**2015高教社杯全国大学生数学建模竞赛**

**承 诺 书**

我们仔细阅读了中国大学生数学建模竞赛的竞赛规则.

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人（包括指导教师）研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛规则的, 如果引用别人的成果或其他公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们郑重承诺，严格遵守竞赛规则，以保证竞赛的公正、公平性。如有违反竞赛规则的行为，我们将受到严肃处理。

我们参赛选择的题号是（从A/B/C/D中选择一项填写）： B

我们的参赛报名号为（如果赛区设置报名号的话）：

所属学校（请填写完整的全名）： 江苏科技大学

参赛队员 (打印并签名) ：1. 刘硕

2. 令狐岩松

3. 余骁

指导教师或指导教师组负责人 (打印并签名)： 王承毅

日期： 2020 年7 月22 日

赛区评阅编号（由赛区组委会评阅前进行编号）：

**2015高教社杯全国大学生数学建模竞赛**

**编 号 专 用 页**

赛区评阅编号（由赛区组委会评阅前进行编号）：

赛区评阅记录（可供赛区评阅时使用）：

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 评  阅  人 |  |  |  |  |  |  |  |  |  |  |
| 评  分 |  |  |  |  |  |  |  |  |  |  |
| 备  注 |  |  |  |  |  |  |  |  |  |  |

全国统一编号（由赛区组委会送交全国前编号）：

全国评阅编号（由全国组委会评阅前进行编号）：

“互联网+”时代的出租车资源配置

**摘要**

本文通过研究“互联网+”时代的出租车资源配置，分别调查了北京、上海、广州等八个城市的基本属性和出租车行业信息，在不同时空上分析了供求匹配程度。接着从两家公司的出租车补贴方案上设立指标，解释了是否缓解了打车难的问题。最后建立模型和相应的优化算法设计从不同角度和情况下给出多个新打车平台的补贴方案。

**针对问题一**，在广泛讨论影响出租车资源供求水平的因素后，收集不同时空的城市基本属性，通过**主成分分析**，从复杂因子中找出关系并计算出全国八座重点城市出租车行业的供给水平和需求水平，并建立科学的**供求匹配函数**来综合求出各个城市不同时空下的**供求匹配指标。**从空间来分析，在全国八座重点城市中西安市的“供求匹配”**程度最高为0.9965**而北京市的“供求匹配”**程度最低为0.5842**；从时间来分析，在2009年~2014年中2013年出租车“供求匹配”**程度最高，高于99%**而2014年“供求匹配”**程度最低且低于85%。**

**针对问题二：**首先列出且量化了2014年上半年滴滴和快的两家公司的补贴金额政策的变化记录，在问题一模型的基础上**增加新的因子**作为重要参量，建立起**缓解率指标模型**并以时间为轴分析了两家公司在补贴政策实行过程中对各个城市“打车难”问题的缓解程度和效果，并且与没有补贴政策实行的情况下作对比最终得出结论：最终得出结论：**两家公司的补贴政策整体上确实对“缓解打车难”有帮助，但是帮助并不大，且对于各个城市的缓解程度不均衡，甚至加深了某些城市的问题。**

**针对问题三：**延伸分析了问题二中两家公司出现的一些问题，以**因地制宜的补贴政策**为主，针对这些矛盾设计出了可行的**组合优化模型**，采用经过细节修改过的遗传算法对模型进行求解，得出了**补贴金额尽量少，收益尽量大，且所有城市都可以“缓解打车难”**前提下的八座城市的**最优补贴金额方案**。接着在对打车软件平台用户信息进行分析和拟合后，使用**logistic模型**和拟合出来的参数，结合补贴金额方案，给出了**不同时间段下具体的补贴方案**，并根据不同时间段的特点对补贴方案的合理性进行了论述。

综合来说，本模型结合了大量的因子，找出了不同城市的供求匹配程度的关系，并以此为基础从缓解率、补贴方案、用户增长等较为全面的分析了当下重点城市“互联网+”出租车资源配置的情况，且**本模型的优势在于可以轻松结合或删改大量的因素，并因地制宜给出不同时空下的最佳补贴方案**。

**关键词：主成分分析、供求匹配指标、组合优化模型、遗传算法、logistic模型**

### 一、问题重述

人们在城市的出行过程中，出租车是重要的交通工具之一，但随着城市人口密度和各行各业汽车密度的增加，对交通运行带来了不便，所以即便一个城市的出租车的供应量大于需求量，也会出现打车难的问题。为了缓解这个问题，加上“互联网+”时代的到来，有多家公司比如滴滴、快的等依托移动互联网建立了打车软件服务平台，乘客可以事先预约打车，司机可以根据预约地点和当时所处位置，进行信息互通，为了争取用户使用，各家公司推出了不同的补贴方案。问题要求搜集相关数据，完成如下问题：

**问题一：**根据搜集的数据，构造匹配指标，并分析在不同时间和不同空间的出租车资源的“供求匹配”程度；

**问题二：**分析滴滴、快的等打车软件公司的出租车补贴方案是否对“缓解打车难”有帮助，并得出结论；

**问题三：**根据问题（1）和（2）的结果，设计合理的补贴方案，并论证给出方案的合理性，提出建议。

### 二、问题分析

出租车是市民出行的重要交通工具之一，“打车难”是人们关注的一个社会热点问题。随着“互联网+”时代的到来，有多家公司依托移动互联网建立了打车软件服务平台，实现了乘客与出租车司机之间的信息互通，因此研究“互联网+”时代打车软件公司补贴方案问题显得尤为重要。

### 2.1 问题一分析

问题一要求我们试建立合理的指标，并分析不同时空出租车资源的“供求匹配”程度。首先要确定有关“供求匹配”的相关指标。在广泛讨论影响出租车资源供求水平的因素后，收集不同时空的城市基本属性。然后选用主成分分析法，从复杂因子中找出关系并计算出全国不同重点城市出租车行业的供给水平和需求水平，并建立科学的供求匹配函数来综合求出各个城市不同时空下的供求匹配指标。

### 2.2 问题二分析

问题二要求我们分析各公司的出租车补贴方案是否对“缓解打车难”有帮助。首先不同公司的补贴方案不同，不同时期的补贴方案也不同，因此需要先搜集主要打车公司的补贴方案。然后列出且量化了2014年上半年滴滴和快的两家公司的补贴金额政策的变化记录，在问题一模型的基础上增加新的因子作为重要参量，建立缓解率指标模型并以时间为轴分析了两家公司在补贴政策实行过程中对各个城市“打车难”问题的缓解程度和效果，最终得出结论。

### 2.3 问题三分析

问题三要求我们设计一种补贴方案，并论证其合理性。首先根据问题二可以发现现有公司补贴方案的弊端，因此在现有基础之上延伸分析了问题二中两家公司出现的一些问题，以因地制宜的补贴政策为主，针对这些矛盾设计出了可行的组合优化模型，算法上采用经过细节修改过的遗传算法对模型进行求解，可以得出补贴金额尽量少，收益尽量大，且所有城市都可以“缓解打车难”前提下的八座城市的最优补贴金额方案。在对打车软件平台用户信息进行分析和拟合后，使用logistic模型和拟合出来的参数，结合补贴金额方案，可以给出不同时间段下具体的补贴方案，并根据不同时间段的特点对补贴方案的合理性进行论述。

### 三、模型假设与约定

1）假设短时间内的同一个社区模型的社会环境系统是不变的；

2）假设短时间内的同一个出租车交通系统的出租车总数量是不变的；

3）假设不考虑天气、突发事件等非人为可控因素的影响；

4）假设不考虑出租车行业内部不正当竞争等隐形因素的影响；

5）假设不考虑城市道路拥堵等客观因素的影响

### 四、符号说明及名词定义

|  |  |
| --- | --- |
| 符号 | 符号说明 |
|  | 协方差矩阵 |
|  | 出租车资源供给水平 |
|  | 出租车资源需求水平 |
|  | 出租车资源供给水平对需求水平的匹配值 |
|  | 出租车资源需求水平对供给水平的匹配值 |
|  | 缓解率 |
|  | 没有补贴时加上新因素后计算的匹配程度 |
|  | 添加某一补贴金额时计算的匹配程度 |
|  | 组合效应 |
|  | 补贴金额向量 |
|  | 相应分配给城市的每单补贴金额 |
|  | 相应城市计算后的缓解率 |
|  | 平均缓解率 |
|  | 种群中第个染色体的适应度值 |

### 五、问题一模型建立与求解

### 5.1 模型准备

出租车资源的供给水平与需求水平都有各自的影响因素，两者之间还存在复杂的内部关系[1]。本文从供给与需求两方面分别筛选我国出租车资源供求的主要影响因素，建立出租车资源供求的指标体系。本文选取指标遵循了有效性原则、可行性原则、科学性原则以及实用性原则，综合分析选定了起步价、城市面积、市辖区生产总值、市辖区公共汽（电）车的运营数量、出租车月营业额、市辖区出租车的运营数量这六个指标作为出租车资源供给水平的指标以及选定了起步价、城市面积、市辖区生产总值、市辖区公共汽（电）车的运营数量、出租车月营业额、市辖区出租车的运营数量、私家车的数量、城市人口数量、市辖区道路面积这九个指标作为出租车资源需求水平的指标。由于在供给水平指标和需求水平指标中存在一些相关性较强的因素，因此为了方便分析本文采用主成分分析对指标信息进行提取[2]。

### 5.2 供求匹配程度模型

### 5.2.1 主成分分析模型原理

主成分分析(Principal Components Analysis，PCA)也称为主分量分析，是一种对高维数据进行降维从而达到简化数据结构的方法，因此可将原始多数具有一定线性关系的指标进行降维重新组合成少数互不相关的综合指标，而这少数综合指标可以反映原始多数指标的关键信息[3]。

### 5.2.2 主成分分析模型建立步骤

要分析出租车资源的“供求匹配”程度，本文分别从六个供给水平指标和九个需求水平指标建立主成分分析模型。通过建立主成分分析模型消除原始多数指标的一定线性关系[4]，主要步骤如下：

**Step1：**为个原始指标，记协方差矩阵为A。

**Step2：**为组合成少数互不相关的综合指标，建立原始指标的线性组合，其数学模型为

 (1)

其中，。

**Step3：**确定主成分：找寻一个使协方差矩阵对角化的正交矩阵

 (2)

得：

**Step4：**选取主成分

 （3）

### 5.2.3 供求匹配程度模型建立

本文基于文献[]对两系统状态协调度函数的定义，对出租车资源供给与需求匹配程度进行计算。用和分别表示出租车资源的供给水平和需求水平，建立相应的匹配程度函数如下：

 (4)

 (5)

其中，为出租车资源供给水平对需求水平的匹配值，为出租车资源需求水平对供给水平的匹配值。因为实际中这两个变量不可能达到完全匹配，所以在计算时将、分别取0.8和0.8。和分别表示出租车资源需求水平和供给水平的方差。为需求水平对供给水平的匹配程度，为出租车资源供给水平对需求水平的匹配程度。因此，本文利用相关统计指标计算将出租车资源供求匹配程度定义为，具体如下计算:

 (6)

### 5.2.4模型求解

通过分析我国主要城市在相同年份的出租车资源“供求匹配”程度与以西安市为例分析同城市在不同年份的出租车资源“供求匹配”程度，从地域和时间两个维度对我国出租车资源利用及优化配置情况进行实证分析与比较说明。

**（一）在空间的出租车资源“供求匹配”程度分析**

根据选取的我国出租车资源供给水平和需求水平的指标，基于数据可得性与不同规模城市的代表性，根据中国国家统计局和中国经济社会大数据研究平台等收集整理了我国8个主要城市2015年的出租车资源供给水平与需求水平的指标数据，运用MATLAB软件进行主成分分析。

* 供给水平指标的分析

首先对供给水平数据进行正态分布检验，并将数据标准化处理之后，计算得各个指标之间的皮尔逊相关系数，得出各个指标之间的相关系数矩阵。从矩阵中可以发现，某些指标之间具有很大的相关性，如直接作为影响供给水平的因素，不仅数据集不易使用，而且造成关键信息的交叉，失去客观性。主成分分析通过对高维数据进行降维，从而简化数据结构。经过计算，相关系数矩阵的特征值、相应的特征向量以及贡献率如下表：

表5.2.4-1：供给水平指标相关系数矩阵



表5.2.4-2：供给水平指标特征值与累计贡献率

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 特征向量矩阵 | a1 | a2 | a3 | a4 | a5 | a6 |
| x1 | 0.4529 | -0.4073 | 0.1010 | -0.5071 | 0.0533 | 0.5990 |
| x2 | -0.1835 | 0.0956 | 0.9597 | 0.0645 | -0.1415 | 0.1091 |
| x3 | 0.5263 | 0.0717 | -0.0359 | 0.2533 | -0.8057 | -0.0571 |
| x4 | 0.4534 | 0.4000 | 0.0423 | 0.5406 | 0.4759 | 0.3374 |
| x5 | 0.2805 | -0.7048 | 0.1904 | 0.3376 | 0.2507 | -0.4599 |
| x6 | 0.4472 | 0.4039 | 0.1714 | -0.5179 | 0.1964 | -0.5484 |
| 特征值 | 3.4581 | 1.3405 | 0.9443 | 0.2256 | 0.0300 | 0.0015 |
| 贡献率 | 0.5763 | 0.2234 | 0.1574 | 0.0376 | 0.0050 | 0.0003 |
| 累计贡献率 | 0.5763 | 0.7998 | 0.9571 | 0.9947 | 0.9997 | 1.0000 |

从表5.2.4-2中可以观察到前三个主成分的累计贡献率达到95.7%，这三个主成分反映了出租车资源供给水平指标的95.7%的信息，符合主成分分析的原则。另外，还可以观察到前两个主成分的累计贡献率高达79.9%，可以理解为前两个主成分可能是出租车资源供给水平的最主要指标。

在出租车资源供给水平指标中，主成分是起步价和市辖区公共汽（电）车运营数量的组合，因此主成分可以定义为出租车资源供给收益因子；主成分是市辖区生产总值和市辖区出租车运营数量的组合，因此主成分可以定义为出租车资源供给要素因子；主成分为城市面积，可以定义为出租车资源供给保障因子。

对我国8个主要城市的出租车资源供给水平进行评分，具体结果如下表：

表5.2.4-3：我国主要城市出租车资源供给水平的主成分得分

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 城市 | 主成分得分 | | | 综合得分 |
|  |  |  |  |
| 北京 | 2.1317 | 2.3774 | 0.1656 | 1.8657 |
| 成都 | -0.5429 | -0.0188 | -0.2605 | -0.3742 |
| 上海 | 3.1019 | -1.1563 | 0.0371 | 1.6040 |
| 杭州 | -0.0827 | -1.3785 | 0.1729 | -0.3431 |
| 广州 | 0.5648 | -0.5288 | -0.5509 | 0.1260 |
| 南昌 | -2.2586 | 0.3873 | -1.0900 | -1.4488 |
| 西安 | -1.2679 | 0.2416 | -0.6205 | -0.8091 |
| 哈尔滨 | -1.6462 | 0.0762 | 2.1462 | -0.6204 |

* 需求水平指标的分析

对需求水平数据进行正态分布检验，并将数据标准化处理之后，计算得各个指标之间的皮尔逊相关系数，得出各个指标之间的相关系数矩阵。从矩阵中可以发现，某些指标之间具有很大的相关性，如直接作为影响需求水平的因素，不仅数据集不易使用，而且造成关键信息的交叉，失去客观性。主成分分析通过对高维数据进行降维，从而简化数据结构。经过计算，相关系数矩阵的特征值、相应的特征向量以及贡献率如下表：

表5.2.4-4：需求水平指标相关系数矩阵



表5.2.4-5：部分需求水平指标特征值与累计贡献率

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 特征向量矩阵 | a1 | a2 | a3 | a4 | a5 | a6 |
| x1 | 0.314 | -0.4747 | 0.1469 | -0.3021 | -0.1901 | 0.3913 |
| x2 | -0.1341 | 0.1451 | 0.8993 | 0.2727 | 0.2055 | 0.0568 |
| x3 | 0.4148 | -0.0652 | -0.0258 | -0.1099 | 0.2512 | -0.5518 |
| x4 | 0.3916 | 0.2742 | 0.0104 | 0.1296 | 0.1156 | -0.3191 |
| x5 | 0.1788 | -0.6576 | 0.1794 | 0.3555 | -0.283 | -0.1953 |
| x6 | 0.3646 | 0.2426 | 0.2047 | -0.4956 | 0.0139 | 0.3423 |
| x7 | 0.3103 | 0.422 | -0.0407 | 0.3908 | -0.6866 | 0.1259 |
| x8 | 0.4059 | 0.0243 | 0.1852 | -0.1534 | -0.0014 | -0.1671 |
| x9 | 0.3635 | -0.0647 | -0.2426 | 0.5072 | 0.5417 | 0.4882 |
| 特征值 | 5.4985 | 1.6421 | 1.0161 | 0.474 | 0.2178 | 0.085 |
| 贡献率 | 0.6109 | 0.1825 | 0.1129 | 0.0527 | 0.0242 | 0.0094 |
| 累计贡献率 | 0.6109 | 0.7934 | 0.9063 | 0.959 | 0.9832 | 0.9926 |

从表5.2.4-5中可以观察到前三个主成分的累计贡献率达到90.6%，这三个主成分反映了出租车资源供给水平指标的90.6%的信息，符合主成分分析的原则。另外，还可以观察到前两个主成分的累计贡献率高达79.3%，可以理解为前两个主成分可能是出租车资源需求水平的最主要指标。

在出租车资源需求水平指标中，主成分是市辖区生产总值、市辖区公共汽（电）车运营数量、城市人口数量的组合，因此主成分可以定义为出租车资源需求收益因子；主成分是私家车数量，因此主成分可以定义为出租车资源需求要素因子；主成分为城市面积，可以定义为出租车资源需求保障因子。

对我国8个主要城市的出租车资源需求水平进行评分，具体结果如下表：

表5.2.4-6：我国主要城市出租车资源需求水平的主成分得分

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 城市 | 主成分得分 | | | 综合得分 |
|  |  |  |  |
| 北京 | 3.1936 | 2.3659 | 0.1543 | 2.6483 |
| 成都 | -0.0577 | 0.5263 | -0.2941 | 0.0304 |
| 上海 | 3.2940 | -1.7719 | 0.2826 | 1.8988 |
| 杭州 | -0.6682 | -1.2619 | 0.1688 | -0.6835 |
| 广州 | 0.7807 | -0.7702 | -0.8332 | 0.2673 |
| 南昌 | -3.2683 | 0.3153 | -0.8632 | -2.2471 |
| 西安 | -1.2261 | 0.3913 | -0.8067 | -0.8482 |
| 哈尔滨 | -2.0479 | 0.2051 | 2.1916 | -1.0661 |

根据我国8个主要城市出租车资源供给与需求水平的主成分得分，按照供求匹配程度模型计算可得各个城市出租车资源“供求匹配”程度，并且按照该匹配程度进行排序，结果如下：

图5.2.4-1：各城市“供求匹配”雷达图

表5.2.4-7：各城市“供求匹配”程度及其排序

|  |  |  |
| --- | --- | --- |
| 城市 | “供求匹配”程度 | 排序 |
| 北京 | 0.584203 | 8 |
| 成都 | 0.917531 | 4 |
| 上海 | 0.850667 | 6 |
| 杭州 | 0.961723 | 3 |
| 广州 | 0.994256 | 2 |
| 南昌 | 0.656102 | 7 |
| 西安 | 0.996538 | 1 |
| 哈尔滨 | 0.906557 | 5 |

从空间的出租车资源“供求匹配”程度分析来看，除北京市出租车资源“供求匹配”程度最低约为0.5842，不足60%，其它城市出租车资源“供求匹配”程度相对较高，高于60%。并且西安市出租车资源“供求匹配”程度最高为0.9965，广州、杭州仅次于西安的出租车资源“供求匹配”程度，排名第2和第3，且都高于95%。

**（二）在时间的出租车资源“供求匹配”程度分析—以西安市为例**

根据西安市统计年鉴和统计公报收集整理了2009~2014年西安市出租车资源供给与需求水平的指标数据。并且由于本文调查期间西安市出租车起步价没有变化，所以对于时间纵向维度西安市出租车资源“供求匹配”程度没有较大影响，而不同年份的价格指数对出租车资源供求具有显著影响，所以此处将起步价从供求匹配程度指标中剔除，用城市居民消费价格指数代替。该指标能够代表随着时间变化，人们生活收入水平以及对商品价格的可接受度，对于时间纵向维度的分析更具合理性。

依据建立的供求匹配程度模型，按照以上步骤，可以求得西安市出租车资源在供给水平和需求水平的主成分得分：

表5.2.4-8：2009~2014年西安市出租车资源供给水平的主成分得分

|  |  |  |  |
| --- | --- | --- | --- |
| 年份 | 主成分得分 | | 综合得分 |
|  |  |  |
| 2009 | 0.0193 | -0.8638 | -0.1619 |
| 2010 | -0.1213 | -0.6327 | -0.2262 |
| 2011 | 0.2361 | -0.4440 | 0.0965 |
| 2012 | 1.0526 | 0.1346 | 0.8642 |
| 2013 | 1.2929 | 0.4145 | 1.1126 |
| 2014 | 1.4288 | 0.5095 | 1.2401 |

表5.2.4-9：2009~2014年西安市出租车资源需求水平的主成分得分

|  |  |
| --- | --- |
| 年份 | 主成分综合得分（） |
| 2009 | -0.0433 |
| 2010 | 0.1804 |
| 2011 | 0.3936 |
| 2012 | 0.8537 |
| 2013 | 1.1304 |
| 2014 | 1.4238 |

根据西安市出租车资源供给与需求水平的主成分得分，按照供求匹配程度模型计算可得西安市出租车资源“供求匹配”程度，并且按照该匹配程度进行排序，结果如下：

图5.2.4-2：西安市各年份“供求匹配”雷达图

表5.2.4-10：西安市各年份“供求匹配”程度及其排序

|  |  |  |
| --- | --- | --- |
| 年份 | “供求匹配”程度 | 排序 |
| 2009 | 0.9835 | 2 |
| 2010 | 0.9294 | 5 |
| 2011 | 0.9710 | 4 |
| 2012 | 0.9778 | 3 |
| 2013 | 0.9933 | 1 |
| 2014 | 0.8426 | 6 |

从表中可以发现，根据本文分析的2009~2014年西安市出租车资源“供求匹配”程度及其排序可见，随着时间变迁，西安市出租车资源“供求匹配”程度呈现相对较为平稳的波动性变化。除了2014年“供求匹配”程度最低且低于85%，其余年份的“供求匹配”程度均高于90%，而2013年出租车“供求匹配”程度最高，高于99%。

### 六、问题二模型建立与求解

### 6.1 模型准备

为分析各公司的出租车补贴方案是否对“缓解打车难”有帮助，本研究寻求了2014年上半年滴滴和快的两家公司的补贴金额政策的变化记录。

表6.1：两家公司的补贴金额政策的变化记录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 快的出租车公司 | | | 滴滴出租车公司 | | |
| 实行日期 | **补贴金额** | **实行天数** | **实行日期** | **补贴金额** | **实行天数** |
| 1.20 | 20 | 27 | 1.10 | 20 | 37 |
| 2.17 | 19 | 1 | 2.17 | 22.5 | 1 |
| 2.18 | 21 | 14 | 2.18 | 26 | 17 |
| 3.04 | 16 | 1 | 3.07 | 21.5 | 16 |
| 3.05 | 11 | 17 | 3.23 | 14 | 25 |
| 3.22 | 10 | 56 | 5.17 | 10 | 53 |
| 2.17 | 6 | 53 | 7.09 | 2 | 34 |
| 7.09 | 2 | 31 | 8.12 | 0 | 1 |
| 8.09 | 0 | 1 |  | | |

### 6.2 模型扩展—缓解率模型建立

在问题一模型的基础上增加新的因子出租车平均里程、出租车平均单价、出租车平均单数/日作为重要指标，得出缓解率指标[5]模型：

 (7)

其中，为缓解率，为没有补贴时加上新因素后计算的匹配程度，为某一补贴金额时计算的匹配程度。

### 6.3 模型求解

**（一）无补贴方案政策**

在问题一模型的基础上增加新的因子出租车平均里程、出租车平均单价、出租车平均单数/日作为重要指标，根据问题一建立的供求匹配程度模型，可求得在没有补贴方案下各城市的出租车资源“供求匹配”程度，并且按照该匹配程度进行排序[6]，见下表：

表6.3-1：各城市“供求匹配”程度及其排序

|  |  |  |
| --- | --- | --- |
| 城市 | “供求匹配”程度 | 排序 |
| 北京 | 0.993713 | 3 |
| 成都 | 0.999035 | 1 |
| 上海 | 0.966396 | 8 |
| 杭州 | 0.992843 | 4 |
| 广州 | 0.992389 | 5 |
| 南昌 | 0.988780 | 6 |
| 西安 | 0.997814 | 2 |
| 哈尔滨 | 0.970472 | 7 |

**（二）有补贴方案政策**

可以发现，滴滴和快的两家出租车公司在不同时间段内实行了不同的补贴金额方案政策，因此本研究可以分析两家公司在不同的时间段内不同的补贴金额对各城市“供求匹配”程度的影响。根据建立的缓解率模型，分析两家公司不同的补贴金额方案政策是否对“缓解打车难”有帮助，具体缓解效果如下：

表6.3-2：滴滴出租车公司的部分缓解效果

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 补贴金额 | 20 | | 22.5 | | 26 | | 21.5 | |
| 指标 | **匹配度** | **缓解率** | **匹配度** | **缓解率** | **匹配度** | **缓解率** | **匹配度** | **缓解率** |
| 北京 | 0.9978 | 0.41% | 0.9979 | 0.43% | 0.9942 | 0.06% | 0.9954 | 0.17% |
| 成都 | 0.9987 | -0.03% | 0.9974 | -0.16% | 0.9976 | -0.13% | 0.9969 | -0.21% |
| 上海 | 0.9768 | 1.09% | 0.9729 | 0.68% | 0.9841 | 1.84% | 0.9733 | 0.72% |
| 杭州 | 0.9955 | 0.27% | 0.9985 | 0.57% | 0.9975 | 0.47% | 0.9984 | 0.56% |
| 广州 | 0.9961 | 0.37% | 0.9954 | 0.31% | 0.9975 | 0.52% | 0.9958 | 0.35% |
| 南昌 | 0.9828 | -0.60% | 0.9865 | -0.23% | 0.9878 | -0.10% | 0.9783 | -1.06% |
| 西安 | 0.9979 | 0.01% | 0.9981 | 0.03% | 0.9957 | -0.21% | 0.9958 | -0.20% |
| 哈尔滨 | 0.9913 | 2.04% | 0.9828 | 1.28% | 0.9840 | 1.40% | 0.9918 | 2.20% |
| 平均缓解率 | 0.45% | | 0.36% | | 0.48% | | 0.32% | |

表6.3-3：快的出租车公司的部分缓解效果

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 补贴金额 | 20 | | 19 | | 21 | | 16 | |
| 指标 | **匹配度** | **缓解率** | **匹配度** | **缓解率** | **匹配度** | **缓解率** | **匹配度** | **缓解率** |
| 北京 | 0.9978 | 0.41% | 0.9993 | 0.57% | 0.9968 | 0.31% | 0.9944 | 0.07% |
| 成都 | 0.9987 | -0.03% | 0.9999 | 0.09% | 0.9978 | -0.12% | 0.9987 | -0.02% |
| 上海 | 0.9768 | 1.09% | 0.9773 | 1.14% | 0.9759 | 0.99% | 0.9803 | 1.44% |
| 杭州 | 0.9955 | 0.27% | 0.9975 | 0.47% | 0.9967 | 0.39% | 0.9944 | 0.16% |
| 广州 | 0.9961 | 0.37% | 0.9950 | 0.27% | 0.9959 | 0.36% | 0.9951 | 0.28% |
| 南昌 | 0.9828 | -0.60% | 0.9873 | -0.15% | 0.9806 | -0.82% | 0.9844 | -0.44% |
| 西安 | 0.9979 | 0.01% | 0.9981 | 0.03% | 0.9969 | -0.09% | 0.9993 | 0.15% |
| 哈尔滨 | 0.9913 | 2.04% | 0.9898 | 2.00% | 0.9918 | 2.21% | 0.9876 | 1.78% |
| 平均缓解率 | 0.45% | | 0.55% | | 0.40% | | 0.43% | |

从表中可以发现，由于平均缓解率整体上均大于0，所以两家公司的补贴政策整体上确实对“缓解打车难”有帮助。但因为每一种补贴方案政策实行的时间长度不一致，可以分析平均缓解率随时间变化而变化的情况，同时为了更好地说明问题，我们对平均缓解率进行加权，可以得到各个城市的加权平均缓解率以及在政策实行期间两家公司的加权平均缓解率。

图6.3：两家公司平均缓解率的变化情况

表6.3-4：各城市及出租车公司的加权平均缓解率

|  |  |  |
| --- | --- | --- |
| 出租车公司 | 滴滴 | 快的 |
| 北京 | 0.14% | 0.09% |
| 成都 | -0.03% | -0.01% |
| 上海 | 0.65% | 0.54% |
| 杭州 | 0.27% | 0.10% |
| 广州 | 0.23% | 0.17% |
| 南昌 | 0.09% | 0.14% |
| 西安 | -0.04% | 0.04% |
| 哈尔滨 | 0.76% | 0.47% |
| 加权平均缓解率 | **0.26%** | **0.19%** |

从图中可以发现，两家出租车公司的整体缓解率处于下降趋势，体现了补贴方案后期无力的状况。同时从表中可以发现，两家公司在政策实行期间各城市的加权平均缓解率从整体上来看有利于缓解打车难的问题，但是对于有些城市这两家公司的补贴方案政策却没有起到缓解的作用，出现了各个城市的缓解程度不均衡，甚至加深了某些城市的打车难问题。最后，滴滴出租车公司在政策实行期间的加权平均缓解率高于快的出租车公司，体现了滴滴公司整体缓解率较好。

### 七、问题三模型建立与求解

### 7.1 问题分析

在问题二中可以看出会出现政策对某些城市没有缓解作用的问题，并且各个城市缓解作用不同，有的相差很大，所以如果要设计打车软件平台制定补贴政策，要从两个方面考虑，一个是补贴金额的具体制定，一个是执行政策的时间。其中补贴金额的具体制定，应该要因地制宜，根据不同城市的情况制定不同的补贴政策，使得补贴金额尽可能少，整体平均缓解率最高。因此补贴金额的具体制定问题实质上是不同城市不同补贴金额的组合优化问题。

### 7.2 组合优化的规划模型

本研究设计一个关于指标Q的组合优化模型的规划式，约束条件是所有城市的平均缓解率都为正数，也就是因地制宜的补贴政策会使每个城市的打车难问题都得到缓解。规划模型的数学规划式如下：

 (8)

 (9)

其中，为补贴金额向量，为相应分配给城市的每单补贴金额，为相应城市计算后的缓解率，为平均缓解率。

### 7.3 遗传算法模型

为了求解组合优化的规划模型，我们采用了遗传算法。遗传算法简称GA，是以自然选择和生物遗传理论为基础，将生物进化过程中的“物竞天择，适者生存”的规律与群体内部的染色体的随机信息交换机制相结合，是一种高效的全局寻优搜索算法[7]。

### 7.3.1 模型建立

遗传算法包括三个基本操作：选择、交叉和变异。这些基本操作又有许多不

同的方法，使得遗传算法在实用时具有不同的特色[8]。

基本过程如下：

**Step1:**计算开始时，一定数目个体即种群随机地初始化；

**Step2:**计算每个个体的适应度函数，产生第一代；

**Step3:**如果不满足优化准则，按适应度选择个体，父代要求基因重组（交叉）

而产生子代；

**Step4:**所有的子代按一定概率变异

**Step5:**重新计算子代的适应度；

**Step6:**子代被插入到种群中将父代取而代之，构成新一代，这一过程一直到

满足优化准则为止。

遗传算法的流程图如下图所示：

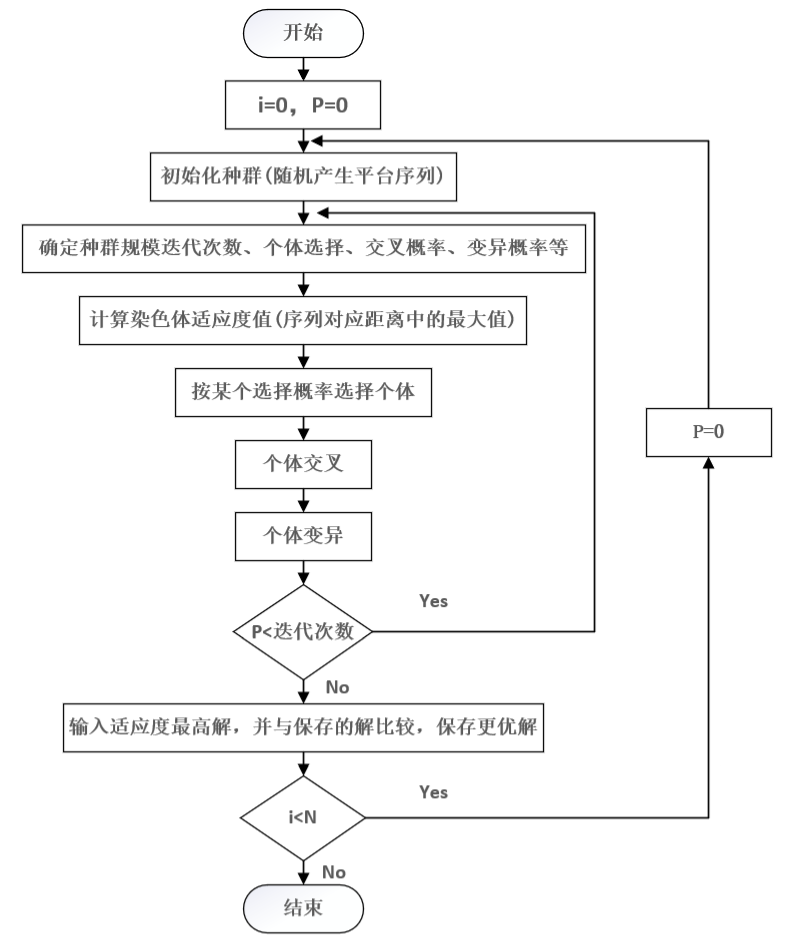


图6.3.1：遗传算法流程图

### 7.3.2 针对问题和模型的改进遗传算法的细节

1）产生初始种群

在产生方法上有两种：一是完全随机产生，它适合于对问题的解无任何先验知识的情况。随机性较强，因而也较公正。二是某些先验知识可转变为必须满足的一组要求，然后在满足这些要求的解中在随机地选取样本[9]。

虽然第二种方法这样选择初始种群可使遗传算法更快的达到最优解。种群有一定的目标性和代表性，但取例不如完全随机的广泛，而且先验知识是否可靠也是一个问题。尤其是在前面结果可以看出，不一定补贴的金额越大就对某个城市越有利，所以综合来说，我们选择完全随机产生的方法。

并且在分析问题后，我们的种群的内容并不是常规的二进制值，而是一段选取自0~20的长度为8目标序列，作为我们的补贴金额向量，这样的数据结构可以方便我们计算适应度和交叉。

2）适应度函数

根据模型和种群内容，我们可以直接根据种群的个体序列来对主成分分析法中的价格因子进行修改，再重新计算缓解度后直接求出平均缓解度除以个体序列综合（即补贴金额总和）的结果，作为适应度大小。并且为了方便后面的筛选，在返回值上应该同时设置最大值和最小值，并且暂时修改成负值，这样的话可以直接挑选最小值进行选择。对于某个个体，适应度越小说明该个体越优秀，适应度越大说明个体越差。

 (10)

其中，,为种群个体数，为补贴金额向量，为相应分配给城市的每单补贴金额，为相应城市计算后的缓解率，为平均缓解率。

3）选择

针对本题的特点来设置，选择将使适应度较小(优良)个体有较大的存在机会，而适应度较大（低劣）的个体继续存在的机会也较小。我们采用赌轮选择机制，令表示群体的适应度值之总和，表示种群中第个染色体的适应度值，它产生后代的能力正好为其适应度值所占份额。

4）交叉

基于类路径表示的编码方法，要求一个个体的染色体编码中不允许有重复的基因码，也就是说要满足任意一个路口必须而且只能匹配一个平台的约束。基本遗传算法的交叉操作生成的个体一般不能满足这一约束条件。这里我们选择部分匹配交叉()方法。

部分匹配交叉操作要求随机选取两个交叉点，以便确定一个匹配段，根据两个父个体中两个交叉点之间的中间段给出的映射关系生成两个子个体。

5）变异

上面分析过，基于二进值编码的变异操作不能适用，不能够由简单的变量的翻转来实现。在序列选择问题中个体的编码是目标的序列，随机的在这个序列抽取两个路口，然后交换他们的位置。这样就实现了个体编码的变异，算法如下：

1、产生两个0到1之间的随机实数；

2、将这两个随机实数转化为0到n（目标路口数）-1之间的随机整数；

3、将这两个随机整数指代的路口进行交换。

### 7.3.3 模型求解

### 7.3.3.1 补贴总金额的具体制定

按照遗传算法的算法流程，注意到问题和模型的改进遗传算法的细节，在种群数量200、迭代次数150、总共随机种群次数5000次的情况下得出最优解。在满足所有城市的打车难问题都得到了缓解的前提下，花费补贴金额最少且收益最大的最佳方案如下：

表7.3.3：补贴金额最佳方案

|  |  |
| --- | --- |
| 城市 | 每单司机和乘客补贴总金额（元） |
| 北京 | 5 |
| 成都 | 9 |
| 上海 | 2 |
| 杭州 | 8 |
| 广州 | 3 |
| 南昌 | 1 |
| 西安 | 4 |
| 哈尔滨 | 2 |

### 7.4 政策执行的具体时间

### 7.4.1 logistic模型建立

资源和环境对自然界事物增长的阻滞作用体现在对增长率的影响上，使随着自然界事物数量的增加而下降，将表示为的函数，并且取既简单又便于应用的线性减函数。其数学模型为：

 (11)

 (12)

其中，为内禀增长率，为自然界事物容量。

### 7.4.2 模型求解

通过查阅资料发现，除了滴滴和快的两个平台其他平台的用户覆盖率只占6.3%。由于不同出租车公司之间存在市场资源的竞争，我们因此选取较低的期望用户数量为100万。同时通过滴滴打车在2014年1月到3月的用户数增长数据来分析，通过拟合和最小二乘估计法可以得到新晋平台在举办补贴政策活动的用户数目期望内禀增长率为0.283。

至此本研究已经得到了logistic模型的参数容量和内禀增长率，假设初始用户有5000人，带入logistic模型中可知：

1. 在0~7天时导数小于1，此时用户逐渐开始被吸引，这里的用户包括司机和顾客。司机的数量虽然远远小于乘客，但是会带动更多的乘客消费，所以可以分配80%补贴金额给司机，20%补贴金额给乘客。此时的方案如下：

表7.4.2-1：前期补贴方案

|  |  |  |
| --- | --- | --- |
| 城市 | 补贴司机金额 | 补贴乘客金额 |
| 北京 | 4 | 1 |
| 成都 | 7 | 2 |
| 上海 | 2 | 0 |
| 杭州 | 4 | 2 |
| 广州 | 2 | 1 |
| 南昌 | 1 | 0 |
| 西安 | 3 | 1 |
| 哈尔滨 | 2 | 0 |

1. 在7~20天的时间段内，增长率逐渐达到最大，导数逐渐变大，此时司机所占的因素变少，乘客所占的因素变大，此时可以分配50%补贴金额给司机，50%补贴金额给乘客，方案如下：

表7.4.2-2：中期补贴方案

|  |  |  |
| --- | --- | --- |
| 城市 | 补贴司机金额 | 补贴乘客金额 |
| 北京 | 2 | 3 |
| 成都 | 5 | 4 |
| 上海 | 1 | 1 |
| 杭州 | 4 | 4 |
| 广州 | 2 | 1 |
| 南昌 | 0 | 1 |
| 西安 | 2 | 2 |
| 哈尔滨 | 1 | 1 |

1. 在21~31天的时间段内，增长率逐渐下降，导数逐渐趋于0，此时因为公司可以通过竞争获取更多的容量，所以仍然分配20%补贴金额给司机，80%补贴金额给乘客，去保持竞争力和活力，方案如下：

表7.4.2-3：后期补贴方案

|  |  |  |
| --- | --- | --- |
| 城市 | 补贴司机金额 | 补贴乘客金额 |
| 北京 | 1 | 4 |
| 成都 | 2 | 7 |
| 上海 | 0 | 2 |
| 杭州 | 2 | 4 |
| 广州 | 1 | 2 |
| 南昌 | 0 | 1 |
| 西安 | 1 | 3 |
| 哈尔滨 | 0 | 2 |

1. 31天后增长率不明显，这时候我们就应该根据实际的经营情况去更新全新的补贴金额方案和具体的时间规划。

### 八、模型评价与推广

对于问题一，首先对查询到的指标数据进行了PCA主成分分析进行降维，根据主成分的分析分别选取了3个主成分作为后续的影响因素。PCA 在对原有指标进行降维后形成了互不关联的主成分，因此可以去除评价指标之间的相互影响。同时可以在评价指标较多时可以保留原有信息的绝大部分信息，从而可以减少评价指标的数量。因此使用主成分分析对降维具有良好的可推广性。在计算“供求匹配度”时，我们根据供需关系建立了供求匹配模型，该模型能够很好的利用现有数据进行计算，因此在相关供求问题中有很好的适用性。

对于问题二，在问题一模型的基础上增加新的因子作为重要参量，建立起缓解率指标模型。该模型通过增加供求缓解率来评判不同公司的补贴政策是否对打车难问题有所缓解，可以直观的通过单一变量的变化来反映结果，说明模型灵敏度较好，适用于新增变量类问题。未来可以考虑加入更多因子进行分析

对于问题三，针对因地制宜问题设计了可行的组合优化模型，算法上采用经过细节修改过的遗传算法对模型进行求解，可以得出最优解。我们使用logistic模型和拟合出来的参数，结合补贴金额方案，可以给出不同时间段下具体的补贴方案。说明本模型设计出的补贴方案非常全面，在论证分析中也得出较好的结论。因此本模型在补贴方案问题中有很大的应用空间。未来可以考虑加入其他智能算法进行求解。

### 九、参考文献

* [1]Hao Li,Shengyuan Zhong,Yongzhen Wang,Jun Zhao,Minxia Li,Fu Wang,Jiebei Zhu. New understanding on information’s role in the matching of supply and demand of distributed energy system[J]. Energy,2020,206.
* [2]朱桂玲.基于供求匹配模型对机场出租车问题的规划研究[J].广西质量监督导报,2020(01):8.
* [3]王小玉,陈琳.基于主成分分析与迭代最近点的三维膝关节配准[J/OL].北京邮电大学学报:1-6[2020-07-22].https://elkssl0a75e822c6f3334851117f8769a30e1clib.v.just.edu.cn:4443/10.13190/j.jbupt.2019-165.
* [4]王军.基于主成分回归分析的颜色与物质浓度辨识研究[J].商丘师范学院学报,2020,36(09):16-18.

### [5]王学红.浅析农机购置补贴对促进农机推广的作用[J].南方农机,2020,51(13):60.

### [6]余黎峰,张玉萌.基于成本视角的WEEE处理行业补贴政策研究——以A公司为例[J].会计之友,2020(14):39-45.

### [7]邓翔,彭杰,吕一清.基于量子遗传算法优化的新Prophet模型及其验证[J/OL].系统工程:1-19[2020-07-22].https://cc0eb1c56d2d940cf2d0186445b0c858lib.v.just.edu.cn:4443/kcms/detail/43.1115.N.20200720.1555.002.html.

### [8]王晓坤,王端,齐少璞,胡赟,张东辉.基于遗传算法的钠冷快堆堆芯流量分区优化设计方法[J/OL].原子能科学技术

### [9]徐红涛,陈春波,郑宏伟,罗格平,杨辽,王伟胜,吴世新.基于相关分析和自适应遗传算法的盐渍化建模变量和参数优选[J].地球信息科学学报,2020,22(07):1497-1509.

### 十、附录

**PCA.m**

function[ F,FF ] = PCA(xx,c)

%load data1.mat % 主成分聚类

% load data2.mat % 主成分回归

x = xx;

for i = 1:8

x(i,9) = x(i,9)-c;

x(i,10) = ((x(i,7)\*10000/x(i,1))/30)/x(i,9);

end

% 注意，这里可以对数据先进行描述性统计

% 描述性统计的内容见第5讲.相关系数

[n,p] = size(x); % n是样本个数，p是指标个数

%% 第一步：对数据x标准化为X

X=zscore(x); % matlab内置的标准化函数（x-mean(x)）/std(x)

%% 第二步：计算样本协方差矩阵

R = cov(X);

%% 注意：以上两步可合并为下面一步：直接计算样本相关系数矩阵

R = corrcoef(x);

%disp('样本相关系数矩阵为：')

%disp(R)

%% 第三步：计算R的特征值和特征向量

% 注意：R是半正定矩阵，所以其特征值不为负数

% R同时是对称矩阵，Matlab计算对称矩阵时，会将特征值按照从小到大排列哦

% eig函数的详解见第一讲层次分析法的视频

[V,D] = eig(R); % V 特征向量矩阵 D 特征值构成的对角矩阵

%% 第四步：计算主成分贡献率和累计贡献率

lambda = diag(D); % diag函数用于得到一个矩阵的主对角线元素值(返回的是列向量)

lambda = lambda(end:-1:1); % 因为lambda向量是从小大到排序的，我们将其调个头

contribution\_rate = lambda / sum(lambda); % 计算贡献率

cum\_contribution\_rate = cumsum(lambda)/ sum(lambda); % 计算累计贡献率 cumsum是求累加值的函数

%disp('特征值为：')

%disp(lambda') % 转置为行向量，方便展示

%disp('贡献率为：')

%disp(contribution\_rate')

%disp('累计贡献率为：')

%disp(cum\_contribution\_rate')

%disp('与特征值对应的特征向量矩阵为：')

% 注意：这里的特征向量要和特征值一一对应，之前特征值相当于颠倒过来了，因此特征向量的各列需要颠倒过来

% rot90函数可以使一个矩阵逆时针旋转90度，然后再转置，就可以实现将矩阵的列颠倒的效果

V=rot90(V)';

%% 计算我们所需要的主成分的值

m =2;

F = zeros(n,m); %初始化保存主成分的矩阵（每一列是一个主成分）

for i = 1:m

ai = V(:,i)'; % 将第i个特征向量取出，并转置为行向量

Ai = repmat(ai,n,1); % 将这个行向量重复n次，构成一个n\*p的矩阵

F(:, i) = sum(Ai .\* X, 2); % 注意，对标准化的数据求了权重后要计算每一行的和

end

%% 计算供求匹配系数

FF = zeros(n,1);

for j = 1:n

for i = 1:m

FF(j,1) = FF(j,1)+F(j,i)\*(contribution\_rate(i,1)/cum\_contribution\_rate(m,1));

end

end

end

**PCA\_1.m**

function[ F,FF ] = PCA\_1(xx,c)

%load data1.mat % 主成分聚类

% load data2.mat % 主成分回归

x = xx;

for i = 1:8

x(i,9) = x(i,9)-c(1,i);

x(i,10) = ((x(i,7)\*10000/x(i,1))/30)/x(i,9);

end

% 注意，这里可以对数据先进行描述性统计

% 描述性统计的内容见第5讲.相关系数

[n,p] = size(x); % n是样本个数，p是指标个数

%% 第一步：对数据x标准化为X

X=zscore(x); % matlab内置的标准化函数（x-mean(x)）/std(x)

%% 第二步：计算样本协方差矩阵

R = cov(X);

%% 注意：以上两步可合并为下面一步：直接计算样本相关系数矩阵

R = corrcoef(x);

%disp('样本相关系数矩阵为：')

%disp(R)

%% 第三步：计算R的特征值和特征向量

% 注意：R是半正定矩阵，所以其特征值不为负数

% R同时是对称矩阵，Matlab计算对称矩阵时，会将特征值按照从小到大排列哦

% eig函数的详解见第一讲层次分析法的视频

[V,D] = eig(R); % V 特征向量矩阵 D 特征值构成的对角矩阵

%% 第四步：计算主成分贡献率和累计贡献率

lambda = diag(D); % diag函数用于得到一个矩阵的主对角线元素值(返回的是列向量)

lambda = lambda(end:-1:1); % 因为lambda向量是从小大到排序的，我们将其调个头

contribution\_rate = lambda / sum(lambda); % 计算贡献率

cum\_contribution\_rate = cumsum(lambda)/ sum(lambda); % 计算累计贡献率 cumsum是求累加值的函数

%disp('特征值为：')

%disp(lambda') % 转置为行向量，方便展示

%disp('贡献率为：')

%disp(contribution\_rate')

%disp('累计贡献率为：')

%disp(cum\_contribution\_rate')

%disp('与特征值对应的特征向量矩阵为：')

% 注意：这里的特征向量要和特征值一一对应，之前特征值相当于颠倒过来了，因此特征向量的各列需要颠倒过来

% rot90函数可以使一个矩阵逆时针旋转90度，然后再转置，就可以实现将矩阵的列颠倒的效果

V=rot90(V)';

%% 计算我们所需要的主成分的值

m =2;

F = zeros(n,m); %初始化保存主成分的矩阵（每一列是一个主成分）

for i = 1:m

ai = V(:,i)'; % 将第i个特征向量取出，并转置为行向量

Ai = repmat(ai,n,1); % 将这个行向量重复n次，构成一个n\*p的矩阵

F(:, i) = sum(Ai .\* X, 2); % 注意，对标准化的数据求了权重后要计算每一行的和

end

%% 计算供求匹配系数

FF = zeros(n,1);

for j = 1:n

for i = 1:m

FF(j,1) = FF(j,1)+F(j,i)\*(contribution\_rate(i,1)/cum\_contribution\_rate(m,1));

end

end

end

**pipei.m**

%计算各个城市的供求匹配程度

function[score] = pipei(xx,xy,c)

[F1,FF1] = PCA(xx,c);

[F2,FF2] = PCA(xy,c);

for i = 1:8

fangcha\_1 = sum((FF1(:,1)-mean(FF1)).^2)/(length(FF1));

fangcha\_2 = sum((FF2(:,1)-mean(FF2)).^2)/(length(FF2));

end

score1 = zeros(8,1);

for i = 1:8

u1 = exp(-((0.2\*FF1(i,1)).^2)/fangcha\_1);

u2 = exp(-((0.2\*FF2(i,1)).^2)/fangcha\_2);

temp = 1;

if u1>u2

temp = u2/u1;

score1(i,1) = temp;

else

temp = u1/u2;

score1(i,1) = temp;

end

end

score = score1;

end

**pipei\_1.m**

%计算各个城市的供求匹配程度

function[score] = pipei\_1(xx,xy,c)

ori = pipei(xx,xy,0);

[F1,FF1] = PCA\_1(xx,c);

[F2,FF2] = PCA\_1(xy,c);

for i = 1:8

fangcha\_1 = sum((FF1(:,1)-mean(FF1)).^2)/(length(FF1));

fangcha\_2 = sum((FF2(:,1)-mean(FF2)).^2)/(length(FF2));

end

score1 = zeros(8,1);

for i = 1:8

u1 = exp(-((0.2\*FF1(i,1)).^2)/fangcha\_1);

u2 = exp(-((0.2\*FF2(i,1)).^2)/fangcha\_2);

temp = 1;

if u1>u2

temp = u2/u1;

score1(i,1) = temp;

else

temp = u1/u2;

score1(i,1) = temp;

end

end

%这里直接返回平均缓解度

huanjie = zeros(8,1);

for j = 1:8

huanjie(j,1) = (score1(j,1) - ori(j,1))/ori(j,1);

end

score = sum(huanjie(:,1))/8;

end

**logistic.m**

%求期望时间

function[y,y1] = logistic(xm,r,x0,t)

y = xm/(1+((xm/x0)-1)\*exp(-r\*t));

y1 = r\*y\*(1-y/xm);

end

**test1.m**

%计算两家公司的缓解率

%xx,xy为设定好的因子数据

%首先计算没有减费时各个城市的供求匹配程度

ori = pipei(xx,xy,0);

%手动敲入快的和滴滴的补贴政策 KD\_butie DD\_butie

%接下来建立两种结果的矩阵 每一列是相对应的城市供求匹配程度

KD\_result = zeros(8,9);

DD\_result = zeros(8,8);

for i = 1:9

temp = pipei(xx,xy,KD\_butie(1,i));

KD\_result(:,i) = temp(:,1);

end

for i = 1:8

temp = pipei(xx,xy,DD\_butie(1,i));

DD\_result(:,i) = temp(:,1);

end

%接下来计算缓解率

KD\_huanjie = zeros(8,9);

DD\_huanjie = zeros(8,8);

for i = 1:9

for j = 1:8

KD\_huanjie(j,i) = (KD\_result(j,i) - ori(j,1))/ori(j,1);

end

end

for i = 1:8

for j = 1:8

DD\_huanjie(j,i) = (DD\_result(j,i) - ori(j,1))/ori(j,1);

end

end

%接下来计算对于每个补贴金额 对于所有城市的平均缓解率

KD\_huanjie\_mean = zeros(1,9);

DD\_huanjie\_mean = zeros(1,8);

for i = 1:9

KD\_huanjie\_mean(1,i) = sum(KD\_huanjie(:,i))/8;

end

for i = 1:8

DD\_huanjie\_mean(1,i) = sum(DD\_huanjie(:,i))/8;

end

%接下来计算两个平台在政策实行时间内的加权平均缓解率

%DD\_time和KD\_time是手动输入的相对应的政策实行时间

temp = 0;

for i = 1:9

temp = temp+KD\_time(1,i)\*KD\_huanjie\_mean(1,i);

end

KD\_huanjie\_jiaquan = temp/sum(KD\_time(1,:));

temp = 0;

for i = 1:8

temp = temp+DD\_time(1,i)\*DD\_huanjie\_mean(1,i);

end

DD\_huanjie\_jiaquan = temp/sum(DD\_time(1,:));

%政策实行期间各个城市的加权平均缓解率

KD\_huanjie\_city = zeros(8,1);

for j = 1:8

temp = 0;

for i = 1:9

temp = temp+KD\_time(1,i)\*KD\_huanjie(j,i);

end

KD\_huanjie\_city(j,1) = temp/sum(KD\_time(1,:));

end

DD\_huanjie\_city = zeros(8,1);

for j = 1:8

temp = 0;

for i = 1:8

temp = temp+DD\_time(1,i)\*DD\_huanjie(j,i);

end

DD\_huanjie\_city(j,1) = temp/sum(DD\_time(1,:));

End

mmin = 99999;

best = zeros(1,8);

tStart = tic; % 算法计时器

%%%%%%%%%%%%自定义参数%%%%%%%%%%%%%

%[cityNum,cities] = Read('dsj1000.tsp');

%根据任务单的坐标定义cities

for num = 1:200

cityNum = 8;

maxGEN = 100;

popSize = 100; % 遗传算法种群大小

crossoverProbabilty = 0.9; %交叉概率

mutationProbabilty = 0.1; %变异概率

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% 随机生成城市位置

%cities = rand(2,cityNum) \* 100;%100是最远距离

% 计算上述生成的城市距离

% 生成种群，每个个体代表一个路径

pop = zeros(popSize, cityNum);

for i=1:popSize

pop(i,:) = randperm(20,8);

end

offspring = zeros(popSize,cityNum);

%保存每代的最小路劲便于画图

minPathes = zeros(maxGEN,1);

% GA算法

for gen=1:maxGEN

% 计算适应度的值，即路径总距离

[fval, sumDistance, minPath, maxPath] = fitness(xx, pop,xy);

% 轮盘赌选择

tournamentSize=4; %设置大小

for k=1:popSize

% 选择父代进行交叉

tourPopDistances=zeros( tournamentSize,1);

for i=1:tournamentSize

randomRow = randi(popSize);

tourPopDistances(i,1) = sumDistance(randomRow,1);

end

% 选择最好的，即距离最小的

parent1 = min(tourPopDistances);

[parent1X,parent1Y] = find(sumDistance==parent1,1, 'first');

parent1Path = pop(parent1X(1,1),:);

for i=1:tournamentSize

randomRow = randi(popSize);

tourPopDistances(i,1) = sumDistance(randomRow,1);

end

parent2 = min(tourPopDistances);

[parent2X,parent2Y] = find(sumDistance==parent2,1, 'first');

parent2Path = pop(parent2X(1,1),:);

subPath = crossover(parent1Path, parent2Path, crossoverProbabilty);%交叉

subPath = mutate(subPath, mutationProbabilty);%变异

offspring(k,:) = subPath(1,:);

minPathes(gen,1) = minPath;

end

%fprintf('代数:%d 最短路径:%.2fM \n', gen,minPath);

% 更新

pop = offspring;

% 画出当前状态下的最短路径

end

if minPath < mmin

best = subPath;

mmin = minPath;

end

end

tEnd = toc(tStart);

fprintf('时间:%d 分 %f 秒.\n', floor(tEnd/60), rem(tEnd,60));

function [childPath] = crossover(parent1Path, parent2Path, prob)

% 交叉

random = rand();

if prob >= random

[l, length] = size(parent1Path);

childPath = zeros(l,length);

setSize = floor(length/2) -1;

offset = randi(setSize);

for i=offset:setSize+offset-1

childPath(1,i) = parent1Path(1,i);

end

iterator = i+1;

j = iterator;

while any(childPath == 0)

if j > length

j = 1;

end

if iterator > length

iterator = 1;

end

if ~any(childPath == parent2Path(1,j))

childPath(1,iterator) = parent2Path(1,j);

iterator = iterator + 1;

end

j = j + 1;

end

else

childPath = parent1Path;

end

end

**test2.m**

function [ fitnessvar, sumDistances,minPath, maxPath ] = fitness( xx, pop,xy)

% 计算整个种群的适应度值

[popSize, col] = size(pop);

temp = zeros(1,col);

sumDistances = zeros(popSize,1);

fitnessvar = zeros(popSize,1);

%计算

for i=1:popSize

temp = pop(i,:);

sumDistances(i,1) = -pipei\_1(xx,xy,temp)/sum(temp(1,:));

end

minPath = min(sumDistances);

maxPath = max(sumDistances);

for i=1:length(sumDistances)

fitnessvar(i,1)=(maxPath - sumDistances(i,1)+0.000001) / (maxPath-minPath+0.00000001);

end

end

function [ mutatedPath ] = mutate( path, prob )

%对指定的路径利用指定的概率进行更新

random = rand();

if random <= prob

[l,length] = size(path);

index1 = randi(length);

index2 = randi(length);

%交换

temp = path(l,index1);

path(l,index1) = path(l,index2);

path(l,index2)=temp;

end

mutatedPath = path;

end

function [ output\_args ] = paint( cities, pop, minPath, totalDistances,gen)

gNumber=gen;

[~, length] = size(cities);

xDots = cities(1,:);

yDots = cities(2,:);

%figure(1);

title('GA TSP');

plot(xDots,yDots, 'p', 'MarkerSize', 14, 'MarkerFaceColor', 'blue');

xlabel('横坐标');

ylabel('纵坐标');

axis equal

hold on

[minPathX,~] = find(totalDistances==minPath,1, 'first');

bestPopPath = pop(minPathX, :);

bestX = zeros(1,length);

bestY = zeros(1,length);

for j=1:length

bestX(1,j) = cities(1,bestPopPath(1,j));

bestY(1,j) = cities(2,bestPopPath(1,j));

end

title('最佳路线示意图（仅表示路线顺序）');

plot(bestX(1,:),bestY(1,:), 'red', 'LineWidth', 1.25);

legend('目标货格', '路径');

axis equal

grid on

%text(5,0,sprintf('迭代次数: %d 总路径长度: %.2f',gNumber, minPath),'FontSize',10);

drawnow

hold off

end

function [n\_citys,city\_position] = Read(filename)

fid = fopen(filename,'rt');

location=[];

A = [1 2];

tline = fgetl(fid);

while ischar(tline)

if(strcmp(tline,'NODE\_COORD\_SECTION'))

while ~isempty(A)

A=fscanf(fid,'%f',[3,1]);

if isempty(A)

break;

end

location=[location;A(2:3)'];

end

end

tline = fgetl(fid);

if strcmp(tline,'EOF')

break;

end

end

[m,n]=size(location);

n\_citys = m;

city\_position=location;

fclose(fid);

end