GIS 空间分析中两种改进的路径规划算法

Two Improved Path Planning Algorithms in GIS Spatial Analysis

邱育红　（上海电视大学南汇分校 上海　201300）

摘要：通过对经典 Dijkstra 算法和启发式搜索的分枝算法的分析，分别对它们进行了改进。在 VC 下编制程序进行了实验，表明改进的 Dijkstra 算法可以减少大量的无关节点的计算，使其时间复杂性得到降低，同时也减少了运算空间开销；改进的分枝算法则可以提高搜索到最优路径的成功率。

关键词：路径规划　GIS　空间分析　Dijkstra 算法

1　引言

空间分析是 GIS 的一项十分重要的任务和最具有特色的功能，是基于地理目标的位置和形态特征的空间数据分析，其目的是提取和发现隐含的空间信息或规律，是空间数据挖掘和知识发现的基本方法之一。空间分析不是简单地通过“检索”、“查询”或“统计”从地理数据库中提取时空信息，而是利用各种空间分析模型及空间操作对地理数据库中的空间数据进行深加工，进而产生新的知识。空间分析是空间数据的空间特性与非空间特性的联合分析，亦即拓扑与属性数据的联合分析。路径分析是 GIS 的基本功能，其核心是最优路径的求解。所谓最优路径是指网络两个结点之间阻抗最小的路径。“阻抗最小”有多种理解，如基于单因素考虑的时间最短、费用最低、风景最好、路况最佳、收费站最少和经过乡村最多等。和基于多因素综合考虑的风景最好且经过乡村最多，或时间较短和路况较佳且收费站最少等。最优路径的求解算法有几十种如基于贪心策略的最近点接近法、最优插入法，基于启发式搜索策略的分枝算法，基于局部搜索策略的对边交换调整法以及广泛采用的 Dijkstra 算法等。本文主要讨论 Dijkstra 和启发式搜索的分枝算法。

2　Dijkstra 算法的问题和改进

2.1　Dijkstra 算法的基本步骤

最短路径是指在网络 D(V,A) 中，找出从起点 v_s 出发到终点 v_e 的累计行程最短路径。Dijkstra 算法是解最短路径问题的一种算法，它不仅可以找到最短的 (v_s,v_e) 路径，而且可以给出从起点 v_s 出发到图中所有节点的最短路径，算法过程如下：

第一步：初始化，起点 s 设置为 $d_s=0$，p_s 为空；其它所有点 i，$d_i=\infty$，记 $p_i=?$（未知）；标记点 $k=s$（s 为起点的编号）。

第二步：距离计算，计算从所标记的点 k 到与其直接相连的所有其它未作标记的点 i 的距离 l_{ki}，并令 $d_i=\min[d_i, d_k+l_{ki}]$。第三步：选取下一点，从上述结点集中，选取 d_i 最小所对应的点为最短路径中的下一连接点 j，并作标记。

第四步：找到 j 的前一点，从已标记的所有点中，找到直接连接点 j 的前一点 $l*$，并令 $j=i*$ 作为前一点。

第五步：标记点 j。如果所有的点均已标记，则最短路径寻找结束；否则，记 $k=j$，转第二步继续。

2.2　Dijkstra 算法应用举例

利用 Dijkstra 算法求图 1 中结点 A 到其它各结点的最优路径。根据算法的实现步骤，其计算过程可归纳为表 1 所示。从表 1 可以看出，A 到 P 的最短路径为 A－C－F－J－M－O－P，且 A 到 P 的距离 d(A,P)＝18.3。在求 A 到 P 最短路径过程中，A 到其余各点的最短路径也相应求。

出。若以计算一次 $d_i=\min[d_i, d_k+l_{ki}]$ 为计算单位，则利用 Dijkstra 算法计算 A 到 P 的最短路径时所需要的计算次数为 $T(P)=15+14+\cdots\cdots+2=119$ 次。

表 1　采用 Dijkstra 算法求解 A 到其它各结点最优路径的过程

序号	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	- - -	4.2	3.4	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
2	- - -	4.2	3.4/A	∞	9.0	6.9	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
3	- - -	4.2/A	- - -	8.6	8.3	6.9	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
4	- - -	- - -	- - -	8.6	8.3	6.9/C	∞	∞	11.9	10.9	∞	∞	∞	∞ ∞	∞	
5	- - -	- - -	- - -	8.5	8.3/8	- - -	∞	10.3	11.2	10.9	∞	∞	∞	∞	∞	∞
6	- - -	- - -	- - -	8.6/B	- - -	- - -	11.5	10.3	11.2	10.9	∞	∞	∞	∞	∞	∞
7	- - -	- - -	- - -	- - -	- - -	- - -	11.5	10.3/D	11.2	10.9	13.5	13.7	∞	∞	∞	∞
8	- - -	- - -	- - -	- - -	- - -	- - -	11.5	- - -	11.2	10.9/F	13.5	13.7	13.1	∞	∞	∞
9	- - -	- - -	- - -	- - -	- - -	- - -	11.5	- - -	11.2/E	- - -	13.5	13.7	13.1	∞	∞	∞
10	- - -	- - -	- - -	- - -	- - -	- - -	11.5/D	- - -	- - -	- - -	13.5	13.7	13.1	∞	∞	∞
11	- - -	- - -	- - -	- - -	- - -	- - -	- - -	- - -	- - -	- - -	13.5	13.7	13.1/J	∞	16.1	∞
12	- - -	- - -	- - -	- - -	- - -	- - -	- - -	- - -	- - -	- - -	13.5/H	13.7	- - -	18.0	16.1	∞
13	- - -	- - -	- - -	- - -	- - -	- - -	- - -	- - -	- - -	- - -	- - -	13.7/H	- - -	15.9	16.1	∞
14	- - -	- - -	- - -	- - -	- - -	- - -	- - -	- - -	- - -	- - -	- - -	- - -	- - -	15.9/L	16.1	18.7
15	- - -	- - -	- - -	- - -	- - -	- - -	- - -	- - -	- - -	- - -	- - -	- - -	- - -	- - -	16.1M	18.3

2.3 Dijkstra 算法的不足

在地理信息系统中，网络模型的规模常常较大，结点数成千上万，并且对网络模型的查询要求实时性。虽然，Dijkstra 算法在理论上可行，但是该算法在求解从起点到某一终点的最短路径过程中，时间复杂性为 $O(n^2)$。在求解网络系统中多点对乃至所有结点之间的最短路径，可重复 Dijkstra 算法多次或 n 次，时间复杂性为 $O(n^3)$。因此，对于海量数据的地理信息，传统的 Dijkstra 算法计算量较大，时间花费较多，效率非常低。

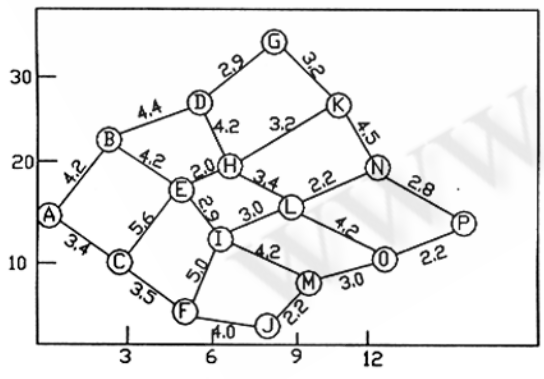

图 1　最优路径计算网络图

2.4 改进 Dijkstra 算法的基本思想和实现

表 1 中的数值大多数是无穷大，都是无用的运算，如果结点数很多，将及其浪费运算时间。由于 $d_i = \min[d_i, d_k + l_{ki}]$，i 结点是否在上次已经被计算出最短路径未知，当前结点 i 是否与结点 k 相连也未知，也就是 l_{ki} 未知，这时 d_k 是已知，故本次计算的 d_i 到底是不是无穷大，取决于上一步 d_i 数值和 l_{ki} 的数值。从表达式可以看出，只要这两个数值不都是无穷大，本次计算的 d_i 就不会无穷大。所以在上面 Dijkstra 算法实现第二步时，先判断一下，只要原来的 d_i，l_{ki} 的数值中至少有一个不是无穷大，才进行下面的计算 $d_i = \min[d_i, d_k + l_{ki}]$，这样就保证了当预见 d_i 是无穷大时，不对它进行计算，避免大量无效的计算，提高了搜索效率。

利用改进的 Dijkstra 算法求图 1 中结点 A 到其它各结点的最优路径，同 Dijkstra 算法基本一致，只是表 1 中所有无穷大标记的部分在改进 Dijkstra 算法中被省去了，利用改进的 Dijkstra 算法计算 A 到 P 最短路径时所需计算次数为 T(P) =47 次。由此可见，改进的 Dijkstra 算法确实减少了计算量。

2.5 实验对比

为了更好地说明改进 Dijkstra 算法的有效性，利用 VC 语言自行编制了最短路径搜索程序，并对 5 幅网络地图进行计算，见表 2。

从表 2 可以看出，两种算法计算量有很大区别，改进的 Dijkstra 算法较之经典的 Dijkstra 算法在计算量方面有很大幅度的减少，而且这种减少程度在结点数目增加时会变得越来越明显，对于 GIS，由于地图都很大，使用改进 Dijkstra 算法的改进效果将非常显著。

表 2　改进 Dijkstra 算法和经典 Dijkstra 算法计算次数

结点数	弧段数	经典 Dijkstra 算法	改进 Dijkstra 算法	百分比
16	24	119	47	39.5%
32	55	465	134	28.8%
43	75	861	234	27.2%
62	111	1830	441	24.1%
78	139	2926	540	18.5%

3　分枝算法的问题和改进

3.1　分枝算法的基本思想

分枝算法在人工智能中是一种典型的启发式搜索算法。通过选择合适的估价函数，指导搜索朝着最有希望的方向前进，以求得最优解。分枝算法中，关键是求估价函数：f(n) =g(n) +h(n)。其中，g(n)是从起点 vs 到当前结点 n 已付出的代价，h(n)是从当前结点到目标结点 ve 的代价估计函数，必须保证 h(n)≤h*(n)(其中 h*(n)是从当前点到目标点的实际最小代价)。

3.2　分枝算法的步骤

(1) 给定起始结点标记，对它的没有标记过的子结点进行扩展。

(2) 对每一个子结点计算评价函数值，按评价值的大小进行排列，找出评价值最小的结点，并给它作标记，如果当前结点就是目标结点，则停止搜索。

(3) 否则，对最新被标记的结点进行第 2)步处理，并记录最短路径。

3.3　分枝算法分析

分枝算法是利用对问题的了解和对问题求解过程的了解，寻求某种有利于问题求解的启发信息，从而利用这些启发信息去搜索最优路径。它不用遍历整个地图，而是每一步搜索都根据启发函数朝着某个方向搜索，当地图很大很复杂时，它的计算复杂性大大优于 Dijkstra 算法，是一种搜索速度非常快、效率非常高的算法。但是它也有缺点，启发性信息是人为加入的，有很大的主观性，直接取决于操作者的经验，对于不同的情形要用不同的启发信息和启发函数，且它们的选取难度比较大，很大程度上找不到最优路径。下面通过具体例子说明。

利用分枝算法求图 1 中从 A 点出发到 N 点的最优路径。将评价函数 f(n) =g(n) +h(n)中的 g(n)取为当前结点到起始结点 A 的最短距离，而 h(n)取为当前结点到目标结点 N 的欧氏距离，在应用分枝算法时除采用上面 Dijkstra 算法所用过的拓扑结构外，还应该再给定所有结点的坐标 Node(x,y)，如各点坐标为 A(0,13)，B(3,16)，C(3,11)……。根据分枝算法的具体实现步骤，可求得从 A 到 N 的最短路径是 A—C—E—H—L—N，其距离是 d(A,N) =16.6。查看表 1 可知，用 Dijkstra 算法搜索的最优路径是 A—B—E—H—L—N，其路径长度 15.9，很明显分枝算法没有找到最优路径，而且通过比较两条路径可以发现，当采用分枝算法搜索路径时，从第二个结点就把最优路径舍弃了。

3.4　分枝算法改进思想及实现

为了克服最优路径可能被轻易舍弃的缺点，本文提出采用多次搜索的方法，用增大计算量为代价来换取尽量多的最优路径备选结果。具体的方法如下：将经典的分枝算法搜索出原始最优路径中的结点依次进行封堵后，再按照经典分枝算法搜索在每一次封堵情况下的最优路径。最后将这些新的最优路径与原始最优路径进行对比以便确定最后的最优路径。现举例说明改进分枝算法的具体应用。利用改进的分枝算法求图 1 中从 A 点出发到 N 点的最优路径。

(1) 按分枝算法寻找路径得到：A—C—E—H—L—N，路径长度 16.6。

(2) 封闭此路径中结点 C 后得到的最优路径为：A—C—E—H—L—N，路径长度 15.9。

(3) 封闭此路径中结点 E 后得到的最优路径为：A—C—F—I—L—N，路径长度 17.1。

(4) 封闭此路径中结点 H 后得到的最优路径为：A—C—E—I—L—N，路径长度 17.2。

(5) 封闭此路径中结点 L 后得到的最优路径为：A—C—E—H—K—N，路径长度 18.7。

对前面求得的 5 种路径长度进行对比，得到最优路径为 A—B—E—H—L—N，其长度为 15.9，从而将此路径定位改进分枝算法求得的最优路径。查看表 1 可知，此路径正是采用 Dijkstra 算法时求得的最优路径。

3.5　实验对比

为了进一步验证改进分枝算法的有效性，利用 VC

(下转第 40 页)

(上接第 35 页)

编制最短路径搜索程序。以 78 个结点的地图作为对象得到实验结果如下:采用经典分枝算法对 77 个结点分别进行路径规划,有 45 个找到了最优路径。而采用改进的分枝算法对 77 个结点进行路径规划时,有 68 个找到了最优路径,有 8 个结点虽未找到最优路径但得到了比经典分枝算法更短的路径,只有一个结点和经典分枝酸法结果一致。这充分说明,改进的分枝算法较经典的分枝算法在搜索最优路径的成功率方面具有明显的优势。

4 结束语

本文对经典 Dijkstra 算法和分枝算法进行了改进,改进后的算法具有以下特点。改进的 Dijkstra 算法能在很大程度上节省计算量,提高路径规划速度;改进的分枝算法虽在一定程度上增大了计算量(但远远小于 Dijkstra 算法的计算量),却大大增大了搜索到最优路径的成功率。

参考文献

1 吴立新、史文中,地理信息系统原理与算法[M],北京:科学出版社,2003.

2 郭仁忠、空间分析[M],武汉:武汉测绘科技大学出版社,1996.

3 王家耀,空间信息系统原理[M],北京:科学出版社,2001.

4 徐立华,求解最短路径问题的一种计算机算法[J],系统工程,1993,33(4):62-67.

5 龚洁辉、白玲、高健美,最短路径算法的改进及其实现[J],解放军测绘学院学报,1998,15(2):23.

6 Lee J. Calculation of the shortest path sbyoptimal decomposition. IEEE Trans Syst Man Cybern,1982(3):410.