

机场出租车的调度问题

摘要： 本文首先用 PCA 主成分分析排序影响因素，根据相关程度逐次建模，构建司机的博弈效益指数 $Cost$ 作为决策依据。然后建立先验模型预测结果，与基于 NARX 时间序列模型优化的 BP 神经网络模型结果进行对比，并用蒙特卡洛方法进行仿真检验，证明了模型的合理性与对相关因素的依赖性。之后基于 $M/M/2$ 的排队论，设立决策变量、约束条件与目标函数，采用基于精英决策机制的多目标微粒群算法（MOPSO），求解出上车点数量与候车区车辆数。最后，本文选择采用单目标微粒群算法（PSO），求解结果短途司机优先权次数，并进行数据验证，并给出了一种提供“优先权”的方式。

在问题（一）中，考虑到出租车司机决策相关影响因素的多样性，首先构造 PCA 主成分分析法分析影响机场乘客数量的变化规律和出租车司机的收益的 9 个指标，进行相关性排序。再根据排序结果，以司机减少成本提高收益的原则为起点，进行实证分析，逐次建立出司机经济效益的变量：单位价时比模型、出租车日间-夜间计价模型、ARIMA 预测等待载客花费时间模型以及时间误差的 Logistic Regression 逻辑回归预测模型。然后综合上述模型变量，以效益为目标函数，建立司机的博弈效益指数 $Cost$ 。若某时刻计算 $Cost > 0$ ，说明去机场收益高于在市区，此时决策为留在机场，反之则回市区。

在问题（二）中，本文首先构建了先验预测模型，量化司机的已知信息与经验判断，预测出租车收益结果。然后在完成对深圳市数据采集提取后，用三点估计法作为计量依据，将 10 月 8 日至 10 月 13 日数据作为训练集，10 月 14 日数据作为验证集，代入基于 NARX（时间序列自回归优化）的 BP 神经网络模型中，求解出 14 日白天、夜间情况下市区与机场的出租车收益。在此基础上，将先验模型与 NARX-BPNN 模型曲线进行对比，发现契合度较高。结果共同显示出出租车司机在早上五点到下午三点，下午五点半到次日凌晨一点更愿意在机场接客，其余时间更愿意送完客后回到市里。最后用蒙特卡洛方法仿真对预测模型进行多次模拟检验，得到最终期望预测值，结果基本与 NARX-BPNN 模型结果一致，进一步证明了模型的合理性和对相关因素的依赖性。

在问题（三）中，首先基于排队论 $M/M/2$ 的排队方式定义乘车效率，引入乘客等待概率函数、乘客滞留机场时间函数。再以上车点数与候车区车辆数目为决策变量，上车点数量为约束条件，设立目标函数为司机每小时收益最大与乘客总等待时间最短。在此基础上，针对此动态规划问题，本文采用基于精英决策机制的多目标微粒群算法（MOPSO），求解上车点数与候车区车辆数目。最后，本文代入问题（二）收集到的 14 日数据作进行验证，对模型进行规划求解。求解得候车区数目区间中心值为 3218，A 路上车点数 10 个，B 路上车点数 4 个。

在问题（四）中，首先在问题（三）的出租车收益函数基础上设立长短途出租车收益模型。然后建立优先权规则，得出约束条件：短途司机优先权次数使用次数大于 1 次小于 60 次，建立目标函数：长短途车差值最小。最后采用单目标微粒群算法（PSO），求解结果即短途司机优先权次数，经优化后为 5 次。最后带入数据进行正确性检验并给出一个解决方案。

关键词： ARIMA 算法；NARX-BPNN 模型；微粒群算法；蒙特卡洛仿真

1 问题的重述

1.1 问题背景

近年来国内航空运输业的蓬勃发展，各个国内的大型枢纽机场为提升运力，进行了诸如扩建，引入大数据科学管理系统等等举措，这些措施在提升机场客流吞吐量上，卓有成效。然而随着机场客流吞吐量的不断提升，中短途客运需求亦在不断提升，但是绝大多数机场路侧交通的接续运输能力并没有随之提升。机场“转乘难”问题，屡屡被乘客所批评，如何提高机场路侧交通的接续运输能力，是提高综合运输效率和机场服务质量的一大关键因素

“机场出租车”作为转乘交通的主要工具之一，始终存在着“拒载短途乘客”、“上客区不便于上客”等问题。通过结合乘客数量变动规律，建立出租车司机选择决策模型，给出司机的选择策略，通过分析司机的行为模式，有助于机场管理者，针对出租车运营者行为偏好，给予合适激励，最大程度提高乘客乘车效率，与司机接单意愿。

1.2 问题的提出

问题一：在综合考虑机场乘客数量的变化规律后，结合出租车司机的收益曲线，随后影响出租车司机决策结果相关因素的影响机理，最后建立出租车司机选择决策模型，并给出司机的选择策略。

问题二：收集国内某一城市的机场与出租车的相关数据，将数据引入问题（一）的决策模型，得出该机场出租车司机的选择方案，最后分析模型的合理性和对相关因素的依赖性。

问题三：为了减少出租车排队载客和乘客排队乘车的时间，对拥有两条并行车道的机场“乘车区”，进行上车点设计，在保证车辆和乘客安全的条件下，合理安排出租车和乘客，使得总的乘车效率最高。

问题四：机场出租车的载客收益由载客期的行驶里程决定，显而易见短途载客的出租车收益低于长途载客的出租车。因此在允许出租车多次往返载客情况下，机场管理者将给予短途载客且再次返回的出租车一定的“优先权”，给出一个可行的“优先”安排方案，使得该机场出租车的收益尽量均衡。

2 问题的分析

2.1 问题一的分析

问题一要求给出与出租车司机决策相关因素的影响机理，综合考虑机场乘客数量的变化规律和出租车司机的收益，建立出租车司机的选择策略模型。题目指出司机决策，归根结底都是受收益影响，司机决策的最终目的，在于减少成本提高收益。

此外司机可观测的确定信息包含有：某时间段抵达的航班数量和“蓄车池”里已有的车辆数，司机的经验通过判断某个季节与某时间段抵达航班的多少和可能乘客数

量的多寡等数据影响司机的决策。因而本文需要对在实际中影响出租车司机决策的确定与不确定因素，进行机理分析。

本文首先选取出租车辆数据、乘客数量、天气 3 类因素，9 个指标分别是：出租车进机场数；出租车出机场数；滞留机场出租车数目；机场有出租车需求的乘客数；平均需求人车比；机场需求出租车台数；温度；湿度；风速；压力；天气情况，其中天气情况进行数值化处理：例如晴 or 多云记为 1，小雨记为 2。随后进行 PCA 分析，为研究多种因素的影响机理，问题（一）在方法上，本文建立了基于预测机制的司机调度决策模型。

该模型的具体方案是：利用时间序列自回归模型（Nonlinear autoregressive exogenous model，NARX）与 BP 神经网络进行 2 维数值预测，该模型可以通过上一时刻数据分析出下一时段出租车进出机场数、需求数与滞留数等预测值。

2.2 问题二的分析

题目要求问题（二）在进行数据采集后，对问题（一）模型进行数据验证。首先在数据的采集上，本文在出租车运营数据上利用了深圳市政府公开的运营车辆数据集，时间周期为 2018 年 10 月 8 日至 10 月 14 日，进行数据清洗后有 2 万条数据；符合大样本量要求，在逐时航班数据来自于飞常准（该网站与民航局有数据合作），在逐时天气数据上利用了 Wunderground（全球气象统计）。在出租车载客里程上，由于里程数据过多，超出运算设备的能力要求，本文利用三点估计的方法提取（17km, 20.69%）、（37km, 62.94%）、（40km, 16.37%）这三个里程点作为计量依据。随后将 10 月 8 日至 10 月 13 日作为训练集，10 月 14 日作为验证集，代入 NARX 时间序列算法与 BP 神经网络训练模型。从而求解出 14 日白天、夜间情况下市区与机场的出租车收益。

在模型的合理性上，本文构建了司机的先验预测模型。即量化问题（一）中司机的已知信息与经验判断能力。利用 10 月 13 日数据模拟司机对 14 日的预测，得出此时出租车收益结果。然后与 NARX 模型结果进行对比，即可证明模型优势。

本文利用蒙特卡洛实验法，模拟司机的先验模型，由于预测误差满足正态分布，所以用蒙特卡洛实验法，对预测模型进行多次实验，得到最终期望预测误差用预测结果代替了出租车司机先验模型，经过目标值计算后发现收益提高了

2.3 问题三的分析

问题（三）指出，在某时刻的机场出租车候车区，经常会出现出租车排队载客和乘客排队乘车的情况。机场管理部门需要设置“上车点”，在保证车辆和乘客安全的条件下，合理安排出租车和乘客，使得总的乘车效率最高。

首先本文在建立目标函数前，本文为保证模型真实性，在出租车的时间成本上引入了新的参数：乘客滞留机场时间，即乘客航班到达后，由机舱走出，提取行李后行走至上车点的用时，由于缺少相关数据，本文利用正态分布函数进行拟合，同时假设正常情况下滞留用时最大不超过 1 小时。随后在此基础上针对此优化问题，依据决策变量：上车点数与候车区车辆数目，约束条件：上车点数量，可改变值为设立对应目标函数：司机每小时收益最大与乘客总等待时间最短。

最后在规划问题的求解上，由于这是一个动态规划问题，本文采用基于精英决策机制的多目标微粒群算法（MOPSO），求解结果上车点数与候车区车辆数目。在数据验证上，本文代入问题（二）收集到的 14 日数据，对模型进行规划求解。

2.4 问题四的分析

在一般情况下，出租车司机的长途载客收益远大于短途收益，因而司机在意愿上倾向于承接长途订单（订单里程数大于 25 公里）。因而问题（四）中，机场管理部门为降低拒载率，准备通过给予短途载客“优先权”，即缩减短途司机的等待时间，从而达到提高短途司机收益的目的，最终使得长短途司机的收益持平，因而“优先”安排方案，也可以看作一个优化问题。

首先在问题（三）的出租车收益函数基础上，分别设立长短途出租车收益模型。其中出租车的行驶里程分布，为了便于数据验证，这里使用问题（二）中，三点估计法所提取的三个乘客载里程，作为长短途司机平均收益预测依据。在“优先权”规则上：长途车无“优先权”，短途车“优先权”的由里程距离决定，优先权只能使车辆位置提升至“弱优先权”车辆的下一位置。从而得出约束条件：短途司机优先权次数使用次数大于 1 次小于 60 次，随后建立目标函数：长短途车差值最小，其中被“插队后”的长途车等待时间需包含所有短途车等待时间，而短途车需要则是包含所有里程更短途的车辆等待时间，最后试算短途车需要行使几次“优先权”，即增加长途车等待时间，才能收益与长途车持平。

最后在规划问题的求解上，问题（四）同样是一个动态规划问题，本文选择采用单目标微粒群算法（PSO），求解结果即短途司机优先权次数，在数据验证上，本文代入问题（二）收集到的 14 日数据，对模型进行规划求解。

3 模型的假设及符号的说明

3.1 模型的假设

（1）模型适用于仅拥有单一机场的城市，不存在一城市多机场的市场竞争、分流问题。

（2）假设出租车司机驶出机场用时少于 5 分钟，机场待客司机为：GPS 定位处于机场停车区，且速度为零；返回市区接客司机 GPS 定位同处于机场停车区，但速度大于 0。

（3）假设每辆出租车平均乘车人数 1.25~1.5 人，具体数值调整依据具体日期情况进行调整。

（4）假设依据宝安机场实际航班时刻与航班数据测算，宝安机场航班机型最大载客能力为 241 人，平均上座率 80%。

（5）深圳出租车车费：起步价 10 元/2km，里程价 2.6 元/km，候时费 0.8 分钟，夜间（23.00-6.00）起步价 13 元，夜间里程价 3.38 元/km，燃油附加费 1 元，返空费：1.3 元/公里（25km<总里程<50），1.6 元/公里（50km<总里程）。

（6）假设选择搭乘出租车的乘客，以及等待接客的出租车，都遵守机场规章制度，在指定区域，上车、载客。

3.2 符号说明

符号	符号注释
t	司机低速行驶率
V	出租车行驶速度
m_x	出租车计价标准
Z	停候车区域机场出租车辆数
z	机场乘客人数
J_t	进入机场出租车数
C_t	离开机场出租车数
$Cost$	决策变量
x_t	乘客需求率（出租车）
R	出租车收益
d_c	乘客出行距离
$P(t_c)$	机场滞留时间的概率分布函数
t_c	乘客机场滞留时间（离散型分布）
$E(t_c)$	乘客机场滞留时间期望值
i	每小时从上车点驶离机场出租车数
B	上车点数
$Costomer$	乘客等待时间

4 模型的准备

4.1 数据的采集

4.1.1 出租车数据^[1]

本文使用的是深圳市出租车实时动态行车数据，数据采集时间周期为 2018 年 10 月 8 日至 10 月 14 日。深圳市出租车实时动态行车数据是运用 GPS 定位设备记录的出租车行驶位置、速度、方向、里程等数据。其数据的内容过于充实。本文需要对其部分数据条目进行剔除，因此在将数据带入模型建立之前需要对数据进行预处理：

①以机场下车区 GPS 位置为圆心，以 0.0005° （ $\approx \pm 55$ 米）为半径作为筛选条件，即处在东经 $113.8222^\circ \sim 113.8232^\circ$ ，北纬 $22.6215^\circ \sim 22.6225^\circ$ 范围内，提取符合问题要求的样本。

表格 1：出租车实时动态数据样例

车牌号	地图经度	地图纬度	GPS 时间	方向	行驶记录仪速度	里程
粤 B*****	113.82087	22.612825	20181008/155531	151	56	11817
粤 B*****	113.82059	22.613325	20181008/170922	151	47	11830
.....
粤 B*****	113.82082	22.612825	20181008/033521	149	0	31290
粤 B*****	113.81960	22.614248	20181008/223214	333	59	20364

②对①中数据，使用 EXCEL 软件的数据透视表进行分类计量，分时段汇总区分速度为 0 与速度>0 的出租车速度。

表格 2：出租车数据（筛选后）样例

时段	8 日（留机场/速度=0）	8 日（出机场/速度>0）
0	949	393
1	1489	794
2	2101	518
.....
22	739	256
23	800	261
总计	12292	7483

4.1.2 天气数据^[2]

深圳市天气数据是通过 Wunderground 网站采集的，其中数据信息包含气温、露点、风向、风速、大气压、阴晴，以三个小时为间隔。本文选择了与出租车信息相对应的 2018 年 10 月 8 日至 14 日数据，因该网站为外国网站，所用信息以外国习惯为主，如使用华氏度。且包含有非量化因素，为使程序能够更好的识别数据，需要进行数据预处理。

①将数据中的华氏度转换为摄氏度，公式为 $C = \frac{F - 32}{1.8}$

F 为华氏度， C 代表摄氏度。

②量化数据。原数据风向表示为东西南北，指向标规则：令北向为 0° ，顺时针旋转为正，量化风向，如东向风为 90° 。将原数据中阴晴天气进行分类并赋予不同的权值以反映其特征，如表所示：

表格 3：天气数据样例

天气分类	1	2	3	4
气象特征	晴、多云	小到中雨、雪	中到大雨、雪，雾、霾	恶劣、极端天气
影响权数	0%	0%~10%	-10%~30%	20%~60%

其中第三类天气影响权数出现负数，即出现该类天气可能会导致进港人数增多，这可能是因为航班延误导致航班扎堆进入机场，使得某个时刻进港人数增多。

4.1.3 航班数据^[3]

航班数据来自飞常准网络平台，同样根据出租车数据选择记录 8 日至 14 日所有到达与驶离航班信息。根据民航局相关信息显示，该时段航班绝大多数按照 2018 年夏秋航班计划表安排班次，给数据统计工作带来了便利。航班数据的作用是通过航班数据判断进港乘客人数。部分进港航班信息，如下表所示：

依据此类数据，通过对飞机型号、载客能力以及平均上座率进行初步加总，即可得出某时时刻，深圳机场有转运需求的乘客数量。

表格 4：航班数据样例

航班号	航空公司	飞机型号	往来城市	出发时间	到达时间
ZH9890	深圳	A320	合肥-> 深圳	6:30	8:35
CZ8596	南方	A321	合肥-> 深圳	6:30	8:45
.....
SC1181	山东	B738	济南-> 深圳	6:35	9:25

4.2 数据的预处理

综上，本文将收集到绝大多数原始数据，对数据进行初步简单处理后，由于本文采集与处理过程同时进行，因此这里不再追叙，汇总至同一 EXCEL 文件，样例表的标题行与第一行数据如下。

表格 5：汇总表样例

Time1008-1014	出租车进机场数	出租车出机场数	滞留机场出租车数目	机场出租车需求乘客	平均需求人数/台数	机场需求出租车台数
0-8	949	393	41	547	3	183
1 行·续	温度/° C	湿度	风速/mph	压力/in. w. c	天气（晴多云 1，雨 2	
2 行·续	24.44	0.89	7	29.9	1	

5 模型的求解

5.1 问题一的模型建立与求解

5.1.1 司机决策策略的博弈研究

司机的决策与多个因素有关,本文综合考虑机场乘客数量变化与出租车司机收益,选择出租车数量与天气因素两大类因素.其中出租车数量因素包括出租车进出机场数、滞留数与需求量共4因素,天气因素包括温度、湿度、风速、大气压、阴晴共5因素,合计9因素.司机的选择只有两种:前往到达区排队等待载客返回市区、直接放空返回市区拉客。

第一种需要到指定地排队等候,排队进场载客,等待时间长短取决于排队出租车和乘客的数量多少,付出一定的时间成本.第二种需要付出空载费用和可能损失潜在的载客收益.

针对9种因素,首先抽取所有数据,找出影响要素并逐次降维,然后研究各要素之间相互影响的关系,进行主成分分析.

PCA（主成分分析法）的建立:

第一步:旋转坐标轴,

第二步:去掉一些特征,进而留下的特征可以作为原始数据的近似.

第三步:计算原始特征矩阵的协方差矩阵,得原始矩阵 X ,并且其每一列都要减去每一列的均值:

$$A = \left(\frac{1}{m-1} X^T X \right) \quad (1)$$

然后对协方差矩阵 A 进行特征值分解,根据分解得到的特征值和特征向量,按照特征值大小把对应的特征向量从小到大按列排成矩阵,去掉特征值小的最后几列,形成矩阵 A' , A' 的前几列就是主成分,成分排名如下表所示:

表格 6: 主成分排名表

主要成分	机场对运载能力需求	短时交通流情况	经济要素
因子组成	乘客数,即出租车数需求数	天气、车辆滞留数	司机效益
因子得分排名	3	2	1

说明:从乘客数量考虑,机场乘客中有一定比例的出站乘客,其中有一定比例选择乘坐出租车,乘客需求量大则司机更有倾向到机场寻找乘客,从而影响司机的决策.乘客通过数量变化影响司机决策,本文主要关注其需求出租车数,其计算公式为:

$$x_i = \frac{\text{人数}}{\text{车辆需求}} \quad (2)$$

其中 x_i 表示乘客对出租车的需求率.

5.1.2 模型建立：基于 NARX 时间序列自回归的效益博弈模型^[4]

PART 1 尝试性分析

经济利益是影响司机决策的主要成分. 经济学的实证分析可以发现, 出租车司机们在平等的对局中各自利用对方的策略变换自己的对抗策略, 达到盈利的目的, 司机对其他司机的特征、策略空间及收益函数信息了解的不够准确, 只能了解客观因素. 如果直接采用博弈论矩阵, 则会形成一个维度为司机数量 X 的超高维矩阵, 在参与者众多的情况下难以决策. 所以作为理性人, 机场出租车司机选择决策更多地取决于自己的收益, 如果相对直接放空返回市区拉客而言, 前往到达区排队等待载客返回市区可以达到更大的收益, 那么出租车司机就会选择前往到达区排队侯客. 定义预测司机的经济效益的变量: 单位价时比 R . 司机在日常情况下一天盈利 m_a , 运营时间 t_a , 则:

$$R_{daily} = \frac{m_a}{t_a} \quad (3)$$

当司机选择在机场接客时, 其付出路程时间 t_x , 等待时间 T_{wait} , 获得收益 m_x , 此时:

$$R_{do} = \frac{m_s + m_a}{t_x + T_{wait} + t_a} \quad (4)$$

当司机选择不接客, 直接返回市区时, 假设司机回城路线与接客后送客路线一致, 那么到达相同位置所需时间仍为 t_x , 此时考虑到司机在路上可能的接客情况以及路况的随机性, 新增一个时间的误差范围 t_0 :

$$R_{undo} = \frac{m_a}{t_x \pm t_0 + t_a} \quad (5)$$

t_0 在下文会进行详细地建模与讨论. 此时构建公式让上述两项相减比较大小:

$$R_{recommend} = R_{do} - R_{undo} = \frac{m_x}{m_a} - \frac{T_{wait} \pm t_0}{t_x \mp t_0 + t_a} \quad (6)$$

其中, 若 $R_{recommend} > 0$, 司机选择去等候接客, 反之直接回市里.

PART 2 出租车收益模型的初步构建

在建立深圳市出租车效益模型前, 假设出租车停驶率 $t\% = 10\%$, 白天平均车速 $V_d = 30$, 夜间平均车速 $V_n = 40$.

深圳市出租车计价标准，见下表：

表格 7：深圳市出租车计价标准

计价条目	计价数额	单位
起步价	10	元/2km
里程价	2.6	元/km
候时费	0.8	元/分钟
夜间（23.00-6.00）起步价	13	元/2km
夜间（23.00-6.00）里程价	3.38	元/km
燃油附加费	1	元
返空费（25km<总里程<50）	1.3	元/km
返空费（50km<总里程）	1.6	元/km

从而得出，出租车日间计价模型：

$$m_{x_d} = \begin{cases} 10 & , \quad d \leq 2 \\ (10 + 2.6 \times (d - 2) + 0.8 \cdot \frac{d}{V_A}) \times t\% & , \quad 2 < d \leq 25 \\ 69.8 + 3.38 \times (d - 25) + 0.8 \times (\frac{d}{V_A}) \times t\% & , \quad d > 25 \end{cases} \quad (7)$$

以及出租车夜间计价模型：

$$m_{x_n} = \begin{cases} 13 & , \quad d \leq 2 \\ 10 + 2.6 \times (d - 2) \times 1.3 + 0.8 \times (\frac{d}{V_A}) \times t\% & , \quad 2 < d \leq 25 \\ 90.74 + (d - 25) \times 4.394 + 0.8 \cdot (\frac{d}{V_A}) & , \quad d > 25 \end{cases} \quad (8)$$

其中 d 代表距离， $t\%$ 代表低驶率， V_A 代表出租车平均行驶速度

PART 3 司机等待载客花费时间：

旅客从飞机降落开始，直到到达出站口，需要经过走出飞机、到达行李厅、等待行李，取行李，到达出站口等一系列过程。旅客到达出站口所需时间 $T_{arrival}$ 由所有环节时间和决定；并且正常情况下，假设上述过程中所有环节旅客所花费时间相互独立。

因此，本文假设旅客从飞机降落时刻起到到达出站口共所需时间 $T_{arrival}$ 服从概率密度为 $f(T_{wait} = t)$ 的正态分布。这是一个动态的时间序列。本文拟采用 NARX 时间序列自回归模型进行预测。

本文通过在模拟在最坏情况下的决策来计算出租车司机等待乘客时间分布. 首先假设在某一时间段, 出租车司机均匀、随机、等概率地到达乘客候车区并排队等候.

此时出租车司机落入时间长度为 t 间隔的概率密度. 其取决于时间间隔出现的概率和时间间隔的长度. 时间间隔出现的概率越大, 出租车司机落入概率就越大; 时间间隔长度越大, 司机落入概率也越大. 因此认为, 司机落入时间长度为 t 间隔的概率密度与时间间隔本身出现的概率密度 $P(t) = \lambda e^{-\lambda t}$ 与时间间隔长度 t 之积 $P(t) \times t$ 呈现正比例关系.

其次, 当司机处于长度为 t 的时间间隔时, 最坏情况下他最多需要等 t 的时间就可以等到上车的乘客. 故而最坏情况下, 司机候客时间 $T_{wait} = t$ 的概率密度 $f(T_{wait} = t)$ 与 $P(t) \times t$ 成正比, 通过归一化可得:

$$P(t) = \lambda e^{-\lambda t} \quad (9)$$

$$f(T_{wait} = t) = \frac{P(t)t}{\int_0^{+\infty} P(t)t dt} = \frac{\lambda e^{-\lambda t} t}{\int_0^{+\infty} \lambda e^{-\lambda t} t dt} = \lambda^2 e^{-\lambda t} t \quad (10)$$

在司机候车时间分布的基础上计算累积概率分布即可获知司机等待一定时间后能载上客人的累积概率 $F(T_{wait})$ (简称乘车概率):

$$F(T_{wait}) = \int_0^{T_{wait}} f(t) dt = \int_0^{T_{wait}} \lambda^2 e^{-\lambda t} t dt = 1 - \lambda T_{wait} e^{-\lambda T_{wait}} - e^{-\lambda T_{wait}} \quad (11)$$

Part 4 时间误差的逻辑回归模型

在司机从机场接到乘客到将乘客送到目的地, 误差范围是以 t_x 为基准, 但不一定是中心点的一个范围, 于是本文结合 sigmoid 函数, 线性回归函数, 把线性回归模型的输出作为 sigmoid 函数的输入. 于是最后就变成了逻辑回归模型:

$$t_0 = \sigma(\omega^T x t_x) = \frac{t_x}{1 + e^{\omega T_x}} \quad (12)$$

ω 为待训练的参数, 所以需要利用一组采集到的真实样本, 构建损失函数, 训练出参数 w 的值. 将单个样本认为是一个事件, 那么此事件发生的概率就是:

$$P(y_i | x_i) = p^{y_i} (1 - p)^{1 - y_i} \quad (13)$$

采集到个数为 N 的数据, 合事件发生的总概率就是将每一个样本发生的概率相乘, 即采集到这组样本的概率:

$$P_\alpha = P(y_1 | x_1) P(y_2 | x_2) P(y_3 | x_3) \dots P(y_n | x_n) = \prod_{n=1}^N p^{y_n} (1 - p)^{1 - y_n} \quad (14)$$

是一个函数, 并且未知的量只 ω . 由于连乘很复杂, 本文通过两边取对数来把连乘变成连加的形式, 即损失函数为:

$$F(\omega) = \ln P_\alpha = \sum_{n=1}^N [y_n \ln p + (1 - y_n) \ln(1 - p)] \quad (15)$$

经过最大似然估计 MLE , 选取的某个 ω^* 刚好使得总概率 P_α 取得最大值的时候. 我们就认为这个 ω^* 就是我们要求得的 ω^* 的值. 从而可以确定随机值 $0.7t_x < t_0 < 1.05t_x$.

PART5 决策 1：出租车司机利益模型

在等候机场的利益越大，司机越有可能作出留在机场的决策，这是根本因素。因出租车白天夜间价格有所差别，故将两者分离讨论。首先要考虑当前机场出租车总数，其数越大，司机为获得利润付出的时间越多，也就越不利于留在机场决策。机场出租车总数目 z 可表示为：

$$z_{t-1} = \int_{-\infty}^{t-1} J_t dt - \int_{-\infty}^{t-1} C_t dt \quad (16)$$

式 Z_{t-1} 表示训练前一天的出租车总数， J_t 代表进车数， C_t 代表出车数。

将司机获得利润根据白天黑夜、机场市区分为四类分别计算。机场距离市区距离为 32km，根据获得数据用 Matlab 使用蒙特卡洛方法对全年司机的每小时收益 $R_{d_c} = 34.32$ 进行仿真模拟得到单辆出租车单日白天市区收益：

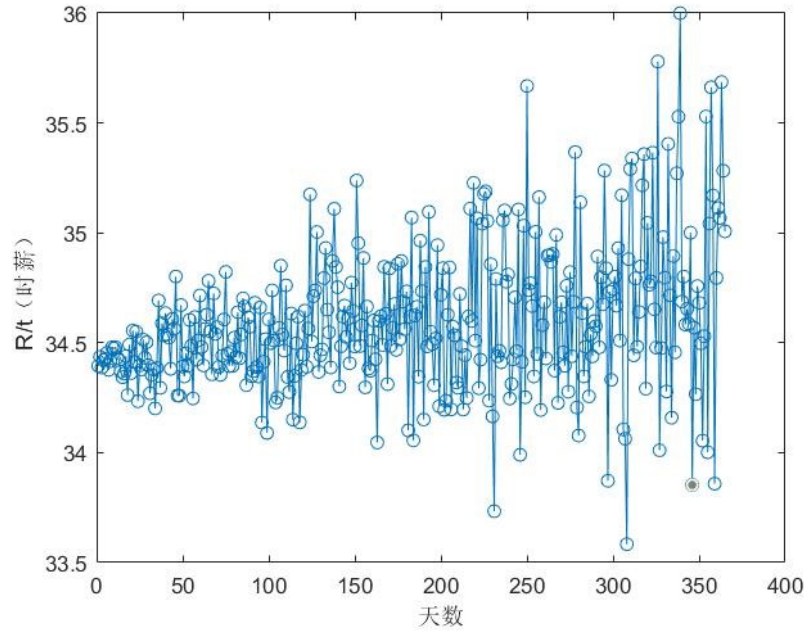


图 1：司机时薪统计图

从而得到司机的平均时薪： $R_{d_c} = 34.32$ （元/小时）

根据公式计算单日白天机场收益：

$$R_{d_a} = \frac{m_{x_d}(d_a)}{\frac{d_a}{V_d} + \frac{32}{V_d}} = \frac{m_{x_d}(d_a)}{d_a + 32} \cdot V_d \quad (17)$$

单日夜間市区收益

$$R_{n_c} = 34.32 \times 1.3 = 44.616 \quad (18)$$

单日夜间机场收益

$$R_{n_a} = \frac{m_{x_n}(d_a)}{\frac{d_a}{V_n} + \frac{32}{V_n}} = \frac{m_{x_n}(d_a)}{d_a + 32} \cdot V_n \quad (19)$$

其中 d_a 代表出租车平均行驶距离, $m_{x_d}(d_a)$ 代表白天出租车行驶距离为 d_a 下计价标准, V_d 代表出租车白天平均行驶速度, V_n 代表出租车夜间平均行驶速度.

本文依据三点估计法, 对乘客所乘出租车行驶距离进行概率估计, 得到结果:
市区出租车三点出行距离:

$$d_c = \begin{cases} 1.5, 20.69\% \\ 2.1, 62.94\% \\ 2.8, 16.37\% \end{cases} \quad (20)$$

机场出租车三点出行距离:

$$d_a = \begin{cases} 17, 20.69\% \\ 62, 62.94\% \\ 40, 16.37\% \end{cases} \quad (21)$$

将市区与机场的出租车平均收益在白天与黑夜分别作差, 记为 $Cost$, 其具体计算公式:

$$Cost = \begin{cases} \text{白天} \begin{cases} R_{d_a} - R_{d_c}, hour = 6 \sim 22 \\ -\infty, hour = 22 \sim 2 \end{cases} \\ \text{黑夜} \begin{cases} R_{n_a} - R_{n_c}, hour = 23 \sim 5 \\ -\infty, hour = 5 \sim 6 \end{cases} \end{cases} \quad (22)$$

若 $Cost > 0$, 说明去机场收益高于在市区, 此时决策为留在机场, 反之去市区.

5.2 问题二的求解与检验

5.2.1 基于 NARX 改进的 BPNN 模型求解

针对机场的选取, 为保证数据的完整性与普适性, 本文拟选取国内人流量名列前茅的几个机场; 同时考虑到数据的可靠性, 如果一个城市有两个及以上的机场, 司机的决策的复杂程度就会指数上升, 而且要考虑两个机场的相互影响, 数据的准确性也会大打折扣. 所以本文没有考虑北京与上海, 而是选取人流量靠前, 且只有一个机场的深圳作为本文的研究对象.

机场出租车的运载能力需求会因为航班安排、当日天气等影响以及疫情、自然灾害等爆发性事件具有不确定性与动态性, 与此同时, 也因为居民出行总体的规律性和交通条件约束具有周期性和相关性的影响, 出租车司机对当前客观情况的预测会受到很多随机因素的影响, 具有很大的时变性和不确定性. 针对这种不确定性, 本文使用 Matlab 建立 BPNN 模型在输入 9 种数据情况下经过上文公式作出司机 4 种数据的预测. 再通过 NARX 优化模型, 得出结果.

PART1 对 BPNN 进行设计

激活函数为单极性 S 型函数和双曲正切 S 型函数, 自动设置学习率 η . BPNN 的停止准则为网络的均方误差足够小或者训练足够的次数. 初始权值由数据读入. 隐层结构为单隐层, 函数为问题一所建立目标函数及其附属模型.

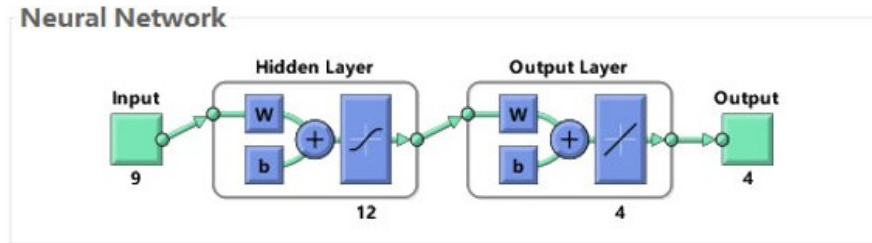


图 2: 基于 NARX 改进的 BP 神经网络流程图

PART2 对预处理好的数据进行训练.

首先初始化网络的突触权值和阈值矩阵; 然后进行训练样本的呈现; 再进行前向传播计算; 再进行误差反向传播计算并更新权值; 最后采用蒙特卡洛方法随机生成迭代, 用新的样本进行步骤 3 和 4, 直至满足停止准则. 本文共有 14 天的数据, 对前 13 天的数据进行 1000 次蒙特卡洛模拟训练, 使用第 14 天的数据进行预测.

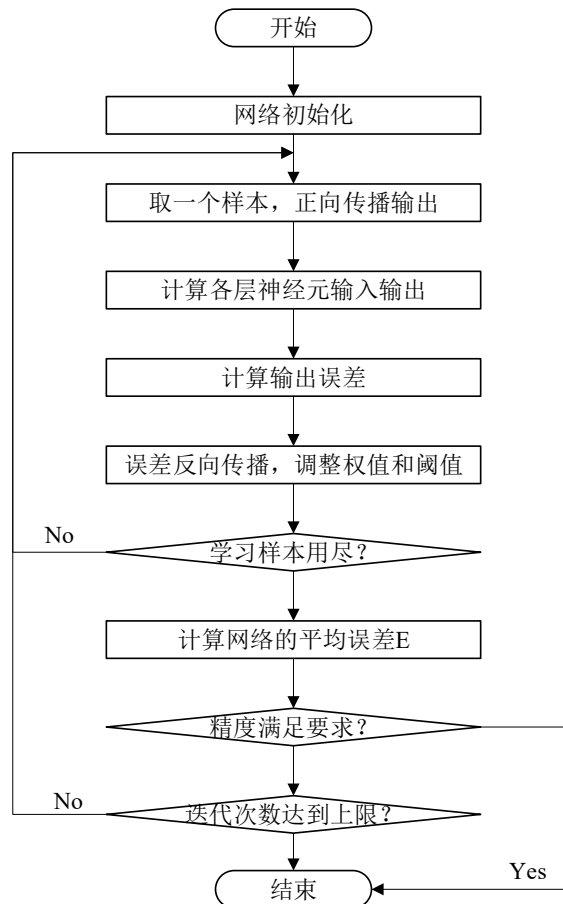


图 3: 机器训练逻辑图

PART3 基于 BP 神经网络模型的司机选择策略

利用得到的输出值, 经上文的公式可以算出机场乘客数量的变化规律和出租车司机的收益.

司机的决策模型中运用了三种应用广泛的预测模型: 均值模型、NARX 时间序列自回归模型和 BP 神经网络模型. 本文先基于 non-linear auto-regressive model with exogenous inputs (NARX) 时间序列自回归模型进行等待时间的预测. 在查找了更多具体的航班及旅客数据之后, 整理得到更多客流特征, 使用 BP 神经网络模型进行一步 4 维数值预测, 并在之后对 BP 神经网络模型进行优化, 达到更准确的下一时间段内乘客数量和出租车经济效益预测值的效果.

如图所示, 在平均线之上的时间段司机留在机场收益高于市区, 在早上五点到下午三点, 下午五点半到次日凌晨一点更愿意在机场接客.

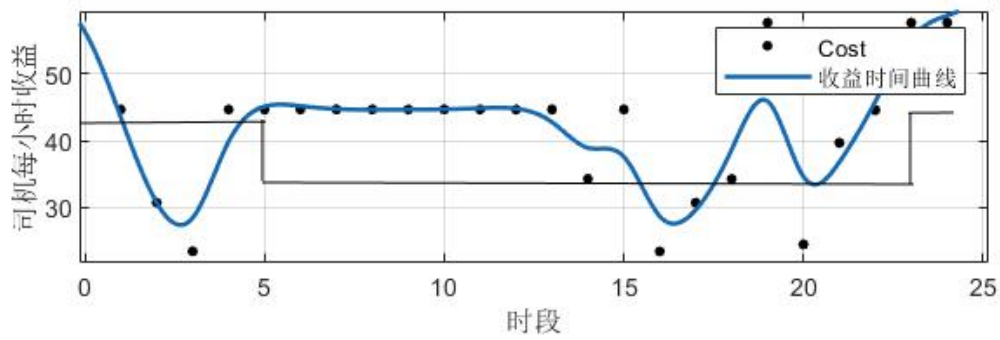


图 4: 司机每小时收益预测拟合曲线

5.2.2 问题二的程序优化

NARX-BPNN 预测. 为模拟真实情况, 本文先站在司机的角度考虑这个问题.

Case 1: 不使用 NARX-BPNN, 建立先验模型定义 $Cost$ 1

不确定数量的先验概率分布是在考虑一些因素之前表达对这一数量的置信程度的概率分布, 为了保证预测的准确性, 先建立先验模型. 先验模型中司机知道的数据: 当时的时刻、当时的天气、上个时段滞留的出租车数. 只根据司机所知的条件对司机当前决策进行判断. ε 是司机主观判断参数, 取以 1 为中心的正态分布.

$$Cost_t = R_0 \times f(t) \times e \times \frac{Z}{Z} \quad (23)$$

$$Cost_t = Cost_{t-1}$$

运行先验模型, 得到数据 $R_{do} = 41.4802$, $R_0 = 36.969$, 因此此时 $R_{recommand} > 0$ 故此时司机应当留在机场.

Case 2: 采用本方案后

$$Cost_t' = \Delta(J_{t-1} \dots \dots wea_{t-1}) \quad (24)$$

运行 BPNN 模型. 经过 NARX 不断迭代, 最终预测值 $R_{do} = 43.7228$, $R_0 = 36.969$, $R_{recommand} > 0$, 从而验证此时司机应当去机场

5.2.3 蒙特卡洛方法仿真模拟检验^[5]

定义 1: 司机滞留机场意愿 x

司机去机场送客后选择排队的意愿为 x ， x 越大，代表司机越愿意在机场接人。

本文拟采用梅特罗波利斯算法 (Metropolis Method) 通过构建一个稳态系统，在状态之间转移采样的方式对司机个人意愿进行模拟。该稳态系统基于马尔可夫链，生成新样本只依赖于上一个样本和状态，具有更好的适用性。

1、给定一个初始值作为当前值，初始值假定为 0。

2、对于每个当前值 x ，根据建议分布函数随机一个新样本 x^*

系统以一定的概率 $rand() < A$ 接受 x^* 成为新当前值，若未接受，当前样本再次被接纳，并再次成为当前值。样本接受率：

$$A = \min(1, \frac{eval(x^*)}{eval(x)}) \quad (25)$$

梅特罗波利斯算法要正常运行，必须使所构建的系统经过足够多的状态转移后，收敛到一个平稳状态。到达平稳状态后，下一次转移到某一个状态（具体到这里是采样点）的概率就成为一个平稳概率分布（此时采样点被选中的概率就接近于目标分布函数中该点的值）。而要构建这样一个稳态系统，有一个充分非必要条件是（即细致平衡条件）：

$$eval(x_n) \times ref(x' | x_n) \times \alpha(x' | x_n) = eval(x') \times ref(x_n | x') \times \alpha(x_n | x') \quad (26)$$

将目标概率函数作为建议分布函数，有：

$$ref(x' | x_n) = normal(x | x_n, \sigma^2) \quad (27)$$

$$ref(x' | x_n) = ref(x_n | x') \quad (28)$$

又因为：

$$\alpha(x_n | x') = \min(1, \frac{eval(x')}{eval(x)})$$

可证：

$$eval(x_n) \times \alpha(x' | x_n) = eval(x') \times \alpha(x_n | x') \quad (29)$$

满足细致平衡条件，即最终可以收敛到稳态分布。训练量分别为 100、500、1000、5000 时最终获得的样本数据分布如下：

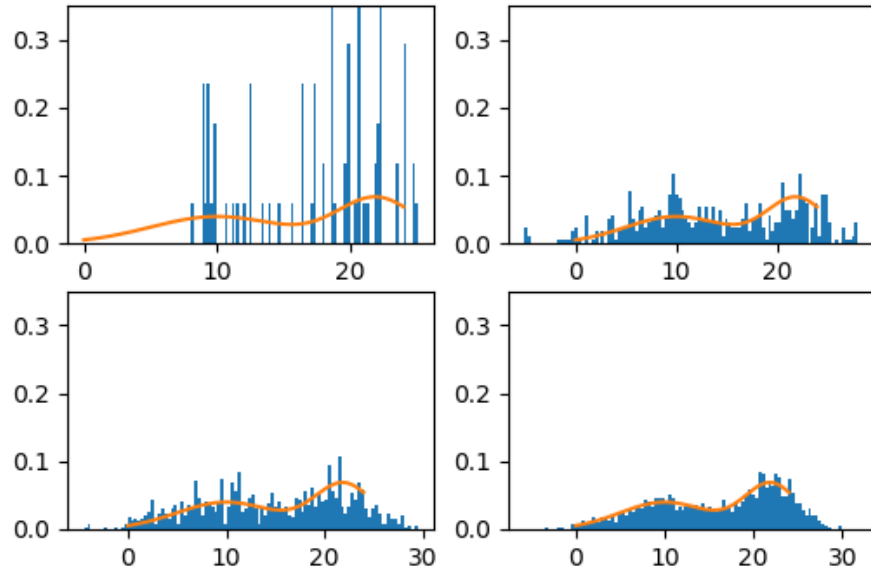


图 5：出租车服务意愿指数

可以看到，当训练样本足够多的时候，司机的送客意愿与 NARX-BPNN 模型所得结果基本一致，模型的合理性和对相关因素的依赖。

5.3 问题三的模型建立与求解

5.3.1 多目标粒子群 (MOPSO) 算法优化^[6]

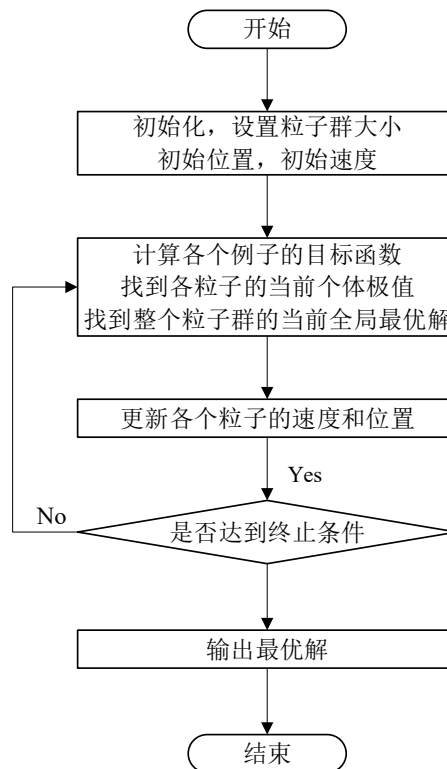


图 6：PSO 算法逻辑图

问题（三）指出，在某时刻的机场出租车候车区，经常会出现出租车排队载客和乘客排队乘车的情况。机场管理部门需要设置“上车点”，在保证车辆和乘客安全的条件下，合理安排出租车和乘客，使得总的乘车效率最高。

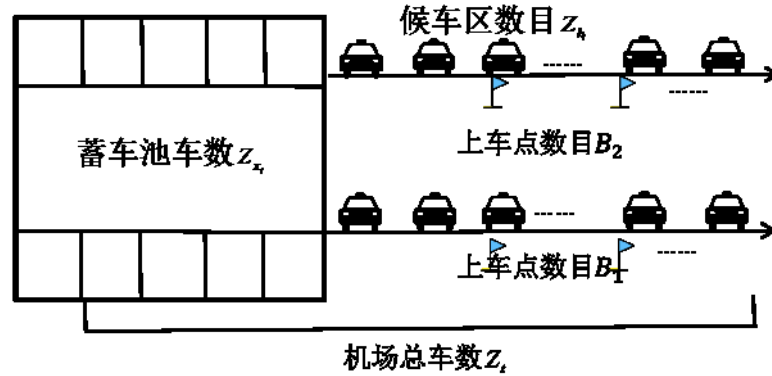


图 7：机场出租车待客区示意图

由上图可知：出租车待客区车辆总数

$$Z_t = Z_{xt} + Z_{ht} \quad (30)$$

5.3.2 决策参数的确定

首先本文在建立目标函数前，本文为保证模型真实性，在出租车的时间成本上引入了新的参数，乘客航班到达后，由机舱走出，提取行李后行走至上车点的用时，乘客滞留机场时间。由于缺少相关数据，本文利用正态分布函数进行拟合，同时假设正常情况下滞留用时最大不超过 1 小时。

$$t_c = \begin{cases} P(t_c) = e^{-t_c} + 0.00339 & \lambda = 1 \\ P(60) = 1 - \int_0^{60} P(t_c) dt_c & t_c \leq 60 \text{ min} \end{cases} \quad (31)$$

对乘客机场滞留时间的期望值进行测算

$$E(tc) = \int_0^{60} P(tc) \times tc \times dt = 20.0018 \text{ min} \approx 20 \text{ min} \quad (32)$$

随后在此基础上针对此优化问题，依据决策变量：上车点数与候车区车辆数目，约束条件：上车点数量，可改变值为设立对应目标函数：司机每小时收益最大与乘客总等待时间最短。

14 日数据：假设每辆承载人数为 2 人，因而滞留旅客数：

$$Z_{L_t} = f(X_{t-1} - Z_{t-1}) \times 2 + (X_t \times 2) \quad (33)$$

候车区车数：

$$Z_{ht} = \begin{cases} \frac{z_{L_{t-1}}}{2} & , z_{L_{t-1}} > 0 \\ R & , z_{L_{t-1}} \leq 0 \end{cases} \quad (34)$$

如果无滞留旅客，即一个上车点停留出租车，以保证乘客随来随走，如果存在前

一时刻滞留旅客，则处理前一时刻滞留。

5.3.3 目标函数的建立

在规划问题的求解上，由于这是一个动态规划问题，本文采用基于精英决策机制的多目标微粒群算法（MOPSO），求解结果上车点数与候车区车辆数目。在数据验证上，本文代入问题（二）收集到的 14 日数据，对模型进行规划求解。MOPSO 双目标优化目标函数：

$$\begin{cases} f1 = \max(Cost_{average\cdot last}) \\ f2 = \min(Customer_{last}) \end{cases} \quad (35)$$

其中出租车每小时收益函数：

$$Cost_{average\cdot last} = \frac{\int_0^{24} Cost_{average}(t) \times dt}{24} \quad (36)$$

将上式，代入出租车新效益量，出租车行驶距离 d 车速 V ，每小时驶离机场出租车数量 i （即用于得出前方出租车在等待时间），最后加入乘客出舱步行至候车区队列所用时间 t 估计量，此外考虑到车辆上车点数 B 改变速度慢于候车区车辆数 Z_{ht} ，因而最大驶离机场车辆数为二者中最小值。

$$Cost_{average}(t) = \frac{t \times Cost_{new}(t)}{\frac{d_{ai} + 32}{V_{d/n}} \int_1^{60} \int_0^{floor(\frac{60}{t_c})} \min[B, Z_{ht}] \times i \times t_c \times P(t_c) \times dt di} \quad (37)$$

乘客等待时间函数：

$$Customer_{last} = \frac{\int_0^{24} Customer(t) \times dt}{24} \quad (38)$$

代入乘客，出租车车速 V ，乘客出舱步行至候车区队列所用时间 t

$$Customer = \int_1^{60} \int_0^{floor(\frac{60}{t_c})} \min[B, Z_{ht}] \times i \times t_c \times P(t_c) \times dt di \quad (39)$$

约束条件即上车点数目 B ，两条并行车道的候车区，因而有最小上车点数目为 2，最大数模由上一个时刻滞留乘客数（滞留时间超过 60 分钟）决定，得出下式：

$$2 \leq B \leq \frac{Z_{L_{t-1}}}{6} \quad (40)$$

运用 Matlab，在问题一二主程序的基础上进行 MOPSO 优化，经过不断迭代，最后 B 的 PG(1,1) 矩阵为 (10,4)， $Z_{(h_t)} = (3218, 3218)$ 。即下文示意图结果：

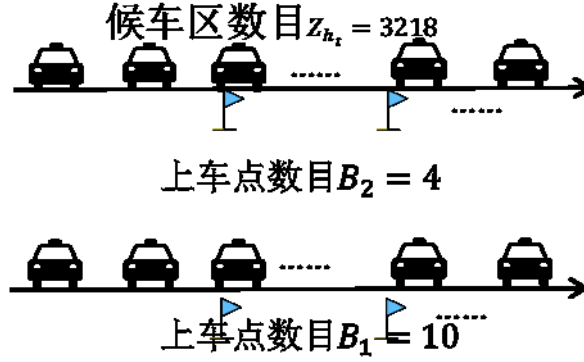


图 8：候车区试算结果示意（14 日数据）

5.4 问题四的模型建立与求解

5.4.1 基于模型一的目标函数单目标粒子群(MOPSO)算法优化^[7]

在一般情况下，出租车司机的长途载客收益远大于短途收益，因而司机在意愿上倾向于承接长途订单（订单里程数大于 25 公里）。因而问题（四）中，机场管理部门为降低拒载率，准备通过给予短途载客“优先权”，即缩减短途司机的等待时间，从而达到提高短途司机收益的目的，最终使得长短途司机的收益持平，因而“优先”安排方案，也可以看作一个优化问题。

首先在问题（三）的出租车收益函数基础上，分别设立长短途出租车收益模型。其中出租车的行驶里程分布，为了便于数据验证，这里使用问题（二）中，三点估计法所提取的三个乘客载里程，作为长短途司机平均收益预测依据。在“优先权”规则上：长途车无“优先权”，短途车“优先权”的由里程距离决定，优先权只能使车辆位置提升至“弱优先权”车辆的下一位置。从而得出约束条件：短途司机优先权次数使用次数大于 1 次小于 60 次，随后建立目标函数，长短途收益持平：

$$f = \min |Cost_L - Cost_S| \quad (41)$$

其中长途车收益函数，依据优先权规则，在原有离开机场的出租车数 C_t ，需要在分母上加入，安排在其前方的短途车数据，乘客出舱步行值上客点队列的期望值 $E(t_c)$ ，从而得出最终的长途车收益函数：

$$Cost_L(t) = C_t \times \left[P_{a2} \times \frac{m_{xn2}}{2.3 + n + C_t \times P_{a1} \times E(t_c)} + P_{a2} \times \frac{m_{xn3}}{2.4 + n + C_t \times P_{a1} \times E(t_c)} \right] \quad (42)$$

而短途车收益函数，依据优先权规则，需要在分母上加入，安排在其前方的短途等待时间，并加入使用次数 N ，从而得出最终的短途车收益函数：

$$Cost_S(t) = C_t \times P_{a1} \times \frac{m_{xn1}}{1.633 + n + \frac{17}{v} + \int_1^{60} floor \frac{60}{t_c} \times N \times P(t_c) \times dt_c} \quad (43)$$

其中被“插队后”的长途车等待时间需包含所有短途车等待时间，而短途车需要则是包含所有里程更短途的车辆等待时间，最后试算短途车需要行使几次“优先权”，即增加长途车等待时间，才能收益与长途车持平。

最后在规划问题的求解上，问题（四）同样是一个动态规划问题，本文选择采用单目标微粒群算法（PSO），求解结果即短途司机优先权次数，在数据验证上，本文代入问题（二）收集到的 14 日数据，对模型进行规划求解，验证。

运行 MOPSO 算法，得到 $f=747.6069$ ，优化的 $N=7.8345$ ，向上取整得 8，即允许短途车插队 8 次，才能使短途运输出租车（返回机场）的效益达到长途出租车（不返回机场）的程度。因为数据中长途车数量占优，而长途车的时盈利在问题一中已经证明要大于短途车，所以有可能出现优化结果没有代表性的情况。在这种情况下，调高概率 $Pa1$ 的值，调低概率 $Pa2$ 、 $Pa3$ 的值，令短途车占比增大，优化后得 $N=4.6361$ ，向上取整得 5 即允许短途车插队 5 次。

5.4.2 实操方案

Part 1：优先权信息的行使流程

由于两条出租车车道都是双车道，所以客观上可以允许超车行为发生。在符合要求的出租车进入队列时，经过 ETC 识别后，系统得到响应，无人车得到信号，移动到相应位置指引目标出租车汇入车流，并发出特殊声音提示前后车辆注意行车安全，等目标出租车汇入车流后，无人车等待下一次系统响应。画出示意图说明



图 9：车辆优先权的使用说明

Part 2：出租车与行人的干扰减少方案

考虑到车辆、乘客之间的相互干扰，建议在出租车发车时，机场工作人员可以设立指示标引导乘客上车时：区分携带大件行李的旅客与未携带大件行李的旅客。将内侧两条车道指定给携带大件行李的旅客用，外侧的两条车道指定给未携带行李的旅客使用。

Part 3：人性化方案

考虑到部分特殊乘客会对上车队列造成较大影响，因此可以另开辟为绿色通道，为行动不便的乘客服务，这些乘客可以不进入排队队伍直接登车。

6 模型的优点与改进

6.1 模型的优点

（1）数据源于 2018 年深圳市政府公开数据集，内容包含十一万辆登记运营车辆，其

中包含所有出租车数据，此外内容条目全面，记录周期 5 分钟/条。由于拥有大量数据，在问题二验证前，进行数据清洗，对经纬度的筛选条件精确至 ± 50 米，最后剩余符合条件数据达 20000 余条，因而第二问验证数据能反映真实水平。

（2）问题一中，改进后的 BPNN 不易形成局部极小值，更易得到全局最优值。BP 神经网络中极小值比较多，故很容易陷入局部极小值。经过 NARX 改进，并通过蒙特卡洛多次随机来实现后，初始权值和阈值随机性足够好，可得到全剧最优解。本文为隐含层的选取进行了大量分析与理论指导，并采用 NARX 改进联系相关性。训练时学习新样本有遗忘旧样本的趋势，本模型通过增强记忆补足此劣势。

（3）问题二中引入司机的先验模型与问题一中模型进行对比检验，证明了问题一的模型更优，并用蒙特卡洛模拟进行预测，证明了模型的合理性与优越性。

（4）问题三数据较少，本文运用模拟的方式进行数据仿真，同时利用多目标优化，在效率的考量上同时考虑司机与乘客满意度，较好的解决了数据缺乏的问题，平衡各方利益。在计算能力较低的情况下，本文通过三点估计和期望值的选取与计算，在不影响整体模型准确性的前提下，给出了令人满意的统计数据。

6.2 模型的改进方向

（1）数据时间跨度过小，仅搜集了一周时间长度的数据，天气这个影响因素的机理分析相当有限，缺乏无节假日、四季变化数据验证，使得模型在更长时间跨度的应用上具有一定的限制。

（2）此外受到数据采集地区的影响，无法计量市场竞争对出租车司机决策的影响。

7 参考文献

- [1] 深圳市政府, 深圳市营运车辆 GPS 数据,https://opendata.sz.gov.cn/data/dataSet/toDataDetails/29200_00403602#,2019-09-12.
- [2]Wunderground,Historical Weather,<https://www.wunderground.com/history>,2019-09-13.
- [3]飞常准,非常准大数据,<https://data.variflight.com/>,2019-09-13.
- [4]MathWorks, 使用 NARX 网络和时延网络进行建模和预测,<https://ww2.mathworks.cn/help/deeplearning>,2019-09-13.
- [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] 陈庆瀚, 学习粒子群最佳化 (PSO) 程式设计,<http://ccy.dd.ncu.edu.tw/~chen/resource/psd/psd.htm>,2019-09-13.
- [7]张泉峰. 首都机场接续运输协调保障技术研究实现[D].电子科技大学,2015.

8 附录

8.1 代码

PCA 代码

```
clear
gj=load('shuju.txt');
gj=zscore(gj);
r=corrcoef(gj);
[x,y,z]=pcacov(r)
f= repmat(sign(sum(x)),size(x,1),1);
x=x.*f
num=3;
df=gj*x(:,[1:num]);
tf=df*z(1:num)/100;
[stf,ind]=sort(tf,'descend');
stf=stf',ind=ind'
```

蒙特卡洛实验法 Matlab 代码 1

```
clear
dt=1/365.0;
S0=34.42;
r=0.011;
sigma=1.0;
expTerm=r*dt;
stddev=sigma*sqrt(dt);
nDays1=365;
for nDays=1:nDays1
    nTrials=10000;
    for j=1:nTrials
        n = randn(1,nDays);
        S=S0;
        for i=1:nDays
            dS = S*(expTerm+stddev*n(i));
            S=S+dS;
```



```

end
S1(nDays,j)=S;
end
end
S2=mean(S1');
plot(S2', '-o')
figure(2)
hist(S1(365,:),0:0.5:65)

```

蒙特卡洛实验法 python 代码

```

import numpy as np
import matplotlib.pyplot as plt
import math

def normal(mu,sigma,x):
    return 1/(math.pi*2)**0.5/sigma*np.exp(-(x-
mu)**2/2/sigma**2)

def eval(x):
    return normal(10,5,x) + normal(22,3,x)

def ref(x_star,x):
    return normal(x,10,x_star)

N = [100,500,1000,5000]
fig = plt.figure()
for i in range(4):
    X = np.array([])
    x = 7 #initialize x0 to be 0.1
    for j in range(N[i]):
        u = np.random.rand()
        x_star = np.random.normal(x,10)
        A = min(1,eval(x_star)/eval(x))
        if u < A:
            x = x_star
    X=np.hstack((X,x))

```

```

ax = fig.add_subplot(2,2,i+1)
ax.hist(X,bins=100,density=True)
x = np.linspace(0,24,5000)

ax.plot(x,eval(x)/2)
ax.set_ylim(0,0.35)
ax.text(0,2.5,'I=%d'%N[i])

fig.suptitle('Taxi service willingness index')

plt.savefig('Taxi service willingness index.png',dpi=100)

plt.show()

```

基于 NARX 与蒙特卡洛模拟的 BP 神经网络模型代码

```

close all
clear
echo on
clc
N=1000;
DATA=xlsread('DATAALLgai.xlsx',1,'B2:L169');
J=floor(DATA(:,1)*0.1);
C=floor(DATA(:,2)*0.135);
Z=zeros(168,1);
Z(1,1)=J(1)-C(1)
for i=1:167
Z((i+1),1)=J(i)-C(i)+Z(i,1);
end
X= (floor(DATA(:,6))*4);
parameter=DATA(:,7:11);
P=[J,C,Z,X,parameter];
T=[J,C,Z,X];

clc

Ptrain=P(1:143,:);

```

```

Ptrain=Ptrain';
Ttrain=T(2:144,:);
Ttrain=Ttrain';
clc
for i=1:N
    net=newff(Ptrain,Ttrain,[12],{'tansig','purelin'},'t
raingdm')
    net.trainParam.lr = 0.05;
    net.trainParam.mc = 0.9;
    net.trainParam.epochs = 1000;
    net.trainParam.goal = 1e-7;
    [net,tr]=train(net,Ptrain,Ttrain);
    Ptest=P(144:167,:);
    Ptest=Ptest';
    Ttest=T(145:168,:);
    Ttest=Ttest';
    Tpred = sim(net,Ptest) ;
    for j=1:4
        for z=1:24
            if Tpred(j,z)<0
                Tpred(j,z)=0;
            end
        end
    end
end

Jpred(i,:)=floor(Tpred(1,:));
Cpred(i,:)=floor(Tpred(2,:));
Xpred(i,:)=floor(Tpred(4,:));
Zpred(i,:)=floor(Tpred(3,:));
E(((4*(i-1))+1):((4*(i-1))+4),:) = abs(Tpred -
Ttest);
clc
echo off
end
dal=17;

```

```

pa1=0.2069;
da2=37;
pa2=0.6294;
da3=40;
pa3=0.1637;
mxd1=49.0907;
mxd2=110.5573;
mxd3=120.7133;
mxn1=60.768;
mxn2=143.616;
mxn3=156.81;

for i=1:18
    xxx=Z((149+i))-X((149+i));
    n=0;
while xxx>0
    n=n+1;
    xxx=xxx-(X(149+i)*n);
    if (i+n)>17
        Cost(i)=34.32;
        break
    end
end
    if ((Z(149+i))/(X(149+i)))>5
        choose(i)=-100;
        Cost(i)=34.32;
    else

choose(i)=pa1*(mxd1/(1.633+n))+pa2*(mxd2/(2.3+n))+pa3*(m
xd3/(2.4+n))-34.32;

Cost(i)=pa1*(mxd1/(1.633+n))+pa2*(mxd2/(2.3+n))+pa3*(mxd
3/(2.4+n));
        end
    end
for i=1:6

```

```

        xxx=Z((144+i))-X((144+i));
        n=0;
    while xxx>0
        n=n+1;
        xxx=xxx-(X(144+i)*n);
        if (i+n)>5
            Cost(i+18)=44.616;
            break
        end
    end
    if ((Z(144+i))/(X(144+i)))>5
        choose(i+18)=-100;
        Cost(i+18)=44.616;
    else

choose(i+18)=pa1*(mxn1/(1.633+n))+pa2*(mxn2/(2.3+n))+pa3*
*(mxn3/(2.4+n))-44.616;

Cost(i+18)=pa1*(mxn1/(1.633+n))+pa2*(mxn2/(2.3+n))+pa3*(
mxn3/(2.4+n));
        end

    end

    Costaverage=sum(Cost)/24;

    for j=1:N
        for i=1:18
            xxx=Z((149+i))-X((149+i));
            n=0;
        while xxx>0
            n=n+1;
            xxx=xxx-(Xpred(j,(4+i))*n);
            if (i+n)>17
                Costnew(i,j)=34.32;
                break
            end
        end
    end
end

```

```

choosenew(i,j)=pa1*(mxd1/(1.633+n))+pa2*(mxd2/(2.3+n))+p
a3*(mxd3/(2.4+n))-34.32;

```

```

    if choosenew(i,j)>0

```

```

Costnew(i,j)=pa1*(mxd1/(1.633+n))+pa2*(mxd2/(2.3+n))+pa3
*(mxd3/(2.4+n));

```

```

    else

```

```

        Costnew(i,j)=34.32;

```

```

    end

```

```

end

```

```

for i=1:6

```

```

    xxx=Z((144+i))-X((144+i));

```

```

    n=0;

```

```

while xxx>0

```

```

    n=n+1;

```

```

    xxx=xxx-(Xpred(j,i)*n);

```

```

    if (i+n)>5

```

```

        Costnew(i+18,j)=44.616;

```

```

        break

```

```

    end

```

```

end

```

```

choosenew(i+18,j)=pa1*(mxn1/(1.633+n))+pa2*(mxn2/(2.3+n)
)+pa3*(mxn3/(2.4+n))-44.616;

```

```

    if choosenew(i+18,j)>0

```

```

Costnew(i+18,j)=pa1*(mxn1/(1.633+n))+pa2*(mxn2/(2.3+n))+
pa3*(mxn3/(2.4+n));

```

```

    else

```

```

        Costnew(i+18,j)=44.616;

```

```

    end

```

```

end

```

```

Costaveragenew(j)=sum(Costnew(:,j))/24;

```

```

end
Costaveragelast=sum(Costaveragenew)/N;
MSE=mse(E);

```

MOPSO 优化代码

```

clc
N=2;
Iteration=2;
c1=2;
c2=2;
M=2;
w_max=0.9;
w_min=0.4;
D=1;
PL1=zeros(N,1);
PL2=zeros(N,1);
PL=[PL1,PL2];
f1=zeros(2*N,1);
f2=zeros(2*N,1);
f=[f1,f2];

Zh=zeros(N,24);
R=zeros(N,1);

a=zeros(24,1);
b=[(X(144:167)-Z(144:167)),a];
ZL(1:24,1)=2*(X(145:168)+max(b,[],2));

for i=1:59
tc(i)=i;
Ptc(i)=exp((-1)*i)+0.00339;
end
tc(60)=60;
Ptc(60)=1-sum(Ptc(1:59));

for i=1:N

R(i,1)=floor((rand(1,1).*(12-2))/2)*2+2;

```

```

        if ZL(1,1)>0
Zh(i,1)=ZL(1,1)/2;
        else
            Zh(i,1)=R(i,1);
        end
        c=[R(i,1),Zh(i,1)];
        MIN(i)=min(c);

end

x_font=zeros(N,1);
x_P=[R, x_font];
v1=rand(N,1).*2.5;
v=[v1];
y=zeros(N,1);

        for i=1:N

for j=1:24
    customer(i,j)=0;
    for k=1:60
        for l=1:(floor(60/k))
            customer(i,j)=customer(i,j)+k*Ptc(k)*l*MIN(i)
        end
    end
end

costaverage(i,j)=C(144+j)*Costnew(j,1)/((49*pa1+69*pa2+7
2*pa3)/35+customer(i,j)/60)
end

PL1(i)=- (sum(sum(costaverage(i,:)))/24);

PL2(i)=sum(sum(customer(i,:)))/24;

        end

        PL=[PL1,PL2];
        pb=x_P;
        n=find(PL1==min(PL1));
        pg=x_P(n,:);

```



```

f1=[PL1;PL1];
f2=[PL2;PL2];

for t=1:Iteration
    x_Q=x_P;
    for i=1:N
        w=w_max-(t-1)*(w_max-
w_min)/max((Iteration-1),1);
        v(i,:)=w*v(i,:)+c1*rand*(pb(i,1:D)-
x_Q(i,1:D))+c2*rand*(pg(1:D)-x_Q(i,1:D));

        if v(i,1)<=-2.5
            v(i,1)=-2.5;
        end
        if v(i,1)>=2.5
            v(i,1)=2.5;
        end

        x_Q(i,1:D)=x_Q(i,1:D)+v(i,1:D);

        if x_Q(i,1)<=2
            x_Q(i,1)=2;
        end
        if x_Q(i,1)>=12
            x_Q(i,1)=12;
        end
        x_Q(i,1)=floor(x_Q(i,1)/2)*2;

        x=[x_Q;x_P];

        k=i+N;

f1(k)=f1(i);

f2(k)=f2(i);

K=i;

    if ZL(1,1)>0
        Zh(K,1)=ZL(1,1)/2;
    end
end

```

```

        else
            Zh(K,1)=x(K,1);
        end
        c=[x(K,1),Zh(K,1)];
        MIN(K)=min(c);

for j=1:24
    customer(i,j)=0;
    for k=1:60
        for l=1:(floor(60/k))
            customer(i,j)=customer(i,j)+k*Ptc(k)*l*MIN(K)
        end
    end
end

costaverage(i,j)=C(144+j)*Costnew(j,1)/((49*pa1+69*pa2+7
2*pa3)/35+customer(i,j)/60)
end

f1(K)=-(sum(sum(costaverage(i,:)))/24);

f2(K)=sum(sum(customer(i,:)))/24;


        f=[f1,f2];


    end

    front = 1;
    F(front).f = [];
    individual = [];
    x(:,D+1)=0;
    for i = 1 : 2*N

        individual(i).n = 0;

        individual(i).p = [];
        for j = 1 : 2*N

```

```

        dom_less = 0;
        dom_equal = 0;
        dom_more = 0;
        for k = 1 : M
            if (f(i,k) < f(j,k))
                dom_less = dom_less + 1;
            elseif (f(i,k) == f(j,k))
                dom_equal = dom_equal + 1;
            else
                dom_more = dom_more + 1;
            end
        end
        if dom_less == 0 && dom_equal ~= M
            individual(i).n = individual(i).n
+ 1;

            elseif dom_more == 0 && dom_equal ~= M
                individual(i).p = [individual(i).p
j];
        end
    end
    if individual(i).n == 0
        x(i,D+1) = 1;
        F(front).f = [F(front).f i];
    end
end
while ~isempty(F(front).f)
    Q = [];
    for i = 1 : length(F(front).f)
        if ~isempty(individual(F(front).f(i)).p)
            for j = 1 :
length(individual(F(front).f(i)).p)

individual(individual(F(front).f(i)).p(j)).n = ...

individual(individual(F(front).f(i)).p(j)).n - 1;
                if
individual(individual(F(front).f(i)).p(j)).n == 0

x(individual(F(front).f(i)).p(j),D+1) = ...
                    front + 1;

                Q = [Q
individual(F(front).f(i)).p(j)];

```

```

                                end
                            end
                        end
                    end
                front = front + 1;
                F(front).f = Q;
            end
            [temp,index_of_fronts] = sort(x(:,D+1));
            for i = 1 : length(index_of_fronts)
                sorted_based_on_front(i,:) =
x(index_of_fronts(i),:);
            end

            current_index = 0;

            for i=1:N
                x_P(i,:)=sorted_based_on_front(i,:);
            end
            pg=sorted_based_on_front(1,:);
            for i=1:N

                if ZL(1,1)>0
                    Zh(i,1)=ZL(1,1)/2;
                else
                    Zh(i,1)=x_P(i,1);
                end
                c=[x_P(i,1),Zh(i,1)];
                MIN(i)=min(c);

            for j=1:24
                customer(i,j)=0;
                for k=1:60
                    for l=1:(floor(60/k))
                        customer(i,j)=customer(i,j)+k*Ptc(k)*l*MIN(i)
                    end
                end
            end

            costaverage(i,j)=C(144+j)*Costnew(j,1)/((49*pa1+69*pa2+7
2*pa3)/35+customer(i,j)/60)
        end

```

```
f1(i)=- (sum(sum(costaverage(i,:)))/24);
```

```
f2(i)=sum(sum(customer(i,:)))/24;
```

```
end
```

```
f=[f1,f2];
```

```
for i=1:N
```

```
    if f1(i)<=PL1(i) | f2(i)<=PL2(i)
```

```
        if f1(i)<=PL1(i)
```

```
            PL1(i)=f1(i) ;
```

```
        end
```

```
        if f2(i)<=PL2(i)
```

```
            PL2(i)=f2(i) ;
```

```
        end
```

```
        pb(i,:)=x_P(i,:);
```

```
    end
```

```
end
```

```
end
```

```
n=find(PL1==min(PL1));
```

```
pg=x_P(n,:);
```

```
if ZL(1,1)>0
```

```
Zh(n(1),1)=ZL(1,1)/2;
```

```
else
```

```
    Zh(n(1),1)=pg(n(1),1);
```

```
end
```

```
d=[pg(1,1),Zh(n(1),1)];
```

```
MIN(1)=min(d);
```

```
for j=1:24
```

```
    customer(n,j)=0;
```

```
    for k=1:60
```

```
        for l=1:(floor(60/k))
```

```
            customer(n,j)=customer(n,j)+k*Ptc(k)*l*MIN(1)
```

```
        end
```

```
    end
```

```

costaverage(n(1),j)=C(144+j)*Costnew(j,1)/((49*pa1+69*pa
2+72*pa3)/35+customer(n,j)/60)
end

```

```

COSTaverage=abs(-(sum(sum(costaverage(n(1),:)))/24));

```

```

Customer=sum(sum(customer(n,:)))/24;

```

PSO 优化代码

```

clc
N=8;
Iteration=8;
c1=2;
c2=2;
M=1;
w_max=0.9;
w_min=0.4;
D=1;
PL1=zeros(N,1);
PL=[PL1];
f1=zeros(2*N,1);
f=[f1];

for i=1:N

    NNN(i,1)=floor(rand(1,1).*59)+1;

end

x_font=zeros(N,1);
x_P=[NNN, x_font];
v1=rand(N,1).*12;
v=[v1];
y=zeros(N,1);

```

```

        for i=1:N
for j=1:24
LONG(j)=C(144+j)*(pa2*mxn2/(2.3+pa1/3)+pa3*mxn3/(2.4+pa1/3));
for o=1:60
ZZZ=o*NNN(i,1)*Ptc(o);
end
SHORT(j)=C(144+j)*pa1*mxn1/(1.633+17/35+ZZZ/60)
end
PL1(i)=abs(sum(LONG)-sum(SHORT))/24;

        end

        PL=[PL1];
        pb=x_P;
        n=find(PL1==min(PL1));
        pg=x_P(n,:);

        f1=[PL1];
        f2=[PL2];

        for t=1:Iteration
            x_Q=x_P;
            for i=1:N
                w=w_max-(t-1)*(w_max-w_min)/max((Iteration-1),1);
                v(i,:)=w*v(i,:)+c1*rand*(pb(i,1:D)-x_Q(i,1:D))+c2*rand*(pg(1:D)-x_Q(i,1:D));

                if v(i,1)<=-12
                    v(i,1)=-12;
                end
                if v(i,1)>=12
                    v(i,1)=12;
                end

                x_Q(i,1:D)=x_Q(i,1:D)+v(i,1:D);
            end
        end
    end
end

```

```

        if x_Q(i,1)<=2
            x_Q(i,1)=2;
        end
        if x_Q(i,1)>=12
            x_Q(i,1)=12;
        end
        x_Q(i,1)=floor(x_Q(i,1));

        x=[x_Q;x_P];

        k=i+N;

f1(k)=f1(i);

f2(k)=f2(i);

K=i;

for j=1:24
    LONG(j)=C(144+j)*(pa2*mxn2/(2.3+pa1/3)+pa3*mxn3/(2.4+pa1/3));
    for o=1:60
        ZZZ=o*x(K,1)*Ptc(o);
    end
    SHORT(j)=C(144+j)*pa1*mxn1/(1.633+17/35+ZZZ/60)
end
f1(K)=abs(sum(LONG)-sum(SHORT))/24;
    f=[f1];
end

front = 1;
F(front).f = [];
individual = [];
x(:,D+1)=0;
for i = 1 : 2*N

    individual(i).n = 0;

    individual(i).p = [];
    for j = 1 : 2*N
        dom_less = 0;
        dom_equal = 0;
    end
end

```



```

        dom_more = 0;
    for k = 1 : M
        if (f(i,k) < f(j,k))
            dom_less = dom_less + 1;
        elseif (f(i,k) == f(j,k))
            dom_equal = dom_equal + 1;
        else
            dom_more = dom_more + 1;
        end
    end
    if dom_less == 0 && dom_equal ~= M
        individual(i).n = individual(i).n
+ 1;

        elseif dom_more == 0 && dom_equal ~= M
            individual(i).p = [individual(i).p
j];

        end
    end
    if individual(i).n == 0
        x(i,D+1) = 1;
        F(front).f = [F(front).f i];
    end
end
while ~isempty(F(front).f)
    Q = [];
    for i = 1 : length(F(front).f)
        if ~isempty(individual(F(front).f(i)).p)
            for j = 1 :
length(individual(F(front).f(i)).p)

individual(individual(F(front).f(i)).p(j)).n = ...

individual(individual(F(front).f(i)).p(j)).n - 1;
                if
individual(individual(F(front).f(i)).p(j)).n == 0

x(individual(F(front).f(i)).p(j),D+1) = ...
                    front + 1;

                Q = [Q
individual(F(front).f(i)).p(j)];
            end
        end
    end
end

```

```

        end
    end
    front = front + 1;
    F(front).f = Q;
end
[temp,index_of_fronts] = sort(x(:,D+1));
for i = 1 : length(index_of_fronts)
    sorted_based_on_front(i,:) =
x(index_of_fronts(i),:);
end

current_index = 0;

for i=1:N
    x_P(i,:)=sorted_based_on_front(i,:);
end
pg=sorted_based_on_front(1,:);
for i=1:N

    for j=1:24
LONG(j)=C(144+j)*(pa2*mxn2/(2.3+pa1/3)+pa3*mxn3/(2.4+pa1
/3));
for o=1:60
ZZZ=o*x_P(i,1)*Ptc(o);
end
SHORT(j)=C(144+j)*pa1*mxn1/(1.633+17/35+ZZZ/60)
end
f1(i)=abs(sum(LONG)-sum(SHORT))/24;

        end

f=[f1];
for i=1:N
    if f1(i)<=PL1(i)
        if f1(i)<=PL1(i)
            PL1(i)=f1(i) ;
        end
        pb(i,:)=x_P(i,:);
    end
end

```

```

end

end

n=find(PL1==min(PL1));
pg=x_P(n,:);

for j=1:24
LONG(j)=C(144+j)*(pa2*mxn2/(2.3+pa1/3)+pa3*mxn3/(2.4+pa1/3));
for o=1:60
ZZZ=o*pg(1,1)*Ptc(o);
end
SHORT(j)=C(144+j)*pa1*mxn1/(1.633+17/35+ZZZ/60);
end
FFF=abs(sum(LONG)-sum(SHORT))/24;

```

8.2 预处理后的数据

Time1008-1014	出租车进机场数	出租车出机场数	滞留机场出租车数目	机场出租车需求乘客	平均需求人数/台数	机场需求出租车台数	温度/°C	湿度	风速/mph	压力/in. w	天气 (晴多云1,
0-8											
1	1489	794	41	547	3	183	24.44	0.89	7	29.9	1
2	2101	518	222	403	3	135	24.44	0.89	7	29.9	1
3	371	154	238	115	3	39	28.89	0.62	7	29.9	1
4	261	239	231	144	3	48	28.89	0.62	7	29.9	1
5	333	468	201	202	3	68	28.89	0.62	7	29.9	1
6	204	369	171	749	3	251	28.89	0.63	11	29.8	1
7	434	195	188	461	3	155	28.89	0.63	11	29.8	1
8	494	155	216	403	3	135	28.89	0.63	11	29.8	1
9	678	125	266	864	3	288	27.78	0.69	9	29.8	1
10	672	1865	81	58	3	20	27.78	0.69	9	29.8	1
11	552	293	96	662	3	222	27.78	0.69	9	29.8	1
12	341	126	113	1555	3	519	26.11	0.71	4	29.8	1
13	96	49	115	691	3	231	26.11	0.71	4	29.8	1
14	171	76	121	490	3	164	26.11	0.71	4	29.8	1
15	327	306	112	1354	3	452	24.44	0.81	2	29.8	1
16	76	130	102	115	3	39	24.44	0.81	2	29.8	1
17	114	166	90	2333	3	779	24.44	0.81	2	29.8	1
18	94	89	87	1728	3	576	23.33	0.84	2	29.8	1
19	278	195	88	1210	3	404	23.33	0.84	2	29.8	1
20	519	174	116	2822	3	942	23.33	0.84	2	29.8	1
21	199	87	124	1238	3	414	23.33	0.86	2	29.8	1
22	739	256	163	2218	3	740	23.33	0.86	2	29.8	1
23	800	261	207	1181	3	395	23.33	0.86	2	29.8	1
0-9											
1	922	838	197	432	1	432	24.44	0.86	2	29.8	1
2	1327	297	289	461	1	461	24.44	0.86	2	29.8	1
3	306	119	303	29	1	29	28.89	0.65	2	29.8	1
4	161	310	277	202	1	202	28.89	0.65	2	29.8	1
5	447	237	289	288	1	288	28.89	0.65	2	29.8	1
6	339	667	232	634	1	634	28.89	0.67	4	29.8	1
7	434	397	221	202	1	202	28.89	0.67	4	29.8	1
8	770	351	250	58	1	58	28.89	0.67	4	29.8	1
9	792	263	293	288	1	288	27.78	0.74	7	29.8	1
10	948	1934	126	1210	1	1210	27.78	0.74	7	29.8	1
11	1054	524	160	691	1	691	27.78	0.74	7	29.8	1
12	839	343	197	1526	1	1526	25.56	0.76	9	29.8	1
13	688	465	203	2419	1	2419	25.56	0.76	9	29.8	1
14	1116	484	249	950	1	950	25.56	0.76	9	29.8	1
15	879	712	240	403	1	403	24.44	0.83	4	29.8	1
16	511	356	243	518	1	518	24.44	0.83	4	29.8	1
17	263	337	223	605	1	605	24.44	0.83	4	29.8	1
18	476	172	247	1037	1	1037	23.33	0.9	2	29.8	2
19	658	482	247	1958	1	1958	23.33	0.9	2	29.8	2
20	739	261	285	950	1	950	23.33	0.9	2	29.8	2
21	445	219	299	547	1	547	23.33	0.88	7	29.8	2
22	719	455	309	1958	1	1958	23.33	0.88	7	29.8	2
23	766	359	337	691	1	691	23.33	0.88	7	29.8	2
0-10											
1	1031	360	391	576	1	576	24.44	0.88	2	29.8	2
2	543	346	398	346	1	346	24.44	0.88	2	29.8	2
2	305	214	399	720	1	720	24.44	0.88	2	29.8	2