Software Engineering and Applications 软件工程与应用, 2024, 13(2), 281-293
Published Online April 2024 in Hans. https://www.hanspub.org/journal/sea
https://doi.org/10.12677/sea.2024.132028

Hans汉斯

自适应沙丘猫鲸鱼优化算法

朱美芬，王联国*

甘肃农业大学信息科学技术学院，甘肃 兰州

收稿日期：2024年3月5日；录用日期：2024年4月22日；发布日期：2024年4月30日

摘 要

鲸鱼优化算法原理简单、参数较少、全局搜索能力强，但在迭代后期易陷入局部最优且求解精度较低。本文将自适应收敛因子策略和沙丘猫群优化算法中随机搜索策略引入到鲸鱼优化算法中，提出了一种自适应沙丘猫鲸鱼优化算法。首先，采用自适应收敛因子策略，动态地调整算法参数a，使搜索更具连续性、稳定性与多样性，提高优化精度；其次，将沙丘猫群优化算法中引入到鲸鱼优化算法中，防止算法陷入局部最优，提高全局搜索能力；然后，通过基准测试函数进行仿真实验，并与其他几种智能群体算法进行比较，仿真实验结果表明，改进算法具有较高的优化性能；最后，利用改进算法求解机械工程优化问题，验证了改进算法的有效性和实用性。

关键词

鲸鱼优化算法，沙丘猫优化算法，收敛因子，工程优化

An Adaptive Sand Cat and Whale Optimization Algorithm

Meifen Zhu, Lianguo Wang*

College of Information Sciences and Technology, Gansu Agricultural University, Lanzhou Gansu

Received: Mar. 5th, 2024; accepted: Apr. 22nd, 2024; published: Apr. 30th, 2024

Abstract

The whale optimization algorithm boasts a simple principle, fewer parameters, and strong global search ability; however, it tends to fall into local optimum and yields low solution accuracy in the later stages of iteration. In this paper, we introduce the self-adaptive convergence factor strategy and random search strategy from the sand cat swarm optimization algorithm into the whale optimization algorithm, proposing an adaptive sand cat whale optimization algorithm. Firstly, the self-adaptive convergence factor strategy is employed to dynamically adjust the algorithm para-

*通讯作者。

文章引用: 朱美芬, 王联国. 自适应沙丘猫鲸鱼优化算法[J]. 软件工程与应用, 2024, 13(2): 281-293.
DOI: 10.12677/sea.2024.132028

meter a, enhancing the continuity, stability, and diversity of the search, and improving optimization accuracy. Secondly, the sand cat swarm optimization algorithm is incorporated into the whale optimization algorithm to prevent it from getting trapped in local optimum and enhance its global search ability. Subsequently, simulation experiments are conducted using benchmark test functions and compared with several other intelligent swarm algorithms. The simulation experiment results demonstrate that the improved algorithm exhibits superior optimization performance. Finally, the improved algorithm is applied to solve optimization problems in mechanical engineering, validating its effectiveness and practicality.

Keywords

Whale Optimization Algorithm, Sand Cat Optimization Algorithm, Convergence Factor, Engineering Optimization

1. 引言

智能群体算法是通过模拟自然界中的群体行为而衍生出的一种简单灵活，并且能够有效解决复杂优化问题的优化算法。

2016 年 Mirjaili 等人[1]提出了一种群体智能优化算法——鲸鱼优化算法(Whale Optimization Algorithm)，该算法通过模拟鲸鱼捕食时的泡泡网行为特征，采用收缩包围机制和螺旋更新位置的方式来模拟鲸鱼群体的包围、追捕和攻击猎物的过程，由此来实现优化搜索的目标。这类算法借鉴了仿生学原理，将鲸鱼的行为特征应用于解决优化问题的过程中。鲸鱼优化算法在执行过程中需要平衡探索新解空间和对已有解进行优化，以全面涵盖潜在最优解，增强全局搜索能力，并有效应对复杂情况。该算法具备设计简洁、参数少、全局搜索功能良好等优点。但鲸鱼优化算法也存在难以跳出局部最优、收敛速度慢、易早熟等问题，因此，国内外学者对 WOA 进行了相关的研究改进。其中，Chakraborty 等人[2]提出一种饥饿搜索的鲸鱼优化算法，将饥饿状态下的群体行为和座头鲸的狩猎行为相结合，避免算法陷入局部最优，提高算法的全局搜索能力和收敛速度；Strumberger 等人[3]提出一种混合算法(WOA-AEFS)，将 WOA 与人工蜂群(ABC) [4]算法、萤火虫算法(FA) [5]结合，保持种群多样性，避免算法陷入局部最优解，提高了全局搜索能力和收敛速度。Lee 等人[6]通过将遗传算法和热交换优化算法与鲸鱼优化算法相融合，利用遗传算法的强探索能力和热交换优化算法的强开发能力，提升算法的优化性能；Garip 等人[7]采用从 5 个不同的混沌映射中获得的序列替换 WOA 中的随机参数，扩展了种群的搜索范围，提高了算法的寻优精度。

针对鲸鱼优化算法在迭代后期易陷入局部最优、收敛速度慢和求解精度低等问题，本文引入沙丘猫算法[8]，提出一种自适应沙丘猫鲸鱼优化算法(An adaptive sand cat and whale optimization algorithm, SCWOA)。首先，引入自适应收敛因子，动态地调整算法参数，增加搜索的连续性与稳定性，提高搜索效率和优化精度；然后，将沙丘猫群优化算法中随机搜索策略和鲸鱼优化算法交替运行，提升全局搜索能力；最后，通过仿真实验验证算法改进的有效性并用 SCWOA 求解工程优化问题。

2. 鲸鱼优化算法

鲸鱼优化算法是一种基于自然界中鲸鱼觅食行为的启发式优化算法。自然界中座头鲸群体庞大，喜

欢捕食靠近水面的小鱼群和磷虾，鲸鱼优化算法利用群体捕食行为来进行优化搜索，这种捕食行为主要有包围猎物、泡泡网攻击和搜寻猎物三个阶段。

2.1. 包围捕食

鲸鱼优化算法的核心思想是通过群体内个体之间的信息交流和随机游走来识别目标并进行包围。将群体中最优位置作为目标时，其他个体会围绕最优个体移动。通过式(1)来更新位置：

$$D = \left| C \cdot X^*(t) - X(t) \right| \tag{1}$$

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

$$a = 2 - \frac{2t}{t_{\max}} \tag{3}$$

$$A = 2a \cdot r_1 - a \tag{4}$$

$$C = 2 \cdot r_2 \tag{5}$$

其中，t 为当前迭代次数；X^* 为猎物位置；X 为当前位置；A 和 C 为系数向量用来调节距离；r_1 和 r_2 为[0, 1]之间的随机向量；$t_{\max}$ 为最大迭代次数；a 为收敛因子，随着迭代次数增加从 2 线性递减到 0。

2.2. 气泡网捕食

当鲸鱼发现猎物时，它们会吸入大量的水，下潜约 12 米，通过在猎物周围形成螺旋状气泡带，同时向水面游动的方式来捕食猎物。它们利用这种向上的螺旋运动，逐渐缩小包围圈，以捕获猎物。在鲸鱼优化算法中，利用收缩包围机制和螺旋更新位置来描述这一捕食行为。

$$X(t+1) = X^*(t) + D' \cdot e^{bl} \cdot \cos(2\pi l) \tag{6}$$

其中，b 用来定义对数螺旋的形状；$D' = \left| X^*(t) - X(t) \right|$ 表示鲸鱼与当前全局最优个体之间的距离；l 是[−1, 1]的随机数。

鲸鱼围绕着猎物收缩圈螺旋游动，在收缩包围和螺旋更新中选择相同概率更新鲸鱼的位置，其数学模型如式(7)所示：

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

其中，p 是[0, 1]之间的随机数。

2.3. 随机搜索

鲸鱼群通过根据彼此位置进行随机搜索，在收缩圈外游动，以扩大搜索范围并最终达到全局寻优的目的，数学模型如式(8)和式(9)所示：

$$D = \left| C \cdot X_{rand} - X \right| \tag{8}$$

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

其中，X_{rand} 是从当前种群中选择的一个随机位置向量。

3. 自适应沙丘猫鲸鱼优化算法

3.1. 收敛因子指数衰减策略

鲸鱼优化算法在寻优过程中需要平衡其探索和开发能力，以取得更好的寻优结果。由式(3)、(4)可知，

参数 A 的值随着收敛因子 a 的变化而不断变化，即寻优过程中，WOA 的全局探索和局部开发能力的转换是通过调整收敛因子 a 来实现的。事实上，WOA 的寻优过程极为复杂，控制参数 a 线性递减策略不能适应寻优过程的实际情况，可能会导致收敛精度不高或易陷入局部最优，在求解高维多峰函数优化问题时易陷入局部最优。基于上述考虑，本文提出一种指数衰减策略，其具体表达式为：

$$a = 2 - 2 \cdot \left(\frac{t}{t_{\max}} \right)^{\frac{1}{2}} \tag{10}$$

上式使得 a 的取值在迭代过程中呈指数下降，使得算法更加稳定和收敛。这有助于算法在搜索空间中进行平滑的变化，更稳定地调整搜索步长，避免因为参数变化过快而导致突然跳变或不连续的情况，降低不稳定性，也使得算法可以更好地覆盖整个搜索空间，具有更好的探索性。改进前后收敛因子 a 的变化曲线图如图 1 所示。

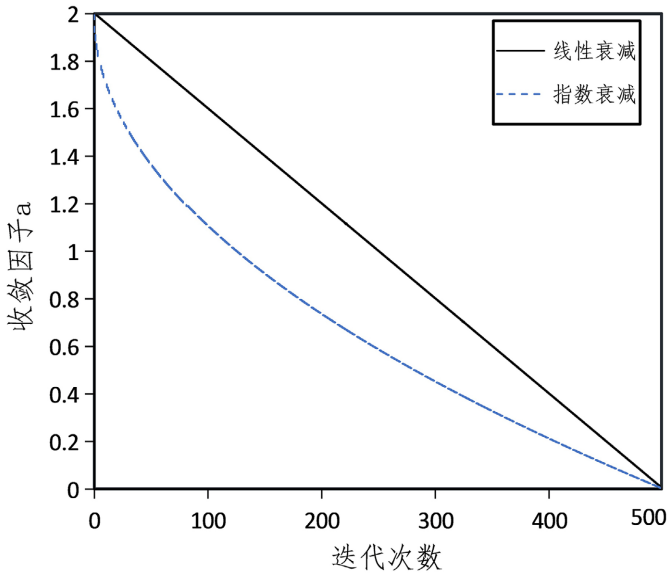

Figure 1. Comparison of the change curve of convergence factor a
图 1. 收敛因子 a 变化曲线对比

3.2. 混合沙丘猫策略

3.2.1. 沙丘猫群优化算法

沙丘猫群优化算法(Sand Cat swarm optimization, SCSO)是土耳其学者 Amir Seyyedabbasi [8]于 2022 年最新提出的一种模拟沙丘猫生存行为的元启发式算法。沙丘猫能够探测低于 2 kHz 的低频，也具有难以置信的挖掘猎物的能力。沙丘猫群优化模拟了沙丘猫的两个主要行为：搜寻猎物和攻击猎物。还提出一种用于平衡探索和利用的机制。其中，沙丘猫的猎物搜索机制依赖于低频噪声发射。

在搜索猎物的过程中，假设沙丘猫的听力灵敏范围 2 kHz，沙丘猫常规灵敏范围将随着迭代过程从 2 线性的降为 0。每只沙丘猫会根据最优解、自己当前的位置和 $\vec{r}$ 来更新自己位置。位置更新公式如下：

$$\overrightarrow{\mathrm{Pos}}(t+1) = \vec{r} \cdot \left(\overrightarrow{\mathrm{Pos}_{\mathrm{bc}}}(t) - rand(0,1) \cdot \overrightarrow{\mathrm{Pos}_{\mathrm{c}}}(t) \right) \tag{11}$$

其中，$\overrightarrow{\mathrm{Pos}_{\mathrm{bc}}}$ 为最佳候选位置，$\overrightarrow{\mathrm{Pos}_{\mathrm{c}}}$ 为当前位置，$\vec{r}$ 为灵敏度范围。

假定沙丘猫的灵敏度范围是一个圆，通过轮盘赌法在圆上选取一个随机角度 θ，以此确定移动方向。

根据式(12)和式(13)计算得出攻击猎物阶段中最优位置与当前位置。

$$\overrightarrow{\mathrm{Pos}_{\mathrm{rand}}}=\left|rand\left(0,1\right)\cdot\overrightarrow{\mathrm{Pos}_{\mathrm{b}}}\left(t\right)-\overrightarrow{\mathrm{Pos}_{\mathrm{c}}}\left(t\right)\right| \tag{12}$$

$$\overrightarrow{\mathrm{Pos}}\left(t+1\right)=\overrightarrow{\mathrm{Pos}_{\mathrm{b}}}\left(t\right)-\vec{r}\cdot\overrightarrow{\mathrm{Pos}_{\mathrm{rand}}}\cdot\cos\left(\theta\right) \tag{13}$$

其中，$\overrightarrow{\mathrm{Pos}_{\mathrm{b}}}$ 为最优解。

探索和利用阶段是通过自适应的 $\vec{r}$ 和 R 保证，$\vec{r}$ 随着迭代过程从 2 线性地降低为 0，以逐渐靠近猎物而不会丢失或跳过，参数 R 是区间 $\left[-2\vec{r},2\vec{r}\right]$ 的随机值，波动范围会随 $\vec{r}$ 变小。当 R 的随机值为[−1, 1]时，沙丘猫的下一个位置可以是当前位置与狩猎位置之间的任意位置。SCSO 算法在$|R|$小于或等于 1 时攻击猎物，否则进行探索和寻找新的猎物，其数学模型如式(14)所示：

$$\vec{X}\left(t+1\right)=\begin{cases}\overrightarrow{\mathrm{Pos}_{\mathrm{b}}}\left(t\right)-\vec{r}\cdot\overrightarrow{\mathrm{Pos}_{\mathrm{rand}}}\cdot\cos\left(\theta\right) & \left|R\right|\le 1\\ \vec{r}\cdot\left(\overrightarrow{\mathrm{Pos}_{\mathrm{bc}}}\left(t\right)-rand\left(0,1\right)\cdot\overrightarrow{\mathrm{Pos}_{\mathrm{c}}}\left(t\right)\right) & \left|R\right|>1\end{cases} \tag{14}$$

3.2.2. 混合沙丘猫

将 SCSO 中随机搜索策略和参数引入 WOA，更好地在解空间中进行全局搜索，提高找到全局最优解的概率。

根据生成的随机数判断此次迭代使用沙丘猫算法还是鲸鱼算法来更新位置，两种算法交替进行，以增加搜索的多样性和全局探索能力。通过这种交替的方式，两种算法可以相互补充，提高整体优化性能，从而更好地适应不同问题。

3.3. SCWOA 算法流程

SCWOA 算法流程如下：

步骤一：随机初始化一组鲸鱼群体，进入迭代循环，当迭代次数 t 小于最大迭代次数时，执行以下步骤。

步骤二：对当前位置进行边界检查，并计算适应度值，更新全局最优解和最优位置。

步骤三：生成随机数选择通过沙丘猫优化还是鲸鱼优化两种方式进行位置更新。

步骤四：进入沙丘猫算法阶段的概率为 q，使用轮盘赌法选择随机角度，通过随机搜索对每个个体进行位置更新；

步骤五：进入鲸鱼优化算法阶段的概率为 1 − q，计算参数 A、C，通过生成的随机概率 p 选择更新方式来更新位置；

步骤六：迭代次数加 1，重复步骤二到步骤五，直到达到最大迭代次数；

最终，输出最优适应度值、最优位置。

3.4. SCWOA 算法时间复杂度

SCWOA 算法的时间复杂度主要由沙丘猫优化和鲸鱼优化两部分组成，假设种群规模为 N，问题维度为 D，最大迭代次数为 T，执行沙丘猫优化的概率为 q，执行鲸鱼优化的概率为 1 − q，则沙丘猫优化部分的时间复杂度为 $O\left(q\cdot N\cdot D\cdot T\right)$，鲸鱼优化部分的时间复杂度为 $O\left(\left(1-q\right)\cdot N\cdot D\cdot T\right)$，因此，SCWOA 算法的时间复杂度为：

$$O\left(\mathrm{SCWOA}\right)=O\left(q\cdot N\cdot D\cdot T\right)+O\left(\left(1-q\right)\cdot N\cdot D\cdot T\right)=O\left(N\cdot D\cdot T\right) \tag{15}$$

WOA 的算法时间复杂度为：

$$O(\mathrm{WOA}) = O(N \cdot D \cdot T) \tag{16}$$

3.5. 仿真函数测试

此次仿真实验基于 Windows10 (64 bit)操作系统，处理器 Intel (R) Core (TM) i7-7500U CPU 编程采用 MATLAB R2021a 软件。

为验证本文 SCWOA 的性能，引入 13 个基准测试函数进行实验(具体函数表达式见参考文献[9])，其中，F1~F7 为单峰函数(其中 F5 在维度大于 3 时为多峰函数，F6 为非连续函数)，F8~F13 为多峰函数。实验中，算法种群规模为 30，维度为 30，最大迭代次数为 500，独立运行 30 次后的 Min (最小值)、Max (最大值)、Ave (平均最优值)和 Std (标准差)为最终测试结果，其中，平均最优值体现算法的收敛速度和求解准确度，标准差体现算法的稳定性和健壮性，最大值与最小值体现可行解的质量。实验结果如表 1 所示。

Table 1. Simulation function test results of WOA and SCWOA
表 1. WOA 与 SCWOA 的仿真函数测试结果

Function	Algorithm	Min	Max	Ave	Std
F1	WOA	7.09E−84	1.30E−70	4.32E−72	2.37E−71
	SCWOA	3.30E−128	6.78E−87	2.18E−106	1.19E−105
F2	WOA	4.56E−60	2.88E−48	1.05E−49	5.26E−49
	SCWOA	4.64E−79	1.38E−65	6.59E−67	2.54E−66
F3	WOA	1.30E+04	7.11E+04	4.38E+04	1.32E+04
	SCWOA	1.25E−112	4.40E+04	4.34E+03	1.25E+04
F4	WOA	6.32E−01	8.48E+01	4.29E+01	2.64E+01
	SCWOA	2.45E−60	7.51E−06	5.18E−07	1.65E−06
F5	WOA	27.33	28.83	28.02	0.49
	SCWOA	26.15	28.83	27.98	0.96
F6	WOA	5.48E−02	8.62E−01	3.48E−01	1.84E−01
	SCWOA	2.07E−02	3.01E+00	1.19E+00	9.00E−01
F7	WOA	2.13E−04	2.28E−02	4.71E−03	5.73E−03
	SCWOA	4.89E−06	1.44E−03	1.89E−04	2.81E−04
F8	WOA	−12569.13	−6498.79	−10181.39	1869.28
	SCWOA	−12569.49	−7015.00	−10597.14	1.98e+03
F9	WOA	0	5.68E−14	1.89E−15	1.04E−14
	SCWOA	0	0	0	0
F10	WOA	8.88E−16	7.99E−15	5.39E−15	2.42E−15
	SCWOA	8.88E−16	7.99E−15	1.95e−15	2.31e−15
F11	WOA	0	1.11E−16	7.40E−18	2.82E−17
	SCWOA	0	0	0	0
F12	WOA	6.15E−03	7.50E−02	2.23E−02	1.61E−02
	SCWOA	3.98E−04	9.15E−02	2.98E−02	2.07E−02
F13	WOA	1.18E−01	1.03E+00	4.59E−01	3.33E−01
	SCWOA	1.70E−02	2.80E+00	1.41E+00	1.13E+00

从测试函数在最小值、最大值、平均最优值、标准差四方面的优化效果来看，函数 F1~F7 中，除 F6 以外，SCWOA 测得的四个值相较于 WOA 效果更优，在函数 F8~F13 中，F8 的稳定性比 WOA 差一些，

在复杂多模态函数 F9、F11 中取得理论最优值，在 F10、F12、F13 中均取得一定优化效果。

为验证自适应沙丘猫鲸鱼算法的收敛性分别选取表 1 中的五个函数——F2、F4、F5、F7、F12 绘制收敛曲线来对比基本鲸鱼优化算法(收敛曲线图展示算法的收敛趋势变化)，横坐标为迭代次数，纵坐标表示函数最优值(等同于最小值)的对数，如图 2~图 7 所示。

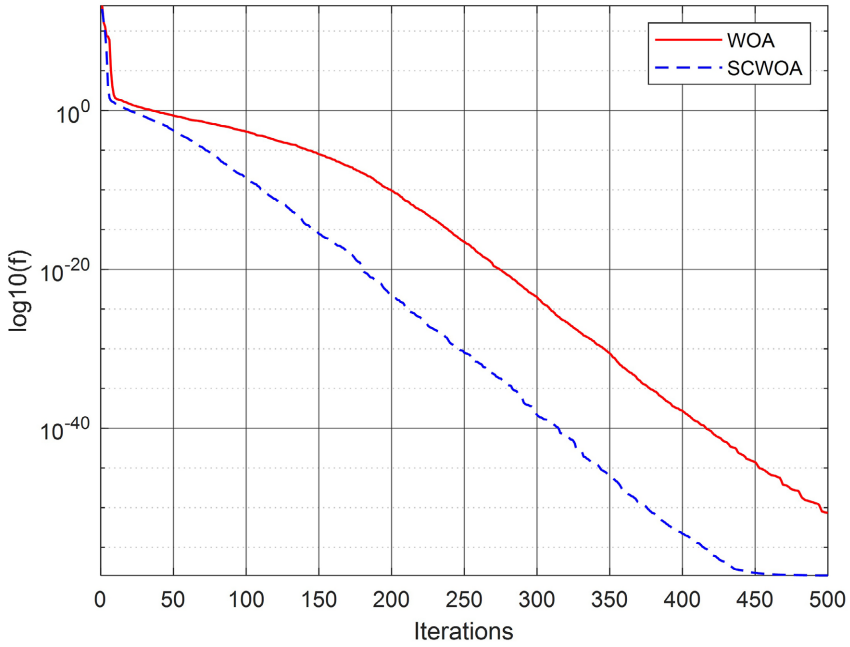

Figure 2. Convergence curve of F2
图 2. F2 的收敛曲线

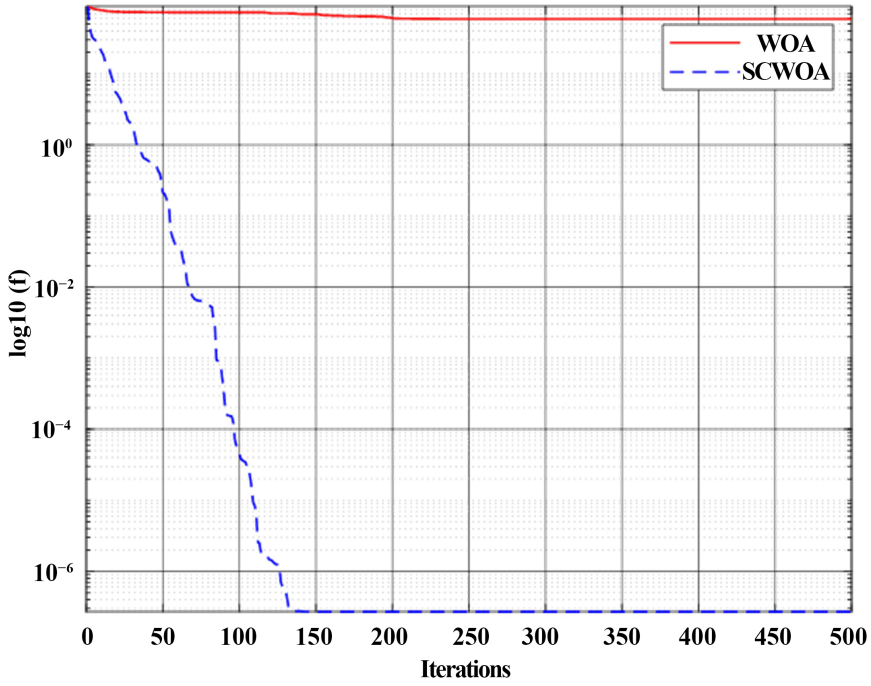

Figure 3. Convergence curve of F4
图 3. F4 的收敛曲线

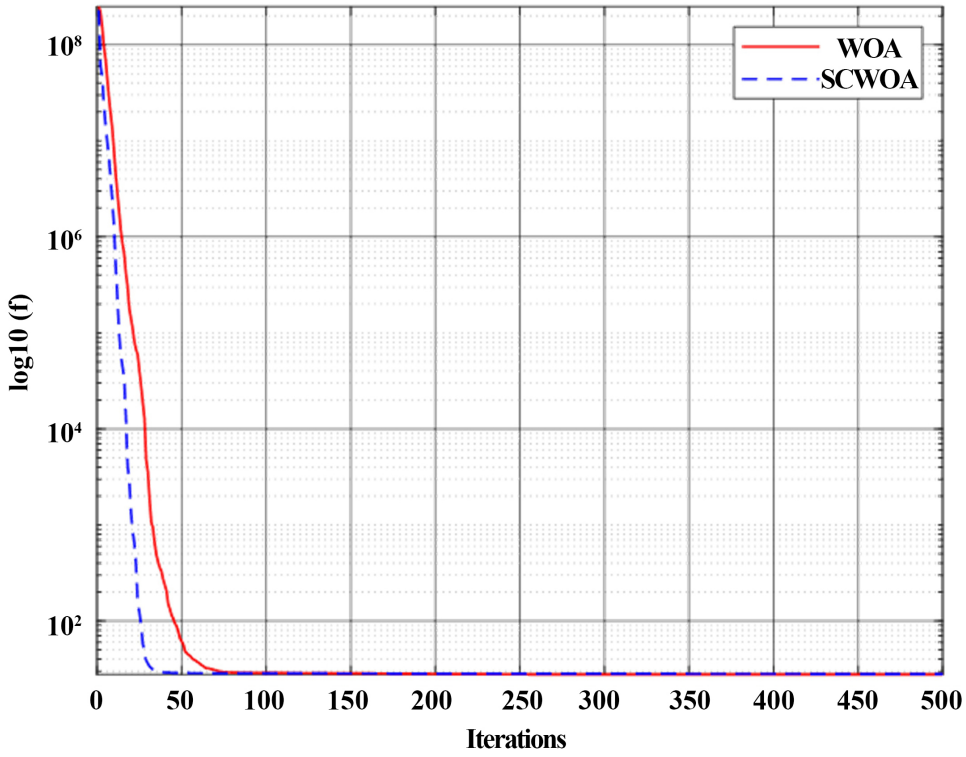

Figure 4. Convergence curve of F5
图 4. F5 的收敛曲线

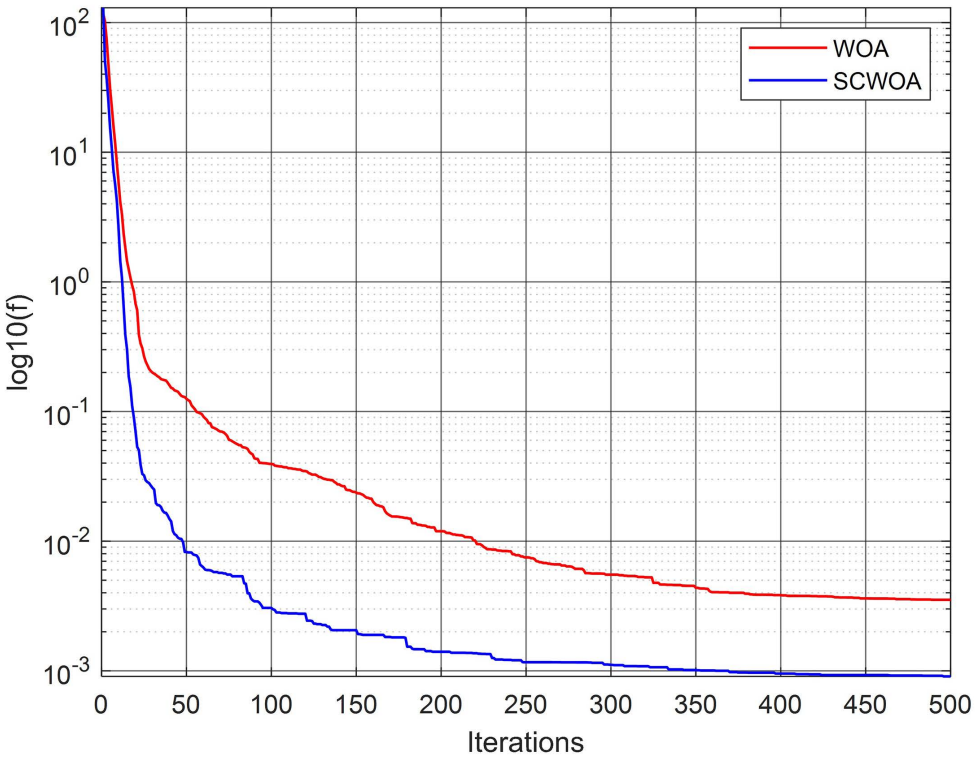

Figure 5. Convergence curve of F7
图 5. F7 的收敛曲线

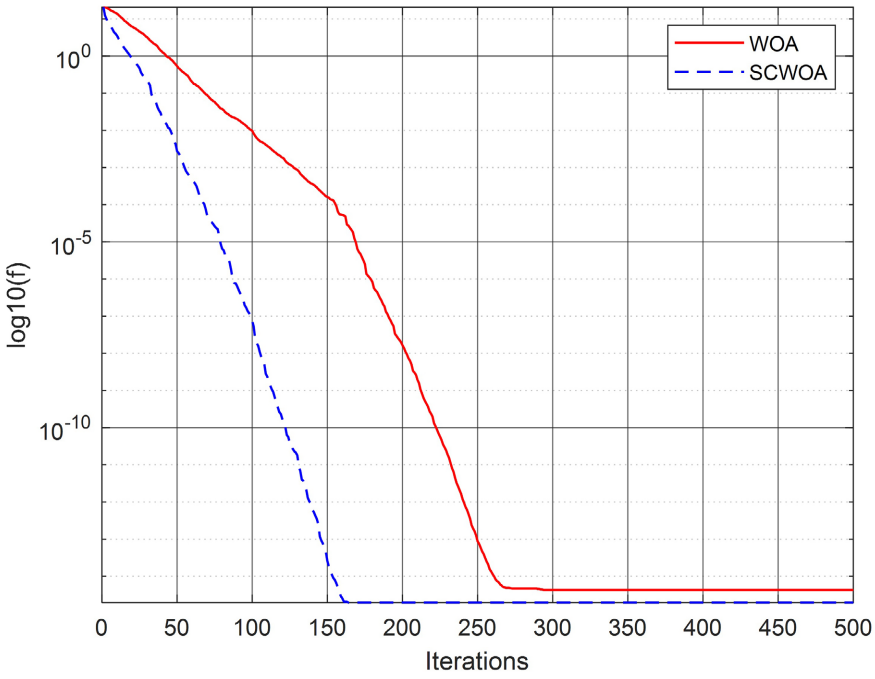

Figure 6. Convergence curve of F10
图 6. F10 的收敛曲线

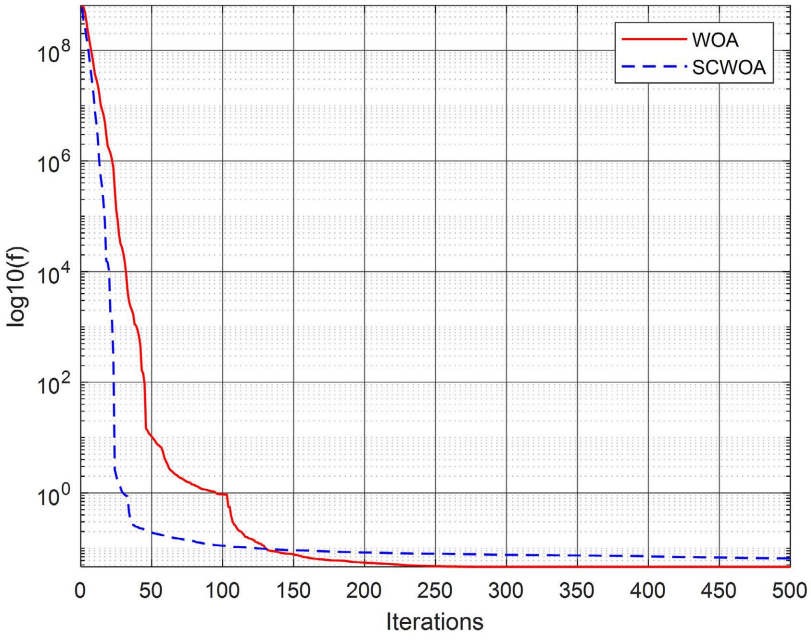

Figure 7. Convergence curve of F12
图 7. F12 的收敛曲线

分析函数收敛图，F2 迭代次数约为 10 时开始收敛，接着减缓收敛速度，直至迭代次数约 450 时已收敛，F4、F5、F7、F10 分别持续收敛至约 130、50、400、160 代时便已收敛，收敛速度快的同时取得更高的收敛精度，F12 前期收敛速度优于 WOA，约在迭代次数为 100 时收敛。总体来看，SCWOA 的收敛的速度与精度相较 WOA 更为显著。

3.6. 与其他智能算法性能对比分析

为了验证改进算法的寻优性能，以本文提出的 SCWOA 对测试函数进行数值实验，并与其他一些方法——基于余弦控制因子和多项式变异的鲸鱼优化算法(CPWOA) [10]、正余弦算法(SCA) [11]、灰狼优化算法(GWO) [12]进行对比。除 WOA、SCWOA 外，其余算法的实验数据取自原文献，实验中各智能算法的种群规模 N 为 30，最大迭代次数 Tmax 为 500，其他参数设置同原文献。各实验均独立运行 30 次，取 30 次实验的平均值(Ave)、标准差(Std)为统计结果。表 2 为具体实验结果(对比结果的最优值加粗表示)。

Table 2. Experimental results of 13 test functions by different algorithms
表 2. 不同算法对 13 个测试函数的实验结果

Function	SCA		GWO		WOA		CPWOA		SCWOA	
	Ave	Std	Ave	Std	Ave	Std	Ave	Std	Ave	Std
F1	1.67E−06	8.31E−06	1.07E−27	1.88E−27	4.32E−72	2.37E−71	2.82E−02	2.39E−02	**2.18E−106**	1.19E−105
F2	4.60E−03	1.74E−02	7.94E−17	5.55E−17	1.05E−49	5.26E−49	9.03E−02	2.19E−02	**6.59E−67**	2.54E−66
F3	**1.52E−06**	5.26E−06	2.07E−05	5.46E−04	4.38E+04	1.32E+04	7.11E+02	3.45E+02	4.34E+03	1.25E+04
F4	2.29E−05	8.01E−06	6.46E−07	5.67E−07	4.29E+01	2.64E+01	6.03E−01	2.74E+00	**5.18E−07**	1.65E−06
F5	268.33	570.13	27.01	0.59	28.02	0.48916	**3.02E−02**	1.00E−02	27.98	0.96
F6	**9.07E−10**	3.10E−10	9.06E−01	3.52E−01	3.48E−01	1.84E−01	5.21E+02	7.03E+02	1.19E+00	9.00E−01
F7	1.80E−02	1.40E−02	2.53E−03	1.11E−03	4.71E−03	5.73E−03	4.32E+00	7.12E+00	**1.89E−04**	2.81E−04
F8	−2764.909	281.36	−5.85E+03	6.86E+02	−10181.39	1869.28	4.23E−01	1.17E−01	**−10597.14**	1.98e+03
F9	16.583	8.143	3.20E+00	4.00E+00	1.04E−14	1.04E−14	5.76E+01	1.40E+01	**0**	0
F10	0.6403	0.9086	1.00E−13	5.90E−14	5.39E−15	2.42E−15	2.97E+00	1.85E+00	**2.55E−15**	2.59E−15
F11	0.2039	0.1499	7.19E−03	1.40E−02	7.40E−18	2.82E−17	9.98E−01	3.13E−16	**0**	0
F12	0.6842	0.9531	5.37E−02	3.50E−02	2.23E−02	1.61E−02	**3.44E−04**	1.67E−04	2.98E−02	0.02566
F13	**2.20E−03**	4.50E−03	6.82E−01	2.43E−01	4.59E−01	3.33E−01	3.98E−01	0.00E+00	1.41E+00	1.13E+00

从表 2 可以看出，SCWOA 相较于 CPWOA，除 F3、F5、F12 外，优化效果更为显著；相较于 GWO、SCA 与 WOA，仅有函数 F3、F6、F13 优化效果略显逊色。

4. 改进算法的应用

4.1. 机械设计优化问题

选取设计变量、列出目标函数、给定约束条件是构造优化设计问题数学模型的重要步骤。该问题的数学模型一般可以描述为如下约束非线性优化设计问题：

$$\left.\begin{aligned} &\min\ f(X) \\ &\text{s.t. } g_j(X)\geq 0 \quad j=1,2,\cdots,q \\ &\quad h_p(X)=0 \quad p=1,2,\cdots,m \\ &\quad X_i^1 \leq X_i \leq X_i^u \quad i=1,2,\cdots,n \end{aligned}\right\} \tag{17}$$

其中，X 为设计变量，$X=(X_1,X_2,\cdots,X_n)$，$X\in R^n$，$f(X)$ 为目标函数；g_j 为第 j 个不等式约束；h_p 为第 p 个等式约束；X_i^1 和 X_i^u 分别为设计变量的上下界。

4.2. 约束优化问题的处理方法

群体智能优化算法求解约束优化问题时需要将约束问题转化为无约束优化问题进行求解。将约束优化问题转化为无约束优化问题的主要处理方法是罚函数法。该方法的基本思想是通过引入惩罚项并将其添加到目标函数中以构成惩罚函数，使得约束优化问题转化为一系列无约束极值子问题，然后按照无约束优化方法来求解这些子问题。该策略将对求解过程中违反约束条件的迭代点进行“惩罚”，进而迫使无约束子问题的极小值点趋向于满足约束条件。

罚函数的一般形式为：

$$F(X)=f(X)+\lambda\left(h^2(X)+\left[\min\left(0,g(X)\right)\right]^2\right) \tag{18}$$

其中，$F(X)$为惩罚函数，$f(X)$为优化问题的原始目标函数，λ为惩罚因子；$h^2(X)$和$\left[\min\left(0,g(X)\right)\right]^2$分别为与等式有关的惩罚项和与不等式有关的惩罚项。

4.3. 设计实例及仿真实验

利用 SCWOA 优化以下 2 个机械设计问题，通过计算机仿真来验证所提出算法的性能。实例参照文献[13]。为了体现比较的公平性，SCWOA 与其他算法作对比时，实例 1 和实例 2 中 SCWOA 参数设置分别为：$N=30$，$T_{\max}=500$。另外，因实例均受约束限制，SCWOA 采用罚函数法处理约束越界。算法独立运行 30 次并取最优值，具体实验结果如表 3、表 4 所示。

4.3.1 三架桁杆设计优化

三杆桁架是一种常见的结构形式，广泛应用于桥梁、建筑物和机械设备等领域。三杆桁架的设计优化是指通过调整杆件的尺寸、形状和连接方式等参数，使得结构在满足一定约束条件下，具有最佳的性能和经济性。其目标函数和约束条件数学模型如下：

$$\left.\begin{aligned}
&\min f(X)=\left(2\sqrt{2}x_1+x_2\right)\cdot l\\
&g_1(X)=\frac{\sqrt{2}x_1+x_2}{\sqrt{2}x_1^2+2x_1x_2}P-\sigma\le 0\\
&g_2(X)=\frac{x_2}{\sqrt{2}x_1^2+2x_1x_2}P-\sigma\le 0\\
&g_3(X)=\frac{1}{\sqrt{2}x_2+x_1}P-\sigma\le 0\\
&0\le x_i\le 1,\ i=1,2
\end{aligned}\right\} \tag{19}$$

表 3 是本文 SCWOA 与其他算法获得最优解的比较结果。可以看出，SCWOA 获得的最优解优于其他算法。

Table 3. Optimization results of three-bar truss problem
表 3. 三杆桁架问题优化结果

Algorithm	x_1	x_2	x_3	x_4	x_5	$f(x)$
GOA	6.3223	5.1836	4.3773	3.5155	2.1176	1.3426
SCA	6.5976	5.7487	4.643	3.0381	2.0192	1.3757
WOA	5.0785	6.2373	4.9722	3.4064	2.7117	1.3981
SCWOA	5.9797	5.2931	4.516	3.5212	2.1652	**1.3401**

4.3.2. 悬梁臂

悬梁臂问题是一个结构工程设计问题，与方形截面悬臂梁的重量优化有关。悬臂梁一端刚性支撑，垂直力作用在悬臂的自由节点上。梁由 5 个具有恒定厚度的空心方形块组成，其高度(或宽度)为决策变量，且厚度固定(此处为 2/32/32/3)。这一问题可以用以下数学等式表示：

$$\left.\begin{array}{l}\min f(X)=0.0624\left(x_1+x_2+x_3+x_4+x_5\right)\\ g(X)=\dfrac{61}{x_1^3}+\dfrac{37}{x_2^3}+\dfrac{19}{x_3^3}+\dfrac{7}{x_4^3}+\dfrac{1}{x_5^3}-1\le 0\\ 0.01\le x_i\le 100,\ i=1,2,3,4,5\end{array}\right\} \tag{20}$$

表 4 是本文 SCWOA 与其他算法获得最优解的比较结果。可以看出，SCWOA 获得的最优解优于其他算法。

Table 4. Optimization results of cantilever beam problem
表 4. 悬梁臂问题优化结果

Algorithm	x_1	x_2	$f(x)$
GOA	0.79084	0.40215	263.8993
SCA	0.78442	0.42191	264.0574
WOA	0.74043	0.56523	265.9491
SCWOA	0.78847	0.40882	**263.8960**

5. 结束语

针对鲸鱼优化算法在优化部分高维函数问题时的缺陷，提出了一种自适应沙丘猫鲸鱼优化算法。本文所做的主要工作总结为：1) 通过参数调整策略，提高算法的搜索能力和收敛速度；2) 通过将沙丘猫算法和鲸鱼算法结合，增加多样性，较好地平衡算法的全局探索能力和局部搜索能力，提高搜索效率；3) 进行基准函数仿真实验，并与其它算法进行比较，结果表明 SCWOA 算法寻优效果更好；4) 将改进算法应用到工程优化问题中，并取得较好的优化效果。下一步的工作是研究如何运用 SCWOA 算法解决农业相关优化问题。

基金项目

基于多尺度卷积神经网络的当归病害识别技术研究(项目编号 21YF5GA088)。

参考文献

[1] Mirjalili, S. and Lewis, A. (2016) The Whale Optimization Algorithm. *Advances in Engineering Software*, **95**, 51-67. https://doi.org/10.1016/j.advengsoft.2016.01.008

[2] Chakraborty, S., Saha, A.K., Chakraborty, R., *et al*. (2022) HSWOA: An Ensemble of Hunger Games Search and Whale Optimization Algorithm for Global Optimization. *International Journal Intelligent Systems*, **37**, 52-104. https://doi.org/10.1002/int.22617

[3] Strumberger, I., Bacanin, N., Tuba, M., *et al*. (2019) Resource Scheduling in Cloud Computing Based on a Hybridized Whale Optimization Algorithm. *Applied Sciences*, **9**, Article 4893. https://doi.org/10.3390/app9224893

[4] Karaboga, D. (2005) An Idea Based on Honey Bee Swarm for Numerical Optimization, Report-TR06. Erciyes University, Kayseri.

[5] Yang, X.S. (2010) Firefly Algorithm, Stochastic Test Functions and Design Optimisation. *International Journal of Bioinspired Computation*, **2**, 78-84. https://doi.org/10.1504/IJBIC.2010.032124

[6] Lee, C.Y. and Zhuo, G.L. (2021) A Hybrid Whale Optimization Algorithm for Global Optimization. *Mathematics*, **9**, Article 1477. https://doi.org/10.3390/math9131477

[7] Garip, Z., Cimen, M.E., *et al*. (2019) The Chaos-Based Whale Optimization Algorithms Global Optimization. *Chaos Teory and Applications*, **1**, 51-63.

[8] Seyyedabbasi, A. and Kiani, F. (2023) Sand Cat Swarm Optimization: A Nature-Inspired Algorithm to Solve Global Optimization Problems. *Engineering with Computers*, **39**, 2627-2651. https://doi.org/10.1007/s00366-022-01604-x

[9] 吴迪, 吴美莲, 吴杭蕖, 等. 融合知识共享和精英反向学习的成长优化算法[J]. 闽南师范大学学报(自然科学版), 2023, 36(4): 51-61.

[10] 黄清宝, 李俊兴, 宋春宁, 等. 基于余弦控制因子和多项式变异的鲸鱼优化算法[J]. 控制与决策, 2020, 35(3): 559-568.

[11] Mirjalili, S. (2016) SCA: A Sine Cosine Algorithm for Solving Optimization Problems. *Knowledge-Based Systems*, **96**, 120-133. https://doi.org/10.1016/j.knosys.2015.12.022

[12] Mirjalili, S., Mirjalili, S.M. and Lewis, A. (2014) Grey Wolf Optimizer. *Advances in Engineering Software*, **69**, 46-61. https://doi.org/10.1016/j.advengsoft.2013.12.007

[13] Bayzidi, H., Talatahari, S., Saraee, M., *et al*. (2021) Social Network Search for Solving Engineering Optimization Problems. *Computational Intelligence and Neuroscience*, **2**, Article ID: 8548639. https://doi.org/10.1155/2021/8548639