

嵌入趋化算子的 PSO 算法及其在多阈值分割中的应用

张新明^{1,2} 涂 强¹ 尹欣欣¹ 冯梦清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 针对标准粒子群优化(Particle Swarm Optimization, PSO)算法在优化选择多阈值时易陷入局部最优等问题,提出一种嵌入趋化算子的 PSO 算法。首先针对标准 PSO 算法具有较强的全局寻优能力但局部搜索能力较差,而细菌觅食优化(Bacterial Foraging Optimization, BFO)算法具有较强的局部搜索能力的特点,将 BFO 算法中具有较强局部搜索能力的趋化算子嵌入到 PSO 算法中,形成一种嵌入趋化算子的 PSO 算法(Chemotaxis Operator embedded PSO, COPSO),以此优势互补,使 COPSO 算法既有较强的全局搜索能力,又有较强的局部搜索能力。最后将 COPSO 算法用于最大熵多阈值图像分割中,得到最佳阈值向量。实验结果表明:与标准的 PSO、BFO 和 GA 算法相比,该算法具有更好的优化效果和更短的寻优时间。

关键词 粒子群优化算法,细菌觅食优化算法,图像分割,多阈值分割

中图分类号 TP391.41 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.2.065

Chemotaxis Operator Embedded Particle Swarm Optimization Algorithm and its Application to Multilevel Thresholding

ZHANG Xin-ming^{1,2} TU Qiang¹ YIN Xin-xin¹ FENG Meng-qing¹

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

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

Abstract The standard particle swarm optimization (PSO) algorithm is easy to trap into local optimum when selecting the optimal thresholds in multilevel thresholding, so a novel PSO algorithm by embedding the chemotaxis operator was presented. The standard PSO algorithm often possesses the strong global search ability but poor local search ability, while the feature of bacterial foraging optimization (BFO) is just reverse. The BFO's chemotaxis operator with good local search ability is embedded into the PSO, and the chemotaxis operator embedded PSO (COPSO) algorithm is got. On the basis of complementary advantages, the COPSO has both good global search ability and local search ability. The optimal threshold vectors can be obtained by applying the COPSO algorithm to multilevel image thresholding based on maximum entropy. The experimental results demonstrate that the COPSO algorithm can get better optimization effect and shorter optimization time compared with standard PSO, BFO and GA.

Keywords Particle swarm optimization algorithm, Bacterial foraging optimization algorithm, Image segmentation, Multilevel thresholding segmentation

1 引言

图像分割是图像处理中的基本技术,是自动目标识别的关键步骤^[1]。阈值法因简单且性能稳定成为图像分割中的最基本方法。阈值法的关键在于寻找最优的阈值,根据图像中目标的个数,阈值法又可以分为单阈值法和多阈值法。单阈值分割是相对简单的分割方法,多阈值分割问题则可转化为一组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优门限组合,计算量庞大,耗时较多,难以实际应用。

为了降低计算复杂度,常使用各种智能优化算法来搜寻最优阈值^[2]。

粒子群算法是由 Eberhart 和 Kennedy 于 1995 年提出的一种基于群体智能方法的演化计算技术^[3,4]。它因具有原理简单、参数少、所需领域知识少等优点,在许多实际优化问题中取得了非常好的应用^[5],但是它也有着易陷入局部极小值点、搜索精度不高、后期收敛速度慢等缺陷^[6]。因此针对标准 PSO 算法存在的缺陷,国内外学者通过研究分析提出了许多的改进方法;文献^[7]提出了一种基于去个性化理论的 PSO

到稿日期:2015-02-16 返修日期:2015-06-27 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,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;尹欣欣(1990—),女,硕士生,主要研究方向为数字图像处理;冯梦清(1990—),女,硕士生,主要研究方向为数字图像处理和智能优化算法。

算法;文献[8]提出了一种考虑最优、次最优和交叉算子的 PSO 算法;文献[9]提出了一种混合型 PSO 算法。这些改进 PSO 算法的性能得到了不同程度的提升,但仍存在复杂寻优问题时局部搜索能力差和成功率低等问题。

本文首先将 BFO 算法中具有较强局部搜索能力的趋化算子嵌入到 PSO 算法中,构建趋化算子的 PSO 算法,以大大提高 PSO 算法局部搜索能力,使其具有更高的搜索精度;然后将 COPSO 算法引入到基于最大熵的多阈值分割方法中的多阈值优化选择,得到分割的最佳阈值向量。仿真实验表明,新算法不仅优化效果更佳,而且寻优时间更短。

2 嵌入趋化算子的 PSO 算法(COPSO)

2.1 标准粒子群优化算法

PSO 算法起源于对鸟群捕食行为的研究,鸟群找到食物最简单有效的策略就是搜索目前离食物最近的鸟的周围区域。受此启发,将 PSO 算法中的每一个粒子看作优化问题的一个可行解,并由目标函数为它确定一个适应值。把每个粒子在飞行过程中所经历的最优位置看作粒子本身所找的最优解,也叫个体极值(P_p)。把整个种群所经历的最优位置叫作整个群体目前所找到的最优解,即全局极值(P_g)。每个粒子通过上述两个极值来不断地更新自己,从而产生新一代群体。这样,粒子就将追随当前的最优粒子而行动,并经过迭代搜寻得到最优解。

设目标搜索空间为 D 维,粒子的种群规模为 N 。用 D 维向量 $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})^T$ 来表示第 i 个粒子的位置,用 $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})^T$ 来表示其速度,用 $P_{pi} = (p_{pi1}, p_{pi2}, \dots, p_{piD})^T$ 表示每个粒子经历过的最佳位置,用 $P_g = (p_{g1}, p_{g2}, \dots, p_{gD})^T$ 表示所有粒子经历过的最佳位置。则在每一次的迭代中,粒子速度和位置按式(1)和式(2)进行更新。

$$v_{id}^{k+1} = \omega v_{id}^k + c_1 \text{rand}_1^k (P_{pid}^k - x_{id}^k) + c_2 \text{rand}_2^k (P_{gd}^k - x_{id}^k) \quad (1)$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \quad (2)$$

其中, $i=1, 2, \dots, N$; $d=1, 2, \dots, D$; k 为当前迭代次数; ω 为非负惯性因子; c_1, c_2 是非负学习因子, rand_1 和 rand_2 是介于 $[0, 1]$ 之间的随机数。

2.2 改进的粒子群优化算法

BFO 算法是由 Kevin M. Passion 于 2002 年提出的一种新的智能仿生优化算法^[10],该算法模拟了大肠杆菌在人体肠道内搜寻食物的 3 种基本行为,即趋化、复制和驱散,使得细菌可以不断游向营养丰富的区域。BFO 算法以局部搜索能力见长,主要体现在趋化操作上。

趋化指细菌朝着营养丰富区域聚集的行为,主要模拟了大肠杆菌在觅食过程中的两种基本运动:翻转(tumble)和前进(swim)。翻转指菌体向任意方向游动单位步长;前进指当细菌完成一次翻转后,若细菌目前位置的营养分布函数值(函数适应度值)得到改善,将沿同一方向继续前进若干步,直至它的营养分布函数值不再改善,或达到预定的前进步数为止。

因为 BFO 算法中的趋化行为具有变方向搜索特性,在同一个方向前进若干步后,可以根据它的营养分布函数值的变化来决定是否继续沿该方向搜索,这样大大提高了细菌的局部搜索能力,使其具有较高的搜索精度。BFO 算法趋化算子

的流程如图 1 所示。

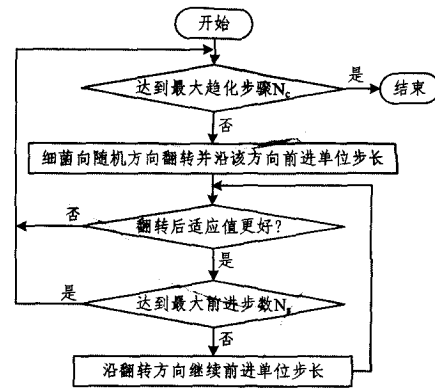


图 1 BFO 算法趋化算子的流程

为了弥补 PSO 算法局部搜索能力的不足,本文将 BFO 算法的趋化算子嵌入到 PSO 算法中。其基本思想为:先由 PSO 算法完成全局空间的搜索任务,保存个体和群体的最优信息,再将每一个粒子都看成细菌,由 BFO 算法的趋化算子完成局部搜索的任务,在趋化算子中,细菌需要完成翻转行为,即通过向随机方向前进一个单位步长来调整自身位置。细菌完成翻转的位置更新如式(3)所示:

$$x_{id}^{k+1} = x_{id}^k + C(i)\Delta(i) / \sqrt{\Delta^T(i)\Delta(i)} \quad (3)$$

其中, $C(i)$ 为第 i 个细菌的前进步长, $\Delta(i)$ 表示第 i 个细菌在随机方向上的一个单位向量。值得注意的是,对于前进步长而言,如果 $C(i)$ 过大,可能会使到达最优解附近的细菌跳过最优解;如果 $C(i)$ 过小,可能会使 BFO 算法陷入局部最优。标准的 BFO 算法采用的是固定步长,在解决某些简单的优化问题时较容易搜索到最优解,但对于复杂的优化问题,其全局的搜索能力就会明显下降。所以为了使细菌能够依照自身情况去改变自己前进的步长,本文采取一种利用线性动态函数作为细菌前进的动态步长的方法。细菌趋化算子前进动态步长函数如下:

$$C(i) = C_{\max} - (C_{\max} - C_{\min}) * k / \text{Maxgen} \quad (4)$$

其中, C_{\max} 表示预定义的最大前进步长, C_{\min} 表示预定义的最小前进步长,经多次试验发现, $C_{\max} = 4$ 、 $C_{\min} = 1$ 时,算法可取得最佳性能; k 表示当前迭代次数, Maxgen 表示最大迭代次数。这里细菌在开始时搜索较大区域,较快地定位最优解的大致位置,随着迭代次数的增大,细菌步长变小,开始精细的局部搜索。该方法加快了收敛速度,使得算法具有更高的搜索精度。

COPSO 算法的基本步骤如下:

Step 1 参数初始化,包括对种群规模 N 、最大迭代次数 Maxgen 、惯性权重因子 ω 、学习因子 c_1 和 c_2 、趋化步骤数 N_c 、最大前进步长 C_{\max} 、最小前进步长 C_{\min} 、最大前进步数 N_f 等基本参数的设置;

Step 2 粒子群位置 x_i 和速度 v_i 初始化;

Step 3 计算粒子群中每个粒子的适应度值,并初始化粒子群中的个体极值 P_{pi} 、全局极值 P_g ;

Step 4 按式(1)和式(2)对粒子的速度和位置进行更新;

Step 5 计算粒子群中每个粒子的适应度值,并对粒子当前位置适应度值、个体极值 P_{pi} 、全局极值 P_g 更新;

Step 6 按图 1 执行趋化操作算子;

Step 7 判断是否到达最大迭代次数,若是,则结束;否则,转到 Step 4。

COPSO 算法的流程如图 2 所示。

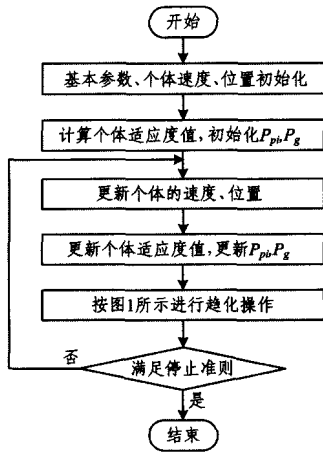


图 2 COPSO 算法流程

3 COPSO 应用于基于最大熵的图像分割中

最大熵阈值法^[11]是图像分割的常用方法,最先由 Kapur 等人提出。根据 Shannon 熵概念,对于 L 个灰度级在 $\{0, 1, 2, \dots, (L-1)\}$ 范围内变化的图像,其熵定义为:

$$H = - \sum_{i=0}^{L-1} p_i \ln p_i \quad (5)$$

其中, p_i 为第 i 个灰度出现的概率。在单阈值情况下,设阈值 t 将图像划分为目标 A 和背景 B 两类,则图像的熵 $H(t)$ 定义为 $H_A(t)$ 与 $H_B(t)$ 之和,即

$$H(t) = H_A(t) + H_B(t) = \sum_{i=0}^t \frac{p_i}{P_t} \ln \frac{p_i}{P_t} - \sum_{i=t+1}^{L-1} \frac{p_i}{1-P_t} \ln \frac{p_i}{1-P_t} \quad (6)$$

其中, $P_t = \sum_{i=0}^t p_i$, 使熵 $H(t)$ 取得最大值时的 t^* 就是分割目标与背景的最佳阈值。扩展到多阈值分割时,设有 D 个阈值,即把图像分割成 $D+1$ 个部分,此时最大熵算法的多阈值分割的目标函数定义如式(7)所示:

$$H([t_1, t_2, \dots, t_D]) = H_0 + H_1 + H_2 + \dots + H_D = - \sum_{i=0}^{t_1-1} \frac{p_i}{w_0(0, t_1)} \ln \frac{p_i}{w_0(0, t_1)} - \sum_{i=t_1}^{t_2-1} \frac{p_i}{w_1(t_1, t_2)} \ln \frac{p_i}{w_1(t_1, t_2)} - \dots - \sum_{i=t_{D-1}}^{t_D-1} \frac{p_i}{w_{D-1}(t_{D-1}, t_D)} \ln \frac{p_i}{w_{D-1}(t_{D-1}, t_D)} - \sum_{i=t_D}^{L-1} \frac{p_i}{w_D(t_D, L-1)} \ln \frac{p_i}{w_D(t_D, L-1)} \quad (7)$$

其中, $w_0(0, t_1) = \sum_{i=0}^{t_1-1} p_i$, $w_1(t_1, t_2) = \sum_{i=t_1}^{t_2-1} p_i$, \dots , $w_D(t_D, L-1) = \sum_{i=t_D}^{L-1} p_i$ 。

因为传统的最大熵算法获得阈值的方式是穷举计算方式,所以随着阈值个数的增加,算法的运算耗时会呈指数级增加。然而阈值的求解与准则函数有着直接的关系,因此可以考虑将阈值求解的问题转化为参数阈值 t 的优化问题,即

$$\varphi(t_1^*, t_2^*, \dots, t_D^*) = \arg \max_{(t_1, t_2, \dots, t_D)} (H[t_1, t_2, \dots, t_D]) \quad (8)$$

这样,基于最大熵的多阈值图像分割问题可以转化为求

解式(8)的最大值问题。为解决传统多阈值分割方法耗时长的的问题,本文采用 COPSO 算法优化目标函数以获得最优阈值向量。

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 $[0, 255]$ 个灰度级之间采用 COPSO 算法进行优化选择,使得式(8)取得最大值的一组解即为最优解。其基本步骤如下:

Step 1 读入将要分割的图像,若是彩色图像,则转化为灰度图像,并得到图像的灰度直方图;

Step 2 设置算法的参数,如种群数目 N 、最大迭代次数 $Maxgen$ 、阈值个数 D 、惯性因子 w 、学习因子 c_1 和 c_2 、趋化步数 N_c 、最大前进步数 N_s 等;

Step 3 初始化粒子群体,即初始化粒子的位置特征和速度特征,位置范围设定值为 $[0, 255]$,位置是灰度组合,即是一个取整数的、由小到大排序的阈值向量;

Step 4 将式(8)作为 COPSO 算法中粒子(细菌)的适用度函数,保存最大函数值的位置,不断迭代直到满足迭代停止条件,所得到的全局最优值即是优化得到图像的最优阈值向量 t^* ;

Step 5 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 实验结果与分析

为了检测 COPSO 算法的有效性,以式(8)作为适应度函数,将 COPSO 算法分别与标准的 PSO 算法、BFO 算法^[2] 以及 GA 算法的图像分割的结果作比较。限于篇幅,本文仅以灰度级为 256 的 Cameraman.tif (256 * 256) 和 Lena.png (512 * 512 * 3) 图像作为示例说明。为了公平起见,同时保证能使各自的算法优化效果最佳,且尽量达到稳定收敛,对各算法的参数设定如下:标准 PSO 算法和 GA 算法的种群规模都为 52;最大迭代次数(Maxgen)设置一样,且为了满足不同阈值选择的需要,Maxgen 将随着阈值个数的增加而增加,表示为 $Maxgen = 2 * D^2 + (D-1) * 20 + 20$,以达到自动调整 Maxgen 的目的;GA 算法其他参数设置:交叉概率为 0.65,变异概率为 0.05;COPSO 算法和 BFO 算法的种群规模均为 26,最大迭代次数为 $Maxgen/N_c$;BFO 算法其他参数设置:趋向性操作次数为 10,最大前进步数为 4,复制和驱散操作次数都为 2,细菌被驱散的概率为 0.25;COPSO 算法中的粒子群算法与标准 PSO 算法的参数设置相同:学习因子 c_1 和 c_2 均为 2.1,惯性因子 w 为 0.6299;COPSO 算法中的趋化算子参数的设置为:趋向性操作次数为 3,最大前进步数为 4,且采用自适应调整步长的方式。所有实验均在主频 3.10GHz 的 CPU 和内存为 4GB 的 PC 上进行,操作系统采用 Windows 7,并采用 MATLAB R2014a 编程语言实现。

用 4 种算法分别对图像进行 2、3、4、5 阈值分割,其中每个算法运行 30 次,同时以穷举(Exhaustive, Exh)算法运行 1 次求得的最佳分割阈值向量、运行时间以及最优解为对照标准,对 4 种算法的优化性能做出评价。表 1 列出了每种算法运行 30 次取得最优解中最好的一次所对应的阈值向量(T 指阈值数)及平均运行时间(Time)。表 2 列出了 4 种算法运行 30 次得到的最优解的均值(Mean)、最差值(Worst)以及算法的成功率(SR,即寻得和穷举算法所得最优解相同的次数与

算法运行总次数之比)。表 1 和表 2 中加黑的值均为较优的值。图 3 和图 4 分别显示 Cameraman 和 Lena 的原始图像及用 COPSO 算法得到的 3 阈值和 5 阈值分割的结果。

表 1 最优阈值及运行时间

image	T	Cameraman		Lena	
		Optimal thresholds	Time (s)	Optimal thresholds	Time (s)
Exh	2	127,192	0.3031	133,191	0.2499
	3	43,103,192	22.3709	113,158,201	13.1962
	4	43,96,145,196	1397.9735	107,139,170,204	635.3195
	5	24,61,99,145,196	61014.1994	107,138,168,198,226	24800.7432
GA	2	127,192	0.089999	133,191	0.089947
	3	40,103,192	0.16175	114,157,200	0.1622
	4	43,98,143,194	0.24668	107,136,170,204	0.25342
	5	25,65,98,138,193	0.34816	105,135,164,193,225	22.1692
PSO	2	127,192	0.07766	133,191	0.084067
	3	43,103,192	0.13558	113,158,201	0.13922
	4	43,96,145,196	0.20042	107,139,170,204	0.19554
	5	24,61,99,145,196	0.26017	107,138,168,198,226	0.25471
BFO	2	127,192	0.16922	133,191	0.16596
	3	43,103,192	0.24158	113,158,201	0.24292
	4	43,96,145,196	0.39146	107,139,170,204	0.39356
	5	24,61,199,145,196	0.54262	107,138,168,198,226	0.53792
COPSO	2	127,192	0.055589	133,191	0.05484
	3	43,103,192	0.095264	113,158,201	0.095712
	4	43,96,145,196	0.14545	107,139,170,204	0.14296
	5	24,61,99,145,196	0.19204	107,138,168,198,226	0.18681

表 2 4 种算法获得的最优解的均值、最差值和成功率

Image	T	Method	Mean	Worst	SR
Cameraman (256×256)	2	GA	12.1666	12.1598	3.33%
		PSO	12.1686	12.1678	86.7%
		BFO	12.1687	12.1685	80%
		COPSO	12.1688	12.1688	100%
		GA	15.2050	15.1593	0
	3	PSO	15.2244	15.1529	83.3%
		BFO	15.2259	15.2224	36.7%
		COPSO	15.2274	15.2274	100%
		GA	18.3427	18.2837	0
		PSO	18.3955	18.3955	100%
	4	BFO	18.3954	18.3950	80%
		COPSO	18.3955	18.3955	100%
		GA	21.0357	20.9623	0
		PSO	21.0935	21.0470	13.3%
		BFO	21.1445	21.1434	66.7%
5	COPSO	21.1446	21.1446	100%	
	GA	12.0710	12.0604	20%	
	PSO	12.0721	12.0721	100%	
	BFO	12.0720	12.0719	83.3%	
	COPSO	12.0721	12.0721	100%	
Lena (512×212)	2	GA	15.0155	14.9977	0
		PSO	15.0264	15.0264	100%
		BFO	15.0260	15.0252	16.7%
		COPSO	15.0264	15.0264	100%
		GA	17.6320	17.6001	0
	3	PSO	17.6656	17.6167	86.7%
		BFO	17.6731	17.6729	70%
		COPSO	17.6732	17.6732	100%
		GA	20.0825	19.9903	0
		PSO	20.1710	20.0041	96.7%
	4	BFO	20.1767	20.1758	53.3%
		COPSO	20.1768	20.1768	100%
		GA	20.0825	19.9903	0
		PSO	20.1710	20.0041	96.7%
		BFO	20.1767	20.1758	53.3%
5	COPSO	20.1768	20.1768	100%	

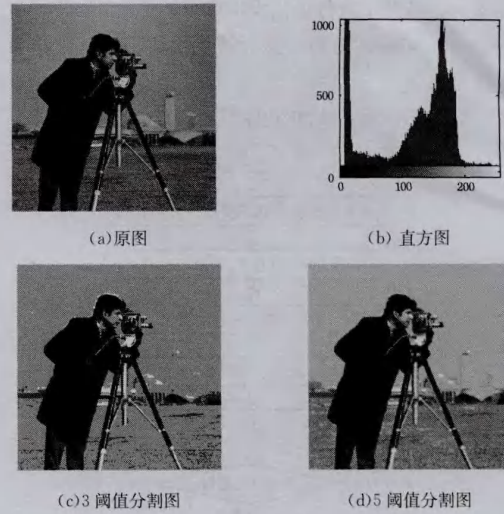


图 3 Cameraman 图像及其分割结果

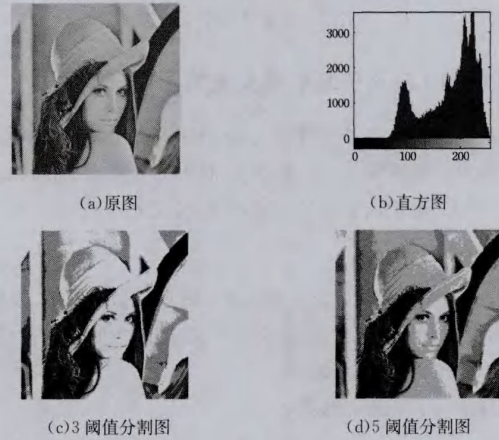


图 4 Lena 图像及其分割结果

从表 1 可以看出,GA 算法仅在 2 阈值分割时能准确收敛到穷举算法得到的最佳阈值,PSO 算法、BFO 算法以及 COPSO 算法在 4 种阈值选择中均能收敛到最优阈值上,但 COPSO 算法在运行速度上要明显快于其他算法,例如对 Lena 图像进行 3 阈值分割时,GA 算法耗时为 0.1622s,相对于穷举算法的 13.1962s,GA 算法将时间缩短了 81.36 倍;PSO 算法耗时为 0.1392s,相对于穷举算法时间缩短了 94.79 倍;BFO 算法耗时为 0.24292s,相对于穷举算法时间缩短了 54.32 倍;而 COPSO 算法耗时为 0.0957s,相对于穷举算法时间缩短了 137.87 倍。同时可以看出,COPSO 算法的运行速度比 PSO 算法或 BFO 算法都要快,这是因为相对于 PSO 算法而言,COPSO 算法由于具备 BFO 算法所有的较高的局部搜索能力,在种群数目初始化时使用较少的种群数目就能搜索到问题的最优解,减少了算法的循环次数,自然缩短了算法的运行时间;而 COPSO 算法相对于 BFO 算法而言,由于具备 PSO 算法较强的全局搜索能力,可以省去 BFO 算法的复制和驱散操作,大大降低了算法的复杂度,也使得算法的运行时间大幅缩短。

从表 2 可以看出,对于图像 Cameraman 和 Lena,COPSO 算法在图像的 2、3、4、5 阈值分割中获得的最大熵均值是 4 种算法中最大的,其最差值也是 4 种算法中最好的。再从成功率上分析,COPSO 算法在 4 种阈值选择中的成功率都能够达

到100%。PSO算法和BFO算法,其成功率随着阈值数的增加会出现明显波动,而GA算法在3、4、5阈值分割时其成功率为0,从对比中可以看出,COPSO算法有着更好的稳定性和准确性。

由此可见,COPSO算法充分发挥了BFO算法局部寻优能力强的优势,弥补了PSO算法局部搜索能力不足的缺陷,大大提高了PSO算法的优化性能。对于复杂的多阈值分割,无论是在全局收敛的可靠性上还是在收敛速度方面,COPSO算法都明显优于BFO算法和PSO算法这两个单一算法。这说明本文所做的改进是有效的。

结束语 针对标准粒子群算法在解决复杂的多维阈值选择优化问题时存在的局部搜索能力弱、易早熟且收敛速度慢等问题,提出了将细菌觅食优化算法中具有较强局部搜索能力的趋化算子嵌入到PSO算法的方法,大大提高了PSO算法的局部搜索能力,使其具有较高搜索精度的同时,还具有更快的寻优速度。基于COPSO算法的多阈值图像分割实验结果表明,COPSO算法的图像多阈值分割法运行速度快、稳定性好,能够取得较好的优化效果。

参考文献

- [1] Zhang Xin-ming, Xue Zhan-ao, Zheng Yan-bin. Fast and precise two-dimensional Renyi entropy image thresholding[J]. Pattern Recognition and Artificial Intelligence, 2012, 25(3): 411-418 (in Chinese)
张新明, 薛占熬, 郑延斌. 二维直方图准分的Renyi熵快速图像阈值分割[J]. 模式识别与人工智能, 2012, 25(3): 411-418
- [2] Sathya P D, Kayalvizhi R. Modified bacterial foraging algorithm based multilevel thresholding for image segmentation[J]. Engineering Applications of Artificial Intelligence, 2011, 24(4): 595-615
- [3] Eberhart R C, Kennedy J. A new optimizer using particle swarm theory[C]// Proceedings of the sixth International Symposium on Micro Machine and Human Science. 1995, 1: 39-43
- [4] 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
- [5] Barbieri R, Barbieri N, de Lima K F. Some applications of the PSO for optimization of acoustic filters[J]. Applied Acoustics, 2015, 89: 62-70
- [6] Liu Y, Niu B, Luo Y. Hybrid learning particle swarm optimizer with genetic disturbance[J]. Neurocomputing, 2015, 151: 1237-1247
- [7] Yu Fei, Li Yuan-xiang, Wei Bo, et al. Particle swarm optimization based on deindividuation theory[J]. Control and Decision, 2013, 28(10): 1520-1524 (in Chinese)
喻飞, 李元香, 魏波, 等. 一种基于去个性化理论的粒子群算法[J]. 控制与决策, 2013, 28(10): 1520-1524
- [8] He G, Huang N J. A new particle swarm optimization algorithm with an application[J]. Applied Mathematics and Computation, 2014, 232: 521-528
- [9] Zhang Hui-yun, Huang Xiao-wei, Zhang Hong-hua, et al. Study on hybrid particle swarm optimization algorithms[J]. Application Research of Computers, 2011, 28(5): 1631-1633 (in Chinese)
章慧云, 黄晓伟, 张红华, 等. 混合型粒子群优化算法研究[J]. 计算机应用研究, 2011, 28(5): 1631-1633
- [10] Passino K M. Biomimicry of bacterial foraging for distributed optimization and control[J]. Control Systems IEEE, 2002, 22(3): 52-67
- [11] 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
- [11] Li Ran, Gang Zong-liang, Zhu Xiu-chang. A Global Reconstruction Model of Images Using Block Compressed Sensing [J]. Signal Processing, 2012, 28(10): 1416-1422 (in Chinese)
李然, 干宗良, 朱秀昌. 基于分块压缩感知的图像全局重构模型[J]. 信号处理, 2012, 28(10): 1416-1422
- [12] Wang Shang-li. Research on image recovery method based on compressed sensing [D]. Xi'an: Xidian University, 2012 (in Chinese)
王尚礼. 压缩感知图像重建算法研究[D]. 西安: 西安电子科技大学, 2012
- [13] Gan L. Block compressed sensing of natural images [C]// International Conference on Digital Signal Processing. Cardiff, 2007: 403-406
- [14] Rudin L I, Osher S, Fatemi E. Nonlinear total variation based noise removal algorithms [J]. Physica D, 1992, 60: 259-268
- [15] Z Hai-bo, Zhu Xiu-chang. Sampling adaptive block compressed sensing reconstruction algorithm for images based on edge detection [J]. Journal of China Universities of Posts and Telecommunications, 2013, 20(3): 97-103
- [16] 中国百科网. 详解 ISO12233 Chart(分辨率测试标板)使用方法 [EB/OL]. <http://www.chinabaike.com/t/35899/2013/0802/1357958.html>

(上接第310页)

- [5] Tropp J, Gilbert A. Signal Recovery from Random Measurements via Orthogonal Matching Pursuit [J]. IEEE Trans. on Information Theory, 2007, 53(12): 4655-4666
- [6] Mun S, Fowler E. Block compressed sensing of images using directional transforms [C]// Proc. IEEE Intern. Conf. on Image Processing. USA, 2009: 3021-3024
- [7] Wang Rong-fang, Jiao Li-cheng, Liu Fang, et al. Block-based adaptive compressed sensing of image using texture information [J]. Acta Electronica Sinica, 2013, 41(38): 1506-1514 (in Chinese)
王荣芳, 焦李成, 刘芳, 等. 利用纹理信息的图像分块自适应压缩感知 [J]. 电子学报, 2013, 41(38): 1506-1514
- [8] Khanh Q D, Shim H B J. Deblocking filter for artifact reduction in distributed compressive video sensing [C]// Visual Communications and Image Processing (VCIP). 2012: 1-5
- [9] Mun S, Fowler J E. Residual reconstruction for block-based compressed sensing of video [C]// Proc. of Data Compression Conf. . USA, 2011: 183-192
- [10] Candes E, Romberg J, Tao T. Stable signal recovery from incomplete and inaccurate measurements [C]// Comm. on Pure and Applied Mathematics. 2006: 1207-1223