



数学建模培训课程论文

优秀论文改写再现课程报告

班级：21信息与计算科学3班

姓名：李宁

学号：202111080068

序号：67

大组别：i

小组成员：李宁、田博松、齐昊坤

2023年 8 月

**基于系统模拟的机场出租车决策与安排模型**

**摘要**

随着世界经济一体化的不断发展，我国航空业发展迅速，机场的旅客吞吐量不断攀升，旅客离港时出租车是机场重要的散客渠道之一。 由于不同时空，不同地域机场规模不同，如何解决离港出租车排队问题的研究具有深远意义。本文针对某特定机场，从司机，机场两个角度分别对该问题进行了探究。

针对问题一，首先,我们综合分析了影响司机决策的确定因素与随机因素，并研究了这些因素对司机决策的作用机理。接着,根据作用机理，得到出租车两种方案下的收益计算公式，我们以收益最大化为决策准则，建立出租车司机选择决策模型。最后，我们将模拟数据代入**决策模型**，在给定条件下模拟司机的选择策略，将模拟结果与生活实际进行对比分析，说明了我们模型的可行性。

针对问题二，需要我们合理安排上车点数量，使得机场乘车效率最高。于是，我们以乘车效率为优化目标，安全因素为约束条件，上车点数量为决策变量，建立**单目标优化模型**。其中，我们通过合理制定机场出租车乘车区运行规则，利用计算机模拟的方法，计算得到各方案对应的乘车效率。在人和车数量相当时，在保证安全性和方便管理的前提下，上车点数量设为8个时效率较高，乘客和出租车平均用时都在4分钟左右。

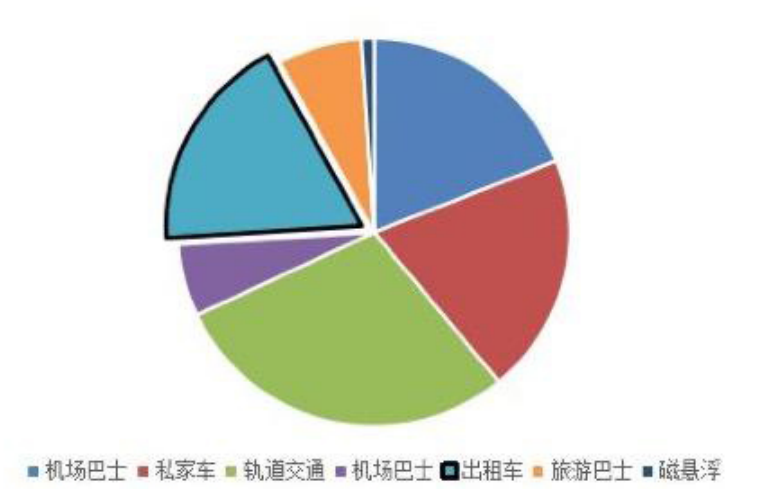
针对问题三，我们将机场排队的出租车分为长途车和短途车这两大类，其中短途车数量远小于长途车数量。接着，我们分别研究了长途车与短途车在机场与市区间的往返情况,对比两者在相同观测时间内的收益与相同观测空间内的空载率。然后，以长短途出租车的空载率差值、收益差值最小为优化目标，建立排队位置**最优化模型**，在输入短途里程后可得到相应的优先排队位置，同时发现，短途里程越短，车辆排的位置越靠前。

针对问题四，我们分别给出了基于时间和基于路程的两种“短途票”方案，满足时间或路程条件的司机能够凭短途票享有“优先权”。两种衡量标准都围绕**收益平衡思想**设定，取短途司机利润与正常司机利润相等时对应额临界值。两种方法各有利弊，适用于不同条件的机场。

**关键词：决策模型 单目标优化模型 最优化模型 收益平衡思想**

**一、问题重述**

乘坐航班到达机场后，多数乘客会前往市区的目的地。出租车因其无需转乘，方便快捷，全时段服务等优势条件深受出港旅客青睐。下图是上海浦东机场2018年旅客集散方式示意图。



其中，乘坐出租车的乘客数量达到了总旅客数的18%。可以看出，出租车是机场主要的集散方式之一。而国内多数机场采用进站客流与出站客流分离的规划思想，即送客与接客区域相隔离。于是送客到机场的出租车司机都将会面临两个选择:

(A)前往到达区排队等待载客返回市区。出租车必须到指定的“蓄车池”排队等候，依“先来后到”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，需要付出一定的时间成本,如图2。



以浦东机场为例，每位能在航站楼成功载乘客离去的出租车司机，都必须先经过P7蓄车场几小时的耐心排队。经过漫长等待后，被随机分配开往P1或P2停车场缓冲区再进行半小时到一小时的等待，最后才能在候客区载到客人。

“整个等待若花4小时算幸运的，有时候要等候七八个小时。”--某出租车司机。

(B)直接放空返回市区拉客。出租车司机会付出空载费用和可能损失潜在的载客收益。在浦东机场，漫长的等待让很多的哥望而却步，他们选择直接返回市区

“等这么久真的很不划算，不如回去拉客能多赚点。” 一一某出租车司机。

通过查询，司机可以得到在该时间段抵达的航班数量，同时也可以观测到“蓄车池”里已有的车辆数，结合两者司机可以在(A)、(B)间作出决策。司机的决策通常与其个人的经验判断有关，比如在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。

离港乘客若要搭乘出租车，则须前往指定“乘车区”排队，按先后顺序乘车。机场出租车管理人员负责放行出租车进入候客区以及安排乘客上车。要求结合实际，建立数学模型研究下列问题:

(1)讨论决策相关因素及其影响机理，结合考虑机场乘客数量变化和出租车司机收益，建立出租车司机的选择决策模型，给出选择策略。

(2)结合国内某一机场及其所在城市出租车的相关数据，给出出租车司机的选择方案，并分析模型的合理性和对相关因素的依赖性。

(3)某机场“乘车区”现有两条并行车道。求应如何设置“上车点”，并合理安排出租车和乘客，在保证车辆和乘客安全的条件下，使得总的乘车效率最高。

(4)出租车载客收益与载客的行驶里程有关，乘客的目的地有远有近，出租车司机不能选择乘客和拒载，但允许出租车多次往返载客。故，拟对某些短途载客再次返回的出租车给予“优先权”，使出租车的收益均衡，试给出“优先”安排方案。

**二、问题分析**

本文要解决的是飞机场离港出租车上客系统优化问题。问题一、二要求从司机角度，结合机场情况作出空载返回或是排队等候的决策，使得利润最大化。而第三、第四问要求从机场角度考虑如何安排上车点，提高乘车效率以及如何设置短途返回“优先机制”以能够均衡司机收益。

**2.1问题一分析**

首先从出租车司机的角度出发，建立司机载客决策模型，并给出司机选择策略。为解决司机到达机场后是进入“蓄车池”等待载客还是直接空载返回市区的选择问题，我们需要考虑影响司机做出决策的因素并研究因素对司机决策过程的作用机理。最后基于收益，为出租车司机做出更有益于自己的决策。我们需要通过计算机对上述司机决策模型进行模拟，确定的信息由我们根据实际情况直接给出，不确定的信息基于特殊概率分布给出，从而得到不同情况下司机的选择策略，也展现出模拟的可行性，为第二问的求解奠定基础。

**2.2问题二分析**

在第一问建立的决策模型基础上，结合实际的机场及出租车情况，给出出租车司机决策方案并讨论模型的合理性及对因素的依赖性。尽量避免大型机场:大型机场作为国际、国内中转枢纽，交通方式多样，各类影响较多。在不同时间段需要纳入考虑因素太多，会对五个根本因素与的决策关系产生影响。为保证模拟结果的合理性,我们需要采用多次模拟得到出现次数更多的一种方案作为自己最终的决策结果。

**2.3问题三分析**

在保证乘客和车辆安全的基础上设置出租车接客区的上车点并使得总乘车效率最高。这实质上是一个优化问题，需要我们以上车点数量为决策变量，在满足乘客和车辆安全的约束条件下，使乘车效率尽可能高。在计算每种上车点设置方案下的乘车效率时，我们需要根据实际情况，合理制定机场出租车载客的运作规则，并利用计算机多次模拟统计得到出租车和乘客总乘车效率。

**2.4问题四分析**

由于规定出租车司机不能选择乘客和拒载，所以存在出租车乘载目的地距机场较近的乘客。在这种情况下，必然造成出租车司机的收益损失。浦东国际机场为了解决短途旅客与司机引发的矛盾，引入GPS短途职能识别系统，对出租车进行长短途分流，设置两个缓冲区与蓄车场，短途司机直接进入第二个缓冲区进行载客，而其他司机依次排队进入两个缓冲区进行载客。

**三、模型假设**

1、乘客上车时间为90秒【1】。

2、决策的选择仅考虑经济因素，忽略个人倾向等其他因素。

3、出租车行驶默认为相同速度的匀速运动

**四、符号说明**

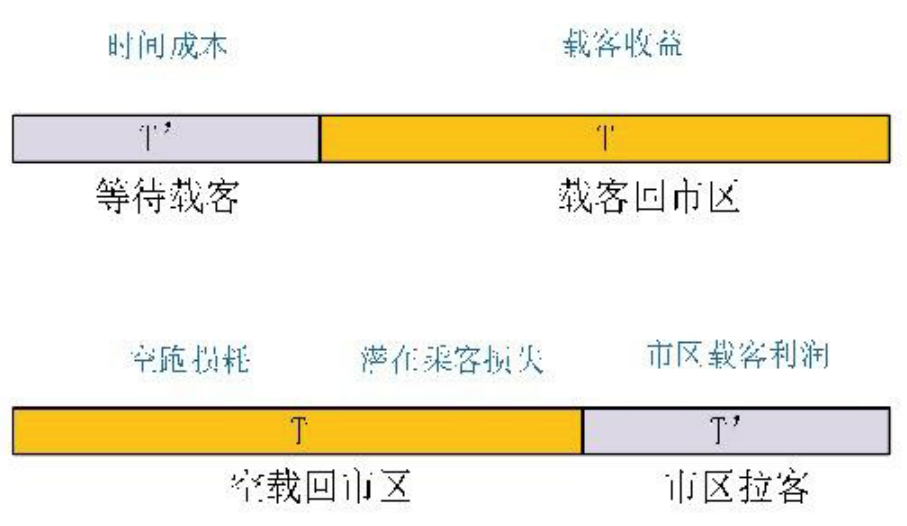
|  |  |  |
| --- | --- | --- |
| 符号 | 含义 | 单位 |
| n | 乘坐出租车离港的客流量 | 人 |
| m | 机场出租车流量 | 辆 |
| T | 等待载客时间 | h |
| T’ | 从机场返回市中心时间 | h |
| x | 从机场返回市中心路程 | km |
| τ | 乘客上车时间 | s |
| X | 每条停车带所能停放的车辆数 | 辆 |
| W1 | 载客返回市中心收益 | 元 |
| W2 | 在市中心拉客收益 | 元 |
| V1 | 载客返回市中心油耗 | 元 |
| V2 | 空载返回市中心油耗 | 元 |
| P1 | 排队等候决策下的利润 | 元 |
| P2 | 空载返回决策下的利润 | 元 |
| δ | 市中心拉客的每小时收益 | 元/h |
| π1 | 时间成本系数 |  |
| π2 | 潜在收益系数 |  |
| φ | 乘车效率 | 辆/h |
| v | 车流速度 | km/h |
| L | 接客区长度 | m |
| D | 出租车长度 | m |
| S | 两辆出租车之间保持的安全距离 | m |

**五、模型的建立与求解**

**5.1问题一模型的建立**

**5.1.1决策影响因素机理分析**

分别对(A)排队等待载客，(B) 空载返回市区的收益进行分析。为公平起见，取时间段相同，默认空载返回后便开始在市区拉客。两种方法各时段的成本及收益如下图所示:



**5.1.2决策模型的建立**

确定目标约束条件，讨论时间成本问题。

时间成本【2】，即“货币时间价值"，其定义为:时间成本是指一定量资金在不同时点上的价值量产差额。时间成本也可以引申为:在等待时间内造成的市场机会的丢失。对于本题中出租车司机而言，无论作出等候载客或是空载返回的决策，均会产生相应的时间成本。

对采取等候载客的司机而言，其在蓄车池中等候的时间产生时间成本:假设不需等候而直接载客返回市区，省下的等候时间可以在市区拉客获得收益。该丢失的潜在盈利收益即为折算为金钱的时间成本。

综上可建立决策模型为

max Z=Q∙P1+(1-Q)∙P2

P1=W1-V1-π1TP2=W2-V2-π2TW1=f(x)W2=τTT=g[mt]

**5.2问题一模型的求解**

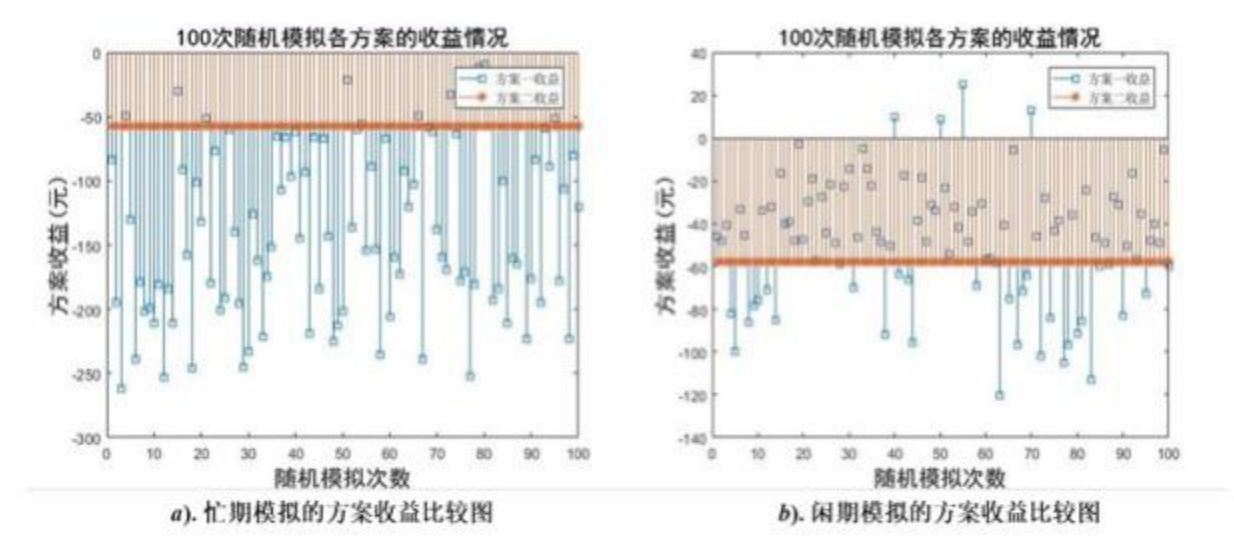
为给出司机的选择策略，我们根据上述建立的模型，通过下面的流程去计算各方案收益，并得到判断结果。

首先，我们需要输入司机能够观测和已知的一些数据，具体的有司机抵达的时间，因为不同时间(夜间、白天)出租车的收费方式不同，在抵达时间前后一个小时内的航班数量以及“蓄车池”里已有的车辆数。

然后设置非确定因素(随机因素):司机根据个人经验估计每个航班的载客数服从[M1, M2]上的均匀分布选取，这里我们不妨取M1=100，M2 =200。乘客乘车里程服从概率分布，里程最大值是从机场到市中心距离为25.1km。利用公式，计算两种策略的计价收入、等待时间、时间成本和油耗。

下面我们采用一组生成模拟数据，来验证我们建立模型的可行性。

假设司机到达机场的时间点是上午10点，他观测到9点至10点航班数为4，10点至11点航班数为6。在“蓄车池”排队等待的出租车共有50辆，将求解中确定的数据代入判断模型，得到的结果如图所示。



我们发现，当航班数减少，同时已经排队的出租车增多后，绝大部分的方案一收益都低于方案二的收益，所以司机很大概率会选择方案二空载返回市区，而不会在机场等待较长载客。以上结果均符合实际情况，在一定程度上体现我们模型的可行性。

**5.3问题二模型的建立**

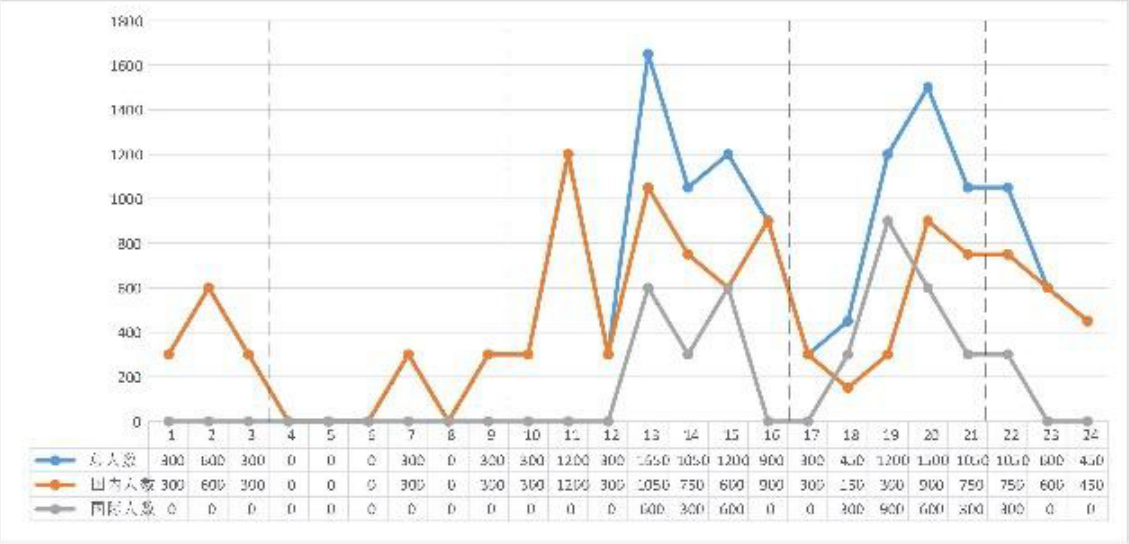
考虑到小型机场数据收集困难且样本容量过小,大型机场数据量过于庞大，不便于统计分析。为兼顾数据的可靠性以及处理的便利性，决定选择中型机场进行分析。并保证机场所在城市较为发达，以保证足够的出租车流量。如此才能有足够大的样本空间，较为客观地评价第一问的模型合理性。

综上采用了苏南硕国际机场进行分析处理，得到相关数据如下。





紧接着对机场航班情况进行分析考察，于是可以结合入港飞机数给出各时间段离开机场客流量示意图。



假设模拟时不会出现出租车等待乘客的情况，即出租车到达上车点时总有人上车。设每小时到达机场车道排队的出租车数量为m,出租车司机平均等待时间为T。为了真实地反映实际情况，以秒为单位进行模拟。模拟开始时，只有乘客在机场进行等待，上车点没有 出租车排队。另，根据实际设定队列最大容量为200辆车。

模拟的每一秒首先会产生一个0到1之间的随机数，若该随机数小于P,则有一辆新到的出租车排到队列的末尾，并赋予其属性值。之后检查队列第一辆出租车的上车时间是否到达。若未到达，第一辆车上车时间减1秒;若到达，算出其等待时间,加到排队队列的总排队等待时间上，之后退出队列。之后另第二辆车成为队列首辆车，开始下一秒的模拟。重复上述步骤，最终可算出该m值下的出的T值。为了避免随机数产生的偶然性，模拟若干次取平均值，作为最终结果。改变m, T随之改变,用同样的方法得到所需要的数据点，进行拟合。

**5.4问题二模型的求解**

对模拟的不同时长进行整合比较，得到如下结果：



由表可以看到，同一模拟时间段下，m越大，等待时间越长;不同模拟时段下，模拟时间越长，平均等待时间越长。两个小时的模拟更贴合现实，因为第二个小时开始时会存在等待的出租车，存在“拒绝”情况也很实际;但之后等待时间大幅增加，这是偏离现实的，因为车主在等待时间过长情况下会选择离开而不是等待。所以一小时的模拟提供了下限值，而两小时的模拟提供了比较合适的上限值。因此，最终选择1.5小时的模拟时长进行数据拟合分析。

综上，将上文给出的返回路程x、收费标准g(x)以及出租车流量m三个根本因素代入问题一模型中并求解，可以给出各时段决策方案。

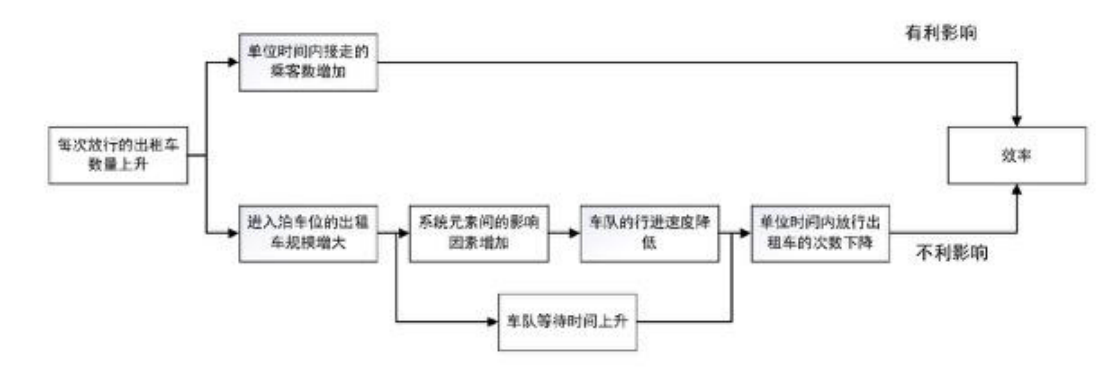


**5.5问题三模型的建立**

**5.5.1分析上客系统**

交通流理论(Traffic Flow Theory)【3】【4】是研究交通流随时间流随时间和空间变化规律的模型和方法体系。其中最重要的三个参数为:交通量、速度和密度。交通量为速度和密度的乘积。

根据交通流理论，交通系统通行能力大小与交通参与者密度有密切关系。若增加泊位(驶入出租车)数量可以增加乘车效率，但当泊位数量超过某阈值时，乘车效率不增反降。该情况下通行能力的下降主要体现在车流平均速度的减缓。具体的影响分析见下图。



**5.5.2结合约束确定约束条件**

结合上述分析得到目标函数：max δ=mt

并根据出租车驶入、接客、驶离等数据得到总用时为

t1=Lv

t2=m2-1S+m2Dv

t3=k3m2-1+τ

t=t1+t2+t3

再进一步得到接客安全距离为

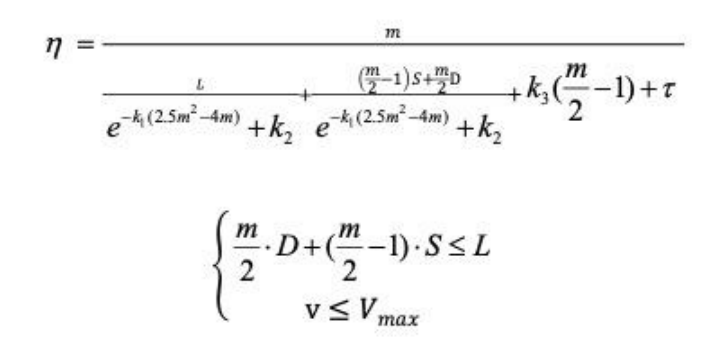
m2∙D+m2-1∙S≤L

综上所述，整理分析得到约束条件：

v=e-k1(2.5m2-4m)+k2t1=Lvt2=m2-1S+m2Dvt3=k3m2-1+τt=t1+t2+t3m2∙D+m2-1∙S≤Lv≤Vmax

**5.6问题三模型的求解**

根据上述模型，代入目标函数，得到：



当m= m'时，乘车系统的效率最大。对式求对自变量m的偏导，令其值为0，即可求出m的值，即：

ϑδϑm=0

**5.7问题四模型的建立**

目前机场出租车运营可能存在的情况根据载客距离分为:远途载客、中距离载客、短距载客三种情况受机场排队的接人的限制，出租车均付出了一定的时间成本，但是对于短途运输而言，可会出现司机消耗了大量时间，却不能获得足够多的利润，因此，造成了出租车收入不均衡的现象。

本问紧紧围绕司机的收益均衡性对“优先机制”进行设计。将载客前往市中心的司机利润作为标准利润，尽量通过“优先机制”使得短途载客的司机利润尽可能接近标准利润。

参考【5】国内机场对优先机制的设计，现借用“短途票”方案，即为了均衡短距载客出租车的收益，可以领一张免排队的短途票，送完客人后，这些司机就可以进入特殊接客区(无需排队)。

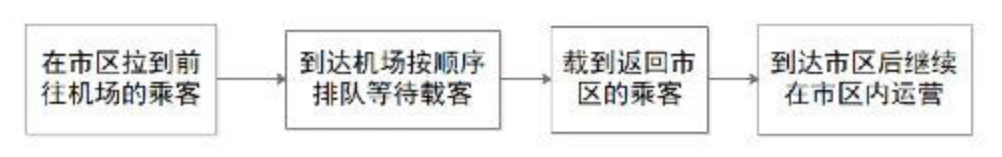
根据观察得到范围时间内短途车活动流程为



进一步得到其收益为

W=2WL+2WX2Lv+mv+2xv+m0/v

同理，对长途汽车进行观测



得到其收益为

W=2WL+W(T)2Lv+mv+T

结合上述条件分情况讨论，得到优化模型为

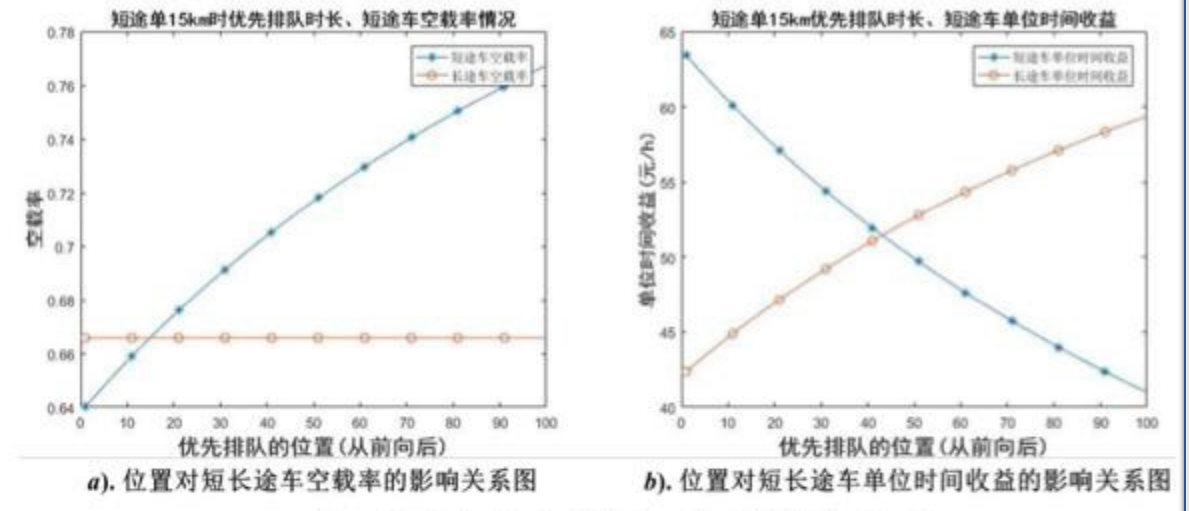
βshort=mv+xv+m0/v2Lv+mv+2xv+m0/vWshort=2WL+2WX2Lv+mv+2xv+m0/vβlong=m/v2Lv+m/vWlong=2WL+W(T)2Lv+mv+T

**5.8问题四模型的求解**

根据前面建立的模型，我们根据短途车上一个短途单的行驶里程Xxshort,来判断它这次应该优先安排在什么位置。

优先安排的最终目的是为了使短途车和长途车在空载率和单位时间的收益上都尽量均衡，于是我们先直观地观察短途车空载率、收益与上一个短途单的关系。

首先，我们观察当改变优先排队位置时，短途车空载率、收益与长途车的对比，以直观找到较优的排队位置。我们假设机场来了一个上一单行程为15km的短途车，那么不同排队位置的结果如图所示。



每一个短途行驶里程，都对应一个最优的排队位置，且短途越短，出租车优先排队位置越靠前，当行驶里程达到24km时，优先排队位置为90，已经接近排队总长度。粗略观察，当行驶里程在0~5km之内，应该排在10位之前，当行驶里程在5~12km之内，应该排在10~20 位之间，当行驶里程在12- -20km之间，应该排在20- 40位之间，当行驶里程大于20km，排队位置迅速接近总队列的尾部。

当我们按照最优排队方案排队之后，短途车的空载率与长途车的空载率近似相等，并且短途车的单位时间收益与长途车的单位时间收益也近似相等，达到了我们均衡长途车和短途车空载率、收益的目的，反映出这种优先排队方案的合理性。

**六、模型评价与推广**

**6.1模型的优点**

1、本文将出租车工作状态进行模拟，符合实际情况，在效率提升方面的求解得到较为精确的结果。

2、本文建立的是基于三维分析的模型，与工作实际非常吻合，经过灵敏度分析得到模型的稳健性好，使用范围广。

**6.2模型的缺点**

1、由于专业知识的欠缺以及题目假设条件导致的简化，本文件建立的模型无法完全仿真实际控制情况。

2、本模型研究的主要目标是中小型机场，应用到实际规划时，避免不了可能遇到规划大性机场的情况，此时对于模型本身而言，改变的不仅是参数数量的变化，参数的个数也会发生变化。比如:大城市的机场乘客的出行选择会更多，因为大型机场交通枢纽的陆侧交通方式更多，选择出租车的比例自然就会下降。而且此时出租车的管理排队问题也会变得更复杂。

**6.3模型的推广**

1、.模型可以较为真实的模拟出大面积动态信号状况，可以用于长时间检测。

2、模型可以推广到众多运输工作，通过本文建立的模型可以根据数据优化，调节出租车工作状态使其效率最高。

**七、参考文献**

【1】包丹文,郭唐仪，华松逸基于SP/RP融合数据的机场旅客出行方式选择行为分析[J].武汉理工大学学报(交通科学与工程版),2015,39(04):763-767.

【2】胡国强，吴树畅，《成本管理会计》[M]. 北京:西南财经大学出版社，2006.8.

【3】 Zhaodong Z,Shaohui C,Yanquan Y,et al.VISSIM Simulation Based Expressway Exit Control Modes Research[J].Procedia Engineering,2016,137:738-746.

【4】 Patel A, Mathew T V, Venkateswaran J. Real-time adaptive signal controller for non-lane following heterogenous road traffic[C].Communication Systems and Networks(COMSNETS),2016 8th International Conference On.IEEE,2016:1-6.

【5】]ttps://news.sina.cn/gn/2018- 10-28/detail- ifxeuwws8817753.d.html?pos= 3&vt=4

**附录**

**问题一模型求解程序**

sets :

t/1..4/ :P1,P2,n,m,h,Q;

endsets

data:

n=29 243 238.95 207 .42;

m=8.67 65 33.8 49;

enddata

max=@sum(t(i):Q(i)\*P1(i)+(1-Q(i))\*P2(i));

@for(t(i) :P1(i)=33-6.48-44.3\*h(i)/60);

@for(t(i) :P2(i)=44.3\*h(i)/60-6.48-3.3/ (h(i)/60+0.36)\*0.36);

@for(t(i) :h(i)=@if (m(i) #gt#12.6,2e-7\*m(i)^4-7e-05\*m(i)^3+0.0086\*m(i)^2+0.0578\*m(i)-0.6777，0)) ;

@for(t(i) :@free(P1(i)) ;@free(P2(i)) ;@bin(Q(i)));

End

**问题二模型求解程序**

**1、C语言队列模拟程序**

#include<stdio. h>

#include<stdlib. h>

#include<time. h>

#defineMAXCAR 60 //最大队列长度

#define MIN\_ PER \_HR 60. 0，

Typedefstructitem

{

long Arrive;

int BordingTime;

}Item;

Typedefstructnode

{

Item item;

structnode \* Next;

}Node;

Typedefstructqueue

{

Node \* front;

Node \* rear;

int items;

}Queue;

void InitializeQueue (Queue \* pq);

int QueueIsFul1 (constQueue \* pq);

int QueueIsEmpty (constQueue \* pq) ;

int QueueItemCount (constQueue \* pq) ;

int EnQueue(Itemitem, Queue \* pq);

int DeQueue(Item \* pitem, Queue \* pq);

void EmptyTheQueue (Queue \* pq) ;

staticvoid CopyToNode(Itemitem, Node \* pn) ;

staticvoid CopyToItem(Node \* pn, Item \* pi);

int NewCar (doublex);

Item Cart ime (longwhen) ;

double Average (doubleAve[]，intn);

int main (void)

{

Queue Line;

Item Temp;

double Hours;//模拟小时数

int Perhour;//每 小时平均来多少辆出租车

int Cycle;

double CycleLimit;

long TurnAway = 0;//因队列已满被拒绝的出租车数量

long Cars = 0;//加入队列的顾客数量

long Served = 0;//在模拟时间段接到乘客地出租车数量

long Sumline = 0;//累积的队列总长

int WaiTime = 0;//当前到第- ~辆车载客出发的时间

double MinPerCars;// 出租车到来的平均时间

long SumLineWait = 0://队列累积的等待时间

int t, tt;

double Ave[1000] = {}:

double SumAv[1000] = {};

puts("请输入模拟次数(不要超过1000次) : ");

scanf\_ s("%d"， &t);

tt = t://tt为循环次数

puts(”请输入模拟时长:”);

scanf\_ \_s("%lf"， &Hours) ;

puts("请输入平均每小时到机场排队的出租车数量: ");

scanf\_ s("%d”， &Perhour) ;

srand( (unsignedint) time (0));

while (tt--)//模拟开始

{

Cars = 0; Served = 0; SumLine = 0;

TurnAway二0; WaiTime 三0; SumLineWait = 0;

InitializeQueue (&Line);

CycleLimit = MIN\_ PER\_ JHR \* Hours \* 60;

MinPerCars = (MIN PER\_HR \* 60) / Perhour :

for (Cycle = 0.0; Cycle < CycleLimit; Cycle++)

{

if (NewCar (MinPerCars))

{

if (QueueIsFull (&Line))

TurnAway++;

else

{

Cars++;

Temp = Cartime (Cycle) ;

EnQueue(Temp，&Line) ;

}

}

if (WaiTime <= 0 & !Queue IsEmpty (&Line))

}

DeQueue (&Temp，&Line) ;

WaiTime = Temp. BordingTime;

SumLineWait += Cycle - Temp. Arrive;

Served++;

}

if (WaiTime > 0)

WaiTime--;

SumLine += QueueI temCount (&Line) ;

}

if (Cars > 0)

{

printf("------------------------\n");

printf("%-40s%-151d\n"，”进 行排队的出租车总数(辆) :”，Cars);

printf("%-40s%- 151d\n"，“接到乘客的出租车数量(辆) :”，Served) ;

printf("%- 40s%- 151d\n"，” 由于场地限制拒绝的出租车数量(辆) :”, TurnAway) ;

printf("%- 40s%.2f\n"，” 出租车平均等待时间(秒) :”，(double) SumLineWait / Served) ;

}

else

printf("No Cars!");

Ave[tt] = (double)SumLineWait / Served;

SumAv[tt] = Cars;

EmptyTheQueue (&Line) ;

printf("\n");

}

printf("-------------------------------\n");

printf("%d次模拟出租车等待时间平均值为%.2f分钟，进行排队出租车的””(%.2f个小时)平均数为%.2f辆”，t, Average(Ave, t) 1 60, Hours, Average (SumAv, t)); :

getchar();

getchar();

return 0;

};

int NewCar (doublex)

{

if (rand() \* x / RAND \_MAX< 1)

returntrue;

else

returnfalse;

}

Item Cartime (longwhen)

{

Item cust;

//cust. BordingTime = (rand() %3 + 1) \* 60;

cust. BordingTime = 90;

cust.Arrive = when;

return cust;

}

void InitializeQueue(Queue \* pq)

{

pq->front = pq->rear = NULL:

pq->items = 0;

}

int QueueIsFull (constQueue \* pq)

{

returnpq->items == MAXCAR;

}

int QueueIsEmpty (constQueue \* pq)

{

returnpq->items == 0;

}

int QueueItemCount (constQueue \* pq)

{

returnpq->items; .

}

int EnQueue(Itemitem, Queue \* pq)

{

Node \* pnew;

if (QueueIsFull(pq))

returnfalse;

pnew = (Node \*) mal loc (sizeof (Node)):

if (pnew == NULL)

{

fprintf (stderr, "Unble to allocate me

exit(1);

}

CopyToNode(item, pnew) ;

pnew ->Next = NULL:

if (QueueIsEmpty (pq))

pq->front = pnew;

else

pq->rear- >Next = pnew;

pq->rear = pnew;

pq->items++;

returntrue; .

}

int DeQueue(Item \* pitem, Queue \* pq)

{

Node \* pt;

if (Queue IsEmpty (pg))

returnfalse;

CopyToItem(pq- >front, pitem) ;

pt = pq->front;

pq->front = pq->front- >Next;

free(pt) ;

pq->i tems-- ;

if (pq->items == 0)

pq->rear = NULL;

returntrue;

}

void EmptyTheQueue (Queue \* pq)

{

Item dummy ;

while (!QueueIsEmpty (pq))

DeQueue (&dummy, pq);

}

staticvoid CopyToNode(Itemitem, Node \* pq)

{

pq->item =item;

}

statievoid CopyToItem(Node \* pq, Item\* pi)

{

\*pi=pq->item;

}

double Average (doubleAve[], intn)

{

int i;

double Sum = 0. 0;

for(i=0;i<=n-1;i++)

{

Sum += Ave[i];

}

return Sum /n;

}

**2、MATLAB程序**

Syms y v w yita1 T

y = w-yita1\*T -yita1\*0. 11\*w +v;

ans1 = diff(y, v) ;

ans2 = diff(y, w) ;

ans3 = diff(y, yita1);

ans4 = diff(y, T);

**问题三模型求解程序**

Syms m L k1 k2 k3 tao S D

f = m/ (L/exp (-k1\* (2. 5\*m -4) \*m+k2)+( (m/2-1)\*S+ (m\*D) /2) /exp(-k1\* (2. 5\*m-4) \*m+k2)+tao+ (m/2 -1)\*k3);

dd = diff(f,m) ;