

“互联网+”时代的出租车资源配置模型

摘要

本文通过建立出租车资源“供求匹配”度指标评定出租车资源匹配效率，及分析补贴方案对打车难指数的影响，而后在现有补贴方案基础上设立增加推动城市出租车整体分布平稳的补贴方案和降低打车难指数的等车时间补贴方案，同时论证其合理性。

针对问题一，建立出租车资源“供求匹配”度指标评价模型。首先，建立定义为出租车需求量和供给量差值绝对值的最大值同出租车需求量和供给量总和比值的不同时空出租车资源“供求匹配”度指标及评价标准。选择具有代表性的西安、上海两城市进行具体求解分析，因各城市出租车具体分布情况不同，需要先运用参数均值聚类法对城市区域进行划分；然后对所查找的日出租车分布及需求数据进行多项式拟合，但由于数据量过少，需要先运用三次样条插值法及线性插值法进行插值，而后添入由出租车分布数据确定的各分区的函数参数，得到各分区具体函数；最后将各分区出租车资源“供求匹配”度加权平均得到该城市整体出租车资源“供求匹配”度，同时将所得数据与比对表比对得到该城市出租车资源“供求匹配”度评价。其中西安及上海的出租车资源“供求匹配”度评价如下：

西安及上海的出租车资源“供求匹配”度具体值及评价结果表

城市	出租车资源“供求匹配”度具体值	评价结果
西安	0.76916	良
上海	0.96638	差

针对问题二，建立打车难指标模型。首先分析“滴滴出行”和“快的打车”两公司的补贴方案，知补贴可分为对乘客的补贴、对出租车司机的补贴和其它补贴特点，然后探寻出其对出租车供给、需求的刺激函数。而刺激供、需函数与乘客等待时间和出租车空载率具有函数关系，同时乘客等待时间和出租车空载率均为出租车的打车难指标的评判因子，将两因子与打车难指标建立函数关系，最终得到补贴方案与打车难指标的函数及评价标准。通过查询相关数据，确定函数系数，分析西安、上海两城市补贴前后打车难指标变化。最终得出补贴方案对缓解打车难有一定效果，但仍然存在补贴方案无效的情况，及其具体是西安市补贴方案每单给乘客和司机分别补贴 11 元和 6 元。

针对问题三，根据已有的出租车补贴方案的弊端，我们提出了新的打车软件服务平台的设计思路及补贴方案，分别建立以降低空载率为目的的出租车均衡分配补贴模型、以提高匹配效率为目的的最优供求匹配动态任务分派模型和以降低打车难指数为目的的等待时长补贴模型，证明该补贴方案的合理性。最后对西安地区进行模型模拟，假定存在 100 辆车和 100 位乘客，并将其地理位置随机分配，通过 Matlab 软件求解线性规划模型，得到最终匹配矩阵，估算出等待时间，并给出相应补贴金额。

关键字：出租车软件服务平台 匹配程度 打车难指数 出租车软件补贴

一、问题重述

改革开放以来，我国的国民经济快速发展，人民的生活水平也不断提高，我国是人口大国，始终坚持着“以人为本”建设小康社会，时刻关注着民生问题。如今，市民出行需求在数量、质量上都在不断提升，许多市民因为公交车里乘客的拥挤、行驶的速度缓慢以及不能直接到达目的地，而倾向于将出租车作为出行的重要选择，但是“打车难”的现象也日益严重，并逐渐成为人们关注的一个社会热点问题。

随着科技的发展，“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，如“滴滴打车”、“快的打车”、“优步打车”等等，乘客将软件下载至手机，方便快捷地实现了自己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，不仅节约了时间同时节省了资金。

我们将打车问题具体化，通过搜集相关数据，运用数学思想，建立数学模型来研究出租车资源配置中的以下问题：

- (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，将设计什么样的补贴方案，并论证其合理性。

二、模型假设与符号说明

2.1 模型假设

- (1) 出租车每次只能接一单生意，不能同时接两单以上的生意；
- (2) 出租车的供给量及需求量仅受到打车软件补贴方案的影响，不考虑天气、政府政策等其他影响因素；
- (3) 打车软件不存在软件漏洞，或出现使用过程中信息获取失败的情况，影响乘客与司机间信息的及时更新；
- (4) 乘客在使用软件打车后在等待出租车前来的过程中，不会放弃已选择的出租车。

2.2 符号说明

符号	符号的含义
P	出租车“供求匹配”度
$Q(t, x, y)$	在 t 时刻点 (x, y) 处出租车的需求量
$D(t, x, y)$	在 t 时刻点 (x, y) 处出租车的供给量

θ	多项式拟合程度的比例系数
β_i	第 <i>i</i> 个分区的出租车需求量比例系数
$Q(t, \beta_i)$	第 <i>i</i> 个分区打车需求函数
α_i	第 <i>i</i> 个分区的出租车供给量比例系数
$D(t, \alpha_i)$	第 <i>i</i> 个分区出租车供给函数
R	“打车难”指标
$m_1(t)$ 、 $m_2(t)$ 、 $m_3(t)$	乘客补贴、司机补贴、其他非现金补贴
$Q(t)$	补贴方案实施时的需求函数
$D(t)$	补贴方案实施时的供给函数
T	乘客等车时间
δ	出租车的空载率

三、问题一的分析、模型建立与求解

3.1 问题分析

问题一要求建立合理的指标，并分析不同时间、空间出租车资源的“供求匹配”程度。首先，定义出租车资源“供求匹配”是指在某一确定时空中，出租车的需求量与出租车的供给量是相等的。然后将出租车需求量和供给量差值绝对值的最大值与出租车需求量与供给量和的比值定为出租车资源“供求匹配”评价指标。可以具体求得特定时间段、具体区域的评价指标值和评价分类等级。

对于具体的城市计算，首先需要对空间进行划分，运用 K-Means 算法建立聚类分析模型，根据某时刻出租车分布的位置及相应的数量，作出立体等量线图，选取特定聚类中心点，以欧氏距离作为相似度将距离中心点最近的其它点聚类，重复此过程直至新的聚类中心与原聚类中心相等或距离函数开始收敛，确定对城市的划分。然后，建立出租车资源供求函数模型，在已有的数据基础上结合三次样条插值法及线性插值法进行数据的增添，由此利用 Matlab 软件编程， $\theta \geq 90\%$ 为衡量标准对数据进行多项式拟合，再由另一天的数据进行检验，得到一天中出租车的供给量及需求量关于时间的函数。根据各个分区内供给量及需求量占城区总供给量及需求量的比例，我们可以得到每个分区的供求函数。最后，以供求不匹配度 P_i 作为第*i*个分区评判指标，并建立城市评判指标。然后，我们以西安及上海为代表城市，具体应用所建立的城市评价指标模型，对出租车资源的“供求匹配”程度进行分析。

3.2 模型建立与求解

3.2.1 模型建立

1. 建立指标评价模型

首先，定义出租车资源“供求匹配”是指在某一确定时空中，出租车的需求

量与出租车的供给量是相等的。然后，根据定义建立“供求匹配”评价模型。起初，在某一特定时间段中，将出租车需求量和供给量差值绝对值的最大值定义为“供求匹配”评价指标，但不同城市的出租车数量具有量级上的差异。因此，在原先的基础上，引进出租车需求量与供给量的指标，具体的，用出租车需求量和供给量差值绝对值的最大值与出租车需求量与供给量和作比，并将其重新定义为“供求匹配”评价指标。指标表示方式如下：

$$P = \max_{t \in [t_1, t_2]} \frac{|Q(t, x, y) - D(t, x, y)|}{Q(t, x, y) + D(t, x, y)}$$

其中 P 为出租车“供求匹配”度， $Q(t, x, y)$ 为在 t 时刻点 (x, y) 处出租车的需求量， $D(t, x, y)$ 为在 t 时刻点 (x, y) 处出租车的供给量， t_1 、 t_2 为所求出租车“供求匹配”评价度的时间起止点。点属 (x, y) 于评价指标所考虑区域范围。通过本评价标准即可求得所考虑区域在特定时间范围的出租车“供求匹配”度。且当 P 值越接近 1，则说明该地区在指定时间段的匹配度越差，当 P 值越接近 0，则说明该地区在指定时间段的匹配度越好。

而对于 P 具体值的匹配程度评价，我们是根据所查阅文献资料以及相关数据，以及结合 5 种不同级别城市具体事例计算，建立如下的确定指标评价评定标准表：

表 1：“供求匹配”程度对照表

P 值	匹配程度
$P > 0.9$	差
$0.9 > P > 0.8$	较差
$0.8 > P > 0.7$	良
$0.7 > P > 0.6$	较优
$0 < P < 0.6$	优

2. 指定城市的具体算法

由于城市的行政划分不同，无法笼统的得出城市确定具体坐标时出租车需求和供给的函数，因此需要先将城市进行分区，然后才可以得到各个分区的供给和需求函数。

(1) 建立聚类分区模型

根据打车高峰期出租车分布的位置及相应的数量，作出三维立体等量线图。我们以标志性的建筑或路段作为该附近出租车所处位置名称进行统计，运用 K-Means 算法迭代更新的思想进行聚类。聚类过程如下：在所有点中随机选取了 k 个点作为初始聚类中心，该位置坐标为 $M_i(a_i, b_i)$ ($i = 1, 2, \Lambda, k$)，剩余其它 k' 点的位置坐标 $N_j(a'_j, b'_j)$ ($j = 1, 2, \Lambda, k'$)，通过计算其它点与这些聚类中心点的欧氏距离：

$$d_{ij} = \sqrt{(a_i - a'_j)^2 + (b_i - b'_j)^2}$$

作为相似性的评价指标，即认为两个点的距离越近，相似度越大，可分为同一个区。将其它点分配给距离最近的聚类中心点所在的聚类。然后对划分好后的聚类重新进行上述步骤，这一过程不断重复直到新的聚类中心与原聚类中心相等或距

离函数开始收敛，分区结束。

对所有点进行聚类分区之后，很难看出每个分区的边界，因此需要对边界进行运算，绘制出区域的边界过程如下：首先建立平面直角坐标系，以坐标系的原点为中心点均分为 m 等份区域，每个区域的角度为 $360/m$ 。

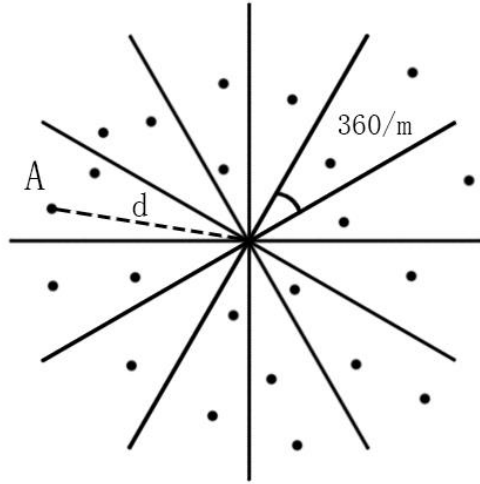


图 1: m 等份区域图

之后，将 k 个聚类中心点放入该坐标系，使得中心点与坐标原点重合。我们已知每个点的坐标，通过计算其它点与中心点所形成的三角函数关系，即可得到每个点与中心点所形成的夹角，进而将这些点归入之前所划分的 m 等份区域。依次计算每个区域里每个点距离中心点的距离，记录各区域内距离中心点最远的点，如图中 A 点。最后，将每个区域内的最远点相连接，即可得到点集的相应边界，同时确定了以 k 个聚类中心点将该城市完整划分了 k 个分区。

(2) 建立出租车资源供求函数模型

(i) 数据的预处理

我们发现获取的数据时间间隔不一并且数量偏少，而我们需要较多的数据，因此首先运用三次样条插值法若不适用则采用线性插值法，在已有的数据基础上进行数据的增添。三次样条插值法避免了因高次插值具有不收敛稳定性差的缺点，同时样条富有弹性的特性弥补了低次线性插值的光滑性较差的不足。多个样条相互弯曲连接后沿其边缘画出的曲线就是三次样条曲线。求解三次样条插值函数过程如下：

若函数 $F(t)$ 在节点 t_1, t_2, Λ, t_n 处的函数值为：

$$F(t_i) = x_i \quad i = 1, 2, \Lambda, n$$

并且关于这个节点集的三次样条函数 $G(t)$ 满足：

$$(i) \text{ 插值条件: } G(t_i) = x_i \quad i = 1, 2, \Lambda, n$$

$$(ii) \text{ 连续性条件: } \lim_{t \rightarrow t_i} G(t) = G(t_i) = x_i \quad i = 1, 2, \Lambda, n-1$$

(iii) 一阶导数连续条件: $\lim_{t \rightarrow t_i} G'(t) = G'(t_i) \quad i = 1, 2, \Lambda, n-1$

(iv) 二阶导数连续条件: $\lim_{t \rightarrow t_i} G''(t) = G''(t_i) \quad i = 1, 2, \Lambda, n-1$

则求解出三次样条函数 $G(t)$ 即为三次样条插值函数，当曲线中有趋于直线的情形时，拟合效果不好，也不能解决具有垂直切线的问题，二阶导数不连续时，将产生较大的误差，因此当不适用三次样条插值法时，我们使用线性插值法。线性插值是通过连接两个已知点的直线来确定在这两个点之间一个未知点的取值方法。过程如下：

首先必须满足插值函数 $G(t)$ 与 $F(t)$ 在节点 t_1, t_2, Λ, t_n 处的函数值相同，即为：

$$G(t_i) = F(t_i) = x_i \quad i = 1, 2, \Lambda, n$$

假设我们已知两点坐标 (t_1, x_1) 与 (t_2, x_2) ，要得到 $[t_1, t_2]$ 区间内由两点确定的直线上某

点 (t, x) 。根据 $\frac{x - x_1}{x_2 - x_1} = \frac{t - t_1}{t_2 - t_1} = \lambda$ ，即可得到 $x = x_1 + \lambda(x_2 - x_1)$ ，以此增添两点间的

数据，为后续的数据拟合奠定基础，但无法确定 $G(t)$ 具体的函数形式。

我们可得到插值函数 $G(t)$ 代表的样本数据曲线，根据图像的走势，利用多项式回归，通过 Matlab 软件编程对图线进行拟合，可得到拟合函数 $f(t)$ 。运用最小二乘思想，设拟合程度的比例系数为 θ ，则

$$\theta = 1 - \frac{RSS}{TSS}$$

其中残差平方和：

$$\begin{aligned} RSS &= \int_{t_1}^{t_n} e_i^2 dt \\ &= \int_{t_1}^{t_n} [f(t) - G(t)]^2 dt \end{aligned}$$

总平方和：

$$TSS = \int_{t_1}^{t_n} [G(t) - \bar{G}(t)]^2 dt$$

用从低至高的不同阶数多项式依次进行拟合，最初我们选取 $\theta \geq 95\%$ 作为判定拟合结果的标准，但考虑到本文研究出租车的空车在运量及打车的需求量波动频繁且幅度差距较大，做出来的拟合曲线在对未来预测时会出现较大的偏差，因此我们选取 $\theta \geq 90\%$ 为衡量标准，即可认为拟合程度达到要求。

我们需要查找某城市某一天出租车的供给量以及乘客打车需求量的数据，运用三次样条插值法，使数据得以充足，再由上述拟合数据的方法得到了出租车在该天每时每刻的供给量 $D_1(t)$ 及需求量 $Q_1(t)$ 关于时间的函数。再根据另外某一天的

数据进行检验，同样由最小二乘思想，以 θ 作为阈值，若 $\theta \geq 90\%$ ，则代表拟合结果是可用的。

(ii) 出租车需求函数

由于考虑到该城市不同分区内的乘客对出租车需求量的情况不同，因此第 i 个分区打车需求的函数表达式为： $Q(t, \beta_i) = \beta_i Q_1(t)$ ①

$$\beta_i = \frac{\text{第}i\text{区需求量}}{\text{城市总需求量}} \quad ②$$

β_i 表示第 i 个分区的出租车需求量比例系数。

(iii) 出租车供给函数

该城市不同分区内的出租车供给量即空车在运量的情况也不同，因此第 i 个分区出租车供给的函数表达式为： $D(t, \alpha_i) = \alpha_i D_1(t)$ ③

$$\alpha_i = \frac{\text{第}i\text{区供给量}}{\text{城市总供给量}} \quad ④$$

α_i 表示第 i 个分区的出租车供给量比例系数。

(3) 城市具体“匹配程度”指标求解

根据以上的所确定的函数，可以确定出第 i 个分区中的出租车“供求匹配”度，然后结合该城市所有分区的评判指标，加权平均得到该城市的“供求匹配”程度，其表达式如下。

$$P = \sum_{i=1}^k P_i \alpha_i \quad (i = 1, 2, \Lambda, k)$$

其中 P_i 的具体表达式为：

$$P_i = \max_{t \in [t_1, t_2]} \frac{|Q_i(t, p(x, y)) - D_i(t, p(x, y))|}{Q_i(t, p(x, y)) + D_i(t, p(x, y))} \quad (i = 1, 2, \Lambda, k)$$

且点 $p(x, y)$ 属于第 i 个分区。

3.2.2 模型求解

根据上述建立的模型，我们以分析西安和上海出租车资源为例，因为这两个城市可以称为打车软件的起源地，并且在软件投入市场后，使用的人数日益增加，在全国的消费者中占有较大的比例。

1. 搜集数据及预处理

首先，需要搜集城市出租车的分布及打车需求量的数据，但这类数据属于企业的内部资料，我们尝试了多种途径都无法获取，最终找到了“滴滴快的智能出行平台”（网址为 v.kuaidadi.com）。点开该网页后，选择 web 开发者中的浏览器控制台，再根据网页数据分析，标志成 XHR 消息其中含有 ZIP 数据，将我们需要的 ZIP 压缩数据保存即可，由此分步得到西安及上海出租车分布及打车需求量的数

据。

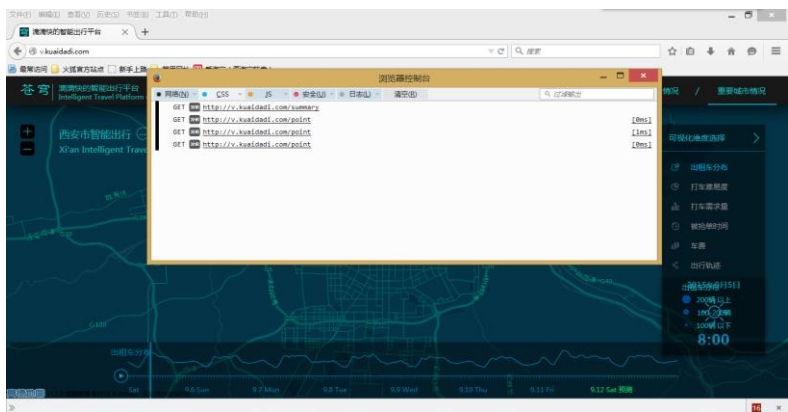


图 2：搜集数据图

2. 西安及上海的指标求解

(1) 西安市

(i) 聚类分区

我们搜集到 2015 年 9 月 10 日这一天西安出租车的分布及打车需求量的数据，在实际情况下，西安打车的早高峰期大致在 10 点左右，一般的上下班时间大多数市民选择公共交通，因此我们根据在 10 点出租车分布的位置及相应的数量，作出三维立体等量线图。

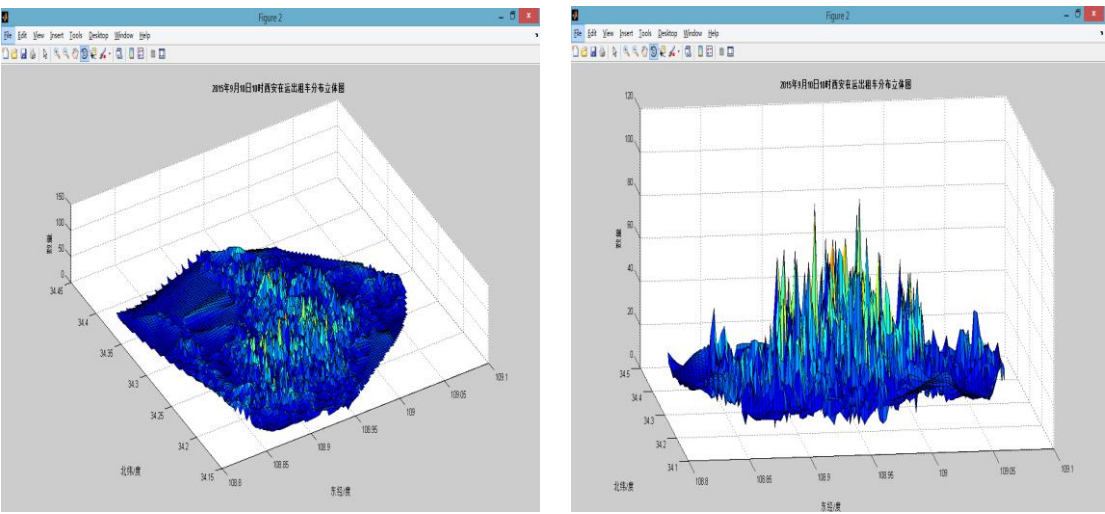


图 3：2015 年 9 月 10 日 10 点西安出租车在运分布不同视角立体等量线图

在图中有 11 个出租车在运数量峰值点，利用建立的聚类分区模型，随机选取 11 个点为初始聚类中心，其它点与该中心点间的欧氏距离最小为目标函数，每个点以其经纬度作为该点的位置坐标，将其它点分配至距离最近的中心点，在对所有点进行聚类之后，重新进行上述步骤，这一过程不断重复直到新的聚类中心与原聚类中心相等或距离函数开始收敛，分区结束。再绘制每个分区的边界，所建立的坐标系以原点为中心点均分为 8 等份区域，每个区域的角度为 $360/8$ 度，将中心点与坐标原点重合，将其它点归入这 8 个区域中，将各区域内距离中心点最远的点相连接，即得到相应边界，最终将西安划分为 11 个分区。分区结果如下图所示：

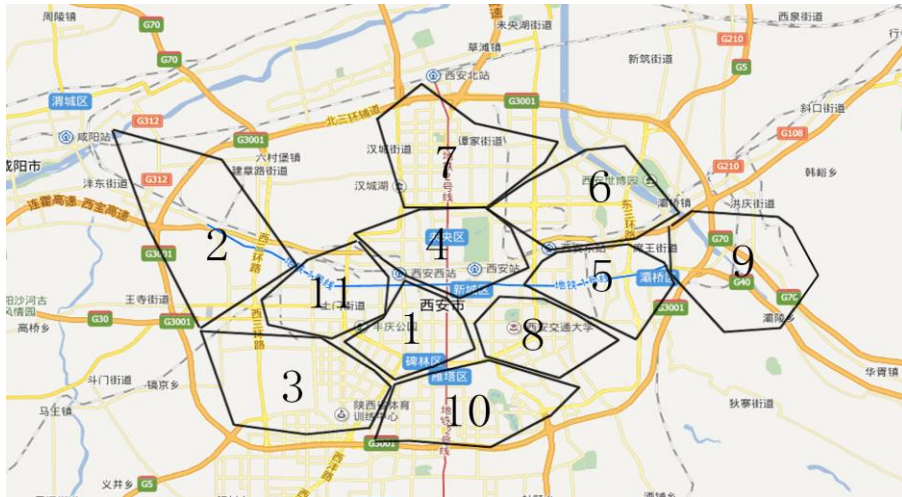


图 4：西安分区图

(ii) 拟合供求函数

由建立的出租车资源供求函数模型，运用三次样条插值法及线性插值法，在已获取的数据基础上增添更多用于拟合函数的数据，并运用 Matlab 分别拟合得到出租车在运量及需求量的函数。

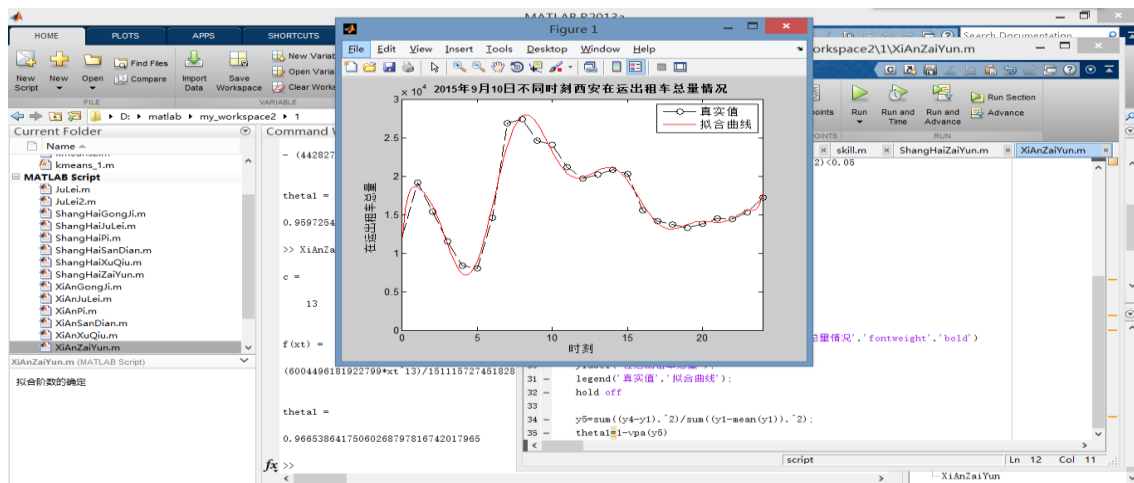


图 5：2015 年 9 月 10 日西安出租车在运量拟合函数

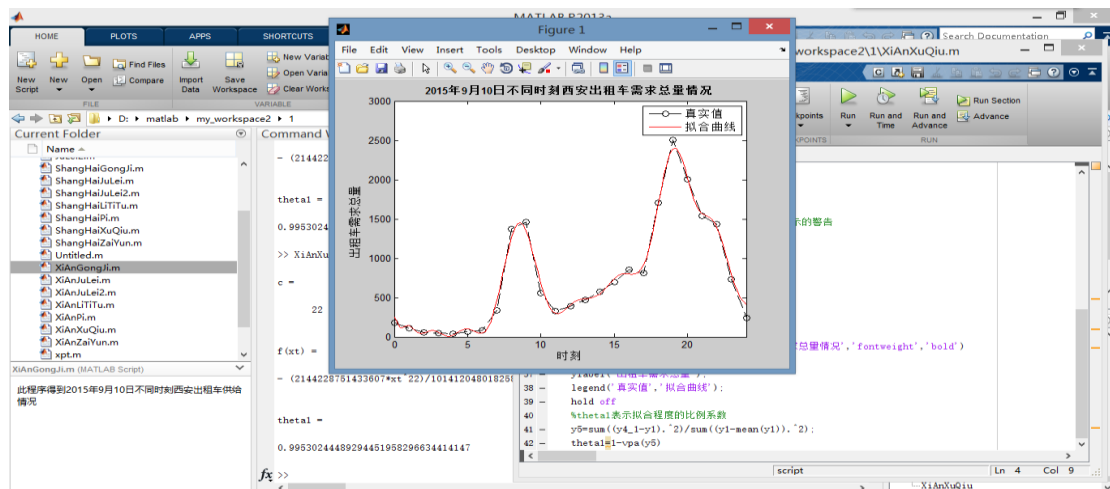


图 6：2015 年 9 月 10 日西安出租车需求拟合函数

但是为了之后得到供求不匹配度，我们需要的数据为出租车的供给量，即空

车在运量，但无法直接获取供给量的数据。我们通过数据分析，得到乘客的平均等待时间为 15 分钟，因此我们运用出租车在运量减去前 15 分钟的乘客需求量，得到当前的出租车供给量，再由多项式拟合得到如下结果：

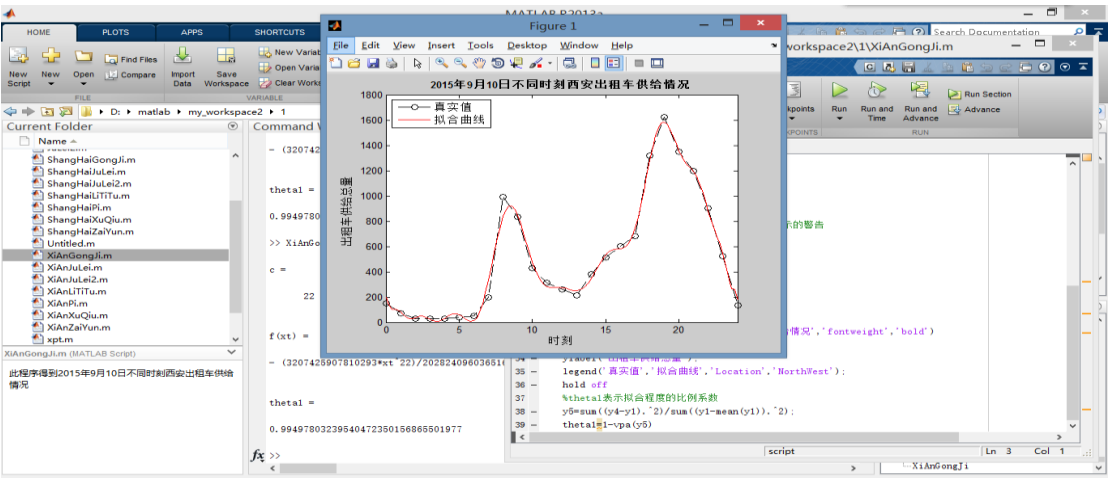


图 7：2015 年 9 月 10 日西安出租车供给拟合函数

经计算，出租车供给量与需求量的函数拟合程度 θ 值分别为 95.9725%、99.6155%，均大于 90%，可认为拟合程度达到要求，拟合函数可以使用。

表 2：西安出租车供给及需求函数

函数	西安出租车
供给函数	$D1(xt) = -(3207425907810293 * xt^{22}) / 20282409603651670423947251286016 + (6536126813547615 * xt^{21}) / 158456325028528675187087900672 - (3096880563601467 * xt^{20}) / 618970019642690137449562112 + (905815685028217 * xt^{19}) / 2417851639229258349412352 - (5859371070086633 * xt^{18}) / 302231454903657293676544 + (3473515506559737 * xt^{17}) / 4722366482869645213696 - (6251562180418525 * xt^{16}) / 295147905179352825856 + (8722310737356789 * xt^{15}) / 18446744073709551616 - (1194176735816519 * xt^{14}) / 144115188075855872 + (4133761437988275 * xt^{13}) / 3602879701896396 - (5662115724679661 * xt^{12}) / 4503599627370496 + (6121169706031259 * xt^{11}) / 56294995342131 - (2593587266555487 * xt^{10}) / 35184372088832 + (3406693994801575 * xt^9) / 8796093022208 - (6821327461862607 * xt^8) / 4398046511104 - (5084704079783413 * xt^7) / 1099511627776 - (2730345961922407 * xt^6) / 274877906944 + (8053520395732303 * xt^5) / 549755813888 - (3783899749972077 * xt^4) / 274877906944 + (7914850010495531 * xt^3) / 1099511627776 - (6328194098778205 * xt^2) / 4398046511104 - (1965205281936441 * xt) / 8796093022208 + 7034099226953473 / 35184372088832$
需求函数	$Q2(xt) = -(2144228751433607 * xt^{22}) / 10141204801825835211973625643008 + (278576970496001 * xt^{21}) / 4951760157141521099596496896 - (8621870802798059 * xt^{20}) / 1237940039285380274899124224 + (5151028800909423 * xt^{19}) / 9671406556917033397649408 - (1064186004026791 * xt^{18}) / 37778931862957161709568 + (2581215075138221 * xt^{17}) / 2361183241434822606848 - (2378330231456161 * xt^{16}) / 73786976294838206464 + (6803244295370179 * xt^{15}) / 9223372036854775808 - (7649488040276821 * xt^{14}) / 576460752303423488 + (850953155439203 * xt^{13}) / 4503599627370496 - (2402462893860275 * xt^{12}) / 1125899906842624 + (1342019021269307 * xt^{11}) / 70368744177664 - (4717560645809999 * xt^{10}) / 35184372088832 + (6455761571046379 * xt^9) / 8796093022208 - (6775104359312433 * xt^8) / 2199023255552 + (5338691068392695 * xt^7) / 549755813888 - (6134368797890861 * xt^6) / 274877906944 + (4929515882794193 * xt^5) / 137438953472 - (2604095402295297 * xt^4) / 68719476736 + (6542430745099143 * xt^3) / 274877906944 - (4009537120589005 * xt^2) / 549755813888 + (4267604824163107 * xt) / 8796093022208 + 8698408045866383 / 35184372088832$

由①②式，我们可得到第*i*个分区出租车的需求函数 $Q(t, \beta_i)$ ；由 ③④式，得到第*i*个分区出租车的供给函数 $D(t, \alpha_i)$ ， $i=1,2,\Lambda, 11$ 。计算得到第*i*个分区的出租车供给比例系数 α_i 及需求比例系数 β_i 结果如下：

表 3：西安分区供求比例系数结果表

西安分区	α	β
1	0.076	0.083
2	0.082	0.077
3	0.149	0.149
4	0.142	0.135
5	0.091	0.091
6	0.101	0.125
7	0.110	0.124
8	0.101	0.112
9	0.090	0.085
10	0.029	0.031
11	0.030	0.012

(iii) 评价指标

根据以上所求到的函数指标，以及西安城市分区确定，先确定出各 11 个分区的匹配程度数据，其数据分别为：

表 4：西安市 11 个分区的“供求匹配”程度值

区号	1	2	3	4	5	6	7	8	9	10	11
P 值	0.786542	0.758606	0.770523	0.758626	0.771398	0.810957	0.792511	0.791304	0.759593	0.789136	0.654981

再结合加权平均的方法，利用函数：

$$P = \sum_{i=1}^k P_i \alpha_i \quad (i=1,2,\Lambda, k)$$

最终得到西安市的“供求匹配”程度为 $P=0.76916$ 。再与“供求匹配”程度对照表比对，可知，西安的“供求匹配”程度等级为一般。

(2) 上海市

(i) 聚类分区

我们搜集到 2015 年 9 月 10 日这一天上海出租车的分布及打车需求量的数据，考虑到上海的经济水平及发展速度高于西安，且出行的人流量较大，上海打车的早高峰期大致在 9 点左右，因此我们根据 9 点出租车分布的位置及相应的数量，作出三维立体等量线图。

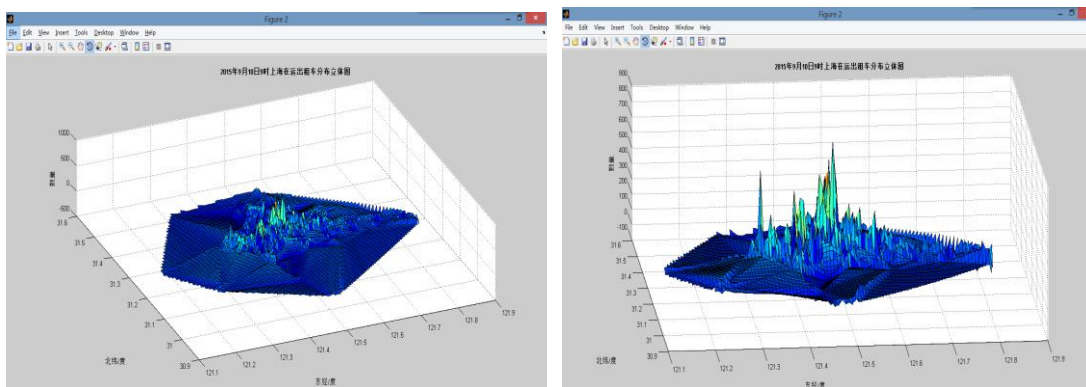


图 8: 2015 年 9 月 10 日 10 点上海出租车在运分布不同视角立体等量线图

在图中有 15 个出租车在运数量峰值点, 利用建立的聚类分区模型, 以最高点为聚类分区的中心点, 我们得到 15 个聚类中心点, 再以其它点与中心点间的欧氏距离最小为目标函数, 每个点以其经纬度作为该点的位置坐标, 将其它点聚类至距离最近的中心点。在对所有点进行聚类之后, 再绘制每个分区的边界, 所建立的坐标系以原点为中心点均分为 8 等份区域, 每个区域的角度为 $360/8$ 度, 将中心点与坐标原点重合, 将其它点归入这 8 个区域中, 将各区域内距离中心点最远的点相连接, 即得到相应边界, 最终将上海划分为 15 个分区, 如下图所示:

图 9: 上海分区图

(ii) 拟合供求函数

由建立的出租车资源供求函数模型，我们首先增添数据，对数据处理得到出租车的供给量数据，并运用 **Matlab** 分别拟合得到出租车供给及需求函数。

图 10: 2015 年 9 月 10 日上海出租车供给拟合函数

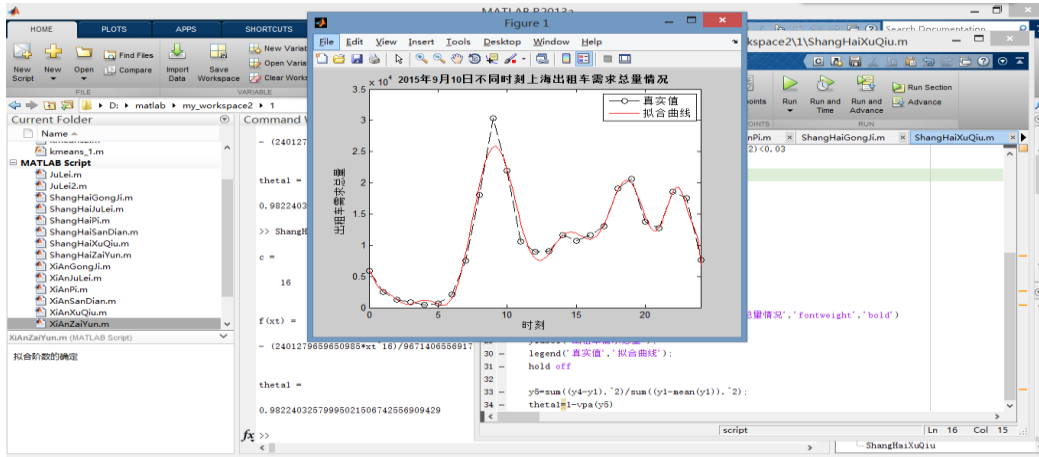


图 11：2015 年 9 月 10 日上海出租车需求拟合函数

经计算，出租车供给量与需求量的函数拟合程度 θ 值分别为 97.2161%、98.2240%，均大于 90%，可认为拟合程度达到要求，拟合函数可以使用。

表 5：上海出租车供给及需求函数

函数	上海出租车
供给函数	$D_0(t) = (7485355993172379 \cdot t^{20}) / 79228162514264337593543950336 - (438342472509979 \cdot t^{19}) / 19342813113834066795298816 + (3035718628471209 \cdot t^{18}) / 1208925819614629174706176 - (1610374139570469 \cdot t^{17}) / 9444732965739290427392 + (4684163543301997 \cdot t^{16}) / 590295810358705651712 - (4948313814419229 \cdot t^{15}) / 18446744073709551616 + (7852071876010713 \cdot t^{14}) / 1152921504606846976 - (298268947217459 \cdot t^{13}) / 2251799813685248 + (2246280942484617 \cdot t^{12}) / 1125899906842624 - (3290055834153119 \cdot t^{11}) / 140737488355328 + (7491403977478181 \cdot t^{10}) / 35184372088832 - (1647873296996751 \cdot t^9) / 1099511627776 + (8866660068026755 \cdot t^8) / 1099511627776 - (8962271272951597 \cdot t^7) / 274877906944 + (6637709410672437 \cdot t^6) / 68719476736 - (6945734520943715 \cdot t^5) / 34359738368 + (4865955916243901 \cdot t^4) / 17179869184 - (8402578987808037 \cdot t^3) / 34359738368 + (3818253896836985 \cdot t^2) / 34359738368 - (4393418302056267 \cdot t) / 274877906944 + 3118986393668967 / 274877906944$
需求函数	$Q_0(t) = -(2401279659650985 \cdot t^{16}) / 9671406556917033397649408 + (3506628649449021 \cdot t^{15}) / 75557863725914323419136 - (2308315153826953 \cdot t^{14}) / 590295810358705651712 + (3619706570887317 \cdot t^{13}) / 18446744073709551616 - (7524460693461519 \cdot t^{12}) / 1152921504606846976 + (2729639243468191 \cdot t^{11}) / 18014398509481984 - (5672995663961671 \cdot t^{10}) / 2251799813685248 + (2131318095730349 \cdot t^9) / 70368744177664 - (1155586451912119 \cdot t^8) / 4398046511104 + (3572035120930019 \cdot t^7) / 2199023255552 - (1921772421210759 \cdot t^6) / 274877906944 + (2775482047298817 \cdot t^5) / 137438953472 - (5083187137340841 \cdot t^4) / 137438953472 + (1341595924525593 \cdot t^3) / 34359738368 - (5239012746351475 \cdot t^2) / 274877906944 - (7370522834745623 \cdot t) / 8796093022208 + 1652097692509909 / 274877906944$

由①②式，我们可得到第 i 个分区出租车的需求函数 $Q(t, \beta_i)$ ；由 ③④式，得到第 i 个分区出租车的供给函数 $D(t, \alpha_i)$ ， $i = 1, 2, \Lambda, 15$ 。计算得到第 i 个分区的出租车供给比例系数 α_i 及需求比例系数 β_i 结果如下：

表 6：上海分区供求比例系数结果表

上海分区	α	β
1	0.104	0.094
2	0.092	0.09
3	0.034	0.038
4	0.141	0.134
5	0.050	0.045
6	0.076	0.075
7	0.030	0.028
8	0.103	0.102
9	0.006	0.006
10	0.018	0.018
11	0.009	0.009
12	0.179	0.176
13	0.032	0.028
14	0.047	0.054
15	0.079	0.103

(iii) 评价指标

根据以上所求到的函数指标，以及上海市城市分区确定，先确定出各 15 个分区的匹配程度数据，其数据分别为：

表 7：上海市 11 个分区的“供求匹配”程度值

区号	1	2	3	4	5	6	7	8
P 值	0.96315	0.966046	0.969939	0.965081	0.962838	0.966221	0.964152	0.966298
区号	9	10	11	12	13	14	15	
P 值	0.967881	0.967009	0.964814	0.966209	0.962246	0.970939	0.974433	

再结合加权平均的方法，利用函数：

$$P = \sum_{i=1}^k P_i \alpha_i \quad (i=1,2,\Lambda,k)$$

最终得到上海市的“供求匹配”程度为 $P=0.96638$ 。再与“供求匹配”程度对照表比照，知上海市的“供求匹配”程度等级为差。

3.3 结果检验

我们选取西安和上海这两个城市另外一天的出租车需求量及供给量，对拟合的供求函数进行检验。例如，选取 2015 年 9 月 9 日这天的数据，由所拟合的供求函数，取不同时刻得到相应的供求量的函数值，运用最小二乘思想，以拟合比例系数 $\theta \geq 90\%$ 为检验的衡量标准，若符合要求，则函数适用。经验证，西安出租车用该天的需求量数据计算的 θ 值为 95.245%，供给量数据计算结果为 96.674%；上海出租车用该天的需求量、供给量数据计算的 θ 值分别为 96.256%、94.785%均满足检验标准，故通过检验，拟合函数是适用的。

3.4 结果分析

1. 算例分析结果

根据模型对西安和上海两个城市的“供求匹配”程度进行求解，且其最终得到的“供求匹配”程度分别为：

表 8：西安和上海“供求匹配”程度等级结果表

城市	“供求匹配”程度等级
西安	良
上海	差

查询相关资料，知上海属于人数众多热门旅游城市，其对于出租车的需求基数非常大，但上海市的出租车数量与需求相比是不足的，上海市出租车的总数在近几年的年增加量不足 10%，因此产生了“供求匹配”程度等级为差的最终结果。而西安作为比上海人口总数较少的一级城市，虽然没有像上海那样的大量出租车需求，但存在古城旅游需求，因此西安市的出租车需求量仍然是较大的，不过其增长速度并不大，近五年的增长率均不超过 8%。西安市出租车需求总量与西安市出租车总量增长速度较为平衡，因此，最终得到西安市“供求匹配”程度为较差。

上海市与西安市的“供求匹配”程度均为不理想状态，其主要原因为西安、上海均作为城市总人口基数较大的热门旅游城市，有大量的出租车需求，但出租车的供给量不能达到实际需求。不过，相对而言，西安的需求量较小，匹配程度较高，且最终的匹配程度最终等级为一般。

2. 指标模型分析

通过运用西安及上海两个城市的具体算例分析，得到最终的评价结果分别为一般和差。然后查找两个地区关于出租车匹配相关的问卷调查分析，知西安与上海两城市的实际出租车匹配结果与本指标体系建立结果一致。且查的资料的结果是：上海评价为差，西安评价为较易。由于指标体系建立的分级方式不同，因此可能略有差异。但最终结果是有高类似度的。即说明本模型的指标建立是有效的。

四、问题二的分析、模型建立与求解

4.1 问题分析

问题二要求分析各公司的出租车补贴方案是否对“缓解打车难”有所帮助。

首先，对各公司的出租车补贴方案进行分析，我们查找了使用人数最多具有代表性的“滴滴出行”（原名“滴滴打车”）及“快的打车”软件情况。现将“滴滴打车”与“快的打车”的补贴政策变化作如下整理：

表 9：滴滴出行与快的打车补贴方案

软件 时间	滴滴出行补贴方案	快的打车补贴方案	补贴特点
2014 年 1 月 10 日	司机、乘客双向每单补贴 10 元	无	补贴初期，对金额的补贴刺激了乘客对出租车的需求，提高了司机工作的积极性
2014 年 1 月 20 日	不变	司机、乘客双向每单补贴 10 元	
2014 年 2 月 10 日	司机、乘客双向每单补贴 5 元	乘客每单补贴 10 元、司机每单补贴 5 元	补贴金额的下降使得客户少量流失
2014 年 2 月 17 日	乘客每单补贴 10-15	乘客每单补贴 11 元，司机	补贴高潮期，两家公司的竞

	元, 司机每单补贴 5 元	每单补贴 5-11 元	争使得补贴金额变动频繁, 并且采用随机金额补贴, 最高达到 20 元, 同时也推出其他补贴方式, 大大的刺激了出租车的需求及供给。
2014 年 2 月 18 日	开启“游戏补贴”模式, 同时也上调乘客补贴至 12-20 元且每天 3 次, 并送“全民飞机大战”中的战机	用支付宝付车费满 5 次, 在淘宝和天猫可享受退货包邮, 每单最少给乘客补贴 13 元且每天 2 次	
2014 年 3 月 4 日	部分城市补贴调整到起步价至 20 元不等	乘客每单补贴 10 元, 司机补贴不变	
2014 年 3 月 5 日	不变	乘客每单补贴 5 元	补贴热度下降期, 补贴金额的下调导致使用打车软件的客户流失, 但逐渐趋于稳定的使用量。
2014 年 3 月 7 日	乘客每单补贴随机 6-15 元	不变	
2014 年 3 月 22 日	不变	乘客每单补贴 3—5 元	
2014 年 3 月 23 日	乘客每单补贴 3-5 元	不变	
2014 年 5 月 17 日	乘客补贴“归零”		补贴后期, 对乘客及司机的补贴依次归零, 打车软件对出租车的供求无明显影响。
2014 年 7 月 9 日	司机每单补贴降为 2 元		
2014 年 8 月 9 日	取消对司机接单的常规补贴		

由上表及所查阅的资料, 在补贴政策实施期间, “滴滴打车”投入十数亿资金补贴给司机和乘客, 有上亿用户参与到多轮促销战中, 同时在补贴热潮的五个月中, “快的打车”覆盖的城市从 40 多个迅速扩展至近 300 个, 每天帮助数百万人出行。在这样一个能够及时了解到打车信息的服务平台, 并且具有补贴政策优惠条件的情况下, 大大刺激了乘客对出租车的需求, 拉动了出租车行业的发展。

我们有以下结论:

(1) “滴滴出行”、“快的打车”分别于 2012 年 6 月和 8 月进入市场至 2013 年底, 由于社会的熟知度较低, 使用人数较少, 为提高软件的使用人数, 于 2014 年 1 月发起补贴优惠;

(2) 自 2014 年 1 月至 3 月这两家软件对司机和乘客进行双重补贴活动, 并在此期间两大软件先后频繁的调整乘客与司机之间的补贴, 对出租车的供求量及供给量产生重要的影响;

(3) 在 2014 年 3 月底, 已有多地政府交通管理部门发布对打车软件的限制措施, 随后两家打车软件公司也分别下调补贴, 烧钱大战出现降温。至 8 月已经对乘客和司机的补贴归零, 此时注册打车软件的人数与之前已经有了大幅度的提高。

(4) 虽然补贴的热潮已经结束, 但在之后的特殊日期如节假日等, 偶尔推出优惠措施, 吸引用户的关注和使用, 避免了打车软件市场的萧条。

下图为 2012-2015 年打车软件累计注册用户量的条形统计图:

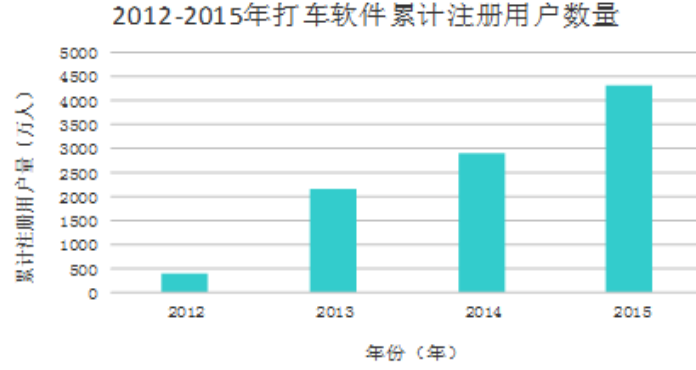


图 12: 2012-2015 年打车软件累计注册用户量条形统计图

由上图可直观地看出, 从 2012 年至 2015 年打车累计的注册用户量持续上升, 在 2012 年注册人数还不足 500 万人, 而在 2013 年注册人数有了明显的上升, 这段时间是打车软件的宣传推广使用期, 让更多的市民了解到打车软件的便捷性, 并尝试使用。从 2014 年至 2015 年, 注册人数再次出现大幅度上升, 在此期间受到打车软件补贴热潮的影响, 由此吸引了大量的市民选择这样的出行方式, 甚至使得有些乘客愿意增加打车时间, 也使司机愿意在拥堵的时间运行或开往偏远地区。

出现“打车难”的情况, 意味着乘客等车的时间较长, 或者有空车的情况但因开往的地方偏远, 司机不愿接单。我们以乘客等车时间及出租车的空载率衡量打车的难易程度, 打车软件采取的补贴措施调整了出租车的供给量和需求量分布, 而供求量的变化决定了等车时间及空载率, 进一步影响到“打车难”指数的取值。

4.2 模型建立与求解

4.2.1 建立模型

我们定义“打车难”指标受到乘客等车时间 T 及出租车空载率 δ 的影响, 以及司机拒载的情形, 因拒载的情况下等车时间无限长, 可将其归为乘客等车时间的影响。乘客等车时间越长, 出租车空载率越小, 代表打车越难, 有:

$$R = c_1 T + \frac{c_2}{\delta}$$

其中 R 为“打车难”指标, R 越大代表打车越难。

以下分步对上式中的函数进行具体说明:

(1) 建立供求函数

根据打车软件公司实施的补贴政策, 我们发现出租车的需求与供给受到补贴政策的影响显著, 其中包括乘客补贴 $m_1(t)$ 、司机补贴 $m_2(t)$ 以及其他非现金补贴 $m_3(t)$, 在问题一中我们已经得到了需求函数 $Q_0(t)$ 以及供给函数 $D_0(t)$, 打车软件的补贴方案导致需求及供给在原来函数的基础上发生改变, 建立新的需求函数 $Q(t)$ 及供给函数 $D(t)$ 如下:

$$Q(t) = [k_1 m_1(t) - k_2 m_2(t) + k_3 m_3(t)] + Q_0(t)$$

$$D(t) = [l_1 m_1(t) + l_2 m_2(t) + l_3 m_3(t)] + D_0(t)$$

其中, k_1, k_2, k_3 表示相应的补贴方式对需求量刺激的比例系数, l_1, l_2, l_3 表示相应的补贴方式对供给量刺激的比例系数。当对司机的补贴金额 $m_2(t)$ 增加时, 司机接单的数量增加, 减少了乘客的等车时间, 使得同一时间内出租车的需求降低。

定义 E 表示需求量与供给量的差值

$$E = |Q(t) - D(t)| \quad (7)$$

$$= |n_1 m_1(t) + n_2 m_2(t) + n_3 m_3(t) + Q_0(t) - D_0(t)|$$

其中 $n_1 = k_1 - l_1$, $n_2 = -k_2 - l_2$, $n_3 = k_3 - l_3$ 。因为司机补贴 $m_2(t)$ 的系数 n_2 一定为负, 我们直接考虑 $m_1(t) = m_3(t) = 0$ 这种仅对司机补贴的特殊情况, 此时司机补贴金额的增加, 缩短了供给量和需求量之间的差值, 使供求匹配度缩小, 使“打车难”的问题得以缓解。

(2) 建立等车时间及空载率模型

“打车难”的问题主要体现为乘客的等车时间 T 以及出租车的空载率 δ , 空载率为一天内出租车无客时行驶路程与行驶总路程的比值。

$$T = \begin{cases} b_1(Q - D) & Q \geq D \\ \max\left\{T_0, \frac{b_2}{D - Q}\right\} & Q < D \end{cases} \quad (8)$$

T_0 为乘客等车的最小满意时间, b_1, b_2 为转化为时间的比例系数。

当 $Q \geq D$ 时, 出租车的需求量大于供给量, 此时等车时间受到等车人数的影响, 我们以需求量与供给量的差值进行衡量; 当 $Q < D$ 时, 出租车的供给量大于需求量, 此时我们以乘客等车的最小满意时间及供给量与需求量差值的反函数比较, 建立等车时间的数学模型。

$$\delta = \begin{cases} \max\left\{\delta_0, \frac{b_3}{Q - D}\right\} & Q > D \\ b_4(D - Q) & Q \leq D \end{cases} \quad (9)$$

δ_0 为出租车空载率的最大满意程度, b_3, b_4 为转化为空载率的比例系数。

当 $Q \leq D$ 时, 出租车的供给量大于需求量, 此时空载率以供给量与需求量的差值进行衡量; 当 $Q > D$ 时, 出租车的需求量大于供给量, 此时我们以出租车空载率的最大满意程度及需求量与供给量差值的反函数比较, 建立出租车空载率的数学模型。

(3) 建立打车难易指标

记 R 为打车难易指标, 乘客等车时间越长, 出租车空载率越小, R 越大导致

代表打车越难，

$$R = c_1 T + \frac{c_2}{\delta}$$

其中， c_1, c_2 为比例系数，将⑦⑧⑨式带入上式，得到最终“打车难”指标为

$$R = \begin{cases} e_1 E + c_2 \max\left\{\delta_0, \frac{b_3}{E}\right\} & Q > D \\ c_1 \max\left\{\frac{b_2}{E}, T_0\right\} + e_2 \frac{1}{E} & Q \leq D \end{cases} \quad (10)$$

其中， $e_1 = c_1 b_1$ ， $e_2 = \frac{c_2}{b_4}$ 。

4.2.2 模型求解

仍然以西安及上海为代表，根据已经建立的打车难易指标 r ，比较这两个城市在补贴方案实施前后“打车难”的程度。经查找相关数据，乘客等车的最小满意时间 $T_0 = 5\text{min}$ ，出租车空载率的最大满意程度 $\delta_0 = 30\%$ 。因为西安和上海的经济情况不同以及当地人们消费水平的差异，导致因补贴对出租车供给量及需求量造成的刺激程度不同，即相应的比例系数不同。我们分别分析西安和上海补贴政策对“打车难”的影响。

表 10：西安、上海供求刺激比例系数

西安供求刺激系数		上海供求刺激系数	
需求量刺激系数	供给量刺激系数	需求量刺激系数	供给量刺激系数
$k_1 = 8 \ k_2 = 7 \ k_3 = 2$	$l_1 = 1 \ l_2 = 10 \ l_3 = 1$	$k_1 = 3 \ k_2 = 2 \ k_3 = 0.2$	$l_1 = 0.5 \ l_2 = 3 \ l_3 = 0.2$

(1) 西安市

(i) 补贴前的打车难易指数

打车公司未实施打车方案前，不存在各类补贴，即 $m_1(t) = m_2(t) = m_3(t) = 0$ ，

代入⑩式中得到 $R = 2.7023$ ，这是西安未补贴时打车难的指数。

(ii) 补贴后的打车难易指数

依据我们搜集到的打车公司的补贴方案，任选某一时间对应的补贴情况。例如在 2014 年 2 月 10 日“滴滴出行”补贴方案为司机与乘客双向补贴 5 元，即 $(m_1(t), m_2(t), m_3(t)) = (5, 5, 0)$ ，得到其 $R = 1.6523$ 。在 2014 年 2 月 17 日“快的打车”

补贴方案为乘客补贴 11 元，司机补贴 6 元，即 $(m_1(t), m_2(t), m_3(t)) = (11, 6, 0)$ ，得到 $R = 3.1423$ 。

(2) 上海市

(i) 补贴前的打车难易指数

打车公司未实施打车方案前，不存在各类补贴，即 $m_1(t) = m_2(t) = m_3(t) = 0$ ，

代入⑩式中得到 $R = 8.5603$ ，这是上海未补贴时打车难的指数。

(ii) 补贴后的打车难易指数

依据我们搜集到的打车公司的补贴方案，任选某一时间对应的补贴情况。例如在 2014 年 1 月 10 日“滴滴出行”补贴方案为司机与乘客双向补贴 10 元，即 $(m_1(t), m_2(t), m_3(t)) = (10, 10, 0)$ ，得到其 $R = 8.428$ 。在 2014 年 3 月 5 日“快的打车”补贴方案为乘客补贴 5 元，司机补贴 10 元，无其他非金额补贴，即 $(m_1(t), m_2(t), m_3(t)) = (5, 10, 0)$ ，得到 $R = 8.5516$ 。

4.3 结果分析

1. 算例分析结果

我们分别计算出了西安和上海实施补贴政策前“打车难”指数，以此作为标准与补贴后的具体补贴情况得到的“打车难”指数进行比较：

(1) 若补贴前的打车难指数小于补贴后的打车难指数，则说明补贴方案没有缓解“打车难”的问题；若补贴后的打车难指数大于补贴后的“打车难”指数，则说明补贴方案缓解了“打车难”的问题。

(2) 在计算西安补贴后某天的“打车难”指数中，出现了大于未补贴时的指数，可能是由于司机在就近的范围内接单即可获得补贴，因此更不愿去偏远的地区，使得当地的乘客打车难的情况不但没有缓解，反而更加严重。

2. 指标模型分析

由上述模型的建立，我们得到了受到乘客等车时间及出租车的空载率影响的“打车难”指数，用其来分析补贴方案是否对“打车难”的问题进行有效缓解。一般情况下补贴后对该问题会有所缓解，但对于一些偏远地区该下问题有可能会加重。

五、问题三的分析、模型建立与求解

5.1 问题分析

本问中要求创建一个新的打车软件服务平台，设计补贴方案，并论证其合理性。根据这个问题，我们进行如下因素分析：

1. 创建新打车软件的动机

根据问题一中的出租车供求匹配度、问题二中的打车难指数以及其对应的算例分析，发现存在下列问题。

(1) 出租车供求匹配度不高。问题一中的两个算例的匹配度均不为满意状态，明显的反映出实际生活中出租车供给和需求的匹配差异。

(2) 问题二中，在有补贴条件下，一定程度的缓解了打车难的问题，但仍旧出现了补贴无效用的情况。这就反映出补贴的方案存在缺陷。

2. 打车软件平台建立的思路

针对以上问题，我们首先分析问题二提供的各公司出租车补贴方案，可以看出在打车软件服务平台建立初期阶段，各公司对司机、乘客双向补贴的力度都很大，目的在于吸引更多的司机和乘客接触并使用该服务平台。随着时间的推移，平台注册用户积累到一定数量后，各公司逐渐减少补贴金额。之后，继续减少补贴金额，只在节假日或者举办活动期间给予部分用户补贴。我们所建立的服务平台将继续沿用问题二提供的补贴方案，并在此基础上提出了以下新方案：

(1) 针对不同时段、不同区域给予相应的补贴，用来保证在整个城市范围内，达到乘车载客的一致性，公平性，均等性。以求降低空载率。

(2) 提出期望等待时间的指标, 考虑当实际等待时间大于期望等待时间时, 给予用户相应的补贴, 减少打车难度指数。

最后, 根据所提出的新方案, 建立模型。建立模型时以增高出租车供求匹配度和降低打车难指数为目标, 提出成本函数的概念, 最终分别建立了以降低空载率为主要目的的出租车均衡分配补贴模型、基于人车匹配的最优供求匹配动态模型以及以降低打车难指数为目的的等待时长补贴模型。在进行具体计算时, 还计算了所考虑地区出租车组成供给量的最佳配置, 同时可为政府对于该地区出租车配置量提供建议。

5.2 模型建立与求解

5.2.1 模型建立

1. 建立出租车均衡分配补贴模型

在问题二中我们发现在补贴后, 出现了“打车难”不但没有缓解反而加重的特殊情况, 这种情况的出现往往是由于司机不愿开往偏僻地区, 导致空载率较高, 因此我们建立出租车均匀分配补贴模型, 使出租车开往任何地方都具有均等性。

对以下三种情况出租车接单进行加倍补贴:

1) 行驶偏僻的地区; 2) 途径拥堵的路段; 3) 人流出行的高峰期。

t 时刻两点间的成本函数表示为:

$$D(t, A, B) = \lambda(t, A, B)d(A, B) \quad \lambda(t, A, B) \geq 1$$

其中 $D(t, A, B)$ 表示 t 时刻行驶 A, B 两点间的成本, $\lambda(t, A, B)$ 为行驶 A, B 两点间补贴系数, 当其为 1 时表示没有进行补贴。

2. 建立最优供求匹配动态模型

在问题一中我们建立了出租车供求匹配度指标, 为了达到对出租车资源的充分调动和利用, 我们需要进一步建立供求匹配动态模型, 直接确定每个出租车去往的需求位置。

设在 t 时刻, 有 X 辆出租车可供给, 同时有 Z 名乘客对出租车存在需求 (出租车车辆数利用文献中的方法计算得到)。若 $X < Z$ 时, 即出租车的供给量小于需求量, 我们向政府反映实际情况, 申请增加出租车的供给量。若 $X \geq Z$ 时, 为保证一一匹配可进行, 在 X 辆可供给的出租车中选择 Z 辆作为实际的可供给量, 以 t 时刻第 i 个车到第 j 个目的地的成本最小思想, 我们建立如下的优化模型:

$$\begin{aligned} \min V &= \sum_{j=1}^Z \sum_{i=1}^X G_{ij}(t) \alpha_{ij} \\ s.t. &\begin{cases} \sum_{i=1}^X \alpha_{ij} = 1 & j = 1, \Lambda, Z \\ \sum_{j=1}^Z \alpha_{ij} = 1 & i = 1, \Lambda, Z \\ \alpha_{ij} = 0 \quad or \quad 1 \end{cases} \end{aligned}$$

以所有出租车开往相匹配乘客的总成本 V 为目标函数, 以第 i 个车响应第 j 个

乘客需求 α_{ij} 为决策变量，响应记为 1 否则为 0。定义 t 时刻第 j 个乘客搭乘第 i 个出租车的成本函数：

$$G_{ij}(t) = \lambda(t, A(i), B(j)) \cdot d(A(i), B(j))$$

若出租车已经载有乘客，记 $G_{ij}(t) = \infty$ ，表示该车在时刻 t 不能进行匹配。通过 Matlab 软件运算，可得到最佳的供求匹配方案。

3. 建立等待时间模型

乘客打车等待时间，作为决定打车难指数的影响因素，其中也包含司机拒载的情况，我们希望将打车难的现象得以缓解，因此以降低打车难指数为目的，建立等待时间模型。

在上述建立的最优供求匹配动态模型中，我们已经将得到的最优结果，记为第 j_0 个乘客搭乘第 i_0 个出租车。根据距离时间公式，建立等待时间模型如下：

$$T_{ij} = \frac{d((t, A(i_0), B(j_0)))}{v(i_0)}$$

其中， T_{ij} 表示在时刻 t 第 j_0 个乘客搭乘第 i_0 个出租车的等待时间， $d((t, A(i_0), B(j_0)))$ 为在时刻 t 第 j_0 个乘客所在位置 B 与搭乘第 i_0 个出租车所在位置 A 之间难的距离， $v(i_0)$ 表示第 i_0 个出租车的行驶速度。

T_0 为乘客等车的最小满意时间，若 $T_{ij} \leq T_0$ ，则认为达到缓解打车难的目的；若 $T_{ij} > T_0$ ，即乘客等车时间较长，可对其进行一定金额的补贴，但是如果该乘客不愿等待，则交易失败，没有达到缓解打车难的目的。对于交易失败的情况，我们可以通过调整补贴金额，降低这种现象发生的概率。

4. 合理性分析

在上述三点中，我们分别基于出租车的空载率、供求匹配度以及乘客等待时间建立模型，必然符合模型所需条件。其中供求匹配度恰好为问题一中建立的指标，对出租车资源进行合理评判，而出租车的空载率与乘客等车时间均为问题二中决定打车难指数的影响指标，且通过上述三个模型的建立，使得供求匹配度提高，打车难指数下降，这是符合我们期望的状态，因此所建立的模型是合理的。

5. 2. 2 模型求解

以西安市数据为例进行所建模型的应用。由于不存在出租车和乘客位置点的数据，因此对人车匹配模型进行模拟求解。求解过程如下。

1. 计算出租车供给量的最佳配置量

首先计算出出租车承担的西安居民出行周转量：

$$F_1 = I_1 G_1 S_1 R_1 = 855.29 \times 3.06 \times 6.61 \times 2.56\% = 442.87$$

其中， F_1 表示西安出租车承担的城市居民出行周转量（万人·km）， I_1 为西安

居民人口总量（万人）； G_1 为西安居民人均日出行次数； S_1 为西安居民平均选择出租车出行的距离（km） R_1 为西安居民选择出租车在出行方式体系中所占的比例。

然后计算出出租车承担的西安流动人口出行周转量，

$$F_2 = I_2 G_2 R_2 S_2 = 47.46 \times 2.85 \times 34\% \times 6.61 = 303.99$$

式中： F_2 为出租车承担的流动人口出行周转量（ 10^4 人·km）

I_2 为流动人口人口总量（ 10^4 人）； G_2 为流动人口人均日出行次数； R_2 为流动人口出行方式结构中出租车所占的比例； S_2 为流动人口平均以出租车方式出行的距离（km）

2. 建立广义距离矩阵

首先，查找西安市的地理纬度跨度和经度跨度，计算出西安市的实际长度和纵向高度。将其以 10m 的间隔进行横纵的划分，并求出广义距离矩阵。但由于划分过细，数据量过大，Matlab 软件运行该程序需要耗费大量的时间。因此，我们仅以西安市的城区为代表进行划分，且取 200m 为划分间隔进行划分，得到划分的城区图。

以每个划分小区间的中心点记为该小区间的代表点，继而定义广义距离，即任意两点之间的距离定义为该两点所在区间的距离，而且两区间的距离定义是指两区间的横向和纵向连接线的长度。其示意分析图如下：

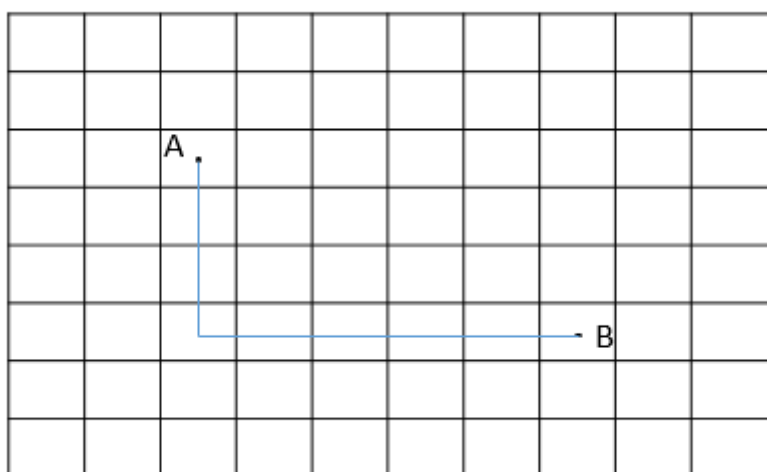


图 13：广义距离示意图

通过对西安地区的城区的划分和运用 Matlab 求解得到任意两点的广义距离胞元数组，其部分数据截图如下：

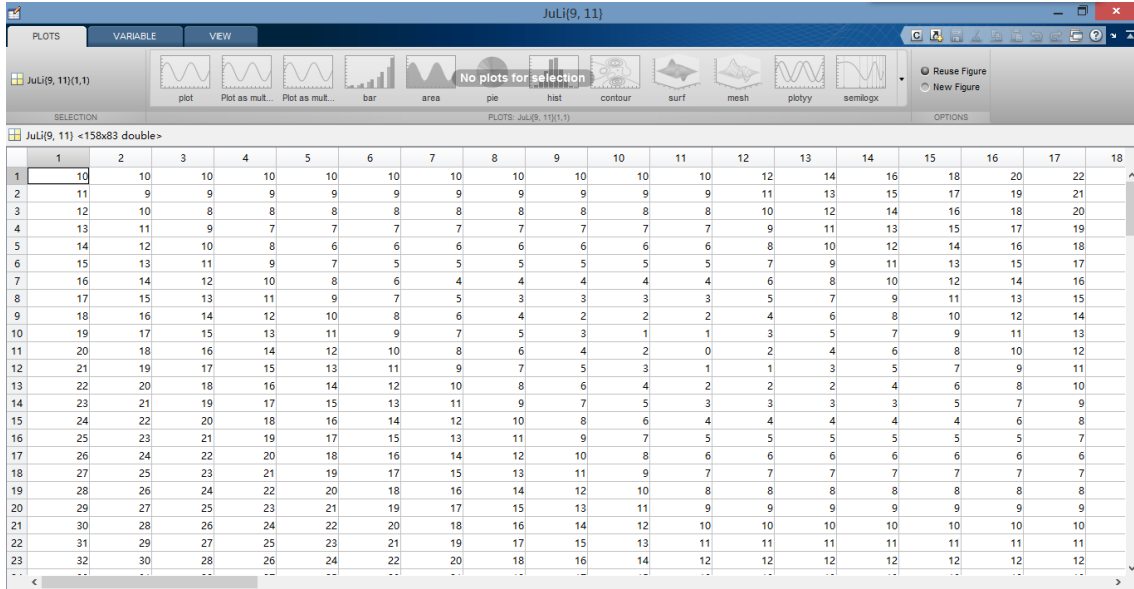


图 14：广义距离胞元数组部分数据截图

3. 建立成本函数

由西安的出租车供求情况，我们确定以下三种情况出租车接单进行加倍补贴：1) 行驶偏僻的地区；2) 途径拥堵的路段；3) 人流出行的高峰期。具体做法为将以上三种情况的区间点矩阵进行合并，统一计算。

t 时刻两点间的成本函数表示为：

$$D(t, A, B) = \lambda(t, A, B)d(A, B) \quad \lambda(t, A, B) \geq 1$$

当出现上面三种情况时，给定补贴系数 $\lambda(t, A, B) = 2$ ，其他情况该值为 1，即不进行补贴。

4. 建立最优供求匹配动态模型

由以上计算结果已经得到成本函数矩阵，根据已经建立的优化模型：

$$\begin{aligned} \min V &= \sum_{j=1}^Z \sum_{i=1}^Z G_{ij}(t) \alpha_{ij} \\ s.t. &\begin{cases} \sum_{i=1}^X \alpha_{ij} = 1 & j = 1, \Lambda, Z \\ \sum_{j=1}^Z \alpha_{ij} = 1 & i = 1, \Lambda, Z \\ \alpha_{ij} = 0 \quad or \quad 1 \end{cases} \end{aligned}$$

用 Lingo 软件进行计算，求得最终的人车匹配矩阵 α ，且矩阵部分数据截图如下：

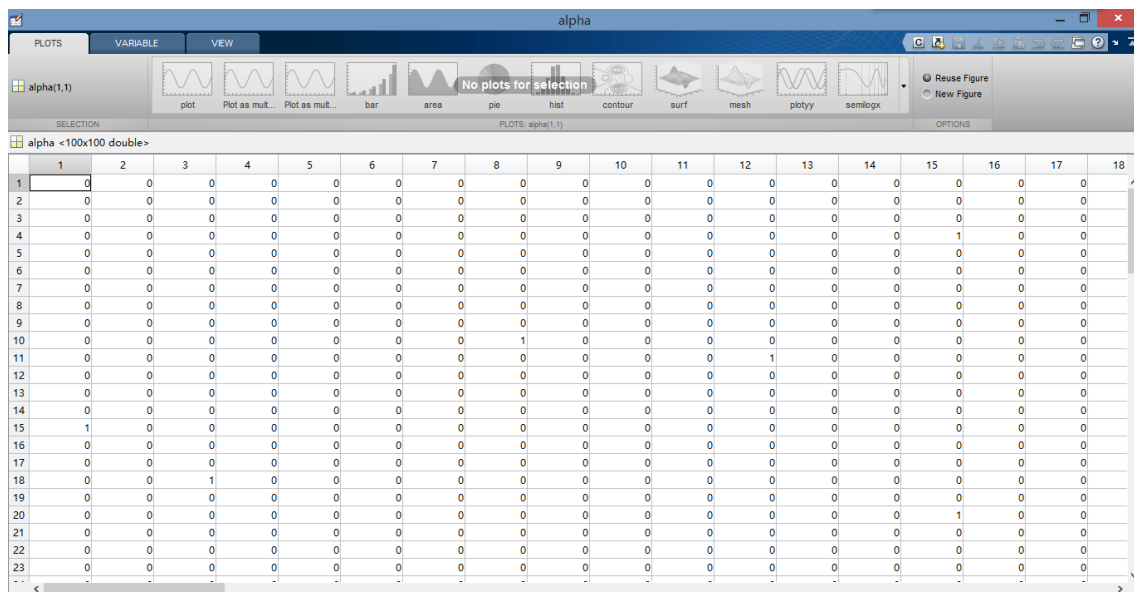


图 15：人车匹配矩阵部分数据截图

5. 等待时间计算

由线性规划模型已解得人车匹配矩阵。假设方程：

$$T_{ij} = \frac{d((t, A(i_0), B(j_0)))}{v(i_0)}$$

中的 $v(i_0) = 40\text{km/h}$ 是恒定值，即可求得 100 组人车匹配对的等待时间，且等待时间矩阵的数据如下：

表 11：等待时间数据表

5.5	1.95	1.6	2.3	11.4	3.35	4	4.35	4.1	1.7
2.75	1.95	1.65	2.3	5.7	6.8	8	2.75	4	1.8
2.7	3.9	1.65	2.35	2.9	6.8	4.05	5.2	4	1.9
5.4	1.85	1.75	2.35	6	3.5	4.1	2.5	4	1.9
2.6	3.6	1.8	4.9	6	3.55	4.1	2.4	1.75	1.9
5	1.8	1.9	2.55	3.05	7.2	4.15	2.25	1.75	2.05
2.25	1.8	3.8	5.1	3.05	3.75	4.2	4.5	1.65	2.05
4.4	1.75	2.15	2.6	6.1	7.6	4.25	2.2	1.6	2.05
4.1	1.55	2.15	5.3	3.1	7.6	4.25	2.15	1.65	4.4
4	1.55	2.2	2.65	3.15	3.85	8.7	2.1	1.65	2.25

定义期望等待时间为 5min，当乘客实际等待时间大于 5min 时，为其提供补贴 5 元现金。

5.3 结果分析

由查找到的数据，通过所建立的模型，我们可以计算得到西安所供给的出租车与需要打车的乘客之间一一匹配的对应关系，以及最短的等车时间，也可用于其他地区的分析。这样的结果是令人满意的，分步建立的数学模型形式上也较为简单，易于分析和理解。将其运用于实际生活中，也有较强的适用性。在初始的补贴方案、均衡的补贴方案、优化的补贴方案是具有高有效性的。

六、模型的优缺点评价

6.1 模型的优点

(1) 我们在建立模型之后，根据搜集到代表性城市的相关数据，都将模型应用到具体的实例中，体现了模型的易操作和适用性。

(2) 在问题一中，我们运用 K-Means 算法建立聚类分析模型，便于计算，有效地对空间进行划分，并且划分区域简单明了。在对数据的拟合过程中，结合三次样条插值法及线性插值法进行数据的增添，即保证了增添数据的准确性，同时为之后拟合多项式奠定基础。

(3) 在问题二中，找到两家具有代表性的打车软件，根据其补贴政策进行分析，并较好的以空载率及打车等待时间作为衡量“打车难”的标准。

(4) 在问题三中，我们在前两问的基础上建立更加人性化、智能化的打车软件服务平台，并论证了其满足合理性。

6.2 模型的缺点

(1) 问题一中对空间和时间的划分不够细致，主要考虑具有代表性的城市，及打车的高峰期，从而分析不够全面。

(2) 问题三中我们建立的打车软件平台，有些乘客因对等车时间不满而存在交易失败的情况。

七、问题的进一步考虑与推广

本文研究了“互联网+”新时代，打车软件对出租车的资源配置问题，我们建立了出租车的供给函数，定义了供求匹配度，分析了在不同时空出租车的资源配置情况，又建立了打车难指数模型，以及最后设计了新的打车软件平台。

我们可以以本文的模型为基础，进一步推广至国内外更多地区和不同时节及节假日情况下，研究出租车的资源配置情况。以及更全面的针对不同的补贴政策，考虑对“打车难”问题的影响。

八、参考文献

- [1] SHAW ShihLung, Modeling of taxi drivers' experience for routing applications, Science China(Technological Sciences), 中国科学:技术科学(英文版), 2010, S1期: 12-15
- [2] 吕玉强, 秦勇, 贾利民, 董宏辉, 贾献博, 孙智源, 基于出租车 GPS 数据聚类分析的交通小区动态划分方法研究, 物流技术 2010, 5 (216): 86-88
- [3] 陆建, 王炜, 城市出租车拥有量确定方法(J), 交通运输工程学报, 2004, 4(1): 1671-1637

%第一问程序：附录一至附录十二
%第二问程序：附录十三至附录十四
%第三问程序：附录十五至附录十六

附录一

%XiAnGongJi.m

%此程序得到 2015 年 9 月 10 日不同时刻西安出租车供给情况
%求得西安市出租车供给拟合函数并作出 2015 年 9 月 10 日不同时刻西安出租车供给情况图像

```
%gongJi 表示当前的出租车供给量
gongJi=[152,70,30,31,28,41,51,200,991,836,431,313,261,213,380,512,601,680,1321,1623,1351,119
7,903,523,134];
x=0:24;
plot(x,gongJi,'k--o');
xlim([0,24]);

x1=min(x):0.2:max(x);
%得到对 gongJi 插值后的数据 y1
y1=interp1(x,gongJi,x1,'linear');

%拟合阶数的确定
for i=1:30
    y2=polyfit(x1,y1,i);
    Y=polyval(y2,x1);%计算拟合函数在 x 处的值
    if sum((Y-y1).^2)/sum((y1-mean(y1)).^2)<0.005
        c=i;
        break;
    end
end
%得到 c 即为多项式的阶数
warning off %消除拟合多项式时提示的警告
syms xt f(xt)
%f(xt)表示拟合曲线函数
f(xt)=poly2sym(y2,xt)
y3=eval(f(xt));
y4=subs(y3,xt,x1);
hold on
plot(x1,y4,'r');
title('2015 年 9 月 10 日不同时刻西安出租车供给情况','fontweight','bold')
xlabel('时刻');
ylabel('出租车供给总量');
```

```
legend('真实值','拟合曲线','Location','NorthWest');  
hold off  
%theta1 表示拟合程度的比例系数  
y5=sum((y4-y1).^2)/sum((y1-mean(y1)).^2);  
theta1=1-vpa(y5)
```

附录二

%XiAnXuQiu.m

%此程序得到 2015 年 9 月 10 日不同时刻西安出租车需求总量情况

%求得西安市出租车需求拟合函数并作出 2015 年 9 月 10 日不同时刻西安出租车需求情况图像

%xianXu 表示从 9 月 4 日至 9 月 11 日不同时刻出租车需求数据

%xian_x2 表示 9 月 10 日不同时刻的出租车需求数据

```
xianXu=[244,228,108,134,80,144,545,2098,2739,1302,842,1049,850,956,1155,1741,1930,2120,260  
9,1869,2020,2648,2620,1010,119,82,60,42,43,85,129,197,313,269,371,526,368,455,671,1048,764,5  
99,624,587,538,538,389,295,145,64,30,46,21,75,249,823,1047,530,387,437,420,381,483,731,637,6  
12,893,493,331,479,274,190,87,63,24,32,24,78,236,830,993,430,318,296,220,265,303,479,499,479,  
604,413,423,479,382,195,122,99,46,32,33,59,177,940,1161,434,278,282,274,266,375,568,513,752,1  
424,917,593,690,393,219,108,65,36,78,46,62,342,1784,1660,496,474,482,581,575,633,864,485,718  
,1035,680,474,408,410,179,112,57,47,36,69,80,335,1375,1463,558,328,386,471,572,701,856,808,17  
07,2501,2005,1536,1430,733,244,506,405,250,257,232,480,796,1722,1711,1295,1170,1347,1307,13  
43,1390,1551,1491,1548,1789,1589,1416,1400,1106,949];
```

```
xian_x2=xianXu(144:168);
```

```
x=0:24;
```

```
plot(x,xian_x2,'k--o');
```

```
xlim([0,24]);
```

```
x1=min(x):0.2:max(x);
```

%得到对 xian_x2 插值后的数据 y1

```
y1=interp1(x,xian_x2,x1,'linear');
```

%拟合阶数的确定

```
for i=1:50
```

```
    y2=polyfit(x1,y1,i);
```

```
    Y=polyval(y2,x1);%计算拟合函数在 x 处的值
```

```
    if sum((Y-y1).^2)/sum((y1-mean(y1)).^2)<0.006
```

```
        c=i
```

```
        break;
```

```
    end
```

```
end
```

%得到 c 即为多项式的阶数

```
warning off %消除拟合多项式时提示的警告
```

```
syms xt f(xt)
```

%f(xt)表示拟合曲线函数

```
f(xt)=poly2sym(y2,xt)
```

```
y3_1=eval(f(xt));
```

```
y4_1=subs(y3_1,xt,x1);
```

```

hold on
plot(x1,y4_1,'r');
title('2015 年 9 月 10 日不同时刻西安出租车需求总量情况','fontweight','bold')
xlabel('时刻');
ylabel('出租车需求总量');
legend('真实值','拟合曲线');
hold off
%theta1 表示拟合程度的比例系数
y5=sum((y4_1-y1).^2)/sum((y1-mean(y1)).^2);
theta1=1-vpa(y5)

```

附录三

%XiAnZaiYun.m

%此程序得到 2015 年 9 月 10 日不同时刻西安在运出租车总量情况

%求得西安市在运出租车拟合函数并作出 2015 年 9 月 10 日不同时刻西安在运出租车总量情况图像

%xianZai 表示从 9 月 4 日至 9 月 11 日不同时刻在运出租车数据

%xian_z2 表示 9 月 10 日不同时刻的在运出租车数据

```
xianZai=[11072,8692,6974,5083,5188,9396,15435,21388,24944,24689,22412,20546,20345,20464,18571,15152,12484,12590,12031,11294,11408,11603,12199,13212,11753,9896,7361,5387,5653,10325,17458,22292,24135,23160,20497,18502,18071,18438,16695,13181,10387,11509,11548,11461,11526,11600,12416,12649,11808,9503,7208,5276,5988,11693,18963,17592,16023,18302,18507,17593,17805,19804,19356,14284,10994,11013,9941,10605,11385,12044,13142,14086,11920,9423,7457,5207,5482,10998,19335,19688,17941,19497,19211,19556,19587,20069,19200,14729,11454,11759,11199,11791,12609,12141,12081,13663,11696,9558,6973,5894,5766,10320,18248,18174,16365,17911,19566,18766,19136,21262,20005,15548,12119,11094,9005,9622,10997,11089,12314,13702,10860,8776,6594,5958,5745,9780,16527,15579,15181,18730,17762,17112,16729,17687,16734,13799,10890,10265,9692,9555,11111,11749,12333,12406,19140,15433,11502,8346,7983,14640,26875,27430,24634,24065,21171,19698,20220,20839,20283,15603,14131,13687,13251,13803,14541,14395,15283,17230,11806,9690,7283,5755,5931,10613,17611,19218,17963,19907,19571,18788,18607,19448,18367,14488,11673,11588,10605,10973,11382,11646,12156,13213];
```

```
xian_z2=xianZai(144:168);
```

```
x=0:24;
```

```
plot(x,xian_z2,'k--o');
```

```
xlim([0,24]);
```

```
ylim([0,30000]);
```

```
x1=min(x):0.2:max(x);
```

%得到对 xian_z2 插值后的数据 y1

```
y1=interp1(x,xian_z2,x1,'spline');
```

%拟合阶数的确定

```
for i=1:30
```

```
    y2=polyfit(x1,y1,i);
```

```
    Y=polyval(y2,x1);%计算拟合函数在 x 处的值。
```

```
    if sum((Y-y1).^2)/sum((y1-mean(y1)).^2)<0.05
```

```
        c=i
```

```
        break;
```

```
    end
```

```
end
```

%得到 c 即为多项式的阶数

```
warning off          %消除拟合多项式时提示的警告
```

```

syms xt f(xt)
%f(xt)表示拟合曲线函数
f(xt)=poly2sym(y2,xt)
y3=eval(f(xt));
y4=subs(y3,xt,x1);
hold on
plot(x1,y4,'r');
title('2015 年 9 月 10 日不同时刻西安在运出租车总量情况','fontweight','bold')
xlabel('时刻');
ylabel('在运出租车总量');
legend('真实值','拟合曲线');
hold off

y5=sum((y4-y1).^2)/sum((y1-mean(y1)).^2);
%theta1 表示拟合程度的比例系数
theta1=1-vpa(y5)

```


附录四

%XiAnJuLei.m

%此程序运用 kmeans 函数求得 2015 年 9 月 10 日上午 10 点西安市出租车的分布区域划分情况
%A1 存放的是经过筛选后(对不符合西安市经纬度的点进行剔除)10 点西安市出租车的分布数据
%A1(:,1)表示经度, A(:,2)表示纬度, A(:,3)代表出租车数量

load A1
%用 kmeans 函数对 A1 中的数据进行划分, 分为 11 个聚类
[IDX,C,sumd,D]=kmeans(A1(:,1:2),11);
%IDX 表示每个样本点所在的类别
%C 表示所聚类别的中心点坐标位置
%sumd 表示每个类内各点到中心点的距离之和
%D 表示每个点到各类中心点的距离

comb=[IDX,A1];
[M,N]=size(comb);

%对 n1:n11 赋初值
n1=1;n2=1;n3=1;n4=1;
n5=1;n6=1;n7=1;n8=1;
n9=1;n10=1;n11=1;

%fenQu1 表示在一类区域里的点, fenQu2 表示在二类区域里的点, 依次类推
for i=1:M

```
    switch comb(i,1)
        case 1
            fenQu1(n1,1:3)=comb(i,2:4);n1=n1+1;
        case 2
            fenQu2(n2,1:3)=comb(i,2:4);n2=n2+1;
        case 3
            fenQu3(n3,1:3)=comb(i,2:4);n3=n3+1;
        case 4
            fenQu4(n4,1:3)=comb(i,2:4);n4=n4+1;
        case 5
            fenQu5(n5,1:3)=comb(i,2:4);n5=n5+1;
        case 6
            fenQu6(n6,1:3)=comb(i,2:4);n6=n6+1;
        case 7
            fenQu7(n7,1:3)=comb(i,2:4);n7=n7+1;
        case 8
            fenQu8(n8,1:3)=comb(i,2:4);n8=n8+1;
```

```
case 9
    fenQu9(n9,1:3)=comb(i,2:4);n9=n9+1;
case 10
    fenQu10(n10,1:3)=comb(i,2:4);n10=n10+1;
case 11
    fenQu11(n11,1:3)=comb(i,2:4);n11=n11+1;
end
end
%即把所有数据分为 11 个聚类
```

附录五

%XiAnLiTiTu.m

%2015 年 9 月 10 日上午 10 点西安市的出租车分布的散点图和立体图

load A1 %A1 存放的是经过筛选后(对不符合西安市经纬度的点进行剔除)10 点西安市出租车的分布数据

x=A1(:,1);

y=A1(:,2);

z=A1(:,3);

figure(1)

scatter(x,y,5,z)

title('2015 年 9 月 10 日 10 时西安在运出租车分布散点图','fontweight','bold')

xlabel('东经/度');

ylabel('北纬/度');

figure(2)

[X,Y,Z]=griddata(x,y,z,linspace(min(x),max(x),100),linspace(min(y),max(y),100),'cubic');%插值%
surf(X,Y,Z);

title('2015 年 9 月 10 日 10 时西安在运出租车分布立体图','fontweight','bold')

xlabel('东经/度');

ylabel('北纬/度');

zlabel('数量'); %2015 年 9 月 10 日上午 10 点西安市的出租车分布的散点图和立体图

load A1 %A1 存放的是经过筛选后(对不符合西安市经纬度的点进行剔除)10 点西安市出租车的分布数据

x=A1(:,1);

y=A1(:,2);

z=A1(:,3);

figure(1)

scatter(x,y,5,z)

title('2015 年 9 月 10 日 10 时西安在运出租车分布散点图','fontweight','bold')

xlabel('东经/度');

ylabel('北纬/度');

figure(2)

[X,Y,Z]=griddata(x,y,z,linspace(min(x),max(x),100),linspace(min(y),max(y),100),'cubic');%插值%
surf(X,Y,Z);

title('2015 年 9 月 10 日 10 时西安在运出租车分布立体图','fontweight','bold')

xlabel('东经/度');

ylabel('北纬/度');

zlabel('数量');

附录六

%XiAnQiuP.m

%alpha1 表示的是 2015 年 9 月 10 日上午 10 点西安市出租车的分布情况聚类后不同区域占总区域的占比

%beta1 表示的是 2015 年 9 月 10 日上午 10 点西安市出租车需求量的聚类后不同区域占总区域的占比

```
alpha1=[0.076,0.081,0.148,0.142,0.090,0.100,0.110,0.100,0.089,0.028,0.030];
```

```
beta1=[0.083,0.077,0.149,0.135,0.091,0.125,0.124,0.112,0.085,0.031,0.012];
```

```
x=0:24;
```

```
x1=min(x):0.2:max(x);
```

```
for i=1:11
```

```
    N_max(i)=(alpha1(i)*y3-beta1(i)*y3_1)/(alpha1(i)*y3+beta1(i)*y3_1);
```

```
    max_t=subs(N_max(i),xt,x1);
```

```
    max_p=abs(max_t);
```

```
    maxValue(i)=max(double(max_p));
```

```
end
```

```
for i=1:11
```

```
    temp_Value(i)=alpha1(i)*maxValue(i);
```

```
end
```

```
P=sum(temp_Value)
```

%即为西安出租车资源的匹配程度 P

附录七

%ShangHaiGongJi.m

%此程序得到 2015 年 9 月 10 日不同时刻上海出租车供给情况

%求得上海市出租车供给拟合函数并作出 2015 年 9 月 10 日不同时刻上海出租车供给情况图像

%gongJi2 表示当前的出租车供给量

gongJi2=[11118,14261,13221,9829,3341,1478,642,554,320,1251,6407,15392,27685,13489,11908,7252,10324,11385,9733,9960,10283,13442,16823,12332,10278];

x=0:24;

plot(x,gongJi2,'k--o');

xlim([0,24]);

x1=min(x):0.2:max(x);

%得到对 gongJi2 插值后的数据 y1

y1=interp1(x,gongJi2,x1,'linear');

%拟合阶数的确定

for i=1:50

 y2=polyfit(x1,y1,i);

 Y=polyval(y2,x1);%计算拟合函数在 x 处的值。

 if sum((Y-y1).^2)/sum((y1-mean(y1)).^2)<0.03

 c=i

 break;

 end

end

%得到 c 即为多项式的阶数

warning off

syms xt f(xt)

%f(xt)表示拟合曲线函数

f(xt)=poly2sym(y2,xt)

y3=eval(f(xt));

y4=subs(y3,xt,x1);

hold on

plot(x1,y4,'r');

title('2015 年 9 月 10 日不同时刻上海出租车供给情况','fontweight','bold')

xlabel('时刻');

ylabel('出租车供给总量');

legend('真实值','拟合曲线','Location','NorthWest');

hold off

%theta1 表示拟合程度的比例系数

y5=sum((y4-y1).^2)/sum((y1-mean(y1)).^2);

theta1=1-vpa(y5)

附录八

%ShangHaiXuQiu.m

%此程序得到 2015 年 9 月 10 日不同时刻上海出租车需求总量情况

%求得上海市出租车需求拟合函数并作出 2015 年 9 月 10 日不同时刻上海出租车需求情况图像

%shangHaiXu 表示从 9 月 4 日至 9 月 11 日不同时刻出租车需求数据

%shangHai_x2 表示 9 月 10 日不同时刻的出租车需求数据

```
shangHaiXu=[6458,2625,1772,1191,1205,3431,10921,27226,52042,47726,42139,46740,55510,563
40,54632,42717,58384,82059,80443,47638,46307,53288,51384,31595,4280,2694,1178,874,781,154
1,2014,4634,8183,8538,6718,6434,6888,7190,6947,6472,7546,9675,8011,7336,7007,6178,5122,350
1,1873,1069,681,514,621,2033,6227,10862,14647,9113,6224,6300,6805,7816,7584,7088,8138,1148
9,10539,8344,7222,8413,6944,4004,2033,1115,767,595,616,2381,7781,18579,29412,18419,7447,63
55,7125,7924,7476,7219,8138,12383,12969,9359,9002,11265,8534,4657,2141,1216,942,541,643,21
43,7271,16784,28495,20012,8561,6985,7674,9436,8113,8252,9566,14150,15970,10420,9526,12514
,11052,5444,2390,1278,962,442,615,2055,7462,17351,28939,19913,9267,8162,8165,11118,9811,10
061,11016,16920,19119,12622,11951,16319,13651,5930,2516,1318,913,484,642,2117,7527,18050,
30310,21911,10571,8994,9012,11585,10719,11665,13041,19043,20641,13759,12787,18533,17558,
7627,5360,3907,3059,2350,2867,5479,8116,10776,13454,11199,8889,8862,9001,9409,8966,8893,9
529,11091,10771,9637,9563,10425,9622,7079];
```

```
shangHai_x2=shangHaiXu(144:168);
```

```
x=0:24;
```

```
plot(x,shangHai_x2,'k--o');
```

```
xlim([0,24]);
```

```
x1=min(x):0.2:max(x);
```

%得到对 shangHai_x2 插值后的数据 y1

```
y1=interp1(x,shangHai_x2,x1,'linear');
```

%拟合阶数的确定

```
for i=1:30
```

```
    y2=polyfit(x1,y1,i);
```

```
    Y=polyval(y2,x1);%计算拟合函数在 x 处的值。
```

```
    if sum((Y-y1).^2)/sum((y1-mean(y1)).^2)<0.03
```

```
        c=i
```

```
        break;
```

```
    end
```

```
end
```

%得到 c 即为多项式的阶数

```
warning off
```

```
syms xt f(xt)
```

%f(xt)表示拟合曲线函数

```
f(xt)=poly2sym(y2,xt)
```

```

y3_1=eval(f(xt));
y4_1=subs(y3_1,xt,x1);
hold on
plot(x1,y4_1,'r');
title('2015 年 9 月 10 日不同时刻上海出租车需求总量情况','fontweight','bold')
xlabel('时刻');
ylabel('出租车需求总量');
legend('真实值','拟合曲线');
hold off
%theta1 表示拟合程度的比例系数
y5=sum((y4_1-y1).^2)/sum((y1-mean(y1)).^2);
theta1=1-vpa(y5)

```

附录九

%ShangHaiZaiYun.m

%此程序得到 2015 年 9 月 10 日不同时刻上海在运出租车总量情况

%求得上海市在运出租车拟合函数并作出 2015 年 9 月 10 日不同时刻上海在运出租车总量情况图像

%shangHaiZai 表示从 9 月 4 日至 9 月 11 日不同时刻在运出租车数据

%shangHai_z2 表示 9 月 10 日不同时刻的在运出租车数据

```
shangHaiZai=[160903,117069,78082,54699,55571,97936,151409,201873,240253,255116,244837,210605,200079,206442,210955,214227,212300,187577,177850,183217,187309,192873,192292,173965,133897,96571,76379,51736,54604,92097,143927,186873,224561,228904,232606,210493,201865,207748,208725,207629,205344,189374,184270,186459,191883,199538,204468,192130,159920,112833,74311,49097,51194,105877,185582,199608,205747,245836,261025,233858,225943,225580,217613,218865,214889,183247,181982,202987,227281,233400,234812,221296,175948,124540,84127,55996,56244,112428,194580,185139,175798,206756,233874,221856,218274,220445,222759,225638,215239,181958,177899,204215,232727,244307,243287,240859,199433,140354,88812,56768,55072,109508,187446,186385,177614,205887,223333,213928,210333,210359,215653,216165,203694,171518,166281,197312,225420,238634,238694,233988,197636,147383,93917,60263,59021,115933,189625,180427,173310,205487,222271,206112,203216,200575,205471,205952,195655,166603,161740,190324,217838,233278,235204,236998,202363,149760,97499,63638,61204,113946,188309,180132,174708,210286,219499,208129,204718,204116,206122,207839,191562,165190,158112,186555,215099,227961,237861,233385,179804,129229,86328,57166,57921,111566,186236,190367,187585,217759,236036,221939,216894,218513,219300,221234,212506,181955,177034,200327,222912,234085,237603,230173];
```

```
shangHai_z2=shangHaiZai(144:168);
```

```
x=0:24;
```

```
plot(x,shangHai_z2,'k--o');
```

```
xlim([0,24]);
```

```
ylim([0,300000]);
```

```
x1=min(x):0.2:max(x);
```

%得到对 shangHai_z2 插值后的数据 y1

```
y1=interp1(x,shangHai_z2,x1,'spline');
```

%拟合阶数的确定

```
for i=1:30
```

```
    y2=polyfit(x1,y1,i);
```

```
    Y=polyval(y2,x1);%计算拟合函数在 x 处的值。
```

```
    if sum((Y-y1).^2)/sum((y1-mean(y1)).^2)<0.05
```

```
        c=i
```

```
        break;
```

```
    end
```



```

end
%得到 c 即为多项式的阶数
warning off
syms xt f(xt)
%f(xt)表示拟合曲线函数
f(xt)=poly2sym(y2,xt)
y3=eval(f(xt));
y4=subs(y3,xt,x1);
hold on
plot(x1,y4,'r');
title('2015 年 9 月 10 日不同时刻上海在运出租车总量情况','fontweight','bold')
xlabel('时刻');
ylabel('在运出租车总量');
legend('真实值','拟合曲线');
hold off

y5=sum((y4-y1).^2)/sum((y1-mean(y1)).^2);
%theta1 表示拟合程度的比例系数
theta1=1-vpa(y5)

```

附录十

%ShangHaiJuLei.m

%此程序运用 kmeans 函数求得 2015 年 9 月 10 日上午 9 点上海市出租车的分布区域划分情况
%A2 存放的是经过筛选后(对不符合上海市经纬度的点进行剔除)9 点上海市出租车的分布数据
%A2(:,1)表示经度, A(:,2)表示纬度, A(:,3)代表出租车数量

load A2

%用 kmeans 函数对 A1 中的数据进行划分, 分为 15 个聚类

[IDX,C,sumd,D]=kmeans(A2(:,1:2),15);

%IDX 表示每个样本点所在的类别

%C 表示所聚类类的中心点坐标位置

%sumd 表示每个类内各点到中心点的距离之和

%D 表示每个点到各类中心点的距离

comb=[IDX,A2];

[M,N]=size(comb);

%对 n1:n15 赋初值

n1=1;n2=1;n3=1;n4=1;

n5=1;n6=1;n7=1;n8=1;

n9=1;n10=1;n11=1;n12=1;

n13=1;n14=1;n15=1;

%fenQu1 表示在一类区域里的点, fenQu2 表示在二类区域里的点, 依次类推
for i=1:M

switch comb(i,1)

case 1

fenQu1(n1,1:3)=comb(i,2:4);n1=n1+1;

case 2

fenQu2(n2,1:3)=comb(i,2:4);n2=n2+1;

case 3

fenQu3(n3,1:3)=comb(i,2:4);n3=n3+1;

case 4

fenQu4(n4,1:3)=comb(i,2:4);n4=n4+1;

case 5

fenQu5(n5,1:3)=comb(i,2:4);n5=n5+1;

case 6

fenQu6(n6,1:3)=comb(i,2:4);n6=n6+1;

case 7

fenQu7(n7,1:3)=comb(i,2:4);n7=n7+1;

case 8

fenQu8(n8,1:3)=comb(i,2:4);n8=n8+1;

```

case 9
    fenQu9(n9,1:3)=comb(i,2:4);n9=n9+1;
case 10
    fenQu10(n10,1:3)=comb(i,2:4);n10=n10+1;
case 11
    fenQu11(n11,1:3)=comb(i,2:4);n11=n11+1;
case 12
    fenQu12(n12,1:3)=comb(i,2:4);n12=n12+1;
case 13
    fenQu13(n13,1:3)=comb(i,2:4);n13=n13+1;
case 14
    fenQu14(n14,1:3)=comb(i,2:4);n14=n14+1;
case 15
    fenQu15(n15,1:3)=comb(i,2:4);n15=n15+1;
end
end
%即把所有数据分为 15 个聚类

```

附录十一

%ShangHaiLiTiTu.m

%2015 年 9 月 10 日上午 9 点上海市的出租车分布的散点图和立体图

load A2 %A2 存放的是经过筛选后(对不符合上海市经纬度的点进行剔除)9 点上海市出租车的分布数据

```
x=A2(:,1);
```

```
y=A2(:,2);
```

```
z=A2(:,3);
```

```
figure(1)
```

```
scatter(x,y,5,z)
```

```
title('2015 年 9 月 10 日 9 时上海在运出租车分布散点图','fontweight','bold')
```

```
xlabel('东经/度');
```

```
ylabel('北纬/度');
```

```
figure(2)
```

```
[X,Y,Z]=griddata(x,y,z,linspace(min(x),max(x),100)',linspace(min(y),max(y),100),'cubic');%插值%
```

```
surf(X,Y,Z);
```

```
title('2015 年 9 月 10 日 9 时上海在运出租车分布立体图','fontweight','bold')
```

```
xlabel('东经/度');
```

```
ylabel('北纬/度');
```

```
zlabel('数量');
```

附录十二

%ShangHaiQiuP.m

%alpha2 表示的是 2015 年 9 月 10 日上午 9 点上海市出租车的分布情况聚类后不同区域占总区域的占比

%beta2 表示的是 2015 年 9 月 10 日上午 9 点上海市出租车需求量的聚类后不同区域占总区域的占比

```
alpha2=[0.104,0.0917,0.034,0.140,0.050,0.076,0.030,0.103,0.006,0.017,0.0093,0.178,0.031,0.047,0.078];
```

```
beta2=[0.094,0.090,0.038,0.134,0.045,0.075,0.028,0.102,0.006,0.018,0.0088,0.176,0.028,0.054,0.102];
```

```
x=0:24;
```

```
x1=min(x):0.2:max(x);
```

```
for i=1:15
```

```
    N_max(i)=(alpha2(i)*y3-beta2(i)*y3_1)/(alpha2(i)*y3+beta2(i)*y3_1);
```

```
    max_t=subs(N_max(i),xt,x1);
```

```
    max_p=abs(max_t);
```

```
    maxValue(i)=max(double((max_p)));
```

```
end
```

```
for i=1:15
```

```
    temp_Value(i)=alpha2(i)*maxValue(i);
```

```
end
```

```
P=sum(temp_Value)
```

%即为上海出租车资源的匹配程度 P

附录十三

%XiAnQiuR.m

```
%此程序求解 R，即打车难易度（值越大表示越难打到车）
%m1,m2,m3 分别表示对乘客的补贴金额，对出租车的补贴金额以及其他补贴金额
function R=XiAnQiuR(m1,m2,m3)
load QD_XiAn
%QD_XiAn 存放的是第一问中拟合后的值，其中 Q_XiAn 表示出租车需求量，D_XiAn 表示出租车供给量
global Q_XiAn D_XiAn      %声明全局变量
Q0=mean(Q_XiAn);
D0=mean(D_XiAn);
aver=Q0-D0;
%载入初始化数据
load ChuShi_XiAn
k1=8;k2=7;k3=2;
l1=1;l2=10;l3=1;
n1=k1-l1;
n2=-k2-l2;
n3=k3-l3;
e=qiuE(m1,m2,m3);
q=qiuQ(m1,m2,m3);
d=qiuD(m1,m2,m3);
if q > d
    R=t1*e+c2*max(sig,b3/e);
else
    R=c1*max(b2/e,t0)+t2/e;
end
%三个子函数分别求 Q,D,E
%其中 Q 表示补贴后的出租车需求量，D 表示补贴后的出租车供给量
function Q=qiuQ(m1,m2,m3)
    Q=(k1*m1-k2*m2+k3*m3)+Q0;
end

function D=qiuD(m1,m2,m3)
    D=(l1*m1+l2*m2+l3*m3)+D0;
end

function E=qiuE(m1,m2,m3)
    E=abs(n1*m1+n2*m2+n3*m3+aver);
end
end
```

附录十四

%ShangHaiQiuR.m

```
%此程序求解 R，即打车难易度（值越大表示越难打到车）
%m1,m2,m3 分别表示对乘客的补贴金额，对出租车的补贴金额以及其他补贴金额
function R=ShangHaiQiuR(m1,m2,m3)
load QD_ShangHai
%QD_ShangHai 存放的是第一问中拟合后的值，其中 Q_ShangHai 表示出租车需求量，
D_ShangHai 表示出租车供给量
global Q_ShangHai D_ShangHai      %声明全局变量
Q0=mean(Q_ShangHai);
D0=mean(D_ShangHai);
aver=Q0-D0;
%载入初始化数据
load ChuShi_ShangHai
k1=3;k2=2;k3=0.2;
l1=0.5;l2=3;l3=0.2;
n1=k1-l1;
n2=k2-l2;
n3=k3-l3;
e=qiuE(m1,m2,m3);
q=qiuQ(m1,m2,m3);
d=qiuD(m1,m2,m3);
if q > d
    R=t1*e+c2*max(sig,b3/e);
else
    R=c1*max(b2/e,t0)+t2/e;
end
%三个子函数分别求 Q,D,E
%其中 Q 表示补贴后的出租车需求量，D 表示补贴后的出租车供给量
function Q=qiuQ(m1,m2,m3)
    Q=(k1*m1-k2*m2+k3*m3)+Q0;
end

function D=qiuD(m1,m2,m3)
    D=(l1*m1+l2*m2+l3*m3)+D0;
end

function E=qiuE(m1,m2,m3)
    E=abs(n1*m1+n2*m2+n3*m3+aver);
end
end
```

附录十五

%WangGeJuLi.m

```
%此程序将西安市化作网格，求出每个网格中心与其他网格中心的距离
%并将最终随机分布的人和出租车的坐标间的距离矩阵导出
%31620 和 16740 表示的是西安市市区内经纬度经过转化得到的数据，单位为 m
P=floor(31620/200);
Q=floor(16740/200);

%JuLi 是大小为 158*83 的胞元数组，每个胞元大小为 158*53，里边存放的是其余网格点和当前序号坐标的距离
for i=1:P
    for j=1:Q
        for m=1:P
            for n=1:Q
                JuLi{i,j}(m,n)=abs(m-n)+abs(n-j);
            end
        end
    end
end

%划分四个偏僻地区
for i=1:floor(P/4)
    for j=1:floor(Q/4)
        JuLi{i,j}=2*JuLi{i,j};
    end
end

for i=1:floor(P/4)
    for j=floor(Q*3/4):Q
        JuLi{i,j}=2*JuLi{i,j};
    end
end

for i=floor(P*3/4):P
    for j=1:floor(Q/4)
        JuLi{i,j}=2*JuLi{i,j};
    end
end

for i=floor(P*3/4):P
    for j=floor(Q*3/4):Q
        JuLi{i,j}=2*JuLi{i,j};
    end
end
```



```

        end
    end

%划分一个拥挤区域
for i=floor(P*3/8):floor(P*5/8)
    for j=floor(Q*3/8):floor(Q*5/8)
        JuLi{i,j}=2*JuLi{i,j};
    end
end

%划分一个高峰区域
for i=floor(P/4):floor(P/2)
    for j=floor(Q/2):floor(Q*3/4)
        JuLi{i,j}=2*JuLi{i,j};
    end
end

%在 158*83 的矩阵中产生随机数，per 中存放随机的 100 人及他们的坐标
r1=randi(200,158,83);
[m1,ind1]=sort(r1(:),'descend');
szX1=size(r1);
[i1,j1]=ind2sub(szX1,ind1);
q1=[i1,j1,m1];
per=q1(1:100,:);

%在 158*83 的矩阵中产生随机数，taxi 中存放随机的 100 辆车及其坐标
r2=randi(200,158,83);
[m2,ind2]=sort(r2(:),'descend');
szX2=size(r2);
[i2,j2]=ind2sub(szX2,ind2);
q2=[i2,j2,m2];
taxi=q2(1:100,:);

%计算 Cij, Cij 表示 100 个随机分布的出租车和 100 个随机分布的人之间的距离
% Cij=d(p(i),p(j))
for i=1:100
    for j=1:100
        Cij(i,j)=JuLi{taxi(i,1),taxi(i,2)}(per(j,1),per(j,2));
    end
end

%将 Cij 的数据写入 Cij.txt 以便在 lingo 中求解非线性约束规划
dlmwrite('F:\Cij.txt',Cij);

```

附录十六

%YouHua.txt

!该程序使用 lingo 求解非线性约束规划;

model:

sets:

I1/1..100/:ii;

J1/1..100/:jj;

links(I1,J1):C,X;

endsets

data:

ii=1,2,3,4,5,6,7,8,9,10;

jj=1,2,3,4,5,6,7,8,9,10;

C=@file('F:\Cij.txt');

enddata

min=@sum(links(i,j):C(i,j)*X(i,j));

@for(I1(i):@sum(J1(j):X(i,j))=1);

@for(J1(j):@sum(I1(i):X(i,j))=1);

@for(links(i,j):bin(X(i,j)));

end