

基于量子粒子群算法的 Otsu 图像阈值分割

冯 斌, 王 璋, 孙 俊

(江南大学 信息工程学院, 江苏 无锡 214122)

摘 要: 二维 Otsu 方法同时考虑了图像的灰度信息和像素间的空间邻域信息, 是一种有效的图像分割方法。针对二维 Otsu 方法计算量大的特点, 采用量子粒子群算法来搜索最优二维阈值向量, 每个粒子代表一个可行的二维阈值向量, 通过各个粒子的飞行来获得最优阈值。结果表明, 所提出的方法不仅能得到理想的分割结果, 而且计算量大大减少, 达到了快速分割的目的, 便于二维 Otsu 方法的实时应用。

关键词: 图像分割; 二维 Otsu 方法; 优化; 粒子群算法; 量子粒子群算法

中图分类号: TP391.41 **文献标识码:** A **文章编号:** 1000-7024(2008)13-3429-03

Image threshold segmentation with Otsu based on quantum-behaved particle swarm algorithm

FENG Bin, WANG Zhang, SUN Jun

(School of Information Technology, Southern Yangtze University, Wuxi 214122, China)

Abstract: 2D Otsu method, which considers the gray information and spatial neighbor information between pixels in image simultaneously, is an efficient image segmentation method. However, the computational burden of finding optimal threshold vector is very large for 2D Otsu method. An optimization method, i.e., quantum-behaved particle swarm optimization (QPSO), is used to find the best 2D threshold vector, in which each particle represents a possible 2D threshold vector, and the best 2D threshold is obtained through the flying among particles. Experimental results show that the proposed method can not only obtain ideal segmentation results but also decrease the computation cost reasonably, and it is suitable for real time application.

Key words: image segmentation; 2D Otsu; optimization; particle swarm algorithm; quantum-behaved particle swarm algorithm

0 引 言

图像分割是图像处理和计算机视觉中基本而且关键的技术之一,其目的是将目标与背景分离,为后续的分类、识别和检索提供依据。图像分割方法包括阈值法、边缘检测法和区域跟踪法等。其中阈值分割是一种实现最为简单、使用最为普遍的分割方法。阈值选取方法有多种如直方图双峰法、最大熵法、Otsu 法、矩量保持法、梯度统计法以及这些方法在二维上的推广等。在这些方法中, Otsu 法以其分割效果好、适用范围广而得到了广泛的应用,但是和其它分割方法一样,它同样存在计算量大、计算时间长的问题。

粒子群优化(particle swarm optimization, PSO)算法是基于群体智能(swarm intelligence)理论的优化算法,它通过群体中粒子间的合作与竞争产生的群体智能指导优化搜索。2004 年 Sun 等在研究了 Clerc 等人关于粒子收敛行为的研究成果后,从量子力学的角度提出了一种新的 PSO 算法模型。认为粒子具有量子行为,并根据这种模型提出了量子粒子群算法 quan-

tum-behaved particle swarm optimization(QPSO)^[1],并且得到了很好的实验效果。

1 粒子群算法和量子粒子群算法

1.1 粒子群算法 (PSO)

由 Kennedy 和 Eberhart 提出的 PSO 算法^[2],起源于对简单社会的模拟,最初设想是模拟对鸟群觅食的过程,后来发现 PSO 是一种很好的优化工具。PSO 优化算法与其它进化算法相类似,也是将寻优的参数组合成群体,通过对环境的适应度来将群体中的个体向好的区域移动。与其它进化算法不同,在描述个体时,将其看成是 D 维寻优搜索空间的一个没有体积的微粒(点),结合微粒的历史最佳位置和群体历史最佳位置信息,以一定的速度向目标值逼近。许多科学和工程问题都可归结成最优化问题,每个优化问题的潜在解都是搜索空间中的一个粒子。第 i 个微粒可以表示成 D 维向量 $X = [X_{i1}, X_{i2}, \dots, X_{iD}]$,微粒的速度表示成 $V_i = [V_{i1}, V_{i2}, \dots, V_{iD}]$ 这个微粒经历的最佳位置(对应于最好的适应度)表示为 $P_i = [P_{i1}, P_{i2}, \dots, P_{iD}]$ 也称为 pbest。

收稿日期: 2007-07-18 E-mail: jsufeng@vip.sina.com

基金项目: 国家自然科学基金项目 (60474030)。

作者简介: 冯斌 (1956 -), 女, 江苏南通人, 博士, 副教授, 研究方向为计算机控制技术; 王璋 (1983 -), 男, 江苏镇江人, 硕士研究生, 研究方向为进化计算及其应用; 孙俊 (1971 -), 男, 江苏无锡人, 博士研究生, 研究方向为人工智能、进化计算。

群体所有微粒经历的最好位置的索引号用 g 表示,记为 P_g ,也称为 $gbest$,第 i 个微粒从 n 代进化到 $n+1$ 代,通过下式进行更新。

$$v[] = v[] + c1 * rand() * (pbest[] - present[]) + c2 * rand() * (gbest[] - present[]) \quad (1)$$

$$present[] = present[] + v[] \quad (2)$$

式中 $v[]$ ——粒子的速度, $present[]$ ——当前粒子的位置。 $pbest[]$ and $gbest[]$ 如前定义 $rand()$ ——介于(0,1)之间的随机数。 $c1$, $c2$ ——学习因子。通常 $c1 = c2 = 2$ 。

PSO 算法概念简单、容易实现、搜索速度快范围大,跟其它优化算法相比,它的优点突出。

1.2 量子粒子群算法 (QPSO)

有很多试验已经证明了 PSO 算法不能收敛于全局最优解,甚至于局部最优解,许多学者已经采用很多方法来改进粒子群优化算法的性能。2004 年 Sun 等在研究了 Clerc 等人关于粒子收敛行为的研究成果后,从量子力学的角度提出了一种新的 PSO 算法模型^[3-4]。认为粒子具有量子行为,并根据这种模型提出了量子粒子群算法 (QPSO),在量子空间中粒子的满足聚集态的性质完全不同,它可以在整个可行解空间中进行搜索,因此 QPSO 算法的全局搜索的性能远远优于标准 PSO 算法。在量子空间中,粒子的速度和位置是不能同时确定的。通过波函数来描述粒子的状态,并通过求解薛定谔方程得到粒子在空间某一点出现的概率密度函数,又通过蒙特卡罗随机模拟方式得到粒子的位置方程^[5,6]。

在具有量子行为的粒子群优化算法中,粒子的主要迭代公式

$$mbest = \frac{1}{m} \sum_{i=1}^M P_i = \left(\frac{1}{M} \sum_{i=1}^M P_{i1}, \frac{1}{M} \sum_{i=1}^M P_{i2}, \dots, \frac{1}{M} \sum_{i=1}^M P_{id} \right) \quad (3)$$

$$P_{id} = \phi * P_{id} + (1 - \phi) * P_{gd} \quad \phi = rand \quad (4)$$

$$x_{id} = P_{id} \pm \beta * |mbest - x_{id}| \ln \left(\frac{1}{\mu} \right) \quad \mu = rand \quad (5)$$

式中 $mbest$ ——粒子群 $pbest$ 的中间位置 P_{id} —— P_{id} 和 P_{gd} 之间的随机点 ϕ 和 μ ——[0,1] 的随机数 β ——QPSO 的收缩扩张系数。

QPSO 的算法流程可以这样描述:

- (1) 初始化粒子群;
- (2) 根据式(3)计算 $mbest$ 的值;
- (3) 求每一个粒子的适应度值,比较求出 P_{id} ;
- (4) 对于每一个粒子比较 P_{id} ,求得 P_{gd} ;
- (5) 更新 P_{gd} ;
- (6) 对于粒子的每一维,根据式(4),在 P_{id} 和 P_{gd} 之间取得一个随机点;
- (7) 根据式(5)获得一个新的位置;
- (8) 重复(2)~(7)直到条件不满足,则迭代过程结束。

1.3 量子粒子群算法的优点

量子粒子群算法能克服一般粒子群算法在收敛性能上的缺陷,因为它具有很多特性:量子系统是一个复杂的非线性系统,符合状态重叠原理,因此量子系统比一个线性系统具有更多的状态;量子系统中一个粒子能以某一确定的概率出现在搜索空间中的任一位置,因为粒子没有一个确定的轨迹;传统 PSO 算法中,有限的搜索范围将粒子限制在一个固定的区域,在 QPSO 算法中,粒子以某一确定的概率出现在整

个可行的搜索空间中任一位置,甚至是远离中间点的位置。这样的位置也可能比当前群体中的 $gbest$ 具有更好的适应值。

2 基于量子粒子群算法的 Ostu 图像分割

2.1 二维 Ostu 图像分割

Otsu 法是由日本人大津首先提出的^[7],也称大津阈值法或最大类间方差法。该方法以图像的一维直方图为依据,以目标和背景的类间方差最大为阈值选取准则,在很多情况下都能取得很好的阈值。实际应用中,由于用一维直方图来确定阈值往往会造成错误分割,常利用原始图像和其邻域平滑图像联合直方图,将一维 Otsu 法推广到二维而得到二维 Otsu 阈值分割方法,使分割效果得到改善,但二维直方图的引入,大大增加了计算复杂性,在很大程度上限制了该算法的应用范围。

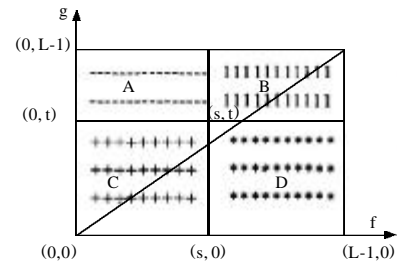


图 1 二维直方图

设一幅图像 $f(x,y)$ 的灰度级为 $(0,1,\dots,L-1)$,其邻域平滑图像 $g(x,y)$ (以 3×3 邻域均值作为该像素灰度值) 的灰度级也为 L ,对于图像中的任何一个像素,就有了一个二元组:像素灰度值 i 和邻域平均灰度值,设像素灰度值为 i ,且邻域平均灰度值为的像素点数为 j ,图像总像素为 M ,则二维联合概率的密度为

$$p_{ij} = \frac{f_{ij}}{M} \quad \text{且} \quad \sum_{i=1}^{L-1} \sum_{j=1}^{L-1} p_{ij} = 1 \quad \sum_{i=1}^{L-1} \sum_{j=1}^{L-1} f_{ij} = M \quad (6)$$

任意给定一个阈值 (s,t) 就可以将图像分割成图 1 所示的 4 个区域: A、B、C、D。其中,对角线上的两个区域 B 和 C 分别对应于目标和背景。而远离对角线的区域 A 和 D 则对应于边缘和噪声。

设目标和背景分别为 C、C0,其出现的概率分别为

$$w_0 = \sum_{i=0}^{s-1} \sum_{j=0}^{t-1} p_{ij} \quad w_1 = \sum_{i=s}^{L-1} \sum_{j=t}^{L-1} p_{ij} \quad (7)$$

两类对应的均值矢量为

$$\mu_0 = (\mu_{00}, \mu_{01})^T = \left(\sum_{i=0}^{s-1} \sum_{j=0}^{t-1} i p_{ij}, \sum_{i=0}^{s-1} \sum_{j=0}^{t-1} j p_{ij} \right) \quad \mu_1 = (\mu_{10}, \mu_{11})^T = \left(\sum_{i=s}^{L-1} \sum_{j=t}^{L-1} i p_{ij}, \sum_{i=s}^{L-1} \sum_{j=t}^{L-1} j p_{ij} \right) \quad (8)$$

二维直方图的均值矢量为

$$\mu = (\mu_0, \mu_1)^T = \left(\sum_{i=0}^{L-1} \sum_{j=0}^{L-1} i p_{ij}, \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} j p_{ij} \right) \quad (9)$$

在大多数情况下,远离对角线的概率可忽略不计,即假设图 1 中 A、D 两部分的概率为零。此时,可以证明

$$w_0 + w_1 \rightarrow 1 \quad w_0 \mu_0 + w_1 \mu_1 \rightarrow \mu_1 \quad (10)$$

则,类间离散矩阵为

$$S_b = \sum_{k=0}^1 w_k [(\mu_k - \mu)(\mu_k - \mu)^T] \quad (11)$$

使用离散矩阵的迹作为背景和类间的距离测度函数,即

$$r_{trace}(S_b) = w_0 [(\mu_0 - \mu)^2 + (\mu_0 - \mu)^2] + w_1 [(\mu_1 - \mu)^2 + (\mu_1 - \mu)^2] \quad (12)$$

2.2 基于量子粒子群算法的 Otsu 图像分割

适应度值即是计算适应度函数所得到的值,它的大小是粒子群算法中选择个体极值和全体极值的依据。适应度函数是根据具体问题设计的,通常在目标函数并不复杂的情况下,可以直接将目标函数选择为适应度函数。本文即以距离测度函数为 $r_{trace}(S_b)$ 适应度函数^[8],求其最大值,即

$$f(s,t)=\max r_{trace}(S_b) \tag{13}$$

图像灰度值为[0,255]之间的正整数,而根据QPSO更新公式得到的位置均为连续值,所以在每次速度更新后都对要其进行取整操作,同时检查位置是否越界(>255或<0)。由于量子粒子群算法与粒子群算法相比较,具有较强的鲁棒性,控制的参数也较少。在本文算法中将惟一的参数 beta 定义为[0,1]之间的随机数。具体的流程如下:

- 步骤 1 随机生成 popsize 个粒子,粒子的位置在(0, 255)之间产生,设置最大迭代次数 MAXITER。
- 步骤 2 根据式 (13) 来计算粒子的适应度。更新每个粒子的个体极值 pbest(i)(i=1,2,3...,popsize)和整个粒子群的全局的极值 gbest(i)(i=1,2,3...,popsize)。
- 步骤 3 根据式(5)来更新粒子的位置。
- 步骤 4 令 t = t+1,返回步骤 2,直到 t = MAXITER。
- 步骤 5 输出粒子群的位置,即输出最佳的阈值的向量 (s^*,t^*) ,分割后的图像 $f_{s^*,t^*}(x,y)$ 为

if $f(x,y)<s^*$ and $f(x,y)<t^*$ then $f_{s^*,t^*}(x,y)=1$;
if $f(x,y)>s^*$ or $f(x,y)>t^*$ then $f_{s^*,t^*}(x,y)=0$;

3 实验结果和实验分析

为了验证本文算法的有效性,采用本文的方法对 Fourviere 图进行图像分割实验。所有的实验都在 Petium 3.0G 的 PC 上利用 Matlab 7.0 进行实验。为了使所做的实验更加具有对比性,fourviere 图的大小为(256*256)。对 fourviere 图分别采用穷举法,PSO 算法和 QPSO 算法进行分割。为了客观的比较算法的性能,对每一幅图像运行 10 次的 pso 算法和 qpso 算法。得到的分割后的图像如图 3 和图 4 所示。



图 2 原图 图 3 一维图像分割 图 4 二维图像分割

由图 3 和图 4 效果所见,二维分割的效果要比一维分割的效果好,后者有好多的错分点,而前者的错分点较少,这主要是因为二维的 Otsu 法考虑了图像像素的灰度信息和像素间的位置关系。对于二维的 Otsu 法,采用穷举法,ps0 算法和 qpso 算法所得的最优的向量阈值,计算时间,最大目标函数和达优数。

实验共运行 10 次,表中的达优数是指运行结果和穷举法结果相同的次数。从表 1 可以看出,在相同条件下,QPSO 和基本 PSO 计算时间差别不大,但两者均不到穷举法的 5%,说明了粒子群的 Otsu 法在计算速度方面的优势,显然这种优势

表 1 3 种算法性能比较

方法	最佳阈值向量	最优目标值	运行时间/s	达优数
穷举法	(70,127)	4 069.5	1 806.542	—
PSO	(70,127)	4 069.5	43.140 6	6
QPSO	(70,127)	4 069.5	40.781 3	10

还会随着阈值维数的增多而加大。同时 QPSO 的达优率为 100%,而基本 PSO 只有 60%,这是由于 qpso 算法的量子性质确保算法收敛,同时提高算法的精度,改善了 PSO 陷入局部最优,保证了计算结果的精确和稳定。与此同时,运行 10 次的 pso 算法和 qpso 算法的平均收敛曲线进行比较如图 5 所示。

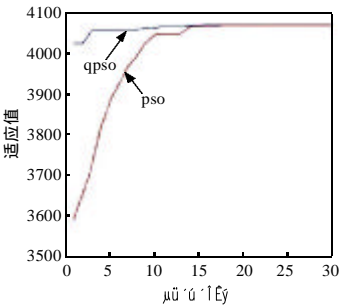


图 5 pso 和 qpso 算法平均收敛曲线

如图 5 所示,两种粒子群算法只需要迭代十几次就能达到最佳的收敛值,搜索到全局的最优阈值向量和全局最优值。qpso 与 pso 相比较而言,具有更快的迭代次数,因此 qpso 的计算时间也要优于 pso 运行的时间。实验结果表明,QPSO 算法是一种高效的优化方法,用它来计算二维 Otsu 方法的最佳阈值向量不仅能搜索到全局最佳阈值向量而且计算代价很小,为二维 Otsu 方法的实时应用提供一个新的途径。

4 结束语

二维 Otsu 方法是一种有效的图像分割方法,不仅考虑了图像的灰度信息,而且考虑了像素间的空间邻域信息,在图像的信噪比较低时,该方法能够得到比一维 Otsu 方法更好的分割结果。由于此法需要根据二维类间方差最大的准则计算二维最优阈值向量,与一维 Otsu 方法相比,其计算量增加。为此,本文提出采用 QPSO 算法来计算最优二维阈值向量。结果表明,QPSO 算法不仅能有效地搜索到全局最优二维阈值向量,而且计算量减少提高了,其效果也优于 PSO 算法,为二维 Otsu 方法的实时应用提供了一个新的途径。

参考文献:

[1] Feng B, Xu W B. Quantum oscillator model of particle swarm system [J]. Computer Engineering, 2006,32(20):18-21.
[2] Feng B, Xu W B. Adaptive particle swarm optimization based on quantum oscillator model[C].Singapore: Proceeding of the IEEE Conference on Cybernetics & Intelligent Systems,2004:291-294.
[3] Sun J, Feng B, Xu W B. Particle swarm optimization with particles having quantum behavior [C]. Oregon, USA: Proceeding of Congress on Evolutionary Computation, 2004:325-331.

(下转第 3434 页)

表1 决策表

U	make (m)	weight (w)	power (p)	comp (c)	mileage (ma)
u_1	USA	M	M	M	H
u_2	USA	H	M	M	H
u_3	USA	L	L	H	M
u_4	USA	M	M	M	H
u_5	USA	H	M	M	H
u_6	JAPAN	M	M	M	M
u_7	JAPAN	H	M	M	M
u_8	JAPAN	L	H	M	M
u_9	JAPAN	M	M	M	M
u_{10}	JAPAN	H	M	H	M

步骤1 计算可辨识矩阵 M ,如表2所示。CORE = $\{m\}$ 。

表2 可辨识矩阵

U	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9	u_{10}
u_1	\emptyset	\emptyset	wpc	\emptyset	\emptyset	m	mw	$mwpc$	m	$mwpc$
u_2		\emptyset	wpc	\emptyset	\emptyset	mw	m	$mwpc$	mw	mc
u_3			\emptyset	wpc	wpc	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset
u_4				\emptyset	\emptyset	m	mw	$mwpc$	m	$mwpc$
u_5					\emptyset	$mwpc$	m	$mwpc$	mw	mc
u_6						\emptyset	\emptyset	\emptyset	\emptyset	\emptyset
u_7							\emptyset	\emptyset	\emptyset	\emptyset
u_8								\emptyset	\emptyset	\emptyset
u_9									\emptyset	\emptyset
u_{10}										\emptyset

步骤2 RED=CORE= $\{m\}$ 。分别计算属性的知识量：

$U//\{m\} = \{\{u_1, u_2, u_3, u_4, u_5\}, \{u_6, u_7, u_8, u_9, u_{10}\}\}$,此时 $n_1 = 5, n_2 = 5$,

$W(m) = 1 \times 5 \times 5 = 25$;

$U//\{w\} = \{\{u_1, u_4, u_6, u_9\}, \{u_2, u_5, u_7, u_{10}\}, \{u_3, u_8\}\}$,此时 $n_1 = 4, n_2 = 4$,

$n_3 = 2$,则 $W(w) = 4 \times (4+2) + 4 \times 2 = 32$;

$U//\{p\} = \{\{u_1, u_2, u_4, u_5, u_6, u_7, u_9, u_{10}\}, \{u_3\}, \{u_8\}\}$,此时 $n_1 = 8, n_2 = 1$,

$n_3 = 1$,则 $W(p) = 8 \times (1+1) + 1 \times 1 = 17$;

$U//\{c\} = \{\{u_1, u_2, u_4, u_5, u_6, u_7, u_8, u_9\}, \{u_3, u_{10}\}\}$,此时 $n_1 = 8, n_2 = 2$,则

$W(c) = 8 \times 2 = 16$ 。

此时 $|M| = 0$ 。

步骤3 $A_R = C - RED = \{m, w, p, c\} - \{m\} = \{w, p, c\}$,由上可知： $W(c) < W(p) < W(w)$,可见 c 的知识量最小 ,将其从 M 中的每一项中删除 ,这时没有出现只有一个元素的项。

步骤4 计算 $W_{RED}(w)$ 和 $W_{RED}(p)$:

$U//\{m, w\} = \{\{u_1, u_4\}, \{u_2, u_5\}, \{u_3\}, \{u_6, u_9\}, \{u_7, u_{10}\}, \{u_8\}\}$,此时 $n_1 =$

$2, n_2 = 2, n_3 = 1, n_4 = 2, n_5 = 2, n_6 = 1$,则 $W(m, w) = 2 \times 8 + 2 \times 6 + 1 \times 5 + 2 \times 3 + 2 \times 1 = 41$ 。

$W_{RED}(w) = W_{(m)}(w) = W(m, w) - W(w) = 41 - 25 = 16$ 。

同理 $U//\{m, p\} = \{\{u_1, u_4, u_6, u_9\}, \{u_3\}, \{u_5, u_7, u_8, u_{10}\}, \{u_2\}\}$,此时

$n_1 = 4, n_2 = 1, n_3 = 4, n_4 = 1$,则 $W(m, p) = 4 \times 6 + 1 \times 5 + 4 \times 1 = 33$ 。

$W_{RED}(p) = W_{(m)}(p) = W(m, p) - W(p) = 33 - 25 = 8$ 。

由上可知 $W_{(m)}(p) < W_{(m)}(w)$,可见此时应选择属性 p ,将其从 M 中含 p 的每一项中删除 ,这时出现了只有一个元素的项 w ,将 M 中包含 w 的项删除 ,并将 w 加入到约简中 ,得到当前的约简为 $\{m, w\}$ 。

步骤5 继续计算得到 $|M| = 0$,于是停止计算 ,输出 $\{m, w\}$ 。集合 $\{m, w\}$ 为所求的约简。

4 结束语

本文对粗集理论进行了讨论 ,在分析基于粗集理论的属性约简算法的基础上 ,提出了新的改进算法。本文算法以知识量作为启发式信息 ,每次删除知识量小的属性 ,直到找到约简为止。分析及实例表明本文提供的方法和技术是可行的。它在能够很有效地进行系统的约简 ,并且能找到系统的较优约简 ,但不一定能够找到系统的最优约简。在未来工作中 ,可以用本算法与能够找到系统最优约简的算法相结合 ,在时间允许的范围内去找最优约简。目前 ,粗集中属性约简得到各行业的广泛应用 ,该方法虽然能挖掘到很多有价值的信息 ,为人们科学决策提供帮助 ,但仍有不少问题有待于进一步研究。

参考文献:

- [1] 王国胤.粗糙集理论与知识获取[M].西安:西安交通大学出版社,2000.
- [2] Liang Jiye. The algorithm on knowledge reduction in incomplete information systems[J]. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2002, 10(1): 95-103.
- [3] Zhao Kai, Wang Jue. A reduction algorithm meeting users' requirements[J]. J Compute & Technol, 2002, 17(5): 578-593.
- [4] Yao Y Y. Information granulation and rough set approximation [J]. International Journal of Intelligent systems, 2001, 16(1): 87-104.
- [5] 杜金莲,迟忠先,翟巍.基于属性重要性的逐步约简算法[J].小型微型计算机系统,2003,24(6):976-978.
- [6] 徐燕,怀进鹏.基于区分能力大小的启发式约简算法及其应用[J].计算机学报,2003,26(1): 1-7.
- [7] 胡可云.基于概念格和粗糙集的数据挖掘方法研究[D].北京:清华大学,2001:64-70.
- [8] 陶志,许宝栋,汪定伟.一种基于分明矩阵的启发式知识约简方法[J].系统工程与电子技术,2005(4):734-736.
- [9] 亿珍珍,赵克,许威.一种基于 Rough 集的知识库冗余性化简研究[J].计算机工程与设计,2004,25(10):1731-1733.

(上接第 3431 页)

- [4] Sun J, Xu W B. A global search strategy of quantum-behaved particle swarm optimization [C]. Singapore: Proceeding of IEEE Conference on Cybernetics and Intelligent Systems, 2004: 111-116.
- [5] 曾建潮,介婧,崔志华.微粒群算法[M].北京:科学出版社,2004: 104-105.
- [6] Yin P Y. A discrete particle swarm algorithm for optimal polygonal approximation of digital curves [J]. Journal of Visual Communication and Image Representation, 2004, 15(2): 241-260.
- [7] Jiang C W, Bompard W. A hybrid method of chaotic particle swarm optimization and linear interior for reactive power optimization[J]. Mathematics and Computers in Simulation, 2005, 68(1): 57-65.
- [8] Salman A, Ahmad I, Madani S A. Particle swarm optimization for task assignment problem[J]. Microprocessors and Microsystems, 2002, 26(8): 363-371.