

doi: 10.11835/j.issn.1000-582X.08.012

基于 Iterative 映射和非线性拟合的鲸鱼优化算法

李赛宇, 鞠传香, 丁航奇

(山东理工大学 计算机科学与计算学院, 山东 淄博 255000)

摘要:为解决鲸鱼优化算法中收敛速度慢和寻优精度低等问题,提出一种基于 Iterative 映射和非线性拟合的鲸鱼优化算法(NWOA)。首先,该算法利用了 Iterative 映射对鲸鱼种群初始化,保证初始种群的多样性;其次,采用非线性拟合的策略对收敛因子和惯性权重进行改进,以平衡算法的全局勘测能力和局部开发能力。通过对 13 种函数进行仿真实验,从均方差和平均值的角度分析,改进后算法寻优精度显著提高,且稳定性较强。实验结果表明 NWOA 与传统的鲸鱼优化算法相比,收敛速度明显加快。

关键词:鲸鱼优化算法; Iterative 映射; 非线性拟合; 函数优化

中图分类号: TP301.6

文献标志码: A

文章编号: 1000-582X(2023)08-120-12

Whale optimization algorithm based on iterative mapping and nonlinear fitting

LI Saiyu, JU Chuanxiang, DING Hangqi

(School of Computer Science and Technology, Shandong University of Technology, Zibo, Shandong 255000, P. R. China)

Abstract: In order to solve the problems of slow convergence speed and low optimization accuracy in whale optimization algorithm, a whale optimization algorithm based on iterative mapping and nonlinear fitting(NWOA) is proposed. Firstly, iterative mapping is taken advantage to initialize whale population, which guarantees initial population diversity. Secondly, nonlinear fitting strategy is used to improve the convergence factor and inertia weight to balance the global survey ability and local development ability of the algorithm. Through the simulation test of 13 functions, the improved algorithm has a significant improvement in precision and stability from the point of mean square error and average value. The experimental results show that the convergence speed of the algorithm is faster than that of the traditional whale optimization algorithm.

Keywords: whale optimization algorithm; iterative mapping; nonlinear fitting; function optimization

鲸鱼优化算法(whale optimization algorithm, WOA)是由格里菲斯大学 Mirjalili 教授^[1]等于 2016 年提出的一种在海洋中模拟鲸鱼捕杀猎物行为的新群智能优化算法。WOA 具有参数少,优化原理简单,操作易实

收稿日期: 2020-09-22 网络出版日期: 2021-08-13

基金项目: 大学生创新创业训练计划项目; 国家重点研发资助项目(2018YFB1402500)。

Supported by Innovation Entrepreneurship Training Program for College Students and National Key R & D Project (2018YFB1402500).

作者简介: 鞠传香(1978—), 女, 主要从事计算机算法方向研究, (E-mail)chuanxiangju@sina.com。

现的优点。与果蝇觅食优化算法(FOA)^[2]、粒子群算法(PSO)^[3]和灰狼优化算法(GWO)^[4]相比,WOA算法在寻优精度和收敛速度上更具有优势。然而,该算法也存在后期收敛速度慢,容易陷入局部最优解的缺陷。目前,一些学者对鲸鱼优化算法进行了改进,Trivedi等^[5]在基本的WOA算法中引入了自适应策略,实验结果表明改进的算法在收敛速度和寻优精度上优于原算法。Mafarja等^[6]将模拟退火算法的思想引入到基本WOA算法上,提出以WOA-SA算法,并在特征子集的选择方面得到应用。钟明辉等人^[7]提出一种随机调整控制参数的改进鲸鱼优化算法(EWOA),用来处理复杂优化问题。凌颖等^[8]为了平衡算法的全局搜索与局部开发能力,提出了一种引入Levy飞行轨迹策略的LWOA。郭振洲^[9]使用了柯西逆累积分布函数对种群进行变异,并采用自适应权重方法提高算法的收敛精度。王坚浩等人^[10]采用混沌反向学习策略来初始化种群,提出基于混沌搜索策略的鲸鱼优化算法(CWOA)。除此之外,鲸鱼优化算法在飞行航迹缺失数据预测^[11],无人机航线规划^[12],车间调度^[13],图像分割^[14]等领域有广泛应用。为解决鲸鱼优化算法中收敛速度慢和寻优精度低等问题提出基于Iterative映射和非线性拟合的鲸鱼优化算法(NWOA)。

1 鲸鱼优化算法

鲸鱼优化算法是受气泡捕食法^[15]的启发,主要分成3个阶段:收缩包围捕食,气泡攻击捕食,随机搜寻猎物。在基本的WOA中,定义鲸鱼种群数量为 N ,空间维度为 D ,每一个鲸鱼所对应的位置为 $L_i = (L_i^1, L_i^2, L_i^3, \dots, L_i^D)$, $i = 1, 2, \dots, N$,其中猎物所在的位置就是最优鲸鱼的位置。

1.1 收缩包围捕食

在鲸鱼搜寻猎物位置时,发觉猎物位置可能在当前距离猎物最近的鲸鱼位置附近,其他鲸鱼将会朝向前距离猎物最近的鲸鱼位置靠近,达到更新自身位置的目的。换句话说,鲸鱼并不确定猎物的实际位置,只是朝当前最优鲸鱼的方向移动,从而包围猎物。在基本的WOA中,首先需要知道个体和当前最优鲸鱼之间的距离

$$D = |C \cdot L^*(t) - L(t)|, \quad (1)$$

$$C = 2 \cdot r, \quad (2)$$

式中: t 表示当前迭代次数; $L^*(t)$ 表示当前最优鲸鱼位置; $L(t)$ 表示鲸鱼个体位置; C 表示摆动因子。鲸鱼的位置更新公式为

$$L(t+1) = L^*(t) - A \cdot D, \quad (3)$$

$$A = 2 \cdot a \cdot r - a, \quad (4)$$

式中: A 表示收敛因子; a 随着迭代次数增加从2线性递减到0。在式(2)和式(4)中的 r 为 $[0, 1]$ 的随机数。

1.2 气泡攻击捕食

鲸鱼潜入海中,以螺旋姿态向上游,并吐出大小不一的气泡。气泡上升到海平面,构成一张网,把猎物包裹起来。首先计算出个体与当前最优鲸鱼位置之间的距离,其次模拟鲸鱼沿着螺旋状轨迹向猎物位置游去,该过程的数学模型如下

$$D = |L^*(t) - L(t)|, \quad (5)$$

$$L(t+1) = L^*(t) - D \cdot e^{bl} \cdot \cos(2\pi l), \quad (6)$$

其中: b 为螺旋状常数; l 为 $(-1, 1)$ 之间的随机数。

为了实现收缩包围捕食和气泡攻击捕食2种机制同步更新,引用了参数 p 来控制鲸鱼位置的更新方式,公式如下所示

$$L(t+1) = \begin{cases} L^*(t) - D \cdot e^{bl} \cdot \cos(2\pi l), & \text{if } p < 0.5 \\ L^*(t) - A \cdot D, & \text{if } p \geq 0.5 \end{cases}, \quad (7)$$

式中, $p \in [0, 1]$ 。

1.3 随机搜寻猎物

为了增强种群的多样性,鲸鱼也会根据身边个体来改变自身位置,而不是根据猎物的位置。在鲸鱼优化算法中,通过控制收敛因子的值确定搜索范围。当 $|A| > 1$ 时,鲸鱼执行随机搜寻猎物策略,进行全局勘测,避免种群陷入局部最优。其公式如下

$$D = |C \cdot L_{\text{rand}}(t) - L(t)|, \quad (8)$$

$$L(t+1) = L_{\text{rand}}(t) - A \cdot D, \quad (9)$$

式中, $L_{\text{rand}}(t)$ 是当前随机一个鲸鱼的位置。

2 改进的鲸鱼优化算法

2.1 Iterative映射初始化

WOA算法采用随机生成的方式来初始化种群,这种方式可能会破坏种群的多样性。研究表明混沌序列具有很好的随机性和遍历性,文献[16]中提出混沌序列的伪随机性可使其在搜索空间具有更好均匀性,尽可能均匀分布在空间。采用Iterative映射^[17]的方法对种群进行初始化,其方程式如下

$$x_{i+1} = \sin\left(\frac{b\pi}{x_i}\right), \quad (10)$$

其中, $b \in [0,1]$ 。

2.2 非线性拟合的收敛因子和惯性权重

在鲸鱼优化算法中,参数 a 随着迭代次数增加从2线性递减到0。由式(4)可知,参数 a 决定了收敛因子 A 的变化,且收敛因子 A 协调算法全局勘测和局部开发的能力。换句话说,WOA算法的全局勘测和局部开发的能力很大程度上依赖于参数 a 的值。在迭代初期,算法以全局勘测为主, a 的值较大且收敛速度较快;在迭代末期,算法以局部开发为主, a 的值较小且收敛速度较慢。基于上述分析,采用非线性拟合的策略调控收敛因子 A ,表达式为

$$a(t) = -u \cdot \log(t/\text{Max_iter}), \quad (11)$$

其中: t 为当前迭代次数; Max_iter 为最大迭代次数; u 是一个常数。

惯性权重是在粒子群算法中的一个重要参数。文献[18]提出惯性权重的大小可以影响算法的全局勘测能力和局部开发能力,并加快收敛速度和提高寻优精度。当惯性权重较大时,算法偏向全局勘测;当惯性权重较小时,算法偏向局部开发。在基本的鲸鱼优化算法中,引用惯性权重来控制个体位置更新当前最优鲸鱼对其影响力。借鉴文献[19]中的一种拟合下降非线性惯性权重策略,通过一系列模拟实验,提出基于非线性拟合的惯性权重,数学模型如下

$$w(t) = -(w_{\text{max}} - w_{\text{min}}) \cdot t^\eta + w_{\text{min}}, \quad (12)$$

其中: w_{max} 和 w_{min} 分别为惯性权重的最大值和最小值; η 是一个常数参数。将式(11)分别带入公式(3)和(6),得到如下更新位置的公式

$$L(t+1) = w \cdot L^*(t) - A \cdot D, \quad (13)$$

$$L(t+1) = w \cdot L^*(t) - D' \cdot e^{bl} \cdot \cos(2\pi l). \quad (14)$$

2.3 NWOA算法流程图

在基本的WOA算法上进行改进,流程图如图1所示。

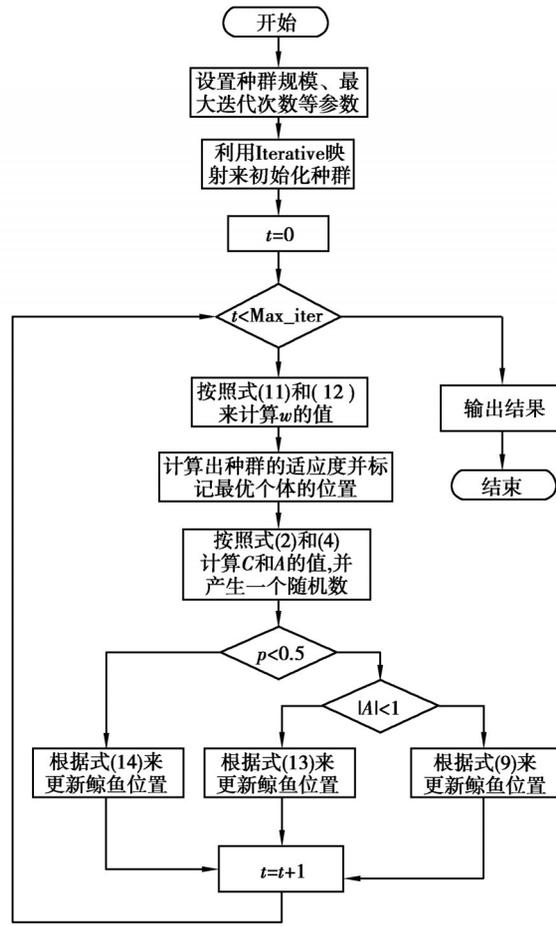


图 1 NWOA 算法流程图

Fig. 1 NWOA algorithm flowchart

3 仿真实验与结果分析

3.1 仿真平台

本次实验的运行环境为:操作系统:Windows 10(64 bit);内存:8GB;编程环境:Matlab R2020(a)。

3.2 测试函数

为了证明基于 Iterative 映射和非线性策略鲸鱼优化算法的可行性,选取了 13 个基准测试函数如表 1 所示。函数大体上可以分为两类: F_1 - F_7 表示单峰测试函数; F_8 - F_{13} 表示多峰测试函数。

表 1 参数的详细值

Table 1 Detailed values of parameters

编号	函数	维度	范围	理论最优值
F_1	$f_1(x) = \sum_{i=1}^n x_i^2$	30	$x_i \in [-100, 100]$	0
F_2	$f_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	$x_i \in [-10, 10]$	0
F_3	$f_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	30	$x_i \in [-100, 100]$	0
F_4	$f_4(x) = \max\{ x_i , 1 \leq i \leq D\}$	30	$x_i \in [-100, 100]$	0
F_5	$f_5(x) = (\sum_{i=1}^n x_i^2)^2$	30	$x_i \in [-100, 100]$	0
F_6	$f_6(x) = \sum_{i=1}^n (x_i + 0.5)^2$	30	$x_i \in [-100, 100]$	0
F_7	$f_7(x) = \sum_{i=1}^n x_i^4 + \text{random}(0,1)$	30	$x_i \in [-1.28, 1.28]$	0

续表1

编号	函数	维度	范围	理论最优值
F_8	$f_8(x) = -\sum_{i=1}^n x_i \sin \sqrt{ x_i }$	30	$x_i \in [-500, 500]$	-12 569.5
F_9	$f_9(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	$x_i \in [-5.12, 5.12]$	0
F_{10}	$f_{10}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right)\right) + 20 + e$	30	$x_i \in [-600, 600]$	0
F_{11}	$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^n (x_i^2) - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	$x_i \in [-5, 10]$	0
F_{12}	$f_{12}(x) = \frac{\pi}{30} \left\{ 10 \sin^2(\pi y_i) + \sum_{i=1}^{n-1} (y_i - 1)^2 \right.$ $\left. [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2 \right\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{(x_i + 1)}{4}$	30	$x_i \in [-32, 32]$	0
	$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \leq x_i \leq a \\ k(-x_i - a)^m & x_i < -a \end{cases}$			
	$\lambda(x) = \sum_{i=1}^{n-1} \sin(3\pi x_i)$			
	$\tau(x) = \sum_{i=1}^{n-1} (x_i - 1)^2 [1 + \sin^2(2\pi x_{i+1})]$			
	$\zeta(x) = \sum_{i=1}^{n-1} (x_i - 1)^2 [1 + \sin(3\pi x_{i+1})]$			
F_{13}	$f_{13}(x) = 0.1 [\lambda(x) + \zeta(x) + \tau(x)] + \sum_{i=1}^n u(x_i, 5, 100, 4)$	30	$x_i \in [-50, 50]$	0
	$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i < a \\ 0 & -a \leq x_i \leq a \\ k(-x_i - a)^m & x_i \geq a \end{cases}$			

3.3 参数设置

在本实验中,把提出的NWOA算法与WOA、EWOA、CWOA、IWOA算法^[20]进行对比,来验证NWOA算法的性能。其中:种群规模为30;最大迭代次数Max_iter为500。所有算法的参数如表2所示。

表2 参数的详细值

Table 2 Detailed values of parameters

算法	参数	值
WOA	a	从2到0线性下降
	a_{initia}	2.00
IWOA	a_{final}	1.00
	u	2.00
EWOA	a_{max}	2.00
	a_{min}	1.00
CWOA		2.00
	a_{initia}	0.00
	$a_{\text{final}} w_{\text{initia}}$	0.90
	w_{final}	0.20
NWOA	b	0.80
	u	0.25
	η	0.10
	w_{max}	1.20
	w_{min}	0.80

3.4 结果和实验分析

对上述每个算法在全部测试函数中都独立运行 50 次,列出最优解,最差解,平均值和均方差来评价算法的优劣。单峰测试函数有一个显著特点是只有一个全局最优解,并不存在局部最优解。可以用这种函数来测试算法收敛性。其实验结果如表 3,从实验结果知,NWOA 算法在 F_1 - F_5 这 5 个测试函数上无论是最优解、最差解、平均值还是均方差都达到了函数的理论最优值, F_6 - F_7 在平均值也是优于其他 4 种算法。均方差表示算法与平均值的偏离程度,均方差越小,说明算法的稳定性越高。观察表 3 可知,NWOA 算法在均方差上均小于或等于上述其他算法。这表明改进的鲸鱼优化算法(NWOA)在收敛精度和稳定性上优于其他算法。

表 3 单峰函数实验结果

Table 3 Experimental results of unimodal function

函数	结果	WOA	EWOA	IWOA	CWOA	NWOA
F_1	最优解	3.319 9e-86	2.808 5e-93	0	0	0
	最差解	2.870 9e-69	8.620 3e-77	0	0	0
	平均值	5.797 4e-71	3.611 3e-78	0	0	0
	均方差	4.059 4e-70	1.554 9e-77	0	0	0
F_2	最优解	1.844 9e-58	2.082 1e-57	0	1.985 2e-230	0
	最差解	4.822 4e-48	2.317 7e-44	0	1.362 1e-201	0
	平均值	9.960 2e-50	4.650 6e-46	0	2.725 6e-203	0
	均方差	6.816 2e-49	3.277 6e-45	0	0	0
F_3	最优解	2.241 8e+03	3.069 2e+00	0	0	0
	最差解	6.217 9e+04	4.098 0e+04	0	0	0
	平均值	4.195 6e+04	1.819 0e+04	0	0	0
	均方差	1.173 1e+04	1.361 3e+04	0	0	0
F_4	最优解	1.741 0	13.190 5	9.051 4e-221	3.729 0e-228	0
	最差解	88.299 6	99.142 5	1.810 3e-222	6.276 2e-190	0
	平均值	56.519 2	83.491 5	1.472 4e-222	1.261 9e-191	0
	均方差	27.446 5	18.749 4	0	0	0
F_5	最优解	2.792 9e-174	1.516 3e-177	0	0	0
	最差解	1.153 2e-143	1.290 8e-148	0	0	0
	平均值	2.379 9e-145	2.582 6e-150	0	0	0
	均方差	1.633 2e-144	1.825 5e-149	0	0	0
F_6	最优解	0.087 3	0.318 5	0.101 9	0.163 4	0.101 0
	最差解	1.482 1	2.812 7	1.625 2	1.034 3	1.070 7
	平均值	1.434 6	1.193 4	0.702 1	0.562 8	0.539 6
	均方差	0.275 3	1.358 4	0.398 4	0.590 0	0.224 1
F_7	最优解	3.585 1e-05	4.606 3e-04	6.454 1e-07	1.843 9e-07	8.812 0e-07
	最差解	0.017 1	0.087 9	4.259 3e-04	6.883 1e-04	2.145 0e-04
	平均值	0.004 1	0.013 0	8.340 9e-05	7.007 7e-05	6.387 0e-05
	均方差	0.003 8	0.014 6	8.781 9e-05	7.433 2e-05	5.422 1e-05

F_1 - F_7 函数值收敛曲线如图2~8所示。通过观察曲线变化趋势,发现NWOA算法的收敛速度优于其他算法。对于 F_3 和 F_4 ,EWOA、WOA算法在原点左右附近徘徊,陷入局部最优解。这表明改进的NWOA算法收敛速度较快。

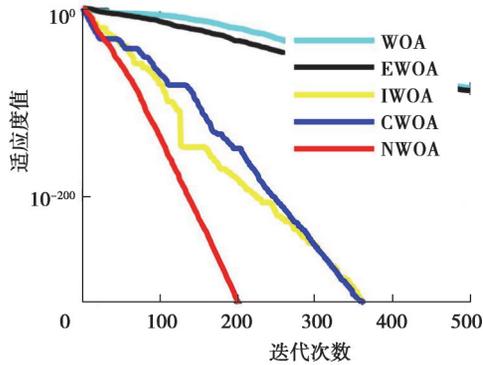


图2 F_1 收敛曲线图

Fig. 2 F_1 convergence curve chart

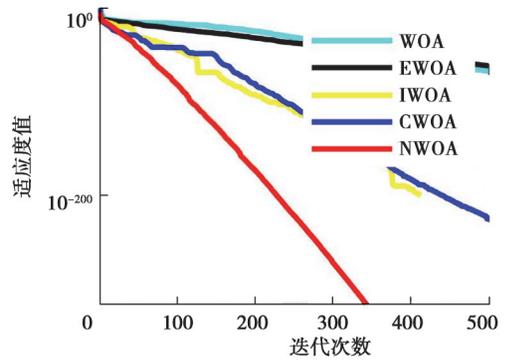


图3 F_2 收敛曲线图

Fig. 3 F_2 convergence curve chart

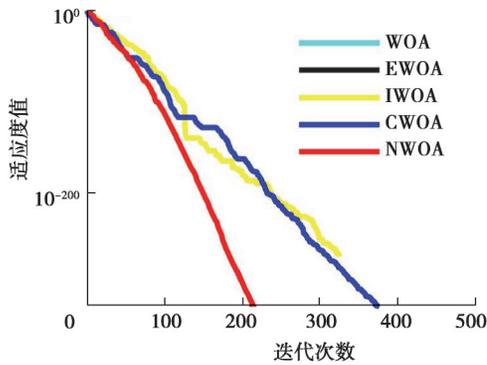


图4 F_3 收敛曲线图

Fig. 4 F_3 convergence curve chart

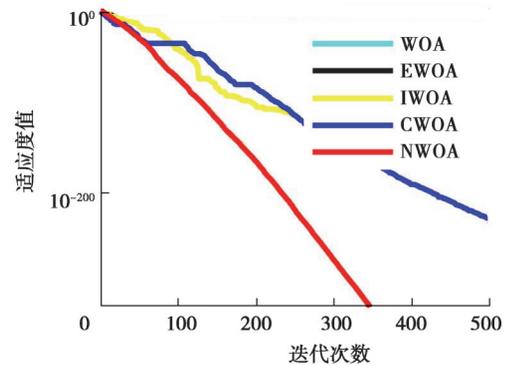


图5 F_4 收敛曲线图

Fig. 5 F_4 convergence curve chart

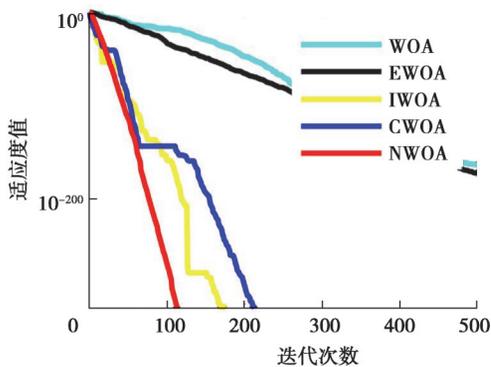


图6 F_5 收敛曲线图

Fig. 6 F_5 convergence curve chart

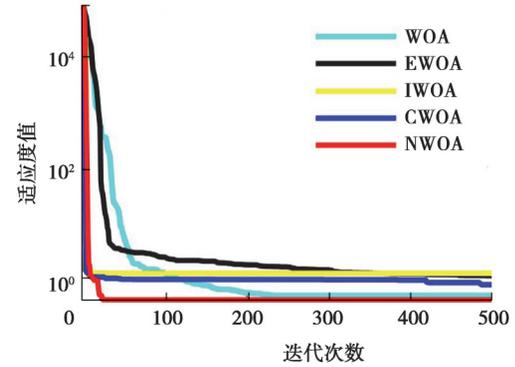


图7 F_6 收敛曲线图

Fig. 7 F_6 convergence curve chart

多维测试函数的实验结果如表4所示。多维函数可验证算法的全局勘测能力。在表4中可以看出,对于 F_9 、 F_{11} 测试函数优化中,NWOA算法达到理论最优值,稳定性较高;对于 F_{10} 、 F_{12} 、 F_{13} 这三个函数来说,NWOA算法在收敛精度和稳定性上优于其他算法;就测试函数 F_8 而言,NWOA算法在收敛精度上仅次于IWOA算法,但稳定性强于其他算法。实验结果表明,NWOA算法与其他4种算法相比收敛精度较高,稳定性较好,且在全局勘测能力上更优。

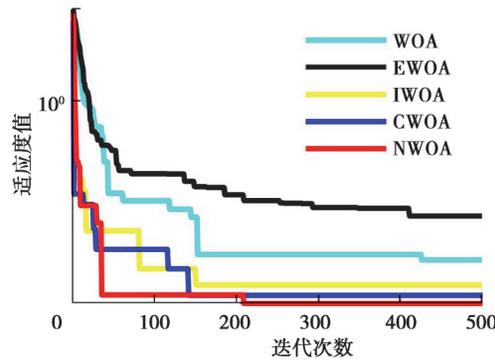


图 8 F_7 收敛曲线图

Fig. 8 F_7 convergence curve chart

表 4 多峰函数测试结果

Table 4 Experimental results of multimodal function

函数	结果	WOA	EWOA	IWOA	CWOA	NWOA
F_8	最优解	-1.256 9e+04	-1.256 0e+04	-1.256 9e+04	-9.576 2e+03	-1.256 4e+04
	最差解	7.161 9e+03	-4.681 9e+03	-5.056 8e+03	-2.273 5e+19	-4.723 9e+03
	平均值	-1.020 9e+04	-6.885 0e+03	-1.217 6e+04	-4.547 0e+17	-2.012 5e+04
	均方差	1.820 3e+03	1.751 4e+03	1.217 8e+03	3.215 2e+18	1.106 8e+03
F_9	最优解	0	0	0	0	0
	最差解	5.684 3e-14	53.354 4	0	0	0
	平均值	1.136 9e-15	2.149 8	0	0	0
	均方差	8.038 9e-14	9.325 9	0	0	0
F_{10}	最优解	8.881 8e-16				
	最差解	7.993 6e-15	7.993 6e-15	8.881 8e-16	8.881 8e-16	8.881 8e-16
	平均值	24.440 9e-15	4.227 7e-15	8.881 8e-16	8.881 8e-16	8.881 8e-16
	均方差	2.486 4e-15	1.953 8e-15	0	0	0
F_{11}	最优解	0	0	0	0	0
	最差解	0.137 0	0.351 0	0	0	0
	平均值	0.002 7	0.028 6	0	0	0
	均方差	0.019 4	0.078 7	0	0	0
F_{12}	最优解	0.082 4	0.737 3	0.032 3	0.074 4	0.052 4
	最差解	1.241 8	2.415 4	1.723 2	0.845 3	0.689 5
	平均值	0.574 4	1.483 1	0.461 4	0.437 1	0.333 1
	均方差	0.224 7	0.404 0	0.352 9	0.175 7	0.134 3
F_{13}	最优解	0.002 9	0.012 9	0.002 7	0.008 9	0.007 9
	最差解	0.292 3	0.294 8	0.240 2	0.068 2	0.081 6
	平均值	0.031 2	0.084 5	0.067 3	0.036 8	0.027 5
	均方差	0.041 3	0.049 5	0.240 2	0.016 5	0.015 4

图 9~14 表示 F_8 - F_{13} 函数的收敛图。对于函数 F_8 、 F_{12} 、 F_{13} , NWOA 算法的收敛速度优于其他算法。对于函数 F_9 - F_{11} , NWOA 算法的收敛速度远超过 WOA、EWOA 算法, 与 IWOA、CWOA 算法相差较小。这表明 NWOA 算法收敛速度较快。

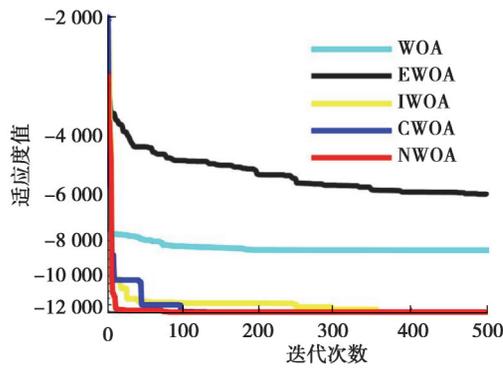


图9 F_8 收敛曲线图

Fig. 9 F_8 convergence curve chart

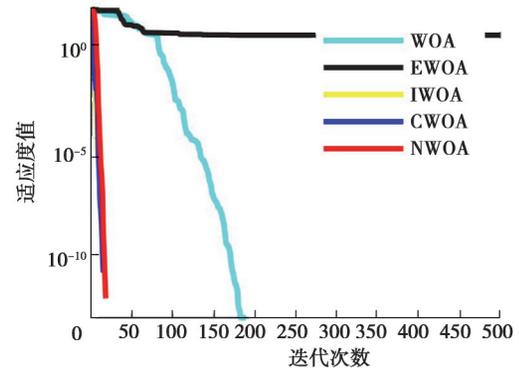


图10 F_9 收敛曲线图

Fig. 10 F_9 convergence curve chart

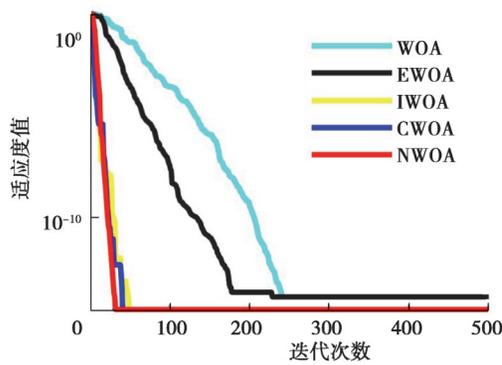


图11 F_{10} 收敛曲线图

Fig. 11 F_{10} convergence curve chart

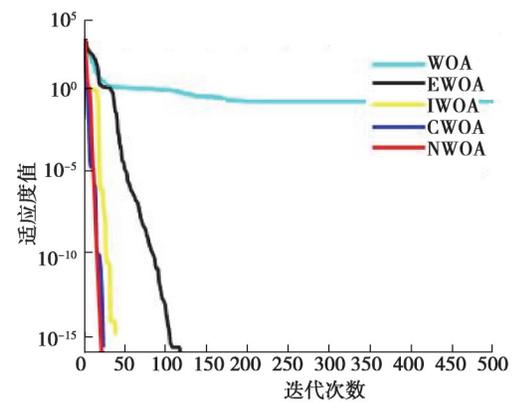


图12 F_{11} 收敛曲线图

Fig. 12 F_{11} convergence curve chart

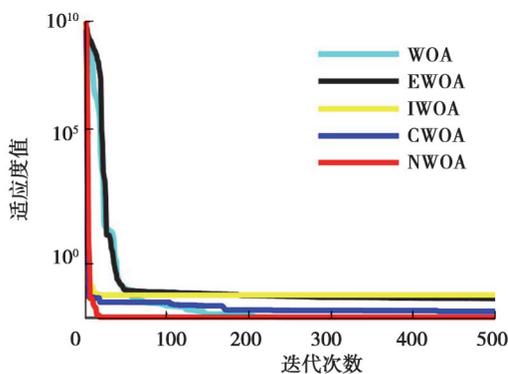


图13 F_{12} 收敛曲线图

Fig. 13 F_{12} convergence curve chart

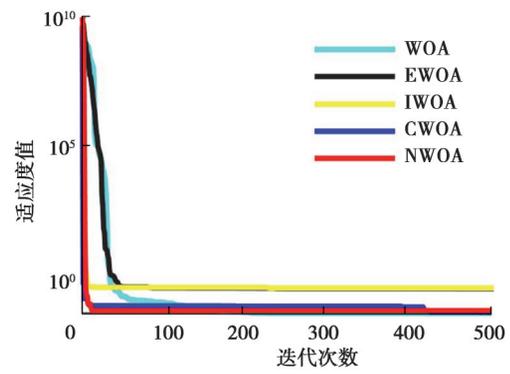


图14 F_{13} 收敛曲线图

Fig. 14 F_{13} convergence curve chart

综上所述,无论是单峰函数还是多峰函数,NWOA算法的实验结果显著更优,具有较快的收敛速度和较高的寻优精度。为了更好的验证NWOA算法的性能,表5列出了上述5种算法函数优化平均值的对比,实验的设置种群数量 $N=30, 50, 100$ 。由表5的数据可知,NWOA算法在函数迭代优化时,对函数 F_1-F_5, F_9 和 F_{11} 都已经迭代到理论最优值,无法提升寻优精度。在 F_6-F_8 和 F_{12} 中,随着种群数量的增加,提升了寻优精度。这充分说明改进算法具有较强的鲁棒性和较高寻优精度。

表 5 不同算法平均值比较

Table 5 Comparison of several algorithms mean

函数	种群数量	WOA	EWOA	IWOA	CWOA	NWOA
F_1	30	5.797 4e-70	3.611 3e-78	0	0	0
	50	9.681 0e-85	2.703 1e-96	0	0	0
	100	4.110 2-94	1.208 1e-113	0	0	0
F_2	30	9.960 2e-50	4.650 6e-46	0	2.725 6e-203	0
	50	5.793 5e-54	4.445 1e-56	6.205 5e-209	1.120 5e-207	0
	100	8.727 6e-57	4.536 4e-65	3.023 8e-199	1.968 3e-212	0
F_3	30	4.195 6e+04	1.819 0e+04	0	0	0
	50	2.985 3e+04	1.175 0e+04	0	0	0
	100	1.562 6e+04	1.143 8e+04	0	0	0
F_4	30	56.519 20	83.491 5	1.472 4e-222	1.261 9e-191	0
	50	42.884 8	85.263 8	6.209 2e-204	8.804 8e-199	0
	100	23.105 1	80.669 2	1.562 1e-198	8.829 4e-198	0
F_5	30	2.379 9e-145	2.582 6e-150	0	0	0
	50	3.937 5e-163	3.276 9e-186	0	0	0
	100	1.110 9e-190	1.113 1e-227	0	0	0
F_6	30	1.434 6	1.193 4	0.702 1	0.562 8	0.539 6
	50	0.968 1	0.801 0	0.581 5	0.412 1	0.376 4
	100	0.705 4	0.324 8	0.508 6	0.354 0	0.219 2
F_7	30	0.004 1	0.013 0	8.340 9e-05	7.007 7e-05	6.387 0e-05
	50	0.001 9	0.006 5	3.276 9e-05	4.371 7e-05	2.842 3e-05
	100	0.001 1	0.003 1	2.295 1e-05	2.192 2e-05	1.794 7e-05
F_8	30	-1.020 9e+04	-6.885 0e+03	-1.207 6e+04	-4.547 0e+04	-2.012 5e+04
	50	-1.115 8e+04	-7.195 6e+03	-1.243 7e+04	-4.993 0e+04	-1.118 6e+04
	100	-1.193 7e+04	-8.333 1e+03	-1.254 7e+04	-1.434 2e+04	-1.092 3e+04
F_9	30	1.136 9e-14	2.149 8	0	0	0
	50	0	0	0	0	0
	100	1.136 9e-15	0	0	0	0
F_{10}	30	24.440 9e-15	4.227 7e-15	8.881 8e-16	8.881 8e-16	8.881 8e-16
	50	4.654 1e-15	4.298 8e-15	8.881 8e-16	8.881 8e-16	8.881 8e-16
	100	3.872 5e-15	4.227 7e-15	8.881 8e-16	8.881 8e-16	8.881 8e-16
F_{11}	30	0.002 7	0.028 6	0	0	0
	50	0.005 3	0.003 9	0	0	0
	100	0.005 4	0.002 8	0	0	0
F_{12}	30	0.574 4	1.483 1	0.461 4	0.437 1	0.333 1
	50	0.010 7	0.060 8	0.035 9	0.023 5	0.020 8
	100	8.194 0e-04	0.021 7	0.027 7	0.012 0	0.013 8
F_{13}	30	0.031 2	0.084 5	0.067 3	0.036 8	0.027 5
	50	0.198 4	1.140 0	0.350 2	0.330 6	0.268 3
	100	0.035 3	0.859 4	0.359 5	0.204 1	0.181 0

3.5 算法时间复杂度分析

对于基本的 WOA,种群数量是 N ,空间维度是 D ,最大迭代次数是 Max_iter ,对应时间复杂度是 $O(\text{Max_iter} * N * D)$ 。NWOA 算法主要在基本 WOA 上对初始化种群、收敛因子和惯性权重这 3 方面进行改进。

算法的时间复杂度分析如下:初始化种群的时间复杂度为 $O(N*D)$,收敛因子和惯性权重的时间复杂度为 $O(\text{Max_iter})$ 。NOWA算法总的时间复杂度为 $O(\text{Max_iter}*N*D+N*D+2*\text{Max_iter})$ 。根据大 O 准则定义知,当迭代次数较大时,改进算法时间复杂度近似等于基本WOA时间复杂度。表6列出了13种测试函数分别在WOA、IWOA、EWOA、CWOA和NWOA算法的运行30次所累积的总时间。从表6的统计结果看,NOWA算法相较于基本WOA算法30次实验累积时间都有不同程度减少。尤其是函数 F_7 和 F_{11} ,效果较明显。就函数 F_2 和 F_5 而言,NOWA算法在30次实验中累积时间比IWOA略微较长,其他实验结果都是NOWA算法执行时间较短,效率较高。除此之外,通过观察数据来看,这5种算法的时间复杂度实验结果相差不是特别大,都在同一个数量级。

表6 30次实验各算法的累积时间

Table 6 Cumulative time of 30 experiments					
函数	WOA	EWOA	IWOA	CWOA	NWOA
F_1	3.632 4	4.441 3	3.572 3	4.106 9	3.439 7
F_2	3.661 4	3.863 2	3.353 9	4.650 3	3.502 3
F_3	10.513 2	15.398 5	10.725 9	13.952 9	10.348 6
F_4	3.539 4	5.384 0	3.432 6	4.264 4	3.263 7
F_5	3.470 5	3.873 2	2.836 4	4.169 4	3.387 2
F_6	3.447 7	3.783 4	3.488 0	4.274 2	3.341 7
F_7	7.562 7	9.490 1	6.538 7	9.201 7	6.506 1
F_8	4.797 5	5.607 8	4.005 1	5.797 5	4.153 4
F_9	3.895 2	5.816 2	3.455 8	4.612 3	3.324 9
F_{10}	3.892 6	6.993 6	4.005 8	4.963 3	3.673 5
F_{11}	5.835 9	5.371 5	4.126 6	6.125 9	4.713 5
F_{12}	14.894 9	16.446 6	15.235 2	18.554 8	14.769 3
F_{13}	14.918 9	18.441 9	16.591 4	16.005 7	14.671 1

4 结 论

针对鲸鱼优化算法中存在收敛速度慢和寻优精度低等问题,提出一种改进的鲸鱼优化算法(NWOA)。采用Iterative映射来初始化种群,使种群尽可能均匀分布在空间,提高算法的收敛速度;利用非线性拟合策略构造收敛因子和惯性权重,以平衡算法的全局勘测能力和局部收敛能力。通过对13个测试函数进行仿真实验,结果表明NWOA算法收敛速度较快且收敛精度高,体现了该算法优越的性能。

参考文献

- [1] Mirjalili S, Lewis A. The whale optimization algorithm[J]. Advances in Engineering Software, 2016, 95: 51-67.
- [2] Iscan H, Gunduz M. An application of fruit fly optimization algorithm for traveling salesman problem[J]. Procedia Computer Science, 2017, 111: 58-63.
- [3] Kennedy J, Eberhart R. Particle swarm optimization[C]//Proceedings of ICNN'95 - International Conference on Neural Networks. November 27 - December 1, 1995. Perth, WA, Australia: IEEE, 2002: 1942-1948.
- [4] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer[J]. Advances in Engineering Software, 2014, 69: 46-61.
- [5] Trivedi I N, Pradeep J, Narottam J, et al. Novel adaptive whale optimization algorithm for global optimization[J]. Indian Journal of Science and Technology, 2016, 9(38): 319-326.
- [6] Mafarja M M, Mirjalili S. Hybrid Whale Optimization Algorithm with simulated annealing for feature selection[J]. Neurocomputing, 2017, 260: 302-312.
- [7] 钟明辉, 龙文. 一种随机调整控制参数的鲸鱼优化算法[J]. 科学技术与工程, 2017, 17(12): 68-73.

- Zhong M H, Long W. Whale optimization algorithm based on stochastic adjustment control parameter[J]. Science Technology and Engineering, 2017, 17(12): 68-73.(in Chinese)
- [8] 凌颖,杨春燕,黎新,等.改进灰狼优化算法的研究[J].科学技术创新,2020(16):61-63.
Ling Y, Yang C Y, Li X, et al. Research on improving Grey Wolf optimization algorithm [J]. Science and Technology Innovation,2020(16):61-63. (in Chinese)
- [9] 郭振洲,王平,马云峰,等.基于自适应权重和柯西变异的鲸鱼优化算法[J].微电子学与计算机,2017,34(9):20-25.
Guo Z Z, Wang P, Ma Y F, et al. Whale optimization algorithm based on adaptive weight and cauchy mutation[J]. Microelectronics & Computer, 2017, 34(9): 20-25.(in Chinese)
- [10] 王坚浩,张亮,史超,等.基于混沌搜索策略的鲸鱼优化算法[J].控制与决策,2019,34(9):1893-1900.
Wang J H, Zhang L, Shi C, et al. Whale optimization algorithm based on chaotic search strategy[J]. Control and Decision, 2019, 34(9): 1893-1900.(in Chinese)
- [11] 石旭东,姜鸿晔.基于鲸鱼优化小波神经网络的飞行航迹缺失数据预测方法[J].计算机应用与软件,2020,37(7):200-205.
Shi X D, Jiang H Y. Prediction method of flight track missing data based on whale optimization wavelet neural network[J]. Computer Applications and Software, 2020, 37(7): 200-205.(in Chinese)
- [12] 吴坤,谭劭昌.基于改进鲸鱼优化算法的无人机航路规划[J].航空学报,2020,41(S2):107-114.
Wu K, Tan S C. Path planning of UAVs based on improved whale optimization algorithm[J]. Acta Aeronautica et Astronautica Sinica, 2020, 41(S2): 724286.(in Chinese)
- [13] 王思涵,黎阳,李新宇.基于鲸鱼群算法的柔性作业车间调度方法[J].重庆大学学报,2020,43(1):1-11.
Wang S H, Li Y, Li X Y. An improved whale swarm algorithm for flexible job-shop scheduling problem[J]. Journal of Chongqing University, 2020, 43(1): 1-11.(in Chinese)
- [14] 张东洋,杨永辉,储茂祥,等.基于2维鲸鱼优化加权的WGG-Otsu算法的钢板表面缺陷图像分割[J].安徽大学学报(自然科学版),2020,44(3):72-77.
Zhang D Y, Yang Y H, Chu M X, et al. Steel plate surface defect image segmentation based on two-dimensional weighted WGG-Otsu with whale optimization[J]. Journal of Anhui University (Natural Science Edition), 2020, 44(3): 72-77.(in Chinese)
- [15] 凌颖.鲸鱼优化算法及其应用研究[D].南宁:广西民族大学,2018.
Ling Y. Research on whale optimization algorithm and its application[D].Nanning: Guangxi University for Nationalities, 2018. (in Chinese)
- [16] 王日宏,李祥,李娜.基于高斯扰动和混沌初始化的狼群算法[J].计算机工程与设计,2019,40(10):2879-2884.
Wang R H, Li X, Li N. Wolf pack algorithm based on Gaussian disturbance and chaotic initialization[J]. Computer Engineering and Design, 2019, 40(10): 2879-2884.(in Chinese)
- [17] 王梦娜,王秋萍,王晓峰.基于 Iterative 映射和单纯形法的改进灰狼优化算法[J].计算机应用,2018,38(S2):16-20,54.
Wang M N, Wang Q P, Wang X F. Improved grey wolf optimization algorithm based on iterative mapping and simplex method [J]. Journal of Computer Applications, 2018, 38(S2): 16-20, 54.(in Chinese)
- [18] 王涛,Chellali R.非线性权重和收敛因子的鲸鱼算法[J].微电子学与计算机,2019,36(1):11-15.
Wang T, Chellali R. Whale optimization algorithm with nonlinear weight and convergence factor[J]. Microelectronics & Computer, 2019, 36(1): 11-15.(in Chinese)
- [19] 石丽莉.一种拟合下降非线性惯性权重 PSO 算法[J].技术与市场,2019,26(7):22-24.
Shi L L. A PSO algorithm for fitting decreasing nonlinear inertia weight[J]. Technology and Market, 2019, 26(7): 22-24. (in Chinese)
- [20] 覃溪,龙文.基于随机差分变异的改进鲸鱼优化算法[J].中国科技论文,2018,13(8):937-942.
Qin X, Long W. An improved whale optimization algorithm based on stochastic differential mutation[J]. China Science Paper, 2018, 13(8): 937-942.(in Chinese)