

# 一种离散混合蛙跳算法及其应用<sup>①</sup>



张强, 郭玉洁

(东北石油大学 计算机与信息技术学院, 大庆 163318)

通讯作者: 郭玉洁, E-mail: gyujieii@126.com

**摘要:** 为提高混合蛙跳算法在求解高维复杂函数和离散优化问题的性能, 提出一种离散混合蛙跳算法. 首先, 引入扰动系数来调控青蛙个体的移动距离, 从而更好的平衡迭代中算法的全局探索和局部开发能力; 其次, 利用螺旋更新位置策略使算法能够在最优解附近进行更加精细的搜索; 同时, 采用随机搜索策略, 提高算法的全局搜索能力; 另外, 通过借鉴 2-opt 方法, 实现全局最优解变异, 丰富种群的多样性; 最后, 利用改进的 Sigmoid 函数对个体位置进行离散化处理. 通过对 9 个典型的基准函数和油田措施规划方案的仿真实验表明, 相较于对比的算法, DSFLA 的收敛精度和寻优速度有明显的提升.

**关键词:** 离散混合蛙跳算法; 扰动系数; 螺旋位置更新; Sigmoid; 油田措施规划

引用格式: 张强, 郭玉洁. 一种离散混合蛙跳算法及其应用. 计算机系统应用, 2020, 29(12): 144-153. <http://www.c-s-a.org.cn/1003-3254/7689.html>

## Discrete Shuffled Frog Leaping Algorithm and its Application

ZHANG Qiang, GUO Yu-Jie

(School of Computer and Information Technology, Northeast Petroleum University, Daqing 163318, China)

**Abstract:** A shuffled frog leaping algorithm is proposed to overcome the defects of fall into local optimum easily and the lack of ability to solve discrete optimization problems when solving high-dimensional complex problems. In the proposed algorithm, using the perturbation coefficient to regulate the movement of individual frog distance, so as to better balance the global search and local development capabilities of the algorithm; using the spiral update position strategy to enable the algorithm to perform a more comprehensive and refined search near the optimal solution; using a random search strategy to improve the global search ability of the algorithm; using the 2-opt method to implement the global optimal solution mutation to increase the diversity of the population; the SFLA algorithm is discretized by the improved Sigmoid function. The optimization experiments are conducted on the 9 benchmark functions and oilfield measures planning. Simulation results show that the proposed DSFLA has a better search performance.

**Key words:** DSFLA; perturbation coefficient; spiral position update; Sigmoid; oilfield measures planning

混合蛙跳算法 (Shuffled Frog Leaping Algorithm, SFLA)<sup>[1]</sup> 是 Eusuff 和 Lansey 通过模拟青蛙种群的觅食过程而设计出的一种群智能优化算法, 具有概念简单、参数少且全局寻优能力强等优点. 目前, 许多研究者将其成功应用到不同的工程领域, 雷德明等<sup>[2]</sup> 提

出一种新型蛙跳算法 (SFLA), 用于求解低碳混合流水车间调度问题, 得到高效率的调度方案. 徐俊等<sup>[3]</sup> 将改进后的混合蛙跳算法 (ISFLA) 用于求解云 workflow 调度, 实验证明相同的迭代次数下 ISFLA 优化的效果更好. 张萍等<sup>[4]</sup> 将人工鱼群与蛙跳算法结合起来改进矢量匹

① 基金项目: 国家自然科学基金 (61702093); 黑龙江省自然科学基金 (F2018003)

Foundation item: National Natural Science Foundation of China (61702093); Natural Science Foundation of Heilongjiang Province (F2018003)

收稿时间: 2020-04-15; 修改时间: 2020-05-15; 采用时间: 2020-05-26; csa 在线出版时间: 2020-11-30

配法,实现电压传输函数的有理拟合.陈科尹等<sup>[5]</sup>提出一种改进的混合蛙跳算法,有效解决了传统采摘机器人相机标定方法的不足.蔡宁等<sup>[6]</sup>提出一种融入 Pareto 思想的改进混合蛙跳算法 (SFLA) 用于求解实际资源约束下拆卸线平衡问题,实验证明改进 SFLA 具有良好的综合求解优势.杨哲等<sup>[7]</sup>将改进实数编码混合蛙跳算法 (IR-SFLA) 和二进制编码的 (IB-SFLA) 分别应用到水电站经济负荷分配和机组组合问题,实验证明 IR-SFLA 算法能有效提升水资源利用效率,IB-SFLA 在求解大规模机组电力调度问题时可获得高质量的解.张异<sup>[8]</sup>设计一种求解包装配送问题的混沌蛙跳布谷鸟算法,实验证明 CFLCSA 可以得到高效率的配送策略.艾子义等<sup>[9]</sup>提出一种新型蛙跳算法 (SFLA) 用于求解低碳柔性作业车间调度问题,实验表明新型 SFLA 具有较强的搜索能力和竞争力.李荣波等<sup>[10]</sup>针对标准混合蛙跳算法的不足,提出一种改进混合蛙跳算法 (ISFLA),来求解梯级水库优化调度问题,得到了较好的优化结果.

目前,针对 SFLA 算法已有了许多先进的研究成果,但对于离散混合蛙跳算法的理论研究相对较少,且缺乏相对的实际应用研究.因此,本文提出了一种离散混合蛙跳算法 (Discrete Shuffled Frog Leaping Algorithm, DSFLA),在 DSFLA 算法中引入扰动系数来调控最劣解的跳跃方向,从而协调迭代过程中算法的全局探索和局部开发能力;并将扰动系数作为阈值,引用螺旋更新位置来改进族群内个体的更新策略,精细算法的局部搜索能力;同时为了提高算法的全局探索能力,随机选择种群内的参照个体进行位置更新;此外采用全局最优解变异提高种群的多样性,有助于算法跳出局部最优;最后利用改进的 Sigmoid 函数对 SFLA 进行离散化处理保证了个体的多样性.对改进效果进行仿真验证,并将 DSFLA 与其它 9 种优化算法在求解油田措施规划模型上进行对比,DSFLA 取得了很好的寻优效果.

## 1 离散混合蛙跳算法原理

### 1.1 基本混合蛙跳算法

基本混合蛙跳算法 (SFLA)<sup>[1]</sup> 数学模型思想是:一个湿地里有若干块石头,青蛙可以在石头上跳跃寻找食物最多的点,每只青蛙带有不同的信息,将青蛙种群分为不同的组,通过不同的青蛙个体进行信息交流,实现局部搜索,当局部搜索进行到一定程度时,混合种群中的各组青蛙,实现种群内部的信息交换.根据这一仿

生机制,基本蛙跳算法的寻优过程主要由 4 部分组成.

#### (1) 种群初始化

在湿地中随机生成  $N$  个青蛙组成的初始种群,其中  $d$  维解空间的第  $i$  个青蛙表示为  $X_i = \{x_{i1}, x_{i2}, \dots, x_{id}\}$ ,  $i = 1, 2, \dots, N$ .

#### (2) 划分族群

通过计算种群内青蛙的适应度值,根据该值对青蛙进行降序排列;并将种群划分为  $m$  个族群,每个族群包含  $n$  个青蛙,其中  $N=m \times n$ .

#### (3) 局部搜索

##### ① 基于局部最优解的族群内个体更新策略

将第  $i$  个族群中最优解和最劣解分别标记为  $X_{ib}$ ,  $X_{iw}$ ,  $D$  为跳跃步长,  $[D_{\min}, D_{\max}]$  为跳跃步长的范围,在迭代过程中循环对每个族群中的  $X_{iw}$  进行局部搜索,族群最优解的更新方式如式 (1)、式 (2) 所示.若得到的新解  $X'_{iw}$  的适应度值优于  $X_{iw}$ ,则取代原来族群中的解;否则将进行操作②.

$$D = rand * (X_{ib} - X_{iw}) \quad (1)$$

$$X'_{iw} = X_{iw} + D, \quad D_{\min} \leq D \leq D_{\max} \quad (2)$$

##### ② 基于全局最优解的族群内个体更新策略

将种群中适应度最好的解标记为  $X_g$ ,采用全局最优解  $X_g$  代替  $X_{ib}$  更新  $X_{iw}$ ,全局最优解的更新方式如公式 (3) 和 (4) 所示,若产生的新解  $X'_{iw}$  仍未优于  $X_{iw}$ ;则进行操作③,随机产生一个新解取代  $X_{iw}$ .

$$D = rand * (X_g - X_{iw}) \quad (3)$$

$$X'_{iw} = X_{iw} + D, \quad D_{\min} \leq D \leq D_{\max} \quad (4)$$

##### ③ 随机生成新个体更新策略

$$X'_{iw} = rand * D_{\max} \quad (5)$$

其中,  $rand$  为  $[0, 1]$  随机数组成的向量,  $D_{\max}$  表示跳跃步长的最大边界值.

#### (4) 全局混合

将完成局部搜索后的族群个体重新混合并排序,再次分组和进行族群内部更新,直到算法满足结束条件.

### 1.2 离散混合蛙跳算法原理

#### 1.2.1 族群内个体更新原理改进

基本混合蛙跳算法的族群内个体更新策略主要体现在族群内最劣解不断向最优解的位置方向移动.当迭代到达一定程度后,族群内最优解和全局最优解难以被更新,最劣解朝某一固定方向不断跳跃,易导致算法陷入局部最优.为了提高算法跳出局部最优的能力,

本文引入扰动系数  $A$  来调节个体的跳跃步长, 并将扰动系数  $A$  作为概率阈值, 采用标准 SFLA 族群个体更新位置和螺旋更新位置两种策略来模拟青蛙的觅食行为.

基本混合蛙跳算法的跳跃步长规则如图 1 所示, 可以看出族群中的最劣解  $X_w$  跳跃位置的范围, 被限定在当前位置与族群最优位置之间, 这种跳跃规则限制了算法的搜索空间. 因此, 本文引入扰动系数  $A$  来调控青蛙的跳跃步长. 由式 (7) 可知,  $A$  的取值范围是  $[-\alpha, \alpha]$  中一个随机值, 当  $|A| > 1$  时, 青蛙将扩大跳跃范围, 有更大的可能去寻找潜在优秀解 (改进后的跳跃步长规则如图 2 所示); 当  $|A| \leq 1$  时, 蛙群将缩小跳跃范围在局部区域进行精细搜索. 在迭代初期, 扰动系数的值较大, 保证算法的全局搜索能力, 随着迭代的增加, 扰动系数逐渐减小, 保证算法的局部搜索能力, 从而平衡 SFLA 算法在优化过程中的全局探索和局部开发能力.

$$\alpha = 2 - t * ((2) / T_{max}) \quad (6)$$

$$A = 2\alpha \cdot rand - \alpha \quad (7)$$

$$D = rand * (A * X_{ib} - X_{iw}), \quad D_{min} \leq D \leq D_{max} \quad (8)$$

其中,  $t$  表示当前迭代次数,  $T_{max}$  表示最大迭代次数,  $rand$  为  $[0, 1]$  随机数组成的向量.

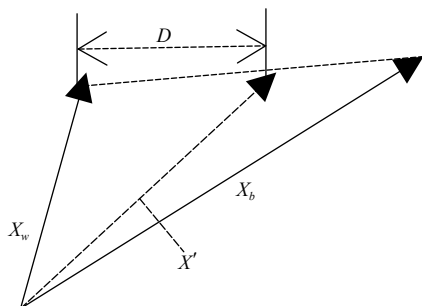


图 1 基本跳跃步长规则

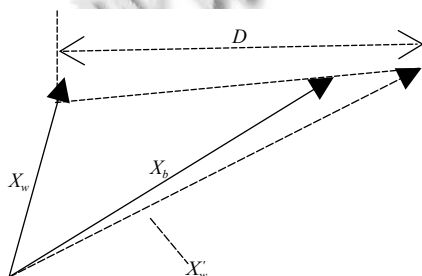


图 2 改进的跳跃步长规则

由进化理论可知优秀个体的周围很大程度上存在更多的潜在优秀个体; 基本 SFLA 算法的迭代后期, 族

群内最优解保持不变, 易导致算法陷入局部最优. 为了提高算法的局部搜索能力, 引用螺旋更新位置策略来改进族群内个体的更新方式, 螺旋更新位置策略使最劣解沿着螺旋运动轨迹向最优解的位置方向移动, 螺旋运动方式能对最优解附近区域进行更加全面精细的搜索, 寻找到更多潜在的优秀解. 本文将引入的扰动系数作为阈值, 当  $|A| \leq 1$  时, 采用螺旋更新位置策略; 当  $|A| > 1$  时, 则采用标准 SFLA 族群内个体更新策略. 其更新公式表达如下:

$$X'_{iw} = \begin{cases} D \cdot e^{bl} \cdot \cos(2\pi l) + X_{ib}, & |A| \leq 1 \\ X_{iw} + D, & |A| > 1 \end{cases} \quad (9)$$

其中,  $l$  为  $[-1, 1]$  随机概率,  $b$  为限定对数螺旋形状的常数 (通常值取 1),  $X_{ib}$  表示族群内最优位置.

### 1.2.2 随机搜索策略

在基本混合蛙跳算法迭代过程中, 随机生成新个体的更新策略具有一定的混乱性和无序性, 不利于算法的收敛. 因此本文通过随机选择种群中的一个青蛙个体作为参照青蛙, 以此来寻找其它更优的解. 这样既实现了种群内部信息的充分交流, 同时增强了算法的搜索能力, 使该算法能在全局范围内进行搜索. 其更新公式表达如下:

$$D = rand * (X_{rand} - X_{iw}) \quad (10)$$

$$X'_{iw} = X_{iw} + D, \quad D_{min} \leq D \leq D_{max} \quad (11)$$

其中,  $X_{rand}$  表示一个随机位置向量,  $rand$  为  $[0, 1]$  随机数组成的向量.

### 1.2.3 全局最优解变异

全局最优解作为重要的信息源会影响整个群体的走势, 但在种群迭代后期, 全局最优解难以被更新, 易导致算法停滞. 因此, 本文借鉴 2-opt 方法对全局最优解进行变异, 通过让种群内邻域解参与到信息交互中, 从而增加种群的多样性, 有助于算法在迭代后期跳出局部最优. 其更新公式表达如下:

$$\begin{aligned} X_g &= \{x_1, x_2, \dots, x_m, x_{m+1}, \dots, x_n, x_{n+1}, \dots, x_d\} \\ &\quad \downarrow \\ X_g &= \{x_1, x_2, \dots, x_{n+1}, x_n, \dots, x_{m+1}, x_m, \dots, x_d\} \end{aligned} \quad (12)$$

其中,  $X_g$  表示全局最优位置,  $d$  表示优化空间的维度.

根据文献 [11] 提出的通过计算种群的多样性度量值来评测种群多样性的方法, 在 DSFLA 算法中引入全局最优解变异策略, 利用式 (13) 计算的其种群的多样性度量值, 并将结果与标准 SFLA 算法做对比, 对比结果如图 3 所示.



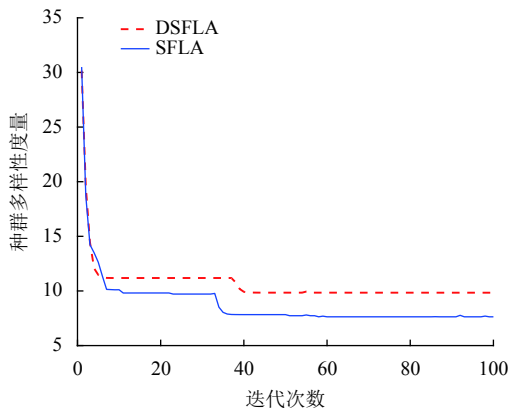


图3 种群多样性度量

$$Q(t) = \frac{\sum_{i=1}^N \sqrt{\sum_{j=1}^D (x_{ij}^t - \bar{X}_j)^2}}{N * T_{\max}} \quad (13)$$

其中,  $T_{\max}$  表示最大迭代次数,  $N$  表示种群规模,  $D$  表示数据维数,  $x_{ij}^t$  表示迭代  $t$  次第  $i$  个种群的第  $j$  个分量,  $\bar{X}_j$  表示所有种群的第  $j$  个分量的平均值. 上述间距公式,  $Q(t)$  表示种群多样性度量值, 其值越大表示种群平均间距越大, 种群多样性越大.

由图3可知, 随着迭代次数的不断增加, DSFLA 算法种群多样性度量值逐渐趋向于比 SFLA 算法较大的一个值, 因此 DSFLA 算法的种群多样性更大.

#### 1.2.4 离散化原理

由于实际工程应用中所计算的变量都是离散的, 因此, 本文提出一种离散二进制蛙跳算法编码方式对青蛙位置进行离散化处理. 为了保证迭代过程中青蛙的位置只能取 0 或 1, 通常使用 Sigmoid 函数<sup>[12]</sup> 将位置压缩到  $[0, 1]$  区间, 其公式如式 (14)、式 (15) 所示, 式 (14) 中的矩阵运算需用到“./”. 利用标准 Sigmoid 函数对青蛙个体位置映射的结果如图4所示, 可以看出, 映射结果大都集中在 0.35~0.65 之间, 若直接用标准 Sigmoid 函数对映射结果离散化处理, 则个体容易产生“靠拢”现象, 会导致算法陷入局部最优, 过早停滞.

$$s(x) = 1./(1 + \exp(-x)) \quad (14)$$

$$x = \begin{cases} 1, & \text{rand} < s(x) \\ 0, & \text{otherwise} \end{cases} \quad (15)$$

为了提高算法的搜索能力, 本文通过对最优位置和当前位置的差值进行映射, 同时建立跳跃步长  $D$  与位置转换概率间的关系来改进 Sigmoid 函数; 映射结果如图5所示, 可以看出, 映射结果值分布在 0~1 之间,

更好的保证了离散化后个体的多样性, 为算法的优化过程奠定了多样性基础. 改进后的 Sigmoid 函数公式表达如下:

$$x = 2 * \text{rand}(\text{VarSize}).*(x_g - x) \quad (16)$$

$$s(x) = |2./(1 + \exp(-x)) - 1| \quad (17)$$

$$x = \begin{cases} 1, & D \geq 0, \text{rand} \leq s(x) \\ 0, & D < 0, \text{rand} \leq -s(x) \end{cases} \quad (18)$$

其中,  $x_g$  表示当前最优位置,  $x$  表示位置向量,  $D$  表示跳跃步长,  $\text{VarSize}$  表示维数大小. “./”和“.\*”表示矩阵运算, 在计算中保持矩阵维度一致.

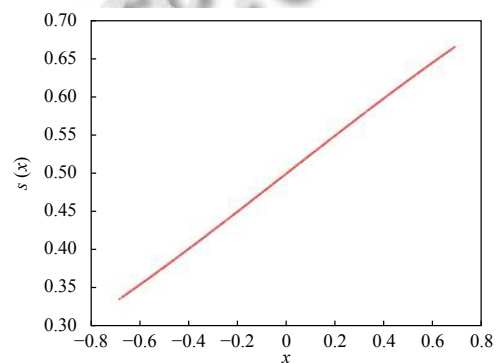


图4 标准 Sigmoid 函数

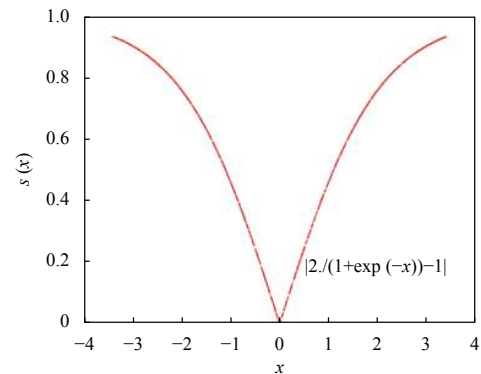


图5 改进 Sigmoid 函数

### 1.3 离散混合蛙跳算法

离散混合蛙跳算法 (DSFLA) 的具体操作步骤如算法1所示.

#### 算法1. 离散混合蛙跳算法

1. 初始化参数 (种群规模  $N$ , 族群数  $m$ , 维数  $D$ , 局部搜索次数  $L$ , 最大迭代次数  $T_{\max}$ );
2. 随机生成初始青蛙种群;
3. for  $t=1:T_{\max}$
4. 应用式 (6)、式 (7) 更新参数  $\alpha, A$ , 计算每个个体  $x_i$  的适应度值  $F$ , 按照适应度值降序排列, 并进行族群划分;
5. for  $j=1:L$

```

6. for i=1:m
7.     计算第 i 个族群的局部最优解  $x_{ib}$ , 局部最差解  $x_{iw}$ , 全局最优解  $x_g$ ;
8. if  $|A| \leq 1$ 
9.     应用式 (6)~式 (9) 得到新的  $x'_{iw}$ ;
10. else
11.     应用式 (6)~式 (9) 得到新的  $x'_{iw}$ ;
12. end if
13. 应用式 (15)~式 (17) 对更新后的个体  $x'_{iw}$  进行离散化处理, 得到新的  $x'_{iw}$ ;
14. if  $F(x'_{iw})$  better than  $F(x_{iw})$ 
15.      $x_{iw} = x'_{iw}$ 
16. else
17.     应用式 (3)、式 (4) 得到新的  $x'_{iw}$ ;
18.     应用式 (15)~式 (17) 对更新后的个体  $x'_{iw}$  进行离散化处理, 得到新的  $x'_{iw}$ ;
19. if  $F(x'_{iw})$  better than  $F(x_{iw})$ 
20.      $x_{iw} = x'_{iw}$ 
21. else
22.     应用式 (10)、式 (11) 得到新的  $x'_{iw}$ ;
23.     应用式 (15)~式 (17) 对更新后的个体  $x'_{iw}$  进行离散化处理, 得到新的  $x'_{iw}$ ;
24.      $x_{iw} = x'_{iw}$ 
25. end if

```

```

26. end if
27. end for
28. end for
29. 应用式 (12) 对全局最优解  $x_g$  进行变异, 得到新的  $x'_g$ ;
30. if  $F(x'_g)$  better than  $F(x_g)$ 
31.      $x_g = x'_g$ ;
32.      $F(x_g) = F(x'_g)$ ;
33. end if
34. end for

```

## 2 实验仿真测试

本文选用了 9 个常用的基准测试函数<sup>[13]</sup> 如表 1 所示. 将离散混合蛙跳算法 (DSFLA), 同基本混合蛙跳算法 (SFLA)<sup>[11]</sup>, 鲸鱼优化算法 (WOA)<sup>[14]</sup>, 灰狼算法 (GWO)<sup>[15]</sup>, 飞蛾扑火算法 (MFO)<sup>[16]</sup>, 布谷鸟算法 (CSA)<sup>[17]</sup>, 蝙蝠算法 (BAT)<sup>[18]</sup>, 多元宇宙算法 (MVO)<sup>[19]</sup>, 粒子群算法 (PSO)<sup>[20]</sup>, 乌贼算法 (CFA)<sup>[21]</sup> 9 种优化算法进行实验对比. 参数设置如表 2 所示, 最大迭代次数为 100, 种群数为 60, 重复 10 次, 采用平均值 (mean), 标准差值 (std) 以及迭代中的最优值 (best) 来评价算法的性能.

表 1 基准测试函数

名称	测试函数	公式	初始范围	维数	最优值
F1	Sphere	$\min \sum_{i=1}^D x_i^2$	[-100, 100]	100	0
F2	Rotated	$\min \sum_{i=1}^D \left( \sum_{j=1}^i x_j \right)^2$	[-100, 100]	100	0
F3	Quartic	$\min \sum_{i=1}^D ix_i^4 + random[0, 1)$	[-1.28, 1.28]	100	0
F4	Rosenbrock	$\min \sum_{i=1}^D \left[ 100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2 \right]$	[-30, 30]	100	0
F5	Rastrigrin	$\min 10 * D + \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i)]$	[-5.12, 5.12]	100	0
F6	Griewank	$\min \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-600, 600]	100	0
F7	Step	$\min \sum_{i=1}^D ( x_i + 0.5 )^2$	[-100, 100]	100	0
F8	Ackley	$\min -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + \exp(1)$	[-32, 32]	100	0
F9	Schwefel 2.21	$\min \{ x_i , 1 \leq i \leq D\}$	[-10, 10]	100	0

表 3 为 10 种优化算法在 9 个测试函数中的实验结果对比. 图 6~图 14 为对应的迭代图. 以最优值和平均值为评判标准, 通过比较 10 种算法的优化结果可知, 在函数参数条件设定相同的情况下, DSFLA 寻优结果最好. 上述实验结果主要原因是 DSFLA 引入扰动系数更新策略来随机选取族群最优解的位置作为最差解的

跳跃方向, 可以更好的平衡算法在不同迭代时期的全局搜索和局部开发能力, 使青蛙可以更快的靠近食物; 同时将扰动系数作为阈值, 引用螺旋更新位置来改进族群内个体更新策略, 对最优解位置附近进行精细搜索, 使青蛙能够更准确的找到食物的位置, 从而提高了算法局部搜索的精度.

以方差为评判标准, DSFLA 的收敛精度和方差都较优于其它 9 种算法. 实验结果主要原因是在迭代过程中, 其它个体会逐渐集中在最优个体的位置附近, 导致算法过早停滞; DSFLA 算法通过全局最优解变异的方式增加种群的多样性, 避免青蛙个体全部聚集在某个局部最优位置; 同时采用随机搜索策略, 提高算法的全局搜索能力.

综上所述, 针对标准混合蛙跳算法在求解高维复杂函数时存在收敛精度低和寻优速度慢等问题, 本文采用的改进的策略能够明显提高了 DSFLA 算法的收敛速度和寻优性能.

表 2 算法参数设置

算法	参数设置
DSFLA	$m = 20, L = 20, b = 1$
SFLA	$m = 20, L = 20$
WOA	$b = 1$
GWO	$w = 0.5, wdamp = 0.99, c1 = 1, c2 = 2$
MFO	$b = 1$
CSA	$pa = 0.25, \alpha = 1$
BAT	$A = 0.5, r = 0.5, Qmin = 0, Qmax = 2$
MVO	$WEP\_Max = 1, WEP\_Min = 0.2, p = 0.6$
PSO	$c1 = 1, c2 = 2, w = 0.5, wdamp = 0.99$
CFA	$r1 = 2, r2 = -1, v1 = -1.5, v2 = 1.5$

表 3 实验结果对比表

名称	指标	DSFLA	SFLA	WOA	GWO	MFO	CSA	BAT	MVO	PSO	CFA
F1	Mean	5.15E+03	2.14E+04	1.48E+04	2.41E+04	2.13E+05	2.65E+05	2.47E+05	7.57E+04	4.32E+04	8.53E+04
	Std	3.14E+04	2.33E+04	5.31E+04	5.44E+04	3.34E+04	1.76E-10	2.62E+04	7.23E+04	3.02E+04	1.22E+05
	Best	<b>4.67E-90</b>	1.53E+04	7.35E-12	5.05E+01	1.55E+05	2.65E+05	1.87E+05	5.62E+03	2.67E+04	4.70E+00
F2	Mean	1.47E+04	2.59E+05	9.83E+05	1.78E+05	7.62E+05	1.38E+06	1.15E+06	3.45E+05	2.84E+05	8.84E+05
	Std	1.04E+05	1.46E+05	9.28E+04	2.16E+05	2.55E+05	3.46E+04	8.17E+04	2.40E+05	1.89E+05	1.74E+05
	Best	<b>1.61E-83</b>	2.21E+05	9.54E+05	2.85E+04	5.54E+05	1.34E+06	1.10E+06	1.59E+05	1.51E+05	7.20E+05
F3	Mean	2.22E+01	2.16E+01	5.28E+01	1.46E+02	1.13E+03	1.61E+03	6.60E+02	3.92E+02	1.02E+02	1.72E+03
	Std	1.92E+02	1.86E+02	2.49E+02	3.88E+02	2.97E+02	2.84E-01	4.02E+02	5.18E+02	2.22E+02	1.42E+01
	Best	<b>3.99E-05</b>	6.12E-01	5.86E-03	1.68E-01	6.99E+02	1.61E+03	3.81E+02	1.58E+01	3.62E+01	1.71E+03
F4	Mean	9.75E+06	1.54E+07	3.38E+07	6.31E+07	9.75E+08	1.12E+09	6.27E+08	2.24E+08	4.63E+07	1.07E+09
	Std	9.70E+07	1.06E+08	1.78E+08	1.78E+08	1.18E+08	3.47E+06	3.37E+08	3.43E+08	1.04E+08	2.00E+07
	Best	<b>9.80E+01</b>	3.34E+06	9.84E+01	2.38E+03	7.44E+08	1.12E+09	1.68E+08	1.88E+06	2.01E+07	1.05E+09
F5	Mean	5.80E+01	8.43E+02	1.85E+02	6.49E+02	1.46E+03	1.59E+03	1.23E+03	1.30E+03	9.11E+02	1.98E+02
	Std	2.59E+02	8.37E+01	4.03E+02	3.76E+02	1.14E+02	4.97E+00	1.38E+02	1.32E+02	1.03E+02	4.56E+02
	Best	<b>0</b>	8.11E+02	6.82E-13	1.93E+02	1.29E+03	1.59E+03	1.05E+03	9.58E+02	7.95E+02	5.75E-06
F6	Mean	7.00E+01	7.98E+02	1.10E+02	1.93E+02	2.31E+03	2.40E+03	2.51E+03	7.53E+02	4.00E+02	4.80E+02
	Std	3.01E+02	2.00E+02	4.16E+02	4.72E+02	2.27E+02	5.59E+00	4.40E+01	7.27E+02	3.11E+02	8.97E+02
	Best	<b>0</b>	6.96E+02	1.41E-13	1.26E+00	1.88E+03	2.40E+03	2.42E+03	6.91E+01	2.18E+02	1.12E-03
F7	Mean	4.84E+03	2.12E+04	1.49E+04	2.17E+04	2.21E+05	2.68E+05	2.21E+05	6.81E+04	3.98E+04	3.66E+04
	Std	3.36E+04	2.56E+04	4.95E+04	5.01E+04	2.71E+04	1.88E+03	2.45E+04	7.27E+04	2.85E+04	8.69E+04
	Best	<b>2.44E+01</b>	1.81E+04	8.62E+00	5.85E+01	1.80E+05	2.65E+05	1.66E+05	4.77E+03	2.40E+04	2.99E+01
F8	Mean	7.67E-01	1.29E+01	3.59E+00	7.69E+00	2.06E+01	2.08E+01	2.02E+01	1.62E+01	1.61E+01	2.04E+01
	Std	3.16E+00	9.46E-01	6.35E+00	6.13E+00	1.96E-01	3.34E-02	2.20E-01	3.63E+00	1.38E+00	1.94E-01
	Best	<b>8.88E-16</b>	1.26E+01	4.44E-07	2.79E+00	2.03E+01	2.08E+01	1.99E+01	1.04E+01	1.49E+01	2.02E+01
F9	Mean	2.18E+00	4.28E+01	9.07E+01	4.92E+01	9.44E+01	9.43E+01	9.07E+01	8.38E+01	5.28E+01	9.43E+01
	Std	1.10E+01	5.20E+00	1.13E+00	2.74E+01	1.02E+00	1.86E-13	1.24E+00	8.79E+00	1.17E+01	5.87E-01
	Best	<b>3.85E-43</b>	4.20E+01	9.03E+01	1.94E+01	9.33E+01	9.43E+01	8.97E+01	6.86E+01	4.44E+01	9.41E+01

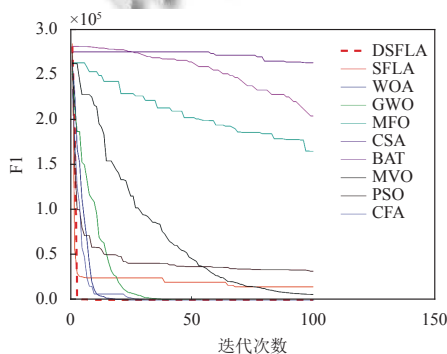


图 6 F1 迭代图

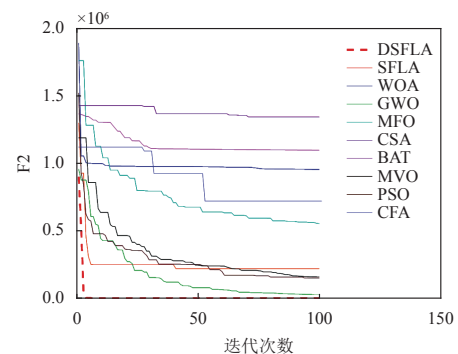


图 7 F2 迭代图

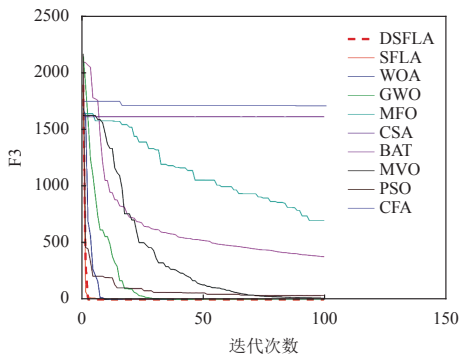


图8 F3 迭代图

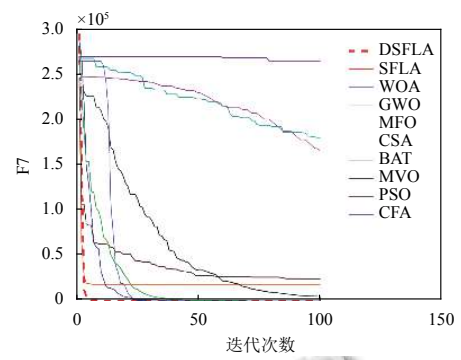


图12 F7 迭代图

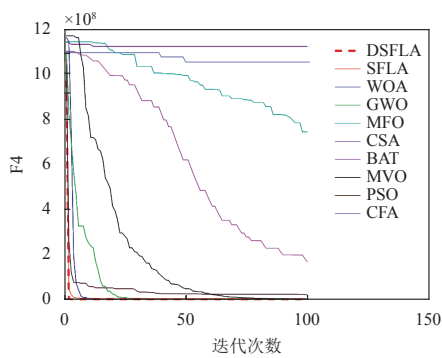


图9 F4 迭代图

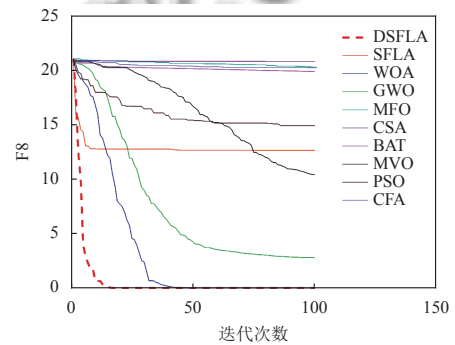


图13 F8 迭代图

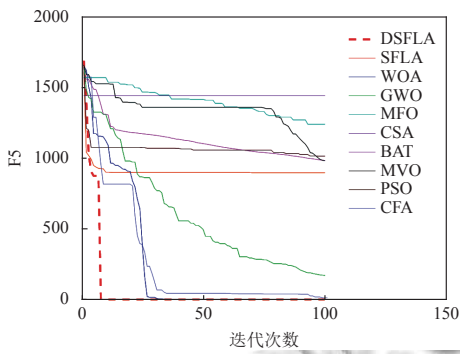


图10 F5 迭代图

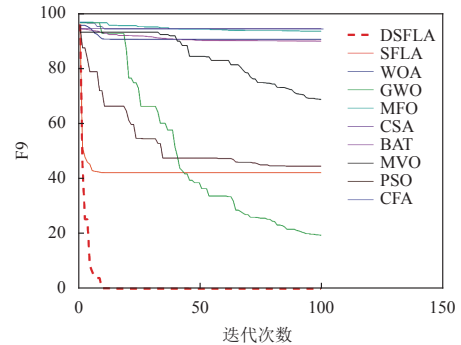


图14 F9 迭代图

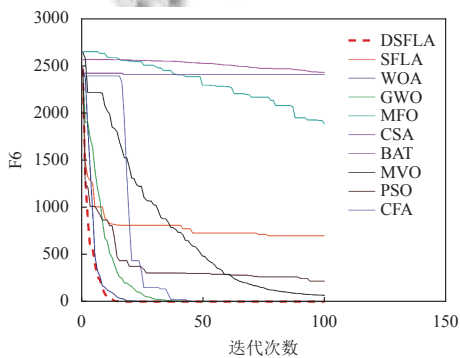


图11 F6 迭代图

### 3 基于 DSFLA 的油田措施方案优化

本文以年度利润和累积利润投入比最大为目标函数,建立油田措施规划模型,通过和其它9种算法优化结果进行对比,分析 DSFLA 在优化油田措施规划中的性能。

#### 3.1 实验油田措施规划数学模型

##### (1) 目标函数

油田措施规划问题是在满足油田开采的约束条件的前提下,分配开采井组,使油田的年度利润和累积利润投入比最大,其目标函数如下:

$$\max F_1 = \frac{\sum_{i=1}^N \sum_{j=1}^K e^{-dt} * (x1_{ij} + y1_{ij}) * a - \sum_{i=1}^N \sum_{j=1}^K z1_{ij} * b - \sum_{i=1}^N \sum_{j=1}^K w1_{ij} * c}{\sum_{i=1}^N \sum_{j=1}^K e^{-dt} * (x1_{ij} * price_{ij} + y1_{ij} * price_{ij})} \quad (19)$$

$$\max F_2 = \frac{\sum_{i=1}^N \sum_{j=1}^K e^{-dt} * (x2_{ij} + y2_{ij}) * a - \sum_{i=1}^N \sum_{j=1}^K z2_{ij} * b - \sum_{i=1}^N \sum_{j=1}^K w2_{ij} * c}{\sum_{i=1}^N \sum_{j=1}^K e^{-dt} * (x2_{ij} * price_{ij} + y2_{ij} * price_{ij})} \quad (20)$$

式中,  $F_1$ 是年度利润,  $F_2$ 是累积利润,  $N$ 为井数目,  $K$ 为措施数,  $a, b, c$ 分别是吨油成本、吨水成本、吨液成本,  $d$ 为递减率,  $t$ 为开发时间,  $x1_{ij}$ 为第  $i$  口油井第  $j$  个措施的年度增油量,  $y1_{ij}$ 为第  $i$  口水井第  $j$  个措施的年度增油量,  $z1_{ij}$ 为第  $i$  口井第  $j$  个措施的年度增液量,  $w1_{ij}$ 为第  $i$  口井第  $j$  个措施的年度增注量,  $x2_{ij}$ 为第  $i$  口油井第  $j$  个措施的累积增油量,  $y2_{ij}$ 为第  $i$  口水井第  $j$  个措施的累积增油量,  $z2_{ij}$ 为第  $i$  口井第  $j$  个措施的累积增液量,  $w2_{ij}$ 为第  $i$  口井第  $j$  个措施的累积增注量,  $price_{ij}$ 是第  $i$  口井措施数为  $j$  的价格。

(2) 约束条件

1) 措施增油量约束

$$\sum_{i=1}^N \sum_{j=1}^K e^{-dt} * x1_{ij} = t_o \quad (21)$$

$$0 \leq \sum_{i=1}^N \sum_{j=1}^K e^{-dt} * y1_{ij} - t_e \leq t_e * \delta \quad (22)$$

式中,  $t_o$ 是油井增油目标, 取值为  $7.1 \times 10^4 t$ ,  $t_e$ 是水井增油目标, 取值为  $7.5 \times 10^4 t$ ,  $\delta$ 为误差百分比, 取值为 0.03。

2) 措施增液量和增注量约束

$$\left| t_l - \sum_{i=1}^N \sum_{j=1}^K z1_{ij} \right| \leq t_l * \delta \quad (23)$$

$$\left| t_i - \sum_{i=1}^N \sum_{j=1}^K w1_{ij} \right| \leq t_i * \delta \quad (24)$$

式中,  $t_l$ 是增液量上限, 取值为  $18 \times 10^4 t$ ,  $t_i$ 是增注量上限, 取值为  $30 \times 10^4 m^3$ 。

3) 措施数上限约束

$$0 \leq \sum_{i=1}^N \sum_{j=1}^K K_{ij} \leq Limit_{ij} \quad (25)$$

式中,  $Limit_{ij}$ 是采油井的措施数上限。

3.2 实验分析

为验证所提算法在求解油田措施规划模型中的性能, 本章采用 10 种优化算法针对 2490 口措施井进行方案分配。算法的参数设置如表 1 所示, 最大迭代次数为 1000, 种群大小为 100, 实验结果如图 15, 图 16 所示。图 15, 图 16 表示的是 10 种优化算法迭代情况图。表 4, 表 5 表示 10 种优化算法得到的最优值、最差值、平均值和标准差。

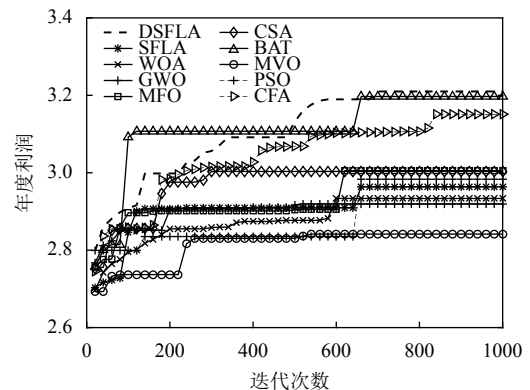


图 15 年度利润对比图

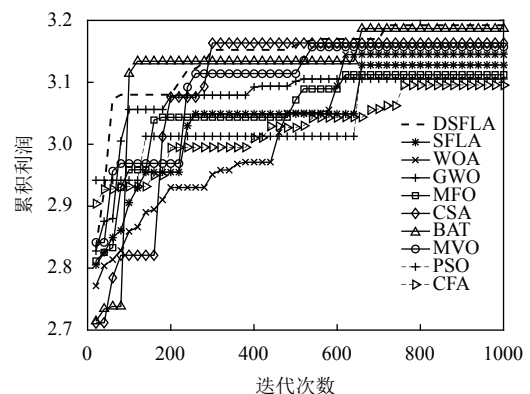


图 16 累积利润对比图



表4 年度利润  $F_1$  算法结果对比

算法	最优值	最差值	平均值	标准差
DSFA	<b>3.21606</b>	2.79079	3.12455	0.11251
SFLA	2.96366	2.70348	2.91201	0.06387
WOA	2.93361	2.69699	2.88064	0.05713
GWO	2.91966	2.76671	2.90086	0.03366
MFO	3.00531	2.7562	2.93435	0.06573
CSA	3.00331	2.75187	2.97180	0.06521
BAT	3.19702	2.75711	3.11430	0.10306
MVO	2.84151	2.69295	2.81320	0.04596
PSO	2.98277	2.79898	2.88420	0.07480
CFA	3.15099	2.74461	3.04269	0.10193

表5 累积利润  $F_2$  算法结果对比

算法	最优值	最差值	平均值	标准差
DSFA	<b>3.19224</b>	2.83043	3.14252	0.06683
SFLA	3.12797	2.80520	3.04524	0.08669
WOA	3.14477	2.77148	3.02661	0.11550
GWO	3.10526	2.82759	3.07719	0.05844
MFO	3.11228	2.81187	3.05554	0.07463
CSA	3.16357	2.71155	3.09220	0.13735
BAT	3.18686	2.71423	3.12039	0.11758
PSO	3.15711	2.84126	3.09750	0.08642
MVO	3.14522	2.94218	3.05203	0.07329
CFA	3.09580	2.90385	3.02562	0.05687

由表4和表5的实验结果可知,从最优值、最差值方面来看,DSFLA得到的优化结果最好。在迭代过程中,DSFLA算法引入扰动系数更新策略,更好的调控族群内最劣解和最优解之间的距离;并将扰动系数作为阈值,使用基于族群最优解的个体更新位置和螺旋更新位置两种策略,精细算法的局部搜索能力;此外,为了增加种群多样性,采用全局最优解变异的方式;同时,采用随机搜索策略来提高算法的全局搜索能力;为了保证离散化处理后青蛙种群的多样性,采用改进的Sigmoid函数离散化位置向量,避免种群的多样性不会在离散化处理后突然骤减,提高了算法跳出局部最优的能力。

从标准差和平均值方面可知,DSFLA算法相较于其它算法有较好的稳定性。此外,从图15和图16中可知,相较于其它算法,CSA算法最早陷入局部最优,BAT、GWO、MFO、WOA和PSO算法在600代左右停滞,SFLA算法在700代以后也不再变化,而DSFLA算法的收敛曲线在700以后仍有上升的趋势,由于最优解变异和随机搜索策略提高了DSFLA在迭代后期跳出局部最优能力,并且扰动系数更新策略扩大DSFLA算法的搜索空间,因此同其他9种算法相比,DSFLA可以得到高质量的解。

综上所述,DSFLA算法在求解油田措施规划模型时,能够显著提升油田开采的经济效益。

## 4 结论与展望

本文提出一种离散混合蛙跳算法(DSFLA),该算法引入扰动系数、螺旋位置更新策略、随机搜索策略以及全局最优解变异策略来对标准混合蛙跳算法的更新策略进行改进,此外,为了让改进的算法更适合求解离散优化问题,利用改进的Sigmoid函数对位置进行离散化处理。通过9个基准测试函数的实验结果表明了改进策略的高效性,并成功的将其应用到油田措施规划模型优化问题中,结果表明,与其它9种算法的优化结果相比,DSFLA算法求解质量更优。今后将进一步研究本算法在其它工程领域的应用。

## 参考文献

- 1 Eusuff MM, Lansey KE. Optimization of water distribution network design using the shuffled frog leaping algorithm. *Journal of Water Resources Planning and Management*, 2003, 129(3): 210–225.
- 2 雷德明, 杨冬婧. 基于新型蛙跳算法的低碳混合流水车间调度. *控制与决策*, 2019, 35(6): 1329–1337.
- 3 徐俊, 项倩红, 肖刚. 基于改进混合蛙跳算法的云工作流负载均衡调度优化. *计算机科学*, 2019, 46(11): 315–322. [doi: 10.11896/jsjcx.181001866]
- 4 张萍, 李婷, 张宏伟, 等. 基于人工鱼群和蛙跳混合算法的特高压电力变压器的建模研究. *电瓷避雷器*, 2018, (6): 7–12, 19.
- 5 陈科尹, 邹湘军, 关卓怀, 等. 基于混合蛙跳优化的采摘机器人相机标定方法. *农业机械学报*, 2019, 50(1): 23–34. [doi: 10.6041/j.issn.1000-1298.2019.01.002]
- 6 蔡宁, 张则强, 张颖, 等. 资源约束下拆卸线平衡问题的建模与改进混合蛙跳算法. *中国机械工程*, 2019, 30(17): 2091–2099. [doi: 10.3969/j.issn.1004-132X.2019.17.011]
- 7 杨哲, 杨侃, 吴云, 等. 改进二进制-实数编码混合蛙跳算法在水电机组短期发电调度中的应用. *天津大学学报(自然科学与工程技术版)*, 2019, 52(9): 979–989.
- 8 张异. 基于包装配送问题的混沌蛙跳布谷鸟算法研究. *包装工程*, 2019, 40(5): 174–179.
- 9 艾子义, 雷德明. 基于新型蛙跳算法的低碳柔性作业车间调度. *控制理论与应用*, 2017, 34(10): 1361–1368. [doi: 10.7641/CTA.2017.60768]
- 10 李荣波, 纪昌明, 孙平, 等. 基于改进混合蛙跳算法的梯级

- 水库优化调度. 长江科学院院报, 2018, 35(6): 30–35.
- 11 韩超, 段纬然, 贾长治, 等. 基于改进多种群 PSO 算法的火炮随动系统调节器参数优化. 火力与指挥控制, 2020, 45(6): 56–61, 66.
- 12 Saremi S, Mirjalili S, Lewis A. How important is a transfer function in discrete heuristic algorithms. *Neural Computing and Applications*, 2015, 26(3): 625–640.
- 13 龚文引. 差分演化算法的改进及其在聚类分析中的应用研究 [博士学位论文]. 武汉: 中国地质大学(武汉), 2010.
- 14 Mirjalili S, Lewis A. The whale optimization algorithm. *Advances in Engineering Software*, 2016, 95: 51–67.
- 15 Kennedy J, Eberhart R. Particle swarm optimization. *Proceedings of International Conference on Neural Networks (ICNN'95)*. Perth, Australia. 1995. 1942–1948.
- 16 Mirjalili S, Mirjalili SM, Hatamlou A. Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Computing and Applications*, 2016, 27(2): 495–513.
- 17 Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. *Advances in Engineering Software*, 2014, 69: 46–61.
- 18 Mirjalili S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems*, 2015, 89: 228–249.
- 19 Yang XS, Deb S. Cuckoo search via Lévy flights. *Proceedings of 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*. Coimbatore, India. 2009. 210–214.
- 20 Yang XS. A new metaheuristic bat-inspired algorithm. In: González JR, Pelta DA, Cruz C, *et al*, eds. *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*. Berlin: Springer, 2010. 65–74.
- 21 Eesa AS, Brifcani AMA, Orman Z. Cuttlefish algorithm—A novel bio-inspired optimization algorithm. *International Journal of Scientific & Engineering Research*, 2013, 4(9): 1978–1986.