动态双子群协同进化果蝇优化算法*

韩俊英 刘成忠 王联国

(甘肃农业大学 信息科学技术学院 兰州 730070)

摘 要 针对基本果蝇优化算法(FOA)寻优精度不高和易陷入局部最优的缺点,提出动态双子群协同进化果蝇优化算法(DDSCFOA).该算法在运行过程中根据群体的进化水平,动态地将整个种群划分为先进子群和后进子群;先进子群采用混沌算法在局部最优解邻域内进行精细的局部搜索,后进子群采用基本 FOA 算法进行全局搜索,较好地平衡局部搜索能力和全局搜索能力;两个子群间的信息通过全局最优个体的更新和种群个体的重组进行交换. DDSCFOA 算法能跳出局部极值,避免陷入局部最优. 仿真结果表明,动态双子群协同进化的策略有效可行,DDSCFOA 算法比基本 FOA 算法具有更好的优化性能.

关键词 果蝇优化算法, 群体智能, 协同进化, 早熟收敛
中图法分类号 TP 181

Dynamic Double Subgroups Cooperative Fruit Fly Optimization Algorithm

HAN Jun-Ying, LIU Cheng-Zhong, WANG Lian-Guo
(*College of Information Science and Technology*, *Gansu Agricultural University*, *Lanzhou* 730070)

ABSTRACT

In order to overcome the demerits of basic Fruit Fly Optimization Algorithm (FOA), such as low convergence precision and easily relapsing into local optimum, a dynamic double subgroup cooperative Fruit Fly Optimization Algorithm (DDSCFOA) is presented. Firstly, the whole group is dynamically divided into advanced subgroup and backward subgroup according to its own evolutionary level. Secondly, a finely local searching is made for advanced subgroup in the neighborhood of local optimum with Chaos algorithm, and a global search with FOA is made for backward subgroup, so that the whole group keeps in good balance between the global searching ability and local searching ability. Finally, two subgroups exchange information by updating the overall optimum and recombining the subgroups. DDSCFOA can jump out of local optimum and avoid falling into local optimum. The experimental results show that the strategy of dynamic double subgroup cooperative evolution is effective and feasible, DDSCFOA is much better than basic FOA in convergence velocity and convergence precision.

Key Words Fruit Fly Optimization Algorithm, Swarm Intelligence, Cooperative Evolution, Premature Convergence

* 国家自然科学基金项目(No. 61063028)、甘肃省自然科学基金项目(No. 1208RJZA133)、甘肃省科技支撑计划项目(No. 1011NKCA058)、甘肃省教育厅科研基金项目(No. 1202-04)、甘肃省高等学校科研基金项目(No. 2013A-060)资助
收稿日期:2013-02-27;修回日期:2013-05-14
作者简介 **韩俊英**(通讯作者),女,1975 年生,硕士,副教授,主要研究方向为优化计算、农业信息技术. E-mail:hanjy@gsau.edu.cn. **刘成忠**,男,1969 年生,副教授,博士研究生,主要研究方向为智能决策支持系统. **王联国**,男,1968 年生,博士,教授,主要研究方向为计算智能及其工程应用、智能信息处理.

1 引 言

果蝇优化算法(Fruit Fly Optimization Algorithm, FOA)[1-3]是由台湾博士潘文超在2011年6月提出的一类新的全局优化进化算法,该算法源于对果蝇觅食行为的模拟,可广泛应用于科学和工程领域,也可混合其它的数据挖掘技术一起使用. 现已将其成功应用于求解数学函数极值[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回归(Least Squares Support Vector Regression, LSSVR)用于建立回转干燥窑干燥速率模型[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. 由于FOA提出较晚,目前国内外的研究尚处于起步阶段,研究成果还较少,理论也不成熟,因此FOA算法的相关研究迫切需要展开.

FOA与其它群智能算法比较,不但算法简单易理解(如粒子群算法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,且FOA所需调整的参数较少,仅有3个,而其它经典智能优化算法所需调整的参数至少为5个,如粒子群算法PSO需调整5个参数[8]、人工鱼群算法AFSA需调整5个参数[9]、蚁群算法ACO需调整7个参数[10]、遗传算法GA需调整5个参数[11]、细菌觅食优化算法需调整8个参数等[11]. 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较难研究清楚,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与其它全局优化算法(如遗传算法、粒子群算法等)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.

本文针对FOA的寻优精度不高和易陷入局部最优的缺点,提出了一种动态双子群协同进化果蝇优化算法(Dynamic Double Subgroups Cooperative Fruit Fly Optimization Algorithm, DDSCFOA),该算法在迭代进化过程中,根据群体的进化水平,即适应度函数取值,动态地将整个种群划分为分别以当代最优个体和最差个体为几何中心的先进子群和后进子群;根据不同子群的特点,先进子群采用混沌算法在最优个体邻域内进行精细的局部搜索,后进子群在最优个体的指导下,采用基本FOA算法进行全局搜索,平衡整个种群的"开发"和"探索"能力;两个子群间的信息通过最优个体的更新和种群个体的重组进行交换,从而有效避免FOA易陷入局部最优的缺点,提高进化后期算法的收敛速度和精度. 6个基准测试函数的对比实验结果说明,本文算法的全局收敛性和寻优精度得到较显著提高.

2 果蝇优化算法

果蝇优化算法是一种基于果蝇觅食行为推演出的寻求全局优化的新方法. 果蝇本身在感官知觉上优于其它物种,尤其是在嗅觉与视觉上. 果蝇的嗅觉器官能较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,且往该方向飞去.

依据果蝇搜索食物特性,将果蝇优化算法归纳为以下步骤[1].

step 1 初始化参数. 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.

step 2 赋予果蝇个体利用嗅觉搜寻食物之随机方向与距离,$RandomValue$ 为搜索距离:

$$X_i = X\_axis + RandomValue,$$

$$Y_i = Y\_axis + RandomValue.$$

step 3 由于无法得知食物位置,因此先估计果蝇个体与原点的距离 $Dist_i$,再计算果蝇个体味道浓度判定值 S_i,此值为距离的倒数:

$$Dist_i = \sqrt{X_i^2 + Y_i^2},$$

$$S_i = \frac{1}{Dist_i}.$$

step 4 将味道浓度判定值 S_i 代入味道浓度判定函数(即适应度函数 $Fitness\ function$),用来求出果蝇个体的味道浓度 $Smell_i$:

$$Smell_i = Function(S_i).$$

step 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):

$$[bestSmell\ bestindex] = \min(Smell_i).$$

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

$$Smellbest = bestSmell,$$
$$X\_axis = X(bestindex),$$
$$Y\_axis = Y(bestindex).$$

step 7　进入迭代寻优,重复执行 step 2~step 5,并判断最佳味道浓度是否优于前一迭代最佳味道浓度,并且当前迭代次数小于最大迭代数 $Maxgen$,若是则执行 step 6.

3　动态双子群协同进化果蝇优化算法

FOA 在整个迭代寻优进化过程中只向当前最优果蝇个体学习,一旦发现本次迭代的最优个体,所有个体都聚集到该个体位置,若该个体并不是全局最优,极易使算法陷入局部最优,降低收敛速度和收敛精度,带来早熟收敛的问题. 要克服早熟收敛问题,就必须提供一种机制,让算法在发生早熟收敛时,能跳出局部最优,进入解空间的其它区域继续进行搜索,直到找到全局最优解[12].

社会学经验告诉我们,全局最优解一般存在于局部最优解的附近;整个种群的进化速度并不取决于较优秀的个体,反而更大程度取决于比较落后的个体. 基于上述两点启示,本文提出一种动态双子群协同进化果蝇优化算法 DDSCFOA,该算法以基本果蝇优化算法 FOA 运算流程为主体流程,在迭代进化过程中,首先,计算种群中果蝇个体 i 分别与当代最优个体和当代最差个体之间的距离 $Dist_{i\_best}$ 和 $Dist_{i\_worst}$,若 $Dist_{i\_best} \leqslant Dist_{i\_worst}$,则将果蝇个体 i 划分到以当代最优个体为几何中心的先进子群;否则,将其划分到以当代最差个体为几何中心的后进子群(两个子群中的个体及个体数量是随着迭代进化过程动态变化的);然后,根据不同子群的特点,先进子群采用混沌算法将个体通过混沌映射规则映射到混沌变量空间的取值区间内,利用混沌变量的遍历性和规律性寻优搜索,最后将获得的优化解线性转化到优化空间,从而实现在当代最优个体邻域内进行精细的局部搜索,保证整个种群的"开发"能力;后进子群在当代最优个体指导下,采用 FOA 算法进行全局搜索,保证整个种群的"探索"能力;两个子群间的信息通过最优个体的更新和种群的个体重组进行交换,以此克服基本 FOA 算法寻优精度不高和易陷入局部最优的缺点,从而提高整个算法的求解精度及求解效率.

DDSCFOA 算法具体流程如下.

step 1　初始化参数. 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.

step 2　执行 FOA 算法 step 2 ~ step 4.

step 3　找出该果蝇群体中味道浓度最佳的果蝇(最优个体),味道浓度最差的果蝇(最差个体):

$$[bestSmell\ bestindex] = \min(Smell_i),$$
$$[worstSmell\ worstindex] = \max(Smell_i).$$

step 4　分别记录并保留最佳味道浓度值 $bestSmell$、最优果蝇个体位置 X_b,Y_b 坐标,及最差味道浓度值 $worstSmell$、最差果蝇个体位置 X_w,Y_w 坐标:

$$Smellbest = bestSmell,$$
$$X_b = X(bestindex),$$
$$Y_b = Y(bestindex),$$
$$Smellworst = worstSmell,$$
$$X_w = X(worstindex),$$
$$Y_w = Y(worstindex).$$

step 5　分别计算果蝇个体 i 与最优个体的距离 $Dist_{i\_best}$ 及与最差个体的距离 $Dist_{i\_worst}$:

$$Dist_{i\_best} = \sqrt{(X_i - X_b)^2 + (Y_i - Y_b)^2},$$
$$Dist_{i\_worst} = \sqrt{(X_i - X_w)^2 + (Y_i - Y_w)^2}.$$

step 6　若 $Dist_{i\_best} \leqslant Dist_{i\_worst}$,则将果蝇个体 i 划分到先进子群,转 step 7;否则,将果蝇个体 i 划分到后进子群,转 step 12.

step 7　首先利用式(1)将果蝇个体位置 X_i,Y_i 转换为混沌变量 CX_i,CY_i;然后利用基于 $Logistic$ 映射的混沌变换算式(2)将混沌变量 CX_i,CY_i 进行变换;最后利用式(3)将变换后的混沌变量 CX_i,CY_i 转化为搜索空间内果蝇个体新位置 X'_i,Y'_i:

$$CX_i = \frac{X_i - a_i}{b_i - a_i},\quad CY_i = \frac{Y_i - a_i}{b_i - a_i}; \tag{1}$$

$$CX_i = 4CX_i * (1 - CX_i),\quad CY_i = 4CY_i * (1 - CY_i); \tag{2}$$

$$X'_i = a_i + CX_i(b_i - a_i),\quad Y'_i = a_i + CY_i(b_i - a_i). \tag{3}$$

step 8　先估计新位置 X'_i,Y'_i 与原点之距离 $Dist'_i$,再计算新位置味道浓度判定值 S'_i:

$$Dist'_i = \sqrt{X'^2_i + Y'^2_i},$$
$$S'_i = \frac{1}{Dist'_i}.$$

step 9　将新位置味道浓度判定值 S'_i 代入味道浓度判定函数,求出新位置的味道浓度 $Smell'_i$:

$$Smell'_i = Function(S'_i).$$

step 10 若 $Smell'_i < Smellbest$，则 $Smellbest = Smell'_i$，$X_b = X'_i$，$Y_b = Y'_i$，转 step 16；否则，直接转 step 11.

step 11 若 $Smell'_i > Smellworst$，则 $Smellworst = Smell'_i$，$X_w = X'_i$，$Y_w = Y'_i$，转 step 16；否则，直接转 step 16.

step 12 果蝇个体利用视觉向最优个体位置飞去：

$$X_i = X_b + RandomValue,$$

$$Y_i = Y_b + RandomValue.$$

step 13 计算果蝇个体位置 X_i，Y_i 与原点之距离 $Dist_i$，再计算味道浓度判定值 S_i 和味道浓度 $Smell_i$.

step 14 若 $Smell_i < Smellbest$，则 $Smellbest = Smell_i$，$X_b = X_i$，$Y_b = Y_i$.

step 15 若 $Smell_i > Smellworst$，则 $Smellworst = Smell_i$，$X_w = X_i$，$Y_w = Y_i$.

step 16 进入迭代寻优，重复执行 step 3 ~ step 15，直至当前迭代次数等于最大迭代数 $Maxgen$ 或已达到目标精度要求或理论最优值.

4 实验与结果分析

4.1 标准测试函数

以求 6 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 DDSCFOA 算法性能. 函数名称、函数形式、搜索区间、函数最优值和收敛目标精度见表 1，除 f_6 为 2 维函数外，其它函数均为多维函数. 测试软件平台为 Windows XP，Matlab7.1，机器主频为 P4(1.7GHz)，内存为 1GB.

4.2 本文算法收敛性测试结果与分析

实验中参数设置为：群体规模 $Sizepop = 30$，最大迭代数 $Maxgen = 150$，搜索距离 $RandomValue \in [-1, +1]$；随机初始化果蝇群体位置为表 1 中各函数的搜索区间.

4.2.1 固定进化迭代次数的收敛速度和精度

固定进化迭代次数 150 次，将 FOA 和 DDSCFOA 两种算法全局寻优函数最小值所得的优化均值、标准差和平均运行时间作为评价指标. 6 个测试函数经过 20 次连续运行后的实验结果如表 2 所示，表中优化均值等于全局寻优函数最小值的算术平均. 虽然 DDSCFOA 的平均运行时间比 FOA 长，但除 f_3 外，对其它所有函数，DDSCFOA 的优化均值和标准差均优于 FOA. 值得一提的是 Schaffer f_6 是一个具有无数个极小值点的多峰函数，其中只有一个最小值点，一般算法都较难找到全局最优点[13]，DDSCFOA 对这个函数能很快达到理论最优值 0，标准差也为 0，有效避免 FOA 陷入局部最优的缺点. 图 1 是 6 个测试函数优化均值对数值进化曲线(为方便进化曲线的显示和观察，本文对所有函数的适应度取以 10 为底的对数)，图中实线是 FOA 的进化曲线，虚线是 DDSCFOA 的进化曲线. 进化曲线也表明，除 f_3 外，DDSCFOA 的收敛速度和收敛精度明显优于 FOA. 因此，总体来说 DDSCFOA 的收敛精度、收敛速度和收敛稳定性均优于 FOA.

f_3 出现例外，是因为该函数在靠近最优点区域的适应值地形为香蕉状，变量之间具有较强的相关性，因此 DDSCFOA 算法的优势体现不出来.

表 1 6 个测试函数

Table 1 6 benchmark test functions

函数	函数形式	搜索区间	函数最小值	目标精度
Sphere f_1	$f_1(x) = \sum_{i=1}^{n} x_i^2$	$[-100,100]$	0	10^{-5}
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	10^{-6}
Rosenbrock f_3	$f_3(x) = \sum_{i=1}^{n-1}(100(x_{i+1}^2 - x_i)^2 + (x_i - 1)^2)$	$[-30,30]$	0	30
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	10^{-5}
Quadric f_5	$f_5(x) = \sum_{i=1}^{n}\left(\sum_{j=1}^{i} x_j^2\right)$	$[-100,100]$	0	10^{-5}
Schaffer f_6	$f_6(x) = \frac{\sin^2\sqrt{x_1^2 + x_2^2} - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2} + 0.5$	$[-100,100]$	0	10^{-7}

表 2　2 种算法实验结果比较

Table 2　Experimental result comparison of 2 algorithms

函数	维数	FOA			DDSCFOA		
		优化均值	标准差	平均运行时间 /s	优化均值	标准差	平均运行时间 /s
f_1	30	0.0030	3.7912e004	0.09779495	3.8994e017	1.7355e-016	0.35430485
f_2	30	1.2790e-005	4.4299e-006	0.1319063	6.8122e-012	3.0465e-011	0.56301995
f_3	30	28.3240	0.2353	0.11507955	28.7161	0.0052	0.56436135
f_4	30	0.1552	0.3619	0.127425	1.5143e-011	6.7720e-011	0.5375662
f_5	30	0.0443	0.0062	0.38099125	7.9406e-014	3.5511e-013	0.6555413
f_6	2	5.6133e-005	9.9566e-006	0.08412375	0	0	0.11331055

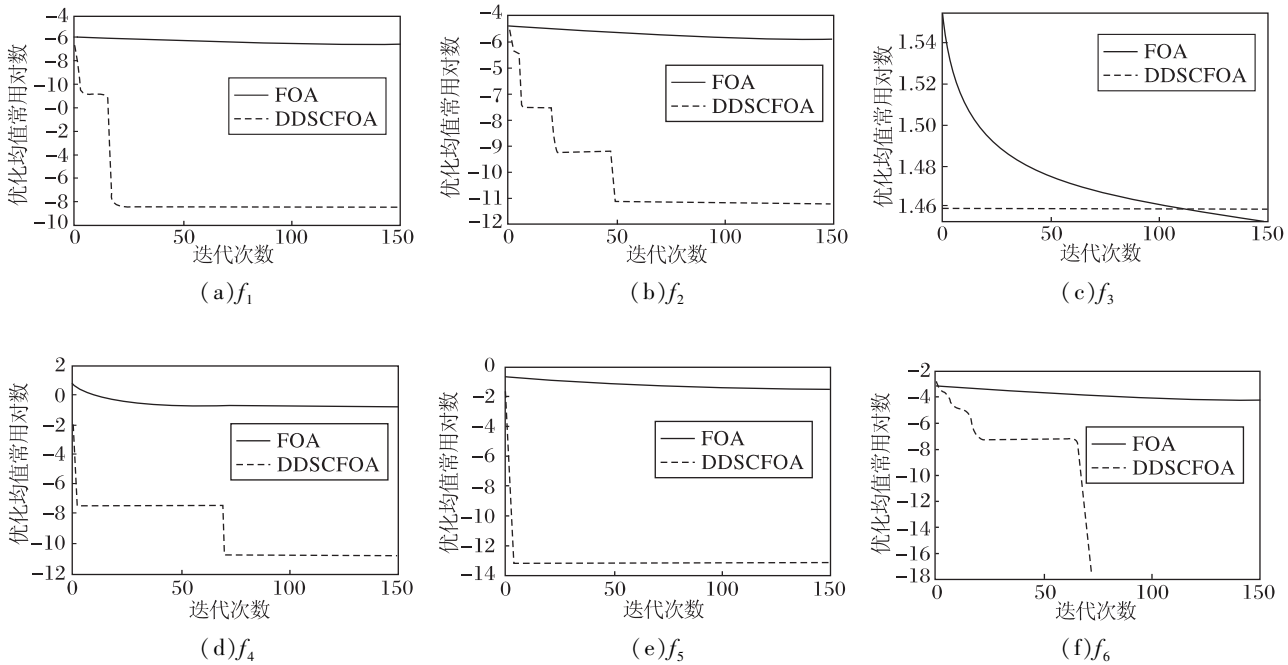


图 1　f_1 ~ f_6 在两个实验中优化均值进化曲线

Fig. 1　Optimization mean fitness evolutionary curves of f_1 ~ f_6 in 2 experiments

表 3 是本文 DDSCFOA 算法与文献中报道算法的优化均值比较,表中“—” 表示文献中没有报道. 由表 3 可看出,在试验条件(最大迭代次数) 较苛刻情况下,本文 DDSCFOA 算法比文献中报道算法具有更高的收敛精度(文献[9] 和[14] ~ [17] 的数据来自文献[8]).

4.2.2　固定收敛精度下的平均迭代次数与成功率

在表 1 中指定的收敛精度下,将 FOA 和 DDSCFOA 两种算法全局寻优函数最小值所需的平均迭代次数和成功率作为评价指标,6 个测试函数经过 20 次连续运行后的实验结果如表 4 所示. 表中,

$$\text{成功率} = \frac{\text{达到精度的运行次数}}{\text{总实验次数}};$$

平均迭代次数 = 达到精度的迭代次数算术平均;“ -” 表示迭代次数超过了最大迭代次数;“—” 表示文献中没有报道.

表 3　各算法的平均最小值

Table 3　Mean minimum of algorithms

算法	f_2	f_3	f_4	f_6
DDSCFOA	6.8122e-012	28.7177	1.5143e-011	0
SPSO[14]	0.13	34.1	101.7	—
LDWPSO[14]	0.0242	60.8	71.6	—
APSO[14]	0.0039	33.6	50.1	—
AFSA[9]	106.5802	1.09e+10	306.8284	0.049889
GAFSA[9]	0.000014	24.73051	66.74231	0.000583
文献[15]	0.4033	1610.359	46.4689	—
文献[16]	0.008614	39.118488	81.689550	0.002915
文献[17]	0.002095	50.798139	57.194136	0.000155

由表 4 可知,DDSCFOA 算法在所有函数上均获得 100% 的成功率,且在较少平均迭代次数(1 ~ 8.55 次) 内就达到本文指定的目标精度;而 FOA 算法除函数 f_3 外,在其它所有函数上的成功率都为 0,

即使在函数 f_3 上,所需的平均迭代次数也相当于 DDSCFOA 算法所需平均迭代次数的 47 倍. 总体来说 DDSCFOA 算法具有较高的优化性能和实用性.

由表 4 还可看出,与参考文献相比,DDSCFOA 在参数比较苛刻的情况下,在所有函数上均获得 100% 的成功率,所需的平均迭代次数最多只需文献[14]的 1.02%,文献[9]的 6.11%(文献[14]中 f_1, f_2, f_3, f_4 对应的目标精度分别为 0.01,0.1,100,100,$Maxgen$ = 10 000;文献[9]中 $f_1, f_2, f_3, f_4, f_5, f_6$ 对应的目标精度分别为 0.0001,0.0001,100,100,0.0001, 0.0001, $Maxgen$ = 2000). 总体来说,DDSCFOA 比文献[14]和文献[9]中算法具有更快的收敛速度和更高的收敛可靠性及实用性.

4.3 本文算法在高维函数上的优化性能测试实验

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[18-19]. 为突出说明本文算法在高维函数上性能,将 DDSCFOA 和 FOA、粒子群优化算法 PSO、混合蛙跳算法 SFLA 及文献[11]中的遗传算法 GA 和细菌觅食优化算法 BFO 进行比较,具体参数设置:$Sizepop$ = 15, $Maxgen$ = 800(与文献[11]保持一致),FOA 和 DDSCFOA 中搜索距离 $RandomValue \in [-1, +1]$;PSO 中惯性权重 $\omega = 0$, $c_1 = c_2 = 1.8$;$SFLA$ 中 $Sizepop$ = 200,种群数 n = 20,每个子群的个体个数 m = 10,混合迭代次数 $Maxgen_1$ = 500,内迭代次数 $Maxgen_2$ = 10.

1) 以连续运行 20 次所得的优化均值(全局寻优函数最小值的算术平均)和标准差作为算法性能的衡量指标. 6 种算法在 50 维函数 $f_1 \sim f_5$ 上的实验结果如表 5 所示,表中"—"表示文献中没有报道. 从中可看出,除 f_3 外,DDSCFOA 在其它函数上的优化均值和标准差均显著优于 FOA、PSO、SFLA 和文献[11]的 GA 和 BFO,即使在 f_3 上,DDSCFOA 的优化均值和标准差也与 FOA、文献[11]的 GA 和 BFO 相当,但比 PSO 和 SFLA 要优,说明 DDSCFOA 在高维函数上依然具有较高的优化性能,且在高维函数上的优化性能优于 FOA、PSO、SFLA 和文献[11]的 GA 和 BFO. 同时还可看出 DDSCFOA 所需的平均运行时间比 GA 和 BFO 要短,但比 FOA、PSO 和 SFLA 要长,说明 DDSCFOA 具有较好的实用性.

2) 为测试并比较本文算法 DDSCFOA 在高维函数上的优化性能随函数维数增加而变化的情况,$f_1 \sim f_5$ 维数从 60 ~ 100 变化,分别采用 FOA、DDSCFOA、PSO 和 SFLA 共 4 种算法对其进行优化,连续运行 20 次所得的优化均值(全局寻优函数最小值的算术平均)和相对变化率作为算法性能的衡量指标. 实验结果如表 6 所示,表中,

$$\text{相对变化率} = \frac{|\text{50 维的优化均值} - \text{100 维的优化均值}|}{\text{50 维的优化均值}},$$

50 维的优化均值取自表 5. 从中可看出,随着函数维数的增加,4 种算法的优化性能都降低,即优化均值增大;不论维数如何变化,依然是 DDSCFOA 的优化性能最好,即优化均值最小;4 种算法中 DDSCFOA 的相对变化率也是最小的. 4 种算法在 5 个函数上的优化均值随函数维数变化而变化的趋势线如图 2 ~ 图 6 所示,图中纵坐标用优化均值表示,横坐标为函数维数. 从中可看出,在所有函数上,PSO 和 SFLA 优化均值随函数维数增加单调递增,且维数越大,优化均值的增长率越大;在函数 f_3 上,FOA 和 DDSCFOA 优化均值随函数维数增加单调递增,但增长率较小;在其它函数上,FOA 和 DDSCFOA 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.

表 4 在目标精度下的平均迭代次数与成功率比较

Table 4 Comparisons of mean iterations and success rate for the goal

函数	测试项目	FOA	DDSCFOA	SPSO[14]	LDWPSO[14]	APSO[14]	AFSA[9]	GAFSA[9]
f_1	平均迭代次数 (成功率)	– (0)	2 (1)	528 (1)	2183 (1)	484 (1)	– (0)	575 (1)
f_2	平均迭代次数 (成功率)	– (0)	2 (1)	413 (0.74)	2133 (1)	454 (1)	– (0)	970 (1)
f_3	平均迭代次数 (成功率)	47 (1)	1 (1)	911 (1)	2477 (1)	994 (1)	– (0)	1076 (1)
f_4	平均迭代次数 (成功率)	– (0)	2 (1)	196 (0.58)	1710 (1)	420 (1)	– (0)	944 (1)
f_5	平均迭代次数 (成功率)	– (0)	2 (1)	— —	— —	— —	– (0)	580 (1)
f_6	平均迭代次数 (成功率)	– (0)	8.55 (1)	— —	— —	— —	1092 (0.18)	140 (1)

表 5　50 维函数上的算法性能比较

Table 5　Performances comparison of algorithms for 50 dimensions functions

算法	测试项目	f_1	f_2	f_3	f_4	f_5
FOA	优化均值	0.0015	7.3784*ee*−006	48.0099	0.0501	0.0356
	标准差	9.0597e−004	4.2560*ee*−006	0.3683	0.3035	0.0218
	平均运行时间/s	0.2845	0.4452	0.3685	0.3847	0.8945
DDSCFOA	优化均值	5.4186*ee*−012	1.8345*ee*−012	48.5559	2.9354*ee*−013	6.0077*ee*−012
	标准差	2.3814*ee*−011	5.6680*ee*−012	0.0284	1.3078*ee*−012	1.9079*ee*−011
	平均运行时间/s	1.2350	1.7215	1.0791	1.8199	2.5539
PSO	优化均值	1.3929*ee*+002	3.5890	1.3066*ee*+004	1.7653*ee*+002	1.5483*ee*+003
	标准差	2.5841*ee*+001	5.4631*ee*−001	9.2145*ee*+003	0.9857*ee*+001	4.2589*ee*+001
	平均运行时间/s	0.2456	0.5422	0.3211	0.3145	1.0879
SFLA	优化均值	8.5084	1.0560	3.0567*ee*+002	3.9047*ee*+001	5.7441*ee*+002
	标准差	3.6408	5.4666*ee*−002	9.6091*ee*+001	1.5787*ee*+001	2.2196*ee*+002
	平均运行时间/*s*	0.7796	1.2218	0.8016	1.1649	0.9382
GA[11]	优化均值	—	1.505	50.481	66.791	—
	标准差	—	1.15	35.48	11.57	—
	平均运行时间/s	—	—	32.938	—	—
BFO[11]	优化均值	—	1.028	50.64	3.73	—
	标准差	—	0.008	0.23	0.36	—
	平均运行时间/s	—	—	2.719	—	—

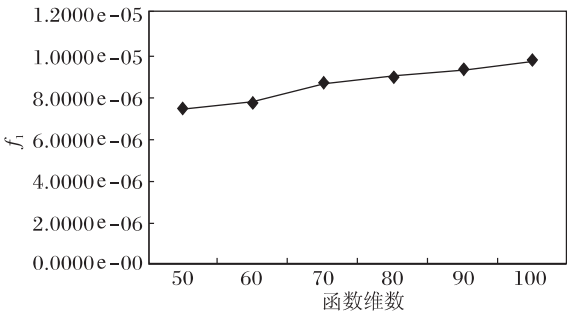
表 6　高维函数上的算法性能比较

Table 6　Performances comparison of algorithms for high dimensions functions

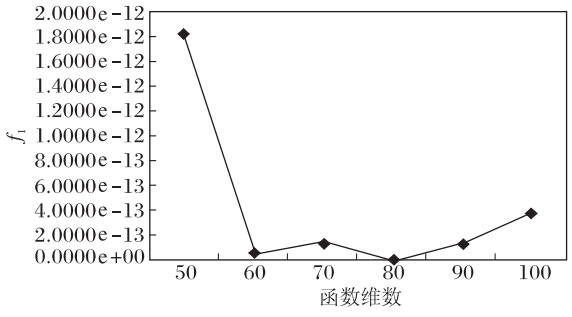
函数	维数	优化均值			
		FOA	DDSCFOA	PSO	SFLA
f_1	60	0.0020	1.0164*ee*−012	5.0069*ee*+002	4.8131*ee*+001
	70	0.0026	1.2140*ee*−011	1.3374*ee*+003	8.0165*ee*+001
	80	0.0032	4.0144*ee*−013	2.2484*ee*+003	1.0671*ee*+002
	90	0.0040	1.6595*ee*−012	4.9789*ee*+003	1.5238*ee*+002
	100	0.0046	4.0261*ee*−015	8.4757*ee*+003	1.9702*ee*+002
	相对变化率/%	206.67	99.81	2984.93	2215.59
f_2	60	7.6836*ee*−006	6.8184*ee*−014	5.8441	1.4745
	70	8.6716*ee*−006	1.5063*ee*−013	1.3239*ee*+001	1.7648
	80	8.8343*ee*−006	0.0000*ee*+000	2.5836*ee*+001	1.9489
	90	9.2611*ee*−006	1.3791*ee*−013	4.3462*ee*+001	2.3129
	100	9.6487*ee*−006	4.0494*ee*−013	6.3714*ee*+001	2.8082
	相对变化率/%	30.77	77.93	1675.27	165.93
f_3	60	5.8111*ee*+001	5.8522*ee*+001	1.0754*ee*+005	8.4379*ee*+002
	70	6.8097*ee*+001	6.8397*ee*+001	5.8677*ee*+005	1.1541*ee*+003
	80	7.8249*ee*+001	7.8295*ee*+001	7.9411*ee*+005	1.7732*ee*+003
	90	8.8463*ee*+001	8.8167*ee*+001	3.1087*ee*+006	2.3957*ee*+003
	100	9.8158*ee*+001	9.8038*ee*+001	6.3133*ee*+006	3.0716*ee*+003
	相对变化率/%	104.45	101.91	48218.55	904.87
f_4	60	5.0100*ee*−002	9.1104*ee*−012	2.4020*ee*+002	7.8004*ee*+001
	70	9.9800*ee*−002	2.3968*ee*−012	3.2715*ee*+002	8.7394*ee*+001
	80	9.9800*ee*−002	1.2974*ee*−011	3.9089*ee*+002	1.2883*ee*+002
	90	5.0100*ee*−002	2.7959*ee*−012	4.5872*ee*+002	1.4811*ee*+002
	100	1.4950*ee*−001	1.6449*ee*−013	5.4566*ee*+002	1.6290*ee*+002
	相对变化率/%	198.40	43.96	209.10	317.19
f_5	60	5.7100*ee*−002	2.3398*ee*−011	1.7979*ee*+004	1.4040*ee*+003
	70	8.5300*ee*−002	1.4941*ee*−010	3.8622*ee*+004	2.2956*ee*+003
	80	1.1940*ee*−001	9.7114*ee*−011	9.9625*ee*+004	4.0582*ee*+003
	90	1.6060*ee*−001	1.0808*ee*−013	1.8550*ee*+005	6.0444*ee*+003
	100	2.1580*ee*−001	5.4514*ee*−014	3.6932*ee*+005	8.0638*ee*+003
	相对变化率/%	506.18	99.09	23753.29	1303.84

综上所述,可得出本文算法 DDSCFOA 对高维函数仍有较强的优化能力;且随着函数维数的增加,DDSCFOA 算法较之 FOA、PSO 和 SFLA 来说,优化性能更加突出.

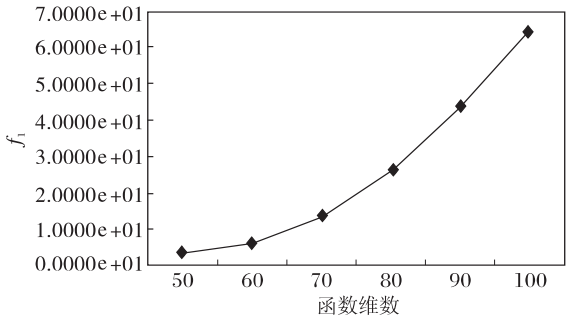
由上可知,本文提出的 DDSCFOA 比 FOA 和参考文献中报道的其它经典优化算法具有更快的收敛速度和更高的收敛精度,尤其在高维函数上的优化性能更加突出.



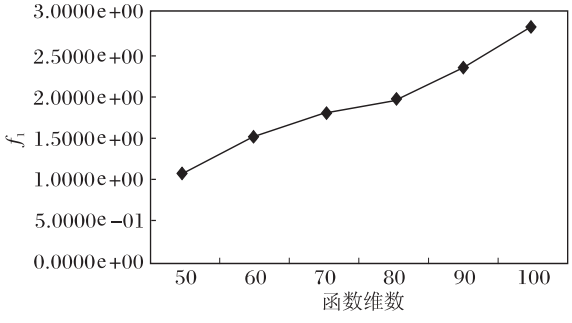
(a) FOA



(b) DDSCFOA



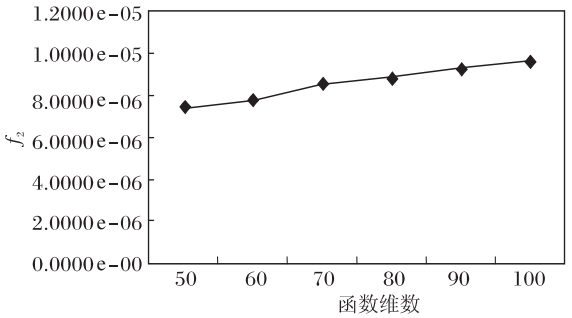
(c) PSO



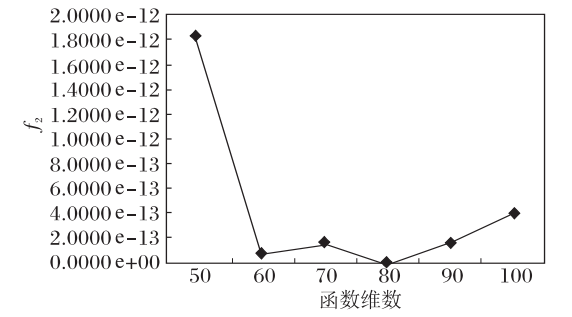
(d) SFLA

图 2 f_1 适应度随函数维数变化曲线

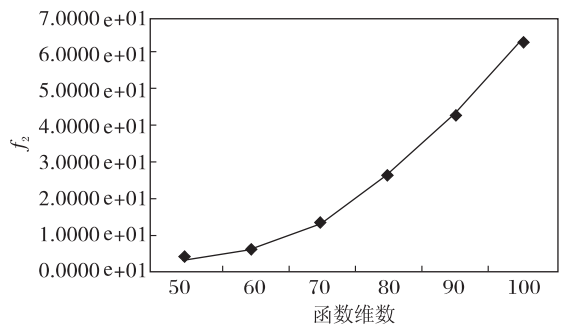
Fig. 2 Curve of f_1 fitness changing with dimensions



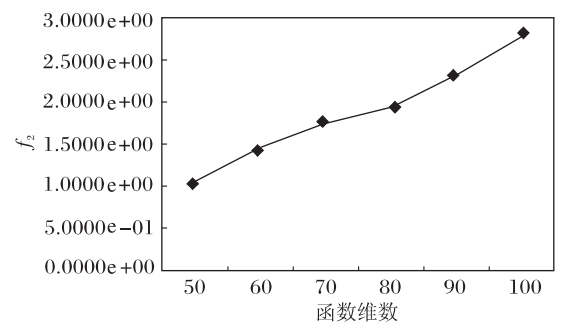
(a) FOA



(b) DDSCFOA



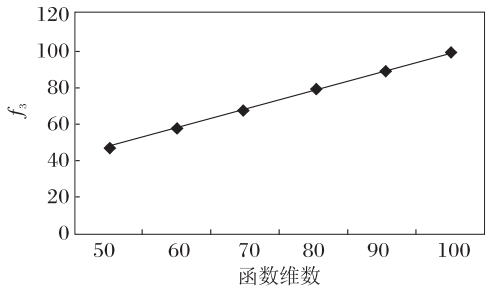
(c) PSO



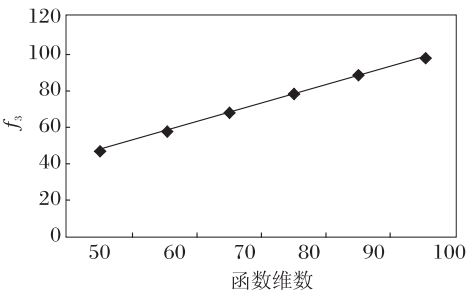
(d) SFLA

图 3 f_2 适应度随函数维数变化曲线

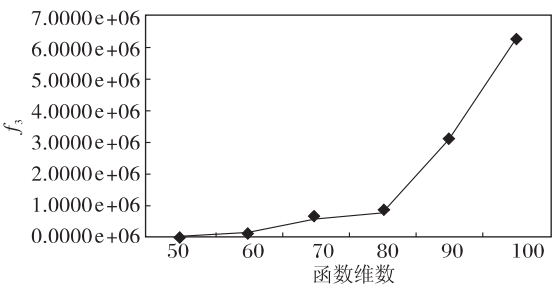
Fig. 3 Curve of f_2 fitness changing with dimensions



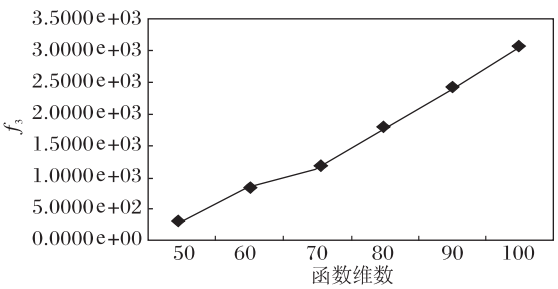
(a)FOA



(b)DDSCFOA



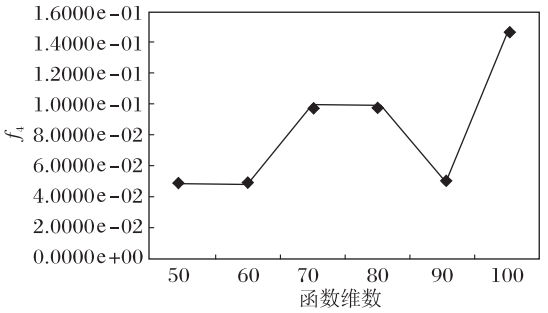
(c)PSO



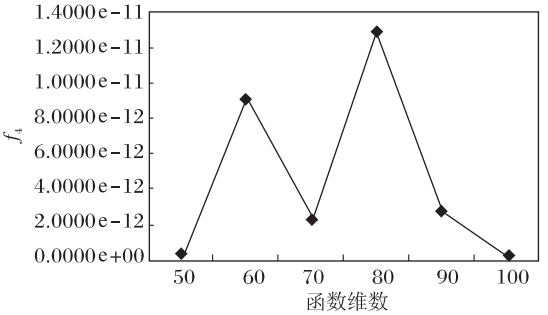
(d)SFLA

图 4　f_3 适应度随函数维数变化曲线

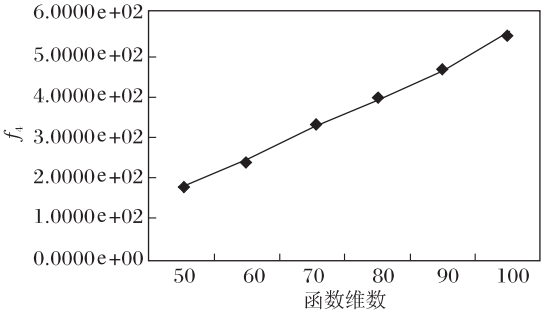
Fig. 4　Curve of f_3 fitness changing with dimensions



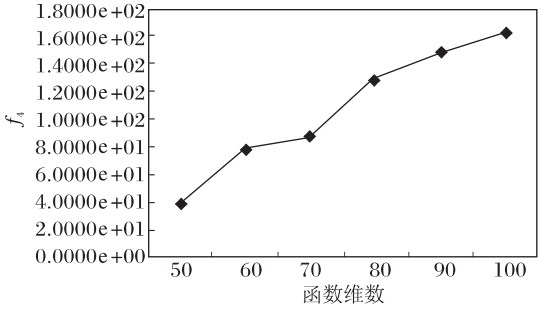
(a)FOA



(b)DDSCFOA



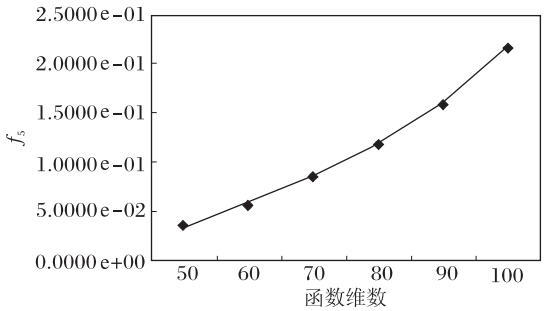
(c)PSO



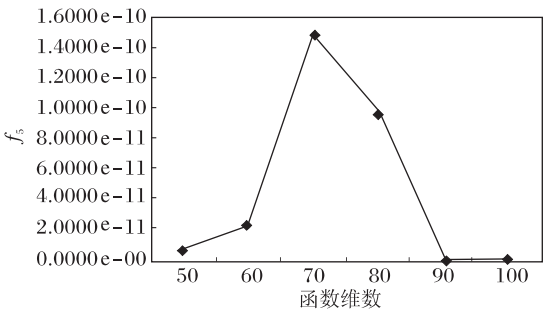
(d)SFLA

图 5　f_4 适应度随函数维数变化曲线

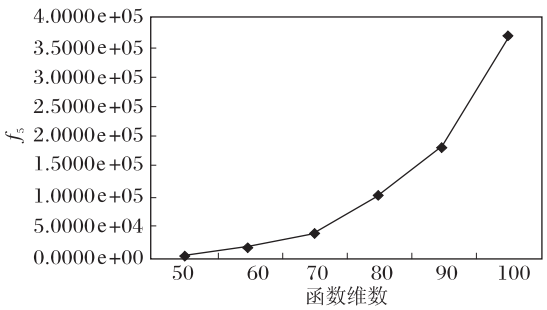
Fig. 5　Curve of f_4 fitness changing with dimensions



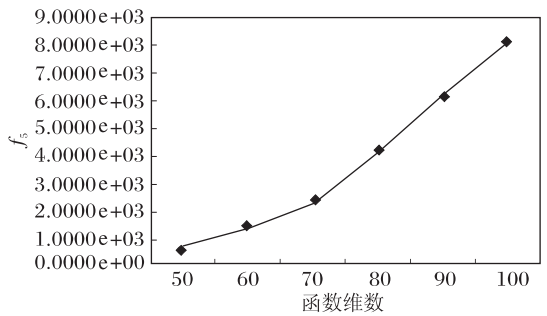
(a) FOA



(b) DDSCFOA



(c) PSO



(d) SFLA

图 6 f_5 适应度随函数维数变化曲线

Fig. 6 Curve of f_5 fitness changing with dimensions

5 结束语

针对果蝇优化算法寻优精度不高和易陷入局部最优的缺点,对其进行改进,提出一种动态双子群协同进化果蝇优化算法. 该算法在迭代进化过程中,根据种群的进化水平,动态地将整个种群划分为两个具有不同功能的子群,从而有效保证算法的"探索"和"开发"能力的平衡,提高算法的收敛精度和跳出局部极值的能力. 针对数值优化的实验结果也表明,该算法在收敛速度、收敛精度及收敛可靠性上均比基本果蝇优化算法有较大的提高.

参 考 文 献

[1] Pan W T. A New Fruit Fly Optimization Algorithm: Taking the Financial Distress Model as An Example. Knowledge-Based Systems, 2012, 26(2): 69-74

[2] Pan W T. Fruit Fly Optimization Algorithm. Taipei, China: Tsang Hai Book Publishing Co., 2011: 10-12 (in Chinese)
(潘文超. 果蝇最佳化演算法. 台北,中国: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. Journal of Taiyuan University of Technology: Social Sciences Edition, 2011, 29(4): 1-5 (in Chinese)
(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. 太原理工大学学报:社会科学版, 2011, 29(4): 1-5)

[4] Tian Yuan, Zhang Bide, Liu Daiwei, *et al*. Condition Trend Prediction of Hydroelectric Sets Based on Fruit Fly Optimization Algorithm Optimized GRNN. Water Resources and Power, 2012, 30(12): 127-129 (in Chinese)
(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. 水电能源科学, 2012, 30(12): 127-129)

[5] Wang Xin, Du Kang, Qin Bin, *et al*. Drying Rate Modeling Based on FOALSSVR. Control Engineering of China, 2012, 19(4): 630-633 (in Chinese)
(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, 2012, 19(4): 630-633)

[6] Wang Xuegang, Zou Zoujian. Identification of Ship Manoeuvring Response Model Based on Fruit Fly Optimization Algorithm. Journal of Dalian Maritime University, 2012, 38(3): 1-5 (in Chinese)
(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. 大连海事大学学报, 2012, 38(3): 1-5)

[7] Xiao Zhengan. Application of Improved FOA on Audio Signal Blind Separation [EB/OL]. [2012-06-05]. http://www.cnki.net/kcms/detail/11.2127.TP.20120605.1519.002.html (in Chinese)
(肖正安. 改进 FOA 算法在语音信号盲分离中的应用 [EB/OL]. [2012-06-05]. http://www.cnki.net/kcms/detail/11.2127.TP.20120605.1519.002.html

[8] Hu Wang, Li Zhishu. A Simpler and More Effective Particle Swarm Optimization Algorithm. Journal of Software, 2007, 18(4): 861-868 (in Chinese)
(胡 旺,李志蜀. 一种更简化而高效的粒子群优化算法. 软件学报, 2007, 18(4): 861-868)

[9] Wang Lianguo, Hong Yi, Shi Qiuhong. Global Edition Artificial Fish Swarm Algorithm. Journal of System Simulation, 2009, 21(23): 7483-7486 (in Chinese)
(王联国,洪 毅,施秋红. 全局版人工鱼群算法. 系统仿真学报, 2009, 21(23): 7483-7486)

[10] Ma D W, Guo X W, Deng L. Ammunition Scheduling of Carrier-Based Aircraft Based on Modified Ant Colony Algorithm. Journal of System Simulation, 2012, 24(6): 1207-1211 (in Chinese)
(马登武,郭小威,邓 力. 基于改进蚁群算法的舰载机弹药调度. 系统仿真学报, 2012, 24(6): 1207-1211)

[11] Hu Jie. Research on the Modified Bacteria Foraging Optimization Algorithm and Its Application. Ph. D Dissertation. Wuhan, China: Wuhan University of Technology, 2012 (in Chinese)
(胡 洁. 细菌觅食优化算法的改进及应用研究. 博士学位论文. 武汉:武汉理工大学, 2012)

[12] Lǚ Zhensu, Hou Zhirong. Particle Swarm Optimization with Adaptive Mutation. Acta Electronica Sinica, 2004, 32(3): 416-420 (in Chinese)
(吕振肃,侯志荣. 自适应变异的粒子群优化算法. 电子学报, 2004, 32(3): 416-420)

[13] LI Li, Niu Ben. Particle Swarm Optimization. Beijing, China: Metallurgical Industry Press, 2009 (in Chinese)
(李 丽,牛 奔. 粒子群优化算法. 北京:冶金工业出版社, 2009)

[14] Lin Chuan, Feng Quanyuan. New Adaptive Particle Swarm Optimization Algorithm. Computer Engineering, 2008, 34(7): 181-183 (in Chinese)
(林 川,冯全源. 一种新的自适应粒子群优化算法. 计算机工程, 2008, 34(7): 181-183)

[15] Angeline P J. Evolutionary Optimization versus Particles Swarm Optimization: Philosophy and Performance Differences // Proc of the 7th International Conference on Evolutionary Progranming. San Diego, USA, 1998: 601-610

[16] Clerc M, Kennedy J. The Particle Swarm-Explosion, Stability, and Convergence in a Multidimensional Complex Space. IEEE Trans on Evolutionary Computation, 2002, 6(1): 58-73

[17] Eberhart R C, Shi Y. Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization // Proc of the Congress on Evolutionary Computation. La Jolla, USA, 2000, I: 84-88

[18] Li M S, Ji T Y, Tang W J, *et al*. Bacterial Foraging Algorithm with Varying Population. BioSystems, 2010, 100(3): 185-197

[19] Jin Qibing, Zhao Zhenxing, Su Xianjing, *et al*. PSO Algorithm with High Speed Convergence Based on Particle Health. Journal of Chemical Industry and Engineering, 2011, 62(8): 2328-2333 (in Chinese)
(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. 化工学报, 2011, 62(8): 2328-2333)

动态双子群协同进化果蝇优化算法

作者： 韩俊英， 刘成忠， 王联国， HAN Jun-Ying， LIU Cheng-Zhong， WANG Lian-Guo
作者单位： 甘肃农业大学 信息科学技术学院 兰州730070
刊名： 模式识别与人工智能

ISTIC EI PKU

英文刊名： Pattern Recognition and Artificial Intelligence

年，卷(期)： 2013(11)

参考文献(19条)

1. Pan W T A New Fruit Fly Optimization Algorithm:Taking the Financial Distress Model as An Example 2012(02)

2. 潘文超 果蝇最佳化演算法 2011

3. 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[期刊论文]-太原理工大学学报(社会科学版) 2011(04)

4. 田源;张彼德;刘代伟 基于果蝇优化算法的GRNN水电机组状态趋势预测[期刊论文]-水电能源科学 2012(12)

5. 王欣;杜康;秦斌 基于果蝇优化算法的LSSVR干燥速率建模[期刊论文]-控制工程 2012(04)

6. 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 2012(03)

7. 肖正安 改进 FOA 算法在语音信号盲分离中的应用 2012

8. 胡旺;李志蜀 一种更简化而高效的粒子群优化算法[期刊论文]-软件学报 2007(04)

9. 王联国;洪毅;施秋红 全局版人工鱼群算法[期刊论文]-系统仿真学报 2009(23)

10. 马登武;郭小威;邓力 基于改进蚁群算法的舰载机弹药调度[期刊论文]-系统仿真学报 2012(06)

11. 胡洁 细菌觅食优化算法的改进及应用研究 2012

12. 吕振肃;侯志荣 自适应变异的粒子群优化算法[期刊论文]-电子学报 2004(03)

13. 李丽;牛奔 粒子群优化算法 2009

14. 林川;冯全源 一种新的自适应粒子群优化算法[期刊论文]-计算机工程 2008(07)

15. Angeline P J Evolutionary Optimization versus Particles Swarm Optimization:Philosophy and Performance Differences 1998

16. Clerc M;Kennedy J The Particle Swarm-Explosion,Stability,and Convergence in a Multidimensional Complex Space 2002(01)

17. Eberhart R C;Shi Y Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization 2000

18. Li M S;Ji T Y;Tang W J Bacterial Foraging Algorithm with Varying Population 2010(03)

19. 靳其兵;赵振兴;苏晓静 基于粒子健康度的快速收敛粒子群优化算法[期刊论文]-化工学报 2011(08)