

一种基于自适应搜索策略的改进萤火虫算法

于 干¹, 金丹丹²

¹阜阳师范大学信息工程学院, 安徽 阜阳

²阜阳师范大学计算机与信息工程学院, 安徽 阜阳

Email: yugan@fynu.edu.cn

收稿日期: 2020年9月7日; 录用日期: 2020年9月18日; 发布日期: 2020年9月25日

摘 要

萤火虫算法(FA)是一种基于群智能的优化技术, 它在很多优化问题上表现出较好的性能。然而, 它求解复杂优化问题时存在一些问题, 如收敛速度慢, 精度低。针对这些问题, 本文提出了一种新的萤火虫算法(取名AFA), 该方法使用了三种混合策略, 以获得好的优化性能。它首先使用一种自适应的参数方法来动态改变步长参数, 然后应用一种改进的搜索策略来消除吸引力, 于是, AFA不再包含光吸收系数和初始吸引力这2个参数; 再使用反向学习来提高解的精度。仿真结果表明, 本文提出的AFA算法优化结果优于MFA及PAFA算法。

关键词

萤火虫算法, 自适应搜索策略, 反向学习, 优化

An Improved Firefly Algorithm Based on Adaptive Search Strategies

Gan Yu¹, Dandan Jin²

¹College of Information Engineering, Fuyang Normal University, Fuyang Anhui

²School of Computer and Information Engineering, Fuyang Normal University, Fuyang Anhui

Email: yugan@fynu.edu.cn

Received: Sep. 7th, 2020; accepted: Sep. 18th, 2020; published: Sep. 25th, 2020

Abstract

Firefly algorithm (FA) is a recently proposed optimisation technique, based on swarm intelligence, which has shown good optimisation performance. However, FA suffers from slow convergence and low accuracy of solutions. To improve this case, this paper presents a new firefly algorithm (AFA)

by using three hybrid strategies to obtain a good optimisation performance. First, an adaptive parameter method is used to dynamically changing the step factor. Second, AFA uses a modified search strategy and eliminates the concept of attractiveness. So, HFA does not include two parameters, absorption coefficient and initial attractiveness. Third, a concept of opposition-based learning is used for improving the accuracy of the global best solution. Experiments on some benchmark problems show that AFA is superior to mimetic FA (MFA) and probabilistic attraction-based FA (PAFA).

Keywords

Firefly Algorithm, Adaptive Search Strategies, Opposition-Based Learning, Optimisation

Copyright © 2020 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 标准萤火虫算法

萤火虫算法是由剑桥大学的 Yang 教授提出的一种群智能算法, 它模拟了萤火虫的闪烁求偶行为。类似于粒子群算法, 萤火虫算法也是一种基于群体的随机搜索算法。群体中的每个个体(萤火虫)是对应问题的一个候选解。萤火虫算法的搜索依靠个体之间的吸引而产生移动来完成, 适应值较好(较亮)的萤火虫具有较大的吸引力, 使得适应值较差(较暗)的萤火虫向其移动。

个体(萤火虫)之间的吸引力定义为:

$$\beta(r_{ij}) = \beta\beta_0 e^{-\gamma r_{ij}^2} \quad (1)$$

其中 r_{ij} 表示萤火虫之间的距离, 它定义为:

$$r_{ij} = \|X_i - X_j\| = \sqrt{\sum_{d=1}^D (x_{id} - x_{jd})^2} \quad (2)$$

对于两个不同的萤火虫 X_i 和 X_j , 适应值较差(较暗)的萤火虫将向较好的萤火虫移动。假设 X_j 优于 X_i , 则 X_i 向 X_j 移动, 移动方式表示为:

$$x_{id}(t+1) = x_{id}(t) + \beta_0 e^{-\gamma r_{ij}^2} (x_{jd}(t) - x_{id}(t)) + \alpha \varepsilon_{id}(t) \quad (3)$$

标准 FA 中的参数都事先设定的, FA 的搜索能力受到其控制参数(如步长因子)的影响, 会导致算法收敛早熟; 标准 FA 因参数设置不当而导致算法无法收敛或收敛速度过慢。为了解决这两类问题, 使算法具有较好的优化性能, 需要对标准 FA 进行改进。

2. 改进算法的实现

萤火虫算法[1][2]是基于以下三个理想化的特征提出的: (1) 萤火虫不分性别, 即萤火虫之间的相互吸引只考虑个体发光的亮度; (2) 吸引力与发光亮度成正比, 与个体之间的距离成反比; (3) 萤火虫的亮度由待优化的目标函数值决定, 即 $I_i = f(x_i)$ 。FA 的关键思想是亮度小的萤火虫被亮度大的萤火虫吸引而向其移动, 并更新自身的位置。萤火虫的发光亮度取决于自身所处位置的目标值, 亮度越高所表示的目标值越好, 吸引其他萤火虫的能力也越强。若相邻的个体亮度相同, 萤火虫则随机移动。

在标准的 FA 中, 每个萤火虫都能被人群中其他明亮的萤火虫所吸引, 这种吸引力机制称为完全吸引模式[3][4], 在该模型下, 标准 FA 具有双环操作, 因此, 计算时间复杂度很高, 同时 FA 的搜索能力

受到其控制参数(如步长因子)的影响[5]。为了解决这个问题, 本文提出了一种自适应的参数策略来动态调整步长因子, 来消除吸引力。

1) 自适应搜索策略

一些文献指出, FA 的搜索能力受到其控制参数(如步长因子)的影响。为了克服这个问题, 本文使用了一种自适应的参数策略来动态调整步长因子 α 的值。

$$\alpha(t+1) = \left(\frac{1}{9000}\right)^{1/t} \alpha(t) \quad (4)$$

其中, t 指进化的代数。

基于 Rao 等人提出的 Jaya 算法, 本文针对萤火虫的移动方式进行了下面改进:

$$x_{id}(t+1) = x_{id}(t) + r_{1d}(x_{jd}(t) - |x_{id}(t)|) - r_{2d}(x_{wd}(t) - |x_{id}(t)|) \quad (5)$$

其中, r_{1d} 和 r_{2d} 是 2 个 [0,1] 之间的随机数, X_w 是当前种群中的最差解。与原有的萤火虫移动公式相比, 上述改进公式消除了吸引力的概念。因此, 我们新提出的算法不再包含初始吸引力和光吸收系数两个参数。

2) 反向学习过程

为了加快算法的收敛, 本文使用了反向学习策略(Opposition-Based Learning OBL)。对于当前种群中的最好解 X_{best} , 本文利用 OBL 产生一个反向解 X_{best}^* 。

$$X_{best}^*(t) = \bar{a} + \bar{b} - X_{best}(t) \quad (6)$$

其中, $[\bar{a}, \bar{b}]$ 表示当前种群的搜索区间。如果新产生的反向解 X_{best}^* 优于 X_{best} , 则使用 X_{best}^* 替换 X_{best} 。一些研究表明[6][7], 反向学习策略 OBL 有较高的概率找到的反向解比当前解更好。

3) 算法实现过程

Begin

 Initialise the population;

 while the stopping condition is satisfied do

 Update the step factor according to equation (4);

 for $i = 1$ to N do

 for $j = 1$ to N do

 if $f(X_j) < f(X_i)$ then

 Conduct the movement according to equation (5);

 Compute the fitness value of X_i ;

 end if

 end for

 Conduct the X_{best}^* according to equation (6);

 if X_{best}^* be better than X_{best}

$X_{best} = X_{best}^*$

 end if

 end for

 end while

End

3. 使用基准函数来测试 AFA 算法的性能

3.1. 测试函数

为了验证 AFA 算法性能, 本文使用了 7 个基准函数进行测试[8] [9] [10] [11], 所有测试函数均为最小值优化函数且全局最优解均为零。

测试函数 1:

$$F1(X) = \sum_{i=1}^D x_i^2, x_i \in [-100, 100] \quad (7)$$

测试函数 2:

$$F2(X) = \sum_{i=1}^D |x_i| + \prod_{i=1}^D x_i, x_i \in [-10, 10] \quad (8)$$

测试函数 3:

$$F3(X) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j \right)^2, x_i \in [-100, 100] \quad (9)$$

测试函数 4:

$$F4(X) = \max_i (|x_i|, 1 \leq i \leq D), x_i \in [-100, 100] \quad (10)$$

测试函数 5:

$$F5(X) = \sum_{i=1}^{D-1} \left(100(x_i - x_i^2)^2 + (x_i - 1)^2 \right), x_i \in [-30, 30] \quad (11)$$

测试函数 6:

$$F6(X) = \sum_{i=1}^D i x_i^4 + \text{rand}(0, 1), x_i \in [-1.28, 1.28] \quad (12)$$

测试函数 7:

$$F7(X) = 418.9829 \cdot D - \sum_{i=1}^{D-1} \left(x_i \sin \left(\sqrt{|x_i|} \right) \right), x_i \in [-500, 500] \quad (13)$$

3.2. 测试结果分析

测试实验中上述 7 个函数维度 D 分别设置为 10 和 30, 并将 AFA 的计算结果与 MFA 和 PAFA 进行比较, 结果显示, 本文提出的 HFA 优于其它两种改进的 FA 算法。所有算法的终止条件均设置为适应值函数最大个数(MaxFEs)。维度 $D = 10$ 时, MaxFEs 设置为 $1.0e+04$; 维度 $D = 30$ 时, MaxFEs 设置为 $5.0e+04$ 。对于两种不同的维度值, 算法的其他参数 α , β_0 , γ 分别设置为 0.2, 1.0, 及 $1/\Gamma 2$ 。

表 1 展现了当维度 $D = 10$ 时, 经过 30 代的演化计算三种算法所得到的最优函数值。从结果来看 AFA 函数结果除函数 F7 外均优于 MFA。求解函数 F7 问题时, 算法 MFA 和 PAFA 的优化结果优于 AFA 算法的结果。与 MFA 算法类似的是, AFA 算法求解 F1 至 F6 函数所表现出的其他性能(收敛速度、不易陷入局部寻优等)亦优于 PAFA 算法, 如对于所有的测试函数求解过程中, 当 AFA 算法已找到最优函数值时, 算法 MFA 和 PAFA 仍陷入局部寻优过程。正如文章开始提到的“标准 FA 的搜索能力受到其控制参数(如步长因子)的影响, 会导致算法收敛早熟”, 通过自适应策略, 本文提出的 AFA 算法不易陷入局部寻优的过程。

表 2 展现了当维度 $D = 30$ 时, 经过 30 代的演化计算三种算法所得到的最优函数值。如表所示, AFA 求解 F1, F2, F3, F5, F6, F7 函数展现了较好的算法性能。MFA 在求解 F4 函数时优于 AFA。相对于 PAFA, 求解 F1, F2, F5, F6 函数时 MFA 能够找到更加精确地解。求解 F3, F4, F7 函数时 PAFA 性能

优于 AFA。

Table 1. Computational results of each algorithm for $D = 10$

表 1. 维度 $D = 10$ 各算法寻优结果

测试函数	MFA	PAFA	AFA
	最优解	最优解	最优解
F1	2.08e+02	2.59e+01	6.19e-05
F2	2.83e+00	8.46e-01	2.68e-03
F3	4.26e+02	1.56e+01	4.52e-01
F4	8.65e+00	1.21e+00	1.32e-01
F5	2.04e-02	6.49e-03	4.16e-03
F6	1.87e+03	1.42e+03	1.08e+03
F7	3.01e+01	1.84e+01	3.12e+01

Table 2. Computational results of each algorithm for $D = 30$

表 2. 维度 $D = 30$ 各算法寻优结果

测试函数	MFA	PAFA	AFA
	最优解	最优解	最优解
F1	5.34e+02	8.82e-04	1.68e-09
F2	1.01e+01	1.42e-02	1.12e-05
F3	2.15e+03	8.08e-03	3.12e-01
F4	8.65e+00	1.29e-02	1.32e+01
F5	6.96e-02	1.62e-02	1.28e-02
F6	6.53e+03	6.12e+03	5.98e+03
F7	1.83e+02	2.78e+01	3.26e+01

图 1 展示的是当维度 $D = 10$, 算法 AFA、MFA 和 PAFA 求解函数 F1, F2 时的算法收敛过程。

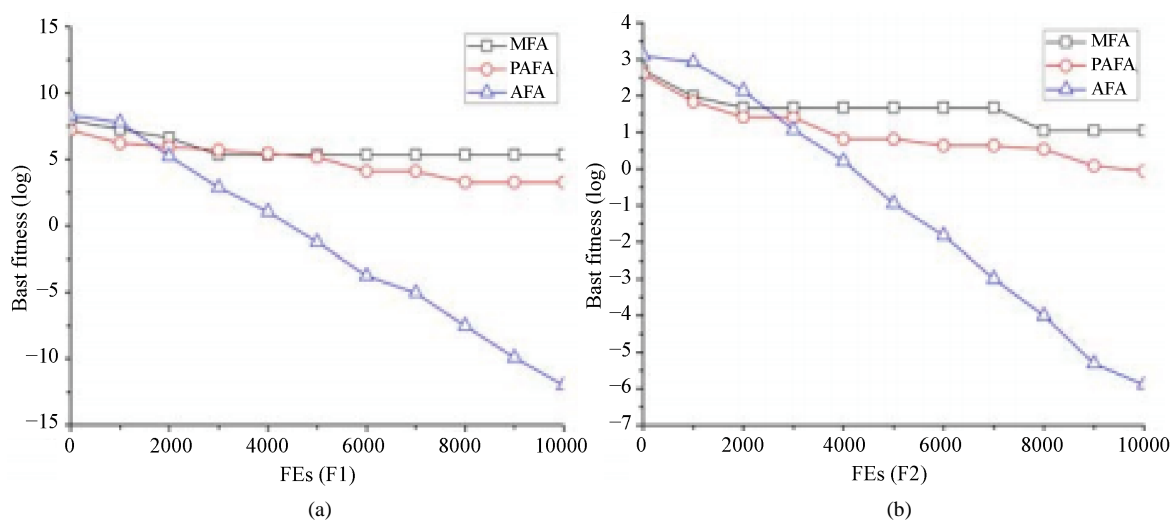


Figure 1. The search processes of AFA, MFA, and PAFA for $D = 10$, (a) function F1; (b) function F2

图 1. $D = 10$, AFA、MFA 及 PAFA 的收敛过程, (a) 功能 F1, (b) 功能 F2

图 2 展示的是当维度 $D = 30$, 算法 AFA、MFA 和 PAFA 求解函数 F1, F2 时的算法收敛过程。

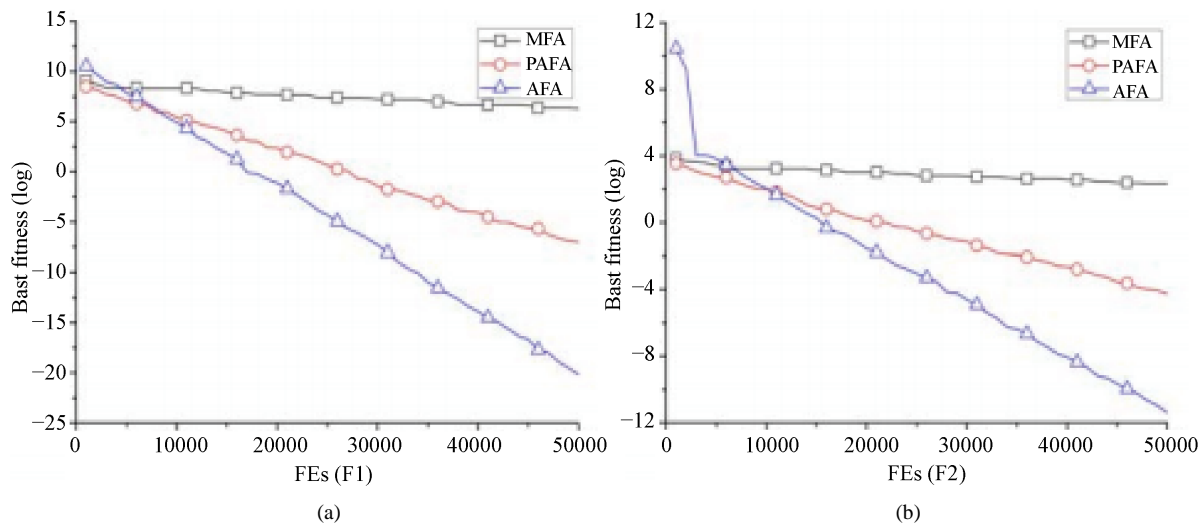


Figure 2. The search processes of AFA, MFA, and PAFA for $D = 30$, (a) function F1; (b) function F2

图 2. $D = 30$, AFA、MFA 及 PAFA 的收敛过程, (a) 功能 F1, (b) 功能 F2

从收敛曲线来看, 本文提出 AFA 的收敛速度也比 MFA 和 PAFA 快。通过标准反向学习过程, AFA 解决了“FA 因参数设置不当而导致算法无法收敛或收敛速度过慢”问题。

测试实验中, MaxFEs 的值设置的较小。当维度从 10 增加到 30 时, MaxFEs 的值从 $1.0e+04$ 增加到 $5.0e+04$, 算法 MFA 和 PAFA 的实验结果值得到了改进, 但如果将 MaxFEs 设置为一个更大的值时, AFA 可能会有更好的表现。通过实验可以得出, MaxFEs 的值对 AFA 的算法性能影响很大。

4. 结束语

本文提出了一种新的数值优化算法 AFA, 该算法基于标准的萤火虫算法 FA, 采用三种改进的措施, 包括自适应的参数方法来动态改变步长参数, 应用一种改进的搜索策略来消除吸引力; 再使用反向学习来提高解的精度。为了验证 AFA 的算法性能, 文章测试了 7 个标准的数值优化函数, 并分别测试了所测 7 个函数不同维度值。实验结果表明, AFA 算法在求解大部分函数时所表现出来的性能都优于算法 MFA 和 PAFA。

然而, 通过函数收敛图可以发现, AFA 在整个优化过程中, 该算法的收敛速度并没有明显优于 MFA 和 PAFA 算法, 如何解决这一问题, 使 AFA 算法性能更好, 这将是作者今后的研究工作之一。

基金项目

本文受到安徽省省级示范实验实训中心项目(2018sxx38); 2018 教育部智融兴教课题(2018A01010); 教育部 2019 年协同育人项目(201901258002); 安徽省教育厅高校优秀拔尖人才资助项目(gxyqZD2020054)等项目资助。

参考文献

- [1] Wang, H., Wang, W.J., Sun, H. and Rahnamayan, S. (2016) Firefly Algorithm with Random Attraction. *International Journal of Bio-Inspired Computation*, **8**, 33-41. <https://doi.org/10.1504/IJBIC.2016.074630>
- [2] Rao, R. (2016) Jaya: A Simple and New Optimization Algorithm for solving Constrained and Unconstrained Optimization Problems. *International Journal of Industrial Engineering Computations*, **7**, 19-34.

-
- <https://doi.org/10.5267/j.ijiec.2015.8.004>
- [3] Cai, X.J., Gao, X.Z. and Xue, Y. (2016) Improved Bat Algorithm with Optimal Forage Strategy and Random Disturbance Strategy. *International Journal of Bio-Inspired Computation*, **8**, 205-214. <https://doi.org/10.1504/IJBIC.2016.078666>
- [4] Wang, H., Sun, H., Li, C.H., Rahnamayan, S. and Pan, J.S. (2013) Diversity Enhanced Particle Swarm Optimization with Neighborhood Search. *Information Sciences*, **223**, 119-135. <https://doi.org/10.1016/j.ins.2012.10.012>
- [5] Fister Jr., I., Yang, X.S., Fister, I. and Brest, J. (2012) Memetic Firefly Algorithm for Combinatorial Optimization. *Bioinspired Optimization Methods and Their Applications*, **103**, 1-14.
- [6] Yu, G. (2016) An improved Firefly Algorithm Based on Probabilistic Attraction. *International Journal of Computing Science and Mathematics*, **7**, 530-536. <https://doi.org/10.1504/IJCSM.2016.081701>
- [7] Yu, G. and Feng, Y.Y. (2018) Improving Firefly Algorithm Using Hybrid Strategies. *International Journal of Computing Science and Mathematics*, **9**, 163-170. <https://doi.org/10.1504/IJCSM.2018.091749>
- [8] Zhang, M.Q., Wang, H., Cui, Z.H. and Chen, J.J. (2018) Hybrid Multi-Objective Cuckoo Search with Dynamical Local Search. *Memetic Computing*, **10**, 199-208. <https://doi.org/10.1007/s12293-017-0237-2>
- [9] Wang, F., Zhang, H., Li, K., Lin, Z., Yang, J. and Shen, X.-L. (2018) A Hybrid Particle Swarm Optimization Algorithm Using Adaptive Learning Strategy. *Information Sciences*, **436-437**, 162-177. <https://doi.org/10.1016/j.ins.2018.01.027>
- [10] Wang, F., Zhang, H., Li, Y., Zhao, Y. and Rao, Q. (2018) External Archive Matching Strategy for MOEA/D. *Soft Computing*, **22**, 7833-7846. <https://doi.org/10.1007/s00500-018-3499-9>
- [11] Cui, Z., Zhangm J., Wang, Y., *et al.* (2019) A Pigeon-Inspired Optimization Algorithm for Many-Objective Optimization Problems. *Science China Information Sciences*, **62**, Article No. 70212. <https://doi.org/10.1007/s11432-018-9729-5>