

# 改进的双群人工鱼群算法

万林浩

重庆工商大学数学与统计学院, 重庆

收稿日期: 2022年5月13日; 录用日期: 2022年6月1日; 发布日期: 2022年6月15日

## 摘要

针对传统人工鱼群算法在后期收敛速度较慢以及难以跳出局部最优值找到全局最优值的问题, 提出了一种改进的双群人工鱼群算法。该算法第一种群采用位置向量交换行为快速寻优, 第二种群采用混乱行为产生的新的鱼群再次寻优, 最终得到两次寻优的结果得到的交叉解。为了证明双群人工鱼群算法的有效性, 在实验中采用了10种经典的测试函数, 并且与前人提出的改进方法包括规范鱼群算法(NFSA)、基于扩展记忆粒子群优化算法的人工鱼群(PSOEM\_FSA)算法、综合改进人工鱼群(CIAFSA)等算法进行全方面综合对比。实验结果表明, 双群人工鱼群算法较规范鱼群算法、基于扩展记忆粒子群优化算法的人工鱼群算法、综合改进人工鱼群算法在局部寻优以及全局寻优具有更精准更有效率的寻优结果。

## 关键词

人工鱼群算法, 位置向量交换, 自适应视野和步长, 混乱行为, 函数优化

# Improved Double Swarm Artificial Fish Swarming Algorithm

Linhao Wan

College of Mathematics and Statistics, Chongqing Technology and Business University, Chongqing

Received: May 13<sup>th</sup>, 2022; accepted: Jun. 1<sup>st</sup>, 2022; published: Jun. 15<sup>th</sup>, 2022

## Abstract

Aiming at the problem that the traditional artificial fish school algorithm converges slowly in the later stage and it is difficult to jump out of the local optimal value to find the global optimal value, an improved dual-group artificial fish school algorithm is proposed. In this algorithm, the first group uses position vector exchange to quickly find the best, and the second fish group uses the new fish group generated by chaotic behavior to find the best again, and finally obtains the cross solution obtained from the results of the two optimizations. In order to prove the effectiveness of

the two-swarm artificial fish swarm algorithm, 10 classic test functions are used in the experiment, and the improved methods proposed by the predecessors include the normalized fish swarm algorithm (NFSA) and the particle swarm optimization algorithm based on extended memory. Artificial fish school (PSOEM\_FSA) algorithm, comprehensive improved artificial fish school (CIAFSA) and other algorithms are comprehensively compared. The experimental results show that the dual-group artificial fish school algorithm is more accurate and more efficient than the standard fish school algorithm, the artificial fish school algorithm based on the extended memory particle swarm optimization algorithm, and the comprehensive improvement of artificial fish swarm algorithm has more accurate and efficient optimization results in local optimization and global optimization.

## Keywords

Artificial Fish School Algorithm, Position Vector Exchange, Adaptive Field of View and Step Size, Chaotic Behavior, Function Optimization

Copyright © 2022 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

人工鱼群算法由李晓磊等人[1]于2002年提出,是在仿真动物群体智能行为基础上的一种寻找最优解的算法,根据某一片水域中鱼存在数目最多的地方就是该水域营养成分最多的地方这一特征来模拟鱼群觅食行为从而获得最优解。该算法由于人工鱼数目较大,且对初始数值要求不高从而有较大的鲁棒性,在诸多领域得到了广泛的应用,例如冷水机的负荷优化[2]、AGV路径优化[3]、特征选择[4]、图像量化[5]、交通控制[6]、智能车辆轨迹规划等[7]。但人工鱼群算法存在后期收敛速度慢,以及存在容易陷入局部极值所导致寻找最优解时效果一般的问题,因此研究学者对此算法提出了很多改进方法,例如,曲良东[8]根据传统人工鱼群算法容易陷入局部极值这一问题引入了非线性动力学系统中的混沌现象,将人工鱼群中的分量进行混搭搜索从而摆脱局部最优解的束缚提高了效率;陈广洲[9]引入了克隆算法中的消亡操作,使得人工鱼群中的个体得到了改进从而提高了算法后期的收敛效率;刘彦君[10]通过对视野以及步长的改进优化了算法收敛速度。

上文提及的人工鱼群改进的算法相较于传统的人工鱼群算法在收敛速度以及容易陷入局部极值的问题上都有很大的改进。本文参考以上学者的基本思想,通过结合如何跳出局部最优解以及如何快速收敛两大问题提出一种改进的双群人工鱼群算法。

## 2. 基本人工鱼群算法

人工鱼是一个抽象化的实体,其中涵盖了某个实体点的自身数据以及一系列行为活动,可以通过环境中所提供的信息做出相应的活动。而环境是由周围信息以及其他人工鱼所构成。

人工鱼获取环境信息是依靠视野 visual 来实现的。其中变量部分为人工鱼的总数  $N$ 、人工鱼个体的状态  $\mathbf{x} = (x_1, x_2, \dots, x_n)$ , 人工鱼移动的最大步长 Step, 人工鱼的视野 Visual、尝试次数 Try-number、拥挤度因子  $\delta$ 、人工鱼个体  $i, j$  之间的距离  $d_{ij} = |x_j - x_i|$ 。函数部分包括人工鱼当前所在置食物浓度  $y = f(\mathbf{x})$  ( $y$  为目标函数值)以及人工鱼各种行为函数。

人工鱼有四种基本行为:觅食行为,聚群行为,追尾行为以及随机行为。

觅食行为：觅食行为是一种食物吸引鱼向其靠近的活动，鱼通过视觉或味觉来感知环境中的营养浓度，从而向其一步步靠近。设置一条人工鱼的状态为  $x_i$ ，随机感知其视野范围内的另一状态  $x_j$ ，若  $x_j$  位置的适应度大于  $x_i$  位置的适应度则向  $x_j$  方向移动一个 Step。否则， $x_i$  继续在其视野内选择状态  $x_j$ ，判断是否满足前进条件，反复尝试 *Tray-number* 次后，仍没有满足前进条件，则执行随机行为。

聚群行为：部分鱼汇集成群，进行集体觅食或者躲避天敌的行为。人工鱼通过比较当前位置  $x_i$  与视觉范围内鱼群中心的位置  $x_j$  的适应度，若  $x_j$  位置的适应度大于  $x_i$  位置的适应度，且中心位置不是很拥挤 ( $y_i/n_f > \delta y_j$ ) 则向中心位置移动一个 Step，否则进行觅食行为。

追尾行为：当一条鱼或某些鱼发现食物时，附近的几条鱼也会追随之来。一条人工鱼搜索附近处于最优位置的人工鱼 ( $d_{ij} < Visula$ ) 当最优位置人工鱼的适应度大于当前人工鱼所处位置的适应度 ( $y_j > y_i$ )，且最优位置人工鱼附近不是很拥挤 ( $y_i/n_f > \delta y_j$ )，则向最优位置人工鱼方向移动一个 Step。

随机行为：它是觅食行为的一个缺省行为，指人工鱼在视野内随机移动。当发现食物时，会向食物逐渐增多的方向快速地移动。

根据以上寻优过程，基本人工鱼群算法的整体执行过程包括以下六个步骤。

Step1: 随机初始化鱼群。设置搜索空间、迭代次数、视野范围以及鱼群总数等参数。设置目标函数。

Step2: 计算初始鱼群的各个个体的适应值，取最优人工鱼的状态与其值赋给公告牌。

Step3: 若迭代次数达到上限，或局部最优达到函数期望值  $p$ ，则输出优化值，结束算法。

Step4: 对每个个体进行评价，对其要执行的行为进行选择，包括觅食行为、聚群行为、追尾行为。

Step5: 人工鱼执行所选择行为进行更新，产生新的鱼群。

Step6: 评价所有个体，若某个体优于公告牌，则将公告牌更新为该个体，迭代次数加 1 并返回 Step3。

### 3. 双群人工鱼群算法

人工鱼群算法具有对初始值的设置要求不高和鲁棒性较强等优点，但同时也存在着一些需要改进的地方：由于视野和步长随机性和随机行为的存在，导致该算法的寻优精度提高难度较大，它只能快速找到局部最优值，在遇到多峰函数时难以快速高效的找到全局最优值。但在很多实际问题的解决中遇到的目标函数往往不是简单的单峰函数而是多峰函数，这时传统的人工鱼群算法就显得较为乏力。因此本文根据人工鱼群算法的上述问题作出了改进。

相比原始的人工鱼群算法上文提到的改进方法在收敛速度方面都有了很大的改进。但是以上的改进方法仅仅在算法的迭代次数上行进了改进，大多数没有涉及动态参数，且单一的寻优模式不能保证最终寻优结果的正确性。本文在双种群人工鱼群算法[11]的基础上提出了一种改进的双群人工鱼群算法，本算法的改进思路如下。

首先本文参考了基因算法[12]中保留优秀基因的方法，让鱼群同时进行位置向量交换行为和混沌行为，位置向量交换行为可以使算法在迭代过程中保留优秀的位置向量从而达到提高收敛速度的目的，而混沌行为通过打乱人工鱼群的方法使算法达到跳出局部最优解的目的。其次对视野和步长进行改进，使得算法在每个阶段的寻优中都能获得合适的视野和步长从而提高寻优效率。最后再将两种行为得到的寻优值进行交叉得到交叉解。这种改进能有效解决原始人工鱼群算法在多峰函数中无法快速收敛到全局最优解的问题。

#### 3.1. 鱼群 1 位置向量交换

每一条人工鱼都有其特定的位置，而这些位置是由不同的特性所构成的不同的位置向量。本文所指的位置向量交换是指将每条人工鱼同一维度下的位置特性提取出来替换当前最优人工鱼的同一维度位置

特征。选出一个当前维度下是最优人工鱼适应度达到最优的特征，并将这一特征替换到最优人工鱼同一维度的位置向量下，使其跳出局部最优解。这样可以大大增加算法的收敛速度。

$N$  为人工鱼群总数， $n$  为总维度， $x_i^k$  为第  $i$  个个体中的第  $k$  维向量特性， $x_{t,best}^k$  为第  $t$  次迭代后适应度最高的人工鱼中的第  $k$  维向量特性， $x_{t+1,best}^k$  为第  $t$  次迭代后适应度最高的人工鱼第  $t+1$  次迭代后的第  $k$  维向量。  $z_i^k$  为第  $t$  次迭代后适应度最高的人工鱼与第  $i$  个个体第  $k$  维位置特征交换完后的位置， $z_{i-1}^k$  为第  $t$  次迭代后适应度最高的人工鱼与第  $i-1$  个个体第  $k$  维位置特征交换完后的位置。  $x_{t,best}$  为第  $t$  次迭代后公告板中的局部最优值， $x_{t-1,best}$  为第  $t-1$  次迭代后公告板中的全局最优值， $x_{t,i}$  为第  $t$  次迭代后第  $i$  个个体。  $P$  为人工鱼群的平均值， $\beta$  的大小含义为迭代时最小变化的阈值。

位置向量交换公式为

$$x_{i+1}^k = \begin{cases} x_i^k, & f(z_i^k) < f(z_{i-1}^k) \text{ and } \omega 1 > W1, \\ x_{t,best}^k, & \text{otherwise,} \end{cases}$$

$$\alpha = \frac{x_{t,best} - x_{t-1,best}}{x_{t,best} - P},$$

$$P = \frac{\sum_{i=1}^N x_{t,i}}{N},$$

$$\omega 1 = \begin{cases} \omega 1 + 1, & 0 \leq \alpha < \beta, \\ 0, & \text{otherwise.} \end{cases}$$

当  $0 \leq \alpha < \beta$  时判定结果陷入局部最优解，此时通过  $\omega 1$  是否大于  $W1$  来判断是否进行位置向量交换。若  $\omega 1 \leq W1$  则对  $\omega 1$  进行加权，直到  $\omega 1 > W1$  进行位置向量交换。

### 3.2. 鱼群 2 混乱行为

位置向量交换可以通过人工鱼群中个体的多样性，使得有多个个体的位置向量可以替换，从而达到跳出局部最优值的困扰，让收敛速度提高。但当人工鱼群中的个体总数数量太少或缺少多样性时就会降低位置向量交换的寻优效率。此时可以加入一种混乱行为，混乱行为是一种当算法陷入局部最优值时随机初始化人工鱼群中所有个体位置重新寻找局部最优值的以达到摆脱局部最优值的行为。将两次寻找到的局部最优值进行位置向量交换，保存更好的位置向量来找到更好的解。

$u_{t,best}^k$  为混乱行为后局部最优值的第  $k$  维特征， $x_{best}^k$  为混乱行为前局部最优值的第  $k$  维特征， $z_{best}$  为混乱行为前局部最优值与混乱行为后局部最优值交换第  $k$  维特征之后的新位置。 $x_{min}$  和  $x_{max}$  分别为混乱行为前人工鱼群个体的位置向量中的最小值和最大值， $rand(x_{min}, x_{max})$  为  $x_{min}$  到  $x_{max}$  范围内的随机值。

混乱行为公式如下所示。

$$u_{t,best}^k = \begin{cases} x_{t,best}^k, & f(z_{best}) < f(x_{best}), \\ x_{t-1}^k, & \text{otherwise,} \end{cases}$$

$$x_i = \begin{cases} rand(x_{min}, x_{max}), & \omega 2 > W2, \\ x_i, & \text{otherwise,} \end{cases}$$

$$\omega 2 = \begin{cases} \omega 2 + 1, & 0 \leq \alpha < \beta, \\ 0, & \text{otherwise.} \end{cases}$$

### 3.3. 自适应视野和步长

在算法寻优过程中视野和步长的变化决定了在解空间中寻找最优解的效率。在训练前期更大的步长和视野显然更为合适，尽可能大的步长和视野可以尽快的获得解空间中的信息。而在寻优的后期阶段大的步长和视野可能会导致算法跳过最优解从而降低寻优效率，所以需要小的视野和步长来一点点逼近最优解。因此固定的视野和步长会限制算法的寻优效率。本文将通过在步长中加入跳跃权值已达到根据视野范围来改变步长的效果，当  $\alpha = 0$  时，对视野范围进行跳跃式缩小。同时加入一种自适应的视野和步长来对算法进行优化，以达到使视野和步长在寻优的各个阶段都能适应。

$visual_t$  为第  $t$  次迭代后的视野范围，同理  $visual_{t-1}$  为第  $t-1$  次迭代后的视野范围。 $\psi$  为视野跳跃权值， $\xi$  为视野收缩权值。 $step_t$  为第  $t$  次迭代后的步长。 $v$  为控制视野缓慢缩小的参数。

自适应视野和步长迭代公式如下所示。

$$\begin{aligned} visual_t &= \psi visual_{t-1} v / (v + 1), \\ v &= 1 - \xi 10t / 11T, \\ step_t &= 0.5 visual_t. \end{aligned}$$

### 3.4. 交叉

假设人  $x_{i,t}$  在经过位置向量交换后得到的最优值为  $x_{i_1,t}$ ，其位置舒适值为  $y_{i_1,t}$ 。在经过混乱行为后得到的最优值为  $x_{i_2,t}$ ，其位置舒适值为  $y_{i_2,t}$ 。将两种方法得到的解交叉得到最优值  $x_{Next}$ ，其对应的舒适值为  $y_{Next}$ 。

$$x_{i,t+1} = \begin{cases} x_{Next}, y_{Next} & \text{优于 } y_{i_1,t} \text{ 和 } y_{i_2,t}, \\ x_{i_1,t}, & y_{i_1,t} \text{ 优于 } y_{i_2,t}, \\ x_{i_2,t}, & y_{i_2,t} \text{ 优于 } y_{i_1,t}. \end{cases}$$

### 3.5. 算法流程

Step1: 设置初始参数，迭代次数，视野大小，步长，人工鱼群人工鱼数量，拥挤度以及目标函数等。

Step2: 随机初始化鱼群，计算人工鱼群的初始适应度，获取最优的人工鱼的适应度赋值给公告板。

Step3: 判断是否需要结束算法，若算法迭代次数达到最大迭代次数或者最优解达到函数期望则结束算法。

Step4: 对每条人工鱼进行评价。计算其要进行的行为，如觅食，聚群，追尾等行为，并且执行各自行为，更新新的人工鱼群状态以及公告板。

Step5: 根据  $\alpha$  和  $\beta$  的大小，判断当前最优解是否陷入局部最优，是否需要执行位置向量交换行为以及混乱行为。

Step6: 对两种行为进行交叉获得新解 Next。

Step7: 迭代次数加 1，返回 Step3。

## 4. 实验及分析

### 4.1. 测试函数

为了验证本文提出的双群人工鱼群算法的可行性和有效性，本文采用四个单峰函数 Sphere、Quadric、Sumpower、ELLiptic，六个多峰函数 Levy、Griewank、Alipine、Rosenbrock、Rastrigin、Ackley 对该算法 MATLAB 上进行测试，以达到测试该算法在多局部极致和单局部极致的全局寻优能力

以及效率。在不同的函数下测试能够证明该算法迭代鲁棒性。

单峰函数中, Sphere 为球形函数, 其函数图像为二维的下呈弧面, 与 Quadric 函数类似, 都有约束力和维度数量成正比的特点。Sumpower 函数的收敛效率是通过前几个维度的值控制函数的精度和其他维度形成的噪声所影响。Elliptic 函数和 Quadric 函数类似都有其函数精度效率和维度成反比的特点。

多峰函数中, Levy 函数和单峰函数的二维图像为谷型谷底有多个局部峰值。Griewank 函数图像为单峰下降的弧面, 有多个局部极值, 并且都在靠近 0 点的平整去对算法寻优有很大考验。Alipine 函数在解空间有类似正弦函数形态的多个局部峰值。Rosenbrock 函数的解空间有 0 和 1 两个局部极值, 并且在 1 处极值向 0 处极值过渡时对算法的寻优能力有较大考验。Rastrigin 函数的解空间中有多个局部极值。Ackley 函数有着远离 0 点处有多个局部极值, 0 点附近成急剧下降的特点。

### 4.2. 实验参数设置

双群人工鱼群算法与传统人工鱼群算法有一个同样的特点就是参数对其影响不是很显著, 对其影响人工鱼群寻优效率影响同样不大, 所以其参数设置只要在合理范围内, 算法就能有效运行。

设定视野大小时, 应使其尽量包含全部解空间。在不同维度下, 解空间的大小与其维度有关, 其值为  $\sqrt{n}(U-L)$ , 其中  $U = x_{\max}$ ,  $L = x_{\min}$ 。但在高维度情况下, 视野范围刚好为  $(U-L)$  时, 并不能使其包含全部解空间, 所以设置视野大小时应使其在一定范围内扩大。本实验将视野范围定为  $1.5(U-L)$ , 步长的大小为  $E = 0.5visual$ 。位置向量交换和混乱行为的判断参数分别为  $M1 = n/2$  和  $M2 = 5n$ 。维度为  $n = 10$ , 最大迭代数  $T = 100$ , 拥挤度因子  $\delta = 0.75$ 。

实验中双群人工鱼群算法参数如表 1 所示。

**Table 1.** Initial parameters of the two-group artificial fish swarming algorithm

**表 1.** 双群人工鱼群算法初始参数

参数	值	参数	值
$\beta$	0.05	Try_number	5
$\omega$	1.1	$\delta$	0.75
$\xi$	100	N	50
$\psi$	0.6	W1	$n/2$
Visual	$U-L$	W2	$5n$
Step	$0.5visual$		

### 4.3. 实验结果及分析

本次实验与近些年来对传统人工鱼群算法进行的改进的算法进行了比较。其中包括规范鱼群算法(NFSA) [13]、基于扩展记忆粒子群优化算法的人工鱼群算法(PSOEM\_FSA) [14]、综合改进人工鱼群算法(CIAFSA) [15]等进行的一定程度的比较。

**Table 2.** Experimental comparison table

**表 2.** 实验对比表

函数名	对比指标	双群人工鱼群算法	NFSA	PSOEM-FSA	CIAFSA
ALipine	Best	$1.52 \times 10^{-64}$	$4.11 \times 10^{-43}$	$4.79 \times 10^{-7}$	$1.94 \times 10^{-6}$
	Mean	$1.93 \times 10^{-59}$	$3.56 \times 10^{26}$	$6.87 \times 10^{-6}$	$8.98 \times 10^{-6}$
	SD	$3.55 \times 10^{-59}$	$9.75 \times 10^{-8}$	$5.39 \times 10^{-6}$	$1.18 \times 10^{-5}$

Continued

Sumpower	Best	$1.33 \times 10^{-41}$	<b><math>1.00 \times 10^{-197}</math></b>	$1.43 \times 10^{-15}$	$4.57 \times 10^{-13}$
	Mean	$1.25 \times 10^{-38}$	<b><math>7.10 \times 10^{-197}</math></b>	$9.70 \times 10^{-10}$	$1.72 \times 10^{-9}$
	SD	$2.44 \times 10^{-35}$	<b>0</b>	$1.69 \times 10^{-9}$	$3.47 \times 10^{-9}$
Elliptic	Best	<b>0</b>	$2.95 \times 10^{-48}$	$1.03 \times 10^{-2}$	$2.18 \times 10^{-3}$
	Mean	<b><math>7.00 \times 10^{-56}</math></b>	$1.09 \times 10^{-46}$	5.73	$8.40 \times 10^{-3}$
	SD	<b><math>1.01 \times 10^{-55}</math></b>	$1.58 \times 10^{-46}$	10.4	$3.51 \times 10^{-3}$
Levy	Best	<b>0</b>	$1.50 \times 10^{-32}$	$4.79 \times 10^{-7}$	$1.94 \times 10^{-6}$
	Mean	<b>0</b>	$1.50 \times 10^{-32}$	$6.87 \times 10^{-6}$	$8.98 \times 10^{-6}$
	SD	<b>0</b>	<b>0</b>	$5.39 \times 10^{-6}$	$1.18 \times 10^{-5}$
Rosenbroc	Best	$8.10 \times 10^{-16}$	<b>0</b>	$1.99 \times 10^{-7}$	$2.54 \times 10^{-14}$
	Mean	$2.36 \times 10^{-12}$	<b><math>1.94 \times 10^{-30}</math></b>	$6.55 \times 10^{-10}$	$1.45 \times 10^{-4}$
	SD	$3.19 \times 10^{-12}$	<b><math>6.14 \times 10^{-30}</math></b>	$1.87 \times 10^{-10}$	$3.81 \times 10^{-5}$
Ackley	Best	<b><math>4.44 \times 10^{-16}</math></b>	$8.88 \times 10^{-16}$	$1.68 \times 10^{-10}$	$1.79 \times 10^{-6}$
	Mean	<b><math>4.44 \times 10^{-16}</math></b>	$3.73 \times 10^{-15}$	$1.02 \times 10^{-4}$	$3.56 \times 10^{-4}$
	SD	<b>0</b>	$1.50 \times 10^{-15}$	$1.11 \times 10^{-4}$	$3.95 \times 10^{-4}$
Sphere	Best	<b><math>2.93 \times 10^{-155}</math></b>	$1.14 \times 10^{-55}$	$3.08 \times 10^{-13}$	$1.00 \times 10^{-14}$
	Mean	<b><math>1.81 \times 10^{-147}</math></b>	$1.14 \times 10^{-52}$	$6.19 \times 10^{-8}$	$6.51 \times 10^{-3}$
	SD	<b><math>3.54 \times 10^{-147}</math></b>	$1.64 \times 10^{-52}$	$1.11 \times 10^{-7}$	$2.59 \times 10^{-3}$
Griewank	Best	<b>0</b>	$9.86 \times 10^{-3}$	$2.93 \times 10^{-8}$	$7.47 \times 10^{-6}$
	Mean	<b>0</b>	$3.49 \times 10^{-2}$	$1.07 \times 10^{-1}$	$1.52 \times 10^{-2}$
	SD	<b>0</b>	$1.67 \times 10^{-2}$	$1.84 \times 10^{-1}$	$3.47 \times 10^{-1}$
Rastrigin	Best	<b>0</b>	<b>0</b>	$3.09 \times 10^{-10}$	$4.56 \times 10^{-9}$
	Mean	<b>0</b>	<b>0</b>	$8.34 \times 10^{-6}$	$2.06 \times 10^{-6}$
	SD	<b>0</b>	<b>0</b>	$2.40 \times 10^{-5}$	$4.55 \times 10^{-6}$
Quadric	Best	<b><math>1.35 \times 10^{-69}</math></b>	$2.10 \times 10^{-14}$	$1.16 \times 10^{-8}$	$9.04 \times 10^{-6}$
	Mean	<b><math>5.05 \times 10^{-63}</math></b>	$4.74 \times 10^{-9}$	$1.16 \times 10^{-4}$	9.34
	SD	<b><math>1.02 \times 10^{-62}</math></b>	$9.13 \times 10^{-9}$	$2.63 \times 10^{-5}$	$2.11 \times 10^1$

由表 2 数据得出下列结论:

1) 在单峰测试函数的测试下, Quadric、Sumpower、Elliptic 中的寻优效率和精度要远远高于其他三种改进的人工鱼群算法, 只有 Sumpower 函数的寻优精度上要低于规范鱼群算法。证明了自适应视野和步长的有效性, 提高了算法的精度。

2) 在多峰值测试函数的测试下, 本算法都能够找到全局最优解。且在 Griewank 函数, Rastrigin 函数, 以及 Levy 函数中的精度能够达到函数的全局最优值 0 值处, 该算法除了在 Rosenbrock 函数的求解精度上要劣于规范鱼群算法, 在其他四个测试函数上都要优于其他算法。充分证明了该算法具有跳出局部最优解的能力, 能够提升效率。

3) 在分割函数的寻优表现上, 该算法的求解精度的效率要高于其他几种算法。但在 Rosenbrock 函数和 Sumpower 函数的寻优表现上要劣于规范鱼群算法, 本文认为是因为规范鱼群算法中的规范行为避免了维度与维度之间相互影响例如控制关系和噪音。这种影响在非分割函数中尤为明显, 而双群人工鱼群

算法无法消除这种影响, 所以造成了在上述两个测试函数中的寻优表现不如规范鱼群算法。

由上述实验结果可知, 改进的双群人工鱼群算法在大多数函数上的寻优效果上都要明显优于原始的人工鱼群算法, 并且这一优势在双峰函数的表现上更是优于原始人工鱼群算法, 有效解决了原始人工鱼群算法在面对多峰函数时难以快速收敛到全局最优解的问题。

## 5. 结语

本文在基本的人工鱼群算法的基础上, 根据该算法收敛速度慢以及不易跳出局部最优值的问题提出了一种双群人工鱼群算法, 并通过经典测试函数与其他改进方法, 例如规范鱼群算法、基于扩展记忆粒子群优化算法的人工鱼群算法、综合改进人工鱼群算法进行了比较。实验结果表明改进的双群人工鱼群算法在跳出局部最优值, 提高收敛速度以及精度上都有较好的表现。有理由认为双群人工鱼群算法在解决组合优化问题时有明显的寻优效果。

## 参考文献

- [1] 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.
- [2] Zheng, Z.X., Li, J.Q. and Duan, P.Y. (2018) Optimal Chiller Loading by Improved Artificial Fish Swarm Algorithm for Energy Saving. *Mathematics and Computers in Simulation*, **155**, 227-243. <https://doi.org/10.1016/j.matcom.2018.04.013>
- [3] Li, G., Liu, Q., Yang, Y., Zhao, F., Zhou, Y. and Guo, C. (2017) An Improved Differential Evolution Based Artificial Fish Swarm Algorithm and Its Application to AGV Path Planning Problems. *Proceedings of 2017 36th Chinese Control Conference*, Dalian, 26-28 July 2017, 2556-2561. <https://doi.org/10.23919/ChiCC.2017.8027746>
- [4] Chen, Y.M., Zhu, Q.X. and Xu, H.R. (2015) Finding Rough Set Reduce with Fish Swarm Algorithm. *Knowledge-Based Systems*, **81**, 22-29. <https://doi.org/10.1016/j.knosys.2015.02.002>
- [5] El-Said, A.S. (2015) Image Quantization Using Improved Artificial Fish Swarm Algorithm. *Soft Computing*, **19**, 2667-2679. <https://doi.org/10.1007/s00500-014-1436-0>
- [6] Ma, C. and He, R. (2019) Green Wave Traffic Control System Optimization Based on Adaptive Genetic-Artificial Fish Swarm Algorithm. *Neural Computing and Applications*, **31**, 2073-2083. <https://doi.org/10.1007/s00521-015-1931-y>
- [7] 袁娜, 史昕, 赵祥模. 基于改进人工鱼群算法的车辆轨迹规划方法[J]. 计算机应用, 2018, 38(10): 3030-3035+3047.
- [8] 曲良东, 何登旭. 一种混沌人工鱼群优化算法[J]. 计算机工程与应用, 2010, 46(22): 40-42.
- [9] 陈广洲, 汪家权, 李传军, 鲁祥友. 一种改进的人工鱼群算法及其应用[J]. 系统工程, 2009, 27(12): 105-110.
- [10] 刘彦君, 江铭炎. 自适应视野和步长的改进人工鱼群算法[J]. 计算机工程与应用, 2009, 45(25): 35-37+47.
- [11] 王付宇, 汤涛. 双种群鱼群算法在分布式投资组合的应用[J]. 系统仿真学报, 2021, 33(9): 2074-2084.
- [12] 席裕庚, 柴天佑, 恽为民. 遗传算法综述[J]. 控制理论与应用, 1996(6): 697-708.
- [13] Tan, W.H. and Mohamad-Saleh, J. (2020) Normative Fish Swarm Algorithm (NFSA) for Optimization. *Soft Computing*, **24**, 2083-2099. <https://doi.org/10.1007/s00500-019-04040-0>
- [14] Duan, Q.C., Mao, M.X., Duan, P., et al. (2016) An Improved Artificial Fish Swarm Algorithm Optimized by Particle Swarm Optimization Algorithm with Extended Memory. *Soft Computing*, **45**, 210-222. <https://doi.org/10.1108/K-09-2014-0198>
- [15] Mao, M.X., Duan, Q.C., Duan, P., et al. (2017) Comprehensive Improvement of Artificial Fish Swarm Algorithm for Global MPPT in PV System under Partial Shading Conditions. *Transactions of the Institute of Measurement and Control*, **40**, 2178-2199. <https://doi.org/10.1177/0142331217697374>