

DOI:10.13196/j.cims.2023.07.025

## “卡车+无人机”模式下带时间窗的取送货车辆路径问题

吴廷映,陶新月,孟婷

(上海大学 管理学院,上海 200444)

**摘要:**无人机由于成本低、速度快、直线飞行等优势被越来越多地用于物流配送。考虑到无人机的载货量小、续航能力有限等因素,针对“卡车+无人机”模式下的带时间窗的取送货车辆路径问题进行优化,建立了以最小化运输成本为目标的混合整数规划模型。设计了自适应大邻域搜索算法对其求解。在该算法中,通过设计基于模型特点的多种高效的破坏算子和修复算子、引入模拟退火接受准则避免陷入局部最优解来提高算法性能。通过算例求解验证了模型和算法的有效性,分析了卡车与无人机联合配送模式对路径成本的影响,为“卡车+无人机”模式的路径规划提供决策依据。

**关键词:**“卡车+无人机”联合配送;自适应大邻域搜索算法;带时间窗的取送货问题;车辆路径问题

中图分类号:U492.2; F252 文献标识码:A

### Pickup and delivery problem with time windows in mode of " truck +drone"

WU Tingying, TAO Xinyue, MENG Ting

(School of Management, Shanghai University, Shanghai 200444, China)

**Abstract:** Drones are increasingly used in logistics delivery due to their low cost, high-speed and straight-line flight. Considering the small cargo capacity, limited endurance and other factors, the pickup and delivery vehicle routing problem with time windows in the mode of " truck + drone" was optimized. A mixed integer programming model with the objective of minimizing transportation cost was proposed and an improved adaptive large neighborhood search algorithm was designed to solve the problem. In this algorithm, the performance of the algorithm was improved by designing various efficient destroy operators and repair operators based on the characteristics of the model and introducing a simulated annealing strategy to avoid falling into local optimum solutions. The effectiveness of the model and the algorithm was verified through the numerical experiments, and the impact of the " truck + drone" on the route cost was analyzed, which provided a decision basis for the route planning of " truck + drone" mode delivery.

**Keywords:** " truck + drone" mode delivery; adaptive large neighborhood search algorithm; pickup and delivery problem with time windows; vehicle routing problem

## 0 引言

随着电子商务的快速发展,物流需求量不断攀升,许多物流企业开始注重配送的效率和经济性。无人机由于成本低、速度快、直线行驶的优点在最后一公里配送中受到众多企业关注,亚马逊、谷歌、DHL 等多家企业已经进行了实际试验<sup>[1-3]</sup>,并且正在逐步构建无人机配送系统。而无人机具有载货量

小、续航里程短、容易受干扰的缺点<sup>[4]</sup>,单独使用无法满足现代物流的需求。采用卡车与无人机联合配送,利用无人机的优势,减少卡车的配送数量,可实现降低运输成本,提高配送效率。

卡车与无人机联合配送在 2015 年 MURRAY 等<sup>[5]</sup>提出后,一直是学者研究的热点问题。MURRAY 等<sup>[5]</sup>建立了飞行助手旅行商问题和并行无人机调度旅行商问题模型,并提出了混合整数规

收稿日期:2022-04-07;修订日期:2022-04-21。Received 07 Apr. 2022; accepted 21 Apr. 2022.

划和启发式求解方法。AGATZ 等<sup>[6]</sup>在文献[5]基础上构建了带无人机的旅行商问题模型,开发了基于局部搜索和动态规划的求解方法。WANG<sup>[7]</sup>等构建了无人机的车辆路径问题模型,验证了卡车与无人机联合配送的配送时间即使在最坏的情况下也少于卡车单独配送的时间。SCRAMENTO 等<sup>[8]</sup>针对无人机车辆路径问题(Vehicle Routing Problem with Drones, VRP-D)模型提出了自适应大邻域搜索算法求解。孟姗姗等<sup>[9]</sup>考虑了无人机一次可携带多个包裹的多卡车-无人机混合配送问题,提出两阶段协同优化算法最小化成本。

带时间窗的取送货车辆路径问题(Pickup and Delivery Problem with Time Windows, PDPTW)不仅在物流配送中有着广泛的应用<sup>[10]</sup>,也一直是学术研究热点,BALDACCI 等<sup>[11]</sup>通过结合整数规划、对偶理论和列生成等多种精确算法对其进行求解。BENT 等<sup>[12]</sup>利用两阶段算法求解 PDPTW,第一阶段使用模拟退火算法减少车辆数量,第二阶段采用大邻域搜索算法最小化路径成本。ROPKE<sup>[13]</sup>等在大邻域搜索算法的基础上提出了求解 PDPTW 的自适应大邻域搜索算法(Adaptive Large Neighborhood Search, ALNS),结果表明,ALNS 在求解大规模问题上具有良好的效果。SARTORI<sup>[14]</sup>等提出了该问题的一个改进的混合算法和一种基于开放数据生成路径问题实例的新方法。NACCACHE 等<sup>[15]</sup>提出了具有时间窗的多次取货和送货问题,并开发了一种具有改进操作的混合自适应大邻域搜索算法求解此问题。GOEKE<sup>[16]</sup>提出了改进的禁忌搜索算法求解带时间窗的电动汽车取送货问题。冷龙龙等<sup>[17]</sup>建立以碳排放为目标的同时取送货低碳选址—路径问题模型,并利用量子超启发式算法对模型进行求解。任腾等<sup>[18]</sup>考虑将电商订单与同城订单共同配送以提高车辆满载率,构建了以个人客户、电商企业客户和运输部门总支出费用最小化为目标的路径优化数学模型,并设计改进的遗传算法求解。

在以往文献中,尚未见到“卡车+无人机”模式下的 PDPTW 的研究,而随着无人机的商用价值不断被挖掘,同城配送、“生鲜到家”、外卖等需求不断上升<sup>[19]</sup>,考虑卡车+无人机的模式优化 PDPTW 可丰富物流配送模式,拓展无人机的业务场景,因此本文针对“卡车+无人机”模式下的带时间窗的取送货车辆路径问题(Pickup and Delivery Problem with Time Windows and Drone,

PDPTW-D)进行研究,建立以最小化车辆数量和路径成本的 PDPTW-D 混合整数规划模型;提出了 ALNS 算法求解该问题。在该算法中,提出了基于模型特点的多种高效的破坏算子和修复算子、引入模拟退火接受准则避免陷入局部最优解来提高算法性能。通过测试不同规模的算例验证了算法的有效性,结果表明采用卡车与无人机的联合运输模式可大量节约路径成本。

## 1 问题描述及模型

### 1.1 问题描述

本文的研究问题可描述为:物流网络中仅有一个配送中心,配送中心有数量不受限制的同种类型卡车和无人机。卡车从配送中心出发,每辆卡车携带一个无人机,车辆载货量不得超过其最大容量,卡车先到取货点取货,而后卡车或无人机将货物送到对应的送货点,配送过程中卡车与无人机均保持匀速行驶。无人机自卡车处出发,每次只能配送载货量和电量续航满足约束的送货点,而后与卡车汇合。无人机配送时,卡车前往下一个客户点完成取送货需求。无人机与卡车汇合后会立即更换电池,确保下一次无人机飞行的续航能力。每个客户点只能由卡车或无人机访问一次,车辆访问必须满足客户时间窗的要求,早于客户时间窗开始时间到达需要等待,晚于时间窗结束时间到达则不得访问。

### 1.2 符号及参数

- $P = \{1, 2, \dots, n\}$ , 表示取货点集合;
- $D = \{n + 1, \dots, 2n\}$ , 表示送货点集合;
- $K = \{1, 2, \dots, k\}$ , 表示卡车集合;
- $M$  表示无穷大数;
- $Q$  表示卡车最大载货量;
- $e$  表示无人机电池最大续航时间;
- $d_{ij}$  表示车辆从  $i$  到  $j$  的行驶距离;
- $c$  表示无人机配送的距离成本系数;
- $[e_i, l_i]$  表示客户  $i$  的时间窗范围;
- $L$  表示发射无人机所用时间;
- $R$  表示回收无人机所用时间;
- $x_{ijk}$  为 0-1 变量, 卡车经过弧  $(i, j)$  时  $x_{ijk} = 1$ ;
- $y_{ijk}$  为 0-1 变量, 无人机经过弧  $(i, j)$  时  $y_{ijk} = 1$ ;
- $s_j$  表示卡车在客户  $j$  的服务时间;
- $t_{ik}$  表示卡车到达客户点  $i$  的时间;
- $t_{di}$  表示无人机到达客户点  $i$  的时间;
- $t_{ij}^T$  表示车辆从  $i$  点行驶到  $j$  的时间。

0 为起点时,  $2n + 1$  为终点,  $D'$  为送货点中无人机可以访问的客户点的集合,  $P'$  为对应的取货点集合。其中  $N = P \cup D$ ,  $V = N \cup \{0, 2n + 1\}$ ,  $D' \subseteq D$ ,  $P' \subseteq P$ 。

### 1.3 PDPTW-D 模型

构建的 PDPTW-D 模型如下:

$$\begin{aligned} \min \quad & (\sum_{k \in K} \sum_{j \in P} x_{0jk}, \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} x_{ijk} d_{ij} + \\ & \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} c d_{ij} y_{ijk}) . \end{aligned} \quad (1)$$

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

$$\sum_{j \in N} x_{ijk} = \sum_{j \in N} x_{j,n+i,k}, \forall i \in P \setminus P', \forall k \in K; \quad (3)$$

$$\sum_{j \in N} x_{ijk} - \sum_{j \in N} x_{j,n+i,k} - \sum_{j \in N} y_{j,n+i,k} = 0, \quad \forall k \in K, \forall i \in P'; \quad (4)$$

$$\sum_{j \in P} x_{0jk} = 1, \forall k \in K; \quad (5)$$

$$\sum_{i \in D} x_{i,2n+1,k} = 1, \forall k \in K; \quad (6)$$

$$x_{0,2n+1,k} = 0, \forall k \in K; \quad (7)$$

$$\sum_{i \in V} x_{iik} = 0, \forall k \in K; \quad (8)$$

$$\sum_{i \in V} y_{iik} = 0, \forall k \in K; \quad (9)$$

$$\sum_{i \in V \setminus \{2n+1\}} x_{ijk} - \sum_{m \in V \setminus \{0\}} x_{jmk} = 0, \quad \forall j \in N, \forall k \in K; \quad (10)$$

$$\sum_{j \in N} y_{ijk} \leq 1, \forall k \in K, \forall i \in N; \quad (11)$$

$$\sum_{j \in N} y_{jik} \leq 1, \forall k \in K, \forall i \in N; \quad (12)$$

$$\sum_{i \in N} y_{ijk} = \sum_{i \in N} y_{jik}, \forall k \in K, \forall j \in D'; \quad (13)$$

$$y_{ijk} + y_{jmk} \leq x_{imk} + 1, \forall k \in K, \forall i \in N, \quad \forall j \in D, \forall m \in N; \quad (14)$$

$$t_{ik} \leq t_{n+i,k}, \forall k \in K, \forall i \in P; \quad (15)$$

$$t_{ik} \leq t_{d,n+i,k}, \forall k \in K, \forall i \in P'; \quad (16)$$

$$td_{ik} \geq t_{ik} - M(1 - \sum_{j \in D'} y_{ijk}), \quad \forall k \in K, \forall i \in N; \quad (17)$$

$$td_{ik} \leq t_{ik} + M(1 - \sum_{j \in D'} y_{ijk}), \quad \forall k \in K, \forall i \in N; \quad (18)$$

$$td_{mk} \geq t_{mk} - M(1 - \sum_{j \in D'} y_{jmk}), \quad \forall k \in K, \forall m \in N; \quad (19)$$

$$td_{mk} \leq t_{mk} + M(1 - \sum_{j \in D'} y_{jmk}),$$

$$\forall k \in K, \forall m \in N; \quad (20)$$

$$Ly_{ijk} + Ry_{jmk} + t_{ik} + t_{im}^T + S_i \leq t_{mk} + M(1 - x_{imk}),$$

$$\forall k \in K, \forall j \in D', \forall i \in N, \forall m \in N; \quad (21)$$

$$td_{jk} \geq td_{ik} + cd_{ij} + L - M(1 - y_{ijk}),$$

$$\forall k \in K, \forall j \in D', \forall i \in N, \forall m \in N; \quad (22)$$

$$td_{jk} + cd_{jm} + S_j + R - M(1 - y_{jmk}) \leq td_{mk},$$

$$\forall k \in K, \forall j \in D', \forall m \in N; \quad (23)$$

$$e_i \leq t_{ik} \leq l_i, \forall k \in K, \forall i \in V; \quad (24)$$

$$e_i \leq td_{ik} \leq l_i, \forall k \in K, \forall i \in D'; \quad (25)$$

$$\sum_{j \in N} (\sum_{m \in V \setminus \{0\}} p_j x_{jmk} + \sum_{m \in V \setminus \{0\}} p_j y_{jmk}) \leq Q,$$

$$\forall k \in K; \quad (26)$$

$$td_{mk} - td_{ik} \leq e + M(2 - y_{ijk} - y_{jmk}),$$

$$\forall k \in K, \forall i \in N, \forall j \in D'. \quad (27)$$

目标函数(1)为最小化路径成本即最小化卡车路径成本和无人机路径成本。约束(2)表示每个取货点都被访问。约束(3)表示对于需求量仅能由卡车满足的送货点,当取货点被访问时,送货点由卡车访问。约束(4)表示对于需求量满足无人机配送的送货点,当取货点被访问时,送货点由卡车或无人机访问。约束(5)表示每辆卡车必须离开配送中心。约束(6)表示每辆卡车必须返回配送中心。约束(7)表示不接受卡车仅在配送中心之间运输。约束(8)~式(9)为去除自循环。约束(10)确保路径连续性。约束(11)~式(12)表示每辆卡车仅携带一架无人机。约束(13)表示无人机每次仅访问一个送货点。约束(14)表示在无人机配送时,没有新的无人机起飞。约束(15)~式(16)表示必须先访问取货点,再访问送货点。约束(17)~式(20)是卡车和无人机的时间同步约束。约束(21)和定义了卡车到达客户点的时间约束。约束(22)~式(23)定义了无人机到达客户点的时间约束。约束(24)~(25)表示时间窗约束。约束(26)表示载重约束。约束(27)为无人机电量约束。

## 2 自适应大邻域搜索算法设计

“卡车+无人机”模式下的带时间窗的取送货车路径问题为 PDPTW 问题的扩展,属于 NP 难问题,精确算法和 CPLEX 等优化求解器难以在可接受的时间内求解该问题的大规模实例。而 ALNS 算法可在较短时间内求解大规模实例,所以本文采用 ALNS 算法进行求解。

ALNS 算法为运用破坏算子和修复算子对解空

间进行搜索的自适应大邻域搜索算法,最早由 Ropke<sup>[13]</sup>在大邻域搜索算法的基础上提出,用以求解带时间窗的取送货车辆路径问题。该算法在搜索过程中根据算子对解的改进情况自适应地选择搜索算子,对当前解改进较多的算子在后续迭代中有更大的概率再次被选中,以达到对解空间的高效搜索进而提高算法的性能。另外,为了跳出局部最优解,该算法还使用了模拟退火新解接受准则。

PDPTW-D 的算法框架如图 1 所示,首先应用贪婪算法构建卡车路线;其次应用 ALNS 最小化车辆数量;再次根据无人机容量约束、时间窗等约束,将送货节点由卡车访问转变为无人机访问以节约路径成本;最后在固定车辆数量的基础上,应用 ALNS 算法最小化路径成本。其中在最小化车辆数量的 ALNS 算法中使用的修复算子是参考 Ropke<sup>[13]</sup>文章中的算子,是只针对纯卡车路径进行优化;在最小化路径成本的 ALNS 算法中使用的卡车优先的贪婪修复算子和卡车优先的后悔值插入修复算子,需要考虑无人机路径。

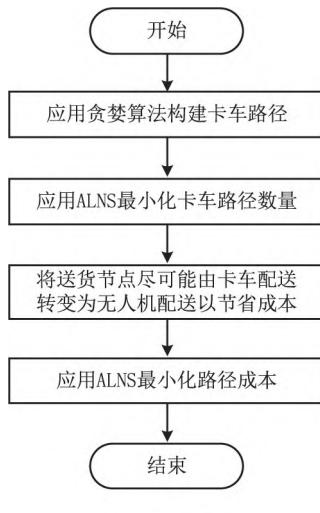


图1 算法框架

ALNS 如算法 1 所示,首先根据算子权重应用轮盘赌的方法选择破坏算子和修复算子对当前解进行破坏和修复,然后应用模拟退火准则判断是否接受新解,最后根据迭代过程中各个算子对解的改进情况更新其权重,直到满足停止准则。每进行一次迭代当前温度  $T$  降为  $T \times \alpha$ ,  $\alpha$  为模拟退火温度冷却速率,算法的停止准则为连续迭代当前解未改进次数、最大总迭代次数和最长运行时间。

#### 算法 1: ALNS 算法框架。

输入: 初始温度  $T_{init}$ ; 当前卡车路线  $S$ ; 冷却速

率  $\alpha$ ;

```

令  $S_{best} = S$ ,  $T = T_{init}$ ,
破坏算子  $\sigma^- = \{1, \dots, 1\}$ , 修复算子  $\sigma^+ = \{1, \dots, 1\}$ ;
do
    根据算子权重利用轮盘赌方法选择破坏算子  $d$ ;
    根据算子权重利用轮盘赌方法选择修复算子  $r$ ;
     $S' = r(d(S))$ ;
    if  $f(S') < f(S_{best})$ 
         $S_{best} = S'$ ; 更新  $\sigma^-$  和  $\sigma^+$  中的算子权重;
    end if
     $T = T \times \alpha$ ;
    while 达到停止准则
        return  $S_{best}$ 
  
```

## 2.1 破坏算子

针对 PDPTW-D 问题的特点,提出相关性破坏算子和随机破坏算子破坏需求。两种算子均为破坏  $q$  个需求,  $q$  为  $1 \sim q_{max}$  之间的随机值。当某一个需求的取货点被破坏时,对应的送货点也被破坏。

### (1) 相关性破坏算子

相关性破坏算子由 SHAW<sup>[20]</sup> 提出,是指通过破坏路径中相似的客户节点来对当前解进行破坏。本文首先随机破坏一个需求,再连续破坏与这个需求相似性最高的  $q - 1$  个需求。文中需求  $i$  与  $j$  的相似性  $R(i, j)$  由距离、时间和装载量 3 方面来决定,其计算公式如下:

$$R(i, j) = \alpha (d_{A(i), A(j)} + d_{B(i), B(j)}) + \beta (|T_{A(i)} - T_{A(j)}| + |T_{B(i)} - T_{B(j)}|) + \gamma |l_i - l_j|,$$

其中  $A(i)$  和  $B(i)$  表示需求  $i$  的取货点和送货点位置,  $A(j)$  和  $B(j)$  表示需求  $j$  的取货点和送货点位置,  $d_{A(i), A(j)}$  和  $d_{B(i), B(j)}$  分别为需求  $i$  和需求  $j$  的取货点和送货点之间的距离,  $T_{A(i)}$  和  $T_{B(i)}$  分别为到达需求  $i$  的取货点和送货点的时间,  $T_{A(j)}$  和  $T_{B(j)}$  分别为到达需求  $j$  的取货点和送货点的时间。 $R(i, j)$  越小,需求  $i$  和需求  $j$  的相似性越高。如果某一个无人机配送的送货点的起飞点或降落点被破坏,则其相应需求的送货点和取货点均被破坏。相关性破坏算子如算法 2 所示:

### 算法 2: 相关性破坏算子。

输入: 当前路线  $S$ , 计划破坏的需求数量  $q$ ;

从  $S$  中随机选取需求  $r$  放入数组  $D$  中,  $D = \{r\}$ ;

while  $|D| < q$

从数组  $D$  中随机选取一个需求  $r$ ;

将  $S$  中剩余需求存放在数组  $L$  中;

根据公式排列数组  $L$ ,  $i < j$  即  $R(r, L[i]) < R(r, L[j])$ ;

从  $[0, 1]$  中随机选择一个数字  $y$ ;

```

D = D ∪ {L[yp | L]};

end while

破坏 D 中所有需求;

if 无人机的起飞点或降落点被破坏
    无人机的访问点及其对应的取货点也被破坏;
end if

return 未被破坏的需求构成的路线 S

```

### (2) 随机破坏算子

随机破坏算子是从路径中随机破坏  $q$  个需求。如果在原路线中无人机的起飞点或降落点被破坏, 则相应的取货点或送货点也应被破坏。相比于相关性破坏算子, 随机破坏算子运行速度更快。虽然可能会出现较差的解, 但有助于搜索的多样性和跳出局部最优解。

## 2.2 修复算子

本文采用卡车优先的贪婪修复算子和卡车优先的后悔值插入修复算子, 具体介绍如下:

### (1) 卡车优先的贪婪修复算子

卡车优先的贪婪修复算子分为两步: 首先将所有被破坏的需求点插入路径由卡车访问, 然后将可由无人机访问的送货点从卡车访问改为无人机访问。定义  $\Delta f_{ik}$  为在路径  $k$  中插入取货点  $i$  及其对应的送货点后目标函数的改变量。在构建卡车访问的路径过程中, 取货点  $i$  及其对应的送货点的位置由  $minc_i$  决定。其中  $c_i = \min_{k \in K} \{\Delta f_{ik}\}$ 。该算子的过程如算法 3 所示。在将由卡车访问的送货节点改为无人机访问的过程中, 首先构建当前求解方案中无人机可以访问的所有送货点的集合  $C$ 。然后随机选择  $C$  中的一个节点  $c$ , 如果  $c$  和其取货点  $c'$  都只与卡车相关, 则根据路径成本判断是否可以使用无人机配送, 并找到这两个客户节点的最佳位置。继续该方法, 直到遍历集合  $C$  中的所有客户节点。检索产生最大节省的排序, 即为当前求解方案。

### 算法 3: 卡车优先的贪婪修复算子。

输入: 当前路线  $S$ ; 未插入的取货点集合  $D$ ;

```

while D ≠ 0 do
    在 D 中随机选择一个取货点 i, D = D \{i\};
    按照 minci, ci = mink ∈ K {Δfik} 插入取货点 i 及其对应的送
    货点;
end while
return S
C = 所有满足无人机容量的送货点;
while C ≠ 0 do

```

```

随机选择 C 中的一个点, C = C \ {c};
if c 点和其对应的取货点 c' 都只与卡车相关, 即不是无人机的起
飞点、降落点和无人机访问点
    S' = S, S = S \ {c, c'}, f(S1) = INF;
    for S 中的每条路径 do
        if 当前路线容量 + qc + qc' < Q
            for 路径上所有相邻的两个节点位置 do
                构建卡车访问 c' 的路径 V = <i, c', i+1>, 无人机访问
                c 的路径 P = <m, c, m+1>, 且 m 点的节点位置在 c' 点
                之后;
                if (c 和 c' 在路径中满足时间窗约束) && (无人机飞行满
                足续航要求) then
                    if 新构建的路径成本 f(S2) < f(S1) then
                        S1 = S2;
                    end if
                end if
            end for
        end if
    end for
    return S1;
    if f(S1) < f(S')
        S = S1;
    else
        S = S';
    end if
end while
return Sbest

```

### (2) 卡车优先的后悔值插入修复算子

该算子的实现分为两步, 第一步以卡车配送的方式插入所有被破坏的取货点和送货点, 定义  $c_i = \Delta f_{i,x_{i2}} - \Delta f_{i,x_{i1}}$  为取货点  $i$  及其对应的送货点插入当前最优路径和次优路径之间的成本差值, 按照  $minc_i$  插入取货点  $i$  及其对应的送货点。第二步再将由卡车配送的送货点改为无人机配送, 该过程与卡车优先的贪婪修复算子中将由卡车访问的节点改为无人机访问的过程相同。

## 2.3 自适应策略

### (1) 算子权重更新

文章中的 ALNS 将当前每种破坏算子和修复算子的权重记为  $w_i$ , 算子权重越大, 说明在以往迭代过程中表现越好, 有更大的可能性探索到更好的解, 所以被选中的概率越大。每次迭代按照  $w_{i,j+1} = (1 - \delta)w_{ij} + \delta\pi_i/\theta_i$  更新权重。其中  $\delta \in [0, 1]$  为权重调整速度系数,  $\pi_i$  为算子  $i$  在上一轮迭代中得到的分数,  $\theta_i$  为上一轮迭代中算子  $i$

被使用的次数。

### (2)选择算子

迭代过程中破坏算子和修复算子的选择使用轮盘赌法则,在每一次迭代中按照概率  $w_j / \sum_{i \in N} w_i$  选择算子  $j$ , 算子的初始权重为 1。

### (3)新解接收准则

为了避免搜索过程陷入局部最优,引入模拟退火接受准则。如果新解优于当前解,则接受。若新解劣于当前解,则根据概率  $e^{-(f(S') - f(S))/T}$  判断是否接受。其中  $T$  为温度,在每次迭代后根据冷却系数降低,  $f(\cdot)$  为目标函数值。随着温度的下降,算法接受当前非改进解的概率随之降低,以此获取局部最优。

## 3 算例分析

为验证问题模型和算法的有效性,本文进行了以下两部分实验:①对比 18 个小规模算例的 CPLEX 和 ALNS 算法的求解结果,证明了模型和算法的有效性;②使用 ALNS 算法对 50 个需求的中规模算例和 100 个需求的大规模算例求解,验证 ALNS 对大规模问题求解的稳定性,分析卡车与无人机联合配送模式对路径成本的影响。

### 3.1 测试算例

由于本文研究的 PDPTW-D 问题没有现成算例,所以在 SACRAMENTO 等<sup>[8]</sup>和 LI 等<sup>[21]</sup>算例的基础上生成了聚类分布 C、随机分布 R 和随机聚类分布 RC3 种类型的算例作为文章的测试算例。算例中客户点的位置和需求量参考文献[8],时间窗参考文献[21]。图 2 为算例生成示意图,图 2a 为 VRP-D 示意图,图 2b 为经过取送货节点配对和添加时间窗后的 PDPTW-D 示意图。在图 2 中,正方形符号表示配送中心,圆圈符号表示送货点,三角形符号表示取货点,图 2a 符号上面的数字表示需求量,图 2b 符号上面的数字( $n, m, t$ )分别表示需求量、时间窗下限和时间窗上限。

### 3.2 测试环境及参数设置

实验的计算机配置为 Inter Core I5-7200U 2.5GHz、4GB 内存,ALNS 使用 C++ 语言编写,IDE 采用 Visio Studio 2019 和 CPLEX 采用

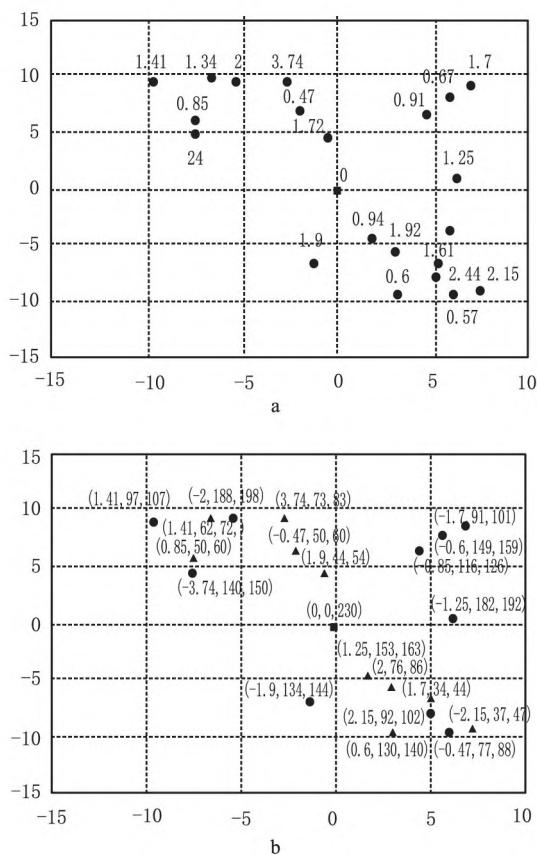


图2 算例生成示意图

CPLEX12.8 默认设置。

本文 ALNS 的相关参数设置为:总迭代次数为 15 000 次,最大未改进的迭代次数为 2 000,算子权重更新准则参数  $\sigma_1 = 33, \sigma_2 = 9, \sigma_3 = 13$ 。模拟退火温度冷却速率  $\alpha = 0.99975$ 。

### 3.3 算例结果

#### 3.3.1 小规模算例求解

由于模型的约束条件和参数过多,即使小规模的算例 CPLEX 求解器也不能全部精确求解,部分算例在有限时间内无法得到最优解,并且在 3 600 秒后解的改进较小,故文章将 CPLEX 求解的运行时间最大限制为 3 600 秒,通过对比 CPLEX 在 3 600 秒内的求解结果和 ALNS 的求解结果,验证 ALNS 算法的有效性。在表 1~表 3 中,AC 表示算例名称,RN 表示路径条数,TC 表示总成本,RT 表示运行时间,Gap 表示 ALNS 算法求解结果与 CPLEX 求解结果的偏差,其值为  $(TC(ALNS) - TC(CPLEX)) / TC(CPLEX)$ 。

表 1 小规模算例求解结果

AC	CPLEX			ALNS			GAP/%
	RN	TC	RT	RN	TC	RT	
lc10-1	2	<b>191.24</b>	0.61	2	<b>191.24</b>	0.33	0
lc10-2	2	<b>172.01</b>	0.78	2	<b>172.01</b>	0.38	0
lr10-1	3	<b>256.96</b>	0.82	3	<b>256.96</b>	0.72	0
lr10-2	2	<b>308.55</b>	1.61	2	<b>308.55</b>	0.56	0
lrc10-1	4	540.67	3 600	4	<b>503.89</b>	1.71	-6.80
lrc10-2	4	428.10	3600	4	<b>377.12</b>	2.67	-11.91
lc15-1	2	<b>284.19</b>	3600	2	<b>284.19</b>	3.78	0
lc15-2	2	<b>438.91</b>	712.81	2	<b>438.91</b>	2.01	0
lr15-1	3	399.22	3 600	3	<b>333.88</b>	4.48	-16.37
lr15-2	4	386.30	3 600	4	<b>333.37</b>	2.74	-13.70
lrc15-1	4	432.55	3 600	4	<b>420.89</b>	3.08	-2.70
lrc15-2	5	568.31	3 600	5	<b>531.19</b>	2.38	-6.53
lc20-1	2	<b>454.85</b>	3 600	2	461.58	3.42	1.48
lc20-2	2	431.81	3 600	2	<b>431.68</b>	4.57	-0.03
lr20-1	5	678.82	3 600	5	<b>548.76</b>	3.67	-19.16
lr20-2	5	581.00	3 600	5	<b>470.31</b>	5.19	-19.05
lrc20-1	7	1043.94	3 600	7	<b>847.77</b>	3.64	-18.79
lrc20-2	7	956.86	3 600	7	<b>779.44</b>	4.46	-18.54

CPLEX 和 ALNS 对 18 个小规模算例求解的结果如表 1 所示,粗体表示两个求解方法对比下更优的运行结果,Gap 为负表示 ALNS 算法求解效果优于 CPLEX。从结果来看,有 11 个算例的 ALNS 结果比 CPLEX 结果更优,成本相差最大达 19.16%;有 6 个算例的 ALNS 结果与 CPLEX 结果相同;仅有 1 个算例的 ALNS 结果略差于 CPLEX。从求解时间来看,ALNS 的运行时间均小于 CPLEX。而 CPLEX 对于 20 个需求在 3 600s 内已经很难得到最优解,而 ALNS 可以在 5 秒内得到被接受的有效解,说明文章提出的问题模型及 ALNS 算法对“卡车+无人机”模式下的取送货路径规划是有效的。

### 3.3.2 中规模与大规模算例求解

为了验证 ALNS 算法对中规模和大规模算例的稳定性,并分析卡车与无人机联合配送模式对路径成本的影响,本文对类型 C、类型 R、类型 RC3 个不同类型、18 组 50 个需求和 100 个需求的算例进行测试,算例结果如表 2 和表 3 所示。

在表 2 和表 3 中, $Z_{IN}$  表示卡车与无人机联合配送模式下的初始解的路径成本, $Z_{PDPTW}$  表示只使用卡车配送的路径成本, $Z_{PDPTW-D}$  表示卡车与无人机联合配送模式下的最终解的路径成本,GAPI 表示与  $Z_{IN}$  相比, $Z_{PDPTW-D}$  节省的路径成本,其值为  $(Z_{IN} - Z_{PDPTW-D}) / Z_{IN}$ ,GAPD 表示与  $Z_{PDPTW}$  相比, $Z_{PDPTW-D}$  节省的路径成本,其值为  $(Z_{PDPTW} - Z_{PDPTW-D}) / Z_{PDPTW}$ 。

表 2 50 组客户点算例求解结果

AC	$Z_{PDPTW}$	$Z_{IN}$	$Z_{PDPTW-D}$	GAPI/%	GAPD/%
lc5_1	306.96	334.79	213.59	36.20	30.42
lc5_2	308.87	253.33	238.15	5.99	22.90
lc5_3	533.63	432.75	351.61	18.75	34.11
lc5_4	509.11	401.22	388.98	3.05	23.60
lc5_5	660.95	635.40	451.76	28.90	31.65
lc5_6	680.31	750.74	485.96	35.27	28.57
lr5_1	328.28	311.25	245.80	21.03	25.12
lr5_2	325.13	351.33	298.02	15.17	8.34
lr5_3	589.74	405.46	303.30	25.20	48.57
lr5_4	592.27	508.03	401.97	20.88	32.13
lr5_5	662.28	706.03	523.64	25.83	20.93
lr5_6	689.47	878.76	634.85	27.76	7.92
lrc5_1	324.99	289.89	267.58	7.70	17.67
lrc5_2	324.92	302.05	262.41	13.12	19.24
lrc5_3	586.11	426.83	359.71	15.73	38.63
lrc5_4	573.41	537.83	394.73	26.61	31.16
lrc5_5	673.99	744.16	482.48	35.16	28.41
lrc5_6	654.64	641.31	581.38	9.34	11.19

从表 2 可以看出 R 类、C 类、RC 类 3 种不同类型的数据的 GAPI 和 GAPD 值大于 0。ALNS 算法将初始解提高 15%~30%，最高将初始解提高了 36.20%，表明 ALNS 可避免局部最优，并且可以获得更优解；采用卡车与无人机联合配送的模式使路径成本节省了 15%~35%，最高节省了 48.57%，表明卡车与无人机联合配送较纯卡车配送可大量节省路径成本，无人机对于中规模的 PDPTW 问题的有效性和必要性。

表 3 100 组客户点算例求解结果

AC	Z <sub>PDPTW</sub>	Z <sub>IN</sub>	Z <sub>PDPTW-D</sub>	GAPI/%	GAPD/%
lc10_1	618.79	484.32	318.85	34.17	48.47
lc10_2	625.86	469.37	382.40	18.53	38.90
lc10_3	849.94	831.39	536.26	35.50	36.91
lc10_4	881.86	531.23	529.49	0.33	39.96
lc10_5	1 303.36	1 354.55	939.58	30.64	27.91
lc10_6	1 270.10	1 198.80	1 095.59	8.61	13.74
lr10_1	653.58	503.07	408.28	18.84	37.53
lr10_2	629.12	503.91	437.12	13.25	30.52
lr10_3	1 004.71	954.70	775.77	18.74	22.79
lr10_4	964.50	695.96	494.30	28.98	48.75
lr10_5	1 348.08	1 217.99	936.20	23.14	30.55
lr10_6	1 310.74	1 222.66	1 091.43	10.73	16.73
lrc10_1	650.91	440.29	419.38	4.75	35.57
lrc10_2	625.76	572.70	420.67	26.55	32.77
lrc10_3	965.61	772.57	543.80	29.61	43.68
lrc10_4	936.99	784.86	725.66	7.54	22.55
lrc10_5	1 371.91	1 257.49	961.66	23.53	29.90
lrc10_6	1 354.43	1 096.57	1 017.47	7.21	24.88

从表 3 可以看出 R 类、C 类、RC 类 3 种不同类型的数据均得出 GAPI 和 GAPD 值大于 0，ALNS 将初始解提高了约 5%~30%，ALNS 最高将初始解提高了 34.17%，表明 ALNS 在避免局部最优和获得更优解的有效性；采用卡车与无人机联合配送的方式节省了 20%~40% 的路径成本，最高节省了 48.75%，表明卡车与无人机联合配送较纯卡车配送可大量节省路径成本，无人机对于大规模的 PDPTW 问题的解决具有有效性和必要性。

通过对类型 C、类型 R、类型 RC3 个不同类型的中规模和大规模算例进行测试求解，可以看出

ALNS 算法将初始解提高了约 15%~30%，算法表现较为稳定，ALNS 可避免局部最优，获得更优解；采用卡车与无人机联合配送的方式节省的路径成本约 20%~35%，表明“卡车+无人机”的运输模式可大量节省路径成本，在 PDPTW 问题中，采用“卡车+无人机”的运输模式能够求解不同类型、不同规模的算例，为实际“卡车+无人机”模式下取送货问题的路径规划提供决策依据。

#### 4 结束语

本文对 PDPTW-D 问题进行了研究，提出了问题的模型，基于模型特点，设计了含有多种高效的破坏算子和修复算子的自适应大邻域搜索算法并进行求解，并利用模拟退火算法防止算法陷入局部最优。通过与 CPLEX 对比小规模算例的运行结果，验证了算法的效率，并通过中规模算例和大规模算例验证了算法的稳定性和有效性。通过结果分析，采用卡车与无人机联合配送的模式可有效降低路径成本，为实际物流网络中提升取送货相关问题的效率提供决策依据。

后续研究问题将考虑货物权重对于卡车运输速率及无人机运送里程的影响，以及软时间窗等因素，丰富问题研究场景。

#### 参考文献：

- [1] LYU Ning. UAV application in logistics industry[J]. Bulletin Of Science Ane Technology, 2021, 37 (2): 106-108 (in Chinese). [吕 宁. 无人机在物流中的应用[J]. 科技通报, 2021, 37(2):106-108.]
- [2] ZHOU Bowen, HUANG Haijun, XU Yi, et al. Parallel task scheduling algorithm based on collaborative and edge in UAV delivery system[J]. Computer Integrated Manufacturing Systems, 2021,27(9):2575-2582(in Chinese). [周博文, 黄海军, 徐 怡, 等. 无人机配送系统中端边协同的并行任务调度算法 [J]. 计算机集成制造系统, 2021,27(9):2575-2582.]
- [3] PAN Nan, CHEN Qiyong, LIU Haishi, et al. Task planning of UAV stocktaking tray in complex industrial storage environment[J]. Computer Integrated Manufacturing Systems, 2021, 27(10):2940-2949(in Chinese). [潘 楠, 陈启用, 刘海石, 等. 复杂工业品仓储环境无人机盘库任务规划[J]. 计算机集成制造系统, 2021,27(10):2940-2949.]
- [4] REN Xuan, HUANG Hui, YU Shaowei, et al. Review on vehicle UAV combined delivery problem[J]. Control and Decision, 2021,36(10):2313-2327(in Chinese). [任璇, 黄辉, 于少伟, 等. 车辆与无人机组合配送研究综述[J]. 控制与决策, 2021,36(10):2313-2327.]

- [5] MURRY C C, CHU A G. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery [J]. *Transportation Research Part C: Emerging Technologies*, 2015, 54: 86–109.
- [6] AGATZ N, BOUMAN P, SCHMIDT M. Optimization approaches for the traveling salesman problem with drone [J]. *Transportation Science*, 2018, 52(4): 965–981.
- [7] WANG X, POIKONEN S, GOLDEN B. The vehicle routing problem with drones: Several worst-case results [J]. *Optimization Letters*, 2017, 11: 679–697.
- [8] SACRAMENTO D, PISINGER D, ROPKE S. An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones [J]. *Transportation Research Part C: Emerging Technologies*, 2019, 102: 289–315.
- [9] MENG Shanshan, GUO Xiuping. Two-stage solution method for truck-drone hybrid delivery [J]. *Industrial Engineering and Management*, 2022, 27(5): 60–68 (in Chinese). [孟姗姗, 郭秀萍. 卡车-无人机混合配送的两阶段求解方法 [J]. 工业工程与管理, 2022, 27(5): 60–68.]
- [10] ZHAO Chang, LIU Yungang, CHEN Lin, et al. Research and development trend of multi-UAV path planning based on metaheuristic algorithm [J]. *Control and Decision*, 2022(5): 1102–1115 (in Chinese). [赵 畅, 刘允刚, 陈 琳, 等. 面向元启发式算法的多无人机路径规划现状与展望 [J]. 控制与决策, 2022(5): 1102–1115.]
- [11] BALDACCI R, BARTOLINI E, MINGOZZI A. An exact algorithm for the pickup and delivery problem with time windows [J]. *Operations Research*, 2011, 59(2): 414–426.
- [12] BENT R, HENTENRYCK P V. A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows [J]. *Computers & Operations Research*, 2003, 33(4): 875–893.
- [13] 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.
- [14] SARTORI C S, BURIOL L S. A study on the pickup and delivery problem with time windows: Matheuristics and new instances [J]. *Computers & Operations Research*, 2020, 124: 105065.
- [15] NACCACHE S, COTE J F, COELHO L C. The multi-pickup and delivery problem with time windows [J]. *European Journal of Operational Research*, 2018, 269(1): 353–362.
- [16] GOEKE D. Granular tabu search for the pickup and delivery problem with time windows and electric vehicles [J]. *European Journal of Operational Research*, 2019, 278(3): 821–836.
- [17] LENG Longlong, ZHAO Yanwei, JIANG Haiqing, et al. Quantum-inspired hyper-heuristics algorithm for low-carbon location-routing problem with simultaneous pickup and delivery [J]. *Computer Integrated Manufacturing Systems*, 2020, 26(3): 815–828 (in Chinese). [冷龙龙, 赵燕伟, 蒋海青, 等. 求解物流配送同时取送货低碳选址—路径问题的量子超启发式算法 [J]. 计算机集成制造系统, 2020, 26(3): 815–828.]
- [18] REN Teng, LUO Tianyu, GU Zhihua, et al. Optimization of urban logistics co-distribution path considering simultaneous pickup and delivery [J]. *Computer Integrated Manufacturing Systems*, 2022, 28(11): 3523–3534 (in Chinese). [任 腾, 罗天羽, 谷智华, 等. 考虑同时取送货的城市物流共同配送路径优化 [J]. 计算机集成制造系统, 2022, 28(11): 3523–3534.]
- [19] XU Qian, XIONG Jun, YANG Zhenhua, et al. Route optimization of takeout delivery vehicles based on adaptive large neighborhood search algorithm [J]. *Industrial Engineering and Management*, 2021, 26(3): 115–122 (in Chinese). [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.]
- [20] SHAW P. A new local search algorithm providing high quality solutions to vehicle routing problems [EB/OL]. [2021-09-26]. <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=8ac9d5287ede9463d33cc0a1daf4191a5e34608d>.
- [21] LI H B, LIM A. A metaheuristic for the pickup and delivery problem with time windows [J]. *International Journal on Artificial Intelligence Tools*, 2003, 12(2): 173–186.

## 作者简介：

吴廷映(1982—),男,苗族,贵州遵义人,讲师,博士,硕士生导师,研究方向:运筹优化、鲁棒优化、智能算法,E-mail: tingyingwu@shu.edu.cn;

陶新月(1997—),女,江苏徐州人,硕士研究生,研究方向:运筹优化,E-mail: taoxinyuetxy@163.com;

孟 婷(1999—),女,安徽寿县人,硕士研究生,研究方向:运筹优化,Email: 2284751678@qq.com。