

## 改进的生物地理学算法及其在图像分割中的应用

张新明<sup>1,2</sup>, 尹欣欣<sup>1</sup>, 冯梦清<sup>1</sup>, 范晓艳<sup>1</sup>

(1. 河南师范大学计算机与信息工程学院, 河南 新乡 453007;

2. 河南省高校计算智能与数据挖掘工程技术研究中心, 河南 新乡 453007)

**摘要:** 为了增强生物地理学优化算法(BBO)在图像多阈值分割应用中的全局搜索能力,提高其优化性能,提出一种改进的生物地理学算法(IBBO)。首先,引入多源迁移算子,该算子能更好地从搜索空间中生成新特征值,有效提高种群的多样性;其次,创建一种新型的动态变异算子,该算子能够动态地改变变异幅度,提高算法运算效率,使算法快速收敛到全局最优解;随后,将原来的精英选择算子改为贪婪选择算子,即采用优胜劣汰的策略加快算法收敛速度;最后将其应用到基于最大熵的多阈值分割中。图像分割实验结果表明,IBBO算法运行速度远远快于穷举算法,优化性能优于标准BBO算法和PSO算法。

**关键词:** 优化算法; 生物地理学优化算法; 图像分割; 多阈值分割; 最大熵

中图分类号: TP391.41 文献标志码: A 文章编号: 1671-637X(2015)12-0024-05

## Improved Biogeography-Based Optimization and Its Application in Image Segmentation

ZHANG Xin-ming<sup>1,2</sup>, YIN Xin-xin<sup>1</sup>, FENG Meng-qing<sup>1</sup>, FAN Xiao-yan<sup>1</sup>

(1. College of Computer and Information Engineering, Henan Normal University, Xinxiang 453007, China;

2. Engineering Technology Research Center for Computing Intelligence & Data Mining, Xinxiang 453007, China)

**Abstract:** In order to enhance the global search ability of Biogeography-Based Optimization (BBO) in multi-threshold image segmentation, and improve its optimization performance, an Improved BBO (IBBO) algorithm is proposed. Firstly, a polyphyletic migration operator is introduced, which can better generate new eigenvalue from the searching space and effectively improve the population diversity. Secondly, a new dynamic mutation operator is created, which can dynamically change the mutation range and improve the operation efficiency of algorithm, enabling the algorithm to quickly converge to the global optimum. Then, a greedy selection operator is used instead of the original elitist selection operator, to accelerate the convergence process by using the strategy of survival of the fittest. Finally, IBBO algorithm is applied to the maximum entropy-based multi-threshold segmentation. Experimental results of image segmentation show that the proposed IBBO algorithm operates much faster than the exhaustive algorithm, and the optimization performance is better than that of the standard BBO algorithm and PSO algorithm.

**Key words:** optimization algorithm; biogeography-based optimization algorithm; image segmentation; multi-threshold segmentation; maximum entropy

### 0 引言

生物地理学算法(BBO)是SIMON D在2008年提出

的一种基于岛屿生物地理学平衡理论的群体进化算法<sup>[1]</sup>。BBO算法拥有进化算法共同的特点,即以一定的概率在候选解之间共享信息;同时,BBO算法也有其独特之处,BBO算法通过个体的适宜度值计算每一代的迁入率和迁出率,使质量低的解有较高的概率接受质量高的解的特征来提高自身的质量。BBO算法自被提出以来,在各种基准函数的优化上表现出优良的性能,并且被成功地应用到各种现实问题中。例如,文献[2]提出

收稿日期:2015-01-23

修回日期:2015-02-15

基金项目:河南省重点科技攻关项目(132102110209);河南省基础与前沿技术研究计划项目(142300410295)

作者简介:张新明(1963—),男,湖北孝感人,教授,研究方向为智能优化算法、数字图像处理和模式识别等。

一种基于实数编码的 RCBBO 算法,用于解决连续函数的全局优化问题;文献[3]将 BBO 算法应用到图像分割中,提出了 BBOIS 算法并证明了其优越性。然而,BBO 算法也有不足,它的收敛速度在进化过程开始时相对较快,但是其后容易陷入局部最优和早熟<sup>[4]</sup>。为了提高 BBO 算法的性能,改正以上缺点,研究者们提出了许多新型的 BBO 算法。例如,文献[5]提出了一种多亲交叉的迁移模型,平衡了算法的探索和开采能力;文献[6]针对变异算子进行研究,提出了基于高斯变异的生物地理学优化模型,进一步提高算法优化精度;文献[7]提出了一种混合迁移算子的算法,并将其应用于约束优化问题得到较好的效果。因此,设计良好的迁移算子和变异算子是提高 BBO 算法优化性能的关键。

阈值分割是一类应用广泛且易于实现的图像分割方法,其关键是如何选取合适的阈值以获得最佳分割效果。目前常用的阈值法主要有最大 Renyi 熵法<sup>[8]</sup>、最大 Tsallis 熵法<sup>[9]</sup>、最大熵法和交叉熵法<sup>[10]</sup>等方法。多阈值分割需要在全灰度范围内搜索一个最佳阈值组合,计算量庞大,时间和空间复杂度较高。研究者们从改进最优阈值搜索机制的角度入手,引入优化算法解决此问题<sup>[11]</sup>。

本文针对图像多阈值分割问题,引入多源迁移算子和创建一种新型的动态变异算子,提高了算法的运算效率,使算法能够快速收敛到全局最优解,并且将这种改进的生物地理学算法 (IBBO) 应用到基于一维灰度直方图的最大熵多阈值分割中,以便获得一种高效的适用于多阈值分割的优化算法。

## 1 基本生物地理学算法

在 BBO 中,每个个体称为一个栖息地,每个栖息地用适宜度指数 (HSI) 衡量其好坏,栖息地的 HSI 与生物多样性成正比。描述栖息地适于居住程度的特征变量组成的  $D$  维实数向量称为适宜度向量 (SIV, 公式中用  $V_{\text{SIV}}$  表示)。栖息地之间通过迁移和突变进行物种之间信息的交换与共享,提高物种多样性。

在生物地理学算法中,每个栖息地的迁入率和迁出率是该栖息地物种数量的函数,用于栖息地之间信息的分享。迁入率  $\lambda_i$  和迁出率  $\mu_i$  的算式为

$$\lambda_i = I(1 - S_i/S_{\max}) \quad (1)$$

$$\mu_i = E(S_i/S_{\max}) \quad (2)$$

式中: $I, E$  分别表示最大迁入率和最大迁出率; $S_i$  表示当前种群数; $S_{\max}$  表示最大种群数。

BBO 包含两大基本的操作即迁移和变异。迁移操作通过分享栖息地之间的特征值改变选中的栖息地的 SIV。迁移操作可以表示为

$$H_i(V_{\text{SIV}}) \leftarrow H_e(V_{\text{SIV}}) \quad (3)$$

式中: $H_i$  是准备迁入的栖息地; $H_e$  是准备迁出的栖息地。迁移操作的伪代码如算法 1 所示。其中, $N$  表示栖息地的数量, $\text{rand}(0,1)$  表示均匀分布在  $(0,1)$  之间的随机实数。

**算法 1** 迁移算子。

```
for  $i = 1$  to  $N$  do
    根据  $\lambda_i$  选择栖息地  $H_i$ 
    if  $\text{rand}(0,1) < \lambda_i$  then
        for  $e = 1$  to  $N$  do
            根据  $\mu_e$  选择栖息地  $H_e$ 
            对  $H_i$  执行式(3)
        end for
    end if
end for
```

变异操作模拟了栖息地生态环境的突变,用于增加种群的多样性。在 BBO 算法中,栖息地  $H_i$  中的一个 SIV 被选中用随机生成的 SIV 代替,其变异概率  $m_i$  与它的物种数量的先验概率  $P_i$  有关。变异概率的算式为

$$m_i = m_{\max}(1 - P_i/P_{\max}) \quad (4)$$

式中: $m_{\max}$  是自定义参数,代表最大变异概率; $P_{\max} = \max P_i$ 。变异算子伪代码如算法 2 所示。

**算法 2** 变异算子。

```
for  $i = 1$  to  $N$  do
    通过概率  $P_i$  计算  $m_i$ , 根据  $m_i$  选择  $H_i(V_{\text{SIV}})$ 
    if  $\text{rand}(0,1) < m_i$  then
        用随机生成的  $V_{\text{SIV}}$  替代  $H_i(V_{\text{SIV}})$ 
    end if
end for
```

## 2 改进的生物地理学优化算法

### 2.1 多源迁移算子

BBO 具有较好的开采能力,是因为迁移算子使得 BBO 能够在栖息地之间分享信息。然而,从式(3)中可以看出,这个算子仅仅通过直接复制原有的特征来生成新解,不能从未开采的搜索空间中生成新解,导致搜索过程可能陷入局部最优,造成 BBO 算法缺乏探索能力。为了增强算法的探索能力,引入了多源迁移算子<sup>[4]</sup>,其表示为

$$H_i(V_{\text{SIV}}) \leftarrow H_e(V_{\text{SIV}}) + \Phi_{i,\text{SIV}}(H_e(V_{\text{SIV}}) - H_r(V_{\text{SIV}})) \quad (5)$$

式中: $H_i$  是选定的要进行迁移的栖息地; $H_e$  是随机选择的一个 HSI 较高的栖息地; $H_r$  是一个随机选择的栖息地,且  $r, i, e$  三者应互不相等; $\Phi_{i,\text{SIV}}$  是服从  $[-1, 1]$  之间均匀分布的随机数。多源迁移算子的伪代码如算

法3所示,通过多达4个栖息地( $H_i, H_e, H_r, H_s$ )的特征来组成新的候选解  $V_i$ 。

**算法3 多源迁移算子。**

对于选定的栖息地  $H_i$

for  $d = 1$  to  $D$  do

if  $rand(0,1) < \lambda_i$  then

根据  $\mu_e$  选择栖息地  $H_e$

if  $rand(0,1) < \mu_e$  then

随机选择  $r \neq i \neq e$

$H_i(V_{SIV}) \leftarrow H_e(V_{SIV}) + \Phi_{i,SIV}(H_e(V_{SIV}) - H_r(V_{SIV}))$

else

随机选择  $s \neq i$

$H_i(V_{SIV}) \leftarrow H_s(V_{SIV})$

end if

else

$H_i(V_{SIV}) \leftarrow H_i(V_{SIV})$

end if

end for

多源迁移算子用相对较好的栖息地作为基础,通过随机选择的栖息地  $H_r$  产生对称干扰,带来更多新的信息,借此开发未探索到的可行空间,扩大搜索范围。如果条件  $rand(0,1) < \mu_e$  没有满足,则说明选择的  $H_e$  不够好,须从其他的栖息地迁入特征。栖息地  $H_e$  的作用在于增强算法的开采能力,而其他两个从当前种群中随机选取的栖息地  $H_r$  和  $H_s$  的作用在于强调算法的探索能力,如此使算法在开采和探索之间取得适当的平衡。

## 2.2 动态变异算子

在标准 BBO 算法中,用随机生成的一个特征值替换选中的要进行变异的栖息地特征值,以此提供更多的搜索目标,但是这种随机的变异算子会导致 BBO 收敛速度慢。为此,提出了一种动态变异算子,在原特征值的基础上,增加一个变异因子,随着迭代次数的改变动态地改变变异的幅度,使算法能够快速收敛到全局最优解。动态变异算子表示为

$$H_k(V_{SIVd}) \leftarrow H_k(V_{SIVd}) \pm ((rand/(G+d)) * H_k(V_{SIVd}) + S_{step}) \quad (6)$$

$$S_{step} = 4 - (4-1) * G/M \quad (7)$$

式中:  $H_k(V_{SIVd})$  表示第  $k$  个候选解的第  $d$  个特征值被选中进行变异操作;  $G$  表示当前种群的迭代次数;  $M$  表示最大迭代次数。候选解随迭代次数的增加逐渐收敛于全局最优解附近,因此,使变异幅度随迭代次数的增加逐渐减小,避免出现偏离最优解太多的变异值,加快了算法的收敛速度。  $S_{step}$  也是动态变化的,其取值范围为  $[1,4]$ ,其值随着迭代次数的增加逐渐减小。  $S_{step}$  的值不仅保证了产生的变异的变化量不为零,而且也起到了

动态调整的作用。  $H_k(V_{SIVd})$  在多阈值分割中表示为解向量中的第  $d$  个阈值分量,随着  $d$  的增加,  $H_k(V_{SIVd})$  也增大。式(6)产生的变化量随着  $d$  的增大而减小,可以有效地平衡各个特征值的变化。

## 2.3 贪婪选择算子

在标准 BBO 算法中,采用精英保留策略保留每一代群体中最优的解。一般选择保留每一代群体中前两个最优解,用以替换下一代群体中最差的两个解。本文采用贪婪选择策略保留每一代的最优解,即将经过迁移和变异操作产生的新一代解集与上一代解集进行比较,若新解优于旧解,就用新解替代旧解;否则,保留旧解,以此加快算法的收敛速度。贪婪选择算子伪代码表示为算法4,其中,  $N$  代表种群总数,  $value$  指解的适宜度值(HSI)。

**算法4 贪婪选择算子。**

for  $i = 1:N$

if  $value(i) < value_{new}(i)$

用新解替代旧解

end if

end for

## 3 基于改进 BBO 算法的图像多阈值分割

### 3.1 最大熵多阈值分割法

假设一幅图像有  $L$  个灰度级 ( $L$  常取 256),  $M$  个像素,灰度值为  $a$  的像素个数为  $h(a)$ ,  $a = 0, 1, 2, \dots, L-1$ , 则  $M = \sum_{a=0}^{L-1} h(a)$ , 灰度值为  $a$  的像素的概率密度为:  $P_a = h(a)/M, 0 \leq a \leq L-1$ 。

基于最大熵的多阈值分割方法描述如下:对于一组任意给定的阈值  $[x_1, x_2, \dots, x_D]$ , 将图像分割成  $D+1$  部分的概率分布所对应的熵为

$$\begin{cases} H_0 = - \sum_{a=0}^{x_1} \frac{P_a}{\omega_0} \ln\left(\frac{P_a}{\omega_0}\right) & \omega_0 = \sum_{a=0}^{x_1} P_a \\ H_1 = - \sum_{a=x_1+1}^{x_2} \frac{P_a}{\omega_1} \ln\left(\frac{P_a}{\omega_1}\right) & \omega_1 = \sum_{a=x_1+1}^{x_2} P_a \\ \vdots \\ H_D = - \sum_{a=x_{D-1}+1}^{L-1} \frac{P_a}{\omega_D} \ln\left(\frac{P_a}{\omega_D}\right) & \omega_D = \sum_{a=x_{D-1}+1}^{L-1} P_a \end{cases} \quad (8)$$

最大熵公式为

$$f([x_1, x_2, \dots, x_D]) = H_0 + H_1 + \dots + H_D \quad (9)$$

使得式(9)取得最大值的一组阈值  $[x_1, x_2, \dots, x_D]$  即为最优的阈值向量。为了取得最优的阈值向量,必须对解空间进行遍历搜索(穷举搜索),但是这种方法计算量非常大,浪费时间和资源。为此,需要引入智能优化算法。

### 3.2 IBBO 算法应用到图像多阈值分割中

假设对图像进行  $D$  阈值分割,则解向量为  $X =$

$[x_1, x_2, \dots, x_D]$ 。以最大熵法作为分割准则,在解空间 0 ~ 255 灰度级之间采用 IBBO 算法进行优化选择,使得式(9)取得最大值的解向量即为最优解。将式(9)作为衡量各个解的 HSI 的目标函数,候选解相当于 BBO 算法中的栖息地,解向量中的阈值相当于栖息地的特征向量。基于 IBBO 算法的图像多阈值分割法的步骤如下所述:

- 1) 读入图片,建立一维直方图,初始化参数,设最大迭代次数为  $M$ ,种群数量为  $N$ ,阈值个数为  $D$ ;
- 2) 随机生成初始化种群  $X$ ,  $X$  为  $N$  行  $D$  列矩阵,并对每一个解的分量由小到大排序;
- 3) 用式(9)评估每个栖息地  $X_j$  的 HSI,  $j = 1, \dots, N$ ,按照 HSI 值的降序对  $X_j$  排序;
- 4) 根据式(1)和式(2)计算迁入率、迁出率;
- 5) 执行算法 3 多源迁移操作;
- 6) 根据式(6)执行动态变异操作;
- 7) 用式(9)评估每个栖息地  $X_j$  的 HSI,  $j = 1, \dots, N$ ;
- 8) 执行算法 4 贪婪选择算子,保留较优的解,并按照 HSI 值的降序对  $X_j$  排序;
- 9) 若没有达到停止准则,转到 4), 否则,循环停止,获得最优阈值向量;
- 10) 用求得的最优阈值向量进行图像分割,输出分割后的图像。

#### 4 实验结果与分析

为了检验 IBBO 算法的有效性,将 IBBO 算法与最大熵分割法则结合,进行图像多阈值分割的实验。将实验结果与 BBO 算法、PSO 算法的图像分割结果进行比较,并与穷举算法的运行速度比较。使用 Cameraman.tif (256 × 256), Lena.png (512 × 512) 两幅图片作为示例说明。IBBO, BBO 和 PSO 3 种算法分别独立运行 30 次,穷举算法运行 1 次。所有实验均在主频 3.10 GHz 的 CPU 和内存为 4 GB 的 PC 上进行,采用 Matlab R2014a 编程语言实现。为了公平起见, BBO 算法、PSO 算法和 IBBO 算法的种群大小  $N$  都设为 50, 最大迭代次数  $M$  也设置相同,为了满足不同阈值选择的需要,  $M$  随着阈值的增加而增加,表示为  $M = 3 \times 2^D + (D - 1) \times 20$ 。PSO 算法参数设置如下:学习因子  $c_1$  和  $c_2$  均为 2.1, 惯性权重为 0.6299。BBO 和 IBBO 算法的参数设置相同:初始变异率为 0.4, 最大迁入率( $I$ )为 1, 最大迁出率( $E$ )为 1。

用 4 种算法分别对图像进行 2、3、4、5 阈值搜索,搜索所得的最优阈值(OT)向量及其运行时间(time)如表 1 所示,其中, time 指算法独立运行 30 次的平均时间,  $T$  指阈值数, Exh 指穷举算法,  $V_{OT}$  是指 30 次运行中使最大熵值最大的阈值向量。表 2 列出了 3 种算法

运行 30 次获得最优值的均值(mean)、最小值(min), 以及成功率( $S_R$ ),  $S_R$  是寻得最优解的次数与算法运行总次数之比, 其中最优解的次数为获得的最大熵值等于穷举算法获得的最大熵值的次数。表 1 和表 2 中加黑的值均为较优的值。图 1、图 2 分别显示 Cameraman 和 Lena 的原始图像及其用 IBBO 算法分割后的图像。

表 1 最优阈值及运行时间

Table 1 Optimal thresholds and running time

算法 $T$	Cameraman		Lena		
	$V_{OT}$	time/s	$V_{OT}$	time/s	
Exh	2	127.92	0.303 1	79,149	0.301 7
	3	43,103,192	22.370 9	59,108,159	22.169 2
	4	43,96,145,196	1 397.973 5	55,99,143,181	1 344.268 4
	5	24,61,99,145,196	61 014.199 4	43,79,114,149,184	59 432.173 6
BBO	2	127.92	0.040 3	79,149	0.044 1
	3	42,98,191	0.083 9	55,105,159	0.089 5
	4	38,92,141,194	0.146 6	52,100,142,182	0.151 9
	5	27,63,99,146,194	0.252 9	51,85,117,153,188	0.255 4
IBBO	2	127.92	<b>0.037 3</b>	79,149	<b>0.040 1</b>
	3	43,103,192	0.077 5	59,108,159	<b>0.079 2</b>
	4	43,96,145,196	0.134 7	55,99,143,181	<b>0.133 6</b>
	5	24,61,99,145,196	<b>0.232 4</b>	43,79,114,149,184	<b>0.227 7</b>
PSO	2	128.93	0.052 0	79,149	0.057 4
	3	43,103,192	0.097 6	59,108,159	0.102 5
	4	43,96,145,196	0.156 0	55,99,143,181	0.157 6
	5	24,61,99,145,196	0.251 7	43,79,114,149,184	0.255 2

表 2 3 种算法获得的最大熵均值、最小值和成功率

Table 2 Mean value, min value of Shannon entropy and the success rate of three algorithms

图像	$T$	算法	mean	min	$S_R/\%$
Cameraman (256 × 256)	2	BBO	12.168 3	12.166 0	47
		IBBO	12.168 8	12.168 8	<b>100</b>
		PSO	12.168 7	12.167 8	97
	3	BBO	15.223 6	15.212 4	0
		IBBO	15.227 4	15.227 4	<b>100</b>
		PSO	15.221 4	15.152 9	90
	4	BBO	18.386 2	18.364 6	0
		IBBO	18.395 5	18.395 5	<b>100</b>
		PSO	18.395 5	18.395 5	<b>100</b>
	5	BBO	21.109 9	21.060 2	0
		IBBO	21.146 6	21.146 6	<b>100</b>
		PSO	21.125 1	21.047 0	80
Lena (512 × 512)	2	BBO	12.698 7	12.697 0	35
		IBBO	12.699 0	12.699 0	<b>100</b>
		PSO	12.699 0	12.699 0	<b>100</b>
	3	BBO	15.762 7	15.755 8	0
		IBBO	15.765 8	15.765 8	<b>100</b>
		PSO	15.765 8	15.765 8	<b>100</b>
	4	BBO	18.572 0	18.530 4	0
		IBBO	18.587 5	18.587 5	<b>100</b>
		PSO	18.587 5	18.587 5	<b>100</b>
	5	BBO	21.214 7	21.192 7	0
		IBBO	21.240 5	21.240 5	<b>100</b>
		PSO	21.240 4	21.240 0	96.66

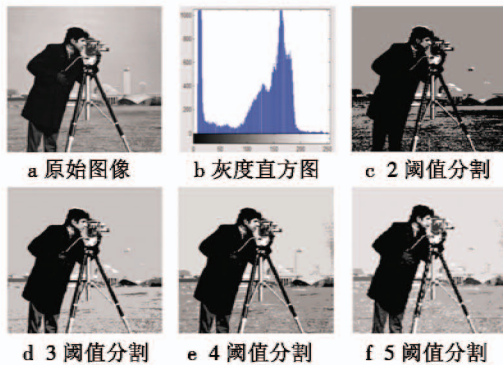


图 1 Cameraman 用 IBBO 算法分割

Fig. 1 Segmented Cameraman image by IBBO algorithm

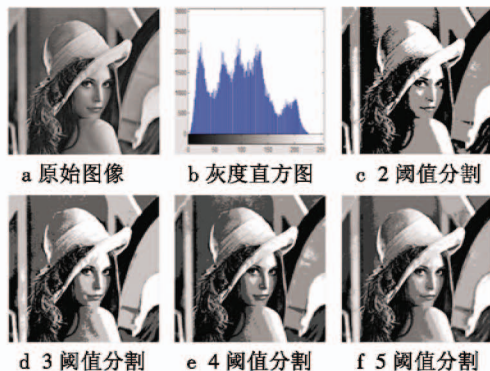


图 2 Lena 用 IBBO 算法分割

Fig. 2 Segmented Lena by IBBO algorithm

从表 1 可以看出,IBBO 算法运行速度最快,其次是 BBO 算法,再次是 PSO 算法,穷举搜索算法运行最慢;并且随着阈值数的增加,4 种算法运行的时间增加,这是因为随着阈值数的增加,寻优难度增加,需要的迭代次数增加。另外,因为穷举搜索算法的计算复杂度为  $O(L^D)$ ,所以耗费时间很长,但其得到的分割阈值一定是准确的。相比于 BBO 算法,IBBO 算法更能够准确寻到最优阈值,这说明算法的改进是有效的。与 PSO 算法比,IBBO 算法运行的时间少,这是由于在 PSO 算法中解向量的适宜度值计算及其比较通过循环语句实现;而在 IBBO 算法中先对解向量集中处理,再通过矩阵运算比较适宜度值;在 Matlab 运算环境中矩阵的运算速度快于基于循环的单个运算,所以 IBBO 算法的运算速度快于 PSO 算法。

从表 2 可以看出,对于图像 Cameraman,IBBO 算法在图像的 2、3、4、5 阈值分割中都能够准确地取得最优阈值,并且成功率都能够达到 100%。其次是 PSO 算法,虽然能够获得最优解,但不是每一次都能成功获得最优解。BBO 算法成功率非常低,在 2 阈值分割中成功率仅为 47%,在 3、4、5 阈值分割中成功率甚至为 0。这说明 BBO 算法的优化性能性远远劣于 IBBO 算法。对于图像 Lena,IBBO 算法和 PSO 算法对 2、3、4 阈值分

割的成功率都达到 100%,而对 5 阈值分割,PSO 算法成功率约为 96.66%,这说明 PSO 算法在阈值数较大的情况下,搜索能力下降,劣于 IBBO 算法。另外,从均值和最小值(见表 2 的 4 和 5 两行)看出,IBBO 算法优于 PSO 算法和 BBO 算法。

综上所述,基于改进的生物地理学算法寻优速度远远快于穷举算法,且快于 BBO 算法和 PSO 算法,稳定性好,应用于图像分割中能够得到较好的分割效果。这说明本文所做的改进是有效的。

## 5 总结

本文提出了一种改进的生物地理学算法,采用多源迁移算子、动态变异算子和不同的精英选择算子,增强了算法的全局搜索能力,提高了优化性能。并将改进的 BBO 算法应用于图像多阈值分割中,实验结果表明,基于改进的生物地理学算法的图像多阈值分割法分割速度快、稳定性好,能够取得较好的分割效果。

## 参考文献

- [1] SIMON D. Biogeography-based optimization [J]. IEEE Transaction on Evolutionary Computation, 2008, 12(6): 702-713.
- [2] GONG W Y, CAI Z H, LING C X, et al. A real-coded biogeography-based optimization with mutation [J]. Applied Mathematics and Computation, 2010, 216(9): 2749-2758.
- [3] 郑肇葆. 生物地理学优化(BBO)在图像分割中的应用 [J]. 武汉大学学报: 信息科学版, 2011, 36(8): 932-935. (ZHENG Z B. Application of biogeography-based optimization to image segmentation [J]. Geomatics and Information Science of Wuhan University, 2011, 36(8): 932-935.)
- [4] XIONG G J, SHIN D Y, DUAN X Z. Enhancing the performance of biogeography-based optimization using phylogenetic migration operator and orthogonal learning [J]. Computers & Operations Research, 2014, 41(1): 125-139.
- [5] LI X, WANG J, ZHOU J, et al. A perturb biogeography based optimization with mutation for global numerical optimization [J]. Applied Mathematics and Computation, 2011, 218(2): 598-609.
- [6] 陈基漓. 基于高斯变异的生物地理学优化模型 [J]. 计算机仿真, 2013, 30(7): 292-295. (CHEN J L. Biogeography-based optimization model based on Gaussian mutation [J]. Computer Simulation, 2013, 30(7): 292-295.)
- [7] MA H P, SIMON D. Blended biogeography-based optimization for constrained optimization [J]. Engineering Applications of Artificial Intelligence, 2011, 24(3): 517-525.

(下转第 58 页)