****  **** 

**中国研究生创新实践系列大赛**

**中国光谷·“华为杯”第十九届中国研究生**

**数学建模竞赛**

|  |  |
| --- | --- |
| **学 校** | **上海大学** |
| **参赛队号** |  |
| **队员姓名** | **1.** |
| **2.** |
| **3.** |

**中国研究生创新实践系列大赛**

**中国光谷·“华为杯”第十九届中国研究生**

**数学建模竞赛**

题 目 **COVID-19疫情期间生活物资的科学管理问题**

摘 要：

# 问题重述

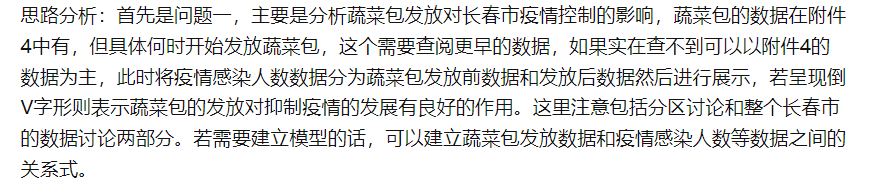
## 问题背景

## 问题重述

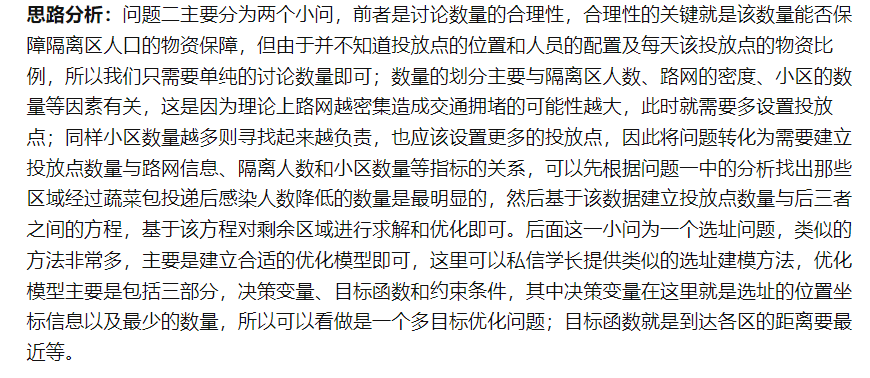
本题给出了长春市COVID-19疫情期间9个区感染人数、隔离人口数量、生活物资投放点、交通网络及各区生活物资和蔬菜包相关数据，基于这些数据解决以下问题：

问题一：针对附件1的数据，判别和分析长春市实施蔬菜包前后的防疫效果。

【整体、9个区画图分析】 设计评价指标？

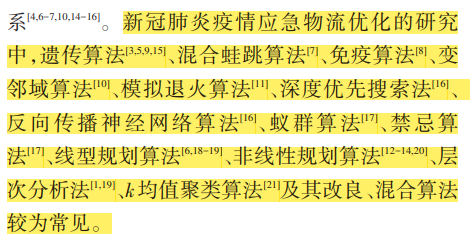


问题二：针对附件2、3、4的数据，讨论生活物资投放点数量设置的合理性，并考虑疫情发展、自然灾害等特殊情况，给出最优的选址数量、规模及潜在的备用场所位置。、



【子问题1. 建立投放点数量与路网信息、隔离人数和小区数量等指标的关系

子问题2. 建立多目标选址优化问题】



<https://blog.csdn.net/TIQCmatlab/article/details/125510530>

<https://download.csdn.net/download/niujianglei/426436?spm=1001.2101.3001.6650.1&utm_medium=distribute.pc_relevant.none-task-download-2%7Edefault%7ECTRLIST%7EPaid-1-426436-blog-125510530.pc_relevant_3mothn_strategy_recovery&depth_1-utm_source=distribute.pc_relevant.none-task-download-2%7Edefault%7ECTRLIST%7EPaid-1-426436-blog-125510530.pc_relevant_3mothn_strategy_recovery&utm_relevant_index=2>

**基于城市大数据的应急服务设施多目标优化选址模型研究\_李金泽**

问题三：针对附件3、5的数据，根据蔬菜包需求及各小区的位置和人口信息，评价并调整4月10号至4月15号蔬菜包供应方案。

【采购量和发放量的关系 ARMA算法】

问题四：针对附件3的数据，讨论大规模封控情境下居民生活物资的发放方案。构建各物资来源、物资集散地、长春市所有小区的有序网络图，考虑卡车运送物资方案，结合指标分析和评价讨论所提发放预案的优势。

【考虑分级的构建配送中心-仓储店-社区需求点的三级网络，以提高客户满意度和降低总的配送成本为优化目标。】

**突发疫情下考虑需求分级的蔬菜社区配送研究\_张春燕**

# 模型假设和符号说明

## 模型假设

## 符号说明

# 问题一的分析与求解

## 问题分析与计算过程

针对附件1和附件5中的数据进行整理，我们发现长春市在3月26日开始实行蔬菜包发放，为对比蔬菜包前后效果，我们首先绘制了全市的新冠感染人数趋势图，观察实施蔬菜包发放前和实行蔬菜包发放后的人数趋势。其次，我们也根据不同区域绘制了新冠感染人数趋势图，分析不同区在发放蔬菜包前后的疫情发展趋势。

图 1全市感染人数趋势图

如图1所示，在疫情开始第30天时全市总感染人数达到顶峰，在疫情开始后第23天长春市开始发放蔬菜包。由于新冠疫情存在着潜伏期一般为1-14天，多数为3-7天，所以我们认为第30天的感染人数大部分在23号前就已经感染新冠病毒。在疫情感染人数达到顶峰后感染人数逐渐下降，直到附件1中的9个区感染总人数全部为0。可以发现发放蔬菜包对于抑制全市的疫情扩散有积极作用，同时我们画出了长春市不同行政区感染人数趋势图，用于对比不同区之间发放蔬菜包前后的效果。

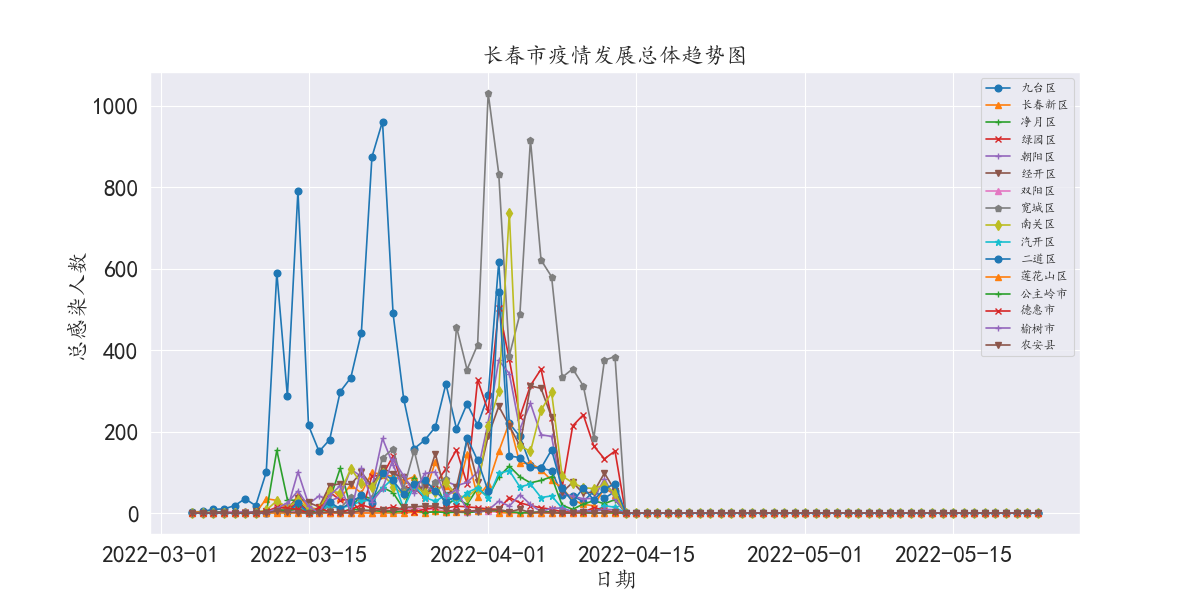


图 2长春市不同行政区感染人数趋势图

以九台区为例，九台区为所有行政区中最先出现疫情的区域同时也是总感染人数最多的区域总感染人数达到9547人，我们观察到九台区在3月14日和3月22日出现了总感染人数的高峰，在4月1日同时也出现了一个小的高峰，但是相比于3月14日和3月22日有明显的下降，这也说明了发放蔬菜包对于抑制疫情态势的升级具有良好的效果。

## 问题求解结果

从上方的结果分析，我们认为发放蔬菜包对于抑制疫情有良好的效果。原因分析：

* 发放蔬菜包解决了长春市居民的基本需求，同时保证了市民们的身体机能和免疫力保持在正常水平有助于抵抗新冠病毒。
* 发放蔬菜包减少了居民外出购物的次数，同时减少了菜市场，超市等人群聚集地区的人流量，大大地减少了感染新冠疫情的几率。
* 集中配送蔬菜包避免了市民因恐慌而造成的哄抢物资，从而破坏市场环境造成物价飞升。有力的控制了物价保障了民生。

# 问题二的分析与求解

## 问题分析

在疫情初期为保证市民的基本生活需求，长春市开始设立投放点用于发放蔬菜包，疫情期间为了减少人员流动和接触同时需要大量的人力资源用于蔬菜包的发放，投放点的数量和选址就显得极为重要。根据附件2中提供的长春市不同区域投放点数量，我们画出了各区投放点情况柱状图，同时我们画出了各区的投放点数目和感染人数占比的趋势图，通过两张图我们对投放点个数进行讨论。

## 投放点数量合理性分析

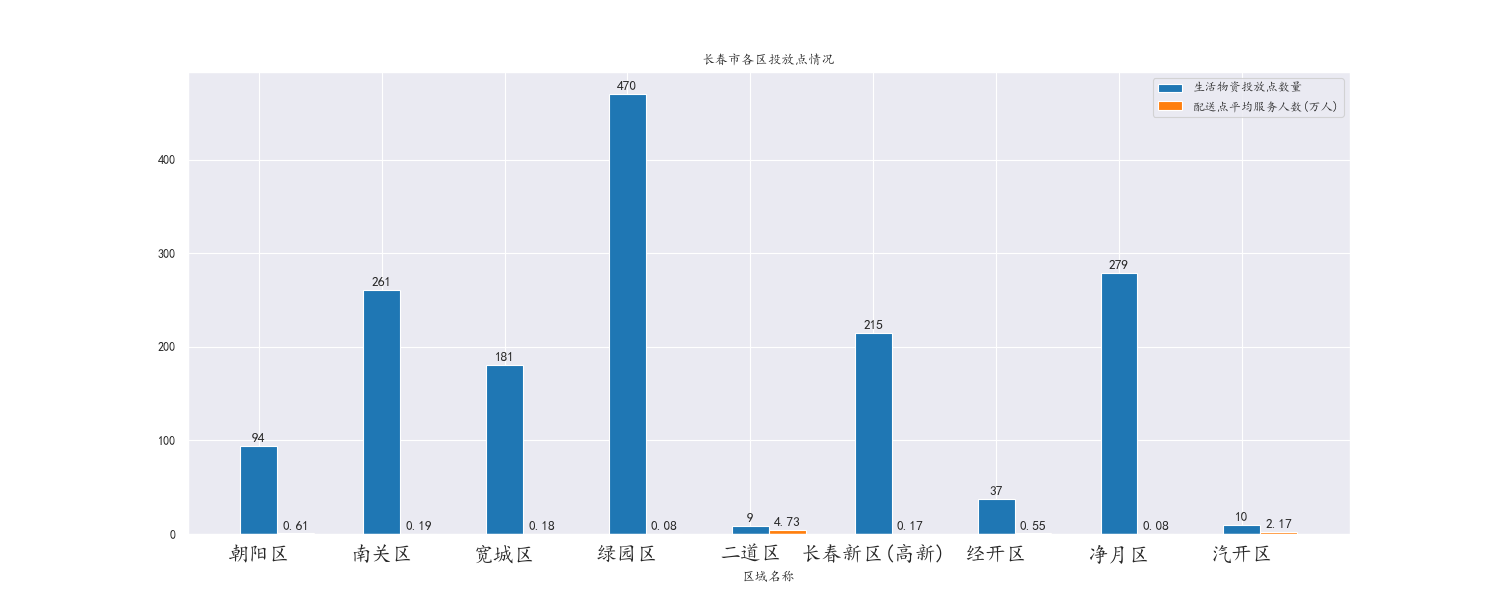


图 3长春市各区投放点个数和投放点平均服务人数

通过图3可以看到绿园区的投放点数最多为470个，二道区的投放点数最少为9个。每个配送点平均服务人数最多的为二道区为4.73万人，最少的为绿园区为0.08万人。从节省人力资源的角度来考虑，我们人数为绿园区的投放点数设置的过多，造成人力资源的浪费，同时在蔬菜配送过程中需要多次运输造成了人员接触次数的增加。



图 4投放点个数与总感染人数占比关系图

从图4可以看出感染人数占比最高的是宽城区，感染人数占比最低的三个区域分别为朝阳区、二道区和汽开区，同时这三个区的每万人平均投放点个数也是偏低的。同时，从地理位置方面考虑，我们发现南关区和二道区和朝阳区都接壤，但是其平均投放点个数远高于其他两个区，但是其总感染人数占比也比另外两个区要高。

综上所诉，我们认为配送点的个数过大会带来人员的过多接触同时占用大量的人力资源，不利于对抗疫情。

## 选址优化分析

上文中我们对长春市疫情期间配送点的数量进行了分区讨论，得出配送点个数过大会带来总感染人数占比的上升这一结论。下面就对投送点个数和位置进行优化。在优化过程中我们是用附件3中的数据绘制出了长春市的路网图，同时我们用K-means算法对长春市的小区进行聚类，并使用平均轮廓系数作为聚类效果的评估指标，运用贝叶斯优化对K-means模型的超参数进行探索，寻找到最优的参数。

* K-means聚类算法

1. means聚类算法是J.B.MacQueen在1967年提出，是经典的并被广泛使用的聚类算法，K-means是基于距离的聚类算法采用欧式距离或者余弦距离作为评价指标，来评价两个对象的相似程度。根据相似程度将对象划分为K个簇，通过不断地循环达到收敛最终得到K个聚类中心。K-means算法流程图如图5所示：

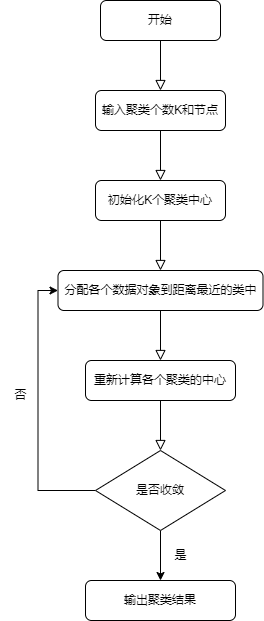


图 5 K-means算法流程图

* 平均轮廓系数

平均轮廓系数是聚类效果好坏的一种评价指标，最早由Peter J. Rousseeuw在1986年提出，它结合了内聚度和分离度两个因素。其计算计算方法如下：

对于某一个节点来说，表示该点与本类其他点的平均距离，为该点与非本类点的平均距离值。平滑轮廓系数的取值范围为[-1, 1]，越接近1表示分类越优秀。

* 贝叶斯优化

目前机器学习领域中常见的超参数优化方法包括网格搜索、随机网格搜索、Halving网格搜索与贝叶斯优化。其中，网格优化、随机网格优化与Halving网格搜索都是在一个大的参数空间中，尽量对所有的点经行验证后返回最优的参数，这一类方法存在着计算时间长的缺点。贝叶斯优化最初是由J Snoek et.在 NIPS 2012 中提出的，并经行了多次的改进。贝叶斯优化是一种近似逼近的方法，用各种代理函数来拟合超参数与模型评价之间的关系，然后选择有希望的超参数组合进行迭代，最后得出效果最好的超参数组合。贝叶斯优化问题由目标函数、域空间、优化算法和结果历史记录四部分组成，其计算流程图如图6所示：

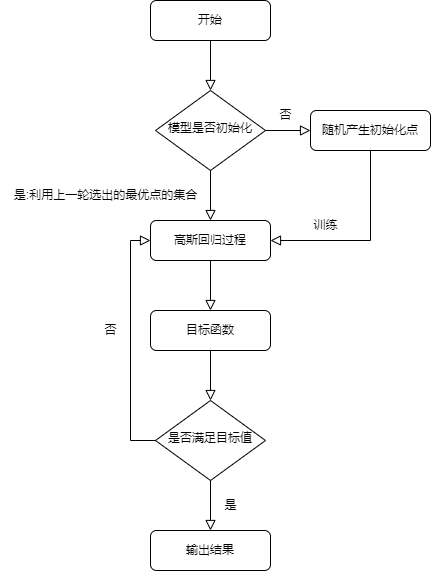


图 6 贝叶斯优化流程图

上面简单介绍了我们选用的方法和其计算步骤，我们依据附件3中的内容根据将小区的坐标作为特征进行K-means聚类，同时我们用贝叶斯优化将聚类的簇数固定在[70, 140]，K-means算法循环次数固定在[300, 1000]，不同初始化质心运行算法次数[10, 50]，用平均轮廓系数作为评价标准。最终我们得到的最优超参数见表1，代码详见附录A。

表格 1 K-means最优超参数

|  |  |  |
| --- | --- | --- |
| 簇数 | 循环次数 | 不同初始化质心运行算法次数 |
| 79 | 672 | 12 |

经过K-means聚类后得到的簇中心点坐标就是我们想要的配送点位置，我们根据附件3中的内容将长春市的路网图和小区分布进行了可视化，见图7。图中黄色的为道路，蓝色的为小区

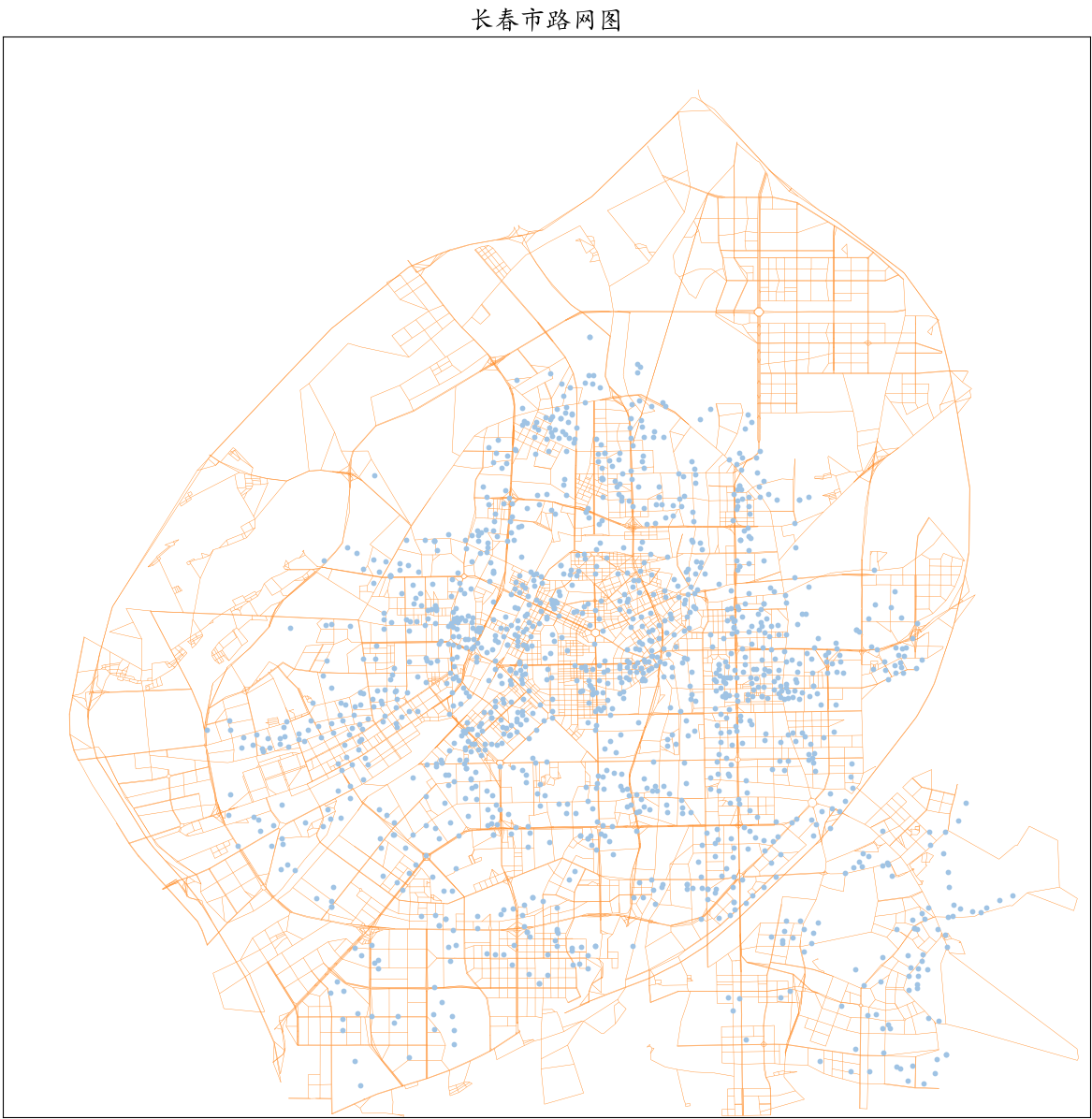


图 7 长春市路网图和小区分布

在可视化出长春市路网图和小区分布过后，我们将预测的配送点位置也进行了可视化，见图8。

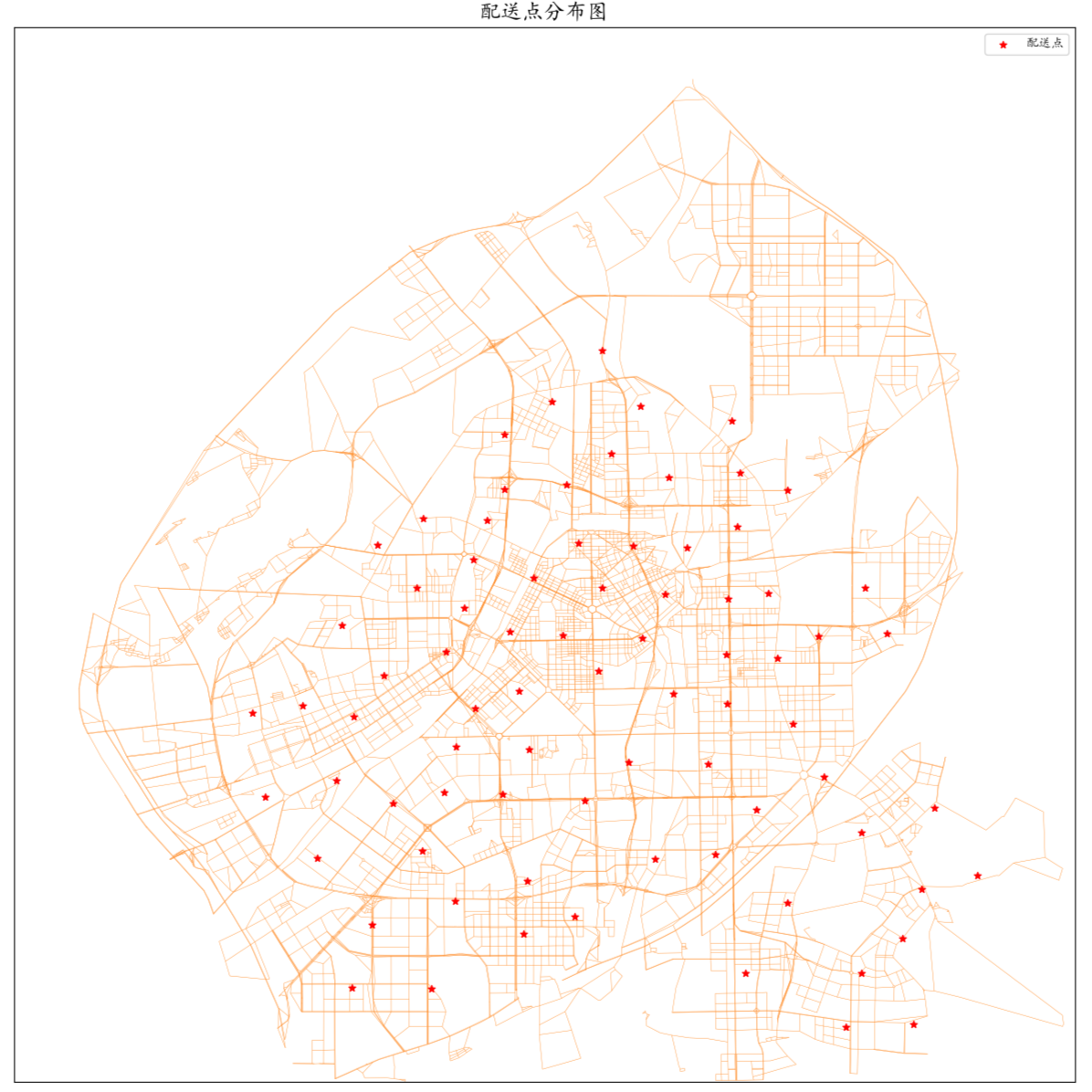


图 8 配送点分布图

图8中的红色五角星标记就是配送点位置，我们同时也将小区的聚类情况进行了可视化，见图9。

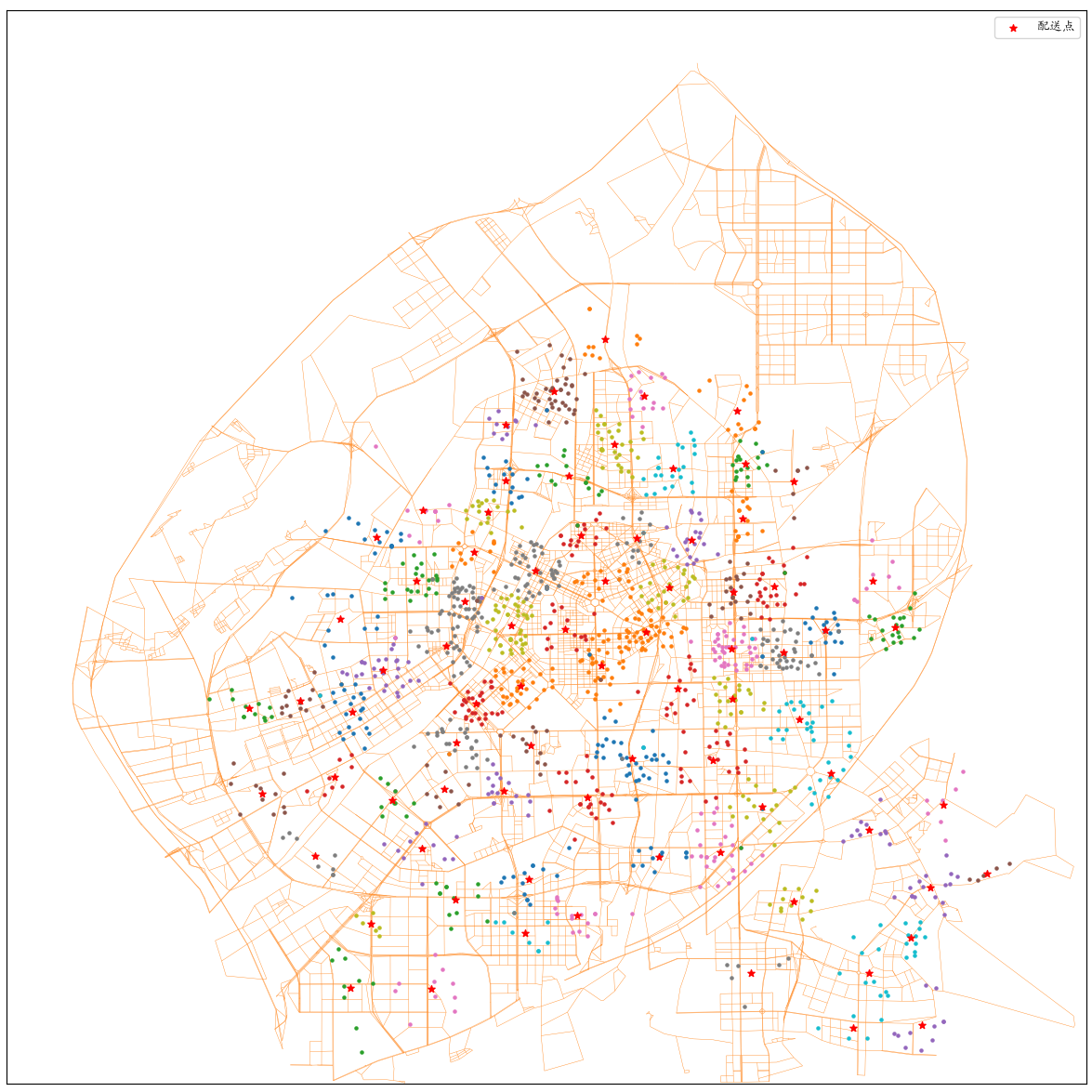


图 9 长春市小区聚类效果图

图9中将1409个小区聚类成了79个簇，每个簇用相同的颜色进行表示，其中红色五角星就是每个小区簇的配送点。

## 结果

通过上面的可视化分析，我们可以看到在长春市市中心小区比较密集，靠近城市边缘则小区个数较少，配送点和小区分布大致相同。由于没有每个区域的范围信息，无法确定配送点的所属区域，所以我们用每个配送点管辖小区的所属区域中出现频次最多的区域作为该配送点的所属区域。每个配送点的具体信息详见表2。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 配送点横坐标 | 配送点纵坐标 | 配送点管辖人数 | 所属区域 | 管辖小区个数 |
| 57.92 | 30.32 | 63,374 | 南关区 | 33 |
| 43.27 | 49.55 | 24,978 | 绿园区 | 13 |
| 68.41 | 57.74 | 45,327 | 二道区 | 18 |
| 30.39 | 28.54 | 17,600 | 汽开区 | 8 |
| 79.86 | 23.66 | 14,697 | 净月区 | 11 |
| 67.34 | 45.83 | 54,153 | 二道区 | 23 |
| 39.35 | 8.81 | 74,454 | 长春新区(高新) | 10 |
| 41.68 | 31.81 | 67,112 | 朝阳区 | 25 |
| 56.34 | 59.54 | 36,243 | 宽城区 | 30 |
| 78.43 | 5.19 | 5,389 | 净月区 | 5 |
| 48.40 | 19.03 | 25,482 | 南关区 | 14 |
| 55.07 | 39 | 95,698 | 朝阳区 | 35 |
| 82.27 | 42.49 | 30,346 | 经开区 | 20 |
| 65.41 | 30.12 | 38,529 | 经开区 | 21 |
| 34.90 | 38.50 | 31,303 | 汽开区 | 29 |
| 50.70 | 64.47 | 67,388 | 宽城区 | 39 |
| 66.09 | 21.59 | 47,063 | 净月区 | 28 |
| 71.98 | 40.17 | 83,304 | 二道区 | 39 |
| 46.77 | 42.71 | 63,583 | 绿园区 | 43 |
| 83.74 | 13.56 | 6,181 | 净月区 | 12 |
| 34.32 | 50.89 | 37,587 | 绿园区 | 14 |
| 59.24 | 42.08 | 83,929 | 南关区 | 45 |
| 41.56 | 17.13 | 28,572 | 长春新区(高新) | 12 |
| 53.17 | 51.11 | 21,808 | 宽城区 | 16 |
| 63.45 | 50.62 | 57,984 | 南关区 | 20 |
| 27.19 | 35.67 | 27,428 | 汽开区 | 14 |
| 67.12 | 40.52 | 63,159 | 二道区 | 40 |
| 42.43 | 44.90 | 42,321 | 绿园区 | 39 |
| 72.94 | 16.94 | 29,635 | 净月区 | 12 |
| 48.04 | 14.05 | 23,557 | 南关区 | 9 |
| 75.86 | 42.25 | 26,665 | 经开区 | 17 |
| 55.42 | 69.32 | 34,788 | 宽城区 | 10 |
| 31.85 | 8.90 | 49,701 | 长春新区(高新) | 9 |
| 53.78 | 26.70 | 54,817 | 南关区 | 23 |
| 46.04 | 27.28 | 25,197 | 长春新区(高新) | 16 |
| 40.53 | 27.46 | 27,491 | 长春新区(高新) | 8 |
| 86.71 | 25.96 | 6,541 | 净月区 | 7 |
| 28.58 | 21.22 | 33,329 | 长春新区(高新) | 8 |
| 69.94 | 25.84 | 27,885 | 净月区 | 18 |
| 73.45 | 33.97 | 25,155 | 经开区 | 19 |
| 46.22 | 56.21 | 34,714 | 绿园区 | 15 |
| 47.57 | 37.09 | 72,478 | 朝阳区 | 30 |
| 37.98 | 46.86 | 50,172 | 绿园区 | 23 |
| 51.72 | 42.35 | 38,173 | 朝阳区 | 15 |
| 84.78 | 5.42 | 15,268 | 净月区 | 11 |
| 90.82 | 19.58 | 3,186 | 净月区 | 5 |
| 80.22 | 46.83 | 34,999 | 经开区 | 9 |
| 68.98 | 10.30 | 19,757 | 净月区 | 6 |
| 44.56 | 53.24 | 26,384 | 绿园区 | 18 |
| 61.70 | 57.32 | 30,536 | 宽城区 | 22 |
| 60.46 | 21.11 | 21,202 | 南关区 | 11 |
| 55.46 | 46.88 | 63,015 | 朝阳区 | 18 |
| 22.53 | 34.99 | 27,411 | 汽开区 | 11 |
| 62.19 | 36.81 | 31,441 | 南关区 | 14 |
| 38.53 | 21.92 | 45,335 | 长春新区(高新) | 11 |
| 72.92 | 56.09 | 24,659 | 二道区 | 6 |
| 59.03 | 64.09 | 43,740 | 宽城区 | 16 |
| 40.76 | 40.76 | 18,108 | 汽开区 | 21 |
| 67.26 | 35.87 | 20,371 | 经开区 | 20 |
| 76.34 | 28.88 | 20,745 | 经开区 | 12 |
| 30.96 | 43.32 | 43,150 | 绿园区 | 10 |
| 67.62 | 62.70 | 44,248 | 二道区 | 11 |
| 52.11 | 56.65 | 28,570 | 绿园区 | 14 |
| 71.10 | 46.36 | 64,019 | 二道区 | 25 |
| 46.27 | 61.36 | 13,773 | 宽城区 | 10 |
| 23.68 | 26.99 | 27,122 | 汽开区 | 11 |
| 52.85 | 15.69 | 23,307 | 南关区 | 15 |
| 48.99 | 47.82 | 93,192 | 绿园区 | 42 |
| 33.76 | 14.86 | 19,773 | 长春新区(高新) | 6 |
| 79.90 | 10.31 | 6,300 | 净月区 | 9 |
| 32.05 | 34.66 | 31,142 | 汽开区 | 20 |
| 68.20 | 52.67 | 42,699 | 二道区 | 14 |
| 35.78 | 26.43 | 40,116 | 长春新区(高新) | 9 |
| 43.52 | 35.38 | 50,881 | 朝阳区 | 30 |
| 85.52 | 18.26 | 15,670 | 净月区 | 16 |
| 48.52 | 31.51 | 64,496 | 朝阳区 | 12 |
| 38.59 | 53.44 | 28,037 | 绿园区 | 10 |
| 58.33 | 50.82 | 29,673 | 宽城区 | 13 |
| 61.35 | 46.26 | 92,210 | 南关区 | 26 |

# 问题三的分析与求解

# 问题四的分析与求解

## 分析与求解

根据题目要求，我们需要计算配送点完成配送所需要的最短时间。首先，为了清楚的观察投送点和小区位置和小区人数的关系，我们将投放点和小区位置和小区人数进行可视化。我们假设一辆大型运输车能满足所有小区的需求量，且车辆是匀速行驶的，因此运输车行驶的路程最短，完成送货的时间最短，所以问题转变为了从配送点出发找一条遍历所有小区并且最终要回到配送点的最短路径。只要最短路径确定下来，那么所需要的最短时间也就确定了。

## 模型评价与改进

## 模型的优点

## 模型的缺点

## 模型的改进

参考文献

附录A 程序代码

# 导入需要的包

import networkx as nx

import matplotlib.pyplot as plt

import pandas as pd

import numpy as np

import matplotlib as mpl

# 贝叶斯优化

from bayes\_opt import BayesianOptimization

# 设置matplotlib的字体

mpl.rcParams['font.sans-serif'] = ['KaiTi']

mpl.rcParams['font.serif'] = ['KaiTi']

# 第三张表，包含每个区的小区及小区的虚拟坐标

data\_3 = pd.read\_excel('./附件3：长春市9个区交通网络数据和主要小区相关数据.xlsx', header=0, sheet\_name=2, usecols=[4, 5, 7])

# 小区横坐标，小区纵坐标

village\_x, village\_y = list(data\_3['小区横坐标'].values), list(data\_3['小区纵坐标'].values)

village\_pos\_1 = np.array([list(pos) for pos in zip(village\_x, village\_y)])

# 导入sklearn中的K-means模型

from sklearn.cluster import KMeans

# 导入平均轮廓函数

from sklearn.metrics import silhouette\_score

# 定义目标函数

def rf\_cv(n\_clusters, n\_init, max\_iter):

model = KMeans(n\_clusters=int(n\_clusters), n\_init=int(n\_init), max\_iter=int(max\_iter))

model.fit(village\_pos\_1)

label = model.labels\_

val = silhouette\_score(village\_pos\_1, label)

return val

# 定义贝叶斯优化的超参数范围

pbounds = {

'n\_clusters':(70, 140),

'n\_init':(10, 50),

'max\_iter':(300, 1000)

}

# 创建贝叶斯优化器

optimizer = BayesianOptimization(

f=rf\_cv, # 黑盒目标函数

pbounds=pbounds, # 取值空间

verbose=2, # verbose = 2 时打印全部，verbose = 1 时打印运行中发现的最大值，verbose = 0 将什么都不打印

random\_state=1,

)

optimizer.maximize( # 运行

init\_points=5, # 随机搜索的步数

n\_iter=1000, # 执行贝叶斯优化迭代次数

)

print(optimizer.max) #输出K-means最优超参数