s.t.} \begin{cases} \textcircled{1} 2 \times r_{ij} < x_i + x_j, \forall i, j \in \{1, \dots, n\} \\ \textcircled{2} 2 \times b_{ij} \leq 2 - x_i - x_j, \forall i, j \in \{1, \dots, n\} \\ \textcircled{3} z \leq \sum_{i \in V} \sum_{j \in V} r_{ij} \times e_{ij} \\ \textcircled{4} z \leq \sum_{i \in V} \sum_{j \in V} b_{ij} \times e_{ij} \\ \textcircled{5} b_{ij}, r_{ij} \in \{0, 1\}, \forall i, j \in \{1, \dots, n\} \\ \textcircled{6} x_i \in \{0, 1\}, \forall i \in \{1, \dots, n\} \\ \textcircled{7} z \in N \end{cases} & (2) \end{aligned}$$

公式 (1) 表示 MLCP 问题寻求两端顶点都属于集合 V_r 的边的数目和两端顶点都属于集合 V_b 的边的数目中较小的那个值最大. 约束①确保只有当顶点 v_i 和 v_j 同时属于集合 V_r 时 r_{ij} 的值为 1. 约束②确保只有当顶点 v_i 和 v_j 同时属于集合 V_b 时 b_{ij} 的值为 1. $\sum_{i \in V} \sum_{j \in V} r_{ij} \times e_{ij}$ 表示两个端点都属于集合 V_r 所构成的边的数目. $\sum_{i \in V} \sum_{j \in V} b_{ij} \times e_{ij}$ 表示两个端点都属于集合 V_b 所构成的边的数目, 约束③、④确保 z 的值小于等于集合 V_r 和 V_b 各自内部边总数中较小的那个值. 约束⑤–⑦约束了变量的取值范围.

2 RLTS 算法

RLTS 算法是当前求解 MLCP 问题表现最好的启发式算法, 本文所提出的 IRLTS 算法正是以 RLTS 算法为基础优化得到的, 本节将详细介绍 RLTS 算法. RLTS 算法主要由局部搜索过程和强化学习机制两部分构成. 局部搜索过程在输入初始解的邻域空间内不断地寻找最优解, 直到算法陷入局部最优. 强化学习机制根据局部搜索过程所获取的新的最优解来不断地更新概率矩阵, 概率矩阵则指导生成新的初始解提供给局部搜索过程. RLTS 算法在生成新初始解的强化学习机制和搜索初始解邻域空间的局部搜索过程间不断切换来寻找最优解. RLTS 算法的框架如算法 1 所示.

算法 1. RLTS 算法.

输入: 无向图 $G=(V, E)$, 时间限制 $t_{\max}$;

输出: 发现的最优解 S_{best} .

```

1. 初始化概率矩阵  $P$ 
2. for  $i=1, \dots, n$  do
3.    $p_{i0}=p_{i1}=1/2$ 
4. while time() $<t_{\max}$  do
5.    $S_{\text{initial}} \leftarrow \text{Subset\_Selection}(P)$ 
6.    $S_{\text{target}} \leftarrow \text{Local\_Search}(S_{\text{initial}})$ 
7.   if  $S_{\text{target}}$  is better than  $S_{\text{best}}$  then
8.      $S_{\text{best}} \leftarrow S_{\text{target}}$ 
9.    $S_{\text{target}} \leftarrow \text{Subset\_Matching}(S_{\text{initial}}, S_{\text{target}})$ 
10.   $P \leftarrow \text{Probability\_Updating}(S_{\text{initial}}, S_{\text{target}}, P)$ 
11. return  $S_{\text{best}}$ ;

```

RLTS 算法采用一个 $n \times 2$ 的矩阵 P 来表示概率矩阵, 其中元素 $p_{ir} (i \in \{1, \dots, n\})$ 表示顶点 v_i 分配给集合 V_r 的概率, 元素 p_{ib} 表示顶点 v_i 分配给集合 V_b 的概率. 在 RLTS 算法的初始阶段, 概率矩阵 P 中的每个元素都被设置为

1/2, 也就是在初始阶段每个顶点被分配到集合 V_r 和 V_b 的概率是相等的. 在迭代过程 (第 4–10 行) 中, RLTS 算法首先根据概率矩阵生成一个初始解 S_{initial} (第 5 行), 然后利用局部搜索过程在该初始解的邻域空间进行搜索, 直至找到一个局部最优解 S_{target} (第 6 行). 为了解决可能存在的对称解问题 (solution symmetry), 子集匹配函数被设计来发现初始解 S_{initial} 和局部最优解 S_{target} 之间的子集对应关系 (第 9 行). 之后, 通过比较初始解 S_{initial} 和局部最优解 S_{target} 之间的差异, 概率矩阵 P 被更新 (第 10 行). 至此, RLTS 算法的一次迭代过程执行完成. 在运行时间达到最大时间限制时, RLTS 算法结束迭代并返回发现的最优解.

2.1 子集分配

本节将详细概述 RLTS 算法的子集分配过程 (算法 1 第 5 行). 概率矩阵 P 中存储了每个顶点 v_i 被分配到集合 V_r 和 V_b 的概率 p_{ir} 和 p_{ib} . RLTS 算法依据轮盘选择策略^[17]来计算分配顶点到集合 V_r 或 V_b 的概率值, 由轮盘选择策略计算可知顶点 v_i 被分配到集合 V_r 的概率值是 $p_{ir}/(p_{ir} + p_{ib})$, 顶点 v_i 被分配到集合 V_b 的概率值是 $p_{ib}/(p_{ir} + p_{ib})$. 具体的子集分配过程如算法 2 所示.

算法 2. 子集分配过程.

输入: 无向图 $G=(V, E)$, 概率矩阵 P ;

输出: 初始解 S_{initial} .

```

1. 初始化概率矩阵  $P$ 
2. for  $i=1, \dots, n$  do
3.    $l = p_{ir}/(p_{ir}+p_{ib})$ 
4.   if random(0, 1) <  $l$  then
5.     将顶点  $v_i$  分配至集合  $V_r$ 
6.   else
7.     将顶点  $v_i$  分配至集合  $V_b$ 
8. return  $S_{\text{initial}}$ ;
```

2.2 局部搜索

RLTS 算法通过局部搜索过程来提高输入初始解的质量, 它主要由两部分构成: 一是对当前解邻域空间的搜索过程, 二是算法陷入局部最优后的扰动机制. 算法 3 是 RLTS 算法局部搜索过程的伪代码, 局部搜索在每次的迭代过程中通过概率的方式^[18]选择一个邻域空间来搜索最优解 (第 6–13 行). 如果在该领域空间内所发现的新的最优解优于搜索历史中记录的最优解, 则历史最优解被更新, 否则搜索失败的次数加 1 (第 14–18 行). 当失败的次数超过 T_p 时, 局部搜索的扰动机制被触发 (第 20–22 行). 扰动结束后所产生的新的初始解被局部搜索过程重新搜索其邻域空间. 当局部搜索算法的迭代次数超过 T_r 时, 局部搜索过程结束 (第 3、5 行). 算法中的扰动机制是通过在集合 V_r 和 V_b 中随机地选择 $k \times |V|$ 对顶点交换来实现的, 其中 k 是一个通过实验来确定的参数. 下面我们将详细地介绍局部搜索过程的主要组成部分: 领域空间 N_1 和 N_2 , tabu 禁忌机制.

算法 3. 局部搜索过程.

输入: 初始解 S_{initial} , 局部搜索最大迭代次数 T_r , 触发扰动的迭代次数 T_p ;

输出: 发现的最优解 S_{best} .

```

1.  $S_{\text{best}} \leftarrow S_{\text{initial}}$ 
2.  $S \leftarrow S_{\text{initial}}$ 
3.  $Iter \leftarrow 0$ 
4.  $Noimprove \leftarrow 0$ 
```

```

5. while  $Iter < T_r$  do
6.   if 小于概率值  $p$  then
7.     构造当前解  $S$  的邻域空间  $N_1$ 
8.     在  $N_1$  中选择未被 tabu 禁止的最优解  $S'$ 
9.      $S \leftarrow S'$ 
10.  else
11.    构造当前解  $S$  的邻域空间  $N_2$ 
12.    在  $N$  中选择未被 tabu 禁止的最优解  $S'$ 
13.     $S \leftarrow S'$ 
14.    if  $f(S) > f(S_{best})$  then
15.       $S_{best} \leftarrow S$ 
16.       $Noimprove \leftarrow 0$ 
17.    else
18.       $Noimprove \leftarrow Noimprove + 1$ 
19.     $Iter \leftarrow Iter + 1$ 
20.    if  $Noimprove > T_p$  then
21.       $Noimprove \leftarrow 0$ 
22.       $S \leftarrow Perturb(S)$ 
23. return  $S_{best}$ 

```

无向图 $G = (V, E)$ 的顶点集合 V 被划分成任意两个不相交的子集 V_r 和 V_b 后会产生一个可行解 S . 搜索空间 Ω 由所有的可行解构成, 也就是 $\Omega = \{V_r : V_b\} : V_r \cap V_b = \emptyset, V_r \cup V_b = V\}$. 显然, 搜索空间的大小为 2^n , 其中 n 为顶点数目. 对于一个可行解 $S = \{V_r, V_b\}$, 该可行解的适应度值为 $f(S) = \min\{|E(V_r)|, |E(V_b)|\}$.

2.2.1 解空间 N_1

对于当前解 $S = \{V_r, V_b\}$, 从子集合 V_r 或 V_b 中任意选择一个顶点 v 移动到另一个子集合, 该行为产生的所有可行解构成解空间 N_1 . 为了快速地计算移动一个顶点后所生成候选解 $S' = \{V'_r, V'_b\}$ 的适应度值, 也就两个子集合 V'_r 和 V'_b 的内部边的数目 $|E(V'_r)|$ 和 $|E(V'_b)|$, RLTS 算法用数组 $\delta_i (i \in \{1, \dots, n\})$ 来保存与顶点 v_i 存在连接且与顶点 v_i 所属子集合相同的顶点数目. 此外, RLTS 算法用数组 D 来保存顶点的度, D_i 存储顶点 v_i 的度. 在此基础上, 我们可以根据下述公式快速地计算出候选解 S' 的适应度值: 如果 $v_i \in V_r$, 则 $|E(V'_r)| = |E(V_r)| - \delta_i$, $|E(V'_b)| = |E(V_b)| + D_i - \delta_i$. 如果 $v_i \in V_b$, 则 $|E(V'_r)| = |E(V_r)| + D_i - \delta_i$, $|E(V'_b)| = |E(V_b)| - \delta_i$. 在移动一个顶点 $v_i \in V$ 后, RLTS 算法可以依照下面公式在 $O(n)$ 内更新数组 δ :

对于顶点 v_i , $\delta_i = D_i - \delta_i$. 对于除顶点 v_i 外的其他与顶点 v_i 相连接的顶点 v_u ,

$$\delta_u = \begin{cases} \delta_u - 1, & \text{如果 } v_u \text{ 属于 } v_i \text{ 移动前的集合} \\ \delta_u + 1, & \text{如果 } v_u \text{ 不属于 } v_i \text{ 移动前的集合} \end{cases} \quad (3)$$

2.2.2 解空间 N_2

对于当前解 $S = \{V_r, V_b\}$, 从子集合 V_r 和 V_b 中各自随机选择一个顶点 v_i 和 v_j , 交换两个顶点所属的子集合, 该行为产生的所有可行解构成解空间 N_2 . 与计算单个顶点 v 移动后所产生候选解适应度值的方式相同, 借助数组 δ , 在完成交换操作后所产生候选解的适应度值也可以在 $O(1)$ 内完成: $|E(V'_r)| = |E(V_r)| + D_i - \delta_i - \delta_j - E_{ij}$, $|E(V'_b)| = |E(V_b)| + D_i - \delta_i - \delta_j - E_{ij}$. 交换操作可以看作是二个顶点移动操作的组合, 数组 δ 在交换操作中的单个顶点移动操作完成后被更新, 更新公式与公式 (3) 相同.

2.2.3 禁忌管理

在局部搜索过程中, 当顶点 v 被从当前的子集合移动到另一个子集合后, 它会很容易被再次选择移动回原有

的子集合. RLTS 算法采用禁忌表 (tabu list)^[19] 的方式来阻止这种短期循环 (short-term cycling) 现象的产生. 当顶点 v 被从当前的子集合移动到另一个子集合后, 它将被禁止在未来的 tt 次迭代内移动回最初的子集合. 只有当移动该顶点所产生解的质量优于当前最优解的质量时, 该禁忌策略可以被违背. RLTS 算法中的禁忌表是通过一个 $|V| \times 2$ 的二维数组 T 来实现的. 每次顶点 v 被从子集合 j ($j=0$ 表示子集合 V_r , $j=1$ 表示子集合 V_b) 移除后, $T[v][j]$ 被更新为 $Iter+tt$, 其中 $Iter$ 是当前的迭代次数. 在算法的迭代过程中, 当 $Iter < T[v][j]$ 时, 顶点 v 禁止被移回子集合 j . 当 $Iter > T[v][j]$ 时, 顶点 v 不受该规则限制.

2.3 子集匹配

MLCP 问题旨在将集合 V 中的顶点划分到两个不相交的子集 V_r 和 V_b . RLTS 算法用数组去表示一个可行解, 下标为 i 的元素置为 1 或 0 表示顶点 v_i 被分配到集合 V_r 或 V_b . 在 RLTS 算法中, 强化学习机制是通过初始解 S_{initial} 和局部最优解 S_{target} 各自分配到集合 V_r 和 V_b 中的顶点个数差异来学习的. 很显然, 这必然会导致一种解对称 (solution symmetry) 问题^[20]. 例如, 无向图 G 有 6 个顶点 $\{v_1, v_2, v_3, v_4, v_5, v_6\}$, 存在两个解, 解 $S_1 = \{V_r, V_b\}$, 其中 $V_r = \{v_1, v_2, v_3\}$, $V_b = \{v_4, v_5, v_6\}$, 解 $S_2 = \{V'_r, V'_b\}$, 其中 $V'_r = \{v_4, v_5, v_6\}$, $V'_b = \{v_1, v_2, v_3\}$. 其实解 S_1 和 S_2 是相同的, 但是在算法中去表示这两个解的数组 $\{0, 0, 0, 1, 1, 1\}$ 和 $\{1, 1, 1, 0, 0, 0\}$ 会被看作是解. RLTS 算法设计子集匹配函数 (subset matching, 算法 1 第 9 行) 来解决该问题, 具体方法如下.

对于初始解 $S_{\text{initial}} = \{V_r, V_b\}$ 以及局部搜索提高后的局部最优解 $S_{\text{target}} = \{V'_r, V'_b\}$. 算法通过下面的方式来处理对称解问题: 如果 $|V_r \cap V'_r| + |V_b \cap V'_b| < |V_r \cap V'_b| + |V_b \cap V'_r|$, 子集匹配函数选择交换局部最优解 S_{target} 两个子集中的顶点, 否则, 局部最优解 S_{target} 的两个子集保持不变.

2.4 强化学习机制

RLTS 算法中的强化学习机制^[21]是根据初始解 S_{initial} 和经过局部搜索后得到的局部最优解 S_{target} 之间各个顶点 v 所属集合的差异来学习的, 它是通过增加或者减小概率矩阵 P 中顶点 v 分配到集合 V_r 或者 V_b 的概率值来实现的, 具体步骤如下.

在经历局部搜索后, 如果顶点 v 依然被保留在原有的集合中, RLTS 算法就采用公式 (4) 来增加顶点 v 保留在原有集合中的概率值, 以及减小移动到另一个集合的概率值. 我们用 u 来表示原有的集合, $u \in \{0, 1\}$.

$$p_{ij} = \begin{cases} \alpha + (1-\alpha)p_{ij}, & j = u \\ (1-\alpha)p_{ij}, & j = 1-u \end{cases} \quad (4)$$

如果在经历局部搜索后, 顶点 v 从集合 V_u 移动到集合 V_{1-u} , RLTS 算法则降低顶点 v 分配到集合 V_u 的概率值, 增加 v 分配到集合 V_{1-u} 的概率值. 具体公式如下, 其中 β 是惩罚因子, γ 是补偿因子.

$$p_{ij} = \begin{cases} (1-\gamma)(1-\beta)p_{ij}, & j = u \\ \gamma + (1-\gamma)\beta + (1-\gamma)(1-\beta)p_{ij}, & j = 1-u \end{cases} \quad (5)$$

最后, RLTS 算法为了防止顶点 v 被过于贪心地分配到某个集合内, 强化学习机制限制概率值的上限和下限区间为 $[0.2, 0.8]$, 这为 RLTS 算法生成初始解增添了一些随机性.

3 IRLTS 算法

IRLTS 算法是我们在 RLTS 算法的基础上提出两点优化策略后得到的一个高效的启发式算法. 第 3.1 节和第 3.2 节将详细地介绍针对 RLTS 算法不足所提出的两点优化策略: 一度顶点规则和 TSBMS 策略. 经过这两个策略优化后, 生成新的高效算法. 我们将其命名为 IRLTS 算法, 该算法将在第 3.3 节介绍.

3.1 一度顶点规则

MLCP 问题是一个经典的 NP 完全问题, 它的解空间随测试用例的顶点数呈指数增长. 为了约简测试用例的顶点数, 进而降低 MLCP 问题的解空间规模, 我们提出一度顶点规则来约简测试用例的顶点数目. 下面给出该规则的有效性分析以及具体定义.

如果顶点 v_i 的度数为 1, 则顶点 v_i 与其邻接顶点 v_j 之间只存在一条边 e_{ij} . 对于当前解 $S = \{V_r, V_b\}$, 若顶点 v_i 和 v_j 所属的集合不同, 则只会出现两种情况: $v_j \in V_r$ 且 $v_i \in V_b$ 或者 $v_j \in V_b$ 且 $v_i \in V_r$. 当 $v_j \in V_r$ 且 $v_i \in V_b$ 时, 边 $e_{ij} \notin E(V_r)$ 且 $e_{ij} \notin E(V_b)$. 若在此基础上将顶点 v_i 移动至集合 V_r 形成一个新的解 $S' = \{V'_r, V'_b\}$, 即 $v_j \in V'_r$ 且 $v_i \in V'_r$, 则 $e_{ij} \in E(V'_r)$, $|E(V'_r)| = |E(V_r)| + 1$, $|E(V'_b)| = |E(V_b)|$. 因为适应度值 $f(S) = \min\{|E(V_r)|, |E(V_b)|\}$, 因此, 如果 $|E(V_r)| < |E(V_b)|$, 则 $F(S') = F(S) + 1$, 适应度值被提高. 如果 $|E(V_r)| \geq |E(V_b)|$, 则 $F(S') = F(S)$, 适应度值不变. 当 $v_j \in V_b$ 且 $v_i \in V_r$ 时与此相同, 我们不再另作分析.

基于上述分析过程, 我们给出一度顶点规则的定义, 即度为 1 的顶点只有在与其邻接顶点所属集合相同时才有可能提高解的适应度值, 故在数据预处理阶段可首先对度为 1 的顶点及其邻接边进行约简, 从而减少测试用例的顶点数目.

由一度顶点规则可知, 度为 1 的顶点只有在与其邻接顶点所属集合相同时才有可能提高解的适应度值. 借助此规则, 我们可以在数据预处理阶段对数据进行约简, 减少测试用例的顶点数目. 具体操作如下: 首先, 我们将测试用例中度为 1 的顶点及其邻接边在测试用例中移除. 其次, 将约简后的测试用例作为 IRLTS 算法的输入用例来获取最优解. 再次, 根据获取的最优解中顶点的划分结果来确定所移除的度为 1 顶点的所属集合, 移除顶点应与其邻接顶点所属集合相同. 最后, 根据补充完整后的测试用例调整适应度值. 在第 4.2 节, 我们通过实验结果来验证一度顶点规则的约简效果.

3.2 TSBMS 策略

局部搜索算法通过不断迭代地从当前解移动到另一个质量更优邻居解的方式来寻找最优解. 如算法 3 所示, RLTS 算法依据概率 p 随机地从解空间 N_1 或者解空间 N_2 中选择一个质量最优的邻居解作为新的当前解. 显然, 在解空间 N_1 中选择一个最优邻居解的时间复杂度是 $O(n)$, 在解空间 N_2 中选择一个最优邻居解的时间复杂度是 $O(n^2)$, 其中 n 表示测试用例中顶点的数目. 当 n 较小时, RLTS 算法可以快速地在两个邻域空间中遍历选择出质量最优的邻居解. 但是随着数据规模的逐渐增大, n 越来越大时, RLTS 在两个空间中通过遍历选择最优邻居解的方式会耗费大量的时间, 对于解空间 N_2 , 该缺陷尤为明显.

为解决局部搜索算法面对大规模测试用例时选择最优邻居解过于耗时的缺点, Cai^[13]提出 BMS 策略, 该策略通过采样择优替代遍历搜索的方式在邻域空间中寻找最优邻居解, 被广泛用于多种大规模组合优化问题的局部搜索算法设计中, 并且有着优秀的表现^[14-16].

BMS 策略通过采样择优的方式寻找最优邻居解, 具体的操作步骤如下: 当算法面对一个较大规模的解空间时, BMS 策略被触发采用. BMS 策略并不采用遍历解空间中所有候选解后选择质量最优解的方式, 因为这种方式显然会耗费大量的时间, 从而使得算法单次迭代的时间较长, 在限定时间内, 算法无法完成多次迭代搜索. 面对大规模解空间, BMS 策略首先从解空间中随机选择 $bmsLen$ 个解 ($bmsLen$ 的值通常根据具体问题来设定), 然后再从 $bmsLen$ 个解中选择出质量最优的一个解来作为该次迭代发现的最优解. 因为 $bmsLen$ 的值通常远小于解空间的大小, 因此 BMS 策略所耗费的时间也远小于遍历搜索方式耗费的时间. BMS 策略取得邻居解的质量虽然不如遍历搜索得到的邻居解的质量, 但是该策略在面对大规模数据造成的巨大解空间时可以降低单次迭代耗费的时间, 从而增加局部搜索算法的迭代次数, 进而找到最优解. 文献 [13] 提供了 BMS 策略的有效性分析, 本文不再赘述.

当 BMS 策略面对较小规模的数据时, 采样择优方式所带来的降低单次迭代运行时间的优势不再明显, 而采样择优所找到的邻居解质量较差的劣势被进一步放大, 此时 BMS 策略不能很好地帮助局部搜索算法搜索最优解. 为了解决 BMS 策略存在的缺陷, 本文提出了优化版本的 BMS 策略, 即两阶段多重选择策略. TSBMS 策略不同于 BMS 策略, 它的搜索过程分为两个阶段: 当局部搜索算法面对的解空间较小时, TSBMS 策略利用遍历搜索的方式来寻找最优邻居解. 因为解空间较小, 遍历搜索可以快速找到最优解, 不耗费时间且能保证所找到邻居解的质量. 当局部搜索面对的解空间较大时, TSBMS 策略则选择利用 BMS 策略以采样择优的方式选择一个较优的邻居解. 具体的步骤如算法 4 所示.

算法 4. TSBMS 策略.

输入: 邻域空间 N , 临界值 $maxT$, 采样个数 $bmsLen$;

输出: 选择的最优邻居解 S .

1. **if** $|N| < maxT$ **then**
2. $S \leftarrow$ 遍历的方式从 N 中选择质量最优的一个解
3. **else**
4. $S \leftarrow$ 从 N 中随机选择 $bmsLen$ 个解, 然后从 $bmsLen$ 个解中选择质量最优的解
5. **return** S

如算法 4 所示, TSBMS 策略根据解空间的大小选择不同的方式寻找最优邻居解. 当解空间的规模小于 $maxT$ 时, TSBMS 策略利用遍历搜索的方式来寻找最优邻居解. 由于解空间的规模较小, 遍历搜索的方式既可以找到最高质量的邻居解, 也不会花费太多的时间. 当解空间的规模大于 $maxT$ 时, TSBMS 策略采用 BMS 策略以采样择优的方式选择质量较优的邻居解, 该策略可以避免遍历搜索导致的大量时间耗费, 快速地返回一个质量较优的邻居解.

3.3 IRLTS 算法

在数据的预处理阶段, IRLTS 算法利用一度顶点规则来约简数据的规模, 进而达到减小解空间规模的目的. 此外, TSBMS 策略被用来改进 RLTS 算法中的局部搜索过程, 改进后的局部搜索过程如算法 5 所示.

算法 5. IRLTS 算法的局部搜索过程.

输入: 初始解 $S_{initial}$, 局部搜索最大迭代次数 T_r , 触发扰动的迭代次数 T_p ;

输出: 发现的最优解 S_{best} .

1. $S_{best} \leftarrow S_{initial}$
2. $S \leftarrow S_{initial}$
3. $Iter \leftarrow 0$
4. $Noimprove \leftarrow 0$
5. **while** $Iter < T_r$ **do**
6. **if** 小于概率值 p **then**
7. 构造当前解 S 的邻域空间 N_1
8. 在 N_1 中根据 TSBMS 策略选择未被 tabu 禁止的最优解 S'
9. $S \leftarrow S'$
10. **else**
11. 构造当前解 S 的邻域空间 N_2
12. 在 N_2 中根据 TSBMS 策略选择未被 tabu 禁止的最优解 S'
13. $S \leftarrow S'$
14. **if** $f(S) > f(S_{best})$ **then**
15. $S_{best} \leftarrow S$
16. $Noimprove \leftarrow 0$
17. **else**
18. $Noimprove \leftarrow Noimprove + 1$
19. $Iter \leftarrow Iter + 1$

```

20. if  $Noimprove > T_p$  then
21.    $Noimprove \leftarrow 0$ 
22.    $S \leftarrow Perturb(S)$ 
23. return  $S_{best}$ 

```

IRLTS 算法搜索过程与第 2 节 RLTS 算法的搜索过程相同, 只是在选择邻居解时采用的搜索策略存在差异, 体现在算法 5 的第 8 行和第 12 行. RLTS 算法在解空间 N_1 和 N_2 中选择最优邻居解时是通过遍历搜索的方法, 在面对大规模测试用例造成的巨大的解空间时, 该方法存在单次迭代过于耗费时间的缺陷. 为此在 IRLTS 的局部搜索过程 (算法 5) 中, 我们利用 TSBMS 策略替代了遍历搜索的方法 (第 8 行和第 12 行). 由于 TSBMS 策略可以根据解空间的大小采用不同的策略选择最优邻居解, 因此 TSBMS 策略不仅大大提高了 RLTS 算法在处理大规模数据时的表现, 同时也保持了 RLTS 算法在面对小规模数据时的优秀表现.

3.4 局部搜索算法

对于启发式算法的分类, 学术界存在多种不同的分类标准. 从搜索的对象, 也就是解的数量这个角度来划分, 启发式算法通常可以被划分为基于种群的启发式算法和基于单一解的启发式算法. 遗传算法、粒子群算法等以多个个体的集合为起点, 搜索过程作用在一个种群上的算法属于基于种群的启发式算法. 在搜索过程中, 算法始终只作用在一个解上, 这种算法属于基于单一解的启发式算法, 本文提出的 IRLTS 局部搜索算法属于基于单一解的启发式算法这一类别.

不同于基于种群的启发式算法具有固定的搜索范式, 如遗传算法的父代选择、交叉变异, 粒子群算法的演化公式等, 局部搜索算法的设计具有较高的灵活性, 研究者通常针对问题特性来设计与问题契合度较高的特定局部搜索算法. 此外, 局部搜索算法的实现也较为简单, 这也加速了算法的推广与使用. 在近几年的现实应用方面, 局部搜索算法在处理一些大规模复杂问题上取得出色的求解表现, 例如支配集问题^[22], 调度问题^[23], 顶点覆盖问题^[24]等. 迅速发展的局部搜索算法在运筹学、统计学等多个学科发挥着突出的作用.

4 实验分析

第 4.1 节, 采用 3 个当前表现最优的启发式算法作为 IRLTS 算法的对比算法, 74 组经典的图测试用例被用来评估 IRLTS 算法的有效性. 第 4.2 节, 通过实验分析了一度顶点规则和 TSBMS 策略的有效性. 第 4.3 节, 分析了 TSBMS 策略中的关键参数对 IRLTS 算法的影响.

4.1 IRLTS 算法与当前最优启发式算法对比

当前求解 MLCP 问题表现最好的 3 个算法分别是 RLTS 算法^[1], GVNS 算法^[12], MS-GVNS 算法^[12], 这 3 个算法的源码链接都在相关论文中被提供, 本文不再赘述. 为了公平地比较 IRLTS 算法与这 3 个对比算法, 本文在同一台主机上运行 4 个算法, 主机的实验环境为 Ubuntu 18.04 版本的操作系统, RAM 为 60 GB, 处理器型号为 Intel Xeon E5-2690v4 2.6 GHz. 4 个算法的程序语言均为 C++. 本文采用的 74 组经典图测试用例 (测试用例下载链接: <http://lcs.ios.ac.cn/~caisw/graphs.html>) 常被用于各类图着色问题^[15,25]、最大割问题^[26]、最大二分问题^[27]、顶点平分问题^[28]等多种图组合优化问题, 是应用较为广泛的一类图测试用例, 其顶点数目从几千到几万不等. 4 个算法的参数通过调参软件 F-race^[1]在 15 组随机选取的测试用例上获取, 具体参数值如表 1 所示. 除上述 3 个表现最好的启发式算法外, 本文同样实现了标准的粒子群算法和遗传算法这两种元启发式算法作为对比算法, 但现有研究中缺少以两种算法为基础, 并针对 MLCP 问题优化的算法, 因此标准版的两个算法在对比中表现不佳. 限于篇幅, 本文对两个算法的实验结果不再展示.

在 74 组测试用例上, 每一组测试用例都被每个算法运行 10 次, 单次运行的截止时间为 1800 s, 这与文献 [1] 中的运行次数以及运行截止时间是一致的. 本文统计 10 次运行所发现的最优值, 平均值以及获取最优值的平均运行时间, 具体数据如附录 A 表 A1 所示. 在表 A1 中, Graph 表示测试用例的名字, |V| 表示测试用例的顶点个数. 对

于每一个算法, 本文展示该算法在相应测试用例上获取的最优值 (best), 平均值 (avg), 获取最优值所用的平均时间 (time (s)). 粗体字表示 4 个算法的结果在对比中占优势的数值. 为更清晰地获取 IRLTS 算法与其他 3 个对比算法的对比结果, 表 2 总结了 IRLTS 算法与 3 个对比算法的对比结果. 第 1 列表示 IRLTS 算法与 3 个对比算法, 第 2 列表示总共 74 组测试用例, 第 3 列表示最优值和平均值两个对比指标, 第 4–6 列表示 IRLTS 算法与对比算法在对比中占优的测试用例个数, 持平的测试用例个数和占劣势的测试用例个数.

表 1 4 个算法的参数值

分组	参数	参数值
RLTS算法	α	0.2
	β	0.2
	γ	0.2
	T_r	50 000
	T_p	500
	k	0.1
	tt	15
GVNS算法	$k_{\max}$	$\beta \times n/2$
	α	0.5
	β	0.2
MS-GVNS算法	$k_{\max}$	$\beta \times n/2$
	α	0.5
	β	0.2
IRLTS算法	α	0.2
	β	0.2
	γ	0.2
	T_r	50 000
	T_p	500
	k	0.1
	tt	15
	$maxT$	1 000 000
	m	10 000

表 2 IRLTS 算法与其他算法的对比结果总结

对比算法	测试用例总个数	对比指标	胜	平	负
IRLTS vs. GVNS	74	best	37	28	9
		avg	33	25	16
IRLTS vs. MS-GVNS	74	best	38	28	10
		avg	32	24	18
IRLTS vs. RLTS	74	best	45	28	1
		avg	58	13	3

从表 2 可以发现, 相较于 GVNS 算法, IRLTS 算法在 37 组测试用例的最优值对比中取得优势, 28 组测试用例的最优值对比中持平, 9 组测试用例的最优值对比中相较 GVNS 算法占劣势. 在平均值指标上, IRLTS 算法占优势、持平、占劣势的测试用例个数分别为 33、25、16. 与 MS-GVNS 算法相比, 在最优值指标上, IRLTS 算法在 38 组测试用例上占优势, 在 28 组测试用例上持平, 在 10 组测试用例上占劣势, 在平均值指标上, IRLTS 算法占优势、持平、占劣势的测试用例个数分别为 32、24 和 18. 与 RLTS 算法相比, 在最优值指标上, IRLTS 算法占优势、持平、占劣势的测试用例个数分别为 45、28 和 1, 在平均值指标上, IRLTS 算法占优势、持平、占劣势的测试用例个数分别为 58、13 和 3. 实验结果表明, 无论最优解还是平均解, 改进后的 IRLTS 算法在大多数测试用例上明显超过当前最优的 3 个算法. 如前文所述, 得益于一度顶点规则和 TSBMS 策略, IRLTS 算法的解空间被减小, 搜索

邻居解的效率更高, 在两个策略的共同作用下, IRLTS 算法在测试用例上取得更好的求解表现.

4.2 所提策略有效性分析

本文提出了两个策略来优化 RLTS 算法, 分别是一度顶点规则和 TSBMS 策略. 本节将通过实验来分析这两条策略的有效性.

一度顶点规则主要是通过约简测试用例中度为 1 的顶点来达到缩减数据规模的目的. 在 74 组测试用例中, 应用一度顶点规则后顶点数目减少的测试用例有 70 组. 被约简的顶点数目与测试用例原有顶点数目的比率被用来表示约简的效果, 比值越大说明测试用例被约简的顶点数越多. 本文统计了 70 组测试用例被约简的比率范围, 具体结果如图 2 所示. 从图 2 可以发现, 有 48 组测试用例被约简的比率小于 0.1, 约简比率处于区间 (0.1, 0.3] 中的测试用例个数为 10, 约简比率处于区间 (0.3, 0.5] 中的测试用例个数为 7, 约简比率大于 0.5 的测试用例有 5 个.

为验证 TSBMS 策略的有效性, 本文随机选择 10 个测试用例来完成 IRLTS 算法的消融实验. IRLTS 算法在移除 TSBMS 策略后就是 RLTS 算法. 因此, 本文在随机选择的 10 个测试用例上运行 IRLTS 算法和 RLTS 算法, 通过对比两个算法的实验结果来验证 TSBMS 策略的有效性. 在表 3 中可以发现, 相较于移除 TSBMS 策略的 RLTS 算法, 使用 TSBMS 策略的 IRLTS 算法在 7 组测试用例的最优值对比中取得优势, 8 组测试用例的平均值对比中取得优势, 3 组测试用例的最优值对比中处于持平, 2 组测试用例的平均值对比中处于持平. 实验结果显示 IRLTS 算法在移除 TSBMS 策略后求解表现明显下降, 这说明 TSBMS 策略可以有效地提升 IRLTS 算法的搜索表现.

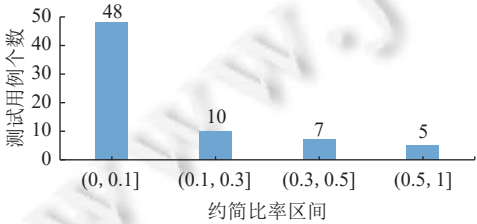


图 2 一度顶点规则的约简效果统计

表 3 TSBMS 策略的有效性分析实验结果

测试用例	RLTS		IRLTS	
	best	avg	best	avg
scc_fb-messages	156317	156116	156435	156159
scc_rt_assad	30	30	30	30
scc_rt_obama	2	2	2	2
scc_rt_bahrain	48	47.9	49	48.8
scc_rt_saudi	30	29.2	30	30
scc_rt_mittromney	43	42.7	48	47.9
scc_rt_gmanews	372	367.5	372	372
scc_rt_lolgop	1668	1658.5	1670	1666
scc_retweet	23238	23011.6	23509	23429.3
sc-nasasrb	330305	329221	642004	633975

4.3 参数影响分析

本文提出的 IRLTS 算法主要是在 RLTS 算法的基础上添加 TSBMS 策略得到的, TSBMS 策略中的两个关键的参数分别是临界值 $maxT$ 和采样个数 $bmsLen$. 本节利用随机选取的 10 组测试用例来分析两个关键参数对 IRLTS 算法的影响. 在每组测试用例上, 不同参数值的 IRLTS 算法被运行 10 次, 运行的截止时间是 1800 s, 算法 10 次运行所产生的最优值和平均值被计算. 本文统计不同参数值的 IRLTS 算法在 10 组测试用例的最优值和平均值对比中占优势的测试用例个数, 以此来衡量参数值对 IRLTS 算法的影响.

图 3 展示了参数值 $maxT$ 对 IRLTS 算法的影响. 图中的 x 轴表示参数 $maxT$ 的不同取值, y 轴表示取不同 $maxT$ 值的 IRLTS 算法在 10 组测试用例的最优值和平均值对比中占优势的测试用例个数. 如图 3 所示, 当 $maxT$ 的值为 1000000 时, IRLTS 算法在 3 组测试用例的最优值对比中取得优势, 在 5 组测试用例的平均值对比中取得优势. 当 $maxT$ 等于 2000000 和 5000000 时, 虽然 IRLTS 算法也在 3 组测试用例的最优值对比中取得优势, 但是这时 IRLTS 算法在平均值上的表现不如 $maxT$ 等于 1000000 时, 因此, 在本研究中, $maxT$ 的默认参数值为 1000000.

图 4 展示了参数值 $bmsLen$ 对 IRLTS 算法的影响. 图中的 x 轴表示参数 $bmsLen$ 的不同取值, y 轴表示 $bmsLen$ 取不同值的 IRLTS 算法在 10 组测试用例的最优值和平均值对比中取得优势的测试用例个数. 由图 4 可知, 当 $bmsLen$ 的值等于 10000 时, IRLTS 算法在 4 组测试用例的最优值对比中取得优势, 以及 3 组测试用例的平均值对

比中取得了优势. 当 $bmsLen$ 取其他值时, IRLTS 算法的表现比 $bmsLen$ 等于 10000 时 IRLTS 算法的表现差, 因此 IRLTS 算法中 $bmsLen$ 参数的默认值为 10000.

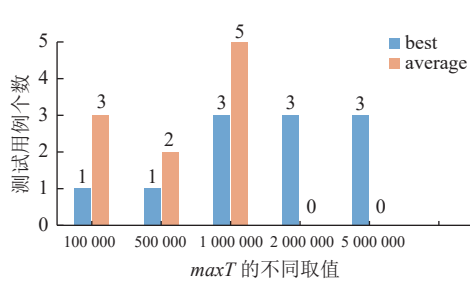


图3 临界值 $maxT$ 对 IRLTS 算法的影响

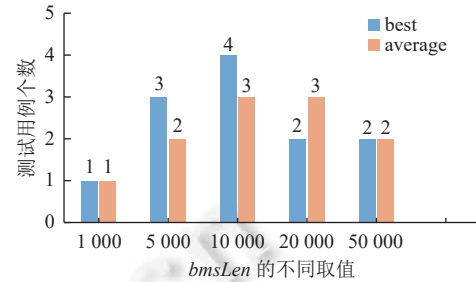


图4 采样个数 $bmsLen$ 对 IRLTS 算法的影响

5 总 结

本文针对当前求解 MLCP 问题表现最好的启发式算法存在的缺陷, 提出了两点优化策略, 一是一度顶点规则, 该规则通过约简测试用例中度为 1 的顶点来减少测试用例的顶点数, 进而降低测试用例的解空间规模, 从而有利于算法在相对较小的搜索空间内寻找最优解. 第 2 个优化策略是 TSBMS 策略. 该策略可以根据解空间规模的大小采用不同的方式在邻域空间中选择最优邻居解, 它有效地提高了算法在处理不同规模测试用例时的求解表现. 本文将两个策略优化后的 RLTS 算法命名为 IRLTS 算法. 74 组经典的测试用例和 3 个当前表现最好的启发式算法被用来验证 IRLTS 算法的有效性. 实验结果表明, 无论最优解还是平均解, IRLTS 算法在大多数测试用例上都超过了 3 个当前最优的启发式算法. 此外, 本文还组织实验验证了所提策略的有效性, 以及分析了关键参数对算法的影响. 在未来的研究中, 我们将在两个方面继续完善我们的工作, 一是对 IRLTS 算法的结构进行优化, 进一步提高 IRLTS 算法的求解表现; 二是针对更大规模的测试用例设计高效的优化策略, 为 MLCP 问题的解决提供更高效的求解算法.

References:

- [1] Sun Z, Benlic U, Li MJ, Wu QH. Reinforcement learning based tabu search for the minimum load coloring problem. Computers & Operations Research, 2022, 143: 105745. [doi: 10.1016/j.cor.2022.105745]
- [2] Ahuja N, Baltz A, Doerr B, Přivětivý A, Srivastav A. On the minimum load coloring problem. Journal of Discrete Algorithms, 2007, 5(3): 533–545. [doi: 10.1016/j.jda.2006.09.001]
- [3] Baldine I, Rouskas GN. Reconfiguration and dynamic load balancing in broadcast WDM networks. Photonic Network Communications, 1999, 1(1): 49–64. [doi: 10.1023/A:1010029100403]
- [4] Thaker D, Rouskas GN. Multi-destination communication in broadcast WDM networks: A survey. Optical Networks Magazine, 2002, 3(1): 34–44.
- [5] Gutin G, Jones M. Parameterized algorithms for load coloring problem. Information Processing Letters, 2014, 114(8): 446–449. [doi: 10.1016/j.ipl.2014.03.008]
- [6] Barbero F, Gutin G, Jones M, Sheng B. Parameterized and approximation algorithms for the load coloring problem. Algorithmica, 2017, 79(1): 211–229. [doi: 10.1007/s00453-016-0259-z]
- [7] Yi JH, Xing LN, Wang GG, Dong JY, Vasilakos AV, Alavi AH, Wang L. Behavior of crossover operators in NSGA-III for large-scale optimization problems. Information Sciences, 2020, 509: 470–487. [doi: 10.1016/j.ins.2018.10.005]
- [8] Li W, Wang GG, Alavi AH. Learning-based elephant herding optimization algorithm for solving numerical optimization problems. Knowledge-based Systems, 2020, 195: 105675. [doi: 10.1016/j.knosys.2020.105675]
- [9] Fei T, Bo W, Jin W, Liu DC. Artificial bee colony algorithm for the minimum load coloring problem. Journal of Computational and Theoretical Nanoscience, 2013, 10(9): 1968–1971. [doi: 10.1166/jctn.2013.3156]
- [10] Ye AS, Zhang ZQ, Zhou XQ, Miao F. Tabu assisted local search for the minimum load coloring problem. Journal of Computational and

Theoretical Nanoscience, 2014, 11(12): 2476–2480. [doi: 10.1166/jctn.2014.3664]

[11] Zhang ZQ, Li ZW, Qiao XB, Wang WJ. An efficient memetic algorithm for the minimum load coloring problem. *Mathematics*, 2019, 7(5): 475. [doi: 10.3390/math7050475]

[12] Herrán A, Colmenar JM, Mladenović N, Duarte A. A general variable neighborhood search approach for the minimum load coloring problem. *Optimization Letters*, 2023, 17(9): 2065–2086. [doi: 10.1007/s11590-022-01861-1]

[13] Cai SW. Balance between complexity and quality: Local search for minimum vertex cover in massive graphs. In: *Proc. of the 24th Int’l Joint Conf. on Artificial Intelligence*. Buenos Aires: AAAI, 2015. 747–753.

[14] Wang YY, Cai SW, Chen JJ, Yin MH. A fast local search algorithm for minimum weight dominating set problem on massive graphs. In: *Proc. of the 27th Int’l Joint Conf. on Artificial Intelligence*. Stockholm: AAAI, 2018. 1514–1522.

[15] Wang YY, Cai SW, Pan SW, Li XM, Yin MH. Reduction and local search for weighted graph coloring problem. In: *Proc. of the 34th AAAI Conf. on Artificial Intelligence*. New York: AAAI, 2020. 2433–2441. [doi: 10.1609/aaai.v34i03.5624]

[16] Chu Y, Liu BX, Cai SW, Luo C, You HH. An efficient local search algorithm for solving maximum edge weight clique problem in large graphs. *Journal of Combinatorial Optimization*, 2020, 39(4): 933–954. [doi: 10.1007/s10878-020-00529-9]

[17] Zhou YM, Hao JK, Duval B. Reinforcement learning based local search for grouping problems: A case study on graph coloring. *Expert Systems with Applications*, 2016, 64: 412–422. [doi: 10.1016/j.eswa.2016.07.047]

[18] Wu QH, Wang Y, Glover F. Advanced tabu search algorithms for bipartite Boolean quadratic programs guided by strategic oscillation and path relinking. *INFORMS Journal on Computing*, 2019, 32(1): 74–89. [doi: 10.1287/ijoc.2018.0871]

[19] Liu H, Zhang JY, Zhang XD, Kurniawan A, Juhana T, Ai B. Tabu-search-based pilot assignment for cell-free massive MIMO systems. *IEEE Trans. on Vehicular Technology*, 2020, 69(2): 2286–2290. [doi: 10.1109/TVT.2019.2956217]

[20] Wu QH, Hao JK. Memetic search for the max-bisection problem. *Computers & Operations Research*, 2013, 40(1): 166–179. [doi: 10.1016/j.cor.2012.06.001]

[21] Zhou YM, Duval B, Hao JK. Improving probability learning based local search for graph coloring. *Applied Soft Computing*, 2018, 65: 542–553. [doi: 10.1016/j.asoc.2018.01.027]

[22] Wang YY, Pan SW, Li CX, Yin MH. A local search algorithm with reinforcement learning based repair procedure for minimum weight independent dominating set. *Information Sciences*, 2020, 512: 533–548. [doi: 10.1016/j.ins.2019.09.059]

[23] Qin T, Peng B, Benlic U, Cheng TCE, Wang Y, Lü ZP. Iterated local search based on multi-type perturbation for single-machine earliness/tardiness scheduling. *Computers & Operations Research*, 2015, 61: 81–88. [doi: 10.1016/j.cor.2015.03.005]

[24] Cai SW, Su KL, Luo C, Sattar A. NuMVC: An efficient local search algorithm for minimum vertex cover. *Journal of Artificial Intelligence Research*, 2013, 46(1): 687–716. [doi: 10.1613/jair.3907]

[25] Lin JK, Cai SW, Luo C, Su KL. A reduction based method for coloring very large graphs. In: *Proc. of the 26th Int’l Joint Conf. on Artificial Intelligence*. Melbourne: ijcai.org, 2017. 517–523. [doi: 10.24963/ijcai.2017/73]

[26] Wu QH, Wang Y, Lü ZP. A tabu search based hybrid evolutionary algorithm for the max-cut problem. *Applied Soft Computing*, 2015, 34: 827–837. [doi: 10.1016/j.asoc.2015.04.033]

[27] Ma FD, Hao JK, Wang Y. An effective iterated tabu search for the maximum bisection problem. *Computers & Operations Research*, 2017, 81: 78–89. [doi: 10.1016/j.cor.2016.12.012]

[28] Tian XL, Ouyang DT, Sun R, Zhou HS, Zhang LM. Two efficient local search algorithms for the vertex bisection minimization problem. *Information Sciences*, 2022, 609: 153–171. [doi: 10.1016/j.ins.2022.07.106]

附录 A

表 A1 IRLTS 算法与 3 个对比算法的具体实验结果

Graph	V	GVNS			MS-GVNS			RLTS			IRLTS		
		best	avg	time (s)	best	avg	time (s)	best	avg	time (s)	best	avg	time (s)
ia-email-univ	1133	2348	2310.2	855.123	2360	2351.9	1099.54	2374	2341.7	939.629	2371	2337.6	508.084
ia-fb-messages	1266	2426	2408.5	892.697	2432	2418.8	1218.91	2447	2445.7	1023.95	2447	2445.6	879.883
web-google	1299	1368	1362.5	720.411	1369	1364	792.187	1374	1370.7	1071.02	1374	1369.7	983.929
bio-yeast	1458	906	901.3	979.732	909	901.5	1024.58	926	923.4	988.297	927	924.9	1321.1
scc_rt_voteonedirection	1833	2	2	0.110235	2	2	0.178833	2	2	6.40863	2	2	8.0546
ca-CSphd	1882	855	851.1	982.513	856	850.8	878.149	864	861.8	925.614	864	863.1	1077.01
scc_fb-messages	1899	156922	156551	959.498	156892	156717	1214.68	156317	156116	1298.93	156435	156159	1114.43

表 A1 IRLTS 算法与 3 个对比算法的具体实验结果 (续)

Graph	V	GVNS			MS-GVNS			RLTS			IRLTS		
		best	avg	time (s)	best	avg	time (s)	best	avg	time (s)	best	avg	time (s)
scc_rt_assad	2035	30	30	0.351526	30	30	0.529732	30	30	48.2563	30	30	3.89631
tech-routers-rf	2113	2994	2932.3	1247.05	3007	2977.5	953.967	3034	2989.3	1285.34	3044	2997.7	918.639
scc_rt+damascus	2962	18	18	6.79516	18	18	4.75015	18	17.9	369.144	18	18	15.6154
web-edu	3031	3205	3194.8	432.876	3207	3199	714.809	3204	3202	1307.75	3212	3205.2	805.874
scc_rt_obama	3040	2	2	0.20293	2	2	0.33525	2	2	0.034042	2	2	0.010975
scc_rt_occupy	3090	26	26	2.46877	26	26	11.1739	26	25.9	488.698	26	26	15.464
scc_rt_lebanon	3370	2	2	0.487716	2	2	0.56618	2	2	5.54838	2	2	0.048443
scc_rt_tlot	3513	4	4	0.86437	4	4	1.59226	4	4	115.462	4	4	2.48009
scc_rt_occupywallstnyc	3594	328	328	51.8201	328	327.9	14.825	328	327.4	478.287	328	328	78.8969
scc_rt_israel	3686	6	6	0.754021	6	6	1.04222	6	6	21.9806	6	6	0.439326
scc_rt_gop	3716	3	3	0.43656	3	3	0.664212	3	3	6.86956	3	3	0.014865
scc_rt_alwefaq	4157	114	114	27.389	114	114	100.64	114	113.9	787.734	114	114	45.1004
ca-GrQc	4158	6291	6228.8	1154.2	6296	6240.2	1348.2	6372	6289.2	1607.18	6457	6430.2	1238.92
scc_rt_oman	4452	5	5	0.624796	5	5	1.19285	5	5	35.0214	5	5	0.374704
scc_rt_tcot	4506	9	9	4.69063	9	9	4.85566	9	9	25.8006	9	9	1.10014
scc_rt_bahrain	4659	49	48.4	37.1457	49	48.9	161.838	48	47.9	667.571	49	48.8	637.36
scc_rt_uae	4757	6	6	1.90099	6	6	2.19519	6	6	14.7885	6	6	1.19163
web-spam	4767	15999	15776.7	1236.32	16001	15902.6	975.142	15864	15654	1618.82	15959	15896.5	967.254
scc_rt_p2	4785	7	7	0.773178	7	7	1.48663	7	7	10.8508	7	7	0.0256
inf-power	4941	3117	3096.6	1335.69	3115	3094.9	1281.2	3091	3067.9	1751.24	3272	3259.6	832.905
scc_rt_libya	5021	11	11	11.7576	11	11	8.69816	11	10.5	156.416	11	11	5.47165
scc_rt_http	5691	1	1	0.463592	1	1	1.49563	1	1	0.074111	1	1	0.030362
scc_rt_ksa	5775	8	8	19.6447	8	8	45.9051	8	7.5	399.028	8	8	10.7453
scc_rt_dash	5968	16	16	27.1349	16	16	39.2852	16	15.2	267.532	16	16	17.6321
ca-Erdos992	6100	3461	3409.7	1647.55	3460	3415	1284.71	3462	3379.6	1702.85	3556	3525.7	1364.03
socfb-MIT	6402	96166	96065.6	771.581	96224	96114.9	767.943	94824	94022.3	1401.37	96324	95782.3	1090.64
socfb-CMU	6621	97886	97421.6	1019.58	97995	97282	872.456	97673	95765.7	1508.31	98333	97793.1	1378.25
scc_rt_qatif	6718	5	5	2.56136	5	5	3.65222	5	5	97.7321	5	5	0.322374
scc_rt_saudi	6805	30	30	84.9264	30	30	45.685	30	29.2	250.653	30	30	46.1046
ia-reality	6809	3625	3442.6	1776.06	3671	3441.1	1723.1	3655	3643.5	1072.24	3709	3689.5	1136.8
scc_reality	6809	1481563	1.48E+06	376.127	1481668	1.48E+06	385.83	1481668	1.48E+06	385.83	1493535	1.49E+06	1216.98
bio-dmela	7393	9966	9791.6	1472.74	9959	9830.5	1041.24	9713	9662.1	1636.57	10080	10046.5	1073.06
tech-WHOIS	7476	24076	23807.1	919.897	24175	23882.5	749.322	23160	22886.1	1552.31	23456	23232.5	948.36
scc_rt_onedirection	7704	93	93	24.5244	93	93	7.30607	93	92.6	378.419	93	93	8.58752
scc_rt_mittromney	7850	48	48	206.992	48	48	141.92	43	42.7	321.16	48	47.9	658.531
scc_rt_gmanews	8330	372	372	143.041	372	372	412.899	372	367.5	520.333	372	372	272.653
scc_twitter-copen	8580	143861	143505	152.601	143983	143683	633.775	142625	141896	1410.88	144561	144432	835.622
scc_rt_justinbieber	9364	138	138	52.7276	138	138	88.4	138	136.7	268.388	138	138	285.221
scc_rt_barachobama	9551	87	87	90.6815	87	87	76.4077	87	84.2	506.641	87	87	11.6438
scc_rt_lolgop	9742	1670	1669.9	272.862	1670	1670	347.313	1668	1658.5	1229.04	1670	1666	1063.69
socfb-Duke14	9885	200002	199176	110.891	199939	198851	88.696	206795	202648	1796.41	207089	204951	380.834
scc_infect-dublin	10972	87096	86558.4	971.972	87085	86710	738.878	86814	86332.1	1799.55	87357	86801.4	1127.91
ca-HepPh	11204	53493	52311.1	1197.16	53272	52275.9	1050.87	46203	45664.1	1799.58	47414	46835.2	1029.91
web-indochina-2004	11358	22518	22385.7	145.895	22513	22394.9	329.941	22562	22392.6	1784.27	22868	22706.1	583.901
socfb-Stanford3	11586	218610	218222	322.056	218584	218357	451.828	215119	213164	1799.33	223509	219717	268.873
web-BerkStan	12305	9349	9331.3	1145.77	9373	9334.4	1214.15	8984	8947.7	1798.91	9438	9413.9	1603.31
socfb-UCSB37	14917	195217	191673	676.15	195356	191089	405.133	163929	161817	1799.24	196517	194156	804.854
web-webbase-2001	16062	12459	12356	1765.74	12496	12381.1	1776.32	9696	9540.7	1799.18	12465	12308.7	834.545
socfb-Uconn	17206	241198	240350	851.68	241556	240240	830.9	188970	187443	1798.94	243154	236876	432.319

表 A1 IRLTS 算法与 3 个对比算法的具体实验结果 (续)

Graph	V	GVNS			MS-GVNS			RLTS			IRLTS		
		best	avg	time (s)	best	avg	time (s)	best	avg	time (s)	best	avg	time (s)
ca-AstroPh	17903	84743	84059.1	1184.88	84718	84236.5	1380.05	70004	69017.1	1798.81	84398	83295.5	702.451
scc_retweet	18469	23929	23850.4	1153.27	23931	23871.3	963.443	23238	23011.6	1576.7	23509	23429.3	1040.42
socfb-UCLA	20453	297857	297632	1747.67	297891	297570	1696.57	228443	223907	1799.18	302939	300062	748.105
ca-CondMat	21363	40538	40401	1680.06	40582	40270.2	1702.69	29103	28916	1798.62	40624	40056.9	1416.76
socfb-Berkeley13	22900	335388	335108	1664.1	335650	334978	1739.75	241661	239034	1798.55	335001	333342	254.17
socfb-Wisconsin87	23831	332110	324438	1497.43	330792	325030	1459.98	237930	236161	1798.55	332495	329405	360.272
tech-as-caida2007	26475	22254	21579.3	1794.99	22346	21878	1787.48	15761	15562.8	1797.88	22828	22175.7	1581.78
soc-epinions	26588	42309	42029.2	1793.39	42293	42019.4	1791.28	31843	31344.9	1798.48	43448	41597.7	1027.66
socfb-Indiana	29732	525273	523775	487.74	525284	524618	640.503	348112	345554	1798.1	524038	520644	565.095
socfb-Uillinois	30795	503654	502679	515.352	503252	502384	543.957	339412	336811	1796.74	514123	507966	632.508
ia-email-EU	32430	22377	22056.4	1758.23	22177	21883.3	1759.53	16171	15996.4	1798.06	23310	22939.6	1163.54
ia-enron-large	33696	75810	73915.8	1795.45	75963	73844	1790.07	50093	49768.2	1795.14	77220	75643.4	687.118
socfb-UF	35111	572593	571216	1583.62	572109	571256	1448.53	377413	376160	1795.25	583203	575653	888.747
socfb-Texas84	36364	598625	595042	1792.26	597885	594984	1797.86	404704	402492	1792.37	597446	594262	566.696
tech-internet-as	40164	35986	34731.2	1784.82	35889	34828.3	1778.34	22663	22576.8	1796.15	36852	34969.5	1583
socfb-Penn94	41536	538265	535965	1743.82	539270	535802	1756.11	344303	342776	1783.59	547446	539845	649.422
sc-nasasrb	54870	639862	634755	1390.57	637975	631671	1789.37	330305	329221	1789.57	642004	633975	1670.38
soc-brightkite	56739	92118	90939.6	1795.72	91910	91132.3	1790.73	54401	54001.9	1759.43	95116	92361.3	1043.95



田新亮(1991—), 男, 博士, 主要研究领域为组合优化, 算法设计与分析.



蒋璐宇(1998—), 女, 博士生, CCF 学生会员, 主要研究领域为可满足性问题, 组合优化问题.



欧阳丹彤(1968—), 女, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为自动推理, 基于模型的诊断.



太然(1997—), 男, 博士生, 主要研究领域为基于模型的诊断.



周慧思(1995—), 女, 博士, CCF 专业会员, 主要研究领域为基于模型的诊断, SAT 问题.



张立明(1980—), 男, 博士, CCF 高级会员, 主要研究领域为 SAT 问题, 基于模型的诊断.