

多舱共配绿色车辆路径问题的改进变邻域搜索算法

肖友刚¹, 曹健¹, 陈婉茹¹, 张得志^{1†}, 李双艳²

(1. 中南大学 交通运输工程学院, 湖南 长沙 410075; 2. 中南林业科技大学 物流与交通学院, 湖南 长沙 410004)

摘要: 针对社区团购前置仓配送场景中“多中心、高时效、多品类、高排放”难题, 本文提出多车场带时间窗的绿色多舱车车辆路径问题(MDMCG-VRPTW), 构建混合整数线性规划模型, 并设计改进的变邻域搜索算法(IVNS)实现求解. 采用两阶段混合算法构造高质量初始解. 提出均衡抖动策略以充分探索解空间, 引入粒度机制以提升局部搜索阶段的寻优效率. 标准算例测试结果验证了两阶段初始解构造算法和IVNS算法的有效性. 仿真实验结果表明, 模型与算法能够有效求解MDMCGVRPTW, 且改进策略提高了算法的求解效率和全局搜索能力. 最后, 基于对配送策略和时效性的敏感性分析, 为相关配送企业降本增效提供更多决策依据.

关键词: 多舱共配; 绿色车辆路径; 均衡抖动; 粒度局部搜索; 改进变邻域搜索算法

引用格式: 肖友刚, 曹健, 陈婉茹, 等. 多舱共配绿色车辆路径问题的改进变邻域搜索算法. 控制理论与应用, 2024, 41(4): 751–762

DOI: 10.7641/CTA.2023.20182

Improved variable neighborhood search algorithm for multi-compartment green vehicle routing problem

XIAO You-gang¹, CAO Jian¹, CHEN Wan-ru¹, ZHANG De-zhi^{1†}, LI Shuang-yan²

(1. College of Transportation Engineering, Central South University, Changsha Hunan 410075, China;

2. College of Logistics and Transportation, Central South University of forestry science and technology, Changsha Hunan 410004, China)

Abstract: Focusing on the problem of “multiple depots, punctuality, multiple products and energy intensity” arising in the distribution scenario of community group purchase, this paper studies the multi-depot multi-compartment green vehicle routing problem with time windows (MDMCGVRPTW). A mixed integer linear programming (MILP) model and an improved variable neighborhood search (IVNS) algorithm are proposed. High quality initial solutions are obtained by a two-stage hybrid (2SH) algorithm. A new balanced shaking heuristic is designed to fully explore the solution space, and a granularity mechanism is introduced to improve the efficiency of local search. The 2SH algorithm and the IVNS algorithm have already demonstrated their effectiveness in solving the benchmarks. The experiment results based on the simulation examples show that the proposed model and algorithm can effectively solve the MDMCGVRPTW, and the improved strategies enhance the exploitation capability of the IVNS algorithm. Finally, some management insights for relevant distribution enterprises are given based on the sensitivity analysis of distribution strategy and timeliness to achieve cost reduction and efficiency increase.

Key words: multi-compartment distribution; green vehicle routing; balanced shaking; granular mechanism; improved variable neighborhood search

Citation: XIAO Yougang, CAO Jian, CHEN Wanru, et al. Improved variable neighborhood search algorithm for multi-compartment green vehicle routing problem. *Control Theory & Applications*, 2024, 41(4): 751–762

1 引言

随着社区团购企业规模的扩大, 前置仓模式得到推广, 其配送正面临以下难题: 其一, 需要协同调度多个配送中心的有限运力; 其二, 需要满足多样化商品对配送环境的差异化要求; 其三, 需要高效地满足客

户的即时化需求; 其四, 在“双碳”背景下, 需要进一步兼顾企业的社会效益, 实现低碳配送. 以往城市物流配送车辆路径问题的相关研究大多围绕单配送中心和单舱配送模式展开, 无法很好地解决当前社区团购配送面临的“多中心、高时效、多品类、高排放”难题.

收稿日期: 2022–03–14; 录用日期: 2023–03–08.

[†]通信作者. E-mail: dzhang@csu.edu.cn; Tel.: +86 13467562358.

本文责任编辑: 宗群.

国家自然科学基金项目(71672193), 湖南省自然科学基金项目(2021JJ30857), 湖南省社会科学基金项目(19YBA378)资助.

Supported by the National Natural Science Foundation of China (71672193), the Natural Science Foundation of Hunan Province (2021JJ30857) and the Social Science Foundation of Hunan Province (19YBA378).

基于此,本文研究多车场多舱车带时间窗的绿色车辆路径问题(multi-depot multi-compartment green vehicle routing problem with time windows, MDMCGVRPTW),是多车场车辆路径问题(multi-depot VRP, MDVRP)、多舱车车辆路径问题(multi-compartment VRP, MCVRP)和绿色车辆路径问题(green VRP, GVRP)的综合,具有理论和现实意义.

针对MDVRP的研究已形成丰富的研究成果,戚远航等人^[1]对MDVRP的初始解构造进行研究,提出泰森多边形构造初始解策略,以蝙蝠算法为核心求解经典MDVRP问题.马冰山等人^[2]结合多配送中心和纯电动物流车的特点,构建带时间窗的半开放式多车场纯电动物流车路径优化模型,并设计蚁群算法进行求解.李洋等人^[3]设计包括高斯聚类算法和增强蚁群算法在内的两阶段算法用于求解MDVRP问题, Sadati等人^[4]对带时间窗限制的MDVRP问题进行研究,设计基于禁忌准则抖动的变邻域搜索算法对问题进行求解.范厚明等人^[5]建立以派遣成本、理货成本,以及运输成本之和最小为目标的多中心开放式,同时,配集货需求可拆分车辆路径优化模型,设计混合遗传算法进行求解. Wang等人^[6]研究带时间窗和需求可拆分的MDVRP问题,以最低运营成本为目标设计混合禁忌遗传算法对问题进行求解.

在GVRP研究方面, Zhang等人^[7]对单车场GVRP进行研究,基于线性油耗计算碳排放成本并作为总成本的组成部分,设计(ribonucleic acid, RNA)蚁群算法实现求解.周鲜成等人^[8]考虑时变速度和载重对碳排放的影响,研究以碳排放成本、车辆使用成本和车窗惩罚成本之和最小为优化目标的多车场GVRP问题,设计改进蚁群算法进行求解.珠兰等人^[9]研究时间依赖型GVRP问题,提出包括油耗成本在内的优化目标,采用嵌套遗传算法对问题进行求解.部分学者综合MDVRP和GVRP的问题特点,对多车场绿色车辆路径问题进行研究,葛显龙和冉小芬^[10]建立以碳排放成本及旅行成本最小、配送时间最短为目标的双目标优化模型,并设计混合遗传禁忌搜索算法实现求解.胡蓉等人^[11]同时考虑载重、行驶速度对油耗的影响,建立以油耗最少为目标的MDVRP模型,设计融合蚁群算法实现求解. Sadati^[12]和Satay对多配送中心混合动力车的路径优化问题进行研究,考虑里程约束和加油策略,以最小距离为目标,开发混合变邻域搜索算法对模型进行优化求解.

多舱车车辆路径问题在近几年得到广泛关注, Chen和Shi^[13]研究考虑城市分时配送的多舱车车辆路径问题,提出了带时间窗的多舱车车辆路径模型,改造单车场算例,并设计了混合粒子群算法进行求解. Frank等人^[14]研究杂货零售商的周期性多温车配送问题,并使用自适应大规模邻域搜索算法实现求解.

Hebler^[15]设计一种两阶段分支定界算法,用于求解可变舱位的多舱车问题,以行驶总距离最小为目标对实例进行求解. Chen等人^[16]考虑油耗对多舱车辆调度的影响,建立考虑油耗成本的生鲜配送模型,使用自适应大规模邻域搜索算法对模型进行求解,并与人工调度算法进行对比.随后, Chen等人^[17]综合MCVRP和GVRP的问题特点,首次提出绿色多舱车车辆路径问题,建立以包括碳排放成本在内的总成本最小为目标的非线性规划模型.以上文献中MCVRP的研究主要针对单车场情景, Alinaghian和Shokouhi^[18]以距离最短为优化目标,研究了多车场多舱车车辆路径问题,但在该研究中尚未考虑时间窗的约束及碳排放对优化目标的影响.

综上所述,已有文献为MDMCVRPTW的研究奠定了基础,但是还有以下问题有待研究: 1) 现有文献大多分开对MDVRP, MCVRP和GVRP进行研究,尚未有模型综合考虑多中心、多舱共配、服务时间窗和碳排放的现实因素; 2) 考虑碳排放的现有VRP研究大多忽视了对非线性模型的线性转化,难以验证所构建模型的准确性和算法的有效性; 3) 涉及GVRP的模型中通常包含与等待时间相关的等待成本,但已有研究在使用启发式算法求解时,忽视了等待时间的决策和优化.

为解决社区团购配送企业面临的“多中心、多品类、高时效、高排放”难题,本文在数学模型构建阶段,以MDVRP模型为基础,引入多车舱容量约束和配送时间窗约束,计算配送过程中产生的固定成本、变动成本、环境成本、货损成本、制冷成本和等待成本,构建综合成本最小化的MDMCGVRPTW混合整数规划(mixed-integer linear program, MILP)模型.在模型求解阶段,根据问题特征设计两阶段混合(2-stage hybrid, 2SH)初始解构造算法和改进变邻域搜索(improved variable neighborhood search, IVNS)算法对问题进行求解.最后,使用标准算例验证的算法有效性,使用仿真数据评估模型和算法准确性和有效性,为社区团购配送企业车辆调度提供理论指导和决策依据.

2 问题与模型

2.1 问题描述

MDMCGVRPTW问题如图1所示,可描述如下:多个配送中心为前置仓进行配送服务,每个配送中心具有一定数量的多舱车,前置仓点的地理位置、各类型货物需求和访问时间窗已知;配送中心位置和配送车辆的各舱室容量已知;以总成本最小化为目标,为车辆规划行驶路径方案.为明确本文适用问题范围,做出以下假设:

1) 每个前置仓有且仅有一辆车访问.

2) 配送车辆匀速行驶,从配送中心出发完成任务后返回原车场.

- 3) 配送车辆为同车型, 车辆各舱室不允许超载.
- 4) 每个前置仓点有多种需求, 不同种类的货物需放在不同隔室进行配送.
- 5) 前置仓点的各类货物需求量均不大于对应车舱载重.

- 6) 配送车辆在行驶过程中产生油耗和碳排放, 多舱车配送的冷藏货物会产生制冷和货损成本.
- 7) 车辆不允许晚于右时间窗到达, 若车辆提前到达, 不能提前服务, 必须等待至左时间窗才能开始服务.

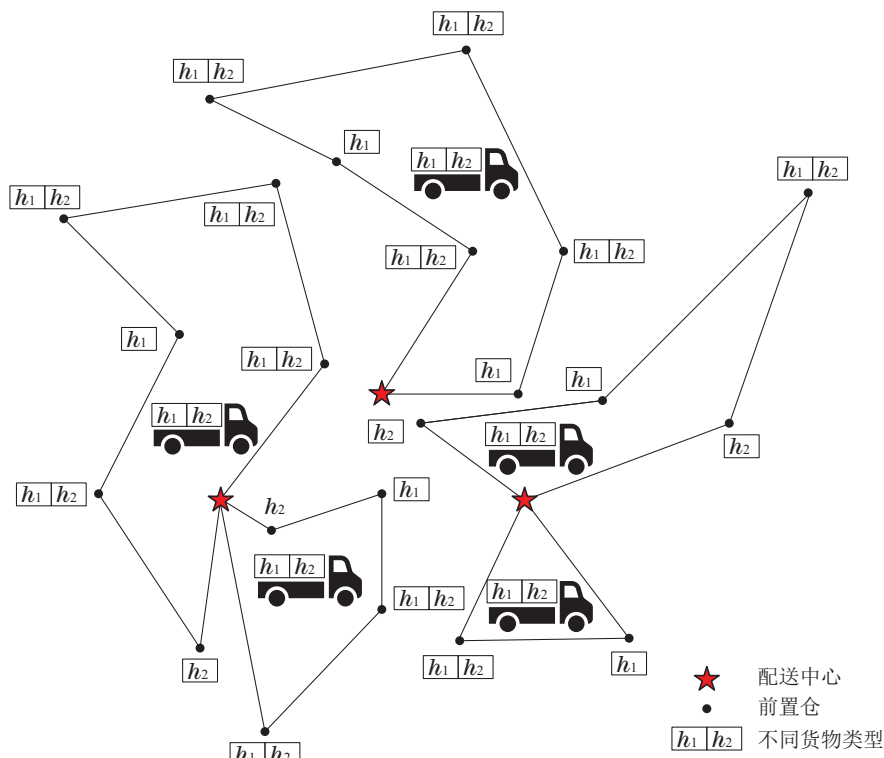


图 1 MDMCGVRPTW问题示例

Fig. 1 MDMCGVRPTW's example

2.2 符号和决策变量

- 1) 符号.
 - C : 前置仓点集合, $C = \{1, 2, \dots, n\}$.
 - D : 配送中心集合, $D = \{n + 1, \dots, n + m\}$.
 - V : 节点集合, $V = \{1, 2, \dots, m + n\}$, $V = C \cup D$.
 - A : 节点之间路径集合, $A = \{(i, j) | i, j \in V, i \neq j\}$.
 - K_d : 配送中心 d 的车辆集合, $d \in D, k_d \in K_d$.
 - H : 需求 h 类型集合, $H = \{1, 2\}$, 分别表示常温货物和冷藏货物, $h \in H$.
 - Q_h : 配送车辆的 h 类产品隔室容量.
 - q_{ih} : 前置仓点 i 对 h 类货物的需求.
 - d_{ij} : 节点 i 与节点 j 点间的距离.
 - $[ET_i, LT_i]$: 前置仓点 i 的时间窗, 其中: ET 表示 Early time window; LT 表示 Late time window.
 - v : 车辆行驶速度.
 - t_i : 前置仓点 i 的服务(装卸)时间.
 - M : 表示一个足够大的正整数.
- 2) 决策变量.

- $w_i^{k_d}$: 配送车辆 k_d 到达节点 i 处的时间.
- W_i : 节点 i 处的等待时间.
- $l_{ijh}^{k_d}$: 配送车辆 k_d 由节点 i 到节点 j 的 h 类货物载重.
- $y_i^{k_d}$: 节点 i 是否被配送车辆 k_d 服务, 是为 1, 否为 0.
- $x_{ij}^{k_d}$: 配送车辆 k_d 是否从节点 i 直接驶向节点 j , 是为 1, 否为 0.

2.3 模型优化目标

MDMCGVRPTW 优化目标为总运输成本 C , 由车辆固定成本 C_1 、行驶成本 C_2 、碳排放成本 C_3 、货损成本 C_4 、制冷成本 C_5 和等待成本 C_6 构成.

- 1) 车辆固定成本 C_1 .

车辆配送都会产生损耗等固定成本, 设 f 为单位车辆使用的固定费用(元/辆), 车辆使用的总固定成本为

$$C_1 = f \sum_{d \in D} \sum_{k_d \in K_d} y_d^{k_d}. \quad (1)$$

- 2) 车辆行驶成本 C_2 .

车辆行驶成本包括配送过程中的人员配备、燃料消耗和车辆维护成本, 其根据行驶路径长度产生成本,

设 c 为单位行驶成本(元/公里), 车辆行驶成本为

$$C_2 = c \sum_{(i,j) \in A} \sum_{d \in D} \sum_{k_d \in K_d} x_{ij}^{k_d} d_{ij}. \quad (2)$$

3) 碳排放成本 C_3 .

碳排放成本与运输途中燃油消耗密切相关. 相关研究表明, 配送车辆的燃油消耗与车辆载重和距离可近似表达为一个线性关系式^[17], 式(3)表示多舱车 k_d 途经边 (i, j) 的燃油消耗, F_0, F_m 分别表示空载和满载的油耗量(升/百公里).

$$F_{ij}^{k_d} = F_0 d_{ij} + \sum_{h \in H} \frac{F_m - F_0}{Q_h} l_{ijh}^{k_d} d_{ij}, \quad (3)$$

将油耗转化为碳排放成本, 设油耗碳排转换系数 e (千克/升)和单位碳排放成本为 P_c (元/千克), 总碳排放成本为

$$C_3 = e P_c \sum_{(i,j) \in A} \sum_{d \in D} \sum_{k_d \in K_d} F_{ij}^{k_d} x_{ij}^{k_d}, \quad (4)$$

4) 货损成本 C_4 .

冷藏货物的货损成本与时间呈指数关系^[7], 引入货损系数 $1 - e^{-at}$, 其中: t 表示时间; a 表示货损时间系数; 总货损成本由运输产生的货损成本 C_{41} 和装卸产生的货损成本 C_{42} 组成; p 表示货物的单位价格(元/吨); a_1, a_2 分别表示运输过程和装卸过程的货损时间系数.

$$C_{41} = \sum_{(i,j) \in A} \sum_{k_d \in K_d} \sum_{d \in D} p l_{ij2}^{k_d} (1 - e^{-a_1 \frac{d_{ij}}{v}}), \quad (5)$$

$$C_{42} = \sum_{i \in V} \sum_{j \in C} \sum_{k_d \in K_d} \sum_{d \in D} p (l_{ij2}^{k_d} - q_{j2}) (1 - e^{-a_2 t_j}), \quad (6)$$

总货损成本为

$$C_4 = C_{41} + C_{42}. \quad (7)$$

5) 制冷成本 C_5 .

制冷成本指冷藏舱制冷设备运行带来的成本, 由运输过程和装卸过程两部分的制冷成本构成^[7], 设 b_1, b_2 为运输过程和装卸过程产生的单位时间制冷成本(元/小时), 制冷成本表示为

$$C_5 = \sum_{(i,j) \in A} \sum_{k_d \in K_d} \sum_{d \in D} b_1 x_{ij}^{k_d} \frac{d_{ij}}{v} + \sum_{i \in C} b_2 t_i. \quad (8)$$

6) 等待成本 C_6 .

等待成本指车辆等待过程中产生的惩罚成本, 设单位等待惩罚成本为 η (元/小时)^[19], 等待成本表示为

$$C_6 = \eta \sum_{i \in C} W_i. \quad (9)$$

2.4 数学模型

本文构建的MDMCVRPTW数学模型如下:

$$\min C = C_1 + C_2 + C_3 + C_4 + C_5 + C_6, \quad (10)$$

$$\text{s.t.} \quad \sum_{d \in D} \sum_{k_d \in K_d} y_i^{k_d} = 1, \forall i \in C, \quad (11)$$

$$\sum_{i \in V} x_{ir}^{k_d} = \sum_{j \in V} x_{rj}^{k_d}, \forall r \in C, d \in D, k_d \in K_d, \quad (12)$$

$$\sum_{j \in C} x_{ij}^{k_d} = \sum_{j \in C} x_{ji}^{k_d} \leq 0, \forall i \in D, d \in D, k_d \in K_d, \quad (13)$$

$$\sum_{d \in D} \sum_{k_d \in K_d} \sum_{i \in V \setminus j} x_{ij}^{k_d} = \sum_{d \in D} \sum_{k_d \in K_d} y_j^{k_d}, \forall j \in C, \quad (14)$$

$$\sum_{d \in D} \sum_{k_d \in K_d} \sum_{j \in V \setminus i} x_{ij}^{k_d} = \sum_{d \in D} \sum_{k_d \in K_d} y_i^{k_d}, i \in C, \quad (15)$$

$$\sum_{i \in V} q_{ih} y_i^{k_d} \leq Q_h, \forall h \in H, d \in D, k_d \in K_d, \quad (16)$$

$$\sum_{j \in C} l_{jh}^{k_d} = \sum_{j \in C} y_j^{k_d} q_{jh}, \forall h \in H, d \in D, k_d \in K_d, \quad (17)$$

$$\sum_{k_d \in K_d} \sum_{d \in D} \sum_{j \in V \setminus i} l_{jih}^{k_d} - \sum_{k_d \in K_d} \sum_{d \in D} \sum_{j \in V \setminus i} l_{ijh}^{k_d} = q_{ih}, \quad (18)$$

$$w_i^{k_d} + W_i + \frac{d_{ij}}{v} + t_i - (M(1 - x_{ij}^{k_d})) \leq w_j^{k_d}, \quad (19)$$

$$ET_i \leq w_i^{k_d} + W_i \leq LT_i, \forall i \in C, d \in D, k_d \in K_d, \quad (20)$$

$$w_i^{k_d} \in \mathbb{R}_+, \forall i \in V, k_d \in K_d, \quad (21)$$

$$W_i \in \mathbb{R}_+, \forall i \in C, \quad (22)$$

$$l_{ijh}^{k_d} \in \mathbb{R}_+, \forall (i, j) \in A, h \in H, d \in D, k_d \in K_d, \quad (23)$$

$$y_i^{k_d} = \{0, 1\}, \forall i \in V, d \in D, k_d \in K_d, \quad (24)$$

$$x_{ij}^{k_d} = \{0, 1\}, \forall (i, j) \in A, d \in D, k_d \in K_d. \quad (25)$$

其中: 式(11)为每个前置仓只能被一辆车服务; 式(12)为前置仓节点车辆进出平衡约束; 式(13)为车辆从配送中心出发完成任务后返回原中心; 式(14)–(15)表示前置仓节点被车辆服务时一定有边与其相连; 式(16)限制路径上所有前置仓对各类货物的总需求量均小于各舱容量; 式(17)表示车辆从配送中心出发时各类货物的载重; 式(18)表示车辆在边上的载重和前置仓节点货物需求的递推关系; 式(19)表示车辆到达前置仓节点的时间与等待时间、服务时间和行驶时间的关系; 式(20)限制车辆在前置仓节点开始服务的时间在时间窗范围内; 式(21)–(25)定义决策变量的取值约束.

2.5 线性化

C_3 中 $x_{ij}^{k_d} l_{ijh}^{k_d}$ 两个决策变量相乘为非线性项, 为方便线性规划求解器求解, 使用大 M 法将非线性项线性化, 设 $l_{ijh}^{k_d} \leq M$, 引入新变量 $z_{ijh}^{k_d}$, 令 $z_{ijh}^{k_d} = l_{ijh}^{k_d} x_{ij}^{k_d}$, 分别取 $x_{ij}^{k_d}$ 为0和1时, 可得以下分段函数:

$$z_{ijh}^{k_d} = \begin{cases} 0, & x_{ij}^{k_d} = 0, \\ l_{ijh}^{k_d}, & x_{ij}^{k_d} = 1. \end{cases} \quad (26)$$

分段函数仍为非线性, 通过大 M 法将分段线性函数使用约束进行表达, 当 $x_{ij}^{k_d} = \{0, 1\}$ 时, 约束(27)–(28)等价于函数式(26), 即

$$z_{ijh}^{k_d} \leq Mx_{ij}^{k_d}, \quad (27)$$

$$l_{ijh}^{k_d} - z_{ijh}^{k_d} \leq M(1 - x_{ij}^{k_d}), \quad (28)$$

其中 $z_{ijh}^{k_d}$ 取值范围满足约束式(29)–(30).

$$z_{ijh}^{k_d} \leq l_{ijh}^{k_d}, \quad (i, j) \in A, h \in H, d \in D, k_d \in K_d, \quad (29)$$

$$z_{ijh}^{k_d} \geq 0, \quad (i, j) \in A, h \in H, d \in D, k_d \in K_d. \quad (30)$$

综上所述, 通过将 C_3 转化为式 (31), 并加入约束式(27)–(30), 实现所构建的非线性模型的线性化, 即

$$C_3 = eP_c F_0 \sum_{(i,j) \in A} \sum_{d \in D} \sum_{k_d \in K_d} d_{ij} x_{ij}^{k_d} + eP_c \sum_{(i,j) \in A} \sum_{d \in D} \sum_{h \in H} \sum_{k_d \in K_d} \frac{F_m - F_0}{Q_h} d_{ij} z_{ijh}^{k_d}. \quad (31)$$

3 改进变邻域搜索算法

3.1 算法框架

多车场带时间窗的车辆路径问题作为车辆路径问题的变体属于NP-难问题, 而多舱车的引入导致模型约束增加, 问题的解空间更加复杂. 文中所构建的MILP模型只能利用商业求解器实现小规模算例求解, 因此, 设计一个高效的启发式求解算法十分必要. 鉴于VNS算法在求解多车场带时间窗车辆路径问题中的优异表现^[4], 本文设计IVNS算法对该问题进行求解, 算法框架如图2所示.

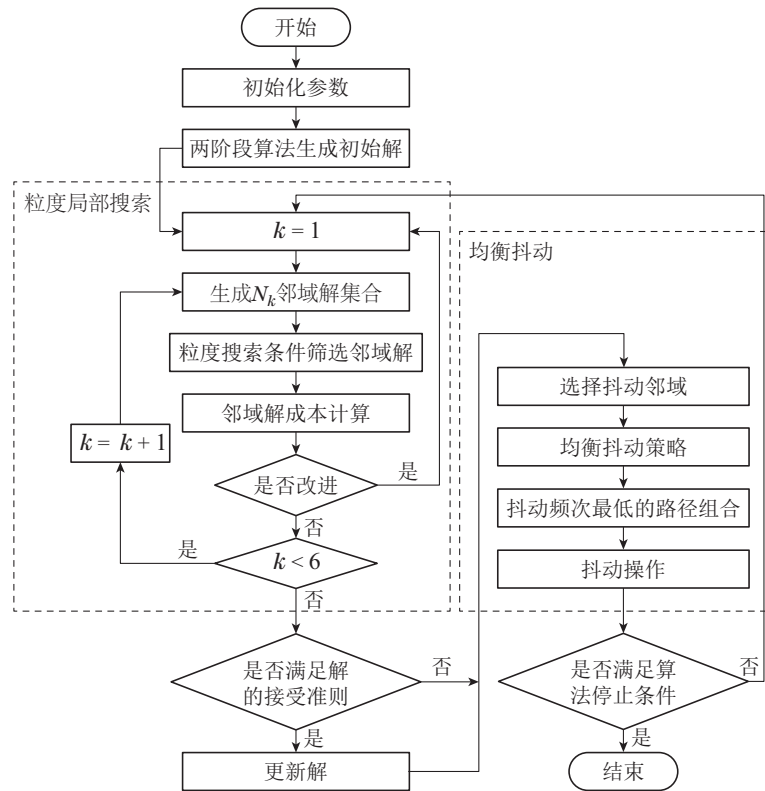


图 2 IVNS流程图

Fig. 2 IVNS's flow chart

IVNS算法首先基于2SH算法得到一个高质量初始解, 随后, 基于VNS算法的基本框架对抖动和局部搜索进行具体的设计和改进. 算法共涉及10个邻域结构, 分别对应单路径、双路径及三路径的操作算子. 在局部搜索中, 采用变邻域下降机制(variable neighborhood descent, VND)^[20]对6个邻域结构进行嵌套粒度搜索. 在抖动中, 采用循环邻域变换策略驱动6个算子

基于所提出的均衡机制对解空间进行探索.

3.2 初始解构造

本文针对多车场特征提出两阶段初始解构造算法. 相关研究表明, 在单车场问题的初始解构造中, 节约 (clarke-wright, CW) 算法相较于插入算法和扫描算法在求解速度和质量方面表现更佳^[21]. 基于此, 在初始解生成的第1阶段, 将前置仓分配至最近配送中心, 并

使用CW算法在各配送中心内构造路径. 为进一步提升解的质量, 优化解的可行性, 在初始解生成的第2阶段, 在违反车场容量约束的配送中心内, 删除超出车辆数限制且包含前置仓点最少的路径, 使用3步后悔值插入(regret-3 insert, RI)算法^[22]重插入被移除的前置仓, 3步后悔值计算公式如下:

$$\max_{i \in S} \left\{ \sum_{j=1}^3 (\Delta f_{i,x_{ij}} - \Delta f_{i,x_{i1}}) \right\}, \quad (32)$$

其中: x_{ij} 表示节点 i 的第 j 小插入评价函数变化值所对应的路径; $\Delta f_{i,x_{ij}}$ 表示节点 i 插入到路径 x_{ij} 的评价函数变化值, 需要注意节点插入路径指插入路径的最优位置, 下面是初始解构造算法具体操作步骤:

步骤 1 初始化, 定义空集 S .

步骤 2 根据欧氏距离将各前置仓分配给最近的配送中心.

步骤 3 基于每一个配送中心分配到的前置仓节点使用CW算法进行路径初始化, 得到 x_{ini} .

步骤 4 对违反车场容量约束的配送中心内的路径按节点数量排序, 删除节点数量最少的路径, 将被移除的节点放入 S , 直至配送中心内使用车辆数不超过限制, 得到初始解 x'_{ini} .

步骤 5 若 S 为空集则算法终止, 并输出 x'_{ini} 作为初始解 x_{ini} , 否则, 执行步骤 6.

步骤 6 通过式(32), 计算 S 中各前置仓节点插入至 x'_{ini} 的3步后悔值, 选择该值最大的点 p , 插入到使 x'_{ini} 评价函数值变化最小的位置, 更新 x'_{ini} , 同时, 更新 $S = S/p$, 返回步骤 5.

3.3 解的评价

因为在求解过程中允许不可行解的出现, 所以需要在解的评价函数中加入惩罚项, 以使解朝着可行的方向不断优化. 本文基于Yeniay^[23]所提出的评价方法, 设计本文的评价函数为

$$F(x) = C(x) + \alpha q_1(x) + \beta q_2(x) + \gamma w(x), \quad (33)$$

其中: $C(x)$ 为基于式(10)计算得到的总成本; $q_1(x)$ 和 $q_2(x)$ 分别为常温舱和冷藏舱的超载量; $w(x)$ 为总时间窗违反量; α, β, γ 为相应惩罚因子.

3.4 前向时间松弛

启发式算法在构造路径时, 通常假设车辆离开配送中心的时间为配送中心的左时间窗, 但该假设会导致车辆在部分前置仓产生不必要的等待时间, 从而增加由等待产生的综合成本. 文献[24]通过优化车辆从枢纽点出发的时间, 在不增加时间窗违反的情况下, 减少路径上车辆的不必要等待时间, 即前向时间松弛. 每条路线的出发时间可调整为

$$ET_d + \min_{i \in \text{route}} \{ [LT_i - (w_i^{k_d} + W_i)]^+ + \sum_{j \leq i} W_j \}, \quad (34)$$

其中: ET_d 指配送中心开始营业时间; LT_i 表示线路上前置仓节点的右时间窗; $w_i^{k_d}$ 表示车辆 k_d 到达前置仓节点 i 的时间; W_i 表示在前置仓节点 i 的等待时间.

3.5 邻域结构

邻域结构的设计是变邻域搜索算法的重要一环, 邻域结构由邻域算子确定, 本文所设计的IVNS算法涉及10个邻域结构, 对应如下10个算子:

N_1 2-opt: 在一条路径中选择两个前置仓节点, 并将其中间的前置仓节点序列翻转.

N_2 Intra swap: 在一条路径中选择两个前置仓节点并交换它们的位置.

N_3 Shift: 在一条路径中选择一个前置仓节点, 将它重新插入同一路径的不同位置.

N_4 Cross over: 在两条路径中分别移除一条边将路径分为两个部分, 连接第1条路径首部和第2条路径尾部、第2条路径首部和第1条路径尾部.

N_5 Inter swap: 交换当前路径中的某前置仓节点与不同路径上另一前置仓节点的位置.

N_6 1-insert: 将某前置仓节点从当前路径中移除, 并插入不同路径上的另一位置.

N_7 2-insert: 将两个连续前置仓节点从当前路径中移除, 并一同插入不同路径上的另一位置.

N_8 Cross exchange (2-1): 交换当前路径中两个连续前置仓节点与不同路径上另一前置仓节点的位置.

N_9 Cross exchange (2-2): 交换两条路径中2个连续前置仓节点的位置.

N_{10} Cross exchange (1-1-1): 选定3条路径中3个前置仓节点, 将第1条路径上的前置仓节点转移至第2条路径, 第2条路径上的前置仓节点转移至第3条路径, 第3条路径上的前置仓节点转移至第1条路径.

3.6 粒度局部搜索

局部搜索通过在当前解 x 的给定邻域结构内进行搜索, 找到最佳改进解 x' . 本文的局部搜索设计有6个邻域结构, 对应的邻域算子集合为 $\{N_1, N_2, N_3, N_4, N_5, N_6\}$, 且采用变邻域下降(VND)策略决定算子的搜索顺序. 局部搜索是IVNS算法中耗时最长的环节. 本文采用粒度搜索机制^[4]提升搜索效率. 粒度搜索机制中规定一个操作被执行的条件是, 当且仅当该操作所生成的解中至少包含以下弧段之一:

- 1) 与配送中心相连的弧段;
- 2) 包含于当前最优解的弧段;
- 3) 评价函数值小于粒度阈值 ϕ 的“短”弧, 粒度阈

值 ϕ 由式(35)求得,即

$$\phi = \sigma \frac{f(x_{ini})}{n + N(x_{ini})}, \quad (35)$$

其中: σ 是一个正系数; n 是前置仓节点数量; $f(x_{ini})$ 和 $N(x_{ini})$ 分别是初始解的评价函数值和路径数量.粒度搜索机制帮助算法集中于更有可能产生优质解的子邻域结构,从而,最大程度降低劣解对搜索效率的影响,粒度局部搜索的具体流程如表1所示.

表1 基于粒度搜索条件的局部搜索算法

Table 1 Granularity mechanism based local search algorithms

输入: x
输出: x'
1: $k \leftarrow 1, x' \leftarrow x$
2: while $k \neq 7$ do
3: $S \leftarrow \Phi$
4: for x'' in $N_k(x')$ do
5: if x'' 满足粒度搜索条件 then
6: $S \leftarrow S \cup \{x''\}$
7: end if
8: end for
9: $s \leftarrow \operatorname{argmin}(f(x'')), x'' \in S$
10: if $f(s) < f(x')$ then
11: $x' \leftarrow s, k \leftarrow 1$
12: else
13: $k \leftarrow k + 1$
14: end if
15: end while

3.7 基于均衡机制的抖动策略

为跳出局部最优,从当前解 x 的给定邻域结构内选择一个解作为下次局部搜索的起点.为尽量探索解空间,本文扰动设计有6个邻域结构,对应的邻域算子集合为 $\{N_5, N_6, N_7, N_8, N_9, N_{10}\}$,并提出基于均衡机制的扰动策略.设所有路径构成 R ,为 N_5, N_6, \dots, N_9 分别构造 $C_{|R|}^2$ 行3列的二维频次表 T_1, T_2, \dots, T_5 ,前两列为路径编号,最后一列为频次 r ,为 N_{10} 构造一个 $C_{|R|}^2$ 行4列的二维频次表 T_6 ,前3列为路径编号,最后一列为频次 r .频次表中的每一行都代表着邻域算子可能操作的一组路径.若一次扰动中选择基于邻域算子操作某组路径而得到的邻域解,则将邻域算子对应频率表中的相应组合的频次加1.扰动解由邻域算子操作当前解中频次最低的路径组合产生.具体流程如表2所示.

均衡抖动策略是一种无放回的抽样,根据问题特征设 N 为所有路径组合总和, n 为抖动阶段已抽取的路径组合总和,设样本总体方差为 σ^2 ,有返回随机抽样对应的方差为 $\sigma_1^2 = \frac{\sigma^2}{n}$,根据有限总体系数

$\frac{N-n}{N-1}$,均衡抖动策略的抽样误差为

$$\sigma_2^2 = \frac{\sigma_1^2}{n} \times \frac{N-n}{N-1}, \quad (36)$$

根据式(36)可以看出,随着均衡策略的执行,有限抽样系数不断减小,即意味着样本方差不断减小.

表2 基于频次的均衡抖动算法

Table 2 Frequency-based balanced shaking algorithm

输入: $x, k, T_1, T_2, \dots, T_6$
输出: x'
1: $S \leftarrow \Phi, m \leftarrow T_k(1, \text{end})$
2: for i in $\text{size}(T_k, 1)$ do
3: if $T_{k_{\text{shake}}}(i, \text{end}) < m$ then
4: $S \leftarrow \{i\}, m \leftarrow T_k(i, \text{end})$
5: else if $T_k(i, \text{end}) = m$ then
6: $S \leftarrow S \cup \{i\}$
7: end if
8: end for
9: $\text{len} \leftarrow \text{length}(S)$
10: $j \leftarrow S(\text{randi}(\text{len}))$
11: $T_k(j, \text{end}) \leftarrow T_k(j, \text{end}) + 1$
12: $x' \leftarrow N_k(x, T_k(j, 1 : \text{end} - 1))$

3.8 解的接受准则

贪婪接受策略容易导致算法陷入局部最优,因此,本文采用Dueck^[25]提出的(record-to-record travel, R-RT)准则,基于设定的评价函数阈值来决策是否接受新解.设 x^* 是当前最优解, x 是当前解, x' 是局部搜索得到的最佳改进解.如果 $f(x') > f(x^*)$,则更新 x 和 x^* 为 x' ,如果 $(f(x') - f(x^*)) / f(x') \leq \varepsilon$,则接受 x' 作为当前解 x , ε 是设定的正参数.

3.9 时间复杂性分析

改进变邻域搜索算法的时间复杂度分析如下:设算例客户点规模为 n ,车辆数为 k ,算法迭代次数为 N ,变邻域下降迭代次数为 G .依据算法描述,计算时间复杂度如下:

1) 初始解构造算法包含两个阶段:

$$T_1(n) = n^2 \log n + n^3.$$

2) 局部搜索阶段包含6个邻域算子操作,即

$$T_2(n) = 6 \times N \times G \times n^2.$$

3) 抖动阶段包含抖动路径选择一个邻域算子的操作,即

$$T_3(n) = N \times k^3 \times 1.$$

4) 解的接受,即

$$T_4(n) = N \times 1.$$

综上,算法的总计算时间复杂度可估算为

$$T(n) = O(n^2 \log n + n^3 + 6 \times N \times G \times n^2 + N \times k^3 + N) = O(n^3 + N \times G \times n^2 + N \times k^3).$$

4 实验设计与分析

在本节中,首先,选取MDVRPTW标准算例验证算法有效性.其次,使用MDMCGVRPTW仿真数据进一步验证模型的正确性和算法的有效性.

所有算法均通过MATLAB R2020a编程实现,操作系统为Windows10,电脑内存为8GB,CPU为Intel i5-9500 3GHz.根据文献[4,26]和多次试验,粒度搜索机制中的正系数 σ 设置为2.5.接受准则中的阈值系数 ε 设置为0.1.最大迭代次数MaxIter设置为5000.最大不改进迭代次数的设置与算例规模相关.当算例规模 $n \leq 50$ 时,Max NonImp设为200; $50 < n < 150$ 时,设为150;当 $n \geq 150$ 时,设为100.本章节实验均基于10次运行的求解结果.

4.1 MDVRPTW算例实验

MDVRPTW算例的目标函数为最小化距离,此外,MDVRPTW还需考虑一个行驶时间约束,因此,该算例实验设置罚函数项为时间窗违反量、载重违反量和行驶时间违反量,罚函数参数的值均设置为100.算例来源: http://www.bernabe.dorronsoro.es/vrp/index.html?Problem_Instances/MDVRPTWInstances.html.

4.1.1 2SH算法有效性实验

为验证本文提出的2SH算法有效性,对比贪婪插入(greedy insert, GI)算法^[16]、RI算法^[19]在MDVRPTW算例下的初始解评价函数值 f_{ini} 以及运行时间 T/s .

由表3可以看出,对比GI和RI初始解生成方法,2SH在初始解质量有明显改进,初始解构造速度上也优于另外两种算法,且对于Pr10之后的时间窗密集型算例,该方法能大幅减少约束违反程度.2SH算法通过融合CW和RI算法,不仅进一步提升了初始解的求解质量,而且弥补了插入算法在运行时间上的不足.

4.1.2 IVNS算法有效性实验

本文对比了复合邻域-萤火虫算法(discrete firefly algorithm with compound neighborhoods, DFCAN)^[26]、禁忌-变邻域搜索算法(efficient variable neighborhood search with tabu shaking, EVTNS)^[4]与提出的IVNS在MDVRPTW标准算例的求解结果.如表4所示, n 为客户点数量, d 为中心数,BKS(best known solution)为目前文献已知的最优解,Best, Avg分别为算法运行10次的据所求得的最优解和平均解,“-”表示该算法未给出相关数据,Gap由式(37)得到.本文选取EVTNS的CPU作为标准,IVNS运行时间 t 由式(38)进

行换算,即

$$\text{Gap}\% = \frac{\text{Best} - \text{BKS}}{\text{BKS}} \times 100\%, \quad (37)$$

$$t = t \times \frac{\text{Mark}_{\text{IVNS}}}{\text{Mark}_{\text{EVTNS}}}. \quad (38)$$

表3 不同初始解生成算法结果比较

Table 3 Comparison of initial solutions obtained by different algorithms

算例	GI		RI		2SH*	
	f_{ini}	T/s	f_{ini}	T/s	f_{ini}	T/s
pr01	1435.01	1.33	1221.96	1.77	1158.81	0.30
pr02	2384.22	7.28	2147.31	8.63	2113.69	1.86
pr03	3642.02	23.00	2639.94	29.65	2618.51	4.29
pr04	4156.90	66.16	4304.46	76.08	3283.41	20.98
pr05	4283.32	101.34	4271.21	106.66	3925.02	35.30
pr06	5504.61	186.14	7512.74	275.22	4398.28	47.33
pr07	1814.83	4.41	1551.43	6.45	1516.03	0.31
pr08	3040.98	23.28	2420.32	32.47	2412.30	4.94
pr09	4172.28	81.07	3548.50	115.47	3077.50	15.49
pr10	5317.71	189.14	4116.20	246.38	4028.77	47.95
pr11	31245.20	0.85	17040.07	1.19	4369.93	0.32
pr12	12830.90	5.84	10033.95	8.68	2250.44	2.26
pr13	28946.83	19.37	5520.40	27.44	2350.38	3.61
pr14	4004.80	45.95	4582.16	55.63	2965.42	18.23
pr15	3597.62	90.11	3141.30	104.61	2840.32	27.35
pr16	4775.73	174.14	3445.88	247.82	3356.66	24.61
pr17	31644.44	3.09	4021.01	4.10	3851.83	1.09
pr18	2660.33	22.02	5242.06	25.98	2301.64	6.10
pr19	3934.91	84.93	2685.27	120.66	2561.85	11.51
pr20	16064.93	188.01	3626.45	201.55	3438.96	34.24

由表4可知,在求解质量方面,本文IVNS算法求解结果与BKS的最小偏差为0.00%,最大偏差为2.26%,平均偏差为0.89%.在求解时间方面IVNS算法在大规模算例上的求解质量和求解速度要弱于EVTNS,但在得到的已知最优解的中小规模算例上IVNS的求解速度要快于EVTNS,通过以上对比分析验证了算法的有效性.

图3给出Pr03的最优解变化趋势图,可以看出算法在300次迭代左右完成收敛,并在450次迭代左右达到最大不改进迭代次数.观察收敛曲线可以发现,算法在迭代中后期陷入局部最优解时,可以快速跳出局部最优,具有良好的寻优能力.表5给出Pr03的最佳路径方案.

4.2 MDMCGVRPTW仿真实验

仿真实验基于上海市叮咚买菜的实际数据,该数据共涉及5个配送中心和191个前置仓配送点.仿真数据中包括地理位置、时间窗和需求量信息.详细仿真数据可通过<https://gitee.com/caojian98/CCTA220182/blob/master/data.txt>获取.

表 4 MDVRPTW算例实验结果对比

Table 4 Comparison of experimental results of MDVRPTW benchmark

算例	n	d	BKS	DFCAN				EVTNS				IVNS			
				Best	Avg	T/s	Gap/%	Best	Avg	T/s	Gap/%	Best	Avg	T/s	Gap/%
Pr01	48	4	1074.12	1074.12	1074.12	-	0.00	1074.12	1074.12	50.18	0.00	1074.12	1074.12	44.21	0.00
Pr02	96	4	1762.21	1762.21	1771.55	-	0.00	1762.21	1763.54	249.97	0.00	1762.21	1770.23	153.12	0.00
Pr03	144	4	2373.65	2373.65	2380.77	-	0.00	2373.65	2375.49	385.47	0.00	2373.65	2374.37	581.57	0.00
Pr04	192	4	2814.34	2822.09	2823.78	-	0.28	2814.34	2819.16	2558.71	0.00	2866.46	2883.71	1035.78	1.85
Pr05	240	4	2964.06	2964.06	2973.84	-	0.00	2965.18	2979.87	678.41	0.04	3012.79	3079.21	1289.12	1.64
Pr06	288	4	3588.78	3588.78	3623.95	-	0.00	3590.58	3601.05	2982.68	0.05	3645.26	3739.22	2892.91	1.57
Pr07	72	6	1418.22	1418.22	1420.49	-	0.00	1418.22	1418.28	108.58	0.00	1418.22	1430.68	93.82	0.00
Pr08	144	6	2096.73	2096.73	2105.96	-	0.00	2096.73	2101.29	371.27	0.00	2101.55	2144.12	284.55	0.23
Pr09	216	6	2712.69	2712.56	2719.88	-	0.00	2717.69	2722.96	1140.11	0.18	2750.42	2839.88	732.37	1.39
Pr10	288	6	3469.29	3469.29	3481.09	-	0.00	3469.29	3485.29	3450.75	0.00	3547.70	3675.08	2083.44	2.26
Pr11	48	4	1005.73	1005.73	1011.56	-	0.00	1005.73	1005.73	10.24	0.00	1005.73	1013.47	39.46	0.00
Pr12	96	4	1464.5	1464.50	1467.43	-	0.00	1464.50	1470.89	355.61	0.00	1464.50	1502.65	286.54	0.00
Pr13	144	4	1994.14	2001.81	2007.21	-	0.38	2001.81	2006.59	232.65	0.38	2010.96	2072.21	613.14	0.84
Pr14	192	4	2195.33	2196.29	2199.15	-	0.04	2195.33	2204.29	1425.31	0.00	2236.47	2293.47	914.41	1.87
Pr15	240	4	2434.94	2436.28	2448.46	-	0.06	2434.94	2445.13	1173.17	0.00	2488.91	2564.87	1320.63	2.22
Pr16	288	4	2836.67	2842.30	2848.55	-	0.20	2850.69	2855.69	1867.41	0.49	2872.44	2967.06	3358.71	1.26
Pr17	72	6	1236.24	1236.24	1237.48	-	0.00	1236.24	1239.81	58.87	0.00	1236.24	1241.35	73.89	0.00
Pr18	144	6	1788.18	1788.18	1794.80	-	0.00	1788.18	1794.29	280.05	0.00	1788.18	1839.17	341.45	0.00
Pr19	216	6	2261.08	2261.08	2267.18	-	0.00	2263.74	2270.59	966.06	0.12	2289.49	2385.64	630.83	1.26
Pr20	288	6	2984.01	2987.24	3006.36	-	0.11	2995.08	3011.52	2111.64	0.37	3024.83	3173.96	2736.26	1.37
Avg	-	-	-	2225.07	2233.18	-	0.05	2225.91	2232.28	1022.86	0.08	2248.51	2303.22	975.31	0.89

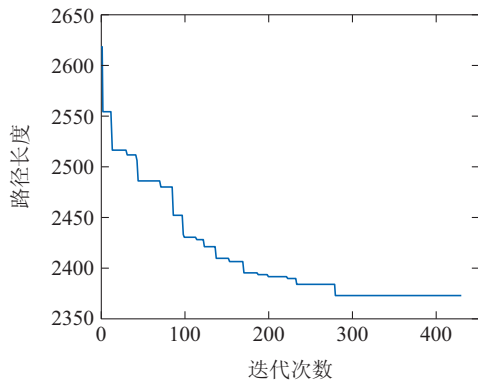


图 3 Pr03最短路径变化趋势图

Fig. 3 The iterative trend of shortest path of Pr03

基于以上实际数据,生成14组不同规模的测试算例. Ins 1~Ins 14算例和对应车辆数据可通过<https://github.com/caojian98/CCTA220182/tree/master>获得. Ins 1~Ins 7设计为小规模算例验证模型正确性, Ins 8~Ins 14设计为大规模算例比较算法有效性.

4.2.1 仿真实验模型参数设置

本文设置MGMCVRPTW模型参数如下: $f = 500$ 元/辆, $c = 5$ 元/千米, $F_0 = 16.5$ 升/百公里, $F_m = 20$ 升/百公里, $e = 2.66$ 千克/升, $P_c = 2$ 元/千克, $a_1 = 0.02$, $a_2 = 0.04$, $p = 5000$ 元/吨, $b_1 = 15$ 元/小时, $b_2 = 20$ 元/小时, $\eta = 4$ 元/小时, $v = 36$ 千米/小时, $Q_1 =$

2.24 吨, $Q_2 = 2.24$ 吨. 目标函数的惩罚参数 α, β, γ 均设置为100.

表 5 本文算法求解的Pr03路径

Table 5 The algorithm solution path of Pr03

车辆行驶路径	总路程
D1-133-129-73-127-10-112-12-78-87-91-D1	
D1-11-118-42-38-20-31-103-27-13-1-139-D1	
D1-39-62-80-3-132-26-17-45-109-107-5-28-35-71-D1	
D1-95-114-110-119-69-40-121-88-52-14-D1	
D2-37-79-19-50-72-100-74-141-24-D2	
D2-117-8-125-34-101-105-70-135-56-4-6-D2	2373.65
D2-75-55-67-85-57-36-61-111-83-115-D2	
D2-51-25-126-76-116-54-120-98-49-D2	
D3-2-134-77-30-7-90-32-94-D3	
D3-102-48-97-64-137-142-113-44-144-99-130-D3	
D4-124-136-60-140-89-9-131-47-D4	
D4-86-21-66-29-65-33-93-59-82-53-D4	
D4-92-63-41-15-143-96-138-81-122-23-84-106-D4	
D4-43-104-123-128-68-46-108-58-22-18-16-D4	

4.2.2 小规模仿真算例实验

为验证提出模型的准确性和算法的有效性,对比IVNS算法与CPLEX在算例Ins 1~Ins 7上的求解结果.

表6给出CPLEX和IVNS算法在各仿真算例中所求得的最优解Opt、运行时间T/s以及IVNS算法所求得的最优解Opt相较于CPLEX的改进幅度Gap/%, Gap小于0表示IVNS算法求解结果较CPLEX有改进. 设置CPLEX运行时间上限为9600s, “*”表示解为CPLEX在规定时间内运行得到的一个可行解.

由表6可知, IVNS的求解效率显著优于CPLEX. 在Ins 1至Ins 3算例中IVNS与CPLEX求得相同当前最优解, 而在Ins 4~Ins 7中, IVNS改进了当前最优解, 且平均改进率达3.82%. 此外, IVNS相较于CPLEX的改进幅度Gap随着算例规模的增大呈现出增加的趋势.

4.2.3 大规模仿真算例实验

为进一步验证算法及改进策略在求解大规模算例的有效性和稳定性, 采用IVNS算法对算例Ins 8~Ins 14进行求解, 并与不考虑均衡扰动机制和粒度搜索策略但算子不变的VNS算法, 以及文献[16]设计的自适应大邻域搜索算法(adaptive large neighborhood search, ALNS)算法进行对比. 表7中给出VNS算法、ALNS算法和IVNS算法在各算例下求得的最优值Opt、平均值Avg, Avg相对于Opt的误差比率Gap/%和运行时间T/s.

如表7所示, 在求解质量方面, IVNS算法求得所有算例的当前最优解, ALNS算法仅在Ins 8上的求得与

IVNS算法相同的当前最优解, 而VNS算法在所有算例的求解质量均不如IVNS算法. 上述结果说明改进策略能够增强算法全局搜索能力, 避免算法陷入局部最优. 求解时间方面, IVNS算法在求解所有算例时所花费的时间明显少于ALNS算法. 在求解的稳定性方面, VNS算法平均Gap为4.31%, ALNS算法平均Gap为3.93%, 而IVNS算法平均Gap最小, 仅为1.69%, 说明IVNS算法的求解具有较强的稳定性. 图4给出由IVNS算法求得的Ins 14最优配送路线, 用不同颜色的路线区分不同的车辆.

表6 IVNS算法与CPLEX实验结果比较

Table 6 Comparison of experimental results between IVNS and CPLEX

算例	CPLEX		IVNS		Gap/%
	Opt	T/s	Opt	T/s	
Ins 1	3687.89	3.45	3687.89	2.91	0.00
Ins 2	3884.12	44.88	3884.12	5.84	0.00
Ins 3	4668.72	456.94	4668.72	18.75	0.00
Ins 4	5318.37*	9600	5296.61	31.90	-0.41
Ins 5	6440.56*	9600	5972.25	42.45	-7.27
Ins 6	7319.09*	9600	6661.62	58.31	-8.98
Ins 7	9094.67*	9600	8176.23	72.27	-10.10
Avg	5773.35	5557.90	5478.21	33.21	-3.82

表7 IVNS算法与VNS算法及ALNS算法实验结果比较

Table 7 Comparison of IVNS experimental results with VNS and ALNS

算例	VNS				ALNS				IVNS			
	Opt	Avg	Gap/%	T/s	Opt	Avg	Gap/%	T/s	Opt	Avg	Gap/%	T/s
Ins 8	10671.81	10990.13	2.98	53.41	9667.20	9850.43	1.90	83.41	9667.20	9729.91	0.65	44.32
Ins 9	14165.15	14636.43	3.33	90.94	13623.86	13978.51	2.60	94.88	13466.96	13602.77	1.00	83.45
Ins 10	17531.91	18186.31	3.73	194.77	17137.61	17682.54	3.18	215.27	16952.00	17133.85	1.07	175.04
Ins 11	22596.62	23591.50	4.40	466.54	21205.78	22038.62	3.93	809.24	21036.41	21272.66	1.12	466.54
Ins 12	24843.36	26038.97	4.81	1295.43	24783.15	25985.61	4.85	1638.75	24264.10	24639.72	1.55	1286.88
Ins 13	27603.77	29062.97	5.29	2209.99	28045.51	29550.05	5.36	2477.17	26590.82	27350.30	2.86	2139.63
Ins 14	30375.12	32080.06	5.61	3352.82	30560.53	32299.85	5.69	3960.19	29138.83	30171.65	3.54	3272.78
Avg	21112.53	22083.77	4.31	1094.84	20717.66	21626.52	3.93	1325.56	20159.47	20557.27	1.69	1066.95

5 敏感性分析

5.1 不同配送策略的敏感性分析

为验证多舱共配策略在前置仓配送中的合理性, 基于算例Ins 8~Ins 14, 分别在单舱配送策略SC(分别执行不同种类货物配送)和多舱共配策略MC下运用IVNS算法进行分析. 单舱数据改造如下: 将车辆均换为单舱车, 单舱车容量为4.48吨, 各配送中心常温单舱车和冷藏单舱车的车辆数均为 $\lceil |K_d|/2 \rceil$. 各算例不同策略下当前最优解Opt和碳排放量CE如表8所示,

表中给出MC策略下Opt和CE相对于SC策略的改进率Gap₁%和Gap₂%.

由表8可知, 在7组较大规模算例中, 多舱共配策略下的总成本和碳排放均低于单舱配送策略, 且平均改进率达20.25%和40.23%. 成本改进率和碳排放改进率随着规模算例的增大均呈上升趋势.

5.2 不同配送时效的敏感性分析

本文通过调整需求点时间窗的宽度来改变配送系统的时效, 探究不同配送时效对配送方案的影响. 在

原始算例Ins 8~Ins 14的基础上, 将各前置仓点的允许服务最晚时间提早20 min(TW1), 15 min(TW2)和推迟15 min(TW3), 生成3组新算例, 并将实验结果同原始算例(normal time window, NTW)进行对比. 表9中给出不同配送时效下, 最优配送策略所对应的当前最优解Opt、碳排放量CE/千克和平均值Avg, 以及各时效下Opt和CE相对于原始时效的Gap/%.

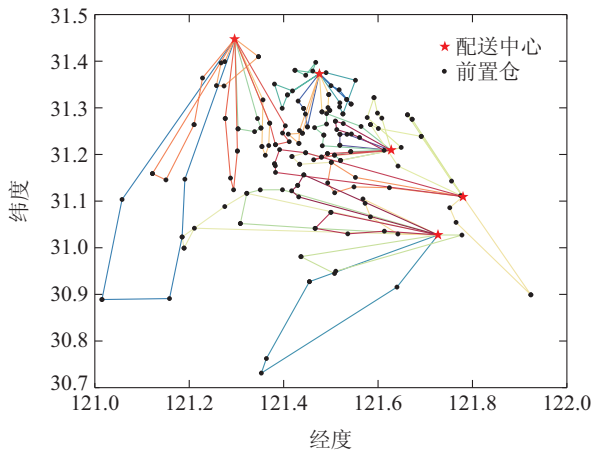


图 4 Ins 14最优配送路线

Fig. 4 The best routes of Ins 14

整体上看, 时间窗的放松可以带来成本和碳排放的下降, 而时间窗的加紧则会带来成本和碳排放的升高. 因此, 对于相关企业而言, 需要平衡服务时效性和配送经济性、环保性. 对比TW1和TW2, 两组算例的时间窗宽度仅相差5 min, 但Opt和CE的变化显著, 可以得出以下结论: 当配送系统的时效性达到一定水平时, 随着系统配送时效的不断提升, 导致的成本和排放的代价将更加显著.

表 8 不同配送策略实验结果对比

Table 8 Comparison of results between different distribution strategies

算例	SC		MC		Gap ₁ /%	Gap ₂ /%
	Opt	CE/千克	Opt	CE/千克		
Ins 8	11042.06	385.46	9667.20	245.98	-12.45	-36.19
Ins 9	15896.84	445.82	13466.96	280.10	-15.29	-37.17
Ins 10	20755.18	533.48	16952.00	330.71	-18.32	-38.01
Ins 11	26526.44	650.82	21036.41	395.68	-20.70	-39.20
Ins 12	31679.78	876.82	24264.10	510.26	-23.41	-41.81
Ins 13	35716.20	995.02	26590.82	596.18	-255.55	-40.08
Ins 14	39394.93	1143.38	29138.83	647.56	-26.03	-43.36
Avg	25858.70	718.68	20159.47	429.49	-20.25	-40.23

表 9 不同配送时效下实验结果对比

Table 9 Comparison of results between different distribution timeliness

算例	TW1		TW2		NTW		TW3	
	Opt	CE/千克	Opt	CE/千克	Opt	CE/千克	Opt	CE/千克
Ins 8	10625.18	265.58	9826.48	260.02	9667.20	245.98	9120.91	242.94
Ins 9	14242.77	306.13	13466.96	289.09	13426.69	280.10	13014.42	270.23
Ins 10	17787.33	364.17	17176.18	350.24	16952.00	330.71	16246.44	325.55
Ins 11	21495.54	437.76	21072.52	400.14	21036.41	395.68	20125.76	387.35
Ins 12	25041.85	557.79	24626.87	535.37	24264.10	510.26	22705.18	493.27
Ins 13	28071.54	643.16	27772.19	599.45	26590.82	596.18	26384.12	585.39
Ins 14	30849.89	706.39	30603.31	650.54	29138.83	647.56	28841.04	636.51
Avg	21159.16	468.71	20649.22	440.69	20159.47	429.49	19491.12	420.18
Gap/%	4.99	9.13	2.46	2.61	0.00	0.00	-3.31	-2.17

6 总结与展望

为解决社区团购前置仓配送难题, 本文针对多车场带时间窗的绿色多舱车车辆路径问题进行研究, 主要结论如下: 1) 本文研究的MCMDGVRPTW不仅考虑多车场协同, 而且考虑多货物种类协同、低碳环保的特点, 是对MDVRP, MCVRP和GVRP的进一步深化和拓展, 具有现实意义和理论意义; 2) 提出的两阶段混合初始解生成算法, 结合节约算法和后悔值插入算法的优点, 适用于约束多样的大规模算例, 能够在较短时间内得到违背约束程度较小的初始解; 3) 经过

多次测试和对比实验, 在设计的改进变邻域搜索算法中, 引入的粒度机制能有效提高局部搜索阶段的寻优效率, 提出的均衡抖动策略可以使算法在同等时间内更充分地探索解空间; 4) 敏感性分析结果表明, 多舱共配策略能够解决多品类货物的协同配送难题. 当配送系统的时效性达到一定水平时, 随着系统配送时效的不断提升, 导致的成本和排放的代价将更加显著. 本文所提出的模型和算法在丰富和拓展VRP相关理论研究的同时, 能够为物流企业配送优化决策提供科学依据. 未来的研究将围绕进一步提升算法效率及需

求可拆分的多舱车车辆调度问题展开。

参考文献:

- [1] QI Yuanhang, CAI Yanguan, CAI Hao, et al. Voronoi diagram-based discrete bat algorithm for multi-depot vehicle routing problem. *Control Theory & Applications*, 2018, 35(8): 1142 – 1150.
(戚远航, 蔡延光, 蔡颢, 等. 泰森多边形的离散蝙蝠算法求解多车场车辆路径问题. *控制理论与应用*, 2018, 35(8): 1142 – 1150.)
- [2] MA Bingshan, HU Dawei, CHEN Xiqiong, et al. An optimization of pure electric vehicle routing problem on half-open multi-distribution center. *Journal of Transportation Systems Engineering and Information Technology*, 2019, 19(6): 199 – 205.
(马冰山, 胡大伟, 陈希琼, 等. 半开放式的多配送中心纯电动车辆路径优化问题. *交通运输系统工程与信息*, 2019, 19(6): 199 – 205.)
- [3] LI Yang, HU Rong, QIAN Bin, et al. Two-stage algorithm for solving multi-depot vehicle routing problem. *Information and Control*, 2020, 49(6): 752 – 760.
(李洋, 胡蓉, 钱斌, 等. 两阶段算法求解多车场车辆路径问题. *信息与控制*, 2020, 49(6): 752 – 760.)
- [4] SADATI M E H, CATAY B, AKSEN D. An efficient variable neighborhood search with tabu shaking for a class of multi-depot vehicle routing problems. *Computers & Operations Research*, 2021, 133: 105269.
- [5] FAN Houming, ZHANG Xuan, REN Xiaoxue, et al. Optimization of multi-depot open split delivery vehicle routing problem with simultaneous delivery and pick-up. *System Engineering—Theory & Practice*, 2021, 41(6): 1521 – 1534.
(范厚明, 张轩, 任晓雪, 等. 多中心开放且需求可拆分的VRPSDP问题优化. *系统工程理论与实践*, 2021, 41(6): 1521 – 1534.)
- [6] WANG Y, LI Q, GUAN X, et al. Collaborative multi-depot pickup and delivery vehicle routing problem with split loads and time windows. *Knowledge-Based Systems*, 2021, 231: 107412.
- [7] ZHANG L Y, TSENG M L, WANG C H, et al. Low-carbon cold chain logistics using ribonucleic acid-ant colony optimization algorithm. *Journal of Cleaner Production*, 2019, 233: 169 – 180.
- [8] ZHOU Xiancheng, LÜ Yang, HE Caihong, et al. Multi-depot green vehicle routing model and its optimization algorithm with time-varying speed. *Control and Decision*, 2022, 37(2): 473 – 482.
(周鲜成, 吕阳, 贺彩虹, 等. 考虑时变速度的多车场绿色车辆路径模型及优化算法. *控制与决策*, 2022, 37(2): 473 – 482.)
- [9] ZHU Lan, MA Xiao, LIU Zhuofan. Time-dependent green vehicle routing problem. *Journal of Transportation Systems Engineering and Information Technology*, 2021, 21(6): 187 – 194.
(珠兰, 马潇, 刘卓凡. 时间依赖型绿色车辆路径问题研究. *交通运输系统工程与信息*, 2021, 21(6): 187 – 194.)
- [10] GE Xianlong, RAN Xiaofen. Pollution routing problem based on time window assignment. *Computer Integrated Manufacturing Systems*, 2021, 27(4): 1178 – 1187.
(葛显龙, 冉小芬. 基于时间窗指派的污染路径问题. *计算机集成制造系统*, 2021, 27(4): 1178 – 1187.)
- [11] HU Rong, CHEN Wenbo, QIAN Bin, et al. Learning ant colony algorithm for green multi-depot vehicle routing problem. *Journal of System Simulation*, 2021, 33(9): 2095 – 2108.
(胡蓉, 陈文博, 钱斌, 等. 学习型蚁群算法求解绿色多车场车辆路径问题. *系统仿真学报*, 2021, 33(9): 2095 – 2108.)
- [12] SADATI M E H, SATAY B. A hybrid variable neighborhood search approach for the multi-depot green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 2021, 149: 102293.
- [13] CHEN J, SHI J. A multi-compartment vehicle routing problem with time windows for urban distribution – a comparison study on particle swarm optimization algorithms. *Computers & Industrial Engineering*, 2019, 133: 95 – 106.
- [14] FRANK M, OSTERMEIER M, HOLZAPFEL A, et al. Optimizing routing and delivery patterns with multi-compartment vehicles. *European Journal of Operational Research*, 2021, 293(2): 495 – 510.
- [15] HEBLER K. Exact algorithms for the multi-compartment vehicle routing problem with flexible compartment sizes. *European Journal of Operational Research*, 2021, 294(1): 188 – 205.
- [16] CHEN L, LIU Y, LANGEVIN A. A multi-compartment vehicle routing problem in cold-chain distribution. *Computers & Operations Research*, 2019, 111: 58 – 66.
- [17] CHEN J, DAN B, SHI J. A variable neighborhood search approach for the multi-compartment vehicle routing problem with time windows considering carbon emission. *Journal of Cleaner Production*, 2020, 277: 123932.
- [18] ALINAGHIAN M, SHOKOUI N. Multi-depot multi-compartment vehicle routing problem, solved by a hybrid adaptive large neighborhood search. *Omega*, 2018, 76: 85 – 99.
- [19] LUO Liang, CHEN Huixuan, WU Zhang, et al. Vehicle routes planning of cold chain distribution for fresh agricultural product based on the dual functions of the traffic and weather conditions. *Systems Engineering*, 2022, 40(6): 67 – 75.
(罗亮, 陈慧璇, 吴张, 等. 交通与天气状况双重作用下生鲜农产品冷链配送的VRPTW. *系统工程*, 2022, 40(6): 67 – 75.)
- [20] BRANDAO J. A memory-based iterated local search algorithm for the multi-depot open vehicle routing problem. *European Journal of Operational Research*, 2020, 284(2): 559 – 571.
- [21] BROYSY O, GENDREAU M. Vehicle routing problem with time windows, part I: Route construction and local search algorithms. *Transportation Science*, 2005, 39(1): 104 – 118.
- [22] 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.
- [23] YENIAY O. Penalty function methods for constrained optimization with genetic algorithms. *Mathematical and Computational Applications*, 2005, 10(1): 45 – 56.
- [24] SAVELSBERGH M W. The vehicle routing problem with time windows: Minimizing route duration. *ORSA Journal on Computing*, 1992, 4(2): 146 – 154.
- [25] DUECK G. New optimization heuristics: The great deluge algorithm and the record-to-record travel. *Journal of Computational Physics*, 1993, 104(1): 86 – 92.
- [26] LI J, LI T, YU Y, et al. Discrete firefly algorithm with compound neighborhoods for asymmetric multi-depot vehicle routing problem in the maintenance of farm machinery. *Applied Soft Computing*, 2019, 81: 105460.

作者简介:

肖友刚 教授, 博士生导师, 目前研究方向为自抗扰控制、智能控制, E-mail: csuxyg@csu.edu.cn;

曹健 硕士研究生, 研究方向为车辆路径问题求解算法及应用, E-mail: 13142120915@163.com;

陈婉茹 博士研究生, 研究方向为物流系统建模优化、智能优化算法研究, E-mail: wrchen@csu.edu.cn;

张得志 教授, 博士生导师, 研究方向为物流系统工程与管理、低碳物流系统优化, E-mail: dzzhang@csu.edu.cn;

李双艳 副教授, 硕士生导师, 研究方向为绿色供应链、生物质回收物流, E-mail: lishuangyan585@163.com.