装　订　线

西安工业大学数学建模竞赛
暨全国大学生数学建模竞赛选拔赛题目

A 题

密封号		2020 年 6 月 27 日

------------------------------------剪　切　线------------------------------------

密封号		2020 年 6 月 27 日

理 学院　　第 056 队

	队员 1	队员 2	队员 3
姓名	葛伟鹏	蒋妮	范宇璇
学号	17100101103	17100306118	17100101118

承　诺　书

仔细阅读了中国大学生数学建模竞赛的竞赛规则.

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。

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。

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，将受到严肃处理。

授权全国大学生数学建模竞赛组委会，可将的论文以任何形式进行公开展示（包括进行网上公示，在书籍、期刊和其他媒体进行正式或非正式发表等)。

参赛选择的题号是：A 题

的参赛报名号为：

所属学校：西安工业大学

参赛队员：1. 葛伟鹏

2. 蒋妮

3. 范宇璇

指导教师或指导教师组负责人 (打印并签名)：

日期：2020 年 6 月 27 日

赛区评阅编号：

优化自来水管道铺设方案

摘要

在村村通自来水工程实施过程中，为保证供水质量以及设备维护方便，需要建立中心供水站，一级供水站和二级供水站，并且保证在给定条件下的铺设总里程达到最小。对给定供水站数目的问题，用分层的最小生成树模型解决；对升级指定要升级数目的供水站问题，用单目标优化模型找出需要升级的供水站，再采用分层的最小生成树模型解决；对要升级若干个供水站保证总里程最小，且对供水站的输送里程做出限制的问题，用多目标优化模型解决。

对给定供水站数目的问题，这是一个图论问题，建立图论与网络优化问题模型，对于图中的供水站分为三层，并对每一层做最小生成树，最后连接不同生成树之间的最短节点可以达总里程最短。用Kruskal算法实现了不同层之间的最小生成树，最后得出若视一级管道可以和二级管道连接0公里的总里程为542.296公里,否则总里程为543.296公里，铺设方案详见正文图（11)、图（12)、图（13)。

对升级指定要升级数目的供水站问题，用单目标优化模型找出需要升级的供水站，再采用分层的最小生成树模型解决，这是一个约束条件下的优化问题，先建立单目标优化模型找出一级供水站到二级供水站最短路中最大的两个，再找出二级供水站对应的最小最短路，最终选择升级 P47 和 P73 供水站，再按照第一问的分层的最小生成树求出II型管道的总里程若视一级管道可以和二级管道连接 0 公里减少了 2.0132 公里，否则减少了 3.0132 公里，铺设方案详见正文图（22)、图（23)、图（24)、图(25)。

对要升级若干个供水站保证总里程最小，且对供水站的输送里程做出限制的问题，用递归的方法先找出现有一级供水站所能到达的二级水站，并保证每个一级水站附件的二级水站满足最小生成树，接着再将未铺设到的地方升级新的一级供水站，接着用此方法继续递归直至所有供水站被覆盖，此时便得到最短总里程。

对于分层的最小生成树模型没有对不分层的最小生成树模型进行对比，得到的结果会没有参考，后边可以对两种模型进行对比，得到参考依据；对要升级若干个供水站保证总里程最小，且对供水站的输送里程做出限制的递归方法，采用的解题策略较为麻烦，后面可以将此模型替换为动态规划问题加以求解，虽然最后得到的并非最优解，但可以简化解题步骤，得到次优解。

本题主要求解的工具是MATLAB软件和Excel软件。

【关键词】图论模型　　分层的最小生成树模型　　Kruskal算法　　优化模型

一、问题重述

随着经济迅猛发展，环境问题迫在眉睫，多地水资源极度匮乏，导致居民饮水已经严重影响起居，无法保证生活状态。现政府大力推举修建自来水工程，解决居民供水问题，某地区已有一个中心供水站，为保证供水质量以及设备维护方便，在村村通自来水工程实施过程中，地区中含有 12 个一级供水站和 168 个二级供水站（各级供水站的位置坐标已知），现在要将中心供水站 A 处的自来水通过管道输送到一级供水站和二级供水站，设计要求，从中心站 A 铺设到一级供水站的管道为Ⅰ型管道，从一级供水站出发铺设到二级供水站的管道为Ⅱ型管道。目前为止，如何有效地安排自来水管道铺设问题、如何在减少资源浪费的前提下成功地解决居民饮水问题，显得尤为重要。

二、问题分析

1. 问题一的分析

此题需要考虑的是如何铺设才能使得总里程数最少的图论与网络优化问题模型，采用分层的最小生成树（MST）的 Kruskal 算法为解题思路。找出中心供水站点 A，将一级供水站和二级供水站分别构造出两层最小生成树，以第二层最小生成树为基点，寻找与之相临近的一级供水站，从而形成关联，构造出第三层最小生成树。因第一层和第二层的最小生成树是不会互相影响的，所以第一层的最小生成树和第二层的最小生成树分别最小方可保证总的路程最小，直至找出满足题意的最短路。

2. 问题二的分析

本题要求将其中两个二级供水站升级为一级供水站，并且要使铺设的Ⅱ型管道总里程最少，采取单目标优化模型和图论与网络优化问题模型，结合分层的最小生成树（MST）的 Kruskal 算法为解题思路。分别以一级供水站和二级供水站为圆心，寻找与之最近且为不同类型的供水站，分别将得到的欧氏距离进行从小到大排序，取最大值，即为所需要升级的二级供水站。

3. 问题三的分析

本题为实现对所有供水站供水，需要将若干个二级供水站升级为一级供水站，要求升级最少的二级供水站以达到总里程最少的目标，采用递归的方法先找出现有一级供水站所能到达的二级水站，并保证每个一级水站附件的二级水站满足最小生成树，接着再将未铺设到的地方升级新的一级供水站，接着用此方法继续递归直至所有供水站被覆盖，此时便得到最短总里程。

三、模型的假设

1. 假设供水站与居民点的距离为直线距离，即忽略掉输水管道的路线问题。
2. 假设供水站与居民点之间的供水费用仅与供水长度有关，和输水量无关。
3. 假设供水站的建设资金是确定的，不会因规模的大小而改变，成本仅为供水成本。
4. 假设供水站和居民区都是理性化的质点。
5. 假设居民的用水量就为人均用水量乘上人口数，而且长期不变。
6. 假设一级供水站到二级供水站可以重合且铺设所需的Ⅱ级管道距离为 0 的情况存在。

四、定义与符号说明

符号	符号说明
X	中心供水站的横坐标
X_i	一级供水站的横坐标
X_j	二级供水站的横坐标
Y	中心供水站的纵坐标
Y_i	一级供水站的纵坐标
Y_j	二级供水站的纵坐标
d	中心供水站到一级供水站之间距离
R_a	任意两一级供水站之间最短距离
R_b	任意两二级供水站之间最短距离
R_c	任意一二级供水站之间最短距离
S_1	一级供水站之间铺设的管道总长度
S_2	二级供水站之间铺设的管道总长度
S	铺设管道总距离

五、模型的建立和求解

1. 问题一

1.1 模型建立

（1）设所求最短路径的目标函数如下：

$$\min S=\sum_{i=1,2} S_i + d$$

（2）约束条件如下：

$$\text{s.t}\begin{cases} S_1=\sum\limits_{1\le a\le 11} R_a \\ S_2=\sum\limits_{1\le b\le 167} R_b \\ d=\min(\sqrt{(X\ \ -X_i)^2+(Y-Y_i)^2}) \end{cases}$$

注：其中R_a与R_b通过 MATLAB 计算所得的欧氏距离中选取距离最短的那一组数据所得，由于一级供水站与二级供水站中有一点是重合的，因此选择那个重合点作为一、二级水站的链接点，所以从一级水站联通到二级水站的距离为 0，不作考虑。

1.2 模型求解及分析

1.2.1 数据预处理

将题中的中心供水站 A 点、一级供水站V（i,j），二级供水站$P(i,j)$的坐标数据导入 Excel 中，并分别计算出一级供水站之间、二级供水站之间的欧氏距离。详见支撑材料附录 3“供水站坐标”、“欧氏距离”。

1.2.2 图像预处理

将题中已知的散点图分离为中心供水站散点图、一级供水站散点图、二级供水站散点图

1.2.3 建立无向图

将预处理过后的散点图建立无向图：中心供水站无向图（图 1）、一级供水站无向图（图 2）、二级供水站无向图（图 3）、三层供水站的无向图（图 4）

以建立三层供水站的无向图为例，程序详见附录 1“程序 1”“LineUp.m”

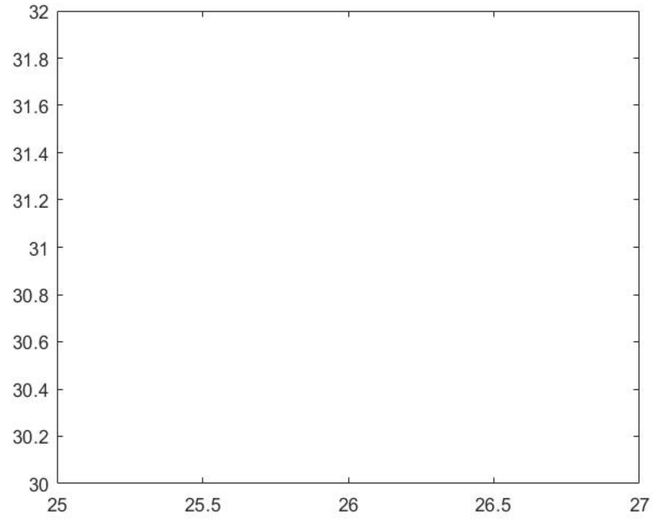

图 1：中心供水站的无向图

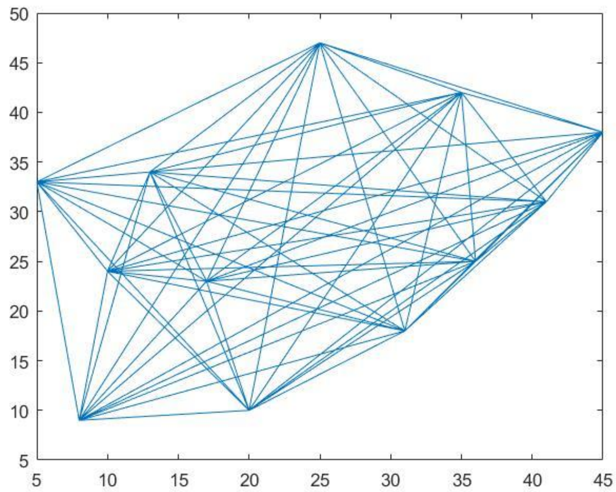

图 2 一级供水站的无向图

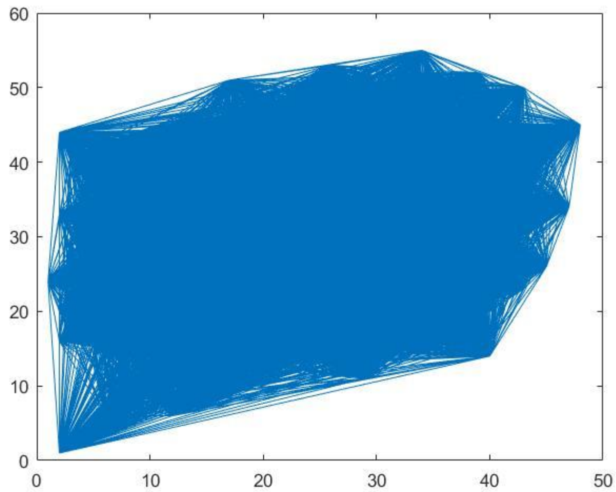

图 3 二级供水站的无向图

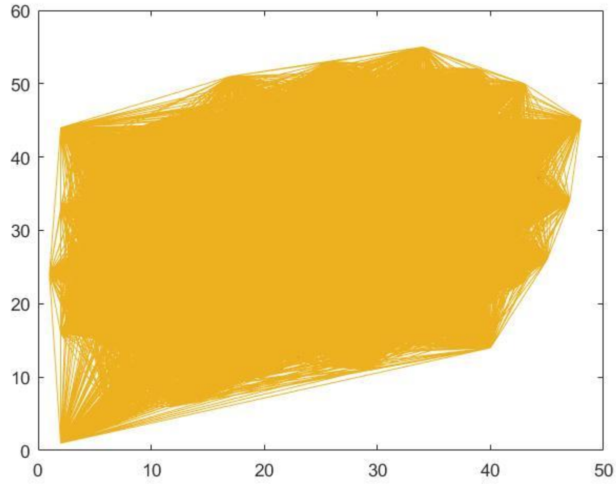

图 4：三层供水站的无向图

1.2.4 使用 MATLAB 工具箱构造最小生成树（MST）

构造每一层的最小生成树，详见程序见附录 1“程序 2”“Graphminspantree.m”;

构造中心站 A 的最小生成树结果见支撑材料附录 2“问题一结果 1”;

构造一级供水站的最小生成树结果见支撑材料附录 2“问题一结果 2”;

构造二级供水站的最小生成树结果见支撑材料附录 2“问题一结果 3”;

在构造完每一层最小生成树之后，寻找最小生成树，其次构造总的最小生成树，详见附录 1“程序 5”“MinTree.m”

注：数据存放于 Excel 中备份，详见支撑材料附录 3“最小生成树”

若使用自己编写 Kruskal 算法程序经实验后非常繁杂，故用 MATLAB 工具箱代替之，以便减少运算量。Kruskal 算法详见附录 1“程序 3”“Kruskal.m”

分别生成最小生成树图：中心供水站最小生成树图（图 5）、一级供水站最小生成树图（图 6）、二级供水站最小生成树图（图 7）、三层供水站的最小生成树图（图 8）

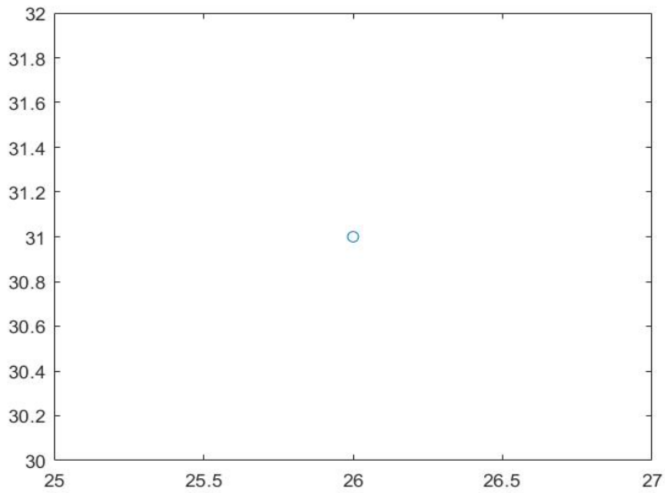

图 5 中心供水站最小生成树图

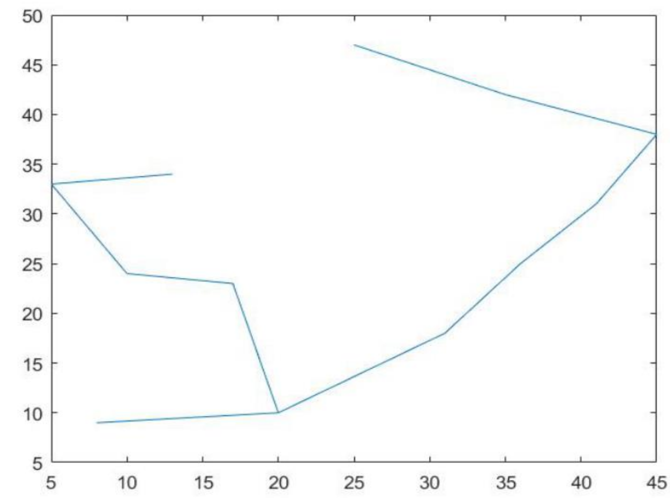

图 6 一级供水站最小生成树图

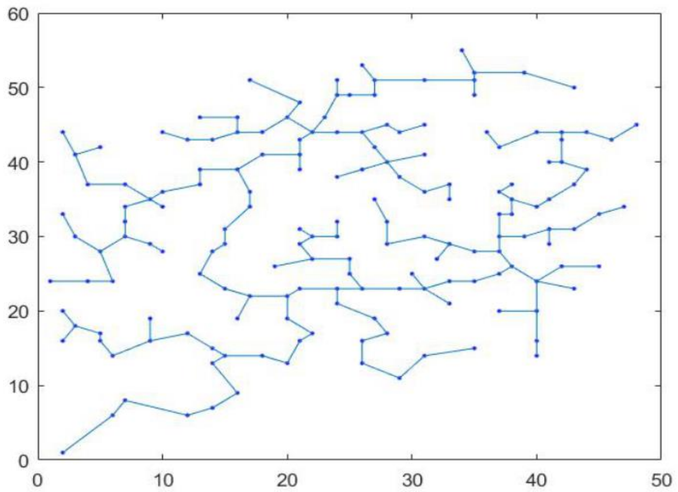

图 7 二级供水站最小生成树图

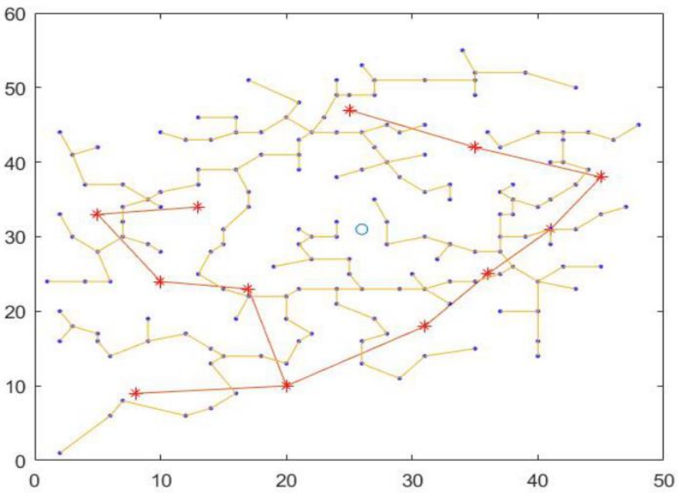

图 8 三层供水站的最小生成树图

1.2.5 寻找最短路

1.2.5.1 找出中心供水站与一级供水站之间的最短距离的一级供水站点

通过对欧式距离的排序找到中心供水站与一级供水站之间的最短距离的一级供水站为 V10，其坐标为（36， 25），欧氏距离为 11.6619，详见图 9，程序见附录 1“程序 4”“Round.m”，结果见表 1

表 1：中心供水站到一级供水站之间的最短距离点的坐标

序号	类型	X 坐标	Y 坐标		序号	类型	X 坐标	Y 坐标
0	A	26	31	→	10	V10	36	25

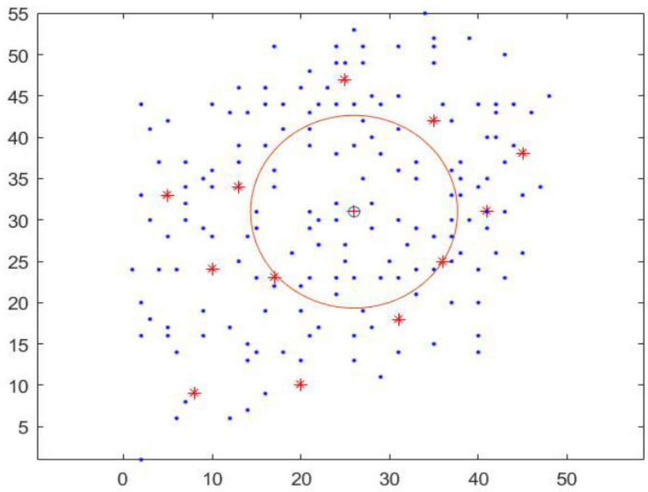

图 9 中心供水站和一级供水站之间的最短距离

1.2.5.2 找出一级水站与二级供水站之间的最短距离的一级供水站点

通过对欧式距离的排序找出一级水站与二级供水站之间的最短距离，结果为：V5→P125，欧氏距离为 1；V10→P135，欧式距离为 1；V11→P139，欧氏距离为 0。详见图 10，结果见表 2

表 2：一级供水站到二级供水站之间的最短距离点的坐标

序号	类型	X 坐标	Y 坐标		序号	类型	X 坐标	Y 坐标
5	V5	17	23	→	135	P125	17	22
10	V10	36	25	→	145	P135	37	25
11	V11	41	31	→	149	P139	41	31

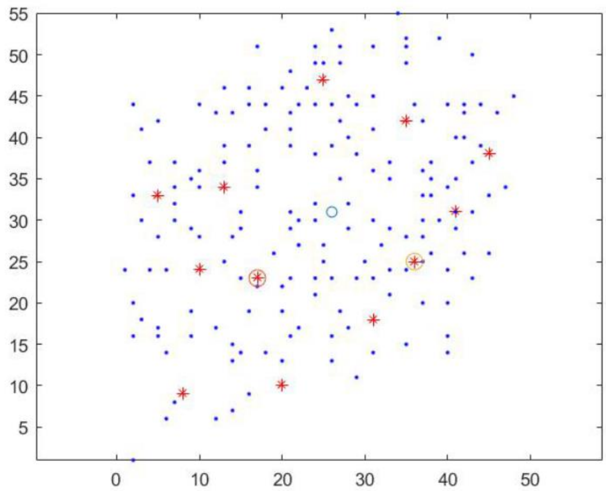

图 10 一级供水站到二级供水站之间的最短距离

1.3 结果

中心供水站向一级供水站的路线为是 A→V10,一级供水站向二级供水站的路线是由 V5→P125 或者 V10→P135 二选一即可。程序详见附录 1“程序 6”“anser.m”

1.3.1 最终铺设路线为 V11→P139（见图 11）

$\min(d) = 11.6619$ 公里；$\min(R_c) = 0$ 公里

$S_1 = 110.8392$ 公里；$S_2 = 419.7949$ 公里

$\min S = \min(d) + \min(R_c) + S_1 + S_2$

$S = 542.296$ 公里

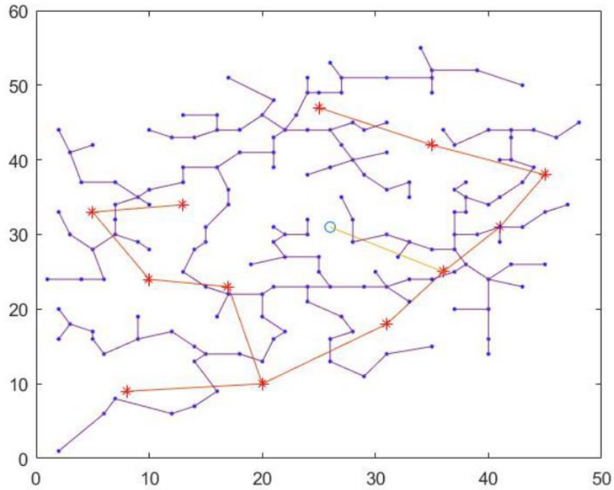

图 11 总铺设路线（V11→P139）

1.3.2 最终铺设路线为 V5→P125（见图 12）

$\min(d) = 11.6619$ 公里；$\min(R_c) = 1$ 公里

$S_1 = 110.8392$ 公里；$S_2 = 419.7949$ 公里

$\min S = \min(d) + \min(R_c) + S_1 + S_2$

$S = 543.296$ 公里

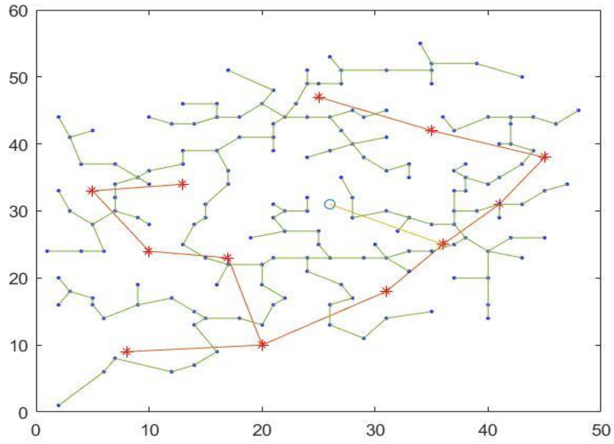

图 12 总铺设路线（V5→P125）

1.3.3 最终铺设路线为 V10→P135（见图 13）

$\min(d) = 11.6619$ *公里*；$\min(R_c) = 1$ *公里*

$S_1 = 110.8392$ *公里*；$S_2 = 419.7949$ *公里*

$\min S = \min(d) + \min(R_c) + S_1 + S_2$

$S = 543.296$ *公里*

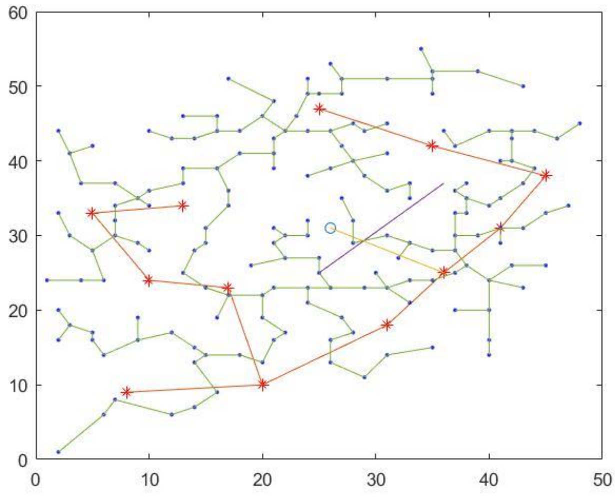

图 13 总铺设路线（V10→P135）

2．问题二

2.1 模型建立

（1）确定两个供水站

设立目标函数求得一级供水站与二级供水站间铺设管道的最远距离：

$$\max R = \sqrt{(X_i - X_j)^2} + \sqrt{(Y_i - Y_j)^2}$$

$$s.t\begin{cases}\sqrt{(X_i-X_j)^2}+\sqrt{(Y_i-Y_j)^2}\le R_c\\1\le i\le 12\\1\le j\le 168\end{cases}$$

设立目标函数求得任一二级供水站与其相邻二级供水站铺设II型管道的最近距离，($Xj1$，$Yj1$)与($Xj2$，$Yj2$)分别表示任意两二级供水站间的坐标

$$\min R_2=\sqrt{(X_{j1}-X_{j2})^2}+\sqrt{(Y_{j1}-Y_{j2})^2}$$

$$s.t\begin{cases}1\le j\le 168\\j_1\ne j_2\end{cases}$$

（2）求出升级后的最短路

在找到计划升级为一级供水站的两个二级供水站的坐标点后，将问题二转化为 1 个总供水站、14 个一级供水站、166 个二级供水站的铺设管道问题，因此与第一问同理，利用 Kruskal 最小生成树原理寻求铺设管道的最佳方案。

故设立目标函数如下：

$$\min S=\sum_{i=1,2}S_i+d$$

$$\text{s.t}\begin{cases}S_1=\sum\limits_{1\le a\le 13}R_a\\S_2=\sum\limits_{1\le b\le 165}R_b\\d=\min(\sqrt{(X\ \ -X_i)^2+(Y-Y_i)^2})\end{cases}$$

注：其中R_a与R_b通过 MATLAB 计算所得的欧氏距离中选取距离最短的那一组数据所得，一级供水站与二级供水站的连接点仍然选择那两个重合点，与第一问同理，不作考虑。

2.2 模型求解及分析

2.2.1 数据预处理

2.2.1.1 同层次的欧氏距离

将题中的中心供水站 A 点、升级过后的一级供水站V（i,j），二级供水站$P(i,j)$的坐标数据导入 Excel 中，并分别计算出一级供水站之间、二级供水站之间的欧氏距

离。详见支撑材料附录 3“供水站坐标 2”、“欧氏距离 2”。

2.2.1.2 找两点

以一级供水站为中心画圆，找到与之相邻距离最大的点

再以二级供水站为中心画圆，找到与之相邻距离最小的点

综合上述进而确定两点为 P47,P73

详见支撑材料附录 3“找点表”

2.2.1.3 不同层次的欧氏距离

计算出不同层次间的欧氏距离，数据记录详见支撑材料附录 3“不同层最短欧氏距离”

2.2.2 图像预处理

将已找到的两个点 P47,P73 均升级为一级供水站，同时更新散点图，详见图 14，“程序 7”“ScatterDiagram.m”

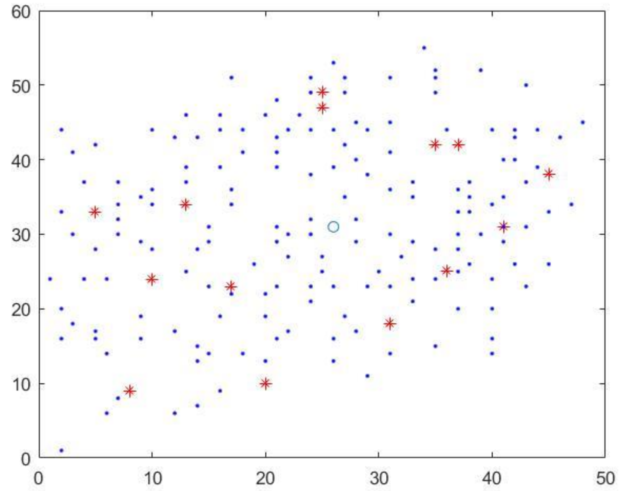

图 14 更新后的散点图

2.2.3 建立无向图

经过预处理后的图像建立两两相连的无向图，详见图 15

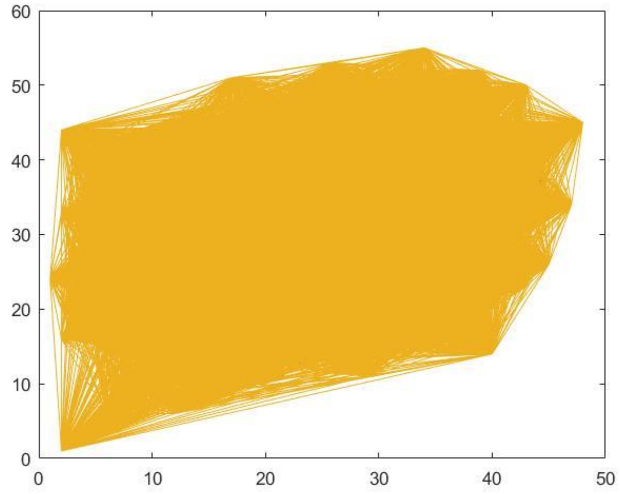

图 15 三层两两相连的无向图

2.2.4 使用 MATLAB 工具箱构造最小生成树（MST）

最小生成树的程序与问题一类似，故此不赘述；

构造中心站 A 的最小生成树结果见支撑材料附录 2“问题二结果 4”；

构造一级供水站的最小生成树结果支撑材料见附录 2“问题二结果 5”；

构造二级供水站的最小生成树结果支撑材料见附录 2“问题二结果 6”；

注：数据存放于 Excel 中备份，将数据转为对角矩阵，进一步转为稀疏矩阵的程序详见附录 1“程序 8”“gauss.m”，数据见支撑材料附录 3“最小生成树 2”

若使用自己编写程序经实验后非常繁杂，故用 MATLAB 工具箱代替之，以便减少运算量。

分别生成最小生成树图：中心供水站最小生成树图（图 16）、一级供水站最小生成树图（图 17）、二级供水站最小生成树图（图 18）、三层供水站的最小生成树图（图 19）

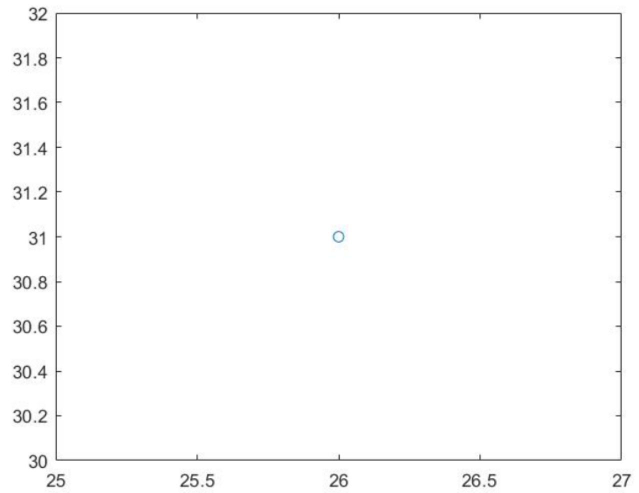

图 16：中心供水站最小生成树

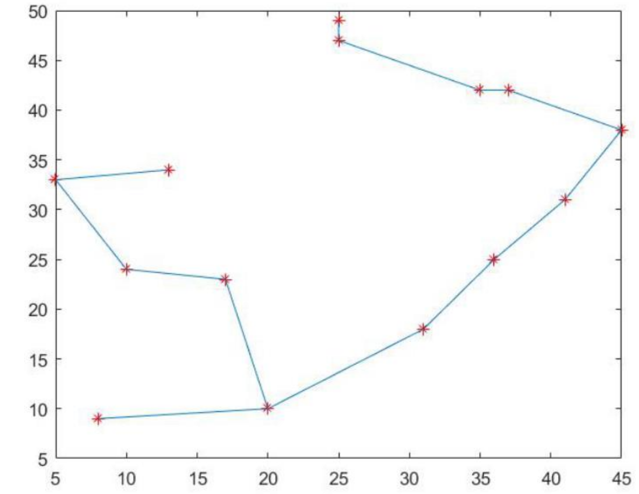

图 17：一级供水站最小生成树

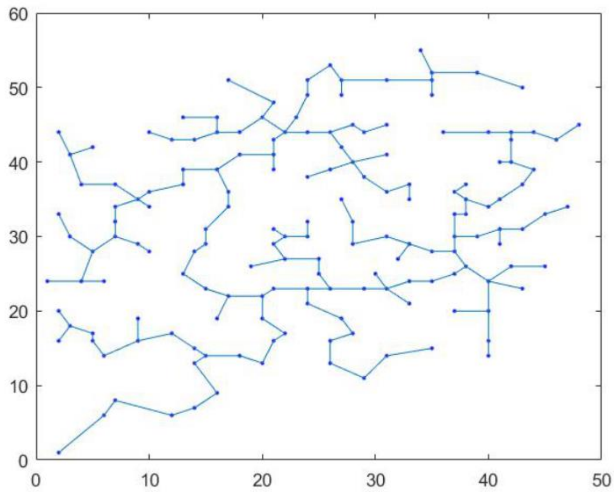

图 18：二级供水站最小生成树

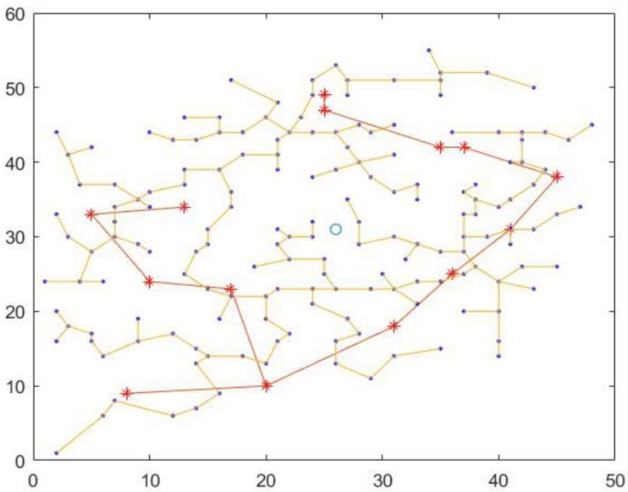

图 19：三层最小生成树的图

2.2.5 寻找最短路

2.2.5.1 找出中心供水站与一级供水站之间的最短距离

通过对欧式距离的排序找到中心供水站与一级供水站之间的最短距离的一级供水站为 V10，其坐标为（36，25），欧氏距离为：11.6619，

详见表 3，图 20

表 3：中心供水站到一级供水站之间的最短距离点的坐标

序号	类型	X 坐标	Y 坐标		序号	类型	X 坐标	Y 坐标
0	A	26	31	→	10	V10	36	25

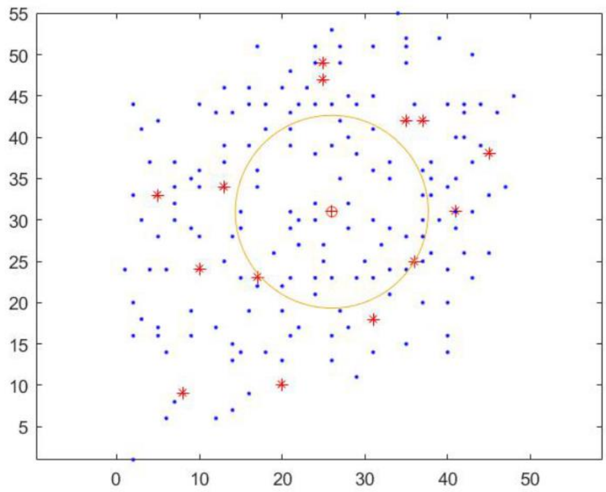

图 20 中心供水站和一级供水站之间的最短距离

2.2.5.2 找一级供水站与二级供水站之间的最短距离

通过对欧式距离的排序找到一级供水站与二级供水站之间的最短距离的路径为：V5→P125，欧氏距离为 1；V10→P135，欧式距离为 1；V14→P135，欧式距离为 1；V11→P139 欧氏距离为 0

详见表 4，图 21

表 4：一级供水站到二级供水站之间的最短距离点的坐标

序号	类型	X 坐标	Y 坐标		序号	类型	X 坐标	Y 坐标
5	V5	17	23	→	135	P123	17	22
10	V10	36	25	→	145	P133	37	25
11	V11	41	31	→	149	P137	41	31
14	V14	25	49	→	84	P72	24	49

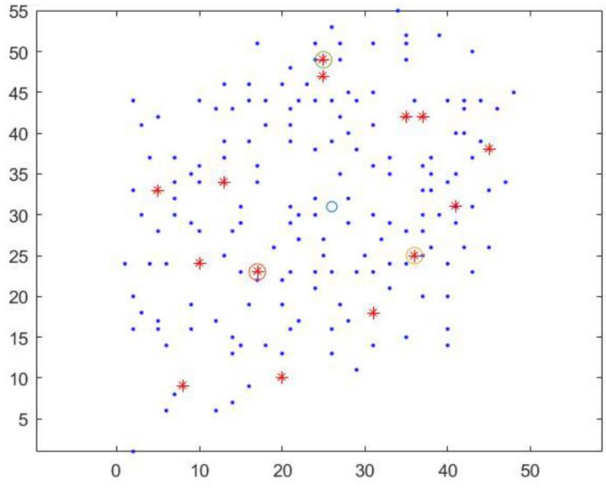

图 21 一级供水站到二级供水站之间的最短距离

2.3 结果

选择中心供水站向一级供水站是由 A→V10,由一级供水站向二级供水站是 V5→P123 或者 V10→P133 或者 V14→P72 或者 V11→P137 四选一即可。

2.3.1 路径 V5→P123（最终路径）

$\min(d) = 11.6619$ 公里；$\min(R_c) = 1$ 公里

$S_1 = 113.0131$ 公里；$S_2 = 417.7817$ 公里

$\min S = \min(d) + \min(R_c) + S_1 + S_2$

$S = 543.4567$ 公里，详见图 22

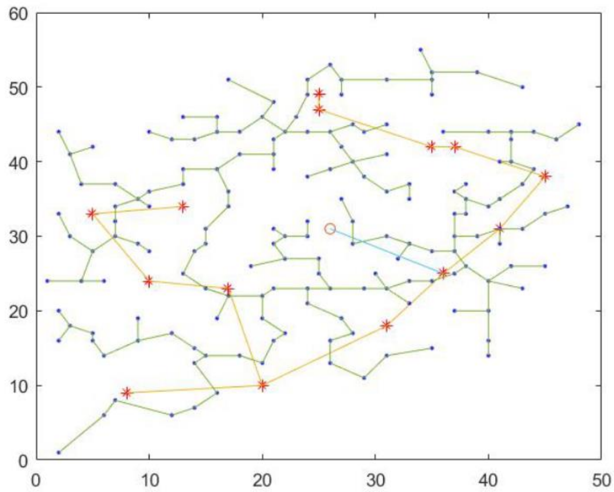

图 22 最终铺设路线（V5→P123）

2.3.2 路径 V10→P133

$\min(d) = 11.6619$ 公里；$\min(R_c) = 1$ 公里

$S_1 = 113.0131$ 公里；$S_2 = 417.7817$ 公里

$\min S = \min(d) + \min(R_c) + S_1 + S_2$

$S = 543.4567$ 公里，详见图 23

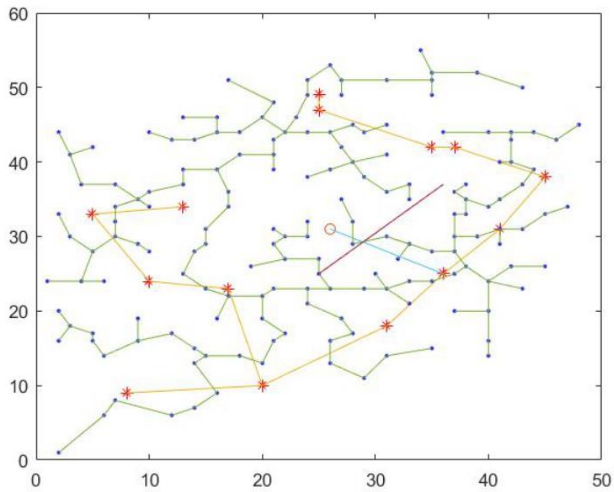

图 23 总铺设路线（V10→P133）

2.3.3 路径 V14→P72

$\min(d) = 11.6619$ 公里；$\min(R_c) = 1$ 公里

$S_1 = 113.0131$ 公里；$S_2 = 417.7817$ 公里

$\min S = \min(d) + \min(R_c) + S_1 + S_2$

$S = 543.4567$ 公里，详见图 24

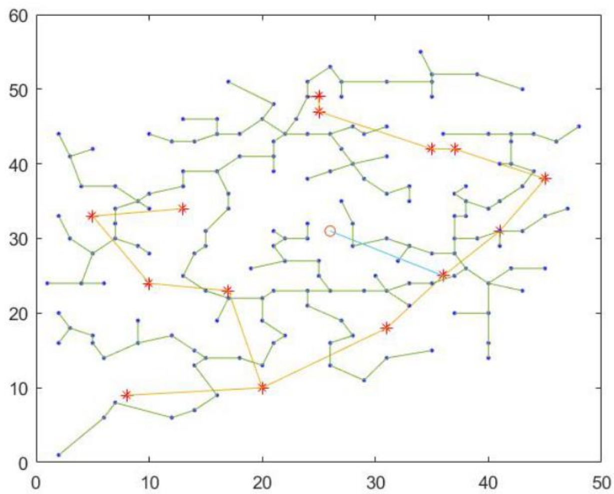

图 24 总铺设路线（V14→P72）

2.3.4 路径 V11→P137

$\min(d) = 11.6619$ 公里；$\min(R_c) = 0$ 公里

$S_1 = 113.0131$ 公里；$S_2 = 417.7817$ 公里

$\min S = \min(d) + \min(R_c) + S_1 + S_2$

$S = 542.4567$ 公里，详见图 25

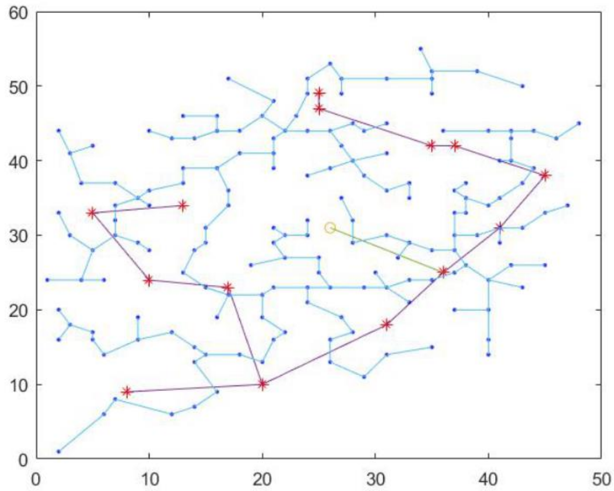

图 25 总铺设路线（V11→P137）

3. 问题三

3.1 模型建立

该配置下最终铺设管道的总里程数为Z，将所有一级供水站放入一个集合V中，即$V=\{V_1,V_2,\ldots,V_{12},V_{12+1},V_{12+2},\ldots,V_{12+x}\}$，每个一级供水站相应的与总供水站$A$的距离放入集合$D$中，即$D=\{d_1,d_2,\ldots,d_{12},d_{12+1},\ldots,d_{12+x}\}$，即$\boldsymbol{V1}$距离总供水站$A$的距离为$\boldsymbol{d1}$，$\boldsymbol{V2}$距离总供水站$A$的距离为$\boldsymbol{d2}$，以此类推。

注：这里的$\boldsymbol{V1 \sim V2}$并不代表具体的供水站坐标，仅为一级供水站数量的一个数学表达

将升级最少数量的供水站与铺设最短路的管道看为两个条件，根据递归的思想，选择“升级最少数量的供水站”作为优先考虑的条件，则为了升级最少数量的供水站，应当将已有的一级供水站进行最大化的利用。

(1) 在已有的 12 个一级供水站中，选择距离总供水站 A 最近的那个一级供水站：$d_1 = \min(\sqrt{(X-X_i)^2+(Y-Y_i)^2})$，第一次选择结束后，将该最短距离对应的供水站$\boldsymbol{V1}$从集合$V$中取出，此时集合$V$内的元素如下：$V=\{V_2,\ldots,V_{12},V_{12+1},V_{12+2},\ldots,V_{12+x}\}$

(2) 在问题一中，求得了任意一级供水站相应距离最近的另一个一级供水站的欧氏距离$\boldsymbol{Ra}$，故以$\boldsymbol{V1}$为圆心，$\boldsymbol{Ra}$为半径画圆，并以$\boldsymbol{V1}$、其对应的最短距离的一级供水站和该圆范围内的二级供水站建立最小生成树，设$\boldsymbol{S1}$为从$\boldsymbol{V1}$这个一级供水站管道输送的总里程，此

时可设立目标函数如下：$\min S_1 = \sum R_b + \sum R_c$，（$S_1 \le 40$）,同问题一一样，这里的$\boldsymbol{Rb}$与$\boldsymbol{Rc}$通过 MATLAB 计算出最优距离，最终得到一个最小生成树。

注：与第一问建立最小生成树的过程不同，在该圆范围内的点生成最小树的过程中，要优先考虑二级供水站的连接（满足总里程数小于 40 公里的前提下），即：在可用的 40 公里路程下，优先将该圆范围内的二级供水站通上水，若二级供水站全部通水后，可用的 40 公里里程数仍有剩余，则考虑去连接$\boldsymbol{V1}$最近距离的那个一级供水站，连接方法依然使用最小生成树，如果$\boldsymbol{V1}$可用的 40 公里里程数不足以将该圆范围内的所有二级供水站连通或是二级供水站全部连通后剩余的里程数不足以连接新的一级供水站，则此时不考虑连接新的一级供水站。

3.2 模型的分析

将一级供水站$\boldsymbol{V1}$可用的 40 公里管道铺设完成后，将会出现以下两种情况：

3.2.1 情况一

在可用的 40 公里内，该圆范围内的二级供水站被全部连通，且剩余里程数足够连接新的一级供水站$V2$。

将该一级供水站作为新的圆点去画圆，且半径同样为$\boldsymbol{Ra}$，得到新的圆后，接着进行与上述步骤相同的计算。此时，$V = \{V_3,\dots,V_{12},V_{12+1},V_{12+2},\dots,V_{12+x}\}$,如果此次计算结果依然满足该情况，则继续运算，直至现有一级供水站全部被覆盖

3.2.2 情况二

在可用的40公里内,该圆范围内的二级供水站没有被全部连通或是刚好全部连通，此时不连接新的一级供水站。此时，集合V仍为，$V = \{V_2,\dots,V_{12},V_{12+1},V_{12+2},\dots,V_{12+x}\}$,然后重新从总供水站出发，在新的集合$V$中寻找距离总供水站最近的一级供水站，找到后仍然进行上述步骤。

3.3 综述模型

按照这样的计算步骤逐个递归，直到所有的一级供水站全部被利用完毕后，开始在剩余没有被连通的二级供水站中选择升级。由于选择建立的模型是由每个一级供水站为圆心，最近距离的其他供水站为半径建立小的生成树，因此在 12 个一级供水站全部可用供水里程用光后，剩余的未连通的二级供水站将位于一级供水站分布较少的部分，至此，便达到了尽量少升级供水站的目的（如果未被连通的二级供水站分散在整个地图的四处，则需要横跨整个地图去建立连接）。

升级点的确立依然秉持距离总供水站越近越好的原则，先在剩余二级供水站中寻找距离总供水站最近的点升级，然后连接周围剩余的二级供水站，如果可用的 40 公里连完后，依然有未连通的二级供水站，则再从其中寻找距离总供水站最近的二级供水站，直至所有的二级供水站都被连通。此时可求得最少的升级供水站的数量X，则

$$Z = \sum_{1 \le i \le 12+x} S_i$$

六、模型的优缺点及改进

1.1 优点

1.1.1 在对问题一分层的最小生成树模型中，将不同级别的供水站分层求最小生成树，思路新颖，简化了计算量，使得算法的运行效率得到提高；

1.1.2 对于问题二的单目标优化模型的计算得出要升级供水站的过程中，避开了穷举法的思路，省去了繁杂庞大的数据处理，使得解题思路更加明确和新颖；

1.1.3 对于问题三中求解最短总里程的过程中，采用了画圆确定限制输送里程的思路，简化了求解的步骤。

1.2 缺点

1.2.1 在对问题一的求解中，只考虑了分层后的最小生成树，没有对整体做最小生成树计算总里程来对比哪种方案的总里程会达到最优，结果可能存在不足；

1.2.2 在对问题二的求解中，找二级供水站的流程过于死板，片面的考虑了只从中心 A 点出发的一条路径，而事实上可以多方面思考此路线，检验是否存在同一中心点出发的多条路径可供选择与判断；

1.2.3 在对问题三的找二级供水站的策略上存在片面性，无法达到过程最优最简的目的。

1.3 改进

1.3.1 针对问题一，需要对比分层的供水站和不分层的供水站之间做出的总里程进行对比，选取最优方案；

1.3.2 针对问题二，找二级供水站时，在选二级供水站的时候要考虑到所有的二级供水站，而不是仅限于一级供水站最近的二级供水站中最远的二级供水站，这样选出的二级供水站会节省更多的Ⅱ型管道，在计算总里程的时候同样需要进行分层的最小生成树和不分层的最小生成树的总里程对比，得到最优方案；

1.3.3 针对问题三，找二级供水站的个数一级二级供水站的坐标要和问题二一样进行所有二级供水站的筛选，最后在计算总里程的时候也需要进行对比得出最优方案。

1.4 模型的推广

此模型可以用在对于类似图论问题模型的解决中；也可以用于解决不同级别的运输问题，比如电路输送、基站建设、燃气输送等；还可以解决资源扩散的问题，实现能源消耗少且效益最大化的目标。

七、参考文献

[1]何国雄，基于 MapReduce 的图聚类算法的研究与实现，21-22，2012。

[2]赵裕，最小生成树问题---Prim 算法与 Kruskal 算法实现（MATLAB 语言实现），https://www.cnblogs.com/zhaoyu1995/p/5055186.html，2020.6.25。

[3]不系舟红枫，自来水管道连接规划模型，1-6，https://ishare.iask.sina.com.cn/f/bvgiaTke64X.html，2020.6.25。

[4]mengdoc2015，自来水管道连接规划模型，http://www.doc88.com/p-3814512094238.html，2020.6.25。

[5]胡运权，运筹学基础及应用（第 5 版），哈尔滨：哈尔滨工业大学出版社，1985 年。

八、附件

附录 1 程序

程序 1：画三层供水站的无向图

LineUp.m

```
C = [X, Y];
C1 = [X1, Y1];
C2 = [X2, Y2];
D = triu(ones(1));
D1 = triu(ones(12));
D2 = triu(ones(168));
gplot(D, C);
hold on;
gplot(D1, C1);
hold on;
gplot(D2, C2);

figure(1);
gplot(D, C);
figure(2);
gplot(D1, C1);
figure(3);
gplot(D2, C2);
```

程序 2：MATLAB 工具箱求最小生成树

Graphminspantree.m

```
A = zeros(1);
A1 = zeros(12);
A2 = zeros(168);

A = sparse(A');
Tree = graphminspantree(A)

Z(Z~=0) = 1;
plot(X, Y, 'o');
hold on;
gplot(Z, C)
hold on

A1 = sparse(A1');
Tree1 = graphminspantree(A1)

Z1(Z1~=0) = 1;
plot(X1, Y1, '*r');
```

```
hold on;
gplot(Z1, C1)
hold on

A2 = sparse(A2');
Tree2 = graphminspantree(A2)

Z2(Z2~=0) = 1;
plot(X2,Y2,'.b');
hold on;
gplot(Z2, C2)
```

程序 3：Kruskal 算法

Kruskal.m

```
A = sparse(A');
A(A==0) = Inf;    B = sparse(12, 12);
link = graphallshortestpaths(B, 'directed', false);
while sum(sum(link)) == Inf
        ind = find(A==min(min(A)));
        [x, y] = ind2sub(size(A), ind);
        for i=1:length(x)
                if link(x(i), y(i))==Inf
                        B(x(i), y(i)) = A(x(i), y(i));
                        A(x(i), y(i))=Inf;?
                end
        end
        link = graphallshortestpaths(B, 'directed', false);
end
```

程序 4：中心水站到一级水站的最短距离程序

Round.m

```
plot(X, Y, 'o');
hold on;
plot(X1, Y1, '*r');
hold on;
plot(X2,Y2,'.b');
hold on;
Round(26, 31);
```

程序 5：总的最小生成树

MinTree.m

```
Z(Z~=0) = 1;
plot(X, Y, 'o');
hold on;
```

```
gplot(Z, C)
hold on

Z1(Z1~=0) = 1;
plot(X1, Y1, '*r');
hold on;
gplot(Z1, C1)
hold on

Z2(Z2~=0) = 1;
plot(X2,Y2,'.b');
hold on;
m = [26, 36];
n = [31, 25];
plot(m, n, '-');
gplot(Z2, C2)

plot(X, Y, 'o');
hold on;
plot(X1, Y1, '*r');
hold on;
plot(X2,Y2,'.b');
hold on;
m = [26, 36];
n = [31, 25];
plot(m, n, '-');
hold on;
Z(Z~=0) = 1;
plot(X, Y, 'o');
hold on;
gplot(Z, C)
hold on

Z1(Z1~=0) = 1;
plot(X1, Y1, '*r');
hold on;
gplot(Z1, C1)
hold on

Z2(Z2~=0) = 1;
plot(X2,Y2,'.b');
hold on;
m = [26, 36];
n = [31, 25];
```

```
plot(m, n, '-');
gplot(Z2, C2)
```

程序 6：求解的结果

anser.m

```
%V5->P125
ans1 = sum(Z1(:));
ans2 = sum(Z2(:));
R01 = min(DIST01(:));
R12 = min(DIST12(5,:));
ans = ans1 + ans2 + R01 + R12;

%V10->P135
ans1 = sum(Z1(:));
ans2 = sum(Z2(:));
R01 = min(DIST01(:));
R12 = min(DIST12(10,:));
ans = ans1+ ans2 + R01 + R12;

%V11->P139
ans1 = sum(Z1(:));
ans2 = sum(Z2(:));
R01 = min(DIST01(:));
R12 = min(DIST12(11,:));
ans = ans1 + ans2 + R01 + R12;

%V14->P72
ans1 = sum(Z1(:));
ans2 = sum(Z2(:));
R01 = min(DIST01(:));
R12 = min(DIST12(14,:));
ans = ans1 + ans2 + R01 + R12;
```

程序 7：更新后的散点图

ScatterDiagram.m

```
plot(X, Y, 'o');
hold on;
plot(X1, Y1, '*r');
hold on;
plot(X2,Y2,'.b');
hold on;
m = [26, 36];
n = [31, 25];
plot(m, n, '-');
```

```
hold on;
m1 = [17, 17];
n1 = [23, 22];
plot(m1, n1, '-');

m1 = [36, 25];
n1 = [37, 25];
plot(m1, n1, '-');

plot(X, Y, 'o');
hold on;
plot(X1, Y1, '*r');
hold on;
plot(X2,Y2,'.b');
Round(26, 31, 11.66190379);

plot(X, Y, 'o');
hold on;
plot(X1, Y1, '*r');
hold on;
plot(X2,Y2,'.b');
Round(17, 23, 1);
Round(36, 25, 1);
Round(41, 31, 0);
Round(25, 49, 1);
```

程序 8：上三角矩阵

gauss.m

```
function B=gauss(dist11,    dist22)
[m, n] = size(dist22);
aa = size(dist11, 2);
for i = 1:aa
        for j = 1:m
                for k = 1:n
                        if(dist22(j, k) == 999)
                                dist22(j, k) = dist11(1, i);
                                i = i+1;
                        end
                end
        end
end
B = dist22;
end
```

附录 2：程序结果

详见支撑文件附录 2

附录 3：数据文档

详见支撑材料附录 3