

GIS中使用改进的Dijkstra算法 实现最短路径的计算

唐文武 施晓东 朱大奎

(南京大学海岸与海岛开发国家试点实验室海洋地理信息系统室, 南京 210093)

摘要 地理信息系统中的空间网络分析有最短路径分析、资源分配分析、等时性分析等等, 而最短路径分析是其中关键的环节, 因而对其算法进行优化很有必要, 为此在传统的Dijkstra算法的基础上, 采用二叉堆结构来实现路径计算过程中优先级队列的一系列操作, 从而提高了该算法的分析效率. 讨论了地理网络数据的组织结构和最短路径的具体实现过程, 并引入了相关概念. 通过具体案例分析表明, 改进算法在提高网络系统空间分析效率方面是可行的.

关键词 Dijkstra算法 优先级队列 二叉堆

中图法分类号: P208 文献标识码: A 文章编号: 1006-8961(2000)12-1019-05

The Calculation of the Shortest Path Using Modified Dijkstra Algorithm in GIS

TANG Wen-wu, SHI Xiao-dong, ZHU Da-kui

(Marine GIS Laboratory of State Pilot Laboratory of Coast and Island Development,
Nanjing University, Nanjing 210093)

Abstract In GIS it is necessary to optimize the analysis function of the shortest path as the hinge of spatial network analysis, which includes shortest path analysis, resource allocation and isochrone, and so on. Here derived from the traditional calculating method, i. e. Dijkstra algorithm, the analysis procedure of the shortest path is improved by adopting the data structure of binary heap to complete the operation of priority queue. Initialization, Extraction, and Relxztion. The topological structure of geographical network data and the detailed implementing steps of the shortest path are also discussed. Furthermore, the visualizing calculation of the shortest path is completed by COM (Component Object Model) techniques, and the calculating procedure is encapsulated into the component of the geographic network class. The complexity analysis and the case of this algorithm shows that the modified algorithm is applicable to improve the efficiency of spatial analysis of the net system.

Keywords Dijkstra algorithm, Priority queue, Binary heap

0 引言

近些年来, 随着国际学术界加强了对GIS基础理论中空间关系的研究, 地理网络分析的研究也有了极大的发展^[1].

网络分析包括最短路径分析、资源配置、等时性问题等等. 但在进行网络分析时, 还要根据不同的网络, 建立起相应的网络分析模型. 这里, 所谓网络模型, 是指将现实中的地理网络实体, 抽象化为网络图

论理论中的网络图, 并通过图论中的网络分析来实现地理网络的最优化问题.

在网络分析中, 最短路径问题的分析是最基本的, 也是最关键的, 如今, 解决最短路径分析问题的方法虽然已经很成熟, 例如以Dijkstra算法为代表的宽度优先搜索方法、动态规划方法等等, 但作为网络分析的关键环节, 由于网络分析的存储量和计算量过于庞大, 算法的效率将直接影响到整个系统的性能. 因此对该算法效率的改进历来是人们研究的热点^[2-5].

本文针对上述问题, 在最短路径分析的经典算法

(即 Dijkstra 算法)的基础上,对网络的数据结构和计算方法作了一系列的改进,改进的 Dijkstra 算法较之 Dijkstra 算法,效率得到一定提高,系统的性能也有相应的优化,并实现了最短路径的可视化计算。

1 网络最优路径分析的实现方法

1.1 网络数据的优化组织

大家知道,空间网络数据是网络模型的基础,而由于空间网络数据具有空间数据基本的属性特征和空间特征,因而在 GIS 中,常将空间事物抽象成点、线、面等几何要素,并在点、线之间建立拓扑关联关系。例如,地理道路网络空间特征中的交叉路口坐标和道路位置坐标,是在地图上借助图形来识别和解释的,而在计算机中,则按照拓扑结构加以定义;而其属性特征有道路名称、道路距离、交通流量等等。

在实际工作中,尽管已拥有了空间数据,关键还要建立起空间数据结构。地理网络数据结构使用的是“弧段和结点”的数据结构,该数据结构乃是建立在图论的基础上的,即将地理网络表现为“由线串联而成的点群”,如今,一般向量式 GIS 均采用这种数据结构。在此结构下,结点可用来定义弧段之间的连接关系。在地理网络中,由于大多数弧段与弧段之间的交点是具有拓扑性的交点,因而由此建立起来的地理网络具有明显的拓扑特征。另外,由于拓扑结构是明确定义空间结构关系的一种数学方法,因此在 GIS 中,它不但用于空间数据的组织,而且在空间分析和应用中都具有重要的意义。一般具有了拓扑结构,就可以很快地确定一种地理实体相对于另一种地理实体的空间位置关系。由此可见,具有拓扑结构也是进行网络分析的必要条件^[5]。

对于网络数据的存储,传统的是采用图论中的邻接矩阵方法,其存储量为 $N \times N$ (N 为网络中结点数)。通常的地理网络,尽管结点很多,但与结点相关联的结点数目并不多,一般都为稀疏图,这样,该存储方法将浪费大量的空间,而且在计算时亦要花费大量的时间遍历无意义的的数据,故本文采用了邻接表的链式存储结构,其存储量为 E (E 为结点列表中,同结点关联的所有弧段数目),一般用邻接表表示图比用邻接矩阵法能节省大量的存储空间,尤其是在表示与结点和边相关信息较多的地理网络时。

1.2 最短路径分析的思想基础

在最短路径算法中,是利用了以下性质,即两结

点间的最短路径包含了其内部其它的最短路径,而算法实现的主要技术是松弛技术,而这种技术的实质是反复减小每个结点实际路径权值的上限,直到该上限等于最短路径的权值^[7]为止。在图论中,最短路径的求法采用了同图的宽度优先搜索方法类似的思想,如 Dijkstra 就提出了按路径长度递增的次序来产生最短路径的算法。

此算法(设网中权值均为非负)首先是把网中的所有顶点分成两个集合,即一个是将以 StartPoint 为源点的已经确定了最短路径的所有终点都并入 S 集合, S 集合的初态应只包含 StartPoint;另一个集合 T 则是尚未确定最短路径的顶点的集合, T 集合的初态则包含除源点 StartPoint 外的网中所有顶点,然后按各顶点与源点 StartPoint 间的最短路径长度递增的次序,来设置优先级队列 Q ,再通过优先级队列 Q 的相应操作,逐个把 T 集合中的顶点加入到 S 集合中去,并使得从 StartPoint 到 S 集合中各顶点的路径长度始终不大于从 StartPoint 到集合 T 中各顶点的路径长度。

1.3 最短路径的具体实现步骤

这里要实现的最短路径属于单源最短路径的问题。其实现过程如下:

输入:网络中进行路径分析的两个结点 StartPoint, EndPoint;

输出:两个结点 StartPoint, EndPoint 之间的最短路径树(ShortPathTree),即最短路径;

实现步骤为:

(1) 选择要进行计算的 StartPoint 和 EndPoint 两个结点;

(2) 对 StartPoint 和 EndPoint 两个结点进行连通分析,即采用宽度优先搜索方法,来快速判断这两个结点之间是否连通,也就是确定是否存在计算最短路径的必要。若连通,则进行(3),否则,退出计算;

(3) 使用改进的 Dijkstra 算法,计算 StartPoint 和 EndPoint 两个结点之间的最短路径(改进的 Dijkstra 算法的计算过程将在下面具体介绍和讨论);

(4) 经过对计算出来的最短路径树进行优化处理之后,生成最终的最短路径树(ShortPathTree),输出并退出。

2 改进的 Dijkstra 算法

在 Dijkstra 算法的计算过程中,通过设置优先

级队列 Q 的操作, 将集合 S 中的结点加入到集合 T 中, 一般的 Dijkstra 算法是采用线性数组来实现优先级队列 Q , 而这里采用二叉堆这种数据结构来实现优先级队列 Q 的一系列操作.

2.1 二叉堆实现优先级队列的有关操作

Williams 在 1964 年提出了堆排序方法, 该方法引入了堆这种特定的数据结构. 这里二叉堆结构可以被视为一棵完全二叉树, 而且其含义表明, 完全二叉树中所有非终端结点的值均不大于 (或不小于) 其左、右子结点的值. 除了用于堆排序之外, 二叉堆最常见的应用是作为高效的优先级队列. 该优先级队列是一种用来维护由一组元素构成集合 S 的数据结构, 而且这一组元素中的每一个都有一个关键字 key. 在分时计算机上, 进行作业调度和进行事件驱动的仿真器都要用到优先级队列, 而且通常采用二叉堆结构来实现优先级队列.

一般作用于优先级队列上的二叉堆的相应操作有:

Heapify(S): 即首先将集合 S 调整成二叉堆, 并设定其根结点具有最小关键字; 然后该操作从堆的根结点开始, 通过对当前结点的左右子树关键字的比较, 来调整相应结点在堆中的正确位置, 即通常所谓的“筛选”过程, 而且此操作为维持堆性质的关键.

Heap-Insert(x, S): $S \cup \{x\} \leftarrow S$, 即将元素 x 插入集合 S , 并调用 Heapify 将其调整成二叉堆. 该操作是首先将堆加以扩展, 即在树的最后一层加一片叶子, 然后遍历由新加的结点叶子到根的路径, 以找到放新元素的合适位置.

Heap-Extract-Min(S): 即抽取具有最小关键字的元素, 并调用 Heapify 将其调整成二叉堆. 该操作可通过对堆的 Heapify 操作来实现. 其运行时间主要花费在调整成二叉堆的操作上.

2.2 改进 Dijkstra 算法的面向对象实现方法

对改进的 Dijkstra 算法的实现, 采用了面向对象的方法, 对网络分析所用到的地理实体进行面向对象封装, 并实现最短路径的可视化计算.

2.2.1 最短路径分析对象的面向对象封装

面向对象的程序设计 (OOP) 是结构化语言的天然延伸. 由于 OOP 先进的编程方法, 会产生清晰而又容易扩展及维护的程序, 且一旦为程序建立了一个对象, 就可以在其它的程序中使用这个对象, 完全不必重新编制繁复的代码, 因而对象的重复使用可以大大地节省开发时间和切实地提高工作效率. 一个对象有 3 个突出特征, 即封装性、继承性、多态性.

这里构建了进行最短路径分析时所用到的地理对象类 TGEONET、TNODELIST、TARCLIST、TNODE、TARC、SHORTPATHTREE. 地理网络类之间的继承关系如图 1 所示:



图 1 最短路径分析类示意图

图中, TGEONET 为进行计算的地理网络类, 最短路径的计算过程就在其中实现; TNODELIST 为地理网络中的结点列表, 它存储了网络结点 TNODE 的信息; TARCLIST 为地理网络中的弧段列表, 它存储了网络弧段 TARC 的信息.

其中, TNODE 类封装网络中结点的信息, 包括顶点的标识 ID、顶点的 X 、 Y 坐标、同顶点连接的弧段列表 ADJARCLIST, 以及在求最短路径时用到的用来存放当前所求最短路径点的列表 CurPath;

TARC 类封装弧段的信息包括弧段的标识 INDEX、弧段的长度 LENGTH、起始点 NODEFROM、终结点 NODETO、组成弧段的节点坐标 XYS 及其节点数 POINTCOUNT;

SHORTPATHTREE 为源点 StartPoint 至 EndPoint 之间的最短路径树, 其中存储的是最短路径树中的结点信息.

2.2.2 改进的 Dijkstra 算法流程框图

改进的 Dijkstra 算法流程框图如图 2 所示.

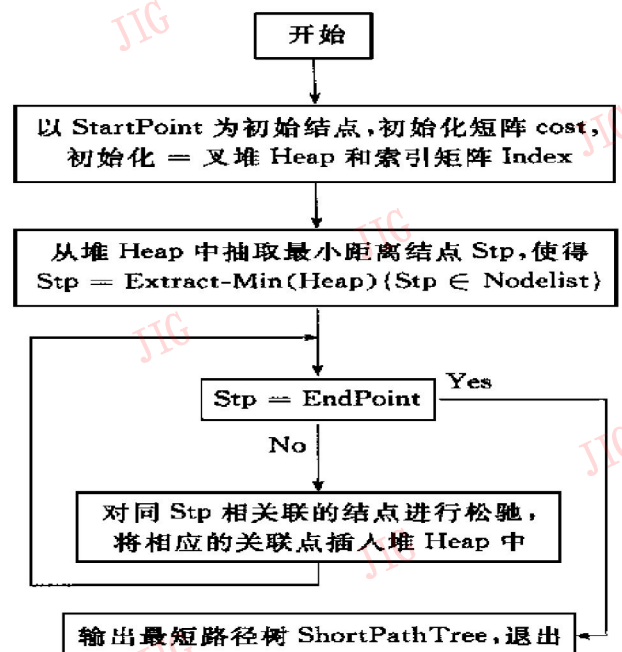


图 2 改进的 Dijkstra 算法流程框图

2.2.3 改进的 Dijkstra 算法具体实现步骤

(1) 初始化操作 首先搜索与源结点 StartPoint 关联的结点 Adj[StartPoint], 然后初始化结点列表 NodeList 中所有结点的权值 cost [Node i], 调用 Heap-Insert 方法实现初始化 Initialize 操作, 来初始化优先级队列 Heap, 同时创建堆 Heap 中结点与结点列表 NodeList 中结点相互关联的索引 Index.

(2) 抽取最短距离结点操作 即对优先级队列 Heap 的操作, 通过调用 Heap-Extract-Min, 选择结点 Node[j], 使得 $cost[j] = \min\{cost[i] \in Node[i] \in Nodelist\}$

其中, Node[j] 为当前求得的从 StartPoint 出发的最短路径终点.

(3) 松弛操作 对从 Node[j] 出发的结点 Node[k] 进行松弛操作, 松弛操作是通过 Decrease-Key 方法来实现, 即若 $cost[j] + cost[j, k] < cost[k]$, 则修改 $cost[k] = cost[j] + cost[j, k]$; 同时, 将结点 Node[k] 通过 Heap-Insert 加到优先级队列 Heap 中, 并相应更新索引表 Index. 而索引表中记录的是结点列表 NodeList 与二叉堆中结点之间的相对位置索引.

(4) 重复步骤 2、3, 直至 Node[j] = EndPoint.

(5) 结束.

2.3 改进的 Dijkstra 算法复杂性分析及相关优化讨论

Dijkstra 算法同宽度优先搜索算法相似, 要遍历从每一结点出发的所有结点, 最终生成的不仅是起点到终点的最短路径, 而且还求出起点到网络图中其它所有结点的最短路径, 实际上生成的是其它结点到起点的最短路径树. 由于传统 Dijkstra 算法是使用线性数组结构, 因此每次操作都要遍历整个结点列表 NodeList, 即顺序遍历整个最短路径树, 其整个算法的运行时间仅为 $O(N^2)$; 而使用二叉堆结构的改进 Dijkstra 算法则仅仅遍历二叉堆, 即仅遍历最短路径树中从根结点到当前进行操作的结点, 即遍历次数仅仅是二叉堆 Heap 的高度 $\lg(N)$, 故算法的执行效率大为提高, 整个算法的运行时间为 $O(E\lg N)$. 表 1 中列出了线性数组与二叉堆实现优先级队列的各种操作的最坏情况紧凑时间界. 从该表中可见, 采用二叉堆来实现优先级队列要比使用线性数组节省时间.

虽然采用二叉堆实现的优先级队列较之使用线

表 1 线性数组与二叉堆实现优先级队列的各种操作的最坏情况紧凑时间界列表

	线性数组	二叉堆
Initialize	$O(N)$	$O(N)$
Extract-Min	$O(N^2)$	$O(E\lg N)$
Decrease-Key	$O(N)$	$O(\lg N)$
Total Time	$O(N^2)$	$O(E\lg N)$

性数组结构要优秀得多, 但在对结点进行松弛操作的时候, 要用到二叉堆的查找 Search 操作, 然而二叉堆不能有效地支持查找操作. 针对这一点, 本文创建了二叉堆 Heap 中的结点位置与结点列表 Nodelist 中结点位置之间的索引 Index, 通过索引表 Index, 能够快速地对二叉堆中的结点在 Nodelist 中的位置进行定位, 这就大大减少了对结点进行定位时所花费的时间, 同时也进一步优化了算法.

3 最短路径的案例分折

本文在进行案例分析时所选用的开发平台为 Delphi4.0、Windows98, 并采用控件化编程的思想, 首先将最短路径分析过程封装到非可视化控件 GeoNet 中, 然后同其它 GIS 功能控件, 如 MapViewer, GeoView 等连接, 快速组装成最短路径分析系统(见图 3).

本文实验所采用的数据是美国某一地区的道路网络数据, 文件中一共有 4 788 条地理实体记录, 经初始化, 对所有记录构建网络所用时间为 10s 左右, 网络共有 2 964 个结点, 而在计算最短路径时, 使用改进的 Dijkstra 算法所花费的时间通常只有几秒钟, 而采用传统的 Dijkstra 算法, 计算最短路径一般要用 10s 多的时间. 由此可见, 改进的 Dijkstra 算法效率较高.

4 总 结

GIS 网络分析中, 最短路径分析过程在众多的地理网络最优化模型分析中扮演着重要的角色, 但由于过去的分析算法较复杂, 花时较多, 因而本文通过对传统的 Dijkstra 算法的改进, 即使用二叉堆结构来实现优先级队列的操作, 在一定程度上优化了最短路径的计算过程, 并降低了算法的时间复杂度, 使时间复杂度达到 $O(E\lg N)$, 而案例中实际数据测试也表明了该算法的可行性.

致 谢 由衷感谢侯玉国先生给予的无私指导

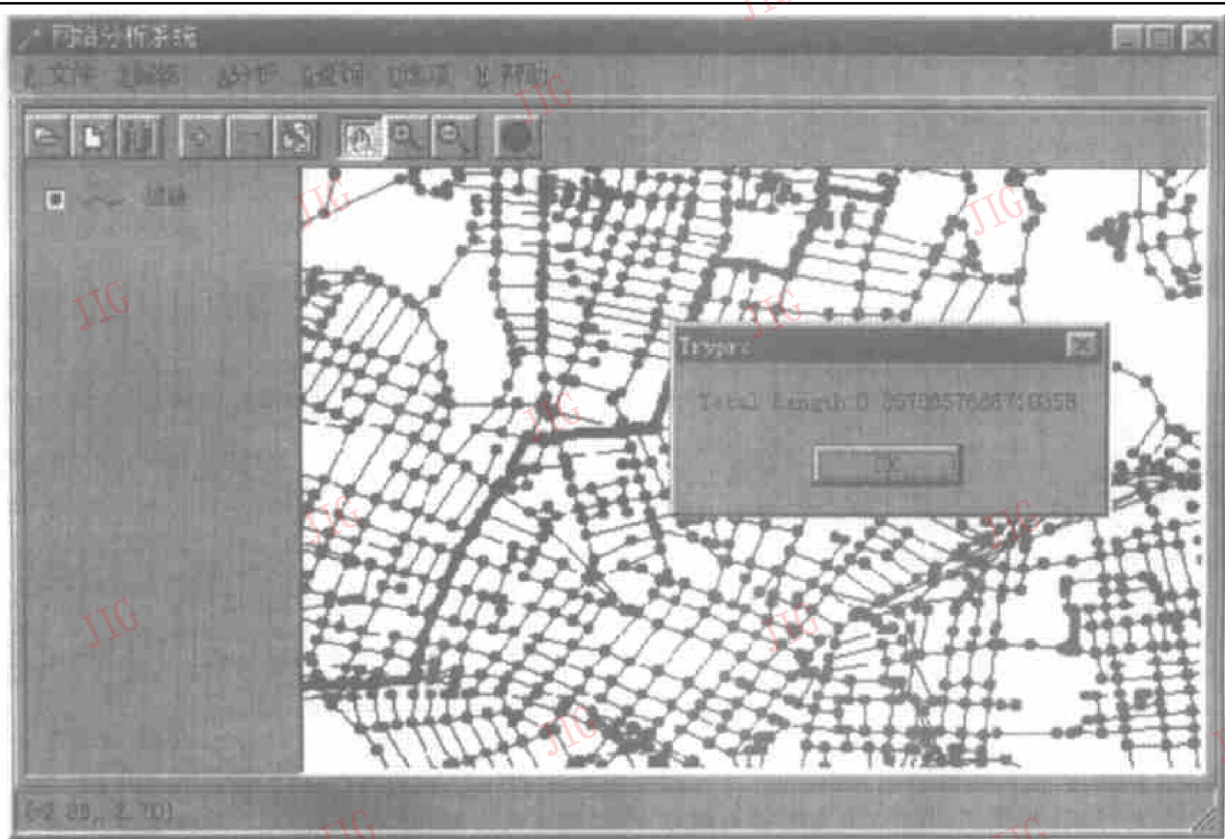


图 3 最短路径分析系统界面

参 考 文 献

- 1 陈军, 赵仁亮. GIS 空间关系的基本问题与研究发展. 测绘学报, 1999, 28(2): 95~ 101.
- 2 Michael Kuby *et al.* A minimax method for finding the k best differentiated paths. *Geographical Analysis*. 1997, 29(4): 298~ 313.
- 3 Miller Harbey J. Measuring spacing-time accessibility benefits within transportation networks: Basic theory and computational procedures. *Geographical Analysis*, 1999, 31(1): 1~ 26.
- 4 Stefanakes E, Kavouras M. On the determination of the optimum path in space. In *Spatial Information theory: A theoretical basis for GIS*. In: *Proceedings of International Conference COSIT '95*. Berlin: Springer, 1995: 241~ 257.
- 5 宫鹏. 城市地理信息系统: 方法与应用, 伯克利: 中国海外地理信息系统协会, 1996.
- 6 潘金贵等. 现代计算机常用数据结构和算法. 南京: 南京大学出版社, 1994.
- 7 王树禾. 图论及其算法. 合肥: 中国科学技术大学出版社, 1990.
- 8 Marco Cantu. *Delphi4.0 从入门到精通*. 北京: 电子工业出版社, 1999.



唐文武 1976 年生, 硕士生. 主要研究领域为空间分析理论、Web GIS、组件式 GIS.

施晓东 1967 年生, 工程师. 主要研究领域为计算机应用和同位素分析.

朱大奎 1935 年生, 教授, 博士生导师. 主要研究领域为海洋 GIS 应用、海洋地质.