

# 基于最少换乘的公交最优路径算法的设计与实现

廖楚江<sup>1</sup> 蔡忠亮<sup>2</sup> 杜清运<sup>2</sup> 王长耀<sup>1</sup>

(1 中国科学院遥感应用研究所遥感科学国家重点实验室, 北京市大屯路 3 号, 100101)  
(2 武汉大学资源与环境科学学院, 武汉市珞喻路 129 号, 430079)

**摘 要:** 提出了基于最少换乘的公交最优路径理论, 在此基础上设计了公交最少换乘的算法。由于算法本身的独特性, 笔者将“图算法”部署到空间网络数据库中加以实现, 利用数据库的快速查询、索引支持和在集合运算方面的优秀性能解决了算法的效率问题。同时还利用此类数据库系统对空间查询的支持, 确保算法在求取最少换乘后可以兼顾距离最短的要求。

**关键词:** 空间网络数据库; 最少换乘; 最优路径

**中图法分类号:** P208

城市公交系统是和城市居民日常生活联系最为紧密的环节之一, 甚至在一定程度上决定着城市居民的生活方式, 因而, 时下众多城市的电子地图产品都把实现公交网络最优路径查询作为其重中之重, 以期使电子地图能够更好地满足用户的需求, 但其离最优还有很大的差距。笔者经过调查研究发现, 存在这一差距的主要原因是电子地图软件的开发者与用户双方对公交最优路径的理解有着明显的分歧。一方面, 软件开发者认为, 公交网络最优路径分析同其他网络分析一样, 也应该是以最短为基础的<sup>[1, 2]</sup>; 另一方面, 多数用户认为, 最少换乘才是关键问题。这两者看似统一, 但其实不然, 因为绝大多数城市公交网络中的站点是依据客流量的大小而设计的, 还有一些是源于政治和历史的原因而形成的, 因而即使把最短路径求得再快再好, 建立于其上的最优乘车方案往往会为达到最短而增加换车次数。

当然, 不排除一些最短路径虽然换乘次数较多, 但由于中途等待的时间很短反而能更快地到达的情况, 因而笔者在最后的系统中仍然保留了这一传统模式的实现, 以使用户可以在两种模式之间进行选择。本文主要就基于最少换乘的最优路径展开讨论。

## 1 基于最少换乘的最优路径算法思想

算法的基本思想如图 1 所示, 在查询从站点 1 到站点 2 的最优路径的过程中, 首先看二者之间是否可以直达, 如果是, 则直接进入最后一步, 按照路线距离进行排序, 给出其中最短的几条线路供用户选择; 如果不是, 则查询站点 1 所能直达的所有站点和能直达站点 2 的所有站点, 对这两个集合求取交集, 如果存在交集, 则结束迭代, 进入最后一步按路线总距离进行排序, 否则, 仍然继续迭代, 求取从站点 1 必须经过 1 次换乘才能够到达的站点集合(涉及集合差与并), 与能够直达 2 的所有站点集合求交, 从而得到必须换乘两次才能到达的乘车方案。交集非空, 则结束迭代, 进入最后一步; 否则, 继续迭代, 直到找到乘车方案为止。

显然, 这一基于“图论”的算法包含了众多的集合运算。对该算法的实现有两种途径可供选择, 一是采用主存数据结构, 实现集合的快速运算, 其中包括快速查找、索引支持以及集合运算等一系列算法; 二是直接利用现存商业数据库已有的在集合运算方面的优秀性能。对于前者, 笔者

曾经采用基于 STL( 标准模板库) 的算法予以了实现。在为这个算法作进一步改进的过程中, 笔者尝试了第二种选择, 结果表明, 其在武汉公交网络查询系统中的效率和稳定性明显更进了一步。

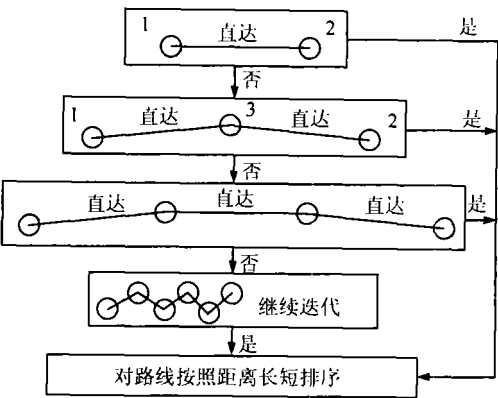


图 1 算法思想  
Fig. 1 Idea of Algorithm

2 空间网络数据库逻辑结构

公交网络图的逻辑结构如图 2 所示, 它主要由三个表组成, 分别为 Route 表、Stop 表和 RouteStop 表。在 Route 表中, 除了传统关系数据库中常见的头两个字段外, 还增加了 Shape 字段, 其中, Linestring 对象类型定义如下<sup>[3]</sup>:

```
CREATE TYPE Linestring AS OBJECT(
    Num_of_Points INT,
```

```
Geometry LineType,
    MEMBER FUNCTION Length get _AppointedPartRouteLength ( long lDepStopPointID, long lReaStopPointID) RETURN NUMBER,
    PRAGMA RESTRICT _ REFERENCES ( Length, WNDSS));
```

由此可以计算指定的两个站点之间的距离。所有线路站点的对应关系在 RouteStop 中进行了实体化, Rank 字段为站点在路线中所处的位置。如可以用如下查询得到从广埠屯能够直达的所有车站:

```
SELECT DISTINCT( S2. Name)
FROM RouteStop R1, RouteStop R2, Stop S1, Stop S2
WHERE R1. RouteID = R2. RouteID AND
R1. StopID = S2. StopID AND
R2. StopID = S2. StopID AND
S1. Name = ' 广埠屯' AND
R1. StopID < > R2. StopID
```

3 算法设计

上文提到, 整个最少换乘算法的思想是一个迭代的过程, 从搜索经过起点站的线路开始, 由线路查找该线路经过的所有站点, 再从这些站点查找经过它们的所有线路, 不断迭代, 直至找到终点站为止。在具体实现中, 可把它演变成一个双向的过程, 即从起点站和终点站同时进行搜索, 直到

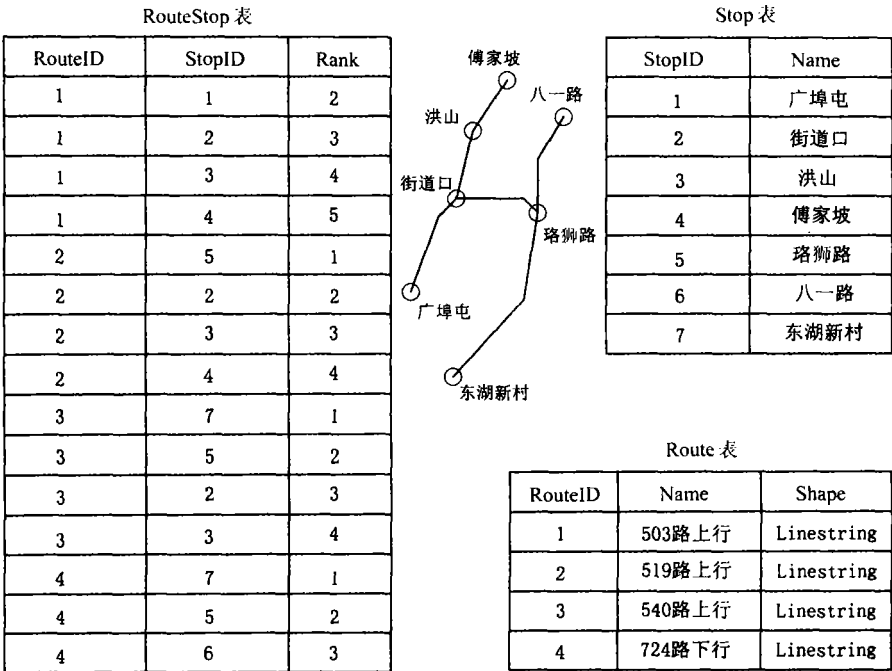


图 2 公交网络图逻辑结构

找到两者相交的站点为止,再从这些线路中寻求最短和最优的线路。下面是算法的伪语言描述。

Public SeachPath()

```
{
    线路集 LineSet1 = ExecuteQuery( 起点站所经过的
所有线路);
    从出发点能够直达的站点集 StopPointSet1 = Union
( LineSet1 中的所有能达站点);
    线路集合 LineSet2 = ExecuteQuery ( 经过终点站的
所有线路);
    能够到达终点站的所有站点集 StopPoint2 = Union
( LineSet2 中的所有能达站点);
    Vector< StopPointSet> StopPointSet Vector;
    StopPoint SetVector. push_back(StopPointSet1);
    while( Intersection ( StopPoint2, StopPointSet1 ) =
= NU LL)
    {
        LineSet3 = ExecuteQuery ( 经过 StopPointSet1 中
每个站点的所有线路);
        该线路集所能到达的站点集 StopPointSet3 =
Union(LineSet3 中的所有能达站点);
        for( int i = 0; i < StopPointSet Vector. size(); i
+ + )
        {
            从出发点必须经 i+ 1 次换乘方能到达的站点集
合 StopPointSet4 = difference( StopPointSet3, StopPoint-
SetVector. at( i) );
        }
        StopPoint Set1 = StopPointSet4;
        StopPoint SetVector. push_back( StopPointSet4 );
    }
    最后一次换乘站点集合 LastStopPointSet = Inter-
section( StopPoint2, StopPointSet1);
    path = GiveBirthOptionalPathPointer ( LastStop-
PointSet);
    Sort( 所有结果) = 最短路径
}
```

在代码中, difference、union 和 intersection 都是关系代数中的基本运算, 可以用 SQL 查询来实现。从整个步骤看来, 算法蕴涵着相当庞大的运算量, 其中对于一次换乘, 其时间复杂度就已经是  $O(n^2)$ , 如果采用常规的主存数据结构, 实现起来将非常困难。笔者曾经通过基于 STL 的数据结构对该算法进行了实现, 但在实际运行过程中, 仍然无法满足需求。本文利用数据库本身在空间查询、索引支持和集合运算上的优秀性能, 使得这一算法最终得到有效实现的同时, 更大程度地满足了需求。以武汉市为例, 它共有 612 个交通站点, 462 条公交线路( 据 2003 年统计), 从中查询

需两次换乘才能到达的站点间的乘车方案, 最多仅耗时 6 s。

## 4 公交网络查询系统应用实例

本算法是隶属于《武汉交通网络查询系统》的一个子任务, 目前该系统为第二版。在原来的版本中, 由于开发周期非常短, 笔者直接在网络版系统中调用了原来为单机版开发的组件, 在组件的实现中采用的是基于 STL 的数据结构和算法, 一定程度上保证了公交换乘查询在网络上的运行效率, 在实际运行中也取得了比较好的效果。但在第二版中, 由于后台支持的数据库由原来的 Access 变成了 Oracle, 从而为空间数据的存储和查询创造了条件。因此, 笔者在这个版本中改用基于数据库结构的算法设计, 剔除了原来系统中的非 Java 部分。实践证明, 新的系统无论在稳定性还是在效率上, 都较前一版本有了很大的改观。

图 3 为该网络查询系统窄带版的界面, 查询从位于武昌高校中心的广埠屯到位于汉口的宗关长途火车站的乘车方案。从图 3 可以看出, 在两站点之间可以乘坐 806 路车直达, 就公交网络的规划设计而言, 理论上, 这两个站点应该是直达的, 但如果采用基于最短路径的查询, 如图 4 所示, 为了尽量走直线, 提供的查询方案显示, 必须换乘一次才能到达。对比两图的线路可以发现, 后者只是在武胜路附近走了一个直线, 距离显示仅少走了 480 m。

为测试算法效率, 笔者选择相距较远的两个站点进行查询, 查询从位于武昌东部的“关山口”到位于汉口北部的“堤角边”的最优路径( 图 5), 结果表明, 该查询耗时仅为 5 s( 算法效率与带宽无关), 只用一次换乘即可实现。同时, 方案的选择也非常多样, 与实际情况中绝大多数人的乘车思路完全吻合, 相对于同类公交查询系统, 其无论是从效率上还是稳定性上, 都要优秀得多。

## 5 结 语

本文提出的基于最少换乘的最优路径思想, 是将“图算法”部署到空间网络数据库中加以实现, 利用数据库在空间查询、索引和集合运算方面的优秀性能来达到目的, 从而减小了系统维护的难度, 缩短了系统的开发周期, 并为今后“图算法”的实现开辟了一个新的途径。当然, 就算法的设计而言, 很多附加因素, 如路面情况、车速、气候条

件等由于数据的实时获取比较困难, 本文没有纳入到考虑范围中来。另外, 就算法的实现而言, 可以通过层次策略来减少 I/O 代价和主存缓冲的需求量, 从而满足更大数据量的要求。



图 3 最优换乘查询 1



图 4 最短路径查询



图 5 最优换乘查询 2

Fig. 3 Query of Best Transfer 1

Fig. 4 Query of Optimum Path

Fig. 5 Query of Efficiency Test 2

参 考 文 献

[ 1 ] Mohammad R K, Cyrus S H. Continuous  $K$  Nearest Neighbour Queries in Spatial Network Databases[ C ]. ST DBM, Toronto, Canada, 2004

[ 2 ] 乐阳. Dijkstra 最短路径算法的一种高效率实现[ J ]. 武汉测绘科技大学学报, 1999, 24(3): 209-212

[ 3 ] Michael Z. Modeling Our World[ M ]. Redlands:

ESRI Press, 2002

[ 4 ] Shekhar S S, Chawla S. 空间数据库[ M ]. 谢昆青译. 北京: 机械工业出版社, 2004

第一作者简介: 廖楚江, 博士生。主要从事交通信息系统、组件 GIS 以及电子地图的网络服务方面的研究。  
E-mail: eren\_sheep@yahoo.com.cn

Modern Realization of Public Traffic Optimal Path  
Based on Least Transfers

LIAO Chujiang<sup>1</sup> CAI Zhongliang<sup>2</sup> DU Qingyun<sup>2</sup> WANG Changyao<sup>1</sup>

(1 State Key Laboratory of Remote Sensing Science, Institute of Remote Sensing Applications,  
Chinese Academy of Sciences, 3 Datun Road, Beijing 100101, China)

(2 School of Resource and Environment Science, Wuhan University, 129 Luoyu Road, Wuhan 430079, China)

**Abstract:** The theory of the optimum query of public transportation based on the least transfers is put forward, and the algorithm to realize it is designed. The graph algorithm is tried to realize based on the spatial network database, making use of the quick query, index support and the collection operation of database to acquire excellent efficiency, and making use of spatial query support of such type of database to ensure that the algorithm can get the shortest path after getting the least transfers.

**Key words:** spatial network database; least transfers; optimum path

About the first author: LIAO Chujiang, Ph D candidate. His research orientation is intelligent transportation system, COM GIS and Web services of electronic map.  
E-mail: eren\_sheep@yahoo.com.cn