文章编号:1000－2367(2019)06－0015－07 **DOI**:10.16366/j.cnki.1000－2367.2019.06.003

基于精英邻居引导的萤火虫算法

汪春峰,褚新月

(河南师范大学 数学与信息科学学院,河南 新乡 453007)

摘 要:由于萤火虫的移动采用的是全吸引模型,所以当迭代过程中有移动时,可能会存在振荡较大、时间计算复杂度较高等问题.为了克服这些不足,提出了一种基于精英邻居引导的萤火虫算法.算法通过利用精英邻居的信息引导萤火虫的移动,减少振荡的发生,降低时间计算复杂度.同时,若某只萤火虫周围不存在精英邻居,则利用自身的信息进行反向学习以提高算法跳出局部最优的能力.数值实验表明本文算法的鲁棒性、寻优精度及搜索速度均优于其他几种算法.

关键词:萤火虫算法;精英邻居;反向学习;时间计算复杂度

中图分类号:O221 **文献标志码**:A

由于确定性算法对解决小规模问题比较有效,对于多模高维问题效率较低.因此,如何高效求解高维问题成了人们亟待解决的事情.群智能算法的提出使得此类问题得以解决.目前已有多种群智能算法被提出,如人工蜂群算法[1－2]、蚁群算法[3]、布谷鸟算法[4]、和声搜索算法[5]、粒子群算法[6]、萤火虫算法[7－9]等.

萤火虫算法(Firefly Algorithm,FA)是在 2008 年由英国剑桥大学学者 Yang 通过对萤火虫个体间的相互吸引和移动过程的探索而提出的一种新型群体智能优化算法[10].它是通过模拟萤火虫总是朝着更亮区域飞行的特性而设计的一种算法[11].尽管它提出的时间相对较晚,但由于其操作简单,设置参数较少,且性能优越,该算法已经被成功应用到诸多领域,如背包问题[12]、T－S 模型辨识[13]、聚类分析[14],配电网问题[15],机组组合问题[16],滴灌管网优化问题[17]等.但由于 FA 采用的是全吸引模型,所以算法本身存在振荡幅度较大,时间计算复杂度较高,寻优精度不高等缺陷.自 FA 提出之后,许多学者对其做了改进,包括设置参数方式的改进;与其他智能算法的结合;引入混沌搜索技术;采用拓扑结构等.比如在文献[18]中,通过采用递减的步长规则,作者提出了一个变步长 FA 算法,它可以很好的平衡搜索精度和收敛速度.为提高萤火虫算法的全局搜索能力,文献[19]将遗传算法(Genetic Algorithm,GA)与萤火虫算法做了混合.文献[20－22]通过利用混沌序列的随机性、遍历性和规律性,基于混沌序列对萤火虫的种群进行初始化以增强其多样性及全局搜索能力.为平衡局部搜索与全局搜索能力,文献[23]中利用 logistic 混沌映射引入了混沌局部搜索算子.文献[24]提出了一种基于随机吸引模型的萤火虫算法(RaFA),大大降低了时间计算复杂度,在此基础上,文献[25]提出了一种基于邻居吸引的萤火虫算法(NaFA),它首先将所有萤火虫放在一个环拓扑结构上,然后取前后各 k 个作为其邻居以引导萤火虫的移动,减少振荡现象的发生,增强稳定性.

为了降低算法的计算复杂度,减少振荡幅度,本文提出了基于精英邻居引导的萤火虫算法.实验结果表明本文算法的性能要比原始 FA,RaFA 和 NaFA 的好.

1 原始的萤火虫算法(FA)

在大自然界中,萤火虫利用自身所发出的光亮作为一种信号去吸引其他个体.FA 算法即是模拟萤火虫

收稿日期:2018-05-14;**修回日期**:2019-01-20.

基金项目:国家自然科学基金(11671122);河南省高等学校重点科研项目(19A110021);河南师范大学个人科研项目结余经费资助专项(校 20180562).

作者简介(通信作者):汪春峰(1978－),男,河南开封人,河南师范大学副教授,博士,主要研究智能优化算法及其应用、最优化理论算法及应用,E-mail:wangchunfeng09@126.com.

的这种行为而提出的群体智能算法.本文以求最小化问题为例,问题的维数设为 D.

1.1 FA 算法的 3 个基本原则

(1)假定萤火虫之间不区分性别,即任意一只比萤火虫 x_i 亮的萤火虫 x_j,都会吸引萤火虫 x_i 向其移动;

(2)萤火虫的吸引力和亮度成正相关,越亮吸引力越强,例如萤火虫 x_j 和萤火虫 x_k 都比萤火虫 x_i 要亮,其中萤火虫 x_k 最亮,则萤火虫 x_i 会向萤火虫 x_k 靠近;

(3)假如没有比萤火虫 x_i 更亮的个体,那么就让萤火虫 x_i 随机移动.

1.2 FA 算法的两个基本要素

亮度 $I(x_i)$ 和吸引度 β 是萤火虫算法的两个基本要素.亮度 $I(x_i)$ 越高其目标函数值 $f(x_i)$ 越优(最小),其定义如下:

$$I = I_0 \times \mathrm{e}^{-\gamma \times r_{ij}^2}, \tag{1}$$

其中 I_0 为萤火虫自身的亮度(此时 $r_{ij}=0$).r_{ij} 一般表示萤火虫 x_i 和萤火虫 x_j 之间的欧式距离,计算公式为:

$$r_{ij} = \sqrt{\sum_{k=1}^{D} (x_{ik} - x_{jk})^2}. \tag{2}$$

由(1)式可以看出距离越远其亮度就会越弱,也即越靠近萤火虫其亮度就越强,其中 γ 为光强的吸收系数.

吸引度 β 定义为

$$\beta = \beta_0 \times \mathrm{e}^{-\gamma \times r_{ij}^2}, \tag{3}$$

其中 β_0 代表最大吸引度,即萤火虫本身(光源)的吸引度.

1.3 萤火虫位置的更新公式

$$x_i^{t+1} = x_i^t + \beta \times (x_j^t - x_i^t) + \alpha \times (r^1 - \frac{1}{2}), \tag{4}$$

其中 t 为迭代次数,β 为吸引度,α 代表步长因子,$\alpha \in [0,1]$,r 为[0,1] 上的随机数.

1.4 FA 算法的基本步骤

步骤 1 在搜索空间内随机初始化所有的萤火虫 x_i,$i = 1,2,\cdots,N$;设置最大迭代次数 I_t^{Max},吸收系数 γ.

步骤 2 计算每只萤火虫的目标函数值 $f(x_i)$.

步骤 3 根据(4)式移动萤火虫 x_i 的位置,记移动后的位置为 $\tilde{x}_i$.

步骤 4 计算 $f(\tilde{x}_i)$,如果 $f(\tilde{x}_i) < f(x_i)$,记 $x_i^{t+1} = \tilde{x}_i$.

步骤 5 如果迭代次数 t 达到最大迭代次数 I_t^{Max},则算法终止,否则转第 3 步.

2 改进的萤火虫算法(ENFA)

在原始 FA 算法中,萤火虫移动采用的是全吸引模型(如图 1(a)),即每只萤火虫都要与其他萤火虫比较,只要与其比较的萤火虫的亮度比它高,都会吸引其向该萤火虫移动,这种模式主要有以下两点缺陷:①萤火虫移动过程中受其他萤火虫影响太多,造成移动过程中振荡太大,从而影响算法的收敛速度;②因每只萤火虫都要与其他萤火虫作比较,当种群规模 N 较大时,时间计算复杂度会较高.基于以上两点不足,本文提出了基于精英邻居引导的萤火虫算法.

2.1 精英邻居引导策略

为减弱萤火虫移动过程中的振荡现象,并减少计算复杂度,本节提出基于精英邻居的吸引模型(如图 1(b)).在图 1(a)中,实心的是比 x_i 亮度高的萤火虫的位置,空心的是比 x_i 亮度低的萤火虫的位置;在图 1(b)中,灰色实心圆和黑色实心圆都是比 x_i 亮度高的邻居,但是灰色的是亮度最高的邻居,也即是最好的邻居.

这里有一个关键问题:如何确定萤火虫 x_i 的邻居? 具体过程如下.

首先计算萤火虫 x_i 与其余萤火虫距离的平均值 m_d

$$m_d = \frac{\sum_{j=1 j\neq i}^{n} d_{ij}}{N-1}. \tag{5}$$

然后计算 x_i 与任一只萤火虫 $x_j(j \neq i)$ 的距离 d_{ij},若

$$d_{ij} \leq r \times m_d, \tag{6}$$

则认为 x_j 是 x_i 的邻居,其中 r 为 0 到 1 之间的随机数.最后,根据邻居的情形确定萤火虫 x_i 的移动方式.

情形 1:存在邻居时,分两种情况讨论:

(a)若邻居中最优萤火虫 x_k 比 x_i 的亮度高,则 x_i 以(7)式向 x_k 移动

$$x_i^{t+1} = x_i + \beta \times r \times (x_k - x_i^t) + \alpha \times \frac{\| x_k - x_i \|}{(u - l)}, \tag{7}$$

其中 u 和 l 分别表示搜索区域变量的上界和下界.

(b)若邻居中最优萤火虫 x_k 不比 x_i 的亮度高,则 x_i 以反向学习策略进行移动,其移动过程如下:

$$x_i^{t+1} = u + l - x_i^t. \tag{8}$$

情形 2:x_i 附近不存在邻居,此时采用蜂群算法中雇佣蜂的搜索策略进行移动,具体移动方式如下:

$$x_i^{t+1} = x_i^t + \varphi \times (x_k - x_i^t), \tag{9}$$

其中 φ 是区间$[-1,1]$内的随机数,$x_k \neq x_i$.

萤火虫 x_i 移动到新位置 x_i^{t+1} 之后,若 x_i^{t+1} 的适应度值 $f(x_i^{t+1})$ 优于种群当前最优解 x^* 的适应度值即 $f(x_i^{t+1}) < f(x^*)$ 则令

$$x^* = x_i^{t+1}, f(x^*) = f(x_i^{t+1}). \tag{10}$$

2.2　ENFA 算法描述

基于以上分析结果,下面给出本文基于精英邻居引导的萤火虫算法(ENFA)的描述.

步骤 1　设置最大迭代次数 I_t^{Max},萤火虫数量 n,步长因子 α.

步骤 2　初始化萤火虫的位置.

步骤 3　由(5)和(6)式确定每只萤火虫的邻居.

步骤 4　由(7)和(9)式确定每只萤火虫的新位置.

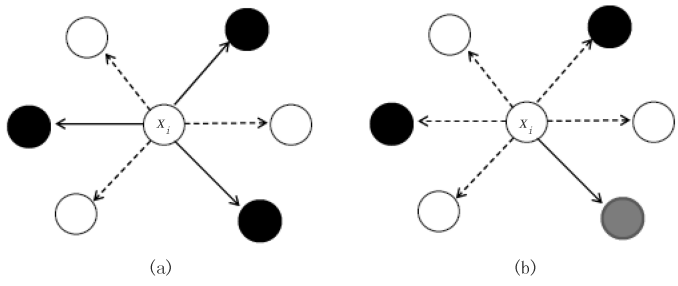

图 1　全吸引模型 (a) 和精英邻居吸引模型 (b)

Fig.1　Full attraction model(a) and Elite neighbor attraction model(b)

步骤 5　更新种群最优解.

步骤 6　若迭代达到最大迭代次数 I_t^{Max},算法停止,否则转第 3 步.

算法的流程图见图 2.

2.3　ENFA 算法的时间计算复杂度

对于时间计算复杂度,设 $O(f)$ 为适应度函数的时间计算复杂度,那么可知原始萤火虫算法的时间计算复杂度为 $O(I_t^{\text{Max}} \cdot N^2 \cdot f)$,其中 I_t^{Max} 是最大迭代次数;ENFA 算法的时间计算复杂度为 $O(I_t^{\text{Max}} \cdot N \cdot f)$,RaFA 算法和 NaFA 算法的时间复杂度分别为 $O(I_t^{\text{Max}} \cdot N \cdot f)$ 和 $O(I_t^{\text{Max}} \cdot k \cdot N \cdot f)$,显然,与其他 3 个算法相比,ENFA 算法大大地降低了时间计算复杂度.

3　实验过程与结果

为了验证 ENFA 算法的性能,本文选取了 CEC2005 中的 18 个标准测试函数用于测试[26],其中 $f_1 - f_{10}$ 为单模测试函数,$f_{11} - f_{18}$ 为多模测试函数,其相关特征如表 1 所示.

为公平起见,设种群的大小均为 40,维数为 30,最大迭代次数为 2 500.在原始 FA 算法、NaFA 算法和 RaFA 算法及 ENFA 算法中,吸引度的初始值 $\beta_0 = 1$,光吸收系数 $\gamma = 1$;在 RaFA 算法、NaFA 算法及 ENFA 算法中,$\alpha_0 = 0.25$. 算法在不同的初始化条件下独立运行 30 次,统计 30 次计算结果的均值、标准差和最小值.本文将计算结果与原始 FA 算法、RaFA 算法和 NaFA 算法做对比.具体比较结果见表 3.

3.1 试验 1:参数 r 的确定

在 ENFA 中,参数 r 的大小直接影响算法的时间复杂度,如何确定它的值比较关键.在此,选取表 1 中的 6 个函数 f_2,f_4,f_{10},f_{12},f_{15} 和 f_{17} 作为确定参数 r 值得测试函数.针对不同 r 的取值,每个函数独立运行 30 次,统计出相应的平均值和标准差,具体结果见表 2.

从表 2 中的数据可以看出,参数 r 的大小对函数 f_2,f_{15} 和 f_{17} 的影响不大,对函数 f_4 而言,r 的值越小实验的效果越好.对函数 f_{12} 而言,当 $r=0.4$ 或 0.6 时,实验效果最好.对函数 f_{10} 而言,当 $r=0.6$ 时,实验效果最好.综上所述,r 的值建议设置为 0.6.

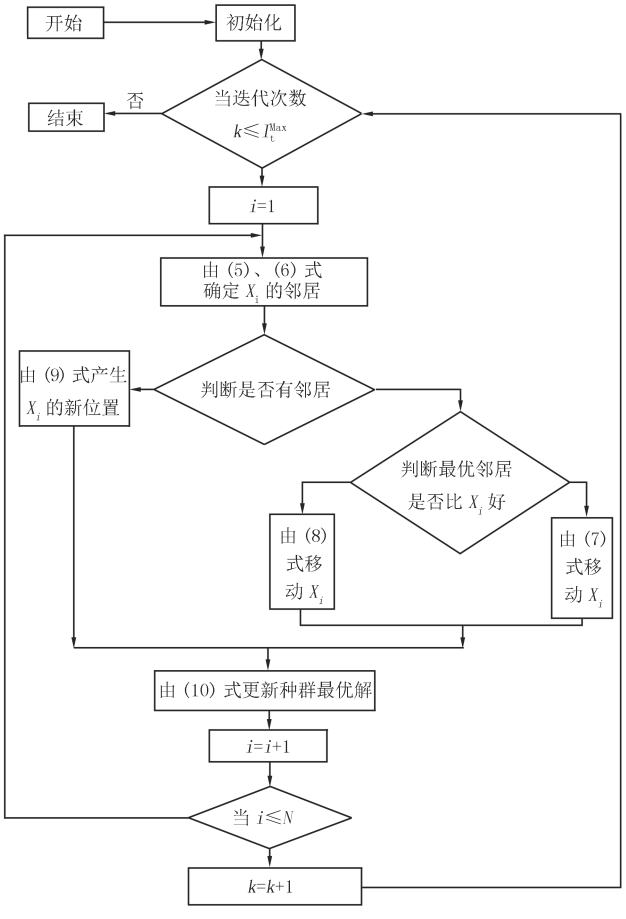

图 2 算法流程图

Fig.2 The flow chart of ENFA

表 1 标准测试函数

Tab.1 Benchmark functions

Functions	Range	Optimalvalue
$f_1=\sum_{i=1}^{n}x_i^2$	[−100,100]	0
$f_2=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{D}\lvert x_i\rvert$	[−10,10]	0
$f_3=\sum_{i=1}^{n}(\sum_{j=1}^{i}x_j)^2$	[−100,100]	0
$f_4=\max_i\{\lvert x_i\rvert,1\leqslant i\leqslant n\}$	[−100,100]	0
$f_5=\sum_{i=1}^{D}ix_i^2$	[−10,10]	0
$f_6=\sum_{i=1}^{D}ix_i^4$	[−1.28,1.28]	0
$f_7=\sum_{i=1}^{D}\lvert x_i\rvert^{(i+1)}$	[−1,1]	0
$f_8=\sum_{i=1}^{D}10^{6\frac{i-1}{D-1}}$	[−100,100]	0
$f_9=\sum_{i=1}^{n}(\lfloor x_i+0.5\rfloor)^2$	[−100,100]	0

续表

Functions	Range	Optimalvalue
$f_{10}=\sum_{i=1}^{n} ix_i^4+\text{random}[0,1)$	$[-1.28,1.28]$	0
$f_{11}=\sum_{i=1}^{D}(x_i^2-10\cos(2\pi x_i)+10)$	$[-5.12,5.12]$	0
$f_{12}=-20\exp(-0.2*\sqrt{\sum_{i=1}^{D}x_i^2/D})-\exp(\sum_{i=1}^{D}\cos(2\pi x_i/D))+20+e$	$[-32,32]$	0
$f_{13}=\frac{1}{4\ 000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1$	$[-600,600]$	0
$f_{14}=0.5+\frac{\sin(\sqrt{\sum_{i=1}^{D}x_i^2})^2-0.5}{(1+0.001\sum_{i=1}^{D}x_i^2)^2}$	$[-100,100]$	0
$f_{15}=\frac{\sum_{i=1}^{D}(x_i^4-16x_i^2+5x_i)}{D}$	$[-5,5]$	−78.332 36
$f_{16}=\sum_{i=1}^{n}\lvert x_i\sin(x_i)+0.1x_i\rvert$	$[-10,10]$	0
$f_{17}=\begin{cases}\sum_{i=1}^{n}(x_i^2-10\cos2\Pi x_i+10), & \lvert x_i\rvert<0.5\\ \sum_{i=1}^{n}[(\frac{\text{random}(2x_i)}{2})^2-10\cos\Pi\text{random}(2x_i)+10], & \lvert x_i\rvert\geq 0.5\end{cases}$	$[-5.12,5.12]$	0
$f_{18}=\frac{\Pi}{n}10\sin^2(\Pi y_1)+\frac{\Pi}{n}\sum_{i=1}^{n-1}(y_i-1)^2[1+10\sin^2(\Pi y_{i+1})]+\frac{\Pi}{n}(y_n-1)^2+\sum_{i=1}^{n}u(x_i,10,100,4)$, $y_i=1+\frac{x_i+1}{4}$, $u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m, & x_i>a\\ 0, & -a\leq x_i\leq a\\ k(-x_i-a)^m, & x_i<-1\end{cases}$	$[-50,50]$	0

表 2　参数 r 取不同值时的计算结果比较

Tab.2　The comparison results with different r

函数	$r=0.2$	$r=0.4$	$r=0.6$	$r=0.8$
f_2	1.22e-21(3.85e-21)	1.40e-21(4.43e-21)	1.41e-21(4.46e-21)	1.22e-21(3.87e-21)
f_4	2.87e-21(6.93e-21)	2.46e-05(7.80e-05)	5.20e-04(0.001 6)	3.54e-04(8.06e-04)
f_{10}	0.001 3(8.78e-04)	0.001 2(0.001 2)	3.95e-04(3.86e-04)	0.001 0(7.64e-04)
f_{12}	3.80e-04(7.35e-04)	9.41e-15(2.01e-14)	1.33e-14(1.13e-14)	2.29e-04(7.26e-04)
f_{15}	−78.332 3(2.48e-07)	−78.332 3(3.12e-07)	−78.332 3(2.85e-07)	−78.332 3(3.80e-08)
f_{17}	1.47e-04(2.53e-04)	1.95e-04(2.65e-04)	2.31e-04(4.09e-04)	3.57e-04(7.11e-04)

3.2　试验 2:ENFA 与其他改进萤火虫算法的比较

为了测试算法 ENFA 的性能,试验 2 以表 1 中给出的 18 个标准函数作为测试函数,将 ENFA 与 FA、RaFA 以及 NaFA 做了比较.对每一个标准测试函数,每种算法下独立运行 30 次,统计各算法所得到的最优解、平均值及标准差,计算结果如表 3 所示.由表 3 数据可以看出,对本文算法(ENFA)而言,除了在函数 f_{10} 中的标准差略差于 FA,在函数 f_{13} 中的标准差略差于 RaFA 之外,对于其他函数,ENFA 的寻优精度比原始萤火虫算法(FA),RaFA 和 NaFA 的要好.

4　结　论

本文对原始萤火虫算法做了深入研究,根据原始萤火虫算法的基本原理以及邻居的思想,提出了基于精英邻居引导的萤火虫算法(ENFA).通过采用精英邻居吸引模型,极大地降低了算法可能发生的振荡现象,

提高了算法的稳定性.实验结果表明,在适当设置参数的条件下,本文算法在求解精度、收敛速度、局部搜索能力与全局搜索能力的平衡以及时间计算复杂度方面都有明显的提高,算法更适用于各种优化问题.后期将进一步研究本文算法的收敛性,改善其性能,并将其应用到更复杂的实际问题中.

表 3 基本测试函数各算法比较结果

Tab.3 Comparison results of different algorithms based on benchmark functions

Function	Algorithm	Min	Mean	Std	Function	Algorithm	Min	Mean	Std
f_1	FA	14.686 9	17.888 3	2.023 3	f_{10}	FA	0.001 0	0.001 6	**4.93e-04**
	RaFA	1.28e+03	2.24e+03	976.327 7		RaFA	0.078 7	0.267 0	0.126 1
	NaFA	1.47e+03	2.74e+03	792.115 5		NaFA	0.155 3	0.401 1	0.224 0
	ENFA	1.12e-184	**2.77e-41**	**8.78e-41**		ENFA	2.30e-04	**0.0012**	6.75e-04
f_2	FA	1.704 0	1.849 8	9.47e-02	f_{11}	FA	47.075 1	59.295 0	10.128 7
	RaFA	12.795 4	19.544 8	3.397 7		RaFA	135.401 2	154.668 7	10.612 5
	NaFA	15.056 9	19.052 9	3.259 9		NaFA	104.771 2	130.701 3	16.023 2
	ENFA	1.58e-120	**1.06e-21**	**3.38e-21**		ENFA	0	**8.73e-06**	**1.73e-05**
f_3	FA	26.130 6	40.109 6	5.360 3	f_{12}	FA	1.909 2	2.247 0	0.155 2
	RaFA	1.76e+03	3.69e+03	1.21e+03		RaFA	9.212 1	10.009 2	0.573 4
	NaFA	1.06e+03	3.13e+03	1.37e+03		NaFA	7.723 3	9.967 1	1.428 8
	ENFA	3.79e-41	**9.91e-33**	**2.76e-32**		ENFA	6.21e-15	**1.19e-14**	**7.89e-15**
f_4	FA	1.489 6	1.610 1	8.78e-02	f_{13}	FA	0.979 4	0.995 3	0.005 8
	RaFA	11.219 2	21.575 2	4.832 9		RaFA	0.987 8	0.995 2	**0.004 1**
	NaFA	17.691 4	22.243 2	5.213 2		NaFA	0.838 3	0.948 5	0.049 0
	ENFA	2.45e-20	**8.81e-10**	**2.75e-09**		ENFA	1.11e-16	**0.048 2**	0.1515
f_5	FA	2.149 1	2.623 6	0.233 5	f_{14}	FA	0.037 2	0.041 3	0.012 9
	RaFA	173.886 4	278.558 7	106.127 0		RaFA	0.451 8	0.474 5	0.013 7
	NaFA	231.968 3	347.443 9	62.327 0		NaFA	0.451 8	0.471 0	0.009 2
	ENFA	4.05e-184	**1.40e-41**	**4.44e-41**		ENFA	2.58e-08	**0.0031**	**0.004 6**
f_6	FA	4.37e-06	9.33e-06	2.29e-06	f_{15}	FA	−70.747 0	−67.350 8	3.330 3
	RaFA	0.081 5	0.170 0	0.105 7		RaFA	−45.280 9	−42.091 3	2.146 4
	NaFA	0.176 8	0.328 3	0.151 2		NaFA	−46.672 2	−42.575 8	3.391 6
	ENFA	8.39e-257	**7.52e-88**	**2.38e-87**		ENFA	−78.332 3	**−78.332 3**	**4.49e-07**
f_7	FA	1.32e-09	6.23e-09	3.92e-09	f_{16}	FA	1.068 5	1.631 8	0.420 8
	RaFA	4.20e-07	1.26e-05	1.18e-05		RaFA	10.653 4	12.027 3	1.411 9
	NaFA	8.79e-08	2.84e-05	4.38e-05		NaFA	7.234 4	10.949 7	2.626 6
	ENFA	4.94e-324	**4.25e-52**	**1.34e-51**		ENFA	1.56e-114	**1.12e-22**	**3.54e-22**
f_8	FA	3.50e+05	5.07e+05	1.49e+05	f_{17}	FA	42.611 8	57.348 7	12.481 5
	RaFA	5.99e+06	2.04e+07	9.52e+06		RaFA	46.185 7	104.191 7	40.608 0
	NaFA	1.00e+07	3.93e+07	2.56e+07		NaFA	59.443 3	81.690 8	14.644 7
	ENFA	7.82e-166	**4.43e-36**	**1.40e-35**		ENFA	0	**1.12e-05**	**1.63e-05**
f_9	FA	3	4.500 0	0.971 8	f_{18}	FA	0.057 5	0.124 2	0.068 8
	RaFA	1	2.400 0	1.173 8		RaFA	9.398 7	170.282 8	446.818 7
	NaFA	3	4.300 0	1.059 3		NaFA	10.222 4	1.46e+03	4.11e+03
	ENFA	0	**0**	**0**		ENFA	4.66e-32	**3.59e-28**	**6.77e-28**

参 考 文 献

[1] Karaboga D, Basturk B. Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problem[J]. Advances in Soft Computing: Foundations of Fuzzy Logic and Sof Computing, 2007, 4529: 789-798.

[2] Karaboga D, Basturk B, Ozturk C. Artificial bee colony (ABC) optimization algorithm for training feed-forward neural networks[J]. Modeling Decisions for Artificial Intelligence, 2007, 4617: 318-319.

[3] 袁培燕，刘萍，高宏卿. 蚁群算法迭代次数的一种优化策略[J]. 河南师范大学学报(自然科学版)，2010，38(4)：48-50.

[4] Yang X S, Deb S. Engineering optimization by cuckoo search. International Journal[J]. Mathematical Modelling and Numerical Optimization, 2010, 1(4): 330-343.

[5] Geem Z W, Kim J H, Loganathan G V. A new heuristic optimization algorithm: Harmony Search[J]. Special Section on Medical Modelling and Simulation, 2001, 76(2): 60-68.

[6] 张新明，王霞，涂强，等. 融合榜样学习和反向学习的粒子群优化算法[J]. 河南师范大学学报(自然科学版)，2017，46(6)：91-99.

[7] Yang X S. Firefly Algorithm. in Engineering Optimization[M]. Hoboken: John Wiley and Sous, Inc, 2010.

[8] Yang X S, He X. Firefly Algorithm: Recent advances and application[J]. International Journal. Swarm Intelligence, 2013, 1(1): 36-55.

[9] Yang X S. Firefly algorithms for multimodal optimization[J]. Stochastic Algorithms: Foundations and Applications, 2009, 5792: 169-178.

[10] Yang X S. Nature-Inspired Metaheyristic Algorithms[M]. UK: Luniver Press, 2008.

[11] 郑楠，颜敏等. 一种改进混沌萤火虫算法[J]. 计算机与应用化学，2014，31(8)：987-998.

[12] Baykasoglu A, ozsoydan F B. An improverd firefly algorithm for solving dynamic mutidimensional knapsack problems[J]. Expert Systems with Applications, 2014, 41(8): 3712-3725.

[13] 吴东固，丁学明. 基于改进萤火虫算法的 T-S 模型辨识[J]. 计算机仿真，2013，30(3)：327-330.

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

[15] 许喆，潘金生，樊淑娴，等. 基于改进萤火虫算法的含 DG 配电网重构方法[J]. 电力系统保护与控制，2018，46(14)：26-32.

[16] 方必武，王波，刘涤尘，等. 基于搜索＋调整的两阶段萤火虫算法求解机组组合问题[J]. 电力系统保护与控制，2016，44(23)：17-23.

[17] 陈际旭，徐淑琴，周豪. 基于萤火虫算法的滴灌管网优化设计研究[J]. 灌溉排水学报，2018，37(9)：48-55.

[18] Yu S H, Zhu S L. A variable step size firefly algorithm for numerical optimization[J]. Applied Mathematics and Computation, 2015, 263: 214-216.

[19] Farahani S M, Abshouri A A, Nasiri B etal. Some hybrid models to improve firefly algorithm performance[J]. International Journal of Artificial Intelligence, 2012, 8(s12): 97-117.

[20] 徐华丽. 一种混沌多样性控制的萤火虫优化算法[J]. 中国科学技术大学学报，2014，104(7)：612-617.

[21] 马小雨，高继勋. 具有全局收敛性的改进萤火虫优化算法[J]. 科学技术与工程，2013，13(11)：2991-2996.

[22] 冯艳红等. 基于混沌理论的动态种群萤火虫算法[J]. 计算机应用，2013，33(3)：796-799.

[23] 刘长平，叶春明. 具有混沌搜索策略的萤火虫算法[J]. 系统管理学报，2013，22(4)：538-543.

[24] Wang H, Wang W J, Sun H, et al. Firefly algorithm with random attraction[J]. International Journal Bio-Inspired Computation, 2016, 8(1): 33-41.

[25] Wang H, Wang W J, Zhou X Y, et al. Firefly algorithm with neighborhood attraction[J]. Information Sciences, 2017, 282-283: 374-387.

[26] Suganthan P N, Hansen N, Liang J, et al. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization[R]. [出版地不详：出版者不详], 2005.

A firefly algorithm based on elite neighborhood guide

Wang Chunfeng, Chu Xinyue

(College of Mathematics and Information Science, Henan Normal University, Xinxiang 453007, China)

Abstract: Since the full attraction model is used in FA, it may result in strong oscillations during the movement and high time complexity. In order to overcome these disadvantages, this paper proposes a modified FA based on elite neighborhood guide. The algorithm leads fireflies to move by using the information of elite neighborhoods, so it can reduce the occurrence of oscillation and accelerate convergence. Meanwhile, if there is no elite neighborhood around some fireflies, they will do opposition-based learning to help the algorithm to get out of the local optima by using their own information. The experimental results show that the proposed algorithm is superior to other algorithms.

Keywords: firefly algorithm; elite neighborhood; opposition-based learning; time complexity

[责任编校　陈留院　赵晓华]