

基于栅格数据的最佳路径分析方法研究

秦昆,关泽群,李德仁,周军其
(武汉大学遥感信息工程学院,武汉 430079)

摘要:讨论了基于栅格数据的最佳路径分析方法。该方法利用 Dijkstra 算法的基本思想和‘节点/联系’模型,首先通过 8 邻域像元算出每个像元到源像元的最小权距离,然后计算后向连接值,最后根据累积权距离栅格和后向连接栅格计算出最佳路径。本文结合实例讲述了应用 Arc/Info 的 GRID 模块进行最佳路径分析的方法和步骤,并提出了改进算法的研究思路。

关键词: Dijkstra 算法;栅格数据;最佳路径分析

中图分类号: TP 75 文献标识码: A 文章编号: 1001-070X(2002)02-0038-04

0 引言

最佳路径分析是遥感和地理信息系统空间分析的重要技术之一,其目的就是找出一条从出发点到终点的最佳路径(距离最小、时间最短或者是最喜欢的路线等)。根据数据类型,可划分为基于栅格数据的最佳路径分析和基于矢量数据的最佳路径分析。基于矢量的最佳路径分析,是沿着具体的路线和节点,找出一条最佳的路线。该路线是原有矢量图中多个线段的组合,主要应用于交通路线的选择等;基于栅格的最佳路径分析,是以栅格数据(图像数据、Arc/Info 的 GRID 数据等)为基础,找出一条最佳路径。该方法主要应用于大范围的区域分析,如森林大火的传播路径分析、疾病的传播路径分析及文化的传播分析等。当前人们对基于矢量的最佳路径研究比较多,但对于基于栅格的最佳路径分析研究得很少,本文重点讨论了基于栅格的最佳路径分析问题。

1 算法分析

基于栅格的最佳路径分析,可利用 Dijkstra 算法的基本思想,并结合栅格数据的特点进行分析。Dijkstra 算法如下:

设 $G = (N, A, W_{kj})$ 是一个弧权为正值的有向网

络, N 为节点序列, A 为弧段序列, W_{kj} 为点 k, j 之间的弧段长度; U 为点 1 到点 j 最短路径长度; P 为有向路径,即 $P = \{ (1, i_1), (i_1, i_2), \dots, (i_{k-1}, i_k), (i_k, j) \}$; T 表示将 G 中的点编成如 $2, 3, \dots, n$ 等序号。计算公式

$$\begin{cases} U_1 = 0 \\ U_j = \min_{k < j} \{ U_k + W_{kj} \} \quad j = 2, 3, \dots, n \end{cases} \quad (1)$$

计算步骤

步骤 0(开始),置 $U_1 = 0, U_j = W_{1j}, j = 2, 3, \dots, n, P = \{ \}, T = \{ 2, 3, \dots, n \}$;

步骤 1(指出永久标号),在 T 中寻找一点 k ,使得 $U_k = \min_{j \in T} \{ U_j \}$ 。置 $P = P \cup \{ k \}, T = T - \{ k \}$,若 $T = \emptyset$,则计算终止,否则进行步骤 2;

步骤 2(修改临时标号),对 T 中每一点 j ,置 $U_j = \min \{ U_j, U_k + W_{kj} \}$,然后返回步骤 1。

这个算法经过 $n-1$ 次循环后必然结束。在整个算法过程中,步骤 1 要作 $(n-1)(n-2)/2$ 次比较,步骤 2 要作 $(n-1)(n-2)/2$ 次加法和 $(n-1)(n-2)/2$ 次比较,总的计算量是 $O(n^2)$ 阶。此算法如图 1 所示。图中 $A、B、C、D、\dots、H$ 为节点,各个相邻节点之间的‘联系’用距离值表示。若想找出节点 A 到节点 D 的最短路径(A 为永久性节点),应依次考察与节点 A 相邻的每个节点,用它们各自到节点 A 的距离(联系值)重新标注这些节点,即 $B(2, A)、G(6, A)$ 。选取标注最小的节点 B ,使其成为新的工

项目(863 计划项目,项目编号 2001AA135081)支持。

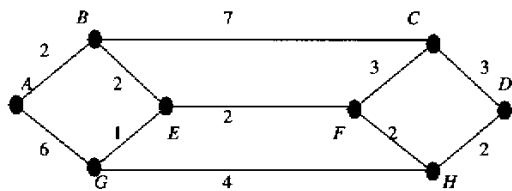


图 1 Dijkstra 算法示意图

作节点。从 B 开始,检查与 B 相连的所有节点,如果 B 的标注与从 B 到所检查的节点距离之和小于所检查节点的标注,就重新标注这个节点,否则标注保持不变。从所检查的节点中找出标注最小的节点 E,使其成为永久性节点。依照上述方法,继续考察其它节点。例如,有两个节点 F 和 G 与 H 相连,用它们的标注分别加上它们到 H 的距离,选取最小的作为 H 的标注,并把 F 标注为永久性节点。依次类推,考察其余节点,直到目的点 D 为止。据此选取的最佳路径如图 2 所示,即 $A \rightarrow B(2, A) \rightarrow E(4, B) \rightarrow F(6, E) \rightarrow H(8, E) \rightarrow D(10, E)$ 。

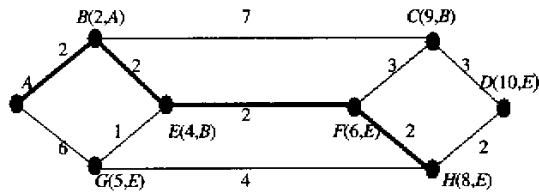


图 2 最佳路径分析结果图

根据 Dijkstra 算法的基本思想,对基于栅格的最佳路径分析算法进行分析。这里的节点是各个相邻的像元,其几何距离均为 1,但是每个像元的阻抗值不同(“联系值”不同),这个阻抗值就是一个像元到另一个像元的权距离。据此,计算从每个单元到源像元的最小权距离。计算权距离需要一个源栅格层和一个代价栅格层。源栅格层可以包括一个或多个区域,可以相连,也可以不相连,且源栅格层中的源像元都有值,非源像元被赋予 NODATA 值(即没有值);每个代价栅格赋予一个阻抗值,用于描述从特定单元移动时的代价。代价栅格的每个单元值表示每单元距离的代价值。代价值可以是旅行时间、花费的金钱或喜欢度等。Arc/Info 的 GRID 模块利用 COSTDISTANCE、COSTPATH 等函数进行基于栅格的最佳路径分析。COSTDISTANCE 函数产生的输出栅格层,记录了到最近源栅格的累计代价。该算法利用了“节点/联系”模型来表达,即每个单元的中心被认为是一个节点,每个节点与相邻节点之间通过“联系”来连结。每个“联系”被赋予一个阻抗值,如果是

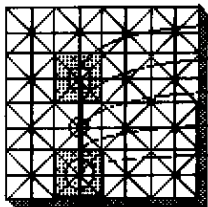
从一个单元到它的 4 个直线方向(上下左右),那么这个“联系”值是该单元的代价值与其相邻单元代价值的平均值,即

$$a_1 = (cost1 + cost2)/2 \tag{2}$$

式中, cost1 是单元 1 的值, cost2 是单元 2 的值, a_1 是从单元 1 到单元 2 的联结长度。累计代价由公式 3 决定,即

$$accum-cost = a1 + (cost2 + cost3)/2 \tag{3}$$

式中, cost2、cost3 分别为单元 2 和单元 3 的值, accum-cost 是从单元 1 移动到单元 3 的累计代价值,如图 3 所示。



始点 (cost1, 单元 1)
 a_1
中间点 (cost2, 单元 2)
 a_2
终点 (cost3, 单元 3)

图 3 “节点/联系”模型的累计代价计算示意图

如果是沿斜线方向,则按公式 4 计算,即

$$a1 = 1.414(cost1 + cost2)/2 \tag{4}$$

用图论的方法产生累加代价距离栅格数据,是确定最低代价栅格层的一种十分有效的方法,它是一个从源像元开始的迭代过程。在第一次计算过程中,因为没有累加代价值,源栅格单元被赋予零值,接着,所有源单元的邻域单元被激活,用公式 2 或公式 4 计算出源像元与邻域像元的“联系”值。因为现在每一个邻域像元都可以到达源像元,所以它们可以被选择并安排到输出累计代价栅格中。从低到高将累加代价值排成一个序列,从以上序列中选择出代价值最小的像元,将其值赋予到输出代价距离栅格中,然后将激活像元序列进行扩展,加入该最小像元的邻域像元。只有那些可能到达源像元的像元才能被添加到激活像元序列中。移动到这些像元的代价值用累计代价公式计算。在激活像元序列中选择出具有最低代价值的像元,其邻域像元被扩展,新的代价值被计算,新的代价像元被加到激活像元序列中。源像元不一定要是相连的,所有不相邻的像元对于激活列表的贡献是平等的,只有具有最低代价值的像元才被选择,其邻域像元才能得以扩展,而不用考虑是从哪个源像元开始计算的。

如果新的像元被选择,并赋予到输出栅格后产生代价值更小的路线后,激活列表中的单元被更新。在更新的过程中可能遇到如下情况,即从激活列表的单元计算后可能有多个值将被赋予同一个输出单

元(类似于前面介绍的 Dijkstra 算法中 *F* 点、*G* 点对 *H* 点的情况),这时,在激活累计代价列表中,具有最低代价的单元值将被安排到输出栅格中。累计代价被计算时,这些代价也计算新的被安排到输出栅格的邻域像元的值,即使邻域像元是从另一个源像元开始计算的。如果新的累计代价值比当前值大,该值被忽略;如果累计值小,那么当前位置的累计代价值将被新的代价值替换,根据那个单元发现了一条更经济、更满意的路径,因此将其加到激活列表中。如果输入栅格有多个源,扩展过程继续安排最经济的代价值到激活列表中,而不用考虑是从哪个源开始的。当增长阵线相遇时,具有最经济代价路径的阵线继续,直到在一个符合条件的单元获取一个代价值为止。

可以想象,当这些扩展模式相遇时,从一个增长模式的像元可能会发现从另一个增长模式更经济,假如这样的话,他们将被重新安排到新的源。当所有单元从激活列表中被选择后,其结果就是累加代价或者是权距离栅格。该处理过程使每个单元获得最低代价值。该过程直到遇到栅格的边缘,或者是视窗的边界,或者是超过最大距离域值时结束。

代价距离栅格确定了从每个像元到源像元的最小代价,它没有显示出是到达哪个源像元以及如何到达的,可以通过“代价后向连接”栅格值的记录连接方向,即用 1 到 8 的值分别表示东、东南、南、西南、西、西北、北、东北 8 个方向(如图 4),重新构造出到源单元的路线,每个像元值记录了哪个方向的邻域像元可以到达源像元。例如,如果值是 1,表示向右是源的方向,如果值是 5,表示向左是源的方向。一旦累计代价栅格和后相联系栅格确定以后,到一个确定目的地的最佳路线就可以获得。

6	7	8
5	0	1
4	3	2

图 4 “联系”
方向示意图

2 实例分析

基于栅格数据的最佳路径分析方法在森林大火的传播、疾病的传播、文化的传播以及野生动物迁移等路径分析中具有很好的应用价值。例如,保护野生动物需要了解野生动物的生活习惯,迁移规律。利用基于栅格数据的最佳路径分析技术可分析野生动物迁移的最佳路径,从而为制定合理的野生动物保护策略提供依据。

例如,在某个地区有一群鹿需要从一个栖息地迁移到另一个栖息地,这些鹿在行动时,有自己的喜

好度,例如喜欢走坡度小的地方,喜欢走南坡,喜欢走不太茂密的森林,而不喜欢走附近有居民的地方。为了计算方便,这里暂且只考虑坡度这一个因素(考虑多个因素时,将这些因素的影响值累加即可)。首先我们对坡度因素进行分级,然后根据鹿的喜好程度赋与不同的权值,该权值就是计算代价值时的阻抗值,因此需要根据 DEM 数据产生坡度图。对这些栅格值重新分级, Arc/Info 提供的 RECLASS 函数和 SLICE 函数可以处理这一问题。RECLASS 函数的产生过程非常简单:

首先建立一个 ASCII 文件,产生原值与新值的对应,例如 1020 7;2030 5;3060 6;……,即 10 到 20 的栅格被赋予 7,20 到 30 的栅格赋予 5,30 到 60 的栅格赋予 6 等等,然后利用 RECLASS 函数重新赋值,如

$$Aspectprefer = reclass (aspect , aspectreclass) \tag{5}$$

其中 *Aspectprefer* 为输出栅格, *aspect* 为原栅格, *aspectreclass* 为 ASCII 文件。

关于 SLICE 函数,可以按相等值分级,例如从 0 到 90 度平均分成 10 级。如

$$Slopeprefer = slice (slope , eqinterval , 10) \tag{6}$$

Slopeprefer 为输出栅格, *slope* 为原栅格, *eqinterval* 表示平均分级。

现在需要创建两个辅助栅格,一个确定源单元,一个确定目的地单元。可以用 SELECT 函数实现,如

$$Source = selectbox (areamap , * , inside) \tag{7}$$

Aim = select (*areamap* , * , inside) (8)
Source、*Aim* 分别为输出源单元和目的地单元, *areamap* 为原有栅格层。

然后,用 COSTDISTANCE 函数计算代价距离栅格,即

$$distance = costdistance (source , costprefer , backlink) \tag{9}$$

以上公式产生了两个栅格,一个是 *distance*,即代价栅格;一个是 *backlink*,即后向联结栅格,用以记录联结方向。

最后,用 COSTPATH 函数找出从起始点到目的地的最佳路线栅格层。即

$$ourpath = costpath (aim , costprefer , backlink) \tag{10}$$

输出栅格 *ourpath* 就是最佳路径,也是鹿从两个居住地移动的最喜欢的路径。该路径的像元位置被

保存起来,如图 5 所示,从图上可直观地看出,从 A 点到 B 点的最佳路径基本上是满足要求的,即走坡度最小,并且距离最近的路线。

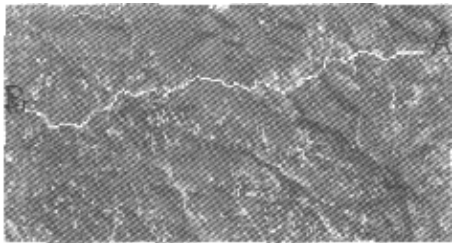


图 5 最佳路径结果图

3 结语与讨论

基于栅格数据的最佳路径分析方法具有很高的适用价值。该算法以“节点/联系”模型为基础,借助 Dijkstra 算法思想,计算从出发点到目的地的最小累计代价,找出一条最佳路径。该算法科学合理,且简单,计算量小,已在大多数基于栅格的 GIS 系统中得以应用,如 Arc/Info 的 GRID 模块和 GRASS 模块等。

但该算法由于只考虑 8 邻域像元方向作为唯一的可能移动方向,存在一定的缺陷,即可能会产生人为的锯齿形路线,因此只是最佳路线的近似,与真实

情况有一定的差距,需要作进一步的改进。有些研究者提出可以与一些非连接的像元进行连接,采用新的代价计算模型,并且应用两层后向连接模型,即分别建立 x 坐标、 y 坐标的后向连接层。采用这些方法可以克服传统的 8 像元或 16 像元连接的缺陷,可以克服锯齿形路线的缺陷,在此不作详细介绍。对于如何克服锯齿状路线这一缺陷,今后还需作进一步研究。

参考文献

[1] XU Jiang - ping, Richard G, Lathror Jr. Improving cost - path in a raster data format[J]. Computer & geoscience, 1994,20(10) :1455 - 1465.

[2] 樊红. Arc/Info 应用与开发技术[M]. 武汉 :武汉测绘科技大学出版社,1999.

[3] 李德仁,关泽群. 空间信息系统的集成与实现[M]. 武汉 :武汉测绘科技大学出版社,2000 年.

[4] 边馥苓,等. 地理信息系统原理与方法[M]. 北京 :测绘出版社,1996.

[5] 刘家壮,王建方. 网络最优化[M]. 武汉 :华中工学院出版社,1987.

[6] 杜瑞甫. 运筹图论[M]. 北京 :北京航空航天大学出版社,1990.

[7] Environmental Systems Research Institute, Inc. [EB/OL]. [http ://www . esri . com](http://www.esri.com).

THE BEST ROUTE ANALYSIS BASED ON RASTER DATA

QIN Kun, GUAN Ze - qun, LI De - ren, ZHOU Jun - qi

(College of Remote Sensing Information Engineering, Wuhan University, Wuhan 430097, China)

Abstract: This paper deals with a method for analyzing the best route on the basis of raster data. Utilizing“ Dijkstra”algorithm and“ node/link”model, the method first calculates the least weight distance from every cell to the resource cell (cells), with the weight distance figured out by the cells of 8 - adjacent directions, then calculates the back - link values, and finally finds the best route according to the accumulative weight distance raster and back - link raster. With an example, this paper describes the method and steps for finding a best route based on the GRID module of Arc/Info software. This paper also points out the advantages and disadvantages of this method, and puts forward the train of thought for improving it.

Key words: Algorithm of dijkstra ; Raster data ; The best route analysis

作者简介:秦昆(1972 -),男,在职博士生。现主要从事遥感和地理信息系统的理论和应用研究。

(责任编辑:刁淑娟)