

启发信息引导的改进萤火虫算法

崔家瑞, 李 擎, 杨柳祎, 王 恒, 张博钰

(北京科技大学 自动化学院, 北京 100083)

摘 要: 萤火虫算法 (FA) 是一种群体智能优化算法, 它基于萤火虫的闪烁和吸引特征模拟萤火虫的社会行为。为解决萤火虫算法后期收敛速度慢, 易陷入局部最优的不足, 对算法进行了改进。提出了两种启发信息引导算法收敛: 第一种借鉴粒子群算法中“全局最优”的思想, 将当前最优的位置作为启发信息, 形成了基于当前全局最优的萤火虫算法 (FAGO); 第二种将贝叶斯估计计算出的最优移动方向作为启发信息, 形成了基于贝叶斯估计的萤火虫算法 (FABE)。最后, 将本文算法在多个常见函数上进行了测试, 并与经典萤火虫算法、近年其他文献改进萤火虫算法进行了对比研究, 结果表明本文所提算法能够加快收敛速度, 提高收敛精度。

关键词: 萤火虫算法; 启发信息; 全局最优; 贝叶斯估计; 数值优化

DOI: 10.15938/j.jhust.2019.01.015

中图分类号: TP18 文献标志码: A 文章编号: 1007-2683(2019)01-0092-07

Improved Firefly Algorithm Based on Heuristic Information

CUI Jia-rui, LI Qing, YANG Liu-yi, WANG Heng, ZHANG Bo-yu

(School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China)

Abstract: Firefly Algorithm (FA) is an optimization algorithm based on swarm intelligence which mimics the social behavior of fireflies based on the flashing and attraction characteristics of fireflies. With the aim to address the disadvantages of the firefly algorithm of slow convergence speed and ease of falling into the local optimum in the later period of the evolution process, the firefly algorithm is improved herein. Two kinds of heuristic information are proposed into the algorithm to guide the convergence of the algorithm. The first one takes the current global best as the heuristic information referencing the “global optimal” idea in particle swarm optimization, therefore, an algorithm called FAGO (Firefly Algorithm based on Global Optimization) is formed. The second one is called FABE (Firefly Algorithm based on Bayesian Estimation) using the optimal moving direction calculated by Bayesian estimation as heuristic information. The improved algorithms in this study are applied to numerical simulations of several classical test functions and compared with traditional FA and some other's research are carried out. The simulation results show that the proposed algorithms can well accelerate the convergence speed and improve the convergence accuracy.

Keywords: firefly algorithm; heuristic information; global optimal; Bayesian estimation; numerical optimization

收稿日期: 2018-10-19

基金项目: 国家自然科学基金(61673098, 61603034); 北京市自然科学基金(3182027); 北京市重点学科共建项目(XK100080537)。

作者简介: 李 擎(1971—), 男, 教授, 博士研究生导师;

杨柳祎(1994—), 男, 硕士研究生。

通信作者: 崔家瑞(1982—), 男, 高级工程师, 硕士研究生导师, E-mail: cuijiarui@ustb.edu.cn。

0 引言

萤火虫算法(Firefly Algorithm, FA)是一种典型的启发式优化算法,由 Yang XS 于 2008 年提出^[1],在图像处理^[2]、光谱数据分类^[3]、流水线调度^[4]等方面得到广泛应用。经典萤火虫算法参数少、原理清晰且易于实现,同时也存在后期收敛速度慢、收敛精度不高、易陷入局部最优的缺陷,为弥补这种缺陷,大量学者对其进行了改进,并取得了一些研究成果。具有代表性的改进方法包括基于参数动态调整的改进算法和基于融合思想的改进算法。文[5]介绍了典型的动态调整参数方法,将萤火虫算法和 Levy 飞行策略结合,利用 Levy 飞行重尾分布的特性,对萤火虫算法的随机移动步长进行了改进,提出了 Levy 飞行萤火虫算法(Levy-flight Firefly Algorithm, LFA);文[6]提出了将萤火虫算法与差分进化方法融合的思路,将较差的萤火虫个体组成新种群,利用差分进化方法进行迭代,较优个体保持原算法不变,提高了算法摆脱局部最优的能力,称为混合进化萤火虫算法(Hybrid Evolutionary Firefly Algorithm, HEFA)。

本文借鉴粒子群算法,引入全局最优位置作为启发信息;结合贝叶斯公式基于先验概率能最大化后验概率的优点,引入计算出的最优移动方向作为启发信息。实验结果表明,两种启发信息各有优势和适用环境,但都能在一定程度上解决后期易陷入局部最优的问题。与现有算法相比,本文所提出的改进方法在很大程度上提高了算法最优解和整体平均解的收敛精度,减少了达到最优解所需的迭代次数。

1 经典萤火虫算法

萤火虫算法模拟自然界中萤火虫个体发光的生物学特性发展而来,其核心思想是:分布在解空间中的萤火虫基于适应度的大小发出不同亮度的光,亮度高的萤火虫会吸引亮度低的萤火虫向其靠拢^[7],若将萤火虫个体的位置作为其对应的解,亮度作为该解的适应度,亮度高的个体持续吸引亮度低的个体从而进行迭代,在迭代的最后,亮度最高的个体所处的位置即为得到的最优可行解(以下简称最优解)。

经典萤火虫算法步骤如下:

步骤 1: 初始化,定义算法参数,包括迭代次数、萤火虫群体规模等;

步骤 2: 在解空间的不同位置随机生成萤火虫,给出初始亮度;

步骤 3: 对一只萤火虫,观察其他萤火虫到自身的相对亮度,向吸引度大于自身亮度的萤火虫个体进行移动,完成一只萤火虫的一次迭代过程;

萤火虫到其自身距离 r 处的相对亮度 I_r 由下式表示:

$$I_r = I_0 e^{-\gamma r^2} \quad (1)$$

上式中, I_0 表示萤火虫在自身位置处的亮度(即适应度),与待优化问题有关, γ 为光吸收因子,表征了传播媒介对于光强的吸收能力。

$k+1$ 时刻,萤火虫 i 向着萤火虫 j 移动,其位置更新公式为:

$$s_i^{k+1} = s_i^k + \beta_0 e^{-\gamma r_{ij}^2} (s_j^k - s_i^k) + \alpha \varepsilon_i^{k+1} \quad (2)$$

上式中, r_{ij} 为萤火虫 i 与 j 之间的欧式距离, α 是步长因子,取值范围一般为 $[0, 1]$,服从高斯分布或者均匀分布; ε_i^{k+1} 是该时刻萤火虫 i 对应的随机向量,决定了在每个维度上进行移动的方向。

步骤 4: 对所有萤火虫个体执行步骤 3 中操作,完成整个萤火虫群体的一次迭代;

步骤 5: 根据情况选择合适的终止条件,判断是否满足,不满足则转步骤 3;

步骤 6: 满足终止条件时,找出此时亮度最大的萤火虫所处的位置,即为得到的最优解。

2 萤火虫算法的改进

2.1 启发信息引入的必要性

标准萤火虫算法中,个体的移动以亮度作为依据,向更亮个体移动的策略保证了算法能够收敛至一个较小的区域内,但无法保证解空间的多样性,极易陷入局部最优,为此,算法提出者在萤火虫个体的移动策略中增加了随机移动项以扩展其全局搜索能力。但现有的随机移动机制导致萤火虫个体容易丢失其已经获得的最优解优势,降低搜索效率,导致算法收敛速度过慢。

基于此,有必要引入其他启发信息,为萤火虫个体提供除亮度外的移动依据。本文设计了两种不同的启发信息,形成了两种不同的改进萤火虫算法。两种引入启发信息的萤火虫算法步骤相同,如图 1 所示:

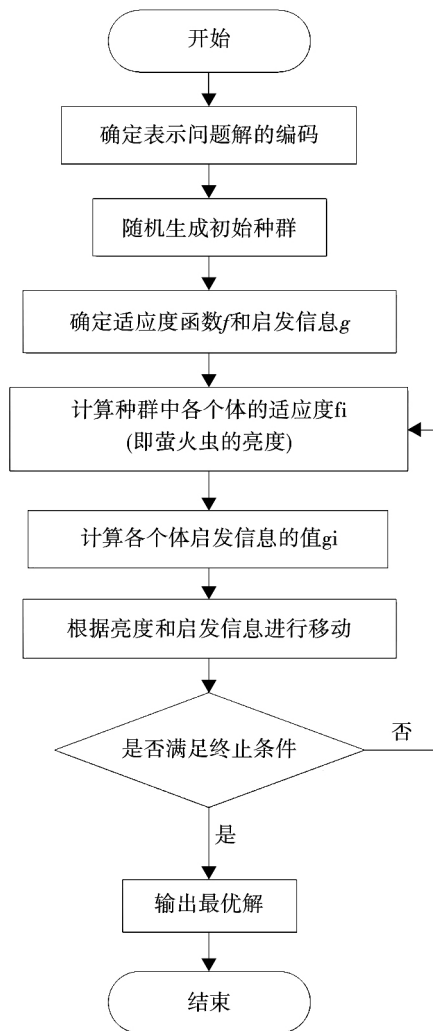


图1 基于启发信息的萤火虫算法流程图

Fig.1 Flow chart of heuristic information-based firefly algorithm

2.2 基于全局最优引导的萤火虫算法

在粒子群优化算法中,每个粒子的速度更新利用了其自身的历史最佳位置 p_{best} 和整个种群的历史最佳位置 g_{best} ^[8],也就是说,粒子群算法中的粒子是有记忆的,这大大加快了粒子群算法的收敛速度。基于以上理论,本文为萤火虫算法引入“全局最优”概念,认为历史中出现过的最亮萤火虫的位置对群体有着持续的吸引力,使萤火虫个体不但朝着当前最优移动,还向着当前全局最优移动一定距离,这就是基于全局最优的萤火虫算法(Firefly Algorithm Based on Global Optimization, FAGO)的基本思路。具体到算法本身,将原位置更新公式(2)改为

$$s_i^{k+1} = s_i^k + \beta_0 e^{-\gamma r_{ij}^2} (s_j^k - s_i^k) + w(g_{best}^k - s_i^k) + \alpha \varepsilon_i^{k+1} \quad (3)$$

式中: g_{best}^k 为 k 时刻记录的全局最优解的位置; w 为随迭代次数增加而减小的系数:

$$w = \frac{k_{max} - k}{k_{max}} \quad (4)$$

式中: k 为当前迭代次数; k_{max} 为最大迭代次数。

2.3 基于贝叶斯估计的萤火虫算法

萤火虫的移动行为由方向和在该方向上的位移组成。在标准萤火虫算法中,个体随机移动项的移动方向是完全随机的,这保证了算法跳出局部最优解的可能,然而,在既定情境下,最优解所处方向一般是固定的,完全随机的移动方式未考虑最优解的方向信息,相当程度上减缓了算法收敛速度。由此,本文考虑为随机移动的方向增加启发信息,保证算法跳出局部最优能力的同时加快收敛速度。

分析算法中萤火虫的移动过程,每只萤火虫个体进行移动后,可以统计出在每个方向进行移动的概率值,适应度的变化情况也可通过适应度函数计算得知,这样,得知了萤火虫在某个方向上移动的概率和在该方向上移动后适应度增加的概率,要求得适应度增加的前提下向该方向移动的概率,就是根据先验概率计算后验概率的问题,贝叶斯估计(Bayesian estimation)正是解决这一问题的方法。贝叶斯估计^[9]指出,若存在 $\cup_{i=1}^n A_i = \Omega$, $A_i A_j = \emptyset$, $P(A_i) > 0$,则称 $A_1, \dots, A_n$ 为完备事件组,可由贝叶斯公式计算出事件 B 发生的情况下 A_i 发生的概率:

$$P(A_i|B) = \frac{P(A_i) * P(B|A_i)}{\sum_{i=1}^n P(A_i) * P(B|A_i)} \quad (5)$$

扩展到萤火虫算法,记解空间的维度数为 m ,萤火虫总个数为 N ,在一次迭代中相对亮度大于萤火虫 j 的个体数为 n ,则萤火虫移动的方向集合 $c_1, c_2, \dots, c_{2m}$ 为一个完备事件组,对于萤火虫个体 j ,在方向 i 上,若有 p 只萤火虫亮度大于其自身,则萤火虫 j 会在方向 i 上移动 p 次,认为萤火虫数量足够多,可根据大数定理,用萤火虫向方向 i 移动的频率近似代替概率,有:

$$P(A = c_i) = \frac{\sum_{k=1}^{N-1} I(A_k = c_i)}{n} = \frac{\sum_{k=1}^p I(A_k = c_i)}{n} \quad (6)$$

$$i = 1, 2, \dots, 2m$$

$I(\cdot)$ 为指示函数,在 $\cdot$ 为真和假时分别取值1和0。

同理,记萤火虫适应度增加为事件 B ,若统计出萤火虫 j 向每个方向移动后适应度是否增加,就可

以计算出条件概率 $P(B|A=c_i)$ 的值:

$$P(B|A=c_i) = \frac{\sum_{k=1}^p I(B, A_k=c_i)}{\sum_{k=1}^K I(A_k=c_i)} \quad (7)$$

将式(6)和式(7)的结果代入式(5),就可以计算出萤火虫 j 在适应度增加的前提下向方向 i 移动的概率 $P(A=c_i|B)$,选择使得该概率最大的前 m 项所对应的方向进行移动,为萤火虫个体的移动带来后发信息。这就是基于贝叶斯估计的萤火虫算法(firefly algorithm based on Bayesian estimation, FABE)的核心思想。

另外,注意到式(5)中分母部分对于所有 i 均相同,是常数项,因此可以省略这一部分,以分子代替整体值,这样可以大大减小计算量,减少算法运行时间。

3 仿真实验

为说明本文提出改进算法的有效性,对 4 种典型的测试函数进行了测试^[14],如表 1 所示。

仿真实验在 Anaconda3 Spyder 平台下运行,运行环境为 Win10 系统下 Intel(R) Core(TM) i5-7300HQ 处理器。萤火虫个数 $N=25$,算法终止条件为满足最大迭代次数 $k_{\max}=200$ 。除 Schaffer 函数外, x 的维度均选取为 20 维,根据文献[1]给出的建议,算法参数选取为: $\alpha=0.6, \beta=1, \gamma$ 值由函数定义域确定, Griewank 选取为 0.001,其余函数中均取 0.01。对 4 个函数分别采用 FA、LFA、HEFA、FAGO、FABE 进行对比实验,为克服实验的随机性,每个函数分别以不同的初始随机值运行 10 次,每种方法采用相同初始随机值以进行对比。

表 1 测试函数
Tab. 1 Test Functions

测试函数	函数公式	取值范围	最优值点	最优值
Sphere	$f_1(x) = \sum_{i=1}^n x_i^2$	$[-100, 100]$	$[0, 0]^n$	0
Schaffer	$f_2(x) = \frac{\sin^2(x_1^2 - x_2^2) - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2} + 0.5$	$[-100, 100]$	$[0, 0]$	0
Griewank	$f_3(x) = \frac{1}{4000} \sum_{i=1}^n (x_i)^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]$	$[0, 0]^n$	0
Rosenbrock	$f_4(x) = \sum_{i=1}^n [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-100, 100]$	$[1, 1]^n$	0

各函数的具体情况分析如下。

3.1 Sphere 函数

Sphere 函数是简单的单峰函数,五种算法在同一维度下的最优值、平均值、耗时及对应方差如表 2

所示。

图 2 展示了最优值和平均值的变化曲线,横轴为迭代次数,纵轴为函数值的对数。

表 2 不同算法对 Sphere 函数的测试效果

Tab. 2 Test results of different algorithms used on Sphere function

算法	最优值		平均值		耗时	
	均值	方差	均值	方差	均值	方差
FA	2.253×10^{-7}	1.09×10^{-12}	0.002	7.03×10^{-5}	1.594	3.28×10^{-3}
LFA	1.692×10^{-8}	9.72×10^{-7}	0.265	1.63×10^{-3}	3.281	6.95×10^{-3}
HEFA	1.524×10^{-8}	5.10×10^{-13}	7.91×10^{-4}	6.25×10^{-5}	5.271	4.21×10^{-3}
FAGO	1.192×10^{-9}	3.15×10^{-16}	1.74×10^{-4}	1.35×10^{-5}	3.385	6.23×10^{-3}
FABE	3.793×10^{-9}	7.43×10^{-15}	4.51×10^{-4}	9.24×10^{-6}	2.930	1.21×10^{-3}

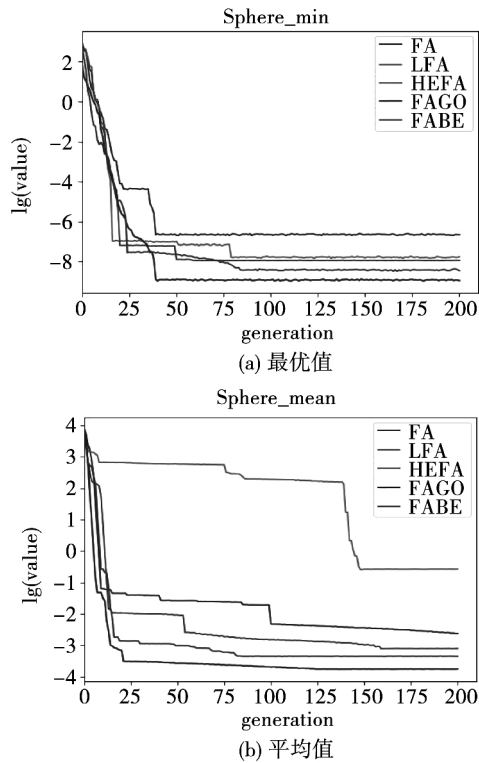


图2 Sphere函数的最优值、平均值变化情况

Fig. 2 Optimal value and average value of the Sphere function

从表2中可以看出:

- 1) 在最优值和平均值方差上, LFA 较大, FAGO 和 FABLE 相对较小, 说明 FAGO 和 FABLE 更为稳定;
- 2) 在最优解收敛精度上, FAGO、FABLE 比其他算法提升 1~2 个数量级, FAGO 略优于 FABLE;
- 3) 在平均解收敛精度上, FAGO 和 FABLE 优势较为明显, 比其他算法提升 2~3 个数量级; LFA 通过动态修改萤火虫算法的参数, 提高了算法最优值的收敛速度和精度, 但无法保证整体均值收敛速度和精度增加。

- 4) 在耗时上, 五种算法均处于同一数量级, FA 耗时最短, FABLE 次之, FAGO 和 LFA 耗时长度相近, HEFA 耗时最长。

从图2中可以看出, FA 最早收敛但过早陷入局部最优, FAGO 收敛速度较快且取得了较高的精度。

3.2 Schaffer 函数

Schaffer 函数是多峰函数, 局部极小值分布较为集中。测试结果如表3及图3所示, 为方便展示, 图3最优值变化曲线纵坐标为函数值的对数, 平均值变化曲线纵坐标为函数值本身。

表3 不同算法对 Schaffer 函数的测试效果

Tab.3 Test results of different algorithms used on Schaffer function

算法	最优值		平均值		耗时	
	均值	方差	均值	方差	均值	方差
FA	4.535×10^{-7}	5.25×10^{-7}	0.416	2.57×10^{-2}	1.613	3.45×10^{-3}
LFA	3.44×10^{-10}	2.35×10^{-8}	0.351	0.15	2.822	1.27×10^{-3}
HEFA	1.477×10^{-8}	4.32×10^{-21}	0.248	7.24×10^{-5}	5.381	4.23×10^{-3}
FAGO	8.39×10^{-12}	3.75×10^{-21}	1.73×10^{-5}	3.27×10^{-5}	2.649	2.27×10^{-3}
FABLE	4.33×10^{-11}	2.94×10^{-23}	1.64×10^{-5}	1.35×10^{-5}	3.036	5.23×10^{-3}

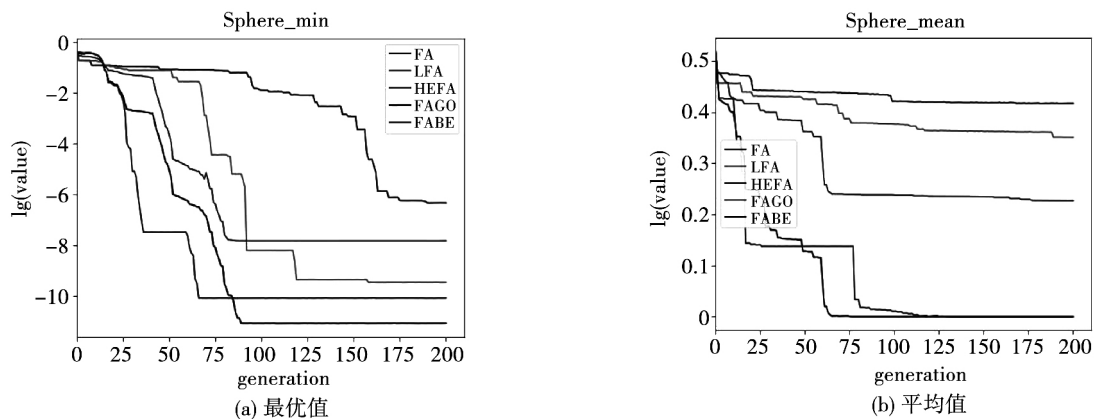


图3 Schaffer函数的最优值、平均值变化情况

Fig. 3 Optimal value and average value of the Schaffer function

从表 3 中可以看出:

1) 五种算法中,在最优值和平均值上,FABE 均最为稳定;

2) 在最优值收敛精度上,FAGO 最优值精度最高,FABE 次之,仅低于 FAGO 一个数量级,平均值的精度方面 FAGO 和 FABE 基本相同,均领先其他算法约 4 个数量级;

3) 在耗时上,FAGO、FABE、LFA 三者较为接近,且均短于 HEFA。图 3 中可以看出,FABE 最快收敛,FAGO 收敛速度略慢于 FABE。

3.3 Griewank 函数

Griewank 函数是多峰函数,局部极小值分布较为广泛。五种算法在 20 维下最优值、平均值、耗时的测试结果如表 4 和图 4 所示。

表 4 不同算法对 Griewank 函数的测试效果

Tab. 4 Test results of different algorithms used on Griewank function

算法	最优值		平均值		耗时	
	均值	方差	均值	方差	均值	方差
FA	0.260	3.57×10^{-4}	42.29	3.765	1.727	5.38×10^{-3}
LFA	2.859×10^{-5}	2.26×10^{-3}	8.69	0.791	3.270	6.25×10^{-3}
HEFA	3.424×10^{-4}	8.21×10^{-8}	13.25	3.271	6.105	5.98×10^{-3}
FAGO	2.366×10^{-4}	6.23×10^{-9}	2.43	7.37×10^{-4}	2.616	6.28×10^{-3}
FABE	1.073×10^{-5}	1.36×10^{-8}	1.30×10^{-4}	2.43×10^{-5}	3.518	4.26×10^{-3}

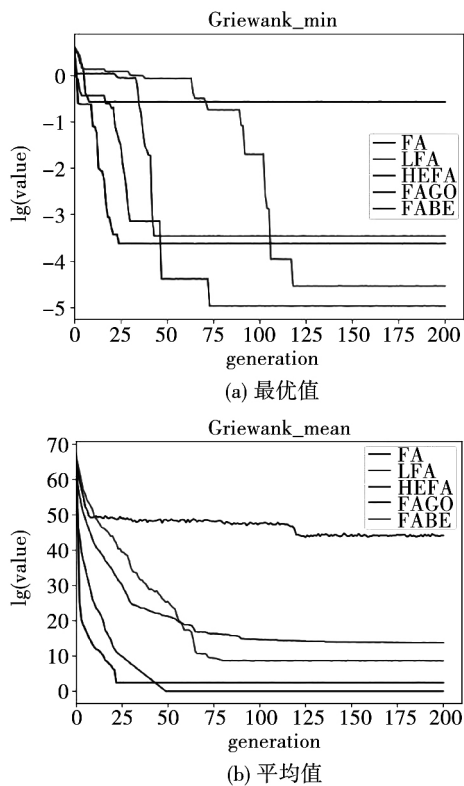


图 4 Griewank 函数的最优值、平均值变化情况

Fig. 4 Optimal value and average value of the Griewank function

从表 4 中可以看出:

1) 相比上文中提到的两个函数,5 种算法的方差均有所增加,说明测试 Griewank 函数时稳定性有所下降,相对而言,FAGO 和 FABE 稳定性较好;

2) 在最优值收敛精度上,FABE 最高,LFA 其次,FAGO 低于 FABE 近两个数量级且收敛速度快于 FABE,说明过早陷入局部最优,分析原因,FAGO 采用当前全局最优作为启发信息,容易被分布较为广泛且彼此相差不大的局部极小值迷惑,寻优能力相对较差;

3) 在耗时上,FABE 和 LFA 接近,FAGO 耗时较短,三者均明显快于 HEFA。

从图 4 中可以看出,FA 过早收敛于较差的局部最优解,HEFA、FAGO 收敛速度较快但精度较差,FABE 收敛精度大于 HEFA,但收敛速度略慢。

3.4 Rosenbrock 函数

Rosenbrock 函数是常见的复杂单峰函数,在其空间内走势平缓,全局最优点处于抛物线的顶点,所以很难收敛到全局最优。测试结果如表 5 及图 5 所示。从表 5 可以看出:

1) 对于 Rosenbrock 函数,几种算法稳定性均较差,FABE 和 HEFA 稳定性最好,FAGO 次之,LFA 稳定性较差;

2) 在收敛精度上,最优值中 FABE 和 FAGO 精度最高且结果相似,在平均值中 FABE 精度略高于 FAGO;

3) 在耗时上与 Griewank 函数中的结论类似,HEFA 耗时最长,FABE 和 LFA 其次,FAGO 耗时较短,FA 最快。

从图 5 可以看出,在收敛速度上,FAGO、FABE 收敛速度均快于其他算法。

表 5 不同算法对 Rosenbrock 函数的测试效果

Tab.5 Test results of different algorithms used on Rosenbrock function

算法	最优值		平均值		耗时	
	均值	方差	均值	方差	均值	方差
FA	8.913×10^{-4}	2.24×10^{-9}	119.82	38.33	1.405	3.44×10^{-3}
LFA	3.651×10^{-6}	1.68×10^{-6}	197.45	53.16	2.984	2.45×10^{-3}
HEFA	4.321×10^{-7}	8.01×10^{-10}	508.4	43.28	5.281	2.35×10^{-3}
FAGO	7.138×10^{-8}	2.23×10^{-10}	19.680	29.04	2.365	3.68×10^{-3}
FABE	2.318×10^{-7}	5.23×10^{-11}	5.281	16.18	3.091	5.27×10^{-3}

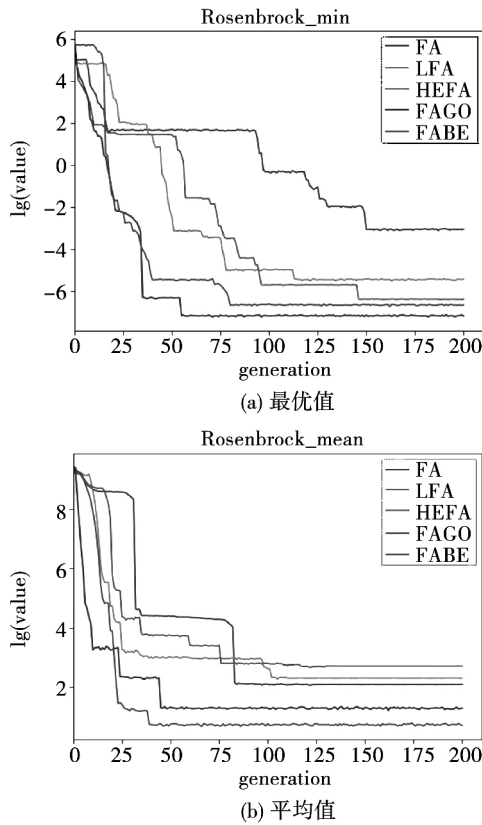


图 5 Rosenbrock 函数的最优值、平均值变化情况

Fig.5 Optimal value and average value of Rosenbrock function

4 结 论

本文为萤火虫算法引入了全局最优和贝叶斯估计两种启发信息,提高了算法最优值和平均值的收敛精度,在一定程度上加快了算法收敛速度。仿真实验结果表明:

LFA、HEFA、FAGO、FABE 最优值的求解精度均优于经典萤火虫算法,FAGO、FABE 精度普遍高于 LFA、HEFA,且具有良好的稳定性,LFA 方差较大。无论是单峰还是多峰函数,FAGO、FABE 平均值精度明显高于 LFA、HEFA 和 FA,且稳定性强于其他算法。

在完成固定迭代次数情况下,FAGO 与 LFA 耗时接近,FABE 耗时略长,但明显快于 HEFA。不考虑过早陷入局部最优的情况,FAGO、FABE 两种算法平均值、最优值达到各自的最小值所需迭代次数均小于其他算法,即收敛速度较快,在多数情况下,FAGO 比 FABE 更快。FAGO 更适合处理单峰问题和局部极小值分布较为集中的多峰问题,极小值分布相对分散的多峰问题更适合选用 FABE。

参 考 文 献:

- [1] YANG X S. Firefly Algorithms for Multimodal Optimization [C] // Berlin, Heidelberg, International symposium on stochastic algorithms. Springer, 2009: 169.
- [2] HUSSELMANN A V, HAWICK K A. Parallel Parametric Optimization with Firefly Algorithms on Graphical Processing Units [C] // Las Vegas, USA, CSREA (16 - 19 July 2012) . 2012: 77.
- [3] ATTIA K A M, NASSAR M W I, EL-ZEINY M B, et al. Firefly Algorithm Versus Genetic Algorithm as Powerful Variable Selection Tools and Their Effect on Different Multivariate Calibration Models in Spectroscopy: A Comparative Study [J] . Spectrochemical Acta Part A: Molecular and Biomolecular Spectroscopy, 2017, 170: 117.
- [4] 刘长平, 叶春明. 置换流水车间调度问题的萤火虫算法求解 [J] . 工业工程与管理, 2012(3) : 56 - 59 + 65.
- [5] YANG X S. Firefly Algorithm, Levy Flights and Global Optimization [C] // London, Springer, 2010: 209.
- [6] ABDULLAH A, DERIS S, MOHAMAD M S, et al. A New Hybrid Firefly Algorithm for Complex and Nonlinear Problem [C] // Berlin, Heidelberg, Springer, 2012: 673.
- [7] YANG X S, Nature-Inspired Optimization Algorithms [M] . Amsterdam, Elsevier Science Publishers, 2014.
- [8] CHENG S, LU H, LEI X, et al. A Quarter Century of Particle Swarm Optimization [J] . Complex & Intelligent Systems, 2018, 1(3) : 1.
- [9] AIMUTAIRI A O. Bayesian Estimation Using (Linex) for Generalized Power Function Distribution [J] . Lobachevskii Journal of Mathematics, 2018, 39(3) : 297.
- [10] Surjanovic, S. Bingham, D. Virtual Library of Simulation Experiments: Test Functions and Datasets [EB/OI] . <http://www.sfu.ca/~ssurjano> 2017 - 08 - 01 / 2018 - 09 - 01.

(编辑: 温泽宇)