



交通信息检测与处理课程设计报告

(嘉定 104 路公交运行管理分析)

班 级： 18 级交信班

姓 名： 赵冠华

学 号： 1852127

提交日期： 2022. 01. 09



目录

0.	目录.....	1
1.	背景.....	2
1.1.	任务背景.....	2
1.2.	任务描述.....	2
2.	任务 1: 数据处理.....	3
2.1.	修正数据漂移.....	3
2.1.1.	划分矩形区域.....	4
2.1.2.	地图匹配.....	7
2.1.3.	时空轨迹转换.....	10
2.1.4.	二次修正.....	12
2.2.	数据剔除和修复.....	14
2.3.	补充数据基础.....	15
3.	任务 2: 公交运行状况定性分析.....	17
3.1.	公交运行轨迹分析.....	17
3.2.	车辆使用情况分析.....	20
3.3.	线路运行指标.....	21
3.3.1.	出发准点率.....	21
3.3.2.	时刻表偏差.....	25
3.3.2.1.	发车间隔.....	28
3.3.3.	运营时间分析.....	31
3.3.4.	Layover 时间分析.....	34
4.	任务 3: 公交路段运行情况分析.....	37
4.1.	路段运营时间及延误.....	37
4.2.	停靠时间分析.....	42
4.3.	站点车间时距分析.....	45
4.4.	站点乘客延误分析.....	46
5.	问题总结与建议.....	47

1. 背景

1.1. 任务背景

本次课程设计的主要目标是能够综合运用《交通信息检测与处理》课程讲授的理论、知识和方法，依据任务书要求，运用一些数据处理的方法，对已有的上海嘉定 104 路公交的 GPS 数据进行处理，并且能用文字、图表等对处理后的结果进行描述，了解并评价嘉定 104 路公交的运行状况。



图 1.1 嘉定 104 路上行线路地图



图 1.2 嘉定 104 路下行线路地图

上图 1.1 与 1.2 为嘉定 104 路公交车的上下行线路地图。

1.2. 任务描述

本次课程设计主要有三个任务：

(1)、数据处理：包含对数据漂移的修正、时空轨迹转换、数据剔除与修复以及对数据基础（起点站出站、终点站进站以及中间停靠站进出站信息）的补充；

(2)、公交运行状况定性分析：包含公交运行轨迹分析、车辆使用情况分析、线路运行指标（准点率、时刻表偏差、发车间隔、运营时间、Layover 时间）分析；

(3)、公交路段运行情况分析：对路段运行指标进行分析，具体包含路段运营时间及延误、停靠时间、站点车间时距、站点乘客延误；

在以上三个子任务完成的基础上，对上海嘉定 104 路公交在 2021 年 9 月 6 日到 9 月 10 日的运行管理进行问题总结并提出建议。



2. 任务 1：数据处理

对数据进行观察发现，原始公交数据存在着包括 GPS 数据缺失、GPS 数据漂移、GPS 数据误差过大等问题，因此本章节将对 GPS 数据采用修正或修复的方法进行进一步处理与筛选。

2.1. 修正数据漂移

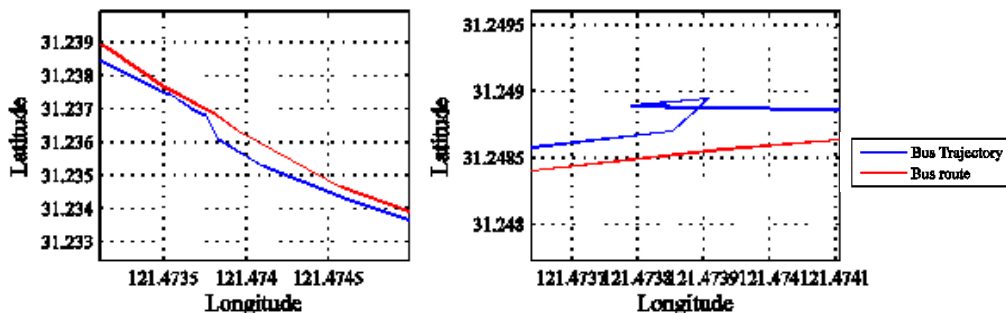


图 2.1 GPS 数据漂移现象

数据漂移指部分车辆的经度或纬度数据存在较大的漂移，超过了 GPS 经纬度数据的正常波动范围，与实际公交线路有一定的偏差。因此对其进行修正，修正主要包括地图匹配与二次修正两个步骤。

已知，根据 GPS 数据表格中的 direction 和 nidx 字段组合可唯一确定一组上行轨迹或下行轨迹。以 9 月 6 日 GPS 数据为例，共有 0~74 条车辆周转记录，由于前 74 次车辆周转记录由一次上行以及一次下行组成，第 75 次车辆周转记录只有一次上行轨迹，故 9 月 6 日当天共有 149 条轨迹，其中 75 条上行轨迹、74 条下行轨迹。同理，统计得到 9 月 7 日至 9 月 10 日嘉定 104 路公交车轨迹数目如表 2.1 所示。

表 2.1 嘉定 104 路公交车轨迹数目统计表

日期	9 月 6 日	9 月 7 日	9 月 8 日	9 月 9 日	9 月 10 日
上行轨迹数目	75	75	75	76	78
下行轨迹数目	74	75	74	75	77
总轨迹数目	149	150	149	151	155

接下来我将以 9 月 6 日 GPS 数据中“nidx=21,direction=0”的轨迹为例进行数据漂移的修正。读取原始 GPS 数据与地图数据，使用 MATLAB 做轨迹图，如图 2.2 所示。

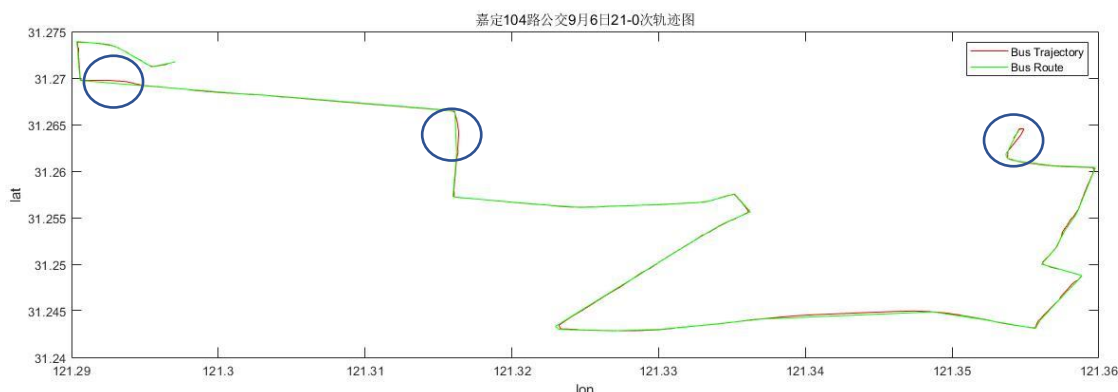


图 2.2 嘉定 104 路公交 9 月 6 日 21 周转上行 GPS 轨迹



观察图 2.2 发现，本次公交车运行轨迹在起点真新车站、黄家花园路临夏路以及曹安公路星华公路出现较为明显的数据漂移现象。故首先对其进行地图匹配。地图匹配的技术路线如图 2.3 所示。

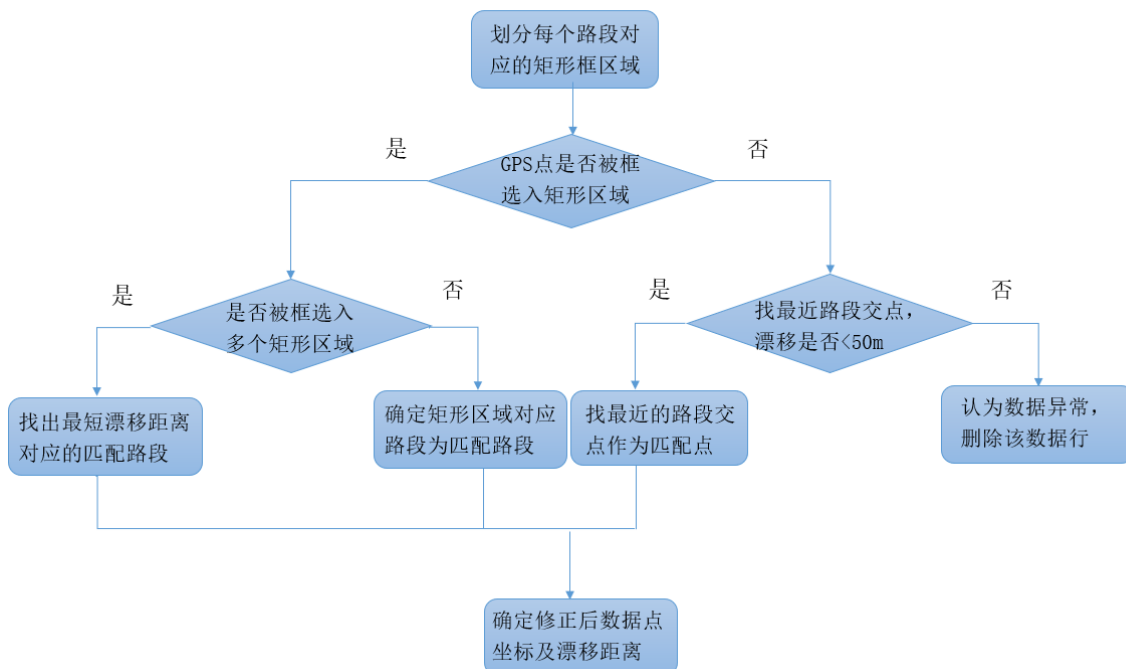


图 2.3 地图匹配技术路线

2.1.1. 划分矩形区域

在划分矩形区域时，以偏移 50m 为界限，为每个路段划定矩形范围。

假设地球为一半径为 R 的表面光滑圆球体，表面上同一经线圈上相差 $1''$ 两点间的距离为 $2\pi R/360/3600$ ；表面上同一纬线圈上相差 $1''$ 两点间的距离为 $2\pi R \times \cos(\text{纬度})/360/3600$ ；当 R 取半径平均值 6371km 时，地球表面上同一经线圈上相差 $1''$ 两点间的距离约为 30.887m，地球表面上同一纬线圈上相差 $1''$ 两点间的距离约为 $30.887m \times \cos(\text{纬度})$ 。由于本次课程设计所涉及地理位置的经度在 $121.297079^\circ \sim 121.354537^\circ$ ，纬度在 $31.2428378^\circ \sim 31.2739215^\circ$ 之间，为计算简便，在进行经纬度与路程长度（米）的换算时规定：

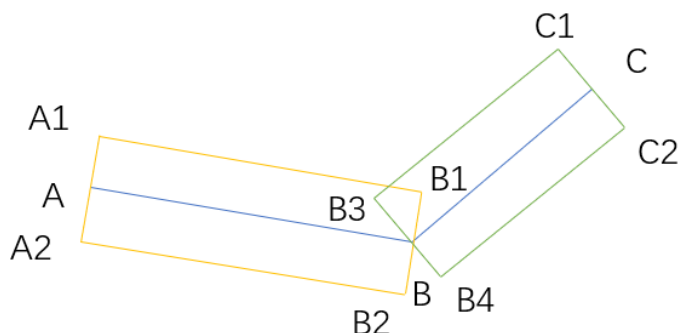
经度（lon 东西方向） $1''=30.887m$

纬度（lat 南北方向） $1''=30.887m \times \cos((31.2428378+31.2739215)/2^\circ)=26.395m$

在距离判断中，对两个点 $(x1, y1), (x2, y2)$ 之间的距离进行经纬度与米的换算，换算依据如下所示：

$$\left\{ \begin{array}{l} \sqrt{(R * dx * \cos(average))^2 + (R * dy)^2} = distance \\ dx = (x1 - x2) * \pi / 180 \\ dy = (y1 - y2) * \pi / 180 \\ R = 6371000m \\ \cos(average) = \cos\left(\frac{31.2428378 + 31.2739215}{2}\right) = 0.85483599 \end{array} \right.$$

为了确定矩形区域，需要对每个端点求出对应的两个点，如图 2.4 所示，对于 A、B、C 三个点而言，对于中间点都应求出与该点相距 50m 且垂直于该点与下一点连线的对应两个矩形端点，以及与该点相距 50m 且垂直于该点与上一点连线的对应两个矩形端点；对于



第一个点应该求出与该点相距 50m 且垂直于该点与下一点连线的对应两个矩形端点；而对于最后一个点求出与该点相距 50m 且垂直于该点与上一点连线的对应两个矩形端点。

图 2.4 地图匹配示意图

设 $A(x_1, y_1)$, $B(x_2, y_2)$, 需要求得 A_1, A_2 的坐标, 则这一步骤通过 MATLAB 的 `vpasolve` 函数解下列方程组而实现:

$$\begin{cases} k = (y_2 - y_1) / (x_2 - x_1) \\ k * \frac{y - y_1}{x - x_1} = -1 \\ ((x - x_1) * 3600 * \text{lon})^2 + ((y - y_1) * 3600 * \text{lat})^2 = 50 * 50 \end{cases}$$

将解储存在每个端点处，绘图，得到图 2.5。

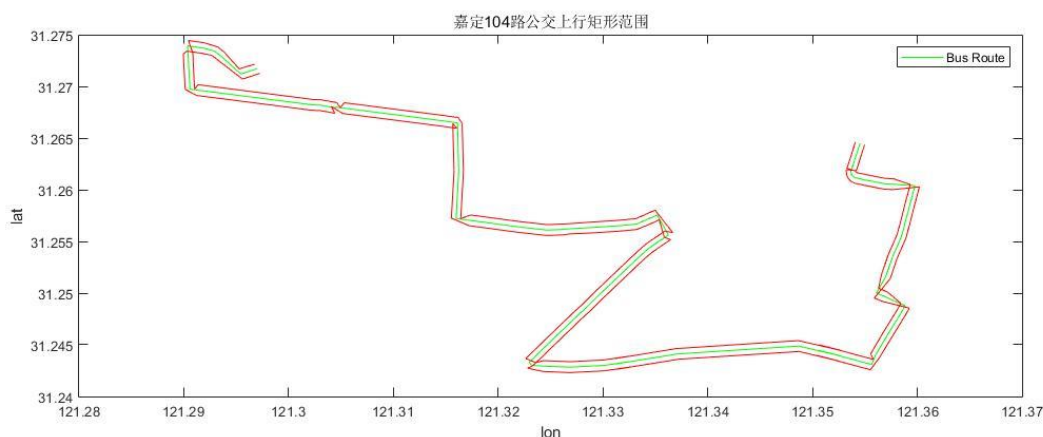


图 2.5 104 路公交上行矩形范围(误)

观察发现，对于每一个端点可以解得两个对应点，但这两个对应点并非按顺序存储为方便矩形区域的绘制与确定，需要对两个点与对应线段的关系（点在路程的左侧还是右侧）进行确定。由于 104 路公交车路线为不规则曲线，故这里将起点以及终点连接，将上述问题转化为采用射线法判断两个点在图形内侧还是外侧。

射线法判断点与多边形关系的原理为：从待判断点引出一条射线，射线与多边形相交，如果交点为偶数，则点不在多边形内，如果交点为奇数，则点不在多边形内。判断条件为：



判断第一个点与多边形的位罝，奇数在内部，偶数在外部；若第一个点在内部，则颠倒第一个点与第二个点的位置。

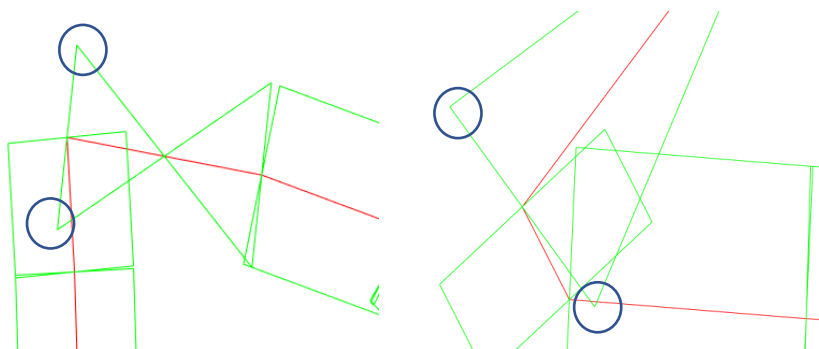


图 2.6 两端点均在多边形外侧

在此基础上，由于在本次课程设计中可能出现两个端点同时都在图形外侧的情况（如图 2.6 所示），因此在绘制矩形时根据长度对 rec 进行再判断，判断程序如下：

```
%3 4再排序
temp=zeros(1,2);
dis23=(rec(2,1)-rec(3,1))*(rec(2,1)-rec(3,1))+(rec(2,2)-rec(3,2))*(rec(2,2)-rec(3,2));
dis24=(rec(2,1)-rec(4,1))*(rec(2,1)-rec(4,1))+(rec(2,2)-rec(4,2))*(rec(2,2)-rec(4,2));
if dis23 > dis24
    temp=rec(3,:);
    rec(3,:)=rec(4,:);
    rec(4,:)=temp;
end
```

在本程序中，由于之前将解储存在了standard(:,5:12)之中，设定外侧点在 5:6 以及 9:10 的范围内。以图 2.4 为例解释得到standard(:,5:12)叙述如下表 2.2 所示：

表 2.2 standard 解释

例子	standard (:,5:6)	standard (:,7:8)	standard (:,9:10)	standard (:,11:12)
起点	A1	A2		
中间点	B1	B2	B3	B4
终点			C3	C4

根据所得端点值绘制上行轨迹的矩形范围，得到图 2.7；同理得到嘉定 104 路公交下行轨迹的矩形范围，展示为图 2.8。在绘图的同时将每个矩形四个角的坐标储存在 excel 表格中。

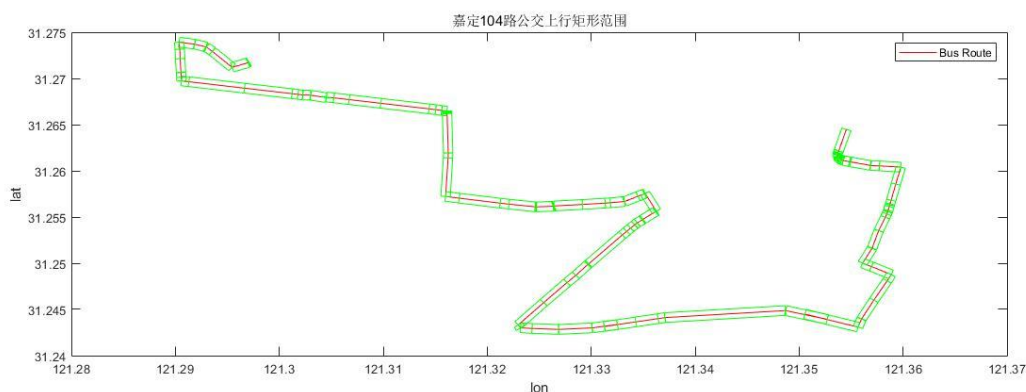




图 2.7 104 路公交上行矩形范围

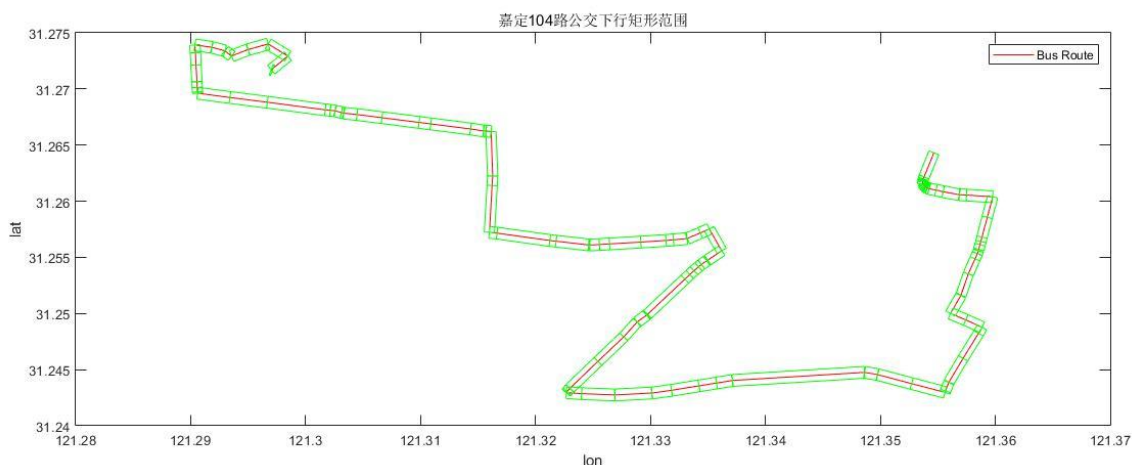


图 2.8 104 路公交下行矩形范围

2.1.2. 地图匹配

确定一个点是否在已知矩形范围内,可以通过MATLAB中的inpolyon函数进行判断,最终得到一个 $\text{rec_num} * \text{dot_num}$ 的矩阵, 值为 1 代表, 该点在对应矩形内部; 为 0 代表该点在对应矩形外部。

```

p=temp(:,1:2);
x=p(:,1);
y=p(:,2);
%对每个矩形(rec_num=110个)进行款选判断,并累积,用table记录 横为矩形 竖为点
table=zeros(dot_num,rec_num);
for j = 1: rec_num
    %画矩形
    xv=[recshang(j,1) recshang(j,3) recshang(j,5) recshang(j,7) recshang(j,1)];
    yv=[recshang(j,2) recshang(j,4) recshang(j,6) recshang(j,8) recshang(j,2)];
    in=inpolyon(x,y,xv,yv);%in=0-在外部 in=1-在内部或边缘
    table(:,j)=in;
end

```

横向统计每个点被框选进的矩形数目,对于所选上行轨迹而言,该值可为 0/1/2/3;并记录点在哪个矩形之中,如表 2.3 所示,同时用 road 字段记录其最后投影点所在路段的起始点,范围为 1~rec_num 以及 999,令需要被删除的点 road=999。

```

%计算每个点对应的矩形个数,并使用rec_flag记录点在哪个矩形之中
times=zeros(dot_num,1);
rec_flag=zeros(dot_num,4);
for i = 1:dot_num
    k=0;
    for j = 1: rec_num
        if table(i,j)==1
            times(i,1)=times(i,1)+1;
            k=k+1;
            rec_flag(i,k)=j;
        end
    end
end
end

```




表 2.3 统计数据点被框选的矩形

196	21	0	0	0	1	1	0	0	0
197	22	0	0	0	2	1	0	0	0
198	22	0	0	0	3	1	0	0	0
199	24	25	26	0	4	1	0	0	0
200	24	25	26	0	5	0	0	0	0
201	24	25	26	0	6	0	0	0	0
202	24	25	26	0	7	0	0	0	0
203	26	0	0	0	8	0	0	0	0
204	26	0	0	0	9	0	0	0	0
205	26	0	0	0					

(1)、当值为 0 时，计算与最近路段的交点，判断其漂移是否<50m。当数据点与交点的连线长度小于或等于 50 米时，将该数据点映射至该交点位置，否则就删除该数据点。

```
%0
if times(i,1)==0
    %计算与standard点的距离，distance为距离的平方
    distance=zeros(standard_num,1);
    for j =1:standard_num
        %数据点a是temp，标准点b是standard
        b=standard(j,1:2);
        distance(j,1)=((b(1,2)-a(1,2))*3600*lat)^2 + ((b(1,1)-a(1,1))*3600*lon)^2;
        if distance(j,1) <= 2500
            times(i,2:3)=standard(j,1:2);
        end
    end
    %当times(:,1:3)=(0 0 0)时说明该数据应当删除
end
```

(2)、当值为 1 时，将数据点投影到矩形对应路段上。首先，通过 rec_flag 获得对应矩形标号，通过标号获得对应路段，并计算数据点到对应路段所在直线的垂足坐标（由于本次课程设计中多次进行垂足的计算，故这里设计一个函数 get_footpoint 方便调用）。

该函数同样采用 solve 函数求解。输入线段两 endpoint 坐标与数据点坐标，设垂足为(x, y)，分别以“数据点与垂足连线垂直于线段所在直线”以及“垂足点与线段两 endpoint 共线（点差法证明斜率相等）”为条件建立方程组，返回垂足坐标与垂线段长度。

```
function p = get_footpoint(x1,y1,x2,y2,x3,y3)
%GET_FOOTPOINT 对应路段的endpoint (x1,y1) (x2,y2), 数据点(x3,y3)
% 返回垂足坐标p(1,1:2)与距离

syms x y
k=(y2-y1)/(x2-x1);
[x,y]=vpasolve([k*((y3-x)/(x3-x))==-1, (y2-y)/(x2-x)==k],[x,y]);
x=double(x);
y=double(y);
p(1,1)=x;
p(1,2)=y;
lon=30.887;
lat=26.395;
p(1,3)=sqrt(((y-y3)*3600*lat)^2 + ((x-x3)*3600*lon)^2);

if times(i,1)==1
    %找到数据点对应矩形
    t=rec_flag(i,1);
    %获得对应路段的endpoint (x1,y1) (x2,y2), 数据点(x3,y3)
    x1=standard(t,1);
    y1=standard(t,2);
    x2=standard(t+1,1);
    y2=standard(t+1,2);
    x3=a(1,1);
    y3=a(1,2);
    %求点在路段上的投影位置
    p = get_footpoint(x1,y1,x2,y2,x3,y3);
    times(i,2:3)=p(1,1:2);
end
```



(3)、当值 >1 时，计算该点到每个对应路段的漂移距离，选择漂移距离最小的垂足作为投影后的数据点坐标。

```
%n
if times(i,1)>1
    %找到数据点对应矩形
    l=times(i,1);
    %对每个路段进行投影，先计算所有并储存
    shadow=zeros(1,3);
    for j=1:l
        t=rec_flag(i,j);
        %获得对应路段的端点(x1,y1)(x2,y2),数据点(x3,y3)
        x1=standard(t,1);
        y1=standard(t,2);
        x2=standard(t+1,1);
        y2=standard(t+1,2);
        x3=a(1,1);
        y3=a(1,2);
        p = get_footpoint(x1,y1,x2,y2,x3,y3);
        shadow(j,:)=p;
    end
end

%选择投影最短的路段:假设第一个最小，不断比较
min=1;
if j == 2
    if shadow(1,3)>shadow(2,3)
        min=2;
    end
else for j=2:l
    if shadow(min,3)>shadow(j,3)
        min=j;
    end
end
%此时min确实为最小，赋值
times(i,2:3)=shadow(min,1:2);
end
```

综上所述，不同数据点所在区域修正情况如表 2.4 所示。所选示例地图匹配前后的上行轨迹比较如图 2.10 所示。

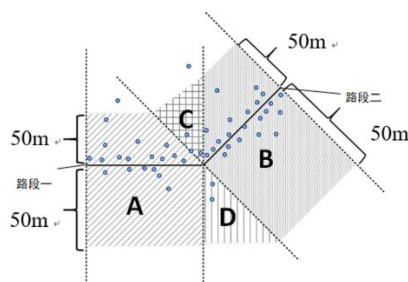


图 2.9 数据点与路段相对位置

表 2.4 不同数据点所在区域修正情况

数据点所在区域	修正后数据点位置	漂移距离
A	路段一	垂线段长度
B	路段二	垂线段长度
C	路段一或路段二	垂线段长度
D	两路段交点	数据点与交点连线长度 $<50m$
其他	删除数据行	大于 50m

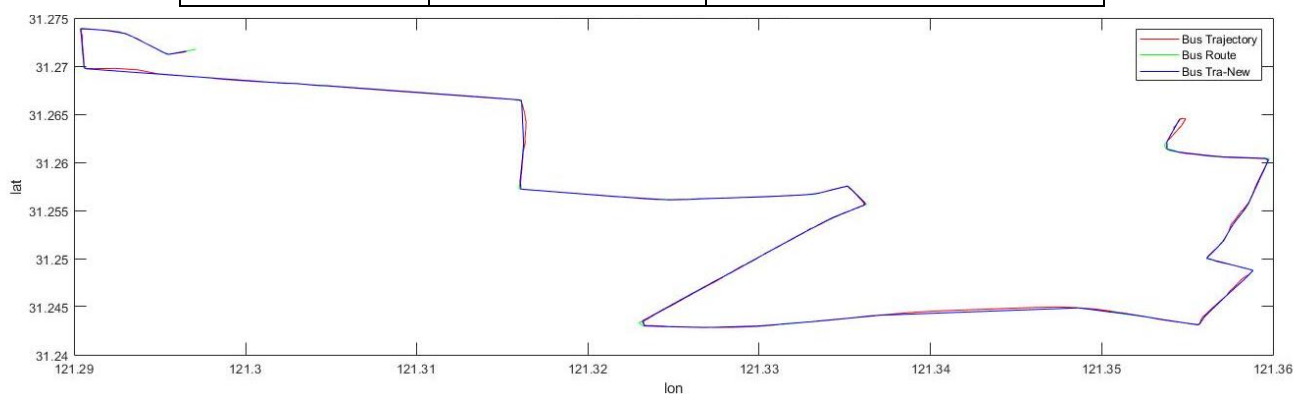


图 2.10 地图匹配前后 104 路公交轨迹展示



2.1.3. 时空轨迹转换

对每个数据点匹配到路段后,将公交位置数据的经纬度坐标转换为时空轨迹,换言之,即计算每个匹配后的数据点离始发站的距离。

任务书中将嘉定 104 路运行线路按照站点和交叉口(即地图数据中 TYPE=1 与 TYPE=2 的端点,值为 1 时是站点,值为 2 时是信号交叉口)进行分段,将这些节点作为基准点,始发站作为公交运行线路时空轨迹的原点,对相邻基准点间 $P_i(x_i, y_i)$ 和 $P_{i+1}(x_{i+1}, y_{i+1})$ 的距离 l_i 按照下式进行了计算。

由于在之前的地图匹配时,我进行矩形范围标记时采用的地图点包括 TYPE=1、TYPE=2 以及 TYPE=3 的中间点,这里为了方便,分段时依旧按照所有点进行分段,也就是说,所有地图的所给点即为基准点。

$$l_i = r * \arccos(\sin(x_i) * \sin(x_{i+1}) + \cos(x_i) * \cos(x_{i+1}) * \cos(y_i - y_{i+1}))$$

式中, x 和 y 分别代表经度和纬度, r 为地球半径。在 MATLAB 中可直接用下列语句计算:

$$l_i = \text{distance}(y_i, x_i, y_{i+1}, x_{i+1}, r);$$

式中, r 为地球半径,取 6371.004km。代码编写如下:

```
jizhun_num=n; %得到基准点个数
%计算基准点之间距离, jizhun_num个点, jizhun_num-1个距离,
%储存在jizuhn(i,5)格之中
r = 6371.004
for i=1:(jizhun_num-1)
    jizhun(i,5) = distance(jizhun(i,2),jizhun(i,1),jizhun(i+1,2),jizhun(i+1,1),r)
end
```

接着,计算数据点 Q 进行地图匹配之后得到的点 X 距离其最近的基准点之间的距离,同样使用 `distance` 命令计算,结果储存在 `temp(:,8)` 之中。选择上一个基准点的依据是地图匹配中得到的 `road` 字段。

```
%计算每个数据点的投影点距离上一个基准点之间的距离, 储存在temp8列
%上一个基准点由road字段得到,当road=999, 该数据点应该被删除, 令距离为0
temp_num=length(temp);
for i= 1:temp_num
    t=road(i,1);%对应路段起点
    if t~=999
        lat=temp(i,7);
        lon=temp(i,6);%投影点
        x=jizhun(t,1);
        y=jizhun(t,2);%起点
        temp(i,8)=distance(y,x,lat,lon,r);
    else
        temp(i,8)=0
    end
end
```

最后,计算数据点离始发站的距离 D , 公式如下,结果储存在 `temp(:,9)` 之中。注意,当 `road=999` 时,令该距离为 0。

$$D = d_k + \sum_{u=1}^{i-1} l_u$$



```
%计算数据点到起始点的距离D，L为路段起点到起始点的距离，  
%D储存在temp(:,9)中  
for i =1 :temp_num  
    if road(i,1)==999  
        temp(i,9)=0  
    else  
        %算路段起点到起始点的距离  
        L=0  
        for j=1:(road(i,1)-1)  
            L=L+jizhun(j,5);  
        end  
        D=L+temp(i,8);  
        temp(i,9)=D;  
    end  
end
```

最后绘制时空图轨迹，以 9 月 6 日 104 路公交第一次周转的上行 GPS 轨迹为例，得到轨迹图 2.11 与图 2.12。其中，将每个时间点的累积行程距离与上一个时间点进行对比，若小于上一个时间点，则将该点标红，示意出现错误。

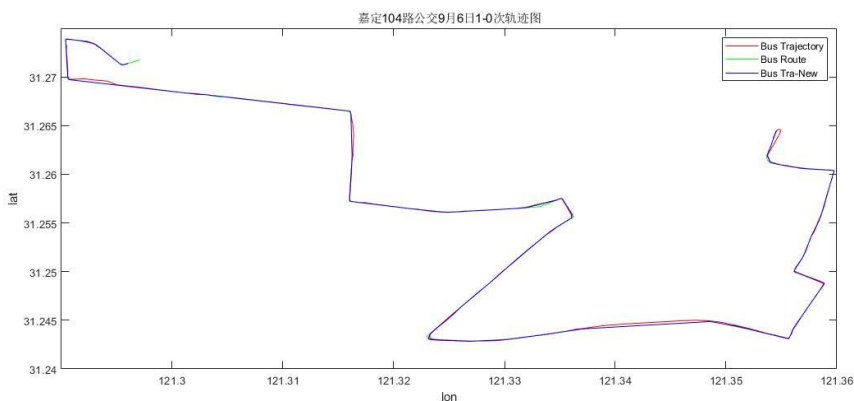


图 2.11 104 路公交 9.6-1-0 轨迹展示

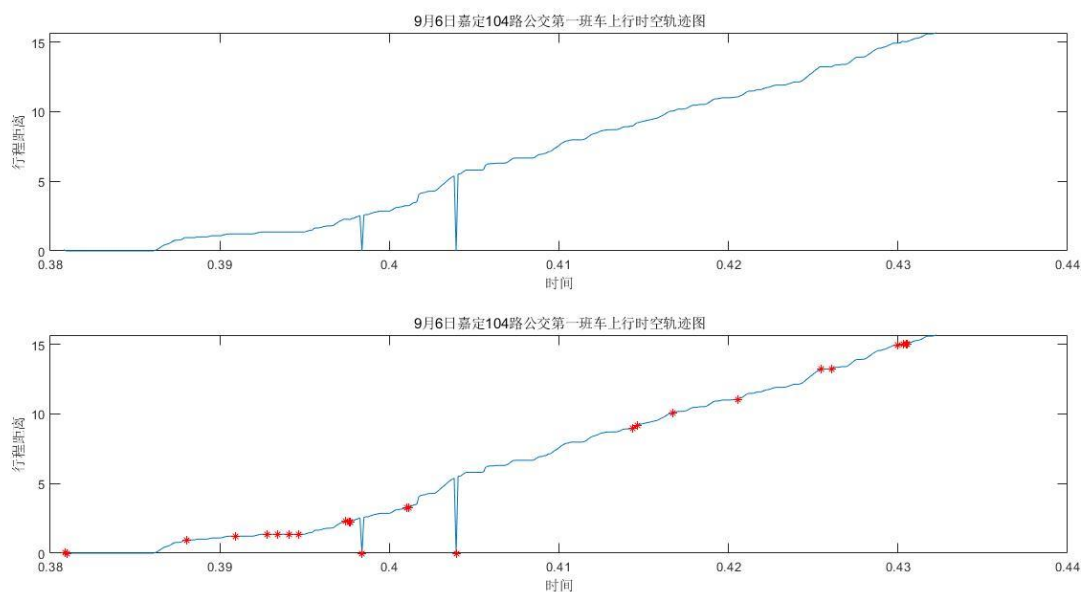


图 2.12 9 月 6 日嘉定 104 路公交第一班车上行时空轨迹图



2.1.4. 二次修正

由于简单的地图匹配可能会在公交车辆转弯的交叉口产生匹配错误，即从修正后的数据点得到的公交运行轨迹有一个掉头的过程，不符合公交的运行规律。因此，还需按照公交运行线路及其规律对数据点进行修正。这里主要是根据前后轨迹点的累计行程距离差以及公交现实行程车速限制，剔除异常数据。

注意，在地图匹配之中还有一类数据：漂移距离大于 50m 的 GPS 数据点，这类数据点同样要进行删除。经过 2.1.3 时空轨迹转换，这类数据已经通过 `temp(:,9)=0` 进行了标记。这里再次使用 `flag` 对其进行赋值，令 `flag=2`。

如图 2.12 所示，可以很明显的发现累积行程距离差为负值（车辆倒退）的点。将这类点以及相邻 GPS 段的行程车速超过 80km/h 的点视为异常数据，并给予标记 `flag=1`，筛选代码如下所示。

```
%%用flag进行标记
%flag初始值为0
%当flag=1时，说明该点为异常数据，需要二次修正
%当flag=2时，说明该数据行需要删除
flag=zeros(temp_num,1);
for i = 2:temp_num
    %漂移距离大于50m的数据需要删除
    if temp(i,9)==0
        flag(i,1)=2;
    else
        %距离为负
        s=temp(i,9)-temp(i-1,9);
        if s<0
            flag(i,1)=1;
        end
        %车速超过80km/h
        time_cha=time(i,1)-time(i-1,1);
        v=s*1000/(time_cha*86400);
        if v*3.6>80
            flag(i,1)=1;
        end
    end
end
end
```

注意，由于 MATLAB 读入的时间为数值格式，故需要对其进行换算。一天 24h，1h=3600s，则数值格式中 0-1 被赋值给了 86400s。计算相邻两个 GPS 轨迹时间相减后的时间数值，将其乘 86400 即为时间相差的秒数。为方便计算，将距离 km 转化为 m，得到速度的单位为 m/s。由于 1m/s=3.6km/h，则阈值 80km/h=22.22m/s，并进行比较。

对于标记为 `flag=1` 的数据进行二次修正。修正方法为：在前面地图匹配中，如果数据点处于 C 区域（即存在可替代匹配点），我们已经事先存储了所有可能的备用路段。对于处于 C 区域的数据异常，寻找备用路段中的新的匹配点，路段被使用后从备用路段中删除。每次修正一个 GPS 数据后重新判断线路 GPS 异常情况，直到找到正确匹配点或不存在可选备用路段。若异常数据不存在可选的备用路段，则删除该匹配点。依次判断每个 GPS 点的异常情况并做相应处理，直到整条路径不存在异常数据。

为方便计算，这里设置判断函数 `yichang.m`，函数表示如下：



```
function TF=yichang(x,y,time_cha,dis_last,new_rec)
%YICHANG 判断新的投影点是否异常
% 输入参数,(x,y)为新的投影点,i为数据点编号,
% time_cha为该点与上一点的时间差值,单位为s
% dis_last是上一点的累积行程距离差
% new_rec是新投影点所在的路段矩形范围起点
% 输出值TF1*2
% TF(1,1) 1不异常,2异常
% TF(1,2) 现在的累积行程距离
% TF(1,1)=1;%赋初值;
%计算新的累积行程距离差
%新路段起点到起始点距离
r=6371.004;
L=0;
for j=1:(new_rec-1)
    L=L+jizhun(j,5);
end
%新投影点到新路段起点距离
lat=jizhun(new_rec,7);
lon=jizhun(new_rec,6);%路段起点
l=distance(lat,lon,y,x,r);
%新累积行程距离
D=L+l;
dis_now=D;
%距离为负
s=dis_now-dis_last;
if s<0
    TF(1,1)=2;
end
%车速超过80km/h
v=s*1000/time_cha;
if v*3.6>80
    TF(1,1)=2;
end
TF(1,2)=dis_now;
end
```

当不存在备用路段时,将 flag 值改为 2。

```
%对于异常点
if (flag(i,1)==1)&&(flag(i,2)==1)
    %不存在备用路段时,将其转化为需要删除的匹配点
    flag(i,1)=2;
end
```

当存在备用路段时,计算新投影点,并判断新点是否满足要求。将满足要求的点 flag 更新为 0。

```
%对于剩余备用路段
if t==2 %还有一个备用路段
    a=temp(i,1:2);%%a是数据点的经纬度坐标

    %得到新投影点
    new_rec=rec_flag(i,2); %新路段对应矩形序号
    x1=standard(new_rec,1);
    y1=standard(new_rec,2);
    x2=standard(new_rec+1,1);
    y2=standard(new_rec+1,2);
    x3=a(1,1);
    y3=a(1,2);
    %得到新点的投影点坐标与垂足距离
    %p(1,1:2)为坐标,p(1,3)为距离
    p=get_footpoint(x1,y1,x2,y2,x3,y3);

    %判断新的点是否是异常点,1不异常,2异常,TF为判断函数
    dis_lastnode=temp(i-1,9);
    TF=yichang(p(1,1),p(1,2),time_cha(i,1),dis_lastnode,new_rec,jizhun);
    if TF(1,1)==1
        %不异常则更新投影点数据
        temp(i,6:7)=p(1,1:2);
        temp(i,9)=TF(1,2);
    else
        flag(i,1)=2 %仍然异常则没有备用线路了,将该匹配点删除
    end
end
```



二次修正后重新绘图，注意不对 flag=2 的点进行绘制，得到图 2.13。

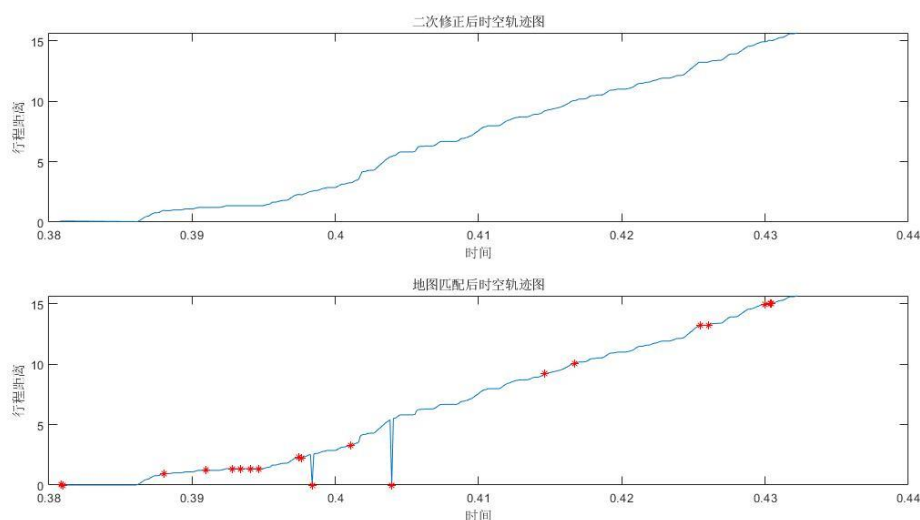


图 2.13 二次修正前后时空轨迹图对比

2.2. 数据剔除和修复

```
%lujing 12经纬度 3位累积行程距离 4为时间
%判断数据修复的位置并利用线性内插修复
%时间差>30s或距离大于50m
%用xiufu储存修复后的数据，并利用k计数
xiufu=zeros(500,4);
xiufu(1,:)=temp(1,:);
k=1;
for i=2:temp_num
    time_cha=(temp(i,4)-temp(i-1,4))*86400;%s
    dis_cha=(temp(i,3)-temp(i-1,3))*1000;%m
    if time_cha >30%判断时间差
        k=k+1;
        xiufu(k,:)=1/2*(temp(i,:)+temp(i-1,:));
        %插入原来数据
        k=k+1;
        xiufu(k,:)=temp(i,:);
    else if dis_cha>50 %判断距离差
        k=k+1;
        xiufu(k,:)=1/2*(temp(i,:)+temp(i-1,:));
        %插入原来数据
        k=k+1;
        xiufu(k,:)=temp(i,:);
    end
end
%均满足要求
if time_cha<=30 && dis_cha<=50
    k=k+1;
    xiufu(k,:)=temp(i,:);
end
end
dis_cha    39.6299
i          372
k          509
len        442
num        442x4 double
raw        442x4 cell
t          372
temp       442x4 double
temp_num   372
text       0x0 cell
time_cha   10.0000
xiufu      509x4 double
```

进行上述修正后，所有 flag=2 的 GPS 轨迹数据点均为需要剔除的数据。除此之外，对于字段 deadheading=“空驶”的数据，也将其 flag 记录为 2，方便讨论。

当一个轨迹中，缺失数据过多时，认为该轨迹异常进行剔除。但是需要记录该轨迹信息，避免后面在进行指标分析时出现问题，如车头时距，发车间隔等。

对于一般的轨迹，进行缺失数据修复。针对前面处理后的数据，对于在一个长时段内（大于 30s）存在 GPS 数据缺失，或者相邻两匹配点间距离大于 50m 的情况，进行数据补全，相当于增大位置信息采样频率，以在后续确定站点进出站信息时减小定位误差。

这里采用线性内插补全缺失的数据。具体代码如下所示。

同样以 9 月 6 日第一天第一班车的上行轨迹为例，修复前共有 372 个数据点，修复后得到 509 个数据点。

2.3. 补充数据基础

为下一步具体分析线路/路段运营指标，先补充数据基础，完善起点站出站、终点站进站以及中间停靠站进出站信息。

起点站的出站除了考虑离站点的距离外，还需考虑速度信息即确保车辆已启动出发。采用的思路是首先找到第一个距离起点站大于 25m 的点，进一步利用后几个 GPS 点的累计距离和时间去判断速度条件（由于出站时速度较慢，速度下限可取 30km/h）。距离和速度都满足的数据为合理的出站数据。

中间停靠站的进出站由于不存在长时间的停站，仅考虑采用距离条件确定进出站。即首先确定各个站点距离起点站的累计距离，根据站点范围阈值（25m），根据各个 GPS 点求得的实际累计距离判断是否进出站。

最后终点站的进站判断也采用站点范围阈值（25m）确定。

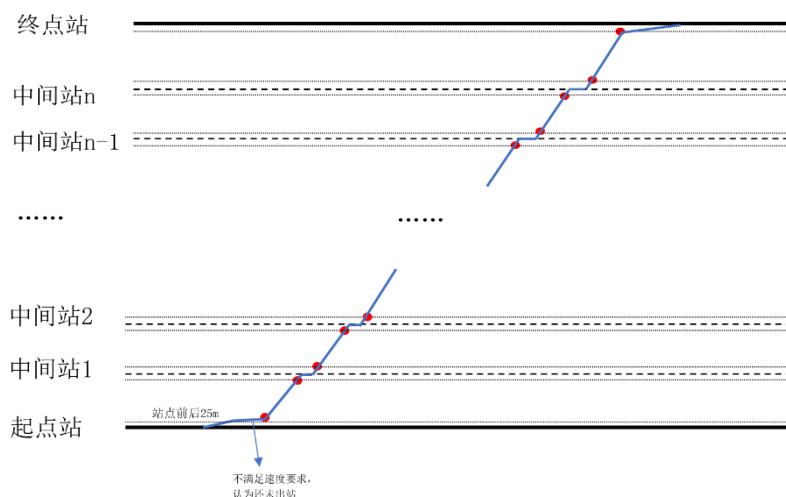


图 2.14 确定起点站出站、终点站进站及中间站进出示意图

同样以嘉定 104 路公交 9 月 6 日第一班上行轨迹为例，该路公交上行共有 24 个站点，起始站为真新车站，终点站为封浜汽车站。首先，计算出这 24 个站点距离起始站：真新车站的累计行程距离，具体代码如下。

```
%共有24个站点，k为站点个数
%确定每个数据点距离起始站的累计行程距离，储存在stan_s第六列，第一站为0
stan_s(1,6)=0;
L=0;
for i=2: stan_s_num
    L=L+stan_s(i-1,5);
    stan_s(i,6)=L;
end
%取出其中的站点数据，储存在zhan中
%zhan 12为经纬度 3为点的类型 4为距离起始点距离
%
k=0
zhan=zeros(24,4);
for i=1: stan_s_num
    if stan_s(i,4)==1
        k=k+1;
        zhan(k,1:2)=stan_s(i,1:2);
        zhan(k,3)=stan_s(i,4);
        zhan(k,4)=stan_s(i,6);
    end
end
```



接着寻找进站点与出站点。对于 104 路上行轨迹而言，第 2 至第 24（即终点站）具有进站点。进站点都是通过站点范围阈值（设定为 25m）进行判断，即对一个站点而言，在所有与其距离小于 25m 的 GPS 轨迹点当中，站前与其距离最远的 GPS 点就是该站点的进站点。由于之前已经计算出每个 GPS 轨迹点的累计行程距离，也已经计算出每个站点的累计行程距离，故只需要按照顺序判断即可。对于出站点，则需要计算该点与站点的对应距离以及下一点与站点的对应距离，当该点满足要求而下一点不满足时，方为对应站点的出站点。具体代码如下。

```
%寻找进站点2-24，将进站点的GPS轨迹点数据记录到zhan56列之中
%第一个站的56列为0，在temp第5列中进行标记，5为0表示为中间点，1为进站点 2为出站点
zhan_s(1,5)=0;
zhan_s(1,6)=0;
x=2;
for i=1:temp_num
    s=(zhan_s(x,4)-temp(i,3))*1000;
    if s<25
        zhan_s(x,5:6)=temp(i,1:2);%找到进站点
        temp(i,5)=1;
        x=x+1;%对x进行更新，寻找下一个站的进站点
    end
    %对最近站进行修改
end

%寻找出站点1-23，78列距离10
y=2;
zhan_s(24,7)=0;
zhan_s(24,8)=0;
for i=1:temp_num-1
    s=(temp(i,3)-zhan_s(y,4))*1000;%该点
    s1=(temp(i+1,3)-zhan_s(y,4))*1000 %下一个点
    if s<=25 && s1>25 %当该点<25且下一个点>25时，该点为距离该站驶出阈值范围内最远的点
        zhan_s(y,7:8)=temp(i,1:2);%找到出站点
        temp(i,5)=2;
        y=y+1;%对y进行更新，寻找下一个站的出站点
    end
    %对最近站进行修改
end
```

其中起点站的出站点除去距离要求外，还需要计算其速度，当该 GPS 点对应的速度值大于速度阈值（30km/h）时，方为起始站的出站点。

绘图，得到图 2.15。

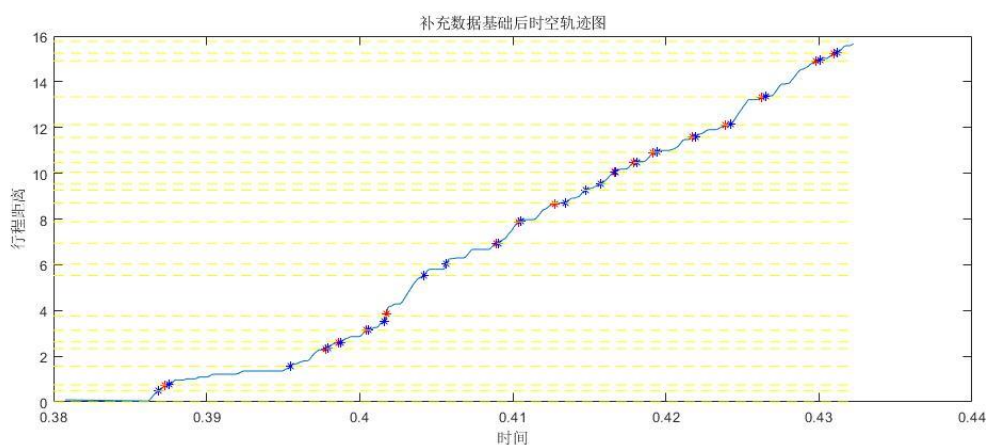


图 2.15 补充数据基础后的时空轨迹图

3. 任务 2：公交运行状况定性分析

3.1. 公交运行轨迹分析

时空图能够直观的表达全天所有正常运行的公交车辆的运行情况。从中可以直接观察到公交车辆发车间隔变化、不同时间段的车头时距分布以及不同时间段的串车现象等。由此得到初步定性结论。在这里我选的对 9 月 7 日嘉定 104 路的上下行公交运行轨迹分别进行分析，得到公交运行上行时空图与公交运行下行时空图如图 3.1 与 3.2 所示。

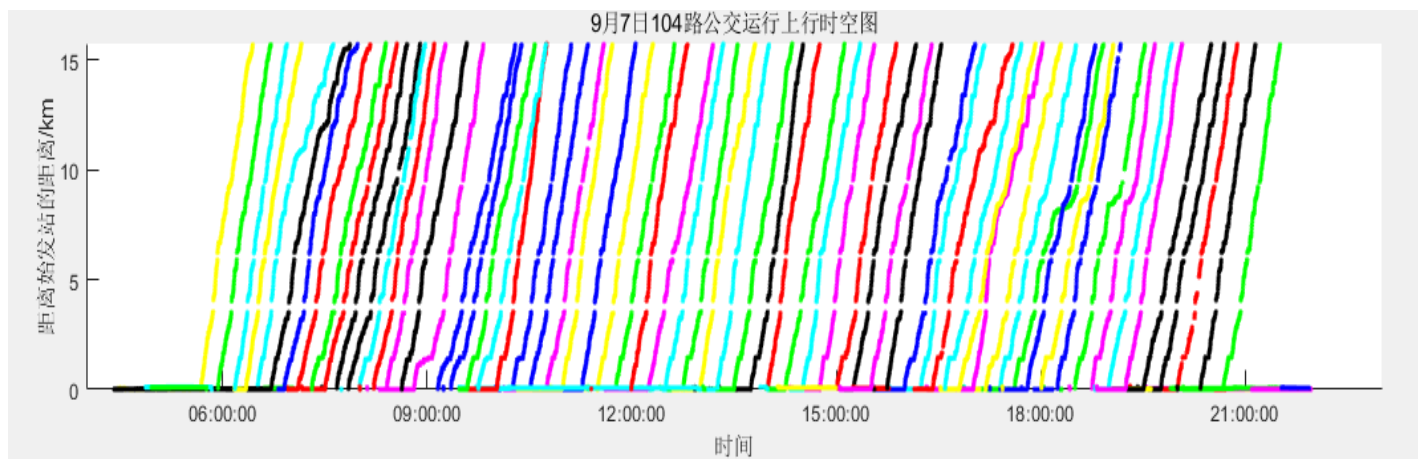


图 3.1 9 月 7 日嘉定 104 路公交运行上行时空图

观察图 3.1，分析 9 月 7 日嘉定 104 路公交运行上行时空图。从图中可以观察到：

- ① 所有的这些轨迹的时间范围从上午 5:00 左右到夜晚 10:00 左右；
- ② 可以判断出在多条轨迹出现折痕的位置便是上行图中的站点；
- ③ 在上午 9:00 左右的轨迹以及下午 18:00 左右的轨迹时间范围较广，刚好处于通常意义上的早晚高峰时间段，因此判断图中这些公交车辆可能遇到了堵车现象；
- ④ 在上午 9:00 左右的轨迹以及下午 18:00 左右的车头时距较其他时间小，考虑为高峰期缓解客流压力公交公司采取的调度安排；同时可以观察到早高峰的主要延误出现在累积行程距离为 2~3km 的范围，而晚高峰的主要延误出现在 8~9km 的范围，这可能与站点所在街道有关，有待后续分析；
- ⑤ 上午 8:30、中午 10:30 与下午 17:30 均出现两辆公交车时空图重合的现象，说明这些时段的公交调度安排有待改进；
- ⑥ 下午 18:00 左右出现后发车的公交车超过先发车的公交车的现象（也就是“串车”现象），从中可以想象出两辆公交车在路线中的行驶状况是十分不满足公交调度安排原则的，因此也需要采取进一步的改进措施。

绘图代码（以上行轨迹为例）如下：



```
1 %上行时空轨迹
2 %读取信息
3 [num, text, raw]=xlsread('0907s.xlsx');
4 temp=num;%1时间 23经纬度
5 [num, text, raw]=xlsread('jizhun.xls'); %12经纬度 3空 4范围 5基准点之间距离
6 jizhun=num;
7 [num, text, raw]=xlsread('rectangle_shang.xlsx');
8 rec_tangle=num;
9
10 [num, text, raw]=xlsread('road_s.xlsx');
11 road=num;
12 temp_num=length(temp);
13 %计算每个数据点的投影点距离上一个基准点之间的距离，储存在temp8列
14 %上一个基准点由road字段得到，当road=999，该数据点应该被删除，令距离为0
15
16 r=6371.004;
17 h=waitbar(0, '计算中，请稍后!');
18 len=temp_num
19 for i= 1:temp_num
20     t=road(i,1):%对应路段起点
21     if t~=999
22         lat=temp(i,7);
23         lon=temp(i,6):%投影点
24         x=jizhun(t,1);
25         y=jizhun(t,2):%起点
26         temp(i,8)=distance(y,x,lat,lon,r);
27     else
28         temp(i,8)=0
29     end
30     str=['计算中...', num2str(100*i/len), '%'];
31     waitbar(i/len,h,str);
32 end
33 close(h);
34 xlswrite('s_dis_GPS2JZ', temp(:,8));
35 %计算数据点到起始点的距离D，L为路段起点到起始点的距离，
36 %D储存在temp(:,9)中
37 [num, text, raw]=xlsread('0907s.xlsx');
38 temp=num;%1时间 23经纬度
39 [num, text, raw]=xlsread('jizhun.xls'); %12经纬度 3空 4范围 5基准点之间距离
40 jizhun=num;
41 [num, text, raw]=xlsread('rectangle_shang.xlsx');
42 rec_tangle=num;
43 [num, text, raw]=xlsread('road_s.xlsx');
44 road=num;
45 temp_num=length(temp);
46 [num, text, raw]=xlsread('s_dis_GPS2JZ.xls');
47 temp(:,8)=num;
48
49 %算每个点到起始点的累计距离，储存在jizhun(:,6)之中
50 jizhun(1,6)=0;
51 for i=2:length(jizhun)
52     jizhun(i,6)=jizhun(i-1,6)+temp(i-1,8);
53 end
54 xlswrite('jizhun_s', jizhun);
55
56 xlswrite('temp_1-8', temp);
57
58 %*****从这里开始做
59 [num, text, raw]=xlsread('road_s.xlsx');
60 road=num;
61 [num, text, raw]=xlsread('temp_1-8.xls');
62 temp=num;
63 temp_num=length(temp); %8为到上一个基准点的距离
64 [num, text, raw]=xlsread('jizhun_s.xls'); %12经纬度 3空 4范围 5基准点之间距离 6到起始点距离
65 jizhun=num;
```



```
72 - h=waitbar(0,'计算中,请稍后!');
73 - len=temp_num;
74 - %第一个点距离为0
75 - temp(1,9)=0;
76 - for i =2 :temp_num
77 -     if road(i,1)==999    %%999数据行删除 令距离为0
78 -         temp(i,9)=0
79 -     else
80 -         %距离=dot到最近基准点距离+该基准点到起始点距离
81 -         k=road(i,1);
82 -         D=jizhun(k,6)+temp(i,8);
83 -         temp(i,9)=D;
84 -     end
85 -     str=['计算中...',num2str(100*i/len),'%'];
86 -     waitbar(i/len,h,str);
87 - end
88 - close(h);
89 - xlswrite('temp_1-9',temp);
90 -
91 - %读取时间
92 - [num,text,row]=xlsread('07st.xlsx');
93 - time=num(:,1);
94 - h=waitbar(0,'计算中,请稍后!');
95 - color=['b';'g';'r';'c';'m';'y';'k'];
96 - color_num(:,1)=color(mod(temp(:,5),7)+1,1);
97 -
98 - figure;
99 - h=waitbar(0,'计算中,请稍后!');
100 - for i = 1:temp_num
101 -     if temp(i,9)~=0
102 -         scatter(time(i,1),temp(i,9),color_num(i,1),'');
103 -         hold on
104 -     end
105 -     str=['计算中...',num2str(100*i/len),'%'];
106 -     waitbar(i/len,h,str);
107 - end
108 - close(h);
109 - xlabel('时间');
110 - ylabel('距离始发站的距离/km');
111 - title('9月7日104路公交运行上行时空图');
112 - set(gca,'XLim',[4*1/24 23/24]);
113 - datetick('x',13);
114 -
```



3.2. 车辆使用情况分析

选择 9 月 7 日嘉定 104 路公交的 GPS 数据，绘制车辆使用情况图，蓝色为上行，红色为下行，以此分析车辆使用情况。

由于 9 月 7 日共有 13 辆车，为每辆车设定一个车牌编号，对应关系如表 3.1 所示。

表 3.1 公交车车牌与车辆代号对应关系

9 月 7 日公交车车牌号与对应号数					
车牌号	沪 A-08706D	沪 A-00576D	沪 A-00577D	沪 A-01202D	沪 A-01339D
车辆代号	1	2	3	4	5
车牌号	沪 A-01722D	沪 A-01880D	沪 A-01976D	沪 A-03593D	沪 A07360D
车辆代号	6	7	8	9	10
车牌号	沪 A07695D	沪 A-09169D	沪 A-09359D		
车辆代号	11	12	13		

绘图，代码如下，结果如图 3.3 所示：

```
%1为车辆ID，共13辆车 2kong 3时间 45经纬度
bus_num=13;
%绘图
figure;
scatter(temp_s(:,3),temp_s(:,1),'b','.');
hold on
scatter(temp_x(:,3),temp_x(:,1),'r','.');
datetick('x',13);
set(gca,'XLim',[4*1/24 23/24]);
set(gca,'YLim',[0 14]);
set(gca,'YTick',[0:1:14]);
set(gca,'YTickLabel',{'','沪A-08706D','沪A-00576D','沪A-00577D','沪A-01202D','沪A-01339D','沪A-01722D','沪A-01880D','沪A-01976D','沪A-03593D','沪A07360D','沪A07695D','沪A-09169D','沪A-09359D'});
ylabel('公交车牌号');
title('9月7日嘉定104路公交车使用情况');
```

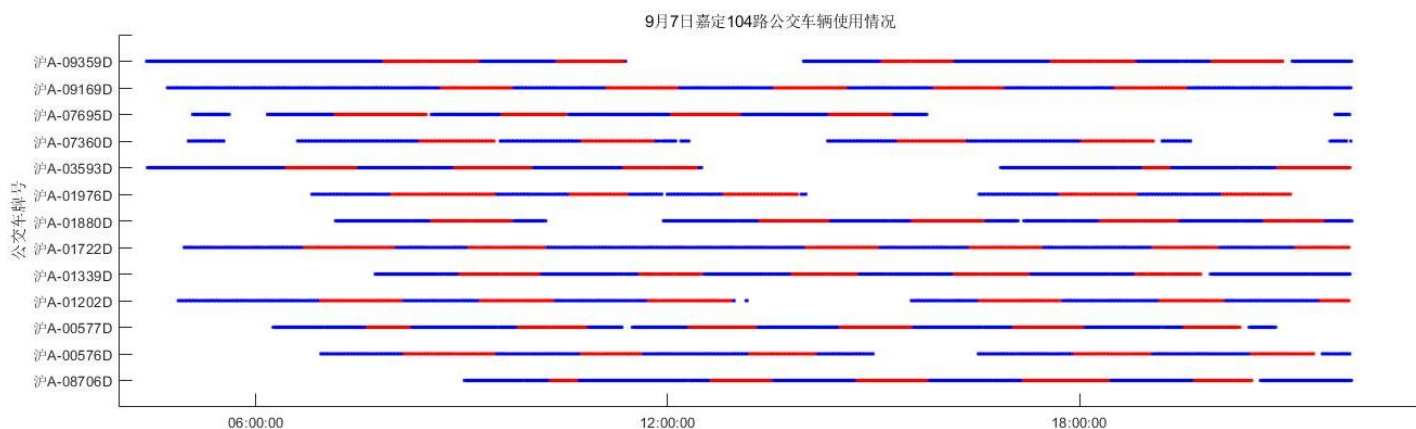


图 3.3 9 月 7 日嘉定 104 路公交车使用情况图

观察图 3.3，发现：

- ① 9 月 7 日共有 13 辆车在运行，其中有两辆在 4:42 分后投入运营，而所有车的 GPS 轨迹在十点前截止；



- ② 上午九点附近所有的十三辆车均在运营；下午六点附近除“沪 A-07695D”外剩余十二辆车均在运营；
- ③ “沪 A-29169D”与“沪 A-01722D”几乎全天都在运营，运营时间范围长达 17 个小时，而其余车都有或长或短的休息时间；
- ④ 观察单次上下行的行驶时间，发现上行公交车有多条轨迹较长，可能出现了堵车等现象；
- ⑤ 有部分车次上下行时间较短，为空驶状态，路线与正常路线不同，速度较快。

3.3. 线路运行指标

本次课程设计还会线路的运行指标进行分析，这里的指标包括：出发准点率指标、时刻表偏差、发车间隔指标分析、运营时间指标分析以及 Layover 时间分析五个部分。

3.3.1. 出发准点率

准点定义为“早一晚二”，也就是说，将出发时间在时刻表前一分钟以及时刻表后两分钟内出发的公交车视为准点出发。

首先对时刻表进行处理，得到每趟班次的早一晚二对应时间阈值，代码如下。

```
%用timetable储存，第一列为车牌号，第45为出发到达的时刻表时间，67为出发早一晚二对应时间，89为到达早一晚二的对应时间
len=length(timetable);
%timetable第一列为车牌序号，五六为出发到达时刻表
for i = 1: len
    chufa=timetable(i,4);
    daoda=timetable(i,5);
    %0~1=0~86400s
    %1min=1/1440
    %早1
    timetable(i,6)=chufa-1/1440;
    timetable(i,8)=daoda-1/1440;
    %晚2
    timetable(i,7)=chufa+2/1440;
    timetable(i,9)=daoda+2/1440;
end
```

判断当前是否准点到达的函数“zhuandian.m”如下代码所示，函数返回值为 1 时说明准点出发，值为 2 时说明早点或晚点。



```
1 function s = zhundian(datebelong, sx, bus, time)
2 %ZHUNDIAN 此处显示有关此函数的摘要
3 % datebelong: 输入数据所属的天数 只有6 7 8 9 10 五个值
4 % sx: 表明是上行还是下行 上行值为0 下行值为1
5 % bus: 表明是哪一辆公交车, 取值为1~13的整数, 共十三个值可取
6 % time 为需要判断的数值
7 % 注意: 早一晚二阈值由于之前计算过了, 因此直接从timetable中读入
8 % timetable第一列为车牌号, 第45为出发到达的时刻表时间, 67为出发早一晚二对应时间, 89为到达早一晚二的对应时间
9 % s为返回值, 值为1时说明是准点, 2说明早点或者晚点
10 %读取timetable, 读取到timetable中
11 if sx==0
12     switch datebelong
13     case 6
14         [num, text, raw]=xlsread('tt06s.xlsx');
15         timetable=num;
16     case 7
17         [num, text, raw]=xlsread('tt07s.xlsx');
18         timetable=num;
19     case 8
20         [num, text, raw]=xlsread('tt08s.xlsx');
21         timetable=num;
22     case 9
23         [num, text, raw]=xlsread('tt09s.xlsx');
24         timetable=num;
25     otherwise
26         [num, text, raw]=xlsread('tt10s.xlsx');
27         timetable=num;
28     end;
29 else
30     switch datebelong
31     case 6
32         [num, text, raw]=xlsread('tt06x.xlsx');
33         timetable=num;
34     case 7
35         [num, text, raw]=xlsread('tt07x.xlsx');
36         timetable=num;
37     case 8
38         [num, text, raw]=xlsread('tt08x.xlsx');
39         timetable=num;
40     case 9
41         [num, text, raw]=xlsread('tt09x.xlsx');
42         timetable=num;
43     otherwise
44         [num, text, raw]=xlsread('tt10x.xlsx');
45         timetable=num;
46     end;
47 end
48 %得到该辆车在该天该顺序下的时刻表, 一般为4~6趟安排, 这里冗余8趟
49 %用bus_time储存
50 %第一列为车牌号, 第45为出发到达的时刻表时间, 67为出发早一晚二对应时间, 89为到达早一晚二的对应时间
51 len=length(timetable);
52 bus_time=zeros(8,9);
53 bus_time(:,1)=bus;
54 k=0;%对bus_time计数
55 for i = 1: len
56     if timetable(i,1)==bus
57         k=k+1;
58         bus_time(k,:)=timetable(i,:);
59     end
60 end
61
62 %对数据进行判断, 并返回值s, s初值为0, 返回值为1时说明是准点, 2说明早点或者晚点
63 s=0;
64 for i=1:k
65     if time < bus_time(i,7) && time > bus_time(i,6)
66         s=1;
67     end
68 end
69 %对时刻表判断后没有符合的对应时刻, 则该班早点或者晚点了
70 if s==0
71     s=2
72 end
73 end
74
```



在本次课程设计中，分别计算嘉定 104 路公交在 9 月 6 日、7 日、8 日、9 日与 10 日的上下行首末班车出发准点率、所有班次平均出发准点率（包括全天、早高峰、晚高峰平均水平）。其中，早晚高峰的定义为：

早高峰区间：7：00-9：30，晚高峰区间：16：30-19：00。

绘图分别得到图 3.4、3.5、3.6。

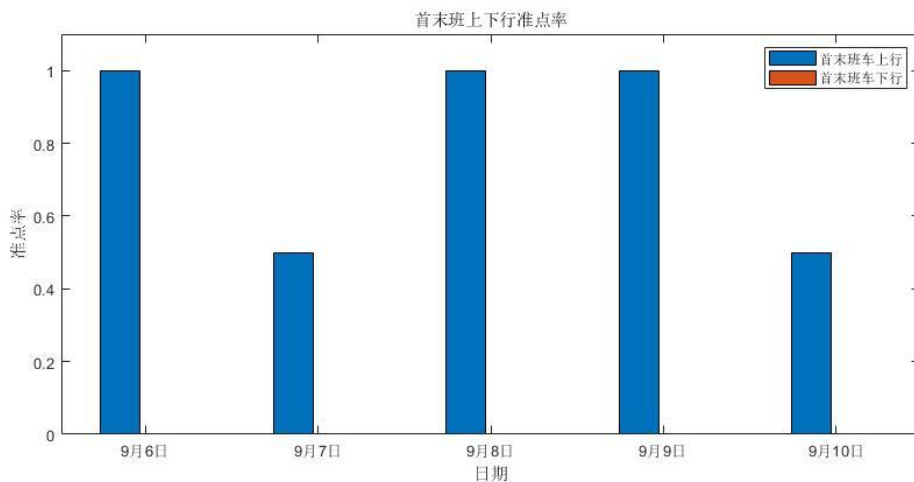
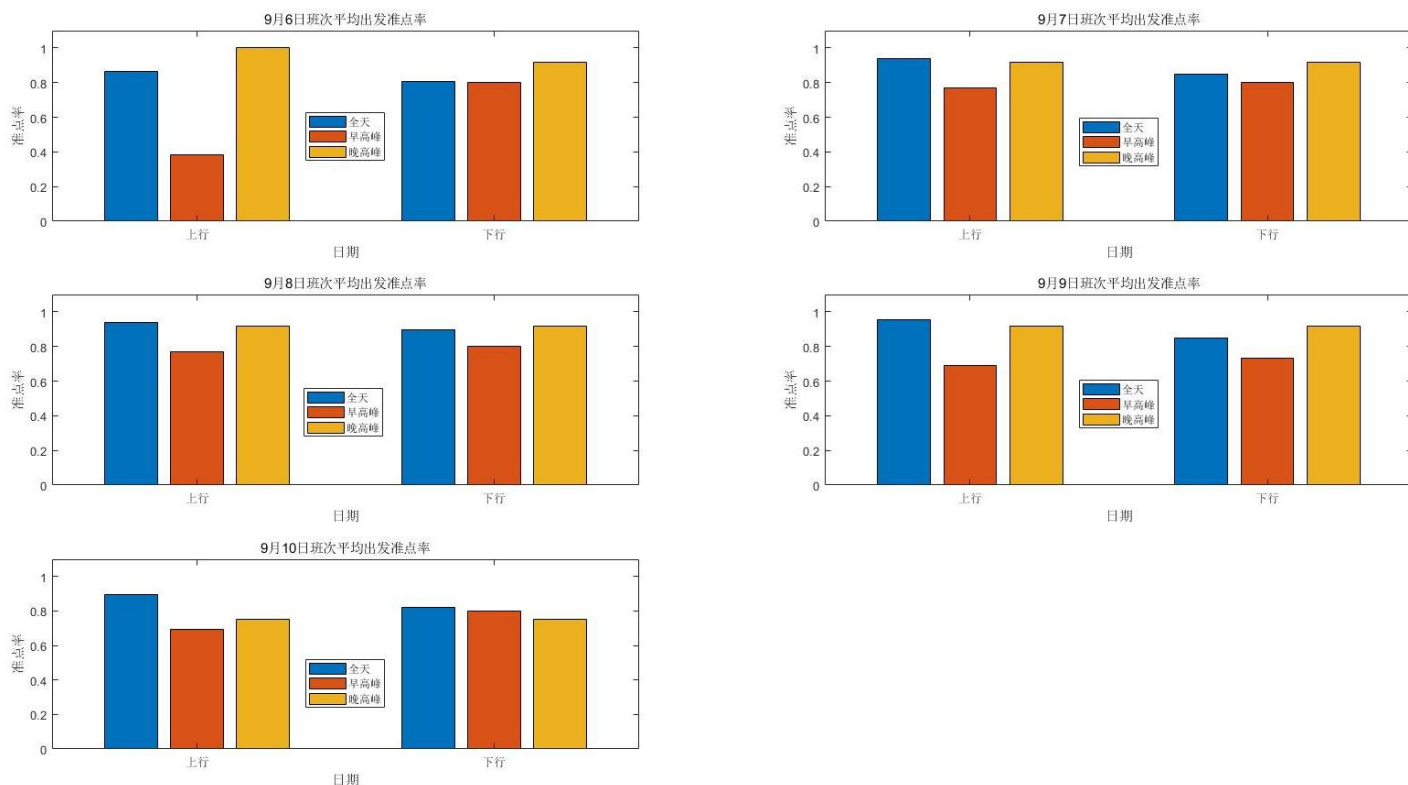


图 3.4 嘉定 104 路公交首末班车上下行准点率，（1）为首班车，（2）为末班车

嘉定 104 路公交的首末班车上下行准点率如图 3.4 所示。观察发现：从 9 月 6 日至 9 月 10 日这五天中，下行（从封浜汽车站到真新车站）首末班车的准点率均为 0，上行的准点率则为 0.5 与 1。因此，需要对嘉定 104 路公交的下行时刻表进行调整。

图 3.5 嘉定 104 路公交全天与早晚高峰平均出发准点率





嘉定 104 路公交全天与早晚高峰平均出发准点率如图 3.5 所示。观察发现：

- ① 对于全天准点率，9 月 6 日至 10 日得到的下行准点率均比上行准点率低，且上行准点率约在 90%，下行准点率约在 85%；
- ② 对于全天上行准点率，9 月 9 日的全天上行准点率最高约为 95%，9 月 6 日的全天上行准点率最低，约为 85%；
- ③ 对于全天下行准点率，9 月 8 日的全天下行准点率最高，约为 90%，9 月 6 日的全天下行准点率最低，约为 80%；
- ④ 早高峰准点率均低于全天准点率，且上行的早高峰准点率均低于下行的早高峰准点率；
- ⑤ 对于早高峰上行准点率，9 月 6 日的早高峰上行准点率最低，约为 40%，其余四天的早高峰上行准点率约在 70%左右；
- ⑥ 对于早高峰下行准点率，9 月 6 日至 9 月 10 日这五天的早高峰下行准点率均在 80%附近；
- ⑦ 对于晚高峰准点率，9 月 6 日的晚高峰准点率高于全天准点率，9 月 10 日的晚高峰准点率低于全天准点率；
- ⑧ 对于晚高峰上行准点率，除 9 月 6 日外，其余四天的晚高峰上行准点率均低于全天上行准点率，9 月 6 日的晚高峰上行准点率约为 1，9 月 10 日的晚高峰上行准点率约为 75%；
- ⑨ 对于晚高峰下行准点率，除 9 月 10 日外，其余四天的晚高峰下行准点率均高于全天下行准点率，9 月 10 日的晚高峰下行准点率最低，约为 75%。



3.3.2. 时刻表偏差

经过 3.3.1 出发准点率分析发现嘉定 104 路公交全天平均准点率最低的是 9 月 6 日。因此分析该天出发/到达与时刻表的偏差。

观察时刻表数据，发现 9 月 6 日至 9 月 10 日这五天中的时刻表除执行车辆不同外，其余出发与到达时刻均相同，因此从 3.3.1 中得到标准的时刻表数据，并用于以下应用。

起点站的出站点要求 GPS 坐标在起点站坐标的 25m 范围内，同时还需要计算其速度，当该 GPS 点对应的速度值大于速度阈值（30km/h）时，方为起始站的出站点；终点站的进站点要求 GPS 坐标在终点站坐标的 25m 范围内。

以 9 月 6 日的 75 条上行轨迹为例，计算出发/到达时刻表偏差。首先得到这 75 条轨迹对应的终点站进站时刻与起始站出站时刻；然后计算时刻表偏差。

同理获得 9 月 6 日的 74 条下行轨迹数据的出发/到达时刻表偏差。

绘图，由于为离散数据，故利用 MATLAB 中的 stem 函数进行绘制，出发/到达与时刻表偏差如图 3.6 所示。

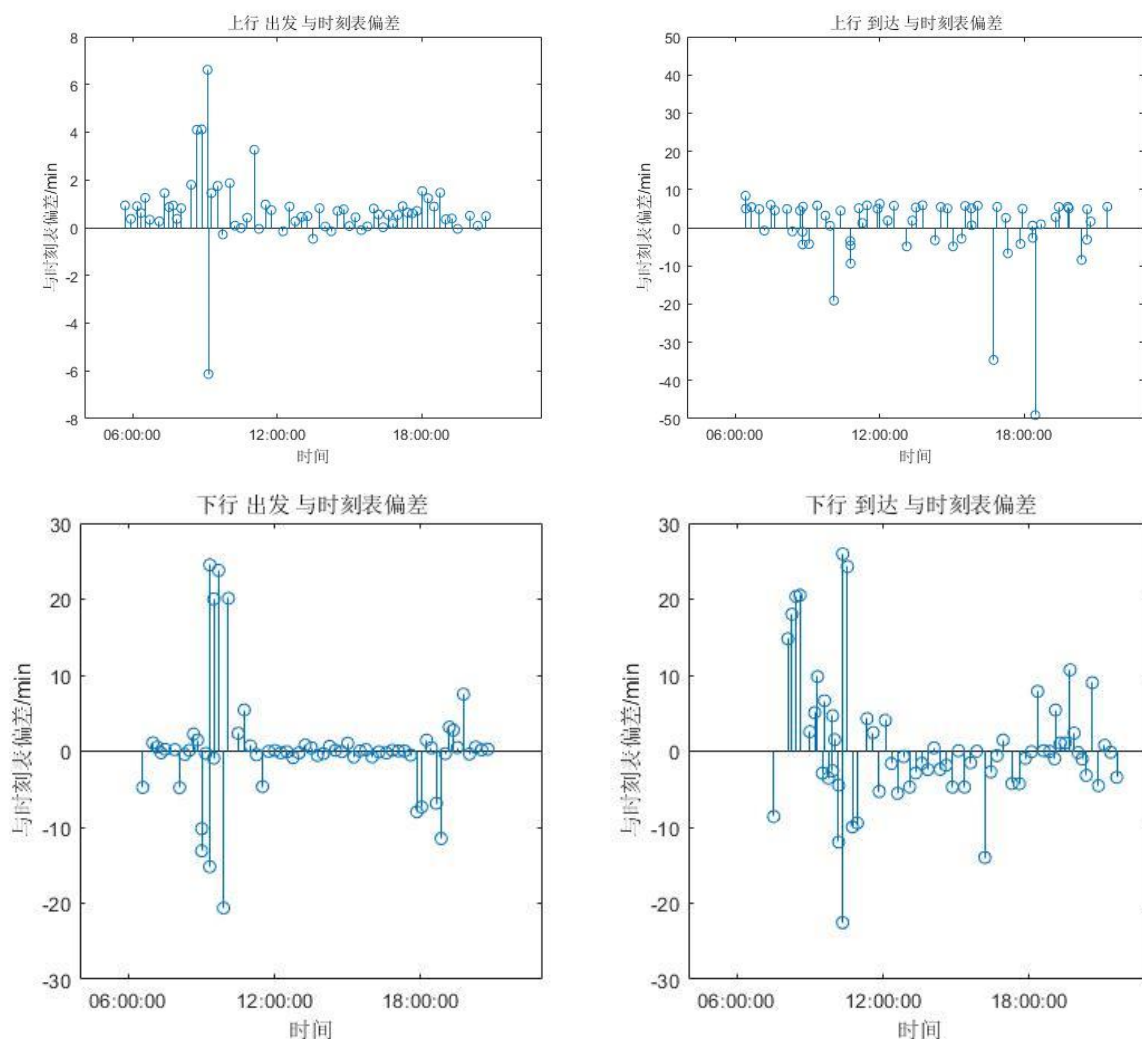


图 3.6 出发/到达与时刻表偏差



观察图 3.6，发现：

- ① 9 月 6 日上行出发与时刻表偏差均在 $\pm 8\text{min}$ 之内，准点率较高；且多为正偏差，也就是说出发时间晚于时刻表时间；其中，较大偏差主要出现在上午 9:00 附近，正好处于早高峰时期，考虑出现道路拥堵现象，导致多班次延误；
- ② 9 月 6 日上行到达与时刻表偏差均在 $\pm 30\text{min}$ 之内，准点率较出发与时刻表偏差低；且较大的偏差均为负偏差，即到达时间早于时刻表时间，主要出现在上午 10:00 左右以及下午 16:00 左右；这一现象出现的原因可能是处于空闲时段，因此需要对时刻表进行些许修改；
- ③ 9 月 6 日下行出发与时刻表偏差除上午 9:00~10:00 出发的班次以外，其余班次均在 $\pm 10\text{min}$ 之内，说明这些时段的公交车运营时刻表安排符合实际需求；而 9:00~10:00 出现的较大正延误可能是因为早高峰而造成的发车延误，上行车辆未能及时到达上行的终点站而耽误了下行的发车安排；同时，较多负延误出现在早晚高峰附近，可能是为了减缓堵车造成的影响而临时调度相对应的公交车提前 5~10min 出发，为合理调度；
- ④ 在 9 月 6 日下行的到达时刻表偏差中，繁忙时段正延误的次数与值均大于下行的出发时刻表偏差，这佐证了早、晚高峰时路线中的道路拥堵现象；而在空闲时段，如正午 14:00 左右，公交车则提前 5~10min 左右成功到达了终点站，这一点与上行的到达时刻表偏差分布规律相同。
- ⑤ 查阅 GPS 轨迹记录，发现 9 月 6 日下行轨迹中，有两趟出现空驶现象；

除此之外，有一点需要注意：在第一次绘制 9 月 6 日下行出发与到达时间差时，我发现最后三个轨迹的出发与到达与时刻表的偏差长达 12 个小时，如下表所示：

表 3.2 9 月 6 日下行轨迹异常数据

公交车	出发时间	到达时间	计发	应到
沪 A-03593D	20:46:54	21:59:40	9:00:00	9:55:00
沪 A-01722D	21:04:48	21:58:03	9:20:00	10:10:00
沪 A-01202D	21:29:05	21:57:27	9:30:00	10:20:00

观察发现，这个现象出现的时间可能有两个：

- (1)、由于临时调度安排或司机个人原因，将该趟公交临时从白班调动为夜班；
- (2)、时间数据记录出现错误。

在我看来，第二个原因的可能性更大，因此在本次下行出发/到达与时刻表偏差的绘制中，人工将记录到的时间数据提前 12 小时，并记录。



MATLAB 程序代码如下所示:

```
%寻找每条轨迹中：：真新车站出站时刻 寻找封浜汽车站到站时刻
%9月6日上行有75条轨迹，故各有75个点
%起始站出站时刻列入chu表中，终点站到站时刻列入dao表中
m=0;%储存起始点的出站点
n=0;%储存终点的进站点。
chu=zeros(75,8);
dao=zeros(75,8);
% 终点站进站点 上两个在终点站25m之外，该点与下一个在终点站25m之内
%%记录下该行数据；
for i=3:len_data-2
    s=data(i,8); %该个
    s1=data(i-1,8); %上一个
    s2=data(i-2,8); %上两个
    s3=data(i+1,8); %下一个
    if s1>25 && s2>25 && s<25 && s3<25
        n=n+1;
        dao(n,:)=data(i,:);
    end
end

%%匹配对应时刻表
%对timetable中的每个时间作差 找到最小的时间 即为对应时刻
%对应时刻存储在9列中
%timetable----第一列为车牌号，第45为出发到达的时刻表时间，67为出发早一晚二对应时间,89为到达早一晚二的对应时间
%出发
time_c=zeros(m,len_time);
for i =1:m
    for j= 1:len_time
        time_c(i,j)=abs((chu(i,1)-timetable_s(j,4))*1440);%min
    end
end
%到达
time_d=zeros(n,len_time);
for i = 1:n
    for j= 1:len_time
        time_d(i,j)=abs((chu(i,1)-timetable_s(j,5))*1440);%min
    end
end
time_c=time_c';
time_d=time_d';
%找到最小值与对应编号
[zhi_c,hao_c]=min(time_c);
[zhi_d,hao_d]=min(time_d);
%计算差并储存在chu dao之中的第10列
%出发
for i =1:m
    chu(i,10)=1440*(chu(i,1)-timetable_s(hao_c(1,i),4));
end
%到达
time_d=zeros(n,len_time);
for i = 1:n
    dao(i,10)=1440*(dao(i,1)-timetable_s(hao_d(1,i),5));
end
```



3.3.3. 发车间隔

对上行和下行方向，分别绘制嘉定 104 路公交在 9 月 6 日、7 日、8 日、9 日与 10 日起点站的发车间隔随时间变化，并分析其分布，观察不同天车辆的发车稳定性，估计其计划发车频率情况等。

MATLAB 代码如下所示，首先根据得到的时刻表数据计算标准发车间隔。

```

1  %%计算每日标准发车间隔
2  %读入数据
3  [num, text, raw]=xlsread('tt06s.xlsx');
4  tt06s=num(:,1);
5  len_06s=length(tt06s);
6  [num, text, raw]=xlsread('tt06x.xlsx');
7  tt06x=num(:,1);
8  len_06x=length(tt06x);
9  [num, text, raw]=xlsread('tt07s.xlsx');
10 tt07s=num(:,1);
11 len_07s=length(tt07s);
12 [num, text, raw]=xlsread('tt07x.xlsx');
13 tt07x=num(:,1);
14 len_07x=length(tt07x);
15 [num, text, raw]=xlsread('tt08s.xlsx');
16 tt08s=num(:,1);
17 len_08s=length(tt08s);
18 [num, text, raw]=xlsread('tt08x.xlsx');
19 tt08x=num(:,1);
20 len_08x=length(tt08x);
21 [num, text, raw]=xlsread('tt09s.xlsx');
22 tt09s=num(:,1);
23 len_09s=length(tt09s);
24 [num, text, raw]=xlsread('tt09x.xlsx');
25 tt09x=num(:,1);
26 len_09x=length(tt09x);
27 [num, text, raw]=xlsread('tt10s.xlsx');
28 tt10s=num(:,1);
29 len_10s=length(tt10s);
30 [num, text, raw]=xlsread('tt10x.xlsx');
31 tt10x=num(:,1);
32 len_10x=length(tt10x);
33 %计算间隔, 储存在ttdates/x的第二列
34 for i= 1:len_06s-1
35     tt06s(i,2)=(tt06s(i+1,1)-tt06s(i,1))*1440;%min
36 end
37 for i= 1:len_06x-1
38     tt06x(i,2)=(tt06x(i+1,1)-tt06x(i,1))*1440;%min
39 end
40 for i= 1:len_07s-1
41     tt07s(i,2)=(tt07s(i+1,1)-tt07s(i,1))*1440;%min
42 end
43 for i= 1:len_07x-1
44     tt07x(i,2)=(tt07x(i+1,1)-tt07x(i,1))*1440;%min
45 end
46 for i= 1:len_08s-1
47     tt08s(i,2)=(tt08s(i+1,1)-tt08s(i,1))*1440;%min
48 end
49 for i= 1:len_08x-1
50     tt08x(i,2)=(tt08x(i+1,1)-tt08x(i,1))*1440;%min
51 end
52 for i= 1:len_09s-1
53     tt09s(i,2)=(tt09s(i+1,1)-tt09s(i,1))*1440;%min
54 end
55 for i= 1:len_09x-1
56     tt09x(i,2)=(tt09x(i+1,1)-tt09x(i,1))*1440;%min
57 end
58 for i= 1:len_10s-1
59     tt10s(i,2)=(tt10s(i+1,1)-tt10s(i,1))*1440;%min
60 end
61 for i= 1:len_10x-1
62     tt10x(i,2)=(tt10x(i+1,1)-tt10x(i,1))*1440;%min
63 end

```



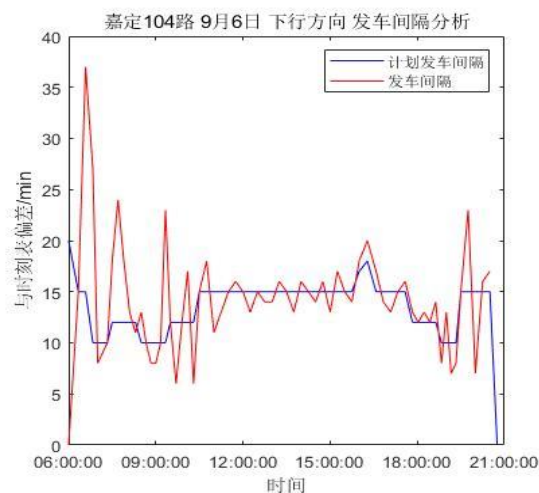
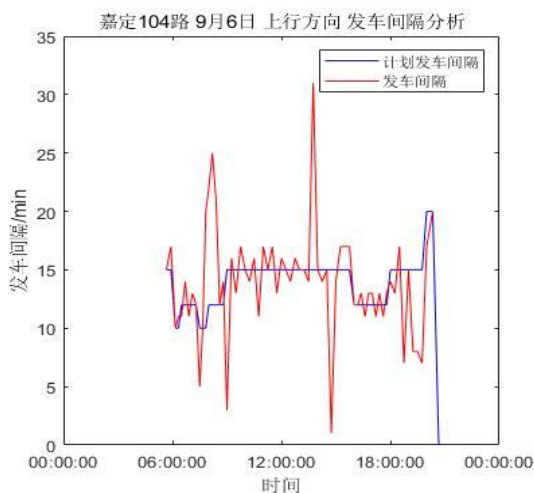

接着对 GPS 数据进行处理，得到实际发车间隔，这里以 9 月 6 日上行数据为例，并绘图。

```

64 %计算每天上行与下行方向的实际发车间隔
65 %读取数据，
66 %由于之前的处理中已经将起始站的出站行数据flag标为1，
67 %因此这里只需要对flag=1的数据进行计算
68 [num,text,row]=xlsread('06s.xlsx');
69 data=num;
70 len_data=length(data);
71 k_6s=0;%对进站点进行统计，列为新表格
72 data_06s=zeros(len_06s,2);
73 for i= 1:len_data
74     if flag(i,1)==1
75         k_6s=k_6s+1;
76         data_06s(i,1)=data(i,1);%将起始站出站点时间储存在表格中
77     end
78 end
79 %计算实际时间差
80 for i = 1:len_06s-1
81     data_06s(i,2)=(data_06s(i+1,1)-data_06s(i,1))*1440;%min
82 end
83 %绘图，规定间隔画蓝色，实际间隔画红色
84 subplot(1,2,1);
85 plot(tt06s(:,1),tt06s(:,2),'b');
86 hold on;
87 plot(data_06s(:,1),data_06s(:,2),'r');
88 xlabel('时间');
89 ylabel('发车间隔/min');
90 title('嘉定104路 9月6日 上行方向 发车间隔分析');
91 legend('计划发车间隔','发车间隔');
92 datetick('x',13);
93 subplot(1,2,2);
94 plot(tt06x(:,1),tt06x(:,2),'b');
95 hold on;
96 plot(data_06x(:,1),data_06x(:,2),'r');
97 xlabel('时间');
98 ylabel('与时刻表偏差/min');
99 title('嘉定104路 9月6日 下行方向 发车间隔分析');
100 legend('计划发车间隔','发车间隔');
101 datetick('x',13);
102

```

得到图 3.7，(1)、(2)、(3)、(4)、(5)分别为 9 月 6 日、7 日、8 日、9 日与 10 日的数



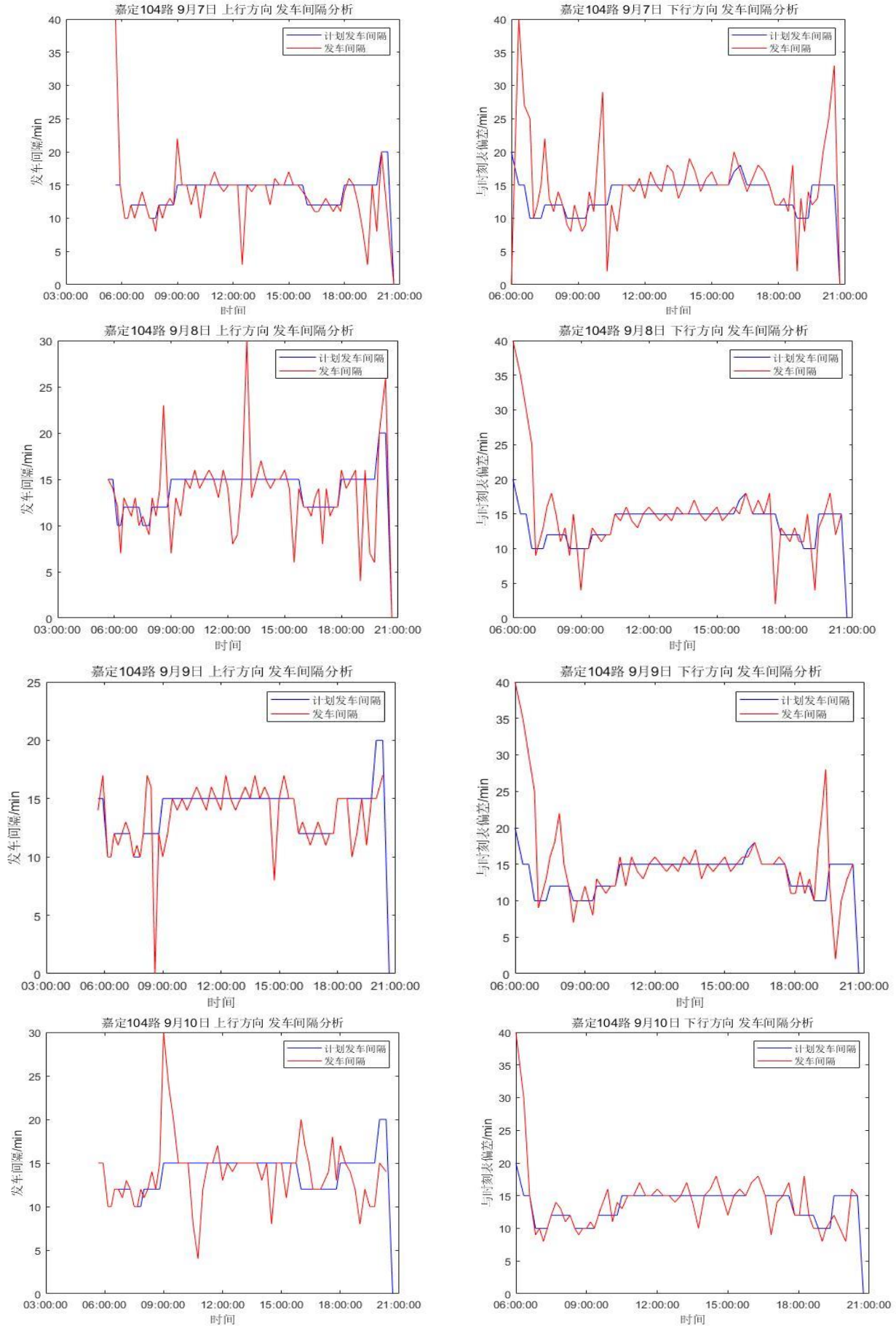


图 3.7 出发/到达与时刻表偏差

由于实际间隔偏差计算时最后一个数据点均设置为 0，因此在分析图像时应该忽略最后一个数据点。观察图 3.7，发现：

- ① 早、晚高峰期间的实际发车间隔与计划时刻表的发车间隔差距较大；
- ② 通常情况下，早高峰发车间隔比计划表大，晚高峰发车间隔比计划表小；
- ③ 9 月 6 日与 9 月 8 日的稳定性较小，其余三天的发车稳定性较大。

3.3.4. 运营时间分析

由区分上下行，分别统计各班次车辆运营时间，并估计运营时间的 85%分位数。

其中，运营时间=车辆到达终点站时间-车辆离开起点站时间

绘制全天运营时间变化情况，并分析时刻表计划运营时间的合理性。

统计每天各班次运营时间，对于不同天的相同班次，计算运营时间的 85%分位数。

分位数的计算过程：先对五个数值排序，然后利用 MATLAB 中的 `prctile` 函数进行 85%分位数的求解。

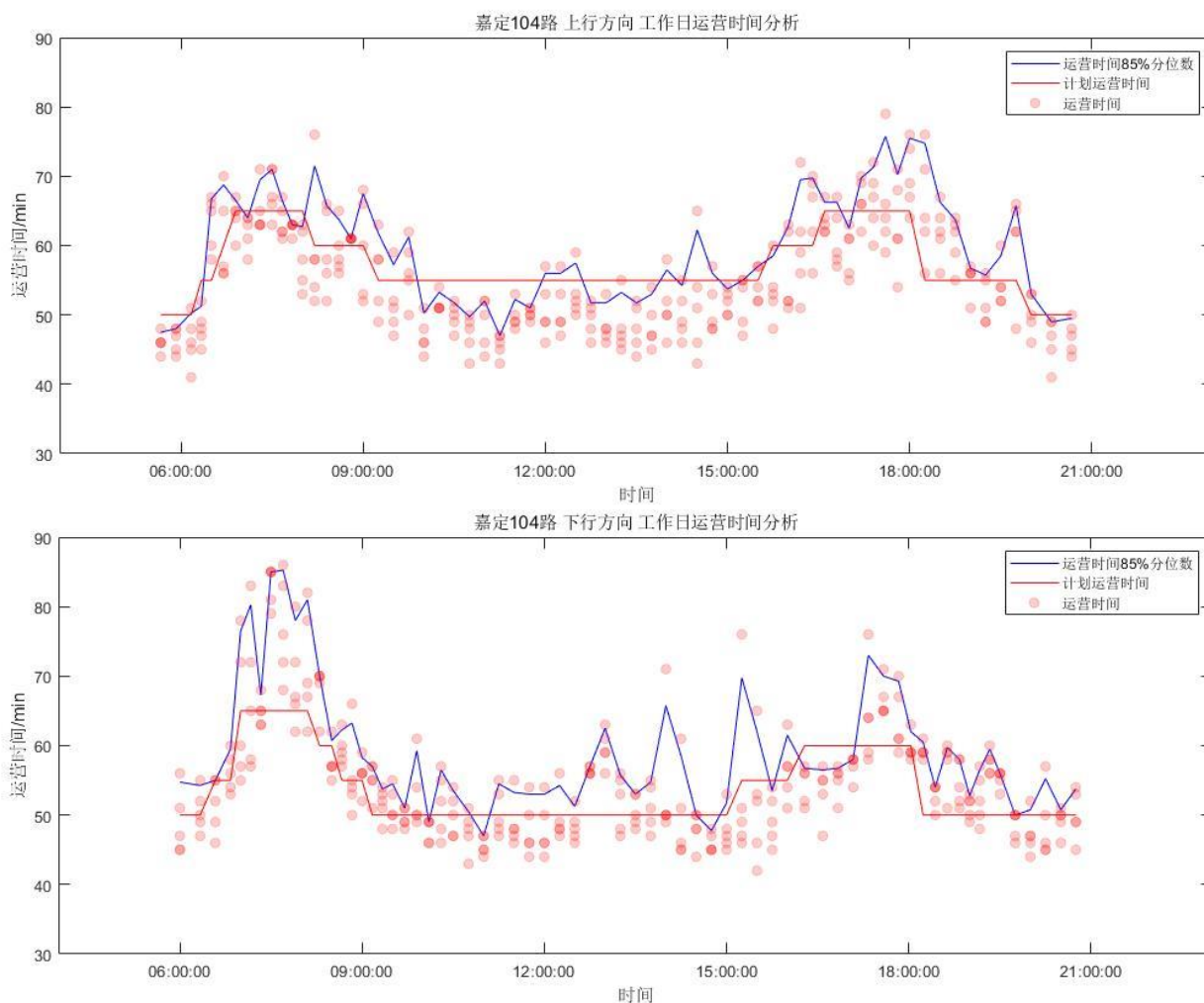


图 3.8 运营时间分析图

MATLAB 代码如下所示：首先根据得到的时刻表数据计算标准运营时间。



```
1 %读入标准运营时间
2 [num,text,row]=xlsread('tt06s.xlsx');
3 tt06s=num(:,1:2);
4 len_06s=length(tt06s);
5 [num,text,row]=xlsread('tt06x.xlsx');
6 tt06x=num(:,1:2);
7 len_06x=length(tt06x);
8 [num,text,row]=xlsread('tt07s.xlsx');
9 tt07s=num(:,1:2);
10 len_07s=length(tt07s);
11 [num,text,row]=xlsread('tt07x.xlsx');
12 tt07x=num(:,1:2);
13 len_07x=length(tt07x);
14 [num,text,row]=xlsread('tt08s.xlsx');
15 tt08s=num(:,1:2);
16 len_08s=length(tt08s);
17 [num,text,row]=xlsread('tt08x.xlsx');
18 tt08x=num(:,1:2);
19 len_08x=length(tt08x);
20 [num,text,row]=xlsread('tt09s.xlsx');
21 tt09s=num(:,1:2);
22 len_09s=length(tt09s);
23 [num,text,row]=xlsread('tt09x.xlsx');
24 tt09x=num(:,1:2);
25 len_09x=length(tt09x);
26 [num,text,row]=xlsread('tt10s.xlsx');
27 tt10s=num(:,1:2);
28 len_10s=length(tt10s);
29 [num,text,row]=xlsread('tt10x.xlsx');
30 tt10x=num(:,1:2);
31 len_10x=length(tt10x);
32 %计算标准运营时间 3列为运营时间 min
33 for i= 1:len_06s
34     tt06s(i,3)=(tt06s(i,2)-tt06s(i,1))*1440;%min
35 end
36 for i= 1:len_06x
37     tt06x(i,3)=(tt06x(i,2)-tt06x(i,1))*1440;%min
38 end
39 for i= 1:len_07s
40     tt07s(i,3)=(tt07s(i,2)-tt07s(i,1))*1440;%min
41 end
42 for i= 1:len_07x
43     tt07x(i,3)=(tt07x(i,2)-tt07x(i,1))*1440;%min
44 end
45 for i= 1:len_08s
46     tt08s(i,3)=(tt08s(i,2)-tt08s(i,1))*1440;%min
47 end
48 for i= 1:len_08x
49     tt08x(i,3)=(tt08x(i,2)-tt08x(i,1))*1440;%min
50 end
51 for i= 1:len_09s
52     tt09s(i,3)=(tt09s(i,2)-tt09s(i,1))*1440;%min
53 end
54 for i= 1:len_09x
55     tt09x(i,3)=(tt09x(i,2)-tt09x(i,1))*1440;%min
56 end
57 for i= 1:len_10s
58     tt10s(i,3)=(tt10s(i,2)-tt10s(i,1))*1440;%min
59 end
60 for i= 1:len_10x
61     tt10x(i,3)=(tt10x(i,2)-tt10x(i,1))*1440;%min
62 end
```



然后计算各天不同班次的运营时间。

```

64 %计算各天不同班次运营时间
65 %由于之前的处理中已经将起始站的出站行数据flag标为1，终点站的flag标为2
66 %因此直接将其相减
67 %以9月6日上行为例
68 %上行
69 [num,text,row]=xlsread('0906s.xlsx');
70 data=num(:,:);
71 len_data=length(data);
72 k=0;%k用来对轨迹计数
73 for i = 1:len_data
74     if flag(i,1)==1
75         k=k+1;
76         time(k,1)=data(i,5);
77     else if flag(i,1)==2
78         time(k,2)=data(i,5);
79     end
80 end
81 end
82 time(:,3)=(time(:,2)-time(:,1))*1440;%min 运营时间
83 %记录在time_zong中，1~5分别为6~10号的运营时间
84 time_zongs(:,1)=time(:,3);

```

最后对于不同天的相同班次，计算运营时间的 85%分位数。

```

for i = 1:len_s
    a=s(i,4:8);
    s(i,9)=prctile(a,85);
end

```

绘图，得到图 3.8。标准红线，分位数蓝线，粉色透明点为所有运营时间，以上行为例。

```

%shang
figure;
plot(s(:,1),s(:,9),'b');
hold on;
plot(s(:,1),s(:,3),'r');
hold on;
scatter(s(:,1),s(:,4),'MarkerFaceColor','r','MarkerEdgeColor','r',...
    'MarkerFaceAlpha',.2,'MarkerEdgeAlpha',.2);
hold on;
scatter(s(:,1),s(:,5),'MarkerFaceColor','r','MarkerEdgeColor','r',...
    'MarkerFaceAlpha',.2,'MarkerEdgeAlpha',.2);
hold on;
scatter(s(:,1),s(:,6),'MarkerFaceColor','r','MarkerEdgeColor','r',...
    'MarkerFaceAlpha',.2,'MarkerEdgeAlpha',.2);
hold on;
scatter(s(:,1),s(:,7),'MarkerFaceColor','r','MarkerEdgeColor','r',...
    'MarkerFaceAlpha',.2,'MarkerEdgeAlpha',.2);
hold on;
scatter(s(:,1),s(:,8),'MarkerFaceColor','r','MarkerEdgeColor','r',...
    'MarkerFaceAlpha',.2,'MarkerEdgeAlpha',.2);

set(gca,'XLim',[1/24*5 1/24*23]);
set(gca,'YLim',[30 90]);
xlabel('时间');
ylabel('运营时间/min');
title('嘉定104路 上行方向 工作日运营时间分析');
legend('运营时间85%分位数','计划运营时间','运营时间');
datetick('x',13);

```


观察图 3.8，发现：

- ① 早、晚高峰时期嘉定 104 路公交的运营时间较长，这一点适用于上行方向与下行方向，且此段时间上下行方向的实际运用时间往往长于时刻表规定的运营时间，这说明早晚高峰期的道路拥堵现象比时刻表制定时预想的更严重一些，而时刻表需要根据数据进行进一步调整；
- ② 早、晚高峰时期嘉定 104 路公交的下行方向运营时间 85%分位数比上行方向更波动；
- ③ 非高峰时段嘉定 104 路公交在上行方向的运营时间往往低于时刻表时间，而下行方向的运营时间有些高于规定时间，有些低于规定时间；且下行方向的运营时间 85%分位数比上行方向更波动。

3.3.5. Layover 时间分析

由于在 3.3.2 中已经分析过 9 月 6 日出发/到达与时刻表偏差，故在这里同样选择 9 月 6 日的数据进行处理，免除一部分工作，从而绘制 9 月 6 日 Layover 时间变化情况，并进行分析。其中，主站指真新公交站，副站指封浜公交站，Layover 时间的定义如下：

Layover 时间=车辆离开站点时间-车辆到达站点时间（即司机在主副站的休息时间）

在绘图时为了更明显的表示每个站点的休息时间，选用气泡图进行绘制得到图 3.10。绘图代码见本节后，横坐标为时间，纵坐标为 Layover 时间。

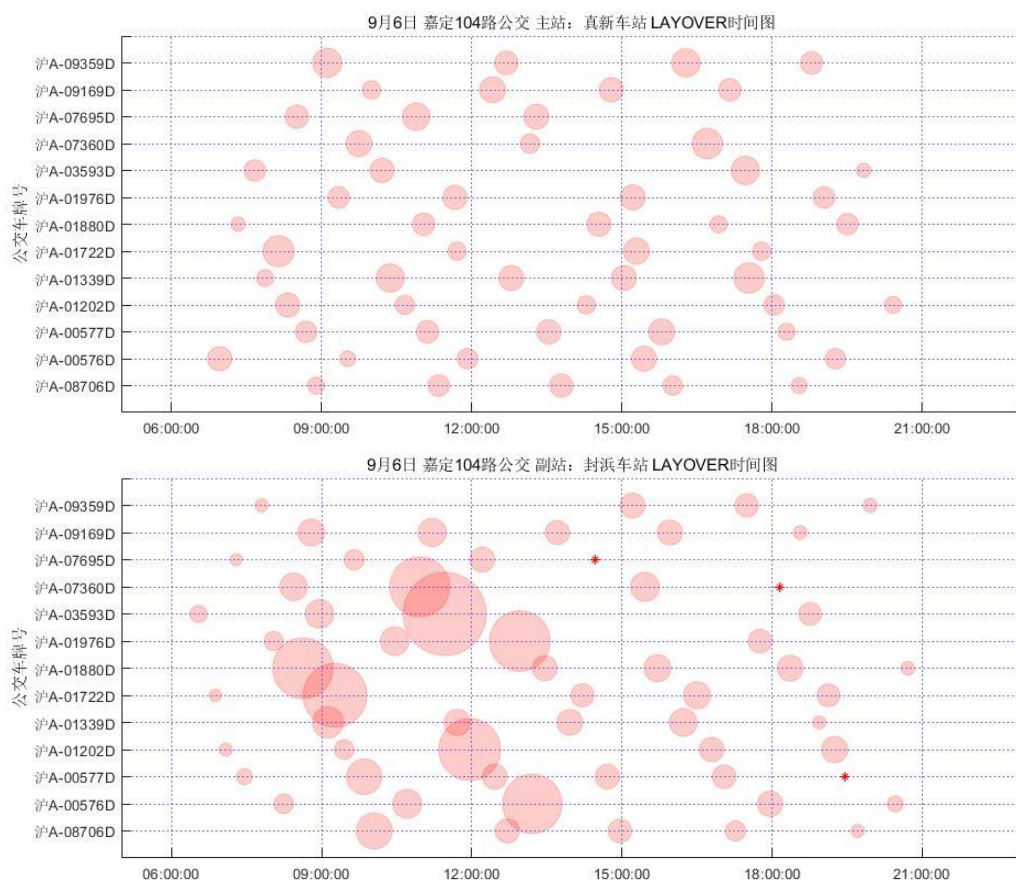


图 3.9 Layover 时间分析图，（1）为主站，（2）为副站

进一步计算每辆车平均休息时间并比较。

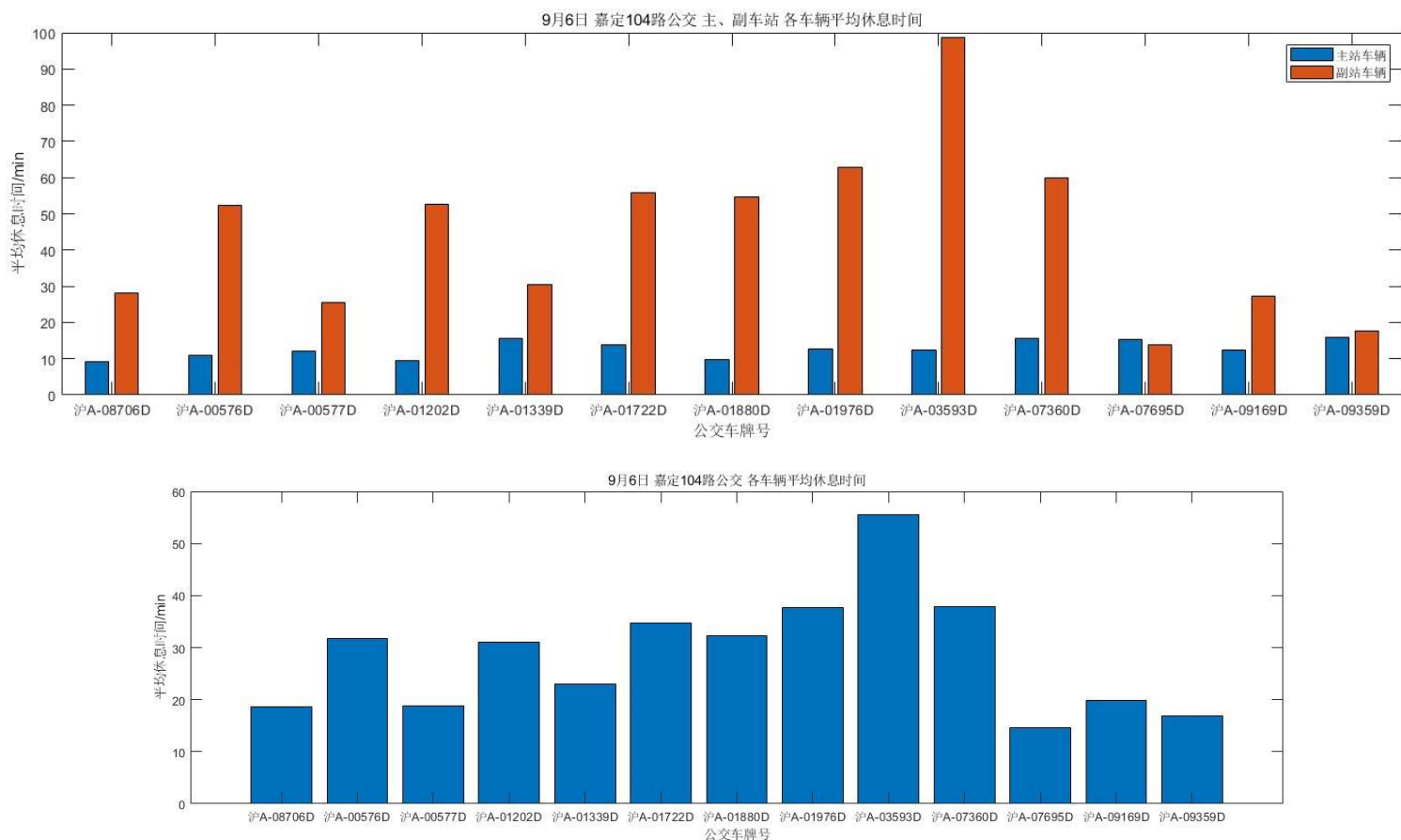


图 3.10 主、副站各公交车平均休息时间

观察图 3.9 与图 3.10，发现：

- ① 副站：封浜车站 的休息时间较 主站：真新车站 长，且副站有部分班次的休息时间为 0；
- ② 对于副站而言，其主要休息时间主要集中在上午 11:00 前后，这可能是因为换班，也可能是因为该时段对嘉定 104 路公交的用户需求不大；
- ③ 绝大多数的车辆的副站休息时长大于 30min，而几乎所有的主站平均休息时长在 10min 左右，主副站的平均休息时长在 30min 左右；
- ④ 所有车辆的副站平均休息时长远大于主站平均休息时长，这说明司机主要在副站进行休息。



MATLAB 代码如下所示:

```
1 %读入数据
2 [num,text,row]=xlsread('0907s.xlsx');
3 datas=num;
4 len_datas=length(datas);
5 [num,text,row]=xlsread('0907x.xlsx');
6 datax=num;
7 len_datax=length(datax);
8 %1为车辆编号 2kong 3时间 45经纬度 6flag
9 %记录下每辆车出发与到达主站的时间 出发与到达副站的时间
10 %上行方向的主站flag为1 副站flag为2
11 %下行方向的主站flag为2 副站flag为1
12 ks=0;
13 kx=0;
14 %zhu fu 1为车辆编号 2kong 3时间 45经纬度 6flag
15 zhu=zeros(150,9);
16 fu=zeros(150,9);
17 for i= 1:len_datas
18     if datas(i,6)==1
19         ks=ks+1;
20         zhu(ks,:)=datas(i,:);
21     else if datas(i,6)==2
22         kx=kx+1;
23         fu(kx,:)=datas(i,:);
24     end
25 end
26 end
27 for i= 1:len_datax
28     if datax(i,6)==2
29         ks=ks+1;
30         zhu(ks,:)=datas(i,:);
31     else if datax(i,6)==1
32         kx=kx+1;
33         fu(kx,:)=datas(i,:);
34     end
35 end
36 end
37 %使zhu fu根据车辆编号排序, 共13辆车
38 %车辆编号与车牌号对应关系和表3.1相同
39 sortrows(zhu,1);
40 sortrows(fu,1);
41 %两个数据为一组计算时间差 (min)
42 %由于是要计算司机在主副站的休息时间, 因此每辆公交车从第二个数据开始计算
43 %第七列储存差值 1位车牌 3为时间
44 m=length(zhu);
45 n=length(fu);
46 for i =2:m
47     if zhu(i-1,1)==zhu(i,1)
48         zhu(i,7)=(zhu(i,3)-zhu(i-1,3))*1440;%min
49     end
50 end
51 for i =2:n
52     if fu(i-1,1)==fu(i,1)
53         fu(i,7)=(fu(i,3)-fu(i-1,3))*1440;%min
54     end
55 end
```

4. 任务 3：公交路段运行情况分析

4.1. 路段运营时间及延误

以相邻两个车站定义路段，分别取早晚高峰时段，计算线路上下行各个路段的路段运营时间和延误，鉴别对公交运行影响最大的路段。

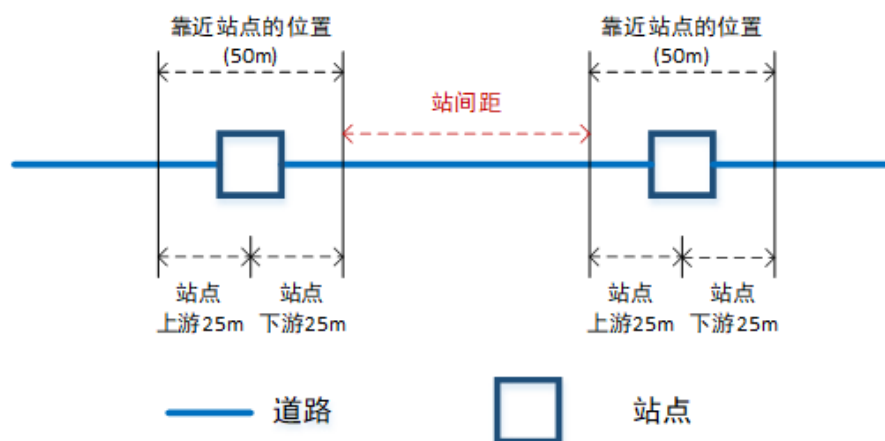


图 4.1 路段定义方式示意图

延误=路段运营时间-自由流运营时间，其中自由流运营时间以 15%分位运营时间表示。根据延误公式的定义，一些距离较长的路段往往有较大的延误，因此进一步计算延误占比，衡量路段的拥堵情况。

延误占比=路段延误/路段运营时间

取 9 月 10 日，绘制该天上下行早晚高峰路段延误情况，MATLAB 代码如下：

```
%%绘图
figure;
%sz
x=data(:,1);
y1=data(:,2);
y2=data(:,3);
y3=data(:,4);

subplot(1,2,1);

yyaxis left;%激活左边的轴
c=bar(y3,'FaceColor',...
[0.941176470588235,0.972549019607843,1]);
set(gca,'YLim',[0 1]);
set(c,'edgecolor','none');%柱状图无边框
ylabel('延误占比');

yyaxis right;%激活右边的轴
plot(x,y1,'color',...
[0.627450980392157,0.321568627450980,0.176470588235294]);
hold on ;
plot(x,y2,'color',...
[0.415686274509804,0.352941176470588,0.803921568627451]);
ylabel('时间/min');

xlabel('路段编号');
title('9月10日 上行 早高峰 路段行程时间及延误');
legend('平均延误','平均运营时间','平均延误占比');
set(gca,'XLim',[0 23]);
set(gca,'XTick',[0:1:22]);
%sw
x=data(:,1);
y1=data(:,5);
y2=data(:,6);
y3=data(:,7);

subplot(1,2,2);
```



```
yyaxis left;%激活左边的轴
c=bar(y3,'FaceColor',...
      [0.941176470588235,0.972549019607843,1]);
set(gca,'YLim',[0 1]);
set(c,'edgecolor','none');%柱状图无边框
ylabel('延误占比');

yyaxis right;%激活右边的轴
plot(x,y1,'color',...
      [0.627450980392157,0.321568627450980,0.176470588235294]);
hold on ;
plot(x,y2,'color',...
      [0.415686274509804,0.352941176470588,0.803921568627451]);
ylabel('时间/min');

xlabel('路段编号');
title('9月10日 上行 晚高峰 路段行程时间及延误');
legend('平均延误','平均运营时间','平均延误占比');
set(gca,'XLim',[0 23]);
set(gca,'XTick',[0:1:22]);

%%%%%%%%下行
figure;
%sz
x=data(:,1);
y1=data(:,8);
y2=data(:,9);
y3=data(:,10);
subplot(1,2,1);
yyaxis left;%激活左边的轴
c=bar(y3,'FaceColor',...
      [0.941176470588235,0.972549019607843,1]);
set(gca,'YLim',[0 1]);
set(c,'edgecolor','none');%柱状图无边框
ylabel('延误占比');

yyaxis right;%激活右边的轴
plot(x,y1,'color',...
      [0.627450980392157,0.321568627450980,0.176470588235294]);
hold on ;
plot(x,y2,'color',...
      [0.415686274509804,0.352941176470588,0.803921568627451]);
ylabel('时间/min');

xlabel('路段编号');
title('9月10日 下行 早高峰 路段行程时间及延误');
legend('平均延误','平均运营时间','平均延误占比');
set(gca,'XLim',[0 23]);
set(gca,'XTick',[0:1:22]);
%sw
x=data(:,1);
y1=data(:,12);
y2=data(:,13);
y3=data(:,14);

subplot(1,2,2);

yyaxis left;%激活左边的轴
c=bar(y3,'FaceColor',...
      [0.941176470588235,0.972549019607843,1]);
set(gca,'YLim',[0 1]);
set(c,'edgecolor','none');%柱状图无边框
ylabel('延误占比');

yyaxis right;%激活右边的轴
plot(x,y1,'color',...
      [0.627450980392157,0.321568627450980,0.176470588235294]);
hold on ;
plot(x,y2,'color',...
```



[0.415686274509804, 0.352941176470588, 0.803921568627451]];

ylabel('时间/min');

xlabel('路段编号');

title('9月10日 下行 晚高峰 路段行程时间及延误');

legend('平均延误', '平均运营时间', '平均延误占比');

set(gca, 'XLim', [0 23]);

set(gca, 'XTick', [0:1:22]);

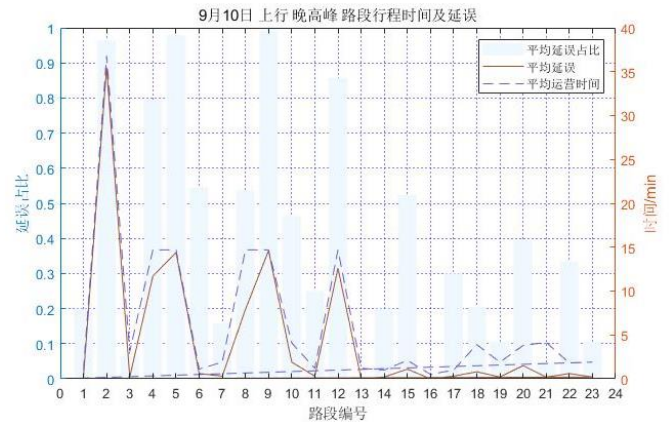
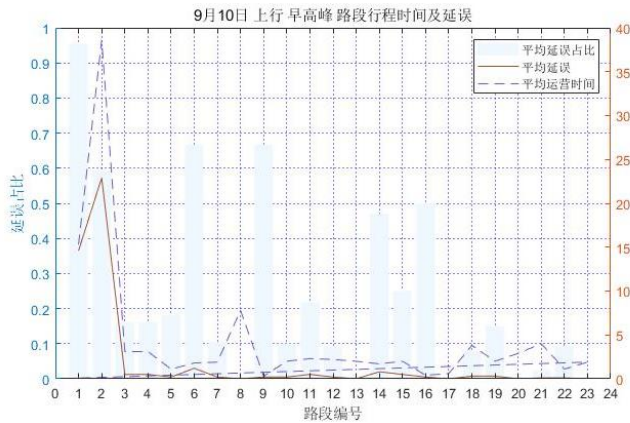


图 4.2 各路段平均行程时间及延误

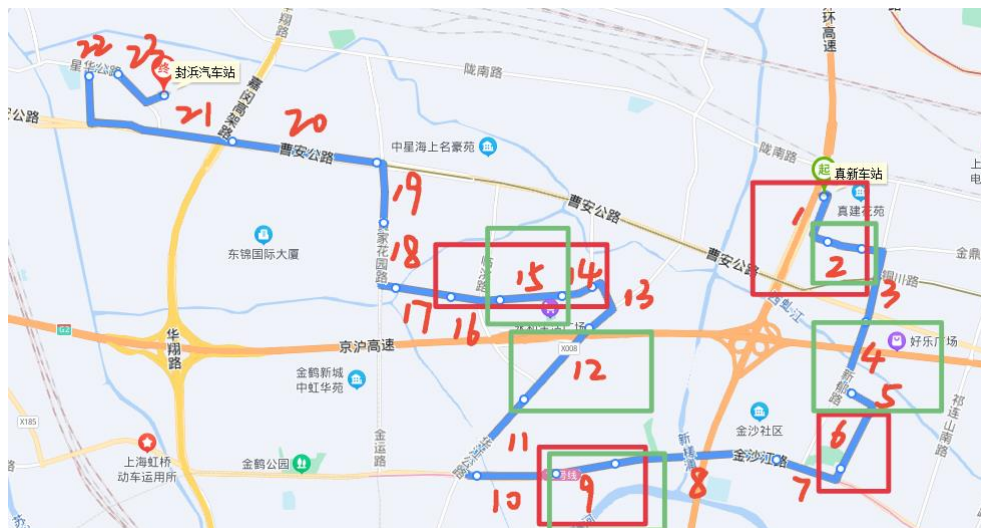


图 4.3 嘉定 104 路公交上行路段

绘图，得到图 4.2，观察上行的早晚高峰延误统计图，发现：

- ① 上行早高峰较为拥堵的路段编号为 1、2、6、9、14、16；
- ② 上行晚高峰较为拥堵的路段编号为 2、4、5、9、12、15；

结合图 4.3 嘉定 104 路公交上行路段进行分析（红色框选为早高峰拥堵路段、绿色框选为晚高峰拥堵路段）：

- ① 早高峰时出现的道路拥堵的 1/2 路段附近有大量居民区、有上海市江桥批发市场，因此存在大量的居民需要从家前往工作地点、商贩前往批发市场进货等原因，导致该路段在早高峰时出现拥堵现象；
- ② 早高峰时出现的道路拥堵的 6 路段附近有大量居民区、金沙绿地公园、以及上海市市庄中学，因此存在居民通勤、学生上学等需求，导致早高峰时拥堵的产生；
- ③ 早高峰时出现的道路拥堵的 9 路段与地铁 13 号线重合，且金沙江西路在路段 9 的



终点处，该路段附近存在江桥工业园区、创意园区以及金沙商务中心，是通勤的终点，因此容易产生道路拥堵；

- ④ 早高峰时出现的道路拥堵的 14 与 16 路段附近依旧是大量的居民区与菜市场、水族批发市场等，居民的通勤需求、商贩等原因共同导致了早高峰期间的拥堵；
- ⑤ 晚高峰路段 2/4/5/9/15 产生拥堵的原因依旧是居民的通勤需求以及部分的商贩进货、卖货需求；而晚高峰路段 12 出现拥堵则可能是因为上海申窑艺术中心及其附近的公司员工下班；

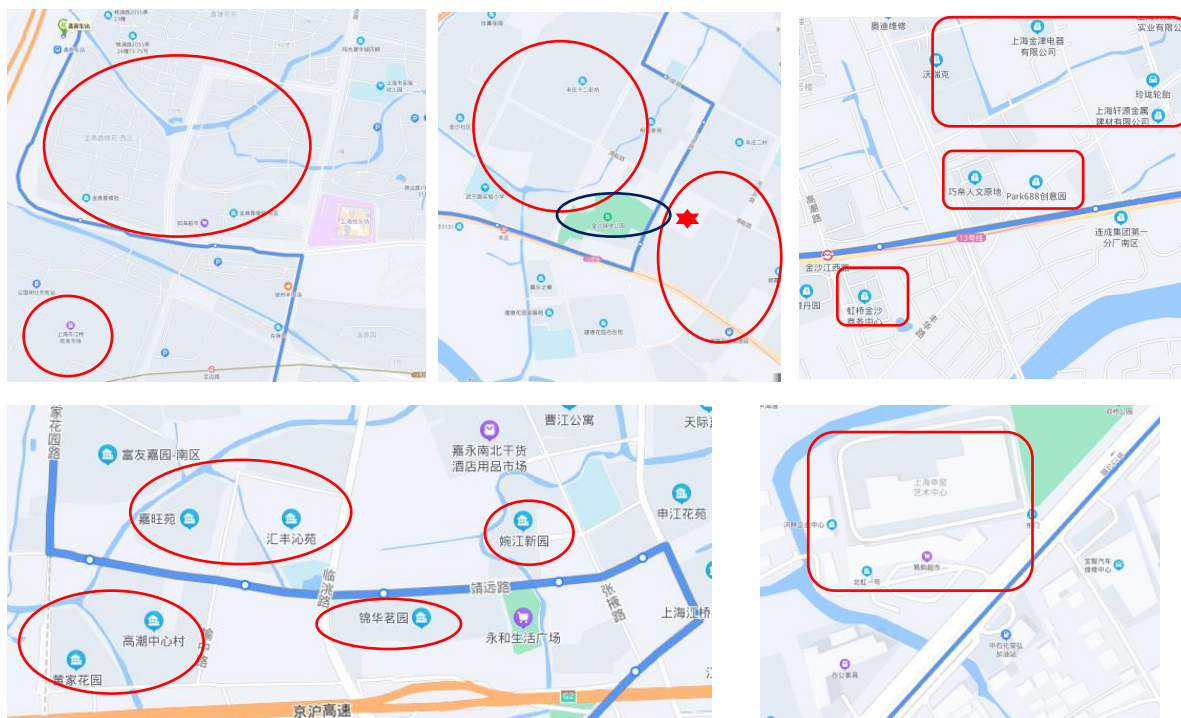


图 4.4 上行路段 1/2/6/9/14/16/12

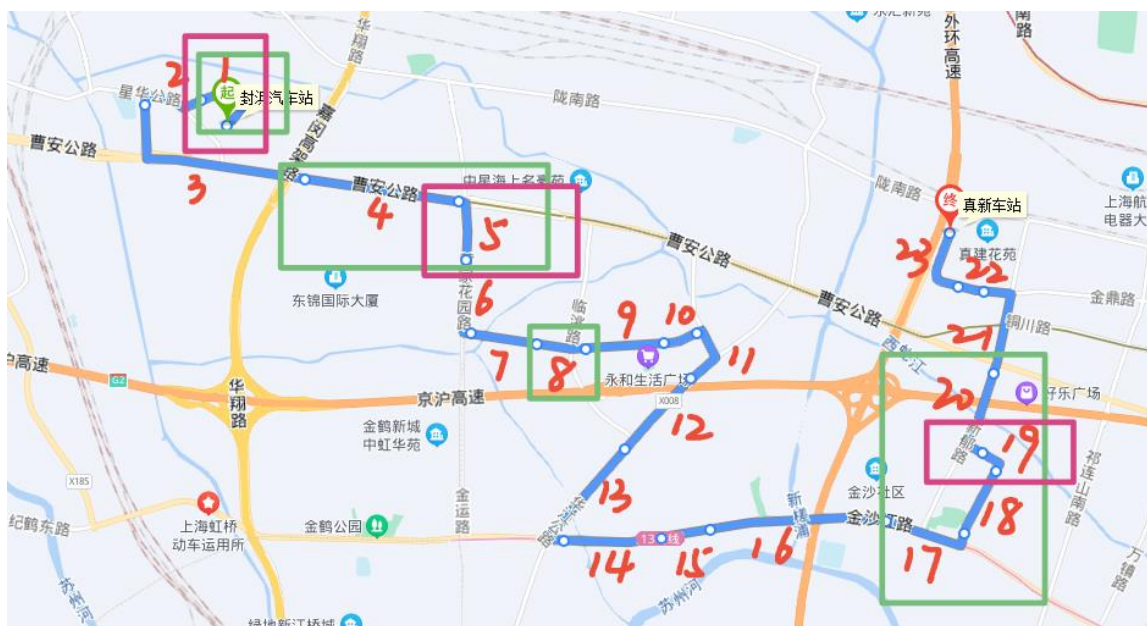


图 4.5 嘉定 104 路公交下行路段

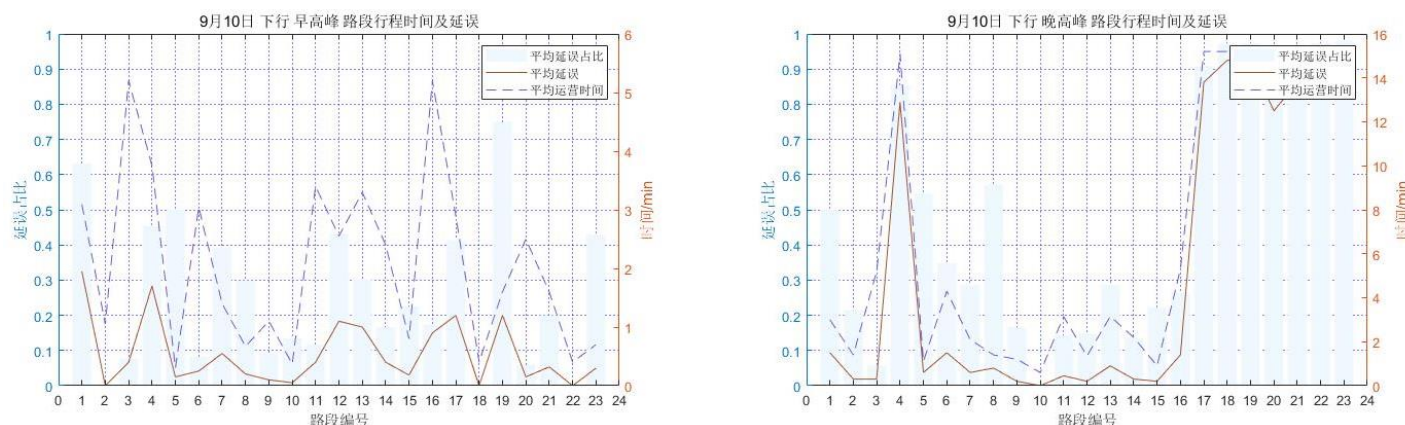


图 4.6 各路段平均行程时间及延误

同理，得到图 4.6，观察下行的早晚高峰延误统计图，发现：

- ① 下行早高峰较为拥堵的路段为 1、5、19；
- ② 下行晚高峰较为拥堵的路段为 1、4、5、8、17、18、19、20；

结合图 4.5 嘉定 104 路公交下行路段进行分析（红色框选为早高峰拥堵路段、绿色框选为晚高峰拥堵路段）：

- ① 在下行统计中路段 17/18/19/20 与 8 附近有许多居民区，具体分析见上行路段，这些居民区导致产生许多出行需求，使得道路容易产生拥堵现象；
- ② 下行路段在黄家花园路以及曹安公路附近产生了相对较大（与下行其他路段相比）的延误，但事实上其值大小与上行对应值的大小相差不大；同时，尽管其延误时间较长，但其延误占比并没有成为最突出的；
- ③ 下行路段在封浜汽车站附近产生了相对较大的延误，原因可能是周围的居民出勤以及学生上学；
- ④ 总体来说，下行的各路段平均延误早高峰最大值在 5min 附近、晚高峰最大值在 15min 附近，远小于上行的同期延误。



图 4.7 下行路段 1/2 附近



4.2. 停靠时间分析

同样取 9 月 10 日，计算所有上行和下行运行轨迹中各个站点的平均停靠时间，记录在表格中，得到图 4.8。



图 4.8 9 月 10 日 104 路公交车上行与下行站点的平均停靠时间

观察图 4.8，对于一些停站时间较长的站点，在地图上标记，并结合站点周围 POI 简要推测原因，发现：



图 4.9 嘉定 104 路上/下行站点分析

- ① 上行路线上站点的平均停靠时间均在 60s 之内，其中 7 号站（丰庄路金沙江路）、14 号站（靖远路华江支路）的平均停靠时间超过 40s；结合地图观察，7 号站与 14 号站附近均有大量居民小区，均拥有一个学校，而且 7 号站附近拥有一个公园，14 号站附近拥有一个生活广场，人流量较大；
- ② 下行路线上站点的平均停靠时间最大值为 100 附近，中间站的波动比上行轨迹上小，但是 3 号站、6 号站以及 24 号站的平均停靠时间均超过了 60s，7 号站、9 号站与 22 号站的平均停靠时间超过了 30s，其余站点的平均停靠时间均在 30s 以下；结合地图观察，这些站点附近均有大量居民区，同时 7/22 号站附近还存在学校，3 号站附近存在购物中心，因此人流量较大，汽车停靠时间较长。

平均停靠时间最长的站点是下行中的 3 号站，由于，因此绘制车辆在此站点——金鼎路定边路停靠时间随着时间的变化。

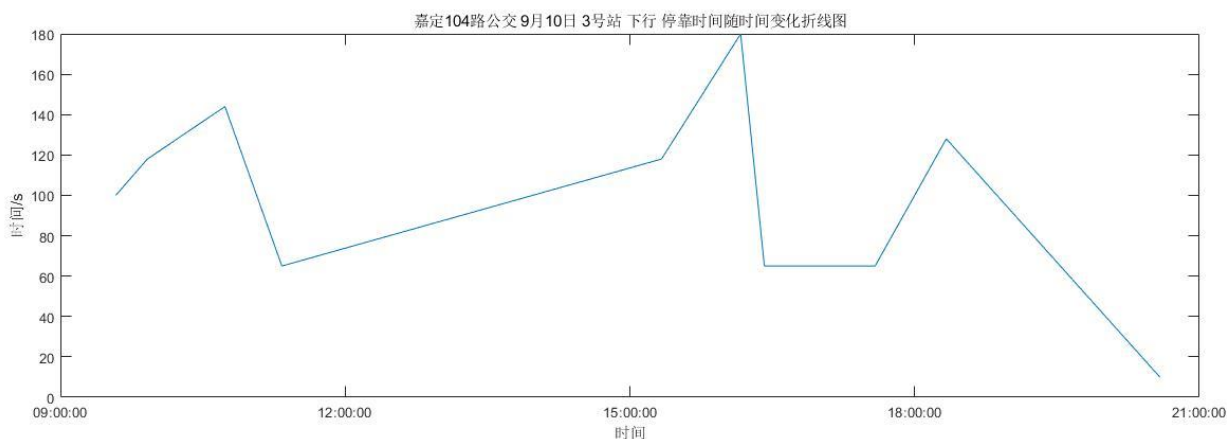


图 4.10 3 号站停靠时间随时间变化图

观察发现，在早、晚高峰期间，3 号站的停靠时间大于 100s，同时下午 16:00 左右达到停靠时间的最高峰，其原因可能是居民通勤导致的公交乘客客流量增大。



计算站点平均停靠时间的 MATLAB 代码如下：

```
[num, text, raw]=xlsread('tingkao.xlsx');
data1=num;
len=length(data1);

%3.1中已经计算了站点之间的行程时间,其中将每个站点的进站出站属性标记在12列,标记在了data的13列中

%计算提取轨迹中的进出站点,以上行为例
jin=zeros(len,12);
chu=zeros(len,12);
k_jin=0;
k_chu=0;
a=zeros(24,80);
for i =1:len
    if data(i,12)==0 %是进站点
        k_jin=k_jin+1;
        jin(k_jin,:)=data(i,12);
    else if data(i,12)==1 %是出站点
        k_chu=k_chu+1;
        chu(k_chu,:)=data(i,12);
    end
end

%计算每个站点的停靠时间
tingkao=zeros(k_jin,2);
tingkao(:,1)=chu(:,1)-jin(:,1);%1列储存时间 2列储存站号
tingkao(:,2)=chu(:,3);
len_tk=length(tingkao);
%a1列用来储存每个站点的停靠次数 后面每列储存一个停靠时间
for i=1:len_tk
    m=tingkao(i,2)%得到是哪个站点;
    a(m,1)=a(m,1)+1;
    a(m,a(m,1)+1)=tingkao(i,1);
end
%计算10号上行每个站点平均停靠时间
for i=1:24
    data(i,1)=i;
    data(i,2)=average(a(i,2):a(i,1));
end
%储存在data中 1站点 2上行平均停靠时间
%下行同理, 3列下行平均停靠时间
%10号分别有78与77个上下行轨迹

%绘图
figure;
%s
plot(data(:,1),data(:,2),'color',...
      [0.415686274509804,0.352941176470588,0.803921568627451]);
hold on;
%x
plot(data(:,1),data(:,3),'color',...
      [0.627450980392157,0.321568627450980,0.176470588235294]);
grid on;set(gca,'GridLineStyle',':', 'GridColor','b','GridAlpha',1);%添加网格虚线
ylabel('时间/s');
xlabel('站点');
title('嘉定104路公交 9月10日 上/下行 站点平均停靠时间');
legend('上行站点平均停靠时间','下行站点平均停靠时间');
set(gca,'XLim',[0 25]);
set(gca,'XTick',[0:1:25]);
set(gca,'YLim',[0 110]);
```



4.3. 站点车间时距分析

取 9 月 10 日公交车 GPS 轨迹数据，分别对上行和下行方向，计算站点车间时距(h)，绘制不同站点车间时距的 boxplot 图，并分析车间时距与行驶距离的关系。MATLAB 代码如下：

```
1 %以上行为例
2 [num,text,row]=xlsread('0910s.xlsx');
3 data=num;
4 len=length(data);
5
6
7 %3.1中已经计算了站点之间的行程时间,其中将每个站点的进站出站属性标记在12列 值=0时为进站点 2列是停靠的站点编号
8 b=zeros(24,1);%b记录站点经过次数
9 a=zeros(24,75);%a记录每个站点每趟公交经过的时刻
10 %统计10号每个站点的公交车进站时刻
11 for i =1:len
12     if data(i,12)==0 %是进站点
13         m=data(i,2);%记录站点编号;
14         b(m,1)=b(m,1)+1;%某站点的第b(m,1)次时刻
15         a(m,b(m,1))=data(i,1);%将时间记录在对应编号后;
16     end
17 end
18 %某站点的车间时距 相邻公交的进站时刻 相减 得到
19 %对a中每一行进行排序,方便计算
20 sort(a);
21 %用time_gap计算车间时距, time_gap的有效列数为a的列数-1
22 time_gap=zeros(24,74);
23 for i=1:74
24     time_gap(:,i)=a(:,i+1)-a(:,i);
25 end
26 %对time_gap转置 方便绘图boxplot
27 time_gap=time_gap';
28 x1=time_gap(1,1:b(1,1)-1);
29 x2=time_gap(2,1:b(2,1)-1);
30 x3=time_gap(3,1:b(3,1)-1);
31 x4=time_gap(4,1:b(4,1)-1);
32 x5=time_gap(5,1:b(5,1)-1);
33 x6=time_gap(6,1:b(6,1)-1);
34 x7=time_gap(7,1:b(7,1)-1);
35 x8=time_gap(8,1:b(8,1)-1);
36 x9=time_gap(9,1:b(9,1)-1);
37 x10=time_gap(10,1:b(10,1)-1);
38 x12=time_gap(12,1:b(12,1)-1);
39 x11=time_gap(11,1:b(11,1)-1);
40 x13=time_gap(13,1:b(13,1)-1);
```

绘图，得到图 4.11。

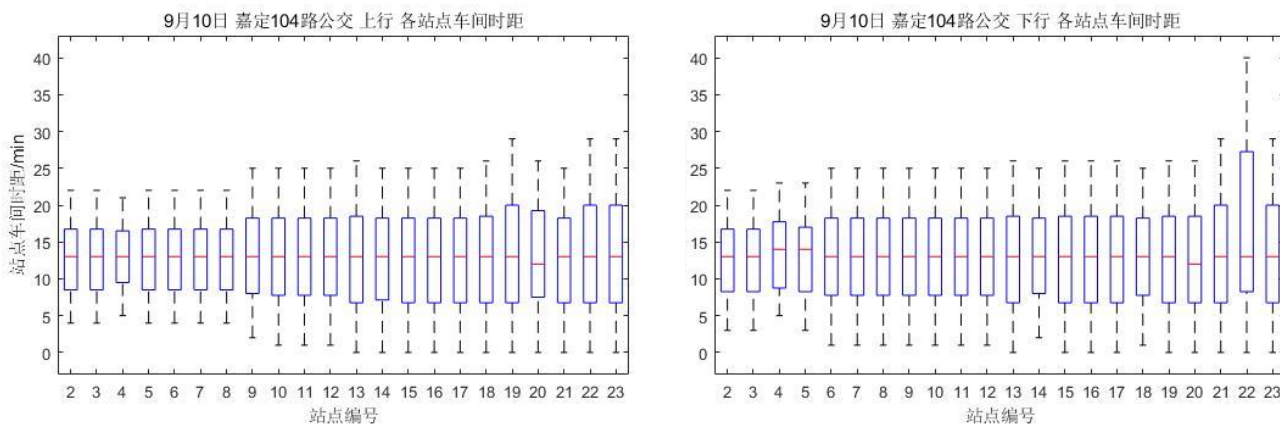


图 4.11 嘉定 104 路 9 月 10 日 上/下行各站点车间时距

观察发现，随着行驶距离的增长，上/下行的各站的车间时距基本处于不断增大的情况；在同样的刻度下，下行车间时距的增大出现站点序号比上行更早。



4.4. 站点乘客延误分析

取 9 月 10 日公交车 GPS 轨迹数据，分析该天乘客站点平均延误情况。其中，站点乘客平均延误可通过下式计算：

$$E(w) = \frac{1}{2}E(h)(1 + \frac{Var(h)}{E(h)^2})$$

MATLAB 代码如下：

```
1 %3.3中已经计算出不同站点的车间时距
2 %以上行为例 x:22*各个站点次数-1
3 x=[x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,x12,x13,x14,x15,x16,x17,x18,x19,x20,x21,x22,x23];
4 %b为站点经过次数
5 %计算样本方差1*22
6 var_s=var(x);
7 %计算均值1*22
8 e_s=mean(x);
9 %计算站点乘客平均延误1*22
10 yanwu=0.5*e_s(1,:)*(1+var_s(1,:)/e_s(1,:)/e_s(1,:));
11 %画图
12 subplot(1,2,1);
13 a=2:1:22;
14 plot(a,yanwu);
15 xlabel(' 站点编号');
16 ylabel(' 站点乘客平均延误/min');
17 title('9月10日 嘉定104路公交 上行 站点乘客平均延误');
18 set(gca,'YLim',[0 15]);
```

绘制，得到图 4.12。

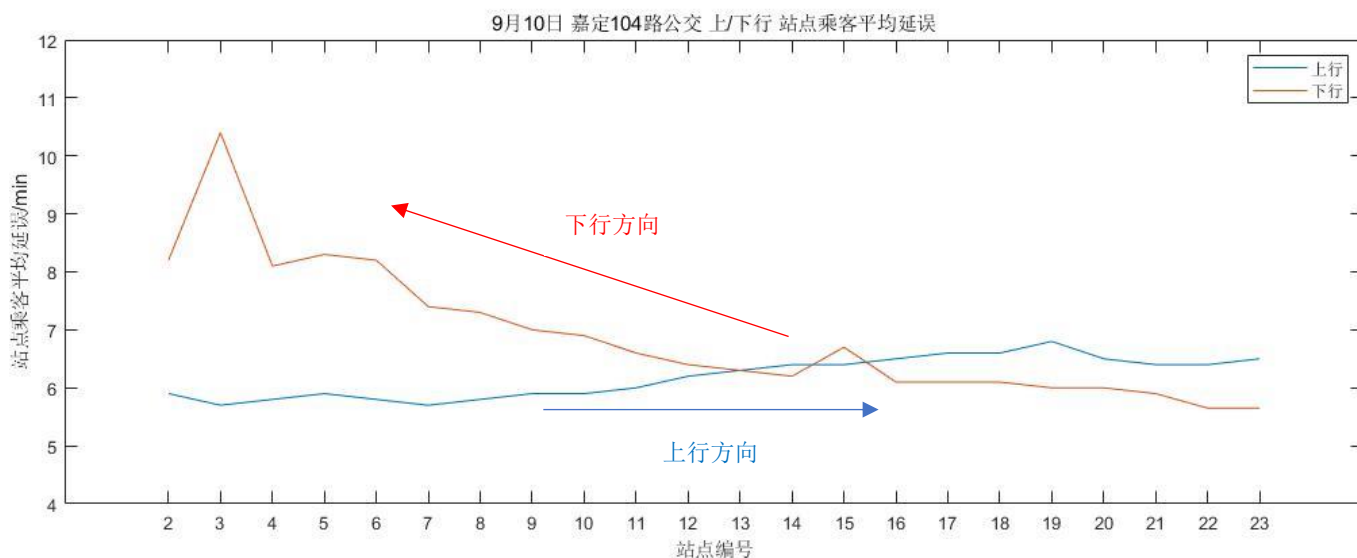


图 4.12 各站点乘客平均等待时间

观察发现：

- ① 无论对于上行方向还是下行方向，随着行驶距离的增加，站点乘客的平均延误呈现增大趋势；
- ② 上行方向的站点乘客平均延误在 7min 以内，下行方向的站点乘客平均延误起伏较大，对于值最大的一个站点——三号站，其乘客平均延误超过了 10 分钟。



5. 问题总结与建议

本次课设首先对嘉定 104 路公交在 2021 年 9 月 6 日至 9 月 10 日的 GPS 轨迹数据进行了数据处理，主要包括地图匹配、二次修正、时空轨迹转换、数据剔除与修复、对数据基础的补充五个部分。

第二部分是对公交运行状况进行了定性分析，主要包括对公交运行轨迹的分析、车辆使用情况的分析、线路运行指标（出发准点率、时刻表偏差、发车间隔、运营时间与 Layover 时间）的分析。

分析得到嘉定 104 路公交运营存在的问题主要如下（简略描述，具体分析见各章节）：

- ① 早、晚高峰时段仍有部分路段存在较为严重的堵塞现象，这导致了较大的时刻表偏差，使出发准点率下降，增大了繁忙站点乘客的平均等待时间；并因此导致部分车辆出现空驶现象以及串车现象。尽管已经减小了该时段的车头时距以缓解压力，但仍存在一定的改善空间；
- ② 空闲时段嘉定 104 路公交出现的时刻表偏差多为负偏差，即公交车提前到达；
- ③ 早、晚高峰的实际发车间隔比时刻表的发车间隔差距较大，通常情况下早高峰发车间隔比计划表大，晚高峰发车间隔比计划表小；
- ④ 嘉定 104 路公交在周一（9 月 6 日）的准点率最差，应改善时刻表安排；

最后对公交路段运行情况进行了分析，主要包括路段运营时间与延误、停靠时间分析、站点车间时距分析以及站点乘客延误分析，总结出多个公交与乘客延误问题较严重的路段与停靠时间较长的站点，并结合路段附近 POI 推测了这些现象出现的原因：

这些现象可以通过改变时刻表安排去改善——早、晚高峰时段适量减少车头时距；；单独设置周一的时刻表安排，增加其准点率，改善其运营表现；等等。

对于串车现象，一种常用的方法是临时调度员指示晚点的公交车在没有乘客下车的站点不停车、限制每一站上车的人数，以此减少吃到的公交车在每一站停留的时间，追赶时刻表。但是这一问题会导致各站点想要上车的乘客滞留，因此可以为公交时刻表预留更多的时间，即“缓冲时间”，让提前到站的公交车在特定车站停留，等到预定时间再出发，即“延时”。对于迟到的公交车，则只能在线路上以更快的速度行驶，或不按线路行驶以追赶时刻表。