

果蝇优化算法的分析^①

韩 虎

(兰州交通大学 电子与信息工程学院, 兰州 730070)

摘 要: 本文针对果蝇优化算法 FOA 由于参数选取不合理而导致运行时间长或陷入局部最优值的问题, 研究了果蝇优化算法主要参数对算法运行时间、优化性能和收敛速度的影响. 以 FOA 全局寻优 6 个标准测试函数最小值为例, 在不同的参数配置下, 进行仿真实验, 对比分析研究得出果蝇优化算法各主要参数对算法性能影响的定性结论, 并给出了各参数恰当的合理取值区间, 以便在算法性能和运行时间之间找到最好的平衡. 试验结果表明, 参数的合理设置, 不但缩短了算法的运行时间, 而且使算法具有较快的收敛速度和较高的收敛精度.

关键词: 果蝇优化算法; 群体智能; 参数分析; 优化性能; 收敛速度

Analysis on Fruit Fly Optimization Algorithm

HAN Hu

(School of Electronic and Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China)

Abstract: In order to overcome the demerits of Fruit Fly Optimization Algorithm(FOA), such as long running time and easily relapsing into local optimum, which are caused by improper parameters setting, this paper mainly researches how the main parameters of FOA influence the following aspects: running time, optimization performance and convergence velocity. Taking six standard testing functions optimization as an example, we make the simulation experiments of FOA under the different parameters. Firstly, qualitative conclusions of influence of important parameters on FOA's performance are drawn by comparative analysis and study. And then the reasonable value ranges of the parameters are given in order to get a better balance between FOA's performance and its running time. Experimental results show that reasonable parameters settings not only shorten FOA's running time, but also speed up its convergence velocity and improve its optimization precision.

Key words: Fruit Fly Optimization Algorithm; swarm intelligence; parameters analysis; optimization performance; convergence velocity

果蝇优化算法(Fruit Fly Optimization Algorithm, FOA)由中国台湾学者潘文超博士于 2011 年 6 月首次提出, 由于算法本身的良好性能, 尤其是该算法是迄今为止所需调整参数最少、进化方程最简单^[1-10]的一种群智能优化算法, 自提出以来, 就得到了国内外学者的广泛关注, 吸引了越来越多不同背景研究人员的注意力^[11-34], 成为交叉学科中一个前沿性研究问题.

智能算法合理的参数取值对算法性能至关重要, 但参数之间的相互影响、相互之间的复杂关系及对算法性能的叠加影响很难研究清楚, 一般都是针对具体

的研究问题, 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.

本文首先分析了果蝇优化算法 FOA 的时间复杂度, 然后对该算法的 3 个可调整参数: 种群规模、迭代步进值和最大迭代次数分别进行分析研究, 其中主要包括各参数取值对算法收敛速度、收敛精度以及算法时间复杂度的影响. 以求 6 个基准测试函数最小值为例进行仿真试验, 得出各参数对算法性能及算法时间复杂度影响的定性结论; 并就各参数的最优取值, 给

^① 基金项目: 甘肃省自然科学基金(1506RJYA006); 兰州市科技局计划项目(214162); 甘肃省教育厅基金项目(42015268)

收稿时间: 2016-05-25; 收到修改稿时间: 2016-06-30 [doi:10.15888/j.cnki.csa.005602]

出了通过大量试验总结出来的比较恰当的经验数值作为参考。

1 果蝇优化算法基本原理

1.1 觅食行为描述

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,甚至能嗅到 40 公里以外的食物源。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^[2]。自然界生活的一群果蝇,在整个觅食过程中,首先,各自发挥自己的嗅觉优势,搜集漂浮在空气中的各种气味,判定优质食物源的方向,然后,朝该方向飞去;最后,等飞近食物位置后,再通过敏锐的视觉靠近食物源与同伴聚集的准确位置。

1.2 基本步骤

果蝇优化算法 FOA 是通过模拟自然界果蝇群体觅食行为而提出的一种基于群体智能的全局随机搜索算法,果蝇群体迭代觅食过程示意图如图 1 所示。该算法将种群的全局搜索和个体的信息交换相结合,在全局搜索基础上,通过个体的信息交换,更新全局最优解,最终在满足最大迭代次数或收敛目标精度条件下终止算法。

果蝇优化算法归纳为以下几个必要的步骤^[1]。

(1) 初始化参数: 种群规模 N , 最大迭代数 M , 随机初始化果蝇群体位置 X_axis, Y_axis ;

(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, 即迭代步进值;

$$X_i = X_axis + RandomValue \quad (1)$$

$$Y_i = Y_axis + RandomValue$$

(3) 由于无法得知食物位置, 因此先估计果蝇个体与原点之距离 $Dist_i$, 再计算果蝇个体味道浓度判定值 S_i , 此值为距离之倒数;

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

$$S_i = 1 / Dist_i \quad (3)$$

(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 $Smell_i$;

$$Smell_i = Function(S_i) \quad (4)$$

(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);

$$[bestSmell \ bestindex] = \min(Smell_i) \quad (5)$$

(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;

$$Smell_{best} = bestSmell$$

$$X_axis = X(bestindex) \quad (6)$$

$$Y_axis = Y(bestindex)$$

(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, 并且当前迭代次数小于最大迭代数 M , 若是则执行步骤(6)。

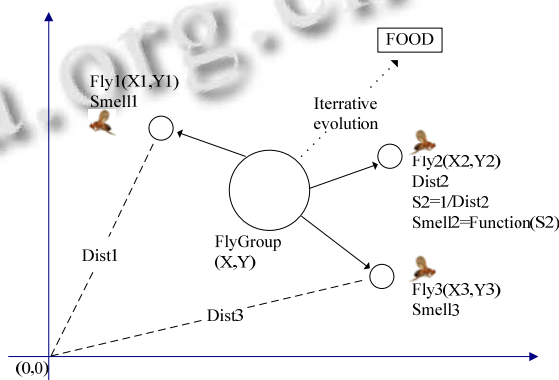


图 1 果蝇群体迭代搜索食物示意图

2 果蝇优化算法时间复杂度分析

2.1 算法时间复杂度分析

在果蝇优化算法 FOA 中, 种群中的所有个体都要进行迭代搜索, 直至达到最大迭代次数或算法达到收敛目标精度要求为止。设种群规模为 N , 最大迭代次数为 M , 每个个体迭代需要的时间为 T , 则果蝇优化算法 FOA 的时间复杂度为 $O(N \cdot M \cdot T)$ 。由于循环体内的计算量, 即每个个体迭代需要的计算量为 $O(1)$, 所以 FOA 算法主要的计算量取决于算法中 N 和 M 的 2 重循环。

2.2 算法参数分析

种群规模 N 是群智能优化算法的一个重要参数。首先, 由上述果蝇优化算法时间复杂度分析可以看出, 当种群规模 N 增大时, 算法的计算时间将会成正比增大; 然后从算法原理的直观分析可以得出, 当种群规模 N 增大时, 提高了种群的多样性, 算法的全局搜索能力会增强, 算法的收敛精度会提高, 算法可以尽早找到最优解, 加速收敛速度。

最大迭代次数 M 是群智能优化算法的另一个重要参数。首先, 由果蝇优化算法时间复杂度分析可以

得出,当最大迭代次数 M 增大时,算法的计算时间也将会正比增大;并且从算法原理的直观分析可以得出,当最大迭代次数 M 增大时,算法收敛到全局最优解的可能性也会增大,即全局搜索能力也会增强,算法的收敛精度也会提高。

迭代步进值 $RandomValue$,即果蝇个体利用嗅觉搜寻食物的随机方向与距离,是果蝇优化算法特有的一个可调整参数,它的取值对算法的收敛速度和收敛精度有非常重要的影响。当迭代步进值 $RandomValue$ 增大时,将会加快算法的收敛速度,但同时可能会降低算法的收敛精度;同理,当迭代步进值 $RandomValue$ 减小时,将会减慢算法的收敛速度,但同时有可能提高算法的收敛精度。

因此,有必要就各参数取值对算法收敛速度、收敛精度以及算法时间复杂度的影响进行详细的定量分析研究,并通过仿真试验就各参数的最优取值给出经验数值作为参考。

3 仿真实验及结果分析

以求 6 个基准测试函数最小值为例,进行仿真实验,对比分析研究果蝇优化算法的各参数对该算法收敛精度、收敛速度及时间复杂度的影响。

测试函数名称、函数形式、搜索区间、函数最优值见表 1。测试软件平台为 Windows XP, Matlab7.1, 机器主频为 P4(1.7G), 内存为 1G。

表 1 6 个测试函数

函数	函数形式	搜索区间	函数最优值
Sphere f_1	$f_1(x) = \sum_{i=1}^n x_i^2$	[-100,100]	0
Griewank f_2	$f_2(x) = 1/4000 \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	[-600,600]	0
Rastrigin f_3	$f_3(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$	[-5.12,5.12]	0
Quadric f_4	$f_4(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j^2)$	[-100,100]	0
Ackley f_5	$f_5(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos 2\pi x_i\right) + 20 + e$	[-100,100]	0
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

3.1 种群规模对算法性能的影响

种群规模 N 分别取值 5, 10, 15, 20, 评估种群规模对算法收敛精度和时间复杂度的影响, 试验中其他参数设置为: 最大进化迭代次数 $M = 150$ 次, 迭代步进值 $RandomValue \in [-1, +1]$; 随机初始化果蝇群体位置为表 1 中各函数的搜索区间。

将果蝇优化算法 FOA 全局寻优函数最小值所得的优化均值、平均运行时间和收敛精度相对变化率作为评价指标。6 个测试函数经过 50 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, 该值体现了算法的收敛精度, 值越小, 说明算法收敛精度越高。平均运行时间=算法运行时

间的算术平均, 即算法运行 1 次所需的平均时间。收敛精度相对变化率=(低种群规模对应的优化均值-高种群规模对应的优化均值)/低种群规模对应的优化均值, 该值体现了算法的收敛精度随着种群规模增大而提高的幅度, 值越大, 说明算法收敛精度提高的幅度越大。从中可以看出, 随着种群规模增大, 算法收敛精度提高了, 平均运行时间呈正比增大; 但是, 收敛精度相对变化率却随着种群规模的增大总体呈现下降趋势。6 个测试函数的优化均值随种群规模增大而变化的趋势线如图 2 所示, 图中纵坐标用优化均值表示, 横坐标为种群规模。从中也可以看出, 在种群规模增大的初期(5~10), 在所有函数上, 优化均值随种群规

模增大单调递减, 并且相对变化率最大, 即算法的收敛精度提高幅度最大; 在种群规模增大的中期(10~15), 优化均值继续随种群规模增大单调递减, 但是相对变化率却在减小; 在种群规模增大的后期(15~20), 在函数 f_3 和 f_4 上, 优化均值随种群规模增大继续减小, 但是相对变化率也更小了, 甚至在函数 f_1 和 f_2 上, 当种群规模增大时优化均值反而也增大了. 因此, 综合上述分析可以得出以下两个结论: (1)增大种群规模, 必然使得算法运行耗费的计算时间成正比增大; (2)虽然增大种群规模, 可以提高算法收敛精度; 但持续一味地增大种群规模, 并未带来算法收敛精度的更进一步

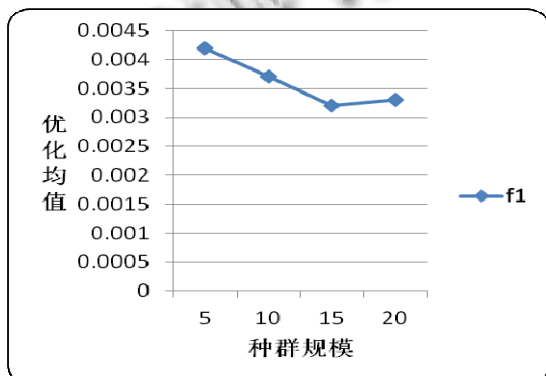
显著提高, 这是因为该算法迭代后期易陷入局部最优所致. 因此, 恰当的种群规模可以在算法性能和运行时间之间找到最好的平衡, 本文建议种群规模为 10~15 之间较为合适.

3.2 最大迭代次数对算法性能的影响

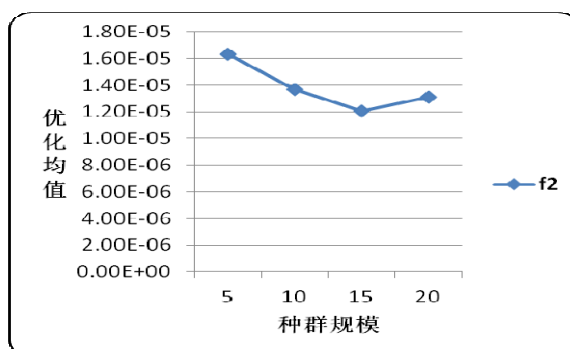
最大迭代次数 M 分别取值 10, 50, 150, 500, 1000, 评估最大迭代次数对算法收敛精度和时间复杂度的影响, 试验中其他参数设置为: 迭代步进值 $RandomValue \in [-5, +5]$, 种群规模 $N = 15$, 随机初始化果蝇群体位置为表 1 中各函数的搜索区间.

表 2 种群规模对算法性能的影响

函数	评价指标	种群规模 N			
		5	10	15	20
f_1 (30 维)	优化均值	0.0042	0.0037	0.0032	0.0033
	平均运行时间(秒)	0.0187	0.0366	0.0522	0.0713
	收敛精度相对变化率(%)	—	11.9048	13.5135	-3.1250
f_2 (30 维)	优化均值	1.6290e-005	1.3654e-005	1.2088e-005	1.3103e-005
	平均运行时间(秒)	0.0236	0.0460	0.0685	0.0912
	收敛精度相对变化率(%)	—	16.1817	11.4691	-8.3968
f_3 (30 维)	优化均值	0.1545	0.1287	0.1270	0.1066
	平均运行时间(秒)	0.0220	0.0437	0.0641	0.0855
	收敛精度相对变化率(%)	—	16.6990	1.3209	16.0629
f_4 (30 维)	优化均值	0.0631	0.0535	0.0497	0.0477
	平均运行时间(秒)	0.0663	0.1315	0.1953	0.2609
	收敛精度相对变化率(%)	—	15.2139	7.1028	4.0241
f_5 (30 维)	优化均值	1.6887	1.6882	1.6878	1.6877
	平均运行时间(秒)	0.0218	0.0424	0.06425	0.0845
	收敛精度相对变化率(%)	—	0.0296	0.0237	0.0059
f_6 (2 维)	优化均值	1.0316e-004	7.7351e-005	6.7075e-005	6.2247e-005
	平均运行时间(秒)	0.0163	0.0354	0.0465	0.0622
	收敛精度相对变化率(%)	—	25.0184	13.284896	7.197913



(a) f_1



(b) f_2

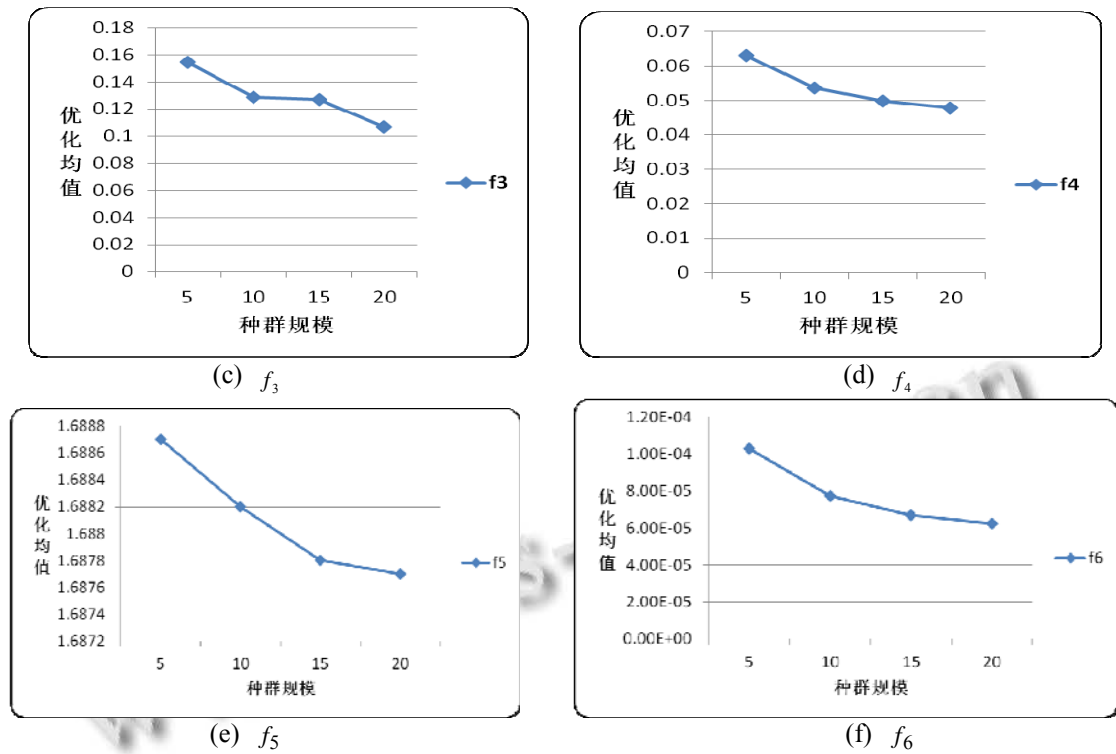


图 2 $f_1 \sim f_6$ 种群规模与优化均值的关系

将果蝇优化算法 FOA 全局寻优函数最小值所得的优化均值和算法运行时间作为评价指标。6 个测试函数经过 50 次连续运行后的实验结果如表 3 所示，表中优化均值=全局寻优函数最小值的算术平均。平均运行时间=算法运行时间的算术平均，即算法运行 1 次所需的平均时间。从中可以看出最大迭代次数 M 越大，优化均值精度(算法收敛精度)越高；同时，平均运行时间也呈正比增大。6 个测试函数的优化均值随最大迭代次数 M 增大而变化的趋势线如图 3 所示，图中纵坐标用优化均值表示，横坐标为最大迭代次数。从中可以看出，在最大迭代次数增大的初期(10~50)，在所有函数上，优化均值随最大迭代次数增大单调递减，并且相对递减幅度最大，即随着最大迭代次数增大，算法的收敛精度明显提高了，而且提高的幅度最大；在

最大迭代次数增大的中期(50~500)，优化均值继续随最大迭代次数增大单调递减，但是相对递减幅度却在减小；在最大迭代次数增大的后期(500~1000)，在函数 f_1 、 f_2 和 f_4 上，优化均值随最大迭代次数增大继续减小，但是相对递减幅度也更小了，甚至在函数 f_3 上，当最大迭代次数增大时优化均值反而增大了。

因此，综合上述分析可以得出以下两个结论：(1)增大最大迭代次数，必然使得算法运行耗费的计算时间正比增大；(2)虽然增大最大迭代次数，可以使算法收敛精度得到提高；但持续增大最大迭代次数，并未带来算法收敛精度的持续的进一步明显提高，这是因为该算法迭代后期易陷入局部最优所致。因此，恰当的最大迭代次数可以在算法性能和运行时间之间找到最好的平衡，本文建议最大迭代次数为 500 左右比较合适。

表 3 最大迭代此时对算法性能的影响

函数	评价指标	最大迭代次数				
		10	50	150	500	1000
f_1 (30 维)	优化均值	0.0058	0.0024	7.0926e-004	1.0905e-004	3.1033e-005
	平均运行时间(秒)	0.00449	0.01931	0.05624	0.16933	0.34176
f_2 (30 维)	优化均值	1.9852e-005	1.0440e-005	5.9672e-006	2.1443e-006	8.9791e-007
	平均运行时间(秒)	0.00627	0.02637	0.07670	0.227119	0.453457
f_3 (30 维)	优化均值	0.2525	0.1631	0.1601	0.1195	0.1990

	平均运行时间(秒)	0.005272	0.023436	0.069740	0.211272	0.424389
f_4 (30 维)	优化均值	0.0921	0.0346	0.0102	0.0015	4.3641e-004
	平均运行时间(秒)	0.014890	0.068044	0.200738	0.631023	1.262140
f_5 (30 维)	优化均值	1.6902	1.6867	1.6862	1.6862	1.6861
	平均运行时间(秒)	0.0047	0.0210	0.2091	0.6977	1.4081
f_6 (2 维)	优化均值	1.6889e-004	3.2693e-005	5.5702e-006	6.0958e-007	1.5871e-007
	平均运行时间(秒)	0.0034	0.0157	0.0466	0.1578	0.3063

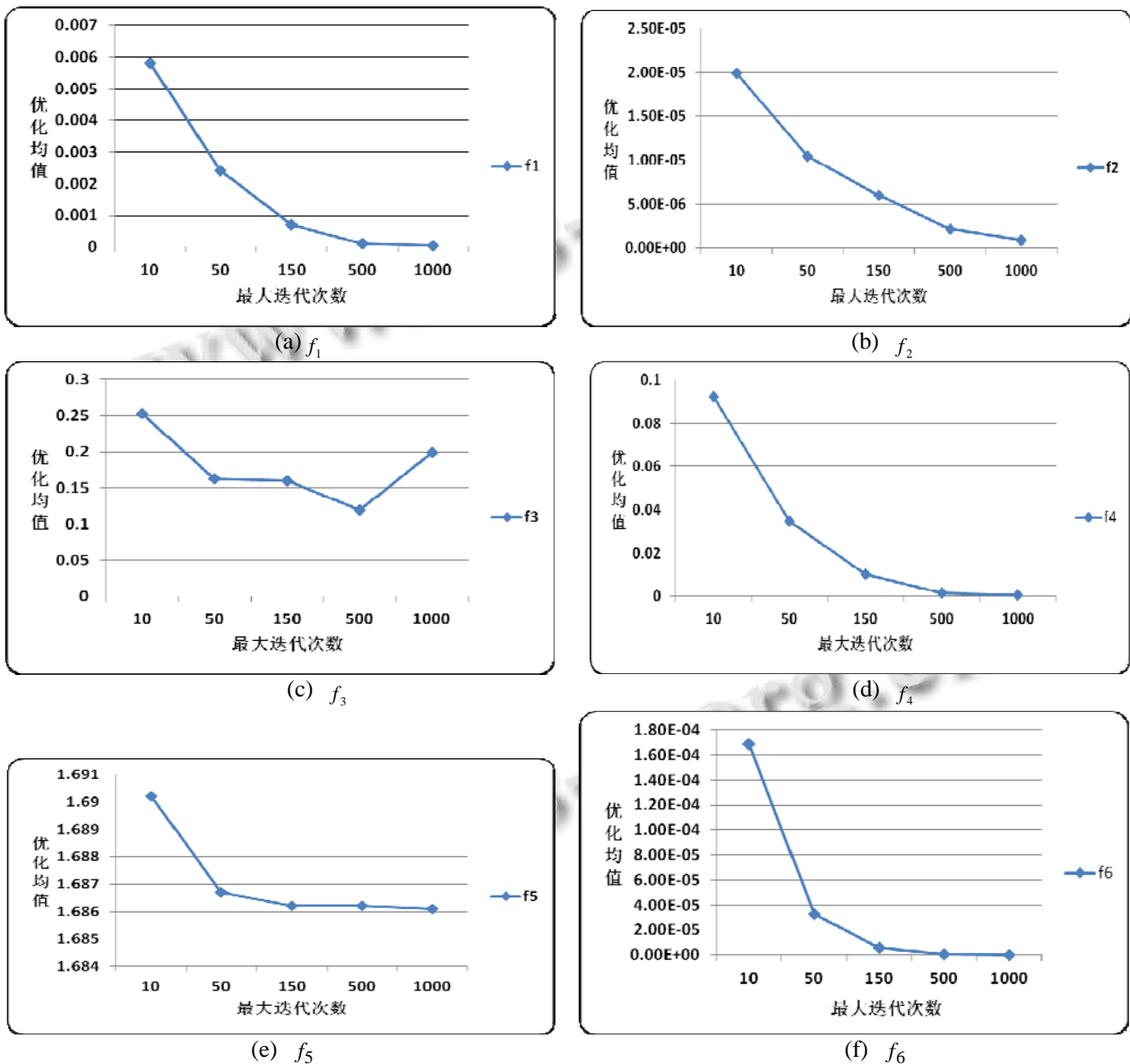


图 3 $f_1 \sim f_6$ 最大迭代次数与优化均值的关系

3.3 迭代步进值对算法收敛速度和收敛精度的影响

实验中参数设置为：固定最大进化迭代次数 $M = 150$ 次，种群规模 $N = 15$ ，搜索距离分别在三个不同的区间取值： $RandomValue \in [-1,+1]$ ，

$RandomValue \in [-5,+5]$ ， $RandomValue \in [-10,+10]$ ，随机初始化果蝇群体位置为表 1 中各函数的搜索区间。评估迭代步进值对算法收敛速度和收敛精度的影响。

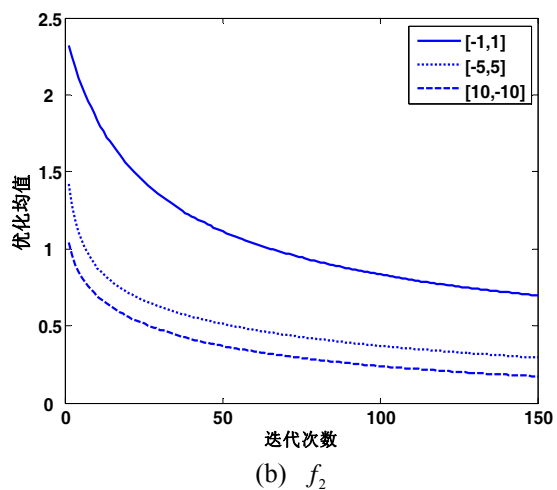
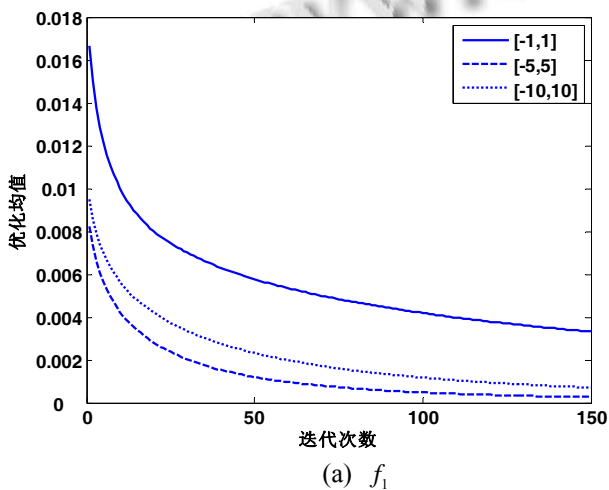
将果蝇优化算法 FOA 全局寻优函数最小值所得

的优化均值、标准差和收敛速度作为评价指标. 6 个测试函数经过 50 次连续运行后的实验结果如表 4 所示, 表中优化均值=全局寻优函数最小值的算术平均. 从中可以看出, 当 $RandomValue \in [-1,+1]$ 扩大到 $RandomValue \in [-5,+5]$ 进行比较时, 优化均值精度得到了比较明显的提高, 即算法的收敛精度明显提高了; 但当 $RandomValue \in [-5,+5]$ 扩大到 $RandomValue \in [-10,+10]$ 进行比较时, 优化均值精度并没有明显的提高, 而是基本持平, 即算法的收敛精度相当. 同时, 可以看出标准差的取值一直比较稳定, 说明算法的稳定性很好. 图 4 是 FOA 的迭代步进值分别在三个不同的区间取值, 全局寻优 6 个测试函数最小值所得的优化均值对数值进化曲线(注: 为了方便进化曲线的显示和观察, 本文对所有函数的优化均值取以 10 为底的对数), 图中实线是 $RandomValue \in [-1,+1]$

的优化均值进化曲线, 图中点线是 $RandomValue \in [-5,+5]$ 的优化均值进化曲线, 图中虚线是 $RandomValue \in [-10,+10]$ 的优化均值进化曲线. 进化曲线也表明, 总体来看, 随着迭代步进值 $RandomValue$ 取值区间的扩大, FOA 算法收敛速度和收敛精度明显提高了; 图中点线和虚线相对于实线来说, 收敛速度和收敛精度具有较大幅度提高; 但虚线相对于点线来说, 即 $RandomValue \in [-5,+5]$ 扩大到 $RandomValue \in [-10,+10]$ 进行比较时, 收敛速度和收敛精度基本持平. 因此, 总体来说, 恰当的迭代步进值不但可以使算法具有较高的收敛精度和收敛速度, 而且, 为了有效平衡算法的“探索”和“开发”能力, 基于上述实验结果, 本文建议迭代步进值 $RandomValue \in [-5,+5]$ 比较合适.

表 4 迭代步进值对算法性能的影响

函数	评价指标	迭代步进值		
		$RandomValue \in [-1,+1]$	$RandomValue \in [-5,+5]$	$RandomValue \in [-10,+10]$
f_1 (30 维)	优化均值	0.0033	7.0663e-004	2.6199e-004
	标准差	4.2160e-004	5.7279e-005	1.5723e-005
f_2 (30 维)	优化均值	1.3940e-005	5.8542e-006	3.4467e-006
	标准差	6.0507e-006	8.8185e-007	3.6454e-007
f_3 (30 维)	优化均值	0.0877	0.1001	9.3075e-005
	标准差	0.2703	0.3022	3.7880e-006
f_4 (30 维)	优化均值	0.0474	0.0102	0.0038
	标准差	0.0065	6.5545e-004	1.7585e-004
f_5 (30 维)	优化均值	1.6877	1.6853	1.6849
	标准差	4.3665e-004	4.0114e-005	1.1978e-005
f_6 (2 维)	优化均值	6.6754e-005	5.7094e-006	1.5777e-006
	标准差	1.8673e-005	4.9432e-007	9.3956e-008



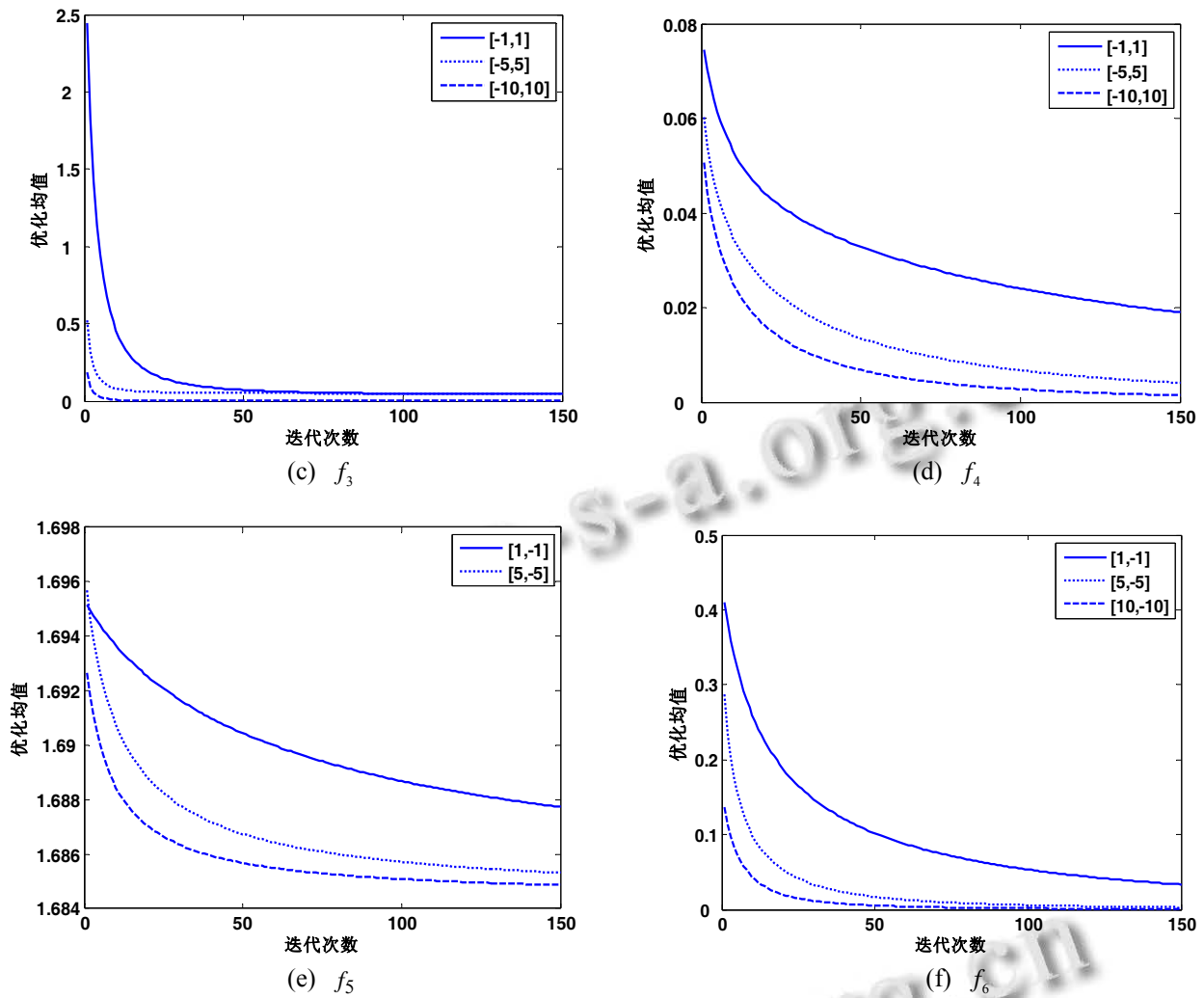


图 4 $f_1 \sim f_6$ 在三个实验中优化均值进化曲线

4 结语

分析了果蝇优化算法的 3 个可调整参数对算法收敛精度、收敛速度、收敛稳定性和时间复杂度等的影响；通过大量仿真实验，分析总结了随着算法参数取值的变化，果蝇优化算法的各方面性能变化的规律，并给出了各参数的恰当取值区间，在算法性能和算法时间复杂度之间找到最好的平衡，在不增大算法耗费计算机时间资源的情况下，使算法具有更好的优化性能和更高的实用性。

参考文献

1 Pan WT. A new fruit fly optimization algorithm: Taking the financial distress model as an example. Knowledge-Based Systems, 2012, 26(1): 69–74.
 2 Elbeltagi E, Hegazy T, Grierson D. Comparison among five

evolutionary-based optimization algorithms. Advanced Engineering Informatics, 2005, 19(1): 43–53.
 3 Dai HD, Zhao GR, Lu JH, Dai SW. Comment and improvement on “A new fruit fly optimization algorithm: Taking the financial distress model as an example”. Knowledge-Based Systems, 2014, 59: 159–160.
 4 Pan QK, Sang HongY, Duan JunH, Gao L. An improved fruit fly optimization algorithm for continuous function optimization problems. Knowledge-Based Systems, 2014, 62: 69–83.
 5 Liu CZ, Huang GB, Zhang RZ, Chai Q. A modified fruit fly optimization algorithm with better balance between exploration and exploitation. Proc. of the 2nd International Conference on Progress in Informatics and Computing (PIC-2014). Shanghai. 2014. 55–60.
 6 潘文超.果蝇最佳化演算法.台北:沧海书局,2011:10–12.
 7 韩俊英,刘成忠,王联合国.动态双子群协同进化果蝇优化算法.

- 模式识别与人工智能,2013,26(11):1057-1067.
- 8 韩俊英,刘成忠.反向认知的高效果蝇优化算法.计算机工程,2013,39(11):223-225.
- 9 韩俊英,刘成忠.基于细菌趋化的果蝇优化算法.计算机应用,2013,33(4):964-966.
- 10 韩俊英,刘成忠.应用反向学习策略的果蝇优化算法.计算机应用与软件,2014,31(4):157-160.
- 11 Zhang YW, Cui GM, Wang Y, Guo X, Zhao S. An optimization algorithm for service composition based on an improved FOA. *Tsinghua Science and Technology*, 2015, 20(1): 90-99.
- 12 金一. 机刻光栅制造系统结构特性与精度控制研究[博士学位论文].合肥:中国科学技术大学,2013,64-78.
- 13 Sheng W, Bao Y. Fruit fly optimization algorithm based fractional order fuzzy-PID controller for electronic throttle. *Nonlinear Dynamics*, 2013, 73 (1-2): 611-619.
- 14 Han J, Wang P, Yang X. Tuning of PID controller based on fruit fly optimization algorithm. *International Conference on Mechatronics and Automation(ICMA)*. 2012. 409-413.
- 15 Liu Y, Wang XJ, Li YJ. A modified fruit-fly optimization algorithm aided PID controller designing. *Proc. of 10th World Congress on Intelligent Control and Automation*. Beijing. 2012. 233-238.
- 16 Li H, Guo S, Li C, Sun J. A hybrid annual power load forecasting model based on generalized regression neural network with fruit fly optimization algorithm. *Knowledge-based Systems*, 2013,37: 378-387.
- 17 Li HZ, Guo S, Zhao HR, Su CB, Wang B. Annual electric load forecasting by a least squares support vector machine with a fruit fly optimization algorithm. *Energies*, 2012, 5: 4430-4445.
- 18 Lin SM. Analysis of service satisfaction in web auction logistics service using a combination of fruit fly optimization algorithm and general regression neural network. *Neural Computing & Applications*, 2013, 22(3-4): 783-791.
- 19 孙立,董君伊,李东海.基于果蝇算法的过热汽温自抗扰优化控制.清华大学学报(自然科学版),2014,54(10):1288-1292.
- 20 郑晓龙,王凌,王圣尧.求解置换流水线调度问题的混合离散果蝇算法.控制理论与应用,2014,31(2):159-164.
- 21 Zhang P, Wang L. Grouped fruit-fly optimization algorithm for the no-wait lot streaming flow shop scheduling. *International Conference on Intelligent Computing*. Springer International Publishing. 2014. 664-674.
- 22 Wang WC, Liu XG. Melt index prediction by least squares support vector machines with an adaptive mutation fruit fly optimization algorithm. *Chemometrics and Intelligent Laboratory Systems*, 2015, 141: 79-87.
- 23 Zheng XL, Wang L, Wang SY. A novel fruit fly optimization algorithm for the semiconductor final testing scheduling problem. *Knowledge-Based Systems*, 2014, 57: 95-103.
- 24 Wang L, Zheng XL, Wang SY. A novel binary fruit fly optimization algorithm for solving the multidimensional knapsack problem. *Knowledge-Based Systems*, 2013, 48: 17-23.
- 25 杨琼,俞立峰,陈小小.一种基于果蝇优化方法的连续查询攻击算法.四川大学学报(自然科学版),2014,51(4): 725-730.
- 26 陈世和,张玉文,李东海,老大中.循环流化床锅炉燃烧系统的自抗扰控制器优化设计.控制理论与应用,2013, 30(12):1589-1594.
- 27 郭凡,丁永生,郝矿荣,任立红,肖纯材.基于果蝇算法优化支持向量回归机的纺丝性能预测.系统仿真学报,2014, 26(10):2360-2364.
- 28 刘志雄,王雅芬,张煜.多种群果蝇优化算法求解自动化仓库拣选作业调度问题.武汉理工大学学报,2014,36(3): 71-77.
- 29 龚玉坛,王大兴,刘化清,赵玉华.黄土塬少井区采用果蝇优化算法构建虚拟井方法.地球物理学进展,2014,29(2): 674-680.
- 30 李霞,孙灵芳,杨明.基于改进 FOA 匹配追踪的超声信号处理研究.仪器仪表学报.2013,34(9),2068-2073.
- 31 周平,白广忱.基于神经网络与果蝇优化算法的涡轮叶片低循环疲劳寿命健壮性设计.航空动力学报,2013,5: 1013-1018.
- 32 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.武汉理工大学学报,2012,34(3):63-68.
- 33 王欣,杜康,秦斌,徐海军.基于果蝇优化算法的 LSSVR 干燥速率建模控制工程.控制工程,2012,19(4),630-633.
- 34 郑伟,孙雪青,郝冬梅,吴颂红.结合 Shearlet 变换和果蝇优化算法的甲状腺图像融合.激光杂志,2014,35(9):70-73,78.