

DOI:10.3969/j.issn.1003-5060.2026.01.002

## 区域限制下的车机协同路径规划问题

马华伟<sup>1,2</sup>, 陈臻琨<sup>1,2</sup>

(1. 合肥工业大学 管理学院, 安徽 合肥 230009; 2. 过程优化与智能决策教育部重点实验室, 安徽 合肥 230009)

**摘要:**考虑到交通拥堵、空运管制等原因造成的车辆、无人机行驶区域限制以及物流配送中存在的耗时长、效率低等问题,结合无人机和车辆的各自特点,文章采用“车辆+无人机”协同配送模式,针对区域限制下的车机协同路径规划问题,提出一种以最小化配送总时长为目标的两阶段求解算法,第1阶段利用最近邻算法生成初始路径,第2阶段采用改进的亚启发式算法对路径进行优化。引入模拟退火机制,避免该算法陷入局部最优;同时设计多种交叉算子和变异算子,进一步提升算法的全局寻优能力;最后,在算例集上进行多种数值实验,验证所提模型和算法的性能及优越性。结果表明,文章算法能有效解决区域限制下的车机协同路径规划问题。

**关键词:**车机协同;区域限制;路径规划;无人机多点配送;两阶段亚启发式算法

**中图分类号:** TP18 **文献标志码:** A **文章编号:** 1003-5060(2026)01-0013-09

## Vehicle routing problem with drones considering restricted area

MA Huawei<sup>1,2</sup>, CHEN Zhenkun<sup>1,2</sup>

(1. School of Management, Hefei University of Technology, Hefei 230009, China; 2. Key Laboratory of Process Optimization and Intelligent Decision Making, Ministry of Education, Hefei 230009, China)

**Abstract:** Considering the regional restrictions imposed on vehicles and drones due to traffic congestion and air transport control, as well as the issues of prolonged delivery times and low efficiency in logistics distribution, a vehicle-drone cooperative delivery mode is proposed. This mode leverages the unique characteristics of both drones and vehicles. To address the challenge of vehicle-drone cooperative path planning under regional restrictions, a two-stage algorithm is developed with the primary aim of minimizing the total delivery time for vehicle-drone operations. In the first stage, an initial path is generated using the nearest neighbor algorithm, while the second stage employs an improved sub-heuristic algorithm to optimize the path. By integrating simulated annealing mechanisms into the genetic algorithm, the risk of falling into local optima is mitigated. Moreover, various crossover and mutation operators are designed to enhance the global optimization capability of the algorithm. The performance and superiority of the model and algorithm are robustly evaluated through comprehensive numerical experiments on a diverse set of examples. The results demonstrate that the proposed algorithm effectively solves the problem of vehicle-drone cooperative path planning under regional restrictions.

**Key words:** vehicle-drone cooperation; restricted area; path planning; drone delivery for multi-customer; two-stage sub-heuristic algorithm(TSA)

随着物流行业的发展,城市末端物流配送在要求时效性的同时,还要避免道路拥堵带来的挑战。无人机相较于车辆,具有速度快、成本低、无

视地面拥堵的优点。但无人机载重有限,续航短,现实中无法单独完成配送任务,因此将无人机与车辆进行组合来完成城市末端物流配送,不仅能

收稿日期:2024-01-15;修回日期:2024-04-26

基金项目:国家重点研发计划资助项目(2019YFE0110300);国家自然科学基金资助项目(72271076)

作者简介:马华伟(1977—),男,安徽滁州人,博士,合肥工业大学教授,硕士生导师,通信作者,E-mail:colt\_mhw@126.com.

有效地提升效率,还有助于降低配送成本。

无人机与车辆的组合模式分为再供应、非同步和同步模式<sup>[1]</sup>。再供应模式是指车辆或者无人机一方单独承担货物的配送任务,另外一方只是作为辅助工具<sup>[2-3]</sup>。非同步模式是指无人机和车辆各自完成配送服务,在配送过程中没有协同关系。文献[4]最早提出车机组合配送的非同步模式;文献[5]在前者的基础上,考虑了多仓库和多车机的情况,并且假设送货受到时间窗的限制;文献[6]提出动态的非同步模式车机配送模型。同步模式是指车辆与无人机协同完成配送任务,车辆在配送过程中充当无人机的起降平台。文献[4]在提出非同步模式的同时,也提出车机组合配送的同步模式,并构建出同步模式的基础模型,此后众多学者在此基础模型上衍生出许多新变体。文献[7-8]考虑同步模式下车机组合配送过程中一辆车可以携带多架无人机以及多组车机组合的变体问题,并且分别使用启发式算法和精确算法对问题进行求解;文献[9-11]分别提出带有取送货的车机组合配送的新变体问题;文献[12]讨论了在车机组合配送中无人机可以在单个飞行架次中完成对多个节点的访问;文献[13]研究了多辆车和多架无人机的新变种问题;文献[14]讨论了需求可拆分的车机组合配送问题。

目前同步模式通常假定车辆和无人机可以直接到达顾客节点,但现实中可能存在限制区域。限制区域是指车辆或无人机无法顺利通行的区域,例如由于无人机交通管制导致存在禁飞区,或由于道路维修等原因导致存在限行区。限制区域的存在使得以往的模型和算法无法适用,因此衍生出对限制区域下车机协同配送问题的研究。对于限制区域,已有的文献都是基于客户分类进行处理。文献[15]以配送时间最小为目标,建立了车辆搭载无人机单投递的模型,并使用禁忌搜索算法完成对模型的求解;文献[16]建立了一车搭载多架无人机的模型,使用蚁群与模拟退火的融合算法对问题进行求解;文献[17]为满足客户的个性化配送需求,以客户需求为目标,建立了考虑区域限制的车辆与无人机协同配送模型,设计了一种大规模邻域搜索算法与遗传算法的混合算法对模型进行求解。目前对于区域限制的处理都是在地图上划分车辆限行区和无人机禁飞区的位置,位于限行区内的顾客节点车辆不能访问,位于禁飞区内的顾客节点无人机不能访问,因此对于限行区内的顾客节点只能安排无人机进行服务,

禁飞区同理。

虽然对区域限制下车机协同配送问题已有相关研究,但未引起足够关注,并且由于无人机能力的提升,目前可以实现无人机在一个架次中完成多次访问。因此提出区域限制下车机协同路径规划问题(vehicle routing problem with drones considering restricted area, VRPD-RA),建立无人机多架次多投递的模型,并扩展到城市末端配送中形成考虑区域限制的车机同步模式,建立以总配送时间最短为目标的整数规划模型。该模型与以往的车机协同配送模型相比,首次将无人机的多架次多投递模型扩展到区域限制场景下,约束条件更加复杂,需要在区域限制的条件下考虑无人机多投递的问题,并且提出一种独创的两阶段亚启发式算法(two-stage sub-heuristic algorithm, TSA)对问题进行求解。TSA 通过引入模拟退火机制来提升算法的全局寻优能力,避免陷入局部最优的可能性;同时改进交叉算子和变异算子,提升算法的执行效率。

## 1 问题描述

VRPD-RA 示意图如图 1 所示,可以描述为图论问题。

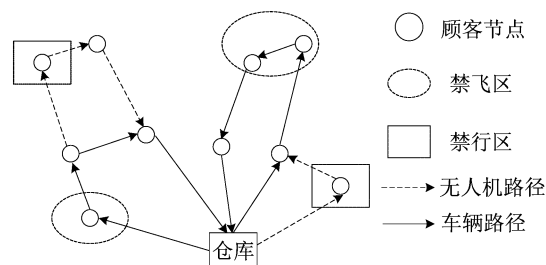


图 1 VRPD-RA 配送路线示意图

在一个完备有向图  $G=(V, A)$  中,  $V=V_C \cup V_0$ ,  $V_0$  表示仓库集,  $V_C$  表示顾客点集,  $A$  为弧集。集合  $V_C$  中共有  $n$  个节点;集合  $V_0$  包括出发点  $O(s)$  与返回点  $O(e)$ ;弧集  $A$  中的每条边  $\langle i, j \rangle$  都有一个非负权重  $d_{ij}$ , 表示从节点  $i$  到节点  $j$  的距离。仓库中有若干同制的车辆,每辆车均搭载一辆相同规格的无人机。车辆的最大载重为  $Q$ ,无人机的最大载重为  $q$ ,单次起飞最大续航时间为  $L_D$ 。若顾客点位于禁飞区内,只能由车辆完成该点的配送;若顾客点位于限行区内,只能由无人机完成该点的配送。VRPD-RA 问题以整体配送时间最短为目标,允许无人机在一次起飞过程中完成多次投递。

为建模需要,作出如下合理假设:

1) 每辆车可以搭载一架无人机,无人机从该车辆起飞后必须回到同一车辆;

2) 无人机有最大续航限制,且最大续航限制已知;

3) 任意节点间距离已知且满足三角形不等式,与限制区不相交时设定为欧氏距离;

4) 无人机只能在顾客节点进行起降,若无人机晚于车辆到达,则车辆必须等待无人机到达后才能继续出发;

5) 无人机的行驶速度快于车辆;

6) 不考虑电池更换、电池充电以及无人机和车辆的装卸货时间。

## 2 模型构建

### 2.1 模型符号

为了方便模型描述,模型用到的符号及其含义见表 1 所示。

表 1 符号及其含义

分类	符号	含义	
集合	$V_C$	顾客节点集合	
	$V_0$	仓库节点集合,包括出发点 $O(s)$ 与返回点 $O(e)$	
	$O(s)$	出发点	
	$O(e)$	返回点	
	$V$	$V=V_C \cup V_0$ , 全部节点集合	
	$K_T$	$K_T=\{1, 2, \dots,  K_T \}$ , 同质车辆集合	
	$V_T$	车辆服务节点集合	
	$V_D$	无人机服务节点集合	
	$V_L$	无人机起飞节点集合	
	$V_R$	无人机降落节点集合	
参数	$R_K$	无人机架次集合	
	$A$	所有弧集合	
	$D_i$	节点需求量	
	$Q$	车辆最大载重	
	$q$	无人机最大载重	
	$L_D$	无人机单次起飞最大续航时间	
	$\Gamma_{ij}^k$	车辆 $k$ 经过弧 $(i, j)$ 的运行时间	
	$\tau_{ij}^k$	车辆 $k$ 搭载的无人机经过弧 $(i, j)$ 的运行时间	
	$T_j^k$	车辆 $k$ 完成节点 $i$ 服务后的累计时间	
	$t_i^k$	车辆 $k$ 搭载的无人机到达节点 $i$ 的累计时间	
决策变量	$\omega_i^k$	车辆 $k$ 与其搭载的无人机在 $i$ 点交互时的等待时间	
	$M$	足够大的正数	
	$S$	绝对值大于等于 2 的数	
	$x_{ij}^k$	0-1 变量,当车辆 $k$ 从节点 $i$ 行驶到节点 $j$ 时值为 1, 否则为 0	
	$y_{pij}^k$	0-1 变量,当车辆 $k$ 搭载的无人机在第 $p$ 个架次中从节点 $i$ 行驶到节点 $j$ 时值为 1, 否则为 0	

### 2.2 模型公式及解释

基于上述符号和变量,可以建立如下的 0-1 型整数线性规划模型。

$$\min Z = \sum_{k \in K_T} \left[ \sum_{i, j \in V, i \neq j} x_{ij}^k (\Gamma_{ij}^k + \omega_i^k) \right] \quad (1)$$

$$\sum_{k \in K_T} \left( \sum_{j \in V_T} x_{ij}^k + \sum_{j \in V_L \cup V_D} y_{pij}^k \right) = 1, \forall i \in V_C \quad (2)$$

$$\sum_{k \in K_T} \sum_{j \in V_C \cup O(s)} x_{O(s)j}^k = \sum_{k \in K_T} \sum_{j \in V_C \cup O(e)} x_{jO(e)}^k \leq |K_T| \quad (3)$$

$$\sum_{k \in K_T} \sum_{i \in V_C \cup O(s)} x_{ij}^k = \sum_{k \in K_T} \sum_{h \in V_C \cup O(s)} x_{jh}^k = 1, \forall j \in V_T \quad (4)$$

$$\sum_{k \in K_T} \sum_{j \in V_C} y_{pij}^k, \forall i \in V_L \cup V_D, \forall p \in R_K \quad (5)$$

$$\sum_{k \in K_T} \sum_{j \in V_C} y_{pij}^k, \forall i \in V_R \cup V_D, \forall p \in R_K \quad (6)$$

$$\sum_{i \in V_L \cup V_D} y_{pij}^k (\sum_{i \in V_C} y_{pij}^k - 1) = 0,$$

$$\forall k \in K_T, \forall j \in V_D \cup V_R, \forall p \in R_K \quad (7)$$

$$\sum_{i \in V_L \cup V_D} y_{pij}^k (\sum_{i \in V_C} y_{pij}^k - 1) = 0,$$

$$\forall k \in K_T, \forall j \in V_D \cup V_R, \forall p \in R_K \quad (8)$$

$$\sum_{i \in V_C} (D_i \sum_{j \in V_T} x_{ij}^k) + \sum_{i \in V_C} (D_i \sum_{p \in R_K} \sum_{j \in V_L \cup V_D} y_{pij}^k) \leq Q, \forall k \in K_T \quad (9)$$

$$\sum_{i \in V_C} (D_i \sum_{j \in V_L \cup V_D} y_{pij}^k) \leq q,$$

$$\forall k \in K_T, \forall p \in R_K \quad (10)$$

$$\sum_{i \in V_C} \sum_{j \in V_C} [y_{pij}^k (\tau_{ij}^k + \omega_i^k)] \leq L_D,$$

$$\forall k \in K_T, \forall p \in R_K \quad (11)$$

$$\sum_{(i, j) \in S, i \neq j} x_{ij}^k \leq |S| - 1,$$

$$\forall S \subseteq V_T, |S| \geq 2, \forall k \in K_T \quad (12)$$

$$\sum_{(i, j) \in S, i \neq j} y_{pij}^k \leq |S| - 1, \forall S \subseteq V(R_K), |S| \geq 2, \forall k \in K_T, \forall p \in R_K \quad (13)$$

$$\sum_{h \in V_C} y_{pih}^k (T_i^k - t_i^k) = 0,$$

$$\forall k \in K_T, \forall i \in V_L, \forall p \in R_K \quad (14)$$

$$\sum_{i \in V_C} y_{pi}^k (T_j^k - t_j^k) = 0,$$

$$\forall k \in K_T, \forall j \in V_R, \forall p \in R_K \quad (15)$$

$$T_j^k \geq T_i^k + \Gamma_{ij}^k - M(1 - x_{ij}^k),$$

$$\forall i, j \in V, \forall k \in K_T \quad (16)$$

$$t_j^k \geq t_i^k + \tau_{ij}^k - M(1 - y_{pij}^k),$$

$$\forall i, j \in V_D, \forall k \in K_T, \forall p \in R_K \quad (17)$$

$$x_{ij}^k, y_{pij}^k \in \{0, 1\} \quad (18)$$

$$T_i^k, t_i^k \geq 0, \forall i \in V, \forall j \in V_T \cup V_D,$$

$$\forall k \in K_T \quad (19)$$

公式(1)为目标函数,表示完成配送任务的车辆总用时最短。

公式(2)为顾客节点流量平衡约束,表示每个顾客节点只能被提供 1 次服务,即无人机和车辆给同一个顾客的送货次数为 1。

公式(3)为车辆仓库节点流量平衡约束,表示从仓库节点出发的车辆数不能超过给定的车辆总数。

公式(4)为车辆顾客节点流量平衡约束。

公式(5)、(6)为无人机起飞节点和降落节点流量平衡约束。

公式(7)、(8)为无人机在提供服务的顾客节点的流量平衡约束。

公式(9)为车辆容量约束,表示车辆服务的顾客以及车辆搭载的无人机服务的顾客的需求量总和不超过车辆容量限制。

公式(10)为无人机架次容量约束,表示在任意一个架次中,无人机服务的顾客的需求量总和不超过无人机容量限制。

公式(11)为无人机架次续航约束,表示无人机在任意架次中行驶时间不能超过无人机续航时间。

公式(12)为车辆子路径消除约束,表示车辆路径中不允许出现子回路。

公式(13)为无人机子路径消除约束,表示无人机路径中不允许出现子回路。

公式(14)、(15)为时间同步约束,表示将车辆和无人机在起飞和降落时的用时调整到相同。

公式(16)、(17)为车辆与无人机的配送时间不等式,表示无人机的起飞和降落需要消耗一个单位时间,该时间分别被添加到车辆和无人机的配送时间中。

公式(18)、(19)为变量的取值约束。

### 3 算法设计

针对协同路径规划的特点,本文设计了一个两阶段亚启发式 TSA 算法对车机协同配送模型进行求解。第 1 阶段基于最近邻思想构造了初始的车机协同路径;第 2 阶段对初始路径进行优化,生成改进后的车机协同路径。本节将介绍初始解构造以及局部搜索算子设计原理。

#### 3.1 初始解构造

基于最近邻思想来生成初始解,通过以下 2 个步骤完成:

1) 构造仅车辆的初始路径,将客户点进行分类,利用最近邻算法将车辆能够访问到的点集规划完毕;

2) 在上一步骤的基础上构造考虑无人机的完整路径,将只能由无人机访问的点按照插入后节省时间的多少排序,依次插入到初始路径中,形成车机协同配送的完整路径。

#### 3.2 局部搜索算子

TSA 算法中引入模拟退火机制,在提高全局寻优能力的同时,降低陷入局部最优的可能性;同时根据车机协同配送的特性,改进局部搜索算子。改进后的局部搜索算子可以针对车机协同配送问题的特点,有效提升算法的效率。

针对车机协同配送的特点,共设计了 6 种局部搜索算子来进行优化,并在个体迭代过程中,使用轮盘赌策略挑选出一个算子进行优化。

1) Vehicle-Exchange 算子。Vehicle-Exchange 算子将分别位于 2 条路径上长度相等的车辆节点弧  $c_i^{+n}$  与弧  $c_j^{+n}$  交换,生成邻域解。Vehicle-Exchange 算子操作示意图如图 2 所示。

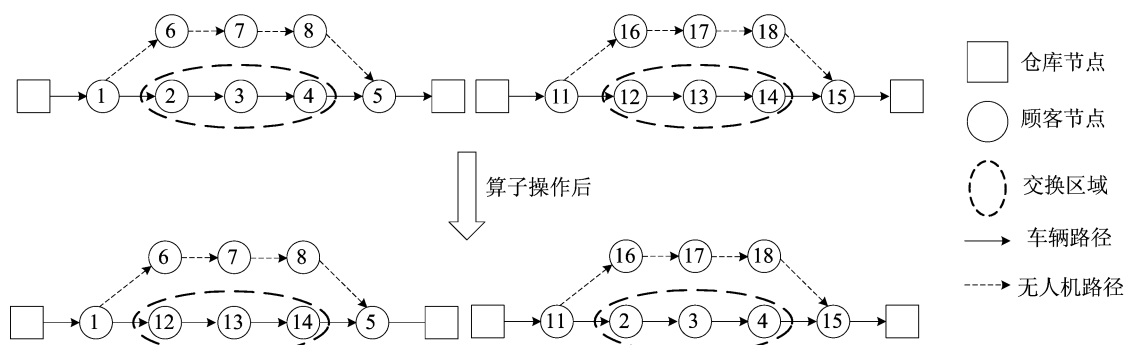


图 2 Vehicle-Exchange 算子操作示意图

2) Drone-Exchange 算子。Drone-Exchange 算子将一条路径上的无人机架次弧  $c_i^l$  与另一条

路径上的无人机架次弧  $c_m^l$  交换,生成邻域解。Drone-Exchange 算子的示意图如图 3 所示。

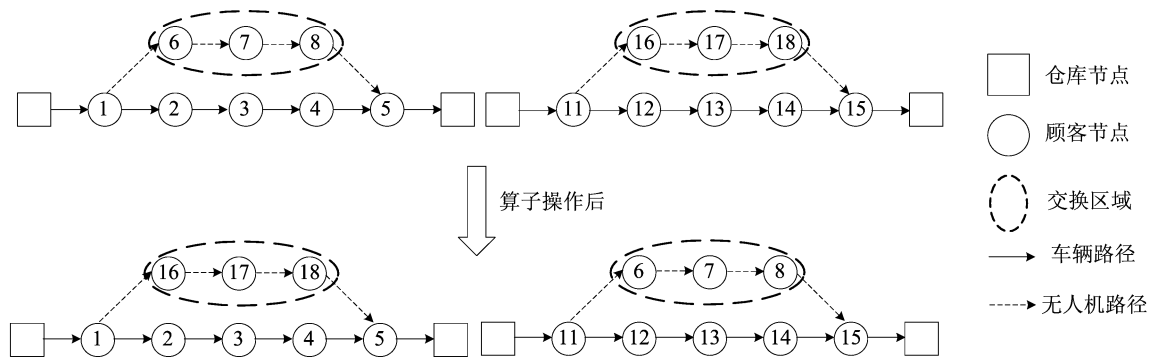


图 3 Drone-Exchange 算子操作示意图

3) Vehicle-Opt 算子。Vehicle-Opt 算子将一条车机协同路径中一段车辆服务节点弧  $c_i^j$  进行逆序操作,生成邻域解。Vehicle-Opt 算子操作示意图如图 4 所示。

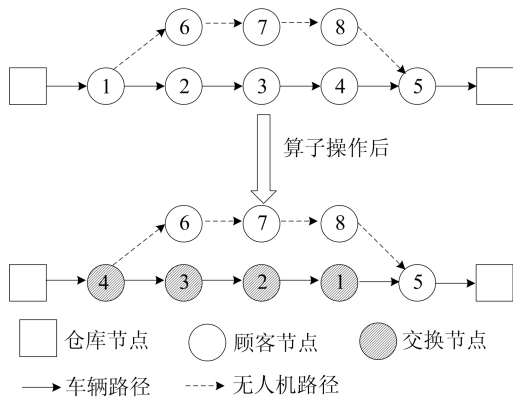


图 4 Vehicle-Opt 算子操作示意图

4) Vehicle-Swap 算子。Vehicle-Swap 算子将同一条车机协同路径上的 2 个不同车辆服务节点交换,生成邻域解。Vehicle-Swap 算子操作示意图如图 5 所示。

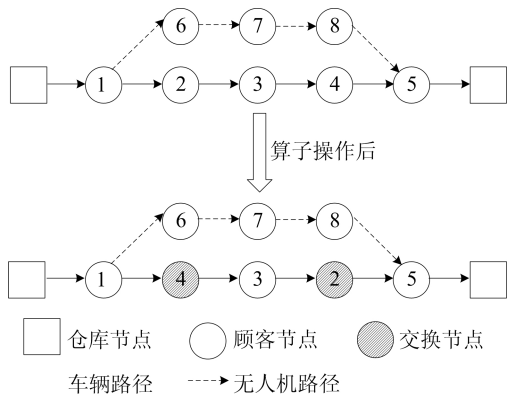


图 5 Vehicle-Swap 算子操作示意图

5) Customer-Swap 算子。Customer-Swap 算子将同一条车机协同路径中的无人机服务节点

与车辆服务节点进行交换。Customer-Swap( $i-j$ )算子表示选中  $i$  个车辆节点和  $j$  个无人机节点,将其进行交换得到新的邻域解。Customer-Swap(1-1)算子操作示意图如图 6 所示。

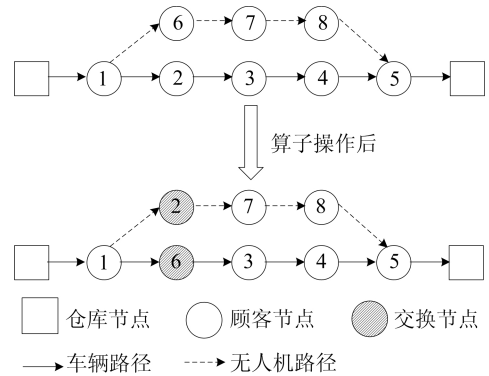


图 6 Customer-Swap 算子操作示意图

6) Node-Move 算子。Node-Move 算子将无人机架次的起飞节点  $d_i$  向后推移到下一个点  $d_{i+1}$  或者是将无人机架次的降落节点提前到上一个点  $d_{i-1}$ ,从而生成邻域解。Node-Move 算子操作示意图如图 7 所示。

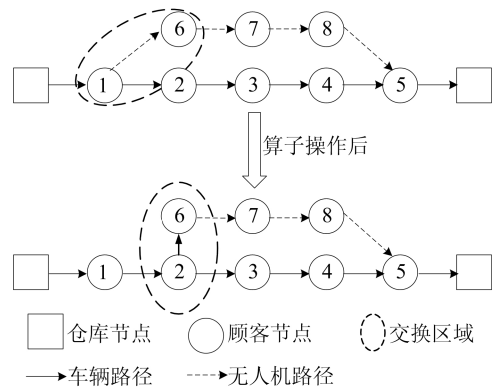


图 7 Node-Move 算子操作示意图

### 3.3 动态调整策略

TSA 算法中共有 6 个邻域搜索算子,在邻域

优化的过程中能否找到最适合的算子对于提升算法求解速度十分有效,因此设计一个动态调整策略用于挑选出最适合的算子。动态调整策略包括 5 个部分,分别是迭代历史的记录、邻域搜索算子的优先级排序、动态调整概率、适应性调整以及重置策略。

在算法开始时,为这 6 个邻域搜索算子设置相同的选择概率;然后在算法执行的过程中,记录每次选择邻域搜索算子后邻域解的质量变化情况,以及当前问题状态和选择的邻域搜索算子等信息,这些信息可以保存在一个历史记录中,用于后续的动态调整;根据历史记录中每个邻域搜索算子的效果,计算每个算子的平均改善程度;对邻域搜索算子进行排序,将效果较好的算子排在前面,效果较差的算子排在后面;根据邻域搜索算子的排序,动态地调整每个算子的选择概率;设置一个动态调整规则,根据每个算子在排序中的位置来确定其选择概率;使效果较好的算子有较高的选择概率,效果较差的算子有较低的选择概率。随着算法的执行,不断地根据历史搜索信息来调整每个邻域搜索算子的选择概率。若某个算子在最近的搜索中表现良好,则可以增加其选择概率;若某个算子长时间没有好的表现,则可以减少其选择概率。此外还要定期重置邻域搜索算子的选择概率,以防止算法陷入局部最优或者过度依赖历史搜索信息。

### 3.4 TSA 算法整体流程

TSA 算法的具体步骤如下所述。

1) 通过对最近邻算法扰动生成初始解集合,对 TSA 算法中所使用到的参数进行初始化。

2) 判断是否达到终止温度。若达到终止温度,则直接输出全局最优解作为最终结果;若没有达到终止温度,则将当前温度下的集合迭代次数设置为 1,进入到下一步骤。

3) 判断集合进化次数是否达到终止进化代数。若达到终止进化代数,则根据降温系数对当前温度进行降温,返回到步骤 2);若没有达到,则进入下一步骤。

4) 从初始种群中挑选出待进化集合,对待进化集合中的每个个体使用动态调整策略,从局部搜索算子中选择一个算法进化。判断进化后的邻域解是否更优,若优于当前解,则用邻域解替代当前解;否则使用 Metropolis 规则进行判断是否接受劣解。当待进化集合中每个个体都进化完毕后,进入下一步骤。

5) 集合进化次数加 1,返回到步骤 3)。

## 4 数值实验

为验证本文 TSA 算法的有效性和可行性,进行参数敏感性分析实验、与 CPLEX 对比实验以及更大规模下的数值实验。由于目前没有针对 VRPD-RA 问题的标准测试数据,在 Augerat 算例集(<http://vrp.atd-lab.inf.puc-rio.br/index.php/en/>)的基础上构建 VRPD-RA 问题的测试集。原有算例集中已有顾客节点的基本信息,在此基础上增加无人机信息,设定无人机速度是车辆速度的 2 倍,无人机续航距离为 120。算法通过 Java 编程实现,运行环境为 3.2 GHz、AMD Ryzen 7 5800H、16 GiB RAM、Windows 11、64 位系统的计算机。

1) 与 CPLEX 软件求解性能的对比实验。为验证 TSA 算法的求解质量,针对 Augerat 算例集,分别使用 TSA 算法和 CPLEX 商业软件进行求解,并比较计算结果。

为保证实验结果的合理性,从算例集的 Set A 和 Set B 中各随机挑选 10 个算例,并从这些算例中节选 10 个客户节点,对顾客节点的位置作一定的简化处理,构建出小规模算例集。规定第 9 个顾客节点处于车辆限行区,第 10 个顾客节点处于无人机禁飞区,其他顾客节点均不受到限制区的影响。将无人机续航设置为 120,最大载重量设置为 50。实验结果见表 1 所列。

表 1 中: $N$  表示算例中顾客点数量; $Q$  表示算例中车辆最大载重; $q$  表示无人机最大载重; $GAP$  表示 TSA 算法与 CPLEX 软件求解的最优解之间的相对误差。

从表 1 可以看出,在全部 20 个算例中,TSA 算法的最优解相对误差不超过 5%,而且 TSA 算法平均求解时间只需要 0.253 s。

在处理小规模问题上,TSA 算法具有很快的求解速度和较高的求解精度;随着问题规模的增大,CPLEX 软件会出现求解时间过长甚至无法求解的情况,而 TSA 算法在处理更大规模的问题时依然适用。

2) 与其他算法求解性能的对比实验。为了验证 TSA 算法在求解更大规模问题上的有效性,将 TSA 算法与文献[15]提出的一种混合禁忌搜索算法(hybrid tabu search with short-path algorithm, SPTS)的求解性能进行对比,实验结果见表 2 所列。

表 1 TSA 算法与 CPLEX 软件求解性能的对比

算例	$N \times Q \times q$	CPLEX 软件		TSA 算法		GAP/%
		最优解	求解时间/s	最优解	求解时间/s	
A1	10×200×50	277.67	6.06	311.47	0.234	12.17
A2	10×200×50	201.56	6.28	204.13	0.218	1.27
A3	10×200×50	268.75	7.27	270.19	0.278	0.54
A4	10×200×50	305.96	11.56	316.79	0.253	3.54
A5	10×200×50	321.90	27.09	326.60	0.268	1.46
A6	10×200×50	264.26	22.56	287.96	0.250	8.97
A7	10×200×50	240.95	7.45	252.97	0.215	4.99
A8	10×200×50	323.42	16.64	343.18	0.218	6.11
A9	10×200×50	292.68	24.74	300.55	0.251	2.69
A10	10×200×50	282.57	5.77	293.48	0.264	3.86
B1	10×200×50	253.21	18.84	266.98	0.284	5.44
B2	10×200×50	238.77	4.67	248.09	0.248	3.90
B3	10×200×50	170.17	16.21	184.40	0.254	8.37
B4	10×200×50	289.00	14.92	303.71	0.266	5.09
B5	10×200×50	256.40	11.43	262.19	0.265	2.26
B6	10×200×50	293.51	17.51	306.75	0.260	4.51
B7	10×200×50	249.46	4.65	264.29	0.261	5.94
B8	10×200×50	251.09	24.17	265.35	0.259	5.68
B9	10×200×50	293.96	13.45	304.00	0.260	3.41
B10	10×200×50	257.69	8.38	264.35	0.256	2.58
平均		266.65	13.48	278.87	0.253	4.58

表 2 TSA 算法与 SPTS 算法求解性能的对比

算例	$N \times Q \times q$	SPTS 算法		TSA 算法		GAP/%
		最优解	求解时间/s	最优解	求解时间/s	
A-n32-k5	32×200×50	523.34	2.303	500.90	1.775	-4.30
A-n33-k5	33×200×50	520.63	1.092	474.67	0.876	-8.80
A-n33-k6	33×200×50	531.98	1.217	502.17	0.909	-5.60
A-n34-k5	34×200×50	585.59	1.140	560.60	0.867	-4.30
A-n37-k6	37×200×50	668.26	1.278	609.42	0.934	-8.80
A-n38-k5	38×200×50	599.50	1.279	550.85	1.116	-8.10
A-n39-k5	39×200×50	610.22	1.638	581.68	1.250	-4.70
A-n45-k7	45×200×50	785.43	1.330	691.71	1.218	-11.90
A-n48-k7	48×200×50	773.16	1.718	702.02	1.382	-9.20
A-n54-k7	54×200×50	862.00	1.615	817.29	1.212	-5.20
A-n61-k9	61×200×50	828.77	1.459	752.10	0.969	-9.30
A-n62-k8	62×200×50	995.15	2.075	904.12	1.522	-9.10
A-n63-k9	63×200×50	1 138.31	1.656	1 049.44	1.085	-7.80
A-n63-k10	63×200×50	956.62	1.445	903.72	1.014	-5.50
A-n65-k9	65×200×50	900.00	1.464	849.43	1.112	-5.60
A-n69-k9	69×200×50	998.23	1.833	890.76	1.609	-10.80
A-n80-k10	80×200×50	1 263.59	2.467	1 176.77	1.704	-6.90
B-n31-k5	31×200×50	341.16	1.500	329.67	1.135	-3.40
B-n34-k5	34×200×50	537.77	1.144	523.10	0.857	-2.70
B-n38-k6	38×200×50	485.79	1.286	455.49	1.044	-6.20
B-n39-k5	39×200×50	397.38	1.753	357.58	1.024	-10.00
B-n41-k6	41×200×50	561.40	1.480	525.26	1.095	-6.40
B-n43-k6	43×200×50	482.38	1.560	468.02	1.233	-3.00
B-n45-k5	45×200×50	559.18	1.838	496.12	1.372	-11.30
B-n45-k6	45×200×50	492.84	1.698	437.92	1.120	-11.10
B-n51-k7	51×200×50	722.39	1.431	657.65	1.060	-9.00
B-n63-k10	63×200×50	981.42	1.408	880.15	1.019	-10.30
B-n64-k9	64×200×50	583.22	1.578	541.11	1.144	-7.20
B-n67-k10	67×200×50	829.27	1.402	729.67	1.133	-12.00
B-n68-k9	68×200×50	855.78	1.775	751.86	1.320	-12.10
平均		712.36	1.560	655.71	1.170	-8.00

表 2 中: $N$  表示算例中顾客点数量; $Q$  表示算例中车辆最大载重; $q$  表示无人机最大载重; $GAP$  表示 TSA 算法与 SPTS 算法求解的最优解之间的相对误差。

表 2 实验是从 Augerat 算例集 Set A 和 Set B 中挑选出 30 个算例组成算例集。对于顾客节点少于 50 的算例,规定第 2 个顾客节点处于车辆限行区,第 5 个顾客节点处于无人机禁飞区;对于顾客节点不少于 50 的算例,规定第 2 个和第 5 个顾客节点处于车辆限行区,第 12 个和第 15 个顾客节点处于无人机禁飞区,并假设其他节点之间的弧不穿过限制区域。

从表 2 可以看出:在全部 30 个算例中,本文提出的 TSA 算法的最优解要优于 SPTS 算法,平均相对误差为  $-8.00\%$ ;TSA 算法的平均求解时间仅为 1.170 s,相较于 SPTS 算法节省了 25%。由此可知,在面对 CPLEX 软件无法顺利求解的

大规模问题时,TSA 算法也能很好地适用。

3) 动态调整策略性能分析实验。为了验证本文提出的动态调整策略能否提高算法求解速度和求解质量,构造一个不使用动态调整策略的 TSA 算法,称为 NTSA(non-dynamic adjustment strategy TSA,NTSA)。从 Augerat 算例集 Set P 中挑选出 10 个算例组成算例集,规定第 2 个和第 5 个顾客节点处于车辆限行区,第 12 个和第 15 个顾客节点处于无人机禁飞区,并假设其他节点之间的弧不穿过限制区域。

为了探究动态调整策略是否能提高算法的求解质量,将 NTSA 与 TSA 的实验结果进行对比,具体见表 3 所列。

表 3 中: $N$  表示算例中顾客点数量; $Q$  表示算例中车辆最大载重; $q$  表示无人机最大载重; $GAP$  表示 TSA 算法与 NTSA 算法求解的最优解之间的相对误差。

表 3 NTSA 算法与 TSA 算法求解性能的对比

算例	$N \times Q \times q$	NTSA 算法		TSA 算法		GAP/%
		最优解	求解时间/s	最优解	求解时间/s	
P-n50-k7	50×200×50	476.09	2.196	472.10	0.978	-0.84
P-n50-k10	50×200×50	502.60	1.685	477.44	0.668	-5.01
P-n55-k7	55×200×50	489.09	1.729	470.87	0.632	-3.73
P-n55-k10	55×200×50	508.85	1.766	496.16	0.691	-2.49
P-n55-k15	55×200×50	531.19	1.695	482.76	0.668	-9.12
P-n60-k10	60×200×50	568.94	1.754	556.46	0.656	-2.19
P-n60-k15	60×200×50	595.11	1.714	556.54	0.654	-6.48
P-n76-k4	76×200×50	686.93	2.006	694.66	0.770	1.12
P-n76-k5	76×200×50	709.26	2.035	694.62	0.761	-2.06
P-n101-k4	101×200×50	851.16	2.999	796.44	1.096	-6.43
平均		591.92	1.958	569.80	0.757	-3.74

从表 3 可以看出:在算例集的 90%算例中,TSA 算法的最优解都优于 NTSA 算法,最优解的 GAP 为  $-3.74\%$ ;并且 TSA 的求解时间仅为 0.757 s,相较于未添加动态调整策略的 NTSA 算法节省 61.3%。

因此,动态调整策略在一定程度上可以提高算法的求解质量。

## 5 结 论

本文研究了区域限制场景下车辆与无人机协同配送的路径规划问题,通过对顾客节点分类的方式对限制区域进行划分,研究结果如下。

1) 以车辆和无人机总配送时间最短为目标,建立 0-1 整数规划模型;设计 TSA 算法,并引入模拟退火机制来提高算法的全局寻优能力,同时也避免了陷入局部最优的可能性。

2) 首次将无人机的多架次多投递模型扩展到区域限制场景,有效地降低了车机协同配送时间;同时改进了交叉算子和变异算子,提升了算法的执行效率。

3) 实验结果表明:TSA 算法在与 CPLEX 软件求解的最优解的相对误差不超过 5%的情况下,求解时间仅为 CPLEX 软件求解时间的 1.85%;同时与 SPTS 算法相比,TSA 算法的求

解结果更优且耗时更短。

综合来看,本文所提出的模型和算法具有可行性和有效性。

本文的研究成果适用于解决区域限制场景下的车机协同路径规划问题,对于可以实现多投递的新型无人机的实际应用有一定的实践意义。

本文仅考虑了客户需求量较小的情况,然而现实中存在客户需求量很大导致无人机无法服务的情况,此时就需要考虑包裹拆分的问题。

此外,本文主要使用启发式算法对模型进行求解,未来还可以进一步考虑使用机器学习等智能方法。

### [参 考 文 献]

- [1] LIANG C, LUO X, CHEN X, et al. Route planning of truck and multi-drone rendezvous with available time window constraints of drones[J]. *Science China Technological Sciences*, 2022, 65(9): 2190-2204.
- [2] WINDRAS MARA S T, RIFAI A P, SLOPHA B M. An adaptive large neighborhood search heuristic for the flying sidekick traveling salesman problem with multiple drops [J]. *Expert Systems with Applications*, 2022, 205: 117647.
- [3] YANPIRAT N, SILVA D F, SMITH A E. Sustainable last mile parcel delivery and return service using drones[J]. *Engineering Applications of Artificial Intelligence*, 2023, 124: 106631.
- [4] MURRAY 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.
- [5] AMINE MASMOUDI M, MANCINI S, BALDACCI R, et al. Vehicle routing problems with drones equipped with multi-package payload compartments [J]. *Transportation Research Part E: Logistics and Transportation Review*, 2022, 164: 102757.
- [6] GU R, LIU Y, POON M. Dynamic truck-drone routing problem for scheduled deliveries and on-demand pickups with time-related constraints[J]. *Transportation Research Part C: Emerging Technologies*, 2023, 151: 104139.
- [7] MBIADOU SALEU R G, DEROUSSI L, FEILLET D, et al. The parallel drone scheduling problem with multiple drones and vehicles[J]. *European Journal of Operational Research*, 2022, 300(2): 571-589.
- [8] ZHOU H, QIN H, CHENG C, et al. An exact algorithm for the two-echelon vehicle routing problem with drones [J]. *Transportation Research Part B: Methodological*, 2023, 168: 124-150.
- [9] 马华伟, 宋洋. 考虑同时取送货的车机协同路径优化问题[J]. *计算机应用研究*, 2023, 40(5): 1335-1340.
- [10] LUO Q, WU G, TRIVEDI A, et al. Multi-objective optimization algorithm with adaptive resource allocation for truck-drone collaborative delivery and pick-up services[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2023, 24(9): 9642-9657.
- [11] 孟姗姗, 郭秀萍. 卡车-无人机联合取送货模式下物流优化[J]. *系统管理学报*, 2022, 31(3): 555-566.
- [12] 马华伟, 马凯, 郭君. 考虑多投递的带无人机车辆路径规划问题研究[J]. *计算机工程*, 2022, 48(8): 299-305.
- [13] 伍国华, 毛妮, 徐彬杰, 等. 基于自适应大规模邻域搜索算法的多车辆与多无人机协同配送方法[J]. *控制与决策*, 2023, 38(1): 201-210.
- [14] 李妍峰, 李佳, 向婷. 需求可拆分的无人机与卡车协同路径优化问题[J]. *工业工程*, 2022, 25(1): 54-63, 143.
- [15] 颜瑞, 陈立双, 朱晓宁, 等. 考虑区域限制的卡车搭载无人机车辆路径问题研究[J]. *中国管理科学*, 2022, 30(5): 144-155.
- [16] 杨雷博, 周俊. 限制区下货车联合无人机配送路径问题研究[J]. *计算机工程与应用*, 2023, 59(12): 326-332.
- [17] ZHOU X, FENG Y. Research on path optimization of vehicle-drone joint distribution considering customer priority [J]. *Complexity*, 2024, 2024: e4933311.

(责任编辑 胡亚敏)