

基于计算机仿真的机场出租车问题的研究

摘 要

针对给出出租车司机选择策略的情况，本文选取了**乘客数量变化规律**、“**蓄车池**”里出租车数量、**上客区通行效率**、**机场与市区的距离**、**市区内出租车需求度**和**个人不确定因素**这六个因素作为影响出租车司机做决策的因素。然后，针对某个出租车个体，建立了**基于 0-1 规划的面向个体司机的决策模型**并利用**决策树**做出出租车司机的决策；针对出租车司机整体，建立了**基于 TOPSIS 理想解法的面向出租车司机群体的吸引度模型**。

针对以现实中某一机场为例的情况，本文以**杭州市萧山机场**为例，收集了 2018 年 6 月 6 日 7:00 到 23:00 的有关数据，利用搜集到的**机场乘客数量变化**、**蓄车池内出租车数量的变化**等数据，基于上一种情况下建立的**TOPSIS 理想解法模型**，我们得到了单天 7:00 到 23:00 机场对司机的**吸引度数值**。再利用**SPSS 软件**，在同一时间序列下，结合有关数据，进行**相关性分析**，发现其相关系数为**0.85**，论证了模型的合理性。然后，针对模型对各个因素的依赖性**问题**，我们做了**控制变量实验**，发现模型对乘客数量的依赖性较强。

针对设置上车点与合理安排乘客和出租车的情况，我们建立了**计算机仿真模型**，以**通行效率最大**为目标，**模拟仿真**解决了泊位区泊位数量，乘客上车时间点，乘客上车点的空间安排的问题。最终得出结论：泊位区应设置 2×3 的车位形式，并对每个泊位设置**一对一上车点**；在乘客所选出租车后面这辆车停稳后，便可放行乘客进入泊位区。

针对均衡出租车司机收益的情况，我们建立了以出租车司机时间效益值方差最小为目标函数的计算机仿真模型。针对优先权方案的给出，我们给出了两种方案：方案一是设置短途载客返回的**出租车优先队列**；方案二是**优先调度蓄车池中搭载短途乘客返回的出租车数量最多的列进入乘车区**。通过计算机仿真，我们定量讨论了两种方案的优劣性。

最后，我们对模型进行了检验。通过比对 VISSIM 平台下的数据，与我们模型得出数据的**差异程度**证明了我们模型的正确性。

关键词：0-1 规划 TOPSIS 理想解法 控制变量实验 计算机模拟仿真

1 问题重述

1.1 问题背景

航空运输业是经济社会发展的晴雨表，我国已成为世界上民航基础设施建设发展最快的国家。2018 年末，我国出现了中国第一个、世界第二个旅客吞吐量破亿的机场——北京首都国际机场。其他机场吞吐量也都呈上升趋势，如国内旅客吞吐量以增速 9.2% 排名第二的杭州萧山机场等。

下飞机的一部分旅客会选择乘坐出租车到达目的地，送客到机场的出租车司机也面临着到“蓄车池”排队等候刚下飞机的旅客和空载返回市区拉客两个选择。随着机场旅客数量的不断增加，为提高收益，司机的决策显得尤为重要。同时，机场管理部门为保证乘车效率等出台的管理方案（如合理设置“乘车点”、“优先权”安排等）也起着很大的作用。

1.2 问题重述

结合实际情况，建立数学模型，解决以下问题：

- (1) 找出影响出租车司机决策的因素，并研究其影响机理。结合机场车客数量规律，为使司机收益尽可能高，建立司机选择时的决策模型，并给出选择策略。
- (2) 收集国内某一机场客流量变化和航班数量等机场相关数据以及机场所在城市的出租车客流量变化、出租车来往机场的收费标准等其他相关数据，利用上一步中建立的模型，给出机场出租车司机合理的选择方案。同时，说明模型的合理性及对相关因素的依赖性。
- (3) 为缓解出租车和乘客排队的情况，帮助机场管理部门设置位置得当的“上车点”，并合理安排出租车和乘客，在保证出租车和乘客安全的情况下，使得乘车效率最高。
- (4) 机场出租车司机的收入与行驶里程有关，司机不能只载目的地较远乘客。为保证出租车司机收入均衡，机场管理部门要给予短途载客再次返回的出租车一定“优先权”，给出可行的“优先”安排方案。

2 问题分析

2.1 问题一分析

出租车司机在送完乘客后，有两种选择，如图 1 所示。要给出司机的选择策略，首先，我们要找出影响出租车司机做决策的因素，并研究这些因素影响司机决策的机理。然后我们选取了**乘客数量变化规律**、“蓄车池”里出租车数量、**上客区通行效率**、**机场与市区的距离**、**市区内出租车需求度**和**个人不确定因素**这六个作为影响因素，并分析它们与司机做决策去机场蓄车池的正负相关性。最后，在考虑机场乘客数量和保证出租车司机收益尽可能高的情况下，针对出租车司机个体和出租车司机整体分别建立合理的数学模型，再做出决策选择。

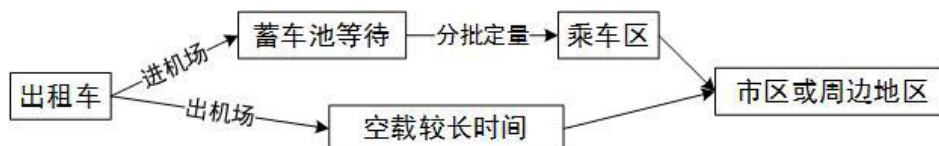


图1 出租车司机决策选择

针对某个出租车司机个体来说，在送完到机场的乘客之后，他（她）有两种选择，“去”和“不去”，即去蓄车池排队等候载客和不去蓄车池直接空载返回市区载客。基于司机有这两种选择，建立基于 0-1 规划的数学模型，再利用决策树给出司机的决策策略。

针对出租车司机整体来说，我们建立基于 TOPSIS 理想解法的模型来研究去机场蓄车池排队等待载客这个选择对出租车司机的吸引度。根据这个机场蓄车池对司机吸引度模型得到司机去蓄车池排队等待的意愿度。得到的数值越高，表明司机去蓄车池的意愿越强烈。意愿越强烈，在出租车司机的这个整体当中，在同一时刻，进入蓄车池的司机数量越多。

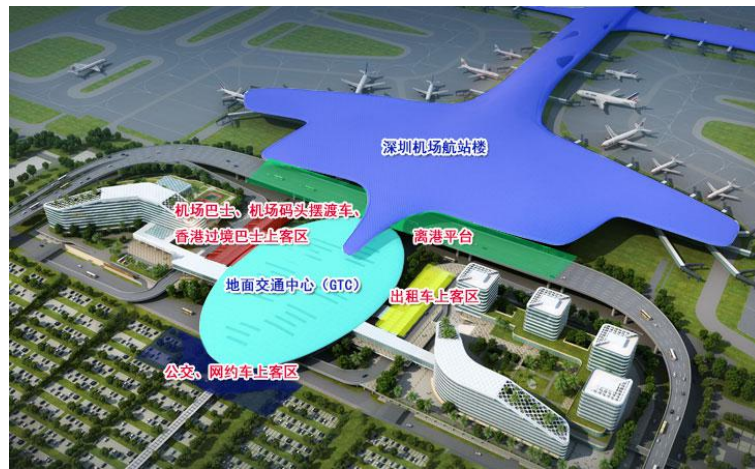


图2 深圳市宝安国际机场平面图

2.2 问题二分析

收集国内某一机场及其所在城市出租车相关数据，我们选择了杭州萧山国际机场作为研究对象。通过阅读文献和网络查询，我们了解到萧山机场与杭州市市中心距离，萧山机场出租车客流量数据，杭州市区对出租车数量需求等相关数据。然后，利用问题一中建立的基于 TOPSIS 理想解法的司机吸引度模型，输入萧山机场的相关数据，得到出租车司机在送客到机场之后直接去蓄车池排队等待载客的意愿度数值。再利用 SPSS 软件，在同一时间序列下，将意愿度和萧山机场进入蓄车池内出租车数量进行线性回归，得到相关性系数分析相关性。之后，我们为了分析模型对各个因素的依赖性，进行了**控制变量实验**，单独分析每个因素对司机吸引度的影响，从而得出结论。

2.3 问题三分析

对于乘客“上车点”的设置和出租车与乘客的合理安排，首先画出双车道的泊车区示意图，我们采用计算机模拟的方式，认为乘客数量足够多，计算出双车道在各种泊位区设置情况下的通行能力。同时，我们还研究了泊位区在空间上的布局，以及时间上乘客进入泊位区的时间点。通过几种方式对比，我们的了一个较优的泊位区上车点的设置和乘客与出租车合理安排的方案。

2.4 问题四分析

针对优先权的方案，为了使出租车司机收益均衡，我们设置了两种方案。第一种是单独开辟一条特殊的出租车排队通道，当搭载短途旅客的出租车司机返回到蓄车池时，无需在队伍后面排队进入泊位区。他们可以进入特殊通道，排队进入泊位区。第二种方案是把蓄车池看成一个矩阵，当有搭载短途旅客的出租车司机再次返回到蓄车池中时，优先调度刚返回来的出租车数量多的那一列进入泊位区。

3 模型假设

3.1 假设内容

- (1) 假设“蓄车池”和等待区没有交通事故发生。
- (2) 假设没有恶劣天气现象，也没有出现航班大量延误集中到达的情况。
- (3) 假设机场管理人员能够准确识别搭载短途载客返回的出租车。
- (4) 假设出租车司机和乘客都听从现场管理人员的指挥和管理。
- (5) 假设送客到达机场的出租车司机之后的收益等价为其送客之后到载下一客人的时间。

3.2 假设可行性

- (1) 我们排除了一些突发的事故，不考虑极端的偶然现象。因为当这种现象出现的概率非常小，所以我们假设所研究的区域没有交通事故发生。
- (2) 因为出租车司机在载客后乘客的目的地本来就具有很大的随机性，而决策时往往是根据经验定性分析各个决策因素的影响，难以定量分析，所以我们可以认为到达机场后出租车的收益可以等价为其送客之后到载下一客人的时间。

4 符号说明

表 1 主要符号说明

符号	描述	单位
N	某一时刻上客区等待的乘客数量	人
C	某一时刻蓄车池内出租车数量	辆
v	上客区车辆通行能力	cpu/h
t_0	当前时间	-
t_1	满足 $\int_{t_0}^{t_1} h(t) dt = N - C$	-
t_2	出租车从机场到市区所需时间	h
L	机场与市区距离	km
V	出租车在机场返回市区途中的速度	km/h
t_3	出租车到达市区后接单时间	h

k_1	比例因子	-
N_1	出租车到市区后接到的订单数目	个
E_1	出租车司机去机场蓄车池直至载客的空载时间	h
E_2	出租车司机回市区直至载客的空载时间	h
Q	在一定时间内通过泊车区的出租车数量	辆
T	总时间	h
t'	乘客到达相应车位的时间	h
t''	放置行李时间	h

5 模型的建立与求解

5.1 问题一的模型

5.1.1 模型准备

(1) 乘客数量变化规律

根据参考文献三可知，分析机场出租车客流量监测区的数据可以发现，随着时间的变化，每年乘客数量都在增加，但是乘客数量增长的趋势都是相似的。再通过分析可知，如果以一年为周期，不同年的对应月份及相同的节假日具有相似的客流量变化趋势；如果以一周为周期，不同星期的同一个星期几具有相似的客流量变化趋势；如果以一天为周期，每天的同一个时间段具有相似的出租车客流量变化趋势。

1) 有关理论的建立

查阅参考文献三可知，任何时间序列模型都可以回归到以下公式：

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t \quad (1)$$

其中， $g(t)$ 表示趋势的变化； $s(t)$ 表示时间序列中的周期性部分； $h(t)$ 表示节日的影响； ε_t 表示一些突发情况。

对于趋势项，随着国民收入的增长，越来越多人选择乘坐飞机出行，中国机场客流量每年都在呈上升趋势。再根据中国民航发布的 2018 年客座率指数月度走势数据，航空客流量呈现淡季和旺季之分，即周期项。



图 3 2018 年客座率指数月度走势图

2) 乘客数量受时间变化的具体分析

① 随年度和季节变化

根据中国民航发布的数据，如图四，我们发现虽然中国旅客运输量整体随年份呈上升趋势，但是在一年内其整体的变化趋势是一致的，比如，基本在每年8月份左右达到这一年的峰值，1月份左右是一年的最低值。

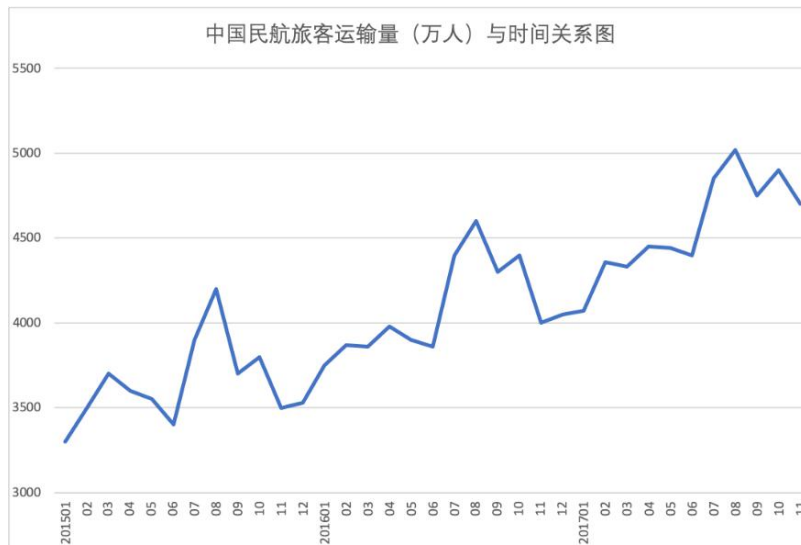


图4 中国民航旅客运输量与时间关系图

② 随每天时间的分布

根据文献5的数据，我们绘制出客流量随时间变化的趋势图，如图4所示。

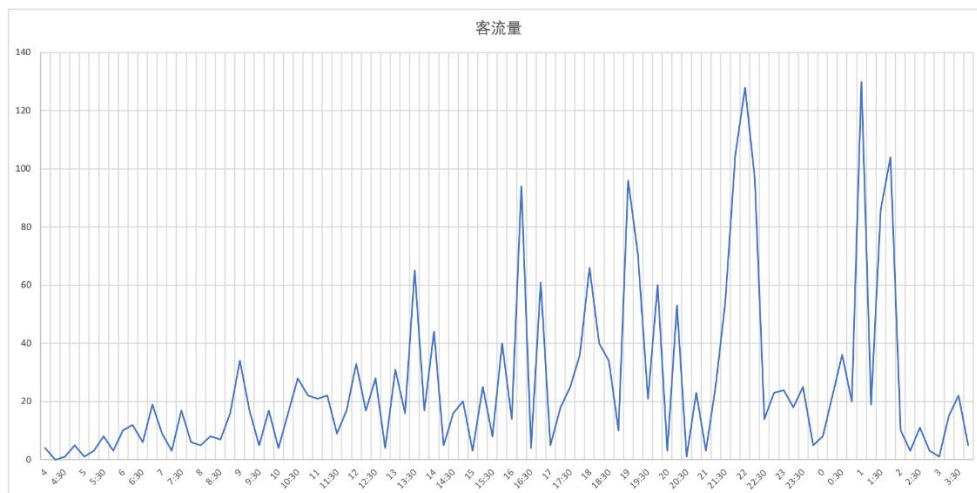


图5 客流量随时间变化图

3) 对司机决策的影响

由上分析我们发现，乘客数量呈现一定的规律性，司机从而可以根据其

经验大致把握乘客把数量变化规律，从而为自己决策提供一定的依据，一般地，我们认为，当其他影响因素一定时，乘客数量越多，司机决策越倾向于前往蓄车池排队载客；乘客数量越少，司机决策越倾向于离开机场前往市区。

(2) 蓄车池车辆数目 C

蓄车池中车辆数目是司机能够观测到的信息，认为 C 越大，司机越不愿意前往蓄车池排队等待，因为可能要付出很大的排队时间成本；反之，则司机更愿意前往蓄车池排队。

(3) 上客区通行效率 v

上客区通行效率认为是上客区实际通行能力，表示一定时间 T 内通过固定截面的车流量 Q 。

$$v = \frac{Q}{T}$$

显然，在乘客充足的情况下，上客区通行效率 v 是限制司机排队时间的主要成本。

(4) 机场距市区距离 L

若司机选择空载返回市区，那么机场距市区距离 L 是一个很重要的因素， L 越大，那么司机选择空载返回市区的成本也越大，则司机会更不情愿选择空载返回市区

(5) 市区出租车需求度 δ

若司机选择返回市区，市区对出租车需求度强烈，会大概率接到更多的乘客，获取更多的利润，那么司机会更强烈的选择返回市区。而需求度与时间有关，如果在早晚高峰即上下班时间，那么出租车的订单数会激增。

(6) 个人不确定因素 ϵ

还应当考虑生活中的一些不确定的因素，比如司机个人的原因，主观上的一些违背常理的想法。

5.1.2 模型建立

(1) 模型 I——基于 0-1 规划的面向个体司机的决策模型

决策变量为：

$$x = \begin{cases} 0, & \text{司机返回市区} \\ 1, & \text{司机去蓄车池} \end{cases} \quad \# (2)$$

目标函数为：使空载时间最短。

$$\max E = x (E_1 + \alpha) + (1 - x)E_2 \quad \# (3)$$

式中， E_1 表示去机场直到载到下一个乘客空载时间； E_2 为离开机场去市区直到载到下一个乘客空载时间。 α 为价格因子，由于考虑到在机场载的客人其收益会比在市区载的客人往往会更大，为了体现出这种优势，故加入了价格因子 α

约束条件为：

1) 供求规则

$$\sigma = \frac{N}{C} \quad \# (4)$$

$$\sigma \begin{cases} > 1, \text{ 供小于求} \\ = 1, \text{ 供等于求} \\ < 1, \text{ 供大于求} \end{cases} \quad \# (5)$$

其中, N 代表此刻上客区等待的乘客数量, C 表示蓄车池内出租车的数量。在一定条件下, 乘客数量 N 与出租车数量 C 满足供求关系。

2) 上客区等待时间

$$E_1 = \begin{cases} \frac{N}{v}, & \sigma \geq 1 \\ t_1 - t_0, & \sigma < 1 \end{cases} \quad \# (6)$$

式中, v 表示上客区通行能力; t_0 代表着当前时间; t_1 满足 $\int_{t_0}^{t_1} h(t) dt = N - C$ 。

我们认为 $\sigma \geq 1$ 时, 即此时出租车供给小于或等于乘客需求, 那么其排队时间主要受限于上客区通行效率。

而当 $\sigma < 1$ 时, 即此时出租车供给大于乘客需求, 那么其排队时间主要受限于乘客数量的变化规律, 即其主要排队时间是在等乘客的到来。

3) 机场前往市区时间

$$t_2 = \frac{L}{V} \quad \# (7)$$

其中, L 表示机场距市区距离; V 是出租车行驶速度。

4) 市区接单时间

$$t_3 = k_1 N_1 \quad \# (8)$$

式中, k_1 为比例因子; N_1 代表着出租车到市区后的订单数目。

在这里我们假设出租车司机到达市区后其在市区的接单时间与市区此时的订单数量成正比。

5) 前往市区耗费时间

$$E_2 = t_2 + t_3 \quad \# (9)$$

综上, 面对个体司机的决策模型为:

$$\begin{cases} \max E = x (E_1 + \alpha) + (1 - x) E_2 \\ \text{s.t.} \left\{ \begin{array}{l} \sigma = \frac{N}{C} \\ E_1 = \begin{cases} \frac{N}{v}, & \sigma \geq 1 \\ t_1 - t_0, & \sigma < 1 \end{cases} \\ t_2 = \frac{L}{v} \\ t_3 = k_1 N_1 \\ E_2 = t_2 + t_3 \\ x \in \{0, 1\}, \alpha \text{ 由司机个人经验决定} \end{array} \right. \quad \# (10) \end{cases}$$

(2) 模型 I 的求解——基于决策树

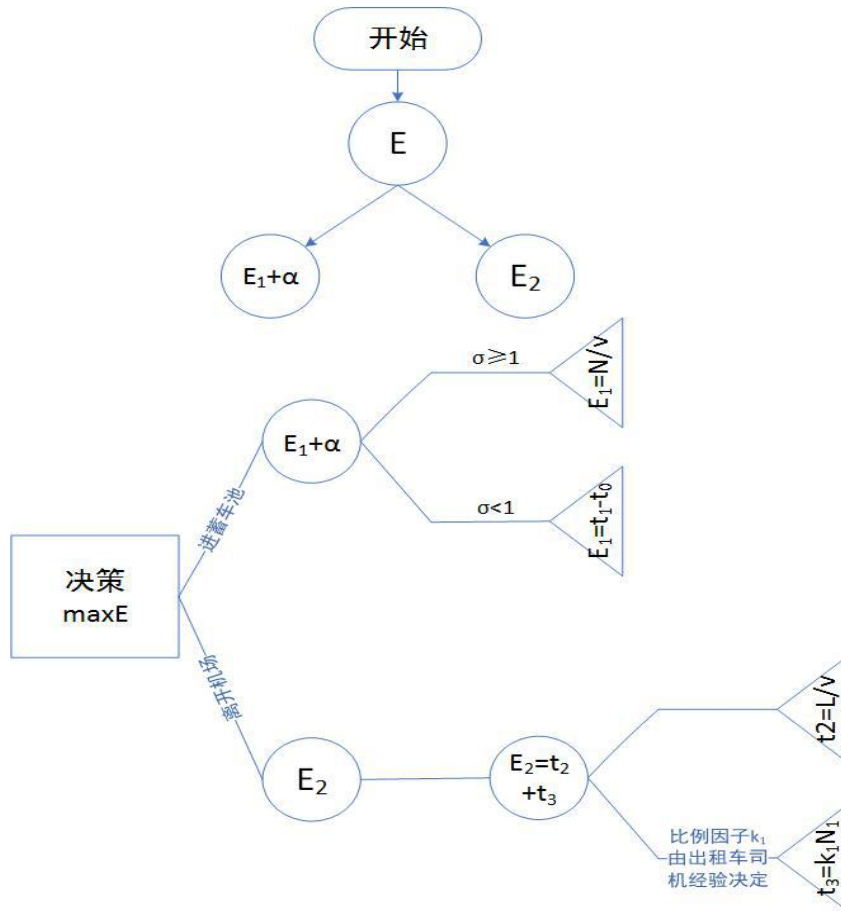


图 6 模型 I 的决策树

(3) 模型 II——基于 TOPSIS 理想解法的面向出租车司机群体的吸引力模型

我们选择了乘客数量变化规律、“蓄车池”里出租车数量、上客区通行效率、机场与市区的距离、市区内出租车需求度和个人不确定因素这六个指标。

- 1) 用向量规划化的方法求得规范决策矩阵。取决策问题的决策矩阵 $\mathbf{A} = (a_{ij})_{m \times n}$ ，规范化决策矩阵 $\mathbf{B} = (b_{ij})_{m \times n}$ ，其中，

$$b_{ij} = \frac{a_{ij}}{\sqrt{\sum_{i=1}^m a_{ij}^2}}, \quad i=1, 2, \dots, m; j=1, 2, \dots, n. \quad \# (11)$$

- 2) 构造加权规范矩阵 $\mathbf{C} = (c_{ij})_{m \times n}$ 。给定各属性的权重向量为 $\mathbf{w} = [5, 3, 2, 3, 3]$ ，则

$$c_{ij} = w_j \cdot b_{ij}, \quad i=1, 2, \dots, m; j=1, 2, \dots, n. \quad \# (12)$$

其中我们特别看重乘客的数量，所以给它赋一个最高权重 5。相对而言，对于其他几个因素，我们认为其在实际过程中的影响没有这么大。

- 3) 确定正理想解 \mathbf{C}^* 和负理想解 \mathbf{C}^0 。

正理想解为：[500, 0, 100km, 0, 1000pcu/h]

负理想解为：[0, 500, 0km, 1, 0]

在正理想解中，上客区乘客数量为 500，蓄车池内没有出租车排队等待，机场到市区的距离为 100km，市区内出租车需求度为 0 且上客区服务效率是 1000pcu/h。在负理想解中，上客区没有乘客等待乘坐出租车，蓄车池内有 500 辆出租车在排队等候下班航班要乘坐出租车的乘客，从机场到市区的距离是 0km，市区内出租车需求度为 1，上客区服务效率为 0 pcu/h。

4) 计算各方案到正理想解与负理想解的距离。备选方案 d_i 到正理想解的距离为

$$s_i^* = \sqrt{\sum_{j=1}^n (c_{ij} - c_j^*)^2}, \quad i = 1, 2, \dots, m; \quad \# (13)$$

备选方案 d_i 到负理想解的距离为

$$s_i^0 = \sqrt{\sum_{j=1}^n (c_{ij} - c_j^0)^2}, \quad i = 1, 2, \dots, m. \quad \# (14)$$

5) 计算各方案的排序指标值（即综合评价指数），即

$$f_i^* = s_i^0 / (s_i^0 + s_i^*), \quad i = 1, 2, \dots, m. \quad \# (15)$$

5.2 问题二的模型

5.2.1 机场出租车司机选择方案——以杭州萧山国际机场为例

1) 数据准备

通过查阅网上资料，我们得知萧山机场到杭州市市中心的距离 $L = 45.4km$ ；根据《道路工程技术标准》得知，出租车司机在从机场到市中心的这条路上行驶时的平均速度 $v = 45km/h$ ，道路通行效率为 400 辆/时。

① 客流量

分析萧山机场 2018 年 6 月 6 日从 7:00 到 23:00 的出租车客流量的数据，得到矩阵 P_1 。

$$P_1 = [125, 139, 159, 264, 288, 299, 312, 241, 203, 158, 234, 234, 210, 193, 198, 126, 98]$$

② 蓄车池车辆数目

由于网络上相关开源数据极度匮乏，我们电话咨询机场有关工作人员得到萧山机场从 7:00 到 23:00 的蓄车池车辆数目数据。整理数据得到矩阵 P_2 。

$$P_2 = [54, 95, 148, 364, 302, 310, 265, 324, 261, 235, 224, 198, 234, 254, 254, 195, 88]$$

③ 杭州市区出租车需求度

分析图 7 数据，得到矩阵 P_3 。

$$P_3 = [0.4, 0.5, 0.6, 0.7, 0.7, 0.7, 0.8, 0.7, 0.6, 0.6, 0.7, 0.8, 0.8, 0.9, 0.8, 0.7, 0.6]$$

表 2 相关数据表

时间	距离 (km)	平均速度 (km/h)	出租车 客流量	市区出租车 需求度	蓄车池 出租车数量
----	------------	----------------	------------	--------------	--------------

7:00	45.4	45	125	0.4	54
8:00	45.4	45	139	0.5	95
9:00	45.4	45	159	0.6	148
10:00	45.4	45	264	0.7	364
11:00	45.4	45	288	0.7	302
12:00	45.4	45	299	0.7	310
13:00	45.4	45	312	0.8	265
14:00	45.4	45	241	0.7	364
15:00	45.4	45	203	0.6	261
16:00	45.4	45	158	0.6	235
17:00	45.4	45	234	0.7	224
18:00	45.4	45	234	0.8	198
19:00	45.4	45	210	0.8	234
20:00	45.4	45	193	0.9	254
21:00	45.4	45	198	0.8	254
22:00	45.4	45	126	0.7	195
23:00	45.4	45	98	0.6	88

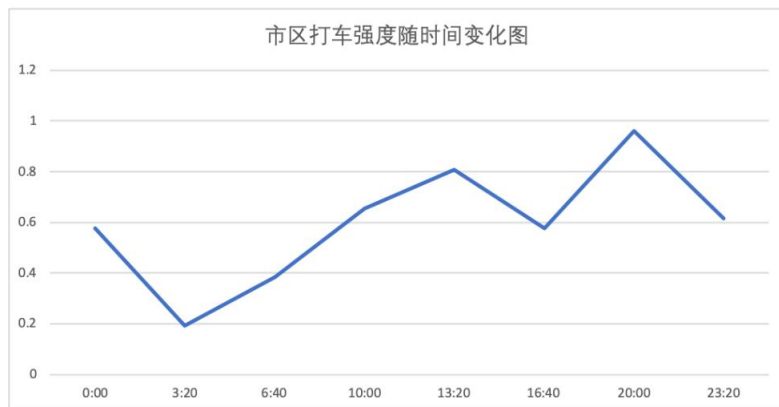


图 7 杭州市市区出租车需求度随时间变化图

2) 模型求解

我们运用 TOPSIS 理想解法求得 7:00 到 23:00 萧山机场对出租车乘客的吸引力如表 3 所示。

表 3 乘客吸引力表

时间	7:00	8:00	9:00	10:00	11:00	12:00	13:00	14:00	15:00
----	------	------	------	-------	-------	-------	-------	-------	-------

吸引力	0.510	0.502	0.493	0.505	0.597	0.602	0.665	0.497	0.473
时间	16:00	17:00	18:00	19:00	20:00	21:00	22:00	23:00	
吸引力	0.386	0.592	0.625	0.522	0.455	0.468	0.364	0.403	

5.2.2 模型合理性分析

我们通过调查得到了 7:00 到 23:00 萧山机场新增的进入蓄车池出租车数量，将此序列与上面得到的各个时刻机场对司机的吸引力序列，利用 SPSS 软件进行相关性分析，结果如下图。

相关性

		司机吸引力	新增出租车数量
司机吸引力	皮尔逊相关性	1	.850**
	Sig. (双尾)		.000
	个案数	17	17
新增出租车数量	皮尔逊相关性	.850**	1
	Sig. (双尾)	.000	
	个案数	17	17

** . 在 0.01 级别 (双尾) , 相关性显著。

图 8 相关性图

我们通过 SPSS 软件发现两列数据的相关性为 0.850，可以认为相关性非常高，即可以在 0.01 的错误概率下，两列数据十分相关。这说明我们的模型与实际结合很紧密，论证了模型的合理性。

5.2.3 相关因素依赖性——控制变量实验

我们通过改变其中一个因素的值，观察吸引力的改变，从而分析模型对此因素的依赖性。我们以 15:00 的数据为例：[203, 261, 400, 45.4, 0.6]

(1) 乘客数量

我们发现吸引力随着乘客数量的增加而增加，但趋势逐渐变慢，知道趋于一个极限值，接近于 1。这意味着，乘客数量对吸引力影响很大，模型依赖性强。

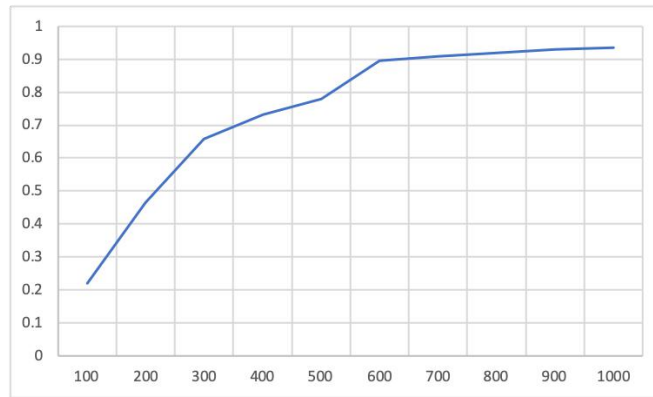


图 9 吸引力随乘客数量变化图

(2) 蓄车池中车辆数目

蓄车池中车辆数目增加，吸引力却逐渐下降，最后下降到一个趋于 0.15 的极限值，模型依赖性不如乘客数目强。

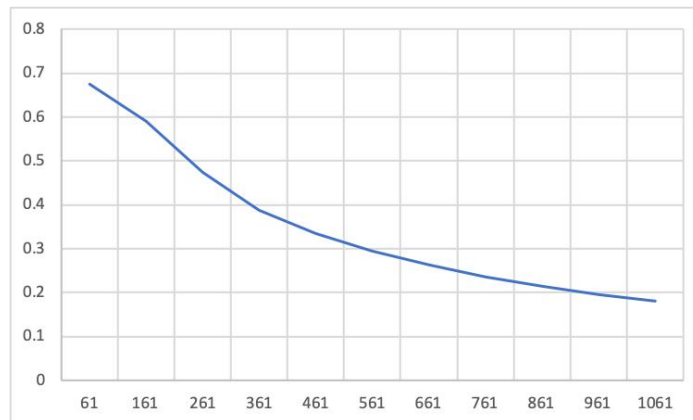


图 10 吸引力随蓄车池中车辆数目变化图

(3) 上客区通行效率

随着上客区通行效率的增加，吸引力也逐渐增加，图中来看趋于 0.52。这是因为上客区通行效率对吸引力的影响还要考虑乘客数量的因素，所以显得依赖性不强。

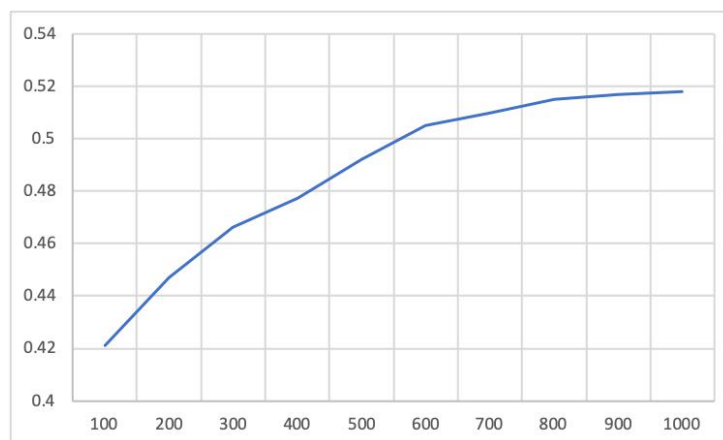


图 11 吸引力随上客区通行效率变化图

(4) 其他因素

而对于其他因素，如机场到市区的距离，由于其会随着实际情况的改变而发生大大的改变，我们认为模型对齐依赖性不强。

5.3 问题三的模型

5.3.1 关于泊车数量的确定

(1) 模型准备

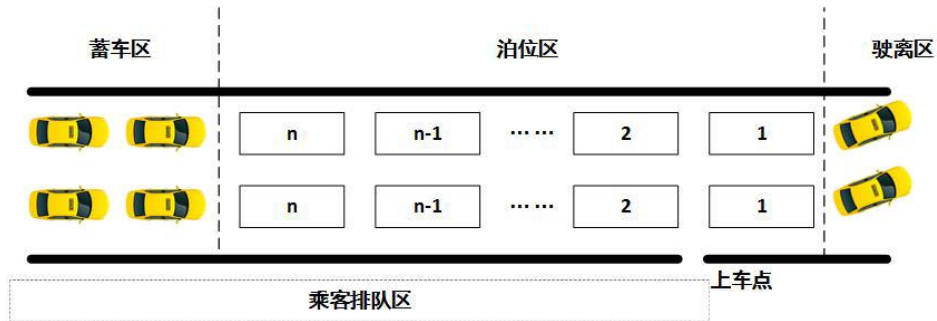


图 12 泊位区示意图

建立泊位区矩阵 $P[2][k]$, 其中, $P[1][1]$ 、 $P[2][1]$ 代表泊位区中两条车道中位于距离驶离区最近的两个车位, 如图 12 中的车位 1 所示。

(2) 模型建立

决策变量为:

$$k = 1, 2, \dots, 10. \quad \#(16)$$

表示泊位区有 $2 \times k$ 个车位数。

目标函数为:

$$\max C^{(k)} = \frac{Q}{T} \quad \#(17)$$

式 (17) 中, $C^{(k)}$ 表示 $2k$ 个车位数时的通行能力; Q 为在一定时间内通过的出租车的数量; T 代表着一定时间。



图 13 蓄车池与上客区平面图

图 13 中, 黄色框划出的区域为蓄车池所在区域, 红色区域为上客区。在上客区一边有乘客排队等待出租车的乘客等待区。上客区中间为泊车位。泊车位中有一定数量的车位供出租车驶入上客区后停车, 等待乘客上车。

1) 人车安全规则

只有当泊车位车辆都驶离时，才允许下一轮出租车从蓄车池驶入泊车区，即此时，

$$p[1][k] + p[2][k] = 0 \# (18)$$

直到本轮次所有出租车在泊位区都停稳后，才允许乘客开始上车。

2) 车辆驶入规则

①蓄车池内的车辆分批次驶入泊车区。

③ 每批次进入泊位区两辆车，每个车道一个。

④ 当前面的两辆车驶离后，下一批次的两辆车才能驶入泊车区。

⑤ 对于最先驶入泊位区的两辆车，我们参考文献三中的数据，认为其驶入时间满足均值为 5s，偏差为 2s 的均匀分布；之后驶入的两辆车的驶入时间满足均值为 2.5s，偏差为 1s 的均匀分布。

⑥ 直到出租车占满泊位区 $2 \times k$ 个车位为止。

3) 乘客上车规则

可以认为当整个泊位区的车辆都停好之后，乘客才能开始上车，整个过程由乘客到达第 x 个车位的时间 t' 和到达车位后放置行李的时间 t'' 组成。

$$t' \sim U(x + 0.5, x + 1.5) \# (19)$$

$$t'' \sim \exp (30) \# (20)$$

式 (19) 表示乘客到达第 x 个车位的时间服从均值为 $x + 1$ 秒，偏差为 0.5 秒的均匀分布。

式 (20) 表示乘客到达相应车位后放置行李的时间满足均值为 30 的指数分布。

4) 出租车驶离规则

对于每一辆处于泊位区的出租车，必须当处于前面的出租车驶离时且乘客上车并安置好行李后，才可离开。认为这一轮次的出租车离开泊位区的时间满足均值为 k 秒，偏差为 1 秒的均匀分布。

(3) 模型求解

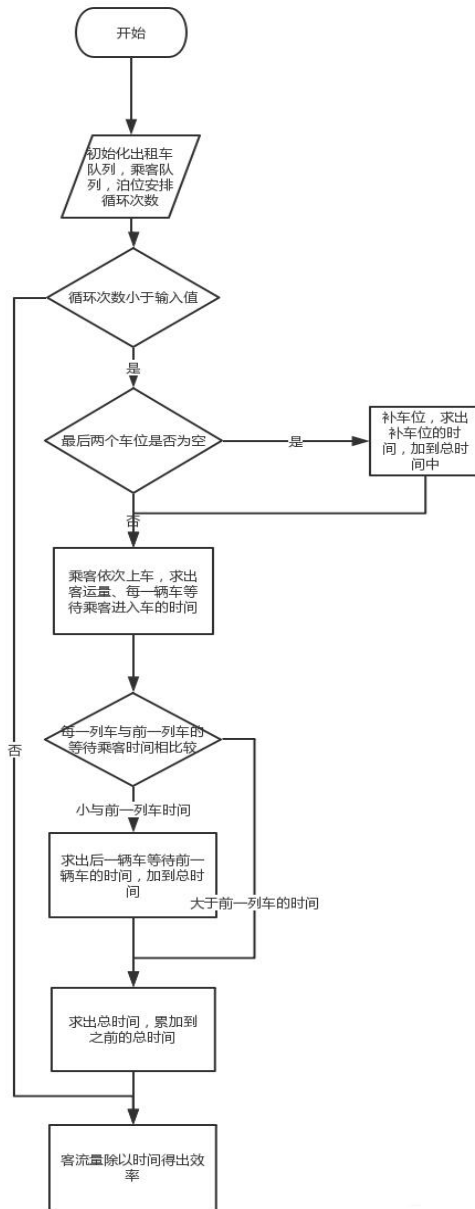


图 12 仿真流程图

(4) 结果分析

由图 13 可知，双车位车道载客效率随泊位增长呈增长趋势。

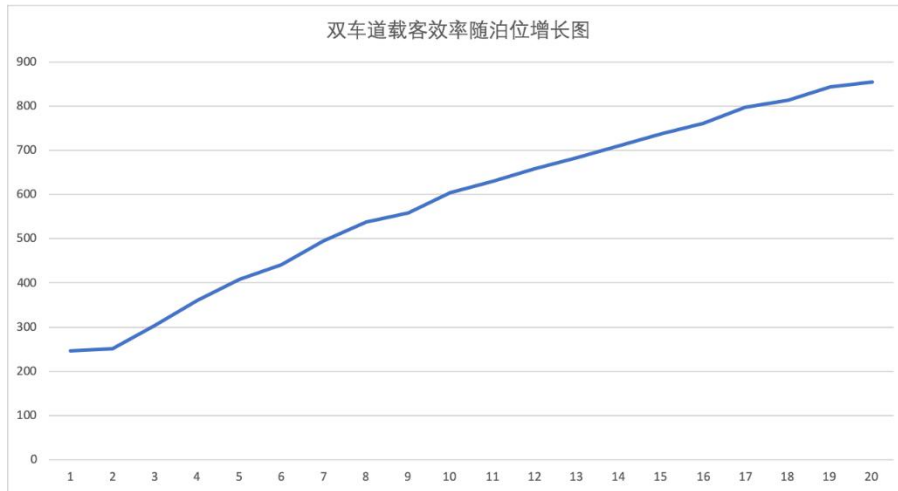


图 13 双车道载客效率随时间增长图

由图 14 可知，双车道载客效率随泊位数量增长而下降。

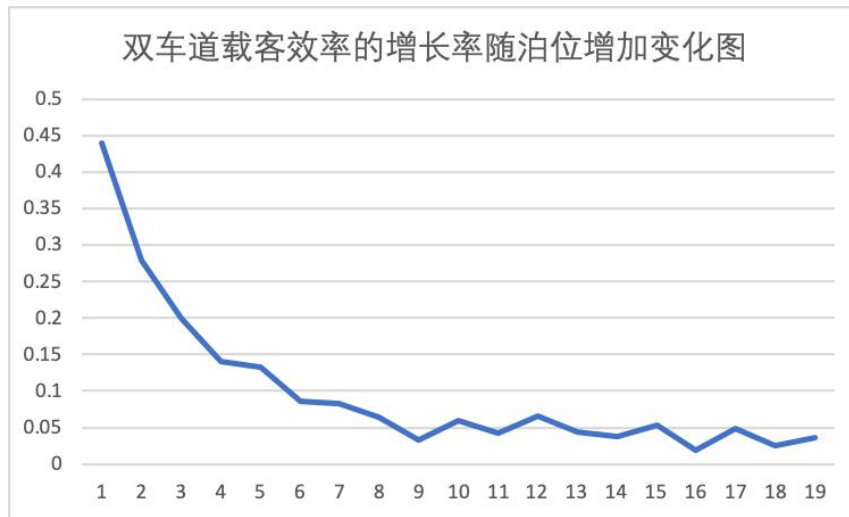


图 14 双车道载客效率增长率随泊位增加变化图

由图 15 可知，泊位区中单车位载客效率随泊位数量增长而下降。

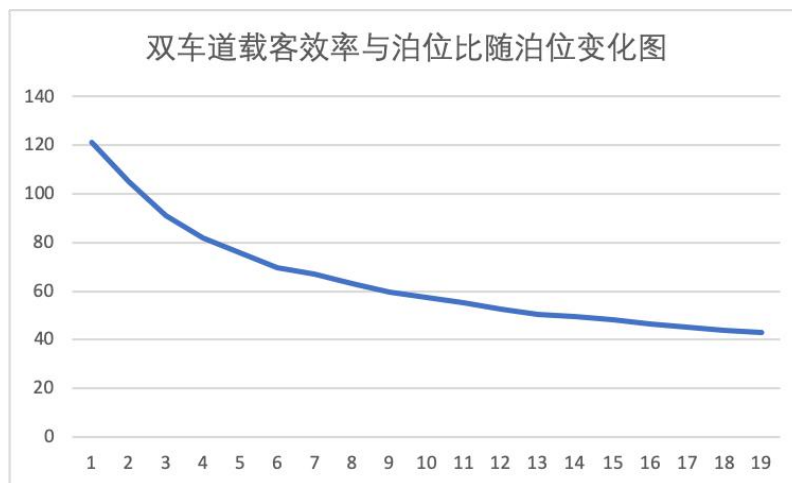


图 15 双车道载客效率与泊位比随泊位变化图

由以上分析可得，随着泊位数量的增加，乘客区整体载客效率在上升，但增长效率在下降。所以我们建议，从提高泊位区整体载客效率和单车位载客效率的目标下，在双车道前提下，泊位区设置为 2×3 个泊位。

5.3.2 乘客放行进入泊位区时间点的安排

规则修改：

我们修改乘客进入泊位区的时间点，由之前的当泊位区所有车辆都停稳后乘客才可进入改为对于搭载第 x 个出租车的乘客，其只需当第 x 个车位和第 $x + 1$ 个车位的出租车停稳，便可进入泊位区。我们针对 2×3 的泊位区设置进行 Matlab 模拟仿真，得到如图 16 所示结果。

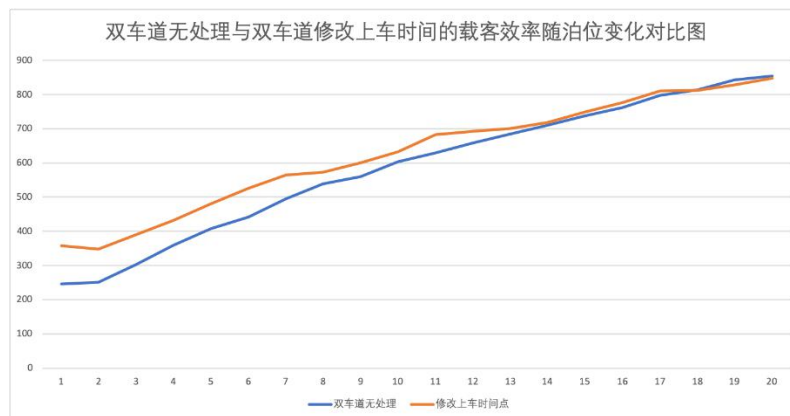


图 16 改变上车时间节点前后通行效率对比图

5.3.3 乘客进入泊位区空间点

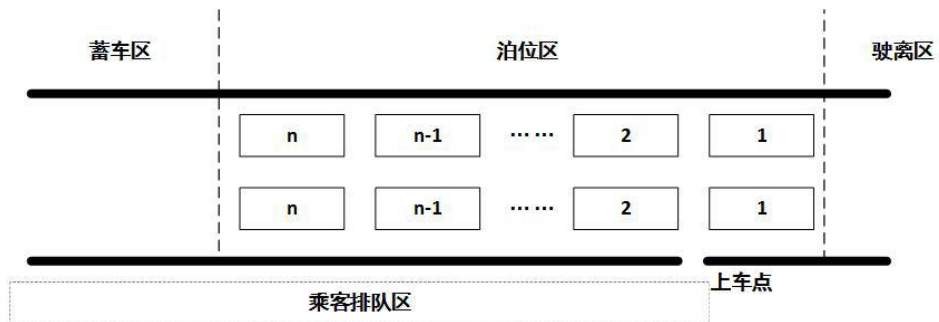


图 17 单上车点示意图

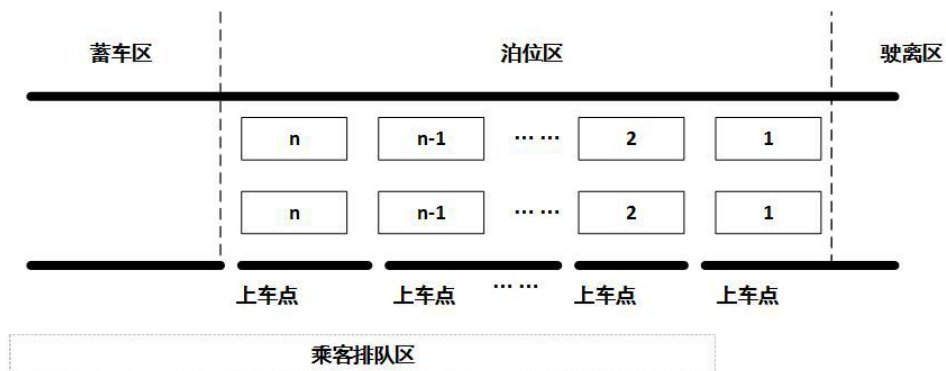


图 18 一对一上车点示意图

图 16 为泊位区空间上只有一个上车点的情形，图 17 为空间上每一对车位

都有对应上车点的情形。

规则修改：

由原来的乘客到达第 x 个车位的时间服从均值为 $x + 1$ 秒，偏差为 0.5 秒的均匀分布，修改为均值为 1 秒，偏差为 0.5 秒的均匀分布。

用 Matlab 仿真得到如下图所示的结果。

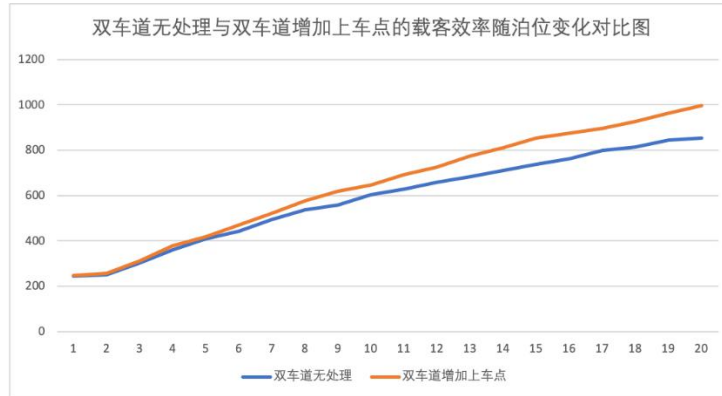


图 19 单上车点与一对一上车点通行效率比较图

5.3.4 最终方案

由以上分析和结果，对于“上车点”的设置我们推荐，采用 2×3 的泊位区设置。空间上我们采用一个车位对应一个上车点；时间上我们采用对于搭载第 x 个出租车的乘客，其只需当第 x 个车位和第 $x + 1$ 个车位的出租车停稳，便可进入泊位区。

5.4 问题四的模型

5.4.1 模型准备

(1) 时间效益值

我们定义时间效益值为出租车司机在机场上客系统中的收益 $m_{\text{总}}$ 与在从进入此系统到出系统所用时间 $t_{\text{总}}$ 的比值。

$$s = \frac{m_{\text{总}}}{t_{\text{总}}} \# (21)$$

(2) 出租车分类

根据出租车第一次所载乘客到达目的地与机场距离的远近程度，我们把出租车分为两类，即接到远程订单不返回的出租车和接到乘客目的地较近的返回机场的出租车。

1) 接到远程订单而不返回：

$$s = \frac{m_1}{t} \# (22)$$

m_1 是远程订单效益值； t 出租车司机从进入蓄车池到离开机场所经历的时间。

2) 接到短程订单而返回：

$$s = \frac{m_1 + m_2}{t_1 + t_2 + t} \# (23)$$

m_1 是短程订单效益值； m_2 是第二次返回接到的远程订单效益值；

t_1 是出租车司机第一次进入蓄车池到离开机场所经历的时间； t_2 是第二次进入蓄车池到离开机场所经历的时间； t' 是第一次离开机场到第二次进入蓄车池所经历的时间。

(3) 乘客的目的地

对于第一次排队进入乘客区搭载乘客的出租车，认为其有 $\sigma = 0.2$ 的概率接到短途订单；对于第二次排队进入乘客区搭载乘客的出租车，认为其接到乘客的目的地为远程；无论出租车第几次来到泊位区载客，只要接到远程订单就不再返回载客。

(4) 仿真系统安排

- 1) 蓄车池是一个 12×10 的矩阵，每个矩阵单位内只放一辆车。
- 2) 泊位区是 2×3 的矩阵，每个矩阵单位内只放一辆车。
- 3) 设置三个上车点，每个上车点对应两个车位。
- 4) 当泊位区所有车辆停稳后，才允许乘客进入，如图 所示。
- 5) 其余仿真规则同问题三。

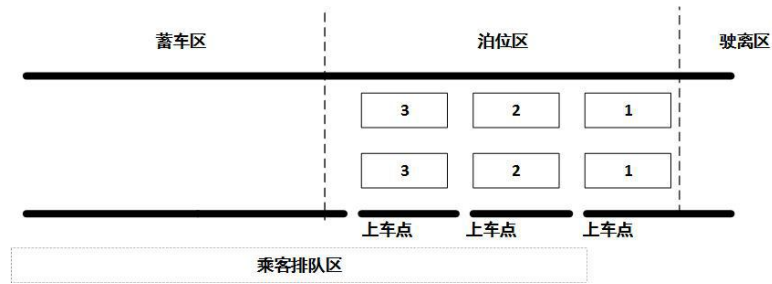


图 20 问题四泊位区设置图

5.4.2 模型建立

目标函数为：

$$A = \sqrt{\frac{\sum_{i=1}^n (s_i - \bar{s})^2}{n}} \# (24)$$

n 是仿真系统出租车总数； \bar{s} 表示仿真系统内所有出租车时间效益值的平均值。

5.4.3 优先权方案

方案一： 设置短途载客返回的出租车优先队列

对于短途载客返回的出租车，不必进入蓄车池，直接进入另外一个特殊队列。每次调度出租车进入乘车区时，优先调度此特殊队列中的出租车。经 Matlab 仿真并与无优先权的情况进行比较，得到下图。具体代码见附录。

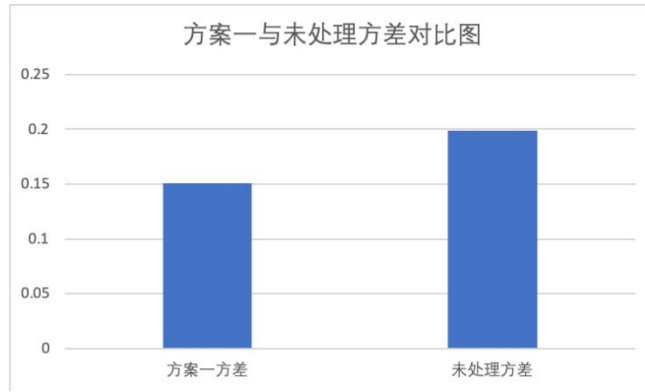


图 21 方案一与未处理方案方差对比图

方案二： 优先调度蓄车池中搭载短途乘客返回的出租车数量最多的列进入乘车区

对于蓄车池中出租车进入乘车区的调度方式，我们采取列调度的方式。我们每次从蓄车池中搭载短途乘客返回的出租车数量最多的列中从前往后调度相应数量的出租车进入乘车区。经 Matlab 仿真并与无优先权的情况进行比较，得到下图。具体代码见附录。

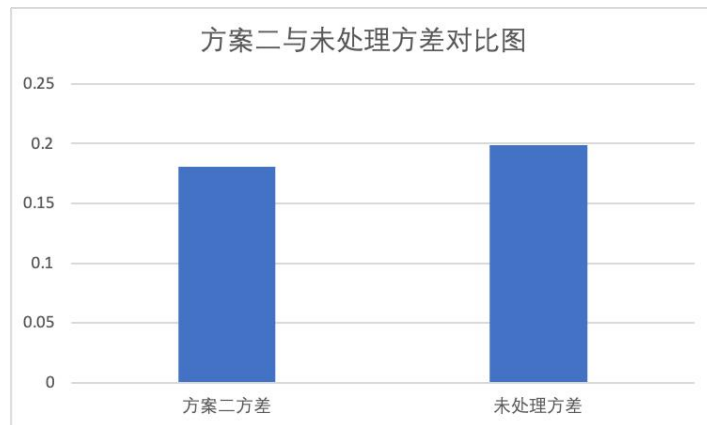


图 22 方案二与未处理方案方差对比图

5.4.4 结果分析

方案一，方案二与未处理方案对比，如下图。

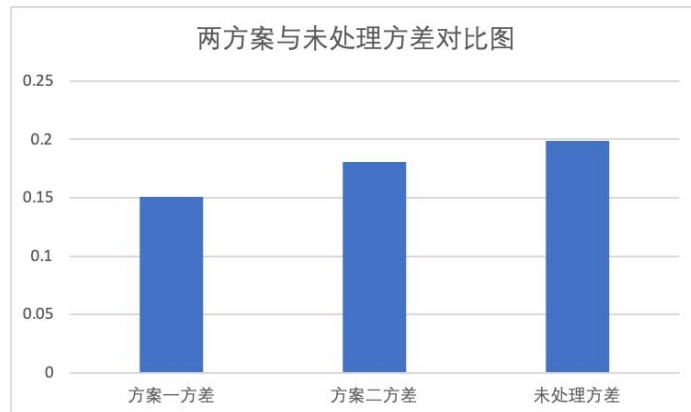


图 23 两种方案与未处理方案方差对比图

方案一方差最低，出租车司机时间效益最均衡。但需要另外设置特殊队列，对实际环境要求高，需要在蓄车池与乘车区之间增加一个缓冲区域。

方案二方差比未处理方案更低，比方案一方差高。出租车司机时间效益较为均衡，但其对实际环境要求不高。

两种方案各有优缺点，可根据实际情况进行选择。

6 模型的检验

我们基于 VISSIM 平台，设置同样的参数，对于双车道并行的情况，对泊车区的情况进行了仿真。我们通过改变泊位区的车位数量，从 $2 \times 1, 2 \times 2, 2 \times 3, \dots, 2 \times 10$ 这十种情况中得到了 VISSIM 平台下的通行能力是数据，并与问题二第一问中的数据进行比对，结果如下表。

表 4 两种模型下的通行效率对比表

泊位数量	2×1	2×2	2×3	2×4	2×5
本文模型	172.95	239.50	312.40	360.04	431.13
VISSIM	170.33	238.43	312.67	360.70	430.65
泊位数量	2×6	2×7	2×8	2×9	2×10
本文模型	462.32	522.90	565.39	602.13	647.30
VISSIM	461.58	521.99	564.74	601.69	647.76

我们取

$$S = \frac{\sum_{i=1}^{10} (a_i - b_i)^2}{10}$$

其中 a_i 代表本文模型数据， b_i 代表 VISSIM 平台下数据。求解得 $S = 1.097$ ，误差非常小，检验了本文模型的正确性。

7 模型的评价与改进

7.1 模型的评价

7.1.1 模型的优点

- (1) 针对出租车司机个人和整体分别建立了模型，能够较好的为司机作出选择策略。
- (2) 第二问我们进行控制变量实验，分析模型对各因素依赖性，并结合实际数据分析模型合理性。
- (3) 设置乘客上车点时，采取了但上车点和多上车点比较的方式，选择出了乘车效率较高的方案。
- (4) 设置乘客上车时间点时，综合考虑了两种情况，并通过对比选择了较优的一种。

- (5) 最后一问的模型从能够保证出租车司机收益均衡出发,建立了时间效益目标函数,综合对比了两种优先权方案,各分析了优缺点。
- (6) 我们第三、四问建立的仿真模型能够较好的模拟得出乘客区的通行能力。

7.1.2 模型的缺点

- (1) 模型具有一定的局限性,不适用于所有的情况。
- (2) 当飞机到港时间不同导致乘客数量差距较大时,此模型具有一定误差。
- (3) 我们搜集的数据具有片面性,可能会不适用于所有的情况。

7.2 模型的改进

- (1) 第一问建立的两个决策模型中可以加入随机变量来模拟随机情况的出现。
- (2) 我们第三、四问的仿真模型可以考虑基于元胞自动机下实现。
- (3) 对于双车道情况下对出租车与乘客的安排,我们还可以考虑斜列式和混合式两种方案。

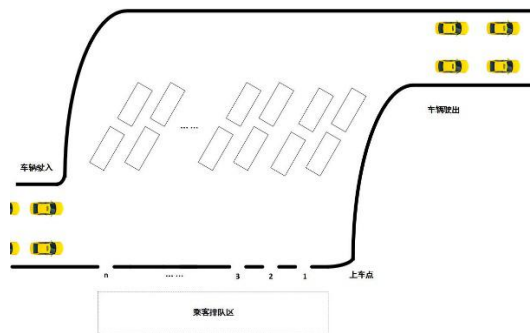


图 24 斜列式

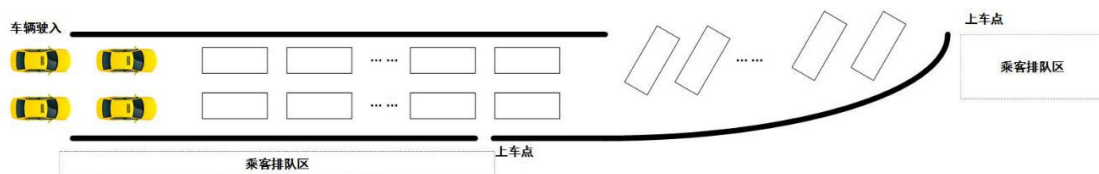


图 25 混合式

参考文献

- [1] 中国民用航空局[OL]. http://www.caac.gov.cn/XXGK/XXGK/TJSJ/201903/t20190305_194972.html
- [2] 深圳宝安国际机场[OL].<http://www.szairport.com/szairport/lwszsq/201311/0083ae3e374fdc992f04c608c45df5.shtml>
- [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] SIRISOMA R, WONG S C, LAM W H K, et al. Empiricalevidencefortaxicustomer-searchmodel[J]. ProceedingsoftheICE-Transport,2010,163(4):203210.
- [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] 姜启源, 谢金星, 叶俊. 数学模型 (第四版) [M]. 北京: 高等教育出版社, 2011.
- [7] 司守奎, 孙玺菁, 数学建模算法与应用[M]. 北京: 国防工业出版社, 2015.

附 录

A.1——问题二代码

```
function [ output_args ] = dierwen(A,W)
%A 为决策矩阵
%W 为权值矩阵
%M 为正指标所在的列
%N 为负指标所在的列

[ma,na]=size(A);
%%ma 为 A 矩阵的行数
%na 为 A 矩阵的列数
for i=1:na

    %按列循环得到[加权标准化矩阵]
    B(:,i)=A(:,i)*W(i);

end
%初始化正理想解和负理想解
V1=zeros(1,na);
V2=zeros(1,na);
%取加权标准化矩阵每列的最大值和最小值
BMAX=max(B);
BMIN=min(B);
for i=1:na

    %循环得到理想解和负理想解，注意判断，不然会超个数
    V1(i)=BMAX(i);
    V2(i)=BMIN(i);
%end
```

```

        %if i<=size(N,2)
        %V1(N(i))=BMIN(N(i));
        %V2(N(i))=BMAX(N(i));
        %end

end

%按行循环求各方案的贴近度
for i=1:ma

    %S1:到正理想点的距离
    %S2: 到负理想点的距离
    %均使用二阶范数
    C1=B(i,:)-V1;
    S1(i)=norm(C1);
    C2=B(i,:)-V2;
    S2(i)=norm(C2);
    %T 为贴近度
    T(i)=S2(i)/(S1(i)+S2(i));

end

output_args=T;

```

A.2 ——问题三代码

```

function [Efficiency] = disanwen1(coloumn)
%THIRD 此处显示有关此函数的摘要
%      此处显示详细说明
P=ones(2,coloumn);          %车道
CarTotal=0;                 %车的总数量
PersonTotal=0;              %人的总数量
TimeSpent=0;                 %所花费的总时间
time=-1;

```

```

rate=1.188;
%上车时间指数分布
time=-1;
for i=1:2000

    if((~P(1,1))&&(~P(2,1)))

        TimeSpent= TimeSpent+3+ (7-3).*rand();
        for j=1:column-1

            TimeSpent=TimeSpent+1.5+(3.5-1.5).*rand();

        end
        P=ones(2, column);
        continue;

    end

    for j=1:column

        if(P(2, column-j+1)>0)           % 2 排车位有车可以坐

            PersonTotal=PersonTotal+1;
            time=[time, exprnd(30)+(j-0.5)+(1).*rand()];
            CarTotal=CarTotal+1;
            P(2, column-j+1)=0;

        end

        if(P(1, column-j+1)>0)           % 1 排车位有车可以坐

            PersonTotal=PersonTotal+1;
            time=[time, exprnd(30)+(j-0.5)+(1).*rand()];
            CarTotal=CarTotal+1;
            P(1, column-j+1)=0;

        end

    end

end

%后面的车判断是否可以开
timetemp=zeros(2, column);
for p=1:2*column

```

```

        if p<coloumn
                timetemp(1,coloumn-p)=time(p);
        else
                timetemp(2,p-coloumn+1)=time(p);
        end
end
end

```

```

function [Efficiency] = disanwen3(coloumn)
%THIRD 此处显示有关此函数的摘要
%      此处显示详细说明
P=ones(2,coloumn);          %车道
CarTotal=0;                 %车的总数量
PersonTotal=0;             %人的总数量
TimeSpent=0;                %所花费的总时间
time=-1;
rate=1.188;
%上车时间指数分布
time=-1;
for i=1:2000

        if((~P(1,1))&&(~P(2,1)))

                TimeSpent= TimeSpent+3+ (7-3).*rand();
                for j=1:coloumn-1

                        TimeSpent=TimeSpent+1.5+(3.5-1.5).*rand();

                end
                P=ones(2,coloumn);
                continue;
        end
end
end

```

```

for j=1:coloumn

    if(P(2,coloumn-j+1)>0)           % 2排车位有车可以坐

        PersonTotal=PersonTotal+1;
        time=[time,exprnd(30)+(j-0.5)+(1).*rand()];
        CarTotal=CarTotal+1;
        P(2,coloumn-j+1)=0;

    end

    if(P(1,coloumn-j+1)>0)           % 1排车位有车可以坐

        PersonTotal=PersonTotal+1;
        time=[time,exprnd(30)+(j-0.5)+(1).*rand()];
        CarTotal=CarTotal+1;
        P(1,coloumn-j+1)=0;

    end

end

%后面的车判断是否可以开
timetemp=zeros(2,coloumn);
for p=1:2*coloumn

    if p<coloumn

        timetemp(1,coloumn-p)=time(p);

    else

        timetemp(2,p-coloumn+1)=time(p);

    end

end

for lieshu=1:coloumn

    for hangshu=1:2

        total=hangshu+lieshu;

```

```

                                timetemp(hangshu, lieshu)=timetemp(hangshu, lieshu)
+2*coloumn-total;
                                end
                                end
                                for k=2:coloumn
                                        for l=1:2
                                                if timetemp(l, coloumn-k+1)>timetemp(l, coloumn-k+1)
                                                        timewait=timetemp(l, coloumn-k+1)-timetemp(l,
coloumn-k+1);
                                                else
                                                        timewait=timetemp(l, coloumn-k+1);
                                                end
                                                timewait=max(timetemp);
                                                time=[time, timewait];
                                                end
                                end
                                end
                                TimeSpent=TimeSpent+max(time);
                                time=0;
                                TimeSpent=TimeSpent-coloumn;
                                end
                                Efficiency=(PersonTotal)/TimeSpent*rate*3600;
                                TimeSpent
                                end
                                function [Efficiency] = disanwen3(coloumn)
                                %THIRD 此处显示有关此函数的摘要
                                % 此处显示详细说明
                                P=ones(2, coloumn); %车道

```

```

CarTotal=0;      %车的总数量
PersonTotal=0;  %人的总数量
TimeSpent=0;    %所花费的总时间
time=-1;
rate=1.188;
%上车时间指数分布
time=-1;
for i=1:2000

    if((~P(1,1))&&(~P(2,1)))

        TimeSpent= TimeSpent+3+ (7-3).*rand();
        for j=1:coloumn-1

            TimeSpent=TimeSpent+1.5+(3.5-1.5).*rand();

        end
        P=ones(2,coloumn);
        continue;

    end

    for j=1:coloumn

        if(P(2,coloumn-j+1)>0)    % 2 排车位有车可以坐

            PersonTotal=PersonTotal+1;
            time=[time,exprnd(30)+(j-0.5)+(1).*rand()];
            CarTotal=CarTotal+1;
            P(2,coloumn-j+1)=0;

        end

        if(P(1,coloumn-j+1)>0)    % 1 排车位有车可以坐

            PersonTotal=PersonTotal+1;
            time=[time,exprnd(30)+(j-0.5)+(1).*rand()];
            CarTotal=CarTotal+1;
            P(1,coloumn-j+1)=0;

        end

    end
end

```

```

end

%后面的车判断是否可以开
timetemp=zeros(2, coloumn);
for p=1:2*coloumn

    if p<coloumn

        timetemp(1, coloumn-p)=time(p);

    else

        timetemp(2, p-coloumn+1)=time(p);

    end

end

end

for lieshu=1:coloumn

    for hangshu=1:2

        total=hangshu+lieshu;

timetemp(hangshu, lieshu)=timetemp(hangshu, lieshu)+2*coloumn-total;

        end

    end

for k=2:coloumn

    for l=1:2

        if timetemp(1, coloumn-k+1)>timetemp(1, coloumn-k+1)

timewait=timetemp(1, coloumn-k+1)-timetemp(1, coloumn-k+1);

```



```

        else

            timewait=timetemp(1, coloumn-k+1);

        end

        timewait=max(timetemp);
        time=[time, timewait];

    end

end

    TimeSpent=TimeSpent+max(time);
    time=0;
    TimeSpent=TimeSpent-coloumn;
end

    Efficiency=(PersonTotal)/TimeSpent*rate*3600;
    TimeSpent

End
function [finished] = disiwen2()
%FANGAN2 此处显示有关此函数的摘要
% 此处显示详细说明
SpentTime=100;
car=zeros(6000,8);
k=1;
finished=zeros(1,8);
temp10=zeros(1,8);
temp9=zeros(1,8);
temp8=zeros(1,8);
temp7=zeros(1,8);
temp6=zeros(1,8);
temp5=zeros(1,8);
temp4=zeros(1,8);
temp3=zeros(1,8);
temp2=zeros(1,8);
temp1=zeros(1,8);
num1=0;

```

```

num2=0;
num3=0;
num4=0;
num5=0;
num6=0;
num7=0;
num8=0;
num9=0;
num10=0;
for i=1:10
    for j=1:12
        car(k,1)=j;
        car(k,2)=i;
        k=k+1;
    end
end

end

wancheng=car(:,8);
while(length(wancheng(wancheng<1))>1)
    for i=1:length(wancheng)/2
        switch(car(1,2))
            case 1
temp1=[temp1;car(i,1),car(i,2),car(i,3),car(i,4),car(i,5),car(i,6),car(i,7),
car(i,8)];
            case 2
temp2=[temp2;car(i,1),car(i,2),car(i,3),car(i,4),car(i,5),car(i,6),car(i,7),
car(i,8)];
            case 3
temp3=[temp3;car(i,1),car(i,2),car(i,3),car(i,4),car(i,5),car(i,6),car(i,7),
car(i,8)];
            case 4
temp4=[temp4;car(i,1),car(i,2),car(i,3),car(i,4),car(i,5),car(i,6),car(i,7),
car(i,8)];
            case 5
temp5=[temp5;car(i,1),car(i,2),car(i,3),car(i,4),car(i,5),car(i,6),car(i,7),

```

```

car(i, 8)];
        case 6

temp6=[temp6;car(i, 1), car(i, 2), car(i, 3), car(i, 4), car(i, 5), car(i, 6), car(i, 7),
car(i, 8)];
        case 7

temp7=[temp7;car(i, 1), car(i, 2), car(i, 3), car(i, 4), car(i, 5), car(i, 6), car(i, 7),
car(i, 8)];
        case 8

temp8=[temp8;car(i, 1), car(i, 2), car(i, 3), car(i, 4), car(i, 5), car(i, 6), car(i, 7),
car(i, 8)];
        case 9

temp9=[temp9;car(i, 1), car(i, 2), car(i, 3), car(i, 4), car(i, 5), car(i, 6), car(i, 7),
car(i, 8)];
        case 10

temp10=[temp10;car(i, 1), car(i, 2), car(i, 3), car(i, 4), car(i, 5), car(i, 6), car(i,
7), car(i, 8)];
        end
        num8=temp8(:, 6);
        num1=temp1(:, 6);
        num2=temp2(:, 6);
        num3=temp3(:, 6);
        num4=temp4(:, 6);
        num5=temp5(:, 6);
        num6=temp6(:, 6);
        num7=temp7(:, 6);
        num9=temp9(:, 6);
        num10=temp10(:, 6);
        num1=length(num1(num1>0));
        num2=length(num2(num2>0));
        num3=length(num3(num3>0));
        num4=length(num4(num4>0));
        num5=length(num5(num5>0));
        num6=length(num6(num6>0));
        num7=length(num7(num7>0));
        num8=length(num8(num8>0));
        num9=length(num9(num9>0));
        num10=length(num10(num10>0));

```

```

zuidazhi=zeros(1,1);

zuidazhi=[zuidazhi,num1,num2,num3,num4,num5,num6,num7,num8,num9,num10];
zuidazhi=max(zuidazhi);
switch (zuidazhi)
    case num1
        begin=1;
    case num2
        begin=2;
    case num3
        begin=3;

    case num4
        begin=4;
    case num5
        begin=5;
    case num6
        begin=6;
    case num7
        begin=7;
    case num8
        begin=8;
    case num9
        begin=9;
    case num10
        begin=10;
end
end
SpentTime=SpentTime+100;
[x,]=length(car);
if x>6
    for i=1:4
        car(i,8)=1;
        car(i,5)=SpentTime;
        car(i,3)=1;
        finished=[finished;car(i,:)];
        car(i,:)=[];
    end
    for i=5:6
        if car(i,6)==0 %只是第一轮
            car(i,5)=SpentTime;

```

```

        a=1800.*rand();
        c=mod(a,100);
        a=a-c;
        car(i,6)=a;
        car(i,4)=1;
        car=[car;car(i,:)];
        car(i,:)=[];
    else
        car(i,7)=SpentTime-car(i,5)-car(i,6);
        car(i,8)=1;
        finished=[finished;car(i,:)];
        car(i,:)=[];
    end
end
end
else
    for i=1:x
        if car(i,6)==0
            car(i,5)=SpentTime;
            a=1800.*rand();
            c=mod(a,100);
            a=a-c;
            car(i,6)=a;
            car(i,4)=1;
            car=[car;car(i,:)];
            car(i,:)=[];
        else
            car(i,7)=SpentTime-car(i,5)-car(i,6);
            car(i,8)=1;
            finished=[finished;car(i,:)];
            car(i,:)=[];
        end
    end
end

end

end
end
function [info] = disiwen1()
%FORTH 此处显示有关此函数的摘要
% 此处显示详细说明
m1=30; %长途赚的钱

```

```

m2=10; %短途赚的钱
CarTotal=6000; %车的总数量
ShortRate=0.2 %短途的比例
LongRate=1-ShortRate; %长途的比例
ShortQueue=[1,2]; %短途出租车第一次出发时的时间和返回的时间
Time=100; %一次泊位的时间
info=zeros(1,1); %存储车的盈利情况
i=0;
TimeSpent=100;
Return=ones(1,2); %第一列存储短途车返回时间、第二列存储车的数量
Total=zeros(6000,3);
LongProfit=0;
ShortProfit=0;
sign=0;
while i<=CarTotal

    ShortNum=ceil(6*normrnd(ShortRate,0.05,1)); %出现了若干辆短途出租车
    Return=[Return;TimeSpent+1800,ShortNum];
    LongNum=6-ShortNum; %长途出租车的数量

    for n=1:ShortNum
        ShortQueue=[ShortQueue;TimeSpent,TimeSpent+1800];
    end

    [hang,]=size(Return); %求短途返回的出租车数量
    for j=1:hang
        if TimeSpent==Return(j,1)
            sign=hang;
        end

    TimeSpent=TimeSpent+Time; %车辆开始进入

    i=i+6;
    bijiaolie=ShortQueue(:,1);
    if length(bijiaolie(bijiaolie>0))>=6
        k=0;
        [h,]=size(ShortQueue);
        while k<6
            for z=1:h
                if(ShortQueue(z,1)>0)
                    a=30.*rand();

```

```

info=[info;(a+30)*60/(ShortQueue(z,1)+ShortQueue(z,2)-TimeSpent-900.*rand())
];
        ShortQueue(z,1)=0;
        k=k+1;
    end
    end
    end
else
    h=length(bijiaolie(bijiaolie>0));
    h2=6-h;
    k=0;
    while k<h
        for z=1:length(bijiaolie)
            if(ShortQueue(z,1)>0)
                a=30.*rand();
            end
        end
    end

info=[info;(a+30)*60/(ShortQueue(z,1)+ShortQueue(z,2)-TimeSpent-900.*rand())
];
        ShortQueue(z,1)=0;
        k=k+1;
    end
    end
    end
    for k=1:h2
        info=[info;30/TimeSpent*60];    %长途出租车的收入情况
    end
end
end

end
end

function [info] = disiwen3()
%FORTH 此处显示有关此函数的摘要
% 此处显示详细说明
m1=30; %长途赚的钱
m2=10; %短途赚的钱
CarTotal=6000; %车的总数量
ShortRate=0.2; %短途的比例

```

```

LongRate=1-ShortRate; %长途的比例
ShortQueue=[1,2]; %短途出租车第一次出发时的时间和返回的时间
Time=100; %一次泊位的时间
info=zeros(1,1); %存储车的盈利情况
i=0;
TimeSpent=100;
Return=ones(1,2); %第一列存储短途车返回时间、第二列存储车的数量
Total=zeros(6000,3);
LongProfit=0;
ShortProfit=0;
sign=0;
while i<=CarTotal

    ShortNum=ceil(6*normrnd(ShortRate,0.05,1)); %出现了若干辆短途出租车
    Return=[Return;TimeSpent+1800,ShortNum];
    LongNum=6-ShortNum; %长途出租车的数量

    TimeSpent=TimeSpent+Time; %车辆开始进入

    i=i+6;

    for k=1:ShortNum
        a=30.*rand();
        info=[info;(a+30)*60/(TimeSpent+a+Time)];
    end

    for l=1:LongNum
        info=[info;30*60/(TimeSpent)];
    end

end

end
end

```