

文章编号: 2095-4980(2016)04-0610-05

## 自适应果蝇优化算法

任新涛, 魏五洲, 杨宁国

(中国白城兵器试验中心, 吉林 白城 137001)

**摘要:** 为了进一步提高果蝇优化算法(FOA)的性能, 提出了一种自适应果蝇优化算法(SAFOA), 设计了果蝇搜索群体模型, 给出了一种自适应搜索步长搜索算法。仿真结果表明, 相比 FOA 算法和递减步长果蝇优化算法(DS-FOA), SAFOA 收敛速度较快, 全局搜索与局部寻优能力强, 并能到达高的收敛精确度。

**关键词:** 果蝇优化算法; 自适应搜索步长; 搜索群体; 收敛速度

**中图分类号:** TN911.5

**文献标志码:** A

**doi:** 10.11805/TKYDA201604.0610

## Self-Adaptive Fruit Fly Optimization Algorithm

REN Xintao, WEI Wuzhou, YANG Ningguo

(Baicheng Ordnance Test Center of China, Baicheng Jilin 137001, China)

**Abstract:** A Self-Adaptive Fruit Fly Optimization Algorithm(SAFOA) is proposed in order to further improve the performance of Fruit Fly Optimization Algorithm(FOA). A fruit fly search group pattern is designed, and then a self-adaptive variable-step search algorithm is put forward. Simulation results indicate that SAFOA features fast rate of convergence, strong global search and local optimization performance, and high convergence precision in comparison with FOA and Diminishing Step Fruit Fly Optimization Algorithm(DS-FOA).

**Key words:** Fruit Fly Optimization Algorithm; self-adaptive variable-step; search group; rate of convergence

果蝇优化算法(FOA)是台湾潘文超于 2011 年提出的, 是基于果蝇觅食行为推演出的一种群体智能寻优算法。该算法旨在利用果蝇敏锐的嗅觉和视觉器官, 提高果蝇个体的多样性, 增大果蝇个体搜索空间, 加快个体收敛速度<sup>[1-2]</sup>。与其他智能群体寻优算法相比, 该算法由于实现起来相对较简单, 收敛速度相对较快, 已成为近年来学者研究的热点之一。目前, 常见的寻优智能算法几乎都存在 2 个缺陷: a) 初始寻优时, 收敛速度快, 而趋近于收敛点时的搜索调整比较慢, 不能保证收敛到局部最优点; b) 种群多样性损失过快, 导致算法易于成熟, 不能达到全局最优<sup>[3]</sup>。FOA 也不可避免存在以上 2 点缺陷。为了提高 FOA 的性能, 近年来学者们也提出了一些新的改进算法, 如文献[4]中由 Chunquan LI 等人提出的通过赋予果蝇个体味道浓度判定值一个加性分量, 来提高果蝇的全局寻优能力, 但文中并没有给出加性变量  $\beta$  的精确求法; 文献[5]中宁剑平等提出了 DS-FOA, 提高了 FOA 的局部搜索能力, 但对多极点寻优问题, 此种方法就容易陷入局部寻优。

本文在前人研究的基础上, 提出了一种 SAFOA, 通过设计果蝇群体模型和提出一种自适应搜索步长算法, 实现了整个搜索过程中全局搜索能力和局部搜索能力的优势互补, 加快了算法的收敛速度, 提高了收敛精确度。

### 1 果蝇优化算法

FOA 的思想源于依靠果蝇敏锐的嗅觉, 搜索漂浮在空气中的各种气味, 获得食物源的方向, 飞近食物位置后, 亦可使用视觉发现食物与同伴集聚的位置, 并且往该方向飞去, 直到寻找到食物<sup>[1,6]</sup>。图 1 为果蝇群体迭代搜索食物示意图。

根据图 1 中果蝇群体搜索食物的特性, 可将 FOA 的原理归纳如下<sup>[7-9]</sup>:

第 1 步: 在设定区间随机初始化果蝇群体位置  $(x, y)$ ;

第 2 步: 根据果蝇群体位置坐标, 赋予每个果蝇个体搜索食物的随机方向与距离  $(x_i, y_i)$ ;

第 3 步: 由于无法得知食物位置, 因此先估计果蝇个体与原点之间的距离 ( $Dist_i$ ), 再计算味道浓度判定值 ( $S_i$ ), 此值为距离的倒数;

第 4 步: 将果蝇个体的味道浓度判定值 ( $S_i$ ) 代入味道浓度判定函数 ( $Fitness\ function$ ), 求出该果蝇个体位置的味道浓度 ( $Smelli$ );

第 5 步: 找出果蝇群体中味道浓度最高的果蝇个体 ( $\max(Smell)$ );

第 6 步: 保留最高味道浓度值与对应的果蝇位置  $(x_k, y_k)$ , 此时果蝇群体利用视觉往该位置飞去;

第 7 步: 进入迭代寻优, 重复执行步骤 2~5, 并判断味道浓度是否优于前一迭代味道浓度, 若是则执行步骤 6。

影响 FOA 精确度和收敛速度的因素主要有群体大小、搜索步长和迭代次数。搜索群体越大, 寻优所需的迭代次数越小, 寻优精确度越高, 但计算量越大, 耗时更长; 搜索步长减小, 会加快果蝇算法的局部寻优速度, 但易于陷入局部最优; 增大迭代次数, 一般会提高果蝇算法的寻优精确度, 但同样增加了寻优时间。

下面提出一种在迭代次数相同条件下, 合理设计群体模型和自适应搜索步长的 SAFOA。

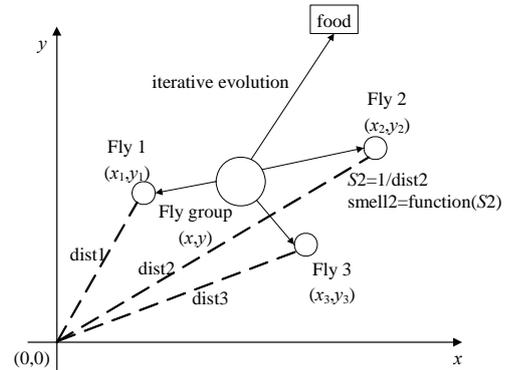


Fig.1 Schematic diagram of fruit fly group's iterative search for food

图 1 果蝇群体迭代搜索食物示意图

## 2 自适应果蝇优化算法

### 2.1 果蝇搜索群体建模

在果蝇群体大小一定的背景下, 为了实现全局搜索能力和局部寻优能力的平衡, 在每次迭代时, 将果蝇搜索群体等量分为 2 个部分, 使其中的一部分按下文的自适应搜索步长算法寻优, 主要作用是加强后期搜索的局部寻优能力, 进而提升算法收敛速度; 另一部分按固定步长寻优, 主要作用是保持算法具有较强的全局搜索能力。

### 2.2 自适应搜索步长算法

#### 1) 固定步长寻优

在随机初始化果蝇群体位置坐标  $(x, y)$  后, 采用式(1)对群体中 1/2 部分的果蝇个体赋予搜索后的位置坐标, 使该部分个体果蝇利用嗅觉搜索食物。

$$\begin{cases} x_i = x + L_r \\ y_i = y + L_k \end{cases} \quad (1)$$

式中  $L_r = L \times \text{rands}(1,1)$ ,  $L_k = L \times \text{rands}(1,1)$  为在固定步长区间  $[-L, L]$  内随机生成的步长值,  $L$  为果蝇个体利用嗅觉搜索的固定步长值。

#### 2) 自适应搜索步长寻优

在随机初始化果蝇群体位置坐标  $(x, y)$  后, 先采用式(1)对群体中另外 1/2 部分的果蝇个体赋予搜索后的位置坐标, 使该部分个体果蝇利用嗅觉搜索食物。

然后找出迭代中整个果蝇群体味道浓度最高的果蝇个体 ( $\max(Smell)$ ), 如果该味道浓度优于前一迭代味道浓度, 记录该果蝇个体的位置坐标  $(x_k, y_k)$ , 则下一次迭代中, 群体中该部分果蝇个体利用嗅觉搜索的步长值按式(2)求解:

$$L' = \frac{|x - x_k| + |y - y_k|}{2} \quad (2)$$

然后采用式(3)对群体中该部分的果蝇个体赋予搜索后的位置坐标, 使该部分个体果蝇利用嗅觉搜索食物。

$$\begin{cases} x_i = x + L' \times \text{rands}(1,1) \\ y_i = y + L' \times \text{rands}(1,1) \end{cases} \quad (3)$$

## 2.3 算法原理

提出的 SAFOA 是基于 FOA 算法, 在每次迭代中, 它将果蝇群体中的个体等量分为 2 组, 按不同的步长进行搜索寻优。自适应搜索步长寻优的那部分果蝇的搜索步长整体上是逐渐减小的, 该部分果蝇个体的局部寻优能力较强, 搜索空间较小; 按照固定步长寻优的那部分果蝇搜索步长较大, 果蝇个体的搜索空间较大, 全局搜索能力较强, 弥补了另一部分果蝇搜索时容易陷入局部最优的不足, SAFOA 流程图如图 2 所示。

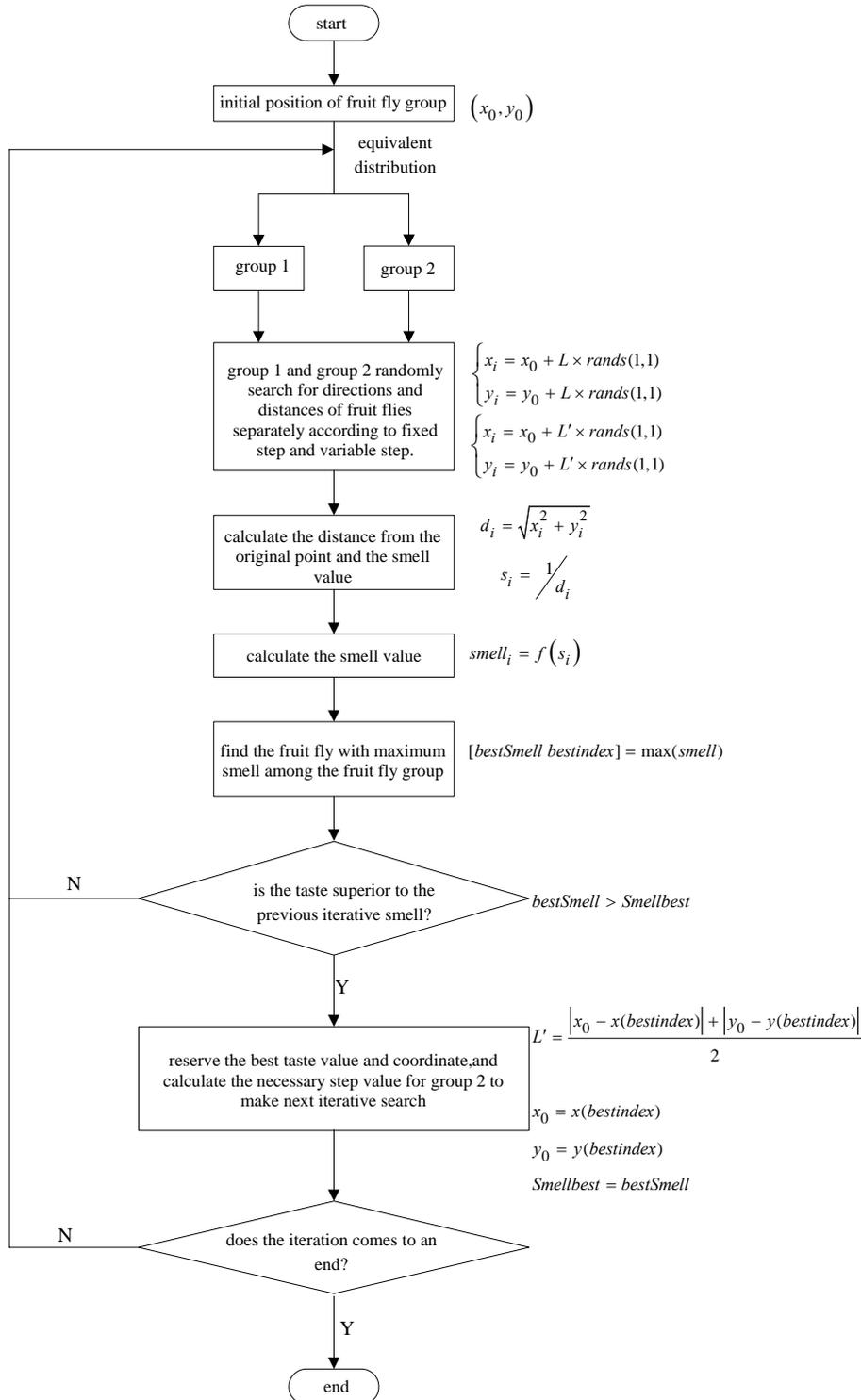


Fig.2 SAFOA flow diagram  
图 2 SAFOA 流程图

### 3 算法验证与结果分析

下面以求解全局最大值为例，研究 SAFOA 的寻优能力。

设置函数为  $y = \frac{30\sin(0.5x-10)}{0.5x-10} + 100$ ，该函数有多个极大值，但最大值仅有 1 个，这样可以辅助验证算法是否存在陷入局部最优的缺陷，其函数曲线如图 3 所示。

#### 3.1 SAFOA 与 FOA 对比

设置的部分仿真参数如下，初始群体位置范围[0,80]，迭代次数 maxgen=500，群体规模 sizepop=60，在迭代过程中，FOA 中果蝇搜索食物的随机飞行方向与距离区间为 [-30,30]；SAFOA 中固定寻优部分果蝇搜索食物的随机飞行方向与距离区间为 [-30,30]，图 4 为经 500 次迭代后的仿真结果。

从图 4 中曲线可以看出，当果蝇搜索步长为 30，采用 FOA 时，陷入局部最优；而采用 SAFOA 的收敛速度相对较快，而且能收敛到全局最优值 130。

#### 3.2 SAFOA 与 DS-FOA 对比

将 SAFOA 与文献[5]中宁剑平等提出的 DS-FOA 法进行比较。设置的部分仿真参数如下，初始群体位置范围[0,80]，迭代次数 maxgen=500，群体规模 sizepop=40，在迭代过程中，SAFOA 中固定寻优部分果蝇搜索步长分别为 30 和 3，DS-FOA 中第 1 次迭代群体搜索步长分别为 30 和 3，图 5 为经 500 次迭代后的仿真结果。表 1 为 SAFOA 与 DS-FOA 的仿真结果统计。

表 1 仿真结果统计表(取 100 次结果)

Table1 Statistical table of simulation results(100 times)

| No. | initial optimization step length | SAFOA convergence statistics              | DS-FOA convergence statistics  |
|-----|----------------------------------|---|--|
| 1   | 30                               | fall into local optimization for 23 times | global convergence   |
| 2   | 3                                | fall into local optimization twice        | all fall into local optimization but global convergence cannot be achieved |

从图 5 和表 1 中的统计数据可以得出，当初始寻优步长等于 30 时，DS-FOA 寻优过程比较稳定，相比 DS-FOA，SAFOA 的收敛速度更快，但收敛稳定性稍差；当初始寻优步长等于 3 时，DS-FOA 陷入局部寻优，无法实现全局收敛，而此时的 SAFOA 收敛速度快且稳定性较好，并能收敛到最优值 130。

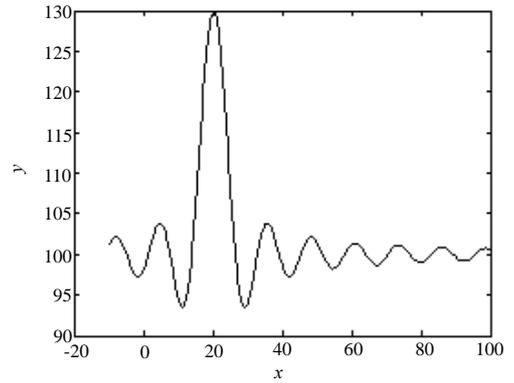


Fig.3 Function curve

图 3 函数曲线

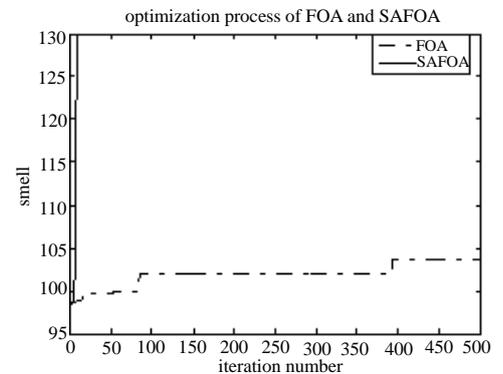
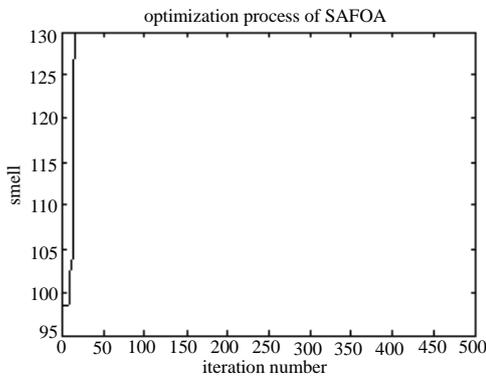
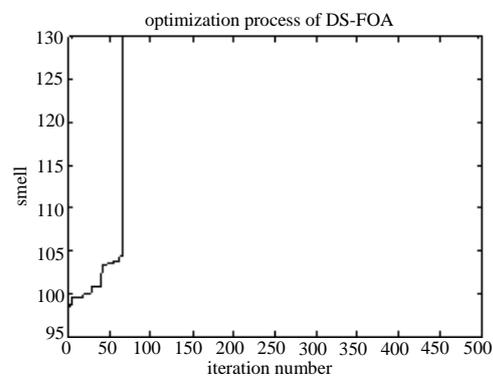


Fig.4 Comparison of FOA and SAFOA optimization

图 4 FOA 与 SAFOA 寻优比较



(a) SAFOA optimization curve(L=30)



(b) DS-FOA optimization curve(L=30)

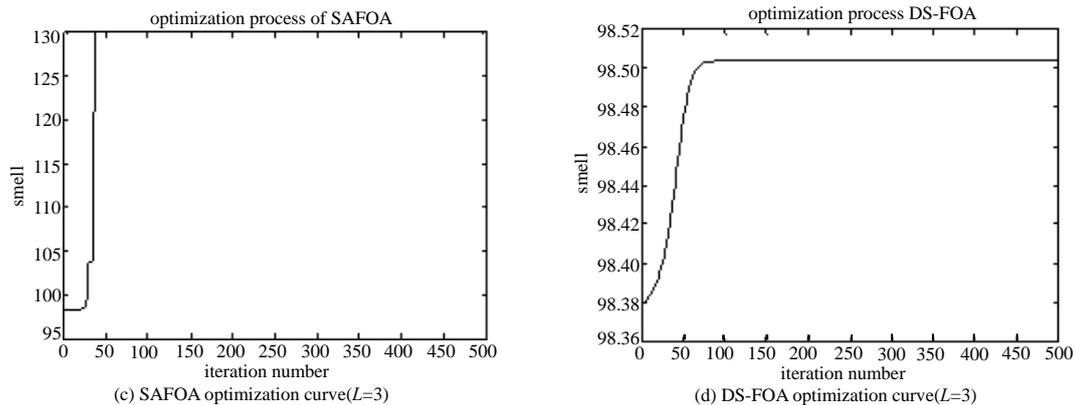


Fig.5 Comparison of SAFOA and DS-FOA optimization  
图 5 SAFOA 与 DS-FOA 寻优比较

#### 4 结论

为了实现 FOA 全局搜索能力和局部寻优能力的优势互补, 通过合理设计群体模型和自适应搜索步长, 提出了一种 SAFOA。通过仿真比较, 得出 SAFOA 的全局收敛能力强, 收敛精确度高且稳定性较好。

#### 参考文献:

- [ 1 ] 潘文超. 果蝇最佳化演算法—最新演化式计算技术[M]. 台中: 沧海书局, 2011. (PAN Wenchao. Fruit Fly Optimization Algorithm: the Financial of Evolutionary Computing Technology[M]. Taizhong, China: Tsang Hai Book Publishing Co., 2011.)
- [ 2 ] 胡能发. 演化式果蝇算法及其应用研究[J]. 计算机技术与发展, 2013, 23(7): 131–137. (HU Nengfa. Evolutionary fruit algorithm and its application research[J]. Computer Technology and Development, 2013, 23(7): 131–137.)
- [ 3 ] 程慧, 刘成忠. 基于混沌映射的混合果蝇优化算法[J]. 计算机工程, 2013, 39(5): 218–221. (CHENG Hui, LIU Chengzhong. Mixed fruit Fly Optimization Algorithm based on chaotic mapping[J]. Computer Engineering, 2013, 39(5): 218–221.)
- [ 4 ] LI Chunquan, XU Shaoping, LI Wei, et al. A novel modified Fly Optimization Algorithm for designing the self-tuning proportional integral derivative controller[J]. Journal of Convergence Information Technology(JCIT), 2012, 7(16): 69–77.
- [ 5 ] 宁剑平, 王冰, 李洪儒, 等. 递减步长果蝇优化算法及应用[J]. 深圳大学学报(理工版), 2014, 31(4): 367–373. (NING Jianping, WANG Bing, LI Hongru, et al. Research on and application of diminishing step fruit fly optimization algorithm[J]. Journal of Shenzhen University(Science and Engineering), 2014, 31(4): 367–373.)
- [ 6 ] 史东亚, 陆键, 陆林军. 基于 RFID 技术和 FOA-GRNN 理论的高速公路道路关闭交通事件对车辆影响的判断模型[J]. 武汉理工大学学报, 2012, 34(3): 63–68. (SHI Dongya, LU Jian, LU Linjun. A judge model of the impact of lane closure incident on individual vehicles on freeways based on RFID technology and FOA-GRNN method[J]. Journal of Wuhan University of Technology, 2012, 34(3): 63–68.)
- [ 7 ] LI Hongze, GUO Sen, ZHAO Huiru, et al. Annual electric load forecasting by a least squares support vector machine with a Fruit Fly Optimization Algorithm[J]. Energies, 2012(5): 4430–4445.
- [ 8 ] 韩俊英, 刘成忠. 自适应变异的果蝇优化算法[J]. 计算机应用研究, 2013, 30(9): 2642–2644. (HAN Junying, LIU Chengzhong. Fruit fly optimization algorithm with adaptive mutation[J]. Application Research of Computers, 2013, 30(9): 2642–2644.)
- [ 9 ] PAN W T. A new Fruit Fly Optimization Algorithm: Taking the financial distress model as an example[J]. Knowledge-Based Systems, 2012, 26: 69–74.

#### 作者简介:



任新涛(1988–), 男, 陕西省兴平市人, 硕士, 工程师, 主要从事雷达信号处理方面的工作. email: renxintao2012@163.com.

魏五洲(1974–), 男, 河南省安阳市人, 硕士, 工程师, 主要从事通信系统与网络技术研究.

杨宁国(1975–), 男, 河北省怀安市人, 硕士, 工程师, 主要从事武器系统鉴定技术研究.