

文章编号: 2095-2163(2020)09-0006-06

中图分类号: TP301

文献标志码: A

# 基于自适应决策算子的鲸鱼优化算法

徐航, 张达敏, 王依柔, 宋婷婷, 樊英

(贵州大学大数据与信息工程学院, 贵阳 550025)

**摘要:** 针对鲸鱼优化算法容易陷入早熟和收敛速度慢等缺陷, 本文提出一种基于自适应决策算子的鲸鱼优化算法, 考虑到算法初始种群也是影响算法性能的关键因素, 将混沌 Singer 映射用于优化算法初始阶段, 生成的初始种群, 使种群的分布更加均匀, 并且提出一个受个体适应度值影响的决策算子, 通过决策算子来决定整个种群的搜索方向, 一定程度上降低了最优区域对领导者的吸引, 同时加入了一个分段的自适应的权重来有效调节算法对最优解的勘探能力。通过对8个基准函数进行仿真实验, 用 CEC2014 测试函数来评估改进后鲸鱼优化算法的性能。实验结果表明, 无论是求解低维函数还是高维函数, 改进鲸鱼优化算法都具备一定的有效性。

**关键词:** 鲸鱼优化算法; 混沌 Singer 映射; 决策算子; 权重

## Whale optimization algorithm based on adaptive decision operator

XU Hang, ZHANG Damin, WANG Yirou, SONG Tingting, FAN Ying

(College of Big Data and Information Engineering, Guizhou University, Guiyang 550025, China)

**[Abstract]** Aiming at the shortcomings of premature whale optimization algorithm and slow convergence speed, a whale optimization algorithm based on adaptive decision operator is proposed. Considering that the initial population of the algorithm is also a key factor affecting the performance of the algorithm, the chaotic Singer map is used to optimize the initial algorithm, which makes the distribution of the population more even. And a decision operator that is affected by the individual fitness value is proposed. The decision operator determines the search direction of the entire population, which reduces the optimal area to the leader to a certain extent. At the same time, a segmented adaptive weight is added to effectively adjust the algorithm's ability to explore the optimal solution. Through simulation experiments on 8 benchmark functions, CEC2014 test function is used to evaluate the performance of the improved whale optimization algorithm. The experimental results show that the improved whale optimization algorithm has certain effectiveness in solving low-dimensional functions and high-dimensional functions.

**[Key words]** whale optimization algorithm; chaotic Singer map; decision operator; weight

## 0 引言

群体智能优化算法是通过模拟自然界中生物的群体规律来实现寻优的元启发式算法, 许多自然启发算法越来越受工程应用问题的青睐, 其实现过程简单且易于执行, 而且这些算法也具有一定绕过局部最优区域的能力<sup>[1]</sup>, 其中具有代表性的是源于座头鲸捕猎行为的鲸鱼优化算法 (Whale Optimization Algorithm, WOA)<sup>[2]</sup>。由于 WOA 拥有出色的性能, 被成功应用于许多优化问题中<sup>[3]</sup>, 但和大多数启发式算法一样, WOA 也是从一个随机的种群开始进行搜索, 由于寻优过程的随机性, 在勘探和开发之间保持适当的平衡是极其困难的, 而且在算法的搜索过程中不断改变领导者的位置, 导致算法在求解优化

问题时可能会迅速收敛到局部最优解而不是全局最优解, 最终导致解决方案的质量下降。为了提升该算法的性能, 有学者已经对 WOA 进行了改进研究, 提出了一种基于自适应权重和柯西变异的鲸鱼优化算法, 且表现出较高的寻优精度和稳定性<sup>[4]</sup>; 提出了混合遗传鲸鱼优化算法来优化频谱利用率和对抗恶意用户模拟授权用户的攻击行为, 结果表明该算法比现有的检测算法具有更高的性能<sup>[5]</sup>。

为了进一步深入研究 WOA, 笔者分别采用混沌思想、决策算子等方法对 WOA 进行改进, 并提出一种改进鲸鱼算法 (Improved Whale Optimization Algorithm, IWOA), 首先利用混沌理论的特点优化算法初始种群位置, 最大限度使初始种群分布均匀,

**基金项目:** 贵州省科学技术基金项目(黔科合基础[2020]1Y254)。

**作者简介:** 徐航(1995-), 男, 硕士研究生, 主要研究方向: 认知无线电、智能优化算法; 张达敏(1967-), 男, 博士, 教授, 主要研究方向: 计算机应用技术、智能算法、信号与信息处理等; 王依柔(1994-), 女, 硕士研究生, 主要研究方向: 认知无线电、智能优化算法; 宋婷婷(1995-), 女, 硕士研究生, 主要研究方向: 认知无线电、智能优化算法; 樊英(1995-), 女, 硕士研究生, 主要研究方向: 计算机应用技术、优化计算。

**通讯作者:** 张达敏 Email: 1203813362@qq.com

**收稿日期:** 2020-07-01

并在算法中加入一个决策算子,通过对领导者的位置进行优化,增加算法跳出局部最优区域的概率,进而增加算法的寻优精度,同时结合文中提出的自适应权重,使勘探和开发之间保持适当的平衡。在实验分析部分,通过对10个基准测试函数进行反复求解,并利用Wilcoxon检验、MAE等方法证明了改进算法的有效性。

## 1 鲸鱼优化算法

WOA是基于座头鲸捕食机理的元启发式优化算法,它模拟了座头鲸的捕猎模式,通过利用搜索种群在搜索空间中移动,针对适应度值大小选择候选解。一般来说,增加搜索种群的大小和迭代次数可以寻找到更好的解,但是会耗费算法的执行时间,在WOA中通过模拟鲸鱼群包围捕食过程、螺旋行进过程和搜寻猎物过程进行数学建模。具体描述如下:

(1)包围捕食过程。在WOA初始阶段,由于鲸鱼群暂时无法获得猎物的方位,所以通过每个个体的初始位置开始搜索,认为当前的最佳候选解就是已获得的最优解,通过不断交流,引导整个种群向着距离最优解最近的个体移动,而距离最优解最近的个体不断向最优解靠近,达到寻优的目的。在分配了最佳解决方案之后,其它代理尝试向最佳搜索代理更新其它代理的位置,通过不断的搜索,座头鲸会将猎物包围起来。包围猎物的过程可以用数学公式(1)和(2)表示:

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

$$D = |C \cdot X^*(t) - X(t)|. \quad (2)$$

其中, $D = |C \cdot X^*(t) - X(t)|$ ;  $X(t)$ 为个体的当前位置; $X^*(t)$ 为当前最优位置向量; $t$ 为当前迭代次数, $A$ 、 $C$ 定义为式(3)和式(4):

$$A = 2ar_1 - a, \quad (3)$$

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

其中, $r_1$ 和 $r_2$ 为 $[0,1]$ 的随机数; $a$ 为随迭代次数增加由2递减到0的参数; $T_{max}$ 为最大迭代次数。

(2)螺旋行进过程。根据座头鲸捕猎的方式,WOA中将泡泡网捕食的方式建模为收缩包围机制和螺旋行进策略,通过模仿座头鲸发现猎物的捕食过程,不断以旋转的方式靠近猎物。

值得注意的是收缩包围机制是通过在迭代过程中将 $a$ 的值从2线性降低到0来实现的,此时的 $A$ 是在 $[-a, a]$ 之间随机取值的,当 $|A| \leq 1$ 时,可得更新后的位置是在向最优位置移动的,通过改变算法参数的方式实现了对猎物的收缩包围。其中螺旋方

程(5)表示如下:

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

其中, $D' = |X^*(t) - X(t)|$ ,表示鲸鱼与猎物之间的距离; $b$ 是螺旋行进方程的常量,本文 $b = 1$ ;  $l$ 为 $[-1,1]$ 之间的随机数。由于鲸鱼的螺旋捕食者不仅在外环上移动,而且还缩小了包围圈,因此数学模型中有一半的概率会选择收缩包围方式和螺旋模型行进来更新鲸鱼的位置。数学公式(6)表示如下:

$$X(t+1) = \begin{cases} X^*(t) - A \cdot D, & p < 0.5; \\ D' \cdot e^{bl} \cdot \cos(2\pi l) + X^*(t), & p \geq 0.5. \end{cases} \quad (6)$$

其中, $p$ 为 $[0,1]$ 之间随机生成的随机数。

(3)搜索猎物过程。在搜索猎物的过程中,当参数 $A$ 满足 $|A| > 1$ 时,鲸鱼会通过当前自身的位置随机寻找猎物,利用这种位置更新方式,鲸鱼群可以离开自身当前所在区域,一定程度上提高了WOA的优化性能。数学公式(7)和(8)表示如下:

$$D_{rand} = |C \cdot X_{rand}(t) - X(t)|, \quad (7)$$

$$X(t+1) = X_{rand}(t) - A \cdot D. \quad (8)$$

其中, $X_{rand}$ 表示从群体中随机选择鲸鱼的位置向量。

## 2 自适应决策鲸鱼优化算法

### 2.1 混沌初始化种群

近几年出现的大多数生物启发算法都是基于随机种群的算法,鲸鱼优化算法也一样,通过概率分布来实现算法种群的随机分布,这种产生初始种群的方式则会导致种群多样性低,往往使寻优的位置不够广泛,利用混沌特性来提高WOA的收敛速度和搜索精度具有重要意义<sup>[6]</sup>,目前优化领域中有多种类型混沌映射,在大部分的研究中使用Tent映射和Logistic映射等进行种群初始化,Singer映射能产生相对均匀的混沌序列。因此,本文采用Singer映射来优化算法初始种群,数学公式描述为式(9):

$$x_{i+1} = P \cdot (7.86x_i - 23.31x_i^2 + 28.75x_i^3 - 13.302875x_i^4). \quad (9)$$

其中, $P \in (0.9, 1.08)$ ,本文取0.98。

在算法初始阶段,通过混沌映射生成位置分布较为合理的个体,扩大了算法中个体在空间中的勘探范围,一定程度上降低了空间中种群被局部极值吸引的可能性,从而提高了算法的性能。

### 2.2 决策算子

在WOA搜索方程中,种群的搜索方向对领导者有很大依赖关系,领先的鲸鱼有时会在局部解决

方案中陷入停滞,特别是在存在多峰的多模态问题中,在局部极值处的停滞是过早收敛的主要原因。鲸鱼个体之间的协作和信息交换机制有助于缓解这一问题,为了增强算法避免早熟能力,本文在算法中加入一个决策算子,将个体的适应度和位置加入到算法的搜索机制中,通过对领导者位置进行处理,加强种群之间的信息交流,进而提高算法的寻优性能,具体策略如式(10)~(12):

$$X_i^{t+1} = \begin{cases} M_i^{t+1}, & \text{if } r_3 < DP; \\ N_i^{t+1}, & \text{else.} \end{cases} \quad (10)$$

$$M_i^{t+1} = \frac{X_{lr1} + X_{lr2} + X_{lr3}}{3}, \quad (11)$$

$$N_i^{t+1} = X_{igbest} + k \cdot (X_1^t + X_2^t + X_3^t). \quad (12)$$

其中,  $X_{lr1}, X_{lr2}, X_{lr3}$  表示随机选择的第  $t$  代的 3 个个体位置;  $X_{igbest}$  表示全局最优位置,  $X_1^t, X_2^t, X_3^t$  表示第  $t$  代适应度值最好的前 3 个个体位置;  $k$  表示  $(0,1)$  之间的随机数。  $DP$  表示受个体适应度值影响决策概率,表示为式(13):

$$DP = \frac{1}{N} \sum_{i=1}^N C(t), \quad (13)$$

其中,  $N$  表示种群个体,  $C_{(t)}$  表示第  $t$  代优化成功的个体数,表示为式(14):

$$C(t) = \begin{cases} 1, & \text{if } fit_{i,t} < fit_{i,t-1} \\ 0, & \text{else.} \end{cases} \quad (14)$$

考虑到在鲸鱼优化算法中,每次迭代时将适应度值最好的位置赋值给鲸鱼领导者,这样的方式导致种群个体汇聚在局部最优区域,往往导致寻优精度较低,为了降低这种缺陷对算法的影响,本文将个体的适应度值纳入影响决策算子的因素中,加入决策算子后的算法领导者不是直接进行下一次迭代过程,而是记录每个个体的适应度值和位置,通过判断决策概率的大小选择筛选公式,通过比较适应度值的大小挑选出一个新的领导者位置,最后引入贪婪机制,比较两个领导者之间的适应度值大小,更新全局最优位置及适应度值,再进入到下一次的迭代过程,通过对最优位置的不断选择,对提高了算法的搜索精度和收敛速度具有重要意义。

### 2.3 分段自适应权重

WOA 与当前大多数元启发式算法一样,惯性权重对算法性能具有很明显的影响,在粒子群算法中已经证明,惯性权重反映的是后一个追随者摆脱前一个位置束缚的能力,较大的惯性权重会使算法拥有较好的全局搜索能力,反之,较小的惯性权重会使

算法拥有较好的局部开发能力<sup>[7]</sup>。通常情况下,较大的权重会使算法快速抵达最优区域附近,而在算法迭代后期,应该适当减小相邻两代之间的联系,此时权重应该较小,因此,本文在改进算法中提出分段自适应的惯性权重,式(15)和式(16):

$$\omega_1 = \omega_{min} + DP \cdot (\omega_{max} - \omega_{min}), \quad \frac{t}{T_{max}} \leq 0.5, \quad (15)$$

$$\omega_2 = \omega_{max} - (\omega_{max} - \omega_{min}) \times \frac{rank(r)}{N}, \quad \frac{t}{T_{max}} > 0.5. \quad (16)$$

其中,  $\omega_{max}$  表示最大惯性权重;  $\omega_{min}$  表示最小惯性权重,当  $\omega$  在  $[0.4, 0.9]$  之间变化时,算法具有最佳的寻优性能;  $DP$  表示算法中个体优化的成功率,  $rank(r)$  表示随机一个个体的秩序数;  $N$  表示种群大小。因此,在迭代的前半部分,权重受群体的优化成功率影响。随着迭代的进行,惯性权重在  $0.4 \sim 0.9$  之间动态变化,使算法的全局搜索和局部开发得到较好的平衡。在迭代的后半部分,随着种群的聚集,继续使用优化成功率将导致权重无法更新,于是采用随机个体的秩序数来更新权值,使迭代后期鲸鱼种群也拥有较大的搜索步长,一定程度上增加了算法跳出局部最优区域的概率。当  $t/T_{max} < 0.5$  时,加入权值的位置更新公式表示为(17)~(19)(同理可得  $t/T_{max} \geq 0.5$  时的位置更新公式):

$$X(t+1) = X^*(t) - \omega_1 \cdot A \cdot D \quad |A| < 1, p < 0.5, \quad (17)$$

$$X(t+1) = X_{rand} - \omega_1 \cdot A \cdot D_{rand} \quad |A| \geq 1, p < 0.5, \quad (18)$$

$$X(t+1) = D^l \cdot e^{bl} \cdot \cos(2\pi l) + \omega_1 \cdot X^*(t), p \geq 0.5. \quad (19)$$

### 2.4 IWOA 执行步骤

结合前文所述的改进策略,本文所提基于自适应决策算子的鲸鱼优化算法步骤如下:

**Step 1** 根据混沌 Singer 映射生成  $N$  个初始化个体,  $N$  为种群大小,设置最大迭代次数  $T_{max}$ 、 $\omega_{max}$ 、 $\omega_{min}$ ;

**Step 2** 由目标函数  $f(x)$  计算出种群中每个个体的目标函数值,并记录最优位置  $X(t)$ ,根据适应度值将  $N$  个个体排序,随机选择个体,记录其秩序数  $rank(r)$ ,同时分别与领导者适应度进行比较,统计  $C(t)$ ,更新决策概率  $DP$ ;

**Step 3** 更新算法中  $r_1, r_2, k$  等参数,同时更新  $a, A, C, l$  等,通过(15)(16)式更新参数  $\omega_1, \omega_2$ ;

**Step 4** 如果迭代次数  $t \leq 0.5T_{\max}$ , 通过比较  $|A|$  的大小, 用随机生成的  $p$  值与 0.5 作比较, 选择相应的位置更新公式, 若  $p < 0.5$  且  $|A| < 1$ , 则按照 (17) 式更新当前个体位置;

**Step 5** 若  $p < 0.5$  且  $|A| \geq 1$ , 则按照 (18) 式搜索猎物;

**Step 6** 若  $p \geq 0.5$ , 按照 (19) 式更新当前位置;

**Step 7** 将  $r_3$  与决策概率进行比较选择决策公式, 同时选择种群中适应度值最好的 3 个位置和随机的 3 个位置, 将保存下来的最优位置  $X^*$  按照 (11) 或 (12) 式进行位置筛选操作, 最后引入贪婪机制, 通过适应度值的大小筛选出一个新的领导者位置, 更新全局最优位置;

**Step 8** 若迭代次数  $t > 0.5T_{\max}$ , 将 (17) ~ (19) 式中  $\omega_1$  更新为  $\omega_2$ , 重复步骤 4 ~ 7;

**Step 9** 判断算法是否满足停止条件, 若达到则停止迭代, 并输出最优解和适应度值, 反之, 重复执行步骤 3~8。

### 3 仿真测试和分析

在对算法进行改进的过程中发现, 不同改进策略对算法的影响各不相同。为了分析本文提出的每个改进方法的效果, 将每个改进方法针对 WOA 进行改进, 分别对应为基于混沌 Singer 映射的鲸鱼优化算法 (Whale Optimization Algorithm Based on Chaotic Singer Map, CWOA)、基于分段自适应权重的鲸鱼优化算法 (Whale Optimization Algorithm Based on Piecewise Adaptive Weight, WWOA) 和基于决策算子的鲸鱼优化算法 (Whale Optimization Algorithm Based on Decision Operator, DWOA)、并将每一种改进算法与改进鲸鱼优化算法 (Improved Whale Optimization Algorithm, IWOA)、鲸鱼优化算法 (Whale Optimization Algorithm, WOA)、和灰狼算法 (Grey wolf optimizer, GWO) 同时在表 1 所示的 10 个基准测试函数下进行重复求解测试, 记录最优值、最差值、均值等评价指标。考虑到维度是影响求解性能的重要因素, 所以将各个测试函数的维度随机设置为 2~200 维, 从而验证算法求解低维和高维的能力。

本文所有测试均在 Window10 操作系统, Inter (R) Core(TM) i5-9400F、2.9GHz 主频、16G 内存的计算机上实现, 编程语言为 MATLABR2014b, 实验最大迭代次数  $T_{\max}$  设置为 1000, 种群大小  $N$  为 30, 实验次数为 30 次, 各算法的参数设置见表 2。

表 1 基准测试函数

Tab. 1 Benchmark functions

编号	函数名	维度	搜索区域	理论值
$f_1$	Sphere	100	$[-100, 100]$	0
$f_2$	Schwefel 2.22	50	$[-10, 10]$	0
$f_3$	Schwefel 2.21	30	$[-100, 100]$	0
$f_4$	Rosenbrock	60	$[-30, 30]$	0
$f_5$	Step	10	$[-100, 100]$	0
$f_6$	Quartic	40	$[-1.28, 1.28]$	0
$f_7$	Penalized1	150	$[-50, 50]$	0
$f_8$	Penalized2	70	$[-50, 50]$	0

表 2 算法主要参数

Tab. 2 Main parameters of algorithm

算法	主要参数
WOA	$a_{\max} = 2, a_{\min} = 0$
IWOA	$\omega_{\max} = 0.9, \omega_{\min} = 0.4, k = r_3 = rand, P = 0.98$
DWOA	$k = r_3 = rand,$
WWOA	$\omega_{\max} = 0.9, \omega_{\min} = 0.4$
CWOA	$P = 0.98, x(1) = 0.71$
GWO	$a_{\max} = 2, a_{\min} = 0$

首先, 为了比较本文提出的几种改进策略是否对 WOA 的性能有所提升, 将 CWOA、DWOA、WWOA 中的参数统一设置为 IWOA 对应的参数, 分析表 3 中的最优值、最差值和均值等实验结果可知, 所提出的改进方法相比于原始 WOA 的寻优性能均有大幅度的提升, 但是每个改进策略对算法的影响存在差异, 其中基于分段自适应权重的改进方法对算法的影响程度是最大的, 说明原始算法中个体摆脱局部最优区域的能力较弱, 也证明了本文所引进权重的有效性; 对比加入权重的改进方法, 基于决策算子的改进方法对算法的影响程度较弱, 但与原始算法相比, DWOA 寻优精度均有很大程度的提升, 说明鲸鱼个体之间的协作和信息交换机制一定程度上缓解了算法早熟的问题, 使鲸鱼群可以在搜索空间中搜索更广阔的区域; 相比于其他二种改进方法, 基于混沌理论的种群初始化改进方法是最差的, 当与原始算法相比, CWOA 也有显著的改进效果, 说明利用混沌特性对初始种群进行优化使得个体在空间中分布更加均匀, 从而搜索到的解的质量也更好。从整体来看, 改进过程是以决策算子和自适应权重为主导, 以混沌 Singer 映射的种群初始化为辅助进行的, 所以融合 3 种改进策略的 IWOA 比每一个种改进策略的寻优效果都要好, WWOA 在求解部分函数时也能达到和 IWOA 同样的效果, 说明此时该函数对权重更加敏感, 而伴随着维度的增加, 大部分的函数评价指标均证明融合每个改进策略的 IWOA 是更加优秀的。

其次, 由于评价指标中的标准差常反映算法的鲁棒性, 分析表 3 中的标准差可以发现包含每个改

表3 测试函数结果对比

Tab. 3 Comparison of test function results

算法	最优值	最差值	均值	标准差	耗时/s	
$f_1$	WOA	8.47E-169	7.49E-1462	5.0E-147	1.37E-146	2.8516
	IWOA	0.00E+00	0.00E+00	0.00E+00	0.00E+00	5.2120
	DWOA	6.28E-165	2.14E+05	8.79E+03	3.98E+04	4.8565
	CWOA	3.44E-166	4.89E-1513	6.5E-152	9.97E-152	3.2564
	WWOA	0.00E+00	6.09E-03	2.03E-04	1.11E-03	4.9380
$f_2$	GWO	3.14E+30	2.10E-28	2.61E-29	4.09E-29	0.6024
	WOA	2.60E-115	1.72E-1011	0.1E-102	3.73E-102	1.5517
	IWOA	0.00E+00	3.60E-2961	6.7E-297	0.00E+00	2.8217
	DWOA	1.24E-112	7.09E-1032	6.3E-104	1.29E-103	2.5564
	CWOA	4.10E-116	2.50E-1019	1.0E-103	4.56E-102	2.1439
$f_3$	WWOA	0.00E+00	3.60E-2861	6.7E-287	0.00E+00	2.5223
	GWO	7.71E-27	1.33E-25	5.25E-26	3.40E-26	0.4019
	WOA	4.01E-01	9.27E+01	3.41E+01	2.70E+01	1.6331
	IWOA	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.0433
	DWOA	2.42E-04	2.28E-01	4.83E-01	6.86E-01	1.6356
$f_4$	CWOA	5.95E-08	3.93E-01	3.45E-01	8.05E-01	1.8250
	WWOA	7.06E-270	4.13E-2401	3.8E-241	0.00E+00	1.8819
	GWO	7.27E-16	1.44E-13	2.31E-14	3.50E-14	0.3334
	WOA	4.19E-07	5.24E-06	6.41E-06	1.82E-06	0.3363
	IWOA	8.49E-12	2.50E-07	2.04E-07	2.19E-07	1.3280
$f_5$	DWOA	7.96E-10	4.41E-06	9.90E-07	1.02E-06	0.7388
	CWOA	2.51E-10	4.91E-06	2.17E-07	1.63E-06	0.5100
	WWOA	4.61E-11	4.50E-05	9.72E-07	1.78E-07	0.6353
	GWO	4.65E-09	9.59E-07	2.44E-07	2.70E-07	0.2639
	WOA	1.41E-02	2.73E-01	7.27E-02	5.42E-02	0.4094
$f_6$	IWOA	1.90E-06	1.80E-05	6.40E-06	3.28E-06	0.9645
	DWOA	7.98E-05	1.29E-04	4.29E-04	2.35E-04	0.7110
	CWOA	4.72E-03	1.41E-02	2.60E-03	3.27E-03	0.7354
	WWOA	3.11E-06	4.13E-04	1.13E-04	7.06E-04	0.8974
	GWO	3.90E-02	2.02E-01	8.52E-02	3.11E-02	0.2884
$f_7$	WOA	8.20E-03	8.99E-01	1.69E-01	1.88E-01	1.1586
	IWOA	8.18E-07	1.68E-06	5.90E-06	3.05E-06	2.7570
	DWOA	2.75E-05	6.91E-03	2.50E-04	1.26E-03	2.3511
	CWOA	3.61E-03	6.98E-03	1.42E-03	1.84E-03	1.7095
	WWOA	7.08E-06	2.51E-04	3.99E-05	4.99E-05	2.1386
$f_8$	GWO	2.73E-02	2.32E-01	8.72E-01	5.01E-01	0.6284
	WOA	8.60E-02	4.16E-01	2.15E-01	8.30E-01	4.0078
	IWOA	1.43E-04	2.22E-03	1.25E-03	4.41E-03	15.7020
	DWOA	3.71E-03	7.68E-01	2.10E-02	1.74E-02	9.0136
	CWOA	9.29E-03	3.63E-01	2.06E-02	7.47E-02	8.6731
$f_8$	WWOA	1.67E-04	4.37E-02	4.16E-02	1.79E-02	11.5088
	GWO	9.69E-01	6.90E-01	7.72E-01	6.37E-01	2.4183
	WOA	1.73E+01	1.36E+01	2.17E+01	3.12E+01	1.4290
	IWOA	9.17E-03	1.08E-01	3.29E-01	2.93E-01	3.4709
	DWOA	2.54E-02	3.11E-01	1.04E+00	5.67E-01	2.4032
$f_8$	CWOA	2.37E+00	2.80E+00	1.16E+00	6.54E+00	2.1103
	WWOA	2.83E-01	1.81E+00	4.08E+00	3.90E-01	2.6270
	GWO	2.76E+00	4.14E+01	3.53E+01	3.35E+01	1.3613

进策略的算法都比 WOA 更稳定,而包含所有改进方法的 IWOA 的标准差都比其他算法要好,从而说明 IWOA 具有一定的稳定性。从平均耗时来看, GWO 平均耗时最短,改进方法的 CWOA、DWOA 和

WWOA3 种算法的平均耗时相对于 WOA 有所增加,因为 3 种算法都是在 WOA 的基础上加入一种改进策略,但总的来说,综合 3 种改进策略的 IWOA 耗时是最长的,因为 IWOA 可以引导搜索更广阔的区域,导致平均耗时有增加。

再次,为了直观体现改进算法的优势,图 1 为本文选取的部分基准测试函数运行 30 次的平均收敛曲线,从收敛曲线可以看出,在 8 个测试函数的求解过程中, IWOA 相比于参与对比的其他算法,具有更快的收敛速度和收敛精度,而且可以发现 IWOA 可以快速搜索到最优解,证明 IWOA 摆脱局部最优区域的能力比其他算法更强。

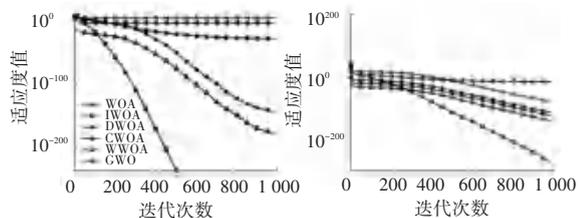
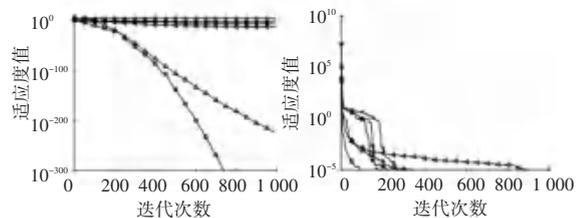
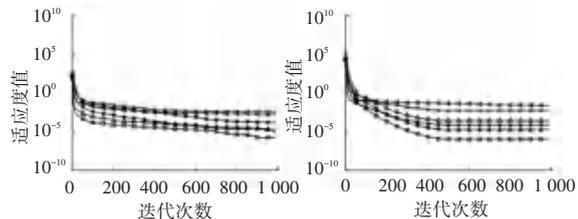
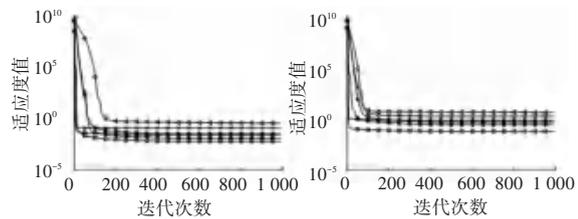
(a)  $f_1$  平均收敛曲线(b)  $f_2$  平均收敛曲线(a)  $f_1$  average convergence curve (b)  $f_2$  average convergence curve(c)  $f_3$  平均收敛曲线(d)  $f_4$  平均收敛曲线(c)  $f_3$  average convergence curve (d)  $f_4$  average convergence curve(e)  $f_5$  平均收敛曲线(f)  $f_6$  平均收敛曲线(e)  $f_5$  average convergence curve (f)  $f_6$  average convergence curve(g)  $f_7$  平均收敛曲线(h)  $f_8$  平均收敛曲线(g)  $f_7$  average convergence curve (h)  $f_8$  average convergence curve

图1 测试函数平均收敛曲线

Fig. 1 Average convergence curve of test function

为了进一步评估 WOA 改进策略的鲁棒性和竞争性, 本文选取了 CEC2014 单目标优化函数中部分单峰、多峰、混合和复合类型的函数进行重复求解, 选取的部分函数见表 4, 其中种群大小为 30, 最大迭代次数为 3000, 维度为 30。

表 5 为部分 CEC2014 函数独立重复运行 30 次得到的平均值和标准差, 分析选取的部分函数的实验结果可知, 相比于参与对比的其他几个算法, 特别是在求解复杂类型函数的时候, IWOA 表现出良好的求解能力, 证明 IWOA 具有一定跳出局部最优区域的能力, 而对于每一种改进方法, 只有基于混沌 Singer 映射的改进算法表现较为平庸, 但相比于

WOA, 都有较大程度的提升, 其他每一个改进算法均表现优异, 从而进一步证明本文提出的改进策略具有较好的寻优能力和鲁棒性。

表 4 CEC2014 基准函数

Tab. 4 CEC 2014 benchmark function

函数	维度	特征	定义域	最优值
CEC03	30	UN	[-100, 100]	300
CEC06	30	MN	[-100, 100]	600
CEC017	30	HF	[-100, 100]	1700
CEC19	30	HF	[-100, 100]	1900
CEC23	30	CF	[-100, 100]	2300
CEC24	30	CF	[-100, 100]	2400

表 5 CEC2014 优化结果对比

Tab. 5 Comparison of optimization results of CEC 2014

函数	指标	IWOA	CWOA	DWOA	WWOA	WOA	GWO
CEC03	Mean	4.4667E+02	3.2374E+03	9.2374E+02	6.4667E+02	3.2374E+04	2.5277E+04
	Std	1.8230E+02	2.2657E+03	1.4657E+03	4.8230E+02	2.2657E+04	7.5968E+03
CEC06	Mean	6.5165E+02	4.2265E+02	8.4138E+02	5.1396E+02	9.6279E+02	5.8644E+02
	Std	3.4596E+02	8.3561E+01	7.5598E+02	2.2570E+02	3.3681E+01	3.5843E+00
CEC17	Mean	1.9690E+03	1.0928E+03	2.6751E+03	1.3549E+03	4.6843E+06	1.7059E+06
	Std	2.3014E+03	8.2867E+02	3.5456E+03	9.2609E+02	3.3583E+06	1.5545E+06
CEC19	Mean	2.3551E+03	3.5377E+03	3.1226E+03	2.9287E+03	2.3729E+03	2.3776E+03
	Std	3.5945E+03	2.8443E+02	4.6983E+03	4.9302E+03	3.6296E+01	1.3304E+01
CEC23	Mean	3.1217E+03	3.5510E+03	4.0737E+03	3.5803E+03	4.4286E+03	3.1436E+03
	Std	2.8596E+03	3.7682E+03	1.4184E+03	4.1102E+03	6.4227E+01	3.3561E+01
CEC24	Mean	3.3824E+03	4.3168E+03	5.1307E+03	3.9464E+03	3.1949E+03	3.1949E+03
	Std	3.6528E+03	6.8959E+02	9.9672E+02	3.4701E+03	4.4479E+00	9.8764E-01

#### 4 结束语

针对鲸鱼优化算法的不足, 本文首先采用混沌 Singer 映射来优化算法中的初始种群, 一定程度上加快了算法的收敛速度; 提出了一个决策算子, 通过对领导者位置的筛选, 提高了算法的寻优精度; 最后, 在算法中加入自适应惯性权重, 平衡了算法的全局搜索和局部开发能力。经过部分基准测试函数的求解实验, 证明了改进算法具有优秀的寻优能力和鲁棒性。

#### 参考文献

[1] RAJABIOUN R. Cuckoo Optimization Algorithm[J]. Applied Soft Computing, 2011, 11(8): 5508-5518.

[2] MIRJALILI S, LEWIS A. The Whale Optimization Algorithm[J]. Advances in Engineering Software, 2016, 95(5): 51-67.

[3] SARATH K S, SEKAR S. Modeling and optimal design of LLC

resonant converter using whale optimization algorithm [J]. International Journal of Modeling, Simulation, and Scientific Computing, 2018, 09(6): 3075-3086.

[4] GUO Z Z, WANG P, MA Y F, et al. Whale optimization Algorithm Based on Adaptive Weight and Cauchy Mutation [J]. Microelectronics & Computer, 2017, 34(9): 20-25.

[5] ELGHAMRAWY S M, HASSANIEN A E. GWOA: a hybrid genetic whale optimization algorithm for combating attacks in cognitive radio network [J]. Ambient intelligence, 2019, 10(11): 4345-4360.

[6] ARORA S, ANAND P. Chaotic grasshopper optimization algorithm for global optimization [J]. Neural Computing and Applications, 2019, 31(8): 4385-4405.

[7] ARAFA D, MABROUK A, SAID L A, et al. Chaotic Flower Pollination and Grey Wolf Algorithms for parameter extraction of bio-impedance models [J]. Applied Soft Computing, 2019, 75: 750-774.