

最小时间路径算法的改进及在路径优化中的应用*

李星毅^{1,2}, 翟晓峰², 施化吉²

(1. 北京交通大学 电子信息学院, 北京 100044; 2. 江苏大学 计算机科学与通信工程学院, 江苏 镇江 212013)

摘 要: 由于城市交通网络中路径行程时间是随着时间的变化而变化的, 求解最小时间路径比较困难, 为此提出把交通网络抽象为时间依赖的网络模型的解决方法。对时间依赖网络模型和理论基础进行分析, 指出文献[1]描述的最小时间路径算法存在的不足, 即不能正确记录路径; 通过引入一个记录路径的数组来对此算法进行改进, 改进后的算法不仅解决了原算法存在的问题, 而且可以满足 $n:1$ 的最短路径搜索, 扩展了原算法的应用范围。最后用实验验证了改进算法的正确性和有效性。

关键词: 时间依赖网络; 最短路径算法; 路径优化

中图分类号: TP301 文献标志码: A 文章编号: 1001-3695(2008)06-1645-03

Improvement of shortest path algorithm and its application to route optimization

LI Xing-yi^{1,2}, ZHAI Xiao-feng², SHI Hua-ji²

(1. School of Electronics & Information Engineering, Beijing Jiaotong University, Beijing 100044, China; 2. School of Computer Science & Telecommunications Engineering, Jiangsu University, Zhenjiang Jiangsu 212013, China)

Abstract: As the time is changing, the travel time is also changing in traffic network. So shortest path search becomes considerably more difficult. For solve the problems, many transportation systems could be represented by networks with travel times that were time-dependent. This paper presented time-dependent networks and its theoretical foundations and analyzed the problems of reference [1]. It proposed an array to note the results to improved the algorithm. The shortest path algorithm in time-dependent networks had a broad application fields. The improved algorithm is proved to be correct and efficient by experiments and practical application.

Key words: time-dependent networks; shortest path algorithm; route optimization

0 引言

最短路径一直是交通规划、计算机科学、GIS 等领域研究的一个热点问题。许多学者和专家就此问题提出了很多优秀的算法。文献[2]对 17 种最短路径算法进行了分析和评价。交通网络中路径优化多采用 Dijkstra、Bellman-Ford-Moore 算法、Floyd 算法、启发式搜索算法 A* 等一些经典算法的改进算法^[3~5]。这些算法都假设交通网络为权值固定的静态网络, 是以路径的空间距离最短作为优化目标的。在实际中, 出行车辆从一个地方出发到达另一个地方主要考虑的是花费时间最少, 而不是空间距离。为此, 本文提出把交通网络抽象为时间依赖的网络模型, 在该模型上求解的最小时间路径即为花费时间最少的路径。

网络中弧段的权值是随着时间的改变而改变的, 弧段的权值与时间存在一定的映射关系, 这样的网络称之为时间依赖的网络。在交通网络中, 根据路段行程时间的实测数据可以预测随后若干时间段内该路段的行程时间, 因此可以建立路段行程时间与时间的一个函数关系, 即路段的行程时间是时间的一个函数。在此基础上, 交通网络就抽象成了一个时间依赖的网络

模型。

本文讨论了基于时间依赖的网络模型及优化条件, 指出文献[1]存在的不能得到正确路径的不足, 给出了改进的最小时间路径算法, 并用一个实例网络来验证改进算法的正确性和有效性。

1 时间依赖网络模型

时间依赖网络记做 (V, E, T) 。其中: $V = \{v_1, v_2, \dots, v_n\}$ 为有限节点集; E 是有限弧集, $E = V \times V$; $T = (g_{ij}(t))$ 是对于每一个弧 $(v_i, v_j) \in E$ 在时间闭区间上定义的时间代价函数。其中: $t \in [t_0, t_m] \subset R$; $g_{ij}(t)$ 是非负实数, 表示从节点 v_i 在 t 时刻出发到达节点 v_j 的行走时间; $[t_0, t_m]$ 是时间闭区间; t_0 是网络节点中的最早出发时间; t_m 是到达的最晚时刻, 即在该时刻以后到达是没有意义的。有时也可以假设 $g_{ij}(t) = \infty, t > t_m$ 。

如果时间是离散的, 时间依赖网络模型记做 $(V \times S, E, T)$ 。其中: S 是将时间离散化后的时间间隔, $S = \{t_0, t_0 + \Delta, t_0 + 2\Delta, \dots, t_0 + (M-1)\Delta\}$; t_0 是所有网络节点中出发最早的时间; Δ 表示一个有效的采样间隔(交通网络中行程时间预测中通常取 5 min, 高峰时间计算中有时也取 15 min); M 是 t_0 到 $t_0 + (M-1)\Delta$ 的时间间隔。

收稿日期: 2007-06-16; 修回日期: 2007-09-24 基金项目: 国家火炬计划资助项目(2004EB33006); 江苏省高校自然科学指导性计划资助项目(05JKD520050)

作者简介: 李星毅(1969-), 男, 副教授, 博士研究生, 主要研究方向为空间数据库、交通信息系统和控制; 翟晓峰(1982-), 男, 硕士研究生, 主要研究方向为算法研究、智能交通系统(zhaixf2001@hotmail.com); 施化吉(1964-), 男, 副教授, 博士研究生, 主要研究方向为信息安全、数据库。

Δ 有效时间段的数目。 $\forall t \in S, g_{ij}(t)$ 是非负实数, $t + g_{ij}(t)$ 总是有定义。对于 $t > t_0 + (M - 1) \Delta$ 时, 定义为 $g_{ij}(t) = \infty$ 。 $V \times S = \{(v_i, t_j) | v_i \in V, t_j \in S\}$ 表示节点的状态空间; 有序对 (v_i, t_j) 表示节点的一个状态, 即节点 v_i 在时刻 t_j 的状态情况。

2 时间依赖网络的理论基础

传统网络中最短路径求解的理论基础与在时间依赖网络中是不同的。本文中假设网络中的弧为非负的, 在传统的网络中, 定理 1 是成立的。

定理 1 网络中最短路径的子路径也是最短路径。
定理 1 是著名的 Dijkstra 和一些标号设置算法的理论基础。时间依赖网络中存在不遵循上面定理的情况, 下面给出一个上面定理不成立的例子, 如图 1 所示。

图 1 的网络中, 节点 a 在到节点 d 的最小时间路径为 (a, b, c, d) , 用时为 20。节点 a 到 c 的最小时间路径为 (a, c) , 用时为 5, 显然该最短路径的子路径 (a, b, c) 不是一条最短路径。可见图 1 的网络不遵循上述定理 1。

时间依赖的网络可分为 FIFO(first in first out) 网络和非 FIFO 网络两种。下面给出定义。

定义 若 $g_{ij}(t)$ 连续, 处处可微并且 $\forall t, g_{ij}(t) < \Delta t + g_{ij}(t + \Delta)$, 则称弧 (v_i, v_j) 为 FIFO 弧。对于时间离散的网络, 若对于任何 $(v_i, v_j) \in E, s + g_{ij}(s) < t + g_{ij}(t), \forall s, s < t$ 且 $t \in S$, 则称弧 (v_i, v_j) 为 FIFO 弧; 不具这种性质的弧称为非 FIFO 弧。如果网络中所有的弧都是 FIFO 弧, 那么该网络就是 FIFO 网络; 包含一条或一条以上的非 FIFO 弧的网络称为非 FIFO 网络。

既适合于 FIFO 网络, 又适合非 FIFO 网络的优化条件如下:

定理 2^[1] $\forall v_i \in V, \forall t \in S$, 设 $f_i(t)$ 表示从节点 v_i 在时间 t 出发到达目的节点 v_N 的时间, 则 $f_i(t)$ 表示从节点 v_i 在时间 t 出发到达目的节点 v_N 的最小时间的充分必要条件是: 对于所有的弧 $(v_i, v_j) \in E, f_i(t) \leq g_{ij}(t) + f_j(t + g_{ij}(t))$ 。

3 时间依赖网络最小时间路径算法

3.1 文献[1] 算法存在的问题

将文献[1] 给出的既适合于 FIFO 网络又适合非 FIFO 网络的 SPTDN 算法应用于实例网络, 发现有些情况下 SPTDN 算法得不到正确的结果, 如图 2 所示。

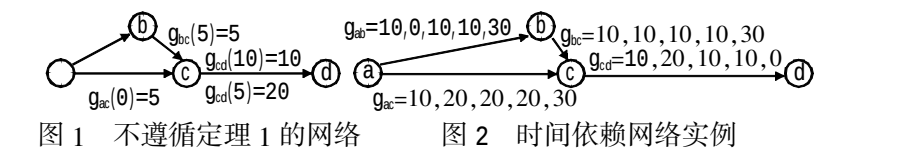


图 2 为一时间依赖网络, 网络中时间间隔 $\Delta = 10, S = \{0, 10, 20, 30, 40\}$ 。 $g_{uv} = 10, 20, 30, 40, 40$ 表示 $g_{uv}(0) = 10, g_{uv}(10) = 20, g_{uv}(20) = 30, g_{uv}(30) = 40, g_{uv}(40) = 40$ 。

对图 2 的网络用文献[1] 给出的 SPTDN 算法进行计算。节点 a 为起点, 0 时刻为出发时刻, 节点 d 为终点, 得到最小行程时间为 $f_a(0) = 30$, 最小时间路径为 (a, b, c, d) 。显然结果中的最小时间路径是不正确的, 正确路径应该为 (a, c, d) 。为

了找出错误的原因, 对 SPTDN 算法的计算过程进行分析。

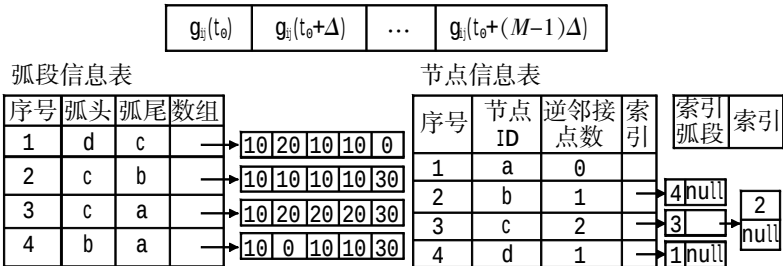
算法中用于记录路径的语句为 $\text{succ}(i) := j$ (节点 i 的后续节点为 j)。设 List 中节点的计算过程为 (d, c, a, b, a) 。算法开始执行, 当计算到节点 c 时, a 的后续节点记录为 c , 即 $\text{succ}(a) := c$; 继续执行, 当计算到 List 中节点 b 时有, $t = 0$ 时 $\text{succ}(a) := c$, 当 $t = 10$ 时 $\text{succ}(a) := b$ 发生错误。这里产生错误的原因是节点 a 在 $t = 0$ 和 $t = 10$ 到达终点 d 的路径不一样。为了解决这一问题, 在记录路径时引入一个时间 t , 用 $i \rightarrow \text{next}[t] = j$ 来记录路径。下面给出改进后的算法实现及实验验证。

3.2 时间依赖网络的存储结构

首先给出时间依赖网络的存储结构。时间依赖的网络比传统网络需要更多的存储空间。对于庞大的网络, 存储结构的设计显得更加重要, 好的存储结构的设计不仅可以降低算法的空间复杂度, 也可以降低算法的时间复杂度。在交通领域内, 采用邻接表数据结构存储网络拓扑数据, 关联节点间查询, 时间复杂度仅为 $O(e/n)$ ^[7]。本文基于逆邻接类型给出一种适用于时间依赖网络的存储结构, 其 C 语言结构定义如下:

```
typedef struct SEdge{ //弧的存储结构
    int sqen; //弧段序号
    int nodeStartid; //弧尾序号
    int nodeEndid; //弧头序号
    int weight[ M ]; //权值
} sedge[ ENUM ];
typedef struct SAdj{ //邻接点结构
    int adjEsqen; //弧段序号
    struct SAdj * psadj; //索引
} sadj;
typedef struct SVNNode{ //节点存储结构
    int sqen; //节点序号
    char nodeid[ 11 ]; //节点 ID
    int adjnum; //邻接点数目
    SAdj * pFirstsdj; //邻接点索引
} svnode[ VNUM ];
typedef struct SResult{ //存放结果
    float fv[ M ]; //最小时间
    int nextNodesqen[ M ]; //后续节点, 用来记录路径
} sresult[ VNUM ];
```

该网络的逆邻接表如图 3 所示。



3.3 最小时间路径改进算法的实现

从源点 v_s 到终点 v_N 的最小时间路径求解过程如下:

a) 初始化。对于所有 $t \in S, f_N(t) = 0$; 其他所有节点 $f_i(t) = \infty$ 。如果 $t + g_{ij}(t) > t_0 + (M - 1) \Delta$ 则 $g_{ij}(t) = \infty$ 。建一个列表 List 将 v_N 插入列表。

b) 取出 List 中的首个节点为当前节点 v_j , 并把该节点从 List 中删除。对于每一个节点 $v_i \in (E^{-1}(j) - \{v_N\})$, 在每一个时间 t , 条件 $f_i(t) > g_{ij}(t) + f_j(t + g_{ij}(t))$ 是否成立, 不成立则

跳到 d)。

c) $f_i(t) = g_{ij}(t) + f_j(t + g_{ij}(t))$; $i \rightarrow \text{next}[t] = j$; 若 v_i 不在 List 中, 则把 v_j 插入 List。

d) 判断 List 是否为空, 若为空算法结束。若不空跳到 b) 继续执行。

以下为算法的 C 语言代码:

```
for( int i=0; i<VNUM; i++) { //初始化结果集
for( int t=0; t<M; t++)
sresult[ i ]. fv[ t] = MAX; //最大值
} for( int t=0; t<M; t++) {
sresult[ VNUM - 1 ]. fv[ t] = 0;
sresult[ VNUM - 1 ]. nextNodesqen[ t] = 4;
}
SList * slist = new SList[ LISTNUM] ;
for( int j=0; j<LISTNUM - 1; j++) { //建立 List
slist[ j ]. pnext = &slist[ j + 1] ;
slist[ j ]. nodesqen = NODESQEN;
}
//算法的主题代码部分
while( first-> nodesqen!= NODESQEN) {
crusqen = first-> nodesqen - 1;
adjnum = svnode[ crusqen ]. adjnum ;
padj = svnode[ crusqen ]. pFirstsdj;
while( adjnum > 0) {
flag = false;
h = padj-> adjEsqen - 1;
g = sedge[ h ]. nodeStartid - 1;
for( int i=0; i<10* M; i+=10) {
k = i/10;
f = k + sedge[ h ]. weight[ k] /10;
if( f <= M - 1) {
if( sedge[ h ]. weight[ k] + sresult[ crusqen ]. fv[ k] < sresult[ g ]. fv[ k]) {
sresult[ g ]. fv[ k] = sedge[ h ]. weight[ k] + sresult[ crusqen ]. fv[ k] ;
sresult[ g ]. nextNodesqen[ k] = crusqen + 1;
plist_one = first->pnext;
while( plist_one-> nodesqen!= NODESQE) {
if( plist_one-> nodesqen == g + 1)
flag = true;
plist_one = plist_one->pnext;
}
if( flag == false)
plist_one-> nodesqen = g + 1;
}
}
adjnum-- ;
padj = padj->psadj;
}
first = first->pnext;
}
```

设网络中节点数为 n , 弧段数为 m , 时间段数目为 M , C 是所有弧中的最大权值。由于一条路径中最多包含 $n - 1$ 条弧段, 每个节点的 $f_i(t)$ 更新次数最大为 nC , 算法至多 nC 次减少每个节点的 $f_i(t)$ (因为每次减少的数量至少为一个单位)。在等价的时间依赖网络中, 节点数为 $n \times M$, 弧段数为 $m \times N$, 因此每个节点最多更新 nMC 次。对网络中所有节点扫描的最大次数为 $\sum_{i \in v} (nC) |E^{-1}(i)|$, 所以算法的最坏时间复杂度为 $O(nmM^2C)$ 。在交通网络中, M 和 C 都是常数, 节点的逆邻接点数目一般也比较小(道路网中一般不超过 4), 因此总的时间复杂度为 $O(n^2)$ 。

4 实验及应用

表 1 列出了对图 2 网络用上述算法进行计算得到的结果。算法结束后得到的最小时间为 $f_a(0)$, 值为 30; 最小用时路径为(a, c, d), 结果正确。

表 1 最小时间路径($f_i(t) / (i \rightarrow \text{next}[t])$), N 表示 null)

节 点	时 间				
	0	10	20	30	40
a	30 /c	20 /b	20 /c	∞ /N	∞ /N
b	30 /c	20 /c	20 /c	10 /c	∞ /N
c	10 /d	20 /d	10 /d	10 /d	0 /d
d	0 /d	0 /d	0 /d	0 /d	0 /d

算法不仅可以得到源点到终点 1 :1 的最小时间路径, 同时可以得到网络中所有节点在不同时刻出发到达终点 n :1 的最小时间路径。改进后的最小时间路径算法有了更加广泛的适用范围。例如, 当城市实时导航系统给出的最小行程时间路径中某路段出现突发事件而被迫改变行进路线时, 只要对当前位置进行相邻次数的计算即可确定新的最小时间路径, 极大地减少了路径搜索的次数。这样可以减少服务器的压力, 这在城市实时导航系统中非常有用。改进后的最小时间路径算法更具实际应用价值。

改进后的最小时间路径算法已经成功地应用于某市交警局中心系统中最短路径的搜索。该系统中以车牌自动识别系统的识别数据为源数据, 通过相邻监测点间的识别数据比对来得到实测路段行程时间; 用指数平滑法预测路段在未来若干个时间段内的行程时间。把行程时间作为时间段的一个函数, 城市交通网络就抽象成了一个时间依赖的网络。

5 结束语

时间依赖网络模型比传统的网络模型有更加广泛的应用背景, 以路段的行程时间为权值的城市道路交通网络是一种典型的时间依赖的网络。最小时间路径算法在智能交通系统(ITS) 的核心部分——路径诱导系统中有很好的应用前景。本文给出的最小时间路径算法的改进算法不仅可以有效地求解 1 :1 的单源点最短路径问题, 而且可以求解 n :1 的多源点最短路径问题, 扩展了最小时间路径算法的应用范围。

参考文献:

[1] 谭国真, 高文. 时间依赖的网络中最小时间路径算法[J]. 计算机学报, 2002, 25(2) : 165- 172.

[2] CHERKASSKY B V, GOLDBERG A V, RADZIK T. Shortest paths algorithms: theory and experimental evaluation[J]. Mathematical Programming, 1996, 73(2) : 129- 174.

[3] 王峰, 游志胜, 曼丽春, 等. Dijkstra 及基于 Dijkstra 的前 N 条最短路径算法在智能交通系统中的应用[J]. 计算机应用研究, 2006, 23(9) : 203- 205.

[4] 谢仕义, 徐兵. 基于 ITS 的加速最短路径搜索算法研究[J]. 计算机工程与应用, 2006, 42(16) : 212- 215.

[5] 吴必军, 李利新, 雷小平, 等. 基于城市道路数据库的最短路径搜索[J]. 西南交通大学学报, 2003, 38(1) : 80- 83.

[6] 陆峰. 最短路径算法: 分类体系与研究进展[J]. 测绘学报, 2001, 30(3) : 269- 275.