基于聚类和线性规划的共享单车调度方案

摘要

共享单车逐渐成为人们出行的重要方式，未来共享单车的需求量将会进一步增加，随之而来的共享单车投放区域和数量等决策成为重要问题，本文通过研究北京市共享单车使用数据，找出早高峰时间段（7：00-9：00）和晚高峰时间段（17：00-19：00）共享单车使用的热门区域，设计出热门区域之间共享单车车辆调度成本最低的方案，旨在帮助用户们更方便地使用共享单车，实现自行车回归城市的目标。

针对问题一，本文采用了两种解法来找出早高峰时间段和晚高峰时间段共享单车使用的热门区块。解法一是采用二层聚类法，以 K-means 算法进行第一层聚类，从单位点（Geohash 解码后的最小区块）到聚类中心，并选出热度（包含订单数量）前 10 的热门聚类中心，循坏 10 次前述过程，共获得 100 个热门聚类中心。观察发现 K-means 的聚类结果并不稳定，但 10 次所得的聚类中心存在高密度区域，因此对 K-means 的热门聚类中心进行 DBSCAN 二次聚类，以确定“收敛点”（高密度区域的坐标平均点），获得 10 个收敛点后多次运行以检验稳定性，发现结果稳定性较高，最终得出的早高峰热门区块中心位置的经纬度坐标为（116.202°E，39.9147°N）（116.299°E，39.9345°N）（116.312°E，39.9683°N）等，晚高峰热门区块的经纬度坐标为（116.19°E，39.9247°N）（116.263°E，39.912°N）（116.299°E，39.85°N）等，完整数据结果请见正文“5.1.2 解法一”部分。解法二是采用区域网格法，通过取单车使用量排名靠前的若干个区域，采用 K-means 算法经过反复尝试将这若干个区域合并成十个大型区域，最终得出早高峰和晚高峰单车使用量的热门区块，早高峰热门区块的中心位置经纬度坐标为（116.4585114°E，39.86595154°N）（116.4955902°E，39.90852356°N）（116.17836°E，39.92637634°N）等，晚高峰热门区块的中心位置经纬度坐标为（116.4585114°E，39.86595154）（116.5189362°E，39.92225647）（116.3733673°E，39.948349）等，完整结果及热门区块所包含的 Geohash 编码表示的原始区域请见正文“5.1.3 解法二”部分。

针对问题二，本文利用 train 数据集和统计工具求出早高峰时段所有区块中出发地对应频率最高的到达地，得出出发地区块与到达地区块的对应关系表，然后仿照问题一求出 test 数据集中早高峰热门出发地区块，通过查表（选择最接近的出发地坐标）预测出 test 数据集中热门区块早高峰时间段内骑行的目的地所在区块。

针对问题三，本文采用线性规划模型，将早高峰和晚高峰热门区块合并考虑，将较为重叠的区块记为一个，得到十七个热力区块，分别计算这十七个热力区块每周七天各自的早高峰和晚高峰时的自行车变化量，即作为终点的次数减去作为起点的次数，差值为正则数量净增长，差值为负则负增长。将正增长的区块视为产地，将负增长的区块视为销地，如此，将单车调度问题化为运输问题，目标为运输总路径最短，以所有负增长区块的负增长量被补为 0 为约束条件，基于运输问题建立线性规划模型后，在 MATLAB 中进行线性规划求解，从而制定每天的共享单车调度计划。

关键词：二层聚类 K-means DBSCAN 热门区块　线性规划　车辆调度

一、问题重述

1.1 问题背景

共享单车自出现以来就迅速推广，深受人们推崇，越来越多的人选择使用它来代替传统的交通方式，更是成为很多城市除公共交通以外的居民首选出行方式，这一低碳的绿色出行方式，不仅有效减轻了城市路网压力和交通拥堵程度，大幅提高城市运作效率，更是一种健康的生活锻炼方式。随着绿色环保、低碳出行理念的进一步推广，未来将会有更多用户选择共享单车这种出行方式，随之而来的共享单车投放决策也成为一大问题。

1.2 问题提出

当前，某品牌共享单车在某城市投放单车数量超过 40 万。用户可以就近在出发地附近的共享单车集中摆放地找到空闲单车，用手机解锁，然后骑到目的地后再把单车停好并在手机上还车。现收集到该城市一周的共享单车使用数据，其中，部分数据经过脱敏处理，单车所在地理位置通过 Geohash 加密，可通过开源方法获得对应经纬度数据。

根据上述条件，我们需要建立数学模型求解以下问题：

（1）根据所给 train 数据集分别找到前十位早高峰时间段（7 点-9 点）和晚高峰时间段（17 点-19 点）共享单车使用的热门区块；

（2）根据第一问所得结果，求出所给 test 数据集在早高峰时间段内出发地的热门区块，并预测其在热门区块的早高峰时间段内骑行的目的地所在区块。

（3）根据前两问结果以及数据判断是否需要人工调解热门区块的单车数量，如果需要人工调节，考虑在满足人群出行需求的条件下以最低成本进行调节的方案。

二、问题分析

2.1 问题一分析

问题一的要求是根据所给 train 数据集分别找到早高峰时间段（7 点-9 点）和晚高峰时间段（17 点-19 点）共享单车使用的前十个热门区块。要解决问题一，需要进行数据预处理，首先从数据集中分别提取出早高峰时间段（7 点-9 点）的共享单车数据和晚高峰时间段（17 点-19 点）的共享单车数据，再分别将早高峰时间段和晚高峰时间段中共享单车的出发地和到达地的 Geohash 字符串编码信息转化成经纬度信息，确定经纬度后，可以采用聚类算法分别将出发地和到达地聚类为出发区域和到达区域，并计算各区域订单数从而识别出前十位热门区块。

2.2 问题二分析

问题二的要求是预测 test 数据集中热门区块早高峰时间段内骑行的目的地所在区块。要解决问题二，首先要利用 train 数据集和统计工具求出早高峰时段所有区块中出发地区块对应频率最高的到达地区块，其次要仿照问题一求出 test 数据集中早高峰热门出发地区块，再根据 train 数据集得出的出发地所在区块与到达地所在区块的对应关系表，通过查表（选择最接近的出发地坐标）预测出 test 数据集中热门区块早高峰时间段内骑行的目的地所在区块。

2.3 问题三分析

问题三的要求是根据前两问结果判断是否需要人工调节热门区块单车数量，若需

要调节，考虑在满足人群出现需求的条件下以最低成本调节。要解决问题三，考虑早高峰和晚高峰的热力区块共二十个，发现早高峰的热力区块与晚高峰的热力区块有三块较为重叠，将较为重叠的区块记为一个区块，于是得到十七个热力区块，分别计算这十七个热力区块每周七天各自的早高峰和晚高峰时的自行车变化量，即作为终点的次数减去作为起点的次数，差值为正则数量净增长，差值为负则负增长。将正增长的区块视为产地，将负增长的区块视为销地，如此，将单车调度问题化为运输问题，目标为运输总路径最短，以所有负增长区块的负增长量被补为 0 为约束条件，基于运输问题建立线性规划模型后，在 matlab 中进行线性规划求解。

三、模型假设

1.寻找共享单车的人会在 500m 之内进行寻找
2.共享单车调度成本只与调度距离成正比
3.共享单车不会损坏；
4.不考虑北京楼盘地势影响
5.忽略天气因素
6.共享单车到达目的地后，会及时归还，不会占用过长时间。
7.共享单车的调度在每天晚上运营结束后进行，且仅在热力区块之间进行，不涉及其他区块

四、符号说明

符号	符号含义	单位
K	聚类数目	个
ϵ	DBSCAN 聚类的邻域半径	度（经纬度）
Min_samples	DBSCAN 聚类的ϵ邻域内样本数阈值	个
m	单车数量净增长区块（产地）的数量	辆
n	单车数量净负增长区块（销地）的数量	辆
c_{ij}	第 i 个产地到第 j 个销地的距离	千米
x_{ij}	第 i 个产地到第 j 个销地的运输量	辆
a_i	第 i 个产地的产量（净增长量）	辆
b_j	第 j 个销地的销量（净负增长量）	辆

五、模型的建立与求解

5.1 问题一模型的建立与求解

5.1.1 数据预处理

由于可能存在信号不良、单车故障和用户误操作等问题导致共享单车与服务器出现通信异常的情况，从而产生错误的订单数据，因此需要对原始的共享单车订单数据进行预处理，以消除误差影响。数据预处理主要包括以下两个方面：

（1）由于早高峰时间段为 7：00-9：00，晚高峰时间段在 17：00-19：00，因此在 MATLAB 中将共享单车骑行起始日期时间不在这两个时间段内的数据剔除，并将早高峰和晚高峰时间段的数据分开成两个数据集，然后对 Geohash 字符串编码进行经纬度转换：将每个 32 位进制编码转化为二进制编码，按类二分法的转化规则将二进制编码

转化成经纬度信息，从而分别得到早高峰出发地、早高峰到达地、晚高峰出发地、晚高峰到达地的所有地点经纬度信息，再用 tabulate 函数统计一个字符串在结构体中出现的次数，即统计出相同出发地或到达地的订单次数，然后将数据保存在“早高峰.xlsx”和“晚高峰.xlsx”中。

（2）通过查看经纬度信息数据，发现有些数据偏离较大，需要删去这些离群点。采用 MATLAB 中的 rmoutliers 函数，将分布于“3σ”以外的个案进行删除。

得到早高峰和晚高峰共享单车出发地和到达地位置坐标信息后，将其进行可视化，得到以下地点分布图：

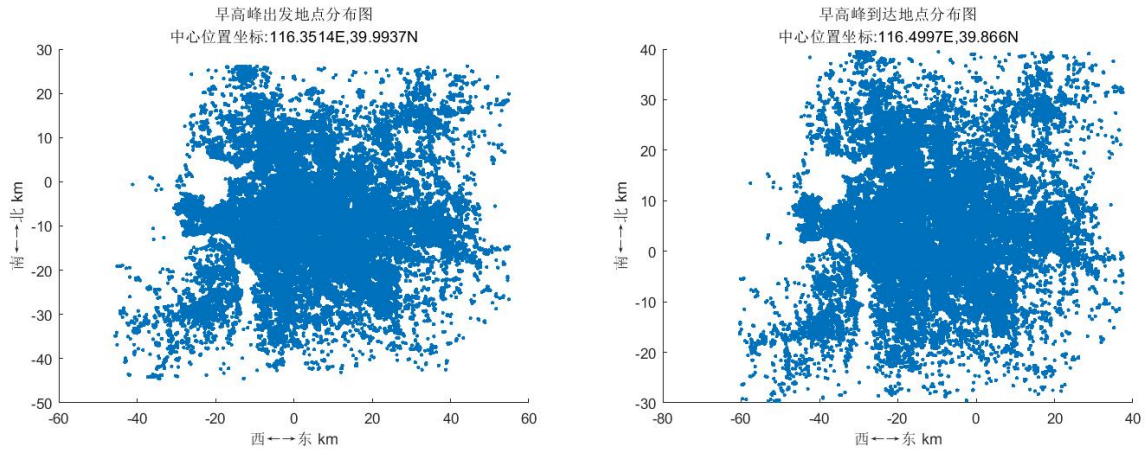

图 1 早高峰出发地点分布和到达地点分布

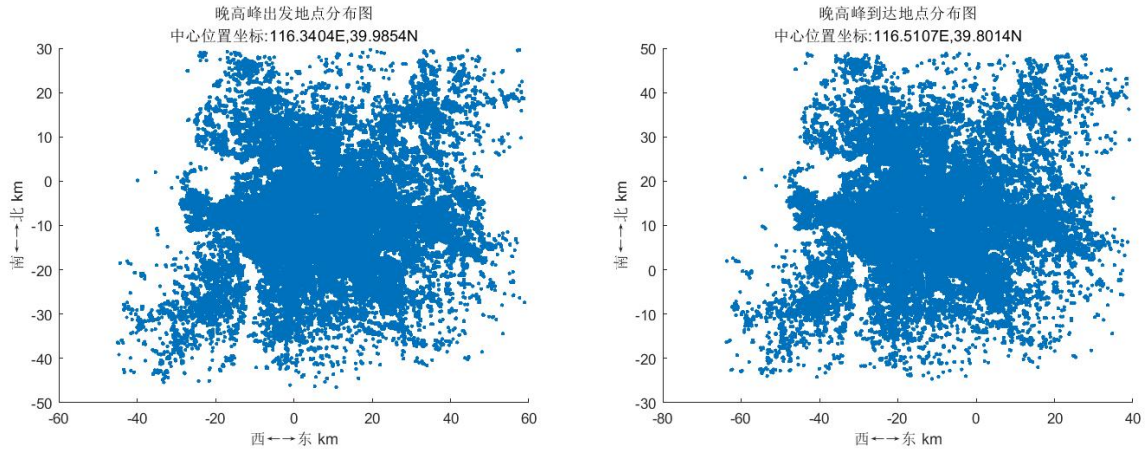

图 2 晚高峰出发地点分布和到达地点分布

5.1.2 解法一：二层聚类法

（1）K-means 聚类

热门区块的识别需要先将众多的共享单车出发地与到达地聚类为出发区域和到达区域，本文采用的方法是首先进行 K-means 聚类算法。K-means 是一种非常经典的聚类算法，因其原理简单、可解释性强而得到广泛应用。K-means 算法的聚类过程简单地说就是把数据点按照某种相似度划分到不同的簇中，使得同一簇内的数据点相似度尽可能的高，不同簇间的数据点相似度尽可能低，本文采取欧式距离作为相似性的评价指标，即认为两个对象的距离越近，其相似度就越大，算法原理是给定一个数据点集合和需要的聚类数目 k， K-means 算法根据距离函数反复把数据分入 k 个聚类中。

K-means 算法缺点是需要人为事先指定数据簇的数目 k，而在该问中无法实现确定

最终的聚类簇的数目，因此本文采用多次运行取最优的方法。因为 Geohash 算法中字符串长度为 7 的区域大小为 153 米×153 米，而通常用户寻找共享单车不会超过一公里，因而本文在取数据簇 k 在 800-5000 类范围内进行试验，以早高峰时间段的出发地点分布为例，对早高峰时间段内的出发地点进行聚类，以经纬度坐标之间的欧式距离不超过 1 公里的任意两个聚类中心的平均距离和最小距离大于 459 米（3×153 米）为标准。经过反复试错，发现取数据簇 k = 2000 时聚类效果最好，任意两个聚类中心的平均距离为 893 米，最小距离为 543 米。早高峰到达地点分布、晚高峰出发地点分布和晚高峰到达地点分布的聚类也是在 2000 类时效果最好。下表列出了数据簇 k = 2000 下早高峰出发地点、早高峰到达地点、晚高峰出发地点和晚高峰到达地点聚类的满足欧式距离在 1 公里内的任意两个聚类中心的平均距离和最小距离。

表 1 一公里内任意两个聚类中心的平均距离和最小距离

数据集	平均距离	最小距离
早高峰出发地点聚类	893m	543m
早高峰达到地点聚类	890m	486m
晚高峰出发地点聚类	905m	549m
晚高峰到达地点聚类	905m	580m

故经过反复试错后，本文最终选择以数据簇 k = 2000，对早高峰出发地点和到达地点以及晚高峰出发地点和到达地点分布进行聚类，聚类效果如下图所示：

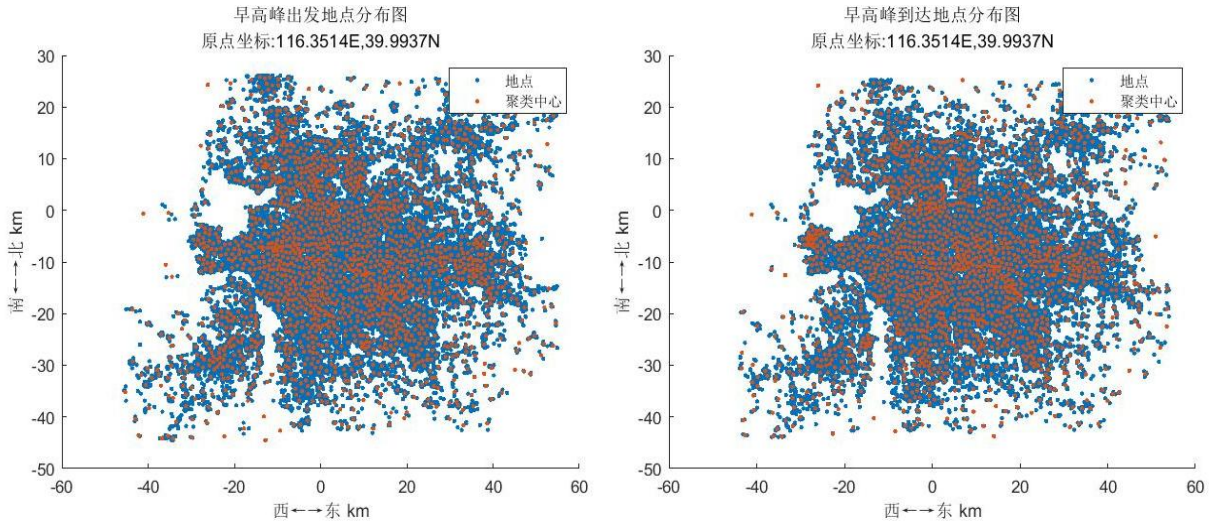

图 3 早高峰出发地点和到达地点欧式分布

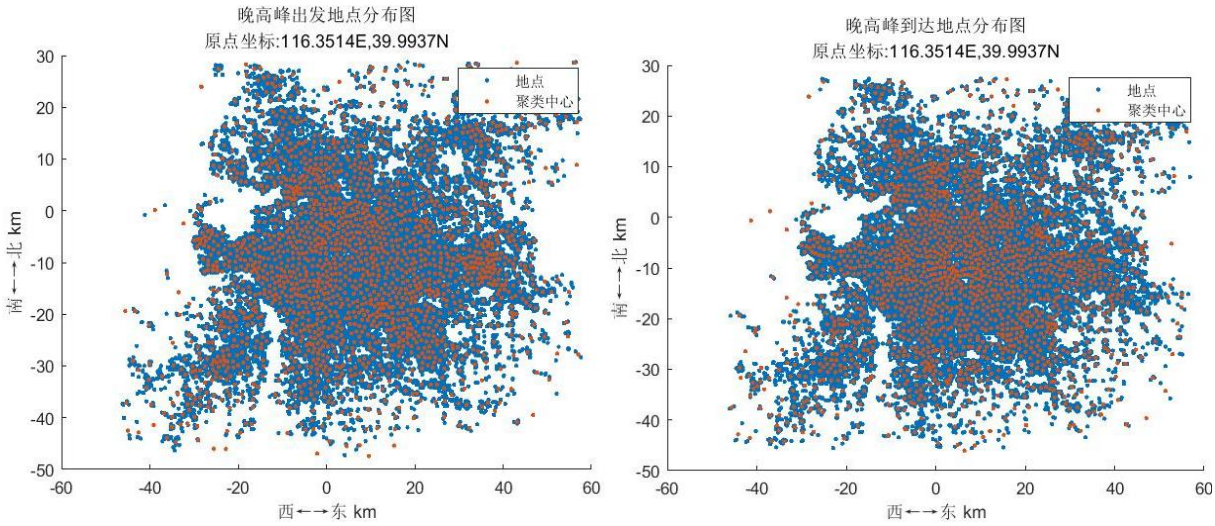

图 4 晚高峰出发地点和到达地点欧式分布

但是由于 K-means 算法选取起始点的随机性，导致每次聚类结果不稳定，差异较大，为优化聚类效果，以求得较为稳定的十个热门区块，采用两层聚类的方法。第一次聚类为之前所述的 K-means 聚类，在前面研究的基础上，从单位点（单位点：geohash 解码后的最小区块）到聚类中心进行 K-means 聚类，选出 10 个热门聚类中心，循环 10 次，共获得 100 个聚类中心。对这 100 个聚类中心进行可视化，以求解晚高峰热门区块为例，观察图 5 可发现虽然 K-means 聚类的结果不稳定，但热门聚类中心有高密度区域，合理猜测在多次循环下会出现数个收敛点。

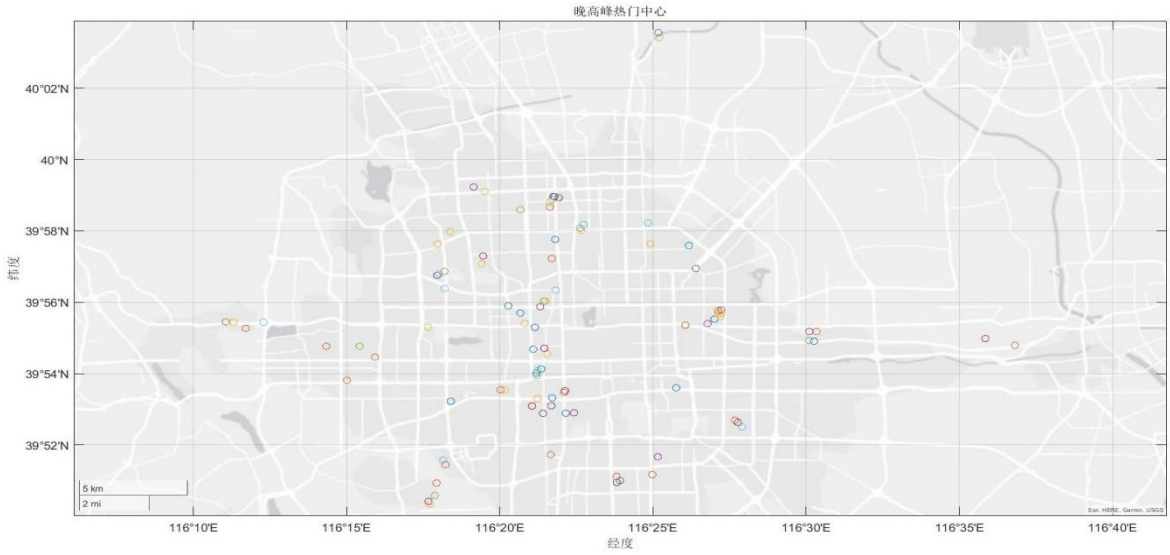

图 5 第一层聚类结果示例

于是在第一次聚类的基础上，进行第二层聚类，采用 DBSCAN 算法，对 K-means 聚类所得的一百个热门聚类中心进行 DBSCAN 二次聚类以确定“收敛点”。

（2）DBSCAN 聚类

DBSCAN（Density-Based Spatial Clustering of Applications with Noise，具有噪声的基于密度的聚类方法）是一种基于密度的空间聚类算法。 该算法将具有足够密度的区域划分为簇，并在具有噪声的空间数据库中发现任意形状的簇，它将簇定义为密度相连的点的最大集合。它是基于一组邻域来描述样本集的紧密程度的，参数(,

min_samples)用来描述邻域的样本分布紧密程度。其中，描述了某一样本的邻域距离阈值，min_samples 描述了某一样本的距离为的邻域中样本个数的阈值。

（2.1）DBSCAN 核心定义

DBSCAN 是基于一组邻域来描述样本集的紧密程度的，参数(ϵ, min_samples)用来描述邻域的样本分布紧密程度。其中，ϵ描述了某一样本的邻域距离阈值，min_samples 描述了某一样本的距离为ϵ的邻域中样本个数的阈值。

假设已知样本集 D = $\{x_i\}_i$，则 DBSCAN 具体的密度和样本描述定义如下：

ϵ邻域:对于$x_i \in D$，其ϵ邻域包含样本集 D 中与x_i的距离不大于ϵ的子样本集，其势记为 N；

核心点:对于任一样本$x_i \in D$,如果其ϵ邻域对应的 N 不小于 min_samples，即如果 N ≥min_samples，则$x_i \in D$是核心点；

密度直达：如果$x_i \in D$位于$x_j \in D$的ϵ邻域中，且x_j是核心对象，则称x_i由x_j密度直达。注意反之不一定成立，即此时不能说x_j由x_i密度直达, 除非x_i也是核心对象；

密度可达：对于x_i和x_j,如果存在样本序列$p_1, p_2, p_3, \ldots, p_n$,满足$p_1 = x_i$，$p_n = j$,且$p_{t+1}$由$p_t$密度直达，则称$x_j$由$x_i$密度可达。也就是说，密度可达满足传递性。此时序列中的传递样本p_t均为核心对象，因为只有核心对象才能使其他样本密度直达。注意密度可达也不满足对称性，这个可以由密度直达的不对称性得出；

密度相连：对于x_i和x_j,如果存在核心对象样本x_k，使x_i和x_j均由x_k密度可达，则称x_i和x_j密度相连。注意密度相连关系是满足对称性的。这一点由密度可达的定义可以很容易得出；

核心点：对某一样本集 D，若样本 p 的ϵ邻域内至少包含 min_samples 个样本（包括样本 p），那么样本 p 称核心点；

边界点：对于非核心点的样本 b，若 b 在任意核心点 p 的ϵ邻域内，那么样本 b 称为边界点；

噪声点：对于非核心点的样本 n，若 n 不在任意核心点 p 的ϵ邻域内，那么样本 n 称为噪声点。

假设 min_samples=4，则下图为以上定义的直观解释。

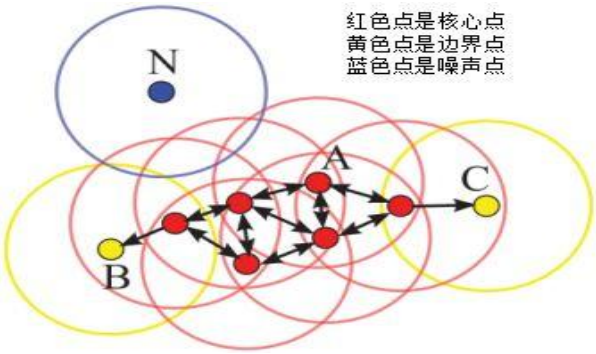

图 6 DBSCAN 核心定义示意图

（2.2）DBSCAN 算法原理

DBSCAN 是基于密度的聚类算法，原理为：只要任意两个样本点是密度直达或密度可达的关系，那么该两个样本点归为同一簇类，上图的样本点 ABCE 为同一簇类。因此，DBSCAN 算法从数据集 D 中随机选择一个核心点作为“种子”，由该种子出发确定相应的聚类簇，当遍历完所有核心点时，算法结束。

DBSCAN 算法可以抽象为以下几步：

1）找到每个样本的邻域内的样本个数，若个数大于等于 min_samples，则该样本

为核心点；

2）找到每个核心样本密度直达和密度可达的样本，且该样本亦为核心样本，忽略所有的非核心样本；

3）若非核心样本在核心样本的邻域内，则非核心样本为边界样本，反之为噪声。

（2.3）参数的调整和二次聚类的结果

由于数据集为 2 维，故设定 min_samples=3。

观察前面第一层聚类结果的图像，发现在热门区域（即图中样本点密集区域）的样本点经纬度坐标下平均距离大约为 0.03，故在 0.01-0.05 区间通过二分法不断调整，使 DBSCAN 聚类的结果为 10 个类别，对每个类别中的元素计算其坐标均值，即认为是该类的中心点，也即所求的 10 个热门中心，二次聚类的结果。

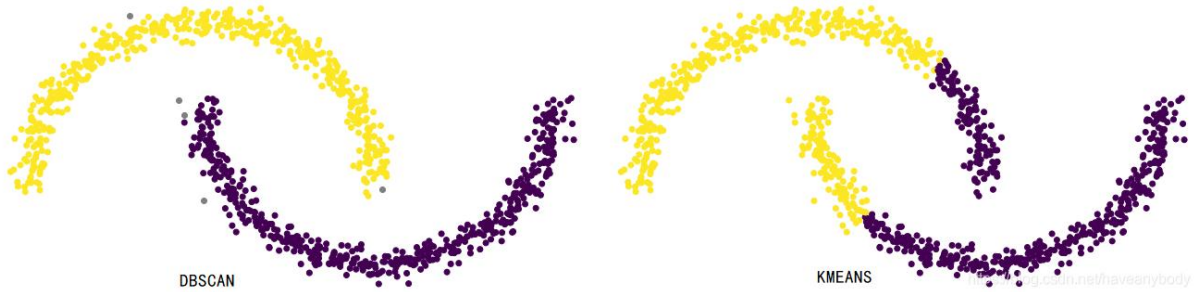

图 7 K-means 聚类和 DBSCAN 聚类效果对比

第二层聚类最终获得的 10 个收敛点如图 8 所示。根据聚类结果，可以得到早高峰和晚高峰时间段内热门区块所代表的位置，如表 2 和表 3 所示。为确定最终的热门聚类中心，还需多次运行检验结果稳定性。图 9 展示了经 DBSCAN 聚类出来的两次早高峰热门中心，表明 DBSCAN 聚类结果较为稳定。

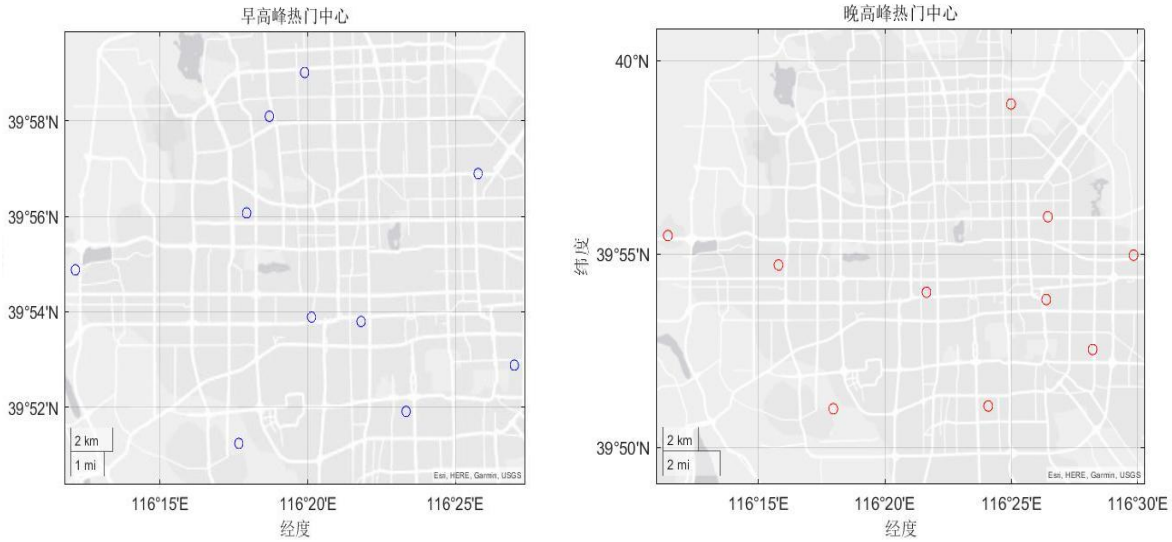

图 8 第二层聚类早高峰和晚高峰热门区块分布

表 2 第二层聚类早高峰热门区块位置

纬度	经度	地址
39.9147	116.202	八角北里
39.9345	116.299	宝联体育公园

39.9683	116.312	中国人民大学信息技术中心
39.9836	116.332	中科院理论物理所
39.9483	116.429	东直门立交
39.8981	116.335	小马厂
39.8966	116.363	北京市府大楼
39.8651	116.389	西革新里
39.8813	116.45	劲松社区

表 3 第二层聚类晚高峰热门区块位置

纬度	经度	地址
39.9247	116.19	杨庄（地铁站）
39.912	116.263	武警总医院
39.85	116.299	北京丰台站
39.9163	116.497	东八里庄
39.9814	116.416	惠新西街北口（地铁站）
39.9328	116.441	北京工人体育馆
39.9003	116.361	长椿街（地铁站）
39.8512	116.401	海户屯（地铁站）
39.8971	116.44	广渠门中学

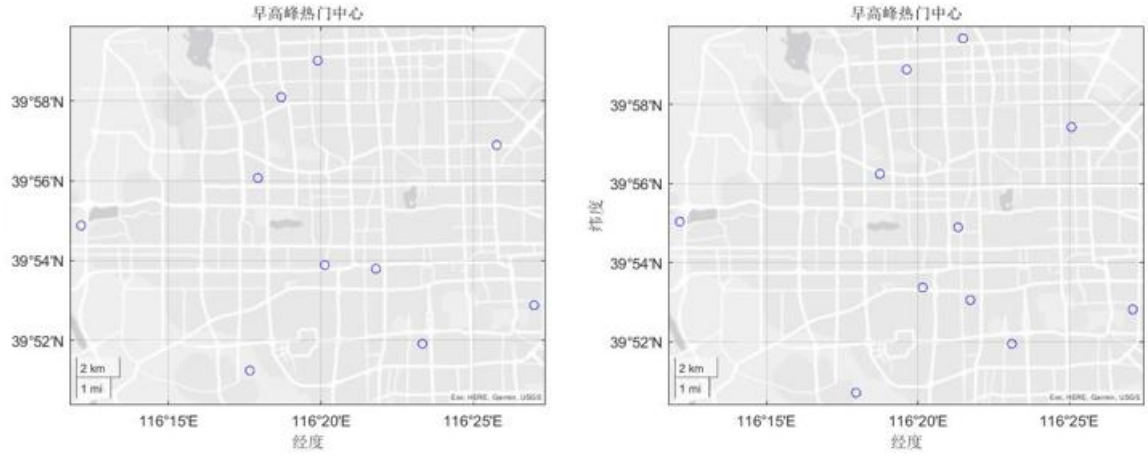

图 9 DBSCAN 聚类的两次结果示例

5.1.3 解法二：区域网格法

由于 K-means 算法随机性较大，导致最终结果不稳定，于是本文同时给出了第二种方法：先统计按照 Geohash 编码所示的每个区域内的单车使用量，每个区域内的单车使用量为该区域及其周边的 8 个区域围成的 3×3 区域内的单车使用量，若某区域处于九宫格的边缘，则该区域内单车使用量即为所需统计的该区域内单车使用量数据，并将统计结果按大小顺序进行排列，得出单车使用量排名前十的区域。但是将统计结果进行可视化后发现存在较为重叠的区域，所以需要取适量个排名靠前的区域（大于十个），直到可视化时这若干个区域形成十个较为独立的大型区域。本文通过 kmeans 算法将这些区域合并为为十个大型区域。经过反复尝试，最终得出早高峰和晚高峰单车使用量排名前十的区域，即为所求早高峰和晚高峰的热门区块，热门区块的经纬度坐标和订单数量如下表所示：

表 4 早高峰热门区块

排序	订单数量	纬度/°N	经度/°E	Geohash 编码
1	3073	39.86595154	116.4585114	wx4ff16，wx4ff14，wx4ff17 wx4ff13，wx4ff15
2	3049	39.90852356	116.4955902	wx4g4bv，wx4g4cm wx4g4ch，wx4g4cj
3	3038	39.92637634	116.17836	wx4e5sq，wx4e5sw
4	2969	39.92225647	116.5010834	wx4g559，wx4g55f wx4g55d，wx4g55c
5	2931	39.92362976	116.4777374	wx4g4eb，wx4g4ec，wx4g4e8
6	2924	39.84535217	116.428299	wx4f9mg，wx4f9mk，wx4f9ms wx4f9me，wx4f9mu，wx4f9mv wx4f9m7，wx4f9mt，wx4f9mm
7	2922	39.90852356	116.5148163	wx4g535，wx4g53h，wx4g52g
8	2897	39.90715027	116.1907196	wx4e5bz，wx4e5by，wx4e5cn wx4e5bx，wx4e5cp
9	2885	39.92362976	116.5175629	wx4g57v
10	2845	39.81239319	116.3665009	wx4drzz，wx4f2p8，wx4drzx wx4f2p2，wx4drzr

表 5 晚高峰热门区块

排序	订单数量	纬度/°N	经度/°E	Geohash 编码
1	3266	39.86595154	116.4585114	wx4ff16，wx4ff14，wx4ff17
2	3248	39.92225647	116.5189362	wx4g57v，wx4g57t wx4g57y，wx4g57w
3	3050	39.948349	116.3733673	wx4g207，wx4g20k，wx4g20s
4	3020	39.84535217	116.3857269	wx4f8mt，wx4f8mm，wx4f8ms wx4f8my，wx4f8mw，wx4f8mq wx4f8mk，wx4f8mv，wx4f8mu

5	2906	39.97306824	116.490097	wx4g6uc，wx4g6ud，wx4g6u9 wx4g6ub，wx4g6u8，wx4g6uf
6	2841	39.94285583	116.3527679	wx4epxm，wx4epxj
7	2802	39.84672546	116.447525	wx4f9vd，wx4f9vf
8	2703	39.84672546	116.3994598	wx4f8tz
9	2469	39.90852356	116.4955902	wx4g4cj
10	2430	39.84535217	116.428299	wx4f9ms，wx4f9mg，wx4f9me wx4f9mk，wx4f9mu，wx4f9mv wx4f9mt，wx4f9mm

在 MATLAB 中通过 geoscatter 函数分别将早高峰热门区块和晚高峰热门区块分布位置图绘制出来，如下图所示：

图 10 早高峰热门区块和晚高峰热门区块分布

5.2 问题二模型的建立与求解

5.2.1 计算早高峰出发到达对应表

通过对 geohash 编码的字段进行统计，计算 train 数据集中早高峰各出发地字段对应的所有到达地字段及其出现次数，选取出现频率最高的字段作为该出发地订单对应的最有可能的目的地。将所有字段转换为经纬度坐标，存储在“早高峰出发到达对应表.xlsx”的 sheet1（以下简称对应表）中。

	A	B	C	D
1	出发经度	出发纬度	到达经度	到达纬度
2	102.7160	25.0426	-179.9993	25.0344
3	103.9849	30.6086	-179.9993	30.6059
4	104.0316	30.5509	-179.9993	30.5427
5	104.0343	30.5853	-179.9993	30.5811
6	104.0467	30.5289	-179.9993	30.5262
7	104.0481	30.5070	-179.9993	30.5180
8	104.0714	30.5138	-179.9993	30.5083
9	104.1387	30.5633	-179.9993	30.5729
10	108.8422	34.2437	-179.9993	34.2368
11	108.9411	34.1517	-179.9993	34.1434
12	109.0661	34.2945	-179.9993	34.2918
13	113.0198	28.2012	-179.9993	28.1930
14	113.0349	28.1202	-179.9993	28.1380
15	113.2395	23.1777	-179.9993	23.1750
16	113.2423	23.1420	-179.9993	23.1448

图 11　早高峰出发到达对应表部分截图

5.2.2 计算 test 数据集中的热门中心

按照第一题相同的思路，通过二层聚类对早高峰各出发地进行聚类，找出其热门聚类中心的经纬度坐标。

5.2.3 查表预测

对 5.2.2 中找出的 10 个热门聚类中心的经纬度坐标在对应表的出发地集合中进行查找，一般情况下只需要查找坐标与其最接近的出发地，其对应的目的地坐标即为最有可能的目的地，将其坐标记录在对应表的 sheet2 中，也可见下方表格。

表 6　早高峰出发到达对应表

出发地经度	出发地纬度	到达地经度	到达地纬度
116.191	39.9213	116.1852264	39.92225647
116.265	39.9074	116.2607574	39.91539001
116.307	39.9614	116.3156891	39.95246887
116.377	39.9754	116.4008331	39.97306824
116.346	39.9092	116.3665009	39.91264343
116.417	39.9875	116.4241791	39.98542786
116.440	39.9599	116.4324188	39.95521545
116.508	39.9153	116.5161896	39.92225647
116.460	39.8819	116.4613	39.9168
116.404	39.8560	116.4008	39.8536

将 10 个热门中心的坐标和 10 个目的地坐标利用 geoscatter 函数绘图，得出 test 数据集中早高峰出发地的热门聚类中心及其最有可能的目的地的分布图，如图 12 所示。

图 12　test 数据集早高峰热门出发地及其到达地预测

5.3 问题三模型的建立与求解

问题三的要求是根据前两问结果判断是否需要人工调节热门区块单车数量，若需要调节，考虑在满足人群出现需求的条件下以最低成本调节。

第一步，构建判断是否需要人工调节的思路。考虑早高峰和晚高峰的热力区块共

二十个，发现早高峰的热力区块与晚高峰的热力区块有三块较为重叠，将较为重叠的区块记为一个区块，于是得到十七个热力区块。对于所求出的热力区块，分别计算其每周七天各自的早高峰和晚高峰时的自行车变化量，即作为终点的次数减去作为起点的次数，差值为正则数量净增长，差值为负则负增长。若正增长，则在热门区域的客户需求可以得到充分满足；若负增长，则热门区域可能出现供不应求的现象，此时即需要进行人工调度，填补其负增长量。为简化模型，本文仅考虑在热力区块之间进行调度。

第二步，建立线性规划模型。将正增长的区块视为产地，将负增长的区块视为销地，如此，将单车调度问题化为运输问题，目标为运输总路径最短，以所有负增长区块的负增长量被补为 0 为约束条件，基于运输问题建立线性规划模型后，在 MATLAB 中进行线性规划求解。

通过对每天所有区块的净增长量分别求和可以发现，所考虑的 17 个区块总是净增长量大于净负增长量，即“产量”大于“销量”，故本问题适用产销不平衡的运输模型。

产销不平衡的运输模型为：

$$\min z = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \quad s.t. \begin{cases} \sum_{j=1}^{n} x_{ij} \le a_i, i = 1\dots m \\ \sum_{i=1}^{m} x_{ij} = b_j, j = 1\dots n \\ x_{ij} \ge 0 \end{cases} \tag{1}$$

其中，共有 m 个产地，n 个销地，x_{ij}表示从第 i 个产地到第 j 个销地的运输量，为非负实数；向量 C=($c_{11}, c_{12}\dots c_{1n}, c_{21}, c_{22}\dots c_{2n}\dots c_{m1}, c_{m2}, c_{mn}$)为运价向量，即从产地到销地的距离向量；向量 A=(a_i)为产量向量，B=(b_j)为销量向量，为净负增长量的-1 倍；

将 17 个地点按净增长量的正负分为 m 个产地和 n 个销地，并通过编程找出对应的几个参数向量，即建立线性规划模型，利用 MATLAB 的 linprog 函数求解，即得解向量 X。再将 X 转换为矩阵格式即可得到运输方案表。

以星期一的数据为例展示求解结果如下：

地点编号	纬度	经度	净增长量
12	39.92225647	116.5189362	-80
14	39.84535217	116.3857269	-39.5
4	39.92225647	116.5010834	-28
9	39.92362976	116.5175629	-26.5
17	39.84672546	116.447525	-22.5
13	39.948349	116.3733673	-11
16	39.94285583	116.3527679	-5
5	39.92362976	116.4777374	4.5
2	39.90852356	116.4955902	13
7	39.90852356	116.5148163	18.5
1	40.2545929	116.4585114	21
8	39.90715027	116.1907196	69.5
3	39.92637634	116.17836	73
10	39.81239319	116.3665009	75.5

6	39.84535217	116.428299	81
15	39.97306824	116.490097	103.5

产地序号＼销地序号	12	14	4	9	17	13	16
5	0	0	0	0	0	0	0
2	0	0	0	0	13	0	0
7	0	18.5	0	0	0	0	0
1	0	0	21	0	0	0	0
8	4.5	0	0	0	0	0	0
3	0	0	0	0	9.5	0	0
10	75.5	0	0	0	0	0	0
6	0	21	7	26.5	0	11	5
15	0	0	0	0	0	0	0

完整结果请见支撑材料：运输矩阵.xlsx ,we1.xls

六、模型的检验和评价

6.1 模型的优点

1. DBSCAN 不需要指定簇类的数量；
2. DBSCAN 可以处理任意形状的簇类；
3. DBSCAN 可以检测数据集的噪声，且对数据集中的异常点不敏感；
4.由图 12 可检验第二题的预测结果符合共享单车适合短距离出行的生活经验，其预测结果较为准确
5.对于第一题热门区块的求解结果，查询了坐标对应的实际地址如表 2、3，可发现确实为北京市较大的居民区或功能地点，符合现实情况

6.2 模型的缺点

1.K-means 算法需要人为指定聚类数目 k，实际问题中无法实现确定最终的 k，需要反复试错，提高了计算的代价；
2.K-means 对于非球形数据集的聚类效果不佳，实际骑车和停车地点分布情况一般是呈非球形分布的；
3.由于 K-means 难以避免的不稳定性，即使二层聚类很大程度上削减了其影响，也不足以消除，实际二层聚类的结果仍有小幅波动;
3.第三题中只考虑了热门区块之间的调运方案，但事实上热门区块的调运应当从附近调运成本更低，而不是选择可能更远的其他热门区块;
4.第二题中对应关系表的预测方式比较简单，并不能精确地反映用户的目的地分布情况，一定程度上缺乏严谨性

参考文献

[1]洪文兴,陈明韬,刘伊灵等.基于 GeoHash 和 HDBSCAN 的共享单车停车拥挤区域识别[J].厦门大学学报(自然科学版),2022,61(06):1030-1037.
[2]宋飞宇. 考虑需求动态变化的共享单车调度问题研究[D].北京交通大学,2022.DOI:10.26944/d.cnki.gbfju.2021.002427.

附录

附录 1

介绍：Geohash 编码解码为经纬度的代码

```
%编码地址解码为经纬度
function [lat,lot]=docode(geoh)
Base32= '0123456789bcdefghjkmnpqrstuvwxyz';
odd = true ;
latitude=[-90,90];
longitude=[-180, 180];
for i=1:7
  for j=1:32
        if Base32(j)==geoh(i)
           bits=j-1;
           break;
        end
  end
   for jj=1:5
        j=6-jj;
      switch j
        case 5
           ad=16;
        case 4
           ad=8;
        case 3
           ad=4;
        case 2
           ad=2;
        case 1
           ad=1;
      end
       if  bitand(bits,ad)
          bit=1;
       else
          bit=0;
       end
        if odd
           mid = (longitude(1) + longitude(2)) / 2;
           if bit==0
              longitude(2)= mid;
           else
              longitude(1)= mid;
           end
        else
            mid = (latitude(1) + latitude(2)) / 2;
           if bit==0
              latitude(2)= mid;
           else
              latitude(1)= mid;
```

```
            end
        end
        odd = ~odd;
    end
end
    lat = (latitude(1) + latitude(2)) / 2;
    lot = (longitude(1) + longitude(2)) / 2;
end
```

附录 2

介绍：问题一中解法一实现代码

```
    %% 初始化参数
    clear;
    clc;
    go=readmatrix('早高峰.xlsx','Sheet','出发');%导入您的数据，如果只包
    % 含一组出发或到达的数据，请在 arr 类变量前全部加上注释符号
    arr=readmatrix('早高峰.xlsx','Sheet','到达');
    c=zeros(1,3);

    %% 读取两点的坐标并删除离群值(经度在前，纬度在后)
    ego=rmoutliers(go);
    earr=rmoutliers(arr);
    lego=length(ego);
    learr=length(earr);

    for i=1:20
    disp(i)%展示循环进度到第 i 次

    %% 第一层聚类
    [idx,vgo]=kmeans(ego,2000);
    lvgo=length(vgo);
    xvgo=vgo(:,1);
    yvgo=vgo(:,2);
    [idy,varr]=kmeans(earr,2000);
    lvarr=length(varr);
    xvarr=varr(:,1);
    yvarr=varr(:,2);

    %% 计算每一类的热力值，选出前 10 位热门中心
    for i=1:lego
    vgo(idx(i),3)=vgo(idx(i),3)+ego(i,3);
    end

    for i=1:learr
    varr(idy(i),3)=varr(idy(i),3)+earr(i,3);
    end

    v=[vgo
    varr];
    v=sortrows(v,3,"descend");
    v=v(1:10,:);
    c=[c
    v];%如果要将出发与到达分开展示，请将 v，c 复制，并分别命名为 go 和 arr 类变量

    end
```

```
%% 第二层聚类
% 在第一层聚类的大循环执行完后，您只需要对本小节不断运行以调试，无需全局 debug
clc
c=c(1:200,1:2);
[id]=dbscan(c,0.01412,3);%调整参数以使 idmax=10，扩大第二个参数会使 idmax 减少
% ，缩小会使其增大，使用二分法或可以使其接近 10
idmax=max(id);%第二层聚类数
center=zeros(idmax,2);
for i=1:idmax
for j=1:200
if id(j)==i
if center(i,:)==[0 0]
center(i,:)=c(j,:);
else
center(i,:)=(center(i,:)+c(j,:))/2;
end
end
end
end

x=center(:,1);
y=center(:,2);
geoscatter(y,x,'blue');
hold on
%如果您将到达和出发分别输出，请使用不同颜色，并用 legend 函数加以标注
title('早高峰热门中心');
```

附录 3

介绍：问题一中解法二实现代码

```
网格区域法
morninglocation=[morningstartlocation;morningendlocation];
morningpinlv=tabulate(morninglocation);
morningpinlv1=cell2mat(morningpinlv(:,2));
morninglocation1=string(morningpinlv(:,1));
morninglocationallnum=zeros(2,length(morninglocation));
morninglocationnum=zeros(2,length(morninglocation1));
for i=1:length(morninglocation1)
      [a,b]=docode(char(morninglocation1(i)));
      morninglocationnum(1,i)=a;
      morninglocationnum(2,i)=b;
end
morninglocationx=morninglocationnum(1,:);
morninglocationy=morninglocationnum(2,:);
x=unique(morninglocationx);
y=unique(morninglocationy);
n=length(x);
m=length(y);
ninemorning=zeros(1,length(morninglocation1));
morningan=zeros(1,length(morninglocation1));
for i=1:length(morninglocation1)
    a1=morningpinlv1(i);
    if
(find(x==morninglocationx(i))==1)||(find(x==morninglocationx(i))==n)||(find(y==morninglocat
iony(i))==1)||(find(y==morninglocationy(i))==m)
        ninemorning(i)=a1;
        break
    end
    morningan(i)=(a1==morningpinlv1(i));
    a2=morningpinlv1(intersect(find(morninglocationx==x(find(x==morninglocationx(i))-
```

```
1)),find(morninglocationy==y(find(y==morninglocationy(i))-1))));

a3=morningpinlv1(intersect(find(morninglocationx==x(find(x==morninglocationx(i))+1)),find(m
orninglocationy==y(find(y==morninglocationy(i))+1))));

a4=morningpinlv1(intersect(find(morninglocationx==x(find(x==morninglocationx(i))+1)),find(m
orninglocationy==y(find(y==morninglocationy(i))-1))));
        a5=morningpinlv1(intersect(find(morninglocationx==x(find(x==morninglocationx(i))-
1)),find(morninglocationy==y(find(y==morninglocationy(i))+1))));

a6=morningpinlv1(intersect(find(morninglocationx==x(x==morninglocationx(i))),find(morninglo
cationy==y(find(y==morninglocationy(i))-1))));
        a7=morningpinlv1(intersect(find(morninglocationx==x(find(x==morninglocationx(i))-
1)),find(morninglocationy==y((y==morninglocationy(i))))));

a8=morningpinlv1(intersect(find(morninglocationx==x(x==morninglocationx(i))),find(morninglo
cationy==y(find(y==morninglocationy(i))+1))));

a9=morningpinlv1(intersect(find(morninglocationx==x(find(x==morninglocationx(i))+1)),find(m
orninglocationy==y(y==morninglocationy(i)))));
        ninemorning(i)=sum([a1 a2 a3 a4 a5 a6 a7 a8 a9]);
    end
    ninemorningshunxu=sort(ninemorning,'descend');
    morningmaxnine=ninemorningshunxu([1:41]);
    morningxuhao=[];
    for i=1:41
       morningxuhao=[morningxuhao find(ninemorning==morningmaxnine(i))];
    end
    morningxuhao=unique(morningxuhao);
    morningweizhi=morninglocationnum(:,morningxuhao);
    loc=morningweizhi;
    geoscatter(morningweizhi(1,:),morningweizhi(2,:));
```

附录 4

介绍：问题二核心代码

```
%% 初始化参数
clear;
load mor_test.mat
clc;
go=[mor_test(:,3) mor_test(:,2) mor_test(:,1)];
%arr=readmatrix('晚高峰.xlsx','Sheet','到达');
cgo=zeros(1,3);
%carr=zeros(1,3);
%% 读取两点的坐标并删除离群值(经度在前，纬度在后)
ego=rmoutliers(go);
%earr=rmoutliers(arr);
lego=length(ego);
%learr=length(earr);

for i=1:10
%% 聚类分析
[idx,vgo]=kmeans(ego,2000);
lvgo=length(vgo);
xvgo=vgo(:,1);
yvgo=vgo(:,2);
%{
[idy,varr]=kmeans(earr,2000);
lvarr=length(varr);
xvarr=varr(:,1);
yvarr=varr(:,2);
%}
```

```
%% 计算每一类的热力值，选出前 10 位热门中心
for i=1:lego
    vgo(idx(i),3)=vgo(idx(i),3)+ego(i,3);
end
%{
for i=1:learr
    varr(idy(i),3)=varr(idy(i),3)+earr(i,3);
end
%}

vgo=sortrows(vgo,3,"descend");
vgo=vgo(1:10,:);
cgo=[cgo
     vgo];

%{
varr=sortrows(varr,3,"descend");
varr=varr(1:10,:);
carr=[carr
      varr];
%}
end

%% 第二层聚类
clc
cgo=cgo(1:100,1:2);
%carr=carr(1:100,1:2);
[idgo]=dbscan(cgo,0.025,3);%调整参数以使 idgomax=10
%[idarr]=dbscan(carr,0.01,3);%调整参数以使 idarrmax=10
idgomax=max(idgo);
%idarrmax=max(idarr);
centergo=zeros(idgomax,2);
%centerarr=zeros(idarrmax,2);
for i=1:idgomax
    for j=1:100
        if idgo(j)==i
            if centergo(i,:)==[0 0]
                centergo(i,:)=cgo(j,:);
            else
                centergo(i,:)=(centergo(i,:)+cgo(j,:))/2;
            end
        end
    end
end
%{
for i=1:idarrmax
    for j=1:100
        if idarr(j)==i
        if centerarr(i,:)==[0 0]
            centerarr(i,:)=carr(j,:);
        else
            centerarr(i,:)=(centerarr(i,:)+carr(j,:))/2;
        end
        end
    end
end
%}
```

```
xgo=centergo(:,1);
ygo=centergo(:,2);
%xarr=centerarr(:,1);
%yarr=centerarr(:,2);
geoscatter(ygo,xgo,'red');
hold on
%geoscatter(yarr,xarr,'blue');
legend('出发')
hold on
title('test 集 早高峰热门出发地');
```

附录 5

介绍：问题三求解代码

```
%% 数据初始化
clear
clc
close all
dis=readmatrix("dis1.xls");
we=readmatrix("we1.xls");
we=we';
dis=dis';
y=zeros(63,7);

s=4;%更改此参数以生成第 s-3 天的运输矩阵
%% 产量销量
c=sortrows([we(:,1) we(:,s)],2);
d=[0 0];%产地及其地址
e=[0 0];%销地及其地址
for i=1:16
if c(i,2)>=0
if d==[0 0]
d=c(i,:);
else
d=[d
c(i,:)];
end
else
if e==[0 0]
e=c(i,:);
else
e=[e
c(i,:)];
end
end
end
ld=length(d);
le=length(e);

%% 运价表
g=[];
for i=1:ld
for j=1:le
g(i,j)=dis(d(i,1),e(j,1));
end
```

```
end

%% 系数矩阵
m1=ld;
n1=le;
a=[];
for i=1:m1
a1=zeros(m1,n1);
a1(i,:)=ones(1,n1);
a=[a a1];
end
b1=[];
for i=1:m1
a10=eye(n1,n1);
b1=[b1 a10];
end
fin=[a;b1];

%% 求解线性规划
f=g(1,:);
for i=2:ld
f=[f g(i,:)];
end

A=fin(1:ld,:);
b=d(:,2);
lb=zeros(le*ld);
Aeq=fin(ld+1:ld+le,:);
beq=-e(:,2);
ub=[];
x = linprog(f,A,b,Aeq,beq,lb,ub);
y=zeros(ld,le);
for i=1:ld
for j=1:le
y(i,j)=x((i-1)*le+j);
end
end
y=[d(:,1) y];
y=[0 e(:,1)'
y];
```