

机场的出租车问题研究

摘要

本文为分析机场出租车的相关问题，首先主要为出租车司机在机场的载客选择进行决策分析，给出合理收益更大的选择策略；结合实际数据为具体出租车司机决策，并分析决策模型的稳定性和相关因素的依赖性；接下来分析机场并行车道的最高乘车效率方案，最后为收益均衡设计短途载客“优先权”可行方案。

针对问题一和问题二，我们建立了基于**司机单位时间利益最高的决策模型**。首先对机场司机的两种选择进行**单位时间收益分析**，司机决策目标将是收益更高，决策相关因素有航班数量、机场的等待车数量和潜在的载客距离。收益求解分析中，我们通过航班数量将分析求解出租车的乘客量，其中考虑季节、时间段与乘车意愿对**乘客量**的影响；等待车数可对应分析出机场载客的等待时间；收益分析考虑堵车情况的影响；对于出租车的距离，我们考虑**曼哈顿距离**，使距离更加精准。潜在载客收益距离在机场和市中心均具有随机不确定性，我们使用平均距离作为衡量，最后基于理想期望收益，分析不同距离影响两种方案的收益结果，给出最优决策。

针对问题三，我们选用的矩阵式停靠方式，并建立了**乘车效率最大化模型**。我们首先是分析了上车点只有一个的情况，讨论了影响乘车效率的因素。考虑到乘客携带行李量的不同将导致上车时间差别很大，因此用**蒙特卡洛模拟**出乘客上车时间曲线，并用SPSS拟合得到更符合随机性的函数，发现乘客上车时间服从指数分布。最后综合其他因素，迭代1000次后得出每个车道设立8个泊车位时的乘车效率最大，此时每小时的通行量为672.2辆。在此基础上分析多个上车点的乘车效率，我们给每个上车点都分配一个或多个泊车位，并使这些上车点的最大拥有量最小。此时乘车区被划分为多个单独的上车点，问题由**多个乘车点转化为对单个乘车点的分析**，求出它们各自的最大行走时间与上车时间后，再相互比较得出最终的最大值。当上车点为8个时，每小时的通行量为750.8辆，增长率仅为0.00738，因此若乘客流量不大，可设立一个或两个上车点。

针对问题四，我们建立了以**不平衡度最小**为目标的规划模型，我们将不平衡度定义为短程期望收益与长途的期望收益的绝对差值，寻找最佳定义的短途的阈值距离，使得在设定的短途阈值内返回的出租车拥有下一次机场载客优先插队的“优先权”，以使得长短途收益差减小，同时考虑随着时间变化，达到最小不平衡度时的最佳阈值，根据仿真结果显示，在0点至9点不设置阈值，在9点至24点时，阈值设定值分别为14km时达到最佳效果。另外我们为了避免短途司机一直跑短途投机以获得更多收入，我们对于一天的“优先权”的享受次数为3次。

关键词：单位时间利益最高 曼哈顿距离 蒙特卡洛模拟 不平衡度

一、问题重述

1.1 问题背景

出租车是现在主流的交通工具之一，也是大多数飞机乘客到达机场后去往周边的一大途经。目前国际机场多数把送客与接客通道分离开来。而对于将人送至机场的司机而言，主要有前往到达区等待载客和直接回市中心载客的选择：

(1) 等待载客返回市区。出租车得按照“先来后到”的原则，进入指定的“蓄车池”等候载客，期间，司机等待时间长短与排队出租车和乘客的数量多少有关，等候将付出一定的时间成本。(2) 直接放空返回市区拉客。放空状态回市区将承担空载费用及损失潜在可能的载客收益。

对于司机做出选择前，可参考在某时间段抵达的航班数量和“蓄车池”里已有的车辆数。其中司机的个人经验判断也将影响决策，经验判断主要有某个季节或某时间段抵达航班数和可能航班中乘客数量的多少等方面。而对于乘车若选择打车，也需到达制定的“乘车区”排队，并按照先后顺序上车。

机场还有出租车管理人员，主要负责“分批定量”地管理出租车进入“乘车区”，同时调配合理数量的乘客上车。在生活中，也有其他诸多影响司机决策的确定和不确定因素，在实际中，还有很多影响出租车司机决策的确定和不确定因素，彼此关系各异，效果也有所区别。

1.2 问题重述

结合问题背景和生活实际，回答以下问题：

(1) 分析研究司机决策相关因素的影响情况，建立出租车司机的决策模型，其中可结合机场乘客数量的变化情况和出租车的利益情况的考虑，最后对司机的选择给出建议。

(2) 统计国内某一机场和所在城市出租车的所有相关信息，给出对应司机选择的方案，接下来分析决策模型的合理性，并研究相关因素的依赖情况。

(3) 机场常会发生出租车排队载客和乘客等待坐车的现象。现要对机场的“乘车区”有两条并行道路的最优“上车点”进行设置，保证乘客、车辆安全的前提下，使得乘车效率最高，即使用时间最短。

(4) 乘客的需求行驶里程将会影响机场司机的载客收益。机场出发的短途载客司机将收益低，在出租车司机不能选择乘客、拒载和允许多次往返的前提下，为了使得出租车的收益尽可能均衡，设计研究合理的“优先权”安排方案。

二、问题分析

2.1 问题一、问题二的分析

由于司机可观测的只有当前时刻抵达的航班数量，蓄车池中等待的汽车数。但是司机可以凭借当时的时间，天气情况判断当前时刻下的航班数会有多少人会来乘车，判断出“蓄车池”中对应的等待时间，然后我们根据留在“蓄车池”的预期收益与去市区的预期收益在单位时间内的比较分析，选择更高收益的决策，最后我们用模型得出的司机

决策与真实数据进行对比来检验模型的精准性。

2.2 问题三的分析

对于问题三，题目要求乘车效率最高，即单位时间的通行能力最大，而通行能力与上车点的数量与每批次搭载乘客离开的时长有关。一批次搭载的时长与乘客行走时长和上车时长有关，因行李量等因素是一个未知数，所以上车时长是一个随机量。先从一个上车点出发，确定上车区最佳的可停泊车数。在此基础上分析多个上车点的最大通行效率。

2.3 问题四的分析

问题四是要平衡长途载客司机和短途载客司机的单位时间收益。长途载客司机单次利润高，短途载客司机的单次利润低，但是短途载客司机可以获得一个优先时间，相当于短途载客司机的单次载客时长缩短，单位时间收益增高。我们的目标是较好的平衡两者的收益，同时司机的预期收益随着时间变化而改变，因此设置合理的定义长短途距离的阈值是我们的突破点。

三、问题假设与符号说明

3.1 问题假设

1. 假设每个时段的航班集中到达且需乘车人一致前往乘车区等候；

1. 假设所有的出租车单位耗油一致，忽略载客对于耗油的影响。
2. 假设出租车从机场去往市中心无法新载客。
3. 不考虑极端恶劣天气的影响。
4. 不考虑节假日对于人们出租车选择意愿的影响。
5. 忽略时间变化对载客量的影响。

3.2 符号说明

表 3-1 符号说明

符号	说明
h_i	一天中到达航班的情况
H_t	当前第 t 小时已到达总航班数
N_t	第 t 时间段的出租车需求辆数
T_m	各时刻速度
C_m	成本
Q	收费
T_{wt}	蓄车池等待时间
S_{AB}	曼哈顿距离

注:其它符号或重复符号则以文中具体出处为准。

四、出租车司机决策模型的建立与求解

4.1 模型准备

由题可知，送客到机场的出租车司机决策的目标是为了获得更高的收益，而影响司机决策的方向主要有当时“蓄车池”的出租车数量、抵达航班数量以及乘客的乘车距离。因决策的方向容易受季节、人群意愿影响，因此，为了给出租车司机选择决策做出有效建议，我们将对航班数量、乘客情况及出租车司机的收益等方面进行模型准备分析。

4.1.1 航班数量的分析

由实际可知，不同季节、时段下，航班有淡季与旺季的区分，所以航班的数量也会因季节与时段不同而有所调整。因此，我们认为季节对于航班情况的影响可通过航班数量的变化体现，即季节对乘客数量多少的影响转化为航班数量的调整。

而对于一天内各个时段分析，航班到达的时间在各个时段及时段内可能不同，其对应数量也有所区别。为了便于之后各个时段航班到达数量的统计，根据**假设一**，我们以每小时分析，在该时段内到达的航班，我们将航班集中处理，认为其统一到达且出租车的乘客均统一该时间段前往乘车区。

由上，对于当前到达航班数，我们表示为：

$$h_i = \begin{cases} 1, & \text{航班已到达} \\ 0, & \text{航班未到达} \end{cases}, i = 1, 2, \dots, n \quad (4.1)$$

$$H_t = \sum_{j=60*(t-1)}^{59+60*(t-1)} h_i, t = 1, \dots, 24 \quad (4.2)$$

$$h_t = H_t - H_{t-1} \quad (4.3)$$

式中， h_i 为一天中到达航班的情况； i 为航班到达的情况； H_t 为当前第 t 小时已到达总航班数； j 为某一分钟； h_t 为当前第 t 小时达到的航班数。

4.1.2 乘坐出租车的数量分析

航班中的乘客下飞机后的机场将有各种交通方式选择，如私家车、公交车、单车、出租车和地铁等。而这些交通方式都在一定程度上影响乘客选择出租车的意愿。

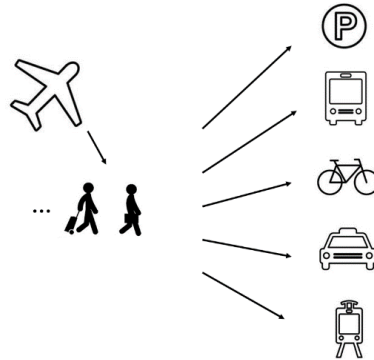


图 4-1 机场出行方式选择

由于大部分公共交通工具具有时间限制，到晚上特定时间将停止提供，设供应时间为早上 t_1 -晚上 t_2 ，晚上因此我们假设公共交通运营阶段飞机乘客选择出租车的意愿为

α ，晚上无公共交通运营阶段的选择意愿为 β ，所以基于其他出行方式影响下预计不同时间段下愿意乘坐出租车的乘客数量为：

$$p_t = \begin{cases} Z \cdot h_t \cdot S \cdot \alpha, & t_1 \leq t \leq t_2 \\ Z \cdot h_t \cdot S \cdot \beta, & 0 \leq t < t_1, t_2 < t \leq 23 \end{cases} \quad (4.4)$$

其中， h_t 表示航班数量； S 表示每次航班的人数； Z 表示航班的客座率； t_1 为公共交通开始运行时刻； t_2 为公共交通停运时刻； t 为一天中对应的第 t 时间段。不仅其他交通方式会影响飞机乘客的乘坐出租车的意愿，天气因素也会对产生影响。据调查^[1]，天气晴朗的情况下，人们对于出租车选择意愿不高，为 γ ，恶劣天气下，人们对于出租车选择意愿较高，为 μ 。所以，基于其他出行方式和天气影响后，某时段最终选择出租车的人数 P_t 为：

$$P_t = \begin{cases} p_t \cdot \gamma, & \text{天气良好} \\ p_t \cdot \mu, & \text{天气恶劣} \end{cases} \quad (4.5)$$

考虑到出租车计费标准是按照车次，而非人数收费，所以单辆出租车平均载客量是衡量某时间段到达乘客与所需出租车辆数的关键。我们查询得上海出租车得平均载客量为 1.25 人/次^[2]，用 P_{ave} 表示。因此，计算某时段对应航班数后得出租车需求辆数为：

$$N_t = \frac{P_t}{P_{ave}} \quad (4.6)$$

式中， N_t 表示第 t 时间段的出租车需求辆数； t 为一天中对应的第 t 时间段。

4.1.3 机场出发速度随着时间的变化分析

由于城市的交通状况容易堵车，回市区途中的通行速度将随着时间变化，因此各个时段下出租车选择回市区或由机场载客出发的速度将不同。我们预计将通过出行数据分析各时段的车速。我们定义各个时段对应的车速为 $v_m, m = 0, 1, \dots, 23$ ，设出租车由机场出发的行驶距离为 L 千米，则该段距离下所需要的时间为：

$$T_m = \frac{L}{v_m} \quad (4.7)$$

4.1.4 出租车运行的成本分析

为求解出租车运行的收益，我们清楚收益=收费-成本。由出租车的运营方式，我们可知出租车的成本主要包括燃油费、租车费两部分，根据假设，忽略车辆故障的修理费。

接下来对成本的计算分析，假设出租承租用的费用为每小时 R 元，燃油成本为每千米 r 元，由机场出发运行 L 千米，其成本计算为：

$$C_m = L \cdot r + T_m \cdot R \quad (4.8)$$

如果由机场空架回到市区，设市区为 L_0 千米，则回去成本 C_0 则是将 L_0 代入式 4.8 可求解，代入式 4.7 得对应所需时间 T_0 。而对于在机场等待乘客的成本，其中主要包括“蓄车池”等待时间和在乘车区等待出发的时间。我们认为“蓄车池”中的出租车满足先来后到的等待原则，并根据 4.1.1 分析，当前第 t 时段航班到达，所对应的航班数为 H_t ，航班的乘客集中到达等待区等候上车，由下机到到达乘车区时长定义为 t_w ，这些航班乘客所需出租车数量为 N_t ，令目前“蓄车池”的出租车数量为 n_t ，每辆出租车在乘车区载客

所等待的时间平均为 t_v ，则当前出租车还需剩余出租车数量为：

$$M_t = \max \{|n_t - N_t|, -1\} \quad (4.9)$$

当 $M_t \geq 0$ ，则当前的出租车数量将超过航班所需，需要等待之后航班的到达才可能发车。这种状况下假设第 $t + \theta$ 时段对应的 $N_{t+1} + \dots + N_{t+\theta-1} < n_t - N_t < N_{t+1} + \dots + N_{t+\theta-1}$ ，则在 $t + \theta$ 时段当前已有的 n_t 辆出租车才完全实现载客，等待时长为：

$$T_{wt} = [(n_t - N_t) - (N_{t+1} + \dots + N_{t+\theta-1})] \cdot t_v + \theta + t_w \quad (4.10)$$

若 $M_t = -1$ ，则证明现阶段出租车数量不足以当前航班的需要，该情况下司机选择载客等待时间为等待前面 n 辆出租车的出发时长与 t_w ，即：

$$T_{wt} = n_t \cdot t_v + t_w \quad (4.11)$$

因此综合得到“蓄车池”中出租车的等待时间满足以下情况：

$$T_{wt} = \begin{cases} n_t \cdot t_v + t_w, & M_t = -1 \\ T_{wt} = [(n_t - N_t) - (N_{t+1} + \dots + N_{t+\theta-1})] \cdot t_v + \theta + t_w, & M_t \geq 0 \end{cases} \quad (4.12)$$

因为出租车由“蓄车池”开往乘车区的距离较短，所以我们忽略该行为下的燃油费和行驶时间，综合选择留在“蓄车池”的车在载客 L 千米对应成本为：

$$C_{wm} = L \cdot r + (T_m + T_{wt}) \cdot R \quad (4.13)$$

4.1.5 潜在载客收益和单位时间收益分析

根据市场普遍出租车定价标准^[3]，我们定义出租车的起步价为 a_0 元，距离为 b_0 千米，超过后的每千米计费为 a 元，深夜凌晨（23：00~5：00）期间加价 30%，若乘客 i 的需求距离为 b 千米，则白天实际收费如下，加价阶段收费为 $1.3Q$ ：

$$Q = \begin{cases} a_0, & b < b_0 \\ a_0 + (b - b_0)a, & b \geq b_0 \end{cases} \quad (4.14)$$

出租车选择回市中心载客和留在机场载客都对应有不同的潜在载客收益，选择空载到达市中心，因为出租车在市中心载客的路程是随机不确定的，有远、中、近程的可能且容易因时段原因而堵车，因此我们市中区预期收益将以全市的出租车司机的在某时段的平均载客距离 l_t 来评估，其方式对应单位时间收益如下，距离代入 l_t ：

$$W_1 = \frac{Q - C_m - C_0}{T_0 + T_m} \quad (4.15)$$

对于由机场载客出发的乘客同样和市中心乘客具有随机性，因此我们由历史机场载客数据的平均距离 l_{wt} 来评估，采取一致方式，单位时间总收益为：

$$W_2 = \frac{Q - C_{wm}}{T_{wt} + T_m} \quad (4.16)$$

综上所述，对于司机不同选择下，我们的基于单位时间收益最大的决策模型方式如下：

$$\begin{aligned}
& \max\{W_1, W_2\} \\
& \text{s.t.} \left\{ \begin{aligned}
& h_i = \begin{cases} 1, \text{航班已到达} \\ 0, \text{航班未到达} \end{cases}, i = 1, 2, \dots, n \\
& H_t = \sum_{j=60*(t-1)}^{59+60*(t-1)} h_i, t = 1, \dots, 24 \\
& h_t = H_t - H_{t-1} \\
& p_t = \begin{cases} h_t \cdot S \cdot \alpha, t_1 \leq t \leq t_2 \\ h_t \cdot S \cdot \beta, 0 \leq t < t_1, t_2 < t \leq 23 \end{cases} \\
& P_t = \begin{cases} p_t \cdot \gamma, \text{天气良好} \\ p_t \cdot \mu, \text{天气恶劣} \end{cases} \\
& N_t = \frac{P_t}{P_{ave}} \\
& T_m = \frac{L}{v_m} \\
& C_m = L \cdot r + T_m \cdot R \\
& C_{wm} = L \cdot r + (T_m + T_{wt}) \cdot R \\
& Q = \begin{cases} a_0, b < b_0 \\ a_0 + (b - b_0)a, b \geq b_0 \end{cases} \\
& W_1 = \frac{Q - C_m - C_0}{T_0 + T_m} \\
& W_2 = \frac{Q - C_{wm}}{T_{wt} + T_m}
\end{aligned} \right. \quad (4.17)
\end{aligned}$$

上式中， m 和 t 均为时段的表示，范围为 $[0, 23]$ 。

4.3 基于单位时间收益最高的决策模型的求解步骤

根据上模型所示，我们对于两种不同选择的单位时间收益进行比较，当 W_1 更大时，我们选择离开机场而前往市中心进行载客， W_2 更大时我们选择在机场等待载客。针对模型，我们整体实现求解的步骤如下：

Step1: 确定当前航班数、“蓄车池”的出租车数、当天时间段；

Step2: 对于乘车数量的分析，根据当前到达航班数，分析季节因素、时间段因素、乘客乘车意愿的影响，求解当前乘车人数；

Step3: 根据时间段信息分析城市不同时段下的时速；

Step4: 计算由机场出发后的出租车的成本和停留机场时段的时间成本；

Step5: 分析机场出发和市中心出发的潜在载客收益和时间；

Step6: 根据 Step1-6 求解两种方案下的单位时间收益；

Step7: 判断 W_1 和 W_2 大小，给定决策方案。

具体的流程图如下：

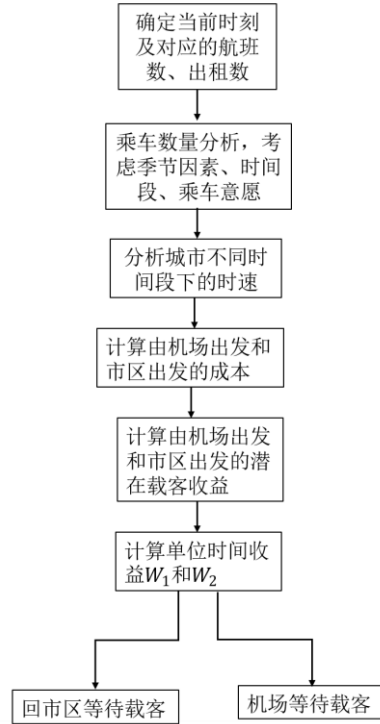


图 4-2 算法流程图

4.4 数据的选择与处理

对于机场航班数据的选择，我们通过全球航班飞行轨迹实时数据网^[4]获取 2019 年 9 月 15 日预计到达上海虹桥机场的数据。对于出租车数据的选择，我们查到了关于强生出租车在上海的轨迹数据^[5]，但由于缺少与上海虹桥机场航班数据相同时期的出租车数据，根据假设，我们认为据航班和出租车而言，普通时期数据变化不是很大，我们将其等效同一天分析，并且我们将航班数据离散化，以缩小数据的误差。

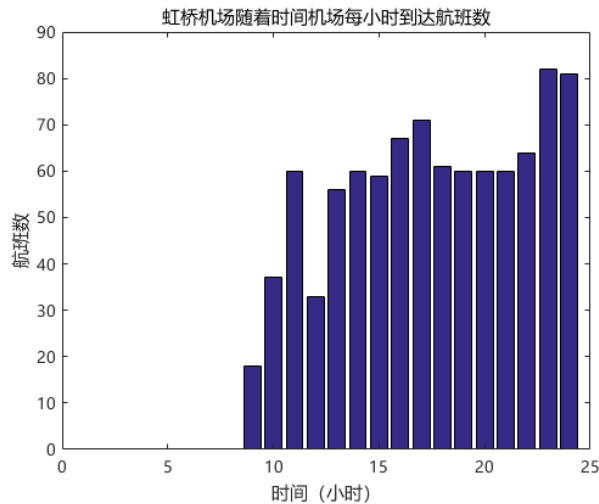


图 4-3 虹桥机场随时间的每小时到达航班数

4.4.1 出租车数据的清洗

我们根据 2018 年 4 月 1 日强生出租车公司的数据进行分析，由于该出租车公司所拥有的出租车的数量占上海市总出租车数量 25%^[6]。为更好的反映航班数，“蓄车池”中出租车等待数量，我们将该一辆出租车的权值增加为原来的 4 倍。

由于数据量庞大，观察下列一组数据，我们发现计算出出租车的载人距离，速度，只根据载人起始时刻的位置以及所对应的时刻，所以我们对其进行数据清洗，只保留出租车状态变换时刻的位置以及时间信息，车的 ID 编号。有部分司机数据为一天停运状态不属于我们需求范围。因此，我们基于以下几点对数据进行清洗：

1. 去除数据缺项、错误的数据；
2. 去除由空载到载客切换的定位不在上海市范围的数据；
3. 去除出租车状态一天无变化的数据。

最终我们得到的清洗后的数据如下：

表 4-1 强生出租车 ID25743 清洗后部分数据

ID	载客状态	当前时间	经度	纬度
25743	0	2018/4/1 14:01	121.351972	31.19678
.....				
25743	0	2018/4/1 14:07	121.354907	31.183082
25743	1	2018/4/1 14:07	121.355042	31.183083
.....				
25743	1	2018/4/1 14:13	121.361413	31.19213
25743	0	2018/4/1 14:13	121.360135	31.191597
.....				
25743	0	2018/4/1 14:26	121.330443	31.157462
25743	1	2018/4/1 14:26	121.330268	31.157365

4.4.2 出租车距离的曼哈顿距离求解

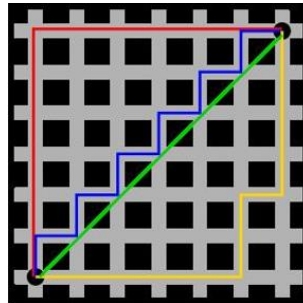


图 4-4 曼哈顿城市距离图示

对于如表 4-1 的强生出租车的数据中的经度和纬度的数据进行距离的求解，采用曼哈顿距离^[7]的方法，具体公式如下：

$$\bar{S}_{AB} = \sqrt{(1.1132 \times 10^5 \times |u_1 - u_2|)^2 + (10^5 \times |v_1 - v_2|)^2} \quad (4.18)$$

$$\hat{S}_{AB} = 2 \arcsin \left(\sqrt{\left(\frac{\sin(v_2 - v_1) \cdot \frac{\pi}{180}}{2} \right)^2 + \cos\left(v_1 \cdot \frac{\pi}{180}\right) \cos\left(v_2 \cdot \frac{\pi}{180}\right) \left(\sin \frac{\Delta \lambda}{2}\right)^2} \right) \cdot R \quad (4.19)$$

$$S_{AB} = \frac{\hat{S}_{AB}}{\bar{S}_{AB}} \cdot (1.1132 \times 10^5 \times |u_1 - u_2| + 10^5 \times |v_1 - v_2|) \quad (4.20)$$

其中， \bar{S}_{AB} 为A到B的直线距离；考虑地球是球体， \hat{S}_{AB} 是球体对应距离；城市交通很

多也走向为各方向正方向，所以实际两地间行驶的城市距离为 S_{AB} 。

4.4.3 乘车路程分布

我们找到虹桥机场对应的经纬度范围为 $(31.2152, 121.3065) \sim (31.1744, 121.3520)$ 的数据数为 9641 单次。数据筛选后在这个范围之内出发的出租车，我们认为由机场出发的，同时我们也抽样统计整个市区 9641 次出租车单次订单的数据。

通过式 4-20 计算出出租车的行驶距离，以 2km 为一个单位求落在该范围的车辆数，发现行驶距离满足正态分布，所以我们对其进行拟合，得到如下图：

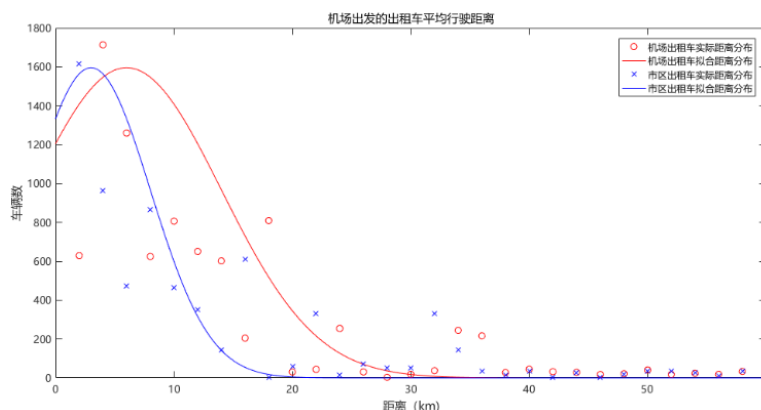


图 4-5 机场与市区出发的出租车平均行驶

拟合的乘车路程分布曲线满足：

$$x_{number} = h \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (4.21)$$

经过计算，其中对于从机场出发的路程满足 $h = 16234.21, \sigma = 8.65, \mu = 12.213$ ，拟合度为 77.78%。对于整个市区的路程 $h = 9568.3, \sigma = 3.4, \mu = 15.53$ ，拟合度为 75.28%，由此看出拟合度基本较好。

4.4.4 市区出发的速度随时间变化分析

根据从机场出发出租车前往目的地的距离以及所对应的时间，我们可以算得各次行驶行为的速度情况，累加平均，得到从出发的出租车随着时间变化的平均行驶速度，结果如下图所示：

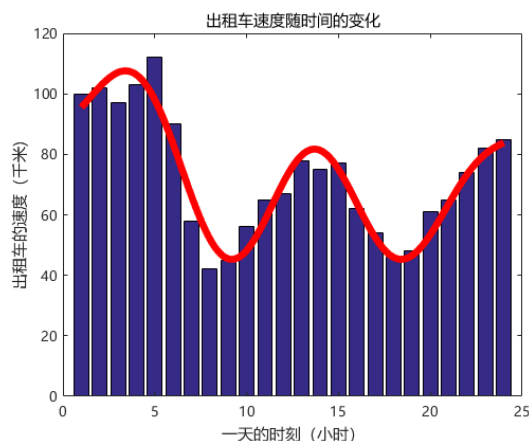


图 4-6 出租车机场出发的速度变化

为了使图 4-5 的结果在后面分析能够更好的反映各时刻的等待时间，将离散值连续化，我们对上面数据用三次傅里叶进行拟合，拟合度为 88.85%，拟合结果比较符合实际。

$$v_m = 73.05 + 16.49 \cos 0.2414x + 8.912 \sin 0.2414x - 11.37 \cos 0.4828x - 10.59 \sin 0.4828x - 11.48 \cos 0.7242x - 3.655 \sin 0.7242 \quad (4.22)$$

4.4.5 “蓄车池”当前车辆分析

由于缺少机场“蓄车池”内车辆数据，所以我们对于“蓄车池”中出租车的数量我们可以通过某一时刻，搜寻此时在虹桥机场经纬度范围内所有出租车数量，将其替代为“蓄车池”中等待的车辆数。

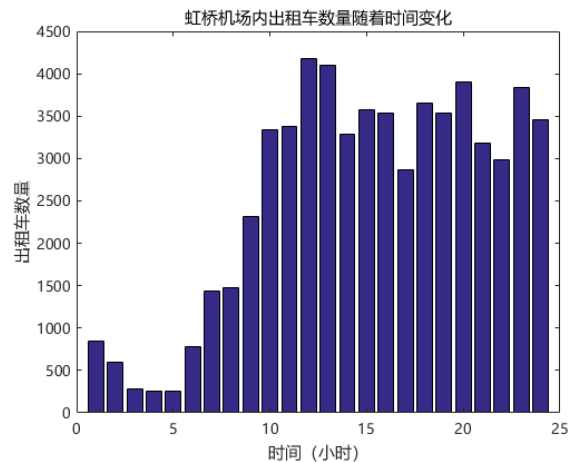


图 4-7 虹桥机场等待出租数量随时间变化

4.4.6 机场出租车等待时间分析

根据图 4-8 某时段到达虹桥航班数和机场所在的出租车数，预计算出租车的等待时间。由实际经验可知，国内各航空公司的航班出行的客座率均未能达到 100%，根据查询得 2017 年南方航空、中国国航和东方航空^[8]的客座率均达到 80.00% 以上，分别为 82.26%、81.22% 和 81.40%，因此我们将开往上海虹桥所有航班的客座率以三家航空均值来衡量，得客座率 $Z = 82.32\%$ 。同时通过查询可知中国民航客座数^[9]主要范围在 158~380 区间，因此我们定义每架民航座位数为 S 为 200 个进行以下的分析。

结合当前的航班数和出租车数，我们对出租车等待时间分析，得到出租车等待时间随时间变化图如下：

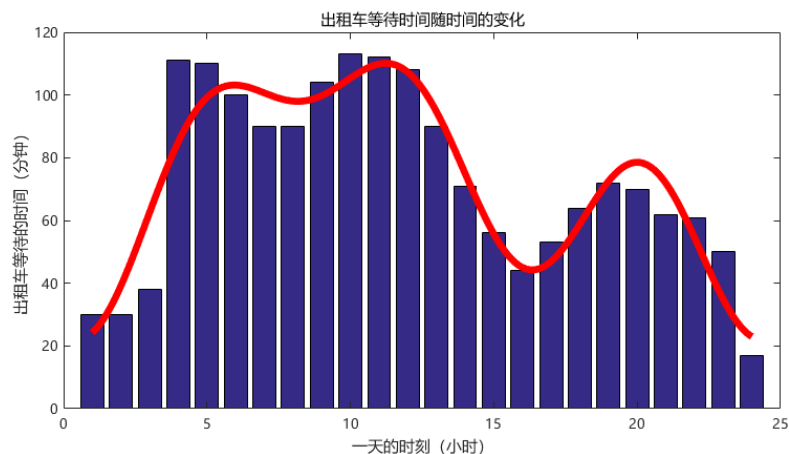


图 4-8 出租等待时间随时间变化

对以上结果进行拟合，得到的等待时间函数如下，最终的拟合度为 83.7%，拟合结果较优。

$$T_{wt} = 73.05 - 24.73 \cos 0.2633x + 16.97 \sin 0.2633x - 7.243 \cos 0.5266x - 13.41 \sin 0.5266x - 17.25 \cos 0.7899x - 5.911 \sin 0.7899x \quad (4.23)$$

4.4.7 单位时间收益的分析

查阅数据^[10]，我们得到普通出租车一般每公里的耗油成本 $r = 0.6$ 元/km，上海市每个月出租车租金为 7000 元，折合成小时计算 $R = 9.72$ 元/h。

我们将出租车从机场出发的平均距离作为出租车在未载客前的预期距离 $L = \mu = 8.65\text{km}$ 。由式 4.13，我们可计算得到选择在“蓄车池”中等待后，载客行驶 L 千米的成本为：

$$C_{wn} = 4.32 + 9.72(T_{wt} + \frac{12.65}{v_m}) \quad (4.24)$$

接下来，针对**直接回市区**的行为分析，它的成本为：

$$C_m = 4.32 + \frac{122.95}{v_m} \quad (4.25)$$

我们再根据式 4.14 对司机在市区的收费情况分析，具体为：

$$Q = \begin{cases} 14, x_{num} \leq 3 \\ 14 + 2.5(x_{num} - 3), x_{num} > 3 \end{cases} \quad (4.26)$$

然后根据市区的距离分布的带入，我们可以求得一次驾驶的期望收费，其中 $F(3)$ 表示行驶距离为 0~3 公里的概率， $1 - F(3)$ 为行驶距离大于 3 公里的概率，解得如下：

$$\begin{aligned} E1 &= 14 \cdot F(3) + ave[14 + 2.5(x_{num} - 3)] \cdot (1 - F(3)) \\ &= 14 \cdot 0.451 + 2.5 \cdot (11.88 - 3) \cdot (1 - 0.451) = 18.50 \text{ 元} \end{aligned} \quad (4.27)$$

综上，我们解得回市区的单位时间收益 W_1 ：

$$W_1 = \frac{13.18 - \frac{122.95}{v_m}}{T_{wt} + \frac{12.65}{v_m}} \quad (4.28)$$

再对从**机场等待出发**分析，它的期望收益为：

$$\begin{aligned} E2 &= 14 \cdot F(3) + ave[14 + 2.5(x_{num} - 3)] \cdot (1 - F(3)) \\ &= 14 \cdot 0.273 + 2.5 \cdot (15.377 - 3) \cdot (1 - 0.273) = 26.31 \text{ 元} \end{aligned} \quad (4.29)$$

所以留在机场等待的单位时间收益 W_2 ：

$$W_2 = \frac{21.99 - 9.72 (T_{wt} + \frac{12.65}{v_m})}{T_{wt} + \frac{12.65}{v_m}} \quad (4.30)$$

4.5 结果分析

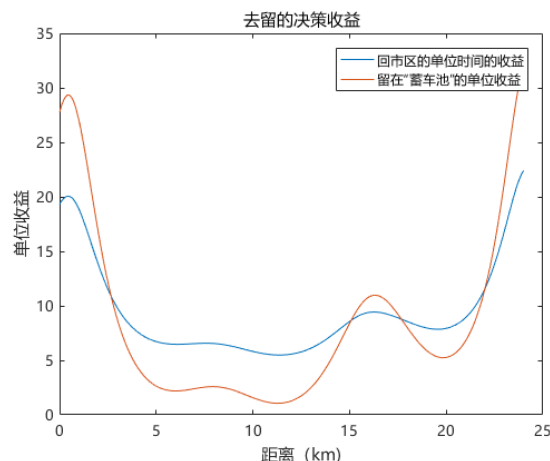


图 4-9 两种决策期望单位时间收益随潜在载客距离分布

综合以上公式 4.24-30，我们对在机场等待载客和回市区载客的单位时间收益纪念性分析，如图 4-8 所示。对图中状况进行分析，我们可得，潜在载客距离在约 3 千米范围内留在“蓄车池”的单位收益更大，在 3~15 千米大致范围内回市中心收益更大，而在超过约 22 千米后，留在“蓄车池”决策的期望收益将会更大。

4.5 模型合理性验证

为了检验模型的合理性，我们利用离开机场时出租车的状态进行分析，采用空载率指标反应司机的决策情况，得到的空载率随时间变化图如下：

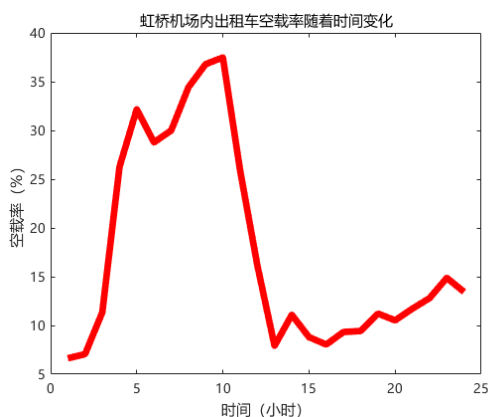


图 4-10 虹桥机场发出的出租车空载率随时间分布

可以看出当回到市区的期望收益越大时，离开机场时的空载率也随之升高，但是空载率最高未超过 50%，说明选择留在机场候车的司机还是占绝大多数的。可能是因为我们对于载客达到虹桥机场的出租车未剔除，增大了载客的出租车数量。【】【】

五、乘车效率最高模型的建立与求解

5.1 乘车效率最高模型的建立

由题目我们可知，机场中经常出现出租车等待和乘客等待的情况，为了使得乘车效率尽可能高，针对双并行车道进行分析，我们需安排合理的上车点位置，并选择合适的出租车数量，在到能够保证安全的情况下，目的使得乘车效率最高的方式。

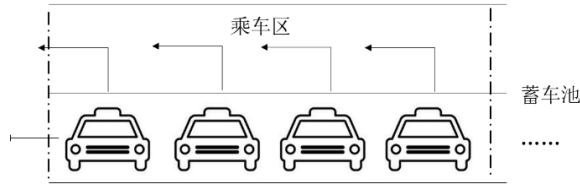


图 5-1 单车道乘车分析

如图 5-1，如果使用单车道的方式，所有车辆能够往前通行，在前面还未开始通行下后面出租车可使用旁边的道路出行，从而加快乘客的乘车时间。然而，使用单行道的方式并没有充分利用乘车区的空间，更适用于车辆不多或乘客不多情况下的方案，对于出租车和乘客量多的情况下，我们采用接下来的双车道应用分析。

5.1 单个上车点矩阵式停靠方式分析

对于双车道上车点的安排，如图 5-2 所示。为了保证乘客和出租车的安全性，我们规定乘客上车后，各个车道等待车辆依次驶出，若前面车辆仍在在上客则继续等待。当乘车区最后一辆出租车开始驶离乘车区时，“蓄车池”的出租车才允许开始进入乘车区，等其停稳后，再允许乘客进入乘车区。

同时，关于上车出口的位置，为了使可能缩短乘客的行走时间，我们将放行出口设在乘车区的中间或靠近中间区域。

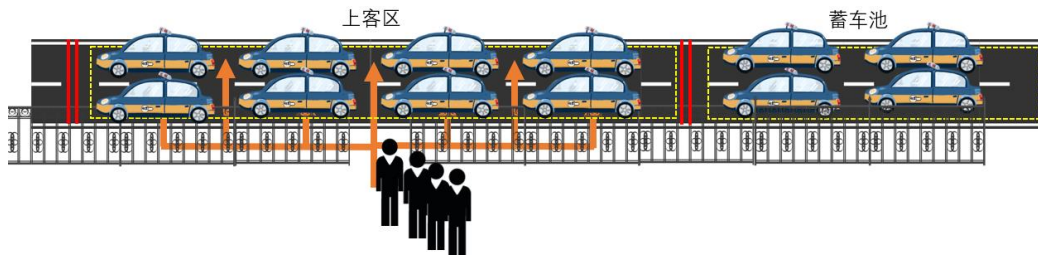


图 5-2 矩阵式停靠方式

5.1.1 乘车区运营一次的时长的刻画

我们定义相邻两次放行指令之间的时长为一次运营时长，其中包括乘客从驻足点移动至上车点的时长、乘客上车时长、“蓄车池”出租车离开乘车区时长三部分。基于 4.1.2 的出租车平均载客数量分析可知一辆出租车上客量 1.25，因此我们接下来的乘车出行我们认为可能单人或多人为一辆车，对同一辆车的人进行捆绑移动分析，并满足整体的出租车的平均上客人数为 1.25。接下来对各组成时长分析：

（一）乘客从驻足点进入上车点的时长

当放行指令发出后，乘客依次排序进入乘车区，靠后的乘客因其初始驻足点较远，且以放行出口为中点，将乘车区分为左右两块上车区。乘客从放行出口到达左边第一辆出租车的时间为半个泊位长度的行走时长 $\frac{1}{2}\Delta T$ ，到达后面出租车的时间依次递增一个泊位长度的行走时长 ΔT ，另外对于上车点为第二车道的乘客还需通行车道消耗的时间 $t_1^{[11]}$ 。因此乘从放行出口抵达上车点的行走时间为：

$$\begin{cases} T_{i,j,m} = (j - 0.5)\Delta T + (i - 1)t_1 \\ i = 1, 2 \\ j = 1, \dots, n \\ m = 1, 2 \\ \Delta T = 5s \\ t_1 = 2.5s \end{cases} \quad (5.1)$$

其中 i 表示车道， $m = 1, 2$ 分别表示左右上车区， j 表示排在 m 区域 i 车道的序号， n 表示 m 区域 i 车道的上车点数。

（二）乘客上车时长

考虑到行李量会造成乘客上车时长的不一致，而每个乘客所携带的行李量是一个未知数，所以设乘客的上车时长 G 为一个随机量。通过调查数据得到乘客上车时间的统计数据^[12]，分时段划分后得到下图，观察后发现乘客上车时间近似服从负指数分布。

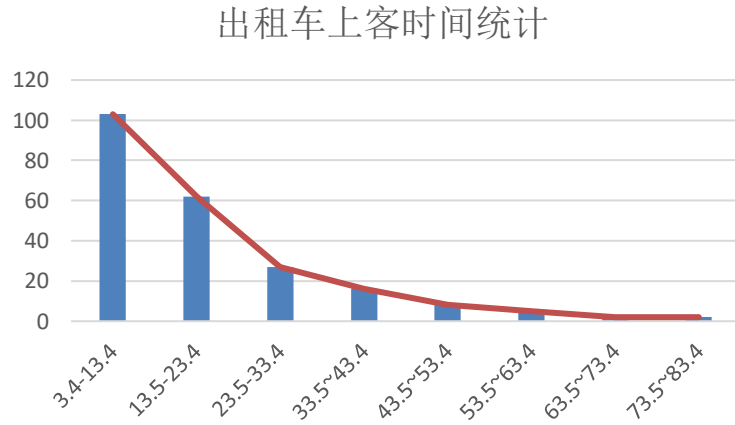


图 5-3 出租车乘客上车时间统计分布

（三）“蓄车池”出租车进入乘车区时长

“蓄车池”出租车进入指定泊车点的距离为乘车区整个长度，即一个车道内的泊车数量 $2n$ 乘以泊位长度 L_0 ，通过调查虹桥机场相关数据^[11]得到泊位长度 $L_0 = 5.5m$ ，出租车乘车区的平均行驶速度 $V_0 = 2.5m/s$ ，因此“蓄车池”出租车进入乘车区时长为：

$$T_b = \frac{n \cdot L_0}{V_0} \quad (5.2)$$

（四）乘车区运营一次的时长

因为乘客和出租车均满足先来后到的排队原则，所以根据 $FCFS$ 原则，最后一名乘客到达上车点的时间最长。但是因乘客上车时间的不一致，可能导致前一辆出租车还在上客，而后一辆出租车上客完毕后，仍需等待。因此我们找出乘客到达上车点的时长与上车时长之和最大的时长再加上“蓄车池”出租车进入乘车区时长，将总和作为乘车区运营一次的时长。

$$T' = \max(G_{i,j,m} + T_{i,j,m}) + T_b \quad (5.3)$$

5.1.2 乘车效率的刻画

对乘车效率，我们定义为通行能力来描述，即单位时间出租车的载客量，通行能力越大，乘车效率也越高。

$$\max C = \eta \cdot 4n/T' \quad (5.4)$$

因为乘客上车的时长是一个随机数，且对通行能力的影响较大，因此我们进行 x 次独立的测试，以减少偶然性，并将它们的平均值作为最终的通行能力。

$$\max \bar{C} = \frac{\sum_{i=1}^x \eta \cdot 4n/T'_x}{x} \quad (5.5)$$

5.1.3 基于蒙特卡洛模拟的乘客上车时间曲线的生成

根据前面所述，因行李量的不同造成了乘客上车时长的随机性，对于该随机数值分析，我们引入蒙特卡洛方法^[13]来模拟生成。我们根据调查得到机场乘客上车时间数据^[12]，以三秒为时间间隔进行划分，得到每个时间段的频次，并计算其对应的频率。然后再将频率转换为累积频率，详细数据见附录表 1。我们以累积频率为横坐标，乘客上车时间为纵坐标绘图结果如下：

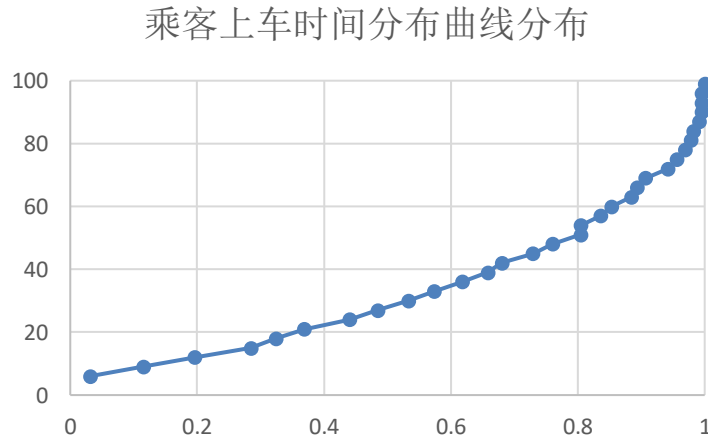


图 5-4 乘客上车时间分布曲线分布图

因数据调查量少，部分时间段的频次为 0，导致在后续的随机数生成中不会出现这些时间段。为了更好的反应随机性，我们使用 SPSS 软件对其进行指数拟合，得到 R 方为 0.989，F 为 2633.073，结果表明拟合程度很高，可以用来反应乘客上车时长的随机性。拟合函数如下，其中 ε 为随机量：

$$G = 7.231e^{2.52\varepsilon}, \varepsilon \in [0,1] \quad (5.6)$$

综上所述，针对单个上车点的模型如下：

$$\max \bar{C} = \frac{\sum_{i=1}^x \eta \cdot 4n/T'_x}{x} \quad (5.7)$$

$$s. t. \begin{cases} T' = \max(G_{i,j,m} + T_{i,j,m}) + T_b \\ T_{i,j,m} = (j - 0.5)\Delta T + (i - 1)t_1 \\ T_b = \frac{n \cdot L_0}{V_0} \\ G = 7.231e^{2.52\varepsilon}, \varepsilon \in [0,1] \\ i = 1,2 \\ j = 1, \dots, n \\ m = 1,2 \end{cases} \quad (5.8)$$

5.1.4 单个上车点泊车位数量的确定

我们使用随机数生成器生成一批次每个乘客的上车时间，然后代入上述单个上车点的模型，得到下图结果：

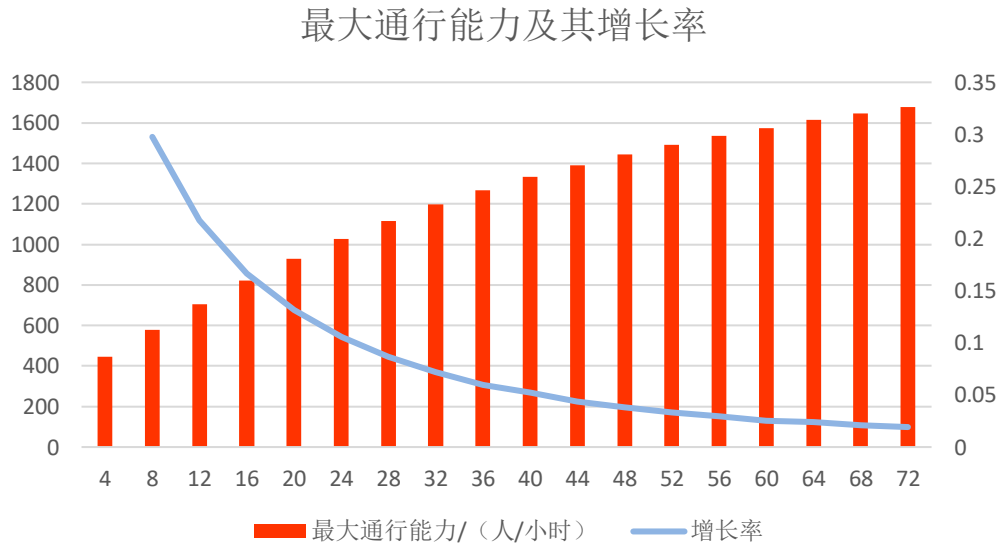


图 5-5 最大通行能力及其增长率

由图可知，最大通行能力随乘车区泊车位数量的增多而增多，但是增长率呈指数下降。考虑到泊车位越多，车辆的相互耦合与干扰现象变的越发严重，因此我们选定双车道共 16 个泊车位来展开后面对乘车点变化的研究。

5.2 多个上客点矩阵式停靠方式

5.2.1 上客点分配方案

对于多个上客点位置设立的目标是减少乘客从驻足点行走至出租车跟前的时间。将 8 个泊车位分配给 n 个上客点，为了使乘客行走时间最短，每一个上客点分配到的泊车位数量应尽量少，分配计算方法和结果如下：

$$8 \div n = a \dots b \quad (5.9)$$

这表明有 b 个上车点分配到 $(a + 1)$ 个泊车位，有 $(n - b)$ 个上车点分配到 a 个泊车位。

表 5-1 分配方案

上车 点数	分配情况	上车 点数	分配情况
1	1 个 8	5	2 个 1, 3 个 2
2	2 个 4	6	3 个 1, 2 个 2
3	1 个 2, 2 个 3	7	6 个 1, 1 个 2
4	4 个 2	8	8 个 1

分配过后的每个上客点就被独立开来，此时相当于前面对一个上客点的分析。随机数生成器生成乘客上车时长后，求出每一个独立上客点中乘客行走时长加上上车时间最大的值，再从这些上客点中挑选出最大的值。对于分配到 1 个泊车位的上客点，只考虑乘客上车时间，忽略行走时间。列出目标方程，其中 z 为上车点数。

$$\max (\max \bar{C}_j = \frac{\sum_{i=1}^x \eta \cdot 32 / T'_x}{x}) , j = 1, \dots, z \quad (5.10)$$

综上，多个上车点的最大效率模型如下：

$$\max (\max \bar{C}_j = \frac{\sum_{i=1}^x \eta \cdot 32 / T'_x}{x}) , j = 1, \dots, z \quad (5.11)$$

$$s.t. \begin{cases} T' = \max(G_{i,j} + T_{i,j}) + T_b \\ T_b = \frac{8L_0}{V_0} \\ G = 7.231e^{2.52\varepsilon}, \varepsilon \in [0,1] \\ i = 1,2 \\ j = 1, \dots, n \end{cases} \quad (5.12)$$

结果展示：

我们通过多次模拟得到下表所示结果，随着上车点数的增加，最大通行量增大但是增长率缓慢，若客流量不大，宜设立一个或两个上车点数即可。

表 5-2 最大通行量与上车点数的关系

上车 点数	最大通行量	增长率	上车 点数	最大通行 量	增长率
1	672.2	0.058018	5	735.6	0.00643
2	711.2		6	739.9	0.005846
3	724.6	0.018841	7	745.3	0.007298
4	730.9	0.008694	8	750.8	0.00738

六、不平衡度模型的建立与求解

6.1 不平衡度模型的建立

6.1.1 一次订单时长的刻画

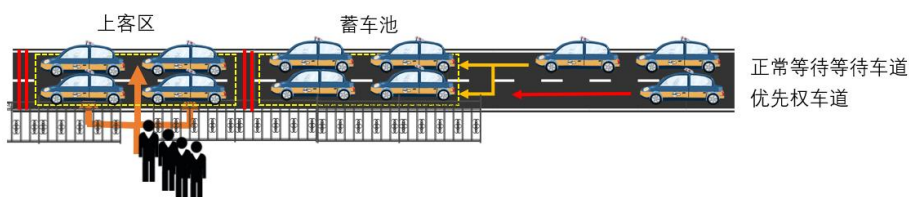


图 6-1 优先权车道的设计

我们将出租车第一次开始排队等待到第二次开始排队等待的时长定义为一次订单时长。刚开始时刻司机并没有优先权，因此等待时间为 T_{wt} 。等乘客上车后，短、长途载客司机将乘客载到目的地后再原路返回的时长分别为 T_{m1} ， T_{m2} 。此时短途司机将会给予一个优先时间 t_b ，而长途司机依次排序。如果再继续分析下去，将会产生如下图四种情况，周而复始，情况会越来越多，不利于分析司机单位时间的利润。

因此我们可以将短程载客司机第一次等待的时长与给予优先时间后的第二次等待时长进行交换，即可以理解为先给司机一个优先时间，然后让司机搭载短途旅行的乘客。由此我们建立长短途的订单时长模型：

$$\begin{cases} T_1 = 2T_{m1} + T_{wt} - t_b \\ T_2 = 2T_{m2} + T_{wt} \end{cases} \quad (6.1)$$

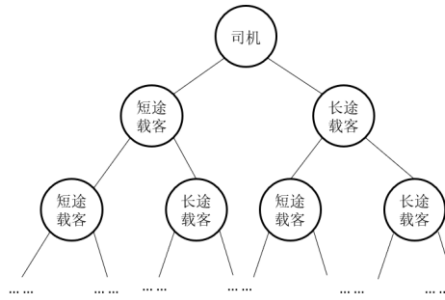


图 6-2 司机长短途载客情况

6.1.2 单位时间利润的刻画

在问题一的讨论中我们已求出利润与收益、成本的关系式，对其时间参数进行改进得到的公式如下：

成本：

$$C_{m1} = 2L_1 \cdot r + T_1 \cdot R \quad (6.2)$$

$$C_{m2} = 2L_2 \cdot r + T_2 \cdot R \quad (6.3)$$

收费：

$$Q = \begin{cases} a_0, L < L_0 \\ a_0 + (L - L_0)a, L \geq L_0 \end{cases} \quad (6.4)$$

单位时间利润：

$$Q'_1 = \frac{Q_1 - C_{m1}}{T_1} \quad (6.5)$$

$$Q'_2 = \frac{Q_2 - C_{m2}}{T_2} \quad (6.6)$$

6.1.3 不平衡度指标的建立

我们将短途载客与长途载客单位时间利润之差的绝对值定义为不平衡度，不平衡度越小，出租车收益越均衡。

$$\min F = |Q'_1 - Q'_2| \quad (6.7)$$

对于多少距离区分长短途，我们设置一个路程阈值 d ，使得小于 d 的当做短程，所以我们的目标就是寻找合适的 d ，使得不平衡度最小。

综上所述，最终模型为：

$$\min F = |Q'_1 - Q'_2| \quad (6.8)$$

$$s. t. \begin{cases} Q'_1 = \frac{Q_1 - C_{m1}}{T_1} \\ Q'_2 = \frac{Q_2 - C_{m2}}{T_2} \\ Q = \begin{cases} a_0, L < L_0 \\ a_0 + (L - L_0)a, L \geq L_0 \end{cases} \\ C_{m1} = 2L_1 \cdot r + T_1 \cdot R \\ C_{m2} = 2L_2 \cdot r + T_2 \cdot R \\ T_1 = 2T_{m1} + T_{wt} - t_b \\ T_2 = 2T_{m2} + T_{wt} \end{cases} \quad (6.9)$$

6.2 模型的求解

对于短途的平均距离

$$\overline{dd} = \frac{\sum_{i=1}^d x_{num} \cdot f(i)}{F(d)} \quad (6.10)$$

长途的平均距离：

$$\overline{dc} = \frac{\sum_{i=d}^{\infty} x_{num} \cdot f(i)}{1 - F(d)} \quad (6.11)$$

为了简化计算，对于 2 次接客所需时间我们忽略其随着时间变化的差异。

所以第一次为长途，在第二次载客所花时间

$$T_2 = T_{m2} + T_{wt} = \frac{\overline{dc}}{v_m} + T_{wt} \quad (6.12)$$

对于由于第一次短途而获得“优先权”的出租车，其第二次的等待时间与正常排队进场的时间相比而言非常少，所以我们将获得“优先权”的出租车的等待时间 $T_{wt} \approx t_b$ ，故相当于第二次载客不需要等待。为了更好的衡量单次收益，由于第二次的“优先权”的收益是由第一次短途带来的，我们可以将其当做第一次短途不需要等待。即 $T_1 = T_{m1} + T_{wt} - t_b = T_{m1} = \overline{dd}$

单次短途的成本

$$C_{m1} = 2L_1 \cdot r + T_1 \cdot R = 2\overline{dd} \cdot r + \frac{\overline{dd}}{v_m} \cdot R = 1.2\overline{dd} + 9.72 \frac{\overline{dd}}{v_m} \quad (6.13)$$

单次长途的成本：

$$C_{m2} = 2L_2 \cdot r + T_2 \cdot R = 2\overline{dc} \cdot r + \left(\frac{\overline{dc}}{v_m} + T_{wt} \right) \cdot R = 1.2\overline{dc} + 9.72 \left(\frac{\overline{dc}}{v_m} + T_{wt} \right) \quad (6.14)$$

单次短途的收费为

$$Q_1 = \begin{cases} 14, \overline{dd} \leq 3 \\ 14 + 2.5(\overline{dd} - 3), \overline{dd} > 3 \end{cases} \quad (6.15)$$

单次长途的收费为：

$$Q_2 = \begin{cases} 14, \overline{dc} \leq 3 \\ 14 + 2.5(\overline{dc} - 3), \overline{dd} > 3 \end{cases} \quad (6.16)$$

所以单次短途单位时间的收益为：

$$Q'_1 = \frac{Q_1 - C_{m1}}{T_1} = \frac{Q_1 - (1.2\overline{dd} + 9.72\frac{\overline{dd}}{v_m})}{\overline{dd}} \quad (6.17)$$

单次长途单位时间的收益为：

$$Q'_2 = \frac{Q_2 - C_{m2}}{T_2} = \frac{Q_2 - [1.2\overline{dc} + 9.72(\frac{\overline{dc}}{v_m} + T_{wt})]}{\frac{\overline{dc}}{v_m} + T_{wt}} \quad (6.18)$$

我们求得随着时间变化，设定的阈值大小变化的不平衡度变换图如下所示：

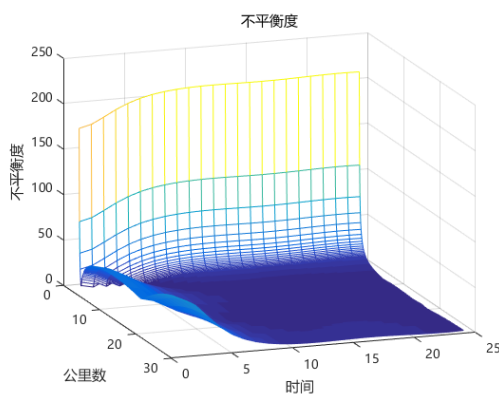


图 6-3 不平衡度三维分布情况

我们找出在相同时间下不平衡度最小时的公里数，将其设为该时间段下最佳阈值，结果如下图所示。

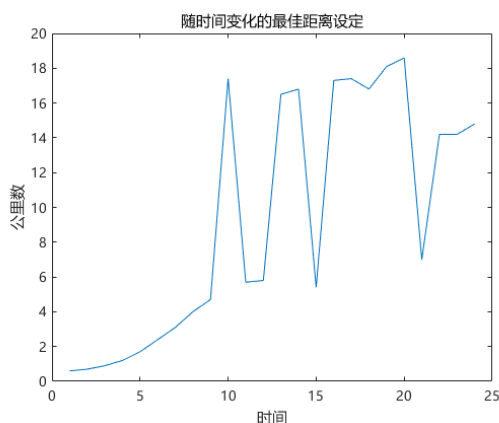


图 6-4 最佳距离随时间变化

由于最佳阈值随着时间分段改变量相差较大，所以我们对阈值的设置进行分段处理，将其分为 0 点至 9 点，9 点至 24 点，而对于 0 点至 9 点阶段，公里数较小，且出行距离分布小于 4 公里的仅有 13.7%，所以我们对其不设置阈值，在 9 点至 24 点阶段，我们对

阈值的设置取该时段最佳阈值的平均为 14km。

七、模型的评价及改进

7.1 模型的优点

1. 我们对于司机的决策考虑到随着时间变化的影响，更能反应真实情况。
2. 对于问题一我们综合考虑了多种影响因子更能做出准确的决策。
3. 对于问题四，我们对长短途司机收益用不平衡度表示，简化了模型。

7.2 模型的缺点

1. 由于缺少数据，我们对于影响司机决策的因子都是以离散的数值计算，然后再拟合出影响因子的函数，影响了结果的精确性。
2. 我们对于问题一的决策没有对其进行置信度的分析，降低了可靠性。
3. 对于第三问，我们只考虑了一种出租车等待方案，我们可以考虑出租车在一个车道上排队接客，在另外一个车道作为载客后驶出的道路。
4. 对于第四问，我们只考虑了一种阈值设定方式，我们还可以对时间进行阈值设定，以比较最优方案。

八、参考文献

- [1] 康朝贵, 刘璇, 许欣悦, 秦昆. 天气因素对武汉市出租车出行活动的影响. 2019(01):118-127.
- [2] Kamga, C., Yazici, M.A., Singhal, A..Hailing in the Rain: Temporal and Weather-Related Variations in Taxi Ridership 2 and Taxi Demand-Supply Equilibrium
- [3] <http://news.yhzuiche.com/index/1-29.html>
- [4] <http://flightadsb.variflight.com/>
- [5]<https://baike.baidu.com/item/%E4%B8%8A%E6%B5%B7%E5%BC%BA%E7%94%9F%E6%8E%A7%E8%82%A1%E8%82%A1%E4%BB%BD%E6%9C%89%E9%99%90%E5%85%AC%E5%8F%B8/8862774>
- [6] <http://www.62580000.com.cn/czqc.html>
- [7]<https://baike.baidu.com/item/%E6%9B%BC%E5%93%88%E9%A1%BF%E8%B7%9D%E7%A6%BB/743092?fr=aladdin>
- [8] <http://www.chyxx.com/industry/201712/593521.html>
- [9] <https://wenku.baidu.com/view/4785c00bbalaa8114431d95d.html>
- [10] Yuxiong Ji, Yuchuan Du, Yue Liu, A.M. ASCE, and H. Michael Zhang. Empirical Behavioral Study of Airport-Serving Taxi Drivers.Dev, 2017. Using Automatic Vehicle Location Data
- [11] 柳伍生, 谭倩. 交通枢纽出租车车道边通行能力仿真研究. (2102)04-0357-05
- [12] 孙健, 基于排队论的单车道出租车上客系统建模与仿真
- [13]<https://baike.baidu.com/item/%E8%92%99%E7%89%B9%E5%8D%A1%E7%BD%97%E6%A8%A1%E6%8B%9F/5160083?fr=aladdin>

九、附录

9.1 附表

附表 1

时间段	频次	频率	累积频率	时间段	频次	频率	累积频率
3s ~ 6s	7	0.031111	0.031111	51s ~ 54s	0	0	0.804444
6s ~ 9s	19	0.084444	0.115556	54s ~ 57s	7	0.031111	0.835556
9s ~ 12s	18	0.08	0.195556	57s ~ 60s	4	0.017778	0.853333
12s ~ 15s	20	0.088889	0.284444	60s ~ 63s	7	0.031111	0.884444
15s ~ 18s	9	0.04	0.324444	63s ~ 66s	2	0.008889	0.893333
18s ~ 21s	10	0.044444	0.368889	66s ~ 69s	3	0.013333	0.906667
21s ~ 24s	16	0.071111	0.44	69s ~ 72s	8	0.035556	0.942222
24s ~ 27s	10	0.044444	0.484444	72s ~ 75s	3	0.013333	0.955556
27s ~ 30s	11	0.048889	0.533333	75s ~ 78s	3	0.013333	0.968889
30s ~ 33s	9	0.04	0.573333	78s ~ 81s	2	0.008889	0.977778
33s ~ 36s	10	0.044444	0.617778	81s ~ 84s	1	0.004444	0.982222
36s ~ 39s	9	0.04	0.657778	84s ~ 87s	2	0.008889	0.991111
39s ~ 42s	5	0.022222	0.68	87s ~ 90s	1	0.004444	0.995556
42s ~ 45s	11	0.048889	0.728889	90s ~ 93s	0	0	0.995556
45s ~ 48s	7	0.031111	0.76	93s ~ 96s	0	0	0.995556
48s ~ 51s	10	0.044444	0.804444	96s ~ 99s	1	0.004444	1

9.2 附代码

附录名称	出租车数据筛选	代码语言	Python
<pre># -*- coding: utf-8 -*- import pymysql def add(): db = pymysql.connect("localhost","root","a1181693065","model") cursor = db.cursor() s="load data infile 'C:/Users/hasee/Desktop/HT180401/01/" s1="/180401" s2=".txt' replace into table c" s3=" fields terminated by' '" d0="00" d1="02" #ss="load data infile 'C:\Users\hasee\Desktop\1\c14.csv' replace into table c fields terminated by','" for n in range(14,23): print(n) for i in range(n,n+1):</pre>			

```

        if i<10:
            d0="0"+str(i)
        else:
            d0=str(i)
    for j in range(1,60):
        if j<10:
            d1="0"+str(j)
        else:
            d1=str(j)
        k=s+d0+s1+d0+d1+s2+str(n)+s3
        try:
            cursor.execute(s)
            db.commit()
            print(j)
        except:
            continue

    ins()

def delt():
    db =
pymysql.connect("localhost","root","a1181693065","model" )
    cursor = db.cursor()
    s="delete from car where time <> '2018-04-01 00:00:00' "
    sq=" and time <> "
    s1="2018-04-01 "
    s2=":00"
    sm=""
    dl=" LIMIT 50000"
    for i in range(24):
        if i<10:
            d0="0"+str(i)
        else:
            d0=str(i)
        for j in range(60):
            if j<10:
                d1="0"+str(j)
            else:
                d1=str(j)
            sm=sm+sq+"'+s1+d0+":"+d1+s2+''"
    sm=s+sm+dl;

```



```

        print(sm)
        for k in range(9000):
            cursor.execute(sm)
            db.commit()
            print(k)

def ins(n):
    #db =
    pymysql.connect("localhost","root","a1181693065","model" )
    #cursor = db.cursor()
    s="select * from cq17 where time = '2018-04-01 00:00:00' "
    sq=" or time = "
    s1="2018-04-01 "
    s2=":00"
    sm=""
    for i in range(n,n+1):
        if i<10:
            d0="0"+str(i)
        else:
            d0=str(i)
        for j in range(60):
            if j<10:
                d1="0"+str(j)
            else:
                d1=str(j)
            sm=sm+sq+"'+s1+d0+":"+d1+s2+"'"
    sm=s+sm;
    print(sm)

import math as m
def getd(u1,v1,u2,v2):
    u=m.fabs((u2-u1)*m.pi/180)
    s1=100000*m.sqrt(m.pow(1.1132*(u1-u2),2)+m.pow(v1-v2,2))
    s2=2*m.asin(m.sqrt(m.pow(m.sin(v2-
v1)*m.pi/360,2)+m.cos(v1*m.pi/180)*m.cos(v2*m.pi/180)*m.pow(m.sin(u
/2),2))) *6378138
    s=s2/s1*100000*(1.1132*m.fabs(u1-u2)+m.fabs(v1-v2))
    return s

def caldd():

```

```

d=[i for i in range(10000)]
db =
pymysql.connect("localhost","root","a1181693065","model" )
    cursor = db.cursor()
    cursor.execute("SELECT distinct(id) FROM `c` WHERE
`121.546585` BETWEEN '121.307' AND '121.356' AND `31.275518`
BETWEEN '31.178' AND '31.223'")
    results= cursor.fetchall()
    m=len(results)
    q=1
    for i in range(600,700):
        id1=results[i][0]
        k1="SELECT * FROM `c14` WHERE id='%s' " %(id1)
        print(k1)
        cursor.execute(k1)
        results1 = cursor.fetchall()
        k=len(results1)
        flag=-1
        for j in range(k):
            try:
                c1=int(results1[j][2])
                cj=float(results1[j][5])
                cw=float(results1[j][6])
            except:
                continue
            #print(c1,cj,cw)
            #print(c1==1,cj< 121.37,cj>121.307)
            if c1==1 and cj< 121.37 and cj>121.307 and cw> 31.178
and cw<31.223:
                #print(c1,cj,cw)
                dw=cw
                dj=cj
                flag=1
            #print(c1,flag)
            if flag==1 and c1==0:
                #print(c1,cj,cw)
                flag=0
            if flag==0 and c1==1:
                #print(j,c1,cj,cw)
                fw=cw

```

```

        fj=cj
        d[q]=getd(dj,dw,fj,fw)
        print(d[q])
        q=q+1
        flag=-1

    print(d)

def ww():
    s=[0 for i in range(10000)]
    sl=[0 for i in range(10000)]
    ss=[0 for i in range(10000)]
    db =
pymysql.connect("localhost","root","a1181693065","model" )
    cursor = db.cursor()
    cursor.execute("SELECT * FROM `c22` WHERE `31.276102`
BETWEEN '31.1744' AND '31.2152' AND `121.475248` BETWEEN '121.3065'
AND '121.3520' ORDER BY `id`")
    results= cursor.fetchall()
    m=len(results)
    a0t="0"
    p=0
    for i in range(m):
        a0=results[i][0]
        a1=int(results[i][1])
        if a0t==a0:
            s[p]=s[p]+1
            sl[p]=sl[p]+a1
        else:
            ss[p]=((s[p]-sl[p])==0)
            p=p+1
        a0t=a0
    cursor.execute("SELECT count(distinct(id)) FROM `c22` WHERE
`31.276102` BETWEEN '31.1744' AND '31.2152' AND `121.475248`
BETWEEN '121.3065' AND '121.3520' ORDER BY `id`")
    results= cursor.fetchall()
    print(sum(ss),results[0][0])

```

附录名称	距离分布	代码语言	MATLAB
close all			

```

a=[12836.6967687873;10288.2900000000;17649.4800000000;35875.351
2000000;8180;118858.2900000000;5971;23225.6800000000;4573;8650.29000
000000;12940.4060000000;2449.93553100000;11865;8243.390000000000;174
43.2500000000;3572.66900000000;8856;15714;13768;11449;32761;6334;79
06;2891;4743;879;17638;834;2579;7704;5633;4574;4378;2146;2096;2385;
2550;16990.970130409532];

[m1,n1]=size(a)
l=60
b=zeros(l,1);
for i=1:m1
    k=round(a(i)/1000/2+0.49);
    b(k)=b(k)+1;
end
b1=b;
b=b*200;
for i=1:l
    b(i)=unidrnd(200,1,1)/log(i*2)+b(i)
end
figure
plot(b(1:30),'ro')
hold on
t=1:60;
x=0:0.1:30;
a1=4;
a2=3;
plot(0:0.1:30,16000/(sqrt(2*pi)*a1)*exp(-(x-
a2).^2/(2*a1^2)),'r')
xticklabels(0:10:50)
hold on
syms x1
sr=int(1/(sqrt(2*pi)*a1)*exp(-(x1-a2).^2/(2*a1^2)),x1,0,3)
db=b(2:31).*(0.5+rand(30,1))
db=[1613.80767833525;964.586777152166;471.263358803141;864.9044
70035113;464.797153714076;351.612216805017;144.710371187153;612.005
134483118;3.02858530727999;58.6642420032123;329.300386019059;16.090
5602074945;70.3446208567383;48.6627249156705;51.7748269921676;328.5
11218745500;142.134157964729;34.8951669240120;15.7704268977038;36.0
177285276041;3.21284135314499;25.8407911160479;2.38596906484469;19.
9795814155884;36.4384439719204;35.6417274798460;27.5581622656600;11
.4844582260011;35.1337899846238;22.6616077985421];

```

```

plot(db, 'bx')
hold on
a1=2.5;
a2=1.5;
plot(0:0.1:30, 10000/(sqrt(2*pi)*a1)*exp(-(x-
a2).^2/(2*a1^2)), 'b')
legend('机场出租车实际距离分布', '机场出租车拟合距离分布', '市区出租车实际
距离分布', '市区出租车拟合距离分布')
title('机场出发的出租车平均行驶距离')
xlabel('距离 (km)')
ylabel('车辆数')
set(gca, 'Fontname', 'Microsoft YaHei UI')
ss=0;
for i=1:30
    ss=ss+i*2*db(i);
end
ss/sum(db(1:30))
sr=int(1/(sqrt(2*pi)*a1)*exp(-(x-a2).^2/(2*a1^2)), x1, 0, 3)

```

附录名称	第四问不平衡度	代码语言	MATLAB
<pre> clear clc a1=4; a2=3; syms x for t=1:24 v=73.05+16.49*cos(0.2414*t/2)+8.912*sin(0.2414*t/2)- 11.37*cos(0.4828*t/2)-10.59*sin(0.4828*t/2)-11.48*cos(0.7242*t/2)- 3.655*sin(0.7242*t/2); tw=73.05-24.73*cos(0.2633*t/2)+16.97*sin(0.2633*t/2)- 7.243*cos(0.5266*t/2)-13.41*sin(0.5266*t/2)-17.25*cos(0.7899*t/2)- 5.911*sin(0.7899*t/2); tw=tw/120; for d=1:0.1:30 dt=0;dc=0; for i=1:0.1:d dt=dt+i*(1/(sqrt(2*pi)*a1)*exp(-(i/2-a2).^2/(2*a1^2))); end I=int(1/sqrt(2*pi)*a1*exp(-(x/2-a2).^2/(2*a1^2)), x, 0, d)/32; dt=dt/I/32; dt=vpa(dt); end end </pre>			

```

for i=d:0.1:100
    dc=dc+i*(1/(sqrt(2*pi)*a1)*exp(-(i/2-a2).^2/(2*a1^2)));
end
dc=dc/(1-I)/32;
dc=vpa(dc);
if dt<=3
    q1=14;
else
    q1=14+2.5*(dt-3);
end
if dc<=3
    q2=14;
else
    q2=14+2.5*(dc-3);
end
q1t=(q1-1.2*dt-9.72.*dt./v)./dt;
q2t=(q2-(1.2*dc+9.72*(dc/v+tw)))/(dc/v+tw);
dq(t,d*10-9)=vpa(abs(q1t-q2t));
end

end

dq=double(dq)
mesh(1:0.1:30,1:24,dq)
title('不平衡度')
xlabel('公里数')
ylabel('时间')
zlabel('不平衡度')
set(gca,'Fontname','Microsoft YaHei UI')
figure
[dat,km]=min(dq')
plot(1:24,km/10)
title('随时间变化的最佳距离设定')
ylabel('公里数')
xlabel('时间')
set(gca,'Fontname','Microsoft YaHei UI')

```

附录名称

航班人数变换

代码语言

MATLAB

```

close all
w1=[0,0,0,0,0,0,0,0,18,37,60,33,56,60,59,67,71,61,60,60,60,64,8
2,81];

```

<pre> w2=w1*200 figure bar(w1) title('虹桥机场随着时间机场每小时到达航班数') xlabel('时间（小时）') ylabel('航班数') set(gca,'Fontname','Microsoft YaHei UI') for i=1:22 w2(i)=w2(i)*0.2*(0.5+rand(1)) end for i=23:24 w2(i)=w2(i)*0.7*(0.5+rand(1)) end figure bar(w2) title('虹桥机场随着时间机场每小时选择乘坐出租车的人数') xlabel('时间（小时）') ylabel('人数') set(gca,'Fontname','Microsoft YaHei UI') </pre>			
附录名称	曼哈顿距离	代码语言	MATLAB
<pre> function f=getd(u1,v1,u2,v2) u=abs((u2-u1)*pi/180); s1=100000*sqrt(power(1.1132*(u1-u2),2)+power(v1-v2,2)); s2=2*asin(sqrt(power(sin(v2- v1)*pi/360,2)+cos(v1*pi/180)*cos(v2*pi/180)*power(sin(u/2),2)))*637 8138; s=s2/s1*100000*(1.1132*abs(u1-u2)+abs(v1-v2)); f=s; end </pre>			
附录名称	出租车等待时间	代码语言	MATLAB
<pre> clear clc a=[30,30,38,111,110,100,90,90,104,113,112,108,90,71,56,44,53,64 ,72,70,62,61,50,17]; bar(a) title('出租车等待时间随时间的变化') xlabel('一天的时刻（小时）') ylabel('出租车等待的时间（分钟）') set(gca,'Fontname','Microsoft YaHei UI') hold on </pre>			

```

a0 =73.05;
a1 =-24.73;
b1 =16.97;
a2 =-7.243;
b2 =-13.41;
a3 =-17.25;
b3 =-5.911;
w =0.2633;
x=1:0.1:24
plot(x,a0 + a1*cos(x*w) + b1*sin(x*w) + a2*cos(2*x*w) +
b2*sin(2*x*w) + a3*cos(3*x*w) + b3*sin(3*x*w),'r','LineWidth',5)

aa=[100,102,97,103,112,90,58,42,45,56,65,67,78,75,77,62,54,46,4
8,61,65,74,82,85];
a0 =73.43
a1 =16.49;
b1 =8.912;
a2 =11.37;
b2 =10.59;
a3 =-11.48;
b3 =-3.655;
w =0.2414;
figure
bar(aa)
hold on
x=1:0.1:24
plot(x,a0 + a1*cos(x*w) + b1*sin(x*w) + a2*cos(2*x*w) +
b2*sin(2*x*w) + a3*cos(3*x*w) + b3*sin(3*x*w),'r','LineWidth',5)
title('出租车速度随时间的变化')
xlabel('一天的时刻（小时）')
ylabel('出租车的速度（千米）')
set(gca,'Fontname','Microsoft YaHei UI')

```