

面向预约拼车服务的混合车队订单指派与路径规划

李 想, 李沁轩, 马红光*

(北京化工大学经济管理学院, 北京 100029)

摘要: 针对电动汽车和燃油汽车混合车队的预约拼车订单指派与路径规划问题, 构建了以最小化运营成本和碳排放成本为目标的混合整数线性规划模型, 提出了一种包含多组破坏与修复算子的改进自适应大邻域搜索(IALNS)算法. 数值算例验证了模型与算法的有效性: 1) 对于小规模问题, IALNS 算法可以快速收敛至近似最优解. 2) 对于大规模问题, IALNS 算法相比于变邻域搜索算法和大邻域搜索算法能够分别节约成本 10.01% 和 13.55%, 并降低计算时间 8.99% 和 22.58%. 3) 电池容量和碳排放价格对于电动汽车的使用均有正向影响. 当电池容量大幅提升时, 碳排放价格的影响减弱. 研究结果可为拼车服务行业的运营管理和政策制定提供决策支持.

关键词: 拼车服务; 混合车队; 路径规划; 碳排放; 大邻域搜索算法

中图分类号: TP273 文献标识码: A 文章编号: 1000-5781(2025)02-0307-14

doi: 10.13383/j.cnki.jse.2025.02.010

Mixed-fleet order assignment and route planning for reserved ride-sharing service

Li Xiang, Li Qinxuan, Ma Hongguang*

(School of Economics and Management, Beijing University of Chemical Technology, Beijing 100029, China)

Abstract: This paper addresses the order assignment and route planning problem for a mixed fleet with electric vehicles and gasoline vehicles to service reserved ride-sharing orders. A mixed integer linear programming model is built to minimize the operating cost and carbon emission cost, and an improved adaptive large neighborhood search (IALNS) algorithm with multiple groups of damage and repair operators is constructed. Numerical experiments demonstrate the effectiveness of the proposed model and algorithm: 1) For small-scale instances, IALNS algorithm can quickly converge to approximate optimal solution. 2) For large-scale instances, compared to variable neighborhood search algorithm and large neighborhood search algorithm, IALNS algorithm can reduce cost by 10.01% and 13.55%, and save computing time by 8.99% and 22.58% respectively. 3) Battery capacity and carbon emission price have positive impacts on the use of electric vehicles. When the battery capacity is significantly increased, the impact of carbon emission price is reduced. These results provide decision support for the operation management and policy formulation of ride-sharing service industry.

Key words: ride-sharing; mixed fleet; route planning; carbon emission; large neighborhood search algorithm

1 引言

“拼车”是我国一种新兴的共享出行模式. 近年来, 滴滴出行、嘀嗒出行和首汽约车等公司先后推出了拼车服务. 传统出租车服务模式下, 乘客乘坐车辆所支付的费用对应整个车辆空间. 而拼车服务模式, 公司

收稿日期: 2021-10-08; 修订日期: 2025-01-14.

基金项目: 国家自然科学基金资助项目(71931001; W2411066; U2469202; 72401019); 北京化工大学一流学科建设专项资金资助项目(XK1802-5).

*通信作者

将车辆的空间售卖给多位具有相似行程的乘客,乘客按照各自的行程支付相应比例的费用.拼车可以降低乘客出行成本,提高车辆利用率,实现乘客与公司双赢,并且还能显著提升城市交通效率,缓解城市交通拥堵,降低城市交通排放等^[1].据市场调研机构 QYResearch 发布的《2021—2027 全球与中国拼车市场现状及未来发展趋势》报告,截至 2019 年底,我国拼车服务累计使用人次 29 亿,市场规模达 181 亿元,预计到 2025 年将达到 1 148 亿元.

对于乘客而言,拼车服务虽然能够节省出行成本,但也降低了出行体验,可能需要承受长时间等待和长距离绕行.2016 年 11 月份成都市的滴滴出行订单数据显示,90% 的拼车订单因候车或绕行发生了延误^[2].深圳舆情研究院发布的《关于深圳市网约车服务的社会满意度调查》报告指出,62.5% 的被调查者出于便捷性与舒适性的考量不会选择拼车.因此,如何实现精准的订单指派与合理的路径规划,提升拼车服务模式的出行体验,是长期困扰拼车服务公司的运营管理难题.

现阶段,我国拼车服务公司使用的车辆主要为燃油汽车.然而,随着我国“双碳”目标的提出,交通行业节能减排压力越来越大,电动汽车的发展和进入加速阶段.根据国务院办公厅印发的《新能源汽车产业发展规划(2021—2035年)》,2035 年电动汽车将成为新销售车辆的主流.在此背景下,未来很长一段时间内电动汽车与燃油汽车并存将成为常态.通过调研多家拼车服务公司了解到,由于电动汽车和燃油汽车在续航里程和续能方式上存在很大差异,二者并存必然会给订单指派和路径规划带来更大的挑战^[3].

国内外学者面向拼车服务的订单指派和路径规划方面进行了广泛研究. Santos 等^[4]等以最小化乘客出行成本和最大化订单完成数量为目标,采用基于插队操作的局部搜索算法对拼车路径规划问题进行求解. Peng 等^[5]等针对面向预约订单的通勤拼车问题,构建了以最小化司机和乘客的出行成本、时间成本及司机奖励为目标的订单指派和路径规划模型,并设计了延迟接受算法进行求解. Cao 等^[6]以最小化出租车行驶距离为目标,综合考虑司机和乘客的利益,在拼车路径规划模型中加入了乘客利益、驾驶员利益和乘客最大等待时间等约束,采用遗传算法对拼车路线进行优化. Chien 等^[7]针对非固定线路公交拼车问题,构建了以最小化乘客等待和乘车时间为目标的整数线性规划模型,并开发了两种分别基于最少等待时间和最短路策略的模拟退火算法. 刘潇等^[8]针对预约拼车模式下的路径规划问题,提出一种改进的 Bellman-Ford 算法,将到达时间和地理位置相似度较高的停靠点归为一类,类内进行最短路径规划,类间根据到达时间信息进行排列粘合. 张玺君等^[9]构建了以最大化搭载率为主要目标、以最小化行驶距离为次要目标、带有车辆容量约束和行驶距离约束的出租车拼车路径规划模型,运用共享策略使种群中较优个体不被破坏,增强遗传算法的全局寻优能力.

上述文献集中于研究单一车型的订单指派和路径规划问题,而面向异构车队的研究更加符合拼车服务的实际运营需求. Breckers 等^[10]考虑了在拼车服务中使用由不同容量和不同功能车辆组成的异构车队,以服务不同类型的用户. Peng 等^[11]研究了客运枢纽站的异构公交拼车路径规划问题,使用大邻域搜索算法最大化运输乘客及包裹的收益. Masmoudi 等^[12]研究了异构医疗车队的订单指派与路径规划问题,其中车辆都是电动汽车,但容量和可提供的医疗资源类型不同,并设计了三种分别结合轮盘赌选择算子、局部搜索单一算子和下降接受准则的变邻域搜索算法. 在此基础上, Malheiros 等^[13]考虑了多场站的异构医疗车队订单指派与路径规划问题,并使用大邻域搜索算法进行求解.

目前,国内外关于拼车订单指派和路径规划问题的研究大多面向由同构车辆组成的车队,虽然有少数研究是面向异构车辆组成的车队,但主要集中在车辆容量的异构,还未有文献针对燃油汽车和电动汽车组成的混合车队开展拼车订单指派和路径规划研究.因此,本文考虑一个同时拥有电动汽车和燃油汽车的拼车服务公司,研究如何实现多车型车辆与多乘客订单之间的最佳匹配、车辆路径优化及电动汽车充电计划.针对此问题,构建了以最小化混合车队的运营成本和碳排放成本为目标,以车辆载客量、电池容量和服务时长等限制为约束的混合整数线性规划模型,并开发了包含多组破坏与修复算子的改进自适应大邻域搜索算法.

本文的主要贡献包括: 1) 首次针对电动汽车和燃油汽车组成的混合车队开展拼车订单指派与路径规划研究,同时协调公司、乘客和环境三方利益,为我国拼车服务公司的运营管理提供了模型支持; 2) 鉴于混合

车队的订单匹配与路径规划属于 NP 难问题, 精确算法无法解决大规模问题^[14,15], 设计了一种改进自适应大邻域搜索算法, 为我国拼车服务公司的运营管理提供了算法支持; 3) 分析了车队配比、电池容量、碳排放价格对拼车服务公司使用电动汽车的影响, 能够为“双碳”目标下拼车服务行业的电动化发展和政策制定提供决策支持.

2 问题描述

考虑同时运营电动汽车和燃油汽车的拼车服务公司. 所有车辆的集合为 $\mathbb{K} = \mathbb{K}^E \cup \mathbb{K}^G$, 其中 \mathbb{K}^E 为电动汽车集合, \mathbb{K}^G 为燃油汽车集合. 每辆车的最大载容量相同, 记为 Q_{\max} . 燃油汽车使用汽油作为动力来源, 而电动汽车使用电池进行供能, 电池容量为 E_{\max} . 基于北汽新能源出租车公司的访谈结果, 电动汽车的电池可拆卸, 电量不足时需要到换电站进行更换, 且在目前电池容量(40 kWh)下, 电动汽车在服务过程中最多需要更换一次电池. 记该公司服务区域内的换电站个数为 m , 一个换电站拥有多条轨道能够同时为多辆电动汽车更换电池.

设该公司需要服务的拼车订单数量为 n , 且均为预约形式, 即在服务开始之前, 公司已知各订单的起点和终点. 在服务开始时, 所有车辆均从场站出发, 依次服务被指派的订单, 并最终返回场站. 燃油汽车的续航里程足够长, 在服务过程中无需加油, 而电动汽车由于电池容量有限, 在服务过程中可能需要到换电站更换电池. 本文要解决的问题是, 在满足所有订单需求的情况下, 确定订单指派关系、车辆行驶路径以及电动汽车换电方案.

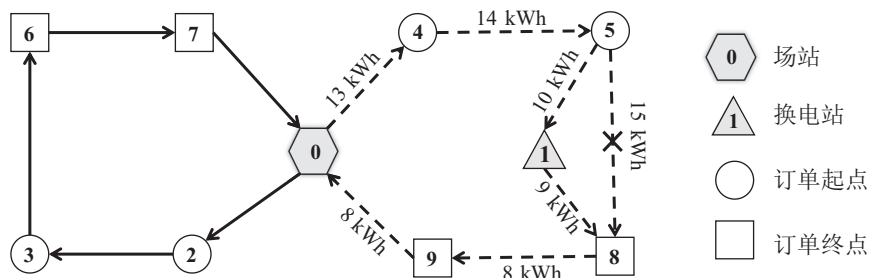


图 1 问题示例

Fig. 1 Illustrative example of the problem

为了更清晰地说明本文要解决的问题, 下面给出一个包含 1 个换电站、1 辆电动汽车(电池容量为 40 kWh)、1 辆燃油汽车及 4 个拼车订单的示例. 如图 1 所示, 节点 0 是场站, 节点 1 是换电站, 节点 2 和节点 6 分别是订单 1 的起点和终点, 节点 3 和节点 7 分别是订单 2 的起点和终点, 节点 4 和节点 8 分别是订单 3 的起点和终点, 节点 5 和节点 9 分别是订单 4 的起点和终点, 路段上的权重表示电动汽车通过该路段的耗电量. 针对此示例, 得到如下的解决方案: 燃油汽车被指派服务订单 1 和订单 2, 其行驶路径为 $0 \rightarrow 2 \rightarrow 3 \rightarrow 6 \rightarrow 7 \rightarrow 0$ (实线箭头), 即订单 1 和订单 2 需要在路段 $3 \rightarrow 6$ 上拼车; 电动汽车被指派服务订单 3 和订单 4, 由于到达节点 4 时的剩余电量($40 - 13 - 14 = 13 \text{ kWh}$) 小于直接行驶至节点 8 所需的电量(15 kWh), 需要绕行到换电站更换电池, 因此电动汽车的行驶路径为 $0 \rightarrow 4 \rightarrow 5 \rightarrow 1 \rightarrow 8 \rightarrow 9 \rightarrow 0$ (虚线箭头), 即订单 3 和订单 4 需要在路段 $5 \rightarrow 1 \rightarrow 8$ 上拼车.

3 混合车队订单指派与路径规划模型

为了生成最优订单指派与路径规划方案, 本节构建一个混合整数线性规划模型. 设场站为 0, 换电站集合为 $\mathbb{F} = \{1, 2, \dots, m\}$, 订单起点集合为 $\mathbb{P} = \{m + 1, m + 2, \dots, m + n\}$, 订单终点集合为 $\mathbb{D} = \{m + n + 1, m + n + 2, \dots, m + 2n\}$, 其中 $m + 1$ 和 $m + n + 1$ 分别代表第 1 个订单的起点和终点,

$m+2$ 和 $m+n+2$ 分别代表第 2 个订单的起点和终点, 依次类推. 此外, 为了建模方便, 添加虚拟场站 $0'$, 要求每辆车每天均从场站 0 出发, 并最终返回到虚拟场站 $0'$. 本文所研究的问题可以用有向图 $G = (N, A)$ 来表示, 其中 $N = \{0, 0'\} \cup F \cup P \cup D$ 为所有节点的集合, $A = \{(i, j) | i, j \in N, i \neq j\}$ 为所有路段的集合. 模型中使用的符号与变量见表 1.

表 1 符号说明
Table 1 Notations

符号	含义	符号	含义
$0, 0'$	场站	g_{ij}	燃油汽车通过路段 (i, j) 的行驶成本, $g_{00'} = 0$
m	换电站数量	e_{ij}	电动汽车通过路段 (i, j) 的耗电量, $e_{00'} = 0$
n	拼车订单数量	d_{ij}	燃油汽车通过路段 (i, j) 的碳排放量, $d_{00'} = 0$
F	换电站集合, $F = \{1, 2, \dots, m\}$	t_{ij}	车辆通过路段 (i, j) 的行驶时间, $t_{00'} = 0$
P	订单起点集合, $P = \{m+1, m+2, \dots, m+n\}$	E_{\max}	电池容量
D	订单终点集合, $D = \{m+n+1, m+n+2, \dots, m+2n\}$	Q_{\max}	车辆最大载容量
N	所有节点的集合, $N = \{0, 0'\} \cup F \cup P \cup D$	$L_{i(n+i)}$	从订单起点 $i \in P$ 到订单终点 $n+i \in D$ 的最大时长
A	所有路段的集合, $A = \{(i, j) i, j \in N, i \neq j\}$	T	公司每天的服务时长
\mathbb{K}^E	电动汽车集合	D_{\max}	碳排放配额
\mathbb{K}^G	燃油汽车集合	π	碳排放价格
\mathbb{K}	所有车辆的集合, $\mathbb{K} = \mathbb{K}^E \cup \mathbb{K}^G$	Q_i^k	车辆 k 离开节点 $i \in N \setminus \{0'\}$ 时的载容量
q_i	节点 $i \in N \setminus \{0'\}$ 的上/下车人数. 当 $i \in P$ 时, $q_i > 0$; 当 $i \in D$ 时, $q_i < 0$; 当 $i \in \{0\} \cup F$ 时, $q_i = 0$	E_i^k	车辆 k 离开节点 $i \in N \setminus \{0'\}$ 时的剩余电量
c_{ij}	电动汽车通过路段 (i, j) 的行驶成本, $c_{00'} = 0$	B_i^k	车辆 k 离开节点 $i \in N \setminus \{0'\}$ 时的时间
		x_{ij}^k	0-1 变量, 表示车辆 k 是否通过路段 (i, j)

3.1 目标函数

拼车服务公司在进行订单指派与路径规划时, 首先考虑的是降低车辆的运营成本, 而运营成本取决于行驶路径. 设一辆电动汽车和一辆燃油汽车通过路段 (i, j) 的行驶成本分别为 c_{ij} 和 g_{ij} , 则拼车服务公司所有车辆总的运营成本为

$$C_v = \sum_{k \in \mathbb{K}^E} \sum_{(i,j) \in A} c_{ij} x_{ij}^k + \sum_{k \in \mathbb{K}^G} \sum_{(i,j) \in A} g_{ij} x_{ij}^k, \quad (1)$$

其中第 1 项表示所有电动汽车的运营成本, 第 2 项表示所有燃油汽车的运营成本.

另外, 燃油汽车在行驶过程中会产生碳排放. 设一辆燃油汽车通过路段 (i, j) 所产生的碳排放量为 d_{ij} , 则拼车服务公司所有燃油车辆总的碳排放量为

$$D = \sum_{k \in \mathbb{K}^G} \sum_{(i,j) \in A} d_{ij} x_{ij}^k. \quad (2)$$

为了控制企业的碳排放量, 我国已于 2017 年底正式启动碳排放权交易机制. 在此机制下, 政府会向企业分配一定的碳排放额度, 而当企业的碳排放量超过其配额时, 需要从碳排放权交易市场中购买额度^[16]. 设拼车服务公司的碳排放配额为 D_{\max} , 市场中的碳排放价格为 π , 则拼车服务公司的碳排放成本为

$$C_E = \max \{ \pi (D - D_{\max}), 0 \}. \quad (3)$$

当 $D \leq D_{\max}$ 时, $C_E = 0$; 当 $D > D_{\max}$ 时, $C_E = \pi (D - D_{\max})$.

综上, 拼车服务公司的目标是最小化混合车队的运营成本与碳排放成本之和, 即

$$\text{Min } (C_v + C_E). \quad (4)$$

3.2 约束条件

拼车服务公司在进行订单指派与路径规划时需要考虑的约束条件包括 4 类: 车辆路径约束、车辆载容量约束、电池容量约束和服务时长约束.

1) 车辆路径约束

$$\sum_{k \in \mathbb{K}} \sum_{j \in \mathbb{N}} x_{ij}^k = 1, \quad \forall i \in \mathbb{P}, \quad (5)$$

$$\sum_{j \in \mathbb{N}} x_{ij}^k = \sum_{j \in \mathbb{N}} x_{j(m+n+i)}^k, \quad \forall k \in \mathbb{K}, \forall i \in \mathbb{P}, \quad (6)$$

$$\sum_{j \in \mathbb{N}} x_{0j}^k = \sum_{j \in \mathbb{N}} x_{j0'}^k = 1, \quad \forall k \in \mathbb{K}, \quad (7)$$

$$\sum_{i \in \mathbb{N}} x_{ih}^k = \sum_{j \in \mathbb{N}} x_{hj}^k, \quad \forall k \in \mathbb{K}, \forall h \in \mathbb{F} \cup \mathbb{P} \cup \mathbb{D}. \quad (8)$$

约束(5)限制每个订单起点能且仅能被访问一次; 约束(6)保证一个订单的起点和终点必须被同一车辆访问; 约束(7)要求每辆车必须从场站出发, 并最终返回到场站; 约束(8)保证车辆路径的连通性, 即一辆车到达某个中间节点后也必须从该节点离开.

2) 车辆载容量约束

$$Q_0^k = 0, \quad \forall k \in \mathbb{K}, \quad (9)$$

$$Q_i^k + q_j - M(1 - x_{ij}^k) \leq Q_j^k, \quad \forall k \in \mathbb{K}, \forall i, j \in \mathbb{N} \setminus \{0'\}, \quad (10)$$

$$Q_j^k \leq Q_{\max}, \quad \forall k \in \mathbb{K}, \forall j \in \mathbb{N} \setminus \{0'\}. \quad (11)$$

约束(9)明确车辆离开场站时处于空驶状态; 约束(10)表示车辆在服务过程中载容量的变化; 约束(11)保证车辆在服务过程中的载容量不能超过其最大载容量 Q_{\max} .

3) 电池容量约束

$$E_i^k = E_{\max}, \quad \forall k \in \mathbb{K}^E, \forall i \in \{0\} \cup \mathbb{F}, \quad (12)$$

$$E_j^k \leq E_i^k - e_{ij}^k x_{ij}^k + M(1 - x_{ij}^k), \quad \forall k \in \mathbb{K}^E, \forall i \in \mathbb{N}, \forall j \in \mathbb{N} \setminus \mathbb{F}, \quad (13)$$

$$E_i^k \geq e_{ij}^k x_{ij}^k, \quad \forall k \in \mathbb{K}^E, \forall i \in \mathbb{N} \setminus \{0'\}, \forall j \in \mathbb{N}. \quad (14)$$

约束(12)明确电动汽车离开场站和换电站时处于满电状态; 约束(13)表示电动汽车在服务过程中剩余电量的变化; 约束(14)保证电动汽车离开某个节点时的剩余电量不低于其到达后继节点所需的耗电量.

4) 服务时长约束

$$B_j^k \geq B_i^k + t_{ij} - M(1 - x_{ij}^k), \quad \forall k \in \mathbb{K}, \forall i, j \in \mathbb{N} \setminus \{0'\}, \quad (15)$$

$$B_i^k \leq B_{n+i}^k, \quad \forall k \in \mathbb{K}, \forall i \in \mathbb{P}, \quad (16)$$

$$B_{n+i}^k - B_i^k \leq L_{i(n+i)}, \quad \forall k \in \mathbb{K}, \forall i \in \mathbb{P}, \quad (17)$$

$$B_i^k + t_{i0'} - B_0^k \leq T, \quad \forall k \in \mathbb{K}, \forall i \in \mathbb{N} \setminus \{0'\}. \quad (18)$$

约束(15)保证时间的不可逆性, 即车辆到达后继节点的时间一定不小于其离开前序节点的时间再加上路段所需的行驶时间, 其中 M 是一个非常大的数. 这里忽略了乘客上下车所需的时间和电动汽车更换电池所需的时间; 约束(16)保证订单的起点和终点在车辆路径中的先后顺序, 即终点一定在起点之后被访问; 约束(17)限制每个订单的服务时长不超过其最大服务时长 $L_{i(n+i)}$; 约束(18)限制每个车辆完成所有服务的耗时不超过公司每天的服务时长.

3.3 混合整数线性规划模型

综上, 本节构建了一个以最小化混合车队运营成本和碳排放成本为目标, 满足车辆路径约束、车辆载客

量约束、电池容量约束和服务时长约束的混合整数规划模型. 由于目标函数中含有非线性成分 C_E , 且是取最小值, 引入变量 \bar{C}_E 将其等价如下线性约束

$$\bar{C}_E \geq 0, \quad (19)$$

$$\bar{C}_E \geq \pi \left(D - D_{\max} \right). \quad (20)$$

经过以上线性化处理, 最终得到如下的混合整数线性规划模型

$$\begin{aligned} & \text{Min } (C_v + \bar{C}_E), \\ & \text{s.t. 式(5) ~ 式(18)}. \end{aligned} \quad (21)$$

模型(21)包含 $(m + 2n + 1) \times (m + 2n + 2) \times |\mathbb{K}|$ 个 0-1 整数决策变量和 $3 \times (m + 2n + 1) \times |\mathbb{K}|$ 个连续决策变量, 以及 $2 \times (m + 2n) \times (m + 2n + 2) \times |\mathbb{K}| + 3 \times (m + 3n + 2) \times |\mathbb{K}| + n + [(m + 2n + 2) \times (m + 4n + 1) + m + 1] \times |\mathbb{K}^E|$ 个约束条件, 其复杂度与换电站数量 m 、拼车订单数量 n 和车辆数量 $|\mathbb{K}|$ 正相关. 在拼车服务公司的实际运营中, 众多订单与车辆的组合使得模型复杂度急剧上升, 精确算法难以对其有效求解, 因此下节设计一种改进自适应大邻域搜索算法.

4 改进的自适应大邻域搜索算法

自适应大邻域搜索(adaptive large neighborhood search, ALNS)算法是 Ropke 等^[17]在 2006 年基于大邻域搜索算法提出的一种启发式算法, 其主要思想是运用多组破坏和修复算子扩大对邻域解的搜索范围, 同时通过评估算子对解的改善效果来调整算子权重, 从而优化算子在迭代搜索过程中的使用频率, 以便更快地找到更好的邻域解. 相比于其他邻域搜索算法 ALNS 算法具有搜索范围大、收敛速度快等优点, 在拼车订单指派和路径规划领域已经得到了广泛应用^[18-20].

本节设计了一种改进的 ALNS(improved ALNS, IALNS)算法来求解混合车队的拼车订单指派与路径规划问题. IALNS 算法通过随机指派与修正操作生成初始解, 并设计了 10 组破坏和修复算子. 在每次迭代中, 根据算子权重, 采用轮盘赌方式从 10 组破坏和修复算子中选择一组算子对初始解进行操作, 生成当前解, 并利用 Metropolis 准则判别是否接受其作为下一次迭代的初始解. 在每次迭代后, 按照一定规则更新算子权重, 从而实现了迭代过程中算子的自适应选择. 算法流程如下:

步骤 1 生成初始解 x , 给定初始温度 T_0 、降温速率 t 和各算子的初始权重, 令最优解 $x^{\text{best}} \leftarrow x$, 当前温度 $T \leftarrow T_0$.

步骤 2 根据算子权重, 采用轮盘赌方式从多组算子中选择一组算子对 x 进行操作, 得到当前解 x' .

步骤 3 当 $f(x') < f(x^{\text{best}})$ 时, 令 $x^{\text{best}} \leftarrow x'$, $x \leftarrow x'$, 并更新所选择算子的权重; 当 $f(x^{\text{best}}) \leq f(x') < f(x)$ 时, 令 $x \leftarrow x'$, 并判断 x' 是否为新解. 若 x' 是新解, 则更新所选择算子的权重; 当 $f(x') \geq f(x)$ 时, 利用 Metropolis 准则判别是否接受 x' . 若接受 x' , 则令 $x \leftarrow x'$, 并更新所选择算子的权重.

步骤 4 更新当前温度 $T \leftarrow Tt$.

步骤 5 检测是否达到迭代停止条件, 如果是, 输出 x^{best} , 否则返回步骤 2.

4.1 初始解的生成

初始解需要满足电池容量、车辆最大载客量和服务时长等限制条件, 具体生成方法共包括 3 步:

第 1 步: 忽略电动汽车的电池容量限制, 将所有订单随机指派给电动汽车或燃油汽车, 并按指派顺序将订单起点和终点依次插入到车辆路径中. 当插入一个订单的起点和终点后, 如果车辆的载客量超过最大载客量, 则重新插入. 完成一个订单插入操作后, 重复上述过程直到插入全部订单为止, 由此得到一个满足车辆最大载客量限制的订单指派与车辆路径方案.

第 2 步: 考虑电动汽车的电池容量限制, 需要识别当前电动汽车的路径是否可行, 若不可行则需要路径中插入换电站执行更换电池操作. 在电动汽车 k 的路径 r_k 中, 依次计算通过每个节点的剩余电量 y_i^k , 若 $y_i^k \leq 0$, 则 i 为该路径的断点, 执行以下操作: 在断点与前续节点之间插入距离前续节点最近的换电站, 确保电动汽车的剩余电量可以到达该换电站, 否则再前移一点插入换电站. 重复上述操作, 直到路径无断点为止, 由此得到一个满足车辆最大载客量约束和电池容量约束的订单指派与车辆路径方案.

第 3 步: 依次计算每个订单的服务时长, 若超过其最大服务时长, 则将超时订单的终点向前移动一位. 重复上述操作, 直到所有订单均不超时. 依次计算每辆车完成所有订单的总时间, 若超过计划周期时长 T , 则将超时订单随机插入未超时车辆的路径中. 重复上述操作, 直到所有车辆均不超时. 由此得到一个满足电池容量限制、车辆最大载客量限制和服务时长限制的初始解.

4.2 破坏和修复算子

由于一个订单的起点和终点是成对出现的, 破坏操作可能导致解的不可行, 需要进行针对性修复. 参考文献[19, 21], 采用破坏和修复组合算子, 包括 6 组针对一条路径操作的算子、3 组针对两条路径操作的算子和 1 组针对 3 条路径操作的算子(见图 3).

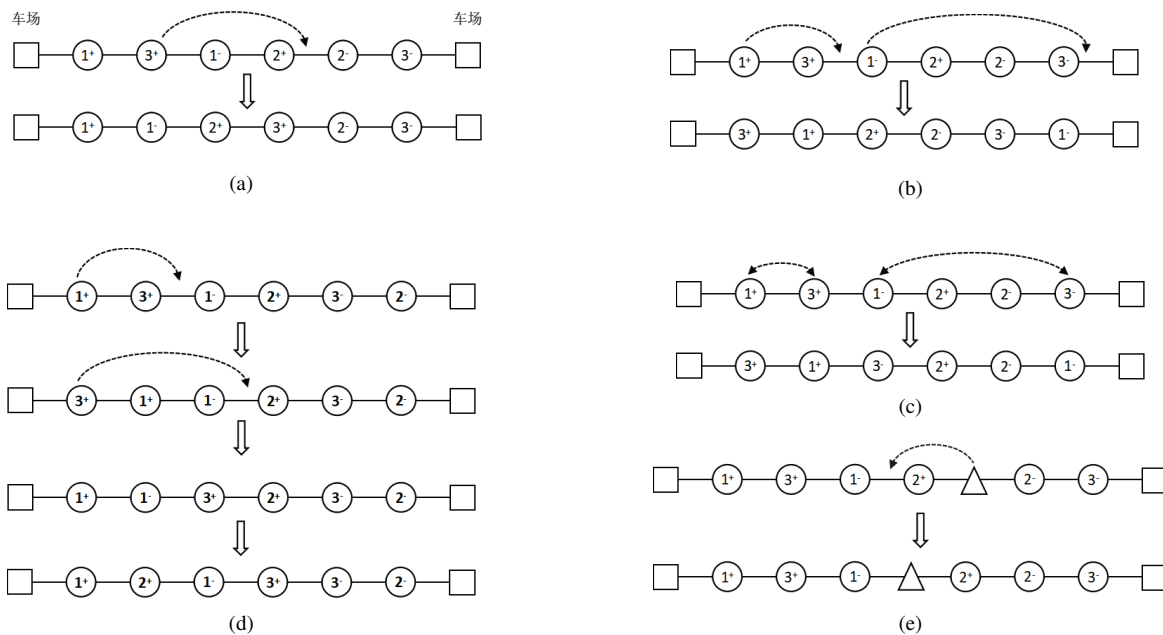


图 3 针对一条路径操作的算子示意图

Fig. 3 Illustrative of operators for a route

1) 针对一条路径操作的算子

①移动节点算子: 随机选取一条订单数大于等于 2 的车辆路径. 随机抽取该路径上一个节点, 若此节点是某个订单的起点(终点), 则移动此订单的终点(起点)到此节点后(前)的随机位置(如图 3(a)).

②移动订单算子: 随机选取一条订单数大于等于 2 的车辆路径. 随机抽出该路径上一个订单的起点和终点, 先随机插入起点, 再随机插入终点到起点后的位置(如图 3(b)).

③交换订单算子: 随机选取一条订单数大于等于 2 的车辆路径. 随机抽出这条路径上的两个订单, 交换两个订单的位置(如图 3(c)).

④起点重新插入算子: 随机选取 1 条订单数大于等于 2 的车辆路径. 固定该条路径上所有订单的终点不动, 按序逐次将这些订单的起点随机插入到其订单终点之前(如图 3(d)).

⑤终点重新插入算子: 随机选取1条订单数大于等于2的车辆路径. 固定该条路径上所有订单的起点不动, 按序逐次将这些订单的终点随机插入到其订单起点之后, 操作原理同④.

⑥移动换电站算子: 与其他算子不同, 此算子是针对换电站位置的操作. 随机抽取一条包含换电站的电动汽车路径, 删除换电站; 选择最佳的相邻节点, 并插入一个距离该相邻节点总路程最小的换电站, 此处最佳是指总成本最小且确保路径不会出现断点(如图3e).

2) 针对两条路径操作的算子

⑦订单移动算子: 随机选取1条车辆路径, 从其中随机抽出1个订单插入另1条时间余量充足的车辆路径. 时间余量充足是指车辆总行驶时长未超过预设的可插入订单时长 $Insert_{max}$ (如图4(a)).

⑧交换订单算子: 随机选取2条不同的车辆路径, 从这2条路径中各随机选取一个订单, 交换两个订单位置(如图4(b)).

⑨燃油汽车订单插入算子: 随机选取1条燃油汽车路径, 从中随机抽取1个订单; 再随机选取一条时间余量充足的电动汽车路径, 将该订单插入. 操作原理同⑦.

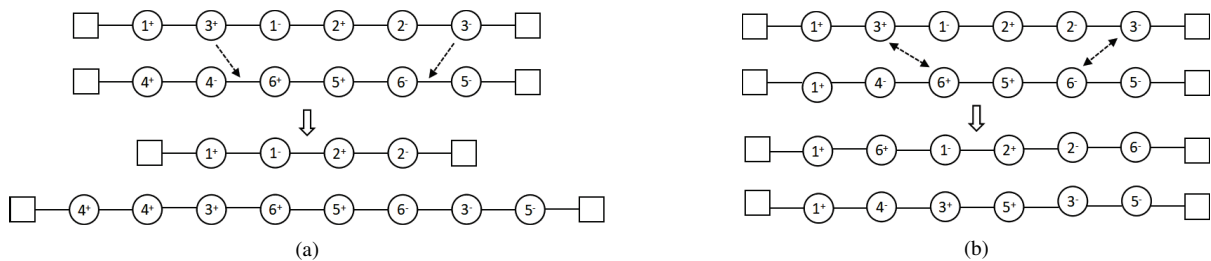


图4 针对两条路径操作的算子示意图

Fig. 4 Illustrative of operators for two routes

3) 针对三条路径操作的算子

⑩交换订单算子: 随机选取三条不同的车辆路径, 从中各自随机抽取一个订单, 并交换这3个订单位置(如图5).

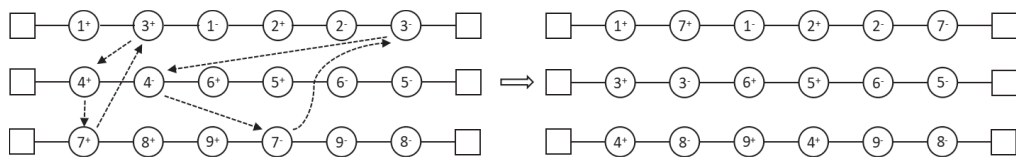


图5 针对三条路径操作的算子示意图

Fig. 5 Illustrative of an operator for three routes

每个算子的修复操作仅能确保生成的邻域解满足路径约束, 但是解可能会违反车辆载客量约束、电池容量约束和服务时长约束. 为了修正不可行的邻域解, 本文提出了以下修正程序:

1) 违反车辆载客量约束的修正程序: 将订单的起点重新插入到具有相同乘客数的其他订单的终点后, 再插入其终点. 若仍违反约束, 将订单放在该条路径的最后.

2) 违反电池容量约束的修正程序: 删除违反电池容量约束的路径中的换电站, 重新插入换电站. 插入方法同4.1节中步骤2;

3) 违反服务时长约束的修正程序: 当算子操作后的解违反服务时长约束时, 恢复为使用算子之前的解.

4.3 更新算子权重

在每次迭代中, 对算子采取自适应的选择策略. 算子选择过程取决于权重, 设所有算子的初始权重为1,

每次迭代后按照如下规则更新权重, 即

$$w_i = \begin{cases} 1, & \text{当 } \beta_i = 0 \\ (1 - \theta)w_i + \theta\beta_i/\gamma_i, & \text{当 } \beta_i \neq 0, \end{cases} \quad (22)$$

其中 w_i 为算子 i 的权重 ($i = 1, 2, \dots, 10$); γ_i 为算子 i 被选中的次数; $\theta \in [0, 1]$ 为反应因子, 用于控制算法对算子表现变化的反应速度. β_i 为算子 i 的累计得分, 每个算子的初始得分为 0, 在迭代过程中根据以下 4 种当前解的表现情况对所选择的算子得分进行更新: 1) 若当前解为新解, 且优于最优解, 则将所选择的算子分数增加 α_1 . 2) 若当前解为新解, 但不优于最优解, 优于初始解, 则将所选择的算子分数增加 α_2 . 3) 若当前解非新解, 但优于初始解, 则将所选择的算子分数增加 α_3 . 4) 若当前解不优于初始解, 但在 Metropolis 准则下接受, 则将所选择的算子分数增加 α_4 . 在情况 1) 中, 将最优解和下一次迭代的初始解替换为当前解, 在情况 2), 3), 4) 中, 将下一次迭代的初始解替换为当前解, 最优解保持不变.

4.4 Metropolis 准则

若当前解不优于初始解, 利用 Metropolis 准则来判别是否接受当前解. 设起始温度为 T_0 , 降温速率为 t , 则当前温度为 $T_i = T_0 t^i$. f_{i-1} 和 f_i 表示初始解和当前解的目标值, 则接受当前解的概率为

$$p = \exp\{(f_{i-1} - f_i)/T_i\}. \quad (23)$$

对于生成的随机数 r , 当 $p < r$ 时, 拒绝当前解, 不将其作为下一次迭代的初始解; 当 $p \geq r$ 时, 接受当前解, 将其作为下一次迭代的初始解.

4.5 迭代停止准则

停止准则用于判断是否停止算法迭代. 常用的迭代停止准则包括: 1) 设定最大迭代次数 I_{\max} , 即当迭代次数达到 I_{\max} 时停止迭代^[22]. 2) 连续迭代 V_{\max} 次得到的最优解不发生变化则停止迭代^[23]. 采用准则 1) 和准则 2) 结合的方式作为 IALNS 算法的停止准则, 即只要达到其中一个条件就停止迭代, 这样既可以控制计算时长又能保证解的质量.

5 数值实验

本节开展数值实验分析模型和算法的有效性. 对模型和算法中的参数进行设置, 利用 9 组不同规模的数值算例进行仿真实验, 对算法性能进行分析, 从服务模式、车辆配比、碳排放价格和电池容量角度, 对模型性能进行分析. IALNS 算法通过 MATLAB 2019a 编程实现, 程序运行环境为 Intel(R) Core(TM) i3-3240 处理器、主频 3.40 GHz、运行内存为 12G 的个人计算机.

5.1 参数设置

模型参数设置如下: 拼车服务公司的计划周期时长为 $T = 240 \text{ min}$; 参考北汽 EU300 换电版汽车, 并考虑电池老化导致的电池容量下降和制冷制热等额外耗电量, 设定电动汽车的电池容量为 $E_{\max} = 40 \text{ kWh}$; 根据《国家综合立体交通网规划纲要》中交通领域碳达峰目标, 设定周期内每辆燃油汽车的碳排放配额为 10 kg ^[16], 超出配额后购买碳排放额度的价格为 $\pi = 0.5 \text{ 元/kg}$. 为了保证乘客的乘车体验, 设定每个订单的最大服务时长为 $L_{i(n+i)} = 2.5t_{i(n+i)}$, 其中 $t_{i(n+i)}$ 为直接从该订单的上车点 i 到其下车点 $n+i$ 的行驶时间.

IALNS 算法的参数设置如下^[19,20]: 权重更新规则中算子分数增加值 $\alpha_1 = 0.4, \alpha_2 = 0.3, \alpha_3 = 0.2, \alpha_4 = 0.1$, 反应因子 $\theta = 0.5$; Metropolis 准则中初始温度 $T_0 = 1000$, 降温速率 $t = 0.984$; 停止准则中最大迭代次数 $I_{\max} = 1000$, 最优解连续未变化次数 $V_{\max} = 350$, 可插入订单时长 $\text{Insert}_{\max} = 200 \text{ min}$.

5.2 算法性能分析

基于北京市 200 个拼车订单的出发地、目的地、出发时间和到达时间数据, 本节随机生成 5 个小规

模和4个大规模算例来分析IALNS算法的性能.基于小规模算例,将IALNS算法与LINGO精确求解器进行比较;基于大规模算例,将IALNS算法与订单指派和路径规划问题研究中广泛使用的变邻域搜索(variable neighborhood search, VNS)算法^[3,24]和大邻域搜索(large neighborhood search, LNS)算法^[12,25,26]进行比较.最后,本节将使用Metropolis准则的IALNS算法与使用固定概率准则的IALNS算法进行比较,分析Metropolis准则的有效性.

5.2.1 小规模算例

由于IALNS属于随机搜索算法,为了消除解的随机性,使用IALNS算法对5个小规模算例(见表2)分别进行10次求解,将平均目标函数值(AVG)和平均计算时间(CPU)与LINGO进行对比,结果见表2.

从表2可以看出,变量数量和约束数量随着订单数量的增加而超线性增加.针对5个小规模算例,IALNS算法与LINGO求解器的最优值偏差均小于1%,但IALNS算法的计算时间远小于LINGO.以算例S-5为例(8个订单、2辆电动汽车、1辆燃油汽车和1个换电站),LINGO计算得到精确解需要3 932 s,而IALNS算法仅用47 s就得到了与精确解偏差只有0.52%的近似最优解,这表明IALNS算法针对小规模算例可以快速地获得高质量的解.

表2 小规模算例的计算结果
Table 2 The results of the small-scale instances

算例	订单数量	电动汽车数量 / veh	燃油汽车数量 / veh	换电站数量	变量数量	约束数量	LINGO		IALNS		与LINGO的偏差(%)
							AVG / 元	CPU / s	AVG / 元	CPU / s	
S-1	4	2	1	1	140	1 133	36.45	19	36.45	15	0.00
S-2	5	2	1	1	192	1 601	35.60	186	35.90	20	0.84
S-3	6	2	1	1	252	2 149	55.30	244	55.50	34	0.36
S-4	7	2	1	1	320	2 777	64.00	920	64.00	42	0.00
S-5	8	2	1	1	396	3 485	76.70	3 932	77.10	47	0.52

5.2.2 大规模算例

使用4个大规模算例来进一步评估IALNS算法应用于现实车队运营管理的表现,如表3所示.

表3 大规模算例
Table 3 The large-scale instances

算例	订单数量	电动汽车数量 / veh	燃油汽车数量 / veh	换电站数量	变量数量	约束数量
L-1	50	4	2	2	11 021	214 578
L-2	100	8	4	3	42 432	1 672 432
L-3	150	12	6	4	94 245	5 595 258
L-4	200	16	8	5	166 460	13 204 752

从变量和约束的数量来看,现有的商用求解器(如LINGO)无法满足实效性需求(求解小规模算例的计算时间已超过3 600 s),因此企业往往选择开发和使用启发式算法.本节将IALNS算法、VNS算法和LNS算法对每个算例各进行10次求解(LNS算法使用与IALNS算法相同的算子),记录平均计算时间(CPU)、平均目标函数值(AVG)、最小目标函数值(BEST)、AVG与BEST的偏差(DEV, $DEV = (AVG - BEST) / BEST$),以及VNS算法、LNS算法与IALNS算法在AVG方面的差距(GAP, $GAP = (AVG_{VNS/LNS} - AVG_{IALNS}) / AVG_{IALNS}$),结果见表4.

表4 大规模算例的计算结果
Table 4 The results of the large-scale instances

算例	CPU / min			AVG / 元			BEST / 元			DEV %			GAP %	
	IALNS	VNS	LNS	IALNS	VNS	LNS	IALNS	VNS	LNS	IALNS	VNS	LNS	VNS	LNS
L-1	7.19	6.21	7.35	605.89	626.90	634.20	589.78	608.08	615.12	2.73	3.09	3.10	3.47	4.67
L-2	9.97	12.25	14.47	1 209.36	1 253.75	1 365.78	1 171.19	1 207.20	1 179.90	3.26	3.86	15.75	3.67	12.93
L-3	16.37	18.58	19.43	1 775.10	1 995.80	2 095.80	1 698.60	1 723.30	1 899.70	4.09	4.21	10.36	12.43	16.07
L-4	19.95	21.92	25.77	2 469.80	2 744.55	2 856.83	2 381.77	2 563.36	2 561.04	3.70	7.07	11.52	11.12	15.64

从表 4 可以看出, 相比于 VNS 算法和 LNS 算法, IALNS 算法能够将计算时间平均降低 8.15% 和 23.81%, 更具有实效性. 当订单数量小于 100 时, IALNS 算法的计算时间都在 10 min 以内; 当订单数量大于 100 时, IALNS 算法的计算时间都未超过 20 min, 这符合实际的运营需求. 在最优性方面, IALNS 算法可以获得更高质量的解, 并且随着算例规模的增大, IALNS 算法的最优性优势更加显著. 以算例 L-4 为例, 与 VNS 算法和 LNS 算法相比, IALNS 算法能够将成本平均降低 11.12% 和 15.64%. 在求解稳定性方面, IALNS 算法在 4 个大规模算例下的 DEV 均小于 5%. 相比之下, VNS 算法和 LNS 算法没有 IALNS 算法稳定, 并且随着算例规模的增大, VNS 算法的 DEV 达到 7.07%, LNS 算法的 DEV 达到 11.52%. 由此可见, IALNS 算法表现更加稳健, 可以在更短的时间内得到运营成本更小的订单指派和路径规划方案.

5.2.3 Metropolis 准则的有效性

在 IALNS 算法中, 采用 Metropolis 准则来判别是否接受当前解. 为了检验 Metropolis 准则的有效性, 将基于 Metropolis 准则的 IALNS 算法与基于固定概率准则的 IALNS 算法进行对比. Metropolis 准则在算法迭代过程中动态调整当前解的接受概率, 而固定概率准则总是以相同相同的概率接受当前解. 针对算例 L-3, 分别将使用 0.1, 0.3, 0.5, 0.7 和 0.9 的固定概率准则的 IALNS 算法与使用 Metropolis 准则的 IALNS 算法进行对比.

由表 5 可以看出, 固定概率准则的计算时间没有明显优势, 且最优性均低于 Metropolis 准则, 与 Metropolis 准则下的目标函数值之间的平均偏差达到了 4.91%, 因此 Metropolis 准则更适合于 IALNS 算法.

表 5 Metropolis 准则与固定概率准则的对比结果

	CPU / min	AVG / 元	与 Metropolis 准则的偏差 %
Metropolis	12.37	1 775.10	—
0.1	11.54	1 848.50	4.13
0.3	12.49	1 842.20	3.78
0.5	13.80	1 835.60	3.41
0.7	12.13	1 874.60	5.61
0.9	10.23	1 910.70	7.64

5.3 模型性能分析

本节分析服务模式、电动汽车和燃油汽车配比、碳排放价格和电动汽车电池容量等因素对混合车队订单指派与路径规划的影响.

5.3.1 拼车模式与不拼车模式对比

基于 4 个大规模算例, 将拼车与不拼车两种服务模式下的计算结果进行对比. 由表 6 可以看出, 对于相同的订单数量和车辆配比, 不拼车模式下的车辆平均行驶里程、碳排放成本、总成本均大于拼车模式, 且随着订单规模增大, 拼车模式能够节约 10.74% 的总成本. 因此, 拼车模式在减少车辆行驶里程和降低碳排放方面有很大优势.

表 6 拼车模式与不拼车模式的对比

算例	是否拼车	车辆平均行驶里程 / km	碳排放成本 / 元	总成本 / 元	与拼车模式的偏差 %
L-1	是	170.00	30.90	605.89	—
	否	185.30	33.68	628.00	3.65
L-2	是	305.35	58.55	1 209.36	—
	否	335.89	64.30	1 282.16	6.02
L-3	是	568.50	100.09	1 775.10	—
	否	653.70	116.13	1 899.58	7.01
L-4	是	907.80	171.78	2 469.80	—
	否	1 116.59	211.37	2 735.10	10.74

5.3.2 电动汽车和燃油汽车配比分析

为了探究混合车队中燃油汽车和电动汽车数量配比对于订单指派与车辆路径的影响, 本节对混合车队的5种配比场景进行对比实验. 在车辆总数保持12 veh不变的情况下, 将电动汽车数量从0 veh增加12 veh, 对比5种车队配比场景下车队的总成本和碳排放成本, 结果如表7所示.

表7 不同车队配比的影响
Table 7 The impact of different fleet configurations

车辆配比	场景1	场景2	场景3	场景4	场景5
电动汽车/veh	0	3	6	9	12
燃油汽车/veh	12	9	6	3	0
总成本/元	1 493.85	1 400.92	1 340.48	1 261.65	1 172.36
碳排放成本/元	300.70	282.65	171.09	31.93	0.00

从表7可以看出, 车队从全燃油汽车过渡为全电动汽车的过程中, 运营总成本从1 493.85元减少至1 121.36元, 碳排放成本从300.70元减少至0元. 因此, 将车队中的燃油汽车更换为电动汽车可以大幅降低运营成本和碳排放成本.

为了进一步探究电动汽车对燃油汽车的替代效果, 设定车队中燃油汽车数量始终为7 veh, 而电动汽车数量从5 veh增加到9 veh. 由表8可以看出, 公司会优先使用电动汽车, 而使用的燃油汽车数量从7 veh减少到3 veh, 同时车队的运营总成本从1 347.71元减少至1 232.39元, 每更换一辆电动汽车可降低运营成本4%~7%.

表8 电动汽车数量的灵敏度分析
Table 8 Sensitivity analysis of the number of electric vehicles

电动汽车数量/veh	5	6	7	8	9
总成本/元	1 347.71	1 340.48	1 331.04	1 270.55	1 232.39
使用的电动汽车数量/veh	5	6	7	8	9
使用的燃油汽车数量/veh	7	6	5	4	3

5.3.3 碳排放价格与电池容量交叉分析

本节分析在不同碳排放价格 π 和电池容量 E_{\max} 下, 全燃油车队、混合车队和全电动车队的运营成本, 其中碳排放价格从0元/kg增加到2元/kg, 电池容量分为40 kWh和80 kWh. 算例中包含50个拼车订单, 车队均配置6辆汽车, 其中混合车队配置3辆燃油汽车和3辆电动汽车, 结果见表9.

表9 碳排放价格和电池容量的影响
Table 9 The impacts of carbon emission unit price and battery capacity

		π 元/kg	0	0.5	1	1.5	2
$E_{\max} = 40$ kWh	全燃油车队		512.08	545.59	606.62	664.23	724.98
	混合车队		548.32	568.17	601.14	635.34	666.29
	全电动车队		607.87	607.87	607.87	607.87	607.87
		π 元/kg	0	0.5	1	1.5	2
$E_{\max} = 80$ kWh	全燃油车队		512.08	545.59	606.62	664.23	724.98
	混合车队		535.46	542.23	550.59	583.23	595.96
	全电动车队		547.27	547.27	547.27	547.27	547.27

由表9可看出, 当电池容量 E_{\max} 为40 kWh时, 碳排放价格 π 为0元/kg和0.5元/kg时, 全电动车队的运营成本为607.87元, 相比于混合车队和全燃油车队成本较高, 这是因为电动汽车在运营过程因更换电池而产生绕路, 增加了运营成本. 随着碳排放价格的提高, 燃油汽车的碳排放成本逐渐增加, 当 $\pi = 1$ 元/kg时, 混合车队中只有3辆燃油汽车, 碳排放成本的增长相比于全燃油车队较小, 因此其运营成本最小; 当 $\pi = 1.5$ 元/kg时, 燃油汽车的运营成本显著增加, 此时全电动车队的运营成本优势开始凸显; 当 $\pi = 2$ 元/kg时, 相比于全燃油车队, 混合车队和全电动车队可分别节约成本8.10%和16.15%.

当电池容量 E_{\max} 提升为 80 kWh 时, 电动汽车的续航里程增加, 换电次数减少, 全电动车队的运营成本从 607.87 元降低至 547.27 元。

此外, 相比于电池容量为 40 kWh 时, 电池容量为 80 kWh 时碳排放价格增加对混合车队运营成本的影响变小。当 $\pi = 0.5$ 元/kg 时, 混合车队运营成本已经低于全燃油车队; 当 $\pi = 1$ 元/kg 时, 全电动车队的运营成本成为 3 种车队中最低的。当 $\pi = 2$ 元/kg 时, 相比于全燃油车队, 混合车队和全电动车队分别节约成本 16.15% 和 24.51%。这说明电池容量和碳排放价格对于拼车服务公司使用电动汽车都具有正向影响, 碳排放价格的增长会促使全燃油车队逐渐向混合车队, 直至全电动车队过渡, 有利于实现碳减排, 并且随着电池容量的大幅提升, 电动汽车将更具有成本优势。

6 结束语

在“双碳”目标下, 城市交通将逐步向电动化发展, 燃油汽车和电动汽车共存将成为常态。针对由燃油汽车和电动汽车组成的混合车队, 本文研究了拼车服务订单指派与路径规划问题, 构建了以最小化运营成本和碳排放成本为目标的混合整数线性规划模型, 实现了公司、乘客和环境三方利益的协调。针对异构车队的派单与径路特征提出了多组破坏与修复算子, 并采用 Metropolis 判别接受准则开发了 IALNS 算法。多组不同规模数值算例的计算结果表明, IALNS 算法能够在短时间内得到小规模问题的近似最优解(偏差小于 1%); 对于大规模问题, IALNS 算法与 VNS 算法(LNS 算法)相比能够降低计算时间 8.15%(23.81%)并节约成本 11.12%(15.64%), 且计算性能更为稳定; 相比于固定概率准则, 使用 Metropolis 准则的 IALNS 算法最优性更好。此外, 对服务模式、车队配比、碳排放价格和电池容量的分析表明, 拼车模式和电动汽车的使用能够有效降低公司成本, 且效果会随着碳排放价格的增加和电池技术的提升更加显著。本文的研究属于拼车订单指派与路径规划问题的静态优化, 未来的研究将考虑实时订单对拼车运营管理的影响, 探讨实时订单指派与路径优化方法。

参考文献:

- [1] 王茂福. 拼车的发展及其效应. 中国软科学, 2010, 11: 54–61.
Wang M F. The evolution and effects of carpooling. China Soft Science, 2010, 11: 54–61. (in Chinese)
- [2] Li W, Pu Z, Li Y, et al. Characterization of ridesplitting based on observed data: A case study of Chengdu, China. Transportation Research, Part C: Emerging Technologies, 2019, 100: 330–353.
- [3] Muelas S, LaTorre A, Peña J M. A variable neighborhood search algorithm for the optimization of a dial-a-ride problem in a large city. Expert Systems with Applications, 2013, 40: 5516–5531.
- [4] Santos D O, Xavier E C. Dynamic taxi and ridesharing: A framework and heuristics for the optimization problem // Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence. AAAI Press, 2013, 2885–2891.
- [5] Peng Z X, Shan W X, Jia P, et al. Stable ride-sharing matching for the commuters with payment design. Transportation, 2020, 47: 1–21.
- [6] Cao Y, Wang S, Li J Y. The optimization model of ride-sharing route for ride hailing considering both system optimization and user fairness. Sustainability, 2021, 13: 902.
- [7] Chien W C, Cho H H, Chang Y C, et al. Metaheuristic algorithm of multi-passengers routing path for ride-sharing vehicle. Internet of Vehicles: Technologies and Services for Smart Cities. Lecture Notes in Computer Science, 2017, 10689: 19–30.
- [8] 刘 潇, 李德敏, 张光林, 等. 预约机制下的共乘路径规划策略. 计算机工程与应用, 2016, 52(20): 92–97.
Liu X, Li D M, Zhang G L, et al. Carpool path planning strategy under reservation mechanism. Computer Engineering and Applications, 2016, 52(20): 92–97. (in Chinese)
- [9] 张玺君, 张祺瑞, 张丽娟, 等. 基于改进遗传算法的出租车共乘线路规划研究. 交通运输系统工程与信息, 2019, 19(6): 123–128.
Zhang X J, Zhang Q R, Zhang L J, et al. Taxi sharing route planning based on improved genetic algorithm. Journal of Transportation Systems Engineering and Information Technology, 2019, 19(6): 123–128. (in Chinese)

- [10] 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: 166–186.
- [11] Peng Z X, Feng R, Wang C Y, et al. Online bus-pooling service at the railway station for passengers and parcels sharing buses: A case in Dalian. *Expert Systems with Applications*, 2021, 169: 114354.
- [12] Masmoudi M A, Hosny M, Demir E, et al. The dial-a-ride problem with electric vehicles and battery swapping stations. *Transportation Research, Part E: Logistics and Transportation Review*, 2018, 118: 392–420.
- [13] Malheiros I, Ramalho R, Passeti B, et al. A hybrid algorithm for the multi-depot heterogeneous dial-a-ride problem. *Computers and Operations Research*, 2021, 129: 105196.
- [14] 靳志宏, 于波, 侯丽晓. 基于配载约束的配送优化问题及其求解算法. *系统工程学报*, 2012, 27(3): 390–398.
Jin Z H, Yu B, Hou L X. Vehicle routing optimization problem and its solution method based on vehicle loading constraints. *Journal of Systems Engineering*, 2012, 27(3): 390–398. (in Chinese)
- [15] 张源凯, 黄敏芳, 胡祥培. 网上超市订单分配与物流配送联合优化方法. *系统工程学报*, 2015, 30(2): 251–258.
Zhang Y K, Huang M F, Hu X P. Integrated optimization approach to order allocation and delivery problem of online supermarket. *Journal of Systems Engineering*, 2015, 30(2): 251–258. (in Chinese)
- [16] 冯晨鹏, 尹绍婧, 肖相泽, 等. 浙江省区域碳排放权配额分配与补偿研究. *系统工程学报*, 2020, 35(5): 577–587.
Feng C P, Yin S J, Xiao X Z, et al. Study of allocation and compensation of regional carbon emission right in Zhejiang Province. *Journal of Systems Engineering*, 2020, 35(5): 577–587. (in Chinese)
- [17] Ropke S, Pisinger D. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 2006, 40(4): 455–472.
- [18] 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.
- [19] Sun B, Sun X, Tsang D H. K, et al. Optimal battery purchasing and charging strategy at electric vehicle battery swap stations. *European Journal of Operational Research*, 2019, 279: 524–539.
- [20] 徐倩, 熊俊, 杨珍花, 等. 基于自适应大邻域搜索算法的外卖配送车辆路径优化. *工业工程与管理*, 2021, 26(3): 115–122.
Xu Q, Xiong J, Yang Z H, et al. Route optimization of takeout delivery vehicles based on adaptive large neighborhood search algorithm. *Industrial Engineering and Management*, 2021, 26(3): 115–122. (in Chinese)
- [21] Folkestad C A, Hansen N, Fagerholt K. Optimal charging and repositioning of electric vehicles in a free-floating carsharing system. *Computers and Operations Research*, 2020, 113: 104771.
- [22] 温昆, 郭鹏, 裴霞, 等. 带时间窗和调货特性的多品类取送货优化算法. *系统管理学报*, 2021, 30(5): 1005–1015.
Wen K, Zheng P, Pei X, et al. Optimization algorithm for multi-commodity pickup and delivery problem with time windows and transshipment. *Journal of Systems and Management*, 2021, 30(5): 1005–1015. (in Chinese)
- [23] 王旭坪, 董杰, 韩涛, 等. 考虑碳排放与时空距离的冷链配送路径优化研究. *系统工程学报*, 2019, 34(4): 555–565.
Wang X P, Dong J, Han T, et al. The optimization of cold chain delivery routes considering carbon emission and temporal-spatial distance. *Journal of Systems Engineering*, 2019, 34(4): 555–565. (in Chinese)
- [24] Muelas S, LaTorre A, Peña J M. A distributed VNS algorithm for optimizing dial-a-ride problems in large-scale scenarios. *Transportation Research, Part C: Emerging Technologies*, 2015, 54: 110–130.
- [25] Roberti R, Wen M. The electric traveling salesman problem with time windows. *Transportation Research, Part E: Logistics and Transportation Review*, 2016, 89: 32–52.
- [26] Lu J, Chen Y N, Hao J K, et al. The time-dependent electric vehicle routing problem: Model and solution. *Expert Systems with Applications*, 2020, 161: 113593.

作者简介:

李想(1982—), 男, 吉林四平人, 博士, 教授, 研究方向: 交通运输管理, 大数据优化等, Email: lixiang@mail.buct.edu.cn;
李沁轩(1997—), 女, 山西太原人, 硕士生, 研究方向: 共享汽车运营管理, Email: 2019200861@mail.buct.edu.cn;
马红光(1989—), 男, 河北邯郸人, 博士, 副教授, 研究方向: 交通运输管理, 运筹优化等, Email: mahg@mail.buct.edu.cn.