文章编号:1671-4598(2024)08-0295-09 DOI:10.16526/j.cnki.11-4762/tp.2024.08.042 中图分类号:TP183;TM614 文献标识码:A

多策略混合的天鹰优化器

刘香怡,梁宏涛,朱 洁
(青岛科技大学 信息科学技术学院,山东 青岛 266011)

摘要:为了解决天鹰优化器集中在全局搜索导致的局部寻优能力略差、依赖初始种群质量和易陷入局部最优的问题,提出一种多策略混合的天鹰优化器;该算法利用改进的Hooke-jeeves优化基本天鹰优化器的初始化种群质量;引入模拟退火概率对易陷入局部最优解进行改进;自适应权重提高前期全局搜索效率,延缓后期局部搜索速度,避免在正解附近徘徊;选取12个基准测试函数进行实验,并将MAO应用于风力发电预测模型优化;实验结果表明,对于单峰函数、多峰函数和固定维函数,MAO比AO等对比函数具有更快的收敛速度和更高的精度;在春夏秋冬数据集上进行仿真实验,对比其他模型1月和10月预测精度提高了15%,4月和8月的预测曲线更加平滑;证实了MAO对于提高风电预测的精度和速度的可行性和实用性。

关键词:天鹰优化器;Hooke-Jeeves算法;模拟退火;自适应权重;风电预测

Aquila Optimizer with Multi-Strategy Integration

LIU Xiangyi, LIANG Hongtao, ZHU Jie
(School of Information Science and Technology, Qingdao University of Science and Techology, Qingdao 266011, China)

Abstract: In order to solve the problems of poor local optimization ability, dependence on the quality of initial population, and easily falling into local optimum caused by the global search of aquila optimizer (AO), a multi-strategy integration AO is proposed. The algorithm utilizes the improved Hooke-jeeves alogrithm to optimize the initialized population quality of the basic aquila optimizer. The simulated annealing probability is introduced to improve the local optimal solution, The adaptive weights improve the efficiency of the global search in the early stage and slow down the local search in the late stage to avoid hovering around the positive solution. Through selecting 12 benchmark test functions for experiments, and the mixed aquila optimizer (MAO) is applied to optimize the wind power prediction model. Experimental results show that for single-peak, multi-peak and fixed-dimension functions, the MAO has faster convergence speed and higher accuracy than comparative functions such as the AO. Simulation experiments are implemented on spring, summer, fall and winter datasets, compared with other models, the prediction accuracy in January and October is improved by 15%, and the prediction curves in april and august are smoother. It is verified that the MAO improves the feasibility and practicability of wind power prediction accuracy and speed.

Keywords: AO; Hooke-Jeeves algorithm; simulated annealing; adaptive weights; wind power forecast

0 引言

天鹰优化器[1](AO,aquila optimizer)是在2021年提出的基于自然界行为的群体智能优化算法,灵感来源于天鹰的捕猎过程,它们会单独在广泛的领地空间寻觅松鼠、兔子等动物,看中猎物后,天鹰会利用它们的速度和锋利的爪子抓取动物。并且提出者证明了天鹰优化器的效果优于粒子群(PSO,particle swarm optimization)、麻雀优化算法(SSA,sparrow search algorithm)等众所周知的启发式优化算法。与其他群智算法相比,天鹰优化器集中在全局搜索,避免了其他算法易陷入局部最优的缺点,有助于算法高效率和高精准率地发现搜索空间。但AO集中于全局搜索导致了局部寻优能力略差、依赖初始种群质量和容易陷入局部最优解[2]。

对于天鹰优化器集中于全局搜索导致的局部寻优能力略差,文献[3]融合了差分进化变异操作和切线飞行策略,改进了两个开发阶段的不足,提高了最优解质量。文献[4]提出了搜索控制因子、高斯突变和基于随机反对的学习方法集成在一起,避免AO陷入局部最优,但增加了额外计算时间。文献[5]将突变算子添加到AO中,对当前解突变,平衡了勘探和开发的同时避免了局部最优。文献[6]引入高斯映射提高初始种群的质量,利用交叉算子保持群体的多样性,防止陷入局部极值,加速算法的全局收敛。文献[7]引入广义正态分布优化算法扩大搜索空间,引入相量算子提高算法的高维优化能力,引入流向算子增强天鹰优化算法的开发性能。文献[8]针对天鹰初始种群会存在分布不均匀的情况,采用改进的Tent混沌映射策略提高算法初始解的质量,引入自适应t分布变异策略改

收稿日期:2024-01-22; 修回日期:2024-02-28。
基金项目:国家自然科学基金项目(61973180;62172249)。
作者简介:刘香怡(1997-),女,硕士研究生。
梁宏涛(1979-),男,博士,副教授。
引用格式:刘香怡,梁宏涛,朱 洁.多策略混合的天鹰优化器[J].计算机测量与控制,2024,32(8):295-303.

善了算法易陷入局部极值的缺陷。文献［9］通过 Circle 混沌映射及反向学习策略合理分配初始种群位置，并结合鲸鱼优化算法改善天鹰优化算法局部最优停滞。文献［10］使用改进型 Logistics 混沌映射初始化天鹰种群，采用改进的反向学习策增加种群的多样性帮助算法跳出局部最优。文献［11］引入鲸鱼优化算法的狩猎策略对天鹰优化器的局部搜索能力进行改进。

目前的研究在一定程度上改进了天鹰优化器原本的缺陷，但是在种群初始化、增强算法局部开发搜索能力、跳出局部最优解等方面还有待于更深入的研究。为此，提出了一种混合天鹰优化器算法（MAO，mixed aquila optimizer）对 AO 的随机初始种群质量不佳，局部勘探效率的固有缺点进行优化。首先通过改进的 Hooke-Jeeves[12] 算法初始化种群，避免因为种群分布不均导致的寻优效果不稳定；其次将模拟退火[13]（SA，simulated annealing）机制引入到天鹰优化器的开发阶段，加快收敛速度，避免局部最优，提高算法在开发阶段的性能。最后，应用 12 个基准函数来评估 MAO 算法的鲁棒性和有效性，且将之应用在风电预测模型中以提高预测精度和计算效率，并结合算例验证了 MAO 的有效性。

1 基本天鹰优化器

AO 主要有 4 种捕食行为，被总结归纳为 AO 算法的 4 种方法：扩大勘探、缩小勘探、扩大开发和收缩开发[14]。

1.1 扩大勘探

通过垂直高翱翔扩大搜索范围，在飞行中狩猎鸟类。

$$X_1(t+1)=X_{\text{best}}(t)\times\left(1-\frac{t}{T}\right)+\left(X_M(t)-X_{\text{best}}(t)\times rand\right) \tag{1}$$

$$X_M(t)=\frac{1}{N}\sum_{i=1}^{N}X_i(t),\forall j=1,2,\cdots,Dim \tag{2}$$

式中，N 为种群规模，Dim 为搜索空间的维度，T 为算法最大迭代次数，$X(t)$ 为 AO 算法在第 t 次迭代中的个体位置，$X(t+1)$ 为 AO 算法在第 $t+1$ 次迭代中的个体位置，$X_{\text{best}}(t)$ 表示到第 t 次迭代为止算法获得的最佳个体，$XM(t)$ 表示第 t 次迭代时的种群平均位置。

1.2 缩小勘探

通过短滑翔攻击的轮廓飞行，在靠近地面的低水平空中攻击猎物。

$$X_2(t+1)=X_{\text{best}}(t)\times Levy(D)+X_R(t)+(y-x)\times rand \tag{3}$$

$$y=r\times\cos\theta \tag{4}$$

$$x=r\times\sin\theta \tag{5}$$

$$r=r_1+U\times D_1 \tag{6}$$

$$\theta=-\omega\times D_1+\theta_1 \tag{7}$$

$$\theta_1=\frac{3\pi}{2} \tag{8}$$

式中，$Levy(D)$ 为莱维飞行分布策略，x、y 为螺旋飞行的形状，r 为搜索步长，取值范围是［1～20］；$U=0.00565$，$\omega=0.005$。

1.3 扩大开发

通过低空飞行和慢下降逐渐攻击行动缓慢的猎物。

$$X_3(t+1)=\left[X_{\text{best}}(t)-X_M(t)\right]\times\alpha-rand+\left[(U_B-L_B)\times rand+L_B\right]\times\delta \tag{9}$$

式中，U_B 和 L_B 分别为给定问题的上界和下界，α 和 δ 是开发调整参数，被固定为 0.1。

1.4 收缩开发

通过俯冲在陆地上行走和抓取大猎物的幼崽。（随机性攻击）

$$X_4(t+1)=QF\times X_{\text{best}}(t)-G_1\times X(t)\times rand-G_2\times Levy(D)+rand\times G_1 \tag{10}$$

$$QF(t)=t^{(2\times rand-1)/(1-T)^2} \tag{11}$$

$$G_1=2\times rand-1 \tag{12}$$

$$G_2=2\times\left(1-\frac{t}{T}\right) \tag{13}$$

式中，$QF(t)$ 为用于平衡搜索策略的质量函数值；G_1 为追踪猎物过程中 AO 的各种运动；G_2 为现行递减的飞行斜率值，取值范围为［0，2］。

在 AO 算法的每个行为中，虽然都添加了贪心算法保留较优的个体以保证算法的寻优能力，但从实验来看。最终离正解还有一定的距离，局部搜索能力欠佳[15]。

2 改进天鹰优化器

从基本 AO 算法的实现公式可以看出：

1）在初始化阶段，种群的随机生成影响了算法的稳定性；

2）局部开发阶段，随着种群的收敛，公式（9）会收敛到常数，公式（10）会向 0 收敛，使得算法的局部搜索能力欠佳。

对于以上影响算法的不利因素，本文对 AO 算法做如下改进。

2.1 利用改进的 Hooke-jeeves 算法初始化种群

在搜索空间随机生成初始种群的方式，会造成初始个体分布不均，进而生成很多差解，需要大量的计算才能得到最优解，降低了运行效率，影响了算法寻优计算能力的稳定性。文献［16］提出了一种基于搜索空间均匀划分的种群初始化方法，改进后的方法能够确保在整个搜索空间都均匀分布初始种群个体，且具有更优的运行时间和收敛特性。

首先将搜索空间均匀划分为 q 个子空间，子向量通过以下得到：

$$S_k=\begin{cases}x_{\min}, & k=1\\ S_{k-1}+l, & 2\leqslant k\leqslant q\\ x_{\max}, & k=q+1\end{cases} \tag{14}$$

其中：$[x_{\min},x_{\max}]$ 是 n 个决策变量搜索域的取值范围，l 为两个子向量之间的距离：

$$l=\frac{x_{\max}-x_{\min}}{q} \tag{15}$$

划分完成后，使用轮盘赌选择法[17]选择初始点。计算每一个子空间所有个体的适应度总和，然后求取各个个体适应度与适应度总和的比值，比值越大，选择的概率越大。若某个天鹰个体的适应度值为 f_i，种群规模为 N，那么这个个体被选中的概率为：

$$P_i = \frac{f_i}{\sum_1^N f_i}, i = 1,2,\cdots,N \tag{16}$$

然后用 Hooke-Jeeves 算法对搜索空间探索，选择局部最优作为初始个体. 具体的过程如下：

1）设置初始值。搜索空间的初始点 x^1、初始步长 s、加速因子 $a \geqslant 1$、衰减率 $b < 1$、精度 $z > 0$、$ej = (0,\cdots,0,1,0,\cdots,0)^T$。令 $y^1 = x^1, k = 1, j = 1$。

2）轴向搜索。从参考点 y^j 出发，在 x 轴方向上按步长间隔顺序地搜索函数的下降点：

$$y^{j+1} = \begin{cases} y^j + se_j, f(y^j + se_j) < f(y^j) \\ y^j - se_j, f(y^j - se_j) < f(y^j) \\ y^j, min[f(y^j + se_j), f(y^j - se_j)] > f(y^j) \end{cases} \tag{17}$$

遍历完成后，如果 $f(y^{n+1}) < f(x^k)$，则进行模式搜索，否则调整步长进行轴向搜索。

3）模式搜索。令 $x^{k+1} = y^{n+1}$，将 $y^1 = x^{k+1} + \alpha(x^{k+1} - x^k)$ 作为新的参考点，令 $k: = k+1, j = 1$。转到 2）。

4）调整步长。如果 $s \leqslant z$，停止搜索，初始个体为 x^k；否则，令 $s: = bs$，$y^1 = x^k$，$x^{k+1} = x^k$，$k: = k+1$，$j = 1$。转到 2）。

最后，将得到的个体作为初始种群的个体，重复上述步骤直至满足初始群体所要求的数目为止，即初始种群构建完成。

假设问题的边界为 [lb，ub]，得到的 [0，1] 区间的点可以通过映射 $lb+$（$ub-lb$）$*$随机数，映射到搜索空间 [lb，ub] 中。

通过以上对种群初始化的优化可以看出，均匀划分搜索空间保证了初始解的均匀分布，避免了算法因为种群随机而造成的运算结果不稳定。Hooke-Jeeves 算法求解简单，收敛速度较快，且具有较好的局部精细搜索能力；同时种群的均匀分布也克服了 Hooke-jeeves 对初始点敏感，而易陷入局部最优的缺点。

2.2　引入模拟退火概率

天鹰优化器在局部寻优阶段容易陷入局部最优，结果会在正解附近徘徊，但距离正解还是有一定的距离。

在迭代开始前，加入一个自适应加权 α[18]调整的线性变化，加速前期搜索并且减慢后期寻优的速度。

$$\alpha = \frac{4}{\pi}\arctan(\frac{t}{T}) \tag{18}$$

模拟退火的提出是基于概率的固体退火的原理，当温度 H 很高时，粒子快速无序运动，这时选择较差解的概率很大；当 T 慢慢降低，粒子趋于有序，选择较差解的概率减小，得到近似最优解。

$$P = e^{-\Delta C/kH} \tag{19}$$

$$H = \varepsilon \times H, 0 < \varepsilon < 1 \tag{20}$$

其中：H 表示温度的模拟参数退火因子，随着迭代次数增加，H 逐渐趋于 0；C 表示新解的代价。

基于模拟退火的以一定概率接受较差解，在天鹰的开发阶段，加入判断机制。以 X_3 为例，如果 X_3 的适应度值小于当前适应度值，那么更新当前适应度。反之以一定的概率 P 接受该差解，计算公式如下：

$$P = \exp\left\{\frac{-[Ffun(X_3) - Ffun()]}{H}\right\} \tag{21}$$

通过上式有概率的接受差解，可以帮助天鹰在局部搜索阶段寻找新的食物，避免了过度寻求最优解而陷入局部最优的缺陷。

2.3　MAO 算法流程

根据以上的算法改进思想，下面给出 MAO 的算法流程，如算法 1 所示。

算法 1：MAO

输入 q、s、a、b、z、N、Dim、T、ε，etc

输出 best position、best fitness

1. Divide the search space

2. Calculate the fitness function values and select the initial individual

3. Explore the search space

4. Building the initial population

5. Define initial parameter(i、e、α、δ, etc)

6. Calculate α

7. While $t < \alpha \times T$ do

8.　for $i = 1: N$ do

9.　Update the $X_M(t)$、x、y、G_1、G_2 , etc

10.　Expanded exploration(X_1):

Update the current solution using Equation(1)

11.　Narrowed exploration(X_2):

Update the current solution using Equation(3)

12.　Expanded exploitation(X_3):

Update the current solution using Equation(9)

13.　Judgment selection mechanism

14.　Narrowed exploitation(X_4):

Update the current solution using Equation(10)

15.　Judgment selection mechanism

16.　end for

17. end while

2.4　MAO 时间复杂度分析

MAO 算法与 AO 算法的框架主要有 3 个不同之处：1）第 3 行采用了新的初始种群生成算法；2）第 7 行为 t 加入了一个自适应加权；3）第 13 行、第 14 行加入了判断选择机制。

时间复杂度是描述一个算法的运行时间，用来衡量算法的运行效率。在本文中由种群初始化、计算适应度值、更新解等共同决定。下面对 MAO 算法的复杂度进行分析。

在标准 AO 算法中，假设种群规模为 N，搜索空间维

度为 Dim，最大迭代次数为 T，解的更新过程的计算复杂度为 $O(T\times D)+O(T\times N\times Dim)$。因此，标准 AO 算法的时间复杂度为 $O[N\times(T\times D+1)]$。

依据算法的描述和时间复杂度的计算规则，MAO 在 AO 的基础上：

1）首先将搜索空间均匀分为 q 个子空间，增加了轮盘赌选择法比较每个个体适应度值 $O(q+N)$；

2）使用 Hooke-Jeeves 探索每个子空间，则种群初始化阶段的时间复杂度为 $O(N)$；

3）迭代期间，加入了自适应加权使得自动变化，每一次迭代都重新计算，那么时间复杂度为 $O(T)$；

4）局部搜索阶段，加入了判断机制，与原算法的时间复杂度一致。

所以，MAO 最终的时间复杂度为：

$$T(MAO)=T(AO)+T(\text{式}(16))+T(Hooke-Jeeves)+T(\text{式}(20))=O(N\times(T\times D+1))+O(q+N)+O(N)+O(T)=O(N)$$

从上述分析可以看出，MAO 和 AO 在时间复杂性上处于同一水平，并不会带来额外的计算量。

3 实验仿真与结果分析

3.1 仿真实验环境

实验设备：Windows10、64 位操作系统，处理器为 11th Gen Intel（R）Core（TM）i5-11400 @ 2.60 GHz 2.59 GHz，内存 16 GB，仿真软件为 Matlab R2021a。

3.2 比较对象和参数设置

本文选取基本天鹰优化器（AO）、粒子群算法（PSO）、模拟退火算法（SA）、麻雀优化算法（SSA），与本文提出的混合天鹰优化器（MAO）进行对比。所有算法的种群规模设置为 30、迭代次数为 500，其他参数见表 1。

表 1 各算法的实验参数

算法	参数
AO	$\alpha=0.1,\delta=0.1,U=0.005\ 6,\omega=0.005$
PSO	$c_1=2,c_2=2,v_{\max}=6$
SA	$\alpha=0.98,q\in(0.8,0.99)$
SSA	$PD=0.2,SD=0.1,ST=0.8$
MAO	$\alpha=0.1,\delta=0.1,U=0.005\ 6,\omega=0.005$

3.3 测试函数

本文采用了 12 个测试函数用于评估 MAO 算法的性能。所有算法的迭代次数、搜索空间维度和种群规模都相同，测试函数的详细信息见表 2。

使用了 3 种类型的基准函数：单峰（f_1-f_4）、多峰（f_5-f_8）和固定维数多峰函数（f_9-f_{12}）。分别用于评价 MAO 的勘探和开发能力的单峰和多峰函数在 50 维度上的结果如表 3 所示。此外，这两种测试函数在 100 维中使用，结果如表 4 所示。最后，表 6 中给出了具有固定维度多峰函数的结果，表明了 MAO 在较低维度中的探索能力。结果证实了所提出的 MAO 在处理这些具有挑战性的测试函数方面的优越性。

表 2 本文使用的实验测试函数

测试函数	函数	维度	搜索范围	最优值
f_1	$f(x)=\sum_{i=1}^{n}x_i^2$	50,100	$[-100,100]$	0
f_2	$f(x)=\sum_{i=0}^{n}\lvert x_i\rvert+\prod_{i=0}^{n}\lvert x_i\rvert$	50,100	$[-10,10]$	0
f_3	$f(x)=\sum_{i=1}^{d}(\sum_{j=1}^{i}x_j)^2$	50,100	$[-100,100]$	0
f_4	$f(x)=\sum_{i=1}^{n-1}[100(x_i^2-x_{i+1})^2+(1-x_i)^2]$	50,100	$[-30,30]$	0
f_5	$f(x)=\sum_{i=1}^{n}(-x_i\sin(\sqrt{\lvert x_i\rvert}))$	50,100	$[-500,500]$	−12 569.5
f_6	$f(x)=\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)+10]$	50,100	$[-5.12,5.12]$	0
f_7	$f(x)=-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2})-\exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i))+20+e$	50,100	$[-32,32]$	0
f_8	$f(x)=1+\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})$	50,100	$[-600,600]$	0
f_9	$f(x)=-\sum_{i=1}^{4}c_i\exp(-\sum_{i=1}^{6}a_{ij}(x_j-p_{ij})^2)$	6	$[0,1]$	−3.32
f_{10}	$f(x)=-\sum_{i=1}^{5}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	$[0,1]$	−10.153 2
f_{11}	$f(x)=-\sum_{i=1}^{7}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	$[0,1]$	−10.402 8
f_{12}	$f(x)=-\sum_{i=1}^{10}[(X-a_i)(X-a_i)^T+c_i]^{-1}$	4	$[0,1]$	−10.536 3

3.4 寻优精度分析

由于单峰函数集只能得到一个最优解，所以可用其来检验 MAO 的收敛速度和求解精度。多峰函数有多个局部极值点，并且在多个有效解中存在全局最优解，用于测试 MAO 的全局搜索能力、局部快速求解能力和跳出局部最优解的能力。加粗的数值表示该算法在测试函数上取得了最优的计算结果。表 3、表 4 展示了 5 个算法在单峰函数、多峰函数上 50 维和 100 维的比较结果；表 5 展示了在 4 个固定维数多峰函数上的比较结果。

表 3 展示出了 MAO 和对比算法在 f_1-f_8 上 50 维度时的平均值和标准差，进行分析可知：

1）MAO 一共取得了 6 个最优结果，并且优势更加突出。PSO 和 SSA 获得了一个最优结果，AO 和 SA 没有获得过最优结果。

2）其中在 f_2 上 MAO 精度略差于 PSO；在 f_8 上的效果差于 SSA，与 AO 和 PSO 的结果相当。

为了更进一步检验 MAO 在更高的维度上也具有良好的优化能力，将实验维度设为 100 继续进行验证，其实验结果如表 4 所示。

从表 4 中可以看出 MAO 的收敛效果在 100 维上，不论是标准差还是平均值都明显好于其对比算法，说明通过 MAO 得到的解的质量和鲁棒性具有明显的优异性。且 MAO

表 3　经典测试函数(f_1-f_8)的比较结果

函数编号		测量值($D=50$)							
		f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
MAO	AVG	$\mathbf{3.35\times10^{-309}}$	2.37×10^{-307}	$\mathbf{4.19\times10^{-300}}$	$\mathbf{5.15\times10^{-9}}$	$\mathbf{-3.17\times10^{3}}$	$\mathbf{3.01\times10^{-14}}$	$\mathbf{4.91\times10^{-16}}$	3.27×10^{-13}
	SD	$\mathbf{2.82\times10^{-308}}$	2.25×10^{-306}	$\mathbf{5.02\times10^{-298}}$	$\mathbf{2.19\times10^{-8}}$	$\mathbf{2.38\times10^{3}}$	$\mathbf{4.48\times10^{-13}}$	$\mathbf{2.27\times10^{-15}}$	2.18×10^{-12}
AO	AVG	3.37×10^{-298}	3.84×10^{-297}	3.25×10^{-294}	2.81×10^{-5}	-1.7×10^{3}	6.93×10^{-5}	8.16×10^{7}	4.91×10^{-13}
	SD	3.05×10^{-297}	0.93×10^{-296}	4.34×10^{-293}	1.08×10^{-4}	1.47×10^{3}	3.82×10^{-4}	2.74×10^{6}	2.37×10^{-12}
PSO	AVG	3.48×10^{-285}	$\mathbf{2.91\times10^{-307}}$	3.77×10^{-300}	5.11×10^{-3}	-0.11×10^{0}	2.33×10^{-1}	1.03×10^{-8}	2.66×10^{-13}
	SD	3.57×10^{-284}	$\mathbf{2.72\times10^{-306}}$	4.35×10^{-299}	2.46×10^{-2}	2.42×10^{3}	5.19×10^{-2}	4.52×10^{-7}	2.26×10^{-12}
SSA	AVG	3.31×10^{-308}	9.47×10^{-299}	7.45×10^{-300}	4.95×10^{-7}	0.33×10^{2}	4.17×10^{-1}	5.87×10^{-3}	$\mathbf{5.84\times10^{-14}}$
	SD	1.94×10^{-307}	7.66×10^{-297}	6.68×10^{-298}	1.51×10^{-6}	2.93×10^{3}	1.96×10^{-2}	0.26×10^{-2}	$\mathbf{3.63\times10^{-13}}$
SA	AVG	4.39×10^{-297}	6.79×10^{-297}	5.13×10^{-263}	3.49×10^{-4}	2.26×10^{3}	5.19×10^{-3}	3.01×10^{-6}	2.34×10^{-13}
	SD	2.86×10^{-296}	3.41×10^{-286}	2.93×10^{-262}	1.19×10^{-3}	2.85×10^{3}	2.74×10^{-2}	5.52×10^{-5}	5.1×10^{-12}

表 4　经典测试函数(f_1-f_8)的比较结果

函数编号		测量值($D=100$)							
		f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
MAO	AVG	$\mathbf{5.81\times10^{-300}}$	$\mathbf{3.16\times10^{-286}}$	$\mathbf{3.94\times10^{-291}}$	$\mathbf{3.86\times10^{-8}}$	$\mathbf{-2.84\times10^{3}}$	$\mathbf{5.71\times10^{-13}}$	$\mathbf{5.18\times10^{-13}}$	$\mathbf{7.36\times10^{-13}}$
	SD	$\mathbf{2.97\times10^{-300}}$	$\mathbf{1.02\times10^{-286}}$	$\mathbf{1.92\times10^{-290}}$	$\mathbf{2.01\times10^{-7}}$	$\mathbf{1.35\times10^{2}}$	$\mathbf{2.45\times10^{-12}}$	$\mathbf{2.64\times10^{-12}}$	$\mathbf{3.19\times10^{-11}}$
AO	AVG	5.12×10^{-251}	3.25×10^{-142}	6.31×10^{-201}	1.79×10^{-3}	-0.18×10^{2}	9.37×10^{-3}	8.18×10^{-5}	3.37×10^{-9}
	SD	2.34×10^{-251}	2.17×10^{-142}	2.90×10^{-201}	3.24×10^{-2}	0.35×10^{2}	9.97×10^{-2}	4.92×10^{-4}	4.12×10^{-8}
PSO	AVG	7.17×10^{-124}	5.82×10^{-142}	9.55×10^{-123}	1.02×10^{-2}	-1.84×10^{2}	8.33×10^{-2}	1.31×10^{-6}	5.73×10^{-10}
	SD	3.63×10^{-123}	3.39×10^{-141}	8.37×10^{-122}	0.39×10^{-2}	1.59×10^{1}	5.79×10^{-1}	3.44×10^{-6}	1.55×10^{-9}
SSA	AVG	2.83×10^{-114}	7.84×10^{-140}	3.91×10^{-284}	9.41×10^{-1}	-2.13×10^{2}	1.37×10^{-1}	7.42×10^{-2}	8.12×10^{-10}
	SD	5.91×10^{-114}	3.04×10^{-139}	2.88×10^{-283}	6.24×10^{-1}	1.33×10^{1}	3.9×10^{0}	3.8×10^{-1}	6.89×10^{-9}
SA	AVG	4.13×10^{-100}	2.19×10^{-83}	2.19×10^{-193}	2.16×10^{-3}	-2.18×10^{3}	3.17×10^{-1}	5.14×10^{-2}	5.13×10^{-8}
	SD	0.37×10^{-009}	4.56×10^{-83}	1.94×10^{-193}	1.53×10^{-2}	2.51×10^{1}	1.2×10^{0}	7.38×10^{-2}	1.97×10^{-7}

表 5　经典测试函数(f_9-f_{12})的比较结果

函数编号		测量值			
		f_9	f_{10}	f_{11}	f_{12}
MAO	AVG	$\mathbf{-3.23\times10^{0}}$	$\mathbf{-10.17\times10^{0}}$	$\mathbf{-10.26\times10^{0}}$	$\mathbf{-10.24\times10^{0}}$
	SD	$\mathbf{1.67\times10^{-1}}$	$\mathbf{2.23\times10^{-1}}$	$\mathbf{3.14\times10^{-1}}$	$\mathbf{2.91\times10^{-1}}$
AO	AVG	-2.34×10^{0}	-1.59×10^{0}	-4.94×10^{0}	-4.96×10^{0}
	SD	3.27×10^{-1}	1.73×10^{-1}	1.17×10^{-1}	2.8×10^{-1}
PSO	AVG	-3.16×10^{0}	-10.01×10^{0}	-10.09×10^{0}	-2.24×10^{0}
	SD	3.21×10^{-1}	2.13×10^{0}	3.46×10^{0}	2.22×10^{0}
SSA	AVG	-3.2×10^{0}	-10.0×10^{0}	-10.02×10^{0}	-3.59×10^{0}
	SD	2.01×10^{-1}	1.25×10^{0}	0.24×10^{0}	3.24×10^{0}
SA	AVG	-2.12×10^{0}	-4.79×10^{0}	-5.24×10^{0}	-2.39×10^{0}
	SD	2.36×10^{-1}	0.26×10^{0}	-2.29×10^{-1}	1.01×10^{-1}

在 100 维和 50 维上相比较，平均值和标准差的变化都比较小。这说明，MAO 不会陷入“维灾难”问题[19]。

表 5 展示了 5 个算法在固定维度多峰函数上的优化结果，从表中可知 MAO 的平均值都更接近最优值，且标准差对比其他 4 个算法都是最小的。因此，无论是在单峰函数、多峰函数还是固定维数多峰函数上，MAO 算法与对比算法相比有较大的提高，得到了很好的优化结果。表明 MAO 在高维函数上也拥有更好的优化能力，比其他算法更具有竞争力。

3.5 收敛曲线分析

算法的优劣可以通过其收敛曲线直观地表现出来，收敛曲线展示了算法的收敛速度和收敛精度。MAO、AO、PSO、SSA、SA 在 30 维情况下，对上述基准函数的部分收敛曲线对比如图 1 所示。观察图中各算法的收敛曲线可以看出 MAO 在 6 个函数下的收敛速度在整个迭代过程中都是比较快的，并且收敛精度都优于其他对比算法；在 f_5 上初期搜索速度慢于 AO，但计算能力优于 AO；在 f_{11} 上收敛速度慢于对比算法，但精度最优。与其他算法相比，MAO 具有更强的全局探索能力，并能跳出局部极值，在寻优能力上具备明显的优势，也进一步地佐证了本文提出的改进策略是有效的。

4 算法应用分析

为了进一步验证 MAO 的实用性和可行性，本文使用 MAO 对 SAE 的初始参数进行优化获得模型最优参数，从而提高模型的运算效率和预测准确率。

风电数据是一种时间序列数据，具有很强的随机性、波动性和不确定性[20]。为了实现风电功率的准确预测，选取可以给出比原始数据更好的特征描述的堆叠自编码器（SAE，stacked auto-encoder），SAE 是由多个自编码器（AE，auto-encoder）连接在一起构建的一个深层的神经网络结构，通过逐层堆叠，每一层都可以学习到数据的不同抽象层次的特征表示[21]，减少模型对训练数据的依赖。和在时间序列预测方面较为突出的长短期记忆网络（LSTM，long short-term memory）模型，对季节性风电出力的典型情景进行预测。

4.1 堆叠自编码器

如图 2 所示，SAE 是一种由多个 AE 构成的深度学习网络，将上层隐藏层的输出转化为下层隐藏层的输入。该网络可以从复杂高维的数据中学习到不同维度和层次的抽

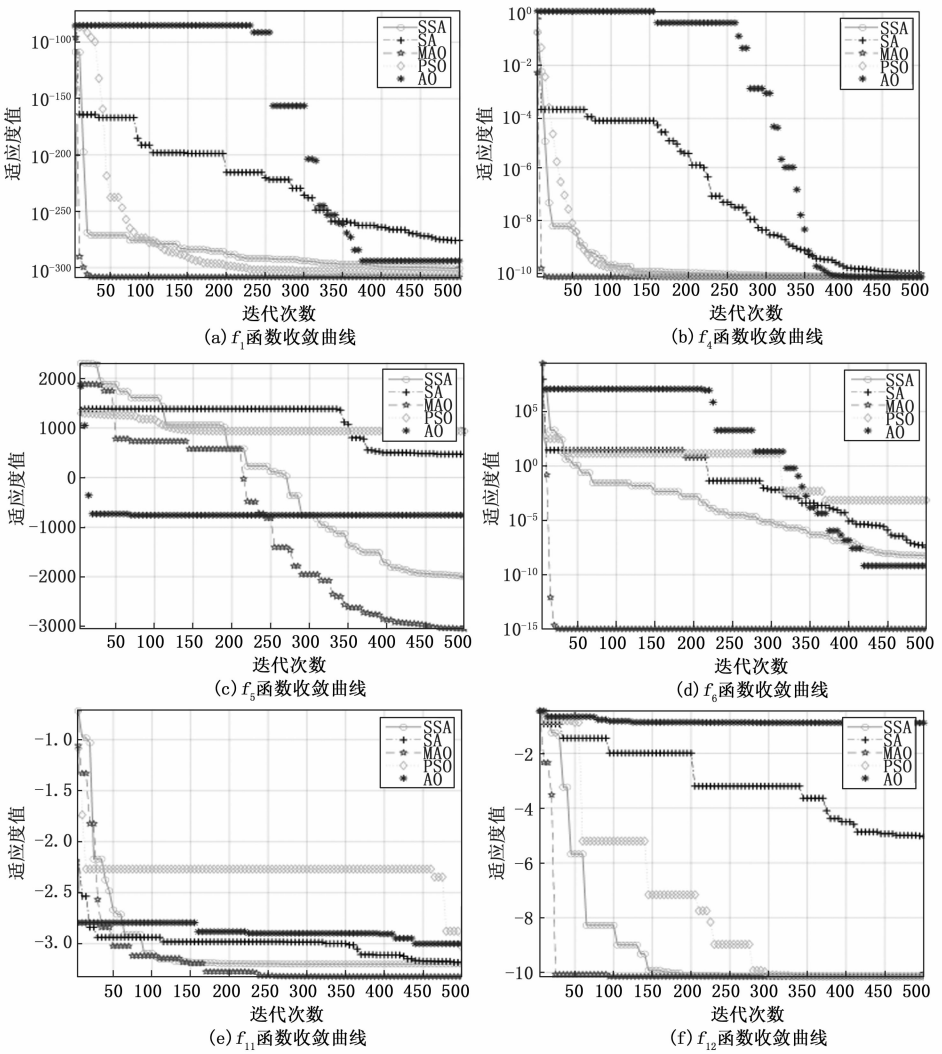

图 1 部分函数收敛曲线图

象特征向量，从 SAE 提取的特征然后被传送到神经网络以预测风速。

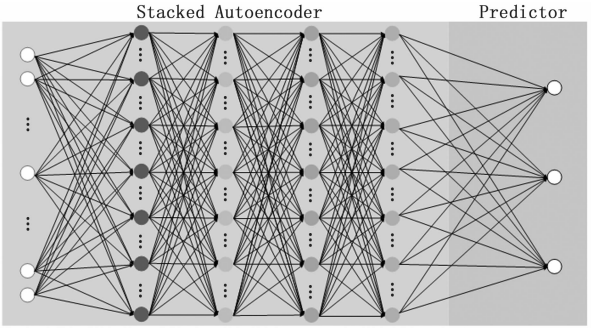

图 2　SAE 结构图

1）首先输入原始数据，采用无监督方式训练第一层编码器，获得参数权重 W_1 和偏置量 b_1。

2）其次根据：

$$z_2 = W_1 * data + repmat(b_1,1,m) \tag{22}$$

$$activation = sigmoid(z_2) \tag{23}$$

将得到的激活函数 activation 的输出作为下一个的输入，以此类推。重复以上步骤直到初始化完成所有 AE。

3）最后把末层 AE 的隐含层的输出作为预测层的输入，使用有监督的方法来对预测层的参数进行训练。

最终，将以上 $n+1$ 层结合起来构建成一个包含 n 个隐藏层和 1 个最终预测层的堆叠自编码网络，这个网络可以对风电数据集进行预测。

4.2　长短期记忆人工神经网络

LSTM 是一种时间循环神经网络，其结构如图 3 所示。从理论上说，在序列加工的过程中，细胞的状态可以不断地传递有关的信息。因此，即使是较早时刻的步长信息也能携带到较后时刻步长的细胞中来。

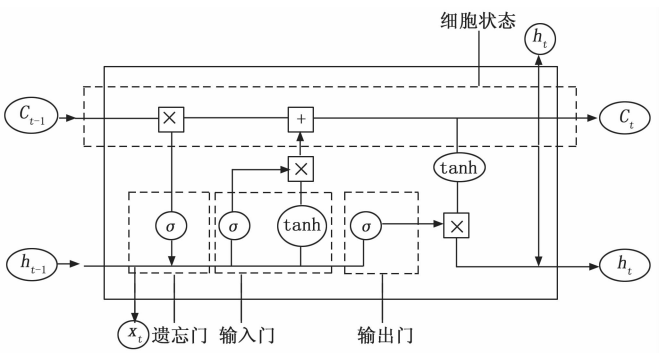

图 3　LSTM 结构图

其中：遗忘门通过式（24）用来遗忘无用信息和保留有用信息。

$$f_t = \sigma(W_f \cdot [h_{t-1},x_t] + b_f) \tag{24}$$

输入门通过式（25）和（26）用于更新细胞状态，由上一层输出的隐藏层和当前输入作为输入。

$$i_t = \sigma(W_i \cdot [h_{t-1},x_t] + b_i) \tag{25}$$

$$\widetilde{C}t = \tanh(W_C \cdot [h_{t-1},x_t] + b_C) \tag{26}$$

输出门通过式（27）和（28）用来确定下一个隐藏层的值，并且将新的细胞状态和新的隐藏状态传递到下一层去。

$$o_t = \sigma(W_o \cdot [h_{t-1},x_t] + b_o) \tag{27}$$

$$h_t = o_t \cdot \tanh(C_t) \tag{28}$$

预测模型如图 4、图 5 所示。随机选取春夏秋冬四季中各一天进行实验对比：春节的风速适中，但风向相对较为多变；秋季的风速相对稳定，风向和风速变化相对较小，

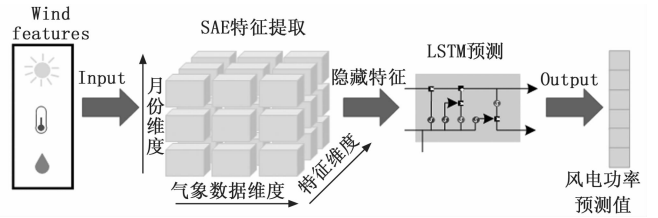

图 4　MAO-LSTM-SAE 结构图

这种风速范围有利于风力发电系统的稳定运行和高效发电；夏季的风速一般较低，风向多变，风能资源相对较少，但通常降水较多，是雷暴和台风等天气系统活跃的季节，不稳定的天气条件会对风力发电机组的功率输出产生一定影响；冬季的风力资源相对丰富，风向和风速变化较大，可以提供强劲的风能资源，但需要灵活地应对。

4.3　实验步骤和方法

为了准确、全面地评价 MAO-LSTM-SAE 模型的性能，采用均方根误差 *RMSE*、平均绝对误差 *MAE* 作为评价指标。*RMSE* 表示测量值域真值曲线之间的拟合程度，*MAE* 直接地反映了预测值与实际值之间的实际差距。评价指标的表达式如下：

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{Y}i - Yi)^2} \tag{29}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n} | \hat{y}i - yi | \tag{30}$$

表 6 给出了 4 个神经网络在测试集中的预测精度评估结果，结果表明，MAO-LSTM-SAE 的预测精度较其他有所提高，实际值与预测值的误差减小。

表 6　预测精度评价指标

算法	*RMSE*/kW	*MAE*/%
MAO-SAE-LSTM	1.280 1	0.901 4
SAE-LSTM	1.593 1	1.431 7
LSTM	1.812 6	1.452 9
SVM	1.862 7	1.565 8

4.4　实验结果与分析

首先设置 MAO 的初始参数，将 SAE-LSTM 的参数设置为 MAO 的目标优化参数，将 MAO 返回的最佳适应度对应的参数输入到 SAE-LSTM 中，最终形成了 MAO-SAE-LSTM 风电发电功率测模型。实验将使用某西班牙风电数据集进行风电发电功率预测，选取 80%的数据作为训练集，其余作为测试集。随机分别选取 4 个季节中的某一天 24 h 的时间节点对比 MAO-SAE-LSTM、SAE-LSTM、LSTM、SVM 这 4 个模型的效果。预测精度结果如图 6 所示。

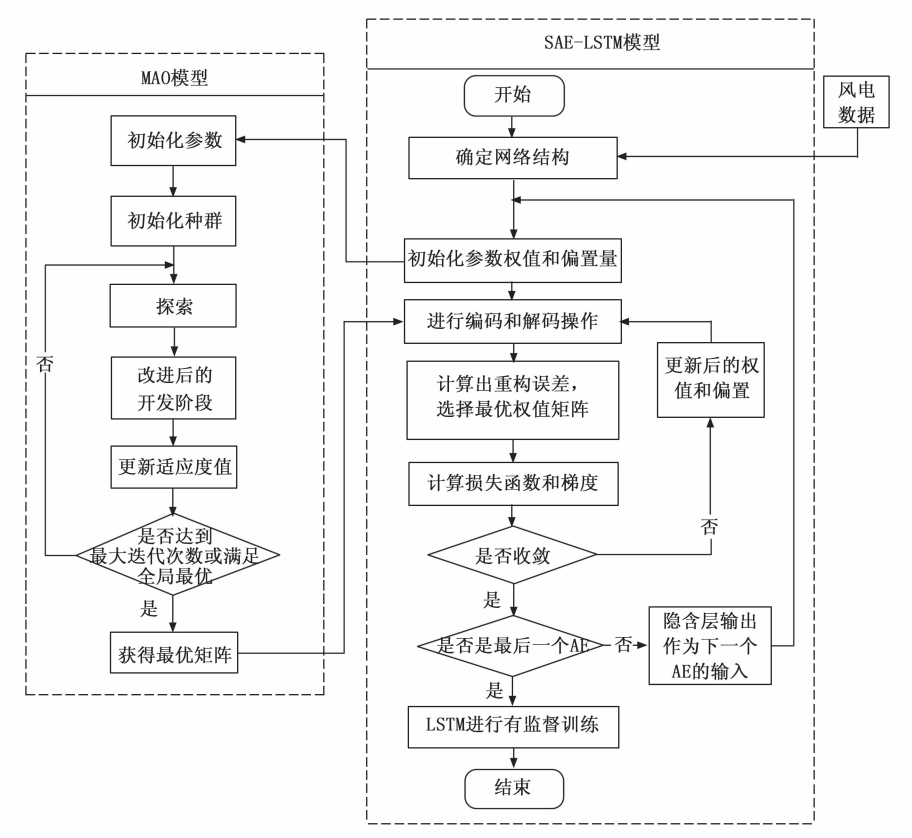

图 5 MAO-LSTM-SAE 流程图

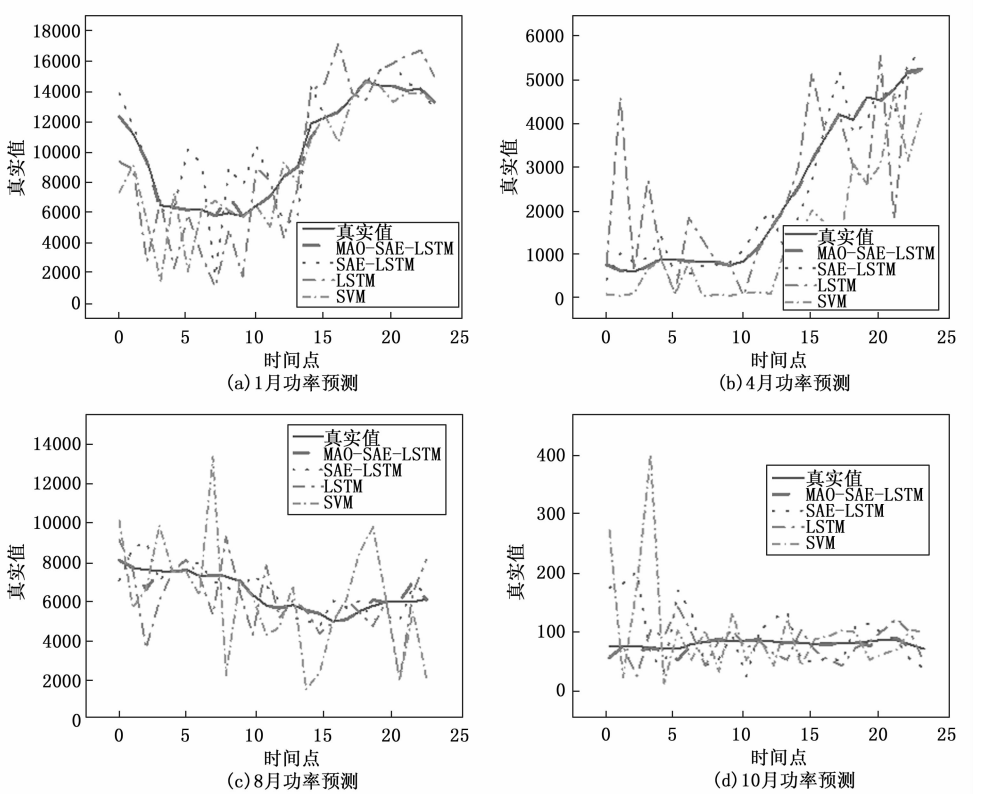

图 6 风力发电预测趋势图

由图 6 可以看出，LSTM 和 SVM 单一模型的预测效果波动较大，SAE-LSTM 模型的组合模型要优于单一模型，更为贴近风电功率的真实值。而加入 MAO 的 SAE-LSTM 预测精度提升明显，几乎贴合于真实值。其中 MAO-SAE-LSTM 模型在此数据集上的预测精度在 1 月和 10 月的预测精度提高了约 15%，在 4 月和 8 月较 SAE-LSTM 更加的平滑，波动更少，与其他对照模型相比，效果显著。通过此应用说明 MAO 具有实际应用能力，提高了对风电数据集的预测精度，进一步验证了 MAO 算法的优越性。

5　结束语

本文将 AO 的搜索空间均匀划分，使用轮盘赌选择法选择初始个体，通过 Hooke-Jeeves 均匀探索搜索空间构建高质量初始种群，改进了随机初始种群的造成的算法不稳定性。其次先加入了一个自适应加权调整了算法前后期的搜索速度，并且将模拟退火概率引入开发阶段以一定概率接受差解，增加了解的多样性，避免陷入局部最优，进一步提高了基本天鹰优化器的开发能力。为了评估 MAO 的性能，使用了 12 个基准函数对算法进行测试。实验结果表明，MAO 的性能明显优于其他算法。然后使用 MAO 优化 SAE-LSTM 模型，进行预测风力发电功率，对比实验证明，MAO-SAE-LSTM 模型具备优异的预测精度。在未来的研究中，将考虑进一步优化天鹰优化器的勘探阶段，加快计算速度和优化精度。并计划将 MAO 继续应用到风力发电功率预测中，为精确预测风电功率、优化风电并网做更深入的研究。

参考文献：

[1] ABUALIGAH L, YOUSRI D, ABD ELAZIZ M, et al. Aquila optimizer: a novel meta-heuristic optimization algorithm [J]. Computers & Industrial Engineering, 2021, 157: 107250.

[2] 龚禹璐，崔龙飞，王典浪，等. 基于多尺度特征提取一改进天鹰算法一长短时神经网络的有载分接开关故障诊断方法 [J/OL]. 现代电力：1-8 [2024-02-08].

[3] 徐亦凤，刘　升，刘宇淞，等. 融合差分变异和切线飞行的天鹰优化器 [J]. 计算机应用研究，2022，39 (10)：2996-3002.

[4] GAO B, SHI Y, XU F, et al. An improved Aquila optimizer based on search control factor and mutations [J]. Processes, 2022, 10 (8): 1451.

[5] PASHAEI E. Mutation-based binary Aquila optimizer for gene selection in cancer classification [J]. Computational Biology and Chemistry, 2022, 101: 107767.

[6] HUANG C, HUANG J, JIA Y, et al. A hybrid Aquila optimizer and its K-means clustering optimization [J]. Transactions of the Institute of Measurement and Control, 2023, 45 (3): 557-572.

[7] 周　玉，裴泽宣，王培崇，等. 引入相量算子和流向算子的天鹰优化算法 [J]. 浙江大学学报（工学版），2024，58 (2)：304-316.

[8] 宗寿松，万俊杰. 基于改进天鹰优化算法的微电网多目标优化调度研究 [J]. 电工电气，2023 (12)：15-22.

[9] 姚天棋，柴　琳，肖　凡，等. 基于改进天鹰优化算法的光伏阵列多峰最大功率跟踪控制 [J]. 热力发电，2023，52 (12)：98-105.

[10] 方　林，蒋晓琳，周庆丰，等. 基于改进天鹰优化器的抱杆结构优化 [J]. 科学技术与工程，2023，23 (27)：11759-11767.

[11] 陈　卓，刘　波，张　源，等. 改进天鹰优化器优化 Elman 神经网络模型预测管道蜡沉积速率 [J]. 科学技术与工程，2023，23 (19)：8179-8186.

[12] KNYPINSKI L. A novel hybrid cuckoo search algorithm for optimization of a line-start PM synchronous motor [J]. Bulletin of the Polish Academy of Sciences-Technical Sciences, 2023, 71 (1): 144586.

[13] CHEN C, LI Y, CAO G, et al. Research on dynamic scheduling model of plant protection UAV based on levy simulated annealing algorithm [J]. Sustainability, 2023, 15 (3): 1772.

[14] ABUALIGAH L, ALMOTAIRI K H. Dynamic evolutionary data and text document clustering approach using improved aquila optimizer based arithmetic optimization algorithm and differential evolution [J]. Neural Computing and Applications, 2022, 34 (23): 20939-20971.

[15] 余秋香. 天鹰优化器的改进及其在多阈值图像分割上的应用 [D]. 南昌：江西理工大学，2023.

[16] 李　钊，袁文浩，任崇广. 基于搜索空间划分与 Canopy K-means 聚类的种群初始化方法 [J]. 控制与决策，2020，35 (11)：2767-2772.

[17] 刘景森，郑智远，李　煜. 一种交互演化改进鲸鱼算法及其收敛性分析 [J]. 控制与决策，2023，38 (1)：75-83.

[18] ZHANG J, HONG L, LIU Q. An improved whale optimization algorithm for the traveling salesman problem [J]. Symmetry, 2020, 13 (1): 48.

[19] 肖辉辉，万常选. 基于多策略的改进花授粉算法 [J]. 软件学报，2021，32 (10)：3151-3175.

[20] LIU Y X, WANG Y J, WANG Q T, et al. Recent advances in data-driven prediction for wind power [J]. Frontiers in Energy Research, 2023, 11: 1204343.

[21] PASHAEI E. Mutation-based binary Aquila optimizer for gene selection in cancer classification [J]. Computational Biology and Chemistry, 2022, 101: 107767.