

# 基于非线性跨代差分进化的花授粉优化算法及其应用研究

梁 靛, 魏亚星, 李义鑫, 贾云健  
(重庆大学微电子与通信工程学院, 重庆 400000)

**摘要:** 针对高维度变量的优化问题, 本文设计了一种基于非线性跨代差分进化的花授粉优化算法. 该算法利用跨代差分进化引导个体逼近最优解, 使算法的局部搜索过程具备导向性, 并设置非线性惯性权重提升算法的搜索收敛速度. 同时, 通过参数自适应调整实现缩放因子和交叉概率的动态更新, 从而提高种群丰富度、减少局部解的数量, 再结合跨代赌轮盘方式以降低陷入局部最优解的概率. 仿真验证表明, 该算法能够在不同维度测试函数下保持较好的寻优特性和稳定性, 尤其在高维度测试函数下的寻优性能更好. 同时, 本文以工业互联网中的无人机智能巡检的路径规划为例, 评估了算法在实际应用中的性能. 实验结果表明该算法可以满足巡检路径规划的低成本、高效率 and 规避外部攻击的需求.

**关键词:** 花授粉算法; 跨代差分进化; 赌轮盘; 智能巡检; 无人机; 路径规划

**基金项目:** 国家自然科学基金面上项目 (No.62071075, No.61971077); 重庆市自然科学基金面上项目 (No.cstc2020jcyj-msxmX0704); 中央高校基本科研业务费项目 (No.2020CDJ-LHZZ-022)

**中图分类号:** TP391

**文献标识码:** A

**文章编号:** 0372-2112(2023)09-2445-12

**电子学报 URL:** <http://www.ejournal.org.cn>

**DOI:** 10.12263/DZXB.20211674

## A Flower Pollination Algorithm Based on Nonlinear Cross-Generation Differential Evolution and Its Application Study

LIANG Liang, WEI Ya-xing, LI Yi-xin, JIA Yun-jian

(School of Microelectronics and Communication Engineering, Chongqing University, Chongqing 400000, China)

**Abstract:** For the optimization problem of high-dimensional variables, we design a flower pollination algorithm based on nonlinear cross-generation differential evolution (FPA-NCDE). The algorithm guides individuals to approximate the optimal solution with cross-generation differential evolution to make local search process oriented. Meanwhile, the nonlinear inertia weight is set to improve the search convergence speed. The scaling factor and crossover probability are dynamically updated by parameter adaptive adjustment to enhance the population richness and reduce the number of local solutions. Combined with the cross-generation roulette wheel, the probability of trapping into local optimal solution is decreased. The performance evaluation verifies that the proposed FPA-NCDE can maintain good optimization characteristics and stability under different dimensional benchmark functions, especially under high dimensional test functions. In addition, FPA-NCDE is applied to unmanned aerial vehicle intelligent inspection of industrial internet to evaluate the performance of the algorithm in practical applications. The experiments results show that FPA-NCDE can satisfy the needs of low cost, high efficiency and avoidance of external attacks in inspection path planning.

**Key words:** flower pollination algorithm; cross-generation differential evolution; roulette wheel; intelligent inspection; unmanned aerial vehicle; path planning

**Foundation Item(s):** National Natural Science Foundation of China (No.62071075, No.61971077); Natural Science Foundation of Chongqing (No.cstc2020jcyj-msxmX0704); Fundamental Research Funds for the Central Universities (No.2020CDJ-LHZZ-022)

## 1 引言

随着物联网和人工智能等科学技术的创新和发展,在工程和工业应用中存在着大量复杂的高维度变量优化问题亟待解决,如城市公共设施规划建设、电子网络与通讯传播、无人汽车智能路径设置、无人机智能巡检等<sup>[1-3]</sup>. 目前针对这些问题的众多研究采用了启发式优化算法,这类算法是基于生物系统特性和物理理论、达尔文进化论、昆虫或动物的群体行为等发展起来的,具有搜索精度高、效率高、鲁棒性强的特点<sup>[4-9]</sup>.

启发式优化算法大致可以分为三类:(1)能够平衡局部搜索与全局搜索的算法,如人工蜂群(Artificial Bee Colony, ABC)算法<sup>[10]</sup>、蜘蛛群集优化(Social Spider Optimization, SSO)算法<sup>[11]</sup>. 这些算法可以很好地将快速性和全局最优特征结合起来,但在搜索过程的后期,会出现算法性能变差的情况.(2)兼具全局收敛性和计算鲁棒性的算法,如狼群算法(Wolf Pack Algorithm, WPA)<sup>[12]</sup>、鲸鱼算法(Whale Algorithm, WA)<sup>[13]</sup>. 这些算法能够同时从多个点展开搜索,具备良好的收敛性能,但会出现搜索效率偏低和种群多样性不足的情况.(3)具有较好种群丰富度的算法,如粒子群优化(Particle Swarm Optimization, PSO)<sup>[14]</sup>、遗传算法(Genetic Algorithm, GA)<sup>[15]</sup>. 然而,这些算法可能会出现数值扰动和陷入局部最优的情况. 综上所述,传统启发式优化算法存在提前收敛和以及数值波动等缺点,在探索最优解的能力上还有待提高.

针对传统启发式优化算法的上述不足, Yang 提出了基于开花植物的花授粉算法(Flower Pollination Algorithm, FPA),并在多目标优化中得到了广泛应用<sup>[16]</sup>. FPA 能够以较高的精度和较好的收敛性能连续完成对多个对象的搜索任务,并且在局部和全局搜索过程中保持较好的平衡<sup>[17-19]</sup>. 这一算法被广泛应用于各类优化问题,如 Nguyen 利用 FPA 来解决优化过程中的数据振荡的问题<sup>[20]</sup>, PANDA 使用 FPA 消除对称和不对称多电平逆变器中的目标谐波<sup>[21]</sup>. 通过分析相关研究<sup>[22]</sup>, FPA 存在三个主要问题:(1)局部搜索过程缺乏搜索方向,可能陷入局部最优,导致搜索效率和收敛性降低.(2)对参数敏感,惯性权重等参数的设置会影响全局搜索最优解的精度和迭代次数.(3)优化问题复杂度的提升增加了局部最优解数量,阻碍了 FPA 获得最优解. 针对 FPA 的不足之处,本文提出了一种基于非线性跨代差分进化的花授粉算法(Flower Pollination Algorithm based on Nonlinear Cross-generation Differential Evolution, FPA-NCDE). 该算法利用跨代差分进化引导个体逼近最优解,使算法的局部搜索过程具备导向性. 同时设置非线性惯性权重使全局搜索初期以较快速度加强搜索能力,提升算法的收敛速度. 此外,通过进行参数

自适应调整实现动态更新缩放因子和交叉概率,提高种群丰富度、减少局部解的数量,再结合跨代赌轮盘方式以降低陷入局部最优解的概率. 为了评估算法在实际应用中的性能,本文以无人机智能巡检的路径规划问题为例,根据优化变量设计出相应的飞行航迹,进行算法的有效性验证.

## 2 基于非线性差分进化的花授粉算法

本文提出的 FPA-NCDE,在全局搜索过程引入了非线性惯性权重,在局部搜索过程引入了跨代差分进化,并针对跨代差分进化进行了参数自适应调整,最后利用跨代赌轮盘选择机制来提升算法寻优精确度,避免陷入局部最优解.

### 2.1 非线性惯性权重设置

传统 FPA 惯性权重是一个随机值,无法指导全局搜索方向,会影响搜索效率. 初始阶段较大的惯性权重有助于提高 FPA 全局搜索能力. 随着迭代次数的增加,惯性权重的减小可以加速局部搜索过程.

线性惯性权重会降低搜索效率并陷入局部最优<sup>[23-25]</sup>,为了平衡局部-全局搜索过程,提高全局搜索精度,基于具有全局探索和局部开发优势的模拟退火算法<sup>[26]</sup>,设计非线性惯性权重如下:

$$k = 0.2 \exp\left(-25 \frac{t}{T_{\max}}\right)^3 + 0.8k_1 \quad (1)$$

$$k_1 = 1 - 0.5h + 0.2s \quad (2)$$

$$h = \left| \frac{\min(\min f(x_i^{t-1}), \min f(x_i^t))}{\max(\min f(x_i^{t-1}), \min f(x_i^t))} \right| \quad (3)$$

$$s = \left| \frac{\min(\text{mean} f(x_i^{t-1}), \min f(x_i^t))}{\max(\text{mean} f(x_i^{t-1}), \min f(x_i^t))} \right| \quad (4)$$

其中,  $i$  为第  $i$  个花粉个体的索引,  $T_{\max}$  为最大迭代次数,  $t$  为当前迭代次数,  $\min(f(x_i^t))$  为当前种群的最小适应度值,  $\min(f(x_i^{t-1}))$  为上一代种群的最小适应度值. 权重  $k$  的变化能够指导全局搜索方向,降低盲目搜索产生的成本消耗. 基于非线性惯性权重,花粉在全局搜索中的迭代公式为:

$$x_i^{t+1} = x_i^t - k \cdot (x_i^t - x_{\text{cen}}^t) + \text{rand} \cdot U_{\min}^i + (U_{\max}^i - U_{\min}^i) \quad (5)$$

其中,  $i$  为第  $i$  个花粉个体的索引,  $x_i^t$  为索引为  $i$  的花粉个体位置,  $U_{\max}^i$  和  $U_{\min}^i$  分别为第  $i$  个花粉个体每个维度的上下限取值,  $x_{\text{cen}}^t$  指每个个体上下限维度的平均值,  $\text{rand}$  是属于  $[0, 1]$  的随机数.

### 2.2 跨代差分进化

在 FPA 局部搜索过程中,存在搜索种群丰富度差,搜索收敛速度低等问题. 差分进化策略可以扩大种群的多样性. 同时为了降低搜索成本,提高搜索效率,搜

索过程应引导花粉个体沿着有效的方向移动. 为了实现兼顾精度和收敛速度的局部搜索, 本文将跨代差分进化<sup>[27]</sup>融入 FPA-NCDE 中, 使其在保持种群多样性和勘探-开发之间取得平衡.

跨代差分进化包括基于邻域的跨代进化策略 (Neighborhood-based Cross-Generation, NCG) 和基于种群的跨代进化策略 (Population-based Cross-Generation, PCG). NCG 和 PCG 通过分析连续世代之间的差异来引导算法搜索方向接近最优解, 结合两代种群的信息可以减少搜索结果的数值振荡.

为提高 FPA-NCDE 在局部搜索过程中的搜索稳定性, 本文设计了反余弦加速因子  $c$ , 使搜索开始时更快地进入局部搜索, 结束时保持较高的搜索速度而不会提前收敛. 其定义如下:

$$c = c_{\min} - (c_{\min} - c_{\max}) \frac{1}{\pi} \arccos \left( -2 \exp \left( \sqrt{d} \frac{t}{T_{\max}} \right) + 1 \right) \quad (6)$$

其中,  $d$  为花粉个体的维数,  $t$  为迭代次数,  $T_{\max}$  为最大迭代次数. 设定  $c_{\max} = 1.0$ ,  $c_{\min} = 0.6$ .

NCG 通过计算种群中当代个体与其他个体之间的欧氏距离, 选择距离当前花粉个体最近的  $T$  个个体组成邻域池, 并基于欧氏距离选择父代种群中  $T$  个个体作为邻域池的成员, 其表达式如下:

$$V_{i,g} = cx_{r_{m_1},g} + F \cdot (x_{r_{m_1},g} - x_{r_{m_2},g-1}) \quad (7)$$

其中,  $V_{i,g}$  表示差分进化后产生的新个体,  $i$  为第  $i$  个花粉个体的索引,  $g$  是差分变异的代数,  $x_{r_{m_1},g}$  是当前代邻域池中随机选取的个体,  $x_{r_{m_2},g-1}$  是从父代个体邻域池中随机选取的个体.  $r_{m_1}$  是从  $\{I_{i,g,1}, I_{i,g,2}, \dots, I_{i,g,T}\}$  选取的随机整数,  $\{I_{i,g,1}, I_{i,g,2}, \dots, I_{i,g,T}\}$  记录了邻域池中个体的索引,  $r_{m_2}$  是从  $\{I_{i,g-1,1}, I_{i,g-1,2}, \dots, I_{i,g-1,T}\}$  随机选取的整数,  $T$  的大小为种群个体总数的 5%.

PCG 通过随机选择父代种群中的个体和当代种群中的个体参与突变过程, 来增加种群的丰富度, 其表达式如下:

$$V_{i,g} = cx_{i,g} + F \cdot (x_{r_{p_1},g} - x_{r_{p_2},g-1}) \quad (8)$$

其中,  $i$  为第  $i$  个花粉个体的索引,  $x_{i,g}$  为当代花粉个体,  $x_{r_{p_1},g}$  为当前种群中随机选取的个体,  $x_{r_{p_2},g-1}$  为父代种群中随机选取的个体,  $r_{p_1}$  和  $r_{p_2}$  为从  $\{1, 2, \dots, N\}$  中选取的随机整数,  $N$  为种群规模. 为了平衡收敛速度和搜索能力, 在变异操作中采用 PCG 和 NCG 的概率相同.

### 2.3 参数自适应调整

NCG 和 PCG 对缩放因子和交叉概率的取值非常敏感, 因此这两个参数会影响搜索过程. 如果缩放因子较小, 搜索过程产生较小的振荡, 种群的丰富度下降. 当

缩放因子较大时, 会增加搜索过程的盲目性, 收敛速度变差, 需要更多的时间来寻找最优解. 交叉概率如果较大, 可以增加种群丰富性, 如果较小会降低种群多样性并使算法收敛速度变差. 对于搜索过程的不同阶段, 不同的参数会影响算法的优化效果. 在参数初始化过程中, 缩放因子  $F_{i,g}$  和交叉概率  $CR_i$  是从  $[F_{\min}, F_{\max}]$  和  $[CR_{\min}, CR_{\max}]$  中随机产生,  $i$  为第  $i$  个花粉个体的索引,  $F_{\min}$  和  $F_{\max}$  分别为缩放因子  $F$  的上下限,  $CR_{\min}$  和  $CR_{\max}$  为交叉概率  $CR$  的上下限. 缩放因子动态更新的表达式如下:

$$F_{i,g} = \text{mean} \left( (F_{\min} + (F_{\max} - F_{\min}) \frac{f_2 - f_1}{f_3 - f_1}) + (F_{\min} + (F_{\max} - F_{\min}) \frac{f_5 - f_4}{f_6 - f_4}) \right) \quad (9)$$

从当前个体的邻域池中随机选取 3 个个体  $x_{p_1}, x_{p_2}, x_{p_3}$ , 其适应度值分别为  $f_1, f_2, f_3$  且  $f_1 < f_2 < f_3$ . 同样, 从父代个体的邻域池中随机选取 3 个个体  $x_{p_4}, x_{p_5}, x_{p_6}$ , 其适应度值分别为  $f_4, f_5, f_6$  且  $f_4 < f_5 < f_6$ .

交叉概率动态更新的表达式如下:

$$CR_{i,g} = \begin{cases} CR_{\min} + \frac{(CR_{\max} - CR_{\min})}{f_{\max} - f_{\min}}, & \text{if } f_i < f_{\text{ave}} \\ CR_{\min}, & \text{if } f_i \geq f_{\text{ave}} \end{cases} \quad (10)$$

其中,  $i$  为第  $i$  个花粉个体的索引,  $f_{\text{ave}}$  是当前种群内所有个体的平均适应度值.

### 2.4 跨代赌轮盘

启发式优化算法常用赌轮盘<sup>[28,29]</sup>来跳出局部最优解. 本文通过设计跨代赌轮盘来降低算法陷入局部解的概率, 在完成对所有个体的搜索迭代后, 选取适应度值最小的  $T$  个个体, 再从当前种群剩余个体中随机选择  $\alpha$  个个体以及随机从父代种群中选择  $\beta$  个父代个体, 共计  $2T$  个个体组成轮盘赌池, 参与轮盘赌的过程. 轮盘池每个花粉个体被选择的概率如下:

$$p_i = \frac{w_i g_i}{\sum_{i=1}^{2T} w_i g_i} \quad (11)$$

其中,  $i$  为第  $i$  个花粉个体的索引,  $w_i g_i$  是第  $i$  个花粉个体适应度值的映射权重. 跨代赌轮盘的过程如下:

- 步骤 1 父代花粉种群中的花粉个体按照适应度值进行升序排序.
- 步骤 2 选择其中排名前  $T$  个花粉个体组成子种群  $P_r$ .
- 步骤 3 从父代种群中的剩余个体中随机选择  $\beta$  个个体.
- 步骤 4 从当代种群中随机选择  $\alpha$  个个体.
- 步骤 5 将子种群  $P_r$  与选出的  $\alpha$  和  $\beta$  个个体组成赌轮盘池  $W_r$ .
- 步骤 6 根据式 (11) 将轮盘池里的个体参与赌轮

盘过程.

在跨代轮盘赌轮中,跨代信息对于花粉个体的搜索过程产生导向性的作用,有利于跳出局部最优解.同时选取的一定数量花粉个体参与赌轮盘的过程,可以兼具“全局勘探”和“局部开发”的功能.

针对传统FPA的不足,本文提出的基于非线性差分进化的花授粉算法FPA-NCDE的伪代码如算法1所示.

算法1 基于非线性跨代差分进化的花授粉算法(FPA-NCDE)

输入:种群规模大小 $N$   
 输出:种群的全局最优解.  
 Begin  
 1. 随机选择全局最优解 $g^*$ ;  
 2. 设置最大迭代次数 $T_{max}$ ,分别从 $[CR_{min}, CR_{max}]$ 和 $[F_{min}, F_{max}]$ 选择初始交叉概率 $CR_0$ 以及初始缩放因子 $F_0$ ;  
 3. 设置转换概率 $P$ ,生成随机值rand和rand1;  
 4. FOR 每一个个体 $x_i$ ;  
 5. 根据式(9)和式(10)分别计算缩放因子 $F_{i,g}$ 和交叉概率 $CR_{i,g}$ ;  
 6. IF rand <  $P$ ;  
 7. IF rand1 < 0.5;  
 8. 对于花粉个体 $x_i$ ,利用PCG策略更新后代个体 $x_{i+1}$ ;  
 9. ELSE:  
 10. 对于花粉个体 $x_i$ ,利用NCG策略更新后代个体 $x_{i+1}$ ;  
 11. ELSE:  
 12. 根据式(5)来更新产生后代个体 $x_{i+1}$ ;  
 13. 根据式(10)对更新后的个体进行交叉选择确定新的后代个体 $x_{i+1}$ ;  
 14. End IF  
 15. 利用跨代赌轮盘方式产生 $x_i^p$ ;  
 16. 将赌轮盘产生的个体 $x_i^p$ 和后代个体 $x_{i+1}$ 的适应度值进行比较,将适应度值小的个体作为后代个体 $x_{i+1}$ ;  
 17. 再将 $x_{i+1}$ 与 $x_i$ 的适应度值进行比较,将适应度值小的个体作为最终后代个体 $x_{i+1}$ 参与下次迭代;  
 18. 将后代个体 $x_{i+1}$ 与全局最优解 $g^*$ 的适应度值进行比较,更新新的全局最优解 $g^*$ ;  
 19. End FOR  
 20. 重复执行Step3~Step19直到迭代次数达到 $T_{max}$ .  
 End

### 3 FPA-NCDE算法性能测试

#### 3.1 算法性能对比

本文通过消融实验来验证各个改进机制对算法寻优精确度、稳定性以及收敛特性的影响.消融实验中所涉及的算法如表1所示,其中FPA-NCDE(w/o CRW)表示不含有跨代赌轮盘和非线性惯性权重设置机制,FPA-NCDE(w/o CDE)表示不含有跨代差分进化和参数自适应调整机制,FPA-NCDE(w/o PAD)表示不含有非线性惯性权重和参数自适应调整机制.√表示采用该机制,×表示不采用该机制.

表1 消融试验相关算法

算法	非线性惯性权重设置	跨代差分进化	参数自适应调整	跨代赌轮盘
FPA-NCDE	√	√	√	√
FPA-NCDE(w/o CRW)	×	√	√	×
FPA-NCDE(w/o CDE)	√	×	×	√
FPA-NCDE(w/o PAD)	×	√	√	√

选择包括低维函数和高维函数在内的8个测试函数<sup>[30,31]</sup>来验证跨代赌轮盘和非线性惯性权重设置对于算法跳出局部最优解,提升寻优精确度的性能.将本文提出的FPA-NCDE和FPA-NCDE(w/o CRW)与PSO、FPA、ABC、GA、SSO进行对比实验,经测试得到的各算法在对应测试函数的寻优值如表2所示.其中,加粗的数据为最优结果,带下划线的数据为次优结果.由表2可知,FPA-NCDE的寻优效果要明显优于FPA-NCDE(w/o CRW)和其他对比算法,而FPA-NCDE(w/o CRW)仅有两个测试函数的优化结果为次优解,说明跨代赌轮盘和非线性惯性权重可以引导算法跳出局部最优解,提高寻优精确度.

选择与上述实验相同的测试函数来验证跨代差分进化和参数自适应调整减少寻优结果的数值振荡、增强算法寻优稳定性的有效性.将FPA-NCDE和FPA-NCDE(w/o CDE)与PSO、FPA、ABC、GA、SSO进行消融实验,得到的各算法在对应测试函数的寻优值方差如表3所示.其中,加粗的数据为最优结果,带下划线的数据为次优结果.可以看出FPA-NCDE在8个函数中有7个函数的寻优结果的方差是最优的,并且相比次优算法,其稳定性提高了至少4个数量级,而FPA-NCDE(w/o CDE)寻优的稳定性仅略优于FPA,比多数对比算法的寻优结果的方差更大、稳定性更差.

将FPA-NCDE和FPA-NCDE(w/o PAD)与PSO、FPA、ABC、GA、SSO进行对比实验,以验证非线性惯性权重和参数自适应调整提升算法收敛速度的有效性.如表4所示,加粗的数据为最优结果,带下划线的数据为次优结果.可以看出在8个测试函数中,FPA-NCDE在5个测试函数上的迭代次数最少,而FPA-NCDE(w/o PAD)对各测试函数寻优所需的迭代次数与GA、SSO接近,仅略优于FPA.

上述消融实验结果表明,FPA-NCDE中的跨带赌轮盘和非线性惯性权重设置可以增强算法寻优的精确度;跨代差分进化和参数自适应调整可以减少数值震荡,使算法保持较好稳定性;非线性惯性权重和参数自适应调整可以使算法保持较好的收敛特性,提高算法的寻优效率.

#### 3.2 FPA-NCDE算法性能验证

本文通过3类测试函数验证FPA-NCDE的性能.

表 2 测试函数优化结果

测试函数	FPA-NCDE	FPA-NCDE (w/o CRW)	FPA <sup>[16]</sup>	ABC <sup>[10]</sup>	PSO <sup>[14]</sup>	GA <sup>[15]</sup>	SSO <sup>[11]</sup>
HARTMANN 6-DIMENSIONAL FUNCTION	<b>3.04E+00</b>	-3.01E+00	-3.01E+00	<b>-3.04E+00</b>	-3.00E+00	-3.01E+00	<u>-3.03E+00</u>
SCHAFFER FUNCTION	<b>0.00E+00</b>	5.00E-03	5.01E-03	1.53E-09	<b>0.00E+00</b>	1.08E-03	<u>3.83E-10</u>
SHEKEL FUNCTION	<b>-1.05E+01</b>	<u>-7.21E+00</u>	-7.20E+00	<b>-1.05E+01</b>	5.80E+00	-6.51E+00	-1.05E+01
ACKLEY FUNCTION	<b>9.57E-10</b>	1.48E+01	1.50E+01	<u>5.75E-06</u>	4.16E+00	1.527E+01	2.90E-01
RASTRIGIN FUNCTION	<b>0.00E+00</b>	2.33E+02	2.35E+02	<u>1.80E-02</u>	3.94E+01	2.43E+02	4.72E+01
SCHWEFEL FUNCTION	<b>3.81E-04</b>	6.76E+03	6.93E+03	1.66E+02	9.29E+03	5.92E+03	<u>6.78E+01</u>
MULTIMODAL FUNCTION	<b>1.01E+00</b>	1.15E+01	1.16E+01	<u>1.15E+00</u>	6.24E+00	1.13E+01	3.55E+00
STRECTCHED V FUNCTION	<b>1.25E-06</b>	1.31E+01	1.32E+01	<u>3.04E-01</u>	2.01E+00	1.41E+01	2.92E+00

表 3 测试函数优化结果方差

测试函数	FPA-NCDE	FPA-NCDE (w/o CDE)	FPA <sup>[16]</sup>	ABC <sup>[10]</sup>	PSO <sup>[14]</sup>	GA <sup>[15]</sup>	SSO <sup>[11]</sup>
HARTMANN 6-DIMENSIONAL FUNCTION	<b>1.04E-16</b>	2.90E-02	2.94E-02	<u>9.06E-16</u>	3.01E-02	2.82E-02	<u>2.25E-02</u>
SCHAFFER FUNCTION	<b>0.00E+00</b>	5.21E-03	5.24E-03	3.41E-09	<b>0.00E+00</b>	1.77E-03	1.00E-10
SHEKEL FUNCTION	<b>1.16E-14</b>	3.47E+00	3.49E+00	<u>6.25E-05</u>	3.37E+00	3.11E+00	1.55E-03
ACKLEY FUNCTION	<b>4.66E-09</b>	9.73E-00	9.75E-00	2.50E-00	6.54E-00	5.86E-01	<u>3.30E-02</u>
RASTRIGIN FUNCTION	<b>0.00E+00</b>	1.58E+01	1.60E+01	<u>7.43E-02</u>	8.91E+00	1.76E+01	1.00E+01
SCHWEFEL FUNCTION	<b>1.01E-12</b>	4.91E+02	4.91E+02	1.16E+02	3.60E+02	4.16E+02	<u>3.60E+01</u>
MULTIMODAL FUNCTION	1.81E+00	<u>2.88E-01</u>	2.93E-01	<b>2.00E-01</b>	8.47E-01	4.55E-01	8.16E-01
STRECTCHED FUNCTION	<b>6.28E-06</b>	1.58E+00	1.59E+00	<u>8.79E-02</u>	1.05E+00	1.22E+00	5.13E-01

表 4 测试函数优化所需迭代次数

测试函数	FPA-NCDE	FPA-NCDE (w/o PAD)	FPA <sup>[16]</sup>	ABC <sup>[10]</sup>	PSO <sup>[14]</sup>	GA <sup>[15]</sup>	SSO <sup>[11]</sup>
HARTMANN 6-DIMENSIONAL FUNCTION	293	943	960	<u>90</u>	<b>86</b>	947	999
SCHAFFER FUNCTION	<b>182</b>	982	991	999	<u>229</u>	992	990
SHEKEL FUNCTION	<u>107</u>	986	998	126	<b>74</b>	970	980
ACKLEY FUNCTION	<u>788</u>	979	998	1000	<b>526</b>	996	1000
RASTRIGIN FUNCTION	<b>229</b>	984	1000	997	<u>695</u>	999	1000
SCHWEFEL FUNCTION	<b>468</b>	982	996	998	1000	971	<u>642</u>
MULTIMODAL FUNCTION	<b>881</b>	973	1000	999	1000	966	1000
STRECTCHED V FUNCTION	<b>104</b>	976	981	1000	<u>634</u>	978	1000

第一类被测函数为低维测试函数,第二类为高维测试函数,第三类为可扩展测试函数.对比其他常见的启发式优化算法,验证 FPA-NCDE 在优化精度和收敛效果方面的情况.

3.2.1 低维度测试函数

选取如表 5 所示的含有低维度决策变量的测试函数<sup>[30,31]</sup>.第一列测试函数,其他各列一列代表不同算法的测试结果.每个算法独立运行 30 次,函数评价次数最大为  $1\ 000 \times D$  ( $D$  为基准函数决策变量的维数).统计分析 (Wilcoxon 符号秩检验显著性水平值为 0.05) 用

于比较 FPA-NCDE 和 FPA、ABC、PSO、GA、SSO 在各测试函数上得到的最优解.在表 5 中,第一列表示的是测试函数以及其中决策变量的维度大小,从第 2 到第 7 列为各算法下针对测试函数进行测试所得到的最优值,括号内数值为方差.当最优值越小表示算法的寻优精确度越好,方差越小表明算法的鲁棒性越好.表中最后一行的“+/-/-”表示 FPA-NCDE 相比其他启发式优化算法表现更好/相同/更差的测试函数数目.FPA-NCDE 算法比其他启发式算法的优化效果差的值用灰色突出表示.

表5 低维测试函数优化结果

测试函数	FPA-NCDE	FPA <sup>[16]</sup>	ABC <sup>[10]</sup>	PSO <sup>[14]</sup>	GA <sup>[15]</sup>	SSO <sup>[11]</sup>
$F_1$ : BEALE FUNCTION (维度2)	<b>9.09E-26</b> ( <b>3.68E-25</b> )	4.14E-03 (7.07E-03)	6.82E-10 (1.32E-09)	<b>0.00E+00</b> ( <b>0.00E+00</b> )	7.81E-02 (1.68E-01)	1.52E-07 (1.70E-07)
$F_2$ : BOHACHEVSKY FUNCTION (维度2)	<b>0.00E+00</b> ( <b>0.00E+00</b> )	2.25E-01 (2.27E-01)	<b>0.00E+00</b> ( <b>0.00E+00</b> )	<b>0.00E+00</b> ( <b>0.00E+00</b> )	6.20E-01 (2.95E-01)	1.93E-06 (1.42E-06)
$F_3$ : COLVILLE FUNCTION (维度4)	<b>0.00E+00</b> ( <b>0.00E+00</b> )	1.73E+00 (1.37E+00)	2.82E-01 (1.74E-01)	<b>0.00E+00</b> ( <b>0.00E+00</b> )	4.79E+01 (3.17E+01)	1.27E-03 (5.92E-04)
$F_4$ : CROSS-IN-TRAY FUNCTION (维度2)	-2.06E+00 (2.04E-11)	-2.06E+00 (1.87E-05)	-2.06E+00 (9.06E-16)	-2.06E+00 (9.06E-16)	-2.06E+00 (1.35E-15)	-2.06E+00 (1.26E-08)
$F_5$ : DROP-WAVE FUNCTION (维度2)	-1.00E+00 ( <b>0.00E+00</b> )	-9.51E-01 (2.72E-02)	-1.00E+00 ( <b>0.00E+00</b> )	-1.00E+00 ( <b>0.00E+00</b> )	-9.98E-01 (1.13E-16)	-9.99E-01 (4.68E-06)
$F_6$ : EASOM FUNCTION (维度2)	-9.99E-01 (4.84E-05)	-7.99E-01 (3.50E-01)	-9.99E-01 (8.42E-06)	-1.00E+00 (0.00E+00)	-5.45E-01 (4.93E-01)	-1.00E+00 (2.14E-07)
$F_7$ : GOLDSTEIN-PRICE FUNCTION (维度2)	<b>3.00E+00</b> (1.02E-15)	3.00E+00 (3.00E-03)	3.00E+00 (1.81E-07)	3.00E+00 (1.21E-15)	3.90E+00 (1.02E+00)	3.000E+00 (3.69E-05)
$F_8$ : HARTMANN 3-DIMEN- SIONAL FUNCTION (维度3)	-3.86E+00 (1.38E-15)	-3.86E+00 (7.53E-05)	-3.86E+00 (2.71E-15)	-3.86E+00 (2.66E-15)	-3.86E+00 (1.82E-05)	-3.86E+00 (4.26E-05)
$F_9$ : HARTMANN 6-DIMEN- SIONAL FUNCTION (维度6)	-3.04E+00 (1.04E-16)	-3.01E+00 (2.94E-02)	-3.04E+00 (9.06E-16)	-3.00E+00 (3.01E-02)	-3.01E+00 (2.82E-02)	-3.03E+00 (2.25E-02)
$F_{10}$ : HOLDER TABLE FUNCTION (维度2)	-1.92E+01 (3.83E-15)	-1.92E+01 (1.41E-03)	-1.92E+01 (7.77E-15)	-1.92E+01 (6.19E-15)	-1.92E+01 (5.22E-15)	-1.92E+01 (8.50E-07)
$F_{11}$ : MICHALEWICZ FUNCTION (维度2)	-1.80E+00 (9.81E-16)	-1.80E+00 (2.76E-04)	-1.80E+00 (6.79E-16)	-1.80E+00 (6.79E-16)	-1.80E+00 (8.77E-08)	-1.80E+00 (2.09E-06)
$F_{12}$ : PERM FUNCTION (维度2)	2.13E-03 (2.13E-03)	1.21E+01 (6.12E+00)	1.86E-02 (1.71E-02)	1.43E-01 (2.69E-01)	4.47E+06 (9.96E+06)	5.74E-02 (1.69E-01)
$F_{13}$ : SCHAFFER FUNCTION (维度2)	<b>0.00E+00</b> ( <b>0.00E+00</b> )	5.01E-03 (5.24E-03)	1.53E-09 (3.41E-09)	<b>0.00E+00</b> ( <b>0.00E+00</b> )	1.08E-03 (1.77E-03)	3.83E-10 (1.00E-10)
$F_{14}$ : SHEKEL FUNCTION (维度4)	-1.05E+01 (1.16E-14)	-7.20E+00 (3.49E+00)	-1.05E+01 (6.25E-05)	5.80E+00 (3.37E+00)	-6.51E+00 (3.11E+00)	-1.05E+01 (1.55E-03)
$F_{15}$ : SIX-HUMP CAMEL FUNCTION (维度2)	-1.03E+00 (1.26E-12)	-1.03E+00 (3.85E-05)	-1.03E+00 (2.26E-16)	-1.03E+00 (2.26E-16)	-1.03E+00 (1.26E-03)	-1.03E+00 (1.12E-06)
$F_{16}$ : SPHERE FUNCTION (维度6)	-1.94E+00 (3.26E-09)	-1.94E+00 (1.26E-05)	-1.94E+00 (4.53E-16)	-1.94E+00 (4.96E-16)	-1.94E+00 (1.07E-06)	-1.94E+00 (8.41E-06)
$F_{17}$ : THREE-HUMP CAMEL FUNCTION (维度2)	4.54E-94 (2.27E-93)	7.00E-05 (1.36E-04)	1.83E-18 (1.59E-18)	4.41E-72 (2.20E-71)	1.37E-04 (1.37E-04)	1.40E-07 (1.24E-07)
+/-/-		14/3/0	7/10/0	5/10/2	15/2/0	13/4/0

相比于 FPA, FPA-NCDE 在 14 个测试函数中表现出最好的性能, 并且其平均误差值在 17 个测试函数中均得到了明显改善, 要优于 FPA. 与 ABC 相比, FPA-NCDE 在 17 个测试函数中有 7 个性能更好. 在剩余 10 个函数中得到与 ABC 相同的最优解. 对于平均误差值, FPA-NCDE 在 12 个函数中优于 ABC, 在 2 个 ( $F_2$ 、 $F_5$ ) 函数中二者相同, ABC 仅在 3 个 ( $F_4$ 、 $F_6$ 、 $F_{11}$ ) 函数中的平均误差值要优于 FPA-NCDE. 总的来说, FPA-NCDE 的搜索精度、平均误差值和稳定性都比 ABC 好. 与 PSO 相比, FPA-NCDE 在 5 个函数值获得的最优解优于 PSO, 在 12 个函数中获得的最优解与 PSO 相同. 对于平

均误差值, FPA-NCDE 在 7 个函数中更优, 在 4 个函数上 ( $F_2$ 、 $F_3$ 、 $F_5$  和  $F_{13}$ ) 二者相同. 对于函数  $F_1$ 、 $F_4$ 、 $F_6$ 、 $F_{11}$ 、 $F_{15}$  和  $F_{16}$ , 虽然 PSO 的平均误差更小, 但 FPA-NCDE 与 PSO 差异较小. 与 GA 相比, 15 个函数中 FPA-NCDE 的优化精度优于 GA, 仅在 2 个函数中二者的优化效果相同. 从平均误差值来看, FPA-NCDE 仅在  $F_5$  上比 GA 差, 在其他函数中平均误差都要小于 GA. 与 SSO 相比, FPA-NCDE 可以在 17 个函数中提供比 SSO 更好或相同的最优解, 并且仅在  $F_8$  的平均误差要比 GA 略差.

结果表明针对低维测试函数, FPA-NCDE 具有更小的数值振荡和更高的稳定性, 而且提高了搜索过程中

优化精度的可靠性.

### 3.2.2 高维度测试函数

选取如表 6 含有高度决策变量的测试函数<sup>[30,31]</sup>, 其余设置与上一组实验相同. 与 FPA 相比, FPA-NCDE 获得的最优解在所有函数上均优于 FPA、PSO、GA、SSO 算法. 在平均误差上, FPA-NCDE 仅在 2 个测试函数上的效果比 FPA 差, 仅有 1 个函数的平均误差比 PSO 和 SSO 差, 而在所有函数上的平均误差都比 GA 好. 相比于 ABC, FPA-NCDE 在 12 个测试函数上取得的最优解更好, 有 1 个二者相同, 有 1 个 ABC 的

效果更优. 在平均误差上, ABC 在 3 个函数上的效果略优于 FPA-NCDE. 此外, FPA-NCDE 在 3 个测试函数上的平均误差为 0, 其稳定性要明显优于其他算法. 在 14 个测试函数上, 相比其他算法的寻优效果, FPA-NCDE 在 5 个函数( $F_{19}$ 、 $F_{22}$ 、 $F_{27}$ 、 $F_{28}$ 、 $F_{32}$ )上至少要提升 10 个数量级, 在 2 个函数( $F_{29}$ 、 $F_{31}$ )上至少提升了 5 个数量级, 在 3 个函数( $F_{30}$ 、 $F_{26}$ 、 $F_{21}$ )上可以直接得到理论上的全局最优解. 综上所述, 针对多数高维测试函数, FPA-NCDE 相比其他寻优算法的寻优效果更加显著.

表 6 高维测试函数优化结果

测试函数	FPA-NCDE	FPA <sup>[16]</sup>	ABC <sup>[10]</sup>	PSO <sup>[14]</sup>	GA <sup>[15]</sup>	SSO <sup>[11]</sup>
$F_{19}$ : ACKLEY FUNCTION (维度 30)	<b>9.57E-10</b> ( <b>4.66E-09</b> )	1.50E+01 (9.75E-00)	5.75E-06 (2.50E-00)	4.16E+00 (6.54E-00)	1.52E+01 (5.86E-01)	2.90E-01 (3.30E-02)
$F_{20}$ : DIXON-PRICE FUNCTION (维度 30)	1.90E-02 (9.01E-02)	3.28E+04 (1.87E+04)	<b>1.04E-02</b> ( <b>5.76E-03</b> )	6.99E+00 (3.34E+00)	3.55E+00 (1.08E+05)	2.54E+00 (9.01E-01)
$F_{21}$ : GRIEWANK FUNCTION (维度 30)	<b>0.00E+00</b> ( <b>0.00E+00</b> )	7.88E+01 (2.09E+01)	5.00E-04 (2.50E-03)	8.59E-01 (4.14E-01)	1.89E+02 (2.50E+01)	1.86E-02 (1.26E-02)
$F_{22}$ : LEVY FUNCTION (维度 30)	<b>4.46E-24</b> ( <b>2.23E-23</b> )	3.56E+01 (8.01E+00)	4.87E-14 (3.32 E-14)	1.53E+00 (9.84E-01)	7.06E+01 (9.29E+00)	1.47E-01 (2.34E-01)
$F_{23}$ : MULTIMODAL FUNCTION (维度 30)	<b>1.01E+00</b> ( <b>1.81E+00</b> )	1.16E+01 (2.93E-01)	1.15E+00 (2.00 E-01)	6.24E+00 (8.47E-01)	1.13E+01 (4.55E-01)	3.55E+00 (8.16E-01)
$F_{24}$ : POWELL FUNCTION (维度 30)	<b>8.79E-05</b> ( <b>4.26E-04</b> )	4.92E+02 (1.86E+02)	4.36E-02 (1.32E-02)	4.90E+00 (3.21E+00)	4.09E+03 (1.06E+03)	1.61E+00 (5.50E-01)
$F_{25}$ : RANA FUNCTION (维度 30)	-1.28E+04 (1.50E+03)	-4.94E+03 (3.81E+02)	-1.22E+04 (2.28E+02)	-3.24E+03 (3.51E+02)	-5.67E+03 (1.89E+03)	-7.98E+03 (8.43E+02)
$F_{26}$ : RASTRIGIN FUNCTION (维度 30)	<b>0.00E+00</b> ( <b>0.00E+00</b> )	2.35E+02 (1.60E+01)	1.80E-02 (7.43E-02)	3.94E+01 (8.91E+00)	2.43E+02 (1.76E+01)	4.72E+01 (1.00E+01)
$F_{27}$ : ROSENBROCK FUNCTION (维度 30)	<b>6.48E-15</b> ( <b>2.28E-14</b> )	2.34E+04 (1.27E+04)	2.74E-01 (2.09E-01)	1.85E+02 (6.63E+01)	2.52E+05 (9.13E+04)	6.78E+01 (3.60E+01)
$F_{28}$ : ROTATED FUNCTION (维度 30)	<b>1.07E-29</b> ( <b>5.37E-29</b> )	4.72E+04 (1.31E+04)	1.64E-11 (1.44E-11)	3.37E+00 (2.87E+00)	1.20E+05 (2.32E+04)	1.18E+00 (2.79E-01)
$F_{29}$ : SCHWEFEL FUNCTION (维度 30)	<b>3.81E-04</b> ( <b>1.01E-12</b> )	6.93E+03 (4.91 E+02)	1.66E+02 (1.16E+02)	9.29E+03 (3.60E+02)	5.92E+03 (4.16E+02)	6.78E+01 (3.60E+01)
$F_{30}$ : STEP FUNCTION (维度 30)	<b>0.00E+00</b> ( <b>0.00E+00</b> )	8.94E+03 (2.33E+03)	<b>0.00E+00</b> ( <b>0.00E+00</b> )	6.54E+01 (4.53E+01)	2.20E+04 (2.86E+03)	2.40E-01 (4.35E-01)
$F_{31}$ : STRETCHED V FUNCTION (维度 30)	<b>1.25E-06</b> ( <b>6.28E-06</b> )	1.32E+01 (1.59E+00)	3.04E-01 (8.79E-02)	2.01E+00 (1.05E+00)	1.41E+01 (1.22E+00)	2.92E+00 (5.13E-01)
$F_{32}$ : TRID FUNCTION (维度 20)	<b>7.31E-57</b> ( <b>3.65E-56</b> )	1.25E+02 (1.25E+02)	2.43E+02 (2.67E+01)	9.55E-01 (4.84E-01)	2.52E+08 (4.65E+08)	5.02E-01 (1.13E-01)
+/-/-		14/0/0	12/1/1	14/0/0	14/0/0	14/0/0

### 3.2.3 可扩展维度测试函数

为了进一步验证本文算法在测试函数下不同维度决策向量下的优化效果. 选择可扩展维度函数  $F_{33}$ <sup>[32]</sup> 和  $F_{34}$ <sup>[33]</sup> 作为测试函数. 对于函数  $F_{33}$ , 它有一个全局最优解  $(0, 0, \dots, 0)_n$ , 在全局最优解和局部最优解之间存在极大解. 因此, 在优化算法的搜索过程中容易陷入局部最

优, 同时易造成算法全局收敛速度慢, 难以搜索得到全局最优解.  $F_{34}$  的全局最优解被大量具有相同函数值的局部最优解包围, 会造成寻优过程产生强烈的数值振荡现象. 针对  $F_{33}$ , 分别取决策变量维度为 10、30、50、70 和 100, 测试结果如表 7. 针对  $F_{34}$ , 分别取决策变量为 2、30、50、70 和 100, 测试结果如表 8.

表7 不同维度下  $F_{33}$  的优化结果

算法	$D=10$	$D=30$	$D=50$	$D=70$	$D=100$
FPA-NCDE	7.98E-03 (2.76E-02)	1.19E-02 (3.31E-02)	7.98E-03 (2.76E-02)	1.19E-02 (3.31E-02)	1.19E-02 (3.31E-02)
FPA <sup>[16]</sup>	2.89E-01 (5.25E-02)	1.18E+00 (1.76E-01)	1.91E+00 (2.09E-01)	2.51E+00 (2.33E-01)	3.12E+00 (3.38E-01)
ABC <sup>[10]</sup>	2.55E-01 (6.50E-02)	1.50E+00 (1.78E-01)	2.70E+00 (1.48E-01)	3.64E+00 (1.22E-01)	4.89E+00 (1.35E-01)
PSO <sup>[14]</sup>	1.59E-01 (5.00E-02)	4.35E-01 (6.37E-02)	6.67E-01 (7.48E-02)	8.39E-01 (8.66E-02)	1.06E+00 (7.57E-02)
GA <sup>[15]</sup>	6.45E-01 (1.10E-01)	1.78E+00 (1.36E-01)	2.60E+00 (1.28E-01)	3.38E+00 (1.60E-01)	4.27E+00 (1.17E-01)
SSO <sup>[11]</sup>	9.98E-02 (2.54E-10)	3.03E-01 (4.54E-02)	5.11E-01 (3.31E-02)	6.35E-01 (4.89E-02)	7.59E-01 (5.00E-02)

$F_{33}$  和  $F_{34}$  在低维度的情况下,部分启发式算法与 FPA-NCDE 的寻优精度接近. 随着维度的增加, FPA-NCDE 算法的寻优精度要明显优于其他启发式算法. 这证明了随着测试函数变量维度的增大,测试函数产生的

局部最优解的数量增多,其他对比算法在搜索最优解的过程中缺乏导向性,导致寻找最优解的精确度下降. 同时由于局部最优解数量庞大,跳出局部最优解的难度增加,因此其他算法的寻优精度要比 FPA-NCDE 差.

表8 不同维度下  $F_{34}$  的优化结果

算法	$D=2$	$D=30$	$D=50$	$D=70$	$D=100$
FPA-NCDE	-1.00E+00 (0.00E+00)	-9.95E-01 (4.95E-03)	-9.97E-01 (4.45E-03)	-9.94E-01 (4.92E-03)	-9.95E-01 (4.92E-03)
FPA <sup>[16]</sup>	-9.93E-01 (4.57E-03)	-9.41E-01 (2.06E-02)	8.96E-01 (2.06E-02)	-8.70E-01 (2.27E-02)	-8.16E-01 (3.00E-02)
ABC <sup>[10]</sup>	-1.00E+00 (0.00E+00)	-9.21E-01 (3.56E-11)	-8.73E-01 (8.47E-10)	-8.21E-01 (1.56E-10)	-7.66E-01 (1.49E-02)
PSO <sup>[14]</sup>	-1.00E+00 (0.00E+00)	-9.63E-01 (5.50E-03)	-9.59E-01 (1.13E-02)	-9.31E-01 (1.78E-02)	-9.21E-01 (1.13E-16)
GA <sup>[15]</sup>	-9.99E-01 (3.39E-16)	-9.21E-01 (2.59E-07)	1.14E+00 (5.70E-02)	1.45E+00 (6.37E-02)	1.78E+00 (6.20E-02)
SSO <sup>[11]</sup>	-9.99E-01 (1.94E-03)	-9.90E-01 (1.13E-10)	-9.86E-01 (9.12E-03)	-9.63E-01 (5.50E-03)	-9.62E-01 (9.14E-10)

在同维度的测试函数下, FPA-NCDE 整体上都具备更优的搜索精度和更好的平均误差. 在低维测试函数情况下,存在 ABC 和 PSO 的部分寻优精度与 FPA-NCDE 相接近的情况. 此时多数测试函数由于变量的维度较低,在进行搜索最优解的过程中,出现的局部最优解数量较少,容易跳出局部最优解. 但随着维度的增加,局部最优解数量会出现指数级增加,并缺乏搜索导向性,无法平衡局部搜索和全局搜索过程,此时 FPA-NCDE 的优越性使算法的搜索精度得到提升,同时相比其他启发式优化算法,不易陷入局部最优解.

## 4 算法应用及性能评估

由于无人机智能巡检的成本低、灵活性好,利用无人机巡检设备代替传统的人工,实现大范围的安全检

查成为当下研究的热点. 无人机智能巡检路径规划主要是根据决策变量在工作场所设计一条飞行航迹. 本文的决策变量考虑飞行约束条件以及飞行代价函数.

### 4.1 飞行约束条件

#### 4.1.1 飞行长度限制

无人机每次转向前都存在一个直线距离进行误差修正. 最小飞行长度影响无人机航迹. 假设飞行航迹起点为  $S$ , 终点为  $E$ , 将  $SE$  分段为  $D+1$  段. 飞行的总长度遵循如下条件:

$$\sum_{m=1}^{D+1} \|P_{m-1}P_m\| \leq L_{\max} \quad (12)$$

其中,  $\|P_{m-1}P_m\|$  是各分段的长度,  $P_0$  是起点,  $P_{D+1}$  是终点,  $L_{\max}$  是总飞行长度的最大值.

#### 4.1.2 转弯角度限制

无人机在相邻两段分段之间的转弯角度遵循以下条件.

$$\frac{(P_{m-1}P_m) \cdot (P_mP_{m+1})}{\|P_{m-1}P_m\| \cdot \|P_mP_{m+1}\|} \geq \cos \theta_{\max} \quad (13)$$

其中,  $\theta_{\max}$  是最大转向角.

#### 4.1.3 飞行高度限制

无人机的飞行高度  $h$  遵循以下条件:

$$H_{\min} < h < H_{\max} \quad (14)$$

$H_{\max}$  为最大飞行高度,  $H_{\min}$  为最小飞行高度.

#### 4.2 飞行代价函数

基于上述约束条件, 考虑能耗成本、转向成本、威胁攻击成本和飞行高度成本作为无人机路径规划算法的性能评估标准.

##### 4.2.1 能耗成本

能耗成本  $Cost_e$  与无人机的航速和总飞行长度有关, 其表达式如下:

$$Cost_e = \eta \cdot \frac{v}{3} \cdot l_m \quad (15)$$

其中,  $\eta$  为能耗因子,  $v$  是无人机的速度,  $l_m$  是飞机第  $m$  个分段航迹的长度.

##### 4.2.2 转向成本

转向成本与转向角有关, 式(16)给出了转向成本  $Cost_s$  的计算.

$$Cost_s = \begin{cases} \theta_j, & 0 \leq \theta_j \leq \frac{1}{3}\theta_{\max} \\ \frac{k}{2}\theta_j, & \frac{1}{3}\theta_{\max} < \theta_j \leq \frac{2}{3}\theta_{\max} \\ k^2\theta_j, & \frac{2}{3}\theta_{\max} < \theta_j \leq \theta_{\max} \end{cases} \quad (16)$$

其中,  $\theta_j$  是转向角,  $k$  是转向系数,  $\theta_{\max}$  是最大转向角度.

##### 4.2.3 威胁攻击成本

飞行轨迹易受到通信雷达、基站和建筑物等威胁目标的影响, 需考虑威胁攻击代价  $Cost_t$ . 各航迹点的威胁程度表示为:

$$f_j(x) = \begin{cases} \frac{a^2 k_j^2}{d_{xy}^2}, & d_{xy} \leq R_j \\ \frac{k_j}{d_{xy}^2}, & R_j < d_{xy} \leq bR_j \\ 0, & d_{xy} > bR_j \end{cases} \quad (17)$$

其中,  $k_j$  为第  $j$  个威胁目标的威胁等级,  $a$  和  $b$  为威胁权重,  $d_{xy}$  是指无人机在点  $x$  距第  $j$  个威胁目标中心的距离,  $R_j$  表示威胁目标的半径. 为了计算每个威胁目标在各分段路径上的无人机所耗费的威胁攻击成本, 将每一分段的长度平均分为 15 个部分. 每一分段的威胁攻击

成本是所在位置  $x$  为  $\frac{2}{15}, \frac{4}{15}, \frac{6}{15}, \frac{8}{15}, \frac{10}{15}, \frac{12}{15}, \frac{14}{15}$  的威胁攻击成本的平均值, 如图 1 所示. 则威胁攻击成本  $Cost_t$  为:

$$Cost_t = \frac{l_m}{7} \sum_{m=1}^N \left[ \sum_{n=1}^7 f_n \left( \frac{2j}{15} l_m \right) \right] \quad (18)$$

$l_m$  为第  $m$  个飞行航迹段的长度,  $N$  为威胁目标的数量.

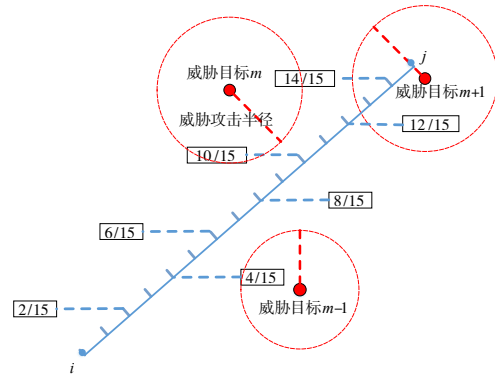


图 1 威胁攻击成本计算模型

##### 4.2.4 飞行高度成本

飞行高度成本与各分段的飞行高度有关, 飞行高度成本  $Cost_h$  为:

$$Cost_h = \mu (h_i - H_{\text{best}}) \quad (19)$$

其中,  $\mu$  为飞行高度因子,  $h_m$  为无人机在第  $m$  个分段的飞行高度,  $h_{\text{best}}$  为最佳飞行高度. 综上, 无人机巡检路径规划的代价函数为:

$$f(x) = \sum_{i=1}^{D+1} (\lambda_1 Cost_t + \lambda_2 Cost_e + \lambda_3 Cost_h) + \lambda_4 \sum_{i=1}^D Cost_s$$

其中,  $x$  表示无人机在路径规划上飞行轨迹点的三维坐标,  $\lambda_1, \lambda_2, \lambda_3, \lambda_4$  分别表示  $Cost_t, Cost_e, Cost_h, Cost_s$  的权重.

#### 4.3 无人机路径规划性能

将本文所设计的算法 FPA-NCDE, FPA, ABC, PSO, GA, SSO 用于路径规划中, 其中飞行高度因子  $\mu$  设为 1.5, 威胁因子  $a$  设为 1.2, 威胁因子  $b$  设为 1.5, 转向系数  $k$  设为 2.5, 最大转向角度  $\theta_{\max}$  设为  $60^\circ$ , 最佳飞行高度  $H_{\text{best}}$  设为 125 m, 最小飞行高度  $H_{\min}$  设为 100 m, 最大飞行高度  $H_{\max}$  设为 200 m. 最大飞行长度  $L_{\max}$  设为 2 000 m, 威胁攻击成本权重  $\lambda_1$  设为 0.5, 能源消耗成本权重  $\lambda_2$  设为 0.3, 飞行高度成本权重  $\lambda_3$  设为 0.5, 转向成本权重  $\lambda_4$  设为 0.3, 威胁目标数量  $N$  设为 3, 分段数量  $D$  设为 15, 能耗因子  $\eta$  设为 0.5.

为了使得算法的运行结果具有代表性, 设定每个算法独立运行 1 000 次, 并将这 1 000 次取得的代价函数平均作为优化结果, 用来验证无人机路径规划的

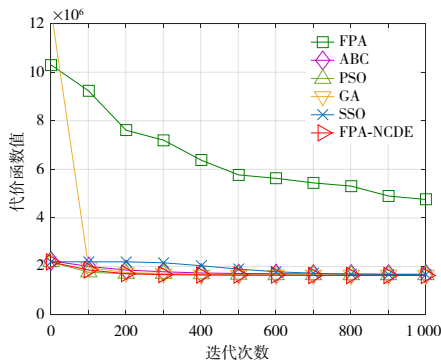
寻优性能. 由表9可知, FPA-NCDE优化效果最好, 代价函数值可稳定在  $1.63\text{E}+06$ . 与 FPA、ABC、PSO、SSO、GA 相比, 其优化效果的精度分别提高 65.7%、2.8%、1.7%、1.7%、0.9%. FPA-NCDE 的寻优波动性较小, 算法平稳. 同时在算法的运行时间上, 仅比耗时最短的 PSO 多 0.35 s. 因此 FPA-NCDE 能以较短的运行时间保证较高的寻优效率, 降低了无人机飞行成本, 并降低了飞行过程中受威胁攻击和环境因素的影响.

图2为代价函数收敛曲线及其局部放大图. 由图2(a)可知, FPA存在过早收敛的缺点. 由图2(b)可知, GA和PSO在迭代次数为400和600时便停止最优值的搜索, 而FPA-NCDE达到最优值时的迭代次数为700

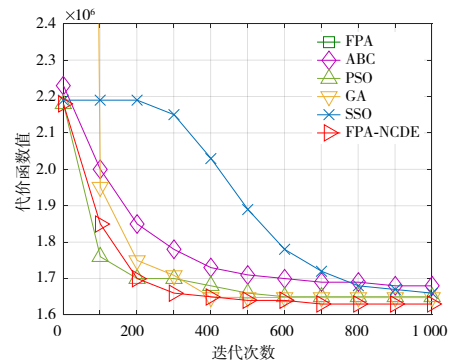
表9 路径规划的优化结果

算法	代价函数平均值	平均误差	算法运行时间/s
FPA-NCDE	1.63E+06	2.92E+03	20.35
FPA <sup>[16]</sup>	4.76E+06	1.69E+06	25.32
ABC <sup>[10]</sup>	1.67E+06	3.07E+03	20.33
PSO <sup>[14]</sup>	1.65E+06	8.06E+03	20.00
GA <sup>[15]</sup>	1.66E+06	1.54E+04	21.52
SSO <sup>[11]</sup>	1.64E+06	2.92E+03	20.45

次, 避免了提前收敛. 总体而言, FPA-NCDE 代价函数收敛速度优于 ABC、FPA 和 SSO, 略差于 PSO 和 GA, 但 FPA-NCDE 的优化结果优于其他算法.



(a) 代价函数收敛曲线

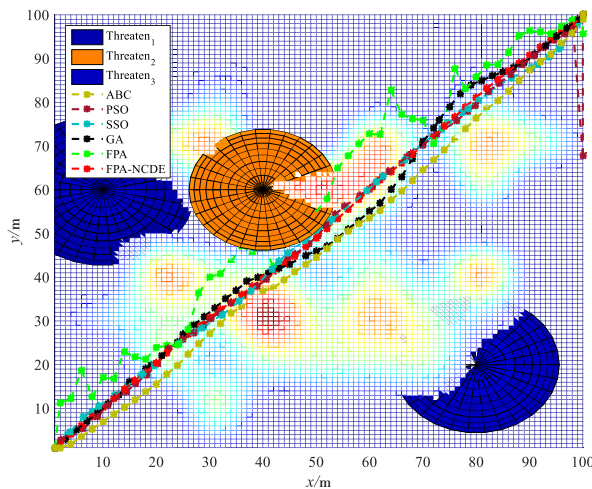


(b) 代价函数收敛曲线局部放大图

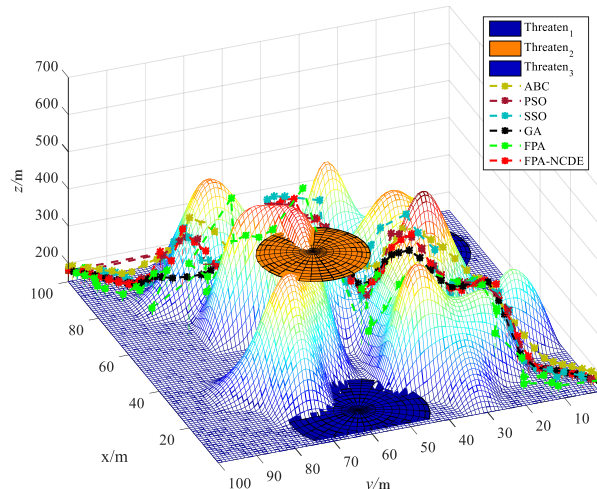
图2 代价函数收敛曲线

图3为无人机在一定区域内的路径规划图. 需要规划一条从区域左下角至右上角的路径, 其中山体为飞行过程中遇到的障碍物, 各色图标表示威胁目标. 从图3(a)所示的俯视图中可以看出, SSO和PSO算法在

寻优过程中波动性剧烈, 在寻优终点处发生巨大波动, FPA-NCDE在寻优过程中的稳定性更好, 曲线光滑, 波动性小, 优化效果要优于其他算法. 从图3(b)所示的三维图可以更为直观的表现出FPA-NCDE整体规避障碍



(a) 无人机路径规划俯视图



(b) 无人机路径规划三维图

图3 无人机路径规划图

物的效果更好,其飞行轨迹较为平滑,不会出现转向幅度较大的现象,整体的飞行效果更好。

## 5 结论

本文基于花授粉算法,设计了一种基于非线性跨代差分进化的花授粉算法,用于实现无人机智能巡检的路径规划任务。本算法相比于传统启发式优化算法,具有更好的寻优性能,可以根据飞行环境指导无人机规避障碍物和外界攻击的同时,降低飞行能耗并缩短飞行距离。仿真结果证明,本算法在寻优精确度、算法寻优平均误差,以及算法的稳定性上都体现出明显的性能优势。本文提出的优化算法不仅能解决低维度变量优化问题,在复杂的高维度变量优化问题上有着较好的性能,未来有望用于无人潜航器路径设计、汽车货物配送路线规划、无人机智能电网故障监测等工业场景之中。

## 参考文献

- [1] 任姚丹琚,戚正伟,管海兵,等.工业互联网边缘智能发展现状与前景展望[J].中国工程科学,2021,23(2):104-111.  
REN Y D J, QIZ W, GUAN H B, et al. Development and prospect of edge intelligence for industrial Internet[J]. Strategic Study of CAE, 2021, 23(2): 104-111. (in Chinese)
- [2] 工信部.工信部发布“5G+工业互联网”10个典型应用场景和5个重点行业实践情况[J].智能制造,2021(3):12.
- [3] 陈健瑞,王景璟,侯向往,等.挺进深蓝:从单体仿生到群体智能[J].电子学报,2021,49(12):2458-2467.  
CHEN J R, WANG J J, HOU X W, et al. Advance into ocean: From bionic monomer to swarm intelligence[J]. Acta Electronica Sinica, 2021, 49(12): 2458-2467. (in Chinese)
- [4] 周鹏,董朝轶,陈晓艳,等.基于阶梯式Tent混沌和模拟退火的樽海鞘群算法[J].电子学报,2021,49(9):1724-1735.  
ZHOU P, DONG C Y, CHEN X Y, et al. A salp swarm algorithm based on stepped tent chaos and simulated annealing[J]. Acta Electronica Sinica, 2021, 49(9): 1724-1735. (in Chinese)
- [5] 谢承旺,张飞龙,陆建波,等.一种多策略协同的多目标萤火虫算法[J].电子学报,2019,47(11):2359-2367.  
XIE C W, ZHANG F L, LU J B, et al. Multi-objective firefly algorithm based on multiply cooperative strategies[J]. Acta Electronica Sinica, 2019, 47(11): 2359-2367. (in Chinese)
- [6] 张新明,王霞,康强,等.GWO与ABC的混合优化算法及其聚类优化[J].电子学报,2018,46(10):2430-2442.  
ZHANG X M, WANG X, KANG Q, et al. Hybrid grey wolf optimizer with artificial bee colony and its application to clustering optimization[J]. Acta Electronica Sinica, 2018, 46(10): 2430-2442. (in Chinese)
- [7] 董明,李敬,索永录,等.基于改进果蝇优化算法正交匹配追踪的超声信号降噪方法[J].电子学报,2022,50(2):493-501.  
DONG M, LI J, SUO Y L, et al. An improved fruit fly optimization algorithm based orthogonal matching pursuit for ultrasonic noise reduction[J]. Acta Electronica Sinica, 2022, 50(2): 493-501. (in Chinese)
- [8] 郝少伟,李勇军,赵尚弘,等.基于改进粒子群算法的多载波NOMA功率分配策略[J].电子学报,2020,48(10):2009-2016.  
HAO S W, LI Y J, ZHAO S H, et al. Multicarrier NOMA power allocation strategy based on improved particle swarm optimization algorithm[J]. Acta Electronica Sinica, 2020, 48(10): 2009-2016. (in Chinese)
- [9] 吕阳,钱斌,胡蓉,等.增强人工蜂群算法求解半导体最终测试调度问题[J].电子学报,2021,49(9):1708-1715.  
LÜ Y, QIAN B, HU R, et al. Enhanced artificial bee colony algorithm to solve semiconductor final test scheduling problem [J]. Acta Electronica Sinica, 2021, 49(9): 1708-1715. (in Chinese)
- [10] KARABOGA D, BASTURK B. On the performance of artificial bee colony (ABC) algorithm[J]. Applied Soft Computing, 2008, 8(1): 687-697.
- [11] CHANDRAN T R, REDDY A V, JANET B. An effective implementation of social spider optimization for text document clustering using single cluster approach[C]//2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT). Piscataway: IEEE, 2018: 508-511.
- [12] CHEN X Y, TANG C J, WANG J, et al. A novel hybrid wolf pack algorithm with harmony search for global numerical optimization[C]//2017 3rd IEEE International Conference on Computer and Communications (ICCC). Piscataway: IEEE, 2018: 2164-2169.
- [13] GU W. An improved whale optimization algorithm with cultural mechanism for high-dimensional global optimization problems[C]//2020 IEEE International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA). Piscataway: IEEE, 2020: 1282-1286.
- [14] EBERHART R, KENNEDY J. A new optimizer using particle swarm theory[C]//MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science. Piscataway: IEEE, 2002: 39-43.
- [15] CHENG Y F, SHAO W, ZHANG S J, et al. An improved multi-objective genetic algorithm for large planar array thinning[J]. IEEE Transactions on Magnetics, 2016, 52(3): 1-4.
- [16] YANG X S. Flower pollination algorithm for global optimization[M]//Unconventional Computation and Natural Computation. Berlin, Heidelberg: Springer Berlin Heidelberg,

- 2012: 240-249.
- [17] DANESHVAR DEHNAVI S, SHIRANI A, MEHRJERDI H, et al. New deep learning-based approach for wind turbine output power modeling and forecasting[J]. IEEE Transactions on Industry Applications (Early Access), 2020, (99): 1.
- [18] SOMPRACHA C, RUKKAPHAN S. Fractional-order system identification of temperature process rig control system using flower pollination algorithm[C]//2021 9th International Electrical Engineering Congress (iEECON). Piscataway: IEEE, 2021: 309-312.
- [19] SOMPRACHA C. Design of optimal PIDuAv controller for temperature process control system by flower pollination algorithm[C]//2021 9th International Electrical Engineering Congress (iEECON). Piscataway: IEEE, 2021: 313-316.
- [20] NGUYEN T T, SHIEH C S, HORNG M F, et al. Parallelized flower pollination algorithm with a communication strategy [C]//2015 Seventh International Conference on Knowledge and Systems Engineering (KSE). Piscataway: IEEE, 2016: 103-107.
- [21] PANDA K P, BANA P R, PANDA G. FPA optimized selective harmonic elimination in symmetric—asymmetric reduced switch cascaded multilevel inverter[J]. IEEE Transactions on Industry Applications, 2020, 56(3): 2862-2870.
- [22] TARCEWSKI T, GRZESIAK L M. An application of novel nature-inspired optimization algorithms to auto-tuning state feedback speed controller for PMSM[J]. IEEE Transactions on Industry Applications, 2018, 54(3): 2913-2925.
- [23] 李会荣, 彭娇. 带有非线性惯性权重和柯西变异的粒子群优化算法[J]. 计算机与数字工程, 2021, 49(7): 1325-1329, 1362.  
LI H R, PENG J. Improved particle swarm optimization algorithm with nonlinear inertia weight and cauchy mutation [J]. Computer & Digital Engineering, 2021, 49(7): 1325-1329, 1362. (in Chinese)
- [24] 王生亮, 刘根友. 一种非线性动态自适应惯性权重 PSO 算法 [J]. 计算机仿真, 2021, 38(4): 249-253, 451.  
WANG S L, LIU G Y. A nonlinear dynamic adaptive inertial weight particle swarm optimization[J]. Computer Simulation, 2021, 38(4): 249-253, 451. (in Chinese)
- [25] YENDURIG, NARALASETTI V. A nonlinear weight-optimized maintainability index of software metrics by grey wolf optimization[J]. International Journal of Swarm Intelligence Research, 2021, 12(2): 1-21.
- [26] YAN D P, CAO H, YU Y J, et al. Single-objective/multiobjective cat swarm optimization clustering analysis for data partition[J]. IEEE Transactions on Automation Science and Engineering, 2020, 17(3): 1633-1646.
- [27] QIU X, XU J X, TAN K C, et al. Adaptive cross-generation differential evolution operators for multiobjective optimization[J]. IEEE Transactions on Evolutionary Computation, 2016, 20(2): 232-244.
- [28] GAO H, FU Z, PUN C M, et al. An efficient artificial bee colony algorithm with an improved linkage identification method[J]. IEEE Transactions on Cybernetics, 2022, 52(6): 4400-4414.
- [29] COBOS C, MENDOZA M, LEÓN E. A hyper-heuristic approach to design and tuning heuristic methods for web document clustering[C]//2011 IEEE Congress of Evolutionary Computation (CEC). Piscataway: IEEE, 2011: 1350-1358.
- [30] BREST J, GREINER S, BOSKOVIC B, et al. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems[J]. IEEE Transactions on Evolutionary Computation, 2006, 10(6): 646-657.
- [31] ZHANG J Q, SANDERSON A C. JADE: Adaptive differential evolution with optional external archive[J]. IEEE Transactions on Evolutionary Computation, 2009, 13(5): 945-958.
- [32] SRINIVAS M, PATNAIK L M. Adaptive probabilities of crossover and mutation in genetic algorithms[J]. IEEE Transactions on Systems, Man, and Cybernetics, 1994, 24(4): 656-667.
- [33] JIANG B L W. Optimizing complex functions by chaos search[J]. Cybernetics and Systems, 1998, 29(4): 409-419.

#### 作者简介



梁 靓 女, 1981 年 9 月出生于重庆市, 现为重庆大学副教授, 博士/硕士生导师, 主要研究方向为新一代移动通信、可信物联网等。  
E-mail: liangliang@cqu.edu.cn

魏亚星 男, 1996 年 3 月出生于内蒙古乌兰察布市, 现为重庆大学微电子与通信工程学院硕士研究生, 主要研究方向为网络性能优化、网络智能。

E-mail: wyx@cqu.edu.cn

李义鑫 男, 1992 年 10 月出生于重庆市, 现为重庆大学微电子与通信工程学院博士研究生, 主要研究方向为区块链、可信评估等。

E-mail: liyixin@cqu.edu.cn

贾云健 男, 1976 年 8 月出生于山西省太原市, 现为重庆大学教授, 博士/硕士生导师, 主要研究方向为通信与计算融合、新一代移动通信等。

E-mail: yunjian@cqu.edu.cn