



## 智能交通系统课程设计

题目：VMS 可变信息板对交通系统的影响分析

姓 名： 王倩妮

班 级： 交通 2015-02 班

学 号： 2015112956

任课教师： 郑芳芳

2018 年 5 月 8 日

## 目录

一、 文献综述.....	1
1.1 先进的出行者信息服务系统.....	1
1.2 路径选择与交通诱导.....	1
二、 问题描述及实现过程.....	2
2.1 问题描述.....	2
2.2 实现过程.....	3
2.2.1 模型建立.....	3
2.2.2 程序实现.....	4
三、 评价指标及计算方法.....	8
3.1 平均延误.....	8
3.2 平均性能.....	8
3.3 总延误.....	9
3.4 总排队长度.....	9
3.5 路径 1 延误可变性.....	9
3.6 路径 2 延误可变性.....	9
3.7 总延误可变性.....	9
四、 影响分析.....	10
4.1 分析方法.....	10
4.2 提供瞬时交通信息与预测交通信息分析.....	10
4.3 不同敏感度分析.....	12
4.4 正常交通状况与交通事件发生情况下.....	15
4.5 综合分析.....	16
五、 心得体会.....	20
六、 参考文献.....	21
附录.....	22
附录 1: 实现程序.....	22
附录 2: 评价结果表格.....	27

## 一、文献综述

### 1.1 先进的出行者信息服务系统

先进的出行者信息服务系统（Advanced Traveler Information System, ATIS）是一种能够交通需求管理手段，通过地理信息数据、交通运行数据、道路信息数据、公共交通信息数据等使用计算机和通信技术采集数据、传输数据从而对数据进行进一步处理，将有价值的信息进行发布，向出行者提供出行前或出行途中的道路交通信息服务。<sup>[1]</sup>

ATIS 主要提供①公共交通信息服务②出行者信息服务③路径诱导及导航服务④个性化信息服务等，作为一种有效的交通需求管理手段，近年来国内外均对其进行了较为深入的研究，希望其达到缓堵保畅、提供交通系统运行效率的作用。但由于交通系统的复杂性与信息系统构建的复杂性以及 ATIS 的广泛性，此方向仍具有较大的发展潜力。

以美国、欧洲为主导的西方与近代发展迅猛的日本在 ATIS 领域都有多年的研究与实践经验，研究多以路径诱导为核心，通过实时的交通信息帮助出行者选择起讫点间的最优路径，从而能够最大限度防止拥堵的发生，提高安全、效率，满足出行者需求。

ATIS 的信息发布手段有诸如可变信息板、交通广播系统、车载终端等，近年来国内外在这些方面均有所发展，一些大中城市也逐步布设信息发布设施，建立信息中心等，为信息的发布提供便利。但目前在我国，仍存在一下较为突出的问题①信息缺乏共享，造成资源浪费，信息系统不够完善；②如可变信息板的信息发布设施布局不合理，信息显示不统一，缺乏规范性；③由于发布信息的滞后性，诱导信息准确性差。

近年来，国内外研究者针对不同 ATIS 信息发布手段，通过不同的认识角度与研究方法对出行者的路径选择行为进行了一定的研究，比如 C.LEE<sup>[2]</sup>采用状态偏好(SP)方法研究了驾驶员依从性的影响因素，并利用二元 Logistic 回归和人工神经网络(ANN)对驾驶员依从性模型进行了研究。李志纯<sup>[3]</sup>等利用离散选择理论中的层次选择结构模型和交通规划理论中的随机均衡方法,研究了先进的旅行者信息系统对出行者终点选择,方式分担和路径选择行为的综合影响,建立了一个与网络均衡条件等价的数学规划模型,设计了模型的求解算法,并用一算例分析了市场渗透率和信息质量对出行者选择行为的影响。Hai Yang<sup>[4]</sup>通过构建混合平均分配模型，分析了驾驶员在不同获取信息能力时的路径行为选择，建立了 ATIS 的性能和需求均衡的存在性、唯一性和稳定性，并提出了一个迭代过程来计算 ATIS 市场渗透和由此产生的均衡网络流模式。

随着科学技术的发展，也有研究者开始关注车联网条件下的交通信息服务，车联网条件下的交通诱导因与实际状况有着较大差异，现还处于起步阶段。

### 1.2 路径选择与交通诱导

交通信息服务与交通诱导有着密切的联系，交通诱导也是智能交通系统的重要功能，交通诱导需要通过一系列数据采集、传输、处理分析手段，获知路网上的车辆分布，在

一定程度上合理引导交通流的时空分布。由于交通诱导与信息采集、处理、发布与出行者对于信息的理解和反映等因素都有着密切的关系，因此交通诱导效率提升一直是难点。目前在交通诱导方面主要分为交通信息对出行行为的影响研究以及又到信息发布策略研发两类。<sup>[5]</sup>

对于交通信息对出行行为影响研究主要通过调查统计(包含行为调查 RP 与意向调查 SP)<sup>[6]</sup>、仿真模拟实验两大研究方法，通过对出行行为数据综合分析，定性或定向分析交通信息对出行行为的影响。

对于诱导信息的发布策略研究大多通过研究网络交通流理论，通过均衡路网中的交通流量、缩短出行者的出行时间，进行发布形式、发布时间(诱导周期)、信息时空发布策略等进行研究，国内外均对以上与交通诱导密切相关的影响因素，利用各类方法进行了深入研究。学者们建立各类数学建模，设计各类算法，通过计算机模拟仿真、结合具体案例的 case study 验证等手段展开验证，推进了交通信息诱导技术的发展。但目前交通诱导的研究还存在一定问题，比如对于出行者接受信息作出选择的过程的分析并不够真实；人们对于诱导信息的反映的量化仍有难度，难以刻画、调查；信息发布主要考虑管理者的意愿，未充分考虑到 UE(User Equilibrium)与 SO(system optimization)的区别，从而造成了人们对于诱导信息的服从程度的差异性等。

因此，ATIS 仍有较大发展空间，在理论研究方面仍需更加贴近实际交通运行状况，此外在实际应用方面还需大量投资、先进管理手段进行前期建设、运营维护与更新，以保证 ATIS 的使用效果，从而真正达到缓解交通拥堵的目的。

## 二、问题描述及实现过程

### 2.1 问题描述

#### ● 问题描述

现系统内有两条道路：路径 1、路径 2，系统输入可以选择在两条路径之一行驶，两个路径上分别有外部交通量  $P_1(t)$ 、 $P_2(t)$  输入系统之中，可变信息板在入口处为驾驶员提供信息(不提供建议)，可变信息板对于系统运行状况的影响因为一系列因素而具有不同的效果。

若出行路径不受限制的出行者接收到 VMS 提供的交通信息时，会发生什么情况？请依照题目所提供数据，分别对以下情况做评估：

- 1) 当分别提供瞬时交通信息和预测交通信息时
- 2) 不同敏感性情况下
- 3) 正常交通状况与交通事件发生情况下

分别从不同的评价指标（如：平均延误、平均系统性能、总延误）来表述。

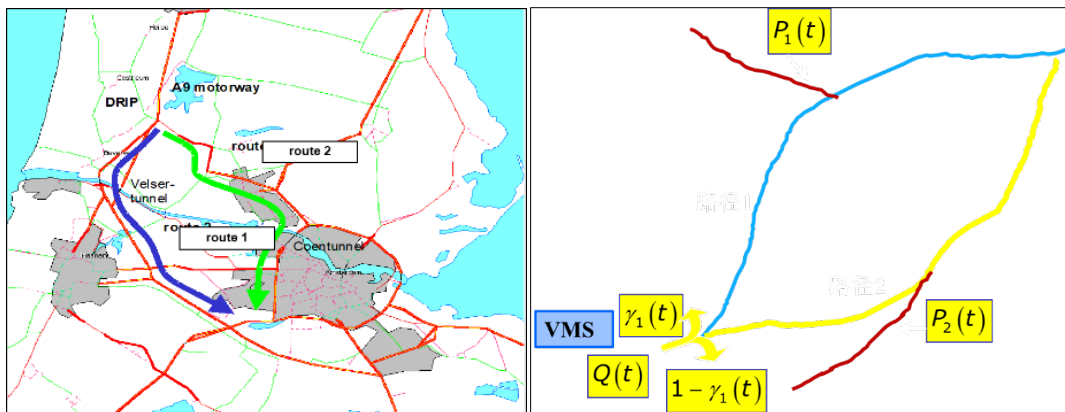


图 1 荷兰阿姆斯特丹路网地理位置示意图 图 2 荷兰阿姆斯特丹路网概要及输入示意图

● 数据定义

表 1 题设数据定义表

序号	符号	定义
1	$i(t)$	交通需求 (veh/h)
2	$p1(t)$	路径 1 的外部交通需求量 (veh/h)
3	$p2(t)$	路径 2 的外部交通需求量 (veh/h)
4	$i1(t)$	路径 1 的实际交通需求量 (veh/h)
5	$i2(t)$	路径 2 的实际交通需求量 (veh/h)
6	$C1(t)$	路径 1 的容量 (通行能力) (veh/h)
7	$C2(t)$	路径 2 的容量 (通行能力) (veh/h)
8	$r1(t)$	路径 1 的排队车辆数
9	$r2(t)$	路径 2 的排队车辆数
10	$L1(t)$	路径 1 的排队长度 (km)
11	$L2(t)$	路径 2 的排队长度 (km)
12	$D1(t)$	由于车辆排队导致路径 1 的延误 (h)
13	$D2(t)$	由于车辆排队导致路径 2 的延误 (h)
14	$sp(t)$	选择路径 1 的百分比
15	$D_{tot}$	总延误 (h)
16	$P_{tot}$	总的系统性能 (veh x km)

## 2.2 实现过程

### 2.2.1 模型建立

进行 VMS 可变信息板对交通系统影响的分析需按照以下逻辑过程构建:

- A. 排队长度与行程时间计算, 需知路径流量;
- B. 路径流量取决于路径选择比率, 需知路径选择比率;
- C. 路径选择比率由出行路径不受限制的出行者的习惯性选择参数  $\beta_0$  和对交通信息的

敏感性  $\beta$ ，以及其他出行者(出行路径受限)的“缺省选择率”决定；

因此该过程是循环迭代的过程，每步骤的计算公式如下：

A. 排队车辆数与行程时间

- 排队车辆数  $N_j$  (程序中使用  $r_j$  表示排队车辆数)

$$N_j(t + \Delta t) = N_j(t) + \Delta t [Q_j(t - T_j^{free}) + P_j(t) - C_j(t)] \quad [1]$$

- 瞬时行程时间  $T_j^{inst}$

$$T_j^{inst}(t + \Delta t) = T_j^{free} + \frac{N_j(t)}{C_j(t)} \quad [2]$$

- 预测行程时间  $T_j^{exp}$

$$T_j^{exp}(t) = T_j^{free} + \frac{N_j(t + T_j^{free})}{C_j(t + T_j^{free})} \quad [3]$$

B. 路径流量  $Q_j$  (程序中使用  $i_j$  表示)

$$Q_j(t) = \gamma_j(t) Q(t) \quad [4]$$

C. 路径选择比率  $\gamma_j$  (程序中使用  $sp$  表示选取路径 1 的比例)

$$\begin{aligned} \gamma_j(t) &= [1 - P^{nc}] \gamma_j^{capitives}(t) + P^{nc} \gamma_j^{nc}(t) \\ \gamma_j^{nc}(t) &= \beta_0 + \beta [T_k^{inst}(t) - T_j^{inst}(t)] \quad or \\ \gamma_j^{nc}(t) &= \beta_0 + \beta [T_k^{exp}(t) - T_j^{exp}(t)] \end{aligned} \quad [5]$$

### 2.2.2 程序实现

通过对模型迭代过程进行分析，现依靠 Python 编程构建此迭代过程。程序总体实现框架如下图所示。在循环过程构建时，以时间  $k$  为索引，因此经实践发现，分别构建反馈控制控制与预测控制过程，能够有效避免时间的前推与后推过程产生的问题。而诸如敏感度、事件是否发生、事件持续时长等参数，可在后期通过循环遍历，并调用反馈控制函数与预测控制函数实现变化与结果输出。下面，将分别对各个过程进行说明。

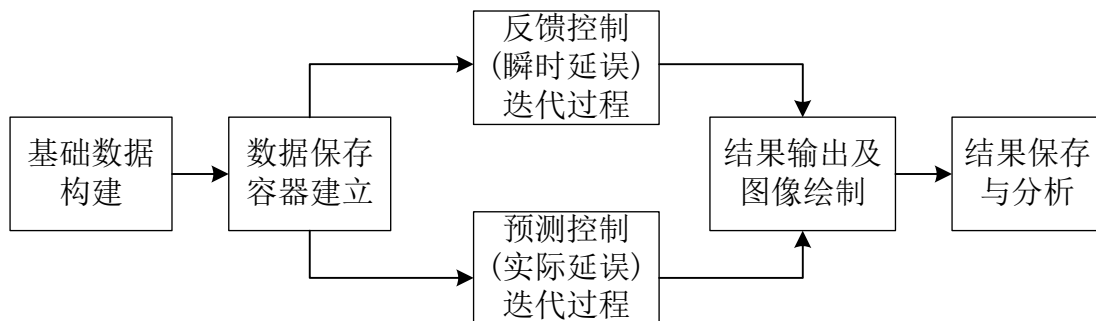


图3 程序实现流程

#### • Step1: 基础数据构建

此过程中部分基础数据固定，因此，首先对于这些参数进行设定，参数数值与含义

可由下表获知。(由姓名学号生成 code=218)

表 2 程序基础数据对应表

数值	含义	数值	含义
T = 552	仿真时长	TT1 = 18	路径 1 自由流时间 min
basic_i = 2180	基础交通流量 veh/h	TT2 = 16	路径 2 自由流时间 min
basic_P1 = 2491	外部交通需求量 P1veh/h	basic_L1 = 30	路径 1 长度 km
basic_P2 = 3354	外部交通需求量 P2veh/h	basic_L2 = 27	路径 2 长度 km
basic_C1 = 5232	基础路径 1 通行能力 veh/h	B0 = 0.58	缺省路径 1 选择比例
basic_C2 = 5232	基础路径 2 通行能力 veh/h	TTacc = 20	事件持续时长 min
L_car = 0.005	车辆平均长度 km	acc_start	事件开始时间

### ● Step2: 数据保存容器建立

使用 Python 第三方库 numpy 进行数据处理, 首先对后期需要迭代的变量进行 array 初始化, 而后生成迭代所需的总需求量数据。

分别初始化  $i_1, C1, C2, P1, P2, i_1, i_2, D1, D2, r_1, r_2, L1, L2, Dtot, Ptot$  为长度为  $T=552$  的全 0 矩阵,  $sp$  初始化为数值均为默认选择比例的长度为  $T$  的矩阵。将  $P1, P2, C1$  按照基础数据设定为长度为  $T$  的矩阵,  $C2$  按照是否有事件发生分为  $0.1 * basic\_C2$  与  $basic\_C2$ 。

总需求量数据生成函数与原始文件生成方法一致。此处不再赘述。

### ● Step3: 控制迭代过程

控制过程的构建是一个循环迭代的过程, 关键在于找到循环起点与确定递推步骤。两种方式均需要率先指定前自由行驶时间(18/16s)内的部分数据, 之后按照如下顺序进行递推:

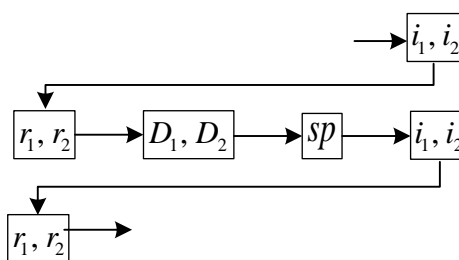


图 4 控制迭代过程示意图

#### ➤ Step3(1): 反馈控制(瞬时延误)迭代过程

反馈控制指将  $t$  时刻的数据状况告知  $t$  时刻获取 VMS 可变信息板的驾驶员, 使驾驶员做出相应决策的过程。

由于反馈控制采用瞬时的概念, 因此本循环只采用时间变量  $k$  遍历时间  $T$ 。

循环过程如下图所示:

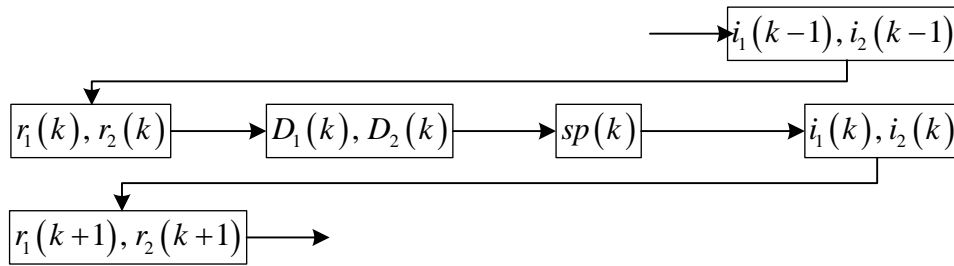


图 5 反馈控制迭代过程示意图

若  $k$  属于路径 1 和(或)路径 2 的起始的自由时间, 则将  $r_1[k], r_2[k]$  设定为 0, 否则, 利用公式  $r_j[t] = r_j[t-1] + \frac{i_j[t-1-T^{free}] + P_j[t-1] - C_j[t-1]}{60}$  进行计算。随后, 使用

$D_j = \frac{60 \times r_j}{C_j}$  计算  $D_1[k], D_2[k]$ 。对于  $sp$ (选择路径 1 的比率)的计算,  $sp[k]$  使用

$B_0 - B \cdot (D_1[k] - D_2[k])$  得到结果。计算得到  $sp[k]$  后即可由  $i_j(t) = ii(t) \times sp(t)$  得到  $i_1[k]$ , 利用  $i_2[k] - i_1[k]$ , 即可得到  $i_2[k]$ , 以此即可将  $i_1[k], i_2[k]$  带入  $r_j$  计算公式, 进入下一循环。

➤ Step3(2): 预测控制(实际延误)迭代过程

预测控制指将  $t + T_j^{free}$  的数据状况告知  $t$  时刻获取 VMS 可变信息板的驾驶员, 使驾驶员做出相应决策的过程。

时间的同步性对于递推的正确性尤为重要, 由于路径 1、2 时间差的存在, 循环采用  $j, k$  双线同步进行, 使  $k - TT_1$  与  $j - TT_2$  为同一时刻。时间维度示意图如下图所示:



图 6 预测控制迭代时间维度示意图

循环过程示意如下图所示:

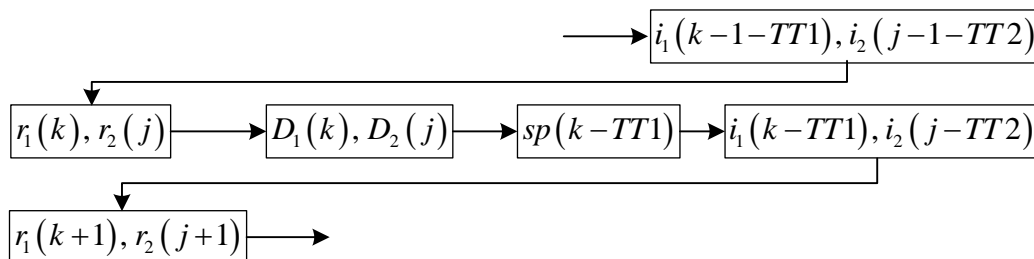


图 7 预测控制迭代过程示意图

使  $k$  遍历时间  $T$ , 在本题中  $j=k-2$ 。在起始的自由时间内, 将  $r_1[k], r_2[j]$  设定为 0,



随后, 使用  $D_j = \frac{60 \times r_j}{C_j}$  计算  $D1[k], D2[j]$ ; 若非起始自由时间, 则排队车辆数  $r1[k], r2[j]$

使用  $r_j[t] = r_j[t-1] + \frac{i_j[t-1-T^{free}] + P_j[t-1] - C_j[t-1]}{60}$  计算得到, 从而利用相同公式得到  $D1[k], D2[j]$ 。

对于  $sp$ (选择路径 1 的比率)的计算, 分路径 2 是否发生事件分别定义且路径 1 选择发生在  $k-TT1$  时刻。若路径 2 不发生事件, 则非起始自由时间内,  $sp[k-TT1]$ 采用  $B0-B*(D1[k]-D2[j])$ 进行计算; 若路径 2 发生事件, 则非起始自由时间且非事件持续时间内,  $sp[k-TT1]$ 也同样采用  $B0-B*(D1[k]-D2[j])$ 进行计算, 若事件时间内,  $sp[k-TT1]$ 采用  $B0-B*(D1[k]-(D2[j]+acc*TTacc))$ 进行计算,  $acc=1$  表示发生事件,  $TTacc$  指事件持续时间长。

计算得到  $sp[k-TT1]$ 后即可由  $i_j(t) = ii(t) \times sp(t)$  得到  $i1[k-TT1], i2[k-TT1]$ , 以此即可将  $i1[k-TT1], i2[k-TT1]$  带入  $r_j$  计算公式, 进入下一循环。

#### ● Step4、5: 结果输出图像绘制及结果保存

由于需要对于不同控制状态下, 不同事件发生状态及不同敏感度状况下的系统运行指标进行分析, 经排列组合可知, 本题目中的所有可能性如下所示:

表 3 题设参数可能性组合表

序号	是否事件发生 Accident	敏感度 B	是否反馈控制 react
0-20	0	[0.00,1.00];变化步长 0.05	1
21-41	1	[0.00,1.00];变化步长 0.05	1
42-62	0	[0.00,1.00];变化步长 0.05	0
63-83	1	[0.00,1.00];变化步长 0.05	0

由于需要通过图像对问题进行分析, 因此构造 `drawpicture` 函数, 为后期调用进行图像绘制提供方便。采用 Python 第三方绘图库 `matplotlib.pyplot` 子图构建方法构建  $2*2$  子图。

- 子图 1 绘制随时间变化路径 1、2 排队长度的变化;
- 子图 2 绘制随时间变化路径 1、2 延误的变化;
- 子图 3 绘制总需求及路径 1、2 的需求随时间的变化情况;
- 子图 4 绘制随时间变化选择路径 1 用户的比例。

在此过程中还需对  $x, y$  轴显示范围、坐标名称、图例设置、图片附注(参数状况)进行修改。最后添加保存命令用于图片保存。

生成图像效果如下图所示:

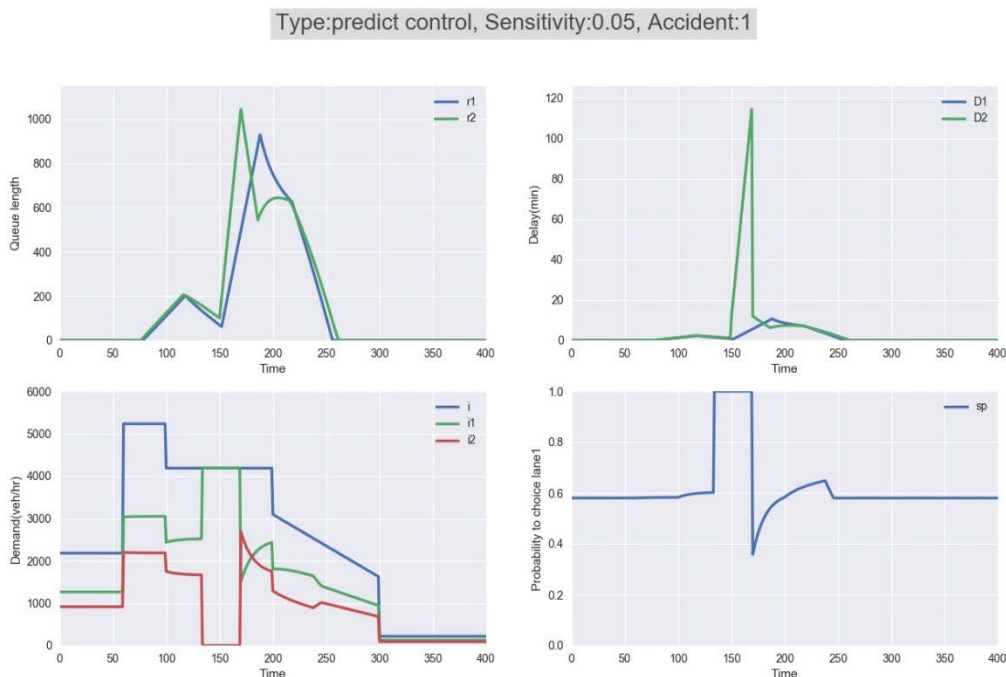


图 8 生成图像效果

我们还需通过参数的输出,对系统的运行进行评价,因此在这一过程中,我们选择:  
 ①平均延误、②平均性能、③总延误、④总排队长度、⑤路径 1 延误可变性、⑥路径 2 延误可变性、⑦总延误可变性七个指标进行输出,用于后期指标分析。指标计算方法具体见第三部分。通过将输出指标写入 Excel 文件,便于后期进行分析。

### 三、评价指标及计算方法

#### 3.1 平均延误

延误是指车辆在行驶中,由于受到驾驶员无法控制的或意外的其他车辆的干扰或交通控制设施等的阻碍所损失的时间。平均延误即每辆车在行驶中平均损失的时间。

平均延误使用下列公式进行计算:

$$AverageDelay = \frac{\sum_{t=0}^{T-1} D_{tot}(t)}{\sum_{t=0}^{T-1} ii(t)}$$

$$D_{tot}(t) = i_1(t)D_1(t+T_1^{free}) + i_2(t)D_2(t+T_2^{free})$$
[6]

其中  $i_1$ ,  $i_2$ ,  $D_1$ ,  $D_2$  可由程序计算得到。

#### 3.2 平均性能

本题目中使用交通需求\*路径长度定义“性能”。

平均性能的定义如下:

$$Averageperformance = \frac{\sum_{t=0}^{T-1} P_{tot}(t)}{\sum_{t=0}^{T-1} ii(t)} \quad [7]$$

$$P_{tot}(t) = i_1(t) \times L_1 + i_2(t) \times L_2$$

其中  $i_1, i_2$  可由程序计算得到,  $L_1, L_2$  为路径 1、2 的长度, 分别为 30km 与 27km。

### 3.3 总延误

总延误的定义如下:

$$SumDelay = \sum_{t=0}^{T-1} D_1(t) + \sum_{t=0}^{T-1} D_2(t) \quad [8]$$

其中  $D_1, D_2$  可由程序计算得到。

### 3.4 总排队长度

总排队长度定义:

$$sumrow = \frac{\sum_{t=0}^{T-1} r_1(t) + \sum_{t=0}^{T-1} r_2(t)}{T} \quad [9]$$

其中  $r_1, r_2$  可由程序计算得到。

### 3.5 路径 1 延误可变量

延误可变量指延误的偏差状况, 使用“标准差”对这一指标进行表示。路径 1 延误可变量定义如下:

$$stdD1 = \sqrt{\frac{\sum_{t=0}^{T-1} (D_1(t) - \bar{D}_1)^2}{T-1}} \quad [10]$$

### 3.6 路径 2 延误可变量

路径 2 延误可变量定义如下:

$$stdD2 = \sqrt{\frac{\sum_{t=0}^{T-1} (D_2(t) - \bar{D}_2)^2}{T-1}} \quad [11]$$

### 3.7 总延误可变量

总延误可变量定义如下:

$$stdDt_{tot} = \frac{1}{60} \sqrt{\frac{\sum_{t=0}^{T-1} (Dt_{tot}(t) - \bar{Dt}_{tot})^2}{T-1}} \quad [12]$$

## 四、影响分析

### 4.1 分析方法

采用对照分析的方法，本题设中共有 A. 是否事件发生 Accident B.敏感度 C.是否反馈控制 react 三个变量，分析某个要素的影响状况时，控制其他要素保持不变。分析主要通过对比可视化结果与系统输出指标进行，分析内容诸如峰值、均值、标准差(波动程度等)。

此外，在进行附图分析时，排列的先后程度为[反馈控制→预测控制]; [事件发生→无事件发生]; [敏感度小→敏感度大]。

### 4.2 提供瞬时交通信息与预测交通信息分析

提供瞬时信息与预测信息的区别，已在模型建立与程序实现部分进行详细叙述，此处不再分析二者的区别。

选用发生与未发生事件，敏感度分别为 0.05, 0.5, 0.8 状态下采用瞬时信息与预测信息得到的指标变化图。

对比方式：横向对比每张图的左右子图

通过对比，我们不难发现以下结论：

- ① 采用瞬时信息相比采用预测信息，会给系统带来更多的扰动。这种扰动具体体现在路径 1,2 的需求量、选择路径 1 用户的比例、路径 1,2 的排队长度呈现更强的波动性，图像具有更多的峰。这种现象的明显程度也会随着敏感度的提高而变得明显。此规律也可通过对比下表中相同色块的 stdDtot, stdD1, stdD2 数据得到。
- ② 采用瞬时信息相比采用预测信息会带来更长时间的延误，此规律可以通过对比左右子图的延误变化图像峰值得到，与此同时，此规律也可以通过对比下表中相同色块的 sumdelay, averagedelay 数据得到。
- ③ 排队长度、路径选择、延误间存在着相互作用，这点也可以在模型中体现。在一定阈值以上，选择某路径的人越多，该路径排队车辆数排队长度就越长，在该路径上运行的延误就越大，在驾驶员得知这类消息后就会做出新的决定，经过一定程度的“路径选择博弈”后，选择各条路径的状况趋于稳定(需求在 0-300 内处于较大水平，相关变量在 <300 时便趋于稳定)。

表 4 输出指标结果

Ty- pe	Sensit -ivity	Acc	avgdel ay	avgperfor mance	sumdelay	sumrow	stdD1	stdD2	stdDtot
0	0.05	0	0.4910	28.7541	228.0053	36.0182	0.4970	0.5616	40.3227
0	0.5	0	0.4957	28.7569	226.3193	35.7519	0.5230	0.5321	40.6888
0	0.8	0	0.4961	28.7570	226.3138	35.7510	0.5246	0.5304	40.7103
0	0.05	1	2.6077	28.9178	2606.0294	232.7125	2.5549	13.1976	163.5765
0	0.5	1	2.6261	28.9272	2567.9988	232.2693	2.5184	12.8876	163.7396

0	0.8	1	2.6281	28.9278	2566.4099	232.2645	2.5219	12.8735	163.7960
1	0.05	0	0.4881	28.7604	228.5935	36.1112	0.5034	0.5711	40.4066
1	0.5	0	2.3758	28.7139	1154.0876	182.3124	1.8548	2.5419	145.9018
1	0.8	0	2.4625	28.7134	1195.6273	188.8745	1.9361	2.6182	152.1353
1	0.05	1	4.0061	28.9123	3218.0091	238.9725	2.4830	19.8877	408.7292
1	0.5	1	3.9679	28.9184	4190.9145	372.4512	3.6825	19.9859	254.4982
1	0.8	1	4.0824	28.9085	4183.8656	379.0741	3.7289	19.5260	261.3640

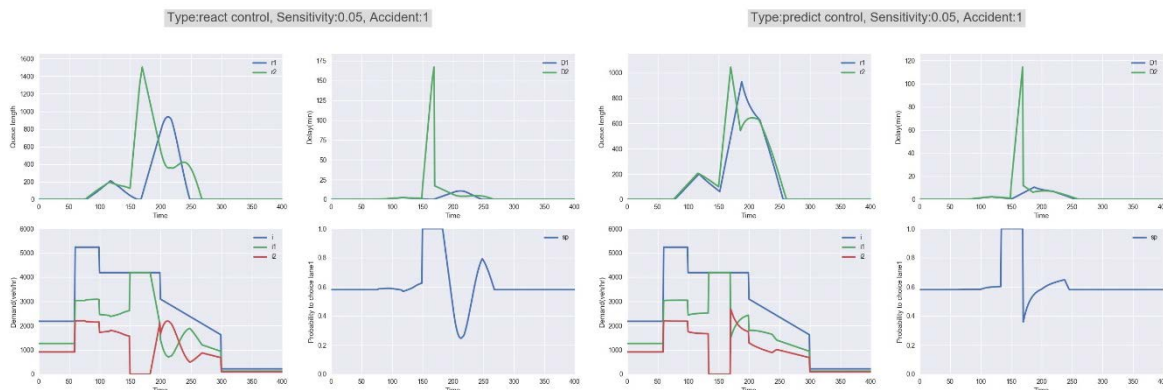


图9 发生事件、敏感度为0.05时瞬时信息与预测信息指标变化图

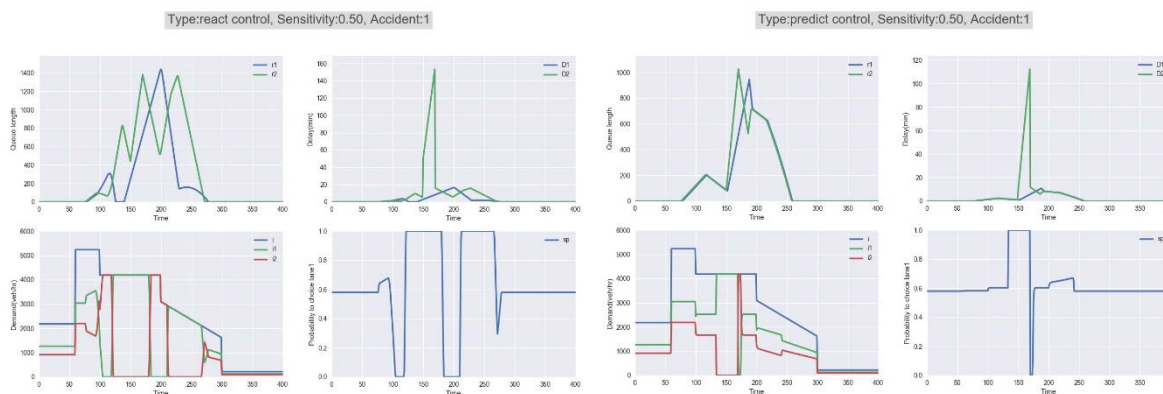


图10 发生事件、敏感度为0.50时瞬时信息与预测信息指标变化图

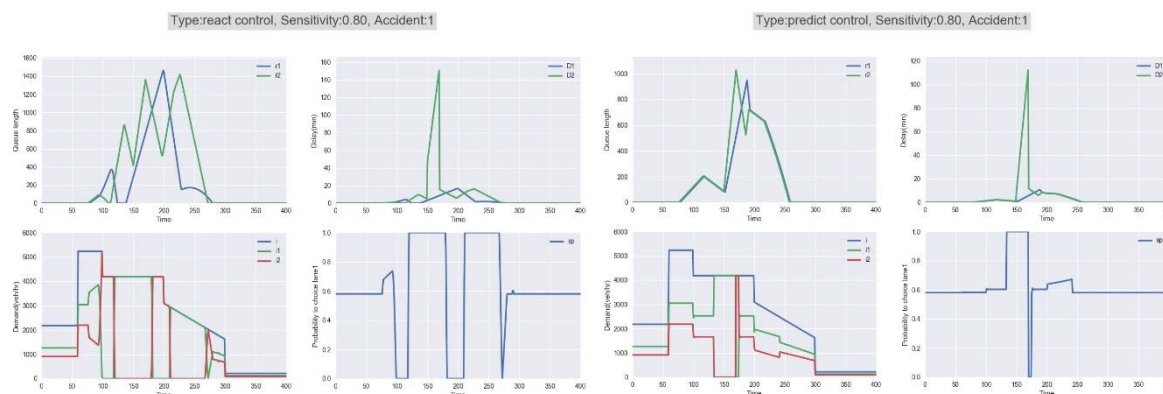


图11 发生事件、敏感度为0.80时瞬时信息与预测信息指标变化图

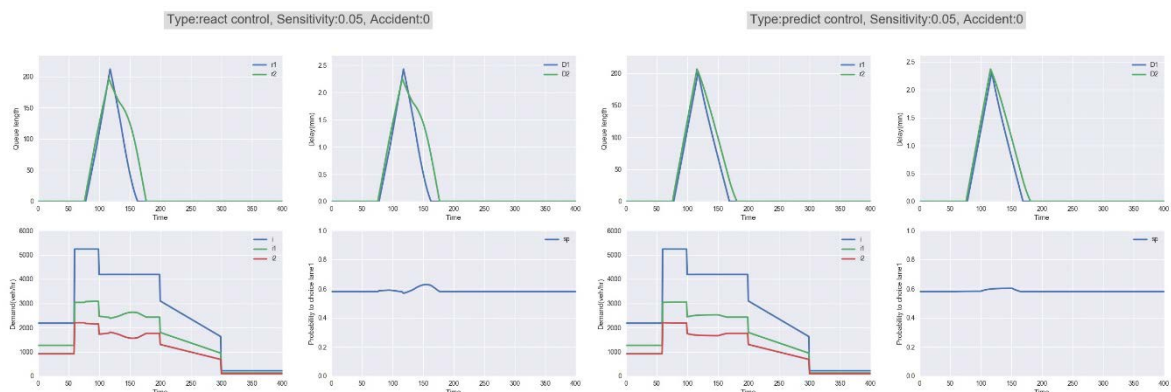


图 12 未发生事件、敏感度为 0.05 时瞬时信息与预测信息指标变化图

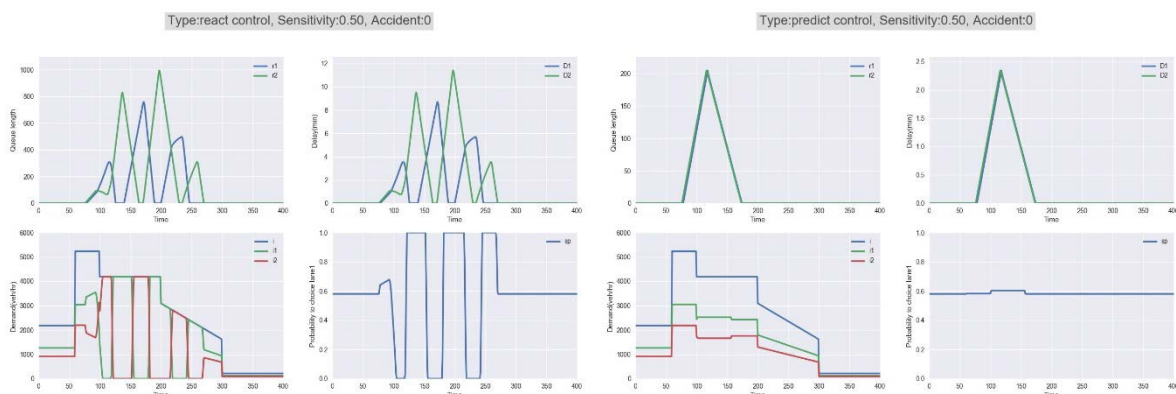


图 13 未发生事件、敏感度为 0.50 时瞬时信息与预测信息指标变化图

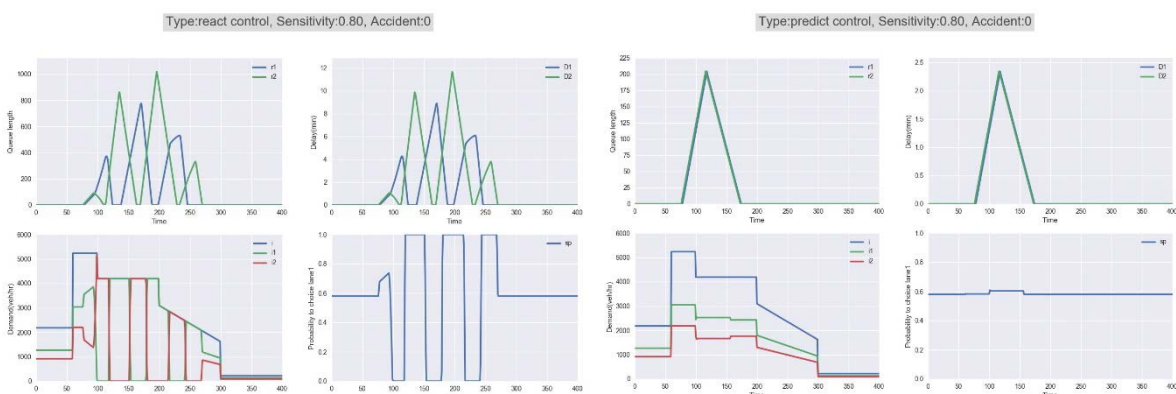


图 14 未发生事件、敏感度为 0.80 时瞬时信息与预测信息指标变化图

### 4.3 不同敏感度分析

敏感度指人们对于 VMS 可变信息板提供信息的接受程度，指人们对于两条路径的延误差作出选择(本题中通过  $sp$  表示选择路径 1)的情况。

首先，选用发生与未发生事件，敏感度分别为 0, 0.05, 状态下采用瞬时信息与预测信息得到的指标变化图。

横向对比每张图的左右子图，我们能够发现

- ① 敏感度在 0 与 0.05 变化的过程中造成了最大的影响，之后随着敏感度的均匀增加，变化幅度逐渐减小，说明毫不敏感与存在一定敏感度对于结果存在显著影响。通过



观察子图 1,2 与子图 1,3 可见这种改变体现在, 有事件状况下, 获知信息后, 人们在事件时段内选择路径 1, 事件结束后, 由于路径 1 延误过大而更多人选择路径 2, 路径 2 产生延误, 路径 1 延误逐渐疏解; 在无事件的情况下, 获知信息后, 系统将逐渐趋向于一种“均衡”状况, 路径 1、2 的排队长度、延误等参数均趋于一致。

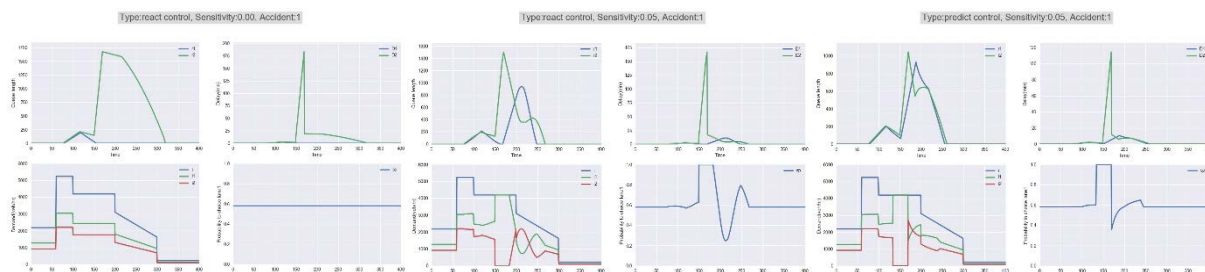


图 15 敏感度 0 有事件,敏感度 0.05 有事件反馈控制, 敏感度 0.05 有事件预测控制对比

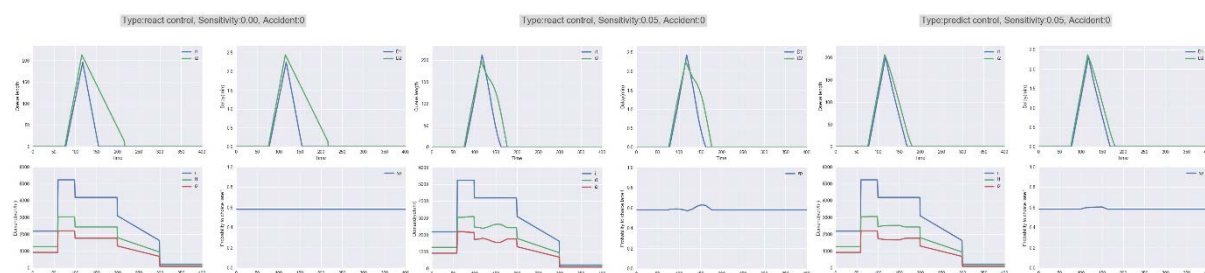


图 16 敏感度 0 无事件,敏感度 0.05 无事件反馈控制, 敏感度 0.05 无事件预测控制对比

② 敏感度变化对瞬时信息(反馈控制)的影响远大于敏感度变化对预测信息(预测控制)的影响。以无事件发生的情况为例, 选取敏感度为 0.1, 0.15, 0.6, 0.9 的反馈控制与预测控制效果图, 通过反馈控制效果图可见, 随着敏感度的增加, 选择路径 1 比例的线型有了明显的变化, 由双峰逐渐过渡到四峰的状态, 且峰值增加, 峰值持续时长增长; 路径 1,2 的排队长度(车辆数) 的峰值、均值随敏感度增加而增加(y 坐标维度存在差别, 需放大观看), 路径 1,2 延误变化呈现相似趋势; 路径 1,2 的需求量变化波动性更强, 差异更加极端。而通过预测控制效果图可见, 随敏感度的增加, 各项指标无明显变化。因此可见, 敏感度变化对反馈控制影响更大。通过预测控制分析同样可以印证此结论, 此处不再附图分析。此外结论①中, 随着敏感度均匀变化, 各指标变化幅度减小的结论也可由该图示证明。以上结论也可通过比较下表不同色块间的 avgdelay, sumdelay, sumrow, stdD1, stdD1, stdDtot 数据得到。

表 5 输出指标结果

Type	Sensitivity	Acc	avgdelay	avgperformance	sumdelay	sumrow	stdD1	stdD2	stdDtot
0	0.1	0	0.4925	28.7556	226.7274	35.8164	0.5086	0.5474	40.4942
0	0.15	0	0.4936	28.7562	226.4918	35.7791	0.5139	0.5416	40.5677
0	0.6	0	0.4959	28.7570	226.3154	35.7513	0.5237	0.5314	40.6983
0	0.9	0	0.4962	28.7570	226.3138	35.7510	0.5249	0.5301	40.7143
1	0.1	0	0.5652	28.7737	263.9021	41.6889	0.5374	0.5881	40.6989
1	0.15	0	1.3998	28.7587	666.7306	105.3241	1.1365	1.4799	80.0262
1	0.6	0	2.4190	28.7129	1174.3865	185.5190	1.8974	2.5735	148.8004

1	0.9	0	2.5065	28.7153	1215.8294	192.0658	1.9751	2.6427	154.8697
---	-----	---	--------	---------	-----------	----------	--------	--------	----------

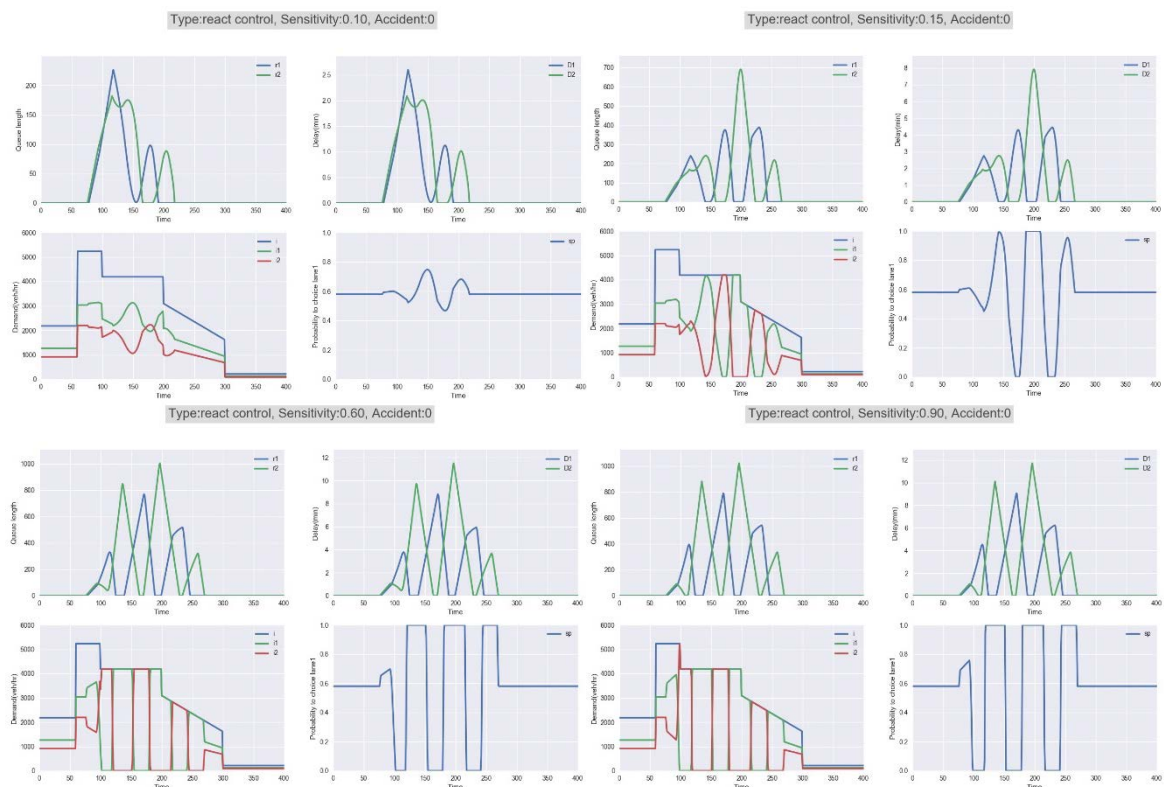


图 17 无事件反馈控制随敏感度增加

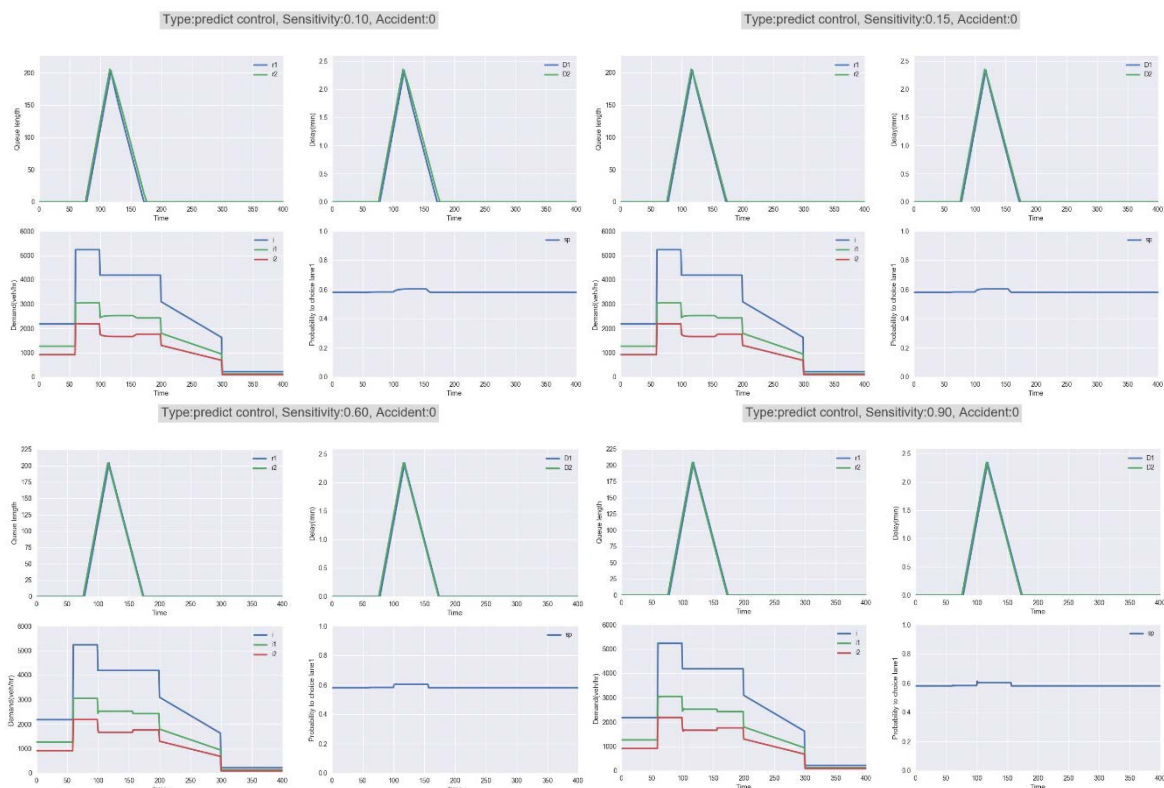


图 18 无事件预测控制随敏感度增加



### 4.4 正常交通状况与交通事件发生情况下

系统运行过程中，路径 2 存在发生事件的可能性，且事件发生与否与系统运行各项指标有着密切的联系，现做如下分析。

- ① 无论反馈控制与否与敏感度数值为何值，有事件发生时，子图 1,2 趋势非同步；无事件发生时，子图 1,2 趋势为同步。这种非同步出现的原因主要是事件发生对于系统的影响。其中子图 1 为路径 1,2 排队长度(排队车辆数)随时间的变化状况，子图 1,2 为路径 1,2 延误随时间的变化。

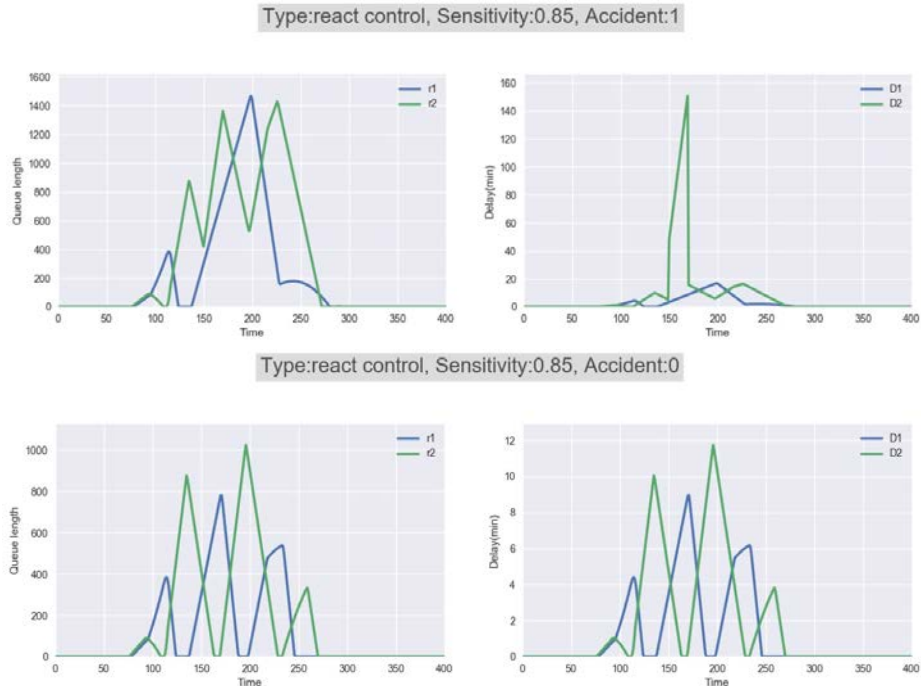


图 19 反馈控制，敏感度为 0.85 时事件发生与否子图 1,2 的变化状况

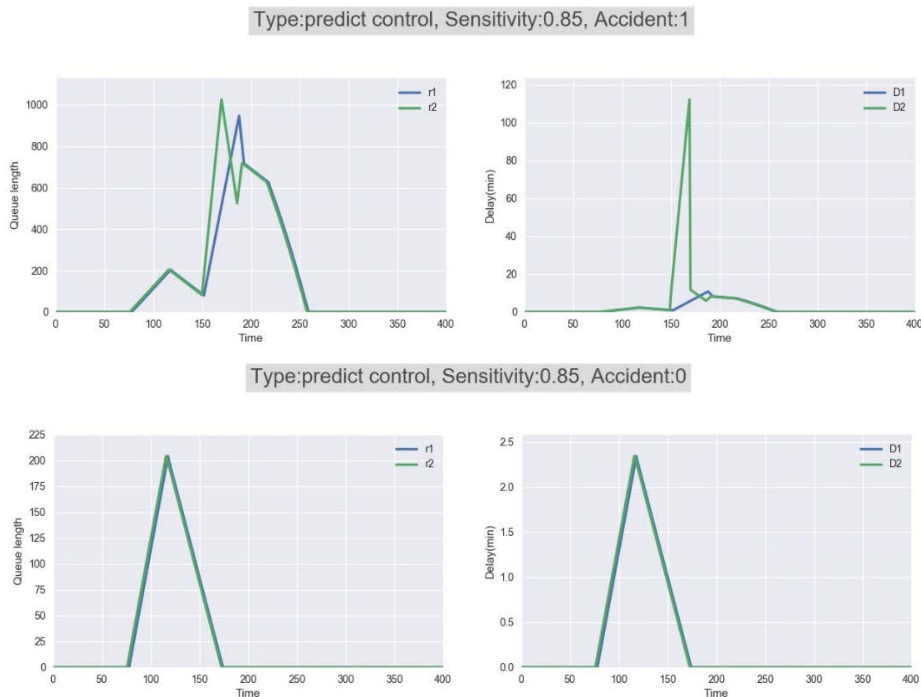


图 20 预测控制，敏感度为 0.85 时事件发生与否子图 1,2 的变化状况

- ② 事件发生时段事件产生的影响远大于其他因素对于系统的影响，但系统具有一定稳定性，在事件发生后的一段时间内即可恢复正常。通过事件发生状态下的延误数值，可见在事件发生时段内，事件发生路径延误较高，人们普遍会选择替代路径出行。而这种选择将造成替代路径的拥堵，随着事件的结束，在几次路径决策过程后系统将恢复至稳定状态。

### 4.5 综合分析

交通系统的状态与三个自变量因素均有着密切的联系，下面的分析将以敏感度为主要变化因素，分析在不同条件下，随敏感度变化，系统指标的变化状况，与此同时分析得到较优的敏感度区间。

不同条件下敏感度不同带来的差异可以通过以下四个动图较为生动地展现，通过动态图像可以看到敏感度的差异造成的指标在时间维度上的变化状况。通过动图也可以观察到 4.1-4.4 部分分析得到的相关结论。

**注：以下为动态 gif，可双击查看效果较好！**

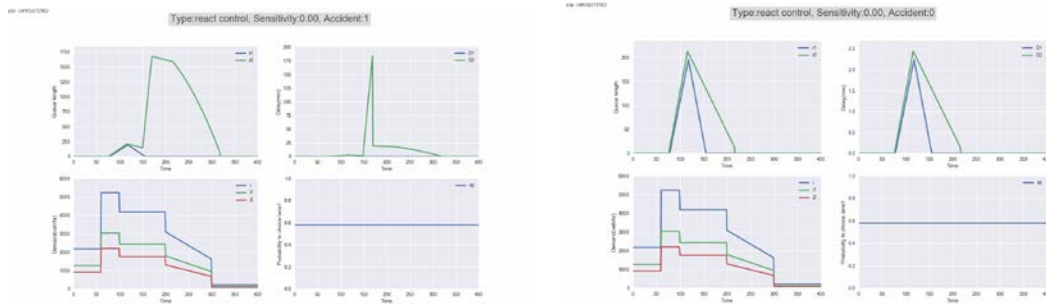


图 21 反馈控制发生事件，指标随敏感度变化动图 图 22 反馈控制无事件，指标随敏感度变化动图

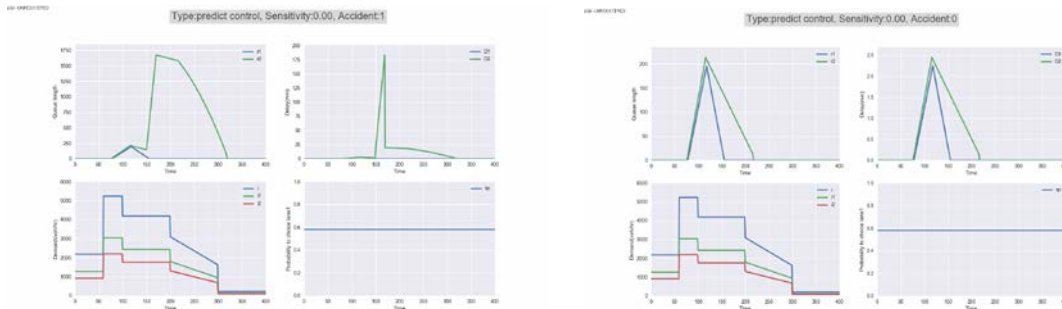


图 23 预测控制发生事件，指标随敏感度变化动图 图 24 预测控制无事件，指标随敏感度变化动图

动图反映了时间维度的差异性，而随敏感度变化总体运行状况则需要通过总体指标进行观察。考虑到指标间的相似性，考虑以敏感度为横坐标，选取平均延误、平均性能、总排队长度、总延误可变性为纵坐标分别绘制图像。以比较同一状态下各指标的变化情况。

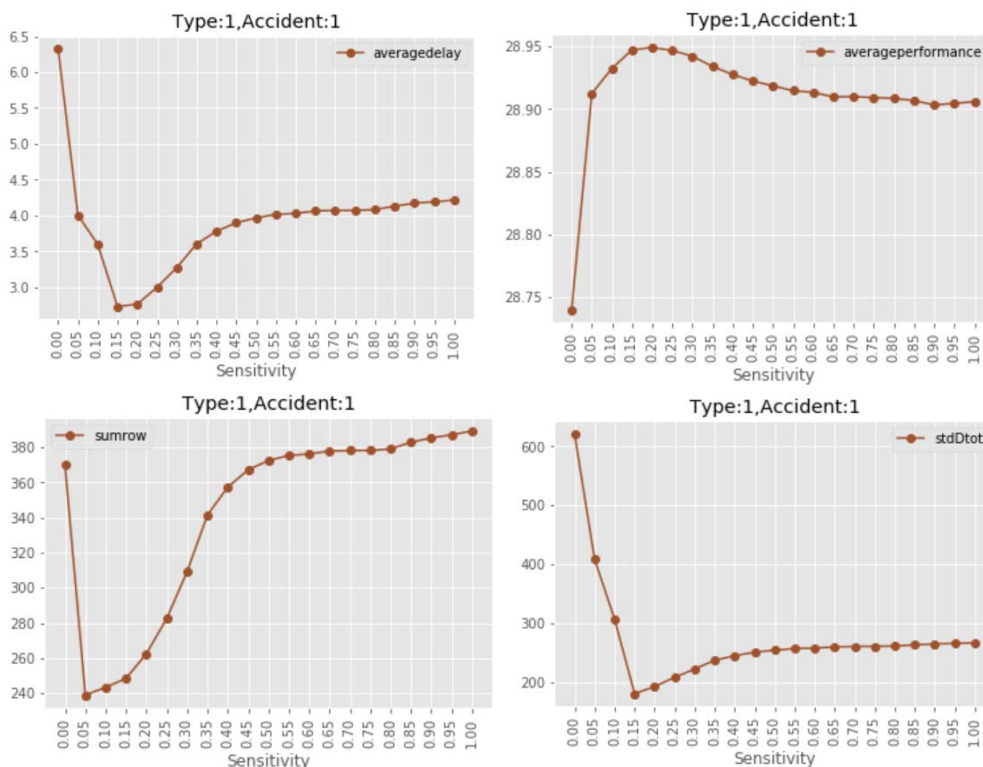


图 25 反馈控制、事件发生时总体指标随敏感度变化图

- 在反馈控制，有事件发生的状态，随着敏感度的增加，平均延误先下降后上升，平均性能先上升后有小幅下降且下降速度逐渐减慢，总排队长度先下降后逐渐上升，上升速度先快后慢最后又有小幅加快，总延误可变性先下降后上升最后趋于平稳。由此可见在反馈控制且有事件发生的情况下，敏感度在[0.15,0.20]区间系统可以得到较好的效果。造成这种现象主要是因为反馈控制信息与实际信息存在一定差异，即在一定程度上提供了错误信息，这种信息对于系统将产生不良影响。

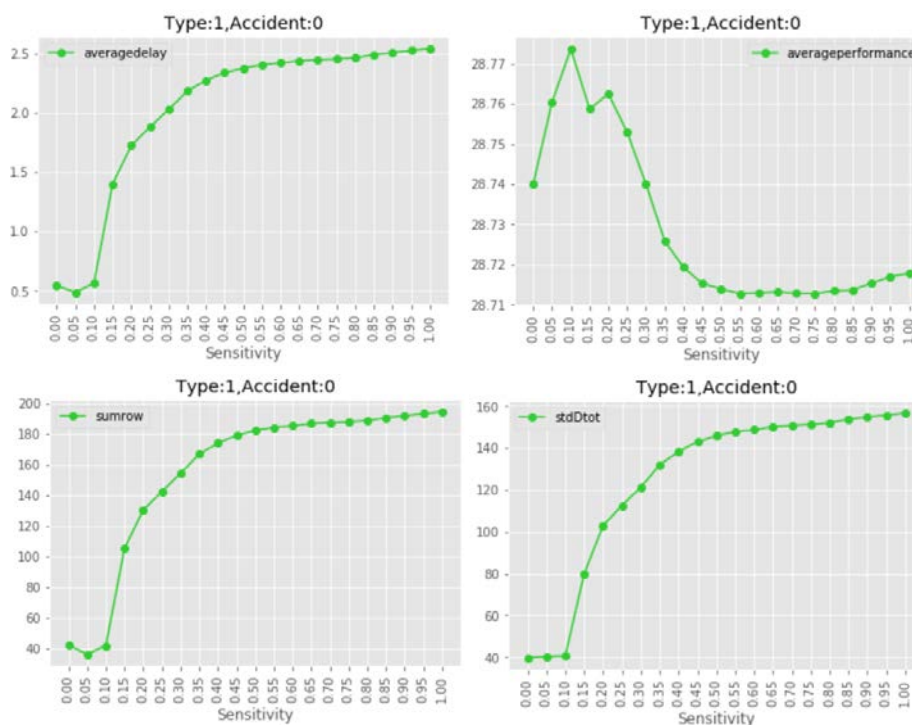


图 26 反馈控制、无事件发生时总体指标随敏感度变化图

- 在反馈控制、无事件发生的状况，随着敏感度的增加，平均延误先较平缓而后逐渐上升且上升速度下降，平均性能先上升后下降再趋于平缓，总排队长度、总延误可变性趋势与平均延误的趋势一致。在反馈控制，无事件发生的状态，敏感度在 0.10 左右系统运行效果较好。

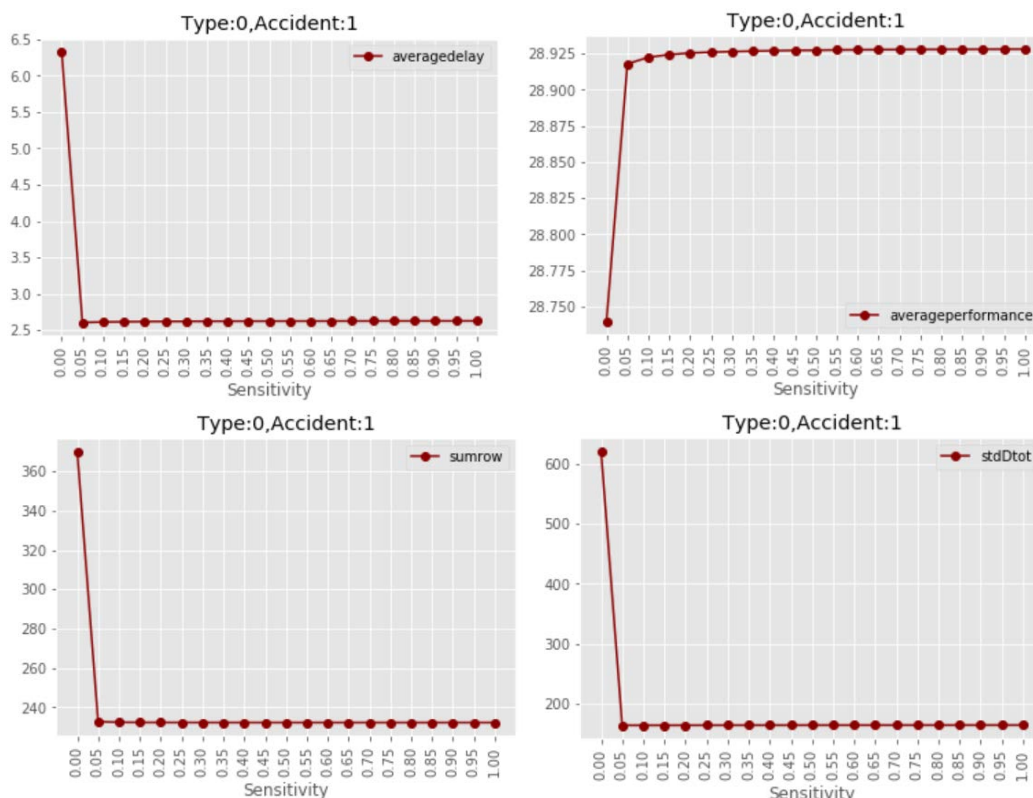
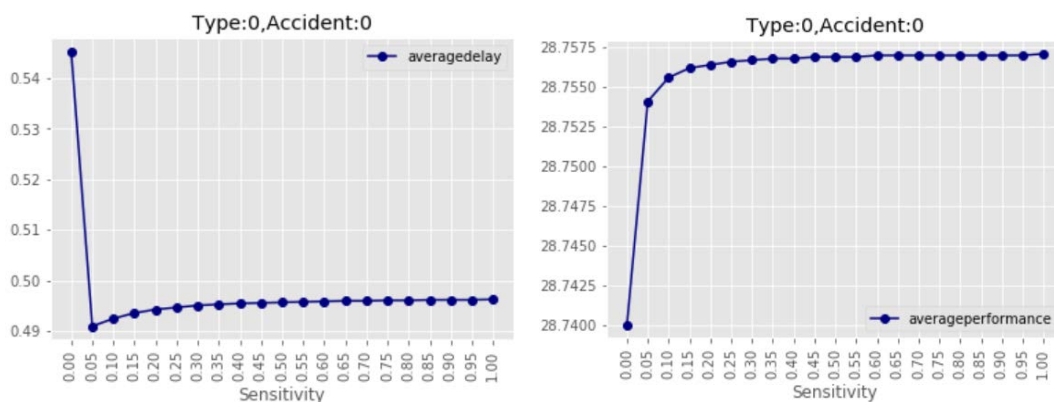


图 27 预测控制、事件发生时总体指标随敏感度变化图

- 在预测控制、有事件发生时，敏感度为 0 与敏感度大于 0 有着本质差异。而敏感度大于 0 范围内，总指标随敏感度变化几乎没有变化。这是由于预测控制提供的信息较为准确，能够引导驾驶员做出更为正确的决定。可将[0-0.05]区间数据加密，从而寻找到变化较快的区间。





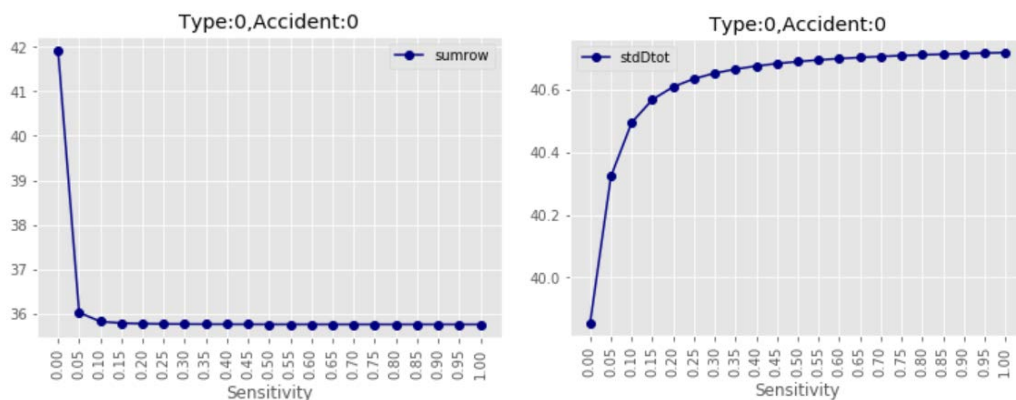


图 28 预测控制、无事件发生时总体指标随敏感度变化图

- 在预测控制、无事件发生时，敏感度为 0 与敏感度大于 0 同样有着较大差异，而随着敏感度的增加，四项指标均处于较为稳定的水平，因此在预测控制状态下，若敏感度达到 5%即可达到较好的效果。

接下来，通过绘制四种状态下，同一指标随敏感度变化的变化状况，可以比较不同状态间的差异性。这种差异性体现在预测控制效果整体由于反馈控制，事件的发生会对系统带来影响，阻碍系统的正常运行，但通过 VMS 等信息手段提供可靠性高的信息，能够减弱这种影响，并使系统尽快回复正常。

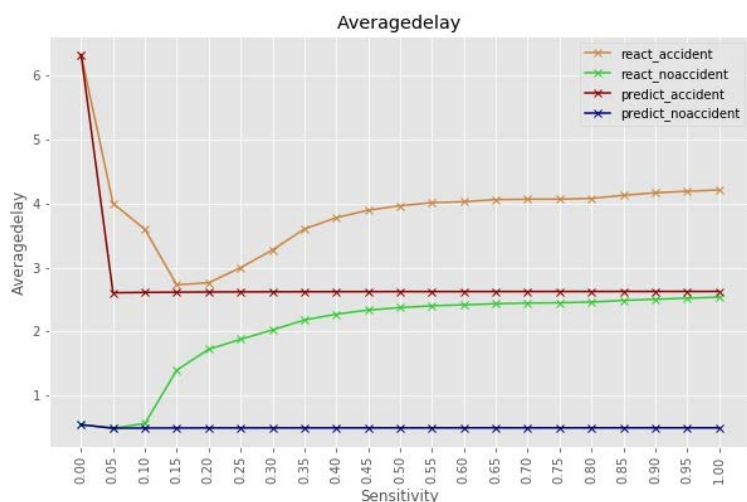


图 29 四种状态下平均延误随敏感度变化图

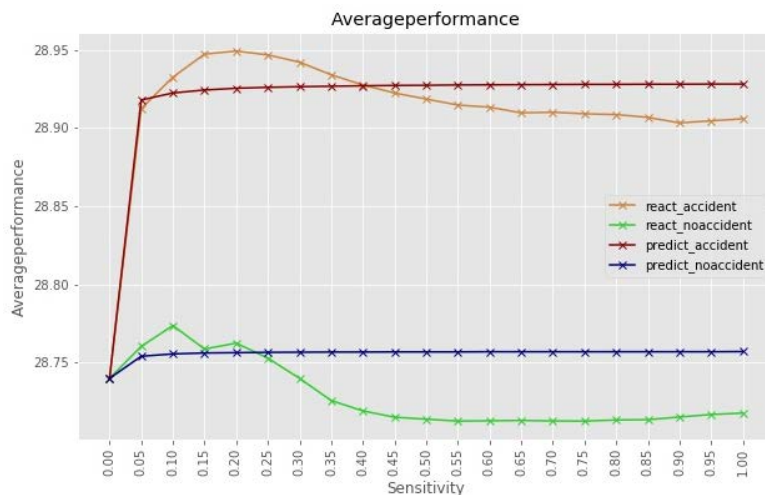


图 30 四种状态下平均性能随敏感度变化图

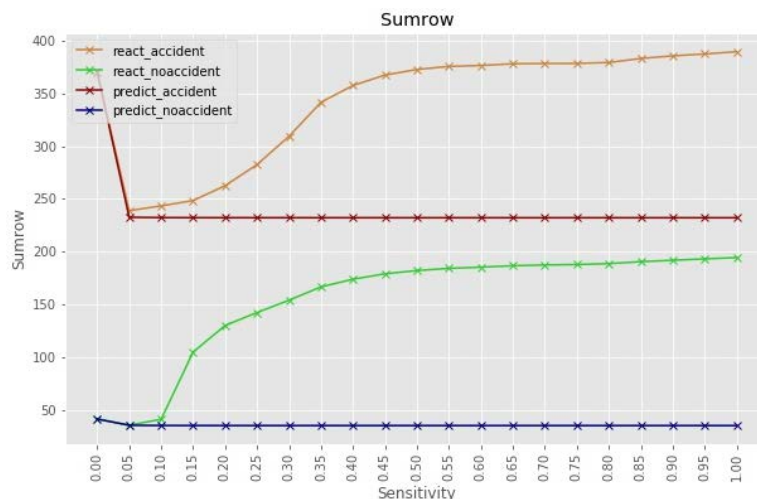


图 31 四种状态下总排队车辆数随敏感度变化图

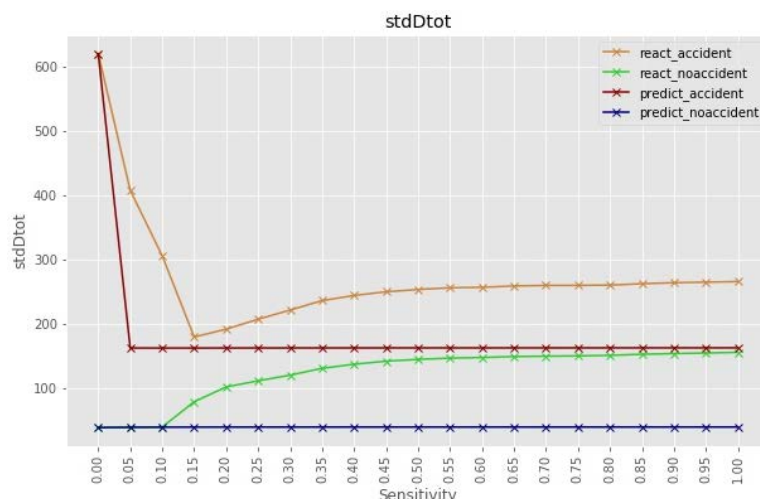


图 32 四种状态下总延误标准差随敏感度变化图

## 五、心得体会

通过完成《VMS 可变信息板对交通系统的影响分析》课程设计，我有很多收获。首先，在进行文献综述的过程中，通过文献查阅，我对于先进的交通信息服务系统有了更加深刻的认识，看到了近年来相关领域的发展成果。

之后，编程实现模型的过程，加深了我对模型的认识与理解，搞清模型如何迭代并不是一件容易的事情，加之模型在时间维度上构建，稍有不慎就会造成偏差，造成“使用了初始化值带入模型计算”的错误。由于数据间相互依赖，错误的检查也变得有些繁琐。通过实践，我认为，模型在时间维度上的递推方法尤为重要，与之相关的一系列细节不能有疏漏。在错误发生时，需要通过理解模型本身去有逻辑地排查错误，这样可以大大提高效率。使用程序得到的数据与 Excel 计算得到的数据存在较小偏差，可能是由于计算精度不同造成这一现象，对于分析没有影响。

此外，通过数据分析的过程我也看到了自己在数据分析方面存在的不足。可视化过

程完成后，如何将可视化结果反映出的规律进行有逻辑的表达十分重要，未来我还需要在这方面有所提高。

## 六、参考文献

- [1] 江筱薇. VMS 影响下驾驶员路径选择机理及信息发布策略研究[D].东南大学,2017.
- [2] LEE C. Developing Driver Compliance Based Operations Model for ATIS Applications[D]. Madison, Wisconsin: The University of Wisconsin-Madison, 2004.
- [3] 李志纯,黄海军.先进的旅行者信息系统对出行者选择行为的影响研究[J].公路交通科技,2005(02):95-99.
- [4] Yang H. Multiple equilibrium behaviors and advanced traveler information systems with endogenous market penetration[J]. Trans Res, 1998, 32(3):205-218.
- [5] 孙棧. VMS 对驾驶员路径选择行为的影响研究及效果分析[D].北京交通大学,2017.

## 附录

## 附录 1: 实现程序

```

# -*- coding: utf-8 -*-
"""
Created on Fri Apr 27 23:19:15 2018

@author: lenovo
"""

##模块导入
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
#####基础数据#####
results=[]
T = 552 #仿真时长
basic_i = 2180 #基础交通流量 veh/h
basic_P1 = 2491 #基础外部交通需求量 P1veh/h
basic_P2 = 3354 #基础外部交通需求量 P2veh/h
basic_C1 = 5232 #基础路径 1 通行能力 veh/h
basic_C2 = 5232 #基础路径 2 通行能力 veh/h
TT1 = 18 #路径 1 自由流时间 min
TT2 = 16 #路径 2 自由流时间 min
basic_L1 = 30 #路径 1 长度 km
basic_L2 = 27 #路径 2 长度 km
L_car = 0.005 #车辆平均长度 km
B0 = 0.58 #缺省路径 1 选择比例
TTacc = 20 #事件持续时长 min
acc_start = 150
#####模型数据设置#####Question:两次循环逻辑能否解释
##反馈控制的情况

def reactTrue():
    Dtot=np.ones(T)
    for k in range(T):
        if maxTT1==1:
            if k<=TT2:
                r1[k] = 0
                r2[k] = 0
            elif (k<=TT1)&(k>TT2):
                r1[k]=0
                r2[k]=max(0,(r2[k-1]+(i2[k-TT2-1]+P2[k-1]-C2[k-1])/60))
            else:
                r1[k] = max(0,(r1[k-1]+(i1[k-TT1-1]+P1[k-1]-C1[k-1])/60))
                r2[k] = max(0,(r2[k-1]+(i2[k-TT2-1]+P2[k-1]-C2[k-1])/60))
            D1[k] = 60*r1[k]/C1[k]
            D2[k] = 60*r2[k]/C2[k]
            sp[k] = min(1,max(0,(B0-B*(D1[k]-D2[k])))
            i1[k] = ii[k]*sp[k]
            i2[k] = ii[k] - i1[k]
        elif maxTT1==0:
            if k<=TT1:

```



```

        r1[k] = 0
        r2[k] = 0
    elif (k<=TT2)&(k>TT1):
        r1[k]=max(0,(r1[k-1]+(i1[k-TT1-1]+P1[k-1]-C1[k-1])/60))
        r2[k]=0
    else:
        r1[k] = max(0,(r1[k-1]+(i1[k-TT1-1]+P1[k-1]-C1[k-1])/60))
        r2[k] = max(0,(r2[k-1]+(i2[k-TT2-1]+P2[k-1]-C2[k-1])/60))
    D1[k] = 60*r1[k]/C1[k]
    D2[k] = 60*r2[k]/C2[k]
    sp[k] = min(1,max(0,(B0-B*(D1[k]-D2[k])))
    i1[k] = ii[k]*sp[k]
    i2[k] = ii[k] - i1[k]
else:
    pass
L1 = L_car*r1
L2 = L_car*r2
Ptot=basic_L1*i1+basic_L2*i2
for k in range(T):
    if maxTT1==1:
        if k<(T-TT1):
            Dtot[k] = i1[k]*D1[k+TT1]+i2[k]*D2[k+TT2]
        elif (k>=T-TT1)&(k<T-TT2):
            Dtot[k] = i1[k]*0+i2[k]*D2[k+TT2]
        else:
            Dtot[k] = 0
    elif maxTT1==0:
        if k<T-TT2:
            Dtot[k]=i1[k]*D1[k+TT1]+i2[k]*D2[k+TT2]
        elif (k>=T-TT2)&(k<T-TT1):
            Dtot[k]=i1[k]*D1[k+TT1]+i2[k]*0
        else:
            Dtot[k]=0

#输出指标
averagedelay=np.sum(Dtot)/np.sum(ii)
averageperformance=np.sum(Ptot)/np.sum(ii)
sumdelay=np.sum(D1)+np.sum(D2)
sumrow=(np.sum(r1)+np.sum(r2))/T
stdD1=np.std(D1)
stdD2=np.std(D2)
stdDtot=np.std(Dtot)/60
result = [react,B,acc,averagedelay,averageperformance,sumdelay,sumrow,stdD1,stdD2,stdDtot]
results.append(result)

def drawpicture():
    if react==1:
        mode='react control'
    elif react==0:
        mode='predict control'
    else:
        pass
    plt.figure(figsize=(15,9))
    plt.style.use("seaborn")
    plt.suptitle("Type:%s,                                     Sensitivity:%.2f,
    Accident:%d%(mode,B,acc),fontSize=20,backgroundColor='gainsboro',alpha=0.8)
    plt.subplot(2,2,1)

```

```

l1, = plt.plot(r1,lw=2.5)
l2, = plt.plot(r2,lw=2.5)
mmax=max(max(r1),max(r2))
plt.xlim(0,400)
plt.ylim(0,mmax+0.1*mmax)
plt.xlabel('Time')
plt.ylabel('Queue length')
plt.legend([l1, l2], ['r1', 'r2'])
plt.subplot(2,2,2)
l1,=plt.plot(D1,lw=2.5)
l2,=plt.plot(D2,lw=2.5)
mmax=max(max(D1),max(D2))
plt.xlim(0,400)
plt.ylim(0,mmax+0.1*mmax)
plt.xlabel('Time')
plt.ylabel('Delay(min)')
plt.legend([l1, l2], ['D1', 'D2'])
plt.subplot(2,2,3)
l1,=plt.plot(ii,lw=2.5)
l2,=plt.plot(i1,lw=2.5)
l3,=plt.plot(i2,lw=2.5)
plt.xlim(0,400)
plt.ylim(0,6000)
plt.xlabel('Time')
plt.ylabel('Demand(veh/hr)')
plt.legend([l1, l2,l3], ['i', 'i1','i2'])
plt.subplot(2,2,4)
l1,=plt.plot(sp,lw=2.5)
plt.xlim(0,400)
plt.ylim(0,1)
plt.xlabel('Time')
plt.ylabel('Probability to choice lane1')
plt.legend([l1],[ 'sp'])
plt.savefig("T%s_A%d_S%.2f.jpg"%(mode,acc,B))

```

```
def reactFalse():
```

```

    Dtot=np.ones(T)
    for k in range(T):
        j = k - (TT1-TT2)
        if maxTT1==1:
            if j<0:
                r1[k] = 0
                D1[k] = 60*r1[k]/C1[k]
            elif (j>=0)&(k<=TT1):
                r1[k]=0
                r2[j]=0
                D1[k] = 60*r1[k]/C1[k]
                D2[j] = 60*r2[j]/C2[j]
            else:
                r1[k] = max(0,(r1[k-1]+(i1[k-TT1-1]+P1[k-1]-C1[k-1])/60))
                r2[j] = max(0,(r2[j-1]+(i2[j-TT2-1]+P2[j-1]-C2[j-1])/60))
                D1[k] = 60*r1[k]/C1[k]
                D2[j] = 60*r2[j]/C2[j]
        if acc==0:
            if k>TT1:
                sp[k-TT1] = min(1,max(0,(B0-B*(D1[k]-D2[j])))
            elif acc==1:

```

```

    if (k>TT1)&(k<acc_start+TT1):
        sp[k-TT1] = min(1,max(0,(B0-B*(D1[k]-D2[j])))
    elif (k>=acc_start+TT1)&(k<acc_start+TTacc+TT1):
        sp[k-TT1] = min(1,max(0,(B0-B*(D1[k]-(D2[j]+acc*TTacc))))
    elif k>=acc_start+TTacc+TT1:
        sp[k-TT1] = min(1,max(0,(B0-B*(D1[k]-D2[j])))
    else:
        pass
    i1[k-TT1] = ii[k-TT1]*sp[k-TT1]
    i2[j-TT2] = ii[k-TT1] - i1[k-TT1]
elif maxTT1==0:
    pass
else:
    pass
L1 = L_car*r1
L2 = L_car*r2
Ptot=basic_L1*i1+basic_L2*i2
for k in range(T):
    if maxTT1==1:
        if k<(T-TT1):
            Dtot[k] = i1[k]*D1[k+TT1]+i2[k]*D2[k+TT2]
        elif (k>=T-TT1)&(k<T-TT2):
            Dtot[k] = i1[k]*0+i2[k]*D2[k+TT2]
        else:
            Dtot[k] = 0
    elif maxTT1==0:
        if k<T-TT2:
            Dtot[k]=i1[k]*D1[k+TT1]+i2[k]*D2[k+TT2]
        elif (k>=T-TT2)&(k<T-TT1):
            Dtot[k]=i1[k]*D1[k+TT1]+i2[k]*0
        else:
            Dtot[k]=0
#输出指标
averagedelay=np.sum(Dtot)/np.sum(ii)
averageperformance=np.sum(Ptot)/np.sum(ii)
sumdelay=np.sum(D1)+np.sum(D2)
sumrow=(np.sum(r1)+np.sum(r2))/T
stdD1=np.std(D1)
stdD2=np.std(D2)
stdDtot=np.std(Dtot)/60
result = [react,B,acc,averagedelay,averageperformance,sumdelay,sumrow,stdD1,stdD2,stdDtot]
results.append(result)

```

```

react=1#反馈控制
Blist=np.linspace(0,1,21)
for B in Blist:
    for acc in [0,1]:
        ii = np.zeros(T)#交通需求
        C1 = np.zeros(T)#路径 1 通行能力
        C2 = np.zeros(T)#路径 1 通行能力
        P1 = np.zeros(T)#路径 1 输入流量
        P2 = np.zeros(T)#路径 2 输入流量
        i1 = np.zeros(T)#路径 1 实际需求
        i2 = np.zeros(T)#路径 2 实际需求

```

```

sp = np.ones(T)*B0#选择路径 1 百分比
D1 = np.zeros(T)#路径 1 延误
D2 = np.zeros(T)#路径 2 延误
r1 = np.zeros(T)#路径 1 排队车辆数
r2 = np.zeros(T)#路径 2 排队车辆数
L1 = np.zeros(T)#路径 1 排队长度
L2 = np.zeros(T)#路径 2 排队长度
#Dtot = np.zeros(T)#总延误
Ptot = np.zeros(T)#总系统性能
#####数据设置#####
#P1,P2#
P1[:] = basic_P1
P2[:] = basic_P2
#C1,C2#
C1[:] = basic_C1
for i in range(T):
    if (i>=acc_start)&(i<acc_start+TTacc):
        C2[i] = basic_C2*(1-0.9*acc)
    else:
        C2[i] = basic_C2
##
for i in range(T):
    if i<60:
        ii[i] = basic_i
    elif (i>=60)&(i<100):
        ii[i] = C1[i]
    elif (i>=100)&(i<200):
        ii[i] = 0.8*C1[i]
    elif (i>=200)&(i<300):
        ii[i] = basic_i+(0.7*C1[i]-basic_i)*(0.05*C1[i]-i)/100
    else:
        ii[i] = 0.1*basic_i
if TT1>=TT2:
    maxTT1=1
else:
    maxTT1=0
react=True()
drawpicture()
writer = pd.ExcelWriter('output.xlsx')
results1=pd.DataFrame(results)

results1.columns=["Type",'Sensitivity','Accident','averagedelay','averageperformance','sumdelay','sumrow','stdD1','stdD2','stdDtot']
results1.to_excel(writer,'Sheet1')
writer.save()

Blist=np.linspace(0,1,21)
react=0#预测控制
for B in Blist:
    for acc in [0,1]:
        ii = np.zeros(T)#交通需求
        C1 = np.zeros(T)#路径 1 通行能力
        C2 = np.zeros(T)#路径 1 通行能力
        P1 = np.zeros(T)#路径 1 输入流量
        P2 = np.zeros(T)#路径 2 输入流量

```

```

i1 = np.zeros(T)#路径 1 实际需求
i2 = np.zeros(T)#路径 2 实际需求
sp = np.ones(T)*B0#选择路径 1 百分比
D1 = np.zeros(T)#路径 1 延误
D2 = np.zeros(T)#路径 2 延误
r1 = np.zeros(T)#路径 1 排队车辆数
r2 = np.zeros(T)#路径 2 排队车辆数
L1 = np.zeros(T)#路径 1 排队长度
L2 = np.zeros(T)#路劲 2 排队长度
#Dtot = np.zeros(T)#总延误
Ptot = np.zeros(T)#总系统性能
#####数据设置#####
#P1,P2#
P1[:] = basic_P1
P2[:] = basic_P2
#C1,C2#
C1[:] = basic_C1
for i in range(T):
    if (i>=acc_start)&(i<acc_start+TTacc):
        C2[i] = basic_C2*(1-0.9*acc)
    else:
        C2[i] = basic_C2
#i#
for i in range(T):
    if i<60:
        ii[i] = basic_i
    elif (i>=60)&(i<100):
        ii[i] = C1[i]
    elif (i>=100)&(i<200):
        ii[i] = 0.8*C1[i]
    elif (i>=200)&(i<300):
        ii[i] = basic_i+(0.7*C1[i]-basic_i)*(0.05*C1[i]-i)/100
    else:
        ii[i] = 0.1*basic_i
if TT1>=TT2:
    maxTT1=1
else:
    maxTT1=0
reactFalse()
drawpicture()
writer = pd.ExcelWriter('output2.xlsx')
results1=pd.DataFrame(results)
results1.columns=['Type','Sensitivity','Accident','averagedelay','averageperformance','sumdelay','sumrow','stdD1','stdD2','stdDtot']
results1.to_excel(writer,'Sheet1')
writer.save()

```

## 附录 2：评价结果表格

Ty pe	Sensi tivity	A cc	avgdel ay	avgperfor mance	sumdelay	sumrow	stdD1	stdD2	stdDtot
0	0	0	0.5452	28.7400	265.3964	41.9249	0.4593	0.6503	39.8555
0	0.05	0	0.4910	28.7541	228.0053	36.0182	0.4970	0.5616	40.3227
0	0.1	0	0.4925	28.7556	226.7274	35.8164	0.5086	0.5474	40.4942

0	0.15	0	0.4936	28.7562	226.4918	35.7791	0.5139	0.5416	40.5677
0	0.2	0	0.4943	28.7564	226.4086	35.7660	0.5169	0.5384	40.6083
0	0.25	0	0.4947	28.7566	226.3727	35.7603	0.5188	0.5363	40.6340
0	0.3	0	0.4951	28.7567	226.3528	35.7572	0.5202	0.5350	40.6518
0	0.35	0	0.4953	28.7568	226.3409	35.7553	0.5212	0.5340	40.6647
0	0.4	0	0.4955	28.7568	226.3317	35.7538	0.5219	0.5332	40.6746
0	0.45	0	0.4956	28.7569	226.3243	35.7527	0.5225	0.5326	40.6825
0	0.5	0	0.4957	28.7569	226.3193	35.7519	0.5230	0.5321	40.6888
0	0.55	0	0.4958	28.7569	226.3172	35.7516	0.5234	0.5317	40.6940
0	0.6	0	0.4959	28.7570	226.3154	35.7513	0.5237	0.5314	40.6983
0	0.65	0	0.4960	28.7570	226.3140	35.7510	0.5240	0.5311	40.7020
0	0.7	0	0.4960	28.7570	226.3138	35.7510	0.5242	0.5308	40.7051
0	0.75	0	0.4961	28.7570	226.3138	35.7510	0.5244	0.5306	40.7079
0	0.8	0	0.4961	28.7570	226.3138	35.7510	0.5246	0.5304	40.7103
0	0.85	0	0.4962	28.7570	226.3138	35.7510	0.5248	0.5302	40.7124
0	0.9	0	0.4962	28.7570	226.3138	35.7510	0.5249	0.5301	40.7143
0	0.95	0	0.4962	28.7570	226.3138	35.7510	0.5251	0.5300	40.7160
0	1	0	0.4963	28.7571	226.3138	35.7510	0.5252	0.5298	40.7174
0	0	1	6.3300	28.7400	4140.5476	369.9066	0.4593	21.3405	621.0791
0	0.05	1	2.6077	28.9178	2606.0294	232.7125	2.5549	13.1976	163.5765
0	0.1	1	2.6137	28.9223	2585.4983	232.4151	2.5206	13.0344	163.5207
0	0.15	1	2.6170	28.9242	2578.0884	232.3313	2.5113	12.9742	163.5358
0	0.2	1	2.6197	28.9253	2574.4599	232.3051	2.5109	12.9436	163.5784
0	0.25	1	2.6217	28.9259	2572.2952	232.2914	2.5124	12.9250	163.6244
0	0.3	1	2.6230	28.9263	2570.8506	232.2821	2.5137	12.9126	163.6506
0	0.35	1	2.6241	28.9266	2569.8182	232.2753	2.5153	12.9037	163.6854
0	0.4	1	2.6250	28.9269	2569.0602	232.2728	2.5166	12.8970	163.7115
0	0.45	1	2.6257	28.9271	2568.4706	232.2709	2.5176	12.8918	163.7289
0	0.5	1	2.6261	28.9272	2567.9988	232.2693	2.5184	12.8876	163.7396
0	0.55	1	2.6265	28.9274	2567.6128	232.2680	2.5191	12.8842	163.7447
0	0.6	1	2.6268	28.9275	2567.2911	232.2669	2.5196	12.8814	163.7447
0	0.65	1	2.6272	28.9276	2567.0189	232.2660	2.5203	12.8790	163.7600
0	0.7	1	2.6275	28.9277	2566.7855	232.2652	2.5209	12.8769	163.7737
0	0.75	1	2.6278	28.9278	2566.5833	232.2646	2.5214	12.8751	163.7857
0	0.8	1	2.6281	28.9278	2566.4099	232.2645	2.5219	12.8735	163.7960
0	0.85	1	2.6283	28.9279	2566.2571	232.2645	2.5223	12.8721	163.8051
0	0.9	1	2.6285	28.9279	2566.1212	232.2645	2.5226	12.8709	163.8132
0	0.95	1	2.6287	28.9280	2565.9997	232.2645	2.5229	12.8697	163.8204
0	1	1	2.6289	28.9280	2565.8903	232.2645	2.5232	12.8687	163.8269
1	0	0	0.5452	28.7400	265.3964	41.9249	0.4593	0.6503	39.8555
1	0.05	0	0.4881	28.7604	228.5935	36.1112	0.5034	0.5711	40.4066
1	0.1	0	0.5652	28.7737	263.9021	41.6889	0.5374	0.5881	40.6989
1	0.15	0	1.3998	28.7587	666.7306	105.3241	1.1365	1.4799	80.0262
1	0.2	0	1.7239	28.7625	825.2451	130.3648	1.4421	1.8188	102.9552

1	0.25	0	1.8806	28.7529	902.7932	142.6152	1.5652	1.9664	112.5670
1	0.3	0	2.0267	28.7400	976.8113	154.3079	1.6465	2.1323	121.3095
1	0.35	0	2.1821	28.7258	1056.5754	166.9083	1.7259	2.3258	132.0271
1	0.4	0	2.2730	28.7192	1102.4624	174.1571	1.7827	2.4300	138.3318
1	0.45	0	2.3378	28.7152	1134.8075	179.2667	1.8278	2.4992	143.0522
1	0.5	0	2.3758	28.7139	1154.0876	182.3124	1.8548	2.5419	145.9018
1	0.55	0	2.4039	28.7127	1167.2632	184.3938	1.8825	2.5623	147.8029
1	0.6	0	2.4190	28.7129	1174.3865	185.5190	1.8974	2.5735	148.8004
1	0.65	0	2.4364	28.7131	1183.1852	186.9090	1.9108	2.5945	150.2786
1	0.7	0	2.4465	28.7128	1187.7141	187.6244	1.9231	2.5995	150.8468
1	0.75	0	2.4517	28.7127	1190.1950	188.0163	1.9288	2.6056	151.2786
1	0.8	0	2.4625	28.7134	1195.6273	188.8745	1.9361	2.6182	152.1353
1	0.85	0	2.4885	28.7136	1207.5145	190.7523	1.9583	2.6346	153.8111
1	0.9	0	2.5065	28.7153	1215.8294	192.0658	1.9751	2.6427	154.8697
1	0.95	0	2.5234	28.7169	1224.0103	193.3581	1.9856	2.6581	155.7517
1	1	0	2.5395	28.7178	1231.8820	194.6016	1.9944	2.6718	156.7849
1	0	1	6.3300	28.7400	4140.5476	369.9066	0.4593	21.3405	621.0791
1	0.05	1	4.0061	28.9123	3218.0091	238.9725	2.4830	19.8877	408.7292
1	0.1	1	3.5985	28.9324	3212.2946	243.4449	2.5799	19.2309	306.9445
1	0.15	1	2.7318	28.9471	3042.4374	248.5039	2.7584	16.5557	180.6469
1	0.2	1	2.7645	28.9491	3010.5574	262.5681	2.9280	15.2677	192.8645
1	0.25	1	3.0001	28.9466	3181.9048	282.5896	3.1023	15.7098	208.2957
1	0.3	1	3.2766	28.9420	3496.2331	309.3052	3.2881	17.1037	222.5947
1	0.35	1	3.6055	28.9339	3893.3987	341.4741	3.4884	18.9794	237.2182
1	0.4	1	3.7819	28.9274	4063.1241	357.3219	3.5861	19.6619	245.2236
1	0.45	1	3.8989	28.9223	4153.7352	367.1937	3.6473	19.9430	250.9526
1	0.5	1	3.9679	28.9184	4190.9145	372.4512	3.6825	19.9859	254.4982
1	0.55	1	4.0159	28.9146	4193.2298	375.3433	3.7022	19.8356	257.1396
1	0.6	1	4.0312	28.9132	4192.2130	376.2187	3.7100	19.7736	257.9568
1	0.65	1	4.0644	28.9096	4183.3145	377.8544	3.7218	19.5915	260.0131
1	0.7	1	4.0706	28.9100	4176.6693	378.1411	3.7224	19.5123	260.8467
1	0.75	1	4.0717	28.9090	4176.0069	378.1798	3.7237	19.5041	260.9165
1	0.8	1	4.0824	28.9085	4183.8656	379.0741	3.7289	19.5260	261.3640
1	0.85	1	4.1312	28.9067	4214.1870	382.8450	3.7532	19.5916	263.5043
1	0.9	1	4.1712	28.9032	4223.9145	385.3287	3.7661	19.5360	265.0253
1	0.95	1	4.1937	28.9045	4236.7028	387.1131	3.7743	19.5522	265.9425
1	1	1	4.2154	28.9058	4263.7270	389.2607	3.7870	19.6827	266.8411