

文章编号: 2095-2163(2024)01-0076-09

中图分类号: TP301.6

文献标志码: A

融合混沌映射和乘除算子的花授粉算法及应用

夏艺瑄, 贺兴时

(西安工程大学 理学院, 西安 710048)

摘要: 花授粉算法是一种群智能算法,广泛应用于各个领域。本文针对该算法存在收敛精度低、收敛速度慢、稳定性差等不足,提出基于混沌映射和乘除算子的花授粉算法(MDFPA)。首先,利用混合混沌映射更好的随机分布能力生成初始种群,减小随机误差给算法带来的影响;其次,反双曲正切函数有良好的性能,将其引入动态转换概率中,替换固定转换概率,提高算法收敛能力,有利于控制全局搜索和局部搜索之间的平衡;最后,在全局搜索阶段引入乘除算子对花粉位置进行放缩,在空间里进行充分搜索,避免算法陷入局部最优。通过对12个测试函数的仿真实验,表明该算法在单峰、多峰和固定维度测试函数上性能都优于花授粉算法(FPA)、粒子群算法(PSO)、布谷鸟算法(CS)和蜂群算法(ABC),寻优性能显著性提高,具有更快的收敛速度和更优的求解精度。将MDFPA算法应用于PID(Proportion Integration Differentiation)控制器参数优化中,结果表明优化后的PID控制器性能更优。

关键词: 花授粉算法; 收敛速度; 混合混沌映射; 乘除算子; PID参数优化

Flower pollination algorithm fusing chaotic mapping and multiplication-division operators and applications

XIA Yixuan, HE Xingshi

(School of Science, Xi'an Polytechnic University, Xi'an 710048, China)

Abstract: Flower pollination algorithm is a swarm intelligence algorithm that is widely used in various fields. In view of the problems such as low convergence accuracy, slow convergence speed and poor stability of the algorithm, flower pollination algorithm based on chaotic map and multiplication and division operator is proposed. First of all, the better random distribution ability of the hybrid chaotic map is used to generate a more regular and uniform initial population position to reduce the impact of random errors on the algorithm; secondly, because of the good performance of the inverse hyperbolic tangent function, it is introduced into the adaptive transformation probability to replace the fixed conversion probability to improve the convergence ability of the algorithm, which is helpful to control the balance between global search and local search. finally, in the stage of global search, the multiplication and division operator is introduced to scale the pollen position to fully search the space to prevent the algorithm from entering the local optimal value. Through the simulation experiments of twelve test functions, the results show that the performance of the algorithm is better than FPA, PSO, CS and ABC algorithm in unimodal, multi-peak and fixed-dimension test functions. The optimization performance is significantly improved, and the convergence speed and accuracy are faster. The MDFPA algorithm is applied to PID parameter optimization, and the results show that the optimized PID controller has better performance.

Key words: flower pollination algorithm; convergence speed; hybrid chaos mapping; multiplication and division operator; PID parameter optimization

0 引言

近年来,最优化问题的研究随着科技的发展逐渐受到了更多学者的关注,传统的算法无法解决现实中复杂的优化问题。受生物的生存法则和行规

律等启发,提出了不需要依赖问题梯度信息的群智能算法。群智能算法由于其良好的通用性,越来越受到关注,出现了众多算法,如布谷鸟算法(Cuckoo Search, CS)、蚁群算法(Ant Colony Optimization, ACO)、萤火虫算法(Firefly Algorithm, FA)、蝙蝠

基金项目: 陕西省自然科学基金基础研究计划(2023-JC-YB-064)。

作者简介: 夏艺瑄(1999-),女,硕士研究生,主要研究方向:智能算法、概率论与数理统计。

通讯作者: 贺兴时(1960-),男,硕士,教授,主要研究方向:智能算法。Email: xsh1002@126.com

收稿日期: 2023-10-11

算法(Bat Algorithm, BA)、粒子群优化算法(Particle Swarm Optimization, PSO)等。

2012年, Yang^[1]受到显花植物授粉过程的启发, 提出了一种新型群智能算法即花授粉算法(Flower Pollination Algorithm, FPA)。花授粉算法由全局搜索和局部探索两种机制组成, 分别对应显花植物的异花授粉和自花授粉两个过程, 通过调整概率来平衡全局和局部两个搜索阶段。相较于其他群智能算法, 花授粉算法概念简单、参数少、花恒定性容易实现, 成功的应用在许多领域中, 如图像处理、经济调度与调控、路径规划、预测等。同其他群智能算法一样, 花授粉算法仍有一定的不足, 如易陷入局部最优、迭代后期收敛速度慢、寻优精度差等。目前国内外学者针对其不足进行改进, 张水平等^[2]提出利用霍尔顿序列生成初始种群, 改进后生成的初始种群位置分布更加规律和均匀; 宁杰琼等^[3]提出利用 Logistic 映射产生混沌值, 并以此更新初始花粉位置, 使其均匀的分布在搜索空间, 但到了迭代后期, 种群容易丧失多样性, 会导致发生早熟收敛, 且易陷入局部最优。为了在迭代后期能够更好保持种群的多样性, 汪海等^[4]提出在局部授粉阶段引入遗传算法杂交的思想, 花粉之间可以实现信息交流; 王兴凡等^[5]提出在局部授粉阶段加入柯西变异以增加种群多样性, 但对于有些方程组还无法求出全部的解。转换概率控制着全局搜索和局部搜索两个阶段的切换, 这一重要参数会控制算法的进化方向和优化性能。Rodrigues 等^[6]提出在迭代过程中动态地调整转换概率和步长影响因子, 但算法容易陷入局部最优; Liu^[7]等提出将韦伯分布函数和迭代次数结合, 实现动态控制转换概率, 但目前算法在应用方面较为欠缺。为了增强 FPA 的搜索能力, 汪海等^[8]提出了具有记忆信息的花授粉算法, 通过引入时滞调整算子来控制搜索过程, 将自适应调整算子引入自花授粉阶段, 算法更具有灵活性, 收敛精度更高, 但收敛性证明不足且理论基础薄弱。

针对 FPA 算法存在的不足, 本文提出基于混沌映射和乘除算子的花授粉算法, 具体改进如下:

(1) 将 logistic 映射和 sine 映射混合, 使用 logistic-sine 混沌映射来产生更规律、均匀的初始种群;

(2) 提出基于反双曲正切函数的动态转换概率, 替换原本的固定概率, 提高算法的寻优效率;

(3) 将乘除算子引入异花授粉过程中, 针对最优位置进行放缩, 提高迭代初期的寻优随机性。

1 基础花授粉算法

花授粉算法是受到自然界中显花植物授粉现象启发提出的, 一方面花在授粉过程中依赖蜜蜂等传粉者完成异花授粉; 另一方面是在风和其他因素的帮助下完成自花授粉。为了简化模拟开花过程, 提出以下 4 条理想条件:

(1) 异花授粉过程通过 Levy 飞行进行传播;

(2) 非生物授粉通过自花授粉过程模拟, 作为局部授粉;

(3) 花的恒常性与授粉过程中两朵花的相似性成正比例;

(4) 转换概率 P 控制生物或非生物授粉之间的转换。

将其转换为数学表达式进行描述, 异花授粉公式表示为式(1):

$$X_i^{t+1} = X_i^t + L * (g_b - X_i^t), i = 1, 2, \dots, N \quad (1)$$

其中, X_i^{t+1} 和 X_i^t 分别表示第 i 个花粉在第 $t+1$ 次和第 t 次迭代时的位置; g_b 是当前种群的最优解; L 是服从 Levy 分布的随机步长, 用式(2)表示:

$$L \sim \frac{\lambda \Gamma(\lambda) \sin\left(\frac{\lambda \pi}{2}\right)}{\pi} \frac{1}{s^{(1+\lambda)}}, s \gg s_0 \gg 0 \quad (2)$$

其中, $s = \mu / |v|^{1/\lambda}$, $\mu \sim N(0, \sigma^2)$, v 服从标准正态分布。

自花授粉过程更新, 式(3):

$$X_i^{t+1} = X_i^t + \varepsilon * (X_i^t - X_k^t) \quad (3)$$

其中, $\varepsilon \in U(0, 1)$, X_i^{t+1} 和 X_k^t 是不同于 i 的两个随机花粉。

2 改进的花授粉算法

2.1 混合混沌映射产生初始种群

基本的花授粉算法采用随机的方法产生的初始种群存在随机误差, 直接影响了算法的寻优性能, 位置不均匀也会导致寻优时间过长。本文在产生初始种群阶段引入一种结合了 logistic 映射和正弦映射的混合混沌映射。logistic 映射、正弦映射和混合映射的数学描述分别用式(4)~式(6)表示:

$$x_{i+1} = \mu x_i (1 - x_i) \quad (4)$$

$$x_{i+1} = \mu \sin(\pi x_i) / 4 \quad (5)$$

$$x_{i+1} = \mu x_i (1 - x_i) + (4 - \mu) \sin(\pi x_i) / 4 \quad (6)$$

其中, x 是随机生成的数字序列, μ 是混沌的乘数。

三种映射的分岔图如图 1 所示, 可见混合混沌

映射具有更好的随机分布能力。因此,本文选择 logistic-sine 映射产生初始种群。

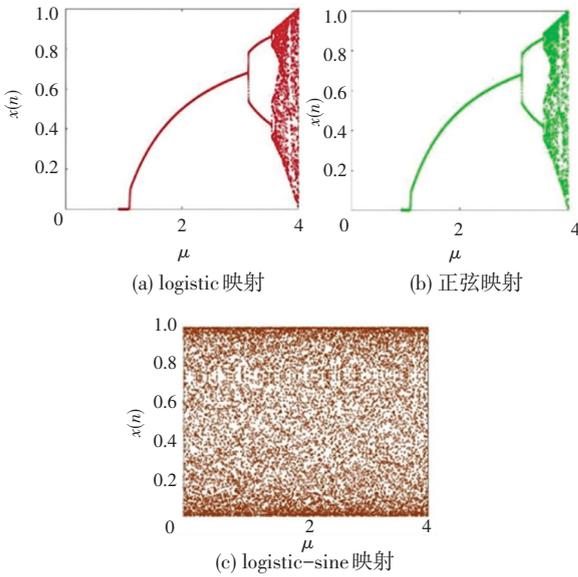


图1 混沌序列分岔图

Fig. 1 The bifurcation diagram of chaotic sequence

2.2 动态转换概率

转换概率 P 控制着花授粉算法异花授粉过程和自花授粉过程的转换,以控制全局搜索和局部搜索之间的平衡,影响着算法的性能^[9]。在基本的 FPA 算法中,固定概率 $P = 0.8$,不能根据迭代次数的增加而进行调整,不利于两种搜索机制之间的平衡,所以考虑在发现概率中引入动态转换概率,使得算法更具有灵活性。由于反双曲正切函数有良好的性能,所以将其引入转换概率中,式(7):

$$P_t = P_{\max} + \operatorname{arctanh}(\log(\pi t/10T))(P_{\max} - P_{\min}) \quad (7)$$

其中, P_{\max} 和 P_{\min} 分别表示概率的最大值和最小值,取 1.0 和 0.25; t 表示当前迭代次数; T 表示最大迭代次数; P_t 是一个关于 t 的非线性单调递减函数,其变化曲线如图 2 所示。

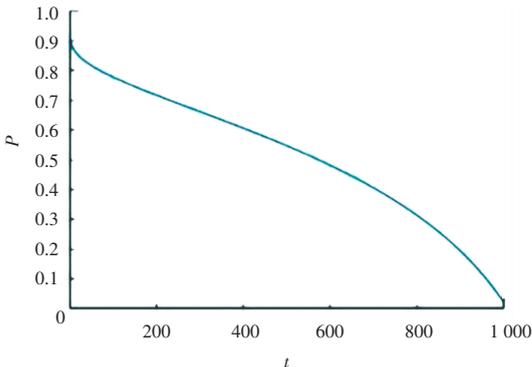


图2 动态转换概率曲线分布

Fig. 2 The curve distribution of adaptive transformation probability

在迭代初期概率较大,算法整体侧重于全局搜索过程;随着迭代次数的增加,算法后期则更加侧重于局部开发,实现在算法搜索前期以较大概率进行全局探索、在算法搜索后期以较大概率进行局部开发,提升了算法的收敛速度。

2.3 乘除算子

为了提高 FPA 跳出局部极值的能力,在异花授粉过程引入新型乘除算子。使用乘法算符和除法算符的数学计算,可以在搜索机制内得到高度分布的值,由于乘除算子的高度分散而无法轻易接近目标^[10]。

在异花授粉过程中,基于除法搜索策略和乘法搜索策略寻找更好的解决方案。在搜索过程中,取随机数 r_2 ,当 $r_2 < 0.5$ 时,引入除法算子,否则引入乘法算子。在算子中考虑了随机缩放系数,以产生更多的多样化过程,并探索搜索空间的不同区域。本文提出了位置更新方程,式(8):

$$X_i^{(t+1)} = \begin{cases} (X_i^t + L * (X_i^t - X_{best}^t)) / (1 - (t/T)^\alpha + \varepsilon) * R, & r_2 < 0.5 \\ (X_i^t + L * (X_i^t - X_{best}^t)) * (1 - (t/T)^\alpha) * R, & \text{else} \end{cases} \quad (8)$$

其中, $R = ((UB - LB) * v) + LB$, X_i^t 表示当前迭代中第 i 个解; $X_i^{(t+1)}$ 表示下一次迭代中的第 i 个解; X_{best}^t 是当前最优解; α 是敏感参数,本文取 0.2; ε 是一个小整数; UB 和 LB 分别表示上限值和下限值; v 是调整搜索过程的控制参数,该参数固定为 0.5。

2.4 算法步骤

Step 1 初始化参数,种群规模 N 、维度 d 、最大迭代次数 T 以及转换概率最大值 P_{\max} 与最小值 P_{\min} ;

Step 2 初始化种群,使用混合混沌映射初始化花粉的位置,计算各初始花粉的适应度值,并记录当前最优花粉位置与适应度值;

Step 3 根据式(7)计算转换概率;

Step 4 若随机数 $rand > P$,则按照式(8)进行异花授粉过程;否则,根据式(3)进行自花授粉过程,更新花粉位置;

Step 5 更新最优解,并计算全局最优值。判断是否满足终止条件,若满足则输出全局最优解,否则转至 Step 3,开始下一次迭代。

3 实验仿真与分析

3.1 测试环境与测试函数

本文实验采用 Intel® Core™ i9-12900H CPU,主频为 2.50 GHz,内存 16 GB,64 位操作系统计算

机,编程环境为 Matlab R2022b。为了检验算法的有效性,本文选取了不同难度的 12 个测试函数对其进行性能测试,见表 1。其中, $f_1(x) \sim f_6(x)$ 是只有一个全局最优解的单峰测试函数; $f_7(x) \sim f_{10}(x)$ 为具有多个局部最优值和一个全局最优值的多峰测试函

数。设置维度为 50 和 100, 分别进行实验记录; $f_{11}(x) \sim f_{12}(x)$ 是固定维度测试函数, 维度 $D = 2$ 。采用平均值、标准差作为评价指标, 平均值可以很好的检验算法的收敛精度和搜索能力; 标准差可以反映算法的寻优稳定性。

表 1 测试函数
Table 1 Test functions

函数名称	函数表达式	解空间	最优值
Sphere	$f_1(x) = \sum_{i=1}^n x_i^2$	$[-100, 100]$	0
Schwefel2.22	$f_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	$[-10, 10]$	0
Schwefel1.2	$f_3(x) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j \right)^2$	$[-100, 100]$	0
Sum squares	$f_4(x) = \sum_{i=1}^n ix_i^2$	$[-10, 10]$	0
Sum of Different Powers	$f_5(x) = \sum_{i=1}^d x_i ^{i+1}$	$[-1, 1]$	0
Zakharov	$f_6(x) = \sum_{i=1}^n x_i^2 + \left(\frac{1}{2} \sum_{i=1}^n ix_i \right)^2 + \left(\frac{1}{2} \sum_{i=1}^n ix_i \right)^4$	$[-5, 10]$	0
Rastrigin	$f_7(x) = \sum_{i=1}^d [x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-5.12, 5.12]$	0
Griewank	$f_8(x) = 1 + \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-100, 100]$	0
Powell	$f_9(x) = \sum_{i=1}^{d/4} [(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4]$	$[-4, 5]$	0
Alpine	$f_{10}(x) = \sum_{i=1}^n x_i \sin(x_i) + 0.1x_i $	$[-10, 10]$	0
Three-Hump Camel	$f_{11}(x) = 2x_1^2 - 1.05x_1^4 + x_1^6/6 + x_1x_2 + x_2^2$	$[-5, 5]$	0
Matyas	$f_{12}(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$	$[-10, 10]$	0

3.2 对比算法及参数设置

本文选取 FPA、PSO、CS 和 ABC 4 种算法进行对比实验。为了更准确地验证算法的优劣, 设定种群规模 $N = 25$, 最大迭代次数 1 000 次, 各算法的相关参数设置见表 2。

表 2 算法参数设置

Table 2 Algorithm parameter settings

算法	参数设置
MDFPA	$P_{\max} = 1, P_{\min} = 0.2, \mu = 0.5, \alpha = 0.5$
FPA	$P = 0.8, n = 25$
PSO	$c_1 = c_2 = 2, \omega = 1, \omega_{\text{damp}} = 0.81$
CS	$P = 0.25, n = 25$
ABC	$\delta = 1.5, n = 25$

3.3 算法求解精度比较

在相同的测试环境运用 5 种算法对 12 个测试函数分别运行 50 次, 实验结果见表 3~表 14。

表 3 函数 $f_1(x)$ 仿真结果

Table 3 Simulation results of function $f_1(x)$

算法	$D = 50$		$D = 100$	
	Std	Mean	Std	Mean
MDFPA	0	0	0	0
FPA	3.55E+02	1.30E+03	6.19E+02	5.55E+03
PSO	2.11E+01	1.20E+02	1.76E+02	1.33E+03
CS	1.84E+00	4.50E+00	1.29E+02	6.27E+02
ABC	4.52E+02	1.72E+03	4.26E+03	6.39E+04

表4 函数 $f_2(x)$ 仿真结果Table 4 Simulation results of function $f_2(x)$

算法	$D = 50$		$D = 100$	
	Std	Mean	Std	Mean
MDFPA	0	0	0	0
FPA	3.58E+00	2.77E+01	7.82E+00	7.46E+01
PSO	1.71E+01	3.19E+01	3.10E+01	7.81E+01
CS	4.03E+09	2.80E+09	0.00E+00	1.00E+10
ABC	1.22E+00	5.31E+00	5.85E+04	3.35E+04

表5 函数 $f_3(x)$ 仿真结果Table 5 Simulation results of function $f_3(x)$

算法	$D = 50$		$D = 100$	
	Std	Mean	Std	Mean
MDFPA	0	0	0	0
FPA	5.45E+02	1.90E+03	1.54E+03	8.94E+03
PSO	1.75E+03	6.46E+03	8.65E+03	4.04E+04
CS	2.19E+02	1.16E+03	1.50E+03	9.74E+03
ABC	6.34E+03	1.04E+05	2.75E+04	4.12E+05

表6 函数 $f_4(x)$ 仿真结果Table 6 Simulation results of function $f_4(x)$

算法	$D = 50$		$D = 100$	
	Std	Mean	Std	Mean
MDFPA	0	0	0	0
FPA	5.45E+01	2.75E+02	4.24E+02	2.37E+03
PSO	9.59E+02	8.14E+02	5.56E+03	6.06E+03
CS	4.84E-01	1.28E+00	5.81E+01	2.37E+02
ABC	4.70E+01	2.48E+02	2.06E+03	2.11E+04

表7 函数 $f_5(x)$ 仿真结果Table 7 Simulation results of function $f_5(x)$

算法	$D = 50$		$D = 100$	
	Std	Mean	Std	Mean
MDFPA	0	0	0	0
FPA	1.54E-08	1.33E-08	2.33E-07	1.49E-07
PSO	9.75E-06	1.33E-05	9.46E-06	1.31E-05
CS	7.52E-12	5.31E-12	7.96E-10	5.29E-10
ABC	6.33E-02	1.72E-01	3.24E-01	1.32E+00

表8 函数 $f_6(x)$ 仿真结果Table 8 Simulation results of function $f_6(x)$

算法	$D = 50$		$D = 100$	
	Std	Mean	Std	Mean
MDFPA	0	0	0	0
FPA	4.07E+01	1.50E+02	1.10E+02	5.77E+02
PSO	3.85E+02	1.69E+03	5.31E+02	4.45E+03
CS	4.82E+01	2.79E+02	1.47E+09	9.20E+09
ABC	4.01E+01	6.47E+02	7.84E+01	1.71E+03

表9 函数 $f_7(x)$ 仿真结果Table 9 Simulation results of function $f_7(x)$

算法	$D = 50$		$D = 100$	
	Std	Mean	Std	Mean
MDFPA	0	0	0	0
FPA	3.31E+01	2.10E+02	4.48E+01	5.02E+02
PSO	4.06E+01	3.88E+02	6.25E+01	8.78E+02
CS	1.53E+01	1.42E+02	3.05E+01	3.58E+02
ABC	1.72E+01	4.81E+02	4.16E+01	1.20E+03

表10 函数 $f_8(x)$ 仿真结果Table 10 Simulation results of function $f_8(x)$

算法	$D = 50$		$D = 100$	
	Std	Mean	Std	Mean
MDFPA	0	0	0	0
FPA	2.47E+00	1.24E+01	7.78E+00	5.20E+01
PSO	2.02E-02	1.02E+00	5.80E-02	1.42E+00
CS	3.88E-02	1.03E+00	1.20E+00	6.46E+00
ABC	3.64E+00	1.63E+01	5.49E+01	5.64E+02

表11 函数 $f_9(x)$ 仿真结果Table 11 Simulation results of function $f_9(x)$

算法	$D = 50$		$D = 100$	
	Std	Mean	Std	Mean
MDFPA	0	0	0	0
FPA	1.11E+01	5.49E+01	1.24E+02	4.51E+02
PSO	0	0	0	0
CS	2.76E+03	3.38E+03	2.40E+03	4.40E+03
ABC	4.42E-01	8.42E-01	1.11E+01	3.91E+01

表12 函数 $f_{10}(x)$ 仿真结果Table 12 Simulation results of function $f_{10}(x)$

算法	$D = 50$		$D = 100$	
	Std	Mean	Std	Mean
MDFPA	0	0	0	0
FPA	3.07E+00	2.35E+01	5.93E+00	5.41E+01
PSO	6.84E+00	1.42E+01	1.12E+01	3.31E+01
CS	1.71E+00	1.33E+01	3.02E+00	3.53E+01
ABC	3.41E+00	4.79E+01	9.22E+00	1.49E+02

表13 函数 $f_{11}(x)$ 仿真结果Table 13 Simulation results of function $f_{11}(x)$

算法	$D = 2$	
	Std	Mean
MDFPA	0	0
FPA	3.08E-15	2.32E-15
PSO	1.66E-06	1.72E-06
CS	3.72E-42	2.34E-42
ABC	2.13E-08	2.67E-08

表 14 函数 $f_{12}(x)$ 仿真结果

Table 14 Simulation results of function $f_{12}(x)$

算法	$D = 2$	
	Std	Mean
MDFPA	0	0
FPA	1.47E-17	1.09E-17
PSO	5.02E-08	4.70E-08
CS	1.89E-45	1.07E-45
ABC	6.89E-07	9.15E-07

$f_1(x) \sim f_6(x)$ 均为单峰函数,能够验证算法的收敛速度和局部开发能力。 $f_1(x)$ 函数通常被用于评价算法的后期搜索能力,因为其全局最小值附近函数变化极慢; $f_4(x)$ 通常用来检查算法的寻优能力。由表 3~表 8 可以看出,对比算法在求解单峰函数时,均陷入局部最优值或收敛速度过慢,而 MDFPA 在 50 维、100 维下,均可收敛到理论最优值,表现出显著的寻优能力。

多峰函数多用于检验算法的全局搜索能力和跳出局部极值的能力。 $f_7(x)$ 是比较典型的非线性多模态函数,其局部最优值个数和维数成正比且跌宕起伏不定,维数越大则越容易陷入局部最优值,由表 9 可以看出 4 个对比算法的寻优能力随着维度的增加而减弱且都未收敛到理论最优值,而 MDFPA 都能够达到理论最优值。 $f_8(x)$ 是经典的多模态函数,是智能算法最难求解的问题,因为其搜索空间大并且有许多的局部最优值。由表 10 可见,MDFPA 均可以寻得全局最优值,而其他的 5 种算法均不能达到。 $f_9(x)$ 和 $f_{10}(x)$ 是复杂的多峰函数,MDFPA 也表现出显著的搜索能力和跳出局部最优的能力,能够获得全局最优值。PSO 算法在 $f_9(x)$ 函数上能够达到全局最优值,但在其他函数上未能达到,而 FPA、CS 和 ABC 在多峰测试函数上都无法达到理论最优值。

$f_{11}(x)$ 和 $f_{12}(x)$ 是固定维度测试函数,检验算法的全局搜索和局部搜索之间的性能。MDFPA 均可稳定的收敛到理论最优值,而其他 4 种算法均未收敛到理论最优值。

3.4 算法收敛曲线分析

MDFPA 和其他 5 种对比算法在函数上的收敛曲线能够直观地反映出 5 种算法的收敛速度和收敛精度,如图 3~图 12 所示,横坐标为算法的迭代次数,纵坐标为算法适应度的对数。

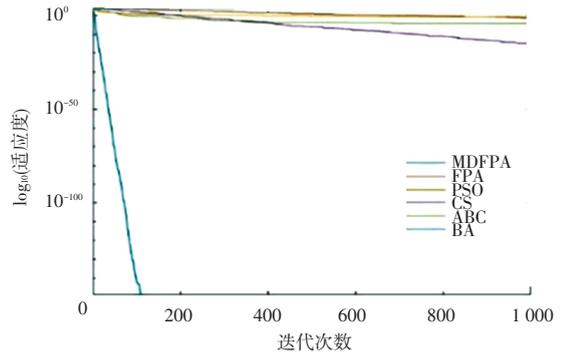


图 3 函数 $f_1(x)$ 收敛曲线

Fig. 3 The convergence curves of function $f_1(x)$

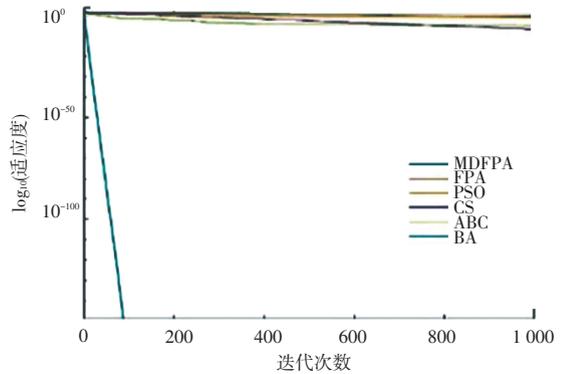


图 4 函数 $f_2(x)$ 收敛曲线

Fig. 4 The convergence curves of function $f_2(x)$

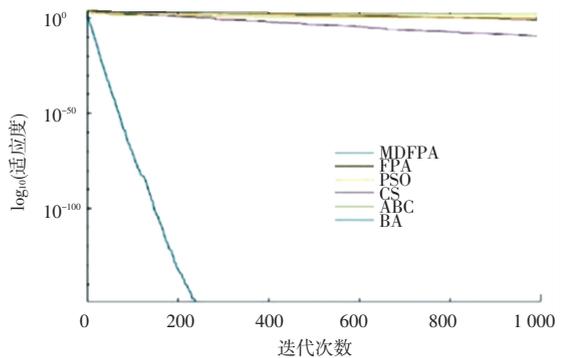


图 5 函数 $f_3(x)$ 收敛曲线

Fig. 5 The convergence curves of function $f_3(x)$

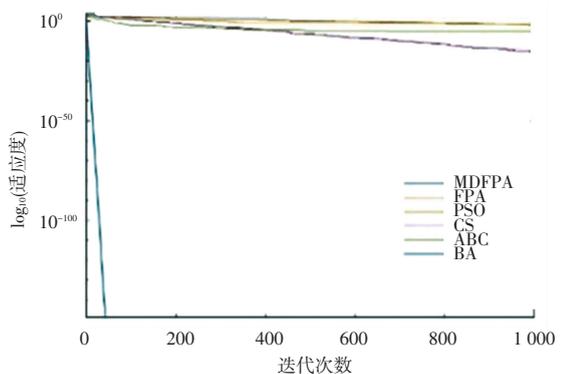
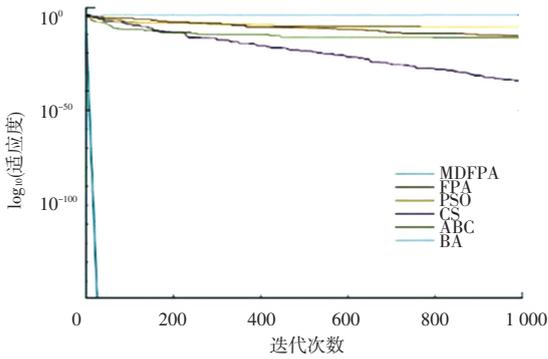
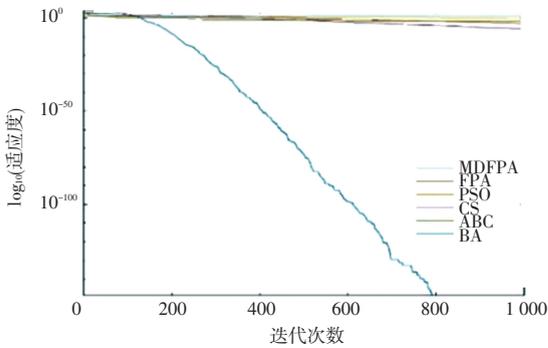
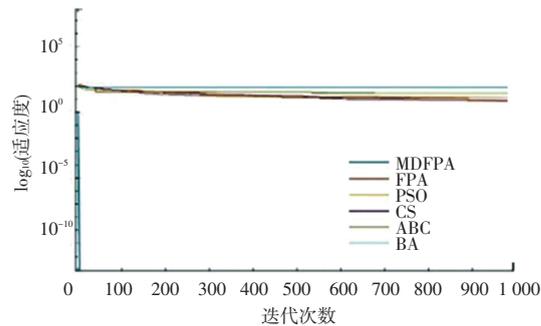
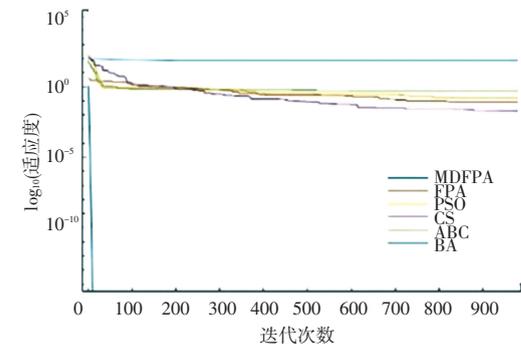
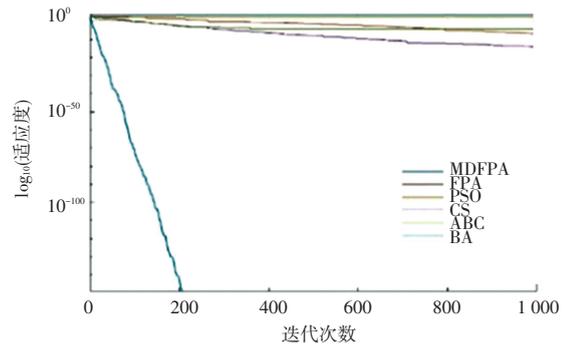
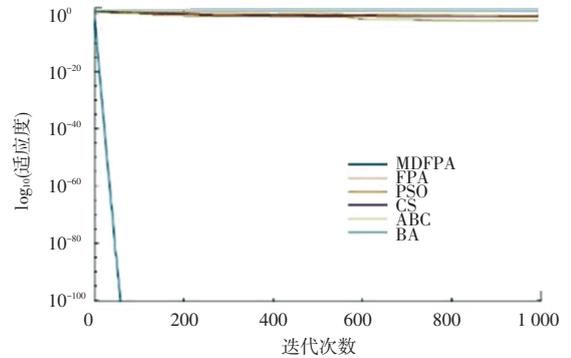
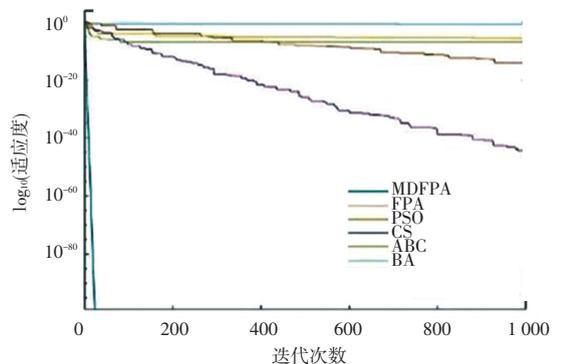


图 6 函数 $f_4(x)$ 收敛曲线

Fig. 6 The convergence curves of function $f_4(x)$

图7 函数 $f_5(x)$ 收敛曲线Fig. 7 The convergence curves of function $f_5(x)$ 图8 函数 $f_6(x)$ 收敛曲线Fig. 8 The convergence curves of function $f_6(x)$ 图9 函数 $f_7(x)$ 收敛曲线Fig. 9 The convergence curves of function $f_7(x)$ 图10 函数 $f_8(x)$ 收敛曲线Fig. 10 The convergence curves of function $f_8(x)$ 图11 函数 $f_9(x)$ 收敛曲线Fig. 11 The convergence curves of function $f_9(x)$ 图12 函数 $f_{10}(x)$ 收敛曲线Fig. 12 The convergence curves of function $f_{10}(x)$

$f_{11}(x) \sim f_{12}(x)$ 在固定维度 $D = 2$ 时的函数收敛曲线如图13和图14所示, MDFPA也表现出了较强的收敛能力,可以看出FPA算法的收敛速度较慢,收敛精度一般,而改进后的MDFPA算法无论在低维度还是高维度、单峰还是多峰函数都具有较好的收敛精度和较快的收敛速度,算法具有较强的搜索能力。

图13 函数 $f_{11}(x)$ 收敛曲线Fig. 13 The convergence curves of function $f_{11}(x)$

维度为50时,由图3~图12中 $f_1(x) \sim f_{10}(x)$ 的函数收敛曲线可以看出, MDFPA算法在收敛速度上明显优于FPA、PSO、CS和ABC算法,能够在最小的迭代次数内收敛到理论最优值。 $f_1(x) \sim f_6(x)$ 是单峰

函数,用来测试算法的收敛速度,可以看出 MDFPA 算法有较好的寻优能力和较快的收敛速度;而 $f_7(x) \sim f_{10}(x)$ 是多峰函数,用来测试算法跳出局部最优的能力,可见 MDFPA 没有出现梯形曲线的情况,克服了容易陷入局部最优值的缺点。

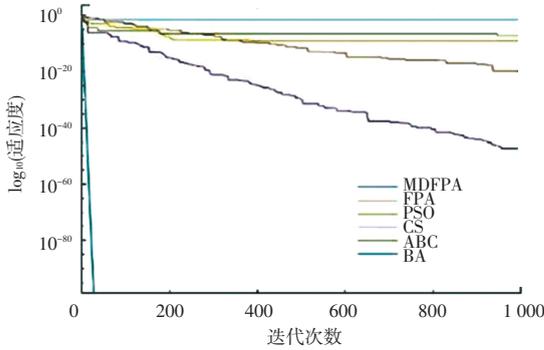


图 14 函数 $f_{12}(x)$ 收敛曲线

Fig. 14 The convergence curves of function $f_{12}(x)$

3.5 统计检验

Wilcoxon 符号秩检验是一种非参的假设检验,用于判断两个数据集是否来自相同分布的总体。本文采用 Wilcoxon 符号秩检验来判断 MDFPA 算法与其他算法是否有显著性差异。取每种算法维度设置为 50 时在 12 个函数上运行 50 次的误差平均值进行符号秩检验,结果见表 15。

表 15 Wilcoxon 符号秩检验结果

Table 15 The wilcoxon signed-rank test results

	MDFPA	FPA	PSO	CS	ABC
+/-/≈	-	0/15/0	0/14/1	0/14/1	0/15/0
R+/R-	-	120/0	105/0	120/0	120/0
p-value	-	0.001	0.001	0.001	0.001
<0.05	-	YES	YES	YES	YES
Rank	1.03	3.47	3.90	2.67	4.07

符号“+”、“-”、“≈”表示以 MDFPA 算法为参考算法时的符号检验结果,符号“R+”和“R-”分别表示对应符号的秩和。p-value 表示 Wilcoxon 检验结果,当值小于 0.05 时,认为 MDFPA 算法优于另一种算法。由表 15 可以看出,MDFPA 与其他算法在这 12 个函数上的 P 值远远小于 0.05,认为 MDFPA 在求解上述函数时比 FPA、PSO、CS 和 ABC 都有显著性优势。

4 PID 参数优化

PID 控制器是由比例单元 P、积分单元 I 和微分单元 D 组成的一种线性控制。针对不同的问题时,参数的调整起到决定作用,必须进行合理的参数调

整。

本文选取目标函数,式(9):

$$F = \int_0^{\infty} (w_1 |e(t)| + w_2 u^2(t)) dt \quad (9)$$

其中, $e(t)$ 是输入值和输出值的误差; $u(t)$ 是控制量; w_1 和 w_2 是取值范围为 $[0, 1]$ 的权重。

二阶系统传递函数,式(10):

$$G(s) = \frac{50}{0.125s^2 + 7s} \quad (10)$$

其中, s 为复频率。

将 MDFPA 与 FPA、PSO、CS、BAT 作为对比算法对二阶系统进行仿真,得到的优化适应度曲线和系统阶跃响应输出曲线如图 15 和图 16 所示。由图 15 可见,MDFPA 算法收敛速度较快,易找到全局最优解,其他算法易出现梯形折线;由图 16 所示,MDFPA 算法的超调量较小,且调整时间也较短,说明 MDFPA 算法优化参数的系统稳定性好。

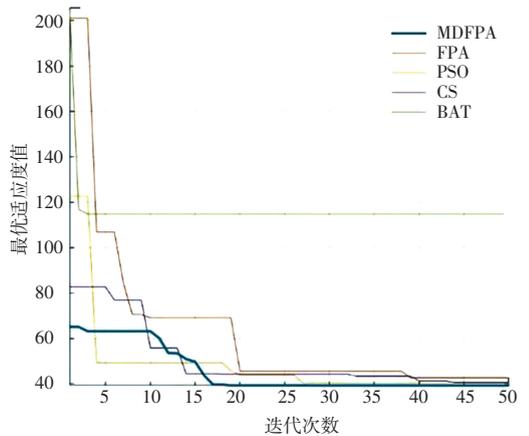


图 15 适应度变化曲线

Fig. 15 Variation curve of fitness

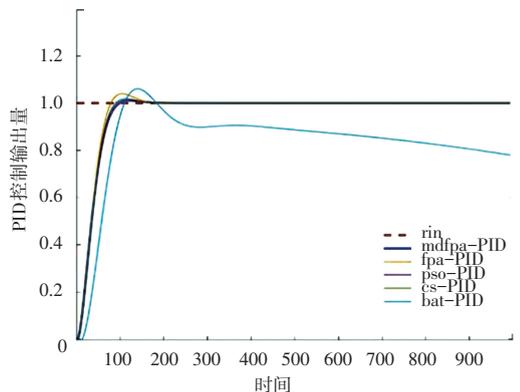


图 16 系统阶跃响应输出曲线

Fig. 16 System step response output curve

5 结束语

针对花授粉算法易陷入局部最优、迭代后期收

敛速度慢、寻优精度差等不足,本文分析基本花授粉算法的寻优过程,提出了基于混沌映射和乘除算子的花授粉算法。首先,在产生初始种群时,将 logistic 混沌映射和 sine 混沌映射混合产生 logistic-sine 混合混沌映射来产生初始种群,可以产生更规律、均匀的初始种群位置,提高算法的寻优效率。用动态转换概率替换原本的固定概率,并将融合数学优化器概率的乘除算子加入异花授粉过程中,能够针对最优位置进行放缩,提高迭代初期的寻优随机性。通过对 12 个测试函数的仿真实验,对比分析了 FPA 算法、PSO 算法、CS 算法和 ABC 算法。实验结果表明,改进后的 MDFPA 算法具有较快的收敛速度和较好的收敛精度,适用于低维、高维的复杂函数问题,在 PID 的参数优化中也表现出良好的性能,下一步将在 MDFPA 的参数和应用方面做进一步的研究。

参考文献

[1] YANG X S. Flower pollination algorithm for global optimization

(上接第 75 页)

杂的新闻故事结构其检测效果有待提高。下一步可以基于深度学习神经网络的模型框架,以视频特征编码方式为依据,对视频密集描述、段落描述等分析新闻视频边界,提升新闻视频分割的多样性。

参考文献

- [1] 姜帆,章毓晋. 新闻视频的场景分段索引及摘要生成[J]. 计算机学报,2003(7):859-865.
- [2] 谭洁,吴玲达,应龙. 基于动画特征的视频摘要方法[J]. 计算机应用研究,2009,26(10):3960-3962.
- [3] 陈家兴. 基于局部特征的视频检索算法研究[D]. 广州:华南理工大学,2016.
- [4] 冀中,苏育挺,宋星光,等. 基于上下文信息的新闻故事单元分割[J]. 天津大学学报,2009,42(2):153-158.
- [5] 王娟,蒋兴浩,孙铤锋. 基于边界归类的新闻视频故事分割算法[J]. 上海交通大学学报,2016,50(9):1384-1389.
- [6] 刘嘉琦,封化民,闫建鹏. 基于多模态特征融合的新闻故事单元分割[J]. 计算机工程,2012,38(24):161-165.
- [7] WANG H, GUO L. Research on face recognition based on deep learning[C]//Proceeding of the 2021 3rd International Conference on Artificial Intelligence and Advanced Manufacture (AIAM). IEEE, 2021: 540-546.

- [C]//Unconventional Computation and Natural Computation, Lecture Notes in Computer Science, 2012: 240-249.
- [2] 张水平,高栋. 基于动态调整和协同搜索的花授粉算法[J]. 计算机工程与应用,2019,55(24):46-53.
- [3] 宁杰琼,何庆. t-分布扰动策略和变异策略的花授粉算法[J]. 小型微型计算机系统,2021,42(1):64-70.
- [4] 汪海,赵齐辉,刘升. 带杂交算子的自适应混合花粉授粉算法[J]. 计算机与数字工程,2018,46(6):1069-1073.
- [5] 王兴凡,孟继志. 基于柯西变异的自适应花粉授粉算法[J]. 智能计算机与应用,2018,8(4):106-111.
- [6] RODRIGUES D, DE ROSA G H, PASSOS L A, et al. Adaptive improved flower pollination algorithm for global optimization [C]//YANG XS, HE XS. Nature-Inspired Computation in Data Mining and Machine Learning. Cham: Springer, 2020: 1-21.
- [7] LIU J, LIU L, LI Y. A differential evolution flower pollination algorithm with dynamic switch probability[J]. Chinese Journal of Electronics, 2019, 28(4): 737-747.
- [8] 汪海,赵齐辉,刘升. 具有记忆信息的花授粉算法[J]. 上海工程技术大学学报,2018,32(1):68-72.
- [9] 第五杨萌,贺兴时. 基于随机惯性权重的自适应花粉算法[J]. 纺织高校基础科学学报,2020,33(3):81-85.
- [10] ABUALIGAH L, DIABAT A, MIRJALILI S, et al. The arithmetic optimization algorithm [J]. Computer Methods in Applied Mechanics and Engineering, 2021, 376: 11360.

- [8] 周家亮. 基于聚类算法的视频浓缩技术研究[D]. 武汉:华中科技大学,2015.
- [9] YANG G, FENG W, JIN J, et al. Face mask recognition system with YOLOV5 based on image recognition [C]//2020 IEEE 6th International Conference on Computer and Communications (ICCC). IEEE, 2020: 1398-1404.
- [10] 梁学战,朱明. 融合多特征的新闻故事自动分割[J]. 计算机应用,2009,29(4):959-961.
- [11] 余伶俐,周开军,邱爱兵. 基于 Elman 神经网络的语音情感识别应用研究[J]. 计算机应用研究,2012,29(5):1809-1814.
- [12] ZHU G, IGLESIAS C A. Computing semantic similarity of concepts in knowledge graphs [J]. IEEE Transactions on Knowledge and Data Engineering, 2016, 29(1): 72-85.
- [13] WU Q, LI Q, ZHOU J J, et al. A film and TV news digest generation method based on HanLP [C]//Proceeding of the 2020 IEEE Intl Conf on Parallel and Distributed Processing with Applications, Big Data and Cloud Computing, Sustainable Computing and Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom). IEEE, 2020: 1197-1204.
- [14] FENG B, CHEN Z, ZHENG R, et al. Multiple style exploration for story unit segmentation of broadcast news video [J]. Multimedia Systems, 2014, 20: 347-361.