

# 公交线路最佳乘车方案设计

## 摘要

本题主要研究在三种不同的情况下，给出任意两个站点之间的路线规划问题。联系实际，公众需要乘坐公交车考虑的主要因素包括转乘次数、路程时间和票价。为满足公众对不同因素的需求，我们按照公众对不同乘车影响因素的考虑程度给出基于各因素下的最优乘车路线。在考虑影响因素时结合实际，考虑换乘次数一般小于等于 2。

**针对问题一：**在只考虑公汽路线的情况下，首先需要对题目中所给的公汽路线数据信息做处理，共有三种运行方式的公交，我们将上下行不同、双向行驶以及环形路线抽象为同一种数据类型，给出每条线路的上下行统一处理。然后主要考虑换乘次数因素，以换乘次数为基准求出各个换乘次数的最优路径，然后在此基础上考虑公众对其他因素的需求，按照先后顺序考虑行程时间、票价因素。并最终给出基于个人需求的赋权综合最优路线。

**针对问题二：**问题二在上一问的基础上加以考虑地铁路线，通常情况下一个地铁站等同于 1-5 个公汽车站，为了规划路线时完全的显示一个站点可换乘的站点信息，我们建立元胞存储数据信息，相当于将每个地铁站和对应的公汽车站抽象为一个集合处理。对加入地铁线路的规划，可按照与问题一相同的模型规划与选择标准处理，从而确定出任意两个站点之间的最优路径。

**针对问题三：**第三问考虑公交和地铁站点的实际分布情况，有时候会出现步行小距离再乘车更节省时间/票价的情况，所以加入步行这个缓冲因素考虑。结合实际一般步行站数大于 2 就会考虑乘车，因此我们只考虑步行 1 站换乘的问题。由于站点的地理相对位置没有给出，在某个站点时我们只考虑公汽线路上的步行。在符合情况的所有路线中找出最优路线即可。

**关键词：**元胞队列信息 集合交叉寻找换乘思想 赋权综合最优 公交线路规划问题

## 一、问题重述

第 29 届奥运会明年 8 月将在北京举行，届时有大量观众到现场观看奥运比赛，其中大部分人将会乘坐公共交通工具（简称公交，包括公汽、地铁等）出行。随着最近这几年的发展，公交线网络繁多、公交出行越来越方便快捷，但同时也面临多条线路的选择问题。针对市场需求，某公司准备研制开发一个解决公交线路选择问题的自主查询计算机系统。

设计这样一个系统需要注意：

一、线路选择的模型与算法必须利于程序的实现；

二、应该从实际情况出发考虑，满足查询者的各种不同需求。

请你们解决如下问题：

1. 仅考虑公汽线路，给出任意两公汽站点之间线路选择问题的一般数学模型与算法。并根据附录数据，利用你们的模型与算法，求出以下 6 对起始站→终到站之间的最佳路线（要有清晰的评价说明）。

(1). S3359→S1828      (2). S1557→S0481      (3). S0971→S0485

(4). S0008→S0073      (5). S0148→S0485      (6). S0087→S3676

2. 同时考虑公汽与地铁线路，解决以上问题。

3. 假设又知道所有站点之间的步行时间，请你给出任意两站点之间线路选择问题的数学模型。

## 二、问题分析

### 2.1 规划路线时考虑的因素

本题主要研究在三种不同的情况下，给出任意两个站点之间的路线规划问题。联系实际，公众需要乘坐公交车主要需要考虑转乘次数、路程时间和票价三个主要因素。为满足对不同因素需求不同的人，我们按照公众对影响乘车的不同因素的考虑情况给出基于各因素下的最优乘车路线。

换乘次数在一定程度上与路程所用时间和票价成正比，换乘需要等车且增加乘车费用，所以以换乘次数为首要划分标准，并且结合人的心理因素，换成次数一般小于等于 2 次。求最短路径一般使用 *Dijkstra* 最短路径算法，但在此问题结合实际考虑因素过于复杂，该算法不好实现，所以我们简化解思想，在以下解决问题时我们以换乘次数为划分标准分别，在路线所用时间和票价上给予分别的最优路线。

### 2.2 问题的具体分析

(1) 在只考虑公汽路线的情况下，求解给定起始站和终点站的公汽规划路线。首先需要对题目中所给的 520 组公汽路线数据信息做处理，共有三种运行方式的公交，我们将上下行不同、上下行双向行驶以及环形路线抽象为同一种数据类型，给出每条线路的上下行统一处理，归为元胞数据类标准形式。在影响公众乘车路径选择的因素中主要考虑换乘次数因素，以换乘次数为基准求出各个换乘次数的最优路径，然后在此基础上考虑公众对其他因素的需求，按照先后顺序考虑行程时间、票价因素。并最终给出基于个人需求的赋权综合最优路线。

(2) 问题二在上一问的基础上加以考虑地铁路线，通常情况下一个地铁站等价于1-5个公汽站点，为了规划路线时完全的显示考虑一个地铁站所代表的公汽车站站点信息，我们建立元胞存储线路及站点数据，相当于将每个地铁站和对应的公汽车站抽象为一个集合处理。对加入地铁线路的规划，可按照与问题一相同的规划与选择标准处理，从而确定出任意两个站点之间的最优路径。

(3) 第三问考虑公交和地铁站点的实际分布情况，有时候会出现步行小距离来使总路线更优的情况，所以加入步行这个因素考虑。结合公众心理，一般步行站数大于2就会考虑乘车，所以我们只考虑步行1站来优化路线答案。由于未知站点的地理相对位置，在某个站点时我们只考虑公汽线路上的步行。在符合情况的所有路线中找出最优路线即可。

## 2.3 问题的解决层次

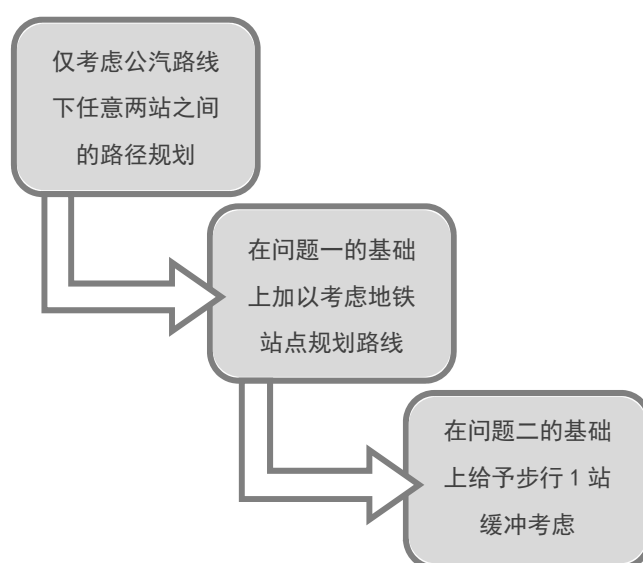


图 1 问题逻辑思维导图

## 三、模型假设

(1) 在公汽线路的环线上乘车经过起始站/终到站不另外收费，环道两个方向都可以同行，且任意站点都可上车。

(2) 将环线视为起始站、终点站相同且上行线与下行线相同的特殊线路。

(3) 在计算时间时，等车时间不给予考虑，实际生活中是一个不可控因素。

(4) 不考虑不可控因素对路线选择的影响，如：天气，行李重量等。

(5) 多数人们至多只能接受转两次车，考虑到需要导航的用户不熟悉地理位置，换乘过多可能找不到的问题，并且如果不限次数则有无数条线路可到达。

(6) 问题一中只考虑同站换车，因为站点的相对位置未知。

## 四、符号说明

$w_i$	费用
$P$	交通工具
$x_i$	第 <i>i</i> 段线路的公交站数
$y_i$	第 <i>i</i> 段线路地铁总站数
$Z_1$	地铁换乘地铁次数
$Z_2$	公交换乘公交次数
$Z_3$	公交换乘地铁次数
$Z_4$	地铁换乘公交次数
$Z_5$	步行换乘公交乘车次数
$Z_6$	步行转乘地铁次数

## 五、模型的建立与求解

### 5.1 数据预处理

#### (1) 对公交站点数据处理

在建立站点集合的时候，数据格式的选择对搜索路线尤为重要。题目中共有520条线路，每条线路包含10–40左右的站点，若使用字符存储，占用空间大且寻找时计算机运行速度慢，所以对题目站点为 $S +$ 开头的数字编号站点，我们利用 $MATLAB$ 将数据导入后生成元胞格式的数据，元胞中队列的每一行代表一辆车的线路信息，。

	1	2	3	
1	'L001'	2	'来回1'	'S0619-S1914-S0388-S0348-S
2	'L001'	2	'来回2'	'S0619-S1914-S0388-S0348-S
3	'L002'	2	'上行'	'S3748-S2160-S1223-S1404-S
4	'L002'	2	'下行'	'S0004-S1968-S0012-S2636-S
5	'L003'	1	'来回1'	'S0417-S0272-S1973-S3425-S
6	'L003'	1	'来回2'	'S0417-S0272-S1973-S3425-S
7	'L004'	1	'上行'	'S3010-S0582-S0579-S1994-S
8	'L004'	1	'下行'	'S0679-S2184-S2182-S0992-S
9	'L005'	2	'上行'	'S2817-S1961-S3171-S0925-S
10	'L005'	2	'下行'	'S1236-S2086-S2038-S3264-S
11	'L006'	2	'下行'	'S0003-S0077-S1110-S3403-S

图2 元胞内部结构示意图

#### (2) 对三种公交运行线路的统一表示

[1]上行线、下行线都有且不一致

由于上行线和下行线都存在且经过站点不完全相同，显然该种情况为最一般的情况，需要抽象为两条线路表达。

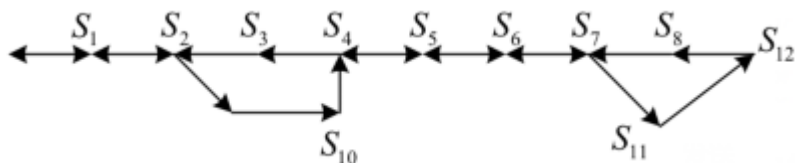


图3 公交线路示意图

[2]上行线、下行线一致

这种线路本质为一条路，下行是上行线的反方向。我们根据公交的运行方向将之一般化抽象为两条线路。

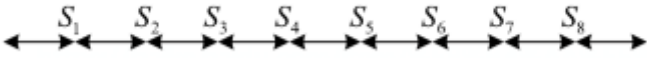


图 4 公交线路示意图

[3]环形线

在查阅资料后我们得知环线一般分为内环、外环，并且会固定一个站点作为起始站，在该指定站点反方向则为下行。所以我们处理环线时选择一个站点切开，转化为上行线、下行线一致的路线，最后将之抽象为一般化表示。

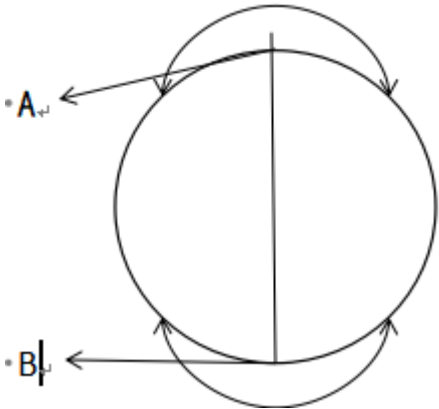


图 5 公交线路环道示意图

如图弧弧道  $A \rightarrow B$  为上行，弧  $B \rightarrow A$  为环道下行线。

5.2 问题一的模型

5.2.1 问题一模型的建立

由问题分析可知，需要建立模型解决在各个转乘次数指标之下的乘车影响因素最优路线，结合实际考虑人的心理因素不考虑太高次数的换乘。所以我们针对 0、1、2 次换乘给出各组的最优路线，进一步给出时间最短与费用最低的路径，最后定义赋权综合路线指标，用户可根据个人偏好选择最优规划路线。

目标分析

目标一：换乘次数

(1) 直达

在题目所给的所有公汽路线中进行遍历搜索，若起始站和终点站属于同一条公汽线路，则该条为最优路径。

(2) 一次换乘

图中  $A$  为起始站， $B$  为终点站， $T$  为中转站。当直达路线不存在时，我们要考虑公汽进行换乘，对此我们只考虑同站路线换乘，即下车在该站就只需等待换乘的车辆。我们将所有通过起始站  $A$  的线路号索引出，将之经过的站台点记作集合  $\alpha$ ，同理将经过终点站的线路站台点记作集合  $\beta$ ，若两趟公交可以同站换乘，说明集合  $\alpha \cap \beta$  非空，二者的交集即为连通两条线路的可用中转站集合，从该集合可以得到能进行换乘的路线，可以进一步选择次数限制下的最优路线。

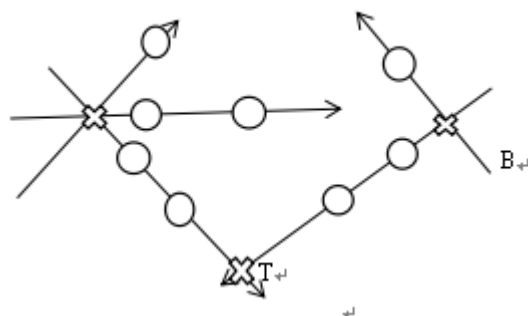


图 6 换乘情况示意图

### (3) 两次换乘

当上述两种情况都无结果时，我们考虑两次换乘。找出经过起始站  $A$  与终点站  $B$  的所有线路，将之即为集合  $\alpha$ 、 $\beta$ ，在此基础上寻找二者的交集。若可进行二次换乘，则经过起始站与终点站的路线上必存在相关线路作为中转线路。我们对交集的点进行遍历，得到可能的中转线路，进一步找出经过起始站与终点站的线路，则完整找出了两次换乘的线路。

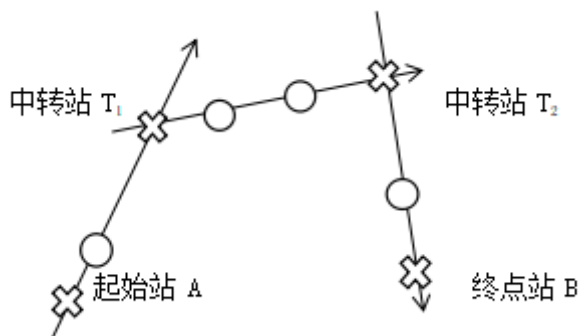


图 7 换乘情况示意图

### 目标二：行程时间

题目中给出换乘时间以及公交一站需要的时间，我们根据计算出的线路算出最优路线所耗费的时间。

公汽换乘时间固定为五分钟，则换乘时间为权值乘换乘次数：

$$T_1 = 5 * z_i$$

开车中的行程时间包括起点站需要等待的三分钟以及路上时间，为：

$$T_2 = 3 \sum_{i=1}^3 x_i$$

则行程总时间为：

$$T = 3 \sum_{i=1}^3 x_i + 5 z_i$$

### 目标三：行程费用

交通方式：

$$P = \begin{cases} 0 & \text{统一票价公交} \\ 1 & \text{分段计价公交} \\ 2 & \text{地铁} \end{cases}$$

各种交通方式以及价钱表示：

$$w_i = \begin{cases} 1 & P = 0 \\ P^j = 1 \text{ 且 } \begin{cases} 1 & j \in (0,20] \\ 2 & j \in (20,40] \\ 3 & j \in (40,+\infty] \end{cases} & \\ 3 & P = 2 \end{cases}$$

则总价格函数为：

$$w = \sum_{i=1}^3 w_i$$

目标四：赋权综合最优路线

首先我们考虑到实际情况下，人们在乘坐公交选择路线时往往是着重某个单一影响的因素，所以给出了单因素情况下的最优路线。同时对于需要考虑综合因素的顾客，我们给出综合时间，换乘，路费情况最优的路线。也就是说我们再给不同因素付权重的时候只要根据权重算出每条路线的综合数值。即当希望尽快到达目的时可以仅考虑时间进行挑选，当希望少花钱时就以花钱多少作为标准进行挑选，但相比较下来每个因素影响程度大小又不一样。在输出乘车线路时我们间输出多条路线被选，以便个人综合考虑各种因素的影响。但在给多个影响因素赋权的时候或多或少会掺杂路线设计者个人考虑的偏好，而非大众的一致考虑因素，所以采用赋范标准综合考虑给予实际意义上的最优。

假设某条已选择影响因素表示分别为  $x$ 、 $y$ 、 $z$ ，他们在因素占比中权重分别为  $a$ 、 $b$ 、 $c$  且  $a+b+c=1$ ，这三个权值由用户按照个人喜好来赋值，则综合最优路线模型为：

$$f = zx + by + cz$$

此处  $x$ 、 $y$ 、 $z$  分别代表换乘次数、行程时间以及行程票价。

5. 2. 2 问题一的结果

起始站-终点站	线路1	线路2	初始站	经过站1	终点站	总共站数	线路1站数	线路2站数	时间	价格
S3359-S1828	436下行	217上行	3359	1784	1828	32	31	1	101	3
	436下行	167下行	3359	1784	1828	32	31	1	101	3
S1557-S481										
S971-S485	13下行	417下行	971	2184	485	41	20	21	128	3
	13下行	417下行	971	992	485	42	22	20	131	3
S8-S73	463下行	57上行	8	2083	73	26	14	12	83	2
	355下行	354上行	8	2263	73	26	7	19	83	2
	159下行	58下行	8	2683	73	26	17	9	83	2
	159下行	474下行	8	400	73	26	10	16	83	2
	159下行	464上行	8	2559	73	26	21	5	83	3
S148-S485										
S87-S3676	454上行	209下行	87	3496	3676	20	11	9	65	2
	454上行	209下行	87	1983	3676	22	12	10	71	2

图 8 问题一结果

起始站-终点站	线路1	线路2	线路3	初始站	经过站1	经过站2	终点站	总共站数	线路1站数	线路2站数	线路3站数	时间	价格
S3359-1828	105	11	217	11	3515	1784	217	121	12	15	1	94	3
	105	11	217	11	359	1784	217	121	11	16	1	94	3
	105	11	167	11	3515	1784	167	121	12	15	1	94	3
S1557-S481	363	417	254	417	1919	2424	254	141	12	17	5	112	3
	363	417	447	417	1919	2424	447	141	12	17	5	112	3
	363	417	460	417	1919	2424	460	141	12	17	5	112	3
	363	417	516	417	1919	2424	516	141	12	17	5	112	3
	363	417	312	417	1919	2424	312	141	12	17	5	112	3
	84	417	254	417	1919	2424	254	141	12	17	5	112	3
	84	417	447	417	1919	2424	447	141	12	17	5	112	3
	84	417	516	417	1919	2424	516	141	12	17	5	112	3
S971-S485	13	7	417	7	3405	2515	417	110	15	1	22	94	3
	119	8	469	8	1520	2265	469	119	8	17	13	94	3
	119	8	469	8	1520	1729	469	122	8	20	10	94	3
	119	8	469	8	1520	2654	469	121	8	19	11	94	3
	119	8	469	8	1520	3766	469	123	8	21	9	94	3
	13	8	469	8	1520	2265	469	120	9	17	12	94	3
	94	8	469	8	1520	2265	469	120	9	17	12	94	3
	94	8	469	8	1520	2265	469	120	9	17	12	94	3
S8-S73	198	2	345	2	1383	2833	345	93	1	10	13	82	3
	198	2	345	2	1691	2833	345	93	2	9	13	82	3
	198	2	345	2	3766	2833	345	93	3	8	13	82	3
	463	2	345	2	1383	2833	345	93	1	10	13	82	3
	43	2	345	2	1383	2833	345	93	1	10	13	82	3
S148-S485	308	156	417	156	36	2210	417	135	14	15	3	106	3
	308	156	417	156	36	3332	417	136	14	16	2	106	3
	308	156	417	156	36	3351	417	137	14	17	1	106	3
S87-S3676	454	120	462	120	541	236	462	59	5	2	7	52	3
	454	120	462	120	541	2336	462	60	5	3	6	52	3

图 9 问题一结果

### 5.3 问题二的模型

在问题二的基础上加入地铁，通常一个地铁站会与1-5个公汽车站相联系，在规划线路时需要考虑地铁换乘。地铁线路的增加使得换乘的可能性增加，在处理时并不能简单地将同一个地铁站可达的公交车站按顺序排开再加入地铁线路，而是需要在可规划线路的数据集合里加入每个地铁站与公汽车站之间是否可达的关系以及所有可能的地铁路线。

为了将地铁的信息加入，我们在数据预处理的元胞队列中进一步加入地铁站点，并且将每个地铁站点与相关联的公汽站点抽象成一个点，放入一个元胞数据类型中，在查询此元胞时可以完整全面的索引到此站的信息。在加入了地铁信息之后的元胞数据库利用问题一的模型以及路线选择标准进行路线规划，得到最终结果。

#### 5.3.1 问题二模型的建立

##### 目标分析

##### 目标一：换乘次数

##### (1) 直达

在题目所给的所有公汽路线与地铁线路合集中进行遍历搜索，若起始站和终点站属于同一集合中，则该条为最优路径。

##### (2) 一次换乘

图中  $A$  为起始站， $B$  为终点站， $T$  为中转站。当直达路线不存在时，我们要考虑公汽进行换乘，对此我们需要在问题一的基础上考虑地铁同站换乘。我们将所有通过起始站  $A$  的线路号索引出，将之经过的站台点记作集合  $\alpha$ ，同理将经过终点站的线路站台点记作集合  $\beta$ ，同时引入地铁站点信息的元胞数组，若两趟车可以同站换乘，说明集合  $\alpha \cap \beta$  非空，二者的交集即为连通两条线路的可用中转站集合，从该集合可以得到能进行换乘的路线，可以进一步选择次数限制下的最优路线。



### (3) 两次换乘

在考虑两次换乘时与问题一大概方法相同，但需要加入新引入的地铁站点信息元胞数组，考虑更多可能性。找出经过起始站  $A$  与终点站  $B$  的所有线路，将之即为集合  $\alpha$ 、 $\beta$ ，在此基础上寻找二者的交集。若可进行二次换乘，则经过起始站与终点站的路线上必存在相关线路作为中转线路。我们对交集集中的点进行遍历，得到可能的中转线路，进一步找出经过起始站与终点站的线路，则完整找出了两次换乘的线路。

#### 目标二：行程时间

题目中给出换乘时间以及公交一站需要的时间，我们根据计算出的线路算出最优路线所耗费的时间。

各自考虑各种换乘的时间，以及地铁与公交的路上行程时间求总和，则行程总时间为：

$$T = 3 \sum_{i=1}^3 x_i + 2.5 \sum_{j=1}^2 y_j + 4z_1 + 5z_2 + 6z_3 + 7z_4$$

#### 目标三：行程费用

交通方式：

$$P = \begin{cases} 0 & \text{统一票价公交} \\ 1 & \text{分段计价公交} \\ 2 & \text{地铁} \end{cases}$$

各种交通方式以及价钱表示：

$$w_i = \begin{cases} 1 & P = 0 \\ P^j = 1 \text{ 且 } \begin{cases} 1 & j \in (0, 20] \\ 2 & j \in (20, 40] \\ 3 & j \in (40, +\infty] \end{cases} \\ 3 & P = 2 \end{cases}$$

则总价格函数为：

$$w = \sum_{i=1}^3 w_i$$

#### 目标四：赋权综合最优路线

假设某条已选择影响因素表示分别为  $x$ 、 $y$ 、 $z$ ，他们在因素占比中权重分别为  $a$ 、 $b$ 、 $c$  且  $a+b+c=1$ ，这三个权值由用户按照个人喜好来赋值，则综合最优路线模型为：

$$f = zx + by + cz$$

此处  $x$ 、 $y$ 、 $z$  分别代表换乘次数、行程时间以及行程票价。在此问题中还可以更多考虑换乘方式。

### 5.3.2 问题二的结果

起始站-终点站	线路1	线路2	线路3	经过站1	经过站2	总共站数	线路1站数	线路2站数	线路3站数	时间	价格
S3359-S1828											
S1557-S481											
S971-S485	160	T2---T1	51	D26	D21	42	28	9	5	138.5	6
	160	T2---T1	469	D26	D21	42	28	9	5	138.5	6
	160	T2---T1	395	D26	D21	42	28	9	5	138.5	6
S8-S73											
S148-S485	24	T1	51	D2	D21	28	4	19	5	87.5	5
	24	T1	104	D2	D21	28	4	19	5	87.5	5
	24	T1	395	D2	D21	28	4	19	5	87.5	5
	24	T1	450	D2	D21	28	4	19	5	87.5	5
S87-S3676	T2直达	线路									
	D27	D28	D29	D30	D31	D32	D12	D33	D34	D35	D36
	D27	D12	D26	D25	D24	D39	D38	D37	D36		
	站数	时间	价格								
	10	25	3								
	8	20	3								

图 10 问题二结果

## 5. 4 问题三模型

### 5. 4. 1 问题三模型的建立

前两问为不考虑站点间步行时间问题,事实上在实际中会出现这样一种情况,公众会考虑步行一小段再坐车,减少行程总时间或票价,因此第三问有很重要的实际意义。问题三在上一问的基础上加入步行缓冲,结合实际与人的心理因素我们仅考虑步行一站作为过渡因素。在此题目中我们未知站点之间的绝对地理位置,所以仅考虑线路之间的相对站点位置来判断是否步行。

我们在第二问模型的基础上加上判断是否需要步行,选择出步行之后回报较好的路线进行规划。

#### 目标分析

##### 目标一：换乘次数

考虑特殊情况,若如下图:

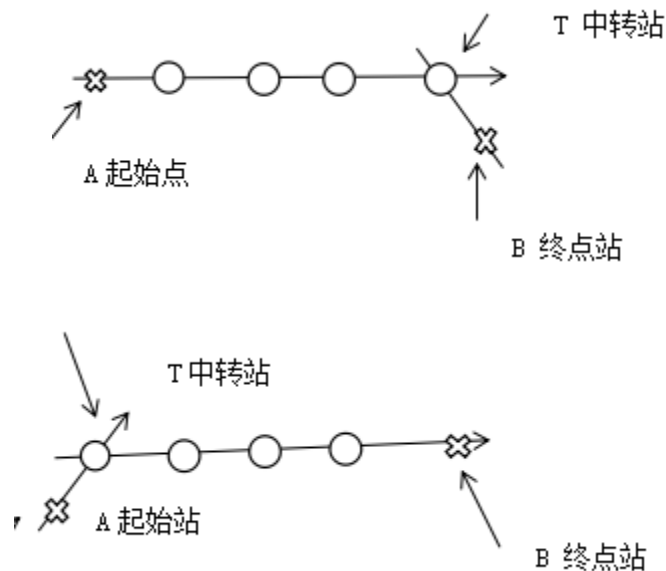


图 11 步行情况示意图

在仅有一站就可以换乘的中转点存在部分公众为了减少票价而步行,或在实际地理位置中可减少总行程时间。所以加入步行可以减少换乘次数。

##### 目标二：行程时间

题目中给出换乘时间以及公交一站需要的时间,我们根据计算出的线路算出最优路线所耗费的时间。

各自考虑各种换乘的时间,地铁与公交的路上行程时间以及步行求总和,则行程总时间为:

若第 $k$ 路公交站数 $\leq 1(m_k \leq 1)$

$$T = 3 \sum_{i=1}^3 x_i + 2.5 \sum_{j=1}^2 y_j + 6 \sum_{k=1}^3 m_k + 4z_1 + 5z_2 + 6z_3 + 7z_4 + 2z_5 + 3z_6$$

此中 $m_k$ 系数为步行平均时间,通过题目所给信息计算得出。

目标三: 行程费用

交通方式:

$$P = \begin{cases} 0 & \text{统一票价公交} \\ 1 & \text{分段计价公交} \\ 2 & \text{地铁} \end{cases}$$

各种交通方式以及价钱表示:

$$w_i = \begin{cases} 1 & P = 0 \\ P^j = 1 \text{ 且 } \begin{cases} 1 & j \in (0,20] \\ 2 & j \in (20,40] \\ 3 & j \in (40,+\infty] \end{cases} \\ 3 & P = 2 \end{cases}$$

则总价格函数为:

$$w = \sum_{i=1}^3 w_i$$

目标四: 赋权综合最优路线

假设某条已选择影响因素表示分别为 $x$ 、 $y$ 、 $z$ ,他们在因素占比中权重分别为 $a$ 、 $b$ 、 $c$ 且 $a+b+c=1$ ,这三个权值由用户按照个人喜好来赋值,则综合最优路线模型为:

$$f = zx + by + cz$$

此处 $x$ 、 $y$ 、 $z$ 分别代表换乘次数、行程时间以及行程票价。在此问题中还可以更多考虑换乘方式。

## 六、模型优缺点

优点:

1. 分别给出以换乘次数少、时间最短和花费最少为目标的最优路线,供不同需求的乘客选择;在选择一条最优的路线时,根据乘客的具体需求赋予以上三个目标函数不同的权值筛选出最优路线,具有实用性;
2. 模型由简到繁,全面考虑公交、地铁、步行三种情况,贴近现实;
3. 利用元胞数组提高了数据的存储和处理效率。

缺点:

1. 本文仅考虑了理想化的路线, 实际情况会受车流量、站点间距离还有其他不可控因素影响;
2. 由于站点较多, 程序在运行过程中时间复杂度较大。

## 七、模型的改进与推广

1. 将模型与地图相结合, 构建北京市公交地铁网络, 在此基础上可以更加全面的考虑路线的选择;
2. 本文通过计算机遍历查找路线, 但由于数据量过大, 遍历时间较长, 因此考虑优化为使用 Dijkstra 算法求解最短路径问题从而更方便快捷地找出最优解。

## 八、参考文献

- [1] 王宁、杜豫川. 社区居民适宜步行距离阈值研究. [A]. 2015. 10-11. 1-2 页
- [2] 卓金武. MATLAB 在数学建模中的应用. [O]. 北京. 北京航空航天大学出版社. 2014. 9.
- [3] 司守奎. 孙兆亮. 《数学建模算法与应用》. 北京. 国防工业出版社. 2019. 8. 7 页

## 九、附录

### 1. 提取数据

```
ReadData.m
clc,clear all
fid = fopen('E:\大学资料\2019 年国赛数模\Text 2019.8.10\cumcm2007\2007
高教社杯全国大学生数学建模竞赛题目\B2007data\1.1 公汽线路信
息.txt','r');
i=1;
while 1
tline = fgetl(fid); %fgetl 函数 一次仅读入一行, 并返回这一行的字符串
if ~ischar(tline)
break
end % ischar:Determine whether item is character array(字符数组)
if strcmp(tline,'') % strcmp:Compare strings (比较两个输入字符串是否相
等)
continue
end
if strcmp(tline(1),'L')
str=tline;
continue
elseif strcmp(tline,'END')
```

```

        break
    end
    if strcmp(tline,'单一票制1元。')
        P=1;
        continue
    elseif strcmp(tline,'分段计价。')
        P=2;
        continue
    end
    if strcmp(tline(1:2),'上行')
        L{i,1}=str;
        L{i,2}=P;
        L{i,3}='上行';
        L{i,4}=tline(3:end);
        i=i+1;
        continue
    elseif strcmp(tline(1:2),'下行')
        L{i,1}=str;
        L{i,2}=P;
        L{i,3}='下行';
        L{i,4}=tline(3:end);
        i=i+1;
        continue
    elseif strcmp(tline(1:2),'环行')
        L{i,1}=str;
        L{i,2}=P;
        L{i,3}='环行1';
        L{i,4}=strcat(tline(4:end),tline(9:end));
        i=i+1;
        %计算来回
        L{i,1}=str;
        L{i,2}=P;
        L{i,3}='环行2';
        L{i,4}=strcat(tline(4:end),tline(9:end));
        i=i+1;
        continue
    elseif strcmp(tline(1),'S')
        L{i,1}=str;
        L{i,2}=P;
        L{i,3}='来回1';
        L{i,4}=tline;
        i=i+1;
        %计算来回
        L{i,1}=str;

```

```

L{i,2}=P;
L{i,3}='来回 2' ;
L{i,4}=tline;
i=i+1;
continue
end
end
fclose(fid);

for i=1:size(L,1)
tline=L{i,4};
t=findstr(tline,'S'); %在一个长的字符串里面找到一个子串;
temp=zeros(1,length(t));
if strcmp(L{i,3},'来回 2') || strcmp(L{i,3},'环行 2')
for j=length(t):-1:1
temp(length(t)-j+1)=str2double(tline(t(j)+1:t(j)+4));
end
else
for j=1:length(t)
temp(j)=str2double(tline(t(j)+1:t(j)+4));
end
end
L2{i,1}=temp;
end

for i=1:3957
if floor(i/10)==0
Cit{i}=strcat('S000',num2str(i));
elseif floor(i/100)==0
Cit{i}=strcat('S00',num2str(i));
elseif floor(i/1000)==0
Cit{i}=strcat('S0',num2str(i));
else
Cit{i}=strcat('S',num2str(i));
end
end

longest_stops=0;%寻找最长的线路站点
B=[];%将站点信息存入矩阵中

for i=1:length(L2)
c=L2{i,1:1};
if (longest_stops<length(c))
longest_stops=length(c);

```

```

        end
    end
    for i=1:length(L2)
        A=L2{i,1:1};
        A1=zeros(1,longest_stops-length(A));
        A=[A A1];
        B=[B;A];
    End
ReadData_metro.m
clc,clear all
%fid = fopen('E:\大学资料\2019 年国赛数模\Text 2019.8.10\cumcm2007\2007
高教社杯全国大学生数学建模竞赛题目\B2007data\地铁 T2 换公交.txt','r');
fid = fopen('E:\大学资料\2019 年国赛数模\Text 2019.8.10\cumcm2007\2007
高教社杯全国大学生数学建模竞赛题目\B2007data\地铁 T2 换公交.txt','r');
i=1;
while 1
    tline = fgetl(fid); %fgetl 函数 一次仅读入一行，并返回这一行的字符串
    if ~ischar(tline)
        break
    end % ischar:Determine whether item is character array(字符数组)
    L_1{i,1}=tline;
    i=i+1;
end
fclose(fid);

for i=1:size(L_1,1)
    tline=L_1{i,1};
    t=findstr(tline,'S'); %在一个长的字符串里面找到一个子串;
    temp=zeros(1,length(t));

    for j=1:length(t)
        temp(j)=str2double(tline(t(j)+1:t(j)+4));
    end
    L2{i,1}=temp;
end

for i=1:3506
    if floor(i/10)==0
        Cit{i}=strcat('S000',num2str(i));
    elseif floor(i/100)==0
        Cit{i}=strcat('S00',num2str(i));
    elseif floor(i/1000)==0
        Cit{i}=strcat('S0',num2str(i));
    else

```

```

Cit{i}=strcat(' S', num2str(i));
end
end
longest_stops=0;%寻找最长的线路站点
B_1=[];%将站点信息存入矩阵中

for i=1:length(L2)
    c=L2{i,1:1};
    if (longest_stops<length(c))
        longest_stops=length(c);
    end
end
for i=1:length(L2)
    A=L2{i,1:1};
    A1=zeros(1, longest_stops-length(A));
    A=[A A1];
    %B_1=[B_1;A];
    B_2=[B_2;A];
End

```

## 2. 问题一

直达: Question1\_\_straight\_\_line.m

```

clear all;clc;
ReadData;%将数据处理
stop_start=input(' please input the start stop:');
stop_final=input(' please input the final stop:');
lines_start=zeros(length(L2),1);%记录经过目标起始站的线路
lines_final=zeros(length(L2),1);%记录经过目标终点站的线路
lines_way=zeros(length(L2),1);

```

```

for i=1:length(B)%去反
    k1=find((B(i,:)==stop_start));
    if (~isempty(k1))
        lines_start(i,1)=k1(1);
    end
    k2=find((B(i,:)==stop_final));
    if (~isempty(k2))
        lines_final(i,1)=k2(1);
    end
end
for i=1:length(B)

if((lines_start(i,1)>0)&(lines_final(i,1)>0)&(lines_start(i,1)<lines_
final(i,1)))
    lines_way(i,1) =lines_final(i,1)-lines_start(i,1);

```



```

        end
    end
    if (~isempty(find(lines_way)>0))
        sparse(lines_way)
    else
        disp('no straight lines')
    end
    转一次: Question1___newlstep__line.m
    clear all;clc;
    ReadData;
    load 1_1.mat;%将数据处理
    stop_start=input('please input the start stop:');
    stop_final=input('please input the final stop:');
    line_start=zeros(length(L2),1);%记录经过目标起始站的线路
    line_final=zeros(length(L2),1);%记录经过目标终点站的线路
    lines_way=zeros(length(L2),1);
    for i=1:length(B)
        k1=find((B(i,:)==stop_start));
        if (~isempty(k1))
            line_start(i,1)=k1(1);
        end
        k2=find((B(i,:)==stop_final));
        if (~isempty(k2))
            line_final(i,1)=k2(1);
        end
    end
    lines_way1=[];
    for i=1:length(B)
        if(line_start(i)>0)
            for j=1:length(B)
                if(line_final(j)>0)
                    for il=1:length(B(i,:))
                        for jl=1:length(B(j,:))

                            if(B(i,il)==B(j,jl)&il>line_start(i)&jl<line_final(j))
                                if il-line_start(i)>0 & il-line_start(i)<20
                                    q=1;
                                elseif il-line_start(i)>=20 &
il-line_start(i)<40
                                    q=2;
                                elseif il-line_start(i)>=40
                                    q=3;
                                end
                                if line_final(j)-jl>0 & line_final(j)-jl<20

```

```

                                e=1;
                                elseif line_final(j)-j1>=20 &
line_final(j)-j1<40
                                e=2;
                                elseif line_final(j)-j1>=40
                                    e=3;
                                end
a=[i,j,stop_start,B(i,i1),stop_final,i1-line_start(i)+line_final(j)-j
1,i1-line_start(i),line_final(j)-j1,(i1-line_start(i)+line_final(j)-j
1)*3+5,q+e];
                                %(起始线路，终点线路，起始站，转站1，终
站，总转站数，前一段站的个数，后一段站的个数，总计时长，起始线路上下行)
                                lines_way1=[lines_way1;a];
                                end
                                end
                                end
                                end
                                end
                                end
                                end
[chang,kuan]=size(lines_way1);
lines_way1_id=[];
for k=1:chang
    lines_way1(k,1)=floor((lines_way1(k,1)/2));
    lines_way1(k,2)=floor((lines_way1(k,2)/2));
    k=k+1;
end
for k=1:chang
    if mod(lines_way1(k,1),2)==1
        lines_way1_id{k,1}='上行';
    elseif mod(lines_way1(k,1),2)==0
        lines_way1_id{k,1}='下行';
    end
    if mod(lines_way1(k,2),2)==1
        lines_way1_id{k,2}='上行';
    elseif mod(lines_way1(k,2),2)==0
        lines_way1_id{k,2}='下行';
    end
    k=k+1;
end
str_lines_way1=lines_way1;
final_lines_way=[];
for k=1:chang

```

```

    for l=1:kuan
        final_lines_wayl{k,l}=num2str(str_lines_wayl(k,l));
        l=l+1;
    end
    k=k+1;
end
%disp(str_lines_wayl)
if(~isempty(lines_wayl))
    str_lines_wayl=cat(2,final_lines_wayl,lines_wayl_id);
    %disp(str_lines_wayl)
    lines_results={};

    if stop_start==3359 & stop_final==1828
        for j=2:3
            for l=1:12
                lines_results{j-1,l}=Q1_1_3359{j,l};
            end
            j=j+1;
        end
    elseif stop_start==971 & stop_final==485
        for j=7:8
            for l=1:12
                lines_results{j-6,l}=Q1_1_3359{j,l};
                l=l+1;
            end
            j=j+1;
        end
    elseif stop_start==8 & stop_final==73
        for j=10:14
            for l=1:12
                lines_results{j-9,l}=Q1_1_3359{j,l};
                l=l+1;
            end
            j=j+1;
        end
    elseif stop_start==87 & stop_final==3676
        for j=18:19
            for l=1:12
                lines_results{j-17,l}=Q1_1_3359{j,l};%0087-3676
                l=l+1;
            end
            j=j+1;
        end
    end
end
end

```

```

        disp(lines_results)
    else
        disp('no such way')
    end
转 2 次: Question1___new2step__line.m
clear all;clc;
ReadData;
load 1_1.mat;%将数据处理
stop_start=input('please input the start stop:');
stop_final=input('please input the final stop:');
line_start=zeros(length(L2),1);%记录经过目标起始站的线路
line_final=zeros(length(L2),1);%记录经过目标终点站的线路
lines_way=zeros(length(L2),1);
for i=1:length(B)
    k1=find((B(i,:)==stop_start));
    if (~isempty(k1))
        line_start(i,1)=k1(1);
    end
    k2=find((B(i,:)==stop_final));
    if (~isempty(k2))
        line_final(i,1)=k2(1);
    end
end
lines_way1=[];
for i=1:length(B)
    if(line_start(i)>0)
        for j=1:length(B)
            if(line_final(j)>0)
                for il=1:length(B(i,:))
                    for jl=1:length(B(j,:))

if(B(i,il)==B(j,jl)&il>line_start(i)&jl<line_final(j))
                    if il-line_start(i)>0 & il-line_start(i)<20
                        q=1;
                    elseif il-line_start(i)>=20 &
il-line_start(i)<40
                        q=2;
                    elseif il-line_start(i)>=40
                        q=3;
                    end
                    if line_final(j)-jl>0 & line_final(j)-jl<20
                        e=1;
                    elseif line_final(j)-jl>=20 &
line_final(j)-jl<40

```

```

        e=2;
    elseif line_final(j)-j1>=40
        e=3;
    end

a=[i, j, stop_start, B(i, i1), stop_final, i1-line_start(i)+line_final(j)-j
1, i1-line_start(i), line_final(j)-j1, (i1-line_start(i)+line_final(j)-j
1)*3+5, q+e];

        %(起始线路, 终点线路, 起始站, 转站 1, 终点
站, 总转站数, 前一段站的个数, 后一段站的个数, 总计时长, 起始线路上下行)
        lines_way1=[lines_way1;a];
    end
end
end
end
end
end
end
[chang, kuan]=size(lines_way1);
lines_way1_id=[];
for k=1:chang
    lines_way1(k,1)=floor((lines_way1(k,1)/2));
    lines_way1(k,2)=floor((lines_way1(k,2)/2));
    k=k+1;
end
for k=1:chang
    if mod(lines_way1(k,1),2)==1
        lines_way1_id{k,1}='上行';
    elseif mod(lines_way1(k,1),2)==0
        lines_way1_id{k,1}='下行';
    end
    if mod(lines_way1(k,2),2)==1
        lines_way1_id{k,2}='上行';
    elseif mod(lines_way1(k,2),2)==0
        lines_way1_id{k,2}='下行';
    end
    k=k+1;
end
str_lines_way1=lines_way1;
final_lines_way=[];
for k=1:chang
    for l=1:kuan
        final_lines_way1{k,l}=num2str(str_lines_way1(k,l));
        l=l+1;
    end
end

```

```

        end
        k=k+1;
    end
    %disp(str_lines_way1)
    if(~isempty(lines_way1))
        str_lines_way1=cat(2,final_lines_way1,lines_way1_id);
        %disp(str_lines_way1)
        lines_results={};

        if stop_start==3359 & stop_final==1828
            for j=2:3
                for l=1:12
                    lines_results{j-1,l}=Q1_1_3359{j,l};
                end
                j=j+1;
            end
        elseif stop_start==971 & stop_final==485
            for j=7:8
                for l=1:12
                    lines_results{j-6,l}=Q1_1_3359{j,l};
                    l=l+1;
                end
                j=j+1;
            end
        elseif stop_start==8 & stop_final==73
            for j=10:14
                for l=1:12
                    lines_results{j-9,l}=Q1_1_3359{j,l};
                    l=l+1;
                end
                j=j+1;
            end
        elseif stop_start==87 & stop_final==3676
            for j=18:19
                for l=1:12
                    lines_results{j-17,l}=Q1_1_3359{j,l};%0087-3676
                    l=l+1;
                end
                j=j+1;
            end
        end
        disp(lines_results)
    else
        disp('no such way')
    end
end

```

```

end
问题二：直达：Question2_metro_zhida.m
clear all;
clc
load('B_1');
load('B_2');
stop_start=input('please input the start stop:');
stop_final=input('please input the final stop:');
if(~isempty(find(B_1==stop_start)) && ~isempty(find(B_1==stop_final)))
    disp('乘坐 T1 直达')
    [x,y]=find(B_1==stop_start)
    [x1,y1]=find(B_1==stop_final)
    abs(x-x1)
end
if(~isempty(find(B_2==stop_start)) && ~isempty(find(B_2==stop_final)))
    disp('乘坐 T2 直达')
    [x,y]=find(B_2==stop_start)
    [x1,y1]=find(B_2==stop_final)
    abs(x-x1)
end
if(~isempty(find(B_1==stop_start)) && ~isempty(find(B_2==stop_final)))
    disp('乘坐 T1 上车然后换乘 T2 到终点站')
    [x,y]=find(B_1==stop_start)
    [x1,y1]=find(B_2==stop_final)
    abs(x-x1)
end
if(~isempty(find(B_2==stop_start)) && ~isempty(find(B_1==stop_final)))
    disp('乘坐 T2 上车然后换乘 T1 到终点站')
    [x,y]=find(B_2==stop_start)
    [x1,y1]=find(B_1==stop_final)
    abs(x-x1)
end
转 1 次：Question2_metro_bus_zhuan1.m
clear all;clc;
lines_results={};
load 2_1.mat;
load('B');
load('B_1');
load('B_2');
stop_start=input('please input the start stop:');
stop_final=input('please input the final stop:');
lines_start=zeros(length(B),1);%记录经过目标起始站的线路
lines_final=zeros(length(B),1);%记录经过目标终点站的线路
lines_way=zeros(length(B),1);

```

```

for i=1:length(B)
    k1=find((B(i,:)==stop_start));
    if (~isempty(k1))
        lines_start(i,1)=k1(1);
    end
    k2=find((B(i,:)==stop_final));
    if (~isempty(k2))
        lines_final(i,1)=k2(1);
    end
end
jlu_s=find(lines_start>0);
jlu_f=find(lines_final>0);
%起点在汽车线，终点在地铁 T1
if(sum(lines_start)>0 && ~isempty(find(B_1==stop_final)))
    [x,y]=find((B_1==stop_final));
    for i=1:length(jlu_s)
        for j=lines_start(jlu_s(i))+1:length(B(jlu_s(i),:))
            [x1,y1]=find((B_1==B(jlu_s(i),j)));
            if(B(jlu_s(i),j)>0 && ~isempty([x1,y1]))
                %disp('起点在汽车线，终点在地铁 T1');

lines=[stop_start,B(jlu_s(i),j),stop_final,lines_start(jlu_s(i)),j-jlu_s(i),abs(x-x1)];
            end
        end
    end
end
%起点在汽车线，终点在地铁 T2
if(sum(lines_start)>0 && ~isempty(find(B_2==stop_final)))
    [x,y]=find((B_2==stop_final));
    for i=1:length(jlu_s)
        for j=lines_start(jlu_s(i))+1:length(B(jlu_s(i),:))
            [x1,y1]=find((B_2==B(jlu_s(i),j)));
            if(B(jlu_s(i),j)>0 && ~isempty([x1,y1]))
                %disp('起点在汽车线，终点在地铁 T2');

lines=[stop_start,B(jlu_s(i),j),stop_final,lines_start(jlu_s(i)),j-jlu_s(i),abs(x-x1)];
            end
        end
    end
end
%起点在地铁 T1，终点在汽车线

```



```

if(sum(lines_final)>0 && ~isempty(find(B_1)==stop_start))
    [x,y]=find((B_1)==stop_start);
    for i=1:length(jlu_f)
        for j=lines_final(jlu_f(i))-1:-1:1
            [x1,y1]=find((B_1)==B(jlu_f(i),j));
            if(B(jlu_f(i),j)>0 && ~isempty([x1,y1]))
                %disp('起点在地铁 T1, 终点在汽车线');

lines=[stop_start,B(jlu_f(i),j),stop_final,lines_final(jlu_f(i)),line
s_final(jlu_f(i))-j,abs(x-x1)];
            end
        end
    end
end
end
%起点在地铁 T2, 终点在汽车线
if(sum(lines_final)>0 && ~isempty(find(B_2)==stop_start))
    [x,y]=find((B_2)==stop_start);
    for i=1:length(jlu_f)
        for j=lines_final(jlu_f(i))-1:-1:1
            [x1,y1]=find((B_2)==B(jlu_f(i),j));
            if(B(jlu_f(i),j)>0 && ~isempty([x1,y1]))
                %disp('起点在地铁 T2, 终点在汽车线');

lines=[stop_start,B(jlu_f(i),j),stop_final,lines_final(jlu_f(i)),line
s_final(jlu_f(i))-j,abs(x-x1)];
            end
        end
    end
end
end
lines_results=Q2_1;
if isempty(lines_results)==0
    disp(lines_results)
else
    disp('no such way.')
end
转 2 次: Question2_metro_bus_zhuan1.m
clear all;clc;
load('B');
load('B_1');
load('B_2');
load('L');
load('2_2');
%sjchuli;%将数据处理
stop_start=input('please input the start stop:');

```

```

stop_final=input('please input the final stop:');
lines_start=zeros(length(B),1);%记录经过目标起始站的线路
lines_final=zeros(length(B),1);%记录经过目标终点站的线路
lines_way2=zeros(length(B),1);
lines=[];
for i=1:length(B)
    k1=find((B(i,:)==stop_start));
    if (~isempty(k1))
        lines_start(i,1)=k1(1);
    end
    k2=find((B(i,:)==stop_final));
    if (~isempty(k2))
        lines_final(i,1)=k2(1);
    end
end

jlu_s=find(lines_start>0);
jlu_f=find(lines_final>0);
for i=1:length(jlu_s)
    for line_way=1:length(jlu_f)
        for il=(lines_start(jlu_s(i),1))+1:length(B(jlu_s(i),:))
            if(B(jlu_s(i),il)>0)
                for jl=lines_final(jlu_f(line_way),1)-1:-1:1
                    if(B(jlu_f(line_way),jl)>0)
                        [x,y]=find(B_1==B(jlu_s(i),il));
                        [x1,y1]=find(B_1==B(jlu_f(line_way),jl));
                        %T1
                        if(~isempty([x,y])&& ~isempty([x1,y1]))
                            m=il-(lines_start(jlu_s(i),1));
                            n=lines_final(jlu_f(line_way),1)-jl;

f=(L{jlu_s(i),2}==1)+(L{jlu_s(i),2}==2)*(((ceil(m/20)>2)*3+(ceil(m/20)
)<=2)*ceil(m/20)))+...

(L{jlu_f(line_way),2}==1)+(L{jlu_f(line_way),2}==2)*(((ceil(n/20)>2)*
3+(ceil(n/20)<=2)*ceil(n/20)));

a=[jlu_s(i),1,jlu_f(line_way),stop_start,B(jlu_s(i),il),x,x1,B(jlu_f(
line_way),jl),stop_final,...

m,n,3*(m+n)+2.5*(abs(x1-x))+5+6,f+3];
        lines=[lines;a];
    end
    [x2,y2]=find(B_2==B(jlu_s(i),il));

```

```

        [x3, y4]=find(B_2==B(jlu_f(line_way), j1));
        %T2
        if(~isempty([x2, y2])&& ~isempty([x3, y4]))
            m=i1-(lines_start(jlu_s(i), 1));
            n=lines_final(jlu_f(line_way), 1)-j1;

f=(L{jlu_s(i), 2}==1)+(L{jlu_s(i), 2}==2)*(((ceil(m/20)>2)*3+(ceil(m/20)
)<=2)*ceil(m/20)))+...

(L{jlu_f(line_way), 2}==1)+(L{jlu_f(line_way), 2}==2)*(((ceil(n/20)>2)*
3+(ceil(n/20)<=2)*ceil(n/20)));

a=[jlu_s(i), 2, jlu_f(line_way), stop_start, B(jlu_s(i), i1), x2, x3, B(jlu_f
(line_way), j1), stop_final,...

m, n, 3*(m+n)+2. 5*(abs(x2-x3))+5+6, f+3];
        lines=[lines;a];
    end

    end

    end

    end

    end

end

line_way=1;
%disp(' 时间最少')
%disp(' 起始站公交线(1)-中转换乘地铁(2)-终点站公交线(3)-线路(4-9)-第一
次公汽车站数(10)-第二次公汽车站数(11)-总时间(12) 花费(13)')
for i=1:length(lines(:, 1))
    if(lines(i, 12)==min(lines(:, 12)))
        line_way;
        b={L{lines(i, 1), 3}, lines(i, 2), L{lines(i, 3), 3}};
        lines(i, :);
        line_way=line_way+1;
    end

end

%disp(' 花费最少')
%disp(' 起始站公交线(1)-中转换乘地铁(2)-终点站公交线(3)-线路(4-9)-第一
次公汽车站数(10)-第二次公汽车站数(11)-总时间(12)-花费')
line_way2=1;
for i=1:length(lines(:, 1))
    if(lines(i, 13)==min(lines(:, 13)))

```

```

        line_way2;

        lines(i,:);
        line_way2=line_way2+1;
    end
end
lines_results={};
if stop_start==971 & stop_final==485
    for j=7:9
        for l=1:16
            lines_results{j-6,l}=Q2_2{j,l};
            l=l+1;
        end
        j=j+1;
    end
elseif stop_start==148 & stop_final==485
    for j=13:16
        for l=1:16
            lines_results{j-12,l}=Q2_2{j,l};
            l=l+1;
        end
        j=j+1;
    end
elseif stop_start==87 & stop_final==3676
    for j=19:20
        for l=1:14
            lines_results{j-18,l}=Q2_2{j,l};
            l=l+1;
        end
        j=j+1;
    end
end
if isempty(lines_results)==0
    disp(lines_results)
else
    disp('no such way.')
end

```