第46卷第2期 中国科学技术大学学报 Vol. 46, No. 2
2016年2月 JOURNAL OF UNIVERSITY OF SCIENCE AND TECHNOLOGY OF CHINA Feb. 2016

文章编号:0253-2778(2016)02-0120-10

保持个体活性的改进FA算法

陆克中[1],章哲庆[1],孙 俊[2]

(1.池州学院计算机科学系,安徽池州 247100;2.江南大学物联网学院,江苏无锡 214021)

摘要:在对基本FA算法进行分析的基础上,指出FA算法存在全局搜索能力不足,以及因聚集而早熟收敛现象.为克服FA算法的不足,提出了保持个体活性的改进FA(IFA)算法,分别从γ值自适应调节、过程萤火虫位置更新、聚集萤火虫重新激活以及最优个体的局部处理等多个方面对基本FA算法进行了改进.通过在10个基准测试函数上的测试,并与基本FA算法、PSO算法、ABC算法和其他改进FA算法进行对比,实验结果表明,改进的IFA算法能够很好地保持种群的多样性,具有较快的收敛速度与较好的求解精度,适合复杂函数优化问题.

关键词:群智能;萤火虫算法;多样性;函数优化

中图分类号:TP18 **文献标识码**:A doi:10.3969/j.issn.0253-2778.2016.02.00

引用格式:Lu Kezhong, Zhang Zheqing, Sun Jun. Improving firefly algorithm by keeping individual activity[J]. Journal of University of Science and Technology of China, 2016,46(2):120-129.
陆克中,章哲庆,孙俊. 保持个体活性的改进FA算法[J]. 中国科学技术大学学报,2016,46(2):120-129.

Improving firefly algorithm by keeping individual activity

LU Kezhong[1], ZHANG Zheqing[1], SUN Jun[2]

(1. *Department of Computer Science, Chizhou University, Chizhou* 247100, *China*;
2. *School of IoT Engineering, Jiangnan University, Wuxi* 214021, *China*)

Abstract: There are some disadvantages in the basic firefly algorithm(FA), such as low solving precision, premature convergence and etc. To overcome these disadvantages, a novel improved FA (IFA) was proposed that keeps individual activity. Firstly, an adaptive control for gamma value was designed by using swarm distance. Secondly, a position calculation for fireflies was updated by using the search process information. Thirdly, a special mutation for the firefly swarm was executed to activate individuals and to make them explore the search space when losing activity. Finally, a perturbation and local search method for the best individual was proposed. Based on ten multi-model test functions, the test results show that the IFA has a better convergence speed and precision than the basic FA, PSO, ABC and other improved FA. The improved FA is a good method for complex function optimization.

Key words: swarm intelligence; firefly algorithm; diversity; function optimization

收稿日期:2015-01-16;**修回日期**:2015-12-01

基金项目:国家自然科学基金(61170119),安徽省自然科学研究项目(KJ2016A514)资助

作者简介:陆克中(通讯作者),男,1976年生,硕士/副教授. 研究方向:智能优化算法及应用、生物信息学等. E-mail:luke76@163.com

0 引言

近年来，模拟自然界特点的自然计算方法得到了长足发展. 最近，剑桥大学的 Yang[1] 受萤火虫利用自身荧光传递信息这种群体行为的启发，提出了一种新颖的群智能优化算法——萤火虫算法(firefly algorithm，FA). FA 算法与其他自然计算方法相似，是基于群体的优化搜索算法，无需梯度信息，具有很强的并行性与通用性. 另外，FA 算法具有结构简单、调节参数少、易于操作实现等特点，因而自提出以来，就得到了国内外学者的较大关注，并在函数优化[2]、经济调度[3]、图像处理[4]、证券投资[5]、聚类分析[6]、自动控制[7]等领域得到了应用.

FA 算法提出时间较晚，不论是理论基础，还是算法设计，尚处在初期阶段，还有很大的发展空间. 文献[8]指出，FA 算法存在早熟收敛、后期收敛速度慢、求解精度较低等缺点. 为了克服这些缺点，国内外学者对 FA 算法进行了有益的改进，取得了一些研究成果. 文献[9]运用混沌映射产生均匀分布的萤火虫初始位置，获得质量较好的初始解，并在搜索过程中对适应值低的部分萤火虫进行混沌扰动，以保持群体活性，减小陷入局部最优的可能性. 文献[10]应用混沌映射产生初始萤火虫个体，并在搜索过程中，依据最优个体利用混沌映射生成多个新的个体，随机替换萤火虫群体中等量个体，以保持种群的多样性. 文献[11]应用多种混沌映射技术，分别对光吸收系数与吸引度 进行混沌调节，实验结果显示，混沌调节有利于提升 FA 算法的全局搜索能力. 文献[12]利用莱维飞行(L'evy flights)的随机性与各向同性，用莱维飞行来改写 FA 算法中的局部随机搜索部分，能有效提高算法的搜索精度. 文献[13]提出离散 FA 算法，用于解决离散优化问题. 还有的文献[14-15]将 FA 算法与其他自然计算方法相混合，形成混合优化算法，通过取长补短提高了整个算法的性能.

本文在对基本 FA 算法进行分析的基础上，分别从 γ 值自适应调节、过程萤火虫位置更新、聚集萤火虫重新激活以及对最优个体的处理等多个方面，对 FA 算法进行了改进，以保持群体活性，增强 FA 算法的局部与全局搜索能力.

1 基本 FA 算法

2008 年 Yang 模拟自然界萤火虫发光模式与信息交换行为，提出了 FA 算法. 为使算法简单、实用，使用了以下 3 个理想化规则：①萤火虫不分雌雄；②萤火虫的吸引度与自身的荧光亮度成正比，且萤火虫间的相互吸引随着距离的增加而下降；③萤火虫的荧光亮度取决于问题的目标函数.

根据上面的规则可知，荧光亮度和吸引度是 FA 算法中的两个关键因素. 假设萤火虫的群体规模为 m，问题空间为 d 维，第 i 只萤火虫在 d 维空间中的位置表示为 $\boldsymbol{x}_i=(x_{i1},x_{i2},\cdots,x_{id})$.

定义 1　荧光亮度

$$I = I_0 \mathrm{e}^{-\gamma r} \tag{1}$$

式中，γ 为光强吸收因子，表示荧光亮度受传播距离与传播媒介的影响而变化的特性，理论上 $\gamma\in[0,\infty)$，但在实际应用中，$\gamma=O(1)$，常取$[0.01,100]$之间的某一常数. I_0 为萤火虫的最大荧光亮度，即为 $r=0$ 处的自身荧光亮度，取决于需要寻优的目标函数值，一般用式(2)表示：

$$I_0 = f(\boldsymbol{x}) \tag{2}$$

式中，f 为待优化函数，$\boldsymbol{x}$ 为某一萤火虫的空间位置向量；$f(\boldsymbol{x})$越好则 I_0 就越高. r 表示两只萤火虫之间的空间距离，一般用式(3)的欧式距离表示：

$$r_{ij} = \| \boldsymbol{x}_i - \boldsymbol{x}_j \| = \sqrt{\sum_{k=1}^{d}(x_{i,k}-x_{j,k})} \tag{3}$$

定义 2　吸引度

$$\beta = \beta_0 e^{-\gamma r^2} \tag{4}$$

式中，β_0 为最大吸引度，即光源($r=0$ 处)的吸引度，通常设定为 1；γ，r 的意义同定义 1.

定义 3　位置更新公式

$$\boldsymbol{x}_i(t+1) = \boldsymbol{x}_i(t) + \beta(\boldsymbol{x}_j(t)-\boldsymbol{x}_i(t)) + \alpha\boldsymbol{\varepsilon}_i \tag{5}$$

上式表示的是萤火虫 i 受萤火虫 j 的吸引而发生位置改变. 式中，t 为迭代次数，x_i 与 x_j 分别表示萤火虫 i 与 j 的空间位置；β 表示萤火虫 j 对萤火虫 i 的相对吸引度；α 为步长因子，一般按照式(6)进行变化；ε_i 是一个服从高斯分布或均匀分布的随机向量，其简化形式为 rand$-1/2$，rand 为$[0,1]$内服从均匀分布的随机数.

$$\alpha = \alpha' * \mathrm{delta}^t \tag{6}$$

式中，α'为初始步长因子，delta$\in(0,1]$，t 为迭代次数.

求解最大值的算法流程描述如下：

生成初始萤火虫群体{ x_i，i=1,2,…,n}；
由式(1)计算萤火虫 i 的荧光亮度 I_i
设置参数 γ，β_0 和最大迭代次数 Tmax

```
while(t<Tmax)
for i=1:n
  for j=1:n
    if(I_j>I_i)
      由式(3)计算 x_i, x_j之间的距离 r_ij
      由式(4)计算 x_j对 x_i的吸引度 β
      由式(5)计算 x_i的新位置
    end if
  end for j
end for i
确定最优萤火虫个体,并且 t++
end while
```

2 FA 算法分析

由式(4)的吸引度表达式可知,当 $\gamma=1$, $r>2$ 时, $e^{-\gamma r^2}\to 0$,故而 $\beta\to 0$,代入式(5),有

$$\boldsymbol{x}_i(t+1)=\boldsymbol{x}_i(t)+\beta(\boldsymbol{x}_j(t)-\boldsymbol{x}_i(t))+\alpha\boldsymbol{\varepsilon}_i\approx \boldsymbol{x}_i(t)+\alpha\boldsymbol{\varepsilon}_i \tag{7}$$

当 $\gamma=1$, $r<0.2$ 时, $e^{-\gamma r^2}\to 1$. 故而 $\beta\to\beta_0$,代入式(5),有

$$\boldsymbol{x}_i(t+1)=\boldsymbol{x}_i(t)+\beta(\boldsymbol{x}_j(t)-\boldsymbol{x}_i(t))+\alpha\boldsymbol{\varepsilon}_i\Rightarrow$$
$$\boldsymbol{x}_i(t+1)\approx\boldsymbol{x}_i(t)+\beta_0(\boldsymbol{x}_j(t)-\boldsymbol{x}_i(t))+\alpha\boldsymbol{\varepsilon}_i\Rightarrow$$
$$\boldsymbol{x}_i(t+1)\approx\beta_0\,\boldsymbol{x}_j(t)+(1-\beta_0)\,\boldsymbol{x}_i+\alpha\boldsymbol{\varepsilon}_i$$

由于 β_0 通常设定为 1,所以上式可化简为

$$\boldsymbol{x}_i(t+1)\approx\boldsymbol{x}_j(t)+\alpha\boldsymbol{\varepsilon}_i \tag{8}$$

当 $r>2$ 时,即萤火虫 i 和 j 相距比较远时,由式(7)可知,萤火虫 j 对 i 的吸引非常小,这时萤火虫 i 的位置改变(即搜索),变换为在自身周边进行局部随机搜索. 当 $r<0.2$ 时,即萤火虫 i 和 j 相距比较近时,由式(8)可知,萤火虫 i 被萤火虫 j 所吸收,这时萤火虫 i 的位置改变(即搜索),变换为在萤火虫 j 周边进行局部随机搜索.

对于 $\gamma\neq 1$ 的情况, γ 影响萤火虫 j 对萤火虫 i 的吸引范围. 相比 $\gamma=1$ 时的情况,当 $\gamma>1$ 时,其值越大,则萤火虫 j 的吸引范围就越小;当 $\gamma<1$ 时,其值越小,则萤火虫 j 的吸引范围就越大.

由上面分析可知,当萤火虫相对分散(即距离比较远)时,FA 搜索退化为局部随机搜索,丧失全局搜索能力. 要想增加全局搜索能力,需要对光强吸收因子 γ 进行调节. 另外,由 FA 算法可知,萤火虫个体搜索,就是向其周边较优个体靠近,这势必造成萤火虫群体局部聚集. 一旦萤火虫群体发生了局部聚集,萤火虫个体就趋同于局部最优个体,丧失了单个个体的活性. 如果局部最优是全局极值,则能获得全局最优解,但更多的时候,局部最优不一定是全局最优,这样就导致 FA 算法早熟收敛.

下面使用式(9)的四峰函数,来直观地分析 FA 算法的寻优过程.

$$f(x)=e^{-(x-4)^2-(y-4)^2}+e^{-(x+4)^2-(y-4)^2}+2e^{-x^2-(y+4)^2}+2e^{-x^2-y^2} \tag{9}$$

如图 1 所示,四峰函数存在 4 个极大值,坐标分别是(0,−4),(0,0),(−4,4),(4,4),对应的极值分别为 2,2,1,1.

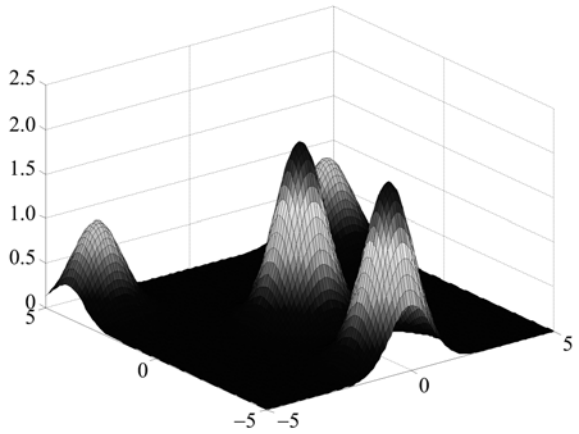

图 1 四峰函数

Fig. 1 Four peaks function

利用标准 FA 算法,对四峰函数进行优化处理. 其中 $\alpha'=0.2$,群体规模 $m=30$,最大迭代次数 $T_{\max}=100$.

第一种情况,设定 $\gamma=1$, x 与 y 范围为[−5,5],结果如图 2 所示. 从图中可以看出,萤火虫个体在其邻域范围内各自收敛到 4 个极值点.

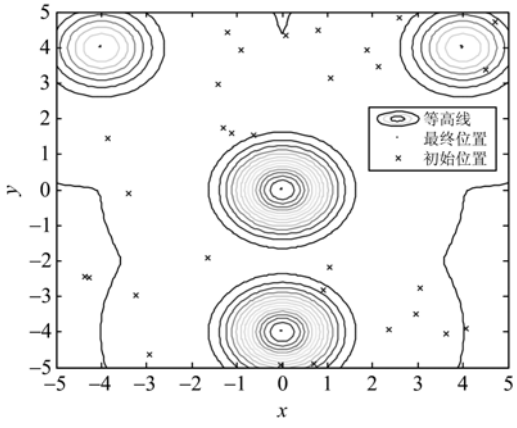

图 2 第一种情况萤火虫位置变化图

Fig. 2 The fireflies' positions shift in 1st case

第二种情况,设定 $\gamma=0.1$, x 与 y 范围为[−5,

5],结果如图 3 所示.这里 γ 值比第一种情况要小,这样萤火虫的吸引范围就越大,结果所有萤火虫个体都收敛到中间极值点.

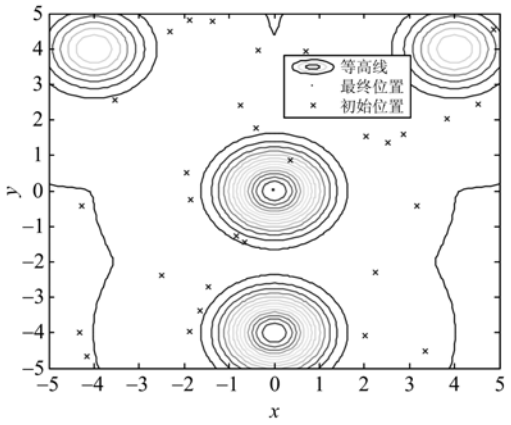

图 3　第二种情况萤火虫位置变化图

Fig. 3　The fireflies' positions shift in 2nd case

第三种情况,设定 $\gamma=0.01$, x 与 y 范围[−100,100],结果如图 4 所示.尽管这里 γ 值已经比较小了,但是由于搜索区间变得比较大,致使萤火虫之间的距离 r 仍然比较大,结果 FA 算法搜索退化成如式(7)那样,在萤火虫个体周边做随机搜索,算法难以收敛.

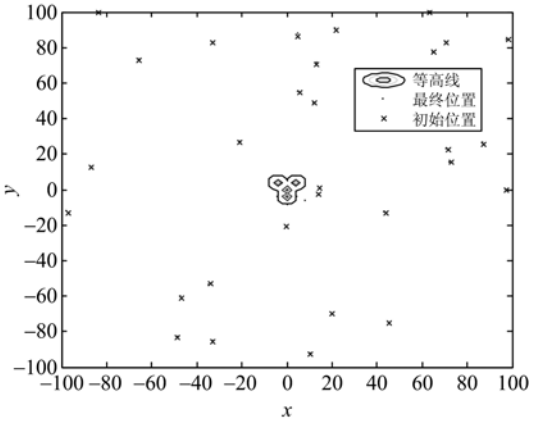

图 4　第三种情况萤火虫位置变化图

Fig. 4　The fireflies' positions shift in 3rd case

第四种情况,设定 $\gamma=1$, x 与 y 初始范围为[0,5],求解范围为[−5,5],结果如图 5 所示.萤火虫群体,受初始范围影响,向较好的个体靠近,最终聚集到右上角局部最优点而无法自拔.

3　保持个体活性的改进 FA 算法

3.1　自适应 γ 调节

由前面的吸引度 $\beta=\beta_0 e^{-\gamma r^2}$ 分析可知,γ 值对萤火虫个体之间的相互吸引起到很大的调节作用.如果 γ 值设置不当,则对 FA 算法的搜索性能影响很大.如何设置 γ 值,标准 FA 算法只是给出了建议,建议 γ 值设置在[0.01,100]区间,通常设置为 1.这种建议对求解区域比较小的时候还适用,如图 2 和 3 那样,但是对于求解空间比较大的情况,特别是高维空间函数求解时,萤火虫个体之间的距离一般非常大,这时采用建议的区间对 γ 取值就会造成 $e^{-\gamma r^2}\to 0$,由式(7)可知,FA 算法退化为局部随机搜索算法,效果如图 4 所示.当 $\gamma=1$, $r\in[0.2,2]$ 时,由 $e^{-\gamma r^2}$ 的图像可知,$e^{-\gamma r^2}$ 有较好的调节效果,由此得到 γ 为任意值,$\gamma r^2\in[0.04,4]$,令 $\gamma_0=\gamma r^2$,则有

$$\gamma=\gamma_0/r^2,\ \gamma_0\in[0.04,4] \tag{10}$$

式中,r 为萤火虫群体距离,可设为当前代数中最差萤火虫位置与最优萤火虫位置之间的距离.这样,γ 的值不同于标准 FA 中那样设置一个固定值,而是根据萤火虫群体的进化而自适应变化,当群体距离比较大时,γ 值就变小,反之 γ 值就变大,以保持萤火虫群体间的相互吸引,起到增强 FA 全局搜索能力的作用.

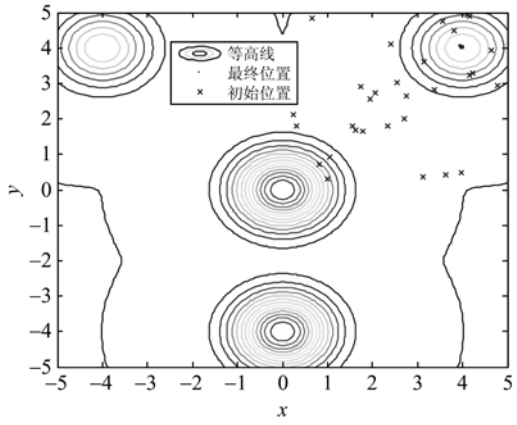

图 5　第四种情况萤火虫位置变化图

Fig. 5　The fireflies' positions shift in 4th case

3.2　位置更新

在标准的 FA 算法中,每一次迭代中,萤火虫 i 向其他萤火虫选择靠近时,算法仅仅根据式(5)改变萤火虫 i 的位置,并没有重新计算改变位置后萤火虫 i 的荧光亮度 I_i,结果造成萤火虫 i 通过迭代快速地向其作用范围内的最优萤火虫个体靠拢,这样就造成过程中萤火虫位置信息没有被利用,既不利于局部搜索,也因为快速收敛而易于早熟收敛,如图 5 所示.为此,我们在每一次迭代过程中,保存萤火虫 i 的新位置,并计算其对应的荧光亮度.在对萤

火虫 i 的每次迭代结束之后，取迭代中搜寻到的最亮萤火虫 i 位置来更新当前萤火虫 i. 部分流程如下：

```
for i=1:n
  for j=1:n
    if(I_j>I_i)
      由式(3)计算 x_i, x_j之间的距离 r_ij.
      由式(4)计算萤火虫 x_j对 x_i的吸引度 β.
      由式(5)计算萤火虫 x_i新位置.
      保存萤火虫 x_i的新位置于向量 x_i0,并计算其对应的
      荧光亮度.
    end if
  end for j
  用向量 x_i0中最亮的萤火虫代替萤火虫 x_i.
end for i
```

3.3 保持个体活性

由上面分析知，标准 FA 算法在搜索过程中因趋同于局部较优的萤火虫，易于发生聚集现象. 一旦聚集，则萤火虫个体因趋同于局部最优个体而失去了个体活性，致使算法搜索能力大为降低，如图 5 所示. 故此，对聚集的个体，需要采用策略来增强个体的活性. 这里，我们通过比较萤火虫之间的距离来判定，当二者的距离小于一给定 ε 值时，则判定为聚集，需要激活. 这里，我们利用式(5)，通过增大 α 的值(给定一个较大值 α_0)，让荧光亮度值较低的个体做较大尺度的变异操作，改变趋同性问题. 操作如下：

```
if r_ij<ε
  x_i(t+1)=x_i(t)+β(x_j(t)-x_i(t))+α_0ε_i
else
  x_i(t+1)=x_i(t)+β(x_j(t)-x_i(t))+αε_i
```

3.4 对最优个体的处理

在标准 FA 算法中，最优个体没有作任何处理，这样导致最优个体自身进化停滞，不利于算法的进一步细化收敛. 文献[8]给出在最优个体附近随机操作的方法，来提高最优个体的进化. 这种方法有一定的效果，但由于是随机的变化，缺乏启发信息. 由前分析知，最优个体附近易于聚集更多的萤火虫个体，并且个体的进化主要靠周边个体间的相互吸引而进行. 如果采用上面保持个体活性的方法，个体的活性增强了，但是由于最优个体附近个体很少，使得最优个体的细化非常慢，甚至停滞不前. 为此，本文在最优个体附近保留一定量萤火虫个体，利用这些个体之间的信息交换来对最优个体位置进行微调，对周边其他个体仍然进行活性增强操作，同时对每一代的最优个体按式(11)进行高斯扰动，以帮助其跳出局部极值点，提高算法全局搜索能力[10].

$$\boldsymbol{x}\_n = \boldsymbol{x}_g * (1 + \mathrm{Gaussian}(\delta)) \tag{11}$$

$$\boldsymbol{x}_g = \begin{cases} \boldsymbol{x}\_n, & \text{if } f(\boldsymbol{x}\_n) < f(\boldsymbol{x}_g) \\ \boldsymbol{x}_g, & \text{others} \end{cases} \tag{12}$$

式中，$\boldsymbol{x}_g$ 为最优个体位置，$\boldsymbol{x}\_n$ 为扰动后的位置.

4 实验结果与分析

为了检验本文提出的保持个体活性的改进萤火虫算法(IFA)性能，选取 10 个标准测试函数进行了实验仿真，并与基本萤火虫算法(FA)、混沌萤火虫算法(CDFA)[9]、粒子群算法(PSO)[16]和人工蚁群算法(ABC)[17]进行了对比.

4.1 实验参数设置

所有测试函数的维数 $d=30$，群体规模 $m=30$，最大迭代次数 $T_{max}=2000$. 萤火虫算法中 $\gamma=1.0$，$\beta_0=1.0$，$\alpha'=2$，delta$=0.99$，另外改进的 IFA 算法中 $\gamma_0=4$，$\alpha_0=100$，最优个体附近保留群规模 1/4 的个体. PSO 算法中 $c1=c2=2$，权重 w 从 0.9 到 0.4 线性变化. ABC 算法中，群体规模 NP$=60$，其中观察蜂与雇佣蜂各占 50%，侦察蜂的 limits$=100$. 各算法的适应度值取为测试函数的值，每个函数独立运行 50 次.

4.2 测试函数

10 个标准测试函数被使用. 这些函数均为复杂的非线性多峰测试函数，存在大量的局部极值区域，可有效检验算法的群体多样性、全局搜索性能、逃离局部极值并避免早熟的能力. 具体函数内容参见表 1.

4.3 结果分析

表 2 给出了不同算法对 10 个测试函数的最优值、最差值、中间值、平均值与标准差. 从表中可以看出，FA 算法的中值在函数 f_1，f_2，f_7，f_8 和 f_9 上要优于 PSO 算法，在函数 f_1，f_8 和 f_{10} 上要优于 ABC 算法，这体现了基本 FA 算法具有良好的整体优化性能，但与其他群智能算法一样，也存在早熟收敛等情况，特别是在函数 f_4，f_5 和 f_6 上优化效果更差. FA 算法在均值方面的统计效果远不如在中值方面，从标准差的变化上也可以看出来，FA 算法的标准差一般比较大，即算法优化效果的波动性比较大. 有如前面的分析，由于 FA 算法参数自适应比较差，影响群体之间的相互吸引，若群体中存有个体靠近

表 1　测试函数

Tab. 1　Benchmark functions

No.	名称	定义	维数	范围	理论值
f_1	Quadric	$f_1=\sum_{i=1}^{d}(\sum_{j=1}^{i}x_j)^2$	30	$[-100,100]$	0
f_2	Tablet	$f_2=10^6x_i^2+\sum_{i=2}^{d}x_i^2$	30	$[-100,100]$	0
f_3	Schaffer	$f_3=\sum_{i=1}^{d-1}(x_i^2+x_{i+1}^2)^{0.25}\times[\sin(50\times(x_i^2+x_{i+1}^2)^{0.1}+1]$	30	$[-2.408,2.408]$	0
f_4	Schwefel	$f_4=418.9829d-\sum_{i=1}^{d}x_i\sin(\lvert x_i\rvert^{1/2})$	30	$[-500,500]$	0
f_5	Griewank	$f_5=\frac{1}{4000}\sum_{i=1}^{d}x_i^2-\prod_{i=1}^{d}\cos(x_i/\sqrt{i})+1$	30	$[-600,600]$	0
f_6	Rastrigrin	$f_6=\sum_{i=1}^{d}(x_i^2-10\cos2\pi x_i+10)$	30	$[-5.12,5.12]$	0
f_7	Rosenbrock	$f_7=\sum_{i=1}^{n}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	30	$[-50,50]$	0
f_8	Sphere	$f_8=\sum_{i=1}^{d}x_i^2$	30	$[-100,100]$	0
f_9	Ackly	$f_9=-20e^{-0.2\sqrt{\sum_{i=1}^{d}x_i^2/d}}-e^{\sum_{i=1}^{d}\cos(2\pi x_i)/d}+20+e$	30	$[-32,32]$	0
f_{10}	Zakharov	$f_{10}=\sum_{i=1}^{d}x_i^2+(\sum_{i=1}^{d}0.5ix_i)^2+(\sum_{i=1}^{d}0.5ix_i)^4$	30	$[-5,10]$	0

表 2　改进 FA 与其他算法的性能对比

Tab. 2　Performance comparison between IFA and other algorithms

函数	算法	最小	最大	中间值	均值	标准差
Quartic	FA	2.69e01	2.48e03	2.52e02	4.20e02	5.17e02
	IFA	**1.54e−33**	**1.52e−16**	**3.54e−25**	**3.96e−18**	**2.18e−17**
	PSO	8.95e02	5.01e03	2.4e03	2.5e03	9.50e02
	ABC	3.06e03	9.92 e03	7.50 e03	7.02 e03	1.82 e03
	CDFA[9]	4.5e−11	7.5e−09	6.6e−10	1.3e−09	1.4e−09
Tablet	FA	4.13e−16	3.93e02	1.27e−05	9.69	5.65e01
	IFA	**2.68e−49**	**1.55e−40**	**1.34e−44**	**4.07e−42**	**2.21e−41**
	PSO	2.09e−06	4.83e−04	4.79e−05	7.99e−05	8.88e−05
	ABC	4.84e−16	7.08e−14	9.20e−16	2.44e−15	9.89e−15
	CDFA[9]	2.2e−26	1.6e−23	2.5e−24	3.9e−24	4.7e−24
Schaffer	FA	10.41	27.94	19.63	19.47	4.25
	IFA	3.29e−13	**1.08e−05**	**6.62e−09**	**4.70e−07**	**1.71e−06**
	PSO	1.7407	12.348	4.9004	5.254	2.17
	ABC	8.72e−04	7.08e−03	3.28e−03	3.51e−03	1.61e−03
	CDFA[9]	**0**	9.15	0.29	1.53	2.27
Schwefel	FA	3.47e03	6.67e03	3.55e03	4.38e03	1.09e03
	IFA	**3.81e−04**	**5.51e−04**	**3.81e−04**	**3.85e−04**	**2.39e−05**
	PSO	2.02e03	5.37e03	3.39e03	3.48e03	7.34e02
	ABC	**3.81e−04**	1.18e02	3.81e−04	2.48	16.75

续表

函数	算法	最小	最大	中间值	均值	标准差
Griewank	FA	13.40	97.65	37.85	40.82	19.17
	IFA	**0**	**2.22e−16**	**0**	**2.22e−17**	**5.49e−17**
	PSO	1.89e−05	8.77e−02	1.01e−02	1.56e−02	1.95e−02
	ABC	2.22e−16	7.39e−03	1.53e−14	2.94e−04	1.35e−03
	CDFA[9]	**0**	6.7e−02	8.6e−03	1.3e−02	1.6e−02
Rastrigrin	FA	29.84	108.45	50.74	53.17	16.24
	IFA	**0**	**0**	**0**	**0**	**0**
	PSO	21.96	62.47	37.80	39.01	9.25
	ABC	**0**	9.23e−14	1.77e−15	5.15e−15	1.37e−14
	CDFA[9]	**0**	3.99	4.0e−12	0.58	1.08
Rosenbrock	FA	14.73	1.29e03	27.39	1.00e02	2.36e02
	IFA	**3.76e−19**	**2.59**	**5.41e−04**	**07.27e−02**	**0.36**
	PSO	2.25e01	2.49e03	1.12e02	2.99e02	4.71e02
	ABC	1.10e−02	4.62	0.14	0.44	0.76
	CDFA[9]	2.3e−03	78.56	8.32	12.28	17.19
Sphere	FA	2.38e−17	28.08	4.27e−17	0.58	3.97
	IFA	**7.62e−88**	**3.04e−46**	**8.07e−60**	**9.80e−48**	**4.97e−47**
	PSO	1.025e−06	4.12e−04	5.60e−05	8.58e−05	9.20e−05
	ABC	4.66e−16	1.13e−15	7.42e−16	7.62e−16	1.33e−16
Ackly	FA	3.90e−09	2.1201	5.17e−09	0.51678	0.7502
	IFA	**8.88e−16**	**4.43e−14**	**9.59e−16**	**1.19e−15**	**5.02e−15**
	PSO	6.04e−04	0.12	3.99e−03	1.07e−02	2.01e−02
	ABC	1.74e−13	7.89e−13	3.38e−13	3.63e−13	1.20e−13
Zakharov	FA	6.2156	138.01	27.441	32.871	23.645
	IFA	**1.61e−19**	**8.33e−15**	**4.14e−17**	**4.59e−16**	**1.55e−16**
	PSO	6.04	34.63	16.31	17.01	5.96
	ABC	1.38e02	3.07e02	2.46e02	2.40e02	3.19e01

最优解，则结果会变好，否则，结果就会较差，从而导致算法波动性较大.

本文提出的改进IFA算法，采用了多种控制策略，增强了算法的自适应能力与多样性控制能力，使得算法在10个测试函数上，均具有良好的搜索精度，特别是对于PSO与ABC等算法难以优化的f_1，f_4，f_7和f_{10}函数，IFA算法也获得非常好的实验结果.在已知的几个测试函数方面，本文提出的IFA算法同样优于文献[9]中提出的CDFA算法.另外，实验结果的标准差进一步表明，IFA算法具有较好的适应性与鲁棒性.

图6～15给出了10个测试函数的寻优曲线.从图中可看出，基本FA算法优化趋势趋缓，特别是在算法中后期，往往呈直线状态，说明基本FA算法易于陷入局部最优而难以自拔，致使算法中后期优化停滞，算法的优化效果一般.从10张图中可看出，本文提出的IFA算法比基本FA算法有着较快的下降速度，说明改进的算法寻优速度快，而且全局搜索能力明显增强.由于采取了γ值自适应调节、过程萤火虫位置更新、聚集萤火虫重新激活以及对最优个体的处理等多种策略，IFA算法增强了群体的多样性与搜索能力，即使在搜索后期，也能寻找到正确的寻优方向，因而取得了良好的搜索效果. 从图上也能明显看出，改进的IFA算法在优化速度与优化精度上，也比经典的PSO与ABC算法来得好.

5 结论

萤火虫算法结构比较简单、易于实现，具有较好的寻优能力.但基本FA算法受求解区域影响，对高维、大空间问题优化效果一般.本文在对FA算法进行分析的基础之上，采用γ值自适应调节、过程萤火虫位置更新、聚集萤火虫重新激活以及对最优个体的处理等多种策略改进FA算法，提出保持个体活性的改进FA算法.通过在10个基准测试函数上的测试，并与基本FA算法、其他改进FA算法，以

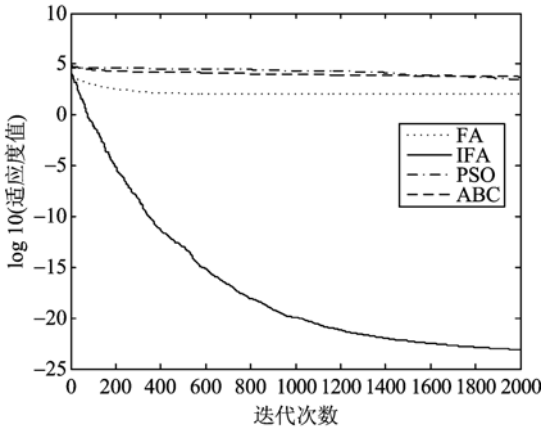

图 6　Quartic 函数收敛曲线

Fig. 6　Convergence process of Quartic function

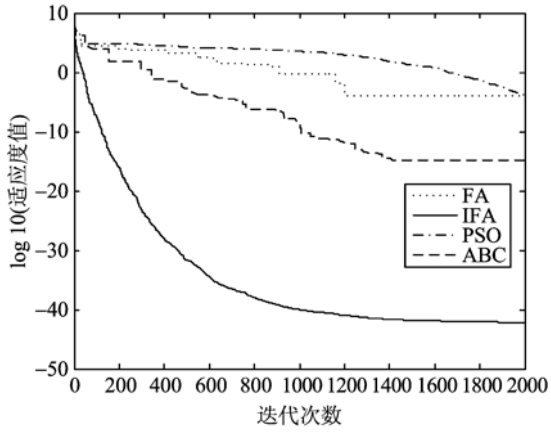

图 7　Tablet 函数收敛曲线

Fig. 7　Convergence process of Tablet function

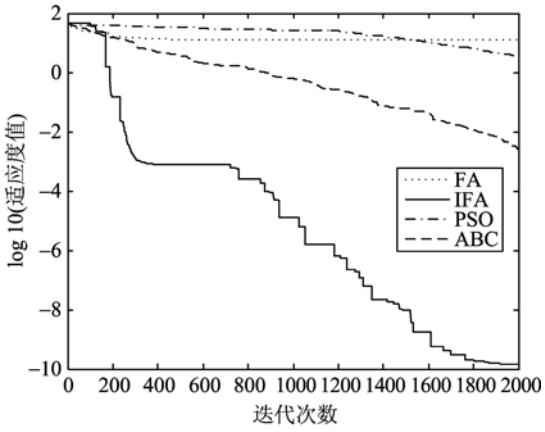

图 8　Schaffer 函数收敛曲线

Fig. 8　Convergence process of Schaffer function

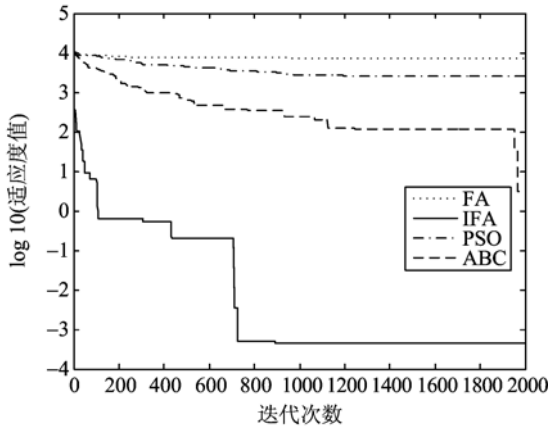

图 9　Schwefel 函数收敛曲线

Fig. 9　Convergence process of Schwefel function

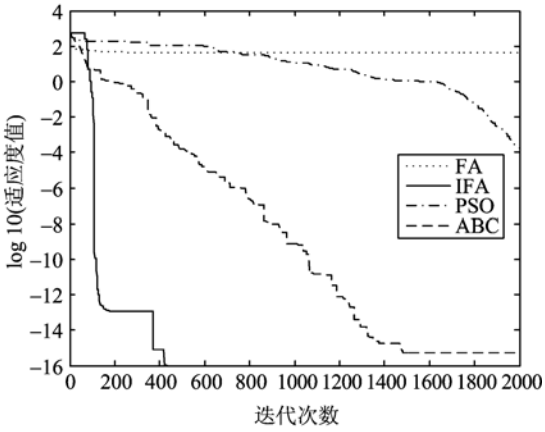

图 10　Griewank 函数收敛曲线

Fig. 10　Convergence process of Griewank function

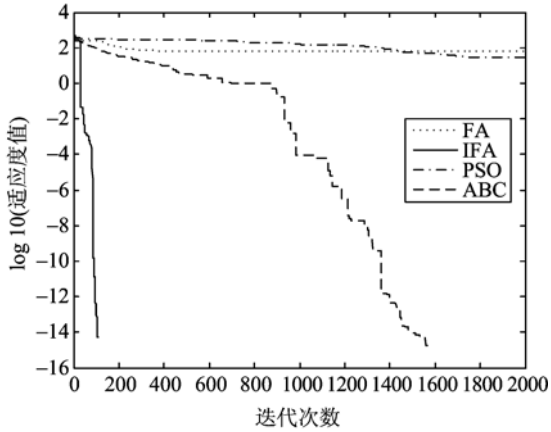

图 11　Rastrigrin 函数收敛曲线

Fig. 11　Convergence process of Rastrigrin function

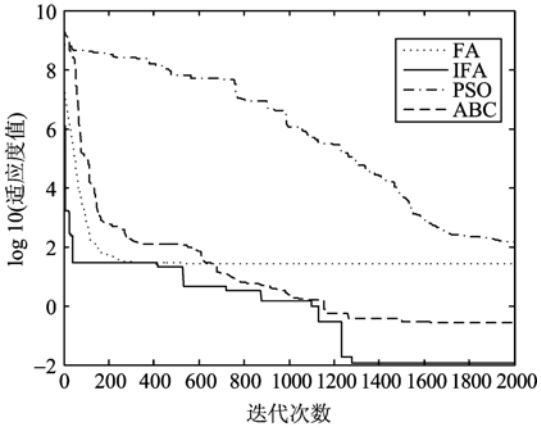

图 12 Rosenbrock 函数收敛曲线

Fig. 12 Convergence process of Rosenbrock function

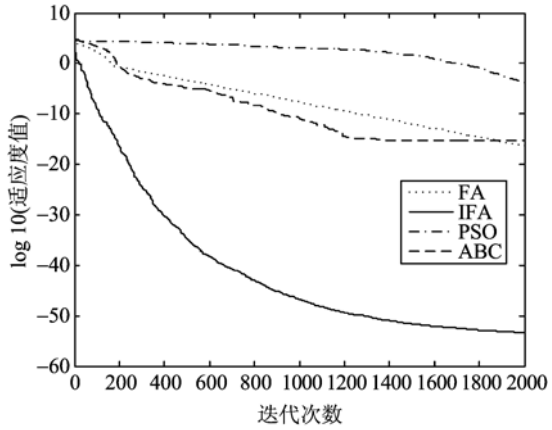

图 13 Sphere 函数收敛曲线

Fig. 13 Convergence process of Sphere function

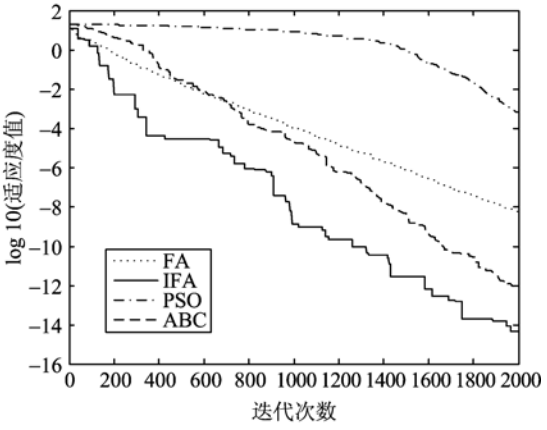

图 14 Ackly 函数收敛曲线

Fig. 14 Convergence process of Ackly function

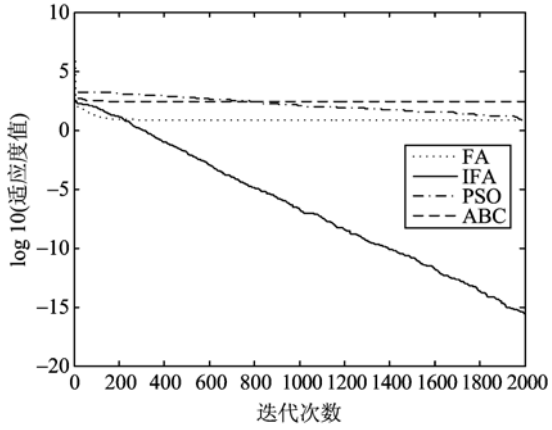

图 15 Zakharov 函数收敛曲线

Fig. 15 Convergence process of Zakharov function

及 PSO 与 ABC 算法进行对比，仿真结果表明，本文提出的 IFA 算法能够很好地保持种群的多样性，具有较快的收敛速度与较好的求解精度，而且算法的适应性与鲁棒性也比较好.

IFA 是在基本 FA 基础上改进而成的，由于 FA 算法存在对两个个体之间的比较，使得算法的时间复杂度比较高，要高于传统的 PSO 算法，与逐维改变的 ABC 算法相当，因而 IFA 算法的时间复杂度也比较高，特别是在改进的 IFA 算法中，需要重新计算改变位置后新萤火虫的荧光亮度，进一步增加了算法的时间复杂度. 另外，在 IFA 算法的后期，寻优速度有所下降，这也表明，依据式(10)进行的参数调节方法以及 α_0 的确定方法还存在调节空间. 下一阶段，我们一方面将致力于各参数的调节控制，同时也将改进算法应用到实际工程优化问题中.

参考文献(References)

[1] Yang X S. Nature-Inspired Metaheuristic Algorithms [M]. Beckington, UK: Luniver Press, 2008.

[2] Yang X S. Multiobjective firefly algorithm for continuous optimization [J]. Engineering with Computers, 2013, 29: 175-184.

[3] 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: 1 180-1 186.

[4] Horng M H. Vector quantization using the firefly algorithm for image compression [J]. Expert Systems with Applications, 2012, 39(1): 1 078-1 091.

[5] Kazem A, Sharifi E, Hussain F K, et al. Support vector regression with chaos-based firefly algorithm for stock market price forecasting [J]. Applied Soft Computing, 2013, 13(2): 947-958.

[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] Coelho L D, Mariani V C. Firefly algorithm approach based on chaotic Tinkerbell map applied to multivariable PID controller tuning[J]. Computers & Mathematics with Applications, 2012,64 (8): 2 371-2 382.

[8] Łukasik S, Żak S. Firefly Algorithm for Continuous Constrained Optimization Tasks[C]// Proceedings of the 1st International Conference on Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems. Springer-Verlag Berlin Heidelberg, 2009:97-106.

[9] Xu Huali, Su Shoubao, Yan Rengrong et al. A firefly algorithm with chaotic diversity control[J]. Journal of University of Science and Technology of China, 2014, 44(7):612-617.
徐华丽,苏守宝,严仍荣,等.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, 44(7):612-617.

[10] Feng yanhong, LiuJianqin, He Yichao. Chaos-based dynamic population firefly algorithm[J]. Journal of Computer Applications,2013, 33(3):796-799,805.
冯艳红,刘建芹,贺毅朝. 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, 805.

[11] Gandomi A H, Yang X S, Talatahari S. Firefly algorithm with chaos[J]. Commun Nonlinear Sci Numer Simulat, 2013,18: 89-98.

[12] Yang X S. Firefly Algorithm, L'evy Flights and Global Optimization[EB/OL]. (2010-03-07)[2014-12-01]. http:// arxiv.org/abs/1003.1464.

[13] Sayadi M K, Hafezalkotob A, Jalali S G. Firefly-inspired algorithm for discrete optimization problems: An application to manufacturing cell formation[J]. Journal of Manufacturing Systems, 2013,32:78-84.

[14] Aruchamy R, Vasantha K D. A comparative performance study on hybrid swarm model for micro array data[J]. International Journal of Computer Applications, 2011,30(6):10-14.

[15] Fister J I, Yang X S, Fister I, et al. Memetic firefly algorithm for combinatorial optimization [C]// Proceedings of the 5th International Conference on Bioinspired Optimization Methods and Their Applications. Piscataway:IEEE Press, 2012:75-86.

[16] Shi Y,Eberhart R C. Empirical study of particle swarm optimization [J]. Congress on Evolutionary Computation, 1999, 3(12):32-49.

[17] Karaboga D. An idea based on bee swarm for numerical optimization [R]. Kayseri, Turkey: Erciyes University, 2005.