基于双层规划的鲜奶配送站点的最优化设置

摘要

在现代社会，生鲜配送与人们的生活紧密相连，它不仅仅是商品的简单流通，更是满足消费者日常需求的重要环节。其中，鲜奶作为生鲜产品中的一大类，其配送的时效性和新鲜度更是备受关注。然而，在物流的“最后一公里”环节中，鲜奶配送往往面临着诸多挑战，如何合理设置鲜奶配送站点的位置和数量，成为了牛奶公司必须面对并解决的实际难题。

对于问题一, 本文通过量化配送距离、车辆数和建站数，运用**线性加权法**进行了**无量纲处理**。利用 **K-means 聚类算法**和**肘部法则**确定建站数量，对于聚类所得每一类，采用**双层规划方法**构建模型，得出每一簇中距离之和最短且成本较低的网点，将其作为配送站设置点。总而得到最优的鲜奶配送站设置方案。

对于问题二, 本文针对题述要求，在**最长 10 分钟配送时间**的限制下，寻求最小配送站建设成本的最优方案。本文先将时间限制转化为实际距离限制，将实际距离限制转变为**二维坐标图中**的距离限制。从而引入网点在配送站点最大配送距离内的限制条件，对双层规划模型中的上层规划进行改进。利用 MATLAB，计算得出了在设定的配送时间限制下，建设成本最低的配送站点建设方案。

对于问题三, 鉴于题述中对于配送站点方案的需求，即均衡各鲜奶配送站的配送量，本文首先计算了所有配送网点的总订奶量，并进一步在确定了最优聚类数 K 的条件下，计算得出平均订奶量。随后，基于这一平均订奶量，设定一个配送量的**均衡区间 [a,b]**。对聚类结果进行判断，对于不均衡的情况，更改簇边缘网点聚类位置，最终使所有配送站各自的配送量 p_h 均落在此**均衡区间 [a,b]** 内，即每个配送站的配送量已达到**均衡状态**，这样既避免了资源的过度浪费，也确保了配送的及时性。而后改进问题二所建立的双层规划模型，在下层规划中，额外加入了鲜奶的配送成本作为考量因素，得到选定鲜奶配送站下鲜奶配送的最优路径。最终，通过**新的双层规划模型**得到新的最优鲜奶配送站的设置方案。

关键字：二维图　K-means 聚类　双层规划　线性加权法　不同约束条件下最优配送站点设置

一、 问题重述

1.1 问题背景

生鲜配送作为与民众日常生活紧密相连的重要环节，”最后一公里”的物流瓶颈问题尤为凸显。特别是对于鲜奶配送而言，时效性是其核心要素。鲜奶配送站每日清晨，必须运用非冷链小型物流车辆，确保当日鲜奶在第一时间准确、高效地送达至片区内的各个分销网点。因此，如何科学、合理地规划配送站点的布局与数量，已然成为牛奶公司亟待解决的实际问题。

1.2 问题要求

问题 1 在配送时间尽量缩短的情况下，得到最经济的配送站点设置方案。

问题 2 在确保配送时效不超过 10 分钟的条件下，得出最小化建设成本的鲜奶配送站的最优设置方案。

问题 3 在配送量均衡的条件下，得出最小化配送成本和较高配送时效与效率的鲜奶配送站的最优设置方案。

二、 问题分析

2.1 问题一分析

对于问题一, 需要在配送效率与配送时效较高条件下，最小化配送成本以得到鲜奶配送站的最优设置方案。

首先，对问题一进行合理简化，将各个抽象概念转化为具体可量化的指标。如用网点间的距离指代配送速度，用配送车辆数指代配送成本，用配送中心建站数指代建设成本。从而能够衡量各因素对配送站点设置方案的影响。

同时，对经济成本与配送距离对方案设置的影响权重进行深入分析，确定两者各自的占比，再进行无量纲处理。随后，利用 K-means 聚类算法，根据网点分布的密集程度进行聚类，根据肘部法则以确定合适的建站数量。完成聚类后，计算同一分区中各个网点到其余网点的距离之和，将每个分区中距离之和最短的网点作为配送站点。最后，运用双层规划方法，制定出在配送速度和配送成本、建设成本上均达到优化目标的鲜奶配送站设置方案。

2.2 问题二分析

对于问题二需要在配送时间为 10 分钟以内的条件下，考虑最小的配送站建设成本，得到最优的配送站点设置方案。其等同于在问题一基础上增加一个限制：配送站点的配送最大配送距离为 S，即对双层规划中上层规划的目标函数增加了额外的约束条件，其为网点在配送站的配送范围内，其公式表达式为：$D_{ij} \leq d, i \in I, j \in J$。由运动学公式 S=vt，求出配送站的配送最大配送距离为 S，依照附录所给图上距离与实际距离的比例尺求出 d 的大小。即可完成对问题一所建模型的改进与优化，从而得到解决问题二的数学模型。依照所建立模型利用 Matlab 计算得到在题设的配送时间限制下，建设成本最低的配送站点建设方案。

2.3 问题三分析

对于问题三，题述要求需要保证配送站点方案中各鲜奶配送站的配送量均衡。因此，本文首先计算了所有网点的总订奶量，以及最优聚类数 K 下的平均订奶量，并依据平均订奶量设置配送量的均衡区间 [a,b]。若所有配送站各自的配送量 p_h 在此均衡区间 [a,b] 内，则可判定每个配送站的配送量均衡，既不造成资源浪费，也不延误配送。根据上述分析，本文基于问题二所建立数学模型的基础上，在进行双层规划寻找最优配送站设置方案前，建立了一个新的目标函数：$a \leq p_h \leq b$，并规划其约束条件，使得每一簇中设置的配送站配送量均衡，同时本文优化问题二的双层规划模型，于下层规划中额外考虑鲜奶的配送成本以得到最佳配送路径。从而得到配送量均衡的新鲜奶配送站的设置方案。

三、 模型假设

- 题述网点中，两两网点间或配送站与网点间的道路均为直线。
- 每个鲜奶配送站都有足够的非冷链小型物流车，且配送站的规模容量可以满足网点订奶量。
- 一个网点有且仅由一个鲜奶配送站供应。
- 每辆非冷链小型物流车的时速恒定不变，不受道路车流量影响。

四、 符号说明

表 **1** 符号说明 **1**

集合	说明
I	该市网点集合，I ={1,2，. . . ，92}
J	候选配送站集合,J ={1,2, . . . ，92}
N	配送车辆集合 N={1,2,...,n}
I_k	聚类后每一簇中的网点集合 I_k={1,2,3,...,f}

表 **2** 符号说明 **2**

变量	取值范围	说明
h_j	{0,1}	判断是否选择 j 点建设配送站。
z_{ij}	{0,1}	判断是否选择配送站 j 作为网点 i 的配送点
y_{ijn}	{0,1}	判断车辆 n 是否从配送站 j 出发为网点 i 配送鲜奶
ω_i	$\{Z_+\}$	网点 i 的订奶量
r_{ij}	$\{Z_+\}$	配送站 j 配送鲜奶到网点 i 的单位成本
D_{ij}	$\{Z_+\}$	配送站 j 到网点 i 的距离
C_{nj}	$\{R_+\}$	配送站 j 派遣车辆 n 的单位时间成本
C_n	$\{R_+\}$	车辆 n 的自身成本
C_j	$\{R_+\}$	单个配送站的建设成本
T	$\{R_+\}$	单个车辆从网点 i 到配送点 j 的行驶时间

五、 问题一模型的建立和求解

5.1 模型建立

5.1.1 数据可视化

根据所给附录数据，绘制题述片区中网点及网点间道路的二维坐标图。

5.1.2 K-means 聚类确定配送站建站个数

Step1: K-means 聚类对于题述片区中网点及网点间道路的二维坐标图，需要确定配送站的设置个数。针对这一问题，本文采用 K-means 聚类。

K-means 聚类是随机从数据集中选取 K 个点作为初始聚类中心，然后计算各个样本到聚类中心的距离，通过距离大小把样本归到离它最近的聚类中心所在的类。计算新形成的每个聚类的数据对象的平均值来得到新的聚类中心，调整到相邻两次的聚类中心没有任何变化为止。

Step2: 肘部法判断选取最佳聚类簇 K

对于最佳聚类簇 K 的选取方法，常用肘部法判断。肘部法利用观察 K 与 SSE 的变化获得，SSE 随着 K 的增大而减小，并且在达到第一个临界转折点处，SSE 减小速度变缓，这个临界转折点就可以考虑为最佳聚类性能的点，该点对应的 K 就是最佳聚类数。

Step3: 判断聚类结果是否收敛

同时，在 K-means 算法中，每一轮迭代完成后，都需要判断聚类结果是否收敛。为此，在此定义一个目标函数，即差平方和函数（Sum of the Squared Error）, 定义如下：

$$SSE = \sum_{i=1}^{k} \sum_{x \in C_i} d(x, c_i)^2$$

其中，x 是集合 D 的对象，C_i 代表第 i 个簇，c_i 是 C_i 的中心，$C_i = \frac{1}{m_i} \sum_{x \in C_i} x, m_i$ 是中心数据对象的个数。K-means 算法迭代执行过程中，SSE 的值会不断减小。当前后两轮迭代所得到的 SSE 保持不变或者二者之间的差异小于某个预设的门限值 ε，就可以认为聚类结果已收敛。

5.1.3 采用双层规划寻找最优鲜奶配送站的设置方案

在确定鲜奶配送站的建站个数后，仍需要考虑 K-means 聚类后每一簇中鲜奶配送站的建站位置。从而得到最优的鲜奶配送站设置方案。

针对这一问题，本文采用双层规划区寻找最优的鲜奶配送站的设置方案，双层规划在决策过程中，同时考虑两个不同层次或两个不同方面的规划或策略，从而很好地考虑到上层决策者和下层执行者之间的关系，进行规划。在此，对于聚类后的每一簇，定义 I 为网点集合，$I = \{0, 1, 2, ..., 92\}$, J 为候选配送站集合, 有 $J = \{1, 2, ..., 92\}$, N 为配送车辆集合,$N = \{1, 2, ..., n\}$

为了寻找最优鲜奶配送站设置方案，设计双层规划流程图如图1所示：

Step1: 建立上层规划模型

上层规划模型：用于确定配送中心最优选址。题述中，鲜奶配送站应做到在配送在途时间尽可能短的同时最经济地配送鲜奶。

据此可以得到目标函数应由鲜奶配送站的总建设成本、配送点与各网点间的距离之和，以及鲜奶配送成本三者之和组成，并且应取其最小值，则目标函数为：

$$\min F_u = \sum_{j \in J} C_j h_j + \sum_{i \in I} \sum_{j \in J} D_{ij} z_{ij} \tag{1}$$

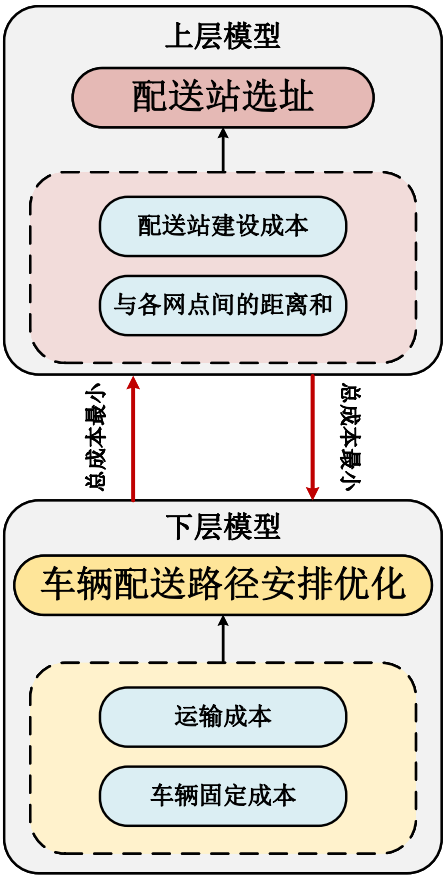

图 **1**　双层规划流程图

式(1)中，其中 C_j 为一个配送站的建设成本，$h_j = \begin{cases} 0, & j\text{点非配送点} \\ & \\ 1, & \text{选择 j 点为配送点} \end{cases}$，$D_{ij} \in Z^+$，$z_{ij} = \begin{cases} 0, & \text{网点 i 非配送站 j 配送} \\ & \\ 1, & \text{网点 i 有配送站 j 配送} \end{cases}$，$\omega_i$ 表示网点 i 的订奶量，r_{ij} 表示网点 i 与配送站 j 之间配送鲜奶的单位成本。

式(1)中，第一项 $\sum_{j\in J} C_j h_j$ 表示总建设成本，第二项 $\sum_{i\in I}\sum_{j\in J} D_{ij} z_{ij}$ 表示配送点与各网点间的距离。

目标函数的约束条件为：

$$s.t.\begin{cases} \sum_{j\in J} z_{ij}=1, & i\in I \\ z_{ij}\le h_j, & i\in I \quad j\in J \\ \sum_{j\in J} h_j = K & \\ z_{ij},h_j\in\{0,1\}, & i\in I \quad j\in J \end{cases} \tag{2}$$

其中，$\sum_{j\in J} z_{ij}=1$ 表示每个网点有且仅有一个配送站服务；$z_{ij}\le h_j$ 表示只有建设的配送站才可以提供服务；$\sum_{j\in J} h_j = K$ 表示建设的配送站数量为 K；$z_{ij}\in h_j$ 表示条件满足的状态。K 为 K-means 聚类的最佳聚类数。

Step2: 建立下层规划模型

下层规划模型：用于确定鲜奶最佳配送路径，即行驶最短行程、使用最少时间、花费最低成本完成鲜奶配送任务。

据此可以得到目标函数应由车辆的配送成本和车辆的固定使用成本之和组成，并且应取其最小值，则目标函数为：

$$\min F_2=\sum_{j\in J} TC_{nj}h_j+\sum_{j\in J} nC_n h_j \tag{3}$$

式中，$T=\dfrac{D_{ij}}{V}$ 表示单个车辆从网点 i 到配送点 j 的行驶时间，C_{nj} 表示配送站 j 所派的配送车辆 n 单位时间内所消耗的成本，n 表示配送站 j 使用的总非冷链小型物流车车辆数，C_n 表示车辆 n 的固定使用成本。$\sum_{j\in J} TC_{nj}h_j$ 表示车辆的配送成本，$\sum_{j\in J} nC_n h_j$ 表示车辆的固定使用成本。

其约束条件为：

$$s.t.\begin{cases} \sum_{i\in I}\sum_{n\in N} y_{jin}-h_j\le 0, & i\in I \quad j\in J \\ \sum_{n\in N} y_{ijn}\le 1, & i\in I \quad j\in J \\ \sum_{n\in N} w_n = N-1 & \end{cases} \tag{4}$$

式中，$y_{ijn}=\begin{cases} 1, & \text{车辆 n 从配送站 j 出发为网点 i 配送鲜奶} \\ 0, & \text{车辆 n 不从配送站 j 出发} \end{cases}$。

$\sum_{i\in I}\sum_{n\in N} y_{jin} - h_j \geq 0$ 表示非冷链小型运输车均是从鲜奶配送站派出的，$\sum_{i\in I}\sum_{n\in N} y_{jin} - h_j \leq 0$ 保证所有网点均有车辆配送货物；$\sum_{n\in N} w_n = N - 1$ 保证每个网点只能出现在一台配送车辆的配送路径上；$\sum_{i\in w_n} q_{ni} \leq Q_n$ 保证每辆配送的鲜奶总量都小于该车的最大容量。

在双层规划进行求解前，仍需考虑上下层各自目标函数的不可公度性，对目标函数进行预处理。

5.1.4 采用线性加权法对目标函数进行无量纲化处理

本文采用线性加权法对目标函数进行无量纲化处理。线性加权法是根据目标的重要性确定一个权, 以目标函数的加权平均值为评价函数, 使其达到最优。线性加权法关键是要确定每个目标的权重, 它反映不同目标在决策者心中的重要程度，重要程度高的权重就大，重要程度低的权重就小。相较于 ε 约束法、理想点法等方法更为简便、快捷。

该方法的基本步骤如下：

Step 1: 确定每个目标的权系数

$$\sum_{j=1}^{m} w_j = 1.$$
$$0 \leq w_j \leq 1,$$
$$j = 1, 2, 3, ..., m.$$

式中，w_j 为各目标的权系数。

Step 2: 写出评价函数

$$\sum_{j=1}^{m} w_j f_j(x)$$

Step 3: 求评价函数最优值

$$\min \sum_{j=1}^{m} w_j f_j(x),$$
$$s.t. x \in \Omega$$

上述步骤中，m 表示目标函数中子目标的个数，f_j 指代子目标函数，w_j 表示子目标函数的权系数，x 表示决策变量，Ω 表示可行域。

本文第一问建立的模型中，上层规划中，建设成本与最短配送距离的数量级差距较小，无需进行归一化处理，但两者的量纲不同，应用线性加权法进行无量纲处理。

本文对于成本与距离的两个子目标的的期望权重均取 0.5，则上层规划的目标函数变换如下：

$$\min F_u = 0.5\sum_{j\in J} C_j h_j + 0.5\sum_{i\in I}\sum_{j\in J} D_{ij} z_{ij}$$

下层规划中数据的量纲一致且数量级相差不大，仍为：

$$\min F_2 = \sum_{j\in J} TC_{nj} h_j + \sum_{j\in J} nC_n h_j$$

5.2 模型求解

5.2.1 数据可视化

Step 1: 导入附录数据，列出各网点及连接道路的二维坐标图，如2所示:

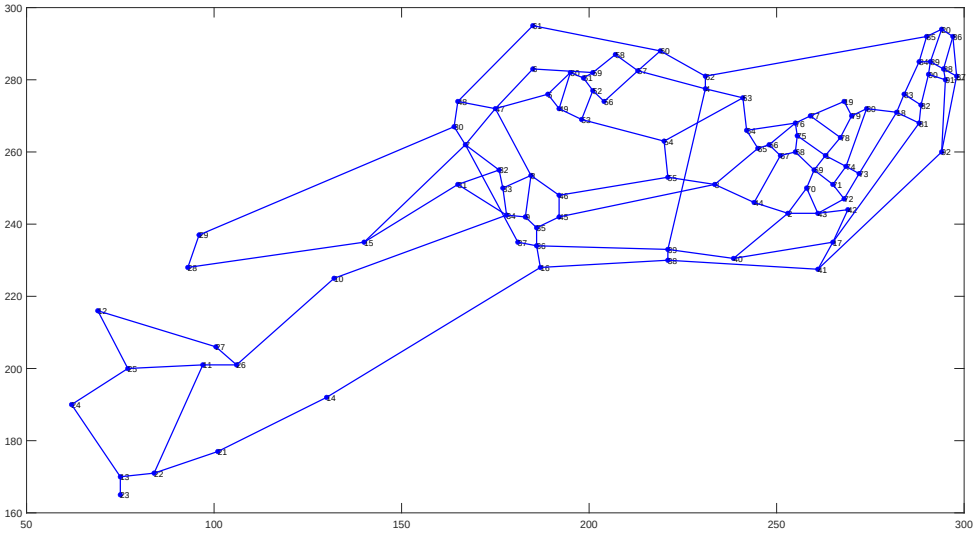

图 2　二维坐标图

Step 2: 基于欧氏距离 $D = \sqrt{(\Delta x)^2 + (\Delta y)^2}$，计算得出各网点间道路的权重（距离）。

使用起始网点和终止网点分别为顶点对组 x，y，再基于权重向量 D 构建赋权无向图，如图3所示：

5.2.2 K-means 聚类模型求解

Step 1: 绘制 SSE 与 K 的线性关系图

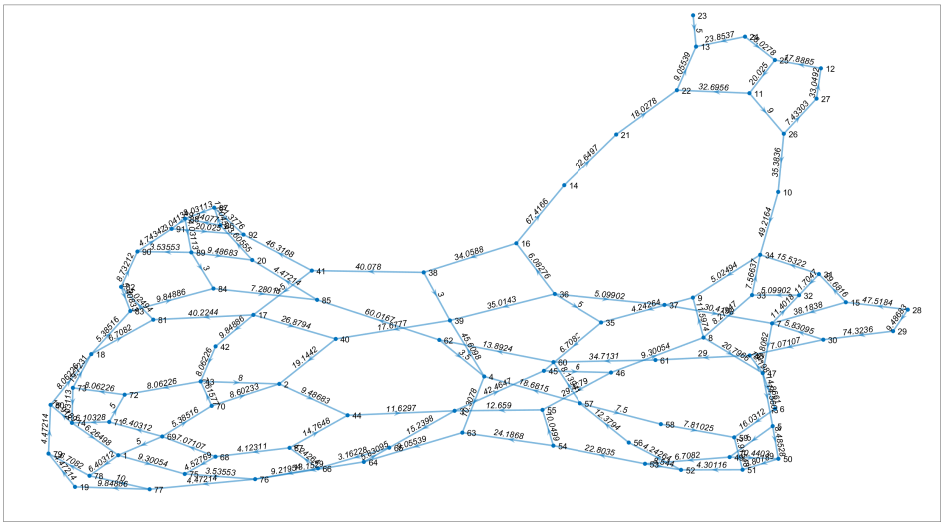

图 **3**　赋权无向图

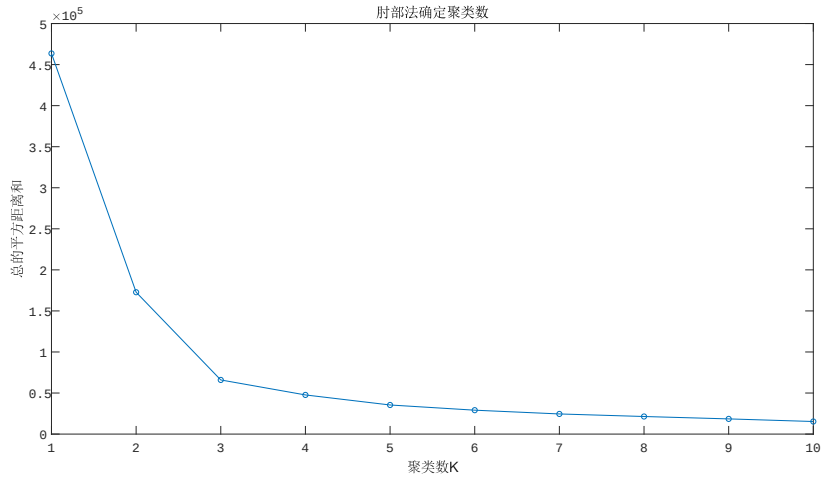

图 **4**　**SSE** 与 **K** 线性关系图

SSE 与 K 的线性关系如图4所示, 得到聚类结果收敛效果最好的聚类数从而确定建站个数。

Step2: 通过肘部法则得到最佳聚类数

由肘部法则易得 K=3 为最佳聚类数，即建站个数为 3，把样本归到离它最近的聚类中心所在的类，得出 K-means 聚类结果图5所示：

Step3: 合理调整不连通网点

在使用 K-means 方法简单聚类后注意到，得到的第三个聚类之间的部分网点并不直接连通，出于符合实际情境以及便于后续解题，手动将不连通的网点归于第二类，得从而到最终的聚类结果，修改后 K-means 聚类结果图如图6所示：

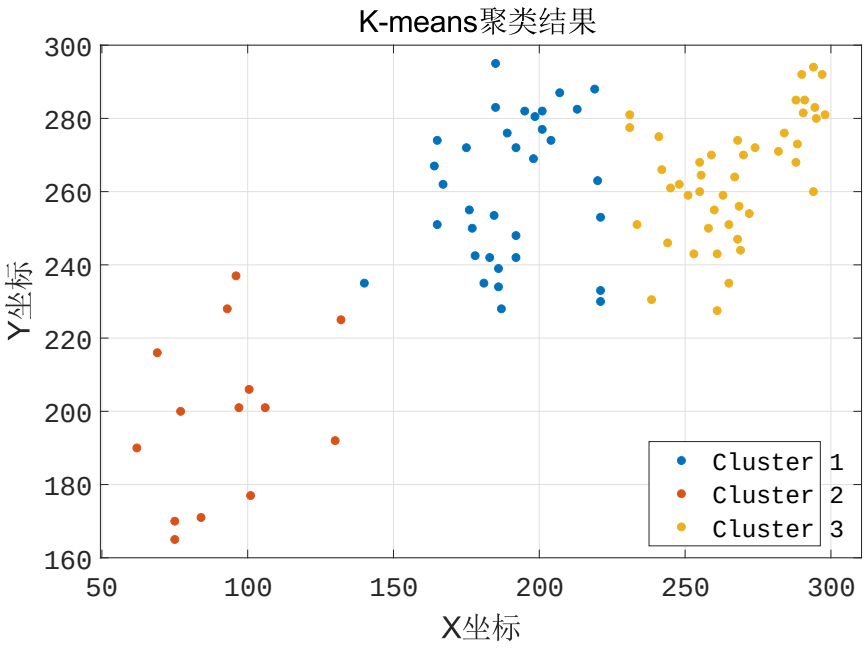

图 5　K-means 聚类结果图

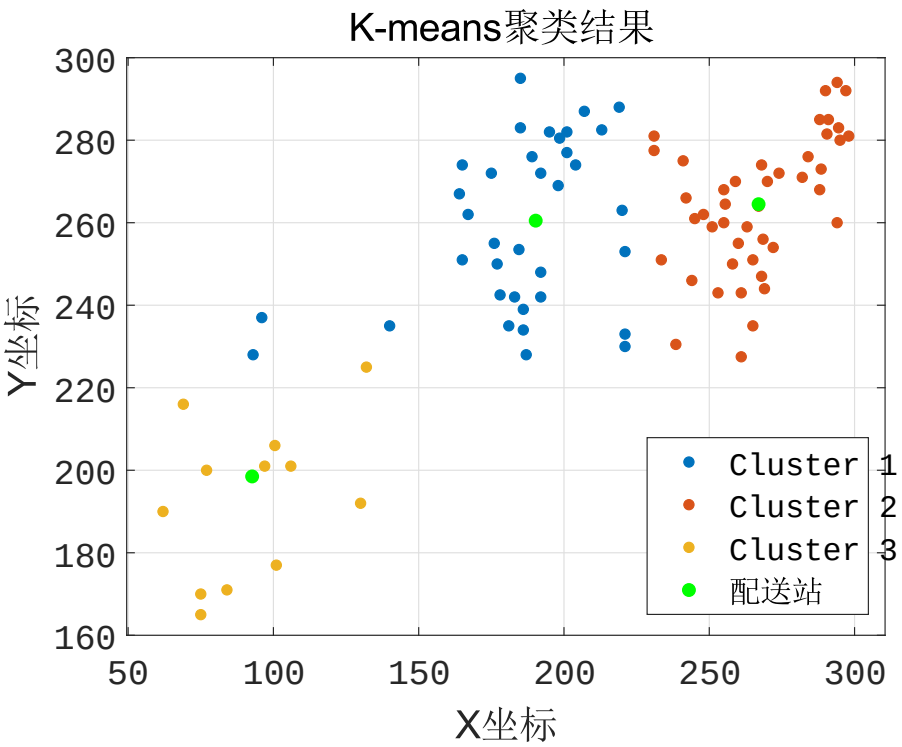

图 6　K-means 聚类结果图 (修改后)

Step4: 对每一簇构建赋权无向图

聚类后每一簇单独由网点对组 x，y 和权重向量 D 构建赋权无向图：

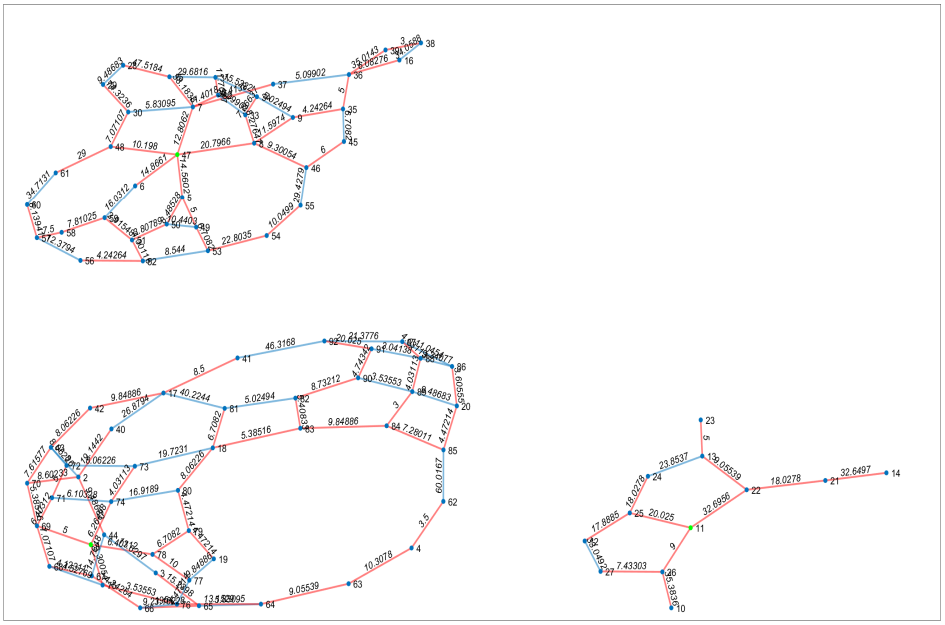

图 7　赋权无向图

5.2.3 求解最优配送站设置点

Step1: 在每一簇中运用双层规划

在上层规划中，目标函数为：

$$\min F_u = 0.5\sum_{j\in J} C_j h_j + 0.5\sum_{i\in I}\sum_{j\in J} D_{ij} z_{ij}$$

约束条件为：

$$s.t.\begin{cases} \sum_{j\in J} z_{ij} = 1, & i \in I \\ z_{ij} \leq h, & i \in I \quad j \in J \\ \sum_{j\in J} h_j = K & \\ z_{ij}, h_j \in \{0,1\}, & i \in I \quad j \in J \end{cases}$$

在下层规划中，目标函数为：

$$\min F_2 = \sum_{j\in J} TC_{nj}h_j + \sum_{j\in J} nC_n h_j$$

约束条件：

$$s.t.\begin{cases} \sum_{i\in I}\sum_{n\in N} y_{jin} - h_j \le 0, & i\in I \quad j\in J \\ \sum_{n\in N} y_{ijn} \le 1, & i\in I \quad j\in J \\ \sum_{n\in N} w_n = N-1 \end{cases}$$

Step2: 绘制出有 3 个配送中心时的最优配送站的设置方案

利用 Matlab，求解得出最优配送站设置方案如图8所示：

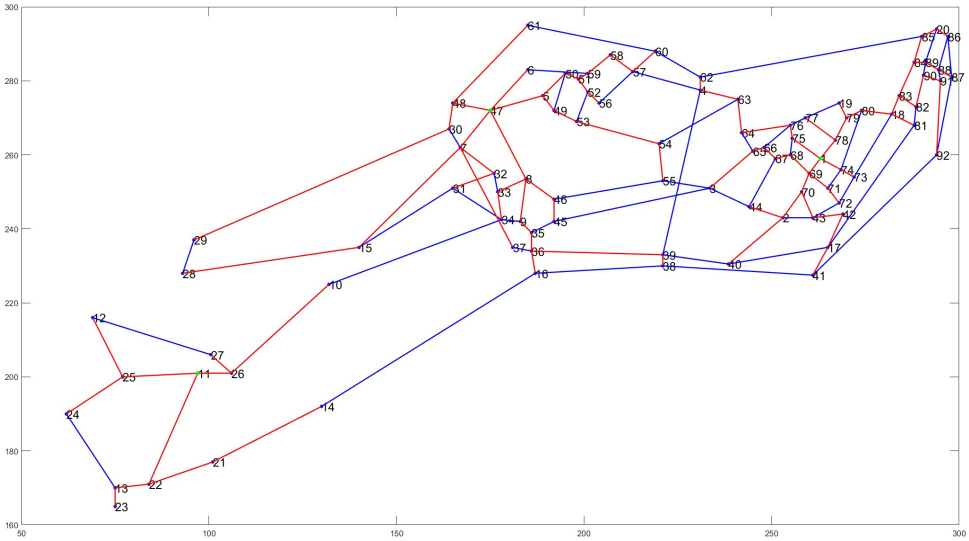

图 **8**　最优配送站设置方案

图中的的蓝色节点为网点，绿色节点为配送站，蓝线为非运送路径，红线为运送路径。

即设置 1、47、11 号网点为鲜奶配送站。同时由 matlab 可计算出此设置方案下的配送路径总长度 S_3=3039.413 m

Step3: 绘制出有 1 个配送中心时的最优配送站的设置方案进行对比

同时，在不进行聚类的情况下，通过双层规划得出了仅设置一个配送站时的最佳设置网点（3 号网点）与配送路径，如图9所示：

得到该设置方案下的配送路径总长度 S_1=6393.765 m。

比较与易得:$S_1 > S_3$，即在设置 1、47、11 三个网点作为鲜奶配送站，可以在尽量缩短配送在途时间的条件下，使配送站设置方案最经济。

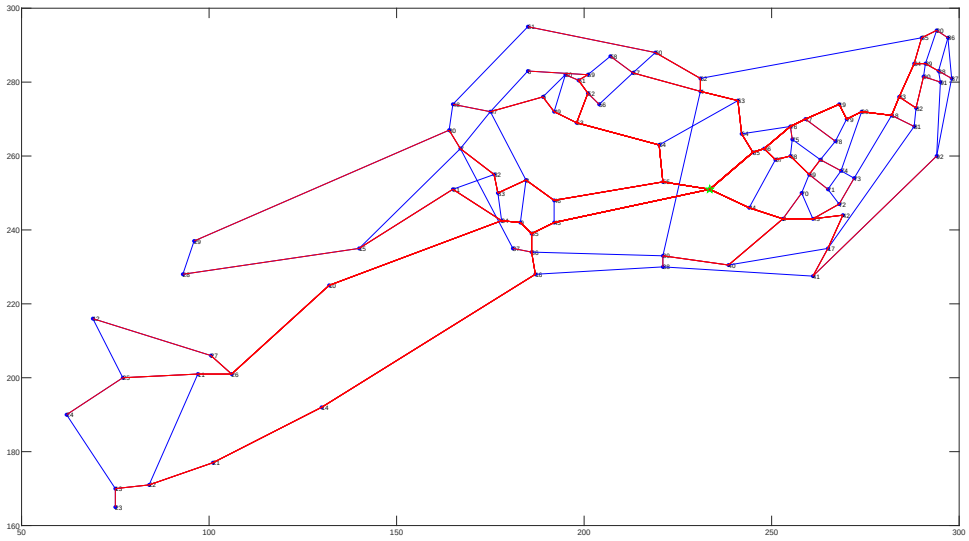

图 9　最佳设置网点（3 号网点）与配送路径

5.3 求解结果

问题一求解结果如下表所示：

表 3　问题一求解结果

设定配送站个数	配送站网点号	总距离	需要物流车数量	总成本
1	3	6393.77	24	较高
3	47,1,11	3039.41	32	较低

对比表中数据，由于本文模型假设每辆非冷链小型物流车的时速恒定不变，不受道路车流量影响，故尽量缩短配送总距离即可满足尽量缩短配送时间的题设条件，故而选择网点 1、11、47 三个网点设置配送站是在尽量缩短配送距离的条件下，总成本较低的最经济的鲜奶配送站设置方案。

六、 问题二的模型的建立和求解

6.1 模型建立

6.1.1 建立同问题一相同的数学模型

本题要求配送站在 10 分钟内将鲜奶配送到网点,同时给出物流车的时速为 $20km/h = \frac{1000}{3}m/min$，因而已知条件有物流车时速为 $20km/h = \frac{1000}{3}m/min$，配送时间阈值为 $t = 10\,\text{min}$，由运动学公式 $s = \frac{v}{t}$ 可得二维坐标图中配送点的最大配送范围为 $s = 3333.33m$。

此处需注意附录中规定坐标的单位长度为 1 毫米且给定比例尺为：图上距离：实际距离 =1mm ： 10m ，故按附录所给比例可得二维坐标图中的配送站的最大辐射距离为 d=333.33 m。

6.1.2 在问题一所建立模型的基础上改进双层规划

对于问题二，上层规划在第一问基础上，增加一个约束条件：$D_{ij} \leq d$ ，表示网点在配送站的配送范围内，d 表示新建配送站配送距离上限。

Step1: 建立上层规划模型

上层规划模型：用于确定配送中心最优选址。题述中，鲜奶配送站应做到在配送在途时间尽可能短的同时最经济地配送鲜奶。

据此可以得到目标函数应由鲜奶配送站的总建设成本、配送点与各网点间的距离之和，以及鲜奶配送成本三者之和组成，并且应取其最小值，则目标函数为：

$$\min F_u = \sum_{j \in J} C_j h_j + \sum_{i \in I} \sum_{j \in J} D_{ij} z_{ij}$$

此目标函数中，建设成本与最短配送距离的数量级差距较小，无需进行归一化处理；但两者的量纲不同，应用线性加权法进行无量纲处理。由于本问中题干限制配送距离，要求考虑建设成本，故本问对于成本与距离两个子目标的期望权重分别取 0.8 与 0.2，则上层规划的目标函数变换如下：

$$\min minF_u = 0.8 \sum_{j \in J} C_j h_j + 0.2 \sum_{i \in I} \sum_{j \in J} D_{ij} z_{ij}$$

目标函数的约束条件为：

$$s.t. \begin{cases} \sum_{j \in J} z_{ij} = 1, & i \in I \\ z_{ij} \leq h, & i \in I \quad j \in J \\ \sum_{j \in J} h_j = K & \\ D_{ij} \leq d, & i \in I \quad j \in J \\ z_{ij}, h_j \in \{0, 1\}, & i \in I \quad j \in J \end{cases}$$

Step2: 建立下层规划模型

下层规划模型：用于确定鲜奶最佳配送路径，即行驶最短行程、使用最少时间、花费最低成本完成鲜奶配送任务。题述鲜奶配送站的任务就是在缩短配送在途时间，限制

配送鲜奶配送站配送范围的条件下，最经济地配送鲜奶。目标函数是由鲜奶配送站的总建设成本、配送点与各网点间的距离两者之和最小，目标函数公式表达如下：

$$\min F_2 = \sum_{j\in j} TC_{nj}h_j + \sum_{j\in j} nC_n h_j + \sum_{j\in j} C_j h_j$$

其约束条件为：

$$s.t. \begin{cases} \sum_{i\in I}\sum_{n\in N} y_{jin} - h_j \le 0, & i \in I \quad j \in J \\ \sum_{n\in N} y_{ijn} \le 1, & i \in I \quad j \in J \\ \sum_{n\in N} w_n = N - 1 \end{cases}$$

6.2 模型求解

对于问题一所选择配送站配送方案判中，设每一个鲜奶配送点的最大配送距离，通过比较与 d(二维坐标图中配送点的最大配送距离) 两者的大小，判断问题一中的配送站设置方案是否符合问题二所要求的配送辐射范围。

利用 Matlab，可得出 $M_1 = 69.93 < d$，$M_{11} = 83.37 < d$，$M_{47} = 72.72 < d$ 故问题一所得鲜奶配送站的设置方案依旧符合问题二的题设条件。

6.3 求解结果

问题二求解结果如下表所示：

表 4　问题二求解结果

配送站网点号	最远单配送路径	配送范围阈值	是否符合题设要求
47	72.73		符合
1	83.37	333.33	符合
11	69.93		符合

对图表数据分析可知，问题一所得鲜奶配送站的设置方案依旧符合问题二对鲜奶配送站最远配送距离的限制。即最优的配送站点设置方案为：设置网点 1、11、47 为鲜奶配送站点。

七、 问题三的模型的建立与求解

7.1 模型建立

对于问题三，需在问题二所建双层规划模型基础上额外考虑配送量对建站点的影响，改进双层规划流程图如下：

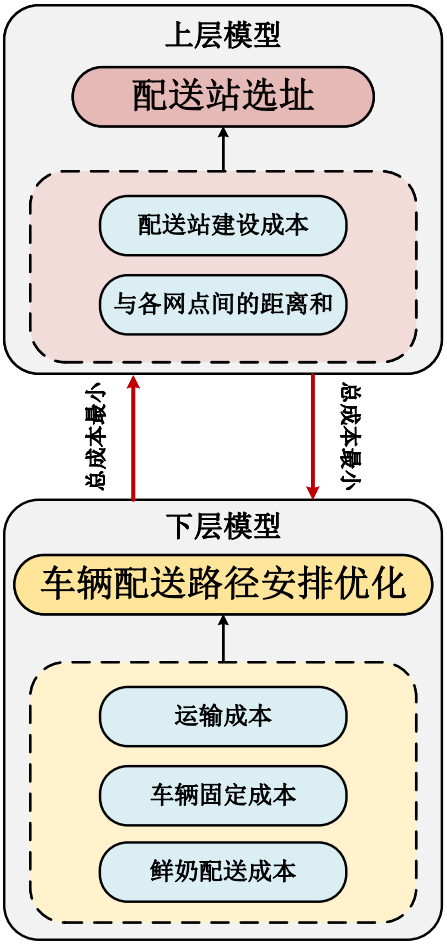

图 **10**　问题三双层规划流程图

Step1: 建立上层规划模型

上层规划模型：用于确定配送中心最优选址。题述鲜奶配送站的任务就是在缩短配送在途时间，限制配送鲜奶配送站配送范围的条件下，最经济地配送鲜奶。

目标函数是鲜奶配送站的总建设成本、配送点与各网点间的距离两者之和最小，目标函数表达式为：

$$minF_u = \sum_{j \in J} C_j h_j + \sum_{i \in I} \sum_{j \in J} D_{ij} z_{ij}$$

由于此目标函数中，建设成本与最短配送距离的数量级差距较小，无需进行归一化处理；但两者的量纲不同，应用线性加权法进行无量纲处理。由于本问中要求配送量均衡，从而避免造成资源浪费或延误配送，而建设成本的要求较松散，故本问对于成本与距离两个子目标的期望权重分别取 0.3 与 0.7，则上层规划的目标函数变换如下：

$$minF_u = 0.3\sum_{j\in J} C_j h_j + 0.7\sum_{i\in I}\sum_{j\in J} D_{ij} z_{ij}$$

其约束条件为：

$$s.t.\begin{cases} \sum_{j\in J} z_{ij} = 1, & i \in I \\ z_{ij} \le h, & i \in I \quad j \in J \\ \sum_{j\in J} h_j = K & \\ D_{ij} \le d, & i \in I \quad j \in J \\ z_{ij}, h_j \in \{0,1\}, & i \in I \quad j \in J \end{cases}$$

Step2: 建立下层规划模型

下层规划模型：用于确定鲜奶最佳配送路径，即行驶最短行程、使用最少时间、花费最低成本完成鲜奶配送任务。

据此可以得到目标函数应由车辆的配送成本和车辆的固定使用成本之和组成，并且应取其最小值，三者量纲相同，数量级差距较小，无需对目标函数进行无量纲化处理，则目标函数为：

$$minF_2 = \sum_{j\in J} TC_{nj} h_j + \sum_{j\in J} nC_n h_j + \sum_{i\in I}\sum_{j\in J} \omega_i r_{ij} D_{ij}$$

式中，$\sum_{i\in I}\sum_{j\in J} {}_i r_{ij} D_{ij}$ 表示鲜奶配送成本。r_{ij} 表示网点 i 与配送站 j 之间配送鲜奶的单位成本，ω_i 表示网点 i 的订奶量。

其约束条件为：

$$s.t.\begin{cases} \sum_{i\in I}\sum_{n\in N} y_{jin} - h_j \le 0, & i \in I \quad j \in J \\ \sum_{n\in N} y_{ijn} \le 1, & i \in I \quad j \in J \\ \sum_{n\in N} w_n = N - 1 & \end{cases}$$

7.2 模型求解

7.2.1 依据问题一二聚类结果进行均衡鲜奶配送量的修订

若要均衡配送站的配送量，需要对网点进行新的规划，从而保证每一配送站点的配送量在均衡区间 $[0.9\bar{x}, 1.1\bar{x}]$ 内，目标函数表达式为：

$$0.9\bar{x} \leq p_h \leq 1.1\bar{x}$$

其中 h 表示第 h 个配站点，且 $h \in \{1, 2, ..., K\}$，K 为最优聚类数。

定义 I_k 为聚类后每一簇的网点集合，即 $I_k = \{1, 2, 3, ..., f\}$，$_i$ 表示网点 i 的订奶量。

目标函数的约束条件为：

$$s.t. \begin{cases} \bar{x} = \frac{1}{K}\sum_{i=1}^{92}\omega_i, \\ h \in \{1, 2, ..., K\}, \\ K = 3, \\ p_h = \sum_{k=1}^{K}\sum_{i \in I_k}\omega_i \\ \sum_{k=1}^{K} I_k = 92, \end{cases}$$

网点重新分区规划后，单独由每一簇包含的所有网点对组 $x \square y$ 和权重向量 D 构建出三个赋权无向图如下：

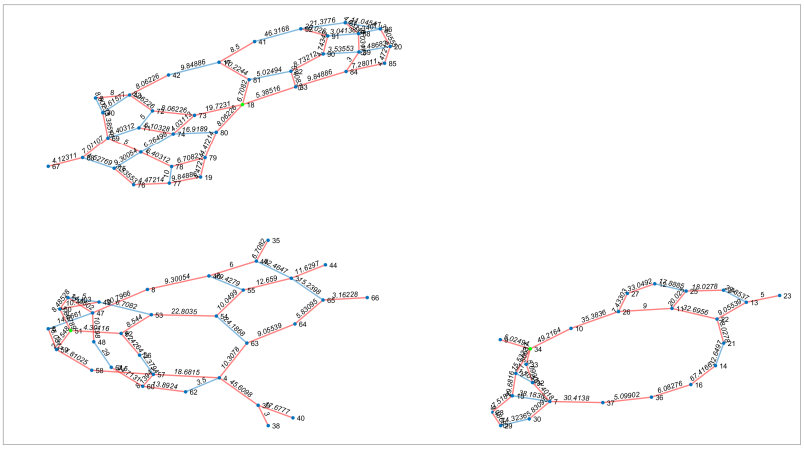

图 **11**　均衡配送量后的的赋权无向图

7.2.2 运用双层规划求解最优配送站的设置点

上层规划的目标函数是鲜奶配送站的总建设成本、配送点与各网点间的距离两者之和最小，目标函数的表达式为：

$$minF_u = 0.3\sum_{j\in J} C_j h_j + 0.7\sum_{i\in I}\sum_{j\in J} D_{ij} z_{ij}$$

其约束条件为：

$$s.t.\begin{cases} \sum_{j\in J} z_{ij} = 1, & i \in I \\ z_{ij} \le h, & i \in I \quad j \in J \\ \sum_{j\in J} h_j = K & \\ D_{ij} \le d, & i \in I \quad j \in J \\ z_{ij}, h_j \in \{0,1\}, & i \in I \quad j \in J \end{cases}$$

下层规划的目标函数是车辆的配送成本、车辆的固定使用成本与鲜奶配送成本三者之和的最小值，目标函数表达式为：

$$minF_2 = \sum_{j\in J} TC_{nj} h_j + \sum_{j\in J} nC_n h_j + \sum_{i\in I}\sum_{j\in J} \omega_i r_{ij} D_{ij}$$

式中，$\sum_{i\in I}\sum_{j\in J} {}_i r_{ij} D_{ij}$ 表示鲜奶配送成本。r_{ij} 表示网点 i 与配送站 j 之间配送鲜奶的单位成本，ω_i 表示网点 i 的订奶量。

其约束条件为：

$$s.t.\begin{cases} \sum_{i\in I}\sum_{n\in N} y_{jin} - h_j \le 0, & i \in I \quad j \in J \\ \sum_{n\in N} y_{ijn} \le 1, & i \in I \quad j \in J \\ \sum_{n\in N} w_n = N - 1 & \end{cases}$$

利用 Matlab 求解，得到 J=51、18、34。可绘制出最优配送站设置方案如图12所示：

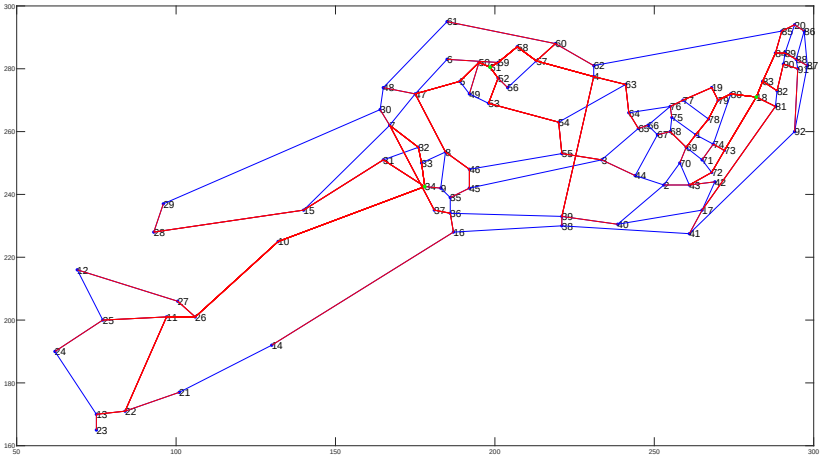

图 **12**　最优配送站设置方案

此时设立 51 号网点为配送点的总配送量 p_1=3980，设立 18 号网点为配送点的总配送量 p_2=4210，设立 34 号网点为配送点的配送量 p_3=4260。

7.3 求解结果

问题三求解结果如下表所示：

表 **5**　问题三求解结果

配送站网点号	配送量	均衡区间	配送量是否均衡
51	3980		均衡
18	4210	[3735,4565]	均衡
34	4260		均衡

对表中数据分析可知，在均衡各配送站的配送量的条件下，重新规划网点 18、34、54 为鲜奶配送站的设置方案，可以既不造成资源浪费，又不延误配送。

八、 模型评价

8.1 模型的优点

- 二维坐标图：二维坐标图直观易懂，能够有效地可视化数据，帮助人们更好地理解数据之间的关系和规律。
- K-means 聚类：是一种简单而有效的聚类算法，易于实现且计算速度快，适用于大规模数据集。数据点划分为 K 个簇，具有直观性和可解释性。

- 双层规划模型：能够在含有多个决策者或多个层次的决策问题中进行优化，在复杂系统中具有较好的适用性和解释性，可优化性高。

8.2 模型的缺点

- 二维坐标图：在高维数据集上失去了效果，无法很好地展示多维数据的复杂性和关联性。
- K-means 聚类：K-means 对初始质心的选择敏感，可能收敛于局部最小值。此外，K-means 假设簇的形状是球形的（本文对二维坐标系的网点进行聚类，簇的形状是圆形），对非球形簇效果不佳, 会导致聚为一类的网点间并无直接或间接的道路连接。
- 双层规划：复杂度较高，计算成本大，求解过程可能非常耗时。此外，在本文实际的应用过程中，参数的设定和问题的建模较难，需考虑因素较为复杂。

参考文献

[1] 司守奎, 孙玺菁. 数学建模算法与应用[M]. 国防工业出版社, 2011.

[2] 王小川. 北京航空航天大学出版社[M]. MATLAB 神经网络 43 个案例分析, 2013.

[3] 史峰. MATLAB 智能算法 30 个案例分析[M]. 北京航空航天大学出版社, 2011.

[4] 万孟然, 叶春明. 基于双层规划的物流配送中心选址及配送优化[J]. 复杂系统与复杂性科学: 1-8.

[5] 栾峦. 基于双层规划的生鲜农产品冷链配送中心选址及路径优化研究[D]. 北京交通大学, 2019.

[6] 袁昊, 苏凯凯, 滕泽慧, 等. 基于双层规划的配送中心选址及路径优化方法研究[J]. 物流工程与管理, 2023, 45(10): 38-42.

附录 A　代码文件列表

表 6　**Add caption**

文件名	文件描述
problem1.m + cluster.m	问题 1 求解
problem2.m	问题 2 求解
problem3.m	问题 3 求解

附录 B　代码

problem1.m

```
%% 读取数据

node_location = xlsread('data.xls',"鲜奶网点的位置、订货量");
node_ways = xlsread('data.xls',"鲜奶网点之间的道路");
node_num = node_location(:,1);  % 网点编号
x_node = node_location(:,2);    %网点x坐标
y_node = node_location(:,3);    %网点y坐标

location = [x_node, y_node];    %网点坐标

start_node = node_ways(:,1);    %起始网点
end_node = node_ways(:,2);      %终点


%% 构建邻接矩阵，画图


matrix=zeros(92,92);
for i = 1:140
    X=node_ways(i,1);
    Y=node_ways(i,2);
    matrix(X,Y)=1;
    matrix(Y,X)=1;
end


gplot(matrix,location,"-*")
%% 全距离矩阵

dist_all=zeros(92,92);    % 距离矩阵
```

```
31
32  for i = 1:92
33      for j = 1:92
34          if i ~= j
35              % 计算网点i和网点j之间的距离
36              dist_all(i, j) = sqrt((location(i, 1) - location(j, 1))^2 + (location(i
        , 2) - location(j, 2))^2);
37          end
38      end
39  end
40
41
42
43  %% 构建距离矩阵
44  %计算公路的起点城市与公路的终点城市的距离
45  dist = zeros(1,140);
46  for i=1:140
47      sx=location(start_node(i),1);%起始城市的横坐标
48      ex=location(end_node(i),1);%终止城市的横坐标
49      sy=location(start_node(i),2);%起始城市的纵坐标
50      ey=location(end_node(i),2);%终止城市的纵坐标
51      dist(i)=sqrt(abs(sx-ex)^2+abs(sy-ey)^2);%勾股定理求距离
52  end
53
54  s = start_node';
55  t = end_node';
56  w = dist;
57
58  G = graph(s,t,w);        %无向图
59   %G = digraph(s,t,w);    %有向图
60
61  figure
62  plot(G, 'EdgeLabel', G.Edges.Weight, 'linewidth', 2)
63  set( gca, 'XTick', [], 'YTick', [] );
64
65  %%
66
67  %这个D值在第二三问中要用到
68  D = distances(G); % D为两两节点之间的最短距离
69  temp = 0;
70  for i=1:92
71      for j=1:92
72          temp = temp+D(i,j);
73      end
74      acc_dist(i) = temp;
75      temp = 0;
76  end
```

```
format long g
index = find(acc_dist==min(acc_dist))
Sum_dist_1 = min(acc_dist);
%Q = zeros(92,20);
myplot = plot(G, 'EdgeLabel', G.Edges.Weight, 'linewidth', 2);  %首先将图赋给一个变量
highlight(myplot, index, 'NodeColor', 'g')
for i=1:92
    [P,d] = shortestpath(G,index,i);

    writematrix(P,'M.xls','WriteMode','append')
    % 在图中高亮我们的最短路径
    highlight(myplot, P, 'EdgeColor', 'r')   %对这个变量即我们刚刚绘制的图形进行高亮处理（给边加上r红色）
    hold on
end


%% 实际地理图
figure;
%绘制城市坐标散点图
for i=1:92
    %将前20号的直销中心单独标出
        plot(location(i,1),location(i,2),'b.','MarkerSize',15)%坐标点用蓝色点表示
        text(location(i,1),location(i,2),num2str(node_location(i,1)),'FontSize',8);%标记城市标号
        hold on;
end

%连接各城市间的公路
for i=1:140
    start_x=[location(node_ways(i,1),1),location(node_ways(i,2),1)];%起始城市的横坐标与结束城市的横坐标
    end_y=[location(node_ways(i,1),2),location(node_ways(i,2),2)];%起始城市的纵坐标与结束城市的纵坐标
    plot(start_x,end_y,'b-','LineWidth',1.2)%画线
    hold on;
end
%%
P= xlsread('M.xls');
for i=1:92
    len = find(~isnan(P(i,:)));
    for j=1:length(len)-1
    start_x=[location(P(i,j),1),location(P(i,j+1),1)];%起始城市的横坐标与结束城市的横坐标
    end_y=[location(P(i,j),2),location(P(i,j+1),2)];%起始城市的纵坐标与结束城市的纵坐标
```

```
    plot(start_x,end_y,'r-','LineWidth',1.2)%画线
    hold on;
    end
end
%将配送中心单独标记出来，方便区分
%计算出配送中心的标号为城市3
plot(location(3,1),location(3,2),'gp','MarkerSize',10) %坐标点用绿色*表示

%% 需要的运输车数量
data = P;
for i = 1:92
    len = length(find(~isnan(data(i,:))));
    currentRow = data(i,1:len);
    for j = 1:size(data, 1)
        %len = length(find(~isnan(data(j,:))));
        nextRow = data(j,1:length(find(~isnan(data(j,:)))));
        if (length(currentRow) < length(nextRow)) && isequal(currentRow, nextRow(1:
    length(currentRow)))
            % 删除被包含的行
            data(i,:) = [0];
            break; % 跳出内层循环
        end
    end

end
zero_rows = all(data == 0, 2);

% 删除全部为0的行
car_num = data(~zero_rows, :);
%% 肘部图

features = location;

% 肘部法确定聚类数
sum_of_squared_distances = [];
max_clusters = 10;

for k = 1:10
    [~, w, sumd] = kmeans(features, k, 'Replicates', 10);
    sum_of_squared_distances = [sum_of_squared_distances; sum(sumd) ];
end
%绘制肘部法
figure;
plot(1:max_clusters, sum_of_squared_distances, '-o','LineWidth',1.2);
xlabel('聚类数K');
ylabel('总的平方距离和');
title('肘部法确定聚类数');
```

```
set(gca, 'FontName', 'Microsoft YaHei', 'FontSize', 15);

%% 聚类
% 选择最佳聚类数
optimal_clusters= 3; % 根据图形选择的最佳聚类数

% 执行K-means聚类
[idx, C] = kmeans(features, optimal_clusters, 'Replicates', 10);

% 绘制聚类结果
figure;
gscatter(x_node,y_node,idx);
%scatter(C(:,1),C(:,2),'filled','green')
hold on;
xlabel('X坐标');
ylabel('Y坐标');
title('K-means聚类结果');
legend('Cluster 1', 'Cluster 2', 'Cluster 3','配送站');
grid on;
hold off;
set(gca, 'FontName', 'Microsoft YaHei', 'FontSize', 15);

%%


for i = 1:3
    clusterIndices = (idx == i);
    if i == 1
        cluster1 = node_num(clusterIndices);
    end
    if i == 2
        cluster2 = node_num(clusterIndices);
    end
    if i == 3
        cluster3 = node_num(clusterIndices);
    end
end

start_node1 = [];
start_node2 = [];
start_node3 = [];
end_node1 = [];
end_node2 = [];
end_node3 = [];
dist1 = [];
dist2 = [];
dist3 = [];
```

```
for i = i:140
    if ismember(node_ways(i),cluster1)
         start_node1 = [start_node1,start_node(i)];
         end_node1 = [end_node1,end_node(i)];
         dist1 = [dist1,dist(i)];
    end
    if ismember(node_ways(i),cluster2)
         start_node2 = [start_node2,start_node(i)];
         end_node2 = [end_node2,end_node(i)];
         dist2 = [dist2,dist(i)];
    end
    if ismember(node_ways(i),cluster3)
         start_node3 = [start_node3,start_node(i)];
         end_node3 = [end_node3,end_node(i)];
         dist3 = [dist3,dist(i)];
    end
end

s_c1 = start_node1;
t_c1 = end_node1;
w_c1 = dist1;

G1 = graph(s_c1,t_c1,w_c1);%无向图
 %G = digraph(s,t,w);%有向图
 s_c2 = start_node2;
t_c2 = end_node2;
w_c2 = dist2;

G2 = graph(s_c2,t_c2,w_c2);%无向图

s_c3 = start_node3;
t_c3 = end_node3;
w_c3 = dist3;

G3 = graph(s_c3,t_c3,w_c3);%无向图

figure
plot(G2,'EdgeLabel', G2.Edges.Weight, 'linewidth', 2)
set( gca, 'XTick', [], 'YTick', [] );
hold on

%end
```

cluster.m

```
%% 聚类 分3类
% 选择最佳聚类数
optimal_clusters = 3; % 根据图形选择的最佳聚类数
```

```
4
5  % 执行K-means聚类
6  [idx, C] = kmeans(features, optimal_clusters, 'Replicates', 10);
7  %% after 手动调整
8
9  % 绘制聚类结果
10 figure;
11 gscatter(x_node,y_node,idx);
12 hold on
13 scatter(C(:,1),C(:,2),'filled','green')
14 hold on;
15 xlabel('X坐标');
16 ylabel('Y坐标');
17 title('K-means聚类结果');
18 legend('Cluster 1', 'Cluster 2','Cluster 3','配送站');
19 grid on;
20 hold off;
21 set(gca, 'FontName', 'Microsoft YaHei', 'FontSize', 15);
22
23 %% 画图
24
25
26 for i = 1:3
27     clusterIndices = (idx == i);
28     if i == 1
29         cluster1 = node_num(clusterIndices);
30     end
31     if i == 2
32         cluster2 = node_num(clusterIndices);
33     end
34     if i == 3
35         cluster3 = node_num(clusterIndices);
36     end
37 end
38
39 start_node1 = [];
40
41 end_node1 = [];
42
43 dist1 = [];
44 for i = 1:140
45     if all(ismember(node_ways(i,:),cluster1)) || all(ismember(node_ways(i,:),
    cluster2)) || all(ismember(node_ways(i,:),cluster3))
46          start_node1 = [start_node1,start_node(i)];
47          end_node1 = [end_node1,end_node(i)];
48          dist1 = [dist1,dist(i)];
49     end
```

```
    % if ismember(node_ways(i),cluster2)
    %       start_node2 = [start_node2,start_node(i)];
    %       end_node2 = [end_node2,end_node(i)];
    %       dist2 = [dist2,dist(i)];
    % end
    % if ismember(node_ways(i),cluster3)
    %       start_node3 = [start_node3,start_node(i)];
    %       end_node3 = [end_node3,end_node(i)];
    %       dist3 = [dist3,dist(i)];
    % end
end

s_c1 = start_node1;
t_c1 = end_node1;
w_c1 = dist1;

G1 = graph(s_c1,t_c1,w_c1);%无向图


%求出距离和
D_c3 = distances(G1);
D_c3(isinf(D_c3)) = 0;

% 计算每一行的和，找出和最小的行
row_sums = sum(D_c3, 2);

for i = 1:3

 clusterIndices = (idx == i);

    if i == 1
        row_sums_1 = row_sums(clusterIndices);
        [min_sum(i), min_row(i)] = min(row_sums_1);
        cluster_min_num(i,:) = [cluster1(min_row(i)),min_sum(i)];  %每个聚类的配送站，即其距离和
    end
    if i == 2
        row_sums_2 = row_sums(clusterIndices);
        [min_sum(i), min_row(i)] = min(row_sums_2);
        cluster_min_num(i,:) = [cluster2(min_row(i)),min_sum(i)];
    end
    if i == 3
        row_sums_3 = row_sums(clusterIndices);
        [min_sum(i), min_row(i)] = min(row_sums_3);
        cluster_min_num(i,:) = [cluster3(min_row(i)),min_sum(i)];
    end
end
```

```
96
97  %配送站
98  center_idx = cluster_min_num(:,1);
99
100 % 距离和
101 Sum_dist_3 = sum(cluster_min_num(:,2));
102
103 %%
104 figure
105 myplot = plot(G1,'EdgeLabel', G1.Edges.Weight, 'linewidth', 2) ;
106 highlight(myplot, cluster_min_num(:,1), 'NodeColor', 'g')
107 %set( gca, 'XTick', [], 'YTick', [] );
108 hold on
109
110 for j = 1:3
111 for i = 1:92
112     [S_P,d] = shortestpath(G1,cluster_min_num(j,1),i);
113
114     writematrix(S_P,'Cluster_short.xls','WriteMode','append')
115     % 在图中高亮我们的最短路径
116     highlight(myplot,S_P, 'EdgeColor', 'r')   %对这个变量即我们刚刚绘制的图形进行高
        亮处理（给边加上r红色）
117     hold on
118 end
119 end
120
121 %% 实际地理图
122 figure;
123 %绘制城市坐标散点图
124 for i=1:92
125     %将前20号的直销中心单独标出
126         plot(location(i,1),location(i,2),'b.','MarkerSize',15)%坐标点用蓝色点表示
127         text(location(i,1),location(i,2),num2str(node_location(i,1)),'FontSize',15)
        ;%标记城市标号
128         hold on;
129 end
130
131 %连接各城市间的公路
132 for i=1:140
133     start_x=[location(node_ways(i,1),1),location(node_ways(i,2),1)];%起始城市的横坐
        标与结束城市的横坐标
134     end_y=[location(node_ways(i,1),2),location(node_ways(i,2),2)];%起始城市的纵坐标
        与结束城市的纵坐标
135     plot(start_x,end_y,'b-','LineWidth',1.5)%画线
136     hold on;
137 end
138
```

```
P= xlsread('Cluster_short.xls');
for i=1:92
    len = find(~isnan(P(i,:)));
    for j=1:length(len)-1
    start_x=[location(P(i,j),1),location(P(i,j+1),1)];%起始城市的横坐标与结束城市的横坐标
    end_y=[location(P(i,j),2),location(P(i,j+1),2)];%起始城市的纵坐标与结束城市的纵坐标
    plot(start_x,end_y,'r-','LineWidth',1.5)%画线
    hold on;
    end
end
%将配送中心单独标记出来，方便区分
%计算出配送中心的标号为城市
plot(location(47,1),location(47,2),'g.','MarkerSize',17) %坐标点用绿色*表示
plot(location(1,1),location(1,2),'g.','MarkerSize',17) %坐标点用绿色*表示
plot(location(11,1),location(11,2),'g.','MarkerSize',17) %坐标点用绿色*表示

%% 需要的运输车数量
data = P;
for i = 1:92
    len = length(find(~isnan(data(i,:))));
    currentRow = data(i,1:len);
    for j = 1:size(data, 1)
        %len = length(find(~isnan(data(j,:))));
        nextRow = data(j,1:length(find(~isnan(data(j,:)))));
        if (length(currentRow) < length(nextRow)) && isequal(currentRow, nextRow(1:length(currentRow)))
            % 删除被包含的行
            data(i,:) = [0];
            break; % 跳出内层循环
        end
    end

end
zero_rows = all(data == 0, 2);

% 删除全部为0的行
car_num = data(~zero_rows, :);
```

problem2.m

```
%43,11,1
%max = 333.3
max_D_3 = max(D(3,:))

max_D_43 = max(D_c3(43,:))
max_D_11 = max(D_c3(11,:))
```

```
max_D_1 = max(D_c3(1,:))
```

problem3.m

```
% 定义每个网点的供应量和需求量

supply_demand = node_location(:, 4);


for i = 1:3

 clusterIndices = (idx == i);

    if i == 1
        supply_demand_1 = supply_demand(clusterIndices);
        supply_demand_all(i) = sum(supply_demand_1);
    end
    if i == 2
        supply_demand_2 = supply_demand(clusterIndices);
        supply_demand_all(i) = sum(supply_demand_2);
    end
    if i == 3
        supply_demand_3 = supply_demand(clusterIndices);
        supply_demand_all(i) = sum(supply_demand_3);
    end
end

avg_supply_demand = sum(supply_demand_all) / 3;
%% 均衡配送量
%supply_demand_center = zeros(3, 1);
center_idx = cluster_min_num(:,1);  %配送站

idx_supply = idx;
while (min(supply_demand_all) < (0.9*avg_supply_demand)) ||  (max(supply_demand_all
    ) > (1.1*avg_supply_demand))
for k = 1:3
    %clusterIndices = (idx_supply == k);
    while (supply_demand_all(k) < (avg_supply_demand)) %||  (supply_demand_all(k) >
     (1.1*avg_supply_demand))
        % 将未分配的网点中到该配送站点距离最近的网点分配给该站点
        unassigned_idx =  setdiff(node_num(idx_supply == 2 | idx_supply == 1),
    node_num(idx_supply == k));
        [~, idx_min] = min(dist_all(unassigned_idx, center_idx(k)));
        i = unassigned_idx(idx_min);
        idx_supply(i) = k;
        for j = 1:3
            clusterIndices1 = (idx_supply == j);
            supply_demand_all(j) = sum(supply_demand(clusterIndices1));
```

```
        end
        %supply_demand_all
        %center_idx(k) = [center_idx(k); i];
    end
end
end
% centers = data(center_idx, 1:2);
% disp('第三问的配送站点：');
% disp(centers);

%% 找出三个配送站点


for i = 1:3
    clusterIndices = (idx_supply == i);
    if i == 1
        cluster1 = node_num(clusterIndices);
    end
    if i == 2
        cluster2 = node_num(clusterIndices);
    end
    if i == 3
        cluster3 = node_num(clusterIndices);
    end
end

start_node1 = [];

end_node1 = [];

dist1 = [];
for i = 1:140
    if all(ismember(node_ways(i,:),cluster1)) || all(ismember(node_ways(i,:),
    cluster2)) || all(ismember(node_ways(i,:),cluster3))
         start_node1 = [start_node1,start_node(i)];
         end_node1 = [end_node1,end_node(i)];
         dist1 = [dist1,dist(i)];
    end
    % if ismember(node_ways(i),cluster2)
    %      start_node2 = [start_node2,start_node(i)];
    %      end_node2 = [end_node2,end_node(i)];
    %      dist2 = [dist2,dist(i)];
    % end
    % if ismember(node_ways(i),cluster3)
    %      start_node3 = [start_node3,start_node(i)];
    %      end_node3 = [end_node3,end_node(i)];
    %      dist3 = [dist3,dist(i)];
```

```
    % end
end

s_c1 = start_node1;
t_c1 = end_node1;
w_c1 = dist1;

G1 = graph(s_c1,t_c1,w_c1);%无向图

%求出距离和
D_supply = distances(G1);
D_supply(isinf(D_supply)) = 0;
% 计算每一行的和
% 找出和最小的行
row_sums = sum(D_supply, 2);

for i = 1:3

 clusterIndices = (idx_supply == i);

    if i == 1
        row_sums_1 = row_sums(clusterIndices);
        [min_sum(i), min_row(i)] = min(row_sums_1);
        cluster_min_num(i,:) = [cluster1(min_row(i)),min_sum(i)];  %每个聚类的配送站，即其距离和
    end
    if i == 2
        row_sums_2 = row_sums(clusterIndices);
        [min_sum(i), min_row(i)] = min(row_sums_2);
        cluster_min_num(i,:) = [cluster2(min_row(i)),min_sum(i)];
    end
    if i == 3
        row_sums_3 = row_sums(clusterIndices);
        [min_sum(i), min_row(i)] = min(row_sums_3);
        cluster_min_num(i,:) = [cluster3(min_row(i)),min_sum(i)];
    end
end

%配送站
center_idx = cluster_min_num(:,1)

% 距离和
Sum_dist_3 = sum(cluster_min_num(:,2))

%%
figure
myplot = plot(G1,'EdgeLabel', G1.Edges.Weight, 'linewidth', 2) ;
```

```
highlight(myplot, cluster_min_num(:,1), 'NodeColor', 'g')
%set( gca, 'XTick', [], 'YTick', [] );
hold on

for j = 1:3
for i = 1:92
    [S_P,d] = shortestpath(G1,cluster_min_num(j,1),i);

    writematrix(S_P,'Supply_short.xls','WriteMode','append')
    % 在图中高亮我们的最短路径
    highlight(myplot,S_P, 'EdgeColor', 'r')   %对这个变量即我们刚刚绘制的图形进行高亮处理（给边加上r红色）
    hold on
end
end

%% 实际地理图
figure;
%绘制城市坐标散点图
for i=1:92
    %将网点单独标出
        plot(location(i,1),location(i,2),'b.','MarkerSize',15)%坐标点用蓝色点表示
        text(location(i,1),location(i,2),num2str(node_location(i,1)),'FontSize',15);%标记城市标号
        hold on;
end

%连接各城市间的公路
for i=1:140
    start_x=[location(node_ways(i,1),1),location(node_ways(i,2),1)];%起始城市的横坐标与结束城市的横坐标
    end_y=[location(node_ways(i,1),2),location(node_ways(i,2),2)];%起始城市的纵坐标与结束城市的纵坐标
    plot(start_x,end_y,'b-','LineWidth',1.5)%画线
    hold on;
end

P= xlsread('Supply_short.xls');
for i=1:92
    len = find(~isnan(P(i,:)));
    for j=1:length(len)-1
    start_x=[location(P(i,j),1),location(P(i,j+1),1)];%起始城市的横坐标与结束城市的横坐标
    end_y=[location(P(i,j),2),location(P(i,j+1),2)];%起始城市的纵坐标与结束城市的纵坐标
    plot(start_x,end_y,'r-','LineWidth',1.5)%画线
    hold on;
```

```
    end
end
%将配送中心单独标记出来，方便区分
%计算出配送中心的标号为城市
plot(location(51,1),location(51,2),'g.','MarkerSize',17) %坐标点用绿色*表示
plot(location(18,1),location(18,2),'g.','MarkerSize',17) %坐标点用绿色*表示
plot(location(34,1),location(34,2),'g.','MarkerSize',17) %坐标点用绿色*表示

%% 需要的运输车数量
data = P;
for i = 1:92
    len = length(find(~isnan(data(i,:))));
    currentRow = data(i,1:len);
    for j = 1:size(data, 1)
        %len = length(find(~isnan(data(j,:))));
        nextRow = data(j,1:length(find(~isnan(data(j,:)))));
        if (length(currentRow) < length(nextRow)) && isequal(currentRow, nextRow(1:length(currentRow)))
            % 删除被包含的行
            data(i,:) = [0];
            break; % 跳出内层循环
        end
    end

end
zero_rows = all(data == 0, 2);

% 删除全部为0的行
car_num = data(~zero_rows, :);
```