铁路最优路线的问题

摘要

本文解决的是铁路最优路线的问题，首先对题目所给 2013 年全国列车时刻表的相关数据进行处理与分析，然后针对不同的问题建立模型，最后选择铁路最优路线。

对于问题一：针对任意两个站点之间的最优铁路路线，确定以换乘时间最短、乘车费用最少为目标函数的多目标线性规划模型， 并设计出算法，通 MATLAB 编程实现，得到丹东→宜昌、天津→拉萨、白城→青岛起点到终点的最优路线如下：

起点站	离开时间	车次	到达时间	换乘站	离开时间	车次	到达时间	终点站	所用时间
丹东	第 1 天 8:02	K190/ K187	第 2 天 12:45	镇江	第 1 天 15:35	D3006 D3007	第 1 天 21:42	宜昌东	37 小时 40 分
天津	第 1 天 13:46	T5684 /T568	第 1 天 15:38	北京西	第 1 天 20:00	T27	第 3 天 15:40	拉萨	49 小时 54 分
白城	第 1 天 10:23	K7304	第 1 天 15:19	长春	第 1 天 16:27	K704/ K701	第 2 天 15:19	青岛	28 小时 56 分

对于问题二：从宜昌出发至华东五市旅游后在返回宜昌，引进图论的知识，将问题简化成求解 6 个点之间无返回的最佳路径问题，以整个行程乘车时间最短和乘车费用最小为目标函数，建立多目标线性规划模型，运用模拟退火算法解出整个旅游行程的最优路线如下：

起点站	离开时间	车次	到达时间	终点站	所用时间
宜昌东	14:40	G588/G585	20:26	南京南	5 小时 46 分
南京南	21:08	G21	21:52	无锡东	0 小时 44 分
无锡东	21:54	G21	22:04	苏州北	0 小时 10 分
苏州	16:39	G7583	18:08	杭州东	1 小时 29 分
杭州东	15:25	G7516	16:10	上海虹桥	0 小时 45 分
上海虹桥	13:54	D3006/D3007	21:42	宜昌东	7 小时 48 分

关键字： 多目标线性规划模型　模拟退火算法　最优路线

一：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 问题的相关信息

附录一给出 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。

1.3 需要解决的问题

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：丹东→宜昌、天津→拉萨、白城→青岛起点到终点的最优路线。

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。

二：问题假设与符号说明

2.1 模型假设

假设一：换乘时，第二路段的开车时间晚于第一路段的到达时间，皆为可选择的乘车路线。

假设二：火车从起点发车后都能在额定的时间里到达终点站。

假设三：列车的乘车费用按所行驶的距离确定。

假设四：在旅游 6 个城市时，两个城市之间只有一条最优的乘车线路。

2.2 符号说明

符号	含义
n	表示换乘次数

i, j, k, l, p, q	表示站点
W	表示总费用
t_n	表示换乘时间间隔
t_k	表示到达第 k 个站点的时间
t_j	表示到达第 j 个站点的时间
t_l	表示到达第 l 个站点的时间
t_s	表示总时间
Δt_1	表示第一次换乘的时间间隔
t_{ij}	表示第 i 个站点到第 j 个站点的单位时间
t_{jk}	表示第 j 个站点到第 k 个站点的单位时间
t_{pq}	表示第 p 个站点到第 q 个站点的单位时间
t_{kl}	表示第 k 个站点到第 l 个站点的单位时间
ϖ_{ij}	表示第 i 个站点到第 j 个站点单位距离的费用
s_{ij}	表示第 i 个站点到第 j 个站点的单位距离
ϖ_{jk}	表示 j 第个站点到第 k 个站点单位距离的费用
s_{jk}	表示第 j 个站点到第 k 个站点的单位距离

三：问题分析

本文研究的是铁路最优路线的问题，根据题中所给 2013 年全国列车时刻表的相关数据，探讨任意两个站点之间最优的铁路路线。在一票难求需要考虑换乘的情况下，不同的出行者对换乘次数、出行耗时以及乘车费用往往具有不同的需求，应选择换乘次数、耗时和花费都令乘客满意的最优路线。

3.1 问题一的分析

根据出行者的不同需求建立以换乘时间最短和乘车费用最少为目标函数，以换乘次数最少、换乘单次时间间隔最短、换乘时间总间隔最小、第一路段到达时间早于第二路段开车时间四个条件为约束条件，建立多目标线性规划模型，从若干个可能的方案中寻求某种意义下的最优方案。考虑到出行者对换乘次数有一个最大的承受上限，所以限定最大换乘次数为 2，即最多利用 3 条火车路线从起点到达终点站；限定单次换乘时的时间间隔不能超过 90 分钟；换乘时间间隔总和为 180 分钟内；第一路段到达时间早于第二路段开车时间，以减少所选择的路线。在求解过程中将目标函数分成优化级再逐步求解，搜索出换乘次数小于等于 2 的所有可行路线，分别计算出这些路线所花费的时间和费用，根据出行者不同的

需求，即选择目标函数，得出最优的铁路路线。再根据模型和算法求出丹东→宜昌、天津→拉萨、白城→青岛起点到终点的最优路线。

3.2 问题二的分析

旅游产业已经成为当今世界的第一产业[1]。国际上普遍认为旅游就是以休闲、娱乐、探亲访友或者商务目的进行的旅行活动。根据从宜昌出发乘坐火车至华东五市旅游后在返回宜昌，建立数学模型求出整个行程的最优路线。这是一个从起点出发旅游，最后回到出发点的问题。这与著名的旅行商问题（TSP）相似。引进图论的知识，将 6 个站点看成图论中的 6 个点，如果两个站点之间有可达路线，那么两点之间连接用有向边表示，列车在两个站点之间行驶的时间赋为有向边的权。问题二就简化成求解 6 个点之间无返回的最佳路径问题。以每个城市只有一条到达的路线和一条离开的路线、每个城市没有任何子回路为约束条件，以整个行程乘车时间最短和乘车费用最小为目标函数建立多目标线性规划模型，运用模拟退火算法求解出整个旅游行程的最优路线。

四：数据处理

4.1 车次编号

根据附录一给出 2013 年全国列车时刻表数据，对列车车次进行编号，利于将数据导入 MATLAB 进行计算，编号结果如表 1。

表 1：车次编号

车次编号	车次	车站编号	车站	到站时间	出发时间
1	1043	2358	西安	21:13:00	21:13:00
2	1044	838	奎屯	18:33:00	18:33:00
3	1051	809	天津	15:51:00	15:51:00
…	…	…	…	…	…
2342	G332	637	哈尔滨西	19:10:00	19:10:00
…	…	…	…	…	…
4681	Z94/Z91	2358	西安	17:08:00	17:08:00
4682	Z96/Z97	63	上海	19:07:00	19:07:00
4683	Z98/Z95	818	太原	19:45:00	19:45:00

4.2 时间处理

在考虑换乘的情况下，即先从乘车站到换乘站，再从换乘站到目的站，为了搜集所有从换乘站到达目的站的路线，并计算在换乘站的等待时间，对时间进行处理，例：表 2。

表 2：等待时间的处理

起点站	车次	到达时间	中点站	离开时间	车次	终点站	等待时间
天津	1136/1133	22:35	北京西	22:00	T27	拉萨	23 小时 25 分
白城	6340	16:47	长春	16:27	K704/K701	青岛	23 小时 40 分
丹东	K190/K187	11:56	南京	15:58	K1512/K1513	宜昌东	4 小时 2 分

五：问题一的解答

针对问题一，考虑到不同的出行者对换乘次数、出行耗时以及乘车费用往往具有不同的需求，建立以换乘时间最短和乘车费用最少为目标函数，以换乘次数最少、换乘单次时间间隔最短、换乘时间总间隔最小、第一路段到达时间早于第二路段开车时间四个条件为约束条件的多目标线性规划模型，得到乘坐列车的最优路线。

5.1 模型的建立

5.1.1.约束条件建立

（1）换乘次数的约束：

由于任意两个站点之间不一定都有直达列车，即使有直达列车也需要考虑票已售罄情况下最优的换乘方案，所以换乘次数应有一个最大的承受上限，本题限定最大换乘次数为 2，即：

$$1 \le [n] \le 2$$

（2）单次时间间隔的约束

由于换乘过程中时间间隔的不确定性，将换乘时间间隔进行限制。根据假设一：换乘时，第二路段的开车时间晚于第一路段的到达时间，皆为可选择的乘车路线，限定单次换乘时的时间间隔不能超过 90 分钟。即：

$$0 < t_n \le 90$$

（3）总时间间隔的约束

由于换乘过程中出行者不希望候车的时间过长，将换乘时间间隔总和进行限制。换乘时间间隔总和限制为 180 分钟内，即：

$$\sum_{n=1}^{n} \Delta t_n \le 180 \left(n = 1, 2\right)$$

（4）第一路段到达时间早于第二路段开车时间的约束

由于换乘过程中，第二路段到达终点的铁路路线可能过多，限制第一路段到达时间早于第二段开车时间。

$$\begin{cases} t_j < t_k < t_l \sin(n-1)\dfrac{\pi}{2} & (\text{换乘都在当天}) \\ t_j < t_k + 24(m-1) < (t_l + 24(m-1))\sin(n-1)\dfrac{\pi}{2} & (\text{换乘不在第一天}) \end{cases}$$

5.1.2.目标函数建立

（1） 换乘时间最短

要使任意两个站点之间的线路最优，火车在两点之间行驶的时间就要最短。行驶时间又包括乘车时间和换乘等待的时间，且换乘等待时间的总和不得超过180分钟。根据以上条件分别建立换乘一次与换乘两次乘车花费的总时间的目标函数，即：

1. 换乘一次的总时间

换乘一次的总时间为从第 i 站点到第 j 个站点的时间 t_{ij}，换乘时间间隔 Δt_1 和第 j 个站点到第 k 个站点的时间 t_{jk} 这三者的累加和最短。即：

$$\min t_s = t_{ij} + \Delta t_1 + t_{jk}$$

2. 换乘两次的总时间

换乘两次的总时间为从第 i 站点到第 j 个站点的时间，换乘两次时间间隔 $\Delta t_1, \Delta t_2$，第 j 个站点到第 k 个站点的时间和第 k 个站点到第 l 个站点的时间 t_{kl} 这四者的累加和最短。即：

$$\min t_s = t_{ij} + \Delta t_1 + \Delta t_2 + t_{jk} + t_{kl}$$

（2） 乘车费用最少

要使任意两个站点之间的线路最优，火车在两点之间乘车的费用就要最小。乘车费用根据选择的车型、座位不同而又有较大的差异，所以从起点到终点的过程中，乘坐每一段火车应付的费用应该分开考虑，分别建立换乘一次与换成两次总费用的目标函数，即：

1. 换乘一次总费用

从第 i 站点到第 j 个站点的费用为单位距离费用 ϖ_{ij} 乘以距离 s_{ij}，第 j 个站点到第 k 个站点的费用为单位距离费用 ϖ_{jk} 乘以距离 s_{jk}，换乘一次总费用为两者的累加和最少，即：

$$\min W = \varpi_{ij} s_{ij} + \varpi_{jk} s_{jk}$$

2. 换乘二次总费用

从第 i 站点到第 j 个站点的费用为单位距离费用 ϖ_{ij} 乘以距离 s_{ij}，第 j 个站点到第 k 个站点的费用为单位距离费用 ϖ_{jk} 乘以距离 s_{jk}，第 k 个站点到第 l 个站点的费用为单位距离费用 ϖ_{kl} 乘以距离 s_{kl}，换乘二次总费用为三者的累加和最少。即：

$$\min W = \varpi_{ij}s_{ij} + \varpi_{jk}s_{jk} + \varpi_{kl}s_{kl}$$

5.1.3 综上所述，得到组合优化模型为：

$$\begin{cases} \min t_s = t_{ij} + (2)^{n-1}\Delta t_n + t_{jk} + \sin(n-1)\dfrac{\pi}{2}t_{kl} \\ \min W = \varpi_{ij}s_{ij} + \varpi_{jk}s_{jk} + \sin(n-1)\dfrac{\pi}{2}\varpi_{kl}s_{kl} \end{cases}$$

$$s.t.\begin{cases} 1 \le n \le 2 \\ \sum_{n=1}^{n}\Delta t_n \le 60\ (n=1,2) \\ 0 < t_n \le 15 \\ t_j < t_k < t_l \sin(n-1)\dfrac{\pi}{2} \\ t_j < t_k + 24(m-1) < (t_l + 24(m-1))\sin(n-1)\dfrac{\pi}{2} \end{cases}$$

5.2 算法的描述

5.2.1 考虑直达

设任意两个站点为站点 A（起点）和站点 B（终点），通过站点 A 的所有列车线路记为 $P(A)$，通过站点 B 的所有列车线路记为 $P(B)$，如 $P(A)\cap P(B)\neq\varnothing$，则找出 $P(A)$ 与 $P(B)$ 的交集，即乘一次车就可由站点 A 到达站点 B。若可以一次到达，则找出交集中两点之间乘车时间最短，花费最少令乘客满意度最大的一条最优线路。

5.2.2 考虑换乘

（1）换乘一次

设任意两个站点为站点 A（起点）和站点 B（终点），通过站点 A 的所有列车线路计为 $P(A)$，通过站点 B 的所有列车线路为 $P(B)$，找出 $P(A)$ 中所有能转乘车次的集合 $P(A^z)$（假设有 N_1 次列车）如果 $P(A^z)\cap P(B)\neq\varnothing$，则找出此交集，按顺序找出这个交集中的车由哪些车次转来，并在交集中找出乘车时间，换

乘时间间隔和花费都令乘客满意的路线。即知站点 A 经一次换乘即可到达目的的站点 B 的最优路线。

转乘车次的集合$P(A^z)$的确定方法。令$P(A_i)$为经过站点 A 的第i辆车次。2.将集合$P(A^z)$置为空集，求出$P(A_i)$所经过的除去站点 A 外的所有站点的集合，记为$S(A_i)$（假设有N_2次）。其中$(i=0,1,2,...,N_1)$。3.求出经过站点$S(A_i)$的所有车次，记为$S(A_{ij})$。则得到线路的集合，记为$P(S(A_{ij}))$其中$(j=0,1,2,...,N_2)$。4.将得到的集合$P(S(A_{ij}))$加到转乘车次的集合$P(A^z)$中，重复步骤 3，直到$j=N_2$。5.重复步骤 3 和步骤 4，直到$i=N_1$。最后确定出转乘车次的集合$P(A^z)$。

（2）换乘两次

找出$P(A^z)$中车的所有能转乘的车次集合，记为$P(A^{zz})$（集合$P(A^{zz})$的求法同$P(A^z)$）。如果$P(A^{zz})\cap P(B)\neq\varnothing$。则找出交集，按顺序找出交集中的车由哪些车次转来。并在交集中找出乘车时间，换乘时间间隔和花费都令乘客满意的路线。即知站点 A 经两次换乘即可到达目的的站点 B。

5.3 模型的求解

针对任意两个站点之间的最优铁路路线，确定以换乘时间最短、乘车费用最少为目标函数的多目标线性规划模型，利用多目标优化算法，通过 MATLAB 编程实现（见附录 1），得到丹东→宜昌、天津→拉萨、白城→青岛起点到终点最优路线如表 3：

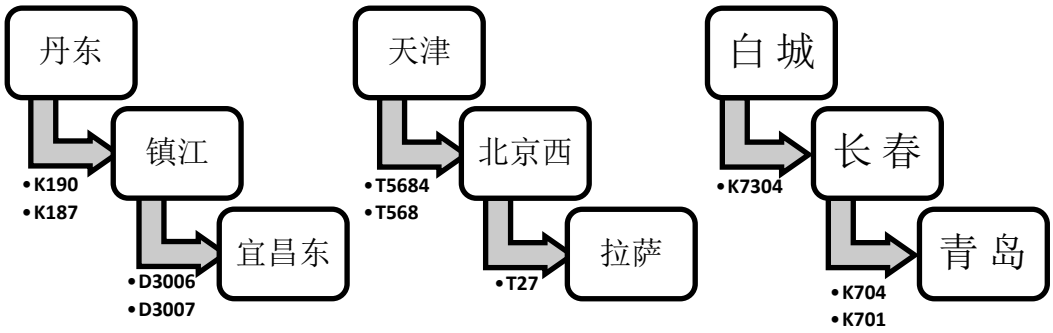
表 3：丹东→宜昌、天津→拉萨、白城→青岛最优路线

起点站	离开时间	车次	到达时间	换乘站	离开时间	车次	到达时间	终点站	所用时间
丹东	第 1 天 8:02	K190/ K187	第 2 天 12:45	镇江	第 1 天 15:35	D3006 D3007	第 1 天 21:42	宜昌东	37 小时 40 分
天津	第 1 天 13:46	T5684 /T568	第 1 天 15:38	北京西	第 1 天 20:00	T27	第 3 天 15:40	拉萨	49 小时 54 分
白城	第 1 天 10:23	K7304	第 1 天 15:19	长春	第 1 天 16:27	K704/ K701	第 2 天 15:19	青岛	28 小时 56 分

5.4 模型的结果分析

根据多目标线性规划模型，利用多目标优化算法，通过 MATLAB 编程得出结果如上表。通过结果可知：丹东→宜昌所选择的最优路线为：丹东→镇江→宜昌，分别乘坐的是 K190/K187 次列车和 D3006/D3007 次列车，全程耗时 37 小时 40 分钟；天津→拉萨所选择的最优路线为：天津→北京西→拉萨，分别乘坐 T5684/T568

次列车和 T27 次列车，全程耗时 49 小时 54 分钟；白城→青岛所选择的最优路线为白城→长春→青岛，分别乘坐 K7304 次列车和 K704/K701 次列车，全程耗时 28 小时 56 分钟。由所得结果可检验模型的正确性，证明所建的多目标线性规划模型与实际较为相符，能帮助出行者从若干个可能的方案中寻求某种意义下的最优方案。为了更加形象的表示换乘路线，制作流程图如下：



六：问题二模型的建立与求解

针对问题二：从宜昌出发乘坐火车至华东五市旅游后在返回宜昌，引进求解ＴＳＰ的模型，以每个城市只有一条到达的路线和一条离开的路线、每个城市没有任何子回路为约束条件；以整个行程乘车时间最短和乘车费用最小为目标函数建立多目标线性规划模型，运用模拟退火算法求解出整个旅游行程的最优路线。引进图论的知识，记 $G=(V,E)$ 为赋权图，$V=(1,2,...,6)$ 为顶点集，E 为边集，一个顶点到另外一个顶点的时间 t_{pq} 已知 $(t_{pq}>0, t_{pp}=\infty, p\in V, q\in V)$。

6.1 模型的建立

6.1.1 约束条件

（1）列车路线的约束

假设从城市 p 到城市 q 仅存在一条令游客满意度最大的线路，而其他路线的乘车时间和乘车费用都不是最优的。城市 p 到城市 q 的列车路线为 x_{pq}，如果游客在最优的路线上，则记 $x_{pq}=1$，如果在其他线路上，则记 $x_{pq}=0$。即：

$$x_{pq}=\begin{cases}1 & \text{从城市p到城市q的最佳线路}\\ 0 & \text{其他路线}\end{cases}$$

（2）到达、离开城市的路线的约束

为使旅游线路最优，那么到达每个城市 $V=(1,2,...,6)$ 旅游经过一次即可，所以每个城市只需要找到一条到达最优的线路和一条离开最优的线路。

即：

$$\sum_{p=1}^{6} x_{pq} = 1 \quad p \in V$$

$$\sum_{p=1}^{6} x_{pq} = 1 \quad q \in V$$

（3）子回路的约束

从宜昌出发乘坐火车至华东五市旅游后在返回宜昌，可看做 6 个城市，6 条火车路线，即每个城市不能有任何的子回路。

即：

$$\sum_{p \in s}^{p=1} \sum_{q \in s} x_{pq} \le |s| - 1 \quad \forall s \subset v, 2 \le |S| \le 5$$

（4）第一路段到达时间早于第二路段开车时间的约束

由于换乘过程中，第二路段到达终点的铁路路线可能过多，限制第一路段到达时间早于第二段开车时间。

$$\begin{cases} t_j < t_k < t_l \sin(n-1)\dfrac{\pi}{2} & (\text{换乘都在当天}) \\ t_j < t_k + 24(m-1) < (t_l + 24(m-1))\sin(n-1)\dfrac{\pi}{2} & (\text{换乘不在第一天}) \end{cases}$$

6.1.2 目标函数建立

在旅游的过程中，既要考虑旅游的两个城市之间乘车时间最短、所花的费用最少，也要考虑整个旅游行程乘车所花的时间最短和乘车所花费用最少。

（1）两个城市之间

考虑旅游的两个城市之间乘车时间最短$(\min t_{pq})$、所花费用最少$(\min W_{pq})$，两个城市之间乘车考虑到换乘次数 n $(n = 1,2)$，即：

$$\begin{cases} \min t_{pq} = t_{ij} + (2)^{n-1} \Delta t_n + t_{jk} + \sin(n-1)\dfrac{\pi}{2} t_{kl} \\ \min W_{pq} = \varpi_{ij} s_{ij} + \varpi_{jk} s_{jk} + \sin(n-1)\dfrac{\pi}{2} \varpi_{kl} s_{kl} \end{cases}$$

（2）六个城市之间

考虑整个旅游行程六个城市之间乘车所花的时间最短$(\min T)$、所花费用最少$(\min W)$。即：

$$\min T=\sum_{p=1}^{6}\sum_{q=1}^{6}\min\left(t_{pq}x_{pq}\right)$$

$$\min W=\sum_{p=1}^{6}\sum_{q=1}^{6}\min\left(w_{pq}x_{pq}\right)$$

6.1.3 综上所述，旅游的最优路线模型为：

$$\begin{cases}\min T=\sum_{p=1}^{6}\sum_{q=1}^{6}\min\left(t_{pq}x_{pq}\right)\\ \min W=\sum_{p=1}^{6}\sum_{q=1}^{6}\min\left(w_{pq}x_{pq}\right)\end{cases}$$

$$s.t.\begin{cases}\sum_{p=1}^{6}x_{pq}=1 \quad p\in v\\ \sum_{q=1}^{6}x_{pq}=1 \quad q\in v\\ \sum_{p\in s}^{p=1}\sum_{q\in s}x_{pq}\le|s|-1 \quad \forall s\subset v,2\le|S|\le 5\\ t_j<t_k+24(m-1)<\left(t_l+24(m-1)\right)\sin(n-1)\frac{\pi}{2}\\ t_j<t_k<t_l\sin(n-1)\frac{\pi}{2}\\ x_{pq}\in\{0,1\} \quad p\in v,q\in v\end{cases}$$

6.2 基于模拟退火算法求解模型

6.2.1 算法的准备

基于第一问模型与算法考虑换乘一次与换乘两次，求解出一个城市与另外 5 个城市的最佳线路，并算出最佳线路的乘车时间(T)与乘车费用(W)。其中应该注意的是从城市 p 到城市 q 的最佳线路与城市 q 到城市 p 的最佳线路是不同的，需要分开计算。

由于模拟退火算法只能求解单目标的模型最优解，所以根据以乘车时间最短、乘车费用最少的多目标线性规划模型，运用多目标评价函数的乘除法将多目标转化成单目标函数。由于多目标函数是求解乘车时间和乘车费用的最小值，所以定义一个函数为乘车时间与花费费用的乘积(TW)，再求解函数的最小值$(\min C)$，即可得到乘车时间与乘车费用的最小值。

$$\min C=TW=\left(\sum_{p=1}^{5}\sum_{q=1}^{5}\min\left(t_{pq}x_{pq}\right)\right)\left(\sum_{p=1}^{5}\sum_{q=1}^{5}\min\left(w_{pq}x_{pq}\right)\right)$$

6.2.1 算法的步骤

（1） 解空间与初始解

旅游问题的解空间 S 是遍历 6 个城市恰好一次的所有回路，是 6 个城市排列的集合。解空间 S 可表示为 $\{1,2,3,4,5,6\}$ 的所有排列的集合。即：

$$S=\left\{(c_1,c_2,c_3,c_4,c_5,c_6)\,|\,(c_1,c_2,c_3,c_4,c_5,c_6)\text{为}\{1,2,3,4,5,6\}\text{的循环排列}\right\}$$

其中，每一个排列 S_i 表示遍历 6 个城市的一个路径，$c_i=j$ 表示第 i 次访问城市 j，模拟退火算法的最优解和初始状态没有强的依赖关系，故初始解为随机函数生成一个 $\{1,2,3,4,5,6\}$ 的随机排列作为 S_0。

（2） 目标函数

旅游问题的目标函数即为访问所有城市的时间与费用的乘积 (TW)，也可称为代价函数，即：

$$\min C=TW=\left(\sum_{p=1}^{5}\sum_{q=1}^{5}\min\left(t_{pq}x_{pq}\right)\right)\left(\sum_{p=1}^{5}\sum_{q=1}^{5}\min\left(w_{pq}x_{pq}\right)\right)$$

通过模拟退火算法求出目标函数的最小值，相应的 (S) 即为旅游问题的最优解。即：

$$S=\left(c_1^*,c_2^*,c_3^*,c_4^*,c_5^*,c_6^*\right)$$

（3） 新解产生

新解的产生对问题非常重要。新解可通过分别或交替使用以下两种方法来产生：

1. 二变换法

任选序号 $u,v(u<v<6)$，交换 u 和 v 之间的访问顺序。例如：原路径为 $\{c_1,c_2,c_3,c_4,c_5,c_6\}$,选择序号 3,4。变换后的新路径 $\{c_1,c_2,c_4,c_3,c_5,c_6\}$

2. 三变换法

任选序号 $u,v,w(\mathrm{u}\le\mathrm{v}<\mathrm{w})$，将 u 和 v 之间的路径插到 w 之后访问。例如：原路径为 $\{c_1,c_2,c_3,c_4,c_5,c_6\}$，选择序号 2,3,4。变换后的新路径为 $\{c_1,c_4,c_2,c_3,c_5,c_6\}$。

（4） 目标函数差

计算变换前的解 $C\left(s_i^{'}\right)$ 和变换后的解 $C(s_i)$ 目标函数的差值 $\Delta C^{'}$，即：

$$\Delta C^{'}=C\left(s_i^{'}\right)-C(s_i)$$

（5） Metropolis 接受准则

以新解与当前解的目标函数差定义接受概率 P，即：

$$P=\begin{cases}1 & \Delta C^{'}<0\\ \exp\left(-\Delta\mathrm{C}^{'}/T\right) & \Delta C^{'}>0\end{cases}$$

若 $\Delta C^{'}<0$，则接受新路径。否则，以概率 $\exp\left(-\Delta C^{'}/T\right)$ 接受新路径，$\exp\left(-\Delta C^{'}/T\right)$ 在大于 0 小于 1 之间的随机数则可接受新路径。

（6） 降温

利用选定的系数进行降温。$T_{k+1}=\alpha T_k\left(k=1,2,3,4,5\right)$，得到新的温度。由于 α 可以取 0.5 到 0.99 之间的任何实数，所以在这里取 $\alpha=0.9$。

（7） 结束条件

用选定的终止温度 e, 判断退火是否结束。若 T<e, 输出当前状态。

6.3 模型的求解

利用问题一的模型和算法，运用 MATLAB 编程求解出其中一个城市到另外 5 个城市的路线。并在其中的路线中找出两个城市之间乘车时间和乘车费用，令游客满意度最大旅游路线。再利用模拟退火算法求解出从宜昌乘火车到上海、南京、杭州、苏州、无锡旅游最后回到宜昌的最佳旅游路线（见附录 2），得出的结果如表 4。

表 4：宜昌到华东五区的最优路线

起点站	离开时间	车次	到达时间	终点站	所用时间
宜昌东	14:40	G588/G585	20:26	南京南	5 小时 46 分
南京南	21:08	G21	21:52	无锡东	0 小时 44 分
无锡东	21:54	G21	22:04	苏州北	0 小时 10 分
苏州	16:39	G7583	18:08	杭州东	1 小时 29 分
杭州东	15:25	G7516	16:10	上海虹桥	0 小时 45 分
上海虹桥	13:54	D3006/D3007	21：42	宜昌东	7 小时 48 分

6.4 模型的结果分析

利用模拟退火算法求解得到从宜昌到华东五市旅游的最佳旅游路线为：宜昌东→南京南→无锡东→苏州北→杭州东→上海虹桥→宜昌东，解得的最优旅游路线符合此问所建立的多目标线性规划模型及基于模拟退火算法的求解模型，充分验证了本模型的正确性。最佳旅游路线符合实际中的旅行路线，大大减少了游客时间和费用的损耗，在时间最少，费用最低的情况下可以到达每一个城市游玩，且不会重复到达同一个城市。最佳旅游路线图如图 1。

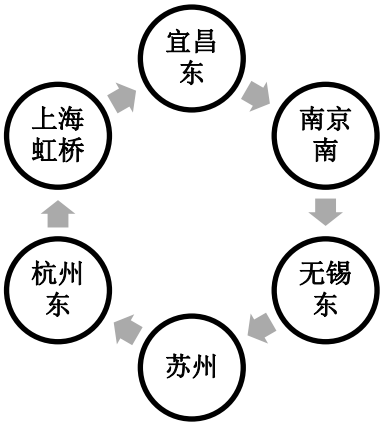


图 1：最佳旅游路线

七：模型的评价与改进

7.1 模型的评价

（1） 优点

1. 综合考虑了出行者选择路线时换乘次数、出行耗时以及乘车费用的影响，针对不同的需求建立多目标线性规划模型。
2. 考虑到出行者对换乘次数有一个最大的承受上限，所以限定最大换乘次数为 2，即最多利用 3 条火车路线从起点到达终点站。
3. 第二问中运用模拟退火算法求解模型，与其他算法相比,模拟退火算法描述简单、使用灵活、运用广泛、运行效率高。

（2） 缺点

1. 在模型一种由于只限定了最大换乘次数，单次换乘时间间隔，换乘时间间隔总和，所输出的铁路路线具有一定的局限性，应考虑更多影响因素，如出行者乘车的疲劳度，两站点之间最短距离等。
2. 运用模拟退火算法求解模型时收敛速度较慢，执行时间较长。

7.2 模型的改进

在模型一中，仅建立了以换乘时间最短和乘车费用最少为目标函数的多目标线性规划模型，为了得到更加优化的铁路路线，应增加目标函数得出另出行者更加满意的路线方案。

在基于模拟退火算法求解模型中，从宜昌出发去五个城市旅游，换乘过程中，仅选择了第一路段下车后 60 分钟内到达下一个城市的最优路线，实际中应考虑到达一个城市游玩的时间。

7.3 模型的推广

本文所建立的基于模拟退火算法求解模型不仅适用铁路最优路线问题，还可以较高效率求解最大截问题、旅行商问题、调度问题、函数优化问题等。

八：参考文献

【1】 丁向阳，《构建国际一流旅游城市，迈出向“世界城市”行进的第一步》，第十届世界旅游旅行大会论坛，《中国旅游产业》媒体事业部，2010 年 6 月 20

【2】 冯剑，岳琪，模拟退火算法求解 TSP 问题。

【3】 王朝晖，杨洁，公交线路中的最优路线的查询算法与设计。

【4】 石季英，李红艳，曹明增，马有明，物流中心选址与动态运输线路选择的研究。

九：附录

附录一

```
clc
clear
[a,b,c]=xlsread('车站');
ycz=c(:,1);
ycc=c(:,2);
%% 去掉重复的车站
cz=unique(ycz);
cc=unique(ycc);
%% 给车站以及车次编号
czh=zeros(length(cz),1);
cch=zeros(length(cc),1);
for i=1:length(cz)
    czh(i,1)=i;
end
for i=1:length(cc)
    cch(i,1)=i;
end
%% 将未处理的原始数据用编号代替
for i=1:length(cz)
    ycz(strcmp(ycz,cz{i}))={i};
end
for i=1:length(cc)
    ycc(strcmp(ycc,cc{i}))={i};
end
ycc=cell2mat(ycc);
ycz=cell2mat(ycz);
%% 计算火车的线路
lujin=zeros(length(cch),65);
for i=1:length(cch)
    lujin(i,1:length(ycz(find(ycc==i))))=ycz(find(ycc==i));
```

```
end
%% 重新排列时刻表
ddsk=zeros(length(cch),65);
lksk=zeros(length(cch),65);
sk1=a(:,1);
sk2=a(:,2);
for i=1:length(cch)
    ddsk(i,1:length(ycz(find(ycc==i))))=sk1(find(ycc==i));
end
for i=1:length(cch)
    lksk(i,1:length(ycz(find(ycc==i))))=sk2(find(ycc==i));
end
%% 寻找所给的城市的车站
A1=ismember(cz,'丹东');
A1=find(A1==1);
A2=ismember(cz,'宜昌东');
A2=find(A2==1);
B11=ismember(cz,'天津');
B11=find(B11==1);
B12=ismember(cz,'天津南');
B12=find(B12==1);
B13=ismember(cz,'天津西');
B13=find(B13==1);
B14=ismember(cz,'天津北');
B14=find(B14==1);
B2=ismember(cz,'拉萨');
B2=find(B2==1);
D1=ismember(cz,'白城');
D1=find(D1==1);
D2=ismember(cz,'青岛');
D2=find(D2==1);
```

```
%% 计算换乘一次的方案
%% 丹东到宜昌的路线
[c1,z1]=find(lujin==A1);
[c2,z2]=find(lujin==A2);
s=1;
for i=1:length(c1)
    for j=1:length(c2)
        for k=1:65
            for l=1:65

if(lujin(c1(i),k)==lujin(c2(j),l)&&lujin(c1(i),k)~=0&&lujin(c2(j),l)~=0&&z1(i)<k&&
z2(j)>l);
                A1sk(s,1)=lksk(c1(i),z1(i));
                a1(s,1)=c1(i);
                A2sk(s,1)=ddsk(c1(i),k);
                A3(s,1)=lujin(c1(i),k);
                A3sk(s,1)=lksk(c2(j),l);
                a2(s,1)=c2(j);
                A4sk(s,1)=ddsk(c2(j),z2(j));
                s=s+1;
            end
        end
    end
end
% 得出路线
for s=1:length(A3)

luxian1(s,:)=[cz(A1),A1sk(s),cc(a1(s)),A2sk(s),cz(A3(s)),A3sk(s),cc(a2(s)),A4sk(s),c
z(A2)];
end
```

```
% 计算花费时间最短的路线
for s=1:length(luxian1)
    t1=A2sk(s)-A1sk(s);
    t3=A4sk(s)-A3sk(s);
    if(A2sk(s)-floor(A2sk(s))>A3sk(s)-floor(A3sk(s))-1/24)
        t2=A3sk(s)-floor(A3sk(s))+1-(A2sk(s)-floor(A2sk(s)));
    else
        t2=A3sk(s)-floor(A3sk(s))-(A2sk(s)-floor(A2sk(s)));
    end
    luxian1(s,10)={t1+t2+t3};
end
[x1,y1]=min(cell2mat(luxian1(:,10)));
x1=luxian1(y1,:)
%% 天津到拉萨的路线
[c11,z11]=find(lujin==B11);
[c12,z12]=find(lujin==B12);
[c13,z13]=find(lujin==B13);
[c14,z14]=find(lujin==B14);
c1=[c11;c12;c13;c14];
z1=[z11;z12;z13;z14];
[c2,z2]=find(lujin==B2);
s=1;
for i=1:length(c1)
    for j=1:length(c2)
        for k=1:65
            for l=1:65

if(lujin(c1(i),k)==lujin(c2(j),l)&&lujin(c1(i),k)~=0&&lujin(c2(j),l)~=0&&z1(i)<k&&
z2(j)>l);
                    B1(s,1)=lujin(c1(i),z1(i));
                    B1sk(s,1)=lksk(c1(i),z1(i));
```

```
                    b1(s,1)=c1(i);
                    B2sk(s,1)=ddsk(c1(i),k);
                    B3(s,1)=lujin(c1(i),k);
                    B3sk(s,1)=lksk(c2(j),l);
                    b2(s,1)=c2(j);
                    B4sk(s,1)=ddsk(c2(j),z2(j));
                    s=s+1;
                end
            end
        end
    end
    % 得出路线
    for s=1:length(B3)

luxian2(s,:)=[cz(B1(s)),B1sk(s),cc(b1(s)),B2sk(s),cz(B3(s)),B3sk(s),cc(b2(s)),B4sk(s)
,cz(B2)];
    end
    % 计算花费时间最短的路线
    for s=1:length(luxian2)
        t1=B2sk(s)-B1sk(s);
        t3=B4sk(s)-B3sk(s);
        if(B2sk(s)-floor(B2sk(s))>B3sk(s)-floor(B3sk(s))-1/24)
            t2=B3sk(s)-floor(B3sk(s))+1-(B2sk(s)-floor(B2sk(s)));
        else
            t2=B3sk(s)-floor(B3sk(s))-(B2sk(s)-floor(B2sk(s)));
        end
        luxian2(s,10)={t1+t2+t3};
    end
    [x2,y2]=min(cell2mat(luxian2(:,10)));
    x2=luxian2(y2,:)
```

```
%% 白城到青岛的路线
[c1,z1]=find(lujin==D1);
[c2,z2]=find(lujin==D2);
s=1;
for i=1:length(c1)
    for j=1:length(c2)
        for k=1:65
            for l=1:65

if(lujin(c1(i),k)==lujin(c2(j),l)&&lujin(c1(i),k)~=0&&lujin(c2(j),l)~=0&&z1(i)<k&&
z2(j)>l);
                                D1sk(s,1)=lksk(c1(i),z1(i));
                                d1(s,1)=c1(i);
                                D2sk(s,1)=ddsk(c1(i),k);
                                D3(s,1)=lujin(c1(i),k);
                                D3sk(s,1)=lksk(c2(j),l);
                                d2(s,1)=c2(j);
                                D4sk(s,1)=ddsk(c2(j),z2(j));
                                s=s+1;
                        end
                    end
                end
            end
        end
    end
    % 得出路线
    for s=1:length(D3)

luxian3(s,:)=[cz(D1),D1sk(s),cc(d1(s)),D2sk(s),cz(D3(s)),D3sk(s),cc(d2(s)),D4sk(s),c
z(D2)];
    end
    % 计算花费时间最短的路线
```

```
for s=1:length(luxian3)
    t1=D2sk(s)-D1sk(s);
    t3=D4sk(s)-D3sk(s);
    if(D2sk(s)-floor(D2sk(s))>D3sk(s)-floor(D3sk(s))-1/24)
        t2=D3sk(s)-floor(D3sk(s))+1-(D2sk(s)-floor(D2sk(s)));
    else
        t2=D3sk(s)-floor(D3sk(s))-(D2sk(s)-floor(D2sk(s)));
    end
    luxian3(s,10)={t1+t2+t3};
end
[x3,y3]=min(cell2mat(luxian3(:,10)));
x3=luxian3(y3,:)
```

附录二

```
clc
clear
[a,b,c]=xlsread('车站');
ycz=c(:,1);
ycc=c(:,2);
%% 去掉重复的车站
cz=unique(ycz);
cc=unique(ycc);
%% 给车站以及车次编号
czh=zeros(length(cz),1);
cch=zeros(length(cc),1);
for i=1:length(cz)
    czh(i,1)=i;
end
for i=1:length(cc)
    cch(i,1)=i;
end
%% 将未处理的原始数据用编号代替
```

```
for i=1:length(cz)
    ycz(strcmp(ycz,cz{i}))={i};
end
for i=1:length(cc)
    ycc(strcmp(ycc,cc{i}))={i};
end
ycc=cell2mat(ycc);
ycz=cell2mat(ycz);
%% 计算火车的线路
lujin=zeros(length(cch),65);
for i=1:length(cch)
    lujin(i,1:length(ycz(find(ycc==i))))=ycz(find(ycc==i));
end
%% 重新排列时刻表
ddsk=zeros(length(cch),65);
lksk=zeros(length(cch),65);
sk1=a(:,1);
sk2=a(:,2);
for i=1:length(cch)
    ddsk(i,1:length(ycz(find(ycc==i))))=sk1(find(ycc==i));
end
for i=1:length(cch)
    lksk(i,1:length(ycz(find(ycc==i))))=sk2(find(ycc==i));
end
%% 寻找所给的城市的车站
A=ismember(cz,'宜昌东');
A=find(A==1);
B1=ismember(cz,'南京');
B1=find(B1==1);
B2=ismember(cz,'南京南');
B2=find(B2==1);
```

```
C1=ismember(cz,'无锡');
C1=find(C1==1);
C2=ismember(cz,'无锡东');
C2=find(C2==1);
C3=ismember(cz,'无锡新区');
C3=find(C3==1);
D1=ismember(cz,'苏州');
D1=find(D1==1);
D2=ismember(cz,'苏州北');
D2=find(D2==1);
D3=ismember(cz,'苏州园区');
D3=find(D3==1);
D4=ismember(cz,'苏州新区');
D4=find(D4==1);
E1=ismember(cz,'杭州');
E1=find(E1==1);
E2=ismember(cz,'杭州东');
E2=find(E2==1);
F1=ismember(cz,'上海');
F1=find(F1==1);
F2=ismember(cz,'上海南');
F2=find(F2==1);
F3=ismember(cz,'上海西');
F3=find(F3==1);
F4=ismember(cz,'上海虹桥');
F4=find(F4==1);
%% 计算换乘一次的方案
%% 宜昌到南京的路线
[c1,z1]=find(lujin==A);
[c21,z21]=find(lujin==B1);
[c22,z22]=find(lujin==B2);
```

```
c2=[c21;c22];
z2=[z21;z22];
s=1;
for i=1:length(c1)
    for j=1:length(c2)
        for k=1:65
            for l=1:65

if(c1(i)==c2(j)&&lujin(c1(i),k)~=0&&lujin(c2(j),l)~=0&&z1(i)<z2(j));
                    sk1(s,1)=lksk(c1(i),z1(i));
                    a1(s,1)=c1(i);
                    sk2(s,1)=ddsk(c2(j),z2(j));
                    B(s,1)=lujin(c2(j),z2(j));
                    s=s+1;
                end
            end
        end
    end
end
% 得出路线
for s=1:length(B)
    luxian1(s,:)=[cz(A),sk1(s),cc(a1(s)),sk2(s),cz(B(s))];
end
% 计算花费时间最短的路线
for s=1:length(luxian1)
    t1=sk2(s)-sk1(s);
    luxian1(s,6)={t1};
end
[x1,y]=min(cell2mat(luxian1(:,6)));
x1=luxian1(y,:)
%% 南京到无锡的路线
```

```
[c21,z21]=find(lujin==B1);
[c22,z22]=find(lujin==B2);
c2=[c21;c22];
z2=[z21;z22];
[c31,z31]=find(lujin==C1);
[c32,z32]=find(lujin==C2);
[c33,z33]=find(lujin==C3);
c3=[c31;c32;c33];
z3=[z31;z32;z33];
s=1;
for i=1:length(c2)
    for j=1:length(c3)
        for k=1:65
            for l=1:65

if(c2(i)==c3(j)&&lujin(c2(i),k)~=0&&lujin(c3(j),l)~=0&&z2(i)<z3(j));
                    B(s,1)=lujin(c2(i),z2(i));
                    sk1(s,1)=lksk(c2(i),z2(i));
                    a1(s,1)=c2(i);
                    sk2(s,1)=ddsk(c3(j),z3(j));
                    C(s,1)=lujin(c3(j),z3(j));
                    s=s+1;
                end
            end
        end
    end
end
% 得出路线
for s=1:length(C)
    luxian2(s,:)=[cz(B(s)),sk1(s),cc(a1(s)),sk2(s),cz(C(s))];
end
```

```
% 计算花费时间最短的路线
for s=1:length(luxian2)
    t1=sk2(s)-sk1(s);
    luxian2(s,6)={t1};
end
[x2,y]=min(cell2mat(luxian2(:,6)));
x2=luxian2(y,:)
%% 无锡到苏州的路线
[c31,z31]=find(lujin==C1);
[c32,z32]=find(lujin==C2);
[c33,z33]=find(lujin==C3);
c3=[c31;c32;c33];
z3=[z31;z32;z33];
[c41,z41]=find(lujin==D1);
[c42,z42]=find(lujin==D2);
[c43,z43]=find(lujin==D3);
[c44,z44]=find(lujin==D4);
c4=[c41;c42;c43;c44];
z4=[z41;z42;z43;z44];
s=1;
for i=1:length(c3)
    for j=1:length(c4)
        for k=1:65
            for l=1:65

if(c3(i)==c4(j)&&lujin(c3(i),k)~=0&&lujin(c4(j),l)~=0&&z3(i)<z4(j));
                C(s,1)=lujin(c3(i),z3(i));
                sk1(s,1)=lksk(c3(i),z3(i));
                a1(s,1)=c3(i);
                sk2(s,1)=ddsk(c4(j),z4(j));
                D(s,1)=lujin(c4(j),z4(j));
```

```
                    s=s+1;
                end
            end
        end
    end
end
% 得出路线
for s=1:length(D)
    luxian3(s,:)=[cz(C(s)),sk1(s),cc(a1(s)),sk2(s),cz(D(s))];
end
% 计算花费时间最短的路线
for s=1:length(luxian3)
    t1=sk2(s)-sk1(s);
    luxian3(s,6)={t1};
end
[x3,y]=min(cell2mat(luxian3(:,6)));
x3=luxian3(y,:)
%% 苏州到杭州的路线
[c41,z41]=find(lujin==D1);
[c42,z42]=find(lujin==D2);
[c43,z43]=find(lujin==D3);
[c44,z44]=find(lujin==D4);
c4=[c41;c42;c43;c44];
z4=[z41;z42;z43;z44];
[c51,z51]=find(lujin==E1);
[c52,z52]=find(lujin==E2);
c5=[c51;c52];
z5=[z51;z52];
s=1;
for i=1:length(c4)
    for j=1:length(c5)
```

```
                for k=1:65
                    for l=1:65

if(c4(i)==c5(j)&&lujin(c4(i),k)~=0&&lujin(c5(j),l)~=0&&z4(i)<z5(j));
                                D(s,1)=lujin(c4(i),z4(i));
                                sk1(s,1)=lksk(c4(i),z4(i));
                                a1(s,1)=c4(i);
                                sk2(s,1)=ddsk(c5(j),z5(j));
                                E(s,1)=lujin(c5(j),z5(j));
                                s=s+1;
                            end
                        end
                    end
                end
            end
        end
        % 得出路线
        for s=1:length(E)
            luxian4(s,:)=[cz(D(s)),sk1(s),cc(a1(s)),sk2(s),cz(E(s))];
        end
        % 计算花费时间最短的路线
        for s=1:length(luxian4)
            t1=sk2(s)-sk1(s);
            luxian4(s,6)={t1};
        end
        [x4,y]=min(cell2mat(luxian4(:,6)));
        x4=luxian4(y,:)
        %% 杭州到上海的路线
        [c51,z51]=find(lujin==E1);
        [c52,z52]=find(lujin==E2);
        c5=[c51;c52];
        z5=[z51;z52];
```

```
[c61,z61]=find(lujin==F1);
[c62,z62]=find(lujin==F2);
[c63,z63]=find(lujin==F3);
[c64,z64]=find(lujin==F4);
c6=[c61;c62;c63;c64];
z6=[z61;z62;z63;z64];
s=1;
for i=1:length(c5)
    for j=1:length(c6)
        for k=1:65
            for l=1:65

if(c5(i)==c6(j)&&lujin(c5(i),k)~=0&&lujin(c6(j),l)~=0&&z5(i)<z6(j));
                        E(s,1)=lujin(c5(i),z5(i));
                        sk1(s,1)=lksk(c5(i),z5(i));
                        a1(s,1)=c5(i);
                        sk2(s,1)=ddsk(c6(j),z6(j));
                        F(s,1)=lujin(c6(j),z6(j));
                        s=s+1;
                    end
                end
            end
        end
    end
    % 得出路线
    for s=1:length(F)
        luxian5(s,:)=[cz(E(s)),sk1(s),cc(a1(s)),sk2(s),cz(F(s))];
    end
    % 计算花费时间最短的路线
    for s=1:length(luxian5)
        t1=sk2(s)-sk1(s);
```

```
        luxian5(s,6)={t1};
    end
    [x5,y]=min(cell2mat(luxian5(:,6)));
    x5=luxian5(y,:)
    %% 上海到宜昌的路线
    [c61,z61]=find(lujin==F1);
    [c62,z62]=find(lujin==F2);
    [c63,z63]=find(lujin==F3);
    [c64,z64]=find(lujin==F4);
    c6=[c61;c62;c63;c64];
    z6=[z61;z62;z63;z64];
    [c1,z1]=find(lujin==A);
    s=1;
    for i=1:length(c6)
        for j=1:length(c1)
            for k=1:65
                for l=1:65

if(c6(i)==c1(j)&&lujin(c6(i),k)~=0&&lujin(c1(j),l)~=0&&z6(i)<z1(j));
                                F(s,1)=lujin(c6(i),z6(i));
                                sk1(s,1)=lksk(c6(i),z6(i));
                                a1(s,1)=c6(i);
                                sk2(s,1)=ddsk(c1(j),z1(j));
                                s=s+1;
                            end
                        end
                    end
                end
            end
        end
    % 得出路线
    for n=1:s-1
```

```
        luxian6(n,:)=[cz(F(n)),sk1(n),cc(a1(n)),sk2(n),cz(A)];
end
% 计算花费时间最短的路线
for s=1:length(luxian6)
        t1=sk2(s)-sk1(s);
        luxian6(s,6)={t1};
end
[x6,y]=min(cell2mat(luxian6(:,6)));
x6=luxian6(y,:)
```