



DOI:10.12404/j.issn.1671-1815.2403647

引用格式:蔡启文,马继东.基于BSO_ALNS算法的城市低碳物流路径优化[J].科学技术与工程,2025,25(9):3921-3930.

Cai Qiwen, Ma Jidong. Optimization of urban low-carbon logistics path based on BSO_ALNS algorithm[J]. Science Technology and Engineering, 2025, 25(9): 3921-3930.

基于BSO_ALNS算法的城市低碳物流路径优化

蔡启文, 马继东*

(东北林业大学机电工程学院, 哈尔滨 150040)

摘要 城市物流终端配送的路径规划是控制运输成本的关键。为解决城市低碳物流的路径规划问题,提出一种头脑风暴-自适应大邻域搜索算法(brain storm optimization_adaptive large neighborhood search,BSO_ALNS)进行求解。首先,建立以最小车辆运输总成本为优化目标的基于车辆油耗的带容量和时间窗约束的低碳车辆路径模型(capacitated vehicle routing problem with time windows,CVRPTW)。其次,利用头脑风暴算法(brain storm optimization,BSO)全局搜索,采用贪婪策略提高初始解质量,引入启发式交叉策略提升全局搜索质量;利用自适应大邻域搜索算法(adaptive large neighborhood search,ALNS)局部搜索,设计10种破坏和修复算子,引入自适应权重机制,结合模拟退火准则避免陷入局部最优。通过选取Solomon中不同规模的C、R、CR等类型实例对BSO_ALNS算法进行性能测试。以最短路径距离为目标,BSO_ALNS算法解与历史最优解误差均在1.5%内;以最小车辆运输总成本为目标,对比BSO和ALNS,BSO_ALNS均取得最优解。证明所提算法能够有效地解决城市低碳物流路径优化问题。

关键词 BSO_ALNS算法; CVRPTW; 低碳物流; 路径优化

中图分类号 U492.2; **文献标志码** A

Optimization of Urban Low-carbon Logistics Path Based on BSO_ALNS Algorithm

CAI Qi-wen, MA Ji-dong*

(College of Mechanical and Electrical Engineering, Northeast Forestry University, Harbin 150040, China)

[Abstract] The layout of the distribution path of urban logistics terminals is the key to controlling transportation costs. In order to solve the path planning problem of urban low-carbon logistics, a brainstorming-adaptive large neighborhood search algorithm (BSO_ALNS) was proposed. Firstly, a low-carbon vehicle path model with capacity and time window constraints based on vehicle fuel consumption (CVRPTW) was established with the optimization goal of minimum total vehicle transportation cost. Secondly, the brainstorming algorithm (BSO) was used to improve the quality of the initial solution, and the heuristic crossover strategy was introduced to improve the quality of the global search. Using the adaptive large neighborhood search (ALNS) local search, ten kinds of damage and repair operators were designed, and the adaptive weighting mechanism was introduced, combined with the simulated annealing criterion to avoid falling into the local optimum. The performance of the BSO_ALNS algorithm was tested by selecting C, R, CR and other types of instances of different scales in Solomon. Taking the shortest path distance as the goal, the error between the BSO_ALNS algorithm solution and the historical optimal solution is within 1.5%. With the goal of minimizing the total cost of vehicle transportation, the optimal solution is obtained BSO_ALNS compared with BSO and ALNS. It is proved that the proposed algorithm can effectively solve the problem of urban low-carbon logistics path optimization.

[Keywords] BSO_ALNS algorithm; CVRPTW; low-carbon logistics; path optimization

随着社会和技术快速发展,电子商务已经基本覆盖人们生活所需的各个方面,物流配送需求量相应逐年增加,随之产生的碳排放污染日益严峻。根据2024年4月北京市碳排放权电子交易平

台公布的《北京市生态环境局北京市统计局关于公布纳入北京市碳排放权交易管理的2023年度碳排放单位名单的通知》^[1],顺丰、中通等物流企业年度二氧化碳排放量达到了5000t(含)以上。物流

收稿日期:2024-05-16 修订日期:2024-12-29

基金项目:国家自然科学基金(31870537)

第一作者:蔡启文(1998—),男,满族,辽宁锦州人,硕士研究生。研究方向:计算机路径优化仿真。E-mail:c1156917763@163.com。

*通信作者:马继东(1972—),男,汉族,黑龙江哈尔滨人,博士,副教授,硕士研究生导师。研究方向:系统建模与仿真。E-mail:954601021@qq.com。

配送车辆的路径规划不仅与企业成本和客户满意度相关,还是碳排放量的重要影响因素。因此,考虑基于车辆油耗的带容量和时间窗约束的低碳 CVRPTW 具有重要意义。

CVRPTW 是车辆路径问题(vehicle routing problem, VRP)的一个分支,其主要研究的是在满足客户时间窗约束和车辆容量约束的前提下,求解针对不同目标的配送服务中的车辆路径规划和车辆使用问题。王迪^[2]等提出了使用贪婪鲸鱼优化算法求解以最小车辆总运输成本和最大客户满意度为优化目标的 VRPTW,提高了算法的全局寻优能力和收敛速度;夏小云等^[3]提出了使用自适应大邻域搜索的人工蜂群算法求解以最短行车路径为优化目标的 CVRP,更新了 Set A 算例中 4 个已知最优解;何美玲等^[4]提出了使用改进蚁群算法求解以最小车辆总运输成本为优化目标的 VRPTW,增强了算法综合性能,降低了配送成本;徐伟华等^[5]提出了使用改进遗传算法求解以最短行车路径为优化目标的 CVRP,提高了所求解的质量与稳定性;张凯月等^[6]提出了使用融合遗传算法的数字孪生技术求解以配送总成本最小为目标的 CVRP,避免不确定因素对运输成本的影响;Saksuriya 等^[7]提出了使用结合局部搜索的粒子群算法求解以带客户和服务商约束间的通用兼容匹配性为目标的 VRPTW,提高了算法求解速度和质量。通过对文献梳理可知,尽管目前针对 CVRPTW 相关问题有一定的相关研究,但在以下方面仍有完善空间:一方面,上述文献多以最短路径为目标函数;部分以最小车辆总运输成本为目标,但只是根据运输距离对成本进行求解,未考虑不同节点间车辆载重的变化对车辆油耗及碳排放成本的影响,与企业实际需求仍有差距。另一方面,在面对复杂的带容量和时间窗的低碳车辆路径优化问题时,应考虑多个算法相结合以增强算法寻优能力,进而获得较高质量解。

综上所述,现建立以车辆固定使用成本、车辆油耗成本、车辆碳排放成本及车辆等待时间惩罚成本构成的最小车辆总运输成本为优化目标的 CVRPTW 模型,提出一种基于自适应大邻域搜索算法的 BSO_ALNS 算法对模型进行求解。最后,通过选取 Solomon 中小、中、大等各个规模算例进行数值实验,从而证明该算法的有效性。

1 问题描述与基本假设

1.1 问题描述

CVRPTW 问题模型图如图 1 所示,存在一个配送中心和一组待服务客户节点,其地理位置、时间

窗及需求量已知。已知一组单一车型的物流配送车辆,物流配送车辆从配送中心出发,并在规定时间内返回配送中心,其间在每个客户服务窗内对该客户进行不重复服务,要求必须需满足客户及配送中心的时间窗和车辆容量约束,如果在左时间窗前到达客户节点,需等到时间窗才能开始配送,规划出最小车辆运输总成本的路径方案。

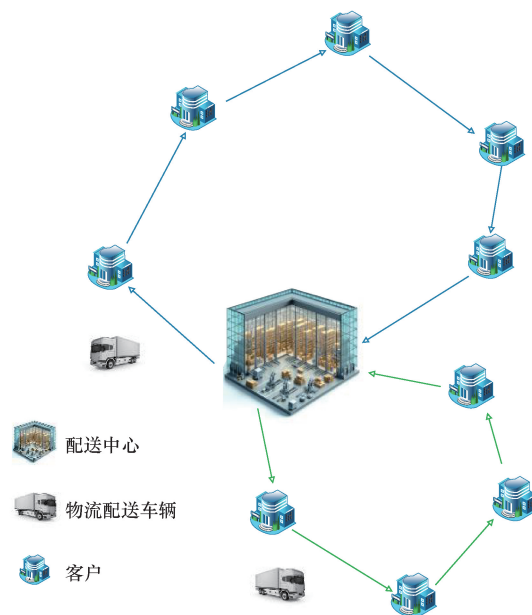


图 1 CVRPTW 模型图

Fig. 1 CVRPTW model

1.2 基本假设

(1) 假设已知各个客户节点的位置与所需货物的数量、服务时间窗及服务时间。

(2) 假设每辆物流配送车行驶速度恒定,且均为 60 km/h。

(3) 假设每辆物流配送车配送货物总量不得超过该车容量约束。

(4) 假设每个客户节点只能由一辆物流配送车服务一次。

(5) 假设物流配送车提前到达客户节点则需等待到达左时间窗才能开始服务,不能晚于右时间窗到达。

(6) 假设已知配送中心地理位置和时间窗,且物流配送车需不早于左时间窗在配送中心出发,不晚于右时间窗返回配送中心。

2 符号说明与模型建立

2.1 符号说明

G : 配送中心及客户节点集合 $G = \{0, 1, 2, \dots, n\}$, 其中 $G = \{0\}$ 为配送中心, $G = \{1, 2, \dots, n\}$ 为客户节点。

H :物流配送车车辆集合, $H = \{1, 2, \dots, h\}$ 。

α :物流配送车固定使用成本。

β :物流配送车单位油耗成本。

θ :物流配送车单位碳排放成本。

λ :物流配送车单位等待时间惩罚成本。

V :物流配送车行驶速度。

Q_h :物流配送车 h 最大装载容量。

P_{ih} :物流配送车 h 在客户节点 i 需配送货量。

i, j :物流中心或者客户节点,即为节点 i 和节点 j ,其中 $i, j \in G = \{0, 1, 2, \dots, n\}$ 。

d_{ij} :节点 i 和节点 j 之间的欧氏距离。

t_{ij} :从节点 i 到节点 j 行驶所需时间。

h_{ti} :物流配送车到达节点 i 时间。

h_{tr} :物流配送车离开节点 i 时间。

h_{bi} :物流配送车在节点 i 开始服务时间。

h_{tsi} :物流配送车在节点 i 服务时间。

L_i :节点 i 的左时间窗。

R_i :节点 i 的右时间窗。

CL:配送中心 $G = \{0\}$ 左时间窗。

CR:配送中心 $G = \{0\}$ 右时间窗。

CO_{phe} :物流配送车 h 空载状态下的油耗消耗率。

CO_{phf} :物流配送车 h 满载状态下的油耗消耗率。

CO_{ijhq} :物流配送车 h 在装载 q 千克货物状态下经过节点 i 和节点 j 间车辆燃油消耗量。

FE:表示燃油排放参数,通常将其取值为 2.621 kg/L。

CE_{ijhq} :物流配送车 h 在装载 q 千克货物经过节点 i 和节点 j 之间的车辆碳排放量。

x_{ijh} :0-1 变量,物流配送车 h 是否从客户节点 i 行驶至客户节点 j ,是则取值为 1;否则取值为 0。

y_{ih} :0-1 变量,物流配送车 h 是否配送客户节点 i 的货物,是则取值为 1;否则取值为 0。

2.2 油耗及碳排放量计算

目前城市物流配送工作仍多采用燃油商用车执行,碳排放产生难以避免。根据相关研究可知,碳排放量与车辆燃油消耗量呈正相关,而影响燃油消耗因素有很多,主要考虑车辆的载重负荷、行驶距离等因素。

车辆计算燃油消耗方法众多,采用“负载估计法”^[8]对油耗进行计算。物流配送车的载重负荷、行驶距离与燃油消耗表达式为

$$CO_{ijhq} = \left(CO_{phe} + \frac{CO_{phf} - CO_{phe}}{Q_h} P_h \right) d_{ij} \quad (1)$$

车辆碳排放量与油耗关系表达式为

$$CE_{ijhq} = CO_{ijhq} PE \quad (2)$$

2.3 模型构建

由于路线距离最短的路径方案不一定是车辆运输总成本最小的路径方案,因此基于问题描述、符号说明以及相关因素分析,建立以车辆固定使用成本、车辆油耗成本、车辆碳排放成本及车辆等待惩罚成本构成的最小车辆总运输成本为目标的带容量和时间窗约束的低碳车辆路径优化模型,具体模型如下。

$$\begin{aligned} \min Z = & H\alpha + \sum_{h \in H} \sum_{i, j \in G} d_{ij} x_{ijh} CO_{ijhq} \beta + \\ & \sum_{h \in H} \sum_{i, j \in G} d_{ij} x_{ijh} CE_{ijhq} \theta + \\ & \sum_{i \in G} \max(L_i - h_{ti}, 0) \lambda \end{aligned} \quad (3)$$

$$t_{ij} = \frac{d_{ij}}{V}, \forall (i, j) \in G \quad (4)$$

$$CL \leq h_{t0}, \forall h \in H \quad (5)$$

$$h_{t0} \leq CR, \forall h \in H \quad (6)$$

$$h_{ti} \leq R_i \sum_{j \in \Delta^+(i)} x_{ijh}, \forall i \in G, \forall h \in H \quad (7)$$

$$x_{ijh} = y_{jh}, \forall (i, j) \in G, \forall h \in H \quad (8)$$

$$L_i \sum_{j \in \Delta^+(i)} x_{ijh} \leq h_{tsi} \leq R_i \sum_{j \in \Delta^+(i)} x_{ijh}, \quad (9)$$

$$\forall i \in G, \forall h \in H$$

$$\sum_{i \in G} P_{ih} \leq Q_h, \forall h \in H \quad (10)$$

$$\sum_{h \in H} y_{ih} = 1, \forall i \in G \quad (11)$$

$$x_{ijh} \in \{0, 1\}, \forall (i, j) \in G, \forall h \in H \quad (12)$$

$$y_{ih} \in \{0, 1\}, \forall i \in G, \forall h \in H \quad (13)$$

$$\sum_{h \in H} \sum_{i \in G} x_{1ih} = \sum_{h \in H} \sum_{i \in G} x_{i1h} \quad (14)$$

上述模型中,式(3)为目标函数,表示以车辆的固定使用成本、油耗成本、碳排放成本和等待时间惩罚成本构成的最小车辆运输总成本;式(4)表示车辆从节点 i 和节点 j 之程所需时间;式(5)表示物流配送车 h 必须不早于配送中心左时间窗出发;式(6)表示物流配送车 h 必须不晚于配送中心右时间窗返回配送中心;式(7)表示物流配送车 h 到达客户节点 i 时间不晚于该节点右时间窗,其中 $\Delta^+(i)$ 表示从节点 i 出发的弧的集合;式(8)表示流量守恒,即到达和服务每个客户的车辆数相同;式(9)表示物流配送车 h 对客户节点 i 开始服务时间不超出该节点左右时间窗;式(10)表示物流配送车 h 的车辆货物装载数量不大于该车最大装载容量;式(11)表示每个客户节点货物只能由一辆车进行配送;式(12)、式(13)为 0-1 变量约束;式(14)表示车辆从配送中心出发,最终返回配送中心。

3 算法设计

3.1 算法描述

CVRPTW 属于 NP-hard 问题,一般使用启发式算法求解。头脑风暴算法^[9-10]作为一种全局寻优的智能优化算法,求解速度快且稳定性好,但容易陷入局部最优。自适应大邻域搜索算法^[11]通过使用自适应机制选择不同的破坏、修复算子提高搜索解空间能力,可有效避免算法迭代陷入局部最优,且具有较好局部寻优性和鲁棒性。

综上所述,将自适应大邻域搜索算法融入头脑风暴算法局部搜索策略中,生成 BSO_ALNS 算法,其中头脑风暴算法执行全局搜索,自适应大邻域搜索算法执行局部搜索。BSO_ALNS 算法需要输入算法相关参数、客户货物需求量及时间窗、车辆载重量及油耗参数和配送中心时间窗等数据,输出 CVRPTW 车辆配送方案和最小车辆运输总成本。在执行 BSO 进行全局搜索时,利用贪婪策略生成初始种群,利用聚类、启发式交叉等策略进行种群更新。在执行 ALNS 进行局部搜索时,利用设计的 5 种破坏算子和 5 种修复算子对更新后种群中的染色体进行邻域搜索,通过对比邻域解与原解的优劣对算子进行赋分,进而更新算子的权重,从而实现权重自适应。在 ALNS 中加入模拟退火接受准则,以一定概率接受较差解,避免在邻域搜索时陷入局部最优。ALNS 生成新的种群和最优解返回 BSO 主循环中,进行比较与迭代。当 BSO 全局搜索达到最大迭代次数或者生成超过设定次数的相同解,则循环停止。算法具体流程如图 2 所示。

3.2 编码解码与构造初始种群

3.2.1 编码解码

本文算法采用自然数编码的方式,将配送中心设为 0,对客户节点进行从 1 开始进行编号,将所有客户节点编码到一条染色体中(不包括配送中心),不同数字排列代表不同行车方案,确保每个客户只能由一辆车服务一次。假设所研究的 CVRPTW 中有 1 个配送中心和 9 个客户,则按上述编码规则可得染色体 {6,3,5,8,9,2,4,1,7}。

根据车载容量和时间窗约束进行解码^[12],首先以配送中心 0 为起点,将客户节点 6 置于一条新路径中的第一个访问节点,然后计算客户节点 3 是否满足车载容量和时间窗等约束条件,若满足约束,则将该点插入当前路径并计算后续 {5,8,9,2,4,1,7} 节点直至当前路径违反约束。若当前路径加入客户节点 8 违反约束,则将配送中心 0 置于客户节点 5 后,构成一条路径方案 {0,6,3,5,0}。随即构

建一条新配送路径,将客户节点 8 作为首个访问节点,而后重复上述解码规则计算剩余客户节点,最终得到 {0,6,3,5,0}, {0,8,9,2,4,0}, {0,1,7,0} 3 条路径方案。染色体编码及解码示例如图 3(a)和图 3(b)所示。

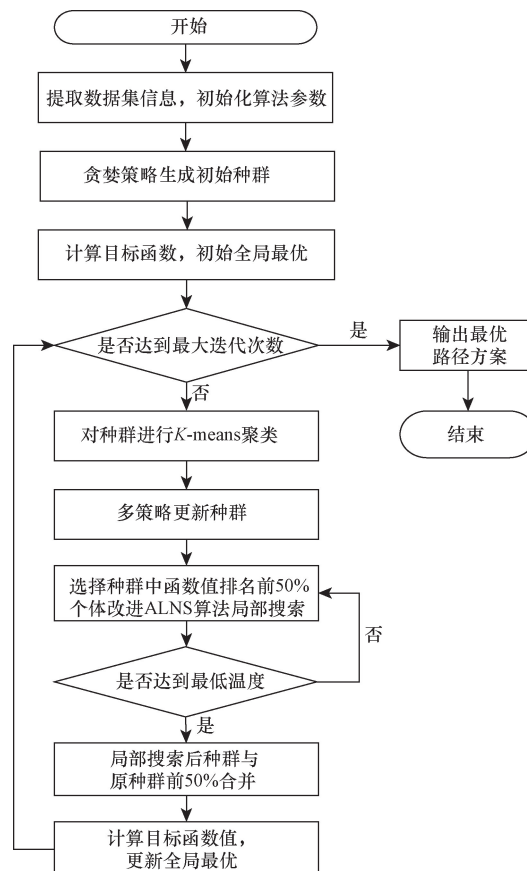


图 2 BSO_ALNS 流程图

Fig. 2 BSO_ALNS flow chart



图 3 染色体编码及解码

Fig. 3 Chromosome encoding and decoding

3.2.2 构造初始种群

对于以最小车辆运输总成本为目标的 CVRPTW 模型求解,引入车载容量和时间窗约束,采用贪婪策略提高初始种群质量,并能加快收敛速度。构造初始种群的步骤如下。

步骤 1 随机从 N 个客户节点中选出一个作为染色体的起始基因。

步骤 2 通过时间窗和容量约束从未在染色体中的客户节点中筛选出满足条件的可选择客户节点

群,分别计算选出从当前客户节点到各个可选客户节点所需的车辆运输总成本,选择车辆运输总成本最小的客户节点作为下一位基因,循环操作,直到所有客户节点都插入染色体。

步骤 3 重复步骤 1 和步骤 2 M 次,即可生成规模为 M 的初始种群,种群为 $M \times N$ 的矩阵。

3.3 聚类替换

3.3.1 聚类

计算每个染色体的适应度函数值,采用 K -means 聚类方法把 M 个染色体分成 K 个簇,将染色体储存到对应的簇中并按照适应度函数值排序,选出最佳适应度的染色体作为对应簇的聚类中心。

3.3.2 替换

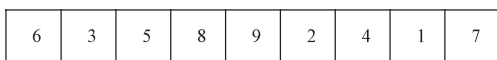
生成一个 $[0, 1]$ 的随机数 R_1 ,若 R_1 小于给定替换聚类中心概率 $p_{replace}$,则从 K 个聚类中心中随机选择一个聚类中心 Q ,并用一个新产生的随机解更新 Q 。

3.4 种群更新

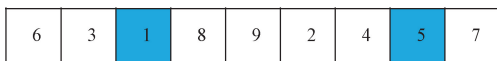
更新种群主要是以下两种生成新染色体的方法:①随机选出某一个聚类中心的个体进行 2-opt 操作;②选出某两个聚类中心的个体进行启发式交叉操作。一个 $[0, 1]$ 的随机数 q ,如果小于 0.5 则选择一个聚类中心,否则选择两个聚类中心。

3.4.1 单聚类中心 2-opt 操作

随机选择其中两个位置,将这两个位置上的元素进行交换。如图 4(a)和图 4(b)所示。已知染色体 635892417,交换第三位和第八位上的元素,则交换后的解为 631892457。



(a)交换前染色体



(b)交换后染色体

图 4 染色体交换

Fig. 4 Chromosome exchange

3.4.2 双聚类中心启发式交叉操作^[13-14]

生成一个 $[0, 1]$ 的随机数 p ,若 $p < 0.5$ 采取第一种交叉方式,否则采取第二种交叉方式。

1) 第一种交叉方式

实例如图 5 所示。首先,在第一个被选择的聚类中心的个体 $select_ind1$ 中随机选择一个位置 $l1$ 上的基因 $g1$;其次,找到第二个被选择的聚类中心的个体 $select_ind2$ 中 $l1$ 位置上的基因 $g2$,再回到个体 $select_ind1$ 中找到基因 $g2$ 所在的位置 $l3$;然后找到个体 $select_ind2$ 中位置上量 $l3$ 的元素 $g3$,重复此操作,直至形成一个闭环。如图 5(a)所示,以 $select$

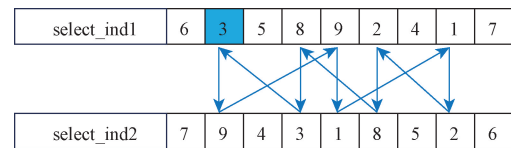
$_ind1$ 中 2 上的基因 3 为起点,进行寻找,最后形成闭环。

将选中的元素的位置保持不变,将个体 $select_ind1$ 和个体 $select_ind2$ 中未选择的元素互相交换位置生成两个新的个体 $child1$ 和 $child2$,如图 5(b)和图 5(c)所示。

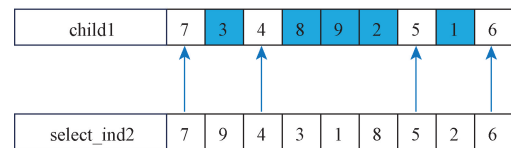
2) 第二种交叉方式

实例如图 6 所示。首先,在第一个被选择的聚类中心的个体 $select_ind1$ 中随机选择一组基因 $e1$;其次在第二个被选择的聚类中心个体 $select_ind2$ 中找到 $e1$ 中所有基因的位置;如图 6(a)所示。

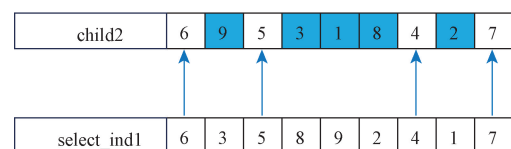
其次保持 $select_ind1$ 和 $select_ind2$ 中未选择的基因位置不变,将选择的基因按照出现顺序依次交换 $select_ind1$ 和 $select_ind2$ 中未选择基因的位置,生成两个新的个体 $child1$ 和 $child2$,如图 6(b)和图 6(c)所示。



(a)交叉位置确定



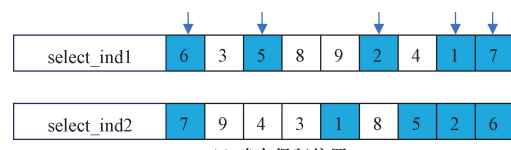
(b)交叉后新个体child1



(c)交叉后新个体child2

图 5 第一种交叉方式

Fig. 5 The first way of crossing



(a)确定保留位置



(c)交叉后新个体child2

图 6 第二种交叉方式

Fig. 6 The second way of crossing

3.5 局部搜索-自适应大领域算法

目标函数是最小车辆总运输成本,与车辆使用

数量、油耗量及等待时间相关。因此设计的大邻域搜索算子主要与上述3个方面相关,共设计5种“破坏”算子与5种“修复”算子。

3.5.1 破坏算子

1) 随机破坏算子

从当前染色体中随机选择 z 个客户节点移除,并将这 z 个客户节点放入待插入客户节点集合 Insert 中。

2) 相关破坏算子

相关破坏算子是由 Shaw^[15]提出的,其核心思想是找出相关性高的一组客户节点,将其放入待插入集合中并从当前路径方案中移除。具体计算流程为:从当前染色体中随机选择出客户节点 j ,分别计算染色体中其余 i 个客户节点与其相似度 $\text{sim}(i)$, sim 越大相关性越小。对所求 sim 进行升序排列,将最小即相关性最大的客户节点移除,重复上述操作 z 次,并将这 z 个客户节点放入待插入客户节点集合 Insert 中。

相关破坏算子主要与距离、货物需求量、时间窗以及服务时间相关,具体表达式如式(15)所示。

$$\begin{aligned} \text{sim}(i) = & \mu_1 d_{[j+1, n(i)+1]} + \mu_2 \text{abs}\{P_{j+1} - P_{[n(i)+1]}\} + \\ & \mu_3 \{\text{abs}[L_{j+1} - L_{n(i)+1}] + \\ & \text{abs}[R_{j+1} - R_{n(i)+1}]\} + \\ & \mu_4 \text{abs}[h_{\text{tsj}} - h_{\text{tsn}(i)}] \end{aligned} \quad (15)$$

式(15)中: μ_1 、 μ_2 、 μ_3 、 μ_4 分别为距离、货物需求量、时间窗以及服务时间的权重系数。

3) 最大成本破坏算子

对染色体进行解码,分别计算不同路径中删除每个客户节点前后的车辆总运输成本的差值,将差值最大的客户节点移除放入待插入客户节点集合 Insert 中,并更新染色体。最大成本移除算子主要与车辆的固定使用成本、油耗成本、碳排放成本和等待时间惩罚成本相关。

4) 整车破坏算子

将染色体解码成多条路径方案并计算相应车辆运输总成本,选出配送成本最大的车辆,将该车辆对应的所有客户节点移除并放入待插入客户节点集合 Insert 中。

5) 最大等待时间破坏算子

对染色体进行解码路径,分别计算不同路径中删除每个客户节点所节约的等待时间,将等待时间最大的客户节点移除放入待插入客户节点集合 Insert 中并更新染色体,重复上述操作 z 次。

3.5.2 修复算子

1) 最小碳排放插入算子

将染色体解码成路径方案。对集合 Insert 中

每个客户节点,寻找其符合约束的可插入位置;若所有现有路径方案均无法满足约束,则将该点放入新车进行服务。分别计算集合 Insert 中第 i 个客户节点插入不同位置所增加的碳排放成本,将该点插入到最小增量位置,同时在 Insert 中删除该点并更新染色体。重复上述操作,直至集合 Insert 成为空集。

2) 后悔值插入算子

后悔值是客户节点在次优插入位置和最优插入位置的成本差值。针对待插入客户节点集合 Insert 中每个节点,寻找其符合约束的可插入位置,计算所有节点后悔值。由于不同节点可能会有相同插入位置,则根据后悔值判断插入。循环迭代直至集合 Insert 中所有客户插入染色体中。

3) 最小等待时间插入算子

针对集合 Insert 中的每个客户节点,寻找符合约束条件的所有可插入位置,分别计算不同位置所增加的等待时间,选取等待时间增加最少的位置插入该客户节点,并更新染色体如果没有可插入位置,则新增车辆服务该点。重复上述操作,直至集合 Insert 中所有节点插入染色体。

4) 最小成本插入算子

在满足约束的基础上,寻找集合 Insert 中每个待插入客户节点的所有可插入位置。分别计算每个客户节点插入不同位置所增加的车辆运输总成本,选择成本增加最少的位置插入节点,并更新染色体。若所有现有路径方案均无法满足约束,则将点放入新路径中。针对集合 Insert 中所有节点循环上述操作。

5) 随机插入算子

针对所有待插入客户节点,寻找其符合约束的可插入位置,计算客户节点 i 插入不同位置增加的车辆运输总成本,在成本升序排列前50%的位置中随机选取一个位置插入,更新染色体。若所有现有路径方案均无法满足约束,则将点放入新车进行服务。将所有待插入客户节点循环上述操作插入染色体中形成新解。

3.5.3 接受准则和自适应机制

1) 接受准则

如果在迭代过程中只接受优解,则种群可能会被同质染色体占领,导致算法陷入局部最优。本文引入模拟退火接受准则^[16],对于在邻域搜索过程中产生的较差解,以概率 $e^{-\frac{\text{obj_new} - \text{obj}}{T_{\text{em}}}}$ 接受,其中 e 为自然常数, obj_new 为新解的目标函数值, obj 为当前解目标函数值。随着模拟退火温度 T_{em} 的下降,接受较差解概率也随之逐渐下降。

2) 自适应机制

提出了共 10 种算子,初始寻优各算子权重相同,利用轮盘赌算法分别选择破坏和修复算子。由于各算子寻优能力有所不同,采用自适应机制,根据不同算子探索解与原始解的质量对比情况为算子赋分,并更新算子权重。自适应机制可以使 BSO_ALNS 算法在局部搜索过程中,以更大几率选择探索解的能力的算子进行寻优。算子赋分表如表 1 所示,权重更新公式如式(16)所示。

$$\text{Insert_weight} = \alpha \text{Insert_weight} + (1 - \alpha) \frac{\text{Insert_scores}}{\text{Insert_times}} \quad (16)$$

式(16)中:Insert_weight 为算子权重; α 为权重系数;Insert_score 为算子累计得分;Insert_times 为该算子选择次数。

表 1 算子赋分表

Table 1 Operator assignment table

解的质量	赋分
新解 < 全局最优解	6
全局最优解 < 新解 < 原解	3
新解 = 原解	1
新解 < 原解	0

3.6 迭代终止条件

BSO_ALNS 算法设置两种迭代终止条件,只要满足两种条件之一即可终止算法迭代并输出最优车辆路径方案及最小车辆运输总成本。两种迭代终止条件如下:①实际总迭代次数达到设置最大迭代次数;②种群历代最优解连续不变的迭代次数达到预设的阈值 gen_num。

4 算例实验与数值分析

4.1 算例试验

4.1.1 算例选择

为验证所提出的 BSO_ALNS 算法求解 CVRPTW 的有效性,选择 Solomon 标准算例库^[17]中 C、R、RC 等类型大中小规模算例进行两个数值仿真实验。实验 1 使用本文算法求解 Solomon 标准算例 VPR 问题,通过与现有最优距离进行对比,验证本文算法的有效性。实验 2 分别对比 BSO、ALNS 和 BSO_ALNS 算法求解的最小车辆运输总成本,验证 BSO_ALNS 算法求得解的质量有所提升。

4.1.2 参数设置

BSO 算法参数设置如下:最大迭代次数 Maxgen = 150;种群规模 Popsiz = 100,选取聚类数目为 10 个;随机解替换概率 $p_{\text{replace}} = 0.1$;选择 1 个聚类概率 $p_{\text{one}} = 0.5$;选择 2 个聚类概率 $p_{\text{two}} = 0.5$;选择 1

个聚类中心概率 $p_{\text{one_center}} = 0.3$;选择 2 个聚类中心概率 $p_{\text{two_center}} = 0.3$ 。ALNS 算法参数设置如下:初始温度 $T_{\text{em}} = 500$;冷却系数 $\varphi = 0.98$;各算子初始权重均为 0.2。为满足 Solomon 标准算例约束,设车辆固定成本 $\alpha = 300$;物流配送车单位油耗成本 $\beta = 9$;物流配送车单位碳排放成本 $\theta = 0.5$;物流配送车单位等待时间惩罚成本 $\lambda = 1$ 。

4.2 数值分析

4.2.1 实验 1-最优值对比

选取 Solomon 标准算例中 C101、C102、C201、C202、R101、R102、R202、R203、RC101、RC106、RC201、RC204 等 12 个算例,分别对上述算例 25、50、100 的不同规模客户节点采用 BSO_ALNS 算法求解最小车辆路径距离,与各不同规模算例的历史最优解进行对比,数据如表 2 所示。

通过表 2 可知,BSO_ALNS 求解各不同规模算例的最短路径距离与历史最短路径距离进行比较,其误差均不超过 1.5%,说明本文算法的有效性。

4.2.2 实验 2-不同算法求解结果对比

选取 Solomon 标准算例中 100 规模客户节点的 C106、C206、R101、R202、RC106 和 RC204 等 6 个算例进行仿真实验,分别针对上述算例,采用 BSO、ALNS 以及 BSO_ALNS 算法求解以最小车辆总运输成本为目标的 CVRPTW。3 种算法求解的不同规模算例最小车辆总运输成本结果对比如表 3 所示。

通过表 3 可知,针对不同规模的算例,BSO_ALNS 算法求解成本均为 3 种算法中最优解。其中,与 BSO 算法相比较,在求解不同类型大、中、小规模算例时,BSO_ALNS 算法求解成本平均降低 10%;与 ALNS 算法相比较,在求解不同类型大规模算例时,BSO_ALNS 算法求解成本平均降低 3%。

以算例 C106-100、RC106-50、R202-25 为例,BSO、ALNS、BSO_ALNS 这 3 种算法求解以最小车辆运输总成本为目标的 CVRPTW 迭代变化趋势最小车辆运输总成本变化趋势对比如图 7 所示。BSO_ALNS 算法求解算例 C106-100、RC106-50、R202-25 最优路线图最优配送方案路径图如图 8 所示。

通过图 7 对比可知,针对不同类型以及不同规模的算例,相较于 BSO 和 ALNS 算法而言,BSO_ALNS 初始解质量大幅提高且均能求得最优解。对比 ALNS 算法,BSO_ALNS 算法收敛速度明显提升。对比 BSO 算法,BSO_ALNS 算法求解质量明显提升,求解成本降低,证明 BSO_ALNS 算法在求解 CVRPTW 问题上的有效性。

表2 BSO_ALNS 与最优解对比表
Table 2 Comparison table of BSO_ALNS and optimal solution

算例	客户数	头脑风暴_自适应大邻域搜索算法		历史最优解		误差率/%
		车辆数	行驶距离	车辆数	最优距离	
C101	25	3	191.813 6	3	191.3	0.27
	50	5	363.246 8	5	362.4	0.23
	100	10	828.936 9	10	827.3	0.20
C102	25	3	190.737 6	3	190.3	0.23
	50	5	362.170 8	5	361.4	0.21
	100	10	828.936 9	10	827.3	0.20
C201	25	2	215.542 6	2	214.7	0.39
	50	3	361.796 5	3	360.2	0.44
	100	3	591.556 6	3	589.1	0.42
C202	25	1	215.542 6	2	214.7	0.39
	50	3	361.796 5	3	360.2	0.44
	100	3	591.556 6	3	589.1	0.42
R101	25	8	618.329 9	8	617.1	0.20
	50	12	1 046.701 1	12	1 044.0	0.26
	100	19	1 651.611 4	20	1 637.7	0.85
R102	25	7	548.107 8	7	547.1	0.18
	50	11	911.442 6	11	909.0	0.27
	100	18	1 475.002 9	18	1 466.6	0.57
R202	25	4	411.487 4	4	410.5	0.24
	50	5	703.318 6	5	698.5	0.69
	100	7	1 042.700 5	8	1 029.6	1.27
R203	25	3	392.325 2	3	391.4	0.24
	50	5	610.747 3	5	605.3	0.90
	100	6	880.502 4	6	870.8	1.11
RC101	25	4	462.155 9	4	461.1	0.23
	50	8	945.576 8	8	944.0	0.17
	100	15	1 642.986 1	15	1 619.8	1.43
RC106	25	3	346.505 4	3	345.5	0.29
	50	6	724.653 1	6	723.2	0.20
	100	12	1 378.215 8	12	1 372.7	0.40
RC201	25	3	361.241 0	3	360.2	0.29
	50	5	686.311 6	5	684.8	0.22
	100	8	1 276.197 5	9	1 261.8	1.14
RC204	25	3	300.234 6	3	299.7	0.18
	50	3	444.966 2	3	444.2	0.17
	100	4	788.662 6	4	783.5	0.66

表3 BSO_ALNS 与 BSO、ALNS 对比表
Table 3 Comparison between BSO_ALNS and BSO、ALNS

算例	客户数	头脑风暴算法			自适应大邻域搜索算法			头脑风暴_自适应大邻域搜索算法		
		车辆数目	路径距离	运输成本	车辆数目	路径距离	运输成本	车辆数目	路径距离	运输成本
C106	25	3	258.761 1	1 310.239 8	3	258.761 1	1 310.239 8	3	258.095 8	1 308.945 2
	50	6	500.733 4	2 596.073 5	5	365.822 9	1 935.201 6	5	365.822 9	1 935.201 6
	100	11	1 346.907 7	4 991.238 8	10	843.469 6	3 953.449 0	10	828.936 9	3 938.481 2
C109	25	3	251.769 1	1 172.418 0	3	226.878 3	1 148.788 9	3	224.829 9	1 147.172 3
	50	5	400.039 4	1 955.849 9	5	368.240 0	1 915.467 9	5	363.246 8	1 910.185 9
	100	10	1 078.933 0	4 229.329 7	10	970.415 6	4 123.877 0	10	828.936 9	3 938.481 2
R101	25	8	699.042 3	3 390.345 2	8	705.649 8	3 377.575 5	8	705.649 8	3 377.575 5
	50	12	1 196.728 4	5 089.230 8	11	1 211.971 5	4 726.779 1	11	1 216.861 6	4 725.728 6
	100	20	1 927.350 5	8 458.239 2	20	1 950.317 3	8 435.957 2	19	1 853.227 4	8 040.439 7
R202	25	2	865.354 6	1 580.402 5	2	1 002.021 9	1 569.498 8	2	1 007.674 2	1 560.566 5
	50	3	1 249.868 2	2 110.978 5	2	1 127.920 7	1 662.168 8	2	1 091.298 9	1 640.497 8
	100	4	2 099.407 8	3 334.298 4	4	1 818.343 3	2 951.872 2	4	1 758.371 7	2 906.870 9
RC106	25	3	355.132 4	1 315.142 1	3	355.132 4	1 315.142 1	3	355.132 4	1 315.142 1
	50	6	742.129 9	2 633.732 9	6	741.855 8	2 629.956 0	6	741.265 6	2 628.194 2
	100	14	1 664.741 9	6 051.504 2	13	1 522.940 5	5 599.014 5	13	1 483.028 0	5 522.929 7
RC204	25	1	531.986 4	872.138 0	1	573.004 0	866.492 5	1	572.957 2	854.504 5
	50	2	678.099 9	1 399.928 7	2	737.461 7	1 343.882 1	2	737.461 7	1 343.882 1
	100	3	1 390.214 9	2 339.759 4	3	1 370.243 2	2 326.480 9	4	1 297.665 2	2 194.740 8

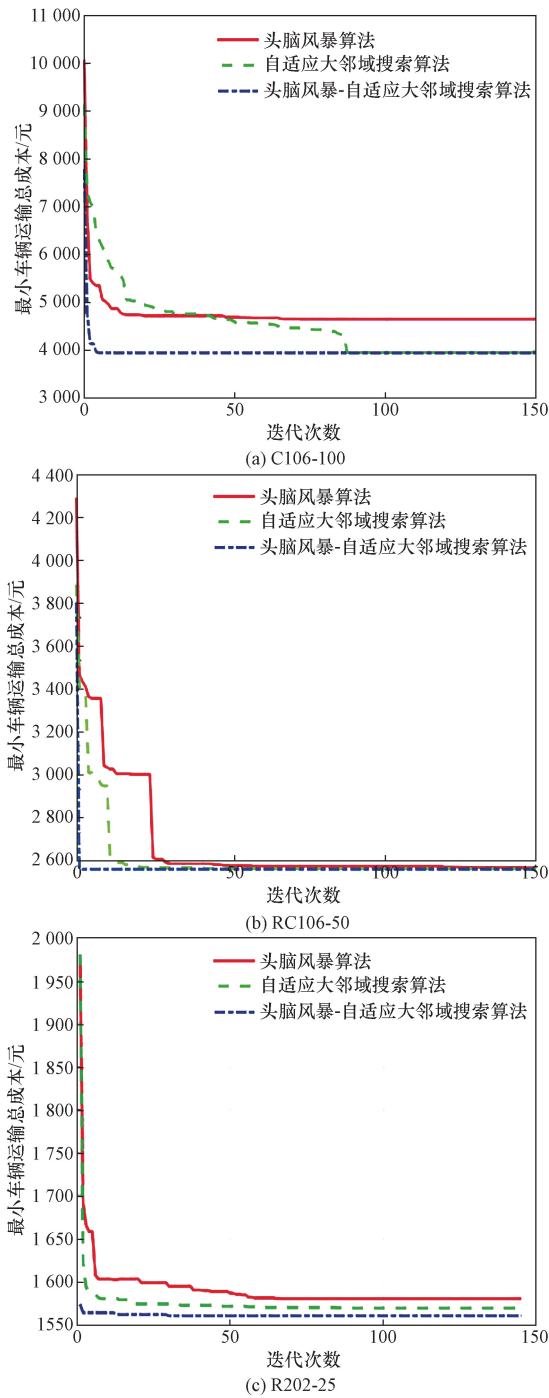


图7 最小车辆运输总成本变化趋势对比图
 Fig. 7 Comparison of the change trend of the total cost of minimum vehicle transportation

5 结论

针对考虑车辆油耗的带容量和硬时间窗约束的低碳车辆路径问题,综合考虑了车辆使用成本,车辆油耗成本,车辆碳排放成本以及等待时间惩罚成本,提出了以最小车辆运输总成本为目标的CVRPTW模型。针对经典BSO算法的不足,重新设计了基于贪婪策略、启发式交叉策略、邻域算子、自

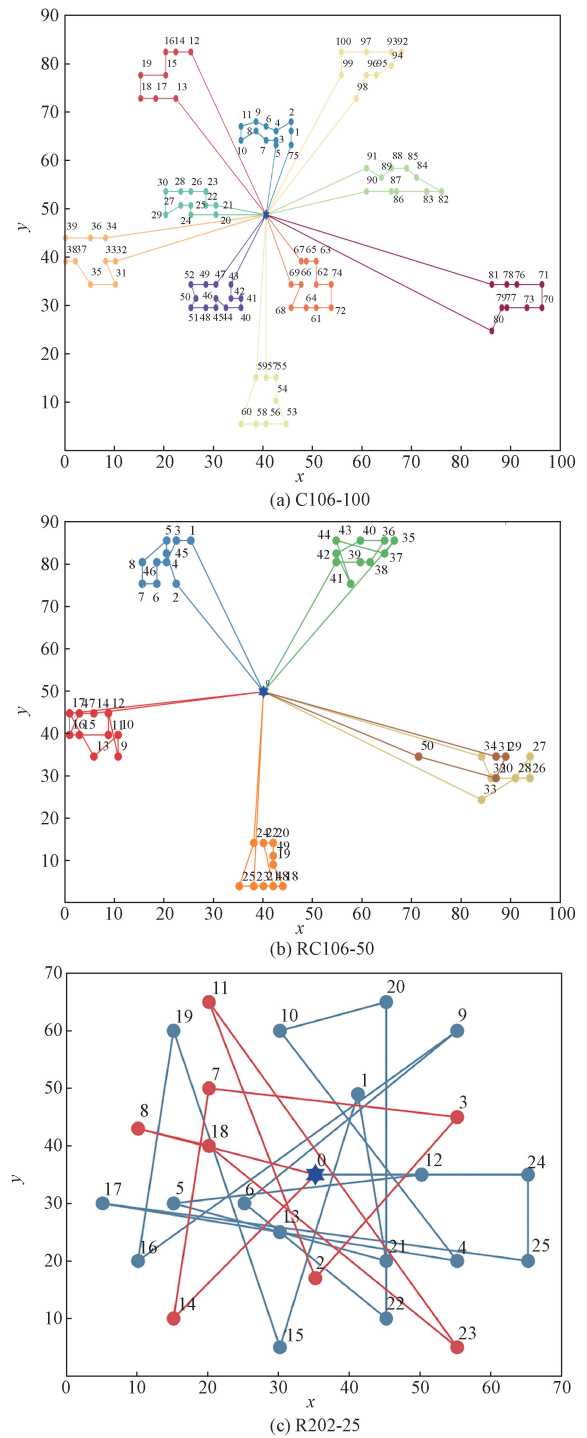


图8 最优配送方案路径图
 Fig. 8 Optimal delivery route diagram

适应机制及模拟退火等策略的BSO_ALNS算法进行求解。通过选取Solomon中不同类型和规模的标准算例进行试验,分析结果得出如下结论。

- (1)以最短运输距离为目标求得路径方案其相应运输成本未必最小,而本文所提出的CVRPTW模型在考虑降低企业物流配送成本的同时减少碳排放量;不仅符合企业实际需求,而且响应国家双碳战略。
- (2)在相同的Solomon数据集中测试:以最短路

径距离为目标,BSO_ALNS 求解不同规模算例的最短距离与历史最优解误差均不超过 1.5%。以最小车辆运输总成本为目标,BSO_ALNS 算法相较本文对比算法均能求得最优值,尤其针对大中规模算例,求解质量较对比算法提升 3%~10%,证明本文算法求解 CVRPTW 的有效性,有一定参考价值。

CVRPTW 问题只考虑了车辆载重因素对油耗的影响,在后续研究中,应考虑客户需求点动态变化、道路拥堵等更多因素的影响,进而设计更高效的算法。

参 考 文 献

- [1] 北京市生态环境局,北京市统计局. 关于公布纳入北京市碳排放权交易管理的 2023 年度碳排放单位名单的通知[EB/OL]. (2024-04-16) [2024-05-16]. <https://www.bjets.com.cn/article/zcfg/202404/20240400002615.shtml>.
Beijing Municipal Bureau of Ecological Environment, Beijing Municipal Bureau of Statistics. Notice on the announcement of the list of carbon emission units in 2023 that will be included in the management of carbon emission trading in Beijing[EB/OL]. (2024-04-16) [2024-05-16]. <https://www.bjets.com.cn/article/zcfg/202404/20240400002615.shtml>.
- [2] 王迪,金辉. 贪婪鲸鱼优化算法求解带时间窗的快速末端配送路径问题[J]. 计算机应用与软件, 2020, 37(6): 263-268, 315.
Wang Di, Jin Hui. Greedy whale optimization algorithm solves the problem of terminal distribution path of express with time window[J]. Computer Applications & Software, 2020, 37(6): 263-268, 315.
- [3] 夏小云,庄鹤林,杨火根,等. 自适应大邻域搜索的人工蜂群算法求解带容量约束车辆路径问题[J]. 计算机集成制造系统, 2022, 28(11): 3545-3557.
Xia Xiaoyun, Zhuang Helin, Yang Huogen, et al. Artificial bee colony algorithm for adaptive large neighborhood search to solve vehicle routing problem with capacity constraint[J]. Computer Integrated Manufacturing Systems, 2022, 28(11): 3545-3557.
- [4] 何美玲,魏志秀,武晓晖,等. 基于改进蚁群算法求解带软时间窗的车辆路径问题[J]. 计算机集成制造系统, 2023, 29(3): 1029-1039.
He Meiling, Wei Zhixiu, Wu Xiaohui, et al. Solving vehicle routing problem with soft time window based on improved ant colony algorithm[J]. Computer Integrated Manufacturing Systems, 2023, 29(3): 1029-1039.
- [5] 徐伟华,邱龙龙,张根瑞,等. 求解带容量约束车辆路径问题的改进遗传算法[J]. 计算机工程与设计, 2024, 45(3): 785-792.
Xu Weihua, Qiu Longlong, Zhang Genrui, et al. Improved genetic algorithm for solving vehicle routing problem with capacity constraint[J]. Computer Engineering and Design, 2024, 45(3): 785-792.
- [6] 张凯月,温海骏,陈跃鹏,等. 基于数字孪生技术的危化品车辆运输路径优化方法[J]. 科学技术与工程, 2023, 23(34): 14676-14681.
Zhang Kaiyue, Wen Haijun, Chen Yuepeng, et al. Vehicle transportation path optimization method for hazardous chemicals based on digital twin technology[J]. Science Technology and Engineering, 2023, 23(34): 14676-14681.
- [7] Saksuriya P, Likasiri C. Hybrid heuristic for vehicle routing problem with time windows and compatibility constraints in home healthcare system[J]. Applied Sciences, 2022, 2022, 12(13): 12136486.
- [8] 张金良,李超. 碳排放影响下的动态配送车辆路径优化研究[J]. 中国管理科学, 2022, 30(9): 184-194.
Zhang Jinliang, Li Chao. Research on route optimization of dynamic delivery vehicles under the influence of carbon emissions[J]. CMS, 2022, 30(9): 184-194.
- [9] Shi Y. Brain storm optimization algorithm[C]//International Conference in Swarm Intelligence. Berlin: Springer, 2011: 303-309.
- [10] 张凯月,温海骏,陈跃鹏,等. 新冠疫情期间应急物资车辆路径优化方法[J]. 科学技术与工程, 2023, 23(6): 2518-2525.
Zhang Kaiyue, Wen Haijun, Chen Yuepeng, et al. Vehicle route optimization method for emergency supplies during the new crown epidemic[J]. Science Technology and Engineering, 2023, 23(6): 2518-2525.
- [11] 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[J]. 工业工程与管理, 2021, 26(3): 115-122.
Xu Qian, Xiong Jun, Yang Zhenhua, et al. Routing optimization of takeaway delivery vehicle based on adaptive large neighborhood search algorithm[J]. Industrial Engineering and Management, 2021, 26(3): 115-122.
- [12] 马祥瑞,马良,张惠珍. 不确定车辆数的多约束车辆路径问题[J/OL]. 计算机工程与应用, 1-11 [2025-01-10]. <http://kns.cnki.net/kcms/detail/11.2127.tp.20240104.1625.046.html>.
Ma Xiangli, Ma Liang, Zhang Huizhen. Multi-constrained vehicle routing problem with uncertain number of vehicles[J/OL]. Computer Engineering and Applications, 1-11 [2025-01-10]. <http://kns.cnki.net/kcms/detail/11.2127.tp.20240104.1625.046.html>.
- [13] 路雪刚,张雪花,张梦桃. 基于改进鲸鱼优化算法的畜禽废弃物运输路径优化问题[J]. 科学技术与工程, 2022, 22(25): 11120-11129.
Lu Xuegang, Zhang Xuehua, Zhang Mengtao. Optimization of livestock and poultry waste transportation path based on improved whale optimization algorithm[J]. Science Technology and Engineering, 2022, 22(25): 11120-11129.
- [14] 周鲜成,李松明,王莉,等. 考虑非线性能耗的时间依赖型电动车辆路径模型及改进的鲸鱼优化算法[J/OL]. 中国管理科学, 1-15 [2025-01-10]. <https://doi.org/10.16381/j.cnki.issn1003-207x.2023.1103>.
Zhou Xiancheng, Li Songming, Wang Li, et al. Time-dependent electric vehicle path model considering nonlinear energy consumption and improved whale optimization algorithm[J/OL]. CMS, 1-15 [2025-01-10]. Chinese Journal of Management Science, <https://doi.org/10.16381/j.cnki.issn1003-207x.2023.1103>.
- [15] Shaw P. Using constraint programming and local search methods to solve vehicle routing problems[J]. Principles and Practice of Constraint Programming, 1998(1): 417-431.
- [16] 陈华根,吴健生,王家林,等. 模拟退火算法机理研究[J]. 同济大学学报(自然科学版), 2004(6): 802-805.
Chen Huagen, Wu Jiansheng, Wang Jialin, et al. Mechanism of simulated annealing algorithm[J]. Journal of Tongji University (Natural Science Edition), 2004(6): 802-805.
- [17] Capacitated Vehicle Routing Problem Library. VRPTW benchmarks[EB/OL]. (2022-04-08) [2024-05-16]. <http://vrp.atd-lab.inf.puc-rio.br/index.php/en/>.