

# 铁路最优路线

## 摘要

本文解决的是铁路最优路线的设计问题。在设计两地通行和循环旅游路线的问题上,综合考虑旅客的换乘次数、乘车时间和乘车费用等因素,分别建立了两个多目标最优化模型。

对于问题一,建立了多目标最优化模型。建立的5个目标分别为乘车和换乘时间最短、乘车费用最低、换乘次数最少以及换乘距离最短。首先把站点简化成了网络节点,通过构建乘车网络来分析此题;然后采用信息熵法确定了各研究目标的权重得到了各可行路径(依次为6条、12条、4条)的综合目标值,利用综合目标值对可行路径排序(最优路径的综合目标值依次为0.0797、0.0310、0.1639),得到了较满意的出行方案;最后根据近期的铁路旅客列车时刻表,计算了丹东→宜昌、天津→拉萨、白城→青岛的最优路线如下表:

起点站	终点站	换乘次数	中转站	时间	费用
丹东	宜昌东	1	常州	47 时 9 分	404
天津	拉萨	1	郑州	51 时 33 分	386
白城	青岛	1	长春	28 时 58 分	222

对于问题二,在第一问模型的基础上,继续采用多目标模型。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,循环路线可以利用多目标最优化模型来求解,通过 MATLAB 编程,得到旅游的最优路径:宜昌东——>南京南——>无锡——>苏州——>上海——>杭州——>宜昌东。在此次最优化的旅游路线的指导下,总共花费 863 元,所用的时间为 8 小时 46 分钟。最终得到最佳具体旅游路线如下:

旅游最优路线						
路线	宜昌东-南京南	南京南-无锡	无锡-苏州	苏州-上海	上海-杭州	杭州-宜昌
车次	D3008 D3005	D3008 D3005	D3008 D3005	D3008 D3005	G7505	G586
费用	304 元	68 元	16 元	32 元	73 元	370 元
用时	5 时 57 分	1 时 14 分	17 分钟	33 分钟	45 分钟	7 时 3 分

关键字: 图论 多目标规划 信息熵

## 一、问题重述

### 1.1 问题背景

随着六次大提速和客运专线的大量开通,旅客列车开行的数量和种类越来越多,这些硬件设施的加强使得越来越多的旅客选择铁路出行,这就要求铁路部门在加强硬件建设的同时提高服务能力,其中,针对旅客出行要求提供可选择的乘车方案就是一个值得研究的问题。

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工具。

### 1.2 需要解决的问题

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) 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。

### 1.3 题目所给信息

附录 1 给出 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。

## 二、模型的假设与符号说明

### 2.1 模型的假设

假设 1: 假设问题一中换乘时间适中, 即不存在错过列车和跨天的情况。

假设 2: 假设问题一中只考虑直达列车票已售罄的情况。

假设 3: 假设换乘站间的距离为零。

## 2.2 模型的符号说明

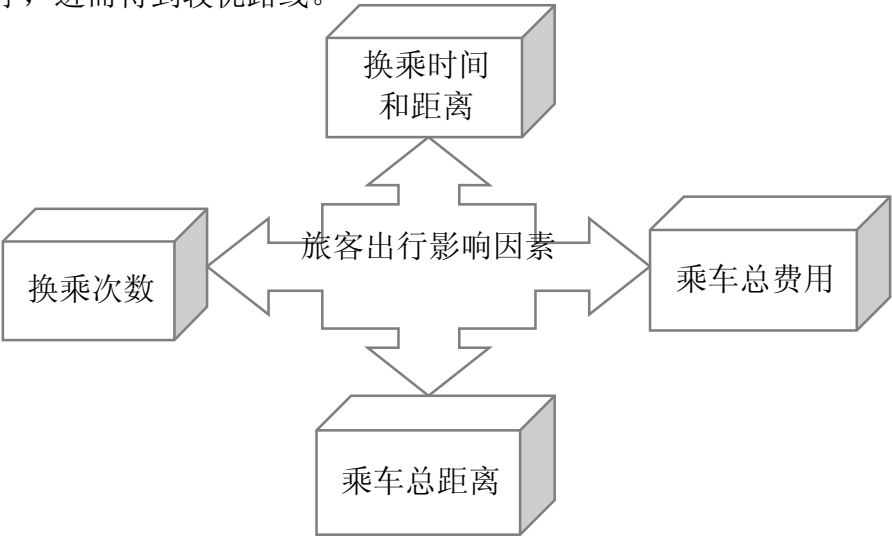
符号	说明
$G$	有向图
$L$	列车集合
$V$	列车经过的站点集合
$V_{l_i}$	列车 $l_i$ 经过的站点集合
$T_{l_i}$	列车 $l_i$ 经过站点的时间集合
$A$	列车 $l_i$ 的类型
$P$	列车的票价集合
$V'$	换乘站点对集合
$D$	换乘站间的距离集合
$t_{ij}(l_i)$	乘 $l_i$ 次列车从 $i$ 站到 $j$ 站的乘车时间
$N_{\max}$	最大换乘次数
$d(v_{l_k}^i, v_{l_m}^j)$	两个车站间的换乘距离
$t_{ij}(l_i, l_i')$	从 $i$ 站到 $j$ 站换乘 $l_i'$ 次需换乘间隔
$p_{ij}(l_i)$	$l_i$ 次列车从 $i$ 站到 $j$ 站的乘车费
$x_{ij}^{l_i}$	从 $i$ 站到 $j$ 站是否乘坐了 $l_i$ 次列车
$M$	可行路径条数

## 三、问题分析

本文主要研究的是铁路最优路线的设计问题, 首先考虑一个站点到另一个站点的最优线路, 再解决巡回路线旅游的问题, 实际上研究的是图论中的组合优化, 把每个景点看作图中的一个顶点, 景点之间到达的路线可看作对应顶点间的边, 费用、时间、路程等作为边的权值, 形成了旅行问题的加权网络图。

3.1 问题一的分析

问题一需要求出以下起点到终点的最优路线，即：丹东→宜昌、天津→拉萨、白城→青岛。首先根据题目条件设出两个站点之间的距离、车次始发和终到的时间差、换乘时间和距离，其中换乘次数是旅客最直观的感受，不能太多，则设出换乘次数（不大于 2 次）；然后建立多目标函数进行求解，并采用信息熵法求出各研究目标的权重得到各可行路径的综合目标值，利用综合目标值对可行路径进行排序，进而得到较优路线。



3.2 问题二的分析

问题二解决的是旅游循环路线的选择问题。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，可以在问题一的基础上进行分步求解，然而与问题一不同的是，这里不仅仅是考虑到达两地的路线问题，并且要考虑到旅游心理，即在时间有限的情况下，人们对于花费主观因素已经降低，人们更愿意花较多的钱得到高品质的服务，于是在求得由宜昌东到上海的最佳路线后，可以直接考虑坐高铁回到宜昌。

四、数据分析与处理

对数据进行如下处理：  
首先，对附件 1 的火车站站台进行处理编号，经过 MATLAB 编程（见附录一），得到 2876 个站台（此处指附录部分数据）：

表 1：站点编号表

西安	1	塘沽	55	巴彦高勒	59	...	丰顺	2809	东方	2863
咸阳	2	济南	56	临河	60	...	揭阳	2810	信宜	2864
杨陵镇	3	泰山	57	五原	61	...	潮州	2811	高州	2865

⋮	⋮	⋮	⋮	⋮	⋮	...	⋮	⋮	⋮	⋮
开原	52	胶州北	106	月山	110	...	八达岭	2860		
盘锦北	53	乌海西	107	济源	111	...	延庆	2861		
葫芦岛北	54	乌海	108	运城	112	...	九龙	2862		

然后对附件 1 的时间进行处理，经过 Excel 编辑的函数，得到在同一车次列车运行的区间内，相邻两个车站火车运行的时间差。

## 五、问题一的求解

### 5.1 问题一准备工作

作为最普遍的交通出行方式，铁路出行一直是人们的首选，在选择乘车路线时，人们往往考虑多种因素的影响，为了构建最优化的铁路乘车路线，就势必得了解乘客出行时更看中哪些重要因素，通过网上数据查找，整理得到下图：

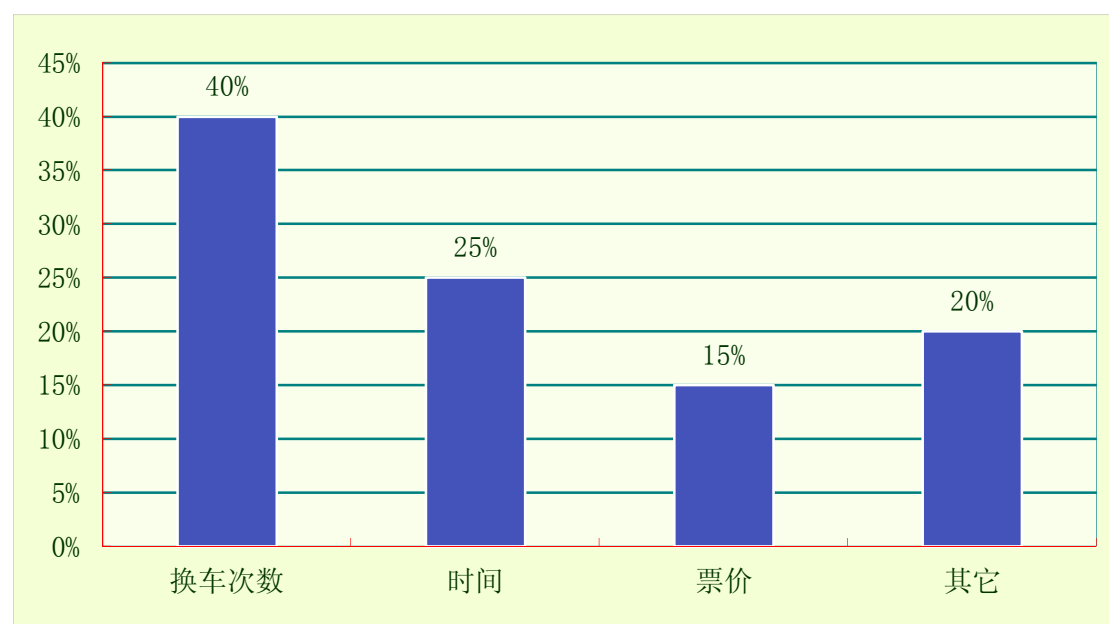


图 1：旅游影响因素对比图

在简化的路网基础上，以两站间通过的列车作为有向弧，两站间列车运行时间作为弧的权，以列车的到发时刻作为节点的权值，构建乘车网络。

构建的有向图： $G=(V,L)$

1) 列车集合： $L=\{l_i|i=1,2,\cdots,N\}$ （其中 N 为列车数量）

2) 列车经过的站点集合： $V=\{V_{l_i}|l_i\in L,i=1,2,\cdots,N\}$

1. 列车  $l_i$  经过的站点集合： $V_{l_i}=\{v_{l_i}^1,v_{l_i}^2,\cdots,v_{l_i}^m\}$

2. 列车  $l_i$  经过站点的时间集合： $T_{l_i}=\{(t_{l_i}^1(u),t_{l_i}^2(u))|u\in V_{l_i}\},l_i\in L$ （其中

$t_l^1(u), t_l^2(u)$  分别为列车到达和离开  $u$  站的时刻。)

3. 列车  $l_i$  的类型:  $A = \{a_{l_i} | l_i \in L\}$

3) 列车的票价集合:  $P = \{p_{ij}(l_i) | i, j \in V_{l_i}, l_i \in L\}$  (其中为乘  $l_i$  次列车从  $i$  站到  $j$  站的票价。)

4) 换乘站点对集合:  $V' = \{(v_{l_k}^i, v_{l_m}^j) | v_{l_k}^i, v_{l_m}^j \in V\}$

5) 换乘站间的距离集合:  $D = \{d(v_{l_k}^i, v_{l_m}^j) | v_{l_k}^i, v_{l_m}^j \in V\}$  (其中  $d(v_{l_k}^i, v_{l_m}^j)$  表示  $l_k, l_m$  在站点的换乘距离, 由假设 3 令  $D = 0$ 。)

## 5.2 模型的建立

旅客出行路径的主要影响因素包括换乘次数和距离, 乘车时间和票价费用等。

1) 乘车时间: 所选线路、始发终到站里程和所选列车种类都是乘车时间的影响因素, 这里用  $t_{ij}(l_i)$  表示乘  $l_i$  次列车从  $i$  站到  $j$  站的乘车时间,

$$\text{即: } t_{ij}(l_i) = t_{l_i}^1(j) - t_{l_i}^2(i)$$

2) 换乘次数: 旅客在一次出行中可能要面临着换乘不同列车的问题, 换乘次数太多必然会影响旅客的乘车心情, 换乘次数越少的路径相对被选择的概率越大<sup>[1]</sup>。设最大换乘次数  $N_{\max}$ , 一般  $N \leq 2$  是比较合理的,

在此题的求解中为简化难度, 将换乘次数规定为 1 次。

3) 换乘距离: 考虑到换乘可能会在不同的车站, 这直接关系到换乘的方便与否, 则两个车站间的换乘距离为:  $d(v_{l_k}^i, v_{l_m}^j)$ , 本题中不考虑站点之间的距离差, 则距离为零。

4) 换乘时间间隔: 换乘时间间隔为一列车的到达时刻与另一辆车出发时刻之间的时间差, 假设 2 中已提出换乘时间间隔合理的假设, 对合理的时间间隔提出一个紧接续标准  $T_0$ <sup>[2]</sup> (一般取  $T_0 = 20 \text{ min}$ ), 则  $l_i$  次列车从  $i$  站到  $j$  站换乘  $l_i'$  次需换乘间隔为:  $t_{ij}(l_i, l_i') = t_{l_i'}^1(j) - t_{l_i}^2(i)$  (其中

$$t_{l_i'}^1(j) - t_{l_i}^2(i) \geq T_0)$$

5) 乘车费用: 铁路票价因列车、座位类型不同而存在差异, 附录中已给出部分站点之间的乘车费用, 在此设  $l_i$  次列车从  $i$  站到  $j$  站的乘车费用为:  $p_{ij}(l_i)$

最终构建出如下模型:

设  $x_{ij}^{l_i}$  表示从  $i$  站到  $j$  站是否乘坐了  $l_i$  次列车, 且:  $x_{ij}^{l_i} = \begin{cases} 1, & \text{乘坐了 } l_i \text{ 列车} \\ 0, & \text{未乘坐} \end{cases}$

(其中  $i, j \in v, l_i \in L$ )  $o, d$  分别为始发站和终到站, 构建出如下多目标模型:

$$\begin{cases} \min z_1 = \sum_{i \in v} \sum_{j \in v} \sum_{l_i \in L} [t_{l_i}^1(j) - t_{l_i}^2(i)] x_{ij}^{l_i} & (1) \\ \min z_2 = \sum_{i \in v} \sum_{j \in v} \sum_{l_i \in L} x_{ij}^{l_i} - 1 & (2) \\ \min z_3 = \sum_{i \in v} \sum_{j \in v} \sum_{l_i, l'_i \in L} \sum_{m \in V'} x_{ij}^{l_i} x_{jm}^{l'_i} d(v_{l_i}^j, v_{l'_i}^j) & (3) \\ \min z_4 = \sum_{i \in v} \sum_{j \in v} \sum_{l_i, l'_i \in L} \sum_{m \in V'} x_{ij}^{l_i} x_{jm}^{l'_i} t_{ij}(l_i, l'_i) & (4) \\ \min z_5 = \sum_{i \in v} \sum_{j \in v} \sum_{l_i \in L} p_{ij}(l_i) x_{ij}^{l_i} & (5) \end{cases}$$

$$\begin{cases} \sum_{l_i \in L} \sum_{j \in v} x_{oj}^{l_i} = 1 & (6) \\ \sum_{l_i \in L} \sum_{i \in v} x_{im}^{l_i} - \sum_{l_i \in L} \sum_{j \in v} x_{mj}^{l_i} = 0; m \neq o, d; \forall m \in V & (7) \\ s.t. \begin{cases} \sum_{l_i \in L} \sum_{i \in v} x_{jd}^{l_i} = 1 & (8) \\ \sum_{l_i \in L} \sum_{i \in v} \sum_{j \in v} x_{ij}^{l_i} - 1 \leq N_{\max} & (9) \\ x_{ij}^{l_i} \in \{0, 1\}; \forall i \in V, j \in V, l_i \in L & (10) \end{cases} \end{cases}$$

其中 (1) 式到 (5) 式分别表示乘车时间最短、换乘次数最少、换乘距离最短、换乘时间最短和费用最少; (6) 式到 (8) 式保证始发站与终到站之间有一条可行路径; (9) 式使得换乘次数控制在两次之内; (10) 式是 0-1 变量约束。

### 5.3 问题一的求解

多目标优化问题的绝对最优解往往是不存在的<sup>[3]</sup>, 因此最短路算法并不能解决多目标最优路径的问题, 处理多目标模型的方法有多种, 如理想点法、平方和加权法、极大极小法, 线性加权法等, 其中加权法是最容易理解的方法, 根据各目标函数的重要程度构造评价函数, 这里采用信息熵法确定各目标的权重, 对各条可行路径进行综合目标值计算, 找出较优路径方案。

信息熵法是一种常用的确定多目标权重的方法, 基于“差异驱动”的原理<sup>[4]</sup>, 突出局部差异, 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

设可行路径条数为  $M$ , 那么  $M$  条可行路径与 5 个目标值就构成了一个决策矩阵, 设这个决策矩阵为:  $MA^{od}_{M \times 5}$ , 但是由于决策矩阵的元素为区间数, 无法直接利用熵值法, 因此要进行量化处理, 具体步骤如下:

1) 将决策矩阵  $MA^{od}_{M \times 5}$  转化为相离度矩阵;

2) 将相离度矩阵规范化, 得到规范化矩阵  $R_{ij}^{od} = [r_{ij}^{od}]_{M \times 5}$ ;

3) 将规范化矩阵  $R_{ij}^{od} = [r_{ij}^{od}]_{M \times 5}$  进行列归一化运算<sup>[4]</sup>, 得到归一化矩阵

$$H_{ij}^{od} = [h_{ij}^{od}]_{M \times 5};$$

4) 计算第  $j$  个目标的熵值  $S_j = -(\ln M)^{-1} \sum_{i=1}^M h_{ij}^{od} \ln h_{ij}^{od}, j=1, 2, \dots, 6$ , 当

$$h_{ij}^{od} = 0 \text{ 时, 令 } h_{ij}^{od} \ln h_{ij}^{od} = 0;$$

5) 计算第  $j$  个目标的权重系数  $w_j^{od} = \frac{1 - S_j^{od}}{\sum_{k=1}^5 (1 - S_k^{od})}, j=1, 2, \dots, 6$

6) 计算路径  $i$  的综合目标值  $z_i^{od} = \sum_{j=1}^5 w_j^{od} r_{ij}^{od}, i=1, 2, \dots, M$

最后根据得到的综合目标值, 按照从大到小对可行路径进行排序, 排在前面的路径为较优路径。

得到的可供选择路径如下表 (程序见附录二):

表 2: 丹东到宜昌路径选择

起始点	目的地	耗时	费用	换乘次数	中转站	车次
丹东	宜昌东	37 小时 40 分钟	603	1	常州	K190/K187, D3006/D3007
		37 小时 40 分钟	617	1	无锡	K190/K187, D3006/D3007
		37 小时 40 分钟	572	1	镇江	K190/K187, D3006/D3007
		47 小时 9 分钟	404	1	常州	K190/K187, D3006/D3007
		47 小时 9 分钟	392	1	镇江	K190/K187, D3006/D3007
		42 小时 4 分钟	354	1	南京	K190/K187, K696/K697

并且求得模型中的综合目标值分别为 0.1757, 0.1072, 0.2771, 0.0797, 0.2161, 0.1442 根据综合目标值的大小排序进而得出最优路线为第 4 条路径即



丹东—常州—宜昌，车次 K190/K187，D3006/D3007。

表 3：天津到拉萨路径选择

起始点	目的点	耗时	费用（元）	车次	换乘次数	中转站
天津	拉萨	66 时 57 分	522	'1230/1227' 'T164/T165'	1	上海
		66 时 57 分	515	'4310/0000' 'T164/T165'	1	上海
		67 时 41 分	516	'K2082/K2083' 'T164/T165'	1	上海
		51 时 33 分	386	'K257' 'T164/T165'	1	郑州
		62 时 14 分	460	'K346/K347' 'T164/T165'	1	南京
		67 时 20 分	525	'K518/K515' 'T164/T165'	1	上海
		62 时 01 分	451	'K554/K551' 'T164/T165'	1	南京
		68 时 13 分	524	'K58/K55' 'T164/T165'	1	上海
		71 时 53 分	518	'K257' 'T22/T23'	1	广元
		83 时 37 分	654	'T122/T123' 'T264/T265'	1	广州
		59 时 01 分	432	'T122/T123' 'T264/T265'	1	武昌
		60 时 16 分	431	'T238/T235' 'T264/T265'	1	武昌

并且求得模型中的综合目标值分别为 0.0813, 0.0812, 0.0917, 0.0310, 0.1074, 0.0981, 0.1170, 0.1065, 0.0446, 0.1274, 0.0328, 0.0810, 则可以根据综合目标值的大小排序进而得出最优路线为第 4 条路径天津—郑州—拉萨，车次为'K257' 和' T164/T165'。

起始点	目的地	耗时	费用	中转次数	中转站	车次
白城	青岛	28 小时 58 分钟	222	1	长春	K7304 K1056/K1053
		29 小时 7 分钟	232	1	长春	K7304 K704/K701
		31 小时 51 分钟	208	1	黄村	K1302 K712/K709
		38 小时 40 分钟	273	1	丹东	2690/2688/2685 K958/K955

表 4：白城到青岛路径选择

并且求得模型中的综合目标值分别为 0.1639, 0.1889, 0.3010, 0.3462 根据综合目标值的大小排序进而得出最优路线为第 1 条白城—长春—青岛，车次 K7304K704/K709。

#### 5.4 问题一结果的分析与检验

为了验证问题一所求结果的合理与准确性，将所得结果与最新车次<sup>[5]</sup>情况对比

表 5:2014 年 8 月丹东到宜昌路径选择

起始点	丹东	丹东	丹东	丹东
目的点	宜昌东	宜昌东	宜昌东	宜昌东
耗时	1 天 12 小时 58 分	1 天 13 小时 47 分	1 天 1 小时 34 分	1 天 8 小时 28 分
换乘次数(次)	1	1	1	2
换乘时间	3 小时 14 分	2 小时 50 分	4 小时 20 分	4 小时 13 分
费用(元)	428 起	511.5 起	758 起	338.5 起
距离(公里)	2899	3037	2684	2684
车型	空调特快、动车组	空调特快、动车组	空调快速、高速动车	空调快速、空调特快
路线	丹东-南京-南京南-宜昌东	丹东-镇江-宜昌东	丹东-北京-北京西-宜昌东	丹东-天津-郑州-宜昌东

表 6:2014 年 8 月天津到拉萨路径选择

起始点	天津	天津	天津	天津
-----	----	----	----	----

目的点	拉萨	拉萨	拉萨	拉萨
耗时	2 天 7 时 5 分	1 天 23 时 5 分	2 天 12 时 9 分	2 天 0 时 5 分
换乘次数(次)	1	1	1	1
换乘时间	5 时 45 分	3 时 38 分	9 时 41 分	4 时 22 分
费用(元)	440 起	619 起	463 起	390.5 起
距离(公里)	4206	4294	4714	3884
车型	空调特快	高速动车、空调特快	空调特快、空调快速	空调特快
路线	天津-郑州-拉萨	天津-徐州-拉萨	天津-宝鸡-拉萨	天津-北京-拉萨

表 7:2014 年 8 月白城到青岛路径选择

起始点	白城	白城	白城	白城
目的点	青岛	青岛	青岛	青岛
耗时	1 天 6 时 9 分	1 天 8 时 9 分	23 时 0 分	1 天 8 时 8 分
换乘次数(次)	1	1	1	1
换乘时间	9 时 10 分	9 时 12 分	5 时 34 分	6 时 17 分
费用(元)	383.5 起	222.5 起	494 起	239.5 起
距离(公里)	2068	1847	1797	1989
车型	空调普快、动车组	空调普快、空调快速	空调快速、高速动车	空调普快、空调快速
路线	白城-北京-青岛	白城-天津-青岛	白城-秦皇岛-青岛	白城-沈阳-青岛

表 8:问题一的解答

起点站	终点站	换乘次数	中转站	时间	费用
丹东	宜昌东	1	常州	47 时 9 分	404
天津	拉萨	1	郑州	51 时 33 分	386
白城	青岛	1	长春	28 时 58 分	222

由上述对比可以发现，应用所给数据建立的模型求解出来的结果具有一定的参考价值，最新数据与以往的乘车情况相比，价格上总体呈上升趋势，这与社会

经济条件变好有一定关联，同时由于生活节奏日益加快，人们对于时间的要求越来越高，可以看到乘车时间有缩短趋势，由此可预测到未来乘车时间将会愈来愈短，紧跟人们的生活节奏。

## 六、问题二的求解

### 6.1 问题二准备工作

旅游作为一首人们放松，休闲的生活插曲，人们都乐于追求它。可是面对现实生活的各种约束，例如：假期少，资金短缺以及各种交通困扰等等，所以当有那么一次旅游的机会时，人们都想选择一条最优的旅游线路，尽情地享受旅游带来的快乐。因此，为了构建旅游最优线路模型，通过咨询与调查的信息，整理，作出图，如下：

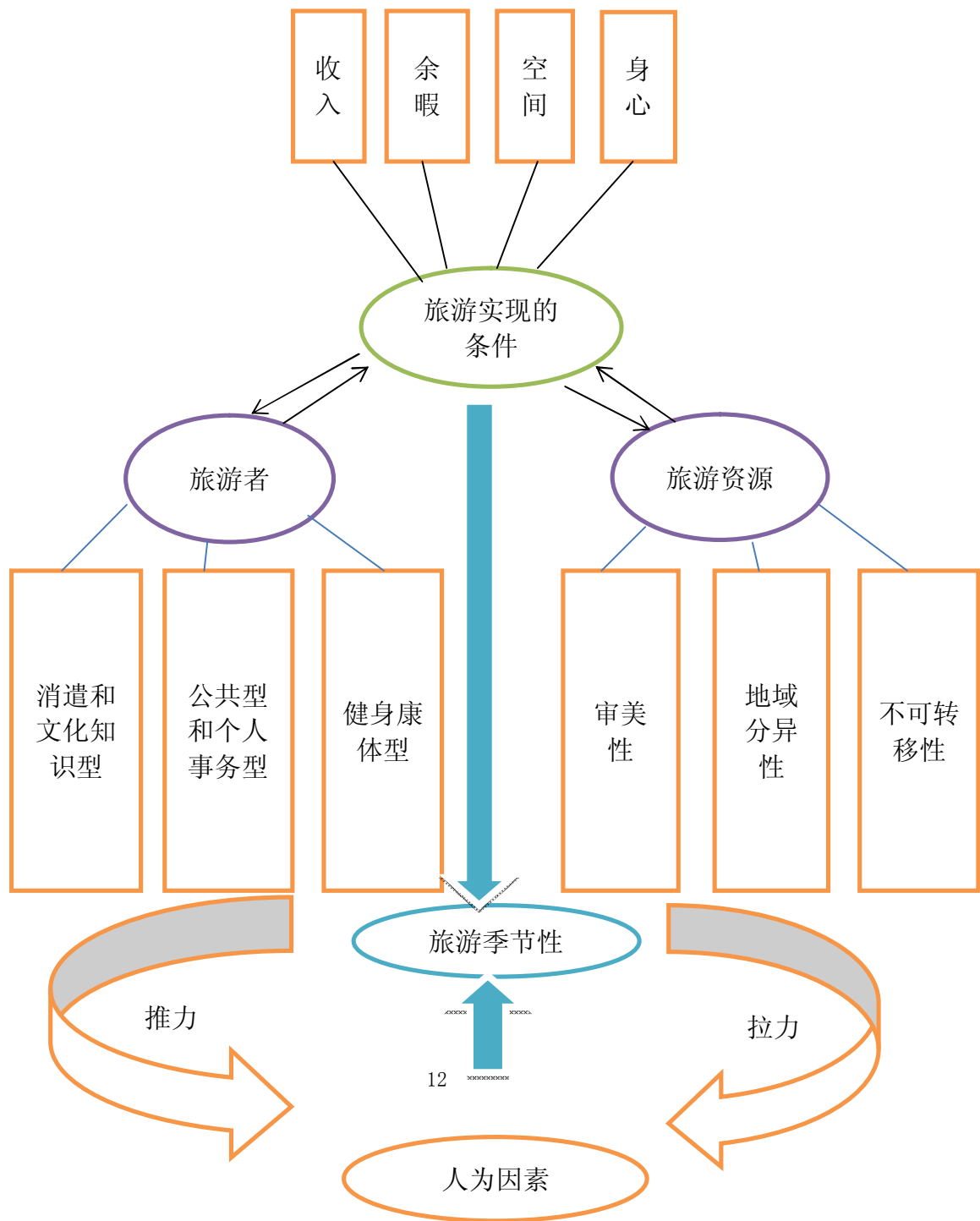


图 3：旅游分析

6.2 问题二的求解

按照与第一问类似的步骤进行求解，得到结果如下（程序见附录三）：

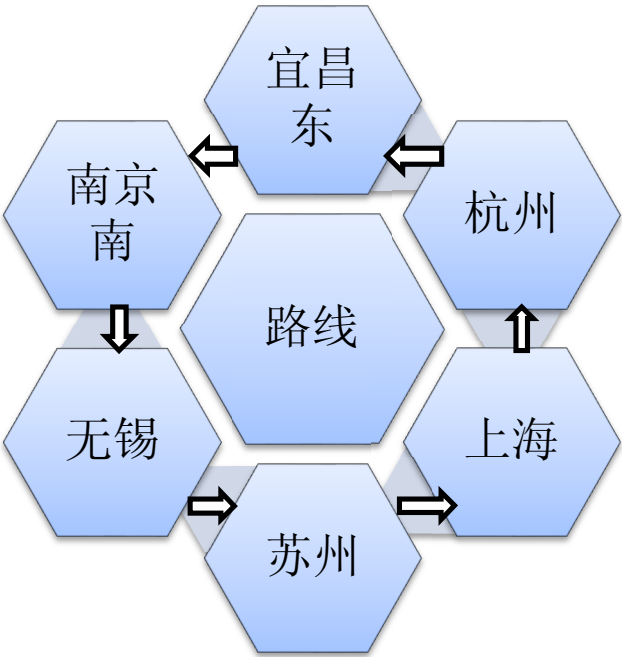


图 4：旅游路线循环图

表 9：旅游最优路线表

	旅游最优路线					
路线	宜昌东-南京南	南京南-无锡	无锡-苏州	苏州-上海	上海-杭州	杭州-宜昌
车次	D3008 D3005	D3008 D3005	D3008 D3005	D3008 D3005	G7505	G586

### 6.3 问题二的结果分析

与问题一一样，将所得路线与 2014 年网上旅游路线进行对比：

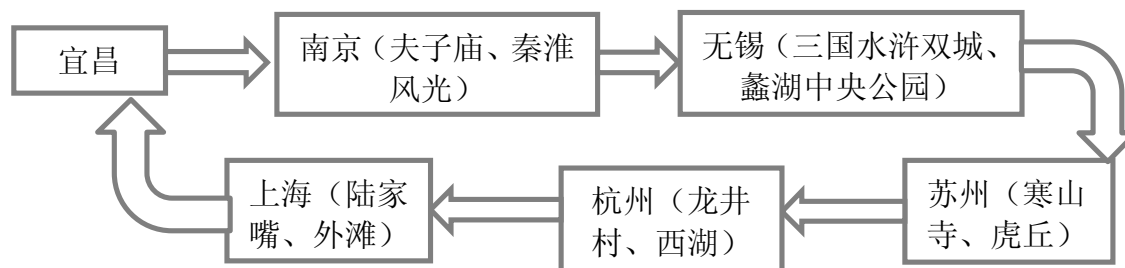


图 3:2014 年较佳旅游线路选择 (1)

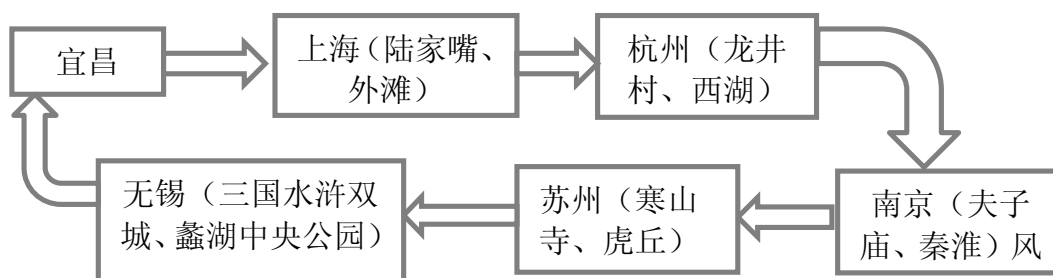


图 4:2014 年较佳旅游线路选择 (2)

由此可知，虽然各路径顺序相差不大，但是一条成功的行程推出后并非一劳永逸，因为这条行程是经过当时的市场调查，设计时也是针对某一部分旅客当时的情况的，例如以时间为设计支点的行程随着时间的推移，必然会随着其依傍的因素变化而日益不能适应新的市场要求。一条行程在经过了推广期、成熟期后必然将走上淘汰期，这是不以设计者的已知为转移的，因此设计者的思维要不断推陈出新，考虑的更为全面，与时俱进。

## 七、模型的评价

### 7.1 模型的优点

(1) 建立多目标规划模型。针对多种约束条件下构建的多目标规划模型与单目标规划模型相比，前者所用的方法更加先进，对结果的优化程度也相对更高，同时也是结果更加科学、精准。

(2) 采用求信息熵的方式对每个目标进行加权转化为综合单目标求解，更加科

学。

(3) 模型注重理论与现实的结合。构建模型得到的理论结果之后，就会查找相应的实际值，通过理论与实际对比，就能进一步增加模型的精确度。

## 7.2 模型的缺点

(1) 没有考虑换乘时间过长错过车次和过夜的问题，这样时间差不能得到科学的计算。

(2) 模型一的约束条件太多，就使得模型存在一定的局限性，特别是针对不同人群而言，或者在不同的时期，他们对于铁路出行所看中的重要因素不尽相同，因此，需要针对不同人群，在不同时间，构建与之相符合的出行模型。

# 八、模型的改进与推广

## 针对模型一：

### 8.1 改进一：

问题一的求解中没有考虑换乘时间过长错过车次和过夜的问题，然而实际情况中换乘时间间隔可能过长或太短，若差值小于 1 个紧接续标准  $T_0$ （上文中已做说明，一般取  $T_0=20\text{min}$ ），则时差应加一天，即 1440min，那么乘坐  $l_i$  次列车从  $i$  站到  $j$  站所需的换乘间隔时间为：

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

### 8.2 改进二：

另外舒适性是现代旅客交通运输追求的服务特性之一，在模型改进中采用文献[6]的研究成果，用旅行疲劳恢复时间来描述旅客选择不同车次出行的舒适程度，铁路旅客选择某车次出行的疲劳程度越大，需要的恢复疲劳的时间就越长。因此，这里选择乘坐某车次的疲劳恢复时间的价值来表示乘坐该车次的舒适度。铁路旅客乘坐  $l_i$  次列车从  $i$  站到  $j$  站的疲劳恢复时间为：

$$g[t_{ij}(l_i)] = T_{\max} / \{1 + \alpha_{l_i} \exp[-\beta_{l_i} t_{ij}(l_i) / 60]\}$$

其中  $T_{\max}$  表示恢复疲劳所需的最长时间参数，通常取 14 到 15 小时<sup>[7]</sup>； $\alpha_{l_i}$  表

示当乘车时间为 0 时，选择  $l_i$  次列车类型的疲劳恢复时间（即出行疲劳恢复时间的最小值）； $\beta_{l_i}$  表示单位出行时间的疲劳恢复时间强度系数，其值越大则疲劳恢复时间越长<sup>[8]</sup>，参数  $\beta_{l_i} > 0$ 。如果  $T_{\max}$  取 15 小时，则参数  $\alpha, \beta$  的取值由相应的列车种类决定。

## 针对模型二：

旅游的发生依赖于人们闲暇时间的多少，时间因素对旅游者选择路线来说有一下影响：

一是旅游者闲暇时间的多少与分布。不难发现出游时间与闲暇时间成正相关，而旅游者的闲暇时间分布影响到旅游业的淡、旺季分布。因此在下文中要将旅游者出行的时间进行人性化分类<sup>[6]</sup>，在这个分类下进行讨论时间分布。

二是旅游者群体的分类与差异。旅游是人产生的行为，在旅游中是起决定作用的主体，这直接关系到旅游时间与旅游路线分布的格局，因此本题中要对旅游群体进行分类研究，所谓“适之优之”，因地制宜，量身定做应为解题之本。

## 8.3 改进三：

旅游本身存在淡旺季，针对不同时期，不同人群，旅客所追求的最优的旅行线路也不相同，因此，通过在不同季节构建与之对应的模型，才能有效的选出相应的最优化铁路出行线路。

### 8.3.1 淡旺季分类下的最佳路线

旅游资源由于受到自然力和社会因素的作用，往往在时间上呈现出一定的变化性，即具有一定的节律性，表现出所谓的淡旺季差异，或出现其他费周期性的变化<sup>[9]</sup>，淡旺季的划分一般依据旅游者数量、旅游者支出及住宿天数。

- (1) 若该游客在淡季出游则拥挤状况、缺票情况等出现较少，对时间的要求较低。
- (2) 若该游客在旺季出游则考虑拥挤状况、缺票、住宿等问题。

### 8.3.2 旅游人群分类下的最佳路线

#### (1) 消遣型和文化知识旅游者

这两类旅游者主要选择观光旅游、娱乐旅游等类型的旅游活动，前者多与家庭成员或单位组织在一起，把旅游作为他的职业无关活动，其主要目的就是为了放松，后者主要通过地域、风土人情了解历史和文化，两者的共同点都是从旅游地中真实获得所需，因此在考虑路线时，对时间的要求相对较松，形式多样化占据主导地位。在本次路线中多考虑景点的选择。



## (2) 公共型和个人事务型旅游者

这两类旅游者一般具有一定的社会地位或者人物关系相对复杂，出游的目的可能是由于工作的需要或处理事务，出游的季节性较差，因此时间恰恰是需要考虑的重中之重，那么在本次的路线中需要将时间因素占的比例增大。

## (3) 健身康体型旅游者

这种类型的旅游者是指那些在惯常生活环境以外，以健康、医疗为主要目的的旅游者，他们主要参加体育旅游、保健旅游和生态旅游，他们一般拥有较高的收入和较多的闲暇时间，但是旅途一定要保证身心愉悦，则在考虑出行路线的问题上，对时间和费用的要求相对较低，而旅途转乘次数一定要合理。

## 九、参考文献

- [1]郭彦云 城市轨道交通有效路径研究
- [2]崔炳谋等 铁路旅客旅行换乘方案优选算法[J] 中国铁道科学 2007, 28
- [3]王丽颖等 多目标规划模型的应用研究 白城师范学院学报 2010-12-15
- [4]朱方霞等 熵值法 安徽大学学报 2005.9 第 30 卷第 5 期
- [5]<http://train.qunar.com/stationToStation.htm?fromStation=%E4%B8%B9%E4%B8%9C&toStation=%E5%AE%9C%E6%98%8C&date=2014-08-16>
- [6]马勇 旅游学概论 高等教育出版社 1998.8
- [7]傅云新 旅游学概论 暨南大学出版社 2003 73-96
- [8]孙文昌 旅游学导论 青岛出版社 1992 66-100
- [9]史峰等 客运专线相关旅客列车开行方案研究[J] 铁道学报 2004 26 (2)

## 十、附录

附录一：

```
clc;clear;
[a b c]=xlsread('附件一','sheet1','E2:E49370');
[C,ia,ic]=unique(b,'stable');
```

附录二：

```
% i0Öµ·È·¶ÈÖÖ
% È·¶Ö,±ê²ãÈÖÖ
clear;
clc;
x=[99.3600      380952974700.0000      104.8900      1.6400
9.0500      8.2800      101452980.1100      1167334984012.1600
102.1300      228127.0000      5304940.0000      29817.0000
92.1100      466610267000.0000      111.6500      4.3300
5.4400      8.1600      75883744.8300      861424359113.9300
```

94.6000	1253922.0000	8087900.0000	85194.0000
109.1000	305694698312.7900	110.8800	2.6900
4.1700	8.4000	457459478.2500	1179235081612.4700
110.0600	468307.0000	677090.0000	63273.0000
109.1000	595020144866.6500	99.3000	2.0100
4.5700	4.3000	783157007.2100	5007646096891.2900
109.2600	1993306.0000	1930440.0000	65491.0000
98.2200	372654923215.5300	107.1300	1.4400
8.5100	23.8000	18848868.8700	704821878325.1800
101.1000	162951.0000	62990.0000	22051.0000
104.6500	365675537000.0000	111.7600	2.1400
5.0200	9.3900	27417364.3700	89952506168.7600
104.3300	87487.0000	670.0000	5195.0000
101.2900	366895501851.6100	101.8200	1.9400
8.5100	5.5800	11549288.9200	1258455637300.0000
103.9700	309613.0000	1539200.0000	9674.0000
100.8500	392520087435.9700	106.2700	0.7600
9.5300	8.6500	19599155.2300	48932431697.2000
101.4400	82859.0000	305990.0000	25023.0000
96.3600	347542443470.0800	99.6000	1.2600
9.6900	18.0100	22939492.6400	1297226915351.5100
99.0900	29917.0000	172930.0000	15043.0000
97.3700	100.0000	99.6400	4.6800
9.2600	158995068.0700	15077285740000.0000	98.4000
1688822.0000	3021080.0000	226205.0000	

```

];
y=[]
[m,n]=size(x);
for i=1:n
    y(:,i)=x(:,i)/sum([x(:,i)]) % Ô-Ê¼¼ÖÖ¹éÒ»»-
end
for l=1:n
    s(1,l)=0;
    for j=1:m
        p(1,l)=y(j,l)*log(y(j,l))
        s(1,l)=s(1,l)+p(1,l)
    end
end
k=(log(m))^(-1)
e=-k*s
h=ones(1,n)-e
w=h/sum(h) % Ö,±êÈ"ÖÖÖµ
sum(w)
% ¼ÆËä×Û°ÏÆÀ¼ÛÖµ

```

```
g=y*w'
```

附录三:

```
clc
clear
[a, d, p] = xlsread('144p0');
[ta, td, tp] = xlsread('timetable');
str_de = '0E2y';
str_st =
'μ';% 'E' | EÎÔâÊäÈèÈ«¹úÁ½, öμøμã¶½ÉÔÔμÃ³ö»»³ÈÔ»´ÎμÄÈùÓÐÂ·Ïβ£¬°ÍÃ¿, ö·
½°, ¶ÔÓ|μÄ×Û°Ï±ê×¼£¬Ã¿, ö»»³È·½°, ¶ÔÓ|ÓëmatÔÐμÄrealtrain°Írealtranstatio
n

b1 = find(strcmp(d(:,4), str_de)==1);
b2 = find(strcmp(d(:,4), str_st)==1);
n1 = length(b1);
n2 = length(b2);

c1 = cell(n1, 1);
c2 = cell(n2, 1);

for i = 1:n1
    c1{i} = d{b1(i), 1};
end

for j = 1:n2
    c2{j} = d{b2(j), 1};
end

[C1, ia, ib] = unique(d(:, 1), 'stable');
[C2, ja, jb] = unique(d(:, 1), 'stable');

temp = p(2:end,3);
temp = cell2mat(temp);
index = find(temp==1);
train = cell(length(index), 1);
traintime = cell(length(index), 1);
for i = 1:length(index)-1
    train{i} = p(index(i)+1:index(i+1), :);
    traintime{i} = tp(index(i)+1:index(i+1), :);
end
```

```

train{length(index)} = p(index(end)+1:size(p, 1), :);
traintime{length(index)} = tp(index(end)+1:size(tp, 1), :);

newC1 = C1(2:end);
[X1, Y1] = size(c1);
Index1 = zeros(X1, Y1);

newC2 = C2(2:end);
[X2, Y2] = size(c2);
Index2 = zeros(X2, Y2);

for m = 1:X1
    Index1(m) = find(strcmp(newC1(:, 1), char(c1(m, 1))));
end
for n = 1:X2
    Index2(n) = find(strcmp(newC2(:, 1), char(c2(n, 1))));
end
P1 = train(Index1);
tP1 = traintime(Index1);
P2 = train(Index2);
tP2 = traintime(Index2);

Place1 = cell(X1, 1);
Money1 = cell(X1, 1);
TrainNum1 = cell(X1, 1);
Place2 = cell(X2, 1);
Money2 = cell(X2, 1);
TrainNum2 = cell(X2, 1);
for m = 1:X1
    Place1(m) = {P1{m}(:, 4)};
    Money1(m) = {P1{m}(:, 9:12)};
    TrainNum1(m) = {P1{m}(1, 1)};
end
for n = 1:X2
    Place2(n) = {P2{n}(:, 4)};
    Money2(n) = {P2{n}(:, 9:12)};
    TrainNum2(n) = {P2{n}(1, 1)};
end

r=[];
t=[];
n=0;
for i=1:length(Place1)

```

```

for j=1:length(Place2)
    sat = intersect(Place1{i}, Place2{j});
    sat(strcmp(sat, str_de) | strcmp(sat, str_st)) =
[]; %È¥³ý»¼¯ÖĐ:ÉÄÜ°ü°¬ÆðÊ¼μø°ÍÄ:μÄμøμÄÇé:ö *Ö±´ıŁ¬:ÉĐĐ,Ä
    if ~isempty(sat)
        n=n+1;
        result{n}=sat;
        r=[r i];
        t=[t j];
    end
end
end

flag_de = false;
flag_st = false;
flag = false;

lineproject = zeros(length(result), 2);
st_de_pos = zeros(length(result), 2); %ÆðÊ¼Ö¼°ÍÖÖμăÖ¼ÖÚ³μ´İÖĐμÄİ»ÖÄ
transtation = cell(length(result), 1);
k = 0;
for m = 1:length(result)
    % if k == 388
    % end
    k1 = 0;
    trantmp = cell(length(result{m}), 1);
    for n = 1:length(result{m})
        compos = result{m}(n); %ÖĐ×ªÖ¼

        depoint = find(strcmp(Place1{r(m)},
compos)); %ÖÖμ¼ÖĐ×ªÖ¼ÖÚ¹ýÖÖμă³μ´İÖĐμÄİ»ÖÄ
        for i = depoint:length(Place1{r(m)})
            if strcmp(str_de, Place1{r(m)}(i))
                flag_de = true;
                break;
            end
        end
    end

    stpoint = find(strcmp(Place2{t(m)},
compos)); %ÖÖμ¼ÖĐ×ªÖ¼ÖÚ¹ýÆðμă³μ´İÖĐμÄİ»ÖÄ
    for j = 1:stpoint
        if strcmp(str_st, Place2{t(m)}(j))
            flag_st = true;
            break;
        end
    end
end

```

```

        end
    end

    if flag_de && flag_st
        if ~flag %¿ØÖÆÖ»½ØÈèÒ»´Î
            k = k + 1;
            lineproject(k, 1) = t(m);
            lineproject(k, 2) = r(m);
            st_de_pos(k, :) = [j, i];
            flag = true;
        end
        k1 = k1 + 1;
        trantmp(k1) = compos; %¿ØÖÆ°İ,ñµÄÖÐ×ªÖ¼´æÔÚ
    end
    flag_de = false;
    flag_st = false;
end
if k1 > 0 %½«Âú×ãİð¼µÄÍ-Ò»³µ´İµÄÖÐ×ªÖ¼´æÔÚÒ»Æð
    transtation(k) = {trantmp(1:k1)};
end
flag = false;
end
%È¥³ÝÈßÓàÊÝ¼Ý
lineproject(lineproject==0) = [];
projectnum = length(lineproject)/2; %×Û·½°,ÊÝÄ¿
lineproject = reshape(lineproject, projectnum, 2);
st_de_pos = st_de_pos(1:projectnum, :);
transtation = transtation(1:projectnum);

allprojectnum = 0; %È«²¿·½°,£¬°üÀ´¶ÔÓ!ÖÐ×ªÖ¼
for m = 1:projectnum
    allprojectnum = allprojectnum + length(transtation{m});
end

first_train = TrainNum2(lineproject(:, 1));
second_train = TrainNum1(lineproject(:, 2));

first_traintimetable = tP2(lineproject(:, 1));
second_traintimetable = tP1(lineproject(:, 2));

realproject = zeros(allprojectnum, 2); %×İÖÖ·½°,
real_st_de_pos = zeros(allprojectnum, 2); %ÆðÊ¼Ö¼ÓèÖÖµãÖ¼Î»ÖÃ
transtation_pos = zeros(allprojectnum, 2); %ÖÐ×ªÖ¼Î»ÖÃ
realtrain = cell(allprojectnum, 2); %¶ÔÓ!³µ´İ*

```

```

realtranstation = cell(allprojectnum, 1); %TÓÓ!ÖÐ×ªÖ¼*
first_traintime = cell(allprojectnum, 2); %ÚÓ»Á³³µµ½´îÊ±¼ä°íÀè¿ªÊ±¼ä*
second_traintime = cell(allprojectnum, 2); %ÚÓ¶Á³³µµ½´îÊ±¼ä°íÀè¿ªÊ±¼ä*
internaltime = zeros(allprojectnum, 1); %µÊ´ýÊ±¼ä*

k = 0;
for m = 1:projectnum
    for n = 1:length(transtation{m})
        compos = transtation{m}(n); %ÖÐ×ªÖ¼
        depoint = find(strcmp(Place1{lineproject(m, 2)},
compos)); %Öòµ½ÖÐ×ªÖ¼ÓÚ´ýÖòµã³µ´îÖòµÄî»ÖÄ
        stpoint = find(strcmp(Place2{lineproject(m, 1)},
compos)); %Öòµ½ÖÐ×ªÖ¼ÓÚ´ýÆðµã³µ´îÖòµÄî»ÖÄ
        [val1, vl1, in1] =
calculatetime(first_traintimetable{m}(stpoint, :));
        [va2, vl2, in2] =
calculatetime(second_traintimetable{m}(depoint, :));
        waitTime = vl2 - val1;
        if waitTime > 20 && waitTime < 3*60
            k = k + 1;
            realproject(k, :) = lineproject(m, :);
            real_st_de_pos(k, :) = st_de_pos(m, :);
            transtation_pos(k, :) = [stpoint, depoint];
            realtrain(k, :) = [first_train(m), second_train(m)];
            first_traintime(k, :) = first_traintimetable{m}(stpoint,
2:3);
            second_traintime(k, :) = second_traintimetable{m}(depoint,
2:3);
            internaltime(k) = waitTime;
            realtranstation(k) = compos;
        end
    end
end
end
%ÈÝ³ýÈðÓàÊýÝ
realproject(realproject==0) = [];
realprojectnum = length(realproject)/2; %È·ÇÐ½°,ÊýÄ¿
realproject = reshape(realproject, realprojectnum, 2);
real_st_de_pos = real_st_de_pos(1:realprojectnum, :);
transtation_pos = transtation_pos(1:realprojectnum, :);
realtrain = realtrain(1:realprojectnum, :);
first_traintime = first_traintime(1:realprojectnum, :);
second_traintime = second_traintime(1:realprojectnum, :);
realtranstation = realtranstation(1:realprojectnum);
internaltime = internaltime(1:realprojectnum);

```

```

totaltime = zeros(realprojectnum, 1); %×ÛÊ±¼ä*
for m = 1:realprojectnum
    [va1, vl1, in1, da1] = calculatetime(tP2{realproject(m,
1)}(real_st_de_pos(m, 1), :)); %µ½´iÖÐ×ªÖ¾Ê±¿Ì
    [va2, vl2, in2, da2] = calculatetime(tP2{realproject(m,
1)}(transtation_pos(m, 1), :)); %ÆÖÊ¼Ê±¿Ì
    waitTime1 = 1440 * (da2 - da1) + va2 - va1;
    [va3, vl3, in3, da3] = calculatetime(tP1{realproject(m,
2)}(transtation_pos(m, 2), :)); %ÀË¿ªÖÐ×ªÖ¾Ê±¿Ì
    [va4, vl4, in4, da4] = calculatetime(tP1{realproject(m,
2)}(real_st_de_pos(m, 2), :)); %µ½´iÊ±¿Ì
    waitTime2 = 1440 * (da4 - da3) + vl4 - vl3;
    totaltime(m) = waitTime1 + waitTime2 + internaltime(m);
end

totalmoney = zeros(realprojectnum, 1); %×Û»··Ñ*
for m = 1:realprojectnum
    if m == 1
        end
        totalM = zeros(2, 1);
        %iÈ´ÓÖÐ×ªµÖ¾ÀÐ¶iÊÇ·ñÓÐìØ¶×ù´iÊ¬Òòîª´Ê¹Ã»ÓÐ,Ã³µ´iÊ¬ÆÖÊ¼Ö¾Ê±¼Û´!
        Ò²·Ç¿Ö
        Mo = [cell2mat(Money2{realproject(m, 1)}(transtation_pos(m,
1), :)); %Á¼¶¶Ê¬µÛÒ»¶¶ÊÇµÛ¼,ö³µ´iÊ¬µÛ¶¶¶¶ÊÇ,Ã³µ´iÖÐµÛ¼,öÖ¾
        cell2mat(Money2{realproject(m, 1)}(real_st_de_pos(m, 1), :));
        cell2mat(Money1{realproject(m, 2)}(transtation_pos(m, 2), :));
        cell2mat(Money1{realproject(m, 2)}(real_st_de_pos(m, 2), :))];
        for n = 1:2
            jtrain = char(realtrain{m, n});
            pM = find(~isnan(Mo(2*n-1, :)));
            if isempty(pM)
                totalM(:) = 0;
                break;
            end
            if jtrain(1) >= '0' && jtrain(1) <= '9' %ÆÖí´³µ
                ptmp = pM(1);
            else %ÆäËû³µ
                if isempty(find(pM == 2, 1))
                    ptmp = pM(1);
                else
                    ptmp = 2;
                end
            end
        end
    end
end

```



```

    m1 = Mo(2*n-1, pM(1));
    m2 = Mo(2*n, pM(1));
    totalM(n) = (-2 * n + 3) * (m1 - m2); %1->1, 2->-1
end
totalmoney(m) = totalM(1) + totalM(2);
end
[g, w] = calsynvalue([internaltime, totalmoney, totaltime]);
g

```