

移动目标自主导航中

限制性节点的结构设计及算法实现

The Node Structure Design of Auto - navigation System Based on Traffic confine rule

陈红英 (广州 华南师范大学计算机系 510631)

毛革非 (广东神盾移动资讯有限公司)

杨宜民 (广东工业大学自动化学院)

摘要:本文针对移动目标自主导航中禁止左转、右转等节点限制性问题,提出了以双指针二级邻接表为存储结构的设计方案,并据此修改了最短路径算法。

关键词:限制性节点 最短路径 分级检索 存储结构 地理信息系统 自主导航 双指针邻接二级表

1 引言

移动目标自主导航系统是智能交通的重要组成部分。它采用了 GPS、GIS、计算机语音、图像、实时通信、软硬件系统等多种技术。系统核心功能是:利用 GPS 获得移动目标的基本状态,如通过 GPS 接收仪的 GPRMC 制式判断 GPS 接收状态是否正常,通过 GPGGA 制式获得经度、纬度、高度,通过 GPRMC 制式获得速度、方向角、时间,通过 GPGSV 制式获得可见卫星、接收卫星的编号和数量等;利用 GIS 数据实现最优路径规划、电子地图显示和地理信息查找;利用 GPS 信息和 GIS 数据进行路网匹配,实现自主导航。

在自主导航过程中,司机经常要查找最优路径,但实际的交通规则中经常具有交通限制(如某一个路口禁止左转、某条道路限时通行等)的节点,采用一般有向图存储结构不

图为例,计算从上下九到香江野生动物园的最短路径,涉及到 15800 个需要计算的节点,计算量非常大,对系统的实时性有很大的影响。对此,我们将一般的存储结构改为以双指针二级邻接表为基础结构,并设计改进了最短路径算法,对长距离地图涉及数据量大的问题我们采用次优最短路径算法的方法,较好地解决了复杂路段、大量节点最短路径计算中存在的超时问题。

2 限制性节点问题提出及解决方案 可行性分析

交通限制:交通管理部门对某些路段的道路通行设置了一些交通规则,以达到减少道路拥堵,疏导交通等目的,常见的有:

- (1) 转向限制:禁止右转、禁止左转;
- (2) 回转限制:禁止掉头;
- (3) 行驶限制:单向行驶、隔离带、双实线;
- (4) 限时行驶:只能在规定的时段内行驶。如图 1,包含四种交通限制:

① 禁止掉头: B、C、E 分别存在不同方向的掉头信息和禁止掉头,节点 B 处, A 到 B 可以掉头,但是从 C 到 B 则禁止掉头。

② 禁止转弯:节点 C 处存在禁止转弯信息,禁止 D 经过 C 左转到 B。



图 1 交通限制下道路实例

能满足它的要求。同时由于最短路径的计算量大,以广州地

③ 禁止直行:禁止 D 经过 C 直行到 G。

④ 只能单行:道路 EH 为节点 E 到 H 的单向行驶线路。

此外,路段 AB、BC、CE 有隔离带,节点 B 处可以掉头是为了减少节点 C 处的交通压力。

针对这些限制,在建立路网拓扑结构时,常规有向图存储结构难于表述。目前描述图的常规方法主要有:邻接矩阵、邻接表、十字链表、邻接多重表。笔者曾经提出并成功使用了二级标记邻接表,经过不断改进,我们提出了双指针二级邻接表,各种方法的优缺点分析如下:

- 采用邻接矩阵具有编程简单的优点,但是它不但空间开销大,而且结构本身不能表示具备交通限制的路网节点结构。如图 2 所示,节点 C 可到节点 G、E、D、B,节点 D 可到节点 C,那么,节点 D 经过节点 C 可通行到节点 B 或 G,与节点 D 到 C 时不能左转、不能直行相违背。

- 采用三维数组等方式能够解决交通限制表示问题,但首先,三维数组需要较大的存储空间,对于嵌入式系统不太适用,其次计算时,时间开销较大,时间复杂度由 $O(n \log n)$ 或 $O(n * n)$ 延长为 $O(n * n * n)$,系统的实时性大大降低,整个系统的性价比难于兼顾。

- 采用邻接表结构计算有向图最短路径有节省空间、易得到顶点的出度、易于编程等优点。但它的描述不适合有交通限制的节点判断。就本实例而言,如果采用邻接表结构,那么节点 C 的邻接点是:G、E、D、B,节点 D 的邻接点是 C,那么,节点 D 经过 C 可通行到 B,违反了禁止左转的交通限制;节点 D 可以经过节点 C 直行到达节点 G,违反了节点 C 处禁止 D 处行来的车辆直行的规定。

- 采用十字链表可用于有向图,但判断节点的传递关系与交通限制矛盾。也会面临邻接表同样问题,因为它们只能反映邻接点,即二个节点之间的连接情况,不能处理通过三个相连节点才能判断的通行法则。如从 D 点出发,经过节点 C,能否到达 G 点。

- 采用二级标记邻接表可判断节点关系,但存在二个不足:一是标记位虽少但也占用了空间。二是二级节点指针的作用发挥不充分。笔者通过不断改进,应用“双指针二级邻接表”的路网拓扑结构进一步优化地解决了这个问题。

- 采用的双指针二级邻接表可判断节点关系且时间复杂度小,空间复杂度小,非常适合表示具备交通限制的路网节点结构。

3 双指针二级邻接表的解决方案

双指针二级邻接表的基本原理是:节点分为头节点、一级节点、二级节点三种类型,一级节点表示与头节点所有出弧直接相邻的节点(由出度组成),二级节点表示与头节点所

	A	B	C	D	E	F	G	H
A	0	1	0	0	0	0	0	0
B	1	0	1	0	0	0	0	0
C	0	1	0	1	1	0	1	0
D	0	0	1	0	0	0	0	0
E	0	0	1	0	0	1	0	1
F	0	0	0	0	1	0	0	0
G	0	0	1	0	0	0	0	0
H	0	0	0	0	0	0	0	0

2 采用邻接矩阵表示节点关系

有入弧直接相邻的节点(由入度组成)。头节点仍以数组表示,与常规邻接表区别在于增加了一个指向父节点的指针:当指针为空时,搜索与该头节点出弧相邻的第一个一级节点,当指针非空时,搜索与该头节点入弧相邻的第一个二级节点。对于二级节点,根据父节点、当前头节点、子节点三点共同判断连通性。具体结构描述如下:

3.1 一级节点

一级节点是与头节点直接相连的邻接点。

adjvex	next	weight
--------	------	--------

```

Typedef struct FirNode{
int adjvex; //邻接点编号
struct FirNode * next; //指向下一个邻接点的指针
double weight; //权值
} FirNode ;
    
```

3.2 二级节点

二级节点是头节点的父节点+头节点的子节点,结构如下:

fathervex	* bronext	* firchild
-----------	-----------	------------

```

Typedef struct SecNode{
int fathervex; //与头节点相邻的父节点编号
struct SecNode * bronext; //指向下一个父节点的指针
struct FirNode * firchild; //指向来自 fathervex,经过当前节点可到达的子节点
} SecNode ;
    
```

3.3 头节点

adjvex	mark	* firadjnode	* firfathermode
--------	------	--------------	-----------------

```
#define vexNum 90000; //定义路网节点数量
typedef struct HeadNode{
int adjvex; //顶点编号
struct FirNode * firadjnode; //指向第一个邻接点的指针
struct SecNode * firfathermode; //指向第一个父节点的指针,
```

```
} HeadNode, TwoAdjList[ vexNum];
```

按此结构我们用双指针二级邻接表

采用如图 3 所示的存储结构,当头节点的 * firfathermode 和 * firadjnode 为二个互斥指针,当指针 * firfathermode 为空时,表明和这个头节点相邻的任何节点都不存在交通限制,搜索时直接通过一级节点,查找头节点的所有邻接点;当指针 * firfathermode 非空时,表明和这个头节点相邻的某个节点存在交通限制,指针指向二级节点,也即本头节点的父节点,每个父节点有二个指针,一个指向本头节点的下一个父节点,另一个指向本头节点的后继节点链表。图 3 中,节点 C 的 * firfathermode 指针非空,则其后继指针指向节点 C 的第一个父节点 G,节点 G 的 firchild 指针指向来自节点 G(父节点),经过当前节点 C 可到达的 C 节点的第一个子节点 E,节点 G 的 Next 指针指向节点 C 的下一个父节点 E。

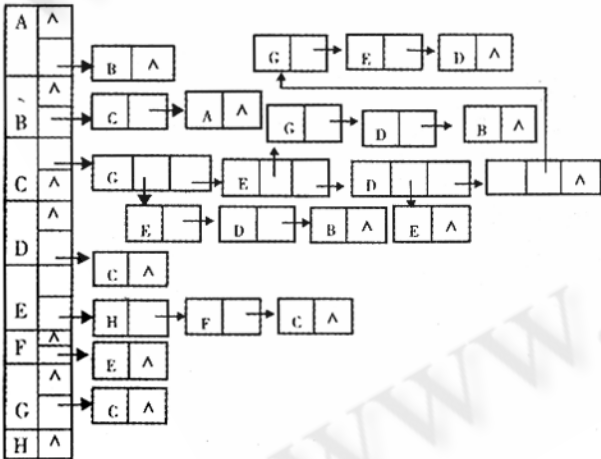


图 3 采用双指针二级邻接表表示交通限制节点结构

可见,双指针二级邻接表可以轻易地表示出从某一个节点出发,经过相邻节点,可以到达哪个节点,如从 D 点出发,经过节点 C 只可以到达节点 E,合理表述了不能经过 C 左转到节点 B 和不能直行到达节点 D 的交通限制。解决了邻接矩阵和常规邻接表不能解决的由三个节点才能判断的限制条件。下一步,我们需要根据存储结构修改最优路径

算法。

4 最优路径算法

Dijkstra 最适合于计算两点间的最短路径问题。我们根据存储结构对常规的 Dijkstra 算法进行改进。

4.1 改进 Dijkstra 算法

通过分步方法求出最短路径。每一步产生一个到达新目的顶点的最短路径。下一步所能达到的目的顶点通过类似贪婪准则选取:在还未产生最短路径的顶点中,选择路径长度最短的目的顶点。也就是说, Dijkstra 的方法按路径长度顺序产生最短路径。根据存储结构的特点,求解从起源点 s 到点 j 的最短路径算法的改进如下:

(1) 初始化。起源点设置为:a) $d_s = 0$, p_s 为空;b) 所有其他点: $d_i = \infty$, $p_i = 0$;c) 标记起源点 s,记 $k = s$,其他所有点设为未标记的。

(2) 计算从点 k 到 j 的直接连接距离 l_{kj} ,当点 k 的父指针为空时,搜索 j 点的邻接点,当点 k 的父指针非空时,通过点 k 的父节点搜索邻接点。

(3) 检验从所有已标记的点 k 到其直接连接的未标记的点 j 的距离,并设置:

$$d_j = \min[d_j, d_k + l_{kj}]$$

(4) 选取下一个点。从所有未标记的结点中,选取 d_i 中最小的一个 i:

$$d_i = \min[d_i, \text{所有未标记的点}]$$

点 i 就被选为最短路径中的一点,并设为已标记的。

(5) 找到点 i 的前一点。从已标记的点中找到直接连接到点 i 的点 j*, 作为前一点,设置:

$$i = j^*$$

(6) 标记点 i。如果所有点已标记,则算法完全推出,否则,记 $k = i$,转到(2)再继续。

4.2 复杂度分析

从 GIS 路网结构分析,通常一个城市具备交通限制的节点数量有限,占整个路网节点比例的 1% 以下,路口岔道 $e < 5$,其中 e 表示岔道口的个数。采用双指针二级邻接表,存储空间上相对于邻接表多支出为 $n \times 4$,n 为路网节点数量,4 代表每个指针占用的空间。对于含有交通限制的节点,因为既要考虑节点的入度、也要考虑节点的出度,使得指向一级节点的指针非空的头节点空间存储支出由 e 变为 $e * e$,其中 e 表示路口岔道数(通常 $e < 5$),二级节点多支出 $e * e * n / 100 < 25 * n / 100 = n / 4$,头节点指针位多支出 $4 * n$,那么,相对于邻接表多支出为 $(4 * n + n / 4) * \max\{\text{length}(\text{FirNode}), \text{length}(\text{SecNode}), \text{length}(\text{HeadNode})\}$,相当于 $O(n)$ 。时间复杂度上,多支出的开销为 $(4n + n/4)$,采用堆结构,计算速度为 $O(4n + n/4 + n \log n)$,即 $O(n \log n)$ 。

由于交通限制节点相对于全部节点而言非常稀疏,占路

网比例的 1% 以下,采用双指针二级邻接表作为节点存储结构,整个系统的时间、空间复杂度和采用邻接表结构在同一个数量级别。

5 结论

以下是该导航系统的运行实例,车辆出行时,电子地图首先显示出车辆的当前位置,并以语音方式提示“您现在在 XX 路上”;当车主输入一个起点和一个终点地名时(起点没有输入则当前点自动视为起点),电子地图上自动给出一条最佳路线(时间、距离、费用),并高亮度显示;随着车辆沿着这个预定路线的行使,根据用户设置的时间、距离、频率等提示方式,自动以图形、语音、文字方式提示“前方 XX 米左(右)转(或直行)进入 XX 路”;当司机偏离预定行使路线时,自动切换新的最优路径,并提示;当车辆行驶超出地图屏幕时,电子地图自动换屏。

本电子地图包含 14866 个节点、12032 条道路,计算一条长度为 62 公里的道路,采用双指针二级邻接表为存储结构、采用改进的最短路径算法,不但成功地解决了交通限制问题,还保证了整个计算时间小于 2 秒,在实际运行中取得了良好的效果。

参考文献

- 1 Goodchild, Michael F., 1999, GIS and Transportation: Status and Challenges. Proceedings of International Workshop on Geographic Information Systems for Transportation (GIS-T) and Intelligent Transportation Systems (ITS).
- 2 Noronha, Val, 1999, Towards ITS Map Database Interoperability? Database Error and Rectification. Proceedings of International Workshop on Geographic Information Systems for Transportation (GIS-T) and Intelligent Transportation Systems (ITS).
- 3 Tomlinson, R. F., 1967. An Introduction to the Geographic Information System of the Canada Land Inventory. Dept. of Forestry and Rural Development Ottawa, Canada.
- 4 卢开澄、卢华明,图论及其应用(第二版),清华大学出版社,1997。
- 5 严蔚敏、吴伟民,数据结构,清华大学出版社,1997。
- 6 米涅卡 E. 网络和最优化算法,李家滢、赵关旗译,中国铁道出版社,1984。