

# 铁路最优路线问题

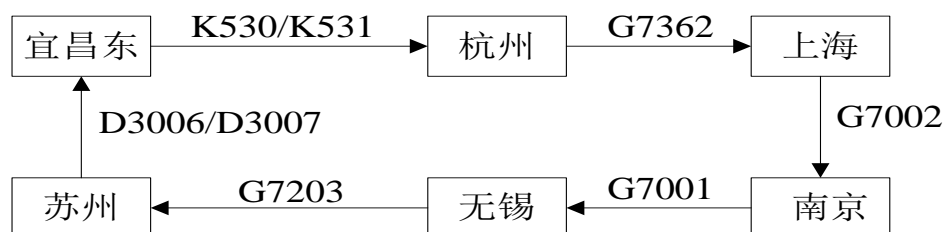
## 摘要

本文要解决的是铁路旅客选择最优乘车路线问题,综合考虑不同旅客选择出行路线的影响因素,分别建立了多目标最优化模型和 0-1 整数规划模型对不同的问题进行求解。

针对问题一:限制换乘次数最多不超过两次,以乘车时间,票价费用,换乘间隔时间损失最小为目标函数,建立了有约束的多目标最优化模型。根据路网结构、列车开行特点并结合信息熵法设计求解算法,通过 MATLAB 语言编程实现,对问题求解后得到如下结果:

旅途	最优路线	乘车费用/元	乘车时间/h
丹东→宜昌	K190/K187 在南京站转乘 K696/K697	399.735	40.8
天津→拉萨	Z194/Z191 在石家庄北站转 T27	460.8	58.4
白城→青岛	K7310 在长春站转 K704/K701	282.96	27

针对问题二:首先利用图论知识将所求问题转换为最佳旅行商回路问题,然后综合考虑每个城市经过一次,且最终回出发地点等约束条件,基于问题一的基础,以所有边的权值之和最小为目标函数,建立 0-1 整数规划模型。通过 LINGO 软件编程求解得到最优旅游路线如下:



经过计算后,得到该路线的总乘车时间为 31 小时 45 分钟,最小乘车费用为 827 元。

最后对模型的优缺点进行评价,并提出了改进与推广。

关键词: 多目标最优化      0-1 整数规划      信息熵法      最佳旅行商回路

# 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、安全性高等优点。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 题目所给信息

附录 1 给出了 2013 年全国列车时刻表数据。

包含以下信息：列车车次、列车类型（普快，空调快速，动车…）、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。

## 1.3 需要解决的问题

根据以上信息解决以下问题：

- 1) 给出任意两个站点之间的最优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售罄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。
- 2) 假设你从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 模型假设与符号说明

## 2.1 模型假设

- (1) 假设旅客乘坐不同类型火车时均选择该类火车中的最低票价；
- (2) 假设所有火车都按时刻表正常运行，无晚点，早点等意外情况发生；
- (3) 假设旅客选择出行路线主要与乘车时间，票价费用，换乘次数，换乘间隔时间有关，忽略其他因素的影响；
- (4) 假设在旅游时，经过每个城市只逗留一次。

## 2.2 符号说明

符号	符号说明
$o$	出发地站台
$d$	目的地站台
$k$	列车编号
$l_k$	编号 $k$ 的列车
$L$	所有列车的集合
$v_{l_k}^i$	$l_k$ 列车经过的站点
$V_{l_k}$	$l_k$ 列车经过的站点集合
$V$	所有站点的集合
$(v_{l_k}^i, v_{l_n}^j)$	表示列车 $l_k$ 在 $i$ 站至 $j$ 站换乘列车 $l_n$
$V^t$	换乘对点 $(v_{l_k}^i, v_{l_n}^j)$ 的集合
$x_{ij}^{l_k}$	从 $i$ 站到 $j$ 站是否乘坐 $l_k$ 次列车
$t_{l_k}^1(i)$	列车 $l_k$ 驶进 $i$ 站的时刻
$t_{l_k}^2(i)$	列车 $l_k$ 离开 $i$ 站的时刻
$t_{ij}(l_k)$	乘 $l_k$ 次列车从 $i$ 站到 $j$ 站所用的乘车时间
$t_{ij}(l_1, l_2)$	$l_1$ 次列车从 $i$ 站到 $j$ 站换乘 $l_2$ 次列车所需换乘间隔时间
$p_{ij}(l_k)$	乘 $l_k$ 次列车从 $i$ 站到 $j$ 站所用的票价
$z_i$	目标 $i$ 的值
$L_k$	经过 $o$ 站和 $d$ 站的列车
$L_m$	表示只经过 $o$ 站的列车集合
$L_n$	表示只经过 $d$ 站的列车集合
$L_q$	表示不经过 $o$ 站和 $d$ 站的列车集合
$m$	可行路径条数
$H_j$	评价指标的熵值
$x_{ij}$	表示游客是否从城市 $i$ 到城市 $j$ ,
$w_{ij}$	城市 $i$ 到城市 $j$ 的权值

### 3 问题分析

在实际应用中,由于出行者考虑问题的角度不同,对最佳乘车路线的理解就不同,该问题不能简单地抽象为最短路问题,需要以不同出行者出行路径选择的影响因素来进行综合性的分析。具体分析如下:

**对问题一的分析:**题目要求给出任意两个站点之间的最优铁路路线的模型以及算法。由此可知应该采用有约束的多变量最优化模型进行分析。对于一般旅客来说,选择出行路径主要考虑乘车时间,票价费用,换乘次数,换乘间隔时间等因素,由于旅客一般不会考虑换乘次数较多的路径,所以为了方便计算最优乘车路线,限制最大换乘次数为2,并且在此情况下保证从出发站到达目的站有可行路径。综上分析,可以以乘车时间,票价费用,换乘间隔时间损失最小为目标函数,建立有约束的多目标最优化模型<sup>[1]</sup>,并给出求解模型的相关算法。最后分别对直达,一次换乘,两次换乘这三种情况进行分析,找出所有可行路径,再从中求解最优铁路路线,

**对问题二的分析:**该问题要求给出一次旅游中的最优行程路线。可以分别运用旅行商回路问题<sup>[2]</sup>和数学规划方法建模求解,具体如下:首先利用图论知识对所求问题进行适当的转换,将每个旅游城市看作图中的一个节点,各城市之间的铁路路线看作图中对应节点间的边,各条铁路路线的综合目标值看作对应边上的权,那么所给各城市间的铁路网就转化为加权网络图 $G$ ,根据题意可知,需要寻找一条遍历各个节点的最优回路,即找到一条所有边的权值之和最小的回路。然后综合考虑每个城市经过一次,且最终回出发地点等约束条件,以所有边的权值之和最小为目标函数,建立0-1整数规划模型进行求解,得到最优旅游路线。

### 4 数据分析与处理

#### 4.1 简化列车票价

在我国铁路网中,列车的等级可分为:动车组、直特列车、特快列车、快速列车及普快列车等,席座可分为:一等座、二等座、特等座、软座、硬卧、软卧等。铁路票价因列车等级、席座的不同而各异。

若分别分析各个等级及席座的列车票价,计算量将非常庞大,为了简化计算,需要对列车票价进行适当简化。由于乘客在选择列车座位时,出于经济方面的考虑,更倾向于选择票价较低的席座。因此,在计算过程中统一选用各列车等级的最低票价(如普通列车选用硬座票价)。

## 4.2 限制换乘次数

换乘次数是指乘客在完成一次出行过程中所换列车的次数，一般情况下，乘车时间差不多时，旅客更倾向于选择换乘次数较少的路径<sup>[3]</sup>。由于换乘次数是直接影响乘客路径选择的一个重要因素，为了方便计算最优乘车路线，取最大换乘次数为 2 。即：

$$N_{\max} = 2$$

所以在已知起点站和终点站时，乘车路线可以分为 3 种情况：直达，换乘一次，换乘两次，可以有效地简化分析过程。

## 5 问题一的解答

### 5.1 模型准备

根据对问题一的分析，在已知起点站和终点站时，首先找出所有的可行路线，再综合考虑不同旅客选择出行路线的影响因素，以总乘车时间最小、总票价费用最少、换乘间隔时间损失最小为目标函数，最大换乘次数以及在换乘次数的限制下出发地到目的地存在可达路径为限制条件，建立多目标铁路旅客乘车方案优化模型并进行求解。

### 5.2 模型导出

#### 5.2.1 目标函数的确定

旅客选择出行路径的主要影响因素包括乘车时间、票价费用、换乘次数、换乘间隔时间等。根据对问题一的分析，限制最大换乘次数为 2 ，接下来依次对乘车时间、票价费用、换乘间隔时间进行分析。

假设列车车次为  $l_k$  ，其经过的站点为  $V_{l_k}$  ，则可以得到所有列车的集合为：

$$L = \{l_k \mid k = 1, 2, \dots, N\}, \text{ 途经站点的集合为: } V = \{V_{l_k} \mid l_k \in L, k = 1, 2, \dots, N\}。$$

$x_{ij}^{l_k}$  表示从  $i$  站到  $j$  站是否乘坐了  $l_k$  次列车，则有：

$$x_{ij}^{l_k} = \begin{cases} 1, & \text{乘 } l_k \text{ 次列车从 } i \text{ 站到 } j \text{ 站} \\ 0, & \text{else} \end{cases}, i \in V_{l_k}, l_k \in L$$

#### 目标一：总乘车时间

在一列火车上的乘车时间为终点站进站时刻与起点站出站时刻之间的时间差。假设列车  $l_k$  驶进  $i$  站的时刻为  $t_{l_k}^1(i)$ ，离站时刻为  $t_{l_k}^2(i)$  ( $i \in V_{l_k}$ )，用  $t_{ij}(l_k)$  表

示乘  $l_k$  次列车从  $i$  站到  $j$  站所用的乘车时间，则有：

$$t_{ij}(l_k) = t_{l_k}^1(j) - t_{l_k}^2(i)$$

$o$ 、 $d$  分别为乘客出行的始发站、终点站，考虑到其中可能的换乘情况，则路途中的总乘车时间为所乘坐列车的乘车时间之和，即：

$$z_1 = \sum_{i \in V} \sum_{j \in V} \sum_{l_k \in L} [t_{l_k}^1(j) - t_{l_k}^2(i)] \cdot x_{ij}^{l_k}$$

### 目标二：总票价费用

根据分析，乘车票价选用各列车种类的最低票价（如普通列车选用硬座票价）。假设乘  $l_k$  次列车从  $i$  站到  $j$  站所用的票价为  $p_{ij}(l_k)$ ，则可以得到乘客从出发地至目的地途中所花费的总票价费用为：

$$z_2 = \sum_{i \in V} \sum_{j \in V} \sum_{l_k \in L} p_{ij}(l_k) \cdot x_{ij}^{l_k}$$

### 目标三：中转换乘损失

换乘间隔时间为从  $l_1$  次列车换乘  $l_2$  次列车的时间，即  $l_2$  次列车出发时刻与  $l_1$  次列车到达时刻之间的时间差。考虑到列车运行时间的随机性，假设乘客能赶上列车的最短换乘间隔时间为  $T_0$ ，取  $T_0 = 20\text{min}$ ，无法在  $T_0$  内赶到换乘车站时，时差应加一天，即  $1440\text{min}$ 。则乘坐  $l_1$  次列车到达  $i$  站，然后在  $j$  站换乘  $l_2$  次列车（一般  $i$  站和  $j$  站为同一车站或统一城市内的车站）所需的换乘间隔时间为：

$$t_{ij}(l_1, l_2) = \begin{cases} t_{l_2}^2(j) - t_{l_1}^1(i) & , t_{l_2}^2(j) - t_{l_1}^1(i) \geq T_0 \\ t_{l_2}^2(j) - t_{l_1}^1(i) + 1440 & , t_{l_2}^2(j) - t_{l_1}^1(i) < T_0 \end{cases}$$

换乘间隔时间的长短直接影响着乘客出行的舒适度。如果换乘间隔时间过短，则可能会错过换乘列车，造成车票的浪费，如果换乘间隔时间过长，则需要等待较长时间，跨天时还需要吃饭、住宿等，增加换乘成本。一般来说，换乘间隔时间在 3~5 小时较为合理。为了描述换乘时间间隔对乘客产生的效用，可以用其损失大小对它进行量化，建立损失评价函数如下：

$$r[t_{ij}(l_1, l_2)] = \begin{cases} 1 + \frac{t_{ij}(l_1, l_2) - 20}{180 - 20} & , 20 \leq t_{ij}(l_1, l_2) < 180 \\ 1 & , 180 \leq t_{ij}(l_1, l_2) < 300 \\ 1 + \frac{t_{ij}(l_1, l_2) - 300}{1460 - 300} & , 300 \leq t_{ij}(l_1, l_2) < 1460 \\ 2 & , t_{ij}(l_1, l_2) > 1460 \end{cases}$$

在路途中，乘客由于中转换乘造成的总损失的影响因素主要包括换乘次数和换乘间隔损失，故采用二者的乘积对它进行描述。得到其表达式为：

$$z_3 = \sum_{i \in V} \sum_{j \in V} \sum_{l_k \in L} r[t_{ij}(l_1, l_2)] \cdot (x_{ij}^{l_k} - 1)$$

### 5.2.2 约束条件的确定

1.乘客必须选择 1 列火车从出发地  $o$  站台出发，故有

$$\sum_{j \in V} \sum_{l_k \in L} x_{oj}^{l_k} = 1$$

2.在换乘次数的限制下，需保证出发地到目的地是可达的，所以有：

$$\sum_{i \in V} \sum_{l_k \in L} x_{im}^{l_k} - \sum_{j \in V} \sum_{l_k \in L} x_{mj}^{l_k} = 0$$

其中， $\forall m \in V, m \neq o, d$

3.根据分析，最大中转换乘次数为 2，即：

$$\sum_{i \in V} \sum_{j \in V} \sum_{l_k \in L} x_{ij}^{l_k} - 1 \leq 2$$

4.乘客必须乘坐一列火车到达终点站  $d$  站台，则有：

$$\sum_{i \in V} \sum_{l_k \in L} x_{id}^{l_k} = 1$$

### 5.2.3 模型的建立

综上所述，建立多目标铁路旅客乘车方案优化模型<sup>[1]</sup>为：

$$\begin{cases} \min z_1 = \sum_{i \in V} \sum_{j \in V} \sum_{l_k \in L} [t_{l_k}^1(j) - t_{l_k}^2(i)] \cdot x_{ij}^{l_k} \\ \min z_2 = \sum_{i \in V} \sum_{j \in V} \sum_{l_k \in L} p_{ij}(l_k) \cdot x_{ij}^{l_k} \\ \min z_3 = \sum_{i \in V} \sum_{j \in V} \sum_{l_k \in L} r[t_{ij}(l_1, l_2)] \cdot (x_{ij}^{l_k} - 1) \end{cases}$$

$$s.t. \begin{cases} \sum_{j \in V} \sum_{l_k \in L} x_{oj}^{l_k} = 1 \\ \sum_{i \in V} \sum_{l_k \in L} x_{im}^{l_k} - \sum_{j \in V} \sum_{l_k \in L} x_{mj}^{l_k} = 0 \\ \sum_{i \in V} \sum_{j \in V} \sum_{l_k \in L} x_{ij}^{l_k} - 1 \leq 2 \\ \sum_{i \in V} \sum_{l_k \in L} x_{id}^{l_k} = 1 \end{cases}$$

## 5.3 模型求解及结果分析

### 5.3.1 考虑换乘次数的可行路线的具体算法：

在已知起始站和终点站时，可以利用旅客列车时刻表的站点序列得到多条可行路线。根据分析，分情况找出所有直达，换乘一次，换乘两次的路线即可，求解思路如下：

**求解直达路线：**首先分别找出经过起始站  $o$  站和终点站  $d$  站的所有列车，即  $L_o$  和  $L_d$ ，然后令  $L_k = L_o \cap L_d$ ，得到经过  $o$  站和  $d$  站的列车  $l_k \in L_k$ 。当列车在  $o$  站的发车时刻早于其在  $d$  站的到达时刻，即  $t_k^2(o) < t_k^1(d)$  时，列车可以从  $o$  站直达  $d$  站，于是可以得到所有直达列车的集合。

**求解换乘一次路线：**中转换乘只能发生在  $o$  站以后的站台和  $d$  站以前的站台之间。令  $L_m = L_o - L_d$  表示只经过  $o$  站的列车， $L_n = L_d - L_o$  表示只经过  $d$  站的列车，若从  $o$  站乘坐  $l_m$  次列车在  $v_{l_m}, v_{l_n}$  站间可以换  $l_n$  次列车到达  $d$  站，即对于所有的  $v_{l_m} \in V_{l_m}^o, v_{l_n} \in V_{l_n}^d$ ，有  $(v_{l_m}, v_{l_n}) \in V^t$  且  $t_{l_m}^2(o) < t_{l_m}^1(v_{l_n})$ ， $t_{l_n}^2(v_{l_n}) < t_{l_n}^1(d)$ ，则可以得到换乘一次的路线。

**求解换乘两次路线：**换乘两次表明旅客需乘坐 3 列火车，令  $L_q = L - L_o \cup L_d$ ， $l_q \in L_q$  表示不经过  $o$  站和  $d$  站的列车，则可以得到旅客换乘两次的可行路线为：从  $o$  站乘坐  $l_m$  次列车在站台  $v_{l_m}, v_{l_q}^1$  之间换乘  $l_q$  列车，之后在站台  $v_{l_q}^2, v_{l_n}$  之间换乘  $l_n$  列车到达  $d$  站。即当  $\forall v_{l_m} \in V_{l_m}^o, v_{l_q}^1, v_{l_q}^2 \in V_{l_q}, v_{l_n} \in V_{l_n}^d$ ，有  $(v_{l_m}, v_{l_q}^1) \in V^t$ ， $(v_{l_q}^2, v_{l_n}) \in V^t$  且  $t_{l_q}^2(v_{l_q}^1) < t_{l_q}^1(v_{l_q}^2)$  时，可以得到换乘两次的路线。

根据上述思路，得到旅客出行乘车的可行方案具体算法<sup>[4]</sup>步骤如下：

Step1: 令最大换乘次数  $N_{\max} = 2$ ， $L_o = \{l | o \in V_l, l \in L\}$ ， $L_d = \{l | d \in V_l, l \in L\}$ ；

Step2: 若  $L_o = \emptyset$  或  $L_d = \emptyset$ ，则停止计算，否则令  $L_k = L_o \cap L_d$ ；

Step3: 如果  $L_k = \emptyset$ ，转至 Step4，否则，对于所有的  $l_k \in L_k$ ，若有  $t_{l_k}^2(o) < t_{l_k}^1(d)$ ，输出  $l_k$  次列车从  $o$  站到  $d$  站的路径，转至 Step4；

Step4: 令  $L_m = L_o - L_d$ ， $L_n = L_d - L_o$ ，若  $L_m = \emptyset$  或  $L_n = \emptyset$ ，则停止计算，否则令  $V_{l_m}^o = \{v_{l_m} \in V_{l_m} \text{ 且 } t_{l_m}^2(o) < t_{l_m}^1(v_{l_m})\}$ ， $V_{l_n}^d = \{v_{l_n} \in V_{l_n} \text{ 且 } t_{l_n}^2(v_{l_n}) < t_{l_n}^1(d)\}$ 。若  $\forall v_{l_m} \in V_{l_m}^o, v_{l_n} \in V_{l_n}^d$ ，有  $(v_{l_m}, v_{l_n}) \in V^t$ ，转至 Step5，否则转至 Step6；



Step5: 输出  $l_m$  列车从  $o$  到  $V_{l_m}$  的站点及  $l_n$  列车从  $V_{l_n}$  到  $d$  的站点, 构成一次换乘路径, 转至 Step6;

Step6: 令  $L_q = L - L_o \cup L_d$ , 若  $L_q = \emptyset$ , 停止计算, 否则对于所有的  $l_q \in L_q$ , 若  $(v_{l_m}, v_{l_q}^1) \in V^t$ ,  $(v_{l_q}^2, v_{l_n}) \in V^t$ ,  $\forall v_{l_m} \in V_{l_m}^o, v_{l_q}^1 \in V_{l_q}, v_{l_q}^2 \in V_{l_q}, v_{l_n} \in V_{l_n}^d$ , 且  $t_{l_q}^2(v_{l_q}^1) < t_{l_q}^1(v_{l_q}^2)$ , 转至 Step7, 否则, 停止计算;

Step7: 输出  $l_m$  列车从  $o$  到  $V_{l_m}$  的站点,  $l_q$  列车从  $V_{l_q}^1$  到  $V_{l_q}^2$  的站点及  $l_n$  列车从  $V_{l_n}$  到  $d$  的站点, 构成两次换乘路径。

算法流程图如图 1 所示:

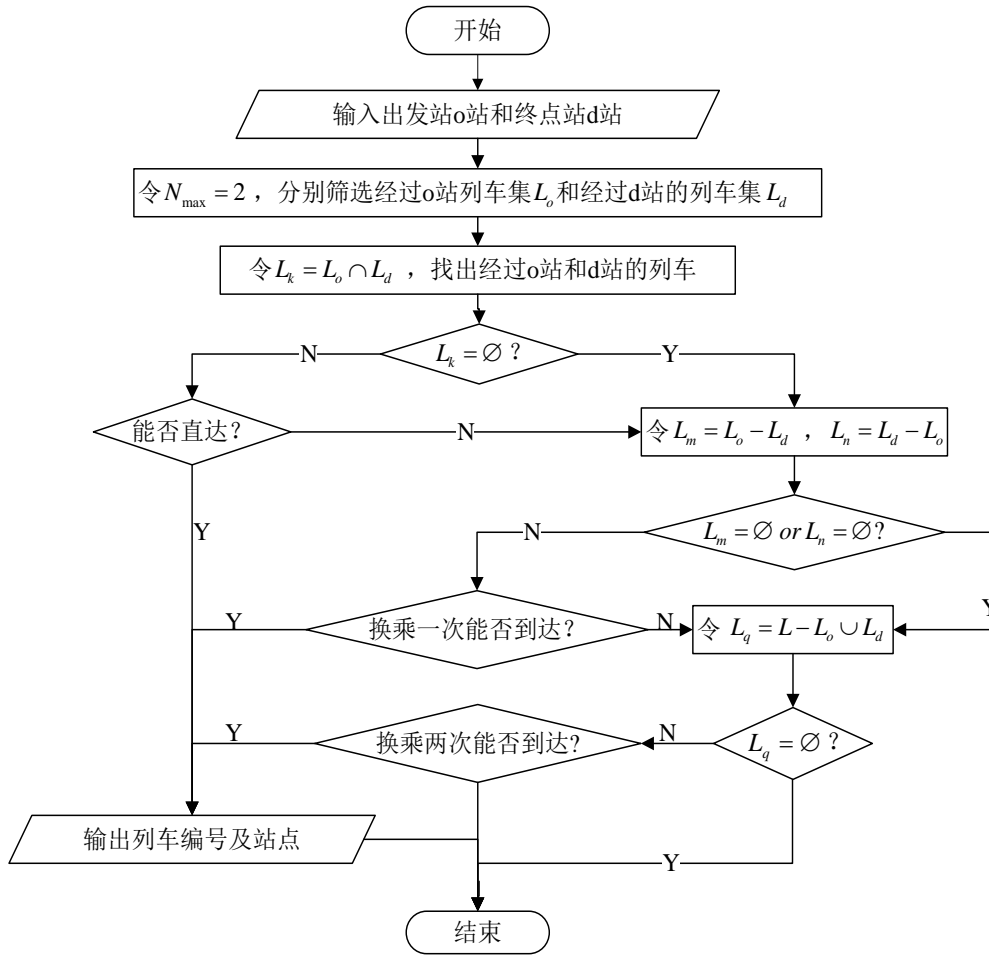


图 1: 旅客乘车可行方案算法流程图

### 5.3.2 基于多目标最优化模型的最优路线具体算法:

根据上述算法, 可以得到在换乘次数不超过 2 次的情况下, 从  $o$  站到  $d$  站所有可行路径的集合, 接下来根据多目标铁路旅客乘车方案最优化模型利用信息熵法<sup>[1]</sup>从中选择最优路线。

设可行路径条数为  $m$ ，那么， $m$  条可行路径及 3 个目标值就构成了一个决策矩阵，设这个决策矩阵为  $M$ ，利用熵值赋权法选择最优路径的算法如下：

Step1: 写出从  $o$  站至  $d$  站的决策矩阵  $M$ ，即由可行路径和 3 个目标值构成的矩阵；

Step2: 对决策矩阵进行规范化，采用标准差法，即  $r_{ij} = \frac{m_{ij} - \bar{m}_j}{s_j}$ ，

其中  $\bar{m}_j = \frac{1}{n} \sum_{i=1}^n m_{ij}$ ， $s_j = \sqrt{\frac{1}{n} \sum_{i=1}^n (m_{ij} - \bar{m}_j)^2}$ ，得到规范化矩阵  $R$ ；

Step3: 对规范化矩阵  $R$  进行归一化处理，即  $z_{ij} = \frac{r_{ij}}{\sum_{i=1}^n r_{ij}}$ ，得到归一化矩阵  $Z$ ；

Step4: 确定评价指标的熵值，即  $H_j = -k \sum_{i=1}^n z_{ij} \ln z_{ij}$

其中， $k$  为调节系数： $k = \frac{1}{\ln n}$ ， $n$  为样本数量，即路线数量； $j$  为指标数。

Step5: 将熵值转化为权重值，即  $d_j = \frac{1 - H_j}{3 - \sum_{j=1}^3 H_j}$ ， $j=1,2,3$

Step6: 计算路径  $i$  的综合目标值，即  $W_i = \sum_{j=1}^m d_j z_{ij}$ 。

其算法流程图如下：

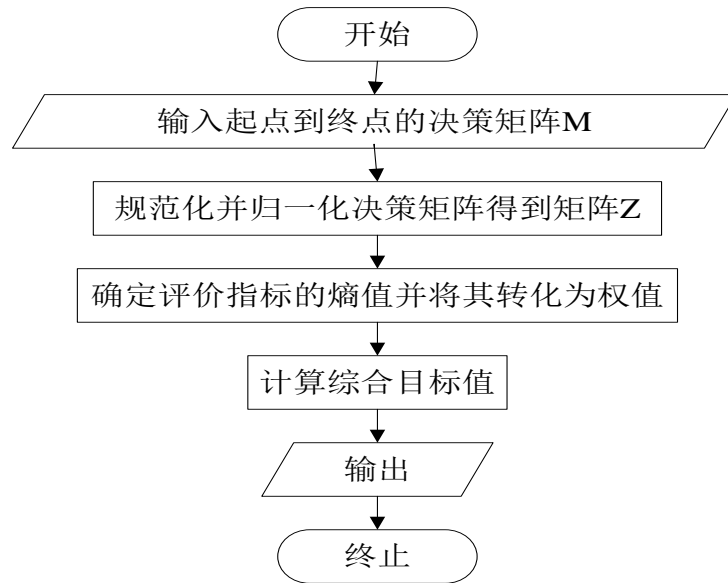


图 2: 最优路线具体算法流程图

根据计算得到的综合目标值，按照从小到大对可行方案进行排序，排在前面的路径即为较优路径。

5.3.3 根据算法求解及结果分析

根据上述算法及附录中数据，利用 MATLAB 语言编程（程序见附录 1）实现。分别选出丹东→宜昌、天津→拉萨、白城→青岛的可行乘车方案，然后基于多目标铁路旅客乘车方案优化模型<sup>[1]</sup>求解各方案的综合目标值。

经计算，丹东→宜昌、天津→拉萨、白城→青岛均没有直达路线，并且都能在一次换乘中得到最优路径，故暂不考虑中转换乘两次的情况。

1、丹东→宜昌：

经过计算可得到在换乘一次的情况下共有 6 种路径，将各可行路径的综合目标值按照大小顺序排列，结果如表 1 所示：

表 1：丹东至宜昌的较优乘车方案及综合目标值

排序	乘车方案	综合目标值
1	K190/K187 在南京站转乘 K696/K697	0.11
2	K190/K187 在上海站转乘 K696/K697	0.13
3	K190/K187 在无锡站转乘 D3006/D3007	0.18
4	K190/K187 在常州站转乘 D3006/D3007	0.18
5	K190/K187 在镇江站转乘 D3006/D3007	0.19
6	K190/K187 在苏州站转乘 D3006/D3007	0.20

根据表 1 对比各方案的综合目标值，得到丹东至宜昌的最优乘车方案为：在丹东乘坐 K190/K187 次列车在南京站转乘 K696/K697 次列车到达宜昌东站。其中换乘间隔为 84 分钟，乘车总费用为 399.735 元，乘车总时间为 2448min，即 40.8h，由此，出于实际的方面考虑，该方案是合理的。

2、天津→拉萨：

求解得到在换乘一次的情况下共有 20 种路径，经过计算可得各可行路径及其综合目标值，排在前 10 的路径的综合目标值及乘车方案如表 2 所示：

表 2：天津至拉萨的较优乘车方案及综合目标值

排序	乘车方案	综合目标值
1	Z194/Z191 在石家庄北站转 T27	0.0422
2	Z194/Z191 在太原站转 T27	0.0423
3	K388/K385 在宝鸡站转 T22/T23	0.0424
4	K7728/K7725 在北京西站转 T27	0.0425
5	T244/T241 在徐州站转 T164/T165	0.0435
6	K388/K385 在西安站转 T264/T265	0.0442
7	T184/T181 在郑州站转 T264/T265	0.0476
8	T131/T134 在蚌埠站转 T164/T165	0.0482
9	T131/T134 在南京站转 T164/T165	0.0487
10	T131/T134 在无锡站转 T164/T165	0.0491

由表 2 可知，天津至拉萨的最优乘车方案为：

在天津乘坐 Z194/Z191 次列车在石家庄北站转 T27 次列车到达拉萨站，其中换乘间隔为 102 分钟，乘车总费用为 460.8 元，乘车总时间为 3506min，即 58.4h，由此可知该方案是合理的。

**3、白城→青岛：**

经过计算可得到各可行路径及其综合目标值，在换乘一次的情况下共有 33 种可行路径。排在前 10 的路径的综合目标值及乘车方案如表 3 所示：

表 3：白城至青岛的较优乘车方案及综合目标值

排序	乘车方案	综合目标值
1	K7310 在长春站转 K704/K701	0.0251
2	K1302 在开原站转 K1056/K1053	0.0265
3	K1302 在昌图站转 K1056/K1053	0.0265
4	K1302 在沈阳北站转 K704/K701	0.0266
5	K1302 在铁岭站转 K704/K701	0.0266
6	1468 在大虎山站转 K704/K701	0.0269
7	1468 在兴城站转 K1056/K1053	0.027
8	1468 在沟帮子站转 K704/K701	0.027
9	1468 在锦州站转 K704/K701	0.027
10	1468 在山海关站转 K704/K701	0.0271

由表 3 可知，白城至青岛的最优乘车方案为：

在白城乘坐 K7310 次列车在长春站转 K704/K701 次列车到达青岛，其中换乘间隔为 311 分钟，乘车总费用为 282.96 元，乘车总时间为 1618min，即 27h，出于实际的方面考虑可知该方案是合理的。

**6 问题二的解答**

**6.1 模型准备**

根据对问题二的分析，首先利用图论知识将所求问题转换最佳旅行商回路问题，然后综合考虑每个城市经过一次，且最终回出发地点等约束条件，以所有边的权值之和最小为目标函数，建立 0-1 整数规划模型<sup>[4]</sup>进行求解。

**6.2 模型导出**

**6.2.1 目标函数的确定**

首先确定决策变量，用  $x_{ij}$  表示游客从城市  $i$  到城市  $j$ ，引入 0-1 变量。令：

$$x_{ij} = \begin{cases} 1 & , \text{旅客从城市 } i \text{ 到达城市 } j \\ 0 & , \text{旅客不从城市 } i \text{ 到达城市 } j \end{cases}$$

将每个旅游城市看作图中的一个节点,各城市之间的铁路路线看作图中对应节点间的边,各条铁路路线的综合目标值看作对应边上的权,假设城市  $i$  到城市  $j$  的权表示为  $w_{ij}$ 。

将每条路径上的权值最小作为目标函数,得到其表达式为:

$$\min \sum_{i=1}^6 \sum_{j=1}^6 w_{ij} x_{ij}$$

### 6.2.2 约束条件的确定

根据模型假设 4,约束条件为每个地点去且仅去过一次,并且 6 个地点之间必须构成一个回路。具体包括:

在旅游途中,从城市  $i$  出发能且仅能到达一个城市,即:

$$\sum_{j=1}^6 x_{ij} = 1, \quad i = 1, 2, \dots, 6$$

并且有且仅有一个城市会在旅游途中前往城市  $j$ , 即:

$$\sum_{i=1}^6 x_{ij} = 1, \quad j = 1, 2, \dots, 6$$

不考虑重复游览一个城市和除出发城市之后的返程情况,故有:

$$x_{ij} + x_{ji} \leq 1$$

旅游线路中只能有且仅有一个回路, 即:

$$x_{ij} + x_{jk} + x_{ki} \leq 2$$

### 6.2.3 模型的建立

综上所述,建立最佳旅游线路的 0-1 整数规划模型<sup>[4]</sup>如下:

$$\begin{aligned} & \min \sum_{i=1}^6 \sum_{j=1}^6 w_{ij} x_{ij} \\ & s.t. \begin{cases} \sum_{i=1}^6 x_{ij} = 1, \quad j = 1, 2, \dots, 6 \\ \sum_{j=1}^6 x_{ij} = 1, \quad i = 1, 2, \dots, 6 \\ x_{ij} + x_{ji} \leq 1 \\ x_{ij} + x_{jk} + x_{ki} \leq 2 \end{cases} \end{aligned}$$

6.3 模型求解及结果分析

利用 LINGO 软件编程<sup>[5]</sup>（程序见附录 2）对上述模型进行求解，求解得到权值最大的最佳旅游路线如下图：

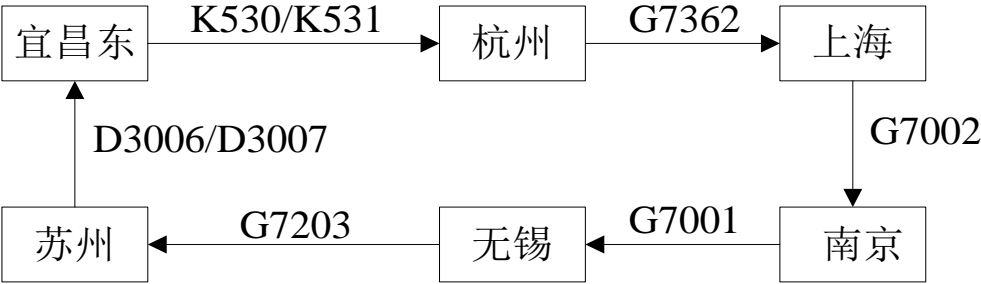


图 3：最佳旅游路线图

由图 3 可知，最佳旅游路线为宜昌东→杭州→上海→南京→无锡→苏州→宜昌东，车次见上图。经过计算，得到各行程乘车时间及费用如下表：

表 4：各行程乘车时间及费用

旅途	乘车时间	乘车费用/元
宜昌搭乘 K530/K531 次列车到达杭州	21 小时 25 分钟	168.5
杭州搭乘 G7362 次列车到达上海	1 小时 26 分钟	92.5
上海搭乘 G7002 次列车到达南京	1 小时 39 分钟	139.5
南京搭乘 G7001 次列车到达无锡	0 小时 55 分钟	79.5
无锡搭乘 G7203 次列车到达苏州	0 小时 15 分钟	19.5
苏州搭乘 D3006/D3007 次列车返回到宜昌东	7 小时 15 分钟	322.5
总行程	31 小时 45 分钟	827

由上表可知，此路线总共乘车时间是 31 小时 45 分钟，总乘车费用为 827 元。对一次旅游来说，该乘车时间相对于一次旅行来说比较合理。由于乘坐的火车多为高铁，所以费用较高，但基本还在人们的可接受的范围之内。并且乘车时间和到达时间一般在早上，在旅途中的乘车时间基本上不长，这样旅客在每个城市都能有较好的旅行时间及休息时间。故说明该模型可以很好地解决最优旅游线路问题。

7 模型的评价、改进及推广

7.1 模型评价

1、模型优点：

本文以旅客列车时刻表为基础，设计了利用每个旅客列车时刻表的站点序列

搜索出行路径的算法,该算法不需要对每个站点都进行标号就能够得到多条可行路径,提高了路径的搜索速度。

本文采用的信息熵法是一种常用的确定多目标权重的方法,由各个样本的实际数据求得最优权重,反映了指标信息熵值的效用价值,避免了人为的影响因素,因而给出的指标权重更具有客观性,从而具有较高的再现性和可信度。

关于游客旅行线路的模型,进行了合理的假设,简化了次要因素,把问题转化为图论上最佳旅行商回路问题来解决,同时给出了数学规划模型,运用 LINGO 软件进行求解,使问题得到了比较合理的解决。

## **2、模型缺点:**

问题一中建立的多目标最优化模型在考虑影响旅客选择路线的因素时,忽略了旅行的舒适度,换乘站间的距离等其他重要因素,使模型趋于理想化,可能会与实际情况有一定差距。问题二中设计旅游路线时,没有考虑在各个城市的游玩时间,有一定的局限性。若作为旅游参考,要根据实际情况来选择使用。

## **7.2 模型改进**

可以对问题一中的模型加以适当的改进,将换乘距离及舒适度考虑在内,在原有模型的基础上增加换乘距离最短,舒适度最佳这两个目标函数,重新建立多目标优化模型,使其更完善,贴近实际。

问题二中,在寻求最优回路时,可以采用单亲遗传算法,该算法基于 K—随机近邻算法,同时结合 4 种变异算子,在选择操作中又结合了贪心策略,从而具有更好的搜索能力,以此代替原模型的算法,具有更高的计算效率,及更快的搜索速度。

## **7.3 模型的推广**

本文中建立的基于最优路径算法的多目标最优化模型不但可以应用于铁路最优路线问题,在实际生活中,更可以广泛应用于互联网的寻址计算、智能交通系统、城市地理信息系统以及军事地理信息系统中。另外,通过对本题进行分析所建立起来的最佳旅行商回路模型,也能较好地解决对旅游线路的设计问题,缩短旅游的行程,提高旅游线路的性价比等等。这个模型还可以推广到其他的领域中去,例如通过对路线的合理设计,减小行车的路程。

# **8 参考文献**

- [1] 杨信丰,多目标铁路旅客乘车方案优化模型及算法研究,中国铁道科学. 2010

- [2] 冯爱芬, 最佳旅游线路的设计与算法, 中国重要会议论文, 2008.
- [3] 霍亮, 铁路旅客乘车行为分析与客流分配研究. 中南大学, 2006.
- [4] 查伟雄, 熊桂林. 铁路旅客 OD 矩阵推算模型及算法设计. 系统工程, 2004,
- [5] 谢金星、薛毅, 优化建模与 LINDO/LINGO 软件, 北京: 清华大学出版社, 2005

## 9 附录

### 附录 1

主程序

```
clc;clear;
fid=fopen('station1.txt','r');
temp=textscan(fid,'%s%s%s%s%s%s%s');
Checi=temp{1,1};
Zhanxu=temp{1,2};
Chezhan=temp{1,3};
Riqi=temp{1,4};
Daoda=temp{1,5};
Likai=temp{1,6};
Juli=temp{1,7};
%数据矩阵:车次,站点,日期,时间,路程
A=[Checi,Zhanxu,Chezhan,Riqi,Daoda,Likai,Juli];
Strat_End1(A);%转一站
Start_End2(A);%转两站
Wenti2(A);%求各个转战点的权重系数 u_max
```

问题一:

```
function [ttt1,min_checi]=Start_End(A,i1,i2)
A1=A(:,3);
A7=A(:,7);

[b1,m1,n1]=unique(A1);
n1;%城市编号
max(n1);
b={};t1={};t2={};
for i=1:max(n1)
    a=A(find(n1(:)==i),1);
    t1{i}=A(find(n1(:)==i),5);
    t2{i}=A(find(n1(:)==i),6);
    b{i}=a;
end
b;%b 为起点站的车次
size(b);
```



```

t1;%到达时间
t2;%离开时间
A2=A(:,1);
[b2,m2,n2]=unique(A2);
n2;%车次编号
%i1=input('请输入起点城市编号!\n')
%i2=input('请输入终点城市编号!\n')
i11=n2(find(n1==i1));%起点车次编号
i22=n2(find(n1==i2));
zhida_id=intersect(i11,i22);%直达车车次编号
if isempty(zhida_id)
    fprintf('此路行不通!');
    ttt1=0;
    min_checi=0;
else
    bb={ };dd={ };
    for i=1:size(zhida_id)
        A11=A(find(n2==zhida_id(i)),4:6);
        aa1=A11(find(n1(find(n2==zhida_id(i)))==i1),1:3);
        aa2=A11(find(n1(find(n2==zhida_id(i)))==i2),1:3);
        bb{i}=aa1;
        dd{i}=aa2;
    end
    bb;dd;p=[];
    for i=1:size(zhida_id)
        fprintf('直达车编号为%d 的起点时间\n',zhida_id(i));
        bb{i};
        dd{i};
        cc=char(bb{i});
        ee=char(dd{i});
        day1=cc(1,:);%直达车的日期
        day1=str2num(day1);
        st1=cc(2,:);%直达车到达时间
        endt1=cc(3,:);%直达车离开时间
        fprintf('直达车编号为%d 的终点时间\n',zhida_id(i));
        day2=ee(1,:);%直达车的日期
        day2=str2num(day2);
        st2=ee(2,:);%直达车到达时间
        endt2=ee(3,:);%直达车离开时间
        if day1>day2
            fprintf('此直达车日期错误!\n');
        elseif day1==day2
            p(i)=Time_judge(endt1,st2);
            if p(i)>0
                fprintf('乘车时间为:%d 分钟\n',p(i));
            else
                fprintf('此直达车日期错误\n');
            end
        end
    end
end

```

```

        else
            tt=day2-day1;
            p(i)=Time_judge(endt1,st2);
            p(i)=p(i)+tt*1440;
            fprintf('乘车时间为:%d 分钟\n',p(i));
        end

    end

    %checinum=zhida_id(find(p>0))%可以直达车的编号
    ttt=p(find(p>0));%可以直达车的时间
    if isempty(ttt)
        fprintf('日期错误!\n');
        ttt1=0;
        min_checi=0;
    else

        min_checi=zhida_id(find(p==min(ttt)));%最小时间的直达车次

        min_checi=min_checi(1);%最小时间的直达车次中之选出一辆车
        ttt1=min(ttt);%最小时间
    end
end

end

end

```

```

function []=Strat_End1(A)%转一次车
A1=A(:,3);
dis=str2num(char(A(:,7)));
[b1,m1,n1]=unique(A1);
    n1;%城市编号
    A2=A(:,1);
[b2,m2,n2]=unique(A2);
n2;%车次编号
i1=input('请输入起点城市编号!\n')
i2=input('请输入终点城市编号!\n')
i11=n2(find(n1==i1));%起点车次编号
i22=n2(find(n1==i2));%终点车次编号
ii0=setdiff(i11,i22);%只经过起点站车次的编号
iie=setdiff(i22,i11);%只经过终点站
aa={ };
id0=find(n1==i1);
for i=1:size(ii0)
    if n1(id0(i))==i1 & n2(id0(i))==ii0(i)
        a(i,1)=id0(i);
    end
end
end
a;%起点城市的编号索引

```

```

ide=find(n1==i2);
for i=1:size(iie)
    if n1(ide(i))==i2 & n2(ide(i))==iie(i)
        b(i,1)=ide(i);
    end
end
b;%终点城市的编号索引

t0=A(a,4:6);
day1=t0(:,1);
jinzhan1=t0(:,2);%进站时间
chuzhan1=t0(:,3);%出站时间

te=A(b,4:6);
day2=te(:,1);
jinzhan1=te(:,2);%进站时间
chuzhan2=te(:,3);%出站时间

ci02={ };
for i=1:size(t0,1)
    ci0=[];
    ci0(:,i)=n1(find(n2==ii0(i)));%车次编号为 ii0(2)的所有城市
    qiyuchengshi=A(find(n2==ii0(i)),4:6);%车次编号为 ii0(2)的所有时间
    i1chengshi=t0(i,:);%第二个起点站的时间
    qiyuchengshi1=char(qiyuchengshi);
    qiyuday=qiyuchengshi1(1:size(qiyuchengshi1,1)/3,:);

    qiyudaoda=qiyuchengshi1(size(qiyuchengshi1,1)/3+1:2*size(qiyuchengshi1,1)/3,:);

    qiyuchu=qiyuchengshi1(2*size(qiyuchengshi1,1)/3+1:3*size(qiyuchengshi1,1)/3,:);
    i1changshi1=char(i1chengshi);
    i1day=i1changshi1(1,:);
    i1daoda=i1changshi1(2,:);
    i1chu=i1changshi1(3,:);
    i1num=str2num(i1day);
    qiyunum=str2num(qiyuday);

    ci01=zeros(size(qiyunum,1),i);

    for k=1:size(qiyunum)
        if i1num==qiyunum(k)
            p=Time_judge(i1daoda,qiyudaoda(k,:));
            if p>0

                ci01(k,i)=ci0(k,i);
            end
        end
    end
end

```

```

elseif i1num<qiyunum(k)
    ci01(k,i)=ci0(k,i);
else
    ci01(k,i)=0;
end
end

ci01(:,i);
ci02{i}=ci01(:,i);
end
%可能的中转站
ci02;%记下起点中对应每一个车次对应城市的所有编号

aaa={};
for i=1:size(ci02,2)
    aa=[];
    aa=ci02{i};
    aaa{i}=aa(find(aa~=0));
end
aaa;
for i=1:size(aaa,2)
    aaa{i};
end

ci03={};
for i=1:size(te,1)
    ci0e=[];
    ci0e(:,i)=n1(find(n2==iie(i)));%车次编号为 ii0(2)的所有城市
    qiyuchengshi=A(find(n2==iie(i)),4:6);%车次编号为 ii0(2)的所有时间
    i1chengshi=te(i,:);%第二个起点站的时间
    qiyuchengshi1=char(qiyuchengshi);
    qiyuday=qiyuchengshi1(1:size(qiyuchengshi1,1)/3,:);

    qiyudaoda=qiyuchengshi1(size(qiyuchengshi1,1)/3+1:2*size(qiyuchengshi1,1)/3,:);

    qiyuchu=qiyuchengshi1(2*size(qiyuchengshi1,1)/3+1:3*size(qiyuchengshi1,1)/3,:);
    i1changshi1=char(i1chengshi);
    i1day=i1changshi1(1,:);
    i1daoda=i1changshi1(2,:);
    i1chu=i1changshi1(3,:);
    i1num=str2num(i1day);
    qiyunum=str2num(qiyuday);

    ci011=zeros(size(qiyunum,1),i);

    for k=1:size(qiyunum)
        if i1num==qiyunum(k)

```

```

        p=Time_judge(i1 daoda,qiyudaoda(k,:));
        if p<0

            ci011(k,i)=ci0e(k,i);
            end
        elseif i1 num>qiyunum(k)
            ci011(k,i)=ci0e(k,i);
        else
            ci011(k,i)=0;
        end
    end
    end

    ci011(:,i);
    ci03{i}=ci011(:,i);
end

%可能的中转站
ci03;%记下终点中对应每一个车次对应城市的所有编号
bbb={ };
for i=1:size(ci03,2)
    bb=[];
    bb=ci03{i};
    bbb{i}=bb(find(bb~=0));
end
bbb;
for i=1:size(bbb,2)
    bbb{i};
end
size(aaa,2);
aa1=zeros(size(aaa,2),size(aaa,2));
for i=1:size(aaa,2)
    mm=size(aaa{i});
    aa1(1:mm,i)=aaa{i};
end
aa2=aa1(:);
aa2=aa2(find(aa2~=0));
bb1=zeros(size(bbb,2),size(bbb,2));
for i=1:size(bbb,2)
    nn=size(bbb{i});
    bb1(1:nn,i)=bbb{i};
end
bb2=bb1(:);
bb2=bb2(find(bb2~=0));

for i=1:size(aa2)
    for j=1:size(bb2)

        if aa2(i)==bb2(j)
            dd2(i)=aa2(i);

```

```
        end
    end
end
```

```
    fprintf('中转站的城市编号!')
    dd2=dd2(find(dd2~=0));
    dd3=unique(dd2)
```

```
%disp('起点到中转站的时间及车次编号!')
```

```
    for i=1:size(dd3,2)
        [ttt1(i),min_checi1(i)]=Start_End(A,i1,dd3(i));
    end
    ttt1;%起点到中转站的时间
    %disp('起点到中转站的车次编号!')
    min_checi1;%起点到中转站对应最短时间的车次
```

```
%disp('中转站到终点的时间及车次编号!')
    for i=1:size(dd3,2)
        [ttt2(i),min_checi2(i)]=Start_End(A,dd3(i),i2);
    end
    ttt2;%中转站到终点的时间
```

```
%解析城市
    for i=1:size(dd3,2)
        name1=find(n1==dd3(i));
        name2=find(n2==min_checi1(i));
        namenum=intersect(name1,name2);
        name11(i,:)=A1(namenum,:);
        %name{i}=char(name(i));
    end
```

```
    disp('中转站城市!')
    name11=char(name11)
```

```
%disp('起点到中转站的车次编号!')
    min_checi1;%起点到中转站对应最短时间的车次
    for i=1:size(dd3,2)
        kind1(i)=unique(A(find(n2==min_checi1(i)),1));
    end
    disp('起点到中转站的车次类型!')
    kind1=char(kind1)
```

```
%disp('中转站到终点的车次编号!')
    min_checi2;%中转站到终点的对应最短时间的车次
```

```

for i=1:size(dd3,2)
    kind2(i)=unique(A(find(n2==min_checi2(i)),1));
end
disp('中转站到终点的车次类型!')
kind2=char(kind2)

%换乘间隔
for i=1:size(dd3,2)
    x11=find(n1==dd3(i));
    x12=find(n2==min_checi1(i));
    xj1=intersect(x11,x12);
    %daodi=A(xj1,4:6);
    daodi=char(A(xj1,4:6));
    daodi1=daodi(1:2,:);%到达中转站的时间

    dis1=dis(xj1);%到达中转站的距离大小

    x0=find(n1==i1);
    x01=find(n2==min_checi1(i));
    x0j=intersect(x0,x01);
    dis0=dis(x0j);%起点的距离的大小

    y11=find(n1==dd3(i));
    y12=find(n2==min_checi2(i));
    yj1=intersect(y11,y12);%中转站到终点的编号
    zhongdi=char(A(yj1,4:6));
    zhongdi1=zhongdi(1:2:3,:);%离开中转站的时间

    z11=find(n1==dd3(i));
    z12=find(n2==min_checi2(i));
    zj1=intersect(z11,z12);
    dis12=dis(zj1);%离开中转站的距离大小

    y111=find(n1==i2);
    y112=find(n2==min_checi2(i));
    yj11=intersect(y111,y112);
    dis2=dis(yj11);%终点距离的大小
    ppt=Time_judge(daodi1(2,:),zhongdi1(2,:));
    if ppt>0
        huanshi(i)=ppt;
    else
        huanshi(i)=1440-abs(ppt);
    end
    zjuli1(i)=(dis2-dis12);
    zjuli2(i)=(dis1-dis0);

end
end

```

```
%disp('换乘间隔')
huanshi;%换乘间隔
```

```
%disp('起点到中转站的距离')
zjuli1;%起点到中转站的距离
%disp('中转站到终点的距离')
zjuli2;%中转站到终点的距离
```

```
%转战的总时间
for i=1:length(ttt1)
    tttz(i)=ttt1(i)+ttt2(i);
end
%disp('起点到终点的总时间')
tttz;
```

```
%转战的总费用
for i=1:size(dd3,2)
    kk1=kind1(i,:);
    if kk1(1)=='K' | kk1(1)=='T'
        f1(i)=zjuli1(i)*0.135;
    elseif kk1(1)=='G'
        f1(i)=zjuli1(i)*0.360;
    elseif kk1(1)=='D'
        f1(i)=zjuli1(i)*0.345;
    else
        f1(i)=zjuli1(i)*0.1172;
    end
end
%disp('起点到中转站的费用')
f1;
```

```
for i=1:size(dd3,2)
    kk2=kind2(i,:);
    if kk2(1)=='K' | kk2(1)=='T'
        f2(i)=zjuli2(i)*0.135;
    elseif kk2(1)=='G'
        f2(i)=zjuli2(i)*0.359;
    elseif kk2(1)=='D'
        f2(i)=zjuli2(i)*0.345;
    else
        f2(i)=zjuli2(i)*0.1172;
    end
end
%disp('中转站到终点的费用')
f2;
```



```

for i=1:size(dd3,2)
    f3(i)=f1(i)+f2(i);
end
%disp('转战的总费用')
f3;

pp=Youhua(f3,ttt,huanshi);

```

```

%最优路线
disp('最优中转站')
yzhan=name11(pp,:)%最优中转站
disp('起点到中转站最优车次')
zyche1=kind1(pp,:);
disp('中转站到终点的最优车次')
zyche2=kind2(pp,:);

end

```

```

function []=Start_End2(A)
A1=A(:,3);
[b1,m1,n1]=unique(A1);
n1;%城市编号
A2=A(:,1);
[b2,m2,n2]=unique(A2);
n2;%车次编号
i1=input('请输入起点城市编号!\n')
i2=input('请输入终点城市编号!\n')
i11=n2(find(n1==i1));%起点车次编号
i22=n2(find(n1==i2));%终点车次编号
ii0=setdiff(i11,i22);%只经过起点站车次的编号
iie=setdiff(i22,i11);%只经过终点站
iii=union(ii0,iie);
iiz=setdiff(n2,iii);%不经过起点站与终点站的所有列车的编号
size(iiz);
%n1(find(n2==iiz(3)));
zchengshi1={ };
for i=1:size(iiz)
    iiii=n1(find(n2==iiz(i)));
    zchengshi1{i}=iiii;
end
zchengshi1;%不经过起点站与终点站的所有城市的编号
size(zchengshi1);

%

```

```

aa={ };
id0=find(n1==i1);
for i=1:size(ii0)
    if n1(id0(i))==i1 & n2(id0(i))==ii0(i)
        a(i,1)=id0(i);
    end
end
a;%起点城市的编号索引

ide=find(n1==i2);
for i=1:size(iie)
    if n1(ide(i))==i2 & n2(ide(i))==iie(i)
        b(i,1)=ide(i);
    end
end
b;%终点城市的编号索引

%

t0=A(a,4:6);
day1=t0(:,1);
jinzhan1=t0(:,2);%进站时间
chuzhan1=t0(:,3);%出战时间

te=A(b,4:6);
day2=te(:,1);
jinzhan1=te(:,2);%进站时间
chuzhan2=te(:,3);%出战时间

%起点之后的城市与终点之前的城市

ci02={ };
for i=1:size(t0,1)
    ci0=[];
    ci0(:,i)=n1(find(n2==ii0(i)));%车次编号为 ii0(2)的所有城市
    qiyuchengshi=A(find(n2==ii0(i)),4:6);%车次编号为 ii0(2)的所有时间
    i1chengshi=t0(i,:);%第二个起点站的时间
    qiyuchengshi1=char(qiyuchengshi);
    qiyuday=qiyuchengshi1(1:size(qiyuchengshi1,1)/3,:);

    qiyudaoda=qiyuchengshi1(size(qiyuchengshi1,1)/3+1:2*size(qiyuchengshi1,1)/3,:);

    qiyuchu=qiyuchengshi1(2*size(qiyuchengshi1,1)/3+1:3*size(qiyuchengshi1,1)/3,:);
    i1changshi1=char(i1chengshi);
    i1day=i1changshi1(1,:);
    i1daoda=i1changshi1(2,:);
    i1chu=i1changshi1(3,:);
    i1num=str2num(i1day);

```

```

    qiyunum=str2num(qiyuday);

    ci01=zeros(size(qiyunum,1),i);

    for k=1:size(qiyunum)
        if i1num==qiyunum(k)
            p=Time_judge(i1 daoda,qiyudaoda(k,:));
            if p>0

                ci01(k,i)=ci0(k,i);
            end
        elseif i1num<qiyunum(k)
            ci01(k,i)=ci0(k,i);
        else
            ci01(k,i)=0;
        end
    end

    ci01(:,i);
    ci02{i}=ci01(:,i);
end
%可能的中转站
ci02;%记下起点中对应每一个车次对应城市的所有编号

aaa={ };
for i=1:size(ci02,2)
    aa=[];
    aa=ci02{i};
    aaa{i}=aa(find(aa~=0));
end
aaa;
for i=1:size(aaa,2)
    aaa{i};
end

ci03={ };
for i=1:size(te,1)
    ci0e=[];
    ci0e(:,i)=n1(find(n2==iie(i)));%车次编号为 ii0(2)的所有城市
    qiyuchengshi=A(find(n2==iie(i)),4:6);%车次编号为 ii0(2)的所有时间
    i1chengshi=te(i,:);%第二个起点站的时间
    qiyuchengshi1=char(qiyuchengshi);
    qiyuday=qiyuchengshi1(1:size(qiyuchengshi1,1)/3,:);

    qiyudaoda=qiyuchengshi1(size(qiyuchengshi1,1)/3+1:2*size(qiyuchengshi1,1)/3,:);

    qiyuchu=qiyuchengshi1(2*size(qiyuchengshi1,1)/3+1:3*size(qiyuchengshi1,1)/3,:);

```

```

    i1changshi1=char( i1chengshi);
    i1day=i1changshi1(1,:);
    i1daoda=i1changshi1(2,:);
    i1chu=i1changshi1(3,:);
    i1num=str2num(i1day);
    qiyunum=str2num(qiyuday);

    ci011=zeros(size(qiyunum,1),i);

    for k=1:size(qiyunum)
        if i1num==qiyunum(k)
            p=Time_judge(i1daoda,qiyudaoda(k,:));
            if p<0

                ci011(k,i)=ci0e(k,i);
            end
            elseif i1num>qiyunum(k)
                ci011(k,i)=ci0e(k,i);
            else
                ci011(k,i)=0;
            end
        end
    end

    ci011(:,i);
    ci03{i}=ci011(:,i);
end
%可能的中转站
ci03;%记下终点中对应每一个车次对应城市的所有编号
bbb={};
for i=1:size(ci03,2)
    bb=[];
    bb=ci03{i};
    bbb{i}=bb(find(bb~=0));
end
bbb;
for i=1:size(bbb,2)
    bbb{i};
end
size(aaa,2);
aa1=zeros(size(aaa,2),size(aaa,2));
for i=1:size(aaa,2)
    mm=size(aaa{i});
    aa1(1:mm,i)=aaa{i};
end
aa2=aa1(:);
aa2=aa2(find(aa2~=0));

    bb1=zeros(size(bbb,2),size(bbb,2));
    for i=1:size(bbb,2)

```

```

        nn=size(bbb{i});
        bb1(1:nn,i)=bbb{i};
    end
    bb2=bb1(:);
    bb2=bb2(find(bb2~=0));
    aa2;
    bb2;
    %aa2 记下的是起点之后的车次对应
    %bb2 对应的是终点之前的车次对应的城市的编号

```

```

j1={};j11=zeros(size(zchengshi1,2),size(zchengshi1,2));
for i=1:size(zchengshi1,2)
    zchengshi1{i};
    jj=intersect(aa2,zchengshi1{i});
    j1{i}=jj;
    dim1=size(j1{i});
    j11(1:dim1,i)=j1{i};
end

```

```

j11;
j11=j11(:);
j11=j11(find(j11~=0));
j11=unique(j11);%可能第一次中转站城市编号
%x1=find(j11==1667)

```

```

j2={};j22=zeros(size(zchengshi1,2),size(zchengshi1,2));
for i=1:size(zchengshi1,2)
    jjj=intersect(bb2,zchengshi1{i});
    j2{i}=jjj;
    dim2=size(j2{i});
    j22(1:dim2,i)=j2{i};
end

```

```

j22;
j22=j22(:);
j22=j22(find(j22~=0));
j22=unique(j22);%可能第二次中转站城市编号
size(j22);
%x2=find(j22==1622)
%丹东到宜昌东第一次转战城市可能有 168 个
%丹东到宜昌东第二次转战城市可能有 197 个

```

```

% x1=find(j11==1667)
% x2=find(j22==1622)
% x11=n2(find(n1==j11(x1)));
% x22=n2(find(n1==j22(x2)));
% zhida_id1=intersect(x11,x22)

```

%转战 1 到转战 2 的筛选（1）

```

j112=zeros(size(j11));
for i=1:size(j11)
    x1=n2(find(n1==j11(i)));
    x22={ };
    for j=1:size(j22)
        x2=n2(find(n1==j22(j)));
        x22{j}=x2;
    end
    zhida_id1={ };
    for k=1:size(x22,2)
        zhida_id1{k}=intersect(x1,x22{k});
        if ~isempty(zhida_id1{k})
            j112(i)=j11(i);
            break;
        end
    end
end

j112;
j113=j112(find(j112~=0))%新的转战 1 的城市编号
size(j113);

%转战 1 到转战 2 的筛选 (2)
j116=zeros(size(j22));
for i=1:size(j22)
    q1=n2(find(n1==j22(i)));
    q11={ };
    for j=1:size(j113)
        q2=n2(find(n1==j113(j)));
        q11{j}=q2;
    end
    zhida_id2={ };
    for k=1:size(q11,2)
        zhida_id2{k}=intersect(q1,q11{k});
        if ~isempty(zhida_id2{k})
            j116(i)=j22(i);
            break;
        end
    end
end

j116=j116(find(j116~=0))
size(j116);%新的转战 2 的城市编号

%判断两个中转站能否直达
% k1={ };
% for i=1:size(j113)
%     kk=[];
%     for j=1:size(j116)
%         kk(j)=Iszhida(A,j113(i),j116(j))

```

```

%
%     end
%     k1{i}=kk;
% end
% k1
% size(k1)

%起点、转战地 1，转战地 2，终点共四个城市筛选

%起点到转战 1 的筛选
% j114=zeros(size(j113));
% for i=1:size(j113)
%     y1=n2(find(n1==j113(i)));
%     y2=n2(find(n1==i1));
%     z1=intersect(y2,y1);
%     if ~isempty(z1)
%         j114(i)=j113(i);
%     end
% end
% j114;
% j114=j114(find(j114~=0));%无效筛选
% size(j114);
%
%转战 2 到终点的筛选
% j115=zeros(size(j22));
% for i=1:size(j22)
%     y11=n2(find(n1==j22(i)));
%     y22=n2(find(n1==i2));
%     z2=intersect(y11,y22);
%     if ~isempty(z2)
%         j115(i)=j22(i);
%     end
% end
% j115=j115(find(j115~=0));%无效筛选
% size(j115);

%计算时间
% for i=1:size(j113)
%     for j=1:size(j116)
%         [ttt1,min_checi]=Start_End(A,j113(i),j116(j));
%     end
% end
% ttt1
% min_checi
% size(ttt1)
% size(min_checi)

```

end

function [pp]=Youhua(f,ct,ht)% 费用,乘车时间,换车间隔

for i=1:size(ht,2)

if ht(i)<180 & ht(i)>=20

r(i)=1+(ht(i)-20)/(180-20);

elseif ht(i)>=180 & ht(i)<300

r(i)=1;

elseif ht(i)>=300 & ht(i)<1460

r(i)=1+(ht(i)-300)/(1460-300);

else

r(i)=2;

end

end

r; %换车间隔的评价函数

X=[f',ct',r']

max1=max(X);

for i=1:size(X,1)

for j=1:size(X,2)

X1(i,j)=X(i,j)/max1(j);

end

end

X1;

s=sum(X1,1);

k=1/log(size(ht,2));

X11=[];X22=[];

for i=1:size(X1,1)

for j=1:size(X1,2)

X11(i,j)=X1(i,j)/s(j);

end

end

X11;

for i=1:size(X11,1)

for j=1:size(X11,2)

X22(i,j)=X11(i,j)\*log(X11(i,j));

end

end

X22;

h=sum(X22,1).\*(-k);

for i=1:length(h)

d(i)=(1-h(i))/(size(X,2)-sum(h));



```

end
d

for i=1:size(X11,1)

    for j=1:size(X11,2)
        y(j)=X11(i,j)*d(j);
    end
    u(i)=sum(y);
end
u
pp=find(u==min(u));

end

```

## 附录 2

问题二：

```

function []=Wenti2(A)
dis=str2num(char(A(:,7)));
A1=A(:,3);
[b1,m1,n1]=unique(A1);
n1;%城市编号
A2=A(:,1);
[b2,m2,n2]=unique(A2);
n2;%车次编号
i1=[933 63 485 1446 2242 1336];
i2=[933 63 485 1446 2242 1336];
t1t1={};min_checi1={};
for i=1:size(i1,2)
    for j=1:size(i2,2)
        % if i~=j
            [t1t1(j),min_checi1(j)]=Start_End(A,i1(i),i2(j));
        % end
    end
    t1t1{i}=t1t1;
    min_checi1{i}=min_checi1;
end
t1t1{2};%时间

size(t1t1);
min_checi1{1};
size(min_checi1);

%一个起点城市分别到六个终点城市的车次类型
for i=1:size(i1,2)
    aa=min_checi1{i};
    for j=1:size(i2,2)

```

```

        kind1=unique(A(find(n2==aa(j)),1))
    end
    kind1;
    kind1=char(kind1);
    kind11{i}=kind1;
    end
    %disp('起点到终点的车次类型!')
    kind11{1};
    for i=1:size(i1,2)
        bb=min_checi1{i};
        for j=1:size(i2,2)

            y11=find(n1==i2(j));
            y112=find(n1==i1(i));
            if bb(j)==0
                disj(j)=0;
            else
                y22=find(n2==bb(j));
                yj1=intersect(y11,y22);%终点距离编号
                yj2=intersect(y112,y22);%起点距离编号

                disj(j)=dis(yj1)-dis(yj2);
            end
        end
        disj1{i}=disj;
    end
    disj1;

    %起点到终点的费用
    for i=1:size(i1,2)
        zjuli=disj1{i};
        for j=1:size(i2,2)
            kk1=kind11{i};
            if j>size(kk1,1)
                kk2=0;
            else
                kk2=kk1(j,:);
            end
            if kk2(1)=='K' | kk2(1)=='T'
                f(j)=zjuli(j)*0.135;
            elseif kk2(1)=='G'
                f(j)=zjuli(j)*0.459;
            elseif kk2(1)=='D'
                f(j)=zjuli(j)*0.445;

            else
                f(j)=zjuli(j)*0.1172;
            end
        end
    end
end

```

```

f1{i}=f;
end
f1{2};% 费用

for i=1:size(i1,2)
    for j=1:size(i2,2)
        [u_max(i,j),pp(i,j)]=Youhua2(f1{i},ttt1{j});
    end
end
u_max
pp

end

```

```

function [u_max,pp]=Youhua2(f,ct)% 费用,乘车时间,换车间隔
X=[f',ct'];

```

```

max1=max(X);
for i=1:size(X,1)
    for j=1:size(X,2)
        if X(i,j)==0;
            X1(i,j)=0.001;%mean1(j);
        else
            X1(i,j)=X(i,j)/max1(j);
        end
    end
end
X1;

```

```

s=sum(X1,1);

```

```

k=1/log(size(ct,2));
X11=[];X22=[];
for i=1:size(X1,1)
    for j=1:size(X1,2)
        X11(i,j)=X1(i,j)/s(j);
    end
end
X11;

```

```

for i=1:size(X11,1)
    for j=1:size(X11,2)
        X22(i,j)=X11(i,j)*log(X11(i,j));
    end
end
X22;

```

```

h=sum(X22,1).*(-k);
for i=1:length(h)
    d(i)=(1-h(i))/(size(X,2)-sum(h));
end
d

```

```

for i=1:size(X11,1)

    for j=1:size(X11,2)
        y(j)=X11(i,j)*d(j);
    end
    u(i)=sum(y);
end
u;

```

```

u_max=min(u);

```

```

pp=find(u==min(u));

```

```

end

```

```

function p=Time_judge(y1,y2)

```

```

t1=datevec(y1);

```

```

t2=datevec(y2);

```

```

t1=datenum(t1);

```

```

t2=datenum(t2);

```

```

if t2>t1

```

```

    t = t2-t1;

```

```

    t = datevec(t);

```

```

    t = t(4:end);

```

```

    p = t(1)*60+t(2);

```

```

elseif t1==t2

```

```

    p = 0;

```

```

    pp = -1;

```

```

else

```

```

    t=t2-t1;

```

```

    t=datevec(t);

```

```

    t=t(4:end);

```

```

    p = t(1)*60+t(2)-24*60;

```

```

end

```

```

p;

```

```

end

```

LINGO 程序:

```

model:

```

```

!0-1 整数规划;

```

```

sets:
!6 个出发地;
Start_area/1..6/:s;
!六个目的地城市;
End_area/1..6/:e;
!权值;
link(Start_area,End_area):w,x;
endsets

!目标函数;
min=@sum(link(i,j):x(i,j)*w(i,j));

!约束条件;
@for(End_area(j):@sum(Start_area(i):x(i,j))=1);
@for(Start_area(i):@sum(End_area(j):x(i,j))=1);
@for(link(i,j):x(i,j)+x(j,i)<=1);
@for(link(i,j):@bin(x(i,j)));

data:
w=0.0003 0.0345 0.0605 0.0520 0.0395 0.0392
    0.0333 0.0008 0.0127 0.0199 0.0053 0.0152
    0.0451 0.0308 0.0312 0.0540 0.0174 0.0214
    0.0535 0.0159 0.0312 0.0008 0.0221 0.0270
    0.0208 0.0094 0.0092 0.0242 0.0084 0.0061
    0.0324 0.0205 0.0102 0.0394 0.0044 0.0006;

enddata

```