

GIS 领域最短路径搜索问题的一种高效实现

王开义¹⁾ 赵春江¹⁾ 胥桂仙²⁾ 宋晓宇¹⁾

¹⁾(国家农业信息化工程技术研究中心, 北京 100089) ²⁾(中央民族大学计算机系, 北京 100081)

摘 要 目前在 GIS 领域, 对最短路径搜索问题的研究和应用较多, 其中最短路径搜索算法的效率问题是普遍关注和在实际应用中迫切需要解决的问题. 通过对基于 Dijkstra 最短路径搜索算法的优化途径的分析, 从算法本身和数据存储结构两个方面同时对此问题的解决方案进行了优化, 提出了直线优化 Dijkstra 算法, 并进行了必要的证明和适用条件论述. 此方案应用到“全国主要城市间公路信息查询”系统中, 取得了较为满意的效果, 同时也给出了相关的测试数据.

关键词 地理信息系统(420·3040) Dijkstra 最短路径

中图法分类号: TP391.41 **文献标识码:** A **文章编号:** 1006-8961(2003)08-0951-06

A High-efficiency Realization Way of the Shortest Path Search Problem in GIS Field

WANG Kai-yi¹⁾, ZHAO Chun-jiang¹⁾, XU Gui-xian²⁾, SONG Xiao-yu¹⁾

¹⁾(National Engineering Research Center for Information Technology in Agriculture, Beijing 100089)

²⁾(The Center University for Nationality, Beijing 100081)

Abstract There are many researches and applications about the shortest path searching in GIS field at present. Algorithmic efficiency of the shortest path searching is problem which has come to front and need to be resolved in application. In this paper, Author analyzes the optimization means based on the Dijkstra's shortest path algorithm and brings forward his optimization schemes beeline optimizing Dijkstra's shortest path algorithm from two aspects optimization of algorithm and data storage configuration. At same time, author discusses necessary testifying and application conditions in this paper. This scheme was applied in China Main City Highway Information Searching System, and has got satisfied effects. This paper also offers concerned testing data about Beeline Optimizing Dijkstra's Shortest Path Algorithm.

Keywords GIS, Dijkstra, The shortest path

0 引言

近些年来, 地理信息系统(GIS)在交通、公安、土地资源管理、城市规划等方面都得到了广泛且深入的应用. 这些应用领域中的地理信息系统经常涉及最短路径搜索问题, 例如城市公交系统的最短路径搜索系统, 公安系统的紧急出警和救助系统、城市供水、供电、供气管线的规划设计系统等. 本文只讨论一般公路交通网络中两结点间的最短路径搜索问题, 其他相关的问题, 如中国邮路问题等虽在实际应

用中也有较大的需求, 但暂不列入本文的讨论范围.

目前基于地理信息系统的最短路径搜索算法研究很多^[1~7], 其中 1959 年迪克斯特拉(Dijkstra)提出的单源问题算法是最适合拓补网络中两结点间最短路径搜索的算法之一, 本文将此算法称为“原始 Dijkstra 算法”^[8,9]. 后人在此基础上进行了大量的优化^[1~9], 本文在原始 Dijkstra 算法的基础上, 从核心算法和数据存储结构两方面进行了改进, 并应用到基于 WebGIS 的“全国主要城市间公路信息查询系统”, 通过实际测试, 速度和内存消耗两项指标都取得了比较满意的效果.

1 GIS 领域最短路径搜索算法的高效实现

1.1 最短路径搜索算法的优化途径

原始 Dijkstra 算法将网络结点分为未标记结点、临时标记结点和永久标记结点 3 种类型. 网络中所有结点首先初始化为未标记结点, 在搜索过程中和最短路径结点相连接的结点为临时标记结点, 每一次循环都是从临时标记结点中搜索距源点路径长度最短的结点作为永久标记结点, 直至找到目标结点或者所有结点都成为永久标记结点才结束算法.

在原始 Dijkstra 算法中, 临时标记结点无序地存储在无序表中, 这无疑成为 Dijkstra 算法的瓶颈, 因为每次在临时标记结点中搜索路径最短的结点时, 都要遍历所有的临时标记结点. 解决的办法就是将临时标记结点按照最短路径排序, 每个搜索过程不必全部遍历或者只较少地遍历临时标记点, 这也是目前各种基于原始 Dijkstra 算法的各种优化算法的重要出发点之一; 另外一个优化途径是尽量减少最短路径分析过程中搜索的临时结点数量, 从而尽快到达目标结点.

1.2 直线优化 Dijkstra 算法

原始 Dijkstra 算法一种行之有效的优化方法是: 减小算法中成功搜索的搜索范围, 以尽快到达目标结点. 其核心思想是: 在所研究的网络可以被看成平面网络的条件下, 将临时标记结点到源点的最短路径与本临时标记结点到目标结点的直线距离之和作为此临时标记结点的一个属性值, 这个属性值将作为从临时标记结点集合中选取永久标记结点的依据, 即选取此属性值最小的临时结点作为永久标记结点, 这种优化方法称为直线优化 Dijkstra 算法. 此方法使得 Dijkstra 算法的搜索方向智能地趋向于目标结点, 减少了算法中遍历的结点个数, 从而提高了搜索速度.

直线优化 Dijkstra 算法与原始 Dijkstra 算法的搜索过程比较示意图如图 1 所示.

由图 1 可见, 原始 Dijkstra 算法的搜索过程, 可近似为以源点为圆心的一系列同心圆, 搜索过程没有考虑终点所在的方向或位置, 在从源点出发的搜索过程中, 其他结点与终点被搜索到的概率是相同的; 直线优化 Dijkstra 算法的搜索过程, 可近似为以源点和终点为焦点的一系列同心椭圆. 因为永久标

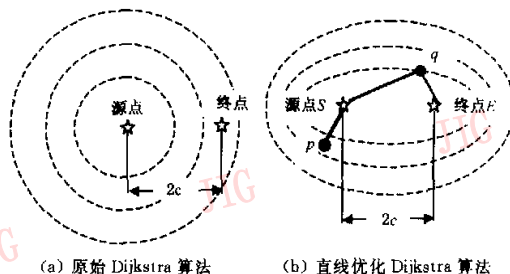


图 1 两种算法结点搜索过程示意图

记点的选取原则为: 当前结点距源点的最短距离与此结点到终点的直线距离之和最小者被选取为永久标记点, 所以其搜索过程明显趋向于终点, 搜索过程到达终点的速度明显快于原始 Dijkstra 算法, 搜索到的结点少于原始 Dijkstra 算法.

如图 1(b) 所示, 设 $S_p=2, S_q=5, pE=8, qE=2$. 从源点出发进行原始 Dijkstra 算法搜索, 首先选取结点 p 和 q 作为临时标记点, 因为 S_p 长度小于 S_q 长度, 所以选取结点 p 为永久标记结点, 再以 p 为新的源点继续进行下一轮搜索, 而结点 q 则是最后被淘汰的. 如果采用直线优化 Dijkstra 算法, 则因为 $S_p+pE=10, S_q+qE=7$, 所以结点 q 被选取为永久标记点. 如图 2 所示, L_1 为临时标记结点 P_1 距源点的最短路径长度, L_n 为临时标记结点 P_n 距源点的最短路径长度, D_1 为结点 P_1 到终点的直线距离, D_n 为结点 P_n 到终点的直线距离, P_1 和 P_n 的连线为 C . 原始 Dijkstra 算法永久标记结点的选取原则是: 如果 $L_1 > L_n$, 则选取 P_n 为永久标记结点, 如果 $L_1 < L_n$, 则选取 P_1 为永久标记点. 直线优化 Dijkstra 算法永久标记点的选取原则是: 如果 $L_1 + D_1 > L_n + D_n$, 则选取 P_n 为永久标记结点, 如果 $L_1 + D_1 < L_n + D_n$, 则选取 P_1 为永久标记结点.

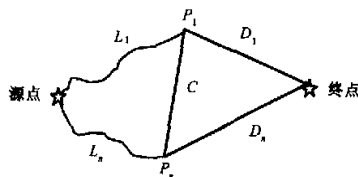


图 2 直线优化 Dijkstra 算法

直线优化 Dijkstra 算法永久标记结点选取原则的一个前提条件是 L_1 为 P_1 到源点的最短路径, 即 P_1 的最短路径值不会再被修改, 也即

已知: $L_1 + D_1 < L_n + D_n$ 证明: $L_n + C > L_1$

因为 $L_n + D_n > L_1 + D_1$
所以 $L_n + D_n - D_1 > L_1$
又因为 $C \geq D_n - D_1$
所以 $L_n + C > L_1$ 证毕.

直线优化 Dijkstra 算法特别适合于网络中弧的权值为弧长度,并且可以认为整个网络在同一平面内.在地理信息系统中,一般在较小的地理范围内,可以用平面代替地球表面;网络中两结点间的直线距离最短是此优化算法的另外一个前提条件.

直线优化 Dijkstra 算法优于原始 Dijkstra 算法的关键在于其最大搜索范围得到了明显的减小,从而使搜索速度得到提高,图 3 显示了两种算法的最大搜索范围.图中圆为原始 Dijkstra 算法的最大搜索范围,椭圆为直线优化 Dijkstra 算法的最大搜索范围.

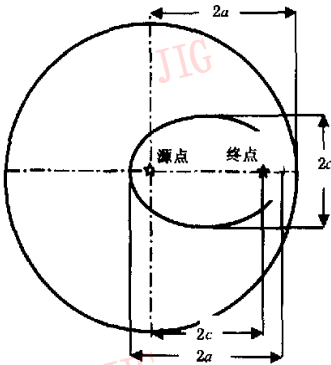


图 3 原始 Dijkstra 算法和
直线优化 Dijkstra 算法最大搜索范围示意图

设最短路径长度为 $2a$,则原始 Dijkstra 算法的最大搜索圆半径极限为 $2a$,直线优化 Dijkstra 算法的最大搜索椭圆长轴长度为 $2a$,椭圆上一点到两个焦点的直线距离之和为 $2a$,设其短轴长度为 $2b$,源点和终点为焦点.用圆和椭圆的面积来近似表示两种算法所搜索的结点数,其分别为

圆的面积: $S = 4\pi a^2$

椭圆面积: $S = \pi a^2 \sqrt{1 - e^2}, e = c/a$

因 $\sqrt{1 - e^2} < 1$,所以图 3 所示的圆和椭圆面积之比大于 4:1,即说明原始 Dijkstra 算法的最大搜索范围是直线优化 Dijkstra 算法最大搜索范围的 4 倍以上,表 1 中的数据进一步证明了这一点.

直线优化 Dijkstra 算法最大搜索范围与源点在网络中的位置有关.当源点接近网络边界或者处在

表 1 不同实现方案的 Dijkstra 算法所搜索的
永久标记结点数比较

起点—终点	原始 Dijkstra 算法	直线优化 Dijkstra 算法	两种算法 永久标记点 个数比值
北京—乌鲁木齐	19 466	4 902	4
上海—昆明	10 544	2 530	4
西安—武汉	3 131	548	6
西安—广州	14 580	2 975	5
武汉—重庆	6 803	778	9

网络边界上时,其搜索的结点数目可能不会明显地少于原始 Dijkstra 算法的.总之,直线优化 Dijkstra 算法搜索的结点数目要比原始 Dijkstra 算法少,具体的数量根据网络的具体情况不同而有所差异.另外,当源点和终点不连通时,两种算法的搜索范围是相同的,它们的搜索终止条件都是搜索到整个网络所有与源点连通的结点而未找到终点.在这种情况下,直线优化 Dijkstra 算法与原始 Dijkstra 算法搜索的结点数相同,但直线优化 Dijkstra 算法的开销要高于原始 Dijkstra 算法,因为直线优化 Dijkstra 算法要处理两结点间的最短距离计算问题.

1.3 最短路径搜索算法的选取

根据对原始 Dijkstra 算法优化方法的论述可知,其优化途径有两个:一是对临时标记结点进行排序,二是减小永久标记结点的数量.

对临时标记结点的排序选取堆排序方法^[8],将结点最短路径值作为堆排序的关键字,采用堆排序有如下考虑:

(1)充分利用了现有堆数据,大大降低了数据比较的频率.堆排序的主要运行时间消耗在建初始堆上,而对于 Dijkstra 算法来说,整个最短路径的搜索过程中进行的 n 次堆排序只需要建一个堆,从而明显克服了堆排序的主要缺点,使堆排序的优势更加明显地发挥出来.

(2)Dijkstra 算法中结点最短路径值的修改总是比原来的值小. Dijkstra 算法中采用小根堆即堆顶元素为关键字值最小的元素.堆中某元素的关键字(即结点的最短路径值)修改后,进行堆的重新整理操作时,只需要判断此元素是否需要和其父结点进行位置调整.所以 Dijkstra 算法中堆的重新整理操作要比传统堆的重新整理过程简单、高效.

(3)堆排序只需要一个记录大小的辅助空间.快速排序需要 $O(\log n)$ 的辅助存储空间,而归并排序则需要 $O(n)$ 的辅助存储空间.

基于以上的讨论,对原始 Dijkstra 算法(算法 1)、堆优化 Dijkstra 算法(算法 2)、直线优化 Dijkstra 算法(算法 3)、堆优化结合直线优化 Dijkstra 算法(算法 4)的速度测试结果如表 2。

表 2 4 种算法运行时间比较

单位:ms

起点	终点	算法 1	算法 2	算法 3	算法 4	路径长 (km)
北京	乌鲁木齐	190	140	60	50	3 236.67
上海	昆明	100	80	50	40	2 414.61
上海	长春	160	170	70	60	3 136.27
西安	广州	190	110	60	50	1 861.69
武汉	海口	200	140	180	160	+∞

实验数据采用全国 1 : 1 000 000 公路网,共有 22 320 个结点,55 772 个弧段。

操作系统:Windows 2000 Professional.

硬件设备:CPU: Intel Pentium MMX200,内存:128M,硬盘:6.4G.

最短路径搜索算法采用堆排序结合直线优化 Dijkstra 算法;临时标记结点采用堆来存储;永久标记结点的搜索采用直线优化 Dijkstra 算法,可进而达到从两个方面分别对 Dijkstra 算法进行优化的目的。

1.4 数据存储结构

最短路径搜索问题一般都采用结点-弧结构的存储结构表示。这一方面清晰地表达了网络的拓扑结构,另一方面更加便于 Dijkstra 算法的实现。根据堆优化结合直线优化 Dijkstra 算法的特点,设计的数据结构如图 4 所示。

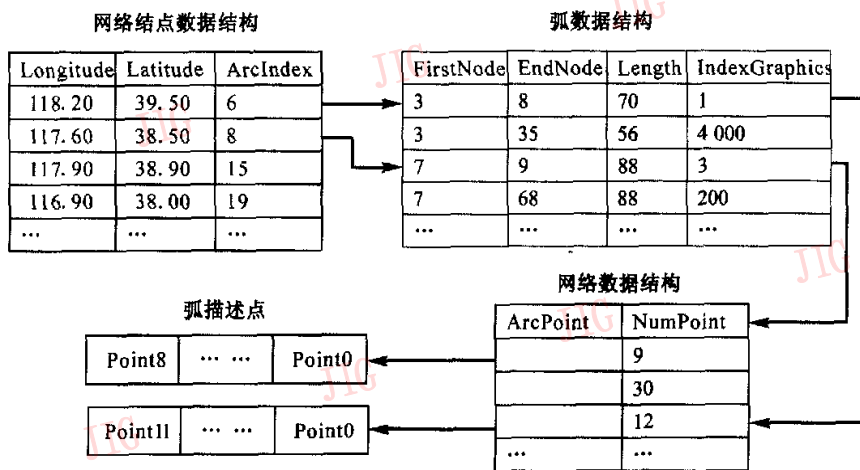


图 4 数据结构

以下对数据结构进行简单的描述:

(1)网络结点数据结构说明(PointRecord)

ArcIndex:以此结点为起点的第 1 个弧在弧数组中的下标

Latitude:结点纬度

Longitude:结点经度

(2)弧数据结构说明(ArcRecord)

FirstNode:弧的起点编号

EndNode:弧的终点编号

Length:弧的长度

IndexGraphics:此弧在网络数组中的索引(网络数组下标)

(3)弧描述点数据结构说明(Point)(包括弧结点在内的整个弧的描述点)

Latitude:结点纬度

Longitude:结点经度

(4)网络数据结构说明(GraphicRecord):

CountPoint:弧的结点个数

ArcPoint:结点数组指针

(5)最短路径结点数据结构说明(DijkstraPoint)

HeapID:此结点在堆中的编号

PreNode:最短路径中此结点的前一结点

ShortestLen:源点到此结点的最短路径长度

由图 4 可见,此数据结构完全是为最短路径搜索算法量身定制的。其能够提高效率的最突出一点是:网络结点数据结构存储了以此结点为起点的第 1 个弧的数组下标,这样就不需要遍历整个弧数组

来搜索以此结点为起点的弧;此外,弧数组中各个弧的信息,以弧的起始结点为关键字排序存储,所以起始结点相同的弧是连续存储的,这种结构极大方便了查找某一结点的所有连通结点,此数据结构的其他环节都按照这样的指导思想设计,使算法实现起来更加流畅。

最短路径结点数据结构为最短路径搜索算法中使用的结点数据结构,网络结点数据结构、弧数据结构和网络数据结构联合构成整个网络描述结构。在全国主要城市间公路信息查询系统的开发中,最短路径结点数据是每个客户请求线程都需要建立的数据结构,整个网络的描述结构只在服务器响应第 1 个客户的请求时建立,并存储在应用程序的数据堆中,其他客户的请求虽然都需要访问此数据,但不需要重新建立整个拓扑网络的描述数据。

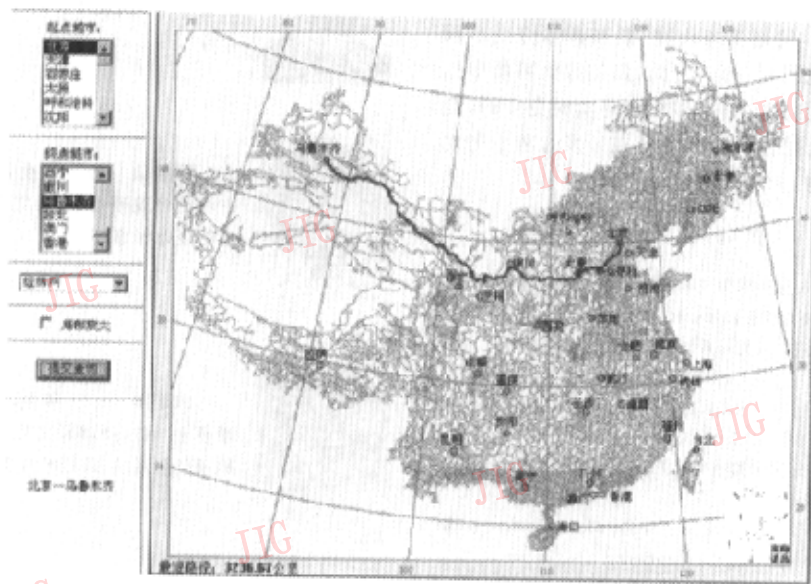
最短路径搜索数据文件,文件结构描述如表 3 所示。

表 3 Dijkstra 算法数据文件结构

文件区域	结构	数据类型	长度(字节)
文件头(36 字节)	文件标志 (File Type)	Int	4
保留位			32
	图的结点数 ($nPoint$)		4
数据(N 字节)	图的弧段个数 ($nArc$)	Int	4
	弧段数据记录	ArcRecord	$12 \times nArc$

2 应用实例

应用于全国主要城市间公路信息查询系统,采用的最短路径搜索算法是:堆排序优化算法与直线优化 Dijkstra 算法的结合,这样系统响应速度明显快于系统采用原始 Dijkstra 算法的响应速度,取得了较为满意的效果,系统运行界面如图 5 所示。



此图为例示意图

图 5 全国主要城市间公路信息查询系统——最短路线查询(北京—乌鲁木齐,经纬网叠加)

此系统在网络带宽为 56K 的环境下运行,除了第 1 次访问本系统稍有延迟外,用户进行各个城市间的最短路径查询操作感觉不到此系统的延迟。

3 结 论

直线优化 Dijkstra 算法是在平面条件下进行的

优化,从算法的速度、资源消耗和稳定性 3 方面综合考虑,是目前了解的各种 Dijkstra 最短路径搜索优化算法中较优秀的一个,其适用的网络应具有以下特征:

- (1) 网络可以近视为平面网络。
- (2) 具有确定的空间位置。
- (3) 弧的权值为弧的长度。

而堆排序优化 Dijkstra 算法则适合各种特征的网络。原始 Dijkstra 算法要求网络弧的权值为非负数,堆排序优化 Dijkstra 算法和直线优化 Dijkstra 算法当然也要遵循原始 Dijkstra 算法的要求。

地理信息系统中最短路径搜索功能的实现还需要根据不同应用领域的具体特征进行改进和优化,例如铁路领域的最短路径搜索问题等。除了实现过程中在数据结构和算法上要周密考虑,其他每一个细节问题也要根据具体情况采用效果最好的实现方法或者技术,才可以开发出综合性能很好的最短路径搜索功能模块。在具体编程中优化处理了许多细节问题,这些细节对整个系统的性能提高同样起到了推动作用。

目前地理信息系统中最短路径搜索的一些串行算法研究(包括本文的研究)已经取得了比较满意的效果,因此,进一步可以考虑进行并行计算的研究,以取得更大的突破。

此外,还要考虑网络中弧的权值问题。一方面,弧的权值可能不只是弧的长度,根据具体应用不同还需要考虑其他因素;另一方面,弧的权值可能根据时间等参数的变化而变化,即权值是实时更新的,全国主要城市间公路信息查询系统在开发过程中考虑了弧的权值问题,开发了相应的接口。

参考文献

- 1 徐业昌,李树祥,朱建民等.基于地理信息系统的最短路径搜索算法[J].中国图象图形学报,1998,3(1):39~43.
- 2 陆锋.最短路径算法:分类体系与研究进展[J].测绘学报,2001,30(3):269~275.
- 3 严寒冰,刘迎春.基于GIS城市道路网最短路径算法探讨[J].计算机学报,2000,23(2):210~215.
- 4 吴京,景宁,陈宏盛.最佳路径的层次编码及查询算法[J].计算机学报,2000,23(2):184~189.
- 5 Cherkassky B V, Goldberg A V, Radzik. Shortest paths algorithms: Theory and experimental evaluation [J]. Mathematical Programming, 1996, 73: 129~174.

- 6 Bertsekas D P. A simple and fast label correcting algorithm for shortest paths[J]. Networks, 1993, 23: 703~709.
- 7 Zhan F B, Noon C E. Shortest path algorithms: An evaluation [J]. Using Real Road Networks [J]. Transportation Science, 1998, 32(1): 65~73.
- 8 严蔚敏,吴伟民.数据结构(第2版)[M].北京:清华大学出版社,1997:159~281.
- 9 方世昌.离散数学[M].西安:西安电子科技大学出版社,1995:254~256.



王开义 1974年生,助理研究员,2001年获吉林大学硕士学位,主要研究领域为地理信息系统、图形图像处理等。



赵春江 1964年生,研究员,1994年获中国农业大学农学博士学位,主要研究领域为计算机农业应用、专家系统、精准农业等。



胥桂仙 1975年生,讲师,2002年获吉林工学院硕士学位,主要研究领域为图形图像处理等。



宋晓宇 1973年生,2002年获新疆大学硕士学位,助理研究员,主要研究领域为地理信息系统、遥感技术在农业中的应用。