

文章编号: 1004-4353(2018)04-0322-06

# 变步长和拥挤度因子的自适应人工鱼群算法

林 敏

(莆田学院 信息工程学院, 福建 莆田 351100)

**摘要:** 为了改进传统的人工鱼群算法会随着迭代的深入而导致算法易陷入局部最优的问题,以及固定的参数导致算法收敛慢和求解精度不高的问题,提出了一种改进的人工鱼群算法. 首先结合迭代次数,为移动步长引入一个权值;然后以每条人工鱼的视野范围所构成的子群为小生境,结合子群最优解与当前人工鱼状态,为拥挤度因子引入一个变异策略. 数值实验结果表明,本文提出的算法收敛速度快、精度高、鲁棒性强,优于传统的人工鱼群算法和文献[4]提出的算法.

**关键词:** 人工鱼群算法; 子群; 移动步长; 权值; 拥挤度因子; 变异策略

**中图分类号:** TP18

**文献标识码:** A

## Self-adaptive improved artificial fish swarm algorithm with changing step and crowding factor

LIN Min

(*Electronics and Engineering College, Putian University, Putian 351100, China*)

**Abstract:** In order to improve the traditional artificial fish swarm algorithm, which is easy to fall into local optimum, and the fixed parameters lead to slow convergence and low precision of the algorithm, during the deepening of the iteration, an improved artificial fish swarm algorithm is proposed. Firstly, by combining the number of iterations, a weight value is introduced for the moving step. Then, taking the subgroup of each artificial fish as the niche, combining the optimal solution of the subgroup with the current state of artificial fish, a variation strategy was introduced for crowding factor. The results of numerical experiments show that the proposed algorithm has the advantages of fast convergence, high accuracy and strong robustness, and is superior to the traditional artificial fish swarm algorithm and the algorithm proposed in literature [4].

**Keywords:** artificial fish swarm algorithm; subgroup; moving step length; weight factor; crowding factor; mutation strategy

2002 年,李晓磊等<sup>[1]</sup>提出了一种新型的元启发式优化算法——人工鱼群算法. 该算法的特点是:对初值和参数选择要求不高,鲁棒性、并行性好,全局搜索能力强,收敛速度快.但随着优化问题复杂度的增加,人工鱼群算法也突显出一些不足:前期收敛速度快,后期收敛速度慢;容易陷入局部最优解;虽然能快速定位最优解所在的解域,却很难得到高精度的解.对此,学者们提出了一些改进算法<sup>[2-11]</sup>,例如:朱旭辉<sup>[2]</sup>在人工鱼群算法中通过增加可行解的接受条件,减少鱼群的觅食行为次数,调整每条人工鱼的移动步长,提高了算法的运行效率.吴昌友<sup>[3]</sup>给出了一种新的初始鱼群产生方法,通过为鱼群的 3 个算子引入自适应移动步长和变异策略,提高了算法的全局搜索能力.易正俊<sup>[4]</sup>利用反向点原理重生鱼群,并采用正态分布动态调整拥挤度因子,提高了算法的收敛速度和最优值精度.廖孝勇等<sup>[5]</sup>基于膜计算思

想,通过引入差异因子,提出了一种能够提升运行速度和收敛精度的改进的自适应鱼群算法.李君等<sup>[6]</sup>将最速下降法嵌入到基本人工鱼群算法中,改善了算法的运行速度、收敛速度和收敛精度.

上述改进算法虽然在一定程度上提升了人工鱼群算法的优化性能,但在算法的收敛速度和收敛精度方面仍需进一步地提高.为此,本文从每条人工鱼的视野范围所构成的子群入手,分析迭代次数与鱼群的步长、拥挤度因子之间的关联,以及它们对整体优化性能的影响,提出一种变步长和拥挤度因子的自适应人工鱼群改进算法,并通过数值仿真实验验证本文方法的有效性.

## 1 标准人工鱼群算法

### 1.1 相关定义

假设初始鱼群数量为  $N$ , 解的维度为  $D$ , 个体搜索的上限和下限分别为  $Ub$  和  $Lb$ , 按下式进行初始化:

$$X_i = Lb + \text{rand}(1, D) \cdot (Ub - Lb). \quad (1)$$

初始化后,则随机生成  $N$  条人工鱼,用  $Y_i = f(X_i)$  来表示当前人工鱼所在位置的食物浓度,即目标函数;人工鱼个体之间的距离表示为  $d_{i,j} = \|X_i - X_j\|$ ;  $Visual$  表示人工鱼的视野感知半径;  $Step$  表示人工鱼在觅食时移动的最大步长;  $\delta$  表示人工鱼聚群和追尾时的拥挤程度;  $try\_number$  为最大尝试次数.

人工鱼群算法模型中包含 3 种主要的算子,即觅食、聚群和追尾,它们决定算法的优化性能和最优解搜索的精度.

### 1.2 觅食

假设在  $t$  时刻第  $i$  条人工鱼的状态为  $X_i$ , 按式(2)在其视野范围内随机选择一个状态为  $X_j$  的人工鱼,并分别计算它们的目标函数值.如果  $Y_i < Y_j$  (以求极大值为例),则按式(3)所示,向  $X_j$  方向移动一步;若不满足前进条件,利用式(2)再次计算,重新选择状态  $X_j$ ,并再次判断是否满足前进条件;以此反复尝试  $try\_number$  次后,如果仍不满足前进条件,则按式(4)随机移动一步,其中  $\text{rand}()$  代表  $0 \sim 1$  之间的随机数.

$$X_j = X_i + \text{rand}() \cdot Visual, \quad (2)$$

$$X_{\text{next}} = X_i + \text{rand}() \cdot Step \cdot \frac{X_j - X_i}{\|X_j - X_i\|}, \quad (3)$$

$$X_{\text{next}} = X_i + \text{rand}() * Step. \quad (4)$$

### 1.3 聚群

假设在  $t$  时刻第  $i$  条人工鱼的状态为  $X_i$ , 探索当前视野范围内(即  $d_{i,j} < Visual$ ) 的伙伴数目  $n_i$  及中心位置  $X_c$ . 如果  $Y_c / n_i > \delta Y_i$ , 则表明伙伴中心有较多的食物并且不太拥挤,此时人工鱼( $X_i$ )向伙伴的中心位置  $X_c$  方向前进一步,形成新的状态  $X_{\text{next}}$ , 如式(5)所示;否则,执行觅食算子.

$$X_{\text{next}} = X_i + \text{rand}() \cdot Step \cdot \frac{X_c - X_i}{\|X_c - X_i\|}. \quad (5)$$

### 1.4 追尾

假设在  $t$  时刻第  $i$  条人工鱼的状态为  $X_i$ , 探索当前视野范围内(即  $d_{i,j} < Visual$ ) 目标函数值最大的伙伴( $X_{\text{max}}$ ), 及在其视野范围内(即  $d_{i,\text{max}} < Visual$ ) 的  $n_{\text{max}}$  条人工鱼. 如果存在  $Y_{\text{max}} / n_{\text{max}} > \delta Y_i$ , 则表明伙伴  $X_{\text{max}}$  的状态较优,具有较高的食物浓度并且周围不太拥挤,此时按式(6)向  $X_{\text{max}}$  方向前进一步;否则,执行觅食算子.

$$X_{\text{next}} = X_i + \text{rand}() \cdot Step \cdot \frac{X_{\text{max}} - X_i}{\|X_{\text{max}} - X_i\|}. \quad (6)$$

### 1.5 公告板

在人工鱼的寻优进程中,设立一个公告板,用来记录每一次迭代后,鱼群中具有最大函数值的人工鱼状态及其位置.鱼群每次执行完一遍觅食算子、聚群算子和追尾算子后,都将具有最优状态的人工鱼

的函数值与公告板上的函数值进行比较,如果该人工鱼的函数值优于公告板上的函数值,则用其函数值取代公告板上的函数值,同时记录该人工鱼的位置.

2 变步长和拥挤因子的自适应人工鱼群算法

2.1 步长的改进

步长的取值能够直接影响算法的优化精度.步长越大,精度越低;步长越小,精度越高<sup>[12]</sup>.在算法前期,本文采用较大的步长,以加快算法的收敛;在算法后期,本文采用较小的步长,以尽可能逼近极值,取得较高的优化精度.为了平衡收敛性和优化精度,本文结合迭代次数,为步长引入一个权值.权值计算如式(7)所示,其中  $t$  为当前迭代次数,  $T$  为总迭代次数.通过控制这个权值,使得在迭代初期,鱼群能够以较大的步长移动,加快收敛;而在迭代后期以较小的步长逼近极值,以取得较高的优化精度.

$$Step = Step \cdot \left(1 - \frac{t}{T}\right).$$

(7)

2.2 拥挤度因子的改进

拥挤度因子  $\delta$  的设置,是为了控制某个极值点附近聚集的鱼群数量.如果设置的  $\delta$  值过大,鱼群容易聚集在某个极值点附近,使得鱼群过分早熟,让优化进程陷入局部最优;如果设置的  $\delta$  值过小,鱼群虽然容易摆脱局部极值,搜索到全局最优解,但需要增多迭代次数,使得搜索全局极值时间变长.

由于每条人工鱼在搜索过程中的状态不同,本文借助小生境<sup>[13]</sup>的思想,以每条人工鱼视野构成的鱼群子群作为一个小生境<sup>[14]</sup>.如果当前人工鱼的状态与此小生境中最优状态的人工鱼的函数值相近,说明在这个子群构成的小生境中,已出现局部极值点;此时应该设置较小的拥挤度因子  $\delta$ ,以加强人工鱼的游动,增加搜索机会,防止鱼群早熟.如果当前人工鱼的状态与此小生境中最优状态的人工鱼的函数值相差较大,说明当前人工鱼离小生境的最优值较远,此时应设置较大的拥挤度因子  $\delta$ ,以加强聚群行为和追尾行为,使该人工鱼向鱼群中心或最优位置移动,加快收敛速度.在每条人工鱼视野构成的小生境中,每条人工鱼将自身的状态  $Y_i$  与小生境中局部极值  $Y_{best}$  进行对比,并采用式(8)的变异策略,更新形成下一代鱼群拥挤度因子  $\delta'$ .拥挤度因子的改进见式(9).

$$m = \frac{|Y_{best} - Y_i|}{Y_{best}},$$

(8)

$$\delta' = \delta * m.$$

(9)

算法经过上述步长和拥挤度因子的改进,变为一种自适应的鱼群算法,算法流程如下:

- 步骤 1 给定初始化鱼群位置及相关参数.
- 步骤 2 计算每条人工鱼的函数值,选取函数值最优的人工鱼个体,并将其相关信息记录在公告板上.
- 步骤 3 分别更新每条人工鱼状态,即执行聚群算子和追尾算子,选取函数值最优的人工鱼与公告板进行比较,如果更优则更新公告板内容.
- 步骤 4 更新每条人工鱼相对应的步长和拥挤度因子,进入下一次迭代.
- 步骤 5 将更新后的人工鱼函数值与公告板相比较,如果更优则更新公告板内容.
- 步骤 6 判断是否达到终止条件,如果满足终止条件,算法结束,输出最优值;否则,转到步骤 3.

改进后的人工鱼群算法流程图如图 1 所示,其中虚框部

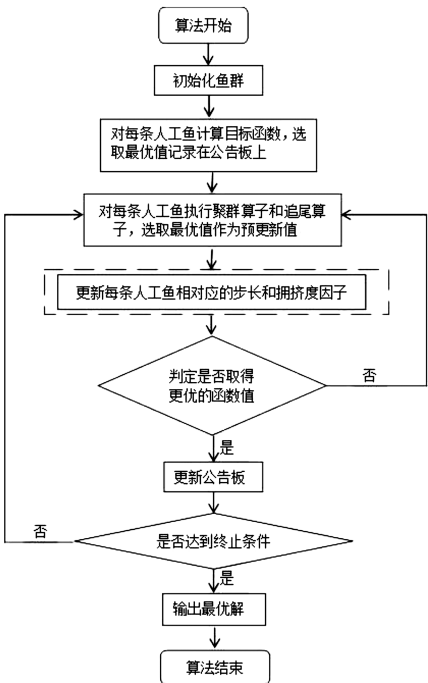


图 1 算法流程图

分为算法的改进部分.

### 3 实验分析

#### 3.1 测试函数

为了验证算法的性能,选取 9 个标准测试函数进行数值实验<sup>[15]</sup>,其中 4 个为求极大值,5 个为求极小值.各函数说明如下:

F1 
$$f(x) = \prod_{i=1}^D \frac{\sin x_i}{x_i}.$$
 (10)

该非线性函数的搜索范围为 $[-10,10]$ ,在 $(0,0)$ 处,有极大值 1.同时,该函数具有许多局部极值,这些局部极值产生的振荡容易使寻优算法陷入局部极值,因此适合于验证算法的寻优性能.

F2 
$$f(x) = 200 - (x_1^2 + x_2 - 11)^2 - (x_1 + x_2^2 - 7)^2.$$
 (11)

该函数是改进的 Himmelbau 函数,搜索范围为 $[-6,6]$ .在函数的定义域范围内有 4 个非均匀分布且比较平坦的相等峰值,其理论最优解为 200.

F3 
$$f(x) = \left( \frac{3.0}{0.05 + x_1^2 + x_2^2} \right)^2 + (x_1^2 + x_2^2)^2.$$
 (12)

该函数的搜索范围为 $[-5.12,5.12]$ ,在 $(0,0)$ 处取得全局最优解 3 600,且在 $(5.12,5.12)$ 、 $(-5.12,-5.12)$ 、 $(5.12,-5.12)$ 、 $(-5.12,5.12)$ 处取得局部最优解,均为 2 748.78.

F4 
$$f(x) = 0.5 - \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}.$$
 (13)

该函数是 Schaffer 函数,搜索范围为 $[-5,5]$ ,在 $(0,0)$ 处可取得全局最大值 1.同时,该函数具有无数个极小值点.由于该函数在最优解附近振荡强烈,很难找到全局最优值,因此适于验证算法的寻优性能.

F5 
$$f(x) = \sum_{i=1}^D x_i^2.$$
 (14)

该函数是单峰的 Sphere 函数,搜索范围为 $[-100,100]$ ,对应的全局最小值为 0.

F6 
$$f_5(x) = \sum_{i=1}^{D-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2).$$
 (15)

该函数是多峰的 Rosenbrock 函数,搜索范围为 $[-30,30]$ ,对应的全局最小值为 0.

F7 
$$f(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10).$$
 (16)

该函数是多峰的 Rastrigin 函数,搜索范围为 $[-5.12,5.12]$ ,对应的全局最小值为 0.

F8 
$$f(x) = \sum_{i=1}^D \left( \sum_{j=1}^i x_j \right)^2.$$
 (17)

该函数是多峰的 Schwefel 1.2 函数,搜索范围为 $[-100,100]$ ,对应的全局最小值为 0.

F9 
$$f(x) = \frac{1}{4\,000} \left( \sum_{i=1}^D x_i^2 \right) - \prod_{i=1}^D \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1.$$
 (18)

这是非线性多模态的 Griewank 函数,搜索范围为 $[-600,600]$ ,对应的全局最小值为 0.

#### 3.2 固定迭代次数下的实验

为验证本文算法的有效性,采用标准人工鱼群算法、文献[4]算法和本文提出的算法,对上述 9 个测试函数进行测试.测试参数设置如下:初始人工鱼群规模  $N=50$ ,初始拥挤度因子  $\delta=0.618$ ,初始步长  $Step=0.3$ ,视野半径  $Visual=2.5$ ,最大尝试次数  $try\_number=5$ ,最大迭代次数为 100.测试函数维度为 2,3 种算法各自运行 50 次,统计结果如表 1 所示.

从表 1 中可以看出,在 3 种算法中函数 F5、F6、F8、F9 均无最优解,但本文算法得到的精度相对较高.这说明,本文算法在固定迭代次数下(函数维度为 2)其收敛精度优于其他两种算法.

表 1 3 种算法在固定迭代次数下的测试结果

函数	算法	最优值	最差值	均值	标准差
F1	标准算法	0.999 999	0.992 106	0.998 387	0.002 0
	文献[4]算法	0.999 999	0.995 528	0.999 195	0.001 2
	本文算法	1	1	1	0
F2	标准算法	199.996	197.249	199.678	0.509 6
	文献[4]算法	199.999	199.138	199.843	0.201 9
	本文算法	200	199.99	200	9.717E-14
F3	标准算法	3 597.60	2 748.78	3 145.01	3.88E+02
	文献[4]算法	3 575.02	2 220.94	2 791.30	2.35E+02
	本文算法	3 600	2 748.78	3 120.75	5.86E+02
F4	标准算法	0.999 822	0.962 776	0.993 702	0.003 8
	文献[4]算法	0.999 187	0.983 383	0.991 024	0.002 3
	本文算法	1	0.990 073	0.993 194	0.004 1
F5	标准算法	6.06E-10	3.65E-07	7.75E-08	1.57E-07
	文献[4]算法	1.46E-10	1.70E-07	1.31E-07	3.55E-08
	本文算法	7.62E-17	1.92E-08	9.60E-09	1.36E-08
F6	标准算法	1.82E-09	2.72E-07	5.57E-08	6.35E-08
	文献[4]算法	8.62E-11	7.33E-08	1.35E-08	1.89E-08
	本文算法	6.72E-19	4.27E-08	2.13E-08	3.09E-08
F7	标准算法	1.00E-07	0.995 0	0.132 7	0.344 0
	文献[4]算法	1.52E-07	0.995 2	0.132 7	0.344 0
	本文算法	0	0.998 3	0.083 9	0.257 5
F8	标准算法	1.02E-01	2.75E-07	3.95E-08	5.50E-08
	文献[4]算法	2.07E-11	6.72E-08	1.45E-08	1.74E-08
	本文算法	1.20E-23	6.74E-08	7.25E-09	1.03E-08
F9	标准算法	0.000 777	0.035 982	0.016 265	0.009 8
	文献[4]算法	0.000 760	0.281 465	0.074 338	0.051 6
	本文算法	0.000 161	0.218 883	0.056 928	0.050 1

3.3 不同维度下 3 种算法的性能验证

选择函数 F1 验证不同维度下 3 种算法的优化性能. 图 2 是维度为 5 的 3 种算法的迭代曲线. 图 3 是维度为 9 的 3 种算法的迭代曲线. 由图 2 和图 3 的曲线可知, 3 种算法的最优值求解精度在高纬度下都大幅下降, 但本文提出的算法仍优于其他 2 种算法. 图 4 是 3 种算法在不同维度(1~10)下的迭代曲线, 图 4 中曲线进一步验证了本文算法的优越性.

4 结束语

本文提出了一种变步长和拥挤度因子的自适应人工鱼群算法. 实验表明, 本文算法在保证精度的前

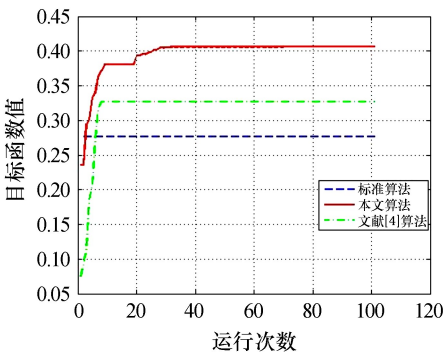


图 2 维度 D=5 的迭代曲线



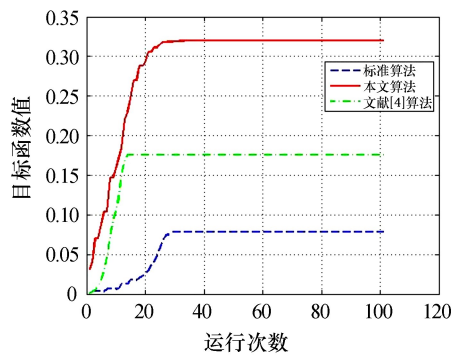


图 3 维度 D=9 的迭代曲线

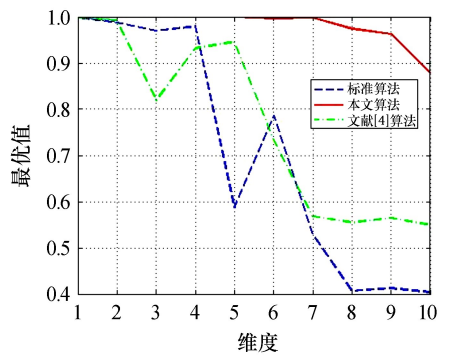


图 4 不同维度(1~10)下的迭代曲线

提下,能提高算法的收敛速度,可有效避免函数陷入局部最优解,且运行效率明显高于传统的人工鱼群算法和文献[4]提出的算法.由此说明本文的算法是可行的,具有一定的实际应用价值.

参考文献:

[1] 李晓磊,邵之江,钱积新,等.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.

[2] 朱旭辉.变步长自适应的改进人工鱼群算法[J].计算机科学,2015,42(2):210-216.

[3] 吴昌友.一种改进的人工鱼群优化算法[J].智能系统学报,2015,10(3):465-469.

[4] 易正俊.自适应重生鱼群优化算法[J].计算机应用与软件,2016,33(6):227-230.

[5] 廖孝勇,李尚键,孙隼华,等.一种基于膜计算的改进人工鱼群算法[J].小型微型计算机系统,2014,35(5):1142-1146.

[6] 李君,梁昔明.人工鱼群算法收敛速度改进优化仿真[J].计算机仿真,2018,35(1):232-238.

[7] 张严,楚晓丽.一种改进的人工鱼群算法[J].计算机系统应用,2011,20(5):199-201.

[8] 刘东林,李乐乐.一种新颖的改进人工鱼群算法[J].计算机科学,2017,44(4):281-286.

[9] 梁昔明,李君.改进步长和视野的人工鱼群算法[J].北京建筑大学学报,2017,33(2):47-53.

[10] 费腾,张立毅.配送中心选址分配问题的 DNA-鱼群算法研究[J].计算机工程与科学,2016(5):938-945.

[11] 任可强,高晓林.基于 AFSA 和 PSO 融合优化的 AdaBoost 人脸检测算法[J].小型微型计算机系统,2016(4):861-865.

[12] 王联国,施秋红.人工鱼群算法的参数分析[J].计算机工程,2010,36(24):169-171.

[13] Thomsen R. Multimodal optimization using crowding-based differential evolution[C]//IEEE Congr Evol Comput. New Jersey: IEEE, 2004:1382-1389.

[14] Li X. Efcient differential evolution using speciation for multimodal function optimization[C]//The 7th Annual Conf on Genetic and Evolutionary Computation. New York: ACM Press, 2005:873-880.

[15] Cheng M Y, Lien L C. Hybrid artificial intelligence-based PBA for benchmark functions and facility layout design optimization[J]. Journal of Computing in Civil Engineering, 2012,26(5):612-624.