

带时间窗的时间依赖型同时取送货车辆路径问题研究

何美玲¹, 杨梅¹, 韩珣^{*2a, 2b}, 武晓晖¹

(1. 江苏大学, 汽车与交通工程学院, 江苏 镇江 212013; 2. 四川警察学院, a. 智能警务四川省重点实验室,
b. 道路交通管理系, 四川 泸州 646000)

摘要: 针对带时间窗的时间依赖型同时取送货车辆路径问题(Time Dependent Vehicle Routing Problem with Simultaneous Pickup-Delivery and Time Windows, TDVRPSPDTW), 本文建立以车辆固定成本、驾驶员成本、燃油消耗及碳排放成本之和为优化目标的数学模型; 并在传统蚁群算法的基础上, 利用节约启发式构造初始解初始化信息素, 改进状态转移规则, 引入局部搜索策略, 提出一种带自适应大邻域搜索的混合蚁群算法(Ant Colony Optimization with Adaptive Large Neighborhood Search, ACO-ALNS)进行求解; 最后, 分别选取基准问题算例和改编生成TDVRPSPDTW算例进行实验。实验结果表明: 本文提出的ACO-ALNS算法可有效解决TDVRPSPDTW的基准问题; 相较于模拟退火算法和带局部搜索的蚁群算法, 本文算法求解得到的总配送成本最优值平均分别改善7.56%和2.90%; 另外, 相比于仅考虑碳排放或配送时间的模型, 本文所构建的模型综合多种因素, 总配送成本平均分别降低4.38%和3.18%, 可有效提高物流企业的经济效益。

关键词: 物流工程; 同时取送货车辆路径问题; 蚁群算法; 时间依赖; 时间窗

Time-dependent Vehicle Routing Optimization Considering Simultaneous Pickup-delivery and Time Windows

HE Meiling¹, YANG Mei¹, HAN Xun^{*2a, 2b}, WU Xiaohui¹

(1. School of Automotive and Traffic Engineering, Jiangsu University, Zhenjiang 212013, Jiangsu, China;
2a. Intelligent Policing Key Laboratory of Sichuan Province, 2b. Department of Transportation Management,
Sichuan Police College, Luzhou 646000, Sichuan, China)

Abstract: To solve the time-dependent vehicle routing problem with simultaneous pickup-delivery and time windows (TDVRPSPDTW), this paper proposes a mathematical model with the sum of vehicle fixed cost, driver cost, fuel consumption and carbon emission cost as the optimization objective. Based on the traditional ant colony optimization, this paper introduces a hybrid ant colony optimization with adaptive large neighborhood search (ACO-ALNS). It uses heuristic initialization of pheromones, improves state transition rules, and uses local search strategies to improve solution quality. Benchmark problem instances and adapted TDVRPSPDTW instances are utilized for experimentation. The experimental results demonstrate the effectiveness of the proposed ACO-ALNS algorithm in solving the benchmark problem of TDVRPSPDTW. Compared to the simulated annealing and ant colony optimization with local search, the proposed algorithm improves the optimal value of total distribution cost by an average of 7.56% and 2.90%, respectively. In addition, the presented model incorporates multiple factors, resulting in an average reduction of 4.38% and 3.18% in total distribution costs compared to models that only consider carbon emissions or delivery time. This

收稿日期: 2024-04-14

修回日期: 2024-05-06

录用日期: 2024-05-13

基金项目: 教育部人文社会科学研究青年基金/ The Humanity and Social Science Youth Foundation of Ministry of Education of China (21YJCZH180); 智能警务四川省重点实验室开放课题/ Opening Project of Intelligent Policing Key Laboratory of Sichuan Province (ZNJW2023KFMS004); 江苏省研究生科研创新计划项目/ Jiangsu Province Graduate Research Innovation Program (KYCX22_3674)。

作者简介: 何美玲(1982-), 女, 江苏江都人, 副教授。

*通信作者: hanxun@scpolicec.edu.cn

improvement can effectively enhance the economic benefits of logistics enterprises.

Keywords: logistics engineering; vehicle routing problem with simultaneous pickup-delivery; ant colony optimization; time-dependent; time windows

0 引言

随着现代物流业与电子商务的快速发展,越来越多的客户同时存在取货和送货需求及严格的配送时间窗限制,对配送时效的要求也愈发严苛,若只考虑单一的取货或送货需求进行配送服务,会致使车辆空载率高、交通拥堵加剧、配送效率低及配送成本高等问题。此外,实际生活中,车辆的行驶速度并非恒定,受路网交通状况的影响处于不断变化,在此背景下,衍生的带时间窗的时间依赖型同时取送货车辆路径问题(TDVRPSPDTW)研究具有重要意义。TDVRPSPDTW是经典车辆路径问题(Vehicle Routing Problem, VRP)的重要扩展,包含了当前广受关注的两个变体:带时间窗的时间依赖型车辆路径问题(Time-Dependent Vehicle Routing Problem with Time Windows, TDVRPTW)和带时间窗的同时取送货车辆路径问题(Vehicle Routing Problem with Simultaneous Pickup-Delivery and Time Windows, VRPSPDTW)。

关于TDVRPTW,ZHOU等^[1]以电动汽车为配送载体,提出带有部分模型的可变邻域搜索算法最小化由行驶距离、车辆使用数量及能源消耗构成的总成本。周鲜成等^[2]指出,时变路网下,行驶距离最短对应的行驶时间并不一定最短,因此,在利用运输领域估算空气污染物排放方法,即MEET模型度量油耗与碳排放成本的基础上,通过行驶时间计算车辆使用成本和人力成本,构建时间依赖型的绿色车辆路径模型。珠兰等^[3]在构建TDVRPTW模型时,将碳排放因素考虑在内,提出两阶段混合搜索算法求解,减少环境污染。

VRPSPDTW是在同时取送货车辆路径问题(Vehicle Routing Problem with Simultaneous Pickup and Delivery, VRPSPD)的基础上考虑客户点的时间窗约束,现有相关研究主要在考虑问题特性与目标函数存在差异。WANG等^[4]和LIU等^[5]分别提出协同进化遗传算法和混合爬山算法求解VRPSPDTW,以实现车辆使用数量和行驶距离加权和的最小化。辛禹辰等^[6]考虑客户需求或时间窗频繁变动对配送方案的影响,提出基于等待策略的干扰管理方法,构建VRPSPD模型。随着低碳理念的发展,在VRPSPDTW中亦考虑碳排放特性。

ZHOU等^[7]放宽时间窗约束,在考虑客户点不确定性取货需求的同时,将燃油消耗和碳排放纳入优化目标,但却假设车辆行驶速度恒定,忽略了车辆速度变化对油耗与碳排放量的影响。

上述研究仅在时间窗约束下单独关注车辆路径问题的时间依赖性 or 客户潜在的同时取送货需求特性,未能综合考虑两种特性,现阶段内,以TDVRPSPDTW为对象展开的研究还十分有限。ZHOU等^[8]虽将TDVRPSPDTW扩展到两级车辆路径问题,提出一种模因算法最小化运输、装卸、库存及碳排放成本,但其碳排放的度量仅与距离、载重及等待时间相关。

综上,本文以TDVRPSPDTW为研究对象,考虑车辆的时变速度、载重及发动机性能等因素,采用综合排放模型度量燃油消耗及碳排放,以车辆固定成本、驾驶员成本、燃油消耗及碳排放成本之和构成的总配送成本最小化为目标函数建立数学模型,并依据问题特点改进蚁群算法,提出一种带自适应大邻域局部搜索策略的混合蚁群算法(ACO-ALNS)进行求解,为物流企业制定合理配送方案提供理论借鉴与参考。

1 问题描述及数学模型

1.1 问题描述

本文研究的TDVRPSPDTW问题可描述为:某配送中心拥有若干同质车辆,向一定数量的客户点提供同时取送货服务。每个客户点的时间窗、送货需求及取货需求是固定已知的。车辆从配送中心出发,行驶速度具有时变特性,因行驶时段的不同而发生变化,完成客户点的访问任务后,最终返回配送中心。车辆需在客户点给定的时间窗内进行服务,若其在客户点最早允许的服务时间之前到达,则进行等待。要求在满足所有客户点需求的前提下,以总配送成本最小规划行驶路线。如图1所示。

1.2 时间依赖函数

本文采用ICHOUA等^[9]提出的基于行程车速的时间依赖函数描述时变路网特性,认为车辆在足够短时间段内的实时行驶速度可以视作其在该时间段内的固定速度,如图2所示。

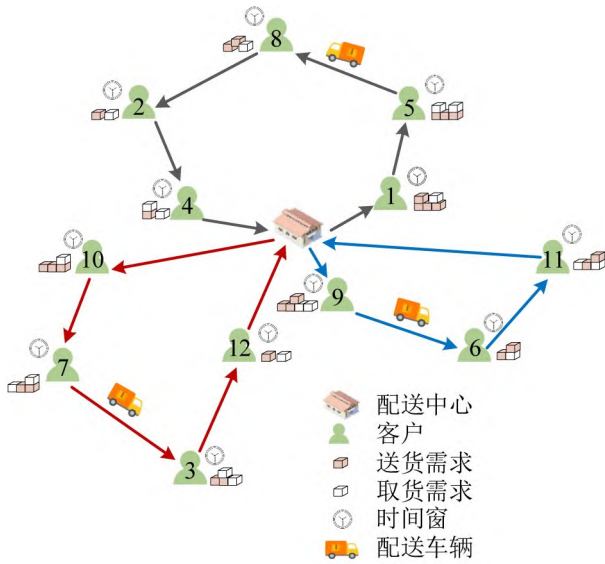


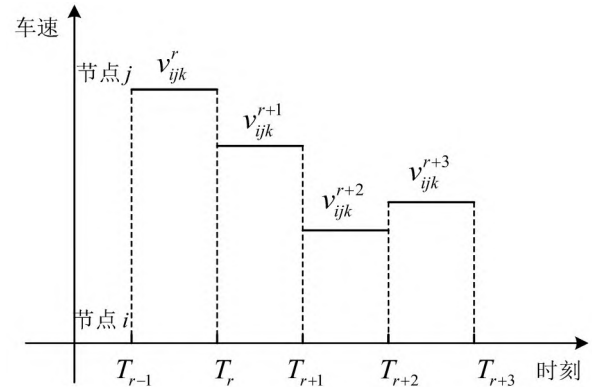
图1 TDVRPSPDTW示意图

Fig. 1 Schematic diagram of TDVRPSPDTW

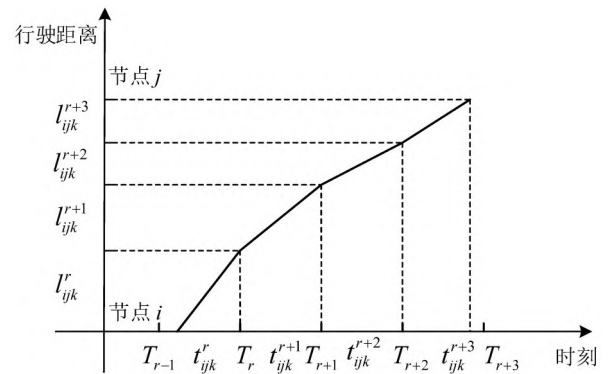
将配送中心的开放时间划分为 R 个长度为 L 的时间段, 不同时间段的车速不同。 $T = \{T_0, T_1, \dots, T_R\}$ 为时间点集合, $[T_{r-1}, T_r]$ 表示第 r 个时间段, 该时段内配送车辆 k 由客户点 i 前往客户点 j 的车速为 v_{ijk}^r , 行驶距离为 l_{ij}^r , 行驶时间为 t_{ijk}^r , l_{ij} 为客户点 i 到客户点 j 的距离, U_{ij}^r 为车辆 k 在第 r 个时间段后继续由客户点 i 前往客户点 j 所需的剩余距离。假设配送车辆 k 从客户点 i 出发的时刻 $c_{ik} \in [T_{r-1}, T_r]$, 车辆 k 从客户点 i 到客户点 j 的总行驶时间 t_{ijk} 通过以下步骤计算:

Step 1 计算开始时间段内的行驶时间。 $l_{ijk}^r = v_{ijk}^r(T_r - c_{ik})$, 若 $l_{ijk}^r \geq l_{ij}$, 则 $t_{ijk}^r = l_{ij} / v_{ijk}^r$, $t_{ijk} = t_{ijk}^r$, 结束计算, 输出 t_{ijk} ; 否则, $U_{ij}^r = l_{ij} - l_{ijk}^r$, $t_{ijk}^r = T_r - c_{ik}$, $r = r + 1$ 。

Step 2 计算其他时间段内的行驶时间。 $l_{ijk}^r = v_{ijk}^r L$, 若 $l_{ijk}^r < U_{ij}^{r-1}$, 则 $U_{ij}^r = U_{ij}^{r-1} - l_{ijk}^r$, $t_{ijk}^r = L$, $r = r + 1$, 重复 Step 2; 否则, $t_{ijk}^r = U_{ij}^r / v_{ijk}^r$, $t_{ijk} = \sum_{r \in T} t_{ijk}^r$, 结束计算, 输出 t_{ijk} 。



(a) 不同时间段的车速



(b) 不同时间段的行驶距离与行驶时间

图2 车辆的离散速度和连续行驶时间

Fig. 2 Discrete speed and continuous travel time of vehicles

1.3 数学模型

模型参数及变量如表1和表2所示。

表1 TDVRPSPDTW 模型参数

Table 1 Parameters of TDVRPSPDTW model

参数	含义	参数	含义
N	客户点集合, $N = \{1, 2, \dots, n\}$	C_r	单位燃油消耗成本
N_0	所有节点集合, $N_0 = \{0\} \cup N$	C_p	单位碳排放成本
V	所有车辆集合, $V = \{1, 2, \dots, k\}$	e	碳排放系数
T	时间间隔集合, $T = \{1, 2, \dots, r\}$	S_r	第 r 时间段的开始时刻
d_j	客户点 j 的送货需求量, $j \in N$	E_r	第 r 时间段的结束时刻
p_j	客户点 j 的取货需求量, $j \in N$	v_{ijk}^r	第 r 时间段, 车辆 k 从节点 i 到节点 j 的车速, $i, j \in N_0$
l_{ij}	节点 i 与节点 j 之间的距离, $i, j \in N_0$	t_{ijk}^r	第 r 时间段, 车辆 k 从节点 i 到节点 j 的行驶时间, $i, j \in N_0$
$[a_0, b_0]$	配送中心的开放时间	l_{ijk}^r	第 r 时间段, 车辆 k 从节点 i 到节点 j 的行驶距离, $i, j \in N_0$
R	配送中心开放时间分段数	λ	燃料的热值
$[a_i, b_i]$	客户点 i 的时间窗, $i \in N$	C	发动机摩擦系数
s_i	客户点 i 的服务时长, $i \in N$	Z	发动机转速
W	车辆最大载重	P	发动机排量
m_0	空车自重	∂	传动系数
C_f	车辆固定成本	g	重力常数
C_t	驾驶员单位时间成本	δ	空气阻力系数

表2 TDVRPSPDTW模型变量
Table 2 Variables of TDVRPSPDTW model

符号	含义
x_{ijk}^r	车辆 k 在第 r 时间段内是否由节点 i 访问节点 j , 是为1, 否为0
y_{ik}	客户点 i 是否由车辆 k 服务, 是为1, 否为0
h_{ik}	车辆 k 到达客户点 i 的时间
w_{ik}	车辆 k 在客户点 i 的等待时长, 若 $h_{ik} < a_i$, $w_{ik} = a_i - h_{ik}$; 否则, $w_{ik} = 0$
c_{ik}	车辆 k 离开客户点 i 的时间
Q_{ijk}	车辆 k 从节点 i 前往节点 j 的载重

本文以车辆固定成本、驾驶员成本、燃油消耗及碳排放成本构成的总配送成本为优化目标, 其中, 燃油消耗及碳排放成本采用综合排放模型计算油耗得到。根据该模型, 车辆 k 在时间段 r 内从节点 i 到 j 的油耗 f_{ijk}^r 为

$$f_{ijk}^r = \lambda \left[CZPI_{ijk}^r / v_{ijk}^r + \partial g(m_0 + Q_{ijk}) l_{ijk}^r + \partial \delta l_{ijk}^r (v_{ijk}^r)^2 \right] \quad (1)$$

构建的完整数学模型为

$$Z_1 = \min F = C_f \sum_{k \in V} \sum_{j \in N} \sum_{r \in T} x_{ijk}^r + C_l \left(\sum_{k \in V} \sum_{i, j \in N_0} \sum_{r \in T} l_{ijk}^r x_{ijk}^r + \sum_{k \in V} \sum_{i \in N} \sum_{j \in N} \sum_{r \in T} w_{ik} y_{ik} + \sum_{i \in N} s_i \right) + C_r \sum_{k \in V} \sum_{i, j \in N_0} \sum_{r \in T} f_{ijk}^r x_{ijk}^r + C_p e \sum_{k \in V} \sum_{i, j \in N_0} \sum_{r \in T} f_{ijk}^r x_{ijk}^r \quad (2)$$

约束条件为

$$\sum_{k \in V} \sum_{i \in N_0} \sum_{r \in T} x_{ijk}^r = 1, \forall j \in N \quad (3)$$

$$\sum_{i \in N_0} \sum_{r \in T} x_{ijk}^r - \sum_{i \in N_0} \sum_{r \in T} x_{jik}^r = 0, \forall j \in N; \forall k \in V \quad (4)$$

$$\sum_{j \in N_0} Q_{ijk} = \sum_{i, j \in N} \sum_{r \in T} x_{ijk}^r d_j \leq W, \forall k \in V \quad (5)$$

$$\sum_{i \in N_0} Q_{i0k} = \sum_{i, j \in N} \sum_{r \in T} x_{ijk}^r p_i \leq W, \forall k \in V \quad (6)$$

$$\sum_{i \in N_0} \sum_{r \in T} x_{ijk}^r (Q_{ijk} - d_j) = \sum_{i \in N_0} \sum_{r \in T} x_{jik}^r (Q_{jik} - p_j), \forall j \in N; \forall k \in V \quad (7)$$

$$0 \leq Q_{ijk} \leq W, \forall i, j \in N_0; \forall k \in V \quad (8)$$

$$S_r x_{ijk}^r \leq c_{ik} x_{ijk}^r \leq E_r x_{ijk}^r, \forall i, j \in N_0; \forall k \in V; \forall r \in T \quad (9)$$

$$\sum_{r \in T} x_{ijk}^r (h_{ik} + w_{ik} + s_i + t_{ijk}^r) \leq h_{jk}, \forall i, j \in N_0; \forall k \in V \quad (10)$$

$$a_i \leq h_{ik} + w_{ik} \leq b_i, \forall i \in N_0; \forall k \in V \quad (11)$$

$$c_{ik} = h_{ik} + w_{ik} + s_i, \forall i \in N_0; \forall k \in V \quad (12)$$

$$\sum_{j \in N} \sum_{r \in T} x_{ijk}^r \leq 1, \forall k \in V \quad (13)$$

$$x_{ijk}^r \in \{0, 1\}, \forall i, j \in N_0; \forall k \in V; \forall r \in T \quad (14)$$

式(2)表示最小化总配送成本 F , 其中, 第1项为车辆固定成本 F_1 , 第2项为驾驶员成本 F_2 , 后2项分别为燃油消耗成本 F_3 和碳排放成本 F_4 ;

式(3)确保每个客户只被访问一次; 式(4)表示车辆到达客户点服务完毕之后, 从此处离开, 保证路径的连续性; 式(5)表示车辆从配送中心出发的载重等于所有要访问客户点的送货量之和; 式(6)表示车辆回到配送中心时的载重为所有已访问客户点的取货量之和; 式(7)表示车辆经由客户点后的载重平衡; 式(8)表示车辆在任何节点处均不超过其最大载重量; 式(9)保证车辆在特定的时间间隔内离开; 式(10)表示车辆到达节点时间的约束; 式(11)表示客户点访问的时间窗约束; 式(12)表示车辆 k 离开节点 i 的时间等于到达时间、等待时长与服务时长之和; 式(13)确保每辆车仅使用一次; 式(14)为决策变量属性约束。

2 混合蚁群算法

相比于一般的局部搜索算法, 自适应大邻域搜索算法 (Adaptive Large Neighborhood Search, ALNS) 使用多种破坏和修复算子, 可对更大的邻域空间进行搜索, 获得质量更高的解。基于此, 本文提出一种带自适应大邻域局部搜索的混合蚁群算法 ACO-ALNS 求解 TDVRPSPDTW, 求解流程如图3所示。首先, 利用节约启发式初始化信息素值, 弥补 ACO 前期信息素匮乏的不足; 其次, 结合研究问题特点改进蚂蚁的状态转移规则; 最后, 嵌入 ALNS 局部搜索策略, 帮助其跳出局部最优, 增强算法的整体寻优能力。

2.1 初始化

本文采用节约启发式方法初始化信息素值 τ_0 为

$$\tau_0 = 1/nL_0 \quad (15)$$

式中: n 为服务客户的数量; L_0 为节约启发式构造配送方案的行驶距离。

2.2 状态转移规则

本文采取伪随机概率选择, 即通过给定的转移控制参数 $r_0 \in [0, 1]$ 与随机数 $r_1 \in [0, 1]$ 的关系进行确定性和探索性选择节点, 并引入距离节约值、等待时间及时间窗宽度改进蚂蚁的状态转移规则。当蚂蚁 k 从节点 i 选择下一节点 j 时, 其状态转移规则为

$$j^k = \begin{cases} \operatorname{argmax}_{j \in M_i^k} \left[(\tau_{ij})^\alpha (\varphi_{ij}/l_{ij})^\beta (1/\chi_j)^\omega (1/\vartheta_j)^\gamma \right], & r_1 \leq r_0 \\ p_{ij}^k, & r_1 > r_0 \end{cases} \quad (16)$$

$$p_{ij}^k = \begin{cases} \frac{(\tau_{ij})^\alpha (\varphi_{ij}/l_{ij})^\beta (1/\chi_j)^\omega (1/\vartheta_j)^\gamma}{\sum_{b \in M_i^k} (\tau_{ib})^\alpha (\varphi_{ib}/l_{ib})^\beta (1/\chi_b)^\omega (1/\vartheta_b)^\gamma}, & j \in M_i^k \\ 0, & j \notin M_i^k \end{cases} \quad (17)$$

式中: α 为信息素重要程度因子; β 为能见度重要程度因子; ω 为等待时间重要程度因子; γ 为时间窗宽度重要程度因子; τ_{ij} 为节点间的信息素浓度;

l_{ij} 为节点间的距离; p_{ij}^k 为蚂蚁 k 从节点 i 选择下一节点 j 的状态转移概率; M_i^k 为当蚂蚁 k 在节点 i 时由满足车辆载重、时间窗约束的未配送节点构成的集合; φ_{ij} 为距离节约值; χ_j 为客户点 j 的等待时间, 即客户点 j 的左时间窗 a_j 与车辆到达该点的时间 h_j 之差; ϑ_j 为客户点 j 的时间窗宽度, 即其左右时间窗之差。

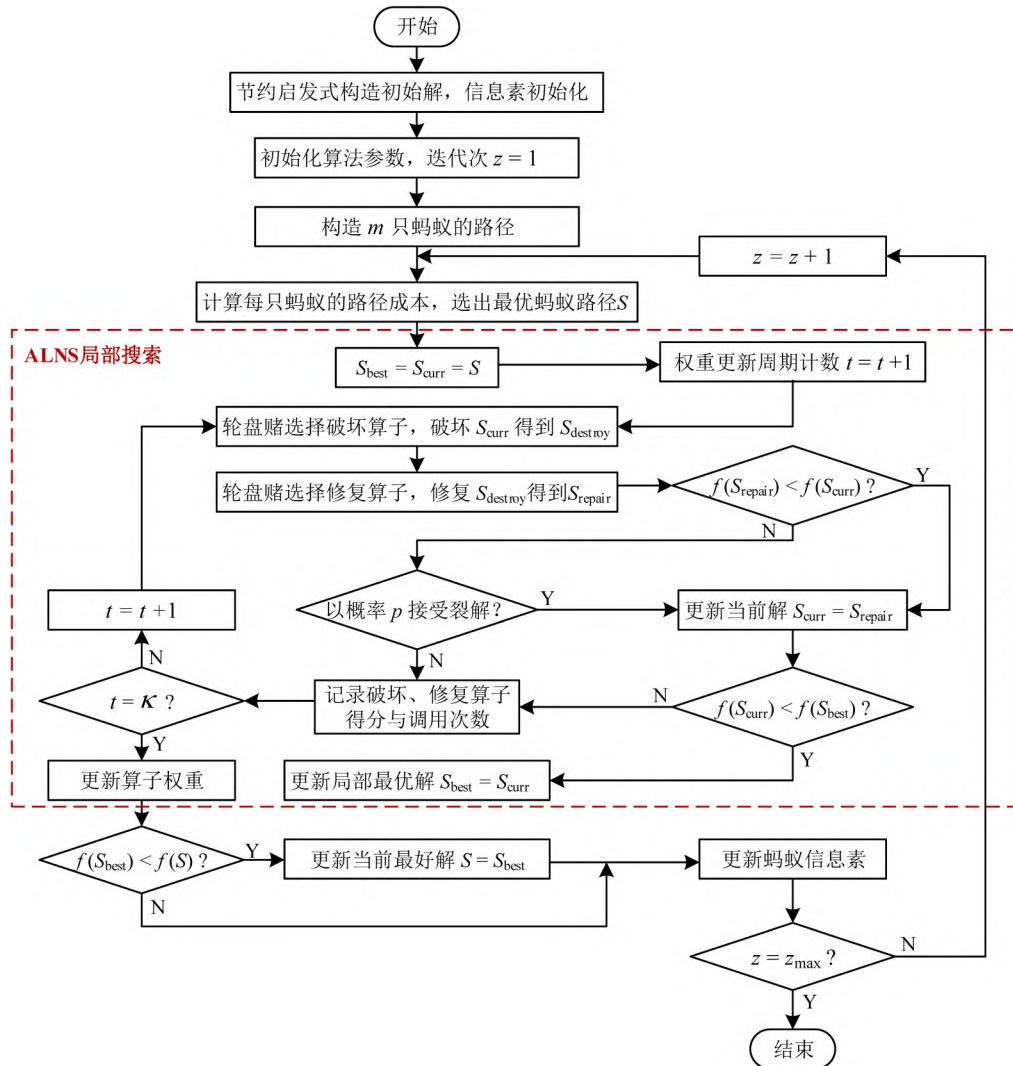


图 3 ACO-ALNS 算法的求解流程

Fig. 3 Flowchart of ACO-ALNS algorithm

$$\varphi_{ij} = l_{i0} + l_{0j} - l_{ij} \quad (18)$$

$$\chi_j = a_j - h_j, h_j < a_j \quad (19)$$

$$\vartheta_j = b_j - a_j \quad (20)$$

式中: l_{i0} (l_{0j}) 为客户点 i (配送中心) 到配送中心 (客户点 j) 的距离。

2.3 局部搜索的 ALNS 算法

本文利用 ALNS 算法对最优蚂蚁路径进行局部搜索, 以获得更高质量的解决方案。ALNS 算法

中的每种算子均被分配一个选择概率 ψ_u , 即

$$\psi_u = \pi_u / \sum_{u=1}^m \pi_u, u = 1, \dots, m \quad (21)$$

式中: π_u 为算子 u 的权重; m 为算子个数。

迭代初始时, 每种算子有相同的初始权重, 基于 ψ_i 通过轮盘赌选择其中一种破坏算子对当前解 S_{curr} 进行破坏, 再以同样方式选择修复算子进行修复操作, 得到修复解 S_{repair} , 然后, 根据算子的表现

给予不同的分值,得分越高,表明该组算子性能越好,具体得分规则如下($\mu_1 > \mu_2 > \mu_3 > 0$):

(1) S_{repair} 优于当前解,且是全局最优解,得 μ_1 分;

(2) S_{repair} 优于当前解,得 μ_2 分;

(3) S_{repair} 以一定的概率 p (Metropolis接受准则)被接受作为当前解,得 μ_3 分。

在迭代过程中记录算子的调用次数与得分情况,达到周期 κ 后,动态更新算子权重,即

$$\pi_u = (1 - \zeta)\pi_u + \zeta\theta_u / \eta_u \quad (22)$$

式中: ζ 为算子权重更新系数; θ_u 为算子 u 在给定更新周期内的累计得分; η_u 为算子 u 在给定更新周期内的累计调用次数。 θ_u 和 η_u 在每次权重更新后重置为0。

2.3.1 破坏算子

本文设计4种破坏算子,具体如下。

(1) 相似性破坏算子

通过删除路径中相似的客户节点实现对当前解的破坏。衡量两个客户点 i 和 j 的相似性 $X(i, j)$ 时,本文考虑节点间的距离、时间窗、取送货需求量以及是否位于同一路径这4个因素。该算子的步骤为:随机选择一个客户点 i 删除,将其添加到删除列表 G 中,计算各个未删除客户点与客户点 i 的相似性,按照相似性从小到大的顺序排列后,选择第 $\text{rand}^{|G|}$ 个客户点删除,其中, rand 为 $[0, 1]$ 间的随机数; H 为平衡随机性与相关性选择的位置参数, $|G|$ 为删除列表的大小。重复此步骤,直至删除 q 个客户点。

$$X(i, j) = \rho_1 \frac{l_{ij}}{\max_{i, j \in N} (l_{ij})} + \rho_2 \frac{|d_i - d_j| + |p_i - p_j|}{\max_{i, j \in N} (|d_i - d_j| + |p_i - p_j|)} + \rho_3 \frac{|a_i - a_j| + |b_i - b_j|}{\max_{i, j \in N} (|a_i - a_j| + |b_i - b_j|)} + \rho_4 D \quad (23)$$

式中: D 为客户点 i 和 j 是否在同一路径上,若是为1,否则为0; ρ_1 、 ρ_2 、 ρ_3 、 ρ_4 分别为行驶距离、时间窗、取送货需求量以及是否同一路径的权重系数。

(2) 随机破坏算子

随机选取当前解中的 q 个客户点删除,并放入删除列表 G 中,得到破坏解 S_{destroy} 。

(3) 成本最差破坏算子

删除对目标成本影响最大的客户点。迭代时,将所有未删除客户点按从当前解决方案删除所带

来的成本减少值降序排列,选择成本减少值最大的客户点删除,重复此步骤,直至删除 q 个客户点。

(4) 路径最差破坏算子

选择客户数量最少的路径,删除该路径中的所有客户点。

2.3.2 修复算子

为将删除列表 G 中的客户点插入最优位置,本文的修复算子如下。

(1) 贪婪修复算子

将 G 中插入成本最低的客户点优先插入到路径中,以得到修复解 S_{repair} 。插入成本为插入客户点前后的成本之差,在满足所有约束的前提下,计算 G 中每客户点在每条路径中的最佳插入位置和相应的插入成本。将所有客户点按插入成本进行排序,选取插入成本最小的客户点插入其对应的最优位置,并从 G 中删除该客户点。更新集合 G 中的客户点,重新计算其他客户点的插入成本,进行排序和插入操作,重复进行以上步骤,直至 G 为空。

(2) 后悔值修复算子

在贪婪修复算子的基础上,同时考虑最优与次优配送路径,具有很好的前瞻性。在满足时间窗以及车辆装载容量约束的前提下,计算 G 中每个客户点的后悔值,即客户点在最好插入位置上相应的插入成本与第二好插入位置上的插入成本之差。将所有客户点按后悔值进行排序,选取后悔值最大的客户点,按照最好插入位置进行修复,并从 G 中删除该客户点。更新集合 G 中的客户点,重新计算其他客户点的后悔值,进行排序和插入操作,重复进行以上步骤,直至 G 为空。

2.4 信息素更新规则

采用全局信息素更新方式,更新式为

$$\tau_{ij} = (1 - \phi)\tau_{ij} + \Delta\tau_{ij}^{\text{bs}} \quad (24)$$

$$\Delta\tau_{ij}^{\text{bs}} = \begin{cases} B/nL_{\text{best}}, & \text{若弧}(i, j)\text{属于当前最好路径} \\ 0, & \text{否则} \end{cases} \quad (25)$$

式中: ϕ 为信息素挥发因子; $\Delta\tau_{ij}^{\text{bs}}$ 为弧 (i, j) 上的信息素增量; B 为信息素强度; L_{best} 为当前迭代中最好解的路径长度。

3 实验及结果分析

3.1 算法及模型参数设置

经过反复实验,算法的参数设置如下:最大迭代次数 $z_{\text{max}} = 100$, 蚂蚁数量 $m = 40$, 信息素重要程度因子 $\alpha = 2$, 能见度重要程度因子 $\beta = 1$, 等待时

间重要程度因子 $\omega=3$, 时间窗宽度重要程度因子 $\gamma=2$, 转移控制参数 $r_0=0.75$, 信息素挥发因子 $\phi=0.25$, 信息素浓度 $B=100$; ALNS 中, 删除客户数量 $q \in [0.2n, 0.4n]$, 相似性破坏算子中权重系数 $\rho_1, \rho_2, \rho_3, \rho_4$ 分别为 0.4、0.3、0.2、0.1, 得分规则 μ_1, μ_2, μ_3 依次为 10、5、0, 算子权重更新周期 $\kappa=80$, 更新系数 $\zeta=0.3$ 。算法代码使用 MATLAB R2019b 编写, 在 Intel Core i5-1035G1 1.00 GHz (16.00 GB RAM) 和 Windows10 操作系统的环境下运行。

模型中, 涉及的车辆参数取值参考 MAJIDI^[10] 的研究, 其余参数参考文献 [7]: 车辆固定成本 C_f 为 250 元·辆⁻¹, 驾驶员单位时间成本 C_l 为 0.5 元·min⁻¹, 单位燃油消耗成本 C_r 为 6.43 元·L⁻¹, 单位碳排放成本 C_p 为 0.05 元·kg⁻¹, 碳排放系数 e 为 3.0959 CO₂·kg⁻¹。

3.2 算法有效性验证

当前, 还未有 TDVRSPDTW 的测试基准算例, 而 TDVRPTW 和 VRSPDTW 问题是本文研究的基础。因此, 基于 TDVRPTW 和 VRSPDTW 的通用算例, 进行本文提出 ACO-ALNS 算法有效性的验证。

(1) TDVRPTW 算例

TDVRPTW 算例来自于文献 [11], 该算例是 Figliozzi 在 Solomon 构造的 VRPTW 算例基础上, 假设配送中心的开放时间 $[a_0, b_0]$, 即 $[0, 1236]$ 代表车辆 1 d 的工作时间, 并将其均分为 5 个时间段: $[0, 0.2b_0)$ 、 $[0.2b_0, 0.4b_0)$ 、 $[0.4b_0, 0.6b_0)$ 、 $[0.6b_0, 0.8b_0)$ 和 $[0.8b_0, b_0]$, 增加 4 类考虑不同情况的速度时间依赖函数(a、b、c、d)创建而成。本文以 C1 类算例为例, 设置车辆数为主要目标, 车辆行驶距离为次要目标进行实验。

本文 ACO-ALNS 算法求解 TDVRPTW 算例 10 次的最优解与迭代路径构造提高算法(IECI)^[11]、自适应大邻域搜索算法(ALNS)^[12]的对比如表 3 所示。表 3 中, N_v 为配送方案所用车辆总数, T_d 为配送方案的总行驶距离。从表 3 可以看出, 针对不同的速度时间依赖函数, 本文 ACO-ALNS 算法求得的车辆数与两种对比算法相同, 均为 10 辆; 在车辆行驶距离上, 本文算法的求解结果均优于 IECI 算法, 平均改善比为 3.66%; 与 ALNS 相比, 本文算法

仅在 TD2a、TD3b、TD1c、TD1d 和 TD3d 这 5 种时间依赖函数下求解得到的行驶距离略长, 但平均偏差小于 1%, 在其余 7 种时间依赖函数下, 求解结果的平均改善比为 0.64%。由此表明: 相较 IECI 和 ALNS 两种算法, 本文的 ACO-ALNS 算法具有优异性能, 能够较好地求解 TDVRPTW 问题。

表 3 TDVRPTW 算例对比结果
Table 3 Comparison results of TDVRPTW instances

时间依赖函数	IECI ^[11]		ALNS ^[12]		ACO-ALNS	
	N_v	T_d	N_v	T_d	N_v	T_d
TD1a	10	879	10	839	10	839
TD2a	10	864	10	834	10	840
TD3a	10	880	10	862	10	850
TD1b	10	892	10	838	10	833
TD2b	10	893	10	850	10	843
TD3b	10	866	10	828	10	839
TD1c	10	865	10	829	10	840
TD2c	10	863	10	837	10	837
TD3c	10	862	10	854	10	840
TD1d	10	872	10	831	10	836
TD2d	10	856	10	840	10	840
TD3d	10	867	10	832	10	838

(2) VRSPDTW 算例

VRSPDTW 算例来自文献 [4], 该算例共有 65 个, 包含客户规模为 10、25 和 50 的各 3 个, 以及客户规模为 100 的 56 个, 涉及 6 种类型(C1、C2、R1、R2、RC1、RC2), 每种类型具有不同的特点: C1 客户点位置为聚类分布, 时间窗较窄, 车辆最大容量偏小; C2 客户点位置聚类分布, 时间窗较宽, 车辆最大容量偏大; R1 客户点位置随机分布, 时间窗较窄, 车辆最大容量偏小; R2 客户点位置随机分布, 时间窗较宽, 车辆最大容量偏大; RC1 客户点位置混合聚类随机分布, 时间窗较窄, 车辆最大容量偏小; RC2 客户点位置混合聚类随机分布, 时间窗较宽, 车辆最大容量偏大。考虑不同客户规模及类型, 本文从中选取 27 个算例进行实验。将目标函数设置为最小化车辆使用数(主要目标)和行驶距离(次要目标), 运用提出的 ACO-ALNS 混合启发式算法独立求解 10 次, 最优实验结果与商业求解器 CPLEX^[4]、协同进化遗传算法(CO-GA)^[4]、并行模拟退火算法(P-SA)^[13]、离散布谷鸟算法(DCS)^[14]及带破坏和修复策略的蚁群算法(ACO-DR)^[15]进行对比。对比结果如表 4 所示。

表4 VRPSPDTW算例对比结果
Table 4 Comparison results of VRPSPDTW instances

算例	客户数	CPLEX ^[4]		CO-GA ^[4]		P-SA ^[13]		DCS ^[14]		ACO-DR ^[15]		ACO-ALNS	
		N_v	T_D	N_v	T_D	N_v	T_D	N_v	T_D	N_v	T_D	N_v	T_D
RCdp1001	10	3	348.98	3	348.98	3	348.98	3	348.98			3	348.98
RCdp1004	10	2	216.69	2	216.69	2	216.69	2	216.69			2	216.69
RCdp1007	10	2	310.81	2	310.81	2	310.81	2	310.81			2	310.81
RCdp2501	25	5	551.05	5	551.05	5	552.21	5	551.05			5	551.05
RCdp2504	25	7*	738.32*	4	473.46	4	473.46	4	473.46			4	473.46
RCdp2507	25	7*	634.20*	5	540.87	5	540.87	5	540.87			5	540.87
RCdp5001	50	9	994.18	9	994.18	9	994.70	9	994.18			9	994.18
RCdp5004	50	14*	1961.53*	6	725.59	6	725.90	6	725.59			6	733.21
RCdp5007	50	13*	1814.53*	7	809.72	7	810.04	7	809.72			7	809.72
Cdp101	100			11	1001.97	11	992.88	11	998.29	11	970.30	11	997.78
Cdp103	100			10	897.65	10	958.66	10	923.05	10	913.46	10	928.06
Cdp105	100			11	983.10	11	989.86	11	981.45	11	989.59	11	983.10
Cdp201	100			3	591.56	3	591.56	3	591.56	3	591.56	3	591.56
Cdp204	100			3	590.60	3	594.07	3	590.60	3	590.60	3	590.60
Cdp208	100			3	588.32	3	599.32	3	588.32	3	588.32	3	588.32
Rdp101	100			19	1653.53	19	1660.98	19	1658.65	20	1650.06	19	1660.46
Rdp103	100			14	1216.16	14	1226.77	14	1228.48	14	1227.62	13	1331.48
Rdp112	100			10	974.03	10	966.06	10	977.59	10	1025.84	10	986.30
Rdp201	100			4	1280.44	4	1286.55	4	1281.63	5	1236.13	4	1254.72
Rdp204	100			3	775.23	2	848.01	3	776.00	3	771.89	2	862.37
Rdp209	100			3	930.26	3	947.80	3	930.26	4	894.88	3	939.25
RCdp101	100			15	1652.90	15	1659.59	15	1654.32	15	1658.58	15	1669.91
RCdp104	100			11	1188.49	10	1268.43	10	1269.31	11	1183.80	11	1192.78
RCdp105	100			14	1581.26	14	1581.54	14	1581.26	14	1569.55	14	1573.60
RCdp201	100			4	1587.92	4	1513.72	4	1520.56	4	1383.79	4	1417.59
RCdp202	100			4	1211.12	4	1273.26	4	1242.92	4	1197.36	4	1161.79
RCdp208	100			3	878.57	3	862.89	3	862.89	3	862.66	3	853.09

注：*表示超出内存值。

根据表4的结果,不难看出,针对客户数为10、25和50的9个VRPSPDTW算例,商业求解器CPLEX仅能求解5个,其余4个因超出内存值提前终止,而本文的ACO-ALNS算法均能找到与之相同或更优的解;另外,本文的ACO-ALNS算法仅RCdp5004算例在行驶距离上与CO-GA和DCS存在1.05%的偏差,其余8个算例的车辆数及车辆行驶距离与CO-GA和DCS完全相同,而P-SA只有6个一致。其次,在客户数为100的18个算例中,本文的ACO-ALNS算法相较其他算法均取得不同程度的改善。具体而言:与CO-GA相比,本文算法均取得与其相同或更少的车辆数,在车辆数相同的16个算例中,有10个的行驶距离相同或更优,行驶距离最高可减少10.73%;与P-SA相比,本文算法共有13个算例取得与其相同或更好的解,行驶距离最高可减少8.75%;而相比于DCS,本文算法减少

了2个算例的车辆数,15个算例在车辆数相同的情况下,有9个得到了相同或更优的行驶距离,行驶距离最高可减少6.77%;与ACO-DR相比,本文算法可取得5个更优的车辆数,在余下车辆数相同的13个算例中,有7个的行驶距离相同或更优,行驶距离最高可减少3.85%。最后,从整体来看,本文的ACO-ALNS算法取得与其他4个算法相同或更好解的算例有15个,而CO-GA、P-SA、DCS和ACO-DR(未给出客户数为10、25和50的算例结果)分别为16、10、14、6个,由此可充分验证本文算法的有效性。

3.3 TDVRPSPDTW算例分析

选取部分VRPSPDTW算例,根据本文研究问题的特点,将配送中心的开放时间 $[a_0, b_0]$ 对应 $[0, 240]$ 均分成5个时间段,即 $[0, 0.2b_0)$ 、 $[0.2b_0, 0.4b_0)$ 、

$[0.4b_0, 0.6b_0]$ 、 $[0.6b_0, 0.8b_0]$ 及 $[0.8b_0, b_0]$, 并使用文献 [11] 中的 6 个速度时间依赖函数: TD1a=[1.00 1.60 1.05 1.60 1.00]、TD2a=[1.00 2.00 1.50 2.00 1.00]、TD3a=[1.00 2.50 1.75 2.50 1.00]、TD1b=[1.60 1.00 1.05 1.00 1.60]、TD1c=[1.60 1.60 1.05 1.00 1.00] 及 TD1d=[1.00 1.00 1.05 1.60 1.60], 以此生成 TDVRPSPDTW 算例, 进行算法性能比较、时变路网特性及不同目标函数分析。

3.3.1 算法性能比较

在 TD1a 时间依赖函数下, 利用模拟退火算法

(SA)^[16]、带局部搜索的蚁群算法 (Ant Colony Optimization with Local Search, ACO-LS) 与 ACO-ALNS 算法求解 TDVRPSPDTW 算例, 进一步分析本文算法性能。其中, SA 算法参数与文献 [16] 相同, ACO-LS 算法在保持蚂蚁数量、状态转移相关参数及迭代次数等与本文一致的情况下, 将 ALNS 替换为插入、交换及 2-opt 3 种邻域结构混合的局部搜索策略。针对同一算例, 不同算法均独立运行 10 次, 结果如表 5 所示, 表中, F_{best} 和 F_{ave} 为总配送成本的最优值和平均值, SD 为标准差。

表 5 TD1a 时间依赖函数下不同算法求解 TDVRPSPDTW 算例的结果对比

Table 5 Comparison of results for solving TDVRPSPDTW instances with different algorithms under TD1a time-dependent function

算例	客户数	SA			ACO-LS			ACO-ALNS		
		F_{best}	F_{ave}	SD	F_{best}	F_{ave}	SD	F_{best}	F_{ave}	SD
RCdp1001	10	1427.77	1427.77	0.00	1427.77	1427.77	0.00	1427.77	1427.77	0.00
RCdp1004	10	921.22	922.85	1.72	924.48	924.48	0.00	921.22	921.22	0.00
RCdp1007	10	1258.87	1258.87	0.00	1258.87	1258.87	0.00	1258.87	1258.87	0.00
RCdp2501	25	2645.84	2709.21	90.93	2634.79	2685.08	34.00	2634.79	2634.79	0.00
RCdp2504	25	2303.77	2382.48	148.48	2305.52	2323.50	15.98	2294.98	2294.98	0.00
RCdp2507	25	2623.14	2629.56	10.57	2611.10	2613.94	6.08	2611.10	2611.10	0.00
RCdp5001	50	5273.33	5763.03	328.75	5032.88	5363.74	212.53	4958.39	4963.05	4.42
RCdp5004	50	4026.20	4150.60	160.59	3668.39	3757.69	109.79	3427.59	3601.51	126.57
RCdp5007	50	4446.92	4620.81	177.50	4137.84	4292.62	134.45	4060.01	4070.56	8.71
RCdp101	100	10895.23	11254.04	310.19	9250.62	9504.04	158.39	8652.11	8832.39	136.70
RCdp104	100	7727.60	8337.04	476.21	6813.76	7129.40	179.59	6165.75	6411.87	148.98
RCdp107	100	7947.31	8501.43	404.52	6989.28	7241.50	150.98	6425.50	6621.25	143.14

针对客户规模从 10, 25, 50, 100 变化的 12 个 TDVRPSPDTW 算例, 本文的 ACO-ALNS 算法求得的最优值和平均值均小于或等于 SA 和 ACO-LS 算法, 其中, 最优值的平均改善比分别为 7.56% 和 2.90%, 且该优势随客户数的增加, 愈发明显, 当客户数为 100 时, 最优值的最大改善比分别达 20.59% 和 9.51%。此外, 从标准差来看, 本文算法仅求解 Rcdp5004 算例时, 略大于 ACO-LS 算法, 其余算例的标准差均较小。由此充分证明, 本文算法可有效且稳定求解不同规模下的 TDVRPSPDTW 问题。

3.3.2 时变路网特性

RCdp5001 算例在时变路网与静态路网(固定速度)下配送方案的对比结果如表 6 所示, 其中, 静态路网的车速为对应时变路网的平均速度。表 6 中, I_T 为车辆总行驶时间(min), I_w 为车辆的总等待时间(min), F 为总配送成本(元), F_2 为驾驶员成本(元), F_3 为燃油消耗成本(元), F_4 为碳排放成本(元), 同一时间依赖函数下的第二行是对应静态路网下的结果。RCdp5001 算例在 TD2a 时间依赖函数下, 时变路网与对应静态路网的配送路径如图 4 所示。

表 6 RCdp5001 算例时变路网与对应静态路网的对比

Table 6 Comparison of RCdp5001 instance under time-varying and corresponding static road network

时间依赖函数	N_v	T_0	I_T	I_w	F	F_2	F_3	F_4
TD1a	11	1081.53	1333.71	304.38	4958.39	1069.04	1112.56	26.78
	11	1076.45	1338.87	326.77	4938.02	1082.82	1079.22	25.98
TD2a	9	934.07	1008.90	222.16	4127.85	865.53	988.52	23.80
	9	1035.91	1072.37	258.57	4220.44	915.47	1030.17	24.80
TD3a	8	955.26	890.18	143.98	3895.25	767.08	1101.65	26.52
	9	936.38	833.08	386.42	4089.41	859.75	956.64	23.03

可以看出,由于考虑车辆行驶速度的不同,在时变路网与静态路网下,所生成配送方案的配送成本与配送路径存在一定差异。具体地,RCdp5001算例在TD1a时间依赖函数下,时变路网总配送成本略高于对应静态路网,但在TD2a下,时变路网的配送方案在车辆行驶距离、行驶时间、等待时间以及各项成本均有不同程度的降低;当时间依赖函数

为TD3a时,考虑不同速度的时变路网生成的配送方案所需车辆数量较静态路网减少一辆,总配送成本也降低4.75%。因此,车辆行驶速度对配送方案的制定存在很大影响,但现实生活中,车辆并非一直匀速行驶,本文考虑速度时变特性所构建的TDVRPSPDTW模型与实际情况更加贴合。

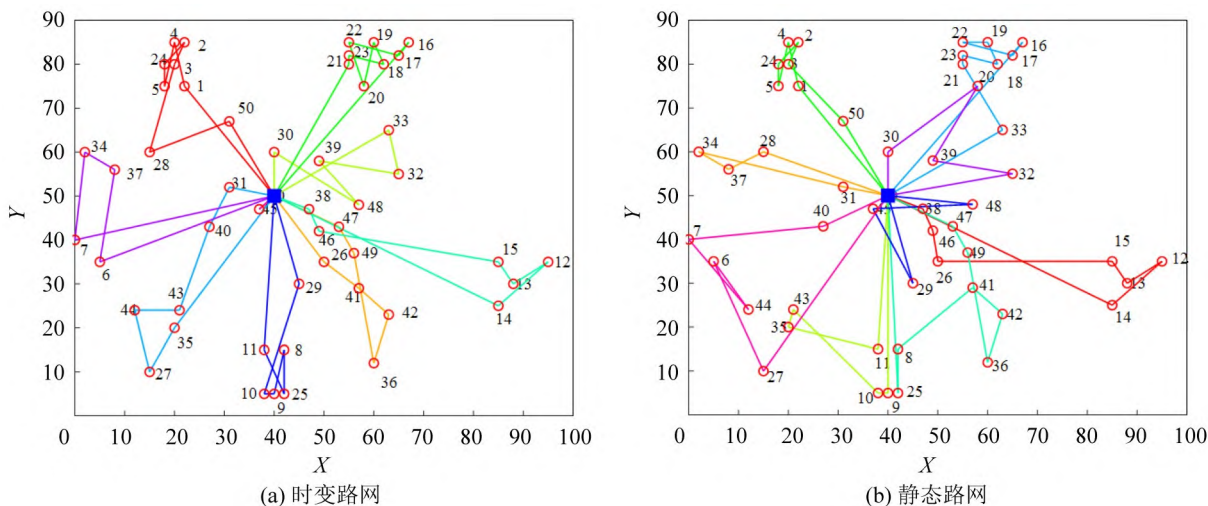


图4 RCdp5001算例在TD2a时变路网与对应静态路网下的配送路径

Fig. 4 Distribution routes for RCdp5001 instance under TD2a time-varying and corresponding static road network

在TD1a、TD2a及TD3a这3种时间依赖函数构建的时变路网下,9个算例的求解结果如表7所示。由表7可知,随着路网平峰车速的增加,车辆使用数量减少,但车辆使用数量的减少存在极限,例如,RCdp2501和RCdp2504算例在TD2a路网下行驶,车辆数较TD1a均减少一辆,但车速增加变为TD3a路网时,车辆数并未减少,原因在于,车速增加提高了配送车辆满足客户点时效要求的能力,单车在同等时间长度内,可访问更多客户点,但由于车辆容量的限制,单车可访问的客户点数量存在上限。反过来,较慢的车速会降低配送车辆满足客户时间窗要求的能力,单车可访问的客户点数量减少,车辆的容量不再是限制其配送多个客户点的关键因素,为保证配送时效,需要派遣更多车辆。此外,随着路网车速的提高,总配送成本呈现降低的趋势,其原因在于,车速的提高不仅可减少车辆的使用数量,还使单位时间内车辆的可行行驶距离增加,大幅减少车辆的行驶时间。

以上分析是基于同一情景下时变路网的速度变化展开的,为进一步探讨不同情景时变路网对于配送方案制定的影响,本文采用TD1a、TD1b、TD1c和TD1d这4种时间依赖函数代表的不同情景,对

RCdp5004算例进行求解,求解结果如表8所示。可以看出,时变路网高峰发生时段的变化对车辆的使用数量没有影响,但会导致车辆行驶距离、行驶时间、等待时间、总配送成本以及各项成本的明显波动。其中,当高峰发生在配送中心开放时间的第1和第5时间段(时间依赖函数为TD1a)时,得到的配送方案除等待时间稍短外,其余各项指标均较高,而总成本最低的配送方案出现在TD1b情境下,即车辆在配送中心开放的第2和第4时间段内遭遇高峰。由此表明,物流企业制定配送方案时不仅需考虑车辆的行驶速度变化,还需结合当地城市的交通特征。

3.3.3 不同目标函数

为衡量不同目标函数对配送方案决策的影响,除本文的优化目标函数 Z_1 外,分别考虑最小化碳排放量 A_e 的优化目标函数 Z_2 和最小化配送时间 A_s (所有车辆的行驶时间、等待时间及服务时间之和)的优化目标函数 Z_3 ,在保持算法与模型参数等其他条件一致的情况下,求解RCdp5001、RCdp5001和RCdp5007算例在TD1a时变路网下的配送方案。结果如表9所示。

表7 不同时间依赖函数下TDVRPSPDTW算例求解结果

Table 7 Results of TDVRPSPDTW instances under various time-dependent functions

算例	时间依赖函数	N_v	T_D	I_T	F	F_2	F_3	F_4
RCdp1001	TD1a	3	348.98	436.41	1427.77	310.01	359.11	8.65
	TD2a	3	341.88	365.52	1400.54	281.44	360.42	8.68
	TD3a	3	314.75	298.75	1392.69	275.26	358.79	8.64
RCdp1004	TD1a	2	218.68	268.33	921.22	191.16	224.66	5.41
	TD2a	2	218.68	223.63	911.84	175.40	230.88	5.56
	TD3a	2	218.68	193.70	922.41	163.73	252.60	6.08
RCdp1007	TD1a	3	243.17	299.96	1258.87	253.03	249.83	6.01
	TD2a	3	243.17	258.03	1245.25	234.12	254.99	6.14
	TD3a	2	254.12	243.95	1009.23	213.27	289.00	6.96
RCdp2501	TD1a	6	551.70	705.79	2634.79	551.04	570.03	13.72
	TD2a	5	540.44	587.94	2302.46	468.05	570.67	13.74
	TD3a	5	542.44	527.02	2324.76	441.33	618.54	14.89
RCdp2504	TD1a	5	533.17	669.27	2294.98	482.10	549.64	13.23
	TD2a	4	515.08	556.07	1962.65	406.26	543.31	13.08
	TD3a	4	501.14	460.94	1958.02	370.37	573.84	13.81
RCdp2507	TD1a	6	567.10	700.21	2611.10	513.37	583.68	14.05
	TD2a	5	558.56	588.39	2293.30	437.35	591.61	14.24
	TD3a	4	530.55	487.42	2008.08	378.60	614.68	14.80
RCdp5001	TD1a	11	1081.53	1333.71	4958.39	1069.04	1112.56	26.78
	TD2a	9	934.07	1008.90	4127.85	865.53	988.52	23.80
	TD3a	8	955.26	890.18	3895.25	767.08	1101.65	26.52
RCdp5004	TD1a	7	829.55	1054.83	3427.59	798.20	858.72	20.67
	TD2a	7	758.01	808.53	3275.35	699.36	806.57	19.42
	TD3a	6	733.84	702.86	2982.39	615.28	846.72	20.38
RCdp5007	TD1a	9	924.76	1138.31	4060.01	835.26	951.84	22.91
	TD2a	8	856.62	895.75	3648.09	718.83	907.42	21.85
	TD3a	7	787.75	724.34	3325.43	640.21	913.23	21.99

表8 不同情景时变路网RCdp5004算例的求解结果

Table 8 Results for RCdp5004 instance under various scenarios with time-varying road networks

时间依赖函数	N_v	T_D	I_T	I_w	F	F_2	F_3	F_4
TD1a	7	829.55	1054.83	41.57	3427.59	798.20	858.72	20.67
TD1b	7	801.58	988.26	13.12	3346.78	750.69	826.20	19.89
TD1c	7	826.46	1039.88	1.81	3394.83	770.35	853.93	20.56
TD1d	7	794.81	999.72	54.97	3368.75	777.34	821.63	19.78

表9 不同优化目标的求解结果

Table 9 Results of solution for various optimization objectives

优化目标	Z_1			Z_2			Z_3		
	F	A_E	A_S	F	A_E	A_S	F	A_E	A_S
RCdp5001	4958.39	535.67	2138.08	5234.93	526.85	2228.73	5063.62	589.82	2118.23
RCdp5004	3427.59	413.45	1596.40	3662.33	397.84	1632.30	3670.21	422.42	1543.50
RCdp5007	4060.01	458.28	1670.50	4119.63	456.87	1795.78	4094.69	473.40	1661.59

可以看出,针对每个算例,本文算法在不同的优化目标下均能求得最佳值,说明其对于考虑不同优化目标的TDVRPSPDTW是可行且有效的;其次,当使用单一目标作为目标函数时,碳排放量或配送时间达到最优,但总配送成本与本文所构建的模型相比显著增加,平均增加百分比分别达4.38%

和3.18%,表明较少碳排放量或较短配送时间的获得需牺牲一定的经济成本,而本文所构建的模型综合考虑了车辆固定成本、驾驶员时间成本、燃油消耗及碳排放成本等多种因素,可最大限度地降低总配送成本,提高物流企业的经济效益。

4 结论

本文得到的主要结论如下。

(1) 算法有效性验证的实验结果表明: 针对 TDVRPTW 算例, 相较于迭代路径构造算法和自适应大邻域搜索算法, 本文算法在车辆数完全相同的情况下, 车辆行驶距离的平均改善比分别达到 3.66% 和 0.64%; 针对 VRPSPDTW 算例, 本文算法与 CPLEX 求解器相比, 均可找到与之相同或更优的解, 而相较于协同进化遗传算法、并行模拟退火算法、离散布谷鸟算法及带破坏和修复策略的蚁群算法, 本文算法求解得到的车辆行驶距离最高可分别减少 10.73%、8.75%、6.77% 及 3.85%。由此表明, 本文提出的混合蚁群算法具有良好的寻优效果, 可有效解决 TDVRPTW 和 VRPSPDTW。

(2) TDVRPSPDTW 算例分析的实验结果表明: 针对不同规模的 TDVRPSPDTW 算例, 与模拟退火算法和带局部搜索的蚁群算法相比, 本文提出的混合蚁群算法求解得到的总配送成本最优值平均分别降低 7.56% 和 2.90%; 相比于仅考虑碳排放或配送时间的模型, 本文所构建的模型综合多种因素, 总配送成本平均分别降低 4.38% 和 3.18%, 同时, 在同一情景下, 随着路网平峰车速的提高, 总配送成本呈现降低的趋势, 但路网情景的变化会影响配送方案的制定, 因此, 物流企业应综合考虑车辆的行驶速度及当地城市的交通特征, 因地制宜地规划车辆的配送路径。本文提出的模型与算法可以为物流企业解决时变路网下考虑同时取送货的车辆路径问题提供借鉴与参考。

参考文献

- [1] ZHOU S, ZHANG D, JI B, et al. A MILP model and heuristic method for the time-dependent electric vehicle routing and scheduling problem with time windows[J]. *Journal of Cleaner Production*, 2024, 434: 140188.
- [2] 周鲜成, 刘长石, 周开军, 等. 时间依赖型绿色车辆路径模型及改进蚁群算法[J]. *管理科学学报*, 2019, 22(5): 57-68. [ZHOU X C, LIU C S, ZHOU K J, et al. Improved ant colony algorithm and modelling of time-dependent green vehicle routing problem[J]. *Journal of Management Sciences in China*, 2019, 22(5): 57-68.]
- [3] 珠兰, 马潇, 刘卓凡. 时间依赖型绿色车辆路径问题研究[J]. *交通运输系统工程与信息*, 2021, 21(6): 187-194. [ZHU L, MA X, LIU Z F. Time-dependent green vehicle routing problem[J]. *Journal of Transportation Systems Engineering and Information Technology*, 2021, 21(6): 187-194.]
- [4] WANG H F, CHEN Y Y. A genetic algorithm for the simultaneous delivery and pickup problems with time window[J]. *Computers & Industrial Engineering*, 2012, 62(1): 84-95.
- [5] LIU F, WANG L, GUI M, et al. A hybrid heuristic algorithm for urban distribution with simultaneous pickup-delivery and time window[J]. *Journal of Heuristics*, 2023, 29(2): 269-311.
- [6] 辛禹辰, 李润超, 杨华龙. 基于等待策略的同时送取货车车辆路径问题干扰管理研究[J]. *交通运输系统工程与信息*, 2023, 23(5): 155-161. [XIN Y C, LI R C, YANG H L. Disturbance management of VRPSDP based on waiting strategy[J]. *Journal of Transportation Systems Engineering and Information Technology*, 2023, 23(5): 155-161.]
- [7] ZHOU J, LI H, GU Y J, et al. A novel two-phase approach for the bi-objective simultaneous delivery and pickup problem with fuzzy pickup demands[J]. *International Journal of Production Economics*, 2021, 234: 108057.
- [8] ZHOU G, LI D, BIAN J, et al. Two-echelon time-dependent vehicle routing problem with simultaneous pickup and delivery and satellite synchronization[J]. *Computers & Operations Research*, 2024, 167: 106600.
- [9] ICHOUA S, GENDREAU M, POTVIN J. Vehicle dispatching with time-dependent travel times[J]. *European Journal of Operational Research*, 2003, 144(2): 379-396.
- [10] MAJIDI S, HOSSEINI MOTLAGH S M, IGNATIUS J. Adaptive large neighborhood search heuristic for pollution-routing problem with simultaneous pickup and delivery[J]. *Soft Computing*, 2018, 22(9): 2851-2865.
- [11] ANDRES FIGLIOZZI M. The time dependent vehicle routing problem with time windows: Benchmark problems, an efficient solution algorithm, and solution characteristics[J]. *Transportation Research Part E: Logistics and Transportation Review*, 2012, 48(3): 616-636.
- [12] 范厚明, 李荡, 孔靓, 等. 模糊需求下时间依赖型车辆路径优化[J]. *控制理论与应用*, 2020, 37(5): 950-960. [FAN H M, LI D, KONG L, et al. Optimization for time dependent vehicle routing problem with fuzzy demand[J]. *Control Theory & Applications*, 2020, 37(5): 950-960.]
- [13] WANG C, MU D, ZHAO F, et al. A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup-delivery and time windows[J]. *Computers & Industrial Engineering*, 2015, 83: 111-122.

下转第262页

- 11) [2024-03-04]. <http://zygh.xa.gov.cn/zwgk/ghjh/64a3d59af8fd1c1a703495a5.html>.]
- [14] PANG L, JIANG Y, WANG J, et al. Research of metro stations with varying patterns of ridership and their relationship with built environment, on the example of Tian**, China[J]. Sustainability, 2023, 15(12): 9533.
- [15] 贺鹏, 李雯茜, 李妍, 等. 建成环境对轨道交通客流非线性影响的空间特征[J]. 交通运输系统工程与信息, 2023, 23(3): 187-194. [HE P, LI W X, LI Y, et al. Spatial patterns of nonlinear effects of built environment on Beijing subway ridership[J]. Journal of Transportation Systems Engineering and Information Technology, 2023, 23(3): 187-194.]

上接第 242 页

- [14] 王超, 刘超, 穆东, 等. 基于离散布谷鸟算法求解带时间窗和同时取送货的车辆路径问题[J]. 计算机集成制造系统, 2018, 24(3): 570-582. [WANG C, LIU C, MU D, et al. VRPSPTW problem solving by discrete cuckoo search[J]. Computer Integrated Manufacturing Systems, 2018, 24(3): 570-582.]
- [15] WU H, GAO Y. An ant colony optimization based on local search for the vehicle routing problem with simultaneous pickup-delivery and time window[J]. Applied Soft Computing, 2023, 139: 110203.
- [16] CHEN J, LIAO W, YU C. Route optimization for cold chain logistics of front warehouses based on traffic congestion and carbon emission[J]. Computers & Industrial Engineering, 2021, 161: 107663.