

GIS 网络分析的图简化方法研究

王杰臣¹, 张伟², 毛海城³

(1 南京大学 城市与资源学系, 江苏 南京 210097; 2 江苏省城乡规划设计研究院, 江苏 南京 210008;

3 南京市建设委员会, 江苏 南京 210008)

Study of Graph Simplifying Method in GIS Network Analysis

WANG Jie-chen¹, ZHANG Wei², MAO Hai-cheng³

(1 Department of Urban and Resources Science, Nanjing University, Nanjing 210097, China; 2 Institute of Urban Design and Planning of Jiangsu province, Nanjing 210008, China; 3 Nanjing Municipal Construction Committee, Nanjing 210008, China)

Abstract Network analysis is one kernel subject in GIS space analysis. The number of practical networks' points and arcs is the main cause to effect the operation speed. In this paper, starting from the view of economizing in memory space and increasing operation speed, the united structure of point-arc has been used to express network, and the authors provided several kinds of methods to simplify the graph in advance. That includes to justify the graph's connected characteristics; identify tip vertex and cut vertex, etc. A more simply network will be created and that would increase the operation speed greatly. And the practicing method is also given here.

Key words: GIS; network analysis; graph

摘要: 网络分析是GIS空间分析功能的核心问题之一。在算法无关前提下,网络的节点与弧段数量是影响运行速度的主要原因。从几个方面将复杂的网络实体简化,即对图预先进行连通性判定、消除图中梢顶点、搜索关键路径等,从而获得较简洁的网络模型,提高网络分析的速度与效率。

关键词: 地理信息系统; 网络分析; 图

1 问题的提出

网络分析是GIS空间分析功能的核心问题之一。根据应用问题性质的不同,GIS中网络分析方法也很多,许多问题都有专门的高效的算法,如最小成本计算、位址选择模型、节点连通性等问题基本上等价于最短路径问题,可以采用国内外公

认较好的Dijkstra算法^[1~7];许多网络流分析问题可以采用线性规划方法解决^[5,6]。然而有些较为复杂的应用问题,如多个约束条件下的路径选优,需要针对一条完整的路径进行多个目标值的综合判断,而不能从路径的局部信息作出判断。这就需要进行图的遍历,对每条可选路径都要进行判定与比较。通常有两种遍历图的路径:深度优先

收稿日期: 1999-11-22; 修回日期: 2001-03-01

基金项目: 地理信息科学江苏省重点实验室开放基金(2000104)

作者简介: 王杰臣(1973-),男,安徽太湖人,现为南京大学城市与资源学系博士后研究人员。主要从事地理信息系统、遥感的理论与应用研究。

(DFS)和广度优先(BFS)搜索^[4, 7, 8, 11, 12]。但当图的节点或弧段数过多时,会因可选路径的“组合爆炸”导致搜索时间急剧增长^[8, 12]。这种情况下需要对图作一些“预处理”,主要包括以下方面:进行连通性判断将图划分为几个相互不连通的子图;

逐层剥除悬挂枝以简化网络,因为任意一条不重弧的路径都不可能历经悬挂点与悬挂枝(起始点与终止点除外);寻找关键路径,将一个连通子图划分为若干片,各片之间通过惟一的割弧连

通,这可使搜索局限在几个子片内部进行,降低搜索深度,提高搜索效率。本文中笔者分别讨论了上述问题的求解算法,并给出相应的实现方法。

2 数据结构约定

表示图的数据结构主要有节点邻接矩阵^[2, 4, 9]及GIS中常用的点、弧数据结构^[10, 13, 14]。本处采用后者,本文所论述的图简化算法的表述均采用此约定,表1为该结构的C语言描述。

表1

Tab 1

struct ARC (弧结构)	struct POINT (点结构)
{	{
int ArcNo; //弧编号	int PointNo; //节点编号
int start_pt_no; //起始点编号	float x, y; //地理坐标
int end_pt_no; //终止点编号	int keyD; //割点标志
int keyD; //割弧标志	int tipD; //梢点标志
int tipD; //悬挂弧标志	int depth; //对某起始点的最小深度
int depth; //对某起始点的最小深度	int subNo; //所在的不连通子图编号
int subNo; //所在的不连通子图编号	int cutNo; //所在连通子图内
int cutNo; //所在连通子图内	int NeartArcNum; //关联弧数量
float *attrib_value; //弧的各种属性值	int *NearArcNo; //各关联弧的编号
}	float *attrib_value; //节点的各种属性值
	}

与采用节点邻接矩阵描述图的方法相比,该结构形式上稍复杂但数据冗余度较低,且易于表达具有多重属性的节点与弧段^[13, 14]。它除了记录图上节点与弧段关联关系外,还增加了一些属性项。以ARC结构为例,这些属性项为割弧标志、悬挂弧标志、对某起始点的最小深度、所在子图编号、所在片编号等。可以看出,该数据结构实际上是在GIS软件所建立的图的拓扑结构基础上扩充得到的。以当前应用较广泛的GIS软件ARC/INFO为例,弧段与节点关联关系及其属性记录在文件AAT.DBF中,该文件为标准DBF格式数据库文件,易于添加新属性域,从而由ARC/INFO实现数据管理与维护。

3 图的连通性

连通性是图论的一个重要概念,如果图 G 中某两个顶点 v_i 和 v_j 之间存在路径,则这两点在 G 中就叫做连通的;如果 G 中每对顶点之间都存在一条路径,则图 G 是连通的^[11]。在网络分析中,当起始点与终止点分别在不连通的子图内时,让计

算机去搜索最优路径是没有意义的,而且此时的搜索费用最大,必须从起始点开始在其子图内搜索全部路径,最后才发现没有通路。如果事先对图进行连通性分析,将其划分为几个子图,各节点、弧段给一标志数记录它所在子图编号,则可避免以后的网络分析中出现这类无意义的盲目搜索。可以证明遍历原图与遍历几个相互不连通子图的等价性。设图 G 由相互不连通的子图 G_1, G_2, \dots, G_n 构成,显然, G 的任一子图 G_i 内的遍历路径 s 也是 G 内的一条路径(子集与全集关系);同时,对图 G 内的路径 s ,设其起始点 v_0 在子图 G_i 内,根据路径内节点的连通性可知, s 的每个节点均在子图 G_i 内,因此 s 必然是子图 G_i 的遍历路径之一。也就是说,图 G 内的路径 s 必为某一子图 G_i 内的路径,子图 G_i 内的路径也必为原图 G 内的路径,由此可见,遍历原图与遍历几个相互不连通子图是等价的。

本研究中我们采用相邻节点低值传递方法(图1)进行连通性判定,其行为与病毒传染极为相似^[14]。步骤是:给每个节点一个标志变量,并

以节点编号作为初始值(每个节点开始时均被认为不连通); 每个节点对相邻节点进行“ 低值传递”, 传递后的标志数以相邻节点的较小标志数代

替; 重复执行上述过程 N 次直至系统稳定, 当不再有节点标志数发生改变时终止。

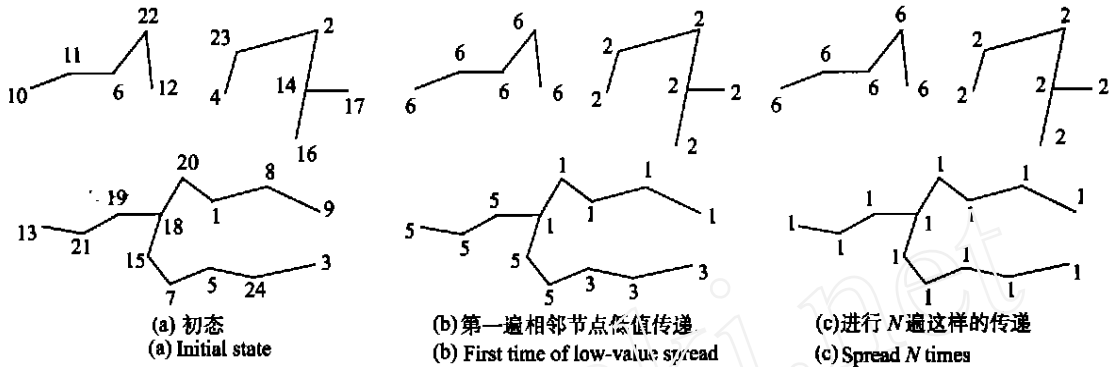


图 1 相邻节点的低值传递过程

Fig 1 Process of low-value spread between adjacent vertex

最后获得的节点标志变量必然具有以下特征: 连通节点具有相同的标志数, 不连通的节点标志数不同。这也表明同一子图内的所有节点具有相同的标志数。对弧段来说, 处理方法完全相同, 只需在每次修改标志数时, 同时修改弧段的标志数即可。可以看出, 步骤中的重复次数 N 为图的任意两个连通节点的邻接阶数(最小步数)中的最大值。以图 2 为例, 该图中弧段数为 6 364 条, 但其最大深度仅为 74, 故循环传递 74 遍即可使系统达到稳定并终止循环。表 2 为算法的 C 代码。

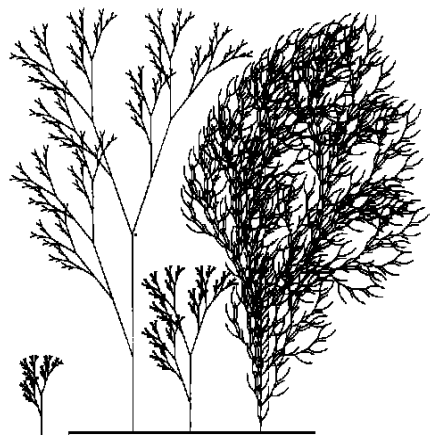


图 2 原始网络图

Fig 2 The source network

表 2

Tab 2

```

for (i= 0; i< 节点数 pNum; i+ + )pt[i].subNo= i; //初始化
for ( ; ; )
{ int SpreadTime= 0; //标志数发生改变的次数
  for (int j= 0; j< 节点数 pNum; j+ + )
  {
    for(int k= 0; k< 点 j 的邻接弧数 pt[j].NearArcNum; k+ + ) //邻点传递
    {
      B= 点 j 的第 k 邻弧所指向的节点;
      if (点 B 与 j 的标志数不同: pt[B].subNo != pt[j].subNo)
      { SpreadTime+ = 1;
        pt[B].subNo= pt[j].subNo = 两者间的较小数值
      }
    }
  }
}
if (SpreadTime= 0) break; //若不再有标志数发生改变, 表明系统稳定, 终止循环
}

```

4 悬挂节点与悬弧

悬挂节点是指关联弧数等于 1 的点, 该关联弧就称为悬弧^[11]。在路径选取问题中, 一般地, 一条优选的路径不会历经同一弧或节点两次以上。悬挂点处只有一条流入弧而没有流出弧(或相反), 因而优选路径不可能历经这类节点与弧段, 可见这类节点与弧段是可以消除的, 除非起始或终止节点本身是悬挂点, 此时需要保留这两个节点及相关弧段。此外, 悬挂点的剥除是逐层进行的, 首先消除的是最外层悬挂点, 之后会有一些新的节点成为悬挂点, 需要不断消除, 直到整个网络变成一个没有“梢”的仅由多边形和割弧构成的网络。图 2 所示为一原始网络图, 图 3 为消除悬挂点及悬弧后的网络。表 3 为逐层消除悬挂点及悬弧

的过程的 C 代码。

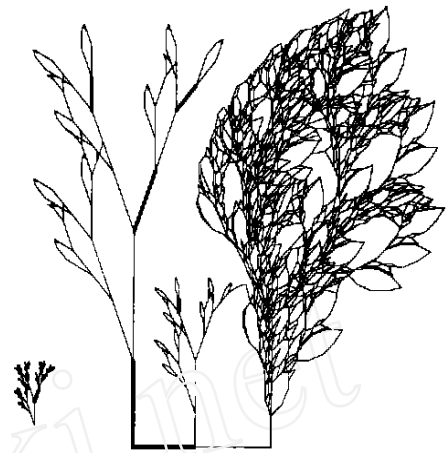


图 3 消除悬挂点及悬弧后的网络
Fig 3 Network without tip vertex and pendant edge

表 3

Tab 3

```

for(i= 0; i< 节点数 pNum; i+ + ) pt [i] tip D= 0; //初始化(tip D= 0 为非悬挂点)
for( ; ; ) //逐层消除
{ int RemoveNum = 0;
  for(i= 0; i< 节点数 pNum; i+ + ) if( pt[i] tip D = 0) //层内循环判断
  { int PendantNum = 0; //用于记录点 i 已有的悬挂弧数量
    for(int j= 0; j< 点 i 的邻弧数 pt[i].NearArcNum; j+ + )
      if(弧 j 为悬弧: arc[j] tip D= 1) PendantNum + = 1;
    if( pt[i].NearArcNum - PendantNum = 1) //除了已经判定出的悬弧外, 若仅剩
    { RemoveNum + = 1; //一条弧, 则它必为此层内的悬弧
      pt[i] tip D = 1; //设定该点及邻弧的标志
      for(int j= 0; j< pt[i].NearArcNum; j+ + ) arc[pt[i].NearArcNo[j]] tip D = 1;
    }
  }
  if(RemoveNum = 0) break; //若没有标志数发生改变, 表明系统稳定, 终止循环
}

```

5 割点与关键路径

图 G 有顶点 v_i , 如果图 $G-v_i$ 由比 G 更多数目的片组成, 则 v_i 是 G 的割点^[11], 下面的定理给出了割点的等价定义: 顶点 v 是图 G 的割点, 当且仅当存在两个不同于 v 的顶点 u 和 w , 并使 v 位于每条 $u-w$ 路径上^[11]。同样地可以定义割弧, 利用割弧可以将连通图划分为若干片, 而割弧则是各连通片间的唯一路径, 或称关键路径。

设图 G 为有 m 条割弧的连通图, 其顶点数和边数分别为 n 和 e_0 。 G 去除割弧后的图 G 由 k 个互不连通的子图 G_1, G_2, \dots, G_k 构成, G_i 的顶点数和边数分别为 n_i 和 e_i 。以 DFS 和 BFS 进行图的遍历为例(两者的时间复杂度相同), 利用 2 维数

组或邻接表结构表示图的存储结构时(本文约定的数据结构实际上是邻接表结构的扩展), 遍历 G 的时间复杂度分别为 $O(n^2)$ 、 $O(n + e)^{[4]}$, 而遍历子图 G_i 的时间复杂度为 $O(n_i^2)$ 、 $O(n_i + e_i)$ 。当给定搜索源 P 和搜索目标 Q 时, 可首先确定 P, Q 所在子图以及子图间的关键路径, 从而将图的遍历缩小到必经的若干子图内, 遍历的总时间复杂度为 $O(n_j^2)$ 、 $O(n_j + e_j)$ (其中 $1 \leq j \leq k$, 且 G_j 为 PQ 必然穿越的子图), 由于 $n_i \leq n, e_i \leq e$, 故总时间复杂度比直接进行图 G 的遍历低。以图 4 为例(图中每个节点代表一个子图, 弧段为子图间的关键路径), 其中子图总数为 22, 任一遍历路径必经的子图数为 1~ 6, 若各子图的复杂度相

同,则采用邻接表结构表示图时,分片搜索的时间复杂度可降低到直接搜索的 $1/22 \sim 6/22$ 。对于路径搜索问题,如利用 Dijkstra 算法搜索最短路径,其时间复杂度为 $O(n^2)$,此时分片搜索将更显著地提高效率(呈指数方式变化);一些特定的路径问题需要判断每条完整路径而不能采用 Dijkstra 算法,此时可选路径数随搜索深度增加而急剧增加,发生“组合爆炸”(设搜索深度为 d ,节点的平均广度为 w ,路径组合数 N 为 w^d 。当 $d=20, w=4$ 时, $N=1.2 \times 10^{12}$),而图的分片可降低搜索深度,避免进入没有搜索价值的片中进行无意义搜索,从而提高搜索效率。

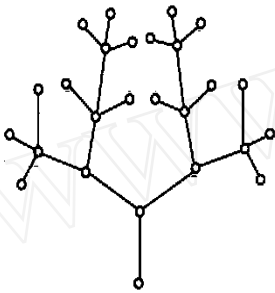


图 4 图的块划分

Fig 4 Graph separat

割弧判定条件很简单:将该弧断开,判断它的两个端点是否仍然连通。在具体实现中,需要对每条弧进行判定,当网络的弧段较多时,这需要较长时间。因此,我们分两步来进行判断:首先利用有界深度优先搜索法^[12]快速检测可疑弧段,以该弧的一个端点 P 为始点,向外搜索 n 步(限定的较小值),判断能否到达另一端点 Q 。该过程可编制为一个函数: `fast_detect (int ArcNo, int Depth)`,在主调程序中不断加大限制深度(`depth`)的值进行检测,这样可疑弧段被层层检测出来,最后获得“最可疑”的弧段。对于这些弧段,需要进行确切性判定,判断它是否是真正的关键弧,方法与前述的连通性判断相同,即采用邻接节点低值传递方法进行(图 3 中粗线所示为关键路径)。

6 实例与讨论

本文探讨了几种图的简化处理过程及其实现方法,从执行效率看,图的连通性、悬挂节点与悬挂弧的判定比较迅速;割点与关键路径的判定花费时间相对较多,本文例图的处理时间为 12 s。需要说明的是,文中所指的悬挂节点或悬挂弧“消除”并

非“删除”,而只是赋予该节点或弧段一个标志数,表明它是“悬挂”的,因此不会破坏原有的拓扑关系与数据精度。这样,在进行网络分析如最短路径分析中进行图的遍历时,遇到这类弧段时不必继续向下盲目搜索。图 5 为本研究采用的一个测试图件,该图中节点数和弧段数分别为 3 261、6 498。图中粗线所示为利用 Dijkstra 算法获得的一些最短路径,在原图不进行分块、分片的情形下,搜索时间平均达 23.82 s,反之搜索时间仅为 0.16 s(CPU 为 300MHz)。利用有界深度优先搜索法进行路径搜索时,当搜索深度 D 为 10~18 时,路径遍历时间分别为 0.17 s、0.39 s、0.99 s、2.69 s、7.47 s、20.76 s、58.55 s、166.09 s、473.12 s(本处测试路径遍历时间的方法是,设定搜索源 P 和目标 Q ,且使 $P-Q$ 经过的弧段数超过搜索深度 D ,这样做将无法搜索到 $P-Q$ 的路径,但可获得搜索全部深度不超过 D 的路径的时间)。该结果表明,搜索深度每增加 1 步,时间耗费近似地增加到原来的 3 倍(耗费时间的递增率与图上节点的平均出度有关,图 5 中节点平均出度约为 4,由于不重弧搜索的限制,因此每个节点下一步的可选分支数仅为 3),当搜索深度达到 18 以上时,搜索时间已难以被用户接受。对于较大规模的网络,该限制深度很难满足应用需要,而将图进行分片处理后,片内采用较小的限制深度,对整个网络而言,该限制深度实际上已被“放大”,从而提高这类问题的处理效率和处理能力。

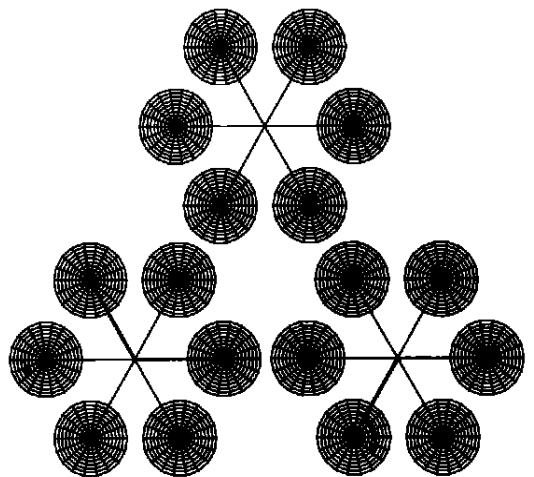


图 5 搜索实例一

Fig 5 Sample graph No. 1

需要指出的是,对于 GIS 中的网络问题,本文所作的研究还只是非常初步的,在应用时尚存

在一些局限,主要问题是,即使进行了图简化,仍可能存在较为复杂的片。笔者认为,在以下方面还有待进一步研究:

1 如何进行没有悬弧、割弧的连通图的简化,即本文讨论的简化过程之后进行进一步的图简化。这类图具有网络状(而不是具有分枝的树状结构)外观(例如,一个 $M \times N$ 的grid网),在图内进行遍历时仍可能有较大的搜索深度,现有搜索技术很难完成穷举式遍历。

2 研究图简化后的图搜索算法。实际上,本文探讨的是图搜索之前图的“预处理”过程,在图的遍历过程中,不同搜索算法的执行效率也有很大差异。当前常见的图搜索算法包括深度优先搜索法、广度优先搜索法、有界深度优先搜索法、代价驱动搜索法、局部择优搜索法、全局择优搜索法等^[8,12],在类型可分为分治法、动态规划方法、贪心法、回溯法、分枝限界法等^[7]。这些算法各有优缺点,其中深度优先搜索和广度优先搜索是一种盲目搜索,执行效率稍低但通用性强;其他算法实际上是这两种算法基础上的改进,在搜索过程中加入了启发性“知识”,对特定问题可显著提高搜索效率,但由于需要先验知识而降低了通用性,有些算法只能针对特定性质的问题,当目标函数不具备“局部最优则全局最优”等性质时,这些算法并不适用。

7 结 论

通过本文讨论的几种图的简化处理过程,可以将复杂的网络实体转化为较简洁的网络模型。最后的网络由几个相互不连通的子图构成,同一子图内,通过割弧将其划分为若干片;并且所有的悬挂点与悬弧都被消除。这样,在整个网络中进行路径搜索的源问题可以转化为在各子片内搜索的问题,而各子片的复杂度远小于初始网络,这在路径优选等网络分析中,尤其是针对较大规模的网络,可较大地提高处理效率和处理能力。

参考文献:

- [1] LI Xiumu. Fundamental Graph Theory [M]. Wuhan: Press of Engineering College of Mid China, 1982. (in Chinese)
- [2] LU Kaicheng. Graph Theory and It's Application [M]. Beijing: Tsinghua University Press, 1984. 67-70. (in Chinese)
- [3] GUO Renzhong. Spatial Analysis [M]. Wuhan: Press of Wuhan Technical University of Surveying and Mapping, 1997. (in Chinese)
- [4] YAN Weiming, WU Weiming. Data Structures [M]. Beijing: Tsinghua University Press, 1992. (in Chinese)
- [5] GAN Ying'ai, TIAN Feng, et al. Operational Research [M]. Beijing: Tsinghua University Press, 1997. (in Chinese)
- [6] DU Duanfu. Operational Graph Theory [M]. Beijing: Beijing Aeronautics University Press, 1990. (in Chinese)
- [7] ZHU Hong, CHEN Zengwu, DUAN Zhenhua, et al. Algorithm Design and Analysis [M]. Shanghai: Shanghai Science and Technical Document Press, 1989. (in Chinese)
- [8] SEHLDT H. C. Language and Artificial Intelligence [M]. Beijing: HOPE Computer Technical Cooperation of CAS, 1991. (in Chinese)
- [9] GONG Jiehui, BALING, GAO Jianmei. Improvement of Algorithm of Shortest Route and the Object-Oriented Practicing Method [J]. Journal of the PLA Institute of Surveying and Mapping, 1998, 15(2). (in Chinese)
- [10] CONG Shi, LIU Jiping, ZHANG Jiaqing. Study of GIS's Network Analysis Method [J]. Remote Sensing Information, 1998, (12). (in Chinese)
- [11] SWAMY M N S, THULASIRAMAN K. Graph, Network and Algorithm [M]. Beijing: Higher Education Press, 1988. (in Chinese)
- [12] TU Xu-yan. Artificial Intelligence and It's Application [M]. Beijing: Press of Electronic Industry, 1988. (in Chinese)
- [13] WANG Jiechen, MAO Haicheng, YANG De-zhi. United Structure of Point-Arc for Network Graph and It's Application in GIS's Shortest Path Searching [J]. Acta Geodaetica et Cartographica Sinica, 2000, 29(1). (in Chinese)
- [14] WANG Jiechen, YANG De-zhi, ZHANG Wei An. Improvement Algorithm of Shortest Route Analysis [J]. Journal of the PLA Institute of Surveying and Mapping, 1999, 16(4). (in Chinese)