

# 控制与决策

Control and Decision

## 基于APSODE-MS算法的无人机航迹规划

鲁亮亮, 代冀阳, 应进, 赵玉坤

引用本文:

鲁亮亮, 代冀阳, 应进, 赵玉坤. 基于APSODE-MS算法的无人机航迹规划[J]. *控制与决策*, 2022, 37(7): 1695–1704.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2020.1818>

---

## 您可能感兴趣的其他文章

Articles you may be interested in

### 面向多目标侦察任务的无人机航线规划

UAV trajectory planning for multi-target reconnaissance missions

控制与决策. 2021, 36(5): 1191–1198 <https://doi.org/10.13195/j.kzyjc.2019.1284>

### 城市低空环境中多旋翼无人机在线航线规划方法

An online route planning method for multi-rotor drone in urban environments

控制与决策. 2021, 36(12): 2851–2860 <https://doi.org/10.13195/j.kzyjc.2020.0557>

### 无人飞行器航迹方案的VIKOR择优评价

Unmanned aerial vehicle path scheme optimal evaluation based-VIKOR

控制与决策. 2020, 35(12): 2950–2958 <https://doi.org/10.13195/j.kzyjc.2019.0415>

### 基于滚动时域粒子群优化的视频去雾算法

Receding horizon particle swarm optimization based video defogging algorithm

控制与决策. 2021, 36(9): 2218–2224 <https://doi.org/10.13195/j.kzyjc.2019.1183>

### 微型无人机集群低时延组网规划方法

A low delay networking planning method for micro UAV swarm

控制与决策. 2020, 35(11): 2696–2706 <https://doi.org/10.13195/j.kzyjc.2018.1549>

# 基于 APSODE-MS 算法的无人机航迹规划

鲁亮亮, 代冀阳<sup>†</sup>, 应进, 赵玉坤

(南昌航空大学 信息工程学院, 南昌 330100)

**摘要:** 无人机航迹规划是指在环境威胁与自身约束条件下, 规划一条安全可行的航迹, 是实现无人机自主化飞行的关键技术之一. 为实现无人机在不同城市环境下能够快速规划一条安全可靠的航迹, 提出一种基于自适应粒子群差分进化-最小捕捉 (APSODE-MS) 算法的无人机航迹规划方法. 首先, 建立城市环境航迹规划数学模型, 以航程距离、威胁约束、违背约束代价 3 者的加权和作为目标函数; 其次, 在 PSO 算法中引入自适应非线性惯性权重, 根据粒子偏离全局最优解的程度分配不同的搜索模式, 结合动态差分进化 (DE) 算法加快粒子的收敛速度, 引入改进的正态扰动提高跳出停滞与早熟现象的能力; 最后, 筛选关键航迹点, 并采用最小捕捉轨迹 (MS) 算法对航迹进行光滑处理. 仿真结果表明, 所提出的 APSODE-MS 航迹规划方法能够在不同城市仿真环境下较好地完成规划任务, 并能获得更优的航路, 从而验证算法的有效性和鲁棒性.

**关键词:** 航迹规划; 粒子群算法; 惯性权重; 差分进化算法; 最小捕捉算法; 正态扰动

中图分类号: TP273 文献标志码: A

DOI: 10.13195/j.kzyjc.2020.1818

引用格式: 鲁亮亮, 代冀阳, 应进, 等. 基于 APSODE-MS 算法的无人机航迹规划 [J]. 控制与决策, 2022, 37(7): 1695-1704.

## UAV trajectory planning based on APSODE-MS algorithm

LU Liang-liang, DAI Ji-yang<sup>†</sup>, YING Jin, ZHAO Yu-kun

(School of Information Engineering, Nanchang Hangkong University, Nanchang 330100, China)

**Abstract:** Unmanned aerial vehicle (UAV) trajectory planning is to plan a safe and feasible track under the environmental threats and self-constraints. It is one of the key technologies to realize the autonomous flight of an UAV. In order to quickly plan a safe and reliable UAV path in the complex urban environment, this paper presents a hybrid adaptive particle optimization with differential evolution and minimum snap (APSODE-MS) for the UAV path planning in the city. Firstly, this paper establishes a mathematical model for urban environmental trajectory planning, and the weighted sum of flight distance, threat constraint, and violation constraint cost is taken as the objective function. Secondly, the adaptive nonlinear inertia weight is introduced into the PSO algorithm, and different search modes are assigned according to the degree of deviation of the particles from the global optimal solution. The dynamic DE algorithm is used to accelerate the convergence rate of the particles, and the improved normal perturbation is introduced to improve the ability to break out of stagnation and precocity. Finally, the key track points are screened, and the minimum snap (MS) algorithm is used to smooth the track. The simulation results show that the proposed APSODE-MS path planning method can complete the planning task well and obtain a better path in different city simulation environments, thus verifying the effectiveness and robustness of the algorithm.

**Keywords:** trajectory planning; particle swarm optimization; inertia weight; differential evolution; minimum snap; normal disturbance

## 0 引言

无人机航迹规划问题<sup>[1]</sup>(UAV track planning problem, UTPP)是指在复杂的任务环境下,为无人机规划一条由起始点到目标点,且满足无人机飞行性能的最优飞行航迹. 无人机航迹规划是自主飞行的重要环节,在航迹规划过程中,既要考虑任务环境的复

杂性,还要考虑可行的飞行代价以及无人机自身性能的约束条件.

近年来,研究者们针对无人机航迹规划问题的研究主要分为两类:数学计算方法和智能启发方法. 数学计算方法通过建立航迹规划问题的最优化数学模型,采用优化算法对模型进行求解,如整数规划法<sup>[2]</sup>、

收稿日期: 2020-12-27; 录用日期: 2021-04-21.

基金项目: 国家自然科学基金项目 (61663032).

<sup>†</sup>通讯作者. E-mail: djyang@163.com.

梯度法<sup>[3]</sup>、图论方法<sup>[4]</sup>等. 智能启发算法的实质是模拟自然界“优胜劣汰”的思想,在解决复杂、多约束条件的航迹规划问题上有很强的适用性,如A\*搜索算法<sup>[5]</sup>、人工势场法<sup>[6]</sup>、粒子群算法<sup>[7]</sup>、遗传算法<sup>[8]</sup>、差分进化算法<sup>[9]</sup>等. 随着无人机智能化系统的快速发展,人们对无人机的要求也不断提高,尤其是在三维真实环境下的航迹规划方面,使该领域更加注重算法的适应性、收敛性、密切贴合实际等要求. 数学计算方法通过简化三维仿真环境,降低维度进行求解;智能启发方法能够灵活创建三维仿真模型,其中粒子群算法(particle swarm optimization, PSO)简单且不需要变异操作,能够综合考虑威胁区域、飞行性能限制等约束条件,相比于其他算法更易实现. 因此,本文重点研究粒子群算法下的航迹规划.

虽然粒子群算法操作简单易实现,能够解决诸多三维航迹规划问题,但在处理高维复杂的仿真环境问题依然存在如下难点.

1) 如何构建符合实际的目标函数指导粒子向最优解附近进化,种群中的每个粒子作为问题的备选解,粒子的好坏决定了搜索效率以及最终结果. 文献[10]提出了一种最小威胁曲面的方法将三维环境转化为二维投影,简化了模型难度,但未考虑无人机自身的飞行约束问题;文献[11]采用混沌的逻辑映射初始化粒子,增加了粒子的多样性.

2) 在最优解探索过程中,如何选择全局(探索更多最优区域)和局部(全局最优区域寻找最优解)搜索模式,以及如何加快算法收敛速度是要解决的主要问题. 文献[12]采用一种多模态延迟粒子群优化算法,根据粒子的进化状态切换不同的进化模式,有效降低了局部收敛的发生,但算法收敛速度较慢;文献[13-15]通过结合其他优化算法增加算法的多样性,加快粒子的收敛速度,但线性递减的惯性权重与学习因子不能很好地适应复杂的优化问题.

3) 在航迹光滑处理过程中,从航迹中筛选关键航迹点可以有效降低航程距离. 然而,拟合算法,如三次B样条<sup>[16]</sup>、最小二乘法<sup>[17]</sup>等,均采用多点拟合的方法,而关键航迹点带来的航迹点间距差异较大问题将导致航迹凹凸严重,增大与威胁区域碰撞的可能. 因此如何选取关键航迹点,且采用少数航迹点实现航迹的拟合是当前一个难点.

为解决以上难点,本文提出一种基于自适应粒子群差分进化-最小捕捉(adaptive particle optimization and differential evolution-minimum snap, APSODE-MS)算法的无人机航迹规划方法. 具体优点与特征如下.

1) 构建目标函数,本文将航程距离和威胁约束及性能违背约束加权和作为目标函数,根据三者不同的重要程度赋予大小不等的权值. 根据粒子偏离最优解的程度,将粒子划分为开发型、一般型、探索型. “开发型”适合最优解的搜索,“探索型”利于最优解区域的搜索,“一般型”兼顾“开发型”和“探索型”的优点. 通过这种方式可以根据自身特征指导粒子进化.

2) 在粒子进化中,首先根据粒子的类型划分至收敛速度不同的惯性权重函数中,使“开发型”具有良好的局部搜索能力,“探索型”具有良好的全局搜索能力,“一般型”兼顾局部和全局搜索;结合差分进化(DE)算法,采用动态缩放比例因子平衡全局与局部搜索的关系;引入正态扰动,根据迭代次数动态调整大小,使迭代前期能帮助APSODE算法跳出局部最优,后期同APSODE算法进行最优解搜索.

3) 根据无人机靠近障碍物前飞行状态不发生变化的结论,从算法规划的航迹点中选取关键航迹点,根据相邻航迹点的间距计算对应的时间节点,采用最小捕捉轨迹(minimum snap, MS)算法,在位置与速度的约束条件下,求得各个时间段之间的拟合多项式函数.

## 1 航迹规划建模

三维航迹规划是指在特定的飞行区域M内,根据任务信息,规划一条航程最短且满足威胁约束和性能约束的航迹. 本文将航程距离和威胁约束及性能违背约束线性加权和作为对航迹优劣评价的总目标函数及评价机制,函数具体如下:

$$\min f(x) = \omega_1 f_L + \omega_2 f_T + \omega_3 f_C. \quad (1)$$

其中: $f_L$ 、 $f_T$ 、 $f_C$ 分别为航程距离、威胁约束、性能违背约束代价; $\omega_i$ 为对应的影响权值,  $\sum_{i=1}^3 \omega_i = 1$ .

### 1.1 航程代价函数

航程是指无人机由起始点到目标点之间的空间距离,航程大小是评价航迹优劣的关键因素之一. 航程代价函数如下:

$$f_L = \sum_{i=1}^{n-1} a_1 l_i. \quad (2)$$

其中: $a_1$ 为航程代价配平系数, $l_i$ 为第*i*个航迹点到*i+1*个航迹点之间的直线距离, $n$ 为航迹点的个数.

### 1.2 威胁约束惩罚函数

威胁约束是评价航迹可行性的关键因素,如图1所示.

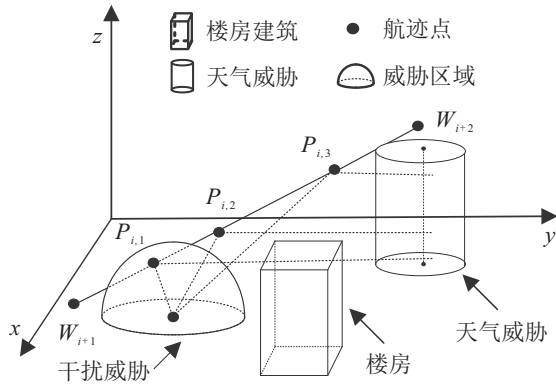


图1 威胁约束计算示意图

以长方体、半球和圆柱体模型分别表示城市环境中楼房建筑、禁飞区域和天气威胁区域. 以第*i*个航段为例, 在  $W_i W_{i+1}$  中等距插入  $k$  个航迹点, 记  $P_i = (p_{i,1}, p_{i,2}, \dots, p_{i,k})$ , 威胁约束惩罚函数为

$$f_{\text{threat}} = \sum_{i=1}^{n-1} \sum_{t=1}^k a_2 \left( \sum_{j=1}^{m_1} f_A(p_{i,t}) + \sum_{j=1}^{m_2} f_C(p_{i,t}) + f_B(p_{i,t}) \right). \quad (3)$$

其中:  $a_2$  为威胁约束配平系数,  $n$  为航迹点的个数,  $m_1$ 、 $m_2$  分别为干扰源和天气威胁的个数,  $k$  为插入的航迹点的个数,  $f_{A_j}(p_{i,t})$ 、 $f_B(p_{i,t})$ 、 $f_{C_j}(p_{i,t})$  分别为禁飞区域、楼房建筑和天气威胁区域的惩罚函数.  $f_{A_j}(p_{i,t})$  为第  $i$  个航迹段第  $t$  个插入点到第  $j$  个禁飞区域的影响程度, 其计算公式如下:

$$f_{A_j}(p_{i,t}) = \begin{cases} 0, & d_j > R_j; \\ K_j(R_j - d_j), & d_j < R_j. \end{cases} \quad (4)$$

其中:  $d_j$  为第  $t$  插入点到第  $j$  个禁飞区域中心点的距离,  $R_j$  为第  $j$  个禁飞区域的影响半径,  $K_j$  为对应的惩罚系数.

楼房建筑采用栅格的方法, 以距离长方体边界 5m 以外的区域作为安全区域.  $f_B(p_{i,t})$  为第  $i$  个航迹段第  $t$  个插入点的影响程度, 其公式如下:

$$f_B(p_{i,k}) = \begin{cases} 0, & C_{\text{index},i,k} \in M; \\ T, & C_{\text{index},i,k} \notin M. \end{cases} \quad (5)$$

其中:  $C_{\text{index},i,k}$  为第  $i$  个航迹段第  $k$  个节点的索引值,  $T$  为低于楼房建筑安全距离的惩罚,  $M$  为安全区域的栅格集合.

$f_{C_j}(p_{i,t})$  为第  $i$  个航迹段第  $t$  个插入点到第  $j$  个天气威胁区域的影响程度, 其计算公式如下:

$$f_{C_j}(p_{i,t}) = \begin{cases} 0, & b_j > r_j; \\ k_j(r_j - b_j), & b_j < r_j. \end{cases} \quad (6)$$

其中:  $b_j$  为第  $t$  插入点到第  $j$  个天气威胁区域中线的

距离,  $r_j$  为第  $j$  个禁飞区域的影响半径,  $k_j$  为对应的惩罚系数.

### 1.3 性能约束惩罚函数

由于无人机自身性能的限制, 要求无人机必须在满足硬性约束条件下才能完成飞行任务. 无人机性能约束主要包括最小航段约束、最大航程约束、飞行高度约束以及最大转弯角约束. 性能约束惩罚函数如下:

$$f_{\text{cst}} = a_3 \sum_{i=1}^q c_i. \quad (7)$$

其中:  $a_3$  为性能约束配平系数,  $c_i$  为第  $i$  个约束的惩罚值.

#### 1.3.1 最小航段约束

无人机在高速飞行情况下, 当需要改变飞行姿态时, 需要预留一段缓冲距离, 要求当前航迹段长度  $l_i$  大于最小航迹段长度  $L_{\min}$ , 约束条件如下:

$$l_i > L_{\min}, \quad i = 1, 2, \dots, n-1. \quad (8)$$

$$C_{1\text{violation}} = \begin{cases} 0, & l_i > L_{\min}; \\ q_1, & l_i \leq L_{\min}. \end{cases} \quad (9)$$

其中:  $n$  为航迹点的个数; 式(9)为当无人机的航迹长度小于最小航迹段长度时的惩罚,  $q_1$  为惩罚大小.

#### 1.3.2 最大航程约束

由于无人机自身携带的能量有限, 无法长距离飞行, 存在最大航程限制, 最大航程约束条件如下:

$$\sum_{i=1}^n l_i \leq L_{\max}. \quad (10)$$

$$C_{2\text{violation}} = \begin{cases} 0, & \sum l_i < L_{\max}; \\ q_2, & \sum l_i \geq L_{\max}. \end{cases} \quad (11)$$

其中:  $L_{\max}$  为最大航迹长度;  $l_i$  为第  $i$  个航迹段的航迹长度; 式(11)为当无人机的航迹长度大于最大航程时的惩罚,  $q_2$  为惩罚大小.

#### 1.3.3 高度约束

无人机在城市间执行运输或救援任务时, 飞行过低会增加与地面或低层建筑碰撞的概率, 而过高则无人机的续航能力会降低, 因此无人机高度约束为

$$h_{\min} \leq z_i \leq h_{\max}. \quad (12)$$

$$C_{3\text{violation}} = \begin{cases} 0, & h_{\min} \leq z_i \leq h_{\max}; \\ q_3, & z_i \leq h_{\min} \text{ OR } z_i \geq h_{\max}. \end{cases} \quad (13)$$

其中:  $z_i$  为第  $i$  个航迹点的高度;  $h_{\min}$ 、 $h_{\max}$  为允许的最小最大高度; 式(13)为当无人机的飞行高度超出或低于高度约束范围时的惩罚,  $q_3$  为惩罚大小.

### 1.3.4 最大转弯角约束

无人机在高速飞行时,无法完成大角度的转弯动作,无人机只能在有限角度范围内转弯,因此最大转弯角约束条件可表示为

$$\frac{a_i a_{i+1}}{\|a_i\| \|a_{i+1}\|} \geq \varphi_{\max}. \quad (14)$$

$$C_{4\text{violation}} = \begin{cases} 0, & \varphi_i \geq \varphi_{\max}; \\ q_4, & \varphi_i < \varphi_{\max}. \end{cases} \quad (15)$$

其中:  $a_i = (x_{i+1} - x_i, y_{i+1} - y_i, z_{i+1} - z_i)$  为第  $i$  个航迹段的向量;  $\varphi_{\max}$  为最大转弯角,  $\varphi_i$  为第  $i$  个航迹段到第  $i+1$  个航迹段的转弯角; 式(15)为当无人机飞行高度超出或低于高度约束范围时的惩罚,  $q_4$  为惩罚大小.

## 2 基于APSO-MS航迹规划

无人机航迹规划模型实质上可以看作是多目标问题的求解. 若采用数学计算的方法求解三维航迹规划模型, 则其建模难度和计算量随着维度的增大呈指数性增长, 因此通常采用智能优化算法进行求解.

传统粒子群算法是通过跟踪个体极值和种群极值的变化实现寻优的, 算法操作简单, 而且有着优秀的收敛性. 然而, 随着迭代次数的增加, 粒子间的相似度逐渐升高, 粒子出现贫化现象, 导致粒子陷入局部最优无法跳出. 对于差分进化算法而言, 在求解目标优化问题时采用交叉、变异以及淘汰机制, 一定程度上考虑了多变量之间的相关性, 有着很强的适用性, 但处理复杂优化问题时, 易出现早熟和搜索停滞现象. 因此, 为提高算法的效率, 增大最优解求解的可能, 采用混合算法APSO-MS对模型进行求解.

### 2.1 自适应粒子群算法(APSO)

PSO算法是一种群智能进化算法, 种群中的每个粒子均可看为一个候选解, 且存储了各个粒子个体最优和全局最优的适应度值  $f_{\text{best}}$ 、 $f_{\text{gbest}}$  以及对应的位置信息  $p_{\text{best}}$ 、 $g_{\text{best}}$ , 粒子通过学习  $p_{\text{best}}$ 、 $g_{\text{best}}$  更新粒子的速度、位置信息. 公式如下:

$$\begin{cases} v_i^{t+1} = \omega v_i^t + c_1 r_1 (p_{\text{best}} - x_i^t) + c_2 r_2 (g_{\text{best}} - x_i^t), \\ x_i^{t+1} = x_i^t + v_i^{t+1}. \end{cases} \quad (16)$$

其中:  $v_i^t$  为第  $i$  个粒子第  $t$  次迭代的速度信息,  $x_i^t$  为粒子对应的位置信息,  $c_1$  和  $c_2$  分别为自身学习因子和群体学习因子,  $r_1$  和  $r_2$  为  $D$  维  $(0,1)$  之间均匀分布的随机数,  $\omega$  为惯性因子.

惯性权重大小表示保持前一时间运动状态的能力. 较大的  $\omega$  利于最优解区域的搜索, 较小的  $\omega$  利于最优解的优化. 目前对惯性权重的改进方法多采

用线性递减<sup>[18]</sup>或分段线性的求解方法, 且只考虑了迭代次数单个影响因素. 然而, 当粒子很快收敛到最优解附近时, 较大的惯性权重并不利于最优解的求解. 而且种群中的每个粒子有着自己的搜索特性, 较大的粒子适合最优解区域的搜索, 较小的粒子适合可行解的优化.

本文按照种群粒子适应度值的大小, 将种群中的粒子划分为“开发型”“一般型”和“探索型”3种类型, “开发型”粒子用于可行解的优化, “探索型”粒子用于探索最优解区域, “一般型”粒子兼顾前面两者的特点. 综合考虑迭代次数和粒子适应度值对最优解求解的影响, 针对不同的搜索类型建立分段非线性惯性权重函数, 函数图像如图2所示.

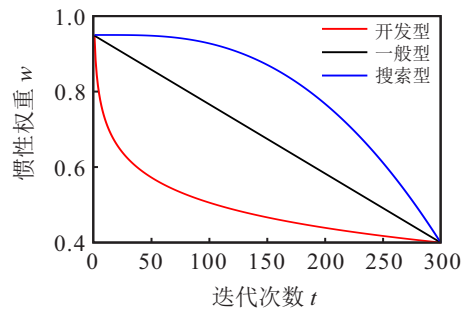


图2 惯性权重变化

“开发型”惯性权重收敛速度最快, “探索型”惯性权重收敛速度最慢, “一般型”采用线性递减的方法. 因此, 相比于惯性权重线性递减的方法, 分段非线性函数更能根据粒子的特性调整惯性权重的大小, 加快粒子的收敛速度. 具体函数如下所示:

1)  $f_{i,\text{best}} < f_{\text{ap}}$  时, 有

$$\omega = \omega_{\max} - (\omega_{\max} - \omega_{\min}) \times \left( \frac{\log_{10} t}{\log_{10} T_{\max}} \right)^{\left( 1 + \frac{f_{i,\text{best}} - f_{\text{gbest}}}{f_{\max} - f_{\text{gbest}}} \right)}. \quad (17)$$

2)  $f_{\text{ap}} < f_{i,\text{best}} < f_{\text{mean}}$  时, 有

$$\omega = \omega_{\max} - (\omega_{\max} - \omega_{\min}) \times \frac{t}{T_{\max}}. \quad (18)$$

3)  $f_{i,\text{best}} > f_{\text{mean}}$  时, 有

$$\omega = \omega_{\max} - (\omega_{\max} - \omega_{\min}) \times \left( \frac{\log_{10} t}{\log_{10} T_{\max}} \right)^{\left( 10 + 5 \cdot \frac{f_{i,\text{best}} - f_{\text{gbest}}}{f_{\max} - f_{\text{gbest}}} \right)}. \quad (19)$$

其中:  $f_{\text{ap}}$  为小于平均适应度值  $f_{\text{mean}}$  的所有粒子的平均值,  $f_{i,\text{best}}$  为第  $i$  个粒子的最优适应度值,  $f_{\text{gbest}}$  为种群最优适应度值,  $f_{\max}$  为种群最大适应度值.

### 2.2 改进差分进化算法

粒子群收敛速度较慢, 且只考虑了粒子与个体最优和种群最优的相关性, 差分进化算法在一定程度上考虑了多个粒子之间的相关性问题, 而且采用交叉、

变异以及淘汰的机制,增加了粒子进化的多样性,改善了全局搜索能力,加快了粒子的收敛速度.差分进化算法通过对个体进化方向的扰动实现个体的变异操作.变异公式如下:

$$u_i^{k+1} = x_{\text{best}}^k + F \times (x_{r_1}^k - x_{r_2}^k). \quad (20)$$

其中:  $x_{r_1}^k$ 、 $x_{r_2}^k$  为第  $k$  次迭代的种群中互不相等的个体,  $r_1$ 、 $r_2 \in \{1, 2, \dots, n\}$  且  $i \neq r_1 \neq r_2$ ;  $x_{\text{best}}^k$  为第  $k$  代种群中最好的个体;  $F$  为缩放比例因子.

缩放因子决定了搜索的步长,但搜索步长过大易导致算法搜索停滞,步长过小将导致算法早熟收敛.对此,本文设计一种动态的收缩因子,使得较大的适应度值具有较强的搜索能力,较小的适应度值具有较强的开发能力.动态收缩因子计算公式如下:

$$F = \begin{cases} \frac{\text{fit}(i) - \text{fit}_{\min}}{\text{fit}_{\max} - \text{fit}_{\min}}, & \text{CR} \geq \frac{\text{fit}(i)}{\text{fit}_{\max}}; \\ 1 - 0.5 \frac{\text{fit}(i)}{\text{fit}_{\max}}, & \text{CR} < \frac{\text{fit}(i)}{\text{fit}_{\max}}. \end{cases} \quad (21)$$

其中:  $\text{CR}$  为交叉因子,  $\text{fit}(i)$  为个体  $i$  对应的适应度值,  $\text{fit}_{\min}$ 、 $\text{fit}_{\max}$  为种群适应度最大值和最小值.

### 2.3 正态扰动

针对 DE 算法易出现搜索停滞和早熟收敛等现象,文献[19]提出了 rand 随机数产生的扰动改变粒子的进化方向的方法,以跳出局部最优.本文利用正态分布产生的随机数扰动增大跳出早熟收敛的能力.正态分布如图3所示.

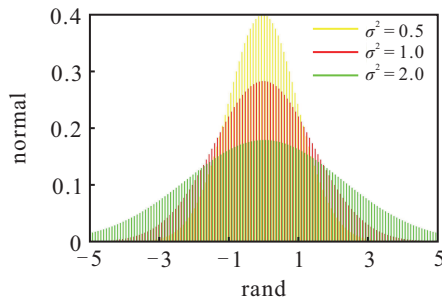


图3 正态函数分布

本文利用适应度值和迭代次数调节函数峰值的大小,使函数在迭代前期或较大适应度值时拥有较大波动,帮助 DE 算法跳出早熟收敛,迭代后期或较小适应度值时拥有较小的波动,帮助实现算法可行解的优化,避免发生停滞现象.正态扰动公式如下:

$$g_i(x) = A \left(1 - \frac{t}{T_{\max}}\right) \cdot r \cdot X \cdot \frac{f_{i,\text{best}}}{f_{\max}}. \quad (22)$$

其中  $X$  和  $\sigma^2$  为

$$X = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad (23)$$

$$\sigma^2 = 10 \times \left(0.01 + 2\frac{t}{T_{\max}}\right).$$

这里:  $A$  为扰动幅值,  $r$  为  $(0,1)$  的随机数,  $x$  为  $D$  维  $(-5, 5)$  的随机数,  $t$  为当前迭代次数,  $T_{\max}$  为最大迭代次数,  $f_{i,\text{best}}$  为第  $i$  个粒子的适应度值,  $f_{\max}$  为种群最大的适应度值.

本文取  $\mu = 0$ , 式(20)经正态扰动为

$$u_i^{k+1} = x_{r_1}^k + F \times (x_{r_2}^k - x_{r_3}^k) + g_{i,1}^k. \quad (24)$$

### 2.4 基于 APSODE 航迹规划算法的步骤

基于 APSODE 算法的航迹规划流程如图4所示,具体步骤如下.

step 1: 设置种群规模  $S$ 、粒子维数  $D$ 、最大迭代次数  $T_{\max}$ , 初始化粒子速度、位置信息;

step 2: 根据式(17)~(19)计算惯性权重,由(16)更新粒子速度和位置;

step 3: 由式(1)计算新粒子的适应度值  $f(p_i)$ , 并与个体最优  $f(p_{i,\text{best}})$  和全体最优  $f(g_{i,\text{best}})$  作比较,更新最优粒子  $p_{i,\text{best}}$ 、 $g_{\text{best}}$  和最优适应度值  $f(p_{i,\text{best}})$ 、 $f(g_{i,\text{best}})$ ;

step 4: 由式(21)更新缩放比例因子  $F$ , 经变异操作生成变异向量  $u_i$ , 交叉因子生成实验向量  $t_i$ ;

step 5: 由式(1)计算粒子  $p_i$  和实验向量  $t_i$  适应度值,选择最优的个体保留至下一代,并将个体最优  $f(p_{i,\text{best}})$  与全体最优  $f(g_{i,\text{best}})$  作比较,选择最优粒子,更新最优粒子  $p_{i,\text{best}}$ 、 $g_{\text{best}}$  和最优适应度值  $f(p_{i,\text{best}})$ 、 $f(g_{i,\text{best}})$ ;

step 6: 判断是否达到最大迭代次数,若达到则输出最优结果,若未达到则返回 step 2.

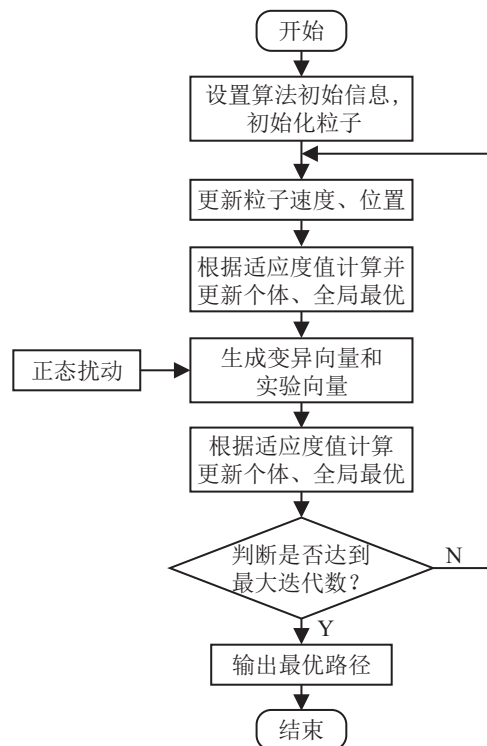


图4 基于 APSODE 算法的航迹规划流程

## 2.5 基于APSO算法的航迹规划伪代码

**算法1** 基于APSO算法的航迹规划伪代码如下。

```

input: population: NP; dimebison: D; genetation: T
output: the best path  $t = 1$ 
while  $t \leq T$ 
  adaptive PSO
  for  $i=1$  to NP do
    update inertia weight
     $\omega_i = \text{adaptive}(t, T, f^t(p_i), f^t_{\max})$ 
    update velocity
     $v_i^{t+1} = \omega_i \cdot v_i^t + c_1 r_1 (p_{\text{best}} - x_i^t) + c_2 r_2 (g_{\text{best}} - x_i^t)$ 
    update position
     $x_i^{t+1} = x_i^t + v_i^{t+1}$ 
    update personal best
    update global best for the generation
  DE algorithm
  normal disturbance
   $X = \text{normality}(t, T, f(p_i), f_{\max}(p))$ 
  mutation and crossover
  for  $j=1$  to D do
     $s_i = \text{mutation}(x_i(t+1));$ 
     $u_{j,i} = \text{crossover}(s_{j,i}, x_{j,i}(t+1))$ 
  end for
  greedy selection
end for
 $t = t + 1;$ 
end while

```

apply MS to the global optimal solution return the best path

## 2.6 关键航迹点的选取与MS算法曲线拟合

智能优化算法规划的航迹由线性航迹段组成,由于无人机自身性能限制,在转折点处无法安全完成飞行动作,需要对航迹点进行光滑处理。由于规划的航迹点往往存在异常或不必要的情况,从中筛选关键航迹点有利于降低航迹的复杂程度和航程。

当无人机在靠近威胁区域前或远离威胁区域时,其飞行姿态不会发生明显的变化。如图5所示,  $P_1 \sim P_6$  为规划的航迹点,将航迹点用直线段首尾相连。

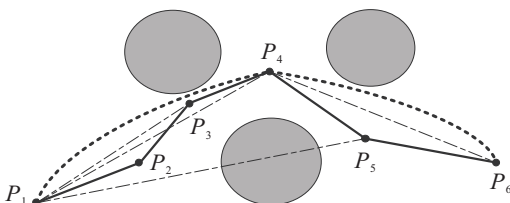


图5 关键航迹点的拟合曲线

从图5中可以看出,起始点  $P_1$  与  $P_2 \sim P_4$  之间以及  $P_4$  与  $P_5, P_6$  之间的连线与障碍物之间没有相交的节点,其他点的连线均与障碍物相交,因此关键航迹点为  $P_1, P_4, P_6$ 。由图5中拟合的虚线可知,关键航迹点的选取可以很大程度上降低航迹的复杂程度和航程。由图5可知,关键航迹点的选取有效降低了航迹的复杂程度和航程,同时也导致航迹点间的间距差异较大。常见的拟合算法,如三次B样条、最小二乘法等,多采用多点拟合方法对航迹进行光滑处理,但面对点与点间距差异较大的情况时,常使得拟合曲线凹凸严重,增加了与障碍物碰撞的可能。文献[20-22]采用MS算法,将航迹点的概念扩展为满足一个或多个约束条件(位置、速度、加速度)的时间点,求取最小捕捉轨迹下的二范数积分,得到各个时间段之间多项式的系数。该算法解决了必须由多个航迹点拟合曲线的问题。

最小捕捉轨迹算法可以看作从起始点  $P_0$  ( $t_0 = 0$ ) 开始,经过  $m$  个航迹点  $P_k$  ( $k \in \{1, 2, \dots, m\}$ ),到达终止点  $P_m$  ( $t_m = T$ ) 的问题。假设无人机在整个航迹规划中速度  $v$  保持不变,则各节点对应的时间为

$$t_k = \frac{\sum_{i=1}^k L_i}{v}, \quad (25)$$

其中  $L_i$  为  $P_{i-1}$  到  $P_i$  的直线距离。

最小捕捉算法针对每个笛卡尔自由度  $x, y$  和  $z$  分别求解。定义  $\sigma_i$  ( $i \in \{x, y, z\}$ ) 为每个状态的轨迹,第  $k$  个时间段  $j$  次前的系数为  $a_{j,k}$ ,则  $m$  段  $n$  项分段多项式如下:

$$\sigma_i = \begin{cases} \sum_{j=0}^n a_{j,1} \cdot t^j, & t_0 < t < t_1; \\ \sum_{j=0}^n a_{j,2} \cdot t^j, & t_1 < t < t_2; \\ \vdots \\ \sum_{j=0}^n a_{j,m} \cdot t^j, & t_{m-1} < t < t_m. \end{cases} \quad (26)$$

设  $f_i(t)$  为第1时间段第  $i$  个状态的多项式函数,多项式系数为  $a_l = [a_{0,l}, a_{1,l}, \dots, a_{n,l}]^T$ ,则多项式函数表达式为

$$f_i(t) = [1, t, t^2, \dots, t^n] \cdot a_l = k^T \cdot a_l. \quad (27)$$

## 3 仿真分析

为验证改进后APSO-MS算法在不同复杂环境下依然均有很好的稳定性和鲁棒性,本文在Matlab 2016a仿真环境下,进行多组仿真实验,实验内容如下。

### 3.1 参数设定

航迹规划的实质是在特定的约束条件下,规划一条由起始点到目标点之间的可行航线. 设置飞行区域为  $M = \{300\text{ m} \times 300\text{ m} \times 100\text{ m}\}$ , 最小飞行距离  $L_{\min} = 10\text{ m}$ , 最大最小飞行高度  $h_{\max} = 50, h_{\min} = 20$ , 飞行速度为  $200\text{ m/s}$ , 最大转弯角  $\varphi_{\max} = 60^\circ$ . 3D 地形中威胁类型包括楼房建筑、禁飞区域和天气威胁. 楼房建筑采用以长宽为  $L$ 、高度为  $H$  的长方体, 禁飞区域采用半径为  $R$  的半球, 天气威胁采用半径为  $R$ 、高度为  $H$  的圆柱. 本文分别进行如下两组实验: 实验1的威胁类型包括楼房建筑和禁飞区域, 实验2的威胁类型包括楼房建筑、禁飞区域和天气威胁. 表1和表2分别记录了两种不同仿真实验环境下的障碍物信息.

表1 实验1威胁参数

威胁类型	坐标/m	最大威胁半径/m
楼房建筑	(25, 65)	$L = 30, H = 60$
	(100, 70)	$L = 40, H = 60$
	(85, 135)	$L = 30, H = 50$
	(155, 135)	$L = 30, H = 50$
	(125, 185)	$L = 30, H = 50$
禁飞区域	(150, 50, 0)	$R = 50$
	(50, 140, 0)	$R = 40$

表2 实验2威胁参数

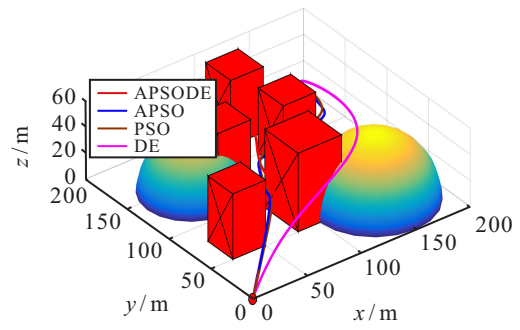
威胁类型	坐标/m	最大威胁半径/m
楼房建筑	(35, 65)	$L = 30, H = 60$
	(100, 60)	$L = 40, H = 80$
	(75, 135)	$L = 30, H = 80$
	(135, 135)	$L = 30, H = 80$
	(255, 165)	$L = 30, H = 60$
	(185, 255)	$L = 30, H = 60$
	(185, 185)	$L = 30, H = 100$
禁飞区域	(265, 225)	$L = 30, H = 100$
	(150, 50, 0)	$R = 50$
	(50, 140, 0)	$R = 50$
天气威胁	(210, 210, 0)	$R = 40$
	(110, 200)	$R = 30, H = 100$
	(200, 100)	$R = 30, H = 100$

### 3.2 航迹规划算法的验证

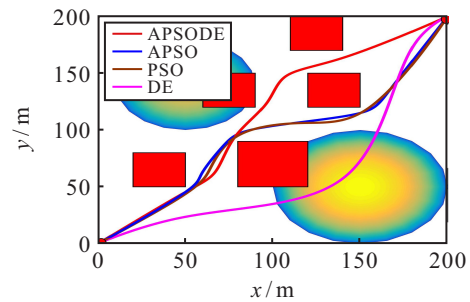
为验证改进算法的有效性和鲁棒性, 本文在不同复杂程度仿真环境下, 将 APSODE 算法与 APSSO、PSO、DE 经 MS 算法拟合前后相比较, 由仿真结果对算法进行评价.

#### 3.2.1 验证算法的可行性

为验证 APSODE 算法的可行性和高效性, 将 APSODE 算法与 APSSO、PSO、DE 算法进行比较, 4 种算法设置相同的最大迭代次数  $Gen = 200$ , 种群规模  $NP = 50$ , 粒子维度  $D = 15$ . 以起始点与目标点直线距离的 1.3 倍对应的适应度值作为算法预期精度, 仿真结果如图6所示.



(a) 三维仿真环境

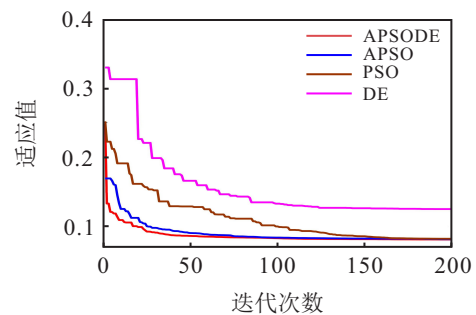


(b) 二维仿真图形

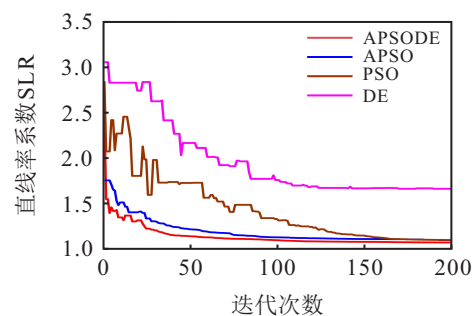
图6 实验1算法规划航迹

由图6可以看出, 4种算法均能有效规避楼房建筑与禁飞区域, 取得可行的航迹. 由图6(b)可以看出, 所设计的 APSODE 算法规划的航程最短且更平滑, APSSO 算法次之, DE 算法航程最差, 选择绕远的方式来规避威胁区域.

实验1 适应度与 SLR 变化如图7所示, 实验1 统计结果如表3所示.



(a) 适应度值变化



(b) SLR 变化

图7 实验1 适应度与 SLR 变化

由图7 适应度值与 SLR 变化曲线和表3 统计

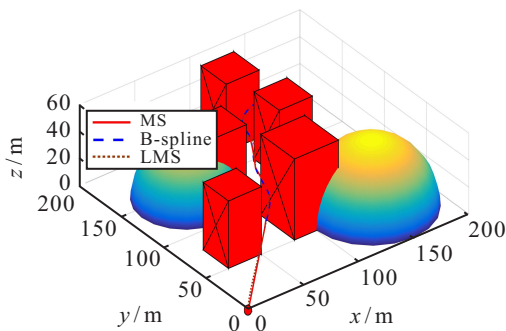
表3 实验1统计结果

算法	最优适应度值	平均适应度值	标准差	最优航程	平均航程	平均时间
APCODE	0.081	0.0837	0.0014	306.134	321.01	9.17
APSO	0.0817	0.0846	0.0027	313.19	323.59	13.5
PSO	0.0821	0.0866	0.0035	313.69	331.02	18.5
DE	0.1251	0.1416	0.0232	472.43	526.11	40
APCODE-MS	0.0805	0.0829	0.0017	305.50	319.27	
APSO-MS	0.0815	0.0839	0.0022	311.87	322.71	
PSO-MS	0.082	0.0862	0.0029	312.83	329.86	
DE-MS	0.086	0.099	0.007	330.71	382.0	

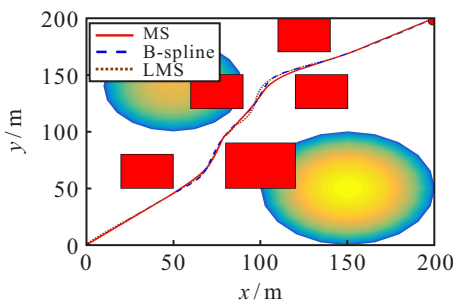
数据可知,APCODE算法与APSO、PSO、DE算法相比,收敛速度最快、精度更高,算法到达预定精度的平均运行时间最短.与APSO算法相比,加入DE算法粒子间的变异操作增大了最优解区域的搜索能力,加快了算法的收敛速度,相比于DE算法,加入正态扰动能够更快跳出搜索停滞和早熟现象;从APCODE算法与传统的PSO算法比较结果可知,改进算法通过将粒子划分不同的搜索模式,采用非线性惯性权重的方法,能够快速搜索到最优解区域,加快算法的收敛速度,且算法有更好的鲁棒性,到达预定精度的平均时间更短.从MS算法拟合前后数据可知,关键航迹点的选取能够有效改善算法的航程距离和精度,对于异常点较多的航迹,如DE算法,效果最为明显.

3.2.2 验证MS拟合算法的有效性

为验证MS拟合算法的有效性,对实验1的数据采用MS算法、三次B样条(B-spline)和最小二乘法(least square method, LSM)分别进行光滑处理,仿真结果如图8所示,拟合数据如表4所示.



(a) 三维仿真环境



(b) 二维仿真环境

图8 不同拟合算法规划航迹

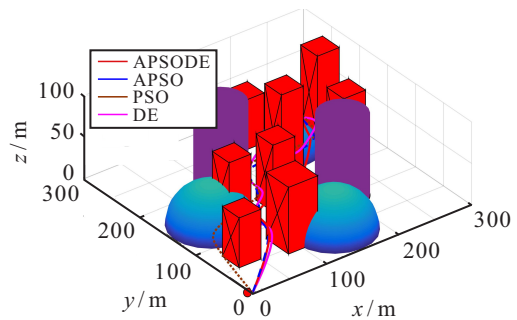
表4 拟合算法比较结果

	最优适应度值	平均适应度值	最优航程	平均航程
原始数据	0.0810	0.0837	306.13	321.0
MS	0.0805	0.0829	305.503	319.269
B-spline	0.0810	0.0841	307.9165	321.76
LMS	0.0824	0.0920	311.87	363.37

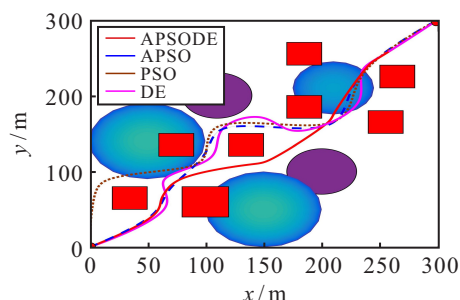
由图8仿真结果和表4拟合算法比较结果可知,3种拟合算法均能很好地对规划航迹进行光滑处理,其中MS算法的适应度值和航程最优,拟合曲线更平滑,且优于原始数据.B-spline拟合结果与原始数据相近.LSM在部分拐弯处凹凸性较差,且平均适应度值与航程远大于原始数据.

3.2.3 验证算法在复杂环境下的有效性

为验证APCODE算法在复杂环境下的有效性,本节在实验1的基础上增添了障碍物的数量,增大了仿真环境的范围,且引入了天气威胁对航程规划的影响.4种算法设置相同参数,以起始点与目标点直线距离的1.3倍对应的适应度值作为算法的预期精度,仿真结果如图9所示.



(a) 三维仿真环境



(b) 二维仿真环境

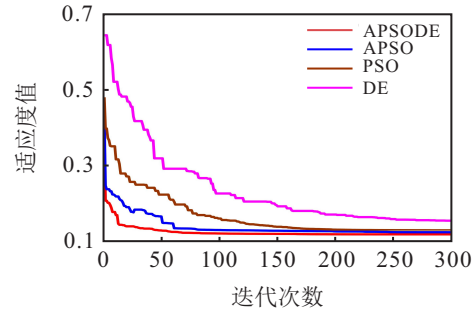
图9 实验2算法规划航迹

由图9可以看出,在复杂环境下4种算法均能有效规避楼房建筑和禁飞区域,以及天气威胁区域,获得可行的航迹.从图9(b)可以看出,APSODE算法航程最短且航迹更平滑,APSO算法次之,PSO算法全局搜索能力较弱,未能收敛到最优解附近,DE算法航程最长,选择绕远的方式规避威胁区域.

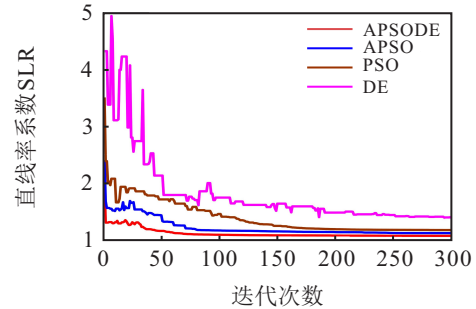
实验2适应度与SLR变化如图10所示,实验2统计结果如表5所示.

由图10适应度值与直线率SLR的变化曲线和表5可知,在复杂环境下,APSODE算法依然能保持优秀的鲁棒性和收敛精度,且达到预定精度的平均运行时间相比于其他3种算法分别降低了29.7%、46.6%、70.6%.与APSO算法相比,加入DE算法的变异操作加快了算法的收敛速度.由图10(b)可以看出,与传统PSO相比,采用非线性惯性权重的方法能够快速搜索到最优解附近,加快了算法的收敛速度,且有着优秀的收敛精度和鲁棒性;在复杂环境下,DE算法极易陷入早熟现象,SLR曲线波动更为明显.从MS算法拟合前后数据可知,算法经MS拟合后的适应度值与航

程均有较为明显的改善,其中DE算法尤为明显.



(a) 适应度值变化



(b) SLR变化

图10 实验2适应度与SLR变化

表5 实验2统计结果

算法	最优适应度值	平均适应度值	标准差	最优航程	平均航程	平均时间
APSODE	0.1185	0.1229	0.0034	451.66	462.20	20.7
APSO	0.1253	0.1329	0.0042	474.42	509.89	30.36
PSO	0.1290	0.1368	0.0074	495.46	529.77	38.77
DE	0.1554	0.2113	0.0406	589.61	745.67	74.5
APSODE-MS	0.1156	0.1194	0.0025	444.81	462.20	
APSO-MS	0.1235	0.1322	0.0042	475.38	508.24	
PSO-MS	0.1288	0.1365	0.0072	495.76	522.75	
DE-MS	0.1444	0.1629	0.0251	531.82	789.85	

### 4 结论

本文提出了基于 APSODE-MS 的无人机航迹规划方法,用于在复杂城市环境下快速规划一条可行、优秀的航迹,提高无人机的执行效能,得出以下主要结论.

1) 与传统 PSO 和 DE 算法相比,APSODE 算法采用自适应惯性方法,通过将粒子分配不同的搜索任务,使算法在迭代开始阶段能够获得较小的适应度值和航程,通过引入 DE 算法和正态扰动,增大了种群的多样性,提高了算法的收敛速度.

2) 在不同的仿真环境下,APSODE 算法依然能够规划出一条优秀的航迹,拥有良好的适应性,且算法适应度值与航程的标准差远低于 APSO、PSO、DE 算法,鲁棒性更强.

3) MS 算法在不同复杂程度的环境下均表现出较好的拟合效果,关键航迹点的选取有效降低了航迹

的航程距离和复杂度.

未来城市环境中存在各种不确定因素,且更加注重多无人机协同执行规划任务,如何在不确定环境下实现多无人机协同航迹规划,是今后的研究方向.

### 参考文献(References)

[1] Shen C, Shi Y, Buckham B. Integrated path planning and tracking control of an AUV: A unified receding horizon optimization approach[J]. IEEE/ASME Transactions on Mechatronics, 2017, 22(3): 1163-1173.

[2] Park J, Karumanchi S, Iagnemma K. Homotopy-based divide-and-conquer strategy for optimal trajectory planning via mixed-integer programming[J]. IEEE Transactions on Robotics, 2015, 31(5): 1101-1115.

[3] Lyu H G, Yin Y. Fast path planning for autonomous ships in restricted waters[J]. Applied Sciences, 2018, 8(12): 2592.

[4] Pehlivanoglu Y V. A new vibrational genetic algorithm

- enhanced with a Voronoi diagram for path planning of autonomous UAV[J]. *Aerospace Science and Technology*, 2012, 16(1): 47-55.
- [5] 张哲, 吴剑, 代冀阳, 等. 基于改进A-Star算法的隐身无人机快速突防航路规划[J]. *航空学报*, 2020, 41(7): 254-264.  
(Zhang Z, Wu J, Dai J Y, et al. Fast penetration path planning for stealth UAV based on improved A-Star algorithm[J]. *Acta Aeronautica et Astronautica Sinica*, 2020, 41(7): 254-264.)
- [6] Chen Y B, Luo G C, Mei Y S, et al. UAV path planning using artificial potential field method updated by optimal control theory[J]. *International Journal of Systems Science*, 2016, 47(6): 1407-1420.
- [7] Tharwat A, Elhoseny M, Hassanien A E, et al. Intelligent Bézier curve-based path planning model using chaotic particle swarm optimization algorithm[J]. *Cluster Computing*, 2019, 22(2): 4745-4766.
- [8] Yang Q, Yoo S J. Optimal UAV path planning: Sensing data acquisition over IoT sensor networks using multi-objective bio-inspired algorithms[J]. *IEEE Access*, 2018, 6: 13671-13684.
- [9] Yu X B, Li C L, Zhou J F. A constrained differential evolution algorithm to solve UAV path planning in disaster scenarios[J]. *Knowledge-Based Systems*, 2020, 204: 106209.
- [10] 方群, 徐青. 基于改进粒子群算法的无人机三维航迹规划[J]. *西北工业大学学报*, 2017, 35(1): 66-73.  
(Fang Q, Xu Q. 3D route planning for UAV based on improved PSO algorithm[J]. *Journal of Northwestern Polytechnical University*, 2017, 35(1): 66-73.)
- [11] Shao S K, Peng Y, He C L, et al. Efficient path planning for UAV formation via comprehensively improved particle swarm optimization[J]. *ISA Transactions*, 2020, 97: 415-430.
- [12] Song B Y, Wang Z D, Zou L. On global smooth path planning for mobile robots using a novel multimodal delayed PSO algorithm[J]. *Cognitive Computation*, 2017, 9(1): 5-17.
- [13] Abhishek B, Ranjit S, Shankar T, et al. Hybrid PSO-HSA and PSO-GA algorithm for 3D path planning in autonomous UAVs[J]. *SN Applied Sciences*, 2020, 2(11): 1-16.
- [14] 单文昭, 崔乃刚, 黄蓓, 等. 基于PSO-HJ算法的多无人机协同航迹规划方法[J]. *中国惯性技术学报*, 2020, 28(1): 122-128.  
(Shan W Z, Cui N G, Huang B, et al. Multiple UAV cooperative path planning based on PSO-HJ method[J]. *Journal of Chinese Inertial Technology*, 2020, 28(1): 122-128.)
- [15] Wang Y B, Bai P, Liang X L, et al. Reconnaissance mission conducted by UAV swarms based on distributed PSO path planning algorithms[J]. *IEEE Access*, 2019, 7: 105086-105099.
- [16] 黄刚, 李军华. 基于AC-DSDE进化算法多UAVs协同目标分配[J]. *自动化学报*, 2021, 47(1): 173-184.  
(Huang G, Li J H. Multi-UAV cooperative target allocation based on AC-DSDE evolutionary algorithm[J]. *Acta Automatica Sinica*, 2021, 47(1): 173-184.)
- [17] Zhou X B, Miao F, Ma H J, et al. A trajectory regression clustering technique combining a novel fuzzy C-means clustering algorithm with the least squares method[J]. *ISPRS International Journal of Geo-Information*, 2018, 7(5): 164.
- [18] Shi Y, Eberhart R. A modified particle swarm optimizer[C]. *IEEE International Conference of Evolutionary Computation*. Anchorage, 1998: 69-73.
- [19] Das S, Mandal A, Mukherjee R. An adaptive differential evolution algorithm for global optimization in dynamic environments[J]. *IEEE Transactions on Cybernetics*, 2014, 44(6): 966-978.
- [20] Iskander A, Elkassed O, El-Badawy A. Minimum snap trajectory tracking for a quadrotor UAV using nonlinear model predictive control[C]. *The 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES)*. Giza, 2020: 344-349.
- [21] Alzaydi A, Sahaluddin M, Islam M, et al. Robotic manipulator task sequencing and minimum snap trajectory generation[J]. *Arabian Journal for Science and Engineering*, 2020, 45(8): 6865-6886.
- [22] Almeida M M D, Moghe R, Akella M. Real-time minimum snap trajectory generation for quadcopters: Algorithm speed-up through machine learning[C]. *2019 International Conference on Robotics and Automation (ICRA)*. Montreal, 2019: 683-689.

### 作者简介

鲁亮亮(1996—), 男, 硕士生, 从事无人机的航迹规划与跟踪控制的研究, E-mail: 411609750@qq.com;

代冀阳(1966—), 男, 教授, 博士生导师, 从事鲁棒控制理论及应用、智能控制技术及其应用等研究, E-mail: djiyang@163.com;

应进(1982—), 男, 讲师, 硕士, 从事智能控制技术及其应用的研究, E-mail: azx519@foxmail.com;

赵玉坤(1990—), 男, 硕士, 从事智能控制的研究, E-mail: 1948443817@qq.com.

(责任编辑: 闫妍)