

文章编号:1671-6833(2018)01-0059-08

均衡策略粒子群算法在图像分割中的应用

夏星宇¹, 高 浩¹, 王创业²

(1. 南京邮电大学 自动化学院, 江苏 南京 210046; 2. 安徽省蚌埠市供电局, 安徽 蚌埠 233000)

摘 要: 针对图像阈值分割方法由于其穷举的性质使分割时间随着阈值数目的增加而无法满足不同图像处理要求的问题, 提出一种基于均衡策略的粒子群进化算法来缩短分割的时间. 改进的算法在粒子运行过程中引入均衡因子以增强个体获得较大搜索能力的可能性, 确保它能进行有效的全局搜索; 此外, 为了增强算法的局部搜索能力, 在群体进化方向中引入一个扰动因子, 从而使得个体能够在该方向上获得更多的局部搜索机会. 基于熵准则的 Kapur 用来作为验证提出的算法优劣. 标准测试函数和标准图像实验结果表明, 提出的算法相比于其它比较算法而言, 在寻优能力和收敛速度上获得了更好的成绩.

关键词: 多阈值; 粒子群进化算法; 搜索能力; 最大熵法

中图分类号: TP18 文献标志码: A doi:10.13705/j.issn.1671-6833.2018.01.012

0 引言

图像分割的目的是将一幅图像划分成有意义的区域或部分, 其中每部分区域具有相似的特征. 近几年来, 学者们提出了很多较为有效的图像分割技术^[1]. 它们大致可以分为两类: 一类是利用图像的灰度直方图, 通过其特征来决定分割的最佳阈值^[2]. 另一类是构造一个适应度函数, 通过优化适应值函数来决定最优的阈值, 如基于熵的阈值分割技术, 贝叶斯误差最小化等. 这些技术在医学图像、人脸识别、计算机视觉等领域已经取得了不错的效果. 然而, 其中的一些方法计算非常耗时, 如 Kapur 和 Otsu 算法^[3-4]的运算时间相当长, 且随着阈值数目的增加运算时间呈指数增长, 其穷举的搜索策略限制了它在处理实际多阈值问题中的应用.

近年来, 为缩短运算时间, 研究人员已经成功地将很多优化算法引入到阈值图像分割中来. 作为基于种群的优化策略, 进化算法受自然界生物的启发, 主要采用 3 个主要操作(选择、变异和交叉)来产生后代. 由于有着运算速度快和全局搜索能力较强等特点, 近几年, 进化计算已经广泛地应用到了图像分割领域. Tao 等^[5]采用蚁群算法来优化基于熵的目标分割适应值函数, 寻找最优

的阈值. Yin 等^[6]提出一种改进的遗传算法来缩短多阈值分割的时间. Gao 等^[7]将粒子群优化算法引入到图像阈值分割领域, 实验证明它可以取得和穷举搜索相近的实验结果. Yin^[8]将基于协作策略的量子粒子群算法 QPSO 应用到图像分割中来, 实验证明当处理同一分割问题时, 该算法可以取得比其他同比优化算法更好的效果.

作为较为流行的进化算法之一, 粒子群算法兼有进化计算和群智能的特点. 与其他进化算法相类似, 该算法模拟鸟群集体飞行觅食的行为, 通过鸟之间的集体协作与竞争使群体达到目的. 算法已经成功地应用到工业生产、模式识别、医学处理等多个领域^[9-10]. 但由于粒子群算法具有早熟收敛的缺点, 因此笔者提出一种基于均衡策略的粒子群算法 (particle swarm optimization with an equilibrium strategy, EPSO), 在提出一个有价值的引导点的同时, 合理地赋予粒子在全局和局部搜索的可能性, 较好地均衡了算法的全局和局部搜索能力. 同时把改进后的粒子群算法应用到基于 Kapur 最大熵图像分割中, 有效地解决图像阈值分割耗时这一问题.

1 粒子群进化算法

作为一种基于种群的进化算法, 粒子群进化

算法^[11]操作简便,近年来,受到了广泛的关注.在粒子群中,固定规模的种群代表了一系列的可行解.每个备选解被称为一个“粒子”,多个粒子共存,合作寻优,近似鸟群寻找食物.算法先生成初始种群,即在可行解空间中随机初始化一群粒子,每个粒子都为优化问题的一个可行解,并由目标函数为之确定一个适应值.在每一次迭代过程中,粒子将跟踪两个极值,一个为粒子本身迄今找到的最优解 P_p ,另一为全种群迄今找到的最优解 P_g .基本迭代公式为:

$$v_{id}(t+1) = v_{id}(t) + c_1r_1 \cdot (p_{pid} - x_{id}(t)) + c_2r_2 \cdot (p_{gd} - x_{id}(t));$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1),$$

式中: $i = 1, 2, \cdots, S; d = 1, 2, \cdots, n$; r_1 和 r_2 是服从 $U(0,1)$ 分布的随机数;学习因子 c_1 和 c_2 为非负常数, $c_1 = c_2 = 2$; $v_{id} = [-v_{max}, v_{max}]$, v_{max} 是由用户设定的速度上限; p_{pid} 以及 p_{gd} 分别代表个体最优和群体最优在第 d 维上的解.

2 基于均衡策略的粒子群算法

以往的研究成果表明,传统的粒子群算法具有早熟收敛的缺陷,为在保持粒子群算法收敛速度快的基础上提升算法的全局搜索能力,笔者提出了一种新方法.首先,为了在算法运行过程中一直能够为粒子群提供一个较大的搜索能力,使得粒子群能够保持跳出当前局部搜索区域的概率,笔者提出了一个均衡因子:

$$g = \text{abs}(r_3(0, a_1)/r_4),$$

式中: r_3 为符合高斯分布的随机序列; r_4 是服从 $U(0,1)$ 分布的随机数; a_1 为该高斯分布的标准差.实验结果表明,当 a_1 为 0.35 时,改进后的粒子群算法能够取得较优的实验结果,并保证算法在迭代初期获得较强的全局搜索能力和在后期具有较优的局部搜索能力.

粒子群迭代公式:

$$x_{id}(t+1) = Q_d \pm \alpha \cdot g \cdot (P_{md} - x_{id}(t)),$$

其中, $Q_d = r_1 \cdot p_{pid} + (1 - r_1) \cdot p_{gd} + r_3 \cdot (P_{gd} - \text{round}(P_{gd})).$

与以往算法不同的是式(5)不仅包含了文献[12]证明的粒子群收敛进化方向 $Q_d = r_1 \cdot p_{pid} + (1 - r_1) \cdot p_{gd}$,同时还给出了一个扰动因子 $r_3 \cdot (p_{gd} - \text{round}(p_{gd}))$. r_1 是服从 $[0,1]$ 的均匀分布的随机数, r_3 是服从标准正态分布的随机数.这样不仅给粒子进化提供了一个有效的方向,同时能够保证个体在运行过程中特别是在 $p_{pid} = p_{gd}$ 的情

况下,通过借鉴群体历史最优解 P_g 的数值产生一个相对较小的值,给予个体一个较小的扰动,保证个体能够在局部区域进行有效搜索.

式(4)的第二项则是保证了粒子的均衡搜索能力.它设置的前提是希望能够在提升粒子的全局搜索能力的同时依然赋给个体局部搜索的较大可能性.其中 $P_m = \sum_{i=1}^S \frac{P_{pi}}{S}$, 它的设置是为了群体中每一个个体的历史最优解 P_{pi} 均可以对粒子的当前迭代的搜索能力产生影响.由于在算法初期, P_{pi} 具有较大的区别,因此能够提高粒子在初期的全局搜索能力;而在算法的迭代后期, P_{pi} 具有较大的相似性,能够赋予个体较好的局部搜索能力. α 是 MDPSO 需要调节的参数,它的作用和惯性因子相似,其值的大小同样影响了个体的搜索能力.通过实验比较, α 在迭代过程中由 0.5 下降到 0.35, 可以获得较好的实验结果.为了验证均衡因子的效果,使用蒙特卡罗方法对 g 进行了 50 000 模拟,其概率密度分布图如图 1 和 2 所示,其中图 1 表示的是 $g \in [-2, 2]$ 的分布图,图 2 表示的是 $g \notin [-2, 2]$ 的分布图.

通过对图 1 和 PSO 算法的 c_1 以及 c_2 比较,相对于 c_1 以及 c_2 的取值范围 $[-2, 2]$ 之间的均匀采样,很容易发现 EPSO 算法能够在 $[-1, 1]$ 之间产生更多的解(大致为 40 000 次),从而能够在 Q_d 确定的局部区域产生更为集中的解,这样就可以使得种群获得更多的局部搜索机会;而图 2 的结果表明相对于标准的 PSO 算法的 c_1 以及 c_2 的取值范围, EPSO 能够产生数值更大的备选解,最大值可达到 3.2×10^4 , 这就意味着种群在寻优过程中能够获得更多跳出局部最优解的能力.

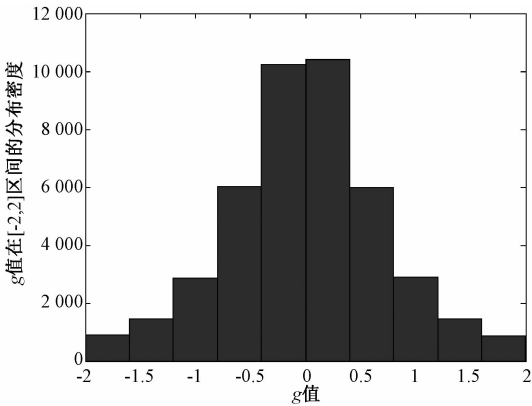
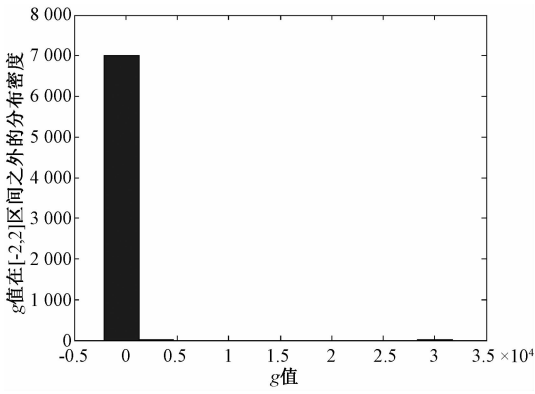


图 1 $g \in [-2, 2]$ 之间的分布图
Fig. 1 The distribution of $g \in [-2, 2]$

图 2 $g \notin [-2, 2]$ 的分布图Fig. 2 The distribution of $g \notin [-2, 2]$

EPSO 算法步骤如下。

- (1) 依照初始化过程,对粒子群的随机位置和速度进行初始设定;
- (2) 计算每个粒子的适应值;
- (3) 对于每个粒子将其适应值与所经历过的最好位置 P_p 的适应值进行比较,若较好,则将其作为当前的最好位置;
- (4) 对于每个粒子将其适应值与全局所经历的最好位置的适应值进行比较,若最好,则将其作为当前的全局最好位置;
- (5) 根据方程(4)和方程(5)对粒子的速度和位置进行进化;
- (6) 如未达到结束条件通常为足够好的适应值或达到一个预设最大代数,则返回步骤(2)。

3 对比试验

3.1 标准测试函数试验设计及结果

为了分析笔者提出的 EPSO 的总体性能,笔者设计了多种测试实验对 PSO^[12]、QPSO^[13]、HRPSO^[14]、HWPSO^[15]、CCPSO^[16] 以及 EPSO 进行了对比分析。同时使用 8 个标准测试函数^[11] 问题来进行多方面考察和比对。笔者测试所有算法使用的个体总群个数都设为 20 (ABC 算法根据算法流程个体总数设置为 10)。测试函数中 $f_1 - f_3$ 是单模函数, $f_4 - f_6$ 是具有多个局部最小点的多模函数,函数的评估次数设置为 $2\,000 \times$ 测试函数维数。 $f_7 - f_8$ 是低维函数,函数的评估次数设置为 10 000。所有测试函数的理论最优解均为 0。最好的对比结果在表 2 和表 4 中加黑显示。

对于单模函数来说,由于 EPSO 使用了 Q_d 点指引整个群体进行搜索,在整个粒子进化过程中一直给予粒子进化提供了一个有价值的方向,特别是当出现 $P_{pd} = P_{gd}$ 的情况时,扰动因子依然能

够提供给粒子在局部区域更多的搜索机会。相对于同样使用文献[10]提出收敛点的 QPSO 来说,由于 EPSO 算法使用了扰动因子,能够提供给粒子群体更多的局部搜索机会。另外,根据图 2 的实验结果,相对于传统的学习因子, EPSO 算法的 g 因子能够在 $[-2, 2]$ 局部范围内提供给粒子以更多的在小范围内搜索的机会,因此粒子可以在 Q 点定义的局部范围内进行精确搜索,从而在单模函数上获得较快和较为精确的解。对于复杂的多模函数,同其他比较算法相比,由于 EPSO 算法使用了均衡因子,使得个体能够获得更大的解,具有较强的跳出局部最小点的能力,从而提高了算法特别是在迭代后期搜索整个解空间的可能性。此外由于算法的局部搜索能力也较为优异,在全局和局部搜索能力上获得了较好的平衡,因此在多模函数上获得了更为精确和满意的结果。表 1 为标准测试函数,表 2 为各种比较算法在测试函数的比较结果。根据表 2 的研究结果表明, EPSO 在低维函数上依然取得较优的结果。因此,可以得出结论, EPSO 不仅在高维函数上,同时在低维函数上表现出了较优的搜索能力。

3.2 图像函数测试设计及结果分析

由 Kapur 提出的基于熵准则的图像分割方法由于可以针对基于直方图的图像提供良好的图像分割而得到了大量应用。传统的 Kapur 方法是基于双阈值的图像分割方法,近年来它已经被成功地应用到多阈值分割领域。具体描述如下。

最佳阈值 $\{t_1^*, t_2^*, \dots, t_{M-1}^*\}$ 要满足以下公式:

$$f([t_1, t_2, \dots, t_{M-1}]) = H_0 + H_1 + \dots + H_{M-1}; \quad (6)$$

$$w_0 = \sum_{i=0}^{t_1-1} P_i; \quad (7)$$

$$H_0 = \sum_{i=0}^{t_1-1} \frac{P_i}{w_0} \ln \frac{P_i}{w_0};$$

$$w_{M-1} = \sum_{i=t_{M-1}}^{L-1} P_i;$$

$$H_{M-1} = \sum_{i=t_{M-1}}^{L-1} \frac{P_i}{w_{M-1}} \ln \frac{P_i}{w_{M-1}}; \quad (8)$$

$$P_i = h(i)/N, \quad (9)$$

其中, N 代表图像包含的像素个数; $L = 255$ 表示最大的灰度级; $h(i)$ 表示的是在第 i 灰度级上图像像素的个数。在 $0 \sim L$ 间改变对应的各个阈值,求满足式(6)为最大值的 $\{t_1, t_2, \dots, t_{M-1}\}$, 此时的 $\{t_1, t_2, \dots, t_{M-1}\}$ 便是最佳阈值。显然要得到 $\arg \max \{f(t_1, t_2, \dots, t_{M-1})\}$, 必须对区间所有的灰度值进行多级熵值计算,最后比较得到最大的熵值,其计算量较大,而且随着分割阈值数目的

表 1 标准测试函数
Tab.1 The benchmark functions

函数名	测试函数公式	测试函数 变化范围	测试函数 最小值	测试函数 最优解
f_1	$\sum_{i=1}^n x_i^2$	$[-100,100]$	0	0
f_2	$\sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	$[-100,100]$	0	0
f_3	$\sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	$[-10,10]$	0	0
f_4	$\sum_{i=1}^n (x_i^2 - 10\cos(2\pi x_i) + 10)$	$[-5.12, 5.12]$	0	0
f_5	$-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^n \cos 2\pi x_i\right)$	$[-32,32]$	0	0
f_6	$\begin{cases} 20e\pi/n\{10\sin^2(\pi y_1) + \\ \sum_{i=1}^{n-1} (y_i - 1)^2 \cdot [1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\} \end{cases}$	$[-50,50]$	0	0
f_7	$100[(x_2 - 10\theta)^2 + (\sqrt{(x_1^2 + x_2^2)} - 1)^2] + x_2^2$	$[-10,10]$	0	(1, 0, 0)
f_8	$(\exp(x_1) - x_2)^4 + 100(x_2 - x_3)^6 + (\tan(x_3 - x_4))^4 + x_1^8$	$[-1,1]$	0	(0, 1, 1, 1)

表 2 各种比较算法在测试函数的比较结果(平均值(方差))

Tab.2 The comparison results between different algorithm on benchmark functions (Mean values (Standard deviation))

算法 名称	f_1		f_2		f_3		f_4	
	维数 = 10	30	10	30	10	30	10	30
	$FES = 20\ 000$	60 000	20 000	60 000	20 000	60 000	20 000	60 000
PSO	1.17E-20	1.58E-13	6.70E-5	347.160 0	1.23E-11	0.004 1	5.647 0	45.043 0
	(6.30E-20)	(4.17E-13)	(1.80E-4)	(228.770 0)	(2.58E-11)	(0.028 4)	(2.679 2)	(10.249 0)
QPSO	9.97E-41	3.50E-21	2.81E-4	206.150 0	2.57E-13	3.15E-11	3.210 0	26.530 2
	(4.20E-40)	(8.60E-21)	(8.30E-4)	(97.930 0)	(1.76E-12)	(1.55E-10)	(1.698 1)	(8.997 0)
MRPSO	5.34E-72	5.70E-69	0.003 0	1.599 8	8.54E-37	2.57E-36	4.603 1	18.942 0
	(3.20E-71)	(2.30E-68)	(0.004 3)	(1.167 0)	(6.00E-36)	(1.70E-35)	(3.218 4)	(5.659 0)
HWPSO	9.77E-15	2.04E-9	0.002 6	97.127 0	2.89E-4	5.287 9	5.710 6	45.530 0
	(2.20E-14)	(2.70E-9)	(0.004 2)	(47.880 0)	(8.51E-4)	(18.422 0)	(3.049 3)	(10.853 0)
CCPSO	1.17E-59	3.30E-45	2.50E-8	41.580 0	1.36E-33	9.84E-27	2.394 0	23.440 0
	(7.20E-59)	(1.30E-44)	(6.60E-8)	(68.450 0)	(3.00E-33)	(2.75E-26)	(1.071 6)	(5.425 0)
EPSO	3.20E-124	2.20E-161	1.71E-5	1.099 2	2.88E-76	1.09E-54	1.976 7	2.340 0
	(2.00E-123)	(9.60E-161)	(9.30E-45)	(0.937 0)	(1.73E-75)	(2.17E-54)	(1.662 9)	(1.450 6)
算法 名称	f_5		f_6		f_7		f_8	
	维数 = 10	30	10	30	3	4		
	$FES = 20\ 000$	60 000	20 000	60 000	10 000	10 000		
PSO	5.87E-11	0.046 2	3.59E-23	0.145 6	5.07E-31	5.53E-12		
	(2.22E-10)	(0.226 4)	(9.40E-23)	(0.509 7)	(6.20E-31)	(8.40E-12)		
QPSO	3.30E-15	3.25E-11	4.71E-32	0.984 0	8.16E-103	1.83E-9		
	(1.36E-15)	(1.70E-10)	(3.30E-47)	(6.443 2)	(7.90E-103)	(2.35E-9)		
MRPSO	0.201 0	1.59E-14	0.074 6	0.064 3	4.24E-148	4.89E-8		
	(0.506 0)	(3.80E-15)	(0.159 3)	(0.126 0)	(1.10E-147)	(4.81E-8)		
HWPSO	3.18E-8	3.36E-5	2.82E-18	0.020 7	2.19E-24	1.64E-12		
	(4.31E-8)	(1.07E-4)	(8.90E-18)	(0.046 4)	(9.13E-24)	(2.92E-12)		
CCPSO	0.023 1	0.030 0	0.006 2	0.056 0	1.72E-101	8.89E-11		
	(0.162 0)	(0.210 2)	(0.043 5)	(0.104 0)	(4.20E-101)	(1.20E-10)		
EPSO	1.17E-15	2.50E-15	4.71E-32	1.57E-32	1.52E-150	1.23E-18		
	(1.75E-15)	(6.90E-16)	(3.30E-47)	(1.10E-47)	(5.50E-150)	(1.90E-18)		

增加其时间复杂度为 $O(255^{M-1})$, 这将严重阻碍了 Kapur 方法的应用以及推广. 为了减少标准 Kapur 算法的运算量, 笔者利用 EPSO 快速收敛特性解决标准 Kapur 算法存在的这一难题. 此外, 为了验证 EPSO 算法的具体性能, 笔者设计了 5 类比较试验: ①ACO 算法^[17]; ②GA-L 算法^[18]; ③ABC 算法^[5]; ④PSO 算法^[11]; ⑤CQPSO^[9] 算法; ⑥QPSO^[13] 算法; ⑦EPSO 算法. 这些算法的参

数设置都严格按照相对应的文献进行处理. 所有算法的比较在 Core 2 Duo 3.2GHZ 电脑使用 MATLAB 语言完成. 分割的阈值分别设为 2、4、6, 对应的函数评价次数为 2 000. 测试的图像分别采用比较算法中最常用的图像以及伯克利大学提供的标准测试图像, 图 3 为示例图像. 此外, 由于标准 Kapur 算法在阈值数超过 4 以后的运算时间过长, 因此只列出阈值数为 2 和 4 的结果.

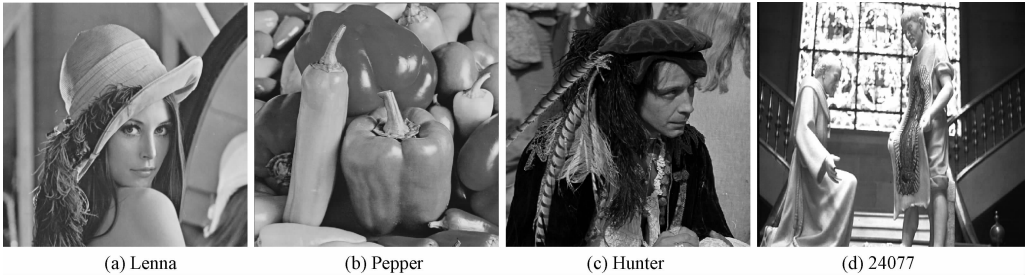


图 3 测试图像
Fig.3 Tested images

从表 3 和 4 的结果可以看出, 笔者提出的 EPSO 算法获得了与标准 Kapur 算法最为接近的适应度值, 同时根据表 3 得到的结果, 基于 EPSO 算法的 Kapur 分割方法所耗费的 CPU 时间随着阈值的增加并不特别明显, 远远降低了标准 Kapur 算法的时间复杂度, 因此, 相对于标准算法, 笔者提出的算法有效地提高了算法的运行效率. 此外, 如表 4 所示, 与 ABC、ACO、PSO、CQPSO、GA、QPSO 6 个基于进化算法的图像分割方法相比, EPSO 在 4 幅图片上均取得了最佳的适应度值和方差, 这主要是和各自算法的搜索策略有关. 由于 ABC 算法采用单维的搜索策略, 使得算法在每一维上都能够进行精细的搜索, 保证了算法的可靠性. 同时由于使用了贪婪搜索策略, 能够给群体进化提供一个较优的方向, 在全局和局部搜索能力上获得了较好的平衡, 因此获得了较好的寻优结果. 蚁群算法由于使用了信息素的概念, 个体在寻优的过程中比遗传算法获得更多的借鉴群体最优解的机会, 因此在比较的图像上获得比遗传算法更好的结果, 但相对于其他群体算法, 信息素的更新较为缓慢, 因此在有限的迭代次数内分割效果差于其他的基于群体算法的分割方法. 而遗传算法相对于其他比较算法来说, 个体在寻优过程中只和父代进行经验交换, 因此收敛速度最慢, 在规定的时间内无法获得优异的成绩, 效果最差. PSO 以及 QPSO 算法虽然相对于其他进化算法有着收敛速度快的优点, 但却易于陷入局部最优, 因此虽然相对于 GA 和 ACO 算法来说获得较好的

实验结果, 但相对于 ABC 和改进的粒子群算法却成绩较差. CQPSO 算法由于使用了协作策略, 因此能够在一定程度上克服维数的束缚, 在高维上 (除去笔者提出的算法) 获得了较好的试验结果. 笔者提出的算法由于使用了均衡因子, 在算法迭代过程中 (特别是迭代后期) 能够获得更多的在全局范围内搜索的机会, 相对于 PSO 及其改进算法提升了全局搜索能力, 从而能够保证最终获得解的可靠性. 同时, 由于算法仍然使用了群体和个体历史最优解的概念, 因此继承了 PSO 算法收敛速度快的特点. 此外, 由于在搜索方向上引入了扰动因子, 能够在算法迭代过程中不断地给予个体一个微动力, 使得个体能够在局部范围内获得更多的搜索机会, 因此具有搜索更为精确解的能力. 图 4 给出了各种比较算法在 Berkely 标准图像分割测试库上测试获得的名次排名总和 (越小越好), EPSO 算法相对于其他算法获得了最好的名次, 同样验证了算法的有效性. 结合表 3 所示结果, PSO 和 EPSO 算法在 2 维和 4 维上的分割耗费的时间为同一量级, 而分割效果远远优于 PSO 算法, 进一步说明本方法改进的必要性. 同时, 对于标准方差这一考核指标, EPSO 也取得了比其他 5 种算法更为优异的稳定性, 这说明笔者所提出的优化算法在分割过程中更加稳定, 鲁棒性好. 图 5 为 EPSO 算法在示例图像阈值为 6 时的分割结果. 整体而言, EPSO 在整个测试图片 4 个阈值上都取得了最好的优化成绩和标准偏差, 说明该算法可以取得更好的分割效果和更稳定的分割性能.

表 3 标准 Kapur、PSO、EPSO 在示例图的比较结果

Tab.3 Comparison among Kapur, PSO, and EPSO on the tested images

分割图像	分割阈值数 $M - 1 = 2$		分割阈值数 $M - 1 = 4$	
	适应度函数值	分割阈值	适应度函数值	分割阈值
Lena	12. 119 9	103, 167	18.050 2	91, 131, 173, 234
Pepper	12. 423 6	84, 150	18. 212 4	58, 91, 131, 171
Hunter	13. 058 7	93, 234	19. 672 2	65, 118, 180, 234
24077	12. 608 6	108, 176	18. 901 8	61, 113, 163, 213
CPU time of Kapur	3. 074		7 890. 193	
CPU time of PSO	2. 732		3. 984	
CPU time of EPSO	2. 528		3. 942	

表 4 基于进化算法的 Kapur 分割方法的比较结果

Tab.4 The comparison results between EAs based Kapur

分割图像	$M - 1$	适应度函数值 (标准方差)						
		EPSO-K	ABC-K	ACO-K	GAL-K	PSO-K	CQPSO-K	QPSO-K
LENN A	2	12. 119 9	12. 119 9	12. 119 8	12. 118 4	12. 119 9	12. 119 9	12. 119 9
		(1. 78E-15)	(1. 78E-15)	(3. 24E-4)	(0. 0018)	(1. 78E-15)	(1. 78E-15)	(1. 78E-15)
	4	18. 019 7	17. 973 1	17. 788 1	17. 924 4	17. 948 7	17. 812 6	17. 853 7
		(0. 074 4)	(0. 046 8)	(0. 168 5)	(0. 088 0)	(0. 150 0)	(0. 156 2)	(0. 107 7)
	6	25. 418 6	25. 402 9	24. 928 1	24. 854 4	25. 341 2	25. 281 0	25. 381 0
		(0. 031 8)	(0. 030 7)	(0. 223 8)	(0. 160 9)	(0. 087 8)	(0. 1473)	(0. 104 5)
PEPPL	2	12. 423 6	12. 423 6	12. 423 3	12. 422 9	12. 423 6	12. 423 6	12. 423 6
		(8. 88E-15)	(8. 88E-15)	(5. 97E-4)	(7. 14E-4)	(8. 88E-15)	(8. 88E-15)	(8. 88E-15)
	4	18. 209 5	18. 209 4	18. 179 2	18. 168 9	18. 206 0	18. 203 8	18. 205 8
		(0. 002 4)	(0. 003 2)	(0. 023 6)	(0. 023 3)	(0. 010 3)	(0. 014 4)	(0. 009 4)
	6	25. 696 6	25. 694 7	25. 526 3	25. 400 6	25. 641 6	25. 652 3	25. 643 7
		(0. 032 6)	(0. 007 5)	(0. 110 3)	(0. 093 7)	(0. 059 1)	(0. 074 2)	(0. 087 4)
HUNTER	2	13. 058 7	13. 058 7	13. 040 8	13. 055 4	13. 058 7	13. 058 7	13. 058 7
		(7. 11E-15)	(7. 11E-15)	(0. 033 1)	(0. 003 3)	(7. 11E-15)	(7. 11E-15)	(7. 11E-15)
	4	19. 665 2	19. 650 7	19. 502 4	19. 482 0	19. 657 9	19. 649 1	19. 651 3
		(0. 009 8)	(0. 018 3)	(0. 130 4)	(0. 090 4)	(0. 012 9)	(0. 032 8)	(0. 014 7)
	6	27. 663 3	27. 654 9	27. 319 6	27. 253 6	27. 588 3	27. 604 7	27. 594 3
		(0. 015 7)	(0. 017 7)	(0. 207 9)	(0. 111 9)	(0. 064 6)	(0. 078 3)	(0. 097 4)
240 77	2	12. 608 5	12. 608 4	12. 605 9	12. 607 0	12. 608 5	12. 608 1	12. 608 5
		(4. 1E-4)	(5. 4E-4)	(0. 002 5)	(0. 001 1)	(4. 10E-4)	(0. 001 7)	(4. 10E-4)
	4	18. 901 8	18. 895 3	18. 880 9	18. 848 4	18. 896 9	18. 898 1	18. 890 5
		(3. 55E-15)	(0. 004 5)	(0. 031 5)	(0. 023 1)	(0. 007 7)	(0. 004 2)	(0. 006 8)
	6	31. 658 3	31. 611 3	31. 353 1	31. 165 1	31. 418 3	31. 462 4	31. 437 4
		(0. 059 2)	(0. 023 2)	(0. 244 2)	(0. 107 2)	(0. 142 2)	(0. 136 3)	(0. 185 7)

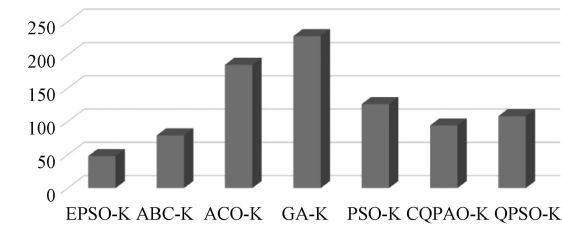


图 4 基于进化算法的 Kapur 分割方法在 Berkeley 标准测试库测试的名次

Fig. 4 The ranking of EAs based Kapur algorithm on Berkeley image database

4 结 论

笔者根据粒子群进化算法本身的特点,首先提出了一个新的均衡策略,使得算法可以在保证全局搜索能力的条件下加快收敛速度.然后在此基础上采用了一个新的引导个体,使得个体能够在局部范围内更为精确地搜索,从而得到更优的优化效果.改进算法在标准测试函数以及图像阈值分割中取得了比其他算法更好的成绩,从而



图 5 EPSO 在阈值为 6 时的分割图

Fig.5 The segmented images with six thresholding by EPSO

证明了其有效性. 以后的工作将继续关注如何在保证分割时间较短的情况下更高地提高算法的优化效果.

参考文献:

[1] 王忠勇,贾萌,候中新,等. 基于形态学特征的颗粒图像分割和计数[J]. 郑州大学学报(工学版), 2015, 36(2): 80-84.

[2] 刘松涛,殷福亮. 基于图割的图像分割方法及其新进展[J]. 自动化学报, 2012, 38(6): 911-922.

[3] SAHOO P K, KAPUR J N, WONG A K C. A new method for gray-level picture thresholding using the entropy of the histogram[J]. Computer vision graphics image processing, 1985, 29(3): 273-285.

[4] OTSU N. A threshold selection method from gray-level histograms[J]. IEEE transactions on system, man and cybernetics, 1979, 9(1): 62-66.

[5] TAO W B, JIN H and LIU L M. Object segmentation using ant colony optimization algorithm and fuzzy entropy[J]. Pattern recognition letter, 2007, 28 (7): 788-796.

[6] YIN P Y, CHEN L H. A fast iterative scheme for multilevel thresholding methods [J]. Signal process, 1997, 60 (3): 305-313.

[7] GAO W F, LIU S Y, HUANG L L. A novel artificial bee colony algorithm based on modified search equation and orthogonal learning [J]. IEEE transactions on cybernetics, 2013, 43(3), 1011-1024.

[8] YIN P Y. Multilevel minimum cross entropy threshold selection based on particle swarm optimization [J]. Applied mathematics and computation, 2007, 184 (2): 503-513.

[9] GAO H, XU W, SUN J, et al. Multilevel thresholding for image segmentation through an improved quantum-behaved particle swarm algorithm [J]. IEEE transactions on instrumentation and measurement, 2010, 59

(4): 934-946.

[10] 朱晓东,刘丹,李广. 基于混合优化算法的模糊系统辨识[J]. 郑州大学学报(工学版), 2015, 36(4): 10-14.

[11] KENNEDY J, BLACKWELL T. Particle swarm optimization [J]. Swarm intelligence, 2007, 1(1): 33-57.

[12] CLERC M, KENNEDY J. The particle swarm-explosion, stability, and convergence in a multi-dimensional complex space [J]. IEEE transactions on evolutionary computation, 2002, 6(1): 58-73.

[13] SUN J, FENG B, XU W. Particle swarm optimization with particle having quantum behavior [C]// Congress on Evolutionary Computation. Portland, OR: IEEE Press, 2004: 325-331.

[14] GAO H, XU W B. A new particle swarm optimization and its globally convergent modifications [J]. IEEE transactions on system, man, and cybernetics, Part B, 2011, 41(5): 1334-1351.

[15] LING S H, IU H H C, CHAN K Y, et al. Hybrid particle swarm optimization with wavelet mutation and its industrial applications [J]. IEEE transactions on system, man and cybernetics, Part B, 2008, 38(3): 743-763.

[16] PARK J B, JEONG Y W, SHIN J R, et al. An improved particle swarm optimization for nonconvex economic dispatch problems [J]. IEEE transaction on power systems, 2010, 25(1): 156-166.

[17] AKAY B. A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding [J]. Applied soft computing, 2012, 13(6): 3066-3091.

[18] CAO L, BAO P, SHI Z K. The strongest schema learning GA and its application to multilevel thresholding [J]. Image vision computing, 2008, 26(5): 716-724.

Multi-level Image Segmentation Based on an Improved Particle Swarm Optimization with an Equilibrium Strategy

XIA Xingyu¹, GAO Hao¹, WANG Chuangye²

(1. College of Automation, Nanjing University of Posts and Telecommunications, Nanjing 210046, China; 2. State Grid Guizhen Electric Power Company, Bengbu, 230000, China)

Abstract: The computation time of some multi-level threshold segmentation techniques needs were too long to bear, and it grew exponentially with the number of thresholds increased. This paper proposed a particle swarm optimization with an equilibrium strategy for shorting its computation time. First, during iterations, a balance operator for individuals to have more chances to search in the global area was introduced. Furthermore, for enhancing the local search ability of our proposed algorithm, a disturbance operator was also been introduced in this paper which enabled the individual had more opportunities to make a precise search. The improved algorithm enables particles had more chances to jump out of a local area for enhancing their global search ability. Meanwhile, a valuable point to guide the search direction of the particles was introduced. Then it accelerated the convergence rate of the improved algorithm. Kapur method was used in this paper to test the performance of the proposed method. Experimental results showed that our proposed algorithm showed more power and fast search ability when compared with the other population-based algorithms.

Key words: multi-level threshold; particle swarm optimization; search ability; kapur method

(上接第 58 页)

[10] 张占龙,王科,李德文,等. 变压器谐波损耗计算及影响因素分析[J]. 电力系统保护与控制, 2011, 39(4): 68-72, 78.

[11] 刘书铭,施红,冯蕾. 考虑集肤效应与邻近效应的变压器谐波损耗模型[J]. 电力自动化设备, 2015, 35(3): 133-139.

[12] SADATISB, TAHANIA, DARVISHIB, et al. Comparison of Distribution Transformer Losses and Capacity under Linear and Harmonic Loads[C] // 2nd IEEE international conference on power and energy. Johor Baharu, Malaysia: IEEE, 2008, 1265-1269.

[13] 李琼林,邹磊,刘会金,等. 电力变压器谐波损耗仿真计算与实验研究[J]. 电网技术, 2013, 37(12): 3521-3527.

[14] 夏焰坤. 电气化铁路单三相组合式变压器电气特性分析[J]. 郑州大学学报(工学版), 2016, 37(4): 11-14, 35.

[15] IEEE Recommended Practice for Establishing Liquid-Filled and Dry-Type Power and Distribution Transformer Capability When Supplying Nonsinusoidal Load Currents: IEEE Std C57.110—2008[S].

Quantification Calculation and Modeling Simulation of Distribution Network Losses Considering Harmonic Factor

WEI Chaofeng^{1,2}, LI Qionglin², JIANG Jiandong¹, TANG Yuzheng², WANG Chun¹

(1. School of Electrical Engineering, Zhengzhou University, Zhengzhou 450001, China; 2. State Grid Henan Electric Power Research Institute, Zhengzhou 450052, China)

Abstract: Accurate line loss calculation was the basis of realizing economic operation of power grid. The harmonic pollution in distribution network was becoming increasingly serious. Because of harmonic factors, the distribution network operation state and equipment parameters have changed greatly, and the additional losses caused should not be neglected. Based on the analysis of the traditional calculation method of line loss, this paper systematically analyzed the mechanism of harmonic effect on distribution transformer and power line. By introducing the transmission line AC resistance coefficient and transformer harmonic loss factors, a loss calculation model considering harmonic was put forward. Then, combined with the typical distribution transformer and transmission line, the harmonic loss was calculated quantitatively. Finally, the validity and accuracy of the model were verified by modeling and simulation.

Key words: harmonic; line loss; AC resistance coefficient; harmonic loss factor; modeling and simulation