

出租车的动态调度问题

摘要

在大数据时代，通过分析一些重要的客流量大的枢纽场所的客流数据，设计出交通枢纽的合理的调度方式，成为了一个非常重要的课题。本题以机场陆侧的出租车交通组织模式为切入点，站在不同的角度对机场出租车的组织模式提出了需求。本文使用了基于排队论构建的模型，分析了不同场景下，不同的出租车组织模式对出租车运输效率的影响。利用 MATLAB 进行了交通场景的仿真模拟，并使用多种优化算法求解出最优解，最后分析了模型的依赖性和优缺点。

针对问题一 在问题一中，由于出租车司机能观测到某时间段抵达的航班数量和“蓄车池”里已有的车辆数，故针对选择排队等待拉客还是直接选择返回市区应以这能观测到的两个参数作为主要参考依据。因此，我们只需要简单地做出线性规划问题，求出对应于不同参数采取不同的选择。由于航班来临的时刻人数会出现一种符合正态分布的曲线到达‘乘车区’，而此时‘蓄车池’中出租车数量的多少又会正比于等待时间，于是可以推测出选择等待后需要等待的时间，然后假定在相同长一段时间内，直接返回市区考虑平均每分钟收益和在等待区载客并载客后的收益进行比较来综合判断如何选择。

针对问题二 根据河南郑州新郑国际机场的出租车动态变化网页，经过一昼夜爬虫后得到一昼夜的出租车的动态变化表，根据该段时间内到达郑州航班的多少，选择以 2 个小时内作为基准，代入问题一所建立的出租车等待模型，综合对比理论值和实际值之间的区别并求取平均值，求得平均相对误差不超过 0.1036。然后根据问题一得到的分界值来进行选择。在机场出租车系统正常调度的情况下，相对误差不超过 0.1，但同时也对当时出租车以外的交通系统的变化有着较大的依赖性。

针对问题三 对于题目给定的双车道并行的场景，设计了不同的“上车点”和乘客安排方式。对于单停车道模型和双停车道模型，本文分别使用了排队论中的 $M/M/1$ 和 $M/M/2$ 模型对乘客在队列中的平均等待时间进行了分析和计算。本文对两种不同的排队模型进行了比较，得出了在不同条件下优先采取某种排队模型的结论。并证明了在乘客到达速率小于出租车到达速率的条件下，优先采用双停车道模型，在乘客到达速率大于出租车到达速率的条件下，优先考虑单停车道模型。

针对问题四 对于问题四，我们认为载客的行程越长，出租车的载客收益也就越大，于是采取对载客行程较短的出租车给予他们一定的优先权。综合考虑出租车在上一次等待时长和这一次载客的收益，得出该出租车的“优先值”，然后设置一个“优先阈值”，大于该“优先阈值”即可直接插入到优先级高的车道，并且拥有优先载客的权力，但是对于低优先级的车，被插队次数不应超过一定的值，于是通过对上述优先值算法，“优先阈值”，被插队次数最大值进行遗传算法，退火算法，粒子群算法得到每个出租车收益的方差的最小值时所满足的条件，根据该条件来安排一个可行的“优先”方案。

关键词 排队论 动态调度 遗传算法 退火算法 粒子群算法

1 问题重述

1.1 问题背景

出租车是乘客往返市区与机场的主要交通方式之一。机场有客流密度大的特点，而乘客需要在固定的出租车上客点乘车。由于上客点地点和条件有限，且上客的效率受到了乘客、机场航班、出租车司机等多种因素的影响。所以需要一种优化的解决方案能够提高乘客在出租车上客点的乘车效率，以减少乘客和出租车司机的等待时间。

对于司机而言，离开下客区后，选择直接离开机场空载返回、或是排队等待载客，将会对司机的收益产生影响，而排队的等待时间便是影响司机收益的关键。所以，需要一种可靠的算法，能够根据便于获得的一些数据来协助司机的决策过程。此外，由于出租车乘客前往目的地的随机性，部分司机可能在等待较长时间之后，乘客前往的目的地较近，从而导致司机的收益相对偏低。所以需要一种合理的方式，使得出租车司机的收入能够保持尽可能均衡。

1.2 问题重述

国内机场一般将出租车乘车区分为上客区和下客区两个分开的区域，并且对于离开上客区的出租车，司机可以根据的意愿选择放空车辆直接离开或前往“蓄车池”载客。对于司机，他们可以得到当前“蓄车池”内的车的数量以及当前的航班数量来决定是否进入“蓄车池”等待，并且司机通常还可以根据自己的经验，结合季节、时间等因素判断乘客数量的多少。如果司机选择直接放空返回，则可能会有返回时的空载费用和载客的潜在收益的损失。对于机场管理者，他们需要在上客地点的等候区采取“分批定量”放行的措施来让乘客依次进入上客区乘车。

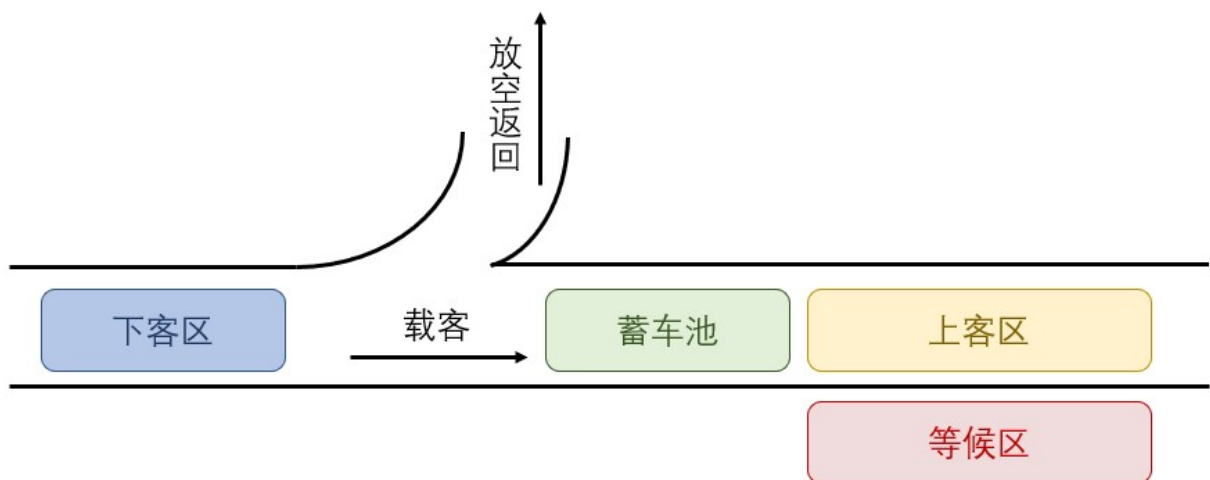


图 1: 机场出租车乘车区域示意图

根据以上条件，我们需要研究如下的 4 个问题：

- (1) 建立一个模型，用于计算得到出租车司机载客与放空返回的选择策略。建立模型时综合考虑机场在不同时间的旅客数量的变化规律和其它的相关因素，使得出租车的做出的策略能够使其收益最大化。
- (2) 获取国内某一机场的实际数据，根据建立的模型和机场的实际数据计算出司机的决策策略，将计算得到的数据（如预期出租车等待时间）与真实数据进行比较和分析，并由此可以出模型的合理性与对相关因素的依赖程度。
- (3) 机场在高峰时段，乘客和出租车可能会排起长队。机场乘车区设置有两条并行车道，建立一个模型，确定合理的“上车点”的位置，并合理安排出租车的行驶方式和乘客的上车方式，在确保安全的前提下，使得乘客的平均排队时间能够最短，乘车效率达到最大。
- (4) 出租车在机场载客的目的地是随机的，并且出租车司机无法自主选择载客或拒绝载客，所以出租车司机的收益可能会出现不均衡的情况。为了解决这个问题，考虑给予返回的短途载客的司机在下次载客时的优先权。建立一个模型，给出可行的优化方案，使得出租车司机们的收益能够尽可能均衡。

2 模型假设

1. 考虑到每个人上车的时间相差较小，可以认为每个人从乘车区上车到出租车起步离开的时间固定，我们记这个时间为 t_0 。
2. 由于到港航班的分布与相对均匀，本文认为在考虑的一段时间之内，航班到达机场的时间是等间隔的。
3. 根据统计学规律，在一段时间之内，各个航班到达的乘客前往出租车乘车区域乘车的人数符合泊松分布。
4. 根据统计学规律，每一个到达航班的需要乘坐出租车的乘客到达出租车乘车等候区的时间满足正态分布。
5. 由于汽车在等待时由于仍在运行中，其耗油量近似为正常运行时的 $\frac{1}{2}$ ，所以可以认为出租车在“蓄车池”等待期间消耗的油量较正常运行时更低。
6. 根据统计学规律，从机场离开的出租车前往目的地的距离满足正态分布的规律。

3 符号说明

表3列出了本文需要的符号。

表 1: 符号说明

符号	意义	单位
N_f	一段时间内的航班数量	架
N_c	蓄车池内的出租车数量	辆
T	模型考察的总时间长度	min
t_w	出租车司机等待的时间	min
t_r	出租车从机场返回市区的时间	min
n_i	第 i 架航班上需要乘坐出租车的人数	人
M_0	一架航班乘坐出租车的人数最大值	人
Δt_f	航班的时间间隔	min
p_i	第 i 架航班乘客乘坐出租车的概率	
q_i	第 i 架航班乘客抵达候车区的时间分布	
λ_1	航班乘坐出租车人数分布的均值	
λ_2	乘客到达乘车区时间分布参数	
μ_1	乘客到达乘车区时间正态分布均值	
μ_2	乘客上车间隔指数分布参数	
N	开始时蓄车池内出租车能够容纳的乘客人数	人
\bar{t}_0	平均每辆车上车所需耗费时间	min
t_1	到达等候区乘客累计达到 N 的时刻	min
t_2	载客出租车返回市区的时刻	min
t_{extra}	出租车额外等待的时间	min
ε_0	平均每辆车的载客数	人 / 辆
E_1	每辆出租车从空载返回到 t_2 时刻的利润	元
E_2	每辆出租车从进入蓄车池到 t_2 时刻的利润	元
\bar{E}_o	出租车正常行驶平均每分钟油费	元 / min
\bar{E}_{wo}	出租车排队时平均每分钟油费	元 / min
\bar{E}_i	出租车在市区内运营平均收入	元 / min
\bar{E}_c	出租车平均载客返回收入	元
W_{q1}	双车道停车模型乘客平均等待时间	min
W_{q2}	单车道停车模型乘客平均等待时间	min
C	单车道停车模型服务台数目	
\bar{E}_a	机场出租车单位时间平均收入	元 / min
d	行驶里程	m
ω	出租车返程收入	元

4 问题分析

4.1 问题一的分析

问题一本质就是要建立一个出租车司机关于一段时间内航班数量 N_f 与当前蓄车池内出租车的数量 N_c 的决策模型 $f(N_f, N_c)$ ，而这个模型可以分成如下的两个部分：

- (1) 出租车时间等待时间 t 与航班数量 N_f 和蓄车池内出租车数量 N_c 的预测模型 $t_w(N_f, N_c)$ ，通过计算机模拟的方法，得出可行域内的不同 N_f 和 N_c 下的 t_w 的数值，可以绘制出 t_w 关于 N_f 和 N_c 的函数关系图像。
- (2) 建立数学模型，得到出租车不载客利润 E_1 和出租车载客利润 E_2 与出租车等待时间 t_w 之间的函数关系 $E_1(t_w)$ 和 $E_2(t_w)$ ，并做出出租车司机的决策 $\max(E_1(t_w), E_2(t_w))$ ，由此得到出租车司机关于等待时间的决策函数 $E(t_w)$ 。

将如上的两个数学模型得到的函数 $t(N_f, N_c)$ 和 $E(t_w)$ 联立，最终得到出租车司机关于 N_f 和 N_c 的决策函数 $f(N_f, N_c)$ 。

4.2 问题二的分析

问题二的本质是需要根据搜集国内的某一个机场的实际数据，带入问题一中得到的模型，得到不同条件下出租车司机的最佳选择策略，由此对问题一中的模型进行数值上的验证，所以问题二的解决需要分为如下几个步骤：

- (1) 收集国内某一城市的机场航班数量分布情况、出租车出入场、场内出租车数量、该城市机场到市区出租车均价、单位时间油耗价格和出租车单位时间正常收入。
- (2) 根据收集到的数据，带入问题一中建立的模型，计算出出租车司机等待时间的阈值 t_{w0} 。
- (3) 根据收集到的参数，绘制出 $t_w - N_f, N_c$ 图像，作平面 $\alpha: t_w = t_{w0}$ ，得到平面 α 和 t_w 平面的交线 $l: \alpha \cap t_w(N_f, N_c)$ ，则 l 即为 t_w 的阈值曲线。
- (4) 根据收集到的数据，随机抽取部分数据，带入上述模型进行计算，得到预计的等待时间 \hat{t}_w ，与实际的等待时间进行比较，分析模型的误差和对相关因素的敏感程度。

4.3 问题三的分析

问题三要求给出一个合理的乘客“上车点”设置方式，在保证安全的前提之下，能够降低乘客的平均等待时间。在此前提之下，可以建立如下的两种数学模型：

- (1) **双车道停车模型**。两条并行车道都可以停车、上车，但为安全考虑，只在最前端开设一个上车点供上车，乘客可以从出租车队列的最前端穿越马路乘坐出租车。这种模型，可以用排队论中的 $M/M/2$ 模型表示，并通过排队论模型计算出乘客的平均等待时间 W_{q1} 。
- (2) **单车道停车模型**。两条并行车道，靠航站楼的一边可以设置多个停车点，排多条队伍上客，另一条车道用于出租车车辆通行，使得每个停车点出租车相互不干扰。这种模型，可以用排队论中多个独立的 $M/M/1$ 模型表示，并可以由此用排队论模型计算出乘客的平均等待时间 W_{q2} 。

通过对如上的两种数学模型进行分别建模和求解，可以分别求解出两种模型中的乘客平均等待时间 W_q ，并根据平均等待时间对其进行比较，从而得到两种方案中的最优方案。

4.4 问题四的分析

问题四的题目描述了一个机场出租车目的地的随机分布导致的收益不均衡的问题，并要求给出优化方案。而对于机场而言，可行的优化方案是设定一个最优的用于对出租车进行分类的“优先级”计算方式。并设置一个合理的阈值，来划分“高优先级”和“低优先级”，并允许“高优先级”的出租车能够在下一次返回机场时，通过“插队”的方式优先载客，缩短等待时间。对于这个问题，我们可以先使用计算机模拟出租车系统的运行过程，计算出每辆出租车的单位时间的平均收益，由此计算出所有出租车的收入的方差。之后可以使用遗传算法等多种算法来优化模型中的各种参数，得到最合理的优先级计算方法和高低优先级划分阈值。

可以将这个问题分为以下几个过程来解决：

- (1) 建立数学模型，给出出租车优先级关于各个自变量与参数的关系。
- (2) 编写计算机程序，模拟机场和出租车的运行过程，并且根据初始设定的参数计算出出租车的优先级，最终统计出出租车单位时间内收益的方差。
- (3) 使用优化算法，找到使方差值最小的参数值，并完成模型的灵敏度分析和评价。

5 模型建立

针对本题，本文一共建立了三个数学模型：用于出租车司机决策的模型、用于设置机场停车点的模型、用于短途载客返回出租车优先权的模型。下面依次对这三个模型的建立作详细说明。

5.1 用于出租车司机决策的模型

用于出租车司机决策的模型一共分为出租车等待时间 $t_w(N_f, N_c)$ 和出租车司机决策 $E(t_w)$ 两个部分，以下依次描述这两个模型的具体建立过程，并将两个模型综合得到最终的决策模型。

5.1.1 出租车等待时间

乘客在乘车区排队等待出租车采用排队论中 $M/G/1/\infty/N/FCFS$ 模型进行分析。出租车的等待时间分为两部分，一部分是乘客上车占用的时间 Bt_0 ，另一部分是由于乘坐出租车的乘客前往乘车点的时间间隔过大，导致的出租车空闲等待的时间：

$$t_{extra} = \sum_i t_{extrai} \quad (1)$$

由此可以得到等待时间 t_w 的计算公式：

$$t_w = N_c \bar{t}_0 + t_{extra} \quad (2)$$

航班的之间的间隔 Δt_f 与总时间 T 和航班数目 N_f 满足如下的关系：

$$\Delta t_f = T/N_f \quad (3)$$

对于一段时间内航班到达的数量分布，本文认为其满足均匀分布的规律。其中，对于每一架航班乘坐出租车的人数 n_i ，其与每一架航班乘坐出租车的最大人数 M_0 和该架航班上的乘客乘车概率 p_i 满足如下关系：

$$n_i = M_0 \cdot p_i \quad (4)$$

由此，我们可以通过公式4来确定不同时间段到达机场的乘客的数量，对于概率 p_i ，本文认为其满足均值为 λ_1 的泊松分布：

$$p_i(X = k) = \frac{\lambda_1}{k!} e^{-\lambda_1} \quad (5)$$

对于进入等候区排队的乘客，我们使用 FCFS 模型。其中对于每一架到达的航班，从航班到达，至需要乘坐出租车的乘客抵达出租车等候区排队的时间，本文认为其关于时间的分布满足均值为 μ_1 ，标准差为 σ 的正态分布：

$$q_i((i-1)\Delta t_f + t) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(t-\mu_1)^2}{2\sigma^2}}, \quad t \geq 0 \quad (6)$$

对于此时在蓄车池中的车辆数量，其数量 N_c 与能够承载的乘客人数 N 满足如下关系：

$$N = N_c \cdot \varepsilon_0 \quad (7)$$

根据以上描述的符号关系，与式7联立，我们可以列出承载乘客与到达乘客的关系：

$$\sum_i \int_0^{t_1} M_0 \cdot p_i q_i(t) dt = N_c \cdot \varepsilon_0 \quad (8)$$

5.1.2 出租车收益与等待时间关系

出租车的收益可以分为空再返回和等待载客两部分进行分别分析。分析的时间段为出租车下客完成（时间起点），至载客返回的出租车抵达市区的时间 t_2 。

(1) 空载返回出租车收益。

空载返回的出租车由于无需等待，所以其能够在空载返回市区后直接载客运营，故空载返回的出租车的收入为其在市区运营时的收入，其耗费为全程的油费。于是我们可以得到如下空载返回出租车收益 E_1 关于 t_2 的关系：

$$E_1(t_2) = \overline{E}_i \cdot (t_2 - t_r) - \overline{E}_o \cdot t_2 \quad (9)$$

(2) 载客出租车收益。

由于载客出租车需要在蓄车池内等待乘客，而根据假设，蓄车池内等待不消耗任何成本，所以载客出租车的收入为载客从机场返回市区的收入，消耗为返回市区时的油耗。于是我们可以得到如下空载返回出租车的收益 E_2 的表达式：

$$E_2 = \overline{E}_c - t_r \cdot \overline{E}_o - t_2 \cdot \overline{E}_{wo} \quad (10)$$

此外，我们还可以得到出租车等待时间 t_w 和载客出租车返回市区的时间 t_2 时间的关系：

$$t_2 = t_w + t_r \quad (11)$$

将式9、10、11联立，我们可以得到用于司机决策的 t_w 的阈值，由此建立用于出租车司机决策的模型。

5.2 用于设置机场停车点的模型

用于设置机场停车点的模型一共分为两类：双车道停车模型和单车道停车模型。先分别建立两种模型。

(1) 双车道停车模型。

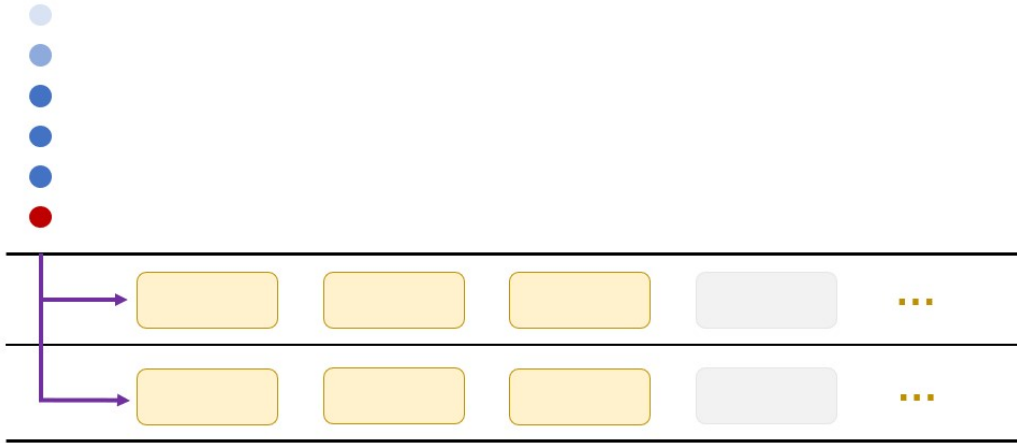


图 2: 双车道停车模型示意图

双车道停车模型示意图如图2所示。为了安全考虑，由于在这种模型中，乘客需要穿越马路到达外侧车道乘车，所以乘客不应当在汽车可能行驶的队列中后部穿越马路乘车，模型中的“上车点”只有一个，设置在出租车队列的最前方。队列中的候车乘客可以等待车辆停稳之后，从出租车队列的最前方穿越马路，在两辆车中选择一辆出租车搭乘。

在上述前提条件下，采用经典的排队论建立模型。这是一个典型的单队列、双服务台模型，并且顾客到达时间满足参数为 λ_2 的负指数分布，服务时间（即相邻前后辆车在队列最前方停稳的时间间隔）满足参数为 μ_2 的负指数分布，即排队论中的 $M/M/2$ 模型。根据该排队论模型，我们有如下公式 [1]：

$$W_{q1} = \frac{\lambda_2}{\mu_2(\mu_2 - \lambda_2)} \quad (12)$$

根据式12即可计算顾客的平均等待时间，从而对双车道停车模型的乘车效率进行评估，并与单车道停车模型进行比较，得到较优的方案。

(2) 单车道停车模型。

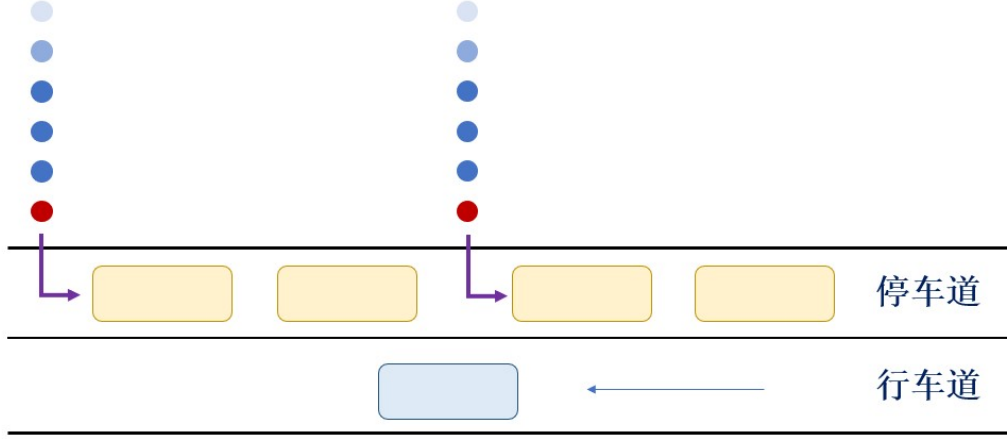


图 3: 单车道停车模型示意图

单车道停车模型示意图如图3所示。在这种模型中，靠近航站楼的一侧可以设置多条队伍，对应多个停车点。在排队论中，这个模型即为多条互相独立的队伍和多个服务台，本文定义服务台的数量为 C 。在这个模型中，顾客到达的时间间隔符合参数为 $\frac{\lambda_2}{C}$ 的负指数分布，车辆服务时间符合参数为 $\frac{\mu_2}{C}$ 的负指数分布。所以该模型为 C 个互相独立的 $M/M/1$ 模型的组合。设变量 ρ 满足：

$$\rho = \frac{\frac{\lambda_2}{C}}{C \cdot \frac{\mu_2}{C}} = \frac{\lambda_2}{C\mu_2} \quad (13)$$

可以用如下的公式 [1] 计算其顾客平均等待时间 W_{q2} ：

$$W_{q2} = \frac{1}{\lambda_2} \cdot \frac{C^C \rho^{C+1} \left(\sum_{n=0}^{C-1} \frac{C^n}{n!} \rho^n + \frac{C^C}{C!} \cdot \frac{\rho^C}{1-\rho} \right)^{-1}}{C! (1-\rho)^2} \quad (14)$$

分别带入实际数据，计算式12和式14的值，将二者进行比较，可以得出相应的最优解。

5.3 用于短途载客返回出租车优先权的模型

对于该模型，假设出租车从机场出发，抵达目的地的里程满足均值为 μ ，标准差为 σ 的正态分布 $d \sim N(\mu, \sigma^2)$ ，并有如下关于里程阈值 d_0 的返回策略：

$$\text{返回策略} \begin{cases} \text{远程，不返回机场} & t \geq t_0 \\ \text{近程，50\% 几率返回机场} & t < t_0 \end{cases} \quad (15)$$

定义优先级函数 $r(\overline{E}_a, t_w)$ 满足：

$$r(\overline{E}_a, t_w) = \alpha \cdot \overline{E}_a - t_w \quad (16)$$

其中 α 为参数。

定义单位时间平均收益 \overline{E}_a 满足：

$$\overline{E}_a = \frac{\omega(d) - \overline{E}_o \cdot t_{\text{运行}}}{t_w + t_{\text{运行}}} \quad (17)$$

其中 $\omega(d)$ 为出租车的收费，与路程 d 相关，可以代入某城市的出租车具体收费价格进行价格计算。

定义优先级阈值 r_0 ，规定满足如下关系：

$$\text{优先级别} \begin{cases} \text{低级} & r(\overline{E_a}, t_w) > r_0 \\ \text{高级} & r(\overline{E_a}, t_w) \leq r_0 \end{cases} \quad (18)$$

机场的出租车入口分为两条车道，分别为普通车道和优先车道。当出租车第一次进入机场时，认为其默认进入普通车道排队；当出租车在短时间内再次返回机场并选择载客时，机场会根据前述规则判断该出租车辆的优先级，优先级高的车辆进入优先车道排队，优先级低的车辆进入普通车道。优先车道与普通车道按照 $\beta : 1$ 的比例进入上车区载客。每次车辆进入机场时，上一次进入机场的数据即被清空，其相关数据将被重新计算。

定义评价函数为出租车单位时间平均收入的方差：

$$\sigma^2(\overline{E_a}) = \frac{1}{N_m} \sum_i (\overline{E_{ai}} - \mu_m)^2 \quad (19)$$

其中 N_m 代表总出租车数目， μ_m 代表出租车的总平均单位时间收益。该模型本质是一个多参数的最优化问题，目标是： $\min \sigma^2(\overline{E_a})$ 。使用优化算法得到最优解及其参数，即完成此题的目标。

6 问题解答

6.1 问题一的解答

对于问题一，本文建立了出租车司机的决策模型。这个模型分为出租车等候时间和出租车收益两个部分，根据以上建立的模型分别对这两个模块进行求解。

对于出租车等候时间，在选定的时间范围之内，建立了航班到达、乘客前往乘车区，和乘客在乘车区排队上车的模型，即出租车的等待时间 t_w 关于航班数量 N_f 和蓄车池内出租车数量 N_c 的函数关系，并使用如图4所示的程序模拟了这一过程。

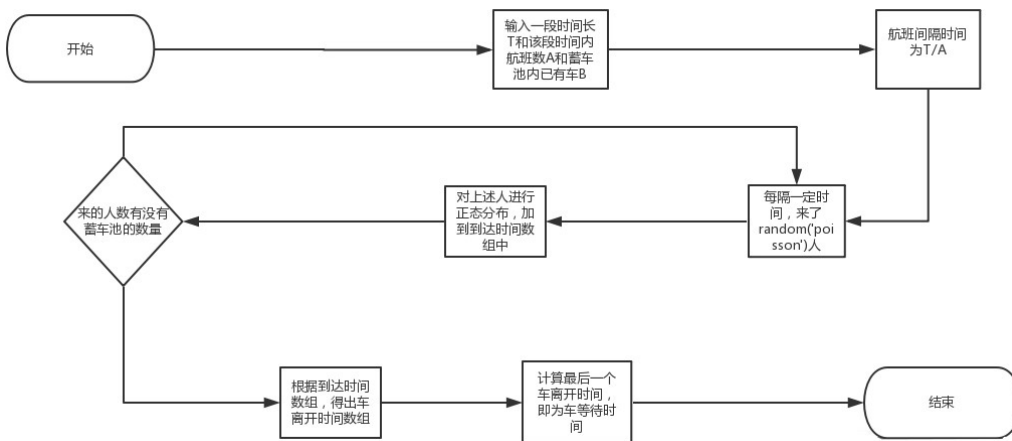


图 4: 出租车等待时间模拟程序框图

该计算机程序的算法的运行主要分为如下几个步骤：

- (1) 程序接收输入的两个数据：一段时间内的航班数量 N_f 和此时蓄车池内的出租车数量 N_c ，由此计算出航班之间的间隔时间。

- (2) 当每一架航班到达时，程序调用 random 函数根据泊松分布的规律，计算出每一架航班的人数。
- (3) 将每一架航班到达的人数按照正态分布的时间规律，加入到不同时间段内的人数的数组中，实现排队队列的模拟功能。直到到达队列中的人数达到蓄车池内的出租车能够容纳的人数 N 为止，结束循环。
- (4) 模拟一个队列，将各个时间的排队的数量依次加入乘车队列中，与此同时蓄车池内的出租车按照一定的时间间隔进行载客。最后计算整个过程花费的时间，并输出结果。

通过运行如图所示的程序，我们得到了不同条件下的出租车司机需要等待的时长。取不同的 N_f 和 N_c 的值，输入程序，可以得到如图5所示的函数图像。

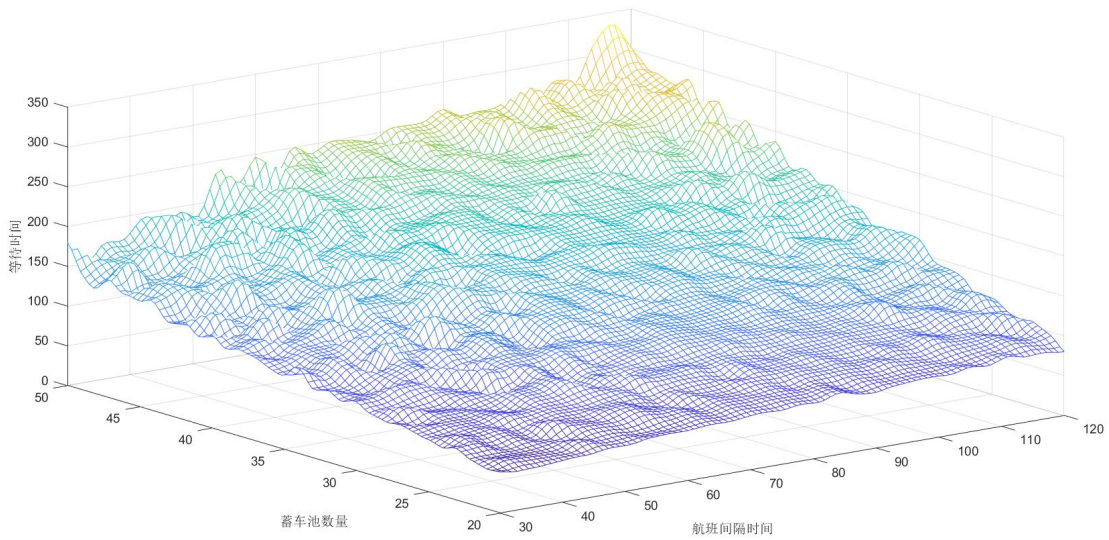


图 5: $t_w(N_f, N_c)$ 函数图像

对于出租车收益，通过联立式9、式11和式10可以得到等待时间 t_w 的分界值：

$$t_{w0} = \frac{\overline{E_c} - \overline{E_{wo}} \cdot t_r}{\overline{E_t} - \overline{E_o} + \overline{E_{wo}}} \quad (20)$$

综上所述，根据如下关系，可以得出出租车的最佳决策：

$$\begin{cases} \text{空载返回} & t_w(N_f, N_c) \geq t_{w0}, \\ \text{等待载客} & t_w(N_f, N_c) < t_{w0}. \end{cases} \quad (21)$$

6.2 问题二的解答

对于问题二，我们从各种渠道收集了关于国内某城市的机场和出租车的多种数据，用于对问题一中建立的模型进行正确性的分析与评估。

(1) 出租车等待时间阈值计算。

我们收集了问题一中的模型中所需要的基本数据，这些数据在表中列出。

表 2: 城市出租车基本数据 [2-3]

项目	值	单位
平均车速	31.23	km/h
空驶率	21.13%	
单价	1.82	元 / km
油费	0.54	元 / km
$\overline{E_c}$	62	元
t_r	50	min

根据这个表格, 可以计算出单位时间出租车平均收入 $\overline{E_t} = 0.75 \text{元}/\text{min}$, 单位时间出租车正常行驶平均油费 $\overline{E_o} = 0.28 \text{元}/\text{min}$, 排队中出租车单位时间平均油费 $\overline{E_{wo}} = 0.14 \text{元}/\text{min}$ 。将该数据与上表数据代入式20进行计算, 计算得到的阈值:

$$t_{w0} = \frac{\overline{E_c} - \overline{E_{wo}} \cdot t_r}{\overline{E_t} - \overline{E_o} + \overline{E_{wo}}} = 90 \text{ min} \quad (22)$$

即若预计的等待时间 $\hat{t}_w \geq 90 \text{ min}$ 时, 出租车选择空载返回; 否则选择前往蓄车池内等待载客。

(2) 出租车等待时间预测。

根据某城市机场在一天内航班起降的数据, 以及一天之内各个时间段的场内出租车数和出租车数据, 使用问题一中的模型对出租车的等待时间进行了计算。其中, 到达乘客乘坐出租车的占比为 28%, 每架航班平均载客数量为 100 人 [2, 4], 取 $\mu_1 = 10, \sigma = 6$, 得到如图6所示的函数图像:

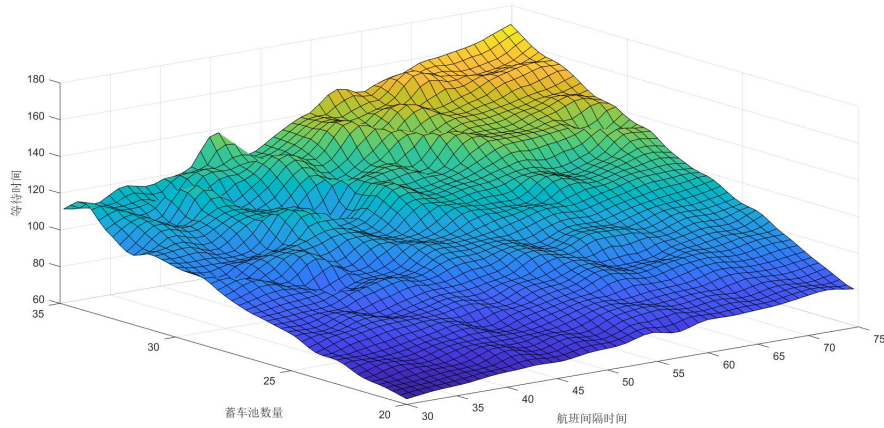


图 6: 实际参数下的 $t_w(N_f, N_c)$ 函数图像

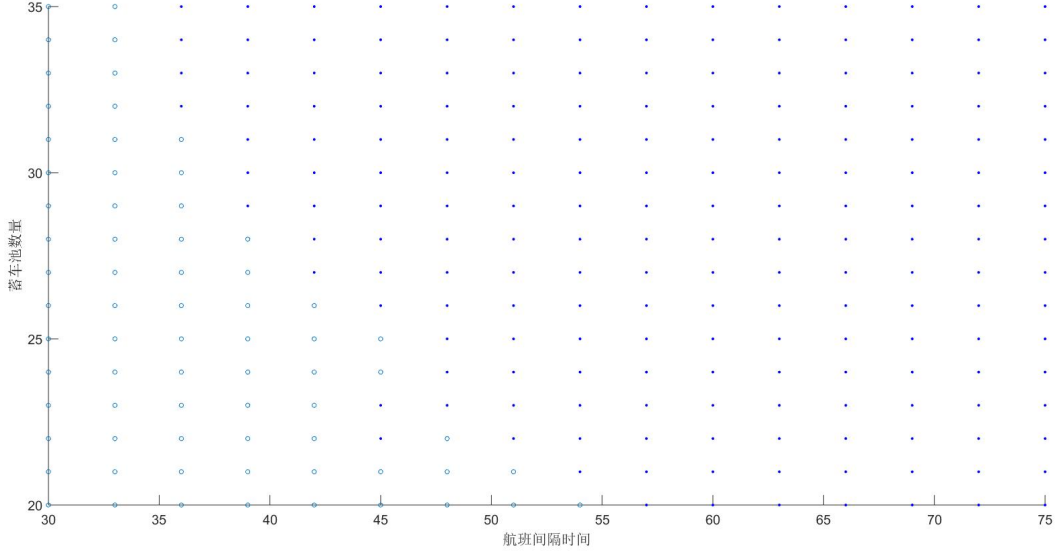


图 7: 出租车司机决策预测

图6反映了在不同的 N_f 和 N_c 下的 \hat{t}_w 和 t_w 的大小分布情况，图7反应的是在不同的 N_f 和 N_c 下，出租车做出的决策的预测是空载返回（实心点）或是进入蓄车池等待载客（空心点）。

根据预测得到的不同 N_f 和 N_c 下的 \hat{t}_w 与 t_w 进行比较，计算出模型的相对误差：

$$e_1 = \frac{|\hat{t}_w - t_w|}{t_w} \times 100\% = 10.36\% \quad (23)$$

由此可以看出，该模型与实际情况的误差在合理范围之内，可以用于预测现实情况下的司机的等待时间。

6.3 问题三的解答

本文建立了关于问题三的两中可能的模型：双车道停车模型和单车道停车模型，现在对两种模型代入特定的数据，求解模型的最优值。考虑到实际中航站楼现有的设施情况以及客流秩序安排的便捷性和合理性，我们在单停车道模型中取乘车点数目 $C = 2$ ，带入式14进行计算，并化简得到：

$$W_{q2} = \frac{4\rho^3}{\lambda_2(1 - \rho^2)} \quad (24)$$

若实际情况中，单停车道模型相较于双车道停车模型更好，即：

$$\frac{\lambda_2}{\mu_2(\mu_2 - \lambda_2)} > \frac{4\rho^3}{\lambda_2(1 - \rho^2)} \quad (25)$$

解不等式25，得：

$$1 < \frac{\lambda_2}{\mu_2} < 2 \quad (26)$$

当 $\frac{\lambda_2}{\mu_2} \geq 2$ 时，由于两个模型的服务速度均小于顾客到访的速度，故队伍的长度不收敛，无法计算出此时的平均等待时间，此时需要采用例如限流等措施来控制人流量大小。

根据国内某机场统计得到的数据平均值 [4]，我们可以得到如下表的参数值：

表 3: 问题三模型的实际参数

项目	值
μ_2	9.375
λ_2	17.68
λ_2/μ_2	1.86

由此可知, 根据此机场该时间段的实际数据, 使用单车道停车模型 ($C = 2$) 更为合适。综上所述, 在模型定义的范围內 ($0 < \frac{\lambda_2}{\mu_2} < 2$), 我们可以得到如下的选择方案与人流量、车流量的关系:

$$\begin{cases} \text{双车道停车模型} & 0 < \frac{\lambda_2}{\mu_2} \leq 1 \\ \text{单车道停车模型}(C = 2) & 1 < \frac{\lambda_2}{\mu_2} < 2 \end{cases} \quad (27)$$

6.4 问题四的解答

根据上述的算法, 我们设计了如下伪代码的算法来模拟这个过程, 计算出出租车单位时间的方差平均值 $\overline{E_a}$ 。

Algorithm 1 模拟机场出租车算法

Input: 参数 α 、 β 和阈值 r_0

Output: 出租车的单位时间平均收入的方差 σ^2

```

1: while 剩余的出租车数量  $\geq 0$  do
2:   if 下一辆出租车的优先级为高 then
3:     加入优先车道
4:   else
5:     加入普通车道
6:   end if
7:   按照  $\beta : 1$  比例从车道内选出一辆车
8:    $d \leftarrow N(\mu, \sigma^2)$ 
9:   if  $d > d_0$  and  $choose = True$  then
10:     $r \leftarrow f(d, t_w)$ 
11:    if  $r \leq r_0$  then
12:      优先级为高
13:    else
14:      优先级为低
15:    end if
16:  else
17:     $\overline{E_{ai}} \leftarrow \frac{1}{T_{total}} \sum_j E_j(d)$ 
18:  end if
19: end while
20:  $\sigma^2 \leftarrow var(\overline{E_a})$ 

```

使用该算法，可以求得在不同参数条件下的方差 σ^2 ，要调整参数使方差最小化，可以使用如下的几种全局优化算法来快速求得条件范围内的最小值。

(1) 遗传算法。

遗传算法 (Genetic Algorithms, GA) 代表了一种用于了解多个参数关于模型的结果的相关因素的方法，而任何东西都可以用作遗传算法的数据。包括数字、文字、图像等。在遗传算法的进化计算的过程中，模拟生物的基因变异或交换的过程，允许那些适应度高的的基因进一步繁殖，适应度低的基因则死亡 [5]。

通过遗传算法，我们可以很轻松地计算出多参数的连续函数在指定范围内的最优解。我们构建了如下的有限范围内的优化问题，用于遗传算法的计算：

$$\begin{aligned} \min \quad & \sigma^2(\overline{E_a}) \\ \text{s.t.} \quad & 1 \leq \alpha \leq 100 \\ & 1000 \leq r_0 \leq 10000 \\ & 1 \leq \beta \leq 10 \end{aligned} \tag{28}$$

设定其它参数（详见附录中程序数据）后，另外将上述范围输入遗传算法运行，最终计算的方差的最小值： $\sigma_{min}^2 = 0.0272$ ，此时 $\alpha = 99.8, r_0 = 2421.8, \beta = 2.8$ 。

(2) 模拟退火算法。

模拟退火算法 (Simulated Annealing, SA) 是一种局部搜索算法的扩展。它模拟了金属在退火过程中的能量变化，在每一次迭代的过程中以一定的机率选择相邻区域能量较大的状态 [6]。它从理论上而言是一种全局最优算法，能够快捷地找到一个多元函数的全局最小值或最大值点。

设定如遗传算法中的参数和数据范围，输入模拟退火算法进行计算，最终计算的方差的最小值： $\sigma_{min}^2 = 0.0281$ ，此时 $\alpha = 99.9, r_0 = 4898.1, \beta = 9.2$ 。

(3) 粒子群算法。

粒子群优化算法 (Particle Swarm optimization, PSO) 是通过模拟鸟群的觅食行为，形成的一种基于群体协作原理的随机搜索算法。在 PSO 算法中，所有的粒子都有一个适应值和速度，通过不断迭代，跟踪个体极值和全局极值，来找到全局的最优解 [7]。

设定如遗传算法中的参数和数据范围，输入粒子群算法进行计算，最终计算的方差的最小值： $\sigma_{min}^2 = 0.0271$ ，此时 $\alpha = 100.0, r_0 = 2420.8, \beta = 1.0$ 。

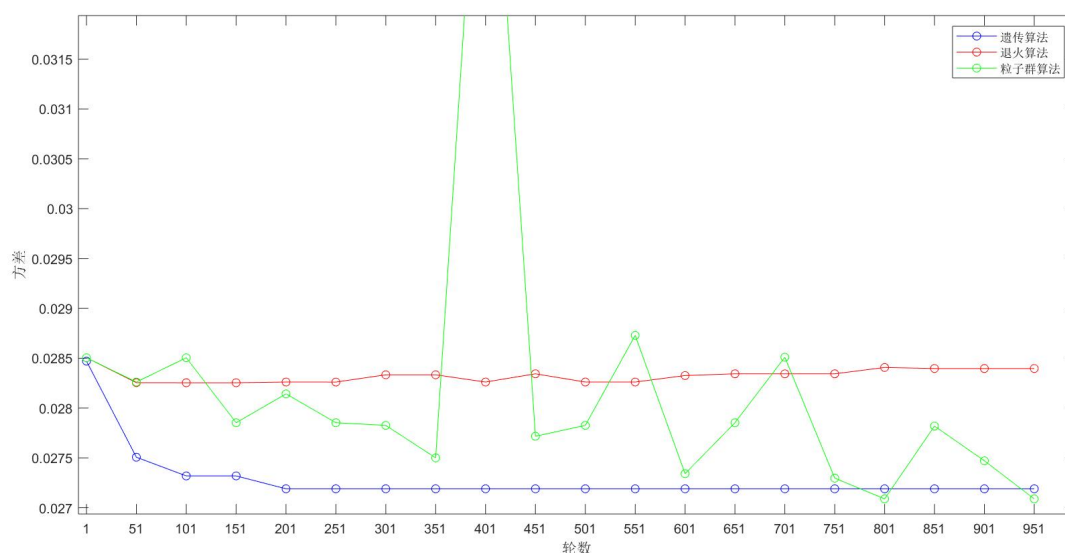


图 8: 三种优化算法的比较

图为以上三种算法收敛的过程，通过比较以上三种全局优化算法，可以看出粒子群算法对本模型最为适用。收敛速度快，且得到的方差结果最小。

综上所述，对于本模型假定的其余参数，当模型的三个主要参数为 $\sigma_{min}^2 = 0.0271, \alpha = 100.0, r_0 = 2420.8, \beta = 1.0$ 时，出租车司机的收益最为均衡。

7 模型总结

7.1 灵敏度分析

我们对出租车司机决策模型做出分析。对于不同的机场，其蓄车池的容量可能有较大的差距，而即使是对同一座机场，不同的时间也会有不同的航班到达间隔。而考虑到出租车司机在获取这两个信息时都可能存在一定的偏差，因此，有必要对该模型对蓄车池的容量和航班到达间隔的变化的灵敏度做出分析。

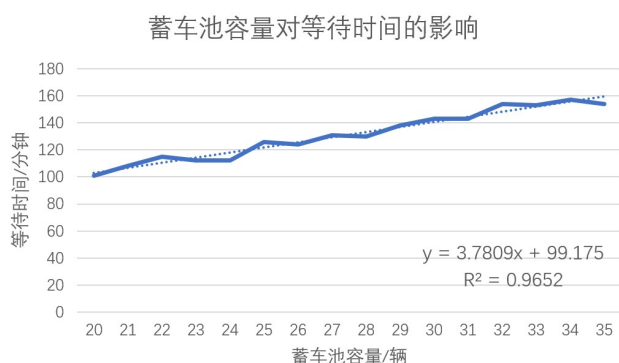


图 9: 蓄车池容量对等待时间的影响

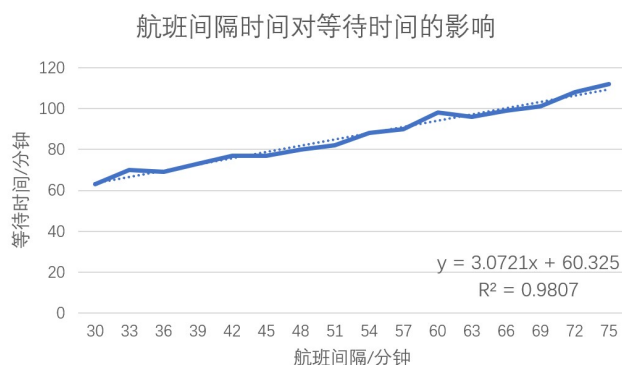


图 10: 航班间隔时间对等待时间的影响

从模型计算得出的数据和图9、图10容易看出，蓄车池容量的变化对等待时间的影响，比航班间隔变化对等待时间的影响更大。也即等待时间对蓄车池的容量更加敏感。因此对于司机来说，正确的判

断蓄车池内此时正在等候的汽车数目，更加有利于其做出正确的判断。这也与实际情况符合较好：一般的停车场都会用电子显示屏显示出此时停车场内的容量，而航班到达间隔这个信息要稍加难获取一些。

7.2 模型优点

1. 模型在建立时经历了较为广泛的调研，考虑了很多不同的实际情况，在设置不同的参量时使用了大量来源于实际生活的数据，增强了模型的普适性以及实际意义。
2. 在建立模型时，我们使用了计算机编程的方法，对每个模型的过程都进行了仿真，将预测得到的数据和实际数据进行了对比并将对比结果进行了可视化表示，提高了结果的准确度，也使得结果更加直观。
3. 在解决问题二的时候，我们采用了网络爬虫技术收集了一座机场的数据，得到了大量真实且精确的数据，也避免了把人力无谓地浪费在实地收集数据上。
4. 在解决问题四时，我们采用了遗传算法来对模型中的参数进行优化，使参数的选择更加合理。

7.3 模型缺点

1. 由于本题涉及到的实际情景非常复杂，可能影响结果的变量有很多，我们选择性地忽略了其中一些对结果影响不大的变量作为近似，比如问题三中隧道和车辆的长度等问题。
2. 由于模型考虑了较多的实际情况，导致有些过程略显繁琐。

参考文献

- [1] 何选森. 随机过程与排队论[M]. 长沙: 湖南大学出版社, 2010.
- [2] 岳喜展. 机场陆侧交通需求预测及集散道路方案设计[D]. 长春: 吉林大学, 1994.
- [3] 公交 II 行业: 出租车深度报告[R]. 广州: 广发证券, 2015.
- [4] 郑州机场陆侧交通需求预测及交通组织方案研究[C]//北京: 中国城市规划学会城市交通规划学术委员会: 中国城市规划学会, 9.
- [5] M L R. An introduction to genetic algorithms[J]. Information Processing Management, 1997, 33 (3):407.
- [6] 陈华根, 吴健生, 王家林, 等. 模拟退火算法机理研究[J]. 同济大学学报(自然科学版), 2004, 32(6): 802-805.
- [7] 王旖旎. 粒子群算法综述[J]. 网络导报·在线教育, 2012.

A 模型代码

A.1 出租车等待时间计算代码

```

1      %% 设置导入选项
2      opts = spreadsheetImportOptions("NumVariables", 5);
3
4      % 指定工作表和范围
5      opts.Sheet = "data";
6      opts.DataRange = "A2:E1232";
7
8      % 指定列名称和类型
9      opts.VariableNames = ["VarName1", "VarName2", "VarName3", "VarName4", "VarName5"];
10     opts.SelectedVariableNames = ["VarName1", "VarName2", "VarName3", "VarName4", ...
        "VarName5"];
11     opts.VariableTypes = ["categorical", "double", "double", "double", "double"];
12     opts = setvaropts(opts, 1, "EmptyFieldRule", "auto");
13
14     % 导入数据
15     tbl = readtable("data.xlsx", opts, "UseExcel", false);
16
17     %% 转换为输出类型
18     VarName1 = tbl.VarName1;
19     VarName2 = tbl.VarName2;
20     VarName3 = tbl.VarName3;
21     VarName4 = tbl.VarName4;
22     VarName5 = tbl.VarName5;
23
24     %% 清除临时变量
25     clear opts tbl
26
27
28     %% 设置导入选项
29     opts = spreadsheetImportOptions("NumVariables", 3);
30
31     % 指定工作表和范围
32     opts.Sheet = "Sheet1";
33     opts.DataRange = "A1:C242";
34
35     % 指定列名称和类型
36     opts.VariableNames = ["VarName1", "VarName2", "VarName3"];
37     opts.SelectedVariableNames = ["VarName1", "VarName2", "VarName3"];
38     opts.VariableTypes = ["double", "double", "double"];
39
40     % 导入数据
41     data2 = readtable("data2.xlsx", opts, "UseExcel", false);
42

```

```

43 %% 转化为输出类型
44 VarName6=data2.VarName1;
45 VarName7=data2.VarName2;
46 VarName8=data2.VarName3;
47 %% 清除临时变量
48 clear opts
49
50
51 %% 导入每分钟的航班
52 flights=zeros(1440,1);
53 for i=2:242
54     if VarName6(i,1)≥18&&VarName6(i,1)≤23
55         tt=(VarName6(i,1)-18)*60+VarName7(i,1);
56     else
57         tt=(6+VarName6(i,1))*60+VarName7(i,1);
58     end
59     if tt≠0
60         flights(tt,1)=flights(tt,1)+1;
61     end
62 end
63
64 %% 生成等待时间模型
65 x1=30:3:75;
66 y1=20:1:35;
67 z1=zeros(16,16);
68 x2=[]; x3=[]; y2=[]; y3=[];
69 ii=1;
70 for i=x1
71     jj=0;
72     for j=y1
73         total=0;
74         jj=jj+1;
75         for k=1:20
76             total=total+waittime(1,i,j);
77         end
78         z1(ii,jj)=floor(total/20);
79         %if floor(total/20)≥90
80         %     x2=[x2,i];
81         %     y2=[y2,j];
82         %else
83         %     x3=[x3,i];
84         %     y3=[y3,j];
85         %end
86     end
87     ii=ii+1;
88 end
89 figure
90 xx=linspace(30,120,100);
91 yy=linspace(20,50,100);
92 zz=griddata(x1,y1,z1,xx,yy','cubic');
93 surf(xx,yy,zz);

```

```

94 xlabel('航班间隔时间');
95 ylabel('蓄车池数量');
96 zlabel('等待时间');
97
98
99
100 %% 生成误差模型
101 wucha=zeros(1231,1);
102 wucha2=zeros(1231,1);
103 lilun=zeros(1231,1);
104 shiji=zeros(1231,1);
105 %x1=[]; x2=[]; y1=[]; y2=[];
106 for i=1:1000
107     num=0;
108     for j=i:i+120
109         if flights(j,1)≠0
110             num=num+1;
111         end
112     end
113     if num==0
114         continue;
115     end
116     tmp=waittime(num,120,VarName3(i,1));
117     tmp2=i;
118     ww=0;
119     numofleave=0;
120     while numofleave≤VarName3(i,1)&&VarName5(tmp2+30,1)≤VarName3(i,1)-numofleave
121         tmp2=tmp2+30;
122         ww=ww+30;
123         numofleave=numofleave+VarName5(tmp2,1);
124     end
125     if VarName5(tmp2+30,1)>VarName3(i,1)-numofleave
126         ww=ww+(VarName3(i,1)-numofleave)/VarName5(tmp2+30,1)*30;
127     end
128     ww=round(ww);
129     if ww≥90
130         x1=[x1,floor(120/num)];
131         y1=[y1,VarName3(i,1)];
132     else
133         x2=[x2,floor(120/num)];
134         y2=[y2,VarName3(i,1)];
135     end
136     wucha(i,1)=(tmp-ww);
137     wucha2(i,1)=abs(tmp-ww)/ww;
138     lilun(i,1)=tmp;
139     shiji(i,1)=ww;
140 end
141 x1=x1(1:200);
142 y1=y1(1:200);
143 x2=x2(1:200);
144 y2=y2(1:200);

```

```

145 anss=wucha2(1:130,:);
146 xd=mean(anss);

```

A.2 出租车等待时间计算代码子函数

```

1     function f = waittime(inputArg1,inputArg2,inputArg3)
2     N=inputArg1;      %未来一段时间内来的航班数
3     T=inputArg2;      %未来一段时间的持续时间minutes
4     A=floor(T/N);     %minutes;航班每隔多少时间来一趟
5     B=inputArg3;      %蓄车池个数
6     t0=1;             %每个人上车需要1minutes
7     Δ=25;             %每来一个航班人走到出租车点的平均时间
8     t=0;              %time
9     C=28;             %航班人数来乘车的平均数
10    sigma=6;          %方差
11    arrive=zeros(1000,1);
12    leave=zeros(1000,1);
13    currentnum=0;
14    while currentnum≤B
15        t=t+1;
16        if(mod(t,A)==0)
17            n=random('poisson',C);
18            currentnum=currentnum+n;
19        else
20            n=0;
21        end
22        for i=1:n
23            time=floor(normrnd(t+Δ,sigma));
24            while time<t
25                time=floor(normrnd(t+Δ,sigma));
26            end
27            arrive(time,1)=arrive(time,1)+1;
28        end
29    end
30    currentnum=0;
31    t=0;
32    while 1
33        t=t+1;
34        if arrive(t,1)≠0
35            break;
36        end
37    end
38    i=t;
39    while currentnum≤B
40        if arrive(i,1)≠0
41            if arrive(i,1)≤4
42                num=round(rand()*arrive(i,1));
43                while num==0
44                    num=round(rand()*arrive(i,1));

```

```

45         end
46     else
47         num=round(rand()*4);
48         while num==0
49             num=round(rand()*4);
50         end
51     end
52     arrive(i,1)=arrive(i,1)-num;
53     currentnum=currentnum+num;
54     if i<t
55         leave(t+t0,1)=leave(t+t0,1)+num;
56         t=t+t0;
57     else
58         leave(i+t0,1)=leave(i+t0,1)+num;
59         t=i+t0;
60     end
61 else
62     i=i+1;
63 end
64 end
65 f=t;
66 end

```

A.3 出租车费用计算代码

```

1     function f = fee(distance)
2     distance=distance/1000;
3     if distance<=3
4         f=14;
5     else
6         if distance<=15
7             f=14+(distance-3)*2.5;
8         else
9             f=14+30+(distance-15)*3.6;
10        end
11    end
12 end

```

A.4 计算优先权的函数

```

1     function f = fun1(waitime,shouyi,afa)
2     if shouyi==0
3         f=-waitime;
4     else
5         f= -waitime+shouyi*afa;
6     end

```

A.5 模拟基于优先权出租车调度代码

```

1      function fangcha=thepath(x)
2  global oo;
3  global resukts;
4  afa=x(1);
5  r0=x(2);
6  bili=x(3);
7  oo=oo+1;
8  %% 设置导入选项
9  opts = spreadsheetImportOptions("NumVariables", 2);
10
11 % 指定工作表和范围
12 opts.Sheet = "Sheet1";
13 opts.DataRange = "A2:B5000";
14
15 % 指定列名称和类型
16 opts.VariableNames = ["VarName1", "VarName2"];
17 opts.SelectedVariableNames = ["VarName1", "VarName2"];
18 opts.VariableTypes = ["double", "double"];
19
20 % 导入数据
21 tbl = readtable("suijibiao.xlsx", opts, "UseExcel", false);
22
23 %% 转换为输出类型
24 suijishu1 = tbl.VarName1;
25 suijishu2 = tbl.VarName2;
26
27 %% 清除临时变量
28 clear opts tbl
29
30
31 %%
32 allnum=500; %汽车总数
33 v=250; %平均车速30km/h 250m/min
34 youfei=0.7; %油费
35 fenjie=10000; %分界线 长短距离 训练 m
36 s1=1:allnum; %车排队序列 低优先级 下标代表目前所在位 数值代表车的标号
37 s2=[]; %高优先级
38 waitime=zeros(allnum,1); %车目前等待时间 下标代表车编号
39 avevas([1:allnum],1)=aveva; %车目前平均收益 下标代表车标号
40 leave=0; %不会再回来车的数量
41 curnum=0; %目前被超车的次数
42 times=zeros(allnum,1); %时间事件表 若值不为0, 即为该时刻有车来排队
43 allwaitime=zeros(allnum,1); %每一辆车 总等待时间
44 alldistance=zeros(allnum,1); %每一辆车 总路程
45 results=zeros(allnum,1); %结果 最后的总收益

```



```

46 t=0; %计时
47 backto=zeros(allnum,1); %回来的次数
48 sigma=3000; %标准差
49 mu=10000; %平均值
50 MAX=9999;
51 zz=1;
52 zzz=1;
53 while leave<allnum
54     t=t+1; %时间加1
55
56     % 所有正在等待车时间加一
57     for i=s1
58         if i≠0
59             waitime(i,1)=waitime(i,1)+1;
60             allwaitime(i,1)=allwaitime(i,1)+1;
61         end
62     end
63     for i=s2
64         if i≠0
65             waitime(i,1)=waitime(i,1)+1;
66             allwaitime(i,1)=allwaitime(i,1)+1;
67         end
68     end
69
70     % 将返程车加入排队队列中
71     for j=1:allnum
72         if times(j,1)==t %j为返程的车
73             if fun1(waitime(j,1),results(j,1),afa)<r0
74                 s2=[s2,j];
75                 backto(j,1)=fun1(waitime(j,1),results(j,1),afa);
76             else
77                 s1=[s1,j];
78                 backto(j,1)=fun1(waitime(j,1),results(j,1),afa);
79             end
80             waitime(j,1)=0;
81
82         end
83     end
84
85     if length(s1)+length(s2)==0
86         continue;
87     end
88
89     % 从排队序列选出下一次走的车
90     if ~isempty(s2)&&(curnum<bili || isempty(s1))
91         curno=s2(1,1);
92         curnum=curnum+1;
93         if length(s2)>1
94             s2=s2(2:end);
95         else
96             s2=[];

```

```

97         end
98     else
99         curno=s1(1,1);
100         curnum=0;
101         if length(s1)>1
102             s1=s1(2:end);
103         else
104             s1=[];
105         end
106     end
107
108     % 获得随机路程
109     tmp=suijishu1(zz);
110     zz=zz+1;
111     while tmp≤0
112         tmp=suijishu1(zz);
113         zz=zz+1;
114     end
115     if tmp≥fenjie
116         results(curno,1)=results(curno,1)+fee(tmp)-tmp/1000*youfei;
117         alldistance(curno,1)=alldistance(curno,1)+tmp;
118         leave=leave+1;
119     else
120         results(curno,1)=results(curno,1)+fee(tmp)-tmp/1000*youfei;
121         alldistance(curno,1)=alldistance(curno,1)+tmp;
122         therand=suijishu2(zzz);
123         zzz=zzz+1;
124         if therand≥0.5
125             results(curno,1)=results(curno,1)-tmp/1000*youfei;
126             alldistance(curno,1)=alldistance(curno,1)+tmp;
127             times(curno)=round(t+tmp/v*2);
128         else
129             leave=leave+1;
130         end
131     end
132 end
133
134 for i=1:allnum
135     allwaittime(i,1)=allwaittime(i,1)-i+1;
136     avevas(i,1)=results(i,1)/(allwaittime(i,1)+alldistance(i,1)/v);
137 end
138 fangcha=var(avevas);
139 resukts(oo,1)=fangcha;
140 end

```