

求解矩形包络问题的量子行为粒子群优化算法

薛迎春, 孙 俊, 须文波

(江南大学 信息工程学院, 江苏 无锡 214122)

(xueyingchun1982@163.com)

摘 要:介绍了一种利用量子行为粒子群算法(QPSO)求解矩形包络的方法。矩形包络是将二维不规则形状样片用它们的最佳包络矩形来代替,是服装排料的第一步。实验结果表明量子行为粒子群算法比粒子群算法,遗传算法能更好地解决求二维不规则形状样片的矩形包络的问题。

关键词:矩形包络;量子行为粒子群算法;粒子群算法;遗传算法

中图分类号:TP391.41 **文献标识码:**A

QPSO algorithm for rectangle-packing optimization

XUE Ying-chun, SUN Jun, XU Wen-bo

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

Abstract: An improved QPSO (Quantum-behaved Particle Swarm Optimization) to solve rectangle-packing problems was proposed. The rectangle-packing is to replace two-dimensional irregular objects with their best rectangle, which is the first step of the clothing layout. The experimental results show that QPSO is better at solving the layout problem than PSO and GA (Genetic Algorithm).

Key words: rectangle-packing; QPSO (Quantum-behaved Particle Swarm Optimization); PSO (Particle Swarm Optimization); GA (Genetic Algorithm)

0 引言

在服装工业生产中,对于一匹布料和需要裁剪下来的衣片,要将这些衣片可以根据实际要求进行一定旋转且互相不覆盖的条件下,使得布料的使用率最高,整个布局的过程称为马克制造,产生的一个布局就称为一个马克。在实际的生产中产生的马克可能并不是最优的布局,但一般要求是接近最优的,实际的衣片形状则可以是任意的。计算机服装辅助设计的马克制造第一步就是求任意衣片的矩形包络^[1]。

本文的主旨是将比较先进的 QPSO 算法^[2,3]应用于排料第一步求解矩形包络,从实验结果来看,在排料优化的求解矩形包络中 QPSO 优于 PSO 和 GA。

1 算法介绍

1.1 PSO 算法

PSO 算法由 James Kennedy 和 Russell Eberhart 共同提出的,源于对鸟群捕食的行为研究,并利用了生物学家 Frank Heppner 的生物群体模型。PSO 中,每个优化问题的解都是搜索空间中一只鸟。我们称之为“粒子”。

所有的粒子都有一个由被优化的函数决定的适应值(fitness value),每个粒子还有一个速度决定他们飞翔的方向和距离。然后粒子们就追随当前的最优粒子在解空间中搜索。PSO 初始化为一群随机粒子(随机解),然后通过迭代找到最优解。在每一次迭代中,粒子通过跟踪两个“极值”来更新自己。第一个就是粒子本身所找到的最优解,这个解叫作个体极值 pbest;另一个极值是整个种群目前找到的最优解,

这个极值是全局极值 gbest。另外也可以不用整个种群而只是用其中一部分最好粒子的邻居,那么在所有邻居中的极值就是局部极值。在找到这两个最优值时,粒子根据如下的公式来更新自己的速度和新的位置。

$$V_i(t+1) = V_i(t) + c_1 \text{rand}_1(P_p(t) - X_i(t)) + c_2 \text{rand}_2(P_g(t) - X_i(t)) \quad (1)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (2)$$

rand_1 和 rand_2 是介于(0,1)之间的随机数, c_1, c_2 是加速常数,通常在 0 ~ 2 间取值。 c_1 调节微粒飞向自身最好位置方向的步长, c_2 调节微粒向全局最好位置飞行的步长。为了减少在进化过程中,微粒离开搜索空间的可能性,微粒的速度被限定在 $[-V_{\max}, V_{\max}]$ 内。 P_p 是先前定义的个体极值 pbest。 P_g 是先前定义的全局极值 gbest。

文献[6]对(1)式作了如下的改动:

$$V_i(t+1) = w * V_i(t) + c_1 \text{rand}_1(P_i(t) - X_i(t)) + c_2 \text{rand}_2(P_g(t) - X_i(t)) \quad (3)$$

其中 w 是非负数,称为惯性权重。当 $w = 1$ 时,(3)式和基本粒子群算法的速度进化方程一样,文献[5]建议 w 的取值范围为 $[0, 1.4]$ 。而当 $w > 1.2$ 时算法则较多地陷入局部极值。

1.2 QPSO 算法

PSO 是基于种群的进化搜索技术,但它不是一个能保证全局收敛的算法^[8]。因为 PSO 的进化方程式使所有粒子在一个有限的样本空间中搜索。根据粒子群的基本收敛性质,受量子物理基本理论的启发, Sun 等人提出的 QPSO^[2,3] 算法是对整个 PSO 算法进化搜索策略的改变,并且进化方程中不需要速度向量,而且进化方程的形式更简单,参数更少且更容

收稿日期:2006-03-29; 修订日期:2006-07-02 基金项目:国家自然科学基金资助项目(60474030)

作者简介:薛迎春(1982-),女,吉林长春,硕士,主要研究方向:CAD 辅助设计、图形图像处理; 孙俊(1971-),男,江苏无锡,博士研究生,主要研究方向:人工智能; 须文波(1946-),男,江苏无锡人,教授,博士生导师,主要研究方向:人工智能、计算机控制、嵌入式操作系统、并行计算、模式识别。

易控制。QPSO 算法,在搜索能力上优于所有已基本的或改进的 PSO 算法。

为了保证算法的收敛性,每一个粒子必须收敛于各自的 p 点, $p = (p_1, p_2, \dots, p_d)$, 第 i 个粒子 p 点的第 d 维坐标为:

$$p_d = (\varphi_1 * p_{id} + \varphi_2 * p_{gd}) / (\varphi_1 + \varphi_2) \quad (4)$$

其中 $\varphi_1 = \text{rand}(0,1)$, $\varphi_2 = \text{rand}(0,1)$ 或者 $p_d = \varphi * p_{id} + (1 - \varphi) * p_{gd}$, $0 < \varphi < 1$ 。

我们在粒子群中引入了一个全局点 $mbest$ 来计算粒子的下一迭代步的变量 L , 它定义为所有粒子的局部最好位置的平均值。公式如下:

$$mbest = \frac{1}{M} \sum_{i=1}^M P_i = \left[\frac{1}{M} \sum_{i=1}^M P_{i1}, \dots, \frac{1}{M} \sum_{i=1}^M P_{id} \right] \quad (5)$$

其中 M 是粒子的个数, p_i 是粒子 i 的局部最好位置。于是粒子的迭代方程变为:

$$x(t+1) = p \pm \beta * |mbest - x(t) * \ln(1/u)| \quad (6)$$

其中 $u = \text{rand}(0,1)$, β 被称为收缩扩张系数, 调节它的值能控制算法的收敛速度。一般而言, β 值在算法运行是从 1.0 线性减小到 0.5 时, 可以达到比较好的效果, 即

$$\beta = (1.0 - 0.5) * (MAXITER - T) / MAXITER + 0.5;$$

其中 $MAXITER$ 是迭代的最大次数。式(6)被称为具有量子行为的粒子群算法(简称 QPSO)。

QPSO 的算法流程为:

初始化种群的每个粒子的位置向量

Do

For $i = 1$ to 粒子数大小 M

计算粒子的目标函数值 $f(x)$

If $f(x_i) < f(p_i)$ then $p_i = x_i$

$p_g = \min(p_i)$ 运用公式(5)找到全局点 $mbest$

For $d = 1$ to 维数 D

计算每个粒子的随机点

$\varphi_1 = \text{rand}(0,1)$, $\varphi_2 = \text{rand}(0,1)$

$p = (\varphi_1 * p_{id} + \varphi_2 * p_{gd}) / (\varphi_1 + \varphi_2)$

$u = \text{rand}(0,1)$

更新每个粒子的位置[公式(6)以 0.5 的概率取加、以 0.5 的概率取减]

if $\text{rand}(0,1) > 0.5$

$$x_{id} = p - \beta * |mbest - x(t)| * \ln(1/u)$$

else

$$x_{id} = p + \beta * |mbest - x(t)| * \ln(1/u)$$

直到终止条件满足(说明:针对每一维的位置坐标算法流程中的 u 是不同的随机数与 if $\text{rand}(0,1) > 0.5$ 中的 $\text{rand}(0,1)$ 不同不能替换)。

在 QPSO 中,粒子的状态只须用位置向量来描述,并且算法中只有一个收缩扩张系数 β , 对这个参数的选择和控制是非常重要的,它关系到整个算法的收敛性能。

1.3 遗传算法

生物进化是一个奇妙的优化过程,它通过选择淘汰、突然变异、基因遗传等规律产生适应环境变化的优良物种。遗传算法 GA (genetic algorithm) 就是根据生物进化思想而启发得出的一种全局优化算法。其基本原理就是把问题的解表示成染色体(简称个体),在算法中表示成二进制编码串,在算法开始时给出一群染色体(称为初始种群),即一组假设解,然后把假设解置于问题的环境中,计算每个个体的适应度值,按适者生存的原则从中选择适应环境能力强的个体进行交叉、

变异、突变过程,产生适应能力更强的新一代种群。如此一代代进化,最后将收敛到一个染色体上,即得到问题的最优解。

2 多边形

2.1 任意多边形面积

由于多边形是任意的,计算不规则的多边形的面积比较困难。我们采用的方法是将不规则多边形分解成一组三角形,而三角形面积由顶点坐标所规定,因此不规则多边形的面积 S 可按如下的公式计算:

$$S = \frac{1}{2} \left| \sum_{k=1}^n (X_k Y_{k+1} - X_{k+1} Y_k) \right|$$

其中, n 为顶点个数, X, Y 为各顶点坐标。图 1 中的多边形的面积为 5。

2.2 多边形的凸壳

排料时寻找不规则多边形外接矩形的方法往往是先找到不规则多边形各个顶点横坐标和纵坐标的最大值,由此求出不规则多边形的外接矩形,这样求出的外接矩形并不是面积最小的外接矩形,因为找到不规则多边形的最小外接矩形对于提高原材料的利用率至关重要,所以必须寻求新的方法求不规则多边形的矩形包络。

对于任意的多边形来讲,可能是凸的、凹的或者是有边交叉的,这使得求其外接矩形非常复杂,对于任意不规则多边形,第一步就是求其“凸壳”,“凸壳”是计算几何中最普遍、最基本的一种结构,不仅自身有许多特性,而且还是构造其他几何形体的有效形态。在应用中,许多实际问题可以归结为“凸壳”问题。

设 S 是平面(E^2)中的点集。用 $CH(S)$ 表示点集 S 的凸壳, $BCH(S)$ 表示 S 的凸壳边界:

定义 设 S 是平面上的非空点集, p_1, p_2 是 S 中任意两点,如果 $p = t \times p_1 - (1 - t) \times p_2$ 属于 S , 其中 $0 \leq t \leq 1$, 则称 S 是凸集。这就是说,如果 S 中任意两点所连线段全部位于 S 中,那么 S 是凸的。

1972 年,格雷厄姆发表了一篇题为“An efficient algorithm for determining the convex hull of a finite planer set”的著名论文。这是计算几何领域中具有重要意义的早期卓越成果,文中所提出的方法称为格雷厄姆方法。根据凸多边形的定义,凸多边形的各顶点必须在该多边形任意一条边的同一侧。这是格雷厄姆方法的依据。此方法步骤如下:

(1) 设凸集中 y 坐标最小的点为 p_1 , 把 p_1 同凸集中其他各点用线段连接,并计算这些线段与水平线的夹角。然后按夹角大小及到 p_1 的距离进行词典式分类,得到序列 p_1, p_2, \dots, p_n 依次连接这些点,便得到一个多边形,其中 p_1 是凸壳边界的起点, p_2 与 p_n 也必是凸壳顶点。

(2) 删去 p_1, p_2, \dots, p_n 中不是凸壳顶点的点。方法如下:

begin

① $k = 4$

② $j = 2$

③ if p_1 和 p_k 分别在 $p_{k-j-1}p_{k-j}$ 两侧
then 删去 p_{k-1} , 后继顶点编号减 1,
 $k = k - 1, j = j - 1$

else

p_{k-1} 暂为凸壳顶点,并记录

④ $j = j + 1$, goto ③, 直至 $j = k - 1$

⑤ $k = k + 1$, goto ②, 直至 $k = n + 1$

End

(3) 顺序输出凸壳顶点

由于点集中有 n 个点, 步骤(2) 中转移到2的次数不超过 n , 每个顶点至多被删去一次, 则删去顶点的个数也不可能超过 n . 因此步骤(2) 需要线性时间. 步骤(1) 要计算 $n-1$ 个夹角, 并按夹角分类, 计算每个夹角只需要常数时间, 计算 $n-1$ 个夹角耗费线性时间, 分类要时间 $O(n \log n)$. 因此格雷厄姆方法的时间复杂性为 $O(n \log n)$. 经过论证: 格雷厄姆方法是求解平面点集凸壳问题的最佳算法. 求凸壳前后的结果如图1和图2所示.

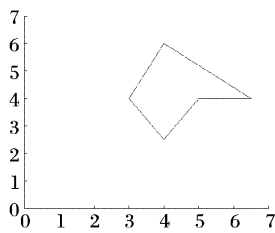


图1 求凸壳前

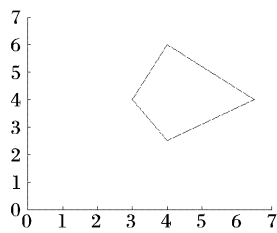


图2 求凸壳后

3 矩形包络

矩形包络法是将外形各异的样片用它们的最佳包络矩形来代替, 从而把不同样片的排料问题最终归结为大小不等的矩形优化布局问题. 最佳包络矩形就是包络多边形且面积最小的矩形.

3.1 代数距离

过原点作直线后, 多边形的顶点分布在直线的两边, 引入代数距离的概念, 我们规定将位于直线上方的点到直线的距离为正距离, 位于直线下方的点到直线的距离为负距离, 然后求出顶点到直线的距离并进行排序, 找出最大距离和最小距离记为 d_{\max}, d_{\min} .

3.2 点在直线上的投影

将凸多边形的各个顶点向过原点的直线做投影, 原始坐标为 x, y 投影后的坐标为 x', y' , 过原点的直线的与 x 轴的夹角为 θ , 则投影点的坐标公式为:

$$x' = \cos(\theta) \times \sin(\theta) \times y + \cos(\theta)^2 \times x$$

$$y' = \sin(\theta)^2 \times y + \cos(\theta) \times \sin(\theta) \times x$$

在投影坐标中找出横坐标值最大和最小的点, 如果横坐标都相同则找纵坐标值最大和最小的点, 这两个点记为 p_{\max}, p_{\min} . 如图3所示.

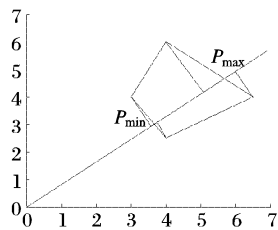


图3 矩形包络求解过程

3.3 矩形四个顶点的确定

投影后点 p_{\max}, p_{\min} 分别移动 d_{\max}, d_{\min} 得到四个点, 以该四点为顶点的矩形就是多边形在该角度下的矩形包络.

利用 QPSO 求多边形的矩形包络就是寻找过原点的直线斜率, 使得依此直线为基准确定的多边形外接矩形最大限度地接近该多边形的矩形包络, 具体算法流程如下:

① 在解空间中随机初始化群体大小为 M 的各个粒子位置坐标信息(即过原点的直线斜率), 并由此求出各个粒子的适应性值信息(即外接矩形的面积). 初始化各个粒子局部最优位置信息(即: 位置坐标和适应性值)为当前的位置信息,

寻找全局最优位置的位置信息.

② 根据公式(4)(5)(6) 更新粒子的位置坐标信息.

③ 修改局部最优点: 通过评价函数评价每个粒子的适应值, 若粒子当前适应值小于它当前最佳位置的适应值则用当前位置信息替换当前最佳位置信息(p_{best}).

④ 修改全局最优点: 对每个粒子, 比较其当前最佳位置的适应值与全局最佳位置的适应值, 若当前最佳位置适应值小则用当前最佳位置信息替换全局最佳位置的信息.

⑤ 判断终止条件是否满足, 若不满足返回 ②, 否则算法结束(算法终止条件采用最大迭代次数).

4 实验结果分析

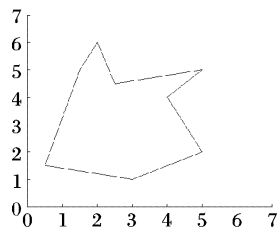


图4 实验所用第二个多边形

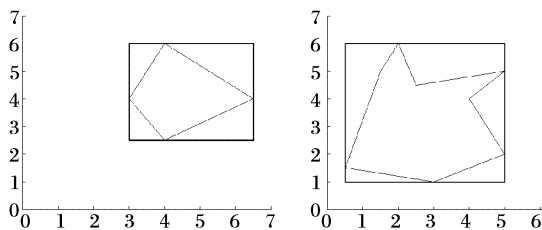
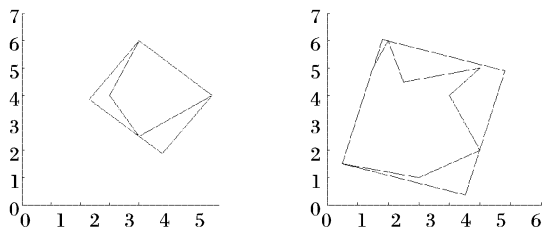
图5 与 x 轴夹角为 0 的矩形包络

图6 面积最小的矩形包络

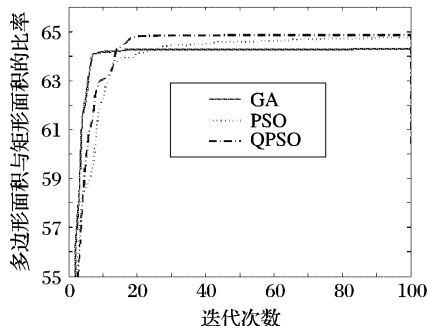


图7 多边形(1)与矩形面积的比率和迭代次数关系图

选取图1中的多边形和图4的多边形(该多边形的面积为12.875)进行实验, 当直线与横坐标的夹角为0时结果如图4所示, 分别用 GA, PSO, QPSO 算法迭代1000次后结果差距很小, 各算法得到的结果如图6所示. 图5所示的第一个矩形面积为12.25, 多边形占矩形面积的40.82%, 第二个矩形面积为22.5, 多边形占矩形面积的57.2% 而在图6中所示的第一个矩形面积为8.75, 多边形占矩形面积的57.14%, 第二个矩形面积为19.85, 多边形占矩形面积的64.85%.

图7、图8三条曲线分别是求解矩形包络时用QPSO, PSO (下转第2073页)

特征,但是得到的结果却和使用全部像素的特征得到的结果相当,显示该算法在一定程度上避免了冗余计算。同时,由粗到精的多尺度分割算法也比较有效地减少了分割结果图中由特征估计的误差而造成的无意义小区域的数量,从而改善了分割效果。

表1 不同算法分割的时间代价对比

分割结果编号	(b)	(c)	(d)	(e)	(f)	(g)	(h)
约耗时/s	7.266	2.984	7.860	5.781	51.250	1.375	0.640

表1给出了在上述对比实验中,各个算法运行所用的时间。实验所使用的硬件为迅驰 M 1.86GHz 的 CPU 和 1G 的内存。从表1中可以看出,相对于现有的纹理特征估计算法,估计形态学多重分形指数特征的时间复杂度并不算大,通过使用基于四叉树的多尺度分割,计算时间进一步减少了超过一半,远远低于其他五种方法的耗时。这充分显示出本文所提出的分割算法在处理数据量庞大的遥感图像时的优势。

4 结语

本文提出的形态学多重分形估计算法有效结合了形态学分形维数和多重分形分析的优点,能够更加准确而全面的刻画图像的局部尺度特性。本文构造的基于图像四叉树的多尺度分割算法,通过由粗到精的逐级分割过程不但保持了分割的精度,而且大大降低了分割算法的时间复杂度。该算法虽然是针对多重分形特征设计的,但也被应用于基于其他特征的图像分割算法。对比实验表明,本文所述的遥感图像粗分割算法可以用相对较低的计算代价得到更有效的分割结果。

参考文献:

- [1] MANDELBROT BB. How long is the coast of Britain? Statistical self-similarity and fractal dimension[J]. Science, 1967, 156: 636

-638.

- [2] PENTLAND AP. Fractal based description of natural scenes[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1984, 6(6): 661-674.
- [3] SARKAR N, CHAUDHURI BB. Texture segmentation using fractal dimension[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1995, 17(1): 72-77.
- [4] FENG J, LIN WC, CHEN CT. Fractal box-counting approach to fractal dimension estimation[A]. Proceedings of ICPR'96[C]. Vienna, 1996. 854-858.
- [5] DU G, YEO TS. A novel multifractal estimation method and its application to remote image segmentation[J]. IEEE Transactions on Geoscience Remote Sensing, 2002, 40(4): 980-982.
- [6] 李厚强, 刘政, 凯林峰. 基于分形理论和 Kohonen 神经网络的纹理图像分割方法[J]. 计算机工程与应用, 2001, 37(7): 44-46.
- [7] HARALICK RM, SHANMUGAM K, DINSTEN J. Texture features for image classification[J]. IEEE Transactions on System, Man and Cybernetics, 1973, 3(1): 610-621.
- [8] LAWS K. Textured Image Segmentation[R]. Technical Report, USCIPR Report 940. Los Angeles: University of Southern California, 1980.
- [9] MANJUNATH BS, CHELLAPPA R. Unsupervised texture segmentation using Markov random fields [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1991, 13(3): 478-482.
- [10] 夏勇, 赵荣椿, 江泽涛. 一种基于数学形态学的分形维数估计方法[J]. 中国图象图形学报, 2004, 9A(6): 760-766.
- [11] 章毓晋. 图像分割[M]. 北京: 科学出版社, 2001.
- [12] BEZDEK JC. Pattern recognition with fuzzy objective function algorithms[M]. Plenum Press, New York, 1981.

(上接第 2070 页)

和 GA 算法求得的多边形与矩形包络面积比率和迭代次数关系图,曲线表明算法都是收敛的。两张图都表明 QPSO 算法在收敛速度上要比 PSO 算法的收敛的速度要快并且形成的矩形包络的面积也比 PSO 算法的产生的矩形包络的面积小。QPSO 算法和 GA 算法相比,虽然在收敛速度上要比 GA 算法稍慢,但 QPSO 算法的优化结果要比 GA 算法好(即:产生的矩形面积小于 GA 算法产生的矩形)。

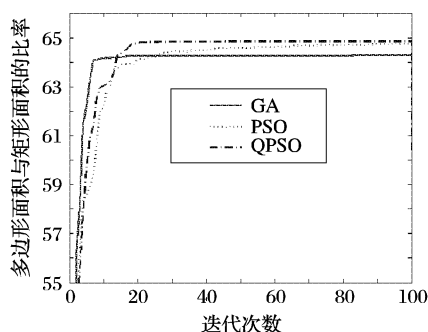


图8 多边形(2)与矩形面积的比率和迭代次数关系图

5 结语

矩形包络问题无论是在实际工程应用中还是在科学研究中都有十分重要的意义,本文对采用 QPSO 算法求解矩形包

络问题进行了研究,并比较了基于 QPSO 算法、PSO 算法和 GA 算法求解矩形包络问题的结果。比较表明在求解矩形包络问题时 QPSO 算法的收敛能力优于 PSO 算法,虽然速度略低于 GA 算法但收敛结果优于 GA 算法,由此表明 QPSO 算法在求解矩形包络问题领域具有较高的使用价值。

参考文献:

- [1] Orsini A. A Two Stage Solution of the Cutting Problem[A]. Information Processing, 17 North Halland[C], 1976. 3-14.
- [2] SUN J, XU WB. A Global Search Strategy of Quantum-behaved Particle Swarm Optimization[A]. Proceedings of IEEE conference on Cybernetics and Intelligent Systems[C], 2004. 111-116.
- [3] SUN J, FENG B, XU WB. Particle Swarm Optimization with Particles Having Quantum Behavior[A]. Proceedings of 2004 Congress on Evolutionary Computation[C], 2004. 325-331.
- [4] KENNEDY J, EBERHART RC. Particle Swarm Optimization[A]. Proceedings of the IEEE International Joint Conference on Neural Networks[C], 1995. 1942-1948.
- [5] SHI Y, EBERHART RC. A Modified Particle Swarm Optimization [A]. Proc IEEE International Conference on Evolutionary Computation[C], 1998. 1945-1950.
- [6] 曾建潮, 介婧, 崔志华. 微粒群算法[M]. 北京: 科学出版社, 2004. 104-105.