



## 混合优化算法求解同时送取货车辆路径问题

李 珺<sup>+</sup>, 段钰蓉, 郝丽艳, 张维维

兰州交通大学 电子与信息工程学院, 兰州 730070

+ 通信作者 E-mail: lijane@mail.lzjtu.cn

**摘要:**为了给各物流企业在车辆配送路径规划方面提供合理有效的决策支持,针对单配送中心的配送模式,研究带时间窗约束的同时送取货车辆路径问题(VRPSDPTW),建立以总配送成本最小化为目标的数学模型。根据模型的特征,提出基于模拟退火(SA)与自适应大规模邻域搜索(ALNS)相结合的混合优化算法(SA-ALNS)。采用基于时间与距离加权的插入启发式算法构造问题的初始解;引入多种删除、插入算子,以自适应选择策略进行路径优化,并通过反馈机制,逐渐调整各操作算子被选择的概率,使算法更倾向于选择寻优效果较好的算子;使用模拟退火机制的Metropolis准则控制解的更新。仿真实验中测试了56个大规模算例,对比了p-SA算法、DCS算法和VNS-BSTS等其他智能优化算法并进行统计分析,结果证明该算法在求解带时间窗约束的同时送取货车辆路径问题的可行性和优越性,研究成果极大丰富了车辆路径问题(VRP)的相关研究。

**关键词:**车辆路径问题(VRP);同时送取货;模拟退火算法(SA);自适应大规模邻域搜索算法(ALNS);时间窗  
**文献标志码:**A **中图分类号:**TP301

## Hybrid Optimization Algorithm for Vehicle Routing Problem with Simultaneous Delivery-Pickup

LI Jun<sup>+</sup>, DUAN Yurong, HAO Liyan, ZHANG Weiwei

School of Electronic and Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China

**Abstract:** In order to provide reasonable and effective decision support for logistics enterprises in vehicle distribution path planning, this paper studies the vehicle routing problem with simultaneous delivery-pickup and time windows (VRPSDPTW) for single distribution center distribution mode, and establishes a mathematical model with the objective of minimizing the total distribution cost. According to the characteristics of the model, a hybrid optimization algorithm (SA-ALNS) based on the combination of simulated annealing (SA) and adaptive large-scale neighborhood search (ALNS) is proposed. An insertion heuristic algorithm based on time and distance weighting is used to construct the initial solution of the problem. A variety of delete and insert operators are introduced to optimize the path with adaptive selection strategy. Through the feedback mechanism, the probability of each operator being selected is gradually adjusted to make the algorithm more inclined to choose the operator with better optimization effect. The Metropolis criterion of simulated annealing mechanism is used to control the solution updating. In the simulation experiment, 56 large-scale examples are tested, and other intelligent optimization algorithms such as p-SA algorithm, DCS algorithm and VNS-BSTS are compared and statistically analyzed. The results

**基金项目:**甘肃省科技厅自然科学基金(1506RJZA084);甘肃省教育厅科研基金(1204-13);兰州市科技局科研基金(2015-2-74)。This work was supported by the Natural Science Foundation of Science and Technology Department of Gansu Province (1506RJZA084), the Scientific Research Fund of Gansu Provincial Department of Education (1204-13) and the Scientific Research Fund of Lanzhou Science and Technology Bureau (2015-2-74).

**收稿日期:**2021-05-10 **修回日期:**2021-07-26

show that the algorithm is feasible and superior in solving the vehicle routing problem with simultaneous delivery-pickup and time windows. The research results greatly enrich the related research of vehicle routing problem (VRP).

**Key words:** vehicle routing problem (VRP); simultaneous delivery-pickup; simulated annealing (SA); adaptive large-scale neighborhood search (ALNS); time windows

近年来,为了避免环境污染,很多国家制定了相应的法律法规,要求企业对整个产品的生命周期负责,不仅要提供客户满意的配送服务,还要进行废弃物的回收利用。因此,企业有了构建正向物流配送和逆向废旧物品回收相结合的物流系统的迫切需求;而客户往往只在特定的时间窗内接受服务。为了提升服务质量和客户满意度,企业不仅需要完成同时送取货的作业,还要额外考虑客户的服务时间窗。因此,如何设计出合理完善的带时间窗的双向物流系统,成为理论研究和实践中关注的重要问题。在运筹学领域,该类问题被称为带时间窗和同时送取货的车辆路径问题(vehicle routing problem with simultaneous delivery-pickup and time windows, VRPSDPTW)。

VRPSDPTW 问题作为车辆路径问题(vehicle routing problem, VRP)的重要扩展,属于 NP-hard 问题。这类问题的解决方法主要分为精确算法和启发式算法。2002年, Angelelli 等<sup>[1]</sup>最早提出 VRPSDPTW 问题,使用精确算法中的分支-价格法求解算例规模为 20 个客户点的小型算例。当客户规模增大时,精确算法不能在合理的时间范围内求得满意解。学者们开始尝试设计不同的启发式算法来解决 VRPSDPTW 问题。Cao 等<sup>[2]</sup>提出了一种改进的遗传算法,根据用户的优先顺序重新设计了编码方式。当最大客户数目为 8 时,改进算法能有效找到最优解,然而,在现实生活中,客户数目远远多于 8 个。Boubahri 等<sup>[3]</sup>设计了一个多智能体群体系统来解决 VRPSDPTW 问题。尽管这种解决方法很新颖,但是作者没有提供任何数值实验结果。2012年, Wang 等<sup>[4]</sup>采用协同进化遗传算法解决 VRPSDPTW 问题,并在经典的 Solomon 数据集的基础上设计了 VRPSDPTW 基准实例,实验结果表明,所提出的协同进化遗传算法能够在较短的时间内找到满意解。王超等<sup>[5]</sup>运用回溯搜索优化算法来求解该问题,选取了测试数据集中的 6 个算例进行了实验。王超等<sup>[6]</sup>提出了一种离散布谷鸟算法来求解该问题。除此之外,还有改进全局人工鱼群算法<sup>[7]</sup>、大邻域搜索算法<sup>[8]</sup>等。

目前关于 VRPSDPTW 问题,虽然学者们已经进行了一些研究,但是仍然缺乏有效的求解方法。迄今为止,VRPSDPTW 问题的解决,主要采用启发式算法,并且以 Wang 等设计的测试数据集作为国际通用数据集,各类文献中的已知算法在对算例进行测试时,都无法找到所有算例的最优解,因此,高效、稳定的求解方法成为学者们追求的目标。

为了更好地解决 VRPSDPTW 问题,通过大量的实验发现,模拟退火算法(simulated annealing, SA)在依据概率突跳特性进行降温寻优的过程中,对较差的解具有一定的容忍性,使得算法的全局寻优能力优于很多当前流行的智能优化算法,同时能够在一定程度上克服对初始解的依赖性;自适应大规模邻域搜索算法(adaptive large-scale neighborhood search, ALNS)<sup>[9]</sup>能够较好地避免算法在寻优过程中陷入局部极值,且在邻域搜索的过程中,通过一些启发式信息来指导寻优,能够在一定概率上获得高质量的解。将模拟退火和大规模邻域搜索算法相结合,本文提出一种混合优化算法(SA-ALNS),以标准模拟退火算法作为主体,用插入启发式算法<sup>[10]</sup>构造问题的初始解,结合自适应大规模邻域搜索算法进行寻优。实验结果表明,本文所提出 SA-ALNS 有效缩短了物流配送路径长度,节约配送成本,为各企业解决 VRPSDPTW 问题提供有效的解决方案。

## 1 VRPSDPTW 问题

### 1.1 问题描述

VRPSDPTW 问题可以描述为:一个配送中心拥有若干型号相同且载重能力相同的车辆,为分布在不同位置的已知客户提供送货、取货服务。每一个客户的地理位置信息已知、送货量和取货量已知、服务时间窗以及持续服务时间已知。要求配送中心在满足客户需求的基础上规划出合理的配送路线,最小化总配送成本,并满足以下假设:

- (1) 每个客户只能由一辆车完成送货和取货服务,且需求不可拆分;
- (2) 每辆车均从配送中心出发,在送取货任务完

成后返回配送中心；

(3) 每辆车的载货量不能超过车辆载重；

(4) 在客户规定的服务时间窗内, 先进行送货服务, 再进行取货服务；

(5) 每一条道路都畅通, 不考虑道路突发状况。

### 1.2 符号定义

为了方便构造 VRPSDPTW 数学模型, 需要对所用到的符号进行说明:

(1) 参数

$N$  表示需要服务的客户集合和配送中心;  $N_i$  表示需要服务的客户集合,  $i \in \{1, 2, \dots, n\}$ ;  $N_0$  表示配送中心;  $K$  表示车辆集合,  $K = \{1, 2, \dots, m\}$ ;  $D_i$  表示每个客户  $i$  的送货量;  $P_i$  表示每个客户  $i$  的取货量;  $D_{ij}$  表示客户  $i$  到客户  $j$  的欧氏距离 ( $i, j \in N$ );  $[ET_i, LT_i]$  表示客户  $i$  的服务时间窗, 其中  $ET_i$  为客户  $i$  允许的最早开始服务的时间,  $LT_i$  为客户  $i$  的最晚开始服务时间;  $S_i$  表示客户  $i$  所需的持续服务时间;  $Q$  表示车辆的最大载重量。

(2) 决策变量

$x_{ij}^k$  表示车辆  $k$  是否从客户  $i$  行驶到客户  $j$ , 若是, 取值为 1, 否则取值为 0;  $y_i^k$  表示客户  $i$  是否由第  $k$  辆车负责服务, 若是, 取值为 1, 否则取值为 0。

(3) 衍生变量

$St_i$  表示客户  $i$  的开始服务时间;  $T_{ij}$  表示从客户  $i$  到客户  $j$  所花费的时间;  $p_{0k}$  表示车辆  $k$  从配送中心出发时的装载量;  $p_{ik}$  表示车辆  $k$  在离开客户  $i$  时的装载量,  $i \in \{1, 2, \dots, n\}$ 。

### 1.3 VRPSDPTW 数学模型

$$\min z = \sum_{k \in K} \sum_{i \in N} \sum_{j \in N} D_{ij} x_{ij}^k \quad (1)$$

$$\text{s.t.} \sum_{i \in N} \sum_{k \in K} x_{ij}^k = 1, \forall j \in N \quad (2)$$

$$\sum_{j \in N} x_{0j}^k = \sum_{i \in N} x_{i0}^k = 1, \forall k \in K \quad (3)$$

$$p_{0k} = \sum_{i=1}^n \left( D_i \sum_{j=0, j \neq i}^n x_{ij}^k \right), k \in K \quad (4)$$

$$p_{ik} = p_{(i-1)k} - D_i + P_i, i \in N \quad (5)$$

$$p_{ik} \leq Q, i \in N, k \in K \quad (6)$$

$$St_j = \begin{cases} St_i + S_i + T_{ij}, ET_i < St_i < LT_i, \forall i \in N \\ ET_i + S_i + T_{ij}, St_i \leq ET_i, \forall i \in N \end{cases} \quad (7)$$

其中, 式(1)为目标函数, 表示最小化车辆总距离; 约束(2)表示每个客户都只能由一辆车服务; 约束(3)

保证所有车辆从配送中心出发最终返回配送中心; 约束(4)表示车辆在离开配送中心时的装载量即某条路径上所有客户的送货量总和; 约束(5)表示车辆在离开每个客户时的装载量, 即离开上一个客户时的装载量减去本客户的收货量(车辆的送货量), 再加上本客户的寄货量(车辆的取货量); 约束(6)保证车辆在任何时刻的装载量都不超过车辆的最大载重量; 约束(7)表示满足客户的服务时间窗要求。

## 2 VRPSDPTW 问题的混合优化算法设计

混合优化算法(SA-ALNS)在SA的基础上结合ALNS思想进行寻优。SA-ALNS中设计了多个插入、删除算子, 且为每个算子赋予分数和权重, 在邻域搜索的过程中根据各个算子的分数和权重动态选取插入算子和删除算子对现有解进行重构, 从而得到质量更好的解。由于ALNS的接受准则和停止准则一般是根据目标值来判断, 这种方法存在准确性差、鲁棒性弱的缺点, 引入SA中的退火准则来控制解的更新, 提高解的质量。

### 2.1 编码方式

算法采用整数编码, 编号 0 表示配送中心, 编号 1, 2, ...,  $n$  表示客户点。令  $o = [o[1], o[2], \dots, o[L]]$ , 为负责服务客户的车辆顺序排列的组合。假设为客户服务的车辆规划路径如图 1 所示, 则  $o = [o[1], o[2]] = [0\ 7\ 2\ 3\ 9\ 1\ 0\ 0\ 6\ 5\ 8\ 4\ 0]$ 。从图 1 可以看出, 共有 9 个需要服务的客户, 配送中心一共启用了 2 辆车, 第一辆车从配送中心出发, 分别为客户 7、客户 2、客户 3、客户 9、客户 1 提供送取货服务, 最后返回配送中心; 第二辆车从配送中心出发, 服务客户 6、5、8、4 后, 返回配送中心。

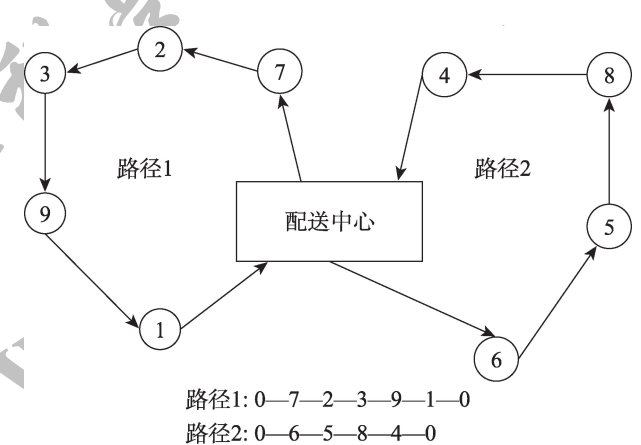


图 1 车辆规划路径

Fig.1 Vehicle planning route

## 2.2 初始解构造

采用插入启发式算法构造 VRPSDPTW 问题的初始解。用插入法构造初始解时,由于客户有服务时间窗约束,故采用 Solomon 的基于距离与时间加权的插入标准<sup>[10]</sup>。

$$c_{11}(i, u, j) = d_{iu} + d_{uj} - d_{ij} \quad (8)$$

式中,  $u$  为待插入客户点;  $d_{ij}$  表示客户  $i$  到客户  $j$  的距离;  $c_{11}(i, u, j)$  表示客户  $u$  插入  $i$ 、 $j$  两点间后,路径距离的增量。

$$c_{12}(i, u, j) = St_{ju} - St_j \quad (9)$$

式中,  $St_{ju}$  表示将客户  $u$  插入客户  $i$ 、 $j$  之间后,客户  $j$  的新开始服务时间;  $St_j$  为客户  $j$  的原开始服务时间;  $c_{12}(i, u, j)$  表示客户  $u$  插入路径后对客户  $j$  的开始服务时间的影响。

$$c_1(i, u, j) = c_{11}(i, u, j) + vc_{12}(i, u, j) \quad (10)$$

式中,  $v$  为车辆当前行驶速度,  $c_1(i, u, j)$  表示客户  $u$  插入路径后,对路径产生的影响。在配送过程中,为了同时考虑路径增加部分和服务时间增加部分对整体配送过程的影响,引入了车辆行驶当前速度值,通过当前速度与增加服务时间的乘积,将时间量转化为路程量,使得衡量的标准统一为距离,方便计算;考虑到车辆在配送过程中可能会出现交通拥堵的情况,车辆的行驶速度会发生变化,式(10)在速度动态变化的情况下能更精确地反映待插入客户对整个路径的影响程度,为求解多通路车辆路径问题提供一种思路。

$$c_2(i, u, j) = \lambda d_{ou} - c_1(i, u, j) \quad (11)$$

式中,  $c_2(i, u, j)$  表示插入标准值。通过  $c_2(i, u, j)$  筛选出距客户  $i$  和客户  $j$  越近且距离配送中心越远的客户点,优先插入路径中,从而提高后期优化阶段中 SA-ALNS 算法的收敛速度。计算中,引入常数  $\lambda \geq 1$ ,对  $d_{ou}$  进行放大,提高插入标准值的敏感性。

基于距离与时间加权的插入法,过程如下:

(1)生成一条只包含车辆起点和终点的初始路径[0 0];

(2)在所有的客户点中随机选择一个客户点作为种子节点插入初始路径中;

(3)计算剩余客户点在所有可能插入位置的插入标准值;

(4)将插入标准值降序排列,把插入标准值最大的客户点插入到最佳可行的位置,若路径中客户总送货量超过车辆的最大载重量,则重新生成一条路径;

(5)重复步骤(3)、(4),直到所有的客户都插入路径。

## 2.3 路径优化

在寻优阶段,引入删除算子、插入算子和自适应选择策略进行路径优化。删除算子按照一定的规则从路径中删除部分客户点,插入算子将删除的客户点重新添加到路径中的合理位置,得到新解。根据各个删除、插入算子在程序运行中的表现,自适应调整权重,不断地更新算子的选中概率,以不同的方式完成删除和插入操作,提高解的多样性,从而更好地进行全局寻优。

### 2.3.1 删除算子

选用四种删除算子,通过不同的方式随机在路径中选择客户点,将其在路径中删除,为局部寻优做准备。

#### (1)Random removal 算子

该算子从客户集合中随机选取  $m$  个客户,将  $m$  个客户从相应的路径中移除,以增加搜索的多样性。

#### (2)Random schedule removal 算子

该算子会随机选择一条路径,将这条路径上的所有客户删除,为这些客户重新寻找配送位置。这种方式在一定概率上能够减少车辆的使用数量,从而降低物流成本。

#### (3)Worst removal 算子

算子的主要目的是最小化总行驶距离。从客户集合中选取一个客户点,计算并保存该点被删除前后路径距离的节约值,剩余的客户点重复此过程,从而得到全部客户点被删除前后路径距离的节约值,将所有客户点距离的节约值降序排序,选取前  $m$  个节约值较大的客户点从相应的路径中删除。

#### (4)Node distance removal 算子

该删除算子首先计算每条路径的平均距离占用值,然后将平均距离占用值最大的路径中的客户点全部删除。平均距离占用值的计算公式为:

$$f(i) = \frac{D(i)}{N_i} \quad (12)$$

式中,  $D(i)$  表示第  $i$  条路径的总距离,  $N_i$  表示第  $i$  条路径中的客户数量。根据平均距离占用公式,选择一条最不经济的路径删除其中所有客户点。

### 2.3.2 插入算子

文中选用两种插入算子,使用不同的插入规则将删除的客户点重新插入到路径中。

#### (1)Greedy insertion 算子

该方法随机选取一个待插入客户点, 遍历每条路径, 找到所有既满足时间窗约束又不超过车辆最大载重量的可插入位置, 计算该客户点插入到每个可插入位置前后的距离增量, 选择距离增量最小的位置将客户点插入。若遍历所有路径都没有找到符合时间窗和车辆载重约束的可插入位置, 就重新生成一条初始路径, 将客户点添加到其中。重复上述过程, 直到将所有的待插入客户点添加到路径中。

### (2) Regret insertion 算子

Regret insertion 算子采用 2-regret 准则。计算每个客户点在所有可插入位置插入后产生的距离增量  $\Delta d_i$ , 算出距离增量最小的两个值的差  $\Delta d_i^2 - \Delta d_i^1$ , 找到差值最大的那个客户点  $i^*$ , 将其插入到路径中, 计算公式如式(13)。

$$i^* = \arg \max_{i \in S} (\Delta d_i^2 - \Delta d_i^1) \quad (13)$$

式中,  $S$  为全部待插入客户点集合;  $\Delta d_i^1$  表示客户  $i$  的最优插入位置的距离增量;  $\Delta d_i^2$  表示客户  $i$  的次优插入位置的距离增量。若有客户点不能插入到现有的路径中, 则生成一条新路径。重复以上方法, 将剩余待插入客户添加到路径中。该算子优先插入最优位置与次优位置距离较远的, 即对总路程影响较大的客户点, 可以最大程度减少总行驶距离。

### 2.3.3 自适应选择策略

混合算法以模拟退火算法的基本结构进行寻优, 寻优过程中, 以自适应方式选择删除和插入算子。

每一个温度下(以下简称为阶段), 算法进行  $L$  次寻优迭代; 每次迭代时, SA-ALNS 采用轮盘赌机制选择一个删除和一个插入算子完成路径的更新; 之后根据新得到的路径的优劣, 对所选用的删除、插入算子评分。得分分数将影响下一阶段中该算子的权重, 得分越高的算子在下一阶段轮盘赌选择策略中的权重值将越大, 被选中的概率越高。算子  $p$  第  $M$  阶段的得分, 计算公式如式(14)。

$$\pi_{pM}^{\gamma+1} = \pi_{pM}^{\gamma} + \begin{cases} \sigma_1, & \text{新生成的解优于全局最优解} \\ \sigma_2, & \text{新生成的解优于当前解, 但比全局最优解差} \\ \sigma_3, & \text{新生成的解比当前解差} \end{cases} \quad (14)$$

其中,  $\gamma = 1, 2, \dots, L-1$ , 表示第  $M$  阶段的迭代次数, 在每一阶段的首次迭代时, 各个算子的分数均重置为 0, 即  $\pi_{pM}^1 = 0, p = 1, 2, \dots, h$ ;  $\sigma_1, \sigma_2, \sigma_3$  为常数, 表示操作算子在相应情况下所得分数;  $\sigma_1 > \sigma_2 > \sigma_3$ , 本文取  $\sigma_1 = 30, \sigma_2 = 10, \sigma_3 = 6$ 。

在当前温度  $T$  下算法经过  $L$  次迭代之后, 降温, 到达下一个阶段。在新的阶段中, 需要依据各算子在上一轮的得分情况更新权重; 在第一阶段, 算子  $p$  均有一个值为 1 的初始权重  $\omega_{p,1}, p = 1, 2, \dots, h$ , 此时各算子被选中的概率相同; 算法进行到  $M$  阶段时, 算子  $p$  被选择的概率为  $\omega_{p,M} / \sum_{i=1}^h \omega_{i,M}$ 。当温度更新  $M$  次即进入  $M+1$  个阶段, 此时算子  $p$  的权重  $\omega_{p,M+1}$  更新方式如式(15)。

$$\omega_{p,M+1} = (1 - \chi) \omega_{p,M} + \frac{\chi \pi_{pM}}{\varepsilon_{pM}} \quad (15)$$

其中,  $\pi_{pM}$  表示算子  $p$  在第  $M$  个阶段所得分数;  $\varepsilon_{pM}$  表示算子  $p$  在第  $M$  阶段被选择的次数; 常数  $\chi \in [0, 1]$ , 本文取  $\chi$  的值为 0.1。每个算子的权重与该算子所得分数、选择次数有关, 分数越高, 权重越大, 选择概率就越大。算子的权重作为一种启发式信息, 使算法偏向于选择效果好的算子, 从而在很大概率上能够获得更满意的解。

### 2.4 可行解的接受条件和迭代结束标准

SA-ALNS 算法运用模拟退火算法的 Metropolis 准则来控制解的更新。令  $s_1$  表示当前解,  $s_2$  表示新解,  $f(s_1), f(s_2)$  分别为  $s_1, s_2$  的目标函数值,  $T$  表示当前温度,  $\Delta f = f(s_2) - f(s_1)$ 。若  $\Delta f < 0$ , 则用新解代替当前解; 否则计算新解的接受概率  $\exp(-\Delta f/T)$ , 并产生  $[0, 1]$  之间的随机数  $rand$ , 若  $\exp(-\Delta f/T) > rand$ , 则新解  $s_2$  被接受。劣质解的接受概率随着当前温度  $T$  的不断下降而降低。当温度  $T$  低于结束温度时, 算法结束。

### 2.5 混合优化算法流程

**步骤 1** 初始化基本参数: 初始温度  $T_0$ , 每个温度下的最大迭代次数  $L$ , 结束温度  $T_{end}$ 。采用插入启发式算法构造初始解。

**步骤 2** 用轮盘赌机制分别选择一种删除算子和插入算子, 生成一个新解  $s_2$ 。

**步骤 3** 计算  $s_2$  的目标函数值, 根据 Metropolis 准则更新当前解, 并更新各个算子的分数。

**步骤 4** 若达到每个温度下的最大迭代次数  $L$ , 则更新删除、插入算子的权重并执行降温  $T = T_p$  操作, 其中,  $p$  为降温速率。

**步骤 5** 若达到足够低的温度  $T(T < T_{end})$ , 输出最优解, 整个算法结束; 否则回到步骤 2。

混合优化算法的流程图如图2所示。

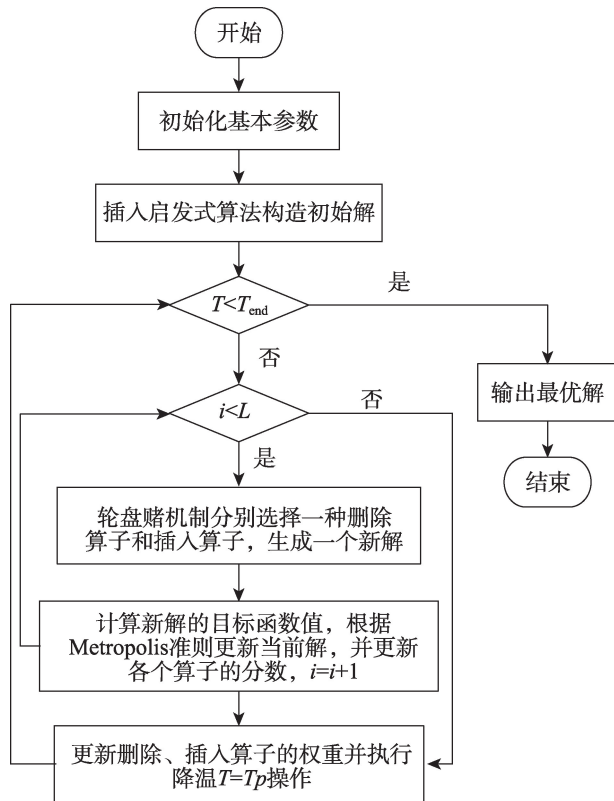


图2 混合算法流程

Fig.2 Hybrid algorithm flow

### 3 实验与分析

程序使用 Matlab (R2016a) 编写, 并在 Intel® Core™ i5-7300HQ CPU@2.50 GHz 处理器, 8 GB 内存和 16 位操作系统的 Windows10 计算机上进行。

为了验证 SA-ALNS 算法的性能, 选取 Wang 等设计的求解 VRPSDPTW 的测试算例 (数据集网址 <http://oz.nthu.edu.tw/~d933810/test.htm>)。共 56 个大规模算例, 可分为 Rdp1、Rdp2、Cdp1、Cdp2、RCdp1、RCdp2, 共 6 类。Rdp1 类和 Rdp2 类中的客户地理位置满足随机分布, 较为分散; 算例 Cdp1、Cdp2 中的客户位置满足聚类分布, 相对较聚集; RCdp1 类和 RCdp2 类客户地理位置通过聚类分布和随机分布混合生成, 部分聚集, 部分分散。此外, Rdp1、Cdp1、RCdp1 算例集具有较紧的时间窗, 较小的车辆装载量; 而 Rdp2、Cdp2、RCdp2 算例集的时间窗较宽, 车辆装载量较大。

SA-ALNS 算法在每个算例上运行 10 次, 并记录最优值。本文提出的 SA-ALNS 算法与文献 [6] 中的 DCS 算法、文献 [11] 中的 p-SA 算法、文献 [12] 中的

ETSP 算法、文献 [13] 中的 ALNS-PR 算法、文献 [14] 中的 VNS-BSTS 算法以及文献 [15] 中的 Par-SAA 算法进行对比。SA-ALNS 算法在 Cdp1 类、Cdp2 类算例上与其他算法最优值的对比结果如表 1 所示, 在 Rdp1 类、Rdp2 类算例上的对比结果如表 2 所示, 在 RCdp1 类、RCdp2 类算例上的对比结果如表 3 所示。

表1 SA-ALNS在Cdp类算例与其他算法的最优值对比

Table 1 Best value comparison between SA-ALNS and other algorithms in Cdp examples

算例	p-SA	DCS	VNS-BSTS	ETSP	ALNS-PR	SA-ALNS
Cdp101	992.88	998.29	976.04	—	976.04	963.96
Cdp102	955.31	954.31	942.45	—	941.49	935.30
Cdp103	958.66	923.05	896.28	—	892.98	888.52
Cdp104	944.73	931.26	872.39	—	871.40	881.94
Cdp105	989.86	981.45	1 080.63	—	1 053.12	983.10
Cdp106	878.29	878.29	963.45	—	967.71	878.29
Cdp107	911.90	912.37	987.64	—	987.64	908.99
Cdp108	1 063.73	978.82	934.41	—	932.88	936.10
Cdp109	947.90	940.49	909.27	—	910.95	932.69
Cdp201	591.56	591.56	591.56	591.56	591.56	591.56
Cdp202	591.56	591.56	591.56	591.56	591.56	591.56
Cdp203	591.17	591.17	591.17	591.17	591.17	591.17
Cdp204	594.07	590.60	599.33	599.87	590.60	591.17
Cdp205	588.88	588.88	588.88	588.88	588.88	588.88
Cdp206	588.49	588.49	588.49	588.49	588.49	588.49
Cdp207	588.29	588.29	588.29	588.29	588.29	588.29
Cdp208	599.32	588.32	588.32	588.32	588.32	588.32

混合优化算法的参数设置: 初始温度设为 500, 截止温度为 0.001, 降温速率为 0.95, 最大迭代次数设为 100。

对 Cdp1 类、Cdp2 类算例进行测试, SA-ALNS 算法在每个算例上的最好结果见表 1。从表中可以看出, 在 Cdp104、Cdp108、Cdp109 算例中, SA-ALNS 算法的实验结果比 VNS-BSTS、ALNS-PR 的结果差; 在 Cdp105 算例中, SA-ALNS 的寻优效果弱于 DCS 算法; 在 Cdp204 算例上, DCS、ALNS-PR 算法的实验结果优于 SA-ALNS; 在除 Cdp204 的其他 Cdp2 类算例中, SA-ALNS 算法均获得了最优解; 本文算法在 Cdp1 类和 Cdp2 类算例上都优于 p-SA 算法。总的来说, SA-ALNS 算法的计算精度较高。

由表 2 可看出, 与 p-SA 算法相比, 在 Rdp103、Rdp104、Rdp107、Rdp108 算例上, SA-ALNS 的求解性能弱于 p-SA, 在其余算例上的求解性能均优于 p-SA;

表2 SA-ALNS在Rdp类算例与其他算法的最优值对比

Table 2 Best value comparison between SA-ALNS and other algorithms in Rdp examples

算例	p-SA	DCS	VNS-BSTS	ETSP	ALNS-PR	Par-SAA	SA-ALNS
Rdp101	1 660.98	1 658.65	1 650.80	—	1 650.80	1 659.76	1 650.20
Rdp102	1 491.75	1 490.13	1 486.12	—	1 486.12	1 490.25	1 485.47
Rdp103	1 226.77	1 228.48	1 294.75	—	1 297.01	1 307.37	1 228.10
Rdp104	1 000.65	1 005.99	984.81	—	984.81	1 000.65	1 011.50
Rdp105	1 399.81	1 340.06	1 377.11	—	1 377.11	1 399.81	1 366.87
Rdp106	1 275.69	1 270.29	1 261.40	—	1 252.03	1 271.95	1 260.90
Rdp107	1 082.92	1 084.00	1 144.02	—	1 121.86	1 172.60	1 085.40
Rdp108	962.48	964.38	968.32	—	965.54	1 003.78	967.40
Rdp109	1 181.92	1 156.90	1 224.86	—	1 194.73	1 177.75	1 155.75
Rdp110	1 106.52	1 108.81	1 101.33	—	1 148.20	1 106.52	1 095.23
Rdp111	1 073.62	1 077.65	1 117.76	—	1 098.84	1 073.62	1 069.50
Rdp112	966.06	977.59	961.29	—	1 010.42	966.06	963.77
Rdp201	1 286.55	1 281.63	1 254.57	1 268.52	1 253.23	—	1 181.73
Rdp202	1 150.31	1 152.65	1 202.27	1 099.61	1 191.70	—	1 065.16
Rdp203	997.84	950.79	949.42	981.48	946.28	—	927.65
Rdp204	848.01	776.00	837.13	775.93	833.09	—	790.96
Rdp205	1 046.06	1 051.38	1 027.49	1 045.12	994.43	—	1 000.14
Rdp206	959.94	957.81	938.63	973.44	913.68	—	948.28
Rdp207	899.82	890.52	912.26	841.24	890.61	—	849.44
Rdp208	739.06	737.05	737.26	740.82	726.82	—	728.83
Rdp209	947.80	930.26	940.29	999.06	909.16	—	909.35
Rdp210	1 005.11	1 005.11	945.97	964.59	939.37	—	939.80
Rdp211	812.44	819.88	805.22	805.53	904.44	—	774.99

表3 SA-ALNS在RCdp类算例与其他算法的最优值对比

Table 3 Best value comparison between SA-ALNS and other algorithms in RCdp examples

算例	p-SA	DCS	VNS-BSTS	ETSP	ALNS-PR	SA-ALNS
RCdp101	1 659.59	1 654.32	1 708.21	—	1 776.58	1 666.12
RCdp102	1 522.76	1 522.76	1 526.36	—	1 583.62	1 494.01
RCdp103	1 344.62	1 344.63	1 336.05	—	1 283.52	1 351.66
RCdp104	1 268.43	1 269.31	1 177.21	—	1 171.65	1 182.42
RCdp105	1 581.54	1 581.26	1 548.38	—	1 548.96	1 566.96
RCdp106	1 418.16	1 419.26	1 408.19	—	1 392.47	1 424.07
RCdp107	1 360.17	1 360.17	1 295.43	—	1 255.06	1 295.88
RCdp108	1 169.57	1 170.12	1 207.60	—	1 198.36	1 164.45
RCdp201	1 513.72	1 520.56	1 437.48	1 441.52	1 406.94	1 326.19
RCdp202	1 273.26	1 242.92	1 412.52	1 216.59	1 414.55	1 159.19
RCdp203	1 123.58	1 087.37	1 064.95	1 106.72	1 050.64	959.17
RCdp204	897.14	822.02	813.74	900.65	798.46	796.12
RCdp205	1 357.44	1 357.44	1 316.06	1 253.44	1 297.65	1 188.50
RCdp206	1 166.88	1 166.88	1 154.36	1 161.89	1 146.32	1 139.60
RCdp207	1 089.85	1 093.37	1 098.64	1 125.50	1 061.84	1 023.51
RCdp208	862.89	862.89	843.30	873.28	828.14	793.98

同DCS算法相比较,SA-ALNS在Rdp104、Rdp105、Rdp107、Rdp108、Rdp204算例上的效果弱于DCS,其余算例的效果优于DCS;与VNS-BSTS算法相比较,在Rdp104、Rdp112、Rdp206算例上,SA-ALNS的实验结果较差,但在其他算例上都领先于VNS-BSTS;同ETSP相比,SA-ALNS在除Rdp204、Rdp207之外的其他算例上都获得最优解;同ALNS-PR相比,SA-ALNS在Rdp104、Rdp106、Rdp108、Rdp205、Rdp206、Rdp208、Rdp209、Rdp210算例上的求解性能弱于ALNS-PR,但在其余算例上求解性能强于ALNS-PR;与Par-SAA算法相比,SA-ALNS在除Rdp104的其他算例上均优于Par-SAA。在Rdp1、Rdp2类算例中,SA-ALNS在82.6%算例上优于p-SA,在78.2%算例上优于DCS,在87.0%算例上优于VNS-BSTS,在81.8%算例上优于ETSP,在65.2%算例上优于ALNS-PR,在91.7%算例上优于Par-SAA。

实验结果分析:由表3可以看出,在RCdp1类算例中,SA-ALNS在求解RCdp101、RCdp103、RCdp106算例时,寻优结果弱于p-SA和DCS,但在其余算例的寻优结果都强于p-SA、DCS;SA-ALNS在求解RCdp103、RCdp104、RCdp105、RCdp106、RCdp107时,求解能力比VNS-BSTS、ALNS-PR弱。在RCdp2类算例,SA-

ALNS算法的运行结果均优于对比算法。

从Cdp1、Cdp2、Rdp1、Rdp2、RCdp1、RCdp2算例的实验结果可以看出,本文所提出的SA-ALNS在求解VRPSDPTW问题中有不错的效果,说明该方法是行之有效的。

图3是SA-ALNS与其他的算法分别在Cdp1类、Cdp2类、Rdp1类、Rdp2类、RCdp1类、RCdp2类算例上的实验结果对比,能够更加清晰地看出不同算法求得的最短距离值之间的差距。

部分算例的路径优化图,如图4。

#### 4 结束语

本文针对VRPSDPTW问题提出了一种SA-ALNS算法,该算法运用插入启发式算法构造初始解,以模拟退火为框架,采用自适应大规模邻域搜索算法进行邻域搜索,并设计了四种删除算子和两种插入算子,增大邻域搜索的范围,从而加速解的收敛。通过Cdp1类、Cdp2类、Rdp1类、Rdp2类、RCdp1类、RCdp2类算例测试,并与其他已知算法对比,最后证明了SA-ALNS算法的优越性和有效性。现如今,人们越来越关注环境保护问题,绿色物流将是未来物流行业的发展趋势,在车辆路径问题中考虑碳排放等约

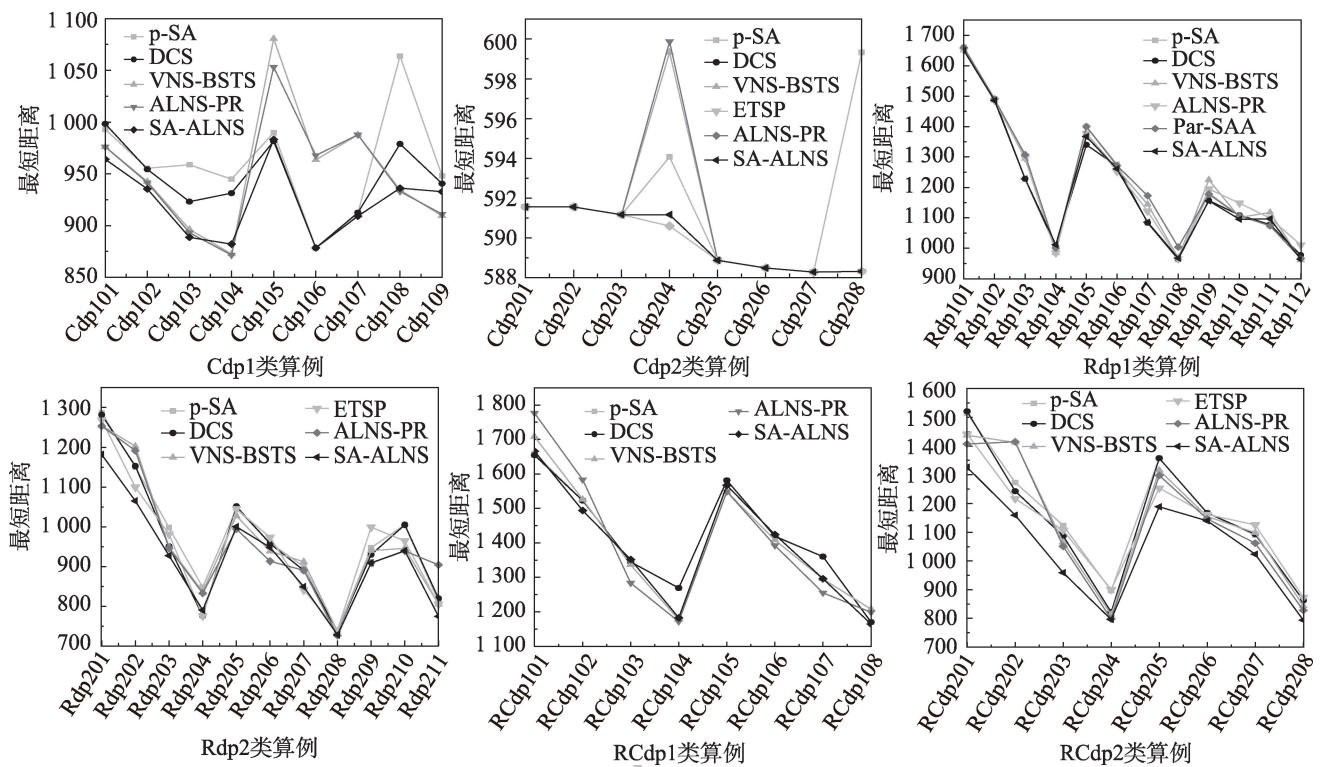


图3 不同算法在不同算例上的对比

Fig.3 Comparison of different algorithms on different examples



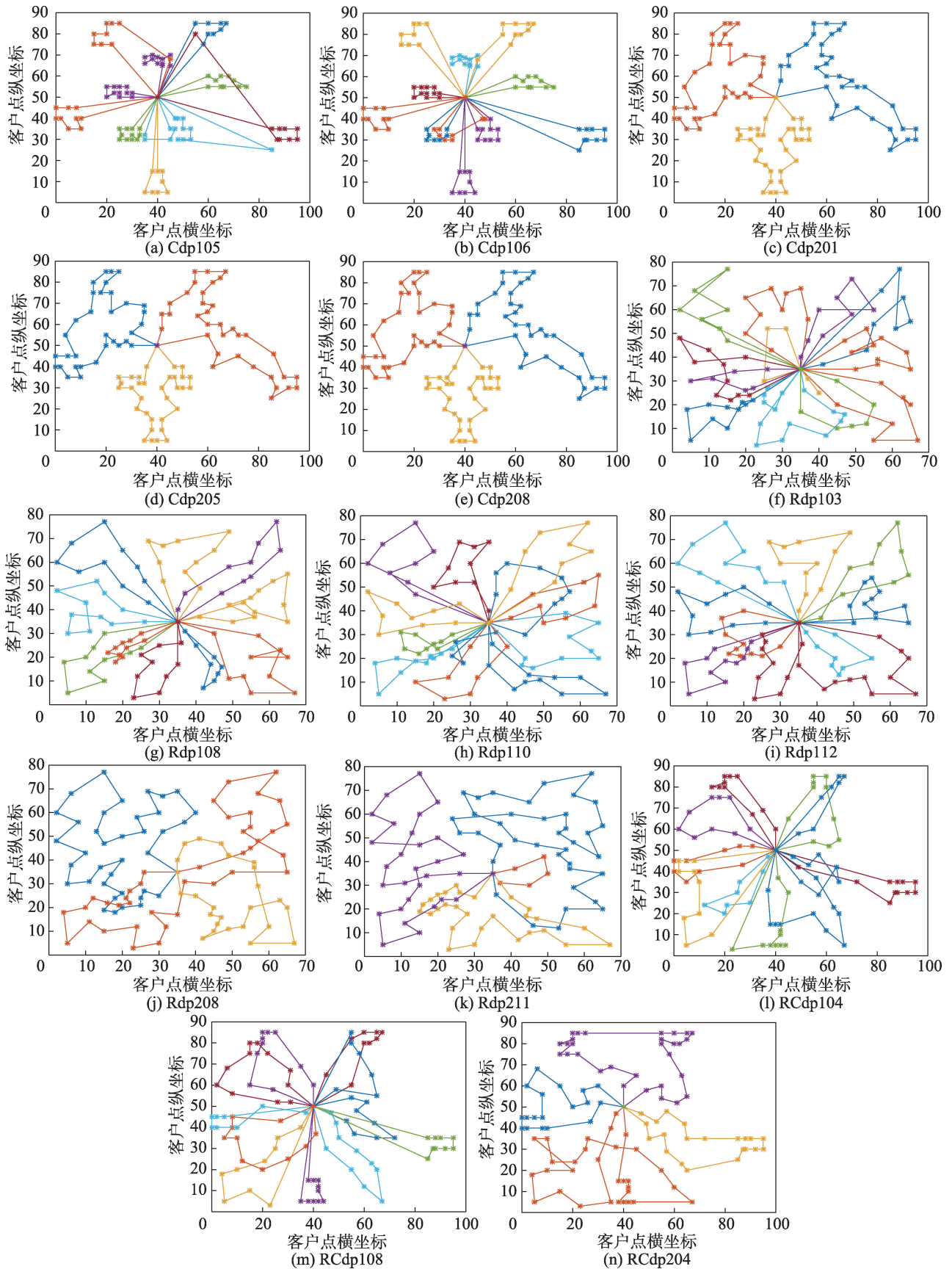


图4 部分算例的路径优化图

Fig.4 Path optimization graph of some examples

束条件,从而衍生出更多复杂的VRP问题,本文所提出的SA-ALNS为解决这些复杂VRP问题提供一种新思路。

### 参考文献:

- [1] ANGELELLI E, MANSINI R. The vehicle routing problem with time windows and simultaneous pick-up and delivery [M]. Berlin, Heidelberg: Springer, 2002.
- [2] CAO E B, LAI M Y. Vehicle routing problem with simultaneous delivery and pick-up with time windows[C]//Proceedings of the 1st International Conference on Transportation Engineering, Chengdu, Jan 1, 2007: 160-166.
- [3] BOUBAHRI L, ADDOUCHE S A, EI MHAMED A. Multi-ant colonies algorithms for the VRPSPDTW[C]//Proceedings of the 2011 International Conference on Communications, Computing and Control Applications, Hammamet, Mar 3-5, 2011. Piscataway: IEEE, 2011: 1-6.
- [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] 王超, 高扬, 刘超, 等. 基于回溯搜索优化算法求解带时间窗和同时送取货的车辆路径问题[J]. 计算机集成制造系统, 2019, 25(9): 2237-2247.
- WANG C, GAO Y, LIU C, et al. Vehicle routing problem with simultaneous delivery and pickup problem solving by backtracking search optimization algorithm[J]. Computer Integrated Manufacturing Systems, 2019, 25(9): 2237-2247.
- [6] 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. VRPSPDTW problem solving by discrete cuckoo search[J]. Computer Integrated Manufacturing Systems, 2018, 24(3): 570-582.
- [7] 黄务兰, 张涛. 基于改进全局人工鱼群算法的VRPSPDTW研究[J]. 计算机工程与应用, 2016, 52(21): 21-29.
- HUANG W L, ZHANG T. Vehicle routing problem with simultaneous pick-up and delivery and time-windows based on improved global artificial fish swarm algorithm[J]. Computer Engineering and Applications, 2016, 52(21): 21-29.
- [8] 程谦, 张大力, 侯立文. 具有时间窗的取送货问题建模和大邻域搜索算法[J]. 哈尔滨商业大学学报(自然科学版), 2016, 32(6): 734-739.
- CHENG Q, ZHANG D L, HOU L W. Study on model for pickup and delivery problem with time windows and large neighborhood search algorithm[J]. Journal of Harbin University of Commerce (Natural Sciences Edition), 2016, 32(6): 734-739.
- [9] ROPKE S, PISINGER D. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows[J]. Transportation Science, 2006, 40(4): 455-472.
- [10] SOLOMON M M. Algorithms for the vehicle routing and scheduling problems with time window constraints[J]. Operations Research, 1987, 35(2): 254-265.
- [11] 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(C): 111-122.
- [12] SHI Y, BOUDOUH T, GRUNDER O. An efficient tabu search based procedure for simultaneous delivery and pick-up problem with time window[J]. IFAC-PapersOnLine, 2018, 51(11): 241-246.
- [13] HOF J, SCHNEIDER M. An adaptive large neighborhood search with path relinking for a class of vehicle-routing problems with simultaneous pickup and delivery[J]. Networks, 2019, 74(3): 207-250.
- [14] SHI Y, ZHOU Y J, BOUDOUH T, et al. A lexicographic-based two-stage algorithm for vehicle routing problem with simultaneous pickup-delivery and time window[J]. Engineering Applications of Artificial Intelligence, 2020, 95(2): 103901.
- [15] ZHANG S Q, MU D, WANG C. A solution for the full-load collection vehicle routing problem with multiple trips and demands: an application in Beijing[J]. IEEE Access, 2020, 8: 89381-89394.



李珺(1974—),女,辽宁辽阳人,博士,副教授,主要研究方向为智能计算、图像处理。

LI Jun, born in 1974, Ph.D., associate professor. Her research interests include intelligent computing and image processing.



段钰蓉(1997—),女,山西运城人,硕士研究生,主要研究方向为智能计算。

DUAN Yurong, born in 1997, M.S. candidate. Her research interest is intelligent computing.



郝丽艳(1994—),女,山西吕梁人,硕士研究生,主要研究方向为智能计算。

HAO Liyan, born in 1994, M.S. candidate. Her research interest is intelligent computing.



张维维(1996—),女,河南洛阳人,硕士研究生,主要研究方向为智能计算。

ZHANG Weiwei, born in 1996, M.S. candidate. Her research interest is intelligent computing.