

赛区评阅编号（由赛区组委会填写）：

---

## 2020 高教社杯全国大学生数学建模竞赛

### 承 诺 书

我们仔细阅读了《全国大学生数学建模竞赛章程》和《全国大学生数学建模竞赛参赛规则》（以下简称为“竞赛章程和参赛规则”，可从全国大学生数学建模竞赛网站下载）。

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上 QQ 群、微信群等）与队外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛章程和参赛规则的，如果引用别人的成果或资料（包括网上资料），必须按照规定的参考文献的表述方式列出，并在正文引用处予以标注。在网上交流和下载他人的论文是严重违规违纪行为。

**我们以中国大学生名誉和诚信郑重承诺，严格遵守竞赛章程和参赛规则，以保证竞赛的公正、公平性。如有违反竞赛章程和参赛规则的行为，我们将受到严肃处理。**

我们授权全国大学生数学建模竞赛组委会，可将我们的论文以任何形式进行公开展示（包括进行网上公示，在书籍、期刊和其他媒体进行正式或非正式发表等）。

我们参赛选择的题号（从 A/B/C/D 中选择一项填写）：   C  

我们的报名参赛队号（12 位数字全国统一编号）：   暂无  

参赛学校（完整的学校全称，不含院系名）：   华中科技大学  

参赛队员（打印并签名）：1.   鲁镇仪  

2.   李欣航  

3.   蒋瀚锐  

指导教师或指导教师组负责人（打印并签名）：   暂无  

（指导教师签名意味着对参赛队的行为和论文的真实性负责）

日期：   2020   年   08   月   19   日

（请勿改动此页内容和格式。此承诺书打印签名后作为纸质论文的封面，注意电子版论文中不得出现此页。以上内容请仔细核对，如填写错误，论文可能被取消评奖资格。）

赛区评阅编号（由赛区组委会填写）：

---

**2020 高教社杯全国大学生数学建模竞赛**

**编 号 专 用 页**

赛区评阅记录（可供赛区评阅时使用）：

评 阅 人						
备 注						

送全国评阅统一编号（由赛区组委会填写）：

全国评阅随机编号（由全国组委会填写）：

（请勿改动此页内容和格式。此编号专用页仅供赛区和全国评阅使用，参赛队打印后装订到纸质论文的第二页上。注意电子版论文中不得出现此页。）

# 机场的出租车问题

## 摘要

现代社会中，人们在日常出行方式当中必然少不了出租车这个选项。如何让乘客和司机利益均得到保障是我们研究的问题。该文主要通过对宏观和微观两种情况建立数学模型，来得到有关出租车司机该如何进行决策的方法和机场管理部门应如何设置接车方式和对短途载客并再次返回的出租车给予“优先权”的具体解决方案。

对于问题一，该文主要建立出宏观地图与司机等待模型，利用题目已知条件知道司机能看到的具體等待时间  $t_{\text{wait}}$  和计价方式还有出租车的平均收益  $a$ 。用分段函数方式来求出了计价方式  $f$  的具体表达式。求出两种收益分别的表达式之后，直接比较这两个表达式，即可得到结果。

对于问题二，我们选取郑州机场的目前的数据，有关于这部分数据均用爬虫取自于网上。利用过去半个小时之内离开飞机场和进入飞机场的车辆数，我们能够得到司机的平均等待时间  $t_{\text{wait}}$  的具体值，再利用郑州的出租车营业方案，得到计价方式  $f$  的具体表达式，这时候直接利用第一问的表达式即可完成此题。

对于问题三，

对于问题四，

**关键字：** 元胞自动机    梯度下降

## 目录

一、问题重述 .....	3
1.1 问题背景 .....	3
1.2 问题的提出 .....	3
二、问题分析 .....	5
2.1 问题一的分析 .....	5
2.2 问题二的分析 .....	5
2.3 问题三的分析 .....	5
2.4 问题四的分析 .....	5
三、模型假设 .....	6
四、符号系统 .....	7
五、建模与求解 .....	8
5.1 出租车的计价模型 .....	8
5.2 问题一的建模与求解 .....	9
5.2.1 模型准备 .....	9
5.2.2 模型的建立 .....	9
5.3 问题二的建模与求解 .....	11
5.3.1 模型的建立 .....	11
5.3.2 模型的求解 .....	11
5.4 问题三的建模与求解 .....	12
5.4.1 模型的建立 .....	12
5.4.2 模型的求解 .....	12
5.5 问题四的建模与求解 .....	13
5.5.1 模型的建立 .....	13
5.5.2 模型的求解 .....	13
参考文献 .....	14
附录 A 爬取郑州机场出租车管理站代码—C-sharp 源代码 .....	15
附录 B 爬取郑州机场航班信息代码—JavaScript 源代码 .....	17

# 一、问题重述

## 1.1 问题背景

由于出租车的选择以及机场管理者所作出的决策对效率，时间，金钱均有着重大的影响。因此，对于机场管理者来说，对于加快机场外交通运营，增加司机收入，提高乘客舒适度，节约出行时间来说，提高机场出租车运营效率是具有重要意义的。

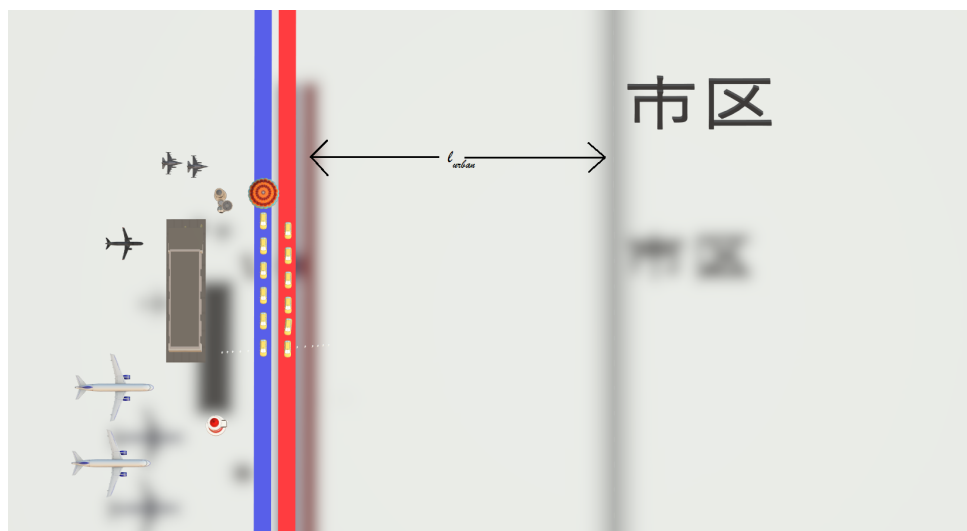


图 1 市区，机场与排队等车模型示意图

## 1.2 问题的提出

该文章主要是研究不同情况下司机该如何来选择最优的方式和机场管理者该如何调整政策使得宏观效率最高。由于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，这导致司机有一个“排队时间成本”，如果不排队接客，则多出的时间就可以到市区接客。但若直接去市区接客，则从机场到市区中间一段距离就是空置的。

当然，为了解决这个问题，除了司机自身的判断之外，很重要的另外一点就是机场管理者的政策。机场管理者设定不同的上下车点，司机接客的方式，对短途载客再次返回的出租车给予一定的“优先权”，均会影响到具体的效益以及收入情况。

在问题一当中，题设较为简单，只需要建立出租车司机选择决策模型，并给出司机的选择策略即可，所以我们利用题目所给出的司机已知的条件。也即大致的出租车司机排队等待时间加服务时间  $t_{wait}$  和出租车的计价方式  $f$  还有具体的城市到飞机场的距离  $l_{urban}$  第一小问希望我们利用这些条件，能够得到最终司机分别在哪一种情况下选择哪一种方案的解析解。

在问题二当中，题目直接要求对于一个具体城市的具体司机的具体选择，这需要考

察网上的信息搜索能力，在第一问已经得到抽象的解析解的时候，题目要求我们只需要把具体数值代入表达式当中得到具有现实参考意义的解。

在问题三当中，

在问题四当中，

## 二、问题分析

### 2.1 问题一的分析

问题一较为简单，因此我们只需要拟定各个不同的定义所代表的含义，目的在于给司机选定到底选择哪一个哪一种方案收益最高。虽然本问并没有给出具体数据，但我们仍然可以把一些量用字母替代。

考虑到司机是能够观察到具体还有多少乘客和出租车的，所以我们直接能够假定司机知道还需要等待多久的时间，这是合理的。由于出租车司机的收入司机也是知道的，所以可以假定市区出租车的平均收入。

通过这些条件，我们选择是否留在机场接客还是直接回市中心是通过比较两者的平均收益得到的。

这时候利用之前得到的出租车收益图函数我们即可得到最终目标的解析解。

### 2.2 问题二的分析

### 2.3 问题三的分析

### 2.4 问题四的分析

### 三、 模型假设

- 认为城市内和城市外行车速度恒定。
- 认为在路上的时候没有堵车情况发生，不考虑出租车关于时间的计费。
- 前两问假设服务时间为定值。
- 假设一辆车只载送一位乘客。
- 假设空载费用损失为油费。
- 假设一个周期内出租车和乘客一一抵消。



#### 四、符号系统

符号	意义
$f$	出租车的收益函数
$t_{\text{wait}}$	出租车排队等待时间加服务时间
$m_{\text{start}}$	出租车起步价
$m_{\text{oil}}$	每一公里出租车的油费价格
$l_{\text{init}}$	开始按里程计价路程
$k$	每公里计费价格
$m_{\text{night}}$	夜价提升价格量
$l_{\text{urban}}$	机场到市区的平均距离
$v_{\text{out}}$	从机场到市区当中的郊区行车速度
$m_{\text{aver}}$	在市区当中的平均收入
$t_{\text{city}}$	司机在市区中平均在每隔多久能接到到机场的客

注：未列出及重复符号以出现处为准

## 五、建模与求解

### 5.1 出租车的计价模型

对于一般而言城市的出租车，当是同一种车型时，相同情况下的成本是相同的。每一公里为  $m_{\text{oil}}$  的油费价格。

并且，仍然是对于同一种车型的出租车来说，一般有几种计费方式混合使用，不同情况下对应这不一样的计费方式。在这里我们选取最常用的计费方式。

假设出租车起步价为  $m_{\text{start}}$ ，在  $l_{\text{init}}$  公里后开始计价，每一公里计费  $k$  元。夜价则在日价的起步价基础上再次增加  $m_{\text{night}}$  的收益。

则白天时前  $l_{\text{init}}$  公里的收益为： $f(l) = m_{\text{start}} - l \cdot m_{\text{oil}}$  元。凌晨时前  $l_{\text{init}}$  公里的收益为： $f(l) = m_{\text{start}} + m_{\text{night}} - l \cdot m_{\text{oil}}$  元。

当过了前  $l_{\text{init}}$  公里时，假设已经开了  $l$  公里的路程，那么则白天时的收益为：

$$\begin{aligned} f(l) &= m_{\text{start}} - l_{\text{init}} \cdot m_{\text{oil}} + (l - l_{\text{init}}) \cdot (k - m_{\text{oil}}) \\ &= m_{\text{start}} + l \cdot k - l \cdot m_{\text{oil}} - l_{\text{init}} \cdot k \quad (l \geq l_{\text{init}}) \end{aligned}$$

凌晨的收益则是白天时的收益  $+m_{\text{night}}$ ，也即：

$$f(l) = m_{\text{start}} + m_{\text{night}} + l \cdot k - l \cdot m_{\text{oil}} - l_{\text{init}} \cdot k \quad (l \geq l_{\text{init}})$$

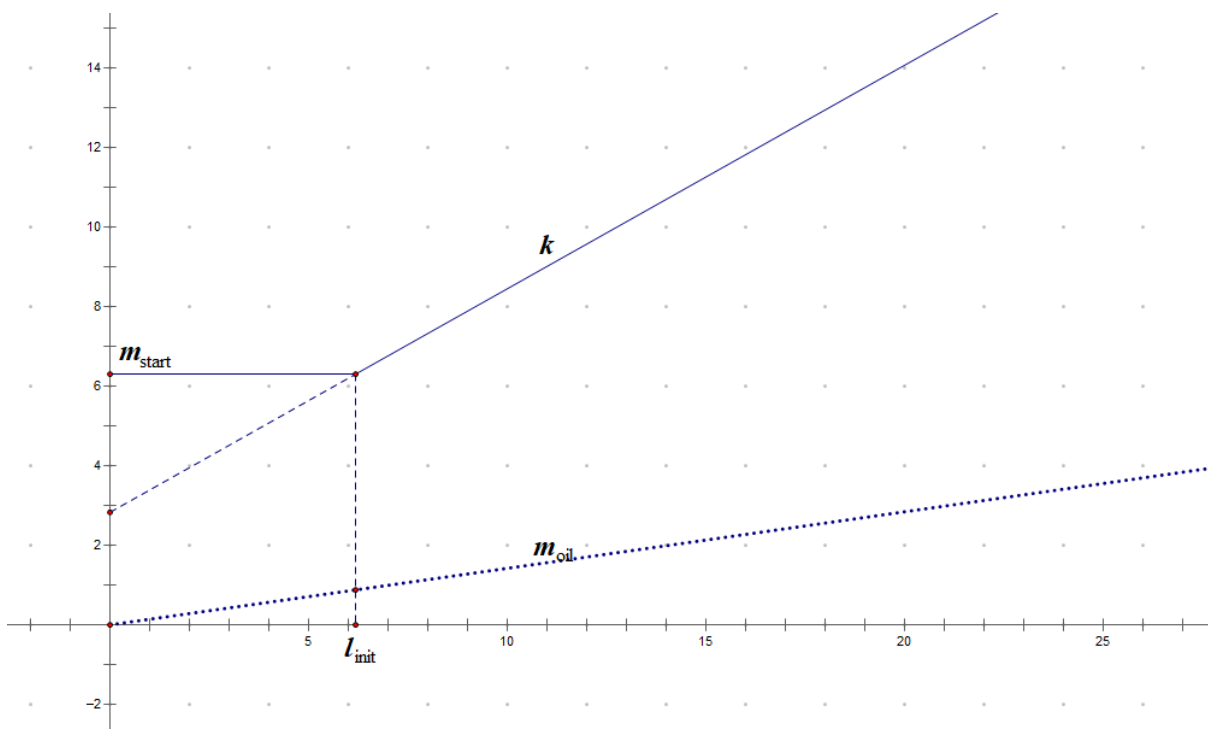


图2 出租车的收益函数图示

## 5.2 问题一的建模与求解

### 5.2.1 模型准备

先明确一直等待载客之后返回市区和直接返回市区拉客这两种情况分别的平均收益该如何计算。假定机场到市区的平均距离为  $l_{\text{urban}}$ ，在市区当中的平均收入记为每一小时平均  $m_{\text{aver}}$  元。在从机场到市区当中的郊区行车速度为  $v_{\text{out}}$ 。司机在市区当中接客的平时间为  $t_{\text{city}}$ 。

### 5.2.2 模型的建立

(1) 若司机选择就在机场接客，那么我们有关于时间的式子：

在机场时间  $t_{\text{wait}}$ ，在高速路上从机场到城市中心的时间为  $\frac{l_{\text{urban}}}{v_{\text{out}}}$ ，在城市当中的时间为  $t_{\text{city}}$  总时间

$$t_1 = t_{\text{wait}} + \frac{l_{\text{urban}}}{v_{\text{out}}} + t_{\text{city}}$$

机场载客的收入为

$$f(l_{\text{urban}}) = m_{\text{start}} - l_{\text{init}} \cdot m_{\text{oil}} + (l_{\text{urban}} - l_{\text{init}}) \cdot (k - m_{\text{oil}})$$

在市区的收入为： $a = m_{\text{aver}} \cdot t_{\text{city}}$  平均收入为：

$$a_1 = \frac{f(l_{\text{urban}}) + a}{t_1} = \frac{m_{\text{start}} - l_{\text{init}} \cdot m_{\text{oil}} + (l_{\text{urban}} - l_{\text{init}}) \cdot (k - m_{\text{oil}}) + m_{\text{aver}} \cdot t_{\text{city}}}{t_{\text{wait}} + \frac{l_{\text{urban}}}{v_{\text{out}}} + t_{\text{city}}}$$

(2) 若司机选择不接客，直接回市区找客人，那么我们仍然可以得到一个关于时间的式子：

这时候没有了在机场时间的出租车排队等待时间加服务时间了，同样，我们有高速路上从机场到城市中心的时间为  $\frac{l_{\text{urban}}}{v_{\text{out}}}$ ，在城市当中的时间为  $t_{\text{city}}$ 。总时间为：

$$t_2 = \frac{l_{\text{urban}}}{v_{\text{out}}} + t_{\text{city}}$$

那么我们只有在市区的收入  $a = m_{\text{aver}} \cdot t_{\text{city}}$ ，并且我们甚至有耗费在路上的油费  $l_{\text{urban}} \cdot m_{\text{oil}}$  所以我们有平均收入为：

$$a_2 = \frac{a - l_{\text{urban}} \cdot m_{\text{oil}}}{t_2} = \frac{m_{\text{aver}} \cdot t_{\text{city}} - l_{\text{urban}} \cdot m_{\text{oil}}}{\frac{l_{\text{urban}}}{v_{\text{out}}} + t_{\text{city}}}$$

我们只需比较这两个的大小，若  $a_1 > a_2$  就只需要解不等式：

$$\frac{m_{\text{start}} - l_{\text{init}} \cdot m_{\text{oil}} + (l_{\text{urban}} - l_{\text{init}}) \cdot (k - m_{\text{oil}}) + m_{\text{aver}} \cdot t_{\text{city}}}{t_{\text{wait}} + \frac{l_{\text{urban}}}{v_{\text{out}}} + t_{\text{city}}} > \frac{m_{\text{aver}} \cdot t_{\text{city}} - l_{\text{urban}} \cdot m_{\text{oil}}}{\frac{l_{\text{urban}}}{v_{\text{out}}} + t_{\text{city}}}$$

解得：

$$t_{\text{wait}} < \frac{\left(\frac{l_{\text{urban}}}{v_{\text{out}}} + t_{\text{city}}\right) \cdot (m_{\text{start}} + l_{\text{urban}} \cdot (k - m_{\text{oil}}) - k \cdot l_{\text{init}} + m_{\text{aver}} \cdot t_{\text{city}})}{m_{\text{aver}} \cdot t_{\text{city}} - l_{\text{urban}} \cdot m_{\text{oil}}} - \left(\frac{l_{\text{urban}}}{v_{\text{out}}} + t_{\text{city}}\right)$$

我们记这个特殊的时间点为  $t_0$ ，

$$\begin{aligned} t_0 &= \frac{\left(\frac{l_{\text{urban}}}{v_{\text{out}}} + t_{\text{city}}\right) \cdot (m_{\text{start}} + l_{\text{urban}} \cdot (k - m_{\text{oil}}) - k \cdot l_{\text{init}} + m_{\text{aver}} \cdot t_{\text{city}} - m_{\text{aver}} \cdot t_{\text{city}} + l_{\text{urban}} \cdot m_{\text{oil}})}{m_{\text{aver}} \cdot t_{\text{city}} - l_{\text{urban}} \cdot m_{\text{oil}}} \\ &= \frac{\left(\frac{l_{\text{urban}}}{v_{\text{out}}} + t_{\text{city}}\right) \cdot (m_{\text{start}} + l_{\text{urban}} \cdot k - k \cdot l_{\text{init}})}{m_{\text{aver}} \cdot t_{\text{city}} - l_{\text{urban}} \cdot m_{\text{oil}}} \end{aligned}$$

等待时间  $t_{\text{wait}}$  比  $t_0$  时间小时，我们就继续在机场进行等待，当等待时间  $t_{\text{wait}}$  比  $t_0$  大时，司机就得离开机场直接前往市中心接客更加划算。这是白天的情况。

当为凌晨时分是夜班的时候，用同样是上述过程可以算出

$$t_0 = \frac{\left(\frac{l_{\text{urban}}}{v_{\text{out}}} + t_{\text{city}}\right) \cdot (m_{\text{start}} + m_{\text{night}} + l_{\text{urban}} \cdot k - k \cdot l_{\text{init}})}{m_{\text{aver}} \cdot t_{\text{city}} - l_{\text{urban}} \cdot m_{\text{oil}}}$$

## 5.3 问题二的建模与求解

### 5.3.1 模型的建立

### 5.3.2 模型的求解

## 5.4 问题三的建模与求解

### 5.4.1 模型的建立

依据接待方式的不同，能最终划归成两种情况：

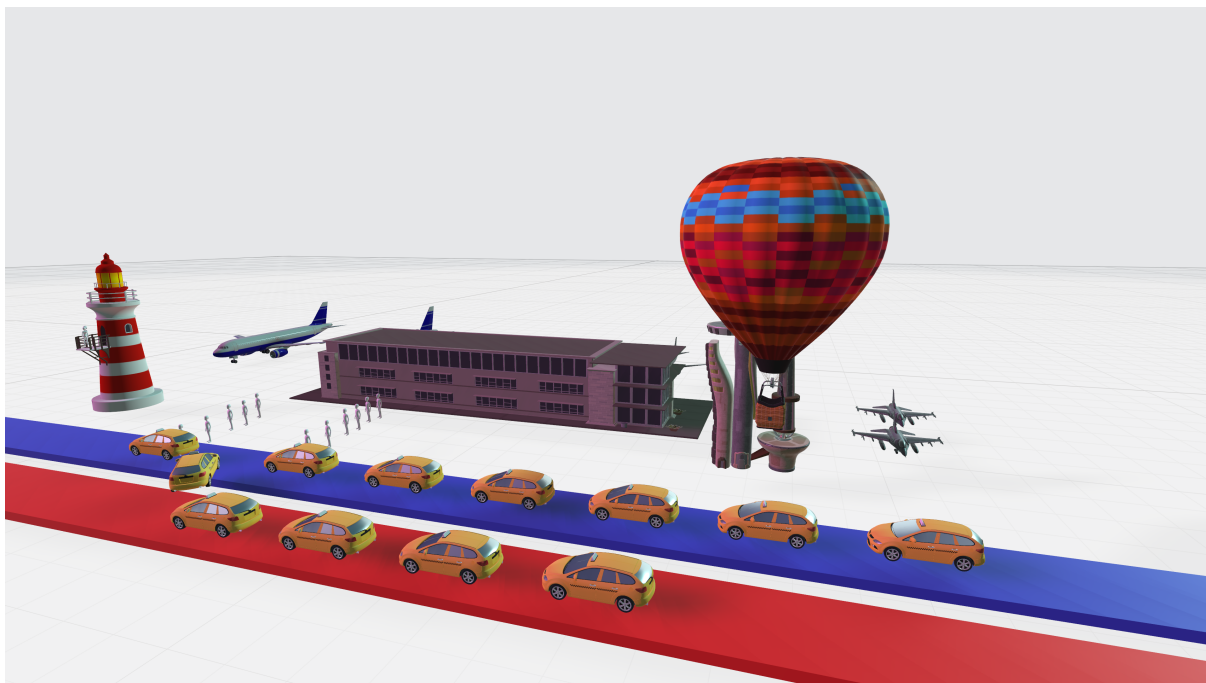


图 3 情况一

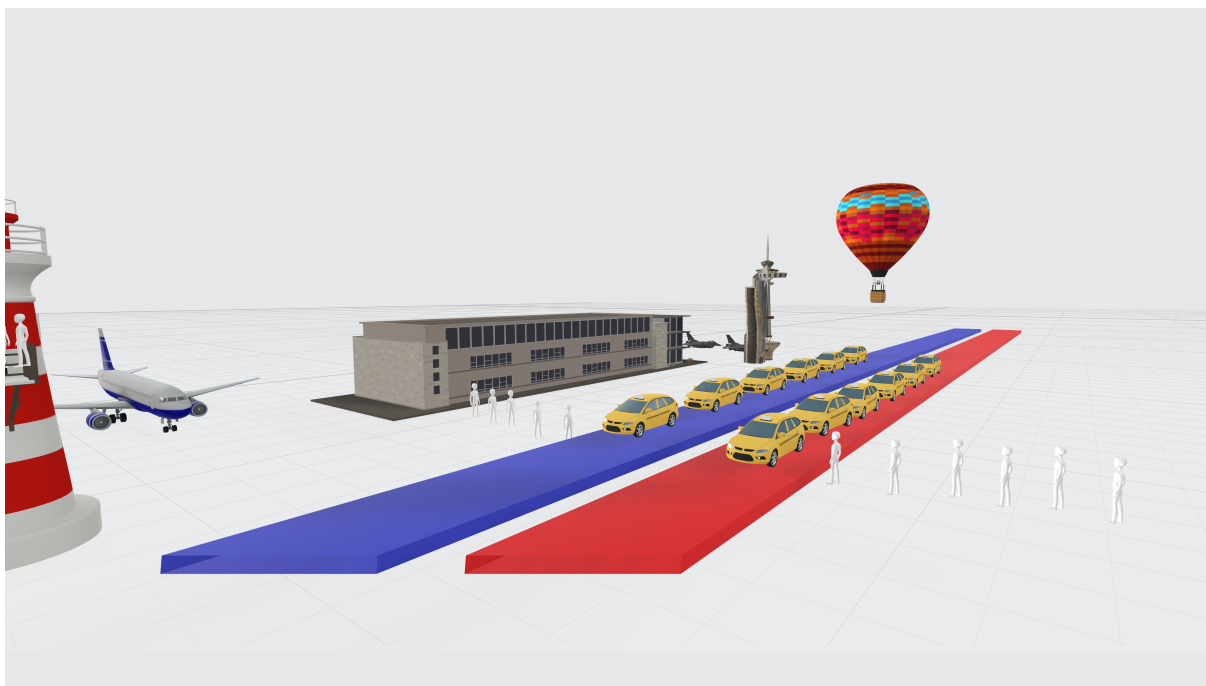


图 4 情况二

#### 5.4.2 模型的求解

## 5.5 问题四的建模与求解

### 5.5.1 模型的建立

### 5.5.2 模型的求解



## 参考文献

- [1] 参考文献

## 附录 A 爬取郑州机场出租车管理站代码—C-sharp 源代码

```
using System;
using System.IO;
using System.Net;
using System.Text.RegularExpressions;
using System.Threading;

namespace c_2019
{
    internal class ResourcePool
    {
        public static string HttpGet(string Url, string contentType)
        {
            try
            {
                string retString = string.Empty;

                HttpWebRequest request = (HttpWebRequest)WebRequest.Create(Url);
                request.Method = "GET";
                request.ContentType = contentType;

                HttpWebResponse response = (HttpWebResponse)request.GetResponse();
                Stream myResponseStream = response.GetResponseStream();
                StreamReader streamReader = new StreamReader(myResponseStream);
                retString = streamReader.ReadToEnd();
                streamReader.Close();
                myResponseStream.Close();
                return retString;
            }
            catch (Exception ex)
            {
                return ex.Message;
            }
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            string head = "time,numbers,enter,leave";
            Console.WriteLine(head);
            using (StreamWriter file = new StreamWriter(@"val.csv", true))
            {
                file.Write(head);
            }
        }
    }
}
```

```

    }
    for (; ; )
    {
        string ans = NetworkAirport.getContent();
        Console.WriteLine(ans);
        using (StreamWriter file = new StreamWriter(@"val.csv", true))
        {
            file.WriteLine(ans);
        }
        Thread.Sleep(5000);
    }
}

class NetworkAirport
{
    public static string getContent()
    {
        try
        {
            string ans = "";
            string head = "http://www.whalebj.com";
            string content = ResourcePool.HttpGet(head + "/xzjc/default.aspx", "");

            string pattern = "截止目前为止 ([0-9]*-[0-9]*-[0-9]* [0-9]*:[0-9]*:[0-9]*) ";
            MatchCollection mc = Regex.Matches(content, pattern);
            string time = mc[0].Value.Substring(7, 19);
            ans += time + ",";

            pattern = "场内待运车辆数为: [0-9]*";
            mc = Regex.Matches(content, pattern);
            string vehiclesNumColl = mc[0].Value.Substring(9);
            ans += vehiclesNumColl + ",";

            pattern = "前半小时进场车辆数为: [0-9]*";
            mc = Regex.Matches(content, pattern);
            vehiclesNumColl = mc[0].Value.Substring(11);
            ans += vehiclesNumColl + ",";

            pattern = "前半小时离场车辆数为: [0-9]*";
            mc = Regex.Matches(content, pattern);
            vehiclesNumColl = mc[0].Value.Substring(11);
            ans += vehiclesNumColl;

            return ans;
        }
        catch (Exception e)
    }
}

```

```

    {
        return "";
    }
}
}
}
}

```

## 附录 B 爬取郑州机场航班信息代码—JavaScript 源代码

```

console.log("载体","飞行","起源","状态","到来");

function getarrivals(){
    for (let i = 1; i < 9; i++) {
        setTime\mathrm{out}{}(() => {
            var timeperiod = document.getElementsByClassName("timeperiod");
            var timeperiodNode = timeperiod[0].childNodes
            var shouldClick = timeperiodNode[i * 2];
            shouldClick.click();
            setTime\mathrm{out}{}(() => {
                var arrivals = document.getElementById("arrivals");
                var nodeColl = arrivals.childNodes;
                for (let j = 0; j < nodeColl.length; j++) {
                    var subTempText = "";
                    var airData = nodeColl[j].childNodes;
                    for (let k = 0; k < airData.length - 1; k++) {
                        var text = airData[k].childNodes[0].textContent;
                        subTempText += "'" + text + "' + ',';
                    }
                    subTempText += "'" + airData[airData.length - 1].childNodes[0].textContent +
                        "'";
                    console.log(subTempText);
                }
            }, 5000);
        }, i * 6000);
    }
}

function getdepartures(){
    for (let i = 1; i < 9; i++) {
        setTime\mathrm{out}{}(() => {
            var timeperiod = document.getElementsByClassName("timeperiod");
            var timeperiodNode = timeperiod[0].childNodes
            var shouldClick = timeperiodNode[i * 2];
            shouldClick.click();
            setTime\mathrm{out}{}(() => {

```

```

var departures = document.getElementById("departures");
var nodeColl = departures.childNodes;
for (let j = 0; j < nodeColl.length; j++) {
    var subTempText = "";
    var airData = nodeColl[j].childNodes;
    for (let k = 0; k < airData.length - 1; k++) {
        var text = airData[k].childNodes[0].textContent;
        subTempText += "'" + text + "' + ',';
    }
    subTempText += "'" + airData[airData.length - 1].childNodes[0].textContent +
        "'";
    console.log(subTempText);
}
}, 5000);
}, i * 6000);
}
}

```