基于排队论的机场出租车选择与调度研究

摘　要

本文为提供机场出租车司机的选择决策方案和短途车优先排车方案，并在双车道上合理设计上车点，建立基于排队论的出租车选择决策模型、双车道上车点设计模型和短途车优先排车模型，运用决策函数、目标优化等方法对模型进行分析，使用 Python 对模型进行求解和验证，为出租车司机提供合理的选择决策，并为机场设计上车点和解决短途车的排队问题提供有效方案。

对于问题一：为研究影响机场出租车司机选择决策的因素，需分析影响司机收益的变量。影响收益的变量分为收入和成本，本题难点在于时间成本的计算，涉及出租车排队等待的时间 t_1。因此本文建立**基于排队论的出租车选择决策模型**，以采取等待策略的司机从到达机场至将乘客送达目的地为一个时间周期，在一个时间周期内比较等待策略和空载返回策略的收益大小，为出租车司机制定合理的选择策略。本文对收益分析，建立两种策略的收益函数分别为：$y_1=c_2-c_{oi\,l}=c_2-0.4785(l_1+l_2)$和 $y_2=at_1-0.4795l_2$，进而提出的选择决策方法为：当 $y_1> y_2$，出租车司机选择等待拉客的策略；当 $y_1< y_2$，出租车司机选择返回市区拉客策略。

对于问题二：本文选择北京首都国际机场和北京市的相关数据为研究对象，从 PlaneFlightTracker 等网站收集相关数据，并进行清洗加工。将数据输入问题一的模型中，以蓄车池内等待车数 p 和时间 t 为决策变量，使用 Python 对决策函数进行求解。

为首都机场出租车司机提供的决策结果如图 6 所示，例如当前时间为 22:00 且蓄车池内等待车数小于 400，则选择等待策略；当前时间为 8:00 且等待车数超过 300 个，则选择空载返回策略。模型加入高斯噪声处理后，能够对选择进行合理化决策，说明该模型合理；分别将变量大小改变 5%和 10%，决策结果会灵敏地发生变化，说明选择结果对相关变量的依赖性强。

对于问题三：首先对并行车道进行分析，绘制双车道上车服务模拟图（如图 8）。则要确定上车点设计原则，只需求得上车点数，上车点位置均匀等距分布在乘客等候区。由于需要综合考虑乘客排队和出租车排队两种情况，本文**建立基于排队论的双车道上车点设计模型**，以平均队伍长度 Ls 为目标变量。当 $\min(L_sd+n)$ 时，队伍长度最短，则等待时间最短，此时 n 为最优上车点的个数。当权重 d=1 时，对函数进行遍历求解，获得最优上车点个数为 2 个，此时乘车效率最高。对模型进行灵敏性分析，结果如表 7 所示，即当权重 d 分别取 0.01, 0.1, 1, 10, 100 时，上车点个数 n 分别为 1, 1, 2, 2, 3，模型灵敏性较好。

对于问题四：设计单独的短途车道，便于短途车更快接单获取收益，如图 10 所示。因此，为确定短途车允许到达泊车区入口的时间，本文建立**基于排队论的短途车优先排车模型**，从短途车接客出发开始至接到下一单结束为一个时间周期，求解在一个时间周期内，长短途出租车获取相等收益时，短途车等待接单的时间 t_d。当短途车系数 γ=0.2 时，模型求解结果如图 12 所示。可以看到，随着时间 t 的变化，每个时间点对应一个准确的等待时间。如短途出租车在晚上 21:00 时，需要等待 20 分钟后可以由短途车道进入蓄车池，使其在下一次入口开放时优先进入泊车区接客，进而使得短途出租车的收益与长途车的收益相同。

关键词：机场出租车 选择决策 调度 排队论 单目标优化 Python

一 问题重述

1.1 背景与研究意义

随着国民经济水平的提高和国内疫情防控工作取得重要成果，机场的进出港航班量与旅客量逐步上升。如何快速安全有序地疏散旅客，不发生大面积滞留，是交通管理部门的一项重要课题[1]。在机场，除了机场巴士、轨道交通以外，乘坐出租车也是旅客抵离机场的主要交通方式。

国内多数机场都是将送客与接客通道分开的。送客到机场的出租车司机都将会面临两个选择：前往等待区排队等待接客还是空载返回市区拉客。有经验的出租车司机会根据当前时间的航班数和机场待拉客的出租车数做出判断。但实际上，还有很多影响出租车司机决策的确定和不确定因素，其关联关系各异，影响效果也不尽相同。

1.2 需要解决的问题

基于上述背景，本文将建立数学模型，解决以下问题：

（1）综合考虑乘客数量变化规律和出租车司机收益，分析影响出租车司机选择的相关因素，建立出租车司机决策模型，并为司机制定合理的选择策略；

（2）选择国内某一城市，收集与出租车和机场相关的数据，为出租车司机制定选择方案，分析模型的合理性和因素的依赖性；

（3）制定“上车点”位置设计原则，在保证安全的前提下合理安排出租车和乘客，使得乘车效率达到最高；

（4）给予短途载客出租车一定的“接客优先权”，制定可行的优先安排方案，使得出租车达到收益相对平和。

二 问题分析

2.1 对问题一的分析

通常情况下，在特殊时间对机场出租车的需求量较大，出租车司机根据经验可能会采取等待策略，但是也可能存在较大的时间成本，比空载返回市区的收益小。

出租车司机的决策决定了最终的收益，为选择收益较高的策略，本文将建立基于排队论的出租车司机选择决策模型，以采取等待策略的司机从到达机场开始等待至将乘客送达目的地为一个时间周期，在一个时间周期内比较等待策略和空载返回策略的收益大小，为出租车司机制定合理的选择策略。

2.2 对问题二的分析

本文选择的研究对象为北京首都国际机场和北京市。首先本文将收集北京首都机场一天内航班数的变化，并根据北京市实际情况对乘客乘坐出租车的概率、航班的平均载客量、单位时间成本等变量赋值，代入问题一所得到的决策函数中进行求解，并对两种决策值进行比较，为司机提供合理的决策方案。

对于模型合理性的验证，本文将该决策方案进行稳定性分析，输入噪声后判断决策是否发生改变；对于模型依赖性的验证，将对方案进行灵敏度分析，改变变量的部分赋值，观察其是否会发生明显的变化。

2.3 对问题三的分析

为解决出租车排队载客和乘客排队乘车的情况，需要设置并行车道，优化上车点的位置。并行车道的上车点设计方案较多，如单上车点出租车排队服务、单列多上车

点服务、双列多上车点服务以及多并列点式上车点服务[2]等。

考虑乘客上车和车辆碰撞等安全因素，针对客流量大的机场和容易造成交通拥堵的大型机场，本文将不同类型的上车点服务系统进行对比分析，选择安全性高、乘车效率高且不易拥堵的服务方式；确定服务系统后，建立排队论模型，确定关于上车点个数的平均等待队长函数，求得平均等待队长的最小值对应的上车点个数即为最佳的上车点位置设计原则。

2.4 对问题四的分析

长途车接客后，从机场去往市区内的目的地送客，后会继续留在市区接客；而短途车送客后，往往会选择返回机场重新接客，否则会导致长途车与短途车的收益差过大。为了给短途出租车提供排队的“优先权”，本文将建立基于排队论的优先排队模型，以短途比例系数 γ 和蓄车池内出租车数 p 为变量，求得在相同的时间段内，长途与短途车收益相同时，短途车的排队策略。

三 基本假设

1. 假设本文收集到的全部数据科学、真实、有效。
2. 假设出租车空载返回市区与接客后空载返回市区距离和时间相同，即成本相同。
3. 假设空载返回市区时无接单。
4. 假设长途车返回市区后不再接机场接客的单。
5. 将出租车司机抽象为一个质点，且假设出租车司机匀速行驶。

四 名词解释与符号说明

符号	意义	单位
T	时间周期	秒
t_1	等待时间	秒
t_2	上车时间	秒
t_3	从机场到达市区的时间	秒
e_1	单位时间内出租车在市区的收益	元/小时
e_2	按照出租车收费标准收入	元
λ	乘客平均到达率	/
μ	服务率	/
W	滞留时间	秒
x	飞机平均载客数	人
m	平均每辆车上的载客数	人
l_1	从蓄车池到接客地的距离	公里
l_2	从机场到市区的距离	公里

c_{time}	时间成本	元
c_{oil}	油费成本	元/公里
a	单位时间内的时间成本	元/小时
y	收益	元
L_s	蓄车池内的平均等待队长	米
p	蓄车池内的出租车数量	辆
n	上车点的数量	个
γ	短途车系数	/
s_1	长途车收益	元
s_2	短途车收益	元
$T^{'}$	长短途车时间周期	秒
$e_2^{'}$	短途车送达乘客收益	元

五 模型的建立和求解

5.1 基于排队论的出租车选择决策模型

5.1.1 选择策略的分析

出租车司机的两种选择分别为等待拉客和空载返回。为寻找较好的选择方案，首先需要对两种策略进行分析，具体过程分别是：

等待策略：司机到达机场后，去往蓄车池按照“先来后到”的原则排队，等待拉客；若可以接客，则到达接客地，将顾客送往目的地，完成任务。

空载返回策略：司机到达机场后，返回市区，在市区内进行接客。

为使得过程展示更加直观，本文用时间轴对上述策略进行描述，如图 1 所示：

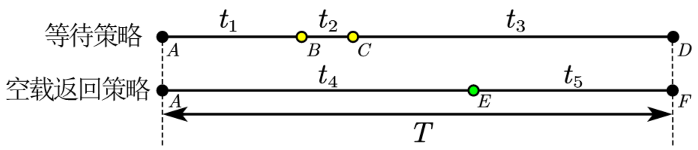


图 1 单个周期两种策略的时间轴

图 1 中符号表示含义如下：

表 1 图 1 中各符号表示及其含义

符号表示	符号含义	符号表示	符号含义
A	机场	t_1	司机在机场蓄车池的等待时间
B	蓄车池	t_2	上车时间
C	接客地	t_3	将乘客从机场送达目的地的时间
D	乘客目的地	t_4	司机从机场返回市区的时间
E	市区	t_5	司机在市区的接送单时间
F	采取空载返回策略的出租车在单个周期结束后的位置点	T	时间周期

由表 1 和图 1 可知：时间周期的划分标准为：等待策略的司机从开始等待直至将乘客送达目的地的时间长度，而最终的策略为在单个时间周期下，比较两种策略的收益大小。

需要注意的是，EF 段为司机在市区的接送单情况。在 t_5 时间段中，出租车可能会未接单，也可能会接受到多个单。

5.1.2 决策函数模型的变量分析

首先，对影响出租车司机收益的变量进行分析。司机能够分析的变量为航班数和蓄车池内出租车的数量信息。具体的变量分析如图 2 所示：

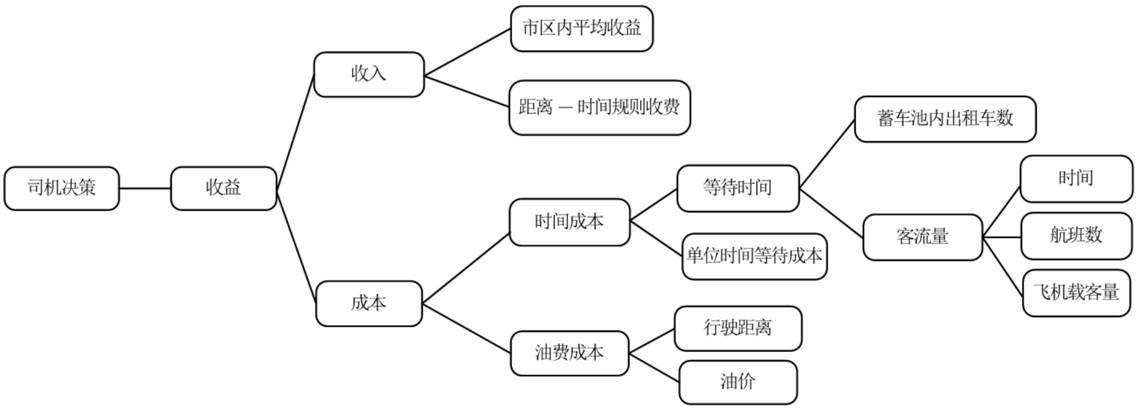


图 2 影响选择决策的变量分析

图 2 中的部分变量分析如下：

收入 e：本文将收入成本分为两类：市区内单位时间收益 e_1 和按照距离-时间规则收费收入 e_2。

对于市区内收益 e_1 而言，由于在（T-t_3）时间段内出租车司机接单数和每单行驶的距离是随机的，因此无法直接按照出租车收费标准和标准油价进行准确计算，求得其准确的收入和成本。因此，本文将统计市区内单位时间出租车的平均收益作为市区内的收益。实际上，受早晚高峰的影响，不同时间段的平均收益有所差异，e_1 的表达为分段函数，具体分段将在问题二中输入实际数据进行解决。

对于从机场到市区的路段而言，不会有新的订单产生，且路段距离确定，因此可以按照出租车收费标准进行准确计算。本文以北京市 2019 年出租车收费标准[3]为例：

白天（早 5:00--晚 22:59）起步价 10 元（三公里以内），超出（含）三公里至十五公里以内的公里数每公里按 2 元计费。超出（含）十五公里以外的公里数（每公里加收 50%空驶费）按 3 元计费；

夜间（晚 23:00-早 4:59）起步价 11 元（三公里以内）其它计费方式同上，但是每公里另加收 20%的夜间费用（不含起步价 11 元）。

设当前时间为 t，从机场到市区的距离为 l_2，则从机场到市区的收入 e_2：

当 $5 \le t < 23$ 时：

$$e_2 = \begin{cases} 10 & 0 \le l_2 < 3 \\ 10 + 2(l_2 - 3) & 3 \le l_2 < 15 \\ 34 + 3(l_2 - 15) & l_2 \ge 15 \end{cases}$$

当 $23 \le t \le 24 \,\&\, 0 \le t < 5$ 时：

$$e_2 = \begin{cases} 11 & 0 \le l_2 < 3 \\ 11 + 2.4(l_2 - 3) & 3 \le l_2 < 15 \\ 39.8 + 3.6(l_2 - 15) & l_2 \ge 15 \end{cases}$$

等待时间 t_1：由图 2 所示，等待时间受蓄车池内出租车数量和客流量的影响。而在蓄车池的等待过程可以近似看做是单队列的排队问题。因此，对于等待时间的求解，本文将建立 M/M/1 的排队论模型[4]。

设平均到达率为 λ，服务率为 μ，平均每名乘客在出租车接客点的平均滞留时间 W_q，通过 p 辆出租车所需的时间为 t_1，即第（n+1）辆出租车所需要等待的时间， 滞留时间 W 为参数（μ-λ）的负指数分布，则平均滞留时间 W_s：

$$W_s = E(W) = \frac{1}{\mu - \lambda}$$

出租车的等待时间可以等效为上该车的乘客的等待时间。假设某出租车前方有 p 辆车，平均每辆车上有 m 个乘客，则其等待时间 t_1 为：

$$t_1 = W_q * m * p = \frac{mp}{\mu - \lambda}$$

一段时间内，对乘客的到达率 λ 设对应法则为 c，即 $\lambda = c(t)$，其中 $c(t)$ 包含飞机载客量、航班数、时间等客观影响因素。设每日不同时间的航班数为 $f(t)$，每个航班上的乘客数 x，乘客乘坐出租车离港的概率 $h(t)$，则：

$$c(t) = f(t) * h(t) * x$$

根据常识，乘客到达率与飞机载客量成正相关；在旅游旺季和特殊时间段时，乘客选择出租车出行的倾向性越高，顾客到达率越高；极端天气影响下，航班数减少，则会影响顾客到达率。将 λ 用 $c(t)$ 表示，则：

$$t_1 = \frac{mp}{\mu - c(t)}$$

时间成本 c_{time}：出租车在单位等待时间内损失的收入。对于等待策略而言，等待时间 t_1 中可能会导致出租车损失一定的收入，其值可以通过出租车在市区内正常接客的单位时间的收入计算。设单位时间内的等待成本为 a，公式表示如下：

$$c_{time} = at_1$$

分析可知，方案一种的时间成本与方案二中的返回市区的收益相同，即：

$$e_1(t_5) = c_{time}$$

油费成本 c_{oil}：按照北京市 92 号汽油最新价格 6.85 元/升[5]计算，设每辆出租车的百里油耗为 7 升，则计算获得单位公里的油费成本为 0.4795 元，即：

$$c_{oil} = 0.4795 * l$$

从机场到市区的距离 l_2：对于市区范围的定义，本文以某城市的市中心为圆心，建立正态分布的辐射图，在该范围的坐标点即为市区。

5.1.3 决策函数模型的建立

基于以上分析和说明，分别构建收益函数 y_1 和 y_2：等待策略收益函数 y_1:

$$y_1 = e_2 - c_{oil} = e_2 - 0.4795 * (l_1 + l_2)$$

空载返回市区策略收益函数 y_2:

$$y_2 = e_1(t_5) - c'_{oil} = \frac{amp}{\mu - c(t)} - 0.4795 * l_2$$

若 $y_1 > y_2$，则出租车司机选择继续等待拉客的策略获取的收益较高；若 $y_1 < y_2$，则出租

车司机选择返回市区拉客策略获取的收益较高。

5.2 排队论模型的仿真

5.2.1 数据收集与清洗

为使得模型的验证具有代表性，本文选择国内北京首都国际机场的 2021 年 8 月 28 日的航班数据，该数据来源于 PlaneFlightTracker 网站[6]，数据真实有效。将收集到的数据进行清洗，删除延误等未抵达机场的航班信息，仅统计成功抵达首都机场的航班数，即函数 $f(t)$的图像：

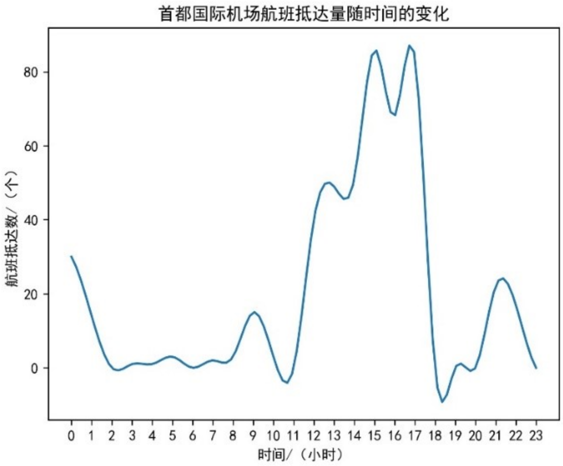


图 3 2021 年 8 月 28 日成功抵达首都国际机场的航班数变化

由图 3 可以看到，当日 13 时-17 时抵达机场的航班数较大，预计对出租车的需求量增加。

根据北京市地铁运营时间规范，晚上 23:00 至次日早 5:00，地铁处于停运状态，因而此时乘客选择出租车离航的概率会有所不同。本文对乘客选择乘坐出租车的概率 $h(t)$ 的假设为：

$$h(t)=\begin{cases}0.1 & 5\le t<23\\ 0.9 & 23\le t<24\&0\le t<5\end{cases}$$

根据经验分析，设民航平均有效载客量 x=80。由公式$c(t)=f(t)*h(t)*x$可以计算得到首都机场乘客对出租车的需求量变化，如下图所示：

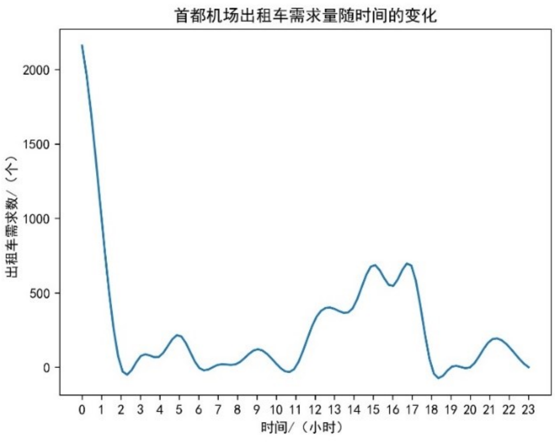


图 4 首都机场对出租车需求量的变化

根据北京首都国际机场和北京市区的分布及出租车的运营情况分析，假设平均每辆出租车载客量 m=2，出租车服务率 μ=0.1，即预估平均上车时间为 10s，从蓄车池到接客点的 l_1=10km，从首都机场到达北京市区的 l_2=50km，北京市区出租车司机平均收益为 50 元/小时，即 $\frac{1}{72}$ 元/秒。

5.2.2 模型求解

5.2.2.1 时间对决策的影响

实际上，出租车前方的车辆数，即蓄车池内出租车的数量 p 是根据具体情况而确定的。本文为对模型进行仿真和求解，首先设当前某出租车司机前方的车辆数 p=100，则将上述变量代入问题一中的决策函数分别进行求解，获取如下结果：

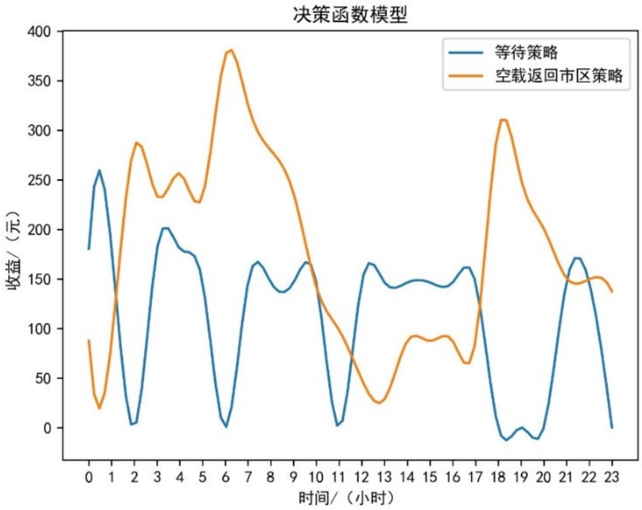


图 5 决策函数随时间的变化

表 2 选择决策结果

策略类别	选择的具体时间
等待策略	0,1,10,12,13,14,15,16,17
空载返回策略	2,3,4,5,6,7,8,9,11,18,19,20,21,22,23

结合图表分析，为出租车司机提供的决策如下：

当时间段为 0-1 点，10 点，12-17 点时，$y_1 > y_2$，此时出租车司机选择继续等待拉客的策略获取的收益较高；

当时间为 2-9 点，11 点，18-23 点时，$y_1 < y_2$，出租车司机选择返回市区拉客策略获取的收益较高。

5.2.2.2 蓄车池内出租车数量和时间双变量对决策的影响

出租车前方的车辆数，即蓄车池内出租车的数量 p 是不断变化的，会对出租车司机的决策产生影响。当 y_1=y_2 时，以时间 t 和蓄车池内出租车数量 p 为双变量，曲线为决策的边界，作图如下：

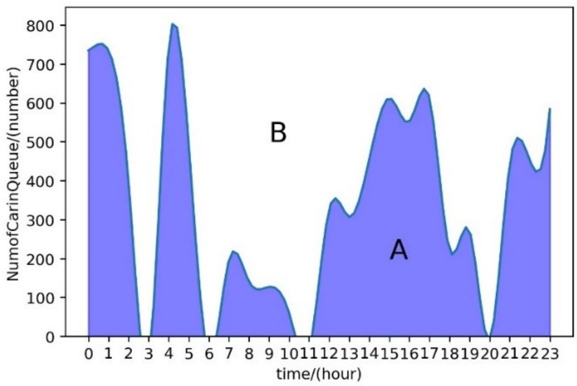


图 6 双变量对选择的决策

分析可知，深色的部分为等待决策时情况，浅色的部分为空闲返回决策的情况。因此，综合考虑时间和蓄车池内出租车数量两个因素，可以为司机提供选择的决策。

5.2.3 模型验证

5.2.3.1 模型合理性验证

为检验模型的合理性，判断该模型是否能在多个场合进行应用，则需要进行稳定性分析，即给不确定因素——航班数加入高斯噪声处理，获取结果如下：

表 3 高斯处理后的选择决策

σ	策略类别	选择的具体时间
0	策略一	0,1,10,12,13,14,15,16,17
	策略二	2,3,4,5,6,7,8,9,11,18,19,20,21,22,23
1	策略一	0,1,10,12,13,14,15,16,17
	策略二	2,3,4,5,6,7,8,9,11,18,19,20,21,22,23
2	策略一	0,1,2,3,4, 12,13,14,15,16,17,21,22,23
	策略二	5,6,7,8,9, 10,11,18,19,20
3	策略一	0,1,12,13,14,15,16,17,21,22,23
	策略二	2,3,4,5,6,7,8,9,10,11,18,19,20
4	策略一	0,1,4,5,12,13,14,15,16,17.21,22,23
	策略二	2,3,6,7,8,9,10,11,18,19,20

可以看到，随时航班数的变化，模型可以寻找到合理的选择决策方案，为司机提供有力的选择参考，因而模型合理。

5.2.3.2 相关因素依赖性分析——灵敏度分析

（1）飞机平均载客量 x。改变决策函数中的参数 x，分别加减 5%、10%的变量值大小，观察模型的结论是否发生变化。设等待策略为策略一，空载返回策略为策略二，决策结果如下：

表 4 平均载客量对决策结果的影响

变化幅度	策略类别	选择的具体时间
-10%	策略一	0,1,10,12,13,14,15,16,17,21,22
	策略二	2,3,4,5,6,7,8,9,11,18,19,20,23
-5%	策略一	0,1,10,12,13,14,15,16,17,21,22
	策略二	2,3,4,5,6,7,8,9,11,18,19,20,23
0%	策略一	0,1,10,12,13,14,15,16,17

	策略二	2,3,4,5,6,7,8,9,11,18,19,20,21,22,23
5%	策略一	0,1,12,13,14,15,16,17
	策略二	2,3,4,5,6,7,8,9,10,11,18,19,20,21,22,23
10%	策略一	0,1,12,13,14,15,16,17
	策略二	2,3,4,5,6,7,8,9,10,11,18,19,20,21,22,23

当 x 增加时，选择乘坐出租车去市区或周边的乘客数量增加，对出租车的需求量增加，出租车司机选择继续等待的收益更大，因而等待策略的时间增加了 21-22 点的时间段，可以灵敏地反映变量 x 对决策结果的影响，灵敏度较高，决策对飞机平均载客量的依赖性较大。

（2）市区内订单的平均收益 a。改变决策函数中的参数 x，分别加减 5%、10%的变量值大小，观察模型的结论是否发生变化。决策结果如下：

表 5 市区内平均收益对决策结果的影响

变化幅度	策略类别	选择的具体时间
-10%	策略一	0,1,10,12,13,14,15,16,17,21,22
	策略二	2,3,4,5,6,7,8,9,11,18,19,20,23
-5%	策略一	0,1,10,12,13,14,15,16,17,21,22
	策略二	2,3,4,5,6,7,8,9,11,18,19,20,23
0%	策略一	0,1,10,12,13,14,15,16,17
	策略二	2,3,4,5,6,7,8,9,11,18,19,20,21,22,23
5%	策略一	0,1,12,13,14,15,16,17
	策略二	2,3,4,5,6,7,8,9,10,11,18,19,20,21,22,23
10%	策略一	0,1,12,13,14,15,16,17
	策略二	2,3,4,5,6,7,8,9,10,11,18,19,20,21,22,23

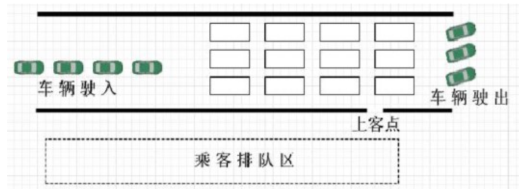
当 a 增加时，当订单平均价格增加时，选择回市区的收益更大，出租车司机选择直接空载返回市区拉客，因而等待策略的时间增加了 21-22 点的时间段，可以灵敏地反映变量 a 对决策结果的影响，灵敏度较高，决策对市区平均收益的依赖性较大。

5.3 基于排队论的双车道上车点位置设计模型

5.3.1 并行车道分析

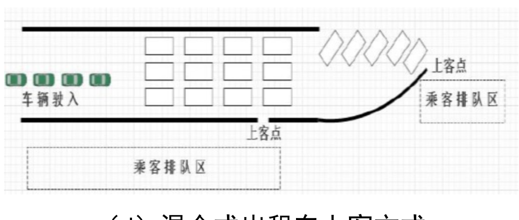
目前国内机场出租车上客区的主要布局形式分为：单车道出租车上客方式、多车道出租车上客方式、斜列式出租车上客方式和混合式出租车上客方式[7]，如图 7 所示：

（a）单车道出租车上客方式



（b）多车道出租车上客方式

（c）斜列式出租车上客方式



（d）混合式出租车上客方式

图 7 国内机场出租车上车点主要上客方式

可以看到，由于车道的限制，单车道出租车上客方式大大限制了出租车的数量，乘车效率较低；而多车道上客方式便于停靠出租车，且在指定区域内上车提高了安全系数，方便搬运与上车；斜列式上客方式的空间利用率高，且可以有效降低人车混行带来的危险；混合式上客结合了多车道和斜列式的优缺点，效率高，且较为安全。由题目可知，本文采用双并行车道，则采用双车道出租车上客方式。参考图 7 的（b），根据题意绘制双车道上车服务模拟图，如图 8 所示。

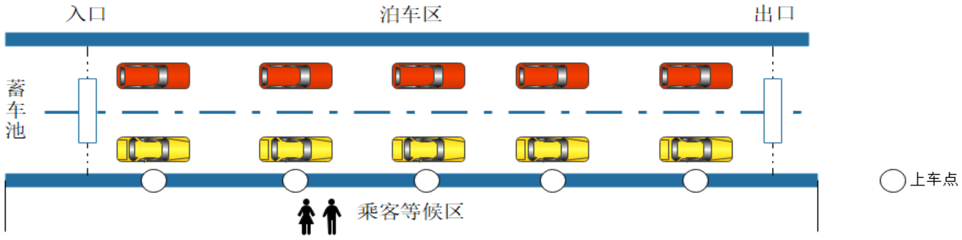


图 8 双车道上车服务

由图 8 可知，双车道上车服务过程为：蓄车池中的出租车按照“先来后到”的顺序依次进入泊车区，到达指定泊位。泊位采用 n 泊位*2 列方式设置。出租车在停稳后，乘客依次从上车点进入泊车区；乘客全部上车后，出口打开，各车道的车辆依次驶出；当每个车道中最后一辆车驶离车位时，打开入口，后续出租车驶入该列，在该列的各个泊位依次停下，等待乘客上车。则为了确定上车点的设计原则，只需要计算上车点的数量即可，确定数量后，上车点均匀分布在乘客等候区。

5.3.2 模型的建立

排队服务系统(即是出租车上车点) 是用来描述排队情况的系统，此系统中将有限的服务台(出租车上车点) 用来服务乘客，乘客来到上车点完成上车动作后离开服务区。当服务台都处于忙碌状态时，乘客需要加入到系统中的排队等待中，直到乘客接受完成服务后离开此服务区；当乘客较少，出租车则加入排队的队伍中等待，直至有乘客接受服务。

假设有 n 个出租车上车点，r 位正在等待的乘客，F 表示出租车的状态，下图为系统状态转移速度图：

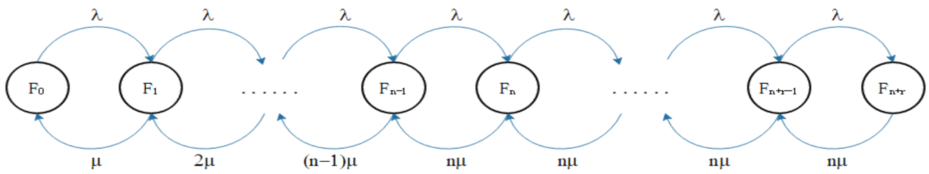


图 9 系统状态转移速度图

根据图 9，建立出租车状态模型，并求得状态方程：

表 6 出租车状态表示

状态表示	状态含义	状态方程
F_0	所有上车点无乘客上车	$K_1=\xi K_0$
F_{n-1}	(n-1)个上车点出租车在服务 1 个上车点空闲	$K_n=\frac{\xi^n}{n!}K_0$
F_n	n 个上车点出租车均在服务 没有乘客等待	$K_{n+1}=\frac{\xi^n}{n\times n!}K_0$

F_{n+r-1}	n个上车点出租车在服务 (r-1)名乘客排队	$K_{n+r}=\dfrac{\xi^n}{n^r\times n!}K_0$

由以上状态方程和 $K_1+K_2+\cdots+K_n+\cdots+K_{n+r}+\cdots=1$ 可得：

$$K_0=\left[\sum_{i=0}^{n}\frac{\xi^i}{i!}+\frac{\xi^{i+1}}{n!(n-\xi)}\right]^{-1}$$

设蓄车池内的平均等待队长为 L_s：

$$L_s=\frac{\xi^{n+1}}{(n-1)!\times(n-\xi)^2}K_0+\xi$$

其中，ξ表示出租车服务系统的负荷水平，且 $\xi=\dfrac{\lambda}{\mu}$。当 ξ=1 时，系统达到平衡状态。

根据问题分析可知，当 min(L_sd+n)时，求得的 n 为最优的上车点的个数， d 为权重，即表示排队长度对模型的影响程度。

5.3.3 模型的求解与验证

设权重 d=1，根据 min(L_sd+n)，依据排队论原理遍历求解。遍历结果为：当 n=2 时，即设置两个上车点效果最佳，此时乘车效率较高。

为验证该模型的灵敏度，本文选择改变权重 d 的大小，分别取 d=0.01, 0.1, 1, 10, 100 五个取值，代入模型求解，获得的上车点数量结果如下：

表 7 不同权重下的上车点个数

权重 d	上车点个数 n
0.01	1
0.1	1
1	2
10	2
100	3

由表 7 可知，改变权重时会导致上车点的数量发生变化。当权重越小，即队列长度对上车点位置设计影响越小时，一个上车点就足以为乘客提供服务；而当权重越大，队列长度对上车点位置设计影响越大时，就需要设计更多的上车点来缓解排队的压力，缩短排队长度，则此时设计的上车点个数为 2 或 3，满足实际情况。由此分析，模型的灵敏性较好，可以随着变量的变化提供不同的上车点设计结果。

5.4 基于排队论的短途车优先排车模型

5.4.1 模型的建立

为使得短途车能够在排队时获取优先权，则需要在蓄车池修建一个单独的短途车道，该道路便于短途车能够在优先级高的情况下尽快接客，具体示意图如图 10 所示：

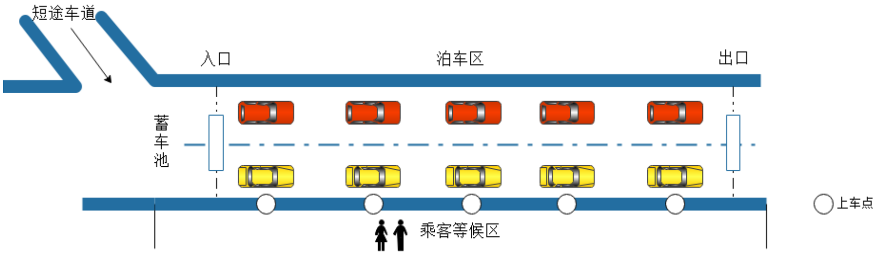


图 10 附加短途车道的双车道上车服务

根据问题四的分析可知，短途车送达乘客后返回机场接单，从短途车出发开始计时，至短途车接到下一单结束为一个时间周期 T'，设送达乘客的时间为 t_c，在机场等待下一单的时间长度为 t_d，则对于短途车而言：

$$T_2' = 2t_c + t_d$$

长途车送达乘客后将在市区继续接客，设长途车将乘客送达乘客的时间 t_a，在市区内继续接单的时间为 t_b，则对于长途车而言：

$$T_1' = t_a + t_b$$

具体时间轴如图 11 所示：

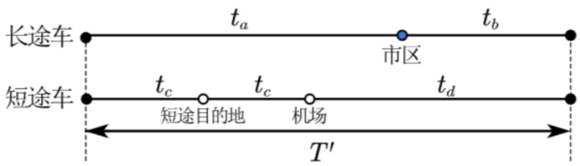


图 11 单个周期长短途车的时间轴

为确定短途车的定义，设短途车系数 γ。γ 越小，短途车送达乘客距离机场的距离越短，优先级可能会越高。则根据定义可知，$t_c = \gamma t_a$，则：

$$T_2' = 2\gamma t_a + t_d$$

为使得长短途车的收益相同，则需分别计算长短途车的收益。设长途车收益为 s_1，短途车收益为 s_2，收益的计算按照问题一的计算规则，则：

$$s_1 = e_1 t_b + e_2 - 0.4795 * l_2$$

$$s_2 = e_2' - 0.959\gamma * l_2 - c_{time} = e_2' - 0.959\gamma * l_2 - at_d$$

其中 e_2'表示短途车行驶 $\gamma * l_2$ 距离（送达乘客目的地）时的收益。

因此，当满足 $\begin{cases} s_1 = s_2 \\ T_1' = T_2' \end{cases}$ 时，求得不同 γ 对应的等待时间 t_d，进而可以确定短途出租车的优先排队原则。

5.4.2 模型的求解

假设短途车系数 γ=0.2，l_2 =50km，蓄车池内的出租车数 p=100，出租车车速 v=60km/h，则从机场到市区的时间 t_a= l_2/v*3600。将上述变量代入问题四建立的函数中，

获取短途车的等待时间如图 12 所示：

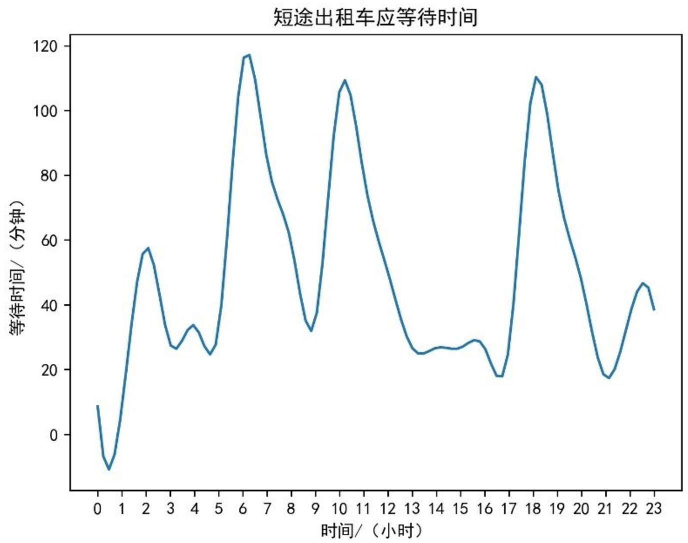


图 12 短途车等待时间 td

可以看到，随着时间 t 的变化，短途车的等待时间会发生相应的变化，且一个时间点对应一个准确的等待时间。如短途出租车在晚上 21:00 时，需要等待 20 分钟后可以由短途车道进入蓄车池，使其在下一次入口开放时优先进入泊车区接客，进而使得短途出租车的收益与长途车的收益相同。

5.4.3 模型的检验

为研究短途车优先排队模型的灵敏性，改变蓄车池内的出租车数 p，分别取 p=100, 150, 200, 250, 300，代入模型进行检验，绘制不同出租车数时等待时间 t_d 随时间的变化曲线：

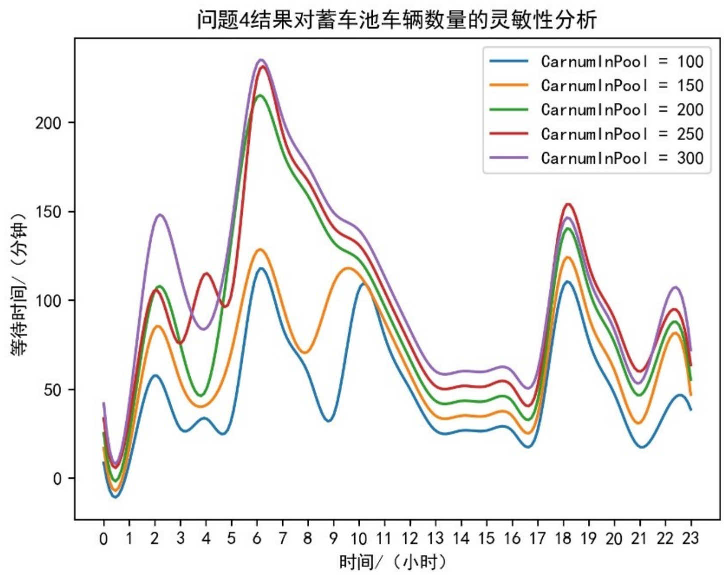


图 13 问题四模型对蓄车池内车辆数量的灵敏性分析

结合图 13 和图 4 分析，可以获得如下结论：

（1）当 p 发生改变时，变量 t_d 会随其发生变化，模型灵敏性较好。

（2）结合图 4，当机场对出租车需求量较大时，即当处于 0-2 点和 15-17 点的时间段时，曲线变化较小，说明此时改变 p 对排队时间较小。这是因为当乘客对出租车的

需求量大时，出租车等待时间极小甚至无需等待，即可接客，因此曲线变化不大。

（3）当机场对出租车需求量较小时，即当处于 2-11 点和 19-23 点的时间段时，曲线变化较大，说明此时改变 p 对排队时间较大。这是因为此时对出租车的需求小，出租车司机需要等待较长的时间才能接客；而 p 越大，则蓄车池内车越多，则需要等待的时间就越久，因而图中曲线从上到下依次为 p=300, 250, 200, 150, 100，与实际情况相符合，模型的合理性好。

六 模型的评价

6.1 模型的优点

（1）结合时间和蓄车池内出租车数两个主要变量对出租车的选择决策进行分析，且具体分析影响时间变量的因素，考虑变量较多，且仿真效果好，决策结果对相关变量的依赖性强，可以为出租车司机提供合理的选择决策方案。

（2）本文建立的模型与实际紧密联系，运用排队论解决问题，便于从图中直观地看到出租车的上车点设置和短途车的调度，可视化效果较好。

（3）使用时间轴，对不同选择策略的出租车、长短途出租车在周期内的状态进行直观描述，在相同时间段内对出租车的收益进行求解，结果具有很强的参考价值。

6.2 模型的缺点

（1）问题一中的决策模型的因素缺少对于天气情况等客观因素的考虑，会影响到决策模型的精度和迁移性。

（2）问题二中对于某些数据的预测具有主观性，没有官方的数据参考，可能会对最后做出的决策产生影响。

6.3 模型的推广

由问题二可知，本文建立的模型合理性较好，且决策结果对相关变量的依赖性强。因此，除北京首都国际机场外，可以在其他机场投入使用，根据当地机场规模和政策制定合理的调度方案，为未来解决机场出租车调度问题提出了解决方案。

七 参考文献

[1] 大兴机场出租车调度站, http://www.brta.com.cn/newsinfo/446690.html?templateId=1133604, 2020

[2]杨晶. 基于排队论的机场乘车区"上车点"的优化设计[J]. 黑龙江科学,2021,12(6):34-35. DOI:10.3969/j.issn.1674-8646.2021.06.011.

[3] 2019 北京出租车收费, http://www.lyktxs.com/news/4698.html, 2019

[4] 北京 92 号汽油最新价格, https://www.icauto.com.cn/oil/price_110000_2.html, 2021

[5] 姚入榕,赵德昌. 基于排队论的机场出租车最优决策模型[J]. 现代商贸工业, 2020, 41(33):29-32. DOI:10.19311/j.cnki.1672-3198.2020.33.013.

[6] 北京首都国际机场, http://www.planeflighttracker.com/2014/02/beijing-capital-airport-flightradar24.html, 2021

[7] 孙健. 基于排队论的航空枢纽陆侧旅客服务资源建模与仿真[D].中国矿业大学(北京),2017.

附录

附录清单：

附录一 问题一的变量计算与统计

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from scipy.interpolate import make_interp_spline
from scipy.optimize import fsolve

#方法一：默认读取第一个表单
df=pd.read_excel('data.xlsx')#这个会直接默认读取到这个 Excel 的第一个表单
df = df.dropna(axis = 0)
df=df[df['Status'] == 'Landed']
data=df.head()#默认读取前 5 行的数据
print("获取到所有的值:\n{0}".format(data))#格式化输出

df['Status'].value_counts()

df.info()
df

dates = pd.to_datetime(df['Arrival'], format = '%H:%M:%S')
dates

# 航班数量在一天内按时间段的分布 f(t) t = 0,1,...,23
# dates.groupby(dates.dt.hour).count().plot(kind="bar", xlabel = 'hour', ylabel = 'flight_count');
def f(t):
    if t in dates.dt.hour.values:
        return dates.groupby(dates.dt.hour).count()[t]
    else:
        return 0
f_cnt = [ f(i) for i in range(24)]
# plt.bar(range(len(f_cnt)), f_cnt)
x=np.array(range(len(f_cnt)))
y=np.array(f_cnt)
model=make_interp_spline(x, y)
xs=np.linspace(0,23,100)
ys=model(xs)
plt.plot(xs, ys)
plt.title("Curve")
```

```
plt.xlabel("X")
plt.ylabel("Y")

# 航班载客量 x 为飞机机型
def g(x=0):
    pass_mean = 80
    return pass_mean
# 乘客乘坐出租车概率 t = 0,1,...,23
def h(t):
    # 假设由于晚间乘车方式减少，乘车概率上升
    # t1：北京地铁(首都机场线)开始运行时间 t2：结束运行时间
    t1, t2 = 6, 23
    res = 0
    k1 = 0.1
    k2 = 0.9
    if t >= t1 and t < t2:
        res = k1
    else:
        res = k2
    return res

# 乘客变化规律
def C(t):
    return f_cnt[t] * g() * h(t)
    # return f(t) * g() * h(t)

c_cnt = [C(i) for i in range(24)]
# plt.bar(range(len(c_cnt)), c_cnt)
x=np.array(range(len(c_cnt)))
y=np.array(c_cnt)
model=make_interp_spline(x, y)
xs=np.linspace(0,23,100)
ys=model(xs)
plt.plot(xs, ys)
plt.title("Curve")
plt.xlabel("X")
plt.ylabel("Y")

# 计算等待时间 t1
def T1(t, n=0):
    # n: 蓄车池中车前车辆数
    m = 2 #平均每辆车上有 m 个乘客
    c = C(t) # 时间段 t 内乘客数可用来代替 lambda
    lam = c/3600
    t_mu = 10 #平均服务间隔 10s
    mu = 1/t_mu # 平均服务率
    ws = 1/(mu-lam) # 平均滞留时间
    # print("---------t1------------")
    # print("n: ", n)
```

```
    # print("乘客数 c：", c)
    # print("lambda: ", lam)
    # print("mu: ", mu)
    if( n > c ):
        # print("1")
        return 3600 + T1((t+1)%24, n-c)
    elif(lam < mu and 1/abs(lam - mu)*m*n > 3600):
        # print("2")
        # 修正时间错误计算
        # print("-----修正时间错误计算------")
        return 3600 + T1((t+1)%24, n-3600*abs(lam - mu))
    elif(lam < mu):
        # print("3")
        return m * n /(mu - lam)
    else:
        # print("4")
        return m * n * t_mu

# 出租车收费标准
def E2(l, t):
    if t >=5 and t < 23:
        qibu, q1, q2, q3 = 10, 2, 34, 3
    else:
        qibu, q1, q2, q3 = 11, 2.4, 39.8, 3.6
    if l < 3:
        e2 = qibu
    elif l < 15:
        e2 = qibu + q1 * (l - 3)
    else:
        e2 = q2 + q3 * (l - 3)
    return e2
```

附录二 选择决策函数及决策结果

```
# 收益函数 y1
def y1(t, n=100):
    l1  = 50 # 机场到市区的距离
    l2  = 10 # 停车场到机场的距离
    oil = 0.4795 # 单位公里的油费成本
    a = 1/72 # 时间成本: 单位等待时间内损失的收入(市区内)
    if(C(t) > 0):
        e2 = E2(l1, t) # 收入
        t1 = T1(t, n)
        c_time = a * t1
        c_oil = oil*(l1+l2)
        # print("-----------y1-------------")
        # print("e2: ", e2)
        # print("t1(min): ", t1/60)
        # print("c_time: ", c_time)
        # print("c_oil: ", c_oil)
```

```
        # return e2 - c_time - c_oil
        return e2 - c_oil
    else:
        return 0

# 收益函数 y2
def y2(t, n=100):
    t1 = T1(t, n)
    t2 = 10
    a = 1/72
    oil = 0.4795 # 单位公里的油费成本
    l1  = 50 # 机场到市区的距离
    e1 = a*(t1+t2)
    return e1 - oil*l1

# 画图
y1_list = [y1(i,400) for i in range(24)]

x=np.array(range(len(y1_list)))
y=np.array(y1_list)
model=make_interp_spline(x, y)
xs=np.linspace(0,23,100)
ys=model(xs)
plt.plot(xs, ys)
plt.title("Curve")


y2_list = [y2(i,400) for i in range(24)]

x=np.array(range(len(y2_list)))
y=np.array(y2_list)
model=make_interp_spline(x, y)
xs=np.linspace(0,23,100)
ys=model(xs)
plt.plot(xs, ys)
plt.legend()

choose_1 = [y1_list[i] >= y2_list[i] for i in range(24)]

print("choose 1: ",[i for i in range(24) if y1_list[i] >= y2_list[i]])
print("choose 2: ",[i for i in range(24) if y1_list[i] < y2_list[i]])
```

附录三 对基于排队论的选择决策模型合理性和灵敏性分析

```
## 出租车选择决策模型对相关因素的依赖性
分别改变了每架航班的平均载客量、市区内收益，观察选择的改变
"""
# 改变平均载客量
# 变化幅度：-10%, -5%, 0 , 5%, 10%
# pass_mean = 72, 76, 84, 88
```

```
# def g(x=0):
#     pass_mean = 88
#     return pass_mean
"""变化幅度：-10%, -5%, 0 , 5%, 10%
- -10%:
   - choose 1:  [0, 1, 5, 12, 13, 14, 15, 16, 17]
   - choose 2:  [2, 3, 4, 6, 7, 8, 9, 10, 11, 18, 19, 20, 21, 22, 23]
- -5%:
   - choose 1:  [0, 1, 12, 13, 14, 15, 16, 17]
   - choose 2:  [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 18, 19, 20, 21, 22, 23]
- 0%:
   - choose 1:  [0, 1, 10, 12, 13, 14, 15, 16, 17]
   - choose 2:  [2, 3, 4, 5, 6, 7, 8, 9, 11, 18, 19, 20, 21, 22, 23]
- 5%:
   - choose 1:  [0, 1, 10, 12, 13, 14, 15, 16, 17]
   - choose 2:  [2, 3, 4, 5, 6, 7, 8, 9, 11, 18, 19, 20, 21, 22, 23]
- 10%:
   - choose 1:  [0, 1, 3, 10, 12, 13, 14, 15, 16, 17, 21, 22]
   - choose 2:  [2, 4, 5, 6, 7, 8, 9, 11, 18, 19, 20, 23]
"""

# 市区内订单的平均收益的改变对决策结果的影响
# a = 50 * 0.9, 0.95, 1.05, 1.1 /h
# 收益函数 y2
def y2(t, n=100):
   t1 = T1(t, n)
   t2 = 10
   a = 1/72 * 1.1
   oil = 0.4795 # 单位公里的油费成本
   l1  = 50 # 机场到市区的距离
   e1 = a*(t1+t2)
   return e1 - oil*l1
# 画图
y1_list = [y1(i,400) for i in range(24)]
x=np.array(range(len(y1_list)))
y=np.array(y1_list)
model=make_interp_spline(x, y)
xs=np.linspace(0,23,100)
ys=model(xs)
plt.plot(xs, ys)
plt.title("Curve")
y2_list = [y2(i,400) for i in range(24)]
x=np.array(range(len(y2_list)))
y=np.array(y2_list)
model=make_interp_spline(x, y)
xs=np.linspace(0,23,100)
ys=model(xs)
plt.plot(xs, ys)
print("choose 1: ",[i for i in range(24) if y1_list[i] >= y2_list[i]])
```

```
print("choose 2: ",[i for i in range(24) if y1_list[i] < y2_list[i]])

"""变化幅度：-10%, -5%, 0 , 5%, 10%
- -10%:
   - choose 1:  [0, 1, 10, 12, 13, 14, 15, 16, 17, 21, 22]
   - choose 2:  [2, 3, 4, 5, 6, 7, 8, 9, 11, 18, 19, 20, 23]
- -5%:
   - choose 1:  [0, 1, 10, 12, 13, 14, 15, 16, 17, 21, 22]
   - choose 2:  [2, 3, 4, 5, 6, 7, 8, 9, 11, 18, 19, 20, 23]
- 0%:
   - choose 1:  [0, 1, 10, 12, 13, 14, 15, 16, 17]
   - choose 2:  [2, 3, 4, 5, 6, 7, 8, 9, 11, 18, 19, 20, 21, 22, 23]
- 5%:
   - choose 1:  [0, 1, 12, 13, 14, 15, 16, 17]
   - choose 2:  [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 18, 19, 20, 21, 22, 23]
- 10%:
   - choose 1:  [0, 1, 12, 13, 14, 15, 16, 17]
   - choose 2:  [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 18, 19, 20, 21, 22, 23]
"""

# 乘坐出租车概率对决策结果的影响
# 乘客乘坐出租车概率 t = 0,1,...,23
# def h(t):
#     # 假设由于晚间乘车方式减少，乘车概率上升
#     # t1：北京地铁(首都机场线)开始运行时间 t2：结束运行时间
#     t1, t2 = 6, 23
#     res = 0
#     k1 = 0.1
#     k2 = 0.9
#     if t >= t1 and t < t2:
#         res = k1
#     else:
#         res = k2
#     return res

# 出租车选择决策模型的合理性
# 为了衡量模型的合理性，我们对每时刻的航班到达量进行了加高斯噪声处理
import random
sigma = [0, 1, 2, 3, 4]

f_cnt = [ f(i) for i in range(24)]
f_cnt_guass = []
for j in range(5):
   f_cnt_guass.append([ f_cnt[i] + random.gauss(0, sigma[j]) for i in range(24)])
   print([ f_cnt[i] + random.gauss(0, sigma[j]) for i in range(24)])

for i in range(5):
   f_cnt = f_cnt_guass[i]
   fig = plt.figure()
```

```
    y1_list = [y1(i,400) for i in range(24)]

    x=np.array(range(len(y1_list)))
    y=np.array(y1_list)
    model=make_interp_spline(x, y)
    xs=np.linspace(0,23,100)
    ys=model(xs)
    plt.plot(xs, ys)
    plt.title("Curve")

    y2_list = [y2(i,400) for i in range(24)]
    x=np.array(range(len(y2_list)))
    y=np.array(y2_list)
    model=make_interp_spline(x, y)
    xs=np.linspace(0,23,100)
    ys=model(xs)
    plt.plot(xs, ys)
    print("choose 1: ",[i for i in range(24) if y1_list[i] >= y2_list[i]])
    print("choose 2: ",[i for i in range(24) if y1_list[i] < y2_list[i]])
    print("-----------")

"""## 蓄车池 因素分析"""

i = 2
def sol(n):
    return y1(i, n) - y2(i, n)
sol(734)
r = fsolve(sol, 0)
r
car_num = []
for i in range(24):
    car_num.append(fsolve(sol, 0))

car_num
fig = plt.figure(dpi=260)
x=np.array(range(len(car_num)))
y=np.array(car_num)
model=make_interp_spline(x, y)
xs=np.linspace(0,23,100)
ys=model(xs)
plt.plot(xs, ys)

plt.xticks([i for i in range(24)])
plt.fill_between(xs, ys[:,0], interpolate=True, color='blue', alpha=0.5)
plt.ylim(ymin=0)
plt.xlabel("time/(hour)")
plt.ylabel("NumofCarinQueue/(number)")
plt.annotate(s='A' ,xy=(15,200) ,xytext=(15,200), fontsize = 18)
plt.annotate(s='B' ,xy=(9,500) ,xytext=(9,500), fontsize = 18)
plt.savefig('./8_29_3.jpg',  bbox_inches = 'tight')
```

```
print([i for i in car_num])
```

附录四 对问题三的求解

```
lam = 1/6
mu = 1/4
eps = lam/mu

# K0
def K0(n):
    k0 = 1/sum([pow(eps,i)/np.math.factorial(i) + pow(eps, i+1)/(np.math.factorial(n) * (n-eps)) for i in range(n)])
    return k0

# Ls
def Ls(n):
    k0 = K0(n)
    ls = pow(eps, n+1)/(np.math.factorial(n-1)*(n - eps)*(n - eps)) * k0 + eps
    return ls

# res = [Ls(i) * 10 + i for i in range(1,10)]

d = [0.01, 0.1, 1, 10, 100]
print("d       n")
for di in d:
    min, pos = 100, 0
    for i in range(1,100):
        temp = Ls(i)*di + i
        if temp < min:
            min, pos = temp, i
    print("{}\t{}".format(di,pos))
```

附录五 对问题四的求解

```
l1 = 50 # 机场距离市区距离 km
v = 60 # 车速
t3 = l1/v * 3600 # 长途车从机场到市区
t = 20 # 当前时间
a = 1/72 # 在市区平均收益
ga = 0.2 # 定义短途车的目的地为长途车的 ga 倍
oil = 0.4795 # 单位公里的油费成本

def sol(t_):
    dt = t_ + (2*ga - 1) * t3
    e2_t3 = E2(l1, t)
    e1_dt = a * dt
    c_1 = a * (T1(t, n))
    e2_ga_t3 = E2(ga * l1, t)
    e1_t = a * t_
    c_oil = oil*l1
    # print("dddd")
```

```
    # print(e2_t3,e2_ga_t3 - c_oil)
    return e2_t3 + e1_dt - c_1 - e2_ga_t3 + e1_t + c_oil

res = []
for n in range(100, 600, 50):
    temp = []
    for t in range(24):
        temp.append(fsolve(sol, 0))
    res.append(temp)

len(res)

for j in range(10):
    res[j] = [i[0]/60 + 100 for i in res[j]]

res

for i in range(5):
    x=np.array(range(len(res[i])))
    y=np.array(res[i])
    model=make_interp_spline(x, y)
    xs=np.linspace(0,23,100)
    ys=model(xs)
    plt.plot(xs, ys, label='CarnumInPool = '+ str(100 +i*50))
plt.legend()
plt.title("问题 4 结果对蓄车池车辆数量的灵敏性分析")
plt.xlabel("")

c_cnt = [C(i)/200 for i in range(24)]
# plt.bar(range(len(c_cnt)), c_cnt)
x=np.array(range(len(c_cnt)))
y=np.array(c_cnt)
model=make_interp_spline(x, y)
xs=np.linspace(0,23,100)
ys=model(xs)
plt.plot(xs, ys)
```