

DOI: 10.13382/j.jemi.B2407401

融合邻域搜索的自适应鲸鱼优化算法*

谢良波 韩 伸 张钰坤

(重庆邮电大学通信与信息工程学院 重庆 400065)

摘要: 鲸鱼优化算法(WOA)是一种高效的群体智能优化算法。与其他智能优化算法相比,WOA由于结构简单,参数少以及强大的优化能力已经被广泛使用。然而,传统的WOA存在收敛速度慢,容易陷入局部最优的问题。为了解决这些问题,本研究提出了一种改进的鲸鱼优化算法(IWOA),该算法采用自适应更新机制,受粒子群算法的启发,在优化过程中引入个体历史最优位置,并通过自适应策略动态调整全局历史最优位置和个体历史最优位置的权重,避免算法陷入局部最优;同时通过邻域搜索策略,在迭代后期围绕全局历史最优位置进行邻域更新,提升算法寻优能力。选取16个典型的基准测试函数以及CEC2014测试集的8个复合函数进行仿真实验,与其他传统及改进的群体智能优化算法相比,IWOA的收敛精度和收敛速度更有优势,验证了IWOA的有效性;并将IWOA应用在焊接梁和压力容器设计2个工程设计问题上,相比于WOA,经济成本分别节约了3.94%、5.58%,验证了算法的有效性。

关键词: 智能优化算法;鲸鱼优化算法;粒子群优化算法;自适应更新机制;邻域搜索策略

中图分类号: TP301.6;TN929.5 **文献标识码:** A **国家标准学科分类代码:** 520.10

Adaptive whale optimization algorithm combining neighborhood search

Xie Liangbo Han Shen Zhang Yukun

(School of Communications and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065)

Abstract: The whale optimization algorithm (WOA) is a highly competitive and efficient swarm intelligence optimization algorithm. In comparison to other intelligent optimization algorithms, WOA offers a simple structure, fewer parameters, and robust optimization capabilities. However, the conventional WOA exhibits slow convergence and falls into local optima easily. To address these issues, this paper proposes an improved whale optimization algorithm (IWOA). The algorithm adopts an adaptive update mechanism, inspired by particle swarm optimization, incorporating the individual's historical best position during the optimization process, and dynamically adjusts the weights of the global best and individual best positions through an adaptive strategy to avoid getting trapped in local optima; at the same time, through neighborhood search strategy, neighborhood updates are carried out around the global historical optimal position in the later stage of iteration to improve the algorithm's optimization ability. 16 typical benchmark test functions and 8 composite functions from the CEC2014 test set are selected for simulation experiments, compared to other traditional and improved swarm intelligence optimization algorithms, IWOA demonstrates superior convergence accuracy and speed, validating its effectiveness; and IWOA is applied to two engineering design problems, welding beam and pressure vessel design, compared with WOA, the economic cost is saved by 3.94% and 5.58%, respectively, verifying the effectiveness of the algorithm.

Keywords: intelligent optimization algorithm; whale optimization algorithm; particle swarm optimization; adaptive update mechanism; neighborhood search strategy

收稿日期: 2024-04-03 Received Date: 2024-04-03

* 基金项目: 重庆市研究生科研创新项目(CYS23453, CYS22473)、重庆市自然科学基金面上项目(CSTB2023NSCQ-MSX0249, CSTB2023NSCQ-MSX0832)、重庆市教委科学技术研究项目(KJQN202300615)资助

0 引言

近年来,优化问题在众多科学和工业学科中引起了极大的关注,成为研究工作的焦点。优化领域中遇到的许多问题都表现出独特的性质,包括高维度、非线性和存在多个局部最优,使得优化问题的复杂性大大提高。为了解决这些问题,从自然现象中汲取灵感并表现出智能行为的启发式算法被人们发掘,并已证明是解决这些领域中常见复杂优化问题的有效方法。常见的启发式算法包括粒子群优化算法 (particle swarm optimization, PSO)^[1]、灰狼优化算法 (grey wolf optimization, GWO)^[2]、飞蛾扑火优化算法 (moth-flame optimization, MFO)^[3]、蚁群优化算法 (ant colony optimization, ACO)^[4]、蜣螂优化算法 (dung beetle optimization, DBO)^[5]、天鹰优化算法 (aquila optimization, AO)^[6]。

在众多启发式算法中, Mirjalili 在 2016 年提出的鲸鱼优化算法 (whale optimization algorithm, WOA) 在广泛的优化问题中显示出巨大的潜力^[7]。由于求解精度高、时间复杂度低的优点, WOA 已广泛应用于图像处理^[8]、室内定位^[9]、机器人路径规划^[10]。然而,与其他优化算法一样, WOA 也存在易陷入局部最优和收敛速度慢的问题。为了解决这些问题,研究人员提出了许多改进方法。文献[11]提出了一种集成多种策略的改进 WOA。首先,在种群初始化过程中引入混沌映射,使粒子分布更加均匀;其次,通过引入自适应权重调整策略平衡算法的搜索和开发能力;最后,为了避免种群陷入局部最优,引入莱维飞行策略,增加了算法的多样性。文献[12]通过参数的指数更新机制平衡算法的搜索和开发阶段,并通过对立学习以及柯西变异进一步提高了种群的多样性,防止算法陷入局部最优。文献[13]采用非线性收敛因子策略,降低参数前期减小速率,从而提高了算法的搜索和开发能力,然后,引入柯西变异对个体位置进行随机干扰,避免算法陷入局部最优。此外,有很多学者提出了混合算法的方法。文献[14]通过自适应权重策略调整 WOA 的收敛速度,引入模拟退火增强算法的全局寻优能力。文献[15]将 WOA 与 PSO 相结合。由于 WOA 的强大开发能力和 PSO 的强大搜索能力,两者的结合可以更好地提高算法的整体性能。文献[16]通过正弦映射初始化种群,使得种群分布更加均匀。为了提高算法的收敛速度,融合差分进化算法 (differential evolution, DE) 的思想,加入了变异和交叉过程。

上述方法提升了算法的寻优精度,但并未解决 WOA 易陷入局部最优和收敛速度慢的固有问题,同时多策略的融合以及混合算法通常会增加算法的复杂度。本研究引入自适应更新机制和邻域搜索策略,提出了一种改进

的鲸鱼优化算法 (improved whale optimization algorithm, IWOA)。自适应更新机制通过将个体历史最优位置引入到搜索过程中,并通过自适应参数动态调整全局历史最优位置和个体历史最优位置的权重,增强了算法前期的全局搜索能力。在优化过程的后续阶段,采用邻域搜索策略进一步增强算法的局部优化能力。

1 鲸鱼优化算法

传统 WOA 是受鲸鱼捕食的自然生物学行为启发而提出的一种优化算法。在算法寻优过程中,包括以下 3 种方式:包围猎物、气泡网捕食和搜索猎物。

1.1 包围猎物

鲸鱼在狩猎过程中,不断地接近猎物。在 WOA 中,鲸鱼的位置对应于空间中的一个解,食物的位置代表最优解。因此,鲸鱼的狩猎行为可以近似为算法向最优收敛的过程。当 WOA 使用包围猎物的方式进行捕食时,位置 $X(t+1)$ 的更新公式如式(1)和(2)所示。

$$X(t+1) = X^{gb}(t) - A \times D \quad (1)$$

$$D = |C \times X^{gb}(t) - X(t)| \quad (2)$$

式中: t 表示迭代次数; $X(t)$ 表示当前鲸鱼位置; $X^{gb}(t)$ 代表当前最佳鲸鱼位置。 A 和 C 是系数,如式(3)~(5)所示。

$$a = 2 - 2 \times t/T \quad (3)$$

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

$$A = 2 \times a \times r - a \quad (5)$$

式中: r 是 $[0, 1]$ 的随机数; a 的取值范围是 $[0, 2]$, 随着迭代次数的增加线性减小; A 的取值范围为 $[-a, a]$; T 表示最大迭代次数。

1.2 气泡网捕食

当采用气泡网捕食的方式接近猎物时, WOA 的寻优过程可以表示为:

$$X(t+1) = D' \times e^{bl} \times \cos(2\pi l) + X^{gb}(t) \quad (6)$$

$$D' = |X^{gb}(t) - X(t)| \quad (7)$$

式中: b 是对数螺旋常数; l 是 $[-1, 1]$ 范围内的随机数。

A 的值根据参数 a 的变化而变化。当 $|A| < 1$ 时, WOA 以 50% 概率选择包围猎物或气泡网捕食的方式进行寻优,可以表示为:

$$X(t+1) = \begin{cases} X^{gb}(t) - A \times D, & p < 0.5 \\ D' \times e^{bl} \times \cos(2\pi l) + X^{gb}(t), & p \geq 0.5 \end{cases} \quad (8)$$

式中: p 为 $[0, 1]$ 上的随机数。

1.3 搜索猎物

为了降低算法陷入局部最优的风险,当 $|A| > 1$ 时,在更新策略中引入了随机个体位置。可以表示为:

$$X(t+1) = X_{rand} - A \times D' \quad (9)$$

$$D'' = |C \times X_{rand} - X(t)| \quad (10)$$

式中: X_{rand} 为当前种群中随机个体的位置。

传统的鲸鱼优化算法通过包围猎物和气泡网捕食两种更新方式,可在算法迭代前期向最优解快速收敛,然而只围绕全局历史最优解进行更新导致算法存在前期易陷入局部最优的不足。搜索猎物的方式引入随机个体进行迭代更新,由于提高了算法后期搜索的随机性,会导致算法收敛速度降低。

2 改进的鲸鱼优化算法

由于全局搜索能力不足,导致传统的 WOA 容易陷入局部最优。为了提高算法的整体性能,本研究引入了两种改进策略:自适应更新机制和邻域搜索策略。这些策略不仅提高了算法前期的全局搜索能力,而且增强了算法后期的局部寻优能力。

2.1 自适应更新机制

传统的 WOA 采用式(8)进行寻优时存在一定的局限性,因为它只考虑当前个体的位置和全局历史最优位置,而忽略了个体历史最优位置,会导致算法的全局搜索能力不足,使其容易陷入局部最优。当采用式(9)进行更新时,随机位置的引入会导致寻优过程中盲目性过高,因此会降低算法收敛速度。受粒子群算法的启发^[17],为了提高算法全局搜索能力,将个体历史最优位置引入位置更新方程中,同时通过自适应参数动态调整全局以及个体历史最优位置的权重,可以更好地平衡算法的全局搜索能力以及局部寻优能力。

当 $|A| > 1$ 时,位置更新方程表示为:

$$X(t+1) = \frac{1}{2} \times a \times X^{pb}(t) + \left(1 - \frac{1}{2} \times a\right) \times$$

$$X^{gb}(t) - A \times D_1 \quad (11)$$

$$D_1 = C_1 \times X^{pb}(t) - X(t) \quad (12)$$

$$C_1 = (2 \times r - 1) \times a \quad (13)$$

式中: $X^{pb}(t)$ 表示个体历史最佳位置。

当 $q < |A| < 1$ 时,其中 q 是常数,位置更新方程如式(14)~(16)所示。

$$X(t+1) = \frac{1}{2} \times a \times X^{pb}(t) + \left(1 - \frac{1}{2} \times a\right) \times$$

$$X^{gb}(t) - A \times D_2 \quad (14)$$

$$D_2 = C_2 \times X^{gb}(t) - X(t) \quad (15)$$

$$C_2 = (2 \times r - 1) \times (2 - a) \quad (16)$$

由式(3)可知随着迭代次数的增加,参数 a 线性减小,同时算法位置更新方程由式(11)变为(14)。在算法的早期阶段,采用式(11)进行位置更新, a 的值相对较大,个体历史最优位置的权重较大。在算法的后期阶段,采用式(14)进行位置更新, a 的值相对较小,全局历史最

优位置的权重较大。自适应更新机制通过引入个体历史最优位置可以提高算法的全局搜索能力。同时参数 a 、 C_1 和 C_2 的自适应变化可以动态调整全局历史最优位置和个体历史最优位置的权重,可以有效地平衡算法的全局搜索和局部寻优能力,从而提高算法的整体性能。

2.2 邻域搜索策略

在算法迭代的后期阶段,种群中的个体大部分处于一个相对较优的位置,为了提高算法的局部寻优能力,以全局历史最优位置为中心,在其邻域内进行寻优,寻优步长的合理设置有利于提高算法的整体性能。本研究将步长设置为自适应步长,随着迭代次数的增加而线性减小。

当 $|A| < q$ 时,更新表达式如式(17)和(18)所示。

$$X(t+1) = X^{gb}(t) + n \quad (17)$$

$$n = (2 \times r - 1) \times a \times n \quad (18)$$

式中:参数 n 表示步长,初始值设置为种群位置范围的上界。

更新方程表明,通过调整 q 的值,可以控制种群何时进入邻域搜索阶段,本研究将 q 值设置为 0.15。此外,步长 n 随着迭代次数的增加而减少。在邻域搜索的初始阶段, n 值较大时,可以帮助算法跳出局部最优。随着迭代次数的增加,算法进入迭代的大后期, n 值越来越小,加强了算法的局部寻优。

2.3 IWOA 算法总结

如图 1 所示,IWOA 的更新过程可以分为 3 个阶段,根据 a 的大小不同选择不同的更新策略。当 $|a| \geq 1$ 时,算法进入全局搜索阶段,此时个体历史最优位置占主导地位,算法的全局搜索能力较强。当 $q < |a| < 1$ 时,算法进入局部寻优阶段,此时全局历史最优位置占主导地位,算法的局部寻优能力较强。当 $|a| \leq q$ 时,算法进入增强的局部寻优阶段,以全局历史最优位置为中心进行邻域更新,进一步增强了算法的局部寻优能力。在算法迭代前期,自适应更新机制通过调整全局历史最优位置和个

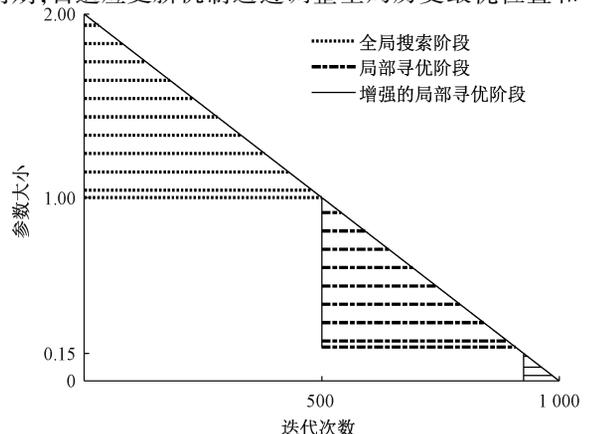


图 1 参数 a 的变化

Fig. 1 Changes of parameter a

体历史最优位置的权重大小,平衡算法局部寻优和全局搜索的能力,可以有效避免算法陷入局部最优。邻域搜索策略可以增强算法后期的局部寻优能力,有效地提高算法的收敛速度。

算法 1 概述了 IWOA 的伪代码,步骤如下:

算法 1 IWOA 算法

```

参数初始化: 种群大小  $N$ , 以及参数  $T, n, q$ , 个体初始位置  $X$ 
(i) ( $i=1, 2, \dots, N$ )
While ( $t < T$ )
    评估适应度值大小,更新个体最优位置以及全局最优位置。
    更新参数  $a, C_1, C_2, n$ 。
    For  $i=1:N$ 
        If  $|A| > 1$ 
            使用式 (11) 更新种群位置
        Else if  $q < |A| < 1$ 
            使用方程式 (14) 更新种群位置
        Else if  $|A| < q$ 
            使用式 (17) 更新种群位置
    End if
End for
 $t=t+1$ 
End while
输出最优值

```

2.4 算法时间复杂度分析

时间复杂度是评价算法性能的一个重要指标,假设种群数量为 N ,迭代次数为 T ,搜索空间维度为 Dim ,可得 WOA 的时间复杂度为 $O(N * T * Dim)$,IWOA 引入了自适应更新机制以及邻域搜索策略,初始化的时间复杂度为 $O(N * Dim)$,自适应更新以及邻域搜索过程,每个步骤的时间复杂度为 $O(N * T * Dim)$,IWOA 并未增加循环嵌套的过程,因此 IWOA 整体的时间复杂度为 $O(N * T * Dim)$,与 WOA 算法的时间复杂度基本一致。

3 仿真结果与分析

3.1 基准测试函数与测试环境

为了验证本研究提出的 IWOA 的性能,共选择了 16 个常用的基准测试函数进行算法性能测试,包括单峰测试函数 $f1 \sim f7$ 、多峰测试函数 $f8 \sim f13$ 和固定维多峰测试函数 $f14 \sim f16$ ^[18]。测试函数如表 1 所示。

测试环境和参数设置:模拟和验证是在 64 位 Microsoft 操作系统上使用 MATLAB R2018b 平台进行的。将种群大小设置为 30,迭代次数设置为 1 000,并重复实验 30 次,以获得最优值,平均值和标准差。

3.2 实验结果与分析

通过对各种基准函数进行测试来评估 IWOA 的性

表 1 基准测试函数

Table 1 Benchmarking function

序号	基准测试函数	维度	范围	理论最优值
$f1$	Sphere	Dim	$[-100, 100]$	0
$f2$	Schwefel 2. 22	Dim	$[-10, 10]$	0
$f3$	Schwefel 1. 2	Dim	$[-100, 100]$	0
$f4$	Schwefel 2. 21	Dim	$[-100, 100]$	0
$f5$	Rosenbrock	Dim	$[-30, 30]$	0
$f6$	Step	Dim	$[-100, 100]$	0
$f7$	Quartic	Dim	$[-1. 28, 1. 28]$	0
$f8$	Schwefel 2. 26	Dim	$[-500, 500]$	$-418. 9 \times Dim$
$f9$	Rastrigin	Dim	$[-5. 12, 5. 12]$	0
$f10$	Ackley	Dim	$[-32, 32]$	0
$f11$	Griewank	Dim	$[-600, 600]$	0
$f12$	Penalized 1	Dim	$[-50, 50]$	0
$f13$	Penalized 2	Dim	$[-50, 50]$	0
$f14$	Shekel's Foxholes	2	$[-65. 53, 65. 53]$	1
$f15$	Shekel's Family1	4	$[0, 10]$	-10
$f16$	Shekel's Family2	4	$[0, 10]$	-10

能。为了验证其有效性,将 IWOA 与其他几种经典的优化算法进行了比较,包括 PSO、DE、MFO、GWO、WOA。测试结果如表 2 所示。

表 2 显示了 16 个基准函数的测试结果,其中单峰和多峰函数的维度设置为 30,对最优的实验结果进行加粗。计算结果表明,IWOA 在函数 $f1 \sim f4$ 、 $f9$ 和 $f11$ 收敛到理论最优值。在函数 $f7$ 、 $f10$ 、 $f12$ 和 $f13$ 上,IWOA 虽然没有收敛到理论最优值,但是结果始终优于其他算法。对于函数 $f5$,WOA 具有最低的标准差,对于函数 $f6$,WOA 具有最低的平均值和标准差,对于函数 $f8$,DE 具有最低标准差。对于函数 $f14$,DE 的性能表现最好,对于函数 $f15$ 、 $f16$,GWO 的性能表现最好。由 Friedman 检验结果可知,IWOA 的秩均值最低为 1. 56。

为了验证 IWOA 在解决高维问题中的性能,将单峰和多峰函数 $f1 \sim f13$ 维数分别设置为 100 和 200。IWOA 除在 $f8$ 的平均值以及 $f5$ 、 $f12$ 、 $f13$ 的标准差上没有达到最优值外,在其他函数上均收敛到最优值。在运行时间消耗上,除 $f3$ 之外,IWOA 的运行时间消耗最少。因此可知即使在更高的维度上,IWOA 的性能也能保持稳定。根据 Friedman 检验,在 100 维和 200 维空间中,IWOA 的秩均值最低,分别为 1. 23 和 1. 15。

为了更直观地体现 IWOA 的收敛性能,选取 9 个具有代表性的测试函数,图 2 给出收敛曲线图,纵轴使用对数尺度。由图 2(a)~(g)可以观察到,IWOA 在单峰函数上表现出强大的局部开发能力,在 $f1 \sim f7$ 上具有最高的收敛精度,并且在收敛速度方面也表现出显著的优势。在多峰函数上,由图 2(h)和(i)可得 IWOA 在函数 $f10$ 、 $f12$ 上实现了最高的收敛精度,收敛速度较快。

表 2 各算法测试结果对比 (30 维)

Table 2 Comparison of test results for various algorithms (30 Dimensions)

函数	准则	PSO	DE	MFO	GWO	WOA	IWOA
f_1	最优值	4.63×10^1	4.02×10^4	5.07×10^{-6}	6.01×10^{-62}	4.29×10^{-164}	0.00×10^0
	平均值	7.57×10^1	6.05×10^4	6.67×10^2	3.35×10^{-59}	1.97×10^{-151}	0.00×10^0
	标准差	1.15×10^1	7.04×10^3	2.54×10^3	5.31×10^{-59}	7.28×10^{-151}	0.00×10^0
	时间	0.076 5	0.720 5	0.107 1	0.123 5	0.048 9	0.047 1
f_2	最优值	3.36×10^1	9.71×10^1	1.40×10^{-4}	9.60×10^{-36}	2.01×10^{-104}	0.00×10^0
	平均值	4.81×10^1	1.14×10^2	2.90×10^1	8.20×10^{-35}	5.07×10^{-96}	0.00×10^0
	标准差	1.03×10^1	7.66×10^0	2.29×10^1	9.46×10^{-35}	2.51×10^{-95}	0.00×10^0
	时间	0.074 6	0.731 5	0.108 3	0.125 6	0.050 6	0.047 8
f_3	最优值	1.92×10^2	5.35×10^4	4.98×10^2	7.27×10^{-21}	3.93×10^3	0.00×10^0
	平均值	2.69×10^2	7.26×10^4	1.90×10^4	1.15×10^{-15}	1.94×10^4	0.00×10^0
	标准差	5.78×10^1	9.56×10^3	1.31×10^4	2.47×10^{-15}	9.14×10^3	0.00×10^0
	时间	0.235 0	0.939 4	0.264 9	0.282 0	0.207 0	0.213 9
f_4	最优值	2.95×10^0	8.00×10^1	5.53×10^1	4.23×10^{-16}	1.83×10^{-1}	0.00×10^0
	平均值	3.54×10^0	8.71×10^1	7.00×10^1	1.27×10^{-14}	3.91×10^1	0.00×10^0
	标准差	2.44×10^{-1}	3.57×10^0	8.71×10^0	1.64×10^{-14}	3.06×10^1	0.00×10^0
	时间	0.078 2	0.775 5	0.108 1	0.124 4	0.049 6	0.046 7
f_5	最优值	2.69×10^4	1.57×10^8	9.69×10^0	2.54×10^1	2.64×10^1	5.57×10^{-5}
	平均值	5.34×10^4	2.22×10^8	2.67×10^6	2.68×10^1	2.72×10^1	7.29×10^0
	标准差	1.38×10^4	2.89×10^7	1.46×10^7	6.82×10^{-1}	5.07×10^{-1}	1.14×10^1
	时间	0.097 1	0.736 0	0.127 9	0.143 8	0.069 3	0.063 3
f_6	最优值	5.07×10^1	5.70×10^4	1.62×10^{-5}	1.38×10^{-5}	7.06×10^{-3}	5.60×10^{-8}
	平均值	7.21×10^1	6.18×10^4	2.99×10^3	6.80×10^{-1}	8.07×10^{-2}	1.08×10^{-1}
	标准差	9.42×10^0	3.74×10^3	6.51×10^3	3.54×10^{-1}	1.10×10^{-1}	2.93×10^{-1}
	时间	0.077 0	0.747 8	0.108 0	0.123 6	0.049 0	0.042 9
f_7	最优值	4.02×10^1	5.57×10^1	2.68×10^{-2}	3.42×10^{-4}	5.28×10^{-6}	4.70×10^{-6}
	平均值	7.47×10^1	9.66×10^1	4.13×10^0	7.65×10^{-4}	1.29×10^{-3}	4.70×10^{-5}
	标准差	1.75×10^1	2.17×10^1	6.11×10^0	3.50×10^{-4}	1.20×10^{-3}	3.89×10^{-5}
	时间	0.186 7	0.841 8	0.219 8	0.233 6	0.158 9	0.153 8
f_8	最优值	-8.12×10^3	-5.09×10^3	-9.52×10^3	-7.56×10^3	-1.26×10^4	-1.26×10^4
	平均值	-6.65×10^3	-4.22×10^3	-8.31×10^3	-6.13×10^3	-1.15×10^4	-1.15×10^4
	标准差	8.79×10^2	3.34×10^2	8.79×10^2	8.23×10^2	1.42×10^3	2.17×10^3
	时间	0.108 0	0.749 2	0.130 5	0.151 1	0.072 2	0.064 3
f_9	最优值	2.82×10^2	3.40×10^2	8.26×10^1	0.00×10^0	0.00×10^0	0.00×10^0
	平均值	3.30×10^2	3.95×10^2	1.55×10^2	6.69×10^{-1}	0.00×10^0	0.00×10^0
	标准差	2.15×10^1	1.78×10^1	3.73×10^1	1.83×10^0	0.00×10^0	0.00×10^0
	时间	0.097 2	0.790 7	0.122 5	0.130 1	0.055 0	0.047 9
f_{10}	最优值	6.59×10^0	2.01×10^1	1.66×10^0	7.99×10^{-15}	8.88×10^{-16}	8.88×10^{-16}
	平均值	7.06×10^0	2.03×10^1	1.62×10^1	1.49×10^{-14}	3.85×10^{-15}	8.88×10^{-16}
	标准差	3.26×10^{-1}	7.18×10^{-2}	5.96×10^0	2.46×10^{-15}	2.10×10^{-15}	0.00×10^0
	时间	0.099 8	0.785 5	0.127 3	0.131 3	0.057 5	0.049 7
f_{11}	最优值	9.69×10^{-1}	3.24×10^2	2.79×10^{-5}	0.00×10^0	0.00×10^0	0.00×10^0
	平均值	9.96×10^1	5.59×10^2	2.42×10^1	1.28×10^{-3}	4.76×10^{-3}	0.00×10^0
	标准差	1.32×10^{-2}	5.64×10^1	4.69×10^1	4.18×10^{-3}	1.84×10^{-2}	0.00×10^0
	时间	0.110 9	0.756 4	0.139 8	0.148 5	0.072 7	0.065 2
f_{12}	最优值	1.92×10^0	2.56×10^8	1.54×10^{-4}	2.89×10^{-6}	4.64×10^{-4}	4.27×10^{-7}
	平均值	2.90×10^0	5.09×10^8	8.53×10^6	3.50×10^{-2}	7.09×10^{-3}	5.36×10^{-3}
	标准差	5.19×10^{-1}	1.08×10^8	4.67×10^7	1.78×10^{-2}	1.31×10^{-2}	1.09×10^{-2}
	时间	0.374 6	1.062 9	0.401 7	0.414 3	0.339 2	0.332 0
f_{13}	最优值	1.01×10^1	6.53×10^8	6.78×10^{-5}	1.01×10^{-1}	2.08×10^{-2}	1.92×10^{-9}
	平均值	1.24×10^1	1.02×10^9	1.69×10^0	5.02×10^{-1}	1.92×10^{-1}	3.61×10^{-3}
	标准差	1.49×10^0	1.90×10^8	4.77×10^0	1.92×10^{-1}	1.70×10^1	6.59×10^{-3}
	时间	0.377 7	1.009 8	0.407 2	0.418 0	0.342 2	0.334 2

续表

函数	准则	PSO	DE	MFO	GWO	WOA	IWOA
f14	最优值	9.98×10^{-1}	9.98×10^{-1}	9.98×10^1	9.98×10^{-1}	9.98×10^{-1}	9.98×10^{-1}
	平均值	3.17×10^0	1.13×10^0	2.61×10^0	3.42×10^0	2.50×10^0	2.21×10^0
	标准差	2.47×10^0	3.02×10^{-1}	2.22×10^0	3.65×10^0	2.92×10^0	2.51×10^0
	时间	0.550 5	1.199 8	0.557 6	0.556 8	0.550 0	0.548 4
f15	最优值	-1.03×10^1	-7.56×10^0	-1.04×10^1	-1.04×10^1	-1.04×10^1	-1.04×10^1
	平均值	-7.98×10^0	-3.21×10^0	-7.93×10^0	-1.04×10^1	-8.69×10^0	-8.84×10^0
	标准差	2.03×10^0	1.47×10^0	3.36×10^0	2.49×10^{-4}	2.64×10^0	2.29×10^0
	时间	0.076 7	0.706 2	0.085 3	0.083 3	0.076 2	0.073 5
f16	最优值	-1.05×10^1	-6.49×10^0	-1.05×10^1	-1.05×10^1	-1.05×10^1	-1.05×10^1
	平均值	-7.96×10^0	-3.48×10^0	-7.26×10^0	-1.05×10^1	-8.72×10^0	-9.33×10^0
	标准差	1.63×10^0	1.13×10^0	3.42×10^0	1.98×10^{-4}	3.01×10^0	2.27×10^0
	时间	0.097 4	0.754 7	0.104 8	0.104 4	0.095 9	0.091 6
Friedman 秩均值		4.53	5.84	3.44	2.81	2.81	1.56

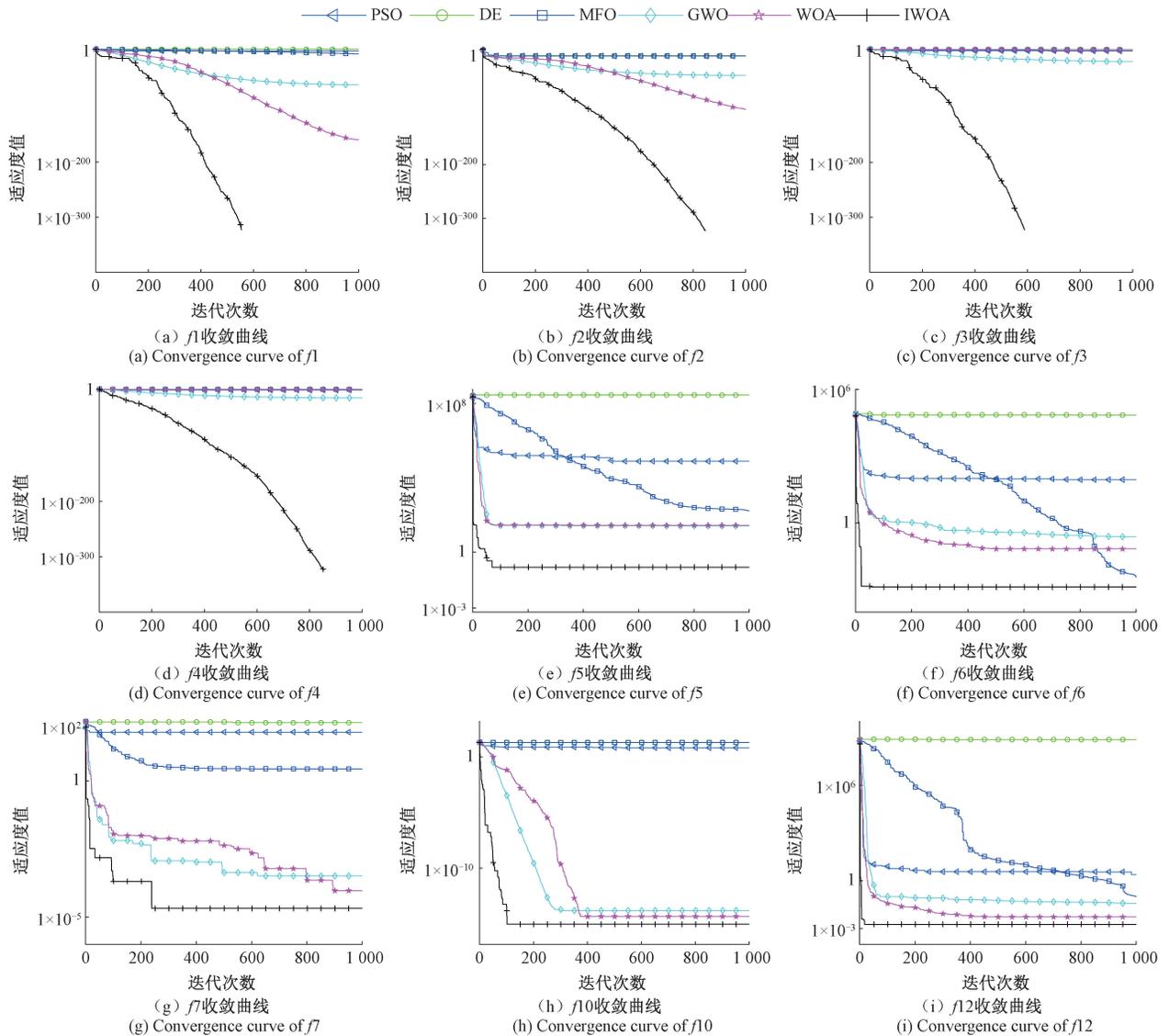


图 2 函数收敛曲线

Fig. 2 Function convergence curve

3.3 策略有效性评估

为了评估不同策略对 WOA 性能提升的影响,将本研究提出的两种策略分别集成到 WOA 中,表示为 WOA-1 和 WOA-2。WOA-1 集成自适应更新机制,WOA-2 集成邻域搜索策略。在本次实验中,将测试函数的维度设置为 30,所有其他参数保持不变。分析得知,对于函数 f_1 、 f_2 ,与 WOA 相比,WOA-1 和 WOA-2 的性能更好。对于函数 f_3 ,与 WOA 相比,WOA-1 性能提高,而 WOA-2 的性能下降。对于函数 f_4 ,与 WOA 相比,WOA-1 性能提高,WOA-2 与 WOA 的性能相近。对于函数 f_5 、 f_7 和 $f_9 \sim f_{11}$,WOA,WOA-2 和 WOA-1 的性能是相近。然而,对于函数 f_6 、 f_{12} ,与 WOA 相比,WOA-1 和 WOA-2 的性能下降。对于函数 f_8 ,WOA-1 的性能下降,而 WOA-2 性能提高。对于函数 f_{13} ,WOA-1 性能下降,WOA-2 性能相近。对于函数 $f_{14} \sim f_{16}$,WOA-2 的性能与 WOA 的性能相近,WOA-1 性能提高。总之,WOA-1 的性能总体优于 WOA,WOA-2 的性能总体与 WOA 相近。当这两种策略都集成到算法时,算法的性能得到了更为显著的提高,证明了策略的有效性。

3.4 Wilcoxon 秩和检验

为了更好地评估 IWOA 与其他算法之间的性能差异,本研究采用非参数检验的统计分析方法 Wilcoxon 秩和检验,对算法性能进行评估,秩和检验是一种衡量两种算法之间差异度大小的统计方法。在显著性水平为 5% 下,当 $P < 5\%$,说明两种算法之间存在显著性差异,在 16 个基准测试函数上,IWOA 与 PSO 在函数 f_{14} 、 f_{16} 上的 P 值大于 5%,IWOA 与 MFO 在函数 f_6 、 f_{14} 、 f_{16} 上的 P 值大于 5%,IWOA 与 WOA 在函数 f_9 、 f_{11} 、 f_{14} 、 f_{15} 上的 P 值大于 5%,IWOA 与 DE、GWO 在 16 个函数上的 P 值均小于 5%,上述结果表明 IWOA 与其他 5 种算法之间性能存在显著性差异,证明算法改进的有效性。

3.5 IWOA 与其他改进算法的性能对比

为了进一步验证算法的有效性,本研究选取了一些较新的改进算法,包括增强型灰狼优化算法^[19](enhanced grey wolf optimization, EGWO)、A-C 参数鲸鱼优化算法(A-C parametric whale optimization algorithm, AWOA)^[20]、增强型鲸鱼优化算法^[21](enhanced whale optimization algorithm, EWOA)、收敛和利用增强鲸鱼优化算法^[12](convergence and exploitability enhancement whale optimization algorithm, MWOA-CEE)、基于混沌映射和自适应迭代策略的鲸鱼优化算法^[22](whale optimization algorithm with chaotic mapping and adaptive iteration strategy, CMAIS-WOA)。选取了 13 个基准测试函数进行对比分析,迭代次数设置为 500,其他参数保持不变。由表 3 可以得知,AWOA 在函数 f_6 、 f_8 、 f_{13} 上平均值最小

且在 f_{12} 上标准差及平均值最小,MWOA-CEE 在函数 f_6 、 f_{13} 上标准差最小,EGWO 在函数 f_8 上标准差最小,CMAIS-WOA 在函数 $f_1 \sim f_4$ 、 f_7 上表现较为优异。在其他函数性能评估上,IWOA 算法表现最好。在求解精度方面,6 种改进算法总体上 CMAIS-WOA 和 IWOA 表现最好。在时间消耗方面,EGWO、AWOA、CMAIS-WOA、IWOA 在 13 个基准测试函数的平均消耗时间分别是 0.09、0.19、1.61、0.06 s,其中 IWOA 的时间消耗最少,远小于 AWOA、CMAIS-WOA 这两种 WOA 的改进算法。因此 IWOA 的整体性能更优(“—”表示该文献未做此实验)。

3.6 CEC2014 测试函数实验分析

为了更好的评估 IWOA 算法的性能,对 CEC2014 基准测试函数集中的 8 个复合函数进行测试,由表 4 可知 IWOA 算法除在第 4 个测试函数中表现不佳外,在其他 7 个函数上,无论是求解精度还是运行时间消耗上都表现最好。

4 IWOA 在工程设计上的应用

在本节中,在 2 个工程设计问题上进行算法的性能评估:焊接梁和压力容器设计问题^[23]。并将 IWOA 的结果与 PSO、DE、GWO、WOA 进行对比。最后给出解决问题的最优成本和相应的最优解。在整个优化过程中,目标函数评估的次数为 1×10^5 次。

4.1 焊接梁设计

焊接梁设计是一个复杂的设计问题,需要满足式(20)中的 5 个不等式,并最小化焊接材料的成本,其中 x_1 、 x_2 、 x_3 和 x_4 分别表示焊缝宽度、横梁长度和焊接梁的高度和厚度。

$$\min f(\mathbf{X}) = 0.04811x_3x_4(x_2 + 14) + 1.10471x_1^2x_2 \quad (19)$$

$$\begin{cases} g_1(\mathbf{X}) = x_1 - x_4 \leq 0 \\ g_2(\mathbf{X}) = \delta(\mathbf{X}) - \delta_{\max} \leq 0 \\ g_3(\mathbf{X}) = P \leq P_c(\mathbf{X}) \\ g_4(\mathbf{X}) = \tau_{\max} \geq \tau(\mathbf{X}) \\ g_5(\mathbf{X}) = \sigma(\mathbf{X}) - \sigma_{\max} \leq 0 \end{cases} \quad (20)$$

表 5 的结果表明 IWOA 的经济成本最低为 1.744,相比于 WOA,成本节约了 3.94%,同时耗时最少。

4.2 压力容器设计

压力容器设计的目的是在满足生产需求的同时降低成本。满足式(22)的 4 个不等式,同时最小化目标函数式(21),其中 z_1 表示外壳的厚度, z_2 表示头部的厚度, y_3 表示内径, y_4 表示长度。

表 3 与其他改进算法结果对比
Table 3 Comparison of results with other improved algorithms

函数	准则	EGWO	AWOA	CMAIS-WOA	EWOA	MWOA-CEE	IWOA
f_1	最优值	7.80×10^{-19}	7.16×10^{-102}	0.00×10^0	5.39×10^{-161}	3.57×10^{-120}	0.00×10^0
	平均值	5.27×10^{-17}	3.71×10^{-85}	0.00×10^0	4.59×10^{-139}	2.78×10^{-112}	3.83×10^{-319}
	标准差	8.19×10^{-17}	2.03×10^{-84}	0.00×10^0	2.37×10^{-138}	1.18×10^{-111}	0.00×10^0
	时间	0.048 8	0.150 0	0.406 1	—	—	0.022 2
f_2	最优值	2.54×10^{-11}	1.66×10^{-53}	0.00×10^0	4.51×10^{-90}	1.66×10^1	7.81×10^{-193}
	平均值	1.18×10^{-10}	1.22×10^{-45}	1.11×10^{-238}	6.64×10^1	5.06×10^{-61}	4.54×10^{-159}
	标准差	1.19×10^{-10}	6.24×10^{-45}	0.00×10^0	3.63×10^{-78}	1.73×10^{-60}	2.48×10^{-158}
	时间	0.052 0	0.154 6	0.480 1	—	—	0.024 0
f_3	最优值	3.68×10^{-5}	1.10×10^{-93}	0.00×10^0	4.07×10^{-40}	2.98×10^{-63}	0.00×10^0
	平均值	3.60×10^{-2}	8.80×10^{-80}	0.00×10^0	9.75×10^{-9}	5.66×10^{-50}	3.60×10^{-311}
	标准差	8.70×10^{-2}	4.72×10^{-79}	0.00×10^0	3.56×10^{-8}	2.17×10^{-49}	0.00×10^0
	时间	0.126 7	0.228 4	2.826 8	—	—	0.106 1
f_4	最优值	5.89×10^{-5}	3.15×10^{-52}	0.00×10^0	1.98×10^{-62}	4.69×10^{-50}	2.88×10^{-194}
	平均值	5.30×10^{-4}	3.38×10^{-44}	1.90×10^{-221}	6.06×10^{-47}	1.25×10^{-47}	3.42×10^{-161}
	标准差	3.95×10^{-4}	1.11×10^{-43}	0.00×10^0	1.82×10^{-46}	3.62×10^{-47}	1.74×10^{-160}
	时间	0.047 7	0.146 0	0.373 9	—	—	0.021 8
f_5	最优值	2.63×10^1	8.21×10^{-1}	2.77×10^1	2.67×10^1	2.52×10^1	2.36×10^{-5}
	平均值	2.75×10^1	2.19×10^1	2.82×10^1	2.76×10^1	2.62×10^1	5.78×10^0
	标准差	7.73×10^{-1}	1.02×10^1	2.61×10^{-1}	5.26×10^{-1}	5.60×10^{-1}	9.58×10^0
	时间	0.057 4	0.155 0	0.712 0	—	—	0.030 6
f_6	最优值	5.24×10^{-1}	2.38×10^{-5}	1.96×10^0	—	6.98×10^{-3}	1.04×10^{-6}
	平均值	1.08×10^0	6.81×10^{-3}	3.39×10^0	—	1.20×10^{-2}	2.17×10^{-2}
	标准差	3.20×10^{-1}	7.30×10^{-3}	8.25×10^{-1}	—	2.75×10^{-3}	5.09×10^{-2}
	时间	0.051 2	0.157 6	0.438 3	—	—	0.022 7
f_7	最优值	6.28×10^{-4}	1.01×10^{-5}	2.75×10^{-6}	—	4.12×10^{-5}	7.94×10^{-7}
	平均值	3.74×10^{-3}	1.19×10^{-4}	9.27×10^{-5}	—	3.03×10^{-4}	1.32×10^{-4}
	标准差	2.24×10^{-3}	1.01×10^{-4}	7.92×10^{-5}	—	2.49×10^{-4}	9.89×10^{-5}
	时间	0.109 9	0.214 3	2.320 9	—	—	0.081 1
f_8	最优值	-7.48×10^3	-1.26×10^4	-1.24×10^4	—	—	-1.26×10^4
	平均值	-6.21×10^3	-1.24×10^4	-8.69×10^3	—	—	-1.11×10^4
	标准差	6.29×10^2	6.52×10^2	1.73×10^3	—	—	2.36×10^3
	时间	0.066 6	0.170 8	1.183 0	—	—	0.035 3
f_9	最优值	5.12×10^{-13}	0.00×10^0	0.00×10^0	0.00×10^0	0.00×10^0	0.00×10^0
	平均值	8.62×10^0	0.00×10^0	0.00×10^0	0.00×10^0	0.00×10^0	0.00×10^0
	标准差	9.89×10^0	0.00×10^0	0.00×10^0	0.00×10^0	0.00×10^0	0.00×10^0
	时间	0.059 4	0.160 2	0.649 7	—	—	0.025 6
f_{10}	最优值	1.48×10^{-10}	8.88×10^{-16}	8.88×10^{-16}	—	8.88×10^{-16}	8.88×10^{-16}
	平均值	1.33×10^{-9}	2.78×10^{-15}	8.88×10^{-16}	—	3.49×10^{-15}	8.88×10^{-16}
	标准差	8.94×10^{-10}	1.80×10^{-15}	0.00×10^0	—	1.60×10^{-15}	0.00×10^0
	时间	0.056 6	0.153 5	0.604 4	—	—	0.025 4
f_{11}	最优值	0.00×10^0	0.00×10^0	0.00×10^0	0.00×10^0	0.00×10^0	0.00×10^0
	平均值	1.44×10^{-2}	0.00×10^0	5.52×10^2	0.00×10^0	0.00×10^0	0.00×10^0
	标准差	3.79×10^{-2}	0.00×10^0	2.26×10^2	0.00×10^0	0.00×10^0	0.00×10^0
	时间	0.059 7	0.155 3	1.373 9	—	—	0.031 4
f_{12}	最优值	3.72×10^{-2}	1.48×10^{-5}	6.74×10^{-3}	5.40×10^{-3}	4.56×10^{-4}	3.61×10^{-6}
	平均值	6.85×10^{-2}	6.06×10^{-4}	2.46×10^{-2}	2.54×10^{-2}	1.16×10^{-3}	2.90×10^{-2}
	标准差	3.65×10^{-2}	4.69×10^{-4}	1.39×10^{-2}	2.09×10^{-2}	1.37×10^{-3}	9.43×10^{-2}
	时间	0.190 3	0.287 5	4.775 8	—	—	0.161 8
f_{13}	最优值	5.59×10^{-2}	2.71×10^{-5}	1.05×10^{-1}	—	5.45×10^{-3}	2.29×10^{-7}
	平均值	8.66×10^{-1}	1.30×10^{-2}	3.56×10^{-1}	—	1.82×10^{-2}	7.98×10^{-2}
	标准差	3.28×10^{-1}	1.47×10^{-2}	1.51×10^{-1}	—	1.34×10^{-2}	2.71×10^{-1}
	时间	0.192 3	0.288 6	4.789 3	—	—	0.162 6

表 4 CEC2014 测试函数结果分析

Table 4 Analysis of CEC2014 test function results

函数	准则	PSO	DE	MFO	GWO	WOA	IWOA
CEC1	最优值	2.62×10 ³	3.32×10 ³	2.63×10 ³	2.62×10 ³	2.50×10³	2.50×10³
	平均值	2.62×10 ³	3.62×10 ³	2.67×10 ³	2.64×10 ³	2.71×10 ³	2.50×10 ³
	标准差	1.48×10 ⁰	1.63×10 ²	5.82×10 ¹	1.31×10 ¹	5.22×10 ¹	0.00×10⁰
	时间	0.189 2	0.569 1	0.210 9	0.212 7	0.175 8	0.173 2
CEC2	最优值	2.62×10 ³	2.87×10 ³	2.63×10 ³	2.60×10³	2.60×10³	2.60×10³
	平均值	2.63×10 ³	2.92×10 ³	2.68×10 ³	2.60×10³	2.61×10 ³	2.60×10³
	标准差	6.76×10 ⁰	2.75×10 ¹	2.93×10 ¹	2.60×10 ⁻²	5.02×10 ⁰	0.00×10⁰
	时间	0.154 1	0.555 1	0.174 9	0.174 8	0.135 4	0.132 4
CEC3	最优值	2.71×10 ³	2.75×10 ³	2.71×10 ³	2.70×10³	2.70×10³	2.70×10³
	平均值	2.72×10 ³	2.81×10 ³	2.72×10 ³	2.71×10 ³	2.72×10 ³	2.70×10³
	标准差	4.11×10 ⁰	2.61×10 ¹	8.77×10 ⁰	5.22×10 ⁰	2.01×10 ¹	0.00×10⁰
	时间	0.178 9	0.593 3	0.203 4	0.207 2	0.163 7	0.157 1
CEC4	最优值	2.70×10 ³	2.71×10 ³	2.70×10³	2.70×10³	2.70×10³	2.71×10 ³
	平均值	2.79×10 ³	2.71×10 ³	2.70×10³	2.75×10 ³	2.73×10 ³	2.73×10 ³
	标准差	5.16×10 ¹	1.10×10 ⁰	1.45×10 ⁰	5.07×10 ¹	4.46×10 ¹	3.37×10 ¹
	时间	0.783 4	1.211 3	0.820 7	0.811 4	0.772 3	0.737 9
CEC5	最优值	3.11×10 ³	3.74×10 ³	3.13×10 ³	3.12×10 ³	3.26×10 ³	2.90×10³
	平均值	3.46×10 ³	4.05×10 ³	3.60×10 ³	3.38×10 ³	3.98×10 ³	2.90×10³
	标准差	3.63×10 ²	1.43×10 ²	2.21×10 ²	1.23×10 ²	2.28×10 ²	0.00×10⁰
	时间	0.779 3	1.185 1	0.813 8	0.794 4	0.768 6	0.648 1
CEC6	最优值	6.13×10 ³	5.71×10 ³	3.73×10 ³	3.70×10 ³	4.79×10 ³	3.00×10³
	平均值	7.93×10 ³	6.94×10 ³	3.92×10 ³	4.15×10 ³	5.66×10 ³	3.00×10³
	标准差	1.03×10 ³	4.58×10 ²	1.47×10 ²	4.20×10 ²	5.51×10 ²	0.00×10⁰
	时间	0.235 0	0.630 9	0.256 8	0.262 0	0.222 9	0.199 0
CEC7	最优值	2.05×10 ⁴	8.10×10 ⁷	7.89×10 ³	2.20×10 ⁴	1.44×10 ⁵	3.10×10³
	平均值	3.50×10 ⁵	1.33×10 ⁸	3.33×10 ⁶	2.00×10 ⁶	1.40×10 ⁷	3.10×10³
	标准差	8.55×10 ⁵	2.95×10 ⁷	4.20×10 ⁶	4.56×10 ⁶	1.02×10 ⁷	0.00×10⁰
	时间	0.260 4	0.651 9	0.275 8	0.289 6	0.241 4	0.216 2
CEC8	最优值	1.32×10 ⁴	4.82×10 ⁵	7.69×10 ³	2.17×10 ⁴	1.48×10 ⁵	3.20×10³
	平均值	2.99×10 ⁴	1.74×10 ⁶	5.41×10 ⁴	1.07×10 ⁵	5.43×10 ⁵	3.20×10³
	标准差	1.15×10 ⁴	4.47×10 ⁵	5.29×10 ⁴	7.13×10 ⁴	4.15×10 ⁵	0.00×10⁰
	时间	0.166 4	0.548 4	0.187 0	0.191 3	0.152 0	0.141 0

表 5 焊接梁设计问题结果对比

Table 5 Comparison of results for welding beam design

算法	最优解				最优成本	时间
	x ₁	x ₂	x ₃	x ₄		
PSO	0.125	5.749	10	0.202	2.023	0.130
DE	0.190	3.725	10	0.196	1.825	2.281
GWO	0.199	3.845	9.389	0.200	1.783	0.144
WOA	0.125	5.643	9.349	0.198	1.847	0.108
IWOA	0.207	3.361	9.122	0.207	1.744	0.095

$$\min f(\mathbf{Y}) = 1.778 1z_2y_3^2 + 0.622 4z_1y_3y_4 + 3.166 1z_1^2y_4 + 19.84z_1^2y_3 \quad (21)$$

$$\begin{cases} g_1(\mathbf{Y}) = 0.009 54y_3 \leq z_2 \\ g_2(\mathbf{Y}) = 0.019 3y_3 \leq z_1 \\ g_3(\mathbf{Y}) = y_4 \leq 240 \\ g_4(\mathbf{Y}) = -\pi y_3^2y_4 - \frac{4}{3}\pi y_3^3 \leq -1 296 000 \end{cases} \quad (22)$$

$$z_1 = 0.062 5y_1 \quad (23)$$

$$z_2 = 0.062 5y_2$$

表 6 的结果表明 IWOA 的经济成本最低为 6 298.984, 相比于 WOA, 成本节约了 5.58%, 同时耗时最少。

表 6 压力容器设计问题结果对比

Table 6 Comparison of results for pressure vessel design

算法	最优解				最优成本	时间
	y ₁	y ₂	y ₃	y ₄		
PSO	13.992	8.176	41.784	180.781	6 770.029	0.077
DE	13.635	6.526	41.906	200	6 783.087	2.265
GWO	15.164	8.234	43.549	159.454	7 061.497	0.107
WOA	16.639	8.364	53.760	71.053	6 556.093	0.090
IWOA	15.009	7.705	48.595	110.030	6 298.984	0.076

将 IWOA 应用于焊接梁和压力容器设计两个具有实际应用价值的工程问题, 进一步验证了算法的有效性。

5 结 论

为了解决传统 WOA 收敛精度低、容易陷入局部最优的问题,本研究提出了一种基于自适应更新机制和邻域搜索策略的 IWOA,在算法前期寻优中引入个体历史最优位置,提高算法全局搜索能力,避免算法陷入局部最优;后期围绕全局历史最优位置进行邻域搜索,提高算法局部寻优能力。选取了典型的基准测试函数和焊接梁、压力容器设计 2 个复杂的工程设计问题通过 MATLAB 进行仿真实验。实验结果表明 IWOA 的求解精度和收敛速率更优,验证了算法的有效性。在后续的研究中,将进一步提高算法的稳定性,并将算法应用到室内定位,图像处理等实际应用领域。

参考文献

- [1] 曹文梁. 随机差分变异粒子群混合优化算法[J]. 电子测量与仪器学报, 2017, 31(6): 928-933.
CAO W L. Random difference mutation particle swarm optimization algorithm [J]. Journal of Electronic Measurement and Instrumentation, 2017, 31 (6): 928-933.
- [2] 张新明, 王霞, 康强, 等. GWO 与 ABC 的混合优化算法及其聚类优化[J]. 电子学报, 2018, 46(10): 2430-2442.
ZHANG X M, WANG X, KANG Q, et al. Hybrid grey wolf optimizer with artificial bee colony and its application to clustering optimization [J]. Acta Electronica Sinica, 2018, 46 (10): 2430-2442.
- [3] MIRJALILI S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm[J]. Knowledge-Based Systems, 2015, 89: 228-249.
- [4] 王宏健, 鄂鑫, 张凯, 等. 改进蚁群算法解决 UUV 集群任务规划问题[J]. 仪器仪表学报, 2022, 43 (9): 238-254.
WANG H J, E X, ZHANG K, et al. Improved ant colony algorithm to solve UUV cluster task planning problem [J]. Chinese Journal of Scientific Instrument, 2022, 43 (9): 238-254.
- [5] 潘志远, 卜凡亮. 基于蜣螂算法优化的 DV-Hop 定位算法 [J]. 电子测量与仪器学报, 2023, 37 (7): 33-41.
PAN ZH Y, BU F L. DV-Hop localization algorithm optimized based on dung beetle optimizer [J]. Journal of Electronic Measurement and Instrumentation, 2023, 37 (7): 33-41.

- [6] 张长胜, 张健忠, 钱斌, 等. 多策略融合的改进天鹰优化算法[J]. 电子学报, 2023, 51(5): 1245-1255.
ZHANG CH SH, ZHANG J ZH, QIAN B, et al. Improved aquila optimization based on multi-strategy integration [J]. Acta Electronica Sinica, 2023, 51 (5): 1245-1255.
- [7] MIRJALILI S, LEWIS A. The whale optimization algorithm [J]. Advances in Engineering Software, 2016, 95: 51-67.
- [8] HUSSIEN A G, HEIDARI A A, YE X, et al. Boosting whale optimization with evolution strategy and gaussian random walks: An image segmentation method [J]. Engineering with Computers, 2023, 39(3): 1935-1979.
- [9] LIU R, LIANG ZH H, WANG ZH Y, et al. Indoor visible light positioning based on improved whale optimization method with min-max algorithm [J]. IEEE Transactions on Instrumentation and Measurement, 2023, 72: 1-10.
- [10] DAI Y N, YU J Y, ZHANG C, et al. A novel whale optimization algorithm of path planning strategy for mobile robots [J]. Applied Intelligence, 2023, 53(9): 10843-10857.
- [11] YANG W B, XIA K W, FAN SH R, et al. A multi-strategy whale optimization algorithm and its application [J]. Engineering Applications of Artificial Intelligence, 2022, 108: 104558.
- [12] LIANG X D, ZHANG Z J. A whale optimization algorithm with convergence and exploitability enhancement and its application [J]. Mathematical Problems in Engineering, 2022, 2022: 1-19.
- [13] SI Q, LI C. Indoor robot path planning using an improved whale optimization algorithm [J]. Sensors, 2023, 23(8): 3988.
- [14] 褚鼎立, 陈红, 王旭光. 基于自适应权重和模拟退火的鲸鱼优化算法 [J]. 电子学报, 2019, 47 (5): 992-999.
CHU D L, CHEN H, WANG X G. Whale optimization algorithm based on adaptive weight and simulated annealing [J]. Acta Electronica Sinica, 2019, 47 (5): 992-999.
- [15] HUANG K W, WU Z X, JIANG CH L, et al. WPO: A whale particle optimization algorithm [J]. International Journal of Computational Intelligence Systems, 2023, 16(1): 115.
- [16] LIU L, ZHANG R. Multistrategy improved whale optimization algorithm and its application [J]. Computational Intelligence

and Neuroscience, 2022, 2022: 1-16.

- [17] 龙文, 蔡绍洪, 焦建军, 等. 一种改进的灰狼优化算法[J]. 电子学报, 2019, 47(1): 169-175.
LONG W, CAI SH H, JIAO J J, et al. An improved grey wolf optimization algorithm [J]. Acta Electronica Sinica, 2019, 47 (1): 169-175.
- [18] WANG CH F, CHU X Y. An improved firefly algorithm with specific probability and its engineering application[J]. IEEE Access, 2019, 7: 57424-57439.
- [19] MILLAH I S, CHANG P C, TESHOME D F, et al. An enhanced grey wolf optimization algorithm for photovoltaic maximum power point tracking control under partial shading conditions [J]. IEEE Open Journal of the Industrial Electronics Society, 2022, 3: 392-408.
- [20] ELHOSSEINI M A, HAIKAL A Y, BADAWY M, et al. Biped robot stability based on an a-c parametric whale optimization algorithm [J]. Journal of Computational Science, 2019, 31: 17-32.
- [21] 冯文涛, 宋科康. 一种增强型鲸鱼优化算法[J]. 计算机仿真, 2020, 37(11): 275-279, 357.
FENG W T, SONG K K. An enhanced whale optimization algorithm[J]. Computer Simulation, 2020, 37 (11): 275-279, 357.
- [22] HSIEH C-H, ZHANG Q, XU Y, et al. CMAIS-WOA: An improved WOA with chaotic mapping and adaptive iterative strategy [J]. Discrete Dynamics in Nature and Society, 2023, DOI:10.1155/2023/8160121.
- [23] KUMAR A, WU G, ALI M Z, et al. A test-suite of non-convex constrained optimization problems from the real-world and some baseline results [J]. Swarm and Evolutionary Computation, 2020, 56: 100693.

作者简介



谢良波 (通信作者), 2016 年于电子科技大学获得博士学位, 现为重庆邮电大学副教授, 主要研究方向为无线感知、混合信号集成电路、FPGA 信号处理。

E-mail: xielb@cqupt.edu.cn

Xie Liangbo (Corresponding author)

received his Ph. D. degree from the University of Electronic Science and Technology of China in 2016. Now he is an associate professor in the Chongqing University of Posts and Telecommunications. His main research interests include wireless sensing, mixed signal integrated circuits, and FPGA signal processing.



韩伸, 2022 年于华北水利水电大学获取学士学位, 现为重庆邮电大学硕士研究生, 主要研究方向为智能优化算法、室内定位。

E-mail: s220101038@stu.cqupt.edu.cn

Han Shen received his B. Sc. degree

from the North China University of Water Resources and Electric Power in 2022. Now he is a M. Sc. candidate in the Chongqing University of Posts and Telecommunications. His main research interests include intelligent optimization algorithms and indoor positioning.



张钰坤, 2021 年于南昌大学获得学士学位, 2024 年于重庆邮电大学获得硕士学位, 主要研究方向为无线定位。

E-mail: 1741904981@qq.com

Zhang Yunkun received his B. Sc. degree

from the Nanchang University in 2021 and M. Sc. degree from the Chongqing University of Posts and Telecommunications in 2024. His main research interest includes wireless positioning.