机场出租车问题的研究

摘要

出租车是乘客去往飞机场的主要交通工具。送客到机场的出租车司机都将面临：在候车区等待载客，还是空返市区拉客的选择。本文基于深圳市的实际数据，在考虑机场乘客数量和出租车收益的基础上，建立了出租车收益和损失比的决策模型。同时，讨论了机场乘车区有两条并行车道，乘车点的设置；对短途载客的出租车，设计了可行的优选方案。

针对问题一，分析影响出租车司机决策的因素，建立模型并给出司机的选择策略。先以等待载客排队时长和载客的可能收益定义出租车单位时间的收益率，再以空返拉客空载费用和潜在的载客收益定义出租车单位时间的损失率，建立以同一时刻收益率与损失率之比的决策模型，根据比值大小帮助司机决定是等待载客还是空返拉客。

针对问题二，收集国内某一城市相关数据，应用于所建模型。采集了 2011 年 4 月 18 日深圳市宝安机场进港航班数量和旅客人数，及当天出租车实时动态行车数据和出租车的运行价格标准。统计全天 24 个时段出租车平均速度、到达机场的数量和市区的载客率，以及 24 个时段航班数量、可能到港人数、可能乘坐出租车的旅客人数。基于问题一建立的模型，得到当天每个时段收益率与损失率比值。结果显示在 8：00-10：00 和 22：00-23：00 时间段内，司机选择在机场等候乘客的收益更高；在其余时间段，直接放空返回市区收益更高。基本与实际相符。

针对问题三，机场乘车区有两条并行车道，上车点的合理设置。关于如何设置上车点使其系统总的乘车效率最高。考虑到出租车需要在机场蓄车池中排队，建立 M/M/2 排队论模型，对比在原始情况下两条并行车道和设置引流车道后的两种系统效率指标，可以得出后一种的乘车效率更高。

针对问题四，为了研究如何使机场载客的出租车司机收益尽量均衡，建立基于 M/M/2 的优先排队模型，对短途返回的出租车赋予优先排队权力，设置专门的排队车道，尽量缩短其排队时间，进一步增加了短途返程出租车司机的收益。

关键词：机场；出租车；收益亏损模型；排队模型；优先权

1 问题重述

出租车是乘客去往飞机场的主要交通工具。送客到机场的出租车司机都将面临：在候车区等待载客，还是空返市区拉客的选择。

1.根据机场乘客数量以及出租车收益等影响因子，进行司机的选择决策模型的建立，同时说明司机的选择策略。

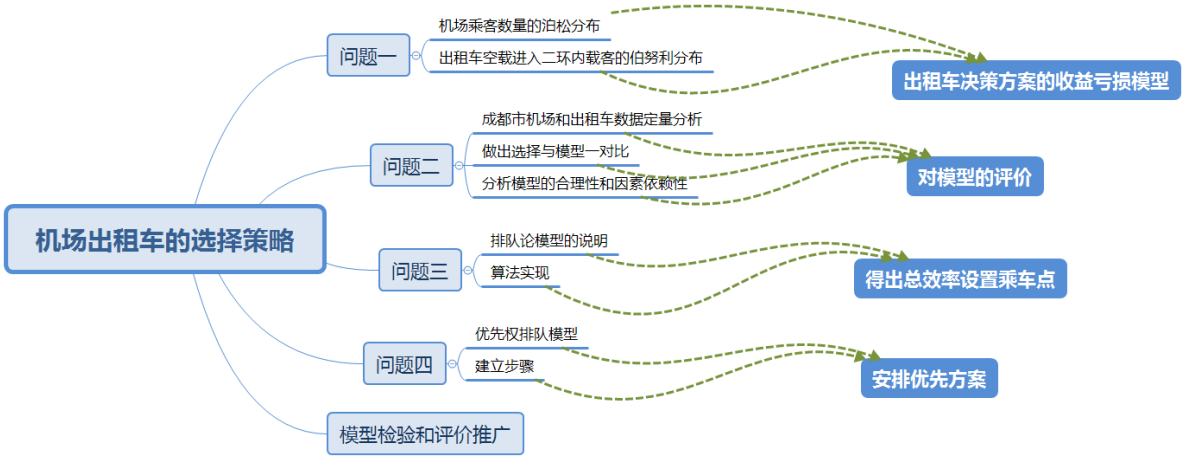
2.参照搜集到的国内机场和出租车有关数据，对建立过的模型进行分析，说明其合理性和对有关影响因子的依赖性，同时给出该机场司机的选择方案。

3.如何设置乘车点以及安排出租车和乘客使得总乘车效率最高。

4.试给出一个“优先”方案使短途载客又返程回机场的出租车司机收益均衡。

2 问题分析

2.1 总体分析



图(1)总体分析思维导图

2.2 具体分析

对于问题一，要得出出租车的决策模型，就需要建立一个衡量指标进行量化。

这里先是根据机场乘客数量的泊松分布、出租车空载进入二环内载客的伯努利分布得到相应的乘客率、空载率，分为某些时间区间，以及对应时间段的收益和潜在收益，由此计算各个时间段的出租车收益率和损失率之比，根据大小来选择，若收益率大于损失率则选择方案 A，否则选择方案 B。

对于问题二，量化已经得到的深圳宝安国际机场[2]和出租车的 GPS 数据[1]，首先根据 GPS 数据画出深圳市出租车轨迹热力图和分布散点图。然后根据不同时间段的出租车分布以及乘客上车数，看总体的大概率趋势，借此给出深圳市保安机场出租车的选择方案，并由此对比问题一模型，用量化的数据说明模型一的

合理性和对相关因素的 依赖程度。

对于问题三，借鉴排队论模型，利用 M/M/s 等待制排队模型算法[3]，计算系统的总效率指标。原始的系统是未设置上车点，只有一个服务台，利用该算法得到原始的效率指标。之后，设置上车点，在两条并行车道上分别加上一条引流服务点，再次计算得到设置后的效率指标，两者对比，可以看出设置上车点后的总乘车率最高。

对于问题四，参照问题三的排队论模型，进一步优化，设定优先权，得到优先权排队模型[4]。对于短途载客又返程的出租车，在返回机场时，设置专门的排队车道，依照返回车辆的最近一次行驶里程判断当前车辆是否是短途载客，如果是就进入优先车道，否则按照正常车道排队。这样设置车道，就能减少短途车排队时间，进一步提高其收益，使得收益尽量均衡。

3 模型假设

1.假设进港人流量常年稳定变化；
2.假设人流量和出租车量持续不断；
3.假设所有的数据来源准确、真实、可靠；
4.假设出租车行驶到二环和二环到市中心的行驶里程一定；
5.假设出租车的收益不受其它不可控因素的影响；

4 符号说明

符号	含义
f_1	收益率
f_2	损失费用率
f	收益率/损失费用率
μ	服务率，单位时间内完成服务的顾客
λ	顾客到达率，单位时间内到达系统的顾客数

5 模型建立和求解

5.1 问题一 模型的建立与求解

5.1.1 机场出租车决策方案的收益亏损模型

机场出租车面临两种选择，一是直接放空返回市区，损失潜在的载客收益，付出空载费用；二是进入机场蓄车池排队进场载客，付出时间成本，无空载费用。两种方案各有利弊，我们通过建立两种决策的收益亏损模型，综合分析出租车排队载客时的等候时

间以及空载返回时因未载客而未得到的收益（即定为亏损）。构建量化指标，分析各时间段的收益、亏损比，为出租车司机做决定提供选择策略。

下面，我们给出收益与亏损的具体定义

（1） 出租车在机场“蓄车池”排队载客返回市中心的收益率f_1为:

$$f_1 = \frac{C_1}{T_1 + T_2} \tag{1}$$

其中：C_1为出租车司机接客的收益，T_1为出租车等客的时间，T_2为出租车接客后到达市中心的时间。机场乘客数量符合泊松分布，T_1随机场乘车乘客数量的变化而变化。假设出租车从机场到市中心的速度恒定，时间T_2也为定值。

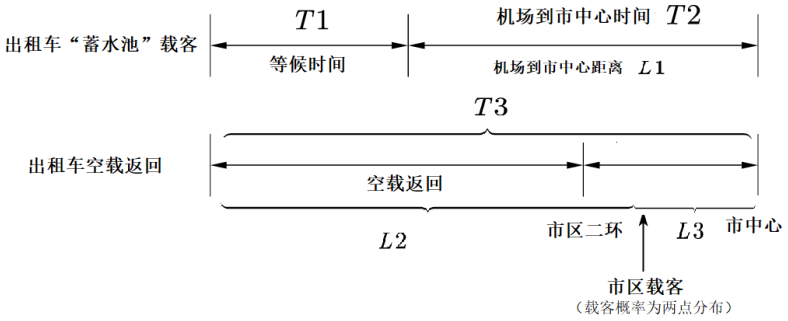
（2） 出租车直接返回市中心的损失费用率f_2为：

$$f_2 = \frac{C_2}{T_3} \tag{2}$$

其中：C_2为出租车的亏损值，即出租车从机场空载直到再次接客的这段里程载客的收益。T_3为出租车从机场直到再次接客的时间。我们假设出租车进入市区二环后才能接到客人，并且接客满足随机过程的伯努利分布，其概率随市区每时段的客流量的变化而变化。

5.1.2 模型的求解

关于两种决策的方案，我们给出收益和损失的各参数以及比率的求解，最终得出各个时段的收益亏损比，给出各个时段出租车司机的决策。



图(2)：出租车两种决策方案各参数的图示

1） 出租车在机场“蓄车池”排队载客的收益率计算

(1) 出租车的等候时间计算

出租车在机场的“蓄车池”车道等候乘客，则等候时间与每一时段人流量的变化密切相关。在航班进港高峰时段，人流量大，自然乘坐出租车的乘客也就越多，出租车等候时间仅为乘客排队时间。在进港的普通时段，等候时间包括乘客的排队时间和车辆静置等待时间。

我们根据机场航班进港大数据可以分析出人流密集的高峰时段以及排队时间和出租车的平静置均等待时间。设乘客排队时间为t_1，出租车静置等待时间为

t_2，则总的出租车侯客时间为：

$$T_1 = \begin{cases} t_1 & \textit{高峰时段} \\ t_1 + t_2 & \textit{普通时段} \end{cases}$$

(2) 出租车从机场到市中心的运行时间

设机场到市中心的距离为L_1，该时段的出租车平均速度为v_1，即$T_2 = \frac{L_1}{v_1}$。

(3) 出租车收益计算

设司机的收益为C_1，根据出租车的运价标准，得出不同时段的收益，C_1随距离 L 的变化而变化，得出下式

$$C_1(L) = \begin{cases} 10 & L \le 2 \\ 10 + 2.6(L-2) & L \ge 2 \end{cases}$$

出租车的计价标准与时间相关，在夜间(23 时-6 时)出租车会增加一定比率的夜班附加费，设为 n。则在夜间收益为 $C_1'(L) = \text{n} \times C_1(L)$。

(4) 出租车的收益率

$$f_1 = \frac{C_1}{T_1 + T_2}$$

2) 出租车直接返回市中心的损失费用率f_2为：

(1) 出租车的亏损

出租车从机场返回到市区二环处，开始载客，载客率受市区高峰时段的影响，符合两点分布。我们设载客率P_i (i=0,1,2…23)，根据出租车大量载客数据分析出每个时段的载客率。L_2是机场到市区二环的距离，L_3是二环到市中心的距离。则出租车的亏损值如下式：

$$C_2(L) = 10 + m(L_2 - 2) + L_3(1 - P_i)$$

(2) 出租车的时间计算

(5) T_3是出租车从机场到市中心的时间。计算方法和制约因素与 1)中出租车从机场到市中心的运行时间求解方法相同。

(3) 出租车的亏损率

$$f_2 = \frac{C_2}{T_3}$$

（3） 出租车两种方案的决策博弈

由(1)(2)公式，我们得到了出租车“留”的收益率f_1 和“去”的损失率f_2。

我们设两种决策比率

$$f = \frac{\mathrm{f_1}}{\mathrm{f_2}} \tag{3}$$

$f > 1$ 即此时刻收益率大于损失率，在机场等待载客对司机更有利。

$f < 1$ 此时刻损失率大于收益率，直接空载回市区载客对司机更有利。

$f = 1$ 此时刻损失率等于收益率，在机场等待载客和直接空载回市区载客对司机收益

影响不大，两种决策皆可。

5.2 问题二 模型的建立和求解

5.2.1 数据采集与数据处理

1）数据采集

我们采集了深圳宝安国际机场的航班进港数据[2]和深圳市出租车实时动态行车数据[1]，运用 GPS 定位设备记录出租车的行驶位置、速度、方向和载客状态的数据。航班进港数据为航班号、实际抵达时间、座位数以及实际乘客人数。

对机场进港航班数据的分析，我们可以获得一天中每时段航班进港数量，预估出机场进港的人流量和乘坐出租车的乘客的数量。其数据的基本格式为

表(1)航班进港数据样例

航班号	到达机场	实际到达时间	最大座位数	实际销售人数
ZHxxx	深圳宝安国际机场	43647.19028	248	51
CZxxx	深圳宝安国际机场	43647.25694	221	90
0xxx	深圳宝安国际机场	43647.14097	267	132
0xxx	深圳宝安国际机场	43647.13681	180	153

通过对出租车实时行车动态数据的分析，我们可以得出深圳市出租车的分布以及运营情况和机场附近出租车的运行情况，进而分析机场附近出租车在抵达机场之后两种决策的选择情况。为分析问题一中模型的合理性和相关因素的依赖性奠定了基础。其基本数据格式为

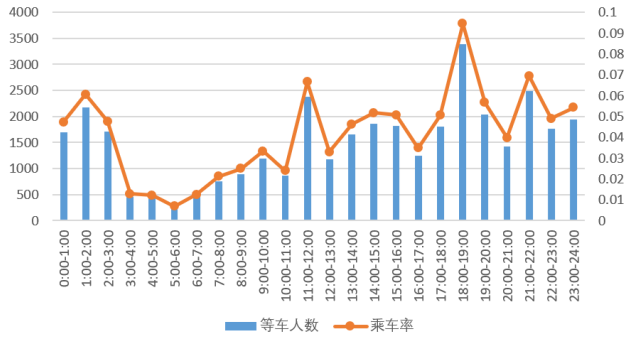
表(2)航班进港数据样例

车牌号	时间	经度	纬度	载客状态	车速
粤 Bxxx	2011/4/18 0:00	114.0815	22.54805	0	33
粤 Bxxx	2011/4/18 0:00	113.9152	22.54122	1	45
粤 Bxxx	2011/4/18 0:00	113.9058	22.55533	0	0
粤 Bxxx	2011/4/18 0:00	114.1173	22.54618	1	0

载客状态为 0 时表示未载客，1 表示载客。

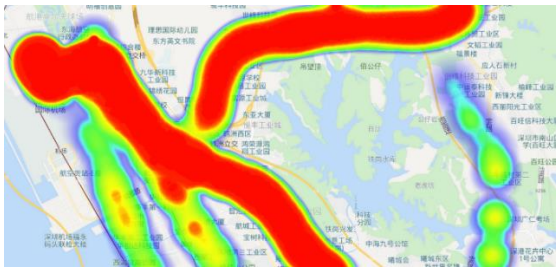
2）数据处理

(1) 我们对机场航班进港数据按时间段进行筛选，得到 24 个时段的进港客流量数据，并将客流量大于 1500 时设为高峰时段，由数据可知，0：00-3：00、11：00-16：00、17：00-20：00 与 21：00-0：00 均为客流高峰时段。客流量与乘坐出租车的人数是正相关的[9]。

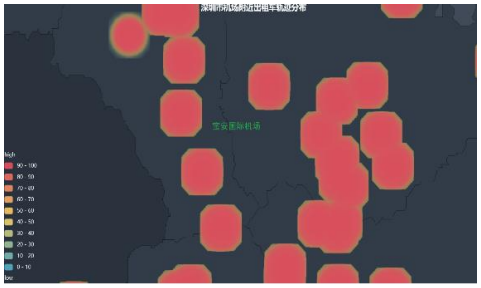


图(3) 机场 24 时段的客流量

(2) 我们对深圳市出租车的 GPS 定位进行分析，得出了深圳出租车轨迹热力图，以及机场附近的出租车轨迹分布图。

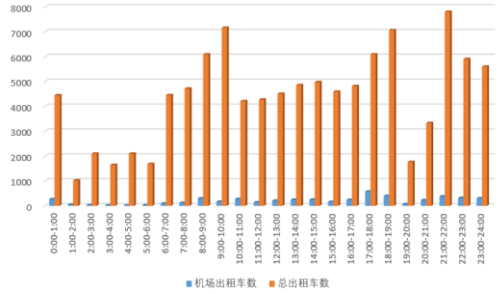


图(4) 深圳出租车轨迹热力图

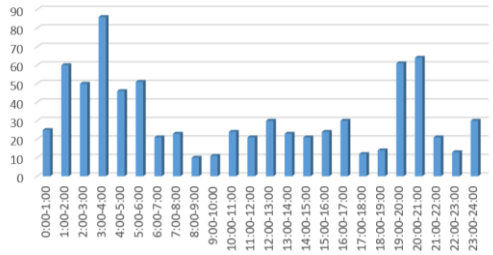


图(5) 机场附近的出租车轨迹分布图

（3）通过筛选得到各时段的出租车在营运的数量，通过出租车经纬度来分析宝安国际机场附近的每时段出租车的数量，分析可得出每时段机场出租车的占比。



图(6)各时间段机场出租车与总在营出租车数量



图(7) 各时段出租车平均速度

(4) 通过分析每个时段每辆在营出租车的车速，可以得出在每时段出租车的平均速度,如图(7)。

5.2.2 深圳宝安国际机场出租车决策分析

在问题一模型给出的决策分析的基础上，将深圳出租车运营情况以及宝安机场的航班数据带入模型中，分析出租车司机的决策与模型的相关度。

1）模型中实际参数的数据收集

(1)每时段市区载客率

我们由深圳市出租车每时段载客车辆与总的在营车辆可以求得，每时段的载客概率。

表（3） 每时段的载客概率P_i

时间段	0:00-1:00	1:00-2:00	2:00-3:00	3:00-4:00	4:00-5:00	5:00-6:00	6:00-7:00	7:00-8:00	8:00-9:00	9:00-10:00	10:00-11:00	11:00-12:00
载客概率	0.36	0.29	0.16	0.19	0.15	0.23	0.3	0.38	0.86	0.78	0.65	0.6
时间段	12:00-13:00	13:00-14:00	14:00-15:00	15:00-16:00	16:00-17:00	17:00-18:00	18:00-19:00	19:00-20:00	20:00-21:00	21:00-22:00	22:00-23:00	23:00-24:00
载客概率	0.69	0.54	0.34	0.35	0.55	0.67	0.73	0.63	0.52	0.87	0.34	0.31

(2)收益、损失值计算

表（4） 每时段的载客概率

起步价	10 元/2 公里
里程价	2.6 元/公里
夜间附加费	按起步价和里程价加收 30%

收益的计算为：

$$C_1(L)=\begin{cases}10 & L\le 2\\ 10+2.6(L-2) & L\ge 2\end{cases}$$

夜间比率 n=1.3。$C_1'(L)=1.3C_1(L)$。

亏损值的计算为：

$$C_2(L)=10+2.6(L_2-2)+L_3(1-P_i)$$

(3)机场“蓄车池”出租车平均等待时间

我们根据机场出租车大量数据的分析，得到在机场等待客人的平均等待时间为 0.3 小时。排队的平均等待时间为 0.1 小时。

$$T_1=\begin{cases}0.1h & \textit{高峰时段}\\ 0.1+0.3h & \textit{普通时段}\end{cases}$$

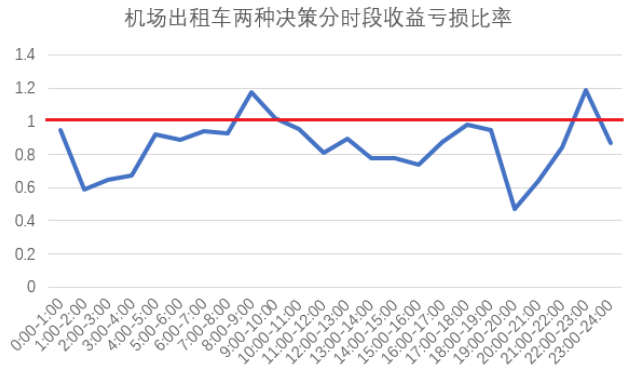
宝安机场到深圳市中心的距离数据

表(5) 机场到深圳市中心的距离数据

机场到二环的距离	二环到市中心
26KM	11KM

2）收益、亏损值的计算

根据以上数据分析和处理以及所给出的实际数据，运用 Excle 将实际数据代入问题一模型的公式得到如下结果



图(8)收益与亏损比率

由收益亏损比率数据可知(具体数据置于支撑材料)，在 8：00-10：00 和 22：00-23：00 时间段内，司机选择在机场等候乘客的收益更高；在其余时间段，直接放空返回市区收益更高。这与现实情况基本相符合。

5.2.3 模型合理性与相关因素依赖性的研究

经过对比收益率和亏损率的两个变量，运用 PyCharm 软件进行相关性的分析可以得到下表：

表(6) 相关性分析

相关性分析	收益率	亏损率
收益率	1	0.932044
亏损率	0.932044	1

可以看出收益率与亏损率的相关系数接近 1，两者的相关性很强，也就是说影响其收益的相关因素对二者的影响相似，说明该模型对相关因素的依赖性不强。

5.3 问题三 模型的建立和求解

5.3.1 问题的分析

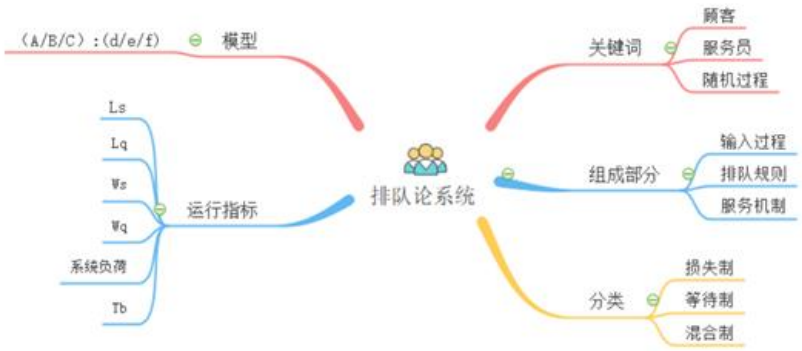
在机场的进港乘客要乘坐出租车，往往会出现排队的情况，通过运用排队论模型的有关原理建立该问题的模型。基于该模型的求解，根据排队论的有关算法，从而得出设置不同上车点时的效率，进行系统总效率指标的对比，最后确定最佳上车点的位置。

5.3.2 排队论

排队论模型分为两大类：M/M/1 和 M/M/s，即“多队列”排队模型和“单队列”排队模型[3~5]。模型中两个 M 表示相继到达乘客的间隔时间与服务时间均服从指数分布。此模型表示，乘客独自抵达，前后抵达时间的间隔服从

参数为λ的负指数分布，s 表示服务台数，各个服务台的服务时间相互独立，且服从参数为μ的负指数分布，乘客抵达时，如果有不工作的服务台，则马上接受服务，否则乘客站成一队等待，并且顾客中途不会离开，一直到乘客接受完服务，才离开系统。

排队论系统模型图：



图(9) 排队论导图

该系统一般有两个输入参数：

λ：顾客到达率，单位时间内到达系统的顾客数。

μ：服务率，单位时间内完成服务的顾客数。

结合问题二出租车数据的分析，可得到一段时间内到达机场的顾客数和出租车载客的数量，由此可以推出λ 与 μ的值

排队论求解步骤：

图(10) 排队论求解过程导图

M/M/s 等待制排队模型的求解方法如下：

当系统达到稳定时，记$P_n = P\{N = n\}$(n=0,1,2,…) 为系统达到稳定状态

后队长 L 的概率分布对于服务台个数为 s 的多服务台系统,有

$$\lambda_n=\lambda(n=0,1,2,\cdots)$$

和

$$\mu_n=\begin{cases}n\mu, & n=1,2,3,\dots s\\ s\mu, & n=s,s+1,\dots\end{cases}$$

记$\omega_s=\frac{\omega}{s}=\frac{\lambda}{s\mu}$, 当$\omega_s<1$，根据生灭过程的推导结果，有以下公式，

$$S_n=\begin{cases}\frac{(\frac{\lambda}{\mu})^n}{n!}, & \\ \frac{(\frac{\lambda}{\mu})^s}{s!}(\frac{\lambda}{s\mu})^n=\frac{(\frac{\lambda}{\mu})^n}{s!s^{n-s}}, n\geq s & \end{cases}\quad n=1,2,\cdots,s$$

故

$$P_n=\begin{cases}\frac{(\omega)^n}{n!}P_0, & n=1,2,\dots,s\\ \frac{(\omega)^n}{s!\,s^{n-s}}P_0, & n\geq s\end{cases}$$

其中

$$P_0=[\sum_{n=0}^{s-1}\frac{(\omega)^n}{n!}+\frac{(\omega)^s}{s!\,(1-\omega_s)}]^{-1}$$

联立以上的式子，可以得到系统中的顾客数量为 m 的概率，当$n\geq s$时，后续到达的顾客必须排队等待，记

$$s(s,\omega)=\sum_{n=s}^{\infty}P_n=\frac{(\omega)^s}{s!(1-\omega_s)}P_0$$

表示 M/M/s 排队系统中，顾客必须等待的概率。在稳定状态下，M/M/s 排队系统的平均队长N_q可以用下式表示：

$$N_q=\sum_{n=s+1}^{\infty}(n-s)P_n=\frac{P_0\omega^s}{s!}\sum_{n=s+1}^{\infty}(n-s)\omega_s^{n-s}=\frac{P_0(\omega)^s}{s!}\frac{d}{d\omega_s}\left(\sum_{n=1}^{\infty}\omega_s^n\right)$$

$$=\frac{P_0\omega^s\omega_s}{s!\,(1-\omega_s)^2}$$

或

$$N_q=\frac{s(s,\omega)\omega_s}{1-\omega_s}$$

当系统进入稳定时，系统中处于服务状态的服务台个数的平均值与服务强度ω相等，推导过程如下：$s_0=\sum_{n=0}^{s-1}nP_n+s\sum_{n=s}^{\infty}P_n=\sum_{n=0}^{s-1}\frac{n(\omega)^n}{n!}P_0+s\frac{\omega^s}{s!(1-\omega_s)}P_0=P_0\omega[\sum_{n=1}^{s-1}\frac{(\omega)^{n-1}}{(n-1)!}+\frac{(\omega)^{n-1}}{(s-1)!(1-\omega_s)}]=\omega$，通过此式可知，处于服务状态的服务台个数的平均值不依赖于服务台个数 s。

5.3.3 模型的求解

（1）在初始状况下，只有一个服务点。

为了验证应用该模型会使得该机场两条并行车道的乘车效率最高，在此假设：若出租车到达流为泊松流，乘客源源不断的进来乘车，车辆的泊松流其强度为 λ=2.1 辆/分钟，并行车道的通关能力为 2.5 辆/分钟。通过时间为指数分布，平均每辆的通过时间为 0.4 分钟。在初始上车点情况下应用以上公式，用 CodeBlocks 软件，利用 C++语言编程得到最终结果：

表（7）系统各项效率指标

输入各项参数	
服务员个数	1
顾客流强度	2.1
服务台能力	2.5
系统效率指标	
损失概率	0
系统的相对通过能力	1
系统的绝对通过能力	2.1
系统内排队顾客的平均数	4.41
顾客的平均排队时间	2.1
占用服务员的平均数	0.84
系统内顾客的平均数	5.25
顾客在系统中平均逗留时间	2.5

（2）设置上车点后，在两条并行车道的两边分别加上引流服务点，之后再次计算系统效率指标如下表：

表（8）系统各项效率指标

输入各项参数	
服务员个数	2
顾客流强度	2.1
服务台能力	2.5
系统效率指标	
损失概率	0
系统的相对通过能力	1
系统的绝对通过能力	2.1
系统内排队顾客的平均数	0.070476
顾客的平均排队时间	0.03356
占用服务员的平均数	0.84
系统内顾客的平均数	0.910476
顾客在系统中平均逗留时间	0.43356

通过对以上两方案的系统各项效率指标对比可以得出，在第二种方案中，设置引流点后，平均排队数及等待时间比方案一少，总的乘车效率更高。

5.4 问题四 模型的建立和求解

5.4.1 问题的分析

基于问题三建立起来的机场出租车排队论模型，使得在并行车道设置服务点之后，系统的总乘车效率变高。

继续考虑到从机场载客到市区的问题，会出现出租车司机的收益的不均衡问题。比如因为出租车司机不能选择和拒绝乘客，然而行驶距离又远近不一。这样就会导致出租车司机收益的差距，对于这部分短途车来说不公平的程度加大。在此，可以依照问题三，利用已经建立起来的排队论模型，进一步优化和升级，给短途载客又返回的出租车司机一定的“优先排队权[4]”。获得优先权的出租车能进一步弥补收益差距。

5.4.2 优先权排队模型背景

例如，有部分顾客如急诊患者要比普通患者急需救助，这时便有优先服务的产生[4]。强占型和非强占型两种是优先权排队的两种方式。

有 c 个服务台，w 是系统总量，顾客到达的时间为参数λ的负指数分布，服务时间为参数为μ的负指数分布，由此 M/M/c/w 系统形成。设H_l表示事件：{系统l有个乘客},H_l=P（M_l）：有l个乘客的概率。

5.4.3 模型建立

1、 系统中存在l个服务窗在为顾客服务的概率为M_l

$$M_l=\begin{cases}\dfrac{w^l\omega^l}{l!}p_0 & 0\le l\le c\\ \dfrac{w^w\omega^l}{l!}p_0 & c\le l\le c\end{cases}$$

其中，

$$M_0=\begin{cases}[\sum_{l=0}^{c-1}\frac{c^l\omega^l}{l!}+\frac{c^c\omega^c(1-\omega^{w-c+1})}{c!(1-\omega)}]^{-1}\ , & \omega\neq 1,\\ [\sum_{l=0}^{c-1}\frac{c^l}{l!}+\frac{c^c(w-c+1)}{l!}]^{-1}\ , & \omega=1\ ;\end{cases}$$

2、系统的损失概率：$P_w=p_w=(\frac{c^c\omega^w}{c!})M_0$;

3、 系统的相对通过能力：H=1−p_w=1−$(\frac{c^c\omega^w}{c!})M_0$;

4、 单位时间平均损失的顾客数：$\lambda_T=\lambda P_w=\lambda p_w=(\frac{\lambda c^c\omega^w}{c!})M_0$;

5、 单位时间进入系统的顾客数：$\lambda_k=\lambda(1-p_w)=\lambda H$;

6、 平均服务中的服务台数：$T_d=l$=$\sum_{l=0}^{c-1}lp_l$+n$\sum_{l=c}^{w}p_l$=cω(1−p_w)=λ^w/μ;

7、 单位时间平均进入系统的顾客数：

$$T_{q0}=\begin{cases}\frac{C^c\omega^{c+1}p_0}{c!(1-\omega)^2}[1-(w-c+1)+(w-c)\omega^{w-c+1}]\ , & \omega\neq 1,\\ \frac{c^c}{2c!}(w-c)(w-c+1)p_0\ , & \omega=1;\end{cases}$$

8、 平均队长：$T_{eo}=T_{q0}+T_w=T_{q0}+\lambda_d$，其中$\lambda_d=\mu c-\sum_{l=0}^{c-1}(c-l)p_l$;

其中$\lambda_d=\mu c-\sum_{l=0}^{c-1}(c-l)p_l$;

10、 平均逗留时间：$F_{e0}=\frac{T_{eo}}{\lambda_d}=T_{qo}+\frac{1}{\mu}$;

11、 平均排队等待时间：$T_{qo}=T_{q0}/\lambda_d$;

接下来定义非强占有限优先权 M/M/c/w 排队系统：

乘客有两级分类：优先权级，无优先权级。

对比问题三模型，增加了以上假设。

顾客到达系统服从参数为λ_i的泊松分布，λ_i为第 i 级顾客的平均到达率：

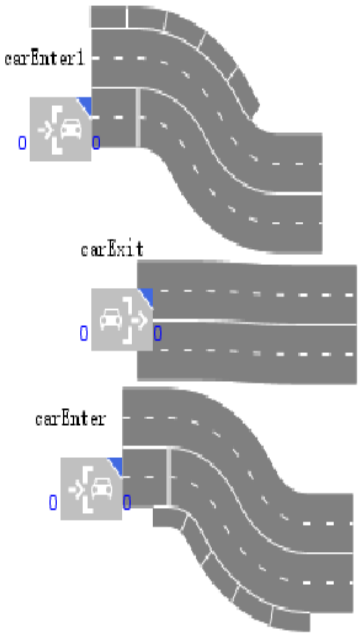
$\mathrm{P}(\lambda_i,t,m)=\frac{(\lambda_i t)^m}{m!}e^{-\lambda_i t}$， m=0，1，2，…；t> 0;

由第一问算得的收益率可以求出出租车的平均收益Y_1.由此可以得到出租车收益的方差$\mathrm{D(X)}=（\mathrm{Y}-\mathrm{Y_1}）^2$,对不同的优先权方案，通过对方差的分析知以下给出的方案方差最小，使得收益尽量均衡。

5.4.4 模型求解

根据以上系统和问题三的排队论模型，针对问题四，对短途载客又返回机场的出租车设定专门的一个优先排队通道。

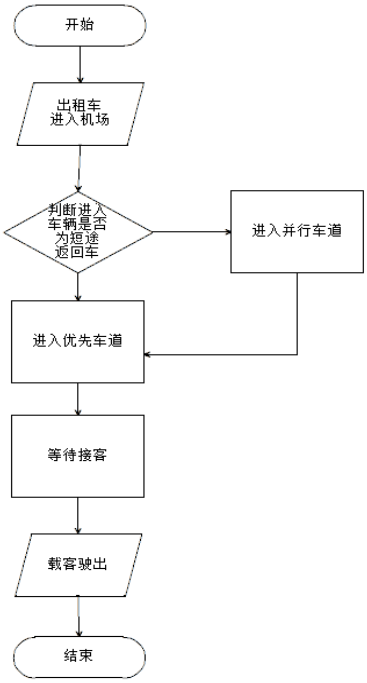
如下图：



图（11）出租车优先车道示意图

这样可以使得短途返回的出租车利用 carExit 车道优先进入车道，进一步收益均衡。

优先权排队模型让返程的出租车尽量优先载客，但是必须是进入特设车道，原来的车道已经有出租车等待接客，不能插入到原车道。此优先排队模型只是针对短途返程的出租车群体，进入原来的车道就要等待许多已经就位的车辆驶出，而设立优先车道使得等待率降低也就是有限优先权系统的效率提高，出租车载客次数增加，收益比之前未设立优先车道要高。



图（12） 方案执行流程图：

由此，可行的方案给出。

通过公里数来衡量出租车的短途载客：

以机场为中心，10 公里为半径的范围，作为出租车短途载客的衡量标准。当出租车载客的目的地在此范围内时，可以享受机场的“优先”服务。

6 模型检验

对于问题一，首先建立了基于收益率和损失率的出租车司机选择决策模型，用单位时间内的出租车载客的费用作为出租车的收益率，用相同单位时间内的出租车空载费用作为损失率，两者的比值大小决定选择方案。在此模型上，依据问题二的数据，进行了数据的验证。虽然得到选择结果，但是没有考虑影响出租车收益的其它因素，比如排队人数、等待时间等。于是，另一种多目标模糊决策模型 更适合研究该机场出租车收益影响因素指标，并利用熵权法进行各项指标的权重赋值，对模型的影响因子有很好的解释。

对于问题二，利用模型一，代入数据得到的值在忽略相关因素的情况下较为准确。

对于问题三，通过排队论模型，依照相关算法得出系统的效率指标，能较好反应上车点的不同对系统总乘车率的影响，由此也能得出上车点最优位置。

对于问题四，在排队模型的基础上进一步优化，增加了优先权，对于短途返程出租车来说，拥有优先进入车道接客的权利，使得其收益更加均衡。

7 模型的评价和推广

7.1 模型缺点

1.首先对于选择决策模型来说，没有实际考虑到影响司机收益的多种指标，在这可以利用上述的多目标模糊决策模型。

2.在问题二中，选取的深圳宝安机场和出租车的数据，没有对得出的乘车率做出定量分析和检验，只是估计值，缺少科学性。

3.问题三中排队论模型是在假设乘车人数及车流强度一定值下的求解，没有考虑到变化时的情形，缺少准确性和普遍性。

4.基于模型二，建立起的模型三是关于优先权的排队模型，只是针对短途出租车系统的优先排队模型，确实模型的普适性。

7.2 模型优点

1.对比多目标模糊决策模型，问题一所建立的选择策略模型更加简单明了，计算方便，具有可推广性。

2.搜集到的深圳宝安国际机场和出租车数据量庞大，更具有代表性。

3.排队论模型和优先权模型，能够多方位观察系统的效率指标，结果的准确性较高。

8 参考文献

[1] 深圳市出租车 GPS 数据分析深圳道路交通情况[DB/OL].http://math.tongji.edu.cn/model/camp2011D.html

[2] 飞常准（Variflight.com）数据平台[DB/OL].https://data.variflight.com/

[3]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.

[4]黄业文,邝神芬.非强占有限优先权 M/M/c/w 排队系统[J].应用概率统计,2018,34(04):364-380.

[5]林思睿. 机场出租车运力需求预测技术研究[D].电子科技大学,2018.

[6]潘晓芳,周顺平,杨林,万波.OD 约束的出租车经验模型与路径规划[J].华南理工大学

学报(自然科学版),2017,45(08):57-64+83.

[7]黄淑祥,杨洪礼,李红玉,王艳慧,刘洪霞.“清除太空垃圾”是否存在商机呢?[J].数学建模及其应用,2016,5(02):41-49+87.

[8]陈蓉素.多目标模糊决策算法研究[J].计算机工程与应用,2014,50(18):67-69.

[9]齐林. 基于 GPS 数据的出租车交通运行特性研究及应用[D].哈尔滨工业大学,2013.

[10]陶化成.处理多目标决策问题的一个方法[J].系统工程理论与实践,1982(02):25-28.

9 附录

源程序 1：深圳市双流机场出租车热力图

```
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
    <script type="text/javascript" src="http://api.map.baidu.com/api?v=1.4"></script>
    <script type="text/javascript" src="http://api.map.baidu.com/library/Heatmap/2.0/src/Heatmap_min.js"></script>
    <title>热力图功能示例</title>
    <style type="text/css">
        ul,li{list-style: none;margin:0;padding:0;float:left;}
        html{height:100%}
        body{height:100%;margin:0px;padding:0px;font-family:"微软雅黑";}
        #container{height:500px;width:100%;}
        #r-result{width:100%;}
    </style>
</head>
<body>
    <div id="container"></div>
    <div id="r-result">
        <input type="button" onclick="openHeatmap();" value="显示热力图"/><input type="button" onclick="closeHeatmap();" value="关闭热力图"/>
    </div>
</body>
</html>
<script type="text/javascript">
var map = new BMap.Map("container");          // 创建地图实例
var point = new BMap.Point(103.95223, 30.57428);
map.centerAndZoom(point,15);              // 初始化地图，设置中心点坐标和地图级别
map.enableScrollWheelZoom(); // 允许滚轮缩放
```

var points =[{"lat":30.669775,"lng":104.055413,"count":50.000000},0.000000},
{"lat":30.669775,"lng":104.055481,"count":50.000000},
{"lat":30.669775,"lng":104.055518,"count":50.000000},
{"lat":30.669775,"lng":104.055570,"count":50.000000},
{"lat":30.669775,"lng":104.055570,"count":50.000000},
{"lat":30.669775,"lng":104.055570,"count":50.000000},
{"lat":30.669775,"lng":104.055570,"count":50.000000},
{"lat":30.669775,"lng":104.055570,"count":50.000000},
{"lat":30.669776,"lng":104.055308,"count":50.000000},
{"lat":30.669776,"lng":104.055337,"count":50.000000},
{"lat":30.669776,"lng":104.055504,"count":50.000000},
{"lat":30.669776,"lng":104.055504,"count":50.000000},
{"lat":30.669776,"lng":104.055504,"count":50.000000},
{"lat":30.669776,"lng":104.055504,"count":50.000000},
{"lat":30.669777,"lng":104.055370,"count":50.000000},
{"lat":30.669777,"lng":104.055370,"count":50.000000},
{"lat":30.669777,"lng":104.055505,"count":50.000000},
{"lat":30.669778,"lng":104.055312,"count":50.000000},
{"lat":30.669778,"lng":104.055419,"count":50.000000}}
if(!isSupportCanvas()){ alert('热力图目前只支持有 canvas 支持的浏览器,您所使用的浏览器不能使用热力图功能~') } //详细的参数,可以查看 heatmap.js 的文档 https://github.com/pa7/heatmap.js/blob/master/README.md //参数说明如下: /* visible 热力图是否显示,默认为 true
opacity 热力的透明度,1-100
radius 势力图的每个点的半径大小
gradient {JSON} 热力图的渐变区间 . gradient 如下所示
{ .2:'rgb(0, 255, 255)', .5:'rgb(0, 110, 255)', .8:'rgb(100, 0, 255)' }
其中 key 表示插值的位置, 0~1. value 为颜色值. */ *heatmapOverlay = new BMapLib.HeatmapOverlay({"radius":50}); map.addOverlay(heatmapOverlay); heatmapOverlay.setDataSet({data:points,max:100}); //是否显示热力图 function openHeatmap(){ heatmapOverlay.show(); } function closeHeatmap(){ heatmapOverlay.hide(); } closeHeatmap(); function setGradient(){ /*格式如下所示: { 0:'rgb(102, 255, 0)', .5:'rgb(255, 170, 0)', 1:'rgb(255, 0, 0)' }*/ var gradient = {}; var colors = document.querySelectorAll("input[type='color']"); colors = [].slice.call(colors,0);
colors.forEach(function(ele){ gradient[ele.getAttribute("data-key")] = ele.value; }); heatmapOverlay.setOptions({"gradient":gradient}); }
//判断浏览区是否支持 canvas function isSupportCanvas(){ var elem = document.createElement('canvas'); return !!(elem.getContext && elem.getContext('2d')); }

</script>

源程序 2：原始排队论算法

```
#include <cmath>
#include <iostream>
using namespace std;
const int maxn = 1007;
double jc(int k) { //阶乘
    double ans = 1;
    for (int i = 2; i <= k; i++) {
        ans *= i;
    }
    return ans;
}
int main() {
    double n = 1, lamda = 2.1, u = 2.5;
    double miu = lamda / u;
    double p0 = 1-miu;
    double p_sun = 0;
    double Q = 1 - p_sun;
    double A = lamda * Q;
    double L_dui = pow(miu, n + 1) *p0 / (n * jc(n)) ;
    L_dui /= (1 - miu / n) * (1 - miu / n);
    double W_dui = L_dui / lamda;
    double K = miu * (1 - p_sun);
    double L_xi = L_dui + miu;
    double W_xi = L_xi/lamda;
    cout << "-----输入各项参数：-----\n";
    cout << "服务员个数：" << n << endl;
    cout << "顾客流强度：" << lamda << endl;
    cout << "服务台能力：" << u << endl;
    cout << "-----系统效率指标：-----\n";
    cout << "损失概率= " << p_sun << endl;
    cout << "系统的相对通过能力= " << Q << endl;
    cout << "系统的绝对通过能力= " << A << endl;
    cout << "系统内排队顾客的平均数= " << L_dui << endl;
    cout << "顾客的平均排队时间= " << W_dui << endl;
    cout << "占用服务员的平均数= " << K << endl;
    cout << "系统内顾客的平均数= " << L_xi << endl;
    cout << "顾客在系统中平均逗留时间= " << W_xi << endl;
    return 0;
}
```

源程序 3：设置乘车点后的排队论算法

```
#include <cmath>
#include <iostream>
using namespace std;
const int maxn = 1007;
double jc(int k) { //阶乘
    double ans = 1;
    for (int i = 2; i <= k; i++) {
        ans *= i;
    }
    return ans;
}
int main() {
    double n = 2, lamda = 2.1, u = 2.5;
    double miu = lamda / u;
    double p0 = 1-miu;
    double p_sun = 0;
    double Q = 1 - p_sun;
    double A = lamda * Q;
    double L_dui = pow(miu, n + 1) *p0 / (n * jc(n)) ;
    L_dui /= (1 - miu / n) * (1 - miu / n);
    double W_dui = L_dui / lamda;
    double K = miu * (1 - p_sun);
```

```
    double L_xi = L_dui + miu;

    double W_xi = L_xi/lamda;

    cout << "-----输入各项参数：-----\n";

    cout << "服务员个数：" << n << endl;

    cout << "顾客流强度：" << lamda << endl;

    cout << "服务台能力：" << u << endl;

    cout << "-----系统效率指标：-----\n";

    cout << "损失概率= " << p_sun << endl;

    cout << "系统的相对通过能力= " << Q << endl;

    cout << "系统的绝对通过能力= " << A << endl;

    cout << "系统内排队顾客的平均数= " << L_dui << endl;

    cout << "顾客的平均排队时间= " << W_dui << endl;

    cout << "占用服务员的平均数= " << K << endl;

    cout << "系统内顾客的平均数= " << L_xi << endl;

    cout << "顾客在系统中平均逗留时间= " << W_xi << endl;

    return 0;

}
```

源程序 4：深圳市出租车分布散点图代码

```
from pyecharts import Geo, Style
import pandas as pd

# import pyecharts
# 导入 excel 表举例
df = pd.read_excel('深圳.xlsx')
df.head()
# 导入自定义的地点经纬度
geo_cities_coords     =     {df.iloc[i]['ID']:     [df.iloc[i]['lng'],
```

```
df.iloc[i]['lat']]
                    for i in range(len(df))}  # 根据文件大小生成字典
attr = list(df['ID'])  # 字典的每个键值
value = list(df['ID'] *2)  # 由于量值的太大，换算一下(散点的颜色就是和这个相关的)
style = Style(title_color="#fff", title_pos="center",
              width=1200, height=600, background_color="#404a59")
# 可视化
geo = Geo('深圳市机场附近出租车轨迹分布', **style.init_style)
geo.add("", attr, value, visual_range=[0, 100], symbol_size=5,
        visual_text_color="#fff", is_piecewise=True,type = 'heatmap',
        is_visualmap=True, maptype='深圳', visual_split_number=10,
        geo_cities_coords=geo_cities_coords)

geo.render('深圳出租车分布 1.html')
```