

局部深度搜索的混合果蝇优化算法

刘成忠^{1,2}, 黄高宝¹, 张仁陟^{1,2*}, 柴强¹

(1. 甘肃省干旱生境作物学重点实验室, 兰州 730070; 2. 甘肃农业大学 资源与环境学院, 兰州 730070)

(* 通信作者电子邮箱 liucz@gsau.edu.cn)

摘要:针对基本果蝇优化算法(FOA)局部深度搜索能力较差且易陷入局部最优的缺点,提出了局部深度搜索的混合果蝇优化算法(SFOALDS)。通过借鉴混合蛙跳算法(SFLA)的更新策略,循环进行局部深度搜索操作,使得SFOALDS既保持了FOA较快的收敛速度,又增强了FOA局部深度搜索能力,有效避免了基本FOA易陷入局部最优的缺点,提高了进化后期算法的收敛速度和精度。仿真实验结果表明,SFOALDS比基本FOA和SFLA有较强的全局寻优性能,并且在高维函数上的优势更加明显。

关键词:果蝇优化算法;混合蛙跳算法;群体智能;局部搜索;早熟收敛

中图分类号: TP18; TP301.6 **文献标志码:** A

Shuffled fruit fly optimization algorithm with local deep search

LIU Chengzhong^{1,2}, HUANG Gaobao¹, ZHANG Renzhi^{1,2*}, CHAI Qiang¹

(1. Gansu Provincial Key Lab of Aridland Crop Science, Lanzhou Gansu 730070, China;

2. College of Resources and Environmental Sciences, Gansu Agricultural University, Lanzhou Gansu 730070, China)

Abstract: In order to overcome the demerits of poor deeply searching ability and easily relapsing into local extremum in basic Fruit Fly Optimization Algorithm (FOA), a new algorithm named Shuffled Fruit Fly Optimization Algorithm with Local Deep Search (SFOALDS) was proposed. The local optimal individual in each group was deeply searched circularly by referencing updating strategy of Shuffled Frog Leaping Algorithm (SFLA). SFOALDS not only efficiently avoids relapsing into local extremum, but also improves convergence velocity and convergence precision in the late evolution. The experimental results show that the proposed algorithm has better global searching performance than basic FOA and SFLA, especially on high dimensional functions.

Key words: Fruit fly Optimization Algorithm (FOA); Shuffled Frog Leaping Algorithm (SFLA); swarm intelligence; local search; premature convergence

0 引言

果蝇优化算法(Fruit fly Optimization Algorithm, FOA)由潘文超于2011年6月提出,是一类新的全局优化群智能算法,源于对果蝇觅食行为的模拟^[1-3],可广泛应用于科学和工程领域,也可混合其他的数据挖掘技术一起使用。目前,其已经成功应用于如求解数学函数极值^[2];微调Z-SCORE模型系数,提高企业财务危机预警的准确率^[2];优化广义回归神经网络进行企业经营绩效评估^[3]和四川省新政航电工程3台机组5个不同部位的振动序列峰峰值预测^[4];优化最小二乘支持向量机(Least Squares Support Vector Machines, LS-SVM)用于建立回转窑干燥速率模型^[5];辨识船舶操纵运动响应模型的结构参数,并用得到的响应模型进行自航模变Z形试验预报^[6];分离盲源语音信号^[7]等。

FOA与其他群智能算法相比,不但算法简单,容易理解(如粒子群算法(Particle Swarm Optimization, PSO)的优化方

程是二阶微分方程^[8],而FOA的优化方程是一阶微分方程),程序代码易于实现,运行时间较少,对搜索空间有一定的自适应能力,具有较强的鲁棒性和较好的收敛性能;而且FOA所需调整的参数比较少,仅有3个,而其他经典智能优化算法所需调整的参数至少为5个(如PSO需调整5个参数^[8]、人工鱼群算法(Artificial Fish Swarm Algorithm, AFSA)需调整5个参数^[9]、蚁群算法(Ant Colony Optimization, ACO)需调整7个参数^[10]、遗传算法(Genetic Algorithm, GA)需调整5个参数^[11]、细菌觅食优化算法需调整11个参数^[11]等),各个参数对算法性能的影响、参数之间的相互影响和复杂关系及对算法性能的二次影响很难研究清楚,一般都是通过大量实验总结出来的经验数值,但参数的取值不当,会严重影响算法的性能,并且导致分析算法复杂度变得异常困难。但同时FOA与其他全局优化算法(如GA、PSO)类似,易陷入局部最优,导致后期收敛速度变慢,收敛精度降低,尤其是对于高维多极值复杂优化问题。根据没有免费的午餐理论^[12],每种进化算法都

收稿日期:2013-10-25;修回日期:2013-12-25。

基金项目:国家自然科学基金资助项目(61063028);甘肃省自然科学基金资助项目(1208RJZA133);甘肃省教育厅科研基金资助项目(2013A-060);甘肃省干旱生境作物学重点实验室开放基金资助项目(GSCS-2012-15);甘肃农业大学青年导师基金资助项目(GAU-QNDS-201213)。

作者简介:刘成忠(1969-),男,甘肃天祝人,副教授,博士研究生,主要研究方向:智能决策支持系统;黄高宝(1965-2012),男,甘肃天水人,教授,博士,主要研究方向:农业生态学;张仁陟(1961-),男,甘肃静宁人,教授,博士,主要研究方向:生态学;柴强(1972-),男,甘肃武威人,教授,博士,主要研究方向:农业生态学。

有各自的优缺点,因此,如何将 FOA 与其他智能优化算法融合是一个重要的研究方向。

混合蛙跳算法(Shuffled Frog Leaping Algorithm, SFLA)^[13]是一种基于群体的亚启发式协同搜索算法,该算法在群中个体所在的模因组内进行局部深度搜索,然后利用模因组混合实现全局信息交换^[14]。本文针对 FOA 局部深度搜索能力较差,迭代后期易陷入局部最优的缺点,借鉴 SFLA 模因组内的局部深度搜索策略,提出一种果蝇优化算法和混合蛙跳算法相融合的新算法——局部深度搜索的混合果蝇优化算法(Shuffled Fruit Fly Optimization Algorithm with Local Deep Search, SFOALDS)。该算法既保持了 FOA 原有的较强的全局搜索能力和较快的收敛速度,又增强了 FOA 的局部深度搜索能力,从而有效平衡了整个种群的“探索”和“开发”能力,提高了算法的收敛速度和精度。用 FOA、SFLA 和 SFOALDS 对 5 个高维基准测试函数进行仿真实验的结果表明,本文算法收敛速度和寻优精度均优于 FOA 和 SFLA。

1 混合蛙跳算法

混合蛙跳算法模拟青蛙群体的觅食特性^[13-14]。对于 d 维问题,随机生成 N 只青蛙(解)组成初始群体 $P = \{X_1, X_2, \dots, X_N\}$,第 i 只蛙表示为 $X_i = [x_{i1}, x_{i2}, \dots, x_{id}]$ 。生成初始群体之后,首先将种群按适应值降序排列,并记录蛙群中最优个体 X_g ;然后将整个蛙群分为 m 个模因组(子群),每个模因组含有 n 只蛙,并且满足 $N = m \times n$ 的关系。其中:第 1 只蛙分入第 1 个子群,第 2 只蛙分入第 2 个子群,……,第 m 只蛙分入第 m 个子群,第 $m+1$ 只蛙重新分入第 1 个子群,依此类推,直到全部青蛙划分完毕。

每一个模因组中具有最优和最差适应值的个体分别记为 X_b 和 X_w ,然后对每个模因组中的 X_w 循环进行局部深度搜索操作,更新策略如下:

$$D = r \cdot (X_b - X_w) \quad (1)$$

$$X_w' = X_w + D; \|D\| \leq D_{\max} \quad (2)$$

式中: r 表示0到1之间的随机数, D_{\max} 表示蛙所允许改变位置的最大值。在经过更新后,如果得到的新个体 X_w' 优于原来的最差个体 X_w ,则取代原来模因组中的 X_w ;如果没有改进,则用 X_g 取代 X_b ,按式(1)、(2)执行局部搜索;如果仍然没有改进,则随机产生一个新个体取代原来的 X_w 。

当所有模因组都完成一定迭代次数局部搜索后,将所有模因组内的蛙混合,并排序和重新划分模因组,再进行局部深度搜索;如此反复,直到满足终止条件。

2 果蝇优化算法

果蝇本身在感官知觉上优于其他物种,尤其是在嗅觉与视觉上,果蝇的嗅觉器官能很好地搜集飘浮在空气中的各种气味,然后飞近食物位置后亦可使用敏锐的视觉发现食物与同伴聚集的位置,并且往该方向飞去^[1-3]。FOA 模拟果蝇群体的觅食特性,因此依据果蝇搜索食物特性,将 FOA 归纳为以下几个必要的步骤^[1-3]:

1) 初始化群体规模 $Sizepop$, 最大迭代数 $Maxgen$, 随机初始化果蝇群体位置 X_axis 、 Y_axis 。

2) 随机赋予果蝇个体利用嗅觉搜寻食物的方向与距离,随机数 $RandomValue$ 为迭代步进值。

$$\begin{cases} X_i = X_axis + RandomValue \\ Y_i = Y_axis + RandomValue \end{cases} \quad (3)$$

3) 估计果蝇个体与原点的距离 $Dist_i$,再计算果蝇个体的味道浓度判定值 S_i , S_i 为距离的倒数。

$$Dist_i = \sqrt{X_i^2 + Y_i^2} \quad (4)$$

$$S_i = 1/Dist_i \quad (5)$$

4) 将味道浓度判定值 S_i 代入味道浓度判定函数(或称为适应度函数 Fitness function),用来求出果蝇个体位置的味道浓度 $Smell_i$ 。

$$Smell_i = Function(S_i) \quad (6)$$

5) 找出该果蝇群体中味道浓度最佳的果蝇(适用于最小化问题)。

$$[bestSmell \quad bestindex] = \min(Smell_i) \quad (7)$$

6) 记录并保留最佳味道浓度值 $bestSmell$ 与其 X 、 Y 坐标,这时候果蝇群体利用视觉向该位置飞去。

$$\begin{cases} Smell_{best} = bestSmell \\ X_axis = X(bestindex) \\ Y_axis = Y(bestindex) \end{cases} \quad (8)$$

7) 进入迭代寻优,重复执行步骤2)~5),并判断最佳味道浓度是否优于前一迭代最佳味道浓度,并且若当前迭代次数小于最大迭代数 $Maxgen$,则执行步骤6);否则,结束算法。

3 局部深度搜索的混合果蝇优化算法

FOA 具有较强的全局搜索能力和较快的收敛速度,但迭代寻优时,向最优个体的聚集行为,极易导致种群多样性的损失,若该个体并不是全局最优,极易使算法陷入局部最优,带来早熟收敛的问题。

SFLA 迭代寻优时,利用模因组内的局部深度搜索策略和模因组混合策略的循环交替,实现全局寻优的目的,因而具有较强的局部深度搜索能力,并且为与其他算法的融合提供了方便。

基于上述分析,本文提出了局部深度搜索的混合果蝇优化算法 SFOALDS,该算法将 FOA 和 SFLA 进行融合,充分发挥 FOA 和 SFLA 各自的优势,既保持 FOA 原有的较强的全局搜索能力和较快的收敛速度,又恰当引入 SFLA 模因组内的局部深度搜索策略,从而平衡整个种群的“探索”和“开发”能力,有效避免了基本 FOA 易陷入局部最优的缺点。

SFOALDS 以 FOA 的运算流程为主体流程, SFOALDS 流程如图 1 所示。

4 仿真实验及结果分析

4.1 实验设计

以求 5 个基准测试函数最小值为例,进行仿真实验,来验证并比较本文提出的 SFOALDS 的性能。函数名称、函数形式、搜索区间、函数最小值和峰值见表 1,所有函数均为高维函数。测试软件平台为 Windows XP, Matlab 7.1, 机器主频为 3.3 GHz, 内存为 2 GB。

设计了三个算法的优化实验来验证并比较本文算法的性能:1) FOA 优化实验;2) SFLA 优化实验;3) SFOALDS 优化实

验。具体实验参数设置为:群体规模 $Sizepop = 200$, 最大混合迭代数 $Maxgen1 = 500$, 最大内迭代数 $Maxgen2 = 10$, 模因组数 $m = 20$, 每个模因组内的个体个数 $n = 10$, 果蝇优化迭代步进值 $RandomValue \in [-1, 1]$; 随机初始果蝇群体位置为各函数的搜索区间。

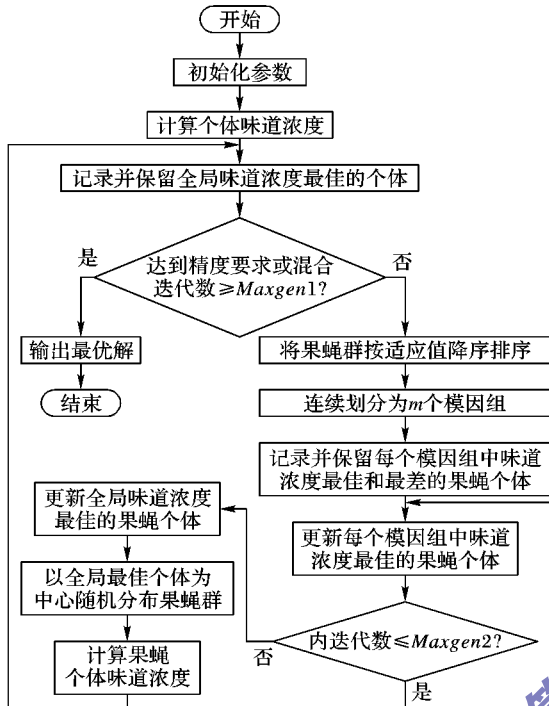


图1 SFOALDS 流程

4.2 实验结果与分析

4.2.1 算法在高维函数上的优化性能对比分析

智能优化算法普遍存在易陷入局部最优,导致后期收敛速度变慢,收敛精度降低的问题,尤其对于高维、多峰复杂优化问题^[15-16],该实验均在高维函数上完成。

以 100 维的单峰函数 f_1 和多峰函数 f_2 为例,分别采用 FOA、SFLA 和 SFOALDS 三种算法对其进行优化,全局寻优函数最小值,从而测试本文算法在高维函数上的优化性能。经过 30 次连续运行后的平均最优适应值(全局寻优函数最小值的算术平均)进化曲线如图 2 所示,图中纵坐标用平均最优适应值的常用对数表示,横坐标为进化代数。从图 2 中可以看出,SFOALDS 的收敛性能和优化精度明显优于 FOA 和 SFLA,尤其是对于 f_2 , SFOALDS 依然能很快达到理论极小值 0,有效避免了 FOA 和 SFLA 陷入局部最优的缺点。

以高维、多峰函数 f_2 、 f_4 和 f_5 为例,函数维数从 40 ~ 100 维变化,分别采用 FOA、SFLA 和 SFOALDS 三种算法全局寻优函数最小值,从而测试本文算法性能随函数维数增加而变化的情况,以优化均值和相对变化率作为评价指标。经过 30 次连续运行后的实验结果如表 2 所示,表中优化均值 = 全局寻优函数最小值的算术平均,相对变化率 = |40 维的优化均值 - 100 维的优化均值| / 40 维的优化均值。从表 2 中可以看出,随着函数维数的增加,三种算法的优化性能都普遍降低了,即优化均值都增大了;但不论维数如何变化,三种算法中依然是 SFOALDS 的优化性能最好,即优化均值最小;还可以看出,三种算法中 SFOALDS 的相对变化率也是最小的,这说明随着函数维数的增加,SFOALDS 优化性能相对降低得最少,从而也说明,随着函数维数的增加,SFOALDS 相对于 FOA 和 SFLA 的优化性能的优势会更加突出。

三种算法在 f_4 和 f_5 上的优化均值随函数维数变化而变化的趋势线如图 3 和图 4 所示,图中纵坐标用优化均值表示,横坐标为函数维数。从图 3、4 中可以看出,FOA 和 SFLA 优化均值随函数维数增加单调递增,并且维数越大,优化均值的增长率越大;SFOALDS 优化均值随函数维数增加不是单调递增的,甚至当函数维数增加时优化均值反而减小了。

需特别指出,SFOALDS 特别适用于优化 f_2 , 不论函数维数如何变化,优化均值都为 0。

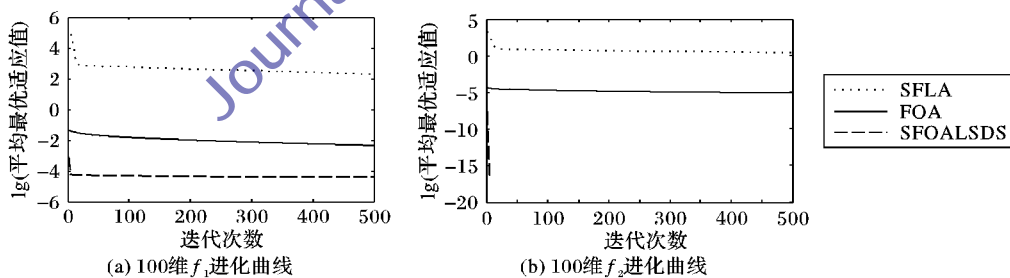


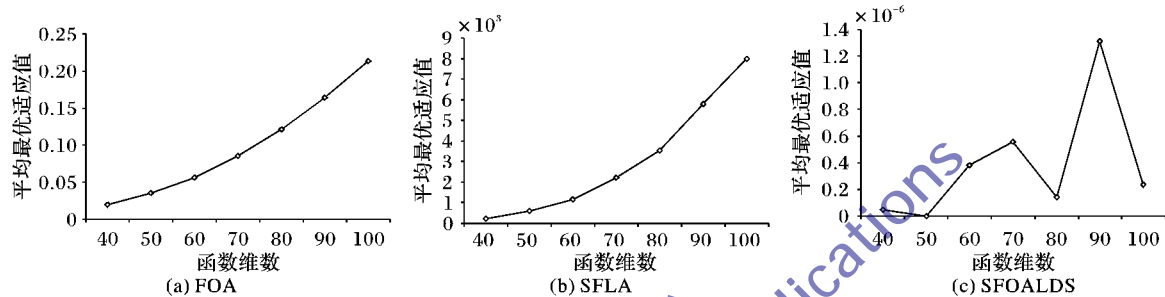
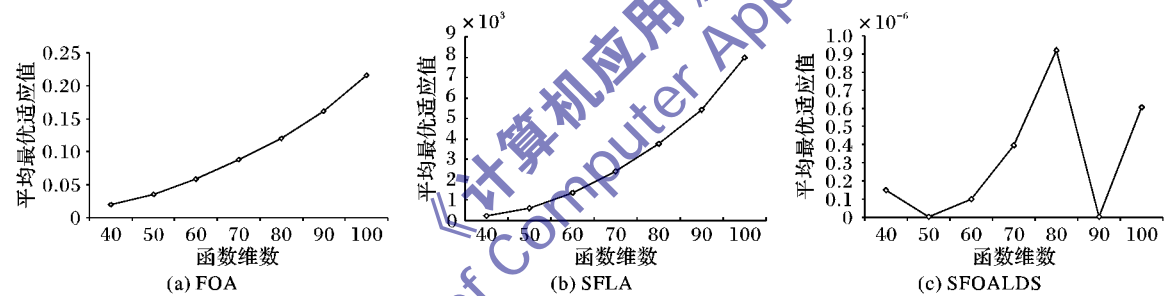
图2 100 维 f_1, f_2 在 3 个实验中的平均最优适应值对数值进化曲线

表1 5 个测试函数

函数	函数形式	搜索区间	最小值	峰值
Sphere (f_1)	$f_1(x) = \sum_{i=1}^n x_i^2$	$[-100, 100]$	0	单峰
Griewank (f_2)	$f_2(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]$	0	多峰
Rosenbrock (f_3)	$f_3(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	$[-30, 30]$	0	单峰
Rastrigin (f_4)	$f_4(x) = \sum_{i=1}^n (x_i^2 - 10\cos(2\pi x_i) + 10)$	$[-5.12, 5.12]$	0	多峰
Quadric (f_5)	$f_5(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j^2 \right)$	$[-100, 100]$	0	多峰

表2 高维、多峰函数上的优化均值比较

函数	算法	维数							相对变化率/%
		40	50	60	70	80	90	100	
f_2	FOA	6.4308E-06	7.1623E-06	8.0067E-06	8.7073E-06	8.6804E-06	9.4800E-06	9.8690E-06	53.46
	SFLA	1.1105E+00	1.2367E+00	1.3808E+00	1.6593E+00	1.9156E+00	2.2610E+00	2.7295E+00	145.79
	SFOALDS	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0
f_4	FOA	1.9700E-02	3.6000E-02	5.7000E-02	8.5900E-02	1.2120E-01	1.6400E-01	2.1360E-01	984.26
	SFLA	2.1331E+02	5.8945E+02	1.1553E+03	2.2091E+03	3.5267E+03	5.7954E+03	7.9990E+03	3649.94
	SFOALDS	4.6963E-08	2.8074E-09	3.8277E-07	5.5493E-07	1.4089E-07	1.3105E-06	2.3330E-07	396.77
f_5	FOA	2.0000E-02	3.5800E-02	5.8200E-02	8.7500E-02	1.2040E-01	1.6110E-01	2.1520E-01	976.00
	SFLA	2.4004E+02	6.1924E+02	1.3389E+03	2.4163E+03	3.7485E+03	5.4176E+03	7.9802E+03	3224.53
	SFOALDS	1.5036E-07	5.3238E-09	1.0217E-07	3.9958E-07	9.2069E-07	2.9578E-09	6.0524E-07	302.53

图3 f_4 平均最优适应值随函数维数变化曲线图4 f_5 平均最优适应值随函数维数变化曲线

综上所述,可以得出本文算法对高维函数仍有很强的优化能力。

4.2.2 与参考文献中经典优化算法的性能对比分析

表3是本文算法SFOALDS与参考文献中各种经典优化算法(AFSA^[9]、标准粒子群优化算法(Standard PSO, SPSO)^[18])和各种改进的混合蛙跳算法^[19-21]的对比结果。

表3 各算法的平均最优适应值对比

函数(30维)	SFOALDS	AFSA ^[9]	SPSO ^[17]	SFLA-OCO ^[18]	REF ^[19]	ODSFLA1 ^[20]	ODSFLA2 ^[20]
f_1	1.77E-10	10121.82	1.10E-24	0.000009	0.002004	7.34E-04	1.95E-08
f_2	0	106.5802	0.13	0.043305	0.090165	1.92E-01	3.30E-01
f_3	2.84E+01	1.09E+10	34.1	—	11.935892	5.98E+01	3.41E+01
f_4	4.99E-09	306.8284	101.7	9.38701	7.198724	8.23E+00	7.62E+00
f_5	1.93E-11	134087.3	—	—	—	—	—

综上所述,整体看来,本文提出的SFOALDS比FOA、SFLA和参考文献中报道的优化算法具有更高的收敛精度。

5 结语

针对果蝇优化算法局部深度搜索能力较差和易陷入局部最优的缺点,将混合蛙跳算法的局部深度搜索策略融入到果蝇优化算法中,提出了一种局部深度搜索的混合果蝇优化算法。该算法充分发挥了FOA和SFLA各自的优点,既保持了

表3中给出了各算法在迭代后得到的平均最优适应值,平均最优适应值指实验全局寻优函数最小值的算术平均;“—”表示参考文献中没有报道。

由表3可以看出,本文提出的SFOALDS在参数(如:迭代次数或搜索区间或种群规模等)比较苛刻的情况下,具有更高的收敛精度。

FOA原有的较强的全局搜索能力和较快的收敛速度,又平衡了整个种群的“探索”和“开发”能力,有效避免了基本FOA易陷入局部最优的缺点。通过优化5个测试函数的仿真结果表明,本文算法收敛速度和寻优精度均优于FOA和SFLA,且该算法具有简单、高效的特点,适用于解决实际工程问题。

参考文献:

- [1] 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.

- [2] PAN W-T. Fruit fly optimization algorithm [M]. Taipei: Tsang Hai Book Publishing Co., 2011: 10-12. (潘文超. 果蝇最佳化演算法[M]. 台北: 沧海书局, 2011: 10-12.)
- [3] PAN W-T. Using fruit fly optimization algorithm optimized general regression neural network to construct the operating performance of enterprises model [J]. Journal of Taiyuan University of Technology: Social Sciences Edition, 2011, 29(4): 1-5. (潘文超. 应用果蝇优化算法优化广义回归神经网络进行企业经营绩效评估[J]. 太原理工大学学报: 社会科学版, 2011, 29(4): 1-5.)
- [4] TIAN Y, ZHANG B, LIU D, *et al.* Condition trend prediction of hydroelectric sets based on fruit fly optimization algorithm optimized GRNN [J]. Water Resources and Power, 2012, 30(12): 127-129. (田源, 张彼德, 刘代伟, 等. 基于果蝇优化算法的 GRNN 水电机组状态趋势预测[J]. 水电能源科学, 2012, 30(12): 127-129.)
- [5] WANG X, DU K, QIN B, *et al.* Drying rate modeling based on FOALSSVR [J]. Control Engineering of China, 2012, 19(4): 630-633. (王欣, 杜康, 秦斌, 等. 基于果蝇优化算法的 LSSVR 干燥速率建模控制工程[J]. 控制工程, 2012, 19(4): 630-633.)
- [6] WANG X, ZOU Z. Identification of ship manoeuvring response model based on fruit fly optimization algorithm [J]. Journal of Dalian Maritime University, 2012, 38(3): 1-5. (王雪刚, 邹早建. 基于果蝇优化算法的船舶操纵响应模型的辨识[J]. 大连海事大学学报, 2012, 38(3): 1-5.)
- [7] XIAO Z. Application of improved FOA on audio signal blind separation [J]. Computer Engineering and Applications, 2013, 49(16): 201-204. (肖正安. 改进 FOA 算法在语音信号盲分离中的应用[J]. 计算机工程与应用, 2013, 49(16): 201-204)
- [8] HU W, LI Z. A simpler and more effective particle swarm optimization algorithm [J]. Journal of Software, 2007, 18(4): 861-868. (胡旺, 李志蜀. 一种更简化而高效的粒子群优化算法[J]. 软件学报, 2007, 18(4): 861-868.)
- [9] WANG L, HONG Y, SHI Q. Global edition artificial fish swarm algorithm [J]. Journal of System Simulation, 2009, 21(23): 7483-7486. (王联国, 洪毅, 施秋红. 全局版人工鱼群算法[J]. 系统仿真学报, 2009, 21(23): 7483-7486.)
- [10] MA D, GUO X, DENG L. Ammunition scheduling of carrier-based aircraft based on modified ant colony algorithm [J]. Journal of System Simulation, 2012, 24(6): 1207-1211. (马登武, 郭小威, 邓力. 基于改进蚁群算法的舰载机弹药调度[J]. 系统仿真学报, 2012, 24(6): 1207-1211.)
- [11] HU J. Research on the modified bacteria foraging optimization algorithm and its application [D]. Wuhan: Wuhan University, 2012. (胡洁. 细菌觅食优化算法的改进及应用研究[D]. 武汉: 武汉大学, 2012.)
- [12] WOLPENT D H, MACREADY W G. No free lunch theorems for optimization [J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 67-82.
- [13] EUSUFF M M, LANSEY K E. Optimization of water distribution network design using the shuffled frog leaping algorithm [J]. Journal of Water Resources Planning and Management, 2003, 129(3): 210-225.
- [14] CUI W, LIU X, WANG W, *et al.* Survey on shuffled frog leaping algorithm [J]. Control and Decision, 2012, 27(4): 481-487. (崔文华, 刘晓冰, 王伟, 等. 混合蛙跳算法研究综述[J]. 控制与决策, 2012, 27(4): 481-487.)
- [15] LI M, JI T, TANG W, *et al.* Bacterial foraging algorithm with varying population [J]. BioSystems, 2010, 100(3): 185-197.
- [16] JIN Q, ZHAO Z, SU X, *et al.* PSO algorithm with high speed convergence based on particle health [J]. Journal of Chemical Industry and Engineering, 2011, 62(8): 2328-2333. (靳其兵, 赵振兴, 苏晓静, 等. 基于粒子健康度的快速收敛粒子群优化算法[J]. 化工学报, 2011, 62(8): 2328-2333.)
- [17] LIN C, FEND Q. New adaptive particle swarm optimization algorithm [J]. Computer Engineering, 2008, 34(7): 181-183. (林川, 冯全源. 一种新的自适应粒子群优化算法[J]. 计算机工程, 2008, 34(7): 181-183.)
- [18] MENG Q, WANG L. Shuffled frog leaping algorithm based on neighborhood orthogonal crossover operator [J]. Computer Engineering and Applications, 2011, 47(36): 54-56. (孟庆莹, 王联国. 基于邻域正交叉算子的混合蛙跳算法[J]. 计算机工程与应用, 2011, 47(36): 54-56.)
- [19] LIU Y. Shuffled frog leaping algorithm with selection and adaptive mutation mechanism [J]. Computer Engineering, 2012, 38(23): 206-210. (刘悦婷. 带有选择和自适应变异机制的混合蛙跳算法[J]. 计算机工程, 2012, 38(23): 206-210.)
- [20] ZHAO P, SHAO Z. Novel improved shuffled frog leaping algorithm [J]. Computer Engineering and Applications, 2012, 48(8): 48-50. (赵鹏军, 邵泽军. 一种新的改进的混合蛙跳算法[J]. 计算机工程与应用, 2012, 48(8): 48-50.)

(上接第 1059 页)

- [10] BANDYOPADHYAY S K, BHATTACHARYYA D, DAS P. Mosquito attack optimization [J]. Ubiquity, 2006, 7(33): Article No. 9.
- [11] FENG X, LAU F C M, YU H. A novel bio-inspired approach based on the behavior of mosquitoes [J]. Information Sciences, 2013, 233: 87-108.
- [12] KENNEDY J, EBERHART R. Particle swarm optimization [C]// Proceedings of the 1995 IEEE International Conference on Neural Networks. Piscataway: IEEE, 1995, 4: 1942-1948.
- [13] WEISSING F J. Born leaders [J]. Nature, 2011, 474(7351): 288-289.
- [14] HE S, WU Q H, SAUNDERS J R. Group search optimizer: an optimization algorithm inspired by animal searching behavior [J]. IEEE Transactions on Evolutionary Computation, 2009, 13(5): 973-990.
- [15] DORIGO M, Di CARO G. Ant colony optimization: a new meta-heuristic [C]// CEC 99: Proceedings of the 1999 Congress on Evolutionary Computation. Piscataway: IEEE, 1999, 2: 1470-1777.
- [16] KARABOGA D, BASTURK B. A powerful and efficient algorithm for numerical function optimization: Artificial Bee Colony (ABC) algorithm [J]. Journal of global optimization, 2007, 39(3): 459-471.
- [17] DORIGO M, Di CARO G, SAMPELS M. Ant algorithms, Proceedings of the Third International Workshop, LNCS 2463 [C]. Berlin: Springer-Verlag, 2002.
- [18] LI J, ZHANG L, LIU L. A parallel immune algorithm based on fine-grained model with GPU-acceleration [C]// ICICIC '09: Proceedings of the 2009 Fourth International Conference on Innovative Computing, Information and Control. Washington, DC: IEEE Computer Society, 2009: 683-686.
- [19] ZHAO J, LIU Q, WANG W, *et al.* A parallel immune algorithm for traveling salesman problem and its application on cold rolling scheduling [J]. Information Sciences: an International Journal, 2011, 181(7): 1212-1223.