文章编号:1671-4598(2023)09-0260-07 DOI:10.16526/j.cnki.11—4762/tp.2023.09.038 中图分类号:TP391.4 文献标识码:A

改进蜜獾算法优化 OTSU 的图像分割研究

崔文静，李 帅，彭天文，梁宏涛
(青岛科技大学 信息科学技术学院，山东 青岛 266061)

摘要：群体智能算法结合图像分割技术已经成为图像处理领域中的新热点，传统的图像分割方法需要大量的人力和时间，蜜獾算法（honey badger algorithm，HBA）可以通过模拟蜜獾觅食的行为来执行优化任务，在寻找解决问题的过程中可以逐步逼近最优解来实现图像分割任务；通过反向学习策略改进蜜獾种群的初始化，提高种群多样性和分布平衡，从而提高算法的整体搜索能力；引入柯西变异因子，对算法计算得到的可行解进行扰动，使算法更易于跳出局部最优，增强算法的局部搜索能力和收敛精度；选取三幅测试图像进行分割验证，实验结果显示，融合改进蜜獾算法和二维 OTSU 算法得到的分割图像精度更高、效果更细致，验证了方法的有效性；综上所述，改进蜜獾算法具有更好的鲁棒性和泛化性，优化的二维 OTSU 算法可以更好地处理复杂场景和图像。

关键词：二维 OTSU 算法；蜜獾算法；反向学习策略；柯西变异；图像分割

Image Segmentation on Improved HBA to Optimize OTSU

CUI Wenjing, LI Shuai, PENG Tianwen, LIANG Hongtao
(School of Information Science and Technology, Qingdao University of Science and Technology, Qingdao 266061, China)

Abstract: Population intelligence algorithm combined with image segmentation technology has become a new hot spot in the field of image processing. Traditional image segmentation methods require a lot of manpower and time, honey badger algorithm (HBA) can perform the optimization task by simulating the behavior of honey badger foraging, and can gradually approach the optimal solution to achieve the image segmentation task in the process of finding a solution to the problem; The initialization of the honey badger population is improved by the backward learning strategy, which improves the population diversity and distribution balance, thus enhancing the overall search ability of the algorithm; Cauchy variation factor is introduced to perturb the feasible solutions calculated by the algorithm, which makes the algorithm easier to jump out of the local optimum and enhances the local search ability and convergence accuracy of the algorithm; three test images are selected for segmentation verification, and the experimental results show that the segmented images obtained by fusing the improved HBA and 2D OTSU algorithm are more accurate and detailed, which verifies the effectiveness of the method. In summary, the improved HBA has better robustness and generalization, and the optimized 2D OTSU algorithm can better handle complex scenes and images.

Keywords: two-dimensional OTSU algorithm; HBA; cauchy mutation; image segmentation

0 引言

图像分割是一个至关重要的预处理方法，在图像识别等应用领域尤为重要。它的主要任务是把所需的前景和不相关的背景进行分割，而图像分割的优劣关系到特征提取和图像识别等其他环节，所以一个良好的图像分割方法是十分关键的[1]。

OTSU 阈值分割法是图像分割中最常用的处理方式，它通过选择前景与背景类间方差最大时的值作为分割阈值[2]，不但方便简单，同时也可以更有效地对多种类型图像进行分割。由于其分割处理的流程更加简便，并且拥有卓越的分割效果，所以拥有十分普遍的应用[3]。OTSU 阈值分割法还有以下几个优点：对于像素值分布单峰的图像，分割效果较好；可以快速处理大型图像，使图像分割过程实现实时处理和低计算成本；与其他图像分割方法相比，阈值分割法的算法简单，易于理解和实现。然而，OTSU 法在处理多峰图像时可能会比较困难，因为它难以找到合适的阈值来正确地将像素分割成两个区域。因此，在处理多峰图像时，提出二维 OTSU 分割法。像素的灰度值分布以及该像素邻域像素的平均灰度值分布是二维 OTSU 阈值

收稿日期:2023-04-02； **修回日期**:2023-04-07。
基金项目:国家自然科学基金项目(61973180;62172249)。
作者简介:崔文静(1997-),女,硕士。
梁宏涛(1979-),男,博士,副教授。
引用格式:崔文静 ,李 帅,彭天文,等.改进蜜獾算法优化 OTSU 的图像分割研究[J].计算机测量与控制,2023,31(9):260-266.

划分需要考虑的两个因素，虽然分割的准确度有所提高，但计算的复杂性却大大增加[4]。为了应对计算复杂度的增加，许多学者引入麻雀算法、灰狼算法等智能优化算法对阈值进行搜索优化改进实时性差的缺陷[5]。

标准蜜獾算法[6]（honey badger algorithm，HBA）是于2021年提出的一种新型智能优化算法，蜜獾算法的思想来源于生物学中的蜜獾，这种动物有着聪明的觅食行为和攻击技能，能够克服各种困难，寻找到合适的食物或藏身之地。蜜獾算法具有结构简单、易实现；能够在多个领域内取得比较好的优化结果，且具有较强的鲁棒性。算法利用蜂蜜吸引度有效地保证了开发能力，有效的引导个体向最优个体靠拢，同时密度因子确保了算法从勘探阶段到开发阶段的平稳过渡[7]。HBA 算法相较于其他群体智能优化算法，其优化性能与收敛速度方面优势更加明显，但是算法在陷入局部最优解的问题上仍然具有提升空间[8]。

本文引入反向学习策略和柯西变异因子对蜜獾算法改进，提出了应用改进蜜獾算法的二维 OTSU 图像分割法，不仅提高了算法的收敛速度，而且可以获得更优的分割阈值，使得分割后的图像更加细致。

1　二维 OTSU 算法

二维 OTSU 算法利用像素灰度值分布和它邻域像素的平均灰度值分布两个维度构建二维直方图进行阈值分割[9]。假设一幅图像所有像素点总和为 $M\times N$，灰度级总数为 L，那么使用像素点灰度值 i 与平均灰度值 j 组成的二元组就可以描述图像中的任意像素点。利用每个二元组产生的频率可计算得出相应的联合概率密度，如式（1）所示。

$$P_{ij}=\frac{f_{ij}}{M\times N} \tag{1}$$

其中：f_{ij} 是像素灰度值为 i 时邻域灰度值为 j 的像素出现的次数。一般在计算过程中取偏离直方图对象线的点的概率为 $P_{ij}=0$。设阈值为（s，t）可以把图像划分为前景 J_0 和背景 J_1，那么前景和背景的概率值分别为：

$$P_{J_0}(s,t)=\sum_{i=1}^{s}\sum_{j=1}^{t}P_{ij} \tag{2}$$

$$P_{J_1}(s,t)=\sum_{i=s+1}^{L}\sum_{j=t+1}^{L}P_{ij} \tag{3}$$

均值矢量如式（4）～（6）所示。其中 μ_0，μ_1 分别为前景和背景的均值矢量，μ 为总体均值矢量。

$$\mu_0=(\mu_{0i},\mu_{0j})^T \tag{4}$$

$$\mu_1=(\mu_{1i},\mu_{1j})^T \tag{5}$$

$$\mu=(\mu_i,\mu_j)^T=\left(\sum_{i=1}^{s}\sum_{j=1}^{t}iP_{ij},\sum_{i=1}^{s}\sum_{j=1}^{t}jP_{ij}\right)^T \tag{6}$$

图像的离散测速矩阵如下所示。

$$S_{\text{img}}=P_{J_0}[(\mu_{0i}-\mu_i)^2+(\mu_{0j}-\mu_j)^2]+P_{J_1}[(\mu_{1i}-\mu_i)^2+(\mu_{1j}-\mu_j)^2] \tag{7}$$

根据式（7）所得离散度越大，类间方差越大。当 max {Simg} 取最大值时，（s，t）为最佳阈值。

2　蜜獾算法的优化

2.1　标准蜜獾算法

标准的蜜獾算法源于蜜獾为了进食而采取的两种行为模式。第一种行为是挖掘猎物，蜜獾通过自身的嗅觉判断猎物的大概位置，并在到达该位置后围绕猎物行动，最终选取适当的地点进行挖掘和捕获猎物。第二种行为是采蜜，蜜獾通过感知环境中的信息，直接定位蜜蜂巢穴并进行采蜜[10]。在算法中，这两种行为被转化为勘探阶段和开发阶段，以实现全局和局部搜索的平衡。在勘探阶段，算法通过多个候选解对问题空间进行探索，并保留最优解进行进一步开发。在开发阶段，算法通过对当前最优解进行局部搜索来获取更优的结果。

在蜜獾算法中，种群初始化是指随机生成一些解作为搜索空间的候选解，在这些解中寻找最优解的过程。种群大小的设定对于算法的性能有很大影响，通常需要根据问题的复杂度和搜索精度来确定种群大小。初始化方式如式（8）所示。

$$x_i=lb_i+r_1\times(ub_i-lb_i) \tag{8}$$

其中：ub，lb 表示搜索空间的上限和下限，r_1 是（0，1）之间的随机数，x_i 为 N 个候选个体的第 i 个个体的位置。

挖掘阶段中，气味强度 I 主要与猎物强度或集中强度 S 和蜜獾个体与猎物之间的距离 d_i 有关。

$$\begin{cases}S=(X_i-X_{i+1})^2\\ d_i=X_{prey}-X_i\\ I_i=r_2\times\dfrac{S}{4\pi d_i^2}\end{cases} \tag{9}$$

由式（9）可得，气味强度 I 与 S 成正比，与距离 d 的平方成反比，距离越近，气味强度与大。其中，r_2 是（0，1）之间的随机数。

挖掘过程中蜜獾的位置更新范围类似于心形线的形状，运动曲线通过式（10）进行模拟。

$$X_{\text{new}}=X_{\text{prey}}+F\times\beta\times I\times X_{\text{prey}}+F\times r_3\times\alpha\times d_i\times\left|\cos(2\pi r_4)\times[1-\cos(2\pi r_5)]\right| \tag{10}$$

$$\alpha=C\times\exp\left(\frac{-t}{t_{\max}}\right) \tag{11}$$

式中，X_{prey} 为猎物的位置，在算法中表示目前全局搜索得到的最优位置，β 代表的是蜜獾捕获猎物的能力（默认取值为6），α 为密度因子，可以确保从勘探阶段到开采阶段的平稳过渡，C 默认为 2，t 为当前迭代次数，$t_{\max}$ 为最大迭代次数，r_3、r_4、r_5 是（0，1）之间的三个不同的随机数，F 表示控制方向的参数，由式（12）确定。

$$F=\begin{cases}1, & r_6\leqslant 0.5\\ -1, & else\end{cases} \tag{12}$$

在采蜜阶段的情况可以用式（13）进行模拟。

$$X_{\text{new}}=X_{\text{prey}}+F\times r_7\times\alpha\times d_i \tag{13}$$

X_{prey} 为猎物位置，r_7 为（0，1）之间的随机数。

2.2 改进蜜獾算法

2.2.1 基于反向学习策略的种群初始化

HBA 采用的方式是随机生成初始种群，随机生成的种群存在分布不均的问题，会造成种群多样性降低，种群品质低下，甚至直接影响算法的收敛速度[11]，因此改进的 HBA 算法采用反向学习策略解决该问题。

反向学习策略不仅可以在传统的机器学习中应用，同时也可用于优化算法中的种群初始化。基于反向学习策略的种群初始化方法可以被归纳为以下几个步骤。

1）定义问题：首先，需要将优化问题定义为一个目标函数，其中包含了需要最小化或最大化的目标以及所有的约束条件。

2）反向生成个体：接下来，采用反向学习策略，在反向群体中选取优秀的种群作为下一代种群[12]。

3）随机扰动：为了增加种群的多样性，可以对每个个体进行随机扰动。这样做不仅可以避免个体陷入局部最优解中，而且还可以提高全局寻优能力。

4）约束检查：最后，需要对生成的每个个体进行约束检查，确保它们满足所有的约束条件。如果某个个体不满足约束条件，则需要重新生成。

假设随机初始化种群为 X_i，则对应的反向群体如式（14）所示。

$$X_i^* = rand(lb + ub) - X_i \tag{14}$$

其中：$rand$ 表示取（0，1）之间的随机数，lb、ub 为搜索空间的下界和上界。

反向学习策略就是通过选择更接近的个体成为种群的初始个体，从而使所有个体都可以离最优解更近一步，以便提高种群所有个体的收敛速度。同时，通过对每个个体进行随机扰动寻找更多有效区域来增加目标群体的多样性，从而提高算法的全局搜索能力[13]。

2.2.2 引入柯西变异策略

蜜獾优化算法主要通过挖掘阶段和采蜜阶段两个过程实现个体位置的更新。挖掘阶段的执行保证了算法前期的全局搜索能够更加广泛和全面；采蜜阶段，蜜獾群体的位置更新方向始终以食物源为目标，有利于深度搜索最优解[14]。虽然，以食物源方向为目标的搜索方法可以加快算法的收敛速度，但同时也存在搜索不充分等问题。如果食物源陷入局部最优，那么围绕食物源进行的搜索也将陷入局部最优[15]。针对上述问题，采用柯西变异因子对算法当前找到的最优解进行自适应变异，食物源的自适应变异过程如式（15）所示：

$$X_{\text{new}} = X_{\text{prey}} + X_{\text{prey}} \oplus \text{cauchy}(0,1) \tag{15}$$

$$f(x) = \frac{1}{\pi}\frac{t}{t+x^2}, -\infty < x < \infty \tag{16}$$

式（16）为标准柯西分布的概率密度函数，Cauchy 分布曲线与高斯分布相似，是一个由峰值到两端缓慢变化的钟形，无线接近 x 轴，该特征保证了可以迅速逃离局部极值点[16]。另外，Cauchy 分布的峰值较低，可以缩短变异后在邻域搜索的时间[17]。因此，获得可行解后采用柯西变异因子进行扰动，有利于增强算法的局部随机搜索能力，从而避免算法陷入局部最优的问题。

2.3 改进蜜獾算法伪代码

引入反向学习策略与柯西变异因子的 MHBA 算法伪代码如下所示。

算法 1：MHBA 算法

初始化最大种群 N，最大迭代次数 T，优化维度 D，搜索下界 lb 和上界 ub；

采用反向学习策略初始化种群数 X_i^*；

计算每个蜜獾个体的适应度值保存最佳位置及适应度值；

While($t<T$) do

for $i=1, 2, \cdots, N$ do

if $r<0.5$, then

标准蜜獾挖掘阶段

else 标准蜜獾采蜜阶段更新

end

end

计算新蜜獾个体的适应度并排序

更新食物源位置及其适应度；

引入柯西变异因子，以避免算法陷入局部最优；

$t=t+1$；

end

输出全局最优个体适应度值及位置

具体流程包括初始化种群数量、迭代次数、优化维度和搜索范围；使用反向学习策略初始化种群数量；计算每个蜜獾个体的适应度值，并保存最佳位置及适应度值；在当前迭代次数小于最大迭代次数之前，对所有蜜獾进行标准蜜獾采蜜或挖掘操作；计算新蜜獾个体的适应度并排序，更新食物源位置及其适应度；通过引入柯西变异因子来避免算法陷入局部最优，并不断更新迭代次数。最终可以输出全局最优个体的适应度值及位置。

2.4 算法测试与分析

2.4.1 基准测试函数

本文实验环境为 11th Gen Intel（R）Core（TM）i5-1135G7，MATLAB2019b。为验证 MHBA 算法的性能，本文选取了 8 个测试函数在 MATLAB2019b 上进行仿真实验。这些测试函数是在计算智能领域被广泛应用的标准基准函数，用于验证各种优化算法的性能。在本文中，使用这些测试函数对比分析了 MHBA、HBA 和 GWO 算法的优化效果。这些算法都是基于群体智能的元启发式算法，适用于各种优化问题，例如函数优化、组合优化和约束优化等。然而，在不同的问题和问题规模下，各个算法的表现往往会有所不同。因此，测试函数的比较结果可以为不同应用场景下的算法选择提供重要的参考。测试函数信息如表 1 所示。其中，$f_1 \sim f_2$ 为单个峰值的测试函数，$f_3 \sim f_5$ 为多个峰值的测试函数，$f_6 \sim f_8$ 为固定维度的测试函数[18]。

表 1　测试函数说明

测试函数	寻优范围	理论最优解
$f_1(x)=\sum_{i=1}^{n}x_i^2$	[−100，100]	0
$f_2(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	[−10，10]	0
$f_3(x)=\sum_{i=1}^{n}-x_i\sin(\sqrt{\lvert x_i\rvert})$	[−500，500]	−12 569.5
$f_4(x)=-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos 2\pi x_i\right)$	[−32，32]	0
$f_5(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	[−600，600]	0
$f_6(x)=\left[\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6}\right]^{-1}$	[−65.536，65.536]	1
$f_7(x)=-\sum_{i=1}^{10}[(x-a_i)(x-a_i)^T+c_i]^{-1}$	[0，1]	−10
$f_8(x)=-\sum_{i=1}^{4}c_i\exp[-\sum_{j=1}^{4}a_{ij}(x_j-p_{ij})^2]$	[0，1]	−3.86

2.4.2　实验结果分析

本实验选择标准 HBA 算法、GWO 算法与 MHBA 算法在选取的 8 个基准测试函数上进行仿真实验。为保证实验公平性，种群规模 $N=30$，空间维度 $D=30$，最大迭代次数 $T=100$，记录每次运行结果。实验结果采用收敛曲线进行描述，方便直观算法的优化效果，如图 1～3 所示，横轴代表迭代次数，横轴代表适应度值[19]。不同算法的测试结果统计如表 2 所示。

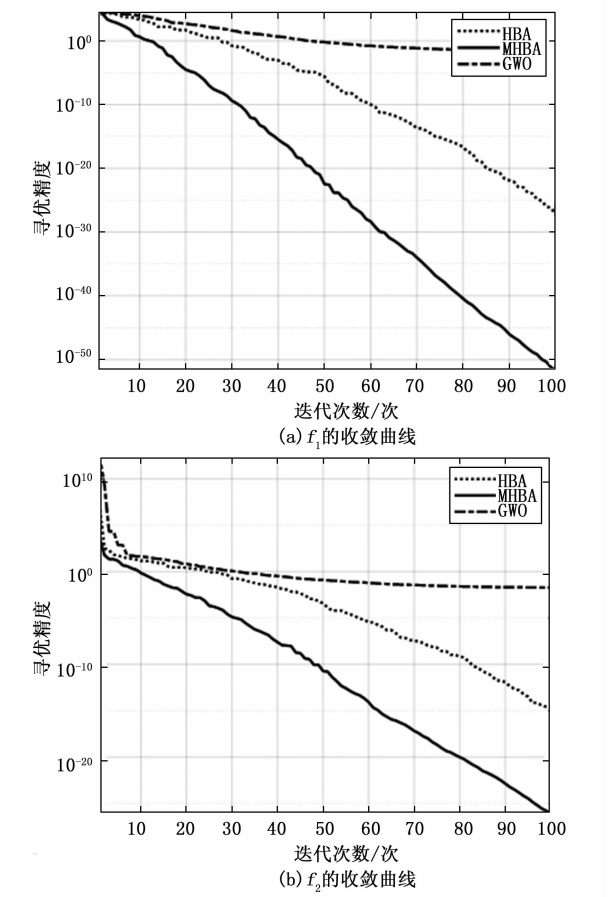


图 1　单峰测试函数上的收敛效果图

表 2　不同算法测试结果统计

函数	算法	最优值	平均值	标准差
F1	MHBA	4.25×10^{-52}	3.56×10^{-52}	1.26×10^{-52}
	HBA	5.31×10^{-28}	2.11×10^{-28}	6.54×10^{-28}
	GWO	1.43×10^{-2}	2.32×10^{-2}	3.54×10^{-2}
F2	MHBA	3.89×10^{-28}	3.94×10^{-28}	3.58×10^{-28}
	HBA	4.21×10^{-16}	9.13×10^{-16}	2.24×10^{-16}
	GWO	8.46×10^{-2}	6.34×10^{-2}	5.78×10^{-2}
F3	MHBA	−7 228.436	−7 208.392	−7 207.363
	HBA	−6 625.895	−6 388.278	−6 386.126
	GWO	−5 507.697	−5 294.635	−5 499.693
F4	MHBA	0	0	0
	HBA	9.25×10^{-15}	3.59×10^{-15}	2.63×10^{-15}
	GWO	5.46×10^{-2}	7.23×10^{-2}	6.24×10^{-2}
F5	MHBA	4.68×10^{-17}	2.38×10^{-17}	3.42×10^{-17}
	HBA	2.53×10^{-15}	8.63×10^{-15}	5.92×10^{-15}
	GWO	3.75×10^{-2}	1.54×10^{-2}	3.64×10^{-2}
F6	MHBA	1	1	1
	HBA	1.34×10^{0}	1.64×10^{0}	1.08×10^{0}
	GWO	9.75×10^{0}	8.36×10^{0}	8.89×10^{0}
F7	MHBA	-10.36×10^{0}	-10.72×10^{0}	2.05×10^{0}
	HBA	-10.12×10^{0}	-9.96×10^{0}	2.49×10^{0}
	GWO	-9.87×10^{0}	-9.74×10^{0}	3.35×10^{0}
F8	MHBA	-3.86×10^{0}	-3.86×10^{0}	1.20×10^{0}
	HBA	-3.86×10^{0}	-3.85×10^{0}	3.36×10^{0}
	GWO	-3.84×10^{0}	-3.83×10^{0}	4.37×10^{0}

通过表 2 的测试函数对比结果可直接反映出算法的寻优效果和收敛速度的能力，MHBA 算法在不同测试函数上都表现出最佳的优化性能。在求解多峰函数 f_4 时，其最优解、平均值和标准差都为 0，说明 MHBA 算法在这个测试函数的求解过程中都达到了理想的最优值。在求解单峰函数 f_1 和 f_2 时，MHBA、HBA、GWO 都没有达到求解精度，但是 HBA 要较 GWO 的寻优精度大幅提升，MHBA 又在 HBA 基础上提升了 10～20 个数量级。在求解固定维度的测试函数时，虽然三种算法都能到达理想的寻优精度附近，但是 HBA 和 GWO 的收敛精度要相对较差一点。

综上所述，在求解单峰测试函数、多峰测试函数以及

固定维度的测试函数时，无论是在算法的寻优精度还是寻优速度上，融入反向学习策略与柯西变异因子的 MHBA 相比于原始的 HBA 以及 GWO 算法均表现出了一定的优势。

对于单峰测试函数 $f_1 \sim f_2$，虽然三种算法在迭代次数 100 次的情况下都没有达到最优解，但是加入反向学习和柯西变异策略的 MHBA 算法的收敛速度远远领先其他两个算法。而且，100 次迭代后 MHBA 算法在精度方面较其他两个算法要提高了 10^{20} 数量级。

对于多峰测试函数 $f_3 \sim f_5$，图 2（a）显示在算法迭代初期，虽然 MHBA 算法的收敛速度相对较慢，但在迭代后期，其收敛速度快速加快并最先达到更高的寻优精度。这是因为 MHBA 算法在柯西变异因子的帮助下，能够更好地跳出局部最优解，并最终接近全局最优解图 2（b）的对比效果更加明显，MHBA 算法不仅表现出了极佳的收敛速度也展现了精准的寻优能力。图 2（c）虽然都没达到最优精度，但 MHBA 的收敛速度要快的多。

固定维度函数上的测试效果可以体现出算法的全局遍历和局部搜索的均衡能力，图 3 显示 MHBA 算法的收敛速度远快于 HBA 算法。虽然 HBA 算法在固定维度函数上得到的结果比较趋近于测试函数理论值，但是图 3（c）可以看出 MHBA 算法的结果精度比 HBA 算法更接近理论最优值。

综上所述，对于上述 8 个基准测试函数而言，MHBA 算法的收敛速度和求解精度较 HBA 算法和经典的 GWO 算法更出色，尤其是在多峰函数的求解过程中，MHBA 算法可以在较短的时间内找到理想的最优解，这对于实际应用中的复杂优化问题具有重要的意义。而在单峰函数的求解过程中，MHBA 算法虽然没有达到理想的精度，但是相比于 HBA 和 GWO 算法仍然表现出了更好的优化性能。在固定维度的测试函数中，三种算法的性能差距不大，但 MHBA 算法仍然略微优于 HBA 和 GWO 算法。实验充分证明了引入反向学习策略和柯西变异因子可以有效提高 HBA 算法收敛速度和精确度，MHBA 算法在全局搜索和局部开发的均衡性上要比 HBA 算法更加优秀。

3 基于 MHBA-OTSU 的图像分割研究

二维 OTSU 算法原理是寻找一个合适的阈值对图像进行分割，通过该阈值分割得到的图像的前景与背景的类间方差最大[20]，因为考虑了图像的灰度信息和空间邻域信息，所以算法存在计算量大、效率低的问题。群智能优化算法通过模拟一些生命种群等的觅食或其他行为，通过群体之间的合作交流，花费较少的计算时间很快的找到更多的食物[21]，被广泛的应用于函数优化及参数寻优等方面。于是本文将 MHBA 方法应用到图像分割领域，以实现阈值的寻优，从而更高效的获得最优阈值。

MHBA 算法以图像最大类间方差函数式（7）作为优化的适应度函数，函数表达式如式（17）所示。以迭代选优的方式求解得到图像的最优分割阈值。算法的寻优边界设置为 0～255。

$$f(x) = \max\{S_{\text{img}}\} \tag{17}$$

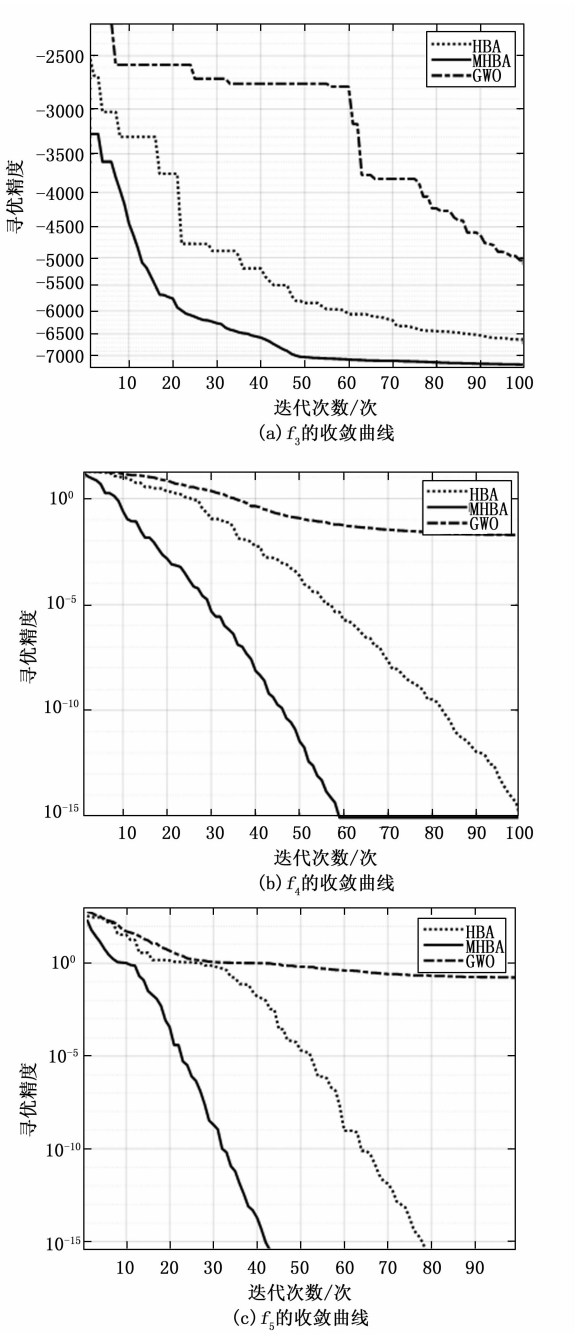


图 2 多峰测试函数上的收敛效果图

MHBA 优化的二维 OTSU 获取图像最优分割阈值的流程如图 4 所示。

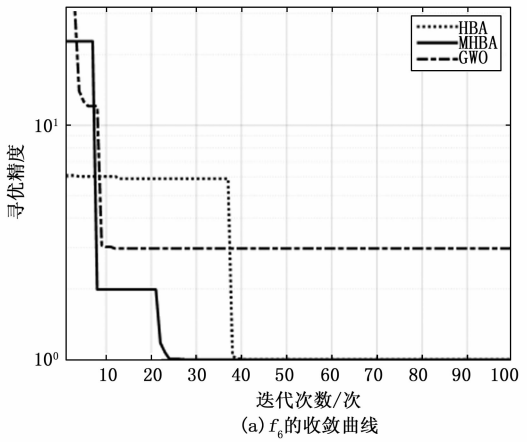
1）初始化种群。设置最大种群数 N，最大迭代次数 T，优化维度 D，搜索空间的上下界 UB、LB 等相关参数；采用反向学习策略对种群进行初始化。

2）载入图像，进行灰度处理，计算最大类间方差。

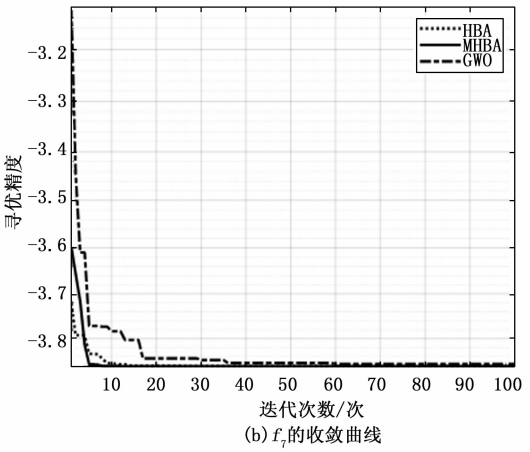
3）将最大类间方差函数式（17）作为蜜獾的适应度函数。

4）计算个体的适应度。根据适应度大小找出食物源的位置并更新个体的位置。

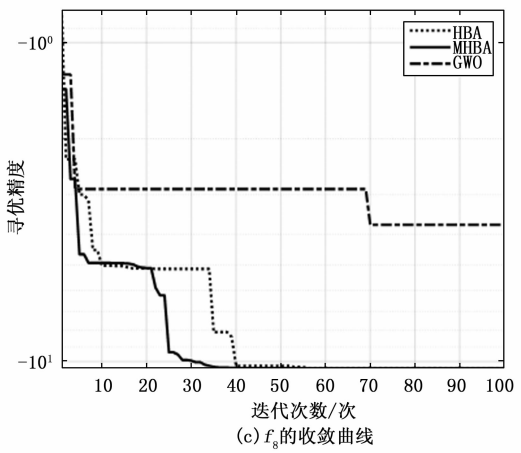
5）引入柯西变异因子，对算法当前得到的可行解进行



(a) f_6的收敛曲线



(b) f_7的收敛曲线



(c) f_8的收敛曲线

图 3　固定维度测试函数上的收敛效果图

自适应扰动，避免算法陷入局部最优。

6）迭代次数增加，判断是否达到终止条件，如果达到，则输出最优解；否则，返回执行步骤 4）。

7）根据算法获得最优解作为阈值进行图像分割。

4　实验仿真与结果分析

实验采用 MATLAB2019b 进行仿真。为保证搜索的阈值可以在最佳阈值附近收敛，算法设置 MHBA 算法的群体规模为 $N=30$，空间维度 $D=30$，最大迭代次数 $T=100$。为了直观对比图像分割所得到的效果，并验证本文改进算法

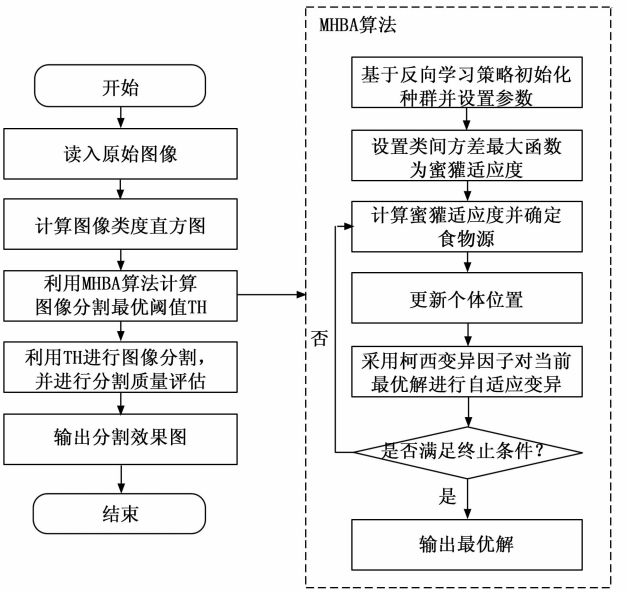


图 4　MHBA-OTSU 的图像分割流程

的可行性[22]，选取伯克利大学提供的开源数据集 BSDS500 中的三幅图像作为分割对象[23]，分别采用二维 OTSU 算法和 MHBA-OTSU 进行图像分割实验对比，分割效果图如图 5 所示。

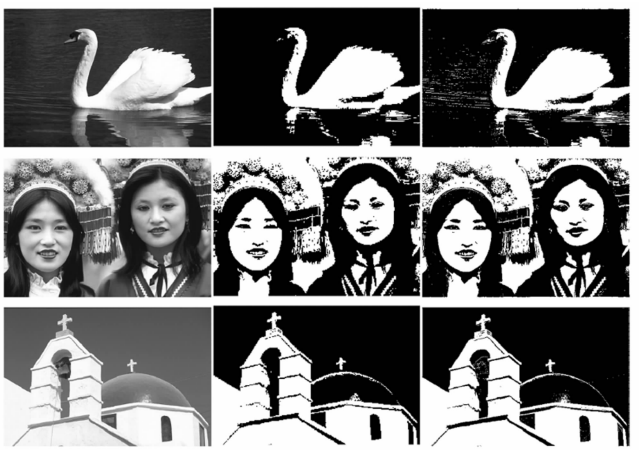

(a)灰度图　(b)二维OTSU　(c)MHBA-OTSU

图 5　图像分割对比图

其中（a）为对图像进行灰度处理后的效果，分别选取了动物、人物及场景三种不同的类别。从图（b）（c）分别为二维 OTSU 和 MHBA-OTSU 进行图像分割的效果图。在动物图像分割中，二维 OTSU 对于水面波纹没有多余的检测，但是因为头部黑色的羽毛与背景颜色太相近导致头部尤其是嘴巴部分产生了较多的缺失，相反 MHBA-OTSU 虽然因为水纹的晃动产生了一定的误判，但是头部检测还是比较完整的，而且在水面倒影的部分也是检测的细节比较到位。虽然人物面部轮廓上两个算法分割效果差不多，但是在细节方面对比是比较明显的，二维 OTSU 分割的人物头饰部分比较模糊，层次分明效果差，而且人物的眼睛

部分由于眼白部分比较少导致二维 OTSU 分割后的整个眼睛呈现黑色，而 MHBA-OTSU 部分对于头饰及眼睛部分的细节处理较好，可以分明的看到人物的黑眼球和白眼球部分。在环境分割图中二维 OTSU 检测的倒钟完全融入了黑色背景部分，同时图像的线条呈现了锯齿状，而 MHBA-OTSU 算法可以将倒钟识别分割出来并且边缘部分要比二维 OTSU 清晰许多，呈现比较光滑的线条。

本文通过计算三种算法的时间效率作为客观评价标准，如表 3 所示。

表 3 算法图像分割时间比较 s

算法	动物图像	人物图像	场景图像
二维 OTSU	1.363 7	4.566 3	2.263 8
MHBA-OTSU	1.216 3	3.430 6	1.196 3

动物和场景图像的原图对象简单，色彩也相对单一，经过灰度处理后的进行分割的时间要少很多。而人物图像的原图色彩相对丰富，而且细节也比较多，在进行图像分割的时候花费的时间要久一些。从表 3 可以看出 MHBA-OTSU 算法在融入改进蜜獾算法后在效率方面有了较大的提高，有效减少了算法的运行时间。

5 结束语

本文提出的改进蜜獾算法，首先加入了反向学习策略进行种群初始化，用于解决种群个体随机性和均衡性冲突的问题，保证了种群分布的优越性，有效避免了算法的过早收敛。其次，引入柯西变异因子对可行解进行扰动，有利于算法跳出局部最优，提高局部搜索过程，提高算法寻优精度。通过对 8 个经典测试函数进行查验，与 GWO 和 HBA 算法进行对比仿真实验结果表明，MHBA 在寻优速度和收敛精度上都有大幅提高，具有收敛速度更快、收敛精度更高的优势。将 MHBA 应用于图像分割领域的 OTSU 阈值寻优，在图像分割方面表现出了分割效果更加细致，表明 MHBA 在图像分割领域不仅提高了算法的收敛速度，而且可以获得更优的分割阈值。

参考文献：

[1] 宋 杰，许 冰，杨森中. 基于自适应步长果蝇优化算法图像分割 [J]. 计算机工程与应用，2020，56 (4)：184 - 190.

[2] 赵 凤，孔令润，马改妮. 多目标粒子群和人工蜂群混合优化的阈值图像分割算法 [J]. 计算机工程与科学，2020，42 (2)：281 - 290.

[3] 罗 钧，杨永松，侍宝玉. 基于改进的自适应差分演化算法的二维 Otsu 多阈值图像分割 [J]. 电子与信息学报，2019，41 (8)：2017 - 2024.

[4] 徐 武，文 聪，唐文权，等. 基于 Lab 颜色空间的融合改进二进制量子 PSO 和 Otsu 优化算法 [J]. 计算机应用与软件，2022，39 (6)：265 - 268，349.

[5] 江 妍，马 瑜，梁远哲，等. 基于分数阶麻雀搜索优化 OTSU 肺组织分割算法 [J]. 计算机科学，2021，48 (S1)：28 - 32.

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

[8] 董红伟，李爱莲，解韶峰，等. 多策略改进的蜜獾优化算法 [J/OL]. 小型微型计算机系统：1 - 12. http://kns.cnki.net/kcms/detail/21.1106.TP.20221123.1326.032.html

[9] HAN E，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.

[10] 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.

[11] 张兴辉，樊秀梅，阿喜达，等. 反向学习的灰狼算法优化及其在交通流预测中的应用 [J]. 电子学报，2021，49 (5)：879 - 886.

[12] 余修武，黄露平，刘 永，等. 融合柯西折射反向学习和变螺旋策略的 WSN 象群定位算法 [J]. 控制与决策，2022，37 (12)：3183 - 3189.

[13] 余伟伟，谢承旺. 一种多策略混合的粒子群优化算法 [J]. 计算机科学，2018，45 (S1)：120 - 123.

[14] ABUALIGAH L，DIABAT A，MIRJALILI S，et al. The arithmetic optimization algorithm [J]. Computer methods in applied mechanics and engineering，2021，376：113609.

[15] JIANG F，WANG L，BAI L. An improved whale algorithm and its application in truss optimization [J]. Journal of Bionic Engineering，2021，18 (3)：721 - 732.

[16] 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.

[17] 李爱莲，全凌翔，崔桂梅，等. 融合正余弦和柯西变异的麻雀搜索算法 [J]. 计算机工程与应用，2022，58 (3)：91 - 99.

[18] 张 琳，汪廷华，周慧颖. 一种多策略改进的麻雀搜索算法 [J]. 计算机工程与应用，2022，58 (11)：133 - 140.

[19] 尹德鑫，张达敏，蔡朋宸，等. 改进的麻雀搜索优化算法及其应用 [J]. 计算机工程与科学，2022，44 (10)：1844 - 1851.

[20] 梁远哲，马 瑜，江 妍，等. 基于分数阶混合蝙蝠算法的 Otsu 图像分割 [J]. 计算机工程与设计，2021，42 (11)：3091 - 3098.

[21] 张 军，温秀平，陈 巍. 融合改进鬣狗优化和 Tsallis 熵的图像分割 [J]. 计算机工程与设计，2022，43 (12)：3493 - 3502.

[22] 曹 爽，安建成. 狼群优化的二维 Otsu 快速图像分割算法 [J]. 计算机工程与科学，2018，40 (7)：1221 - 1226.

[23] 史春天，曾艳阳，侯守明. 群体智能算法在图像分割中的应用综述 [J]. 计算机工程与应用，2021，57 (8)：36 - 47.