

铁路最优化路线问题

摘要

近几年来,在全社会客运量稳步上升的同时,长期以来铁路承运了大量旅客。相对于其他的运输方式,铁路具有时间准确性高、运输能力大、运行比较平稳、安全性高等特点。同时火车也成为了旅行的首选交通工具。虽然目前铁路网络已经比较发达,但是仍然有很多地方之间并没有直接到达的铁路。并且在节假日期间,一些热门路线的火车票很难买到。在这种情况下,需要考虑换乘。因此,如何快速、高效地从众多可行路线中选出最优路线成为了解决问题的关键。

针对问题一:要求在直达列车票已售罄情况下丹东→宜昌、天津→拉萨、白城→青岛的最优换乘方案。采用基于铁路优先的层次搜索算法。。首先,考虑是否有直达的公交线路,如果有,则在多条可以直达目的站点的线路中选择最优的路线;如果不能到达,则考虑换乘 1 次的路线,如果还不能,则考虑换乘 2 次的路线、换乘 3 次的路线等。重复上述过程,直到换乘的次数大于换乘次数的最大值或线路的时间花费已经超过了乘车时间的最大值。可以分别求出在直达、转乘 1 次、转乘 2 次情况下的最佳路线。

出发站→终点站	最佳路线	转乘次数 (次)	硬座费 用(元)	硬卧费用 (元)
丹东(K190/K187)→镇江→宜昌东(D3006/D3007)	2260	1	572	684
天津(T238/T235)→武昌→拉萨(T264/T265)	3594	1	431	829
白城(K7392/K7393)→柳河→青岛(K972/K969)	2896	1	246	397

针对问题二:问题二要求从宜昌出发乘火车到上海、南京、杭州、苏州、无锡旅游最后回到宜昌的整个行程的最佳旅游路线。在问题二中,想要求出最优的旅游线路,首先要确定旅游城市的顺序。因此,要确定一个既省时、省钱而且还可以充分让人享受到游玩乐趣的旅游方案。所以,将问题模拟成旅行商问题。运用 0-1 模型来进行求解,将问题转化为一个整数规划问题。

关键词: 转乘次数 层次搜索算法 HAMILTON 回路 旅行商问题 0-1 模型

1. 问题重述

1.1 问题背景

铁路既是社会经济发展的重要载体之一，同时又为社会经济发展创造了前提条件。近几年来，在全社会客运量稳步上升的同时，长期以来铁路承运了大量旅客。相对于其他的运输方式铁路具有时间准确性高、运输能力大、运行比较平稳、安全性高等特点。同时火车也成为了旅途的首选交通运输工具。虽然目前铁路网络已经比较发达，但是仍然有很多地方之间并没有直接到达的铁路。并且在节假日期间，一些热门路线的火车票总是一票难求。在这种情况下，需要考虑换乘，即先从乘车站到换乘站，再从换乘站到目的站。

1.2 问题的相关数据

在附录 1 中给出了 2013 年全国列车时刻表数据，其中包含以下信息：列车车次、列车类型（普快，空调快速，动车等）、站序、车站、日期（当天，第 2 天，第 3 天）、到达时间、离开时间、里程、硬座/一等座票价、硬卧/二等座票价、软座/二等座票价、软座/特等座票价、软卧票价。

1.3 待解决的问题

根据以上信息，建立模型并解决以下问题：

问题一：给出任意两个站点之间的最佳铁路路线问题的一般数学模型和算法。若两个站点之间有直达列车，需要考虑直达列车票已经售罄的情况下最优的换乘方案。根据附录数据，利用你们的模型和算法求出一下起点到终点的最优路线：丹东→宜昌、天津→拉萨、白城→青岛。

问题二：假设你打算从宜昌出发乘火车到上海、南京、杭州、苏州、无锡旅游最后回到宜昌，请建立相关数学模型，给出整个行程的最优路线。

2. 模型假设

假设一：乘车者的转乘列车的次数不会超过两次；

假设二：在转车时，两次乘车所做的列车类型一样；

假设三：在转乘的时候，考虑到火车站的地方可能不同，因此将中间转乘的时间设定为 30 分钟。

3. 符号说明

符号	符号说明
L_i	第 i 条列车线路标号
$S_{i,g}$	第 i 条公汽路线的第 g 个列车站点标号
T_k	列车的行程时间
v_i	第 i 个城市
x_{ij}	表示从城市 i 旅行到城市 j
d_{ij}	城市 i 旅行到城市 j 的总费用
$ s $	城市所组成集合 s 中元素的个数

4. 问题分析

本文主要研究的是在没有直达的火车票或直达的火车票已经卖完的情况下,如何从众多可行路线中选出耗时最短、转乘次数最少、费用最低的路线成为解决问题的关键。附录一给出了全国列车路线的列车车次、列车类型(普快,空调快速,动车等)、站序、车站、日期(当天,第2天,第3天)、到达时间、离开时间、里程、硬座/一等座票价、硬卧/二等座票价、软座/特等座票价、软卧票价。通过已知条件,来求题目中的两个问题。

问题一的分析

问题一要求在直达火车票已售罄情况下求丹东→宜昌、天津→拉萨、白城→青岛的最优换乘方案。其中,最优的换乘方案应该从以下几个因素来考虑:耗时最短、转乘次数最少、费用最低。因此,采用基于铁路优先的层次搜索算法^[2]。首先,考虑是否有直达的公交路线,如果有,则在多条可以直达目的站点的线路中选择最优的路线;如果不能到达,则考虑换乘1次的路线,如果还不能,则考虑换乘2次的路线、换乘3次的路线等。重复上述过程,直到换乘的次数大于换乘次数的最大值或线路的时间花费已经超过了乘车时间的最大值。可以分别求出在直达、转乘1次、转乘2次的情况下的最佳路线。

问题二的分析

问题二要求从宜昌出发乘火车到上海、南京、杭州、苏州、无锡旅游最后回到宜昌的整个行程的最佳旅游路线。在问题二中,要求求出最优的旅游路线,首先要确定旅游城市的顺序。对于问题二,问题的目的是使旅行的费用最少,要做的决策是如何设定旅行路线,决策所受的约束条件是:每个城市都必须去,但仅能去一次。因此,要确定一个既省时、省钱而且还可以充分让人享受到游玩乐趣的旅游方案。所以,将问题模拟成旅行商问题^[1]。运用0-1模型来进行求解,将问题转化为一个整数规划问题。

5. 建模前的准备

为了后面模型与程序设计的方便,在建立此模型前,有必要做一些准备工作。

5.1 数据的存储

由于所给的数据格式不是很规范,需要将其处理成我们需要的数据存储格式。从所给文件中读出线路上的站点信息,存入txt文档中。根据题目要求,将城市所有的列车路线和列车站点抽象为连接图。目前关于图的存储有两种传统的方法:一是邻接矩阵,另一种是邻接表。由于城市列车网络包含的信息多而复杂,因此采用传统的邻接矩阵和邻接表的方法进行存储会丢失大量的信息,不利于最优乘车线路的搜索。因此,采用一种双邻接表的办法来存储上述的城市的列车网络。

为了解决上述问题,这里定义两个关于此问题的邻接表 A_{L1} (线路邻接表)、 A_{L2} (站点邻接表)。为了便于链表结构的设计,现将三种线路运行方式统一调整为上行和下行运行方式,原路返回型和环型列车线路变换为上行时,按原始路线站点顺序链接,变换为下行时,按原始线路站点逆顺序链接。

邻接表 A_{L1} 表示对每一条线路建立一个单链表,其中,表头结点由以下六个域组成:DATA域代表列车线路编号;INFO域表示此列车线路的计价方式。当

INFO=2 时，表示分别该列车路线分段计价。当 INFO=1 时，表示该列车线路采用单一票价制；NUM 域记录此线路通过的结点数；SPA 域表示预留空间，便于查表时搜索；UP_NEXT 域表示链接该线路的上行；DOWN_NEXT 域表示链接该线路的下行。

邻接表 A_{L2} 表示对每个结点建立一个单链表，表中第 i 个结点表示某条通过该站的列车线路，其中表头结点有以下三个域组成：DATA 域表示标准站名或编号；NUM 域记录通过此站的线路数；NEXTDATA 域引入指针指向由通过该站的线路组成的链表。表 A_{L2} 由多条上述形式的单链表顺序连接而成，线路结点含有一个标志域 UDF (UP_down_flag)，当 UDF=1 时，表示该线路上行。当 UDF=0 时，表示该线路为下行。

5.2 搜寻经过每个站点的列车线路

处理上述的信息，得到通过每个站点的所有列车线路，并将它们存入数据文件中。

有了上述两个邻接表存储信息，可以通过搜索邻接表来进行线路选择，求出换乘次数不大于 λ 的可达路径集 Path。

$$Path = \{p_1, p_2, p_3, \dots, p_n\}$$

式中 p_n 表示换乘次数不大于 λ 的最短路径。对邻接表进行搜索，这里主要使用层次搜索法。假设要求出 A 到 I 的线路选择，可以查找邻接表 A_{L2} ，找到出发点 A 的单链表，查找此单链表的第一个结点，在 A_{L1} 表中搜索到与此结点名称相同的线路单链表，然后对此单链表进行搜索，确定对应的列车路线是否通过终点站 I，若通过，则选择此线路；若通不过，则依次查找此列车路线所通过的各个结点。当换乘次数超过 λ （很明显，换乘次数太多，这条路径并非理想的最佳路径），则舍弃该结点，转而查找 A 单链表的第二个结点……依次类推，即可找到满足条件的路径集 Path(A→I)。

6.问题一的解答

该问题一要求丹东→宜昌、天津→拉萨、白城→青岛的最优换乘方案。其中，最优的换乘方案应该从以下几个因素来考虑：耗时最短、转乘次数最少、费用最低。因此，采用基于铁路优先的层次搜索算法。

6.1 列车路线的数学表示

任意两个站点间的路线有多种情况，根据假设二，如果最多允许换乘两次，则换乘路线分别对应图 1 的 3 种情况。该图中的 A、B 为出发站和终点站，C、D、E、F 为转乘站点。

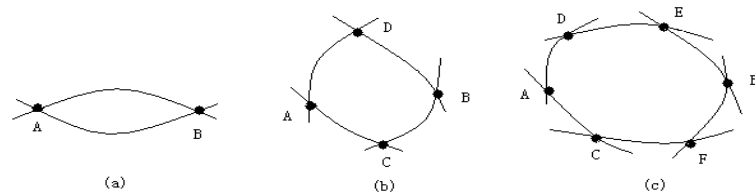


图 1 列车的路线图

对于任意两个列车站点 $S_{i,g}$ 与 $S_{i,g}'$ ，经过 $S_{i,g}$ 的列车线路表示为 L_i ，有 $S_{i,g} \in L_i$ ；

经过 $S_{i,g}'$ 的列车线路表示为 L_i' ，有 $S_{i,g}' \in L_i'$ ；

则第一种情况：直达的路线 L_{s0} （如图 1（a）所示）表示为：

$$L_{s0} = \{ L_{i1} \mid S_{i,g} \in L_{i1}, S_{i,g} \in L_{i1} \}$$

第二种情况：转乘一次的路线 L_{s1} （如图 1（b）所示）表示为：

$$L_{s1} = \{ (L_{i1}, L_{i2}) \mid S_{i,g} \in L_{i1}, S_{i,g}' \in L_{i2}; L_{i1}, L_{i2} \notin LS_0; SC \in L_{i1} \text{ 且 } SC \in L_{i2} \}$$

其中：SC 为 L_{i1} ， L_{i2} 的一个交点；

第三种情况：转乘两次的路线 L_{s2} （如图 1（c）所示）表示为：

$$L_{s2} = \{ (L_{i1}, L_{i2}, L_{i3}) \mid S_{i,g} \in L_{i1}, S_{i,g}' \in L_{i2}, (L_{i1}, L_{i3}), (L_{i3}, L_{i2}) \notin LS_1; (L_{i1}, L_{i2}) \in LS_1 \}$$

通过以上转乘路线的建模过程，可以看出不同转乘次数间可作成迭代关系，进而对更多转乘次数的路线进行求寻。不过考虑到实际情况，转乘次数以不超过 2 次为佳，所以本文未对转乘三次及三次以上的情形做讨论。

6.2 问题一模型的建立

找出了任意两个列车站点间的可行路线，就可以对这些路线按不同需求进行选择，找出最优路线了：

因此，可分为以下几种情况：

第一种：以时间最短作为最优路线的模型：行程时间 T_k 等于乘车时间与转车时间之和。

$$\begin{aligned} \text{Min } T_k &= 3 \times \sum_{m=1}^{Nl_k+1} (MS_{k,m} - 1) + 5 \times Nl_k \\ m &= 1, 2, \dots, Nl_k + 1; \quad k = 1, 2, \dots, k \end{aligned}$$

其中，第 k 路线是以上转乘路线中的一种或几种。

第二种：以转乘次数最少作为最优路线的模型： $\text{Min} N_{lk}$

此模型等效为以上转乘路线按直达、转乘一次、两次的优先次序来考虑。

第三种：以费用最少作为最优路线的模型： $\text{Min} C_k = \sum_m CL_{k,m}$

6.3 问题一模型的算法描述

针对该问题的优化模型，我们采用基于铁路优先的层次搜索算法找出任意两个站点间的可行路线，然后搜索出最优路线。现将此算法运用到该问题中，结合图 2 叙述如下：（该图中的 $S_{i,g}$ 、 $S_{i,g}'$ 、 $S_{1,1}$ 、 $S_{2,1}$ 、 $S_{2,2}$ 表示列车站点， L_1 、 L_2 、 L_3 、 L_4 、 L_5 、 L_6 表示列车线路。其中（a）、（b）、（c）图分别表示了从点 $S_{i,g}$ 到点 $S_{i,g}'$ 直达、转乘一次、转乘两次的情况）

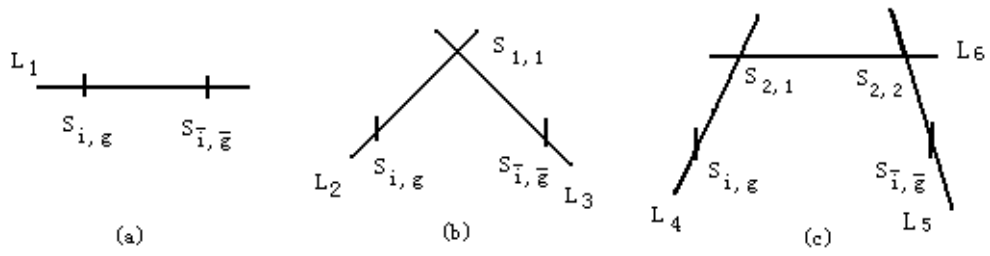


图2 列车直达、转乘图

第一种情况：首先输入需要查询的两个站点 $S_{i,g}$ 与 $S_{i,g}'$ （假设 $S_{i,g}$ 为起始站， $S_{i,g}'$ 为终点站）；

然后，搜索出经过 $S_{i,g}$ 的列车线路 L_i ($i=1,2,\dots,m$) 和经过 $S_{i,g}'$ 的列车线路 L_i' ($i=1,2,\dots,n$)，存入数据文件；判断是 L_i 与 L_i' 是否存在相同路线，若有则站点 $S_{i,g}$ 与 $S_{i,g}'$ 之间有直达路线(如图2中的 L_1)，则该路线是换乘次数最少(换乘次数等于0)的路线，若有多条直达路线，则可以在此基础上找出时间最省的路线；这样可以找出所有直达路线，存入数据文件；

第二种情况：找出经过 $S_{i,g}$ 的列车线路 L_i (如图2中的 L_2) 中的另一站点 $S_{i1,g1}$ 和经过 $S_{i,g}'$ 的列车线路 L_i' 中的另一站点 $S_{i1,g1}'$ 。判断 $S_{i1,g1}$ 与 $S_{i1,g1}'$ 中是否存在相同的点，若存在(如图2中的 $S_{1,1}$) 则站点 $S_{i,g}$ 与 $S_{i,g}'$ 间有一次换乘的路线(如图2中的 L_2 与 L_3)，该相同点即为换乘站点；这样又找出了一次换乘路线，存入数据文件；

第三种情况：再搜索出经过 L_i (如图2中的 L_4) 线路上除了站点 $S_{i,g}$ 的另一站点 $S_{i2,g1}$ (如图2中的 $S_{2,1}$) 的列车线路 L_{i6} (如图2中的 L_6)，找出列车线路 L_{i6} 上的其它站点 $S_{i2,g2}$ ；判断，如果 $S_{i2,g2}$ 与经过 $S_{i,g}'$ 的列车线路 L_i' 中的其它站点 $S_{i2,g2}'$ 存在相同的点(如图2中的 $S_{2,2}$)，则 $S_{i,g}$ 与 $S_{i,g}'$ 间有二次换乘的路线(如图2中的 L_4 、 L_6 、 L_5)，该相同点和点 $S_{i2,g1}$ 是换乘站点；将此二次换乘的路线存入数据文件中；

最后，对上述存储的经过两个站点 $S_{i,g}$ 与 $S_{i,g}'$ 的不同路线，根据不同模型进行最优路线进行搜索，得出查询者满意的最优路线。

6.4 问题一的求解

根据上述所建立的模型，我们将问题分为三种情况求解：

6.4.1 第一种：列车直达的情况：

我们在编写程序时，首先，将求出经过 A 地所有的列车车次的集合为 L_i 。然后，求出经过 B 地所有的列车车次的集合 L_i' 。再求取 L_i 和 L_i' 的交集，最后在所求交集中排除 B→A 的列车。运用 MATLAB 软件得到，丹东→宜昌、天津→拉萨、白城→青岛这 3 条路线中，没有直达的列车。

6.4.2 第二种：转乘一次的情况

首先，求出经过 A 地所有的列车车次中的所有站点的集合为 L_o 。然后，求出经过 B 地所有列车车次中的所有站点的集合为 L_o' 。最后，求取 L_o 和 L_o' 的交集。

则丹东→宜昌、天津→拉萨、白城→青岛这 3 条路线中，所求的列车如下：

从丹东→宜昌的列车路线有：K698/K695→上海→K188/K189，K698/K695→苏州→K188/K189，K698/K695→无锡→K188/K189，K698/K695→常州→K188/K189，K698/K695→镇江→K188/K189，K698/K695→南京→K188/K189。

从天津→拉萨的列车路线有：T122/T123→郑州→T264/T265，T122/T123→武昌→T264/T265，T122/T123→长沙→T264/T265，T122/T123→广州→T264/T265，T122/T123→郴州→T264/T265，T238/T235→郑州→T264/T265，T238/T235→武昌→T264/T265，T238/T235→长沙→T264/T265，T238/T235→郴州→T264/T265，T253→郑州→T264/T265，T253→武昌→T264/T265，T253→长沙→T264/T265，T253→广州→T264/T265，T253→郴州→T264/T265。

从白城→青岛的列车路线有：K7392/K7393→梅河口→K972/K969，K7392/K7393→通化→K972/K969，K7392/K7393→三源浦→K972/K969，K7392/K7393→柳河→K972/K969。

6.5 问题一的分析

由上面的假设三知，在转乘时，我们需要有 30 分钟的缓冲时间。所以，根据假设三，我们刷选出符合条件的车次。然后，为了方便计算，提出假设二：转乘车的类型要一样。因此，我们求出各种情况下的车次线路如下：

从天津到拉萨的车次线路为：

表 1：天津到拉萨的车次

出发站→终点站	耗时（分钟）	转乘次数（次）	硬座费用（元）	硬卧费用（元）	软座费用（元）
天津（T122/T123）→广州→拉萨（T264/T265）	5007	1	654	1209	2106
天津（T238/T235）→武昌→拉萨（T264/T265）	3594	1	431	829	1366

从白城到青岛的车次线路为：

表 2：白城到青岛的车次

出发站→终点站	耗时（分钟）	转乘次数（次）	硬座费用（元）	硬卧费用（元）	软座费用（元）
白城（K7392/K7393）→梅河口→青岛（K972/K969）	2976	1	239	343	555
白城（K7392/K7393）→通化→青岛（K972/K969）	2963	1	259	471	681
白城（K7392/K7393）→三源浦→青岛（K972/K969）	2957	1	254	408	586
白城（K7392/K7393）→柳河→青岛（K972/K969）	2896	1	246	397	570

从丹东到宜昌东的车次路线为：

表 3： 丹东到宜昌东的车次

出发站→终点站	耗时(分钟)	转乘次数(次)	硬座费用(元)	硬卧费用(元)
丹东 (K190/K187) → 镇江 → 宜昌东 (D3006/D3007)	2260	1	572	684
丹东 (K190/K187) → 南京 → 宜昌东 (D3006/D3007)	2220	1	703	881

通过对上述表格的路线分析，选择其中用时最少、花费最少的路线。得到的结果如下表：

表 4： 各条路线的最优路线

出发站→终点站	最佳路线	转乘次数(次)	硬座费用(元)	硬卧费用(元)
丹东 (K190/K187) → 镇江 → 宜昌东 (D3006/D3007)	2260	1	572	684
天津 (T238/T235) → 武昌 → 拉萨 (T264/T265)	3594	1	431	829
白城 (K7392/K7393) → 柳河 → 青岛 (K972/K969)	2896	1	246	397

6. 问题二的解答

对于问题二，问题的目的是使旅行的费用最少，要做的决策是如何设定旅行路线，决策所受的约束条件是：每个城市都必须去，但仅能去一次。按题目所给，将决定变量，目标函数和约束条件用数学符号及式子表示出来。

6.2 模型的建立

对于 6 个城市的旅行问题，设宜昌、上海、南京、杭州、苏州、无锡这 6 个城市分别对应 $v_1, v_2, v_3, v_4, v_5, v_6$ 。根据这 6 个城市，建立一个哈密尔顿道路（在一个无向图 $G(V, E)$ 中，经过每个顶点一次且仅一次的道路）。其中，总长最小的哈密尔顿回路的图称为最优哈密尔顿回路，总长最小的旅行商回路称为最优旅行商回路。

定义 0-1 整数型变量 $x_{ij} = 1$ 表示从城市 i 旅行到城市 j ，否则 $x_{ij} = 0$ 。

则旅行问题的数学模型可表示为一个整数规划问题。

其旅行的总费用为： $\min \sum_{i,j} d_{ij} x_{ij}$ ，其模型所需要满足的条件为

$$\begin{aligned}
 s.t. \quad & \sum_{j=1}^n x_{ij} = 1, i = 1, 2, \dots, n \\
 & \sum_{i=1}^n x_{ij} = 1, j = 1, 2, \dots, n \\
 & \sum_{i,j \in s} x_{ij} \leq |s| - 1, 2 \leq |s| \leq n - 1, s \in \{1, 2, \dots, n\}
 \end{aligned}$$

其中, d_{ij} : 城市 i 旅行到城市 j 的总费用; $|s|$: 集合 s 中元素的个数; 模型中的对称距离 TSP: $d_{ij} = d_{ji}, \forall i, j$; 非对称距离 TSP: $d_{ij} \neq d_{ji}, \exists i, j$

6.3 模型二的求解

6.4 模型二的分析

7.模型的评价、改进和推广

7.1 模型的评价

7.1.1 对于问题一的模型:

优点: 基于线路优先的层次搜索模型比较全面的考虑了城市列车的换乘次数, 乘车时间的花费和经济的开销等, 与实际贴近。而且采用了双邻接表的方法存储铁路线路的网络, 这种存储结构非常适合线路优先的层次搜索策略, 因而在优选乘车路线的过程中效率较高, 可有效的避免了对较差路径的无谓搜索, 同时能对结果进行预测评估, 可有效的减少搜索路径, 从而更容易收敛得到最优解。因此, 可以提高查询效率。

缺点: 在建模与编程的过程中, 使用的数据只是现实数据的一种近似, 因而得出的结论可能与现实情况有一定的差距。

7.1.2 对于问题二的模型

优点: 关于问题二中旅行商的问题所建立的 0-1 模型, 具有很好的时间复杂度, 可以在很短的时间内求出问题的最优解, 从而, 提高解题的效率。

缺点: 随着数据量的增加, 其求解结果的精确度会下降。

7.2 模型的推广

其中, 关于第二问的旅行商问题, 还可以推广为: 最小哈密顿的链接问题、瓶颈问题、非对称旅行商的问题、多人旅行商问题、多目标旅行商问题、依次排序问题和载货量受限的选路径的问题。关于以上的问题, 都可以用模型二来解决。

8.参考文献

[1]黄厚生, 求解旅行商问题的新方法的研究, 天津大学管理学院, 2005

[2]龚翱, 改进的城市公交查询算法的研究, 湖南大学, 2008

9.附录

问题一的程序:

```
[A,B,C]=xlsread('fujianyi.xls');
line=unique(C(:,1));
m=cell(5000,63);
for i=1:length(line)
    index=find(strcmp(C(:,1),line{i}));
    b=C(index,2)';
    for j=1:length(C(index,2)')
        m(i,j)=b(j);
    end
end
```

```

        end
    end

function [num, str]=transform(str, d, mm)
for i=1:length(d)
    if isequal(d{i}, str);
        num=i;
    end
end

function [xl1]=luxian(num1, mm)
    [xl1, p]=find(mm==num1);
end
load mm
load zhandian1
str1= input('请输入起点站 str1:')
str2= input('请输入终点站 str2:')
[num1]=transform(str1, d, mm)
[num2]=transform(str2, d, mm)
[xxl1]=luxian(num1, mm);%经过 num1 的车次
[xxl2]=luxian(num2, mm);%经过 num2 的车次
zd=intersect(xxl1, xxl2);
aa=mm(xxl1, :);%经过 num1 的车次的所有站点
[p, q]=size(aa);
aa=reshape(aa, 1, p*q);
bb=mm(xxl2, :);
[y, u]=size(bb);
bb=reshape(bb, 1, y*u);%经过 num2 的车次的所有站点
ss=intersect(aa, bb);
%ss(find(ss==0))=[];%对 ss 去 0
for i=1:length(ss)
    [l1]=luxian(ss(i), mm);%经过 ss 站点的所有车次
end
Lac=intersect(xxl1', l1')
Lcb=intersect(xxl2', l1')
zz=[];
for i=1:length(Lac);
    for j=1:length(Lcb);
        zz=[zz, intersect(mm(Lac(i), :), mm(Lcb(j), :))];%转乘的
        站点
    end
end
zz=unique(zz);
zz(find(zz==0))=[];%对 zz 去 0

```

```

for i=1:length(Lac)
    inda=find(mm(Lac(i),:)==num1);
    for j=1:length(Lcb)
        indb=find(mm(Lcb(j),:)==num2);
        for k=1:length(zz)
            indc1=find(mm(Lac(i),:)==zz(k));
            indc2=find(mm(Lcb(j),:)==zz(k));
            if inda<indc1&indc2<indb;
                fprintf(' %s--%s--%s\n',line{Lac(i)},d
{zz(k)},line{Lcb(j)})
            end
        end
    end
end
for i=1:length(aa)
    dd=aa(i); %第一个转乘点
    for j=1:length(bb)
        cc=bb(j); %第二个转乘点
        [xx13]=luxian(dd,mm);%经过第一个转乘点的所有车次
        [xx14]=luxian(cc,mm);%经过第二个转乘点的所有车次
        zc2c=intersect(xx13,xx14);
        for k=1:length(zc2c)
            if ismember(mm(zc2c(k),:),dd(i))&ismember(mm
(zc2c(k),:),cc(j))
                indd2=find(mm(zc2c(k),:)==aa(i));
                indc2=find(mm(zc2c(k),:)==bb(j));
                for q=1:length(xx11)
                    inda=find(mm(xx11(q),:)==num1);
                    inddd1=find(mm(xx11(q),:)==aa(i));
                    for p=1:length(xx12)
                        indb=find(mm(xx12(p),:)==num2);
                        indc1=find(mm(xx11(p),:)==bb(j));
                        if inda<indd&inddd1<indc&indc1<ind&in
dd2<indc2
                            fprintf(' %s--%s--%s--%s--%s\n',line{x
x11(q)},d{aa(i)},line{zc2c(k)},d(bb(j)),line{xx12(p)})
                        end
                    end
                end
            end
        end
    end
end
end
end

```

问题二的程序：