

## 基于多策略的改进蜜獾算法及其应用

向海昀<sup>1</sup>, 李鸿鑫<sup>1</sup>, 符晓<sup>2</sup>, 苏小平<sup>1</sup>

(1.西南石油大学 计算机科学学院, 成都 610500; 2.西南石油大学 网络与信息化中心, 成都 610500)

**摘要:** 蜜獾算法(HBA)是一种新型智能优化算法,通过模拟蜜獾觅食行为进行寻优,具有结构简单且收敛速度快等特点。针对HBA在解决高维复杂问题时收敛精度低、收敛速度慢以及全局寻优能力不足等问题,提出一种多策略改进的蜜獾算法(MSHBA)。设计一种限制反向学习机制,随着算法迭代生成限制反向解更新种群,提高种群质量,加快算法收敛速度,引入自适应权重因子,随着迭代次数的变化调节不同寻优路径上的寻优步长,协调算法不同探索阶段,提升算法稳定性,加快收敛速度,构建一种新的饥饿搜索策略,根据种群能量以及全局最差位置改变寻优路径上的寻优步长,避免算法陷入早熟。基于9个标准测试函数对MSHBA、HBA、鲸鱼优化、哈里斯鹰、单一策略等算法在不同维度上进行仿真实验,结果表明,MSHBA具有更优的稳定性和收敛精度,将算法应用于机械设计优化问题并进行结果比较,MSHBA对比原HBA性能优化了88%,适用于求解高维复杂问题。

**关键词:** 蜜獾算法;限制反向学习机制;自适应权重因子;饥饿搜索策略;机械设计

开放科学(资源服务)标志码(OSID):



源代码链接: <https://paste.ubuntu.com/p/c8znCZ6qGj/>

中文引用格式: 向海昀, 李鸿鑫, 符晓, 等. 基于多策略的改进蜜獾算法及其应用[J]. 计算机工程, 2023, 49(12): 78-87.

英文引用格式: XIANG H J, LI H X, FU X, et al. Improved honey badger algorithm based on multi-strategy and its applications[J]. Computer Engineering, 2023, 49(12): 78-87.

## Improved Honey Badger Algorithm Based on Multi-Strategy and Its Applications

XIANG Haiyun<sup>1</sup>, LI Hongxin<sup>1</sup>, FU Xiao<sup>2</sup>, SU Xiaoping<sup>1</sup>

(1.School of Computer Science, Southwest Petroleum University, Chengdu 610500, China;

2.Network and Information Center, Southwest Petroleum University, Chengdu 610500, China)

**[Abstract]** The Honey Badger Algorithm(HBA) is a new type of intelligent optimization algorithm that simulates the foraging behavior of honey badgers. It has the characteristics of a simple structure and fast convergence speed. A multi-strategy improved Honey Badger algorithm(MSHBA) is proposed to address the issues of low convergence accuracy, slow convergence speed, and insufficient global optimization ability of the HBA to solve high-dimensional complex problems. It designs a restricted reverse learning mechanism that updates the population with restricted reverse solutions generated through algorithm iteration, improved population quality, and accelerated algorithm convergence speed. MSHBA introduces adaptive weight factors to adjust the optimization step size for different optimization paths as the number of iterations changes, thus coordinating different exploration stages of the algorithm, improving stability, and accelerating convergence speed; and construct a new hungry search strategy that changes the optimization step size for the optimization path based on population energy and global worst-case position to prevent premature convergence. Based on nine standard test functions, simulation experiments are conducted on the MSHBA, HBA, Whale Optimization, Harris Hawks, and single strategy in different dimensions. The results show that the MSHBA has better stability and convergence accuracy. The algorithm is applied to mechanical design optimization problems and the results are compared. Compared with the original HBA, the MSHBA achieved 88% performance optimization, confirming its suitability for solving high-dimensional complex problems.

**[Key words]** Honey Badger Algorithm(HBA); restricted reverse learning mechanism; adaptive weight factor; hungry search strategy; mechanical design

DOI: 10. 19678/j. issn. 1000-3428. 0066465

基金项目: 国家自然科学基金(61503312)。

作者简介: 向海昀(1982—), 男, 高级实验师、硕士, 主研方向为实验室建设与管理、大数据分析、元启发式智能算法; 李鸿鑫、符晓、苏小平, 硕士研究生。

收稿日期: 2022-12-08

修回日期: 2023-02-21

E-mail: 925484644@qq.com

## 0 概述

全局优化问题存在于数学、医学、经济、工程实践等领域中。在实际问题中具有许多高维度、多极值、非线性等复杂全局优化的情况。传统优化算法难以得到理想的优化效果,而智能优化算法则提供了优化的解决方法。群体智能优化算法搜索单元使用模拟的生物集体智能和社会智能进行导航,是目前快速发展的一种智能技术<sup>[1-2]</sup>。典型的群体智能优化算法有粒子群优化(Particle Swarm Optimization, PSO)算法<sup>[3]</sup>、遗传算法(Genetic Algorithm, GA)<sup>[4]</sup>、哈里斯鹰优化(Harris Hawks Optimization, HHO)算法<sup>[5]</sup>、正余弦优化算法(SCA)<sup>[6]</sup>、鲸鱼优化算法(Whale Optimization Algorithm, WOA)<sup>[7]</sup>等。

蜜獾算法(Honey Badger Algorithm, HBA)是HASHIM等<sup>[8]</sup>提出的一种群体智能优化算法,通过模拟蜜獾觅食行为进行寻优,因其与其他算法仿真实验结果对比表现优异,且算法结构简单,因此具有较好的研究价值。虽然HBA已被证明在收敛速度和探索-开发平衡方面优于模拟退火(Simulated Annealing, SA)算法、PSO算法、蚱蜢优化算法等,但是在解决高维复杂问题时,其寻优能力和稳定性等仍需改进,可将其他策略与其相结合以提高算法性能。国内外对HBA算法在这些问题上的改进研究较少。在对群体智能优化算法改进的领域中,研究人员常将随机学习、反向学习、不同算法与所研究算法相结合,进而提高原算法的寻优性能。如董海等<sup>[9]</sup>提出一种结合差分进化原则的蜜獾算法,增强算法的寻优能力。刘琨等<sup>[10]</sup>在WOA中引入精英反向学习策略初始化种群,以提高种群的质量,利用纵横交叉策略提高种群多样性,防止算法陷入早熟。孟宪猛等<sup>[11]</sup>提出一种新的WOA,利用精英反向学习初始化种群,提高初始解的质量,加入Lévy飞行策略,精确搜索范围,避免算法出现早熟。DENG<sup>[12]</sup>等将Lévy飞行策略以及GA中的算子与HBA相结合,提高算法处理复杂工程问题时的寻优能力。HAN等<sup>[13]</sup>提出将反向学习与混沌机制引入HBA,增强种群质量,提高原算法的收敛速度。NASSEF等<sup>[14]</sup>在HBA中引入狩猎搜索策略进而提高算法的寻优能力,保持了勘探和开发阶段平稳过渡。LEI等<sup>[15]</sup>提出螺旋搜索机制更新HBA的勘探阶段,从而提高算法全局搜索能力,引入准余弦定律更新原算法中密度因子平衡不同阶段的过渡,设计针孔成像策略,进而提高种群多样性。DÜZENLI等<sup>[16]</sup>利用高斯混沌映射处理HBA中的关键随机变量,提高算法搜索能

力,加入对立学习,使问题的搜索空间边界根据对立学习中随机和相反候选解的适应度值而变化,避免算法陷入局部最优。ALSHATHRI等<sup>[17]</sup>将HBA与二维Hénon映射相结合,以提高算法的基本性能,引入基于量子的优化技术用于提高算法搜索能力,平衡勘探和开发2个阶段的过渡。ABD ELAZIZ等<sup>[18]</sup>在HBA中引入Lévy飞行策略,在医学图像领域加快特征选择,有效平衡HBA的不同寻优阶段。YASEAR等<sup>[19]</sup>在HBA中加入反向学习机制,提高种群多样性,用于解决电力消耗成本中的复杂问题。王梓辰等<sup>[20]</sup>提出一种自适应动态WOA,引入自适应惯性权重系数、非线性收敛因子,平衡全局与局部的过渡,引入广义反向学习机制避免算法陷入早熟。研究人员提出一种基于混沌收敛因子和惯性权重的改进WOA<sup>[21]</sup>,利用混沌收敛因子和惯性权重更新算法寻优路径,对寻优过程中最优个体进行多项式变异,避免算法陷入局部最优。

针对HBA算法存在的改进空间,本文提出一种多策略改进的蜜獾算法(MSHBA)。该算法在挖掘模式和蜂蜜模式不同寻优路径更新的基础上引入限制反向学习机制,提高种群质量,增强算法的寻优性能,引入自适应权重因子更新寻优路径,促进不同搜索阶段的平稳过渡。提出一种新的饥饿搜索策略,进一步增强算法跳出局部最优的能力。

## 1 蜜獾算法

### 1.1 种群初始化

HBA算法的模型是通过模拟蜜獾智能行为建立的,其通过模拟蜜獾捕食行为进行寻优。蜜獾种群大小为 $N$ ,HBA算法将种群在一定上下限范围内进行随机初始化,如式(1)所示:

$$x_i = l_{lb,i} + r_1 \times (u_{ub,i} - l_{lb,i}) \quad (1)$$

其中: $x_i$ 代表种群 $N$ 中第 $i$ 个蜜獾的位置;而 $u_{ub,i}$ 和 $l_{lb,i}$ 分别表示搜索区间的上界和下界; $r_1$ 为 $[0,1]$ 的随机数。

### 1.2 捕食强度

蜜獾的捕食能力与第 $i$ 只蜜獾的距离、猎物的集中强度和气味强度有关。 $I_i$ 是猎物的气味强度。当 $I_i$ 增大时,蜜獾速度随之加快。 $I_i$ 的具体定义如下:

$$I_i = r_2 \times \frac{S}{4\pi d_i^2} \quad (2)$$

其中: $r_2$ 为 $[0,1]$ 的随机数; $S$ 为猎物的集中强度; $d_i$ 表示猎物和第 $i$ 只蜜獾之间的距离。

### 1.3 密度因子

非线性密度因子 $\alpha$ 约束随时间变化的寻优路

径,使得算法演变过程更加稳定。非线性密度因子 $\alpha$ 随迭代次数减少,其定义如式(3)所示:

$$\alpha = C \times \exp\left(\frac{-t}{t_{\max}}\right) \quad (3)$$

其中: $t_{\max}$ 表示最大迭代次数; $C \geq 1$ (默认值设为2)。

#### 1.4 粒子位置更新

HBA算法寻优路径分为挖掘模式和蜂蜜模式,这2种模式具有不同的运动轨迹。

##### 1.4.1 挖掘模式

在挖掘模式中,蜜獾的行动轨迹类似心形,通过式(4)可以模拟行动路线轨迹:

$$x_{\text{new}} = x_{\text{prey}} + F_1 \times \beta \times I \times x_{\text{prey}} + F_1 \times r_3 \times \alpha \times d_i \times \left[ \cos(2\pi r_4) \times [1 - \cos(2\pi r_5)] \right] \quad (4)$$

其中: $x_{\text{new}}$ 代表个体的新位置; $x_{\text{prey}}$ 代表猎物在环境中的最佳位置; $\beta \geq 1$ (默认值为6)反映蜜獾个体对于食物的获取能力; $F_1$ 作为搜索方向不断改变的标志; $r_3$ 、 $r_4$ 和 $r_5$ 表示0~1之间3个随机的取值。 $F_1$ 的定义如式(5)所示:

$$F_1 = \begin{cases} 1, & r_6 \leq 0.5 \\ -1, & \text{其他} \end{cases} \quad (5)$$

其中: $r_6$ 属于0~1的随机数。

蜜獾在挖掘模式中的搜索路径主要受猎物的距离 $d_i$ 、猎物的嗅觉强度 $I_i$ 和非线性密度因子 $\alpha$ 的影响。

##### 1.4.2 蜂蜜模式

在蜂蜜模式中,蜜獾改变寻优路径,跟随蜂蜜向导鸟到达目标蜂巢,其路线轨迹的模拟如式(6)所示:

$$x_{\text{new}} = x_{\text{prey}} + F_1 \times r_7 \times \alpha \times d_i \quad (6)$$

其中: $r_7$ 为0~1的随机数; $\alpha$ 和 $F_1$ 由式(3)和式(5)确定。在该阶段,蜜獾的寻优路径与猎物和第 $i$ 只蜜獾之间的距离 $d_i$ 、猎物的最佳位置 $x_{\text{prey}}$ 以及 $\alpha$ 有关,蜜獾在 $x_{\text{prey}}$ 附近进行探索。

## 2 多策略改进的蜜獾算法

### 2.1 HBA存在的问题与改进策略

在实验中发现原HBA算法在解决高维复杂问题中存在收敛速度慢、难以有效寻优等问题。通过分析发现,非线性密度因子 $\alpha$ 随着迭代次数增加而减少,在迭代中期,当 $\alpha$ 的系数 $C$ 默认为2时, $\alpha$ 将降为1,密度因子较低影响全局寻优能力,算法容易陷入早熟。

为解决以上问题,本文通过加入限制反向学习机制以增强种群质量、多样性,加入自适应权重因子

以平衡全局和局部搜索,提升算法寻优性能、稳定性,加入饥饿搜索策略,增强算法搜索能力。本文基于以上对HBA算法的改进,提出一种多策略改进的蜜獾算法。

### 2.2 限制反向学习机制

在蜜獾算法中,蜜獾个体以猎物的气味强度以及猎物距离和随时间变化的非线性密度因子 $\alpha$ 为导向,通过迭代不断寻找最优解,但是当最优解为非全局最优解时,算法寻优结果错误使得算法陷入早熟。针对算法早熟问题,李安东等<sup>[22]</sup>将精英反向学习策略加入到优化算法中,提高种群质量。通过比较原解与反向学习生成的反向解,选择更优个体继续加入种群,使种群范围得到扩大<sup>[23]</sup>。本文在挖掘模式和蜂蜜模式不同路径更新的基础上引入限制反向学习机制,在算法迭代时,对当前的种群个体进行限制反向学习,生成新的限制反向解,提升解的质量,增加种群多样性,加快算法收敛速度。限制反向学习机制表达式如式(7)所示:

$$\bar{x}_i = r_{\text{rand}} \times (u_{\text{ub}} + l_{\text{lb}} - x_i) \quad (7)$$

其中: $\bar{x}_i$ 表示限制反向解; $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$ 表示 $n$ 维空间的1个解; $u_{\text{ub}}$ 、 $l_{\text{lb}}$ 表示在第 $t$ 次迭代时整个种群的动态边界, $u_{\text{ub}} = \max(x_i)$ , $l_{\text{lb}} = \min(x_i)$ ; $r_{\text{rand}}$ 为动态的边界伸缩控制量,其值为 $[0, 1]$ 的1个随机数。

限制反向学习机制相比于反向学习,解的上下限随迭代次数变化且限制了反向解越过边界。相比反向学习机制,限制反向学习机制进一步优化种群质量,使得在增加种群多样性的基础上,加快了算法的收敛速度。

### 2.3 自适应权重因子

在粒子群算法中,惯性权重是其中重要的参数之一,当惯性权重较小时,算法搜索范围更精细,当惯性权重较大时,算法搜索范围更大<sup>[24]</sup>。考虑到捕猎过程中猎物目标对蜜獾位置更新的影响以及自适应权重因子可有效平衡算法在不同阶段的搜索能力,本文提出一种自适应权重因子,如式(8)所示:

$$\omega = \tan\left(\frac{t}{t_{\max}}\right) \quad (8)$$

其中: $t$ 为当前迭代次数; $t_{\max}$ 为最大迭代数。

引入自适应权重因子,HBA中挖掘模式和蜂蜜模式搜索路线的式(4)和式(6)分别更新为式(9)和式(10):

$$x_{\text{new}} = x_{\text{prey}} \times \omega + F_1 \times \beta \times I \times x_{\text{prey}} + F_1 \times r_3 \times \alpha \times d_i \times \left[ \cos(2\pi r_4) \times [1 - \cos(2\pi r_5)] \right] \quad (9)$$

$$x_{\text{new}} = x_{\text{prey}} \times \omega + F_1 \times r_7 \times \alpha \times d_i \quad (10)$$

$\omega$ 随着迭代次数的增加,其值呈非线性增加。在算法迭代初期自适应权重因子的值较小,目标位置模糊,对原蜜獾算法中挖掘模式和蜂蜜模式搜索路线的更新影响不大;在算法迭代的中后期,目标逐渐清晰,此时自适应权重因子的值随迭代次数的增加逐渐变大,目标对蜜獾个体的吸引更强,自适应权重因子增强了局部开发的能力,使得蜜獾能够在算法迭代的中后期更快地找到目标解。加入自适应权重因子能够促进算法全局和局部搜索平稳过渡,提升算法寻优性能和稳定性。

## 2.4 饥饿搜索策略

在蜂蜜搜索阶段,蜜獾在目前发现的猎物位置 $x_{\text{prey}}$ 附近进行搜索。在算法迭代的中后期,算法可能出现早熟情况,由哈里斯鹰优化算法中的逃逸能量理论得到启发,本文提出一种新的饥饿搜索策略。在算法寻优过程中,当蜜獾表现出能量较低、适应度较差时,蜜獾处于饥饿状态,通过改变蜜獾的寻优路径增强搜索能力,从而避免算法陷入局部最优。在蜂蜜模式中,当蜜獾种群数量迭代超过1/2时,引入饥饿搜索策略对原蜜獾寻优路径进行改变。

饥饿搜索策略表达式如下:

$$x_{\text{new}} = x_{\text{prey}} \times \omega + F_1 \times \alpha \times d_i \times r_7 \times \exp(x_{\text{worse}} - x_i) \quad (11)$$

其中: $x_{\text{worse}}$ 表示全局最差的位置; $x_i$ 为当前种群个体的位置。

饥饿搜索策略基于全局最差位置对原搜索路径进行改进,在蜜獾处于饥饿状态时,使其在原搜索路径的基础上实现更广泛的搜索,增强跳出局部最优的能力。

## 2.5 MSHBA算法描述

综上所述,MSHBA执行步骤采用伪代码表述,如算法1所示。

### 算法1 MSHBA算法

输入 相关参数 $t_{\text{max}}, N, \beta, C$

输出  $x_{\text{prey}}$

1. 随机初始化种群及位置
2. 计算个体适应度值 $f_i$ ,记录最佳位置 $x_{\text{prey}}$ 以及该位置的适应值 $f_{\text{prey}}$ 、最差位置 $x_{\text{worse}}$
3. While  $t \leq t_{\text{max}}$
4. for  $i = 1$  to  $N$  do
5. 由式(7)计算限制反向解,并根据原始解与限制反向解的适应度值,更新种群
6. end for
7. 使用式(3)更新递减系数 $\alpha$
8. for  $i=1$  to  $N$  do
9. 使用式(2)计算强度 $I_i$
10. if  $r < 0.5$  then  $r$ 是介于0和1之间的随机数
11. 使用式(9)更新位置 $x_{\text{new}}$

12. else
13. 使用式(10)更新位置 $x_{\text{new}}$
14. if  $i > N/2$  then
15. 使用式(11)更新位置 $x_{\text{new}}$
16. end if
17. end if
18. 记录新的适应度值 $f_{\text{new}}$
19. if  $f_{\text{new}} \leq f_i$  then
20.  $x_i = x_{\text{new}}, f_i = f_{\text{new}}$
21. end if
22. if  $f_{\text{new}} \leq f_{\text{prey}}$  then
23.  $f_{\text{prey}} = f_{\text{new}}, x_{\text{prey}} = x_{\text{new}}$
24. end if
25. end for
26. end While
27. Return  $x_{\text{prey}}$

## 3 实验结果与分析

### 3.1 与其他优化算法性能对比

本文分别选取PSO<sup>[3]</sup>、GA<sup>[4]</sup>、HHO<sup>[5]</sup>、SCA<sup>[6]</sup>、WOA<sup>[7]</sup>以及HBA、MSHBA进行仿真实验分析。为直接对比各算法性能,本文将所选取各算法的最大迭代次数均设为500,种群数量均设为30,算法维度分别为30、200、500,算法具体参数如表1所示。

表1 算法仿真实验参数

Table 1 Simulation experiment parameters of algorithms

优化算法	参数
PSO	$w=1, w_{\text{damp}}=0.99, c_1=1.5, c_2=2$
GA	$P_c=0.7, P_m=0.3$
HHO	$J=[0,2]$
SCA	$a=2$
WOA	$b=1$
HBA	$\beta=6, C=2$
MSHBA	$\beta=6, C=2$

为测试本文所提的MSHBA与原HBA、WOA等算法的性能差异,本文选用9个测试函数进行仿真对比实验,其中 $F_1 \sim F_4$ 为单峰函数, $F_5 \sim F_7$ 为不定维多峰函数, $F_8 \sim F_9$ 为固定维多峰函数。该实验采用标准差值(Std)、平均值(Mean)和最优值(Best)作为评价指标<sup>[25]</sup>,进而分析各算法的性能差别。具体测试函数信息如表2所示。

本文使用MATLAB仿真软件对每个函数在不同算法的应用下进行30次实验,得出与各算法对应不同函数评价指标的结果,如表3所示,加粗表示最优数据。所测试算法在各函数200维度上的收敛曲线如图1所示。

表2 测试函数及具体信息

Table 2 Test functions and specific information

函数	搜索区间	最小值
$F_1(x) = \sum_{i=1}^n x_i^2$	$[-100, 100]$	0.000 0
$F_2(x) = \sum_{i=1}^{30}  x_i  + \prod_{i=1}^{30}  x_i $	$[-10, 10]$	0.000 0
$F_3(x) = \sum_{i=1}^n \left( \sum_{j=1}^i x_j \right)^2$	$[-100, 100]$	0.000 0
$F_4(x) = \max\{ x_i , 1 \leq i \leq 30\}$	$[-600, 600]$	0.000 0
$F_5(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$	$[-5.12, 5.12]$	0.000 0
$F_6(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{30} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{30} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + c$	$[-32, 32]$	0.000 0
$F_7(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]$	0.000 0
$F_8(x) = \sum_{i=1}^{11} \left[ a_i - \frac{x_i(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	$[-5, 5]$	0.000 3
$F_9(x) = \left[ 1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2) \right] \times$ $\left[ 30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2) \right]$	$[-2, 2]$	3.000 0

表3 MSHBA与对比算法在不同函数和维度上的优化性能指标

Table 3 Optimization performance indicators of MSHBA and comparison algorithms on different functions and dimensions

函数	维度	指标	GA	PSO	WOA	HHO	SCA	HBA	MSHBA
$F_1$	30	Mean	$9.55 \times 10^3$	2.68	$7.04 \times 10^{-73}$	$9.58 \times 10^{-97}$	5.94	$5.27 \times 10^{-135}$	<b>0</b>
		Std	$5.24 \times 10^3$	1.13	$3.83 \times 10^{-72}$	$5.00 \times 10^{-96}$	9.31	$1.58 \times 10^{-134}$	<b>0</b>
		Best	$1.93 \times 10^3$	1.08	$7.69 \times 10^{-89}$	$2.32 \times 10^{-114}$	$4.20 \times 10^{-2}$	$3.23 \times 10^{-143}$	<b>0</b>
	200	Mean	$5.31 \times 10^5$	$1.20 \times 10^3$	$2.77 \times 10^{-70}$	$4.53 \times 10^{-95}$	$5.15 \times 10^4$	$8.78 \times 10^{-117}$	<b>0</b>
		Std	$3.14 \times 10^4$	$1.10 \times 10^2$	$1.46 \times 10^{-69}$	$1.75 \times 10^{-94}$	$2.52 \times 10^4$	$4.28 \times 10^{-116}$	<b>0</b>
		Best	$4.21 \times 10^5$	$9.40 \times 10^2$	$9.99 \times 10^{-83}$	$5.92 \times 10^{-112}$	$1.40 \times 10^4$	$2.80 \times 10^{-124}$	<b>0</b>
	500	Mean	$1.49 \times 10^6$	$1.26 \times 10^4$	$1.10 \times 10^{-69}$	$1.39 \times 10^{-91}$	$1.96 \times 10^5$	$6.86 \times 10^{-113}$	<b>0</b>
		Std	$4.68 \times 10^4$	$6.14 \times 10^2$	$4.14 \times 10^{-69}$	$7.35 \times 10^{-91}$	$7.26 \times 10^4$	$2.34 \times 10^{-112}$	<b>0</b>
		Best	$1.38 \times 10^6$	$1.13 \times 10^4$	$1.44 \times 10^{-82}$	$1.80 \times 10^{-111}$	$2.84 \times 10^4$	$1.66 \times 10^{-119}$	<b>0</b>
$F_2$	30	Mean	$4.16 \times 10$	4.25	$4.65 \times 10^{-52}$	$7.24 \times 10^{-52}$	$1.34 \times 10^{-2}$	$3.89 \times 10^{-72}$	<b><math>2.91 \times 10^{-243}</math></b>
		Std	7.42	1.23	$1.06 \times 10^{-51}$	$2.08 \times 10^{-51}$	$1.42 \times 10^{-2}$	$1.17 \times 10^{-71}$	<b>0.00</b>
		Best	$2.38 \times 10$	1.98	$6.50 \times 10^{-58}$	$3.78 \times 10^{-58}$	$5.64 \times 10^{-4}$	$9.40 \times 10^{-76}$	<b><math>3.06 \times 10^{-269}</math></b>
	200	Mean	$5.21 \times 10^2$	$7.19 \times 10^{35}$	$2.97 \times 10^{-49}$	$1.11 \times 10^{-49}$	$2.79 \times 10$	$1.23 \times 10^{-62}$	<b><math>7.23 \times 10^{-223}</math></b>
		Std	$2.43 \times 10$	$3.93 \times 10^{36}$	$1.01 \times 10^{-48}$	$3.77 \times 10^{-49}$	$1.73 \times 10$	$3.23 \times 10^{-62}$	<b>0.00</b>
		Best	$4.81 \times 10^2$	$7.27 \times 10^{02}$	$4.67 \times 10^{-55}$	$2.86 \times 10^{-57}$	3.92	$1.08 \times 10^{-65}$	<b><math>2.82 \times 10^{-246}</math></b>
	500	Mean	$1.19 \times 10^{145}$	$3.62 \times 10^{143}$	$3.53 \times 10^{-48}$	$1.09 \times 10^{-49}$	$1.09 \times 10^2$	$9.76 \times 10^{-61}$	<b><math>2.42 \times 10^{-210}</math></b>
		Std	$6.53 \times 10^{145}$	$1.98 \times 10^{144}$	$1.61 \times 10^{-47}$	$5.19 \times 10^{-49}$	$6.39 \times 10$	$2.06 \times 10^{-60}$	<b>0.00</b>
		Best	$5.09 \times 10^{96}$	$7.25 \times 10^{58}$	$4.98 \times 10^{-56}$	$6.89 \times 10^{-59}$	$2.25 \times 10$	$1.10 \times 10^{-64}$	<b><math>2.10 \times 10^{-250}</math></b>



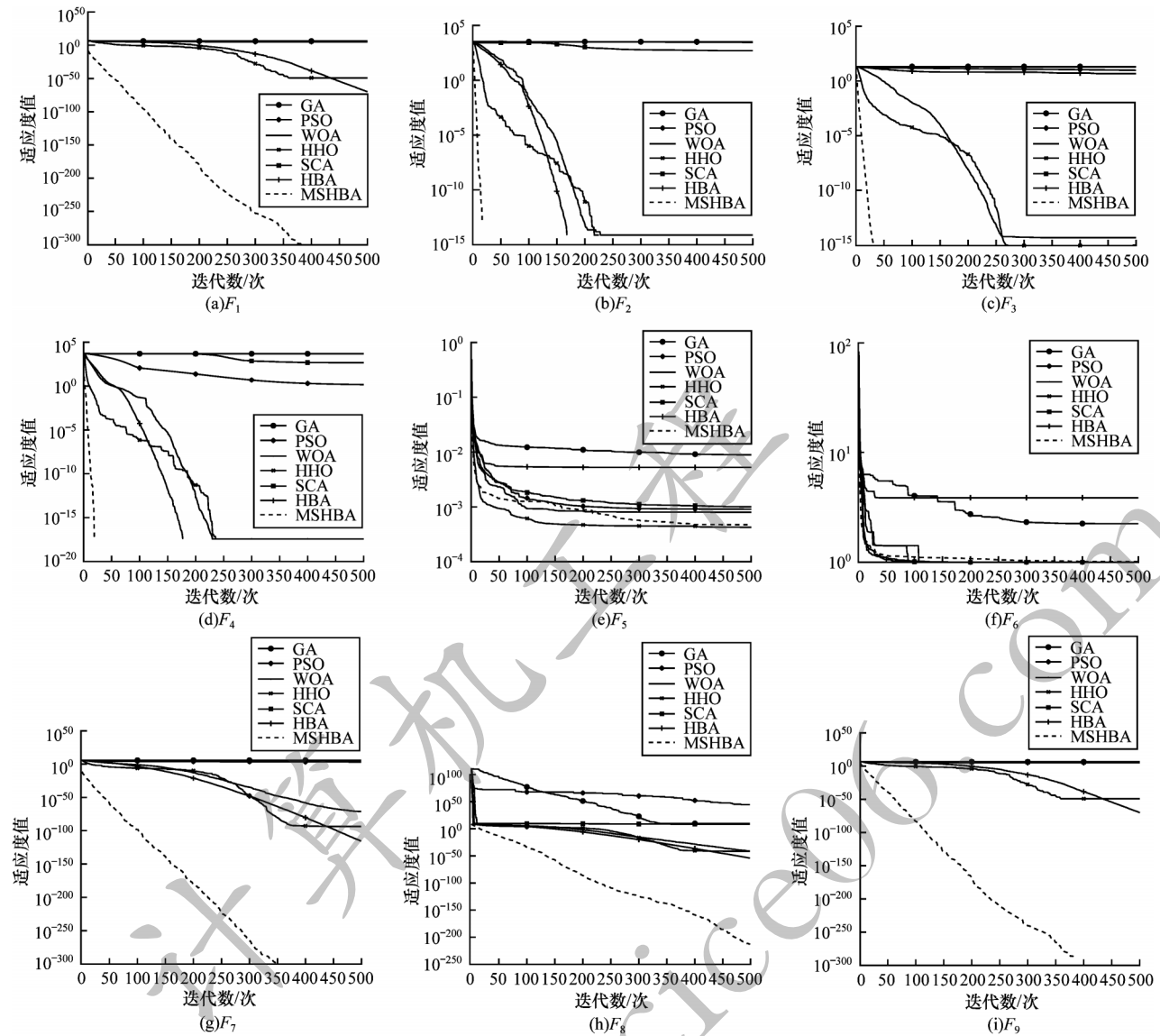


图1 在维度为200时各算法的收敛曲线

Fig.1 Convergence curves of each algorithms at dimension 200

从同一维度测试函数  $F_1 \sim F_4$  的仿真结果可以看出,MSHBA在算法的收敛精度和稳定性上都表现出比 PSO、GA、HHO、SCA、WOA 以及 HBA 更好的性能。在  $F_1$  中,仅有 MSHBA 取得了理论最优值,虽然 HBA 同样表现出良好的稳定性,但在寻优精度上明显不足,MSHBA 改进了其寻优性能。在  $F_2 \sim F_4$  测试结果中,MSHBA 在所测算法中的各评价指标均达到了最佳,相比 HBA 和 WOA 等算法提高了多个数量级,说明 MSHBA 具有更优的稳定性和寻优能力。在  $F_5$ 、 $F_6$  和  $F_7$  测试结果中,所测 HHO、HBA、MSHBA 算法寻优能力差异不大,均表现出了相对其他测试算法更优的稳定性。特别是在  $F_6$  测试结果中,所测算法在寻优时都没能找到全局最优,但是从收敛精度分析,HBA、MSHBA、HHO 均表现出远优于其他测试算法的收敛精度。在  $F_8$  测试结果中,HHO、HBA、MSHBA 均表现出优异的寻优能力,就稳定性而言,MSHBA 对比原 HBA 算法有较大的提

升,平均值最接近表现最好的 HHO。在  $F_9$  测试结果中,多个测试算法均达到最优值,MSHBA 对原 HBA 算法稳定性有了更好的优化。

在维度为 30~500 的变化过程中,所测试算法的寻优性能均有所下降,在单峰测试函数  $F_1 \sim F_4$  测试结果中,即使算法维度达到 500,MSHBA 依然表现出最优的寻优能力和稳定性。在不定维多峰函数测试结果中, $F_5$  和  $F_7$  测试结果表明,随着维度增大,MSHBA 与 HBA、HHO 依然可以得到全局最优解。在  $F_6$  中,SCA 等算法随着维度增加,寻优性能有所下降,MSHBA 则依然可以保持较优的寻优性能。但是所测算法均未得出最优解,MSHBA 相较于 HBA 表现更优的稳定性。

### 3.2 不同改进策略的优化性能分析

本文基于 HBA 提出 3 种不同的改进策略,分别为限制反向学习机制、自适应权重因子、饥饿搜索策略。为研究 3 种单一策略以及混合策略对原 HBA 寻优性能的影响,本文基于 5 种算法对上述 9 个测试函

数进行仿真实验分析。将 HBA、MSHBA 与只采用限制反向学习机制的 HBA(记作 HBA-1)、只采用自适应权重因子的 HBA(记作 HBA-2)、只采用饥饿搜

索策略的 HBA(记作 HBA-3)进行对比,5 种算法参数设置与表 1 中 HBA 参数设置相同,5 种算法在测试函数维度为 30 时的寻优性能结果如表 4 所示。

表 4 在维度为 30 时不同算法性能对比

Table 4 Performance comparison among different algorithms at dimension 30

函数	指标	HBA	HBA-1	HBA-2	HBA-3	MSHBA
$F_1$	Mean	$4.65 \times 10^{-134}$	<b>0.00</b>	<b>0.00</b>	$2.25 \times 10^{-135}$	<b>0.00</b>
	Std	$2.44 \times 10^{-133}$	<b>0.00</b>	<b>0.00</b>	$1.08 \times 10^{-134}$	<b>0.00</b>
	Best	$1.31 \times 10^{-142}$	<b>0.00</b>	<b>0.00</b>	$6.92 \times 10^{-143}$	<b>0.00</b>
$F_2$	Mean	$3.96 \times 10^{-71}$	$1.33 \times 10^{-236}$	$1.72 \times 10^{-174}$	$1.45 \times 10^{-72}$	$5.97 \times 10^{-242}$
	Std	$2.12 \times 10^{-70}$	<b>0.00</b>	<b>0.00</b>	$3.50 \times 10^{-72}$	<b>0.00</b>
	Best	$4.68 \times 10^{-75}$	$2.19 \times 10^{-245}$	$2.69 \times 10^{-192}$	$8.46 \times 10^{-76}$	$4.18 \times 10^{-274}$
$F_3$	Mean	$6.36 \times 10^{-94}$	<b>0.00</b>	$1.52 \times 10^{-308}$	$1.40 \times 10^{-97}$	<b>0.00</b>
	Std	$3.49 \times 10^{-93}$	<b>0.00</b>	<b>0.00</b>	$4.88 \times 10^{-97}$	<b>0.00</b>
	Best	$1.08 \times 10^{-106}$	<b>0.00</b>	<b>0.00</b>	$1.29 \times 10^{-108}$	<b>0.00</b>
$F_4$	Mean	$2.64 \times 10^{-56}$	$1.72 \times 10^{-271}$	$2.67 \times 10^{-144}$	$2.73 \times 10^{-56}$	$8.77 \times 10^{-273}$
	Std	$1.03 \times 10^{-55}$	<b>0.00</b>	$1.42 \times 10^{-143}$	$7.32 \times 10^{-56}$	<b>0.00</b>
	Best	$7.92 \times 10^{-60}$	$3.01 \times 10^{-282}$	$3.80 \times 10^{-185}$	$6.82 \times 10^{-61}$	$2.96 \times 10^{-287}$
$F_5$	Mean	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
	Std	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
	Best	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
$F_6$	Mean	$6.65 \times 10^{-1}$	$8.88 \times 10^{-16}$	$8.88 \times 10^{-16}$	$8.88 \times 10^{-16}$	$8.88 \times 10^{-16}$
	Std	3.64	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
	Best	$8.88 \times 10^{-16}$	$8.88 \times 10^{-16}$	$8.88 \times 10^{-16}$	$8.88 \times 10^{-16}$	$8.88 \times 10^{-16}$
$F_7$	Mean	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
	Std	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
	Best	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
$F_8$	Mean	$6.65 \times 10^{-3}$	$5.66 \times 10^{-4}$	$5.68 \times 10^{-4}$	$5.29 \times 10^{-3}$	$4.63 \times 10^{-4}$
	Std	$9.80 \times 10^{-3}$	$6.15 \times 10^{-4}$	$3.50 \times 10^{-4}$	$8.66 \times 10^{-3}$	$1.30 \times 10^{-4}$
	Best	$3.07 \times 10^{-4}$	$3.07 \times 10^{-4}$	$3.09 \times 10^{-4}$	$3.07 \times 10^{-4}$	$3.07 \times 10^{-4}$
$F_9$	Mean	4.80	3.90	<b>3.00</b>	3.90	<b>3.00</b>
	Std	6.85	4.93	$5.73 \times 10^{-3}$	4.93	$5.52 \times 10^{-3}$
	Best	<b>3.00</b>	<b>3.00</b>	<b>3.00</b>	<b>3.00</b>	<b>3.00</b>

从表 4 可以看出,MSHBA 对 HBA 寻优性能的改进比单一策略的改进效果更明显,对比原 HBA,所有策略都能对其进行有效改进,其中 HBA-3 改进效果略差,HBA-1 和 HBA-2 对 HBA 均有较优的改进效果。

综上所述,MSHBA 在单峰函数性能测试中,表现出最佳的收敛精度和稳定性。在不定维多峰函数性能测试中,MSHBA 同样具有较优的寻优性能。在固定维多峰函数性能测试中,MSHBA 提升了原 HBA 的算法稳定性。对比单一改进策略,MSHBA 表现出更佳的性能。

#### 4 基于 MSHBA 的机械设计优化

针对工程设计与应用领域中常出现数值约束问题,传统的寻优方法(如梯度下降法)效率低、极易陷入局部最优,且难以解决更高维以及非线性的数值优化问题。本文将 MSHBA 算法用于机械设计问题的优化,进而对算法的可行性做进一步检验<sup>[26]</sup>。本文主要优化的机械设计问题是活塞杆设计问题。将粒子群优化算法、鲸鱼优化算法、遗传算法、正余弦优化算法、蜜獾算法以及 MSHBA 应用于活塞杆设计问题优化,并进行仿真实验。各算法具体参数如

表 1 所示,种群数量均为 50,最大迭代次数均为 1 000,每个算法独立运行 30 次后取平均值。

优化活塞杆设计问题的主要目的是在活塞杆从 0° 提升至 45° 时,通过最小化油量定位活塞部件 H、B、D 和 X,如图 2 所示。

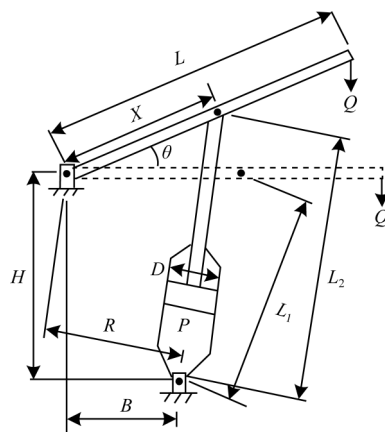


图 2 活塞杆结构问题

Fig.2 Structure problem of piston rod

该活塞杆设计的数学模型如下:

$$\text{minimize } f(H, B, D, X) = \frac{1}{4} \pi D^2 (L_2 - L_1)$$

$$\text{subject to } \begin{cases} G_1 = QL \cos \theta - RF \leq 0 \\ G_2 = Q(L - X) - M_{\max} \leq 0 \\ G_3 = 1.2(L_2 - L_1) - L_1 \leq 0 \\ G_4 = \frac{D}{2} - B \leq 0 \end{cases} \quad (12)$$

其中:

$$\begin{cases} R = \frac{-X(X \sin \theta + H) + H(B - X \cos \theta)}{\sqrt{(X - B)^2 + H^2}}, F = \frac{\pi PD^2}{4} \\ L_1 = \sqrt{(X - B)^2 + H^2} \\ L_2 = \sqrt{(X \sin \theta + H)^2 + (B - X \cos \theta)^2} \\ \theta = 45^\circ, Q = 10\,000 \text{ lbs}, M = 1.8 \times 10^6 \text{ lbs} \\ P = 1\,500 \text{ psi}, L = 240 \\ 0.05 \leq H, B, D \leq 500, 0.05 \leq X \leq 120 \end{cases} \quad (13)$$

这些不等式约束考虑了力平衡、杠杆的最大弯矩、最小活塞行程、几何条件等。表5所示为MSHBA和其他测试算法获得的最佳解决方案,据此可知,MSHBA获得的函数最优解 $[H, B, D, X] = [0.05, 2.05, 4.09, 119.95]$ ,  $f(H, B, D, X) = 8.45$ ,表明MSHBA获得最小油量,对比原算法其性能提高了88%。图3所示为测试算法求解活塞杆设计问题的收敛曲线,证明MSHBA能够快速找到活塞杆设计难题的最佳解决方案。

表5 活塞杆设计问题中各算法的优化结果对比  
Table 5 Comparison of optimization results of various algorithms in piston rod design problem

算法	H	B	D	X	f
PSO	500.00	500.00	2.21	60.07	167.72
WOA	134.96	223.40	2.21	71.82	118.43
GA	60.48	92.76	3.89	94.02	129.46
SCA	0.06	2.13	4.13	119.44	9.04
HBA	200.03	201.23	3.33	96.00	72.04
MSHBA	0.05	2.05	4.09	119.95	8.45

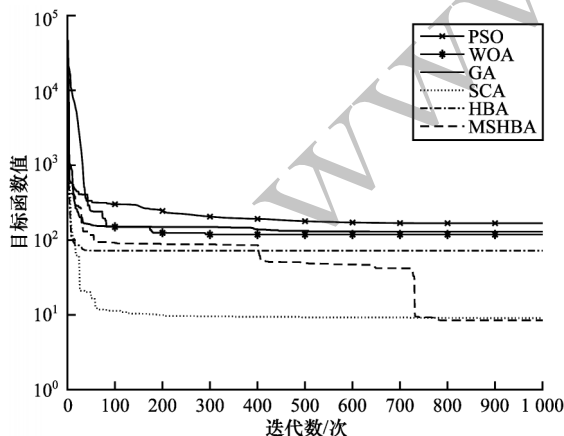


图3 测试算法求解活塞杆设计问题的收敛曲线  
Fig.3 Convergence curves of testing algorithms for piston rod design problem

### 5 结束语

针对蜜獾算法在解决高维复杂问题时存在收敛精度低、稳定性较差的问题,本文提出一种多策略改进的蜜獾算法。该算法引入限制反向学习机制,提高种群质量与多样性及算法寻优性能;引入自适应权重因子,协调全局搜索和局部开发,提升算法稳定性;设计一种新的饥饿搜索策略,防止算法陷入局部最优。为验证本文提出的多策略改进算法的先进性,与经典智能优化算法、新型智能优化算法以及原HBA进行比较,通过对不同算法不同维度在9个测试函数上进行实验,以及将多策略改进的蜜獾算法应用于机械设计优化问题,实验结果表明,本文所提算法在原始HBA算法的基础上,提高了稳定性、收敛速度和收敛精度,可用于解决极复杂的高维问题,具有良好的应用价值。下一步将结合其他优化算法进行性能优化,并将MSHBA应用于组合优化和图像分割,例如车间调度、多级阈值分割等任务。

### 参考文献

[1] 高岳林,杨钦文,王晓峰,等. 新型群体智能优化算法综述[J]. 郑州大学学报(工学版),2022,43(3):21-30.  
GAO Y L, YANG Q W, WANG X F, et al. Overview of new swarm intelligent optimization algorithms[J]. Journal of Zhengzhou University (Engineering Science), 2022, 43(3):21-30. (in Chinese)

[2] 符晓. 云计算中基于共享机制和群体智能优化算法的任务调度方案[J]. 计算机科学,2018,45(S1):290-294.  
FU X. Task scheduling scheme based on sharing mechanism and swarm intelligence optimization algorithm in cloud computing[J]. Computer Science, 2018, 45(S1): 290-294. (in Chinese)

[3] KENNEDY J, EBERHART R C. Particle swarm optimization [C]//Proceedings of IEEE International Conference on Neural Networks. Washington D. C., USA: IEEE Press, 1995:1942-1948.

[4] KATOCH S, CHAUHAN S S, KUMAR V. A review on genetic algorithm: past, present, and future[J]. Multimedia Tools and Applications, 2021, 80(5):8091-8126.

[5] HEIDARI A A, MIRJALILI S, FARIS H, et al. Harris Hawks optimization; algorithm and applications[J]. Future Generation Computer Systems, 2019, 97:849-872.

[6] MIRJALILI S. SCA: a sine cosine algorithm for solving optimization problems[J]. Knowledge-Based Systems, 2016, 96:120-133.

[7] MIRJALILI S, LEWIS A. The whale optimization algorithm[J]. Advances in Engineering Software, 2016, 95: 51-67.

[8] HASHIM F A, HOUSSEIN E H, HUSSAIN K, et al. Honey Badger algorithm; new metaheuristic algorithm for solving optimization problems[J]. Mathematics and Computers in Simulation, 2022, 192:84-110.

[9] 董海,林国栋. 基于改进HBA算法的生鲜闭环供应链网络鲁棒优化设计[J]. 计算机应用研究,2022,39(10):3020-3025.  
DONG H, LIN G D. Robust optimization design of fresh

- closed-loop supply chain network based on improved HBA algorithm[J]. *Application Research of Computers*, 2022, 39(10):3020-3025. (in Chinese)
- [10] 刘琨,赵露露,王辉. 一种基于精英反向和纵横交叉的鲸鱼优化算法[J]. *小型微型计算机系统*, 2020, 41(10):2092-2097.  
LIU K, ZHAO L L, WANG H. Whale optimization algorithm based on elite opposition-based and crisscross optimization[J]. *Journal of Chinese Computer Systems*, 2020, 41(10):2092-2097. (in Chinese)
- [11] 孟宪猛,蔡翠翠. 基于精英反向学习和 Lévy 飞行的鲸鱼优化算法[J]. *电子测量技术*, 2021, 44(20):82-87.  
MENG X M, CAI C C. Whale optimization algorithm based on elite reverse learning and Lévy flight[J]. *Electronic Measurement Technology*, 2021, 44(20):82-87. (in Chinese)
- [12] DENG B. An improved honey badger algorithm by genetic algorithm and Lévy flight distribution for solving airline crew rostering problem[J]. *IEEE Access*, 2022, 10:108075-108088.
- [13] HAN E F, GHADIMI N. Model identification of proton-exchange membrane fuel cells based on a hybrid convolutional neural network and extreme learning machine optimized by improved honey badger algorithm [J]. *Sustainable Energy Technologies and Assessments*, 2022, 52:102005.
- [14] NASSEF A M, HOUSSEIN E H, HELMY B E D, et al. Modified honey badger algorithm based global MPPT for triple-junction solar photovoltaic system under partial shading condition and global optimization [J]. *Energy*, 2022, 254:124363.
- [15] LEI W J, HE Q, YANG L, et al. Solar photovoltaic cell parameter identification based on improved honey badger algorithm[J]. *Sustainability*, 2022, 14(14):8897.
- [16] DÜZENLIT, KUTLU ONAY F, AYDEMİR S B. Improved honey badger algorithms for parameter extraction in photovoltaic models[J]. *Optik*, 2022, 268:169731.
- [17] ALSHATHRI S, ABD ELAZIZ M, YOUSRI D, et al. Quantum chaotic honey badger algorithm for feature selection[J]. *Electronics*, 2022, 11(21):3463.
- [18] ABD ELAZIZ M, MABROUK A, DAHOU A, et al. Medical image classification utilizing ensemble learning and levy flight-based honey badger algorithm on 6G-enabled Internet of Things[J]. *Computational Intelligence and Neuroscience*, 2022, 2022:1-17.
- [19] YASEAR S A, CHANIMI H M A. A modified honey badger algorithm for solving optimal power flow optimization problem [J]. *International Journal of Intelligent Engineering and Systems*, 2022, 15(4):142-155.
- [20] 王梓辰,窦震海,董军,等. 多策略改进的自适应动态鲸鱼优化算法[J]. *计算机工程与设计*, 2022, 43(9):2638-2645.  
WANG Z C, DOU Z H, DONG J, et al. Adaptive dynamic whale optimization algorithm based on multi-strategy improvement[J]. *Computer Engineering and Design*, 2022, 43(9):2638-2645. (in Chinese)
- [21] 王彦琦,张强,朱刘涛,等. 基于改进鲸鱼优化算法的GBDT回归预测模型[J]. *吉林大学学报(理学版)*, 2022, 60(2):401-408.  
WANG Y Q, ZHANG Q, ZHU L T, et al. GBDT regression prediction model based on improved whale optimization algorithm[J]. *Journal of Jilin University( Science Edition)*, 2022, 60(2):401-408. (in Chinese)
- [22] 李安东,刘升. 混合策略改进鲸鱼优化算法[J]. *计算机应用研究*, 2022, 39(5):1415-1421.  
LI A D, LIU S. Multi-strategy improved whale optimization algorithm[J]. *Application Research of Computers*, 2022, 39(5):1415-1421. (in Chinese)
- [23] 耿召里,李目,曹淑睿,等. 基于混合反向学习策略的鲸鱼优化算法[J]. *计算机工程与科学*, 2022, 44(2):355-363.  
GENG Z L, LI M, CAO S R, et al. A whale optimization algorithm based on hybrid reverse learning strategy[J]. *Computer Engineering & Science*, 2022, 44(2):355-363. (in Chinese)
- [24] 徐航,张达敏,王依柔,等. 基于自适应决策算子的鲸鱼优化算法[J]. *智能计算机与应用*, 2020, 10(9):6-11.  
XU H, ZHANG D M, WANG Y R, et al. Whale optimization algorithm based on adaptive decision operator[J]. *Intelligent Computer and Applications*, 2020, 10(9):6-11. (in Chinese)
- [25] 何庆,林杰,徐航. 混合柯西变异和均匀分布的蝗虫优化算法[J]. *控制与决策*, 2021, 36(7):1558-1568.  
HE Q, LIN J, XU H. Hybrid Cauchy mutation and uniform distribution of grasshopper optimization algorithm [J]. *Control and Decision*, 2021, 36(7):1558-1568. (in Chinese)
- [26] 罗仕杭,何庆. 多策略协同改进的阿基米德优化算法及其应用[J]. *计算机应用研究*, 2022, 39(5):1386-1394.  
LUO S H, HE Q. Improved Archimedes optimization algorithm by multi-strategy collaborative and its application [J]. *Application Research of Computers*, 2022, 39(5):1386-1394. (in Chinese)