

郭岩, 李目, 徐佳祺, 等. 基于自适应权重和动态螺旋参数的鲸鱼算法[J]. 湖南科技大学学报(自然科学版), 2025, 40(1): 78-87. doi:10.13582/j.cnki.1672-9102.2025.01.011

GUO Y, LI M, XU J Q, et al. Whale Optimization Algorithm Based on Adaptive Weights and Dynamic Spiral Parameters [J]. Journal of Hunan University of Science and Technology (Natural Science Edition), 2025, 40(1): 78-87. doi:10.13582/j.cnki.1672-9102.2025.01.011

基于自适应权重和动态螺旋参数的鲸鱼算法

郭岩, 李目*, 徐佳祺, 王奕峰, 雷佳伟

(湖南科技大学 信息与电气工程学院, 湖南 湘潭 411201)

摘要:针对鲸鱼优化算法(Whale Optimization Algorithm, WOA)收敛性能差,易陷入局部最优值的问题,提出一种基于自适应权重和动态螺旋参数的鲸鱼优化算法(ADS-WOA).首先,引进一种非线性距离控制因子,提高算法的开发勘探能力;其次,提出一种自适应惯性权重,提高算法后期的收敛速度;最后,改进WOA的螺旋形状参数,改变鲸鱼种群固定的螺旋上升路径,提高算法的收敛精度和收敛速度.将ADS-WOA和其他优化算法在10组基准测试函数上进行测试,并对测试结果进行Wilcoxon秩和检验,以及对3种改进策略和ADS-WOA时间复杂度的分析.结果表明:ADS-WOA算法具有更好的寻优和抗干扰能力,能够应用于高维复杂环境下的目标函数优化问题.

关键词:鲸鱼优化算法;距离控制因子;惯性权重;Wilcoxon秩和检验

中图分类号:TP301

文献标志码:A

文章编号:1672-9102(2025)01-0078-10

Whale Optimization Algorithm Based on Adaptive Weights and Dynamic Spiral Parameters

GUO Yan, LI Mu, XU Jiaqi, WANG Yifeng, LEI Jiawei

(School of Information and Electrical Engineering, Hunan University of Science and Technology, Xiangtan 411201, China)

Abstract: Aiming at the poor convergence performance of whale optimization algorithm (WOA) and the problem of local optimization, a whale optimization algorithm (ADS-WOA) based on adaptive weights and dynamic spiral parameters is proposed. First of all, a nonlinear distance control factor is introduced to improve the exploitation and exploration ability of the algorithm. Secondly, an adaptive inertia weight is proposed to improve the convergence speed of the algorithm in the later stage. Finally, the spiral shape parameters of WOA are improved, the fixed spiral path of whale populations is changed, and the convergence accuracy and convergence speed of the algorithm are improved. ADS-WOA and other optimization algorithms are tested on 10 sets of benchmark functions, and the test results are tested with Wilcoxon rank sum test, as well as an analysis of three improvement strategies and ADS-WOA time complexity. The results show that compared with WOA, ADS-WOA algorithm has better optimization and anti-interference capabilities, and can cope with objective function optimization problems in high-dimensional complex environments.

Keywords: whale optimization algorithm; distance control factor; inertia weights; wilcoxon rank sum test

收稿日期:2023-04-18

基金项目:国家自然科学基金资助项目(61404049);湖南省自然科学基金资助项目(2020JJ6031);湖南省教育厅重点项目资助(22A0324);湖南省教育厅优秀青年基金资助项目(17B094);湘潭市创建国家创新型城市建设专项项目资助(NY-YB20221042)

*通信作者, E-mail: mli@hnust.edu.cn

在过去的几十年里,智能优化算法凭借在求解参数识别、复杂未知问题上的优异性能,受到了广泛的关注,被成功应用于工程设计、计算机科学等领域.智能优化算法受生物群体行为活动和自然规律现象的启发,具有独特的性能机制,经典的代表性算法如遗传算法^[1]、蚁群算法^[2]、粒子群算法^[3]、模拟退火算法^[4]等,都具有简单易行、调节参数少、效果显著的优势.鲸鱼优化算法(Whale Optimization Algorithm, WOA)^[5]作为一种新颖的群智能优化算法,是MIRJALILI等学者受座头鲸特殊捕食行为启发,于2016年提出来的.相比于近年来较新颖的智能优化算法如樽海鞘算法^[6]、灰狼优化算法^[7]、蝴蝶优化算法^[8]在解决高维、复杂、混合问题中性能较差,WOA较好地解决了这些问题.目前,WOA已应用到众多实际工程领域,OULADBRAHIM等^[9]将基于WOA的改进人工神经网络对APIX70钢管进行裂纹识别试验,在预测不同的场景上取得不错的效果;DING等^[10]提出应用WOA的混合预测模型,对短期风力发电进行预测,在预测精度和稳定性方面,性能更优,计算成本更低;EL-FERGANY等^[11]提出利用WOA对PEM燃料电池模型参数进行优化,从而创建了一种精确的PEM燃料电池模型.然而,传统WOA依然存在很多缺陷,如算法前期探索能力欠缺,后期开发能力不足,容易使算法陷入局部最优,同时收敛性能表现也较差.

针对以上缺陷,众多学者对鲸鱼优化算法做出了改进.刘磊等^[12]提出集中全局搜索方法能力的WOA,通过改进螺旋上升策略来改变鲸鱼固定的捕食上升运动方式,一定程度提高了算法的全局寻优能力;程浩森等^[13]针对WOA后期易陷入局部最优,提出耦合双权重因子对后期种群进行随机扰动,进一步增强了后期鲸鱼种群的更新能力;耿召里等^[14]针对WOA在解决高维复杂问题时存在收敛性能不佳、搜索可行解效率较差的问题,提出一种最优最差个体混合反向学习策略,提高WOA的全局寻优能力;赵传武等^[15]提出非线性权重的WOA,结合2种非线性权重对鲸鱼种群的运动方式进行加权操作,维持了WOA探索与开发的平衡;武泽权等^[16]结合准反向学习初始化种群、自适应权重和随机差分变异策略提高了算法性能;黄辉先等^[17]提出在算法前期利用精英个体及时引导种群,提高算法求解的目的性,在后期引入混沌动态权重因子加强算法的全局寻优能力;孔芝等^[18]提出一种自适应搜索策略来优化随机搜索阶段,个体以一定概率选择要更新的方式,以此增加种群多样性,提高算法的全局搜索能力.

本文针对WOA求解复杂非线性优化问题存在的不足,提出一种基于自适应权重和动态螺旋参数的鲸鱼优化算法(Whale Optimization Algorithm based on Adaptive Weights and Dynamic Spiral Parameters, ADS-WOA).首先,引进一种非线性距离控制因子,平衡种群的多样性和收敛性;其次,提出一种自适应惯性权重,提高算法的探索 and 开发能力;最后,改进WOA的螺旋控制参数,动态调整螺旋上升行径,进一步提高了算法的收敛性能.将ADS-WOA和其他优化算法在10组基准测试函数上进行测试,随后对测试结果进行Wilcoxon秩和检验,并对3种改进策略和ADS-WOA时间复杂度进行分析,试验结果表明了ADS-WOA的有效性.

1 鲸鱼优化算法

座头鲸通过协调气泡网攻击和螺旋上升运动,从而达到捕获猎物的目的.具体的捕食行为可分为3个阶段:围捕猎物、泡泡网攻击、搜索包围.

1.1 围捕猎物

座头鲸包围猎物,并以与猎物的相对位置确定自身位置好坏,在猎物位置或离猎物最近的座头鲸被定为最优鲸鱼个体,其他个体将依据最优个体来调整自己的位置,此行为可描述如式(1)和式(2)所示.

$$D = |C \cdot X^*(t) - X(t)|; \quad (1)$$

$$X(t+1) = X^*(t) - AD. \quad (2)$$

式中: D 为包围步长; t 为当前迭代次数; $X(t)$ 为当前个体位置; $X^*(t)$ 为当前最优个体位置; $X(t+1)$ 为更新后的个体位置; A 和 C 为系数向量,如式(3)和式(4)所示.

$$A = 2ar - a; \quad (3)$$

$$C = 2r. \quad (4)$$

式中: r 为元素值在 $[0,1]$ 范围内的随机向量; a 为随迭代次数 t 线性减小的距离收敛因子,如式(5)所示.

$$a = 2 - 2t/T. \quad (5)$$

式中: T 为最大迭代次数.

1.2 泡泡网攻击

WOA 算法将泡泡网攻击行为分为 2 个阶段,即收缩包围阶段和螺旋上升阶段.

1) 收缩包围阶段:减少 a 的取值,实现对 A 取值范围的控制. A 的取值范围为 $[-a, a]$, 当 a 逐渐减小时,使得鲸鱼个体的新位置能够定义在最优个体和当前个体位置之间的任一位置,从而实现对猎物的包围.

2) 螺旋上升阶段:该阶段首先计算当前鲸鱼个体与最优鲸鱼个体的位置距离,在两者位置之间建立一个螺旋上升方程,模拟鲸鱼螺旋上升运动如式(6)所示.

$$X(t+1) = D^* e^{bl} \cos(2\pi l) + X^*(t). \quad (6)$$

式中: $D^* = |X^*(t) - X(t)|$ 为当前鲸鱼个体与最优鲸鱼个体的距离; b 为控制螺旋形状的常数 1; l 为 $[-1, 1]$ 之间的一个随机数.鲸鱼围绕圆周逐渐缩小的螺旋形状路径奔向猎物,为了实现这一行为,设定收缩包围和螺旋上升分别有 50% 的概率进行,以达到收缩包围和螺旋上升同步进行的目的.其数学模型公式如式(7)所示.

$$X(t+1) = \begin{cases} X^*(t) - AD, p < 0.5; \\ D^* e^{bl} \cos(2\pi l) + X^*(t), p \geq 0.5. \end{cases} \quad (7)$$

式中: p 为 $[0, 1]$ 之间一随机数.

1.3 搜索包围

实际上,鲸鱼个体还会依据彼此之间的位置进行随机搜索.通过 $|A| > 1$ 使得鲸鱼个体远离占据最优位置的鲸鱼个体,并根据随机选择的个体位置来更新自己在全局探索阶段的位置.其数学模型如式(8)和式(9)所示.

$$D' = |CX_{\text{rand}}(t) - X(t)|; \quad (8)$$

$$X(t+1) = X_{\text{rand}} - AD'. \quad (9)$$

式中: X_{rand} 为从当前种群中随机选择的鲸鱼个体位置; D' 为随机包围步长.

当 $|A| \leq 1$ 时,鲸鱼种群不再进行随机搜索,转而收缩包围圈并攻击猎物,完成局部开发.

2 改进的鲸鱼优化算法

2.1 非线性距离控制因子改进策略

在基本 WOA 算法中,参数 A 对种群全局探索和局部开发影响显著,而决定 A 取值范围的 a 服从随迭代次数递减的原则.当 a 取值较大时, A 值较大,具有较强的全局探索能力;当 a 取值较小时, A 值较小,具有较强的局部开发能力.平衡种群全局探索和局部开发的能力,对算法性能的提升十分关键.对此,本文引入文献[19]提出的一种非线性距离控制因子策略,如式(10)所示.

$$a = \frac{2}{1 + ue^{t/T-1}}. \quad (10)$$

式中: T 为最大迭代次数;在迭代前期,距离控制因子以较慢的速率递减,算法获得较强的全局探索能力,提高了多样性,而在迭代后期,距离控制因子以较快的速率递减,算法获得较强的局部开发能力,加快了收敛的进程; u 为控制参数,此处 u 取 2.

2.2 自适应惯性权重

惯性权重对鲸鱼优化算法寻优能力的提升,具有非常重要的作用,合适的惯性权重有利于维持算法全局探索和局部开发的平衡.由于 WOA 在优化求解的过程中,线性的惯性权重调整策略若选择不合适,将影响算法的收敛速度.因此,本文提出了一种自适应惯性权重,如式(11)所示.

$$w = \frac{1 + \cos[\pi (\frac{t}{T})^k]}{2}. \quad (11)$$

式中: t 为当前迭代次数; T 为最大迭代次数; k 为调节系数, 决定 w 的收敛速度. w 在前期以较快的速率减小, 确保算法前期具有良好的全局探索能力, 从而跳出局部最优, 维持种群的多样性. 后期 w 以较慢的速率减小, 确保算法后期具有较好的局部开发能力, 加强对最优解的搜寻精度. 因此, 提出适应度惯性权重后, 式(2)和式(6)更新为

$$X(t+1) = w \cdot X^*(t) - A \cdot D. \quad (12)$$

$$X(t+1) = D^* \cdot e^{bl} \cdot \cos(2\pi l) + w \cdot X^*(t). \quad (13)$$

2.3 改进螺旋上升策略

基本鲸鱼优化算法中鲸鱼在搜索猎物的途中, 会按照一半的概率执行螺旋上升或者探索包围策略, 但由于 WOA 的螺旋式上升策略会按照既定的轨迹进行, 从而导致鲸鱼在搜寻猎物时错过最佳猎物, 使算法陷入局部最优. 式(7)中, b 是控制鲸鱼螺旋上升路径的参数, 原始的参数设定为常数 1, 这样会导致鲸鱼的包围行进方式过于单一, 降低鲸鱼解决复杂或存在干扰的问题的能力. 目前, 已有研究学者提出过类似的改进, 如文献[12]提出一种螺旋更新策略, 提高了算法全局搜索性能.

为应对这种不利局面, 本文提出一种螺旋上升策略, 如式(14)所示. 通过一种自适应性螺旋控制参数, 使得鲸鱼跳出固定的猎物搜寻机制, 转而采用自适应的螺旋行进路线, 实现更好包围猎物的目的.

$$\begin{cases} X(t+1) = wX^* + D^* e^{bl} \cos(2\pi l); \\ b = 10^{1+\mu \frac{t}{T}}. \end{cases} \quad (14)$$

式中: μ 为调节螺旋包围圈大小的控制参数; w 为上节提出的自适应惯性权重; T 为最大迭代次数.

与文献[12]所提出的前期加强全局搜索能力不同, 本文设定合理的自适应螺旋参数初始值, 且参数随迭代次数加快增长, 在适当缩小鲸鱼包围圈的情况下, 使得鲸鱼能够快速且精准地收缩包围猎物. 改进螺旋上升策略和文献[12]提出的一种螺旋更新策略的螺旋控制参数 b 随迭代次数如图 1 所示, 这里 μ 设置为 0.7.

该自适应性螺旋控制参数在前期以较小的值避免鲸鱼错过猎物, 从而增加鲸鱼寻求最优解的概率, 提高 WOA 的全局探索能力. 后期参数加快增长, 缩小鲸鱼对猎物的包围圈, 使得鲸鱼个体能够快速获取猎物, 促进了鲸鱼的局部开发能力, 提高了算法的收敛精度.

2.4 ADS-WOA 的执行步骤

对基本 WOA 算法进行以上策略改进后, 得到的 ADS-WOA 伪代码如算法 1 所示.

算法 1: ADS-WOA 算法

输入: 设置种群规模 N , 最大迭代次数 T , 维数 D , 种群活动的上界 ub 和下界 lb .

输出: 最优个体.

1. 采用随机初始化种群;
 2. 计算种群个体的适应度值, 并记录当前最优个体;
 3. **while** ($t \leq T$) **do**
 4. **for** $i = 1$ to N **do**
 5. 根据式(10)计算控制参数 a 的值; 根据式(11)计算控制参数 w 的值;
 6. 更新其他参数 A 、 C 、 l 和 p 的值; 根据式(14)计算螺旋参数 b 的值;
 7. **if** ($p < 0.5$) **then**
-

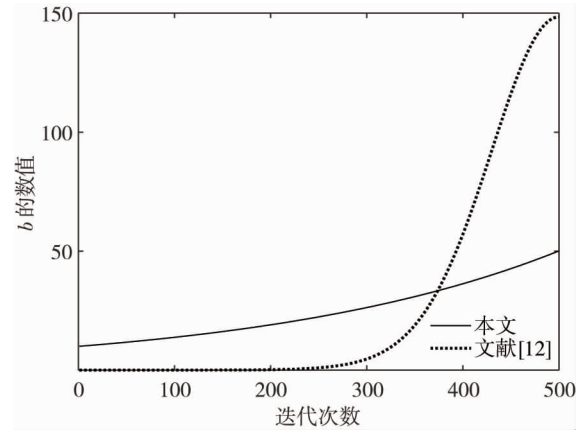


图 1 螺旋控制参数 b 的迭代变化曲线

```

8.      if (  $|A| \geq 1$  ) then
9.          根据式(9)更新当前个体的位置;
10.     else if (  $|A| < 1$  ) then
11.         根据式(2)更新当前个体的位置;
12.     end if
13.     else if (  $p \geq 0.5$  ) then
14.         根据式(14)更新当前个体的位置;
15.     end if
16. end for
17. 更新当前最优个体及位置;
18.  $t = t + 1$ ;
19. end while

```

3 仿真实验和分析

3.1 测试函数及实验环境

选择文献[20]中的 $f_1 \sim f_3, f_6, f_8, f_{10} \sim f_{11}$ 和文献[21]中 f_5, f_9, f_{10} , 一共 10 个测试函数进行仿真实验. 仿真实验环境: 操作系统 Microsoft Windows 10, Intel G5400 CPU, 主频 3.70 GHz, 8 GB 内存, 实验测试软件为 MATLAB R2018a. 评价指标包括优化结果的平均值 (Mean) 和标准差 (STD)^[12], 以及运行时间 (t, s).

3.2 与其他改进优化算法的对比分析

选取的对比算法包括 WOA, IWOA^[16], GS-WOA^[12], C-A-WWOA^[13]. 为确保实验的公正性, 本次实验设置最大迭代次数 $T = 500$, 种群规模 $N = 30$, 测试函数的维数 D 分别为 30 维、200 维、500 维、1 000 维, 各算法均独立运行 50 次, 根据实验数据算出平均值 (Mean) 和标准差 (STD) 2 项指标数据, 黑色加粗字体表示同维度函数上最好的性能指标结果, 实验结果如表 1 所示.

从表 1 可以得知: ADS-WOA 算法在 10 组测试函数的实验结果, 除了在维数为 200, 500, 1 000 维 F_4 的标准差, 以及在 1 000 维 F_5 的双指标和 500 维 F_9 的标准差表现略劣于其他算法外, ADS-WOA 的各项指标均好于或者等于其他算法, 说明改进后的 ADS-WOA 算法整体上获得了较好的收敛精度和稳定性. 在 F_1, F_2, F_3, F_7 , ADS-WOA 和 C-A-WWOA 均取得理论最优值 0, IWOA 和 GS-WOA 没有收敛到理论最优值; 在 F_4 , ADS-WOA 取得平均值指标远好于其他算法, 仅在 200, 500 维时的稳定性稍逊于 WOA, 1 000 维时不如 WOA 和 C-A-WWOA; 在 F_5, F_9, F_{10} , ADS-WOA 取得的平均值和标准差指标较优于其他算法, 只是在 1 000 维度 F_5 和 500 维 F_9 略劣于 GS-WOA, 说明改进后的算法表现出较好的探索和开发能力; 在 F_6 , 该测试函数的理论最优值为 $-418.98D$, ADS-WOA 在各维数上均取得非常接近最优值的平均值指标, 2 项指标远优于 WOA, 稍优于其他算法, 只是在 500 维时, ADS-WOA 和 C-A-WWOA 取得相同的平均值. 其中, 在维数 30, 200, 1 000 时的平均值稍优于 IWOA, GS-WOA, C-A-WWOA, 标准差较优于 IWOA, MWOA, C-A-WWOA; 在 F_7, F_8 , 改进算法都取得相同的测试结果, 优于 WOA, 三者的优化效果相同; 在 F_9, F_{10} , ADS-WOA 得到的平均值和标准差均远优于其他算法, 证明 ADS-WOA 具有较好的收敛精度和鲁棒性.

通过观察不同算法在不同函数的收敛曲线图, 能够直观地了解到 ADS-WOA 算法的收敛情况. 本文给出各种改进算法在维数为 200 维时, $F_1 \sim F_{10}$ 的收敛曲线图, 如图 2 所示. 与其他算法相比, ADS-WOA 算法具有更好的收敛效果.

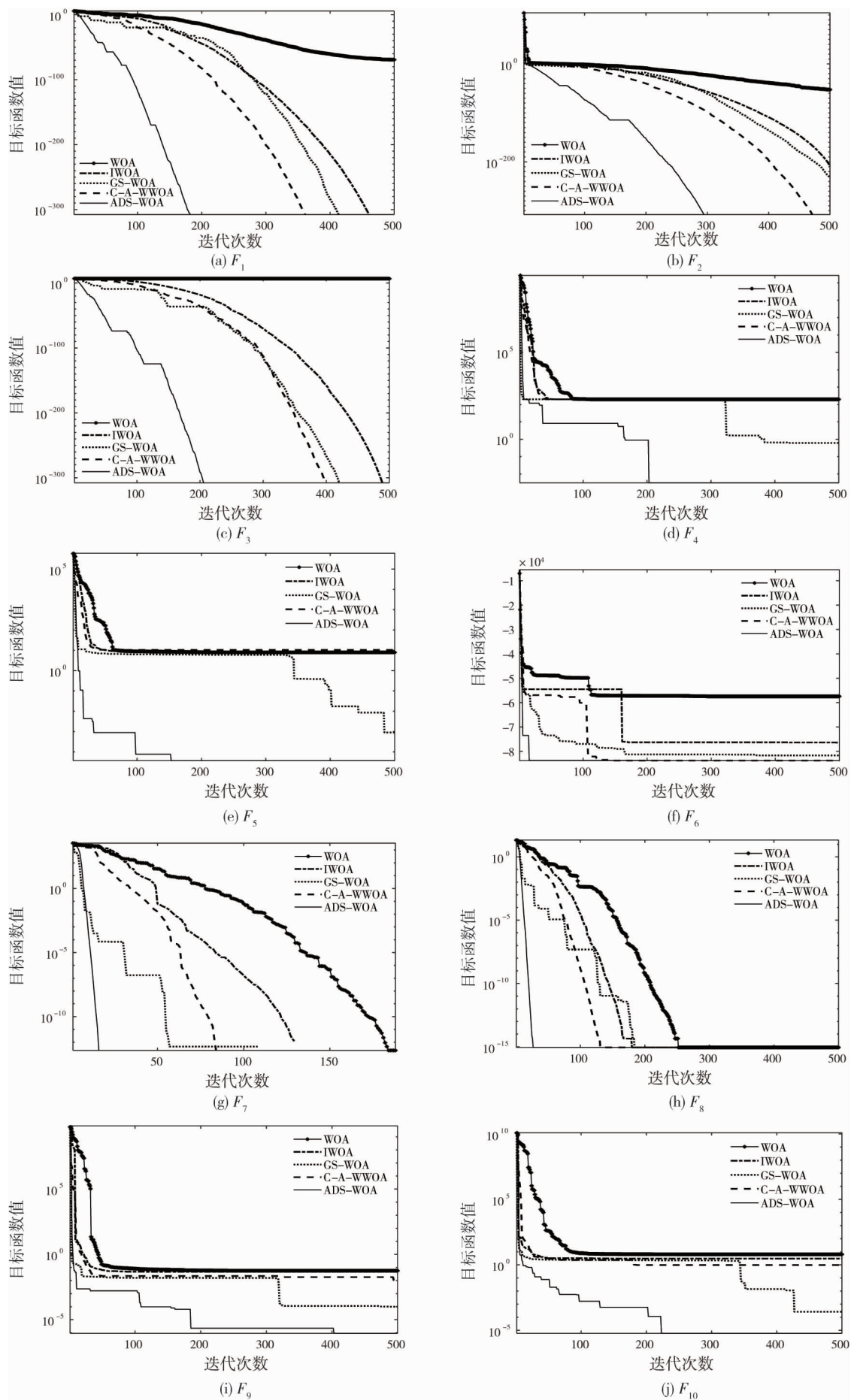


图 2 5 种算法在 10 组测试函数的收敛曲线

3.3 3 种改进策略对算法的性能提升分析

为了验证 3 种改进策略分别对 WOA 性能的提升情况,将仅采取非线性距离控制因子策略改进的 WOA 记为 WOA-1,仅采取自适应惯性权重策略改进的 WOA 记为 WOA-2,仅采取改进螺旋式上升策略改进的 WOA 记为 WOA-3,并分别测试其在 $F_1 \sim F_{10}$ 上的寻优情况.上述算法均运行 50 次,设置最大迭代次数为 500,种群数量为 30,个体维数为 200,记录试验数据如表 2 所示.从表 2 可知:WOA-1 在单峰函数 $F_1 \sim F_3$,有一定收敛精度和稳定性的提升,原因是非线性距离控制因子一定程度上平衡了算法的全局探索与局部开发能力;WOA-2 的整体提升效果较好,特别是对 $F_1 \sim F_3, F_6$ 的提升,原因是自适应惯性权重提高了种群多样性;WOA-3 对 F_5, F_6, F_9, F_{10} 的提升效果较好,提高了一定的收敛精度和稳定性,原因是非线性参数 b 帮助鲸鱼种群跳出局部最优.ADS-WOA 将上述 3 种改进策略整合在一起,最终取得了远好于单一改进策略的性能提升.

表 1 4 种算法在 10 组不同维数基准函数上的寻优数据

函数	维数	WOA		IWOA		GS-WOA		C-A-WWOA		ADS-WOA	
		Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
F_1	30	1.38E-69	9.65E-69	0	0	0	0	0	0	0	0
	200	3.63E-71	2.24E-70	0	0	0	0	0	0	0	0
	500	2.69E-70	1.29E-69	0	0	0	0	0	0	0	0
	1 000	3.39E-67	1.82E-66	0	0	0	0	0	0	0	0
F_2	30	4.47E-50	2.07E-49	2.95E-211	0	2.08E-228	0	0	0	0	0
	200	8.16E-48	5.61E-47	7.51E-209	0	7.75E-229	0	0	0	0	0
	500	1.11E-47	4.85E-47	3.84E-207	0	1.27E-222	0	0	0	0	0
	1 000	2.36E-47	1.59E-46	6.20E-208	0	2.03E-223	0	0	0	0	0
F_3	30	4.30E+04	1.41E+04	0	0	0	0	0	0	0	0
	200	3.00E+07	1.09E+07	0	0	0	0	0	0	0	0
	500	3.00E+07	1.09E+07	0	0	0	0	0	0	0	0
	1 000	1.27E+08	4.87E+07	0	0	0	0	0	0	0	0
F_4	30	2.82E+01	4.53E-01	2.82E+01	3.08E-01	4.26E-01	6.77E-01	2.85E+01	2.73E-01	1.44E-02	5.76E-02
	200	1.98E+02	1.51E-01	1.98E+02	1.74E-01	3.67E-01	1.13E+00	1.98E+02	1.86E-01	6.34E-02	1.58E-01
	500	4.96E+02	3.55E-01	4.96E+02	4.56E-01	9.08E-01	3.46E+00	4.96E+02	4.31E-01	4.19E-01	9.66E-01
	1 000	9.94E+02	8.59E-01	9.93E+02	9.35E-01	2.47E+00	8.61E+00	9.92E+02	6.47E-01	1.64E+00	2.90E+00
F_5	30	4.05E-01	2.67E-01	4.53E-01	1.92E-01	1.25E-03	2.88E-03	6.61E-01	2.48E-01	5.43E-05	1.35E-04
	200	1.06E+01	2.77E+00	8.94E+00	3.04E+00	8.02E-03	1.24E-02	8.16E+00	3.17E+00	6.52E-04	2.18E-03
	500	3.16E+01	7.77E+00	2.57E+01	6.49E+00	3.87E-02	6.05E-02	2.15E+01	8.30E+00	6.67E-03	1.59E-02
	1 000	6.83E+01	1.55E+01	5.48E+01	1.87E+01	2.70E-02	5.41E-02	5.08E+01	1.77E+01	2.96E-02	1.43E-01
F_6	30	-1.02E+04	1.71E+03	1.22E-04	7.13E+02	-1.21E+04	8.01E+02	-1.24E+04	4.83E+02	-1.26E+04	1.96E+00
	200	-6.88E+04	1.15E+04	-8.25E+04	2.67E+03	-7.86E+04	6.33E+03	-8.34E+04	9.20E+02	-8.38E+04	2.40E+01
	500	-1.72E+05	2.84E+04	-2.07E+05	6.85E+03	-2.03E+05	1.25E+04	-2.09E+05	2.38E+03	-2.09E+05	1.23E+03
	1 000	-3.37E+05	5.56E+04	-4.14E+05	8.72E+03	-4.11E+05	1.53E+04	-4.17E+05	4.84E+03	-4.19E+05	2.66E+02
F_7	30	5.68E-15	2.34E-14	0	0	0	0	0	0	0	0
	200	0	0	0	0	0	0	0	0	0	0
	500	0	0	0	0	0	0	0	0	0	0
	1 000	7.28E-14	3.56E-13	0	0	0	0	0	0	0	0
F_8	30	4.51E-15	2.89E-15	8.88E-16	0	8.88E-16	0	8.88E-16	0	8.88E-16	0
	200	4.37E-15	2.51E-15	8.88E-16	0	8.88E-16	0	8.88E-16	0	8.88E-16	0
	500	4.09E-15	2.68E-15	8.88E-16	0	8.88E-16	0	8.88E-16	0	8.88E-16	0
	1 000	3.94E-15	2.46E-15	8.88E-16	0	8.88E-16	0	8.88E-16	0	8.88E-16	0
F_9	30	2.19E-02	1.14E-02	1.65E-02	8.01E-03	7.06E-05	1.64E-04	2.14E-02	1.24E-02	3.98E-06	7.22E-06
	200	6.74E-02	2.64E-02	4.49E-02	1.70E-02	2.63E-05	6.52E-05	2.37E-02	2.26E-02	4.04E-06	7.20E-06
	500	9.29E-02	4.15E-02	5.02E-02	2.11E-02	2.13E-05	4.55E-05	3.35E-02	2.33E-02	1.63E-05	5.15E-05
	1 000	1.07E-01	4.36E-02	1.02E+01	4.33E+00	3.13E-05	7.14E-05	2.83E-02	2.57E-02	7.80E-06	1.18E-05
F_{10}	30	5.37E-01	2.71E-01	3.24E-01	1.18E-01	1.25E-03	2.21E-03	3.28E-01	1.75E-01	9.71E-05	2.43E-04
	200	6.68E+00	2.77E+00	3.50E+00	1.20E+00	4.19E-03	9.36E-03	2.32E+00	1.77E+00	2.87E-04	5.21E-04
	500	1.70E+01	6.19E+00	4.58E-02	2.04E-02	1.41E-02	2.69E-02	7.97E+00	5.29E+00	1.55E-03	4.13E-03
	1 000	3.94E+01	1.21E+01	2.05E+01	7.82E+00	2.55E-02	6.04E-02	1.23E+01	8.94E+00	1.28E-03	2.06E-03

表 2 3 种改进策略在 10 组基准函数上的寻优数据

函数	WOA		WOA-1		WOA-2		WOA-3		ADS-WOA	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
F_1	3.63E-71	2.24E-70	3.89E-112	2.72E-111	0.00E+00	0.00E+00	1.52E-53	6.03E-53	0.00E+00	0.00E+00
F_2	8.16E-48	5.61E-47	3.24E-67	2.11E-66	0.00E+00	0.00E+00	3.21E-43	1.35E-42	0.00E+00	0.00E+00
F_3	5.12E+06	1.81E+06	4.78E+06	1.62E+06	0.00E+00	0.00E+00	4.76E+06	1.65E+06	0.00E+00	0.00E+00
F_4	1.98E+02	1.51E-01	1.98E+02	1.37E-01	1.97E+02	1.18E-01	5.58E+00	1.13E+01	6.34E-02	1.58E-01
F_5	1.06E+01	2.77E+00	1.55E+01	4.30E+00	4.46E+00	2.03E+00	1.41E-01	6.49E-02	6.52E-04	2.18E-03
F_6	-6.88E+04	1.15E+04	-7.07E+04	1.23E+04	-8.29E+04	2.66E+03	-8.34E+04	1.25E+03	-8.38E+04	2.40E+01
F_7	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	9.09E-15	4.46E-14	0.00E+00	0.00E+00
F_8	4.37E-15	2.51E-15	4.51E-15	2.61E-15	8.88E-16	0.00E+00	3.94E-15	2.75E-15	8.88E-16	0.00E+00
F_9	6.74E-02	2.64E-02	1.27E-01	5.92E-02	2.25E-02	1.22E-02	7.73E-04	3.56E-04	4.04E-06	7.20E-06
F_{10}	6.68E+00	2.77E+00	8.32E+00	2.48E+00	1.73E+00	7.31E-01	1.01E-01	8.08E-02	2.87E-04	5.21E-04

3.4 参数 k 和 μ 对算法性能的影响分析

在本节中,为分析系数 k 和参数 μ 对算法性能的影响,探求合适的参数取值,对 k 或 μ 取值为 0.1, 0.3, 0.5, 0.7, 0.9 的算法在 10 组测试函数的寻优性能进行调查,设置本次迭代次数为 500,种群大小为 30,个体维数为 200,独立运行 50 次,性能指标数据汇总分别如表 3 和表 4 所示.从表 3 和表 4 可以看到: $k = 0.3$ 或 $\mu = 0.9$ 时,算法在更多的目标函数上的性能更好.

表 3 不同 k 值对算法性能的影响

函数	$k = 0.1$		$k = 0.3$		$k = 0.5$		$k = 0.7$		$k = 0.9$	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
F_1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F_2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F_3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F_4	6.34E-02	1.58E-01	4.98E-02	7.44E-02	1.36E-01	2.56E-01	2.37E-01	3.88E-01	2.45E-01	6.60E-01
F_5	6.52E-04	2.18E-03	7.39E-04	1.23E-03	1.65E-03	2.72E-03	3.42E-03	3.94E-03	5.15E-03	8.55E-03
F_6	-8.38E+04	2.40E+01	-8.38E+04	5.22E+01	-8.38E+04	5.80E+00	-8.38E+04	5.93E+00	-8.38E+04	7.37E+00
F_7	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F_8	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00
F_9	4.04E-06	7.20E-06	8.24E-06	1.25E-05	1.39E-05	3.37E-05	1.47E-05	2.48E-05	1.35E-05	1.96E-05
F_{10}	2.87E-04	5.21E-04	2.29E-04	3.79E-04	4.84E-04	7.63E-04	5.18E-04	8.57E-04	7.47E-04	1.33E-03

表 4 不同的 μ 值对算法性能的影响

函数	$\mu = 0.1$		$\mu = 0.3$		$\mu = 0.5$		$\mu = 0.7$		$\mu = 0.9$	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
F_1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F_2	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F_3	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F_4	4.46E-02	7.87E-02	6.32E-02	1.55E-01	2.76E-02	4.13E-02	6.34E-02	1.58E-01	3.90E-02	6.57E-02
F_5	2.54E-04	4.09E-04	3.88E-04	8.03E-04	4.37E-04	9.86E-04	6.52E-04	2.18E-03	1.25E-03	3.12E-03
F_6	-8.38E+04	1.64E+01	-8.38E+04	1.96E+01	-8.38E+04	4.46E+01	-8.38E+04	2.40E+01	-8.38E+04	3.83E+01
F_7	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F_8	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00	8.88E-16	0.00E+00
F_9	5.07E-06	1.37E-05	1.32E-05	3.32E-05	4.19E-06	7.90E-06	4.04E-06	7.20E-06	2.55E-06	5.18E-06
F_{10}	3.56E-04	9.83E-04	2.52E-04	5.98E-04	2.28E-04	4.24E-04	2.87E-04	5.21E-04	1.13E-04	2.45E-04

3.5 ADS-WOA 的时间复杂度分析

设定各算法的最大迭代次数为 T , 种群规模为 N , 维度为 D . WOA 的时间复杂度计算分别为线性距离

控制因子 $O(T)$, 常数螺旋因子 $O(TN)$, 总的时间复杂度为 $O(T + TN)$. ADS-WOA 各种改进策略带来的时间复杂度计算分别为非线性距离控制因子为 $O(T)$, 自适应惯性权重为 $O(T)$, 非线性螺旋因子为 $O(TN)$, 所以总的时间复杂度为 $O(2T+TN)$, 与 WOA 的时间复杂度基本相当. 为了更直观地观察到 ADS-WOA 的时间复杂度, 将 ADS-WOA 与其他算法在 $F_1 \sim F_{10}$ 独立运行 50 次, 计算各算法的平均耗时, 如表 5 所示. 从表 5 可知: ADS-WOA 在大部分函数的运行时间比 WOA, IWOA, GS-WOA 的短, 特别是在 F_5 和 F_6 的运行时间明显缩短, 总的运行时间指标上仅次于 C-A-WWOA, 可见 ADS-WOA 不仅提升了算法收敛性能, 还缩短了一定的运行时间.

表 5 各算法在测试函数上的平均运行时间

函数	WOA	IWOA	C-A-WWOA	GS-WOA	ADS-WOA
F_1	0.157 2	0.185 9	0.163 1	0.171 3	0.151 9
F_2	0.167 2	0.199 7	0.162 2	0.182 5	0.161 6
F_3	1.454 7	1.462 2	1.440 0	1.602 5	1.491 3
F_4	0.195 3	0.217 8	0.158 4	0.189 7	0.174 1
F_5	0.162 2	0.192 2	0.140 9	0.190 0	0.137 2
F_6	0.246 6	0.279 7	0.203 8	0.257 8	0.200 9
F_7	0.199 7	0.240 3	0.164 4	0.191 3	0.179 7
F_8	0.200 6	0.202 8	0.154 7	0.208 8	0.185 3
F_9	1.146 9	1.135 6	1.118 4	1.219 7	1.111 9
F_{10}	1.101 6	1.133 4	1.104 4	1.208 4	1.145 9
Sums	5.032 0	5.249 6	4.810 3	5.422 0	4.939 8

3.6 Wilcoxon 秩和检验

仅通过收敛精度和标准差 2 项指标, 还不够客观地反映一种算法的综合性能, 为了进一步验证 ADS-WOA 性能的显著性, 本文对试验数据进行 Wilcoxon 秩和检验. 本次试验选取各算法在 $F_1 \sim F_{10}$ 的测试结果作为检测数据, 其中各算法的运行设置同 3.3 节, 并与 ADS-WOA 试验数据进行对比, 检测结果如表 6 所示.

设定显著性差异值为 5%. 当 $P < 5\%$ 时, 就认定该算法与 ADS-WOA 在某函数上有显著差异; 当 $P > 5\%$ 时, 就认定该算法与 ADS-WOA 在某函数上无显著差异; $P = 1$ 时, 二者无差异; NaN 表示 ADS-WOA 算法与自身比较或与其他算法比较无差异. 从表 6 可以观察到 P 值大都小于 0.05, 证明了 ADS-WOA 的有效性.

表 6 Wilcoxon 秩和检验结果

函数	WOA	IWOA	GS-WOA	C-A-WWOA
F_1	3.31E-20	NaN	NaN	NaN
F_2	3.31E-20	3.31E-20	3.31E-20	NaN
F_3	3.31E-20	NaN	NaN	NaN
F_4	7.07E-18	7.07E-18	6.62E-02	7.07E-18
F_5	7.07E-18	7.07E-18	1.21E-08	7.07E-18
F_6	3.53E-17	5.20E-03	9.93E-14	2.42E-05
F_7	NaN	NaN	NaN	NaN
F_8	9.64E-14	NaN	NaN	NaN
F_9	7.07E-18	7.07E-18	8.00E-03	7.07E-18
F_{10}	7.07E-18	7.07E-18	1.81E-06	8.46E-18

4 结论

- 1) ADS-WOA 在求解高维问题有较好的性能.
- 2) ADS-WOA 在收敛性能、鲁棒性和算法运行速度等方面有显著的提升.
- 3) ADS-WOA 的实际应用效果还有待考证. 下一步的研究重点是将 ADS-WOA 应用于实际的约束工程

问题,并进一步地改进算法的效果.

参考文献:

- [1] 王俊深, 高贵兵, 张红波. 基于复合式遗传算法的电机排产优化[J]. 湖南科技大学学报(自然科学版), 2022, 37(2): 79-87.
- [2] 李鹏, 何宸宇, 刘琪, 等. 基于A*扩展自适应蚁群算法的移动机器人路径规划[J]. 湖南科技大学学报(自然科学版), 2021, 36(2): 85-92.
- [3] 蒋丽, 叶润舟, 梁昌勇, 等. 改进的二阶振荡粒子群算法[J]. 计算机工程与应用, 2019, 55(9): 130-138.
- [4] 陈晟宗, 张纪会, 于守水, 等. 求解旅行商问题的波动温控模拟退火算法[J]. 控制与决策, 2023, 38(4): 911-920.
- [5] MIRJALILI S, LEWIS A. The whale optimization algorithm[J]. Advances in Engineering Software, 2016, 95: 51-67.
- [6] MIRJALILI S, GANDOMI A H, MIRJALILI S Z, et al. Salp Swarm Algorithm: a bio-inspired optimizer for engineering design problems[J]. Advances in Engineering Software, 2017, 114: 163-191.
- [7] MIRJALILI S, MIRJALILI S M, LEWIS A. Grey wolf optimizer[J]. Advances in Engineering Software, 2014, 69: 46-61.
- [8] ARORA S, SINGH S. Butterfly optimization algorithm: a novel approach for global optimization[J]. Soft Computing, 2019, 23(3): 715-734.
- [9] OULADBRAHIM A, BELAIDI I, KHATIR S, et al. Experimental crack identification of API X70 steel pipeline using improved artificial neural networks based on whale optimization algorithm[J]. Mechanics of Materials, 2022, 166: 104200.
- [10] DING Y F, CHEN Z J, ZHANG H W, et al. A short-term wind power prediction model based on CEEMD and WOA-KELM[J]. Renewable Energy, 2022, 189: 188-198.
- [11] EL-FERGANY A A, HASANIEN H M, AGWA A M. Semi-empirical PEM fuel cells model using whale optimization algorithm[J]. Energy Conversion and Management, 2019, 201: 112197.
- [12] 刘磊, 白克强, 但志宏, 等. 一种全局搜索策略的鲸鱼优化算法[J]. 小型微型计算机系统, 2020, 41(9): 1820-1825.
- [13] 程浩淼, 王梦磊, 汪靓, 等. 耦合中心游移和双权重因子的鲸鱼算法及应用[J]. 计算机工程与应用, 2022, 58(13): 74-84.
- [14] 耿召里, 李目, 曹淑睿, 等. 基于混合反向学习策略的鲸鱼优化算法[J]. 计算机工程与科学, 2022, 44(2): 355-363.
- [15] 赵传武, 黄宝柱, 阎跃观, 等. 一种非线性权重的自适应鲸鱼优化算法[J]. 计算机技术与发展, 2020, 30(10): 7-13.
- [16] 武泽权, 牟永敏. 一种改进的鲸鱼优化算法[J]. 计算机应用研究, 2020, 37(12): 3618-3621.
- [17] 黄辉先, 张广炎, 陈思溢, 等. 基于混沌权重和精英引导的鲸鱼优化算法[J]. 传感器与微系统, 2020, 39(5): 113-116.
- [18] 孔芝, 杨青峰, 赵杰, 等. 基于自适应调整权重和搜索策略的鲸鱼优化算法[J]. 东北大学学报(自然科学版), 2020, 41(1): 35-43.
- [19] 张潮, 冯锋. 混沌策略和单纯形法改进的鲸鱼优化算法[J]. 中国科技论文, 2020, 15(3): 293-299.
- [20] 罗钧, 庞亚男, 刘建强. 基于历史认知的鲸鱼算法求解动态能耗[J]. 电子测量与仪器学报, 2022, 36(1): 236-245.
- [21] 徐航, 张达敏, 王依柔, 等. 基于高斯映射和小孔成像学习策略的鲸鱼优化算法[J]. 计算机应用研究, 2020, 37(11): 3271-3275.