

一种多车辆协同多植保无人机作业路径规划方法

徐金华¹, 汪飞², 韩飞¹, 李岩¹

(1. 长安大学 运输工程学院, 陕西 西安 710064; 2. 中交第一公路勘察设计研究院, 陕西 西安 710075)

摘要: 在植保作业中多车辆协同多无人机能突破续航等限制,可大幅提高作业效率.针对多车辆-多植保无人机作业路径规划求解难度大等问题,建立一种多车辆协同多植保无人机路径优化模型,并以无人机最大作业时间最小化为优化目标.同时,提出改进的simulated annealing-Lin Kernighan Helsgaun (SA-LKH)算法求解该模型.所提算法融合K-均值聚类法和凸包插入法生成初始解,利用LKH算法优化每组无人机和车辆路径.此外,构建有向扰动算子,通过扰动每次迭代过程中作业时间最大和最小的路径来实现算法的快速收敛.5种不同规模算例的结果表明,所提算法在求解质量和效率上均优于对比算法.

关键词: 车辆无人机协同;多无人机路径规划;车辆路径规划;改进模拟退火算法;LKH算法

中图分类号: TB 114.1; S 115 文献标志码: A 文章编号: 1005-3026(2024)02-0296-09

A Method for Path Planning of Multi-vehicles Collaboration with Multi-agricultural UAVs

XU Jin-hua¹, WANG Fei², HAN Fei¹, LI Yan¹

(1. College of Transportation Engineering, Changan University, Xi'an 710064, China; 2. CCCC First Highway Consultants Co., Ltd., Xi'an 710075, China. Corresponding author: LI Yan, E-mail: liyan@chd.edu.cn)

Abstract: Multi-vehicle collaboration with multi-agricultural UAV operation can overcome limitations such as endurance and greatly improve efficiency. To address challenges in path planning for collaborative operations involving multi-vehicles and multi-agricultural UAVs, a path optimization model is constructed with the objective of minimizing the maximum operation time of UAVs. The improved simulated annealing-Lin Kernighan Helsgaun (SA-LKH) algorithm is proposed to solve the model. The proposed method combines K-means and convex hull methods to generate the initial solutions, and uses the LKH algorithm to optimize the paths of each group of UAVs and vehicles. Additionally, a directed perturbation operator is constructed to achieve rapid convergence of the algorithm by perturbing the paths with the maximum and minimum operation times during each iteration. The results of five different scale examples show that the proposed algorithm is superior to the comparative algorithm in terms of solution quality and efficiency.

Key words: vehicle and UAV collaboration; multi-UAV path planning; vehicle path planning; improved simulated annealing algorithm; LKH algorithm

我国农业生产模式正在向智能化、机械化、精准化转变.植保无人机因生产作业效率高、综合应用成本低等优势被广泛用于农药喷洒等作业^[1].面对大规模区域作业,将多无人机与多车辆协同作业可解决无人机负载量较小、飞行距离短等问题,能提高作业效率,此方法被称为车辆

协同植保无人机作业.此种作业方式中,植保无人机在车辆的支持下相互协调配合,共同完成农田覆盖作业任务.

车辆与无人机协同作业问题可看作旅行商问题(traveling salesman problem, TSP)和车辆路径问题(vehicle routing problem, VRP)的结合.在

植保作业场景下,无人机负责作业,车辆仅提供补给,且车辆与无人机可相互独立运动。

车辆和无人机协同问题最先由 Murray 等^[2]提出,该问题被分为飞行助手旅行商问题(flying sidekick traveling salesman problem, FSTSP)和并行无人机调度旅行商问题(parallel drone scheduling traveling salesman problem, PDSTSP)。FSTSP 中一辆卡车搭载一架无人机分别为各自客户节点送货。而 PDSTSP 中卡车与无人机是独立的,各自作业互不影响。FSTSP 和 PDSTSP 中车辆和无人机均能执行作业任务,因此上述模型及其扩展均无法直接应用于本文研究的问题。

车辆辅助无人机作业问题(heterogeneous delivery problem, HDP)^[3]仅研究一辆车支持一架无人机的情况。为允许无人机单次配送多个客户节点, Poikonen 等^[4]定义了 MVDRP (multi-visit drone routing problem)。然而,由于上述模型均需假设车辆应在当前对接点回收完所有无人机后才驶入下一对接点,因而也无法直接应用于车辆协同植保无人机作业问题。Othman 等^[5]在 HDP 模型基础上提出了无人机调度问题(drone scheduling problem, DSP),其允许无人机在车辆完成补给后立即起飞,但其假设车辆行驶路线已知。然而,本文拟研究的问题需在规划无人机路径基础上同步规划车辆路径。

无人机进行农药喷洒时需要指定区域完全覆盖,通常被转换成旅行商问题后求解^[6]。常用求解算法有粒子群算法^[7]、蚁群算法^[8]等。车辆协同无人机路径优化问题具有 NP 难(non-deterministic polynomial-time hard)特性^[9],一般采用多阶段启发式算法求解^[10-11],即在各阶段分别规划车辆或无人机路径后采用动态调整等策略求解最优方案。如 Poikonen 等^[4]提出的三阶段启发式算法,第一阶段构建车辆路径,第二阶段将部分车辆路径转换为无人机路径,最后再优化整体路径。

综上,现有研究大多关注于物流运输,较少针对植保作业场景。本文建立考虑无人机飞行续航与载重关系和车辆容量约束的面向植保作业的多车辆-多无人机路径问题(multi-vehicles supported multi-UAVs routing problem, MVSMURP)模型,并提出求解算法。本文主要贡献为:

1) 提出多车辆-多无人机路径优化问题(MVSMURP),并建立整型规划数学模型,模型考虑无人机的载重与续航之间的动态关系,同时也

允许车辆多次往返于仓库点和作业点之间;

2) 构建了改进 SA-LKH 求解算法,融合 K-均值和凸包插入法生成初始解,并提出有向扰动算子以提高求解效率。

1 问题建模

1.1 问题描述

MVSMURP 可以描述为:安排若干车辆和植保无人机协同完成指定区域的农药喷洒作业,车辆与植保无人机对应组合。每个作业点的作物均需被施药,且不能重复。施药操作仅由无人机执行,车辆只为无人机提供农药和电能补给。车辆仅可在道路上自由移动。无人机在电量即将耗尽或农药耗尽时需降落至道路接受车辆补给。当车辆和无人机会合时,先到达的车辆(无人机)需等待后到达的无人机(车辆)。车辆所载物资不足时需返回仓库补充。所有无人机和车辆均从唯一仓库出发,在完成所有任务后各自返回仓库。图 1 为多车辆协同多植保无人机作业示意图。

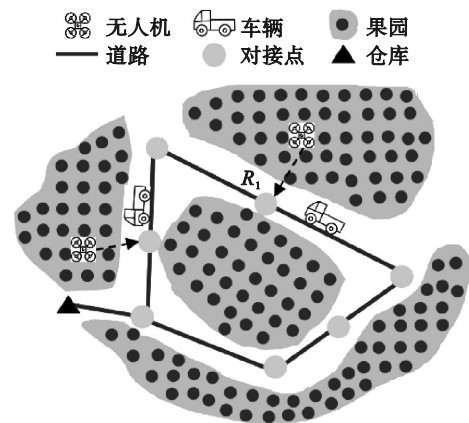


图 1 多车辆协同多植保无人机作业示意图

Fig. 1 Schematic diagram of multi-vehicle collaboration with multi-agricultural UAV operation

1.2 问题假设

本文作出如下假设:

1) 各车辆在道路上自由匀速通行,不考虑车辆在道路会车的影响;

2) 各无人机农药喷洒速度恒定,作业时匀速飞行,仅考虑水平方向运动,垂直方向运动忽略不计;

3) 无人机接受农药或电能补给的耗时忽略不计;

4) 车辆在仓库补充物资过程的耗时忽略不计;

5) 同一对接点允许无人机多架次起降。

1.3 数学建模

为定量描述 MVSMURP, 建立如式(1)~(35) 所示的优化模型, 模型包括目标函数、车辆的路径

约束和补给容量约束, 无人机的路径约束、电量约束和载重约束, 无人机与车辆协同约束以及变量约束等部分. 模型中各符号定义如表 1 所示.

表 1 模型符号定义
Table 1 Definition of model symbols

符号	说明	符号	说明
U	无人机集合, $U = \{1, 2, \dots, u\}$	D_j	路径点 j 农药需求量
B	车辆集合, 车辆与无人机一一对应, $B = \{1, 2, \dots, b\}$, 其中 $b = u$	E_h^u	无人机 u 单位距离能耗
O	果树点坐标集, $O = \{O_1, O_2, \dots, O_n\}$	E_t^u	无人机 u 单位时间能耗
R	对接点集合, $R = \{r_1, r_2, \dots, r_n\}$	f_{\max}	载重惩罚系数最大值
R^0	仓库点集, $R^0 = \{R_0\}$	E_{\max}^u	无人机 u 最大电量
A	所有点的集合, $A = O \cup R \cup R^0$	Q_{\max}^u	无人机 u 最大农药荷载
A_0	起点集合, $A_0 = O \cup R \cup R^0$	E_{\max}^b	车辆 b 所载可为无人机补给的最大电量
A_+	终点集合, $A_+ = O \cup R \cup \{a+1\}$, 其中 $a+1$ 为仓库点	Q_{\max}^b	车辆 b 所载可为无人机补给的最大农药量
R^b	车辆 b 有效对接点集, 即无人机 u 降落的对接点集合	$t_{j,k}^u$	无人机 u 从点 j 出发至离开点 k 的耗时
N_b	车辆 b 的行程数, $N_b = \{1, 2, \dots, n\}$	T_k^u	无人机 u 离开路径点 k 的时刻
m_r^u	无人机 u 在对接点 r 起飞则为 1, 否则为 0	T_k^b	车辆 b 离开路径点 k 的时刻
l_r^u	无人机 u 在对接点 r 降落则为 1, 否则为 0	V_u	无人机 u 飞行速度
E_k^u	无人机 u 离开路径点 k 时剩余电量	V_b	车辆 b 速度
E_j^b	车辆 b 到达点 j 可为无人机补给的剩余电量	F^u	无人机 u 农药喷洒速度
E_s^u	无人机 u 的最低安全电量限值	$x_{j,k}^u$	无人机 u 从点 j 到点 k 为 1, 否则为 0
Q_k^u	无人机 u 离开路径点 k 时剩余农药量	$y_{j,k}^{b,n}$	车辆 b 在行程 n 从点 j 到点 k 为 1, 否则为 0
Q_j^b	车辆 b 到达点 j 时可为无人机补给的剩余农药量	M	大数
$d_{i,j}$	点 i 和点 j 间距离		

1) 目标函数. 根据优化目的, 模型选取无人机最大耗时最小化为目标, 目标函数如式(1)所示.

$$\min Z = \max \{T_{a+1}^u\}_{u \in U}. \quad (1)$$

2) 车辆路径约束.

$$\sum_{i \in R^0 \cup R^b} y_{i,j}^{b,n} = \sum_{i \in R^0 \cup R^b} y_{j,i}^{b,n}, \forall j \in R^0 \cup R^b, \forall b \in B, \forall n \in N_b, \quad (2)$$

$$\sum_{i \in S} \sum_{j \in S} y_{i,j}^{b,n} \leq |S| - 1, \forall b \in B, \forall n \in N_b, \forall S \subseteq R^b, \quad (3)$$

$$\sum_{j \in R^b} y_{R_0,j}^{b,n} = \sum_{k \in R^b} y_{k,R_0}^{b,n} = 1, \forall b \in B, \forall n \in N_b. \quad (4)$$

式(2)表示流量进出平衡, 即车辆到达一个点后必须从这个点离开. 式(3)为支路消除约束. 式(4)表示车辆在每个行程都从仓库出发 1 次并最终仅返回仓库 1 次.

3) 车辆补给容量约束.

$$y_{R_0,j}^{b,n} (E_j^b - E_{\max}^b) = 0, \forall b \in B, j \in R^b, n \in N_b. \quad (5)$$

式(5)表示车辆离开仓库时所载电能为满载.

$$y_{R_0,j}^{b,n} (Q_j^b - Q_{\max}^b) = 0, \forall b \in B, j \in R^b, n \in N_b. \quad (6)$$

式(6)表示车辆离开仓库时所载农药为

满载.

$$y_{j,k}^{b,n} E_k^b \geq E_{\max}^u, \forall b \in B, b = u, \forall j, k \in R^b, n \in N_b. \quad (7)$$

式(7)表示车辆到达任意点时可为无人机补给的电量不少于无人机最大电量.

$$y_{j,k}^{b,n} Q_k^b \geq Q_{\max}^u, \forall b \in B, b = u, \forall j, k \in R^b, n \in N_b. \quad (8)$$

式(8)表示车辆到达任意点时可为无人机补给的农药量不少于无人机最大农药荷载.

$$E_k^b = (E_j^b - E_{\max}^u) y_{j,k}^{b,n},$$

$$\forall b \in B, b = u, \forall j, k \in R^b, n \in N_b. \quad (9)$$

式(9)表示车辆在点 k 所载的无人机电量与在上一个路径点 j 所载量的关系.

$$Q_k^b = (Q_j^b - Q_{\max}^u) y_{j,k}^{b,n},$$

$$\forall b \in B, b = u, \forall j, k \in R^b, n \in N_b. \quad (10)$$

式(10)表示车辆在点 k 所载的农药量与在上一个路径点 j 所载量的关系.

$$(E_j^b - E_{\max}^u) y_{R_0,j}^{b,n} < 0, \forall b \in B, b = u, j \in R^b, n \in N_b. \quad (11)$$

式(11)表示当车辆在点 j 所载无人机电量低于无人机所需最大电量时, 车辆需先返回仓库补充.

$$(Q_j^b - Q_{\max}^u) y_{R_0, j}^{b, n} < 0, \forall b \in B, b = u, j \in R^b, n \in N_b. \quad (12)$$

式(12)表示当车辆在点 j 所载农药量低于无人机最大农药荷载时,车辆需先返回仓库补充.

4) 无人机路径约束.

$$\sum_{j \in O \cup R} x_{R_0, j}^u = \sum_{k \in O \cup R} x_{k, R_0}^u = 1, \forall u \in U. \quad (13)$$

式(13)表示所有无人机仅从仓库出发1次,所有无人机最终仅返回仓库1次.

$$\sum_{i \in O \cup R} x_{i, j}^u = \sum_{i \in O \cup R} x_{j, i}^u, \forall u \in U, \forall j \in O \cup R. \quad (14)$$

式(14)表示出入流量守恒,即当无人机到达某果树点或对接点时,无人机必须从这个点离开.

$$\sum_{u \in U} x_{j, k}^u = 1, \forall j \in A, \forall k \in \{O: j \neq k\}. \quad (15)$$

式(15)保证每棵果树有且仅被覆盖作业一次.

$$x_{j, r}^u l_r^u = x_{r, k}^u m_r^u = 1, \forall u \in U, \forall r \in R, \forall j, k \in O. \quad (16)$$

式(16)表示无人机 u 从果树作业点降落至对接点 r 完成补给后,再次从点 r 起飞至作业区继续作业.

$$\sum_{i \in S} \sum_{j \in S} x_{i, j}^u \leq |S| - 1, \forall u \in U, \forall S \subseteq O \cup R. \quad (17)$$

式(17)为支路消除约束.

5) 无人机电量约束.

$$m_r^u (E_r^u - E_{\max}^u) = 0, \forall r \in R \cup R^0, u \in U. \quad (18)$$

式(18)表示无人机在每次起飞时,其剩余电量为最大电量.

$$E_k^u \geq E_s^u, \forall k \in A, u \in U. \quad (19)$$

式(19)为无人机电量约束,即无人机在每个路径点的剩余电量不能小于最低安全电量限值.其中, E_k^u 计算如下:

$$E_k^u = E_j^u - (E_h^u d_{j, k} x_{j, k}^u + E_t^u t_{j, k}^u x_{j, k}^u) f_p^k(p), \forall j, k \in A, \forall u \in U, \quad (20)$$

$$t_{j, k}^u = \frac{d_{j, k} x_{j, k}^u}{V_u} + \frac{x_{j, k}^u D_k}{F^u}, \forall j, k \in A, \forall u \in U, \quad (21)$$

$$f_p^k(p) = \frac{f_{\max}^k - 1}{Q_{\max}^u} Q_k^u + 1, \forall u \in U. \quad (22)$$

式(20)为无人机 u 离开点 k 时所剩余能量与离开上一路径点 j 时剩余电量的关系;式(21)表示无人机 u 从点 j 出发至离开点 k 的耗时由两部分组成,分别为点 j 与点 k 间飞行运动耗时和点 k 的农药喷洒耗时;式(22)为无人机电载重惩罚系数 $f_p^k(p)$,假设无人机飞行能耗与载重呈现正相关.

$$(E_k^u - E_s^u) x_{k, r}^u l_r^u < 0, r \in R, \forall u \in U. \quad (23)$$

式(23)表示若无人机离开某点时剩余电量小于最低安全电量限值,则降落至对接点.

6) 无人机电载重约束.

$$Q_j^u > 0, j \in A_0, \forall u \in U. \quad (24)$$

式(24)保证无人机 u 在离开任意路径点农药剩余量均不为空.

$$Q_k^u = Q_j^u - x_{j, k}^u D_k, \forall j \in A_0, \forall u \in U. \quad (25)$$

式(25)表示无人机 u 在离开路径点 k 时所剩余农药与离开上一个路径点 j 所剩余农药的关系.

$$(Q_j^u - D_j)(Q_k^u - D_k) x_{j, r}^u l_r^u < 0, D_k, r \in R, \forall j \in A_0, k \in \{A_+: j \neq k\}, \forall u \in U. \quad (26)$$

式(26)表示无人机 u 所载农药量能满足上一个点 j 农药需求,且不能满足下一个点 k 农药需求,则将降落至对接点.

$$m_r^u (Q_r^u - Q_{\max}^u) = 0, \forall r \in R \cup R^0, u \in U. \quad (27)$$

式(27)表示无人机在每次起飞时,其所载剩余农药量为最大量.

7) 无人机与车辆协同约束.

$$l_r^u y_{j, r}^{b, n} = 1, \forall u \in U, u = b, \forall r \in R, \forall j \in R^b \cup R^0, n \in N_b. \quad (28)$$

式(28)表示无人机 u 降落在对接点 r 时,对应的车辆 b 需到此点提供补给.

$$T_r^u \geq T_r^b + M(1 - m_r^u), \forall u \in U, u = b, \forall r \in R. \quad (29)$$

式(29)表示无人机离开对接点 r 的时刻不能早于车辆到达该对接点 r 的时刻,即无人机需与车辆对接后才能起飞离开.

$$T_k^u = T_j^u + t_{j, k}^u x_{j, k}^u, \forall j, k \in A, \forall u \in U, \quad (30)$$

$$T_k^b = T_j^b + \frac{d_{j, k}}{V_b} y_{j, k}^{b, n}, \forall j, k \in R^b \cup R^0, \forall b \in B, n \in N_b, \quad (31)$$

$$T_k^b = \frac{d_{R_0, k}}{V_b} y_{R_0, k}^{b, 1}, \forall k \in R^b, \forall b \in B. \quad (32)$$

式(30)表示无人机 u 离开点 k 的时刻与离开上一路径点 j 时刻的关系;式(31)表示车辆 b 离开点 k 的时刻与离开上一路径点 j 时刻的关系;式(32)表示车辆 b 第1次离开仓库点的时刻为0.

8) 变量约束.

$$y_{i, j}^{b, n} \in \{0, 1\}, \forall b \in B, \forall i, j \in R^0 \cup R^b, n \in N_b, \quad (33)$$

$$x_{i, j}^u \in \{0, 1\}, \forall u \in U, \forall i, j \in A, \quad (34)$$

$$T_{R_0}^u = 0, \forall u \in U. \quad (35)$$

式(33)和式(34)表示决策变量为0-1变量.式(35)表示无人机从仓库出发的起始时刻为0.

2 求解算法

2.1 算法框架

为获取无人机路径和车辆路径,本研究采用

先规划无人机路径,再搜寻车辆路径的思路. 先将各无人机路径规划问题转换为 TSP 求得初步路径. 由于 TSP 问题具有 NP 难特性,而 LKH 算法是目前求解中、大规模 TSP 问题速度最快、精度最高的启发式算法之一^[12],故而采用 LKH 算法优化无人机路径. 为更好地获取全局最优解,将模拟退火算法(simulated annealing, SA)^[13]和 LKH 算法进行结合来求解最优路径.

对于某架无人机,其作业点分布越集中,则其作业路径越短,耗时也就越短. 在 SA-LKH 算法中,首先采用 K-means^[14]划分无人机的初始作业区. 在初始作业区内,通过凸包插入法^[15]生成无人机初始路径,得到初始解. 初始解中各无人机作业耗时可能不均衡,为提高效率,应尽可能地降低耗时最大的无人机作业时长. 因此,本文设计了有向扰动算子,对作业耗时最大和最小的无人机作业路径点进行扰动,加快算法收敛性,避免传统邻域搜索算子的盲目性.

求解流程如图 2 所示,基本步骤如下.

步骤 1 初始化初始温度 T_0 ,同一温度下迭代次数 L ,终止温度 T_c ,衰减因子 α 等参数;

步骤 2 生成初始解,目标函数值 Z ;

步骤 3 对于每组无人机与车辆,先用 LKH 优化无人机路径,再规划车辆路径,并计算目标值 Z' ;

步骤 4 比较目标函数值 $\Delta Z = Z - Z'$,根据 Metropolis 准则判断是否接受新解. 若 $\Delta Z > 0$,接受新解作为当前解;否则,产生区间 $(0, 1)$ 上的均匀分布随机数 τ ,若 $\exp(\Delta Z/T) > \tau$,接受新解为当前解;

步骤 5 若满足终止条件,则进入步骤 7,结

束程序;否则,温度衰减 $T = \alpha T_0$,转至步骤 6;

步骤 6 利用有向扰动算子扰动无人机作业点,返回步骤 3;

步骤 7 输出最终解.

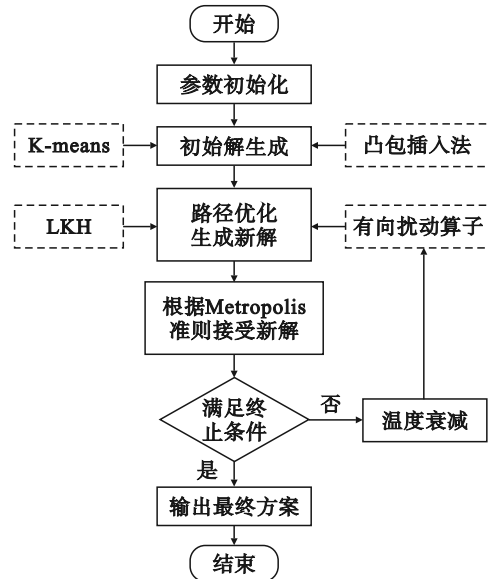


图 2 算法流程图

Fig. 2 Algorithm flow chart

2.2 解的描述

多车辆协同多植保无人机作业路径采用整数编码形式. 图 3 为 2 架无人机和 2 辆车辆路径编码示意图. 无人机从仓库(编号 0)起飞,当电量或农药不足时降落至对接点接受车辆补给,完成补给后继续执行施药任务,施药任务完成后返回仓库. 车辆也从仓库出发,前往对接点提供补给,车辆所载物资不足时返回仓库补充,任务完成后返回仓库. 当所有无人机路径和车辆路径都形成之后才可认为得到一个完整解.

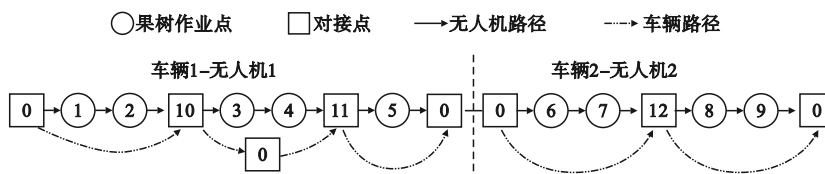


图 3 路径编码示意图

Fig. 3 Schematic diagram of path coding

2.3 初始解生成

首先利用 K-means 法对果树作业点聚类,聚类簇数为无人机个数. 在每个簇团内以凸包插入法求解无人机路径并对路径作可行性检验. 最后,对车辆路径作可行性检验,生成初始解. 具体如下.

步骤 1 初始化聚类簇数 k 为无人机数量,采

用 K-means 法对果树作业点聚类得 k 个区域 $G = \{G_1, G_2, \dots, G_k\}$ 和聚类中心 $\{C_1, C_2, \dots, C_k\}$;

步骤 2 对每架无人机 $u, u \in U$ 执行以下步骤:

步骤 2.1 以仓库点 R_0 为起、终点,利用凸包插入法规划无人机 u 在各区域 g 的路径 $list_g^u$;

步骤 2.2 对路径 $list_g^u$ 作可行性检验,得到

可行路径长度 L_u^g 及无人机 u 降落的对接点 R^u . 计算无人机 u 在各区域 g 的作业耗时 $T_u^g = \frac{L_u^g}{V_u} + \frac{D^g}{F^u}$, 其中, D^g 为区域 g 农药需求总量;

步骤3 将各无人机 u 在各区域 g 作业耗时 T_u^g 按升序排列, 按顺序选出 u 与 g 互不相同的 k 个 T_u^g , 令筛选出的 $T_u^g = T^u$, 确定各无人机 u 所负责的作业区域 g , 以及相应有效对接点集 $R^b, b = u$;

步骤4 对相应的车辆 b 在点集 R^b 中作路径可行性检验, 得车辆路径;

步骤5 输出各无人机作业路径与车辆路径, 得到初始目标函数值 $Z = \max\{T_{a+1}^u\}_{u \in U}$.

2.4 LKH算法

Lin-Kernighan-Helsgaun (LKH) 算法主要对 LK 算法的候选集、初始解、优化规则等方面作出改进^[16]. LKH 算法是在局部搜索 r-opt 算法的基础上进行一系列 k-opt 操作实现. 对于每架无人机执行以下步骤.

步骤1 输入无人机 u 所负责的区域 g 内作业点和仓库点 R_0 , 以基于最小 1-树的 α -measure 为准则生成候选集;

步骤2 依据当前最好的 2 个解及 α -measure 候选集求得初始解 z ;

步骤3 在候选集内通过 LK 算子计算局部最优解 z' ;

步骤4 若 $z' < z$, 则接受该新解, 并将当前最好的 2 个回路的公共边调整到对应顶点的启发集的最前部;

步骤5 重复步骤 2~步骤 4, 直至算法结束, 得到以仓库点 R_0 为起、终点的路径 list_g^u ;

步骤6 对路径 list_g^u 作可行性检验, 并得到相应有效对接点集 $R^b, b = u$;

步骤7 对相应的车辆 b 在点集 R^b 中作路径可行性检验, 得到车辆路径;

步骤8 计算无人机 u 作业耗时 T_{a+1}^u , 求解目标函数值 $Z = \max\{T_{a+1}^u\}_{u \in U}$.

2.5 路径可行性检验

2.5.1 无人机路径可行性检验

无人机路径可行性检验主要是判断路径是否满足电能与农药荷载约束, 若不满足则降落至对接点接受车辆补给, 具体步骤如下.

步骤1 初始化, $i = 1$, 输入无人机 u 路径 list_g^u ;

步骤2 $\text{list}_g^{u,i} = []$, 依次提取路径 list_g^u 中的点 p_j 放入 $\text{list}_g^{u,i}$, 并删除 list_g^u 中的点 p_j , 同时判断无人机 u 在点 p_j 处是否满足 $Q_{p_j}^u > 0$ 且在相邻下一个

点 p_{j+1} 处 $Q_{p_j}^u < 0$ 或判断是否满足 $E_{p_j}^u \geq E_s^u$ 且 $E_{p_{j+1}}^u < E_s^u$, 若满足则在点 p_j 之后插入距点 p_j 最近的对接点 $R^{u,i}$, 即得到路径 $\text{list}_g^{u,i} = [\dots p_j, R^{u,i}]$, 进入步骤 3;

步骤3 $i = i + 1$. 重复步骤 2, 直至 list_g^u 为空. 将 $\text{list}_g^{u,1}, \text{list}_g^{u,2}, \dots, \text{list}_g^{u,i}$ 按顺序拼接即可得到无人机的可行路径以及无人机降落的有效对接点集 $R^b = \{R^{u,1}, R^{u,2}, \dots, R^{u,i}\}, b = u$.

2.5.2 车辆路径可行性检验

车辆路径可行性检验主要是判断路径是否满足车辆荷载约束, 若不满足则需返回仓库, 具体步骤如下.

步骤1 将仓库点 R_0 作为起点, 输入有效对接点集 $R^b = \{R^{u,1}, R^{u,2}, \dots, R^{u,i}\}, b = u$;

步骤2 依次选取 R^b 中的点作为车辆下一路径点, 同时判断当前路径是否满足车辆容量约束, 若不满足, 则在当前有效对接点之前插入点 R_0 , 并进入步骤 3;

步骤3 将当前有效对接点作为起点, 重复步骤 2 直至 R^b 中所有点均已被加入车辆路径点, 在车辆路径的最后插入仓库点 R_0 , 得到车辆可行路径.

2.6 有向扰动算子

步骤1 输入各区域 g 聚类中心 C_g , 各无人机 u 作业耗时 T_u^g ;

步骤2 对于 $\min\{T_u^g\}_{u \in U, g \in G}$ 对应的区域 g , 选取距聚类中心 C_g 最近且不属于区域 g 的 1 个作业点 O_j , 并将该点 O_j 所属类别改为 g ;

步骤3 对于 $\max\{T_u^g\}_{u \in U, g \in G}$ 对应的区域 g , 确定距离聚类中心 C_g 最近的聚类中心 $C_{b, b \neq g, b \in G}$, 选取区域 g 中距离 $C_{b, b \neq g, b \in G}$ 最近的 1 个作业点 O_j , 并将该点 O_j 类别更改为 b ;

步骤4 更新各无人机 u 负责的作业区域.

3 仿真算例

3.1 仿真环境及参数

仿真环境为 Window10 操作系统, 编程语言选择 python, 硬件平台为 Intel(R) Core(TM) i7-9700HQ@3.00 Hz, 内存为 24 GB.

算法参数设置: 初始温度 $T_0 = 200$, 终止温度 $T_c = 0.1$, 衰减因子 $\alpha = 0.9628$.

为验证本文所提算法性能, 选择改进蚁群算法^[6]、改进遗传算法^[17]和改进差分进化算法^[18]作为对比算法, 分别比较各算法目标值结果、运算

时间及收敛性能. 这三种算法为求解路径规划问题的常用方法, 蚁群算法通过信息素来传递信息, 其在路径规划问题方面具有优异性能^[6]. 遗传算法主要是通过染色体交叉或变异操作来搜索解, 这有助于防止在局部最优路径上过早收敛^[17]. 而差分算法保留了遗传算法中基于种群的全局搜索策略, 并采用基于差分的简单变异等操作策略来降低遗传操作的复杂性, 具有较强的全局收敛能力和鲁棒性^[18].

参考文献[6, 17-18]中的参数设置, 各对比算法参数设置为: 改进差分进化算法最大迭代次数为 200, 种群规模为 100, 控制变量 $\lambda=0.5$, 缩放因子 $\beta=0.5$, 交叉概率 $C_r=0.1$; 改进遗传算法最大迭代次数为 200, 种群规模为 100, 变异概率 $P_c=0.05$; 局部搜索的次数为 300; 改进蚁群算法最大迭代次数为 200, 蚂蚁数为 80, 信息启发式系数 $\alpha=1.65$, 期望启发式系数 $\beta=1.4$, 路径选择影响系数 $\gamma=1.4$, 调节系数 $\lambda=1.85$, 信息素蒸发系数 $\rho=0.01$, 任务方向调节系数 $\kappa=0.15$.

为减小随机因素对实验结果的影响, 在每个算例上独立运行 10 次, 取 10 次结果的均值进行比较.

3.2 算例描述

给定 4 架植保无人机(性能参数见表 2)执行农药施药作业. 载重惩罚系数 f_{\max} 为 3^[19]. 假定各车辆速度都为 8 m/s. 随机生成 5 种不同规模作

业点的算例进行仿真验证.

表 2 各型号无人性能参数
Table 2 Performance parameters of various UAVs

型号	E_{\max}/mAh	q_{\max}/L	$V_u/(\text{m}\cdot\text{s}^{-1})$	$F/(\text{L}\cdot\text{s}^{-1})$
UAV ₁	12 000	10	4.5	0.03
UAV ₂	22 000	18	7.3	0.08
UAV ₃	17 000	13	5.8	0.05
UAV ₄	19 000	15	6.4	0.06

3.3 验证实验

各算法求解目标值和运行时间如表 3 所示, 可以看出 SA-LKH 算法在各个规模算例求得的解均优于其他 3 种对比算法. 在 100 个作业点的算例中, 各算法的解较为接近, 这是因为作业点规模较小, 算法求得的可行解都接近于最优解. 相比于改进蚁群算法, 所提算法求解质量仅提升 2.92%~10.17%, 但所提算法的运行时间平均减少约 93.10%. 从表 3 还可以看出, 相比于改进差分进化算法和改进遗传算法, 随着数据规模增加, 本文所提算法求解速度优势有所降低, 但是解的质量明显优于这两种对比算法. 与改进差分进化算法相比, 所提算法求解质量提升 5.67%~21.41%, 求解时间平均减少 35.61%. 与改进遗传算法相比, 所提算法的求解质量平均提升 26.39%, 求解时间平均减少 42.95%. 因此, 所提算法较对比算法具有更好性能.

表 3 各算法结果对比表
Table 3 Comparison of the results of different algorithms

算例 编号	作业点 数量	SA-LKH 算法		改进差分进化算法		改进蚁群算法		改进遗传算法	
		最大作业 时间/s	运行 时间/s	最大作业 时间/s	运行 时间/s	最大作业 时间/s	运行 时间/s	最大作业 时间/s	运行 时间/s
1	100	864.46	5.57	916.44	23.62	890.48	73.34	946.27	29.67
2	263	4 266.03	27.90	5 349.68	67.70	4 418.26	567.32	6 619.09	72.76
3	400	7 138.29	81.85	9 083.24	121.01	7 946.48	1 166.50	9 993.62	137.52
4	573	10 634.24	159.88	12 769.79	175.57	11 164.83	2 281.44	15 580.10	194.77
5	723	14 166.14	354.60	16 600.79	360.16	15 639.37	4 452.37	19 517.02	409.72

为分析各算法求解结果的差异, 采用变异系数(coefficient of variation, CV)来比较各算法所求结果中不同无人机的作业时长. 变异系数越高, 表示变量间的离散程度越大, 反之则越小. 结果如表 4 所示. 可以看出, 改进蚁群算法求得结果的变异系数最高, 表明各无人机间作业时间相差大. 这是因为蚁群算法使用信息素决定移动方向, 在每次迭代过程中较优路径会保留更多的信息素, 而较多的信息素又会吸引更多蚂蚁,

这易陷入局部最优, 且容易出现停滞现象. 改进遗传算法和改进差分进化算法所求结果的变异系数大部分较低, 这表明求得的各无人机作业时间趋于一致, 这是因为这两种算法均通过对个体执行交叉变异等操作来获取较优解, 而这个过程带有随机性, 会使结果向趋同方向演变. 然而, 这两种算法求解的结果差于改进蚁群算法和 SA-LKH 算法, 表明虽然变异交叉操作具有随机性, 但也具有一定的盲目性. SA-LKH 算法求得

无人机间作业耗时的变异系数大于改进遗传算法和改进差分进化算法,但小于蚁群算法的结果,这是因为本文所提算法在每次迭代过程中会从最大和最小两个方向对解进行扰动,使各无人机耗时向趋同方向发展.同时,有向扰动算子使得算法的搜索过程具有方向性,因而求解结果更加接近最优解.

为比较 4 种算法的收敛性能,选取 10 次实验中最优解进行对比.从表 4 和图 4 所示的对比结果可知 SA-LKH 算法的收敛速度最快.原因在于所提算法生成的初始解更接近最优解,并且有向扰动算子可以有方向性地向最优解搜索,加快了算法的收敛速度,相较于对比算法更具有竞争优势.

表 4 各算法对比
Table 4 Comparison of different algorithms

算法	算例	算例 1	算例 2	算例 3	算例 4	算例 5	算例 1	算例 2	算例 3	算例 4	算例 5
		总时长/s	CV/%	总时长/s	CV/%	总时长/s	CV/%	总时长/s	CV/%	总时长/s	CV/%
改进模拟退火 LKH 算法	UAV ₁	855.18		4 225.79		6 655.99		10 570.19		13 985.17	
	UAV ₂	832.86	3.37	3 695.11	6.61	7 137.24	3.14	10 440.38	1.36	13 880.58	0.89
	UAV ₃	801.01		4 263.97		7 069.25		10 313.21		14 166.14	
	UAV ₄	864.46		4 216.39		7 057.66		10 634.24		13 935.80	
改进差分进化算法	UAV ₁	915.40		5 349.68		9 055.40		12 763.99		16 581.56	
	UAV ₂	911.13	0.26	5 270.42	0.70	9 058.35	0.16	12 762.87	0.52	16 574.61	0.08
	UAV ₃	916.44		5 342.99		9 051.01		12 632.48		16 597.31	
	UAV ₄	915.14		5 342.04		9 083.24		12 769.79		16 600.79	
改进蚁群算法	UAV ₁	890.48		3 121.16		7 946.48		8 305.70		15 639.37	
	UAV ₂	890.48	9.59	4 069.66	15.12	7 407.47	4.88	10 006.96	13.06	14 672.47	15.36
	UAV ₃	862.40		4 377.73		7 092.47		11 043.61		12 109.53	
	UAV ₄	722.01		4 418.26		7 309.47		11 164.83		11 286.19	
改进遗传算法	UAV ₁	1 004.83		6 526.52		9 601.96		15 458.32		19 517.02	
	UAV ₂	1 035.91	2.30	6 619.09	1.17	9 993.62	1.77	15 196.03	1.06	19 341.29	0.53
	UAV ₃	1 046.27		6 549.66		9 702.42		15 580.10		19 272.25	
	UAV ₄	997.93		6 433.94		9 866.22		15 471.29		19 378.94	

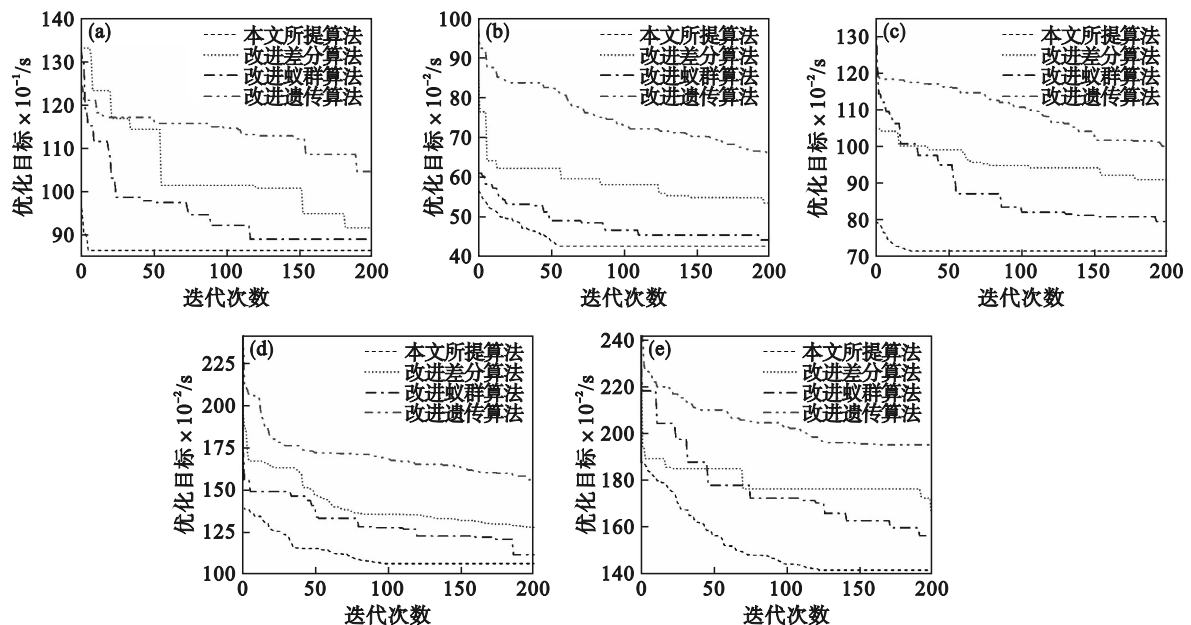


图 4 各算法收敛图

Fig. 4 Convergence diagram of different algorithms

(a)—算例 1; (b)—算例 2; (c)—算例 3; (d)—算例 4; (e)—算例 5.

4 结 论

针对植保作业场景下的特殊性,本文提出一种多车辆协调多植保无人机路径优化数学模型.为高效求解模型,构建了SA-LKH求解算法.该算法融合K-means和凸包插入法生成初始解,并提出有向扰动算子提高求解效率.5种不同规模的仿真算例结果表明,所提算法在可接受时间内能显著提高解的质量.相比于改进蚁群算法、改进差分算法和改进遗传算法,所提算法求解的质量和效率均更优.后续可对含有障碍物区域的协同作业路径优化以及无人机载重等相关作业参数的敏感性进行研究.

参考文献:

- [1] Zhang H J, Qi L, Wu Y L, et al. Numerical simulation of airflow field from a six-rotor plant protection drone using lattice Boltzmann method [J]. *Biosystems Engineering*, 2020, 197: 336-351.
- [2] 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.
- [3] Mathew N, Smith S L, Waslander S L. Planning paths for package delivery in heterogeneous multirobot teams [J]. *IEEE Transactions on Automation Science and Engineering*, 2015, 12(4): 1298-1308.
- [4] Poikonen S, Golden B. Multi-visit drone routing problem [J]. *Computers & Operations Research*, 2020, 113: 104802.
- [5] Othman M S B, Shurbevski A, Karuno Y, et al. Routing of carrier-vehicle systems with dedicated last-stretch delivery vehicle and fixed carrier route [J]. *Journal of Information Processing*, 2017, 25: 655-666.
- [6] Lu L C, Yue T W. Mission-oriented ant-team ACO for min-max MTSP [J]. *Applied Soft Computing*, 2019, 76: 436-444.
- [7] Eldem H, Ulker E. A hierarchical approach based on ACO and PSO by neighborhood operators for TSPs solution [J]. *International Journal of Pattern Recognition and Artificial Intelligence*, 2020, 34(11): 2059039.1-2059039.33.
- [8] 徐久强, 邢佩龙, 孔秋实, 等. 基于改进蚁群算法的双向物流路径优化 [J]. *东北大学学报(自然科学版)*, 2012, 33(9): 1240-1243, 1252.
(Xu Jiu-qiang, Xing Pei-long, Kong Qiu-shi, et al. Logistics routing optimization based on improved ant colony algorithm [J]. *Journal of Northeastern University (Natural Science)*, 2012, 33(9): 1240-1243, 1252.)
- [9] Moshref-Javadi M, Hemmati A, Winkenbach M. A comparative analysis of synchronized truck-and-drone delivery models [J]. *Computers & Industrial Engineering*, 2021, 162: 107648.
- [10] Murray C C, Raj R. The multiple flying sidekicks traveling salesman problem: parcel delivery with multiple drones [J]. *Transportation Research Part C: Emerging Technologies*, 2020, 110: 368-398.
- [11] Dell'Amico M, Montemanni R, Novellani S. Matheuristic algorithms for the parallel drone scheduling traveling salesman problem [J]. *Annals of Operations Research*, 2020, 289(2): 211-226.
- [12] Hu Q R, Lin Z W, Fu J Z. A new global toolpath linking algorithm for different subregions with travelling salesman problem solver [J]. *International Journal of Computer Integrated Manufacturing*, 2022, 35(6): 633-644.
- [13] Kirkpatrick S, Gelatt Jr C D, Vecchi M P. Optimization by simulated annealing [J]. *Science*, 1983, 220(4598): 671-680.
- [14] 朱才华, 孙晓黎, 李岩. 站点分类下的城市公共自行车交通需求预测 [J]. *吉林大学学报(工学版)*, 2021, 51(2): 531-540.
(Zhu Cai-hua, Sun Xiao-li, Li Yan. Forecast of urban public bicycle traffic demand by station classification [J]. *Journal of Jilin University (Engineering and Technology Edition)*, 2021, 51(2): 531-540.)
- [15] 杨毅, 刘亚辰, 刘明阳, 等. 一种基于凸壳的智能服务机器人路径规划算法 [J]. *北京理工大学学报*, 2011, 31(1): 54-58.
(Yang Yi, Liu Ya-chen, Liu Ming-yang, et al. A path planning algorithm based on convex hull for autonomous service robot [J]. *Transactions of Beijing Institute of Technology*, 2011, 31(1): 54-58.)
- [16] Helsgaun K. Solving the equality generalized traveling salesman problem using the Lin - Kernighan - Helsgaun algorithm [J]. *Mathematical Programming Computation*, 2015, 7(3): 269-287.
- [17] Carter A E, Ragsdale C T. A new approach to solving the multiple traveling salesperson problem using genetic algorithms [J]. *European Journal of Operational Research*, 2006, 175(1): 246-257.
- [18] 周辉仁, 唐万生, 王海龙. 基于差分进化算法的多旅行商问题优化 [J]. *系统工程理论与实践*, 2010, 30(8): 1471-1476.
(Zhou Hui-ren, Tang Wan-sheng, Wang Hai-long. Optimization of multiple traveling salesman problem based on differential evolution algorithm [J]. *Systems Engineering: Theory & Practice*, 2010, 30(8): 1471-1476.)
- [19] Song B D, Park K, Kim J. Persistent UAV delivery logistics: MILP formulation and efficient heuristic [J]. *Computers & Industrial Engineering*, 2018, 120: 418-428.