

改进 Dijkstra 算法在 GIS 导航应用中最短路径搜索研究

董 俊¹ 黄传河²

(湖北第二师范学院计算机学院 武汉 430205)¹ (武汉大学计算机学院 武汉 430072)²

摘 要 研究 GIS 在电子导航系统应用中的最短路径搜索效率问题。在电子导航系统中对最短路径的搜索效率要求很高。随着城市发展交通线路剧增,传统的基于 Dijkstra 算法的 GIS 导航系统不能适应日益复杂的交通线路,存在最短路径搜索效率过低的问题。考虑到 GIS 空间分布的特性,提出了改进的 Dijkstra 算法用以解决 GIS 导航中的最短路径搜索问题。改进算法不仅避免了传统 Dijkstra 算法逐个节点遍历搜索,而且根据方向优先特性缩小搜索范围,大大减少了搜索工作量,并通过改变搜索节点存储的数据结构提高了最短路径的搜索效率。实验表明,这种改进算法较之传统算法能够有效提高最短路径的搜索效率,满足了电子导航系统对最短路径搜索效率的要求,取得了满意的结果。

关键词 最短路径,搜索效率,方向优先

中图法分类号 TP301.6 **文献标识码** A

Research on Shortest Path Search of Improved Dijkstra Algorithm in GIS Navigation Application

DONG Jun¹ HUANG Chuan-he²

(College of Computer, Hubei University of Education, Wuhan 430205, China)¹

(School of Computer, Wuhan University, Wuhan 430072, China)²

Abstract Research on the application of GIS navigation system is the shortest path search efficiency. The electronic navigation system has high demand for the shortest path search efficiency. With urban development, traffic lines increase, and the traditional Dijkstra algorithm based on GIS navigation system can not adapt to the increasingly complex traffic lines. The shortest path search efficiency is too low. Considering the GIS spatial distribution characteristics, this paper proposed the improvement Dijkstra algorithm to solve the shortest path search problem in GIS navigation. The improved algorithm not only avoids the case that traditional Dijkstra algorithm traverses the search by node, but also according to priority narrow search direction features range, greatly reduces the workload search, and through the change of the storage of data structure search node improves the shortest path search efficiency. Experiment indicates that compared with the traditional method the improved algorithm can effectively improve the shortest path algorithm of the search efficiency, and satisfy the shortest path search efficiency requirements of the electronic navigation system, and obtains satisfactory results.

Keywords Shortest path, Search efficiency, Direction is preferred

1 引言

地理信息系统^[1] (Geographic Information System, GIS) 是能够收集、管理、分析、操作与地理信息相关数据信息的计算机信息系统,被广泛应用于资源管理、环境监测、电力行业、城市规划等领域。随着计算机技术的发展以及城市交通的发展,电子导航成为人们生活所需的并引起人们重视的一项技术, GIS 因强大的数据处理和网络分析的能力被人们应用在电子导航系统中,并且成为当今的研究热点。GIS 网络分析系统中最关键的是最短路径的搜索问题^[2],最短路径的搜索效率直接决定了电子导航的效率,进而影响电子导航系统的服务质量。由于人们对电子导航系统的搜索效率有极高的要求,因此,在 GIS 系统中最短路径的搜索算法和搜索效率成为人们普遍关注和在实际应用中迫切需要解决的难点问题。

过去,解决 GIS 最短路径问题的算法有很多,其中最常

用也是较有效的是 Dijkstra 算法^[3],这得益于此算法较能适应网络拓扑的变化、程序设计简单、性能较稳定、通用性强等。Dijkstra 算法在 GIS 系统中解决最短路径问题是通过逐一搜索网络中的所有节点,来找到与源节点最近的节点的节点的方式,可见此算法的搜索计算量较大,并且此 Dijkstra 算法存储数据时采用的是邻接矩阵数据结构,占用的内存空间较大,影响了系统的处理效率。将 GIS 用于解决电子导航中起始节点与目的节点之间的最短路径问题时,传统的 Dijkstra 算法不能适应城市交通的快速发展,其繁琐的逐一搜索计算方式无法有效解决日益剧增的交通线路中最短路径的搜索问题,解决最短路径问题时搜索效率过低,不能满足人们对电子导航系统效率的要求,因此人们致力于研究高效解决最短路径问题的算法。

传统的 Dijkstra 算法搜索过程易懂、程序设计简单,但其较大的计算量成为此算法的瓶颈,为了解决传统 Dijkstra 算

法搜索效率不高的问题,提出改进的 Dijkstra 算法并应用于 GIS 导航中。改进 Dijkstra 算法考虑城市交通 GIS 网络的空间分布特性,根据 GIS 空间分布中,待搜索节点与起始节点连线方向和起始节点与目的节点连线方向夹角越小的节点被选取为下一步节点的概率越大,提出方向优先搜索策略,其通过计算方向夹角缩小节点搜索范围,避免了每次搜索都要逐一遍历所有节点,大大减少了搜索计算工作量,且占用的存储空间较少,提高了系统的工作效率,这样利用改进的 Dijkstra 算法提高了 GIS 中最短路径的搜索效率,有效满足了人们对电子导航效率的要求。

2 GIS 导航中最短路径搜索原理

地理信息系统(GIS)在解决实际问题时,将实际的实体对象抽象成点、线、面。点、线、面是 GIS 的 3 个重要元素^[4],此三元素结合可以表示出空间实体的位置、大小、形状等特征。GIS 应用于电子导航系统中,将城市中的交通道路分布抽象为交通网络,在此系统中 GIS 点、线、面 3 个元素分别表示的是:

- 1)点:又称节点,是两条或两条以上交通线路(道路)经过并在此处相交的点;
- 2)线:所有的交通线路抽象为线,具体为两个点(节点)之间的一段道路;
- 3)面:由道路线围成的封闭区域。

GIS 导航系统就是将获取的卫星遥感道路图像按照点、线、面三元素抽象转换为 GIS 空间关系图,同时将点、线、面的数据计算并存储在系统的内存单元中。其中点数据的存储是由坐标和关联线路数据组成的,具体的数据存储形式见表 1。

表 1 节点数据的存储形式

Mode ID	横坐标值	纵坐标值	关联线路
---------	------	------	------

线数据的存储是由起始点、终止点、线路权值数据组成的^[5],具体的数据存储形式见表 2。

表 2 线数据的存储形式

ARC ID	起始点	终止点	线路权值
--------	-----	-----	------

其中线路权值是线数据的关键数据,在导航系统中其值是线路的长度值。面数据的存储是由面序号、周边线、域内节点数据组成的^[6],其具体的数据存储形式见表 3。

表 3 面数据的存储形式

面序号	周边线路数据	域内节点数据
-----	--------	--------

将获取的卫星遥感图像抽象转换为 GIS 空间关系图,并将点、线、面数据存储^[7],以便进一步的最短路径搜索处理。图 1 就是一个转换完成的交通 GIS 空间关系图。

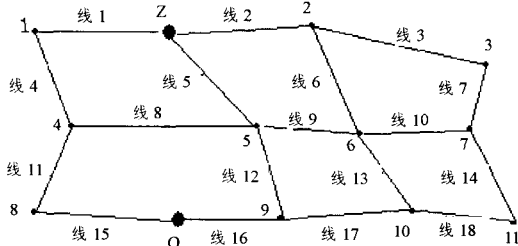


图 1 交通 GIS 空间关系图

图 1 中起始节点为 Q,目的节点为 Z,需要根据搜索算法

找到两节点之间的最短路径。传统的 Dijkstra 算法采用逐一搜索所有节点的方式来完成最短路径的获取,即从起始节点 Q 开始,计算所有节点与起始节点之间的线路权值,并通过比较找到最小权值的节点,记录此节点信息,完成一步移动,然后将下一步节点作为起始节点,重复上述逐一节点的计算,直至达到目的节点,则最终记录得到的途径节点信息集合就是所需的最短路径^[8]。由最短路径搜索效率的计算公式可知:

$$L = \frac{\log(\frac{1}{n}) \cdot t_0}{\alpha \cdot \ln(NT)} \times 100\% \quad (1)$$

式中, n 是所需要搜索的节点数目, t_0 是每个节点搜索的平均耗时,分母表示由遥感图像抽象转换成 GIS 空间关系图像的时间,由于每个节点的平均搜索耗时和同一幅遥感图像抽象转换所耗的时间变化不大,可以近似看做是常量,因此可以看出最短路径的搜索效率与所需搜索的节点数目直接相关,且成反比^[9]。由式(1)可以看出,传统的 Dijkstra 搜索最短路径的方法在节点较少的系统中搜索效率较高,且编程简单便于实现^[10],但是随着现代社会的发展,交通线路越来越多、越来越复杂,这种逐一搜索的 Dijkstra 算法处理复杂的交通线路时需要搜索较多的节点,直接造成最短路径的搜索效率较低,不能满足电子导航对搜索效率的要求。

基于上述分析,设想通过结合 GIS 的空间特性,将 Dijkstra 算法进行改进,来提高最短路径的搜索效率。改进 Dijkstra 算法考虑城市交通 GIS 网络的空间分布特性得知,待搜索的节点中与起始节点的连线方向和起始节点与目的节点的连线方向更接近的节点被选中为下一步节点的概率更大,由此提出方向优先搜索策略来缩小节点搜索范围,大大减少了搜索计算工作量,且占用的存储空间较少,提高了系统的工作效率,这样利用改进的 Dijkstra 算法提高了 GIS 中最短路径的搜索效率,有效满足了人们对电子导航效率的要求。

3 改进 Dijkstra 算法搜索最短路径

在 GIS 的空间关系图中,从起始节点到目的节点可能会有多条路径可以到达,但是根据实际的经验和 GIS 空间分布特性可知,要想到达目的节点,路径的大致方向肯定与起始节点和目的节点之间的连线方向一致,即方向之间的夹角较小。由此可知,搜索最短路径时,所要搜索的下一个节点与起始节点的连线方向和起始节点与目的节点的连线方向较小,即待搜索的节点中,与起始节点连线方向更接近的节点被选中为下一节点的概率较大,因此提出方向优先搜索策略,以缩小待搜索的节点范围。下面详细介绍改进的 Dijkstra 算法来解决最短路径问题。

3.1 数据存储定义和初始化

在 GIS 最短路径搜索系统中,需要定义的数据除上文提到的点、线、面的数据外,还有在搜索过程需要用到的数据存储方式。

1) u_0 固定起点:最短路径搜索过程中的最初起始节点,数据格式为表 1 中的数据存储方式。

2) S 标记节点集:在搜索过程中所有被选取的、途经的所有节点的集合。这些节点的集合就是最终的最短路径所途经的节点,对这些节点的存放是按照表 1 的形式存储的,但是需要将这些节点按照先后顺序连接成一个链表,以反映最短路径的信息。其链表数据的存储方式如图 2 所示。

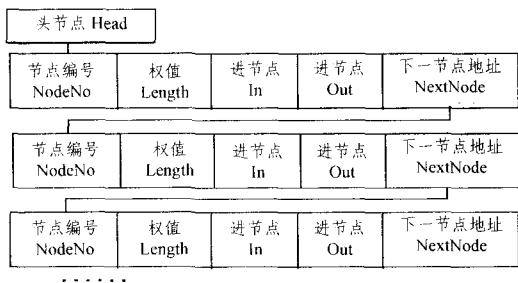


图2 标记节点集链表存储方式

3) J 等待搜索的节点:在栈中存放等待搜索的节点,这些节点的节点结构见表4。

表4 待搜节点的结构

节点编号 NodeNo	权值 Length	进节点 In	出节点 Out	下一节点 NextNode
-------------	-----------	--------	---------	---------------

定义完所有数据存储形式后,对固定起点赋值,将起始节点信息存储到固定起点的存储空间中,对标记节点集的存储链表空间清空,将待经过节点置0,为最短路径的搜索完成初始化准备。

3.2 最短路径的搜索实现

传统的 Dijkstra 算法是按照逐一遍历所有网络节点的广度优先搜索的方式完成最短路径搜索的,因其逐一搜索计算会耗费大量的计算时间和系统内存,可知在 GIS 网络空间广泛、交通道路复杂时,这种算法的效率是极低的。改进的 Dijkstra 算法是基于传统方法进行的改进,根据 GIS 空间特性采用方向优先的搜索原则,缩小搜索范围从而减去许多不必要的搜索网络分支,以保证搜索效率。

基于上文对数据存储的定义和初始化,下面详细介绍改进的 Dijkstra 算法搜索最短路径的步骤:

1) 选取起始节点,寻找与其直接相连的节点(即中间只有一条线路连接的节点),并将这些节点存储为等待搜索的节点形式 J ,对这些节点依次进行方向夹角计算,即选取一个待搜索节点计算其与起始节点的连接方向和起始节点与目的节点的连线方向,及两个连线方向的夹角,设待搜索的节点坐标为 (x, y) ,起始节点的坐标为 (x_q, y_q) ,目的节点的坐标为 (x_z, y_z) ,则两连线方向夹角的具体计算公式为:

$$k_1 = \frac{y - y_q}{x - x_q} \quad (2)$$

$$k_2 = \frac{y_z - y_q}{x_z - x_q} \quad (3)$$

$$\theta = \arctan \left| \frac{k_1 - k_2}{1 + k_1 \cdot k_2} \right| = \arctan \left| \frac{\frac{y - y_q}{x - x_q} - \frac{y_z - y_q}{x_z - x_q}}{1 + \frac{y - y_q}{x - x_q} \cdot \frac{y_z - y_q}{x_z - x_q}} \right| \quad (4)$$

通过计算出的方向夹角,比较并按照方向优先的原则选取夹角最小的节点作为下一步节点,并存放在标记节点集 S 中。将此节点作为新的起始节点进入下一步运算。

以图1所示的 GIS 空间关系图为例,与起始节点 Q 直接相连的是节点8和节点9,由图明显可以看出节点9与起始节点的连线方向比起始节点与目的节点的连接方向更接近,即连线方向夹角最小,因此选取节点9作为下一步节点,并将节点9作为起始节点,而开始的起始节点被存为固定节点 u_0 。

2) 判断上步得到的起始节点是否为目的节点,若是,则完

成最短路径的搜索,结束操作;若不是,则继续将此节点按照步骤1)中的操作进行搜索。

如图1中,根据方向优先原则,节点9与节点5之间的连线方向更接近起始节点9和目的节点的连线方向,因此选取节点5作为下一步节点。可知其搜索过程为:

起始节点 $\rightarrow$ 节点9 $\rightarrow$ 节点5 $\rightarrow$ 目的节点

3) 搜索结束,最终得到标记节点集 S ,将标记节点集中的节点数据存储成链表数据,则完成了最终的最短路径搜索,且最终得到的链表数据就是最短路径。

如图1中,最终搜索完毕后被存在标记节点集 S 中的节点为{起始节点,节点9,节点5,目的节点},此标记节点集中的节点数据建立的链表就是要寻找的最短路径。

综上所述,改进的 Dijkstra 算法通过考虑 GIS 的空间特性,提出方向优先搜索原则,其通过计算夹角选取最小夹角的节点去除了很多不必要的节点分支,缩小了节点搜索的范围,大大简化了计算量,保证了最短路径的搜索效率,较之传统方法效率得到了很大提高。

4 仿真实验及结果分析

为验证改进 Dijkstra 算法的性能,设计仿真实验进行验证。实验选取一张安徽省合肥市某区域卫星遥感图像,需要根据搜索算法搜索出合肥西站到生态公园之间的最短路径。图3为待搜索的卫星遥感图像。

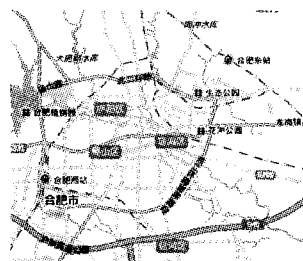


图3 待搜索的卫星遥感图像

分别采用传统的 Dijkstra 算法和改进的 Dijkstra 算法进行搜索实验,实验基于 Windows XP 操作系统,GIS 网络构建体系,采用 Visual C++ 6.0 可视化呈现 GIS 空间关系图,然后采用 C++ 编程语言编程实现两种具体算法,并使用 C++ 6.0 软件统计实验数据。

首先,根据待搜索的原始卫星遥感图像,利用 Visual C++ 可视化软件对原始图像进行 GIS 空间的构建,得到的构建 GIS 空间关系图如图4所示。

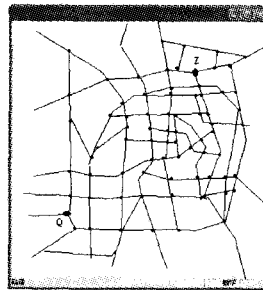


图4 GIS 空间关系图

然后,根据构建的 GIS 空间关系图,分别采用文中提出的新方法和传统方法进行最短路径的搜索,并且在实验中记录两种方法的实验数据和结果。

模型做了对比;建立了响应时间对多种资源参数的多元非线性模型,并通过具体实验数据和建模过程分析说明了该方法能较好地应用于这种多因素影响的复杂环境,能够以较快的速度挖掘出其间较复杂的多元关系,较好地预测了系统的性能变化情况。

需要说明的是,本文的实验是在局域网环境下完成的,所以网络传输的影响可以忽略不计。但在实际应用中,需要同时考虑服务器的处理时间和网络传输延迟,所建立的模型还将有助于分析服务的瓶颈并加以改进。对于运行规律多变且受时效影响较大的系统,需要持续监控并建立多阶段数学模型,后期将继续改进算法,实现系统的持续监控和动态性能评价。

参考文献

[1] Huang Y, Kintala C, Kolettis N, et al. Software Rejuvenation: Analysis, Module and Applications[C]//Proceedings of the 25th Symposium on Fault Tolerant Computer System, Pasadena, CA, 1995:381-390

[2] Garg S, Moorsel A V, Vaidyanathan K, et al. A Methodology for Detection and Estimation of Software Aging[C]//Proceedings of the 9th International Symposium on Software Reliability Engineering. Paderborn, Germany, 1998

[3] Grottke M, Lei Li, Vaidyanathan K, et al. An approach for estimation of software aging in a Web server[J]. IEEE Transactions on Reliability, 2006, 55(3): 411-420

[4] 徐文彬, 齐勇, 侯迪, 等. 基于时间序列分析的应用服务器性能衰

退模型[J]. 西安交通大学学报, 2007, 41(4): 426-429

[5] 赵旭峰, 钱存华, Nakagawa T. 带有周期检测性的软件预防性再生策略[J]. 计算机科学, 2009, 36(8): 158-160, 195

[6] Xie W, Hong Y, Trivedi K. Software Rejuvenation Policies for Cluster Systems under Varying Workload[C]//Proc. 10th Int'l Pacific Rim Dependable Computing Symp(PRDC 2004). Papeete, Tahiti, 2004: 122-129

[7] 赵天海, 齐勇, 沈钧毅, 等. 应用服务器多态老化模型和最优再生策略研究[J]. 系统仿真学报, 2007, 19(8): 1705-1709

[8] Thein T, Park J S. Availability analysis of application servers using software rejuvenation and virtualization[J]. Journal of Computer Science and Technology, 2009, 24(2): 339-346

[9] Silva L M, Alonso J, Torres J, et al. Using Virtualization to Improve Software Rejuvenation[J]. IEEE Transactions on Computers, 2009, 58(11): 1525-1538

[10] You jing, Sun Yu-qiang, Wang Hong-yuan. The relationship research between usage of resource and performance of computer system[C]//2009 WRI World Congress on Software Engineering. 2009, 3: 451-455

[11] Ferreira C. Gene expression programming: Mathematical modeling by an artificial intelligence(2nd Edition)[M]. Berlin: Springer-Verlag, 2006

[12] Jing Y, Lan Z, Hongyuan W, et al. JMeter-based Aging Simulation of Computing System[C]//Proc. Computer, Mechatronics, Control and Electronic Engineering (CMCE2010). Changchun, China, 2010: 282-285

(上接第 247 页)

最后,根据记录的实验结果数据,利用软件工具统计出每种方法完成最短路径搜索所需要搜索的节点总数目,并根据式(1)得到两种方法的搜索效率,将所有的实验结果列表比较,并根据实验结果进行分析。

表 5 所列实验结果数据对比表。

表 5 实验结果数据对比表		
算法	搜索节点总数	搜索效率
传统算法	63	87.7%
改进算法	17	96.1%

由上述实验结果对比分析可知,传统的 Dijkstra 算法在搜索最短路径时需要逐一搜索所有节点,在此实验中完成最短路径的搜索需要搜索的节点总数目为 63 个,可见巨大的搜索量增加了计算压力,使得计算量很大且系统的存储压力较大。由式(1)的搜索效率计算公式可知,搜索的节点数目与搜索效率成反比,因此传统方法逐一搜索的方式直接造成最短路径的搜索效率不高,搜索效率仅为 87.7%,不能满足人们对电子导航系统的实时性要求。而考虑 GIS 空间特性,提出了方向优先的搜索策略,其通过计算连线方向夹角,并选取夹角较小的节点方式去除许多无效的搜索,缩小了搜索的范围,进行此次最短路径的搜索只需要搜索 17 个节点便可完成,大大减少了计算量,使得最终的最短路径搜索效率提高到 96.1%,取得了满意的结果,可见改进的 Dijkstra 算法能够有效提高最短路径的搜索效率,满足电子导航系统的要求。

结束语 提出了改进的 Dijkstra 算法并应用于 GIS 导航中。改进 Dijkstra 算法考虑城市交通 GIS 网络的空间分布特性,提出方向优先搜索策略来缩小节点的搜索范围,大大减少

了搜索计算工作量,且占用的存储空间较少,提高了系统的工作效率,从而提高了 GIS 中最短路径的搜索效率。实验表明,这种改进的 Dijkstra 算法能够保证最短路径搜索的效率,有效满足人们对电子导航效率的要求,具有较高的使用价值。

参考文献

[1] 王卫强,孙强. 求图中受顶点数限制的所有最短路径的算法[J]. 计算机工程与设计, 2008, 29(7): 1754-1757

[2] Han Y. A note of an $O(n^3/\log n)$ time algorithm for all pairs shortest paths[J]. Information Processing Letters, 2008, 105(08): 114-116

[3] 陈益富, 卢潇, 丁豪杰. 对 Dijkstra 算法的优化策略研究[J]. 计算机技术与发展, 2006, 16(9): 73-75

[4] 叶品勇, 都洪基, 沈曦. Dijkstra 算法在最佳抢修路径计算中的应用[J]. 继电器, 2006, 34(12): 39-41

[5] 王凌, 段江涛, 王保保. GIS 中最短路径的算法研究与仿真[J]. 计算机仿真, 2005, 22(1): 117-120

[6] Pallotino S. Shortest path methods: complexity, interrelation and new propositions[J]. Networks, 2008, 14: 257-267

[7] Deo N, Pang C Y. Shortest2path algorithms: taxonomy and annotation[J]. Networks, 2007, 14: 275-323

[8] 库向阳, 史经俭, 罗晓霞. 栅格数据模型中附有条件的最短路径算[J]. 计算机应用, 2008, 28(4): 856-859

[9] 王红梅, 胡明. 基于直接/间接邻边概念的最短路径算法[J]. 计算机应用, 2010, 30(5): 1297-1299

[10] 李小鹏, 郁滨, 李亚敏. 基于 MapX 最短路径搜索算法研究[J]. 计算机工程与设计, 2009, 30(22): 5225-5228

[11] 魏秉铨, 李林, 胡峰. 普适环境下的智能地理信息服务研究[J]. 重庆邮电大学学报: 自然科学版, 2009, 21(5): 667-671