

PSO 和 AFSA 混合优化算法

王联国^{1,2}, 施秋红¹, 洪毅²

(1. 甘肃农业大学信息科学技术学院, 兰州 730070; 2. 兰州理工大学电气工程与信息工程学院, 兰州 730030)

摘要: 结合粒子群优化(PSO)算法和人工鱼群算法(AFSA)的优势, 提出一种 PSO-AFSA 混合算法。将种群分为 2 个子群体, 在每次迭代中, 一个子群体利用 PSO 算法进化, 另一个子群体利用 AFSA 进化, 2 个算法共享整个种群极值信息。通过混合算法对 5 个标准函数进行实验, 并与标准 PSO 算法进行比较, 结果表明混合算法具有更好的优化性能。

关键词: 粒子群优化算法; 人工鱼群算法; PSO-AFSA 混合算法; 群体智能

Hybrid Optimization Algorithm of PSO and AFSA

WANG Lian-guo^{1,2}, SHI Qiu-hong¹, HONG Yi²

(1. School of Information Science and Technology, Gansu Agricultural University, Lanzhou 730070;

2. College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou 730030)

【Abstract】 This paper proposes a hybrid algorithm of Particle Swarm Optimization(PSO) and Artificial Fish Swarm Algorithm(AFSA) by combining the advantages of PSO algorithm and AFSA. Hybrid algorithm divides the swarm into two sub-groups. In each iteration, one sub-group evolves using PSO algorithm, the other sub-group evolves using AFSA, and two algorithms share the information of groups extremum. Through comparing PSO-AFSA hybrid algorithm with standard PSO algorithm in evolving solution to five standard functions, results show that PSO-AFSA hybrid algorithm outperforms PSO algorithm.

【Key words】 Particle Swarm Optimization(PSO) algorithm; Artificial Fish Swarm Algorithm(AFSA); PSO-AFSA hybrid algorithm; swarm intelligence

1 概述

粒子群优化(Particle Swarm Optimization, PSO)算法是 Kennedy 和 Eberhart 于 1995 年提出的基于群体智能方法的演化计算技术^[1], 其思想来源于对鸟群等生物种群觅食行为的研究和对一个简化社会模型的仿真。粒子群优化算法简单、容易实现, 同时又有深刻的智能背景, 因此, 在短短十几年间, 获得了很快的发展, 出现大量的研究成果, 并在函数优化、神经网络训练、模式分类、模糊系统控制以及其他工程领域得到了广泛的应用。然而, 同其他智能优化算法一样, 粒子群优化算法也易陷入局部极值点, 在进化后期收敛慢, 粒子趋于同一化, 失去多样性, 优化精度较差。因此, 许多研究者提出了大量的改进算法^[2-6], 这些改进算法在一定程度上改善了基本粒子群算法的性能。

人工鱼群算法^[7](Artificial Fish Swarm Algorithm, AFSA)是受鱼群行为的启发而提出的一种基于动物行为的群体智能优化算法, 是行为主义人工智能的一个典型应用。该算法具有良好的克服局部极值、取得全局极值的能力, 算法中仅使用了目标问题的函数值, 对搜索空间有一定的自适应能力, 并具有对初值与参数选择不敏感、鲁棒性强、简单易实现和使用灵活等诸多优点。

本文将人工鱼群算法改进后引入到粒子群优化算法中, 提出了一种 PSO 和 AFSA 混合的优化算法。该算法既利用了 PSO 算法追逐当前全局最优点来保证算法的收敛性, 又利用人工鱼群算法的搜索随机性, 加大了算法的搜索范围, 克服了 PSO 陷入局部极值点和 AFSA 算法运行速度慢等缺点。仿真结果表明, 该混合算法具有更好的优化性能。

2 基本 PSO 和改进的 AFSA

2.1 基本 PSO

设一个包含 M 个粒子的粒子群在 D 维空间飞行, 粒子群可用如下参数来表示: $x_i=(x_{i,1}, x_{i,2}, \dots, x_{i,D})$ 为粒子 i 在 D 维空间中的当前位置, $v_i=(v_{i,1}, v_{i,2}, \dots, v_{i,D})$ 为粒子 i 在 D 维空间中的飞行速度, $P_i=(P_{i,1}, P_{i,2}, \dots, P_{i,D})$ 为粒子 i 迄今为止搜索到的最优位置, $P_g=(P_{g,1}, P_{g,2}, \dots, P_{g,D})$ 为整个粒子群迄今为止搜索到的最优位置。粒子 i 在 d 维子空间中的飞行速度和位置按下式调整:

$$v_{id} = wv_{id} + c_1r_1(P_{id} - x_{id}) + c_2r_2(P_{gd} - x_{id}) \quad (1)$$

$$x_{id} = x_{id} + v_{id} \quad (2)$$

$$\begin{cases} v_{id} = v_{\max} & \text{if } v_{id} > v_{\max} \\ v_{id} = v_{\min} & \text{if } v_{id} < v_{\min} \end{cases} \quad (3)$$

其中, $i=1, 2, \dots, M$; $d=1, 2, \dots, D$; w 为惯性权重; c_1 和 c_2 是加速常数; r_1 和 r_2 是 $[0, 1]$ 之间的随机数; $[v_{\min}, v_{\max}]$ 是粒子运动速度的限制范围。

一般情况下, 速度更新公式中的 w 由最大加权因子 w_{\max} 线性减小到最小加权因子 w_{\min} , 即:

$$w_t = w_{\max} - t \frac{w_{\max} - w_{\min}}{T} \quad (4)$$

其中, t 是当前迭代数; T 是总的迭代数; $w_{\max}=0.9$; $w_{\min}=0.4$ 。

基金项目: 甘肃省教育信息化发展战略研究基金资助项目(2007-08)

作者简介: 王联国(1968-), 男, 副教授、博士研究生, 主研方向: 计算智能, 智能信息处理; 施秋红, 硕士研究生; 洪毅, 教授、博士生导师

收稿日期: 2009-09-23 **E-mail:** wanglg@gsau.edu.cn

2.2 改进的AFSA

在基本人工鱼群算法的基础上做以下改进：首先，用整个鱼群的中心位置和全局极值位置代替人工鱼的邻域中心位置和邻域极值位置，从而减少了算法的计算量，提高了运算速度；其次，按照式(5)动态调整人工鱼的视野 $Visual$ 和步长 $Step$ ，较好地平衡了全局搜索能力和局部搜索能力；最后，在人工鱼的觅食行为中，让人工鱼直接移动到搜索到的较好位置，加快了搜索速度。在觅食行为中，重复尝试次数较 try_numbe 少时，为人工鱼提供了随机游动的机会，增加了种群的多样性，从而能跳出局部极值^[8]。

$$\begin{cases} Visual = Visual \times a + Visual_{min} \\ Step = Step \times a + Step_{min} \\ a = \exp(-30 \times (t/T_{max})^s) \end{cases} \quad (5)$$

一般情况下 $Visual$ 初值为 $X_{max}/2$ (X_{max} 为搜索范围的最大值)； $Step$ 为 $Visual/8$ ； $Visual_{min}=0.01$ ； $Step_{min}=0.002$ ； t 为当前迭代次数； T_{max} 为最大迭代次数； s 为 a 曲线由大到小变化的速率，本文 s 取值范围为 $[1,30]$ 。

3 PSO和AFSA混合优化算法

图1为 PSO 和 AFSA 混合优化算法(PSO-AFSA)的流程，该混合算法的基本思想是：将种群分为 2 个子群体，在每次迭代中，一个子群体利用 PSO 算法进化，另一个子群体利用人工鱼群算法进化，然后求出整个群体迄今为止搜索到的最优解。这种算法既利用了 PSO 算法追逐当前全局最优值来保证算法的收敛性，又利用人工鱼群算法的搜索随机性，加大了算法的搜索范围，克服了 PSO 陷入局部极值点和 AFSA 算法运行速度慢等缺点，从而兼顾了算法的优化精度和效率，提高了算法的优化性能。

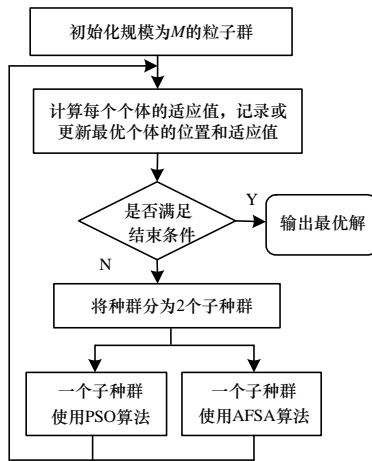


图1 PSO-AFSA 混合算法流程

4 仿真实验

4.1 实验设计

本文以求 5 个基准测试函数的最小值为例，进行仿真实验，评价混合算法的优化性能。测试软件平台为 Visual C++ 和 Windows XP，机器主频为 P4 (1.7 GHz)，内存为 512 MB。

$$f_1(x) = \sum x_i^2$$

$$f_2(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

$$f_3(x) = \sum_{i=1}^{n-1} (100(x_{i+1}^2 - x_i)^2 + (1-x_i)^2)$$

$$f_4(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$$

$$f_5(x) = 0.5 + \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}$$

实验中算法具体参数设置为：种群规模为 20， w 惯性权重因子按式(4)线性递减， $w_{max}=0.9$ ， $w_{min}=0.4$ ， $c_1=c_2=2$ ； $try_numbe=3$ ， $Visual$ 、 $Step$ 按式(5)动态调整， $s=3$ ；各函数优化的参数和目标精度见表1。用基本粒子群优化算法 PSO 和 PSO-AFSA 混合算法分别对上述 5 个测试函数的最小值寻优，最终测试结果采用独立运行 50 次后的平均值。

表1 用于测试改进算法的基准函数参数

函数	维数	搜索范围	理论最优值	目标精度
f_1	30	[-100,100]	0	10^{-5}
f_2	30	[-600,600]	0	10^{-5}
f_3	30	[-30,30]	0	100
f_4	30	[-5.12,5.12]	0	100
f_5	2	[-100,100]	0	10^{-5}

性能评价采用如下方法：(1)固定进化迭代次数，评估算法收敛速度和精度；(2)固定收敛精度目标值，评估算法达到该精度所需的迭代次数；(3)与文献中的优化性能进行比较。

4.2 实验结果及分析

4.2.1 固定进化迭代次数的收敛速度和精度

固定进化迭代次数为 2 000，算法独立运行 50 次，实验结果如表 2 和图 2~图 6 所示。由表 2 可以看出，PSO-AFSA 算法的平均优化结果和最优结果明显好于基本 PSO 算法。图 2~图 6 是函数 f_1 、 f_2 、 f_3 、 f_4 和 f_5 采用 PSO 算法和 PSO-AFSA 算法运行 50 次后得到的平均值的进化曲线，为了避免函数值为 0，对函数值均加上 10^{-7} 作为截止值。从图中可以看出，PSO-AFSA 优化精度较高，同时从 500 代后，PSO-AFSA 收敛速度加快。PSO-AFSA 与 PSO 相比，在收敛精度和收敛速度方面都有显著的提高。

表2 2种优化算法得计算结果

函数	算法	平均适应值	最小适应值	最大适应值
f_1	PSO	6.622 7E-08	1.491 5E-10	6.788 5E-07
	PSO-AFSA	8.582 4E-38	3.864 5E-50	2.081 3E-38
f_2	PSO	1.833 6E-02	5.606 4E-09	1.001 2E-01
	PSO-AFSA	8.986 8E-04	1.023 5E-50	1.231 6E-02
f_3	PSO	8.412 2E+01	2.653 734 7	4.164 8E+02
	PSO-AFSA	2.527 5E+01	1.544 4E+01	2.643 7E+01
f_4	PSO	4.988 8E+01	2.586 8E+01	8.556 6E+01
	PSO-AFSA	4.377 8E-01	1.101 3E-13	1.989 918 23
f_5	PSO	1.360 2E-03	0.000 000 00	9.715 9E-03
	PSO-AFSA	0.000 000 00	0.000 000 00	0.000 000 00

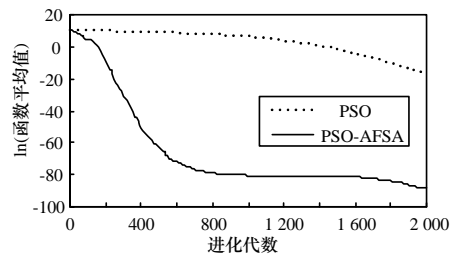


图2 函数 f_1 平均值的进化曲线

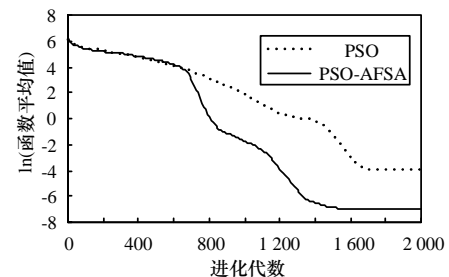


图3 函数 f_2 平均值的进化曲线

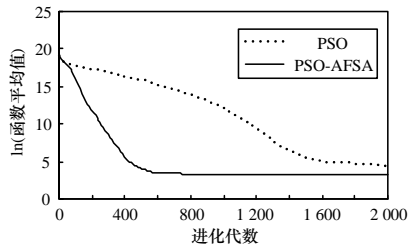


图4 函数 f_3 平均值的进化曲线

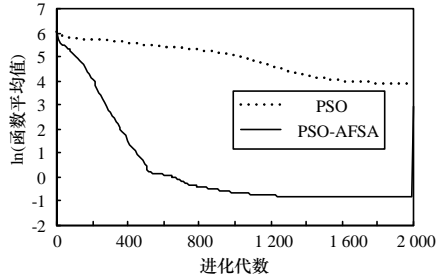


图5 函数 f_4 平均值的进化曲线

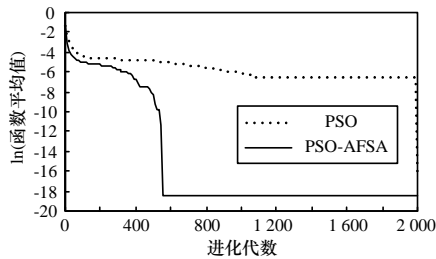


图6 函数 f_5 平均值的进化曲线

4.2.2 固定收敛精度下的进化迭代次数

表3为5个测试函数在表1指定的收敛精度下独立运行50次后的迭代次数(最大迭代次数为2000),其中,成功率=达到目标精度的运行次数÷总实验次数。由表3可以看出: PSO算法对 f_1 、 f_4 和 f_5 函数达到98%以上的成功率,对 f_2 和 f_3 函数分别达到18%和84%的成功率; PSO-AFSA算法对 f_1 函数达到了84%的成功率,对其他4个测试函数都获得了100%的成功率;而且 PSO-AFSA算法对6个测试函数达到目标优化精度的平均迭代次数、最小迭代次数和最大迭代次数都比 PSO 算法少很多。以上结果说明 PSO-AFSA 收敛速度快,优化效果比 PSO 好。

表3 在固定目标精度下的进化迭代次数

函数	算法	成功率/(%)	平均迭代次数	最小迭代次数	最大迭代次数
f_1	PSO	98	1 825	1 710	1 903
	PSO-AFSA	100	571	525	601
f_2	PSO	18	1 835	1 783	1 893
	PSO-AFSA	84	1 201	731	1 765
f_3	PSO	84	1 656	1 475	1 981
	PSO-AFSA	100	439	288	734
f_4	PSO	100	1 198	882	1 461
	PSO-AFSA	100	163	102	455
f_5	PSO	92	855	438	1 709
	PSO-AFSA	100	317	17	555

4.2.3 与参考文献中的优化性能比较

测试函数 f_2 、 f_3 和 f_4 的维数分别设置为10、20和30,相应的循环迭代次数分别设置为1000次、1500次和2000次,其他参数同上。对每个函数进行50次实验,计算算法找到函数平均最优值,并与文献[9]的优化结果进行比较,实验结果见表4。从表4可以看出, PSO-AFSA 算法搜索到的平均最优值远优于文献[9]给出的 PSO、BDPSO 和 IPSO 的优化结果,说明 PSO-AFSA 具有较高的优化性能。

表4 一些改进粒子群优化算法的性能比较

函数	维数	迭代次数	PSO	BDPSO	IPSO	PSO-AFSA
f_2	10	1 000	0.096 5	0.091 5	0.078 4	0.021 3
	20	1 500	0.035 3	0.020 2	0.023 6	0.003 8
	30	2 000	0.014 7	0.010 4	0.016 5	0.000 9
f_3	10	1 000	38.897 1	20.939 5	10.517 2	5.252 7
	20	1 500	86.108 7	57.070 0	75.724 6	15.289 0
	30	2 000	122.661 4	63.207 3	99.803 8	25.275 0
f_4	10	1 000	4.238 9	3.452 5	3.292 8	2.137 5
	20	1 500	21.568 8	20.074 3	16.413 7	1.684 8
	30	2 000	50.216 5	34.660 7	35.018 9	0.437 8

5 结束语

PSO-AFSA 是基于 PSO 和 AFSA 的一种混合进化算法,在利用 AFSA 搜索随机性的同时保持了 PSO 的搜索性能,提高了算法在解空间的探索能力和收敛精度,克服了基本粒子群优化算法容易陷入局部最优的缺陷。通过对5个标准数值函数的优化,结果表明,混合算法具有更好的优化性能。

参考文献

- [1] Kennedy J, Eberhart R. Particle Swarm Optimization[C]//Proc. of IEEE Int'l Conf. on Neural Networks. Perth, Australia: [s. n.], 1995: 1942-1948.
- [2] Eberhart R, Kennedy J. A New Optimizer Using Particle Swarm Theory[C]//Proc. of the 16th International Symposium on Micro Machine and Human Science. Nagoya, Japan: IEEE Press, 1995: 39-43.
- [3] Shi Yuhui, Eberhart R C. A Modified Particle Swarm Optimizer[C]//Proc. of IEEE International Conference on Evolutionary Computation. Piscataway, USA: IEEE Press, 1998: 69-73.
- [4] Higashi N, Iba H. Particle Swarm Optimization with Gaussian Mutation[C]//Proc. of the IEEE Swarm Intelligence Symposium. Indianapolis, Indiana, USA: IEEE Press, 2003: 72-79.
- [5] 胡建秀, 曾建潮. 具有随机惯性权重的 PSO 算法[J]. 计算机仿真, 2006, 27(2): 164-167.
- [6] 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.
- [7] 李晓磊, 邵之江, 钱积新. 一种基于动物自治体的寻优模式: 鱼群算法[J]. 系统工程理论与实践, 2002, 22(11): 32-38.
- [8] 王联国, 洪毅, 赵付青, 等. 一种改进的人工鱼群算法[J]. 计算机工程, 2008, 34(19): 192-194.
- [9] Jiang Yan, Hu Tiesong, Huang Chongchao, et al. An Improved Particle Swarm Optimization Algorithm[J]. Applied Mathematics and Computation, 2007, 193(1): 231-239.

编辑 任吉慧