

深度学习萤火虫算法

赵 嘉^{1,2,3}, 谢智峰¹, 吕 莉^{1,2,3}, 王 晖^{1,2,3}, 孙 辉^{1,2,3}, 喻 祥^{1,3}

(1. 南昌工程学院信息工程学院, 江西南昌 330099;

2. 鄱阳湖流域水工程安全与资源高效利用国家地方联合工程实验室, 江西南昌 330099;

3. 江西省水信息协同感知与智能处理重点实验室, 江西南昌 330099)

摘 要: 为克服萤火虫算法全局寻优精度不高和过早收敛的缺点, 本文提出深度学习萤火虫算法. 算法采用随机吸引模型, 萤火虫随机选择一个粒子学习, 根据历史最优位置构建广义中心粒子, 对其进行一定次数的单维深度学习, 学习后的粒子引导种群进化. 实验发现, 深度学习策略及粒子深度学习次数对算法优化性能的改善起着重要作用. 12个基准测试函数的实验结果表明, 算法的综合寻优性能优于其它8种最近提出的萤火虫算法.

关键词: 全局寻优; 随机吸引模型; 广义中心粒子; 深度学习; 萤火虫算法

中图分类号: TP182; TP391 **文献标识码:** A **文章编号:** 0372-2112 (2018)11-2633-09

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2018.11.010

Firefly Algorithm with Deep Learning

ZHAO Jia^{1,2,3}, XIE Zhi-feng¹, LÜ Li^{1,2,3}, WANG Hui^{1,2,3}, SUN Hui^{1,2,3}, YU Xiang^{1,3}

(1. School of Information Engineering, Nanchang Institute of Technology, Nanchang, Jiangxi 330099, China;

2. National-Local Engineering Laboratory of Water Engineering Safety and Effective Utilization of Resources in Poyang Lake Area, Nanchang, Jiangxi 330099, China;

3. Jiangxi Province Key Laboratory of Water Information Cooperative Sensing and Intelligent Processing, Nanchang, Jiangxi 330099, China)

Abstract: In order to overcome low precision and premature convergence of firefly algorithm, this paper proposes a new method, called firefly algorithm with deep learning. First, firefly algorithm selects a particle to learn according to the random attraction model; second, the method constructs a general center particle based on the best historical position; third, the particle leads the evolution of the population after a certain times of one-dimensional deep learning. Experiments show that the deep learning strategy and the number of deep learning of particles play an important role in optimizing the performance of the algorithm. The experimental results of 12 benchmark functions demonstrate that the comprehensive optimization performance of the proposed algorithm outperforms eight other recently firefly algorithm variants.

Key words: global optimization; random attraction model; general center particle; deep learning; firefly algorithm

1 引言

复杂优化问题无处不在, 例如, 图像压缩^[1]、股票预测^[2]、配水网设计^[3]和动态资源分配^[4]等. 上述问题均具有多模、不连续及不可微等特性, 传统的隐枚举法 (Enumeration Method, MM)^[5]、分支定界法 (Branch and Bound Method, BBM)^[6]、动态规划 (Dynamic Programming Algorithm, DPA)^[7]等优化方法不能高效或者无法解决此类优化问题. 研究者迫切需要一种简单且高效的优化算法来弥补传统方法的不足. 随着群智能算法

(Swarm Intelligence Algorithm, SIA) 的出现和发展, 研究者尝试将诸如粒子群优化算法 (Particle Swarm Optimization, PSO)^[8-10]、布谷鸟搜索算法 (Cuckoo Search, CS)^[11,12]、人工蜂群算法 (Artificial Bee Colony, ABC)^[13]及萤火虫算法 (Firefly Algorithm, FA)^[14,15]等群智能算法应用于上述复杂优化问题.

受自然界萤火虫生物学特性影响, Yang^[14]于2008年提出萤火虫算法. 算法采用全吸引模型, 萤火虫向自身更亮的萤火虫移动, 随着种群进化, 各萤火虫逐渐聚集于最亮的萤火虫附近, 完成寻优任务. 算法的全吸

引模型,使得萤火虫在移动过程极易出现振荡,浪费计算资源、丧失种群多样性,致使种群过早收敛,陷入局部最优。

深度学习^[16]是机器学习领域中一个新兴的研究方向,它通过模拟人脑结构,对数据进行智能分析与处理。即从原始数据开始将每层表示(特征)逐层转换为更高层更抽象的表示,从而发现数据中的各类错综复杂的结构^[17]。

为改善算法全局寻优能力和防止过早收敛,本文提出深度学习萤火虫算法(Firefly Algorithm with Deep Learning, DLFA)。算法使用随机吸引模型替代全吸引模型对种群进行进化;构造广义中心粒子,对其进行一定次数单维深度学习;用学习后的广义中心粒子引导种群进化。实验结果表明,DLFA 算法具有较好的收敛速度和寻优精度。

2 萤火虫算法

萤火虫算法是一种基于群智能的随机搜索技术,其思想源于萤火虫向比自己更亮的萤火虫移动这一生物学特性。在搜索空间,萤火虫的位置表示优化问题的解,亮度对应优化问题的适应值。萤火虫不断向更亮的萤火虫移动,直至达到预设的算法终止条件,完成寻优任务。算法的数学描述如下:

设萤火虫数量 N , 维度 D , 第 i 只萤火虫和第 j 只萤火虫的位置分别表示为 $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$, $i = 1, 2, \dots, N$ 和 $x_j = (x_{j1}, x_{j2}, \dots, x_{jD})$, $j = 1, 2, \dots, N$ 。萤火虫 i 和萤火虫 j 的距离 r_{ij} 计算如下:

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

其中, x_{id} 和 x_{jd} 分别表示第 i 只和第 j 只萤火虫的第 d 维位置。

萤火虫的亮度及吸引度的计算公式如下:

$$I = I_0 e^{-\gamma r_{ij}^2} \quad (2)$$

$$\beta = \beta_0 e^{-\gamma r_{ij}^2} \quad (3)$$

其中, I_0 和 β_0 分别为萤火虫初始亮度和初始吸引度, γ 为光吸收系数。

萤火虫 i 向萤火虫 j 移动的更新公式如下:

$$x_{id}(t+1) = x_{id}(t) + \beta(x_{jd}(t) - x_{id}(t)) + \alpha_i(t)\epsilon \quad (4)$$

其中, $x_{id}(t)$ 和 $x_{jd}(t)$ 分别为萤火虫 i 和 j 的第 t 代第 d 维位置, $\alpha_i(t)$ 表示萤火虫 i 的第 t 代步长因子, ϵ 服从均匀分布, 取值范围为 $[-0.5, 0.5]$ 。

3 深度学习萤火虫算法

FA 采用全吸引模型, 每只萤火虫向所有优于自身的萤火虫移动, 其移动过程易出现振荡, 随着种群进化, 萤火虫逐渐靠近, 成为相似的个体, 致使种群出现停滞,

过早收敛。为保持萤火虫的多样性和防止种群过早收敛, 本文结合随机吸引模型、广义中心粒子和深度学习 3 种策略进行算法寻优。

3.1 随机吸引模型

FA 采用全吸引模型(Full Attraction Model, FAM)进行种群进化, 即每只萤火虫向比自己更亮的萤火虫移动。假定种群大小为 N , 采用全吸引模型的进化方式, 最亮的萤火虫需要移动 0 次, 最暗的萤火虫需要移动 $N-1$ 次, 即种群进化一代需要移动的最大总次数是 $M^{full} = N * (N-1)/2$ 。虽然更多的移动次数可以给萤火虫更多机会进行搜索, 但是这种进化方式会使萤火虫搜索产生振荡, 无法进行深度搜索, 浪费计算资源。为减少算法振荡和节约计算资源, 本文使用随机吸引模型(Random Attraction Model, RAC)代替全吸引模型, 即每只萤火虫随机选择一个萤火虫移动。按照随机吸引模型的进化方式, 种群进化一代需要移动的最大总次数是 $M^{rand} = N-1$ 。当 $N \geq 1$ 时, $M^{rand} \leq M^{full}$, 随着种群粒子个数 N 的增加, 随机吸引模型相比全吸引模型, 算法消耗更少的计算资源, 且不易出现振荡。

3.2 广义中心粒子

FA 算法进化的本质是不断地向更亮的萤火虫靠近, 即向“榜样”学习。因此, 提高 FA 算法寻优性能的重要途径之一是选择优异的“榜样”。Liu 等^[18]于 2006 年提出中心粒子群优化算法(Center particle swarm optimization, CenterPSO), 并通过实验证明了中心粒子比 gBest (全局最优粒子) 更接近最优值。汤可宗等^[19]在文献[18]的基础之上, 构造广义中心粒子(General Center Particle, GCP)和狭义中心粒子(Special Center Particle, SCP), 提出双中心粒子群优化算法(Double Center Particle Swarm Optimization Algorithm, DCPSO)。其中, 狭义中心粒子是文献[18]提到的中心粒子, 它的位置由公式(5)决定; 广义中心粒子由公式(6)决定, 文献[19]实验证明了广义中心粒子比狭义中心粒子更接近理论最优解。

$$x_d^{SCP} = \frac{1}{N} \sum_{i=1}^D x_{id} \quad (5)$$

$$x_d^{GCP} = \frac{1}{N} \sum_{i=1}^D x_{id}^{pbest} \quad (6)$$

其中, x_d^{SCP} 、 x_d^{GCP} 分别表示狭义中心和广义中心粒子的第 d 维位置, x_{id}^{pbest} 表示第 i 个粒子的历史最优位置的第 d 维位置。

3.3 深度学习

深度学习以深层神经网络的方式, 对原始数据进行逐层特征提取, 构建起低级特征到高级语义之间复杂的映射关系^[20]。每个神经元视为一个计算单元, 各神经元通过权重参数建立联系, 逐层协作完成数据处理。

本文将深度学习的思想应用于广义中心粒子寻优. 广义中心粒子视为第一个神经元输入数据, 记为 $x_d^{\text{GCP}}(0)$, 第一个神经元根据公式(7)单维优化 $x_d^{\text{GCP}}(0)$ 得到第二个神经元的输入数据 $x_d^{\text{GCP}}(1)$, 第二个神经元根据公式(7)单维优化 $x_d^{\text{GCP}}(1)$ 得到第三个神经元的输入数据 $x_d^{\text{GCP}}(2)$, 以此类推, 逐层单维优化 DL_count 次, 得到广义中心粒子最终优化结果 $x_d^{\text{GCP}}(DL_count)$.

$x_d^{\text{GCP}}(t+1) = x_d^{\text{GCP}}(t) + \text{cauchy}() (x_{rd}(t) - x_d^{\text{GCP}}(t))$ (7) 其中, $x_{rd}(t)$ 为随机选择的第 r 个萤火虫第 t 代第 d 维位置, $\text{cauchy}()$ 表示比例因子为 1 的柯西分布函数产生的随机数.

广义中心粒子深度学习完成后, 将萤火虫种群各粒子与完成深度学习的广义中心粒子进行比较, 若当前萤火虫适应值差, 则萤火虫利用式(8)更新位置, 反之, 利用式(9)进行柯西突变.

$$x_{id}(t+1) = x_{id}(t) + \beta_0 e^{-\gamma r_{i\text{GCP}}} (x_d^{\text{GCP}}(t) - x_{id}(t))$$
 (8)

$$x_{id}(t+1) = x_d^{\text{gbest}}(t) + \text{cauchy}()$$
 (9)

其中, 公式(8)中的 $r_{i\text{GCP}}$ 表示萤火虫 i 和广义中心粒子之间的距离, 公式(9)中的 $x_d^{\text{gbest}}(t)$ 表示全局最优粒子的第 t 代第 d 维位置.

3.4 算法流程

DLFA 算法流程如下所示:

(1) 初始化萤火虫种群, 设置相关参数, 如: α, β_0, γ 等.

(2) 计算和评价萤火虫亮度(适应值).

(3) 按照随机吸引模型学习规则, 利用式(4)对萤火虫位置进行更新.

(4) 利用式(6)构建广义中心粒子, 并利用式(7)对广义中心粒子进行 DL_count 次深度学习.

(5) 将萤火虫与完成深度学习的广义中心粒子进行比较, 若当前萤火虫适应值差, 则萤火虫利用式(8)更新位置, 反之, 利用式(9)进行柯西突变.

(6) 检验是否满足终止条件, 若满足, 则停止迭代, 输出全局最优位置及其对应的适应值, 否则转到(3).

综上所述, 表 1 给出 DLFA 算法的伪代码. 其中, $pbest$ 和 $gbest$ 分别表示萤火虫历史最优位置和全局最优位置; FES 表示种群评估次数, MAX_FES 表示种群最大评估次数; DL_count 表示广义中心粒子深度学习的次数.

表 1 DLFA 算法伪代码

```

1 Initialize a population of fireflies randomly
2 Calculate the fitness values of each firefly and record  $pBest$  and  $gBest$ 
3  $FES = N$ 
4 While ( $FES < MAX\_FES$ )
5   for  $i = 1$  to  $N$ 
6     Select a firefly  $j$  from the current swarm randomly and  $i \neq j$ 

```

```

7   if  $f(x_i) > f(x_j)$ 
8     Move firefly  $i$  towards firefly  $j$  according to Eq. (4)
9     Calculate the fitness values of the new firefly  $i$ 
10     $FES = FES + 1$ 
11  end if
12 end for
13 Construct the  $x_d^{\text{GCP}}$  according to Eq. (6)
14 Updated the  $x_d^{\text{GCP}}$  by Eq. (7) with  $DL\_count$  times
15  $FES = FES + DL\_count$ 
16 for  $i = 1$  to  $N$ 
17   if  $f(x^i) > f(x^{\text{GCP}})$ 
18     Move firefly  $i$  towards the particle GCP according to Eq. (8)
19     Calculate the fitness values of the new firefly  $i$ 
20      $FES = FES + 1$ 
21   else
22     Mutate  $x_i$  according to Eq. (9)
23     Calculate the fitness values of the new firefly  $i$ 
24      $FES = FES + 1$ 
25   end if
26 end for
27  $Gen++$ 
27 end While

```

4 实验

4.1 测试函数

本文使用 12 个基准测试函数验证 DLFA 算法优化性能. 函数 $f_1 - f_7$ 是单峰函数, 用来检验算法的收敛速度和局部搜索能力; $f_8 - f_{12}$ 是多峰函数, 用来检验算法的寻优精度和逃离局部最优能力. 所有测试函数维度 D 为 30, 具体表述见表 2.

4.2 深度学习次数分析

种群进化一代, 广义中心粒子深度学习 DL_count 次. 深度学习策略中的深度学习次数对算法性能具有较大影响. 在极端情况下, 当 $DL_count = MAX_FES$ 时, 只有广义中心粒子参与进化; 当 $DL_count = 0$ 时, 广义中心粒子不进行深度学习, 算法不具备深度学习能力. 为寻找最佳深度学习次数 DL_count , 本文将 DL_count 取值范围设置为 $[0, 5000]$, 实验间距取值 100, 对所有 12 个测试函数进行实验分析, 记录每次深度学习搜索得到的最优值, 如图 1 所示. 通过分析图 1, 对于函数 $f_1 - f_2$ 和函数 f_9 , 其收敛曲线在区间 $[0, 200]$ 内垂直下降, 完成寻优任务; 对于函数 $f_5 - f_8$ 和 $f_{10} - f_{12}$, 其收敛曲线在整个区间趋于直线, 随 DL_count 值递增波动不明显; 对于函数 $f_3 - f_4$, 其收敛曲线随 DL_count 递增呈“凹”字形; 函数 f_3 和 f_4 分别在区间 $[1000, 2000]$ 和 $[500, 1500]$ 内取值, 算法优化性能最佳. 通过实验, 综合各函数结果, 本文将 DL_count 设置为 1200.

表 2 12 个基准测试函数

| No. | Name | Function | Range | Optimum |
|----------|---------------|---|-----------------|---------------|
| f_1 | Sphere | $f_1(x) = \sum_{i=1}^D x_i^2$ | $[-100, 100]$ | 0 |
| f_2 | Schwefel 2.22 | $f_2(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i$ | $[-10, 10]$ | 0 |
| f_3 | Schwefel 1.2 | $f_3(x) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j \right)^2$ | $[-100, 100]$ | 0 |
| f_4 | Schwefel 2.21 | $f_4(x) = \max\{ x_i , 1 \leq i \leq D\}$ | $[-100, 100]$ | 0 |
| f_5 | Rosebrock | $f_5(x) = \sum_{i=1}^D [100(x_{i+1} - x_i^2)^2 + (1 - x_i^2)^2]$ | $[-30, 30]$ | 0 |
| f_6 | Step | $f_6(x) = \sum_{i=1}^D \lfloor x_i + 0.5 \rfloor$ | $[-100, 100]$ | 0 |
| f_7 | Quartic | $f_7(x) = \sum_{i=1}^D ix_i^4 + \text{random}[0, 1)$ | $[-1.28, 1.28]$ | 0 |
| f_8 | Schwefel 2.26 | $f_8(x) = \sum_{i=1}^D -x_i \sin \sqrt{ x_i }$ | $[-500, 500]$ | $-418.98 * D$ |
| f_9 | Rastrigin | $f_9(x) = \sum_{i=1}^D [x_i^2 - 10 \cos 2\pi x_i + 10]$ | $[-5.12, 5.12]$ | 0 |
| f_{10} | Ackley | $f_{10}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos 2\pi x_i\right) + 20 + e$ | $[-32, 32]$ | 0 |
| f_{11} | Griewank | $f_{11}(x) = \frac{1}{4000} \sum_{i=1}^D (x_i)^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$ | $[-600, 600]$ | 0 |
| f_{12} | Penalized | $f_{12}(x) = \frac{\pi}{D} \left\{ \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + \sin(\pi y_{i+1})] + (y_D - 1)^2 \right.$ $\left. + (10 \sin^2(\pi y_1)) \right\} + \sum_{i=1}^D u(x_i, 10, 100, 4), y_i = 1 + \frac{x_i + 1}{4}$ $u(x_i, a, k, m) = \begin{cases} u(x_i, a, k, m), & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$ | $[-50, 50]$ | 0 |

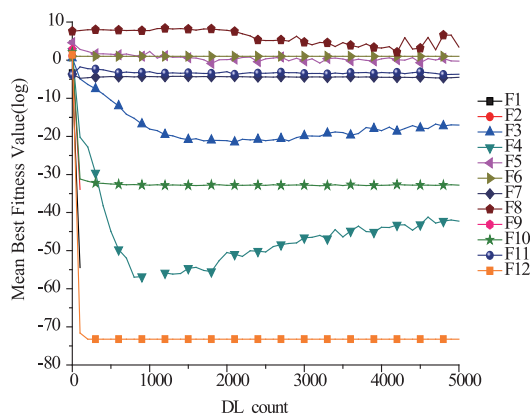


图1 不同深度学习次数下所有测试函数的最优结果对数值

4.3 参数设置

本文将 DLFA 算法与其它 8 种 FA 算法进行比较,各

算法的详细信息及参数设置见表 3. 为公平比较,所有算法的种群个数设置为 20,最大评估次数设置为 $5 * 10^5$. 为避免优化结果的随机差异性,每个测试函数,各算法独立运行 30 次,实验结果取 30 次的平均值.

4.4 实验结果

表 4 给出各种算法的寻优结果及 $w/t/l$ 结果. $w/t/l$ 表示 DLFA 算法与 8 种比较算法相比在 w 函数上胜出, t 个函数上相当, l 个函数上落后. 分析表 4, DLFA 算法的寻优结果在 12 个测试函数上均优于 FA 2008、WSS-FA 和 VSSFA 算法. 与 MFA 和 CFA 算法相比, DLFA 算法的寻优结果在 11 个测试函数上优,仅在 f_6 上与它们相等. 除多峰函数 f_{11} , DLFA 算法在剩余 11 个测试函数上寻优结果优于 FA 2010 算法. 与 RaFA 算法相比, DLFA 算法在 9 个测试函数上优, f_6 上相等,仅 f_8 和 f_{11} 上

表 3 各比较算法

| Algorithm | Parameter setting | Reference |
|--|--|-------------------------------------|
| FA 2008 | $\alpha = 0.2, \beta_0 = 1.0, \gamma = 1.0$ | Yang 2008 ^[14] |
| FA 2010 | $\alpha = 0.2, \beta_0 = 1.0$ | Yang 2010 ^[21] |
| Memetic FA (MFA) | - | Fister et al. 2012 ^[22] |
| FA with chaos (CFA) | $\alpha_{\min} = 0.04, \beta_0 = 1.0, \gamma = 1.0$ | Gandomi et al. 2013 ^[23] |
| Wise step strategy FA (WSSFA) | $\beta_0 = 1.0, \gamma = 1.0$ | Yu et al. 2014 ^[24] |
| Variable step size FA (VSSFA) | $\alpha = 0.2$ | Yu et al. 2015 ^[25] |
| FA with random attraction (RaFA) | $\alpha = 0.2, \beta_0 = 1.0, \gamma = 1/\Gamma^m (m = 2)$ | Wang et al. 2016 ^[26] |
| FA with adaptive control parameters (ApFA) | $\alpha = 0.2, \beta_0 = 1.0, \gamma = 1/\Gamma^m (m = 2)$ | Wang et al. 2017 ^[27] |
| DLFA | $\alpha = 0.2, \beta_0 = 1.0, \gamma = 1/\Gamma^m (m = 2)$ | - |

效果不佳. 除多峰函数 f_{11} , DLFA 算法在 10 个测试函数上寻优结果优于 ApFA, 1 个相等. 单峰函数 f_6 , MFA、CFA、RaFA、ApFA 和 DLFA 算法均寻优到了最优值. DLFA 算法在 4 个单峰函数 (f_1, f_2, f_4 和 f_6) 和 1 个多峰函数 (f_9) 均寻优到最优值. 多峰函数 f_{12} , DLFA 算法的寻优结果最佳.

为更加直观的观察算法变化趋势, 图 2 展示了 12 个测试函数的各算法收敛曲线图. 通过对图 2 的分析,

除函数 f_4, f_7 和 f_8 , DLFA 算法的收敛速度在整个寻优过程均快于其它 8 个函数. 特别地, 在函数 f_1, f_2, f_6 和 f_9 上, 在搜索前期 DLFA 算法快速地完成了寻优任务. 对于函数 f_6 , MFA、CFA、RaFA、ApFA 和 DLFA 算法均提前收敛, 完成寻优任务. 多峰函数 f_8 , DLFA 算法在搜索中期出现了一个极大的“跳跃”, 该函数存在极多局部最优解, 说明 DLFA 算法具备一定的逃离局部最优的能力.

表 4 9 个算法的平均最优适应值

| Function | FA 2008 Mean | WSSFA Mean | VSSFA Mean | FA 2010 Mean | MFA Mean | CFA Mean | RaFA Mean | ApFA Mean | DLFA Mean |
|----------|--------------|------------|------------|--------------|-----------|-----------|-----------|-----------|-----------|
| f_1 | 6.67E+04 | 6.34E+04 | 5.84E+04 | 5.14E-02 | 1.56E-05 | 3.27E-06 | 5.36E-184 | 2.02E-44 | 0.00E+00 |
| f_2 | 5.19E+02 | 1.35E+02 | 1.13E+02 | 1.07E+00 | 1.85E-03 | 8.06E-04 | 8.76E-05 | 1.83E-12 | 0.00E+00 |
| f_3 | 2.43E+05 | 1.10E+05 | 1.16E+05 | 1.26E-01 | 5.89E-05 | 1.24E-05 | 4.91E+02 | 1.01E+01 | 1.58E-09 |
| f_4 | 8.35E+01 | 7.59E+01 | 8.18E+01 | 9.98E-02 | 1.73E-03 | 8.98E-04 | 2.43E+00 | 1.30E-07 | 0.00E+00 |
| f_5 | 2.69E+08 | 2.49E+08 | 2.16E+08 | 3.41E+01 | 2.29E+01 | 2.06E+01 | 2.92E+01 | 2.81E+01 | 1.99E+00 |
| f_6 | 7.69E+04 | 6.18E+04 | 5.48E+04 | 5.24E+03 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 | 0.00E+00 |
| f_7 | 5.16E+01 | 3.24E-01 | 4.43E+01 | 7.55E-02 | 1.30E-01 | 9.03E-02 | 5.47E-02 | 2.76E-03 | 1.26E-02 |
| f_8 | -1.56E+03 | -2.01E+03 | -1.85E+03 | -3.41E+03 | -7.63E+03 | -8.21E+03 | -1.21E+04 | -6.25E+03 | -8.86E+03 |
| f_9 | 3.33E+02 | 3.61E+02 | 3.12E+02 | 4.95E+01 | 6.47E+01 | 5.27E+01 | 2.69E+01 | 1.21E+01 | 0.00E+00 |
| f_{10} | 2.03E+01 | 2.05E+01 | 2.03E+01 | 1.21E+01 | 4.23E-04 | 4.02E-04 | 3.61E-14 | 2.55E-14 | 5.68E-15 |
| f_{11} | 6.54E+02 | 6.09E+02 | 5.47E+02 | 2.13E-02 | 9.86E-03 | 7.91E-06 | 0.00E+00 | 3.33E-16 | 2.99E-02 |
| f_{12} | 7.16E+08 | 6.18E+08 | 3.99E+08 | 6.24E+00 | 5.04E-08 | 8.28E-09 | 4.50E-05 | 1.23E-16 | 1.57E-32 |
| $w/t/l$ | 12/0/0 | 12/0/0 | 12/0/0 | 11/0/1 | 11/1/0 | 11/1/0 | 9/1/2 | 10/1/1 | -/-/- |

为综合评估各算法的优化性能, 对各算法进行 Friedman 检测. 表 5 给出了 9 种算法的秩均值. 秩均值越小, 表示算法综合优化性能最佳. 通过表 5 可知, CFA 和 RaFA 算法的秩均值相同, 说明这 2 种算法改进效果相同; DLFA 算法的秩均值最小, 说明 DLFA 算法优化效果显著, 综合优化性能最佳.

4.5 策略分析

DLFA 算法在 FA 2010 算法的基础之上采用随机吸引模型、广义中心粒子和深度学习 3 种策略进行算法优化性能提升. 为研究各策略对算法优化性能的影响程度, 进行不同策略组合, 形成相应的改进算法, 具体信息见表 6.

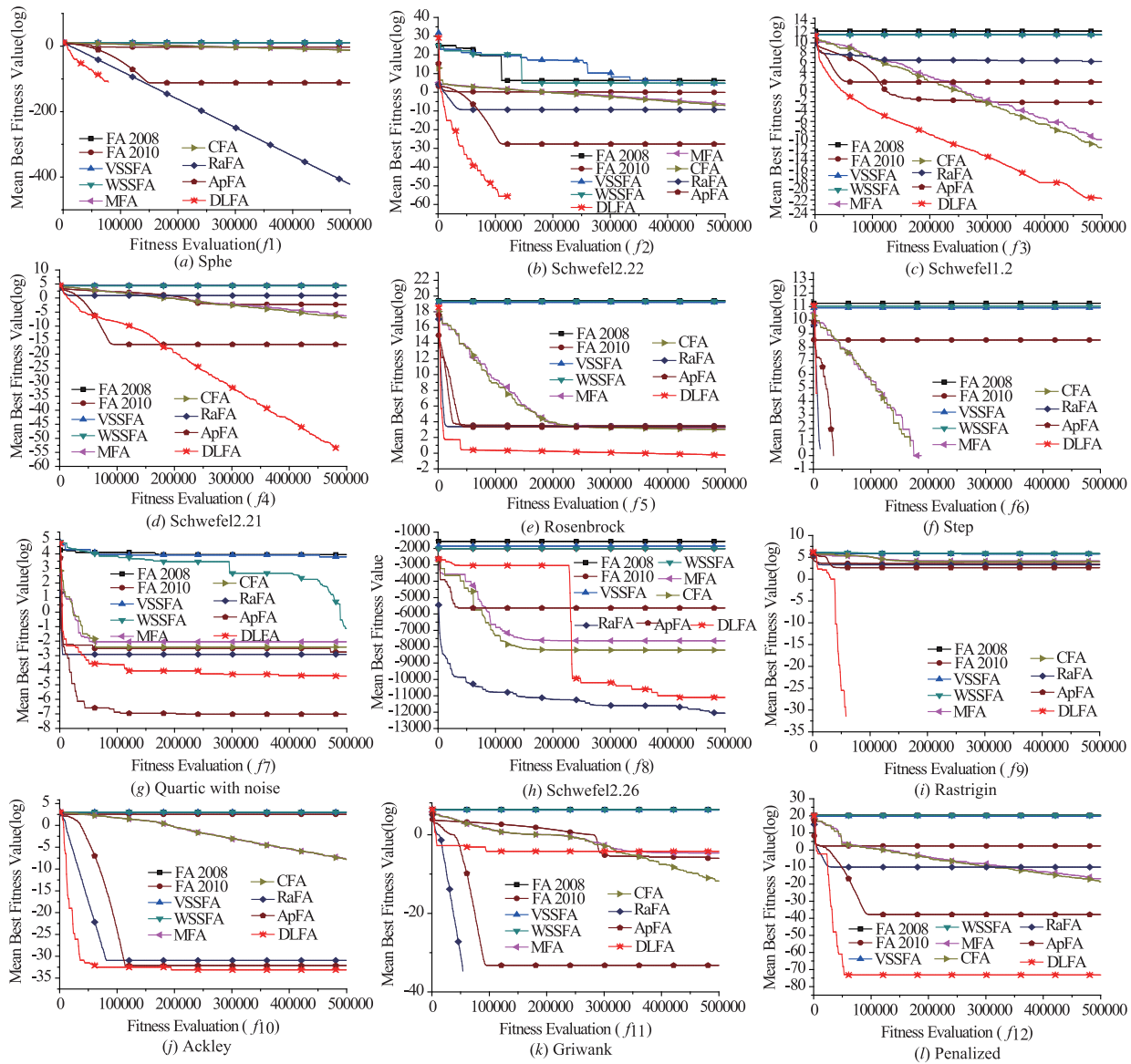


图2 9个算法收敛曲线图

表5 9种算法秩均值

| Algorithm | Rankings |
|-----------|----------|
| DLFA | 1.75 |
| ApFA | 2.75 |
| RaFA | 3.42 |
| CFA | 3.42 |
| MFA | 4.42 |
| FA 2010 | 5.25 |
| VSSFA | 7.38 |
| WSSFA | 7.83 |
| FA 2008 | 8.79 |

为检验组合算法的优化性能,使用4.1节的12个函数进行测试,实验参数设置同4.3节,实验结果见表7.通过表7可知,FA 2010 + RAC算法在9个函数上寻优结果优于FA 2010,在函数 f_{11} 上寻优结果最佳.FA 2010 + RAC + GCP算法在7个函数上优于FA 2010 +

RAC,特别是在函数 f_6 上搜索到最优值,在函数 f_8 上寻优结果最佳.DLFA算法在FA 2010 + RAC + GCP算法基础之上,加入深度学习策略,寻优结果显著改善,11个函数寻优结果优于FA 2010和FA 2010 + RAC算法,9个测试函数寻优结果优于FA 2010 + RA + GCP算法,说明深度学习策略对DLFA算法优化性能提升起着关键作用.

表6 不同组合策略萤火虫算法

| Combination strategies | Combination algorithms |
|--|------------------------|
| FA 2010 | FA 2010 |
| FA 2010 + Radom attraction model | FA 2010 + RAC |
| FA 2010 + Radom attraction model + General center particle | FA 2010 + RAC + GCP |
| FA 2010 + Radom attraction model + General center particle + Deep learning | DLFA |

表 7 不同组合策略萤火虫算法的平均最优适应值

| Function | FA 2010 Mean | FA 2010 + RAC Mean | FA 2010 + RAC + GCP Mean | DLFA Mean |
|----------|--------------|--------------------|--------------------------|-----------------|
| f_1 | 5.14E-02 | 5.23E-02 | 3.02E-02 | 0.00E+00 |
| f_2 | 1.07E+00 | 1.03E+00 | 2.94E+00 | 0.00E+00 |
| f_3 | 1.26E-01 | 1.05E-01 | 1.79E+00 | 1.58E-09 |
| f_4 | 9.98E-02 | 9.18E-02 | 1.13E+01 | 0.00E+00 |
| f_5 | 3.41E+01 | 4.12E+01 | 1.18E+02 | 1.99E+00 |
| f_6 | 5.24E+03 | 4.75E+03 | 0.00E+00 | 0.00E+00 |
| f_7 | 7.55E-02 | 4.77E-02 | 2.61E-02 | 1.26E-02 |
| f_8 | -3.41E+03 | -4.32E+03 | -1.07E+04 | -8.86E+03 |
| f_9 | 4.95E+01 | 3.68E+01 | 1.46E+01 | 0.00E+00 |
| f_{10} | 1.21E+01 | 1.12E+01 | 8.57E+00 | 5.68E-15 |
| f_{11} | 2.13E-02 | 1.36E-02 | 1.70E-02 | 2.99E-02 |
| f_{12} | 6.24E+00 | 7.60E+00 | 4.01E+00 | 1.57E-32 |
| w/t/l | 11/0/1 | 11/0/1 | 9/1/2 | -/-/- |

为直观观察算法变化趋势,图 3 展示了 3 种组合算法在 2 个单峰函数(f_4 和 f_5)和 2 个多峰函数(f_7 和 f_{12})上的收敛曲线图. 通过分析图 3 可知,DLFA 算法在 4 个测试函数上收敛速度均快于其他 3 种组合算法. 另外 3 种组合算法均过早陷入局部最优,深度学习策略能有效延迟算法陷入局部最优的时间,寻优精度更高.

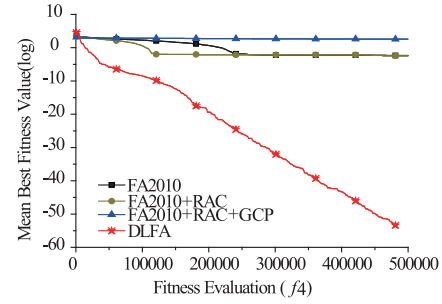
为综合评估 4 种组合算法的优化性能,对各算法进行 Friedman 检测. 表 8 给出了算法的秩均值. 秩均值越小,表示算法综合优化性能最佳. 从表 8 可知,FA 2010 + RAC 算法与 FA 2010 算法、FA 2010 + RAC + GCP 算法与 FA 2010 + RAC 算法、DLFA 算法与 FA 2010 + RAC + GCP 算法的秩均值差值分别为 0.66、0.04、1.25,说明三种改进策略中,深度学习策略改进效果最佳. 同时,DLFA 算法秩均值最小,表明 DLFA 算法寻优性能最佳.

表 8 不同组合策略萤火虫算法秩均值

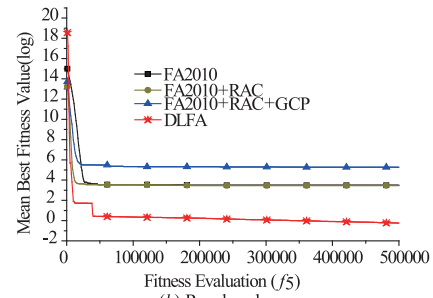
| Algorithm | Rankings |
|--------------------|----------|
| DLFA | 1.38 |
| FA2010 + RAC + GCP | 2.63 |
| FA2010 + RAC | 2.67 |
| FA2010 | 3.33 |

5 结论

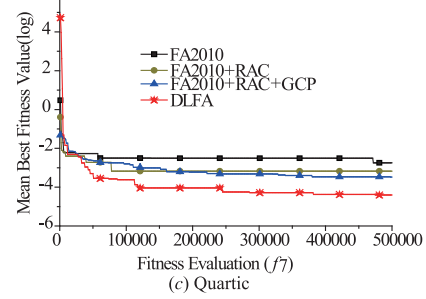
针对萤火虫算法寻优精度低和过早收敛的缺点,本文提出了深度学习萤火虫算法. 算法采用随机吸引模型替代全吸引模型,有效地防止了萤火虫在移动过程中出现振荡,提高了计算资源利用率;构造广义中心粒子并对其进行一定次数单维深度学习后,引导种群



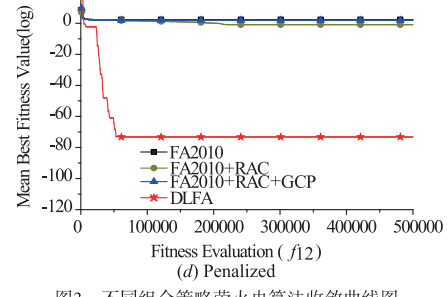
(a) Schwefel 2.21



(b) Rosebrock



(c) Quartic



(d) Penalized

图 3 不同组合策略萤火虫算法收敛曲线图

进化,完成寻优任务. 对深度学习次数、深度学习策略效果进行实验分析,发现深度学习策略、深度学习次数的选择对算法优化性能改善有重要影响. 12 个基准测试函数实验结果表明,DLFA 算法综合优化性能优于其它 8 种新近知名算法. 分析表 7,DLFA 算法在两两极值点之间相距较远的多峰函数 f_8 和 f_{11} 上求解精度差. 下一步研究方向是改进深度学习策略,使其能够更好地在局部极值间距大的多峰函数上发挥更优性能.

参考文献

[1] MASMOUDI A, CHAOUI S, MASMOUDI A. A finite

- mixture model of geometric distributions for lossless image compression [J]. *Signal Image & Video Processing*, 2016, 10(4): 671 – 678.
- [2] ARVALO R, GARCA J, GUIJARRO F, et al. A dynamic trading rule based on filtered flag pattern recognition for stock market price forecasting [J]. *Expert Systems with Applications*, 2017, 81: 177 – 192.
- [3] LEE H M, YOO D G, SADOLLAH A, et al. Optimal cost design of water distribution networks using a decomposition approach [J]. *Engineering Optimization*, 2016, 48 (12) : 2141 – 2156.
- [4] MASHAYEKHY L, NEJAD M M, GROSU D, et al. An online mechanism for resource allocation and pricing in clouds [J]. *IEEE Transactions on Computers*, 2016, 65 (4) : 1172 – 1184.
- [5] YEH W C. A novel node-based sequential implicit enumeration method for finding all d-MPs in a multistate flow network [J]. *Information Sciences*, 2015, 297: 283 – 292.
- [6] BHATI D, SINGH P. Branch and bound computational method for multi-objective linear fractional optimization problem [J]. *Neural Computing and Applications*, 2017, 28(11): 3341 – 3351.
- [7] MATHIEU L, OLIVIER P. Dynamic programming for mean-field type control [J]. *Journal of Optimization Theory & Applications*, 2016, 169(3): 902 – 924.
- [8] KENNEDY J, EBERHART R. Particle swarm optimization [A]. *Proceedings of International Conference on Neural Networks* [C]. Piscataway: IEEE, 1995. 1942 – 1948.
- [9] 陈佩松, 阮玉龙, 戴磊, 等. 区域分割的自适应变异粒子群算法[J]. *电子学报*, 2017, 45(8): 1849 – 1855.
CHEN Kan-song, RUAN Yu-long, DAI Lei, et al. Regional-segmentation self-adapting variation particle swarm optimization[J]. *Acta Electronica Sinica*, 2017, 45(8): 1849 – 1855. (in Chinese)
- [10] 周凌云, 丁立新, 彭虎, 等. 一种邻域重心反向学习的粒子群优化算法[J]. *电子学报*, 2017, 45(11): 2815 – 2824.
ZHOU Ling-yun, DING Li-xin, PENG Hu et al. Neighborhood centroid opposition-based particle swarm optimization [J]. *Acta Electronica Sinica*, 2017, 45(11): 2815 – 2824. (in Chinese)
- [11] YANG X S, DEB S. Cuckoo search via Lévy flights [A]. *Proceedings of World Congress on Nature and Biologically Inspired Computing* [C]. Piscataway: IEEE, 2010. 210 – 214.
- [12] 马卫, 孙正兴. 采用搜索趋化策略的布谷鸟全局优化算法[J]. *电子学报*, 2015, 43(12): 2429 – 2439.
MA Wei, SUN Zheng-xing. A global cuckoo optimization algorithm using coarse-to-fine search [J]. *Acta Electronica Sinica*, 2015, 43(12): 2429 – 2439. (in Chinese)
- [13] KARABOGA D. An Idea Based on Honey Bee Swarm for Numerical Optimization [R]. Kayseri, Turkey: Erciyes University, Engineering Faculty, Computer Engineering Department, 2005. 1 – 10.
- [14] YANG X S. *Nature-inspired Metaheuristic Algorithms* [M]. London: Luniver Press, 2008.
- [15] 杜鹏桢, 唐振民, 陆建峰, 等. 不确定环境下基于改进萤火虫算法的地面自主车辆全局路径规划方法[J]. *电子学报*, 2014, 42(3): 616 – 624.
DU Peng-zhen, TANG Zhen-min, LU Jian-feng, et al. Global path planning for ALV based on improved glow-worm swarm optimization under uncertain environment [J]. *Acta Electronica Sinica*, 2014, 42(3): 616 – 624. (in Chinese)
- [16] LECUN Y, BENGIO Y, HINTON G. Deep learning [J]. *Nature*, 2015, 521(7553): 436 – 444.
- [17] 马世龙, 乌尼日其其格, 李小平. 大数据与深度学习综述[J]. *智能系统学报*, 2016, 11(6): 728 – 742.
MA Shi-long, WUNIRI Qi-qi-ge, LI Xiao-ping. Deep learning with big data: state of the art and development [J]. *CAII Transactions on Intelligent Systems*, 2016, 11(6): 728 – 742. (in Chinese)
- [18] LIU Y, QIN Z, SHI Z, et al. Center particle swarm optimization [J]. *Neurocomputing*, 2007, 70(4): 672 – 679.
- [19] 汤可宗, 柳炳祥, 杨静宇, 等. 双中心粒子群优化算法[J]. *计算机研究与发展*, 2012, 49(5): 1086 – 1094.
TANG Ke-zong, LIU Bing-xiang, YANG Jing-yu, et al. Double center particle swarm optimization algorithm [J]. *Journal of Computer Research and Development*, 2012, 49(5): 1086 – 1094. (in Chinese)
- [20] 孙志远, 鲁成祥, 史忠植, 等. 深度学习研究与进展[J]. *计算机科学*, 2016, 43(2): 1 – 8.
SUN Zhi-yuan, LU Chen-xiang, SHI Zhong-zhi, et al. Research and advances on deep learning [J]. *Computer Science*, 2016, 43(2): 1 – 8. (in Chinese)
- [21] YANG X S. *Engineering Optimization: An Introduction with Metaheuristic Applications* [M]. Hoboken: Wiley Publishing, 2010.
- [22] FISTER I, FISTER I, BREST J, et al. Memetic firefly algorithm for combinatorial optimization [A]. *Proceedings of International Conference on Bioinspired Optimization Methods and their Applications* [C]. Piscataway: IEEE Press, 2012. 76 – 86.
- [23] GANDOMI A H, YANG X E, TALATAHARI S, et al. Firefly algorithm with chaos [J]. *Communications in Nonlinear Science & Numerical Simulation*, 2013, 18(1): 89 – 98.
- [24] YU S, SU S, LU Q, et al. A novel wise step strategy for

- firefly algorithm [J]. International Journal of Computer Mathematics, 2014, 91(12): 2507 – 2513.
- [25] YU S, ZHU S, MA Y, et al. A variable step size firefly algorithm for numerical optimization [J]. Applied Mathematics & Computation, 2015, 263: 214 – 220.
- [26] WANG H, WANG W, SUN H, et al. Firefly algorithm with random attraction [J]. International Journal of Bio-Inspired Computation, 2016, 8(1): 33 – 41.
- [27] WANG H, ZHOU X, SUN H, et al. Firefly algorithm with adaptive control parameters [J]. Soft Computing, 2017, 21(17): 5091 – 5102.

作者简介



赵 嘉 男, 1981 年 9 月生, 安徽桐城人. 教授、硕士生导师、中国电子学会高级会员. 主要研究方向为智能计算与大数据挖掘.
E-mail: zhaojia925@163.com



谢智峰 男, 1993 年 9 月生, 江西南昌人. 在读研究生. 主要研究方向为智能计算.



吕 莉 女, 1982 年 5 月生, 江西贵溪人. 副教授, 硕士生导师. 主要研究方向为智能计算和运动目标跟踪.

王 晖 男, 1982 年 8 月生, 湖北红安人. 博士、副教授、硕士生导师. 主要研究方向为智能计算和并行计算.

孙 辉 男, 1959 年 1 月生, 江西九江人. 博士、教授、硕士生导师. 主要研究方向为智能计算和 Rough 集.

喻 祥 男, 1983 年 7 月生, 江西景德镇人. 博士、讲师. 主要研究方向为智能计算与计算智能.