



## 一种多策略改进鲸鱼优化算法的混沌系统参数辨识

潘悦悦, 吴立飞, 杨晓忠

引用本文:

潘悦悦, 吴立飞, 杨晓忠. 一种多策略改进鲸鱼优化算法的混沌系统参数辨识[J]. 智能系统学报, 2024, 19(1): 176–189.

PAN Yueyue, WU Lifei, YANG Xiaozhong. Parameter identification of chaotic system based on a multi-strategy improved whale optimization algorithm[J]. *CAAI Transactions on Intelligent Systems*, 2024, 19(1): 176–189.

在线阅读 View online: <https://dx.doi.org/10.11992/tis.202303043>

## 您可能感兴趣的其他文章

### 随机交叉自学策略改进的教与学优化算法

Teaching-learning-based optimization algorithm based on random crossover-self-learning strategy

智能系统学报. 2021, 16(2): 313–322 <https://dx.doi.org/10.11992/tis.201910045>

### 基于改进PSO和FCM的模糊辨识

Fuzzy identification based on improved PSO and FCM

智能系统学报. 2019, 14(2): 378–384 <https://dx.doi.org/10.11992/tis.201707025>

### 具有Levy变异和精英自适应竞争机制的蚁狮优化算法

Ant lion optimizer with levy variation and adaptive elite competition mechanism

智能系统学报. 2018, 13(2): 236–242 <https://dx.doi.org/10.11992/tis.201706091>

### 一种求解多模态复杂问题的混合和声差分算法

Hybrid algorithm based on harmony search and differential evolution for solving multi-modal complex problems

智能系统学报. 2018, 13(2): 281–289 <https://dx.doi.org/10.11992/tis.201612030>

### 一种增强局部搜索能力的改进人工蜂群算法

Improved artificial bee colony algorithm based on enhanced local search

智能系统学报. 2017, 12(5): 684–693 <https://dx.doi.org/10.11992/tis.201612026>

### 基于混沌蜂群优化的指纹匹配算法

A fingerprint minutiae matching algorithm based on chaotic bee colony optimization

智能系统学报. 2016, 11(5): 613–618 <https://dx.doi.org/10.11992/tis.201601038>

DOI: 10.11992/tis.202303043

网络出版地址: <https://link.cnki.net/urlid/23.1538.TP.20230919.1042.002>

# 一种多策略改进鲸鱼优化算法的混沌系统参数辨识

潘悦悦<sup>1</sup>, 吴立飞<sup>2</sup>, 杨晓忠<sup>2</sup>

(1. 华北电力大学 控制与计算机工程学院, 北京 102206; 2. 华北电力大学 数理学院信息与计算研究所, 北京 102206)

**摘要:** 针对混沌系统参数辨识精度不高的问题, 以鲸鱼优化算法 (whale optimization algorithm, WOA) 为基础, 提出一种多策略改进鲸鱼优化算法 (multi-strategy improved whale optimization algorithm, MIWOA)。采用 Chebyshev 混沌映射选取高质量初始种群, 采用非线性收敛因子和自适应权重, 提高算法收敛速度, 为了避免算法陷入局部最优, 动态选择自适应  $t$  分布或蚁狮优化算法更新后期位置, 提高处理局部极值的能力。通过对 10 个基准函数和高维测试函数进行仿真试验, 表明 MIWOA 具有良好的稳定性和收敛精度。将 MIWOA 应用于辨识 Rössler 和 Lü 混沌系统参数, 仿真结果优于现有成果, 表明本文 MIWOA 辨识混沌系统参数的高效性和实用性。

**关键词:** 多策略改进鲸鱼优化算法; 混沌系统; 参数辨识; Chebyshev 混沌映射; 自适应  $t$  分布; 蚁狮优化算法; 基准函数; Wilcoxon 秩和检验

**中图分类号:** TP18 **文献标志码:** A **文章编号:** 1673-4785(2024)01-0176-14

中文引用格式: 潘悦悦, 吴立飞, 杨晓忠. 一种多策略改进鲸鱼优化算法的混沌系统参数辨识 [J]. 智能系统学报, 2024, 19(1): 176-189.

英文引用格式: PAN Yueyue, WU Lifei, YANG Xiaozhong. Parameter identification of chaotic system based on a multi-strategy improved whale optimization algorithm[J]. CAAI transactions on intelligent systems, 2024, 19(1): 176-189.

## Parameter identification of chaotic system based on a multi-strategy improved whale optimization algorithm

PAN Yueyue<sup>1</sup>, WU Lifei<sup>2</sup>, YANG Xiaozhong<sup>2</sup>

(1. School of Control and Computer Engineering, North China Electric Power University, Beijing 102206, China; 2. Institute of Information and Computing, School of Mathematics and Physics, North China Electric Power University, Beijing 102206, China)

**Abstract:** Aimed at the problem of low parameter identification accuracy of chaotic systems, a multi-strategy improved whale optimization algorithm (MIWOA) is proposed based on the whale optimization algorithm (WOA). MIWOA uses Chebyshev chaotic mapping to select high-quality initial populations, and nonlinear convergence factor and adaptive weight to improve the convergence speed of the algorithm. In order to avoid falling into local optimal solution, MIWOA dynamically selects adaptive  $t$  distribution or ant lion optimization algorithm to update the later position and improve the ability to handle local extremum. Through simulation experiments on 10 benchmark functions and high-dimensional test functions, it is shown that MIWOA has good stability and convergence accuracy. Applying MIWOA to identify the parameters of Rössler and Lü chaotic systems, the simulation results are superior to existing achievements, indicating the efficiency and practicality of MIWOA in identifying chaotic system parameters in this paper.

**Keywords:** multi-strategy improved whale optimization algorithm; chaotic system; parameter identification; Chebyshev chaotic map; adaptive  $t$  distribution; ant lion optimization algorithm; benchmark function; Wilcoxon rank sum test

在非线性科学领域, 混沌系统的控制与同步是研究热点, 其在信息科学、保密通信等方面应用广泛<sup>[1-2]</sup>, 准确的混沌系统模型是实现同步控制

的基础。然而, 实际应用中, 混沌系统结构复杂, 部分参数不可知, 因此精确辨识混沌系统参数具有重要的现实意义<sup>[3-4]</sup>。

近年来, 许多学者开始关注和研究混沌系统参数辨识问题, 并应用智能优化算法辨识系统参数。Chen 等 (2019 年)<sup>[5]</sup> 使用改进花授粉算法估

收稿日期: 2023-03-30. 网络出版日期: 2023-09-20.

基金项目: 中央高校基本科研业务费专项基金项目 (2021MS045); 华北电力大学国内外联合培养博士生资助项目 (2020).

通信作者: 杨晓忠. E-mail: [yxiaozh@ncepu.edu.cn](mailto:yxiaozh@ncepu.edu.cn).

©《智能系统学报》编辑部版权所有

计混沌和超混沌系统参数。数值模拟证明了新算法的有效性和鲁棒性。Ahandani 等 (2020 年)<sup>[6]</sup> 提出一种混沌复杂进化算法解决混沌系统参数估计问题,数值结果表明改进算法提高了参数估计的收敛速度和精度。Turgut 等 (2021 年)<sup>[7]</sup> 使用单元拓扑改进鲸鱼算法和正余弦算法,辨识混沌系统参数,测试结果证明,新算法成功地解决了参数识别问题。Ebrahimi 等 (2021 年)<sup>[8]</sup> 提出了一种改进洛兹映射混沌优化算法,数值结果表明,改进算法能够高效、准确地估计混沌系统参数。

鲸鱼优化算法 (whale optimization algorithm, WOA) 是一种新型群智能算法<sup>[9]</sup>,具有搜索能力强、计算稳定等特点,在最优控制、光伏系统等方面应用广泛<sup>[10-11]</sup>。然而 WOA 存在全局和局部搜索不平衡,易陷入局部最优等问题,因此国内外学者对 WOA 进行了有效的改进。Chen 等 (2020 年)<sup>[12]</sup> 提出一种准对立混沌 WOA (whale optimization algorithm with chaos mechanism based on quasi-opposition, OBCWOA),数值仿真结果证明了改进算法的全局搜索能力。Chen 等 (2020 年)<sup>[13]</sup> 为提高 WOA 的收敛精度和速度,提出一种双自适应随机备用增强 WOA (reinforced whale optimization algorithm, RDWOA)。仿真结果验证了 RDWOA 的有效性。Chakraborty 等 (2021 年)<sup>[14]</sup> 为解决高维问题,结合多种策略,提出一种增强 WOA (enhanced whale optimization algorithm, eWOA),测试结果表明所提算法能有效解决高维问题。Shen 等 (2023 年)<sup>[15]</sup> 提出一种多策略进化 WOA (whale optimization algorithm based on multi-population evolution, MEWOA),试验结果证明了 MEWOA 求解优化问题的有效性。Deng 等 (2023 年)<sup>[16]</sup> 提出了一种具有多策略混合算法的改进 WOA (improved whale optimization algorithm, IWOA),仿真结果表明, IWOA 有较好的收敛速度和稳定性。

以上改进算法虽然有一定提升,但在收敛精度和速度方面仍有待提高,本文在已有工作的基础上,以 WOA 为基础,使用 Chebyshev 混沌映射产生均匀分布种群,增加初始种群多样性。将收敛因子非线性化,增加自适应权重,兼顾全局搜索和局部挖掘。动态选择自适应  $t$  分布或蚁狮优化算法<sup>[17-18]</sup>,跳出局部最优。通过对 10 个基准函数和高维测试函数进行仿真试验,验证了 MIWOA 的优越性。将 MIWOA 应用于 Rössler 和 Lü 混沌系统参数辨识,仿真结果证明了 MIWOA 辨识混沌系统参数的有效性。

## 1 多策略改进鲸鱼优化算法的建立

### 1.1 基本 WOA

WOA 模拟了鲸鱼捕食行为,其仿生学原理描述如下:

#### 1) 包围捕食阶段。

鲸鱼寻找食物时,当前鲸鱼根据最佳鲸鱼位置更新自身位置,公式为

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

$$X(t+1) = X^*(t) - A \cdot D \quad (2)$$

$$A = 2a \cdot r_1 - a \quad (3)$$

$$C = 2 \cdot r_2 \quad (4)$$

式中:  $a = 2 - 2t/T$ ,  $t$  和  $T$  分别为当前迭代次数和最大迭代次数,  $X^*(t)$  和  $X(t)$  分别为最佳鲸鱼位置和当前位置,  $r_1$ 、 $r_2$  为  $[0,1]$  的随机数。

#### 2) 螺旋更新阶段。

在螺旋更新位置时,鲸鱼通过螺旋向上的方式,靠近群体中最优位置,更新公式为

$$X(t+1) = X^*(t) + D' \cdot e^{bl} \cos(2\pi l) \quad (5)$$

$$D' = |X^*(t) - X(t)| \quad (6)$$

式中:  $D'$  为第  $i$  只鲸鱼和猎物之间的距离,  $b$  为改变螺旋形状的常数,  $l$  为  $[-1,1]$  中的随机数。

在鲸鱼搜索猎物时,收缩包围和螺旋更新同步进行,位置更新公式为

$$X(t+1) = \begin{cases} X^*(t) - A \cdot D, & p < 0.5 \\ X^*(t) + D' \cdot e^{bl} \cos(2\pi l), & p \geq 0.5 \end{cases} \quad (7)$$

式中  $p$  为  $[0,1]$  之间的随机数。

#### 3) 随机搜寻阶段。

鲸鱼通过  $|A|$  大小,选择随机搜寻或者包围捕食。当  $|A| > 1$  时,鲸鱼在包围圈外,通过随机搜寻获得猎物信息,位置更新公式为

$$D = |C \cdot X_{\text{rand}}(t) - X(t)| \quad (8)$$

$$X(t+1) = X_{\text{rand}}(t) - A \cdot D \quad (9)$$

式中  $X_{\text{rand}}$  为当前的一个随机鲸鱼位置。

### 1.2 WOA 的改进

WOA 初始化种群时采用随机搜索策略,全局搜索能力弱。包围捕食和螺旋更新阶段,收敛因子线性递减不能平衡全局和局部搜索。算法后期容易陷入局部最优。针对以上缺点,本文通过 Chebyshev 混沌映射初始化种群。包围捕食和螺旋更新阶段,非线性化收敛因子,加入自适应权重。算法后期,动态使用自适应  $t$  分布或蚁狮优化算法更新鲸鱼位置,对 WOA 进行改进。

#### 1) Chebyshev 混沌映射。

Chebyshev 混沌映射对初值敏感<sup>[19]</sup>,可产生大量无周期实值序列。本文采用 Chebyshev 混沌映射对 WOA 进行种群初始化,迭代方式为

$$x_{n+1} = \cos(k(\arccos(x_n))), x_n \in [-1, 1] \quad (10)$$

式中 $k$ 为阶次。Chebyshev 映射迭代 1 000 次的分布如图 1 所示。

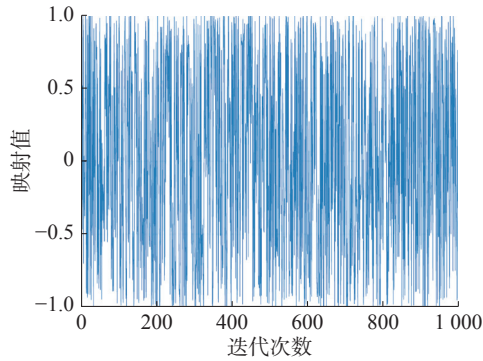


图 1 Chebyshev 映射分布

Fig. 1 Chebyshev map distribution

由图 1 可知,映射值分布于 $[-1, 1]$ , Chebyshev 混沌映射能更均衡地选取初始种群。将 Chebyshev 混沌序列映射到 WOA 解空间中,规定鲸鱼种群数为 $N$ ,随机产生第 1 只鲸鱼个体向量,  $Y = (y_1, y_2, \dots, y_d)$ 。使用式 (10) 对 $Y$ 的各维进行 $n-1$ 次迭代,产生其余的 $n-1$ 只鲸鱼。映射生成 $n$ 只鲸鱼个体为

$$x_{id} = l_d + (1 + y_{id}) \times (u_d - l_d) / 2 \quad (11)$$

式中: $u_d$ 和 $l_d$ 分别为搜索空间第 $d$ 维的上下界, $y_{id}$ 和 $x_{id}$ 分别为第 $d$ 维的第 $i$ 只鲸鱼和第 $i$ 只鲸鱼的坐标值。

## 2) 非线性收敛因子和自适应权重。

WOA 中,参数 $A \in [-a, a]$ 调节全局搜索和局部挖掘。当 $|A| \geq 1$ 时,算法进行全局搜索,当 $|A| < 1$ 时,算法进行局部挖掘。随着迭代的进行, $a = 2 - 2t/T$ 线性递减不能体现实际优化过程,因此改进收敛因子,公式为<sup>[20]</sup>

$$a = 2 - 2 \left( \frac{e^{t/T} - 1}{e - 1} \right)^u \quad (12)$$

式中: $u$ 为大于零的常数,调节 $a$ 的衰减程度。 $T = 500$ 时,图 2 为 $a$ 随 $u$ 的变化曲线。

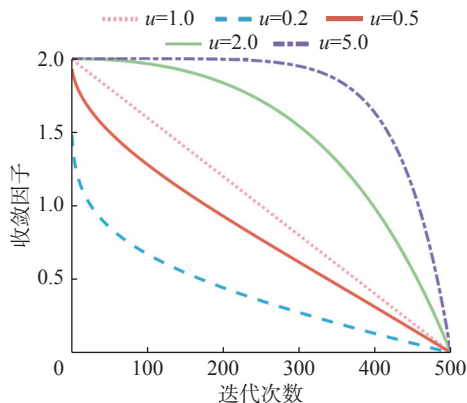


图 2 a 的衰减曲线

Fig. 2 Attenuation curves of a

如图 2,与 WOA 相比, $u$ 越大, $a > 1$ 所占的迭代次数比例越大,算法全局搜索能力越强,局部挖掘能力越弱。反之, $u$ 越小, $a < 1$ 所占的比例越大,算法局部挖掘能力越强,全局搜索能力越弱。所以 $a$ 的变化趋势很大程度地影响优化求解。由于本文提出的 Chebyshev 混沌映射初始化种群策略,可以有效提高算法全局搜索能力,故选取 $u = 0.6$ ,增强局部挖掘能力,加快收敛速度,提高精度。

针对 WOA 后期局部挖掘时,权重固定,不利于算法寻优,本文提出一种自适应权重策略,公式为

$$\omega = \frac{a}{2} \quad (13)$$

$$X(t+1) = \omega X^*(t) - AD \quad (14)$$

$$X(t+1) = \omega X^*(t) - D' e^{bl} \cos(2\pi l) \quad (15)$$

## 3) 自适应 t 分布。

$t$  分布又称学生分布,概率密度函数为<sup>[17]</sup>

$$p_t(x) = \frac{\Gamma\left(\frac{n+1}{2}\right)}{\sqrt{n\pi}\Gamma\left(\frac{n}{2}\right)} \cdot \left(1 + \frac{x^2}{n}\right)^{-\frac{n+1}{2}} \quad (16)$$

式中 $\Gamma\left(\frac{n+1}{2}\right) = \int_0^{+\infty} x^{\frac{n+1}{2}-1} e^{-x} dx$ 。图 3 为 $n$ 取不同值时 $t$ 分布图像对比。

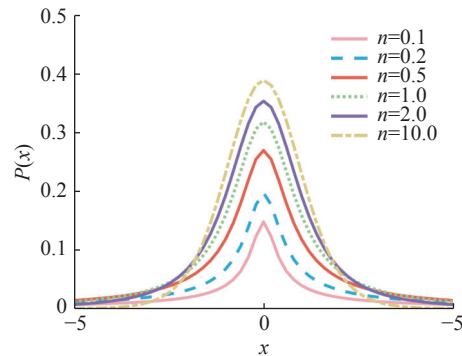


图 3 函数分布对比

Fig. 3 Function distribution comparison

由图 3 可知, $t$  分布的自由度 $n$ 越小,曲线的双尾翘越高,中间峰值越小,整体越平滑。反之,自由度 $n$ 越大,中间峰值越大,整体越陡峭。自适应 $t$  分布对鲸鱼位置的更新为

$$x'_i = x_i + x_i \times t(n) \quad (17)$$

式中: $x'_i$ 为更新后的鲸鱼位置, $n$ 和 $t(n)$ 分别为迭代次数和以迭代次数为自由度的 $t$  分布函数。

## 4) 蚁狮优化算法。

在蚁狮优化算法中,蚂蚁通过随机游走更新位置,公式为<sup>[18]</sup>

$$X(t) = [0, \text{cumsum}(2r(t_1) - 1), \text{cumsum}(2r(t_2) - 1), \dots, \text{cumsum}(2r(t_T) - 1)] \quad (18)$$



式中:  $\text{cumsum}$  为蚂蚁游走位置累加和,  $T$  为最大迭代次数,  $t$  为当前迭代次数,  $r(t)$  为 0 或 1 的随机数。蚂蚁随机游走公式的标准化形式为

$$X_i^t = \frac{(X_i^t - a_i) \times (d_i^t - c_i^t)}{(b_i - a_i)} + c_i^t \quad (19)$$

式中:  $a_i$  和  $b_i$  分别为第  $i$  个变量的最小值和最大值,  $c_i^t$  和  $d_i^t$  分别为第  $t$  代第  $i$  个变量的最小值和最大值。

## 2 混沌系统参数辨识的 MIWOA

### 2.1 混沌系统的参数辨识原理

混沌系统参数辨识原理如图 4 所示。

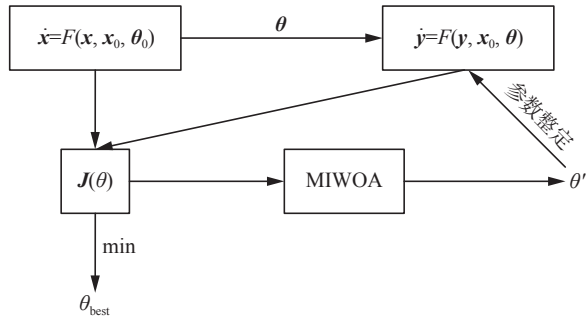


图 4 混沌系统参数辨识原理

Fig. 4 Principle of parameter identification for chaotic system

考虑如下  $n$  维混沌动力学系统:

$$\dot{\mathbf{x}} = F(\mathbf{x}, \mathbf{x}_0, \theta_0)$$

式中:  $\mathbf{x} = [x_1 \ x_2 \ \cdots \ x_n]^T \in \mathbf{R}^n$  为系统的  $n$  维状态变量,  $\mathbf{x}_0$  为初始状态量,  $\theta_0 = [\theta_{10} \ \theta_{20} \ \cdots \ \theta_{m0}]^T$  为参数真实值。当辨识系统参数时, 假设结构为

$$\dot{\mathbf{y}} = F(\mathbf{y}, \mathbf{x}_0, \theta)$$

式中:  $\mathbf{y} = [y_1 \ y_2 \ \cdots \ y_n]^T \in \mathbf{R}^n$  为辨识系统的状态变量,  $\theta = [\theta_1 \ \theta_2 \ \cdots \ \theta_m]^T$  为参数估计值。系统参数辨识是寻找一组最优未知参数, 使系统真值  $\mathbf{x}$  和估计值  $\mathbf{y}$  的误差值最小:

$$\min J(\theta) = \frac{1}{m} \sum_{k=1}^m \|\mathbf{x}_k - \mathbf{y}_k\|^2$$

式中:  $\mathbf{x}_k$  和  $\mathbf{y}_k$  分别为  $k$  时刻的系统真值和估计值,  $m$  为参数辨识状态变量序列的长度。目标泛函  $J(\theta)$  的值越小, 参数辨识的精度越高。MIWOA 流程如图 5 所示。

### 2.2 MIWOA 的时间复杂度分析

设种群规模为  $N$ , 搜索空间维度为  $D$ , 最大迭代次数为  $T$ , WOA 的时间复杂度为  $O(NDT)$ 。本文 MIWOA 以 WOA 为基础进行改进, Chebyshev 混沌映射初始化种群的时间复杂度为  $O(ND)$ , 非线性收敛因子和自适应权重未在基本 WOA 基础上增加循环嵌套, 时间复杂度为  $O(NDT)$ , 动态选择自适应  $t$  分布或蚁狮优化算法的时间为  $t_1$ , 则其

时间复杂度为  $O(NDT + t_1)$ , 虽然 MIWOA 的时间复杂度相对 WOA 有所增加, 但在可接受范围内, 下面通过数值试验说明 MIWOA 的卓越性。

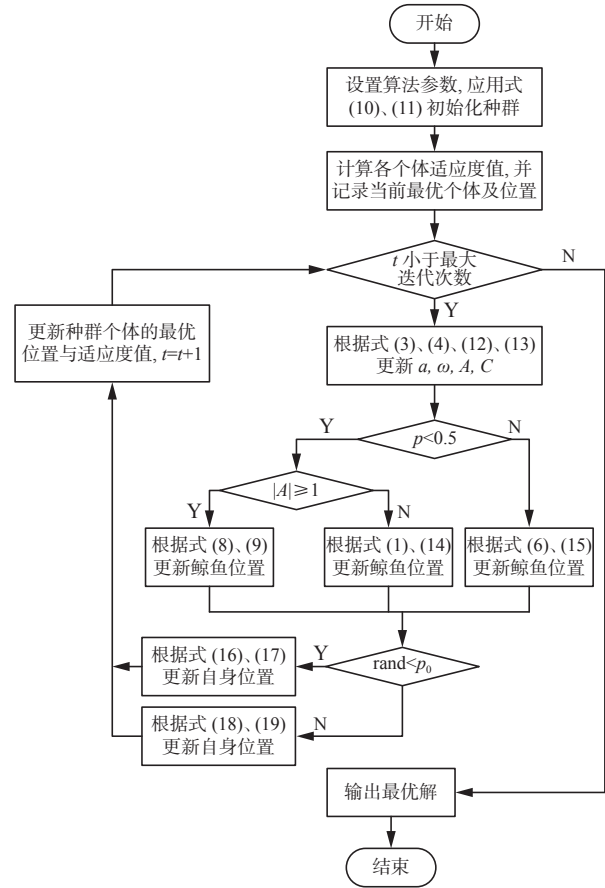


图 5 MIWOA 流程

Fig. 5 Flow chart of MIWOA

## 3 MIWOA 性能测试与分析

### 3.1 测试函数的选取

仿真试验基于 AMD Ryzen R7 5700U CPU@1.80 GHz, 在 Matlab R2019a 环境下运行。为了验证 MIWOA 有更好的寻优性能, 选取表 1 的 10 个基准函数进行测试,  $\varepsilon$  表示绝对误差精度。  $f_1 \sim f_7$  为单峰函数,  $f_8 \sim f_{10}$  为多峰函数, 10 个测试函数的最优值均为 0。

表 1 基准函数

Table 1 Benchmark functions

函数	搜索范围	$\varepsilon$
Sphere Function( $f_1$ )	$[-100, 100]$	0.001
Schwefel's Problem 2.22( $f_2$ )	$[-10, 10]$	0.001
Schwefel's Problem 1.2( $f_3$ )	$[-100, 100]$	0.001
Schwefel's Problem 2.21( $f_4$ )	$[-100, 100]$	0.001
Rosenbrock's Function( $f_5$ )	$[-30, 30]$	0.010

续表 1

函数	搜索范围	$\varepsilon$
Step Function( $f_6$ )	$[-100,100]$	0.010
Quartic Function( $f_7$ )	$[-1.28,1.28]$	0.010
Schwefel's Problem 2.26( $f_8$ )	$[-500,500]$	100.000
Rastrigin's Function( $f_9$ )	$[-5.12,5.12]$	0.010
Ackley's Function( $f_{10}$ )	$[-32,32]$	0.010

## 3.2 MIWOA 与其他智能算法对比

采用 10 个基准函数检验 MIWOA 的性能,与 WOA、粒子群算法 (particle swarm optimization, PSO)<sup>[21]</sup>、人工蜂群算法 (artificial bee colony algorithm, ABC)<sup>[22]</sup>、灰狼算法 (grey wolf optimizer, GWO)<sup>[23]</sup> 和 Harris 鹰算法 (harris hawks optimization, HHO)<sup>[24]</sup> 进行测试结果比较。令  $N = 30$ ,  $D = 30$ ,  $T = 1\,000$ , 运行 30 次, 表 2 为 6 种算法的最优值、最差值、平均值和标准差 (黑色粗体为最优结果)。

表 2 测试函数优化结果  
Table 2 Test function optimization results

函数	指标	WOA	PSO	ABC	GWO	HHO	MIWOA
$f_1$	最优值	$2.57 \times 10^{-165}$	$4.01 \times 10^1$	$2.74 \times 10^{-1}$	$9.58 \times 10^{-61}$	$9.90 \times 10^{-214}$	<b>0</b>
	最差值	$3.15 \times 10^{-149}$	$1.38 \times 10^2$	$1.46 \times 10^0$	$8.42 \times 10^{-59}$	$1.29 \times 10^{-185}$	<b>0</b>
	平均值	$3.21 \times 10^{-150}$	$8.22 \times 10^1$	$6.20 \times 10^{-1}$	$2.04 \times 10^{-59}$	$1.29 \times 10^{-186}$	<b>0</b>
	标准差	$9.44 \times 10^{-150}$	$3.09 \times 10^1$	$3.26 \times 10^{-1}$	$2.37 \times 10^{-59}$	<b>0</b>	<b>0</b>
$f_2$	最优值	$1.81 \times 10^{-111}$	$2.59 \times 10^0$	$5.32 \times 10^{-2}$	$5.40 \times 10^{-36}$	$8.33 \times 10^{-107}$	<b>0</b>
	最差值	$3.52 \times 10^{-102}$	$1.03 \times 10^2$	$5.81 \times 10^1$	$4.23 \times 10^{-34}$	$4.72 \times 10^{-95}$	<b>0</b>
	平均值	$3.58 \times 10^{-103}$	$3.89 \times 10^1$	$1.41 \times 10^1$	$1.30 \times 10^{-34}$	$4.89 \times 10^{-96}$	<b>0</b>
	标准差	$1.05 \times 10^{-102}$	$2.63 \times 10^1$	$2.27 \times 10^1$	$1.18 \times 10^{-34}$	$1.41 \times 10^{-95}$	<b>0</b>
$f_3$	最优值	$3.69 \times 10^3$	$3.59 \times 10^3$	$3.28 \times 10^4$	$3.14 \times 10^{-19}$	$3.62 \times 10^{-176}$	<b>0</b>
	最差值	$4.64 \times 10^4$	$2.10 \times 10^4$	$8.86 \times 10^4$	$1.60 \times 10^{-13}$	$2.18 \times 10^{-149}$	<b>0</b>
	平均值	$2.01 \times 10^4$	$1.10 \times 10^4$	$6.78 \times 10^4$	$1.65 \times 10^{-14}$	$2.41 \times 10^{-150}$	<b>0</b>
	标准差	$1.32 \times 10^4$	$5.27 \times 10^3$	$1.68 \times 10^4$	$4.80 \times 10^{-14}$	$6.51 \times 10^{-150}$	<b>0</b>
$f_4$	最优值	$3.58 \times 10^{-2}$	$1.94 \times 10^0$	$5.27 \times 10^1$	$9.39 \times 10^{-16}$	$1.53 \times 10^{-102}$	<b>0</b>
	最差值	$8.89 \times 10^1$	$6.66 \times 10^0$	$6.61 \times 10^1$	$1.72 \times 10^{-13}$	$2.05 \times 10^{-90}$	<b>0</b>
	平均值	$2.47 \times 10^1$	$3.62 \times 10^0$	$5.92 \times 10^1$	$2.65 \times 10^{-14}$	$2.15 \times 10^{-91}$	<b>0</b>
	标准差	$2.75 \times 10^1$	$1.29 \times 10^0$	$4.60 \times 10^0$	$5.01 \times 10^{-14}$	$6.14 \times 10^{-91}$	<b>0</b>
$f_5$	最优值	$2.79 \times 10^1$	$3.95 \times 10^3$	$1.37 \times 10^6$	$2.79 \times 10^1$	<b><math>1.61 \times 10^{-3}</math></b>	$2.80 \times 10^1$
	最差值	$2.87 \times 10^1$	$6.29 \times 10^3$	$2.29 \times 10^6$	$2.87 \times 10^1$	<b><math>5.45 \times 10^{-3}</math></b>	$2.88 \times 10^1$
	平均值	$2.84 \times 10^1$	$5.36 \times 10^3$	$1.93 \times 10^6$	$2.84 \times 10^1$	<b><math>3.92 \times 10^{-3}</math></b>	$2.82 \times 10^1$
	标准差	$4.23 \times 10^{-1}$	$1.15 \times 10^3$	$4.53 \times 10^5$	$3.96 \times 10^{-1}$	<b><math>1.88 \times 10^{-3}</math></b>	$2.76 \times 10^{-1}$
$f_6$	最优值	$8.77 \times 10^{-2}$	$1.44 \times 10^2$	$4.45 \times 10^{-1}$	$9.96 \times 10^{-1}$	<b><math>1.77 \times 10^{-5}</math></b>	$9.23 \times 10^{-3}$
	最差值	$2.72 \times 10^{-1}$	$3.09 \times 10^2$	$6.81 \times 10^{-1}$	$1.26 \times 10^0$	<b><math>6.11 \times 10^{-5}</math></b>	$2.72 \times 10^{-1}$
	平均值	$1.98 \times 10^{-1}$	$2.43 \times 10^2$	$5.86 \times 10^{-1}$	$1.15 \times 10^0$	<b><math>4.37 \times 10^{-5}</math></b>	$4.41 \times 10^{-2}$
	标准差	$9.03 \times 10^{-2}$	$8.11 \times 10^1$	$1.15 \times 10^{-1}$	$1.30 \times 10^{-1}$	<b><math>2.12 \times 10^{-5}</math></b>	$6.35 \times 10^{-2}$
$f_7$	最优值	$1.78 \times 10^{-4}$	$3.84 \times 10^{-2}$	$2.16 \times 10^{-1}$	$4.57 \times 10^{-4}$	$2.27 \times 10^{-6}$	<b><math>1.59 \times 10^{-6}</math></b>
	最差值	$5.24 \times 10^{-3}$	$1.58 \times 10^{-1}$	$8.24 \times 10^{-1}$	$2.07 \times 10^{-3}$	$1.25 \times 10^{-4}$	<b><math>9.40 \times 10^{-5}</math></b>
	平均值	$2.04 \times 10^{-3}$	$7.45 \times 10^{-2}$	$5.37 \times 10^{-1}$	$9.18 \times 10^{-4}$	$6.13 \times 10^{-5}$	<b><math>2.69 \times 10^{-5}</math></b>
	标准差	$1.57 \times 10^{-3}$	$3.40 \times 10^{-2}$	$1.75 \times 10^{-1}$	$4.24 \times 10^{-4}$	$4.43 \times 10^{-5}$	<b><math>3.15 \times 10^{-5}</math></b>

续表 2

函数	指标	WOA	PSO	ABC	GWO	HHO	MIWOA
$f_8$	最优值	$-1.26 \times 10^4$	$-1.04 \times 10^4$	$-8.22 \times 10^{112}$	$-6.60 \times 10^3$	$-1.26 \times 10^4$	$-1.26 \times 10^4$
	最差值	$-7.69 \times 10^3$	$-5.75 \times 10^3$	$-2.27 \times 10^{119}$	$-4.95 \times 10^3$	$-5.78 \times 10^3$	$-6.37 \times 10^3$
	平均值	$-1.12 \times 10^4$	$-8.14 \times 10^3$	$-2.41 \times 10^{118}$	$-6.06 \times 10^3$	$-1.19 \times 10^4$	$-1.15 \times 10^4$
	标准差	$1.84 \times 10^3$	$1.39 \times 10^3$	$6.77 \times 10^{118}$	$5.09 \times 10^2$	$2.04 \times 10^3$	$1.74 \times 10^3$
$f_9$	最优值	0	$1.23 \times 10^2$	$1.91 \times 10^2$	0	0	0
	最差值	0	$2.58 \times 10^2$	$2.50 \times 10^2$	$9.98 \times 10^{-1}$	0	0
	平均值	0	$1.90 \times 10^2$	$2.31 \times 10^2$	$1.99 \times 10^{-1}$	0	0
	标准差	0	$3.51 \times 10^1$	$1.67 \times 10^1$	$3.99 \times 10^{-1}$	0	0
$f_{10}$	最优值	$8.88 \times 10^{-16}$	$1.71 \times 10^0$	$3.03 \times 10^0$	$7.99 \times 10^{-15}$	$8.88 \times 10^{-16}$	$8.88 \times 10^{-16}$
	最差值	$7.99 \times 10^{-15}$	$2.20 \times 10^1$	$5.29 \times 10^0$	$2.58 \times 10^{-14}$	$8.88 \times 10^{-16}$	$8.88 \times 10^{-16}$
	平均值	$4.80 \times 10^{-15}$	$7.40 \times 10^0$	$4.01 \times 10^0$	$1.58 \times 10^{-14}$	$8.88 \times 10^{-16}$	$8.88 \times 10^{-16}$
	标准差	$2.49 \times 10^{-15}$	$6.32 \times 10^0$	$7.33 \times 10^{-1}$	$4.71 \times 10^{-15}$	0	0

由表 2 数据可知,对于单峰函数  $f_1$ 、 $f_2$ , MIWOA 的寻优结果均为 0, WOA 的计算结果优于 PSO、ABC 和 GWO 算法。在  $f_3$ 、 $f_4$  测试函数上, MIWOA 的计算结果优于 HHO, 可取得理论最小值。对于  $f_5 \sim f_7$  函数, 虽然 MIWOA 没有达到最小值, 但优于 WOA。对于多峰函数  $f_8 \sim f_{10}$ , MIWOA 可收敛到最优值附近, 其中  $f_9$  函数可收敛到最优值, 说明 MIWOA 相对于其他算法有更好的收敛精度和稳定性。

基准测试函数的收敛曲线可以清晰地展现

算法的收敛速度, 图 6(a)~(d) 和 (e)~(f) 分别为单峰和多峰函数的平均收敛曲线。由图 6 可以看出, 对于 6 个测试函数, 在收敛精度相同的情况下, MIWOA 收敛速度更快, 说明 Chebyshev 混沌映射初始化种群策略提高了种群中高质量个体的比例, 非线性收敛因子和自适应权重平衡了全局和局部搜索, 加速收敛。MIWOA 收敛曲线波动下降, 说明自适应  $t$  分布或蚁狮优化算法的动态选择策略有助于算法跳出局部最优, 提高收敛精度。

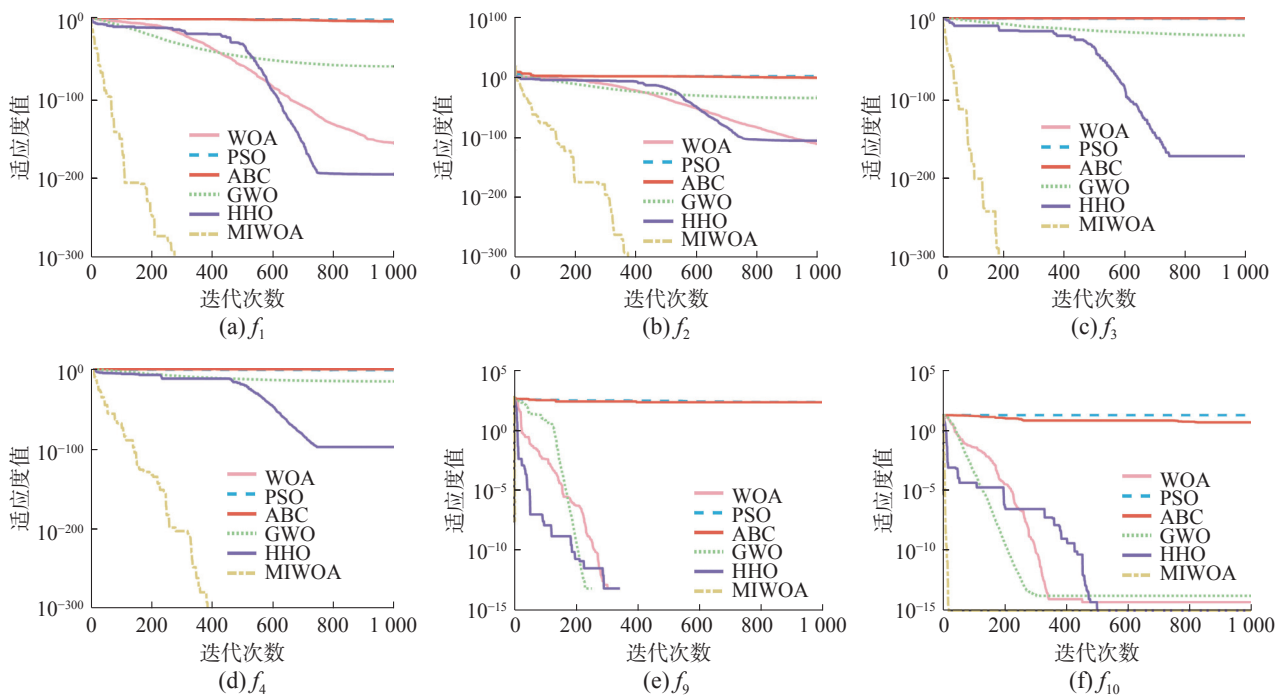


图 6 30 维测试函数的平均收敛曲线

Fig. 6 Average convergence curves of 30 dimensional test functions

### 3.3 Wilcoxon 秩和检验

Wilcoxon 秩和检验能够检测更为复杂的数据分布,并与算法多次运行的数据对比,公平地体现 MIWOA 的优越性<sup>[24]</sup>。实验设定显著性差异为 5%,当  $p < 5\%$  判定两算法有明显差异,反之无明显差异。符号“+”“-”和“=”分别表示 MIWOA 的性能优于、劣于和相当于对比算法,  $N/A$  表示无法进行显著性判断。选取 MIWOA 在 10 个测试函数的运行结果与 WOA、PSO、ABC、GWO 以及 HHO 运行结果进行 Wilcoxon 秩和检验,计算  $p$  值。表 3 结果显示大部分  $p < 5\%$ ,说明 MIWOA 的寻优能力优于 5 种对比算法。

### 3.4 MIWOA 不同改进策略的有效性分析

为比较 3 种改进策略对 MIWOA 性能的影响,令  $N = 30$ ,  $D = 30$ ,  $T = 500$ ,对  $f_1 \sim f_{10}$  进行寻优计算,将 MIWOA 与 WOA、WOA1(采用 Chebyshev 混沌映射)、WOA2(采用非线性收敛因子和自适应

权重)和 WOA3(采用自适应  $t$  分布或蚁狮优化算法)比较,计算结果如表 4。

表 3 Wilcoxon 秩和检验结果  
Table 3 Results of Wilcoxon rank sum test

函数	WOA	PSO	ABC	GWO	HHO
$f_1$	$2.00 \times 10^{-6}$	$2.00 \times 10^{-6}$	$2.00 \times 10^{-6}$	$2.00 \times 10^{-6}$	$2.00 \times 10^{-6}$
$f_2$	$2.00 \times 10^{-6}$	$2.00 \times 10^{-6}$	$2.00 \times 10^{-6}$	$2.00 \times 10^{-6}$	$2.00 \times 10^{-6}$
$f_3$	$2.00 \times 10^{-6}$	$2.00 \times 10^{-6}$	$2.00 \times 10^{-6}$	$2.00 \times 10^{-6}$	$2.00 \times 10^{-6}$
$f_4$	$2.00 \times 10^{-6}$	$2.00 \times 10^{-6}$	$2.00 \times 10^{-6}$	$2.00 \times 10^{-6}$	$2.00 \times 10^{-6}$
$f_5$	$1.25 \times 10^{-1}$	$2.00 \times 10^{-6}$	$2.00 \times 10^{-6}$	$8.51 \times 10^{-2}$	$2.00 \times 10^{-6}$
$f_6$	$6.00 \times 10^{-6}$	$2.00 \times 10^{-6}$	$2.00 \times 10^{-6}$	$2.00 \times 10^{-6}$	$2.00 \times 10^{-6}$
$f_7$	$2.00 \times 10^{-6}$	$2.00 \times 10^{-6}$	$2.00 \times 10^{-6}$	$2.00 \times 10^{-6}$	$1.47 \times 10^{-2}$
$f_8$	$3.13 \times 10^{-1}$	$2.30 \times 10^{-5}$	$N/A$	$2.00 \times 10^{-6}$	$1.40 \times 10^{-2}$
$f_9$	$N/A$	$2.00 \times 10^{-6}$	$2.00 \times 10^{-6}$	$2.00 \times 10^{-3}$	$N/A$
$f_{10}$	$9.00 \times 10^{-6}$	$2.00 \times 10^{-6}$	$2.00 \times 10^{-6}$	$9.03 \times 10^{-7}$	$N/A$
+/-/-	7/1/2	10/0/0	9/1/0	9/0/1	8/2/0

表 4 不同改进策略算法性能对比

Table 4 Performance comparison of algorithms for different improved strategies

函数	指标	WOA	WOA1	WOA2	WOA3	MIWOA
$f_1$	最优值	$1.25 \times 10^{-85}$	$3.72 \times 10^{-84}$	0	0	0
	最差值	$1.27 \times 10^{-71}$	$2.16 \times 10^{-72}$	0	0	0
	平均值	$6.84 \times 10^{-73}$	$1.08 \times 10^{-73}$	0	0	0
	标准差	$2.62 \times 10^{-72}$	$4.04 \times 10^{-73}$	0	0	0
$f_2$	最优值	$1.45 \times 10^{-58}$	$1.14 \times 10^{-57}$	$3.45 \times 10^{-264}$	0	0
	最差值	$2.85 \times 10^{-49}$	$1.73 \times 10^{-49}$	$5.63 \times 10^{-245}$	0	0
	平均值	$1.01 \times 10^{-50}$	$8.04 \times 10^{-51}$	$1.88 \times 10^{-246}$	0	0
	标准差	$5.11 \times 10^{-50}$	$3.12 \times 10^{-50}$	0	0	0
$f_3$	最优值	$1.46 \times 10^4$	$2.48 \times 10^4$	0	0	0
	最差值	$7.97 \times 10^4$	$1.41 \times 10^5$	0	0	0
	平均值	$4.70 \times 10^4$	$8.05 \times 10^4$	0	0	0
	标准差	$1.56 \times 10^4$	$2.81 \times 10^4$	0	0	0
$f_4$	最优值	$1.54 \times 10^0$	$4.04 \times 10^{-1}$	$5.95 \times 10^{-255}$	0	0
	最差值	$8.70 \times 10^1$	$9.66 \times 10^1$	$6.93 \times 10^{-242}$	0	0
	平均值	$3.93 \times 10^1$	$5.40 \times 10^1$	$4.02 \times 10^{-243}$	0	0
	标准差	$3.07 \times 10^1$	$3.29 \times 10^1$	0	0	0
$f_5$	最优值	$2.72 \times 10^1$	$2.71 \times 10^1$	$2.75 \times 10^1$	$1.12 \times 10^{-6}$	$1.38 \times 10^{-5}$
	最差值	$2.88 \times 10^1$	$2.88 \times 10^1$	$2.88 \times 10^1$	$1.51 \times 10^{-1}$	$8.58 \times 10^{-4}$
	平均值	$2.79 \times 10^1$	$2.79 \times 10^1$	$2.81 \times 10^1$	$2.10 \times 10^{-2}$	$3.27 \times 10^{-4}$
	标准差	$4.78 \times 10^{-1}$	$5.12 \times 10^{-1}$	$3.63 \times 10^{-1}$	$3.84 \times 10^{-2}$	$2.84 \times 10^{-4}$



续表 4

函数	指标	WOA	WOA1	WOA2	WOA3	MIWOA
$f_6$	最优值	$1.20 \times 10^{-1}$	$1.01 \times 10^{-1}$	$2.52 \times 10^{-1}$	$1.44 \times 10^{-7}$	$1.27 \times 10^{-7}$
	最差值	$1.29 \times 10^0$	$8.20 \times 10^{-1}$	$8.25 \times 10^{-1}$	$1.40 \times 10^{-2}$	$4.15 \times 10^{-6}$
	平均值	$4.82 \times 10^{-1}$	$3.58 \times 10^{-1}$	$4.80 \times 10^{-1}$	$1.39 \times 10^{-3}$	$1.65 \times 10^{-6}$
	标准差	$2.86 \times 10^{-1}$	$1.98 \times 10^{-1}$	$1.76 \times 10^{-1}$	$3.06 \times 10^{-3}$	$1.46 \times 10^{-6}$
$f_7$	最优值	$1.33 \times 10^{-5}$	$1.66 \times 10^{-4}$	$7.12 \times 10^{-7}$	$2.25 \times 10^{-6}$	$2.03 \times 10^{-6}$
	最差值	$9.45 \times 10^{-3}$	$1.77 \times 10^{-2}$	$3.71 \times 10^{-4}$	$2.71 \times 10^{-4}$	$2.86 \times 10^{-5}$
	平均值	$3.17 \times 10^{-3}$	$4.21 \times 10^{-3}$	$1.01 \times 10^{-4}$	$5.79 \times 10^{-5}$	$1.06 \times 10^{-5}$
	标准差	$2.78 \times 10^{-3}$	$4.76 \times 10^{-3}$	$8.87 \times 10^{-5}$	$5.22 \times 10^{-5}$	$8.36 \times 10^{-6}$
$f_8$	最优值	$-1.26 \times 10^4$	$-1.26 \times 10^4$	$-1.26 \times 10^4$	$-1.26 \times 10^4$	$-1.25 \times 10^4$
	最差值	$-7.44 \times 10^3$	$-7.61 \times 10^3$	$-1.11 \times 10^4$	$-1.13 \times 10^4$	$-1.17 \times 10^4$
	平均值	$-1.02 \times 10^4$	$-1.01 \times 10^4$	$-1.24 \times 10^4$	$-1.24 \times 10^4$	$-1.22 \times 10^4$
	标准差	$1.60 \times 10^3$	$1.75 \times 10^3$	$3.85 \times 10^2$	$2.49 \times 10^2$	$3.40 \times 10^2$
$f_9$	最优值	0	0	0	0	0
	最差值	0	0	0	0	0
	平均值	0	0	0	0	0
	标准差	0	0	0	0	0
$f_{10}$	最优值	$8.88 \times 10^{-16}$	$8.88 \times 10^{-16}$	$8.88 \times 10^{-16}$	$8.88 \times 10^{-16}$	$8.88 \times 10^{-16}$
	最差值	$7.99 \times 10^{-15}$	$7.99 \times 10^{-15}$	$8.88 \times 10^{-16}$	$8.88 \times 10^{-16}$	$8.88 \times 10^{-16}$
	平均值	$4.56 \times 10^{-15}$	$3.97 \times 10^{-15}$	$8.88 \times 10^{-16}$	$8.88 \times 10^{-16}$	$8.88 \times 10^{-16}$
	标准差	$2.51 \times 10^{-15}$	$2.71 \times 10^{-15}$	0	0	0

由表 4 可知, 3 种改进策略对 WOA 均有不同程度的提升, WOA3 的改进效果最好, WOA2 次之, WOA1 的计算结果低于其他 2 种算法的计算结果, 但对 WOA 仍有明显的改进效果。将 3 种改进策略相结合时, 算法搜索最精确, 明显优于 WOA1、WOA2 和 WOA3, 表明 3 种改进策略是有

效的。

### 3.5 MIWOA 与不同策略改进 WOA 对比

为了对比 MIWOA 与其他改进 WOA 的改进效果, 令  $N = 30$ ,  $D = 500$ ,  $T = 500$ , 引用 3 种算法 (OBCWOA<sup>[12]</sup>、eWOA<sup>[14]</sup> 和 MEWOA<sup>[15]</sup>) 的数据, 在相同测试条件下, 对  $f_1 \sim f_{10}$  测试, 结果如表 5。

表 5 MIWOA 与其他改进 WOA 性能对比

Table 5 Performance comparison between MIWOA and other improved WOAs

函数	指标	OBCWOA <sup>[12]</sup>	eWOA <sup>[14]</sup>	MEWOA <sup>[15]</sup>	MIWOA
$f_1$	平均值	$1.44 \times 10^{-272}$	$1.30 \times 10^{-209}$	0	0
	标准差	0	0	0	0
$f_2$	平均值	$2.09 \times 10^{-142}$	$1.25 \times 10^{-105}$	0	0
	标准差	$1.08 \times 10^{-141}$	$4.43 \times 10^{-105}$	0	0
$f_3$	平均值	$3.47 \times 10^{-239}$	$2.76 \times 10^{-110}$	0	0
	标准差	0	$1.30 \times 10^{-109}$	0	0
$f_4$	平均值	$3.02 \times 10^{-134}$	$2.83 \times 10^{-99}$	0	0
	标准差	$1.24 \times 10^{-133}$	$1.47 \times 10^{-98}$	0	0
$f_5$	平均值	$4.95 \times 10^3$	$9.51 \times 10^2$	$8.32 \times 10^2$	$4.98 \times 10^2$
	标准差	$2.26 \times 10^{-1}$	$7.91 \times 10^0$	$3.01 \times 10^0$	$9.44 \times 10^{-2}$

续表 5

函数	指标	OBCWOA <sup>[12]</sup>	eWOA <sup>[14]</sup>	MEWOA <sup>[15]</sup>	MIWOA
$f_6$	平均值	$2.84 \times 10^2$	$2.86 \times 10^1$	$2.06 \times 10^1$	$1.11 \times 10^2$
	标准差	$6.57 \times 10^1$	$3.28 \times 10^1$	$4.09 \times 10^0$	$2.31 \times 10^0$
$f_7$	平均值	$1.80 \times 10^{-4}$	$2.07 \times 10^{-4}$	$3.30 \times 10^{-4}$	$4.15 \times 10^{-6}$
	标准差	$1.47 \times 10^{-4}$	$1.88 \times 10^{-4}$	$3.53 \times 10^{-4}$	$2.80 \times 10^{-6}$
$f_8$	平均值	$-2.05 \times 10^5$	$-2.03 \times 10^5$	$-2.09 \times 10^5$	$-6.21 \times 10^4$
	标准差	$9.53 \times 10^3$	$1.50 \times 10^4$	$3.10 \times 10^2$	$4.04 \times 10^3$
$f_9$	平均值	0	0	0	0
	标准差	0	0	0	0
$f_{10}$	平均值	$1.01 \times 10^{-15}$	$8.88 \times 10^{-16}$	$8.88 \times 10^{-16}$	$8.88 \times 10^{-16}$
	标准差	$6.49 \times 10^{-16}$	0	0	0

表 5 结果显示, 对于函数  $f_1 \sim f_4$ , MIWOA 与 MEWOA 可以在有限次迭代收敛到最优值, 而 OBCWOA 和 eWOA 相对收敛精度较低。对于函数  $f_5 \sim f_{10}$ , MIWOA 可以使用更少的迭代次数收敛到最优值, 说明 MIWOA 比其他改进 WOA 寻优效果更好。

平均绝对误差 (mean absolute error, MAE) 是评价算法有效可行的重要指标, 计算公式为<sup>[23]</sup>

$$E_{\text{MAE}} = \frac{\sum_{i=1}^{N_f} |m_i - o_i|}{N_f} \quad (20)$$

式中:  $m_i$  为算法求解结果平均值,  $o_i$  为各测试函数理论值,  $N_f$  为测试函数个数。表 6 为除理论值非零的  $f_8$  函数外, 4 种算法的 MAE 排序, 由计算结果可知, MIWOA 的 MAE 值最小, 排名第 1, 证明了本文改进策略的有效性。

表 6 4 种改进 WOA 的 MAE 排名  
Table 6 MAE ranking of four improved WOAs

算法	MAE	排名
MIWOA	$6.77 \times 10^1$	1
MEWOA	$9.47 \times 10^1$	2
eWOA	$1.09 \times 10^2$	3
OBCWOA	$5.82 \times 10^2$	4

### 3.6 MIWOA 求解高维函数的实验分析

由上述计算结果可知, 对于低维测试函数, 本文 MIWOA 寻优效果良好, 但实际应用中高维大规模问题普遍存在, 为了检验 MIWOA 求解高维问题的可行性, 将其在 10 个基准函数,  $N = 30$ ,  $T = 500$ ,  $D = 200$ 、500、1000 情况下, 运行 30 次取平均值, 根据文献 [25] 求解算法寻优成功率。绝对误差精度如表 1 所示, 计算结果如表 7。

表 7 高维测试函数优化对比

Table 7 High dimensional test functions optimization comparison

维数	函数	WOA			MIWOA		
		平均值	标准差	成功率/%	平均值	标准差	成功率/%
200	$f_1$	$6.20 \times 10^{-104}$	$1.22 \times 10^{-103}$	100	0	0	100
	$f_2$	$6.90 \times 10^{-61}$	$1.05 \times 10^{-60}$	100	0	0	100
	$f_3$	$2.94 \times 10^6$	$3.68 \times 10^5$	0	0	0	100
	$f_4$	$6.41 \times 10^1$	$3.00 \times 10^1$	0	0	0	100
	$f_5$	$1.96 \times 10^2$	$6.86 \times 10^{-2}$	0	$1.99 \times 10^2$	$1.03 \times 10^{-1}$	0
	$f_6$	$8.71 \times 10^{-1}$	$7.01 \times 10^{-2}$	0	$4.20 \times 10^1$	$1.87 \times 10^0$	0
	$f_7$	$4.88 \times 10^{-4}$	$5.71 \times 10^{-4}$	100	$1.10 \times 10^{-5}$	$9.27 \times 10^{-6}$	100
	$f_8$	$-8.16 \times 10^4$	$2.68 \times 10^3$	0	$-2.69 \times 10^4$	$2.72 \times 10^3$	0
	$f_9$	0	0	100	0	0	100
	$f_{10}$	$4.44 \times 10^{-15}$	0	100	$8.88 \times 10^{-16}$	0	100

续表 7

维数	函数	WOA			MIWOA		
		平均值	标准差	成功率/%	平均值	标准差	成功率/%
500	$f_1$	$2.12 \times 10^{-109}$	$4.20 \times 10^{-109}$	100	0	0	100
	$f_2$	$5.06 \times 10^{-60}$	$8.84 \times 10^{-60}$	100	0	0	100
	$f_3$	$1.65 \times 10^7$	$2.51 \times 10^6$	0	0	0	100
	$f_4$	$9.23 \times 10^1$	$4.44 \times 10^0$	0	0	0	100
	$f_5$	$4.93 \times 10^2$	$2.68 \times 10^{-1}$	0	$4.98 \times 10^2$	$9.44 \times 10^{-2}$	0
	$f_6$	$1.41 \times 10^0$	$2.68 \times 10^{-1}$	0	$1.11 \times 10^2$	$2.31 \times 10^0$	0
	$f_7$	$1.01 \times 10^{-4}$	$6.41 \times 10^{-5}$	100	$4.15 \times 10^{-6}$	$2.80 \times 10^{-6}$	100
	$f_8$	$-2.02 \times 10^5$	$1.02 \times 10^4$	0	$-6.21 \times 10^4$	$4.04 \times 10^3$	0
	$f_9$	0	0	100	0	0	100
	$f_{10}$	$8.88 \times 10^{-16}$	0	100	$8.88 \times 10^{-16}$	0	100
1000	$f_1$	$5.77 \times 10^{-115}$	$1.15 \times 10^{-114}$	100	0	0	100
	$f_2$	$4.31 \times 10^{-64}$	$8.17 \times 10^{-64}$	100	0	0	100
	$f_3$	$6.16 \times 10^7$	$8.23 \times 10^6$	0	0	0	100
	$f_4$	$1.82 \times 10^1$	$3.13 \times 10^1$	20	0	0	100
	$f_5$	$9.89 \times 10^2$	$2.71 \times 10^{-1}$	0	$9.98 \times 10^2$	$1.64 \times 10^{-1}$	0
	$f_6$	$2.21 \times 10^0$	$4.32 \times 10^{-1}$	0	$2.26 \times 10^2$	$3.83 \times 10^0$	0
	$f_7$	$2.22 \times 10^{-4}$	$1.68 \times 10^{-4}$	100	$1.54 \times 10^{-6}$	$1.03 \times 10^{-6}$	100
	$f_8$	$-4.17 \times 10^5$	$2.64 \times 10^3$	0	$-1.27 \times 10^5$	$1.32 \times 10^4$	0
	$f_9$	0	0	100	0	0	100
	$f_{10}$	$4.44 \times 10^{-15}$	$2.25 \times 10^{-15}$	100	$8.88 \times 10^{-16}$	0	100

从表 7 可以看出,两算法在求解高维测试函数时,寻优精度比求低维函数略有下降,但整体仍可达到较高精度,获得理想结果。MIWOA 在 7 个基准函数上寻优成功率达到了 100%,相同维度下,MIWOA 的寻优效果比 WOA 更好,说明 MIWOA 对于求解高维优化问题具有较高的计算

精度和稳定性。

3.7 MIWOA 平均运行时间分析

为了验证 MIWOA 的计算速度,将其与 PSO、ABC、GWO、HHO、WOA、WOA1、WOA2、WOA3 运行 30 次的平均时间对比,  $N = 30$ ,  $D = 1\ 000$ ,  $T = 500$ , 计算结果如表 8。

表 8 基准函数寻优平均时间对比  
Table 8 Comparison of average time for optimization of benchmark function s

函数	PSO	ABC	GWO	HHO	WOA	WOA1	WOA2	WOA3	MIWOA
$f_1$	0.185 965	0.896 558	1.841 394	0.474 676	0.990 446	0.993 390	0.791 372	2.640 294	2.831 416
$f_2$	0.210 007	1.042 312	1.906 151	0.499 580	1.001 265	1.015 737	0.806 834	2.693 293	2.949 955
$f_3$	11.733 200	24.231 920	13.348 200	28.451 700	12.423 210	12.669 270	12.750 030	14.626 220	14.395 090
$f_4$	0.190 802	0.888 369	1.876 443	0.647 402	0.997 229	1.012 785	0.805 426	2.644 809	2.827 659
$f_5$	0.229 961	1.160 296	2.082 310	0.972 187	1.066 210	1.192 295	0.974 036	3.034 523	2.931 688
$f_6$	0.189 715	0.904 281	1.847 458	0.793 947	0.993 877	0.993 818	0.778 792	2.623 457	2.742 954
$f_7$	0.635 681	2.210 740	2.617 710	1.502 435	1.724 080	1.783 997	1.513 049	3.506 316	3.821 165
$f_8$	0.647 423	2.377 715	2.402 996	1.658 702	1.559 784	1.612 248	1.399 342	3.182 873	3.422 051
$f_9$	0.477 926	1.927 073	2.245 431	1.249 936	1.379 308	1.414 927	1.228 820	3.035 198	3.260 680
$f_{10}$	0.494 707	1.996 512	2.219 010	1.252 647	1.428 666	1.410 218	1.190 316	3.099 208	3.261 962

由表 8 可知,对于元启发式算法,WOA 与 ABC、HHO 运行时间相近,可以快速收敛到最优值。对于改进算法,WOA、WOA1 和 WOA2 的平均时长相当,说明 WOA1 和 WOA2 的改进策略未增加 WOA 运行时间。由于算法后期动态选择自适应  $t$  分布或蚁狮优化算法,增加了循环嵌套,WOA3 耗时多于 WOA。融合多种改进策略的 MIWOA 搜索范围变广,寻优时间增加,但均在合理范围内,与理论分析相符。

## 4 MIWOA 辨识混沌系统参数

### 4.1 MIWOA 对 Rössler 混沌系统的参数辨识

Rössler 系统具有简单、非对称吸引子结构,在保密通信中承担非常重要的作用。以典型 Rössler 混沌系统为例,验证 MIWOA 可以精确辨识混沌系统参数。Rössler 系统表达式为<sup>[7]</sup>

$$\begin{cases} \dot{x} = -y - z \\ \dot{y} = x + ay \\ \dot{z} = b + xz - cz \end{cases} \quad (21)$$

当参数  $a = 0.2$ ,  $b = 0.2$ ,  $c = 5.7$  时,系统为混沌状态,其演化过程如图 7 所示。令初值向量为  $[-1 \ 0 \ 1]^T$ ,步长  $h = 0.01$ ,使用四阶 Runge-Kutta 法

解式 (21),利用前 100 个解及 PSO、WOA、MIWOA 辨识待定参数  $[a \ b \ c]^T$ ,各算法独立运行 20 次。

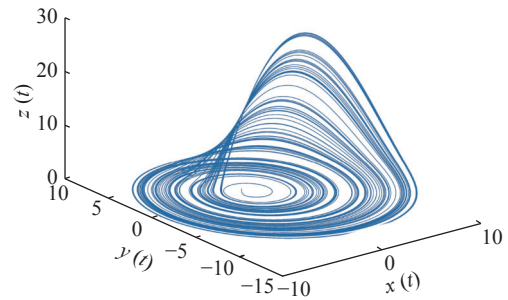


图 7 Rössler 混沌系统动力学轨迹

Fig. 7 Dynamic track of Rössler chaotic system

首先测试 3 种算法在具有一个未知参数的 Rössler 系统参数辨识中的搜索性能,即每次只辨识  $a$ 、 $b$ 、 $c$  3 个参数中的一个,测试结果如表 9。由表中数据可知,对于参数  $a$ ,MIWOA 可收敛到理论最优值,其最差值优于其他 2 种算法的最优值。对于参数  $b$ ,WOA 和 MIWOA 均可达到理论最优值,但 MIWOA 寻优效果更好。对于参数  $c$ ,MIWOA 的适应度值小于 PSO 和 WOA 的适应度值,估计值很接近真实值,说明 MIWOA 具有良好的全局搜索能力和计算鲁棒性。

表 9 不同算法的一维参数估计结果比较

Table 9 Comparison of one-dimensional parameter estimation results for different algorithms

评价指标	算法	实验1 ( $a$ 未知)		实验2 ( $b$ 未知)		实验3 ( $c$ 未知)	
		$a$	$J$	$b$	$J$	$c$	$J$
最优值	PSO	0.200000	$6.45 \times 10^{-13}$	0.199999	$9.08 \times 10^{-13}$	5.699993	$4.87 \times 10^{-12}$
	WOA	0.200000	$1.27 \times 10^{-14}$	0.200000	$2.29 \times 10^{-14}$	5.699999	$5.86 \times 10^{-14}$
	MIWOA	0.200000	0	0.200000	0	5.700000	0
平均值	PSO	0.200000	$4.38 \times 10^{-12}$	0.200001	$4.03 \times 10^{-12}$	5.700005	$1.13 \times 10^{-11}$
	WOA	0.200000	$1.63 \times 10^{-12}$	0.200000	$2.06 \times 10^{-13}$	5.700003	$8.71 \times 10^{-12}$
	MIWOA	0.200000	$1.73 \times 10^{-15}$	0.200000	$4.22 \times 10^{-16}$	5.700000	$1.09 \times 10^{-14}$
最差值	PSO	0.200001	$1.45 \times 10^{-11}$	0.200002	$8.55 \times 10^{-12}$	5.700014	$1.79 \times 10^{-11}$
	WOA	0.199999	$5.75 \times 10^{-12}$	0.200000	$3.72 \times 10^{-13}$	5.700015	$2.15 \times 10^{-11}$
	MIWOA	0.200000	$8.88 \times 10^{-15}$	0.200000	$2.00 \times 10^{-15}$	5.699999	$3.13 \times 10^{-14}$

然后测试各算法在具有 2 个未知参数的混沌系统参数辨识中的搜索性能,计算结果如表 10。从表 10 估计  $a$  和  $b$  可以看出,各算法搜索精度相似,相对于一个未知参数的情况,搜索精度略有下降,但 MIWOA 的计算结果仍优于 PSO 和 WOA。另外 2 种情况也得出同样的结论,验证了 MIWOA 辨识 2 个未知参数 Rössler 系统的可

行性。

最后,使用 3 种算法辨识具有 3 个未知参数的混沌系统,表 11 为各算法运行 20 次的统计结果。由表 11 中数据可知,各算法的搜索精度下降,但 MIWOA 的最差值优于 PSO 和 WOA 的最优值,在 3 种算法中搜索精度最高,表明 MIWOA 对 Rössler 混沌系统参数辨识更有效性。



表 10 不同算法的二维参数估计结果比较

Table 10 Comparison of two-dimensional parameter estimation results for different algorithms

评价指标	算法	实验1 ( $a$ 和 $b$ 未知)			实验2 ( $a$ 和 $c$ 未知)			实验3 ( $b$ 和 $c$ 未知)		
		$a$	$b$	$J$	$a$	$c$	$J$	$b$	$c$	$J$
最优值	PSO	0.199975	0.199974	$5.63 \times 10^{-8}$	0.200796	5.694613	$6.15 \times 10^{-5}$	0.201253	5.701971	$2.36 \times 10^{-5}$
	WOA	0.200059	0.200219	$1.07 \times 10^{-7}$	0.199152	5.702057	$5.07 \times 10^{-6}$	0.199107	5.697626	$7.13 \times 10^{-7}$
	MIWOA	0.199986	0.200016	$1.19 \times 10^{-8}$	0.200231	5.699750	$1.08 \times 10^{-6}$	0.199966	5.699299	$6.91 \times 10^{-8}$
平均值	PSO	0.200000	0.200052	$1.41 \times 10^{-7}$	0.200125	5.698197	$8.91 \times 10^{-5}$	0.199224	5.699131	$3.36 \times 10^{-5}$
	WOA	0.200000	0.199999	$4.16 \times 10^{-7}$	0.200123	5.699675	$1.48 \times 10^{-5}$	0.199780	5.699424	$1.29 \times 10^{-6}$
	MIWOA	0.199999	0.199970	$5.22 \times 10^{-8}$	0.200194	5.700061	$2.84 \times 10^{-6}$	0.200263	5.699959	$8.13 \times 10^{-7}$
最差值	PSO	0.200163	0.200431	$3.59 \times 10^{-7}$	0.196435	5.733973	$1.24 \times 10^{-4}$	0.191669	5.666020	$6.88 \times 10^{-5}$
	WOA	0.199767	0.200866	$1.66 \times 10^{-6}$	0.202415	5.694114	$4.12 \times 10^{-5}$	0.198638	5.696374	$1.66 \times 10^{-6}$
	MIWOA	0.199909	0.199745	$1.15 \times 10^{-7}$	0.200810	5.704961	$4.65 \times 10^{-6}$	0.201158	5.694933	$1.34 \times 10^{-6}$

表 11 不同算法的三维参数估计结果比较

Table 11 Comparison of three-dimensional parameter estimation results for different algorithms

评价指标	算法	$a$	$b$	$c$	$J$
最优值	PSO	0.201349	0.199904	5.699698	$1.06 \times 10^{-3}$
	WOA	0.200603	0.198218	5.693475	$1.02 \times 10^{-4}$
	MIWOA	0.200013	0.199482	5.697990	$2.10 \times 10^{-5}$
平均值	PSO	0.202546	0.211877	5.729951	$1.57 \times 10^{-3}$
	WOA	0.200852	0.200992	5.701104	$7.11 \times 10^{-4}$
	MIWOA	0.199494	0.199268	5.704161	$5.08 \times 10^{-5}$
最差值	PSO	0.214186	0.237701	5.877306	$2.00 \times 10^{-3}$
	WOA	0.187781	0.231605	5.785584	$1.47 \times 10^{-3}$
	MIWOA	0.197197	0.209352	5.735940	$9.02 \times 10^{-5}$

图 8 和图 9 分别是 WOA 和 MIWOA 对 Rössler 混沌系统的参数辨识曲线和适应度值曲线,从图中可以看到 MIWOA 的参数估计值可以快速收敛到真实值,适应度值迅速逼近 0,表明 MIWOA 高效的全局搜索能力和快速收敛速度。

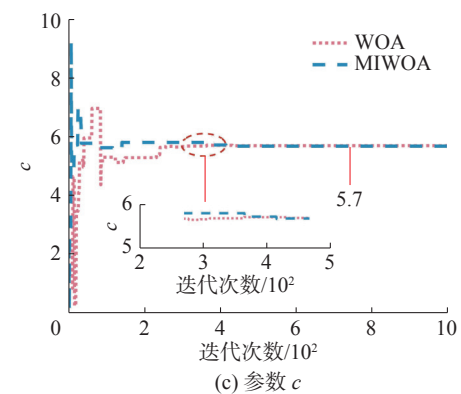
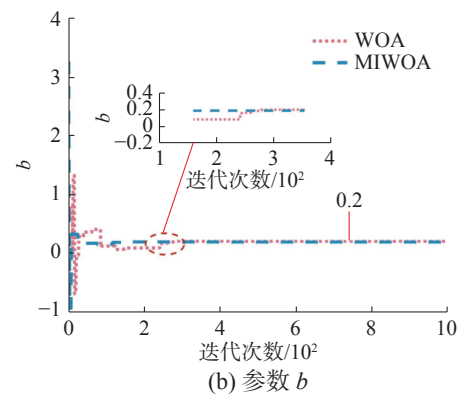
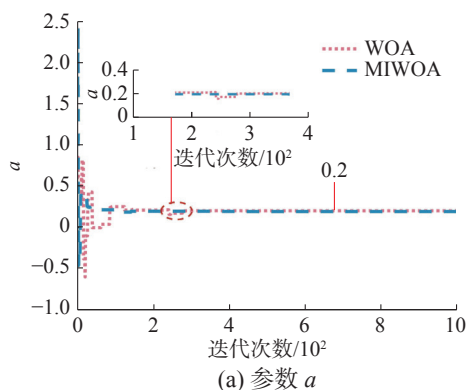


图 8 WOA 和 MIWOA 参数辨识曲线

Fig. 8 Parameter identification curves of WOA and MIWOA

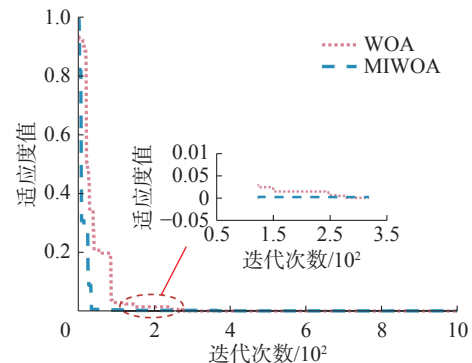


图 9 WOA 和 MIWOA 适应度值曲线

Fig. 9 Fitness value curves of WOA and MIWOA

#### 4.2 MIWOA 对 Lü 混沌系统的参数辨识

为了验证 MIWOA 对混沌系统参数辨识的普适性, 以 Lü 系统为例进行仿真, 表达式为<sup>[26]</sup>

$$\begin{cases} \dot{x} = a(y-x) \\ \dot{y} = -xz + by \\ \dot{z} = xy - cz \end{cases} \quad (22)$$

其中, 系统参数真实值为  $a = 30$ ,  $b = 22.2$ ,  $c = 8.8/3$  时, 系统为混沌状态, 其演化过程如图 10 所示。

令初值向量为  $[-1 \ 0 \ 1]^T$ , 使用 PSO、WOA 和 MIWOA 辨识 Lü 混沌系统 3 个参数未知时的情况, 3 种算法独立运行 20 次, 辨识结果如表 12 所示。

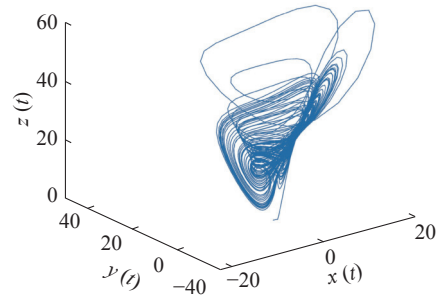


图 10 Lü 混沌系统动力学轨迹

Fig. 10 Dynamic track of Lü chaotic system

表 12 各算法的三维参数估计结果比较

Table 12 Comparison of three-dimensional parameter estimation results for each algorithm

评价指标	算法	$a$	$b$	$c$	$J$
最优值	PSO	30.212 604	22.248 850	2.924 835	$1.0056 \times 10^{-3}$
	WOA	30.024 252	22.265 087	2.932 120	$2.3900 \times 10^{-5}$
	MIWOA	29.999 427	22.139 015	2.935 886	$7.8900 \times 10^{-5}$
平均值	PSO	29.971 539	22.332 746	2.925 084	$1.4468 \times 10^{-3}$
	WOA	29.641 160	21.948 505	2.929 757	$7.6355 \times 10^{-3}$
	MIWOA	29.953 377	22.179 534	2.933 054	$1.6305 \times 10^{-4}$
最差值	PSO	28.718 457	22.784 762	2.900 586	$2.1256 \times 10^{-3}$
	WOA	25.867 088	24.974 265	2.903 107	$3.9871 \times 10^{-2}$
	MIWOA	29.643 899	22.443 769	2.926 306	$2.3000 \times 10^{-4}$

由表 12 中数据可知, 除了 MIWOA 的最优适应度值大于 WOA 的最优适应度值, MIWOA 的计算结果均优于其他 2 种算法, 说明 MIWOA 辨识混沌系统具有普适性。

## 5 结束语

本文以 WOA 为基础, 提出一种 MIWOA。通过分析初始种群分布, 使用 Chebyshev 混沌映射提高了初始种群质量。采用非线性收敛因子和自适应权重, 提高了算法全局和局部搜索能力。动态选择自适应  $t$  分布或蚁狮优化算法更新鲸鱼位置, 避免过早收敛。通过对 10 个基准函数和高维测试函数进行测试, 以及 Wilcoxon 秩和检验, 证明了 MIWOA 的优越性。

将 MIWOA 应用于 Rössler 和 Lü 混沌系统的参数辨识, 仿真结果优于 PSO 和 WOA, 验证了 MIWOA 辨识混沌系统参数的高效性。今后将继续研究 WOA 的优化策略, 将其应用到混沌系统控制与同步等其他领域。

**致谢** 本文作者衷心感谢德国阿尔弗雷德韦格纳研究所 Sergey Danilov 教授和王强博士在写作过程中提出的宝贵建议和意见!

## 参考文献:

- [1] 张昊, 王兴元. 复杂混沌系统同步及其应用 [M]. 北京: 科学出版社, 2021.
- [2] 郭荣伟. 混沌系统的控制问题研究及其应用 [M]. 北京: 科学出版社, 2020.
- [3] DING Zhenghao, LU Zhongrong, LIU Jike. Parameters identification of chaotic systems based on artificial bee colony algorithm combined with cuckoo search strategy[J]. *Science China technological sciences*, 2018, 61(3): 417–426.
- [4] 尚荣华, 焦李成, 刘芳, 等. 智能算法导论 [M]. 北京: 清华大学出版社, 2021.
- [5] CHEN Yang, PI Dechang, WANG Bi. Enhanced global flower pollination algorithm for parameter identification of chaotic and hyper-chaotic system[J]. *Nonlinear dynamics*, 2019, 97(2): 1343–1358.
- [6] AHANDANI M A, KHARRATI H. A corporate shuffled complex evolution for parameter identification[J]. *Artificial intelligence review*, 2020, 53(4): 2933–2956.
- [7] TURGUT M S, SAĞBAN H M, TURGUT O E, et al. Whale optimization and sine-cosine optimization algorithms with cellular topology for parameter identification of chaotic systems and Schottky barrier diode models[J]. *Soft computing*, 2021, 25(2): 1365–1409.
- [8] EBRAHIMI S M, MALEKZADEH M, ALIZADEH M,

- et al. Parameter identification of nonlinear system using an improved Lozi map based chaotic optimization algorithm (ILCOA)[J]. *Evolving systems*, 2021, 12(2): 255–272.
- [9] MIRJALILI S, LEWIS A. The whale optimization algorithm[J]. *Advances in engineering software*, 2016, 95: 51–67.
- [10] 包子阳, 余继周, 杨杉. 智能优化算法及其 MATLAB 实例 [M]. 3 版. 北京: 电子工业出版社, 2021.
- [11] 李士勇, 李研, 林永茂. 智能优化算法与涌现计算 [M]. 2 版. 北京: 清华大学出版社, 2022.
- [12] CHEN Hui, LI Weide, YANG Xuan. A whale optimization algorithm with chaos mechanism based on quasi-opposition for global optimization problems[J]. *Expert systems with applications*, 2020, 158: 113612.
- [13] CHEN Huiling, YANG Chenjun, HEIDARI A A, et al. An efficient double adaptive random spare reinforced whale optimization algorithm[J]. *Expert systems with applications*, 2020, 154: 113018.
- [14] CHAKRABORTY S, SAHA A K, CHAKRABORTY R, et al. An enhanced whale optimization algorithm for large scale optimization problems[J]. *Knowledge-based systems*, 2021, 233: 107543.
- [15] SHEN Ya, ZHANG Chen, GHAREHCHOPOGH F S, et al. An improved whale optimization algorithm based on multi-population evolution for global optimization and engineering design problems[J]. *Expert systems with applications*, 2023, 215: 119269.
- [16] DENG Huaijun, LIU Linna, FANG Jianyin, et al. A novel improved whale optimization algorithm for optimization problems with multi-strategy and hybrid algorithm[J]. *Mathematics and computers in simulation*, 2023, 205: 794–817.
- [17] YAN Aijun, GUO Jingcheng, WANG Dianhui. Robust stochastic configuration networks for industrial data modelling with Student's-t mixture distribution[J]. *Information sciences*, 2022, 607: 493–505.
- [18] AN Yining, WANG Jianzhou, LU Haiyan, et al. Research of a combined wind speed model based on multi-objective ant lion optimization algorithm[J]. *International transactions on electrical energy systems*, 2021, 31(12): e13189.
- [19] ZHANG Liping, ZHU Yue, REN Wei, et al. An energy-efficient authentication scheme based on Chebyshev chaotic map for smart grid environments[J]. *IEEE Internet of things journal*, 2021, 8(23): 17120–17130.
- [20] 黄元春, 张凌波. 改进的鲸鱼优化算法及其应用 [J]. *计算机工程与应用*, 2019, 55(21): 220–226, 270.  
HUANG Yuanchun, ZHANG Lingbo. Improved whale optimization algorithm and its application[J]. *Computer engineering and applications*, 2019, 55(21): 220–226, 270.
- [21] 陈强, 王宇嘉, 梁海娜, 等. 目标空间映射策略的高维多目标粒子群优化算法 [J]. *智能系统学报*, 2021, 16(2): 362–370.
- CHEN Qiang, WANG Yujia, LIANG Haina, et al. Multi-objective particle swarm optimization algorithm based on an objective space mapping strategy[J]. *CAAI transactions on intelligent systems*, 2021, 16(2): 362–370.
- [22] ZHOU Xinyu, WU Yanlin, ZHONG Maosheng, et al. Artificial bee colony algorithm based on adaptive neighborhood topologies[J]. *Information sciences*, 2022, 610: 1078–1101.
- [23] WANG Erlei, XIA Jiangying, LI Jia, et al. Parameters exploration of SOFC for dynamic simulation using adaptive chaotic grey wolf optimization algorithm[J]. *Energy*, 2022, 261: 125146.
- [24] 贾鹤鸣, 刘庆鑫, 刘宇翔, 等. 融合动态反向学习的阿奎拉鹰与哈里斯鹰混合优化算法 [J]. *智能系统学报*, 2023, 18(1): 104–116.  
JIA Heming, LIU Qingxin, LIU Yuxiang, et al. Hybrid Aquila and Harris Hawks optimization algorithm with dynamic opposition-based learning[J]. *CAAI transactions on intelligent systems*, 2023, 18(1): 104–116.
- [25] 刘成汉, 何庆. 融合多策略的黄金正弦黑猩猩优化算法 [J]. *自动化学报*, 2023, 49(11): 2360–2373.  
LIU Chenghan, HE Qing. Golden sine chimp optimization algorithm integrating multiple strategies[J]. *Acta automatica sinica*, 2023, 49(11): 2360–2373.
- [26] LIU Hongwei, HE Ping, LI Guodong, et al. Multi-directional annular multi-wing chaotic system based on Julia fractals[J]. *Chaos, solitons & fractals*, 2022, 165: 112799.

#### 作者简介:



潘悦悦, 博士研究生, 主要研究方向为智能优化算法。E-mail: panyueyue@ncepu.edu.cn。



吴立飞, 高级工程师, 博士, 主要研究方向为系统辨识、过程控制。主持教育部中央高校基本科研业务费专项基金 2 项。发表学术论文 30 余篇。E-mail: wulf@ncepu.edu.cn。



杨晓忠, 教授, 博士生导师, 主要研究方向为智能优化算法、人工智能。主持国家科技重大专项子课题 3 项、国家自然科学基金面上项目 2 项。发表学术论文 100 余篇。E-mail: yxiaoZh@ncepu.edu.cn。