

用于指数熵多阈值分割的改进细菌觅食算法

张新明^{1,2} 涂 强¹ 刘 艳¹

(河南师范大学计算机与信息工程学院 新乡 453007)¹

(河南省高校计算智能与数据挖掘工程技术研究中心 新乡 453007)²

摘 要 针对图像多阈值分割中阈值搜索是有序正整数规划的特点,提出了一种用于指数熵多阈值分割的改进细菌觅食优化(Improved Bacterial Foraging Optimization, IBFO)算法。首先,将标准的细菌觅食优化(Standard Bacterial Foraging Optimization, SBFO)算法的趋化算子改成动态趋化算子以增强趋化操作的自适应性;然后,将 SBFO 中的迁徙算子替换成混合随机和动态的迁徙算子,将迁徙过程划分为两个阶段,第一阶段为随机迁徙,目的是增强全局搜索能力,第二阶段为动态局部迁徙,目的是提高局部搜索能力;随后,丢弃 SBFO 中的感应机制以便加快运行速度;最后,将 IBFO 算法进一步修改以满足有序正整数规划的要求,并将其应用于指数熵多阈值分割方法中。图像分割实验结果表明,与 SBFO, MBFO 和 IPSO 算法相比,提出的 IBFO 方法不仅优化效果更好,而且运行速度更快。

关键词 图像分割,多阈值分割,细菌觅食算法,指数熵

中图分类号 TP181, TP391. 41 文献标识码 A DOI 10. 11896/j. issn. 1002-137X. 2016. 7. 015

Improved Bacterial Foraging Optimization Algorithm Used for Multi-level Threshold Segmentation Based on Exponent Entropy

ZHANG Xin-ming^{1,2} TU Qiang¹ LIU Yan¹

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

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

Abstract In view of the characteristics of ordered positive integer programming of multi-level segmentation methods, an improved bacterial foraging optimization (IBFO) algorithm used for multi-level threshold segmentation based on exponent entropy was proposed in this paper. Firstly, a chemotactic step mechanism of the standard bacterial foraging optimization (SBFO) algorithm is changed into a dynamic chemotactic step approach to improve self-adaptation. Secondly, the original elimination-dispersal operator is replaced with a new one based on combining random mutation and dynamical local mutation, and the random mutation is used in the first phase to enhance the global search ability and the dynamical mutation is used in the second phase to improve local search performance. Thirdly, the communication mechanism of SBFO is abandoned to accelerate the running speed of the algorithm. Finally, IBFO is further modified to fit for the multi-level threshold segmentation based on exponent entropy. Experimental results show that the proposed method has better optimization performance with less computation time compared to SBFO, MBFO and IPSO.

Keywords Image segmentation, Multi-level threshold segmentation, Bacterial foraging optimization algorithm, Exponent entropy

1 引言

图像分割是低级图像处理向高级图像处理转换的关键步骤,一直是图像处理、机器视觉等领域的研究热点^[1]。阈值分割法因简单有效、计算量小和性能稳定而成为图像分割中最重要和应用最广泛的分割技术,其关键是阈值的选取,以便获得最优分割效果。最具代表性的阈值法有最大类间方差法^[2]、最大熵法^[3]、最小误差法^[4]、矩保持法^[5]等,这些方法一定程度上改善了分割效果。多阈值分割法是基于图像的一维灰度直方图来确定最优阈值的方法之一,它适用于多目标分割(对应于一维直方图上的多峰)。但到目前为止,对单阈值

分割的研究较多,而在实际应用中常常遇到的是多阈值分割。而多阈值分割问题是一种有序正整数规划的问题,也是一个 NP 难问题。虽然它可转化为一系列单阈值分割问题来解决,但这需要在全灰度范围内搜索一个最佳阈值组合,耗时较多,因此如果采用穷举搜索方法则难以应用于实际。为了解决多阈值分割中搜索阈值时计算复杂度高的问题,国内外学者进行了大量的研究,提出了许多基于智能优化算法的多阈值分割方法,如文献^[6]提出了一种基于改进粒子群优化算法的 OTSU 多阈值图像分割方法,文献^[7]提出了一种基于布谷鸟搜索算法和 Tsallis 熵的多阈值分割方法等。这些方法由于采用智能优化算法极大地降低了多阈值分割的计算复杂

到稿日期:2015-06-21 返修日期:2015-08-30 本文受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)资助。

张新明(1963-),男,教授,硕士生导师,CCF 会员,主要研究方向为智能优化算法、数字图像处理和模式识别等, E-mail: xinmingzhang@126.com;涂 强(1995-),男,硕士生,主要研究方向为数字图像处理和智能优化算法;刘 艳(1983-),女,硕士生,实验师,主要研究方向为图像处理。

度,使得多阈值分割方法易于应用到实际的分割问题中。文献[8]提出了一种基于标准的细菌觅食优化(Standard Bacterial Foraging Optimization, SBFO)算法的多阈值分割方法,文献[9]将 SBFO 中的感应机制加以修改,提出了一种基于修改细菌觅食优化(Modified Bacterial Foraging Optimization, MBFO)算法的最大熵多阈值分割方法,它将整个群体与当前细菌的影响值计算修改为群体中的最优细菌与当前细菌的影响值计算,降低了计算复杂度,也提高了算法的优化性能。但这种 MBFO 算法在寻找最优阈值时,仍然存在优化效果不佳、运行速度慢的问题。因此,本文针对多阈值分割中阈值搜索的特点,在 SBFO 的基础上,提出了一种改进的细菌觅食优化(Improved Bacterial Foraging Optimization, IBFO)算法,首先,将 SBFO 算法的趋化算子改成动态趋化算子以增强算法搜索的自适应能力,在搜索的初期,步长较大,有利于全局搜索,在搜索的后期,步长较小,有利于局部搜索;然后,将 SBFO 的迁徙算子,改成混合随机和动态的迁徙算子,使得算法的初始阶段为随机迁徙以增强全局搜索能力,后期阶段为动态局部搜索以提高收敛速度;随后,丢弃 SBFO 中的感应机制以便提高运行速度;最后,将 IBFO 算法进行进一步修改以适合有序正整数规划的特点,最终获得一种适用于多阈值分割的高效细菌觅食算法。

2 改进的细菌觅食优化算法

SBFO 算法^[10,11]因具有群体智能算法的并行搜索、易跳出局部极值等优点,被广泛用于许多领域,已经成为生物启发式计算研究领域的又一热点^[12,13]。该算法主要依靠细菌特有的趋化、繁殖、迁徙 3 种行为构建的 3 种算子以及群体感应机制,对最优解进行搜索,进而实现种群的进化。SBFO 算法虽然有许多优点,但仍存在着全局搜索能力不强和计算复杂度高的缺点。本文主要针对其缺点做了 4 点改进。

(1)正如文献[14]指出,不含有群体感应机制的 BFO 算法在收敛速度和精度上反而优于有群体感应机制的 BFO 算法,故本文直接去掉群体感应机制以便降低计算复杂度;

(2)将 SBFO 趋化操作中固定步长改进为动态缩进步长(见式(1)),以此构建动态趋化算子。这样随着趋化操作迭代的执行,逐步对趋化步长进行缩小,即搜索后期增强解的收敛性,搜索前期拓展细菌个体的寻优空间,以此提高算法的自适应性;

$$step = step_{max} - (step_{max} - step_{min}) \times l / N_{at} \quad (1)$$

其中, l 为当前迁徙算子执行次数, N_{at} 为其执行总次数, $step_{max}$ 为最大步长, $step_{min}$ 为最小步长。

(3)为了防止第 i 个细菌随机单位向量 $\Delta(i)$ 为零向量,添加去零机制,以避免无效的趋化操作,使算法更高效。

(4)用混合随机与动态调整迁徙算子替换原迁徙算子,其伪代码如下:

```

1. for i=1 to N do
2.   Select a variable vector  $v_i$  with a probability Ped
3.   if rand(0,1) < Ped then
4.     if  $l < N_{ed}/2$  then
5.        $v_i(g+1) = \text{round}(f_{min} + (f_{max} - f_{min}) \times \text{rand})$ 
6.     else
7.        $v_i(g+1) = v_i(g) + step \times \delta$ 
8.     end if
9.   end if
10. end for

```

其中, f_{min} 是变量的下界, f_{max} 是变量的上界, $rand$ 表示随机产生 0 到 1 之间的随机数, $round()$ 表示取整运算, δ 为二值随机算子,它的随机取值只能为 -1 或者 1。以上迁徙算子将搜索分成两个阶段,第一阶段为随机搜索,能够保证搜索初期种群的多样性,提高全局搜索能力;第二阶段为动态精准搜索,即搜索位于当前解附近,随着迭代次数的增加,搜索范围减少(见式(1)),保证在搜索的后期收敛加快。

通过以上改进能够大幅度提高 BFO 算法的优化效果。

3 基于 IBFO 算法的指数熵多阈值分割

3.1 指数熵多阈值分割

设一幅像素为 M 的图像的灰度级数为 L , 灰度 i 在图像中出现的概率为: $p_i = h_i / M, i = 0, 1, \dots, L-1$; 其中 h_i 是一维灰度直方图,表示 i 在图像中出现的频数。多阈值法就是基于一维直方图依据某种规则获得的一组阈值,然后利用这组阈值对图像进行分割,常常采用最大 Shannon 熵准则。虽然 Shannon 熵是解决不确定问题的有效方法,但是它存在一些不足。例如当 $p_i = 0$ 时, Shannon 熵无定义,以致在阈值法中要作一些特殊的处理。为了弥补这种不足,本文采用另外一种准则,即最大指数熵准则^[15]。基于最大指数熵的多阈值分割方法描述如下:设 D 个阈值 $[t_1, t_2, \dots, t_D]$ 组成一个阈值向量,将一维直方图分成 $D+1$ 个区域,则总的指数熵为:

$$E([t_1, t_2, \dots, t_D]) = E_1 + E_2 + \dots + E_{D+1} \quad (2)$$

$$\text{其中: } \omega_1 = \sum_{i=0}^{t_1} p_i, E_1 = \sum_{i=0}^{t_1} \frac{p_i}{\omega_1} e^{-p_i/\omega_1}, \omega_2 = \sum_{i=t_1+1}^{t_2} p_i, E_2 = \sum_{i=t_1+1}^{t_2} \frac{p_i}{\omega_2} e^{-p_i/\omega_2}, \omega_3 = \sum_{i=t_2+1}^{t_3} p_i, E_3 = \sum_{i=t_2+1}^{t_3} \frac{p_i}{\omega_3} e^{-p_i/\omega_3}, \dots, \omega_{D+1} = \sum_{i=t_D+1}^{L-1} p_i, E_{D+1} = \sum_{i=t_D+1}^{L-1} \frac{p_i}{\omega_{D+1}} e^{-p_i/\omega_{D+1}}.$$

选取的最佳阈值向量应满足:

$$[t_1, t_2, \dots, t_D]^* = \arg \max(E([t_1, t_2, \dots, t_D])) \quad (3)$$

在 D 维 $[1, L-2]$ 的整数空间中求一个满足式(3)的阈值向量,如采用穷举搜索,其计算复杂度为 $O(L^D)$, 计算量巨大。

3.2 IBFO 算法的指数熵多阈值分割

将 IBFO 算法运用到指数熵的多阈值分割中,还需要进行取整和排序等处理以符合有序正整数规划的特点。因此,将 IBFO 用于指数熵多阈值分割中的具体步骤如下。

Step 1 读入待分割的图像,若是彩色图像,首先将其转化为灰度图像,然后计算其一维灰度直方图;再计算其最大灰度级别 f_{max} , 最小灰度级别 f_{min} , 则: $0 \leq f_{min} < f_{max} \leq L-1$;

Step 2 设置参数,如种群数目 N , 最大迭代次数 Max_gen , 阈值个数 D , 趋化步骤数 N_c , 最大前进步长 $step_{max}$ 、最小前进步长 $step_{min}$, 最大前进步数 N_s 等的参数设置;

Step 3 初始化细菌位置,位置范围设定值为 $[f_{min}, f_{max}]$ 内,位置是灰度组合,即是一个正整数、无重复值和由小到大排序的阈值向量。设初始化的解分量为 $x_{i,d}$, 则:

$$x_{i,d} = f_{min} + rand \times (f_{max} - f_{min}), i \in \{1, 2, \dots, N\}, d \in \{1, 2, \dots, D\} \quad (4)$$

Step 4 将式(2)作为 IBFO 算法中细菌的适用度函数,用 E_{last} 保存每一个细菌的函数值;

Step 5 迁徙操作循环: $l = l + 1$;

Step 6 繁殖操作循环: $k = k + 1$;

Step 7 趋化操作循环: $j = j + 1$;

Step 7.1 令细菌 i 如下趋向一步;

Step 7.2 生成一个随机方向向量 $\Delta(i) \in Z^D$, 其每个元素 $\Delta_d(i)$ 都是 -1 或者 1 或者 0 的随机数, 但为了防止每个元素都为 0, 所以必须保证 $\Delta_d(i)$ 不全为 0, 即加上一个去 0 机制;

Step 7.3 移动: 令

$$x(j+1, k, l) = x(j, k, l) + \text{step} \times \Delta(i) / \sqrt{\Delta^T(i)\Delta(i)} \quad (5)$$

细菌沿着旋转后随机产生的方向游动一步长大小 step , 产生一个新解, 这里的 step 随着 l 的变化动态调整(见式(1));

Step 7.4 新解的边界检查、取整运算、去重复值以及由小到大排序, 其中, ①边界检查为: 如果 $x_{i,d} \leq f_{\min}$, 则 $x_{i,d} = f_{\min} + 1$; 如果 $x_{i,d} \geq f_{\max}$, 则 $x_{i,d} = f_{\max} - 1$; ②取整运算为: $x_{i,d} = \text{round}(x_{i,d})$; ③查重和排序: 由于在多阈值分割中, 阈值向量中的分量应满足如下条件: $f_{\min} < t_1 < t_2 < \dots < t_D < f_{\max}$, 所以每次新解产生后应去掉重复值, 即用式(4)来替代重复值并不断检查, 直到无重复值为止; 然后由小到大进行排序; 最后计算适应度函数值 $E(i, j+1, k, l)$;

Step 7.5 游动:

i) 令 $m=0$;

ii) 若 $m < N_s$, 令 $m=m+1$, 若 $E(i, j+1, k, l) > E_{\text{last}}(i)$, 令 $E_{\text{last}}(i) = E(i, j+1, k, l)$, 并采用式(5)产生新解, 采用 Step 7.4 的操作并转到步骤 ii), 否则, 令 $m = N_s$;

Step 7.6 返回到 Step 7.2, 处理下一个细菌 $i+1$;

Step 8 若 $j < N_c$, 返回 Step 7 进行趋化操作;

Step 9 进行繁殖操作: 用一半比较优的解替换一半比较差的解。若 $k < N_m$, 则返回 Step 6;

Step 10 用改进的迁徙算子进行迁徙操作。若 $l < N_{ad}$, 则返回 Step 5, 否则结束寻优, 转 Step 11;

Step 11 输出全局最优解即输出最优阈值向量 t^* ;

Step 12 根据阈值向量 t^* 对原图像进行分割处理, 得到分割后的图像。

4 仿真实验及结果分析

为了验证本文所提方法的有效性, 用本文方法进行图像阈值搜索的仿真实验。限于篇幅, 只选用了 Lena、Cameraman、Peppers 和 House 4 幅图片作为示例说明, 如图 1(a)~图 1(d) 所示, 其对应的直方图如图 1(e)~图 1(h) 所示。算法采用 MATLAB R2014A 语言实现, 所有实验都在 Intel (R) Core(TM) i7-3770 主频为 3.4GHz 的 CPU 和内存为 8GB 的 PC 机上进行。相比穷举搜索阈值算法, 使用智能优化算法能够大幅度提高搜索速度^[7], 而穷举搜索方法一定能获得准确的阈值向量, 但该方法尤其在四阈值及以上的搜索中, 搜索时间极长, 让人难以接受。受制于篇幅, 本实验只比较 4 种优化算法 (IBFO 算法、SBFO 算法^[8]、MBFO 算法^[9] 和改进的粒子群优化 (Improved Particle Swarm Optimization, IPSO) 算法^[6]) 的阈值搜索性能, 后面的 3 种算法都应用于多阈值分割中, 具有一定的可比性。为了公平起见, 在优化对比实验中, 3 种 BFO 优化算法的适应度函数评价次数应尽量保持相同, 而为了使 IPSO 算法获得可比拟的最优解, 在设置种群数量 N 时, 3 种 BFO 算法中 $N=20$, 而 IPSO 的 $N=40$, IPSO 的适应度函数评价次数是前 3 种算法的适应度函数评价次数的 1.5 倍左右 (因为 3 种 BFO 算法中的趋化操作需要评价多次适应度函数, 所以是接近 1.5 倍, 而不是 2 倍)。4 种算法的迭代

次数设置相同: $\text{Maxgen} = 2 \times D^2 + (D-1) \times 20 + 12$, 其中 D 表示适应度函数的维数, 即阈值数。这表明对于不同的阈值数搜索, 适应度函数评价次数不同, 随着阈值数的增加, 搜索次数增加, 也保证了在获得最优解的同时, 运行时间最少。IPSO 算法的其它参数设置同文献^[6], 对于 SBFO、MBFO 和本文提出的 IBFO, 其它参数设置基本相同 (见表 1), 其中, “×”代表不设置此参数。

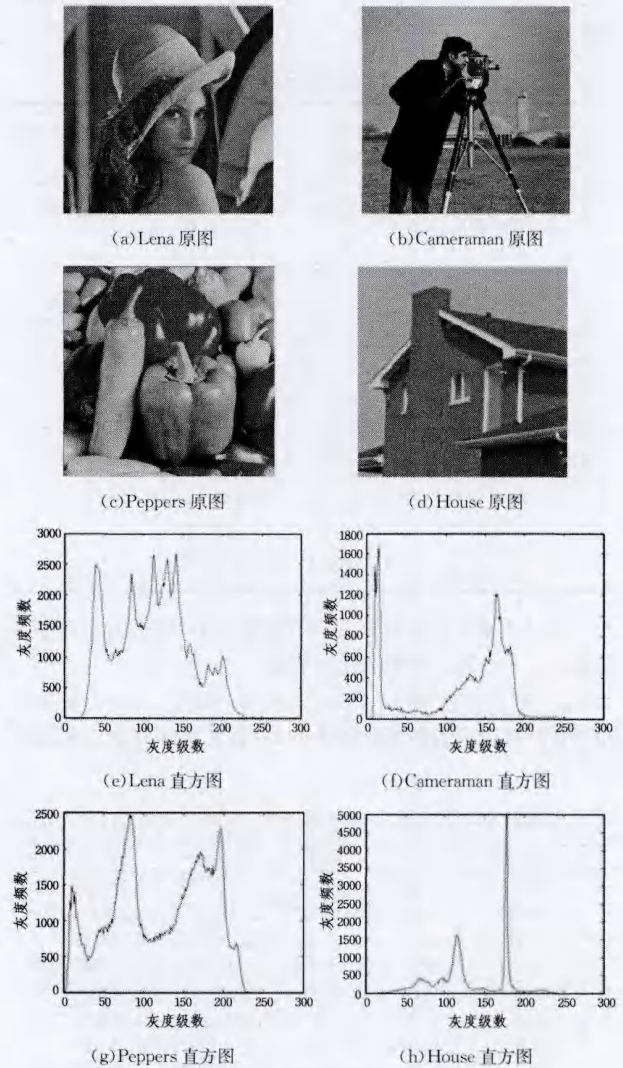


图 1 原图像和其直方图

表 1 3 种细菌觅食优化算法的参数设置

参数	SBFO	MBFO	IBFO
细菌种群数量: N	20	20	20
趋化算子执行次数: N_c	10	10	10
趋化最大前进次数: N_s	4	4	4
繁殖算子执行次数: N_{re}	1	1	1
迁徙算子执行次数: N_{ed}	Maxgen/N_c	Maxgen/N_c	Maxgen/N_c
执行迁徙算子概率: P_{ed}	0.3	0.3	0.3
吸引力的深度: $d_{\text{attractant}}$	0.1	0.1	×
吸引力的宽度: $w_{\text{attractant}}$	0.2	0.2	×
排斥力的高度: $h_{\text{repellant}}$	0.1	0.1	×
排斥力的宽度: $w_{\text{repellant}}$	10	10	×
最大移动步长: step_{max}	15	15	7
最小移动步长: step_{min}	0.1	0.1	1

4.1 优化性能对比

为了评价 4 种优化算法的性能, 将 4 种优化算法用于 4 幅图像的指数熵多阈值搜索中, 并独立运行 30 次, 选取 30 次

中的最大值 Max(最好值)、平均值 Mean、最小值 Min(最差值)、算法的成功次数(Numbers of Success, NS)及运行时间(Time)为评价标准来考察这 4 种算法的寻优性能。其中,一种算法的成功次数是以 OptV 作为评价参考,即独立运行 30 次中该算法寻找的最优值等于 OptV 的次数;其中 OptV 为 4 种算法在同一幅图像和同一阈值数 D 上共运行 30×4 次中搜索获得的指数熵的最大值。OptV 及其对应的阈值向量如表 2 所列。

表 2 4 幅图像的最大指数熵和其对应的最优阈值组合

D	Cameraman		Lena	
	Optimal thresholds	OptV	Optimal thresholds	OptV
4	51,96, 145,196	13.163677	69,107, 145,181	13.211440
5	47,87,121, 154,196	15.736955	58,90,120, 150,183	15.765202
6	32,63,95, 125,156,200	18.297193	52,78,105, 132,161,188	18.300199
D	Peppers		House	
	Optimal thresholds	OptV	Optimal thresholds	OptV
4	35,80, 120,147	13.253757	56,81, 106,187	13.073209
5	36,65,100, 139,177	15.821292	56,81,106, 188,210	15.606159
6	35,64,96, 131,165,200	18.381263	40,60,84, 107,188,210	18.095830

为了凸显本文所提算法具有较强的寻优能力,仅提供 4 幅图像在 4,5 和 6 阈值的优化结果。

4 种优化算法用于 Lena、Cameraman、Peppers 和 House 4 幅图片的优化搜索结果分别见表 3—表 6,其中优者用黑体表示。

表 3 4 种优化算法对 Lena 图像的阈值搜索结果和其运行时间对比

D	Method	Time(s)	Mean	Max	Min	NS
4	IPSO	0.1746	13.211440	13.211440	13.211440	30
	SBFO	0.2211	13.211417	13.211440	13.211018	19
	MBFO	0.1647	13.211426	13.211440	13.211307	20
	IBFO	0.1449	13.211440	13.211440	13.211440	30
5	IPSO	0.2611	15.765119	15.765202	15.765025	16
	SBFO	0.3145	15.765098	15.765202	15.764840	7
	MBFO	0.2366	15.764379	15.765202	15.746543	4
	IBFO	0.2062	15.765176	15.765202	15.765025	25
6	IPSO	0.3664	18.298301	18.300199	18.297064	7
	SBFO	0.4129	18.298026	18.300199	18.296281	1
	MBFO	0.3145	18.299002	18.300199	18.296484	3
	IBFO	0.2769	18.299470	18.300199	18.297774	20

表 4 4 种优化算法对 Cameraman 图像的阈值搜索结果和其运行时间对比

D	Method	Time(s)	Mean	Max	Min	NS
4	IPSO	0.1730	13.163677	13.163677	13.163677	30
	SBFO	0.2186	13.163632	13.163677	13.163384	19
	MBFO	0.1632	13.163616	13.163677	13.163125	20
	IBFO	0.1417	13.163677	13.163677	13.163677	30
5	IPSO	0.2589	15.735431	15.736955	15.722284	23
	SBFO	0.3133	15.736274	15.736955	15.722073	7
	MBFO	0.2349	15.736741	15.736955	15.735805	9
	IBFO	0.2054	15.736955	15.736955	15.736955	30
6	IPSO	0.3673	18.296135	18.297193	18.287566	16
	SBFO	0.4162	18.296499	18.297193	18.294746	2
	MBFO	0.3141	18.296275	18.297193	18.294743	3
	IBFO	0.2753	18.296989	18.297193	18.295183	20

表 5 4 种优化算法对 Peppers 图像的阈值搜索结果和其运行时间对比

D	Method	Time(s)	Mean	Max	Min	NS
4	IPSO	0.1811	13.253757	13.253757	13.253757	30
	SBFO	0.2220	13.253737	13.253757	13.253652	17
	MBFO	0.1641	13.253740	13.253757	13.253595	18
	IBFO	0.1448	13.253757	13.253757	13.253757	30
5	IPSO	0.2590	15.819911	15.821292	15.816132	18
	SBFO	0.3118	15.821003	15.821292	15.816132	5
	MBFO	0.2353	15.821121	15.821292	15.817896	9
	IBFO	0.2054	15.821182	15.821292	15.817996	29
6	IPSO	0.3705	18.380632	18.381263	18.380183	10
	SBFO	0.4177	18.380635	18.381242	18.380064	0
	MBFO	0.3143	18.380860	18.381263	18.380108	2
	IBFO	0.2798	18.381200	18.381263	18.380338	26

表 6 4 种优化算法对 House 图像的阈值搜索结果和其运行时间对比

D	Method	Time(s)	Mean	Max	Min	NS
4	IPSO	0.1789	13.070793	13.073209	13.065214	19
	SBFO	0.2202	13.072624	13.073209	13.065214	17
	MBFO	0.1625	13.072832	13.073209	13.065214	13
	IBFO	0.1402	13.073209	13.073209	13.073209	30
5	IPSO	0.2638	15.575941	15.606159	15.505649	16
	SBFO	0.3235	15.603815	15.606159	15.569099	4
	MBFO	0.2431	15.606005	15.606159	15.605709	7
	IBFO	0.2041	15.606159	15.606159	15.606159	30
6	IPSO	0.3676	18.048926	18.095830	18.007716	9
	SBFO	0.4164	18.071541	18.095765	18.027563	0
	MBFO	0.3123	18.076317	18.095830	18.028457	2
	IBFO	0.2724	18.084696	18.095830	18.038584	21

另外,由于指数熵阈值分割法的阈值搜索寻优困难,表现在即使阈值不同,但最大熵值相差无几;为了区分,所有表中的 Mean、Max 等都采用 6 位小数计数法。限于篇幅,仅显示 Lena 图片在 3、4、5 和 6 阈值的分割结果,如图 2 所示。



图 2 Lena 图像多阈值分割的结果

从表 3—表 6 可以看出,在 4 阈值搜索中 IBFO 算法在 4 幅图片上,不管是 Mean、Max、Min 还是 NS 在 4 种算法中都有最佳的表现,成功次数都为 30,即成功率都为 100%,而 IPSO 算法在 Lena、Cameraman 和 Peppers 的成功率为 100%,但在 House 的成功率为 63%;在 5 阈值和 6 阈值搜索中,IBFO 算法虽然成功率没有完全达到 100%(这是因为随着阈值数的增加,寻优难度增大),但在 4 种算法中不管是 Mean、Min 还是 NS,IBFO 都是最好的,其中在 Peppers 和 House 的 6 阈值搜索中,SBFO 算法成功率为 0,而 IBFO 算法的成功次

数分别为 26 和 21,表现不俗,也大幅度优于 MBFO(2 和 2) 和 IPSO 算法(10 和 9),如表 5 和表 6 所列。

4.2 搜索速度对比

从运行时间看,相比其它 3 种算法,在 3 种阈值搜索中 IBFO 算法耗时最少,虽然从 4 阈值搜索到 6 阈值搜索,随着寻优难度加大,搜索时间增加,但 IBFO 算法的最大耗时不超过 0.3s,见表 3—表 6 中的第 3 列,这是因为 IBFO 算法去掉了感应机制和收敛速度快等。耗时最多的是 SBFO 算法,这是因为其感应机制有较高的计算复杂度且收敛速度慢;其次是 IPSO 算法,这是由于 IPSO 算法的目标评价次数是 3 种 BFO 算法的 1.5 倍,增加了运行时间;再其次是 MBFO 算法,这是因为该算法中仍然含有感应机制,只是对其进行了改进,相比 SBFO 算法,其计算复杂度有所降低,所以 MBFO 算法的耗时比 SBFO 算法少。

总之,不管从解的质量、收敛速度以及算法的运行速度,还是从成功率上看,本文提出的 IBFO 算法要优于 SBFO、MBFO 和 IPSO 算法,这说明本文提出的 IBFO 算法是有效的。

4.3 步长参数的讨论

在 IBFO 算法中,影响优化性能的参数较多,这些参数中大多数在目前 BFO 算法研究的文献中已作讨论。因为 Step 参数对优化性能的影响较大,而且本文对 Step 参数采用改进方案,所以本节仅对 Step 参数进行讨论。为了研究 Step 对算法性能的影响,将参数 Step 递增地取 5 种不同的值,分别为 Step=0.8, Step=1.0, Step=1.5, Step=2.0 和 Step=2.5,并与本文提出的动态调整方案(Dynamical Modulation, DM)进行比较。限于篇幅,仅仅使用 House 图像进行实验。在其它参数不变的情况下,独立运行 IBFO 算法 30 次得到 4、5、6 阈值的最大值 Max、平均值 Mean、最小值 Min、算法的成功次数 NS 以及目标函数评价次数 FEVs(见表 7)。从表 7 可以看出:在 3 种阈值搜索中,不管从获取的最优值、最差值和平均值方面来看,还是成功次数方面来看,DM 都获得了最好的效果。例如在 4、5 阈值搜索中,IBFO 算法在 Step 为 DM 时获得的成功次数都为 30,成功率为 100%,优于 Step=1.0, Step=1.5 和 Step=2.0,而 Step=0.8 和 Step=2.5 的优化效果最差。

表 7 参数 Step 对 House 图像阈值搜索结果影响的对比

D	Step	FEVs	Mean	Max	Min	NS
4	0.8	2895	13.070649	13.073209	13.013599	24
	1.0	3081	13.072676	13.073209	13.065214	28
	1.5	3056	13.072942	13.073209	13.065214	29
	2.0	2969	13.073205	13.073209	13.073097	29
	2.5	2811	13.072902	13.073209	13.065214	20
	DM	2607	13.073209	13.073209	13.073209	30
5	0.8	3775	15.593071	15.606159	15.505649	17
	1.0	4072	15.590634	15.606159	15.544582	20
	1.5	3998	15.599904	15.606159	15.505649	23
	2.0	3973	15.601813	15.606159	15.562797	23
	2.5	3821	15.604675	15.606159	15.562797	17
	DM	3531	15.606159	15.606159	15.606159	30
6	0.8	4478	18.053694	18.095830	17.971937	2
	1.0	4955	18.069668	18.095830	18.026829	7
	1.5	4932	18.069784	18.095830	18.028563	11
	2.0	4909	18.082830	18.095830	18.026829	12
	2.5	4763	18.067610	18.095830	18.028472	4
	DM	4438	18.084696	18.095830	18.038584	21

另外,DM 的 FEVs 最少,这是由于 Step 值递减,在前期 Step 值较大,大幅度向全局最优点靠近,后期 Step 值较小,进一步进行局部搜索,这样减少了趋化操作,所以减少了目标函数的评价次数;Step=1.0 的 FEVs 最多,这是因为 Step=1.0 在整个搜索过程中能够靠近最优解,但值较小,要获得最优解,必须增加趋化操作。Step=0.8 时 FEVs 也比较小,这是因为步长值太小,难以靠近最优解,所以趋化操作也减少。这说明采用动态调整方案不仅提高了优化性能,而且减少了目标函数的评价次数,减少了搜索时间,所以 Step 采用动态调整方案是可行的。

结束语 多阈值搜索问题是一种有序无重复性的正整数规划问题,也是一个 NP 难问题,寻找一种高效的智能优化算法解决多阈值分割中高计算复杂度的问题是研究的热点。本文针对多阈值分割中阈值搜索的特点,在 SBFO 的基础上,提出了一种 IBFO 算法。将 SBFO 算法的趋化算子改成动态趋化算子以增强算法搜索的自适应能力,将原迁徙算子改成混合随机和动态的迁徙算子以增强全局搜索能力和加快收敛速度,丢弃 SBFO 中的感应机制以提高运行速度,并将 IBFO 算法进一步修改以满足多阈值分割中阈值搜索的要求。实验结果验证了本文改进的有效性。但 IBFO 算法在高维多阈值搜索中的成功率还不能令人满意,因此进一步提高 IBFO 算法的搜索性能是下一步研究的方向之一。

参考文献

- [1] Hultmann Ayala H V, dos Santos F M, Mariani V C, et al. Image thresholding segmentation based on a novel beta differential evolution approach[J]. Expert Systems with Applications, 2015, 42(4): 2136-2142
- [2] Zhang Xin-ming, Sun Yin-jie, Zheng Yan-bin. Precise two-dimensional Otsu's image segmentation and its fast recursive realization [J]. Acta Electronica Sinica, 2011, 39(8): 1778-1784 (in Chinese)
张新明,孙印杰,郑延斌. 二维直方图准分的 Otsu 图像分割及快速实现[J]. 电子学报, 2011, 39(8): 1778-1784
- [3] Zhang Xin-ming, Zhang Ai-li, Zheng Yan-bin, et al. Improved two-dimensional maximum entropy image thresholding and its fast recursive realization [J]. Computer Science, 2011, 38(8): 278-283 (in Chinese)
张新明,张爱丽,郑延斌,等. 改进的最大熵阈值分割及其快速实现[J]. 计算机科学, 2011, 38(8): 278-283
- [4] Zhang Xin-ming, Li Zhen-yun, Sun Yin-jie. Fast image thresholding with minimum error based on 2-D oblique segmentation [J]. Electronics Optics & Control, 2012, 19(6): 8-12, 40 (in Chinese)
张新明,李振云,孙印杰. 快速二维直方图斜分最小误差的图像阈值分割[J]. 电光与控制, 2012, 19(6): 8-12, 40
- [5] Zhang Xin-ming, Li Shuang-qun, Zheng Yan-bin. 2-D Shannon entropy image thresholding method based on preserving-moment principle and its fast recursive implementation [J]. Computer Science, 2012, 39(1): 276-280 (in Chinese)
张新明,李双群,郑延斌. 矩不变调整的二维 Shannon 熵图像分割及其快速实现[J]. 计算机科学, 2012, 39(1): 276-280
- [6] Liu Shen-xiao, Wang Xue-chun, Chang Chao-wen. Otsu image segmentation method based on improved PSO algorithm [J].

Computer Science, 2013, 40(8): 293-295 (in Chinese)

刘申晓, 王学春, 常朝稳. 基于改进粒子群优化算法的 OTSU 图像分割方法[J]. 计算机科学, 2013, 40(8): 293-295

- [7] Agrawal S, Panda R, Bhuyan S, et al. Tsallis entropy based optimal multilevel thresholding using cuckoo search algorithm [J]. Swarm and Evolutionary Computation, 2011, 11(8): 16-30
- [8] Sathya P D, Kayalvizhi R. Optimal multilevel thresholding using bacterial foraging algorithm [J]. Expert Systems with Applications, 2011, 38(10): 15549-15564
- [9] Sathya P D, Kayalvizhi R. Modified bacterial foraging algorithm based multilevel thresholding for image segmentation [J]. Engineering Application of Artificial Intelligence, 2011, 24(1): 595-615
- [10] Passino K M. Biomimicry of bacterial foraging for distributed optimization and control [J]. IEEE Control Systems Magazine, 2002, 22(3): 52-67
- [11] Das S, Biswas A, Dasgupta S, et al. Bacterial foraging optimization algorithm: theoretical foundations, analysis, and applications [J]. Foundations of Computational Intelligence, 2009, 203: 23-55
- [12] Zhang Guo-yong, Wu Yong-gang, Tan Yu-xiang. Bacterial foraging optimization algorithm with quantum behavior [J]. Journal of Electronics & Information Technology, 2013, 35(3): 614-621 (in Chinese)

章国勇, 伍永刚, 谭宇翔. 一种具有量子行为的细菌觅食优化算法[J]. 电子与信息学报, 2013, 35(3): 614-621

- [13] Zhang Xin-ming, Yin Xin-xin, Feng Meng-qing. Adaptive bacterial foraging optimization algorithm based on dynamic Gaussian mutation and random one for high dimensional functions [J]. Computer Science, 2015, 42(6): 101-106 (in Chinese)
- 张新明, 尹欣欣, 冯梦清. 动态高斯变异和随机变异融合的自适应细菌觅食优化算法[J]. 计算机科学, 2015, 42(6): 101-106
- [14] Tang W J, Wu Q H, Saunders J R. Bacterial foraging algorithm for dynamic environment [C] // Proceedings of IEEE Conference on Evolutionary Computation. Canada, 2006: 4467-4473
- [15] Zhang Jin-kuang, Wu Yi-quan. Image thresholding based on 2-D oblique exponent entropy method and tent map chaotic particle swarm algorithm [J]. Signal Processing, 2010, 26(5): 703-708 (in Chinese)
- 张金矿, 吴一全. 基于 Tent 映射 CPSO 的二维斜分指数熵阈值分割[J]. 信号处理, 2010, 26(5): 703-708

(上接第 76 页)

张朝晖, 刘永霞, 雷倩. 基于 SC-AdaBoost 的图像目标检测[J]. 计算机科学, 2015, 42(7): 309-313

- [2] Felzenszwalb P F, Girshick R B, McAllester D, et al. Object detection with discriminatively trained part-based models [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010, 32(9): 1627-1645
- [3] Yan J, Lei Z, Wen L, et al. The fastest deformable part model for object detection [C] // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Columbus, USA: IEEE Press, 2014: 2497-2504
- [4] Dalal N, Triggs B. Histograms of oriented gradients for human detection [C] // Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. San Diego, USA: IEEE Press, 2005: 886-893
- [5] Felzenszwalb P F, Girshick R B, McAllester D. Cascade object detection with deformable part models [C] // Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. San Francisco, USA: IEEE Press, 2010: 2241-2248
- [6] Pedersoli M, Vedaldi A, Gonzalez J. A coarse-to-fine approach for fast deformable object detection [J]. Pattern Recognition, 2015, 48(5): 1844-1853
- [7] Dubout C, Fleuret F. Exact acceleration of linear object detectors [J]. Lecture Notes in Computer Science, 2012, 7574(1): 301-311
- [8] Ahmed E, Shakhnarovich G, Maji S. Knowing a good HOG filter when you see it: Efficient selection of filters for detection [M] // Computer Vision-ECCV 2014. Springer International Publishing, 2014
- [9] Alexe B, Deselaers T, Ferrari V. Measuring the objectness of image windows [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(11): 2189-2202
- [10] Huang Kai-qi, Ren Wei-qiang, Tan Tie-niu. A review on image object classification and detection [J]. Chinese Journal of Computers, 2014, 37(6): 1225-1240 (in Chinese)
- 黄凯奇, 任伟强, 谭铁牛. 图像物体分类与检测算法综述[J].

计算机学报, 2014, 37(6): 1225-1240

- [11] Van de Sande K E A, Uijlings J R R, Gevers T, et al. Segmentation as selective search for object recognition [C] // Proceedings of the IEEE Conference on Computer Vision. Barcelona, Spain: IEEE Press, 2011: 1879-1886
- [12] Uijlings J R R, van de Sande K E A, Gevers T, et al. Selective search for object recognition [J]. International Journal of Computer Vision, 2013, 104(2): 154-171
- [13] Cheng M M, Zhang Z, Lin W Y, et al. BING: Binarized normed gradients for objectness estimation at 300fps [C] // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Columbus, USA: IEEE Press, 2014: 3286-3293
- [14] Tang Y, Wang X, Dellandrea E, et al. Fusing generic objectness and deformable part-based models for weakly supervised object detection [C] // Proceedings of the IEEE International Conference on Image Processing. Paris, France: IEEE Press, 2014: 4072-4076
- [15] Everingham M, Van G L, Williams C K I, et al. The pascal visual object classes (voc) challenge [J]. International Journal of Computer Vision, 2010, 88(2): 303-338
- [16] Hosang J, Benenson R, Dollár P, et al. What makes for effective detection proposals? [J/OL]. <http://arxiv.org/pdf/1502.05082/pdf>
- [17] Song Xiu-rui, Wu Zhi-yong. Unsupervised detection of image object with any class [J]. Optics and Precision Engineering, 2014, 22(1): 160-168 (in Chinese)
- 宋修锐, 吴志勇. 图像通用目标的无监督检测[J]. 光学精密工程, 2014, 22(1): 160-168
- [18] Lin Yu-dong, He Hong-jie, Chen Fan, et al. A rigid object detection model based on geometric sparse representation of profile and its hierarchical detection algorithm [J]. Acta Automatica Sinica, 2015, 41(4): 843-853 (in Chinese)
- 林煜东, 和红杰, 陈帆, 等. 基于轮廓几何稀疏表示的刚性目标模型及其分级检测算法[J]. 自动化学报, 2015, 41(4): 843-853