doi：10.13413/j.cnki.jdxblxb.2023165

一种大规模稀疏中国邮递员问题快速求解方法

唐继州，何丽莉，白洪涛
（吉林大学 计算机科学与技术学院，长春 130012）

摘要：针对现有中国邮递员问题求解方法在大规模稀疏路网图上求解效率的瓶颈，提出一种在可接受时间范围内求得可行解的基于蚁群优化的快速求解方法. 该方法针对 Euler 回路求解的奇偶点图上作业法的第二阶段，采用蚁群算法进行求解，同时根据大规模稀疏路网图的特性基于密度峰值聚类算法对方法进行改进：首先在蚁群算法求解前对大规模稀疏路网图进行聚类分割；其次根据邻近节点覆盖率对分割后的节点群进行合并；最后通过改变部分节点所属聚类使各节点群内部节点个数均为偶数. 实验结果表明：在奇偶点图上作业法所能支持的节点规模下，该方法可求得与确定性算法相同的最优解，并在运算时间上达到约 10 倍的效率优化；且该方法在大规模稀疏路网图下可有效提高计算效率，并在可控时间范围内得到优化的可行解，针对 5 000 个节点规模的路网图最快可在 60 s 内完成求解.

关键词：中国邮递员问题；蚁群优化；密度峰值聚类；Euler 图
中图分类号：TP391 **文献标志码**：A **文章编号**：1671-5489(2024)02-0311-09

A Fast Solution Method for Large-Scale Sparse Chinese Postman Problem

TANG Jizhou, HE Lili, BAI Hongtao
(*College of Computer Science and Technology, Jilin University, Changchun* 130012, *China*)

Abstract: Aiming at the bottleneck of solving efficiency of existing Chinese postman problem solving methods on large-scale sparse road network graph, we proposed a fast solution method based on ant colony optimization to obtain feasible solutions in an acceptable time range. This method used ant colony algorithms to solve the second stage of the odd even point graph operation method for Euler's loop solution. At the same time, we improved the method based on density peak clustering algorithm according to the characteristics of large-scale sparse road network graph. Firstly, we clustered and segmented the large-scale sparse road network graph before using the ant colony algorithm to solve the problem. Secondly, we merged the segmented node groups according to the coverage of adjacent nodes. Finally, by changing the clustering of some nodes, the number of internal nodes in each node group was even. The experimental results show that: under the node size supported by the homework method on the odd even point graph, the proposed method can obtain the same optimal solution as the deterministic algorithm and achieve the efficiency optimization of about 10 times in the operation time. The proposed method can effectively improve computational efficiency in large-scale sparse road

收稿日期：2023-05-04.

第一作者简介：唐继州(1999—)，男，汉族，硕士研究生，从事高性能计算的研究，E-mail：931515887@qq.com. **通信作者简介**：白洪涛(1975—)，男，汉族，博士，教授，从事高性能计算与机器学习的研究，E-mail：baiht@jlu.edu.cn.

基金项目：国家重点研发计划项目(批准号：2022YFF06069003).

network graphs and obtain optimized feasible solutions within a controllable time range. When facing road network graphs with a scale of 5 000 nodes, the fastest solution can be completed within 60 s.
Keywords: Chinese postmen problem; ant colony optimization; density peak clustering; Euler diagram

中国邮递员问题(Chinese postman problem, CPP)也称为弧路由问题(arc routing problem, ARP),是一个经典的组合优化问题,可描述为寻找一条从给定起点出发,遍历路网图上的所有边,然后回到起点,使总开销最小的路径. 该问题应用广泛,如邮递员送信、道路勘探、警察巡逻[1]、垃圾车收集垃圾、扫雪车清扫街道[2]、街景图摄制等[3]. 这类问题从目的和效率方面均需一条从某一起点出发,遍历区域中所有边,最后回到起点的最短路径.

CPP问题求解的传统方法以管梅谷[4]提出的奇偶点图上作业法为代表,其核心思想为在原图上添加重复边消除奇数度节点. 针对该方法第二步中最优重复边添加方案的寻找,Edmonds等[5]进一步提出了一般图上的最大权匹配算法,通过不断寻找增广路径找出一个确定的最大匹配,得到最优解. 近年来已提出了多种采用启发式方法求解CPP问题的方案:文献[6]利用Shin等[7]的编码方案对实数进行编码,采用分子规划的算法以及遗传算法进行求解;Ralphs[8]针对遗传算法求解中国邮递员问题的局限性,提出了一种边权动态混合的求解方案;于红斌等[9]引入蚂蚁算法,通过随机概率选择出行方向和最短路线激励策略,改善了常规情况下先进行奇点匹配再求回路的两步求解法,使得在设定目标的约束下可直接求出最优解. 目前的工作大多数在节点数量最多只有几百的规模下进行. 传统方法在路网图规模较小的情况下可得到最优路径. 随着路网规模的扩大,节点数量快速增长,求解时间呈指数级增长,通常需要数小时,甚至数天,难以满足工作的时效性需求. 启发式方法可有效缩短求解时长,得到满足要求的可行解,但随着路网规模的增大,可行解与最优解的偏差也随之增加.

针对现有工作在大规模路网下求解问题的瓶颈,本文基于奇偶点图上作业法提出一种结合分治与启发式思想的中国邮递员问题快速求解方法(fast solution method for large scale sparse Chinese postal problem method, FS-LSSCPP). 对于大规模路网存在的稀疏特性,该方法采用分治思想,将大规模节点群分割为多个小规模节点群[10],以降低每个节点群内部的计算规模,减少搜索空间,降低偏差;并采用启发式算法降低每个小规模节点群内部的求解时长,从而降低有效可行解的求解时间.

1 问题描述

设$G=(V,E)$表示一个无向路网图,其中$V(G)$为顶点集合,$E(G)$为边集. 任意顶点$v\in V$,表示路网图上路的交点或者起、止点. 任意边$e_{ij}\in E$表示两个顶点v_i,v_j之间的边. $w(e)$表示边e的权值,即边的长度. $V(e)$表示边e的顶点集合. 在图G上增加一条边e'_{ij},如果$V(e'_{ij})=V(e_{ij})$,且$w(e'_{ij})=w(e_{ij})$,则e'_{ij}称为e_{ij}的重复边. 图G上增加的重复边集记为E'. $G^*=(V,E^*)$称为图G的扩展多重图,其中$E^*=E\cup E'$.

给定顶点序列$v_1,v_2,\cdots,v_n$,使得$(v_i,v_{i+1})\in E(1\leqslant i<n)$,则该顶点序列$v_1,v_2,\cdots,v_n$表示点$v_1$到点$v_n$的一条路径$R_{v_1,v_n}$. $U(R)$表示路径R的边集,$L(R)$表示路径R所有边的权值总和. 对于路径R_{v_1,v_n},如果$v_1=v_n$,则该路径称为一条回路,记为R_{v_1}. 如果对图G的回路R_{v_1},$E(G)$中所有边在R_{v_1}中出现且仅出现一次,则该回路为Euler回路,记为ER_{v_1}.

中国邮递员问题可描述为:在无向连通图G上,给定起点v_1(起点可任意选择),寻找一条回路R_{v_1}使得$U(R_{v_1})=E(G)$,且$L(R_{v_1})$最小. 若G是Euler图,则$L(R_{v_1})$最小的回路一定是图G的Euler回路. 若G不是Euler图,则CPP与下述问题等价:

1) 求G的重复边集E',使得图G的扩展多重图G^*为Euler图,且$L(E')$最小;

2) 求G^*的Euler回路.

在大规模路网图上,基于奇偶点图上作业法思想,求解CPP问题的基本过程如图1所示,包含如下两个阶段:

1) 找出图G中奇数度节点集合V_{odd},构建由V_{odd}中所有节点间最短路径形成的完全图G_c[11];

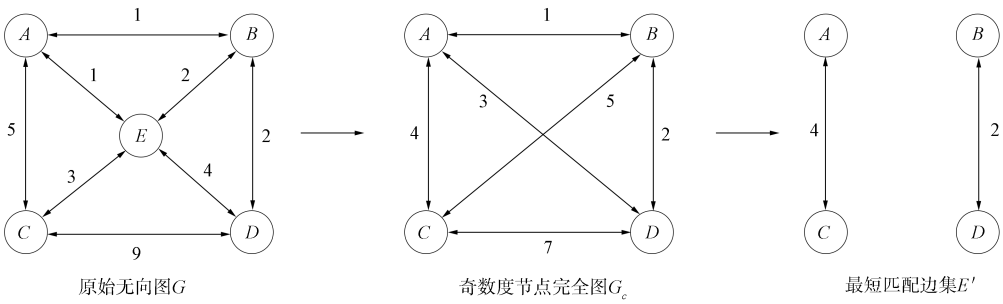


图 1　基于奇偶点图上作业法思想求解 CPP 问题的基本过程

Fig. 1　Basic process of solving CPP problems based on idea of homework method on odd even point graphs

2）在完全图 G_c 中寻找最短匹配边集 E'，即在 G_c 中寻找重复边集 E'，使得图 G 的扩展多重图 G^* 为 Euler 图，且 $L(E')$尽可能最小.

由于第二阶段的求解速度受大规模网络中奇数度顶点数量的制约，是求解整个 CPP 问题的瓶颈，因此，本文主要针对该问题进行研究，提出了 FS-LSSCPP 方法，确定大规模稀疏无向完全图的最小重复边集，从而将原始图扩展为具有最短 Euler 回路的扩展多重图.

2　FS-LSSCPP 方法框架

2.1　基本思想

针对给定路网图 G 上由奇数度节点形成的无向完全图 G_c，FS-LSSCPP 求解最短匹配边集的基本方法如下，流程如图 2 所示.

1）采用密度峰值聚类方法对奇数度节点群进行聚类分割，形成多个子节点群 V_c.

2）通过计算 V_c 中各子节点群自身邻近节点覆盖率，以覆盖率阈值为标准，合并相关节点群，形成多个合并后的子节点群 V_m.

3）计算 V_m 中节点个数为奇数的节点群之间的最小权完美匹配，并修改边界节点所属的类归属，形成节点个数均为偶数的多个子节点群 V_p.

4）在 V_p 中各节点群中分别使用蚁群算法求得各节点群的最短匹配边集 E'.

后续根据各节点群的最短匹配边集 E' 在原路图上添加重复边，形成 Euler 图 G^*，可使用 Fleury 算法在 G^* 中得到从任意起点开始的 Euler 回路.

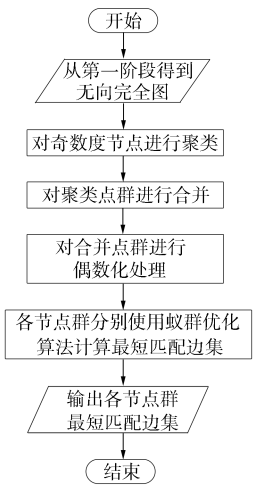


图 2　FS-LSSCPP 方法流程

Fig. 2　Flow chart of FS-LSSCPP method

2.2　奇数度节点群密度峰值聚类

为有效且合理地缩小每个节点群的规模，本文采用密度峰值聚类算法对节点群进行分割. 密度峰值聚类的流程如图 3 所示.

对奇数度节点无向完全图 G_c 中任意奇数度点 x_i，计算该点局部密度 ρ_i 和相对距离 δ_i[12]. 奇数度点 x_i 的局部密度 ρ_i 为节点周围节点疏密程度的量化标准[13]，定义为

$$\rho_i=\sum_{i\neq j}\chi(d_{ij}-d_c),\qquad \chi(x)=\begin{cases}1, & x<0,\\ 0, & x\geqslant 0,\end{cases} \tag{1}$$

其中 d_{ij} 为节点 i 与节点 j 的欧氏距离，d_c 为节点的截断距离，$\chi(x)$为截断核函数. 相对距离 δ_i 表示节点与其他密度更高的节点之间的最小距离[14]，定义为

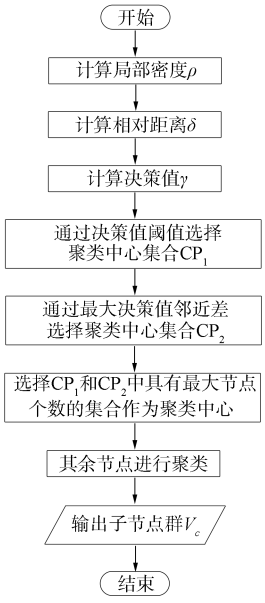


图3 FS-LSSCPP密度峰值聚类流程

Fig. 3 Flow chart of FS-LSSCPP density peak clustering

$$\delta_i=\begin{cases}\max\limits_{i\neq ij}(d_{ij}), & \rho_i=\rho_{\max},\\ \min\limits_{j:\ \rho_j>\rho_i}(d_{ij}), & \rho_i\neq\rho_{\max}.\end{cases}\tag{2}$$

为直观地根据局部密度 ρ_i 和相对距离 δ_i 判断聚类中心，使用决策值 γ_i 对两者进行结合，定义为

$$\gamma_i=\frac{\rho_i-\rho_{\min}}{\rho_{\max}-\rho_{\min}}\times\frac{\delta_i-\delta_{\min}}{\delta_{\max}-\delta_{\min}},\tag{3}$$

其中 $\rho_{\max}$,$\rho_{\min}$ 分别为所有节点局部密度的最大值和最小值，$\delta_{\max}$,$\delta_{\min}$ 分别为所有节点相对距离的最大值和最小值.

决策值 γ_i 可直观地看到每个节点作为聚类中心的特征情况，但原始密度峰值聚类算法并未给出自动选择相关聚类中心的方法，而是根据决策图人工选择，引入了一定的主观性和不确定性. 基于此，FS-LSSCPP 方法将同时使用决策值阈值 $\gamma_{\text{threshold}}$ 和最大决策值邻近差 $\gamma_{d\max}$ 自动选择聚类中心，以提高算法的可用性. 对决策值 γ_i 由大到小排序后，自动选择聚类中心的流程如下：

1) 使用决策值阈值确定聚类中心集合 $\mathrm{CP}_1=\{V_1,V_2,\cdots,V_i\}$，根据排序后的决策值 γ 序列，由前向后选择决策值 γ 超过决策值阈值 $\gamma_{\text{threshold}}$ 的节点作为聚类中心集合 CP_1，一般阈值为 0.5；

2) 使用最大决策值邻近差确定聚类中心集合 $\mathrm{CP}_2=\{V_1,V_2,\cdots,V_j\}$，根据排序后的决策值 γ 序列，由前向后选择具有最大决策值邻近差 $\gamma_{d\max}$ 的节点与其之前的所有节点作为聚类中心集合 CP_2，最大决策值邻近差 $\gamma_{d\max}$ 定义为

$$\gamma_{d\max}=\max_{\{i\geqslant 1\}\cap\{i\leqslant N-1\}}(\gamma_{i+1}-\gamma_{i-1}),\tag{4}$$

其表示节点处的决策值变化趋势.

两种选择聚类中心的方法均对同一有序中心点序列进行顺序操作，因此 CP_1 和 CP_2 具有包含关系. 选择 CP_1 和 CP_2 中节点数量较多的集合作为选定的聚类中心集合，其余节点按相对聚类中心点的距离进行聚类，形成 Num 个子节点群 $V_c=\{C_1,C_2,\cdots,C_{\text{Num}}\}$.

2.3 子节点群合并

最短匹配边集的有效性与子节点群包含的邻近节点数量有关. 子节点群内邻近节点数量过少会导致节点匹配的选择范围减小，从而大概率增加 Euler 回路路径长度[15]，为有效提高最短匹配边集的有效性，本文采用子节点群合并提高节点群中邻近节点的个数.

2.3.1 相关定义

定义 1(邻近节点集合 CN_i) 子节点群 C_i 中每个节点和其邻近节点组成的节点集合，包括子节点群 C_i 中的全部节点和子节点群 C_i 中每个节点在 G_c 中前 m 个最小权边对应的节点.

定义 2(邻近节点集合覆盖率 Coverage) e_{ij} 表示子节点群 C_i 对邻近节点集合 CN_i 中节点的覆盖程度，用公式表示为

$$\text{Coverage}_{ij}=\frac{|N_i|}{|CN_j|},\tag{5}$$

其中 N_i 为同时属于子节点群 C_i 和 CN_j 的节点集合，$|N_i|$ 为节点集合 N_i 中节点的个数，$|CN_j|$ 为邻近节点集合 CN_j 的节点个数.

定义 3(邻近节点集合覆盖率阈值) 子节点群合并的参考标准，邻近节点集合覆盖率阈值与子节点群数量满足如下关系时最短匹配边集可用性较好：

$$\text{threshold}_{\text{cov}}=\frac{1}{\mathrm{e}^{\text{Num}/10}},\tag{6}$$

其中 $\text{threshold}_{\text{cov}}$ 为邻近节点集合覆盖率阈值，Num 为聚类个数.

子节点群合并目标是使各子节点群自身邻近节点集合覆盖率均超过邻近节点集合覆盖率阈值. 子节点群内邻近节点数量 m 的选取与路网图中总奇数度节点的数量有关，一般为总奇数度节点数量的 1%[16]；不同的子节点群个数会影响覆盖率阈值的大小，较少的子节点群数量，节点分布相对集中，需要较高的邻近节点覆盖率阈值才可保证最短匹配边集的可用性；若子节点群个数很多，则节点分布相对分散，每个子节点群内部节点较少，邻近节点覆盖率阈值要求应适当降低.

2.3.2　子节点群合并

子节点群合并流程如下：

1) 计算 V_c 中各子节点群邻近节点集合和邻近节点覆盖率；

2) 合并 V_c 中相关子节点群，直到各子节点群邻近节点覆盖率均满足邻近节点覆盖率阈值.

设合并后的子节点群数量为 n 个，合并后的子节点群记为 V_m，合并流程如下：

程序 1　子节点群合并流程.

输入：子节点群 $V_c=\{C_1,C_2,\cdots,C_{\text{Num}}\}$；

输出：子节点群 $V_m=\{C_1,C_2,\cdots,C_n\}$；

1) 计算中各子节点群相邻节点覆盖率 Coverage_{ii}，Coverage_{ij}

2) for C_i in V_c do

3)　　while $\text{Coverage}_{ii}<\text{threshold}_{\text{cov}}$ do

4)　　选取下一个最大的 Coverage_{ji}

5)　　$\text{Coverage}_{ii}=\text{Coverage}_{ii}+\text{Coverage}_{ji}$

6)　　记录聚类点群 i 需要和聚类点群 j 合并

7)　end while

8) end for

9) 按照记录进行聚类合并，形成 V_m.

2.4　子节点群节点数量偶数化

V_m 中各子节点群内部节点个数可能为奇数，而最短匹配边集 E' 的寻找要求集合的节点个数为偶数，故需对节点个数为奇数的节点群进行偶数化处理才可进行后续的匹配操作.

群节点数量偶数化的过程如下：首先确定所有节点数量为奇数的子节点群的中心节点；然后确定群中心点之间的最小权完美匹配集合；最后在具有匹配关系的两个节点群之间，选择一个边界节点进行群间移动. 边界节点是指具有匹配关系的两个节点群中，与两个群中心点的距离差值最小的节点. 由于奇数度节点群的数量为偶数，所以调整后的子节点群节点数量均为偶数. 子节点群节点数量偶数化流程如下.

程序 2　子节点群节点数量偶数化流程.

输入：子节点群 $V_m=\{C_1,C_2,\cdots,C_n\}$；

输出：子节点群 $V_p=\{C_1,C_2,\cdots,C_n\}$；

1) 使用蚁群算法求 V_m 各子节点群中心节点间最小权完美匹配 E_m

2) for i in E_m do

3)　取出 i 边对应的 V_m 中的子节点群 C_a，C_b

4)　for j in C_a or C_b do

5)　　计算 L_{ja} 和 L_{jb}（j 点到 C_a，C_b 中心的距离）

6)　　找出具有 $\min(L_{ja}-L_{jb})$ 的节点 j（即边界节点）

7)　end for

8)　if $j\in C_a$ then

9)　　修改 j 所属→$j\notin C_a$，$j\in C_b$

10)　else if $j\in C_b$ then

11) 修改 j 所属→$j\notin C_b$，$j\in C_a$

12) end if

13) end for

14) 合并后子节点群为 V_p.

确立奇数度节点群中心点之间的最小权完美匹配集合的方法有很多，如果节点数量较多，可采用类似蚁群算法实现. 群节点数量偶数化的流程如图 4 所示.

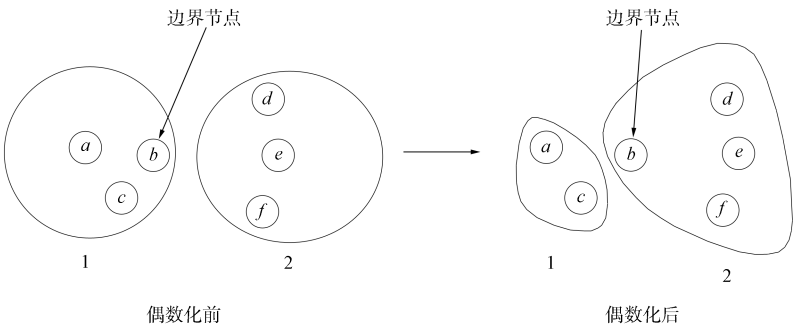


图 4 节点群数量偶数化

Fig. 4 Even number of node groups

2.5 基于蚁群算法求解子节点群内最短匹配边集

2.5.1 蚁群算法整体思路

为保证构造出的 Euler 图总路径长度最短，同时提高求解速率，本文应采用相关启发式方法进行求解. 最短匹配边集的求解可视为寻找最短路径，为同时满足上述需求，本文采用蚁群算法求解各子节点群 V_p 内部最短匹配边集 E'. V_p 中各子节点群依次使用蚁群算法进行求解时，多只蚂蚁同时进行下述操作，具体流程如图 5 所示.

1) 随机选择一个未匹配节点，并根据相关信息素和节点间路径长度计算当前节点选择其余未匹配节点的概率.

2) 根据步骤 1)中计算的概率从其余未选择节点中随机选择一个节点，两者进行匹配，若仍有节点未进行匹配则重复步骤 1)和步骤 2)，直到所有节点均已匹配完成.

3) 所有蚂蚁均完成所有节点的匹配后，从多只蚂蚁所得的匹配结果中选择具有最短添加路径长度的匹配结果，并根据此结果对相关匹配路径上的信息素进行修改.

4) 若当次迭代的最佳匹配结果优于之前迭代的最佳匹配结果，则保存当前的最佳匹配结果作为整体最佳匹配结果；若未达到最大迭代次数，则重复步骤 1)～4)，直到达到最大迭代次数.

5) 当达到最大迭代次数后，用保存的整体最佳匹配结果作为最终结果.

结合 V_p 中各子节点群的最短匹配边集和原路网图可得到从任意起点开始的最短 Euler 回路.

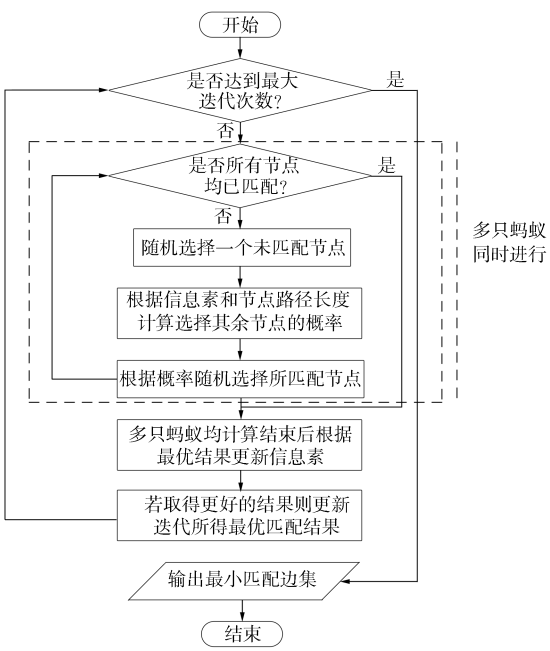


图 5 FS-LSSCPP 蚁群算法流程

Fig. 5 Flow chart of FS-LSSCPP ant colony algorithm

2.5.2 信息素更新方案

在一次迭代结束后，需根据当次迭代结果对信息素进行更新. 传统的信息素更新是每个蚂蚁对信息素更新的叠加，同时信息素的增加量和当前寻找的路径长度有关，该方案中信息素受不同长度的路径和固定的信息素总量影响，但不同长度的路径可

能会产生不同量级的信息素大小，从而使信息素可能会异常增大，导致整体收敛异常，而且所有蚂蚁均会对信息素进行更新，可用性较差的结果对信息素的更新会导致整体向错误的方向收敛，从而加大寻找可用解的难度. 因此，改进方案中采用精英蚂蚁策略，一次迭代只有路径最短的蚂蚁才更新信息素. 同时信息素的增加改为每次增加常量 P_x，从而避免信息素不同数量级的问题：

$$\tau_{ij} = \tau_{ij} + P_x, \quad ij \in MA, \tag{7}$$

其中：τ_{ij} 为节点 i 和节点 j 之间的信息素浓度；MA 为精英蚂蚁的匹配结果集；ij 为匹配结果集中的具体路径；P_x 为信息素常量，大小在 0.05～0.1 内效果较好.

3　实验结果及分析

本文在 Windows10 平台和 Visual Studio 2019 环境下进行实验，输入为奇数度节点完全图，其中所有节点均为奇数度节点，且奇数度节点完全图中各节点间路径长度均随机生成，以模拟实际路网环境中不同路径长度间的随机性，从而排除数据的特殊性导致实验结果的误差；输出为重复路径的添加长度和运行时间.

3.1　小规模 CPP 问题求解结果比较

为证明 FS-LSSCPP 算法所得匹配结果接近于具有最短路径长度的匹配结果，应将同等规模下 FS-LSSCPP 算法结果与确定性算法所得结果进行比较，但限制于确定性算法所能处理问题规模较小，故此处比较确定性算法所能求解范围内的结果，并以此推广到大规模情况. 确定性算法采用线性规划方法，在 170 个节点以下的路径长度比较结果列于表 1(单位：m)，FS-LSSCPP 算法采用 1 000 次迭代 5 次运行平均值. 由表 1 可见，在较小规模下 FS-LSSCPP 算法可得到与线性规划相同的重复路径长度，即 FS-LSSCPP 算法可得到小规模下的最优解.

表 1　不同奇数度节点个数下 FS-LSSCPP 算法和线性规划结果比较

Table 1　Comparison of FS-LSSCPP algorithm and linear programming results under different odd degree node numbers　m

算法	节点个数					
	9	20	51	96	150	170
线性规划	5 253	5 061	6 435	5 949	8 765	17 694
FS-LSSCPP	5 253	5 061	6 435	5 949	8 765	17 694

3.2　小规模 CPP 问题求解时间比较

在同等规模下 1 000 次迭代 FS-LSSCPP 算法和线性规划方法的计算时间比较如图 6 所示. 由图 6 可见，在节点数量较小时，两种算法求解所需时长均较小且差距不大. 随着节点个数的增加，FS-LSSCPP算法的效率优势显著增加. 在 170 个节点时，FS-LSSCPP 算法相比于线性规划方法有约 8 倍的提升效率. 实验结果表明，确定性算法虽然能得到最优解，但整体计算时长随着节点规模的增大而快速增长，当节点规模过大时，将无法在可接受时间内得到所需解；而 FS-LSSCPP 算法时长主要与迭代次数和单次蚁群时长有关，在节点规模上，随着节点规模的增大整体时长将会缓慢线性增加，即使在大规模情况下，也依然可在有限时间内得到可行解.

3.3　大规模 CPP 问题求解结果比较

FS-LSSCPP 算法在使用蚁群算法进行求解前，采用聚类算法针对路网图中节点的稀疏特性对原始奇数度点群进行分割. 下面给出引入聚类后的 FS-LSSCPP 算法相比于单一蚁群算法在结果可用性和整体效率上的提升.

FS-LSSCPP 算法通过聚类缩小节点群规模，有效提高运行效率，并通过聚类合并提高子节点群中邻近节点的覆盖率，从而有利于降低后续蚁群算法寻找的有效重复路径长度. 图 7 为 FS-LSSCPP 算法和单独蚁群算法在不同规模节点下运行结果的比较. 由图 7 可见，随着节点规模的增大，FS-LSSCPP算法在重复路径长度的结果上具有明显优势.

3.4　大规模 CPP 问题求解时间比较

使用聚类方法后可将大规模节点群分割为多个小规模节点群，从而缩小求解空间，提高运行效

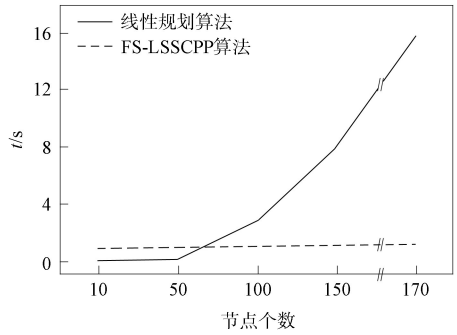


图6 两种算法运行时间比较

Fig. 6 Comparison of running time between two algorithms

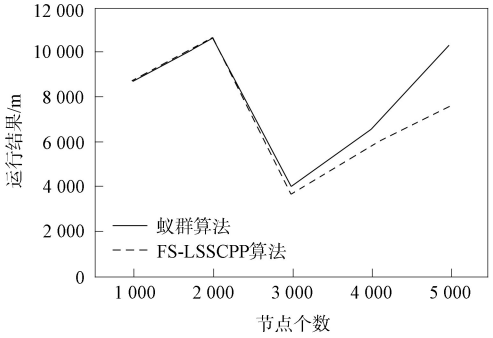


图7 蚁群算法和FS-LSSCPP算法运行结果

Fig. 7 Running results of ant colony algorithm and FS-LSSCPP algorithm

率. 图8为不同规模的节点群分别在蚁群算法和FS-LSSCPP算法下的运行效率. 由图8可见，随着节点规模的增加，FS-LSSCPP算法的计算效率优势逐渐明显.

由蚁群算法和FS-LSSCPP算法运行效率的比较可见，不同节点规模下，FS-LSSCPP算法的求解所需时间明显降低，且随着节点规模的增大，效率的提升也越来越明显.

综上所述，针对现有中国邮递员问题求解方法在大规模稀疏路网图上求解效率的瓶颈，本文结合分治和启发的思想提出了一种有效降低所需时间，同时得到迭代次数内最优Euler回路的大规模稀疏中国邮递员问题快速求解方法，通过分析、计算及实验比对，得到如下结论：

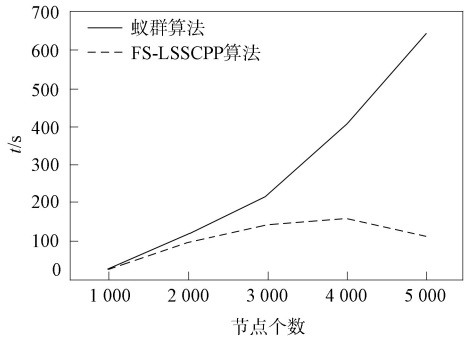


图8 蚁群算法和FS-LSSCPP算法运行效率

Fig. 8 Operation efficiency of ant colony algorithm and FS-LSSCPP algorithm

1) 在确定性算法可支持的节点规模下，本文方法可以极大概率得到最优Euler回路的同时大幅度降低所需时间；

2) 在大规模节点情况下，可在保证结果有效性的前提下有效提高运算效率，降低所需时间.

该方法主要针对无向Euler图的第二阶段构造过程进行求解，针对有向Euler图的构造方法和第一阶段奇数度节点间最短路径的快速求解[17]还有待进一步研究.

参 考 文 献

[1] WILLEMSE E J, JOUBERT J W. Applying Min-Max k Postmen Problems to the Routing of Security Guards [J]. Journal of the Operational Research Society, 2012, 63(2): 245-260.

[2] SALAZAR-AGUILAR M A, LANGEVIN A, LAPORTE G. Synchronized Arc Routing for Snow Plowing Operations [J]. Computers & Operations Research, 2012, 39(7): 1432-1440.

[3] 管梅谷. 关于中国邮递员问题研究和发展的历史回顾 [J]. 运筹学学报, 2015, 19(3): 1-7. (GUAN M G. A Historical Review of the Research and Development of the Chinese Postman Problem [J]. Journal of Operational Research, 2015, 19(3): 1-7.)

[4] 管梅谷. 奇偶点图上作业法 [J]. 数学学报, 1960(3): 263-266. (GUAN M G. Working Method on Odd and Even Point Graphs [J]. Journal of Mathematics, 1960(3): 263-266.)

[5] EDMONDS J, JOHNSON E L. Matching, Euler Tours and the Chinese Postman [J]. Mathematical Programming, 1973, 5(1): 88-124.

[6] SUN J H, MENG Y K, TAN G Z. An Integer Programming Approach for the Chinese Postman Problem with Time-Dependent Travel Time [J]. Journal of Combinatorial Optimization, 2015, 29(3): 565-588.

[7] SHIN S Y, LEE I H, KIM D, et al. Multiobjective Evolutionary Optimization of DNA Sequences for Reliable

DNA Computing [J]. IEEE Transactions on Evolutionary Computation, 2005, 9(2): 143-158.

[8] RALPHS T K. On the Mixed Chinese Postman Problem [J]. Operations Research Letters, 1993, 14(3): 123-127.

[9] 于红斌，薛占熬. 基于蚂蚁算法的中国邮路问题 [J]. 河南师范大学学报(自然科学版)，2011，39(5)：169-171.（YU H B, XUE Z A. China's Postal Route Problem Based on Ant Algorithm [J]. Journal of Henan Normal University (Natural Science Edition), 2011, 39(5): 169-171.）

[10] CORBERÁN A, PLANA I, RODRÍGUEZ-CHÍA A M, et al. A Branch and Cut Algorithm for the Maximum Benefit Chinese Postman Problem [J]. Mathematical Programming, 2013, 141: 21-48.

[11] PEARN W L, CHOU J B. Improved Solutions for the Chinese Postman Problem on Mixed Networks [J]. Computers and Operations Research, 1999, 26(8): 819-827.

[12] 徐晓，丁世飞，丁玲. 密度峰值聚类算法研究进展 [J]. 软件学报，2022，33(5)：1800-1816.（XU X, DING S F, DING L. Research Progress in Density Peak Clustering Algorithms [J]. Journal of Software Science, 2022, 33(5): 1800-1816.）

[13] 孙林，秦小营，徐久成，等. 基于 K 近邻和优化分配策略的密度峰值聚类算法 [J]. 软件学报，2022，33(4)：1390-1411.（SUN L, QIN X Y, XU J C, et al. Density Peak Clustering Algorithm Based on K-Nearest Neighbor and Optimal Allocation Strategy [J]. Journal of Software, 2022, 33(4): 1390-1411.）

[14] 吴斌，卢红丽，江惠君. 自适应密度峰值聚类算法 [J]. 计算机应用，2020，40(6)：1654-1661.（WU B, LU H L, JIANG H J. Adaptive Density Peak Clustering Algorithm [J]. Computer Applications, 2020, 40(6): 1654-1661.）

[15] 李长明，张红臣，王超，等. 一种高效的阴阳 k-Means 聚类算法 [J]. 吉林大学学报(理学版)，2021，59(6)：1455-1460.（LI C M, ZHANG H C, WANG C, et al. An Efficient Yin Yang k-Means Clustering Algorithm [J]. Journal of Jilin University (Science Edition), 2021, 59(6): 1455-1460.）

[16] 胡雅婷，陈营华，宝音巴特，等. 一种增量式 MinMax k-Means 聚类算法 [J]. 吉林大学学报(理学版)，2021，59(5)：1205-1211.（HU Y T, CHEN Y H, BAOYIN B, et al. An Incremental MinMax k-Means Clustering Algorithm [J]. Journal of Jilin University (Science Edition), 2021, 59(5): 1205-1211.）

[17] 王玉，申铉京，周昱洲，等. 一种求解交通网络中最短路径问题的人工蜂群算法 [J]. 吉林大学学报(理学版)，2021，59(5)：1144-1150.（WANG Y, SHEN X J, ZHOU Y Z, et al. An Artificial Bee Colony Algorithm for Solving the Shortest Path Problem in Traffic Networks [J]. Journal of Jilin University (Science Edition), 2021, 59(5): 1144-1150.）

（责任编辑：韩　啸）