# “互联网+”时代的出租车资源配置


## 摘 要

本文围绕“互联网+”时代的出租车资源配置问题，建立了基于层次分析法的模糊综合评判模型，对不同时空出租车资源的“供求匹配”程度进行了分析；结合补贴前后“供求匹配”程度的变化，建立了出租车运营平衡模型，深入分析了补贴方案对“缓解打车难”的影响及影响原因；从服务平台和司机的效益最大化两方面考虑，建立了双目标规划模型，对新的补贴方案进行了求解，并对合理性进行了分析。

对于问题一，建立了基于层次分析法的模糊综合评判模型，对不同时空出租车资源的“供求匹配”程度进行了分析。首先定义出租车供需比、接单效率、里程利用率三个评价指标，然后运用层次分析法确定三个指标在供求匹配指数中的权重，接着对全国的各项数据进行抓取，根据数据区间分布确定隶属度矩阵，最后求解每个城市的三个指标值，根据最大隶属原则判断每个城市的“供求匹配”等级。得到了早晚高峰期比平峰期匹配程度低，南京、上海匹配程度为“优秀”，而北京、沈阳、保定、鞍山匹配程度为“中等”甚至“较差”的结论。

对于问题二，首先结合北京市在“滴滴”“快的”两家公司补贴前后三个阶段的统计数据，运用问题一中的模型对补贴前后“供求匹配”程度进行计算和对比，得到匹配等级从“中等”变为“良好”，所以补贴方案对“缓解打车难”有帮助，但不能完全解决“打车难”问题。为了进一步研究补贴方案如何影响“打车难”现象，应用概率的方法，建立了出租车运营平衡模型，对司机的选择情况进行仿真模拟，求解可知在服务平台建立后乘客平均等待时间从 11 分钟变为 7.6 分钟，明显缩短，因此可得“打车难”得到缓解的根本原因是补贴方案使打车软件在用户中广泛普及。

对于问题三，从服务平台和司机的效益最大化两方面考虑，建立了双目标规划模型，对新的补贴方案进行了求解。首先分析了打车软件产生的新问题，得到了不对乘客进行补贴的决策。然后结合出租车数量、乘客需求量等数据，分别得到服务平台和出租车司机的经济效益函数，以两者经济效益均最大为目标建立双目标规划模型。最后结合哈尔滨市的各项统计数据，求解得到最优补贴方案为每单补贴司机 5.85 元。将求得的补贴方案与实际方案作对比，发现数据较为可靠，且通过稳定性分析证明模型的稳定性较强，在两方面论证了补贴方案的合理性。



**关键词：**层次分析法　模糊综合评判　运营平衡模型　仿真模拟　双目标规划　稳定性分析

# 一、问题重述

出租车是市民出行的重要交通工具之一，“打车难”是人们关注的一个社会热点问题。随着“互联网+”时代的到来，有多家公司依托移动互联网建立了打车软件服务平台，实现了乘客与出租车司机之间的信息互通，同时推出了多种出租车的补贴方案。

请你们搜集相关数据，建立数学模型研究如下问题：

(1) 试建立合理的指标，并分析不同时空出租车资源的“供求匹配”程度。

(2) 分析各公司的出租车补贴方案是否对“缓解打车难”有帮助？

(3) 如果要创建一个新的打车软件服务平台，你们将设计什么样的补贴方案，并论证其合理性。

# 二、问题分析

## 2.1 问题一的分析

分析不同时空出租车资源的“供求匹配”程度，是一个典型的分析和评价的问题。分析与评价的模型有很多，包括包络分析法、层次分析法、模糊综合评价法等。它们各有优点也各有缺点，因为分析“匹配程度”需要考虑多方面因素，有定性的也有定量的，评价标准较为模糊，所以我们倾向于用模糊综合评判求解该问题。另外，因为模糊综合评判各项指标的权重是由专家据经验给出，因而不够客观。而层次分析法的权重是通过两两比较并经过一致性检验确定的，它很好地解决了这个问题，所以我们考虑采用基于层次分析法的模糊综合评判模型来对问题一进行求解。

为了科学系统客观地评价“供求匹配”程度，其相应的评价指标应遵循全面性、科学性、可操作性三个原则。即既要根据能搜索到的数据建立指标，又要全面反映供求匹配程度。而在时间方面，因为上下班高峰期会对出租车的需求量产生较大影响，所以应选择上下班高峰期和平峰期等不同的时间段；空间方面，为体现不同城市经济发展状况和人口密度对匹配程度的影响，也应分别选择一、二、三线代表城市进行全方位分析。

## 2.2 问题二的分析

要分析出租车补贴方案是否对“缓解打车难”有帮助，我们需要分别考虑补贴前和补贴后的供求匹配程度，将两者进行对比。因为问题一中我们要建立模糊综合评判模型，在问题二中我们可以用该模型分别计算补贴前和补贴后的“供求匹配”程度，并将它们进行对比。通过对比结果可以很直观的看出补贴方案是否对“缓解打车难”有帮助。

如果有帮助，我们需要再建立一个新的模型来深入探讨为什么会对“缓解打车难”有帮助。首先考虑到的是补贴方案的实行会使用户更广泛的使用打车软件，而打车软件的使用可能会在一定程度上缓解“打车难”的问题。

如果没有帮助，我们也需要再建立一个模型探讨没有帮助的原因。可以从出租车供需比、乘客等待时间等方面进行建模，比较补贴方案前后两者有无改变。

## 2.3 问题三的分析

两大打车软件服务平台“滴滴”和“快的”通过推出多种补贴方案，建立起大家对打车软件的使用习惯。所以如果要推出新的打车软件，我们的补贴方案目标应该不是缓解打车难，而要在平台收益、乘客满意度、司机收益最大化等方面进行分析。收益最大化，很明显这是一个规划问题，以收益最大化为目标，根据实际情况下的约束条件建立多目标规划模型，求解最优解就可作为补贴方案。

但补贴方案是否合理，还需要从多方面进行论证。比如该方案是否与实际相符，是否超出了平台支付能力，是否会造成新的问题，稳定性如何等等。

## 三、模型假设

1.信息来源可靠
2.出租车乘客的平均出行距离保持不变
3.城市道路交通状况保持不变，不发生堵车等理想情况
4.在司机在平台上接单之后，双方不会出现违约等诚信问题
5.不考虑出租车在接单之后出现故障

## 四、符号说明

$C_i$——某时刻该城市内各采样点乘客需要呼叫的出租车数量
$L$——平均乘车距离
$N$——出租车总量
$P$——出租车平均运价（元/次）
$Q$——出租车乘客需求（次/时）
$\bar{Q}$——乘客的潜在出行需求（次/小时）
$T$——乘客平均乘车时间（小时）
$W$——乘客上车前平均等待时间
$c$——每辆车单位时间经营总成本（元/（车·小时））
$c_0$——每辆车的单位时间固定成本（元/（车·小时））
$\lambda$——出租车平均单位里程耗油量（升/千米）
$x$——燃油价格（元/升）

## 五、模型的建立与求解

### 5.1 问题一的模型建立与求解

分析“供求匹配”程度，要从供求关系和匹配关系两方面建立指标；而对于不同时空，需要综合考虑时间、空间的区别，选取不同的样本点进行分析；结合不同的样本点数据，用综合评判的方法将得到的各指标值进行综合，根据综合评判等级定性描述不同时空出租车资源的“供求匹配”程度。

#### 5.1.1 模型建立

##### 5.1.1.1 指标及时间、空间的确定

（1）指标确定

为了科学系统客观地评价“供求匹配”程度，其相应的评价指标应遵循全面性、科学性、可操作性三个原则。即既要根据能搜索到的数据建立指标，又要全面反映供求匹配程度。根据我们所能搜集到的信息范围，分别建立了出租车供需比、接单效率和里程利用率三个指标：

①出租车供需比

为该城市内出租车供应量 $S$ 和出租车需求量 $C$ 之比，用 $\varphi$ 来表示。我们可以统计出某时刻该城市内各采样点可供呼叫的出租车数量，用 $S_i$ 来表示；以及某时刻该城市内各采样点乘客需要呼叫的出租车数量，用 $C_i$ 来表示。用 n 表示该城市内采样点的个数，则

$$\varphi = \frac{S}{C} = \frac{\sum_{i=1}^{n} S_i}{\sum_{i=1}^{n} C_i}$$

可知，$\varphi$越大，表示该城市出租车的供需矛盾越小，用户打车越容易。

②接单效率

定义为该城市范围内平均接单时间$t$的倒数，用$\eta$来表示。我们可以统计出该城市内每个采样点的平均接单时间$t_i$，则

$$\eta = \frac{1}{t} = \frac{1}{\frac{1}{n}\sum_{i=1}^{n} t_i} = \frac{n}{\sum_{i=1}^{n} t_i}$$

平均接单时间为用户发出订单到出租车司机接受订单的间隔时间，能较好的表征出租车服务效率。所以$\eta$越大，表明服务效率越高。

③里程利用率

为当日载客总里程$L_1$与当日营业总驶里程$L$之比，用$\mu$ 表示。我们可以统计出该城市内每个采样点的当日载客总里程$L_{1i}$与当日营业总驶里程$L_i$，则

$$\mu = \frac{L_1}{L} = \frac{\sum_{i=1}^{n} L_{1i}}{\sum_{i=1}^{n} L_i}$$

里程利用率越大，说明出租车的平均空车时间相对越小，供求效率越高。

（2）时间、空间的确定

时间方面，因为上下班高峰期会对出租车的需求量产生较大影响，所以选择上下班高峰期和平峰期等不同的时间段；空间方面，为体现不同城市经济发展状况和人口密度对匹配程度的影响，分别选择一、二、三线代表城市进行全方位分析。

时间方面：7：00——9：00（上班高峰）、 10：00——12：00（平峰期）
14：00——16：00（平峰期）、 17：00——19：00（下班高峰）
空间方面：北京、上海（一线城市）、 南京、沈阳（二线城市）
保定、丽江（三线城市）

#### 5.1.1.2 基于层次分析法的模糊综合评判模型[1]

（1）评价方法的确定

综合评价方法主要包括数据包络分析法、层次分析法、模糊综合评价法等。数据包络分析法能较好地比较不同主体间的效率，但这种方法的缺陷是无法对定性指标进行评价。层次析法是基于两两比较的方法，但对比较模糊的概念进行评价则较为困难。模糊综合评价法能够对现实生活中大量模糊概念性指标进行良好评价，但缺陷在于各项指标的权重是由专家据经验给出，因而不够客观。

由于资料的误差、一些统计方法的局限性、某些指标只能定性而不能定量描述的影响因素以及不可预见的各方面因素等，使得匹配程度的计算具有极大的模糊性、随机性和未知性。鉴于此，综合以上三种方法的优缺点，本文将层次分析法与模糊综合评价法相结合，充分发挥各方法的优势，采用 AHP—模糊综合评价模型，先通过层次分析法确定各项研究指标的权重，再基于统计数据使用模糊综合评价法对供求匹配关系进行分析。

（2）层次分析法[2]（249~257）

在深入分析实际问题的基础上，将有关的各个因素按照不同属性自上而下分解成若干个层次。同一层的诸因素从属于上一层，同时又支配着下一层的因素，而同一层的各因素间相互独立，最上层为目标层，通常只有一个因素；最下层为对象层，即我们确定的各个指标；中间层为准则层。将每一层次的各要素进行重要程度的两两比较，得到对比较阵，计算对比较阵的最大特征值及其特征向量，从而得到下一层次相对于上一层次的重要性程度，进而确定各因素间的权重。

①首先，从乘客效益和司机效益两方面考虑，建立如下层次结构图

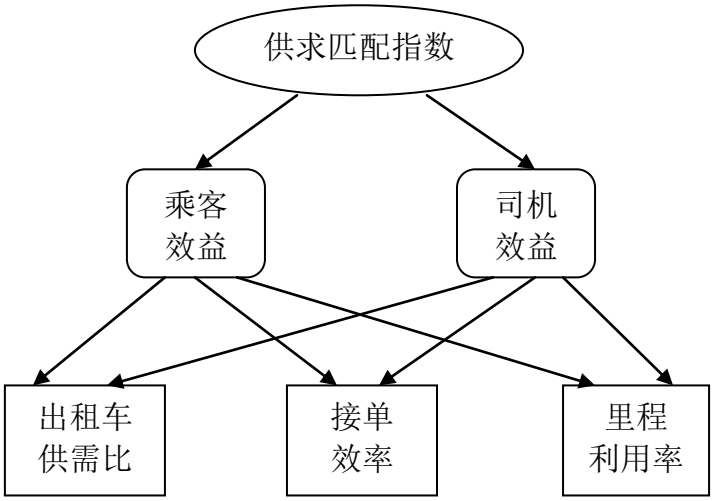


图 5.1-1 层次分析结构图

②利用 1-9 尺度法构造因素间的成对比较矩阵

$$A=(a_{ij})_{n\times n}\ \ (i,j=1,2,\cdots,n)$$

$$a_{ij}=1/a_{ji}$$

③根据已构造矩阵，求解各矩阵的特征向量，特征向量归一化后作为权向量 $\omega$，权向量刻画了下层因素对上层对应因素的影响程度。需要用一致性指标 $CI$，随机一致性指标 $RI$，一致性比率 $CR$ 对矩阵进行一致性检验

$$CI=\frac{\lambda-n}{n-1},CR=\frac{CI}{RI}$$

$n$ 表示所检验矩阵的阶数，$\lambda$ 表示该矩阵的最大特征根；$RI$ 的取值和矩阵阶数有关，查阅资料可知二阶矩阵的 $RI$ 为 0（因为二阶的正互反阵总为一致阵），三阶矩阵的 $RI$ 为 0.58。当 $CR<0.1$ 时，可认为被检验矩阵的不一致程度在容许范围内。

④若通过一致性检验，用其最大特征根对应的归一化特征向量作为权向量 $\omega$，有

$$A\omega=\lambda\omega$$

$$\omega_i'=\frac{\omega_i}{\sum_{k=1}^{n}\omega_k}\ （i=1,2,\cdots,n）$$

即可求出各方案层对准则层、准则层对目标层的权重。而各方案层对目标层的权重为

$$\omega_i=(\omega_A)^T\cdot\omega_{Bi}$$

（3）模糊综合评价

模糊综合评价以模糊数学为基础，将待考察的模糊概念或模糊对象作为一定的模糊集合，建立适当的隶属函数，从而对模糊对象进行定量分析。其主要步骤如下：

①确定评价对象的评价指标$C$，本文共确定了 3 个评价指标。

$$C=(c_1,c_2,c_3)$$

$c_1,c_2,c_3$分别代表出租车供需比、接单效率和里程利用率。

②确定评价集 $V$，本文共确定了 4 个评价等级。

$$V=(v_1,v_2,v_3,v_4)$$

$v_1$, $v_2$, $v_3$, $v$分别代表较差、中等、良好、优秀四个等级。

③建立隶属度矩阵 $R$。在构造了等级模糊子集后，要逐个对被评事物从每个因素上进行量化，即确定从单因素来看被评事物对等级模糊子集的隶属度，从而得到隶属度矩阵 $R$。

$$R=\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1m} \\ r_{21} & r_{22} & \cdots & r_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n1} & r_{n2} & \cdots & r_{nm} \end{bmatrix}$$

④确定评价因素的权向量 $W$。在模糊综合评价中，使用层次分析法确定的评价因素的权向量

$$W=(\omega_1,\omega_2,\cdots,\omega_n)$$

⑤合成模糊综合评价结果矩阵 $S$。运用模糊算子进行权向量和隶属度矩阵间的运算，即：

$$\begin{aligned} S &= W\circ R \\ &= (\omega_1,\omega_2,\cdots,\omega_n)\circ\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1m} \\ r_{21} & r_{22} & \cdots & r_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n1} & r_{n2} & \cdots & r_{nm} \end{bmatrix} \\ &= (s_1,s_2,\cdots,s_n) \end{aligned}$$

其中$s_i$表示被评事物从整体上看对$V$等级模糊子集的隶属程度。

上式中，“$\circ$”为模糊算子，主要包含四种算子，它们在综合程度、体现权数作用方面各有特点。因为在匹配程度的计算中，需要体现明显的权数作用，并且要充分利用 $R$ 的信息，所以我们选用$M(\cdot,\oplus)$算子，计算公式如下

$$A_k=\sum_{j=1}^{m}\omega_j r_{jk},k=1,2,\cdots,n$$

⑥评价等级的确定。根据最大隶属度原则，由糊综合评价结果矩阵 $S$，其中某个某个 $i$ 值使得$s_i$最大，则综合评价属于第 $i$ 等级。

### 5.1.2 模型求解

（1）数据搜索

“苍穹”智能出行平台可以体现当天任何一个小时中，全国各城市出租车供应和需求量、打车难度以及等待接单时间等数据，我们对该平台的数据包进行了抓取，最终得到需要的各项指标值。搜索方法简述如下：

Step1：抓取网站数据包，得到 JSON 格式的数据文件。

Step2：使用 Python 软件解析 JSON 文件，获取在一天当中不同时刻地图上各个数据点的经纬度坐标和数据值。

Step3：将数据中的经纬度坐标通过高德地图的 WebAPI 转换成实际地址，从该地址中得到该数据点所在城市。

Step4：对属于同一城市同一时刻的数据点的数据进行累加。对等待时间进行平均。对于几个较大的城市的数据，直接解析 JSON 即可得到。第三步可省略。

数据搜索程序见附录一。

得到六个城市分别在不同时间段的三项指标数据，选取北京和上海两个城市的数据列于下表。完整数据见附录二。

表 5.1-1 北京和上海的三项指标值

| 一线城市 | 北京 7：00——9：00 | 北京 10：00——12：00 | 北京 14：00——16：00 | 北京 17：00——19：00 | 上海 7：00——9：00 | 上海 10：00——12：00 | 上海 14：00——16：00 | 上海 17：00——19：00 |
|---|---|---|---|---|---|---|---|---|
| 供需比 | 5.0930 | 5.9508 | 3.8570 | 4.7585 | 36.2630 | 30.6551 | 27.1478 | 21.0061 |
| 接单效率 | 0.0298 | 0.0253 | 0.0248 | 0.0240 | 0.0398 | 0.0257 | 0.0218 | 0.0176 |
| 里程利用率 | 70.3% | 68.2% | 68.4% | 68% | 71.3% | 72.1% | 72.3% | 70.5% |

（2）权重确定

利用 1-9 尺度法构造因素间的成对比较矩阵：

第二层对第一层：

$$A=\begin{bmatrix}1 & 3\\ \frac{1}{3} & 1\end{bmatrix}$$

第三层对第二层：

$$B_1=\begin{bmatrix}1 & 2 & 3\\ \frac{1}{2} & 1 & 2\\ \frac{1}{3} & \frac{1}{2} & 1\end{bmatrix} \qquad B_2=\begin{bmatrix}1 & \frac{1}{2} & \frac{1}{4}\\ 2 & 1 & \frac{1}{2}\\ 4 & 2 & 1\end{bmatrix}$$

利用 Matlab 软件求解各矩阵的 $\lambda$、$\omega$、$CR$ 值，结果列如下表：

表 5.1-2 一致性检验表

| 检验矩阵 | 最大特征根 $\lambda$ | 权向量 $\omega$ | 一致性比率 $CR$ |
|---|---|---|---|
| $A$ | 2.00 | [0.7500 0.2500] | 0<0.1 |
| $B_1$ | 3.0092 | [0.5396 0.2970 0.1634] | 0.0079<0.1 |
| $B_2$ | 3.00 | [0.1429 0.2857 0.5714] | 0<0.1 |

由上表可知，*CR* 的值均小于 0.1，说明构造的成对比较矩阵都通过了一致性检验。并且得到准则层对目标的权向量和各方案层对每一准则的权向量，计算得到方案层对目标层的归一化权重为 $\omega=[0.4405\ \ 0.2941\ \ 0.2654]$。

（3）综合评价等级求解

我们选取了多个城市进行三个指标的取样，根据所有取样值的区间分布，将区间进行合理划分，得到自定义的各等级对应的指标值，如下表所示。

表 5.1-3 模糊综合评判等级表

| 指标 | 较差 | 中等 | 良好 | 优秀 |
|---|---|---|---|---|
| 供需比 | <10 | 10~25 | 25~40 | >40 |
| 接单效率 | <0.01 | 0.01~0.025 | 0.025~0.04 | >0.04 |
| 里程利用率 | <60% | 60%~65% | 65%~70% | >70% |

本题参与评价的因素有 3 个，评价标准有 4 个。对于隶属度矩阵中的 $r_{ij}$，表示第 $i$ 种评判指标属于第 $j$ 个等级的可能性。我们借鉴水质等级评判《地表水环境质量标准》的隶属度计算公式[3]

$$\begin{cases} r_{ij}=\begin{cases} 1 & c_i \ge s_j \\ \dfrac{c_i-s_j}{s_{j+1}-s_j} & s_{j+1}<c_i<s_j \\ 0 & c_i \le s_{j+1} \end{cases} \\ r_{i(j+1)}=1-\dfrac{c_i-s_{j+1}}{s_j-s_{j+1}} \end{cases}$$

对每个样本时间段的每个指标来说，$c_i$ 是第 $i$ 个评判标准的值，$s_j$ 是第 $j$ 类的分类标准的边界值。得到所有样本点的隶属度矩阵，以北京为例。其他城市见附录三。

$$7{:}00——9{:}00：\ R_{11}=\begin{bmatrix} 0.5903 & 0.4097 & 0 & 0 \\ 0 & 0.3200 & 0.6800 & 0 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix}$$

$$10{:}00——12{:}00：\ R_{12}=\begin{bmatrix} 0.5951 & 0.4049 & 0 & 0 \\ 0 & 0.9800 & 0.0200 & 0 \\ 0 & 0.3600 & 0.6400 & 0 \end{bmatrix}$$

$$14{:}00——16{:}00：\ R_{13}=\begin{bmatrix} 0.3857 & 0.6143 & 0 & 0 \\ 0 & 0.9867 & 0.0133 & 0 \\ 0 & 0 & 0.6800 & 0.3200 \end{bmatrix}$$

$$17{:}00——19{:}00：\ R_{14}=\begin{bmatrix} 0.4759 & 0.5242 & 0 & 0 \\ 0 & 0.9333 & 0.0667 & 0 \\ 0 & 0 & 0.6000 & 0.4000 \end{bmatrix}$$

运用模糊算子进行权向量和隶属度矩阵间的运算，得到模糊综合评价结果矩阵 $S$，其中某个某个 $i$ 值使得 $s_i$ 最大，则综合评价属于第 $i$ 等级。综合评判程序

见附录四。以北京为例，

$S_{11}=[0.2600\ \ 0.4654\ \ 0.0941\ \ 0.1805]$,综合评价等级为“中等”

$S_{12}=[0.2621\ \ 0.3838\ \ 0.1784\ \ 0.1757]$,综合评价等级为“中等”

$S_{13}=[0.1699\ \ 0.5608\ \ 0.1844\ \ 0.0849]$,综合评价等级为“中等”

$S_{14}=[0.2096\ \ 0.5054\ \ 0.1788\ \ 0.1062]$,综合评价等级为“中等”

按同样的方法得到 24 个样本点的“供求匹配”综合评价等级，列如下表

表 5.1-4 各样本点的综合评价等级

| 城市 | 北京 | 上海 | 南京 | 沈阳 | 保定 | 鞍山 |
|---|---|---|---|---|---|---|
| 7:00——9:00 | 中等 | 良好 | 优秀 | 中等 | 较差 | 中等 |
| 10:00——12:00 | 中等 | 优秀 | 优秀 | 优秀 | 中等 | 良好 |
| 14:00——16:00 | 中等 | 优秀 | 优秀 | 优秀 | 中等 | 良好 |
| 17:00——19:00 | 中等 | 优秀 | 良好 | 较差 | 较差 | 中等 |

### 5.1.3 问题回答

根据表 5.1-4，我们对不同时空出租车资源的“供求匹配程度”进行总结：

（1）时间上来看，上下班高峰期的“供求匹配”程度与平峰期相比较差。

（2）空间上来看，南京、上海等城市的“供求匹配”等级均在“良好”以上，匹配程度较高；北京、沈阳与三线城市的“供求匹配”处于较低等级，匹配程度较低。

### 5.1.4 模型检验与分析

（1）模型检验

通过对“苍穹”智能出行平台的数据包抓取，我们可以得到全国各个城市的打车难度指数。打车难度指数是该智能平台利用自己的评价方法，用 0~10 区间内的整数值表示的打车难易度，其中“0”表示打不到车，“10”表示极容易打到车。它可以在很大程度上反映“供求匹配”程度，所以我们可以选取模型求解得到的部分城市的综合匹配等级，与该智能平台现实的打车难度指数做对照，若吻合度较高，则说明模型求解结果接近真实值，证明模型准确。

首先根据打车难易指数进行匹配等级的划分，0~2.5 代表“较差”，2.6~5.0 代表“中等”，5.1~7.5 代表“良好”，7.6~10.0 代表“优秀”。

不妨取上海、南京、鞍山三个城市，分别作为一、二、三线城市的代表，对抓取的数据进行空间和时间平均处理，并与我们求得的综合匹配等级作对比，列如下表（括号内部为打车难度指数）

表 5.1-5 匹配程度的实际值与计算值对比

| 城市 | 上海（计算） | 上海（实际） | 南京（计算） | 南京（实际） | 鞍山（计算） | 鞍山（实际） |
|---|---|---|---|---|---|---|
| 7:00~9:00 | 良好 | 优秀（8.5） | 优秀 | 优秀（9.5） | 中等 | 优秀（9.5） |
| 10:00~12:00 | 优秀 | 优秀（9.0） | 优秀 | 优秀（8.5） | 良好 | 优秀（9.0） |
| 14:00~16:00 | 优秀 | 优秀（9.0） | 优秀 | 优秀（8.0） | 良好 | 良好（7.5） |
| 17:00~19:00 | 优秀 | 良好（7.5） | 良好 | 优秀（8.0） | 中等 | 良好（7.5） |

（2）对检验结果的分析

通过上表可以看出，除鞍山早高峰时期的两者差别较明显之外，其他采样点的差异均在一个等级以内，说明模型求解结果接近真实值，证明模型具有较高的准确性。而对于两者对比的差异，我们有如下分析：

①在隶属度矩阵确定过程中，对区间的划分有主观性，导致评判结果与实际结果有差异。

②在智能平台中抓取的打车难易度指标不能完全描述“供求匹配”程度。

③鞍山等三线城市的数据量较少，不能完全代表该城市的“供求匹配”程度。

## 5.2 问题二的模型建立和求解

打车软件的服务平台经历了“无补贴——有补贴——取消补贴”的发展阶段，利用问题一中的评价模型，对公司三个阶段的综合指标进行计算和比较，验证补贴方案对“缓解打车难”是否有帮助。然后建立运营平衡模型，对出租车司机在补贴前后的线路选择情况进行仿真模拟，深入分析补贴方案能否缓解“打车难”。

### 5.2.1 模型建立

#### 5.2.1.1 补贴前后“供求匹配”程度对比

通过查阅多方面资料，打车软件服务平台于 2012 年 9 月产生并投入使用；2014 年 1 月实行补贴方案，并经历了对乘客、对司机补贴数额不等的几个阶段。补贴方案使得打车软件兴起，并逐渐发展成为“滴滴”“快的”为主导的格局[4]。2014 年 8 月，两打车软件服务平台相继取消补贴方案。

为此我们以“滴滴”“快的”两公司为分析对象，分别搜索了 2013 年的平均数据、2014 年 2 月的数据，以及问题一中所用到的今日数据，分别代表补贴方案实施之前、实施过程中以及取消补贴后三个阶段。选取北京市为考察对象，分别进行问题一中三个指标的求解，运用问题一中模糊综合评判模型进行供求匹配等级的计算。分析三组数据所求的供求匹配等级差异，来讨论出租车补贴方案是否对缓解打车难有帮助。

因为数据有限，我们只进行了早高峰、中午和晚高峰三个时间段的对比。对比程序见附录五。评价结果矩阵分别如下：

2013 年

早高峰：$S_1=[0.2012\ \ 0.2847\ \ 0.2487\ \ 0]$,综合评价等级为“中等”

中午：$S_2=[0.1790\ \ 0.7619\ \ 0.0591\ \ 0]$,综合评价等级为“中等”

晚高峰：$S_3=[0.3622\ \ 0.2545\ \ 0.2123\ \ 0.1710]$,综合评价等级为“较差”

2014 年 2 月

早高峰：$S_1'=[0.2010\ \ 0.5136\ \ 0.1792\ \ 0.1062]$,综合评价等级为“中等”

中午：$S_2'=[0.1929\ \ 0.2476\ \ 0.3057\ \ 0.2538]$,综合评价等级为“良好”

晚高峰：$S_3'=[0.1637\ \ 0.5117\ \ 0.2716\ \ 0.0531]$,综合评价等级为“中等”

综合以上两组数据以及问题一中北京的评价等级数据，列出三个阶段评价等级对比表

表 5.2-1 补贴前后三个阶段的匹配等级对比

| 时间段 | 补贴前 | 补贴中 | 取消补贴后 |
|---|---|---|---|
| 早高峰 | 中等 | 中等 | 中等 |
| 中午 | 中等 | 良好 | 中等 |
| 晚高峰 | 较差 | 中等 | 中等 |

#### 5.2.1.2 补贴前后“供求匹配”程度变化分析

根据评价的结果可以看出，实行补贴方案后，与方案实行之前相比“供求匹配” 程度有较大的提升，但匹配等级没有达到“优秀”水平，说明没有完全解

决“打车难”问题。在取消补贴政策之后，“供求匹配”程度虽然有下降，但影响并不是很大。

由此我们分析：

（1）实行补贴政策对“打车难”有帮助，但不能完全解决“打车难”问题。因为“打车难”的根本原因在于出租车的供需矛盾。

（2）缓解“打车难”的原因，是因为补贴政策的推出加速了打车服务平的推广，增加市民的使用率。

（3）即使后来补偿政策取消，但是大家还是习惯了用服务平台打车，因此缓解了“打车难”的根本原因是打车服务平台的建立。

为了证明分析的准确性，本文建立了运营平衡模型以证明服务平台的建立才是缓解“打车难”的根本原因。

#### 5.2.1.3 运营平衡模型[5]的建立

等待时间是衡量打车难易度的一个重要因素，我们建立模型来比较平台建立前后的等待时间。出租车行驶分为载客与空驶 2 种状态。在载客阶段，乘客上车处为出行起点，下车处为出行终点，其出行起终点服从于乘客的出行需求。在空驶状态，本次出行起点为上一次的出行终点，而出行终点为最终搜索到乘客处，所以是否使用打车软件对搜索过程中行驶路径的影响至关重要。

（1）无打车软件出租车搜索行为

假设驾驶人选择行为的随机性满足二重指数分布，则从 $j$ 小区出发的空驶出租车选择 $i$ 小区的概率为：

$$P_{ji}=\frac{e^{-\theta\left(t_{ji}+\omega_i\right)D_i}}{\sum_{k\in I_j}e^{-\theta\left(t_{jk}+\omega_k\right)D_i}}$$

式中：$\theta$ 为驾驶人个人特征修正值，该值越大对路网及需求等特征值的不确定性越小，即掌握的准确性越高；$t_{ji}$ 为出租车从 $j$ 小区到 $i$ 小区的最短行驶时间；$\omega_i$ 为出租车在 $i$ 小区的平均搜索时间；$D_i$ 为 $i$ 小区的需求量；$J$ 为搜索起点区域集合；$I_j$ 为搜索目标区域集合，即与 $J$ 相邻交通区的集合。

（2）有打车软件出租车的搜索行为

在打车软件出现后，驾驶人可以通过移动终端实时掌握各小区内需求产生的准确状态，驾驶人选择行为的随机性还是满足二重指数分布。则从 $j$ 小区出发的空驶出租车选择 $i$ 小区的概率为：

$$P_{ji}'=\frac{e^{-\left(t_{ji}+\omega_i\right)}}{\sum_{k\in I_j}e^{-\left(t_{jk}+\omega_k\right)}}$$

载客车辆在 $j$ 小区完成运输转变为空驶，再从 $j$ 小区通过打车软件搜索到乘客，在空驶状态下，以概率 $P_{ji}'$ 驶向下一个 $i$ 小区并接到乘客，则出租车从 $j$ 小区到 $i$ 小区的空驶交通量 $q_{ji}^{v}$ 为：

$$q_{ji}^{v}=D_jP_{ji}=\sum_{i\in I}D_{ij}P_{ji}$$

有从 $j$ 小区到达 $i$ 小区的空驶车辆之和为 $i$ 小区出行总量 $O_i$：

$$O_i=\sum_{j\in J}q_{ji}^{v}=\sum_{j\in J}\sum_{i\in I}D_{ij}P_{ji}$$

出租车总运营时间为总载客时间及总空驶时间。总载客时间 $q^0$ 可表示为:

$$q^0 = \sum_{i\in I}\sum_{j\in J} D_{ij}t_{ij}$$

总空驶时间 $q^v$ 可表示为：

$$q^v = \sum_{i\in I}\sum_{j\in J}\left(q_{ji}^v t_{ji} + \omega_i\right)$$

因此出租车的总运行时间可表示为 $q$：

$$q = \sum_{i\in I}\sum_{j\in J} D_{ij}t_{ij} + \sum_{i\in I}\sum_{j\in J}\left(q_{ji}^v t_{ji} + \omega_i\right)$$

在固定时间内，如以 1h 为统计时间，根据运营时间守恒，则有：

$$q = N$$

其中 $N$ 为出租车规模，即研究的总车辆数。则有平衡状态下使用打车软件进行搜索乘客的出租车运营平衡模型为：

$$\begin{cases} \sum_{j\in J}\sum_{i\in I} D_{ij}P_{ji} = O_i \\ \sum_{i\in I}\sum_{j\in J} D_{ij}t_{ij} + \sum_{i\in I}\sum_{j\in J}\left(q_{ji}^v t_{ji} + \omega_i\right) = N \end{cases}$$

乘客等待时间包括平均搜索时间以及出租车的平均行程时间。所以乘客平均等待时间为 $W_i$：

$$W_i = \omega_i + \frac{\sum_j t_{ji}}{n_i - 1}$$

其中，$n_i$ 为交通小区数。

我们设计了如下图所示的交通网络图，利用此模型来验证打车软件服务平台的建立是否对缓解打车难有帮助。

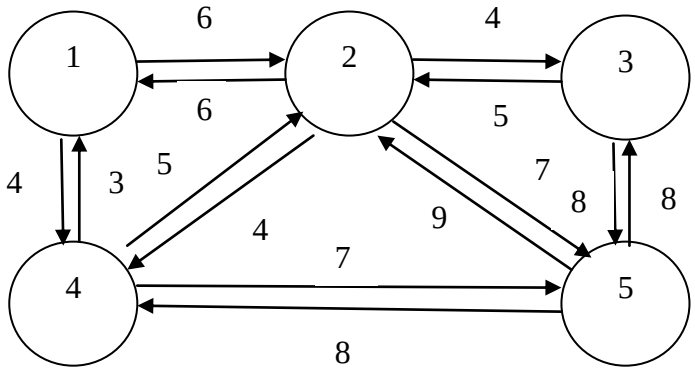


图 5.2-1 模拟交通网络图

城市某区域划分为 5 个交通小区，带编号的圆圈表示各交通区的作用点，带箭头的连接线表示邻接关系，权值表示两点之间不同方向上的最短行时间。出租车出行需求的 OD 矩阵见下表

表 5.2-2 出租车出行需求的 OD 矩阵

| 小区 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | | 30 | 30 | 45 | 15 |
| 2 | 15 | | 15 | 30 | 15 |
| 3 | 45 | 15 | | 30 | 60 |
| 4 | 30 | 60 | 15 | | 15 |
| 5 | 60 | 30 | 30 | 15 | |

假设驾驶人在交通区对之间的路径为最短行程时间的路径，可确定交通区对之间的最短行程时间，结果见下表

表 5.2-3 各交通区对之间的最短行程时间

| 小区 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | | 6 | 10 | 3 | 10 |
| 2 | 6 | | 4 | 5 | 7 |
| 3 | 11 | 5 | | 10 | 8 |
| 4 | 4 | 4 | 8 | | 7 |
| 5 | 12 | 9 | 8 | 8 | |

结合打车软件出现前后出租车司机选择概率的改变，求解上述模型，可以得到两种情况下的等待时间。若打车软件出现后等待时间明显缩短，则说明“打车难”问题得到缓解。

### 5.2.2 模型求解

观察上述模型，方程组包含 $n_{i+1}$ 个方程，变量数为 $n_i$ 个，故该模型不存在唯一精确解，则可尽量缩小等号左边项，使其逼近右边值。令 $e_{rr}$ 为绝对误差值，则有：

$$e_{rr}=\alpha\left|\sum_{j\in J}q_{ji}^{v}-O_i\right|+\beta\left|\sum_{i\in I}\sum_{j\in J}D_{ij}t_{ij}+\sum_{i\in I}\sum_{j\in J}\left(q_{ji}^{v}t_{ji}+\omega_i\right)-N\right|$$

其中 $\alpha$、$\beta$ 分别为两式绝对误差对整体误差的待定权重。取 $\alpha=\beta=0.5$，$N=500$。所以只要让方程两边相减的误差达到最小，转化成无约束最优化问题即可以求解出结果：

$$\min e_{rr}=\alpha\left|\sum_{j\in J}q_{ji}^{v}-O_i\right|+\beta\left|\sum_{i\in I}\sum_{j\in J}D_{ij}t_{ij}+\sum_{i\in I}\sum_{j\in J}\left(q_{ji}^{v}t_{ji}+\omega_i\right)-N\right|$$

通过 matlab 结合遗传算法可以求出最终结果。

遗传算法过程[6]如下：

Step 1:初始化：设置进化代数计数器 $t$=0，设置最大进化代数 $T$，随机生成 $M$ 个个体作为初始群体 $P(0)$。

Step 2:个体评价：计算群体 $P(t)$中各个个体的适应度。

Step 3:选择运算:将选择算子作用于群体。选择的目的是把优化的个体直接遗传到下一代或通过配对交叉产生新的个体再遗传到下一代。选择操作是建立在群体中个体的适应度评估基础上的。

Step 4:交叉运算：将交叉算子作用于群体。遗传算法中起核心作用的就是交叉算子。

Step 5:变异运算：将变异算子作用于群体。即是对群体中的个体串的某些基因座上的基因值作变动。群体 $P(t)$经过选择、交叉、变异运算之后得到下一代群体 $P(t+1)$。

Step 6:终止条件判断:若 $t=T$,则以进化过程中所得到的具有最大适应度个体作为最后解输出，终止计算。

具体求解程序见附录六。

即有无打车平台下，乘客平均等待时间的变化，如下表：

表 5.2-4 各个小区在有无平台时乘客的平均等待时间

| | 小区 1 | 小区 2 | 小区 3 | 小区 4 | 小区 5 | 等待时间 | 误差 |
|---|---|---|---|---|---|---|---|
| 有平台前 | 12.1180 | 10.8652 | 9.8469 | 11.3115 | 11.2094 | 11.0702 | 4.4667 |
| 有平台后 | 8.2529 | 7.7788 | 7.5000 | 6.5449 | 8.0377 | 7.6229 | 3.8202 |

由于遗传算法良好的全局搜索能力，不同 $N$ 下的最优化问题产生的绝对误差值 $e_{rr}$ 均在可接受范围内，得到的结果具有良好的可信度。

由表格可以清楚的看出各个小区在有服务平台的时候，平均等待时间明显缩短，大概缩短了近 4 分钟，由此可以表明在有服务平台时，对“打车难”的现象有帮助，且帮助较大。所以从侧面可以反应出补贴政策对“打车难”有间接性的影响作用。验证了上述的分析。

### 5.2.3 问题回答

通过对北京三个阶段的供求匹配等级的求解，我们得到如下结论：

（1）出租车补贴方案对“缓解打车难”有帮助，但不能完全解决“打车难”问题。

（2）“打车难”得到缓解的主要原因在于打车软件服务平台的建立，以及通过补贴的方式，打车软件在用户生活中得以广泛应用。

## 5.3 问题三的模型建立和求解

两大打车软件服务平台“滴滴”和“快的”通过推出多种补贴方案，建立起大家对打车软件的使用习惯。并且通过问题二的分析，补贴方案并不能完全解决打车难问题，因为打车难问题从根本上是出租车的供需矛盾决定的。如今创建新的打车软件服务平台，我们的补贴方案应该从使服务平台、乘客、司机的效益最大的方向来考虑。

### 5.3.1 模型建立

通过查阅资料，出租车实行补贴方案，在一定程度上缓解“打车难”的同时，也带来了一些新问题。主要有：

①“小费加价”功能兴起，高峰期约车“价高者得”，变相涨价影响社会公平。

②出租车预约现象变多，“摇手招车”的打车方式越来越难打到车。

③因为给予乘客大量补贴，越来越多的乘客选择打出租车，会加重高峰期的打车难度。

我们认为，对乘客进行补贴会增加“小费加价”的几率，且加重高峰期的打车难度，所以我们设计补贴方案时将不再考虑对乘客进行补贴，而只对司机进行补贴。

（1）打车软件服务平台的盈利分析

打车软件服务平台可以从多个方面盈利[7]，包括电召费、传统的广告收益、交易手续费等，其中每一项盈利均与订单数量呈近似正比关系。我们不妨建立盈利与订单数量的正比函数，设比例系数为 $k$，即平均每份订单平台可获利益。

设 $Q$ 为每小时出租车乘客需求[8]，$P$ 为出租车平均运价，$T$ 为乘客平均乘车时间，$W$ 为乘客上车前平均等待时间，三者之间存在一定的函数关系，即

$$Q = f(P,T,W)$$

设 $L$ 为平均乘车距离，$v$ 为平均行驶速度，$N$ 为正在服务中的出租车数量，$\gamma$ 为乘客等待时间系数（车·小时），则有

$$W = \frac{\gamma}{N - QT}$$

根据文献[8]，出租车乘客的出行需求可表示为

$$Q = \bar{Q}\exp(-\alpha(P + \tau T + \kappa W))$$

其中 $\bar{Q}$ 为每小时乘客的潜在出行需求，$\alpha$ 为出租车出行需求的成本弹性系数，$\tau$ 为乘客乘车的单位时间价值，$\kappa$ 为乘客等车的单位时间价值。

设 $\beta$ 为乘坐出租车的乘客中使用打车软件的比例，$m$ 为每单补贴的金额，则平台的每小时收益函数

$$z_1 = (k - m)Q\beta N$$

另外，补贴的金额要在平台的可承受范围之内。设 $m_l$ 、 $m_u$ 分别表示平台给予出租车单位补贴的上限和下限，由平台的资金能力决定。

因此平台的盈利规划模型可表示为

$$\max z_1 = (k - m)Q\beta N$$

$$m_l \le m \le m_u$$

（2）出租车的盈利分析

出租车每小时收入

$$R = PQ$$

出租车每小时成本

$$C = cN = (c_0 + \lambda\mu x)N$$

其中 $c$ 为每辆车单位时间经营总成本，$c_0$ 为每辆车的单位时间固定成本，$\lambda$ 为出租车平均单位里程耗油量，$x$ 为燃油价格。

则出租车每小时经营利润

$$B = R - C = PQ - cN$$

给予补贴后，出租车每小时的经济效益

$$z = B + mN = PQ - (c_0 + \lambda v x_2)N + mQ\beta N$$

此外，运营中的出租车数量应符合该城市的实际情况，且政府的补贴策略要保证出租车经营的经济效益至少不低于经营者接受的利润下限。假设 $N_l$ 、 $N_u$ 分别为出租车运营数量的上下限，$b_l$ 、$b_u$ 分别表示企业可接受的出租车经营单位效益的下限，以及政府允许出租车经营单位效益的上限。因此出租车司机的盈利规划模型可表示为

$$\max z_2 = PQ - (c_0 + \lambda v x_2)N + mQ\beta N$$

$$N_l \le N \le N_u$$

$$b_l N \le PQ - (c_0 + \lambda v x_2)N + mQ\beta N \le b_u N$$

综合服务平台和出租车司机的规划模型，我们得到的双目标规划模型[9]如下：

$$\max z_1 = (k-m)Q\beta N$$
$$\max z_2 = PQ-(c_0+\lambda vx)N+mQ\beta N$$
$$s.t. \quad m_l \le m \le m_u$$
$$N_l \le N \le N_u$$
$$b_l N \le PQ-(c_0+\lambda vx)N+mQ\beta N \le b_u N$$
$$N \in N^*$$

运用 Lingo 软件进行求解，即可求得在约束条件下，使得服务平台和出租车司机的效益均达到最大时的佳补贴金额。

### 5.3.2 模型求解

因为不同城市的出租车数量、每小时乘客的需求会有很大差别，所以不同城市的补贴方案也会有所不同。根据文献[8]，我们可以得到哈尔滨市区的各项参数值，所以我们不妨选择哈尔滨来对补贴金额进行求解。哈尔滨市出租车市场相关参数及变量值如下表

表 5.3-1 哈尔滨市出租车市场相关参数及变量值

| 参数 | 单位 | 数值 |
|---|---|---|
| 单词平均运距 | km | 6.1 |
| 实际平均运行速度 | km/h | 25 |
| 乘客平均出行时间 | h | 0.244 |
| 出行成本需求弹性系数 | 1/元 | 0.045 |
| 出租车乘客等待时间系数 | 辆·时 | 400 |
| 出租车乘客车内时间价值 | 元/时 | 20 |
| 出租车乘客等车时间价值 | 元/时 | 25 |
| 单位时间潜在乘客需求 | 次/时 | 120000 |
| 单位时间出租车运营固定成本 | 元/时 | 20 |
| 出租车单位里程耗油量 | 升/km | 0.1 |
| 哈尔滨市出租车实际数量 | 辆 | 12000 |

根据实际情况，我们取平台补贴标准上限为每单 20.0 元，出租车运营数量上限为 20000 辆，下限为 5000 辆；单位利润上限为 15 元，下限为 10 元。

运用 Lingo 软件进行该双目标规划模型的求解，得到 $m$=5.85。具体求解程序见附录七。

### 5.3.3 问题回答

通过对模型的求解，我们得到最优的补贴方案为每单对司机补贴 5.85 元，不对乘客补贴。

### 5.3.4 模型检验与分析

本文从模型的稳定性与计算结果的可行性两个方面，论证所设计补贴方案的合理性。

（1）对计算结果的可行性分析

为了论证上述模型求出的补贴方案是否合理，是否符合实际。本文结合 2014 年哈尔滨整年的打车软件补贴政策，对计算值是否可行进行评价与分析。2014 年哈尔滨市的补贴方案与计算值之间的关系见下图

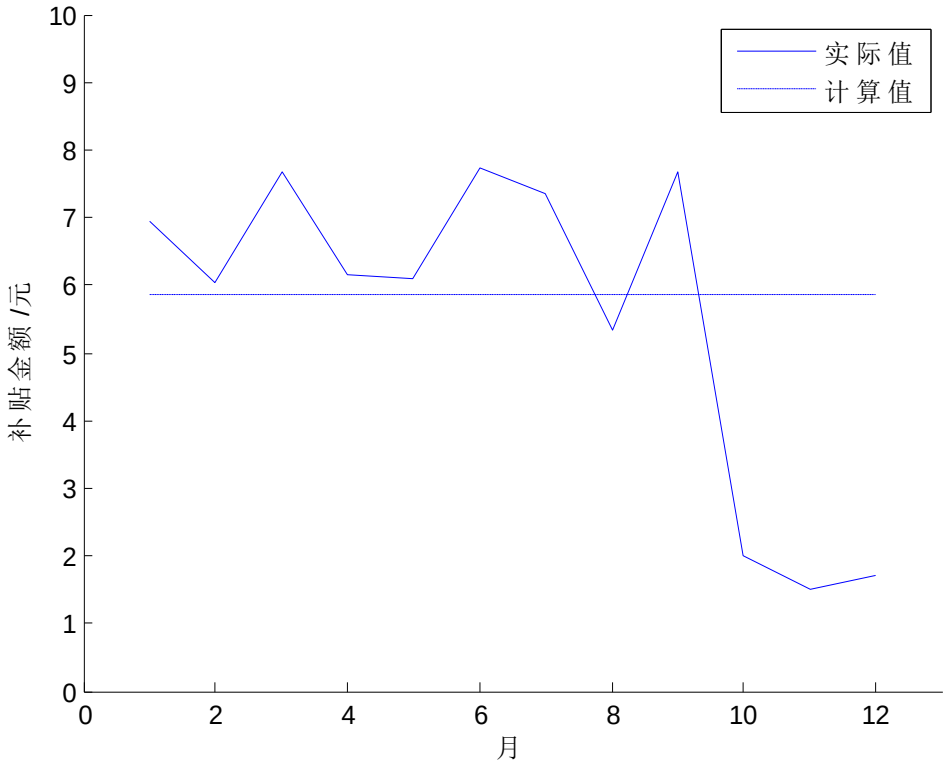


图 5.3-1　2014 年哈尔滨市的补贴方案与计算值之间的关系

由上图可以得到：

①在 9 月份之后，补贴急速下降，其原因有两个：一是多个打车软件合并，竞争变得平缓；二是补贴方案实行一段时间后，用户的打车软件使用开始普及，各打车软件服务平台的目的已经达到。

②从整体上看，实际值在计算理论值上下波动，说明由上述模型计算出的补贴方案与各公司实际的补贴方案相类似，所以设计的补贴方案具有合理性且切符合实际，有一定的准确性。

（2）模型的敏感性分析

为了检验模型的可用性，本文对模型的敏感度进行了分析，改变平台每单的盈利 $k$，观察对司机的补贴 $m$ 有什么变化。利用 MATLAB 进行求解，可得到补贴对盈利能力的敏感度曲线，如下图：

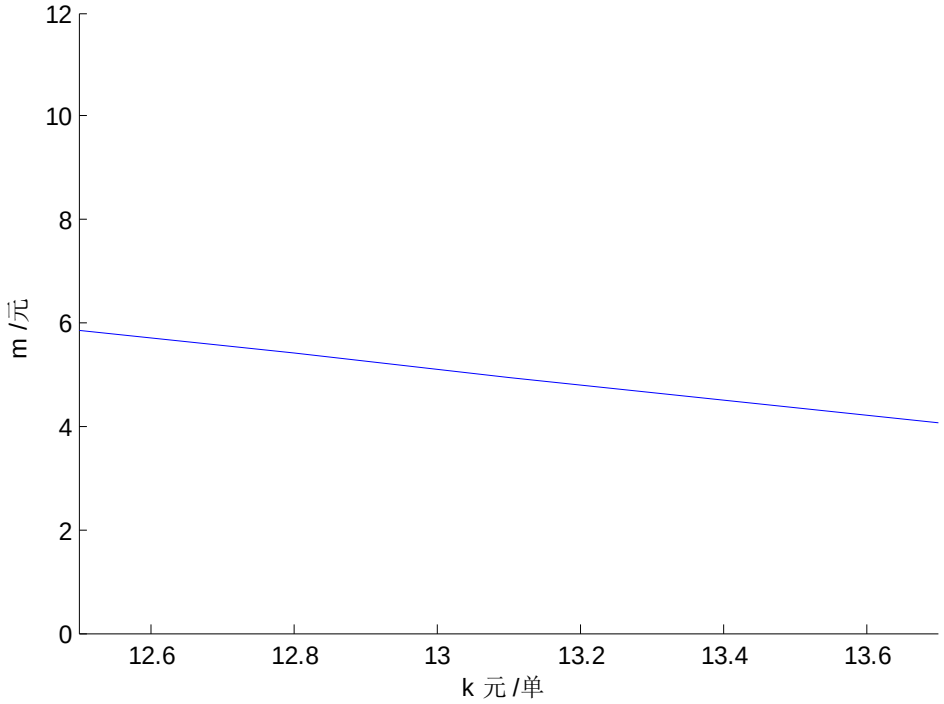


图 5.3-2 补贴 m 对盈利能力 k 的敏感度

分析图可以得到：

①补贴 m 对盈利能力 k 呈负相关，因为平台通过广告等方式每单盈利越多，就不必要在宣传推广软件上投入太多的费用，即补贴司机的费用越少。

②直线斜率较小，可以分析出盈利能力 k 对补贴 m 的影响不大，所建立的模型稳定性好，抗干扰能力强，不会因为一个数据的改变而对结果产生不可忽略的影响。

③对司机的补贴大概落在[4,6]之间，与实际值对比之后，发现结果也符合实际，说明了模型的可靠性。

# 六、模型的评价与改进

## 6.1 模型的优点

1.问题一中采用基于层次分析法的模糊综合评判模型，既实现了对定性定量指标的综合分析，又避免了过于主观的权重确定问题，具有较高的准确度。

2.问题二中通过建立新的运营平衡模型，深入地分析了补贴方案对“缓解打车难”的根本原因，对今后打车软件的补贴和发展方向有较好的指导意义。

3.问题三中建立了双目标规划模型，综合考虑了服务平台和出租车司机的利益问题，结果具有较高的准确性。

## 6.2 模型的缺点

1.模糊综合评判的隶属度矩阵是根据全国所有数据的分布，定义范围区间确定的。若范围区间划定不准确，评价等级就会产生误差。

2.问题二中分析补贴前后供求匹配程度差异，因为只有五个评价等级而无定量指标，无法精确表示差异程度。

3.问题三中因为数据量的限制，只确定了哈尔滨一个城市的补贴方案，使得结果具有片面性。

### 6.3 模型的改进

1.可以选用更精确的模糊综合评判方法，获得更高精确度的隶属度矩阵。
2.建立定量的综合评价指数，以更好地反映补贴前后的供求匹配度改变程度。
3.搜索更为全面的数据，建立更为全面的补贴方案。

## 参考文献


[1]孔繁敏，杨庆瑜，张　亮，打车软件的经济效益评价基于 AHP——模糊综合评价模型[J]，科技和产业，15（4）：52-56，2015
[2]姜启源，谢金星，叶　俊，数学模型[M]，北京：高等教育出版社，2013
[3]赵　威，王盛璋，何　宁，李　亮，杨吟飞，航空钛合金结构件铣削刀具性能模糊综合评判[J]，南京航空航天大学学报，10（44）：711-715，2015
[4]李冬新，栾　洁，滴滴打车的营销策略与发展对策研究[J]，青岛科技大学学报，31（1）：105-108，2015
[4]曹　祎，罗　霞，打车软件背景下出租车运营平衡模型[J]，长安大学学报，35:203-207，2015
[5]马永杰，云文霞，遗传算法研究进展[J]，计算机应用研究，29（4）：1201-1210，2012
[6]于洁涵，梁雪琴，谢绍晖，手机打车软件盈利模式浅析[J]，交通科技与经济，16（2）：63-65，2014
[7]何建平，基于燃油价格变化的城市客运出租车补贴研究[D]，哈尔滨工业大学，1-63，2012
[8]卢晓珊，黄海军，带有空间公平性约束的换乘停车场布局双目标规划模型[J]，系统工程理论与实践，34（9）：2379-2385，2014

# 附录

## 附录一 数据获取程序

```
# getLocation.py，用于获取全国数据，需要安装 Python2.7
#coding=utf-8

#!/usr/bin/env python

import json
import urllib2

f = open('quanguoshijian.json', 'r').read()

t = json.loads(f)

def toStr(x):
    if (len(x) == 0):
        return ''
    return x

def getCityName(latitude, longtitude):
    js = urllib2.urlopen('http://restapi.amap.com/v3/geocode/regeo?location=' + latitude + ',' + longtitude + '&key=28d67f7de0e3b3621b1e1b1746bbd68d&s=rsv3&radius=1000&extensions=all').read()
    res = json.loads(js)
    res = res['regeocode']['addressComponent']
    return res['province'] + toStr(res['city'])

# print t['result']['data'][0][0]
cList = []

f = open('data.txt', 'w')

for i in range(0, 24):
    temp = t['result']['data'][i]
    toAdd = []
    for j in range(0, len(temp)):
        lat = temp[j][1]
        lon = temp[j][2]
        val = temp[j][3]
        o = {}
        o['cityName'] = getCityName(str(lat), str(lon))
        o['time'] = str(i)
        o['val'] = str(val)
        f.write((o['time'] + '\t' + o['cityName'] + '\t' + o['val'] + '\n').encode('gbk'))
        print o['time'] + '\t' + o['val'] + '\t' + o['cityName'] + '\n'
f.close()

#getData.py，用于获取几个大城市的数据，需要计算机安装 curl 和 Python2.7
```

```
import urllib2
import json
import sys
import os

def getDataFromJson(jsontxt, fileName, avg):
	datas = json.loads(jsontxt)
	p = datas['result']['data']
	f = open('./datas/'+fileName, 'w')
	print len(p)
	for i in range(0, 24):
		q = p[i]
		s = 0
		for j in range(0, len(q)):
			#print q[j]
			s += q[j][3]
		if not avg:
			f.write(str(s) + '\n')
		else:
			f.write(str(s / len(q)) + '\n')
		#print i
	f.close()

def getCityData(cityId, dataType):
	cmd='curl
"http://v.kuaidadi.com/point?cityId='+cityId+'&scope=city&date=0&dimension='+dat
aType+'&num=300" -H "Cookie:
Hm_lvt_880ac891821851b8c6b00780215b1293=1441932011" -H "Accept-Encoding:
gzip, deflate, sdch" -H "Accept-Language: en-US,en;q=0.8,zh-CN;q=0.6,zh;q=0.4" -H
"User-Agent: Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/45.0.2454.85 Safari/537.36" -H "Accept: */*" -H
"Referer: http://v.kuaidadi.com/" -H "X-Requested-With: XMLHttpRequest" -H
"Connection: keep-alive" --compressed -o '+cityId+dataType+'.json'
	os.system(cmd)
	print cityId+dataType
	return open(cityId+dataType+'.json','r').read()

def getCityDataA(cId, cName):
	getDataFromJson(getCityData(cId, 'demand'), cName+'demand.txt', False)
	getDataFromJson(getCityData(cId, 'distribute'), cName+'distribute.txt', False)
	getDataFromJson(getCityData(cId, 'response'), cName+'response.txt', True)

cites = [
		('bj','110100'),
		('sh','310100'),
		('nj','320100'),
		('sy','210100')

		]
```

```
for i in range(0,4):
        print cites[i][0]
        print cites[i][1]
       getCityDataA(cites[i][1], cites[i][0])
```

```
%dataImporter.m 用于将获得的几个大城市的 txt 数据导入到 Matlab 中
cityList = {'bj', 'wh', 'nj', 'sh', 'xa', 'gz', 'sz', 'sy', 'cd', 'hz'};

for i=1:10
        cName = cityList{i};
        cDemand = importdata(strcat('./docs/datas/',cName,'demand.txt'));
        cResponse = importdata(strcat('./docs/datas/',cName,'response.txt'));
        cDistribute = importdata(strcat('./docs/datas/',cName,'distribute.txt'));
        t = [cDemand, cResponse, cDistribute];
        k = mat2dataset(t);
        k.Properties.VarNames{1} = 'demand';
        k.Properties.VarNames{2} = 'response';
        k.Properties.VarNames{3} = 'distribute';
        eval(strcat(cName,'=k'));
end

% Baoding and Anshan data
t1 = getCityData('河北省保定市', countryDemand, countryWait);
t2 = getCityData('辽宁省鞍山市', countryDemand, countryWait);
t = bj.distribute / 10;
t = [t1, t];
k = mat2dataset(t);
k.Properties.VarNames{1} = 'demand';
k.Properties.VarNames{2} = 'response';
k.Properties.VarNames{3} = 'distribute';
bd = k;

t =getCityData('辽宁省鞍山市', countryDemand, countryWait);
t = [t2, t];
k = mat2dataset(t);
k.Properties.VarNames{1} = 'demand';
k.Properties.VarNames{2} = 'response';
k.Properties.VarNames{3} = 'distribute';
as = k;

%getCityData.m 用于统计城市数据
function res = getCityData(cityName, vdem, vres)
        i = 1;
        c1 = 1;
        c2 = 1;
        p = zeros(24, 2);
        while i < 5250
                if strcmp(cityName, vdem.city(i))
```

```
                p(c1,1) = vdem.value(i);
                c1 = c1 + 1;
            end
            i = i + 1;
        end
        i = 1;
        while i < 3842
            if strcmp(cityName, vres.city(i))
                p(c2, 2) = vres.value(i);
                c2 = c2 + 1;
            end
            i = i + 1;
        end
        res = p;
end
```

**附录二 其他城市的指标值**

| 二线城市 | 南京 7：00——9：00 | 南京 10：00——12：00 | 南京 14：00——16：00 | 南京 17：00——19：00 | 沈阳 7：00——9：00 | 沈阳 10：00——12：00 | 沈阳 14：00——16：00 | 沈阳 17：00——19：00 |
|---|---|---|---|---|---|---|---|---|
| 出租车供需比 | 31.4149 | 34.0400 | 20.0363 | 20.1222 | 68.0005 | 52.9301 | 24.3631 | 17.0054 |
| 接单效率 | 0.0390 | 0.0401 | 0.0263 | 0.0250 | 0.0512 | 0.0305 | 0.0265 | 0.0216 |
| 里程利用率 | 65.4% | 66.3% | 67.3% | 64.8% | 57% | 59.6% | 61.3% | 57.3% |

| 三线城市 | 保定 7：00——9：00 | 保定 10：00——12：00 | 保定 14：00——16：00 | 保定 17：00——19：00 | 鞍山 7：00——9：00 | 鞍山 10：00——12：00 | 鞍山 14：00——16：00 | 鞍山 17：00——19：00 |
|---|---|---|---|---|---|---|---|---|
| 出租车供需比 | 18.1766 | 9.9320 | 9.3751 | 7.0410 | 15.1440 | 15.1440 | 6.1337 | 3.9723 |
| 接单效率 | 0.0212 | 0.0201 | 0.0225 | 0.0194 | 0.0071 | 0.0164 | 0.0166 | 0.0300 |
| 里程利用率 | 70.3% | 68.2% | 68.4% | 68% | 71.3% | 72.1% | 72.3% | 70.5% |

## 附录三 其他各城市的隶属度矩阵

上海

7:00——9:00： $R_{21}=\begin{bmatrix}0.0000 & 0.0000 & 0.7509 & 0.2491\\ 0.0000 & 0.0000 & 0.9867 & 0.0133\\ 0.0000 & 0.0000 & 0.0000 & 1.0000\end{bmatrix}$

10:00——12:00： $R_{22}=\begin{bmatrix}0.0000 & 0.0000 & 0.3767 & 0.6233\\ 0.0000 & 0.0000 & 0.0467 & 0.9533\\ 0.0000 & 0.0000 & 0.0000 & 1.0000\end{bmatrix}$

14:00——16:00： $R_{23}=\begin{bmatrix}0.0000 & 0.0000 & 0.1432 & 0.8568\\ 0.0000 & 0.7867 & 0.2133 & 0.0000\\ 0.0000 & 0.0000 & 0.0000 & 1.0000\end{bmatrix}$

17:00——19:00： $R_{24}=\begin{bmatrix}0.0000 & 0.7337 & 0.2663 & 0.0000\\ 0.0000 & 0.5067 & 0.4933 & 0.0000\\ 0.0000 & 0.0000 & 0.0000 & 1.0000\end{bmatrix}$

南京

7:00——9:00： $R_{31}=\begin{bmatrix}0.0000 & 0.0000 & 0.4277 & 0.5723\\ 0.0000 & 0.0000 & 0.9333 & 0.0667\\ 0.0000 & 0.0000 & 0.0800 & 0.9200\end{bmatrix}$

10:00——12:00： $R_{32}=\begin{bmatrix}0.0000 & 0.0000 & 0.6027 & 0.3973\\ 0.0000 & 0.0000 & 0.0000 & 1.0000\\ 0.0000 & 0.0000 & 0.2600 & 0.7400\end{bmatrix}$

14:00——16:00： $R_{33}=\begin{bmatrix}0.0000 & 0.6691 & 0.3309 & 0.0000\\ 0.0000 & 0.0000 & 0.0867 & 0.9133\\ 0.0000 & 0.0000 & 0.4600 & 0.5400\end{bmatrix}$

17:00——19:00： $R_{34}=\begin{bmatrix}0.0000 & 0.6747 & 0.3253 & 0.0000\\ 0.0000 & 0.0000 & 0.0000 & 0.0000\\ 0.0000 & 0.0000 & 0.6800 & 0.3200\end{bmatrix}$

沈阳：

7:00——9:00： $R_{41}=\begin{bmatrix}0.0000 & 0.0000 & 0.0000 & 1.0000\\ 0.0000 & 0.0000 & 0.0000 & 1.0000\\ 0.9500 & 0.0500 & 0.0000 & 0.0000\end{bmatrix}$

10:00——12:00： $R_{42}=\begin{bmatrix}0.0000 & 0.0000 & 0.0000 & 1.0000\\ 0.0000 & 0.0000 & 0.3667 & 0.6333\\ 0.9933 & 0.0067 & 0.0000 & 0.0000\end{bmatrix}$

14:00——16:00： $R_{43}=\begin{bmatrix}0.0000 & 0.9575 & 0.0425 & 0.0000\\ 0.0000 & 0.0000 & 0.1000 & 0.9000\\ 0.0000 & 0.2600 & 0.7400 & 0.0000\end{bmatrix}$

17:00——19:00： $R_{44}=\begin{bmatrix}0.0000 & 0.4670 & 0.5330 & 0.0000\\ 0.0000 & 0.7733 & 0.2267 & 0.0000\\ 0.9550 & 0.0450 & 0.0000 & 0.0000\end{bmatrix}$

保定

7:00——9:00： $R_{51}=\begin{bmatrix}0.0000 & 0.5451 & 0.4549 & 0.0000\\ 0.0000 & 0.7467 & 0.2533 & 0.0000\\ 0.0000 & 0.0000 & 0.0000 & 1.0000\end{bmatrix}$

10:00——12:00： $R_{52}=\begin{bmatrix}0.9932 & 0.0068 & 0.0000 & 0.0000\\ 0.0000 & 0.6733 & 0.3267 & 0.0000\\ 0.0000 & 0.0000 & 0.6400 & 0.3600\end{bmatrix}$

14:00——16:00： $R_{53}=\begin{bmatrix}0.9375 & 0.0625 & 0.0000 & 0.0000\\ 0.0000 & 0.8333 & 0.1667 & 0.0000\\ 0.0000 & 0.0000 & 0.6800 & 0.3200\end{bmatrix}$

17:00——19:00： $R_{54}=\begin{bmatrix}0.7041 & 0.2959 & 0.0000 & 0.0000\\ 0.0000 & 0.6267 & 0.3733 & 0.0000\\ 0.0000 & 0.0000 & 0.6000 & 0.4000\end{bmatrix}$

鞍山

7:00——9:00： $R_{61}=\begin{bmatrix}0.0000 & 0.3429 & 0.6571 & 0.0000\\ 0.7100 & 0.2900 & 0.0000 & 0.0000\\ 0.0000 & 0.0000 & 0.0000 & 1.0000\end{bmatrix}$

10:00——12:00： $R_{62}=\begin{bmatrix}0.0000 & 0.3429 & 0.6571 & 0.0000\\ 0.0000 & 0.4267 & 0.5733 & 0.0000\\ 0.0000 & 0.0000 & 0.0000 & 1.0000\end{bmatrix}$

14:00——16:00： $R_{63}=\begin{bmatrix}0.6134 & 0.3866 & 0.0000 & 0.0000\\ 0.0000 & 0.4400 & 0.5600 & 0.0000\\ 0.0000 & 0.0000 & 0.0000 & 1.0000\end{bmatrix}$

17:00——19:00： $R_{64}=\begin{bmatrix}0.3972 & 0.6028 & 0.0000 & 0.0000\\ 0.0000 & 0.0000 & 0.3333 & 0.6667\\ 0.0000 & 0.0000 & 0.0000 & 1.0000\end{bmatrix}$

## 附录四 综合评判程序

```
% judgeCity.m  评判函数
function res = judgeCity(city)
    res = zeros(4, 2);
    n = city.distribute ./ city.demand;
    a = 1 ./ city.response;
    res(1,1) = sum(n(7:8))/2;
    res(1,2) = sum(a(7:8))/2;
    res(2,1) = sum(n(10:11))/2;
```

```
    res(2,2) = sum(a(10:11))/2;
    res(3,1) = sum(n(14:15))/2;
    res(3,2) = sum(a(14:15))/2;
    res(4,1) = sum(n(17:18))/2;
    res(4,2) = sum(a(17:18))/2;
end
%between.m between 函数
function res = between(val, range)
    if val < range(1)
        res = [val/range(1), 1];
        return;
    end
    if val > range(3)
        res = [1,4];
        return;
    end
    if val >= range(1) && val < range(2)
        res = [(val - range(1))/(range(2) - range(1)), 2];
    else
        res = [(val - range(2))/(range(3) - range(2)), 3];
    end
end

%getRMatrix.m
function res = getRMatrix(x)
    step = [10,25,40;...
            0.01, 0.025, 0.04;...
            0.6, 0.65,0.7];
    res = zeros(3,4);
    for i=1:3
        temp = between(x(i), step(i,:));
        res(i, temp(2)) = temp(1);
    end
    t2 = zeros(3,4);
    for i=1:3
        for j=1:4
            if res(i,j) > 0 && res(i,j) < 1
                t2(i, j+1) = 1-res(i,j);
            end
        end
    end
    res = res + t2;
end

%calR.m 进行评价
toCalBj = [
    5.9030, 5.9508, 3.8570, 4.7585;
    0.0298, 0.0253, 0.0248, 0.0240;
    0.7030, 0.6820, 0.6840, 0.6800
];
```

```
toCalSh = [
	36.2630, 30.6511, 27.1478, 21.0061;
	0.0398,    0.0257, 0.0218, 0.0176;
	0.713, 0.721, 0.723, 0.705
];
toCalNj = [
	31.4149, 34.0400, 20.0363, 20.1212;
	0.0390, 0.0401, 0.0263, 0.0250;
	0.654, 0.663, 0.673, 0.684;
];
toCalSy = [
	68.0005, 52.9301, 24.3631, 17.0054;
	0.0512, 0.0305, 0.0265, 0.0216;
	0.57, 0.596, 0.613, 0.573
];
toCalBd = [
	18.1766, 9.9320, 9.3751, 7.0410;
	0.0212, 0.0201, 0.0225, 0.0194;
	0.703, 0.682, 0.684, 0.68
];
toCalAs = [
	15.1440, 15.1440, 6.1337, 3.9723;
	0.0071, 0.0164, 0.0166, 0.0300;
	0.713, 0.721, 0.723, 0.705
];
disp 'bj'
for j=1:4
	getRMatrix(toCalBj(:,j))
	[0.4405,0.2941,0.2654]*getRMatrix(toCalBj(:,j))
end
disp 'sh'
for j=1:4
	getRMatrix(toCalSh(:,j))
	[0.4405,0.2941,0.2654]*getRMatrix(toCalSh(:,j))
end
disp 'nj'
for j=1:4
	getRMatrix(toCalNj(:,j))
	[0.4405,0.2941,0.2654]*getRMatrix(toCalNj(:,j))
end
disp 'sy'
for j=1:4
	getRMatrix(toCalSy(:,j))
	[0.4405,0.2941,0.2654]*getRMatrix(toCalSy(:,j))
end
disp 'bd'
for j=1:4
	getRMatrix(toCalBd(:,j))
	[0.4405,0.2941,0.2654]*getRMatrix(toCalBd(:,j))
end
```

```
disp 'as'
for j=1:4
	getRMatrix(toCalAs(:,j))
	[0.4405,0.2941,0.2654]*getRMatrix(toCalAs(:,j))
end
```

附录五 第二问比较程序

```
p = [pastBjData.num ./ pastBjData.demand, 1./pastBjData.waittime, pastBjData.use];
p = p';
disp '2012:'
for j=1:3
	getRMatrix(p(:,j))
	[0.4405,0.2941,0.2654]*getRMatrix(p(:,j))
end
disp '2013:'
for j=4:6
	getRMatrix(p(:,j))
	[0.4405,0.2941,0.2654]*getRMatrix(p(:,j))
end
disp '2015:'
for j=7:9
	getRMatrix(p(:,j))
	[0.4405,0.2941,0.2654]*getRMatrix(p(:,j))
end
```

附录六 第二问模型的计算

```
D = [
	0, 30, 30, 45, 15;
	15, 0, 15, 30, 15;
	45, 15, 0, 30, 60;
	30, 60, 15, 0, 15;
	60, 30, 30, 15, 0;
];

T = [
	0, 9, 15, 4.5, 15;
	9, 0, 6, 7.5, 10.5;
	16.5, 7.5, 0, 15, 12;
	6, 6, 12, 0, 10.5;
	18, 13.5, 12, 12, 0;
];

D = D / (1.5* 100);
T = T / 1.5 / 60;

O = sum(D, 2);

removeZero = @(x) x - x.*eye(size(x));

getP                                                                    =
```

```
@(w)removeZero(exp(-(T+repmat(w,5,1)))./repmat(sum(removeZero(exp(-(T+repmat(w,5,1)))),2),1,5);
getPOld = @(w)removeZero(exp(-2*(T+repmat(w,5,1)).*repmat(O,1,5)))./repmat(sum(removeZero(exp(-2*(T+repmat(w,5,1)).*repmat(O,1,5)))',1,5);
getQ = @(P)D*P;

% function getQjS in getQjS.m

% tgtfunc
alpha = 0.5;
beta = 0.5;
err = @(w,N)alpha*abs(sum(sum(getQ(getP(w)),2) - O))+beta*abs(sum(sum(D.*T)) - N + sum(sum(getQ(getP(w)) + repmat(w,5,1))));
errOld = @(w,N)alpha*abs(sum(sum(getQ(getPOld(w)) - repmat(O',5,1))))+beta*abs(sum(sum(D.*T)) - N + sum(sum(getQ(getPOld(w)) - repmat(w,5,1))));

N = 100;
f1 = @(w)err(w, N);
f2 = @(w)errOld(w, N);
LB = [0,0,0,0,0];
UB = [inf,inf,inf,inf,inf];
opt=gaoptimset('Generations',500);
disp 'Old-----------'
olds = ga(f1,5,[],[],[],[],LB,UB,[],opt)
f1(olds)
mean(olds)
disp 'New-----------'
new = ga(f2,5,[],[],[],[],LB,UB,[],opt)
f2(new)
mean(new)
```

## 附件七 第三问规划模型求解 lingo 代码

```
!需启用 global solver;

MODEL:
    DATA:
        q = 10000;
        x2 = 7.95;
        l = 6.1;
        p0 = 9;
        b = 1.9;
        v = 25;
        t = 0.244;
        alpha = 0.045;
        gama = 400;
        tao = 20;
        k = 12.5;
```

```
        p = 14.89;
        c0 = 20;
        lam = 0.1;
        Nf = 12000;
    ENDDATA
    SUBMODEL obj1:
        MAX = z1;
        z1 = (k-m)*q*b*n;
        @free(z1);
    ENDSUBMODEL
    SUBMODEL obj2:
        MAX = z2;
        z2 = p*q-(c0+lam*v*x2)*n+m*q*b*n;
        @free(z2);
    ENDSUBMODEL
    SUBMODEL con1:
        @gin(n);
        m >= 0;
        m <= 12.3;
        n >= 5000;
        n <= 20000;
        p*q-(c0+lam*v*x2)*n+m*q*b*n >= 10*n;
        p*q-(c0+lam*v*x2)*n+m*q*b*n >= 15*n;
    ENDSUBMODEL
    SUBMODEL con2:
        ! 目 标 函 数 MIN = @sqr((k-m)*q*b*n-z1) +
@sqr(p*q-(c0+lam*v*x2)*n+m*q*b*n-z2),用第一步的结果替换 z1,z2;
        MIN = @sqr((k-m)*q*b*n-0.4749051E+10 ) +
@sqr(p*q-(c0+lam*v*x2)*n+m*q*b*n-0.4673351E+10 );
        f1 = (k-m)*q*b*n;
        f2 = p*q-(c0+lam*v*x2)*n+m*q*b*n;
    ENDSUBMODEL
    CALC:
        !第一步：计算下面两项，获得 z1 和 z2,用它们替换 con2 目标函数中的
z1 和 z2;
        !@solve(obj1,con1);
        !@solve(obj2,con1);
        !第二步: 完成第一步后将其注释，执行下面一步;

        @solve(con2,con1);
    ENDCALC
END
```