

6-1-2020

Flower Pollination Algorithm with Invasive Weed Optimization

Huihui Xiao

1. College of Computer and Information Engineering, Hechi University, Yizhou 546300, China; ;2. School of Information and Technology, Jiangxi University of Finance and Economics, Nanchang 330013, China;

Yanming Duan

1. College of Computer and Information Engineering, Hechi University, Yizhou 546300, China; ;

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Flower Pollination Algorithm with Invasive Weed Optimization

Abstract

Abstract: In order to overcome the problems of easily relapsing into local extremum and low speed of convergence, a flower pollination algorithm with invaded weeds strategy was proposed. *The algorithm generate dynamically populations, and creased the diversity and effectiveness of population by breeding, spatial diffusion and competition of invasive weeds. It could effectively avoid local optimum, enhance the capacity of global optimization, and improve the convergence speed.* The comparison and analysis results of the 8 CEC2005 benchmark functions, the simulation results show that the proposed algorithm has the advantages of better global searching ability, faster convergence and more precise convergence than those of the basic flower pollination algorithm, differential evolution algorithm and bat algorithm.

Keywords

Flower Pollination Algorithm, optimization performance, Invasive Weed Optimization Algorithm, fitness

Recommended Citation

Xiao Huihui, Duan Yanming. Flower Pollination Algorithm with Invasive Weed Optimization[J]. Journal of System Simulation, 2017, 29(2): 264-272.

具有入侵杂草策略的花朵授粉算法

肖辉辉^{1,2}, 段艳明¹

(1. 河池学院计算机与信息工程学院, 广西 宜州 546300; 2. 江西财经大学信息管理学院, 江西 南昌 330013)

摘要: 针对花朵授粉算法易陷入局部极值、收敛速度慢的不足, 提出一种具有入侵杂草策略的花朵授粉算法。该算法通过入侵杂草的繁殖、空间扩散和竞争策略, 动态生成种群, 增加种群的多样性和有效性, 使算法能有效地避免陷入局部最优, 增强全局寻优能力, 提高收敛速度。通过 8 个 CEC2005 benchmark 测试函数进行测试比较, 仿真结果表明, 改进算法的全局寻优能力明显优于基本的花朵授粉算法、差分进化算法和蝙蝠算法, 其收敛精度、收敛速度、鲁棒性均较对比算法有较大提高。

关键词: 花朵授粉算法; 寻优性能; 入侵杂草算法; 适应度值

中图分类号: TP301.6 文献标识码: A 文章编号: 1004-731X (2017) 02-0264-09

DOI: 10.16182/j.issn1004731x.joss.201702005

Flower Pollination Algorithm with Invasive Weed Optimization

Xiao Huihui^{1,2}, Duan Yanming¹

(1. College of Computer and Information Engineering, Hechi University, Yizhou 546300, China;

2. School of Information and Technology, Jiangxi University of Finance and Economics, Nanchang 330013, China)

Abstract: In order to overcome the problems of easily relapsing into local extremum and low speed of convergence, a flower pollination algorithm with invaded weeds strategy was proposed. The algorithm generate dynamically populations, and creased the diversity and effectiveness of population by breeding, spatial diffusion and competition of invasive weeds. It could effectively avoid local optimum, enhance the capacity of global optimization, and improve the convergence speed. The comparison and analysis results of the 8 CEC2005 benchmark functions, the simulation results show that the proposed algorithm has the advantages of better global searching ability, faster convergence and more precise convergence than those of the basic flower pollination algorithm, differential evolution algorithm and bat algorithm.

Keywords: Flower Pollination Algorithm; optimization performance; Invasive Weed Optimization Algorithm; fitness

引言

2012 年, 受自然界中显花植物花朵授粉过程的启发, 英国剑桥大学学者 Xin-she Yang 提出一种新型元启发式群智能优化算法——花朵授粉算法

(Flower Pollination Algorithm, FPA)^[1], FPA 算法易于各种语言编码, 实现简单, 并采用了莱维飞行机制使其寻优效果较好。FPA 算法正用于不少领域: Xin-she Yang 于 2013 年用其来解决多目标优化问题^[2], 并取得了较好结果; El-henawy I 等人于 2014 年用其解决大整数规划问题^[3]; Marwa Sharawi 等人于 2014 年用其优化无线传感器网络生命周期问题^[4]; Osama Abdel-Raouf 等人于 2014 年用其解决数独谜题^[5]; R. Prathiba 等人于 2014 年用其优化电



收稿日期: 2015-05-06 修回日期: 2015-06-18;
基金项目: 国家自然科学基金(61165015, 61562032),
河池学院科研项目(XJ2015QN003);
作者简介: 肖辉辉(1977-), 男, 江西永新, 博士生,
副教授, 研究方向为智能计算、情感计算; 段艳明
(通讯作者 1978-), 女, 江西永新, 硕士, 副教授,
研究方向为智能计算。

<http://www.china-simulation.com>

力调度问题^[6]。但因其提出时间短, 目前国内外对该算法的研究还处于初级阶段, 算法理论尚未成熟, 研究成果也较少, 因此对 FPA 的研究迫切而有价值。然而, 该算法与粒子群算法类似, 也存在易陷入局部最优且进化后期收敛速度慢等不足, 尤其对于多局部极值、高维的较复杂优化问题。

针对 FPA 算法的不足, 众多国内外学者针对基本花朵授粉算法存在的缺陷进行改进: 如 Wang 等人^[7]考虑到维数对算法的收敛速度和收敛精度的影响, 提出对个体进行逐维改进和引入局部领域搜索策略思想来对算法进行改进; K.Lenin 等人^[8]利用混沌策略改进和声算法增强其种群的多样性, 再把和声算法的最优解作为花朵授粉算法的初始解。上述这些改进在一定程度上避免了局部极值, 提高了算法的寻优能力, 仍未使算法完全避免陷入局部极值, 其收敛精度、稳定性、收敛速度等方面仍有待改进。

本文提出了具有入侵杂草策略的花朵授粉算法(IWOFPA), 该算法利用入侵杂草优化算法中的繁殖操作动态增加种群个数, 再通过空间扩散操作增强种群的多样性, 最后利用改进的竞争生成操作增强种群的有效性和多样性, 从而使算法很大程度上避免陷入局部极值, 提高算法的性能。通过 8 个 CEC2005 benchmark 测试函数的仿真实验结果, 验证了改进算法的有效性和优越性, 改进算法能够有效地避免早熟收敛, 其寻优精度、寻优速度、鲁棒性均明显优于基本的花朵授粉算法。

1 花朵授粉算法

据统计, 目前在自然界中被人类发现的植物大约有 37 万种, 显花植物大约占有 20 万种, 而其中 80% 的植物依靠生物授粉来繁衍后代^[9]。根据显花植物的授粉对象不同, 可分为异花授粉及自花授粉两种。异花授粉一般需要传播者, 自然界的传播者鸟、蜜蜂等能飞行很长的距离, 且其具有莱维飞行的行为, 跳跃或飞行的步长服从莱维飞行分布, 故异花授粉可以发生在距离较远的随机地方, 在 FPA

算法中模拟这种授粉方式为全局寻优(授粉)。另外, 自花授粉是植物成熟的花粉粒传播到自身的花朵上, 在 FPA 算法中模拟这种授粉方式为局部寻优(授粉)。同时, FPA 算法中, 假设每颗显花植物仅只开一朵花, 且每朵花仅产生一个花粉配子, 一朵花或一个配子对应于优化问题中的一个解。文献[4]描述了标准的花朵授粉算法的实现步骤。

花朵授粉算法模拟自然界中显花植物花朵传粉的过程, 需假设其理想条件如下^[1]:

- (1) 生物异花授粉是指携带花粉的传播者(鸟、蜜蜂等)通过莱维飞行进行的全局授粉过程;
- (2) 非生物自花授粉是指植物花朵的自身局部授粉过程;
- (3) 花的常性可以被认为是繁衍概率, 繁衍概率与参与的两朵花的相似性成比例关系;
- (4) 全局授粉和局部授粉之间的转换由转换概率 $p \in [0, 1]$ 控制, 受物理上的邻近性和风等其他因素的影响, 在整个授粉活动中局部授粉是一个非常重要的部分。

2 入侵杂草策略的花朵授粉算法 (IWOFPA)

2.1 入侵杂草优化算法

入侵杂草优化算法 (Invasive Weed Optimization, IWO)^[10]是 Mehrabian 和 Lucas 于 2006 年提出的一种模拟自然界杂草入侵性繁殖的智能优化算法。在该算法中, 求解问题的可行解用杂草来表示, 杂草产生种子, 种子又通过空间扩散发育成杂草, 当种群(杂草)的数量达到设定的最大种群时, 杂草采用“适者生存”的自然规律进行竞争, 保留好的杂草, 淘汰差的杂草。入侵杂草优化算法兼顾了杂草繁殖、空间扩散和竞争的特征, 也正是这些特征使种群具有有效性和多样性:

- (1) 繁殖。根据杂草的适应度值产生种子, 好的(适应度高)杂草产生较多种子, 反之, 产生较少种子。杂草产生种子的数目为

$$\text{seed}_{\text{num}} = \frac{f_i - f_{\min}}{f_{\max} - f_{\min}} (\text{seed}_{\max} - \text{seed}_{\min}) + \text{seed}_{\min} \quad (1)$$

其中: f_i 为第 i 个杂草的适应度值; f_{\max} 和 f_{\min} 分别表示当前杂草中的最大和最小适应度值; seed_{\max} 和 seed_{\min} 分别表示杂草所能产生的最大种子数和最小种子数。

(2) 空间扩散。为了增加杂草的多样性和寻优精度, 杂草生成的种子以标准差为 δ 的正态分布的形式分布在杂草的四周, 标准差 δ 的公式为

$$\delta_i = \delta_{\text{init}} + \left(\frac{t_{\max} - t_i}{t_{\max}}\right)^m (\delta_{\text{init}} - \delta_{\text{final}}) \quad (2)$$

其中: δ_i 为第 i 代的标准差; t_{\max} 为设定的最大迭代次数; t_i 为第 i 代迭代次数; δ_{init} 和 δ_{final} 分别为设定的初始标准差和最终标准差; m 为非线性调和因子。

(3) 竞争生成。杂草经过一定的繁殖扩散后种群(杂草和种子)数量会超过设定的最大种群数 sizepop_{\max} , 此时, 采用贪婪法按照适应度值的高低对杂草和种子进行排序, 选取适应度好的前 sizepop_{\max} 个个体, 其余个体进行淘汰掉。

2.2 IWOFPA 算法的实现

针对花朵授粉算法的不足, 把入侵杂草优化算法的繁殖、空间扩散和竞争的特征融入到花朵授粉算法中, 增强花朵的多样性和有效性, 以达到提高 FPA 算法的寻优能力。同时, 对入侵杂草优化算法的竞争生成部分进行改进, 考虑选取的 sizepop_{\max} 个体的有效性和多样性, 不再是对所有种群按适应度大小排序后取适应度好的前 sizepop_{\max} 个, 而是生成 sizepop_{\max} 个的父个体集 u , 从 u 中随机选取两个个体 x_1 和 x_2 , 再利用公式(3)交叉生成新的个体。

$$\begin{aligned} x_{\text{new}1} &= \beta^{0.5} \cdot x_1 + (1 - \beta^{0.5}) \cdot x_2 \\ x_{\text{new}2} &= \beta^{0.5} \cdot x_2 + (1 - \beta^{0.5}) \cdot x_1 \end{aligned} \quad (3)$$

其中: β 是 $\in(0,1)$ 的随机数。生成 sizepop_{\max} 个的父个体集 u 的方法如下:

(1) 生成随机数 n_1 ($n_1 \in [1, \text{sizepop}_{\max}]$), 将当前种群随机分成 n_1 个组, 选取每个组的最优个体组成 u_1 ;

(2) 在初始边界 $[X_{\max}, X_{\min}]$ 随机生成 $n_1(n_2 \leftarrow \text{sizepop}_{\max} - n_1)$ 个个体, 组成 u_2 。

(3) $u \leftarrow u_1 \cup u_2$ 。

这样, 若个体 x_1 、 x_2 的都来自 u_1 , 则新个体 x_{new} 侧重有效性; 若个体 x_1 、 x_2 的都来自 u_2 , 则新个体 x_{new} 侧重多样性; 若个体 x_1 、 x_2 的分别来自 u_1 , u_2 , 则新个体 x_{new} 可能处于还未搜区域。

因此, 这样得到的新个体更大程度上改善了种群的多样性, 为种群带来了新信息, 能有效地提高算法的收敛速度和跳出局部极值。另外, 采用分组最优而不直接选择最优的 sizepop_{\max} 个个体, 有利于抑制早熟收敛。

竞争生成算子:

Step 1 $n_1 \leftarrow \text{random}[1, \text{sizepop}_{\max}]$, 将当前种群随机分成 n_1 个组, 选取每组中最优个体组成 u_1 ;

Step 2 $n_2 \leftarrow \text{sizepop}_{\max} - n_1$, 在初始边界 $[X_{\max}, X_{\min}]$ 内随机生成 n_2 个个体组成 u_2 ;

Step 3 $u \leftarrow u_1 \cup u_2$;

Step 4 $i \leftarrow 1$, 若 $i \leq \text{sizepop}_{\max}$, 则从 u 中随机选择两个个体按公式(3)生成两个新个体, $i \leftarrow i + 2$, 转 Step4;

Step 5 返回 sizepop_{\max} 个新个体。

因此, IWOFPA 算法的具体步骤如下:

Step 1 初始化参数: 花朵种群规模 sizepop_{\max} , 最大迭代次数 t_{\max} , 杂草所能产生的最大种子数 seed_{\max} 和最小种子数 seed_{\min} , 初始标准差 δ_{init} 和最终标准差 δ_{final} , 非线性调和因子 m ;

Step 2 计算当前种群的适应度值, 记录全局最优值(适应度最好值)及其对应的最优解;

Step 3 如果转换概率 $p > \text{rand}$ 条件成立, 按公式(4)对解进行更新, 并进行越界处理;

$$X_i^{t+1} = X_i^t + L(\text{best}x - X_i^t) \quad (4)$$

其中: X_i^{t+1} 、 X_i^t 分别是第 $t+1$ 代、第 t 代的解, $\text{best}x$ 是全局最优解, L 是步长, L 的计算用公式:

$$L \sim \frac{\lambda \Gamma(\lambda) \sin(\pi \lambda / 2)}{\pi} \frac{1}{s^{1+\lambda}} (s \gg s_0 > 0) \quad (5)$$

其中: $\lambda=3/2$; $\Gamma(\lambda)$ 是标准的伽马函数。

Step 4 倘若转换概率 $p < \text{rand}$ 条件成立, 按公式(6)对解进行更新, 并进行越界处理;

$$X_i^{t+1} = X_i^t + \epsilon (X_j^t - X_k^t) \quad (6)$$

其中: ϵ 是 $[0,1]$ 上服从均匀分布的随机数; X_j^t 、 X_k^t 是相同植物种类的不同花朵的花粉。

Step 5 计算 Step3 或者 Step4 的花朵个体的适应度值, 并记录最优值和最差值;

Step 6 利用 IWO 算法的公式(1)计算种群(花朵)产生种子的数目;

Step 7 利用 IWO 算法的公式(2)计算其标准差, 对种子进行空间扩散, 并进行越界处理;

Step 8 执行竞争生成算子;

Step 9 计算 sizepop_{\max} 个种群的最优适应度值及对应的最优解, 若较全局最优值更优, 则更新全局最优解和全局最优值;

Step 10 判断结束条件, 若满足, 退出程序并输出最优值及最优解, 否则, 转 Step3。

3 IWOFPFA 算法性能仿真实验

3.1 实验环境

本实验测试平台为: 操作系统为 Windows 7, CPU 为 Intel(R) Core(TM) i3-2120, 主频为 3.30 GHz, 内存为 2GB, 编程语言为 MATLAB R2012b。

3.2 测试函数及参数设置

为了验证本文算法的正确性和有效性, 通过对 8 个包括单峰、多峰、低维、高维的测试函数进行仿真来验证算法的改进效果, 并与 FPA、DE 及 BA 算法进行比较。测试函数选用 CEC2005 benchmark 测试函数中的一部分, 其中 $f_1 \sim f_3$ 为单峰函数, $f_4 \sim f_8$ 为多峰函数, 具体为: Shifted Sphere 函数(f_1)、Shifted Schwefel's Problem 1.2 函数(f_2)、Shifted Schwefel's Problem 1.2 with Noise in Fitness 函数(f_3)、Shifted Rosenbrock 函数(f_4)、Shifted Rotated Griewank's Function without Bounds 函数(f_5)、Shifted Rotated Ackley's with Global Optimum on the Bounds 函数(f_6)、Shifted Rastrigin's Function 函

数(f_7)、Expanded Rotated Extended Scffe's F6 函数(f_8)。

本文实验的各种参数设置: 种群个数 $n=15$, 最大迭代次数 $N_{\text{iter}}=300$ 。另外: FPA 算法的转换概率 $p=0.8$; IWOFPFA 算法参数: 转换概率 $p=0.8$, 非线性因子 $m=3$, 初始标准方差 $\text{std_ini}=2$, 最终标准方差 $\text{std_final}=2$, 最大种子数 $\text{seed_max}=20$, 最小种子数 $\text{seed_min}=2$, 最小种群数 $n1=12$, 最大种群数 $n2=20$; DE 算法参数: 杂交参数 $\text{CR}=0.9$, 变异率 $F=0.8$; BA 算法参数: 音量 $A=0.25$, 脉冲率 $r=0.5$, 最低频率 $Q_{\min}=0$, 最高频率 $Q_{\max}=2$ 。

3.3 实验结果及分析

本文的实验方法为: (1) 固定迭代次数, 测试 4 种算法的寻优性能; (2) IWOFPFA 与 FPA 在高维上的性能比较, 以验证 IWOFPFA 在求解高维函数的优势; (3) 与参考文献算法性能对比, 以进一步验证 IWOFPFA 算法的寻优精度。(4) IWOFPFA 算法的时间复杂度分析。

3.3.1 固定迭代次数的寻优性能分析

设置固定迭代次数为 300, 分别采用 IWOFPFA、FPA、DE 和 BA 算法对选用的 8 个测试函数进行求解, 为了防止算法的偶然性带来的误差, 对每个测试函数独立运行 20 次, 计算其最优值、最差值、优化均值和标准方差, 结果如表 1 所示, 其中, 寻优成功率=(找到最优解的次数)/总迭代次数。在实验中, 假定: $|\text{实际求解的最优值} - \text{理论最优值}| < 0.001$, 则认为寻优成功。对于每个测试函数, 其优化均值反映了在固定迭代次数下所达到的解的精度, 标准方差反映了算法的鲁棒性和稳定性。

从表 1 中可以看出, 对于测试函数 $f_1 \sim f_8$, 相比与 FPA、DE 和 BA 算法, 本文提出 IWOFPFA 算法最优值、最差值、优化均值和标准方差均要好。(1) 对于函数 $f_1 \sim f_3$ 和 $f_6 \sim f_8$, 在假定的寻优成功条件下, IWOFPFA 都能 100%地找到最优解。其中, 对于 $f_1 \sim f_3$, IWOFPFA 的性能最多比 FPA 提高 11 个数量

级。(2) 对于函数 f_4 、 f_5 和 f_7 ，虽然 IWOFFPA 没能找到最优解，但其性能均远比 FPA、DE 和 BA 要好。表 1 的实验结果说明无论从寻优精度还是收敛速度，IWOFFPA 算法都取得较大幅度的提高。

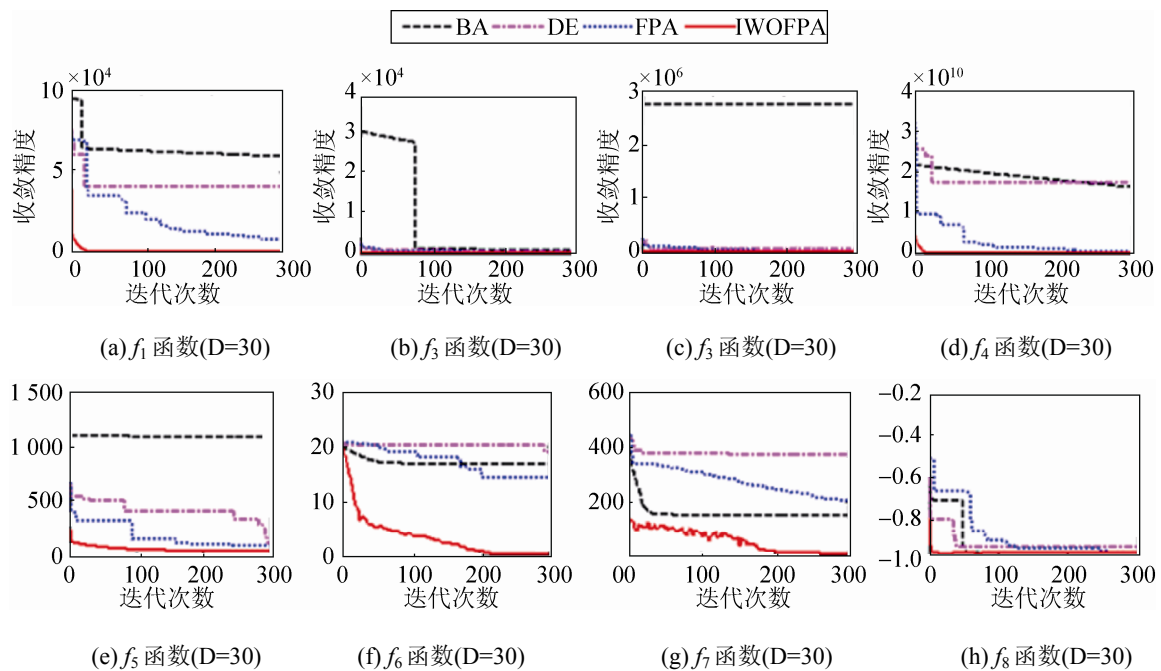
表 1 4 种算法在固定迭代次数下的寻优性能比较

Tab. 1 Optimization performance comparison of 4 kinds algorithm on fixed iterations

函数	维数	算法	最优值	优化均值	最差值	标准方差	寻优成功率/%
f_1	30	BA	2.972 2e+04	9.646 7e+04	1.966 6e+05	4.785 0e+04	0
		DE	6.760 2e+03	3.346 7e+04	5.193 8e+04	1.741 8e+04	0
		FPA	3.235 0e+03	6.009 0e+03	9.968 6e+03	1.637 0e+03	0
		IWOFFPA	2.546 6e-12	5.553 5e-12	9.655 0e-12	2.145 2e-12	100
f_2	30	BA	6.553 0e+04	1.270 3e+06	4.597 6e+06	3.597 2e+04	0
		DE	4.355 8e+04	5.955 4e+04	7.634 7e+04	1.708 4e+04	0
		FPA	2.462 6e+03	5.508 2e+03	9.574 2e+03	1.831 0e+03	0
		IWOFFPA	1.753 8e-06	2.017 2e-05	5.625 5e-05	7. 245 2e-07	100
f_3	30	BA	5.509 8e+05	3.180 9e+06	1.689 8e+07	2.475 7e+06	0
		DE	1.444 8e+04	6.408 9e+04	8.499 6e+04	1.631 0e+04	0
		FPA	4.345 6e+03	1.041 0e+04	1.046 3e+04	4.745 7e+03	0
		IWOFFPA	3.678 1e-07	1.218 6e-06	3.097 0e-06	8.373 9e-07	100
f_4	30	BA	9.112 9e+09	3.559 3e+10	8.562 0e+10	2.224 3e+10	0
		DE	1.106 4e+09	1.298 1e+10	1.960 4e+10	6.386 1e+09	0
		FPA	6.657 0e+07	3.576 6e+08	6.147 5e+08	1.548 6e+08	0
		IWOFFPA	7.950 3	140.249 0	3.213 0e+03	713.108 0	0
f_5	30	BA	303.468 0	896.543 0	1.365 1e+03	292.795 8	0
		DE	51.227 1	306.498 4	475.113 7	159.520 2	0
		FPA	22.039 3	60.244 9	85.492 9	14.862 9	0
		IWOFFPA	1.116 6	30.012 0	65.543 0	16.989 3	0
f_6	30	BA	14.670 0	18.842 0	19.868 6	0.649 6	0
		DE	11.367 8	13.504 5	20.2321	9.450 1	0
		FPA	10.825 4	14.555 2	16.5395	1.448 6	0
		IWOFFPA	2.0 185e-07	3.369 0e-07	4.458 8e-07	7.3 074e-08	100
f_7	30	BA	71.027 3	197.259 9	340.621 4	88.914 0	0
		DE	217.812 3	276.699 5	369.623 8	150.497 4	0
		FPA	174.172 2	205.421 2	256.744 9	20.104 0	0
		IWOFFPA	2.984 9	6.168 8	9.949 6	1.858 2	0
f_8	2	BA	-0.990 3	-0.624 9	-0.503 6	0.162 3	0
		DE	-0.993 6	-0.963 2	-0.963 9	0.023 1	0
		FPA	-0.998 6	-0.987 4	-0.962 8	0.008 9	0
		IWOFFPA	-1	-0.990 8	-0.990 3	0.002 2	100

为了更直观地反映出改进算法的寻优效果，图 1 是 4 种算法对 8 个测试函数收敛曲线图，形象展示了 IWOFFPA 算法和 FPA, DE, BA 算法的对比适应度值的迭代下降过程和全局最优解的收敛速度。从图 1 中可以看出，IWOFFPA 算法具有更高的寻优精度和更快的收敛速度。这表明，本文提出的

IWOFFPA 算法具有更好的寻优能力，其收敛速度、优化精度明显优于 FPA、DE 和 BA 算法。综上所述，IWOFFPA 算法与 FPA 算法、DE 算法、BA 算法相比，寻优性能得到了明显提升，也验证了 IWOFFPA 算法是可行的。

图 1 4 种算法在函数 $f_1 \sim f_8$ 上的收敛曲线Fig. 1 Convergence curve of $f_1 \sim f_8$ about 4 algorithm

3.3.2 IWOFPa 和 FPA 在高维函数上的性能分析

对 4 个高维、多峰函数 $f_4 \sim f_7$ 在不同高维条件下用 IWOFPa 和 FPA 算法独立运行 20 次, 函数维数从 50~300 维变化, 以其优化均值及其平均变化率为评价指标, 从而测试本文算法 IWOFPa 的性能随函数维数的增加而变化的情况, 结果如表 2 所示。其中, 平均变化率 = $\sum((\text{后一次维数的优化均值} - \text{前一次维数的优化均值}) / \text{前一次维数的优化均值}) / 5$ 。从表 2 可以看出, 随着函数维数的增加,

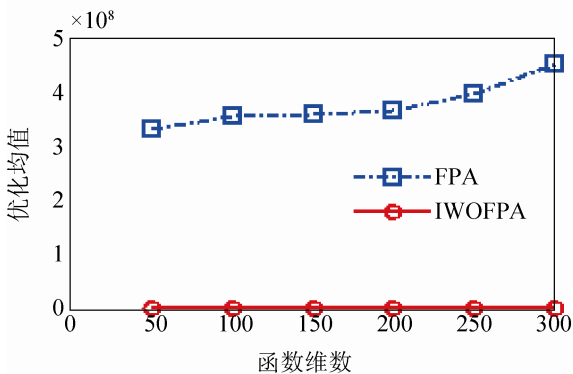
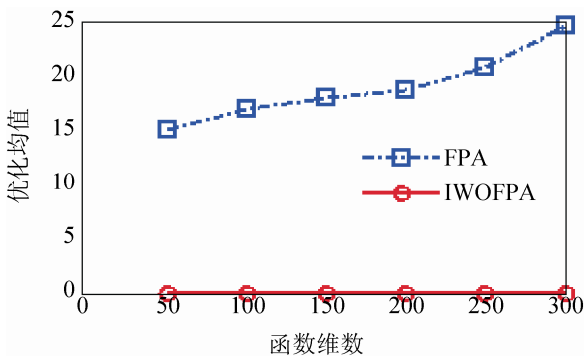
IWOFPa 的优化性能最好, 其优化均值增加的幅度较小, 且都要远小于 FPA, 而 FPA 的优化均值随着函数维数的增加, 优化性能降低, 即优化均值逐渐远离理论最优均值; 另外, 4 个函数的 IWOFPa 算法的平均变化率均要小于 FPA 算法的平均变化率。这说明随着维数的增加, IWOFPa 的优化性能基本不降低, 即不会随着复杂高维多极值函数的维数的增大而陷入“维灾难”, 相对于 FPA 更能突出其优势。

表 2 多峰函数在不同高维上的优化均值比较

Tab. 2 Comparison of optimal mean for multi peak functions on different high dimensions

函数	算法	维数						平均变化率 %
		50	100	150	200	250	300	
f_4	FPA	3.308 1e+08	3.539 4e+08	3.583 3e+08	3.656 7e+08	3.979 2e+08	4.505 9e+08	0.064 7
	IWOFPa	1.458 4e+02	1.491 3e+02	1.521 3e+02	1.560 1e+02	1.586 8e+02	1.6255e+02	0.021 9
f_5	FPA	64.524 4	68.805 3	76.995 1	85.425 2	89.568 2	103.013 2	0.098 6
	IWOFPa	31.096 0	33.101 8	35.377 7	36.322 0	39.281 9	41.476 0	0.059 4
f_6	FPA	14.969 7	16.887 9	17.914 9	18.585 9	20.810 4	24.615 9	0.105 7
	IWOFPa	1.701 5e-06	1.907 2e-06	2.127 0e-06	2.251 9e-06	2.346 9e-06	2.412 4e-06	0.071 8
f_7	FPA	211.406 1	215.042 1	235.900 2	254.004 6	265.263 5	281.406 1	0.059 2
	IWOFPa	6.532 1	6.732 4	7.341 9	7.865 6	8.377 3	8.382 8	0.051 6

为了更直观地对比 IWOFPFA 和 FPA 在高维情况下的性能, 本小节对这两种算法在不同维数下测试 $f_4 \sim f_7$ 函数的优化均值曲线图, 限于篇幅, 只给出 f_4 和 f_6 函数的曲线图, 如图 2、3 所示。从图 2、图 3 可以看出, 对测试函数, FPA 的优化均值随着维数的增加而单调增加, 且增加的幅度较大。而 IWOFPFA 的优化均值在多峰函数中其优化均值增加的幅度较小。因此, 采用入侵杂草策略能有效地使 FPA 跳出局部最优, 在高维函数的优化中, 寻优效果同样要好, 进一步验证了本文算法的可行性和有效性。

图 2 f_4 的优化均值随函数的维数变化曲线Fig. 2 Optimization curve of f_4 with function dimension change图 3 f_6 的优化均值随函数的维数变化曲线Fig. 3 Optimization curve of f_6 with function dimension change

3.3.3 与参考文献算法性能比较

为了进一步验证本算法的鲁棒性和寻优精度, 证明本文算法的优势, 限于篇幅, 只选用函数 f_1 , f_4 , f_6 和 f_7 与参考文献中的 CMIWO^[11], CPSO^[12],

BPSO^[13], CBA^[14], DE^[15], HMDE^[15] 算法对比, 为了文中前后数据的一致性, 本小节中的实验参数与 3.3.1 小节一致, 其仿真实验结果如表 3 所示。由表 3 得知, IWOFPFA 优化性能(优化均值、标准方差)较参考文献中的其它群智能优化算法提高幅度较大。对于函数 f_1 和 f_6 , 本文算法的优化均值和标准方差都显著优于参考文献算法, 分别最多相差 14 个和 11 个数量级; 对于函数 f_4 和 f_7 , 本文算法的优化均值、标准方差优于参考文献[12-13], 但略低于参考文献[11, 14-15], 是因为本文算法的实验条件比参考文献[11, 14-15]要苛刻, 如参考文献[11]中的 CMIWO 算法的维数设为 10 维, 而本文算法的维数为 30 维, 函数维数的增加会一定程度上影响了算法的寻优性能; 对于参考文献[15]中的 DE 和 HMDE 算法, 其种群数(300)和迭代次数(2000)远远高于本文算法, 种群数和迭代次数这两个参数值也是影响算法寻优性能的主要因素, 且这两个参数值与算法的时间复杂度成正比例关系。综上所述, 本文算法在实验条件(如种群数、维数、迭代次数等)更严格的情况下, 本文算法的优化均值、标准方差要好于参考文献中的算法。因此, 本文算法有着更强的鲁棒性和更高的寻优精度, 也表明本文算法是有效可行的。

3.3.4 IWOFPFA 算法的时间复杂度分析

对于一种优秀的群智能算法, 一方面要有好的寻优性能, 另一方面要求时间复杂度低。改进算法是否有效、可行, 除了性能上有较大提高, 算法的时间复杂度应该比较低, 相对于原算法, 运行的时间不能过长。从 IWOFPFA 算法步骤可以看出, Step6~9 是改进算法增加的步骤, 即 IWOFPFA 包括 FPA 基本操作、IWO 的繁殖操作、IWO 空间扩散操作和改进的 IWO 竞争生成操作, 根据上文对这几个操作的详细描述和时间复杂度符号 O 的运算规则, 可以推导出各操作的时间复杂度:

$$T(\text{FPA 基本操作})=O(\text{sizepop}_{\min});$$

$$T(\text{繁殖操作})=O(\text{sizepop}_{\min});$$

$T(\text{空间扩散操作}) = O(\text{sizepop}_{\min})$;

$T(\text{改进的竞争生成操作}) = O(\text{sizepop}_{\min})$ 。

因此, $T(\text{IWOFPFA}) = T(\text{FPA 基本操作}) + T(\text{繁殖操作}) + T(\text{空间扩散操作}) + T(\text{改进的竞争生成操作})$, 化简后可以得到 IWOFPFA 算法的时间复杂度为 $O(\text{sizepop}_{\max})$, 与 FPA 算法的时间复杂度属于同一数量级, 说明 IWOFPFA 算法的时间复杂度没有很大的提高。为了更直观地证明这一点, 从两种算

法的实际运行时间来衡量其时间复杂度的对比, 由于篇幅限制, 本小节只选用测试函数中的多峰函数 f_4 , f_5 和 f_6 三个函数来对本文算法的时间复杂度进行测试并分析, 设置种群数 $\text{sizepop}_{\max}=10$, $\text{sizepop}_{\max}=20$, 进化迭代次数 $\text{maxgen}=300$, 独立运行 20 次, 计算两种算法在不同维数(100 维、300 维和 500 维)所需要的平均运行时间, 结果如表 4 所示(D 为维数)。

表 3 IWOFPFA 算法与参考文献算法的优化均值和标准方差比较

Tab. 3 Optimization mean and standard comparison of IWOFPFA and reference algorithm

函数	性能	IWOFPFA	CMIWO	CPSO	BPSO	CBA	DE	HMDE
f_1	优化均值	5.553 5e-12	3.500e-3	1.07e+2	2.634 2e-02	-	0.001	0.001
	标准方差	2. 145 2e-12	6.900e-3	6.66e+1	1.464 8e-03	-	-	-
f_4	优化均值	140.2490	1.519e+1	4.84e+6	-	258.654 6	8.34e-01	1.044e-03
	标准方差	713.108 0	2.803e+1	5.39e+6	-	-	1.028e+01	2.560e-01
f_6	优化均值	3.369 0e-07	2.126	1.98e+4	0.162 3e+01	222.611 5	7.31e-01	5.688e-05
	标准方差	7.307 4e-08	2.866e-1	6.50e+3	7.147 6e-01	-	5.72e-02	1.065e-05
f_7	优化均值	6.168 8	5.942	2.08e+2	0.613 0e+01	0.455 9	4.3e-02	2.001e-03
	标准方差	1.858 2	1.682	2.67e+1	1.288 7e-01	-	9.41e-02	3.103e-02

表 4 $f_4 \sim f_6$ 的平均运行时间对比

Tab. 4 Average elapsed time comparison of $f_4 \sim f_6$

函数	算法	平均运行时间/s		
		D=100	D=300	D=500
f_4	FPA	0.625 0	1.268 8	1.359 4
	IWOFPFA	2.640 4	3.218 8	4.121 9
f_5	FPA	0.473 4	1.031 3	1.421 9
	IWOFPFA	1.656 3	3.109 4	3.843 8
f_6	FPA	1.015 6	1.359 4	1.593 8
	IWOFPFA	1.734 4	2.481 3	3.328 1

从表 4 的平均运行时间来看, 虽然 IWOFPFA 算法在进入迭代前对花朵个体进行了入侵杂草的操作, 但其平均运行时间比 FPA 算法的平均运行时间只略长一些, 进而说明 IWOFPFA 算法的复杂度较低, 改进算法是可行和有效的。

4 结论

本文针对基本花朵授粉算法存在早熟收敛、收敛精度低、收敛速度慢等不足, 把入侵杂草策略融

入到花朵授粉算法里, 增强种群的有效性和多样性, 进而避免早熟收敛, 提高其收敛速度和收敛精度。通过 8 个 CEC2005 benchmark 测试函数的测试, 仿真结果表明, 改进算法是可行和有效的, IWOFPFA 算法的收敛速度和寻优精度得到较大幅度地提高, 在高维条件下同样表现出较好的优化性能, 并且有较低的复杂度。由于花朵授粉算法的理论和应用研究还处于初始阶段, 还有许多问题有待进一步的研究何解决, 如算法的收敛性分析, 参数设置的理论依据, 以及与其它智能优化算法的有机融合等, 都是将进一步研究的工作。

参考文献:

- [1] Yang X S. Flower pollination algorithm for global optimization [C]// Unconventional Computation and Natural Computation. Berlin, Germany: Springer, 2012: 240-249.
- [2] Yang X S, Karamanoglu M, He X. Multi-objective flower algorithm for optimization [J]. International Conference on Computations Science (S1877-0509), 2013, 18(1): 861-868.

- [3] El-henawy I, Ismail M. An Improved Chaotic Flower Pollination Algorithm for Solving Large Integer Programming Problems [J]. International Journal of Digital Content Technology & its Applications (S1975-9339), 2014, 8(3): 72-81.
- [4] Marwa Sharawi, E Emary, Imane Aly Saroit, et al. Flower Pollination Optimization Algorithm for Wireless Sensor Network Lifetime Global Optimization [J]. International Journal of Soft Computing and Engineering (IJSCE) (S2231-2307), 2014, 4(3): 54-59.
- [5] Osama Abdel-Raouf, Ibrahim El-henawy, Mohamed Abdel-Baset. A Novel Hybrid Flower Pollination Algorithm with Chaotic Harmony Search for Solving Sudoku Puzzles [J]. I. J. Modern Education and Computer Science (S2075-0161), 2014, 7(3): 38-44.
- [6] R Prathiba, M Balasingh Moses, S Sakthivel. Flower Pollination Algorithm Applied for Different Economic Load Dispatch Problems [J]. International Journal of Engineering and Technology (IJET) (S0975-4024), 2014, 6(2): 1009-1016.
- [7] Wang Rui, Zhou Yongquan. Flower pollination algorithm with dimension by dimension improvement [J]. Mathematical Problems in Engineering (S1024-123X), 2014(4): 1-9.
- [8] Lenin K, Reddy B R, Kalavathi M S. Shrinkage of active power loss by hybridization of flower pollination algorithm with chaotic harmony search algorithm [J]. Control Theory and Informatics (S2224-5774), 2014, 4(8): 31-38.
- [9] 王凤鹤, 徐希莲, 陈强, 等. 北京蜜蜂授粉工作研究进展 [J]. 中国蜂业, 2012, 63(9): 28-30. (Wang Fenghe, Xu Xilian, Chen Qiang, et al. The research progress of the bees pollinated in Beijing [J]. Apiculture of china, 2012, 63(9): 28-30.)
- [10] Menrabian A R Lucas C. A novel numerical optimization algorithm inspired from weed colonization [J]. Ecological Informatics (S1574-9541), 2006, 1(4): 355-366.
- [11] 陈欢, 周永权, 赵光伟. 基于混沌序列的多种群入侵杂草算法 [J]. 计算机应用, 2012, 32(7): 1958-1961. (Chen Huan, Zhou Yongquan, Zhao Guangwei. Multi-population invasive weed optimization algorithm based on chaotic sequence [J]. Journal of Computer Applications, 2012, 32(7): 1958-1961.)
- [12] 赵新超, 刘国莅, 刘虎球, 等. 基于非均匀变异和多阶段扰动的粒子群优化算法 [J]. 计算机学报, 2014, 37(9): 2058-2070. (Zhao Xinchao, Liu Guoli, Liu Huqiu, et al. Particle Swarm Optimization Algorithm Based on Non-Uniform Mutation and Multiple Stages Perturbation [J]. Chinese Journal of Computers, 2014, 37(9): 2058-2070.)
- [13] 徐红洋, 田雨波, 黄太安. 基于云变异的云自适应粒子群算法 [J]. 计算机仿真, 2012, 29(11): 251-255. (Xu Hongyang, Tian Yubo, Huang Taian. Cloud Adaptive Particle Swarm Optimization Algorithm Based on Cloud Variation [J]. Computer Simulation, 2012, 29(11): 251-255.)
- [14] 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, 1195. (Liu Changping, Ye Chunming. Bat Algorithm with Chaotic Search Strategy and Analysis of Its property [J]. Journal of system Simulation (S1004-731X), 2013, 25(6): 1183-1188, 1195.)
- [15] 乔俊飞, 傅嗣鹏, 韩红桂. 基于混合变异策略的改进差分进化算法及函数优化 [J]. 控制工程, 2013, 20(5): 943-947. (Qiao Junfei, Fu Sipeng, Han Honggui. A Modified Differential Evolution Algorithm Based on Hybrid Mutation Strategy for Function Optimization [J]. Control Engineering of China, 2013, 20(5): 943-947.)