

“互联网+”时代的出租车资源配置

摘要

本文主要研究的是互联网时代的出租车资源配置问题。建立模糊综合评判模型得到不同城市不同年份“供求匹配”程度综合指标，建立排队论模型得到补贴额度与等待时间的函数关系，建立多元线性规划模型得到合理的补贴方案。

针对问题一，建立模糊综合评价模型分析出租车资源“供求匹配”程度。首先提出了里程利用率，车辆满载率，万人拥有量三个指标描述“供求匹配”程度。接着采用层次分析法得到三个指标的权重，与模糊评判矩阵相乘得到综合指标。利用 BP 神经网络预测缺失统计数据后，计算 2010 到 2015 年 15 个城市的综合指标，其中 2014 年宁波供求匹配程度最好，综合指标为 0.001856，2013 年北京供求匹配程度最差为 0.17182。得到城市越发达供求越紧张，以及 2012，2013 年打车软件出现缓解发达城市供求紧张的结论。

针对问题二，建立多服务台排队模型求解补贴方案能否缓解打车难。首先得到每日载客数与乘客补贴额度的一次函数，接着得到可使用出租车数与司机补贴额度的一次函数，进而得到载客里程与司机补贴额度的函数。根据排队论乘客到达率，出租车服务率，等待时间的定义，得到等待时间指标与补贴额度的函数，发现补贴额度越大，等待时间指标越小打车越容易。计算同期滴滴打车和快的打车补贴方案的等待时间指标分为 0.0083, 0.0016，证明快的打车方案更能缓解打车难。

针对问题三，建立多元线性规划模型得到补贴方案。以第二问中等待时间最小为目标函数，平台收益大于补贴投入为约束条件。得到平时补贴乘客 3.02 元，司机 7.57 元的方案，方案等待时间指标为 0.0039，收益指标为 43.2，节假日补贴乘客 1.86 元，补贴司机 5.27 元的方案，方案等待时间指标为 0.0106，收益指标为 9，第二问滴滴打车的等待时间指标为 0.0083，平日收益指标为-33，节假日收益指标为-90，比较可知我们的方案等待时间短，收益大，因此方案合理。

对问题一城市分类的合理性进行检验，计算同类城市的 5 年指标相关性发现同类城市高度相关，分类合理。对问题二等待时间指标与补贴额度的指标函数模型进行灵敏度分析，改变函数系数，发现乘客补贴对于系数的改变更为敏感。

关键词：模糊综合评判 层次分析法 BP 神经网络 排队论 线性规划

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

对于问题一，由题目可知，这是一个关于建立评价体系的问题。解决这类问题通常可以采用模糊综合评判的方法。题目要求建立指标分析不同时空出租车资源的“供求匹配”程度，首先应该分析影响“供求匹配”的因素，比如：出租车数量，万人拥有量，居民消费水平等。本文从中选取三个常用并且有相关统计的指标：万人拥有量，里程利用率，满载率。然后我们考虑利用模糊综合评判模型把三个指标进行综合。首先应该通过三个指标量的理想值构建隶属函数模糊综合评判矩阵，再利用层次分析法对三个指标赋予权重，最后与模糊综合评判矩阵相乘得到最终评价结果。但是，题目中没有给出数据，需要自己搜集。数据来源可以通过查找相关资料，针对某些年份数据缺失的情况，可以通过BP神经网络预测进行填补。

对于问题二，首先分析了引起打车难的原因，分别为出租车的数量少，司机乘客信息不对称，和司机拒载。各公司的出租车补贴方案对出租车司机和顾客有一定的吸引力，原来不愿意打车的乘客数和不愿意载客的司机数会因为补贴政策有所减少，有效解决了信息不对称以及司机拒载的问题。打出租车的过程其实可以看作是一个排队，在评价排队系统的运行指标中等待时间可以作为衡量打车难易度的指标，可以利用补贴方案与乘客数和司机数之间的关系得到排队系统中乘客到达率，出租车服务率，于是可以得到等待时间和补贴额度之间的函数关系。把各个公司的补贴政策带入函数来判断补贴方案是否对缓解打车难有帮助。

对于问题三，因为问题三只考虑了补贴与等待时间的关系，并得出了补贴越多，等待时间越短的关系。但是从打车软件平台的自身利益角度考虑，不可能无节制地发放补贴，因此我们以等待时间最小为目标，打车软件平台不亏损为约束条件建立多元线性规划模型，从而计算得出最佳补贴方案。

三、模型假设

- 1、采集到的数据真实可靠。
- 2、乘客的到达情况符合泊松分布。

- 3、出租车服务时间服从负指数分布。
- 4、认为平均量可以反应整体的情况，忽略极端情况。

四、符号说明

L_z	载客总里程
L_x	行驶总里程
L	里程利用率
C_z	表示单位时间内载客车辆数
C_T	表示单位时间内通过车辆数。
C	车辆满载率
W_C	为车辆总数， W_R （万人）为人口规模。
W	万人拥有量
A	成对比较阵
w	权向量
R	模糊综合评判矩阵
B	综合评判向量
λ	乘客到达率
μ	出租车服务速率
K	城市的出租车每日载客趟次
V	出租车平均速度
S	载客里程数
C	出租车补贴给乘客金额
M	乘客出行原本应花费
A	城市总出行人数
a	乘坐出租车出行人数
S	出租车公司给司机的补贴
B	司机总数
b	没有拒载习惯的司机数目
T	可用出租车总数
W_q	单服务台排队模型中等待时间
P	打车软件新增用户数
N	为每个新增用户带来的利益

五、问题解答

5.1 问题一的模型建立与求解

针对问题一，我们从出租车的利用情况出发，提出了里程利用率，满载率，万人拥有量三项指标来衡量出租车资源的“供求匹配”程度。并运用了层次分析法，建立模糊综合评判模型得到综合指标进行评价。

5.1.1 模型建立

1、三项指标的提出

(1) 里程利用率，这一指标反应载客效率，如果比例高，说明车辆行驶总里程中空驶里程较少，对于要车的乘客来说可供租用的车辆不多，乘客的等待时间增加，说明供求关系比例紧张。里程利用率的计算公式如下，其中 L_Z (公里) 表示载客总里程， L_X (公里) 表示行驶总里程。

$$L = \frac{L_Z}{L_X} \times 100\% \quad (\text{式 5-1-1})$$

(2) 车辆满载率，是通过在客流集散较为集中的地点选取几个长期观测点，对单位时间通过道路的载有乘客的出租汽车数量占总通过出租汽车数量的比。车辆满载率高说明供求关系比例紧张。车辆满载率计算公式如下，其中 C_Z (辆) 表示单位时间内载客车辆数， C_T (辆) 表示单位时间内通过车辆数。

$$C = \frac{C_Z}{C_T} \times 100\% \quad (\text{式 5-1-2})$$

(3) 万人拥有量，用来描述一定城市规模内车辆的占用量。万人拥有量少说明供求关系比例紧张。万人拥有量计算公式如下，其中 W_C (辆) 为车辆总数， W_R (万人) 为人口规模。

$$W = \frac{W_C}{W_R} \times 100\% \quad (\text{式 5-1-3})$$

2、层次分析法确定三项指标权重

层次分析法步骤：

Step1: 将问题条理化、层次化，构造出一个有层次的结构模型。层次分为三层：目标层、准则层和方案层。

Step2: 比较同一层次元素对上一层次同一目标的影响，从而确定它们在目标中所占的比重。采用两两比较的方法，求出它们对于它们对于同一个目标的重要性的比例标度，标度等级为 1, 2, ..., 9, 1/2, 1/3, ..., 1/9。

(其中 1 到 9 标度的含义为：1 代表两个元素同等重要，3 代表前者稍重要，5 代表前者明显重要，7 代表前者强烈重要，9 代表前者极端重要。2468 为上述判断的中间值。)

Step3: 在单一准则下计算元素相对排序权重，以及判断矩阵一致性检验。

Step4: 计算方案层中各元素对于目标层的总排序权重，从而确定首选方案。

层次分析法中，设要比较的 n 个元素 $y = \{y_1, \dots, y_n\}$ 对目标 Z 的影响，确定他们在 Z 中的比重。每次取两个因素 y_i 和 y_j ，用 a_{ij} 表示 y_i 与 y_j 对 Z 的影响之比：

$$a_{ij} > 0, a_{ji} = \frac{1}{a_{ij}} \quad (i, j = 1, 2, \dots, n) \quad (\text{式 5-1-4})$$

$A = (a_{ij})$ 称为成对比较阵或者判断阵。满足上式的矩阵称为正互反阵。成对比较阵是正互反阵。 A 矩阵的最大特征值记为 λ_m ，设 $w = (w_i)$ 为最大特征值的标准化特征向量，则 $w_1 + w_2 + \dots + w_n = 1$ ，它表示了 y_1, y_2, \dots, y_n 在 Z 中的比重，称为权向量。

求解 λ_m 和 w 时可采用一种近似方法。

将 A 的各个行向量平均后,再标准化,即为 w ,这时, λ_m 的表达式如下:

$$\lambda_m = \frac{1}{n} \sum_{i=1}^n \frac{(Aw)_i}{w_i} \quad (\text{式 5-1-5})$$

λ_m 比 n 大的越多, A 不一致的程度就越严重,用 w 表示 $\{y_1, \dots, y_n\}$ 在 Z 中所占比重时的偏差越大。衡量不一致程度的数量指标叫做一致性指标,它的公式如下:

$$CI = \frac{\lambda_m - n}{n - 1} \quad (\text{式 5-1-6})$$

CI 越大, A 的不一致程度越严重。

Saaty 定义了随机性指标 CR 来衡量一致性指标 CI 。对于固定的 n , 随机地构造正互反阵 \tilde{A} , 其中 $\tilde{a}_{ij} (i < j)$ 随机地从 $1/9, \dots, 1/2, 1, 2, \dots, 9$ 当中取出一数, 由于其随机性, 可以认为这样的 \tilde{A} 是最不一致的, 用充分大的子样得到 \tilde{A} 的最大特征值的平均值 $\tilde{\lambda}_m$, 可得随机性指标公式为:

$$CR = \frac{\tilde{\lambda}_m - n}{n - 1} \quad (\text{式 5-1-7})$$

CR 部分具体值可由下表得到:

表 5-1-1 CR 值对应表

n	1	2	3	4	5	6	7	8	9
CR	0	0	0.58	0.90	1.12	1.24	1.32	1.41	1.45

当满足下式时, 认为 A 的不一致性可以接受。

$$CI \leq 0.1CR \quad (\text{式 5-1-8})$$

3、模糊综合评判模型的建立

已知因子集 $U = \{U_1, U_2, U_3, \dots, U_m\}$, 评语集 $V = \{V_1, V_2, V_3, \dots, V_n\}$, 对因子的权重分配为 U 上的模糊子集, 也就是上一部分层次分析法最后求出的权向量 $w = \{w_1, w_2, \dots, w_m\}$ 。其中 w_i 代表第 i 种因子 U_i 所对应的权重。对第 i 个因子的单因子评价为 V 上的模糊子集 R , 从而可以构成一个模糊综合评判矩阵 R 。

为了进行模糊综合评判矩阵 R 的求解, 需要确定隶属度函数, 根据所查资料我们可以得到目前里程利用率的理想区间在 63%~67%之间, 车辆满载率的理想区间在 65%~70%之间。万人拥有量的理想值是 20 辆每万人。因此我们可以建立三个指标的隶属函数如下, 其中 $f(x_l)$, $f(x_c)$, $f(x_w)$ 为里程利用率, 车辆满载率, 万人拥有量隶属函数:

$$f(x_l) = \begin{cases} \frac{x_l}{63\%} & x_l < 63\% \\ 1 & 63\% < x_l < 67\% \\ \frac{x_l}{67\%} & x_l > 67\% \end{cases} \quad (\text{式 5-1-9})$$

$$f(x_c) = \begin{cases} \frac{x_c}{65\%} & x_c < 65\% \\ 1 & 65\% < x_c < 70\% \\ \frac{x_c}{70\%} & x_c > 70\% \end{cases} \quad (\text{式 5-1-10})$$

$$f(x_w) = \frac{x}{20} \quad (\text{式 5-1-11})$$

本题的评语集 V 和一般情况有所不同，正常情况评语集都是越大越好，或者越大越差，评语集与求解出的指标值呈“单调关系”。而本题的评语集既不是越大越好，也不是越小越好，而是越接近 1 越好。小于 1 说明供求不紧张，出租车利用率低，大于 1 说明供求紧张，乘客等待时间长，因此本题指标值与 1 的偏离程度越小代表供求关系越合适。

利用上述隶属函数可求得模糊综合评判矩阵 R ，再利用层次分析法得到指标权重集 W ，则综合评判向量 B 计算公式如下：

$$B = W \times R \quad (\text{式 5-1-12})$$

3、BP 神经网络算法

因为本题要求我们分析不同时空出租车资源的“供求匹配”程度，我们决定把时定义为年份，把空间定义为发达程度不同的城市。并找寻 2010 到 2014 共 5 年，一线，二线发达，二线中等发达共 15 个典型城市的里程利用率，车辆满载率，万人拥有量。但是由于数据查找极其困难，导致某些交通上关注度较低的城市相关数据在某些年份有缺失，因此，我们采用 BP 神经网络算法利用已有的数据对缺失的统计数据预测，得到完整地数据表格。

BP 算法基本原理是利用输出后的误差来估计输出层的直接前导层的误差，再用这个误差估计更前一层的误差，如此一层一层的反传下去，就获得了所有其他各层的误差估计。算法步骤如下：

Step1: 网络初始化。给各连接权值分别赋一个区间 $(-1, 1)$ 内的随机数，设定误差函数 e ，给定计算精度值 ε 和最大学习次数 M 。

Step2: 随机选取第 k 个输入样本及对应期望输出

$$\begin{aligned} x(k) &= (x_1(k), x_2(k), \dots, x_n(k)) \\ d_0(k) &= (d_1(k), d_2(k), \dots, d_q(k)) \end{aligned} \quad (\text{式 5-1-13})$$

Step3: 计算隐含各层各神经元的输入和输出

$$hi_h(k) = \sum_{i=1}^n w_{ih} x_i(k) - b_h \quad h = 1, 2, \dots \quad (\text{式 5-1-14})$$

$$ho_h(k) = f(hi_h(k)) \quad h = 1, 2, \dots \quad (\text{式 5-1-15})$$

$$yi_o(k) = \sum_{h=1}^p w_{ho} ho_h(k) - b_o \quad o = 1, 2, \dots \quad (\text{式 5-1-16})$$

$$yo_o(k) = f(yi_o(k)) \quad o = 1, 2, \dots \quad (\text{式 5-1-17})$$

Step4: 利用网络期望输出和实际输出，计算误差函数对输出层的各神经元的偏导数

$$\frac{\partial e}{\partial w_{ho}} = \frac{\partial e}{\partial yi_o} \frac{\partial yi_o}{\partial w_{ho}}$$

$$\frac{\partial yi_o(k)}{\partial w_{ho}} = \frac{\partial (\sum_h^p w_{ho} ho_h(k) - b)}{\partial w_{ho}} = ho_h(k)$$

$$\frac{\partial e}{\partial yi_o} = \frac{\partial (\frac{1}{2} \sum_{o=1}^q (d_o(k) - yo_o(k))^2)}{\partial yi_o} = -(d_o(k) - yo_o(k)) f'(yi_o(k)) \quad (\text{式 5-1-18})$$

Step5: 利用隐含层到输出层的连接权值、输出层的 $\delta_o(k)$ 和隐含层的输出计算误差函数对隐含层各神经元的偏导数 $\delta_h(k)$ 。

$$\frac{\partial e}{\partial w_{ho}} = \frac{\partial e}{\partial yi_o} \frac{\partial yi_o}{\partial w_{ho}} = -\delta_o(k) ho_h(k)$$

$$\frac{\partial e}{\partial w_{ih}} = \frac{\partial e}{\partial hi_h(k)} \frac{\partial hi_h(k)}{\partial w_{ih}}$$

$$\frac{\partial hi_h(k)}{\partial w_{ih}} = \frac{\partial (\sum_{i=1}^n w_{ih} x_i(k) - b_h)}{\partial w_{ih}} = x_i(k)$$

$$\begin{aligned} \frac{\partial e}{\partial hi_h(k)} &= \frac{\partial (\frac{1}{2} \sum_{o=1}^q (d_o(k) - yo_o(k))^2)}{\partial ho_h(k)} \frac{\partial ho_h(k)}{\partial hi_h(k)} \\ &= -(\sum_{o=1}^q \delta_o(k) w_{ho}(k)) f'(hi_h(k)) \end{aligned} \quad (\text{式 5-1-19})$$

Step6: 利用输出层各神经元的 $\delta_o(k)$ 和隐含层各神经元的输出来修正连接权值 $w_{ho}(k)$

$$\Delta w_{ho}(k) = -\mu \frac{\partial e}{\partial w_{ho}} = \mu \delta_o(k) ho_h(k)$$

$$w_{ho}^{N+1} = w_{ho}^N + \eta \delta_o(k) ho_h(k) \quad (\text{式 5-1-20})$$

Step7: 利用隐含层各神经元的 $\delta_h(k)$ 和输入层各神经元的输入修正连接权。

$$\Delta w_{ih}(k) = -\mu \frac{\partial e}{\partial w_{ih}} = \delta_h(k) x_i(k)$$

$$w_{ih}^{N+1} = w_{ih}^N + \eta \delta_h(k) x_i(k) \quad (\text{式 5-1-21})$$

Step8: 计算全局误差

$$E = \frac{1}{2m} \sum_{k=1}^m \sum_{o=1}^q (d_o(k) - y_{o_o}(k))^2 \quad (\text{式 5-1-22})$$

Step9: 判断网络误差是否满足要求。当误差达到预设精度或学习次数大于设定的最大次数，则结束算法。否则，选取下一个学习样本及对应的期望输出，返回到第三步，进入下一轮学习。

5.1.2 模型求解

1、层次分析法权重求解

我们从出租车的使用效率出发，并且考虑三项指标非预测量的多少，我们决定把里程利用率定为最重要指标，车辆满载率作为次重要指标，万人拥有量作为第三重要的指标。因此得到判断镇 A 如下所示：

$$A = \begin{bmatrix} 1 & 4 & 2 \\ \frac{1}{4} & 1 & \frac{1}{3} \\ \frac{1}{2} & 3 & 1 \end{bmatrix}$$

将 A 的各个行向量先求和，再求平均，再标准化，即为权向量 W,计算过程如下：

$$\begin{bmatrix} 1 & 4 & 2 \\ \frac{1}{4} & 1 & \frac{1}{3} \\ \frac{1}{2} & 3 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 7.000 \\ 1.583 \\ 4.500 \end{bmatrix} \Rightarrow \begin{bmatrix} 2.333 \\ 0.528 \\ 1.500 \end{bmatrix} \Rightarrow \begin{bmatrix} 0.535 \\ 0.121 \\ 0.344 \end{bmatrix}$$

所以求得权向量为 $W=\{0.535,0.121,0.344\}$ 。下面对 W 进行一致性检验。 $n=3$ 时，CR 取 0.58，0.535,0.121,0.344 均小于 5.8，因此认为不一致性可以接受，权重合理。

下面我们把 2010 到 2015 年 15 个城市里程利用率，车辆满载率，万人拥有量的统计及预测数据（详见附录）带入隶属函数，利用 MATLAB 程序求解得到一个 3 乘 15 的模糊判断矩阵 R，与 W 相乘既得到“供求匹配”程度综合评判结果，见下表所示：

表 5-1-2 一线城市 5 年“供求匹配程度”综合评价指标

年份	2010	2011	2012	2013	2014
城市					
北京	0.1718	0.1609	0.1213	0.0927	0.0962
广州	0.0658	0.0729	0.0801	0.0990	0.1185
深圳	0.0176	0.0165	0.0273	-0.0187	-0.0057
平均值	0.0851	0.0834	0.0762	0.0577	0.0696

表 5-1-3 二线发达城市 5 年“供求匹配程度”综合评价指标

年份	2010	2011	2012	2013	2014
城市					
大连	0.0455	0.0468	0.0474	0.0625	0.0640
青岛	0.0169	0.0058	-0.0127	-0.0300	0.0037
济南	0.0464	0.0176	-0.0209	-0.0478	0.0308
厦门	0.0255	0.0160	0.0120	0.0305	0.0498
南京	-0.0188	0.0002	0.0148	0.0245	0.0242
杭州	0.0334	0.0297	0.0185	0.0155	0.0594
宁波	-0.0097	-0.0170	-0.0215	-0.0201	0.0019
平均值	0.0199	0.0142	0.0054	0.0050	0.0334

表 5-1-4 二线中等城市 5 年“供求匹配程度”综合评价指标

年份	2010	2011	2012	2013	2014
城市					
哈尔滨	0.1092	0.1376	0.1514	0.1651	0.2015
西安	0.0102	-0.0448	-0.0292	-0.0252	-0.0028
沈阳	-0.0856	-0.0672	-0.0244	0.0181	0.0211
武汉	0.0304	0.0704	0.0697	0.0797	0.1736
成都	0.0061	-0.0018	0.0150	0.0282	0.0309
平均值	0.0141	0.0188	0.0365	0.0532	0.0849

从空间上来看，一线城市，二线发达城市，二线中等城市的综合指标都比较接近 1，证明“供求匹配”程度都比较合理，综合指标都大于 1，证明都存在供求紧张的情况，而且一线城市供求紧张的情况明显大于二线发达城市，二线中等城市。

从时间上来看，一线城市，二线发达城市，在 2012,2013 年的时候综合指标变小，证明供求紧张程度得到了缓解。经过查找资料我们得知 2012 年至 2013 年是打车软件开始在大城市出现并迅速风靡的时间，因此，我们得知打车软件的出现有效地缓解了出租车资源的供求紧张局面。

5.1.3 模型检验

在空间的划分上，我们主观的认为出租车资源的“供求匹配”程度与地域的经济情况，人口情况有关，因此我们把空间划分为一线城市，二线发达城市，二线中等城市，但是这种空间划分想法是否合理还需进一步检验，因此，我们用SPSS 软件计算了3个一线城市，7个二线发达城市，5个二线中等城市5年的供求综合指标的相关系数来考察他们之间的相关性，以检验我们对空间划分的合理性。相关系数计算公式如下：

$$\rho_{xy} = \frac{Cov(X,Y)}{\sqrt{D(X)}\sqrt{D(Y)}} \quad (\text{式 5-1-23})$$

计算结果如下表所示：

表 5-1-5 一线城市指标相关系数表

	北京	广州	深圳
北京	1	0.952	0.847
广州	0.952	1	0.750
深圳	0.847	0.750	1

表 5-1-6 二线发达城市相关系数表

	大连	青岛	济南	厦门	南京	杭州	宁波
大连	1	-0.490	-0.322	0.824	0.805	0.356	0.410
青岛	-0.490	1	0.982	0.072	-0.745	0.601	0.575
济南	-0.322	0.982	1	0.258	-0.646	0.726	0.716
厦门	0.824	0.072	0.258	1	0.414	0.768	0.841
南京	0.805	-0.745	-0.646	0.414	1	0.050	0.013
杭州	0.356	0.601	0.726	0.768	0.050	1	0.975
宁波	0.410	0.575	0.716	0.841	0.013	0.975	1

表 5-1-7 二线中等城市相关系数表

	哈尔滨	西安	沈阳	武汉	成都
哈尔滨	1	-0.051	0.915	0.948	0.826
西安	-0.051	1	-0.046	0.087	0.281
沈阳	0.915	-0.046	1	0.754	0.942
武汉	0.948	0.087	0.754	1	0.696
成都	0.826	0.281	0.942	0.696	1

由相关系数计算结果可知，一线城市（北京，广州，深圳）中，北京，广州，深圳三个城市均互为强相关。二线发达城市（大连，青岛，济南，厦门，南京，杭州，宁波）中，除了宁波与南京，南京与杭州，青岛与厦门外，其余城市均为强相关。二线中等城市（哈尔滨，西安，沈阳，武汉，成都）中，除了哈尔滨与西安，西安与沈阳外，其余城市均为强相关。

5.2 问题二的模型建立与求解

针对问题二，我们把打车难转化为等待时间，并求解出对司机，乘客发放的

补贴额度与里程利用率，乘客数的定性的函数关系式，最后建立多服务台排队模型求解出等待时间与补贴额度的定性函数关系式，来分析不同补贴额度对应的等待时间，来分析是否缓解打车难。

5.2.1 模型建立

首先我们总结出造成“打车难”现象的三大原因。

一、出租车的绝对数目少。我国是世界上人口最多的国家，因此对于我们庞大的人口基数来说，虽然国家一直大力发展城市交通，在人口密集城市，交通高峰期总是很难见到空闲的出租车，造成打车难的现象。

二、出租车司机与乘客信息不对称。由于出租车司机和乘客无法知道对方所在地点，会造成司机不知道哪里有活，乘客不知道哪里有车的现象。这样难免会降低出租车的里程利用率，不仅造成出租车资源浪费，出租车公司利润不高，还会对乘客打车造成不方便。

三、出租车司机挑单拒载。部分出租车司机按照自己的意愿选单也是造成打车难现象的原因之一。2011年10月23日CCTV《朝闻天下》栏目播出《聚焦打车难》节目，记者调查显示北京打车难成为常态，一名出租司机称基本上所有司机每天都拒载两到三个客人，而另一名司机主动对记者说出租车拒载是正常现象，不拒载倒不正常了，而拒载的原因常是因为交接方向不对。

各类打车软件出现之初，各公司为吸引用户使用打车软件打车，纷纷投入巨资用作补贴。而用户使用软件打车可以很好的解决出租车司机与乘客信息不对称的问题，而且给司机的补贴可以调动司机的工作积极性，缓解司机拒载现象，理论上有助于缓解打车难问题。下面我们通过建立多服务平台排队模型来具体化这种关系。

出租车和出租车乘客是典型的服务和被服务关系。而且出租车乘客的到达符合泊松分布，因此建立排队模型是可能的。

首先，我们把排队论的参数在本题中的公式表达出来。

1、乘客到达率 λ

乘客到达率 λ 是指单位时间内乘客增加的数量，假设 λ 服从负指数分布，则：

$$f(t) = \begin{cases} \lambda \times e^{-\lambda t} & t > 0 \\ 0 & t \leq 0 \end{cases} \quad (\text{式 5-2-1})$$

根据城市的出租车每日载客趟次 K 可以得到乘客到达率公式：

$$\lambda = \frac{K}{24 \times 60 \times 60} \quad (\text{式 5-2-2})$$

2、出租车服务速率 μ

出租车服务速率 μ 是指平均每辆车单位时间服务完的批次数，假设 μ 服从负指数分布。

$$f(t) = \begin{cases} \mu \times e^{-\mu t} & t > 0 \\ 0 & t \leq 0 \end{cases} \quad (\text{式 5-2-3})$$

根据城市的出租车每日载客趟次 K,出租车平均速度 V，载客里程数 S,可以得到出租车服务率公式：

$$\mu = \frac{K \times V}{L_z} \quad (\text{式 5-2-4})$$

下面我们利用等待时间长短来反应打车难易度，导出具体补贴额度和等待时间之间的关系。

我们可知，当打车软件发给乘客的补贴越多，就会提高乘客使用打车软件的打车的意愿，因此可以使得原本不坐出租车的出行市民选择乘坐出租车，因此出租车的每日载客量就会增加。令出租车补贴给乘客 c 元，乘客出行原本应花费 M 元，城市总出行人数为 A ，乘坐出租车出行人数为 a ，则现在每日客运量 K 与补贴额度 c 的函数关系如下， k_1 是比例调整系数，使得函数关系更具有广泛适用的意义：

$$K = k_1(A - a) \frac{c}{M} + a \quad (\text{式 5-2-5})$$

由于 k_1 ， A, a ， M 都是实际情况中的一些具体常数而非变量，因此我们可以把几个常数和的形式写作一个常数，使得 K 与 c 的定性函数关系更明显，由于 K 和 λ 只是除以常数的关系，因此我们得到简化后的乘客到达率公式如下：

$$\lambda = d_1 c + d_2 \quad (\text{式 5-2-6})$$

当打车软件发给司机的补贴多，可以调动司机的积极性，让原本喜欢拒载的司机也跑起来，因此可认为可用出租车数目变多，令出租车公司补贴给司机 s 元，司机总数为 B ，没有拒载习惯的司机数目为 b ，得到可用出租车总数 T ，与补贴额度 s 的函数关系如下， k_2 是比例调整系数，使得函数关系更具有广泛适用的意义：

$$T = k_2(B - b) \frac{M + s}{M} + b \quad (\text{式 5-2-7})$$

我们认为出租车数目越多，里程利用率就越低，两者呈反比关系，比例系数为 k_3 ，可以得到里程利用率表达式为：

$$L = \frac{k_3}{T} \quad (\text{式 5-2-8})$$

则可得到：

$$L_z = L_x \times \frac{k_3}{T} \quad (\text{式 5-2-9})$$

因此我们可以得到出租车服务率，与补贴额度 s 的函数关系：

$$\mu = \frac{T \times k \times V}{L_x k_3} \quad (\text{式 5-2-10})$$

利用常数合并的方法同样可以得到出租车服务率的最终表达式为：

$$\mu = (d_3 s + d_4)(d_5 c + d_6) \quad (\text{式 5-2-11})$$

单服务台排队模型中等待时间 W_q 公式为：

$$W_q = \frac{\lambda}{\mu(\mu - \lambda)} \quad (\text{式 5-2-12})$$

因为单服务台排队模型的等待时间与 λ ， μ 的趋势关系，与多服务台一致，而多服务台排队模型等待时间关系式比较复杂，我们也只希望得到等待时间与补贴的定性函数关系，因此可以采用单服务排队模型中的等待时间进行计算，此时 W_q 不再反应真实乘客等待时间，只是反应等待时间的一个指标，它越大反应等待时间越大，带入后结果如下：

$$W_q = \frac{d_1 c + d_2}{(d_3 s + d_4)(d_5 c + d_6)((d_3 s + d_4)(d_5 c + d_6) - d_1 c + d_2)} \quad (\text{式 5-2-13})$$

5.2.2 模型求解

因为系数的大小并不影响总体的增减趋势，只是影响增减的快慢，因此我们取式 5-2-13 中的系数全为 1，得到等待时间指标的函数关系如下：

$$W_q = \frac{1}{(s+1)(c+1)s} \quad (\text{式 5-2-14})$$

用 MATLAB 得到其函数图形如下：

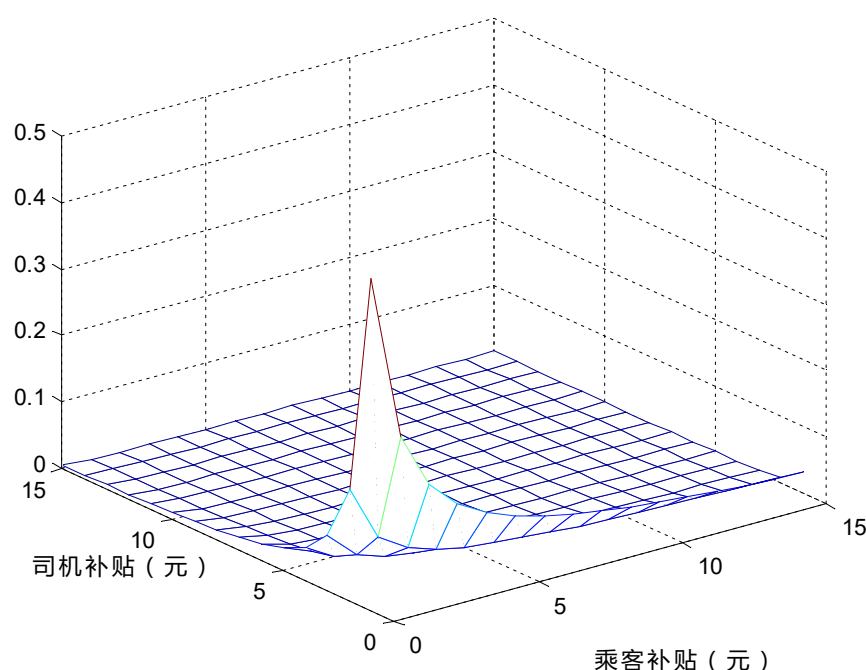


图 5-2-1 等待时间指标函数图形

从上图可以看出，补贴额度越大，等待时间指标越短，打车难得到缓解，且给司机发补贴等待时间下降的更快。

滴滴打车 2014 年 1 月 10 日，滴滴打车乘客车费立减 10 元、司机立奖 10 元的补贴方案等待时间指标为 0.0083。

快的打车 2014 年 2 月 17 日快的打车乘客返现 11 元，司机返 5-11 元的补贴方案等待时间指标为 0.0016，优于滴滴打车。

5.2.3 模型检验

我们对上述模型进行灵敏度检验，改变式 5-2-13 的系数看看 s , c , W_q 的变化情况。当 $d_1=1$, $d_2=5$, $d_3=1$, $d_4=1$, $d_5=1$, $d_6=5$ 时，得到 W_q 的函数图像如下所示：

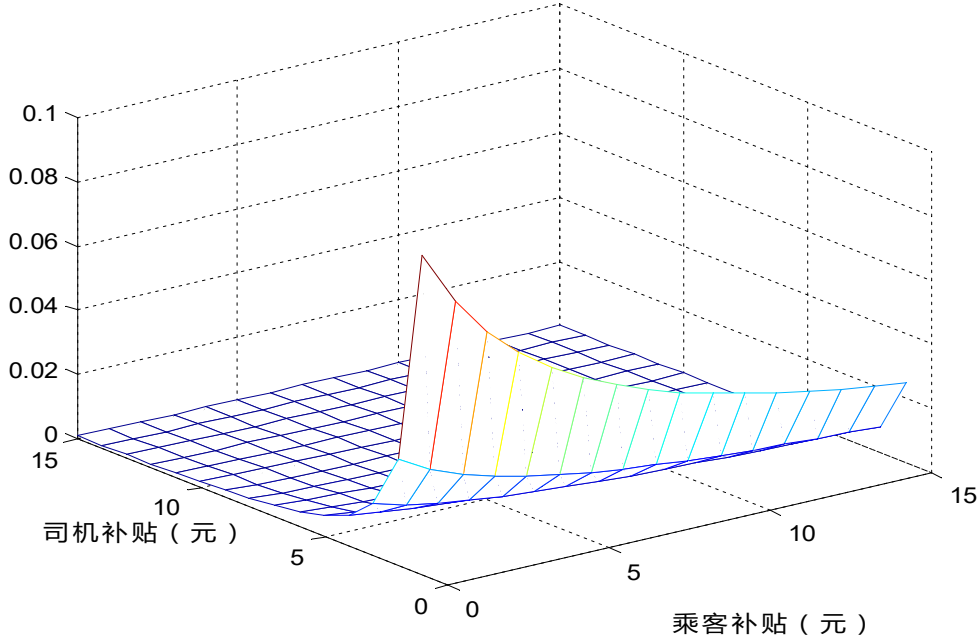


图 5-2-2 变系数后等待时间指标函数图形

从图中可以看出，乘客补贴对于函数系数的变化更为敏感。

5.3 问题三的模型建立与求解

针对问题三，我们以第二问等待时间最小为目标函数，公司投入大于收益为约束条件建立了多元线性规划模型求解出具体的乘客补贴额度，和司机补贴额度。

5.3.1 模型建立

按照问题二的模型可知，补贴越多，等待时间越短，打车难的问题就越得到了缓解。但是作为打车软件服务平台来说，不可能只为服务大众而无条件地给予补贴。因此问题三我们以等待时间最小为目标函数，以公司投入小于获益为约束条件，想要得到权衡两者的 s , c 的合理值。

打车软件公司的补贴投入为每单补贴数乘以接单数，而打车软件的盈利来源于市场资源的占有也就是新增用户数，新增用户数乘以每个用户带来的利益为总收益。

多元线性规划函数：

$$\min W_q = \frac{d_1 c + d_2}{(d_3 s + d_4)(d_5 c + d_6)((d_3 s + d_4)(d_5 c + d_6) - d_1 c + d_2)}$$

$$(s + c)K < P \times N \quad (\text{式 5-3-1})$$

其中， K 为客运量， P 为打车软件新增用户数， N 为每个新增用户带来的利

益。

5.3.2 模型求解

令 Wq 函数中，全部系数为 0。

由统计数据可知，平日里全国 $K=3520$ 万， $P=186.6$ 万， $N=200$ ，由 MATLAB 软件计算可得 $c=3.02$ ， $s=7.57$ 。

节假日某地区 $K=700$ 万， $P=25$ 万， $N=200$ ，由 MATLAB 软件计算可得 $c=1.86$ ， $s=5.27$ 。

得到平时补贴乘客 3.02 元，司机 7.57 元，节假日补贴乘客 1.86 元，补贴司机 5.27 元的补贴方案。

合理性论证：平日补贴乘客 3.02 元，司机 7.57 元时，等待时间指标为：0.0039，收益指标为 43.2。

节假日补贴乘客 1.86 元，司机 5.27 元时，等待时间指标为：0.0106，收益指标为 9。

第二问滴滴打车补贴乘客 10 元，司机 10 元，等待时间指标为：0.0083，平时收益指标为-33。节假日收益指标为-90。

可以看出我们的补贴方案，时间指标，收益指标都由于滴滴打车的补贴方案，因此是合理的。

六、模型评价

6.1 模型优点

- 1、针对部分年限数据缺失的问题，采用 BP 神经网络预测的做法进行解决，得到的数据准确性高误差小，相比于灰色预测等其他预测模型，BP 神经网络不需要检验，十分方便。
- 2、针对问题二常数众多影响计算的问题，利用系数大小不影响增减趋势的特点对函数进行常数简化，更便于计算，并且更容易体现增减趋势。
- 3、问题一采用层次分析法与模糊综合评判相结合的模型，使得评价结果更全面，可靠性高。

6.2 模型缺点

- 1、问题二由于方程系数未知，为了得到更准确的等待时间指数，应该多取几组有典型代表的系数进行计算，求取平均等待时间指数，使结果更准确
- 2、问题三考虑的约束条件过少，设计的利益方只有乘客和打车软件公司。
- 3、排队论指标用的很少，只算了等待时间一个指标。

6.3 模型改进推广

- 1、问题三设计约束条件的时候，可以再加上出租车的增长速度不能小于乘客增长速度，再考虑司机利益，出租车公司利益等更多方面
- 2、问题二排队论的应用应该把平均等待时间，平均逗留时间，排队长度等指标都给计算出来。
- 3、模型可以推广到求一个城市的最佳出租车拥有量，对于城市的规划发展具有重要意义。

参考文献

- [1] 姜启源,谢金星,叶 俊,数学模型(第四版),北京: 高等教育出版社, 2011.01
- [2] 袁长伟,吴群琪,不同目标下城市出租车最优实载率模型,长安大学学报(自然科学版) [J], 34(2): 89-93, 2014.03
- [3] 杨英俊,赵祥模,基于小波神经网络的出租车保有量预测模型[J],公路交通科技, 29(8): 136-141, 2012.08
- [4] 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],科技与产业, 15(2), 52-56, 2015.04
- [5] 周光伟,打车软件的应用对出租车行业的影响及对策分析[J],交通财会,
- [6] 卢 杰,城市出租车需求量预测与数量绘制——以大连市为例[D],东北财经大学, 1-60, 2012.11
- [7] 魏 翔,基于大数据挖掘的出租车行为检测方法研究[D],西南石油大学, 1-48, 2015.01
- [8] 时 锋,本溪市出租汽车行业服务质量综合评价[D],吉林大学, 1-57, 2010.12
- [9] 叶芃衫,基于排队论的快速公交站点延误及缩减策略研究[D],西南交通大学, 1-80, 2014.06
- [10] 冀 静, 郑春晖, 速途研究院: 2013-2014 年打车软件市场分析报告 <http://www.sootoo.com/content/480044.shtml>, 2015.09.11

附录：

附件 1:BP 神经网络数据预测

```
p=[0.6442 0.6550 0.6557] %每个城市每年万人出租车拥有量、里程利用率、满载率
t=[0.6800]
[pn,minp,maxp,tn,mint,maxt]=premnmx(p,t)
dx=[-1,1;-1,1;-1,1]
net=newff(minmax(pn),[13,1],{'tansig','purelin'},'traingdx');
net.trainParam.show=1000; %训练网络
net.trainParam.lr=0.01;
net.trainParam.epochs=10000;
net.trainParam.goal=1e-5;
[net,tr]=train(net,pn,tn); %调用TRAININGDM算法训练BP网络
pnew=p;
pnewn=tramnmx(pnew,minp,maxp);
anewn=sim(net,pnewn); %对BP网络进行仿真
anew=postmnmx(anewn,mint,maxt); %还原数据
y=anew'
```

(注：其余城市及变量参考此条程序)

附件 2:层次分析法计算权重

```
clc,clear
tmp=[1 4 2;
      1/4 1 1/3;
      1/2 3 1];
b=sum(tmp,2) %求和
c=b/3
c1=c/sum(c) %归一化
w=tmp*c1
lamdam=1/3*(w(1,:)/c1(1,:)+w(2,:)/c1(2,:)+w(3,:)/c1(3,:))
ri=[0,0,0.58,0.90,1.12,1.24,1.32,1.41,1.45];
n=3;
ci=(lamdam-n)/(n-1)/ri(n) %一致性检验
```

附件 3:模糊综合评判矩阵

```
clc,clear
f1=@(x)[(x/0.63).*(x<0.63)+1.*(x>=0.63&x<=0.67)+(x/0.67).*(x>0.67)]
f2=@(y)[(y/0.65).*(y<0.65)+1.*(y>=0.65&y<=0.70)+(y/0.7).*(y>0.70)]
f3=@(z)[z/20]
y=[0.6500 0.6750 0.6757 0.7000 0.7053
    0.7500 0.7510 0.7523 0.7624 0.7717
    0.7500 0.7302 0.7354 0.7403 0.7410
    0.5600 0.5620 0.5630 0.5851 0.5857]
```

```

0.5300 0.5643 0.5687 0.5705 0.5951
0.5600 0.5302 0.5354 0.5403 0.5410
0.5900 0.6027 0.5987 0.6005 0.6200
0.7000 0.6858 0.6942 0.7035 0.7040
0.6700 0.6615 0.6610 0.6689 0.6725
0.6200 0.6089 0.5975 0.5970 0.6000
0.5800 0.5833 0.5927 0.6010 0.6070
0.5900 0.4887 0.4968 0.5000 0.5205
0.5800 0.5930 0.6002 0.6140 0.6149
0.7800 0.7740 0.7745 0.7802 0.8157
0.6500 0.6350 0.6500 0.6535 0.6588
];
x=[0.6442 0.6550 0.6557 0.6800 0.6853
0.7255 0.7265 0.7278 0.7379 0.7472
0.6802 0.6854 0.6903 0.6910 0.6925
0.6302 0.6320 0.6330 0.6551 0.6557
0.6143 0.6187 0.6205 0.6451 0.6466
0.7053 0.7074 0.7130 0.7170 0.7190
0.7027 0.6987 0.7005 0.7200 0.7256
0.6358 0.6442 0.6535 0.6540 0.6600
0.6815 0.6810 0.6889 0.6925 0.6964
0.6889 0.6775 0.6770 0.6800 0.6844
0.8229 0.8233 0.8327 0.8410 0.8470
0.6907 0.6887 0.6968 0.7000 0.7205
0.5500 0.5530 0.5602 0.5740 0.5749
0.6875 0.6840 0.6845 0.6902 0.7257
0.6550 0.6700 0.6735 0.6788 0.6825
]
z=[48.40 46.60 40.0500 34 33.4500
19.49 20.45 21.3600 22.34 23.5662
17.50 18.24 18.9500 10.86 12.7500
35.40 35.44 35.4400 36 36.2000
35.50 30.04 26.3400 22 25.4222
30.89 28.45 20.8900 15.5 28.1667
25.15 23.00 22.4500 22.78 23.5226
16.90 20.03 22.4500 23.77 23.6700
24.00 23.45 20.5600 19.6 26.3409
18.53 19.80 20.1200 20 22.7800
24.00 28.35 28.5600 29 33.7043
23.21 24.68 30.1200 34 34.2891
24.20 24.23 25.0300 25 24.2000
16.21 23.78 23.5600 24 31.9414
21.00 21.02 22.0200 23.5 23.4600
]

```

```

y1=f2(y);
x1=f1(x);
z1=f3(z);
r0=[x1(:,1) y1(:,1) z1(:,1)]; %隶属度矩阵
r1=[x1(:,2) y1(:,2) z1(:,2)];
r2=[x1(:,3) y1(:,3) z1(:,3)];
r3=[x1(:,4) y1(:,4) z1(:,4)];
r4=[x1(:,5) y1(:,5) z1(:,5)];
a=[0.535 0.344 0.121]; %权重
r10=a*r0';
r11=a*r1';
r12=a*r2';
r13=a*r3';
r14=a*r4';
R=[r10
    r11
    r12
    r13
    r14];
T=ones(5,15);
B=R-T; %模糊综合评判矩阵

```

附件4:等待时间函数图像

```

clear;
x=0:15;
y=0:15;
[X,Y]=meshgrid(x,y)
C=ones(16);
Z=(X+C)./( (2*Y+C).*( (X+C).^2).*2.*Y);
mesh(X,Y,Z)
x1=xlabel('乘客补贴价格(元)')
x2=ylabel('司机补贴价格(元)')
set(x1,'Rotation',20);
set(x2,'Rotation',-25);

```

附件5:灵敏度分析

```

clear;
x=0:15;
y=0:15;
[X,Y]=meshgrid(x,y)
C=ones(16);
C1=C*5;

```

```

C2=C*5;
C3=C*5;
k1=C;
k2=C;
k3=C;
Z=(k1.*X+C1)./( (k2.*Y+C2).*( (k3.*X+C3).^2).*k2.*Y);
mesh(X,Y,Z)
x1=xlabel('乘客补贴价格(元)')
x2=ylabel('司机补贴价格(元)')
set(x1,'Rotation',20);
set(x2,'Rotation',-25);

```

(注：通过改变C1,C2,C3,K1,k2,k3来进行灵敏度的分析检验)

附件6:节假日线性规划

```

model:
min=1/((s+1)*(c+1)*s);
    700*(s+c)<=200*25;
    c>=0;
    s>=0;
    c<=15;
    S<=15;
end

```

附件7:线性规划

```

model:
min=1/((s+1)*(c+1)*s);
    3520*(s+c)<=200*186.6;
    c>=0;
    s>=0;
    c<=15;
    S<=15;
end

```

附件 8:问题一数据表格

附表 8-1 各城市万人拥有量

城市	2010	2011	2012	2013	2014
年份					
大连	35.40	35.44	35.4400	36.0000	36.2000
沈阳	23.21	24.68	30.1200	34.0000	34.2891
北京	48.40	46.60	40.0500	34.0000	33.4500

广州	19.49	20.45	21.3600	22.3400	23.5662
哈尔滨	24.00	28.35	28.5600	29.0000	33.7043
西安	24.20	24.23	25.0300	25.0000	24.2000
武汉	16.21	23.78	23.5600	24.0000	31.9414
南京	16.90	20.03	22.4500	23.7700	23.6700
成都	21.00	21.02	22.0200	23.5000	23.4600
厦门	25.15	23.00	22.4500	22.7800	23.5226
青岛	35.50	30.04	26.3400	22.0000	25.4222
宁波	18.53	19.80	20.1200	20.0000	22.7800
杭州	24.00	23.45	20.5600	19.6000	26.3409
济南	30.89	28.45	20.8900	15.5000	28.1667
深圳	17.50	18.24	18.9500	10.8600	12.7500

附表 8-2 各城市里程利用率

城市 年份	2010	2011	2012	2013	2014
大连	0.6302	0.6320	0.6330	0.6551	0.6557
沈阳	0.5500	0.5530	0.5602	0.5740	0.5749
北京	0.6442	0.6550	0.6557	0.6800	0.6853
广州	0.7255	0.7265	0.7278	0.7379	0.7472
哈尔滨	0.8229	0.8233	0.8327	0.8410	0.8470
西安	0.6907	0.6887	0.6968	0.7000	0.7205
武汉	0.6875	0.6840	0.6845	0.6902	0.7257
南京	0.6358	0.6442	0.6535	0.6540	0.6600
成都	0.6550	0.6700	0.6735	0.6788	0.6825
厦门	0.7027	0.6987	0.7005	0.7200	0.7256
青岛	0.6143	0.6187	0.6205	0.6451	0.6466
宁波	0.6889	0.6775	0.6770	0.6800	0.6844
杭州	0.6815	0.6810	0.6889	0.6925	0.6964
济南	0.7053	0.7074	0.7130	0.7170	0.7190
深圳	0.6802	0.6854	0.6903	0.6910	0.6925

附表 8-3 各城市满载率

城市 年份	2010	2011	2012	2013	2014
大连	0.5600	0.5620	0.5630	0.5851	0.5857
沈阳	0.5800	0.5930	0.6002	0.6140	0.6149
北京	0.6500	0.6750	0.6757	0.7000	0.7053

广州	0.7500	0.7510	0.7523	0.7624	0.7717
哈尔滨	0.5800	0.5833	0.5927	0.6010	0.6070
西安	0.5900	0.4887	0.4968	0.5000	0.5205
武汉	0.7800	0.7740	0.7745	0.7802	0.8157
南京	0.7000	0.6858	0.6942	0.7035	0.7040
成都	0.6500	0.6350	0.6500	0.6535	0.6588
厦门	0.5900	0.6027	0.5987	0.6005	0.6200
青岛	0.5300	0.5643	0.5687	0.5705	0.5951
宁波	0.6200	0.6089	0.5975	0.5970	0.6000
杭州	0.6700	0.6615	0.6610	0.6689	0.6725
济南	0.5600	0.5302	0.5354	0.5403	0.5410