

交通网络限制搜索区域时间最短路径算法

陆 锋 卢冬梅 崔伟宏

(中国科学院遥感应用研究所, 北京 100101)

摘要 在基于四叉堆优先级队列的改进型 Dijkstra 最短路径算法的基础上, 进一步提出了利用交通网络的空间分布及方位特征构造限制区域的时间最短路径算法。在对城市交通网络空间分布特征进行统计分析的基础上, 针对具体的起、终节点, 设定合理的椭圆限制搜索区域, 以减少算法的搜索规模。针对椭圆限制搜索区域算法由于计算量大而效率不高的弱点, 提出了矩形限制搜索区域算法, 达到既减小算法搜索规模, 又提高算法运行效率的目的。试验结果显示了本文提出的限制搜索区域算法的合理性与有效性。

关键词 最短路径算法 交通网络 限制区域 地理信息系统

0 引言

最短路径问题一直是计算机科学、运筹学、交通工程学、地理信息学等学科的一个研究热点。经典的图论与不断发展完善的计算机数据结构及算法的有效结合使得新的最短路径算法不断涌现, 各具特色^[1-6]。由于这些算法主要诞生于计算机科学及运筹学领域, 在算法的设计过程中只考虑了抽象网络的拓扑特性, 力求通过各种新型的计算机数据结构和运筹学方法, 从理论上减少算法的时间复杂度, 而忽略了具体的网络可能具有的空间分布特性。而这一点正是描述交通网络结构的稀疏图与其它描述拓扑结构的平面图的根本区别所在。如何有效利用交通网络空间分布特性对最短路径算法进行改进, 以减少算法的搜索规模, 提高算法运行效率, 是本文所要探讨的问题。

1 算法实现

1.1 数据结构

算法采用四叉堆优先级队列作为运行数据结构, 采用逆邻接表作为网络拓扑数据结构, 以改进型 Dijkstra 算法作为单源最短路径的实现方法。基于四叉堆优先级队列及逆邻接表的改进型 Dijkstra 算法作者已在文献[7]中详细描述, 在此不再赘述。

值得注意的是, 考虑到道路等级、交通状况等因素, 距离最短路径并不一定所耗时间最短, 在城市街区构成的交通网络中, 由于街道的拥挤程度、路面状况、车道数及其在道路交叉口转弯所耗时间的不同, 更使得距离最短路径与时间最短路径之间存在差异。而时间最短路径对城市交通显然更有意义。本文中实现的最短路径算法为时间最短路径算法。

时间最短路径算法的实现较之距离最短路径复杂。除了考虑弧段的时间阻抗外, 还需考虑道路交叉口的等待时间, 即转弯阻抗。除作者在文献[7]中所描述的数据结构外, 道路的转弯阻抗信息用链表数据结构来实现, 如表 1 所示。

表 1 转弯阻抗数据结构

Node-id	Turn no.	Fnode	Tnode	Impedance
1	tn_1	fn_{11}, fn_{12}, \dots	tn_{11}, tn_{12}, \dots	$imp_{11}, imp_{12}, \dots$
2	tn_2	fn_{21}, fn_{22}, \dots	tn_{21}, tn_{22}, \dots	$imp_{21}, imp_{22}, \dots$
...
s	tn_s	fn_{s1}, fn_{s2}, \dots	tn_{s1}, tn_{s2}, \dots	$imp_{s1}, imp_{s2}, \dots$

考虑转弯阻抗的时间最短路径算法与距离最短路径算法实现方法类似, 区别在于时间最短路径算法需在启发策略的 Bellman 优化条件中加入节点转弯的时间阻抗因子, 以决定节点的累计权值。

1.2 椭圆限制搜索区域算法

产生于计算机科学及运筹学的最短路径算法, 因只考虑网络的拓扑特征或阶段特征, 而忽略了网络的空间分布特征, 使得其搜索过程缺乏方向性, 得

到的是一棵以起点为根的最短路径树。即使构造此路径树的过程在到达终止节点后即结束,使终止节点位于最短路径树的叶节点上,依然有大量的计算是冗余的。这主要归因于算法的全向搜索。

我们知道,给定了一定的起始与终止节点,也就决定了最短路径所对应的大致极限距离 MD 。由此可构造出以起点为中心、 MD 为半径的圆。以起点为原点、由极限距离所决定的地理服务范围内的节点集合是圆内节点集合的真子集。而圆外的节点由于已超出了所估计的极大距离,在算法中可不予考虑。

更进一步,设节点 N 至起点 S 、终点 G 的欧氏距离分别为 $|SN|$ 与 $|NG|$,我们可以把判断条件改为 $|SN| + |NG| > MD$, N 的临界点构成了以 S 、 G 为焦点,以 MD 为长轴的椭圆,如图 1 所示。其直观意义为,即使在 S 与 N 、 N 与 G 之间存在直线路径,由于二者之和已经大于所估计的 S 至 G 的最短路径的极大值 MD ,在算法运行过程中对此节点不予

考虑。

椭圆算法最早见于文献[8],在文献[9]、[10]中又得到应用和发展。算法中椭圆限制的关键是解求合适的椭圆长轴 MD ,以结合起终点坐标构造限制椭圆。我们采用统计分析方法完成这一工作。

首先,在交通网络节点集合中系统抽取具有代表性的一定数目的节点,构造两个节点集合 A 与 B 。则其笛卡尔乘积为

$$C = A \times B = \{(a, b) | (a \in A) \wedge (b \in B)\}$$

C 中的每个元素可看成是待求最短路径的起终点,其欧氏距离为 E_{ab} ,时间最短路径所对应长度为 P_{ab} ,则可设比值系数 $R_{ab} = P_{ab}/E_{ab}$ 。对于所抽取样本,可得到比值系数集合 R ,对 R 中元素进行统计分析,可以得到某一特定值 τ ,使得 R 中总数为满足一定置信水平的元素,其值不大于 τ 。因此,我们就可以用 τ 作为乘常数,利用起、终点坐标求得椭圆长轴 MD 。

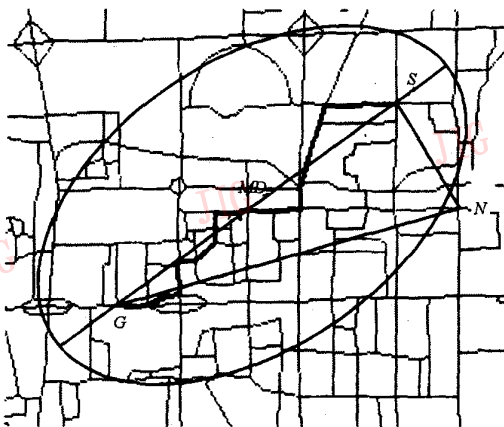


图 1 椭圆限制搜索区域

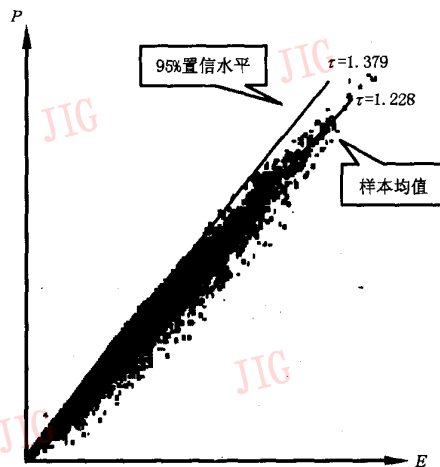


图 2 样本元素比值系数分布

以北京市为例,我们在北京市交通网络中系统地抽取 400 个节点,构造了集合 A 、 B 。则 C 中共包括 40000 个元素。对 C 中的每个元素分别解求 E_{ab} 、 P_{ab} 与 R_{ab} 值,然后对 R_{ab} 组成的集合 R 进行统计,得到在 95% 的置信水平下, τ 约为 1.379。图 2 为 C 中元素所对应的时间最短路径长度 P 与欧氏距离 E 比值分布图。可以看出, P 与 E 之间具有较好的相关性。

椭圆限制搜索算法将搜索节点集限制在一定范围内,大幅度减少了最佳路径算法的搜索规模。然而,此算法需判断每个新扩展出的节点是否落在限定的椭圆内,即使采用基于新扩展节点累积权值已

知的变形椭圆方法,仍需执行大量的乘积与开方计算,占用较多的时间。若最佳路径算法采用早期的深度优先搜索算法,因其在子树的搜索与计算上需耗费大量时间,这时椭圆限制搜索区域算法相对表现出较好的效率。而若采用作者在文献[7]中所提出的基于四叉堆优先级队列及逆邻接表的改进型 Dijkstra 算法,因其采用了更为精致的数据结构,在子树的搜索与计算上耗费的时间已经很少。这时椭圆限制搜索算法已经很难体现出其原先所具备的优越性。当起、终点相距较远时,其效率有时甚至不如无限制区域的全向搜索算法。

1.3 矩形限制搜索区域算法

针对椭圆限制搜索区域算法效率不高的缺点,我们提出矩形限制搜索区域算法。它既继承了椭圆搜索算法对搜索规模合理地进行限制的思想,又避免了算法中大量的乘积与开方计算,具有较椭圆搜索算法更高的效率。其基本思想为求出限制椭圆的最小包含矩形,以此作为限制区域,以减少算法的搜索规模。

在椭圆限制搜索算法中,得到乘常数 τ 后,由起始节点坐标 (x_s, y_s) 、 (x_g, y_g) 可得椭圆方程为:

$$\frac{[\cos\theta(x-a) + \sin\theta(y-b)]^2}{A^2} + \frac{[-\sin\theta(x-a) + \cos\theta(y-b)]^2}{B^2} = 1 \quad (1)$$

其中:

$$\theta = \arctg\left(\frac{y_g - y_s}{x_g - x_s}\right), a = \frac{x_s + x_g}{2}, b = \frac{y_s + y_g}{2} \quad (2)$$

$$A = \frac{\tau}{2} \sqrt{(y_g - y_s)^2 + (x_g - x_s)^2} \quad (3)$$

$$B = \sqrt{A^2 - \frac{(y_g - y_s)^2 + (x_g - x_s)^2}{4}}$$

得到椭圆方程后,分别对 x, y 求偏导数,可得 x, y

```

D[v0]=0; final[v0]=TRUE;
make_heap(A);
for(i=0;i<v0.adj_num;i++){ // 将 v0 邻接节点对应权值插入堆中
    heap_insert(A,D[v0.adj[i]]);
}
ellipse_parameter(); // 由 u0x, u0y, v0x, v0y, τ 计算 A, B, a, b, θ, 构造椭圆方程
rectangle_parameter(); // 计算 xmin, xmax, ymin, ymax, 构造限制矩形
for(i=1;i<n;i++){
    heap_extract_min(A,v); // 从堆中抽出权值最小节点,标识为已处理节点
    if(! (vx>xmin && vx<xmax && vy>ymin && vy<ymax))
        continue; // 判断节点是否落在限制矩形内
    if(v==u0) break; // 到达终点后退出
    final[v]=TRUE;
    for(w=0;w<v.adj_num;w++){
        edge=(NodeInfo[v].first[w]).index_way; // 寻找 vw 对应路段
        if(v==WayInfo[edge].fnode) // 决定路段方向阻抗
            c[v][v-adj[w]]=WayInfo[edge].ft_imp;
        else
            c[v][v-adj[w]]=WayInfo[edge].tf_imp;
    }
    turn_imp=0;
    for(j=0;j<NodeTurn[v].num_turn;j++){ // 决定节点转弯阻抗
        if((NodeTurn[v].fr_node[j]==Father[v]) &&
            (NodeTurn[v].to_node[j]==w)){

```

的极值分别为

$$\begin{aligned} x_m &= a \pm \sqrt{A^2 \cos^2 \theta + B^2 \sin^2 \theta} \\ y_m &= b \pm \sqrt{A^2 \sin^2 \theta + B^2 \cos^2 \theta} \end{aligned} \quad (4)$$

由 x, y 的极值所对应的切线即构成了椭圆的最小包含矩形。如图 3 所示。

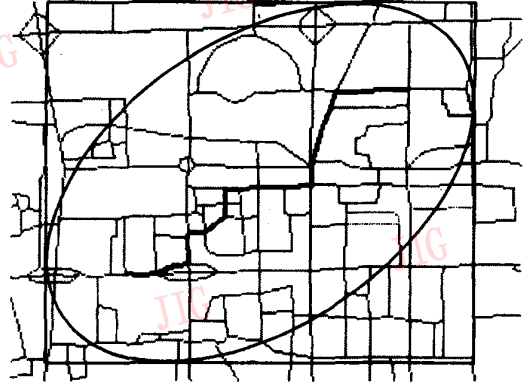


图 3 矩形限制搜索区域

以椭圆的最小包含矩形作为限制搜索区域,对新扩展出的节点,判断其是否落在限制搜索区域内,只需将其坐标与矩形边界进行比较即可,不需要其它复杂运算。基于矩形限制的时间最短路径算法伪码描述如下。

```

turn_imp=(NodeTurn[v].imp[j]);
break;
}
if(! final[v_adj[w]] && (D[v]+c[v][v_adj[w]]+turn_imp<D[v_adj[w]])){
// 所耗时间下降
if(D[v_adj[w]]==INFINITY){ // 未插入堆中的节点
D[v_adj[w]]=D[v]+c[v][v_adj[w]]+turn_imp;
Heap_insert(A,D[v_adj[w]]);
}
else { // 已插入堆中但需松弛节点
D[v_adj[w]]=D[v]+c[v][v_adj[w]]+turn_imp;
Heap_decrease_key(A,pos,D[v_adj[w]]);
}
}
}
}
}
}

```

当起终节点之间方位角 $\theta=k\pi/2$ 时,限制矩形面积取得最小值,为 $4AB$ 。此时限制矩形与限制椭圆面积比值为:

$$\frac{S_{\text{矩}}}{S_{\text{精}}} = \frac{4}{\pi} = 1.2732$$

当 $\theta=(2k+1)\pi/4$ 时,限制矩形面积取得最大值,为 $2(A^2+B^2)$,此时限制矩形与限制椭圆面积比值为:

$$\frac{S_{\text{矩}}}{S_{\text{精}}} = \frac{4\tau^2 - 2}{\pi\tau\sqrt{\tau^2 - 1}}$$

当 $\tau=1.379$ 时,比值为 1.3629。即矩形限制搜索区域算法较之椭圆限制搜索区域算法扩大搜索范围

27.32%—36.29%。虽然搜索范围有所扩大,但从下面的分析可以看出,这种代价仍然是值得的。

2 算法效率分析

为验证椭圆限制搜索区域算法与矩形限制搜索区域算法的效率,我们在北京市交通网络的中心及边缘区域随机抽取 6 组共 60 个节点,各区域分别以 $R_{cc1}, R_{cc2}, R_{ul}, R_{ur}, R_{ll}, R_{lr}$ 表示,计算各组节点之间无限制区域、椭圆限制区域及矩形限制区域下的时间最短路径耗费 CPU 时间,结果如表 2 所示。

表 2 不同限制区域下样本最短路径算法平均运行时间比较 (单位 s/1000 次)

	$R_{cc1}-R_{ul}$	$R_{cc1}-R_{lr}$	$R_{cc1}-R_{cc2}$	$R_{ul}-R_{ur}$	$R_{ul}-R_{ll}$	$R_{ul}-R_{lr}$	$R_{ll}-R_{ur}$
无区域限制	127	104	36	81	75	111	120
椭圆限制	70	41	19	41	59	111	134
矩形限制	68	42	18	38	58	95	115

可以看出,当起点位于网络中心区域,算法搜索范围可向多方向扩展时,椭圆限制算法与矩形限制算法均体现出较好的效率。而当起点位于网络边缘区域,算法搜索范围只能向固定方向扩展时,椭圆限制算法并不一定优于无区域限制算法,当起终节点分别位于网络的角区域时,其效率甚至不如无区域限制算法。而矩形限制算法始终体现出优于无区域限制算法的效率,并在绝大多数情况下优于椭圆

限制算法。此外,由于矩形限制算法较之椭圆限制算法增大了搜索范围,从而进一步提高了最短路径一次搜索成功的置信水平。由此可见矩形限制算法所具有的优越性。

3 结束语

交通网络具有明显的空间分布特性,这是描述

交通网络结构的稀疏图与其它描述拓扑结构的平面图的根本区别所在。本文中,我们提出了利用交通网络的空间分布特征构造限制搜索区域的时间最短路径算法。在对城市交通网络空间分布特征进行统计分析的基础上,针对选定的起、终节点,我们首先利用椭圆算法设定合理的椭圆限制搜索区域,并在此基础上进一步提出矩形限制搜索区域,从而有效地减少了算法的搜索规模,提高了运行效率。试验结果显示了本文提出的限制搜索区域时间最短路径算法的合理性与有效性。

参 考 文 献

- 1 Hung M S, Divoky J J. A computational study of efficient shortest path algorithms. *Computers and Operations Research*, 1988, 15: 567~576.
- 2 Bertsekas D P. A Simple and fast label correcting algorithm for shortest paths. *Networks*, 1993, 23: 703~709.
- 3 Fredman M L, Tarjan R E. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the Association for Computing Machinery*, 1987, 34(3): 596~615.
- 4 Ahuja R K, Mehlhorn K, Orlin J B, Tarjan R E. Faster algo-

卢冬梅 1992年毕业于北京计算机学院软件工程专业,现为中国科学院遥感应用研究所助理研究员,研究方向为全球定位系统应用系统开发。

- 5 Orda A, Rom R. Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length. *Journal of the Association for Computing Machinery*, 1990, 37(2): 213~223.
- 6 Cherkassky B V, Goldberg A V, Radzik T. Shortest paths algorithms: Theory and experimental evaluation. *Mathematical Programming*, 1996, 73: 129~174.
- 7 陆 锋,卢冬梅,崔伟宏,基于四叉堆优先级队列及逆邻接表的改进型 Dijkstra 算法. *中国图象图形学报*(待发表).
- 8 Stig Nordbeck, Bengt Rystedt. *Computer Cartography Shortest Route Programs*. Sweden: The Royal University of Lund, 1969.
- 9 崔伟宏. *空间数据结构研究*, 北京: 中国科学技术出版社, 1995.
- 10 陈行星,崔伟宏. 城市快速反应系统实验研究. *环境遥感*, 1996, 11(3): 227~233.



陆 锋 1991年毕业于武汉测绘科技大学航测与遥感系摄影测量与遥感专业,现于中国科学院遥感应用研究所攻读博士学位。主要从事地理信息系统及全球定位系统的应用研究。

崔伟宏 1962年毕业于莫斯科测绘大学,现为中国科学院遥感应用研究所研究员,博士生导师。研究方向主要包括地理信息系统、空间数据结构、空间决策支持系统、全球定位系统应用及农业信息网络等。

Time Shortest Path Algorithm for Restricted Searching Area in Transportation Networks

Lu Feng, Lu Dongmei and Cui Weihong

(The Institute of Remote Sensing Applications, Chinese Academy of Sciences, Beijing 100101)

Abstract Transportation networks have apparent characteristics of spatial distribution, which is the basic difference between the sparse graphs describing the transportation networks and other planar graphs describing topological structure or hierarchical structure. A time shortest path algorithm based on the restricted searching area is presented in this paper. On the basis of quardary priority queue and inverse adjacent list data structure, a restricted searching area algorithm is established, based on the statistical analysis on the spatial distribution of transportation networks, with the appropriate restricted ellipse and rectangle areas for appointed start and destination nodes. The restricted rectangle searching area is proved more appropriate as the restricted area. Considering the different road travelling impedance and turn impedance, a time shortest path algorithm is presented. The presented algorithms can efficiently decrease the searching scale and improve the running efficiency. It has been proved by some tests on the real-world transportation network.

Keywords Shortest path algorithm, Transportation network, Restricted searching area, GIS