

利用空间特性获取最短路径的研究及实现^①

Study and Implementation of the Shortest Path by using Spatial Characteristic

刘迎春 张莉 (浙江工业大学 教育科学与技术学院 浙江杭州 310032)

摘要: 从分析城市道路网地理相关性特征入手,研究利用道路网的空间特性信息来解决道路网中两点间的最短路径问题。通过建立体现道路网空间特性的数据模型,根据两点间直线距离最短的原理,提出一种道路网两点间最短路径的算法,利用 VC++ 进行了算法实现和最短路径的可视化显示。实验结果证明:利用空间特性信息可以有效地减少最短路径的搜索花费,同时算法的实现和最短路径的可视化不须依赖地理信息系统平台,具有较好的可移植性和实用性。

关键词: 空间特性 道路网 最短路径 算法

1 引言

最短路径问题是基于数据结构中图的理论而产生,在带权图中求最短路径问题是计算机科学中一个十分重要、且被广泛研究的问题。近年来,最短路径问题的研究不只基于抽象的网络模型,更偏重于在具体的应用领域里的算法改进和应用研究,最短路径常用来指地理意义上的距离最短,甚至时间、费用、流量的度量。最短路径问题作为很多领域中选择最优问题的基础,已被广泛应用于交通网络分析、路径导航、急救等各个领域,特别是和地理信息系统结合,使获得的最优路径信息以地图的形式显示在用户面前,提高了最短路径的可视化程度。

2 最短路径算法

最短路径算法一直是最短路径问题的焦点,传统的最短路径算法——Dijkstra 算法是目前解决最短路径问题的理论基础。Dijkstra 算法的基本思路是:设 $G = (V, A)$ 是一个赋权有向图,图中每个顶点都有一对标号 (d_i, p_i) ,其中 d_i 是从起点 s 到点 i 的最短路径的长度; p_i 是从 s 到 i 的最短路径中 i 点的前一点。求解从起点 s 到点 j 的最短路径算法的基本过程如下:

(1) 初始化。起点设置为: $d_s = 0$, p_s 为空;有其他点: $d_i = \infty$; 标记起点 s , 记 $k = s$, 其他所有点设为未标记的。

(2) 检验从所有已标记的点 k 到其直接连接的未标记的点 l 的距离,并设置:

$$d_l = \min[d_l, d_k + l_{kl}]$$

式中, l_{kl} 是从点 k 到 l 的直接连接距离。

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 :

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

点 i 就被选为最短路径中的一点,并设为已标记的。

(3) 找到点 i 的前一点。从已标记的点中找到直接连接到点 i 的点 l^* , 作为前一点,设置:

$$l = l^*$$

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

在实际应用中如路径导航、急救等领域,最短路径是基于实际的道路交通网络图,常用于解决路径距离最短的问题,要求能在响应时间内给出起点到终点的最短路径并能在道路网上显示路径。道路交通网络图

① 基金项目:浙江省教育厅计划项目(20051434)

与抽象的有向图相比,有自身的一些特点,其中最重要的是道路图中的路段(边)之间存在空间地理特性,本文研究利用这种空间特性来设计和实现最短路径算法,并在通用开发环境中实现算法并展示最短路径,以满足没有 GIS 功能支持的应用的需要。

3 道路网络模型的建立

3.1 道路网的基本组成和空间特性

道路网是由众多道路路段相交、相连而形成的纵横交织、错综复杂的道路网络图,路段和交叉路口是道路网络的基本组成单位,它们作为地理实体,具有以下空间特性:

(1) 几何特性: 路段可以看作是线型空间对象——弧,交叉路口可以看作是点型空间对象——点,它们都有一系列地理坐标来表示它们的位置信息,路段还有弧的长度和形状等几何信息。

(2) 空间关系: 空间关系主要包括距离、方向、连通和拓扑等关系,路段的走向、与交叉路口相交的路段、连通的路段等体现了道路网中空间对象间的空间相互关系。

在一个实际的道路图上研究最短路径问题,组成道路网的对象间的空间特性信息不容忽视。以一个城市道路交通图为例,在不考虑高架路、单向路等特殊情况的前提下,我们可以做出如下设定:

(1) 交叉路口为实节点,与交叉路口点相交的路段数不大于 4,不小于 2。

(2) 所有被路口节点分割的道路段称为路段,并假设路段为直线,即只记录路段的起点和终点的坐标数据。对于因路段弧度过大引起的误差,采用在弧的拐点增加交叉路口点的方法,来减小弧度(如图 1)。

(3) 最短路径的度量将以路段弧段的权值进行计算,如实际的路段长度等。

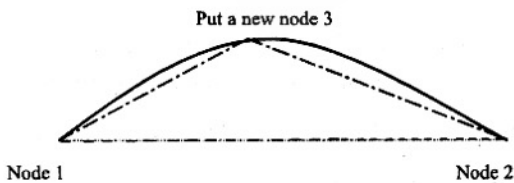


图 1 加路口节点减小弧度数

3.2 道路网络模型的建立

在实现 Dijkstra 算法的过程中,核心步骤就是从未标记的点中选择一个权值最小的弧段,即上面所述算法的(2)~(5)步。这是一个循环比较的过程,如果不采用任何技巧,未标记点将以无序的形式存放在一个

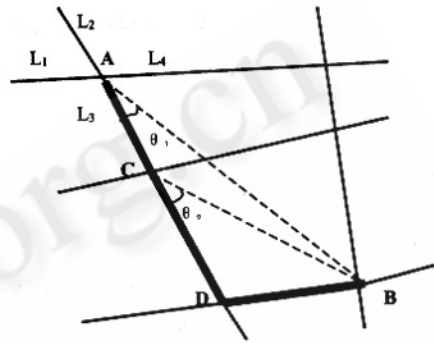


图 2 道路网络图最短路径算法

图 2 道路网络图最短路径算法示例

链表或数组中。那么要选择一个权值最小的弧段就必须把所有的点都扫描一遍,在大数据量的情况下,这无疑是一个制约计算速度的瓶颈。但在实际道路网络中,道路路段间的空间关系作为已知条件,可以充分加以利用以使算法实现的复杂度降低。

为了建立能反映道路空间特性的道路网络模型,我们设计了以下数据结构:

(1) 路口节点表。记录交叉路口节点的编号和坐标(x0,y0)信息。

(2) 路段表。记录路段的编号,路段两个端点的路口节点编号,路段权值(实际长度或其它度量);

为求最短路径的需要,两个表中的数据具有如下的参照和几何关系:

(1) 路段的两个端点是与路段相交的两个交叉路口点,因此路段的两个端点坐标是与路段相交的两个交叉路口点的坐标。

(2) 路段表中具有相同端点编号的不同路段,这些路段为相交在该编号路口节点的路段。

(3) 在某交叉路口相交的路段之间将有一定的夹角,夹角度数的求解可以遵循数学中求直线间夹角的相关定理。

为实际道路交通图建立以上模型工作量较大,为

此我们借助了地理信息系统软件 MapInfo, 在 MapInfo 中利用其丰富的绘图工具分别绘制出路段和交叉路口节点两个矢量图层, 两图层中的点、线对象分别满足 3.1 中的设定, 再把两个图层转换成 MIF 文件格式, MIF 文件中记录了每个路口节点和路段的坐标, 再经一定的程序处理后得出上述的路口节点表和路段表。另外还得出路段弧线坐标表, 利用它可以在 VC++ 中画出道路图, 用于最短路径的显示。

4 基于空间特性的最短路径算法

4.1 夹角最小算法

在实际生活中, 从某起点出发到某终点, 人们总是趋向终点行进以达到通过的路径最短的目的。这种行为体现了数学上两点间直线最短的原理。在道路网上任选两点, 这两点间最理想的最短路径是从起点到终点的直线, 但实际上这个直线存在的可能性很少, 但这条从起点到终点的直线却代表着一个路线的趋势, 从概率的角度, 顺着这个方向的某些路段组成最短路径的可能性较大。

夹角最小算法首先设定起点为当前节点, 在当前节点处, 在与当前节点相交的路段中, 取与当前节点到终点的直线夹角最小的路段, 把该路段作为最短路径的路径集合中的一个元素, 并把当前节点设置成该路段的另一端的节点, 继续寻找与当前节点到终点直线夹角最小的路段, 直到当前节点为终点为止。图 2 显示了用夹角最小算法获得 AB 间最短路径 ACDB 的过程。

采用夹角最小算法搜索的节点数等于最后得到的最短路径经过的节点数, 因此求解的速度快, 并且算法简单, 易于实现。在规整的道路图中, 或经过加节点减小弧度数的处理的道路网中, 通过该算法能快速得到正确的最短路径, 但在弧度大、拐点多的复杂道路网中, 算法求得的路径只是实际最短路径的一个最差解, 但即使在最差的情况下, 求得的最差解可以为改进算法提供一个好的约束条件。

4.2 改进算法

在每个目标点上, 考虑所有在该点的可能路径, 但利用最差解来作为路径选择的一个制约条件, 以此来达到完全正确的最短路径。改进算法的描述如下:

(1) 定义数据结构

Node[] 为具有以下结构的数组:

Node[].no 节点的编号;

Node[].len 节点的当前累加长度;

Node[].father 节点的父节点编号;

Turn 为当前访问到的 Node[] 数组的序号;

Num 为 Node[] 中加入的节点的总数。

(2) 初始化

Node[begin].no = 节点表中起点的编号;

Node[begin].len = 0;

Node[begin].father = "0";

Node[end].no = 节点表中终点的编号;

Node[end].len = 夹角最小算法求得的最差解;

Begin = 1; Turn = 0; Num = 1。

(3) 终止条件

如 $num < turn$, 表示 Node[] 数组中的节点已经处理完了, 终止循环。此时的 Node[end].len 就是求得的最短路径长度, 通过 Node[].father 可以找出最短路径经过的节点, 进而找到最短路径的组成路段, 程序终止。否则继续循环。

(4) 循环执行

turn 加 1, 根据节点表和路段表找出与 Node[turn].no 相交的所有边, 得到这些边的另一端的节点编号, 并逐个判断这些节点:

① 为终点。如该节点当前累加长度小于 Node[end].len, 则把该节点的当前累加长度赋给 Node[end].len, 并且 Node[end].father = Node[turn].no, 此节点不加入 Node[] 数组。

② 为已经访问过的节点。如该节点的当前累加长度小于原来已经访问过的节点, 则把该节点的当前累加长度赋给已访问节点的 len, 把 Node[turn].no 赋给已访问过的节点的 father, 并改变该已访问过节点的所有子节点的累加长度, 该节点不加入 Node[] 数组。

③ 该节点的累加长度加上该节点到终点的直线距离大于 Node[end].len, 则该节点不加入 Node[] 数组。

④ 不符合以上 3 点的节点, 加入 Node[] 数组, num 加 1, 并为 Node[num] 的 no、len、father 赋相应的值。

在改进算法中, 我们采用广度优先的方法来搜索以起点开始的图, 通过夹角最小算法得到的最差解作

为限定条件,使加入搜索节点列表中的节点大大减少,因为考虑了所有可能的路段,最后获得的最短路径是正确的最短路径。

4.3 算法实现

为了验证算法的可行性和效率,实验利用 VC++ 来实现了杭州市道路网的建模、最短路径的计算和显示。通过路段弧线坐标表画出城市路段的地图,在数据库中建立节点表和路段表以及由这两个基本表导出

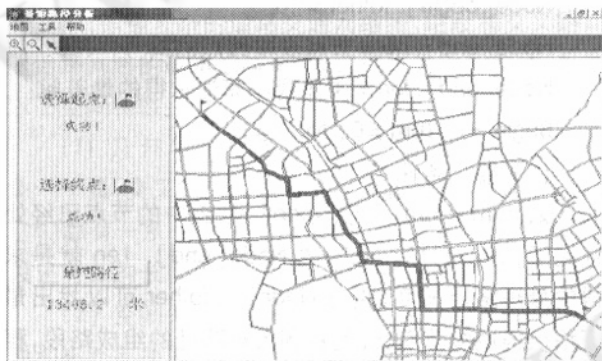


图 3 最短路径分析界面示例

的能方便程序编制的视图。在程序界面中,通过鼠标在图片框中点击来确定起点和终点,经转换和模糊搜索节点表确定最近的交叉路口节点位置,最后利用以上最短路径算法,获得组成最短路径的路段,为这些路

段进行粗线重画,显示在图片框中,并算出最短路径长度。求两点间最短路径的程序界面如图 3 所示。

5 结论

本文最短路径研究是基于实际道路网中本身具有的空间特性信息,路段和节点之间的空间关系通过建立保存有它们实际地理坐标数据的关系表来体现。实验表明,在不借助 GIS 工具的前提下,通过一般开发工具来获取最短路径是可行的。在杭州高教园区管理信息系统中,我们没有借助 GIS 软件,利用图片分割、AJAX 等技术,实现了高教园区地图及地图对象的查询和展示,并利用文中方法实现了高教园区中路径的分析和展示,取得了满意的效果。

参考文献

- 1 严蔚敏、吴伟民,数据结构,北京:清华大学出版社,1997. 186-192.
- 2 刘湘南、黄方、王平、佟志军,GIS 空间分析原理与方法,北京:科学出版社,2005. 293-301.
- 3 乐阳、龚健雅,Dijkstra 最短路径算法的一种高效率实现,武汉测绘科技大学学报,1999,24(3):209-212.
- 4 刘迎春,一种实用的最短路径求解算法,浙江工业大学学报,2000,28(2):169-173.