精英正交学习萤火虫算法

周凌云[1,2] 丁立新[1] 何进荣[1]
(武汉大学软件工程国家重点实验室 武汉 430072)[1]
(中南民族大学计算机科学学院 武汉 430074)[2]

摘 要 针对萤火虫算法后期收敛较慢以及求解精度不高的问题,提出了精英正交学习萤火虫算法。该算法利用精英萤火虫采用正交学习策略来构造指导向量,以保存和发现最优方向信息,从而引导群体更准确地飞向全局最优区域。同时,还采用了自适应步长技术来更好地平衡算法探索与开发能力,采用最小吸引力参数保证高维空间距离过大的个体之间的相互吸引。在6个经典测试函数上与标准萤火虫算法及其它3种改进的萤火虫算法进行了对比,实验结果表明,提出的算法具有较快的收敛速度和较高的收敛精度。

关键词 萤火虫优化,精英,正交学习,指导向量
中图法分类号 TP181 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.10.043

Elite Orthogonal Learning Firefly Algorithm

ZHOU Ling-yun[1,2] DING Li-xin[1] HE Jin-rong[1]
(State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072, China)[1]
(College of Computer Science, South-Central University for Nationalities, Wuhan 430074, China)[2]

Abstract In order to overcome the shortcomings of firefly algorithm such as slow convergence speed and low computational accuracy, an elite orthogonal learning firefly algorithm was proposed. An elite firefly was introduced to construct a guidance vector using the orthogonal learning strategy, which can preserve and discover useful information in the population best positions and direct the swarm to fly toward the global optimal region. At the same time, the method of adaptive step size was used to balance the exploration and exploitation ability of the algorithm, and the minimum attractive parameter was adopted to guarantee the attraction among the fireflies whose distance is large. We compared the proposed algorithm with standard firefly algorithm and other three improved firefly algorithms on six benchmarks, and the results show that the proposed algorithm obtains quicker convergence speed and better solution accuracy.

Keywords Firefly optimization, Elite, Orthogonal learning, Guidance vector

1 引言

萤火虫算法(Firefly Algorithm, FA)是2008年由剑桥大学的Xin-She Yang提出的一种元启发式算法[1,2],它是受萤火虫发光相互吸引的群体行为的启发而提出来的一种群智能优化算法。萤火虫发光吸引同伴,荧光强度越大且距离越近,对同伴的吸引力越强。每只萤火虫在视野范围内寻找更亮的萤火虫,不断地向更亮的萤火虫移动,以此来实现寻优目的。萤火虫算法已成功应用到许多科学、工程领域[3-7],并且表现出良好的性能。然而,它与大多数随机算法一样存在后期收敛速度慢、易陷入局部最优等缺点。

针对以上问题,目前已有一些改进方法。文献[8]将FA算法与模式搜索混合,用FA进行全局搜索,将搜索空间确定在一个较小的范围内,再用模式搜索作局部搜索。这种方法利用了FA算法较强的全局搜索能力和模式搜索较强的局部搜索能力,但它只是两种方法的简单组合,算法的收敛精度和稳定性有所提高,而收敛速度提高不大。文献[9]提出了并行萤火虫算法,在标准测试函数上测试,算法的收敛速度与精度均优于标准萤火虫算法,然而它必须采用多个种群。文献[10]以Levy分布作为随机步长改进了标准FA算法,实验结果优于遗传算法与粒子群算法。文献[11]提出的改进萤火虫算法修改了标准萤火虫算法中萤火虫随机漫步的方式,它的方法是从随机产生的均匀分布的 m 个单位方向向量中选择出使最亮萤火虫的亮度增加最大的方向向量,来更新最亮萤火虫的位置。如果没有找到这样的一个方向向量,最亮萤火虫则保持原来的位置。这种方法可以使最亮萤火虫一直向更接近全局最优的位置移动,从而提高了算法的性能。但是,在问题的维数较高时,很难从随机产生的方向向量中找出最优方向向量。文献[12]的改进方法则是每次迭代中,萤火虫移位后又采用高斯飞行寻找周围更优的方向信息,如果没有找

到稿日期:2014-05-06 返修日期:2014-07-20 本文受国家自然科学基金(61379059,61103248),中央高校基本科研业务费专项资金(CZY14011)资助。

周凌云(1979-),女,博士生,讲师,CCF会员,主要研究方向为计算智能,E-mail:zhouly@mail.scuec.edu.cn;丁立新(1967-),男,教授,博士生导师,主要研究方向为计算智能、云计算;何进荣(1984-),男,博士生,主要研究方向为机器学习。

到更优的方向信息，那么就停在原来的位置。同时它还采用了与问题维度和迭代次数相关的自适应步长技术。

FA算法的主要过程是萤火虫不断地飞向最亮的萤火虫，因此，如何从萤火虫的位置信息中挖掘出更多重要信息，找到更优方向，指引最亮萤火虫飞向更好的搜索区域，从而引导整个种群飞向全局最优区域，是增强FA算法性能的关键。

正交试验设计(Orthogonal Experimental Design，OED)是研究多因素多水平的一种设计方法，它根据正交性从全面试验中挑选出部分有代表性的点进行试验，是一种高效率、快速、经济的实验设计方法。OED方法被Zhang等[13]引入遗传算法，随后OED方法被应用到多种元启发式算法中，如蚁群优化算法[14]、模拟退火算法[15]，以及粒子群算法[16]，均取得了很好的效果。目前，在可查阅到的国内外文献中，还没有发现将OED方法应用到FA算法中的研究。

本文利用OED方法从萤火虫的位置信息中挖掘出更准确的最优方向信息，指导最亮萤火虫飞向最优区域，从而引导整个种群快速准确地向全局最优解收敛。在标准测试函数上进行数值实验，结果表明本文算法在这些测试函数上性能更好。

2 萤火虫算法(FA)

FA算法是一种元启发式的随机优化算法，它用夜空中的萤火虫个体模拟搜索空间中的可行解，将搜索和优化过程模拟成萤火虫相互吸引和位置更新过程，用目标函数的适应度值表示萤火虫的亮度，每个萤火虫都因受到邻近萤火虫中更亮萤火虫的吸引而移动位置。如果没有比这个萤火虫更亮的邻近萤火虫，那么它就随机移动或者不移动[2]。萤火虫之间的吸引力随着它们之间距离的减小而增大。萤火虫i与萤火虫j之间的吸引力计算公式为式(1)[2]。

$$\beta_{ij}=\beta_0 e^{-\lambda r_{ij}^2} \tag{1}$$

式中，β_0表示光源(距离$r=0$)处的吸引力，通常取1。r_{ij}表示两只萤火虫i，j之间的距离，通常使用欧氏距离，计算公式见式(2)，也可以根据实际问题使用其它距离。λ表示媒介对光的吸收率，通常取1。

$$r_{ij}=\| x_i-x_j \|_2=\sqrt{\sum_{k=1}^{d}(x_{ik}-x_{jk})^2} \tag{2}$$

一只萤火虫会飞向更亮的萤火虫，其位置更新方程定义为式(3)[2]。

$$x_i^{t+1}=x_i^t+\beta_{ij}(x_j^t-x_i^t)+\alpha\varepsilon^t \tag{3}$$

式中，x_i^{t+1}表示萤火虫i在$t+1$时刻的位置；等号右侧的第2项表示萤火虫i因受到萤火虫j的吸引而产生的位移；第3项是随机项，其中$\alpha\in[0,1]$，表示步长因子，ε^t是t时刻服从均匀分布的随机因子。

3 精英正交学习萤火虫算法(EOLFA)

距离一定时，亮度最强的萤火虫显然对其他萤火虫能产生较大的吸引力，对整个萤火虫群体飞向全局最优位置起着至关重要的引导作用。文献[11,12]的方法都因获得了较优的方向信息，并用来引导最亮萤火虫移动，从而提高了算法的性能。因此从萤火虫的位置信息中挖掘出更准确的方向信息可以进一步提高算法的性能。本文提出了精英正交学习萤火虫算法(Elite Orthogonal Learning Firefly Algorithm，EOLFA)，利用OED方法，设计正交学习策略，构建指导向量，从而引导萤火虫飞向全局最优位置。

3.1 正交试验设计

已有的研究表明，多种元启发式算法中应用OED方法均大幅提高了它们的性能[16]。本文的目的是利用OED方法设计一种正交学习策略，并构建一个指导向量。这个指导向量能够保存和发现萤火虫位置信息中的有用信息，指向一个更优的搜索方向，引导种群进入更有希望的搜索区域，从而提高FA算法的性能。

OED使用正交表和因子分析，能够以较少的实验次数发现各因素的不同水平之间的最优组合。它是一种系统的推理方法。例如，对于7个相互独立的因素，每个因素具有2个水平的实验。如果测试所有的组合，则需要$2^7=128$次实验。如果用OED方法，采用正交表$L_8(2^7)$和因子分析，仅需要做8次实验就可以找出最优组合。很明显，OED方法大幅减少了实验次数。

一个N因素Q水平的正交表通常记作$L_M(Q^N)$，其中L是正交表的符号，M表示正交表的行数，也就是实验的次数，N表示正交表的列数，也是最多可以容纳的因素个数。文献[17]给出了一个生成2水平正交表的算法。因子分析可以根据M次的实验结果发现不同因素的各个水平之间的最优组合。

定义1(影响度) 正交试验中，设f_i表示第i行的试验结果($i=1,2,\cdots,M$)，则第q个水平对第j个因素的影响度S_{jq}表示如下：

$$S_{jq}=\frac{\sum_{i=1}^{M} f_i\times z_{ijq}}{\sum_{i=1}^{M} z_{ijq}} \tag{4}$$

其中，$j=1,2,\cdots,D$。D是实际因素的个数，满足$D\leqslant N$。z_{ijq}的计算公式为

$$z_{ijq}=\begin{cases}0, & \text{if 第 } i \text{ 行组合第 } j \text{ 个因素使用第 } q \text{ 个水平}\\ 1, & \text{if 第 } i \text{ 行组合第 } j \text{ 个因素没用第 } q \text{ 个水平}\end{cases} \tag{5}$$

通过比较各个水平对每个因素的影响度大小，选出最优水平，确定最优组合。例如，对于求最大化问题，若$S_{j1}>S_{j2}$，则第j个因素应选择第1个水平。

3.2 精英正交学习策略

种群中亮度最强的萤火虫对种群的寻优有极其重要的作用，因此，把当前亮度最强的萤火虫与群体中其它萤火虫区分开，将其定义为精英，记作x_e。

定义2(精英) 对于一个种群$X=(x_1,x_2,\cdots,x_n)^{\mathrm{T}}$，设$f(x)$为目标函数，求其最小值。当第$t$次迭代时，$\forall i\in[1,n]$且$i\neq e,e\in[1,n]$，都有$f(x_i)\geqslant f(x_e)$，则称$x_e^t$为该种群迭代$t$次时的精英。

本文利用OED方法设计正交学习策略，构建一个指导向量x_0^t，用来保存和发现精英周围的最优方向信息。指导向量的计算公式为

$$x_0^t=x_e^t\oplus x_r^t \tag{6}$$

其中，$\oplus$表示正交操作，t表示当前迭代次数，x_r表示将要与精英作正交操作的位置向量。

本文引入正交试验设计方法是基于这样的假设：对于各维度，它们之间是相互独立的。问题的维度对应 OED 中的因素，维数对应因素的个数。x_e 和 x_r 在各维度上的位置对应于每个因素的水平，这样构建出的指导向量 x_0 能充分利用 x_e 和 x_r 的位置信息，是它们位置信息的最优组合。

本文选择两种位置向量作为 x_r。一种是精英最近邻萤火虫的位置。因为最近邻萤火虫是与精英距离最近的萤火虫，除精英外，它的位置信息里面往往包含了更多的最优位置信息，将它与精英进行正交操作，能够得到它们各维位置的最优组合，保存了群体搜索的最优方向信息，这样构建的指导向量必定包含更准确的最优方向信息。下面以优化三维函数 Sphere 函数 $f(x)=x_1^2+x_2^2+x_3^2$ 为例来进一步阐述这个原理。Sphere 函数的全局最优解在[0,0,0]。假设当前精英的位置为[1,1,0]，那么函数的评估值为 2。最近邻萤火虫的位置为[1.2,0,0.9]，那么函数评估值为 2.25。根据正交学习策略，可以得到指导向量 x_0 的位置[1,0,0]，此处的函数评估值为 1，明显更接近全局最优位置。

另一种位置向量通过式(7)计算得到。

$$x_r=x_e+\alpha\times\varepsilon\times scale \tag{7}$$

其中，$\alpha\in[0,1]$，表示步长因子；ε 是服从均匀分布的随机因子；$scale$ 是搜索空间的宽度。从该式可以看出，x_r 是精英周围一个小范围内的随机漫步位置。选择这样一个位置向量的原因是，它与精英构建的指导向量既可以保存精英中的最优位置信息，又可以发现精英周围有希望的方向信息，同时也增强了算法的局部开发能力。

算法 1 描述了用精英正交学习策略构建指导向量的过程。

算法 1 精英正交学习策略 OL

输入：精英位置 x_e，位置向量 x_r，目标函数 f

输出：指导向量 x_0

1)生成一个正交表 $L_M(2^D)$，$M=2^{\lceil\log_2(D+1)\rceil}$；

2)根据 L 生成 M 个测试解 C；

3)for i=1:M

for j=1:D

if (L_{ij}==0) $C_{ij}=X_{ej}$;end

if (L_{ij}==1) $C_{ij}=X_{rj}$;end

end for

end for

4)评价每个测试解；

5)根据式(4)和式(5)，计算各个水平的影响度；

6)根据影响度确定指导向量 x_0。

OED 中 2 水平 D 因素的实验次数 $M=2^{\lceil\log_2(D+1)\rceil}$，显然，$M\leqslant 2D$。用精英正交学习策略构建指导向量是 EOLFA 算法的关键，也是提高算法性能的关键。指导向量是通过正交学习策略来快速获得的最优方向信息，能够引导精英飞向最优位置，加快群体的收敛速度。为了说明精英正交学习策略能够加快算法的收敛速度，以优化 2 维的 Rosenbrock 函数为例，进行初步实验。

初始种群是在搜索空间中随机分布的，且两种算法采用相同的初始种群分布和参数设置，唯一不同的是，EOLFA 使用了精英正交学习策略。Rosenbrock 函数的全局最优解在点(1,1)处，图中用“☆”表示。“.”表示普通萤火虫，“○”表示精英。从图 1 可以看出，两种算法在演化过程中，萤火虫群体的活动区域都不断变化，逐步靠近全局最优解。相比 FA，EOLFA 中精英在演化到第 50 代时就到达了全局最优解，而且，种群能够更快且准确地靠近全局最优解，整个种群收敛得更快。这正是精英正交学习策略起到的作用。

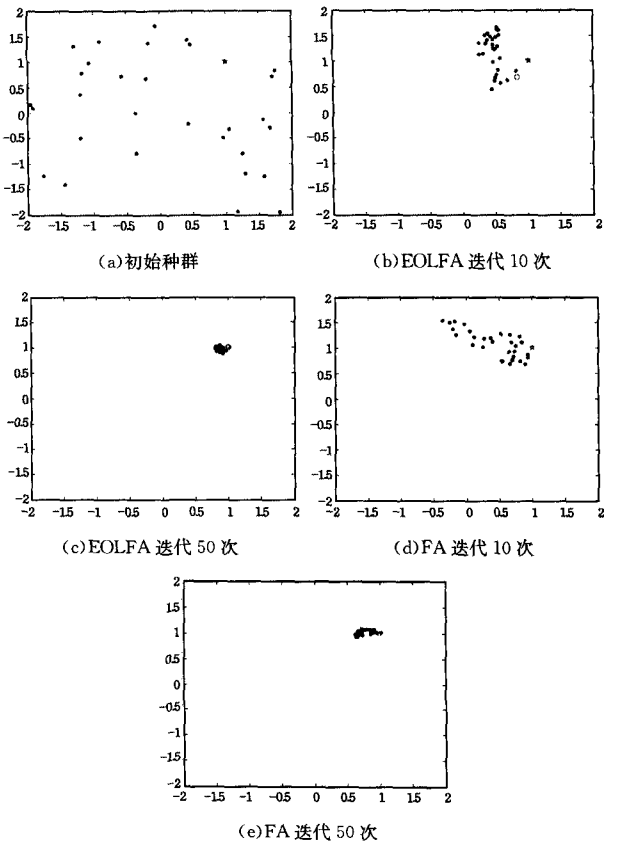

图 1 FA 算法和 EOLFA 算法在不同演化代数时群体的分布情况

3.3 EOLFA 算法

3.3.1 最小吸引力

标准 FA 中，萤火虫吸引力计算如式(1)所示：当 $r_{ij}\to\infty$ 时，$e^{-\lambda r_{ij}^2}\to 0$，则 $\beta_{ij}\to 0$。那么位置更新公式将变为：

$$x_i^{t+1}=x_i^t+\alpha\epsilon^t \tag{8}$$

这意味着距离很大时，萤火虫发光对其它个体的移动几乎没有任何影响和作用，个体的移动成为随机移动，导致算法收敛变慢。高维问题的搜索空间中，萤火虫之间的距离通常都很大。为了保证萤火虫之间距离比较大时，群体之间也能够受到一定的相互吸引力，本文引入最小吸引力参数 β_{min}，它通常取[0,1]之间的一个较小的值，并用式(9)作为萤火虫吸引力计算公式。

$$\beta_{ij}=(\beta_0-\beta_{min})e^{-\lambda r_{ij}^2}+\beta_{min} \tag{9}$$

3.3.2 自适应步长

标准 FA 算法中步长因子 α 采用[0,1]间的固定值，在迭代过程中 α 的值不变。然而，较大的 α 有利于算法的全局探索，但降低了搜索精度，迭代后期也容易出现震荡现象；较小的 α 值有利于算法的局部开发，搜索精度较高。本文采用随着迭代次数而减少的 α，使得 α 的值随着迭代过程而逐步递减，从而更好地平衡算法的探索与开发能力，计算公式如下：

$$\alpha^t=\alpha^{t-1}\delta,\ 0.5<\delta<1 \tag{10}$$

其中，δ 是冷却系数，$\delta \in [0.5,1]$。通过式(10)计算的 α，每次迭代后会得到一个更小的值，从而更好地平衡算法的全局探索能力和局部开发能力。

3.3.3 EOLFA 算法描述

EOLFA 算法可以描述如下。

算法 2 EOLFA 算法

输入：搜索空间 S，目标函数 f

输出：全局最优解和萤火虫最优位置

1)初始化参数 β_0，β_{min}，α，δ，λ 等；

2)初始化萤火虫种群 x_i($i=1,2,\cdots,n$)，计算种群亮度 I，并排序；

3)while (终止条件不满足)

4)for i=1:n

for j=1:i

if $I_i > I_i$

根据式(9)计算吸引力 β_{ij}；根据式(3)更新位置 x_i；

end if

end for

end for

5)越界处理；

6)if rand>p

x_r=最近邻萤火虫位置；

else

按照式(7)计算 x_r；

越界处理；

end if

7)$x_0 = OL(x_e, x_r, f)$；

8)计算种群亮度，并排序；

9)更新当前最优解及最优位置；

10)按照式(10)更新 α 的值；

11)end while

12)输出全局最优解及萤火虫最优位置。

算法中循环的终止条件可以是达到最大迭代次数，也可设置为求解达到某个误差精度。EOLFA 算法的流程主要是在 FA 算法的基本框架中采用了正交学习策略，以一定的概率 p 来交替选择两种位置向量 x_r，并与精英构建指导向量，更新当前最亮萤火虫的位置，更准确地引导群体飞向全局最优解。在计算两只萤火虫之间的吸引力 β_{ij} 时，设置了一个最小值，避免在高维搜索空间中因两只萤火虫距离较远而产生“0”吸引力现象，从而加快算法的收敛速度。同时，参数 α 的值采用了迭代递减方式，更好地平衡了算法的全局探索能力与局部开发能力。

EOLFA 算法主要包含了 3 个部分：初始化、萤火虫位置更新、精英正交学习。其中，初始化、萤火虫位置更新部分与 FA 算法具有相同的复杂度 $O(nt\log(n))$，精英正交学习部分的时间复杂度为 $O(MDt)$，因此，EOLFA 的复杂度仍然是迭代次数 t 的线性关系。

4 模拟实验与结果

4.1 测试函数与实验环境

为了评估 EOLFA 算法的优化性能，选取常用的 6 个标准函数[18,19]对算法进行测试，如表 1 所列。这 6 个函数都是国际上通用的全局优化问题的测试函数，其中 $f1$ 是单峰函数；$f2$ 是简单的多峰函数；$f3-f6$ 是具有多个局部极值的多峰函数，它们在全局最优位置附近有多个局部最优位置，导致算法容易陷于局部最优。近年来，优化高维多模函数已成为优化算法领域的研究热点。

表 1 实验的 6 个测试函数

	函数名	函数形式	搜索空间
f1	Sphere	$f_1(x)=\sum_{i=1}^{D} x_i^2$	$[-100,100]$
f2	Rosenbrock	$f_2(x)=\sum_{i=1}^{D}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	$[-2,2]$
f3	Rastrigin	$f_3(x)=\sum_{i=1}^{D}[x_i^2-10\cos(2\pi x_i)+10]$	$[-5.12,5.12]$
f4	Ackley	$f_4(x)=-20\exp(-0.2\times\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2})-\exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i))+20+e$	$[-32,32]$
f5	Griewank	$f_5(x)=\frac{1}{4000}\sum_{i=1}^{D}x_i^2-\prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}})+1$	$[-600,600]$
f6	Schwefel	$f_6(x)=-\sum_{i=1}^{D}(x_i\sin(\sqrt{\lvert x_i\rvert}))$	$[-500,500]$

选择标准 FA 算法[2]、LFA 算法[10]、MFA 算法[11]和 GDFA 算法[12]与本文提出的 EOLFA 算法进行对比实验。实验运行在一台配置为 Intel Core(TM)2 Duo CPU E4600 @ 2.40GHz，内存 2GB，操作系统为 Windows 8 专业版的计算机上。所有的算法在 Matlab 2013 下进行仿真。算法基本参数设置如下：群体个数均设为 30，参数 $\beta_0=1$，$\alpha=0.05$，$\lambda=1$。萤火虫的初始位置在搜索空间中随机生成。

4.2 EOLFA 算法参数分析

相比 FA 算法，EOLFA 算法中又引入了 3 个参数，分别是最小吸引力参数 β_{min}、冷却系数 δ 和一个交替概率 p。为验证这 3 个参数对 EOLFA 算法的影响，确定其合适的取值范围，采用给定范围内不同的取值在测试函数上分别进行实验。实验测试某一个参数时，其它参数均取定值。该实验选择了表 1 中的 3 个函数 $f1$、$f4$ 和 $f5$ 作为测试函数，EOLFA 算法最大迭代次数取 100，每个函数维度设置为 10，独立运行 10 次，取其平均值，结果如图 2 所示。

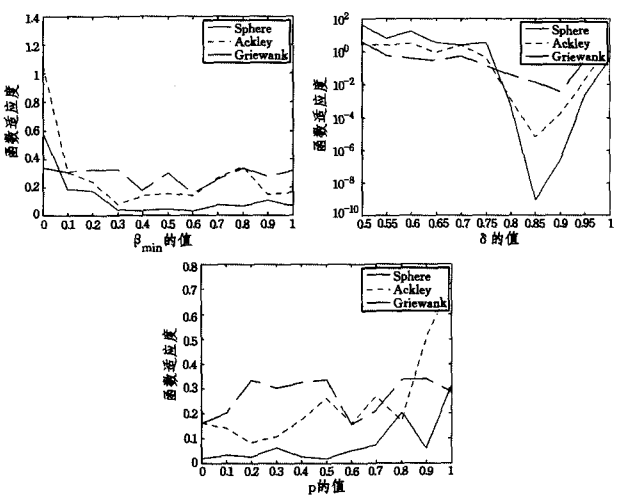

图 2 不同参数对算法性能的影响

从图中可以看出，最小吸引力参数 β_{min} 取[0.1,0.4]，冷却系数 δ 取[0.85,0.97]，交替概率 p 取[0.1,0.6]时，算法有较好表现。

4.3 EOLFA 算法在测试函数上的表现

对所有的测试函数，问题维度取 10 和 30 时分别进行实验；MFA 算法的参数为 $m=10$，GDFA 算法的参数为 $n=10/D$，EOLFA算法的参数为 $\delta=0.97$，$\beta_{min}=0.1$，$p=0.5$。记录 5 种算法在迭代 1000 次后所得解的平均值、最优值和最差值。每个测试函数上独立运行 20 次，记录其平均值。表 2 给出了实验结果，最好的值用粗体显示。

表 2　5 种算法求解的平均值、最优值和最差值

	算法	10 维			30 维		
		平均值	最优值	最差值	平均值	最优值	最差值
f1	FA	3.48e−02	1.24e−02	5.93e−02	2.15e+00	1.74e+00	2.36e+00
	LFA	4.29e−01	2.13e−02	1.19e+00	1.63e+00	1.24e+00	2.02e+00
	MFA	2.64e−06	2.75e−07	5.54e−06	9.14e−01	2.39e−01	1.46e+00
	GDFA	7.45e−07	1.34e−08	2.14e−06	2.34e−01	2.97e−02	6.08e−01
	EOLFA	**2.83e−12**	**2.78e−12**	**2.91e−12**	**9.11e−09**	**1.19e−09**	**2.28e−08**
f2	FA	7.56e+00	6.15e+00	9.33e+00	7.14e+01	3.85e+01	1.10e+02
	LFA	7.28e+00	**5.06e+00**	**8.45e+00**	3.50e+01	2.95e+01	5.58e+01
	MFA	7.91e+00	5.47e+00	9.38e+00	4.45e+01	3.14e+01	6.53e+01
	GDFA	9.09e+00	8.22e+00	9.58e+00	2.82e+01	2.66e+01	2.94e+01
	EOLFA	**7.16e+00**	6.06e+00	8.56e+00	**2.33e+01**	**2.33e+01**	**2.33e+01**
f3	FA	1.22e+01	3.00e+00	2.39e+01	8.53e+01	6.46e+01	1.01e+02
	LFA	1.68e+01	1.19e+01	2.37e+01	6.91e+01	5.67e+01	1.07e+02
	MFA	1.81e+01	1.09e+01	3.78e+01	7.99e+01	5.25e+01	9.80e+01
	GDFA	8.56e+00	4.97e+00	1.39e+01	3.64e+01	2.29e+01	4.98e+01
	EOLFA	**5.97e−01**	**1.01e−02**	**9.95e−01**	**3.18e+00**	**9.95e−01**	**4.97e+00**
f4	FA	3.48e−02	1.24e−02	5.93e−02	2.15e+00	1.74e+00	2.36e+00
	LFA	4.29e−01	2.13e−02	1.19e+00	1.63e+00	1.24e+00	2.02e+00
	MFA	2.64e−06	2.75e−07	5.54e−06	9.14e−01	2.39e−01	1.46e+00
	GDFA	7.45e−07	1.34e−08	2.14e−06	2.34e−01	2.97e−02	6.08e−01
	EOLFA	**2.83e−12**	**2.78e−12**	**2.91e−12**	**9.11e−09**	**1.19e−09**	**2.28e−08**
f5	FA	2.12e−01	5.51e−02	4.00e−01	5.61e−01	4.55e−01	6.95e−01
	LFA	1.97e−02	8.04e−05	3.94e−02	2.52e−01	7.21e−02	3.88e−01
	MFA	4.97e−02	2.88e−02	8.53e−02	2.52e−01	7.21e−02	3.88e−01
	GDFA	9.13e−03	7.80e−03	1.17e−02	1.31e−02	7.39e−03	2.31e−02
	EOLFA	**1.44e−15**	**6.66e−16**	**2.22e−15**	**3.08e−13**	**1.05e−13**	**6.20e−13**
f6	FA	1.59e+02	1.36e+02	2.09e+02	1.95e+02	1.76e+02	2.28e+02
	LFA	1.65e+02	1.30e+02	2.01e+02	2.45e+02	2.33e+02	2.60e+02
	MFA	1.45e+02	1.22e+02	1.82e+02	1.65e+02	1.39e+02	1.94e+02
	GDFA	1.15e+02	6.96e+01	1.69e+02	1.60e+02	1.28e+02	1.91e+02
	EOLFA	**7.32e+01**	**5.74e+01**	**8.03e+01**	**7.34e+01**	**6.89e+01**	**8.03e+01**

从表 2 的结果可以看出，函数维度为 10 时，EOLFA 算法求解的平均值都胜过其它算法；函数维度为 30 时，EOLFA 算法求解的平均值、最优值和最差值均优于其它算法，尤其是函数 $f1$，$f4$，$f5$ 上，EOLFA 算法求解精度远高于 FA、LFA、MFA 和 GDFA。对于函数 $f2$ 和 $f6$，由于函数特性，各算法均未找到全局最优解，但是 EOLFA 算法总能更接近全局最优解，结果好于其它算法。

LFA 算法采用 Levy 分布的随机步长比均匀分布的随机步长有更大的概率跳出局部最优，所以它在大多数函数上表现比 FA 算法好。MFA 算法和 GDFA 算法的寻优精度高于 FA 算法，这是因为 MFA 算法从最亮萤火虫周围的随机方向单位向量中获取较好的方向指引最亮萤火虫飞行，GDFA 算法基于高斯分布的移动来寻找群体飞行的方向，这两种方法都能获得一些最优方向信息。MFA 算法获取方向信息的准确度很大程度上依赖于随机产生的单位方向向量的个数 m，m 的值越大，获取的方向信息越准确。GDFA 算法是对群体中每个个体都基于高斯分布寻找一次方向。EOLFA 算法中，正交学习策略构建的指导向量充分利用群体中已有的最优方向信息，同时结合群体周围的最优方向信息来发现最优方向，它所找到的最优方向信息更准确，所以 EOLFA 算法表现出的性能更好。

图 3、图 4 描述了上述 5 种算法的收敛曲线，受篇幅限制，仅给出了 4 个代表性的函数在 10 维、30 维下求解的收敛结果。

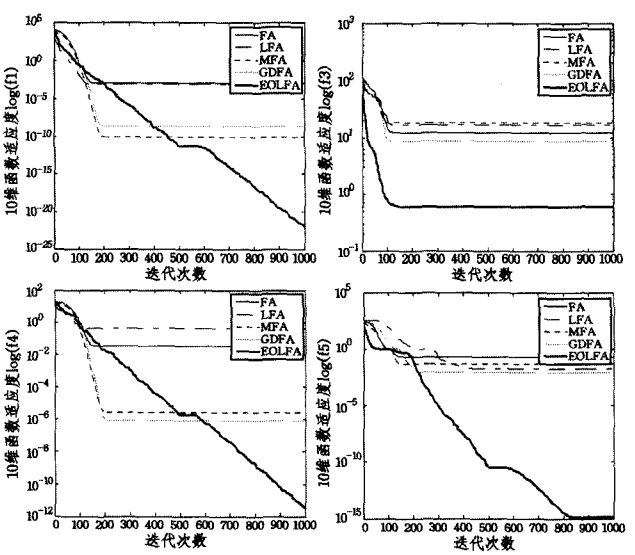

图 3　维度为 10 时算法收敛曲线对比

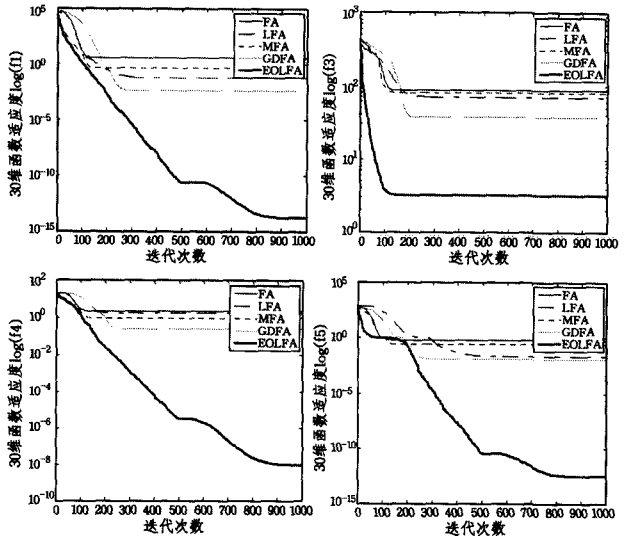

图 4　维度为 30 时算法收敛曲线对比

从图 3、图 4 中可以看出，对于测试函数，FA 算法收敛较慢，EOLFA 算法收敛最快，这是因为它能准确地找到搜索方向。对于函数 $f3$，5 种算法都出现了早熟。对于函数 $f4$，维度为 10 时，MFA 比 FA 收敛更快，但当维度增加到 30 时，算法 MFA 性能明显下降，收敛曲线几乎与 FA 重合，而 EOLFA 算法性能没有明显变化，仍然收敛最快。同样，对于其它函数，当维度增加时，FA、MFA、GDFA 算法性能均明显下降，而 EOLFA 算法对问题维度的增加则相对不那么敏感。这是因为，即使维度比较大时，正交学习策略也能够从较少的实验次数中找出最优方向信息的组合。实验结果说明 EOLFA 算法性能胜过其它 4 种对比算法，并且在问题维度增加时，EOLFA 算法能够产生更加稳定的性能。

结束语　本文提出了一种精英正交学习的萤火虫算法 EOLFA，萤火虫依靠相互之间的吸引而移动，同时选择当前最亮的萤火虫作为精英，根据正交学习策略创建指导向量，保存并发现最优方向信息，并引导群体移动。另外，该算法还借助了最小吸引力参数增强了距离较大时个体之间的吸引力，以及对步长因子施加冷却系统实现自适应步长来提高算法性能。在经典测试函数上展开实验，与多种萤火虫改进算法进行对比，实验结果表明 EOLFA 具有更好的性能。

参 考 文 献

[1] Yang X S. Nature-inspired metaheuristic algorithms [M]. Luniver press, 2010

[2] Yang X S. Firefly algorithms for multimodal optimization[M]// Stochastic algorithms: foundations and applications. Springer Berlin Heidelberg, 2009: 169-178

[3] Marichelvam M K, Prabaharan T, Yang X S. A discrete firefly algorithm for the multi-objective hybrid flowshop scheduling problems[J]. IEEE Transactions on Evolutionary Computation, 2014, 18(2): 301-305

[4] Yang X S, Hosseini S S S, Gandomi A H. Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect[J]. Applied Soft Computing, 2012, 12(3): 1180-1186

[5] Falcon R, Almeida M, Nayak A. Fault identification with binary adaptive fireflies in parallel and distributed systems[C]// 2011 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2011: 1359-1366

[6] Senthilnath J, Omkar S N, Mani V. Clustering using firefly algorithm: Performance study[J]. Swarm and Evolutionary Computation, 2011, 1(3): 164-171

[7] Fister I, JrFister I, Yang X S, et al. A comprehensive review of firefly algorithms[J]. Swarm and Evolutionary Computation, 2013, 13(1): 34-46

[8] Eslami M, Shareef H, Khajehzadeh M. Firefly algorithm and pattern search hybridized for global optimization [M]// Intelligent Computing Theories and Technology. Springer Berlin Heidelberg, 2013: 172-178

[9] Husselmann A V, Hawick K A. Parallel parametric optimisation with firefly algorithms on graphical processing units[C]// Proceedings of the 2012 World Congress in Computer Science, Computer Engineering, and Applied Computing. 2012

[10] Yang X S. Firefly algorithm, Levy flights and global optimization[M]// Research and Development in Intelligent Systems XXVI. Springer London, 2010: 209-218

[11] Tilahun S L, Ong H C. Modified firefly algorithm[J]. Journal of Applied Mathematics, 2012, 2012: 1-12

[12] Farahani S M, Abshouri A A, Nasiri B, et al. A Gaussian firefly algorithm[J]. International Journal of Machine Learning and Computing, 2011, 1(5): 448-453

[13] Zhang Q, Leung Y W. An orthogonal genetic algorithm for multimedia multicast routing[J]. IEEE Transactions on Evolutionary Computation, 1999, 3(1): 53-62

[14] Hu X M, Zhang J, Li Y. Orthogonal methods based ant colony search for solving continuous optimization problems[J]. Journal of Computer Science and Technology, 2008, 23(1): 2-18

[15] Ho S Y, Ho S J, Lin Y K, et al. An orthogonal simulated annealing algorithm for large floorplanning problems[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2004, 12(8): 874-877

[16] Zhan Z H, Zhang J, Li Y, et al. Orthogonal learning particle swarm optimization[J]. IEEE Transactions on Evolutionary Computation, 2011, 15(6): 832-847

[17] Ho S Y, Shu L S, Chen J H. Intelligent evolutionary algorithms for large parameter optimization problems[J]. IEEE Transactions on Evolutionary Computation, 2004, 8(6): 522-541

[18] 纪震，周家锐，廖惠连，等. 智能单粒子优化算法[J]. 计算机学报，2010，33(3)：556-561
Ji Zhen, Zhou Jia-rui, Liao Hui-lian, et al. A Novel Intelligent Single Particle Optimizer[J]. Chinese Journal of Computers, 2010, 33(3): 556-561

[19] Liang J J, Suganthan P N, Deb K. Novel composition test functions for numerical global optimization[C]// Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005 (SIS 2005). IEEE, 2005: 68-75