

大中城市公交线路查询的数据结构及其算法的实现

The data structure of inquiry for public transport route and the realization of its algorithm in major city

王世祥 饶维亚 (长春大学 理学院 吉林长春 130022)

摘要:给出了公交线路查询的数据结构。可以进行 N 次换乘的线路查询。利用动态 SQL 查询技术,在数据库 SQL Server 中,给出了一个优化实现的公交线路查询的实例,并求出了整个线路网络的换乘次数上确界。

关键词:公交线路查询 数据结构 SQL Server 换乘次数上确界

1 引言

本文从数据集合角度,研究了线路与站点的关系,得到了换乘线路的数据结构。并采用线路与站点的对称式数据表,给出了以换乘次数为标志的公交线路的结构式查询算法,可以进行 N 次的换乘线路查询。

我们的算法在两方面作了优化:一是根据人们乘车经验和习惯所作的优化,就是有直达线路时不坐换乘车,有换乘 1 次的线路不选换乘 2 次的线路,依次类推,即每次查询都到查出换乘次数最少的线路为止,提高了查询的效率;二是针对线路选取所作的优化,即要考虑所有可能的线路,又要选取有代表性的线路,具体方法在介绍有关概念后于 1.3 再叙。

2 数据结构和算法

2.1 主要数据表及基本概念

我们采用的主要数据表有两个: `station_table` (简称 `stable`)、`line_table` (简称 `ltable`)。表 `stable` 的行是站点(站点 ID)集,列是线路(线路 ID)集。表 `ltable` 恰好相反,而且其列值是表 `stable` 列值的转置,故两表称为对称式数据表。简图(表结构在下面给出)如下:

stable	L1 L2 ... L147
S1	
S2	
...	
S909	

ltable	S1 S2 ... S909
L1	
L2	
...	
L147	

在表 `stable` 中,线路 Li 上的各站点相应的列值为 1,对于站点 Si 而言,其行上有若干个 1,表示有相应的若干条线路经过 Si ,这些线路的集合叫 Si 的换线行,经过 Si 的每一条线路叫 Si 的过站线,过站线是一个列集(该线路上所有站点的集合)。

起始站设为 `start`,并设它有 m 条过站线(换线行有 m 个元素);终到站设为 `end`,且有 n 条过站线(换线行有 n 个元素)。

换乘 1 次表示乘车两次(乘两个不同线路的车),依此类推。

2.2 公交线路换乘的数据结构

公交线路换乘的数据结构为:

(1) 换乘 0 次(同一线路)。 `start` 的换线行与 `end` 的换线行的交集,交集非空表示可以换乘 0 次,交集为空表示不能换乘 0 次。

(2) 换乘 1 次。 `start` 的 m 条过站线并集(各过站线上站点集合之并集),与 `end` 的 n 条过站线并集的交集为换乘站集,因为此交集(若非空)的元素(站点)既在 `start` 的某一过站线上又在 `end` 的某一过站线上,交集即为换乘 1 次的换乘站集。

(3) 换乘 2 次。 `start` 的 m 条过站线上各站点(可

能的换乘站)的换线行并集与 end 的 n 条过站线上各站点(可能的换乘站)的换线行并集的交集为换乘线集,因为此交集(若非空)的元素(线路)既通过 start 的某条过站线上某站点又通过 end 的某过站线上某站点,相应的两站点是换乘两次的换乘站。

(4) 换乘 3 次。start 的 m 条过站线上各站点(可能的换乘站)的过站线并集与 end 的相应过站线并集的交集为换乘站集(不妨称为中心换乘站集,因为分别向起始站和终到站方向还各有一个换乘站)。

(5) 换乘 4 次。start 的 m 条过站线上各站点(可能的换乘站)的过站线上各站点(可能的换乘站)的换线行并集与 end 的相应换线行并集的交集是换乘线集(中心换乘线集,因为分别向起始站和终到站方向还各有两个换乘站)。

(6) 换乘 5 次。start 的 m 条过站线上各站点的过站线上各站点的过站线并集与 end 的相应过站线并集的交集是中心换乘站集。

由此可以归纳地得出换乘 N 次的规律。其概要是在如下线路中:

起始站(线) - 换乘站 1 - 换乘线 1 - ... - 换乘站 (线) N/2 - ... - 换乘站 N - 终到站(线)

换乘次数为奇数时要先确定中心换乘站 N/2, 换乘次数为偶数时先确定中心换乘线 N/2, 再由中心换乘站(线)确定整个线路。

2.3 确定公交换乘路线的算法

为描述确定换乘线路的算法,先给出几个数据表结构:

表 1 stable 表

列名	数据类型	长度	允许空
SID	varchar	10	
SNAME	varchar	30	√
L1	smallint	2	√
L2	smallint	2	√
...
L147	smallint	2	√

而表 line01, line02, line21, line22 的结构与表 ltable 的相同, stat11, stat12, stat31, stat32 的结构与表 stable 的相同。使用这些表是为了提高查询效率,因为换乘 N

-1 次所用数据表,可能用于 N 次换乘。

表 2 ltable 表

列名	数据类型	长度	允许空
LID	varchar	10	
LNAME	varchar	30	√
S1	smallint	2	√
S2	smallint	2	√
...
S909	smallint	2	√

换乘站(线)集的表 hctable 结构和换乘线路结果表 lgtable 的结构如下:

表 3 hctable 表

列名	数据类型	长度	允许空
HCID	varchar	10	
HCNAME	varchar	30	√

表 4 lgtable 表

列名	数据类型	长度	允许空
起始站	varchar	30	
乘车线	varchar	10	√
换乘站 1	varchar	30	√
换乘线 1	varchar	10	√
...
换乘站 3	varchar	30	√
换乘线 3	varchar	10	√
终到站	varchar	30	

确定公交换乘路线的算法:

2.3.1 确定换乘 0 次的同一线路

(1) 在表 ltable 中查询 start = 1 的所有线路,送入表 line01;在表 ltable 中查询 end = 1 的所有线路,送入表 line02。

(2) 对表 line01 和表 line02 作连接查询(条件为 line01.LID = line02.LID) 求两表的交集,其结果连同 start、end 站名送入表 lgtable。

(3) 如果表 lgtable 非空,则显示结果且查询结

束;否则进入下面换乘 1 次查询。

2.3.2 确定换乘 1 次的线路

(1) 根据表 line01 中的线路数据 Li 产生查询条件 " $Li_1 = 1 \text{ OR } \dots \text{ OR } Li_p = 1$ ";根据表 line02 中的线路数据 Lj 产生查询条件 " $Lj_1 = 1 \text{ OR } \dots \text{ OR } Lj_r = 1$ "。

(2) 在 stable 中由条件 " $Li_1 = 1 \text{ OR } \dots \text{ OR } Li_p = 1$ " 查询出起始站 start 的所有过站线上的站点,送入表 start11。

在 stable 中由条件 " $Lj_1 = 1 \text{ OR } \dots \text{ OR } Lj_r = 1$ " 查询终到站 end 的所有过站线上的站点,送入表 start12。

(3) 查询 start11 和 start12 的交集 (start11.SID = start12.SID) 将其 SID、SNAME 送入表 hctable,所确定的是换乘站集。

(4) 如果表 jgtable 非空,执行(5),查询算法结束;否则进行下面的换乘 2 次查询。

(5) 对表 hctable 建立一个游标 hcursor,遍历这个游标的记录,根据每一个记录的 SID 确定 line01 和 line02 中的记录(条件是 line01.SID = 1, line02.SID = 1, HCID = SID,实际的查询语句要用动态 SQL),加上起始站名和终到站名送入表 jgtable。

2.3.3 确定换乘 2 次的线路

(1) 根据上一项表 start11 中的站点数据 Si 产生查询条件字符串 " $Si_1 = 1 \text{ OR } \dots \text{ OR } Si_p = 1$ ";根据上一项表 start12 中的站点数据 Sj 产生查询条件字符串 " $Sj_1 = 1 \text{ OR } \dots \text{ OR } Sj_r = 1$ "。

(2) 在 ltable 中,由上面两个查询条件,分别查询出相应的线路集,送入 line21 和 line22,它们分别为 start 和 end 的所有过站线上各站点的换线行并集。

(3) 查询 line21 和 line22 的交集,将其 LID、LNAME 数据送入表 hctable,这是中心换乘线集。

(4) 如果表 hctable 为空,则进入下面换乘 3 次的查询,否则继续。

(5) 遍历游标 hcursor 中的记录,即由每一个中心换乘线 LID,先向起始站方向的表 stat11 查出第 1 个满足条件的站点的 SID 和 SNAME,由此从 line01 查出经过这个站点的线路名 LNAME,当然此线路是经过起始站的,实际上是反方向确定换乘线路。这只是换乘线路的一半,另一半是向终到站方向,由中心换乘线 LID 从表 stat12 查出第 1 个满足条件的站点的 SID、SNAME,由此再从 line02 查出经过这个站点的线路名 LNAME,两个

方向的线路合起来就得到整个换乘线路,也就是对每一个中心换乘线记录得到一条从起始站到终到站的换乘线路。

查询换乘 3、4、5 次线路的方法与上面类似,只要注意到站点表与线路表的交替使用。

每次都是由上两个数据表产生查询条件;得到两个新表;再求两个新表的交集,得到中心换乘站(线);最后由中心换乘站(线)确定整个换乘线路。最后这个确定整个换乘线路的过程涉及的数据表会逐次增多,比如,查询换乘 2 次时要用到 stat11、stat12、line01、line02,那么查询换乘 3 次时就要用到 line21、line22、stat11、stat12、line01、line02,等等。

两站之间所有的换乘线路有很多条,考虑实用性和查询的效率,我们对中心换乘站(线)集的每一条记录确定一条换乘路线,即使这样仍有实例,其换乘线路达 30 条。

3 公交线路查询算法的实现

在 SQL Server 中建立了数据库,包含存储过程,实现了南京市市区公交线路查询算法。

首先作两项准备工作,一是检查各站点的名称是否统一、标准,以使查询准确和高效。例如,南京车站与南京火车站,都是指同一站点,但在不同线路中却有不同的名称,需要修正,还有红山动物园与红山森林动物园,集庆门桥与集庆门大桥等等,这样修正后由原来的 955 个站点缩减到 909 个;二是在 excel 文件中的行,输入线路名和相应的站点名,再利用数据库的数据转换功能和几个小程序,建立上述主要数据表 stable 和 ltable(也可手工建立,但如果需要经常作系统维护会比较麻烦,如增加线路、线路改名等)。

编写存储过程时,一要注意数据表的初始化问题,二要运用动态 SQL 查询技术[6][7]。例如上述查询算法中的 1.3.2 确定换乘 1 次的线路时,要根据表 line01 中的线路数据 Li 产生查询条件 " $Li_1 = 1 \text{ OR } \dots \text{ OR } Li_p = 1$ ",可用如下 T-SQL 语句:

```
SET @SQLJoin = '' //变量@SQLJoin 的初始化,变量已定义
SELECT @SQLJoin = @SQLJoin + LID + ' = 1 OR '
FROM line01
SELECT @SQLJoin = LEFT (@SQLJoin, LEN (@SQL-
```

Join) -3) //去掉最后字符串 'OR '

根据下面的求最大换乘次数的算法及相应的程序(存储过程),求得南京市市区公交线路网的最大换乘

次数为 3,因此安排的存储过程可执行换乘 3 次的查询。如果有需要很容易创建查询换乘 4 次以上的存储过程。下面给出一个查询换乘 3 次的具体结果:

表 5 从天保村到化学工业园的 29 条换乘 3 次的乘车路线

起点站	线路	换乘站 1	换线 1	换站 2	换线 2	换站 3	换线 3	终到站
天保村	85 路	湖西路	48 路	丹凤街	30 路	晓庄	135 路	化学工业园
天保村	92 路	青石村	318 路	丁家桥	74 路	晓庄	135 路	化学工业园
天保村	92 路	莫愁新寓	82 路	白下路	30 路	晓庄	135 路	化学工业园
天保村	85 路	湖西路	78 路	傅佐路	74 路	晓庄	135 路	化学工业园
天保村	85 路	湖西路	68 路	岗子村	24 路	晓庄	135 路	化学工业园
天保村	92 路	青石村	32 路	公交总公司	8 路	晓庄	135 路	化学工业园
天保村	85 路	集庆门	35 路	鼓楼	24 路	晓庄	135 路	化学工业园
天保村	85 路	集庆门	317 路	户部街	30 路	晓庄	135 路	化学工业园
天保村	85 路	湖西路	48 路	鸡鸣寺	24 路	晓庄	135 路	化学工业园

省略其余 20 条线路 (lenovo 机,奔腾四 2.99GHz, 256MB 内存,本次查询用时 0.907 秒)。

注:根据[8]中的街道名称和红色站点名称,表 5 涉及的 85 路、78 路等线路中的湖西街似应为湖西路(参见[8]中 14 路、48 路)。

4 公交线路系统的换乘次数上确界

以南京市公交线路为例。首先求出 147 条线路中某一线路到所有其它 146 条线路的换乘次数,对此线路而言的这 146 个换乘次数的最大值是这个线路的最大换乘次数;其次,所有 147 条线路的最大换乘次数中的最大值,是整个公交线路系统的最大换乘次数,即换乘次数上确界。比如这个数为 3,意味着在系统中任意一条线路最多经 3 次换乘就可到达其它任一线路。

我们的算法只考虑确定某一线路的最大换乘次数。算法的主要思路是“放射性”扩散法,即

线路 → 站点 → 线路 → 站点 … → 线路

也就是先由某一线路上的所有站点(十几、二十几个不等),确定这些站点的所有换线行,再由换线行(过站线)确定其上的所有站点,再从站点确定线路,依此类推。

实现“放射性”扩散法时,有两个主要问题。一是站点、线路之间关系复杂,若计算重复了就无法确定换乘次数。我们从一条线路开始,采用“既定既删”方

法,也就是确定了一条线路(的换乘次数)后或用过一个站点后即刻删除它。一条线路上有一个站点被删除,这个线路下次就会被删除;二是站点集的元素会越来越多,以至无法构成查询字符串,我们解决的方法是,当确定“一层”线路后,需要由这些线路的站点(很多个)确定“下一层”线路时,引入一个游标,对这些线路逐个进行处理(站点最多三十几个),用其上站点确定“下一层”线路的子集,最后再将各子集并起来。

要用到的主要数据表有 *ltables*(其结构和初始记录与表 *ltable* 相同)、*stablels*(与 *stable* 相同)、*sxtables*(列为 *LID*, *LNAME*, *HC*)、*sxstat*(列为 *SXID*)、*sxline*(列为 *SXID*)。

确定某一线路(设为 *L0*)到其它线路换乘上限的算法为:

(1) 在表 *stablels* 中查询 *L0* = 1 的 *SID*,送入表 *sxstat*,在表 *stablels* 中删除 *L0* = 1 的记录,并将 *sxtables* 中 *LID* = *L0* 的 *HC* 值改为 1,初始时 *HC* 值全为空。

(2) 由 *sxstat* 的记录构造字符串“*Si*₁ = 1 OR … OR *Sip* = 1”,并以此为条件在 *ltables* 中查询列 *LID*,送入表 *sxline*,同时在表 *ltables* 中删除所有满足条件“*Si*₁ = 1 OR … OR *Sip* = 1”的记录,在表 *sxtables* 中对所有 *LID* 在 *sxline* 中且 *HC* 为空的记录,将其 *HC* 值改为 2。

(3) 建立一个循环,执行(4)~(13),循环的条件是 *sxtables* 中 *HC* 有空值(直到所有 *HC* 值确定为止)

(4) 建立一个遍历表 `sxline` 的游标 `sxlineCur`, 将每个记录的 `SXID` 赋给变量 `@ varsxID`

(5) `@ varsx = @ varsx + 1`, 变量 `@ varsx` 的初始值为 2, 用来给列 `HC` 赋值的。

(6) 遍历游标的循环(7)至(10)。

(7) 清空表 `sxstat`, 在表 `stablels` 中查询 `@ varsxID = 1` 的列 `SID` 送入表 `sxstat`, 并在表 `stablels` 中删除 `@ varsxID = 1` 的记录。

(8) 如果表 `sxstat` 非空, 执行(9); 否则执行(10)。

(9) 由表 `sxstat` 的记录构造字符串 "`Sii = 1 OR ... OR Sip = 1`" 以此条件在 `ltablels` 中查询列 `LID` 插入表 `sxline1` (这个表的结构与 `sxline` 相同, 是子集, 循环结束后是子集的并集) 而且在 `ltablels` 中删除查出的那些记录。

(10) 将 `sxline` 的下一个 `SXID` 赋给变量 `@ varsxID`。

(11) 关闭并删除游标 `sxlineCur`。

(12) 清空表 `sxline`, 将表 `sxline1` 的记录送入表 `sxline`, 清空表 `sxline1`。

(13) 在表 `sxtablels` 中, 对所有 `LID` 在 `sxline` 中且 `HC` 为空的记录, 将其 `HC` 值改为 `@ varsx`。

注意, 上述 `HC` 的值为乘车次数, 其值减 1 就是换乘次数。

参考文献

- 1 赵巧霞、马志强、张发, 以最小换乘次数和站数为目标的公交出行算法[J], 计算机应用. 2002, 24(12): 136-146.
- 2 马良河、刘信斌、廖大庆, 城市公交线路网络图的最短路与乘车路线问题[J], 数学的实践与认识, 2004, 34(6): 38-44.
- 3 杨新苗、王炜、马文腾, 基于 GIS 的公交乘客出行路径选择模型[J], 东南大学学报(自然科学版), 2000, 30(6): 87-91.
- 4 翁敏、毋河海、杜清运、蔡忠亮, 基于公交网络模型的最优出行路径选择的研究, 武汉大学学报, 2004, 29(6).
- 5 陈箫枫、蔡秀云、唐德强, 最短路径算法分析及其在公交查询的应用, 工程图学学报, 2001, 3.
- 6 Nielsen P. Microsoft SQL Server 2000 宝典[M], 刘瑞、陈徽、闫继忠、刘文等译, 北京: 中国铁道出版社, 2004. 3.
- 7 邹建, 中文版 SQL Server 2000 开发与管理应用实例[M], 北京: 人民邮电出版社, 2005. 8.
- 8 73603 部队编制, 南京交通旅游指南图[M], 湖南地图出版社, 2005. 9.