DOI:10.19651/j.cnki.emt.2314061

基于精英差分变异的改进蜜獾算法*

周建新 张力洪 孙腾浩
(华北理工大学电气工程学院 唐山 063210)

摘 要:针对标准蜜獾算法(HBA)易陷入局部最优、搜索精度低、收敛速度较慢等问题,提出基于精英差分变异的蜜獾算法(EDVHBA)。将标准 HBA 中的两种寻优策略所搜寻到的精英解,进行组合差分变异以产生新的精英解,利用 3 个精英解协同指导种群下一轮迭代,可以增加算法解的多样性,防止算法陷入过早收敛;同时改进非线性密度因子和引入新的位置更新策略,提升算法的收敛速度和寻优精度。为验证算法的改进效果和性能,对 8 个经典测试函数进行仿真实验,实验结果表明:与其他群智能算法和改进的 HBA 相比,EDVHBA 在单峰函数中都能搜寻到最优值 0,在多峰函数中迭代 50 次左右就可以收敛到理想最优值,验证了 EDVHBA 具有更好的寻优性能。

关键词:蜜獾算法;非线性密度因子;精英差分变异;逃离捕食者策略
中图分类号:TP301.6 **文献标识码**:A **国家标准学科分类代码**:520.1040

Improved honey badger algorithm based on elite differential variation

Zhou Jianxin Zhang Lihong Sun Tenghao
(College of Electrical Engineering, North China University of Technology, Tangshan 063210, China)

Abstract: Aiming at the problems that the standard honey badger algorithm (HBA) is easy to fall into local optimum, low search accuracy and slow convergence speed, a honey badger algorithm based on elite differential mutation (EDVHBA) is proposed. The elite solution searched by the two optimization strategies in the standard HBA is combined with differential mutation to generate a new elite solution. The use of three elite solutions to guide the next iteration of the population can increase the diversity of the algorithm solution and prevent the algorithm from falling into premature convergence. At the same time, the nonlinear density factor is improved and a new position update strategy is introduced to improve the convergence speed and optimization accuracy of the algorithm. In order to verify the performance of the algorithm, simulation experiments are carried out on eight classical test functions. The results show that compared with other swarm intelligence algorithms and improved HBA, EDVHBA can find the optimal value 0 in the unimodal function, and converge to the ideal optimal value in the multimodal function after about 50 iterations, which verifies that EDVHBA has better optimization performance.

Keywords: honey badger algorithm; nonlinear density factor; elite differential variation; escape predator strategy

0 引 言

现实生活中的大多数优化问题具有高维、非线性、非凸以及约束条件多等特点,简单的使用计算机进行数值计算和数值逼近,难以解决这类问题。群智能算法(Swarm Intelligence)是一类基于群体行为和自组织理论的算法,这些算法通过模拟自然界中群体行为,将其转化为数学模型以解决各种实际问题,如麻雀搜索算法(sparrow search algorithm, SSA)[1]、鲸鱼优化算法(whale optimization algorithm, WOA)[2]、鹈鹕优化算法(pelican optimization algorithm, POA)[3]、灰狼优化算法(grey wolf optimization, GWO)[4]、哈里斯鹰算法(harris hawks optimizer, HHO)[5]等。这类算法在解决复杂高维问题时具有全局搜索能力强、自适应调整、简单高效易实现等特点,使其成为解决实际问题的有力工具。蜜獾算法(honey badger algorithm, HBA)[6]由 Fatma 等于 2021 年提出,该算法模拟了蜜獾捕食、采蜜等行为,具有可调参数少、简单易实现、寻优能力强等优点,但仍具有容易出现早熟收敛、寻优精度差、收敛速度慢等问题。

针对 HBA 算法存在的缺陷,已有学者对其进行了改

收稿日期:2023-07-09
* 基金项目:河北省自然科学基金(F2018209201)项目资助

进和应用。Abasi 等[7]在 HBA 算法后期加入拟反向学习策略，增加种群多样性，并提出自适应变异策略，提升种群中最优解的引导作用。Zhou 等[8]在种群初始化阶段加入立方混沌映射，使种群初始分布更加均匀，在算法后期引入高斯方差函数，防止算法陷入局部最优。宋跃才等[9]将基于佳点集的蜜獾算法与 DV_Hop(distance vector-Hop)算法相结合，解决了传统 DV_Hop 算法定位精度不足的问题。虽然这些文献对 HBA 进行了改进提升并验证了改进后算法的优越性能，但仍具有某些不足之处。比如只是单纯的加入某些初始化策略，却未对权重参数或核心公式进行融合改进；改进方法过于复杂，增加了计算成本；算法测试不全面，未对改进后的算法与他人所改进的算法进行对比实验等。

本文针对上述文献存在的问题，提出基于精英差分变异的蜜獾算法(elite differential variation honey badger algorithm，EDVHBA)，首先改进非线性密度因子，加快算法的收敛速度；其次，提出精英差分变异策略，增加种群中解的多样性，避免算法陷入过早收敛；最后，将逃离捕食者策略引入到蜜獾算法并进行改进，提升算法的迭代速度和寻优精度。将改进的 EDVHBA 算法与近几年提出的群智能优化算法和其他改进的 HBA 算法在 8 个测试函数中进行对比实验，验证本文改进算法性能的优越性并分析不同改进策略的有效性。

1　蜜獾算法(HBA)

蜜獾获取食物的方式主要有两种，即“嗅探”和“跟随”。“嗅探”指蜜獾通过嗅觉来不断挖掘洞穴进行勘探并捕获猎物，该行为也称挖掘模式；“跟随”指蜜獾通过跟随导蜜鸟来定位蜂房位置并盗取蜂蜜，该行为也称采蜜模式；蜜獾算法主要模拟了蜜獾的这两种独特的捕食方式，建立相关数学模型进行迭代寻优。

1)种群初始化：种群初始化在每个特征的取值范围内均匀地生成初始值，保证种群的多样性。具体表达式为：

$$x_i = lb_i + r_1 \times (ub_i + lb_i) \tag{1}$$

式中：r_1 为[0,1]的随机数；x_i 为第 i 只蜜獾的位置；ub_i 和 lb_i 分别为解空间的上下界。

2)定义强度：猎物的气味强度 I_i 影响蜜獾的移动速度，蜜獾距离猎物越近，则气味强度 I_i 越大，蜜獾的移动速度越快。计算公式：

$$I_i = r_2 \times \frac{S}{4\pi d_i^2} \tag{2}$$

$$S = (x_i - x_{i+1})^2 \tag{3}$$

$$d_i = x_{prey} - x_i \tag{4}$$

式中：r_2 为[0,1]的随机数；S 为猎物集中强度；d_i 为猎物与第 i 只蜜獾间的距离。

3)更新密度因子：密度因子 α 主要用来平衡算法的全局搜索和局部开发能力，其中 α 会随着迭代次数的增加而减少。计算公式：

$$\alpha = C \times \exp\left(\frac{-t}{t_{\max}}\right) \tag{5}$$

式中：$t_{\max}$ 为最大迭代次数；$C \geqslant 1$。

4)挖掘模式：在挖掘模式下，蜜獾主要在心形区域内进行位置更新并寻找猎物，具体表达式为：

$$x_{new} = x_{prey} + F \times \beta \times I \times x_{prey} + F \times r_3 \times \alpha \times d_i \times |\cos(2\pi r_4) \times [1 - \cos(2\pi r_5)]| \tag{6}$$

式中：x_{prey} 为当前全局最优食物源位置；r_3、r_4、r_5 为[0,1]的随机数；$\beta \geqslant 1$；F 表示蜜獾改变搜索方向，计算公式：

$$F = \begin{cases} 1, & r_6 < 0.5 \\ -1, & \text{其他} \end{cases} \tag{7}$$

式中：r_6 为[0,1]的随机数；在此模式中，蜜獾的位置更新依赖于 I_i、d_i、α 等参数的取值。

5)采蜜模式：在采蜜模式中，蜜獾主要通过跟随导蜜鸟进行位置更新，具体表达式为：

$$x_{new} = x_{prey} + F \times r_7 \times \alpha \times d_i \tag{8}$$

式中：x_{new} 为蜜獾个体更新后的位置；r_7 为[0,1]的随机数；在采蜜模式中，蜜獾主要围绕最优食物源位置 x_{prey} 附近进行位置更新。

2　改进的蜜獾算法(EDVHBA)

2.1　改进密度因子 α

由 HBA 算法可知，随着密度因子 α 的非线性递减，算法由全局勘探逐渐转为局部开发，但 α 的递减曲线下降较慢且不能收敛至 0，导致算法的搜索性能降低。因此，本文提出一种新的非线性密度因子控制公式，可以提升种群的寻优精度，同时加快收敛速度。计算公式：

$$\alpha' = C - C \times \sin\left(\frac{t}{t_{\max}} \times \frac{\pi}{2}\right) + \sigma \times randn() \tag{9}$$

式中：$t_{\max}$ 为最大迭代次数；$C \geqslant 1$(一般默认为 2)，$randn()$ 为服从正态分布的随机数，σ 为方差；如图 1 所示，与原 HBA 算法相比，在迭代前中期，改进密度因子 α' 下降的更加迅速，可以增加算法前中期的收敛速度；在迭代后期，α' 的变化幅度和速度变得更小，加强了算法的局部搜索能力；同时引入 $randn()$ 正态分布的随机数，对 α' 进行局部扰动，增加解的多样性，避免算法过早收敛。

2.2　精英差分变异策略

在标准的 HBA 算法中，蜜獾个体的位置更新受到最优食物源 x_{prey} 的引导。根据挖掘模式和采蜜模式的公式，蜜獾个体主要围绕 x_{prey} 附近进行位置更新。然而，如果 x_{prey} 只是一个局部最优解，种群中的个体会随着算法的迭代而聚集在这个局部最优解附近，导致算法陷入局部最优。

本文提出精英差分变异策略，对于传统差分进化的变异过程，参与变异的 3 个个体都是随机选择的，最终产生变异个体的适应度差异较大，使得种群演化具有较大的随机性，影响算法整体的优化能力[10-11]。本文将蜜獾种群均分

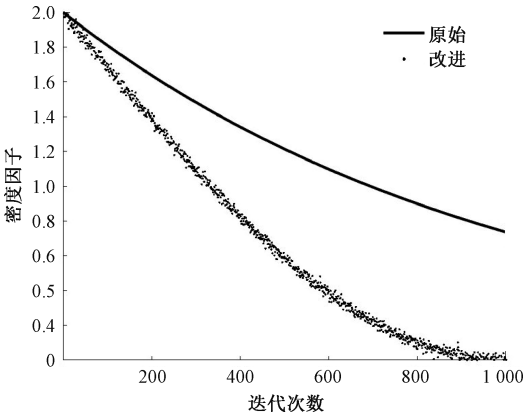

图1 改进前后 α 的对比图

为两组,一组执行挖掘模式进行寻优,得到的精英解设为 x_{new1},另一组执行采蜜模式进行寻优,得到的精英解设为 x_{new2},再将两组种群各寻到的精英解,进行精英变异产生新的精英解 x_{new3}。这种变异策略可以更有针对性地引导种群在适应度更优的食物源的区域产生子代,同时也可避免传统差分进化变异过程的盲目性,提升算法整体的搜索效率。精英变异具体表达式为:

$$x_{new3}(t)=\varphi\times\frac{(x_{new2}-x_{new1})}{2}+\varphi\times(x_{new1}-x(t))+\varphi\times(x_{new2}-x(t)) \tag{10}$$

式中:φ 为变异因子;x_{new1} 为挖掘模式寻到的精英解;x_{new2} 为采蜜模式寻到的精英解,x_{new3} 为经过精英差分变异后产生的新精英解。

最后,采用贪婪选择策略,将挖掘模式、采蜜模式寻到的精英解所对应适应度值,与经过精英变异后产生的新精英解对应适应度相对比,选出3个精英解中适应度最小的解,指导蜜獾下一代种群向全局最优位置移动,解决算法陷入局部最优的问题。具体表达式为:

$$x_{new}(t+1)=\begin{cases}x_{new1}(t);f(x_{new1})<f(x_{new2}),f(x_{new3})\\x_{new2}(t);f(x_{new2})<f(x_{new1}),f(x_{new3})\\x_{new3}(t);f(x_{new3})<f(x_{new1}),f(x_{new2})\end{cases} \tag{11}$$

2.3 逃离捕食者策略

针对蜜獾算法对于多极值问题寻优精度低、不易跳出局部最优等问题,受长鼻浣熊优化算法(coati optimization algorithm,COA)[11-12]启发,融合改进的密度因子 α' 和改变搜索方向 F,提出蜜獾逃离捕食者策略。当蜜獾觅食时受到捕食者攻击,即侦察值大于安全值时[13],蜜獾首先会选择逃离当前位置,在所处位置附近随机生成一个位置,并逐渐逼近安全位置。改进的密度因子 α' 用来平衡蜜獾逃跑的步长,步长会随着迭代进行而逐渐减小,使得算法后期能进行更精确的搜索。这种策略可以有效的提升 HBA 算法对多极值复杂问题的收敛速度和寻优精度。具体表达式为:

$$x'_{new}(t+1)=x(t)+F\times\alpha'\times D\times(lb_{local}+r(ub_{local}-lb_{local})) \tag{12}$$

式中:x'_{new} 为蜜獾逃脱后的新位置;α' 为改进的密度因子;D 为[-1,1]均匀分布的随机数;ub_{local}、lb_{local} 为随迭代次数而更新的上、下界。

D 和 ub_{local}、lb_{local} 的计算公式如下:

$$D=2r-1 \tag{13}$$

$$lb_{local}=\frac{lb}{t} \tag{14}$$

$$ub_{local}=\frac{ub}{t} \tag{15}$$

式中:r 为[0,1]的随机数;ub 和 lb 为解空间的上下界;t 为当前迭代次数。

判断是否执行逃离捕食者策略,当侦察值小于安全值时,执行挖掘和采蜜模式,反之则执行逃离捕食者策略。具体表达式为:

$$x_{new}=\begin{cases}x_{new1}\ or\ x_{new2}, & R<SV\\x'_{new}, & R\geqslant SV\end{cases} \tag{16}$$

式中:R 为[0,1]的随机数表示侦察值,SV 表示安全值。

2.4 算法流程

EDVHBA 算法流程步骤如下:

步骤1)初始化蜜獾的种群规模 N、迭代次数 $t_{\max}$、获取食物的能力 β 和 C 等参数。

步骤2)根据种群搜索空间的边界,随机产生 N 个蜜獾的位置。

步骤3)评估每只蜜獾的适应度值并排序,保存最优适应度的位置 x_{prey}。

步骤4)根据式(2)~(4)和式(9)计算气味强度 I_i 和改进的密度因子 α'。

步骤5)根据式(6)~(8)分别更新蜜獾在挖掘模式的位置 x_{new1} 和采蜜模式的位置 x_{new2}。

步骤6)根据式(10)~(11)对蜜獾种群的两个最优位置进行精英差分变异产生新位置 x_{new3},将新位置的适应度与原最优位置 x_{new1}、x_{new2} 的适应度进行比较,选出适应度最小的位置 x_{new}。

步骤7)先根据式(16)判断是否执行逃离策略,若符合条件,则根据式(12)~(15)更新蜜獾逃离捕食者后的新位置 x'_{new},再次比较 x_{new} 和 x'_{new} 对应的适应度,选出最优位置。

步骤8)判断是否达到最大迭代次数,若满足条件则输出最优解,否则跳转到步骤4)。

3 仿真实验与分析

3.1 实验环境

本文仿真实验基于 Windows 11(64 bit),处理器 AMD R75800 H with Radeon Graphics. 3.20 GHz,电脑内存 16 G,仿真平台 MatlabR2022a。

3.2　对比算法及测试条件

本文选取近几年新提出的群优化算法进行对比实验，选择的算法有麻雀搜索算法（SSA）[1]、鲸鱼优化算法（WOA）[2]、哈里斯鹰算法（HHO）[5]、标准蜜獾算法（HBA）[6]。为保证实验的公平性，每个算法独立运行 50 次，种群规模 N=30，最大迭代次数 $t_{max}=500$，在 dim=30/200 维下进行对比。

3.3　标准测试函数

为了验证 EDVHBA 算法的性能和改进策略的有效性，选用 8 个标准测试函数进行仿真实验，其中 F1～F4 为单峰基准测试函数，用于测试算法的全局搜索性能，F5～F8 为多峰基准测试函数，用于测试算法的局部搜索和收敛性能。标准测试函数如表 1 所示。

表 1　标准测试函数

类型	测试函数	范围	最优值
单峰函数	$F1=\sum_{i=1}^{n}x_i^2$	$[-100,100]$	0
	$F2=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	$[-10,10]$	0
	$F3=\sum_{i=1}^{n}\left(\sum_{j=1}^{i}x_j\right)^2$	$[-100,100]$	0
	$F4=\max_i\{\lvert x_i\rvert,1\leqslant i\leqslant n\}$	$[-100,100]$	0
多峰函数	$F5=\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)+10]$	$[-5.12,5.12]$	0
	$F6=-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)+20+\mathrm{e}$	$[-32,32]$	0
	$F7=\frac{\pi}{n}\left\{10\sin(\pi y_1)+\sum_{i=1}^{n-1}(y_i-1)^2[1+10\sin^2(\pi y_{i+1})]+(y_i-1)^2\right\}+\sum_{i=1}^{n}u(x_i,10,100,4)$，$y_i=1+\frac{x_i+1}{4}u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m, & x_i>a\\ 0, & -a<x_i<a\\ k(-x_i-a)^m, & x_i<-a\end{cases}$	$[-50,50]$	0
	$F8=0.1\{\sin^2(3\pi x_1)+\sum_{i=1}^{n}(x_i-1)^2[1+\sin^2(3\pi x_i+1)]+(x_n-1)^2[1+\sin^2(2\pi x_n)]\}+\sum_{i=1}^{n}u(x_i,5,100,4)$	$[-50,50]$	0

3.4　实验结果对比与分析

为对比各算法求解标准测试函数精度，对各算法的平均值（average，Ave）和标准差（standard deviation，Std）进行对比，平均值能反映算法求解问题的平均精度，标准差能反映算法的稳定性，其测试结果如表 2、表 3 所示，其最优结果已加黑标出。

对于单峰测试函数 F1～F4 中，EDVHBA 算法的平均值和标准差均为最小值且都达到了理论最优值 0，仿真实验的表现最好，而 HHO、HBA 的表现次之，而 WOA 的表现最差，说明改进的 EDVHBA 算法与原 HBA 和对比算法相比，在单峰问题上的收敛精度和稳定性更高，整体性能优于其他算法。对于多峰测试函数，本文改进算法与原 HBA 和对比算法在 F5～F6 表现相同，都能寻到最优值或近似最优值。但在 F7～F8 中，本文改进算法的寻优精度优于其他对比算法，与对比算法中表现最好的 SSA 算法和原 HBA 算法相比，EDVHBA 算法在平均值方面分别提升了约 24 个数量级和 30 个数量级，标准差也提升了约 40 个数量级，表明改进的 EDVHBA 算法在多极值问题上全局寻优能力更强，算法的性能也更加稳定。

另外，随着测试函数维度从 30 维提升到 200 维，算法搜索到最优值的困难程度也会逐渐增加，但本文改进的 EDVHBA 算法无论是在单峰测试函数 F1～F4 中，还是在多峰测试函数 F5～F8 中，其平均值和标准差的变化均较小，并且接近算法在低维条件下测试的最优值。表明本文改进的算法对于高维度的问题具有良好的适应性，无论问题的维度增加还是减少，算法都能够保持优秀的性能并找

表 2 单峰、多峰测试函数 30 维度测试结果对比

算法	统计值	F1	F2	F3	F4	F5	F6	F7	F8
WOA	Ave	1.77×10^{-73}	4.49×10^{-51}	$4.03\times10^{+04}$	$4.86\times10^{+01}$	$0.00\times10^{+00}$	3.52×10^{-15}	2.44×10^{-02}	5.29×10^{-01}
	Std	8.66×10^{-73}	1.49×10^{-50}	$1.16\times10^{+04}$	$2.79\times10^{+01}$	$0.00\times10^{+00}$	2.91×10^{-15}	1.29×10^{-02}	2.63×10^{-01}
SSA	Ave	1.60×10^{-54}	1.94×10^{-29}	1.00×10^{-22}	1.44×10^{-32}	$0.00\times10^{+00}$	4.44×10^{-16}	1.03×10^{-08}	9.21×10^{-08}
	Std	8.75×10^{-54}	1.02×10^{-27}	5.48×10^{-22}	7.66×10^{-32}	$0.00\times10^{+00}$	$0.00\times10^{+00}$	3.04×10^{-08}	3.11×10^{-07}
HHO	Ave	4.17×10^{-93}	9.03×10^{-50}	2.60×10^{-73}	1.80×10^{-50}	$0.00\times10^{+00}$	4.44×10^{-16}	8.52×10^{-06}	9.58×10^{-05}
	Std	2.28×10^{-92}	3.54×10^{-49}	1.42×10^{-72}	8.22×10^{-50}	$0.00\times10^{+00}$	$0.00\times10^{+00}$	1.24×10^{-05}	1.19×10^{-04}
HBA	Ave	2.33×10^{-136}	1.02×10^{-72}	1.97×10^{-97}	3.79×10^{-133}	$0.00\times10^{+00}$	4.44×10^{-16}	1.76×10^{-03}	3.75×10^{-01}
	Std	9.23×10^{-136}	1.81×10^{-72}	8.71×10^{-97}	2.14×10^{-132}	$0.00\times10^{+00}$	$0.00\times10^{+00}$	3.42×10^{-03}	3.36×10^{-01}
EDVHBA	Ave	$\mathbf{0.00\times10^{+00}}$	$\mathbf{0.00\times10^{+00}}$	$\mathbf{0.00\times10^{+00}}$	$\mathbf{0.00\times10^{+00}}$	$\mathbf{0.00\times10^{+00}}$	$\mathbf{4.44\times10^{-16}}$	$\mathbf{1.57\times10^{-32}}$	$\mathbf{1.35\times10^{-32}}$
	Std	$\mathbf{0.00\times10^{+00}}$	$\mathbf{0.00\times10^{+00}}$	$\mathbf{0.00\times10^{+00}}$	$\mathbf{0.00\times10^{+00}}$	$\mathbf{0.00\times10^{+00}}$	$\mathbf{0.00\times10^{+00}}$	$\mathbf{5.57\times10^{-48}}$	$\mathbf{5.53\times10^{-48}}$

表 3 单峰、多峰测试函数 200 维度测试结果对比

算法	统计值	F1	F2	F3	F4	F5	F6	F7	F8
WOA	Ave	3.39×10^{-71}	2.29×10^{-49}	$1.08\times10^{+07}$	$7.52\times10^{+01}$	$0.00\times10^{+00}$	3.64×10^{-15}	6.88×10^{-02}	$6.67\times10^{+00}$
	Std	1.26×10^{-70}	5.60×10^{-49}	$3.13\times10^{+06}$	$2.52\times10^{+01}$	$0.00\times10^{+00}$	2.70×10^{-15}	2.84×10^{-02}	$2.05\times10^{+00}$
SSA	Ave	3.84×10^{-53}	1.16×10^{-28}	8.83×10^{-20}	2.62×10^{-31}	$0.00\times10^{+00}$	4.44×10^{-16}	1.68×10^{-07}	1.29×10^{-05}
	Std	1.99×10^{-52}	4.71×10^{-31}	4.84×10^{-19}	8.32×10^{-31}	$0.00\times10^{+00}$	$0.00\times10^{+00}$	3.33×10^{-07}	2.29×10^{-05}
HHO	Ave	3.61×10^{-92}	1.60×10^{-47}	4.96×10^{-41}	1.34×10^{-49}	$0.00\times10^{+00}$	4.44×10^{-16}	2.86×10^{-06}	1.47×10^{-04}
	Std	1.10×10^{-92}	8.72×10^{-47}	2.72×10^{-40}	5.76×10^{-49}	$0.00\times10^{+00}$	$0.00\times10^{+00}$	4.28×10^{-06}	1.62×10^{-04}
HBA	Ave	7.65×10^{-114}	1.02×10^{-72}	1.77×10^{-65}	1.89×10^{-30}	$0.00\times10^{+00}$	4.44×10^{-16}	4.12×10^{-01}	$1.90\times10^{+01}$
	Std	4.09×10^{-113}	1.81×10^{-72}	9.54×10^{-65}	4.05×10^{-30}	$0.00\times10^{+00}$	$0.00\times10^{+00}$	6.00×10^{-02}	3.98×10^{-01}
EDVHBA	Ave	$\mathbf{0.00\times10^{+00}}$	$\mathbf{0.00\times10^{+00}}$	$\mathbf{0.00\times10^{+00}}$	$\mathbf{0.00\times10^{+00}}$	$\mathbf{0.00\times10^{+00}}$	$\mathbf{4.44\times10^{-16}}$	$\mathbf{2.36\times10^{-32}}$	$\mathbf{1.35\times10^{-32}}$
	Std	$\mathbf{0.00\times10^{+00}}$	$\mathbf{0.00\times10^{+00}}$	$\mathbf{0.00\times10^{+00}}$	$\mathbf{0.00\times10^{+00}}$	$\mathbf{0.00\times10^{+00}}$	$\mathbf{0.00\times10^{+00}}$	$\mathbf{6.96\times10^{-48}}$	$\mathbf{5.57\times10^{-48}}$

到最优解。总体而言,本文所提 EDVHBA 算法,相比原 HBA 算法有较大的提升,在测试函数寻优中具有明显优势。

3.5 收敛性对比与分析

为进一步验证本文改进的 EDVHBA 算法的收敛性能,将 EDVHBA 算法与上述其他群优化算法在 6 个测试函数上的收敛性进行对比,如图 2 所示。EDVHBA 算法在 6 个测试函数中,无论是单峰测试函数还是多峰测试函数,其收敛速度与寻优精度均优于其他群智能算法。在 F1～F3 中,EDVHBA 算法收敛最快,在 200～300 次迭代左右,就可以收敛到理论最优值,表明本文改进的算法局部开发能力更强。在 F5～F7 中,EDVHBA 算法与原 HBA 和其他算法相比,收敛速度更快,跳出局部最优能力更强,并且平均迭代 50 次左右就可以找到高质量的最优值,验证了本文加入相应改进策略的有效性,相较于原 HBA 收敛精度与收敛速度均有显著提升。

3.6 不同改进蜜獾算法对比

为验证本文改进 EDVHBA 算法的优越性,将 EDVHBA 与最新的改进蜜獾算法进行对比,引入莱维飞行和最优个体自适应变异的蜜獾算法(improved honey badger algorithm,IHBA)[14]、基于伯努利移位映射和分段最优递减邻域策略的蜜獾算法(modified honey badger algorithm with multi-strategy,SaCHBA)[15]、引入差分进化变异和交叉策略的蜜獾算法(differential honey badger algorithm,DHBA)[16]。算法种群数量都为 30,迭代次数都为 500,将函数平均值(Ave)和标准差(Std)作为评价指标,对比结果如表 4 和图 3 所示。

在单峰测试函数中,EDVHBA 与 IHBA 在 F1 和 F3 中表现相同,都能寻到最优值,但在 F2 和 F4 中,只有 EDVHBA 寻得最优值且收敛最快,表明其性能优于 SaCHBA 和 DHBA 算法。在多峰测试函数中,在 F5 和 F6 中 4 种算法表现相同,但在 F7 和 F8 中,只有 EDVHBA 最接近理论最优值且标准差最小,收敛曲线下降的最快,表明本文改进的算法在处理复杂问题上稳定性更好,精度更高。总体而言,EDVHBA 算法在 8 个基准测试函数上的整体性能相比于其他改进的蜜獾算法具有较大优势。

3.7 改进策略有效性分析

为了验证 EDVHBA 算法改进策略的有效性,将采用密度因子改进的 HBA-1、精英差分变异改进的 HBA-2、逃离捕食者策略的 HBA-3 和 HBA 算法与 EDVHBA 算法进行对比。每个测试函数独立运行 30 次,将函数平均值(Ave)作为评价标准,实验结果如表 5 所示。

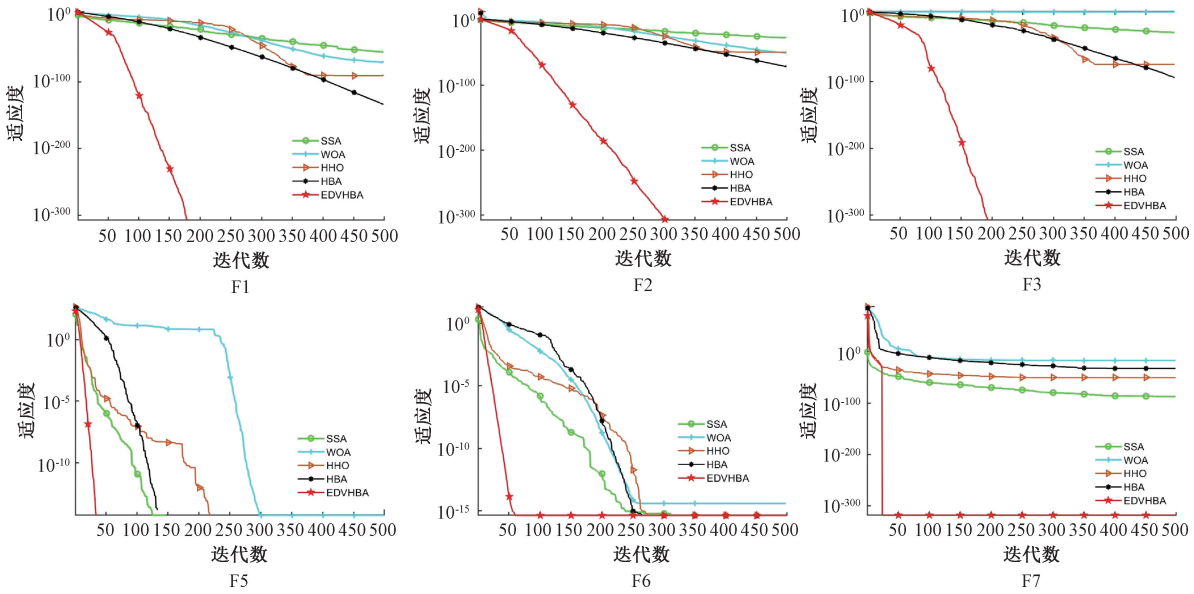

图 2　不同算法收敛曲线对比

表 4　改进的 HBA 算法对比

算法	统计值	F1	F2	F3	F4	F5	F6	F7	F8
IHBA	Ave	**0.00×10^{+00}**	6.56×10^{-312}	**0.00×10^{+00}**	6.65×10^{-256}	0.00×10^{+00}	4.44×10^{-16}	3.33×10^{-08}	1.77×10^{-08}
	Std	**0.00×10^{+00}**	0.00×10^{+00}	**0.00×10^{+00}**	0.00×10^{+00}	0.00×10^{+00}	0.00×10^{+00}	5.26×10^{-08}	2.13×10^{-08}
SaCHBA	Ave	4.10×10^{-212}	3.83×10^{-193}	7.56×10^{-293}	2.02×10^{-167}	0.00×10^{+00}	4.44×10^{-16}	1.81×10^{-02}	3.39×10^{-01}
	Std	0.00×10^{+00}	0.00×10^{+00}	0.00×10^{+00}	0.00×10^{+00}	0.00×10^{+00}	0.00×10^{+00}	5.07×10^{-02}	2.66×10^{-01}
DHBA	Ave	**0.00×10^{+00}**	1.59×10^{-246}	1.28×10^{-292}	3.79×10^{-133}	0.00×10^{+00}	4.44×10^{-16}	2.94×10^{-30}	1.38×10^{-30}
	Std	**0.00×10^{+00}**	6.38×10^{-240}	6.03×10^{-290}	2.14×10^{-132}	0.00×10^{+00}	0.00×10^{+00}	4.73×10^{-30}	9.89×10^{-30}
EDVHBA	Ave	**0.00×10^{+00}**	**0.00×10^{+00}**	**0.00×10^{+00}**	**0.00×10^{+00}**	0.00×10^{+00}	4.44×10^{-16}	**1.57×10^{-32}**	**1.35×10^{-32}**
	Std	**0.00×10^{+00}**	**0.00×10^{+00}**	**0.00×10^{+00}**	**0.00×10^{+00}**	0.00×10^{+00}	0.00×10^{+00}	**5.57×10^{-48}**	**5.53×10^{-48}**

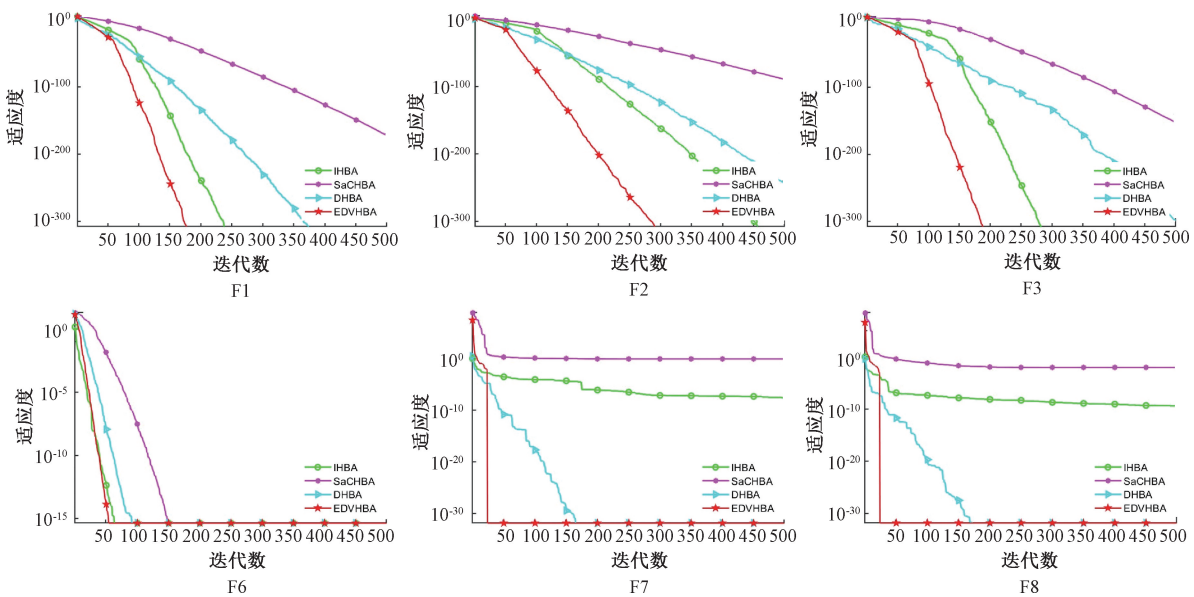

图 3　改进的 HBA 收敛曲线对比

表5 不同改进策略对比

函数	F1	F2	F3	F4	F5	F6	F7	F8
HBA-1	1.14×10^{-176}	2.34×10^{-90}	5.66×10^{-157}	2.48×10^{-82}	$0.00\times10^{+00}$	4.44×10^{-16}	1.84×10^{-02}	9.87×10^{-01}
HBA-2	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	$0.00\times10^{+00}$	4.44×10^{-16}	9.91×10^{-05}	9.59×10^{-01}
HBA-3	6.57×10^{-141}	4.27×10^{-75}	5.37×10^{-108}	1.20×10^{-52}	$0.00\times10^{+00}$	4.44×10^{-16}	**1.57×10^{-32}**	**1.35×10^{-32}**
HBA	3.92×10^{-138}	3.68×10^{-75}	2.79×10^{-103}	2.30×10^{-56}	$0.00\times10^{+00}$	4.44×10^{-16}	2.36×10^{-06}	5.92×10^{-04}
EDVHBA	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	**$0.00\times10^{+00}$**	$0.00\times10^{+00}$	4.44×10^{-16}	**1.57×10^{-32}**	**1.35×10^{-32}**

在测试函数F1～F4中,HBA-2都能寻到理论最优值,在测试函数F7、F8中,HBA-3搜寻到的解最为接近最优值,可见HBA-2、HBA-3对改进算法起到重要的作用。在全部测试函数中,单一改进策略和策略融合后的EDVHBA相比原HBA算法,寻优精度均有不同程度的提升,验证了EDVHBA算法不同改进策略的有效性。

4 结 论

针对HBA算法易陷入局部最优、寻优精度低、收敛速度慢等问题,本文提出精英差分变异策略,将原HBA算法中的两种寻优策略所搜寻到的精英解,进行组合差分变异以产生新的精英解,利用3个精英解协同指导种群逐渐向全局最优区域移动,可以有效防止算法陷入局部最优。对非线性密度因子进行改进,协调算法的探索与开发,提升算法收敛速度。引入浣熊逃离捕食者策略并进行改进,以提升算法对多极值复杂问题的寻优精度。将EDVHBA算法与近几年新提出的群优化算法和改进的HBA算法在8个测试函数中进行对比测试,并分析不同改进策略的有效性,测试结果表明EDVHBA算法的性能最优,具有更快的收敛速度和较准确的寻优精度。后续工作尝试将EDVHBA算法应用到实际工业优化问题中,验证其实际应用性能。

参考文献

[1] XUE J, SHEN B. A novel swarm intelligence optimization approach: sparrow search algorithm[J]. Systems Science & Control Engineering, 2020, 8(1): 22-34.

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

[3] TROJOVSKY P, DEHGHANI M. Pelican optimization algorithm: A novel nature-inspired algorithm for engineering applications[J]. Sensors, 2022, 22(3): 855.

[4] MIRJALILI S, LEWIS A. Grey wolf optimizer[J]. Advances in Engineering Software, 2014, 69(3): 46-61.

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

[6] 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.

[7] ABASI A K, ALOQAILY M. Optimization of CNN using modified honey badger algorithm for sleep apnea detection[J]. Expert Systems with Applications, 2023, 229: 120484.

[8] ZHOU C, GAO B, YANG H, et al. Junction temperature prediction of insulated-gate bipolar transistors in wind power systems based on an improved honey badger algorithm[J]. Energies, 2022, 15(19): 7366.

[9] 宋跃才,林海涛,卞媛,等.基于测距修正和蜜獾优化的改进DV_HOP定位算法[J].电子测量技术,2022,45(9):147-153.

[10] 陈若涵,孙晔,孙洁.基于混沌差分进化算法的车联网计算卸载研究[J].国外电子测量技术,2022,41(12):89-95.

[11] 廉小亲,陈彦铭,刘钰,等.基于差分进化算法的ICP-AES谱线重叠干扰校正方法研究[J].电子测量与仪器学报,2020,34(11):72-83.

[12] DEHGHANI M, MONTAZERI Z, TROJOVSKÁ E, et al. Coati optimization algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems[J]. Knowledge-Based Systems, 2023, 259: 110011.

[13] 王雨虹,王志中,付华,等.多策略改进麻雀算法与BiLSTM的变压器故障诊断研究[J].仪器仪表学报,2022,43(3):87-97.

[14] 董红伟,李爱莲,解韶峰,等.多策略改进的蜜獾优化算法[J].小型微型计算机系统,2024,45(2):293-300.

[15] HU G, ZHONG J, WEI G. SaCHBA_PDN: Modified honey badger algorithm with multi-strategy for UAV path planning[J]. Expert Systems with Applications, 2023, 223: 119941.

[16] 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.

作者简介

周建新,副教授,硕士生导师,主要研究方向为智能控制理论及应用。

E-mail:zhoujianxin1977@126.com

张力洪,硕士研究生,主要研究方向为电力设备状态检测与故障诊断。

E-mail:zhanglihong0502@163.com

孙腾浩,硕士研究生,主要研究方向为复杂系统建模分析与控制。

E-mail:1132170739@qq.com