

文章编号:1671-8879(2005)06-0062-04

城市道路最短路径的 Dijkstra 算法优化

张渭军¹, 王 华²

(1. 长安大学 地球科学与国土资源学院, 陕西 西安 710054;

2. 陕西交通职业技术学院 经济管理系, 陕西 西安, 710021)

摘 要:在研究城市道路网络特征基础上, 建立城市道路网络模型及其数据库, 应用一种改进的 Dijkstra 算法对城市道路进行最短路径查询, 该算法是从起点和终点分别用二叉树按起点到终点和终点到起点的方向进行搜索。在计算某一段最短路径时, 用 Dijkstra 算法时间为 0.23 s, 改进算法时间为 0.20 s。仿真结果表明, 该算法不仅在时间上有所改进, 其时间复杂度由传统 Dijkstra 算法的 $O(n^2)$ 减小为 $O(n)$, 而且其所选的最优路径更符合实际, 是一种寻求最优路径的有效算法。

关键词:交通工程; 道路网络; 数据库; Dijkstra 算法; 最短路径; 二叉树

中图分类号:U491.1

文献标识码:A

Optimization Dijkstra arithmetic for shortest path of urban traffic net

ZHANG Wei-jun¹, WANG Hua²

(1. School of Earth Sciences and Resources Management, Chang'an University, Xi'an 710054, China;

2. Department of Economy and Management, Shaanxi College of Communications Technology, Xi'an 710021, China)

Abstract: This paper established the road net model and database of urban traffic based on the study of the urban traffic road character. The shortest path was queried by the improving Dijkstra arithmetic, the arithmetic used B-tree in the start and end point separately in the direction of start-end and end-start. The times are 0.20 s and 0.23 s when using Optimization Dijkstra arithmetic and the Dijkstra arithmetic. The simulation results indicate that the complex grade in time is minished to $O(n)$ from the Dijkstra's $O(n^2)$, the shortest path is agreed with the practice, the arithmetic is an effective method in selecting the shortest path. 2 tabs, 1 fig, 7 refs.

Key words: traffic engineering; road net; datebase; Dijkstra arithmetic; shortes' path; B-tree

0 引 言

交通网络分析是指通过一定方法, 评价网络特征的区位特征及特征之间的相互关系, 分析个体在交通网络中一定时间内可能的活动范围。交通网络分析的重点是路径分析, 而路径分析的核心为最优路径^[1], 以最短路径为主的最优路径是交通工程学、

地理信息科学研究的一个热点, 它往往是建立在抽象的数学模型即网络模型的基础上, 该模型中最短路径分析就是在指定网络两节点间找一个阻碍强度最小的路径, 它不仅包括距离最短还可以引伸到其他变量如时间、费用、线路容量等。本文在对城市交通网络进行一定数学抽象的基础上, 建立了一个基于地理相对关系的数学模型, 提出求城市道路网络

收稿日期: 2004-11-09

基金项目: 国家自然科学基金项目(600 720 44)

作者简介: 张渭军(1975-), 男, 陕西渭南人, 长安大学讲师, 博士研究生。

最短路径的 Dijkstra 改进算法,该算法的时间复杂度为 $O(n)$,通过实际检验,其结果更符合实际。

1 建立模型及属性数据库

城市交通主要由众多的街道相交、相连而构成,并组成纵横交错、错综复杂的城市交通网络图^[2]。本文以交叉路口点把道路进行分割,这样整个交通网将由交叉路口点和路段组成,并定义交叉路口点为网络的节点,路段为网络的边。建立模型过程就是把现实中的城市道路网络实体抽象为网络图论理论中的网络图,这就要理解它们二者的不同之处,然后通过一定的综合变换把它们有机地对应起来,城市网络图最大特点有以下几个方面:

(1)城市交通特征的空间分布存在显著的方向性差异,同一道路路段的不同方向有不同的通行能力、交通流量和阻抗^[3]。因此,由城市道路网系统抽象出来的网络应该是有向带权图,道路网络系统中的交叉口被抽象为网络图中的节点,而交叉口之间的路段被抽象为网络图中带权弧,它具有不同的通行能力、交通流量和阻抗。

(2)在公路路网分析规划或交通规划研究中,交叉口仅作为路的交叉点或控制段的结点来考虑,其长度等几何要素也要记入里程,相对而言,交叉口的宽度属于比较次要的位置。但对城市公交来说,交叉口宽度占有所有道路的 5% 以上,这样交叉口宽度就不能忽略。交通流在交叉口处有不同程度的延误,这对最短路径查询有影响,同时由于城市的历史变迁并不规整,许多交叉口是多路或畸形交叉,所以在 GIS(地理信息系统)中对城市道路基础图的修正与编辑关键为交叉口定义图形编辑原则。在此主要考虑 3 种情况:①2 条路的十字交叉,这样在交叉口的中心断开;②2 条路的 T 型交叉,同样也是在交叉口断开;③多路或畸形交叉,这要根据实际情况进行处理,必需反映到模型中去。

(3)拓扑关系的表达,在传统平面模型中,道路之间的拓扑关系通过弧段-节点属性表的方式隐式表达,交叉的道路在交叉口分享节点,这是一种典型几何拓扑表达^[4]。网络拓扑与几何拓扑并不是总是一致的,对于具有社会属性的交通网络而言,部分拓扑是人为设定的,如转向限制、转向阻抗等,网络拓扑中的转向限制主要通过平面拓扑转向表的方式来表达,实际建模过程,转向没有阻抗的可以删去,以节省存储空间。

针对以上城市交通网络特点,把城市道路网络

实体抽象为网络图论理论中的网络图,然后在此模型基础上建立相应的属性数据库,本文数据在 Map-Info 环境中进行矢量化,共分 2 个图层:节点图层及道路图层,每个图层有其对应的属性表结构,属性表结构文件定义了空间对象的属性数据的表结构。因此,在节点图层和道路图层的属性表结构文件中定义的节点和道路信息见表 1、表 2。

表 1 路段结点属性表

节点代码	节点名称	节点行车方向	换车时间
NodeID	NodeName	NodeDirection	ChangeTime

表 2 路段信息属性表

路段代码	路段名称	行车方向	路段等级
RoadID	RoadNam	RoadDir	RoadGra
路段长度	拥挤程序	路段起点	路段终点
RoadLeg	RoadCw	RoadF	RoadE

在 *.MIF 文件中主要包含了空间数据^[5],指明了地图的坐标系、属性表结构、地图对象的类型和地理坐标信息等,对图层中每一路段,均记录了该路段的起点和终点经纬度坐标。因此,可以直接对 *.MIF 文件进行操作,从中提取各节点的坐标信息,表 2 中有路段起点、终点,这样表 1、表 2 的属性表就建立了联系。

2 最短路径计算

城市道路网模型为 $G = (V, E)$ 。其中, V 为站点综合后产生的节点集合; E 为相邻两节点的距离即弧段距离,它是 1 个非负实数。

$$E = \{d_{ab} \mid d_{ab} = F(X1, X2, X3 \cdots)\} \quad (1)$$

式中: $X1$ 为简单距离因子,表示道路几何距离; $X2$ 为路面等级因子; $X3$ 为拥挤程度因子。

$X2$ 、 $X3$ 表示路面等级和人、车流量等因素对通行能力的影响,可以使用阻抗值来表示这种影响程度,而阻抗值的大小就需对整个交通区内路段等级和拥挤程度进行分类,然后依次对每一类赋予阻抗值。这样,阻抗值的大小表示路段等级和拥挤程度等对通行能力的影响,虽然这种影响只是定性的,并不能如实表示影响的大小,但却具有横向的比较性。

2.1 Dijkstra 最短路径算法

Dijkstra 算法的基本思路是:假如每点都有一对标号 (d_j, p_j) ,其中 d_j 是从起源点 s 到点 j 的最短路径的长度(从 1 个点点到其本身的长度为 0,2 个互不相通的点的距离为无穷大), p_j 则是从 s 到 j 的最

短路径中 j 点的前一点,求解从起源点 s 到 j 的最短路径算法基本过程为:

(1)初始化,起始点设置为:① $d_s=0$, p_s 为空;
②所有其他点: $d_i=\infty$,对不同的点寻找最短路径时, p_i 不同; ③标记起源点 s ,记 $k=s$,其他所有点设置为未标记的。

(2)检查从所有已标记的点 k 到其直接连接的未标记的点 j 的距离,并设置

$$d_j = \min\{d_j, d_k + l_{kj}\} \quad (2)$$

式中: l_{kj} 为从 k 到 j 的直接连接距离。

(3)选取下一个点,从所有未标记的结点中,选取 d_i 中最小的一个 i 值

$$d_i = \min\{d_i, \text{所有未标记的点 } j\}$$

对应点 i 最短路径中的 1 点,并设为已标定的。

(4)找到点 i 的前一点,从已标记的点中找到直接连接点 i 的点 u 作为前一点,设置

$$i = u$$

(5)标记点 i ,如果所有点已标记,则算法退出,否则记 $k=i$,转到式(2)再继续。

由上可知,顶点越多,循环次数越多,计算时间将成倍增长。Dijkstra 算法所采用的图遍历是以顶点数为基数作二次循环^[6],每重循环的次数均为节点数 n ,它的执行过程产生了以 s 为节点的一棵树,随着算法的进一步执行该树向四面八方延伸,直至到达终点为止,其算法执行的时间复杂度为 $O(n^2)$,它可以寻找出从源点 s 到所有节点的最短路径。显然,从最终结果来看,有些分支对最短路径的求得无益,能否删除这些分支,从而减少计算最短路径的时间,这正是本文提出改进 Dijkstra 算法的原因。

2.2 Dijkstra 算法改进

同上,设 s 和 j 是网络中任意给定的 2 个节点,要寻找 s 到 j 的最优路径,该路径是由边和节点组成的,一般交通网络中,组成 s, j 的各段道路都和 s, j 在同一条方向上是最简单的一种情况,而多数情况下组成 s, j 最优路径的节点和 s, j 边不在 1 条直线上,但其大致方向应是从 s 指向 j 。在此把最短路径分解为多个子问题进行求解,即由 s 到 j 和由 j 到 s 2 个方向的最短路径。算法步骤为:

(1)计算 s, j 的直线距离 $k = \overrightarrow{s, j}$,若 s, j 之间存在 1 条边,则该边为最短路径,若 s, j 上存在 1 条边链即 s, j 由在一个方向的几条路段组成,则该边链和为最短路径,并记录该链所经过的站点及总距离,否则,执行步骤(2)。

(2)分别从 s, j 出发,寻找与 $\overrightarrow{s, j}$ 、 $\overrightarrow{j, s}$ 夹角最小的

边,设该边的另一端点分别为 A_1 和 B_1 即

$$|f(\overrightarrow{s, j}, \overrightarrow{s, A_1})| = \min_{i \in D_1} (|f(\overrightarrow{s, j}, \overrightarrow{s, A_i})|)$$

其中, D_1 为所有与 s 相连的节点数;

$$|f(\overrightarrow{j, s}, \overrightarrow{j, B_1})| = \min_{i \in D_2} (|f(\overrightarrow{j, s}, \overrightarrow{j, B_i})|)$$

其中, D_2 为所有与 t 相连的节点数。

如果 A_1 是从 s 出发所选的点,则把 A_1 当作当前点,用该步骤寻找以 A_1 为起点且与 $\overrightarrow{A_1, j}$ 直线夹角的最小的边,如此反复直至当前所选的点与 j 点重合,这样得到的一个链为最短路径的一个解。当所选点为终止点并该点与 j 不重合时,则这一条链到此结束,然后把所选点的前一个点当作当前点继续执行该步骤,如此循环。同样,对于从 j 出发的另一条链来说和从 s 出发的算法一样。这样 s, j 之间的最短路径的最终结果可能有 2 条、1 条,并记录该链所经过的站点以及阻抗值。该步骤所得的最短路径具有 s, j 两点的方向性趋势,但其条件简单因此需进一步检验。

(3)同样以 s, j 分别为根结点,然后分别用二叉树从 s, j 出发,重复将 s, j 直线段两侧夹角最小的边加入二叉树,对该二叉树的活动节点作 2 个标记,直至与 j 或 s 点重合。这样 s, j 之间的最短路径最多有 4 条结果。为了减少花费时间并使构造的二叉树最小,应采用两种方法:首先对于二叉树的遍历采用广度优先的方法,尽可能减少二叉树的节点遍历数;其次对于子结点在加入路径队列前,对其进行判断,如果为终点 $j(s)$,则不加入路径队列,但要对其影响的节点和子节点累加长度标志进行改变。

通过上述算法之后,最多可能得到 4 条最短路径,并把各路径的阻抗值显示出来,然后通过人工对比,最终选 1 条最优路径。

3 算法效率比较

基于邻接矩阵的传统 Dijkstra 算法,时间复杂度为 $O(n^2)$,即使改用邻接表存储网络数据结构,虽然大大改进算法的空间复杂度,但对事件时间复杂度丝毫没有改善^[7]。当节点数目较大时,算法的时间花费极为庞大。本文的算法复杂度为 $O(n)$,它与节点数呈线性关系,当节点数大时,其优势就更加明显。为了验证本文算法的有效性,以西安城区为例(图 1),在 MapInfo 环境下对地图进行矢量化,然后再在 VB 环境下通过对 MapX 二次开发,实现了最短路径的快速搜索。图中共有交叉路口 538 个,计算机配置为 Pentium III 550,内存为 128 M,硬盘

为 10 G,用 Dijkstra 算法,时间为 0.23 s,用改进算法时间为 0.20 s,并且它选了 4 条最短路径及各条路径的阻抗,由用户比较这几条路径长度及其阻抗,从中选出 1 条最优路径,这比单纯的距离最短更符合实际。图中黑粗线即为起始站点到终点的最优路径,列表框内显示的是该线路所经过的站点,当交叉路口多时该算法时间效果更为明显。

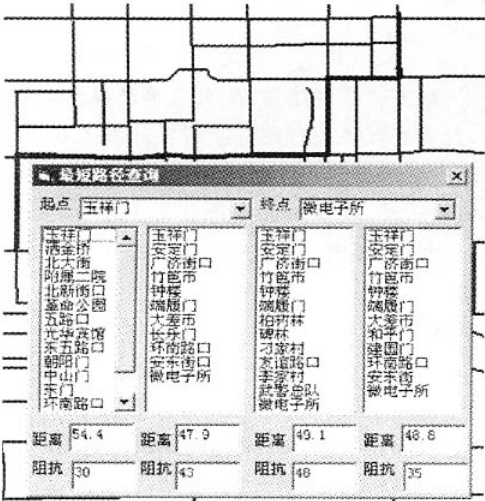


图 1 本文算法查询最短路径结果

该算法查找两点之间最短路径出现 4 条,最终中选距离是 48.8 km,阻抗为 35 的 1 条路为最优路径。当不计阻抗只按距离来讲,当然选距离为 47.9 km 的 1 条路为最短路径,很显然前一种更符合人们的思维习惯。

4 结 语

(1)最短路径是 GIS 网络分析在交通上最重要、最基本的应用,以往最短路径搜寻都是基于几何距离的最短路径,它对于公路交通而言更具意义,但对城市道路而言,距离最短不一定最优,因为还有路面等级和拥挤程度等因素的影响。

(2)本文最短路径搜索是从起点和终点分别用二叉树按其方向性进行搜索,用 Dijkstra 算法,时间为 0.23 s,用改进算法时间为 0.20 s,改进 Dijkstra 算法的效率明显比传统的 Dijkstra 算法效率高,并且搜索结果更符合实际。

(3)须改进的地方是在计算道路阻抗时要考虑更多的影响因子,这有待进一步研究。

参考文献:

References:

[1] 李英姿,张飞舟,林耀海. 智能交通系统中地理信息系统的研究[J]. 中国公路学报,2000,13(3):97-100.
LI Ying-zi, ZHANG Fei-zhou, LIN Yao-hai. Research on geographic information system in intelligent transportation systems[J]. China Journal of Highway and Transport,2000, 13(3):97-100.

[2] 桂 岚,李跃军. 组件式交通地理信息系统设计与开发[J]. 长安大学学报(自然科学版),2002,22(2):24-27.
GUI Lan, LI Yue-jun. Design and development of a components GIS for transportation[J]. Journal of Chang'an University (Natural Science Edition), 2002, 22(2):24-27.

[3] 张国强,晏克非. 城市道路网络交通特性仿真模型及最短路径算法[J]. 交通运输工程学报,2002,2(3):60-62.
ZHANG Guo-qiang, YAN Ke-fei. Simulation model based upon characteristics of urban road network and its shortest path algorithm[J]. Journal of Traffic and Transportation Engineering,2002,2(3):60-62.

[4] 严寒冰,刘迎春. 基于 GIS 的城市道路网最短路径算法探讨[J]. 计算机学报,2000,23(2):210-215.
YAN Han-bing, LIU Ying-chun. A new algorithm for finding shortcut in a city's road net based on GIS technology[J]. Chinese J Computers, 2000,23(2):210-215.

[5] 吴必军,李利新,雷小平. 基于城市道路数据库的最短路径搜索[J]. 西南交通大学学报,2003,38(1):80-83.
WU Bi-jun, LI Li-xin, LEI Xiao-ping. Shortest path searching based on city road database[J]. Journal of Southwest Jiaotong University, 2003,38(1):80-83.

[6] 周培德. 交通道路网中任意两点之间最短路径的快速算法[J]. 计算机工程与科学,2002,24(2):35-37.
ZHOU Pei-de. A fast algorithm for finding the shortest path between arbitrary two points in a traffic road net[J]. Computer Engineering and Science, 2002,24(2):35-37.

[7] 刘灿齐,杨佩昆. 基于最短路径的城市主干道网规划的算法研究[J]. 中国公路学报,2000,13(2):105-107.
LIU Can-qi, YANG Pei-kun. Urban main road network planning algorithm on shortest-path[J]. China Journal of Highway and Transport,2000,13(2):105-107.