

结合聚类分解的增强蚁群算法求解复杂绿色车辆路径问题

胡蓉^{1,2} 李洋^{1,3} 钱斌^{1,2} 金怀平¹ 向凤红¹

摘要 针对带时间窗的低能耗多车场多车型车辆路径问题 (Low-energy-consumption multi-depots heterogeneous-fleet vehicle routing problem with time windows, LMHFVPR_TW), 提出一种结合聚类分解策略的增强蚁群算法 (Enhanced ant colony optimization based on clustering decomposition, EACO_CD) 进行求解. 首先, 由于该问题具有强约束、大规模和 NP-Hard 等复杂性, 为有效控制问题的求解规模并合理引导算法在优质解区域搜索, 根据问题特点设计两种基于 K-means 的聚类策略, 将 LMHFVPR_TW 合理分解为一系列带时间窗的低能耗单车场单车型车辆路径子问题 (Low-energy-consumption vehicle routing problem with time windows, LVRP_TW); 其次, 本文提出一种增强蚁群算法 (Enhanced ant colony optimization, EACO) 求解分解后的各子问题 (LVRP_TW), 进而获得原问题的解. EACO 不仅引入信息素挥发系数控制因子进一步动态调节信息素挥发系数, 从而有效控制信息素的挥发以提高算法的全局搜索能力, 而且设计基于 4 种变邻域操作的两阶段变邻域局部搜索 (Two-stage variable neighborhood search, TVNS) 来增强算法的局部搜索能力. 最后, 在不同规模问题上的仿真和对比实验验证了所提 EACO_CD 的有效性.

关键词 低能耗车辆路径问题, 多车场多车型, 时间窗, 聚类分解, 增强蚁群算法

引用格式 胡蓉, 李洋, 钱斌, 金怀平, 向凤红. 结合聚类分解的增强蚁群算法求解复杂绿色车辆路径问题. 自动化学报, 2022, 48(12): 3006-3023

DOI 10.16383/j.aas.c190872

An Enhanced Ant Colony Optimization Combined With Clustering Decomposition for Solving Complex Green Vehicle Routing Problem

HU Rong^{1,2} LI Yang^{1,3} QIAN Bin^{1,2} JIN Huai-Ping¹ XIANG Feng-Hong¹

Abstract In this paper, an enhanced ant colony optimization combined with clustering decomposition strategy (EACO_CD) is proposed for solving the low-energy-consumption multi-depots heterogeneous-fleet vehicle routing problem with time windows (LMHFVPR_TW). Firstly, since the considered problem is a complex one with strong constraints, large scale and NP-hardness, In order to control the scale of problem and reasonably guide the algorithm to search in the high-quality solution region, two kinds of clustering methods based on K-means strategies are designed to reasonably decompose it into a series of subproblems (i.e., the low-energy-consumption vehicle routing problems with time windows (LVRP_TW) by utilizing the problem characteristics. Secondly, an enhanced ant colony optimization (EACO) to solve the decomposed subproblems (LVRP_TW) for obtaining the solution of the original problem is proposed. Not only a control factor of pheromone decay parameter in EACO is added to adjust the pheromone decay parameter dynamically so as to control the volatilization of pheromone effectively and improve the global search ability of ACO, but also a two-stage variable neighborhood search (TVNS) is designed based on four variable neighborhood operations to enhance its the local search ability. Finally, simulation experiments and comparisons on instances with different scales demonstrate the effectiveness of proposed EACO_CD.

Key words Low-energy-consumption vehicle routing problem (LVRP), multiple depots and heterogeneous fleet, time windows, cluster of decomposition, enhanced ant colony optimization (EACO)

Citation Hu Rong, Li Yang, Qian Bin, Jin Huai-Ping, Xiang Feng-Hong. An enhanced ant colony optimization combined with clustering decomposition for solving complex green vehicle routing problem. *Acta Automatica Sinica*, 2022, 48(12): 3006-3023

收稿日期 2019-12-22 录用日期 2020-05-03

Manuscript received December 22, 2019; accepted May 3, 2020
国家自然科学基金 (61963022, 51665025), 云南省应用基础研究
计划重点项目 (202201AS070030) 资助

Supported by National Natural Science Foundation of China
(61963022, 51665025) and Applied Basic Research Key Project of
Yunnan Province (202201AS070030)

本文责任编辑 王鼎

Recommended by Associate Editor WANG Ding

1. 昆明理工大学信息工程与自动化学院 昆明 650500 2. 昆明理工大学云南省人工智能重点实验室 昆明 650500 3. 云南电力技术有限责任公司昆明 650051

1. School of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500
2. Yunnan Key Laboratory of Artificial Intelligence, Kunming University of Science and Technology, Kunming 650500 3. Yunnan Electric Power Technology Co., Ltd., Kunming 650051

传统的车辆路径问题 (vehicle routing problem, VRP) 由 Dantzig 和 Ramser 于 1959 年首次提出^[1]. 该问题主要描述为在满足车辆载重、容积、行驶里程及客户服务要求等条件的同时, 合理调度车辆出行数量、行车路线、出行时间, 使得总运费用最优化. 随着社会经济的高速发展, 跨区域多车队联合车辆配送需求明显增加^[2-4]. 同时, 当今环保要求越来越严格, 考虑燃油消耗和碳排放等因素的低能耗车辆配送已开始受到重视^[5]. 此外, 日益激烈的市场竞争逼迫企业注重降低物流配送成本和提高客户满意度^[6]. 在此背景下, 研究带时间窗的低能耗多车场多车型车辆路径问题 (Low-energy-consumption multi-depots heterogeneous-fleet vehicle routing problem with time windows, LMHFVPR_TW), 具有十分重要的现实意义. 由于 VRP 为 NP-hard 问题, 而 VRP 可归约为 LMHFVPR_TW, 故 LMHFVPR_TW 也属于 NP-hard 问题, 对其开展研究亦具有较大理论价值.

低能耗车辆路径问题近年已开始受到重视, 但相关研究仍较为有限. 在低能耗多车场车辆路径问题 (Low-energy-consumption multi-depots VRP, LMVRP) 方面, Jabir 等^[7] 在问题中考虑碳排放等因素, 利用蚁群算法与变邻域搜索 (Variable neighborhood search, VNS) 相结合进行求解. 仿真实验表明, 该算法可有效求解小规模和大规模问题. Kaabachi 等^[8] 在问题中考虑距离、碳排放和燃油消耗等因素, 并在蚁群算法中加入插入、交换和 2-opt 等邻域搜索策略以提高算法局部搜索能力, 取得良好的效果. 在低能耗多车型车辆路径问题 (Low-energy-consumption heterogeneous-fleet VRP, LHFVRP) 方面, Xiao 等^[9] 在问题中考虑交通拥堵和碳排放等因素, 并设计融合变邻域搜索和整数规划的迭代搜索算法进行求解. Kwon 等^[10] 在问题中考虑碳排放交易机制 (即各国之间通过碳排放交易市场相互买卖碳排放量, 以保证该国的碳排放量在其规定的配额内), 然后采用混合禁忌搜索算法进行求解. 目前尚无 LMHFVPR_TW 的相关研究.

对于多车场车辆路径问题 (Multi-depot VRP, MVRP)、多车型车辆路径问题 (Heterogeneous-fleet VRP, HFVRP) 和多车场多车型车辆路径问题 (Multi-depot heterogeneous-fleet VRP, MHFVRP) 这几类复杂 VRP, 智能算法已有一定的研究. 现有研究大多对问题进行整体编码和求解, 但由于这些问题比单车场单车型 VRP (即传统 VRP) 具有更多的决策变量和约束条件, 使得解空间扩大且编码解码较为繁琐, 这时仅靠智能算法本身已难

以将搜索快速引导至目标值整体较优的不同区域 (即较优区域) 执行, 容易导致算法实际搜索效率下降. 因此, 近年来部分学者在所提算法中, 先采用某种分解策略将问题合理分解为多个相对简单的子问题, 从而将算法搜索尽量限定于解空间中部分存在优质解的较优区域之内, 然后再用智能算法等求解各子问题并获得原问题的解, 取得良好效果. 在 MVRP 方面, 主要考虑地理条件、客户需求和交货时间等因素, 并采用聚类算法把 MVRP 分解为一系列单个车场 VRP, 然后采用粒子群优化 (Particle swarm optimization, PSO) 算法^[11-12]、遗传算法^[13] 等求解. 在 HFVRP 方面, 主要考虑车辆载重量、车辆运输费用和客户需求等因素, 并采用路径分割法把 HFVRP 分解为一系列单车型 VRP (即旅行商问题 (Traveling salesman problem, TSP)), 进而采用禁忌搜索算法^[14] 等求解. 在 MHFVRP 方面, 由于该问题复杂度高, 为 HFVRP 和 MVRP 的综合, 故大都先采用分解策略将 MHFVRP 分解为一系列 HFVRP (即单车场异构车辆的 VRP), 然后采用路径分割法将各 HFVRP 进一步分解为多个 TSP. 譬如, Dondo 等^[15] 先采用一种启发式聚类算法将全部客户分成若干类, 然后利用路径分割法分配车辆, 从而将原问题转化为多个小规模 TSP, 然后再利用混合整数规划求解器进行求解. Tang 等^[16] 先利用最近邻方法把客户聚类到相应的车场, 然后再用扫描法对每个车场中的客户进行路径分割, 进而将原问题转化为多个 TSP, 最后采用蚁群算法求解. 上述文献中的仿真实验和算法对比验证了解析策略和智能算法结合的必要性. 然而, 对于现实物流配送中大量存在且更为复杂的 LMHFVPR_TW, 尚无结合分解策略的智能求解算法, 故开展相关研究意义重大.

蚁群算法 (Ant colony optimization, ACO) 是一种模拟蚂蚁寻找食物的群智能优化算法, 最早由 Dorigo 等^[17] 提出并首次用于求解 TSP. ACO 借鉴蚂蚁觅食的群体智能行为, 利用信息素浓度矩阵 (以下简称信息素矩阵) 学习和保留蚂蚁在各自行驶路径 (即问题解) 上留下的信息素浓度信息, 并基于该矩阵构建路径转移概率矩阵 (以下简称概率矩阵), 然后通过对概率矩阵采样生成新种群来实现路径搜索 (即解空间搜索) 并引导搜索方向. ACO 采用正反馈机制使得较短行驶路径 (较优解) 的优良信息得以更多积累, 可引导算法在较短时间内到达解空间中存在优质解的区域. 这使得算法具有较强的全局搜索能力, 从而在多种 VRP 上得到成功应用^[7-8, 18-24]. 从 ACO 在 VRP 系列问题上的研究现状

来看, 采用合理的信息素浓度计算策略可控制全局搜索的方向, 引入基于有效邻域操作的局部搜索能进一步提高算法性能, 两者是设计有效 ACO 的关键. 文献调研表明, ACO 求解 LMHFVPR_TW 的研究目前处于空白状态.

本文研究 LMHFVPR_TW 的建模与求解. 在建模方面, 给出包含经济费用、燃油消耗费用和客户满意度费用的运输总费用计算模型, 并建立以最小化运输总费用为优化目标的 LMHFVPR_TW. 在求解方面, 考虑到 LMHFVPR_TW 这类问题的解空间庞大且整体编码解码复杂, 直接采用智能算法难以在较短时间内实现有效搜索, 故提出一种结合聚类分解的增强蚁群算法 (Enhanced ant colony optimization based on clustering decomposition, EACO_CD) 进行求解.

EACO_CD 的结构如图 1 所示 (以 2 个车场

2 类车型为例). 由图 1 可知, EACO_CD 由问题分解阶段和子问题求解阶段组成. 1) 在问题分解阶段, 为确保分解后的各子问题区域尽量覆盖解空间中的优质解区域, 设计两层基于 K-means 的聚类算法对问题进行逐步分解. 第 1 层聚类算法为所提的一种改进平衡 K-means 聚类算法 (Improved balanced K-means algorithm, IBKA), 用于给每个车场分配一定数量的客户, 从而形成一系列带时间窗的低能耗单车场多车型 VRP (LHFVRP with time windows, LHFVRP_TW); 第 2 层聚类算法为所提的一种带粒子群优化的混合 K-means 聚类算法 (Hybrid K-means clustering algorithm with PSO, HKMA), 用于给每个 LHFVRP_TW 中的每类车型合理分配一定数量的客户, 进而得到一系列带时间窗的低能耗 VRP (Low-energy-consumption VRP with time windows, LVRP_TW). 两层聚类

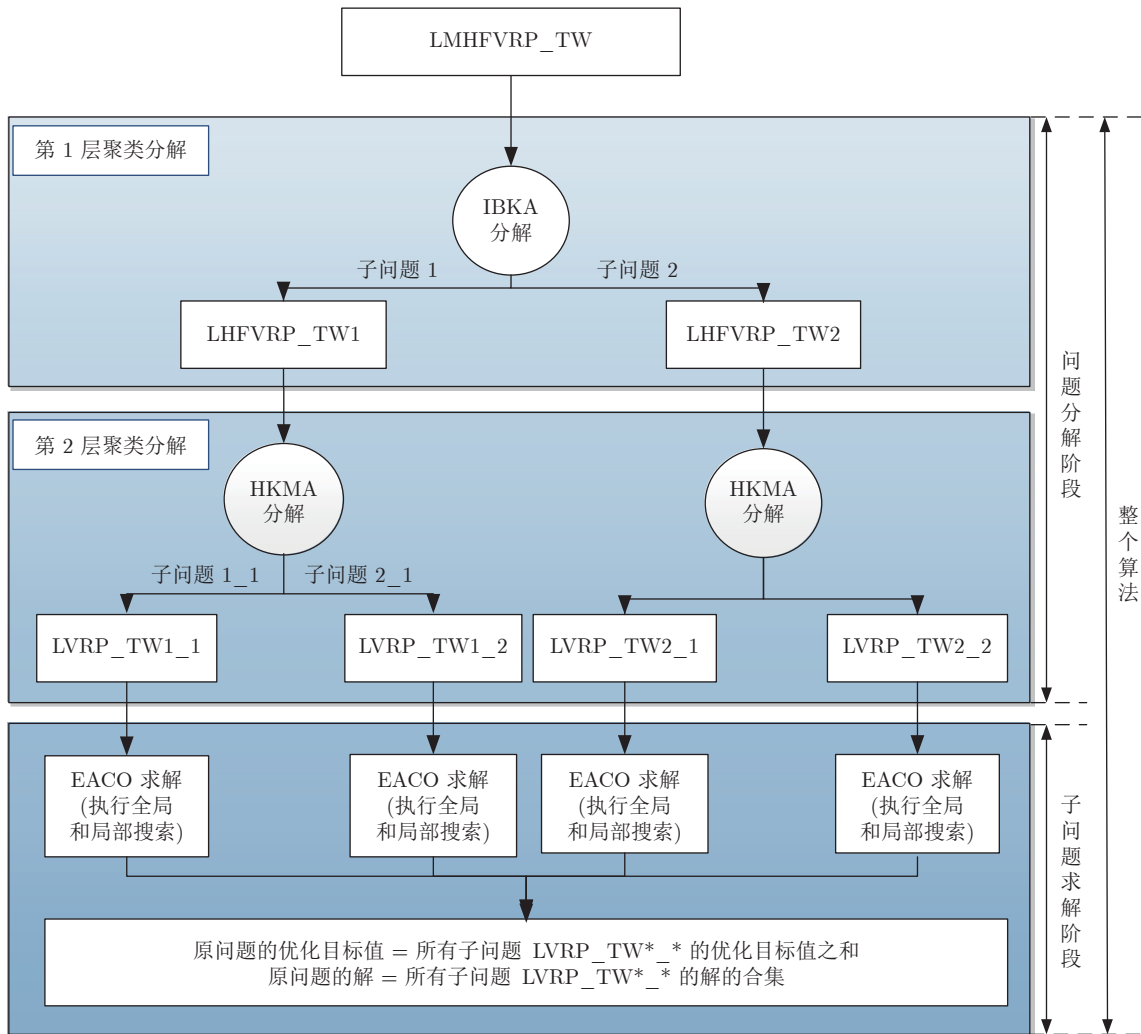


图 1 EACO_CD (EACO_IBKA_HKMA) 结构

Fig.1 Framework of EACO_CD (EACO_IBKA_HKMA)

分解算法将原问题最终分解为子问题 LVRP_TW, 而不是分解为比 LVRP_TW 规模更小的带时间窗的低能耗 TSP, 这使其子问题的解空间区域相对较大, 有利于算法对原问题解空间中更多区域进行搜索, 可望获得更好的解. 2) 在子问题求解阶段, 采用所提的增强蚁群算法 (Enhanced ant colony optimization, EACO) 对每个分解后子问题 (即 LVRP_TW) 的解空间区域进行搜索, 进而将子问题的解合并后得到原问题的解. 在 EACO 中, 采用动态的信息素浓度挥发系数 (以下简称信息素挥发系数), 并进一步加入信息素挥发系数控制因子以调节其取值, 可避免算法过早收敛, 并进一步引导算法全局搜索到达更多的不同区域; 同时, 设计基于 4 种变邻域操作的两阶段局部搜索策略, 用于对全局搜索发现的优质解区域进行细致且高效的搜索, 以平衡全局和局部搜索并增强算法性能. 最后, 通过仿真实验和算法比较验证所提算法的有效性.

1 LMHFVPR_TW 建模与分析

本节建立 LMHFVPR_TW 模型并对问题特点和求解算法进行分析. LMHFVPR_TW 为带时间窗约束的多车场多车型车辆配送问题, 优化目标为最小化运输总费用 (即经济费用、燃油消耗费用和顾客满意度费用之和). 该问题在已知客户位置坐标、车场位置坐标、客户货物需求量、客户时间窗、

各车场车型特征 (表 1 中给出相应的符号定义) 的情况下, 希望找到各车场中车辆的最佳配送路线 (由第 1.3 节问题模型中的决策变量具体取值确定), 从而使总运输费用达到最优. LMHFVPR_TW 的应用场景很多, 譬如电商和连锁超市的商品配送、战区和疫区的紧急物资配送、石化企业成品油的一次和二次配送等.

1.1 问题假设和符号定义

LMHFVPR_TW 满足如下假设:

- 1) 每个车场当中的每一类车型都参与配送任务;
- 2) 每个车场所含的车型种类相同;
- 3) 车辆在一次任务分配中仅进行一次配送;
- 5) 车辆从车场出发, 完成服务后需回到原车场;
- 6) 每个客户都被服务, 被服务的次数有且仅有一次;
- 7) 车辆配送中的货物需尽量在每个客户要求的到达时间段内送达.

LMHFVPR_TW 的相关符号定义如表 1 所示.

1.2 问题优化目标计算模型

LMHFVPR_TW 的优化目标为运输总费用 Z_{Total} , 该优化目标由 3 部分组成: 经济费用 $Z_{Emission-cost}$ 、排放费用 $Z_{Emission-cost}$ 及顾客满意度费用 $Z_{Customer-Satisfaction}$. 其中, 经济费用由运输距离费用 F_1 和车辆固定成本 F_2 组成, 排放费用设定为燃油消耗费用 F_3 , 顾客满

表 1 符号及定义
Table 1 Symbols and definitions

符号	释义	符号	释义
F_1	运输距离费用	H_{PM}	车场 P 中有 H_{PM} 辆 M 类型的车辆
F_2	车辆固定成本	$r(A)$	完成客户子集 A 中所有客户的配送需要的最少车辆数
F_3	燃油消耗费用	N	总共有 N 个客户
F_4	时间窗惩罚费用	V	客户编号集合 $\{1, 2, \dots, N\}$ (0 表示车场)
C_{M1}	第 M 种类型车辆的距离费用系数	M_t	共有 M_t 种类型的车辆
C_{M2}	第 M 种类型车辆的固定发车费用系数	k	车辆编号
C_{M3}	第 M 种类型车辆的燃油费用系数	x_{PMijk}	车场 P 车型 M 的第 k 辆车从客户 i 到客户 j 的决策变量
C_1	配送车辆提前到达的单位惩罚费用	d_{ij}	客户 i 到客户 j 的距离
C_2	配送车辆迟到的单位惩罚费用	ET_i	客户 i 要求的最早到达时间
i	客户点 i	LT_i	客户 i 要求的最晚到达时间
j	客户点 j	S_i	客户 i 要求的卸货时间
P	$\{1, 2, \dots, P_t\}$	车场编号 q_i	客户 i 要求的货物需求量
P_s	全部车场集合	t_i	车辆到达客户 i 的时间
P_t	总共有 P_t 个车场 P	M	车型编号
M_s	全部车型集合 $\{1, 2, \dots, M_t\}$	Q_M	第 M 种车型的最大载重量
H_{PMs}	车场 P 中车型 M 的全部车辆集合 $\{1, 2, \dots, H_{PM}\}$	FUM_{ij}	车型为 M 的车辆从客户 i 到客户 j 之间的耗油量

注: 综合燃油消耗模型中的其他相关参数设定参考文献 [25].

意度费用由软时间窗惩罚费用 F_4 构成.

$$F_1 = \sum_{p=1}^{P_t} \sum_{M=1}^{M_t} \sum_{k=1}^{H_{PM}} \sum_{i=0}^N \sum_{j=0}^N C_{M1} \times x_{PMijk} \times d_{ij} \quad (1)$$

$$F_2 = \sum_{p=1}^{P_t} \sum_{M=1}^{M_t} \sum_{k=1}^{H_{PM}} C_{M2} \times x_{PMijk} \quad (2)$$

$$F_3 = \sum_{p=1}^{P_t} \sum_{M=1}^{M_t} \sum_{k=1}^{H_{PM}} \sum_{i=0}^N \sum_{j=0}^N C_{M2} \times FU_{Mij} \times x_{PMijk} \quad (3)$$

$$F_4 = \sum_{i=1}^N \max \{C_1 \times (ET_i - t_i), 0, C_2 \times (t_i - LT_i)\} \quad (4)$$

其中, 式 (3) 中燃油消耗量 FU_{Mij} 的计算基于文献 [10] 中的综合燃油消耗模型和参数设置, 并假设各类车型的油耗参数相同, 可得到 FU_{Mij} 的计算式为

$$FU_{Mij} = \left(\frac{\omega_1^M}{\bar{v}_{ij}^M} + \omega_2^M + \omega_3^M \times G_{ij}^M + \bar{v}_{ij}^M \times \omega_4^M \right) \times d_{ij} \quad (5)$$

其中, G_{ij}^M 为车型 M 的车辆从客户 i 到客户 j 途中的车辆总重量, \bar{v}_{ij}^M 为车型 M 的车辆从客户 i 到客户 j 的平均速度, $\omega_1^M, \omega_2^M, \omega_3^M, \omega_4^M$ 为常量. 此外, 式 (4) 中车辆到达时间 t_j 的计算为

$$t_j = \sum_{i=0}^N x_{PMijk} \times \left(t_i + \frac{d_{ij}}{\bar{v}_{ij}^M} + s_i \right) \quad (6)$$

综上, 运输总费用 Z_{Total} 的计算为

$$Z_{\text{Total}} = F_1 + F_2 + F_3 + F_4 \quad (7)$$

1.3 问题模型

问题优化目标为

$$\min Z_{\text{Total}} = F_1 + F_2 + F_3 + F_4 \quad (8)$$

问题约束条件为

$$\sum_{M=1}^{M_t} \sum_{j=1}^N \sum_{k=1}^{H_{PM}} x_{PM0jk} = \sum_{M=1}^{M_t} \sum_{j=1}^N \sum_{k=1}^{H_{PM}} x_{PMj0k}, \quad \forall P \in P_s \quad (9)$$

$$\sum_{P=1}^{P_t} \sum_{M=1}^{M_t} \sum_{i=1}^N \sum_{k=1}^{H_{PM}} x_{PMijk} = 1, \quad i \neq j, \quad \forall j \in V \setminus \{0\} \quad (10)$$

$$\sum_{P=1}^{P_t} \sum_{M=1}^{M_t} \sum_{j=1}^N \sum_{k=1}^{H_{PM}} x_{PMijk} = 1, \quad i \neq j, \quad \forall i \in V \setminus \{0\} \quad (11)$$

$$\sum_{j=1}^N x_{PM0jk} = \sum_{j=1}^N x_{PMj0k} = 1, \quad \forall P \in P_s, \quad \forall M \in M_s, \quad \forall k \in H_{PMs} \quad (12)$$

$$\sum_{P=1}^{P_t} \sum_{M=1}^{M_t} \sum_{k=1}^{H_{PM}} \sum_{i \notin A} \sum_{j \in A} x_{PMijk} \geq r(A), \quad \forall A \subseteq V \setminus \{0\}, \quad A \neq \emptyset \quad (13)$$

$$\sum_{j=1}^N \sum_{i=1}^N x_{PMijk} \times q_i \leq Q_M, \quad i \neq j, \quad \forall k \in H_{PMs}, \quad \forall P \in P_s, \quad \forall M \in M_s \quad (14)$$

决策变量为

$$x_{PMijk} = \begin{cases} 1, & \text{若车辆 } P \text{ 车型 } M \text{ 的第 } k \text{ 辆车以客户 } i \\ & \text{到客户 } j \\ 0, & \text{否则} \end{cases} \quad i \neq j, \quad (j, i) \in V, \quad \forall k \in H_{PM}, \quad \forall P \in P_s, \quad \forall M \in M_s \quad (15)$$

其中, 式 (9) 要求从某车场出发的车辆数目和回到该车场的车辆数目相等; 式 (10) 和式 (11) 要求每个客户只能被一辆车服务 1 次; 式 (12) 要求所有车辆均需参与配送, 且配送完后返回原车场; 式 (13) 要求生成的配送方案或问题解不出现不含车场的子回路; 式 (14) 要求车辆配送过程中不能超过其对应车型的额定载重量.

1.4 问题特点分析

由第 1.3 节可知, 本文的 LMHFVPR_TW 为具有非线性和大规模约束的 0-1 整数规划问题. 相对于传统 VRP, 式 (1) ~ (4) 使优化目标不同且更加复杂, 同时式 (4) 使优化目标具有非线性; 多车场和多车型的引入, 使得决策变量由传统 VRP 的 3 维变量 x_{ijk} 变为 5 维变量 x_{PMijk} , 导致决策变量和约束的数量增多, 这使得问题解空间更加庞大和复杂. 此外, 相对于常规 0-1 整数规划问题, VRP 系列问题均含子回路消除约束 (式 (13)), 该类约束的数量为 $2^{N+1} - 2$, 这种指数级的约束数量限制了问题的求解规模. 虽然可以通过增加连续决策变量, 使式 (13) 的约束个数变为客户数 N 的多项式, 从而扩大问题的求解规模, 但此时原问题近似转化为同时含离散和连续决策变量的混合 0-1 整数规划问题, 该近似问题的求解难度并未降低^[26].

1.5 问题求解算法分析

VRP 系列问题 (包括本文的 LMHFVPR_TW) 均可建模为 0-1 整数或混合整数规划模型, 其求解算法分为两类. 一类为运筹学算法, 包括分支定界、

分支切割、动态规划等^[27]. 该类算法主要以线性代数和几何分析为基本工具, 利用问题优化目标函数和约束条件式的结构信息构造遍历或部分遍历解空间的搜索, 可在几分钟至几十分钟内求解较小规模问题 (客户数小于等于 20), 同时经过合适设计, 也可用于求解较大规模问题 (客户数大于等于 80), 但往往求解时间很长^[27-29]. 不同于线性规划问题, 0-1 规划问题具有非凸特性, 其内在几何结构与最优解间的关系仍属开放问题, 目前仅在无约束 0-1 二次规划等少数具有特殊结构的简单问题上存在有效多项式时间求解算法^[6, 28, 30], 尚无通用的最优解多项式时间求解算法. 对于 LMHFVPR_TW 这类复杂的 0-1 规划问题, 采用已有的运筹学算法框架虽然也可设计相应的求解算法, 但算法难以在较短时间获取较大规模问题的满意解. 另一类是智能优化算法 (以下简称智能算法), 包括蚁群优化算法、遗传算法、禁忌搜索算法等^[27]. 该类算法不依赖于问题结构, 而是将问题的约束条件隐式地包含在所设计的解的编码、解码规则中处理, 并利用某种拟人、拟物的机制不断生成新的可行个体或解, 从而引导算法执行搜索, 往往可在几秒或十几秒内就能获得各类 VRP 的满意解^[6, 27, 31].

目前, 各类文献对所提智能算法使用少量个体 (一般种群规模为 20 ~ 100) 运行少量代数 (一般进化代数为 100 ~ 1000) 后, 为何都可获得问题的满意解, 基本都只从算法本身的机制进行解释. 这导致现有研究过于追求新方法层面的创新. 实际上, 智能算法之所以有效, 是由其编码解码规则和智能算法自身机制共同决定的. 智能算法并不直接对 0-1 变量编码, 而是对客户序号排列进行编码解码规则的设计. 这样的排序编码解码, 容易将解中元素的取值限定在满足约束的可行范围, 从而避免繁杂的约束处理, 可明显提升算法搜索效率. 同时, 排序编码解码规则所确定的解空间极为扁平, 问题目标值的变化范围远远小于排序模型解空间的规模, 这非常有利于算法短时间内获取满意解. 具体来说, 譬如优化目标为最小化总行驶距离的单车 VRP (即 TSP), 假如有 100 个客户, 任意两个客户间的距离为在区间 [1, 1000] 上均匀分布的随机数, 且要求车辆从某个客户驻地 (车场) 出发, 服务完客户后返回该驻地, 则其目标值变化范围在 (1, 1000000) 之内 (1000000 为各客户间的距离之和的上限, 实际的最大目标值小于此值), 而解空间规模为 99! (车辆出发位置的客户需扣除), 平均每一个具体目标值对应约 9×10^{149} 个不同排列或解, 这表明数量巨大的不同解具有相同的目标值. 对于其他更为复杂的 VRP (包括本文的 LMHFVPR_TW), 也存在这一情况. 另外, 离散问题本身解之间没有梯度, 相

邻解之间的目标值差别可能很小, 也可能较大. 排序编码解码规则所确定解空间的“极为扁平”性和“无梯度”性, 使得各种智能算法即使只用几秒 (CPU 为 2.6 GHz 的主流 PC 在 1s 内可搜索并评价含 100 个客户的传统 VRP 问题的数万个解) 搜索解空间中极小的区域 (数万的数量级相对于 99! 类似于 1 根针或更小的面积相对于体育场), 也可能到达较广的目标值区域, 同时其内在的寻优机制可驱动算法到达目标值较优的不同区域搜索, 从而获得满意解. 智能算法这种通过对解空间极小区域的搜索来实现对目标值较广且较优区域的搜索, 是其有效的本质原因, 也是现有运筹学算法这类偏遍历或部分遍历解空间的算法难以做到的. 因此, 设计智能算法以实现 LMHFVPR_TW 快速有效求解, 是合理且必要的.

由第 1.4 节和第 1.5 节的分析可知, 对于 LMHFVPR_TW 这类非凸、非线性问题, 直接采用运筹学方法在短时间内获取满意解难度很大, 故本文设计一种智能算法 (即结合聚类分解的增强蚁群算法 EACO_CD) 进行求解.

2 问题求解算法 EACO_CD

本文所提的求解算法 EACO_CD 包含问题分解阶段和子问题求解阶段 (见图 1). 通过两层聚类分解, 原问题 LMHFVPR_TW 转换为 $P_t \times M_t$ 个单车场单车型子问题 LVRP_TW_s, 然后采用本文设计的子问题求解算法 EACO 依次对每个 LVRP_TW 进行求解, 进而可获得原问题的解和优化目标值. 第 2.1 节介绍 EACO_CD 中问题分解阶段的细节, 包含两层聚类算法的细节和复杂度分析; 第 2.2 节介绍 EACO_CD 中子问题求解阶段的细节, 包含 EACO 的细节和复杂度分析; 第 2.3 节介绍 EACO_CD 的总体框架并分析整体复杂度.

2.1 EACO_CD 的问题分解阶段

EACO_CD 的问题分解阶段执行两层聚类算法, 第 1 层聚类算法为 IBKA, 用于将原问题 LMHFVPR_TW 分解为 P_t 个单车场子问题 LHFVPR_TW_s; 第 2 层聚类算法为 HKMA, 用于将每个单车场问题 LHFVPR_TW 进一步分解为 M_t 个单车场单车型子问题 LVRP_TW_s (见图 1). 在对数值型数据进行聚类时, 大都采用 K-means、模糊聚类 (Fuzzy cluster) 等算法^[14, 32-34]. 而针对混合型数据, 可采用 K-prototypes 算法或先将混合型数据转换为数值型数据^[35]. 由于本文问题中的客户属性数据均为值型, 不需要对数据类型进行转换, 同时, K-means 具有线性计算复杂度, 已成功应用于 VRP 系列问题^[12, 14, 33, 36-37], 故本文在 IBKA 和 HKMA 中

采用 K-means 作为基本分解策略。

2.1.1.1 第 1 层聚类算法 (IBKA)

本文问题的优化目标 (见式 (8)) 由 4 部分组成, 其中有两部分与车辆运输距离有关。当车辆距客户较远时, 配送所产生的燃油消耗费用和距离费用都随之增加 (见第 1.2 节模型), 故 IBKA 采用客户之间的欧氏距离作为评价标准, 将客户聚成与车场数目相同的几类, 使得每个车场服务其中一类客户。但是, 随着客户数量逐渐增大, 聚类后会出现每个车场服务的客户数量相差较大。为确保每个车场的客户资源能均衡分配, 设计 IBKA 进行聚类。由于文献 [37] 提出的 K-means 平衡算法仅适用于 2 类客户之间的平衡 ($P_t = 2$), 本节将文献 [37] 提出的 K-means 平衡算法推广客户群, 同时设计车场分配规则, 可更为合理地确定各车场服务的客户群。进而, 获得 P_t 个单车场子问题 LHFVRP_TW。IBKA 的步骤如下。

步骤 1. 对全体客户进行 K-means 聚类。初始化聚类重心为每个车场的坐标位置, 初始化聚类数量为车场数量, 以欧氏距离作为评价指标, 以迭代次数作为终止条件。

步骤 2. 平衡客户群。完成上述步骤后进行 k 类客户群之间的平衡计算, 具体如图 2 所示, 按每类客户群数目由大到小排序, 根据两类客户间的边缘算法^[37], 将客户较多的客户群依次向其余客户较少的客户群移动, 循环操作直至各类客户群中客户数目达到平衡。

步骤 3. 计算每个车场到各聚类重心的距离。列出距离矩阵 A , 设 A 为 $P_t \times P_t$ 的二维矩阵, 矩阵中的每一个元素表示某一车场坐标到某聚类重心的距离, 即

$$A = \begin{pmatrix} d(1, 1) & \cdots & d(1, P_t) \\ \vdots & \ddots & \vdots \\ d(P_t, 1) & \cdots & d(P_t, P_t) \end{pmatrix} \quad (16)$$

其中, $d(i, j)$ 为车场坐标 i 到聚类重心坐标 j 的欧氏距离。

步骤 4. 按设计的分配规则对车场所服务的客

户群进行分配。选取矩阵 A 中不同行 (车场不同) 不同列 (客户群重心不同) 的 P_t 个元素并相加, 共有 $P_t!$ 种不同的组合及对应的累加和 $(S_1, \dots, S_{P_t!})$, 选出其中最小的 S_{\min} (见式 (17)) 所对应的组合中每个元素 $d(i, j)$ 的行、列标号即为车场 i 所服务的客户群 j 。采用 S_{\min} 确定车场服务的客户群有利于整体减小车辆运输距离。

$$S_{\min} = \min(S_1, S_2, \dots, S_l, \dots, S_{P_t!}) \quad (17)$$

其中, S_l 表示矩阵 A 中对不同行不同列元素的第 l 种累加和。

对于 K-means 迭代次数为 $gen1_K$ 、平衡迭代次数为 gen_b 、问题规模为 N_{P_t} (客户数量_车场数量) 的问题, IBKA 的计算复杂度 (以下简称复杂度) T_{IBKA} 分别由 K-means 聚类复杂度 $O(gen1_K \times P_t \times N)$ 、客户平衡复杂度 $O(gen_b \times (P_t \times N)^2)$ (最坏情况)、获取方阵 A 的复杂度 $O(P_t^2)$ 和获取 S_{\min} 的复杂度 $O(P_t \times P_t!)$ 决定。因此,

$$T_{IBKA} = O(gen1_K \times P_t \times N + gen_b \times (P_t \times N)^2 + P_t^2 + P_t \times P_t!)$$

针对 3 车场和 144 客户的本文问题 LMHFV-PR_TW, 采用 IBKA 进行车场划分。根据列出距离矩阵 $A_{(3 \times 3)}$, 分别给出矩阵 $A_{(3 \times 3)}$ 中不同行不同列的 P_t 个元素累加和的全部组合方式, 并计算不同的累加和共 $3!$ 次 (见式 (18))。若 $S_{\min} = S_1$, 即车场 1 服务 A 类客户, 车场 2 服务 B 类客户, 车场 3 服务 C 类客户, 划分结果如图 3 所示。图 3 中上、下子图分别为未采用和采用平衡聚类的结果。从图中可以看出, 采用平衡聚类可使各车场服务的客户数量更加接近, 有利于提高各车场的整体效率和控制子问题规模。

$$\begin{aligned} S_1 &= d(1, 1) + d(2, 2) + d(3, 3) \\ S_2 &= d(1, 1) + d(2, 3) + d(3, 2) \\ &\vdots \\ S_6 &= d(1, 3) + d(2, 2) + d(3, 1) \end{aligned} \quad (18)$$

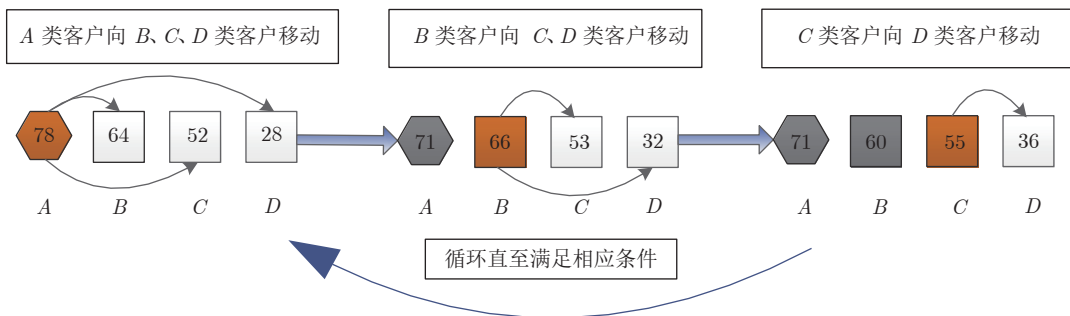
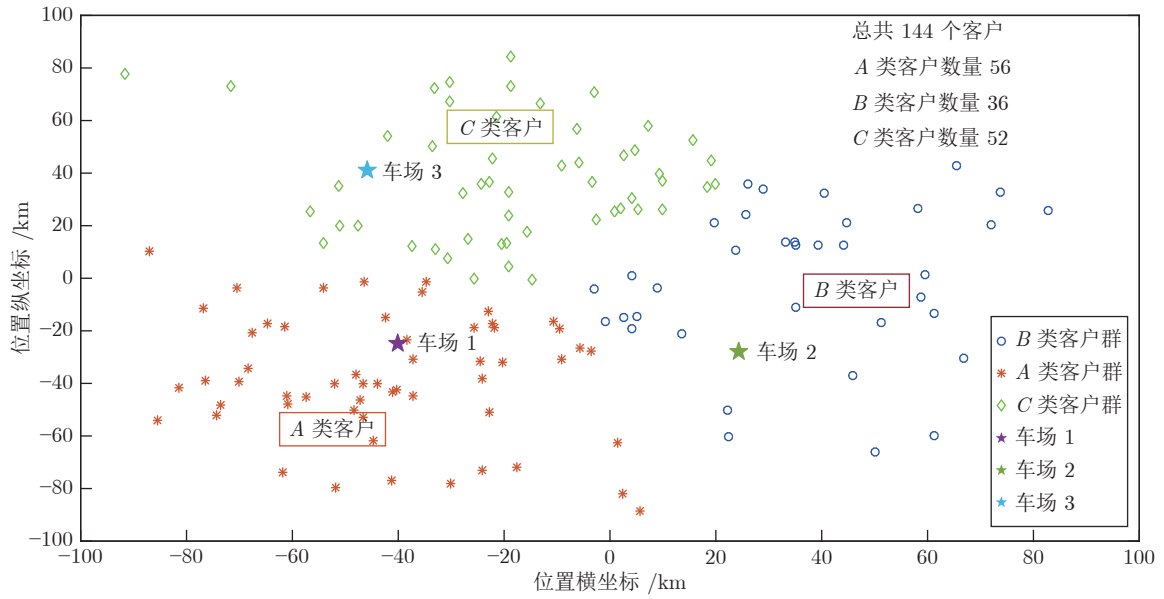
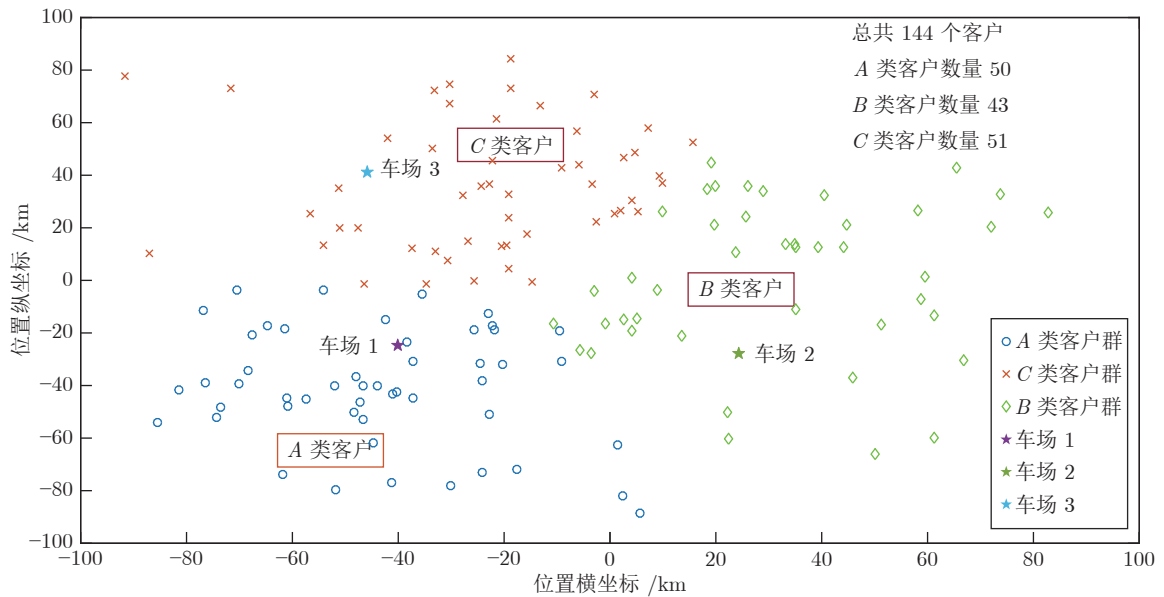


图 2 4 类客户平衡移动示意图

Fig. 2 Diagram of balanced movement for four customer groups



(a) 3 车场 K-means 未平衡聚类
 (a) Unbalanced K-means cluster of three depots



(b) 3 车场 K-means 平衡聚类
 (b) Balanced K-means cluster of three depots

图 3 3 车场 K-means 未平衡聚类与平衡聚类比较

Fig.3 Comparison of unbalanced K-means cluster and balanced K-means cluster of three depots

2.1.2 第 2 层聚类算法 (HKMA)

客户的时间窗、客户的货物需求重量、客户与车场间的距离是影响本文问题优化目标 (见式 (8)) 的重要因素, 故 HKMA 将其作为客户的 3 类属性, 采用 K-means 对客户聚类后再为各类客户分配相应的车型. K-means 聚类时是将每个客户作为一个对象 (质点). 在多属性客户进行 K-means 聚类时, 需将各属性做加权后才可进行. 然而, 上述 3 类

属性对优化目标 (见式 (8)) 的影响程度无法直接确定, 故需对各属性权重系数进行优化, 从而提高子问题分解的合理性.

HKMA 为内外两层结构, 外层执行微粒子群优化 (PSO) 算法, 用于优化属性权重系数向量 λ ($\lambda = (\lambda_1, \lambda_2, \lambda_3)$ 且 $\lambda_1 + \lambda_2 + \lambda_3 = 1$); 内层对基于每个 λ_i 的聚类分解效果进行评价, 并将评价价值 $f_i(\lambda_i)$ 作为 PSO 种群个体的适应度值. HKMA 的

工作机制如图 4 所示, 其中 PSO 种群 P_{PSO} 的每个 λ_i ($i = 1, \dots, M1$) 表示一组权重系数取值, 基于 λ_i 对单车场问题 LHFVRP_TW 做 K-means 聚类可得到 M_t 类客户群, 将 M_t 类车型和 M_t 类客户群逐一配对, 得到单车场单车型子问题 LVRP_TW (记为 $A_{i,j}$, i 和 j 分别为车型和客户群编号, $i, j = 1, \dots, M_t$), $f(A_{i,j})$ 表示对问题 $A_{i,j}$ 采用第 3.1 节的编码规则随机生成 10 个解的评价值 (即子问题总运输费用) 的平均值, F_i 为 $f(A_{i,j})$ 组成的方阵, Sum_f_k ($k = 1, \dots, M_t!$) 表示第 k 个由 F_i 中 M_t 个相互不同行 (车型不同) 不同列 (客户群不同) 元素累加所得的值, f_i_min 取所有 Sum_f_k 中的最小值并赋值给 $f_i(\lambda_i)$. 由 M_t 个车型和客户群均不同的 $A_{i,j}$ 可构成一个 LHFVRP_TW, 故 Sum_f_k 实际上是对第 k 种子问题分解方式合理性的评估, 同时 f_i_min 用于标识相对最为合理的子问题分解方式.

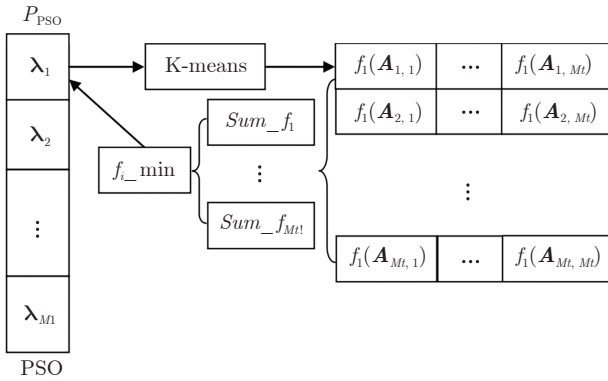


图 4 HKMA 工作机制
Fig.4 Running mechanism of HKMA

令 $Z_{Total}(A_{i,j})$ (见式 (8)) 为个体 $A_{i,j}$ 的目标函数值, λ_{opt} 为 P_{PSO} 的历史最优个体, $subP(\lambda_{opt})$ 为 $f(\lambda_{opt})$ 对应的 M_t 个单车场单车型问题 LVRP_TW 的集合 (具体见前述), $Maxgen(PSO)$ 为 HKMA 中 PSO 算法的运行代数, $gen2_K$ 为 HKMA 中每次执行 K-means 算法的运行代数. HKMA 的具体步骤如下.

步骤 1. 令 $gen = 1$, 随机生成 P_{PSO} 中的每个 λ_i .

步骤 2. (内层): 确定 P_{PSO} 中所有 λ_i 的适应度值 (具体见前述).

步骤 3. (外层): 若 $gen > 1$, 则将当代 P_{PSO} 和父代 P_{PSO} 的个体进行冒泡排序, 并择优保留前 $M1$ 个组成 P_{PSO} . 更新 λ_{opt} 和 $subP(\lambda_{opt})$.

步骤 4. (外层): P_{PSO} 通过 PSO 算法进化后得到新的种群 P_{PSO} , 也即得到 $M1$ 组新的权重系数向量.

步骤 5. 令 $gen = gen + 1$, 若 $gen \leq Maxgen(PSO)$, 则返回步骤 2, 否则输出 $subP(\lambda_{opt})$.

HKMA 的复杂度 T_{HKMA} 由步骤 2 ~ 5 的复杂度决定. 对于单车场问题 LHFVRP_TW, 问题规模为 $N_i \times M_t$ (客户数量_车型数量), 其中 $\sum_{i=1}^{P_t} N_i = N$. 步骤 2 的复杂度由 K-means 聚类复杂度 $O(gen2_K \times M_t \times N_i)$ 、获取方阵 F_i 的复杂度 $O(M_t^2 \times N_i)$ 、获取 f_i_min 的复杂度 $O(M_t \times M_t!)$ 组成, $O(M1 \times (gen2_K \times M_t \times N_i + M_t^2 \times N_i + M_t \times M_t!))$; 步骤 3 的复杂度为 $O(M1^2)$; 步骤 4 的复杂度为 $O(M1)$. 因此, $T_{HKMA} = O(Maxgen(PSO) \times (M1 \times gen2_K \times M_t \times N_i + M1 \times M_t^2 \times N_i + M1 \times M_t \times M_t! + M1^2))$. 由于需采用 HKMA 对 P_t 个单车场子问题 LHFVRP_TW 进行分解, 故第 2 层分解总的复杂度为

$$total(T_{HKMA}) = O(Maxgen(PSO) \times (M1 \times gen2_K \times M_t \times N + M1 \times M_t^2 \times N + P_t \times M1 \times M_t \times M_t! + P_t \times M1^2))$$

在本文算法 EACO_CD 中, HKMA 的参数设置为 $M1 = 10, M2 = 10, Maxgen(PSO) = 10$ 和 $gen2_K = 20$, PSO 中的惯性权重和加速因子的设置与文献 [38] 相同, 具体设置为惯性权重 $\omega = 0.9$, 速度因子 $c_1 = 1.2, c_2 = 1.5$. 针对第 2.1 节中示例问题的车场 1 和其服务的 A 类客户 (即 IBKA 划分得到一个单车场子问题 LHFVRP_TW), 采用 HKMA 进行车型划分, 可得图 5 和图 6. 由图 5 和图 6 可见, A 类客户依据车型和客户的 3 类属性得以进一步划分, 从而将 1 个 LHFVRP_TW 分解为规模更小的 2 个单车场单车型子问题 LVRP_TW 的.

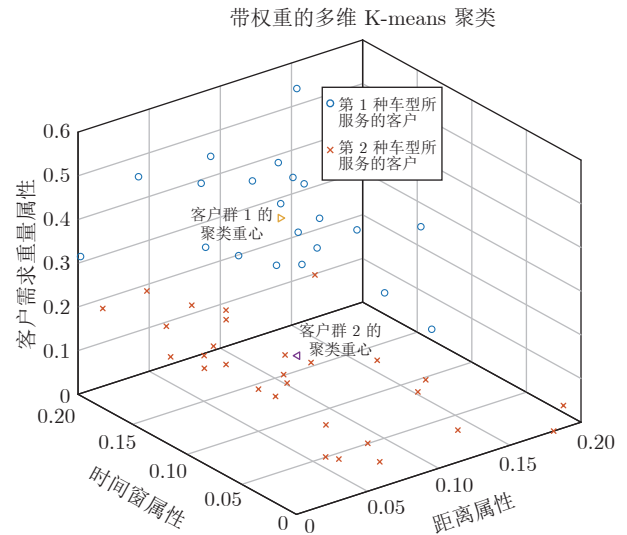


图 5 HKMA 三维聚类效果
Fig.5 The 3D clustering results of HKMA

2.2 EACO_CD 的子问题求解阶段

通过上一阶段的两层聚类分解后, 原问题 LM-

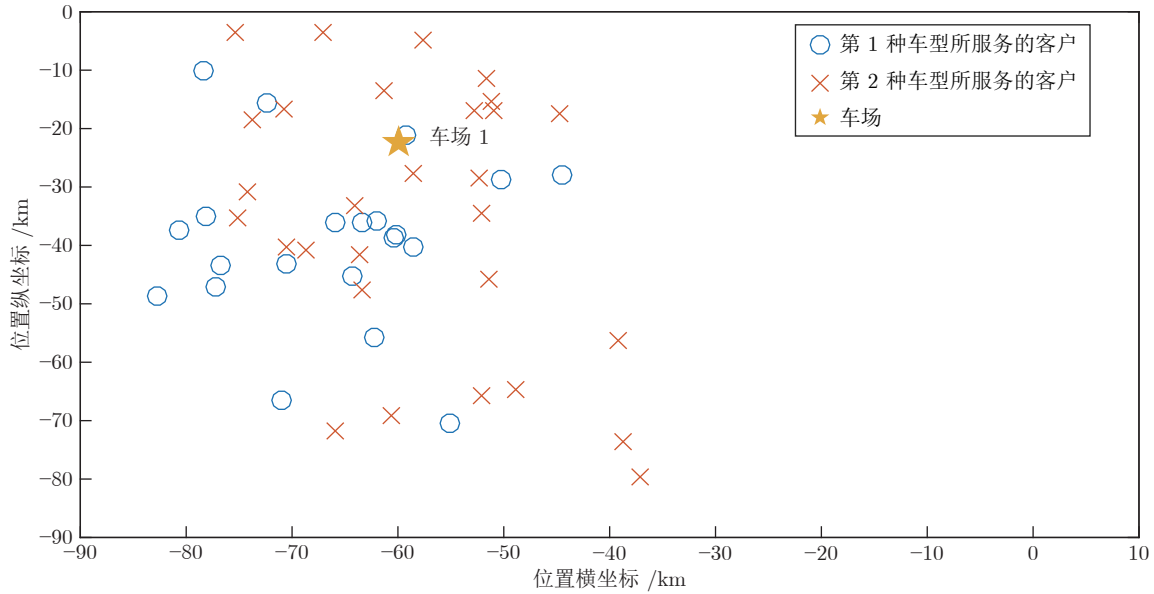


图 6 HKMA 二维结果

Fig.6 The 2D results of HKMA

HFVPR_TW 转换为 $P_t \times M_t$ 个单车场单车型子问题 LVRP_TW_s, 从而本阶段可采用本文设计的子问题求解算法 EACO 依次对每个 LVRP_TW 进行求解, 进而可获得原问题的解和优化目标值 (见图 1). 第 2.2.1 ~ 2.2.4 节介绍 EACO 的细节, 第 2.2.5 节给出 EACO 的复杂度分析.

2.2.1 编码与解码规则

EACO 对各车辆服务的所有客户进行整体编码. 每只蚂蚁代表多辆车, 每只蚂蚁的行驶路径 (即子问题 LVRP_TW 的 l 个解) 为对应各车辆的总行驶路径. 譬如对于第 l 只蚂蚁服务 10 个客户, 可记为: $\pi_{ant(l)} = [1, 4, 9, 5, 10, 2, 3, 6, 7, 8]$.

EACO 采用文献 [39] 中的方法对每只蚂蚁的行驶路径进行解码, 即逐一选择车场中的每辆车, 将 $\pi_{ant(l)}$ 中客户从左向右依次加入当前车辆, 如果当前车辆的载重量大于其服务客户的总需求量, 则继续往该车辆中加入客户, 否则选择下一辆车继续依次加入剩余客户. 譬如, 对于上一段的 $\pi_{ant(l)}$, 假设需两辆车进行服务, 解码后可得 $\pi_{ant(l)} = [\pi_{vehicle(1)}, \pi_{vehicle(2)}] = [1, 4, 9, 5], [10, 2, 3, 6, 7, 8]$.

显然, 当采用上述编解码规则时, 式 (13) 的子回路消除约束自然得到满足, 同时也不违反其他约束.

2.2.2 全局搜索

2.2.2.1 初始化

首先, 利用扫描法 (Sweep algorithm, SWA) 构造 1 只蚂蚁的行驶路径^[40], 并采用该路径对信息

素浓度 τ_{ij} 进行初始化

$$\tau_{ij} = \begin{cases} P_m, & \text{若在由 SWA 所构造的路径中} \\ 1, & \text{否则} \end{cases} \quad (19)$$

其中, i, j 为客户编号, P_m 为信息素浓度值的初始值 (设置为大于 1 的值). 为防止算法过早陷入局部最优^[41], 设置信息素浓度 τ_{ij} 的最大值和最小值为

$$\tau_{ij_min} = \frac{Q_m}{2 \sum_{i=1}^N d_{Di}}, \quad 0 < \tau_{ij_min} < 1 \quad (20)$$

$$\tau_{ij_max} = \frac{Q_m}{\sum_{i=1}^N d_{Di}}, \quad \tau_{ij_max} > P_m \quad (21)$$

其中, N 为车场需服务的全部客户总数, d_{Di} 为车场到客户 i 的距离, Q_m 为常数 ($\sum_{i=1}^N d_{Di} < Q_m < 2 \sum_{i=1}^N d_{Di}$).

然后, 设置信息素浓度增量 $\Delta\tau_{ij} = 0$ 以及种群大小 $popsize = (2/3) \times N$.

2.2.2.2 蚂蚁行驶路径搜索

EACO 每代通过对当前种群中各蚂蚁行驶路径的确定来实现对问题解空间的全局搜索. 每只蚂蚁均利用式 (22) 确定其行驶路径.

$$P_{iu}^l = \begin{cases} \frac{(\tau_{iu})^\alpha \times (\eta_{iu})^\beta}{\sum_{s \notin tabu} ((\tau_{is})^\alpha \times (\eta_{is})^\beta)}, & \text{若 } u \notin tabu \\ 0, & \text{否则} \end{cases} \quad (22)$$

其中, i 为当前客户, u 为下一客户, P_{iu}^l 为第 l 只蚂蚁从 i 到 u 的转移概率, s 为蚂蚁 l 还未服务过的客户, $\tau_{iu}(\tau_{is})$ 为 i 和 $u(s)$ 间的信息素浓度, $\eta_{iu}(\eta_{is})$ 为 i 和 $u(s)$ 之间距离的倒数, α 和 β 为重要程度权重, $tabu$ 为蚂蚁已经搜索过的客户地点集合. 对 $\{P_{iu}^l | u \notin tabu\}$ 进行轮盘赌选择, 即可得到 u 的取值, 从而确定蚂蚁 l 从客户 i 出发需到达的下一客户.

2.2.2.3 信息素浓度更新

EACO 每代对每只蚂蚁的信息素浓度采用式 (23) ~ (26) 依次进行更新

$$\gamma = \begin{cases} \gamma \left(1 - \frac{M_a}{T}\right), & M_a \text{ 迭代周期内, } L_b \text{ 未改变} \\ \gamma \left(1 + \frac{M_b}{T}\right), & M_a \text{ 迭代周期内, } L_b \text{ 已改变} \\ \gamma, & \text{否则} \end{cases} \quad (23)$$

$$\rho^{(t+1)} = \begin{cases} \gamma \rho^t, & \text{若 } \rho_{\min} \leq \gamma \times \rho^t \leq \rho_{\max} \\ \rho_{\min} \text{ 或 } \rho_{\max}, & \text{否则} \end{cases} \quad (24)$$

$$\Delta\tau_{ij}^t = \begin{cases} \frac{W}{L_b}, & \text{若 } \text{edge}(i, j) \in L_b \\ 0, & \text{否则} \end{cases} \quad (25)$$

$$\tau_{ij}^{(t+1)} = \rho^t \times \tau_{ij}^t + \Delta\tau_{ij}^t, \quad \rho^t \in [\rho_{\min}, \rho_{\max}] \quad (26)$$

其中, γ 为信息素挥发系数控制因子 (初始设置为 1), M_a 和 M_b 为迭代次数系数 (均设置为 5), T 为基准值 (设置为 50), L_b 为当前最好解对应的目标函数值, t 表示算法进化的代数 (即算法第 t 次迭

代), ρ^t 为第 t 代的信息素挥发系数 (初始设置为 0.9), ρ_{\min} 和 ρ_{\max} 分别为 ρ^t 的最小值和最大值 (分别设置为 0 和 1), $\Delta\tau_{ij}^t$ 为第 t 代的信息素浓度增量, W 为信息素增量常数 (设置为 500), τ_{ij}^t 为第 t 代的信息素浓度. γ 根据算法每代的实际运行效果动态调整其取值, 可在运行效果好 (差) 时增加 (减小) 信息素浓度, 有利于增强算法合理引导搜索的能力.

2.2.3 局部搜索

本节在基本变邻域搜索 (VNS)^[42] 基础上提出一种两阶段变邻域局部搜索策略 (Two-stage variable neighborhood search, TVNS), 对 EACO 每代发现的当前最好解进一步执行高效且细致的邻域搜索, 以提高解的质量.

TVNS 为在车辆间和车辆内部分别执行多种邻域操作 (见图 7) 的两阶段局部搜索, 两个阶段中各邻域操作均在当前最好解上执行且每次执行的最大次数均设置为 $gen_{local} = 20$. 首先, 第 1 阶段随机选择两辆车, 依次对车辆间的客户执行 “Insert” 和 “Exchange” 邻域操作, 如果在执行操作期间发现更好解, 则更新当前最好解并立刻进入第 2 阶段, 否则各邻域操作均执行至最大次数 gen_{local} 后进入第 2 阶段. 然后, 第 2 阶段逐一选择每辆车, 对每辆车的内部客户依次执行 “2-opt”、“Insert”、“Exchange” 和 “Swap” 邻域操作, 如果在执行操作期间发现更好解, 则更新当前最好解并立刻对下一辆车执行操作, 否则各邻域操作均执行至最大次数 gen_{local} 后再对下一辆车执行操作. 第 2 阶段结束后完成本次局部搜索.

2.2.4 EACO 终止条件

设定终止条件为算法的运行时间, 如果满足运

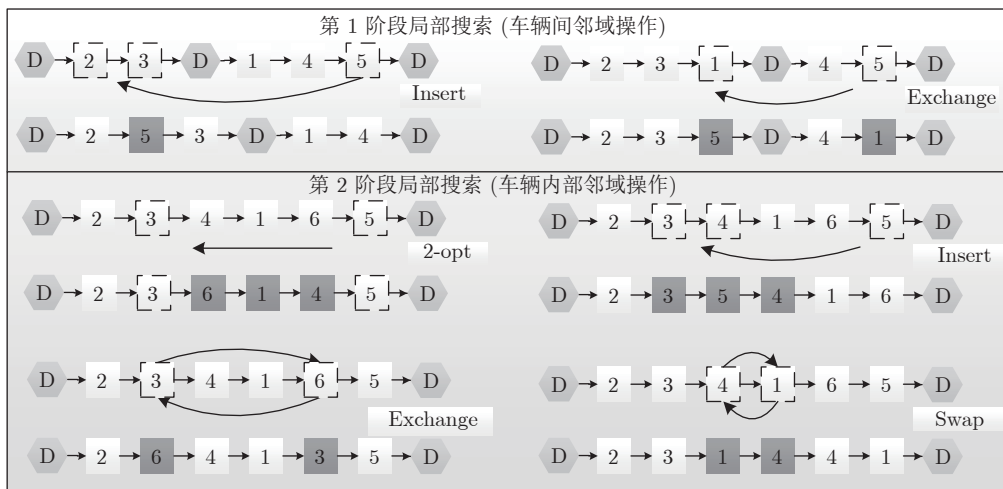


图 7 局部搜索策略

Fig.7 Local search strategy

行时间要求, 则输出服务所有客户的每辆车的行驶路径以及相应的行驶总费用。

2.2.5 EACO 复杂度分析

EACO 为 EACO_CD 中求解由聚类分解 (见第 2.1 节) 得到的单车场单车型子问题 LVRP_TW 的算法。令 EACO 种群规模为 $popsiz_E$, 全局搜索运行代数数为 gen_{global} , 局部搜索每代迭代次数为 gen_{local} , 局部搜索 (含第 1 阶段和第 2 阶段) 执行时实际使用各邻域的平均总次数为 Z (即 $Z \leq 2 + 4 \times \max\{H_{PM} | P = 1, 2, \dots, P_t, M = 1, 2, \dots, M_t\}$), 第 i 个车场的 LVRP_TW 的客户数量为 $N_{i,x}$ ($\sum_{x=1}^{M_t} N_{i,x} = N_i$, $\sum_{i=1}^{P_t} \sum_{x=1}^{M_t} N_{i,x} = N$)。EACO 的算法复杂度 T_{EACO} 由 EACO 的全局搜索复杂度和局部搜索复杂度构成。在每一代中, 全局搜索复杂度由种群初始化复杂度 $O(popsiz_E \times N_{i,x})$ 、信息素矩阵和概率矩阵更新复杂度 $O(N_{i,x}^2)$ 、采样概率矩阵生成蚂蚁路径 (即个体) 的复杂度 $O(popsiz_E \times N_{i,x}^2)$ 、种群评价复杂度 $O(popsiz_E \times N_{i,x})$ 组成, 即为 $O(popsiz_E \times N_{i,x}^2)$; 局部搜索复杂度为 $O(gen_{local} \times Z \times N_{i,x})$ (最坏情况)。因此, $T_{EACO} = O(gen_{global} \times (popsiz_E \times N_{i,x}^2 + gen_{local} \times Z \times N_{i,x}))$ 。由于需采用 EACO 对 $P_t \times M_t$ 个单车场单车型子问题 LVRP_TW_s 进行求解, 故采用 EACO 求解所有 LVRP_TW_s 总的复杂度

$$total(T_{EACO}) = O(gen_{global} \times \left(popsiz_E \times \sum_{i=1}^{P_t} \sum_{x=1}^{M_t} N_{i,x}^2 + gen_{local} \times Z \times \sum_{i=1}^{P_t} \sum_{x=1}^{M_t} N_{i,x} \right) = O(gen_{global} \times (popsiz_E \times \widetilde{\log N} \times N + gen_{local} \times Z \times N))$$

其中, $\widetilde{\log N}$ 表示低于 N 的亚线性复杂度。

2.3 EACO_CD 整体结构及复杂度分析

2.3.1 EACO_CD 整体结构

算法 EACO_CD 的整体结构如图 1 所示 (以 2 个车场 2 类车型为例)。由图 1 可知, EACO_CD 对 $P_t \times M_t$ 个子问题 LVRP_TW_s 依次执行 EACO 进行求解, 最终将获得的各子问题解合并, 得到原问题的解, 同时将各子问题解的优化目标值累加, 得到原问题的优化目标值, 从而实现对原问题的求解。

2.3.2 EACO_CD 复杂度分析

EACO_CD 是对本文问题 LMHFVPR_TW 的整体求解, 其复杂度 T_{EACO_CD} 由分解算法 IBKA 的复杂度 T_{IBKA} (见第 2.1.1 节)、分解算法 HKMA 总的复杂度 $total(T_{HKMA})$ (见第 2.1.2 节)、求解算法

EACO 总的复杂度 $total(T_{EACO})$ 组成 (见第 2.2.5 节)。即

$$T_{EACO_CD} = T_{IBKA} + total(T_{HKMA}) + total(T_{EACO})$$

虽然 T_{IBKA} , $total(T_{HKMA})$ 和 $total(T_{EACO})$ 中的变量较多, 但除 N (客户数量)、 P_t (车场数量)、 M_t (车型数量) 与问题规模相关, Z (局部搜索每次迭代搜索的邻域或个体) 与具体问题相关, $popsiz_E$ 设置为 $2/3 \times N$ (见第 2.2.2.1 节) 以外, 其余变量均与算法相关且实际使用中都被设置为常数。因此, 可得简化式如下:

$$T_{IBKA} = O((P_t \times N)^2 + P_t \times P_t!)$$

$$total(T_{HKMA}) = O(M_t^2 \times N + P_t \times M_t \times M_t!)$$

$$total(T_{EACO}) = O(\widetilde{\log N} \times N^2 + Z \times N)$$

显然, 所有变量的多项式次数不超过 2, 影响 T_{EACO_CD} 的是 T_{IBKA} 中的 $O(P_t!)$ 和 $total(T_{HKMA})$ 中的 $O(P_t \times M_t!)$ 。然而, 实际问题中 P_t 和 M_t 一般都不大于 10, 而若仅计算 10!, 用 CPU 为 2.6 GHz 的主流 PC 可在 1 s 左右完成计算, 故 EACO_CD 对大多数实际问题可在较短时间内结束运算。另外, 如果 $P_t(M_t)$ 很大 (譬如为 50), 可将 IBKA 的步骤 4 (HKMA 的步骤 2) 中穷举不同行不同列的组合改为采样 $P_t(M_t)$ 的多项式次数的组合, 从而扩大 EACO_CD 求解问题的规模。

3 实验比较与分析

本节所有算法的测试均采用以下软硬件配置: 英特尔 I7 处理器 (3.2 GHz), 8 GB 内存, Win10 操作系统, MATLAB2018a 编程环境。

3.1 EACO 参数设定

在第 1 节所提出的目标函数中 (见式 (7)), 距离费用系数设定参考文献 [7], 车辆固定费用参考文献 [9], 燃油费用系数设定参考文献 [43], 时间窗惩罚费用系数参考文献 [16], 具体取值见表 2。

表 2 目标函数中的相关系数
Table 2 Coefficients in the object function

符号	数值
C_{M1}	1.5 (元/km)
C_{M2}	300 ~ 800 (元/辆)
C_{M3}	7.6 (元/l)
C_1	15 (元/h)
C_2	20 (元/h)

EACO 的残留信息素重要程度 α , 启发信息素重要程度 β , 初始化信息素浓度参数 P_m , 信息素增量常数 W 为 4 个主要的参数, 为确定合适的参数组

合,对这4个参数进行正交实验,各参数设置水平如表3所示.由于聚类分解后原问题 LMHFVPR_TW 转变为 $P_t \times M_t$ 个 LVRP_TW_s,故采用 Solomon 标准 VRPTW 数据集中的问题 c101 (100 个客户)进行测试.每组参数组合下的 EACO 在 c101 上独立运行 20 次,取 20 次的平均最小运输总费用值作为平均响应值 ARV,参数设置的正交表见表4,参数的平均响应值和影响力见表5.由表4和表5可得,EACO的参数设置为: $\alpha = 1.25, \beta = 2.5, P_m = 1.1, W = 500$.

表3 主要参数与水平
Table 3 Main parameters and level

主要参数	水平设置			
	1	2	3	4
α	1.25	1.5	1.75	2.0
β	10	1.5	2.0	2.5
P_m	1.1	1.2	1.3	1.4
W	500	1000	1500	2000

表4 参数设置的正交表
Table 4 Orthogonal table of parameter settings

组合编号	水平设置				AVR (元)
	α	β	P_m	W	
1	1	1	1	1	9677
2	1	2	2	2	9625
3	1	3	3	3	9613
4	1	4	4	4	9541
5	2	1	2	3	9745
6	2	2	1	4	9624
7	2	3	4	1	9602
8	2	4	3	2	9593
9	3	1	3	4	9836
10	3	2	4	3	9703
11	3	3	1	2	9654
12	3	4	2	1	9612
13	4	1	4	2	9865
14	4	2	3	1	9689
15	4	3	2	4	9656
16	4	4	1	3	9672

3.2 实验设计及测试问题选取

3.2.1 实验设计

EACO_CD 由 IBKA、HKMA 和 EACO 组成.为验证 EACO_CD 的有效性,先将这3种算法分别与国际期刊上的相近算法进行比较,最后再进行总体比较.测试实验车辆参数见表6.

表5 各参数不同水平下的平均响应值和影响力
Table 5 Average response values and influences table at different levels of each parameter

水平	水平设置			
	α	β	P_m	W
1	9614	9780	9656	9645
2	9641	9660	9659	9684
3	9701	9631	9683	9683
4	9720	9604	9677	9664
极差	106	176	27	39
影响力排名	2	1	4	3

表6 4种不同车型相关参数设置
Table 6 Related parameter settings for four different vehicle types

车型列表	车型参数				
	载重量 (kg)	空车重量 (kg)	平均速度 (km/h)	固定费用 (元)	最大承载货物数 (件)
Type 1	200	1600	60 ~ 80	300 ~ 400	20
Type 2	500	2700	50 ~ 70	400 ~ 500	30
Type 3	600	3500	40 ~ 60	500 ~ 600	40
Type 4	800	5000	30 ~ 50	600 ~ 800	50

首先,为验证 EACO 的有效性,在多车场单车型问题 LMVRP_TW 上,将 EACO1 与国际期刊中的有效算法 DHACO^[37]进行对比.EACO1 为 EACO_CD 不含两层聚类的算法,而 DHACO 为不含分解且直接求解整个问题的一类蚁群算法.EACO1 采用 DHACO 的编码.测试结果见表7.

其次,为单独验证 IBKA 的有效性,在多车场单车型问题 LMVRP_TW 上,将 EACO_IBKA 与 EACO_KM、EACO_NNA 进行对比.这3种算法为在 EACO1 中分别加入3类第1层聚类算法(即 IBKA、K-means^[12,14]、NNA^[11])后得到.测试结果见表7.

然后,为单独验证 HKMA 的有效性,在单车场多车型问题 LHFVPR_TW 上,将 EACO_HKMA 和 EACO_RDA、EACO_RAA、EACO_KEW、EACO2、TSA_RDA^[10]进行对比.前4种算法为在 EACO1 中分别加入4类第2层聚类算法(即 HKMA、RDA^[10]、RAA、KEW)后得到,其中 RDA 根据客户需求量优先选择最大载重量的车型,进而将问题分解为一系列单车 LVRPs (即 TSPs),RAA 随机将客户划分给某种车型,KEW 给每类客户属性分配相同权重后用 K-means 划分车型(即简化的 HKMA),RAA 和 KEW 均将问题分解为一系列单车型 LVRP_TW_s;EACO2 为 EACO1 改用多车型编码后得到,用于直接求解整个问题;TSA_

表 7 EACO_IBKA 与其他算法对比结果
Table 7 Comparison results of EACO_IBKA with other algorithms

N_P	EACO_IBKA				EACO_KM				EACO_NNA				EACO1				DHACO				$T(s)$
	最优	平均	最差	标准差	最优	平均	最差	标准差	最优	平均	最差	标准差	最优	平均	最差	标准差	最优	平均	最差	标准差	
48_2	11118	11800	12163	95	10745	11181	11579	97	11558	12080	12539	99	9650	10390	11026	87	10255	11068	11663	90	10
96_2	17483	18155	18549	183	18037	18379	19146	187	16768	17675	18229	190	15371	17009	17771	169	16011	17559	18161	174	19
144_2	24628	25435	26983	308	24880	25369	27201	314	25435	26710	27702	320	24366	25969	27419	318	24475	26884	28356	328	29
192_2	27522	28546	29649	411	28457	29482	30261	419	28758	29560	31019	428	28379	29838	30618	432	28524	30863	31665	445	38
240_2	31505	32699	34166	508	32517	33677	35235	518	32676	33723	35179	529	32722	34769	35906	534	33643	36475	37487	550	48
288_2	39217	41028	43326	592	40412	42363	44162	604	41179	42142	44696	616	41283	43164	44325	622	42606	43930	45137	641	58
360_2	53748	56268	58544	847	54672	57402	60199	864	55251	57847	59619	881	56276	58653	61938	890	56965	59409	62929	916	72
48_3	10767	11035	11572	83	10221	10827	11115	84	10995	11492	11929	93	9178	9883	10489	81	9754	10529	11095	83	14
96_3	16638	17278	17654	174	17066	17491	18222	182	15957	16821	17349	180	14627	16582	17912	191	15236	17186	18141	184	29
144_3	23443	24211	25685	293	23659	24149	25893	290	24211	25426	26371	302	23194	24720	26101	296	23297	25592	26993	305	43
192_3	26199	27175	28225	392	27090	28066	28826	399	27376	28140	29529	403	27016	28405	29147	407	27153	29381	30145	420	58
240_3	29992	31130	32527	484	30956	32061	33545	494	31108	32105	33491	499	31151	33101	34184	504	32029	34726	35690	519	72
288_3	37337	39062	41251	564	38476	40333	42047	575	39205	41123	42555	581	39305	41096	42202	587	40565	41826	42975	604	86
360_3	51176	53576	55744	806	52056	54656	57320	823	52608	55080	56768	831	53584	55848	58976	839	54240	56568	59920	864	108
48_4	10239	10617	11313	82	9943	10418	10886	84	10807	11144	11671	86	8939	9615	9883	78	9545	10051	10539	80	19
96_4	16062	16943	17257	166	16810	17146	17481	175	15871	16425	17434	160	16012	16907	17336	179	16391	16991	17496	177	38
144_4	22700	23723	25488	279	23650	24242	25187	291	23765	24990	25945	296	22747	24273	25602	288	22841	25114	26506	294	58
192_4	25690	26612	27737	373	26528	27235	27938	383	26846	27620	28988	391	26443	27896	28628	403	26623	28850	29614	411	77
240_4	29375	30447	31877	461	30317	31401	32874	473	30447	31411	32863	483	30545	32484	33502	511	31411	34401	34997	521	96
288_4	36519	38266	40433	537	37624	39493	41240	549	38365	39338	41737	560	38442	40278	41339	583	39692	40974	42135	595	115
360_4	50480	52904	55056	768	51344	53960	56608	786	51904	54392	56088	802	52088	55184	58312	833	53584	55904	59216	850	144
平均值	28183	29377	30724	400	28831	29968	31284	409	29100	30250	31510	416	28634	30289	31553	421	29278	31156	32422	431	—

RDA 为国际期刊中的有效算法, 该算法在禁忌搜索算法中加入 RDA. 测试结果见表 8.

最后, 为验证 EACO_CD 的有效性, 在本文问题 LMHFVPR_TW 上, 将 EACO_CD 和国际期刊中的有效算法 IACO_CD^[6] 和 IHGA^[3] 进行对比. IACO_CD 为两层分解算法, 该算法第 2 层将问题分解为单车辆 LVRP (即 TSPs), 而 IHGA 通过加入车型编码后对问题进行整体求解. 测试结果见表 9.

3.2.2 测试问题选取

本文的测试数据来源于网站 (<http://neo.lcc.uma.es/vrp/>) 中的 MDVRP 数据集 (数据中不存在明显聚类特征的客户位置分布). 其中包括问题 pr01 (48 个客户), pr02 (96 个客户), pr03 (144 个客户), pr04 (192 个客户), pr05 (240 个客户), pr06 (288 个客户), p22 (360 个客户). 由于该数据中未含有车型数据, 故在 HKMA 和 EACO_CD 的有效性验证时加入车型数据 (见表 6), 各比较算法在每个问题上独立运行 20 次, 每个算法设置相同运行时间 T (s). 性能指标为各算法 20 次运行中的最优

值 (Best)、平均值 (Average)、最差值 (Worst) 和标准差 (SD). 各种带分解策略的比较算法每次运行时间为问题分解算法与问题求解算法运行时间的总和, 譬如, EACO_CD 每次运行的时间为 EACO、IBKA 和 HKMA 的运行时间总和, 表 7~9 中粗体为算法运行结果的较小值.

3.3 实验结果比较和分析

3.3.1 验证 EACO 有效性

由表 7 可知, EACO1 (即不含问题分解策略的 EACO_CD) 的最优值和平均值均优于 DHACO. 在算法全局搜索部分, DHACO 和 EACO1 的复杂度无明显差异, 但 EACO1 在全局阶段引入信息素挥发系数控制因子, 进一步动态调节信息素挥发系数, 可有效控制信息素的挥发, 从而提高了算法的全局搜索能力. 在算法局部搜索部分, 相比 DHACO 的 VNS 策略, EACO1 的 TVNS 能有效搜索更多

¹ 表 7~表 9 的完整测试结果可在: <https://pan.baidu.com/s/19sqBboZHLcGfQtZS7Id-Q> 提取码 3ev6 下载.

表 8 HKMA 与其他划分算法的对比结果
Table 8 Comparison results of HKMA and the other dividing algorithms

$N_{-}M_t$	EACO_HKMA			EACO_RDA			EACO_RAA			EACO_KEW			EACO2			TSA_RDA			$T(s)$
	最优	平均	最差	最优	平均	最差	最优	平均	最差	最优	平均	最差	最优	平均	最差	最优	平均	最差	
48_2	12801	13612	13901	13220	14646	15128	12467	13293	13998	12607	13317	13922	12218	13798	14202	13352	14792	15279	15
96_2	16299	16759	17342	17570	19461	20106	16238	17661	18591	16754	17704	18101	16543	17838	18777	17746	19656	20307	29
144_2	22061	23265	24089	23361	25860	26717	22390	23665	24721	22266	23533	24590	22838	23934	25215	23595	26119	26984	44
192_2	24847	25998	26933	25983	26905	28228	26000	27201	28372	26039	26918	28238	26520	27745	28939	26243	27174	28510	57
240_2	25958	27124	28797	26951	28002	29726	29253	30130	30753	29409	30076	30878	30131	31034	31976	27221	28282	30023	72
288_2	32225	33356	33838	33783	34713	36290	32740	35242	37189	32540	35156	37175	34050	36652	38677	34121	35060	36653	87
360_2	48344	50050	51763	50695	52088	54440	49129	52874	55793	48828	52743	53780	51094	54989	58025	51202	52609	54984	108
48_3	11896	12400	13111	12574	13919	14386	11755	11999	13312	11995	12658	13236	11520	12240	13999	12700	14058	14530	21
96_3	15777	15929	16997	16704	18494	19122	16011	16792	17680	15932	16839	17594	16171	16960	17857	16871	18679	19313	44
144_3	20965	21171	22661	22207	24587	25394	21291	22312	23490	21170	22366	23380	21717	22758	23960	22429	24833	25648	65
192_3	23624	24718	25597	24704	25570	26823	24714	25850	26974	24748	25588	26836	25455	26626	27783	24951	25826	27091	87
240_3	24680	25776	27367	25620	26616	28248	27804	28634	29235	27948	28582	29351	28916	29779	30404	25876	26882	28530	108
288_3	30621	31700	32158	32106	32984	34483	31111	33493	35335	30929	33414	35335	32667	35168	37102	32427	33314	34828	129
360_3	45933	47554	48245	48172	49502	51726	46691	50237	53019	47401	50118	51004	49026	52749	55670	48654	49997	52243	162
48_4	10755	10998	12257	11330	12533	12956	10700	11393	11985	10800	11402	11927	10999	11766	11999	11443	12658	13086	29
96_4	14214	14349	14672	15047	16655	17221	14422	15118	15927	14346	15171	15843	14566	15269	16086	15197	16822	17393	57
144_4	18878	19057	20506	19998	22039	22865	19165	20085	21148	19057	20140	21048	19548	20487	21571	20198	22360	23094	87
192_4	21273	22251	23043	22237	23018	24145	22257	23268	24281	22282	23034	24158	22925	23966	25009	22459	23248	24386	116
240_4	22215	23209	24642	23073	23966	25438	25035	25778	26316	25161	25735	26431	26036	26809	27369	23304	24206	25692	144
288_4	27563	28542	29950	28902	29689	31040	28012	30149	31812	27842	30088	31815	29413	31656	33403	29191	29986	31350	173
360_4	41350	42809	44034	43360	44558	46561	42028	45218	47722	41776	43111	45711	44129	47479	50108	43794	45004	47027	216
平均值	24407	25197	25970	25600	26948	28145	25201	26659	27983	25230	26652	27636	26023	27605	28944	25856	27217	28426	—

不同区域, 具有更强的局部搜索能力.

3.3.2 验证 IBKA 有效性

由表 7 可知, EACO_IBKA 总体优于 EACO_KM 和 EACO_NNA. 这表明 IBKA 通过聚类和平滑客户群, 可以更合理地分配车场并较好地实现子问题解耦. 同时, EACO_IBKA 也总体优于不含问题分解策略的整体求解算法 EACO1 和 DHACO. 这说明采用有效的问题分解策略为算法提前确定需搜索的较优区域, 可在一定程度上避免过多的低效搜索, 有利于增强算法性能. 此外, 测试结果显示, 带分解策略的算法更适用于较大规模问题, 原因在于这些问题的解空间更加庞大, 仅依靠智能算法自身进化机制难以快速引导算法发现优质解区域.

3.3.3 验证 HKMA 有效性

由表 8 可知, EACO_HKMA 总体优于其他比较算法. 这表明 HKMA 综合考虑客户的 3 类属性并为各属性设定合适的权重, 可以更合理地划分子问题. 具体而言, EACO_HKMA 明显优于 EACO_RDA 和 TSA_RDA, 这说明 RDA 将问题分解为

更小规模的单车辆 LVRPs (即 TSPs) 会将算法实际搜索空间限定得过小, 从而遗漏较多存在优质解的区域; EACO_HKMA 明显优于 EACO_RAA (除 $N = 96$ 的 1 个较小规模问题) 和 EACO_KEW, 这说明 RAA 随机划分子问题的方式和 KEW 给各属性分配相同权重后再划分子问题的方式, 均无法获取真正优质的搜索区域; EACO_HKMA 也明显优于整体求解算法 EACO2, 这再次验证了将问题先合理分解可把算法搜索直接限定在较优搜索区域进行, 有利于提高算法效率. 此外, 与第 3.3.2 节类似, 本节的分解策略同样对较大规模问题更有效.

3.3.4 验证 EACO_CD 有效性

由表 9 可知, EACO_CD 总体优于 IACO_CD 和 IHGA. 具体而言, EACO_CD 的最优值、平均值、最差值均明显优于 IACO_CD, 且标准差整体较小, 这说明 IACO_CD 将问题分解为更小规模的单车辆 LVRPs (即 TSPs) 会遗漏较多存在优质解的区域, 同时 IACO_CD 自身算法的搜索能力有限, 从而导致该算法的性能和鲁棒性较差; EACO_

表 9 EACO_CD 性能验证
Table 9 Performance verification of EACO_CD

$N_P_t_M_t$	EACO_CD (EACO_IBKA_HKMA)				IACO_CD (IACO_NNA_SWA)				IHGA				$T(s)$
	最优值	平均值	最差值	标准差	最优值	平均值	最差值	标准差	最优值	平均值	最差值	标准差	
48_2_2	12145	12478	12859	232	12880	13212	13509	241	11 632	11 967	12 227	220	15
96_2_2	16370	16526	16904	295	17800	18071	18381	272	15 852	16 011	16 376	285	31
144_2_2	21 010	21 305	21 704	324	24589	24825	25406	336	21049	21640	22154	348	51
192_2_2	23 664	24 760	25 650	556	26040	27250	28218	576	24863	26010	26948	590	71
240_2_2	24 722	25 832	27 426	588	26707	27903	29640	641	27444	28720	29947	692	96
288_2_2	30 690	31 768	32 227	724	33772	34962	35457	803	35311	36550	37076	824	123
360_2_2	39 903	41 303	41 906	951	43907	45448	46111	1059	45507	46098	47781	1197	162
48_3_2	12999	14064	14425	293	13656	14787	15372	336	12 742	13 802	14 337	304	21
96_3_2	16883	17 869	18 993	358	18568	20115	20999	343	16 800	18283	19011	363	48
144_3_2	21 916	23 728	24 668	456	25647	27774	28874	539	23029	24921	25906	491	76
192_3_2	24 257	25 492	26 408	668	26692	28059	29065	671	26688	28052	29053	746	109
240_3_2	24 672	26 558	28 026	792	27884	30021	31675	740	28382	30554	32236	787	144
288_3_2	30 258	31 572	32 860	907	34814	36319	37797	1017	35414	36956	38450	1092	183
360_3_2	39 347	41 061	42 734	1141	43291	45187	47020	1263	47228	49291	51297	1389	243
48_3_3	12284	13 225	13 755	280	12830	13906	14446	322	12 123	13253	13797	287	29
96_3_3	15 867	17 082	17 860	388	17470	18911	19664	368	16514	17874	18584	478	62
144_3_3	20 620	22 321	23 193	438	22488	24336	25294	490	24128	26136	27154	522	102
192_3_3	22 816	23 977	24 832	631	25108	26383	27320	708	26242	27588	28567	743	145
240_3_3	24 209	25 982	27 348	655	26693	28731	30318	770	27859	29995	31632	803	192
288_3_3	28 458	29 693	30 892	828	32742	34157	35530	957	35582	37120	38622	1048	245
360_3_3	37 006	38 610	40 166	1 079	40722	42474	44186	1198	47081	49427	51523	1399	288
平均值	23 814	25 010	25 945	599	26395	27754	28775	650	26737	28107	29175	696	—

CD 的最优值、平均值、最差值、标准差明显优于 IHGA (除 $N = 96$ 的 2 个较小规模问题), 这表明随着车场、车型和客户的增多, 本文问题 LMHFV-PR_TW 的可行解空间呈指数倍增加, 仅采用智能算法 (如 IHGA) 难以在较短时间内完成大范围搜索以获得原问题的优质解, 而 EACO_CD 依靠合理的问题分解能将 EACO 的搜索限定在较优区域之内, 进而利用 EACO 较强的搜索能力可短时间内从较优区域中获得优质解, 从而使其性能和鲁棒性均较强。

综上所述, 结合 IBKA 和 HKMA 的 EACO_CD 是求解本文问题 LMHFVPR_TW 的有效鲁棒算法。

4 结束语

针对多车场多车型绿色车辆路径问题 LMHFVPR_TW, 本文提出一种结合聚类分解策略的增强蚁群算法 (EACO_CD) 进行求解。这是首次采用智能优化算法求解该问题。EACO_CD 包括两个阶段。第 1 阶段为问题分解, 该阶段设计两种聚类分解策略 IBKA 和 HKMA, 用于对 LMHFVPR_TW

TW 进行分解, 从而有效控制问题规模并快速引导算法在较优区域内搜索; 第 2 阶段为子问题求解, 该阶段采用 EACO 对每个分解后的子问题 LVRP_TW 进行求解。在 EACO 的全局搜索中, 通过加入信息素挥发系数控制因子动态调节信息素挥发系数的取值, 可较好地避免算法出现过早收敛, 进而引导算法搜索达到解空间中更多的不同区域。同时, 为增强 EACO 的局部搜索能力, 设计一种两阶段变邻域搜索对全局搜索获得的优质解区域进行细致且高效的搜索。仿真实验和算法比较验证了所提 EACO_CD 是求解 LMHFVPR_TW 的有效算法。对于 HFVPR 和 MHFVPR, 已有的问题分解方法大都将其最终分解为一系列更为简单的单车辆 VRP (即标准 TSP), 这一分解方式是否合理且唯一, 目前尚无人探讨。本文所提的 IBKA 和 HKMA 把 LMHFVPR_TW 有效分解为一系列单车型 VRP (即多车辆 TSP), 使其子问题具有较大的可行解区域, 进而适当扩大了算法的实际搜索空间, 取得了更好的效果。这表明合理选取子问题可在一定程度上提高算法求解性能, 也对复杂车辆路径问题如何

进行分解优化设计具有借鉴意义. 后续研究将把 EACO_CD 和随机处理技术结合, 用于求解动态复杂车辆路径问题.

References

- Dantzig G B, Ramser J H. The truck dispatching problem. *Management Science*, 1959, **6**(1): 80–91
- Anbuudayasankar S P, Ganesh K, Mohapatra S. *Models for Practical Routing Problems in Logistics*. Cham: Springer, 2014.
- Li H Q, Yuan J L, Lv T, Chang X Y. The two-echelon time-constrained vehicle routing problem in linehaul-delivery systems considering carbon dioxide emissions. *Transportation Research Part D: Transport and Environment*, 2016, **49**: 231–245
- Zhao Yan-Wei, Zhang Jing-Ling, Wang Wan-Liang. *Vehicle Routing Optimization Methods for Logistics Distribution*. Beijing: Science Press, 2014.
(赵燕伟, 张景玲, 王万良. 物流配送的车辆路径优化方法. 北京: 科学出版社, 2014.)
- Sbihi A, Eglese R W. Combinatorial optimization and green logistics. *4OR — A Quarterly Journal of Operations Research*, 2007, **5**(2): 99–116
- Chen D S, Batson R G, Dang Y. *Applied Integer Programming*. Hoboken: John Wiley & Sons, 2010.
- Jabir E, Panicker V V, Sridharan R. Design and development of a hybrid ant colony-variable neighbourhood search algorithm for a multi-depot green vehicle routing problem. *Transportation Research Part D: Transport and Environment*, 2017, **57**: 422–457
- Kaabachi I, Jriji D, Krichen S. An improved ant colony optimization for green multi-depot vehicle routing problem with time windows. In: Proceedings of the 18th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD). Kanazawa, Japan: IEEE, 2017.
- Xiao Y Y, Konak A. The heterogeneous green vehicle routing and scheduling problem with time-varying traffic congestion. *Transportation Research Part E: Logistics and Transportation Review*, 2016, **88**: 146–166
- Kwon Y J, Choi Y J, Lee D H. Heterogeneous fixed fleet vehicle routing considering carbon emission. *Transportation Research Part D: Transport and Environment*, 2013, **23**: 81–89
- Geetha S, Vanathi P T, Poonthaler G. Metaheuristic approach for the multi-depot vehicle routing problem. *Applied Artificial Intelligence*, 2012, **26**(9): 878–901
- Geetha S, Poonthaler G, Vanathi P T. Nested particle swarm optimisation for multi-depot vehicle routing problem. *International Journal of Operational Research*, 2013, **16**(3): 329–348
- Ho W, Ho G T S, Ji P, Lau H C W. A hybrid genetic algorithm for the multi-depot vehicle routing problem. *Engineering Applications of Artificial Intelligence*, 2008, **21**(4): 548–557
- Wang Y, Assogba K, Liu Y, Ma X L, Xu M Z, Wang Y H. Two-echelon location-routing optimization with time windows based on customer clustering. *Expert Systems with Applications*, 2018, **104**: 244–260
- Dondo R, Cerdá J. A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows. *European Journal of Operational Research*, 2007, **176**(3): 1478–1507
- Tang Y L, Cai Y G, Yang Q J. Improved ant colony optimization for multi-depot heterogeneous vehicle routing problem with soft time windows. *Journal of Southeast University (English Edition)*, 2015, **31**(1): 94–99
- Dorigo M, Maniezzo V, Colnani A. Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 1996, **26**(1): 29–41
- Wang Su-Xin, Gao Li, Cui Xiao-Guang, Cao Hong-Mei. Study on multi-requirement points vehicle scheduling model and its swarm mix algorithm. *Acta Automatica Sinica*, 2008, **34**(1): 102–104
(王素欣, 高利, 崔小光, 曹宏美. 多需求点车辆调度模型及其群体智能混合求解. 自动化学报, 2008, **34**(1): 102–104)
- Lee C Y, Lee Z J, Lin S W, Ying K C. An enhanced ant colony optimization (EACO) applied to capacitated vehicle routing problem. *Applied Intelligence*, 2010, **32**(1): 88–95
- Yu B, Yang Z Z. An ant colony optimization model: The period vehicle routing problem with time windows. *Transportation Research Part E: Logistics and Transportation Review*, 2011, **47**(2): 166–181
- Ding Q L, Hu X P, Sun L J, Wang Y Z. An improved ant colony optimization and its application to vehicle routing problem with time windows. *Neurocomputing*, 2012, **98**: 101–107
- Yan F L. Autonomous vehicle routing problem solution based on artificial potential field with parallel ant colony optimization (ACO) algorithm. *Pattern Recognition Letters*, 2018, **116**: 195–199
- Chen Xi-Qiong, Hu Da-Wei, Yang Qian-Qian, Hu Hui, Gao Yang. An improved ant colony algorithm for multi-objective vehicle routing problem with simultaneous pickup and delivery. *Control Theory and Applications*, 2018, **35**(9): 1347–1356
(陈希琼, 胡大伟, 杨倩倩, 胡卉, 高扬. 多目标同时取送货车辆路径问题的改进蚁群算法. 控制理论与应用, 2018, **35**(9): 1347–1356)
- Xu H T, Pu P, Duan F. Dynamic vehicle routing problems with enhanced ant colony optimization. *Discrete Dynamics in Nature and Society*, 2018, 2018: Article No. 1295485
- Demir E, Bektaş T, Laporte G. An adaptive large neighborhood search heuristic for the pollution-routing problem. *European Journal of Operational Research*, 2012, **223**(2): 346–359
- Bektaş T, Gouveia L. *Requiem* for the Miller-Tucker-Zemlin subtour elimination constraints. *European Journal of Operational Research*, 2014, **236**(3): 820–832
- Toth P, Vigo D. *The Vehicle Routing Problem*. Beijing: Tsinghua University Press, 2011.
(Toth P, Vigo D. 车辆路径问题. 北京: 清华大学出版社, 2011.)
- Wolsey L A. *Integer Programming*. New York: Wiley, 1998.
- Schneider M, Stenger A, Goeke D. The electric vehicle-routing problem with time windows and recharging stations. *Transportation Science*, 2014, **48**(4): 500–520
- Gu S S. A polynomial time solvable algorithm to linearly constrained binary quadratic programming problems with Q being a tri-diagonal matrix. In: Proceedings of the 5th International Conference on Intelligent Control and Information Processing (ICICIP). Dalian, China: IEEE, 2011.
- Meng X H, Li J, Zhou M C, Dai X Z, Dou J P. Population-based incremental learning algorithm for a serial colored traveling salesman problem. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018, **48**(2): 277–288
- Wang Y, Ma X L, Lao Y T, Wang Y H. A fuzzy-based customer clustering approach with hierarchical structure for logistics network optimization. *Expert Systems with Applications*, 2014, **41**(2): 521–534
- Wang Y, Zhang J, Assogba K, Liu Y, Xu M Z, Wang Y H. Collaboration and transportation resource sharing in multiple centers vehicle routing optimization with delivery and pickup. *Knowledge-Based Systems*, 2018, **160**: 296–310
- Wang B D, Miao Y W, Zhao H Y, Jin J, Chen Y Z. A biclustering-based method for market segmentation using customer pain points. *Engineering Applications of Artificial Intelligence*, 2016, **47**: 101–109
- Ji J C, Pang W, Zhou C G, Han X, Wang Z. A fuzzy k-proto-type clustering algorithm for mixed numeric and categorical data. *Knowledge-Based Systems*, 2012, **30**: 129–135

- 36 Wang Y, Ma X L, Liu M W, Gong K, Liu Y, Xu M Z, et al. Cooperation and profit allocation in two-echelon logistics joint distribution network optimization. *Applied Soft Computing*, 2017, **56**: 143–157
- 37 He R H, Xu W B, Sun J X, Zu B Q. Balanced k-means algorithm for partitioning areas in large-scale vehicle routing problem. In: Proceedings of the 3rd International Symposium on Intelligent Information Technology Application. Nanchang, China: IEEE, 2009.
- 38 Wang Dong-Feng, Meng Li. Performance analysis and parameter selection of PSO algorithms. *Acta Automatica Sinica*, 2016, **42**(10): 1552–1561
(王东风, 孟丽. 粒子群优化算法的性能分析和参数选择. 自动化学报, 2016, **42**(10): 1552–1561)
- 39 Beasley J E. Route first–cluster second methods for vehicle routing. *Omega*, 1983, **11**(4): 403–408
- 40 Gillett B E, Miller L R. A heuristic algorithm for the vehicle-dispatch problem. *Operations Research*, 1974, **22**(2): 340–349
- 41 Yu B, Yang Z Z, Yao B Z. An improved ant colony optimization for vehicle routing problem. *European Journal of Operational Research*, 2009, **196**(1): 171–176
- 42 Mladenović N, Hansen P. Variable neighborhood search. *Computers and Operations Research*, 1997, **24**(11): 1097–1100
- 43 Li Jin, Fu Pei-Hua. Heterogeneous fixed fleet low-carbon routing problem and algorithm. *Computer Integrated Manufacturing Systems*, 2013, **19**(6): 1351–1362
(李进, 傅培华. 具有固定车辆数的多车型低碳路径问题及算法. 计算机集成制造系统, 2013, **19**(6): 1351–1362)



胡蓉 昆明理工大学信息工程与自动化学院副教授. 2004 年获得清华大学自动化系硕士学位. 主要研究方向为调度理论与方法, 智能计算, 决策支持系统.

E-mail: ronghu@vip.163.com

(HU Rong Associate professor at the School of Information Engineering and Automation, Kunming University of Science and Technology. She received her master degree from Tsinghua University in 2004. Her research interest covers scheduling theory and method, intelligent computation, and decision support system.)



李洋 昆明理工大学信息工程与自动化学院硕士研究生. 2009 年获得昆明理工大学电力工程学院学士学位. 主要研究方向为调度理论与智能优化算法.

E-mail: yang.liam@hotmail.com

(LI Yang Master student at the School of Information Engineering and Automation,

Kunming University of Science and Technology. He received his bachelor degree from Kunming University of Science and Technology in 2009. His research interest covers scheduling methods and intelligent optimization algorithms.)



钱斌 昆明理工大学信息工程与自动化学院教授. 2009 年获得清华大学自动化系博士学位. 主要研究方向为调度理论与方法, 智能优化. 本文通信作者.

E-mail: bin.qian@vip.163.com

(QIAN Bin Professor at the School of Information Engineering and Automation, Kunming University of Science and Technology. He received his Ph.D. degree from Tsinghua University in 2009. His research interest covers scheduling theory and method, and intelligent optimization. Corresponding author of this paper.)



金怀平 昆明理工大学信息工程与自动化学院副教授. 2016 年获得北京理工大学博士学位. 主要研究方向为智能计算和软测量方法.

E-mail: jinhuaping@gmail.com

(JIN Huai-Ping Associate professor at the School of Information Engineering and Automation, Kunming University of Science and Technology. He received his Ph.D. degree from Beijing Institute of Technology in 2016. His research interest covers intelligent computation and soft sensor methods.)



向凤红 昆明理工大学信息工程与自动化学院教授. 2002 年获得昆明理工大学博士学位. 主要研究方向为智能优化与控制.

E-mail: xiangfh5447@sina.com.cn

(XIANG Feng-Huang Professor at the School of Information Engineering and Automation, Kunming University of Science and Technology. He received his Ph.D. degree from Kunming University of Science and Technology in 2002. His research interest covers intelligent optimization and control.)