

一种基于关系数据库的出行路径快速检索算法

刘 晟¹, 汪长勤²

(1. 安徽大学 计算机教学部, 安徽 合肥 230039;

2. 安徽大学 计算机学院, 安徽 合肥 230039)

摘要: 人们在外出选择交通路径过程中, 通常根据起点和终点找出可行的出行方案, 但如果考虑中转(无直达时)条件, 则需要从数据量巨大的关系数据库中检索出可行的方案。给出了一种基于关系数据的首尾协同分层结构快速检索算法, 可以对多次中转信息进行查询和匹配, 从而快速得到可行的出行方案, 满足出行方案选择的实际需要。

关键词: 数据挖掘; 关系数据库; 出行方案

中图分类号: TP301.6

文献标识码: A

文章编号: 1674-7720(2010)19-0007-03

The fast searching algorithm of mixed traffic information based on relational database

LIU Sheng, WANG Chang Qin

(1. Department of Computer Studies, Anhui University, Hefei 230039, China;

2. School of Electronic Science and Technology, Anhui University, Hefei 230039, China)

Abstract: People would usually like to find out the feasible traveling scheme according to the starting point and their destination when travelling. However, if transfer is needed to be taken into consideration (nonstop is unavailable), it'll have to be done through searching relational database (RDB) which consists of enormous amount of the database. This article proposes a fast head-stern collaborative searching algorithm based on the hierarchical structure of relational data, which searches out and matches multi-transfer information and works out practical travelling schemes in a short time to meet the physical demand.

Key words: data mining; relational database; travel scheme

随着交通运输和经济的发展, 越来越多的出行者需要考虑合理地选择出行方案。通常可供出行者选择的出行方案比较多, 如何为出行者快速提供到达目的地的可行性出行方案, 是现今旅游、交通运输等行业迫切需要解决的实际问题, 同时也是学者研究的热点和难点之一^[1-3]。为了解决交通出行问题, 研究人员提出了出行路径选择模型与算法。出行选择模型主要以换乘次数最少与出行距离最短为优化目标, 其目的是寻找一条最优路径^[4]。现有的出行路径选择模型多为基于“出行距离最短”或“出行耗时最少”的最短路模型, 而杨新苗等人的研究结果表明“换乘次数”是大部分乘客在选择出行方案时首先考虑的因素, “出行距离最短”为第二目标^[5]。而且出行选择模型的求解思想是将多目标规划问题转化为单目标规划问题, 或者将多目标问题转化为有主次之分的多

层单目标规划问题^[6]。从理论上讲, 出行者的起点到终点的出行方案可多达数千甚至上万条, 而且可选择的交通工具类型也是多源的。因此出行路径选择模型中所涉及的多源交通数据量较大且关系复杂, 目前多选择利用关系数据库存储出行路径选择模型中所涉及的交通基础数据^[6-8]。故对这些交通数据检索并最终确定最优的出行方案需要大量时间, 而目前大多数的交通出行方案的查询只是针对如飞机、火车或者汽车的这种单一的交通工具的点到点查询。本文以关系数据库 SQL Server 为存储工具, 采用基于分层结构首尾协同的出行路径模型进行快速、准确查询出多种交通工具组合的出行方案。

1 路径选择模型

由于交通出行路径查询中涉及多源的交通数据较多, 导致从出行者的起点到终点的出行方案很多。为了

能够快速准确查询出可行的出行方案,本文采用了基于分层结构首尾协同的出行路径模型来快速准确查询可行的出行路线。该模型的基本思想就是同时从起点(S)和终点(T)查询中转站信息,直到找到匹配的可行方案。这样可以相对快速、准确地查询出多种交通工具组合的出行方案。该模型的出行方案查询策略如图1所示。



图1 基于分层结构首尾协同出行方案查询策略图

该模型主要包括以下几个部分:(1)选用SQL Server存储的交通数据及该模型中所产生的中间数据;(2)同时从起点(S)和终点(T)查询中转站信息,然后再对中转站信息进行匹配和查询,直到找到可行出行方案;(3)比较给出可行出行方案。

该模型的具体描述如下:

(1)利用SQL Server建立包括交通信息表、临时堆栈表、方案主表、方案子表和一些辅助临时表等一系列的关系数据表。

(2)从起点(S)开始向前查询出所有经过起点的交通信息集合。设这些信息集合为 S_1 且层次为1;再以 S_1 为起点向前查询经过 S_1 的所有交通信息集合(不合同种交通工具的重复信息),设这些信息集合为 S_2 且层次为2;则第 i 次搜索形成信息集合为 S_i 且层次为 i ;经过若干次搜索后可将以起点为出发点以终点为目的的整个交通数据搜索完毕。

(3)从终点(T)开始向后查询出所有经过终点的交通信息集合。设这些信息集合为 T_1 且层次为1;再以 T_1 为起点向前查询出经过 T_1 的所有交通信息集合,设这些信息为 T_2 且层次为2,则第 j 次搜索形成信息集合为 T_j 且层次为 j ,经过若干次搜索后可将以终点为出发点以起点为目的的整个交通数据搜索完毕。

(4)比较 S_i 中任意中转站集中任意和 T_j 中相同的中转站,找到从起点到终点的出行可行方案。基于分层结构首尾协同的两次以内中转出行路径查询算法的流程图如图2所示。

由图1可以得出如表1所示的直达、一次和二次转车出行条件。

2 算法实现

基于SQL Server的存储平台,设计了包含交通信息表(表2)、临时堆栈表(表3)、方案主表、方案子表和一些辅助临时表等用来存储路径选择过程中所涉及的数据。其中交通信息表格用来存储交通基础信息,该表中包括车次/航班号、站点序号以及站点编号等字段。临时堆栈表用来将每次查询出的车次信息按层次保存。方案子表是出行方案的明细,同一方案而言,如果是直达的,则只有一条数据;如果是中转的,则数据个数为中转次数加

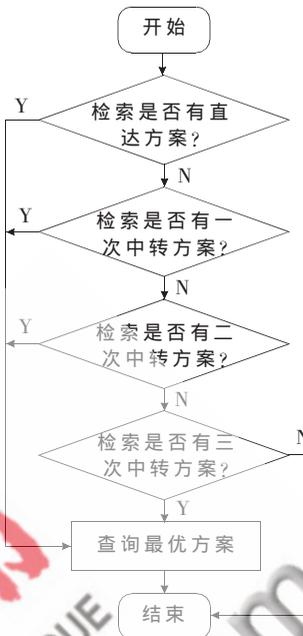


图2 方案查询算法流程图

表1 中转类型及其条件

方案类型	条件
直达	$S_1=[T]$ 或 $T_1=[S]$
一次中转	$S_1=T_1$ 或 $S_2=[T]$ 或 $T_2=[S]$
二次中转	$S_2=T_1$ 或 $S_1=T_2$

表2 交通信息表(T_TRAFFIC_INFO)

列名	列说明
F_VC_TRAIN_NO	车次/航班号
F_NB_SEQ_NO	站点序号
F_NB_STATION_ID	站点编号
F_NB_LOCATION_ID	地点编号
F_VC_TRAFFIC_TYPE	交通类型
.....

表3 临时堆栈表(#T_STACK)

列名	列说明
F_VC_TRAIN_NO	车次/航班号
F_NB_SEQ_NO	站点序号
F_NB_STATION_ID	站点编号
F_NB_LOCATION_ID	地点编号
F_NB_LAY	层次
F_VC_TRAFFIC_TYPE	交通类型
F_VC_OUT_TRAIN_NO	转出车次/航班号
F_VC_OUT_SEQ_NO	转出站点序号
F_NB_OUT_STATION_ID	转出站点编号
F_NB_OUT_LOCATION_ID	转出地点编号
F_NB_IN_SEQ_NO	转入站点序号
F_NB_IN_STATION_ID	转入站点编号

1且数据为方案车次/航班的汇总信息。方案主表是方案子表明细的汇总,具体字段见表4。

表 4 临时方案子表(#T_RESULTZB)

列名	列说明
F_NB_PROJECT_ID	方案编号
F_VC_TRAIN_NO	车次/航班号
F_NB_START_STATION_ID	站点编号
F_NB_END_STATION_ID	地点编号
F_VC_TRAFFIC_TYPE	交通类型
F_BL_START_STATION	是否首发站
.....

3 应用与结果分析

首先,根据图 2 和表 1 确定中转车次数最少的方案,根据方案查询出相应的信息,并将信息保存到临时堆栈表中(为解决同城多个站点中转问题,中转时使用地点编号作为中转条件);然后,生成可行的出行方案并保存到临时结果子表中;最后,对临时结果子表进行汇总并将结果保存到临时结果主表中。

3.1 临时堆栈表数据查询

临时堆栈表的数据是分层的,其 S_1 和 S_2 层是从起点查询的, T_1 和 T_2 层是从终点查询的。其对应的表关联如下:

第一层:

```
T_TRAFFIC_INFO T1, T_TRAFFIC_INFO T2 Where
T2.F_NB_STATION_ID=(S) AND T1.F_VC_TRAIN_NO=
T2.F_VC_TRAIN_NO AND T1.F_NB_SEQ_NO > T2.
F_NB_SEQ_NO
```

第二层:

```
T_TRAFFIC_INFO T1, #T_STACK T2, T_TRAFFIC_IN-
FO T3 WHERE
T2.F_NB_LAY=1 And T2.F_NB_LOCATION_ID=
T3.F_NB_LOCATION_ID And T1.F_VC_TRAIN_NO
!=T2.F_VC_TRAIN_NO And T1.F_VC_TRAIN_NO=
T3.F_VC_TRAIN_NO And T1.F_NB_SEQ_NO > T3.
F_NB_SEQ_NO
```

第三层:

```
T_TRAFFIC_INFO T1, #T_STACK T2, T_TRAFFIC_IN-
FO T3 WHERE T2.F_NB_LAY=4 And T2.F_NB_LOCA-
TION_ID=
```

```
T3.F_NB_LOCATION_ID And T1.F_VC_TRAIN_NO =
T3.F_VC_TRAIN_NO And T1.F_NB_SEQ_NO < T3.F_NB_
SEQ_NO And T1.F_VC_TRAIN_NO != T2.F_VC_TRAIN_NO
```

第四层:

```
T_TRAFFIC_INFO T1, T_TRAFFIC_INFO T2 Where
T2.F_NB_STATION_ID=(T) AND T1.F_VC_TRAIN_NO=
T2.F_VC_TRAIN_NO AND T1.F_NB_SEQ_NO < T2.
```

F_NB_SEQ_NO

3.2 临时结果子表查询

在临时堆栈表数据的基础上,根据表 1 所对应中转类型的条件可直接获得此中转类型的所有方案数;再将具体的方案信息保存到临时结果表中(先保存到子表中,主表信息可根据子表的方案号进行汇总得到)。

3.3 结果分析

将起点、终点及其他参数作为存储过程的入口参数,通过参数便可获得相应出行方案信息,同时可根据最优策略对这些方案进行排序,从而获得出行者所需要的方案。

本文讨论了一种基于分层结构首尾协同的出行路径选择模型,通过对中转信息进行快速检索,并对相应信息判断是否匹配,以便找出相对优化的出行路径。但该算法仅适合起点(S)与终点(T)中都是有若干条线路途经的地点。如果两者中有一点是没有任何线路经过(即为孤点),文中算法对于出现孤点而无法实现中转的情况尚未予以考虑。

参考文献

- [1] 李文勇,王炜,陈学武.公交出行路径蚂蚁算法[J].交通运输工程学报,2004,4(4):102-105.
- [2] 张卫华,陆化普,石琴.公交优先的信号交叉口配时优化方法[J].交通运输工程学报,2004,4(3):49-53.
- [3] DZEROSKI S, LAVRAC N. Relational datamining [M]. Berlin: Sp ringer, 2001.
- [4] 谭满春,李丹丹.基于 Vague 集的公交出行路径选择[J].中国公路学报,2008(5):86-89.
- [5] 杨新苗,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(6):87-91.
- [6] 徐光美,杨炳需,张伟,等.多关系数据挖掘方法研究[J].计算机应用研究,2006(9):8-12.
- [7] AGRAWAL R, SRIKANT R. Mining sequential patterns[C]. Proceedings of the 11 th International Conference on Data Engineering, Los Alamitos: IEEE Computer Society Press, 1995:3214.
- [8] 韩恺,岳丽华,龚育昌.利用关系数据库系统对半结构化数据进行近似查询[J].中国科学与技术大学学报,2005,35(5):674-682.

(收稿日期:2010-07-02)

作者简介:

刘晟,女,1976年生,硕士,主要研究方向:电子商务、企业管理。

汪长勤,男,1980年生,硕士,主要研究方向:信息管理、系统开发。