机场出租车最优收益决策与管理模式探讨

摘要

机场出租车服务表面看起来似乎是按序排队先来后到那样简单，但实际上司机收益与其离开或等待的选择息息相关、选择又受到排队时间与承载随机客人所获收益的影响，机场出租车调度管理十分复杂。本文通过建立数学模型，研究送客到机场的出租车司机做出离开机场或等待接客选择的问题，并为机场调度出租车提出收益均衡且效率最高的优化方案。

对于问题一，是根据实际情况为司机提供可获得更优收益的决策判断问题。选择某时间段内到达航班数、可观测的排队出租车数作为影响出租车司机决策模型的主要影响因素，运用泊松过程对两主要因素进行分析，定义决策函数，对司机两项选择的收入期望做出定性与定量分析并进行判断选择。因此，在得到某时间段内到达航班数、排队出租车数的信息后，司机基于一定个人经验即可做出期望收益更大的选择。

对于问题二，我们从国内各航空公司及上海市出租车公司网站爬取获得相关上海市浦东国际机场的航班信息与机场出租车流量、收益等信息，构建一日内的航班、出租车时间序列曲线，利用 Tableau 软件进行可视化，给出模型图表信息，验证了问题一模型的合理性，再通过控制变量法分析对相关因素的依赖性关系。

对于问题三，我们运用M/M/S模型得出两并列车道应被规划为单车道单独发车为最优的“上车点”位置设置，并利用MATLAB编程进行仿真模拟作图解答进行证明。再优化排队系统费用决策模型决定“上车点”的数量M问题，并从合理安排排队乘客的角度弥补“上车点”模型可能存在的时间偏差问题，综上得出了在保证车辆和乘客安全的条件下，机场出租车乘车效率最高的“上车点”管理安排方案。

对于问题四，我们将使出租车收益尽量均衡的问题进行转化，通过给予短途载客的出租车一定的优先权（优先权表现为缩短排队等待的时间，减少时间成本），变为动态求解每辆出租车收益实际值和理论值差的最小二乘法的最小值问题，列出对最小二乘目标函数的约束条件，运用运筹学理论求解得满足最小值的每辆出租车每日对应的载客任务集和任务时间集，即可再转换为给予出租车“优先权”的“优先可行”安排方案。最后，我们还给出了对安排方案的具体实施提出了建设智能化系统的建议，使得机场出租车服务质量得到改进提升。

本文的最后对所用模型进行了客观评价，分析了模型的实用性合理性，并对模型的缺陷提出了一些改进建议。

关键词：决策管理优化；出租车上车点；通行效率；排队论；任务均衡

一、问题重述

1.1 问题背景

对于送客到机场的出租车司机来说，将会面临两个选择：排队等待载客返回市区或直接放空返回市区拉客。对于机场出租车管理人员来说，将面临乘客上车和出租车辆放行的管理问题，为了达到乘车效率最大化和出租车司机收益的均衡，管理部门需要合理设置上车点和乘客、出租车安排方案。

1.2 问题提出

问题要求结合实际情况，使用搜集的相关数据，建立数学模型分析以下问题：

（1） 分析影响出租车决策的因素，建立出租车司机选择决策模型，给出司机的选择策略；

（2） 使用搜集到的数据，给出该机场出租车司机的选择方案，分析模型合理性和对相关因素的依赖性；

（3） 如何通过设置“上车点”、合理安排出租车和乘客，实现总乘车效率最高；

（4） 为实现出租车收益均衡，对某些短途载客在此返回的出租车给予一定的“优先权”，给出优先安排方案。

二、问题分析

2.1 问题一的分析

综合考虑各种因素对机场乘客数量的变化和出租车的收益的影响，为出租车司机建立合理的选择决策模型，帮助司机获得更大的收益十分重要。司机在机场做出等待载客和放空回市区选择的关键是比较两项期望收入的大小，且这个选择仅存在于当前状态，我们用相同马尔科夫过程的状态点来描述它 建立两种选择期望收入的数学模型。且使用排队论模型对司机排队等待的具体过程进行分析预测。

2.2 问题二的分析

基于问题一建立的模型，需要在网站上爬取某一机场足量的航班信息和城市出租车的各方面信息，带入问题一的模型对其合理性进行分析。影响司机选择的因素有很多，为了分析每一因素的影响，我们可以运用控制变量的方法分别分析每一项对结果的影响，从而分析相关因素的依赖性。

2.3 问题三的分析

管理部门合理的设置“上车点”，有助于保护出租车和乘客的安全并加快行车进程，使得总的乘车效率最高。模型将由位置、数量、乘客管理三部分组成，在已有乘车区两条并行车道的题目要求下，基于 M/M/S 理论从出租车和等待乘客两个角度出发进行分析，并得出如何合理安排三种因素才能够达使模型的最优化。

2.4 问题四的分析

可行的“优先”安排方案，有助于机场出租车的收益尽量保持均衡，进一步改善机场出租车行业的运营环境，提高机场服务品质。我们将该模型转换为对出租车一日期望收益、每辆车实际收益和实际成本模型的解析，并利用运筹学理论

与最小二乘方法计算出合适的任务集与时间集，从而使得机场每辆出租车的收益尽量保持均衡。

三、基本假设和符号说明

3.1 基本假设

（1）出租车正常行驶状态视为匀速直线运动
（2）出租车一日内始终固定服务同一机场的接送客
（3）出租车计价方式按照起步价+里程数计价，夜间行车和郊区行车有加价
（4）司机在机场始终在接客等待队伍内，不考虑进行吃饭等其他活动的时间
（5）出租车离开机场返回市区后一定能接到乘客，只是寻找客人时间有长有短
（6）不考虑出租车在行驶过程中堵车等交通状况和车辆或道路故障的意外情况
（7）司机在机场是否拉客的决策不受当日已拉客人数量的影响，且对其每日工作的总时长没有影响

3.2 符号说明：

符号	解释	单位
t_w	出租车的等待时间	h
n_p	为每辆出租车的载客数量	人
N_p	为到达乘车区的顾客数	人
c	常量参数	-
S	司机可观测前方排队等待的出租车数量	辆
λ	单位时间内到达乘车区的乘客数量	人
k	t_w时间内到达乘车区的顾客数	人
$f0$	机场出租车的起步价	$yuan$
$f1,f2$	出租车每公里的加价（呈阶梯式）	$yuan/km$
d_{a2c}	从机场到市区出租车行驶的距离	km
d_{c2r}	从市区到任意地点出租车行驶的距离	km
C^r	租车的单位时间费用	$yuan$
C^f	单位时间的燃油成本	$yuan$
t_s	市区找乘客的时间	h
v	出租车匀速行驶的速度	km/h
d_0	机场到市中心的距离	km
t	出租车行驶总时间	h
μ	乘客上车的平均效率	-
S_{all}	全天服务机场乘客的出租车数量	辆
D	某时刻所需出租车数量	辆
d_j	机场出发的第 j 次接客的行驶距离	km

四、模型建立与求解

4.1 问题 1：司机选择决策模型

司机在机场面临等待载客和放空回市区的选择，其关键是比较在两种选择下的期望收入的大小。我们将两种选择的期望收入作为目标，进行建模。由于此时的决策仅依赖于当前状态，而不依赖过去历史，可以视为马尔科夫过程的一个状态点进行分析，因此模型的描述借用马氏决策问题中的一些概念。

考虑从选择开始到这两种选择结果最终达到相同的状态的时间段，通过比较他们的效用函数的大小进行决策。由于当他们接客完毕又回到机场时，又面临同样的选择——等待接客或回到市区，所以可将终止状态均设定为：从机场回到市区接客并送客到机场完成订单。

4.1.1 排队等待接客的效用函数

1、接到客人的概率：

排队等待接客的情况需要考虑等待结束后是否能够接到客人。设接到客的概率为 P，P 与机场乘客的数量、机场出租车的数量有关。当该出租车前方出租车可承载乘客量远远大于机场乘客数量时，则无法接到客人。因此我们需要获得乘客出租车需求量。

司机在等待时间 t_w 内观测到的出租车需求量是累计随机事件发生次数的独立增量过程：

$$D = \frac{N_p}{n_p}$$

其中 n_p 为每辆出租车的载客数量，一般取值为 2.2；

N_p为到达乘车区的顾客数，在t_w 时间内服从泊松过程，到达人数的服从以下概率分布[1]：

$$P[N_p(t_w) = k] = \frac{e^{-\lambda t_w}(\lambda t_w)^k}{k!}$$

λ 与近期到达航班数量 N_f 以及乘客在机场选择出租车出行的比率 R_{apt} 有关，实际含义为单位时间内到达乘车区的乘客数量：

$\lambda \propto N_f \qquad \lambda \propto R_{apt}$

N_f 的值司机可以通过每日航班信息获得；R_{apt} 通过测算及经验获得，同一机场的 R_{apt} 有一定的规律，通过历史数据得到，影响 R_{apt} 的因素有工作日/非工作日、一天内的时间段、天气情况等。

$$\lambda = cN_f R_{apt} = cN_f R_{apt}(Isweekday, PoT, Weather)$$

其中 $Isweekday=\begin{cases}1(\text{工作日})\\0(\text{非工作日})\end{cases}$，$PoT$ 表示一天内的时间段，$Weather$ 表示该日的天气情况。同时，根据司机观测可得到前方排队等待的出租车数量 S，为实际出租车供给量。则需求大于供给时将成功接客并获得收益，否则等待无收益，这一概率为：

$$\mathrm{P}=\mathrm{P}(D>S) \text{ 即 } \mathrm{P}=\mathrm{P}\left(\frac{N_p}{n_p}>S\right)=\mathrm{P}(N_p>n_pS)$$

$$\text{求解得出 } \mathrm{P}=1-\sum_{k=1}^{n_pS}\frac{e^{-\lambda t_w}(\lambda t_w)^k}{k!}$$

2、接到客人的收益：

$$R = R_1(d_{a2c},f0,f1) + R_2(d_{c2r},f0,f2)$$

f0 为起步价，*f1,f2* 为每公里的加价（呈阶梯式），根据下表（以上海为例）我们可以知道 f0,f1,f2 的值与行驶总里程数，以及行驶时间点有关[2]。

3、整个过程的成本：

表 1 上海市出租车计价规则

	日间 (05:00 - 23:00)	夜间 (23:00 - 05:00)
0-3公里	14元	18元
3-10公里	2.5元/公里	3.1元/公里
15公里以上	3.6元/公里	4.7元/公里

等待过程：主要是时间成本，等待过程的空载燃油成本较小可以忽略。

单位时间成本用假设租车的单位时间费用来替代，即C^r

行驶过程：主要是燃油的费用，单位时间的燃油成本即C^f

总成本： $C=C^r(t_w+t_S)+C^f(t_Sv+d_{a2c}+d_{c2r})$

4、时间：

平均等待时间 t_w ：出租车在等待接客的过程，可以看成排队论模型，其中 μ表示服务率，与服务台（上车点）的数量和乘客上车所花时间有关

$$t_w=\frac{\lambda}{\mu(\mu-\lambda)}$$

市区找乘客的时间 t_S ：与市区的出租车需求量有关

总时间： $t=t_w+\frac{d_{a2c}}{v}+t_S+\frac{d_{c2r}}{v}$

5、行驶距离：

从机场接客的行驶距离 d_{a2c} 近似服从偏态分布，期望距离视为机场到市中心的距离，远距离行驶的概率明显大于近距离行驶的概率。

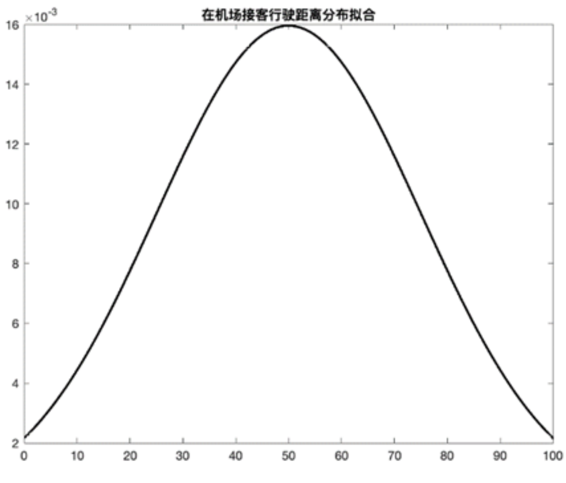


图 1 在机场接客行驶距离分布拟合

在市区接客的行驶距离 d_{c2r} 近似服从正态分布，期望距离小于机场到市中心的距离d_0，$\mathrm{E}(d_{c2r})$ 的值约为 $0.4d_0$。

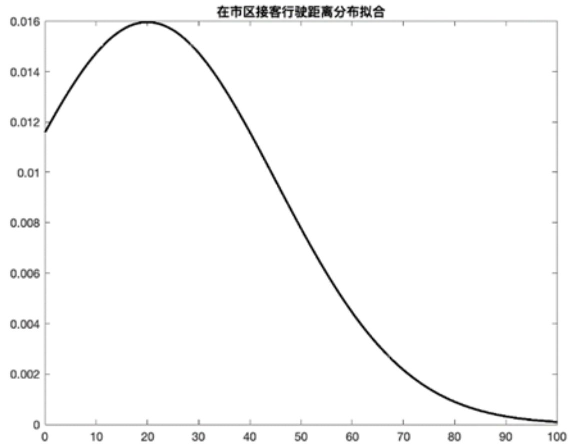


图 2 在市区接客行驶距离分布拟合

6、效用函数：

$$\mathrm{V}_{wait}=\frac{\mathrm{PR-C}}{t}+e \qquad e \text{ 为随机误差，服从 Gumbel 分布}$$

4. 1. 2 空载返回市区的效用函数

（1） 成本：

空载回市区的时间成本：$C^r\frac{d_{a2c}}{v}$

行驶过程燃油成本同第一种选择：$C^f(t_S v+d_{a2c}+d_{c2r})$

总成本：$C=C^r(\frac{d_{a2c}}{v}+t_S)+\ C^f(t_S v+d_{a2c}+d_{c2r})$

（2）收益：

$$R=\ R_2(d_{c2r},f0,f2)$$

（3）时间：

$$t=\ \frac{d_{a2c}}{v}+t_S+\ \frac{d_{c2r}}{v}$$

（4）行驶距离：

在市区接客的行驶距离 d_{c2r} 近似服从正态分布，$\mathrm{E}(d_{c2r})$ 的值约为 $0.4d_0$

（5）效用函数

$$\mathrm{V}_{dep}=\frac{\mathrm{R-C}}{t}+e \qquad e \text{ 为随机误差，服从 Gumbel 分布}$$

4. 1. 3 比较效用函数的期望

$$\mathrm{E}(\mathrm{V}_{wait})=\ \mathrm{E}\left(\frac{\mathrm{PR-C}}{t}+e\right)=\ \mathrm{f}\big(\lambda,t_w,\mathrm{S},f_i,d_{a2c},d_{c2r},t_s\big)$$

$$\mathrm{E}(\mathrm{V}_{dep})=\mathrm{E}\left(\frac{\mathrm{R-C}}{t}+e\right)=\ \mathrm{f}(f_i,d_{a2c},d_{c2r},t_s)$$

在效用函数的期望表达式中，出现的定量有：

n_p 平均每辆车承载乘客量　　　v　行车速度

μ 乘客上车的平均效率　　　$E(t_s)$ 在市区寻找乘客的期望

C^r C^f 租车/燃油单位成本　　　$E(d_{a2c})E(d_{c2r})$ 从机场/市区拉客行驶距离期望

司机观测到的因子有前方排队等待的出租车数量、航班数、当前时间、天气，则可得到以下变量：

S 前方排队等待接客的出租车数量

λ 到达乘车区的乘客数量，与航班数量和乘客在机场选择出租车出行的比率有关，该比率与天气、出行时间有关

t_w 等待时间，与 λ 和 μ 有关

f0 起步价与出行时间有关司机基于观察到的量，通过比较期望的效用函数，

若$E(V_{wait}) > E(V_{dep})$,则选择在机场等待载客；

若$E(V_{wait}) < E(V_{dep})$，则选择直接回市区

差值越大越倾向于做出期望较大一方的选择，决策函数定义为

$$\varphi = \frac{E(V_{wait})}{|E(V_{wait})| + E\left(V_{dep}\right)}$$

φ越大，司机越有理由选择在机场等待。

4.2 问题 2：基于浦东机场数据的模型案例

我们通过搜集上海浦东机场及上海市出租车的相关数据，试图通过模型给出机场出租车司机的选择方案。由题目中司机在某时间段抵达的航班数量和“蓄车池”里已有的车辆数是观测到的数据，我们以时间和观测的出租车数量为基准，输出司机的决策；其余的影响因素通过影响这两个观测数据来影响司机的决策。

4.2.1 数据展示

A. 出租车数据

我们获取了上海市 2018 年某日 11900 辆出租车在一天内行驶情况的数据，其中 3064 辆在其一天的行程中去过浦东机场。数据每秒更新一次（支撑材料中为其中 5 万条样本数据）。有如下有用信息字段：

表 2 出租车数据有用信息字段

车机号	载客状态	接收日期	GPS 时间	经度	纬度	速度	方向
	0：载客 1：空车	年/月/日	00:00:00			千米/小时	偏离正北方向角度

利用 Tableau 软件进行可视化，在本问题中研究对象为当日去过浦东机场的出租车（根据地图拾取坐标：31.105273 度≤经度≤31.177560 度，121.772187 度≤纬度≤121.831578 度），这些车辆一天内所行驶的路径如下图所展示：

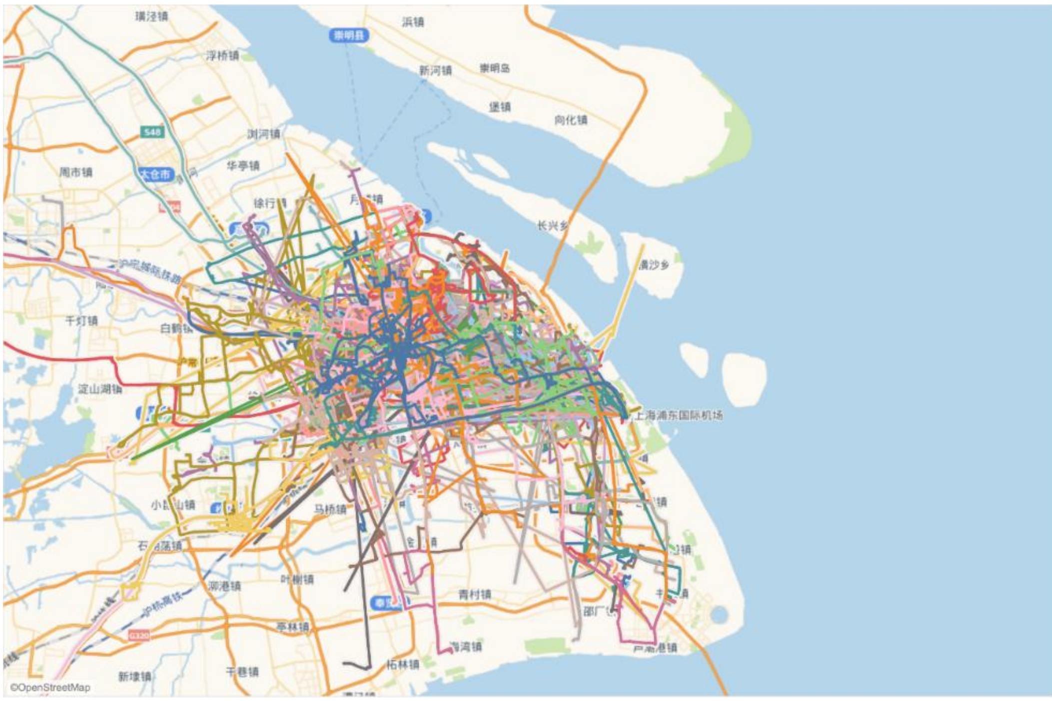
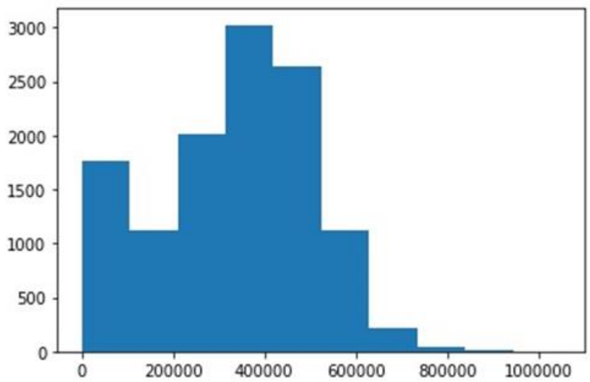


图 3 去过机场的出租车全天路径展示

通过计算得出所有出租车全天所行驶的距离分布直方图如左图所示，均值为 331 公里；为在当天曾去过浦东机场的出租车，这些出租车全天所行驶的距离分布直方图如右图，均值为 392 公里，与我们在问题一中对行驶距离的拟合相似程度极高。



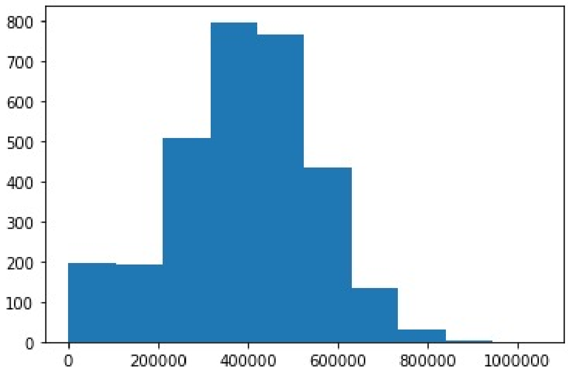


图 4 左：全部出租车行驶距离分布　　图 5 右：去过机场的车行驶距离分布

对于在浦东机场范围内的出租车，我们以一小时为时间间隔，统计其数量，该日内出租车数量变化曲线如图：

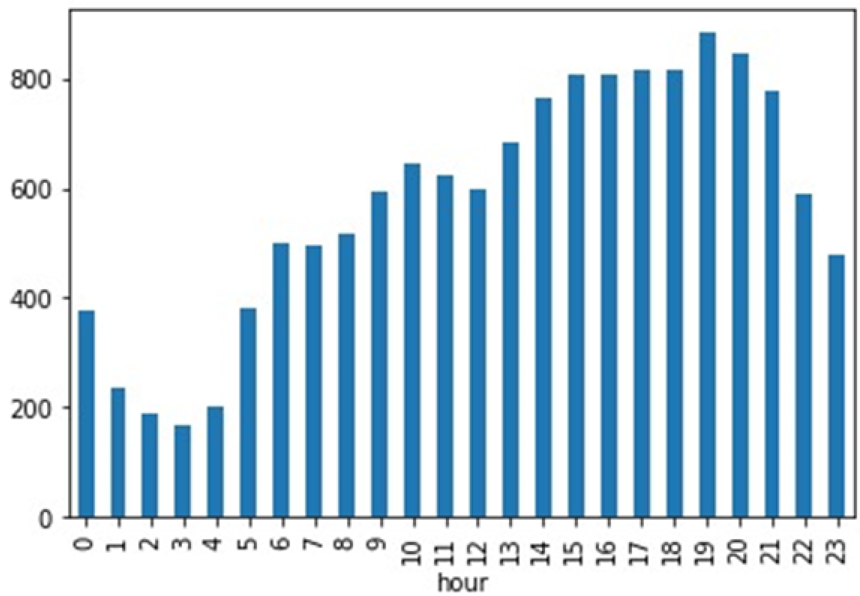


图 6 按小时统计在机场的出租车数量

高峰时段浦东机场范围内同时存在停滞或行驶的出租车数量超过 900 辆，根据我们的计算，空车率在每一时段均达到 80%以上。

B. 航班数据

我们获取了上海浦东机场 3 天的航班出发/到达数据，以一小时为时间间隔，统计到达的航班数量，可视化如图：

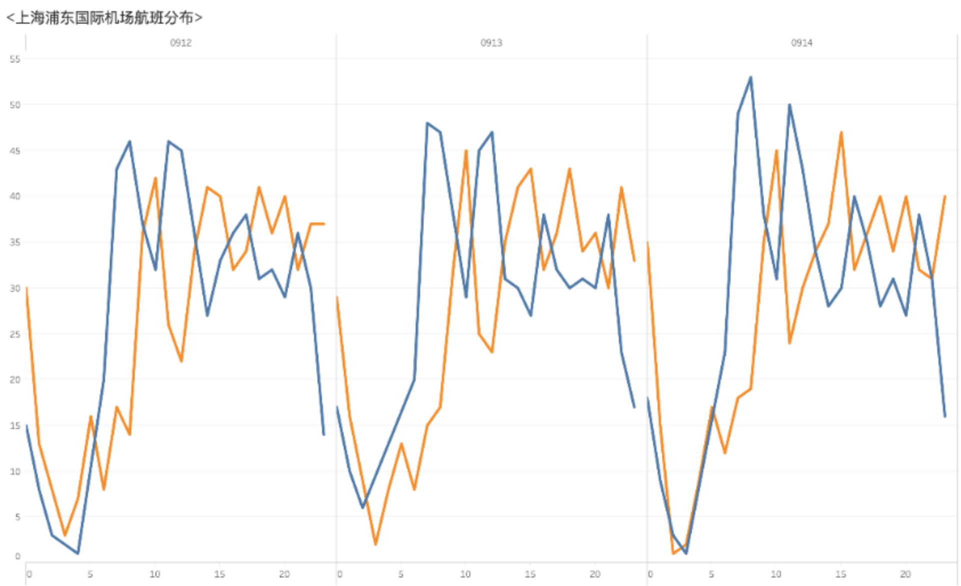


图 7 三天出发/到达航班数据

可以观察到三天的到达航班与出发航班虽时间变化均有一个稳定的分布，于是我们将三天的航班数量取平均值得到的分布情况作为依据，构建一日内航班数量的时间序列曲线。具体数值根据是否为节假日的情况再做调整，一般情况直接使用这里的平均值。

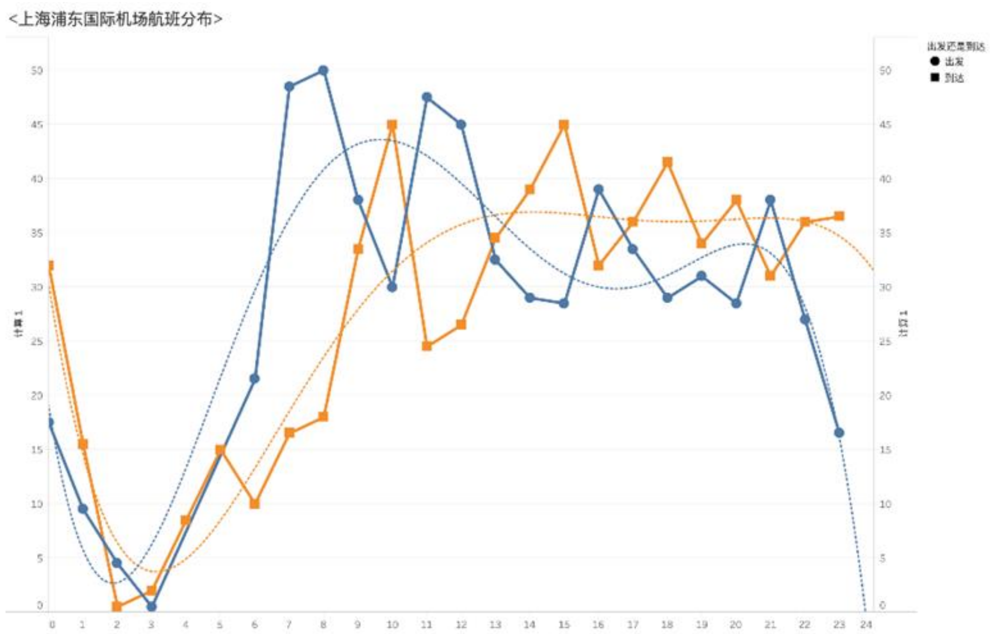


图 8 出发/到达航班数据分布

C. 其他参数的数据

每班次航班平均乘客数量为 150 人

每趟出租车的平均载客量为 n_p = 2.2 人

到达机场乘客选择出租车的平均比率 R_{apt} = 18%

$$\text{出租车计价方式}\begin{cases}\text{日间 f0} = 14, \text{f1} = \begin{cases}2.5\ yuan/km & 3-15\ \text{公里}\\ 3.6\ yuan/km & 15\ \text{公里以上}\end{cases}\\ \text{夜间 f0} = 18, \text{f1} = \begin{cases}3.1\ yuan/km & 3-15\ \text{公里}\\ 4.7\ yuan/km & 15\ \text{公里以上}\end{cases}\end{cases}$$

根据 A 中数据得行驶中的出租车平均速度 v = 31.8 *km/h*

一小时燃油费用 C^f = 0.5 *yuan/km**31.8 *km* = 15.6 *yuan*

时间成本（租车）$C^r = 20$ yuan

根据 A 中数据得市区找到乘客的平均时间 t_s = 0.2*h*

浦东机场上车点的个数 M = 6

服务率 μ 看作每小时可上车的人数 $\mu = \frac{3600\text{s/h}}{20\text{s/人}}$*6= 1080 人/h

每小时到达的人数 λ = $150N_f R_{apt} = 27N_f$

从机场出发到市区的距离期望 $E(d_{a2c}) = 50\ km$

从市区出发到任意位置的距离期望 $E(d_{c2r}) = 20\ km$

4.2.2 对φ的解释

基于以上数据，我们在其他情况均固定的情况下，对每个小时的决策函数φ进行求解，对于φ的值有如下解释：

（1） $\varphi = \frac{\mathrm{E}(\mathrm{V}_{wait})}{|\mathrm{E}(\mathrm{V}_{wait})|+\mathrm{E}(\mathrm{V}_{dep})}$ 的取值决定了实际情况下司机的选择，φ越大司

机越有理由选择在机场等待；

（2） 由于等待的结果有成功载客和不成功载客两种可能，所以 $E(V_{wait})$ 可能为负数，意味着等待的期望结果是不成功载客并付出了时间成本，此时的选择一定是选择离开，此时φ表现为负值，具体数值含义不大；

（3） 当 φ =0.5 时，两种选择带来的期望收益是一致，φ ＜ 0.5 时，选择离开，φ ＞ 0.5，选择等待；

我们认为每个出租车司机都面临会根据观测到的已有排队出租车数量来决策自己是否加入排队的队列中去，出租车每小时总体数量我们已经得到，于是我们通过不同出租车加入排队队列的比例，来计算当前的φ值。

4.2.3 模型数值求解

代入上述数据，最终得到下图，横轴 LX 表示前方排队的出租车数量，X 为排队等待的出租车占实时机场出租车总数的比例，纵轴 0-23 表示从该时刻开始的一个小时时间段，图中颜色深浅代表φ值的大小，方框内为每一时间段中对应的φ最接近 0.5（决策的临界值）的情况：

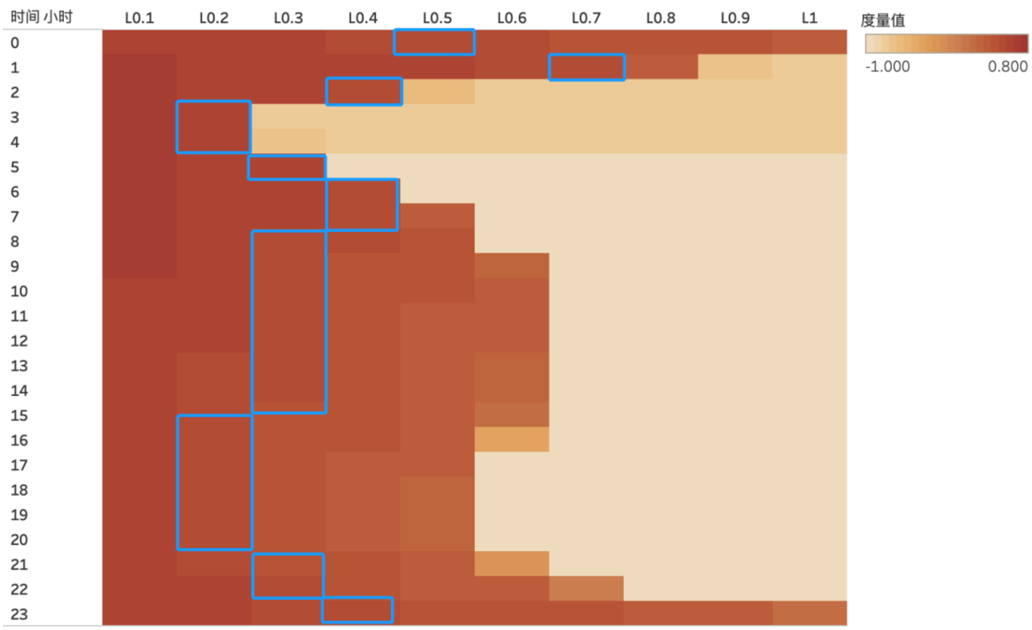


图 9 基于一日数据的决策情况

最终决策的结果当 24 小时已加入队列的出租车数量分别达到

177	153.3	72.8	31.6	36.2	100.8	183.2	184.4
191.2	167.4	180.9	174.6	171.3	195.9	219.3	232.2
155.4	153.6	156.2	168.4	159.8	222.9	168.3	182.8

之后，再来的出租车便不必再加入队列等待而是选择空载回到市区。

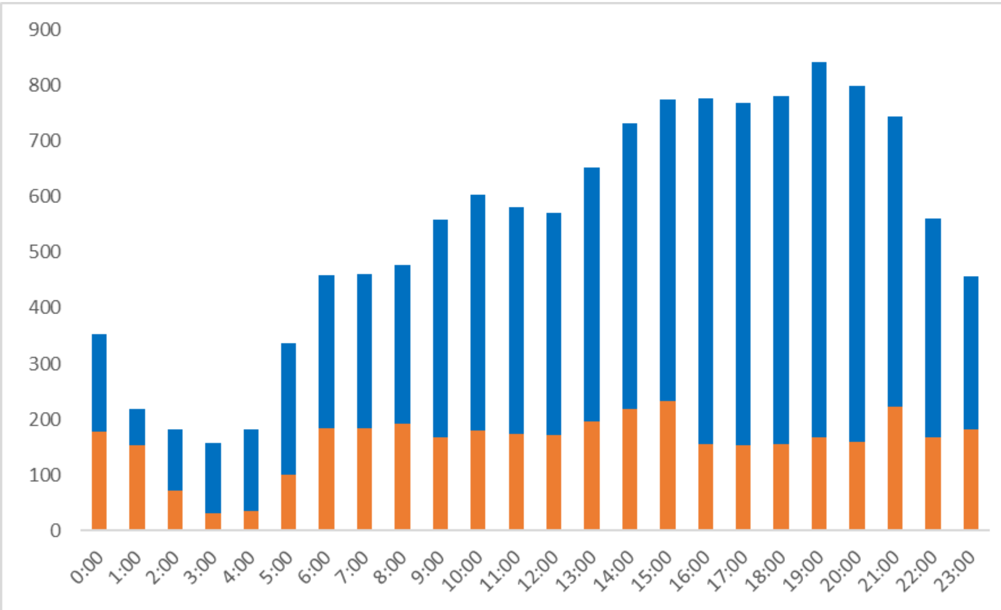


图 9 临界队列出租车数量与全部出租车数量

4. 2. 4 模型的普适性

我们解决了这一日的出租车司机面临空载和等待的选择问题，同时这一模型可以适应于多种影响因素变化的情况。以天气情况为例，得到下面的结果。

给予不同天气以分级，将晴天记为正常天气，阴雨、雷电记为恶劣天气分别为 *weather* 变量赋值，加入到算法中（详情见源代码）。天气越恶劣 *weather* 的值越大。天气变恶劣影响的因素有：乘客选择出租车的比率增大，机场出租车数量减少，到达航班数量减少，市区寻找乘客的时间变短。

取 weather = 1.1，得到如下结果，小图为其他因素相一致时正常天气情况下的φ值分布情况：

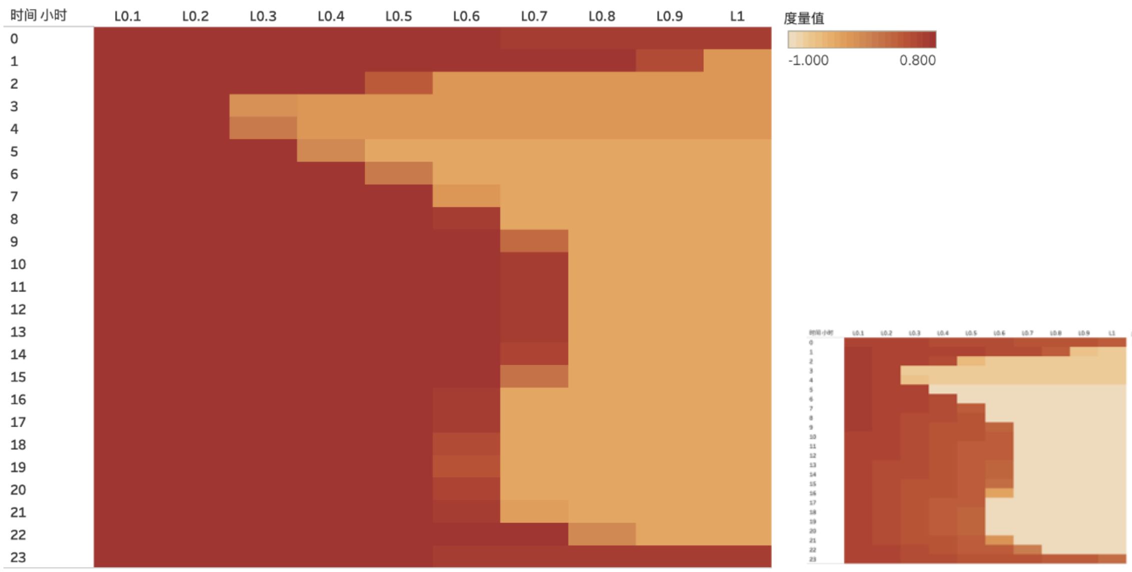


图 10 恶劣天气下决策情况

可以明显看出，φ值整体有所增大，说明在同等时间段和队列比例下，选择在机场等待的倾向性有所增大，选择在机场等待的队伍比例临界值变小。具体影响程度则需要获得实际数据经过模型训练得出。

增加“上车点”的数量同样会提高决策函数的值，由图分别为恶劣天气下 M=1 和 M=6 的情况，“上车点”数量增加时队列内出租车辆数的临界值增大。

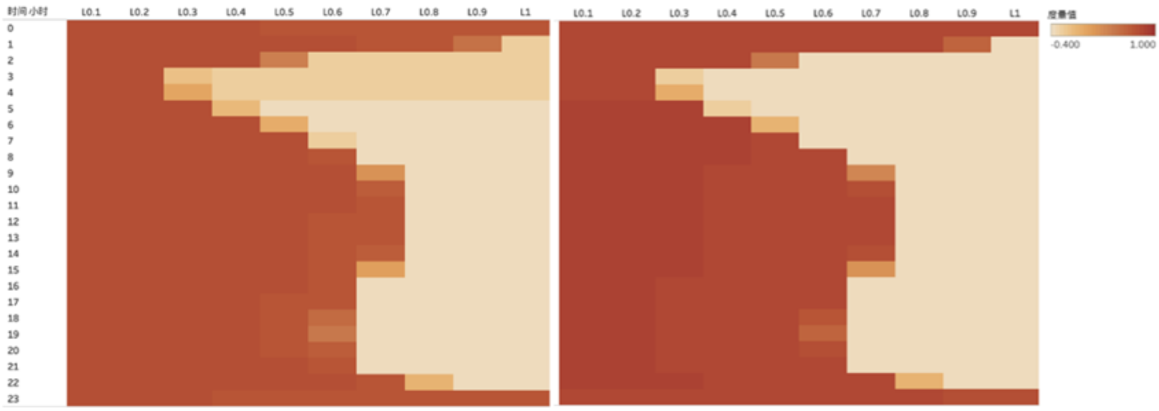


图 11 上车点变化时的决策情况对比

对于其他影响因素我们可做类似分析。

4.3 问题 3：上车点管理模型

4.3.1 上车点位置管理：

从合理安排出租车的角度出发，在现已有两条并行车道可利用的情况下，有以下三种较为经典且常见的“乘车区上车点”设置方案[3]。

1、 单车道依次发车，设置若干个相邻“上车点”

在依次发车的条件下，出租车“上车点”为每辆出租车提供的上客停靠位长约 5m/车，出租车停靠位依次相邻排列，占据道路一列车道，另一列并排车道为通过性行进车道，供其他机动车辆通过。出租车不能随意进出停靠位，必须按批次停车顺序依次进出，每一批次出租车数量等于设置的“上车点”数量。乘客被工作人员安排在各“上车点”排队，各“上车点”均成功完成接客后，第 1 辆车首先出发，然后依次第 2、第 3 辆车……驶出。

在该方法下，多个停靠位的候车乘客可以同时放置行李并上车；但前一辆车行李安置、上车所需的时间、发车时间都会限制后一辆车的出发时间，可能会造成时间极大程度的浪费，总乘车效率极低，但在这种方式下，依次行车能最大程度的保证车辆和乘客的安全。

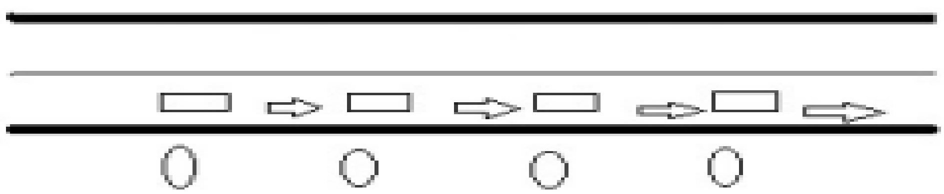


图 12 单车道依次发车示意图

2、单车道单独发车，设置若干个相邻“上车点”

在单独发车的条件下，因为要满足出租车在两车位变道的需求，出租车“上

车点”的停车位要比依次发车停车位距离更长，出租车上客停靠位约需长 8m/车，出租依次发车车才可以单独进出每 1 个上客泊位。另一并排车道为通过性行进车道。出租车在两并行车道间变道时，对通过性行进车道车辆的行驶影响较大，可能会造成该车道的暂时停滞，也可能会造成机动车磕碰等交通事故。

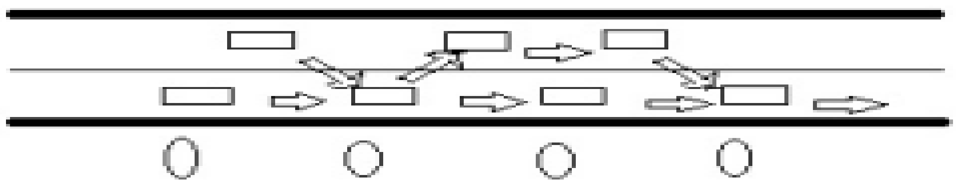


图 13 单车道单独发车示意图

乘客排队方式与依次发车相同，但每个“上车点”的候车乘客放置行李、上车所需的时间、发车时间互不影响；完成接客任务即可从另一车道驶离“上车点”，下一出租车也可从另一车道补充进入“上车点”，但在两车道变道期间，易与临近车道车辆产生碰撞造成交通事故，车辆安全存在一定风险。

3、并列双车道发车，两列均设置若干个相邻“上车点”

在并列车道马路两侧同时设置出租车“上车点”，车道之间可通过隔离设施进行阻隔，并建立天桥供乘客向另一侧车道候车处通行，两条车道均满足第一种情况的依次发车、出租车停车位长也约为 5m/车.。两车道出租车之间各自独立、互不影响，仅对同一车道后续车辆的驶入驶离产生影响。乘客排队方式与上述两种情况均相同，车辆和乘客的安全也能得到较好的保护。

易知，单车道依次发车的效率会远小于单车道单独发车和并列双车道发车，所以我们仅考虑计算后两种情况的通行效率，得出更优的方案。

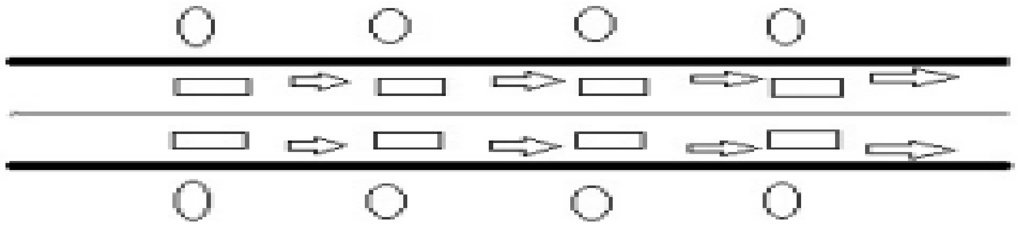


图 14 并列双车道发车示意图

首先，我们做出如下基本假设：

（1）乘客为单条排队队列候车，先到先服务，并听从现场工作人员安排；

（2）乘客的到达服从泊松分布，乘客上车时间服从负指数分布；

（3）排队乘客人数最少为 0，在 t 时间内最多为 N(t)；

（4）出租车补充“上车点”的空位是及时的，即队伍的移动时间不考虑。

（5）有足够的出租车为乘客提供服务；

乘客上车所需时间由行李安置时间、乘客上车所需的时间、司机发车时间所组成，如乘客上车时间累积曲线图所示：

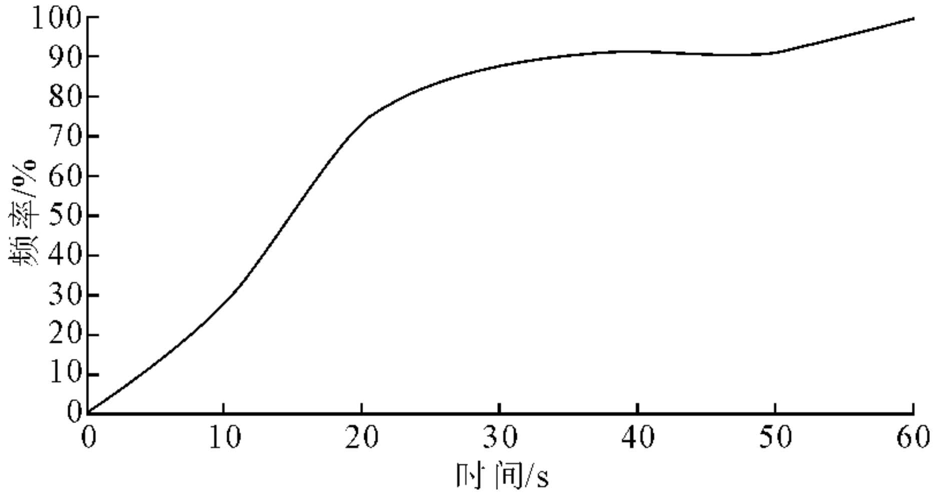


图 15 乘客上车时间累积曲线

约 30％的乘客上车时间小于 10s，约 40％的乘客上车时间介于 10s-20s，所需时间为 20s-60s 的乘客即为最后的 30％，因此我们可以由加权平均计算，得出上车时间均值约为 20s，该值可以作为单车道单独发车平均每车乘客上车所需的时间。但在并列双车道发车的情况下，一批次车移动的时间应该为上车最长时间的乘客所需的时间，我们将最后 30％的乘客上车时间做均值得出为 40s。

又由乘客的到达时间服从泊松分布，是在某时间内的平稳的泊松分布，由[4]知,平稳泊松过程可描述为: N(t)表示(0,t)内到达的顾客数,在(0,t)内排队的乘客数为 k 的概率是:

$$P\{N(\mathrm{t})-\mathrm{N}(0)=\mathrm{k}\}=\frac{\mathrm{e}^{-\lambda \mathrm{t}}(\lambda t)^{\mathrm{k}}}{k!}$$

其中:t ⩾ 0,k = 0, 1, 2,⋯ ,

λ: 平均单位时间内进入队伍的乘客数。

μ: 单位时间内服务的顾客数，为期望服务率；

s: 每种情况下出租车服务停靠位数目

Ls：队长 ，即队伍中的乘客数的期望值；

Lq：排队长 ，系统中排队等待乘车的乘客数期望值(其中 $Lq < Ls$)；

Ws：逗留时间，指一个乘客从开始排队到上车完成的全部时间的期望值；

Wq: 等待时间，指一个乘客在排队等待的总时间的期望值；

由排队论 $M/M/S$ 模型[5]结合二、三两方案可知：

服务强度：连续服务时间/总时间= ρ = $\frac{\lambda}{s\mu}$；（该服务强度为停靠位被使用的概率）

稳态概率：$P=1-\rho=1-\frac{\lambda}{s\mu}$；（该稳态概率为停靠位空闲的概率）

未完成上车服务的平均乘客数：$Ls=\frac{\lambda}{\mu\text{-}\lambda}$；

仍在队伍中排队的平均乘客数：$Lq=Ls\bullet\rho$；

乘客直到完成上车服务仍需时间：$Ws=\frac{1}{\mu\text{-}\lambda}$；

乘客排队等待时间：$Wq=Ws\bullet\rho$；

设每侧车道的停靠位为 p 个，则并列双车道停靠位有 2p 个。

表 3 单双车道服务数据

	出租车接客服务总时间(s/每辆)	出租车服务停靠位(个)
单车道单独发车(sim)	20	p
并列双车道发车(dou)	40	2p

则得到：

$$\frac{\mu-sim}{\mu-dou}=2$$

$$\frac{s-sim}{s-double}=\frac{1}{2},$$

综合上述表达式，则可得到

$$\frac{Ls\text{-}sim}{Ls\text{-}dou}=\frac{Lq\text{-}sim}{Lq\text{-}dou}=\frac{\mathrm{W}s\text{-}sim}{\mathrm{W}s\text{-}dou}=\frac{\mathrm{W}q\text{-}sim}{\mathrm{W}q\text{-}dou}=1+\frac{\mu\text{-}sim-\mu\text{-}dou}{\mu\text{-}sim-\lambda}$$

即；当 $\mu\text{-}sim<\lambda$ 时，单车道单独发车更能减少乘客排队等待的时间，队伍的长度更短，加快了总的乘车效率。反之，当 $\mu\text{-}sim>\lambda$ 时，并列双车道发车和单车道单独发车相比，将减少乘客排队等待的时间，更快地减少队伍的长度，加快了总的乘车效率，但实际上，$\mu\text{-}sim>\lambda$ 的实际意义为单位时间内单车道单独发车可服务乘客人数要大于单位时间内到达的排队乘客数，无论是单双排，每个乘客均可受到出租车的及时服务，此时的比较并没有可参考的现实价值：

根据 MATLAB 仿真模拟，我们假设：$p=4$、$\mu\text{-}sim=3$、$\mu\text{-}dou=1.5$、且 λ 为 2 和 3 时的情况并分别对单车道单独发车和双车道依次发车进行计算，并得出如下图的结果验证我们的模型结论。

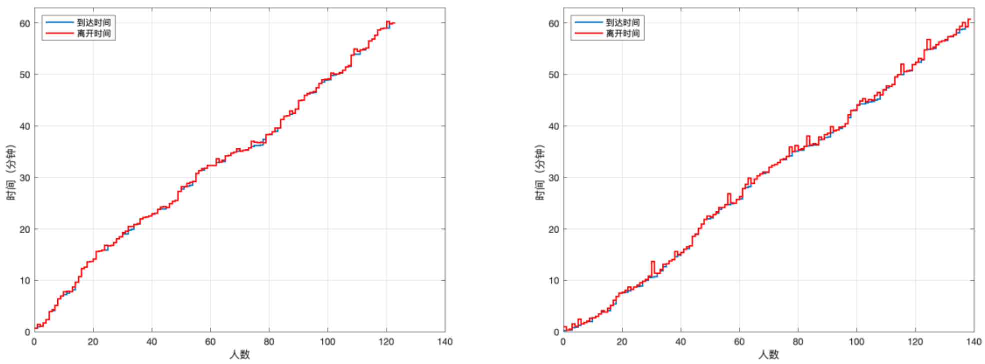


图 16 λ=2 左：单车道 右：双车道

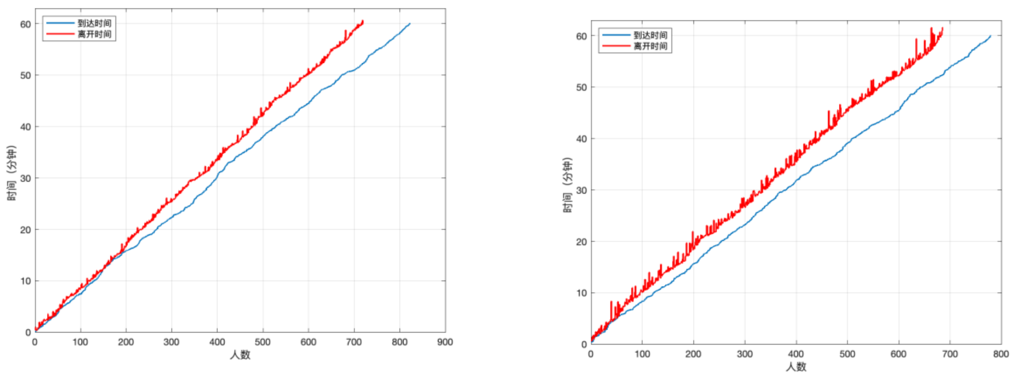


图 17 λ=13 左：单车道 右：双车道

4. 3. 2 上车点数量管理

设有 M 个“上车点”同时服务。对于“上车点”数量的设置，我们对排队系统费用决策模型进行优化。当“上车点”数量增多时，一方面乘客及出租车的等待时间将减短，提高了乘车效率；另一方面增加了成本，同时发生事故的可能性将会上升。

根据排队论的知识，M 个服务台同时工作时

$$L_s = L_q + M\rho = \frac{1}{M!}\frac{(M\rho)^M\rho}{(1-\rho)^2}P_0 + \frac{\lambda}{\mu}$$

乘客在付诸等待的时间成本为 $Z_{psg} = \alpha L_s$ ，α 为单位等待时间成本

出租车司机的时间成本为 $Z_{drv} = C_r\frac{1}{M}$ ，C_r 为单位服务平台司机时间成本

“上车点”成本为 $Z_{cond} = \beta M$ ，β 为单个服务台的服务成本和建设费用

发生事故成本期望为 $Z_{acci} = \gamma M$ ，γ 为单个服务台发生事故的成本期望

前两项成本与 M 呈负相关关系，后两项成本与 M 呈正相关关系，当满足所有成本之和最小时，此时的 M 为最优的“上车点”个数，即

$$\mathrm{Min}\, Z(M) = Z_{psg} + Z_{psg} + Z_{cond} + Z_{cond}$$

$$\text{即} \begin{cases} Z(M) \le Z(C+1) \\ Z(M) \le Z(C-1) \end{cases}$$

$$[L_s(M) - L_s(M+1)]\, M(M+1) + \frac{C_r}{\alpha} \le \frac{\beta+\gamma}{\alpha} M(M+1)$$

$$[L_s(M) - L_s(M-1)]\, (M-1)M + \frac{C_r}{\alpha} \le \frac{\beta+\gamma}{\alpha} (M-1)M$$

实际优化过程中，可以通过得到 $\frac{C_r}{\alpha}$ 与 $\frac{\beta+\gamma}{\alpha}$ ，最终确定“上车点”的数量 M.

4.3.3 等待乘客管理

配合最优化的出租车安排方案，我们再从合理安排乘客的角度出发考虑：

在同一车道设置相邻“上车点”时，由于后一辆车会受到前一辆车的影响，停靠时间会大于等于前一辆车的停靠时间，我们也知道，多个人乘车由于其行李人数因素的影响，上车时间要长于一人乘车花费的时间，且人数越多需要出租车停靠时间也越长。因此，为了使同时停靠的出租车驶出停靠位的时间最短，我们需要将乘客数较少的需求指派给靠前的出租车拉载，一行乘客数较多的指派给靠后的出租车拉载，使得一批次出租车停靠总时间最短，总的乘车效率最高。

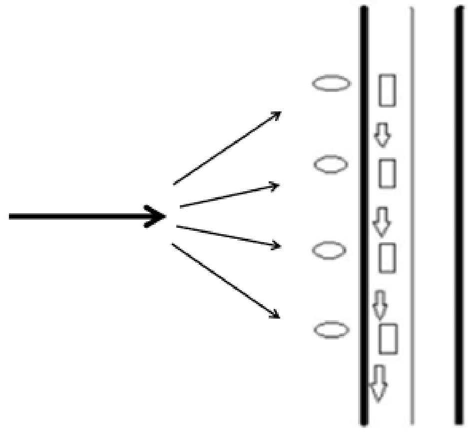


图 18　等待乘客管理模式示意图

因此我们建议在排队处设置工作人员，乘客在现场工作人员有上述原理管理的情况下，根据指示到相应的“上车点”派对等候上车。

4.3.4 小结

根据实际机场出租车实际情况，（1）判断μ与λ的大小关系，相应的实施单车道单独发车或并列双车道发车方案，（2）根据各成本因子的比例大小，设定最优的“上车点”数量，（3）工作人员根据同批乘客的乘车人数，合理安排对应出租车“上车点”位置，使得在保证车辆和乘客安全的条件下，能够使得总的乘车效率最高。

4.4 问题 4：均衡收益模型

现实生活中，出租车将客人送达浦东机场后，由于其回市区的空载成本较高，多数的司机会做出排队接客人回市区的选择。但出租车流量大，出租车排队载客等待时间长，接到短途客人可能会做出拒载和提出不合理要求等违规行为。为了为乘客搭乘出租车提供保障且杜绝司机拒载短途乘客，缓解机场出租车保障压力，稳定机场的管理秩序，我们需要消除因司机运送客人距离差距过大而产生的不公平性。

因此，该问题中我们要尽量实现每辆出租车的受益均衡，实现的方式为给予短途载客的出租车一定的优先权，其优先权的表现即为缩短排队等待的时间。且在乘客乘车需求随时间变化不断增加出租车需求的过程中，我们通过动态设定返回出租车的等待时间，来均衡任务。

4.4.1 出租车一日期望收益

通过对问题 1 日乘客总数 N_p 的分析，我们可以得到全天需要的出租车辆次 D，以及全天服务机场乘客的出租车数量S_{all}。D 为所需出租车的辆次，每一个辆次都将对应一个任务，由此我们将 $\mathscr{D}$ = $\{d_1, d_2, d_3, \dots，d_D\}$看做一日的任务集（d 本身即是任务距离），任务集对应开始时间集 $\mathscr{T}$ = $\{t_1, t_2, t_3, \dots，t_D\}$. 从机场出发的每位乘客的行驶距离$d_j$，同问题 1，与 d_{a2c} 有相同的分布. 收益为：

$$R = R_2(d_{a2c}, f0, f1) = f0 + (d_{a2c} - s_0)f1$$

f0,f1 由行驶距离及行驶时间（日间、夜间）决定。

由此全天所有出租车送客所有的收益可以表达出来为：

$$\mathrm{B} = \mathrm{D}f_0 + \sum_{i=1}^{D}(d_j - s_0)f2$$

其中，s_0 为出租车起步价包含里程，每辆车期望的平均收益即为 B/S_{all}

4.4.2 每辆车的实际收益

设 $x_j^{(k)}$ 表示第 k 辆车（k = 1,2,3,⋯,S_{all}）是否实际拿到第 j 个任务d_j （j =1,2,3,⋯,D）

$$x_j^{(k)} = \begin{cases} 1, & k\text{车接到任务}j \\ 0, & k\text{车未接到任务}j \end{cases}$$

则 k 车的收益为

$$z_k = (\sum_{j=1}^{D} x_j^{(k)} d_j - \sum_{j=1}^{D} x_j^{(k)} s_0)f2 + \sum_{j=1}^{D} x_j^{(k)} f0$$

4.4.3 每辆车的实际成本

出租车的成本分为在途的燃油费用（包括载客和空载两种情况下的），空载的时间成本（包括空载返回机场路程中和在机场等待过程中）。由于我们希望任

务尽可能均衡，则已达到送客路途几乎相等的目标；并且我们假定出租车送完客一定回到机场，则空载返回机场路程也可以看作相等。故存在差别的为等待的时间成本。

由于任务距离存在长短途之分，完成短途任务的出租车司机回到机场后需要保证有时间条件能够进行后续的任务，即给予一些优先权，保证接到尽可能均衡的任务。

第 k 辆车（k = 1,2,3,⋯, S_{all}）在完成第 j 个任务d_j （j =1,2,3,⋯,D）后所需要的等待时间为 $wt_j^{(k)}$，则一整天的等待时长为

$$Wt^{(k)} = \sum_{j=1}^{D} x_j^{(k)} wt_j^{(k)}$$

4. 4. 4 **模型建立：**

此模型为多目标优化模型，优化的目标分别为收益和等待时间尽可能均衡。对于收益我们已获得平均收益值 B/S_{all}，则我们希望全部出租车的收益尽可能靠近 B/S_{all}，波动范围最小；即

$$\mathrm{Min} \sum_{k=1}^{S_{all}} (z_k - \frac{B}{S_{all}})^2$$

$$\sum_{k=1}^{S_{all}} x_j^{(k)} = 1 \quad (\forall \mathrm{j} = 1,2,3, \ldots, \mathrm{D})$$

通过对等待时间的分析，这是保证任务均衡分配后能够完成的重要条件。在没有做优化时，所有出租车回机场接客都从队尾开始排队，跑长途的出租车回到机场等待接客的次数少，因而排队总时间少，而跑短途的出租车需要多次回到机场等待接客，排队等待时间将影响后续接客任务，因而带来更小的收益。如果能够适当缩短已跑短途的出租车的等待时间，虽然多次等待但尽量保持排队等待总时间与长途无太大差别，那么将有条件实现最终收益的均衡。

$$\mathrm{Min} \ \frac{max_{k=1}^{S_{all}} Wt_k - min_{k=1}^{S_{all}} Wt_k}{max_{k=1}^{S_{all}} Wt_k}$$

我们以平均等待时间$t_w = \frac{\lambda}{\mu(\mu-\lambda)}$为基础，结合所跑的路程，去设定每次回到机场的等待时间 $(wt)_j$ ，根据 Chebyshev 不等式：

$$t_w - 3\sigma \le \frac{\sum_{j=1}^{D} (wt)_j}{S_{all}} \le t_w + 3\sigma$$

同时我们希望所跑距离较短的出租车等待的时间相应变短，及在规划$(wt)_j$时需要考虑该辆车已经跑过的距离 $\sum_{i=1}^{j-1} d_i x_i^{(k)}$ ，已跑距离越长，规划等待时间

越长，已跑距离越短，规划的等待时间应越短。根据距离获得的收益得到：

$$(wt)_j^{(k)}c^r=\left[\sum_{i=1}^{j-1}d_ix_i^{(k)}-\sum_{i=1}^{j-1}x_i^{(k)}s_0\right]f2+\sum_{i=1}^{j-1}x_i^{(k)}f0+\ \varepsilon$$

为简化模型，我们直接用出租车行驶距离的平均单价替代分段式计费方式，（如问题 2 中我们所使用的上海距离-单价关系表）

$$(wt)_j^{(k)}c^r=\sum_{i=1}^{j-1}d_ix_i^{(k)}f_{mean}+\varepsilon$$

$$（\forall \mathrm{j}=1,2,3,\dots,D,\forall k=1,2,3,\dots,S_{all}）$$

从上述模型中得到每辆车的任务集 $D_k=\{d_{k1},d_{k2},\dots,d_{km_k}\}$，我们知道要想接到这些顾客还必须满足，对每次完成订单回来的时间要早于任务集中下一个订单的开始时间，于是有

$$t_{ki}+\frac{d_{ki}}{v}+(wt)_{ki}\le t_{k(i+1)}\quad（\forall \mathrm{i}\ =\ 1,2,3,\dots m_k）$$

对于第 k 辆车来说，用原始表达式表达即为

$$t_j+x_{j+1}^{(k)}\left[\frac{d_j}{v}+(wt)_j\right]\le t_{j+1}\quad（\forall \mathrm{j}=1,2,3,\dots,D,\forall k=1,2,3,\dots,S_{all}）$$

最终的优化模型为：

$$\mathrm{Min}\sum_{k=1}^{S_{all}}(z_k-\frac{B}{S_{all}})^2$$

$$\mathrm{Min}\ \frac{max_{k=1}^{S_{all}}Wt_k-min_{k=1}^{S_{all}}Wt_k}{max_{k=1}^{S_{all}}Wt_k}$$

$$\sum_{k=1}^{S_{all}}x_j^{(k)}=1\quad(\forall \mathrm{j}=1,2,3,\dots,\mathrm{D})$$

$$t_w-3\sigma\le\frac{\sum_{j=1}^{D}(wt)_j}{S_{all}}\le t_w+3\sigma$$

$$(wt)_j^{(k)}c^r=\sum_{i=1}^{j-1}d_ix_i^{(k)}f_{mean}\quad(\forall \mathrm{j}=1,2,3,\dots,D,\forall k=1,2,3,\dots,S_{all}）$$

$$t_j+x_{j+1}^{(k)}\left[\frac{d_j}{v}+(wt)_j\right]\le t_{j+1}\quad（\forall \mathrm{j}=1,2,3,\dots,D,\forall k=1,2,3,\dots,S_{all}）$$

最终输出 S_{all} *D 个 0-1 变量的结果 $x_j^{(k)}$，以及 D - S_{all} 个等待时间 $(wt)_j$。

4.4.5 模型管理优化

在已建立模型的基础上，倘若我们要将模型尽可能完美的呈现并得到使用，我们将继续提出以下建议：

1、建立机场出租车自动化管理系统

为出租车安装 GPS 系统，并在机场附近建设地感线圈、摄像系统、数据传输

系统等基础设施，使得机场实现对经过在机场载客的出租车进行实时信息追踪。机场出租车交通管理员可由此信息合理安排出租车的蓄车数量与乘客调度。且 GPS 系统可以对出租车的“ 长途”、“短途”进行识别判断与实时监控 。将出租车实时 GPS 位置信息与该车“长途”、“短途”的实际里程相结合，判断该出租车是否满足限制标准，并为该车下一次返回机场分配的“短途”或“长途”选择给出合理判断。

2、安装旅客排队智能系统，并与出租车调配系统结合

在排队候车的站点设置识别系统，自动统计当前排队人数，为旅客分析所需要的排队等待时间，另一方面，收集不同乘客的旅程距离需求，为乘客匹配适合其旅途的出租车，尽量减少司机因运送客人距离差距过大而产生的不公平性。

3、增加非法运行识别系统

在公共交通停运后，机场会出现非法营运车辆，直接拦截乘客，误导乘客接受不正规交通服务，对乘客人身安全造成巨大隐患，并不利于稳定机场的正常秩序。因此，我们可以在出租车通道口安装摄像识别分析设施等，记录在特定时间段多次重复出现的人员车辆，一旦发现可疑人员，立刻对其进行跟踪与报备，将非法运营人员绳之以法，杜绝影响正常机场秩序与乘客权利的违规揽客行为。

合理安排使得出租车的收益尽量均衡的方案不仅需要建立合适的模型，也需要善于运用创新技术以实现精准的管理与监督体系，使得这些出租车的收益尽量均衡，且以司机与乘客的需求为导向不断改进创新，促进出租车运营管理环境的不断完善。

五、模型评价

5.1 问题一模型评价

正确判断了问题的状态，并将其视为马尔科夫过程的一个现在的状态点去分析，该方法对过程的状态预测效果良好。影响出租车司机选择的变量因素有许多，但该模型将它们都融入司机离开或等待的效用函数的期望表达式，最终选择能带来更大效用期望的选择即可。但模型假设的一个完整的马尔科夫过程可能会难以完成，如司机在路上接到其它合适的订单从而改变其运行状态，且司机做出等待和离开的判断也极大程度与司机个人喜好、经验有关，均存在不确定性。

5.2 问题三模型评价

模型首先分析出了三种经典的在车道旁设置“上车点”的方案，然后基于排队论的 M/M/S 理论对三种方案进行分析，得出在不同情况下最优化的“上车点”方案，将复杂问题简单经典化、数学表达式化，更具有说服力。

然而该 M/M/S 模型要求乘客的到达服从泊松分布，我们据实际可知，乘客应是与航班的到达密切相关的，且呈批次到达排队，满足的乘客到达排队分布难以确定，只能假设在某一段时间内服从 λ 的泊松分布。且出租车补足空位以及在乘客乘车区域的移动也是需要时间的，模型却将它们忽略来得到更简单的方法。

5.3 问题四模型评价

模型使用最小二乘法与运筹学理论将及其抽象化文字化的管理安排转化为最小化全部出租车的收益的实际值与理论值差的平方和公式的问题，并且根据实际情况列出不等式对目标问题公式进行约束，进而得出使出租车收益尽量均衡的可行的优先安排方案。

实际上市区各地距离机场路程相差甚远，但由于其计算难以量化且过于复杂，我们则将市场与机场看作两个定点，距离也因此一定，因此该模型不能十分准确

的模拟真实情形。且对完成载客任务回到机场的司机，优先权的给予存在极大的不确定性，难以定量且司机的等待时间仍然会受到后来司机的上一单载客路程的影响，等待时间不断变化，可能会与最初的预期存在偏差。

参考文献

[1] 鲁子爱，排队论在港口规划中的应用[J]，水运工程:12-16，1997.08。

[2] 上海机场（集团）有限公司，https://www.shanghaiairport.com/，2019.09.15。

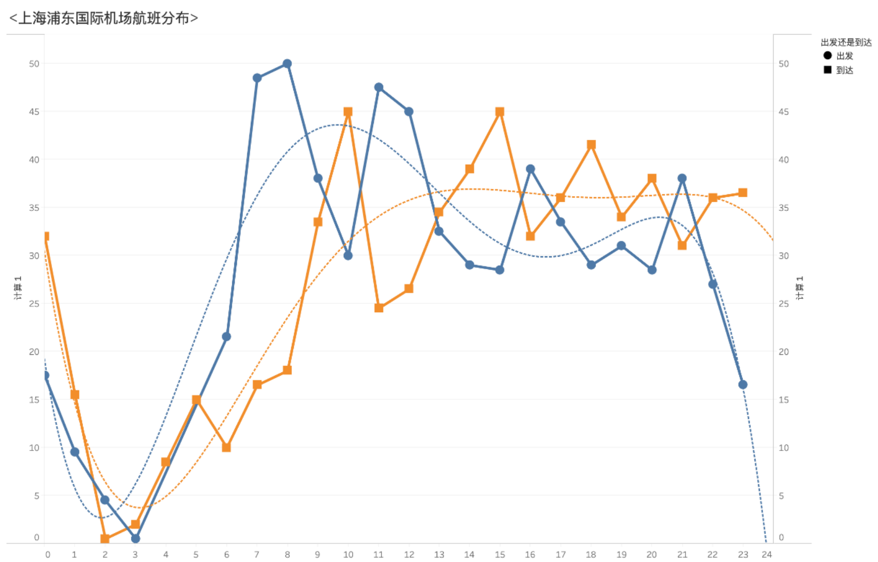
[3] 吴娇蓉、李铭、梁丽娟，综合客运枢纽出租车上客点管理模式和效率分析[J]，交通信息与安全:18-23,2012,30(04)。

[4] 熊光楞、肖田元、张燕立，连续系统仿真与离散事件系统仿真 [M]，北京:清华大学出版社，1999。

[5]徐啊俊,陈志荣，基于 M/M/S 的高校餐厅配餐模型研究与应用[J].信息技术:29-32+36,2019,43(08)。

附录

1. 浦东机场航班拟合曲线



出发曲线趋势线模型

针对平均航班数量(给定 时间 小时) 计算了 5 度的多项式趋势模型。 模型可能在 p <= 0.05 时有意义。 因素 出发还是到达 在 p <= 0.05 时可能有意义。

模型公式:	出发还是到达*(时间 小时^5 + 时间 小时^4 + 时间 小时^3 + 时间 小时^2 + 时间 小时 + 截距)
已建模的观察值数:	46
已筛选的观察值数:	0

模型自由度:	12
残差自由度(DF):	34
SSE (误差平方和):	1520.04
MSE (均方误差):	44.7071
R 平方值:	0.808441
标准误差:	6.68634
p 值(显著性):	< 0.0001

方差分析:

字段	DF	SSE	MSE	F	p 值
出发还是到达	6	1118.7986	186.466	4.17084	0.0030221

趋势线模型

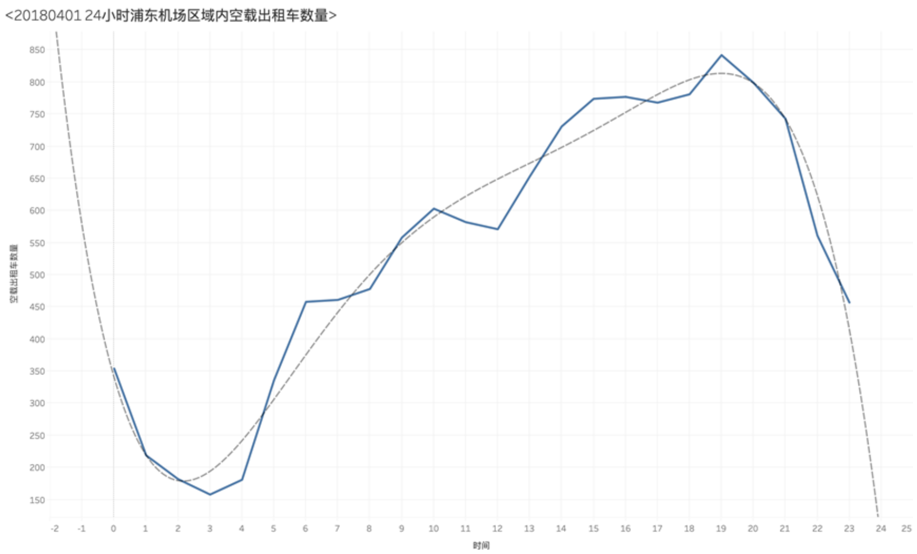
针对 平均航班数量 (给定 时间 小时) 计算了 5 度的多项式趋势模型。 模型可能在 p <= 0.05 时有意义。 因素 出发还是到达 在 p <= 0.05 时可能有意义。

模型公式:	出发还是到达*(时间 小时^5 + 时间 小时^4 + 时间 小时^3 + 时间 小时^2 + 时间 小时 + 截距)
已建模的观察值数:	46
已筛选的观察值数:	0
模型自由度:	12
残差自由度(DF):	34
SSE (误差平方和):	1520.04
MSE (均方误差):	44.7071
R 平方值:	0.808441
标准误差:	6.68634
p 值(显著性):	< 0.0001

方差分析:

字段	DF	SSE	MSE	F	p 值
出发还是到达	6	1118.7986	186.466	4.17084	0.0030221

2. 浦东机场出租车数量拟合曲线



趋势线模型

针对 空载出租车数量 (给定 时间) 计算了 5 度的多项式趋势模型。 模型可能在 $p<=0.05$ 时有意义。

模型公式:	(时间^5 + 时间^4 + 时间^3 + 时间^2 + 时间 + 截距)
已建模的观察值数:	24
已筛选的观察值数:	0
模型自由度:	6
残差自由度(DF):	18
SSE (误差平方和):	33402.1
MSE (均方误差):	1855.67
R 平方值:	0.969156
标准误差:	43.0775
p 值(显著性):	< 0.0001

3. 决策函数 MATLAB 程序源代码：

```
Fi = [];
weather_factor = 1.1; % >1 bad condition
Rapt = 0.18 * weather_factor; %机场乘客选择乘坐出租车的比例
n_p = 2.2;%每辆车平均运载人数
c = 150;%平均每架飞机所载乘客
```

```
d_0 = 50;%机场到市中心的距离
M = 6;%服务台数量
t_on_unit = 20;%上车平均所需时间（秒）
v = 31.8;%出租车匀速行驶的速度
c_f = 0.5;%单位时间的燃油成本
c_r = 20;%租车的单位时间费用
t_s = 0.2/weather_factor;%市区找乘客的时间


tmp = normrnd(0.4*d_0, 25);
while tmp < 0
    tmp = normrnd(0.4*d_0, 25);
end
d_c2r = tmp;%从市区到任意地点出租车行驶的距离

tmp = normrnd(d_0, 25);
while tmp < 0
    tmp = normrnd(d_0, 25);
end
d_a2c = tmp;%从机场到市区出租车行驶的距离

for x = 0:23%时间
    for y = 0.1:0.1:1
        Np = 0:0.01:100;%到达乘车区的顾客数
        S = (-0.00498152*x.^5 + 0.286286*x.^4 ...
            + -6.10099*x.^3 + 57.1602*x.^2 ...
            + -173.203*x + 341.363).*y ./ weather_factor;%司机可观测前方排队等待的出租车数量
        Nf = (-0.000241632*x.^5 + 0.0170489*x.^4 ...
            + -0.444053*x.^3 + 5.03076*x.^2 ...
            + -20.4342*x + 30.4298) ./ weather_factor;%机场航班到达量

        mu = 3600/t_on_unit*M;%乘客上车的平均效率
        lambda=Nf*Rapt*c;%单位时间内到达乘车区的乘客数量
        t_w = S/mu; %出租车的等待时间

        P = 1 - poisscdf(n_p*S, lambda)

        % plot(x,Np,'k','linewidth',2);
        %
        % plot(x,d_c2r,'k','linewidth',2);
        % title('在市区接客行驶距离分布拟合');
        % plot(x,d_a2c,'k','linewidth',2);
        % title('在机场接客行驶距离分布拟合')
        isNight = 1 .* (x>=23|x<=4);%是否是夜间
```

```
        %在机场等待的情况下各项数据
        R1 = pricing(d_a2c,isNight) + pricing(d_c2r,isNight);%收入
        C1 = c_r*(t_w + t_s) + c_f*(t_s*v + d_a2c + d_c2r);%成本
        t1 = t_w + d_a2c/v + t_s + d_c2r/v;%时间

        %回市区接客的情况下各项数据
        R2 = pricing(d_c2r,isNight);
        C2 = c_r*(d_a2c/v + t_s) + c_f*(t_s*v+d_a2c+d_c2r);
        t2 = d_a2c/v+t_s+d_c2r/v;

        %两类情况下的效用函数
        V1 = (P*R1 - C1) / t1
        V2 = (R2-C2) / t2

        Fi = [Fi V1/(abs(V1)+V2)]
    end
end

result = reshape(Fi, [10 24])'
% plot(0:23, Fi)

function R = pricing(distance, isNight)%出租车定价
    if isNight
        unit_p = 18./distance .*(distance>=0&distance<=3)...
        +   (18+3.1*(distance-3))./distance .*(distance>3&distance<=15)...
        +   (55.2+4.7*(distance-15))./distance .*(distance>15);
    else
        unit_p = 14./distance .*(distance>=0&distance<=3)...
        +   (14+2.5*(distance-3))./distance .*(distance>3&distance<=15)...
        +   (44+3.6*(distance-15))./distance .*(distance>15);
    end
    R = unit_p * distance;
End
```

4. 出租车数据分析 Python 源代码

```
get_ipython().run_line_magic('matplotlib', 'inline')
import pandas as pd
import seaborn as sns
import matplotlib.mlab as mlab
import matplotlib.pyplot as plt
import numpy as np
```

```
data = pd.read_csv('all.txt', sep='|', header=None)
data.head()

data.describe()

data.columns = ['ID','Control','Status1','Service','Lamp','Status2','Status3','Meaningless','Date','GPSDate','Lon','Lat','Velocity','Direction','Satellite','Meaningless']

data_airport = data[(data["Lon"]>=121.772183) & (data["Lon"]<=121.831578) & (data["Lat"]<=31.177560) & (data["Lat"]>=31.105273)]

data_airport["Date"] = pd.to_datetime(data_airport["Date"])
data_airport["GPSDate"] = pd.to_datetime(data_airport["GPSDate"])

data_test = data_airport
data_test["hour"] = data_test.GPSDate.dt.hour

hour_taxi = data_test.groupby(by="hour").ID.nunique()

hour_taxi.plot.bar()

data_test.groupby(by="hour").Service.mean()

data_test_whole = data
data_test_whole["Date"] = pd.to_datetime(data_test_whole["Date"])
data_test_whole["GPSDate"] = pd.to_datetime(data_test_whole["GPSDate"])
data_test_whole["hour"] = data_test_whole.GPSDate.dt.hour
data_test_whole.groupby(by="hour").Service.mean()

data_test.groupby(by="ID").Service.mean()

hour_taxi = data_test[data_test['Service'] == 1].groupby(by="hour").ID.nunique()

hour_taxi.plot.bar()

testing_all = data_test_whole.groupby(by="ID").Velocity.sum()*3.6

testing_all.hist(grid=False)

airport_taxi_id = data_airport["ID"].unique()

airport_taxi_whole_city = data_test_whole[data_test_whole["ID"].isin(airport_taxi_id)]
```

```
testing = airport_taxi_whole_city.groupby(by="ID").Velocity.sum()*3.6

testing.hist(grid=False)

np.mean(data_test_whole.groupby(by="ID").Velocity.sum()*3.6)

np.mean(airport_taxi_whole_city.groupby(by="ID").Velocity.sum()*3.6)

sns.distplot(testing, bins=10)

sns.distplot(testing_all, bins=10)

data[data["Velocity"] != 0]["Velocity"].mean()
```

5. 六服务台一日决策函数输出：

	L0.1	L0.2	L0.3	L0.4	L0.5	L0.6	L0.7	L0.8	L0.9	L1
0:00	0.80395929	0.80199603	0.80003479	0.79807558	0.79611839	0.79416322	0.79221007	0.79025893	0.78830981	0.78636263
1:00	0.80466065	0.80339754	0.80213528	0.80087386	0.79961327	0.79835352	0.79709406	0.79303555	0.59557212	-0.3742292
2:00	0.80489258	0.80386113	0.80283023	0.79968162	0.42862916	-0.3924341	-0.3940507	-0.3942457	-0.3944385	-0.3946291
3:00	0.80481062	0.80223617	-0.2606131	-0.3935159	-0.3937335	-0.3939478	-0.3941594	-0.3943685	-0.394575	-0.394779
4:00	0.80453782	0.80303371	-0.0119987	-0.3937285	-0.3939956	-0.394258	-0.3945164	-0.3947709	-0.3950216	-0.3952686
5:00	0.84788518	0.84636867	0.84463522	-0.2565274	-0.5440832	-0.5444212	-0.5447522	-0.5450769	-0.5453954	-0.545708
6:00	0.84754087	0.84567922	0.84381512	0.84180932	-0.0281369	-0.544869	-0.5452656	-0.5456513	-0.5460282	-0.5463966
7:00	0.84721081	0.84501816	0.84282211	0.84062265	0.8377267	-0.3871102	-0.545743	-0.5461836	-0.5466125	-0.5470302
8:00	0.84691814	0.84443185	0.84194118	0.83944613	0.83694667	0.81872135	-0.5394611	-0.5466415	-0.5471133	-0.5475714
9:00	0.84667262	0.84393991	0.84120191	0.83845861	0.83570999	0.83291783	0.346127	-0.5470106	-0.5475214	-0.5480114
10:00	0.84647366	0.84354118	0.84060262	0.83765795	0.83470716	0.83175018	0.78330745	-0.5463669	-0.5478445	-0.5483588
11:00	0.84631322	0.84321962	0.84011925	0.83701208	0.8338981	0.83077727	0.81821774	-0.5390173	-0.5481002	-0.5486333
12:00	0.84617884	0.84295025	0.83971429	0.83647092	0.83322012	0.82996187	0.82074088	-0.5353279	-0.548311	-0.5488594
13:00	0.84605655	0.84270511	0.83934572	0.83597835	0.83260298	0.82921956	0.81215205	-0.544432	-0.5485004	-0.5490622
14:00	0.84593392	0.84245925	0.83897604	0.83548425	0.83198385	0.82847478	0.75758208	-0.5479836	-0.548688	-0.5492628
15:00	0.84580295	0.84219667	0.83858118	0.83495645	0.83132245	0.82767407	0.2246516	-0.548273	-0.5488857	-0.549474
16:00	0.84566314	0.84191632	0.83815957	0.83439283	0.83061608	0.82609523	-0.5279669	-0.5484661	-0.5490939	-0.5496962
17:00	0.84552439	0.84163806	0.83774104	0.83383329	0.82991476	0.80709808	-0.5478075	-0.5486551	-0.5492976	-0.5499133
18:00	0.84541	0.84140865	0.83739596	0.83337189	0.82933639	0.70589698	-0.5481236	-0.5488091	-0.5494635	-0.5500899
19:00	0.84535968	0.84130772	0.83724415	0.83316889	0.82908191	0.61184284	-0.5481862	-0.5488763	-0.5495359	-0.5501669
20:00	0.84543249	0.84145376	0.83746382	0.83346262	0.82945013	0.75894375	-0.5480895	-0.5487789	-0.549431	-0.5500554
21:00	0.84570977	0.84200983	0.8383002	0.83458083	0.8308517	0.8267807	-0.5054708	-0.548402	-0.5490248	-0.5496224
22:00	0.84629803	0.84318918	0.84007349	0.83695093	0.83382149	0.83068514	0.82741383	-0.0723533	-0.5481228	-0.548659
23:00	0.80352708	0.80113259	0.79874113	0.79635267	0.79396722	0.79158478	0.78920533	0.78682886	0.78445539	0.78202875

6. 排队等候仿真 MATLAB 程序源代码：

```
MMS_test(4,500,9,3,60);

ylim([0 63]);
```

```
function out=MMS_test(s,k,lambda,mu,T)
%多服务台
%s——服务台个数
%k——最大顾客等待数
%T——时间终止点
%mu——服务率
%事件表：
%   arrive_time——顾客到达事件
%   leave_time——顾客离开事件
%mintime——事件表中的最近事件
%current_time——当前时间
%L——队长
%tt——时间序列
%LL——队长序列
%c——顾客到达时间序列
%b——服务开始时间序列
%e——顾客离开时间序列
%a_count——到达顾客数
%b_count——服务顾客数
%e_count——损失顾客数
%初始化
arr_mean = 1/lambda;
ser_mean = 1/mu;
arrive_time=exprnd(arr_mean);
leave_time=[];
current_time=0;
L=0;
LL=[L];
tt=[current_time];
c=[];
b=[];
e=[];
a_count=0;
b_count=0;
e_count=0;
arrival = [];
left = [];

%循环
while min([arrive_time,leave_time])<T
    current_time=min([arrive_time,leave_time]);
    tt=[tt,current_time];    %记录时间序列
    if current_time==arrive_time          %顾客到达子过程
        arrive_time=arrive_time+exprnd(arr_mean);  % 刷新顾客到达事件
```

```
        arrival=[arrival,arrive_time];
        a_count=a_count+1; %累加到达顾客数
        if  L<s            %有空闲服务台
            L=L+1;        %更新队长
            b_count=b_count+1;%累加服务顾客数
            c=[c,current_time];%记录顾客到达时间序列
            b=[b,current_time];%记录服务开始时间序列
            tmp = current_time+exprnd(ser_mean);
            if tmp < arrive_time
                tmp = arrive_time;
            end
            leave_time=[leave_time,tmp];%产生新的顾客离开事件
            left = [left,tmp];
            leave_time=sort(leave_time);%离开事件表排序
        elseif L<s+k             %有空闲等待位
            L=L+1;        %更新队长
            b_count=b_count+1;%累加服务顾客数
            c=[c,current_time];%记录顾客到达时间序列
        else               %顾客损失
            e_count=e_count+1;%累加损失顾客数
        end
    else                   %顾客离开子过程
            leave_time(1)=[];%从事件表中抹去顾客离开事件
            e=[e,current_time];%记录顾客离开时间序列
            if L>s    %有顾客等待
                L=L-1;        %更新队长
                b=[b,current_time];%记录服务开始时间序列
                tmp = current_time+exprnd(ser_mean);
                leave_time=[leave_time,tmp];
                left = [left,tmp];
                leave_time=sort(leave_time);%离开事件表排序
            else    %无顾客等待
                L=L-1;        %更新队长
            end
    end
    LL=[LL,L];   %记录队长序列
end
Ws=sum(e-c(1:length(e)))/length(e);
Wq=sum(b-c(1:length(b)))/length(b);
Wb=sum(e-b(1:length(e)))/length(e);
Ls=sum(diff([tt,T]).*LL)/T;
Lq=sum(diff([tt,T]).*max(LL-s,0))/T;
fprintf('到达顾客数:%d\n',a_count)%到达顾客数
fprintf('服务顾客数:%d\n',b_count)%服务顾客数
```

```
fprintf('损失顾客数:%d\n',e_count)%损失顾客数
fprintf('平均逗留时间:%f\n',Ws)%平均逗留时间
fprintf('平均等待时间:%f\n',Wq)%平均等待时间
fprintf('平均服务时间:%f\n',Wb)%平均服务时间
fprintf('平均队长:%f\n',Ls)%平均队长
fprintf('平均等待队长:%f\n',Lq)%平均等待队长
if k~=inf
   for i=0:s+k
     p(i+1)=sum((LL==i).*diff([tt,T]))/T;%队长为 i 的概率
%       fprintf('队长为%d 的概率:%f\n',i,p(i+1));
   end
else
   for i=0:3*s
     p(i+1)=sum((LL==i).*diff([tt,T]))/T;%队长为 i 的概率
%       fprintf('队长为%d 的概率:%f\n',i,p(i+1));
   end
end
fprintf('顾客不能马上得到服务的概率:%f\n',1-sum(p(1:s)))%顾客不能马上得到服务的概率
out=[Ws,Wq,Wb,Ls,Lq,p];left = [left,tmp];
stairs(0:length(arrival)-1,arrival,"LineWidth",1.5);
hold on;
stairs(0:length(left)-1,left,'-r',"LineWidth",1.5);
legend('到达时间','离开时间');
legend('location','northwest');
xlabel('人数');
ylabel('时间（分钟）');
hold off;
grid on;
end
```