

“互联网+”时代的出租车市场资源配置问题研究

摘要

城市出租车行业是现代交通体系中的一个重要的组成部份。然而在我国的大城市当中,“打车难”的问题一直无法得到有效解决。随着“互联网+”时代的到来,各类打车软件相继出现,出租车行业出现巨大变革。居民打车的方式从被动低效的“盲目匹配”转变为主动高效的“在线速配”,真正意义上实现了资源的有效配置。

针对**问题一**,首先在网络上通过“滴滴快滴智能出行平台”以及“出租车 OD 出行平台”进行了大量的数据挖掘,得到 9 月 5 日至 9 月 10 日杭州市不同经纬度坐标下的乘客需求量、出租车分布、乘客等待时间等数据。并在此基础上,抽取了 9 月 8 日 16:00-17:00 时段的打车需求量数据,通过“**watershed**”**水线阈值算法**构建出杭州市道路交通网络的拓扑结构,在任意两个节点之间建立了满足供需匹配的**最少需求量的规划模型**。以期望等待时间为主要约束条件,计算各地所需的最小供给量,并与该地实际需求对比,从等待时间、需求量、满载率、司机搜索时间 4 个角度,建立了评价“供需匹配度”的 4 个指标。通过层次分析法确定权重,使用 **Bonferroni 算子**建立最终“**供需匹配度**”的**评价模型**。同时,从实际与期望的比率出发,通过杭州市空间和时间上的划分,分别对三个时间段三类不同区域计算“供需匹配度”,分析资源分配所存在的问题。我们发现:不同的时间和空间下,出租车的供需之间存在巨大差异。

针对**问题二**,我们首先根据搜集到的数据,以微观经济学的**实证分析方法**为基础,对各种补贴方案所带来的供需比例变化作了定性分析与定量分析,建立需求与供给曲线的“**Marshall**”**剪刀模型**,从社会现象背后的经济规律揭示了“打车难”的原因。之后使用函数模型解方程组,根据不同补贴政策带来的影响程度,构建**正负效应差**指标。在带入实际数据计算后,得出各公司的补贴方案对“缓解打车难”问题确实有帮助的结论。

对于**问题三**,要求制定补贴方案问题,构建**吸引客户的理性选择模型**,利用边际分析的方法确定最优解,从而计算出不同补贴方案对客户的吸引程度。构建新的多目标规划模型,并将问题二模型的方程组作为新的约束条件,计算正负效应差达到最大的情况下的最优补贴方案,并验证其合理性。

本文建立了多种模型来研究各种打车软件对出租车市场资源配置的影响程度,以及补贴政策、等待时间等因素对“供需平衡”的影响,而且基于现行的补贴政策仍存在着不足之处的现状,还进一步构建优化模型进行完善,从而改善出租车资源配置。

关键词 供求匹配度 最少需求量 正负效应差 多目标规划 watershed 算法

一、问题重述

1.1 问题背景

交通运输网络是城市正常运行和发展的基础，合理的公共交通秩序有利于提升整个城市的运转速度和资源分配效率。伴随着市民对出行速度、舒适度等需求的提高，出租车市场呈现日益激烈的局面，受到出租车数量的限制，常常出现“打车难”的现象。

而伴随着互联网时代的来临，各类打车软件相继走入市场，为人们所知。打车软件具有快捷、高效的特点，可以在乘客和出租车司机之间建立直接的联系，尽可能减少出租车的空载时间，同时也为乘客在高峰期时段以及城市繁华路段打车提供了便捷的途径。

同时，包括腾讯、阿里巴巴在内的各大公司为了更有效地在全国范围内推广新型的打车软件，相继推出了各类优惠及补贴方案，很大程度上改变了城市本身出租车的供求匹配程度以及资源分配效率。因此，正确制定打车软件补贴政策可以最大程度上发挥出出租车行业的潜力，并给出租车司机和乘客双方都带来实在的优惠，以改变目前出租车市场存在的资源匹配不合理的现状。

1.2 问题的提出

为了更好地分析出租车市场目前的资源匹配程度以及各类打车软件的出现所带来的影响，本文提出了以下问题：

(1)了解出租车市场目前的资源匹配状况，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)研究各个公司推出的打车软件所提供的补贴方案，同时根据问题一中提出的合理模型分析各类补贴方案是否对“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、问题分析

问题一：

问题一要求建立合理的指标，并分析不同时空出租车资源的“供求匹配”程度。我们以杭州为例，在网络上搜集到与出租车运行状况相关的各类数据作为基本指标，包括乘客需求量、出租车分布、乘客等待时间等。在大量数据的基础上，建立了以乘客需求量为标准的杭州市出租车运行状况区域分化图，建立了 30 个节点，并且找出了任意两个节点之间的最短路程。同时，将一天 24 小时以运营繁忙程度进行划分，得出了不同时间杭州市出租车的通行状况。针对本题中所要求的“供需匹配”程度的评价标准，我们采取约束时间求最小出租车需求量模型，对已确定的各项指标进行了规划，得到了 30

个节点在有时间约束条件下的最小出租车需求量，同时通过改变最短时间约束得到了相应的需求量的变化图。

问题二：

问题二对各公司的打车补贴政策是否对缓解“打车难”有帮助，因此我们计划引入“Marshall 剪刀”供需模型，分别对各公司的打车补贴产生的影响进行定性和定量的分析，并最终建立需求函数，最终得出各个公司的补贴政策带来的影响是否有助于缓解“打车难”的问题。

问题三：

问题三要求在现有打车软件的基础上，以创建一个新的打车软件服务平台为背景，建立新型的打车补贴方案。我们需要对现有的打车软件所存在的优点和弊端进行有效的分析，根据现实生活中环境的变化，将每日高峰期、天气因素、等待时间等多个因素纳入到补贴方案之中，并建立合理的数学函数模型，使消费者和出租车司机的利益均得到最大化，使出租车与乘客需求量之间的“供求资源”配置得到最大的优化。

三、符号说明

Or	乘客出发地组成的集合
De	乘客目的地组成的集合
D_{ij}^1	从 i 区域到 j 区域使用打车软件的乘客需求量
D_{ij}^2	从 i 区域到 j 区域不使用打车软件的乘客需求量
D_{ij}	从 i 区域到 j 区域总乘客需求量
T_i	乘客在 i 区域的平均等待时间
T_{ij}	出租车从 i 区域到 j 区域所产生的时间
T_{ij}^1	乘客从发出打车信号到被确定接单之间的等待时间
T_{ij}^2	乘客从被确定接单到上车之间的等待时间
P_{ij}^1	使用打车软件的乘客从区域 i 到区域 j 所产生的费用
p_{ij}^2	未使用打车软件的乘客从区域 i 到区域 j 所产生的费用
α_1	乘客等待时间的价值参数
α_2	乘客上车时间的价值参数
q_{ij}	乘客从 i 区域到 j 区域的基础车费
ϕ	影响乘客需求和出租车服务的不确定因素
N_{ki}	未使用打车软件从 k 区域到 i 区域搜索乘客的空载出租车数量
N_{kij}	使用打车软件从 k 区域到 i 区域搜索到 j 区域乘客的空载出租车数量

M_{ij}	从 i 区域到 j 区域被占用的出租车数量
t_i	未使用打车软件的出租车在 i 区域搜索乘客的时间
t_{ij}	使用打车软件的出租车从 i 区域到 j 区域搜索乘客的时间
H	整个城市内出租车被占用总时间
\overline{N}_k^1	未使用打车软件的从区域 k 开始搜索乘客的空载出租车总数
\overline{N}_k^2	使用打车软件从区域 k 开始搜索乘客的空载出租车总数

四、模型假设

本文提出以下合理假设：

- (1) 假设不存在交通拥挤、道路施工、公路宽窄等因素对道路通行能力的影响；
- (2) 排除出租车司机因道路拥挤而拒绝在高峰期进入繁华路段的情况；
- (3) 不考虑乘客中途取消订单的情况。

五、模型的建立与求解

5.1 问题一建模与解答

5.1.1 数据的收集与整理

(1) 基于网络上滴滴快的智能出行平台，利用 Firefox 浏览器的控制台抓取网络平台数据，最终获取了 9 月 5 日—9 月 10 日杭州市出租车每小时分布情况、打车难易度、打车需求量、被抢单时间、车费、以及出行轨迹的时间序列数据，再使用集搜客 Goseeker 网页抓取软件，采用网页抓取的方式进行数据挖掘，得到了杭州市在这 5 天内每一个小时各个数据点的经度、纬度、以及各项指标的实际值。

滴滴快的智能出行平台网页如图 1 所示：

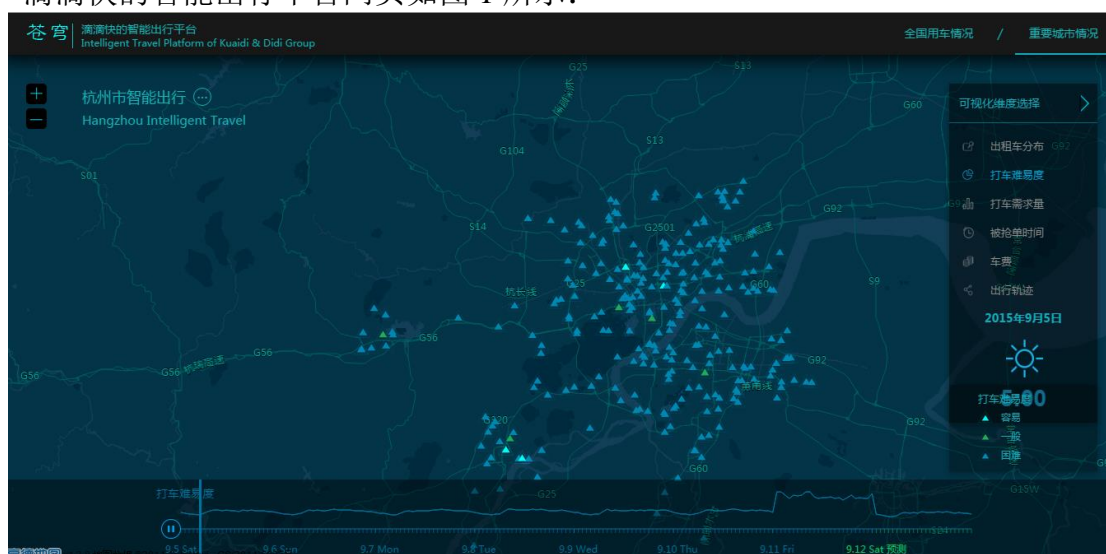


图 1. 滴滴快的智能出行平台网页图

(2) 在得到各项指标数据之后，利用 MATLAB 软件编程将杭州市每日各个地理位置坐标，即不同的经度和纬度，所对应的出租车需求量、供应量等数据从原始数据中提取出来并导入 EXCEL 表格之中，得到模型分析的有效数据，部分如图 2 所示：

表 1 某时间部分有效横截面数据

序号	经度 E	纬度 N	打车需求量
1	120.1788	30.287	309
2	120.1767	30.2826	302
3	120.1792	30.2701	295
4	120.1771	30.2796	282
5	120.1707	30.2679	252
6	120.1752	30.2605	242
7	120.1641	30.3017	240
8	120.1593	30.3036	238
9	120.1754	30.2549	227
10	120.1736	30.2943	215
11	120.1316	30.2845	213
12	120.1451	30.2745	212
13	120.1787	30.2962	204
14	121.3605	31.2041	202

5.1.2 基于“watershed”算法的区域属性加权网络模型

我们首先根据杭州市打车需求信息的密度分布，对杭州市进行区域划分，挖掘每个独立区域的代表特征点，以此作为图的节点。之后，提取所有节点之间的出租车运行路径，并以路径的长短为图进行加权。最后将各个节点的出租车需求量、出租车提供量，乘客的等待信息给节点加入属性。具体步骤如下：

(1) 数据的预处理

我们抽取了正常工作日 9 月 8 日（星期三）16:00-17:00 时段的数据为例，进行统计分析。

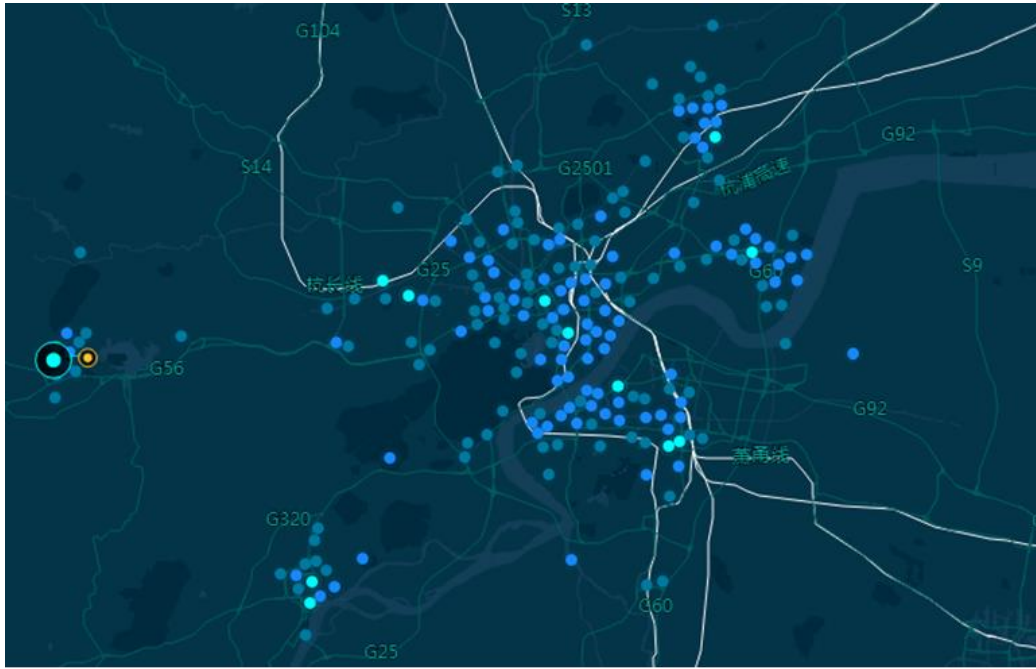


图 2. 9 月 8 日(星期三)16:00-17:00 时段杭州市出租车需求量分布

根据经纬度坐标，得到杭州市的基于打车需求信息的“模拟地图”，由于在建立坐标图的过程中发现原始数据存在位置显著超出杭州市地理范围的奇异值，于是我们根据杭州市地理位置信息，将研究的样本界定在 $[119.5^{\circ}\text{E}, 120.5^{\circ}\text{E}] \times [29.5^{\circ}\text{N}, 31^{\circ}\text{N}]$ 范围之内，再次利用 Matlab 软件实现筛选过程，从 301 条原始数据中筛选出 287 条符合条件的样本数据，并作出散点图，如图 3 所示：

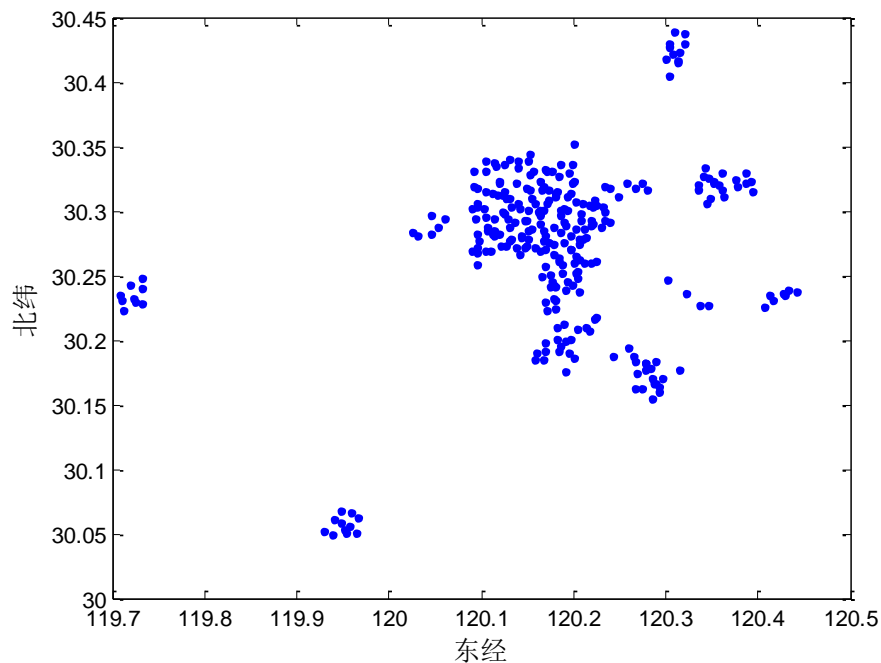


图 3. 打车需求信息节点坐标图

(2) 模型构建

对于这 287 个不同地理位置的地点组成的数据集，采用 watershed 水线阈值算法将数据集进行分类，从而构建出区域属性加权网络模型。具体步骤为：

Step 1: 划分交通小区

首先从地理位置的角度出发，设定一个较大的阈值将两个目标分开，通过减小阈值使得被划分的目标边界相向扩张，从而将现有地点划分成 30 组交通小区，简化该城市的空间模型，具体划分过程如图 4 所示：

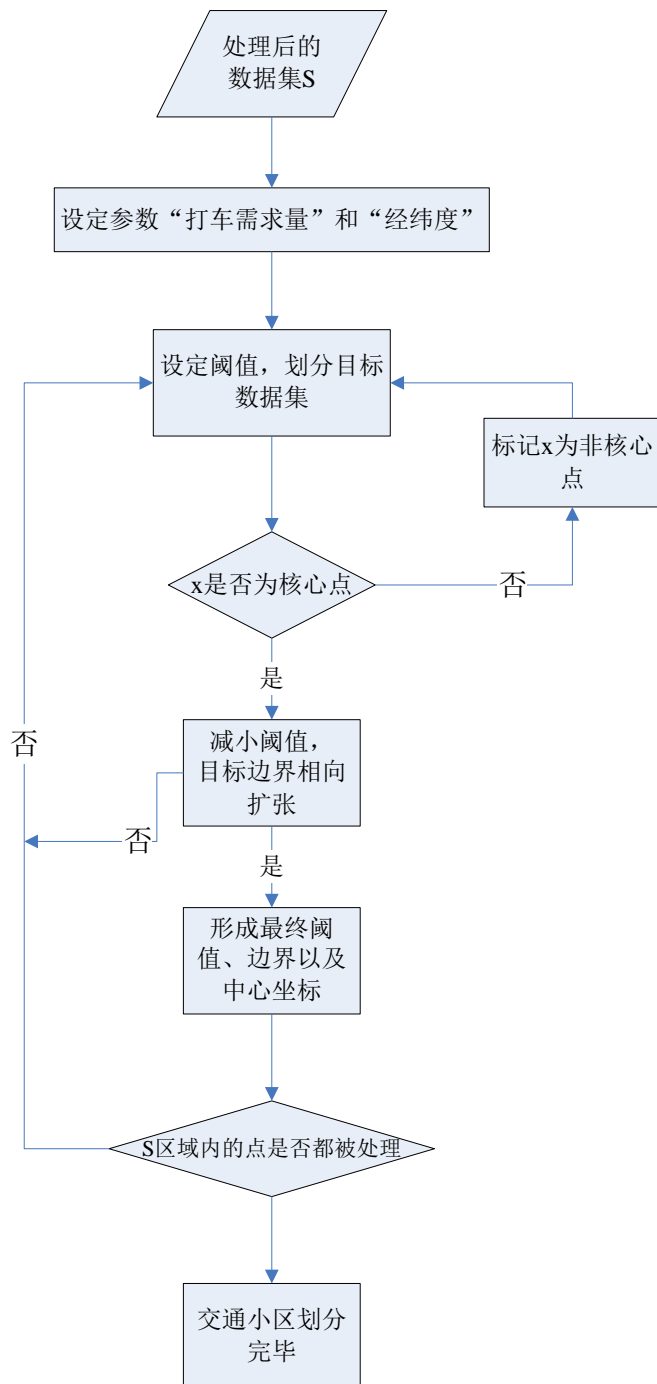


图 4. Watershed 算法划分交通小区流程图

在“打车需求信息节点坐标图”的基础上，采用原始数据中各个位置坐标（以经纬度为准）下的出租车需求量作为划分标准，进行了对原散点图分布的聚类绘制，得到的结果如图 5 所示：

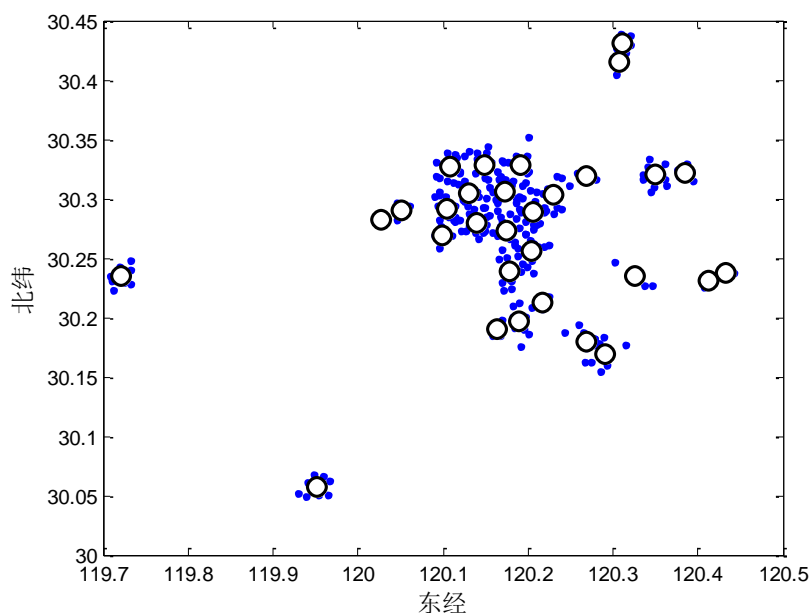


图 5. 出租车需求规划图

对比杭州市城区划分图，不难发现出租车需求量在东经 120.2° 、北纬 30.3° 左右的地区呈现密集分布状态，而此区域主要位于杭州市的上城区和下城区及其附近区域，在出租车需求量分布较为稀疏的地区主要位于半山、江干区、三墩等郊区、城市周边地区，而东经 120.1° 、北纬 30.2° 的区域打车需求量为 0，这正好与当地的地形地貌相吻合，所以结合二者分析可得在丘陵地区打车需求量较多，在山区和水道密集的地区打车需求量较少，这与实际经验是相吻合的，可见以此标准进行划分具有一定的合理性和现实意义，故现将杭州市划分为 30 个通行状况不同的区域，以每个区域的中心位置为节点，建立起不同的交通小区，交通小区规划图如图 6 所示：

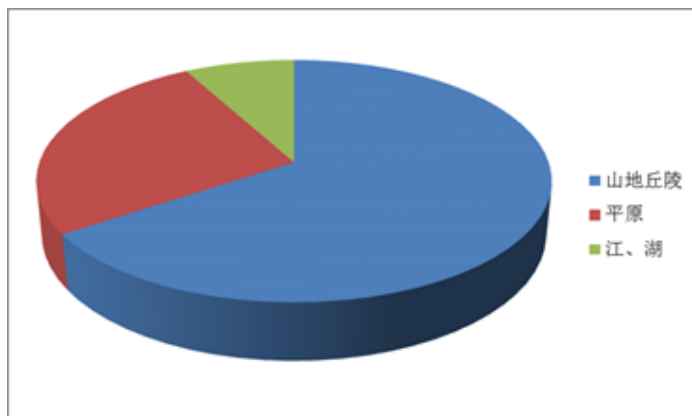


图 6. 杭州市土地面积构成

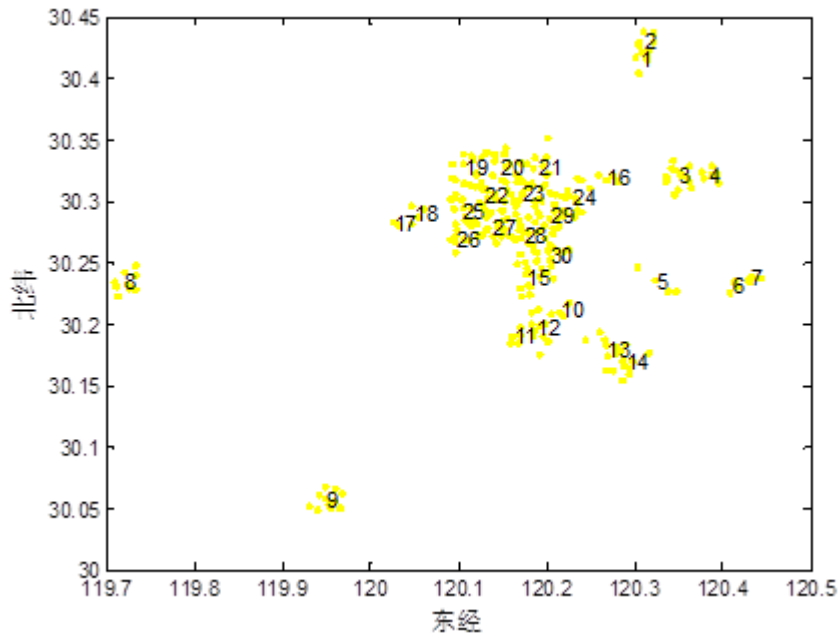


图 7. 交通小区规划图

Step 2 确立交通分析网络，将交通小区分为六个板块

由于杭州市是一个地貌类别多样的城市，素有“七山一水二分田”之说，所以我们在既考虑地形地貌的分布情况，又考虑出租车需求量的情况下，将杭州市分为 6 大板块，分别为：

$$\begin{aligned}
 A_1 &= [1, 2] \\
 A_2 &= [3, 4] \\
 A_3 &= [5, 6, 7] \\
 A_4 &= [8] \\
 A_5 &= [9] \\
 A_6 &= [10, 11, 12, 13, 14]; \\
 A_7 &= [15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30]
 \end{aligned} \tag{1}$$

Step 3 确立板块之间的最短路径

考虑到板块之间由于天然水道或者是山地阻隔之类因素的影响，所以我们在确立各个交通小区之间的最短距离时，首先应当解决板块之间的最短路程 A_{ij} ，并确认相应的边界点，具体如表 2 所示：

表 2 板块之间的最短路径

	A_1	A_2	A_3	A_4	A_5	A_6	A_7
A_1	/	2——1	7——1	8——1	9——1	10——1	16——1
A_2	1——2	/	5——3	8——3	9——3	10——3	16——3
A_3	1——7	3——5	/	8——5	9——5	10——5	30——5
A_4	1——8	3——8	5——8	/	9——8	11——8	17——8

A_5	1——9	3——9	5——9	8——9	/	11——9	15——9
A_6	1——10	3——10	5——14	8——11	9——14	/	15——10
A_7	1——16	3——16	5——30	8——17	9——15	10——15	/

Step 4 确立交通小区之间的最短路径

在确定交通小区之间的最短路径时，主要思想在于先求出板块内交通小区到边缘点之间的路程 dis_{nm} ，然后加上板块之间的最短距离 DIS_{nj} ：

$$DIS_{nj} = dis_{nm} + \Psi \times A_{ij} \quad (2)$$

n 表示出发地， j 表示目的地， i 表示边界点， Ψ 表示的是去往板块外的可能性，如果离开本板块，则 $\Psi=1$ ，否则为 0。

Step 5 构建属性区域网络加权模型：

在不考虑交通拥堵、道路宽窄的影响的前提下，我们依据上式所计算出的最短路径确认出 30 个区域的各自属性，建立起属性区域网络加权模型。

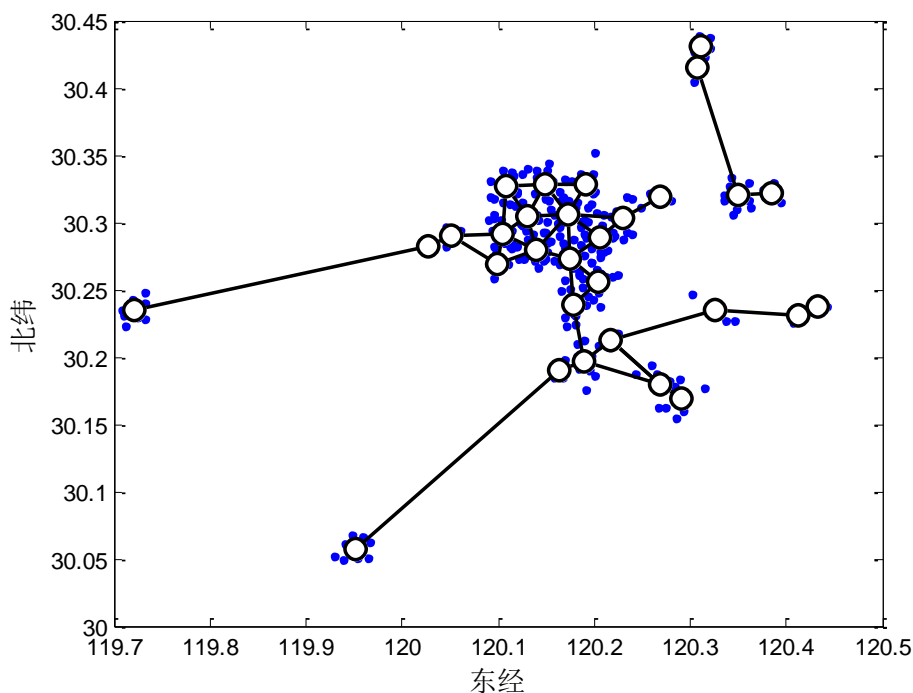


图 8. 交通小区的最短路径图

5.1.3 满足供需匹配的最少出租车需求量规划模型

依据已经构建的 30 个区域之间的交通联通网络，在整个打车运行系统相对稳定的状态下，建立规划模型，以各节点区域的出租车需求总量最少为目标，系统稳定为约束条件，通过求解该模型，可以获取每个节点“供需匹配”情况下的车辆需求信息。

(1) 基本思想

依据公交区域网络模型所设定的不同节点之间的路径,我们分别用 Or 和 De 表示出租车乘客的出发地和目的地,用 σ 表示 Or 与 De 之间的所有路径,模拟乘客的实际打车过程。同时,我们考虑在使用打车软件和未使用打车软件两种情况下,乘客和出租车司机的等待时间以及乘车需求量的差异,构造平衡方程组,以最小出租车需求量为目标,建立以多种指标为基础的时间需求量规划模型。在此基础上,改变最小等待时间的约束条件,得到不同的最小乘车需求量,建立二者之间的函数关系并绘制曲线。

(2) 等车价值衡量指标约束

本指标主要在于将等待时间转换成价值变量,将其量化,从而衡量出乘客的等车价值指标。

首先,考虑乘客在属性区域网络加权模型下的出租车需求量,用 D_{ij}^1, D_{ij}^2 来分别表示出在出行路径相同的情况下,选择采用软件打车和路边随机等待空车的两种不同方式的出租车需求量,二者的总和应当等于出租车需求量总和。即

$$D_{ij} = D_{ij}^1 + D_{ij}^2 \quad (3)$$

为了更好地对两种打车方式的费用做出衡量,我们引入等待时间价值的概念,即在原有打车路程费用的基础上,加入价值变量,将乘客在等待过程中产生的时间消耗纳入费用计算的范围之内,将等待时间转换为价值变量,具体计算公式如下:

$$P_{ij}^1 = q_{ij} + \alpha_1 T_{ij}^1 + \alpha_2 T_{ij} + \alpha_1 T_{ij}^2 \quad \forall (i, j) \in \sigma \quad (4)$$

$$P_{ij}^2 = q_{ij} + \alpha_1 T_i + \alpha_2 T_{ij} \quad \forall (i, j) \in \sigma \quad (5)$$

其中,(2)式 P_{ij}^1 表示在使用打车软件下乘客所花费的总费用,(3)式 P_{ij}^2 表示在未使用打车软件下乘客所花费的总费用。

(3) 打车软件使用率条件约束

一个乘客从出发地 $i \in Or$ 到目的地 $j \in De$ 的过程中,选择使用打车软件的概率可由下列计算公式表示:

$$\frac{D_{ij}^1}{D_{ij}^1 + D_{ij}^2} = \frac{\exp(-\phi P_{ij}^1)}{\exp(-\phi P_{ij}^2) + \exp(-\phi P_{ij}^1)} \quad \forall (i, j) \in \sigma \quad (6)$$

其中 ϕ 为一个非负的参数,反映了乘客的需求与出租车服务之间的不确定程度。

将公式(1)带入到公式(4)之中可以分别计算出在使用和未使用打车软件时的乘客需求量:

$$D_{ij}^1 = D_{ij} \frac{\exp\{-\phi(q_{ij} + \alpha_1 T_{ij}^1 + \alpha_2 T_{ij} + \alpha_1 T_{ij}^2)\}}{\exp\{-\phi(q_{ij} + \alpha_1 T_i + \alpha_2 T_{ij})\} + \exp\{-\phi(q_{ij} + \alpha_1 T_{ij}^1 + \alpha_2 T_{ij} + \alpha_1 T_{ij}^2)\}} \quad (7)$$

$$D_{ij}^2 = D_{ij} \frac{\exp\{-\phi(q_{ij} + \alpha_1 T_i + \alpha_2 T_{ij})\}}{\exp\{-\phi(q_{ij} + \alpha_1 T_i + \alpha_2 T_{ij})\} + \exp\{-\phi(q_{ij} + \alpha_1 T_{ij}^1 + \alpha_2 T_{ij} + \alpha_1 T_{ij}^2)\}} \quad (8)$$

(4) 出租车盈利条件约束

为保证出租车的基本盈利,我们需要对出租车单次出行所得的费用进行约束处理,通常基本盈利的经验公式如下:

$$RE = BP + MP \times M - CO \quad (9)$$

其中，基础价格 **BP** 与每公里单价 **MP** 是由政府定出，而出租车每趟运行成本 **CO** 则是由多种因素共同决定，包括燃油费 X_1 、汽车维护修理费用 X_2 、其他费用 X_3 等。

值得注意的是，随着出租车数量的增加，街道上寻找乘客的空载出租车的数量就会增加，将空载时间的增加转化为价值成本，故可在运行成本加入时间价值成本 $X_4(D_{ij})$ 。

综上所述，我们得到司机每单净赚费用的表达式：

$$W_{ij} = w_{ij} + r_{ij} * a - f(X_1 + X_2 + X_3 + X_4(D_{ij})) \geq \beta \quad \forall (i, j) \in \sigma \quad (10)$$

a 表示每公里的收费额， r_{ij} 表示从 i 区域到 j 区域的公里数， f 表示成本关于各项支出费用的函数关系。

值得注意的是，在此我们要求每单的净赚费用 W_{ij} 不小于某一个常数 β ，以保证出租车行业的正常运行。

(5) 空载条件约束

在考虑出租车的空载量时，一般情况下是通过研究出租车空驶率与出租车乘客最长等车时间之间的关系，利用高斯函数融入对历史数据的拟合，从而得到空驶率。

然而，需要注意的该种方法得到的结果是建立在“供需匹配平衡状态”下的，对于实际出租车资源分配情况并不具有很好的适用性，因此我们在考虑有无打车软件使用的因素，建立起一套全新的“空载时间”算法。

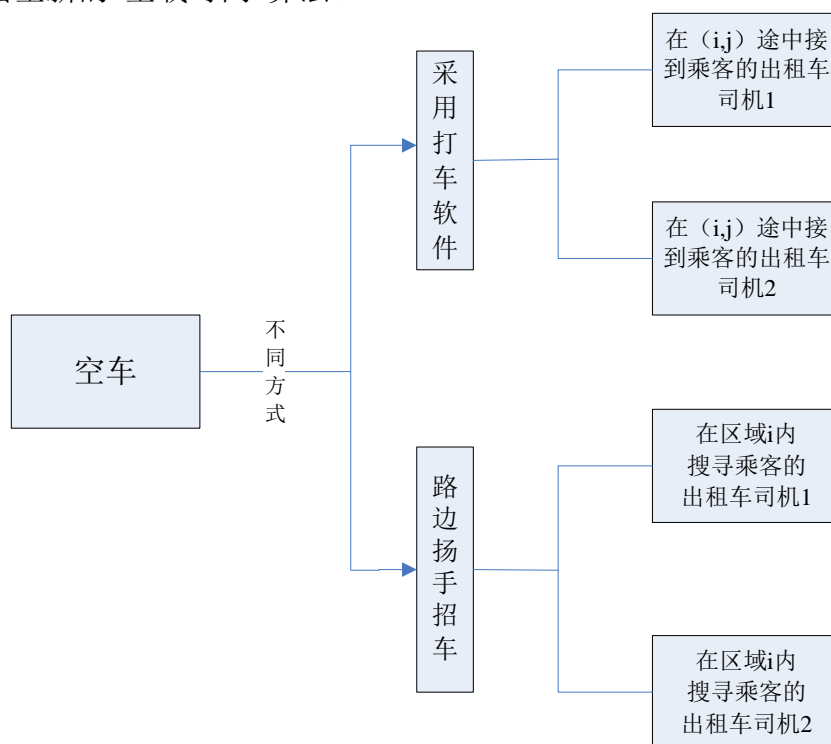


图 9. 使用和未使用打车软件的出租车搜索乘客方式的区别

考虑到在本交通网络当中，地理位置相对固定，所有出租车的载客数是不完全相同的，主要可划分为满载、载客（非满载）、空载三种情况，所以满载的出租车的数量为各个固定驾驶路线当中满载的出租车的数量总和。

然后，我们以出租车被占用的时间为基础，建立平衡方程组。为进一步提升模型的普遍性，将出租车被占用总时间分为满载时间，使用以及未使用打车软件所花费的到达搜索地点和实际搜索所花费的时间，具体计算公式为：

$$\sum_{ij \in \sigma} M_{ij} T_{ij} + \sum_{i \in Or, k \in De} N_{ki} (T_{ki} + t_i) + \sum_{k \in De, (i,j) \in \phi} N_{kij} (T_{ki} + t_{ij} + T_{ij}^2) = H \quad (11)$$

因此，我们可以就不同区域寻找乘客的空载出租车数量建立平衡方程：

$$\overline{N}_k^1 = \sum_{i \in Or} N_{ki} \quad \forall k \in De \quad (12)$$

$$\overline{N}_k^2 = \sum_{(i,j) \in \sigma} N_{kij} \quad \forall k \in De \quad (13)$$

$$\sum_{i, (i,k) \in \sigma} N_{ik} = \overline{N}_k^1 + \overline{N}_k^2 \quad \forall k \in De \quad (14)$$

(6) 目标函数

根据之前的约束条件和平衡方程，我们可以得到从 i 区域到 j 区域出租车需求量关于等待时间的函数，以此来对目标函数进行最优规划，得到在不同期望等待时间下各区域的出租车需求量。

目标函数为：

$$\min DEM = \sum_{ij \in \sigma} D_{ij}^1 + D_{ij}^2 \quad (15)$$

(7) 最终规划模型

最终的满足供需匹配的最少需求量规划模型主要为：

$$\begin{aligned} \min DEM &= \sum_{i,j \in \sigma} D_{ij}^1 + D_{ij}^2 \\ st \left\{ \begin{aligned} D_{ij}^1 &= D_{ij} \frac{\exp\{-\phi(q_{ij} + \alpha_1 T_{ij}^1 + \alpha_2 T_{ij} + \alpha_1 T_{ij}^2)\}}{\exp\{-\phi(q_{ij} + \alpha_1 T_i + \alpha_2 T_{ij})\} + \exp\{-\phi(q_{ij} + \alpha_1 T_{ij}^1 + \alpha_2 T_{ij} + \alpha_1 T_{ij}^2)\}} \\ D_{ij}^2 &= D_{ij} \frac{\exp\{-\phi(q_{ij} + \alpha_1 T_i + \alpha_2 T_{ij})\}}{\exp\{-\phi(q_{ij} + \alpha_1 T_i + \alpha_2 T_{ij})\} + \exp\{-\phi(q_{ij} + \alpha_1 T_{ij}^1 + \alpha_2 T_{ij} + \alpha_1 T_{ij}^2)\}} \\ W_{ij} &= w_{ij} + r_{ij} * a - f(X_1 + X_2 + X_3 + X_4(D_{ij})) \geq \beta \\ \sum_{i,j \in \sigma} M_{ij} T_{ij} + \sum_{i \in Or, k \in De} N_{ki} (T_{ki} + t_i) + \sum_{k \in De, (i,j) \in \phi} N_{kij} (T_{ki} + t_{ij} + T_{ij}^2) &= H \\ \overline{N}_k^1 &= \sum_{i \in Or} N_{ki} \quad \forall k \in De \\ \overline{N}_k^2 &= \sum_{(i,j) \in \sigma} N_{kij} \quad \forall k \in De \\ \sum_{i, (i,k) \in \sigma} N_{ik} &= \overline{N}_k^1 + \overline{N}_k^2 \quad \forall k \in De \end{aligned} \right. \quad (16) \end{aligned}$$

(8) 实际计算结果

根据以时间需求量为因变量的最少需求量规划模型，我们赋予等待时间不同的数值，分别计算此模型的最优解，结果如表 3 所示：

表 3 不同打车方式需求量与等车时间的关系

等车时间 (MIN)	需求量 (使用打车软件/辆)	需求量 (未使用打车软件/辆)
4	678	589
5	598	441
6	532	337
7	623	597
8	641	562
9	589	403
10	559	448

从表中最终数据我们可以看出，当预期等待时间为 6 min 时，该区域 1 min 所产生的出租车需求量最少，为 532 辆。于是我们取 6 min 为理想等待时间，取 532 辆为每分钟理想需求量。同理可得，我们可以基于上述规划模型来分析同一地点，不同时间的出租车资源“供求匹配”的程度，从而得出对出租车的最少需求量。

5.1.4 基于 Bonferroni 算子的“供需匹配度”测量模型

通过满足供需匹配的最少需求量规划模型，我们获得了每个网络节点的出租车最少需求量。进而，根据获取数据的相关属性，我们从乘客的期望需求出发，与实际数据相比较，分别从乘客等待时间、出租车需求量、出租车满载率、司机寻客时间这四个方面，构建比例指标来对出租车与打车需求的“供需匹配”程度进行测量。

(1) 等待时间因子：

$$\gamma_1 = \frac{a}{a_0} \quad (17)$$

其中，a 为理想等待时间、 a_0 为实际等待时间。

(2) 出租车需求因子：

$$\gamma_2 = \frac{b}{b_0} \quad (18)$$

其中，b 为实际出租车需求量、 b_0 为理想出租车需求量。

(3) 出租车满载率因子：

$$\gamma_3 = \frac{c}{c_0} \quad (19)$$

其中， c 为实际出租车满载率、 c_0 为理想出租车满载率。

(4) 司机搜寻时间因子：

$$\gamma_4 = \frac{d}{d_0} \quad (20)$$

其中， d 为理想实际搜寻时间、 d_0 为实际期待搜寻时间。

(2) 基于 Bonferroni 算子的供需匹配度指标

为确定实际交通状况的“供需匹配”程度，我们综合上述四个衡量因子，建立总供需匹配度函数 $\gamma = F(\gamma_1, \gamma_2, \gamma_3, \gamma_4)$ 。

首先，我们要计算出四个基本指标的权重，并假定 $w = (w_1, w_2, w_3, w_4)^T$ 为上述四个指标的权向量，其中 $w_i \in [0, 1]$ ($i = 1, 2, 3, 4$)，且 $\sum_{i=1}^4 w_i = 1$ ，通过层次分析法，通过构造判断矩阵求得最大特征向量，归一化后得到的四个指标的权重为依次为：

$$W = (0.1, 0.5, 0.2, 0.2) \quad (21)$$

进一步，利用 Bonferroni 平均算子对匹配程度的四个指标进行集成，得到匹配度指标，为

$$\gamma = B^{p,q}(\gamma_1', \gamma_2', \gamma_3', \gamma_4') = \left(\frac{1}{n} \frac{1}{n-1} \sum_{i,j=1, i \neq j}^n w_i w_j \gamma_i'^p \gamma_j'^q \right)^{\frac{1}{p+q}} \quad (22)$$

注意， $p, q \geq 0$ 。

以 γ 作为匹配度指标，则根据社会经验易得 γ 越趋近于 1，则说明匹配程度越高。

(3) 实际计算

我们以实际数据与理想数据的作为衡量标准，即 γ 越接近于 1 时，表示匹配程度越高，偏离程度越大，说明实际生活中的出租车运输存在着资源分配不合理的现象。

最终，我们得到的实际的匹配度为 $\gamma = 0.9790$ 。

从结果上我们可以看出，出租车的需求量极大程度的决定了实际的匹配度，并且呈现供不应求的现状同时，出租车的实际满载率、司机实际搜寻时间也低于理想值，导致了匹配度与理想值 1 之间的偏离。

在实际的资源配置之中，应优先考虑消费者对出租车资源的需求量，推行合理的改革政策，使“供求匹配率”有所提升。

5.1.5 不同时空下“供求匹配”的分析

(1) 空间下的“供求匹配度”分析

根据 5.1.2 中建立的区域属性网络加权模型，我们得到了以出租车需求量为基础的杭州市的区域划分，根据密集程度，我们主要将城市分为 CBD 区域、风景胜地以及周边郊区。我们从中可以看出，车辆密集地带主要在城市中心的繁华地段。同时，考虑到杭州的旅游资源，在以西湖为中心的城市地带，出租车的需求量也较大。利用公式 (22)

我们分别得出三类地区的供求匹配度，如表 4 所示：

表 4 三类地区的供求匹配度

	CBD 区域	风景胜地	周边郊区
供求匹配度	0.765	0.679	0.682

由求出的匹配度我们可以看出，CBD 地区和风景名胜地区的供求匹配度明显偏离理想值，说明这两处地点的明显供不应求，而周边郊区虽然乘车的需求量不大，但司机大部分时间不愿意进入郊区地带，所以匹配度也不高。

（2）不同时间下的“供求匹配度”分析

在得出了三个典型区域的“供应匹配度”的基础上，我们又分别求出了平常时段、高峰期、夜间时段三个不同时间段的供求匹配度，如表 5 所示：

表 5 三个不同时间段的供求匹配度

	平常时段	高峰期	夜间时段
CBD 区域	0.802	0.579	0.735
风景胜地	0.792	0.605	0.842
周边郊区	0.842	0.793	0.765

同时，我们计算出来杭州市一天 24 小时的“供求匹配度”变化曲线，如图 12 所示：

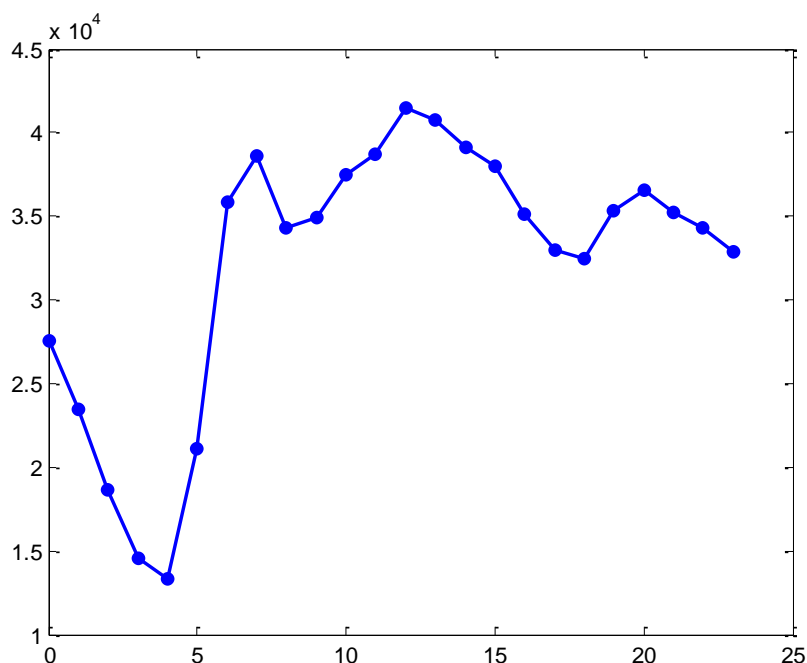


图 10. “供求匹配度”随时间变化图

由图表我们可以看出，在一天范围内，“供求匹配度”存在着较大的波动，从 5 时到 9 时，存在着明显的上升，这是由于，这段时间属于上班高峰期，虽然面临着交通拥堵的可能性，但大部分出租车司机会选择往返于住宅区和商业区之间；中午时分达到最高点，之后存在着明显的下降趋势，市民在下班之后可能会繁华的 CBD 区域进行休闲娱乐活动，受到私家车数量的影响，市区非常拥挤，是等待时间增长，匹配度下降。

5.2 问题二建模与解答

5.2.1 模型分析

首先，我们选取“滴滴打车”和“快的打车”这两家公司作为全国打车软件公司的代表，上网搜集了 2014 年这两家公司的打车补贴政策，具体如表 6 所示：

表 6 打车公司的补贴政策

滴滴打车		快的打车	
时间	补贴政策	时间	补贴政策
1 月 10 日	乘客车费立减 10 元， 司机立奖 10	1 月 20 日	乘客车费返现 10 元， 司机奖励 10 元
2 月 17 日	乘客返现 10-15 元， 新司机首单立奖 50 元	2 月 17 日	乘客返现 11 元， 司机返现 5-11 元
2 月 18 日	乘客返现 12-20 元	2 月 18 日	乘客返现 13 元
3 月 7 日	乘客每单减免随机 6-15 元	3 月 4 日	乘客返现 10 元/单， 司机端补贴不变
3 月 23 日	乘客返现 3-5 元	3 月 5 日	乘客补贴金额 5 元
5 月 17 日	乘客补贴“归零”	3 月 22 日	乘客返现 3-5 元
7 月 9 日	软件司机端补贴降为 2 元/单	5 月 17 日	乘客补贴“归零”
/	/	7 月 9 日	司机端补贴降至 2 元/ 单
8 月 9 日	全面取消司机端的现金补贴		

从上表显示结果来看，打车软件公司所出台的补贴政策主要可以分为两类：一类是给出租车司机使用打车软件给予补贴，一类是给使用打车软件的给予补贴。基于现有的市场行情，我们建立出如下分析模型：

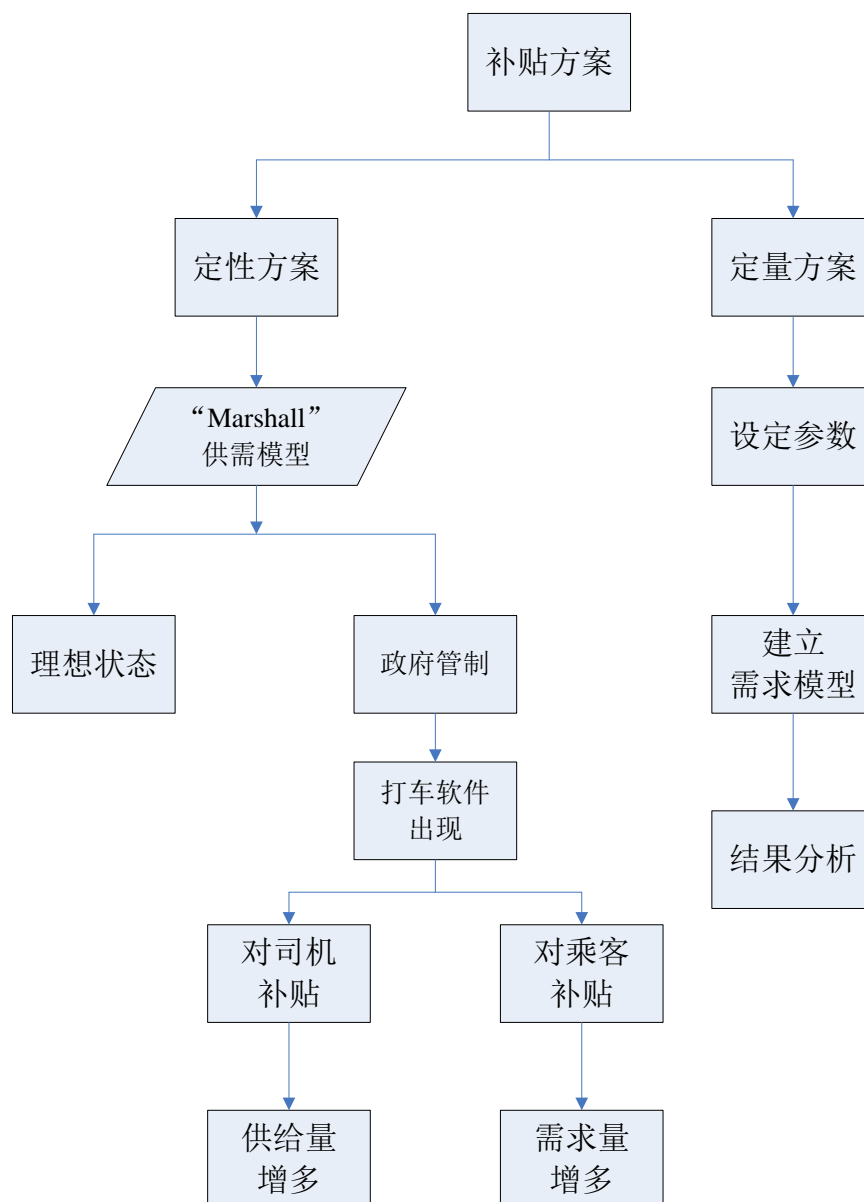


图 11.模型分析总体方案

5.2.2 定性分析

Step 1 分析打车软件出现前，出租车市场的简单模型

在本问题当中，需要各公司的出租车补贴方案对“缓解打车难”这一现象是否有帮助，所以在分析本问题之前，分析在打车软件出现之前的市场经济结构是很有必要的，通过打车软件出现前后的出租车的资源匹配程度，来判断是否解决了“打车难”的问题。

结合数学模型理论和微观经济学理论，可以将出租车看作一种服务性质的特殊商品，则其一般市场规律可以用“Marshall 剪刀”供需模型表示，如图 12 所示：

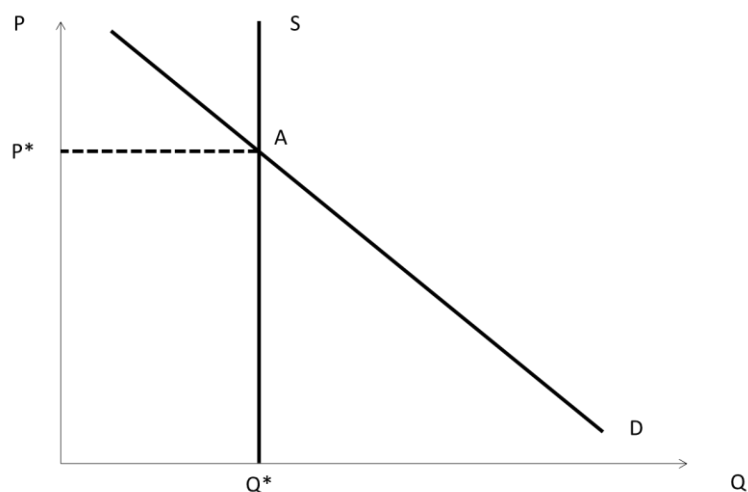


图 12. 理想状态下的“Marshall 剪刀”供需模型图

图中纵坐标 P 表示价格，横坐标 Q 表示数量，曲线 D 表示乘客的需求函数。由图中曲线可以反映人们对商品的需求量与价格 P 成反方向变化关系。即

$$Q = \alpha - kP \quad (\alpha > 0, 0 < k < 1)$$

而图中的函数 S 表示出租车的供给函数，之所以是一条竖线，是因为在此处将出租车本身看作一种特殊的商品，它的投放量是受到限制的，由政府统一规划决定，因此供给量与价格的变化无关。

根据供求理论，如果服务价格是可以自由谈价的，那么市场会在供给曲线和需求曲线的交点 A 处达到均衡稳定的状态，此时市场上的出租车价格是 P^* 。这种价格机制理论被称为可以自动调节市场价格的“看不见的手”。

如果仅仅是这样的情况，供给和需求就会相等，不会出现“打车难”现象。然而事实上，出租车行业的车辆供给量和运营都是被政府严格限制的，因此就出现了如下图的情形：

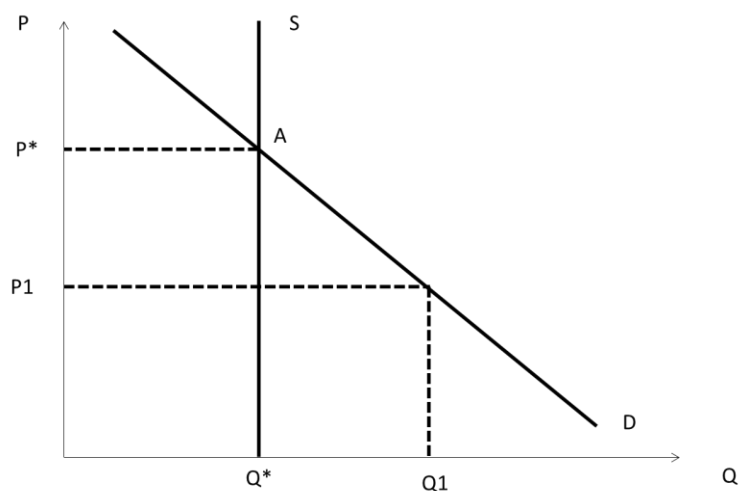


图 13. 政府管制下的“Marshall 剪刀”供需模型图

图中， P_1 为政府制定的相对于 P^* 较低的价格，因此乘客在该出租车价格水平下，对出租车服务的需求量为 Q_1 。当某一特定时间和空间下出现“打车难”现象，说明该条件下的出租车供给小于需求。即： $Q^* < Q_1$ 。但需要注意的是，此时的 Q^* ，即出租车的供给量保持不变， Q_1 表示城市的出租车需求量，需求大于供给，这就解释了为什么会出现“打车难”。

Step 2 分析打车软件出现后，出租车市场的模型：

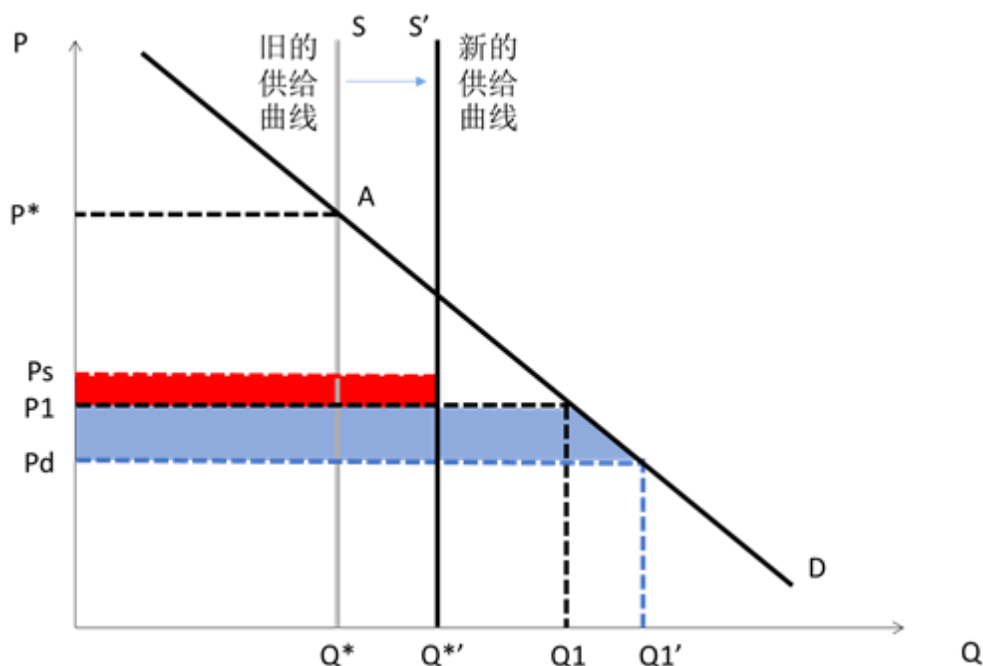


图 14. 政府管制下的“Marshall 剪刀”供需模型图

图中 P_s 表示司机在接受打车补贴后每单的收入单价， P_d 表示乘客在打车补贴后所需支付的价格，阴影部分的积分就是打车软件对乘客和司机的补贴总和，红色部分表示对司机的补贴额，蓝色部分表示对乘客的补贴额。

不难从图上看，打车软件实现了一定程度上调控了出租车行业的交易价格，而且将乘客支付的价格 P_d 与司机收入的价格 P_s 分开，承担了这一二者的差价，即所谓的补贴。

一方面，对于司机来说，打车软件对每单提供一定数量的金额补贴，本质上是提高了司机每单的收入单价，从而使得加入打车软件公司的出租车司机人数增多，使得供给曲线整体向右平移，由 Q^* 增加到 $Q^{*'}$ ，在需求量不变的情况下，加剧了“打车难”问题。

而另一方面，对于乘客来说，打车软件给予乘客优惠和返还措施，本质上是降低了乘客打车所需支付的价格，从而使得愿意使用打车软件叫车的人增多，使得需求量增加从 Q_1 增加到 Q_1' ，在供给量不变的情况下，加剧了“打车难”难题。

可见，打车补贴政策既能带来增大需求量的负面效应，也能带来提高供给量的正面效应，那么最终能否缓解“打车难”问题，就取决于正面效应与负面效应绝对值的相对大小。即：

$$\Delta E = (Q^{*'} - Q^*) - (Q_1' - Q_1) \quad (23)$$

其中， ΔE 为该补贴方案对于“缓解打车难”问题的正负效应差。因此，当 $\Delta E = 0$ 时，该方案正负效应抵消，如果 $\Delta E > 0$ ，则打车软件公司的补贴方案对“缓解打车难”有帮助，反之亦然。

5.2.2 定量分析：

设定需求函数形式：

$$Q = \alpha - kP \quad (24)$$

将图线上两点带入：

$$\begin{cases} Q_1 = \alpha - kP_1 \\ Q_1' = \alpha - kP_d \end{cases} \quad (25)$$

其中 $\alpha > 0$ ， $0 < k < 1$

Q_1 表示未使用打车软件时的需求量， Q_1' 表示使用打车软件时的需求量；

P_1 表示未使用打车软件时出租车价格水平， P_d 表示使用打车软件时出租车价格水平；

再由微观经济学理论得出约束条件为：

$$Q^* = n * \frac{time}{T_{ij} + T_i} \quad (26)$$

$$Q^{*'} = n * \frac{time}{T_{ij}^1 + T_{ij}^2 + T_{ij}} \quad (27)$$

其中，

Q^* 表示未使用打车软件时的供给量， $Q^{*'}$ 表示使用打车软件时的供给量；

$time$ 表示一辆车一天的工作时间， n 表示一个城市所拥有的出租车数量；

将 (22)、(23)、(24) 式与计算式 $\Delta E = (Q^{*'} - Q^*) - (Q_1' - Q_1)$ 联立，可以解出 ΔE 。

5.2.3 模型求解

根据问题一中需求量最小模型，我们计算出当期望等待时间为 6min 时，模型达到最优解，故以 6min 时的需求量为基础，求解出需求函数的表达式以及 ΔE 。

从已建立的满足供需匹配的最少需求量规划模型，以及网络中我们获取到下列数据：

表 7 打车公司的补贴政策

变量	数据
n / 辆	8923
$time$ / H	8
Q_1	337
Q_1'	532
P_1	246
P_d	489

将获得的数据带入到上述公式可得，我们得到了 $\Delta E = 0.325 > 0$ ，由此可以分析出，打车软件的补贴政策确实对“缓解打车难”有帮助。

事实上，从实际情况定性考虑，打车软件的补贴政策一方面会提高乘客乘坐出租车的积极性，使出租车更加供不应求，但更为重要的是，软件的出现建立了乘客与司机之间的直接联系，包括信任度和依赖性，极大程度上减少了双方的时间消费，提高了该城市的交通资源配置效率，最终缓解了“打车难”的困境。

5.3 问题三建模与解答

5.3.1 吸收客户的理性选择模型

从模型二的分析思路我们可以看出，想要解决打车难的问题，我们需要从多个角度分析各种解决方案。

出租车作为一种交通方式，有着自身的定位。如果我们盲目地增加出租车的数目，势必会导致公共交通秩序的混乱。

在模型二得出了打车软件在一定程度上缓解了“打车难”的困境的基础上，我们希望推出一套更加创新的软件补贴方案，通过改变补贴数额，使正负效应差达到最大值。

(1) 基础模型

Step1

确定预算约束线

$$\frac{X}{\lambda_1} + \frac{Y}{\lambda_2} = I \quad (28)$$

X 表示打车软件使用人数

I 表示人均预算最大值，为常量

Step 2

柯布-道格拉斯效用函数

$$u(x_1, x_2) = x_1^c x_2^d \quad (29)$$

其中 c, d 均为表示消费者偏好的正参数，且 $c + d = 1$ 。

Step 3

边际替代率

$$MRS = -\frac{\partial u(x_1, x_2) / \partial x_1}{\partial u(x_1, x_2) / \partial x_2} = -\frac{cx_1^{c-1}x_2^d}{dx_1^c x_2^{d-1}} = -\frac{cx_2}{dx_1} \quad (30)$$

最优化方程：

$$MRS = -\frac{\lambda_1}{\lambda_2} \quad (31)$$

根据最优化条件，可以得出最优解 x_1^* ， x_2^* 。

得到吸收率：

$$\rho = \frac{x_1^*}{x_1^* + x_2^*} \quad (32)$$

(2) 将上述基础模型分别应用于以下三种模式：

模式一：从本打算采用其他出行方式的消费者集合中吸引客户

λ_1 表示打车成本（价格）

λ_2 表示其他出行方式成本（价格）

$$\lambda_1 = P_1 - \Delta P_d \quad (33)$$

λ_2 为一个常量，且为外生变量

ΔP_d 为自变量

模式二：从采用传统打车方式的消费者集合中吸引客户

λ_1 表示打车成本（价格）

λ_2 表示传统打车成本（价格）

$$\lambda_1 = P_1 - \Delta P_d \quad (34)$$

$$\lambda_2 = p_1 \quad (35)$$

ΔP_d 为自变量

模式三：从采用传统打车方式的司机集合中吸引客户

λ_1 表示打车成本（价格）

λ_2 表示传统打车成本（价格）

$$\lambda_1 = P_1 + \Delta P_s \quad (36)$$

$$\lambda_2 = p_1 \quad (37)$$

ΔP_s 为自变量

表 8 三种模式汇总

	模式一	模式二	模式三
λ_1	$\lambda_1 = P_1 - \Delta P_d$	$\lambda_1 = P_1 - \Delta P_d$	$\lambda_1 = P_1 + \Delta P_s$
λ_2	$\lambda_2 = c$	$\lambda_2 = p_1$	$\lambda_2 = p_1$
$\rho_i (i=1,2,3)$	23.7%	35.3%	87.2%

5.3.2 在满足乘客需求条件下的正负效应差最大规划模型

在问题一中，我们建立了多种约束条件下的需求量最小模型。此处，我们仍然利用上述约束条件，但将目标函数改为计算正负效应差 ΔE 的最大值。

(1) 变量修正

在上述吸收客户的理性选择模型之中，我们确定了 3 个模型下受众群体的被吸收率，因此，需要对问题一中“满足打车需求的最少需求量规划模型”中的变量作出修正。

Step 1

设每天的城市出行人数为 C_{ij} ，对使用打车软件的乘客需求量 D_{ij}^1 进行重新的赋值。

$$D_{ij}^1 = D_{ij}^1 + \frac{C_{ij}}{30} \quad (38)$$

Step 2

对使用打车软件的乘客需求量 D_{ij}^1 、未使用打车软件的乘客需求量 D_{ij}^2 进行重新赋值。

$$D_{ij}^1 = D_{ij}^1 + D_{ij}^2 * \rho_1 \quad (39)$$

$$D_{ij}^2 = D_{ij}^2 - D_{ij}^2 * \rho_1 \quad (40)$$

Step 3

对未使用打车软件的出租车司机数为 \overline{N}_k^1 ，使用打车软件的出租车司机数为 \overline{N}_k^2 进行重新赋值。

$$\text{令: } \overline{N}_k^1 = \overline{N}_k^1 - \overline{N}_k^1 * \rho_3 \quad (41)$$

$$\text{令: } \overline{N}_k^2 = \overline{N}_k^2 + \overline{N}_k^1 * \rho_3 \quad (42)$$

(2) 新增约束条件

我们在问题一模型的约束条件的基础上，加入由问题二模型中得到的新的约束条件

Step 1

由“Marshall 剪刀”供需模型得到的约束条件：

$$Q = \alpha - kP \quad (43)$$

$$Q^* = n * \frac{time}{T_{ij} + T_i} \quad (44)$$

$$Q^{*'} = n * \frac{time}{T_{ij}^1 + T_{ij}^2 + T_{ij}} \quad (45)$$

Step 2

由于打车软件进行补贴，乘客所花费的总费用应该减去得到的补贴 ΔP_d ，费用表达式变为：

$$P_{ij}^1 = q_{ij} + \alpha_1 T_{ij}^1 + \alpha_2 T_{ij} + \alpha_1 T_{ij}^2 - \Delta P_d \quad (46)$$

未使用打车软件的乘客的总花费仍为：

$$P_{ij}^2 = q_{ij} + \alpha_1 T_i + \alpha_2 T_{ij} \quad (47)$$

Step 3

对于出租车司机而言，其每单净赚的费用表达式应该考虑得到的补贴 ΔP_s ，故约束条件应该变为：

$$W_{ij} = w_{ij} + r_{ij} * a - f(X_1 + X_2 + X_3 + X_4(D_{ij})) + \Delta P_s \geq \beta \quad (48)$$

(3) 最终规划模型:

$$\max XY = \Delta E$$

$$st \begin{cases} Q = \alpha - kP \\ Q^* = n^* \frac{time}{T_{ij} + T_i} \\ Q^{*'} = n^* \frac{time}{T_{ij}^1 + T_{ij}^2 + T_{ij}} \\ P_{ij}^1 = q_{ij} + \alpha_1 T_{ij}^1 + \alpha_2 T_{ij} + \alpha_1 T_{ij}^2 - \Delta P_d \\ P_{ij}^2 = q_{ij} + \alpha_1 T_i + \alpha_2 T_{ij} \\ W_{ij} = w_{ij} + r_{ij}^* a - f(X_1 + X_2 + X_3 + X_4(D_{ij})) + \Delta P_s \geq \beta \\ \sum_{i,j \in \sigma} M_{ij} T_{ij} + \sum_{i \in Or, k \in De} N_{ki} (T_{ki} + t_i) + \sum_{k \in De, (i,j) \in \phi} N_{kij} (T_{ki} + t_{ij} + T_{ij}^2) = H \\ \overline{N}_k^1 = \sum_{i \in Or} N_{ki} \quad \forall k \in D \\ \overline{N}_k^2 = \sum_{(i,j) \in \sigma} N_{kij} \quad \forall k \in De \\ \sum_{i, (i,k) \in \sigma} N_{ik} = \overline{N}_k^1 + \overline{N}_k^2 \quad \forall k \in De \end{cases}$$

(4) 新型补贴方案

根据不同时段的原始数据，分别利用上述规划模型，在 ΔE 取最大值取得最优解，为具体说明补贴方案，在此我们以总数为最大限额 10 元的补贴为例，研究各个时间段乘客和出租车司机分得的补贴的比例，最优方案具体如下表：

表 9 补贴方案表

	平常时段	高峰时段	夜间时段
乘客	4 元	5 元	1 元
出租车司机	6 元	5 元	3 元

(5) 方案合理性叙述

本模型以“Marshall 剪刀”供需模型为基础，建立正负效应差 ΔE ，因为 ΔE 反应了补贴策略所带来的正效应和负效应的差异，当 $\Delta E > 0$ 时，即反映了补贴策略是有帮助的。我们利用满足乘客需求条件下的正负效应差最大规划模型求得最大的 ΔE ，已满足有帮

助的前提条件，合理性可以得到论证。

六、模型的总结、评价与优化

6.1 模型的总结

6.1.1.对于问题一，共建立了以下 3 个模型来分析不同时空下的供需匹配程度：

（1）在基于“watershed”算法的区域属性加权网络模型当中，我们将杭州市各经纬度及相对应的打车需求量通过“watershed”算法将杭州市进行聚类划分成 30 个交通小区，并且通过确立交通小区的最短路径来判定出各个区域的属性，从而构建出属性区域网加权网络；

（2）供需匹配平衡下的最少需求量规划模型，主要是基于区域网络模型，建立起以各个区域内出租车需求量最少为目标函数的规划模型，以获取各个节点的车辆需求量。

（3）在基于 Bonferroni 算子的“供需匹配度”评价模型中，我们从乘客等待时间、出租车需求量、出租车满载量、司机寻客时间四个方面构建指标因子，进而得出匹配度指标函数 γ ，利用这一指标来衡量出供求的匹配程度。

6.1.2 对于问题二，建立“Marshall 剪刀”模型进行定性和定量地分析：

在本问题的求解过程中，我们结合了经济学理论，构建出“Marshall 剪刀”模型对滴滴打车和快的打车两家公司的补贴政策进行了定性且定量的分析，构造出需求量的规划模型，最终将补贴政策的影响定量化为 ΔE ，最终得出结论：补贴政策有助于缓解“打的难”问题。

6.2 模型评价

优点	缺点
<p>（1）模型解决角度多样，具体问题具体分析，为每一个问题都设计了相应的解决方案。</p> <p>（2）模型中充分考虑了多个影响因素，对于有无打车软件情况下的市场情况都进行了充分的挖掘，使得模型具有更好的合理性和现实意义。</p> <p>（3）在解决问题二时，创造性地结合经济学理论知识和数学建模，构造出了“Marshall 剪刀”模型，其各项理论不仅有严谨的数学证明，更经过了长期社会实践的检验，保证了模型的合理性和科学性。</p>	<p>（1）由于时间有限，搜索渠道受到限制，所以采样的数据缺乏一定的普遍性和完备性。</p> <p>（2）模型的匹配准确率对于部分相关参数灵敏度过高，因此需要在应用于实际需要进行一些模型假设。</p>

6.3 模型优化

(1) 进一步挖掘数据,包括杭州市在连续一个月每天 24 个小时的出租车需求量、采用滴滴打车软件的乘客数量、采用路边招手招车的乘客数量等等、通过对大数据进行整体分析对原模型的参数进行进一步地优化;

(2) 根据对相应参数灵敏度的不同,对模型的假设条件进一步地细化和标准化,并且尝试引入 BP 神经网络算法等智能算法,使得模型具备更好的适用性。

七、参考文献

- [1]刘焕章,斐道武。H-OWA 算子及其在多属性决策中的应用[J],浙江理工大学学报,第 29 卷第 1 期,第 2 页,2012
- [2]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,第 4 期第 1 卷,第 3 页,2014
- [3]周晶,何健敏。城市出租车运营系统的随机分析[J]。管理工程学报,第 1 期,第 2 页,2001
- [4]李旭宏,田锋,顾政华。城市道路网供求分析技术[J]。交通运输工程学报,2001,第 2 期第 2 期,第 3 页,2001

八、附录

附录一:提取有效数据的代码

```
clear all
close all
clc
A='
["",120.1732,30.2858,353],["",120.1745,30.2795,331],["",120.1692,30.2646,3
29],["",120.1719,30.287,317],["",120.1776,30.2584,295],["",120.1273,30.2785
,290],["",121.3281,31.2043,286],["",120.1796,30.2484,277],["",120.1659,30.3
01,275],["",120.1584,30.2967,272],["",120.17,30.2741,270],["",120.1388,30.2
728,248],["",120.1414,30.2792,246],["",120.1487,30.271,233],["",120.1569,30
.302,231],["",120.1639,30.2834,227],["",120.1609,30.2744,218],["",120.1406,
30.2961,214],["",121.3235,31.2013,212],["",120.1574,30.2696,205],["",120.17
72,30.2936,205],["",120.1531,30.295,204],["",120.1763,30.2926,203],["",120.
1622,30.2708,197],["",120.2049,30.2907,196],["",120.1731,30.3009,196],["",1
21.3627,31.1989,191],["",120.1824,30.3054,190],["",120.1644,30.2875,187],["
",120.1262,30.2909,184],["",120.1879,30.2897,179],["",120.1286,30.2968,179]
,["",120.1581,30.3093,177],["",120.125,30.2742,175],["",120.1404,30.2912,17
5],["",120.1967,30.1992,175],["",120.1162,30.2791,174],["",120.1997,30.2766
,173],["",120.1721,30.2534,172],["",120.1128,30.2815,167],["",120.1602,30.2
957,166],["",118.8066,31.9794,166],["",120.1539,30.2871,165],["",120.1958,3
0.3101,163],["",120.103,30.3075,162],["",120.1891,30.2553,162],["",120.2038
,30.2771,160],["",120.2127,30.2672,160],["",120.2016,30.1991,157],["",120.1
856,30.2612,155],["",120.1428,30.2755,155],["",120.1703,30.242,155],["",120
```

.1128,30.2844,152],[",120.1471,30.2676,151],[",120.2111,30.2464,151],[",120.1023,30.2981,150],[",120.1801,30.2203,149],[",120.2027,30.3023,148],[",120.2114,30.2711,146],[",120.1465,30.2774,146],[",120.1294,30.2942,143],[",120.125,30.3074,143],[",120.1024,30.2809,142],[",120.2214,30.2934,142],[",120.1481,30.3109,142],[",120.2014,30.254,142],[",120.178,30.3127,138],[",120.1026,30.2722,136],[",120.1182,30.2958,136],[",120.1551,30.3101,136],[",120.1986,30.194,135],[",120.1981,30.2881,135],[",120.1114,30.3037,135],[",120.1945,30.2861,133],[",120.2067,30.3071,132],[",120.2215,30.2882,129],[",120.2689,30.1782,129],[",120.2761,30.1917,128],[",120.3486,30.3198,128],[",120.1236,30.2816,126],[",120.1818,30.3008,124],[",120.0987,30.2827,124],[",120.2723,30.1698,124],[",120.1368,30.3137,123],[",120.1206,30.2928,123],[",120.1865,30.2778,123],[",120.1891,30.3092,122],[",120.1498,30.2976,122],[",120.1221,30.2852,122],[",120.2059,30.2495,122],[",120.4346,30.2424,122],[",120.1796,30.2037,121],[",120.135,30.3121,120],[",120.2179,30.2857,119],[",120.1835,30.2156,118],[",120.2672,30.1768,118],[",120.1981,30.2789,118],[",120.2043,30.2641,116],[",120.1685,30.3347,114],[",120.1417,30.3079,113],[",120.1053,30.2955,113],[",120.2779,30.171,112],[",120.1367,30.2949,111],[",120.1818,30.3344,110],[",120.443,30.2461,109],[",120.1887,30.293,109],[",120.1893,30.2855,109],[",120.1129,30.3134,109],[",120.3098,30.4195,108],[",120.1575,30.3138,108],[",120.2663,30.1875,108],[",120.2002,30.267,106],[",120.3022,30.4126,106],[",120.1925,30.2642,105],[",120.1761,30.1911,105],[",120.2211,30.265,105],[",120.1815,30.2754,104],[",120.3014,30.4085,103],[",120.0952,30.2792,103],[",120.1744,30.2378,102],[",120.1738,30.2313,102],[",120.2024,30.2764,102],[",120.3438,30.3144,102],[",120.2013,30.282,102],[",120.1941,30.3524,101],[",120.3334,30.3178,101],[",120.0298,30.2866,99],[",120.3406,30.3222,99],[",120.2192,30.2691,99],[",120.1002,30.3101,99],[",120.1827,30.2408,99],[",120.1758,30.1827,98],[",120.3119,30.4327,98],[",120.1847,30.2439,97],[",120.3374,30.3093,97],[",120.2173,30.2817,97],[",120.3155,30.4189,96],[",120.1156,30.3194,96],[",120.2789,30.1659,95],[",120.1577,30.3357,95],[",120.2124,30.2088,94],[",120.2051,30.2467,94],[",120.1009,30.3201,93],[",120.3183,30.4201,92],[",119.7221,30.2288,91],[",120.1985,30.3407,91],[",120.0968,30.316,91],[",120.3167,30.4295,90],[",120.1744,30.2265,90],[",120.1901,30.2281,90],[",120.222,30.2037,90],[",120.198,30.3089,90],[",120.1465,30.3275,90],[",120.1883,30.1991,89],[",120.1326,30.3136,89],[",120.1529,30.3224,88],[",120.221,30.2189,88],[",120.3638,30.312,87],[",120.3006,30.4323,87],[",120.181,30.3158,87],[",120.1997,30.2421,86],[",119.9507,30.0704,86],[",120.1956,30.3144,86],[",120.2185,30.28,85],[",120.1885,30.2507,85],[",119.943,30.0589,84],[",120.2768,30.159,84],[",120.188,30.2598,84],[",120.0919,30.2682,84],[",120.1718,30.3307,84],[",120.1879,30.2355,84],[",119.7336,30.2367,83],[",120.2871,30.1776,83],[",119.9552,30.0558,83],[",119.7269,30.2337,83],[",120.1176,30.3203,83],[",120.1957,30.186,83],[",120.1833,30.1935,83],[",120.2157,30.2575,83],[",120.2222,30.3138,82],[",120.2713,30.321,82],[",120.211,30.2004,82],[",120.2035,30.3061,82],[",120.2892,30.1656,82],[",120.1729,30.3111,81],[",120.2937,30.1685,81],[",120.177,30.3257,81],[",120.1611,30.311,80],[",120.201,30.2445,80],[",120.1236,30.3291,80],[",120.0954,30.2712,80],[",120.1941,30.3304,79],[",120.1647,30.184,79],[",120.2867,30.1696,78],[",130.2879,47.3299,78],[",120.2348,30.3141,78],[",120.193,30.3296,78],[",119.9512,30.0556,76],[",120.4305,30.2263,76],[",120.4184,30.2297,75],[",120.1948,30.3178,75],[",120.3053,30.4347,75],[",120.1478,30.1838,75],[",119.7067,30.2355,75],[",120.4004,30.2293,75],[",119.9438,30.0658,74],[",120.101,30.3326,73],[",120.2784,30.3173,73],[",120.1665,30.2518,73],[",120.1363,30.3318,73],[",120.4153,30.2334,73],[",117.016,36.6651,72],[",120.4241,30.2397,72],[",119.9405,30.0572,72],[",120.1332,30.2681,72],[",120.2691,30.194,72],[",120.2802,30.1751,71],[",120.1953,30.2481,71],[",120.0598,30.2495,71],[",120.2289,30.2951,71],[",119.7073,30.2324,71],[",120.1343,30.3148,70],[",120.115,30.3089,70],[",120.1666,30.3139,70],[",120.1851,30.3464,69],[",120.2643,30.1776,68],[",119.9683,30.0509,68],[",119.951,30.0525,68],[",120.2295,30.2925,67],[",120.335,30.3298,67],[",120.2234,30.3042,67],[",120.2843,30.1922,66],[",119.0518,29.6131,66],[",120.0853,30.2895

```
,66],[",120.1103,30.3273,65],[",120.1022,30.2965,65],[",119.9424,30.0505,65],[",120.3455,30.3205,65],[",120.1551,30.188,65],[",120.3218,30.407,65],[",119.714,30.2217,65],[",120.4507,30.2396,64],[",120.1243,30.3359,64],[",120.2411,30.186,64],[",120.2679,30.1991,64],[",120.0556,30.2486,64],[",120.2364,30.3186,64],[",120.182,30.2991,64],[",120.1847,30.3431,64],[",120.212,30.2471,63],[",120.2479,30.3095,63],[",119.0687,29.6119,63],[",120.1928,30.186,63],[",120.3646,30.3123,63],[",120.2001,30.3574,62],[",120.2015,30.3454,62],[",120.1012,30.3087,62],[",120.1677,30.1773,62],[",120.2734,30.3121,62],[",119.0529,29.6132,62],[",120.2384,30.2935,61],[",120.4493,30.2414,61],[",120.0616,30.2498,61],[",119.0692,29.6128,61],[",119.7286,30.2474,61],[",120.1297,30.3274,61],[",120.1402,30.257,61],[",120.2118,30.1899,60],[",120.2668,30.3102,60],[",120.2663,30.1987,60],[",120.1543,30.179,60],[",120.191,30.1805,60],[",120.0693,30.298,59],[",120.122,30.3067,59],[",120.1887,30.3304,59],[",120.2966,30.1596,59],[",120.1406,30.3343,59],[",120.3107,30.4061,59],[",119.0477,29.6055,59],[",120.2688,30.2128,59],[",120.2724,30.2136,59],[",120.1881,30.2088,59],[",120.1978,30.2568,58],[",120.2899,30.1646,58],[",120.2676,30.3091,58],[",120.1695,30.1763,58],[",108.9553,34.2645,57],[",120.2327,30.1927,57],[",120.1674,30.3206,57],[",120.2155,30.2342,57],[",120.2191,30.2582,57],[",120.2013,30.1938,56],[",120.1776,30.1831,56],[",120.111,30.3328,56],[",120.1577,30.3375,55],[",126.5844,45.7145,55],[",119.9631,30.0659,55],[",120.2547,30.1886,55],[",120.0962,30.3358,55],[",120.1688,30.1927,55],[",120.1017,30.3093,54]]';
B=find(A==' ','');
l=1;
for i=1:size(B,2)
    if i/4~=fix(i/4)
        for u=2:9
            if A(B(i)+u)==' '|| A(B(i)+u)==' '
                E(l)=str2double(A(B(i)+1:B(i)+u-1));l=l+1;break
            end
        end
    end
end
M=reshape(E',3,size(E,2)/3);
N=M'
```

附录二：求一天内供求匹配度随时间变化曲线的代码

```
clear all
close all
clc
A=[27530 23432 18625 14579 13365 21126 35827 38568 34320 34945 37496 38747 41516
40758 39146 38000 35107 33000 32466 35289 36557 35205 34295 32902];
X=0:23;
Z=ones(24,1);
plot(X,A,'b',...
'LineWidth',2,...
'MarkerSize',10,...
'MarkerEdgeColor','k',...
'MarkerFaceColor','w')
hold on
plot(X,A,'ko',...
'LineWidth',3,...
'MarkerSize',2,...
'MarkerEdgeColor','b',...
'MarkerFaceColor','w')
```

附录三：求打车需求信息节点坐标图的代码

```
clear all
close all
clc
x=[121.3242 120.2194 121.3248 120.169 120.173 120.1749 120.1733 120.1688
120.1686 130.2911 120.197 120.2072 120.179 120.1424 120.164
120.1375 120.1314 120.18 120.1621 118.811 120.1469 120.0257
120.1589 120.1678 120.3477 120.1366 120.2191 120.1021
120.1728 120.1887 120.1484 120.1976 120.1878 120.1682
120.1672 120.1289 120.1783 120.1635 120.155 120.1326 120.3517
120.2026 120.2331 120.1913 120.152 120.143 120.179 120.3513
120.1226 120.1535 120.2082 120.1193 120.1308 120.1485
120.1037 120.2688 120.1312 120.1298 120.1462 120.2882
120.1634 120.2224 120.125 120.2164 120.2955 120.1195 120.3607
120.1572 120.191 120.1691 120.1122 120.2014 120.1811 120.1866
120.1563 120.2329 120.0959 120.1861 120.1498 120.0948
120.1106 120.2089 120.186 120.1716 120.1924 120.1496 120.1414
120.3574 120.278 120.1421 120.15 120.3758 120.2226 120.1928
119.733 119.7252 120.3007 120.2035 120.2064 120.1512 120.1525
120.386 120.2167 120.4415 119.7228 120.3131 119.9484 120.195
121.3561 120.2774 120.1242 120.1772 120.1193 120.1957
120.1038 120.1976 120.1056 120.2129 120.2735 120.3206
120.1986 120.0966 120.3356 120.1825 120.3194 120.4334
120.2013 120.3039 120.3146 120.1138 120.0961 120.2037
120.1165 119.9589 120.1836 120.182 120.3941 120.1141 120.1864
120.2017 120.2666 120.1874 120.2062 120.2317 120.1389
120.2784 120.3467 120.3082 120.1635 119.957 120.1986 120.1892
120.3138 120.1832 120.1388 120.2016 120.1116 120.1143
120.0954 120.1665 119.9521 119.9407 120.169 120.4066 120.1715
120.428 120.1977 120.1229 120.1888 120.1686 120.0961 120.303
120.0962 120.151 120.2386 120.2136 120.2039 120.3343 120.0928
120.1401 120.1295 120.4284 120.4153 120.3406 120.1167
120.0528 120.1976 120.2924 120.2671 119.9644 120.4131
120.1658 120.2918 120.1595 120.1064 120.2203 120.2238 120.046
120.1796 120.2836 120.176 119.71 120.23 120.2006 119.9483 120.124
120.1692 120.2891 120.1211 120.105 119.7189 120.1855 120.1935
120.1262 120.092 120.2089 120.3443 120.2245 120.2109 120.1746
120.1044 120.1791 120.0961 120.2805 120.2877 119.9657
120.5894 120.2651 120.2845 120.3612 120.1846 120.1795
120.0462 120.3619 114.5017 120.0922 119.93 120.267 120.2068
120.3776 120.2056 120.3868 120.1263 119.733 126.5936 120.2852
120.1143 120.1374 120.1729 120.2486 120.1643 120.2385
120.1523 120.2123 120.179 120.2201 120.1907 119.9389 120.3045
120.1662 113.4007 120.3463 120.1892 120.2004 119.7321
120.0898 119.0496 120.0305 120.1702 120.1834 119.9536
120.3086 120.2193 120.0609 120.2738 120.0906 120.3418
120.2597 120.3147 120.1818 119.065 120.1575 120.3362 120.2433
120.2328 120.1964 120.1799 119.7074 119.0593 120.225 120.3914
120.2567 119.7111 120.3221 120.1917 120.3017 120.2011
120.1037 119.0409 120.1676 119.0661];%34-9E
y=[31.1989 30.2923 31.2089 30.2705 30.276 30.2508 30.2755 30.2806 30.257
47.3363 30.3136 30.2974 30.2743 30.2793 30.3011 30.2722 30.2786 30.3118 30.2999
31.9818 30.2719 30.2841 30.3056 30.2853 30.31 30.2909 30.2893 30.3022 30.3087
30.2908 30.2924 30.2002 30.2579 30.3005 30.2771 30.2937 30.2426 30.2694 30.2866
30.3034 30.323 30.3069 30.319 30.1987 30.3171 30.2813 30.2879 30.3221 30.2999
30.3099 30.293 30.3227 30.2869 30.318 30.2953 30.1743 30.3097 30.2769 30.2925
30.1667 30.2899 30.309 30.2976 30.3044 30.1706 30.3214 30.3164 30.2714 30.2885
30.3326 30.3142 30.2857 30.2237 30.3358 30.3312 30.3 30.2827 30.3013 30.3385
30.2683 30.2688 30.3059 30.2973 30.3056 30.3007 30.2728 30.2665 30.3202 30.1772
```

```

30.302 30.2855 30.3246 30.2171 30.3118 30.2399 30.2292 30.4178 30.2092 30.2785
30.2783 30.3278 30.3219 30.2074 30.2381 30.2323 30.4169 30.058 30.1906 31.2034
30.1823 30.3363 30.2455 30.2818 30.3302 30.3151 30.2815 30.2843 30.2095 30.1631
30.4378 30.3216 30.2769 30.3201 30.2096 30.4298 30.2385 30.3231 30.4265 30.4234
30.281 30.3032 30.2479 30.3355 30.0661 30.1913 30.3149 30.3147 30.2851 30.1955
30.2649 30.1829 30.2522 30.2742 30.3033 30.3388 30.1824 30.3254 30.4216 30.2963
30.0556 30.2425 30.2124 30.4146 30.264 30.3064 30.2516 30.2825 30.2937 30.2714
30.319 30.0534 30.0618 30.192 30.2256 30.3313 30.2365 30.3359 30.3128 30.2763
30.1982 30.3177 30.4301 30.2587 30.3005 30.3176 30.2801 30.2534 30.3169 30.2947
30.333 30.3402 30.2343 30.2303 30.3271 30.3126 30.2878 30.2699 30.1636 30.162
30.0513 30.2354 30.3161 30.1595 30.1899 30.288 30.2904 30.2618 30.2968 30.314
30.1787 30.3303 30.2303 30.2873 30.352 30.0681 30.3151 30.2291 30.1836 30.2731
30.3384 30.2428 30.2839 30.2454 30.2727 30.3308 30.2782 30.3055 30.3041 30.2858
30.2417 30.3308 30.2658 30.3055 30.3166 30.1664 30.063 27.968 30.1879 30.1551
30.3291 30.3272 30.2414 30.2817 30.3111 38.0096 30.3193 30.0518 30.3177 30.2376
30.3193 30.2622 30.3291 30.3093 30.2285 45.7162 30.1709 30.3369 30.3217 30.3171
30.3116 30.3228 30.292 30.3435 30.2593 30.2318 30.3029 30.2383 30.049 30.4047
30.2491 34.5454 30.2273 30.3019 30.186 30.2482 30.2686 29.6154 30.281 30.2236
30.2616 30.0507 30.4387 30.2599 30.2947 30.3214 30.3014 30.3336 30.1944 30.177
30.2002 29.6112 30.1852 30.2273 30.1873 30.2924 30.2698 30.2314 30.2346 29.6155
30.2184 30.3224 30.3221 30.2225 30.2365 30.1763 30.2473 30.2597 30.2688 29.6167
30.1852 29.6204];%Î³Ë
t=0;
for i=1:size(x,2)
    if x(i)>=120.5 || x(i)<=119.5 || y(i)<=29.5 || y(i)>=31;
        t=t+1;m(t)=i;
    end
end

for j=size(m,2):-1:1
    x(:,m(j))=[];
    y(:,m(j))=[];
end
save F1 x y

plot(x,y,'.');
xlabel('Ë³-','fontsize',12);
ylabel('±±Î³','fontsize',12);

```

附录四：求出租车需求规划图的代码

```

clear all
close all
clc
load F1 x y
x1=[120.30806 120.3116833 120.3488615 120.3852833 120.326575 120.4116667
120.432825 119.7215 119.9507273 120.21614 120.16444 120.18995
120.26779 120.29127 120.1775538 120.2695 120.0281 120.051475
120.1082917 120.1495538 120.1911273 120.130175 120.1736111 120.2308
120.1044571 120.0981571 120.1404412 120.17415 120.2069 120.20358];
y1=[30.41512 30.43105 30.32003846 30.322 30.2346 30.23043333 30.23685
30.23425556 30.05716364 30.21232 30.1901 30.19608 30.17938 30.16921
30.23855385 30.31945 30.28255 30.29025 30.32659167 30.32885385
30.32843636 30.30525 30.30565 30.30363333 30.29152857 30.26878571
30.27990588 30.2735375 30.28865625 30.25625333];
plot(x,y,'.');
hold on

plot(x1,y1,'ko',...
'LineWidth',2,...

```

```

        'MarkerSize',10,...
        'MarkerEdgeColor','k',...
        'MarkerFaceColor','w')
xlabel('距离', 'fontsize',12);
ylabel('时间', 'fontsize',12);

hold on
distance=eye(size(x1,2));
for i=1:size(x1,2)
    for j=i+1:size(x1,2)
        distance(i,j)=sqrt((x1(i)-x1(j))^2+(y1(i)-y1(j))^2);
        distance(j,i)=distance(i,j);
    end
end

```

附录五：求交通小区规划图的代码

```

clear all
close all
clc
load F1 x y
x1=[120.30806 120.3116833 120.3488615 120.3852833 120.326575 120.4116667
120.432825 119.7215 119.9507273 120.21614 120.16444 120.18995 120.26779
120.29127 120.1775538 120.2695 120.0281 120.051475 120.1082917 120.1495538
120.1911273 120.130175 120.1736111 120.2308 120.1044571 120.0981571
120.1404412 120.17415 120.2069 120.20358];
y1=[30.41512 30.43105 30.32003846 30.322 30.2346 30.23043333 30.23685
30.23425556 30.05716364 30.21232 30.1901 30.19608 30.17938 30.16921
30.23855385 30.31945 30.28255 30.29025 30.32659167 30.32885385 30.32843636
30.30525 30.30565 30.30363333 30.29152857 30.26878571 30.27990588 30.2735375
30.28865625 30.25625333];
plot(x,y,'y. ');
hold on
for i=1:30
text(x1(i),y1(i),num2str(i));
end

```

附录六：求交通小区最短路径图的代码

```

clear all
close all
clc
load F1 x y
x1=[120.30806 120.3116833 120.3488615 120.3852833 120.326575 120.4116667
120.432825 119.7215 119.9507273 120.21614 120.16444 120.18995
120.26779 120.29127 120.1775538 120.2695 120.0281 120.051475
120.1082917 120.1495538 120.1911273 120.130175 120.1736111 120.2308
120.1044571 120.0981571 120.1404412 120.17415 120.2069 120.20358];
y1=[30.41512 30.43105 30.32003846 30.322 30.2346 30.23043333 30.23685
30.23425556 30.05716364 30.21232 30.1901 30.19608 30.17938 30.16921
30.23855385 30.31945 30.28255 30.29025 30.32659167 30.32885385
30.32843636 30.30525 30.30565 30.30363333 30.29152857 30.26878571
30.27990588 30.2735375 30.28865625 30.25625333];
m=[1 2;1 3;3 4;5 6;6 7;13 14;10 13;11 12;12 13;12 10;10 13;10 5;11 9;8 17;17
18;18 26;18 25;25 26;25 27;25 22;25 19;26 27;27 22;27 23;27 28;22 19;22 20;22
23;19 20;20 21;20 23;21 23;23 28;23 29;23 24;24 29;24 16;28 30;28 15;30 15;28
29;15 12];
plot(x,y,'. ');

```



```

hold on

for i=1:size(m,1)
plot([x1(m(i,1)),x1(m(i,2))],[y1(m(i,1)),y1(m(i,2))],'k',
'LineWidth',2);hold on
end

plot(x1,y1,'ko',...
'LineWidth',2,...
'MarkerSize',10,...
'MarkerEdgeColor','k',...
'MarkerFaceColor','w')
xlabel('东经','fontsize',12);
ylabel('北纬','fontsize',12);

hold on
distance=eye(size(x1,2));
for i=1:size(x1,2)
    for j=i+1:size(x1,2)
        distance(i,j)=sqrt((x1(i)-x1(j))^2+(y1(i)-y1(j))^2);
        distance(j,i)=distance(i,j);
    end
end
end

```