

改进 PSO 算法及在无人机路径规划中的应用^①



张 姝¹, 汤 淼²

¹(中通服网盈科技有限公司, 南京 210019)

²(南京航空航天大学 民航学院, 南京 211106)

通信作者: 张 姝, E-mail: zhangshu.jsy@chinaccs.cn

摘 要: 在无人机路径规划问题中, 传统算法存在计算复杂与收敛慢等缺点, 粒子群优化算法 (PSO) 得益于其算法原理简单、通用性强、搜索全面等特性, 现多用于无人机航路规划. 然而, 常规 PSO 算法容易陷入局部最优, 本文在优化调整自适应参数的基础上综合引入全局极值变异与加速度项, 以平衡全局和局部搜索效率, 避免种群陷入“早熟”. 对基准测试函数进行测试的结果表明, 本文所提改进 PSO 算法收敛速度更快, 精度更高. 在实例验证部分, 首先提取飞行场景特征, 结合无人机性能约束, 进行环境建模; 然后将多项运行约束和期望的最小化飞行时间均转化为罚函数, 以最小化罚函数作为目标, 构建无人机飞行任务场景下的航路规划模型, 并利用本文所提改进粒子群算法进行求解, 最后通过对比仿真验证了改进粒子群算法的高效性和实用性.

关键词: 粒子群算法; 无人机; 路径规划

引用格式: 张姝, 汤淼. 改进 PSO 算法及在无人机路径规划中的应用. 计算机系统应用, 2023, 32(3): 330-337. <http://www.c-s-a.org.cn/1003-3254/9025.html>

Improved PSO Algorithm and Its Application in Route Planning of UAV

ZHANG Shu¹, TANG Miao²

¹(China Comservice Wangying Technology Co. Ltd., Nanjing 210019, China)

²(College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China)

Abstract: In the path planning of unmanned aerial vehicles (UAVs), the traditional algorithm has the disadvantages of complex computation and slow convergence, while particle swarm optimization (PSO) features simple principle, strong universality, and comprehensive search, which is mainly used in UAV route planning. As the conventional PSO algorithm is easy to fall into the local optimum, this study integrates the global extreme variation and acceleration terms based on the adaptive parameter optimization to balance the global and local search efficiency and avoid the population falling into “premature”. Through the test of a variety of benchmark functions, the results show that the improved PSO algorithm proposed in this study has faster convergence speed and higher convergence accuracy. In the example verification part, the flight scene features are first extracted, and the environment modeling is carried out based on the UAV performance constraints. Then multiple constraints and the expected minimum flight time are converted into penalty functions. With the minimization of penalty functions as the objective, the route planning model is constructed, and the improved PSO algorithm is adopted to solve the problem. Finally, the effectiveness and practicability of the improved PSO algorithm are verified by comparative simulation.

Key words: particle swarm optimization (PSO); unmanned aerial vehicle (UAV); route planning

① 收稿时间: 2022-07-29; 修改时间: 2022-08-26, 2022-09-15; 采用时间: 2022-10-19; csa 在线出版时间: 2022-12-09
CNKI 网络首发时间: 2022-12-13

在地形复杂、环境恶劣等条件下,人们往往无法顺利到达某些区域完成特定任务.无人机技术作为新兴手段在应急救援、物流运输等方面的作用日益凸显,而无人机航路规划作为执行任务的基础更加关键^[1].无人机自主飞行航路规划,是在任务所处地形环境下,寻找一条从起飞点出发到目标区域,高效实现无人机任务,如拍摄影像图、检测生命信号、运送物资等的最优路径的航路设计过程.

无人机航路规划算法主要分为传统经典算法和现代智能算法两大类^[2].前者主要包含有 Dijkstra 算法^[3]、模拟退火算法^[4]以及人工势场法^[5].传统算法在求解多维问题时往往计算量大,过程复杂等问题,而智能优化算法通过规则可在一定时间内得到较好的结果^[6].其中遗传算法稳定性强,但收敛速度不快,容易早熟.稀疏 A* 算法能够减少搜索空间,但求解时间随着规划空间的增大越来越长^[7].而粒子群算法作为一种最典型的人工智能算法,由于其基于种群特性、易于实现和快速收敛等优势,近几年也被广泛用于求解路径规划问题^[8-12].本文提出一种改进 PSO 算法,在优化调整自适应参数的基础上综合引入全局极值变异与加速度项,以平衡全局和局部搜索效率,避免种群陷入“早熟”.通过对多个基准测试函数对改进算子的有效性进行测试验证,结果表明,本文所提改进 PSO 算法收敛速度更快,精度更高.

基于以上研究,以满足约束下的无人机最短飞行时间为目标,构建无人机航路规划模型,采用所提改进 PSO 算法进行求解,经实例验证所提算法在航路规划中具有高效及实用性.

1 改进粒子群优化算法

1.1 粒子群算法概述

PSO 算法初始化为一群随机粒子 X_i , 然后通过迭代找到最优解.在每一次的迭代中,粒子通过跟踪两个“极值” ($p_i^{\text{best}}, g^{\text{best}}$) 来更新自己.在找到这两个最优值后,粒子通过式 (1) 来更新速度和位置:

$$\begin{cases} V_{i+1} = \omega V_i + c_1 r_1 (p_i^{\text{best}} - X_i) + c_2 r_2 (g^{\text{best}} - X_i) \\ X_{i+1} = X_i + V_{i+1} \end{cases} \quad (1)$$

其中, $i = 1, 2, \dots, N$ 表示种群中第 i 个粒子, N 为种群中粒子总数; V_i 是粒子 i 的速度; X_i 是粒子 i 当前位置; c_1 、 c_2 为学习因子; r_1 、 r_2 是在 $[0, 1]$ 区间的随机数; V_i 的

最大值为 $V_{\text{MAX}} > 0$, 如出现 V_i 大于 V_{MAX} , 则令 $V_i = V_{\text{MAX}}$; ω 称为惯性因子, $\omega \geq 0$, 用来平衡群体全局搜索与局部探测能力.粒子当前历史最优位置为 p^{best} , 所有粒子发现的全局历史最优位置为 g^{best} .

粒子群算法的优点是具有较强的鲁棒性,对种群大小敏感性不高,参数少,前期收敛速度快,然而,常规 PSO 算法具有如下两个缺陷:①不能很好地平衡粒子的全局和局部搜索能力;②极易过早陷入迭代停滞.因此,在使用 PSO 算法求解航迹规划问题时,为了提高算法搜索航迹的优越性,克服 PSO 算法的这两个缺陷显得尤为重要.

1.2 自适应参数优化

在进化初期,通常要求粒子的自我学习能力较强而社会学习能力较弱;后期,则要求粒子的自我学习能力较弱而社会学习能力较强.因而对于学习因子进行线性变化,如式 (2) 所示. k 为当前迭代次数; k_{MAX} 为最大迭代次数.通常取 $c_{1\text{MIN}} = c_{2\text{MIN}} = 0$, $c_{1\text{MAX}} = c_{2\text{MAX}} = 2$, 视具体问题定.

$$\begin{cases} c_1 = (c_{1\text{MIN}} - c_{1\text{MAX}}) \frac{k}{k_{\text{MAX}}} + c_{1\text{MAX}} \\ c_2 = (c_{2\text{MAX}} - c_{2\text{MIN}}) \frac{k}{k_{\text{MAX}}} + c_{2\text{MIN}} \end{cases} \quad (2)$$

对粒子群算法的优化研究中,对于惯性权重因子 ω 通常也采用线性减小的计算方式.线性变化的方式在一定程度上能提高搜索的效率及成功率,但是并没有考虑到个体以及种群的实际情况.在前期的搜索过程中可能成效并不明显,且随着迭代次数上升,贸然地减小惯性因子会使得搜索效率降低甚至陷入局部最优无法跳出.因此,本文提出改进的 ω 自适应变化方式,如式 (3) 所示.其中, ω_{sta} 为初始惯性权重,也为最大惯性权重; ω_{end} 为迭代至最佳位置时的惯性权重,也为最小惯性权重;通常取 $\omega_{\text{sta}} = 0.9$, $\omega_{\text{end}} = 0.4$. λ 为控制变化平滑度的因子,根据多次试验 λ 和 γ 均设为 3 较为合适.

$$\omega = \omega_{\text{end}} + (\omega_{\text{sta}} - \omega_{\text{end}}) \exp \left[\left(-\lambda \left(\frac{k}{k_{\text{MAX}}} \right)^\gamma \right) \right] \quad (3)$$

经试验发现,通过对于惯性权重、学习因子的动态调整,PSO 算法的全局搜索能力和局部搜索能力有所改善,但其容易陷入局部极值的问题还是存在.

1.3 针对陷入局部极值的优化

研究粒子陷入局部极值的产生机理,其中文献 [12] 通过严格数学推导,得到粒子收敛情况,如式 (4) 所示:

$$\lim_{t \rightarrow +\infty} x(t) = p^* = \frac{c_1}{c_1 + c_2} \cdot p^{\text{best}} + \frac{c_2}{c_1 + c_2} \cdot g^{\text{best}} \quad (4)$$

由式(4)可知,由于随着迭代次数, c_1 从 $c_{1\text{MAX}}$ 开始逐渐变小, c_2 从 $c_{2\text{MIN}}$ 逐渐变大,则有随迭代次数增加, g^{best} 对应参数较大,同时若粒子陷入局部极值后全局极值 g^{best} 也等于局部极值,因此,可以通过条件触发下的 g^{best} 变异来更新全局极值,帮助粒子脱离局部极值的阻滞环境.当检测到 g^{best} 未达到预测的极值却连续出现达到预设次数,则触发 g^{best} 极值变异,变异算子如式(5)所示:

$$g^{\text{best}} = [1 + \alpha \times U(0, 1)]g^{\text{best}} \quad (5)$$

其中, $U(0, 1)$ 表示均匀分布的随机变量, α 为常数,视具体问题而定,在 $[0.5, 1]$ 间取值较为适宜.

极值变异是为了帮助粒子群跳出局部阻滞,而为了避免陷入局部极值,参考文献[12],并进行参数改进,即在速度更新公式中加上加速度项,如式(6):

$$a(t+1) = B \times (f(x_i(t)) - f(g^{\text{best}}(t))) / (f_m(X(t)) - f(g^{\text{best}}(t))) \quad (6)$$

其中, B 为加速度量程系数,根据多次试验取该维度上粒子运动范围大小的 $1/50$,例如当 x_i 属于 $[-500, 500]$ 时, $B=20$. $f(x_i(t))$ 为第 i 粒子在第 j 次迭代周期的适应度值, $f(g^{\text{best}}(t))$ 表示为上一代粒子的最优适应度值, $f_m(X(t))$ 表示粒子群第 t 次迭代的平均适应度值.分析可知,假如粒子的第 t 代适应度值与 $t-1$ 代粒子群的最优适应度值偏差越大,更新后粒子速度增大,搜索区域更大,利于避免局部最优.

1.4 算法流程

M 表示粒子的维数, N 表示粒子的个数.使用粒子群算法求解最优航迹时,每个粒子 X_i 代表一条待选航路, $X_i = (x_{im}, y_{im}, z_{im})$, $i = 1, 2, \dots, N$, $m = 1, 2, \dots, M$,即 M 表示每条待选航路包含的航路点数, N 表示待选航路数.所提改进粒子群算法流程具体如下.

Step 1. 初始化.初始化粒子群,包括种群大小、迭代次数、初代粒子位置和速度以及其他各参数等.

Step 2. 适应度计算.确定目标函数,根据目标函数得出粒子在搜索区域内适应度值,并保存和更新个体最优及群体最优.

Step 3. 自适应参数计算.计算自适应惯性权重、学习因子、加速度等.

Step 4. 极值变异.比较本次迭代适应度值与前

k 次迭代的适应度值,如果连续 $k+1$ 次群体最优适应度相同或相差在设定范围内,则可判断粒子陷入局部最优,对全局最优值进行变异.

Step 5. 种群更新.用更新后的参数和极值继续对群体中粒子的速度和位置进行更新;如果粒子的速度和位置超出最大值和边界值,则按最值计算.

Step 6. 循环条件检查.根据目标函数值或最大迭代次数判断是否符合循环终止要求,符合条件时结束算法循环,否则转到Step 2继续执行.

PSO算法可用伪代码表示如算法1.

算法1. PSO算法

初始化粒子群及参数;

DO

For 各个粒子

计算其适应度值;

按式(2),式(3),式(6)更新自适应惯性权重、学习因子加速度等参数;

If (粒子个体适应度优于个体历史最优值)

用 X_i 更新历史最佳个体 p_i^{best} ;

End

选取当前粒子群中最优的粒子位置作为全局最优;

If (当前最优粒子优于全局历史最优粒子)

用当前全局最佳粒子更新 g^{best} ;

If (若连续 $k+1$ 次群体最优适应度相同或相差在设定范围内)

按式(5)对全局最优值进行变异;

For 各个粒子

按式(1)更新粒子速度各位置;

End

Until 满足终止条件.

1.5 改进算子有效性仿真实验

在内存16 GB, Intel(R) Core(TM) i7-1165G7 @ 2.20 GHz CPU的计算机上进行仿真,选用Win 10, Matlab R2016a编程实现各测试试验.通过设计对比试验,对改进粒子群算法提出的参数改进和阻滞优化进行验证.设置种群大小为100,最大迭代次数为1000.速度限制设为该维度粒子运动范围阈值大小.依次使用常规PSO算法和添加不同优化方式的PSO算法,分别称为PSO1-PSO6,对6个标准测试函数分别进行试验,验证算法在二维和高维、单模态和多局部极值等不同情况下的性能.为了降低数据的随机性,对于算法PSO1-PSO6,对应6种测试函数的组合都分别独立进行20次运算.

对比算法对应的参数选择和改进设计如下.

(1) 算法PSO1: 常规PSO算法, $\omega=0.8$, $c_1=c_2=$

1.49445.

(2) 算法 PSO2: ω 改进自适应, 学习因子同 PSO1, 即式 (7):

$$\begin{cases} \omega = \omega_{\text{end}} + (\omega_{\text{sta}} - \omega_{\text{end}}) \exp \left[\left(-3 \left(\frac{k}{k_{\text{MAX}}} \right)^3 \right) \right] \\ c_1 = c_2 = 1.49445 \end{cases} \quad (7)$$

(3) 算法 PSO3: ω 、 c_1 、 c_2 均改进自适应, 即式 (8).

(4) 算法 PSO4: 基于 PSO3, 增加加速度项.

(5) 算法 PSO5: 基于 PSO3, 增加极值变异算子.

(6) 算法 PSO6: 本文提出的综合改进 PSO 算法.

$$\begin{cases} c_1 = (c_{1\text{MIN}} - c_{1\text{MAX}}) \cdot \frac{k}{k_{\text{MAX}}} + c_{1\text{MAX}}, \\ c_{1\text{MIN}} = 0.5, c_{1\text{MAX}} = 2 \\ c_2 = (c_{2\text{MAX}} - c_{2\text{MIN}}) \cdot \frac{k}{k_{\text{MAX}}} + c_{2\text{MIN}}, \\ c_{2\text{MIN}} = 0.5, c_{2\text{MAX}} = 2 \\ \omega = \omega_{\text{end}} + (\omega_{\text{sta}} - \omega_{\text{end}}) \exp \left[\left(-3 \left(\frac{k}{k_{\text{MAX}}} \right)^3 \right) \right], \\ \omega_{\text{end}} = 0.4, \omega_{\text{sta}} = 0.9 \end{cases} \quad (8)$$

选择的测试函数如表 1 所示.

表 1 测试函数及参数设置

序号	测试函数	函数公式	x_i 范围	全局最优值	模态	维度
1	Sphere	$f_1(x) = \sum_{i=1}^d x_i^2$	[-100, 100]	$\min f=0$	单峰	30
2	Rosenbrock	$f_2(x) = \sum_{i=1}^d \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$	[-5, 10]	$\min f=0$	单峰非凸	30
3	Griewank	$f_6(x) = \frac{1}{4000} \sum_{i=1}^d x_i^2 - \prod_{i=1}^d \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$	[-600, 600]	$\min f=0$	多峰, 多局部极小点	30
4	Schwefel	$f_2(x) = 418.9829d - \sum_{i=1}^d x_i \cdot \sin(\sqrt{ x_i })$	[-500, 500]	$\min f=0$	多峰	30
5	Rastrigin	$f_4(x) = \sum_{i=1}^d [x_i^2 - 10 \cos(2\pi x_i) + 10]$	[-5.12, 5.12]	$\min f=0$	多峰, 多局部极值点	30
6	Schaffer	$f_7(x_1, x_2) = 0.5 + \frac{(\sin \sqrt{x_1^2 + x_2^2})^2 - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}$	[-100, 100]	$\min f=0$	多峰	2

对 20 次计算结果取平均值, 得到对比试验显示的目标函数收敛结果如图 1(a)–图 1(f) 所示. 由图可知, 常规 PSO 算法前期得到的适应度值普遍较大, 而且很快在 20 代左右就陷入到局部极值直到迭代结束. 在常规 PSO 算法中引入自适应惯性权重后在前期寻优中发生明显改善, 延后了陷入局部最优的时间, 寻到的极值也相对较好, 而叠加动态变化的学习因子后, 这种效果得到了加强, 然而算法仍然难以逃脱局部极值. 在此基础上, 加速度和极值变异均能很好地达到摆脱局部最优的效果. 数据显示, 极值变异的效果明显更好. 但单独使用并不能保证达到全局收敛到最优的效果, 如 Sphere、Rastrigin 中单独使用都没有在预设迭代次数中找到全局最优, 而综合使用则在 70 代左右就成功快速找到极值. 在 Schwefel 函数测试中, 虽然各种算法都未能找到最优解, 但综合使用算子较好平衡了全局和局部寻优, 找到的解质量最高. 从 Schatter 测试中可以发现, 改进算法对于高维的效果更好, 而在二维环境下

仍然是综合使用算子效果更好. 因此, 对比试验充分验证了改进算法的高效性与实用性.

2 路径规划问题建模

无人机根据任务要求, 如何规划穿过危险区域, 避免禁飞区, 躲过飞行威胁或障碍物等, 顺利从起始点安全飞行到各目标点完成任务是需要考虑的重中之重. 因此, 需要根据已知静态环境规划满足各项约束且综合评价高的可飞路径. 为了直观表述路径的预计总飞行成本并便于计算, 针对飞行过程中的各类威胁和约束, 具体包括恶劣天气威胁、地形障碍威胁、地形高度限制、飞行距离限制、转弯角约束、俯仰角约束等, 采用罚函数的方式, 将不利因素转化为罚函数.

2.1 场景建模

2.1.1 山体建模

本文主要考虑山区飞行, 无人机不能通过山脉飞行. 本文采用如式 (9) 所示的数学模型模拟山体:

$$z_m = f_m(x, y) = h_i \cdot \exp \left[- \left(\frac{x - x_{oi}}{x_{si}} \right)^2 - \left(\frac{y - y_{oi}}{y_{si}} \right)^2 \right], i \in (0, N] \quad (9)$$

其中, (x, y) 表示地形投影在水平面上的点坐标,

$f_m(x, y)$ 表示点 (x, y) 处的高程 z_m , N 表示山体数量, h_i 代表山高, (x_{oi}, y_{oi}) 表示第 i 座山体的地理中心坐标, 参数 x_{si} 、 y_{si} 为常数, 分别为第 i 座山峰分别沿 x 轴和 y 轴的高度衰减量, 模拟山峰的陡峭程度.

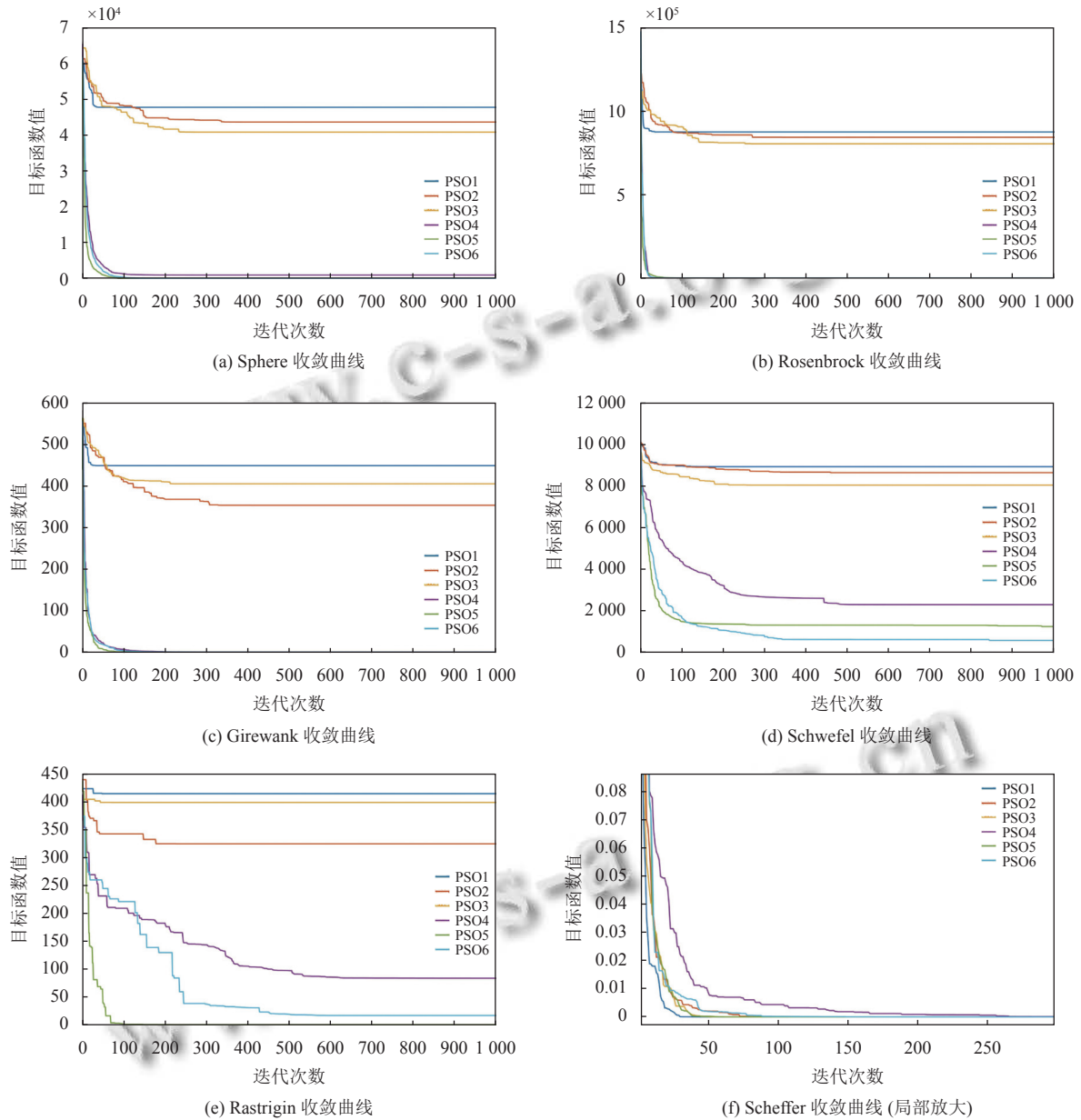


图1 测试函数收敛曲线

2.1.2 地形建模

在无人机飞行场景中, 还要考虑地形的起伏. 同样采用函数模拟的方法进行地形建模. 模拟山区地形起伏的函数表示为式 (10), 其中, $f_i(x, y)$ 表示水平面上点 (x, y) 对应的高程 z_i . 参数 a 、 b 、 c 、 d 、 e 、 f 、 g 、 h 为常数, 通过对这些系数的调整可以模拟不同场景下的地

形. 因此, 总的地形高程可以表征为 $z(x, y)$, 具体如式 (11) 所示:

$$f_i(x, y) = a \cdot \left[\sin(y + b) + c \sin x + \cos \left(d \cdot \sqrt{x^2 + y^2} \right) + e \sin y + f \sin \left(g \cdot \sqrt{x^2 + y^2} \right) + h \cos y \right] \quad (10)$$

$$z(x, y) = z_m + z_t = f_m(x, y) + f_t(x, y) \quad (11)$$

2.1.3 障碍建模

除此以外, 常见的还有恶劣天气及各种因素造成的禁飞区, 无人机飞行过程中需要避开这些危险区域. 根据常见障碍区域廓形, 本文将障碍区域建模为长方体和圆柱体. 因此, 无人机飞行场景建模如图2所示.

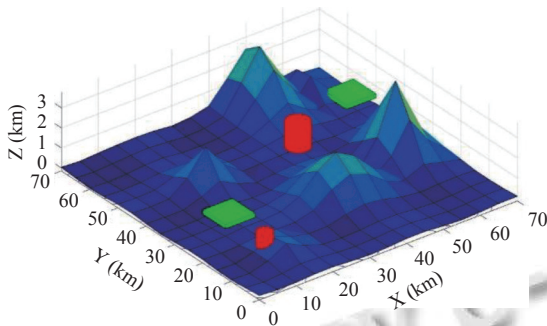


图2 应急无人机飞行场景模型

2.2 约束建模

2.2.1 避障约束

无人机可飞路径是不能从障碍物中穿过的, 所以必须避免与山峰丘陵等地形障碍的碰撞以及飞入威胁天气或者禁飞区范围.

为了保证这种避障行为, 本文采用罚函数, 对存在至少一个航路点落在障碍物内的解进行惩罚. 假设函数 $f(x_i, y_i)$ 返回直角坐标系下无人机在航路点 j 的坐标 (x_i, y_i) 对应地形高度值, γ 用于判断点是否在规则障碍物内, $\gamma < 1$ 表示在障碍物内部, $\gamma \geq 1$ 表示在表面或外部. 根据落入障碍物内的航路点总数, 可以通过式(12)实现惩罚.

$$J_1 = \sum_{i=1}^M \mu_i, \mu_i = \begin{cases} 1, & \text{if } z_i < f(x_i, y_i) \text{ or } \gamma < 1 \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

2.2.2 飞行高度约束

由于无人机的性能限制, 同时一些受地面站控制的无人机需要与地面站联系, 所以一般需要考虑无人机的最大飞行高度限制, 而飞行高度过低又会增大与建筑物、山峰等的碰撞风险, 同时过高或过低的飞行高度也将消耗更多的燃料. 因此在建模中, 相对于地形的最小和最大飞行高度被设置为约束条件. 通过累计计算无人机各航路点的飞行高度(相对于该处地形高度), 再除以规划轨迹中的总航路点数得到平均高度. 使无人机在满足性能要求的情况下倾向于搜索低高度

的航线. 因此对应的罚函数如式(13):

$$J_2 = \begin{cases} \frac{\sum_{i=1}^M (z_i - f(x_i, y_i) - H_{\min})}{M}, & \text{if } z_i > f(x_i, y_i) + H_{\min} \\ \frac{\sum_{i=1}^M (f(x_i, y_i) + H_{\min} - z_i)}{M}, & \text{otherwise} \end{cases} \quad (13)$$

其中, (x_i, y_i, z_i) 是无人机计划路径的第 i 个航路点, $f(x_i, y_i)$ 点 (x_i, y_i) 处的地形高度, H_{\min} 为期望的无人机最小飞行高度, M 是无人机该计划路径的航路点总数.

2.2.3 转角约束

首先, 水平方向上无人机在任意航路点的转弯角度 Ω 不能超过最大转弯角 Ω_{\max} , 如图3所示. 对应的罚函数如式(14)所示:

$$J_3 = \frac{1}{M} \sum_{i=1}^M \delta_i, \delta_i = \begin{cases} 1, & \text{if } \Omega_i > \Omega_{\max} \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

其次, 无人机只能在特定角度范围内进行俯仰动作, 要求无人机在某一航迹点 i 的俯仰角 θ_i 不能大于最大俯仰角 w , 且不能小于最小值 σ , 对应的罚函数如式(15)所示:

$$J_4 = \begin{cases} \frac{\sum_{i=1}^M (\theta_i - w)}{w}, & \text{if } \theta_i > w \\ \frac{\sum_{i=1}^M (\sigma - \theta_i)}{\sigma}, & \text{if } \theta_i < \sigma \\ 0, & \text{otherwise} \end{cases} \quad (15)$$

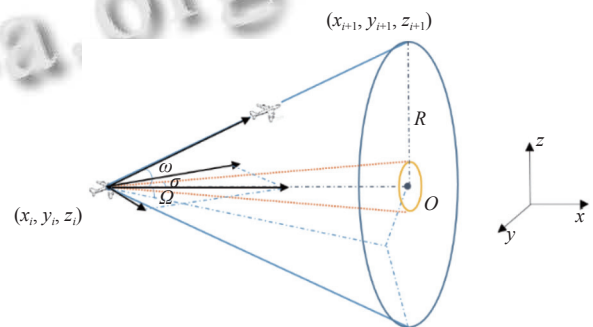


图3 无人机俯仰角和爬升角示意图

2.3 目标函数

无人机应急飞行任务对时效性要求高, 越快到达目标地点越好. 假设无人机在飞行过程中匀速飞行, 因此对于任意一架无人机来说, 在相同的条件下, 路径越短, 飞行时间就越少, 在复杂环境中面临未知风险威胁的概率也会减少. 同时无人机因为自身性能, 飞行受到

最大飞行长度的限制. 因此, 优化目标为满足约束条件下的最小化路径长度, 如式 (16) 所示:

$$\min f = L + \sum_{k=1}^4 J_k \quad (16)$$

然而由于路径长度 L 和罚函数 J 量级相差较大, 故可将路径长度也转化为罚函数, 如式 (17) 所示:

$$J_5 = \begin{cases} L_w - L_{\max} \\ L_{\max} \end{cases} \quad (17)$$

其中, L_{\max} 表示无人机的最大飞行距离, L_w 表示待选飞行路径总长度, 由计算各航路点间距离累加得到. 因此, 可通过求最小化惩罚函数来求解满足约束条件下的最短路径.

因此式 (16) 中的目标函数可转化为式 (18):

$$\min f = \sum_{k=1}^5 J_k \quad (18)$$

3 实例验证

建立仿真环境, 假设无人机的飞行区域在 XOY 平面内为 70 km×70 km 的矩形, 出发点坐标为 (0, 0, 0),

目的地坐标为 (68, 55, 0). 地形模型的参数分别为 $a=0.01, b=c=6, d=1.1, e=1.5, f=2, g=5, h=6$. 山峰模型的参数分别为 $H = [h_1, h_2, \dots, h_6] = [0.8, 2, 3, 1.8, 1.8, 1.5]$; $X_o = [x_{o1}, x_{o2}, \dots, x_{o6}] = [15, 40, 50, 60, 20, 60]$; $Y_o = [y_{o1}, y_{o2}, \dots, y_{o6}] = [12, 25, 65, 30, 45, 62]$; $X_s = [x_{s1}, x_{s2}, \dots, x_{s6}] = [5, 9, 6, 5, 5, 4]$; $Y_s = [y_{s1}, y_{s2}, \dots, y_{s6}] = [5, 8, 10, 7, 5, 6]$. $H_{\min}=100, \Omega_{\max}=74^\circ, w=15^\circ, \sigma=2^\circ$. 分别采用本文所提综合改进 PSO 算法、PSO5 以及文献 [7,12] 中的改进 PSO 算法对飞行路径进行规划. 本节提出的 PSO 算法及 PSO5 算法参数同第 1.5 节, 对比算法相关参数同文献. 设定最大迭代次数为 300, 粒子群数量为 30, 粒子维数为 20. 规划结果如图 4 所示, 本文算法对应黑色航迹, PSO5 对应, 文献 [7] 算法对应绿色航迹, 文献 [12] 算法对应紫色航迹.

分析试验数据, 如表 2 所示, 可以看到, 本文所提算法找到的航路适应值最小, 质量最高, 算法运算时间等性能相对较高. 因此, 基于本文所提改进粒子群算法的无人机航路规划方法具有可行性与有效性.

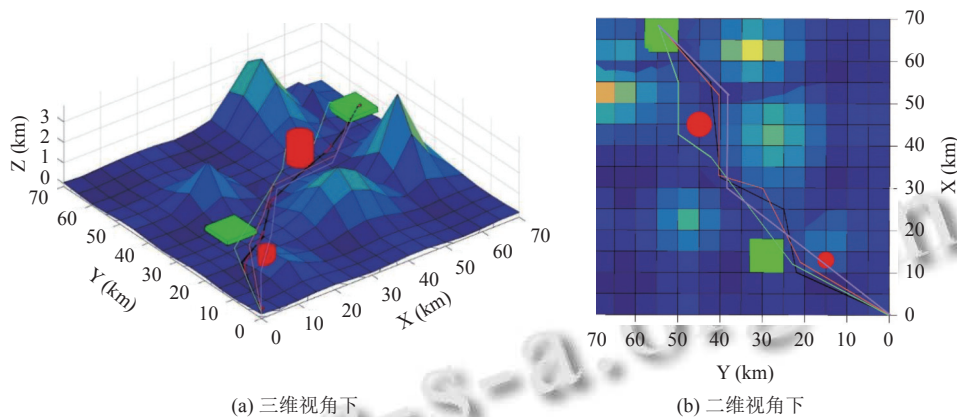


图 4 对比算法飞行路径规划结果效果图

表 2 实例验证结果数据

序号	适应度值	总运行时间 (s)	找到最优解的迭代次数	找到最优解的执行时间 (s)
1	21.75	30.978	21	9.452
2	31.41	25.476	32	11.782
3	98.53	73.986	68	34.352
4	45.89	47.347	49	23.691

4 结束语

针对无人机在执行任务时路径规划的特殊性, 分析飞行环境特征并结合无人机性能限制进行综合建模. 结合粒子群算法参数少、搜索快但极易陷入局部最优

的特性, 提出了一种综合参数自适应、加速度和全局极值变异的无人机路径规划方法. 仿真实验结果表明, 通过引入非线性自适应惯性权重和线性变化学习因子, 动态地调节了算法在全局搜索能力与局部搜索能力之间的平衡. 增加条件触发下的全局极值变异算子, 帮助粒子群尽量避免陷入局部最优, 并提高了粒子群摆脱局部极值的能力. 改进的算法具有较好的搜索能力, 也能保持较好的求解稳定性.

由于在执行特殊任务时, 路径规划时间对于执行任务尤为重要, 即算法的计算时间对寻优的质量具有较大影响, 因此, 后续将继续研究算法效能提升方法.

此外,近几年极端天气情况出现频繁,后期需要进一步研究天气对于航路规划的影响。

参考文献

- 1 邹蒲, 邓吉秋, 刘立, 等. 地质灾害应急中无人机运输与飞行路径规划. 测绘与空间地理信息, 2017, 40(10): 15–17, 20. [doi: 10.3969/j.issn.1672-5867.2017.10.005]
- 2 王琼, 刘美万, 任伟建, 等. 无人机航迹规划常用算法综述. 吉林大学学报(信息科学版), 2019, 37(1): 58–67. [doi: 10.19292/j.cnki.jdxxp.2019.01.008]
- 3 Li M, Zhang F, Fang JY. Optimal path solution based on Dijkstra algorithm. *Frontiers in Economics and Management*, 2021, 2(6): 170–176.
- 4 Xiao SC, Tan XJ, Wang JP. A simulated annealing algorithm and grid map-based UAV coverage path planning method for 3D reconstruction. *Electronics*, 2021, 10(7): 853. [doi: 10.3390/electronics10070853]
- 5 Wang X, Yong EM, He KF, *et al.* Research on artificial potential field route planning method based on threat level assessment. *Proceedings of the 33rd Chinese Control and Decision Conference*. Kunming: IEEE, 2021. 510–516.
- 6 康玉祥, 姜春英, 秦运海, 等. 基于改进 PSO 算法的机器人路径规划及实验. *机器人*, 2020, 42(1): 71–78. [doi: 10.13973/j.cnki.robot.190035]
- 7 李鹏, 李兵舰, 元亮, 等. 一种改进的粒子群优化算法及其在无人机航路规划中的应用. *舰船电子对抗*, 2019, 42(5): 59–64. [doi: 10.16426/j.cnki.jcdzdk.2019.05.015]
- 8 Liu Y, Zhang XJ, Zhang Y, *et al.* Collision free 4D path planning for multiple UAVs based on spatial refined voting mechanism and PSO approach. *Chinese Journal of Aeronautics*, 2019, 32(6): 1504–1519. [doi: 10.1016/j.cja.2019.03.026]
- 9 付兴武, 胡洋. 基于改进粒子群算法的三维路径规划. *电光与控制*, 2021, 28(3): 86–89. [doi: 10.3969/j.issn.1671-637X.2021.03.017]
- 10 杨超杰, 裴以建, 刘朋. 改进粒子群算法的三维空间路径规划研究. *计算机工程与应用*, 2019, 55(11): 117–122. [doi: 10.3778/j.issn.1002-8331.1808-0245]
- 11 Clerc M, Kennedy J. The particle swarm—Explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 2002, 6(1): 58–73. [doi: 10.1109/4235.985692]
- 12 谢勇宏, 孔月萍. 基于改进粒子群算法的三维路径规划. *计算机测量与控制*, 2022, 30(3): 179–182, 191.

(校对责编: 牛欣悦)