

# “互联网+”时代的出租车资源配置

## 摘要

出租车“打车难”是人们关注的一个社会热点问题。与之相关的出租车资源的有效分配问题以及打车软件服务平台的补贴政策备受关注，有重要的研究意义。本文通过建立模糊评价模型对不同时空出租车资源的“供求匹配”程度进行了评价；建立微分方程模型预测补贴方案对“打车难”问题的影响；对补贴前后的打车容易度进行了比较分析；最后建立了优化模型寻求一个较好的补贴方案。

针对问题一，选取乘客等待时间以及供需比作为评价指标，并利用搜索数据确定指标值。匹配程度分为高、中、低三个等级，建立综合评价模型对各个需求点的供求匹配程度进行评价。得到上班高峰时段匹配程度以“低”为主，其在 16 点、18 点、20 点所占比例分别为 4.72%、4.82%、4.65%，在打车“低峰期”的供求匹配程度以“中”为主，其在 0 点、6 点、13 点所占比例分别为 4.49%、4.52%、4.62%。

针对问题二，建立微分方程模型对补贴政策下的司机增长量以及乘客数量进行预测，提出“打车容易度”的概念对打车难度进行评估。最后得出，补贴方案会使需求量少的时段打车更加容易，而在高峰时段增大打车难度。补贴政策取消之后，因仍有用户使用打车软件，可以缓解打车难。

针对问题三，考虑到公司利益以及乘客利益，建立以公司投入产出比最高、打车容易度最大、乘客产出投入比最高为目标的多目标优化模型。利用遗传算法求解，得到新的补贴方案为：对乘客的补贴分两个月进行，且每个月的补贴金额分别为 15 元、2 元；对出租车的补贴分 7 个月进行，且每个月的补贴金额分别为 15 元、13 元、7 元、7 元、8 元、9 元、1 元。

关键词：概率模型、模糊综合评价、微分方程模型、优化模型、遗传算法

## 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，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。

为提高打车服务软件平台的服务效率，需搜集相关数据，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) 若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

随着网络技术的迅猛发展，各种国计民生的服务也从实体逐渐走向网络时代，其中打车服务软件的横空出世就是典型的案例，既打破了传统的出门打车模式，给市民出行带来方便，也在一定程度上解决了打车难这一社会难题。与之相关的出租车资源的有效分配问题以及打车软件服务平台的补贴政策备受关注，有重要的研究意义。

### 2.1 问题一的分析

问题一关键是建立与出租车供与需相关的合理的指标，用来衡量供求匹配程度。指标的选取通常是出租车公司以及乘客都关心的变量。对下单乘客来说，从下单开始等待出租车到来的时间是至关重要的，故对不同的打车点，平均等待时间可作为指标之一。匹配程度可通过供需的比例值或划分若干等级如优良中差等来刻画。总之问题一可建立某种评价模型进行解决，难点是搜索各指标相关数据。

### 2.2 问题二的分析

问题二分析的重点是补贴方案。补贴方案的有无将直接影响到问题一中各指标的数量值，如何定量的得到指标数据的变化是解决这一问题的关键。考虑到打车软件是新兴事物，真实可靠的数据缺乏，可根据现有一些打车服务平台如滴滴打车、快的打车等现有的补贴政策，寻找补贴方案对出租车供需的影响规律，根据规律预测出所需数据。能否缓解打车难可利用问题一的分析方法从匹配程度的改变进行回答。

### 2.3 问题三的分析

问题三分析的重点仍旧是补贴方案，与问题二不同的是，在本问中需要自行设计一种补贴方案，而设计的关键则是如何满足合理性这一要求。考虑到打车软件公司实施补贴方案是为了吸引用户，而由问题二可知补贴方案的实施对打车难现象有一定的影响，因此应分别从打车软件公司和乘客两个方面考虑补贴方案的合理性。基于此分析，可以选择合理的指标以分别从两个方面衡量补贴方案的合理性，再以所选指标构建优化模型，从而求得一种较为合理的补贴方案。而衡量指标的选取与确定则是本题的一个难点。

### 三、 模型假设

1. 假设乘客的等待时间只考虑出租车驶到乘客所在地的行驶时间。
2. 出租车只会服务一定范围内的乘客。
3. 出租车的服务范围与补贴金额成正比。

### 四、 符号说明

符号	说明
$m_j$	第 $j$ 个出租车供应点处的出租车数量
$n_i$	第 $i$ 个乘客需求点处的乘客总数
$\eta_i$	第 $i$ 个乘客需求点处的“供需比”
$t_i$	第 $i$ 个乘客需求点处的乘客等待时间
$S_p(k)$	时刻 $k$ 使用打车软件的乘客在总乘客数中所占的比例
$E$	西安市的“打车容易度”
$\varepsilon_p$	针对乘客的产出投入比
$\varepsilon_c$	针对出租车的产出投入比

（注：其他未提及符号在文中说明）

### 五、 模型建立与求解

#### 5.1 问题一的模型建立与求解

问题一要求搜集相关数据，通过建立数学模型，根据合理的指标分析不同时空出

租车资源的“供求匹配”程度。我们首先确定合理的指标，然后建立概率模型求得各指标的相应值，最后利用模糊综合评价模型对“供求匹配”程度进行分析。

### 5.1.1 指标的确定

通过“滴滴快的智能出行平台——苍穹”网站<sup>[1]</sup>可知，乘客与出租车司机安装打车服务软件 app 后，乘客在手机中点击“我要打车”，并发送一段语音说明具体的所在位置以及要去的地址即可。用车信息将被传送到离乘客较近的出租车司机手机中，司机可以在手机中一键抢单，并和乘客联系，提供打车服务。根据目前打车软件的服务方式，原有的路边即时打车方式将会改变，下单乘客需原地等待一定时间，等待接单司机的到来。故对于乘客来说，等待时间的长短将会影响到乘客的满意程度、打车软件的运行效率、服务质量以及下单是否有效执行。另一方面，乘客借助打车软件出行，这种行为改变了传统出租车司机完全凭个人经验来搜索乘客的模式，司机与乘客之间信息透明，降低了非高峰时段出租车的空载比率，也在一定程度上提高了乘客对出租车的需求以及打车成功率。故打车软件的使用将在数量上改变原有出租车的供应与需求。为了方便问题的分析，不妨将供需比作为一衡量指标。即乘客的平均等待时间以及出租车的供需比是平均供需匹配问题的两个重要指标。

#### (1) 指标的确定

##### 指标 1——“供需比”的确定

记出租车的供应点分布为  $j(j=1,2,\dots,n)$ ，记乘客的需求点分布为  $i(i=1,2,\dots,m)$ 。以出租车供应点  $j$  为圆心， $r$  为半径做圆  $C_{ij}$ ，其分布示意图如图 1 所示。

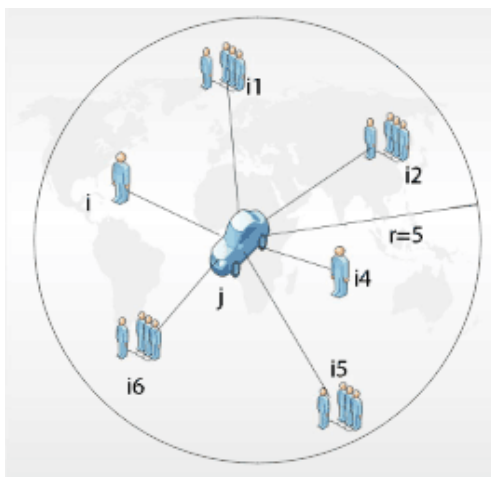


图 1 出租车供应点及乘客需求点的分布图

记图 1 所示圆内第  $i$  个需求点处的乘客总数为  $n_i$ 。由打车软件服务特征可知，第  $j$  个出租车供应点为第  $i$  个乘客需求点提供打车服务的概率  $P_{ji}$  与该需求点处的乘客总数成正比，与两点间的距离  $d_{ij}$  成反比，即：

$$P_{ji} = k \frac{n_i}{d_{ij}} \quad (k \text{ 为常数}) \quad (1)$$

公式 (1) 中， $k$  为比例常数，而  $n_i$  和  $d_{ij}$  未知，需要借助其他信息确定。

当乘客在打车软件客户端发布打车信息后，一定区域内的出租车司机均可通过该软件看到有打车需求的乘客的位置和其打车信息，即可提供打车服务的出租车是若干辆。设第  $i$  个乘客需求点处的出租车供应量  $g_i$  为：

$$g_i = \sum_{j=1}^c m_j P_{ji} \quad (c \text{ 为常数}) \quad (2)$$

其中， $c$  为第  $i$  个乘客需求点范围内的出租车供应点总数， $m_j$  为第  $j$  个出租车供应点处的出租车数量，则第  $i$  个乘客需求点处的“供需比”  $\eta_i$  为：

$$\eta_i = \frac{g_i}{n_i} \quad (3)$$

其中， $g_i$  为第  $i$  个乘客需求点处的出租车供应量， $n_i$  为第  $i$  个乘客需求点处的乘客总数。

## 指标 2——“乘客等待时间”的确定

由模型假设可知，乘客的等待时间只考虑出租车驶到乘客所在地的行驶时间，不考虑出租车的接单时间。而出租车的行驶时间与出租车到乘客所在地的距离以及出租车的速度有关，又由于某个乘客需求点范围内有若干个出租车供应点，每个出租车供应点有若干辆出租车，因此可以采用该需求点范围内所有前往该需求点处的出租车到其的平均距离来衡量乘客的等待时间。

由 (2) 式可知，第  $i$  个乘客需求点处的出租车供应量为  $g_i = \sum_{j=1}^c m_j P_{ji}$ ，则该范围内到第  $i$  个乘客需求点处的所有出租车行驶距离之和为  $\sum_{j=1}^c P_{ji} \cdot d_{ij} \cdot m_j$ 。用该圆范围内所有前往第  $i$  个乘客需求点处的出租车与该需求点之间的距离的期望值来表示该范围内所有前往该需求点处的出租车到该需求点的平均距离  $\bar{d}_i$ ，即：

$$\bar{d}_i = \frac{\sum_{j=1}^c P_{ji} \cdot d_{ij} \cdot m_j}{\sum_{j=1}^c m_j P_{ji}} \quad (4)$$

综上可得第*i*个乘客需求点处的乘客等待时间 $t_i$ 为:

$$t_i = \frac{\bar{d}_i}{v} \quad (5)$$

其中,  $v$  为出租车的行驶速度。

## (2) 指标的求解

由资料<sup>[2]</sup>可知, 乘客等待时间一般不超过 5 分钟, 出租车平均行驶速度为 60km/h, 为保证在乘客等待时间中, 其所在点范围内有出租车能赶到, 因此可得上述圆半径为  $r = v \cdot \bar{t} = 5 \text{ min} \cdot 60 \text{ km/h} = 5 \text{ km}$ 。

以西安市为例, 由“滴滴快的智能出行平台——苍穹”网站<sup>[1]</sup>可以查到西安市各需求点处的乘客总数 $n_i$ 以及供应数 $m_j$ 的值部分如表 1 所示(完整的 $n_i$ 值及 $m_j$ 值见支撑材料)。

表 1 西安市各需求点处的需求量 $n_i$ 及供应量 $m_j$ 的值(部分)

时间	经度	纬度	供应量 $m_j$	需求量 $n_i$
0:00	108.95	34.30	1	5
	⋮	⋮	⋮	⋮
	108.96	34.25	1	4
1:00	109.08	34.28	10	2
	⋮	⋮	⋮	⋮
	108.96	34.25	3	4
⋮	⋮	⋮	⋮	⋮
23:00	108.99	34.33	2	3
	⋮	⋮	⋮	⋮
	108.95	34.27	7	8

表 1 中给出了在 0~23 各整点时刻不同经纬度处的供应量与需求量, 且  $i, j$  的取值分别为  $i=1, 2, \dots, 301$ ;  $j=1, 2, \dots, 301$ 。

由“苍穹”网站可以查得在 0~23 各整点时刻第  $j$  个出租车供应点与第  $i$  个乘客需

求点之间的距离矩阵（见支撑材料），现将 0 点时刻的距离矩阵部分表示如下：

$$d_{ij} = \begin{bmatrix} 11.95 & 7.10 & \cdots & 6.96 \\ 6.93 & 11.79 & \cdots & 12.00 \\ \vdots & \vdots & \vdots & \vdots \\ 1.37 & 6.23 & \cdots & 6.41 \end{bmatrix}_{(301) \times (301)}$$

因为  $\sum P_{ji} = 1$ ，则根据查到的西安市各需求点处的乘客总数  $n_i$  及各整点时刻的距离矩阵  $d_{ij}$  可以确定（1）式中常数  $k$  的值，然后利用 MATLAB 软件求解（1）式可得在 0~23 各整点时刻以第  $i$  个乘客需求点为中心，5km 为半径的圆范围内各出租车供应点处的出租车前往该需求点处的概率矩阵（见支撑材料），现将 0 点时刻的概率矩阵部分表示如下：

$$P_{ji} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 0.023 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0.060 & 0 & \cdots & 0 \end{bmatrix}_{(301) \times (301)}$$

矩阵中的“0”表示“不可行”。

### 1. “供需比”的求解

由“苍穹”网站可以得到各出租车供应点处的出租车数量  $m_j$ ，利用 MATLAB 求解（2）式可以得到西安市各乘客需求点处的出租车供应量  $g_i$  的值（支撑材料）。

根据表 1 中的西安市各需求点处的乘客总数  $n_i$  以及出租车供应量  $g_i$  的值，利用 MATLAB 软件求解（3）式可以得到西安市各乘客需求点处的“供需比”  $\eta_i$  的值部分如表 2 所示（完整的“供需比”见支撑材料）。

表 2 西安市各乘客需求点处的“供需比”  $\eta_i$

序号 $i$	时间	1:00	2:00	...	24:00
1	$\eta_i$	6.91	3.16	...	2.77
$\vdots$		$\vdots$	$\vdots$	$\vdots$	$\vdots$
301		5.07	3.53	...	2.66

表 2 给出了在 0~23 各整点时刻西安市不同乘客需求点处的供需比，且  $i$  的取值为

$i=1,2,\cdots,301$ 。

## 2. “乘客等待时间”的求解

由“苍穹”网站可以得到第 $i$ 个乘客需求点与其所在圆范围内的各出租车供应点之间的距离，以及该范围内各出租车供应点处的出租车数量，又知出租车行驶速度为60km/h，因此利用 MATLAB 软件求解（4）（5）式可得在0~23各整点时刻西安市各乘客需求点处的乘客等待时间 $t_i$ 的值（见支撑材料），现取0点时刻的等待时间部分值如表3所示。

表3 1点时西安市各乘客需求点处的乘客等待时间 $t_i$ （部分）

$i$	1	2	...	301
$t_i$	1.48	1.19	...	0

## 5.1.2 模糊综合评价模型的建立与求解

### （1）模糊综合评价模型的建立

模糊综合评价方法是应用模糊关系合成的原理，从多个因素（指标）对被评价事物隶属等级状况进行综合性评判的一种方法。对于本问，我们通过建立模糊综合评价模型对不同时空出租车资源的“供求匹配”程度进行分析。

首先确定出租车资源“供求匹配”程度的评价因素论域为 $U=(\eta_i, t_i)$ ，其中， $\eta_i$ 为第 $i$ 个乘客需求点处的“供需比”， $t_i$ 为第 $i$ 个乘客需求点处的乘客等待时间。然后将“供求匹配”程度分为三个等级，确定评语等级论域为 $V=(\text{高}, \text{中}, \text{低})$ 。之后进行单因素评价，对每一个 $i$ 值建立模糊关系矩阵 $R_i$ 为：

$$R_i = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \end{pmatrix}, \quad 0 \leq r_{kj} \leq 1 \quad (6)$$

其中， $r_{kj}$ 为评价因素论域 $U$ 中的第 $k$ 个因素对于评语等级论域 $V$ 中第 $j$ 个等级的隶属关系。

确定评判因素权向量 $A=(a_1, a_2)$ ， $A$ 是评价因素论域 $U$ 中各因素对出租车资源“供求匹配”程度的隶属关系，它取决于人们进行模糊综合评判时的着眼点，即根据评判时各因素的重要性分配权重。

最后选择评价的合成算子，将评价因素权向量 $A$ 与每一个 $i$ 值对应的模糊关系矩



阵  $R_i$  合成得到综合评判矩阵  $B_i = (b_{i1}, b_{i2}, b_{i3})$ ，且

$$B_i = A \circ R_i = (a_1, a_2) \circ \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \end{pmatrix} \quad (7)$$

根据模糊乘法模型中的主因素突出模型计算 (7) 式可得综合评价矩阵  $B_i$  中各元素的表达式为：

$$b_{ij} = \vee (a_i \cdot r_{ij} | i=1, 2), \quad j=1, 2, 3 \quad (8)$$

## (2) 模糊综合评价模型的求解

我们以某个乘客需求点处的“供需比”以及乘客的等待时间为指标对不同时空出租车资源的“供求匹配”程度进行分析。通过梯形型隶属度函数对第  $i$  个乘客需求点处的“供需比”及其乘客等待时间进行处理，分别得到其隶属度矩阵部分如表 4 和表 5（见附录）所示（完整隶属度矩阵见支撑材料）。

记第  $i$  个乘客需求点处“供需比”的隶属度矩阵的第  $k$  行为  $C_k$ ，第  $i$  个乘客需求点处乘客等待时间的隶属度矩阵的第  $k$  行为  $D_k$ ，则可以得到每一个  $i$  值对应的模糊关系矩阵  $R_i$  为：

$$R_i = \begin{pmatrix} C_k \\ D_k \end{pmatrix} \quad (9)$$

为了评定不同时空出租车资源的“供求匹配”程度，不失一般性地取权数分配为  $A = (0.5, 0.5)$ 。

综上利用 MATLAB 软件求解 (8) (9) 式可得 9 月 7 日 0 点到 23 点的综合评判矩阵  $B_i$ （见支撑材料），现将 0 点时刻的综合评判矩阵部分表示如下：

$$B_i = \begin{bmatrix} 0.57 & 0.99 & 0.59 \\ 0 & 0 & 1 \\ \vdots & \vdots & \vdots \\ 0.69 & 0.86 & 0.61 \end{bmatrix}_{(301) \times (3)}$$

根据最大隶属度原则可以确定 9 月 7 日 0 点到 23 点各整点时刻西安市不同地区出租车资源的“供求匹配”程度的级别（见支撑材料），现将 0 点时刻的“供求匹配”程度的级别部分列于表 6 所示。

表 6 1 点时刻的“供求匹配”程度的级别（部分）

需求点 <i>i</i>	1	2	3	...	301
级别	高	高	中	...	高

根据“供求匹配”的程度级别值，利用 MATLAB 软件可以得到 9 月 7 日 0 点到 23 点西安市不同地区出租车资源的“供求匹配”程度级别图如图 2（见附录）所示，现选取部分“打车高峰期”及“打车低峰期”的“供求匹配”程度级别图分别如图 3 和图 4 所示。

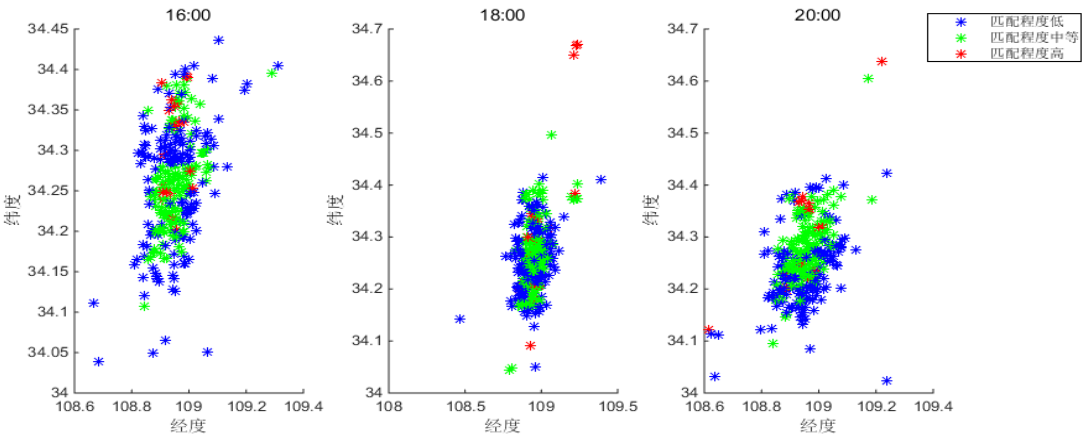


图 3 部分“打车高峰期”的供求匹配程度级别图

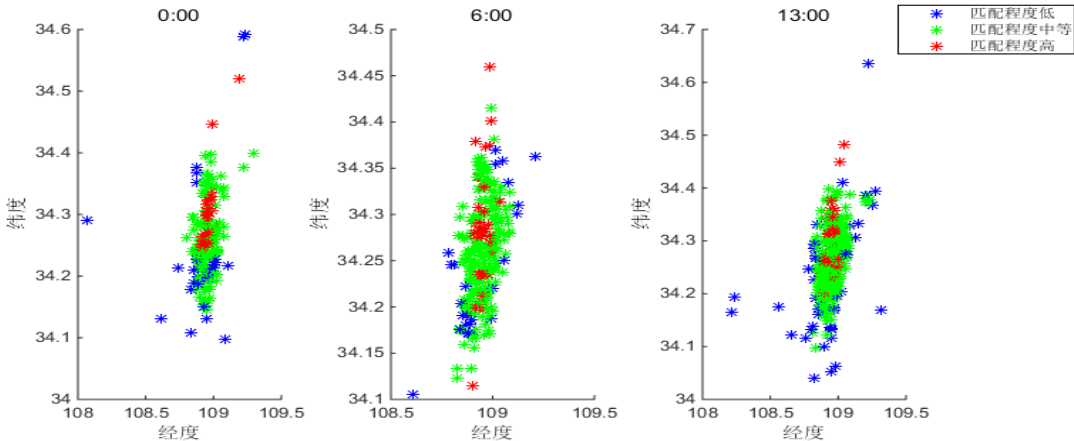


图 4 部分“打车低峰期”的供求匹配程度级别图

图 3 与图 4 中红色点表示供求匹配程度高，绿色点表示供求匹配程度中等，蓝色点表示供求匹配程度低。由图 3 可以看出，在“打车高峰期”16 点、18 点和 20 点，

供求匹配程度的级别以匹配程度低为主，该程度在这三个整点时刻所占的比例分别为 4.72%、4.82%、4.65%。由图 4 可以看出，在“打车低峰期”0 点、6 点和 13 点，供求匹配程度的级别以匹配程度中等为主，该程度在这三个整点时刻所占的比例分别为 4.49%、4.52%、4.62%。

## 5.2 问题二的模型建立与求解

问题二要求建立数学模型，根据相关数据分析各公司的出租车补贴方案是否对“缓解打车难”有帮助。我们首先建立预测模型对未来一段时间内使用打车软件的乘客和司机的数量进行预测，然后通过比较实施补贴方案前后对应时间的西安市“打车容易度”确定各公司的出租车补贴方案是否对“缓解打车难”有帮助。

### 5.2.1 乘客数量预测模型的建立与求解

#### (1) 乘客数量预测模型的建立

由于补贴方案实施后补贴政策会在社会中传播开来，并影响使用打车软件的乘客数量，因此我们根据传染病模型中的 SI 模型建立乘客数量预测模型，用其预测在补贴政策实施后未来一段时间内，西安市使用打车软件的乘客数。

假设在预测的时间段内西安市的乘客总数  $N_p$  不变，乘客人群分为使用打车软件的乘客和未使用打车软件的乘客两类。时刻  $k$  这两类人在总乘客数中所占的比例分别记作  $S_p(k)$  和  $W_p(k)$ 。

又假设每个司机每个月可使  $\lambda S_p(t)$  个不使用打车软件的人变为使用的人，其中  $\lambda$  为与补贴方案有关的函数，根据速途网站<sup>[2]</sup>所给数据可以确定  $\lambda$  的表达式为：

$$\lambda = 0.1 + 0.01\omega_p(k) \quad (10)$$

其中， $\omega_p(k)$  为公司对乘客的补贴函数，在此无法确定其函数式，需借助其他信息确定。

则使用打车软件的乘客的月增长率为  $\lambda N_p W_p S_p$ ，则有：

$$N_p \frac{dS_p}{dk} = \lambda N_p W_p S_p \quad (11)$$

又因为

$$S_p(k) + W_p(k) = 1 \quad (12)$$

再记初始时刻（ $k=0$ ）使用打车软件的乘客的比例为  $S_{p0}$ ，则可建立乘客数量预测模型为：

$$\frac{dS_p}{dk} = \lambda S_p (1 - S_p), \quad S_p(0) = S_{p0} \quad (13)$$

## (2) 乘客数量预测模型的求解

### 1. 初始值的求解

通过查阅新华网陕西频道<sup>[3]</sup>可知,2015年西安市的出租车日均运量为113万人次。通过“滴滴打车”软件的补贴方案(见附录表10)可知,其采用的补贴方案为:对于乘客的补贴分为5个月进行,每个月的补贴金额分别为10元、10—15元、6—15元、3—5元、0元;对于出租车的补贴分为8个月进行,每个月的补贴金额分别为10元、10元、10元、10元、10元、10元、2元、0元。

将实施补贴方案的前一月份作为初始月份,在初始月份中选取某天0时、14时、18时使用打车软件的乘客数为初始值,用其预测在之后月份中对应日期的对应时刻使用打车软件的乘客数。由“苍穹”网站提供的数据可以得到在初始月份某天0时、14时、18时使用打车软件的乘客数,因此可以得到使用打车软件的乘客比例的初始值 $S_{p0}$ 如表7所示。

表7 使用打车软件的乘客比例的初始值

时刻(单位:时)	0	14	18
乘客数(单位:人)	668	1260	1322
$S_{p0}$ (单位:‰)	5.91	11.15	11.70

由表7可以看出,在初始月份中,使用打车软件的乘客在总乘客数中所占比例很低,在所选的三个时刻中最大比例仅为11.70‰。

### 2. 乘客数量预测值的求解

根据所查到的“滴滴打车”公司的补贴方案(见附录表10),可以得到其对乘客的补贴函数 $\omega_p(k)$ 为:

$$\omega_p(k) = \begin{cases} 10, & k=1 \\ 16, & k=2 \\ 10.5, & k=3 \\ 4, & k=4 \\ 0, & 5 \leq k \leq 8 \end{cases} \quad (14)$$

其中， $k$  为月份。

由于（12）式是 logistic 模型，因此其解为：

$$S_p(k) = \frac{1}{1 + \left( \frac{1}{S_{p0}} - 1 \right) e^{-\lambda k}} \quad (15)$$

其中， $S_p(k)$  为时刻  $k$  使用打车软件的乘客在总乘客数中所占的比例， $S_{p0}$  为初始时刻（ $k=0$ ）使用打车软件的乘客的比例。

根据表 9 中使用打车软件的乘客比例的初始值以及“滴滴打车”公司对乘客的补贴函数，利用 MATLAB 软件求解（14）（15）式可得未来 8 个月中，与所选初始值对应日期的 0 时、14 时、18 时使用打车软件的乘客数（见支撑材料），现将实施补贴方案后的第一个月中使用打车软件的乘客数部分列于表 8。

表 8 实施补贴方案后第一个月使用打车软件的乘客数（部分）

时刻	0:00	14:00	18:00
乘客数	820.7438	1543.422	1618.095
	822.7438	1543.422	1620.095
	⋮	⋮	⋮
	815.7438	1540.422	1617.095

## 5.2.2 出租车数量预测模型的建立与求解

### （1）出租车数量预测模型的建立

由于补贴方案实施后会在出租车司机之间传播并影响使用打车软件的出租车数量，因此我们根据传染病模型中的 SI 模型建立出租车数量预测模型，用其预测在未来一段时间内，西安市使用打车软件的出租车数。

与乘客数量预测模型的建立方法类似，故根据乘客数量预测模型的建立方法可以得到：

$$N_c \frac{dS_c}{dk} = \mu N_c W_c S_c \quad (16)$$

其中， $N_c$  为西安市的出租车总数， $S_c(k)$  和  $W_c(k)$  分别为使用打车软件的出租车和未使用的在第  $k$  月份中在出租车总数中所占的比例， $\mu$  为与补贴方案有关的函数。

同样根据速途网站<sup>[2]</sup>所给数据可以确定  $\mu$  的表达式为：

$$\mu = 0.1 + 0.01\omega_c(k) \quad (17)$$

其中  $\omega_c(k)$  为公司对出租车的补贴函数，在此无法确定，需要借助其他信息确定。

又知

$$S_c(k) + W_c(k) = 1 \quad (18)$$

再记初始时刻（ $k=0$ ）使用打车软件的出租车的比例为  $S_{c0}$ ，则可得到出租车供应量预测模型为：

$$\frac{dS_c}{dk} = \mu S_c (1 - S_c), \quad S_c(0) = S_{c0} \quad (19)$$

## （2）出租车数量预测模型的求解

### 1. 初始值的求解

通过查阅新华网陕西频道<sup>[3]</sup>可知，2015 年西安市有出租汽车共 12135 辆。与乘客数量预测模型中求解初始值的方法相同，首先在初始月份中选取某天 0 时、14 时、18 时使用打车软件的出租车数为初始值，用其预测在之后月份中对应日期的对应时刻使用打车软件的出租车数。然后由“苍穹”网站提供的数据得到在初始月份某天 0 时、14 时、18 时使用打车软件的出租车数，因此可以得到使用打车软件的出租车比例的初始值  $S_{c0}$  如表 9 所示。

表 9 使用打车软件的出租车比例的初始值

时刻（单位：时）	0	14	18
出租车数（单位：辆）	2480	3127	1809
$S_{c0}$ （单位：1）	0.204	0.258	0.149

由表 9 可以看出，在初始月份中，使用打车软件的出租车在出租车总数中所占的比例很低，在所选的三个时刻中最大比例仅为 0.258。

### 2. 出租车数量预测值的求解

与乘客数量预测模型中求解乘客数量预测值的方法相同，首先根据所查到的“滴滴打车”公司的补贴方案（见附录表 10），得到其对出租车的补贴函数  $\omega_c(k)$  为：

$$\omega_c(k) = \begin{cases} 10, & 1 \leq k \leq 6 \\ 2, & k = 7 \\ 0, & k = 8 \end{cases} \quad (20)$$

其中， $k$  为月份。

然后按照确定  $\lambda$  的函数表达式的方法，同样不失一般性地给定  $\mu$  的函数表达式为：

由于（18）式是 logistic 模型，因此其解为：

$$S_c(k) = \frac{1}{1 + \left( \frac{1}{S_{c0}} - 1 \right) e^{-\mu k}} \quad (21)$$

其中， $S_c(k)$  为时刻  $k$  使用打车软件的出租车在出租车总数中所占的比例， $S_{c0}$  为初始时刻（ $k=0$ ）使用打车软件的出租车的比例， $\mu$  为与补贴方案有关的函数。

根据表 11 中使用打车软件的出租车比例的初始值以及“滴滴打车”公司对出租车的补贴函数，利用 MATLAB 软件求解（20）（21）式可得未来 8 个月中，与所选初始值对应日期的 0 时、14 时、18 时使用打车软件的出租车数（见支撑材料），现将实施补贴方案后的第一个月中使用打车软件的出租车数部分列于表 11。

表 11 实施补贴方案后第一个月使用打车软件的出租车数（部分）

时刻	0:00	14:00	18:00
出租车数	6450.129	7151.49	5097.628
	6459.129	7152.49	5098.628
	$\vdots$	$\vdots$	$\vdots$
	6450.129	7152.49	5117.628

### 5.2.3 实施补贴方案对“缓解打车难”的影响

#### （1）出租车服务范围的确定

由于对出租车的补贴会使出租车的服务范围增大，通过查阅西安市出租车官网得知，出租车的收费标准为 2 元/公里，因此实施补贴方案后，出租车的服务范围半径  $r(k)$  为：

$$r(k) = 5 + \frac{\omega_c(k)}{2} \quad (22)$$

其中， $k$  为月份， $\omega_c(k)$  为“滴滴打车”公司对出租车的补贴函数。

#### （2）实施补贴方案对“缓解打车难”的影响

根据问题一中对某个乘客需求点处的“供需比”以及乘客等待时间的求解方法，

利用实施补贴方案后的相应数据可以求得西安市各个乘客需求点在实施补贴方案后的“供需比”以及乘客等待时间，将其分别进行归一化处理，并求出归一化后的“供需比”的平均值 $\bar{\eta}$ 以及乘客等待时间的平均值 $\bar{t}$ 。

定义西安市的“打车容易度” $E$ 为 $\bar{\eta}$ 和 $\bar{t}$ 的加权求和，且不失一般性地给定 $\bar{\eta}$ 和 $\bar{t}$ 的权重均为 0.5，即：

$$E = 0.5\bar{\eta} + 0.5\bar{t} \quad (23)$$

其中， $E$ 为西安市的“打车容易度”， $\bar{\eta}$ 为归一化后的西安市各乘客需求点处“供需比”的平均值， $\bar{t}$ 为归一化后的西安市各乘客需求点处乘客等待时间的平均值。

将上述求得的 $\bar{\eta}$ 和 $\bar{t}$ 代入（23）式，利用 MATLAB 软件可以解得在未来 8 个月中与所选初始值对应日期的 0 时、14 时、18 时西安市的“打车容易度”。利用问题一中的相关数据求得 $\bar{\eta}$ 和 $\bar{t}$ ，将其代入（23）式即可解得初始月份（即 $k=0$ ）中该天 0 时、14 时、18 时西安市的“打车容易度”。

综合上述求解结果可以得到实施补贴方案前后对应时间的西安市“打车容易度”如表 12 所示。

表 12 实施补贴方案前后对应时间的西安市“打车容易度” $E$

时 刻 \ 月 份	初 始 月 份	1 月	2 月	3 月	4 月	5 月	6 月	7 月	8 月
0 时	1.261	1.648	1.479	1.394	1.380	1.405	1.418	1.366	1.306
14 时	1.459	0.631	0.565	0.530	0.520	0.524	0.523	0.528	0.517
18 时	0.948	0.398	0.367	0.355	0.357	0.369	0.378	0.403	0.405

根据表 12 可以用 MATLAB 软件画出实施补贴方案前后对应时间的西安市“打车容易度”图像如图 4 所示。



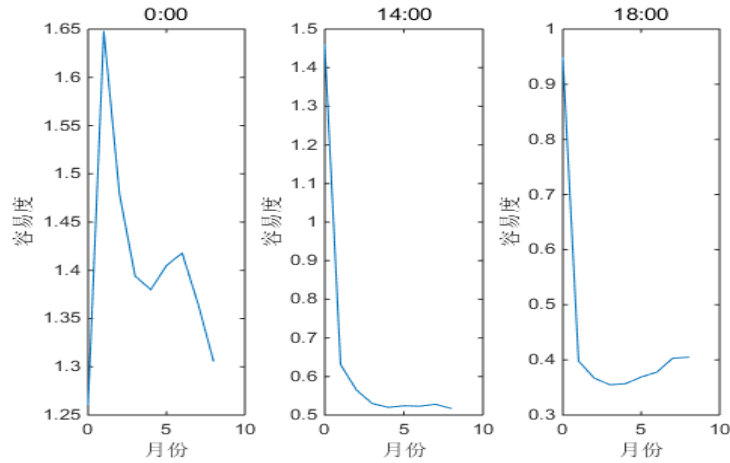


图 5 实施补贴方案前后的西安市“打车容易度”图像

图 5 中横轴表示月份，纵轴表示打车容易度。由图 5 可以看出，实施补贴方案后，在 0:00 的打车容易度较实施前有提高，因此，实施补贴方案缓解了 0:00 的打车难现象。而对于 14:00 和 18:00，实施补贴方案后，该时刻的打车容易度较实施之前均有下降，因此补贴方案的实施加剧了该时刻的打车难程度。造成该现象的可能原因为，在 0:00 使用打车软件的乘客增长数小于出租车增长数，从而缓解打车难现象，而在 14:00 和 18:00 其乘客增长数大于出租车增长数，从而加剧了打车难程度。但补贴政策取消之后，因仍有用户使用打车软件，可以缓解打车难。

### 5.3 问题三的模型建立与求解

问题三要求为一个新的打车软件服务平台设计一种补贴方案，并论证其合理性。我们首先从打车软件公司和乘客两个方面考虑建立优化模型，然后利用遗传算法对其进行求解从而得出一种合理的补贴方案。

#### 5.3.1 优化模型的建立

为了设计一种合理的补贴方案，我们分别从打车软件公司和乘客两个方面来考虑。对于打车软件公司来说，实施补贴方案的目的有两种，一是吸引更多的乘客使用其软件，二是吸引更多的出租车使用其软件。因此可以分别定义针对乘客和出租车的产出投入比来衡量实施补贴方案对软件公司收益的影响。

对于乘客来说，打车难现象与其自身息息相关，由问题二中可以得知，补贴方案的实施加剧了打车“高峰期”以及“普通期”的“打车难”现象，因此乘客对于补贴方案的制定与实施起到一定的约束作用，我们采用问题二中的“打车容易度”来衡量乘客对其的约束作用。

通过参考“滴滴打车”软件的补贴方案，我们设定针对乘客的补贴方案以 5 个月为周期，而针对出租车的补贴方案以 8 个月为周期，即针对乘客的补贴方案为从第一个月开始给补贴，到第 5 个月时补贴变为 0，而针对出租车的补贴方案为从第一个月开始给补贴，到第 8 个月时补贴变为 0。

### (1) 针对乘客的产出投入比的确定

根据问题二中的乘客数量预测模型可得，在预测的时间段内西安市的乘客总数  $N_p$  不变，时刻  $k$  乘客人群中使用打车软件的乘客在总乘客数中所占的比例为  $S_p(k)$ ，且初始时刻（ $k=0$ ）使用打车软件的乘客的比例为  $S_{p0}$ 。假设  $b_p(k)$  为打车软件公司在第  $k$  个月对乘客的补贴金额，已知  $1 \leq k \leq 5$ ，则可以定义针对乘客的产出投入比  $\varepsilon_p$  为：

$$\varepsilon_p = \frac{[S_p(4) - S_p(0)] \cdot N_p}{\sum_{k=1}^5 [S_p(k) - S_p(k-1)] \cdot b_p(k) \cdot N_p} \quad (24)$$

该式说明针对乘客的产出投入比  $\varepsilon_p$  表示了打车软件公司每补贴 1 元给乘客可以增加的使用该软件的乘客数。

### (2) 针对出租车的产出投入比的确定

与确定针对乘客的产出投入比的方法相同，首先根据问题二中出租车数量预测模型可得，在预测的时间段内西安市的出租车总数  $N_c$  不变，时刻  $k$  使用打车软件的出租车在出租车总数中所占的比例为  $S_c(k)$ ，且初始时刻（ $k=0$ ）使用打车软件的出租车的比例为  $S_{c0}$ 。假设  $b_c(k)$  为打车软件公司在第  $k$  个月对出租车的补贴金额，已知  $1 \leq k \leq 8$ ，则可以定义针对出租车的产出投入比  $\varepsilon_c$  为：

$$\varepsilon_c = \frac{[S_c(7) - S_c(0)] \cdot N_c}{\sum_{k=1}^8 [S_c(k) - S_c(k-1)] \cdot b_c(k) \cdot N_c} \quad (25)$$

该式说明针对出租车的产出投入比  $\varepsilon_c$  表示了打车软件公司每补贴 1 元给出租车可以增加的使用该软件的出租车数。

### (3) 优化模型的建立

由问题二可得，西安市的“打车容易度”  $E$  的表达式如（23）式所示。

由于一种合理的补贴方案需要同时使打车软件用户量增加最多且使“打车难”现象最轻，即令针对乘客的产出投入比  $\varepsilon_p$ 、针对出租车的产出投入比  $\varepsilon_c$ 、西安市的“打

车容易度”  $E$  皆最大。因此可以  $\varepsilon_p$ 、 $\varepsilon_c$  和  $E$  三者之和最大为目标函数建立优化模型如下所示：

$$\max \{ \varepsilon_p + \varepsilon_c + E \} \quad (26)$$

### 5.3.2 优化模型的求解

遗传算法起源于对生物系统所进行的计算机模拟研究，是由相关研究人员受到生物模拟技术的启发，而创造出的一种基于生物遗传和进化机制的适用于复杂系统优化的自适应概率优化技术。

由于遗传算法的本质是一种高效、并行、全局搜索的方法，它能在搜索过程中自动获取和积累有关搜索空间的知识，在潜在的解决方案种群中逐次产生一个近似最优的方案，具有实用、高效、鲁棒性强等特征<sup>[4]</sup>，因此我们采用遗传算法对本问进行求解。

将上述优化模型的目标函数看做遗传算法的适应度函数，按照图 5 所示的流程图利用遗传算法对本问所建立的优化模型进行求解。

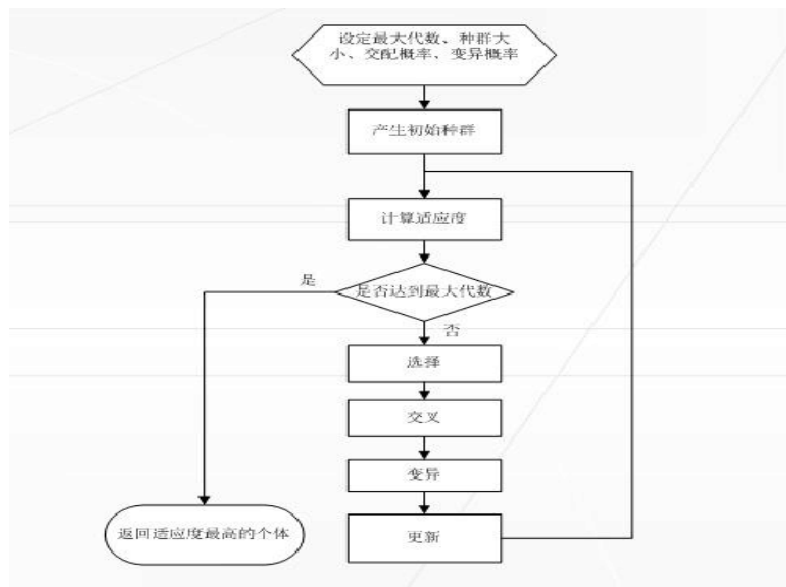


图 6 利用遗传算法求解本模型的流程图

由于“滴滴出行”软件的补贴政策中，第一个月的补贴金额为 10 元，为提高本软件的竞争度，故将第一个月的补贴金额定为 15 元，则利用遗传算法求解上述优化模型可得，最佳适应度为 22.2578，最佳种群为：001000000000111001110111100010010001。

由最佳种群解码可得近似最优的补贴方案为如表 13 所示。

表 13 近似最优补贴方案

对 象	月 份								
	金 额 (元)	1	2	3	4	5	6	7	8
乘客		15	2	0	0	0	0	0	0
出租车		15	13	7	7	8	9	1	0

由表 13 可以看出，新的补贴方案为：对乘客的补贴分五个月进行，且每个月的补贴金额分别为 15 元、2 元；对出租车的补贴分 7 个月进行，且每个月的补贴金额分别为 15 元、13 元、7 元、7 元、8 元、9 元、1 元。

## 六、 模型评价

### 7.1 模型的优点

1. 本文利用模糊综合评价模型，对供求匹配程度进行分级，并将其用图像表示，直观形象地反映了其在不同时空的供求匹配程度情况；
2. 本文利用微分方程模型对之后的变化情况进行预测，科学方便；
3. 本文利用遗传算法对补贴方案进行求解，易得到全局较优解，具有较好的普适性。

### 7.2 模型的缺点

1. 问题一考虑因素较少，可能存在不当之处；
2. 模糊评价模型隶属度函数的给定具有一定的主观性。

## 七、 模型改进及推广

### 8.1 模型改进

评判集的权重向量的确定在综合评判中起重要的作用，通常情况下可以由决策人凭经验给出，但往往带有一定的主观性。为了更客观地反映实际情况，可以采用加权统计法、频数统计法以及转降骨法，或者更一般的模糊关系法来确定<sup>[5]</sup>。

### 8.2 模型推广

本文通过建立概率模型以及模糊综合评价模型对“供需匹配”程度进行评价。概率模型在解决有随机因素影响的问题时具有广泛的应用；模糊综合评价模型是模糊决策中最常用的一种有效方法。

本文建立微分方程模型对补贴之后的供应量以及需求量进行预测,进而求得容易度,将其与实施补贴方案之前进行比较,从而得出结论。此模型在研究实际对象随时间的演变过程,分析其变化规律等方面有着重要的应用。

本文通过遗传算法对多目标规划模型进行求解,得到了全局近似最优解。遗传算法是现代启发式算法,在对 NP-Hard 问题的求解领域中占有重要的地位。

## 八、参考文献

- [1]滴滴快的智能出行平台——苍穹, <http://v.kuaidadi.com>, 2015/9/11。
- [2]速途网, <http://tech.sootoo.com/>, 2015/9/12。
- [3]王一帆, 基于打车软件的出租车服务模式优化研究,  
[http://www.cnki.net/KCMS/detail/detail.aspx?QueryID=1&CurRec=1&recid=&filename=1015028640.nh&dbname=CMFD201501&dbcode=CMFD&pr=&urlid=&yx=&uid=WEEvREcwSl1JHSldSdnQ1ZHFKY1RoN2ZBaUx5UUxRMXFVS1hkaXAzYmMrcm5SMmZrSGhVbjJDQ0JUc1IrVWhlV3dRPT0=\\$9A4hF\\_YAuvQ5obgVAqNKPCYcEjKensW4IQMovwHtwkF4VYPoHbKxJw!!&v=MTg0NjdmYnVkckZpcmxVTC9QVkJyNkc3TzZGdGZJcJVFYlBJUjhlWDFMdXhZUzdEaDFUM3FUclldNMUZYQ1VSTCs=](http://www.cnki.net/KCMS/detail/detail.aspx?QueryID=1&CurRec=1&recid=&filename=1015028640.nh&dbname=CMFD201501&dbcode=CMFD&pr=&urlid=&yx=&uid=WEEvREcwSl1JHSldSdnQ1ZHFKY1RoN2ZBaUx5UUxRMXFVS1hkaXAzYmMrcm5SMmZrSGhVbjJDQ0JUc1IrVWhlV3dRPT0=$9A4hF_YAuvQ5obgVAqNKPCYcEjKensW4IQMovwHtwkF4VYPoHbKxJw!!&v=MTg0NjdmYnVkckZpcmxVTC9QVkJyNkc3TzZGdGZJcJVFYlBJUjhlWDFMdXhZUzdEaDFUM3FUclldNMUZYQ1VSTCs=), 2015/9/12。
- [4]新华网陕西频道, 西安出租车明日迈入 9 元时代 调价期间实行两种运价,  
[http://www.sn.xinhuanet.com/2015-03/31/c\\_1114822983.htm](http://www.sn.xinhuanet.com/2015-03/31/c_1114822983.htm), 2015/9/12。
- [5]秦茜, 公共自行车租赁系统调度问题研究,  
[http://www.cnki.net/KCMS/detail/detail.aspx?QueryID=0&CurRec=1&recid=&filename=1013278348.nh&dbname=CMFD201402&dbcode=CMFD&pr=&urlid=&yx=&uid=WEEvREcwSl1JHSldSdnQ1Y3NpRjNmaFdJdWdGNgluTy9TL254RnNGeXhlMmFPYWRDdzZqZmsxaG5BOGEvOEh1N0h3PT0=\\$9A4hF\\_YAuvQ5obgVAqNKPCYcEjKensW4IQMovwHtwkF4VYPoHbKxJw!!&v=MjM4MTFyRml6bFVMck9WRjI2SGJHLOZ0TElwnUViUElSOGVYMUxleFlTN0RoMVQzcVRyV00xRnJdVjJMK2ZidWQ=](http://www.cnki.net/KCMS/detail/detail.aspx?QueryID=0&CurRec=1&recid=&filename=1013278348.nh&dbname=CMFD201402&dbcode=CMFD&pr=&urlid=&yx=&uid=WEEvREcwSl1JHSldSdnQ1Y3NpRjNmaFdJdWdGNgluTy9TL254RnNGeXhlMmFPYWRDdzZqZmsxaG5BOGEvOEh1N0h3PT0=$9A4hF_YAuvQ5obgVAqNKPCYcEjKensW4IQMovwHtwkF4VYPoHbKxJw!!&v=MjM4MTFyRml6bFVMck9WRjI2SGJHLOZ0TElwnUViUElSOGVYMUxleFlTN0RoMVQzcVRyV00xRnJdVjJMK2ZidWQ=), 2015/9/13。
- [6]韩中庚, 数学建模方法及其应用, 北京: 高等教育出版社, 2005, 335-337。
- [7]姜启源 谢金星 叶俊, 数学模型, 北京: 高等教育出版社, 2011。
- [8]卓金武, MATLAB 在数学建模中的应用, 北京: 北京航空航天大学出版社, 2014

## 九、 附录

### 附录一 程序

所用软件：MATLAB

```
1.tiqushujv.m
clear all
clc
%将苍穹平台的供应量和需求量导出
A='在此输入原始数据';
F=get_num(A);
cnt=size(F,1)/3;
K=zeros(cnt,3);
for k=1:size(F,3)
    j=1;
    for i=1:size(F,1)
        if isempty(str2num(F(i,:,k)))==0
            if mod(i,3)==1
                K(j,1)=str2num(F(i,:,k));
            end
            if mod(i,3)==2
                K(j,2)=str2num(F(i,:,k));
            end
            if mod(i,3)==0
                K(j,3)=str2num(F(i,:,k));
                j=j+1;
            end
        end
    end
    xlswrite('D:\数模国赛\数据\导出数据.xlsx',K,k)
end

%预处理数据
function f=get_num(x)
n=length(x);
k=0;
j=1;
p=0;
for i=1:n
    if x(i)=='['&&x(i+1)=='['
        p=p+1;
        k=0;
    end
    if x(i)>='0'&&x(i)<='9' || x(i)=='.'
        if (x(i-1)<'0' || x(i-1)>'9')&&x(i-1)~='.'
            k=k+1;
            j=1;
        end
        B(k,j,p)=x(i);
        j=j+1;
    end
end
f=B;
2.first.m
clear all
clc
```

```

v=60;
Su=zeros(301,3,24);
De=zeros(301,3,24);
for k=1:24
    S=xlsread('D:\数模国赛\数据\供应量.xlsx',k);
    for i=1:size(S,1)
        for j=1:size(S,2)
            Su(i,j,k)=S(i,j);
        end
    end
    D=xlsread('D:\数模国赛\数据\需求量.xlsx',k);
    for i=1:size(D,1)
        for j=1:size(D,2)
            De(i,j,k)=D(i,j);
        end
    end
    % 求出每个供应点和需求点间距离
    for k=1:24
        map(:,:,k)=get_map(Su(:,:,k),De(:,:,k));
    end
    for k=1:24
        xlswrite('D:\数模国赛\数据\距离矩阵.xlsx',map(:,:,k),k)
    end
    % 求出 5 公里内的距离
    for k=1:size(map,3)
        map_k=zeros(301,301);
        for j=1:size(map,2)
            for i=1:size(map,1)
                if map(i,j,k)<=5 && map(i,j,k)~=0
                    map_k(i,j)=map(i,j,k);
                end
            end
        end
        xlswrite('D:\数模国赛\数据\可行距离矩阵.xlsx',map_k,k)
    end
    % 求出各供应点前往需求点的概率
    for k=1:24
        map_kr=xlsread('D:\数模国赛\数据\可行距离矩阵.xlsx',k);
        su=zeros(size(map_kr,1));
        p=zeros(size(map_kr,1),size(map_kr,2));
        for i=1:size(map_kr,1)
            for j=1:size(map_kr,2)
                if map_kr(i,j)~=0
                    p(i,j)=De(j,3,k)/map_kr(i,j);
                    su(i)=su(i)+p(i,j);
                end
            end
        end
        for i=1:size(p,1)
            if su(i)~=0
                p(i,:)=p(i,:)/su(i);
            end
        end
        xlswrite('D:\数模国赛\数据\概率矩阵.xlsx',p,k);
    end
end

```

```

%每个需求点的供求比和平均等待时间
for k=1:24
    map_kr0=xlsread('D:\数模国赛\数据\可行距离矩阵.xlsx',k);
    p0=xlsread('D:\数模国赛\数据\概率矩阵.xlsx',k);
    map_kr=zeros(301,301);
    p=zeros(301,301);
    fori=1:size(map_kr0,1)
        for j=1:size(map_kr0,2)
            map_kr(i,j)=map_kr0(i,j);
        end
    end
    fori=1:size(p0,1)
        for j=1:size(p0,2)
            p(i,j)=p0(i,j);
        end
    end
    jiedan=zeros(1,size(p,2));
    t=zeros(1,size(p,2));
    b=zeros(1,size(p,2));
    for j=1:length(jiedan)
        b(j)=sum(p(:,j).*Su(:,3,k))/De(j,3,k);
        jiedan(j)=sum(p(:,j).*Su(:,3,k).*map_kr(:,j));
        t(j)=jiedan(j)./sum(p(:,j).*Su(:,3,k));
    end
    xlswrite('D:\数模国赛\数据\供求比.xlsx',b,k);
    xlswrite('D:\数模国赛\数据\平均等待时间.xlsx',t,k);
end

%求得距离矩阵
function f=get_map(Su,De)
f=zeros(size(Su,1),size(De,1));
ke=111.31955;
fori=1:size(f,1)
    for j=1:size(f,2)
        if sum(Su(i,:)==zeros(1,3))||sum(De(j,:)==zeros(1,3))
            f(i,j)=0;
        else
            f(i,j)=abs((De(j,1)-Su(i,1))*ke*cos(Su(i,2)/pi))+abs(De(j,2)-Su(i,2))*ke;
        end
    end
end
end
3.mohu.m
%模糊评价
clear all
clc
gq=xlsread('D:\数模国赛\数据\供求比.xlsx');
for k=1:24
    ls=gqlishudu(gq(:,k),0.5,3,3.5,6);
    xlswrite('D:\数模国赛\数据\供求比隶属度.xlsx',ls,k);
end

sj=xlsread('D:\数模国赛\数据\平均等待时间.xlsx');
fori=1:size(sj,1)
    for j=1:size(sj,2)
        if isnan(sj(i,j))==1
            sj(i,j)=inf;
        end
    end
end

```



```

end
end
end
for k=1:24
ls=sjlishudu(sj(:,k),0.9,1.7,1.8,3);
xlswrite('D:\数模国赛\数据\平均等待时间隶属度.xlsx',ls,k);
end
for k=1:24
    ls1=gqlishudu(gq(:,k),0.5,3,3.5,6);
    ls2=sjlishudu(sj(:,k),0.9,1.7,1.8,3);
ls_m=max(ls1,ls2);
xlswrite('D:\数模国赛\数据\总隶属度.xlsx',ls_m,k);
end
F=zeros(size(ls_m,1),2,24);
for k=1:24
    ls1=gqlishudu(gq(:,k),0.5,3,3.5,6);
    ls2=sjlishudu(sj(:,k),0.9,1.7,1.8,3);
ls_m=max(ls1,ls2);
fori=1:size(ls_m,1)
    [F(i,1,k),F(i,2,k)]=max(ls_m(i,:));
end
xlswrite('D:\数模国赛\数据\模糊评价结果.xlsx',F(:, :,k),k);
end
for k=1:24
jw=xlsread('D:\数模国赛\数据\需求量.xlsx',k);
subplot(4,6,k);
fori=1:301
if F(i,2,k)==1
sy='r*';
end
if F(i,2,k)==2
sy='g*';
end
if F(i,2,k)==3
sy='b*';
end
ifjw(i,1)~=0
scatter(jw(i,1),jw(i,2),sy);
hold on
end
end
end

%供求比的隶属度函数
function f=gqlishudu(X,a,b,c,d)
f=zeros(length(X),3);
fori=1:length(X)
if X(i)<=a
f(i,1)=0;
f(i,2)=0;
f(i,3)=1;
end
if X(i)>a&&X(i)<=b
f(i,1)=(X(i)-a)/(d-a);
f(i,2)=(X(i)-a)/(b-a);
f(i,3)=(d-X(i))/(d-a);
end
end

```

```

if X(i)>b&&X(i)<=c
f(i,1)=(X(i)-a)/(d-a);
f(i,2)=1;
f(i,3)=(d-X(i))/(d-a);
end
if X(i)>c&&X(i)<=d
f(i,1)=(X(i)-a)/(d-a);
f(i,2)=(d-X(i))/(d-c);
f(i,3)=(d-X(i))/(d-a);
end
if X(i)>d
f(i,1)=1;
f(i,2)=0;
f(i,3)=0;
end
end

```

%平均等待时间的隶属度函数

```

function f=sjlishudu(X,a,b,c,d)
f=zeros(length(X),3);
for i=1:length(X)
if X(i)<=a
f(i,1)=1;
f(i,2)=0;
f(i,3)=0;
end
if X(i)>a&&X(i)<=b
f(i,1)=(d-X(i))/(d-a);
f(i,2)=(X(i)-a)/(b-a);
f(i,3)=(X(i)-a)/(d-a);
end
if X(i)>b&&X(i)<=c
f(i,1)=(d-X(i))/(d-a);
f(i,2)=1;
f(i,3)=(X(i)-a)/(d-a);
end
if X(i)>c&&X(i)<=d
f(i,1)=(d-X(i))/(d-a);
f(i,2)=(d-X(i))/(d-c);
f(i,3)=(X(i)-a)/(d-a);
end
if X(i)>d
f(i,1)=0;
f(i,2)=0;
f(i,3)=1;
end
end

```

4.wfwc1.m

%求出补贴对应供应量增加率

```

clear all
clc
gyz=zeros(24,1);
for k=1:24
gy=xlsread('D:\数模国赛\数据\供应量.xlsx',k);
gyz(k)=sum(gy(:,3));
end

```

```

i0=gyz/12115;
i=zeros(24,8);
for k=1:24
for t=1:8
if t>=1&&t<=6
omg=10;
dt=t;
end
if t==7
omg=2;
i0(k)=i(k,6);
dt=1;
end
if t==8
omg=0;
i0(k)=i(k,7);
dt=1;
end
lamda=0.1+0.01*omg;
i(k,t)=1/(1+(1/i0(k)-1)*exp((-1)*lamda*dt));
end
end

```

5.wfwc2.m

% 求出补贴对应需求量增加率

```

clear all
clc
xqz=zeros(24,1);
for k=1:24
xq=xlsread('D:\数模国赛\数据\需求量.xlsx',k);
xqz(k)=sum(xq(:,3));
end
i0=xqz/(12115*65);
i=zeros(24,8);
for k=1:24
for t=1:8
if t==1
omg=10;
dt=1;
end
if t==2
omg=16;
i0(k)=i(k,1);
dt=1;
end
if t==3
omg=10;
i0(k)=i(k,2);
dt=1;
end
if t==4
omg=4;
i0(k)=i(k,3);
dt=1;
end
if t>=5&&t<=8
omg=0;

```

```

i0(k)=i(k,4);
dt=t-4;
end
lamda=0.1+0.01*omg;
i(k,t)=1/(1+(1/i0(k)-1)*exp((-1)*lamda*dt));
end
end

```

6.butie.m

%求出 8 个月补贴对于打车难问题的影响变化

```

clear all
clc
ig=xlsread('D:\数模国赛\数据\i(t).xlsx',1);
iq=xlsread('D:\数模国赛\数据\i(t).xlsx',2);
ige=[ig(1,:);ig(15,:);ig(19,:)];
iqe=[iq(1,:);iq(15,:);iq(19,:)];
gy=zeros(301,3,3);
xq=zeros(301,3,3);
gy(:,1)=xlsread('D:\数模国赛\数据\供应量.xlsx',1);
gy(:,2)=xlsread('D:\数模国赛\数据\供应量.xlsx',15);
gy(:,3)=xlsread('D:\数模国赛\数据\供应量.xlsx',19);
xq(:,1)=xlsread('D:\数模国赛\数据\需求量.xlsx',1);
xq(:,2)=xlsread('D:\数模国赛\数据\需求量.xlsx',15);
xq(:,3)=xlsread('D:\数模国赛\数据\需求量.xlsx',19);
gq=zeros(301,2,3,8);
for j=1:2
for t=1:8
for k=1:3
for i=1:301
gq(i,k,t,j)=(j==1)*(gy(i,3,k)+12115*ige(k,t))+(j==2)*(xq(i,3,k)+12115*65*iqe(k,t));
end
end
end
end
for t=1:8
xlswrite('D:\数模国赛\数据\补贴后供应量.xlsx',gq(:, :, t, 1), t);
xlswrite('D:\数模国赛\数据\补贴后需求量.xlsx',gq(:, :, t, 2), t);
end
n=1;
for t=1:8
for k=1:3
map_k=xlsread('D:\数模国赛\数据\距离矩阵.xlsx',tr(k));
for j=1:size(map_k,2)
for i=1:size(map_k,1)
ifmap_k(i,j)>radius(t)
map_k(i,j)=0;
end
end
end
xlswrite('D:\数模国赛\数据\补贴后可行距离矩阵.xlsx',map_k,n)
n=n+1;
end
end
for k=1:24
map_kr=xlsread('D:\数模国赛\数据\补贴后可行距离矩阵.xlsx',k);

```

```

su=zeros(size(map_kr,1));
p=zeros(size(map_kr,1),size(map_kr,2));
fori=1:size(map_kr,1)
for j=1:size(map_kr,2)
ifmap_kr(i,j)~=0
p(i,j)=gq(j,mo_d(k),ceil(k/3),2)/map_kr(i,j);
su(i)=su(i)+p(i,j);
end
end
end
fori=1:size(p,1)
ifsu(i)~=0
p(i,:)=p(i,:)/su(i);
end
end
xlswrite('D:\数模国赛\数据\补贴后概率矩阵.xlsx',p,k);
end
for k=1:24
map_kr0=xlsread('D:\数模国赛\数据\补贴后可行距离矩阵.xlsx',k);
p0=xlsread('D:\数模国赛\数据\补贴后概率矩阵.xlsx',k);
map_kr=zeros(301,301);
p=zeros(301,301);
fori=1:size(map_kr0,1)
for j=1:size(map_kr0,2)
map_kr(i,j)=map_kr0(i,j);
end
end
fori=1:size(p0,1)
for j=1:size(p0,2)
p(i,j)=p0(i,j);
end
end
jiedan=zeros(1,size(p,2));
t=zeros(1,size(p,2));
b=zeros(1,size(p,2));
for j=1:length(jiedan)
b(j)=sum(p(:,j).*gq(:,mo_d(k),ceil(k/3),1))/gq(j,mo_d(k),ceil(k/3),2);
jiedan(j)=sum(p(:,j).*gq(:,mo_d(k),ceil(k/3),1).*map_kr(:,j));
t(j)=jiedan(j)./sum(p(:,j).*gq(:,mo_d(k),ceil(k/3),1));
end
xlswrite('D:\数模国赛\数据\补贴后供求比.xlsx',b,k);
xlswrite('D:\数模国赛\数据\补贴后平均等待时间.xlsx',t,k);
end
for k=1:3
A=zeros(8,301);
B=zeros(8,301);
for t=1:8
A0=xlsread('D:\数模国赛\数据\补贴后供求比.xlsx',k+3*(t-1));
B0=xlsread('D:\数模国赛\数据\补贴后平均等待时间.xlsx',k+3*(t-1));
fori=1:length(A0)
A(t,i)=A0(i);
end
fori=1:length(B0)
B(t,i)=B0(i);
end
end
end

```

```

xlswrite('D:\数模国赛\数据\补贴后供求比(整合版) .xlsx',A,k);
xlswrite('D:\数模国赛\数据\补贴后平均等待时间(整合版) .xlsx',B,k);
end
S=zeros(8,1,3);
for k=1:3
    A0=xlswrite('D:\数模国赛\数据\补贴后供求比(整合版) .xlsx',k);
    B0=xlswrite('D:\数模国赛\数据\补贴后平均等待时间(整合版) .xlsx',k);
    A0=A0';
    B0=B0';
    A=A0;
    B=zeros(size(B0,1),size(B0,2));
    for i=1:size(B0,1)
        for j=1:size(B0,2)
            if B0(i,j)==0||isnan(B0(i,j))==1
                B(i,j)=0;
            else
                B(i,j)=1/B0(i,j);
            end
        end
    end
    a=0;
    b=0;
    for i=1:size(B,1)
        if A(i,1)==0
            a=a+1;
        end
        if B(i,1)==0
            b=b+1;
        end
    end
    subplot(1,3,k);

    for j=1:size(B,2)
        S(j,1,k)=(sum(A(:,j))+sum(B(:,j)))/(602-a-b)/2;
    end
    plot(1:8,S(:,1,k),'r');
    hold on
end
for k=1:3
    xlswrite('D:\数模国赛\数据\S.xlsx',S(:,k),k);
end

function f=radius(t)
    if t>=1&&t<=6
        f=10;
    end
    if t==7
        f=6;
    end
    if t==8
        f=5;
    end
end

function f=tr(k)
    if k==1
        f=1;
    end
end

```

```

if k==2
    f=15;
end
if k==3
    f=19;
end

```

```

function f=mo_d(k)
if mod(k,3)==0
    f=3;
else
    f=mod(k,3);
end

```

7.fangan.m

%遗传算法求解最优补贴方案

```

clear all
clc
close all
global popsize;
popsize=50;
pcrossover=0.8;
pmutation=0.01;
generationmax=50;
blength=36;
population=round(rand(popsize,blength));
[fit,cum]=fitfun(population);
for i=1:generationmax
    for j=1:2:popsize
        seln=gaselect(population,cum);
        scro=gacrossover(population,seln,pcrossover);
        scnew(j,:)=scro(1,:);
        scnew(j+1,:)=scro(2,:);
        smnew(j,:)=mutation(scnew(j,:),pmutation);
        smnew(j+1,:)=mutation(scnew(j+1,:),pmutation);
    end
    population=smnew;
    [fit,cum]=fitfun(population);
    [fmax,nmax]=max(fit);
    fmean=mean(fit);
    ymax(i)=fmax;
    ymean(i)=fmean;
    greatpop(i,:)=population(nmax,:);
end

```

%适应度函数

```

function [Fitvalue,cumsump]=fitfun(population)
global popsize
for i=1:popsize

```

```

    x1=[transform2to10(population(i,1:4)),transform2to10(population(i,5:8)),transform2to10(population(i,9:12))];

```

```

    x2=[transform2to10(population(i,13:16)),transform2to10(population(i,17:20)),transform2to10(population(i,21:24)),transform2to10(population(i,25:28)),transform2to10(population(i,29:32)),transform2to10(population(i,32:36))];
    Fitvalue(i)=targetfun1(x1)+targetfun2(x2)+targetfun3(x1,x2);

```

```

end
Pperpopulation=Fitvalue/sum(Fitvalue);
cumsump(1)=Pperpopulation(1);
fori=2:popsiz
cumsump(i)=cumsump(i-1)+Pperpopulation(i);
end

%三个目标函数
function f=targetfun1(x)
i00=0.0014;
for t=1:5
if t==1
omg=10;
i0=i00;
dt=1;
end
if t==2
omg=x(1);
i0=ip(1);
dt=1;
end
if t==3
omg=x(2);
i0=ip(2);
dt=1;
end
if t==4
omg=x(3);
i0=ip(3);
dt=1;
end
if t==5
omg=0;
i0=ip(4);
dt=1;
end
lamda=0.1+0.01*omg;
ip(t)=1/(1+(1/i0-1)*exp((-1)*lamda*dt));
end
f=(ip(5)-i00)/((ip(1)-i00)*10+(ip(2)-ip(1))*x(1)+(ip(3)-ip(2))*x(2)+(ip(4)-ip(3))*x(3));

function f=targetfun2(x)
i00=0.2064;
for t=1:8
if t==1
omg=10;
i0=i00;
dt=1;
end
if t==2
omg=x(1);
i0=ip(1);
dt=1;
end
if t==3
omg=x(2);
i0=ip(2);

```



```

dt=1;
end
if t==4
omg=x(3);
    i0=ip(3);

dt=1;
end
if t==5
omg=x(4);
    i0=ip(4);

dt=1;
end
if t==6
omg=x(5);
    i0=ip(5);

dt=1;
end
if t==7
omg=x(6);
    i0=ip(6);

dt=1;
end
if t==8
omg=0;
    i0=ip(7);

dt=1;
end
lamda=0.1+0.01*omg;
ip(t)=1/(1+(1/i0-1)*exp((-1)*lamda*dt));
end

f=(ip(8)-i00)/((ip(1)-i00)*10+(ip(2)-ip(1))*x(1)+(ip(3)-ip(2))*x(2)+(ip(4)-ip(3))*x(3)+(ip(5)-ip(4))*x(4)
+(ip(6)-ip(5))*x(5)+(ip(7)-ip(6))*x(6));

function f=targetfun3(x1,x2)
loadmap_kr.mat
loadp.mat
i001=0.0014;
for t=1:5
if t==1
omg=10;
    i0=i001;

dt=1;
end
if t==2
omg=x1(1);
    i0=ip1(1);

dt=1;
end
if t==3
omg=x1(2);
    i0=ip1(2);

dt=1;
end
if t==4
omg=x1(3);
    i0=ip1(3);

```

```

dt=1;
end
if t==5
omg=0;
            i0=ip1(4);
dt=1;
end
lamda=0.1+0.01*omg;
ip1(t)=1/(1+(1/i0-1)*exp((-1)*lamda*dt));
end
ip1(6)=ip1(5);
ip1(7)=ip1(5);
ip1(8)=ip1(5);
i002=0.2064;
for t=1:8
if t==1
omg=10;
            i0=i002;
dt=1;
end
if t==2
omg=x2(1);
            i0=ip2(1);
dt=1;
end
if t==3
omg=x2(2);
            i0=ip2(2);
dt=1;
end
if t==4
omg=x2(3);
            i0=ip2(3);
dt=1;
end
if t==5
omg=x2(4);
            i0=ip2(4);
dt=1;
end
if t==6
omg=x2(5);
            i0=ip2(5);
dt=1;
end
if t==7
omg=x2(6);
            i0=ip2(6);
dt=1;
end
if t==8
omg=0;
            i0=ip2(7);
dt=1;
end
lamda=0.1+0.01*omg;
ip2(t)=1/(1+(1/i0-1)*exp((-1)*lamda*dt));

```

```

end
loadgy.mat
loadxq.mat
gq=zeros(301,24,8,2);
for j=1:2
for t=1:8
for k=1:24
fori=1:301
                                gq(i,k,t,j)=(j==1)*(gy(i,3,k)+12115*ip2(t))+(j==2)*(xq(i,3,k)+12115*65*ip1(t));
end
end
end
end
    B=zeros(8,301);
    T=zeros(8,301);
for t=1:8

for k=1:24
t_w=zeros(1,size(p,2));
    b=zeros(1,size(p,2));
jiedan=zeros(1,size(p,2));
for j=1:length(jiedan)
    b(j)=sum(p(:,j).*gq(:,k,t,1))/gq(j,k,t,2);
jiedan(j)=sum(p(:,j).*gq(:,k,t,1).*map_kr(:,j));
t_w(j)=jiedan(j)./sum(p(:,j).*gq(:,k,t,1));
ifisnan(b(j))==1
b(j)=0;
end
ifisnan(t_w(j))==1
t_w(j)=0;
end
end
B(t,:)=B(t,:)+b;
T(t,:)=T(t,:)+t_w;
end
end
B=B./24;
T=T./24;
B=B';
T=T';
fori=1:length(T)
if T(i)==0&&isnan(T(i))==1
T(i)=0;
else
T(i)=1/T(i);
end
end
fori=1:301
for j=1:8
if B(i,j)==0&&isnan(B(i,j))==1
B(i,j)=0;
end
if T(i,j)==0&&isnan(T(i,j))==1
T(i,j)=0;
end
if B(i,j)==inf
B(i,j)=0;

```

```

end
if T(i,j)==inf
T(i,j)=0;
end
end
end
S=(sum(B)+sum(T))/602;
f=sum(S);

%选择
function seln=gselect(x,p)
for i=1:2
    r=rand;
    pr=p-r;
    j=1;
    while pr(j)<0
        j=j+1;
    end
    seln(i)=j;
end

%交叉
function scro=gacrossover(population,seln,pc)
blength=size(population,2);
if ifc(pc)==1
    chb=round(rand*(blength-2))+1;
    scro(1,:)=[population(seln(1),1:chb) population(seln(2),chb+1:blength)];
    scro(2,:)=[population(seln(2),1:chb) population(seln(1),chb+1:blength)];
else
    scro(1,:)=population(seln(1),:);
    scro(2,:)=population(seln(2),:);
end

%变异
function snnew=mutation(snew,pmutation);
BitLength=size(snew,2);
snnew=snew;
pmm=ifc(pmutation);
if pmm==1
    chb=round(rand*(BitLength-1))+1;
    snnew(chb)=abs(snew(chb)-1);
end

%判断是否交叉或变异
function f=ifc(p)
test(1,100)=0;
l=round(100*p);
test(1:l)=1;
n=round(rand*99)+1;
f=test(n);

%二进制转化为十进制
function x=transform2to10(Population);
BitLength=size(Population,2);
x=Population(BitLength);
for i=1:BitLength-1
    x=x+Population(BitLength-i)*power(2,1);
end

```

end

附录二 表格

表 4 各乘客需求点处“供需比”的隶属度函数（部分）

时间	匹配程度高	匹配程度中等	匹配程度低
0:00	1	0	0
	⋮	⋮	⋮
	0.830044	0.373903	0.169956
1:00	0.48373	1	0.51627
	⋮	⋮	⋮
	0.551801	0.986037	0.448199
⋮	⋮	⋮	⋮
23:00	0.412205	0.906852	0.587795
	⋮	⋮	⋮
	0.392205	0.862852	0.607795

表 5 各乘客需求点处乘客等待时间的隶属度函数（部分）

时间	匹配程度高	匹配程度中等	匹配程度低
0:00	0.724266	0.723801	0.275734
	⋮	⋮	⋮
	0.730878	0.706445	0.269122
1:00	0.447467	0.783067	0.552533
	⋮	⋮	⋮
	0.546378	0.956161	0.453622
⋮	⋮	⋮	⋮
23:00	0.568017	0.994031	0.431983
	⋮	⋮	⋮
	0.686491	0.822961	0.313509

表 10 “滴滴打车”公司的补贴方案

对 象	月 份 金 额(元)	1	2	3	4	5	6	7	8
乘客		10	10-15	6-15	3-5	0	0	0	0
出租车		10	10	10	10	10	10	2	0

### 附录三 图像

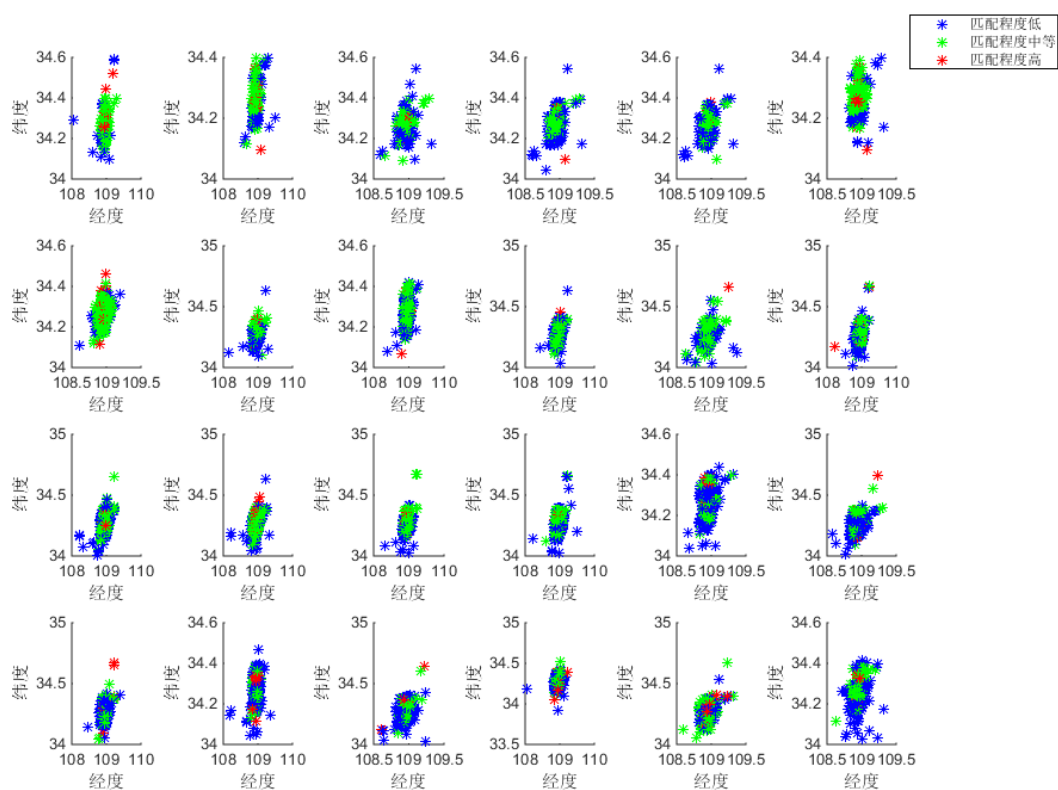


图 2 西安市不同地区出租车资源的“供求匹配”程度级别图