

doi:10.11835/j.issn.1000-582X.2018.11.011

# 应用小生境混沌搜索策略的花朵授粉算法

陈西成<sup>a</sup>, 刘 曙<sup>b</sup>, 范兵兵<sup>a</sup>

(空军工程大学 a.研究生院; b.防空反导学院, 西安 710050)

**摘要:**针对花朵授粉算法(FPA, flower pollination algorithm)存在的全局收敛能力不足、寻优精度低、易早熟等局限,提出一种应用小生境混沌搜索策略的花朵授粉算法(NCFPA, flower pollination algorithm with niche chaotic search strategy)。为增加算法搜索的广度,使用小生境技术保持种群的多样性,提升了算法的全局优化能力;引入逻辑自映射函数产生的混沌序列对精英个体进行局部优化,增强算法的搜索精度;通过经典测试函数对算法性能进行测试。测试结果表明,与花朵授粉算法、差分进化(DE, differential evolution)和蝙蝠算法(BA, bat algorithm)相比,NCFPA表现出较优的全局寻优能力和搜索精度。

**关键词:**花朵授粉算法;小生境;混沌优化;逻辑自映射

**中图分类号:**TP301.6

**文献标志码:**A

**文章编号:**1000-582X(2018)11-092-08

## Flower pollination algorithm with niche chaotic search strategy

CHEN Xicheng<sup>a</sup>, LIU Shu<sup>b</sup>, FAN Bingbing<sup>a</sup>

(a. Graduate School; b. Air and Missile Defence College, Air force Engineering University, Xi'an 710051, P.R.China)

**Abstract:** In view of the limitations of Flower Pollination Algorithm (FPA) such as insufficient global convergence ability, low searching accuracy and premature maturity, a flower pollination algorithm using a niche chaotic search strategy was proposed. In order to increase the breadth of algorithm search, niche technology was applied to keep the diversity of the population and the global optimization ability of the algorithm was improved. The chaotic sequence generated by the logic self-mapping function was introduced to locally optimize the elites of the algorithm. The performance of the algorithm was tested by classic test functions. The results show that compared with Flower Pollination Algorithm, Differential Evolution(DE) and Bat Algorithm(BA), NCFPA has better global search ability and search accuracy.

**Keywords:** flower pollination algorithm; niche; chaos optimization; logic self-mapping function

花朵授粉算法(FPA, flower pollination algorithm)是一种新型的元启发式智能优化算法<sup>[1]</sup>,它采用莱维飞行机制模拟植物通过鸟类、蜜蜂等进行异花传粉的过程,并通过控制参数完成植物异花传粉和自花传粉之间的转换。FPA算法实现简单、寻优解构新颖、参数较少,具有较大的研究价值,现已成功应用于多个领

**收稿日期:**2018-07-16

**基金项目:**国家自然科学基金青年科学基金资助项目(61703412);国家自然科学基金资助项目(61402517)。

Supported by National Natural Science Foundation of China (61703412) and National Natural Science Foundation of China(61402517).

**作者简介:**陈西成(1995—),男,主要从事智能信息处理研究。

刘曙(联系人),男,副教授,主要从事智能信息处理与软件工程研究,(E-mail) 1139899855@qq.com。

域<sup>[2-4]</sup>。Yang 等<sup>[2]</sup>用其解决多目标优化问题,得到满意的结果;Rodrigues<sup>[3]</sup>提出一种二进制编码的花朵授粉算法,并用于解决特征提取问题;王生生等<sup>[4]</sup>将 FPA 与 K-means 相结合,并将新算法成功用于解决混合数据聚类问题。

但是,FPA 在实际应用中仍存在易早熟、全局收敛能力不足、搜索精度低等问题。为此,国内外学者对其进行了改进:Bekdaş 等<sup>[5]</sup>将花朵授粉算法应用于桁架结构设计优化。崔丽群等<sup>[6]</sup>采用和声算法的最优解作为 FPA 的初始解,并通过折射原理丰富算法种群的多样性,提升了算法的寻优精度;肖辉辉等<sup>[7]</sup>在花朵授粉算法中融入复合形法的思想,利用该思想将种群中最差的个体通过种群中心进行反射,使其变为较好的解,有效改善了解的质量。Wang 等<sup>[8]</sup>为消除维数对算法性能的影响,从各个维度对解改进,并引入局部领域搜索策略,提升了基本 FPA 的寻优速度和搜索精度;肖辉辉等<sup>[9]</sup>提出具有入侵杂草策略的花朵授粉算法,利用入侵杂草算法中的繁殖操作动态增加种群的多样性,最后利用改进的竞争生成操作丰富了种群和多样性,能够在一定程度上避免算法陷入局部极值;肖辉辉等<sup>[10]</sup>利用量子行为的相干、叠加和并行等特征丰富了花朵个体的形态,扩大算法的搜索广度,有效避免了算法陷入局部最优。这些方法在一定程度上提高了算法的寻优性能,但是在算法稳定性、搜索精度和收敛速度等方面仍有待改进。

为提高 FPA 算法的全局寻优能力,引入小生境技术,并结合混沌搜索策略对算法进行改进:首先,小生境技术使各子种群互相排斥,追逐各自搜索范围内的极值点,动态形成独立的搜索空间,避免协同中的种群过度汇聚,从而确保了种群的多样性;其次,采用混沌搜索策略对算法的局部寻优能力进行优化,利用混沌的无序性和遍历性等特点提升算法的搜索精度。实验表明,应用小生境混沌搜索策略的花朵授粉算法在全局收敛能力和寻优精度上都有所改善,证明了文中改进措施的有效性。

## 1 花朵授粉算法

自然界中,显花植物的授粉方式可分为自花授粉和异花授粉。异花授粉一般借助传播者(鸟、蜜蜂)等完成,这些携带着植物花粉的传播者能够飞行很远,因此,异花授粉能够克服花朵之间距离的限制,FPA 算法将这种授粉方式称为全局寻优。自花授粉是模拟植物自交的过程,FPA 算法将这种授粉方式称为局部寻优。自花授粉和异花授粉之间的转换通过转换概率  $p$  调节。为简化问题,FPA 算法假设每株植物仅开一朵花,每朵花只产生一个花粉配子,对应一个解。根据显花植物传粉过程,FPA 还需假定具有以下规则:

规则 1:异花授粉需借助传播者(鸟、蜜蜂)等完成,飞行方式为莱维飞行;

规则 2:自花授粉指借助风等非生物条件的授粉方式,称之为局部寻优;

规则 3:花的常性即繁衍概率,与涉及的两朵花的相似性成比例;

规则 4:转换概率  $p \in [0, 1]$ ,由于风等传播者或其他自然因素的影响,植物自花授粉和异花授粉之间发生的概率存在不确定性,在算法中起重要作用。文献[11]中经过大量的实验发现, $p$  值取 0.8 时,算法的寻优效果最好。

## 2 应用小生境混沌搜索策略的花朵授粉算法

### 2.1 小生境技术

小生境是生物学概念,指物种生存所必要的生存环境。生物进化过程中,总是与自己相同的物种生活中一起,共同繁衍后代,这就是小生境自然现象<sup>[12]</sup>。不同小生境种群根据自身所处的环境自主选择不同的进化方向,种群间和内部存在相互竞争,有效保持了物种的多样性<sup>[13]</sup>。

文中将小生境技术引入 FPA 算法,其基本思想是根据某种规则对种群进行初始化分类,划分出若干子种群,各个子种群在相对独立的搜索域对解空间内的局部最优点展开同步搜索,避免过早收敛。该技术的关键是对小生境的划分,文中的方法依据个体适应度来实现:1)定义所有个体的适应度值;2)根据适应度值大小对每个种群的个体进行排序,选择适应度最高的个体  $A = (a_1, a_2, a_3, \dots, a_k)$  为小生境的中心,并计算其他个体与中心的欧式距离,计算公式为

$$\|x_i - x_y\| = \sqrt{\sum_{k=1}^w (x_{ik} - x_{jk})^2}, \quad (1)$$

式中:  $k \in W$ ;  $i = 1, 2, \dots, W-1$ ;  $j = i+1, \dots, W$ 。

为减少计算量,提升迭代后期的求解精度,半径  $R$  需随着迭代次数逐渐增大,小生境数则应随之减少。文中半径的计算方法为

$$R = \max(d_{is}) \left[ \left( \frac{i_{t,\max} - i_t}{i_{t,\max}} \right) u_n \times (u_{\max} - u_{\min}) + u_{\min} \right] - 1, \quad (2)$$

$$\tilde{R} = \left( \frac{i_{t,\max} - i_t}{i_{t,\max}} \right) u_n \times (u_{\max} - u_{\min}) + u_{\min}, \quad (3)$$

$$R = \frac{\max(d_{is})}{\tilde{R}}, \quad (4)$$

式中:  $i_t$  为迭代次数;  $i_{t,\max}$  为最大迭代次数;  $u_{\max}$ 、 $u_{\min}$  与  $u_n$  表示算法的可调参数,用于调节算法迭代中分类的数量;  $d_{is}$  为每个个体距中心的欧式距离;  $\tilde{R}$  为中间变量。

在算法运行过程中,通过小生境的动态聚类,各个子种群在相对独立的空间寻优,避免陷入局部极值,增强了算法的全局性。

## 2.2 混沌优化策略

混沌是非线性系统中特有的一种周期运动现象,运动行为介于规则和随机之间<sup>[14]</sup>。由于混沌运动具有无序性和初值敏感性等特点,作为局部搜索算法时,能够很好地提升算法的搜索效率<sup>[15]</sup>。相比于常用的 Logistic 映射,逻辑自映射函数产生的混沌序列遍历性更优,因此,文中利用其实现混沌优化,数学表达式

$$y_{(n+1),d} = 1 - 2 \times y_{n,d}^2, \quad n = 0, 1, \dots, N, y_{n,d} \in (-1, 1). \quad (5)$$

只要方程式中的迭代初值不为 0,混沌就会发生,混沌映射的定义域为  $(-1, 1)$ ,且不为 0,  $d$  表示搜索空间的维数。在搜索过程中的某一时刻,花朵  $i$  个体的结果位于  $D$  维空间中的位置为  $x_i$ 。

**Step 1:** 将花朵个体对应的解在空间位置的每一维映射至  $[-1, 1]$  上,为

$$L_{id} = \frac{2(y_{id} - a_{id})}{(b_{id} - a_{id})} - 1, \quad d = 1, 2, \dots, D, \quad (6)$$

式中,  $a_{id}$  和  $b_{id}$  分别表示花朵  $i$  第  $d$  维变量的搜索上界和下界。

**Step 2:** 根据式(5)的载波操作,将混沌变量加载至待搜索的个体,从而得到混沌算子操作后的新个体。

**Step 3:** 将获得的新混沌变量序列按照式(7)变换到原解空间中进行性能评价。

$$x'_{id} = 0.5 \times (b_{id} - a_{id}) \times L_{id} + 0.5 \times (b_{id} + a_{id}), \quad d = 1, 2, \dots, D. \quad (7)$$

**Step 4:** 若搜索到更优的解,将这个更优解的位置代替花朵  $i$  解的原先位置;否则,继续下一轮混沌搜索直到预先设定的搜索次数。

为进一步提高搜索效率,避免算法过早陷入局部最优,在算法的初期增大搜索范围;在算法搜索后期的减少搜索范围,以加快收敛速度。采用式(8)动态收缩搜索区域。

$$\begin{aligned} x_{\min,j} &= \max\{x_{\min,j}, x_{g,j} - \rho \times (x_{\max,j} - x_{\min,j})\}, \\ x_{\max,j} &= \max\{x_{\max,j}, x_{g,j} + \rho \times (x_{\max,j} - x_{\min,j})\}, \end{aligned} \quad (8)$$

式中:  $[x_{\min,j}, x_{\max,j}]$  表示第  $j$  维变量的搜索范围;  $x_{g,j}$  表示最优花朵个体的第  $j$  维;式(9)为收缩因子  $\rho$  的计算方式;  $t$  为算法当前迭代次数。

$$\rho(t) = 1 - \frac{1}{1 - \exp(-0.04t + 4)}. \quad (9)$$

同时为在运算速度和求解精度之间取得平衡,不需要对全部花朵进行混沌操作,选取部分优秀个体进行混沌优化。

## 2.3 算法流程

具有小生境混沌搜索策略的花朵授粉算法实现过程如下:

**步骤 1** 初始化 NCFPA 各参数:花朵种群数  $n$ ,种群规模  $m$ ,转换概率  $p$ ,最大迭代次数  $N_{\text{iter}}$ ,精英群体比例  $L$ ,混沌搜索迭代次数  $K$ ,最大搜索次数或搜索精度  $\epsilon$ 。

**步骤 2** 用小生境技术对种群进行分类,使用式(2)~式(4)选择小生境子种群的半径  $R$ ,计算所有花朵

种群中个体的适应度值,将欧氏距离在  $R$  范围内的进行标记,记录每个小生境中的最优花朵及其对应的最优解。

**步骤 3** 若转换概率  $p > \text{rand}$ ,按式(10)更新步骤 2 中的花朵个体对应的解,并进行越界处理。

$$X_i^{t+1} = X_i^t + L(x_{\text{best}} - X_i^t), \tag{10}$$

式中: $X_i^{t+1}$ 、 $X_i^t$  分别是第  $t+1$  代、第  $t$  代花朵个体对应的解; $x_{\text{best}}$  是全局最优解; $L$  是步长, $L$  的计算公式为

$$L \sim \frac{\lambda \Gamma(\lambda) \sin(\pi\lambda/2)}{\pi} \frac{1}{s^{1+\lambda}}, (s \gg s_0 > 0). \tag{11}$$

其中: $\lambda = 1.5$ , $\Gamma(\lambda)$  是标准伽马函数。

**步骤 4** 若转换概率  $p < \text{rand}$ ,按式(12)更新解,并进行越界处理。

$$X_i^{t+1} = X_i^t + \epsilon (X_j^t - X_k^t), \tag{12}$$

式中: $\epsilon$  是  $[0, 1]$  上服从均匀分布的随机数; $X_j^t$ 、 $X_k^t$  是相同植物上不同花朵的花粉。

**步骤 5** 计算步骤 3 和步骤 4 的花朵个体的适应度值,并记录每个小生境种群中的最劣个体和最优个体。

**步骤 6** 对各小生境的花朵种群进行评估:选取前  $n\%$  适应度高的个体作为精英,采取 2.1 节混沌优化策略对精英个体进行优化;选取后  $n\%$  适应度低的最差个体,重新初始化产生新的花朵个体对其替换。

**步骤 7** 若迭代次数到达设定值,则初始化更新最劣的子种群。

**步骤 8** 若算法满足终止条件,则输出最优解,否则转至步骤 3。

### 3 仿真实验与分析

为验证文中提出的应用小生境混沌搜索策略的花朵授粉算法的性能,选取 8 个标准测试函数进行仿真测试,并与 FPA、BA 及 DE 算法进行求解对比。

#### 3.1 标准测试函数

8 个测试函数具体为 Branin ( $f_1$ )、Schaffer F6 ( $f_2$ )、Sphere ( $f_3$ )、Schwefel ( $f_4$ )、Rastrigin ( $f_5$ )、Griewank ( $f_6$ )、Ackley ( $f_7$ )、Rosenrock ( $f_8$ )。其中函数  $f_1 \sim f_7$  均为非线性多峰函数,搜索空间中存在多个局部极值点,能够有效检验算法的全局搜索性能、群体多样性和跳出局部极值的能力;函数  $f_8$  是单峰函数,是测试算法全局收敛性能的经典函数。

#### 3.2 算法参数设置

实验参数设置为:基本 FPA 和 NCFPA 的转换概率  $p = 0.8$ 。BA 算法参数:脉冲音强  $A = 0.25$ ,脉冲率  $r = 0.5$ ,最低频率  $Q = 0$ ,最高频率  $Q = 2$ 。混沌搜索最大迭代次数: $K = 40$ ,精英群体比例为  $15\%$ ;DE 的参数设置:交叉概率  $CR = 0.9$ ,缩放因子  $F = 0.8$ ,变异概率  $G = 0.8$ 。所有算法的最大迭代次数均为  $N\_iter = 300$ ,根据求解函数的维数,种群规模分别取  $m = 20, 30, 40, 60$ ,所有算法独立运行 30 次。

#### 3.3 算法寻优性能仿真效果及分析

表 1 为 4 种算法在不同维度标准测试函数下的测试结果,图 1~图 8 分别为相应的寻优效果图。

表 1 标准函数测试结果  
Table 1 Standard function test results

函数	维数	种群数	算法	最优值	优化均值	标准方差	寻优成功率/%
$f_1$	2	20	BA	0.397 9	0.676 9	0.162 3	39
			DE	0.397 9	0.478 3	0.023 1	60
			FPA	0.397 9	0.429 6	0.008 9	83.3
			NCFPA	0.397 9	0.397 9	0	100

续表

函数	维数	种群数	算法	最优值	优化均值	标准方差	寻优成功率/%
$\int_2$	2	20	BA	-1	-0.990 3	0.040 1	22
			DE	-1	-0.994 1	5.11E-04	87
			FPA	-1	-1	0	100
			NCFPA	-1	-1	0	100
$\int_3$	10	30	BA	1.646 2	3.409 2	1.418 5	0
			DE	0.403 0	2.032 4	1.309 6	0
			FPA	3.30E-13	3.736 5	10.121 0	27.5
			NCFPA	0	0	0	100
$\int_4$	10	30	BA	59.697 3	65.332 1	8.035 4	0
			DE	2.766 7	20.132 2	36.860 5	0
			FPA	7.40E-10	0.016 0	77.553 7	21.5
			NCFPA	0	0	0	100
$\int_5$	20	40	BA	35.152 8	126.997 3	71.096 7	0
			DE	22.505 1	28.845 2	5.519 1	0
			FPA	1.964 6	4.487 7	2.198 9	0
			NCFPA	7.40E-05	0.002 79	0.004 8	71.5
$\int_6$	20	40	BA	1.875 24	12.118 3	21.208 3	0
			DE	0.263 1	7.062 8	11.834 9	0
			FPA	0.032 3	13.935 5	33.354 7	0
			NCFPA	0	3.403 0	0.600 4	54
$\int_7$	30	60	BA	276.977 3	376.665 0	136.780 7	0
			DE	118.438 5	316.052 8	96.356 1	0
			FPA	155.603 2	222.611 5	65.764 1	0
			NCFPA	0	26.679 5	6.376 3E-01	28.5
$\int_8$	30	60	BA	54.075 5	267.100 5	37 707E+03	0
			DE	19.077 5	158.965 0	5.755 9E+02	0
			FPA	18.411 8	76.208 3	1.051 5E+02	0
			NCFPA	0	15.654 5	2.970 0E-01	46.5

文中设定当求得的最优值和理论最优值间的相对误差低于 1% 时, 算法找到最优解。结果表明: 由于函数  $f_1$ 、 $f_2$  的维度较低, 4 种对比算法均能找到最优解, 但 NCFPA 的寻优成功率最高、寻优速度更快、稳定性更优, 寻优曲线如图 1 和图 2 所示。对于函数  $f_3 \sim f_7$ , 由于函数形态均为高维、多峰, 搜索空间中存在大量局部极值, BA 和 DE 均未找到最优解, FPA 仅在  $f_3$  ( $D=10$ )、 $f_4$  ( $D=10$ ) 2 种情况下找到最优解; NCFPA 不仅都能找到最优解, 并且收敛的速度更快, 寻优精度更高, 寻优曲线如图 3~图 7 所示。由于函数  $f_8$  ( $D=30$ ) 只为算法提供少量信息, BA、DE 和 FPA 均陷入局部极值中, 并且寻优精度较差。NCFPA 不仅能够迅速地找到全局最优解, 并且寻优成功率较高, 寻优曲线如图 6 所示。



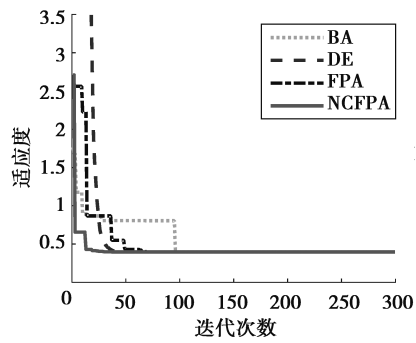


图 1 函数  $f_1$  寻优曲线( $D=2$ )  
Fig. 1  $f_1$  optimization curve( $D=2$ )

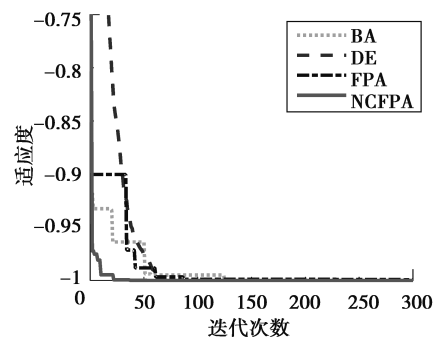


图 2 函数  $f_2$  寻优曲线( $D=2$ )  
Fi. 2  $f_2$  optimization curve( $D=2$ )

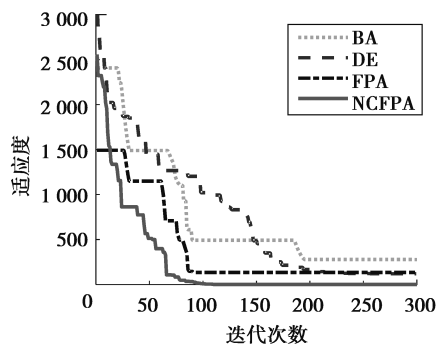


图 3 函数  $f_3$  寻优曲线( $D=10$ )  
Fig. 3  $f_3$  optimization curve( $D=10$ )

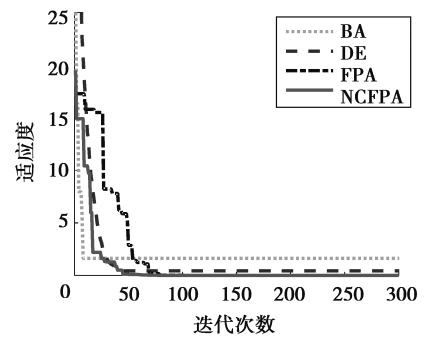


图 4 函数  $f_4$  寻优曲线( $D=10$ )  
Fig. 4  $f_4$  optimization curve( $D=10$ )

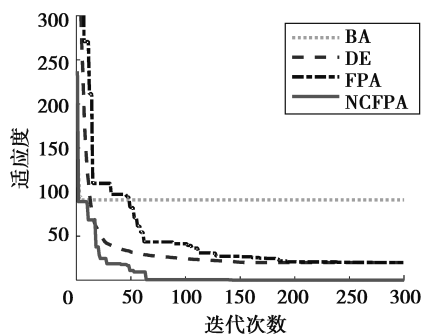


图 5 函数  $f_5$  寻优曲线( $D=20$ )  
Fig. 5  $f_5$  optimization curve( $D=20$ )

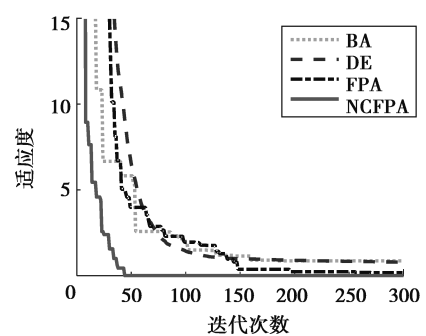


图 6 函数  $f_6$  寻优曲线( $D=20$ )  
Fig. 6  $f_6$  optimization curve( $D=20$ )

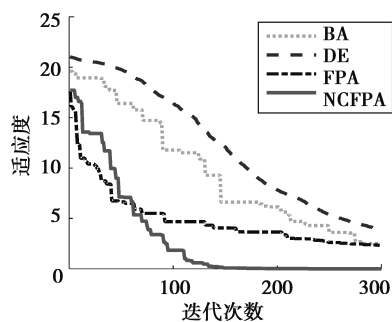


图 7 函数  $f_7$  寻优曲线( $D=30$ )  
Fig. 7  $f_7$  optimization curve( $D=30$ )

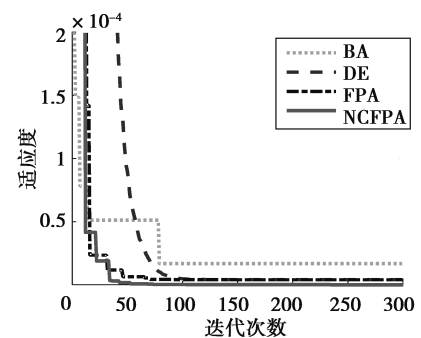


图 8 函数  $f_8$  寻优曲线( $D=30$ )  
Fig. 8  $f_8$  optimization curve( $D=30$ )

仿真结果表明:DE、BA 和 FPA 3 种算法在低维函数中的寻优性能总体近似;在高维复杂空间中,FPA 寻优效果尽管略比 DE 和 BA 更优,但是 3 种算法都表现出易陷入局部最优、搜索精度不高和稳定性差等特点;在相同搜索条件下,NCFPA 的寻优精度和鲁棒性均远优于 DE、BA 和基本 FPA,表现出较好的全局寻优能力。

### 3.4 算法时间复杂度分析

文中将 NCFPA 与 FPA 算法在相同运行环境下,通过运行时间来观察文中改进算法是否满足时间复杂度较低的要求。试验采用  $f_5$ 、 $f_6$ 、 $f_7$  和  $f_8$  4 个函数来测试并分析文中算法的时间复杂度,设置最大迭代次数  $N\_iter=300$ ,种群数  $m=60$ 。2 种算法均独立运行 20 次,计算在不同维数(10 维,30 维和 100 维)下的平均运行时间和最小运行时间,结果如表 2 所示。

表 2 NCFPA 算法的时间复杂度  
Table 2 Time complexity analysis of NCFPA

函数	维数	算法	最优值	平均值
$f_5$	10	FPA	0.127	0.135 67
		NCFPA	0.623	0.654 16
	30	FPA	0.224	0.236 49
		NCFPA	1.036	1.064 31
	100	FPA	0.705	0.726 91
		NCFPA	2.353	2.610 03
$f_6$	10	FPA	0.143	0.153 67
		NCFPA	0.624	0.670 13
	30	FPA	0.259	0.267 43
		NCFPA	1.051	1.100 34
	100	FPA	0.705	0.726 91
		NCFPA	2.393	2.520 14
$f_7$	10	FPA	0.110	0.129 50
		NCFPA	0.612	0.644 32
	30	FPA	0.261	0.272 66
		NCFPA	1.002	1.053 2
	100	FPA	0.658	0.691 84
		NCFPA	2.148	2.446 01
$f_8$	10	FPA	0.121	0.128 74
		NCFPA	0.672	0.693 01
	30	FPA	0.212	0.222 90
		NCFPA	1.036	1.103 26
	100	FPA	0.549	0.560 22
		NCFPA	1.862	2.031 65

表 2 中,NCFPA 比 FPA 算法的平均运行时间略长一点,但 NCFPA 算法的平均运行时间和最小运行时间仍较短,可以看出 NCFPA 算法的时间复杂度较低。NCFPA 算法在不同维度环境下的寻优精度都远高于 FPA 算法,表明改进算法是可行和有效的。

## 4 结 论

将小生境技术和混沌优化策略应用花朵授粉算法中,利用小生境技术掌控花朵群体的全局搜索方向,竞

争机制有助于维持种群的多样性,淘汰策略则能够不断更新子种群,两种策略的结合,成功地避免了算法的早熟收敛,提升了全局寻优能力。引入逻辑自映射混沌序列,丰富了花朵个体的性状,选取精英个体进行混沌优化,显著提高了算法的搜索精度,减少了算法迭代次数。仿真实验表明,NCFPA 算法能够有效克服传统花朵授粉算法存在的全局收敛能力不足、寻优精度低和易早熟的缺陷。

#### 参考文献:

- [ 1 ] Yang X S. Flower pollination algorithm for global optimization[M]. Berlin, Heidelberg:Springer, 2012:240-249.
- [ 2 ] Yang X S, Karamanoglu M, He X. Multi-objective flower algorithm for optimization[J]. Procedia Computer Science, 2013, 18(1):861-868.
- [ 3 ] Rodrigues D, Yang X S, Souza A N D, et al. Binary flower pollination algorithm and its application to feature selection [M]// Recent Advances in Swarm Intelligence and Evolutionary Computation. Berlin: Springer International Publishing, 2015:85-100.
- [ 4 ] 王生生, 杜鹏, 董如意, 等. 改进的花朵授粉算法在微网优化调度中的应用[J]. 吉林大学学报(自然科学版), 2018, 39(3): 334-338.  
Wang S S, Peng D U, Dong R Y, et al. Modified Flower Pollination Algorithm and Applications on Optimization Dispatch of Microgrid[J]. Journal of Northeastern University(Natural Science) , 2018, 39(3): 334-338. (in Chinese)
- [ 5 ] Bekdaş G, Nigdeli S M, Yang X S. Sizing optimization of truss structures using flower pollination algorithm[J]. Applied Soft Computing, 2015, 37(C):322-331.
- [ 6 ] 崔丽群, 张晨, 郑宝林, 等. 基于折射原理的混合型花朵授粉算法[J/OL]. 计算机应用研究, 2019, 36(5).[2018-06-20].<http://www.aocmag.com/article/02-2019-05-003.html>. (in Chinese)  
CUI Liqun, ZHANG Chen, ZHENG Baolin, et al. Hybrid flower pollination algorithm based on refraction principle[J/OL]. Application Research of Computers, 2019, 36(5).[2018-06-20].<http://www.aocmag.com/article/02-2019-05-003.html>.
- [ 7 ] 肖辉辉. 基于单纯形法和自适应步长的花朵授粉算法[J]. 计算机工程与科学, 2016, 38(10):2126-2133.  
XIAO Huihui, Flower pollination algorithm based on simplex method and adaptive step size[J]. Computer Engineering and Science, 2016, 38(10):2126-2133. (in Chinese)
- [ 8 ] Wang R, Zhou Y. Flower pollination algorithm with dimension by dimension improvement[J]. Mathematical Problems in Engineering, 2014, 2014(4):1-9.
- [ 9 ] 肖辉辉, 段艳明. 具有入侵杂草策略的花朵授粉算法[J]. 系统仿真学报, 2017, 29(2):264-272.  
XIAO Huihui, DUAN Yanming. Flower pollination algorithm with invasive weed strategy [J]. Journal of System Simulation, 2017, 29(2):264-272. (in Chinese)
- [10] 肖辉辉, 万常选, 段艳明, 等. 基于引力搜索机制的花朵授粉算法[J]. 自动化学报, 2017, 43(4):576-594.  
XIAO Huihui, WAN Changxuan, DUAN Yanming, et al. Flower pollination algorithm based on gravity search mechanism[J]. Acta Automatica Sinica, 2017, 43(4):576-594. (in Chinese)
- [11] 戴娇, 张明新, 孙昊, 等. 花朵授粉算法的优化[J]. 计算机工程与设计, 2017, 38(6):1503-1509.  
DAI Jiao, ZHANG Mingxin, SUN Hao, et al. Optimization of flower pollination algorithm[J]. Computer Engineering and Design, 2017, 38(6):1503-1509. (in Chinese)
- [12] 赵小强, 何智娥. 基于小生境混沌遗传算法的水资源优化调度[J]. 北京工业大学学报, 2015(9):1334-1340.  
ZHAO Xiaoqiang, HE Zhie. Chaos genetic algorithm based on niche of water resources optimal scheduling research[J]. Journal of Beijing University of Technology , 2015(9):1334-1340. (in Chinese)
- [13] 李锦. 小生境混合蛙跳算法研究与应用[D]. 西安:西安电子科技大学, 2012.  
LI Jin. Research and application of niche shuffled frog leaping algorithm[D]. Xi'an: Xidian University, 2012. (in Chinese)
- [14] 刘长平, 叶春明. 基于逻辑自映射的变尺度混沌粒子群优化算法[J]. 计算机应用研究, 2011, 28(8):2825-2827.  
LIU Changping, YE Chunming. Mutative scale chaos particle swarm optimization algorithm based on self logical mapping function[J]. Application Research of Computers, 2011, 28(8):2825-2827. (in Chinese)
- [15] 刘长平, 叶春明. 具有混沌搜索策略的蝙蝠优化算法及性能仿真[J]. 系统仿真学报, 2013, 25(6):1183-1188.  
LIU Changping, YE Chunming. Bat algorithm with chaotic search strategy and its performance[J]. Journal of system Simulation, 2013, 25(6):1183-1188. (in Chinese)