“互联网+”时代的出租车资源配置

摘要：

出租车司机与乘客信息由于互通出现问题，从而导致“空车行驶”和“打车难”现象的出现，于是打车软件应运而生。因此如何结合相关统计数据，选取合适指标来分析不同时间地点的出租车供求匹配程度，探究补贴方案对于缓解打车难的影响程度，如何设计合适的补贴方案，使得软件公司、司机和乘客达到利益的最大化等问题的研究具有理论价值，又具有实际意义。

本文收集了**北京**市 2015 年 9 月 11 日从 00：00~24:00 一共 24 个小时作为研究对象，空间上选取三个**一线**城市（北京、上海、广东）、三个**二线**城市（南京、杭州、合肥）和三个**三线**城市（台州、三亚、吉林）选取 **13:00** 时的数据作为研究对象。文中建立了三个量化模型从不同侧面研究了出租车供求匹配程度，补贴方案对缓解打车难的影响程度以及合适的补贴方案的设计问题，具体如下：

针对问题一，分别研究了不同时空之下的出租车匹配度的度量，在时间维度上，选择简单的描述统计量作出市民出租车需求量的变化曲线，在空间维度上，对不同城市的出租车数量进行比较分析，这种方法可以粗略的反映在不同时空之下供需关系所处的状态。更进一步，从微观的角度，选取了选取**被抢单时间，空车率，出租车的万人拥有量，成单率**这四个指标，采取 **AHP** 主观赋权法和**熵值法**客观赋权法相结合得到四个指标的权重向量，将这个权重值代入 **TOPSIS**（逼近理想点法）中去，通过定义数据与**正负理想点法**的距离（贴近度）进行排序，得到了微观情况下的不同时空的匹配程度的度量值。结果表明，从时间上看，出租车供求不够匹配的程度产生在上下班高峰期 1~2 个小时之后，且像北京这样的大城市一天之中出租车供求不够匹配的程度也可能处在 20:00-22:00。从空间上看，北京市因为限车和道路拥挤，供求匹配度最低，而杭州市因为外来旅游人数较多，供求匹配度最高。

针对问题二，建立了**双人非合作打车博弈**模型。出租车司机与乘客是博弈的局中人，他们具有相同的纯策略集，即可以自由的选择是否响应打车软件所提供的补贴政策，通过引入混合策略，即将**司机**看成**局中人 1，其**响应补贴政策的概率为 x,**乘客**看做**局中人 2，其**响应补贴政策的概率为 y,文中得到了一般情形下的 Nash 平衡点。通过给出特定的局中人收益矩阵，计算得到了当 x=2/5，y=1/3 时达到 **Nash 平衡点**，结果表明：司机响应补贴政策的概率的响应度为 2/5,乘客对补贴模型的响应度为 1/3，即出租车司机与乘客响应补贴政策的概率均大于 0，说明打车软件的补贴政策确实有助于缓解打车难的问题，且司机响应度高于乘客。

针对问题三，建立了**合作博弈模型**，即从合作博弈的角度来审视补贴方案的合理性问题。因为打车软件公司、出租车司机、乘客三个主体之间有竞争关系，同时也有利益共享的关系，因此“互联网+”时代的出租车补贴行为可以视为一个合作博弈。本文认为一个合理补贴方案的设计实质上一个利益分配关系。文中应用合作对策中 Shapley 值分配分方案，计算出打车软件公司、出租车司机、乘客应该分享合理的收益比例，并给出了其解析式，在参数给定的模拟环境下，他们分别为 5%、85%、10%。

实例分析的结果表明文中所建立的模型具有一定的合理性和有效性。文中**对“打车难”现象提出了切实可行的意见和建议**。

关键字：层次分析法（AHP）；熵值法；TOPSIS；非合作博弈模型；合作博弈模型

1.问题重述

出租车作为普通民众的重要交通工具之一，在民众日常生活出行中扮演着必不可缺的角色。但由于它无固定营运路线和停靠站，易使出租车司机与乘客信息互通出现问题，从而导致“空车行驶”和“打车难”现象的出现，于是打车软件应运而生。打车软件为吸引用户，推出各式各样的补贴方案。大部分的司机和用户开始接受这种线上支付，网上预约叫车的方式，这对城市的出租车资源配置产生了很大影响。

建立合适指标集的情况之下分析不同时间地点的供求匹配程度，从而结合出租车的补贴方案，结合数据探究补贴方案对于缓解打车难是否有帮助，在此基础上可设计创建合适的补贴方案，使得补贴方案在吸引用户的同时达到利益的最大化，这对于软件创造业和城市出租车管理都有重要意义，所以本次研究的内容既具有理论价值，又具有实际意义。

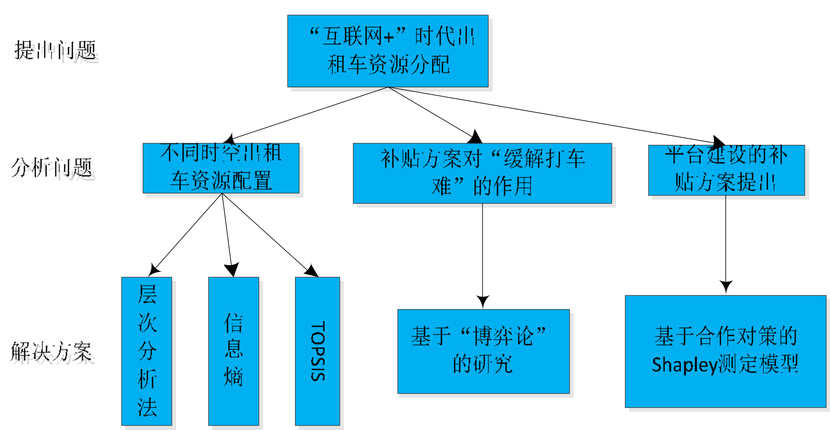
2.问题分析

对于问题一不同时空出租车资源的供求匹配程度的刻画中，一般实际情况下，供求匹配对应就是乘客与司机双方面的满意度大小，即通常情况，供求匹配度越高，乘客司机的满意度也较高；匹配程度越低，则反之。所以可将对供求匹配程度的刻画转换为对满意度的刻画，从而选取相应指标，探讨这些指标对满意度大小影响。

而对于问题二各公司补贴政策对应打车难的问题中，各个公司的补贴方案大致相同，所以应该将主要的目光落在补贴方案如何解决打车难的影响因素问题上。

对于问题三，从合作博弈的角度来审视补贴方案的合理性问题。因为打车软件公司、出租车司机、乘客三个主体之间有竞争关系，同时也有利益共享的关系，因此“互联网+”时代的出租车补贴行为可以视为一个合作博弈。创建新的打车软件服务平台，制定一个方案使得出租车司机，乘客和打车软件公司三方共赢的补贴方案，因此“互联网+”时代的出租车补贴行为可以视为一个合作博弈。本文认为一个合理补贴方案的设计实质上一个利益分配关系。文中应用合作对策中 Shapley 值分配分方案，计算出打车软件公司、出租车司机、乘客应该分享合理的收益比例。

解决问题的思路：



3.模型假设

假设 **1** 数据的来源是可靠的

假设 **2** 除模型中所考虑的因素外其他因素默认对模型无影响

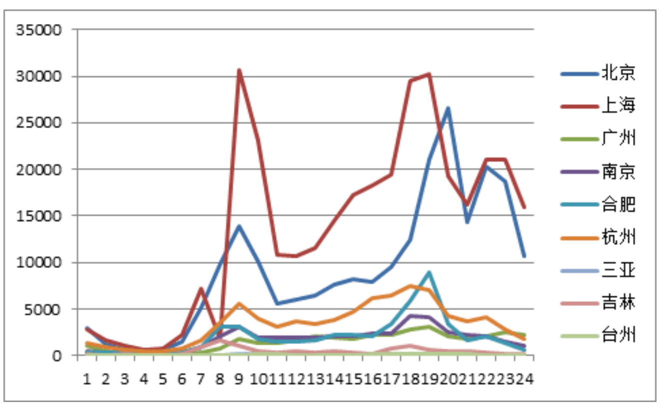
假设 **3** 问题二的模型中假设司机（局中人 1）和乘客（局中人 2）无信息交流。

4.问题一模型的建立与分析

4.1 简单的宏观匹配度探讨

问题一中要求选取指标对匹配程度进行评估，宏观角度可以反映国民生产水平、人口密度、城市规模发展等因素对出租车资源配置的宏观影响，本次模型决定从宏观和微观两个不同的角度以 9 月 11 日 00：00--24:00 这一天三个一线城市（北京，上海，广东），三个二线城市（南京，合肥，杭州），三个三线城市（三亚，吉林，台州）这九个城市为代表，来对问题进行分析，力争完整性和正确性。

先用描述统计量的方法对数据情况进行大概的了解，首先利用一天之内九个城市 24 个时段的需求量画出曲线图，可以从九个城市的需求量来看上海市需求量最大，其次为北京市，而对于二、三线城市需求量要明显小的多，且对应于一整天的变化，上海市、北京市变化量大而其他城市要明显小得多。北京市 75000 台，上海市 45000 台，广州市 33000 台，合肥市 8500 台，南京市 13832 台，杭州市 11590 台，三亚 1020 台，吉林 5000 台，台州 2400 台，从图中如果简单从北京市的出租车数量和北京市出租车需求量两个角度来考虑，那么对于一线城市北京，上海，广州，出租车数量远大于市民对出租车的需求量，这与经济发达，私家车拥有率高有关。上海市供求相差在三者中最小，匹配度最高，而其他两个一线城市北京广州，大部分出租车处于闲置或空车营运状态，尤其是北京，几乎 2/3 的出租车闲置，可能这与北京市本身的单双号限车令相关。而对于二线城市，南京市也出现类似于一线城市的情况，出租车数量远大于需求量，合肥则会在高峰期时段出现打车难的现象，对于三线城市基本来说一直在出租车供应量左右浮动匹配度也还不错。



4.2 指标解释

对实际情况进行分析和查阅相关资料[1][2]得到了一般衡量供应与是否满足需求的指标，总结起来大概可以为:

(1)万人拥有量，

(2)被抢单时间，

(3)空车率,

(4)成单率这几个指标。

下面对这四个指标进行量化。

万人拥有量：这个指标是衡量一个城市绝对拥有量的重要指标，也即单位出租车数量，用公式表示为：

$$万人拥有量=\frac{拥有出租车总量数}{人口总数}$$

平均被抢单时间：这个指标定义为从下单到单被接受的时间间隔。这段时间可以从滴滴打车平台上获取数据。它可以反映供求之间的情况，公式为：

$$平均接单时间t=\frac{全市接单时间总数T}{全市接单总数}$$

空车率：空车率定义为出租车空车的数量，从这个指标中可以得到一个城市出租车是否得到了充分的使用，可以有效里程率来反映空车率，即：

空车率=1-有效里程率

其中有效里程率表示有效里程数（载客时的里程数）和总里程数级之间的比值。

成单率：成单率是度量供求关系的重要关系，它能很大程度的反应两者匹配的情况，因为只有匹配成功，才能称作成单。所以它是一个重要指标，而且发现该数据和时间与地点由较大关系。

在微观模型的建立中首先注意到指标集的重要性并不完全相同，所以要根据不同指标的重要习惯确定它们的权重，其次该模型中最大的想法就是供求匹配是较泛的概念，若要量化可用司机与乘客的满意度进行综合排序，得到供求匹配程度。（供求匹配程度越高，乘客与司机的综合满意度也就越高，而若供求匹配度低，那么满意度也就低）

4.3 权重的确定模型

对于权重的确定采取主观与客观相结合的权重确定模式。主观方面采取层次分析法，客观方面采用熵权法，最后结合起来得到综合的权重，这样的权重更具权威性。

（1）主观的赋权法：层次分析法

Step 1 层次分析法是一种确定权重的重要方法，建立目标层即供求匹配，在建立四个准则层，即这里的指标集，将这四个指标的重要性两两进行比较，用表示 x_{ij} 第 i 因素对于第 j 个因素的重要性的值程度(这里的比较值的确定采用 Saaty 教授提出的 1-9 标度，即 1 表示同等重要，3 表示 x_i 比 x_j 稍微重要，5 表示 x_i 比 x_j 明显重要，7 表示 x_i 比 x_j 强烈重要，9 表示 x_i 比 x_j 极端重要，2，4，6，8 介于上述两个相等等级之间。这样就可以得到一个 n 阶的成对比较矩阵 $X=(x_{ij})_{n*n}$,且其中 $x_{ij}=1/x_{ij}$， $x_{ii}=1, x_{ij}>0$， $i, j=1,2,3,4$；

Step 2 当得到正互反判断矩阵之后，先需要进行一致性的分析，需要先确定出 $step1$ 中

正互反矩阵的特征值和特征矩阵，取特征矩阵的最大值 $l_{\max}$ ，这里使用 Satty 教授定义的一致性指标及其计算公式：

$$CI=(l_{\max}-n)/(n-1),$$

其中ＣＩ表示偏离程度，若 $CI=0$ ，Ａ表示具有完全一致性，为了更容易进行判断，将 CI 与和它同阶的 RI 做比值，得到 CR ， CR 称作一致性比率，若 $CR<0.1$ ，那么 X 矩阵通过一致性检验，否则再重复 $step1, step2$ ；

Step 3 将符合一致性检验的矩阵 X 进行归一化 $W=(w1,w2,w3,w4)$ 。

(2) 信息熵决策模型的建立

熵的概念最早由 Clausius 于 1865 年提出，之后信息论之父 C.E.Shannon 在 1948 年将熵的概念引入信息领域，利用“信息熵”作为衡量信息紊乱程度的测度。熵的概念从宏观上反映了系统在微观状态下的不确定性程度。

信息熵据决策模型的建立

Step1 构建决策矩阵 U ；

Step2 属性的规范化，求决策矩阵的规范化矩阵 R ， r_{ij} 为 R 中的元素，规范化原则：

$$\text{正指标：}\ r_{ij}=\frac{a_{ij}}{\max\limits_{i}\left\{a_{ij}\right\}}$$

$$\text{负指标：}\ r_{ij}=\frac{\min\limits_{i}\left\{a_{ij}\right\}}{a_{ij}}$$

评估指标依性质和作用分为正指标和逆指标。指标数值大小与司法绩效高低一致的，是正指标；指标数值大小与司法绩效高低相反的，是逆指标。

Step3 规范化矩阵 R 归一化， $\tilde{R}=(\tilde{r}_{ij})_{n\times m}$

$$\text{其中：}\ \tilde{r}_{ij}=\frac{r_{ij}}{\sum_{i=1}^{n}r_{ij}}$$

Step4 计算属性 u_j 输出的信息熵：

$$E_j=-\frac{1}{\ln n}\sum_{i=1}^{n}\tilde{r}_{ij}\ln\tilde{r}_{ij}$$

Step5 计算属性权向量：

$$w_j=\frac{1-E_j}{\sum_{k=1}^{m}(1-E_k)}$$

Step6 计算综合属性值 z_i：

$$z_i = \sum_{j=1}^{m} r_{ij} w_j$$

Step7 利用 z_i 的大小对方案进行排序和择优。

4.4 度量匹配程度模型——TOPSIS 方法

因为供求匹配程度直接关系到乘客和司机，匹配程度高，司机可以有选择的接单，空车率大幅下降，乘客下单很快能够上车，那么乘客的满意度和司机的满意度就会上升；若供求匹配程度很差，则反之。所以本文用司机和乘客的满意度来代替匹配程度，将四个评判矩阵分为成本型，价值型。

TOPSIS 算法又称为逼近于理想解的一种排序方法，基本用于解决一个 n 方案和 m 个属性的多属性决策问题，可以设 x^* 为理想解，x^- 为负理想解，通常使用方案和理想解之间的相对贴近度进行判断解的优劣程度。

Step1 本次研究中有 9 个目标，分别对应九个城市，其中一线城市为北京、上海、广州，二线城市为南京、合肥、杭州，三线城市为台州、三亚、吉林，4 个属性分别是城市出租车万人拥有量、平均接单时间、空车率、成单率。其中，第 i 个城市对第 j 个属性的评估值为 x_{ij}，则初始判断矩阵 U 为：

$$U = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ x_{i1} & \cdots & x_{ij} & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}$$

Step2 城市对应的指标量纲不同，需要对决策矩阵进行归一化处理：

$$R = \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ r_{i1} & \cdots & r_{ij} & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ r_{m1} & x_{m2} & \cdots & r_{mn} \end{pmatrix}$$

其中： $r_{ij} = r_{ij} / \sqrt{\sum_{k=1}^{n} r_{ij}^2}, i = 1, 2 \ldots m, j = 1, 2 \ldots n$

Step3 根据层次分析法获取城市属性信息矩阵权重 W，形成判断加权判断矩阵：

$$Z = R*W = \begin{pmatrix} r_{11} & \cdots & r_{1n} \\ \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{mn} \end{pmatrix} * \begin{pmatrix} w_1 & \cdots & 0 \\ \vdots & w_2 & \vdots \\ 0 & \cdots & w_3 \end{pmatrix}$$

$$Z = \begin{pmatrix} f_{11} & \cdots & f_{1n} \\ \vdots & \ddots & \vdots \\ f_{m1} & \cdots & f_{mn} \end{pmatrix}$$

Step4 计算各目标值与理想值之间的欧氏距离:

$$S_i^* = \sqrt{\sum_{j=1}^{m} (f_{ij} - f_j^*)^2}, j = 1, 2, ..., n,$$

Step5 计算各目标的相对贴近度

$$C_i^* = S_i^{'} / (S_i^* + S_i^{'}), i = 1, 2, ..., m.$$

Step6 依照相对贴近度的大小对目标进行排序，形成决策依据。

4.5 模型的实证分析

（1）北京市 24 小时贴近度及其排名

表一 北京市贴近度计算结果

时间	1:00	2:00	3:00	4:00	5:00	6:00	7:00	8:00
贴近度	0.0003	0.0443	0.0200	0.0229	0.0209	0.0571	0.01936	0.3678
时间	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00
贴近度	0.5186	0.3811	0.2088	0.2230	0.2412	0.2849	0.3082	0.2989
时间	17:00	18:00	19:00	20:00	21:00	22:00	23:00	24:00
贴近度	0.3547	0.4656	0.7868	0.9997	0.5364	0.7602	0.7046	0.3990

图一 24 小时贴近度变化曲线

将这些数据按照贴近度大小进行排序可以得到排序：

表二 北京 24 小时匹配度排名

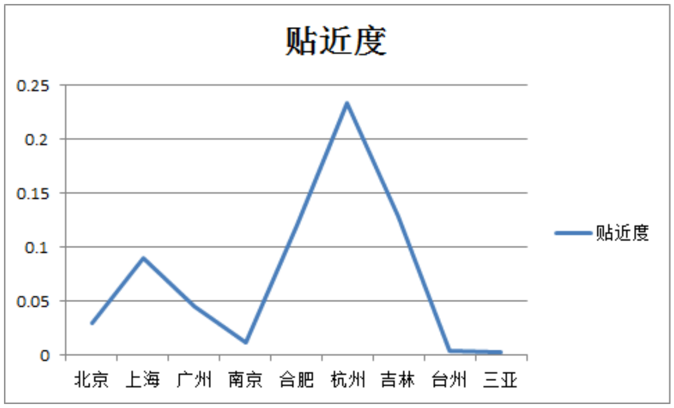
排名	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
时刻	20	19	22	23	21	9	18	24	10	8	17	15	16	14	13	12	11	6	2	4	5	3	7	1

通过北京市 24 个小时的排序匹配度较高的一般为夜间在 19~23 点之间的这段时间，结合实际情况可以看出模型求解出来的结果合理性还是很强的，一方面在该时间段刚好错过了下班放学的高峰期，这个时候打车的人相较之 17：00~19:00 这段时间要少一些，打车相对较容易些，这样的话匹配度就高一些；另一方面和上下午相比，这个时候的匹配度高也从一定角度反映出北京市的夜生活的繁荣度。而凌晨以后城市相对安静，街道上人车流量都很少，这时的匹配度可定会降低。另外从图一（24 小时贴近度变化曲线）基本上可以看做两个高峰（21 点可认为是误差造成的），且高峰形成时间都是在上下班高峰期之后的一到两个小时之间，类似前面的分析，可看出模型从时间的角度来说是非常合理的。

（2）九座代表性城市的贴近度以及排名

表三 九座城市 13:00 贴近度

城市	北京	上海	广州	南京	合肥	杭州	吉林	台州	三亚
贴近度	0.03	0.089	0.0449	0.0115	0.1207	0.2323	0.1280	0.0037	0.0026



图二 九座城市贴近度曲线

表四 九座城市的贴近度排名

排名	1	2	3	4	5	6	7	8	9
城市	上海	杭州	吉林	合肥	广州	北京	南京	台州	三亚

可以看出一线城市中上海的匹配度最高，而北京道路相较之上海道路拥挤的多，而且北京市的单双号限号，一定程度上限制了匹配程度，使得打车变难。另外杭州属于旅游城市，人流量大且一般外来旅游人口，在不认识路的情况下，基本上会选择出租车。而涉及空间的变化情况，受多方面因素影响，且本城市不同时间也会不同，而的数据只是选择了某一天某一时间点下的数据具有很大的局限性。只能大概看出有限的规律。

5.问题二模型的建立与分析

问题二中各公司补贴方案对缓解打车难的帮助问题，通过调查国内几家公司发现，因为各个打车公司的补贴方案并无实质性的差距，基本可分为两种，一种是通过客户端返现，和发放满减红包。且在实际问题中可以看出司机与乘客之间并无约束关系，所以双方是否响应打车软件的补贴政策是自由的，这一点就符合了非合作博弈模型的条件。将在双方非合作博弈模型之下来探讨司机与乘客在响应打车软件时的博弈模型，找到其 Nash 平衡点，

5.1 非合作博弈模型

（1）首先确定该实际问题的局中人集合，因为在该题中考虑打车难，是乘客与司机中间的利益问题，并未涉及到打车软件公司，所以局中人集合为 I={1,2}，其中局中人 1 表示司机，局中人 2 表示乘客。

（2）确定每个局中人的策略集 $S_i = \{s_1, s_2, ..., s_n\}$，

对于局中人 1（司机）其有两种选择即响应打车软件公司的补贴政策 s_1 和不响应打车软件公司的补贴政策 s_2，构成局中人 1（司机）的策略集 $S_1=\{s_1,s_2\}$。

对于局中人 2（乘客）来说，也有两种选择安装软件并使用或者对补贴政策视而不见构成局中人 s_2，乘客的策略集 $S_2=\{s_1,s_2\}$。

设 $P=(x,1-x)$，$Q=(y,1-y)$ 分别表示局中人 1(司机）和局中人 2（乘客） 的混合策略，x 表示响应打车软件所提供的补贴政策的概率，那么 $1-x$ 表示不响应打车软件所提供的补贴政策，y 表示乘客下载并响应打车软件所提供的补贴政策的概率，$1-y$ 相对应于不响应打车软件所提供的补贴政策。这样博弈模型的混合策略的整体局势就可以由 (x,y) 完全决定。

（3）确定局中人 1（司机）和局中人 2（乘客）的赢得矩阵 $H_i=(h_{ij}(s))(i,j=1,2)$,其中矩阵中的 $h_{ij}(s)$ 表示在局中人 1 选择策略 i 和局中人 2 选择策略 j 的情况之下局中人 i 的赢得， 假设局中人 1（司机）和局中人 2（乘客）的赢得矩阵为 ：

$$A=\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, \quad B=\begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix},$$

则将整个双人非合作博弈模型记作 $G=\{I,\{S_i\},\{H_i\}\}\ i=(1,2)$。

设 $E_1(x,y)$ 和 $E_2(x,y)$ 分别表示局中人 1（司机）和局中人 2（乘客）在混合策略 (x,y) 下获得的期望赢得，则

$$E_1(x,y)=(x,1-x)\mathbf{A}\begin{pmatrix} y \\ 1-y \end{pmatrix}=\mathbf{XAY}^T \tag{5.1.1}$$

$$E_2(x,y)=(x,1-x)\mathbf{B}\begin{pmatrix} y \\ 1-y \end{pmatrix}=\mathbf{XBY}^T \tag{5.1.2}$$

根据公式（5.1.1)和公式（5.1.2）可以计算出 Nash 平衡点。

(x,y) 为博弈的平衡点即为所有混合策略之下的最大值，这里为双人非合作的博弈问题，故而可以将 $Nash$ 平衡点的定义转化为如下的几个不等式：

$$E_1(1,y)\le E_1(x,y) \tag{5.1.3}$$

$$E_1(0,y)\le E_1(x,y) \tag{5.1.4}$$

$$E_2(x,1)\le E_2(x,y) \tag{5.1.5}$$

$$E_2(x,0) \le E_2(x,y) \tag{5.1.6}$$

由(5.1.3)-(5.1.4) 式展开化简得：

$$Q(1-x)y - q(1-x) \le 0 \tag{5.1.7}$$

$$Qxy - qx \ge 0 \tag{5.1.8}$$

其中 Q=$a_{11}+a_{22}-a_{21}-a_{12}, q=a_{22}-a_{12}$，则有：

当 Q=0，q=0 时， $0 \le x \le 1, 0 \le y \le 1$

当 Q=0，q>0 时， $x=0, 0 \le y \le 1$

当 Q=0，q<0 时， $x=1, 0 \le y \le 1$

当 $Q \ne 0$ 时，记 $a=\dfrac{q}{Q}$，则有

$$\begin{aligned} &x=0, y \le a \\ &0<x<1, y=a \\ &x>1, y \ge a \end{aligned}$$

同理，由(5.1.5)- (5.1.7)式展开化简得：

$$Rx(1-y) - r(1-y) \le 0 \tag{5.1.9}$$

$$Rxy - ry \ge 0 \tag{5.1.10}$$

其中 R=$b_{11}+b_{22}-b_{21}-b_{12}, r=b_{22}-b_{12}$，对 (5.1.9)- (5.1.10)求解，则有：

当 R=0，r=0 时， $0 \le x \le 1, 0 \le y \le 1$

当 R=0，r>0 时， $0 \le x \le 1, y=0$

当 R=0，r<0 时， $0 \le x \le 1, y=1$

当 $Q \ne 0$ 时，记 $b=\dfrac{r}{R}$，则有

$$\begin{aligned} &x \le b, y=0 \\ &x=b, 0<y<1 \\ &x \ge b, y=1 \end{aligned}$$

将上述不等式组的解结合起来就得到双矩阵博弈的平衡点。

5.2 模型的实证分析

在该模型中，需要确定乘客和司机的赢得矩阵，对于司机来说当司机与乘客都下载使用

该软件时，司机可以与乘客完成配对。司机从而获得相应的补贴，经过查询数据知道司机的收益相较之无补贴政策每天增多 100~200 元的收入。这是记其盈利为 2，而若司机使用打车软件，乘客不用，或者是乘客使用，司机不用，两者无法配对，此时记其的赢得应为-1，即实际意义就是表示，司机大部分时间在空车运营，司机当天拉客所赚的钱小于他的油量消耗。

若是乘客和司机都不使用，即与无打车软件补贴措施时的情况无异，这是的赢得是 1，这样就得到了局中人 1（司机）在不同策略下的赢得，构成的赢得矩阵[7]如下：

$$A=\begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix}$$

而对于乘客来说出租车司机与乘客均响应打车软件的补贴措施，那么乘客一方面节省了路费另一方面打车时间平均减少了 3-4 倍。另外因为软件公司对于出租车偏远地带补贴，使得乘客更加方便，所以记乘客赢得为 3，在司机与乘客信息不统一的情况之下，（司机使用，乘客未使用和乘客使用司机未使用），乘客要消耗大量时间用于等车且花费很贵。这时记司机使用乘客未使用的赢得为-1，乘客使用司机不使用的赢得为-1，司机乘客都不使用时为 0，这样又得到一个赢得矩阵为

$$B=\begin{pmatrix} 3 & -1 \\ -1 & 1 \end{pmatrix}$$

将这个（一）中的公式讨论可以得到：

$$Q=5>0, q=2, a=q/Q=2/5$$

$$R=6>0, p=2, b=p/R=1/3$$

这样就得到双矩阵博弈的解为：

$$\begin{cases} x=0, y\le\dfrac{2}{5} \\ 0<x<1, y=\dfrac{2}{5} \\ x=1, y\ge\dfrac{2}{5} \end{cases}, \qquad \begin{cases} x\le\dfrac{1}{3}, y=0 \\ x=\dfrac{1}{3}, 0<y<1 \\ x\ge\dfrac{1}{3}, y=1 \end{cases}$$

这样 Nash 平衡点有三个分别为：（2/5,1/3），在建立模型之前就分析了这个问题为非合作的博弈模型，这样就得到了 Nash 平衡点为（2/5,1/3）。

上述结果表明：司机响应补贴政策的概率[8]的响应度为 2/5,乘客对补贴模型的响应度为 1/3，即出租车司机与乘客响应补贴政策的概率均大于 0，说明打车软件的补贴政策确实有助于缓解打车难的问题，且司机响应度高于乘客。

6.问题三模型的建立与分析

即从合作博弈的角度来审视补贴方案的合理性问题。因为打车软件公司、出租车司机、乘客三个主体之间有竞争关系，同时也有利益共享的关系，因此“互联网+”时代的出租车补贴行为可以视为一个合作博弈。本文认为一个合理补贴方案的设计实质上一个利益分配关系。文中应用合作对策中 Shapley 值分配分方案，计算出打车软件公司、出租车司机、乘客应该分享合理的收益比例。

6.1 合作 *3* 人对策模型 [9]

本文的 3 人合作对策中，由打车软件公司、出租车司机、乘客三个局中人在采用打车软件上进行合作，他们结成一个联盟，作为一个整体，希望得到尽可能多的收入，这个联盟要把得到的总收入分配给联盟中的每一个成员。

定义 1 [4]设 $N=\{1,2,3\}$ 为局中人集，$V(S)$ 是定义在 I 上的一切子集（即联盟）所形成的集合 2^N 上的映射满足：

(1) $V(\varnothing)=0$ （6.1.1）

(2) $\forall S,T\in 2^N$，则 $V(S\cup T)\supseteq V(S)+V(T)$，其中 $S\cap T=\varnothing$ （6.1.2）

则称 $\Gamma=[I,V]$ 为合作 3 人对策，$V(S)$ 为对策的收益函数。

定义 2 [4]称 $v(\mathrm{s}\cup\{\mathrm{i}\})-v(\mathrm{s})$ 为第 *i* 局中人对联盟 s 合作的“贡献”，其中 $s\subset N$ 。

设 x_{i} 为第 i 局中人从 $v(N)$中获得的分配，i=1，2，3，则有：

$$
\begin{aligned}
x_1&=v(\{1\}),\\
x_2&=v(\{1,2\})-v(\{1\}),\\
x_3&=v(\{1,2,3\})-v(\{1,2\}),
\end{aligned}
\qquad (6.1.3)
$$

上述的分配通常与局中人编号的次序有关，如把局中人 3，2，1 的编号改为$1^{'},2^{'},3^{'}$，则有新的分配方案：

$$
\begin{aligned}
x_1^{'}&=v(\{3\}),\\
x_2^{'}&=v(\{3,2\})-v(\{3\}),\\
x_3^{'}&=v(\{1,2,3\})-v(\{2,3\}),
\end{aligned}
\qquad (6.1.4)
$$

对于其它的编号的次序有对应的分配方案，由于 3 个局中人编号的次序共有 3!种，取各局中人分配的平均值作为局中人的平均“贡献”。

记$j_{\rm i}(v)$ 为第 i 个局中人的平均“贡献”，则有：

$$j_{\rm i}(v)=\frac{1}{{\rm n}!}\sum_{p}[v(s_p^i\cup\{{\rm i}\})-v(s_p^i)\ ] \qquad (6.1.5)$$

其中p 由 1，2，3 组成的 3 级排列，$s_p^i=\{j\,|\,p\ j<i\}$。

将满足s_p^i =s 排列归为一类，（6.1.5）式可以表示为：

$$j_i(v)=\sum_{i\in s}\frac{(n-|s|)!(|s|-1)}{n!}[v(s)-v(s-\{i\})\ ],i=1,2,3 \qquad (6.1.6)$$

其中$|s|$子集 s 中局中人的人数。

称$\Phi(v)=(j_1(v),j_2(v),j_3(v))$为合作 *3* 人对策的 Shapley 值。

设局中人 1 表示打车软件公司，局中人 2 为出租车司机，局中人 3 为乘客。记$V(S)$表示联盟S的合作可以获得收益，

设三个局中人共同合作可以获得 1 个单位的最大收益，即$V(\{1,2,3\})=1$，

若三个局中人互不合作时，则$V(\{i\})=0,i=1,2,3$。

若打车软件公司和出租车司机合作，两者获得收益的比例为$a_1(0<a_1<1)$，

若出租车司机和乘客两者合作，两者获得收益的比例为$a_2(0<a_2<1)$，

若打车软件公司和乘客两者合作，两者获得收益的比例为$a_3(0<a_3<1)$，

即三个局中人合作的收益特征函数[10]为

$$\begin{cases}V(\{i\})=0,i=1,2,3,\\ V(\{1,2\})=a_1,0<a_1<1\\ V(\{1,3\})=a_2,0<a_2<1\\ V(\{2,3\})=a_3,0<a_3<1\\ V(\{1,2,3\})=1\end{cases}$$

由 Shapley 值计算式可以获得相应的收益比例为，此为补贴的比例结果。
打车软件公司应该分享的收益比例为：

$$j_1(v)=v(\{1\})\times\frac{1}{3}+[v(\{1,2\})-v(\{2\})]\times\frac{1}{6}+[v(\{1,3\})-v(\{3\})]\times\frac{1}{6}$$
$$+[v(\{1,2,3\})-v(\{2,3\})]\times\frac{1}{3}=\frac{a_1+a_2+2(1-a_3)}{6}$$

出租车司机应该分享的收益比例为：

$$j_2(v)=v(\{2\})\times\frac{1}{3}+[v(\{2,1\})-v(\{1\})]\times\frac{1}{6}+[v(\{2,3\})-v(\{3\})]\times\frac{1}{6}$$
$$+[v(\{1,2,3\})-v(\{1,3\})]\times\frac{1}{3}=\frac{a_1+a_3+2(1-a_2)}{6}$$

乘客应该分享的收益比例为：

$$j_3(v)=v(\{3\})\times\frac{1}{3}+[v(\{3,1\})-v(\{1\})]\times\frac{1}{6}+[v(\{2,3\})-v(\{2\})]\times\frac{1}{6}$$
$$+[v(\{1,2,3\})-v(\{1,2\})]\times\frac{1}{3}=\frac{a_2+a_3+2(1-a_1)}{6}$$

显然 $\sum_{i=1}^{3} j_i(v)=1$

根据模拟计算，若令 $a_1=\frac{9}{10},a_2=\frac{4}{5},a_3=\frac{1}{5}$，

从而可计算出：

$$j_1(v)=0.05\ ,\ j_2(v)=0.85\ ,\ j_3(v)=0.10$$

7.模型优缺点分析

优点：

（1）主观、客观相结合确定权重，使权重更加的符合实际。

（2）将题目中的供求匹配转换为与最优解距离之间的关系

（3）将问题转化为非合作博弈和合作博弈求最优解，可以解决部分数据不足的问题，且模型应用有创新。

缺点：

（1）数据过于单一，对于空间里的匹配度只选取了某一个时间点，有一定的随机性

（2）影响因素指标过多，选取的有限。完全不考虑司机与乘客之间的交流问题，）确定赢得矩阵有一定的主观性。

（3）出租车打车费受多因素的影响，带有诸多不确定的因素，如打车途中的道路状况问题和当天的天气状况，难以用模型反映出来这些因素对于打车费的影响，所得到的数据只可以作为一种参考。

8. 对于“打车难”现象的若干政策建议

“打车难”问题是全社会都需要关注的问题。本文通过建立数学模型进行量化分析，得出一些重要结论。为此向有关部门提供如下建议报告。

（1）对于打车软件公司提供适当补助，使打车软件可以更加稳健的经营下去。政府部门必须充分考虑补贴政策的有效性，确定一个合适的对于打车软件的补助标准。

（2）增大宣传力度，让更多的居民学会使用打车软件。政府可以通过发行广告的方式，使得居民了解这种新型打车方式，都可以体验到这种方式带给居民生活的新体验，形成“互联网+”时代的新的打车思路，从而缓解“打车难”的问题。

（3）注重对老弱病残者的关怀。政府可让弱势群体在等车通过拨打专门的电话，启用专门的出租车，前来接送，是不会使用网上预约和支付的社会弱势群体可以通过较传统的方式打车，它有利于用有限资源解决更多人的困难，可以解决社会弱势群体的部分打车困难，为他们创造更便捷的打车方式。

（4）长远来看，出租车公司应以市场供求为主要参考，逐步转变成打车软件公司平台、乘客和出租车公司三者相结合。打车软件公司平台、乘客和出租车公司三者之间的关系将会成为决定车费的依据，这不失为一条可以选择的道路。

（5）借鉴西方科技发达国家的打车模式，结合中国国情做出改革根据中国国情，适当调整大型软件开发公司的比例，适当补住一些微校打车软件公司，以此获得平衡，寻求各规模打车软件公司共同发展。

（6） 政府加强运营秩序监管，上调运价，让市民为“打车难”买单。

（7） 政府应调整市场竞争机制，引入新的经营主体，让市场自动调节出租车的收入和换班问题。

参考文献

[1] 衡量出租车供求的三大指标—里程利用率,车辆满载率,万人拥有量 [J].运输经理世界，2007,5（5）

[2] 陈明艺. 出租车数量管制的合理性分析级评估机制研究 [J]. 中国物价,2006,30（8）

[3] 农产品供求信息智能匹配关键技术研究 [J]. 2012（6）

[4] 陈华友, 周礼刚 ,刘金培. 数学模型与数学建模 [M]. 北京：科学出版社.

[5]中国统计年鉴,http://www.stats.gov.cn/.

[6]滴滴快的智能出行平台,v.kuaidadi.com

[7]姜启源,谢金星，叶俊.2011.数学模型[M].第四版.北京：高等教育出版社.

[8]费培之.1998.数学模型实用教程[M].成都:四川大学出版社。

[9]彭勇行.2000.管理决策分析[M].北京:科学出版社.

[10]王连芬,许树柏.1990.层次分析法引论[M].北京:中国人民大学出版社.

附录一（数据）

表 1　24h 打车需求量数据

北京	上海	广州	南京	合肥	杭州	三亚	吉林	台州
3001	2737	1031	447	339	1267	27	158	28
1181	1556	469	338	237	931	29	63	4
534	1084	286	180	254	608	17	105	12
610	667	292	121	154	416	5	59	19
556	731	141	105	143	422	0	79	0
1522	2169	159	342	291	747	39	214	14
5158	7191	323	828	868	1671	19	921	27
9797	1744	733	2029	3065	3544	45	1561	70
13814	30631	1700	3045	3133	5619	127	1041	70
10152	23132	1394	1899	1720	3996	90	397	101
5561	10872	1314	1881	1450	3070	128	377	98
5940	10626	1552	1916	1519	3664	123	442	72
6426	11530	2083	1886	1610	3371	95	303	79
7589	14454	1855	1999	2165	3784	60	420	85
8209	17243	1821	2087	2173	4693	161	367	87
7962	18266	2185	2316	2110	6123	69	166	200
9449	19361	2274	2377	3341	6487	154	693	158
12402	29537	2794	4193	5892	7402	129	1026	198
20960	30230	3030	4163	8934	6997	140	572	120
26638	19307	2008	2425	3425	4320	117	515	93
14288	16264	1779	2173	1672	3713	132	440	94
20250	21058	2124	2117	1988	4149	111	340	66
18769	21030	2451	1497	1306	2737	63	216	25
10629	15940	2231	1005	580	1838	38	144	18

表 2 九座城市 13:00 数据比较

2014 年	人口数(万人)	出租车数（辆）	万人拥有量	成单率
北京	1316.34	75000	56.976161	0.9
上海	1432.34	45000	31.417122	0.91
广州	832.31	33000	39.648689	0.88
合肥	711.5	8500	11.9	0.87
南京	643.09	13832	21.508654	0.9
杭州	706.61	11590	16.402259	0.88
三亚	53.6	1020	19.029851	0.85
吉林	430	5000	11.627907	0.86
台州	548	2400	4.379562	0.79

表 3 北京市 24h 数据变化

需求	万人拥有量	成单率	时间
3001	48.3	0.78	48.25
1181	48.3	0.8	33.34
534	48.3	0.79	35.65
610	48.3	0.87	44.7
556	48.3	0.79	34.01
1522	48.3	0.88	40
5158	48.3	0.9	54.72
9797	48.3	0.77	66.14
13814	48.3	0.67	60.23
10152	48.3	0.55	55.36
5561	48.3	0.67	40.14
5940	48.3	0.77	39.66
6426	48.3	0.67	40.43
7589	48.3	0.55	46.9
8209	48.3	0.67	48.51
7962	48.3	0.65	44.57
9449	48.3	0.53	47.18
12402	48.3	0.66	57.67
20960	48.3	0.7	66.42
26638	48.3	0.8	70.85
14288	48.3	0.94	58.76
20250	48.3	0.8	75.13
18769	48.3	0.76	76.96
10629	48.3	0.64	59.1

附录二（程序）：

```
%%%%%%%%%%%%%%层次分析法%%%%%%%%%%%%%%%
close all
clear all
clc
u1=[1,3,5,7
     1/3,1,3,5
     1/5,1/3,1,3
     1/7,1/5,1/3,1];
[v,d]=eig(u1);
value=diag(d);
C1=max(value);
ci1=(C1-4)/3;
cr1=ci1/0.89;
abw1=v(:,1)/sum(v(:,1));
abw1
cr1

%%%%%%%%%%%%%%北京不同时间的信息熵决策%%%%%%%%%%%%%%%
close all
clear all
clc
T=xlsread('C:\Users\air\Desktop\同一时间段\数据统计\北京','B1:E9');
m=length(T(1,:));
n=length(T(:,1));
for j=1:m
    for i=1:n
        r(i,j)=T(i,j)/max(T(:,j));
    end
end
for j=1:m
    rr(j)=sum(r(:,j));
end
for j=1:m
    for i=1:n
        R(i,j)=r(i,j)/rr(j);
    end
end
```

```
E=zeros(1,m);
for j=1:m
    for i=1:n
        E(j)=R(i,j)*log(R(i,j))+E(j);
    end
end
for j=1:m
    E(j)=E(j)*(-1/log(n));
end
mm=sum(1-E);
for j=1:m
    w(j)=(1-E(j))/mm;
end
w=w.'
for i=1:n
    z(i)=r(i,:)*w;
end
for i=1:n
    z(i)=z(i)/sum(z);
end

%%%%%%%%%%%%%%不同地点的信息熵决策%%%%%%%%%%%%%%
close all
clear all
clc
T=xlsread('C:\Users\air\Desktop\数据统计\数据','B1:E9');
m=length(T(1,:));
n=length(T(:,1));
for j=1:m
    for i=1:n
        r(i,j)=T(i,j)/max(T(:,j));
    end
end
for j=1:m
    rr(j)=sum(r(:,j));
end
for j=1:m
    for i=1:n
        R(i,j)=r(i,j)/rr(j);
```

```
        end
end
E=zeros(1,m);
for j=1:m
        for i=1:n
                E(j)=R(i,j)*log(R(i,j))+E(j);
        end
end
for j=1:m
        E(j)=E(j)*(-1/log(n));
end
mm=sum(1-E);
for j=1:m
        w(j)=(1-E(j))/mm;
end
w=w.'
for i=1:n
        z(i)=r(i,:)*w;
end
for i=1:n
        z(i)=z(i)/sum(z);
end

%%%%%%%%%%%%%%%%   九 座 城 市 的 比 较 直 方 图
%%%%%%%%%%%%%%%%%%%%
close all
clear all
clc
x=1:1:9;
subplot(2,2,1)
y1=[5159 7191 323 828 868 1671 921 27 19];
bar(x,y1,'b');
title('九座城市同时间段打车需求量对比')
xlabel('北京、上海、广州、南京、合肥、杭州、吉林、台州、三亚');
ylabel('打车需求量/辆');
subplot(2,2,2)
y2=[54.72    49.12    40.13    33.337   45.26    30.51    39   32.86    23.14];
bar(x,y2,'b');
title('九座城市同时段平均等车时间对比')
```

```
xlabel('北京、上海、广州、南京、合肥、杭州、吉林、台州、三亚');
ylabel('等车时间/s');
subplot(2,2,3)
y3=[48.3 32.41      18.21      15.26      29.29      20    12    11    10];
bar(x,y3,'b');
title('九座城市万人拥有量对比')
xlabel('北京、上海、广州、南京、合肥、杭州、吉林、台州、三亚');
ylabel('万人拥有量/辆');
subplot(2,2,4)
y4=[0.9   0.91 0.88 0.87 0.9   0.88 0.85 0.86 0.79];
bar(x,y4,'b');
title('九座城市同时段成单率对比');
xlabel('北京、上海、广州、南京、合肥、杭州、吉林、台州、三亚');
ylabel('成单率百分比%')

%%%%%%%%%%%%%%%%%%%%% 北 京 市  24  小 时 变 化 比 趋 势 图 %%%%%%%%%%%%%%
close all
clear all
clc
x=1:1:24;
subplot(2,2,1)
y1=[3001 1181 534 610 556 1522 5158 9797 13814 10152 5561 5940 6426 7589 8209 7962 9449
12402 20960 26638 14288 20250 18769 10629];
plot(x,y1,'-');
title('北京 24 小时打车需求量变化');
xlabel('时间/h');
ylabel('打车需求量/辆');
subplot(2,2,2)
y2=[48.25      33.34      35.65      44.7 34.01      40    54.72      66.14      60.23      55.36
      40.14      39.66      40.43      46.9 48.51      44.57      47.18      57.67      66.42      70.85
      58.76      75.13      76.96      59.1];
plot(x,y2,'-')
xlabel('时间/h');
ylabel('等车时间/s')
title('北京 24 小时内平均等待时间')
subplot(2,2,3)
y3=48.3;
plot(x,48.3,'-')
```

```
xlabel('时间/h')
ylabel('万人拥有量/辆');
title('北京市万人拥有辆');
subplot(2,2,4)
y4=[0.78 0.8   0.79 0.87 0.79 0.88 0.9   0.77 0.67 0.55 0.67 0.77 0.67 0.55 0.67 0.65 0.53 0.66 0.7
    0.8   0.94 0.8   0.76 0.64];
plot(x,y4,'-')
xlabel('时间/h')
ylabel('成单率%');
title('北京市成单率');

%%%%%%%%%%%%%%%%%%TOPSIS 算法函数%%%%%%%%%%%%%%%%%
function [output_args ] = TOPSIS( A,W,M,N )
[ma,na]=size(A);
  for j=1:na
       for i=1;ma
            A(i,j)=A(i,j)/sum(A(:,j))
       end
  end
for i=1:na
       B(:,i)=A(:,i)*W(i);
end
V1=zeros(1,na);
V2=zeros(1,na);
BMAX=max(B);
BMIN=min(B);
for i=1:na
       if i<=size(M,2)
       V1(M(i))=BMAX(M(i));
       V2(M(i))=BMIN(M(i));
       end
       if i<=size(N,2)
       V1(N(i))=BMIN(N(i));
       V2(N(i))=BMAX(N(i));
       end
end

for i=1:ma
       C1=B(i,:)-V1;
```

```
    S1(i)=norm(C1);
    C2=B(i,:)-V2;
    S2(i)=norm(C2);
    T(i)=S2(i)/(S1(i)+S2(i));
end
T
```