

文章编号: 1005-8451 (2018) 12-0023-04

铁路客票快速经由算法的研究与实现

张 霞, 单杏花, 戴琳琳, 尹伊伊

(中国铁道科学研究院集团有限公司 电子计算技术研究所, 北京 100081)

摘 要: 为解决铁路既有运能和路网布局导致的高峰期无票及小站间无直达车的问题, 以计算推荐换乘站为目的, 提出了铁路客票快速经由算法; 通过对比分析既有经由算法, 根据应用程序实现复杂程度, 使用GemFire平台搭建了铁路旅程规划系统模型。经测试, 算法准确高效, 可实现旅客有效中转换乘, 提高铁路客运资源利用率。

关键词: 经由算法; 传递闭包; 邻接表; 邻接矩阵; 弗洛伊德算法

中图分类号: U293.22 : TP39 **文献标识码:** A

Fast path algorithm for railway ticket

ZHANG Xia, SHAN Xinghua, DAI Linlin, YIN Yiyi

(Institute of Computing Technologies, China Academy of Railway Sciences Corporation Limited,
Beijing 100081, China)

Abstract: To solve the problem of difficulties in buying railway tickets in peak time and no direct trains between some small railway stations caused by existing railway transport capacity and road network layout, this paper proposed fast path algorithm for railway ticket to calculate the recommended transfer station; through the comparative analysis of existing path algorithm, and according to the application implementation complexity, using GemFire platform, the railway journey planning system model was built. The test results show that the algorithm is accurate and efficient, and can provide effective passenger transfer solution and improve the utilization ratio of railway passenger resources.

Keywords: fast path algorithm; transitive closure; adjacency list; adjacency matrix; Floyd-Warshall algorithm

近年来, 中国交通运输行业高速发展, 特别是高铁成网后, 高速列车和动车组列车开行密度逐渐增大, 为广大旅客的出行提供了很大的便利。但是, 随着旅客出行数量的增长, 有些问题也日益突出。其中, 站点与站点之间的可达性问题成为困扰旅客出行的主要因素, 具体来说, 旅客在通过网络进行购票时, 经常会遇到始发站与终点站间没有直达车次或直达车次没有余票的情况, 此时, 旅客往往只能自行选择换乘站, 再查询是否有始发站到换乘站以及换乘站到终点站的车次, 这一过程操作复杂, 耗时巨大, 且旅客单靠人力根本无法全面掌握始发站与终点站间的多个换乘线路, 进而往往无法选择合适的换乘站。并且, 在春运等时间段, 还很有可能遭遇所选的换乘站也没有余票的情形, 这导致旅客的出行极为不便。车站发售通票的功能也仅限于路径规划^[1]。

因此, 如何自动为旅客提供有效的换乘方案, 成为目前亟待解决的技术问题。

在此背景下, 本文中的铁路客票经由算法是指, 在给定列车发站 X 和到站 Y 的情况下, 基于中转次数、运行里程、时间、总票价和中转代价等因素, 向旅客提供推荐列车中转线路的算法。

如果将一趟列车从起始站到终点站看作一个有向直线, 沿途各站为该有向直线的各点, 那么经由算法可以转化为有向图的最短路径算法。同时, 所有停靠站都可以很容易地转化为某种唯一编号, 铁路的经由算法就转化为基于 N 个节点的有向图, 计算两个节点之间的最短(最优)路径的问题^[2]。这里, N 为沿途的车站数。

文中所述的基于经由算法的最短路径算法, 是指换乘次数最少条件下的最短(最优)路径, 再辅以考虑里程、时间和票价等综合评分因素, 为旅客提供最合理的出行的接续方案。因此简单的图论最短路算法不能满足经由算法的实际应用需求。

收稿日期: 2018-05-23

基金项目: 中国铁路总公司科技研究开发计划课题(2016X005-B)。

作者简介: 张 霞, 副研究员; 单杏花, 研究员。

1 简单的图论算法

简单的图论算法就是通过有向图广度搜索 (BFS) 或深度搜索 (DFS), 把出发点 i 至到达点 j 的所有路径找出, 然后列出最短路径。在搜索过程中, 需要排除回路, 即经过出发点之后不能再回溯到 i , 且达到终点 j 则停止并返回。

简单的图论算法中, 线路信息即为有向图, 有向图可以通过邻接表 (Adjacency List) 或邻接矩阵 (Adjacency Matrix) 来描述图中各个节点之间的连通属性, 其中, 邻接表使用顶点及其所邻接的边来表示, 而邻接矩阵使用 $N \times N$ 的数组来表示各点间的连通性。图论算法的效率问题直接取决于数据结构的存储结构及方式, 使用邻接表, 时间复杂度为 $O(N+e)$, e 为常量; 而使用邻接矩阵, 时间复杂度为 $O(N^2)$ 。

简单经由算法只能找到路径最短的经由通路^[3], 对于实际客票经由问题, 还需考虑列车票价、中转次数等, 显然不是客票经由算法的最佳选择^[4]。客票经由不是简单的径路问题, 需要考虑车次实际开行情况^[5]。本文基于传统的经典图论算法并结合铁路实际业务, 提出快速经由算法。

2 经由传递闭包

2.1 传递闭包

在数学概念中, 在集合 X 上的二元关系 R 的传递闭包是指包含 R 的 X 上的最小的传递关系。从定义上很容易看出, 对于经由算法来说, 集合 X 就是所有停靠站的集合, 关系 R 就是任意两个停靠站之间是否有列车直达, 即可用邻接矩阵表示, 那么 R 的传递闭包 (简称: 经由传递闭包) 就是任意两个停靠站间是否最终可达。

引入经由传递闭包是为了检验给定的两个停靠站是否最终可达。在快速经由算法中, 根据经由传递闭包中两个停靠站的可达状态, 决定是否继续计算相关中转站, 它是减少无效节点计算的有力方法。

2.2 弗洛伊德算法

计算传递闭包的经典算法是弗洛伊德算法 (Floyd-Warshall algorithm)^[6], 其原理是动态规划: 设路网图为 D , 其中, i 点为发站、 j 点为到站, k 为

路网中的其他站点, $D_{i,j,k}$ 为发站至到站且途径 k 中某个节点的最短路径的代价, i, j, k 为 $1, 2, \dots, n$ 。

(1) 如果最短 (最优) 路径经过点 k , 则

$$D_{i,j,k} = D_{i,k,k-1} + D_{k,j,k-1}。$$

(2) 如果最短 (最优) 路径不经过点 k , 则

$$D_{i,j,k} = D_{i,j,k-1}。$$

$$\text{故 } D_{i,j,k} = \min(D_{i,k,k-1} + D_{k,j,k-1}, D_{i,j,k-1})。$$

在实际运算中, 算法成本需要考虑空间成本及时间成本, 为节约空间, 可以在原基础上通过迭代处理做降维操作, 将空间降至为二维, 如下所示:

```
for k ← 1 to n do
  for i ← 1 to n do
    for j ← 1 to n do
      if (( $D_{j,k} + D_{k,j}$ ) <  $D_{i,j}$ ) then
         $D_{i,j} \leftarrow D_{j,k} + D_{k,j}$ ;
```

$D_{i,j}$ 指发站 i 到达目的地 j 时的总代价, 即车票、耗时、里程等综合代价值, 当 $D_{i,j}$ 为 ∞ , 代表 i, j 两站点之间是不可到达的。

弗洛伊德算法的空间复杂度为 $O(N^2)$, 时间复杂度为 $O(N^3)$ 。这个算法除了可以计算出传递闭包外, 还可以算出每两个顶点间的最短路径。当然这个最短路径中的点到点的代价, 可以是中转次数、里程、运行时间以及票价的加权值。由于该算法的时间复杂度较高, 一个性能优化的做法是提前根据预设权值计算出经由传递闭包矩阵, 那么实际应用中, 需要查询时, 时间消耗仅有 $O(1)$ 。这个算法可以基本满足经由算法的要求, 但是还具有局限性, 比如, 旅客希望自己选择中转次数, 然后设定票价区间或中转时间差, 那么这个算法需要重新计算点到点的代价, 运行时间较长, 而且仅能有一条最短路径的推荐, 不符合旅客购票心理。

3 快速经由算法

快速经由算法需要解决如下旅客需求:

- (1) 接受旅客提出的中转次数的要求;
- (2) 在 (1) 的基础上, 还需要接受旅客对换乘车次的里程, 票价, 中转时间差的要求。

3.1 中转次数

根据旅客提出的中转次数来选择铁路经由路线

时，需要使用邻接矩阵的特性。

设路网图为 G ，具有 N 个乘车站的邻接矩阵 A ，使用二维数组 $N \times N$ 存放各乘车站之间关系数据， $a_{i,j}=1$ 表示 i, j 之间互通，否则 $a_{i,j}=0$ ，表示 i, j 之间不可达。

也就是说, 可以通过 A^k 来判断停靠站 i 到 j 是否存在中转次数为 $k-1$ 的路线, 如果存在, 可以根据以下算法得到所有中转站:

```

function findRouteNode(startNode, endNode, k){
    if(k<=1){
        stack.push(//end of one route
        return;
    }
    if(Ak == 1){
        stack.push(endNode);
    }
    byte [] B = AstartNode,(1,...,n)k-1 & A(1,...,n),endNode
    for( index=0; index <=n; index++){
        if(Bindex == 1){
            findRouteNode(startNode, index,
k-1);

        }

    }
}

```

该算法的时间复杂度为 $O(N^k)$ ，但前提是 A^2, \dots, A^k 都已算好， A^k 一次计算的时间复杂度为 $O(N^3)$ 。

3.2 附加需求

将旅客提出的里程、时间、票价等附加要求加入到查询条件中,需要通过邻接矩阵中附加信息来过滤。在构建邻接矩阵时,除了标注点到点的连通性外,还可以增加类似车次数据。

这些记录着车次信息的矩阵元素, 均可为算法描述时间、费用提供依据。

4 优化及改进设计

快速经由算法可以通过程序具体实现方法来优化性能，改进用户体验。

4.1 数据预处理

数据提前处理生成，并在程序启动时就载入内存。这里的数据包括车次信息，停靠站信息，里程信息，票价信息等^[7]。数据提前处理包括为停靠站编号，将列车相关信息组织成容易访问和获取的结构（其中，可以将全车次哈希成唯一数字 ID，提升比较效率），通过将全车次数据排序预处理操作，提升后续计算的查询操作；根据停靠站信息构建经由邻接矩阵时，将站到站的直达列车信息附加到经由邻接矩阵元素结构中，在这过程中，可以根据预设的条件把一些列车过滤。

4.2 中转次数限定

提供合理的中转次数选择。快速经由算法在理论上支持旅客输入任意中转次数，但每个 A^k 的计算需要 $O(N^k)$ ，比较合适的做法是假设一个中转次数的最高合理数 k ，然后提前生成 A^2, \dots, A^k ，比如认为要求中转 7 次以上基本不可能，那么可以提前生成 A^2, \dots, A^7 。这样，可以很大程度上节省运算时间。

4.3 测试效果

将经由算法根据铁路的业务进行优化后,基于分布式内存数据库集群 GemFire^[8]的硬件环境下搭建了铁路旅程规划原型系统,并进行了实例测试。测试流程为模拟旅客输入基本的查询条件:乘车站、到站,乘车日期,算法根据旅客对时间、票价、换乘概率的要求对接续方案综合评分,高效计算出满足需求的接续方案。

测试样本数据包括既有的铁路路网数据、全国办理客运业务所有车站, 及 1 000 个换乘站样本集, 将火车站及开行车的样本数据初始化为路网图, 如图 1 所示。

[illegible]

图 1 数据初始化

测试评价标准包括：换乘车站是否合理；接续方案是否可靠；综合评价是否易于接受。换乘方案的测试用例如表 1 所示。

表1 换乘方案测试用例

编号	发站-到站	编号	发站-到站
1	三明-马鞍山	2	益阳-平关
3	湛江-义乌	4	宁波东-拉萨
5	海口-章丘	6	石龙-资阳
7	南安-济南	8	海林-南京南
9	玉山-韶关	10	中卫-清远
11	义乌-广州	12	芜湖-聊城
13	松江-哈尔滨	14	渭南-西宁
15	来宾-玉屏	16	海拉尔-长葛
17	阳新-昆明	18	朗乡-毛坝
19	缙云-九台	20	江山-五寨
21	自贡-包头	22	清河县-昭通
23	德州-包头东	24	二龙山屯-宿州
25	溱浦-胶州	26	沁县-阜新
27	凤凰城-祁东	28	姜家-玉门镇
.....			

如图 2 所示，出发城市为北京，到达城市为穆棱，两地之间经过算法计算获得 9 个建议的一次换乘城市的列表（其中，滨江与哈尔滨东、哈尔滨西为同城车站，长春与长春西为同城车站，高花与沈阳、沈阳北、沈阳南为同城车站等），与余票数据结合后，旅客获得联程产品如图 3 所示。

Last input: s1:BJP, s2: MLB, isUseGfCache: Y, isOverrideCachekey: N Input your args or any key use last input. Exit input X.

[1]: 换乘站电报码: BJB 站名: 滨江

[2]: 换乘站电报码: CCT 站名: 长春

[3]: 换乘站电报码: HGD 站名: 高花

[4]: 换乘站电报码: PPT 站名: 四平东

[5]: 换乘站电报码: DZX 站名: 大庆

[6]: 换乘站电报码: AST 站名: 鞍山

[7]: 换乘站电报码: LYT 站名: 辽阳

[8]: 换乘站电报码: DQT 站名: 大石桥

[9]: 换乘站电报码: DFT 站名: 大连北

图2 简单展现一次换乘里程权重优先的方案图

测试过程中，设计了总体测试方案，每轮测试又设计了重点需要解决的方案，选择了具有代表性的城镇，涵盖了省会、大多数大中型城市、部分末梢车站、全部铁路枢纽车站。从测试结果看，测试方案涵盖面广，测试内容较为合理，对修正错误，调整、优化系统内容起到了较好的作用。测试结果表明，算法基于预处理数据方式，计算效率高效，换乘的方案比较合理、实用，初步达到了可以进行公测的条件。



图3 一次换乘结果展示

5 结束语

铁路经由算法根据旅客的个性化需求，计算出接续方案拥有不同的侧重点和灵活性。本文所研究的快速经由算法可以很好地解决根据旅客经由次数需求，以及附加的里程、票价、中转时间差要求的铁路经由问题。通过实际原型系统测试，该算法性能高效，运行平稳，证明了算法的正确性。

参考文献:

[1] 王 华, 季 令. 中转换乘经由在我国铁路客票系统中的应用[J]. 中国铁路, 1999 (3): 39-41.

[2] 王 男, 冯育麒. 车流径路域研究[J]. 铁道学报, 1996, 18 (4): 20-24.

[3] 李引珍, 顾守淮. 铁路网络两顶点间最短路径定向搜索算法[J]. 铁道学报, 1997, 19 (2): 25-27.

[4] 张 彦. 铁路客票中转换乘多径路选择问题的研究[J]. 铁道运输与经济, 1997, 19 (8): 11-13.

[5] 齐玉洁, 陈光伟, 张 彦. 铁路客运经由计算[J]. 铁路计算机应用, 1998, 6 (6): 39-41.

[6] 杨军庆, 安容瑾, 任志国, 等. 基于佛洛依德算法的各院校间最短路径问题的求解[J]. 甘肃科技纵横, 2010, 39 (5): 28-29.

[7] 王明哲, 张振利, 徐 彦, 等. 铁路互联网售票系统的研究与实现[J]. 铁路计算机应用, 2012, 21 (4): 23-25.

[8] 胡必松. 基于列车开行方案的服务网络构建及路径搜索技术研究及系统开发[D]. 北京: 北京交通大学, 2011.

责任编辑 陈 蓉