

基于像素层标识点的双边滤波快速实现

郑丽萍 李俊青 于承敏 张民
(聊城大学计算机学院 聊城 252059)

摘要 双边滤波能在去除噪声的同时有效地保留图像的边缘信息。但双边滤波的时间复杂度高,执行时间长。根据近似层和亮度分层的概念,利用标识点及像素层来快速实现双边滤波。首先根据灰度差值划分图像的像素层,然后在像素层上选择标识点,并利用标识点计算像素层的滤波值,最后通过线性插值获得各像素点的滤波值,并输出滤波图像。该改进算法称为标识点双边滤波(Identification Bilateral Filtering)。在实验中分别对灰度和彩色图像进行了双边滤波。实验结果表明,IBF算法执行时间短,并能获得较好的滤波效果。

关键词 双边滤波,像素层,标识点,欧氏距离
中图分类号 TP391 **文献标识码** A

Fast Implementation of Bilateral Filtering with Identification Point in Pixel Layer

ZHENG Li-ping LI Jun-qing YU Cheng-min ZHANG Min
(College of Computer Science, Liaocheng University, Liaocheng 252059, China)

Abstract Bilateral filtering is a technique to delete images noise while effectively preserving edges. The naïve implementation of the bilateral filtering can be extremely slow. The time complexity is high. According to concepts of approximate layer and intensity layer, a improved bilateral filtering was proposed. This improved algorithm uses identification and pixel layer to realize Bilateral Filtering. This improved algorithm is called identification Bilateral Filtering (IBF). At first, the pixel layer with gray D-value was specified, then identification point in pixel layer was chosen, and the filtering value of every pixel layers with identification point was computed. At last, linear interpolation was used to compute filtering value of pixels and output filtered image. Gray image and color image were taken as research objects in experiment. Experiment results show that the IBF algorithm has short executing time and has a good filtering result.

Keywords Bilateral filtering, Pixel layer, Identification point, Euclidean distance

1 引言

由于成像系统、传输介质和记录设备等的不完善,数字图像在形成过程中会受到多种噪声的污染。在对图像进行压缩编码和传输之前,必须滤除噪声,并保持图像信息的完整。如何快速有效地去除各种噪声,并完整地保持目标的形状、大小及相关几何拓扑特征十分重要。然而,噪声往往并存,单纯采用一种滤波方法达不到满意效果,因此需要多种滤波方法相结合。

双边滤波(Bilateral Filtering)是以经典高斯滤波算法为基础并利用 Overton 等人的思想设计出的一种非迭代的非线性滤波方法^[1]。它是一种保边去噪的滤波方法,应用范围广。但原始的双边滤波算法计算量大,因此如何降低双边滤波的时间复杂度,减少运算时间是需要考虑的一个问题。一些学者已经提出了一些双边滤波的快速实现算法,如 Pham 的分离核法^[2]、Weiss 的局部直方图法^[3]、Durand 的近似层法^[4]、还有双边网格^[5]、双边金字塔方法^[6]等。这些改进算法的时

间复杂度都比原始双边滤波算法的时间复杂度低。

本文以近似层算法和亮度分层为基础,对图像进行分层,形成多个像素层,并利用像素层上的标识点,实现图像的快速双边滤波。该快速双边滤波算法称为标识点双边滤波(Identification Bilateral Filtering, IBF)。

2 双边滤波的基本原理

双边滤波是一种既可有效降低图像噪声,又可保持图像边缘细节的滤波技术。它由 Tamasi 等人提出,能同时利用邻域内像素点的空间邻近度信息和灰度邻近度信息。双边滤波是一种局部加权平均,其加权系数由两个因子决定^[7]。一个是空间权系数因子(空间邻近度因子) σ_s ,一个是灰度权系数因子(灰度邻近度因子) σ_r 。 σ_s 控制像素间的空间距离,其大小决定窗口中高斯函数所包含的像素个数。 σ_s 变大时,作用的像素变多,导致图像变得越模糊,而 σ_r 可以对 σ_s 的变化做出补偿。 σ_r 控制像素间的灰度差值,灰度差值大于 σ_r 的像素之间将不进行运算,保留了图像高频边缘的灰度信息。

到稿日期:2012-09-12 返修日期:2012-12-24 本文受国家自然科学基金项目(61104179),山东省教育厅资助项目(J11LG02, J10LG67),聊城大学自然科学基金(X09031),山东省高校智能信息处理与网络安全重点实验室项目资助。

郑丽萍(1979-),女,博士,副教授,主要研究方向为图像处理、智能算法, E-mail: zhengliping80@163.com; 李俊青(1977-),男,硕士,副教授,主要研究方向为智能调度算法; 于承敏(1972-),女,硕士,副教授,主要研究方向为图像处理; 张民(1972-),女,硕士,副教授,主要研究方向为智能算法。

双边滤波的窗口为矩形,其形式化定义见式(1)和式(2),其中 w_p 是灰度权值和空间权值乘积的加和, BF_p^I 表示图像 I 中像素点 p 的双边滤波结果。 I_p 为像素点 p 的灰度值, $\|p-q\|$ 表示像素点 p 和像素点 q 之间的欧氏距离。 S 表示滤波窗口的范围。

$$BF_p^I = \frac{1}{w_p} \sum_{q \in S} f(\|p-q\|) g(|I_p - I_q|) I_q$$

$$= \frac{1}{w_p} \sum_{q \in S} G_{\sigma_s}(\|p-q\|) G_{\sigma_r}(|I_p - I_q|) I_q \quad (1)$$

$$w_p = \sum_{q \in S} G_{\sigma_s}(\|p-q\|) G_{\sigma_r}(|I_p - I_q|) \quad (2)$$

在式(1)中, G_{σ_s} 和 G_{σ_r} 分别为空间邻近度函数和灰度邻近度函数,其形式为高斯函数,如式(3)~式(5)所示。

$$G_{\sigma}(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right) = 1/e^{-\frac{x^2}{2\sigma^2}} \quad (3)$$

$$f(\|p-q\|) = G_{\sigma_s}(\|p-q\|) = 1/e^{-\frac{(\|p-q\|)^2}{2\sigma_s^2}} \quad (4)$$

$$g(|I_p - I_q|) = G_{\sigma_r}(|I_p - I_q|) = 1/e^{-\frac{(|I_p - I_q|)^2}{2\sigma_r^2}} \quad (5)$$

某像素点的双边滤波过程如图1所示。

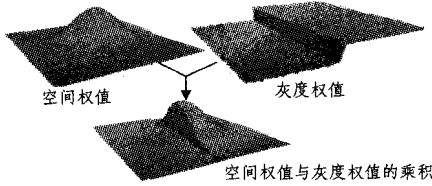


图1 双边滤波(Bilateral Filtering)

双边滤波是利用高斯核的卷积来实现的,可解决高斯滤波后图像边缘模糊的问题。 p 点的滤波值主要被空间相邻并且灰度相似的像素影响。进行滤波时,根据具体情况,设置相应的 σ_s 和 σ_r 。利用函数 G_{σ_s} 和 G_{σ_r} 的乘积加和作为双边滤波的权值,可以减小远距离像素和不同灰度值像素对像素 p 的影响。双边滤波的应用很多,如高动态范围色调映射、对比度增强^[8,9]、3D网格去噪等^[10-13]。

3 快速双边滤波的实现流程

双边滤波的主要问题在于执行时间长,时间复杂度高。本文借鉴文献[4]中的灰度近似层概念和文献[8]中的亮度分层概念,利用像素层的标识点来完成图像的双边滤波。

首先根据灰度差值参数 σ_r 划分图像的像素层,把一幅图像转换成一组图像序列。对每个像素层,选择标识点来简化滤波值的计算,获得各像素层的滤波值,然后计算像素点的滤波值,获得最终的滤波图像。主要流程如图2所示。

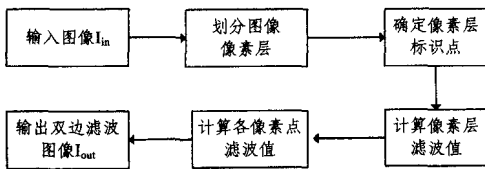


图2 双边滤波快速实现流程

4 基于图像标识点的双边滤波

4.1 图像像素层的划分

一幅图像的灰度值一般都分布在一定范围 $[0, N-1]$ 内, N 为图像灰度级。图像的直方图能直观地显示图像的灰度

分布。在进行滤波时,像素间的灰度差值 σ_r 决定滤波的结果。对于图像 I ,某像素点 p 的灰度值 I_p 在灰度区间 $[\min(I), \max(I)]$ 中,根据 σ_r 的值,把整幅图像的灰度范围分为 n 段,且 $n = \lfloor (\max(I) - \min(I)) / \sigma_r \rfloor$,每个灰度段有两个边界值。通过划分,原始图像被转换成 $(n+1)$ 个中间图像。每幅中间图像的灰度值与某一灰度区间的边界值相同。每幅中间图像就是一个像素层,原始图像中的像素分布在像素层及像素层之间。

4.2 像素层的双边滤波

在计算某像素点 p 的滤波值时,由式(1)可知, $G_{\sigma_r}(|I_p - I_q|)$ 的值依赖于 I_p ,所以选择代表性的灰度值 ' k ' 来代替 I_p ,计算 $G_{\sigma_r}(|k - I_q|)$,使像素点的滤波值计算公式变为:

$$BF_p^I = \frac{1}{w_p} \sum_{q \in \Omega} f(\|p-q\|) g(|I_p - I_q|) I_q$$

$$= \frac{1}{w_p} \sum_{q \in \Omega} G_{\sigma_s}(\|p-q\|) G_{\sigma_r}(|k - I_q|) I_q \quad (6)$$

在本文中, ' k ' 表示相应的像素层灰度值。利用式(6)可计算出每个像素层的双边滤波值。

4.2.1 标识点的确定

在每一层,利用式(6)计算像素层滤波值,对于 $G_{\sigma_s}(\|p-q\|)$, p 的位置需要确定。如果为了简化计算而取 G_{σ_s} 为固定值,则归一化以后,该系数会互相抹掉,相当于只有一个灰度系数起作用,进行的还是单边滤波。

在保持两个权值系数都起作用并简化计算的情况下,本文在每个像素层都选择一个标识点,式(6)中的 p 即为标识点。在每个像素层,像素的位置分布是随机无规律的,且标识点的位置影响标识点与像素之间欧氏距离的平均值。

本文选取图像的中心点作为标识点,以简化滤波值的计算。设图像大小为 $M \times N$,则标识点的位置为 $(M/2, N/2)$;选择图像的中心点为标识点,这样对于某灰度层上所有的像素计算所得的 G_{σ_s} 值就各不相同,在实现双边滤波的同时,计算速度也能改善。

4.2.2 像素层滤波值的计算

原始图像 I 对应 $n+1$ 个像素层,其标号为 $\{l_0, l_1, \dots, l_n\}$,其中 l_0 层的灰度值为 $\min(I)$, l_n 层的灰度值为 $\max(I)$ 。对于每个 l_j , $l_j = l_0 + j\sigma_r$ ($j=0, 1, \dots, n$),式(6)变成式(7),计算相应的滤波值 BF_{l_j} , p 为每个像素层的中心点,获得相应的集合 $L = \{L_0, L_1, \dots, L_n\}$, L 是像素层滤波值的集合。对于原始图像 I ,可获得 $(n+1)$ 个滤波值。

$$L_j = BF_{l_j} = \frac{1}{w_p} \sum_{q \in \Omega} G_{\sigma_s}(\|p-q\|) G_{\sigma_r}(|l_j - I_q|) I_q \quad (7)$$

4.3 像素点的双边滤波

在计算每个像素的滤波值时,需要利用与像素邻近的像素层的滤波值进行线性插值来获得相应的滤波值,如式(8)所示。

设像素的灰度值为 l_k ,位于灰度区间 $[l_{j_1}, l_{j_2}]$ 之间,每个像素点的滤波值都用线性插值法来求,最终获得滤波图像并输出。

$$L_k = \frac{l_k - l_{j_1}}{l_{j_2} - l_{j_1}} L_{j_1} + \frac{l_{j_2} - l_k}{l_{j_2} - l_{j_1}} L_{j_2} \quad (8)$$

4.4 图像双边滤波的实现

用 Matlab 来编程实现所提出的算法。

伪代码: QuickBF

```
Range=input(σr); Room=input(σs);
%输入灰度参数和空间参数;
```

```

Image=imread('原始图像');%输入图像需要滤波的图像;
[M,N]=size(Image);%确定图像的维度及极值;
High=max(max(Image));
Low=min(min(Image));
n=(High-Low)/range;%确定图像的像素层数;
确定标识点 p 的位置(Xp,Yp);
For i=1:n+1
    S(i)=low+(i-1)*range;%确定像素层的灰度值;
     $L_i = BF_{S_i} = \frac{1}{w_{p,q} \in \Omega} \sum G_{\sigma_s}(\|p-q\|) G_{\sigma_r}(|S_i - I_q|) I_q$  %求每个像素层的滤波值;
End;
L={L0,L1,...,Ln} %形成像素层滤波值集合
For every pixel
     $L_k = \frac{l_k - l_{j_1}}{l_{j_2} - l_{j_1}} L_{j_1} + \frac{l_{j_2} - l_k}{l_{j_2} - l_{j_1}} L_{j_2}$ 
    %选取与像素点灰度值最相近的像素层,线性插值得得像素层的滤波值;
End
imshow('滤波图像')。

```

5 标识点的位置优化

在减少运行时间的同时,应尽量考虑像素分布因素,提高滤波效果。在本文中,考虑像素分布的随机性,标识点选在像素层的中心点,可取得一定的滤波效果。但如果遇到像素都分布在图像最下端的情况,滤波结果会不太好。因此,为了进一步提高滤波效果,对标识点的位置进行优化。根据图像的特点,选取 13 个关键坐标点,如图 3 所示,计算滤波值时根据像素 q 的位置,衡量 q 与 13 个坐标点的欧氏距离,选取距离最短的坐标点作为标识点,然后计算相应的滤波值。通过标识点的位置选择,可以改善像素分布极端情况下的图像滤波效果,使算法性能更完善。虽然计算像素点与 13 个坐标点的欧氏距离会增加计算量,但对保持滤波效果有一定的效果。

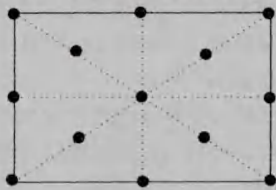


图 3 坐标点分布

6 实验及仿真

6.1 实验过程

在配置为 Intel(R) Core(TM) I5 CPU, 2.99GB 内存的计算机上用 Matlab 编程进行仿真实验。图像用矩阵表示,根据 σ_r 确定像素层的个数。先以灰度图像为实验对象,然后再处理彩色图像。对于彩色图像,用三维矩阵表示,滤波时先把彩色图像转换成 3 个不同的矩阵。按 IBF 算法进行滤波处理之后,再进行组合显示。

实验中先以图像的中心点为标识点进行实验。选择 3 幅不同的图像进行实验。对同一幅图像, σ_s 和 σ_r 选不同的值,对结果进行比较分析。图 4 为 3 幅原始图像。图 5 和图 6 分别为选择不同 σ_s 和 σ_r 值时快速双边滤波的结果。

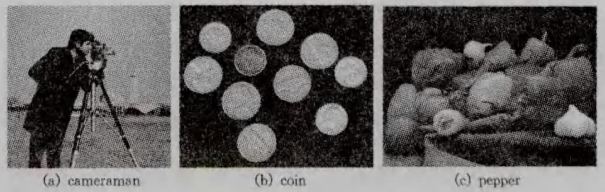


图 4 原始图像

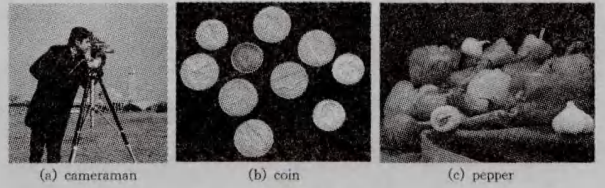


图 5 $\sigma_s=4$ 和 $\sigma_r=4$ 的滤波图像

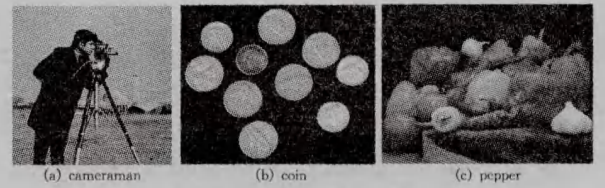


图 6 $\sigma_s=8$ 和 $\sigma_r=8$ 的滤波图像

6.2 实验结果分析

原始的双边滤波算法的执行时间与图像像素个数有关。设 Ω 是图像中像素的个数,则执行时间是 Ω^2 。而 IBF 算法所花时间大大减少,提高了运算效率。 σ_r 越大,所花时间越短。在实验中,原始的双边滤波算法处理 2^9 个像素所花时间约为 1.30594s,而 IBF 算法所花时间约为 0.3245s,运算时间大大提高。设取 σ_s 和 σ_r 为 4,处理 cameraman.tif 所花时间为 0.16781s,处理 coin 所花时间为 0.47203s,处理彩色图像 peppers.png 所用时间为 2.97297s。整个实验中,图像能获得较好的滤波结果,且运行时间比原始的双边滤波算法要短。

与近似层方法相比,该方法对像素层标识点的位置进行优化,使标识点的位置更合理,改善了图像的滤波效果。每个像素层近似滤波值的个数,使得像素可根据自己的位置来确定标识点的最优位置,图像的滤波效果更好。

在具体编程实现时,算法的整个执行可以有两种方法。一种是对每个像素点计算一次滤波值。另一种是先统计整个图像的灰度值,并计算每一个灰度值对应的滤波值;处理每个像素点时,先确定它的灰度值,再查找该灰度值所对应的滤波值,这样可以减少重复计算相同灰度值的像素点滤波值;但是在比较每个像素点的灰度值时,也需要一定计算量。存储图像的矩阵初值,如果是浮点型,则对图像的显示有影响,需要进行整数化。

结束语 灰度值相同的像素其周围的像素值不一定相同,滤波值也应该不同。而每个像素层却只有一个滤波值,因此造成灰度值相同的像素滤波值都相同。文中仅利用像素层和标识点并不能完全正确地计算出所有像素的滤波值。为进一步完善滤波效果,在将来的研究中,考虑利用像素的位置信息,计算像素的滤波值,使灰度相同但位置不同的像素点能计算出不同的滤波值。

参考文献

- [1] Overton K J, Weymouth T E. A noise reducing preprocessing algorithm[A]// Proceedings of IEEE Computer Science Conference on Pattern Recognition and Image Processing[C]. Chica-

go, Illinois, USA, 1979:498-507

[2] Pham T Q, van Vliet L J. Separable bilateral filtering for fast video preprocessing[C]//Proceedings of the IEEE International Conference on Multimedia and Expo, 2005

[3] Weiss B. Fast median and bilateral filtering[J]. ACM Transactions on Graphics, 2006, 25(3):519-526

[4] Durand F, Dorsey J. Fast bilateral filtering for the display of high-dynamic-range images[J]. ACM Transactions on Graphics, 2002, 21(3):257-266

[5] Pairs S, Durand F. A fast approximation of the bilateral filter using a signal processing approach[J]. International Journal of Computer Vision, 2009, 81(1):24-52

[6] Fattal R, Agrawala M, Rusinkiewicz S. Multiscale shape and detail enhancement from multi-light image collections[J]. ACM Transactions on Graphics, 2007, 26(3):51

[7] 李凡, 刘上乾, 秦翰林. 自适应双边滤波红外弱小目标检测方法[J]. 光子学报, 2010, 39(6)

[8] Fleishman S, Drori I, Cohen-Or D. Bilateral mesh denoising[C]//

Proc. ACM SIGGRAPH. San Diego, CA, July 2003:950-953

[9] Jones T R, Durand F, Desbrun M. Non-iterative feature-preserving mesh smoothing[C]//Proc ACM SIGGRAPH. San Diego, CA, July 2003:943-949

[10] Xiao J, Cheng H, Sawhney H, et al. Bilateral filtering-based optical flow estimation with occlusion detection[C]//Proc. European Conference on Computer Vision. Vol. 1, Graz, Austria, May 2006:211-224

[11] Zhang M, Gunturk B K. Multiresolution bilateral filtering for image denoising[J]. IEEE Trans. Image Processing, 2008, 17(12):2324-2333

[12] Zhang M, Gunturk B K. Compression artifact reduction with adaptive bilateral filtering[C]//Proc. SPIE Electronic Imaging, vol. 7257, San Jose, CA, 72571A, 2009

[13] Yang Q X, Tan K H, Ahuja N. Real-time O(1) bilateral filtering[C]//Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Washington D C, IEEE Computer Society-Press, 2009:557-564

(上接第 228 页)

样布局物较多的空间利用率为 0.879681。

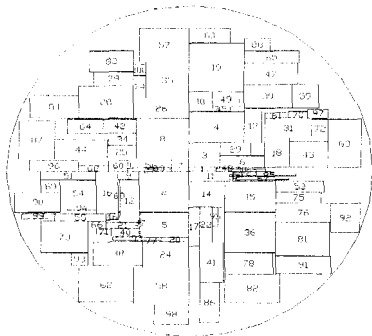


图 1 ACO算法在算例 3 中的最好布局

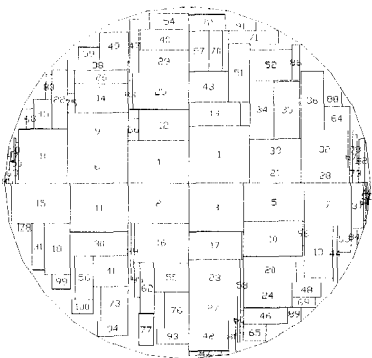


图 2 本文算法在算例 3 中的最好布局

结束语 本文针对复杂工程布局问题,提出了一种人机交互的混合算法,通过对算例进行数值计算和分析比较,验证了本文提出的算法是可行和有效的。本文提出的人机交互的混合算法充分发挥了人机各自的特长,相对现有的仅仅用元启发式算法求解此类工程布局的问题,从计算时间到解的质量都有明显的优势,而且较稳定。而且,本文为如何求解复杂工程布局问题和如何实现人机合作的问题都提供了一种很有效的方法或途径,为带平衡约束的不规则设备的布局求解提供了一种新的方法,该方法可以推广到任意不规则设备的布局问题中。针对任意结构的复杂工程布局问题,是我们下一步的工作。

参考文献

[1] 杨敏,王刚,刘璟,等.用双目标加权遗传算法解决网络磁盘阵列系统下校验散布布局优化问题的研究[J]. 计算机科学, 2005, 32(5):73-75

[2] 韩国栋,肖庆辉,张帆.可重构系统中硬件任务布局布线算法研究[J]. 计算机科学, 2011, 38(11):291-295

[3] Gehring H, Bortfeldt A. A parallel genetic algorithm for solving the container loading problem[J]. International Transactions in Operational Research, 2002, 9(4):497-511

[4] 张德富,彭煜,朱文兴,等.求解三维装箱问题的混合模拟退火算法[J]. 计算机学报, 2009, 32(11):352-357

[5] Teng H F, Sun S L, Liu D Q, et al. Layout optimization for the objects located within a rotating vessel—a three dimensional packing problem with behavioral constraints[J]. Computers & Operations Research, 2001, 28(6):521-535

[6] 钱志勤,滕弘飞,孙治国.人机交互的遗传算法及其在约束布局优化中的应用[J]. 计算机学报, 2001, 24(5):553-559

[7] 于洋,查建中,唐晓君.基于学习的遗传算法及其在布局中的应用[J]. 计算机学报, 2001, 24(12):1242-1249

[8] 李宁,刘飞,孙德宝.基于带变异算子粒子群优化算法的约束布局优化研究[J]. 计算机学报, 2004, 27(7):897-903

[9] 刘建,黄文奇.利用改进的微分进化算法求解带平衡约束的圆形 packing 问题[J]. 信息与控制, 2006, 35(1):103-107

[10] 寇艳红,张其善,李先亮.车载 GPS/DR 组合导航系统的数据融合算法[J]. 北京航空航天大学学报, 2003, 29(3):264-268

[11] Huang W Q, Chen M. Note on: An improved algorithm for the packing of unequal circles within a larger containing circle[J]. Computers & Industrial Engineering, 2006, 50(3):338-344

[12] 陈希,王宁生.基于遗传算法的车间设备虚拟布局优化技术研究[J]. 东南大学学报:自然科学版, 2004, 34(5):627-631

[13] 曾明华,冯恩民.基于改进遗传算法的布局优化子问题[J]. 运筹与管理, 2005, 14(1):13-18

[14] 涂鹏,王星华,蒋红斐.基于遗传算法的道路平面线形优化设计[J]. 华东公路, 2007(1):10-12

[15] 许项东,程琳.城市道路单行系统布局优化的双层规划模型和混合算法[J]. 系统工程理论与实践, 2009, 29(10):180-187

[16] 裴胜玉,周永权,罗淇方.求解约束优化问题的混合粒子群算法[J]. 微电子学与计算机, 2010, 27(4):5-8