Dijkstra 及基于 Dijkstra 的前 N 条最短路径算法 在智能交通系统中的应用*

王 峰¹,游志胜¹,曼丽春¹,高 燕²,汤丽萍¹ (1. 四川大学 计算机学院,四川 成都 610065; 2. 成都信息工程学院,四川 成都 610225)

摘 要:介绍了 Dijkstra 算法在智能交通系统的套牌车查询子系统中的应用 ;分析指出了文献 1]中算法存在的问题 通过对此算法的改进 ;完善了基于 Dijkstra 算法的前 N 条最短路径算法。通过实验及实际应用对改进后的算法进行了验证。

关键词: Dijkstra 算法;智能交通;前 N 条最短路径

中图法分类号: TP301.6 文献标识码: A 文章编号: 1001-3695(2006)09-0203-03

Application of Dijkstra and Dijkstra-based N-Shortest-Paths Algorithm to Intelligent Transportation Systems

WANG Feng¹, YOU Zhi-sheng¹, MAN Li-chun¹, GAO Yan², TANG Li-ping¹

(1. College of Computer Science , Sichuan University , Chengdu Sichuan 610065 , China ; 2. Chengdu University of Information Technology , Chengdu Sichuan 610225 , China)

Abstract: Application of Dijkstra algorithm to intelligent transportation systems is presented. Problems of reference analyzed. Through improvement on the algorithm the Dijkstra-based *N*-Shortest-Paths algorithm is consummated. The improved algorithm is proved to be correct and efficient by experiments and practical application.

Key words: Dijkstra Algorithm ; Intelligent Transportation Systems ; N-Shortest-Paths

在智能交通系统的实践过程中,发现套牌车(一般指没有办理入户手续。盗用别的汽车车牌行驶的汽车)现象越来越严重,给相关部门的车辆管理工作带来了诸多不便。由于车牌图片数据量较大,如果完全由人工来识别套牌车,不仅工作量巨大,而且效率也较低。本文在 Dijkstra 算法^[2]的基础上,设计并实现了一个套牌车查询系统,提高了套牌车的识别效率,并进一步开发了车牌识别系统的应用。同时,在交通咨询和决策支持等应用中,往往不仅需要得到最优路径,还需要得到次优、再次优等路径,即所谓广义最短路径问题^[1]。本文基于 Dijkstra 算法,对文献 1]中提出的算法进行了改进,得到了一个新的前 N 条最短路径算法,满足了实际应用的需求。

1 套牌车查询系统的设计及实现

1.1 设计思想

Dijkstra 算法被公认为是图论^[3]中求解最短路径问题的优秀算法。利用已有的交通数据 ,根据 Dijkstra 算法可以计算出任意两个监控设备安装点之间的最短路径。根据当地的交通状况给定一个平均速度阈值 ,检测同一个车牌的出现地点和时间 ,根据两个出现地点间的最短路径及出现的时间差 ,即可计算出其平均速度。如果此平均速度大于指定的阈值 ,则为套牌车。下面举一个例子:

需要说明的是,考虑到车牌识别系统可能出现车牌识别错误的情况,在程序认定套牌车后,仍需进一步通过人工对识别的图片进行确认,以排除那些识别错误的车牌。一个典型的容易造成识别错误的例子是对车牌上的 E 或 F 进行涂抹导致车牌污损。

例 1 车牌号为京 BF6446 在 2005 年 5 月 30 日那天,于

17 时 23 分 33 秒出现在白广路 ,17 时 26 分 12 秒出现在北河

沿大街北口 其所用时间为 2 分 39 秒 而这两个地点间的最短

路径为8734m则其平均时速为197.75km/h。由北京的交通

状况可判定,在这两个地点出现的车牌信息不是同一辆车,很

1.2 程序实现

可能是套牌车。

套牌车查询子系统是一个相对独立的系统,程序采用 VC++6.0 编写,采用 Access 2000 数据库存放相关的交通信息数据。

首先将监控设备安装点的信息以及彼此间的路径信息导入 Access 数据库中 利用 Dijkstra 算法计算出任意两个监控设备安装点之间的最短路径信息 并存入 Access 数据库中。

进行套牌车图片对比时,程序首先从后台 Oracle 数据库保存的图片中取出相应时间段的图片,然后从这些图片中查找车牌号相同的图片,并根据抓拍时间和 Access 数据库中保存的最短路径信息计算出其平均时速。再与设定的平均速度阈值进行比较,即可判断其是否为套牌车。

收稿日期:2005-07-21;修返日期:2005-10-14

程序主界面如图1所示。



图 1 程序主界面

1.3 时间复杂度分析

Dijkstra 算法中计算一个点到其他点的最短路径共需两次循环 时间复杂度为 α n^2)。这里由于要计算每一对顶点间的最短路径 故需以每个顶点为源点 执行算法 n 次 总的执行时间复杂度为 α α)。

2 基于 Dijkstra 前 N 条最短路径算法的研究与实现

2.1 相关工作

文献 1]对广义最短路径问题进行了图论上的理论分析,并设计了一个算法。经过对此算法的仔细研究,发现文献 1]的算法存在一些问题。本文在文献 1]的基础上,进一步改进并完善了基于 Dijkstra 算法的前 N 条最短路径算法。

2.2 文献[1]中算法存在的问题

现举一个例子来说明文献[1]中算法存在的问题。为描述方便起见,以带权无向图(图2)为例,用Paths[i]表示第i条最短路径,用dist表示路径 Paths[i]的权值。现欲求从顶点 $1\sim4$ 的前七条最短路径,若按文献[1]中的算法,则前七条最短路径分别为

Paths[1]=1-2-4 dist=25 Paths[2]=1-5-4 dist=75;

Paths[3] = 1-5-4 ,dist = 75 ;Paths[4] = 1-2-3-5-4 ,dist = 75 ;

Paths[5] = 1-3-2-4 , dist = 90; Paths[6] = 1-3-2-4 , dist = 90;

Paths[7] = 1-3-5-4, dist = 90_{\circ}

显然 ,此结果集存在两个问题 ① Paths[3]与 Paths[2] ,以 及 Paths[6]与 Paths[5]均是重复路径 ;②由于存在重复路径 ,结果集并不是所要求的前 N 条最短路径。

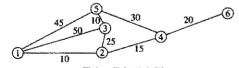


图 2 带权无向图

从上面的结果中可以看出 ,若要求得正确的前 N 条最短路径解集。在基本不改变原算法的基础上 ,可以通过以下两个步骤来实现:

- (1)适当增大 N 值,即为了求得前 N 条最短路径,实际运算时应该输入 N' 值($N' \ge N$)。如上例,为得到前六条最短路径,实验表明运算时需输入的 N' 值必须大于等于 11,即 $N' \ge 11$ 。实验结果如表 1 所示。
- (2)在获得计算结果后 將那些重复的路径删除掉。从表 1 中可以看出 结果集中有五条重复路径(斜体所示记录),将 之删除后得到最终结果集:

Paths[1] = 1-2-4 , dist = 25;

Paths[2] = 1-5-4 dist = 75;

Paths[3] = 1-2-3-5-4 dist = 75;

Paths[4] = 1-3-2-4 dist = 90;

Paths[5] = 1-3-5-4, dist = 90;

Paths [6] = 1-5-3-2-4 dist = 95.

表 1 为求图 2 的前六条最短路径而输入 N'=13 后产生的结果集

起点	终点	权值	最短路径序号	路径
1	4	25	1	1-2-4
1	4	75	2	1-5-4
1	4	75	3	1-5-4
1	4	75	4	1-2-3-5-4
1	4	90	5	1-3-2-4
1	4	90	6	1-3-2-4
1	4	90	7	1-3-5-4
1	4	90	8	1-3-5-4
1	4	90	9	1-3-5-4
1	4	90	10	1-3-5-4
1	4	95	11	1-5-3-2-4

但这种方法的一个问题是:无法明确地知道输入的 N'值 究竟应该是多大。若 N'值输入过小,则无法得到正确的最短路径解集;若输入 N'值过大,则会造成计算资源的浪费,使程序效率降低。同时,当图的顶点数较多,顶点的度比较大,所求的前 N 条最短路径的 N 值较大时,为得到正确的解集,会产生大量的重复信息。以文献[1]中的图 1 为例,经实验验证,为得到从顶点 $1\sim11$ 的前 12 条最短路径 输入的 N'值应该大于或等于 40。当输入 N'=40 时,产生的结果集中共有 28 条重复路径,冗余信息所占比例达到了 70%。为节省篇幅,实验结果不再列出。

2.3 改进后的前 *N* 条最短路径算法的设计与实现

首先,为讨论方便起见,作如下约定:

- (1)Q(VE)为带权图 其顶点数为 m 边数为 n 表示交通 网络。根据实际应用的需求,可以为有向图,也可为无向图。
- $(2)V = \{v_i \mid v_i$ 是图 G 的顶点 i = 1, 2, ..., m }是图 G 的顶点集合。
 - (3) $E = \{e_i | e_i$ 是图 G 的边 j = 1, 2, ..., n)是图 G 的边集合。
- (4)Paths[i] = $\{e_p \mid e_p \in E \mid p = 1 \mid 2 \mid \dots \mid k \mid k \leq n \}$ 是图 G 中从源点 souVex 到目标点 desVex 的第 i 条最短路径。
- (5)CutEdgeSet[i] = $\{e_d \mid e_d \in E, d = 1, 2, ..., r, r \leq n\}$ 是为得到图 G 从源点 souVex 到目标点 desVex 的第 i 条最短路径而删除的边集合。

通过上面的分析,可以发现问题产生的原因是由于在求第i条最短路径的过程中,当在图G中删除 Paths[i-1]中的不同边得到的最短路径P相同时,原算法将相同的最短路径P直接赋给了 Paths[i+1],从而造成了重复路径的产生。为解决此问题,引入候选删除边集合的概念。

定义 1 将计算图 G 从源点 souVex 到目标点 des Vex 的第i 条最短路径而删除的所有可能边集合所构成的集合定义为候选删除边集合 CandidateCutEdgeSel[i]。

初始化时,CutEdgeSet1[i] = \emptyset ,CandidateCutEdgeSet[i] = \emptyset 。 现求第i条最短路径。设 CutEdgeSet1[i - 1] \in Candidate-CutEdgeSet[i - 1],图 G'为图 G 删除 CutEdgeSet1[i - 1] 中各边后所得到的子图 显然 $G' \subseteq G$ 。设 $e_1 \in$ Paths[i - 1] P_1 为图 G' 删除 e_1 后得到的最短路径,则 CandidateCutEdgeSet[i] = CutEdgeSet1[i - 1] \cup { e_1 }。如果 $e_2 \in$ Paths[i - 1],并且 $e_1 \neq e_2$, P_2

为图 G'删除 e_2 后得到的最短路径 若 $P_2 = P_1$ (路径相同)则 e_2 为路径 Paths[i]的一条候选删除边,此时 CandidateCutEdgeSet[i] \cup (CutEdgeSet1[i-1] \cup { e_2 })。依此类推,即可得到候选删除边集合 CandidateCutEdgeSet[i]的值。

在算法实现时,采用链表数据结构来存放各条最短路径的 候选删除边集合。链表节点的数据结构如图 3 所示。

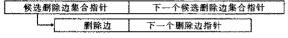


图 3 候选删除边集合链表节点数据结构

引入候选删除边集合后,既可以避免产生重复路径,又不会因为丢弃删除边信息而造成结果集的不完备。图 4 是求图 2 中从顶点 1~4 的前七条最短路径后所产生的数据结构描述示意图(各条路径的候选删除边集合不是唯一的,因算法的具体实现不同可能会不同)。

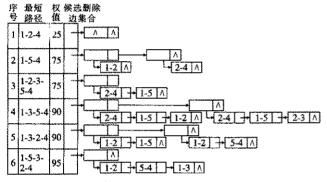


图 4 求图 2 从顶点 1~4 的前七条 最短路径的数据结构描述示意图

算法 1 是改进后的基于 Dijkstra 的前 N 条最短路径算法的完整描述。

算法 1. 改进后的基于 Dijkstra 的前 N 条最短路径算法。 输入 :G-交通网络图(带权无向图或有向图) ;souVex-源点 ;desVex-目标点 ;N-所要求的最短路径数目

输出 Paths[i] i=1 2 \dots N 即图 G 的前 N 条最短路径

```
Begin
    for( i = 0 ; < N ; ++ )//初始化
      CandidateCutEdgeSet[ i ] = \emptyset;
      Paths[i] = NULL;
    for (i = 0, i < N, i + +)
    {
      if(i==0)//求第一条最短路径
        {
          //求图 G 中从 souVex 到 desVex 的最短路径 结果保存在 P 中
          Dijkstra( G souVex ,desVex ,P);
          Paths[ i ] = P ;
      else
        //上一条最短路径不存在 /结束运行
        if ( Paths[i-1] = = NULL) break;
        for ( CutEdgeSet \in CandidateCutEdgeSet[i-1]) {
          //在 G 中将 CutEdgeSet 中各边删除 ,得到子图 G'
          CutEdgeNum = SizeOfSet( CutEdgeSet );
          for j = 0 j < CutEdgeNum <math>j + + j
          {
            edge = memberOfSet( CutEdgeSet ,j );
```

```
dist[ j ] = edge. dist;
            edge. dist = \infty;
        }
        / * 在 G'中依次删除 Paths i - 1 1中的各边 求 G'的最短路径
上一条最短路径的边数 */
    EdgeNum = LengOfPath(Paths[i-1]);
    for j = 0 j < EdgeNum <math>j + + 1
      {
      / * 新删除边集合与已有删除边集合是否有相同的标志 若相同
则为 false */
   flagContinue = true; edge = EdgeOfPath( Paths[i-1],j);
   / * 若 edge ∪ CutEdgeSet 与已有删除边集合相同 ,则没有必要继续
进行下面的运算,因为得到的最短路径是一样的 */
   for (ij = 1 \ ij < N \ ij + +)
    if( CutEdgeSetCompare( edge∪CutEdgeSet ,jj ) = = true )
      flagContinue = false;
      break;
   //删除边集合未出现过 求 G'的最短路径
   if (flagContinue = = true)
    {
      dist = edge. dist //暂存边权值
      edge. dist = ∞ //删除边
      Dijkstra( G souVex desVex P);
      edge. dist = dist ://恢复边
      dist = DistOfPath(P);
      for (k = i, k < N, k + +)
        //路径 P 的权值小于或等于 Paths k 1的权值
        if ( DistOfPath( Paths[ k ]) > = dist && dist ! = \infty )
      //路径 P 尚未出现过
      if ( pathCompareAll( P \Omega ,N - 1 ) = = -1 )
      //将路径 P 插入适当位置 使 Paths[i]中各条路径按升序存放
      for kk = N - 1 : kk > k : kk - - )
        { Paths[ kk ] = Paths[ kk - 1 ];
          CandidateCutEdgeSet[ kk ] = CandidateCutEdgeSet[ kk -1 ];
          Paths[k] = P;
        //得到 Paths[ k ]对应的候选删除边集合
        CandidateCutEdgeSet[ k ] = edge ∪ CutEdgeSet;
        }//end if ( pathCompareAll( P k ))
        //出现候选删除边,为相应的路径新增一个删除边集合
        if ( fExist = pathCompareAll( P k N-1 )> = k )
            CandidateCutEdgeSet[ fExist ] =
          CandidateCutEdgeSe[ fExist ]∪( edge∪CutEdgeSet );
          \/\ end if ( DistOfPath( Paths[ k ])> = dist && dist ! = \infty )
              break:
             }//end if ( flagContinue = = true )
        \frac{1}{2}/end for ( j = 0 j < EdgeNum <math>j + +  )
        for(j = 0 j < CutEdgeNum j + + )//恢复图
          edge = MemberOfSet( CutEdgeSet ,j );
          //原算法的 edge. dist = ∞ 是不正确的
          edge. dist = dist[ j ];
      } //end for ( CutEdgeSet ∈ CandidateCutEdgeSet[ i - 1 ])
      }//end else
    \frac{1}{2} end for i = 0 i < N i + + )
```

对于有向图的情况,只是在为图的边赋值时不同,其他过

(下转第208页)

程完全相同 在此不再赘述。

4 结论

本文中我们提出一种新的协同过滤算法来解决数据稀疏性的问题。同时提高预测的准确性。利用奇异值分解法的结果来进行邻居选择。然后采用 Pearson 方法来得到未打分项目的预测值。试验结果表明这种算法在数据稀疏时算法的准确性超过普通的 Pearson 算法和奇异值分解算法。

参考文献:

- [1] Berry M W et al. Using Linear Algebra for Intelligent Information Retrieva [J]. SIAM Review , 1995 37(4) 573-595.
- [2] Breese J S, Heckerman D, et al. Empirical Analysis of Predictive Algorithms for Collaborative Filtering C. Proc. of the 14th Conference on Uncertainty in Artificial Intelligence, 1998. 43-52.
- [3] Dan Kalman. A Singularly Valuable Decomposition: The SVD of a Matrix J]. The College Mathematics Journal, 1996, 27(1) 2-23.
- [4] Deerwester S, Dumais ST, Furnas GW, et al. Indexing by Latent Semantic Analysis J. Journal of the American Society for Information Science ,1990 41(6) 391-407.
- [5] Golub G H, Van Loan C F. Matrix Computations (3rd edition X M]. Johns Hopkins University Press, 1996.
- [6] Herlocker J L , Konstan J A , Borchers A , et al. An Algorithmic Framework for Performing Collaborative Filtering C]. Proceedings of

- ACM SIGIR '99, ACM press, 1999. 230-237.
- [7] John A Swets. Measuring the Accuracy of Diagnostic System [J]. Science, 1988 240(4857) 1285-1289.
- [8] McJones P. EachMovie Collaborative Filtering Data Set[DB/OL]. DEC Systems Research Center, 1997.
- [9] Michael H Pryor. The Effects of Singular Value Decomposition on Collaborative Filtering R]. Technical Report PCS-TR98-338, Dartmouth College, 1998.
- [10] Resnick P , Iacovou N , et al. GroupLens : An Open Architecture for Collaborative Filtering of Netnews[C]. Proceedings of CSCW '94 , Chapel Hill , NC , 1994. 175-186.
- [11] Sarwar B M , Karypis G , et al. Application of Dimensionality Reduction in Recommender Systems : A Case Study[C]. ACM WebKDD Web Mining for E-commerce Workshop , 2000. 114-121.
- [12] 邓爱林 朱扬勇 施伯乐. 基于项目评分预测的协同过滤推荐算法 [J]. 软件学报 2002, 13(4):1-8.
- [13] 翼俊忠 ,沙志强 ,刘椿年. 一种基于贝叶斯网客户购物模型的商品推荐方法 J]. 计算机应用研究 2005 22(4) 65-71.

作者简介:

孙小华(1973-) 男 江西婺源人 博士研究生 注要研究方向为机器学习、数据挖掘 陈洪(1973-) 男 浙江天台人 硕士 主要研究方向为数据挖掘、软件过程管理 :孔繁胜(1946-) 男 ,上海人 ,教授 ,博导 ,主要研究方向为数据挖掘、地理信息系统。

(上接第205页)

2.4 算法复杂度分析

本文在文献 1]的基础上,通过引入候选删除边集合,解决了原算法存在的问题,其时间复杂度和空间复杂度均与原算法相同。即时间复杂度为 $O(n^3)$;若以链表结构来存储图,则空间复杂度为 O(n)。其中 n 为图 G 的顶点数。其具体分析过程参见文献 1 1。

3 实验及应用

程序用VC ++6.0 实现,实验平台为 $2.0 \, \mathrm{GHz}$ CPU $256 \, \mathrm{MB}$ 内存,Windows 2000 操作系统。应用改进后的算法求解文献 [1]中图 1 从顶点 $1 \sim 11$ 的前八条最短路径,程序运行耗时 $0.035 \, \mathrm{s}$ 运算结果如表 $2 \, \mathrm{fm}$ 示。

表 2 文献 1]图 1从顶点 1~11的前八条最短路径

起点	终 点	权 值	最短路径序号	路 径
1	11	42	1	1-3-7-10-11
1	11	45	2	1-4-7-10-11
1	11	46	3	1-3-5-8-11
1	11	47	4	1-2-5-8-11
1	11	50	5	1-2-3-7-10-11
1	11	53	6	1-3-7-9-11
1	11	53	7	1-4-3-7-10-11
1	11	53	8	1-3-5-9-11

在一个有64个主要路口的"三横两纵一环"的交通网络

中 随机取 40 对路口 计算任意两个路口的前七条最短路径所用平均时间为 2.23s。

4 结论

Dijkstra 算法是图论中求解最短路径问题的一种优秀算法 本文介绍了其在智能交通系统的套牌车查询子系统中的应用。分析了文献 1]中算法存在的问题 提出了一个改进后的基于 Dijkstra 算法的前 N 条最短路径算法 解决了文献 1]算法存在的问题 满足了实际应用的需求。通过实验及实际应用的检验 ,可看出本算法是正确并且有效的。

参考文献:

- [1] 柴登峰 涨登荣. 前 N 条最短路径问题的算法及应用[J]. 浙江 大学学报(工学版) 2002 36(5) 531-534.
- [2] 严蔚敏 吴伟明. 数据结构[M]. 北京 清华大学出版社 1999.
- [3] [美]Bondy JA, Murty USR. 图论及其应用[M]. 吴望名,等. 北京 科学出版社, 1984.

作者简介:

王峰(1975-),男、四川人,博士,主要研究方向为计算机网络、数字图像处理等游志胜(1945-),男,四川人,所长,中国图像图形学会副理事长,教授,博导,主要研究方向为数字图像处理、智能交通、模式识别等;漫丽春(1978-),女,河南人,硕士,主要研究方向为计算机网络,汤丽燕(1977-),女,四川人,助教,硕士,主要研究方向为计算机网络,汤丽萍(1980-),女,浙江人,硕士,主要研究方向为计算机网络。

作者: 王峰, 游志胜, 曼丽春, 高燕, 汤丽萍, WANG Feng, YOU Zhi-sheng, MAN Li-chun

, GAO Yan, TANG Li-ping

作者单位: 王峰, 游志胜, 曼丽春, 汤丽萍, WANG Feng, YOU Zhi-sheng, MAN Li-chun, TANG Li-ping(四川大

学 计算机学院,四川 成都 610065), 高燕, GAO Yan(成都信息工程学院,四川 成都 610225)

刊名: 计算机应用研究 ISTIC PKU

英文刊名: APPLICATION RESEARCH OF COMPUTERS

年,卷(期): 2006,23(9) 被引用次数: 63次

参考文献(3条)

1. 柴登峰, 张登荣 前N条最短路径问题的算法及应用[期刊论文]-浙江大学学报(工学版) 2002(5)

- 2. 严蔚敏; 吴伟明 数据结构 1999
- 3. J A 邦迪; U S R 默蒂; 吴望名; 李念祖 图论及其应用 1984

本文读者也读过(5条)

- 1. 袁红涛. 朱美正 K优路径的一种求解算法与实现[期刊论文]-计算机工程与应用2004, 40(6)
- 2. 王元彪. WANG Yuanbiao 智能交通系统中Dijkstra算法的高效实现[期刊论文]-计算机工程2007, 33(6)
- 3. <u>高松. 陆锋. 段滢滢. GAO Song. LU Feng. DUAN Yingying</u> <u>一种基于双向搜索的K则最优路径算法</u>[期刊论文]-武汉大学学报(信息科学版)2008, 33(4)
- 4. 戴树贵. 陈文兰. Dai Shugui. Chen Wenlan 一个求解k短路径实用算法[期刊论文]-计算机工程与应用2005, 41(36)
- 5. 王明中. 谢剑英. 陈应麟 一种新的Kth最短路径搜索算法[期刊论文]-计算机工程与应用2004, 40(30)

引证文献(53条)

- 1. 王文斌 时钟同步积累误差补偿的传感通信优化仿真[期刊论文]-计算机仿真 2014(10)
- 2. 孙奥, 朱桂斌, 江铁 基于预测信息的时间依赖网络路径规划算法[期刊论文]-计算机工程 2012(22)
- 3. 毛红梅, 甘晟科 求有向图中源点到各结点所有路径的一种实用算法[期刊论文]-微电子学与计算机 2009(03)
- 4. 邹亮, 徐建闽, 朱玲湘 A*算法改进及其在动态最短路径问题中的应用[期刊论文]-深圳大学学报(理工版) 2007(01)
- 5. <u>吴刚, 刘海砚, 王俊超, 张彬彬</u> 基于熵的模糊决策在最优路径选取中的应用[期刊论文]-辽宁工程技术大学学报(自然 科学版) 2012 (04)
- 6. 余宜诚 一种新颖的最短路径发现算法[期刊论文] 吉林工程技术师范学院学报 2011(09)
- 7. 胡慧, 邓安远 基于数据库存储的最优路径查询算法改进[期刊论文]-制造业自动化 2010(06)
- 8. 陶波, 朱玉琴 改进的动态规划法在车辆最短路径问题中的应用[期刊论文]-重庆工学院学报(自然科学版)

2009 (01)

- 9. 胡明, 赵婉婷 基于移动定位信息的最短路径搜索算法[期刊论文]-电脑知识与技术 2009(32)
- 10. 阳西述, 刘怀玉, 胡亚辉 两区域交叉网络图的Dijkstra改进算法[期刊论文] 计算机科学 2014(z1)
- 11. 刘永朋, 孙田明 基于分层移动通信优化算法的应用与仿真[期刊论文]-计算机仿真 2014(05)
- 12. 陈卫东, 赵庆展 P2P环境下不信任节点分层免疫模型仿真分析[期刊论文]-计算机仿真 2014(03)
- 13. 王卫华 相对位置多次变化的自组织网路由模型仿真[期刊论文] 科技通报 2014(03)
- 14. 张勤 Di jkstra最短路径算法的C语言实现[期刊论文]-福州大学学报(哲学社会科学版) 2011(04)
- 15. 原慧琳, 汪定伟 最短路径的可达矩阵算法[期刊论文]-信息与控制 2011(02)
- 16. 李星毅, 翟晓峰, 施化吉 最小时间路径算法的改进及在路径优化中的应用[期刊论文]-计算机应用研究 2008(06)
- 17. 吴春颖, 王士同 基于二元语法的N-最大概率中文粗分模型[期刊论文]-计算机应用 2007(12)

- 18. 刘胜来, 李瑞敏 基于网络最短路径的铁路购票智能推荐算法研究[期刊论文] 铁路计算机应用 2014(03)
- 19. 田涛 P2P网络数据通信优化算法研究与仿真[期刊论文]-计算机仿真 2012(10)
- 20. 黄婷 一种信息低错校验的传感网络优化通信算法[期刊论文]-科技通报 2012(12)
- 21. 陈贤敏 P2P网络快速通信算法研究[期刊论文]-计算机仿真 2011(12)
- 22. 高永强 移动自组通信网络的去噪通信优化过程分析[期刊论文]-计算机仿真 2015(4)
- 23. 王英, 刘天时 基于Kruskal 算法的最短路径算法研究[期刊论文]-重庆文理学院学报(自然科学版) 2009(06)
- 24. 黄镇建, 蔡群英 一种基于稀疏矩阵的求路径集的新算法[期刊论文] 计算机与数字工程 2009(11)

2007 (01)

- 26. 周伟, 余粟 无线网络阴影区域优化通信方法研究与仿真[期刊论文]-计算机仿真 2014(6)
- 27. 高松, 陆锋, 段滢滢 一种基于双向搜索的K则最优路径算法[期刊论文]-武汉大学学报(信息科学版) 2008(04)
- 28. 孙茜茜, 陆南 基于Android与Mobile GIS的新生报到服务系统[期刊论文]-现代电子技术 2013(20)
- 29. 魏二虎, 贾满, 李林燕 最短路径算法的改进方法研究[期刊论文] 测绘信息与工程 2007(04)
- 30. 周斌 基于电路分析法的道路交通分配及最优路径模拟[学位论文]硕士 2009
- 31. 高松, 陆锋 K则最短路径算法效率与精度评估[期刊论文]-中国图象图形学报A 2009(08)
- 32. 黄书力, 胡大裟, 蒋玉明 经过指定的中间节点集的最短路径算法[期刊论文]-计算机工程与应用 2015(11)
- 33. 赵作鹏, 宋国娟, 宗元元, 李晓波, 康清华, 张雪涛, 王艳辉, 许新征 基于D-K算法的煤矿水灾多最优路径研究[期刊论文]-煤炭学报 2015(02)
- 34. 兰岚 城市智能交通管理系统方案研究与设计[学位论文]硕士 2010
- 35. 史占成 CA信任模型的研究与构建[学位论文]硕士 2007
- 36. 任建文, 李刚, 王增平, 甄旭锋 基于背离路径的输电断面搜索新算法[期刊论文]-电网技术 2012(04)
- 37. 李雪华 虚空间理论及其在城市空间聚散分析中的应用[学位论文]硕士 2007
- 38. 孙霞 基于改进遗传算法的城市交通路径诱导系统的研究[学位论文]硕士 2007
- 39. 焦光庭 交通运输科技执法系统关键技术研究及实现[学位论文]硕士 2009
- 40. 胡志苹 基于蚁群算法的海上消防系统航线优选研究[学位论文]硕士 2007
- 41. 孙美娜 天津站地区路径诱导系统关键技术研究[学位论文]硕士 2013
- 42. 熊聪峰 范例推理在高速公路突发事件中的应用研究[学位论文]硕士 2008
- 43. 翟晓峰 车辆诱导系统及关键技术研究[学位论文]硕士 2007
- 44. 彭益兵 轨道交通路网末班车合理售票算法的研究与应用[学位论文]硕士 2010
- 45. 高俊红 110出警线路优化系统的设计与实现[学位论文]硕士 2010
- 46. 王冰 基于随机时间依赖的k期望最短路径研究[学位论文]硕士 2007
- 47. 邓敏皓 SOA-BPM物流信息平台下的车辆路径选择研究[学位论文]硕士 2011
- 48. 苏日娜 基于转移模型的OD算法研究[学位论文]硕士 2008
- 49. 史占成 CA信任模型的研究与构建[学位论文]硕士 2007
- 50. 王志强 城市轨道交通应急决策辅助技术研究[学位论文]博士 2008
- 51. 邵明洋 震后交通系统通过能力研究[学位论文]硕士 2009
- 52. 胡红 大城市应急交通疏散规划及关键技术研究[学位论文]博士 2008
- 53. 郭建宏 林副产品配送优化辅助决策模型及GIS集成研究[学位论文]博士 2008

引用本文格式: <u>王峰. 游志胜. 曼丽春. 高燕. 汤丽萍. WANG Feng. YOU Zhi-sheng. MAN Li-chun. GAO Yan. TANG Li-ping</u> Dijkstra及基于Dijkstra的前N条最短路径算法在智能交通系统中的应用[期刊论文]-计算机应用研究 2006(9)