

文章编号:1000-582X(2012)02-149-06

## 差分变异本质粒子群的模糊熵图像分割

张 伟

(青岛科技大学 自动化与电子工程学院, 山东 青岛 2660422)

**摘 要:**基本本质粒子群算法存在易陷入局部最优以及过早收敛的缺点。在基本本质粒子群算法的基础上,借鉴差分进化中利用差分量对种群进行变异操作的思想,提出了差分变异本质粒子群优化算法。结合图像模糊熵,得到了基于差分变异粒子群优化的模糊熵图像分割算法。算法利用差分变异本质粒子群来搜索使图像模糊熵最大的参数值,得到分割阈值对图像进行分割。通过与其它两种本质粒子群算法的分割结果比较表明该算法取得了令人满意的分割结果,算法运算时间很小,能够满足对煤尘浓度实时精确测量的要求。

**关键词:**本质粒子群;差分变异;模糊熵;图像分割;煤尘图像

**中图分类号:**TP 391.41

**文献标志码:**A

## Image segmentation based on differential mutation bare bones particle swarm optimization and fuzzy entropy

ZHANG Wei

(School of Automation and Electronic Engineering, Qingdao University of Science and Technology, Qingdao 266042, Shandong, P. R. China)

**Abstract:** Basic bare bones particle swarm optimization (BBPSO) is easy to get stuck into local optima. Based on basic BBPSO, using the idea of mutation in differential evolution, a new algorithm named differential mutation bare bones particle swarm optimization (DMBBPSO) is proposed and combined with image fuzzy entropy to obtain a new segmentation algorithm based on DMBBPSO and fuzzy entropy for image segmentation. The proposed algorithm uses DMEBBPSO to explore fuzzy parameters of maximum fuzzy entropy and gets the image segmentation threshold. According to the experiment results, compared with other two algorithms, the proposed algorithm shows better segmentation performance and very low time cost. It can be used to real time and precision measure coal dust image.

**Key words:** bare bones particle swarm optimization; differential mutation; fuzzy entropy; image segmentation; coal dust image

对煤尘图像进行有效的分割是煤尘浓度测量重要研究内容之一,阈值法是最常用的分割方法,阈值法的关键是阈值的选取。用图像灰度模糊熵来确定分割阈值是一种有效的阈值确定方法<sup>[1]</sup>。图像灰度

模糊熵中模糊参数的寻优实际上是一个优化问题。解决优化问题的方法通常有穷举法、遗传算法、蚁群算法、粒子群算法等,其中 Kennedy 和 Eberhart 于 1995 年提出粒子群优化算法由于其优越性成为研

收稿日期:2009-12-04

基金项目:山东省自然科学基金资助项目(Y2008G14)

作者简介:张伟(1975-),男,青岛科技大学博士,主要从事数字图像处理方向的研究,(Tel)13646482801;  
(E-mail)zhangwei@qust.edu.cn。

究热点<sup>[2-6]</sup>。

2003年, Kennedy<sup>[7]</sup>提出了本质粒子群优化算法(BBPSO), 相比普通 PSO, BBPSO 算法更为简单有效。基本 BBPSO 优化算法和普通 PSO 一样, 存在易陷入局部最优以及过早收敛的缺点, 使得该算法难以得到理想的优化效果。对 BBPSO 的改进可以借鉴 PSO 改进算法的思想, 主要的改进 PSO 算法有对惯性因子的改进<sup>[8-11]</sup>, 以及引入遗传算法中的交叉、变异或进化思想对部分粒子进行相应的操作<sup>[12-15]</sup>。借鉴交叉、变异思想, Renato<sup>[16]</sup>在基本 BBPSO 的基础上引入了 Gaussian 和 Cauchy 变异, 取得了比基本 BBPSO 更好的优化结果。差分进化(DE)算法中通过差分策略实现个体变异, 取得了较好的优化效果。针对 BBPSO 优化算法容易陷入局部最优以及过早收敛的缺点, 在基本 BBPSO 基础上, 借鉴差分进化中利用差分量对种群进行变异操作的思想, 提出了一种新的本质粒子群优化算法, 称为差分变异本质粒子群算法(DMBBPSO)。将该算法结合模糊熵应用于标准图像和煤尘图像分割的实验结果表明该算法取得了令人满意的分割结果, 且算法运算时间很小, 能够满足对煤尘浓度实时精确测量的要求。

## 1 差分变异本质粒子群优化算法(DMBBPSO)

### 1.1 本质粒子群算法

粒子群优化算法(PSO)是一种进化计算技术, 最早由 Kennedy 与 Eberhart 于 1995 年提出的。源于对鸟群捕食行为研究的 PSO 同遗传算法类似, 是一种基于迭代的优化工具。PSO 中, 每个个体称为粒子, 所有的粒子都有一个由被优化的函数决定的适应值, 每个粒子还有一个速度决定他们飞翔的方向和距离, 然后粒子就追随当前的最优粒子在解空间搜索。假定粒子规模为  $N$ , 搜索空间为  $M$  维, 则第  $i$  个粒子的位置表示为  $X_i(x_{i1}, x_{i2}, \dots, x_{iM})$ , 第  $i$  个粒子的速度表示为  $V_i(v_{i1}, v_{i2}, \dots, v_{iM})$ , 每一个粒子的具有最好适应值的位置称为个体最好位置, 记为  $P_i(p_{i1}, p_{i2}, \dots, p_{iM})$ , 整个种群中的最好适应值位置称为全局最好位置, 记为  $P_g(p_{g1}, p_{g2}, \dots, p_{gM})$ 。在找到这 2 个最优值时, 每个粒子根据如下公式来更新自己的速度和新的位置

$$v_{ij}(t+1) = \omega v_{ij}(t) + c_1 \text{rand}() (p_{ij}(t) - x_{ij}(t)) + c_2 \text{rand}() (p_{gj}(t) - x_{ij}(t)), \quad (1)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1), \quad (2)$$

其中:  $1 \leq j \leq M$ ,  $c_1, c_2$  为加速常数, 通常在  $0 \sim 2$  间

取值;  $\text{rand}()$  为均匀分布在  $(0, 1)$  上的随机数;  $\omega$  为惯性因子。

2003年, Kennedy<sup>[7]</sup>提出了本质粒子群优化算法(BBPSO), 相比 PSO, BBPSO 忽略了速度项对位置的影响, 用粒子的高斯分布来更新粒子的位置, 即用式(3)代替式(1)、(2)

$$x_{ij} = N(\mu_{ij}, \sigma_{ij}^2), \quad (3)$$

其中  $N$  表示均值为  $\mu_{ij} = (p_{gj} + p_{ij})/2$  方差为  $\sigma_{ij} = |p_{gj} - p_{ij}|$  的高斯分布。

### 1.2 差分进化算法

差分进化的基本步骤和遗传算法一样。初始化种群  $\{x_{i,0} = (x_{1,i,0}, x_{2,i,0}, \dots, x_{D,i,0}) \mid i=1, 2, \dots, NP\}$  是根据均匀分布  $x_j^{\text{low}} \leq x_{j,i,0} \leq x_j^{\text{up}}, j=1, 2, \dots, D$ , 随机产生的, 其中  $D$  为优化问题的维数,  $NP$  为种群规模。初始化之后, 差分进化进入循环进化过程: 变异、交叉、选择。

变异操作: DE 通过差分策略对个体实行变异, 常见的变异策略有

1) DE/rand/1

$$v_{i,g} = x_{r0,g} + F_i(x_{r1,g} - x_{r2,g}), \quad (4)$$

2) DE/current-to-best/1

$$v_{i,g} = x_{i,g} + F_i(x_{\text{best},g} - x_{i,g}) + F_i(x_{r1,g} - x_{r2,g}), \quad (5)$$

3) DE/best/1

$$v_{i,g} = x_{\text{best},g} + F_i(x_{r1,g} - x_{r2,g}), \quad (6)$$

其中:  $r1, r2, r3 \in \{1, 2, \dots, NP\} \setminus \{i\}$  且互不相同;  $F_i$  为缩放因子;  $X_{i,g}$  表示第  $g$  代种群中第  $i$  个个体;  $x_{\text{best},g}$  表示第  $g$  代种群中的最优值。

交叉操作: 利用式(4)对  $x_{i,g}$  和变异个体  $V_{i,g}$  进行交叉操作, 生成实验个体  $u_{j,i,g}$ , 即

$$u_{j,i,g} = \begin{cases} v_{j,i,g}, & \text{if } \text{rand}(0, 1) \leq CR_i \text{ or } j = j_{\text{rand}}, \\ x_{j,i,g}, & \text{otherwise.} \end{cases} \quad (7)$$

其中:  $\text{rand}(0, 1)$  为  $[0, 1]$  均匀分布的随机数;  $j_{\text{rand}}$  为 1 到  $D$  之间的随机量。

选择操作: 利用下式对实验个体  $u_{j,i,g}$  和  $x_{i,g}$  的目标函数进行比较, 对于最小化问题, 则选择目标函数值低的个体作为新种群的个体  $x_{i,g+1}$ , 即

$$x_{i,g+1} = \begin{cases} u_{i,g}, & \text{if } f(u_{i,g}) < f(x_{i,g}); \\ x_{i,g}, & \text{otherwise,} \end{cases} \quad (8)$$

其中  $f$  为目标函数。

### 1.3 DMBBPSO 优化算法

文献[17]将式(3)表示的基本本质粒子群写成式(9)表示的通用公式

$$x_{ij}(t+n) = \gamma_{gj}(t+n)p_{gj} + \gamma_{ij}(t+n)p_{ij}, \quad (9)$$

其中  $\gamma_{gj}(t+n)$  和  $\gamma_{ij}(t+n)$  为满足某种分布的随机数。

从式(9)可以看出,本质粒子群算法对粒子位置的更新依赖于全局最优值  $p_{gj}$  和个体最优值  $p_{ij}$ ,又从差分进化的变异操作看出,差分进化中对个体的变异操作依赖于第  $g$  代的最优值  $X_{best,g}$  和差分量  $(x_{r1,g} - x_{r2,g})$ 。受差分进化思想的启发,在式(9)的基础上,借助 DE/current-to-best/1 差分进化中利用差分量  $(x_{r1,g} - x_{r2,g})$  对种群进行变异操作的思想,提出了式(10)表示的差分变异本质粒子群优化算法,即

$$x_{ij}(t+1) = x_{ij}(t) + F_i(t+1)(P_{ij}(t) - x_{ij}(t)) + F_i'(t+1)(x_{i1j}(t) - x_{i2j}(t)), \quad (10)$$

$$P_{ij}(t) = \lambda p_{gj} + (1-\lambda)p_{ij}, \quad (11)$$

其中,  $r1, r2 \in \{1, 2, \dots, NP\} \setminus \{i\}$  且互不相同,  $F_i, F_i', \lambda \in U(0, 1), F_i, F_i'$  为缩放比例因子,  $\lambda$  为权重因子。

将式(11)代入式(10),化简后

$$x_{ij}(t+1) = x_{ij}(t) + \lambda F_i(t+1) \cdot (p_{gj}(t) - x_{ij}(t)) + (1-\lambda)F_i(t+1)(p_{ij}(t) - x_{ij}(t)) + F_i'(t+1)(x_{i1j}(t) - x_{i2j}(t)), \quad (12)$$

若选择缩放比例因子  $F_i(t+1)$  为趋于 1 的常数,则式(12)可近似记为

$$x_{ij}(t+1) = x_{ij}(t) + \lambda(p_{gj}(t) - x_{ij}(t)) + (1-\lambda)(p_{ij}(t) - x_{ij}(t)) + F_i'(t+1)(x_{i1j}(t) - x_{i2j}(t)). \quad (13)$$

很明显,  $\lambda=1$  时,式(13)中缺少有关  $p_{ij}(t)$  项,即粒子位置的更新不依赖于粒子个体最优值,则粒子失去认知能力,算法收敛速度变快,但容易陷入局部最优;若  $\lambda=0$  时,式(13)中缺少有关  $p_{gj}(t)$  项,即粒子位置的更新不依赖于粒子全局最优值,则粒子失去社会信息共享能力。即  $\lambda$  大小决定了粒子更新时对全局最优值和个体最优值的依赖程度,差分项的加入可以维护算法全局和局部搜寻能力的平衡。

$\lambda$  的大小决定了式(13)的优化性能,可以想象,当迭代运算的开始阶段,粒子位置的更新应更多依赖于个体最优值,随着迭代次数的增加,粒子位置的更新应更多依赖于全局最优值,因此,  $\lambda$  应随着迭代次数的增加而线性增大。基于以上原因,选用下式来更新  $\lambda$

$$\lambda = \lambda_{\min} + t \times (\lambda_{\max} - \lambda_{\min}) / t_{\max}, \quad (14)$$

其中,  $t$  表示迭代次数,  $t_{\max}$  表示总迭代次数,  $\lambda_{\max}$  和  $\lambda_{\min}$  分别表示最大和最小权重因子。

## 2 基于 DMBBPSO 的模糊熵图像分割

### 2.1 图像的最大模糊熵

给定一幅灰度级为  $L$  的  $m \times n$  的图像  $Y$ , 令  $D = \{(i, j): i=0, 1, \dots, m-1; j=0, 1, \dots, n-1\}$  和  $G = \{0, 1, \dots, L-1\}$ , 定义  $y_{ij}$  为图像  $Y$  中  $(i, j)$  点的灰度值。令

$$D_k = \{(i, j): y_{ij} = k, (i, j) \in D\}, k = 0, 1, \dots, L-1,$$

$$h_k = \frac{n_k}{m \times n}, \quad (15)$$

其中  $n_k$  为  $D_k$  中元素的个数, 则  $H = \{h_0, h_1, \dots, h_{L-1}\}$  为图像的直方图, 其概率分布为:  $p_k = h_k, k = 0, 1, \dots, L-1$ 。

根据模糊理论, 图像  $Y$  可看成是一个模糊事件。  $\mu_Y(y_{ij})$  为该点属于某种特征的隶属度,  $0 \leq \mu_Y(y_{ij}) \leq 1, 0 \leq i \leq m, 0 \leq j \leq n$ 。对于单阈值分割, 设分割阈值  $T$  根据灰度值将原始图像的像素分成 2 个模糊集, 即黑(dark)和亮(bright)2 个集合。黑模糊集包含低灰度值的像素, 对应图像的背景; 亮模糊集包含高灰度值的像素, 对应图像的目标。这 2 个模糊集的隶属函数  $\mu_d(k), \mu_b(k)$  可分别定义如下

$$\mu_d(k) = \begin{cases} 1, & k \leq a; \\ 1 - \frac{(k-a)^2}{(c-a) \times (b-a)}, & a < k \leq b; \\ \frac{(k-c)^2}{(c-a) \times (c-b)}, & b < k \leq c; \\ 0, & c < k; \end{cases} \quad (16)$$

$$\mu_b(k) = \begin{cases} 0, & k \leq a; \\ \frac{(k-a)^2}{(c-a) \times (b-a)}, & a < k \leq b; \\ 1 - \frac{(k-c)^2}{(c-a) \times (c-b)}, & b < k \leq c; \\ 1, & c < k. \end{cases} \quad (17)$$

其中参数  $a, b, c$  满足  $0 \leq a < b < c \leq 255, b = (a+c)/2$ , 则  $[a, c]$  是模糊区间,  $[0, a]$  和  $[c, 255]$  是非模糊区间。综合以上各式, 可得这 2 个集合的模糊熵为

$$H_d = \sum_{k=0}^{255} \frac{p_k \cdot \mu_d(k)}{p_d} \ln \left( \frac{p_k \cdot \mu_d(k)}{p_d} \right); \quad (18)$$

$$H_b = \sum_{k=0}^{255} \frac{p_k \cdot \mu_b(k)}{p_b} \ln \left( \frac{p_k \cdot \mu_b(k)}{p_b} \right). \quad (19)$$

则模糊事件的总模糊熵为

$$H(a, c) = H_d + H_b. \quad (20)$$

由信息论可知, 一个事件的熵越大, 所含有的信

息量就越大。为了实现目标与背景的最佳分割,模糊事件的模糊熵应为最大。为此,可通过优化模糊参数 $(a_{\text{opt}}, c_{\text{opt}})$ ,选择其最佳组合,使总模糊熵 $H(a, c)$ 达到最大值,并据此确定最优阈值 $T_{\text{opt}}$ 。最优阈值满足如下条件

$$\mu_d(T_{\text{opt}}) = \mu_b(T_{\text{opt}}) = 0.5. \quad (21)$$

因此,最优阈值为

$$T_{\text{opt}} = b_{\text{opt}} = (a_{\text{opt}} + c_{\text{opt}})/2. \quad (22)$$

## 2.2 基于 DMBBPSO 的模糊熵图像分割算法

根据最大模糊熵原理,基于最大模糊熵的图像分割算法其本质是在图像的整个灰度空间上搜索一组参数 $(a, c)$ 使图像的总模糊熵取最大值的优化问题。将差分变异本质粒子群(DMBBPSO)用于搜索一组最优参数 $(a, c)$ ,从而得到分割阈值 $T_{\text{opt}}$ 对图像进行分割。

基于 DMBBPSO 的模糊熵图像分割算法的基本步骤为

1)初始化:按下式初始化粒子群的位置矩阵 $\mathbf{X}$ ,设定粒子群规模 $N$ 和维数 $D$ (由于需寻优 2 个参数,因此 $D=2$ ),设定

$$x_{ij} = x_{\min} + (x_{\max} - x_{\min}) \times \text{rand}(), \quad (23)$$

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ \dots & \dots \\ x_{N1} & x_{N2} \end{bmatrix}. \quad (24)$$

$\text{rand}()$ 均匀分布在 $(0, 1)$ 上的随机数, $x_{\max}$ 和 $x_{\min}$ 分别为 $x$ 的最大、最小值,一般取 $x_{\max} = L_{\max}$ , $x_{\min} = L_{\min} + 1$ , $L_{\max}$ 和 $L_{\min}$ 分别为图像的最大、最小灰度。

2)选择式(20)作为 DMBBPSO 算法的适应度函数,计算每个粒子的适应值,并根据适应值选择每

个粒子的当前最好位置 $P_i$ 和粒子群的全局最好位置 $P_g$ 。

3)根据式(13)更新粒子的位置。

4)若达到最大迭代次数,则算法结束,否则,转步骤 2)。

5)求出全局最优解 $p_g$ 对应的参数组合 $(a, c)$ ,计算分割阈值 $T_{\text{opt}}$ 对图像进行分割。

## 3 实验结果及分析

利用算法对不同类型图像进行分割实验,并与基本 BBPSO 算法、文献[16]的算法(GMBBPSO)的结果进行对比。实验中相关参数选择如下:粒子群规模 $N=10$ ,维数 $D=2$ ,最大迭代次数 $t_{\max}=50$ , $\lambda_{\min}=0.05$ , $\lambda_{\max}=0.95$ , $F_i=0.8$ 。

实验中采用的图像分别为 Lena、Boat 和两幅真实煤尘图像,它们代表几种不同类型的图像。图 1 是它们的灰度直方图,Lena 图像呈多峰模式;Boat 图像和 Blood 图像为明显的双峰;煤尘图像为单峰模式。

用算法(DMBBPSO)、基本 BBPSO 算法和 GMBBPSO 算法结合模糊熵对 3 种不同类型的图像进行了分割效果比较实验,实验效果如图 2 所示,其中,图(ai)为原始图像;图(bi)为基本 BBPSO 算法的分割结果;图(ci)为文献[16]的分割结果;图(di)为研究算法的分割结果, $i=1, 2, 3, 4, 5$ 。从图 2 的分割结果可以看出,研究提出算法的分割效果明显优于基本 BBPSO 算法和 GMBBPSO 算法,并且从后面表 1 列出数据可以看出所用时间比 GMBBPSO 还小。

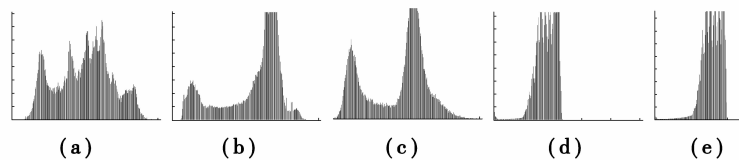


图 1 实验图像直方图

表 1 列出了不同算法的分割阈值、运算时间(s)以及广泛使用的 Sahoo<sup>[18]</sup>等提出的无差异测量。其数据是在 Pentium Dual E2200 2.0GHz CPU、1G RAM 的 PC 机以及 Matlab 7.0 软件环境下得到的。无差异测量定义为

$$u = 1 - 2 \times c \times \frac{\sum_{j=0}^c \sum_{i \in R_j} (y_i - \mu_j)^2}{m \times n \times (L_{\max} - L_{\min})^2}, \quad (25)$$

其中: $c$ 为阈值数量; $R_j$ 为 $j$ 阶分割区域; $y_i$ 为像

素 $i$ 的灰度值; $\mu_j$ 为 $j$ 阶分割区域灰度平均值, $m \times n$ 为图像总的像素点, $L_{\max}$ 和 $L_{\min}$ 为图像的最大最小灰度值。 $u \in [0, 1]$ , $u$ 越接近于 1 说明分割效果越好。

综合图 2 和表 1 可以看出,提出的 DMBBPSO 分割算法在分割的视觉效果和分割性能指标上相比基本 BBPSO 和 GMBBPSO 都具有明显的优势。相对于基本 BBPSO,在运算时间略有增加(增加约 5%)的情况下大大提高了分割效果;相对于 GMBBPSO,在运算时间减少了 20% 以上的情况下

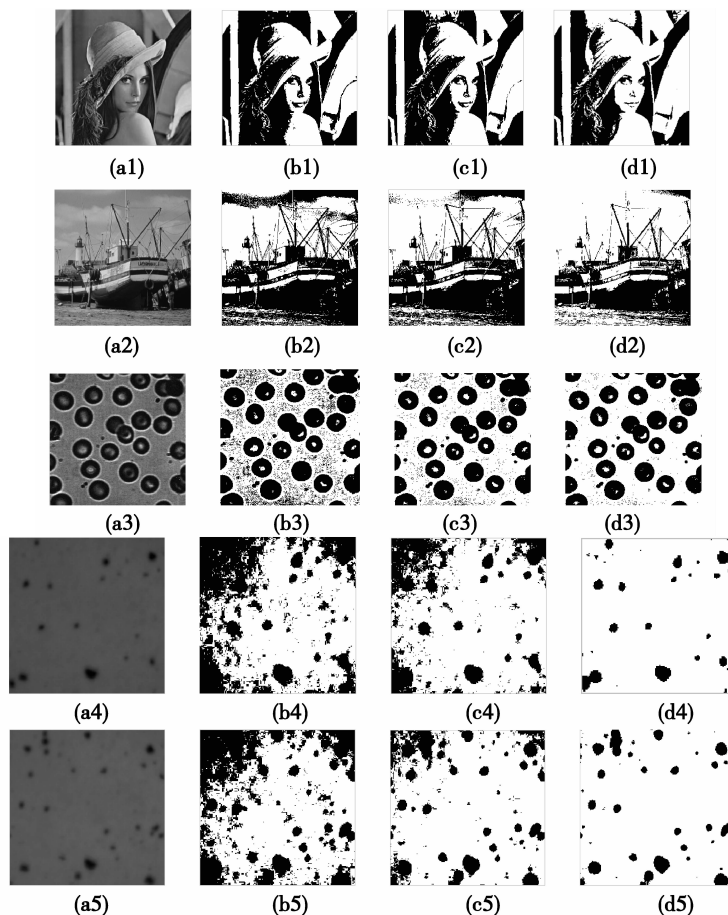


图 2 实验结果比较图

获得了更好的分割效果。

表 1 算法与其它算法进行图像分割性能比较

图像	算法	阈值	时间/s	UM
Lena 256×256	BBPSO	134.510 5	0.058	0.975 8
	GMBBPSO	128.851 7	0.086	0.977 6
	DMBBPSO	122.732 6	0.061	0.978 8
Boat 512×512	BBPSO	160.223 0	0.068	0.958 6
	GMBBPSO	150.835 5	0.092	0.965 0
	DMBBPSO	145.253 7	0.072	0.975 6
Blood 272×265	BBPSO	129.365 8	0.067	0.943 5
	GMBBPSO	114.389 5	0.089	0.968 5
	DMBBPSO	118.796 3	0.071	0.982 6
Coal dust 1 512×512	BBPSO	110.906 8	0.083	0.967 9
	GMBBPSO	107.615 5	0.129	0.988 6
	DMBBPSO	102.820 9	0.087	0.998 9
Coal dust 2 512×512	BBPSO	117.632 6	0.089	0.966 5
	GMBBPSO	114.266 0	0.132	0.987 8
	DMBBPSO	110.992 0	0.090	0.998 6

从上面列出的实验结果和数据可以看出,提出的算法取得了令人满意的分割效果,并且运算时间很小(最长时间在 90 ms 之内),完全能满足实时精确分割的要求,这为煤尘浓度的实时精确测量打下了坚实的基础。

## 4 结 论

受差分进化利用差分量对种群进行变异操作思想的启发,在基本本质粒子群算法的基础上,提出了差分变异本质粒子群(DMBBPSO)算法,DMBBPSO克服了BBPSO存在易陷入局部最优以及过早收敛的问题,提高了优化性能。结合图像模糊熵,利用DMBBPSO搜寻图像模糊参数的最优组合,确定图像的分割阈值。将该算法用于标准图像和煤尘图像分割的实验结果充分表明该算法对不同类型的图像均能取得较好的分割结果,且计算量很小,稍加改进就可以在 DSP 等硬件上实现,能够满足对煤尘浓度实时精确测量的要求。

## 参考文献:

- [1] MURTHY C A, PAL S K. Histogram thresholding by minimizing gray level fuzziness [J]. Information Sciences, 1992, 60(2): 107-135.
- [2] ZHAO B, GUO C X, CAO Y J. A multiagent-based particle swarm optimization approach for optimal reactive power dispatch [J]. IEEE Transactions on Power Systems, 2005, 20(2): 1070-1078.
- [3] TING T O, RAO M V C, LOO C K. A novel approach for unit commitment problem via an effective hybrid particle swarm optimization [J]. IEEE Transactions on Power Systems, 2006, 21(1): 411-418.
- [4] LEANDRO D, SANTOS C, BRUNO M H. Fuzzy identification based on a chaotic particle swarm optimization approach applied to a nonlinear yo-yo motion System [J]. IEEE Transactions on Industrial Electronics, 2007, 54(6): 3234-3245.
- [5] WU J K, ZHU J Q, CHEN G T, et al. A hybrid method for optimal scheduling of short-term electric power generation of cascaded hydroelectric plants based on particle swarm optimization and chance-constrained programming [J]. IEEE Transactions on Power Systems, 2008, 23(4): 1570-1579.
- [6] LIN C J, CHEN C H, LIN C T. A hybrid of cooperative particle swarm optimization and cultural algorithm for neural fuzzy networks and its prediction applications [J]. IEEE Transactions on systems, Man, and Cybernetics Part C: Applications and Reviews, 2009, 39(1): 55-68.
- [7] KENNEDY J. Bare bones particle swarms [C]// Proceedings of the IEEE Swarm Intelligence Symposium, April 24-26, 2003. Indianapolis, Indiana; Institute of Electrical and Electronics Engineers Inc., 2003: 80-87.
- [8] YANG X M, YUAN J S, YUAN J Y, et al. A modified particle swarm optimizer with dynamic adaptation [J]. Applied Mathematics and Computation, 2007, 189(2): 1205-1213.
- [9] TRIPATHI P K, BANDYPADHYAY S, PAL S K. Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients [J]. Information Sciences, 2007, 177(22): 5033-5049.
- [10] ARUMUGAM M S, RAO M V C. On the improved performances of the particle swarm optimization algorithms with adaptive parameters, cross-over operators and root mean square (RMS) variants for computing optimal control of a class of hybrid systems [J]. Applied Soft Computing, 2008, 8(1): 324-336.
- [11] JIAO B, LIAN Z, GU X. A dynamic inertia weight particle swarm optimization algorithm [J]. Chaos, Solitons and Fractals, 2008, 37(3): 698-705.
- [12] NATSUKI H, HITOSHI I. Particle swarm optimization with Gaussian mutation [C]// Proceedings of the IEEE swarm intelligence symposium, April 24-26, 2003. Indianapolis, Indiana; Institute of Electrical and Electronics Engineers Inc., 2003: 72-79.
- [13] CHEN P H. Pumped-storage scheduling using evolutionary particle swarm optimization [J]. IEEE Transactions on Energy Conversion, 2008, 23(1): 294-301.
- [14] LI L Y, LI D R. Fuzzy entropy image segmentation based on particle swarm optimization [J]. Progress in Natural Science, 2008, 18(9): 1167-1171.
- [15] LING S H, LU H H C, CHAN K Y, et al. Hybrid particle swarm optimization with wavelet mutation and its industrial applications [J]. IEEE Transactions on Systems, Man, and Cybernetics Part C: Applications and Reviews, 2008, 38(3): 743-763.
- [16] KROHLING R A, MENDEL E. Bare bones particle swarm optimization with gaussian or cauchy jump [C]// Proceedings of the IEEE 9th Congress on Evolutionary Computation, May 18-21, 2009. Trondheim, Norway; IEEE, 2009: 3285-3291.
- [17] PAN F, HU X H, EBERHART R, et al. An analysis of bare bones particle swarm [C]// Proceedings of the 2008. IEEE Swarm Intelligence Symposium, September 21-24, 2008. St. Louis, MO, USA; IEEE, 2008: 1-5.
- [18] SAHOO P K, SOLTANI S, WONG A K C, et al. A survey of thresholding techniques [J]. Computer Vision Graphics Image Processing, 1988, 41(2): 233-260.

(编辑 侯 湘)