

# 灵活车场多类型叫车接送问题的改进模拟退火算法

陈可嘉, 司徒腾宽, 方云飞<sup>†</sup>, 陈荣晖

(福州大学 经济与管理学院, 福建 福州 350108)

**摘要:** 面对老龄化社会和个性化交通需求的挑战, 灵活车场多类型叫车接送在提供高效、便捷的交通解决方案方面显得尤为重要. 然而, 灵活车场环境下车辆的异构性和调度的复杂性, 使得服务优化面临巨大难题. 本文提出的改进模拟退火算法, 针对这一问题, 显著优化了车辆总行驶距离. 该算法设计了一种基于宽容度与回温机制的解接受策略, 以及3种新的搜索算子, 旨在提高算法稳定性和避免过早陷入局部最优. 与CPLEX求解器和确定性退火算法相比, 本算法在多个测试实例上展现出显著优势. 通过敏感性分析, 研究了乘客最大行驶时间和时间窗口对算法性能的影响, 为实际的叫车接送规划提供了决策支持.

**关键词:** 叫车接送问题; 灵活车场; 模拟退火; 解接受策略优化; 回温策略

**引用格式:** 陈可嘉, 司徒腾宽, 方云飞, 等. 灵活车场多类型叫车接送问题的改进模拟退火算法. 控制理论与应用, 2026, 43(3): 612 – 622

DOI: 10.7641/CTA.2024.40060

## Improved simulated annealing algorithm for multiple depots heterogeneous dial-a-ride problem with flexible depots

CHEN Ke-jia, SITU Teng-kuan, FANG Yun-fei<sup>†</sup>, CHEN Rong-hui

(School of Economics and Management, Fuzhou University, Fuzhou Fujian 350108, China)

**Abstract:** Facing the challenges of an aging society and individualized transportation demands, the multiple depots heterogeneous dial-a-ride problem with flexible depots (MDHDARP-FD) is crucial for providing efficient and convenient transportation solutions. However, the heterogeneity of vehicles and the complexity of scheduling in flexible depots environments pose significant challenges to service optimization. This paper proposes an improved simulated annealing algorithm that significantly optimizes the total vehicle travel distance for the problem. According to the characteristics of the problem, the algorithm designs a solution acceptance strategy based on tolerance and reheating mechanisms, as well as three new search operators to enhance the algorithm's stability and avoid premature convergence to local optimization. Compared with the CPLEX solver and the deterministic annealing algorithm, our algorithm shows significant advantages across multiple test instances. Through sensitivity analysis, the impacts of passengers' maximum ride time and time windows on the algorithm's performances are studied, which provides decision support for actual dial-a-ride planning.

**Key words:** dial-a-ride problem; flexible depots; simulated annealing; solution acceptance strategy optimization; reheating strategy

**Citation:** CHEN Ke-jia, SITU Tengkuan, FANG Yunfei, et al. Improved simulated annealing algorithm for multiple depots heterogeneous dial-a-ride problem with flexible depots. *Control Theory & Applications*, 2026, 43(3): 612 – 622

## 1 引言

在人口日益老龄化的今天, 为老人和病人提供具有接送功能的车辆运输服务显得尤为重要. 叫车接送问题 (dial-a-ride problem, DARP)<sup>[1]</sup>是车辆路径问题 (vehicle routing problem, VRP)的延伸, 需要为乘客提供点到点的服务. 在路径规划时, 不仅需要考虑路径行驶成本, 还要考虑乘客的最大可承受乘车时间、服

务时间窗等.

DARP已证明是NP-hard问题, 多种启发式算法已在此类问题成功应用. 对于标准静态DARP问题, Pfeiffer等<sup>[2]</sup>提出了一种自适应大领域搜索和动态规划算法, 并在德国汉堡市的实例中得到了良好结果; Gschwind等<sup>[3]</sup>给出了一种自适应大领域搜索算法, 通过在修复步骤中评估请求插入的可行性, 该算法在几组算

收稿日期: 2024-01-21; 录用日期: 2024-09-18.

<sup>†</sup>通信作者. E-mail: yf.fang@fzu.edu.cn.

本文责任编辑: 丛爽.

国家社会科学基金项目(23BGL296)资助.

Supported by the National Social Science Foundation of China (23BGL296).

例中找到了新的最优解。

为了契合实际情况, 研究者们引入越来越多的现实因素<sup>[4]</sup>, 例如允许车辆选择乘客需求<sup>[5]</sup>、自适应乘客乘车时间<sup>[6]</sup>、固定线路<sup>[7-8]</sup>等。其中, 多类型叫车接送问题 (heterogeneous DARP, HDARP) 是DARP的一类分支, 该问题考虑了乘客与车辆类型的多样性, 不同类型的乘客可能会对车辆与接送条件有不同的要求, 例如Tóth等<sup>[9]</sup>研究的非紧急患者叫车接送问题, 该问题中普通患者与残疾患者占用车辆位置资源不同, 并且两类患者具有不同的上下车服务时间。

相比于传统的DARP, HDARP需要额外考虑多类型资源的分配问题, 增加了问题的复杂度<sup>[10]</sup>, 精确求解算法难以在短时间内得到高质量的方案, 一些研究针对HDARP设计了启发式算法进行求解: Masmoudi等<sup>[11]</sup>除了考虑特殊乘客与普通乘客对车辆位置需求类型的不同, 还考虑了车辆类型的多样性, 设计了一种混合自适应大领域搜索算法, 通过自适应权重调整局部搜索操作; Chu等<sup>[12]</sup>同样研究了车辆类型的多样性, 将叫车接送与固定线路公交整合, 并在大领域搜索算法中加入了周期性的破坏重构算子。尽管上述改进方法在一定程度上提高了HDARP的求解效率, 但过早收敛, 容易陷入局部最优等缺陷仍需要深入研究。

车场是VRP的重要概念, 是车辆分配路线的起点和终点<sup>[13]</sup>。由于现实问题场景的复杂性, 越来越多的研究开始考虑多车场的场景<sup>[14]</sup>。其中, 多数的研究中车辆的起始与终点车场需要相同, 现实中却不一定都是这样, 因此, Kek等<sup>[15]</sup>提出了灵活车场的概念, 在灵活车场的场景下, 所有的车场均可作为车辆的起始、终点, 即车辆在服务结束后, 可以选择返回任意车场, 而不一定需要返回出发车场。

灵活车场概念的引入在一定程度上为车辆提供更大的选择范围, 优化路径, 降低总行驶距离。Markov等<sup>[16]</sup>考虑了灵活车场下的垃圾回收车辆路径问题, 提出了一种多重邻域搜索的启发式方法; Li等<sup>[17]</sup>考虑了带时间窗的灵活多车场车辆路径问题, 提出了一种混合遗传算法和自适应局部搜索。然而, 以上两项研究仅在可行性规则上允许了灵活车场的情况。随着研究的推进, Kantasa等<sup>[18]</sup>在灵活车场下动态多仓库车辆路径问题中提出了一种迭代随机启发式方法, 该方法为方案中的最后一个节点优先选择就近的车场; Che等<sup>[19]</sup>针对灵活车场下的多车场加油站补给问题提出了一种基于禁忌搜索的自适应大邻域搜索算法, 该算法同样考虑了迭代中的车场调整策略, 通过将方案中的节点与就近的车场匹配, 来获得更优方案。可以发现, 在算法中加入符合灵活车场场景的搜索策略, 能有效地提高该场景下的算法效率。

尽管如此, 对于灵活车场下叫车接送问题的研究

仍然较少。Fang等<sup>[20]</sup>研究了灵活车场下的灵活多车场多类型的叫车接送问题 (multiple depots HDARP-F-D, MDHDARP-F-D) 建立了混合整数模型, 利用CPL-EX求解器验证了该模型的有效性。然而, 目前并没有针对灵活车场下DARP的算法设计。相比于非灵活车场下的DARP, 在灵活车场场景中算法需要考虑灵活车场对于改进解的可能性, 因此, 已有的启发式算法并不能很好地匹配灵活车场下DARP的问题特征。

综上所述, 现有关于DARP的研究主要存在以下问题: 1) 关于DARP研究主要是针对非灵活车场的场景, 对于灵活车场下的DARP研究不够充分, 缺少基于问题特征的算法设计; 2) 针对HDARP的算法设计需要完善, 仍存在过早收敛等问题。

针对这些问题, 本文考虑了叫车接送问题中的灵活车场, 乘客多类型资源需求等情况, 以总行驶距离最小为目标, 提出了一种改进模拟退火算法 (improved simulated annealing algorithm, ISA)。针对多类型多车场等约束带来的问题复杂度增加, ISA设计了一种基于宽容度 (tolerance level, TL) 参数与回温机制的解接受策略, 采用TL参数代替概率接受模式, 增加了算法的稳定性, 并采用回温策略, 避免了算法在近似温度的多次计算, 或是因温度下降过快导致的陷入局部最优。此外, 设计了insert-FD算子、switch-FD算子、D&R-FD算子, 这些算子尤其考虑了灵活车场情景对于解的改进可能, 增加了MDHDARP-FD场景下的搜索效率。通过算例分析, 将本文算法与CPLEX求解器、确定性退火算法 (deterministic annealing, D-A)<sup>[21]</sup>进行比较, 验证算法在运行时间和求解结果的有效性, 并进行敏感性分析。

## 2 MDHDARP-FD问题描述与模型建立

### 2.1 问题描述与假设

MDHDARP-FD考虑一组车辆, 通过一定的路径满足多位乘客的叫车接送需求。每位乘客的需求包括出发地、目的地时间窗, 即车辆需要在一定时间内到达出发、目的地。若过早到达, 则需要等待, 直到时间窗的开始时间才能开始服务。此外, 还需要满足一系列约束, 例如车辆最长行驶时间、乘客最长乘车时间、车辆位置资源等。在此基础上, 需要找到一组车辆路径方案, 在所有乘客都被服务的情况下, 使所有车辆的总行驶距离最小化。

MDHDARP-FD相比传统DARP问题, 额外考虑了多类型以及灵活多车场。其中, 多类型指接送的乘客存在多种资源需求 (例如常规座位、轮椅位、担架位), 对应的, 车辆上配备有多种类型的资源; 灵活多车场指存在多个车场, 并且所有车场均可作为任意车辆的始发、终点。

## 2.2 符号定义

给出MDHDARP-FD模型的符号定义, 见表1.

表1 符号定义  
Table 1 Symbol definition

符号名称	含义
辅助参数	
$r$	车辆资源
$k$	车辆
$i, j$	节点
$M$	出发车场集合
$M'$	到达车场集合
$P$	乘客需求起始集合
$D$	乘客需求目的集合
集合	
$V$	点集合, $V = M \cup M' \cup P \cup D$
$A$	可行弧集
$R$	车辆资源集合
$K$	车辆集合
常规参数	
$m(k)$	车辆 $k$ 的出发车场
$G_k^r$	车辆 $k$ 上配备资源 $r$ 的数量
$q_j^r$	节点 $i$ 对资源 $r$ 的需求数量
$L$	乘客最大乘车时间
$H_k$	车辆 $k$ 的最大行驶时间
$S_i$	节点 $i$ 的服务时间
$e_i$	节点 $i$ 的服务时间窗下界
$l_i$	节点 $i$ 的服务时间窗上界
$c_{ij}$	从节点 $i$ 到节点 $j$ 的行驶距离
$t_{ij}$	从节点 $i$ 到节点 $j$ 的行驶时间
$n$	乘客需求数量
$N$	一个很大的正数
决策变量	
$x_{ij}^k$	车辆 $k$ 经过弧 $(i, j)$ 时为1, 否则为0
$B_i^k$	车辆 $k$ 在节点 $i$ 的开始服务时间
$Q_i^{kr}$	离开节点 $i$ 时车辆 $k$ 上位置资源 $r$ 的使用量

## 2.3 目标函数

由于直接关联到运输效率与成本, 最小化总行驶距离是DARP中最常见的目标函数之一<sup>[10]</sup>. 最小化的总行驶距离有助于降低燃料消耗, 减少运营成本, 并提高服务效率. 因此, 本文模型以最小化总行驶距离为目标, 即

$$\min \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x_{ij}^k. \quad (1)$$

## 2.4 约束条件

基于上述问题描述与符号, MDHDARP-FD的约束条件具体如下:

1) 乘客服务约束.

$$\sum_{k \in K} \sum_{j: (i,j) \in A} x_{ij}^k = 1, \quad i \in P, \quad (2)$$

$$\sum_{j: (i,j) \in A} x_{ij}^k - \sum_{j: (j,n+i) \in A} x_{j,n+i}^k = 0, \quad i \in P, \quad k \in K, \quad (3)$$

式(2)表示所有乘客被且仅被服务一次; 式(3)表示每

个乘客的起点和终点被同一辆车服务, 其中:  $i$ 为出发节点,  $n+i$ 表示从 $i$ 出发的到达节点. 以上约束从乘客角度, 要求每位乘客的服务需求被满足.

2) 路径与车辆分配约束.

$$\sum_{(m(k),j) \in A} x_{m(k),j}^k = 1, \quad k \in K, \quad (4)$$

$$\sum_{j \in M'} \sum_{i: (i,j) \in A} x_{ij}^k = 1, \quad k \in K, \quad (5)$$

$$\sum_{j: (j,i) \in A} x_{ji}^k - \sum_{j: (i,j) \in A} x_{ij}^k = 0, \quad i \in P \cup D, \quad k \in K, \quad (6)$$

式(4)表示每辆车从出发车场始发; 式(5)表示每辆车最终返回到达车场; 式(6)限制了每辆车在路径上的连贯性. 以上约束从车辆角度, 要求每个车辆的路径符合实际要求.

3) 时间约束.

$$e_i \leq B_i^k \leq l_i, \quad i \in V, \quad k \in K, \quad (7)$$

$$B_i^k - B_{m(k)}^k \leq H_k, \quad i \in M', \quad k \in K, \quad (8)$$

$$t_{i,n+i} \leq B_{n+i}^k - (B_i^k + s_i) \leq L, \quad i \in P, \quad k \in K, \quad (9)$$

式(7)为时间窗约束, 每个节点的实际服务时间需在对应时间窗内; 式(8)表示每辆车的最大行驶时间约束; 式(9)表示每个乘客的最大乘车时间约束. 以上约束从时间角度, 要求了包括乘客服务时间窗、车辆最大行驶时间、乘客最大乘车时间在内的时间要求.

4) 时间与资源动态更新约束.

$$B_j^k \geq (B_i^k + s_i + t_{ij}) + N(x_{ij}^k - 1), \quad (i, j) \in A, \quad k \in K, \quad (10)$$

$$Q_j^{kr} \geq (Q_i^{kr} + q_j^r) + N(x_{ij}^k - 1), \quad (i, j) \in A, \quad r \in R, \quad k \in K, \quad (11)$$

$$Q_i^{kr} \geq \max\{0, q_i^r\}, \quad i \in P \cup D, \quad r \in R, \quad k \in K, \quad (12)$$

$$Q_i^{kr} \leq \min\{G_k^r, G_k^r - q_i^r\}, \quad i \in P \cup D, \quad r \in R, \quad k \in K, \quad (13)$$

式(10)定义了节点 $j$ 开始服务的时间; 式(11)定义了离开节点 $j$ 时车辆的资源情况; 式(12)–(13)为车辆资源更新约束. 以上约束从乘客资源需求角度, 要求每个车辆始终满足其路线上的乘客资源需求.

5) 决策变量约束.

$$x_{ij}^k \in \{0, 1\}, \quad (i, j) \in A, \quad k \in K, \quad (14)$$

$$B_i^k \geq 0, \quad i \in V, \quad k \in K, \quad (15)$$

$$Q_i^{kr} \in \{0, 1, 2, \dots\}, \quad i \in V, \quad r \in R, \quad k \in K, \quad (16)$$

式(14)–(16)是一些决策变量约束, 定义了模型中决策变量的取值范围, 确保了解的可行性与合理性.

## 3 传统模拟退火算法

模拟退火算法 (simulated annealing, SA) 主要参考

物体退火降温过程, 使得算法在温度低值时趋向最优解. 模拟退火算法对于解决大规模组合优化问题具有较大的优势, 已在库存路径<sup>[22]</sup>、装配线平衡<sup>[23]</sup>等问题得到了良好的应用. 模拟退火算法一般有4个步骤.

**步骤 1** 设置最大迭代次数 $iterMax$ , 迭代次数 $iter = 0$ , 最低温度 $T_{min}$ , 起始温度 $T_{max}$ , 当前温度 $T = T_{max}$ . 找到问题的一个初始解 $\pi_0$ , 令当前解 $\pi = \pi_0$ .

**步骤 2** 从当前解 $\pi$ 的邻域中扰动产生新解 $\pi'$ , 计算目标函数 $f(\pi')$ , 令 $\Delta f = f(\pi') - \pi$ . 若 $\Delta f \leq 0$ , 则令 $\pi = \pi'$ ; 否则以概率 $\exp(-\Delta f/T)$ 接受解 $\pi'$ . 迭代次数 $iter = iter + 1$ .

**步骤 3** 判断迭代次数, 如果迭代次数 $iter =$

$iterMax$ 则继续执行, 否则转步骤2.

**步骤 4** 判断是否满足终止条件. 如果满足终止条件 $T < T_{min}$ , 那么返回 $\pi$ 为最优解, 算法结束. 如果不满足终止条件, 则令 $T = g(T)$ , 其中 $g$ 为温度变化函数, 转步骤2.

然而, 传统模拟退火算法概率接受解的模式具有较大的随机性, 可能会受到偶然性影响. 当温度较低时, 算法会在近似温度下反复进行, 此外, 若在退火过程中, 温度下降过快, 也可能导致算法陷入局部最优.

### 4 改进模拟退火算法

针对传统模拟退火算法中存在的不足, 本文提出一种改进模拟退火(improved SA, ISA)算法. 该算法框架见图1.

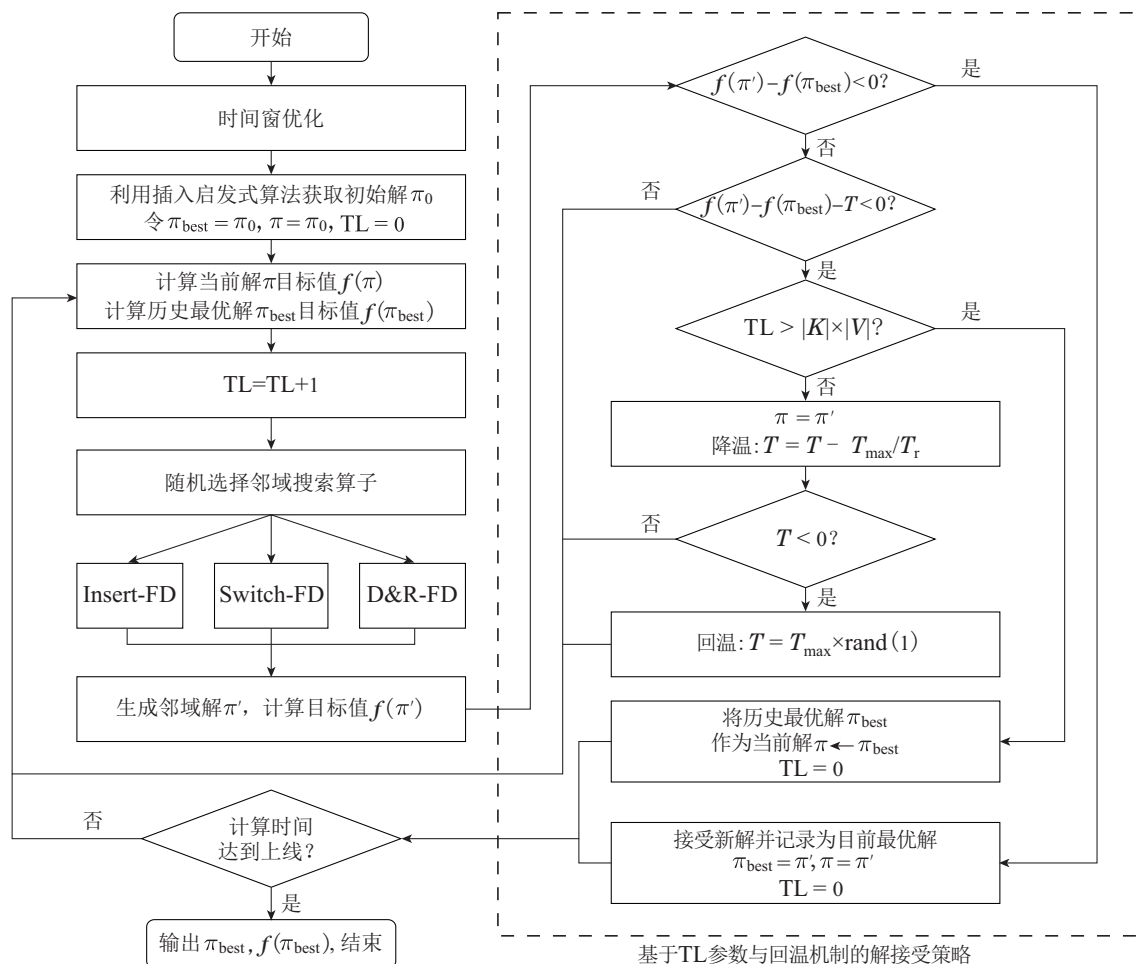


图 1 改进模拟退火算法流程图

Fig. 1 Flow chart of improved simulated annealing algorithm

ISA算法首先基于插入启发式算法得到一个高质量的初始解, 之后进行模拟退火操作. 与传统的模拟退火算法对比, ISA设计了一种基于宽容度 (tolerance level, TL) 参数与回温机制的解接受策略, 引入宽容度参数TL, 在当前邻域解的目标值 $f(\pi')$ 与TL满足一定条件时可接受当前未优化解, 相比概率接受频次将更

为确定, 这将增强算法的稳定性; 引入了回温机制, 即在温度降到0以下时, 将温度升高至一随机高温, 开始新一轮降温. 同时为了使算法可以终止, 将算法终止条件改为迭代次数. 这一改进可以提高算法效率, 避免在同一温度水平多次运行. 此外, ISA设计了3种邻域搜索算子, 针对MDHDARP-FD的问题特征, 强化了

算法的搜索效率.

#### 4.1 时间窗优化

为了提高运行效率, 可对各节点进行时间窗优化操作. 由式(9), 对于出发节点  $i \in P$  以时间窗下界  $e_i$  为例.  $e_i$  不会小于  $e_{n+i} - s_i - L$ , 即不会早于对应到达节点  $n+i$  的时间窗下界减去  $i$  点的服务时间与  $i$  到  $n+i$  点的最短行驶时间. 因此,  $e_i$  的范围可以进一步优化, 即

$$e_i = \max(e_{n+i} - s_i - L, e_i). \quad (17)$$

同理, 剩余时间窗参数的范围可进一步缩小, 见式(18)–(20):

$$l_i = \min(l_{n+i} - s_i - c_{i,n+i}, l_i), \quad (18)$$

$$e_{n+i} = \max(e_i + s_{n+i} + c_{i,n+i}, e_{n+i}), \quad (19)$$

$$l_{n+i} = \min(l_i + s_{n+i} + L, l_{n+i}). \quad (20)$$

经优化后, 所有节点的时间窗更紧凑, 可以减小算法运算难度, 提高效率.

#### 4.2 编码与解码

ISA采用十进制编码方式, 将每部车辆经过的路径按照“起始车场–乘客点–…–乘客点–到达车场”的方式编码. 例如编码“1-3-6-2, 2-4-5-7-8-1”, 表示一共有两辆车, 其中第1辆车从起始车场1出发, 经过乘客点3, 6, 最终到达车场2; 类似地, 第2辆车由起始车场2出发, 依次经过乘客点4, 5, 7, 8, 最终到达车场1. 解码方面, 按照编码计算车辆每次移动的距离, 求和即可得到目标值总行驶距离.

#### 4.3 初始解构造

对于ISA算法, 初始解采用插入启发式算法. 对第1辆车, 将用户依次安排在其路径; 若无法安排用户在该车辆路径, 则考虑将用户安排在第2辆车, 以此类推. 其中, 安排用户需满足结点时间窗、用户旅行时间、车辆资源容量、车辆工作时间等约束. 此外, 在每次插入新用户时, 一般存在多个可插入位置, 计算这些位置下的目标值, 选择满足约束且目标值最小的位置. 其算法框架见算法1(表2).

#### 4.4 邻域搜索

在MDHDARP-FD场景下, 可以通过灵活安排起始、到达车场, 来实现解的优化. 然而, 传统的DARP算子并未考虑这一场景, 为了解决这一问题, 本文设计了3种改进邻域搜索算子, 其主要思路如下:

##### 1) 算子1 Insert-FD.

随机选取当前解的一条路径, 对该路径上的任一乘客需求进行以下操作: 移除该路径上该乘客需求起点和终点, 将该对起点和终点遍历插入到其它路径, 选择除车场外其他路径之和最小的可行路径, 为该路径匹配总距离最小的起始、到达车场, 最终该方案为该算子运行结果.

##### 2) 算子2 Switch-FD.

随机选取当前解的一条路径A以及其上的任一乘客需求, 再随机选取另一条路径B以及其上的任一乘客需求. 将取自路径A的乘客需求起点和终点遍历插入路径B, 取自路径B的乘客需求起点和终点遍历插入路径A, 选择除车场外其他路径之和最小的可行方案, 为该方案匹配总距离最小的起始、到达车场, 最终该方案为该算子运行结果.

表2 算法1: 插入启发式算法

Table 2 Algorithm 1: Insertion heuristic algorithms

<b>输入:</b> 时间窗下界升序排列的节点集合sort, 车辆集合 $K = \{k_1, k_2, \dots, k_{ K }\}$ , 迭代数 $g = 1$ , 最大迭代次数 $g_{\max}$	
<b>输出:</b> 初始化方案 $\pi_0$	
1	初始化方案集 $\theta_1 = \emptyset$
2	<b>for</b> $g = 1$ <b>to</b> $g_{\max}$ <b>do</b>
3	初始化方案: $\pi = \emptyset$
4	<b>if</b> $g > 1$ <b>then</b>
5	对节点集合sort进行随机排序:
	sort = random(sort)
6	<b>for</b> $i = 1$ <b>to</b>  sort  <b>do</b>
7	初始化方案集 $\theta_2 = \emptyset$
8	<b>for</b> $z = 1$ <b>to</b>  K  <b>do</b>
9	<b>if</b> 节点 $i$ 可以插入到车辆 $k_z$ 的路径且满足条件 <b>then</b>
10	<b>for</b> 所有满足条件的插入位置 $p$ <b>do</b>
11	在方案 $\pi$ 中, 将节点 $i$ 插入车辆
12	$k_z$ 的位置 $p$ , 生成新方案 $\pi_{\text{new}}$
13	$\theta_2 = \theta_2 \cup \{\pi_{\text{new}}\}$
14	<b>break</b>
15	$\pi = \theta_2$ 中目标值最小的方案
16	$\theta_1 = \theta_1 \cup \{\pi\}$
17	$\pi_0 = \theta_1$ 中目标值最小的方案
18	<b>return</b> $\pi_0$ ;

##### 3) 算子3 D&R-FD.

随机选取当前解中至少2名, 至多  $N$  名乘客, 将其需求起点与终点从车辆方案中去除. 以随机顺序向解中插入这些乘客, 每次插入时遍历所有可行方案, 并选择其中除车场外其他路径之和最小的可行方案. 最终重新得到一个包含所有乘客的方案, 为该方案匹配总距离最小的起始、到达车场, 最终该方案为该算子运行结果.

以上算子在运算中考虑了灵活车场的特征. 由于所有点到车场的距离固定, 并不受方案中的排序影响, 因此, 在路径中添加新的乘客、寻找最优的方案时, 暂时不考虑车场位置对于方案的影响, 在确定某个方案

的插入最优后, 再为该路径匹配最优的车场安排, 可以在保证搜索质量的同时, 尽可能地减小算子计算量, 提升算法效率. 其中, 算子3是属于破坏与重组 (destroy & rebuild, D&R)算子,  $N$ 为可调整的参数, 通过控制 $N$ 的大小可以控制每次对于当前解的破坏程度.

#### 4.5 基于TL参数与回温机制的解接受策略

根据插入启发式初始化获得初始解后, 初始化各项参数, 算法开始迭代搜索邻域优化. 通过邻域搜索算子, 每次迭代获得邻域解 $f(\pi')$ , 在模拟退火算法的基础上, ISA通过引入TL参数, 回温机制, 改进解的接受策略. 具体步骤见算法2(见表3).

表3 算法2: 基于TL参数与回温机制的解接受策略  
Table 3 Algorithm 2: Solution acceptance strategy based on TL parameter and regeneration mechanism

<p><b>输入:</b> TL参数, 当前解<math>\pi</math>, 邻域解<math>\pi'</math>, 局部最优解<math>\pi_{\text{best}}</math>, 车辆数量<math> K </math>, 节点数量<math> V </math>, 当前温度<math>T</math>, 最高温度<math>T_{\text{max}}</math>, 初始温度<math>T_0</math>, 降温参数<math>T_r</math></p> <p><b>输出:</b> 本轮迭代解<math>\pi</math>, 局部最优解<math>\pi_{\text{best}}</math>, TL参数</p> <pre> 1 if <math>f(\pi') &lt; f(\pi_{\text{best}})</math> then 2   <math>\pi_{\text{best}} \leftarrow \pi'</math>; 3   <math>\pi \leftarrow \pi'</math>; 4   TL <math>\leftarrow 0</math>; 5 else 6   if <math>f(\pi') &lt; f(\pi_{\text{best}}) + T</math> then 7     if TL <math>&gt;  K  \times  V </math> then 8       <math>\pi \leftarrow \pi_{\text{best}}</math>; 9       TL <math>\leftarrow 0</math>; 10    else 11      <math>\pi \leftarrow \pi'</math>; 12      <math>T \leftarrow T - \frac{T_{\text{max}}}{T_r}</math>; 13      if <math>T &lt; 0</math> then 14        T <math>\leftarrow T_{\text{max}} \times \text{rand}(1)</math>; 15 return <math>\pi, \pi_{\text{best}}, \text{TL}</math>;</pre>
---

在该策略中, 温度代表着算法接受劣化解的概率. 高温意味着算法在开始时可以接受较差的解, 从而有机会探索到新的区域. 随着算法的进行, 温度逐渐降低, 算法逐渐减少对劣化解的接受概率, 从而使搜索过程更加聚焦于已发现的优化解附近. 当温度不断降低, 小于0时, 会采取升温至某一随机温度的“回温”策略. 回温是为了减小算法过早陷入局部最优解的概率, 令算法探索更广泛的解空间.

#### 4.6 时间复杂度分析

设算例中乘客规模为 $n$ , 车场总数为 $m$ , 则根据算法流程, 改进模拟退火算法的时间复杂度计算如下:

1) 插入启发式算法包含 $g_{\text{max}}$ 次插入构造初始解操作, 假设插入构造初始解时间复杂度为 $T_0(n)$ , 又因为 $g_{\text{max}}$ 与 $n$ 不相关, 为常数, 则插入启发式算法的时间复杂度 $T_1(n)$ 为

$$T_1(n) = O(g_{\text{max}} \cdot T_0(n)) = T_0(n).$$

对于 $T_0(n)$ , 每次向方案中插入两个节点, 由于乘客数量通常大于车辆数量, 在最坏情况下, 所有乘客均处在一辆车的路径上, 且每个乘客的出发点可插入路径中任意位置, 到达点可插入路径中对应出发点后任意位置. 在第 $i$ 次循环中, 方案数量为 $i \cdot (2i - 1)$ , 又由于需要循环 $n$ 次, 综上可得

$$T_1(n) = T_0(n) = O(n(1 + n(2n - 1))/2) = O(n^3 - \frac{1}{2}n^2 + \frac{1}{2}n) = O(n^3).$$

2) 邻域搜索中包含了3种算子, 每次迭代随机选择其中一个算子进行邻域搜索:

a) Insert-FD算子将一个乘客转移到另一车辆的路径中, 并选择除车场外总距离最小的点, 为其匹配最优车场. 其中, 随机选择乘客的操作与问题规模无关, 而移除乘客, 插入并选择总距离最小的操作可以看成是一次插入启发式算法中的第 $n$ 次循环插入操作, 再加上为该方案匹配最优车场, 因此, 时间复杂度为 $O(n(2n - 1) + m^2) = O(n^2 + m^2)$ ;

b) Switch-FD算子需要交换两个乘客所处车辆, 并通过遍历寻找乘客在另一路径上除车场外总距离最小的可行路径, 并为其匹配最优车场, 时间复杂度上与两次Insert-FD算子类似, 为 $O(2(n^2 + m^2)) = O(n^2 + m^2)$ ;

c) D&R-FD算子的复杂度则类似对当前解进行了至多 $N$ 次的插入操作, 由于 $N$ 为参数, 与问题无关, 则D&R-FD的时间复杂度同样为 $O(n)$ .

综上, 算法的总时间复杂度为

$$T(n) = O(n^3 + 3n^2 + 3m^2) = O(n^3 + m^2).$$

## 5 仿真求解及结果比较

为了验证本文设计的改进模拟退火算法的有效性, 以Braekers等<sup>[21]</sup>提供的datasetU和datasetE两组公开算例中共22个算例进行仿真求解. 其中: datasetU组算例考虑单一类型的乘客需求资源, datasetE组算例引入了4种乘客需求资源类型, 并且不同的乘客对这4种资源的需求不同; 同时, 算例名称代表算例类型、车辆数量和乘客需求数量, 比如E2-20, 表示datasetE组中, 共有2辆车, 20个乘客需求的算例.

使用MATLAB2022编程, 在Intel core i7 2.69 GHz处理器, 16GB内存, Windows11系统的计算机上运行. 参数方面, 采用田口正交试验法确定ISA的参数:

$T = 300, T_{\max} = 100, T_r = 10, g_{\max} = 100, N = 4.$

### 5.1 ISA算法有效性

为检验ISA算法的有效性,对算例Ea4-16进行求解.该算例共有4辆车,分别位于 $(-5, -5), (5, 5), (-5, 5), (5, -5)$ 4个车场,每辆车配备情况为:1个员工座位、2个病人座位、1个担架位和1个轮椅位,车辆最长行驶时间为240分钟.需要完成16个乘客接送需求,每

个乘客需求包含一个起点和一个终点,序号1-16为乘客需求起点的信息,序号17-32则为其需求终点的信息.所有的点在 $[-10, 10]^2$ 的欧式平面随机生成.乘客在起点和终点设置时间窗比例均为50%,时间窗长度为15分钟.乘客在起点和终点的服务时间为3分钟,乘客最长乘车时间为30分钟.不同的乘客对4种位置资源的需求亦有不同.具体信息见表4.

表4 算例Ea4-16的乘客需求信息  
Table 4 Passenger demand information of example Ea4-16

序号	横坐标	纵坐标	服务时间	位置需求				时间窗	时间窗	
				常规座位	病人座位	担架位	轮椅位	下界	上界	
1	6.267	0.981	3	30	0	0	0	1	0	1440
2	-4.718	6.925	3	30	0	1	0	0	0	1440
3	3.254	7.621	3	30	0	1	0	0	0	1440
4	9.654	2.799	3	30	0	0	1	0	0	1440
5	8.575	2.701	3	30	0	1	0	0	0	1440
6	-9.732	-8.314	3	30	0	1	0	0	0	1440
7	9.937	4.969	3	30	0	0	0	1	0	1440
8	-4.604	6.053	3	30	0	0	0	1	0	1440
9	-5.029	5.07	3	30	0	1	0	0	53	68
10	-9.264	-7.979	3	30	0	1	0	0	105	120
11	8.738	-9.966	3	30	0	1	0	0	74	89
12	-3.42	7.617	3	30	0	0	1	0	3	18
13	-5.865	-4.571	3	30	1	1	0	0	172	187
14	6.3	5.322	3	30	0	0	0	1	61	76
15	-2.219	-7.605	3	30	0	1	0	0	179	194
16	-1.654	-7.838	3	30	0	1	0	0	154	169
17	-1.548	-4.124	3	0	0	0	0	-1	138	153
18	-4.818	6.259	3	0	0	-1	0	0	125	140
19	-7.389	0.376	3	0	0	-1	0	0	95	110
20	6.07	-0.549	3	0	0	0	-1	0	195	210
21	-1.663	6.171	3	0	0	-1	0	0	103	118
22	-0.665	-1.272	3	0	0	-1	0	0	103	118
23	-5.005	0.918	3	0	0	0	0	-1	142	157
24	6.2	-7.679	3	0	0	0	0	-1	157	172
25	5.575	-7.408	3	0	0	-1	0	0	0	1440
26	-7.754	3.535	3	0	0	-1	0	0	0	1440
27	2.111	1.439	3	0	0	-1	0	0	0	1440
28	1.706	-1.733	3	0	0	0	-1	0	0	1440
29	-1.796	9.104	3	0	-1	-1	0	0	0	1440
30	-5.5	2.353	3	0	0	0	0	-1	0	1440
31	7.323	-7.149	3	0	0	-1	0	0	0	1440
32	-6.413	0.367	3	0	0	-1	0	0	0	1440

使用改进模拟退火算法求解算例Ea4-16,最大迭代次数为500,最终用时3.27秒,最佳解收敛过程如图2所示,在第255次迭代时得到最佳解265.56,得出的最佳接送路径如表5和图3所示.

通过算例Ea4-16的求解结果可知,本文针对多灵活车场多类型叫车接送问题,给出的改进模拟退火算

法能够在考虑车辆资源配备情况、车辆最长行驶时间、乘客最长乘车时间、时间窗设置等的基础上,得到满足乘客接送需求的最佳车辆路径方案.此外,从图3可以看出,车辆1,2在接送结束后并没有返回其出发车场,而是返回距离接送结束点最近的车场,从而使得总行驶距离更小,体现了灵活车场的优势.

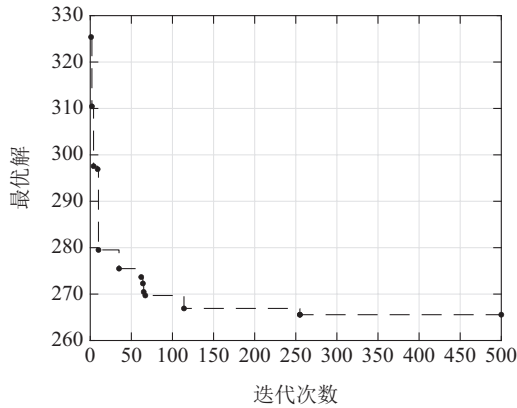


图 2 算例Ea4-16的最佳解收敛过程

Fig. 2 The optimal solution convergence process of example Ea4-16

### 5.2 不同算法求解效率对比

为了评估本文算法的性能, 将ISA与CPLEX求解器、确定性退火算法(DA)<sup>[21]</sup>进行对比实验. 其中, 精

确求解方面, 运用Fang等<sup>[20]</sup>建立的混合整数非线性规划模型, 借助数学规划求解器 CPLEX12.9 进行最优值求解. 设定 CPLEX 最大求解时间为3600秒, 若无法得到最优解, 则记录最小可行解. 另外, 确定性模拟退火算法由Brackens等提出, 是一种针对MDHDARP的启发式算法. 修改DA算法中方案有效性判断规则, 使灵活车场下的方案能够被接受, 满足 MDHDARP-FD 场景.

将全部22个算例作为求解对象, 为了减小算法的偶然性对实验结果的影响, 每个算例下各算法计算5次. 参照文献 [24], ISA, DA对于每个算例最大计算时间=车辆数量 $|K|$ ×乘客数量 $|P|$ ×2(毫秒). 分别记录算例在不同求解方法下求解目标值的最大值、最小值、平均值, 结果见表6. 表6中, CPLEX目标值标注\*号, 表示该值为最优解; DA与ISA的加粗数据, 表示该值为该算例下CPLEX, DA, ISA所得的最佳目标值.

表 5 算例Ea4-16的最佳接送路径和行驶距离

Table 5 The optimal pick-up route and driving cost of example Ea4-16 are calculated

车辆序号	出发车场坐标	最佳接送路径	返回车场坐标	路径行驶距离
1	(-5, -5)	-6-22-16-32-13-29-	(-5, 5)	57.84
2	(5, 5)	-14-3-30-19-10-26-2-18-8-24-4-20-	(5, -5)	81.78
3	(-5, 5)	-12-28-9-25-11-7-27-23-	(-5, 5)	78.53
4	(5, -5)	-5-1-21-17-15-31-	(5, -5)	47.41

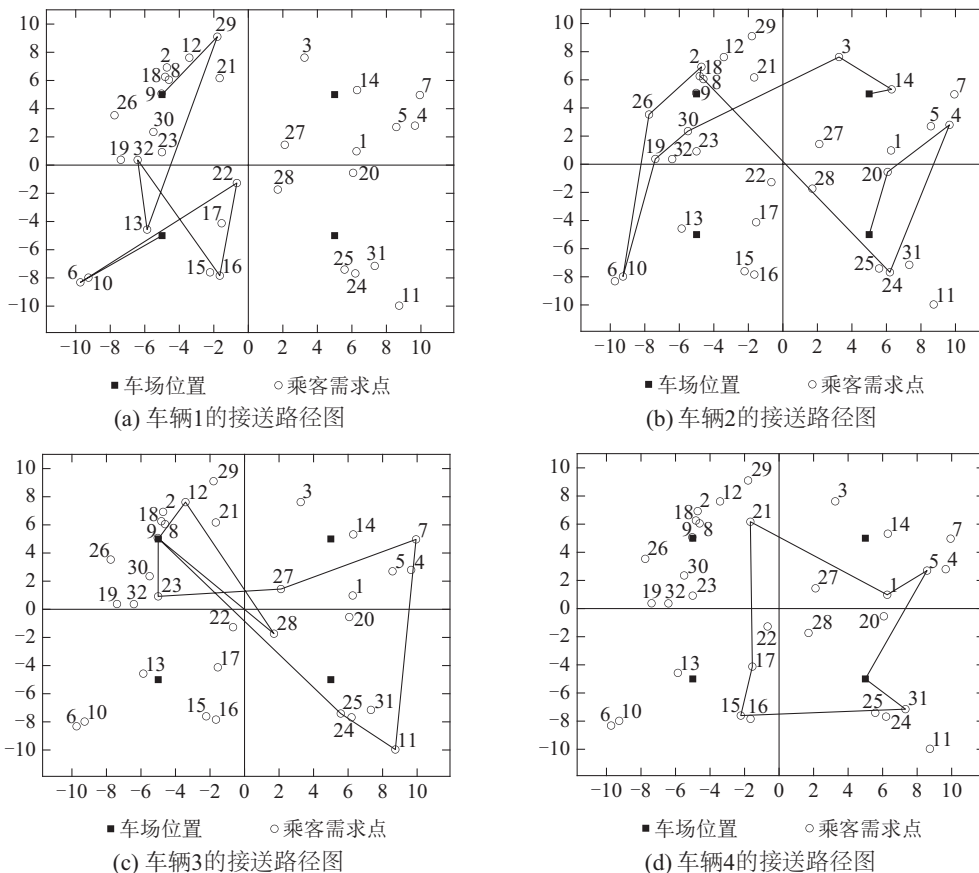


图 3 车辆接送路径图

Fig. 3 Route map of vehicle pick-up

表6 算例求解结果及比较

Table 6 Results and comparison of examples

算例	CPLEX		DA			ISA		
	时间/s	目标值	Max.	Min.	Avg.	Max.	Min.	Avg.
U2-16	1.53	284.2*	<b>284.2</b>	<b>284.2</b>	<b>284.2</b>	<b>284.2</b>	<b>284.2</b>	<b>284.2</b>
U2-20	13.39	333.6*	<b>333.6</b>	<b>333.6</b>	<b>333.6</b>	<b>333.6</b>	<b>333.6</b>	<b>333.6</b>
U2-24	29.09	426.7*	<b>426.7</b>	<b>426.7</b>	<b>426.7</b>	<b>426.7</b>	<b>426.7</b>	<b>426.7</b>
U3-18	152.95	279.0*	283.7	280.4	282.2	<b>279</b>	<b>279</b>	<b>279</b>
U3-24	3360	344.6*	350.7	347.1	348.5	346.6	<b>344.6</b>	345
U3-30	3600	469.1	475.9	473.1	474.2	474.2	471.4	472.1
U4-16	3600	252	252	252	252	242	<b>242</b>	<b>242</b>
U4-24	3600	350.3	367.3	352	359.8	352.8	<b>350</b>	351.5
U4-32	3600	无可行解	470.7	461	464.5	459.5	<b>453.7</b>	457.2
U4-40	3600	无可行解	552.8	537.1	546.3	540.1	<b>527.3</b>	531.6
U4-48	3600	无可行解	660	638.9	650.3	647.1	<b>637.4</b>	641.3
E2-16	0.51	346.8*	<b>346.8</b>	<b>346.8</b>	<b>346.8</b>	<b>346.8</b>	<b>346.8</b>	<b>346.8</b>
E2-20	3.65	359.6*	<b>359.6</b>	<b>359.6</b>	<b>359.6</b>	<b>359.6</b>	<b>359.6</b>	<b>359.6</b>
E2-24	16.76	459.8*	464.1	462.7	463	<b>459.8</b>	<b>459.8</b>	<b>459.8</b>
E3-18	9.97	286.4*	<b>286.4</b>	<b>286.4</b>	<b>286.4</b>	<b>286.4</b>	<b>286.4</b>	<b>286.4</b>
E3-24	57.94	355.9*	<b>355.9</b>	<b>355.9</b>	<b>355.9</b>	<b>355.9</b>	<b>355.9</b>	<b>355.9</b>
E3-30	3600	500.9	<b>500.9</b>	<b>500.9</b>	<b>500.9</b>	501.7	<b>500.9</b>	501.1
E4-16	115.57	265.5*	<b>265.5</b>	<b>265.5</b>	<b>265.5</b>	<b>265.5</b>	<b>265.5</b>	<b>265.5</b>
E4-24	3600	378.6	381.8	375.3	378	374	<b>371.1</b>	372.6
E4-32	3600	502.7	495	489.7	492.4	484.3	<b>479.7</b>	482.2
E4-40	3600	589.1	582.3	570.6	575.4	566.6	<b>562</b>	563.2
E4-48	3600	无可行解	705.7	691.4	699.2	695.7	<b>688.9</b>	692.1

对表6实验数据进行分析,可以发现,对比CPLEX求解器,小规模算例中,ISA在短时间内的多次运算测试中,每次均能得到最优解.中等规模算例中,ISA算法得到的解目标值表现均较好,与CPLEX较长时间求得的最优解差距极小.大规模算例中,CPLEX无法在有效时间内获得最优解甚至无解,而ISA算法求解时间远低于CPLEX,且普遍比CPLEX长时间求得的解更好.

对比DA算法,两个算法在中小规模算例中表现相近,然而当算例规模增大时,ISA在同一算例的多次求解中,目标值最大值、最小值、平均值更小,得到解的表现更好.由于采用了回温策略,避免了算法在类似温度多次求解,ISA算法有更好的跳出局部最优的能力,并且针对MDHDARP-FD场景设计的3种算子,能更灵活地为方案匹配合适的车场.

### 5.3 敏感性分析

#### 5.3.1 乘客最大乘车时间敏感性分析

乘客的最大乘车时间会影响方案中车辆的路径规划,本文算例中乘客最大乘车时间为30 min.为了衡量乘客最大乘车时间的影响,以算例E4-16, E4-24,

E4-32为对象,将乘客最大乘车时间每间隔5 min从20 min变化至60 min,对每个乘客最大乘车时间,ISA运行5次,以其中最优化方案为代表,实验结果见图4.

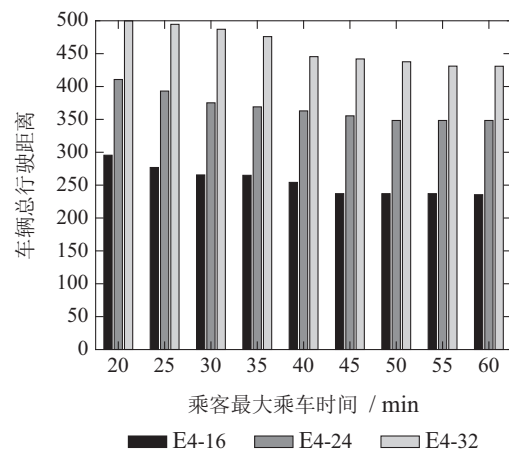


图4 乘客最大乘车时间敏感性分析结果

Fig. 4 Results of sensitivity analysis of maximum passenger travel time

可以发现,乘客最大乘车时间的增加会使总行驶距离不断下降,直到增加到一定程度后,总行驶距离趋于稳定.由于现实中的乘客评价等因素,乘客最大

乘车时间往往考虑群体中的较大值, 因此, 用更精确的乘客最大乘车时间来代替粗略的估计可以带来更小的总行驶距离。

### 5.3.2 时间窗敏感性分析

本文算例中出发, 到达点的时间窗长度为15 min, 通过调整时间窗下界, 研究不同时间窗长度对方案的影响。以算例 E4-16, E4-24, E4-32为对象, 将出发与到达点的时间窗长度每间隔5 min从5 min变化至45 min, 对于每个时间窗长度, ISA运行5次, 以其中最优方案为代表, 结果见图5。

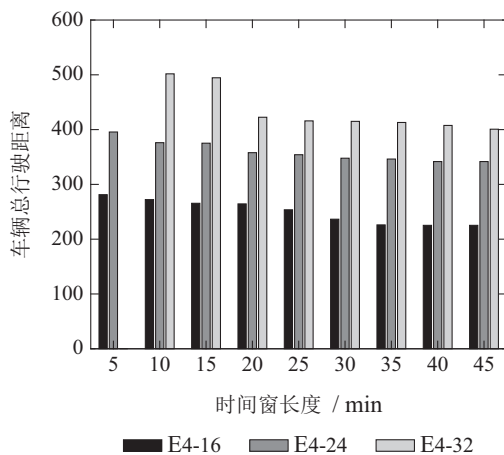


图5 时间窗敏感性分析结果

Fig. 5 Results of sensitivity analysis of time window

图5中, 在时间窗长度为5 min的情况下算例E4-32无解。可以发现, 随着时间窗下界的增加, 总行驶距离呈下降趋势, 特别是算例 E4-32在15 min算例(即原始算例)与20 min算例中, 尽管时间窗长度仅增加5 min, 最小距离减小了14.56%。而当时间窗长度增加到一定程度时, 再提升长度带来的边际总行驶距离骤减。因此, 在处理MDHDARP-FD时, 需要平衡车辆的时效性与总行驶距离。

## 6 结束语

针对多灵活车场多类型的叫车接送问题(MDHDARP-FD), 本文建立了数学模型, 并在传统模拟退火算法基础上, 提出改进算法, 优化传统算法概率更新模式和温度控制方式, 采用了基于TL参数与回温机制的解接受策略, 增加了算法的稳定性, 并能更好地跳出局部最优。采用Insert-FD算子、Switch-FD算子和D&R-FD算子, 增加了算法邻域搜索效率。

通过不同规模的经典算例, 将ISA算法与CPLEX求解器、DA算法对比, 本文算法求解时间普遍远小于CPLEX求解器, 且在中小规模算例中可以得到最优解, 或与最优解差距极小。在大规模算例中, ISA所得解对比DA算法更加稳定, 目标值表现更好, 验证了本文算法的优良性能。通过敏感性分析结果表明, 更精确地评估最大乘客乘车时间有可能有效地减小最大

车辆总行驶距离, 且在时效性与总行驶距离之间的关系需要决策者平衡。当然, 在实际情况下承担运输的车辆情况可能存在差异, 对于异构车辆情况下的叫车接送问题, 是需要进一步研究的内容。

## 参考文献:

- [1] STEIN D M. Scheduling dial-a-ride transportation systems. *Transportation Science*, 1978, 12(3): 232 – 249.
- [2] PFEIFFER C, SCHULZ A. An ALNS algorithm for the static dial-a-ride problem with ride and waiting time minimization. *OR Spectrum*, 2022, 44(1): 87 – 119.
- [3] GSCHWIND T, DREXL M. Adaptive large neighborhood search with a constant-time feasibility test for the dial-a-ride problem. *Transportation Science*, 2019, 53(2): 480 – 491.
- [4] MOLENBRUCH Y, BRAEKERS K, CARIS A. Typology and literature review for dial-a-ride problems. *Annals of Operations Research*, 2017, 259(1/2): 295 – 325.
- [5] RIST Y, FORBES M. A column generation and combinatorialenders decomposition algorithm for the selective dial-a-ride-problem. *Computers & Operations Research*, 2022, 140: 105649.
- [6] FAHMY S A. Hybrid and approximate algorithms for the dial-a-ride problem with adaptive ride times considering different service strategies. *Engineering Optimization*. 2023, 55(8): 1263 – 1277.
- [7] GRINSHPOUN T, SHUFAN E, ILANI H, et al. Graph representation of the fixed route dial-a-ride problem. *Journal of Scheduling*, 2022, 26(5): 479 – 495.
- [8] DETTI P, PAPALINI F, LARA G Z M D. A multi-depot dial-a-ride problem with heterogeneous vehicles and compatibility constraints in healthcare. *Omega*, 2016, 70: 1 – 14.
- [9] TÓTH M, HAJBA T, HORVATH A. MILP models of a patient transportation problem. *Central European Journal of Operations Research*, 2024, 32(4): 903 – 922.
- [10] HO S C, SZETO W Y, KUO Y H, et al. A survey of dial-a-ride problems: Literature review and recent developments. *Transportation Research Part B – Methodological*, 2018, 111: 395 – 421.
- [11] MASMOUDI M A, HOSNY M, DEMIR E, et al. Hybrid adaptive large neighborhood search algorithm for the mixed fleet heterogeneous dial-a-ride problem. *Journal of Heuristics*, 2020, 26(1): 83 – 118.
- [12] CHU J C Y, CHEN A Y, SHIH H H. Stochastic programming model for integrating bus network design and dial-a-ride scheduling. *Transportation Letters-the International Journal of Transportation Research*, 2022, 14(3): 245 – 257.
- [13] KOC C, LAPORTE G, TUKENMEZ I. A review of vehicle routing with simultaneous pickup and delivery. *Computers & Operations Research*, 2020, 122: 104987.
- [14] TAN S Y, YEH W C. The vehicle routing problem: State-of-the-art classification and review. *Applied Sciences-Basel*, 2021, 11(21): 10295.
- [15] KEK A G H, CHEU R L, MENG Q. Distance-constrained capacitated vehicle routing problems with flexible assignment of start and end depots. *Mathematical and Computer Modelling*, 2008, 47: 140 – 152.
- [16] MARKOV I, VARONE S, BIERLAIRE M. Integrating a heterogeneous fixed fleet and a flexible assignment of destination depots in the waste collection VRP with intermediate facilities. *Transportation Research Part B – Methodological*, 2016, 84: 256 – 273.
- [17] LI J, LI Y, PARADALOS P M. Multi-depot vehicle routing problem with time windows under shared depot resources. *Journal of Combinatorial Optimization*, 2016, 31(2): 515 – 532.

- [18] KANTASA-ARD A, CHARGUI T, BEKRAR A, et al. Dynamic multiple Depots vehicle routing in the physical internet context. *IFAC Pa-personline*, 2021, 54(1): 92 – 97.
- [19] CHE A D, WANG W J, MU X H, et al. Tabu-based adaptive large neighborhood search for multi-depot petrol station replenishment with open inter-depot routes. *IEEE Transactions on Intelligent Transportation Systems*, 2023, 24(1): 316 – 330.
- [20] FANG Y F, CHEN K J, CHU F, et al. Model for the multiple depots heterogeneous dial-a-ride problem with flexible depots. *International Conference on Service Systems and Service Management*, 2017: 99 – 103.
- [21] BRAEKERS K, CARIS A, JANSSENS G K. Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots. *Transportation Research Part B – Methodological*, 2014, 67(9): 166 – 186.
- [22] ALVAREZ A, MUNARI P, MORABITO R. Iterated local search and simulated annealing algorithms for the inventory routing problem. *International Transactions in Operational Research*, 2018, 25(6): 1785 – 1809.
- [23] ROSHANI A, GIGLIO D. Simulated annealing algorithms for the multi-manned assembly line balancing problem: Minimizing cycle time. *International Journal of Production Research*, 2017, 55(10): 2731 – 2751.
- [24] RIAHI V, NEWTON M A H, POLASH M M A, et al. Constraint guided search for aircraft sequencing. *Expert Systems with Applications*, 2019, 118: 440 – 458.

#### 作者简介:

**陈可嘉** 教授, 博士, 博士生导师, 目前研究方向为工业工程、智能控制, E-mail: kjchen@fzu.edu.cn;

**司徒腾宽** 硕士研究生, 目前研究方向为智能优化算法研究;

**方云飞** 副教授, 博士, 硕士生导师, 目前研究方向为物流系统工程与管理, E-mail: yf.fang@fzu.edu.cn;

**陈荣晖** 硕士研究生, 目前研究方向为系统工程.