# 算法实验报告

一、凸多边形分割

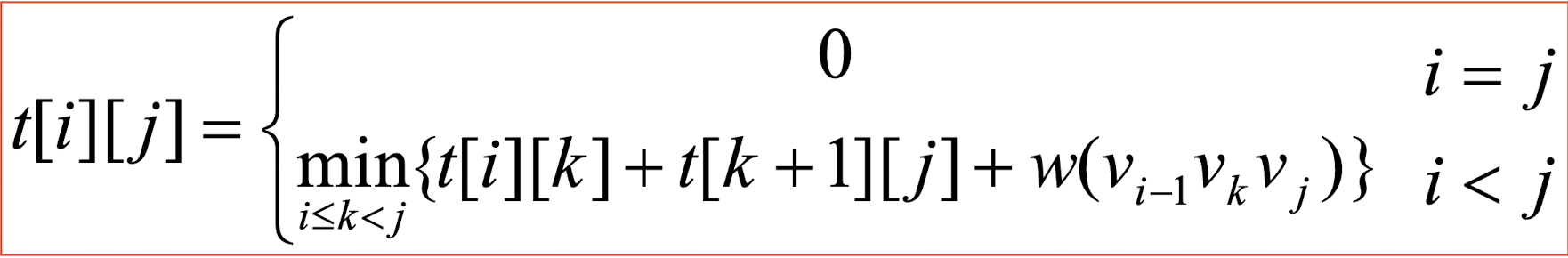
1、实验内容：

给定凸多边形P，以及定义在由多边形的边和弦组成的三角形上的权函数w(如欧氏距离)，要求:确定该凸多边形的三角剖分，使得即该三角剖分中诸三角形上权之和为最小——最优值。

2、算法实现思路：

矩阵连乘/表达式的完全加括号方式问题与三角剖分问题具有相似性，矩阵连乘表达式的完全加括号问题可以表示为一棵完全二叉树，称为表达式的语法树，矩阵连乘表达式的完全加括号问题也可以表示为一棵完全二叉树， 称为表达式的语法树。树根节点对应V0Vn，叶节点对应边Vi-1Vi，非叶结点对应剖分多边形的弦边。

凸多边形三角剖分具有最优子结构性质，所以根据剖分顶点，可以将原问题分解为子问题，最后自底向上求解。求出递归表达式：



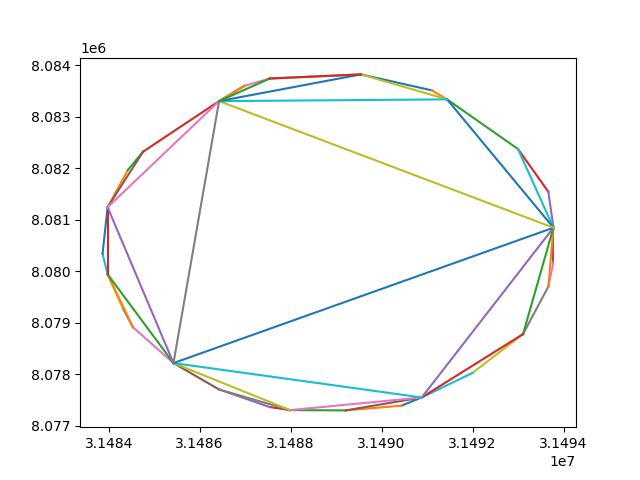
算法1:断点K的选取采用枚举方法，对每个可能的顶点进行计算求解得出最小值。

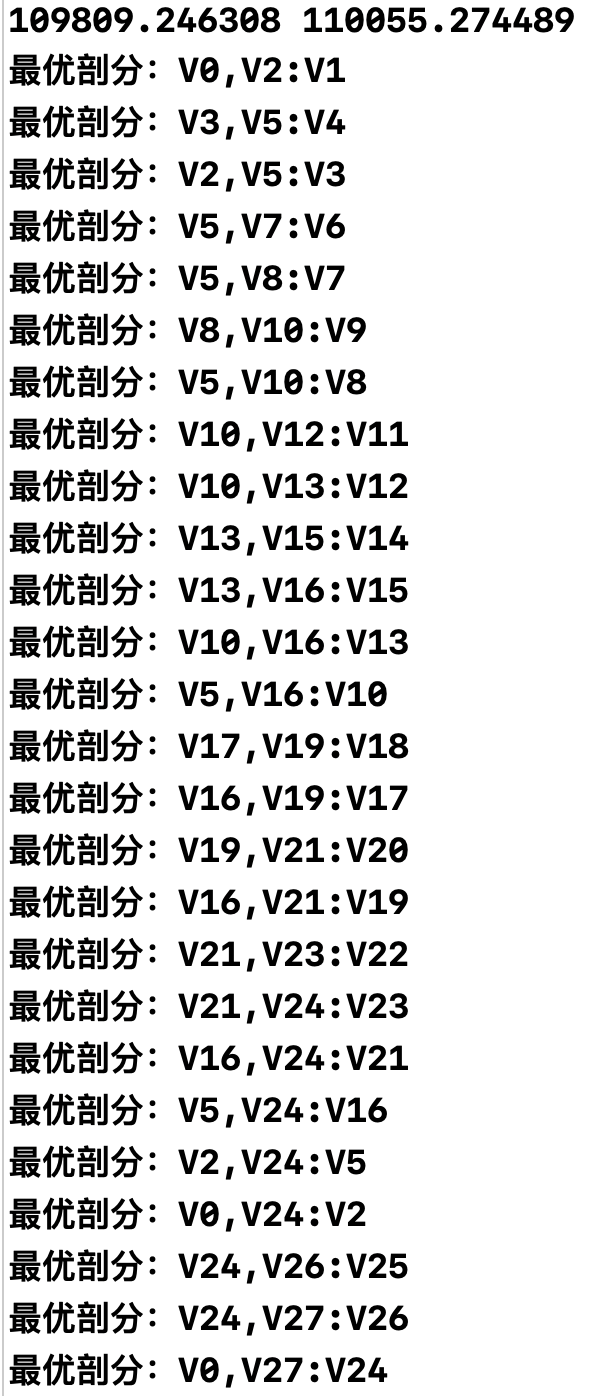
算法2:断点K的选取从固定位置进行选取，比较这K个断点的目标值，从中选取最优断点。我们组选用了i+1，j-1，i+j/2这三个位置作为断点。

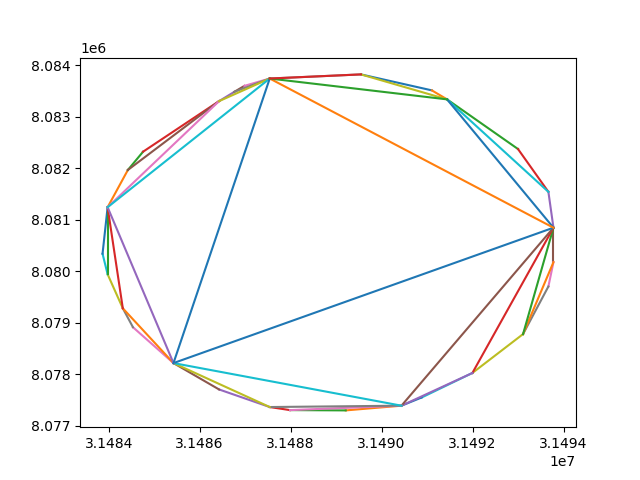
3、运行结果：

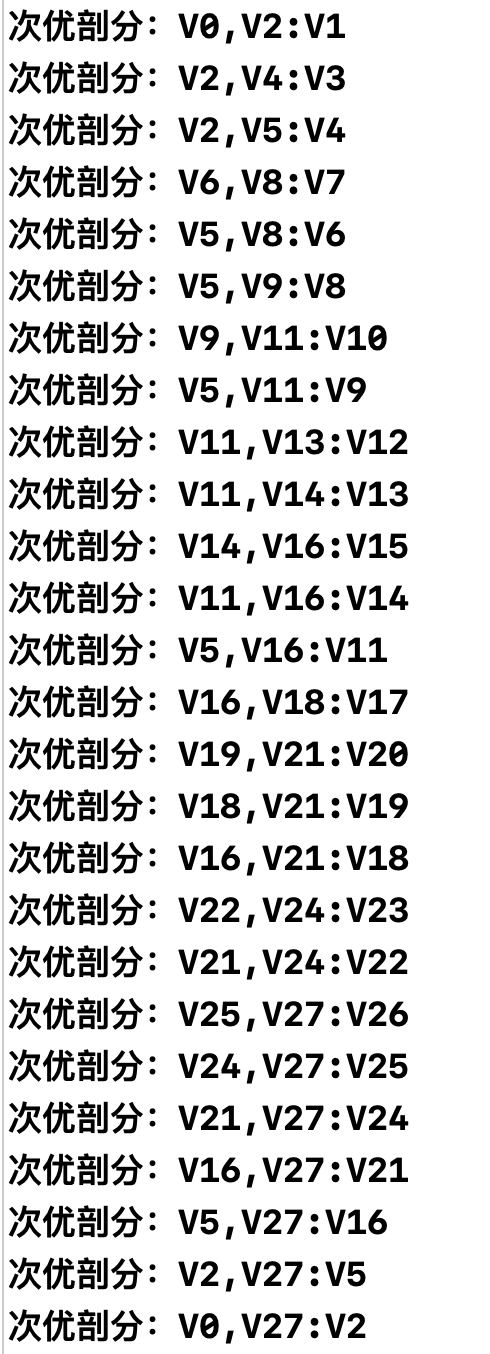
算法一：

29个基站运行结果：



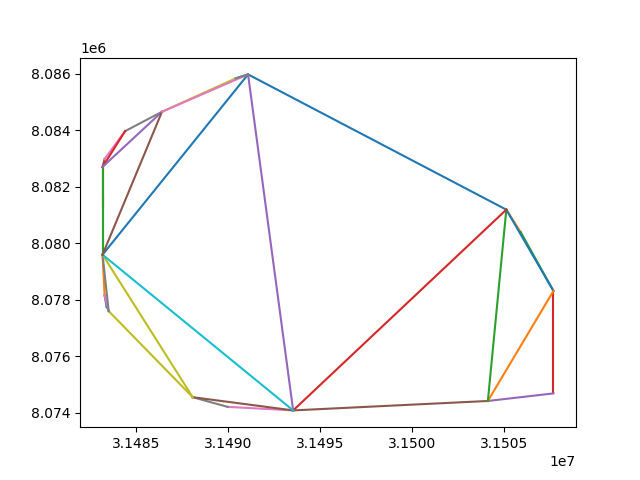


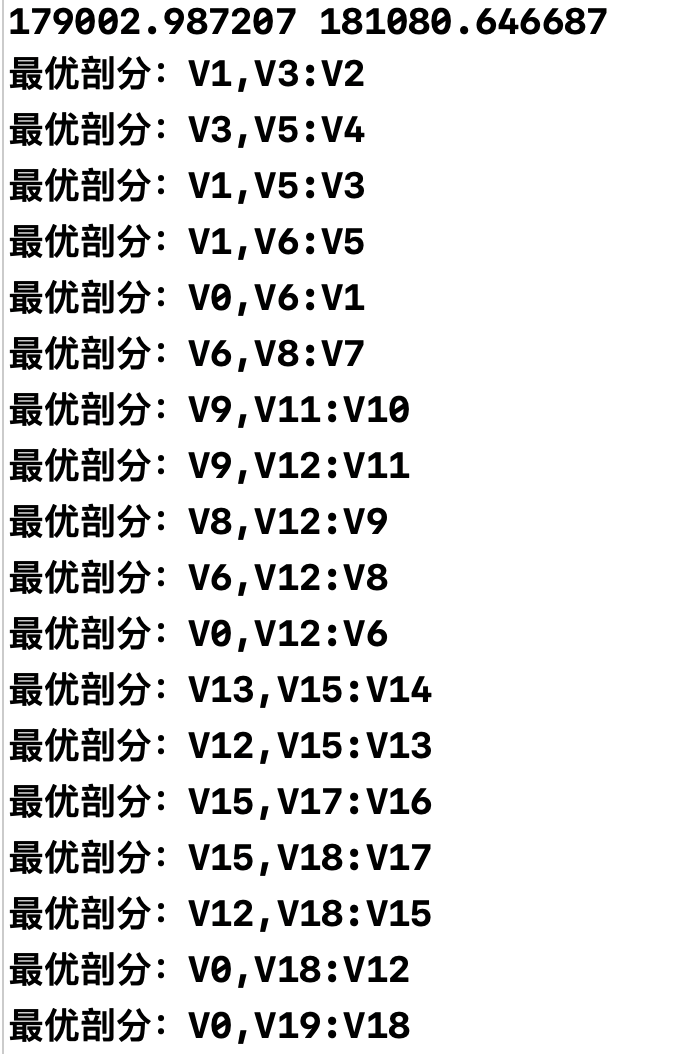




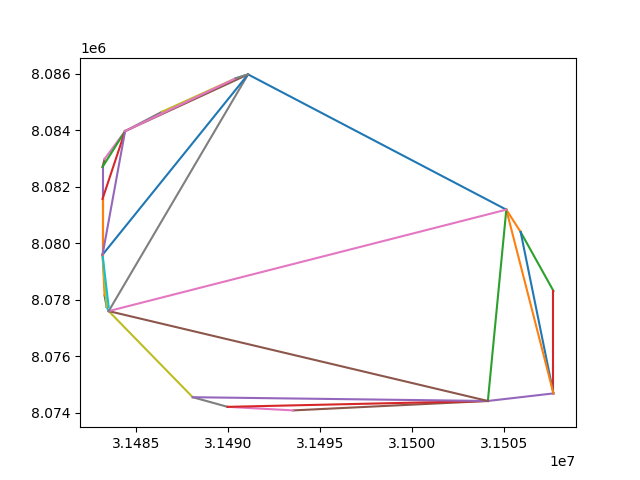
21个基站结果

最优：





次优：

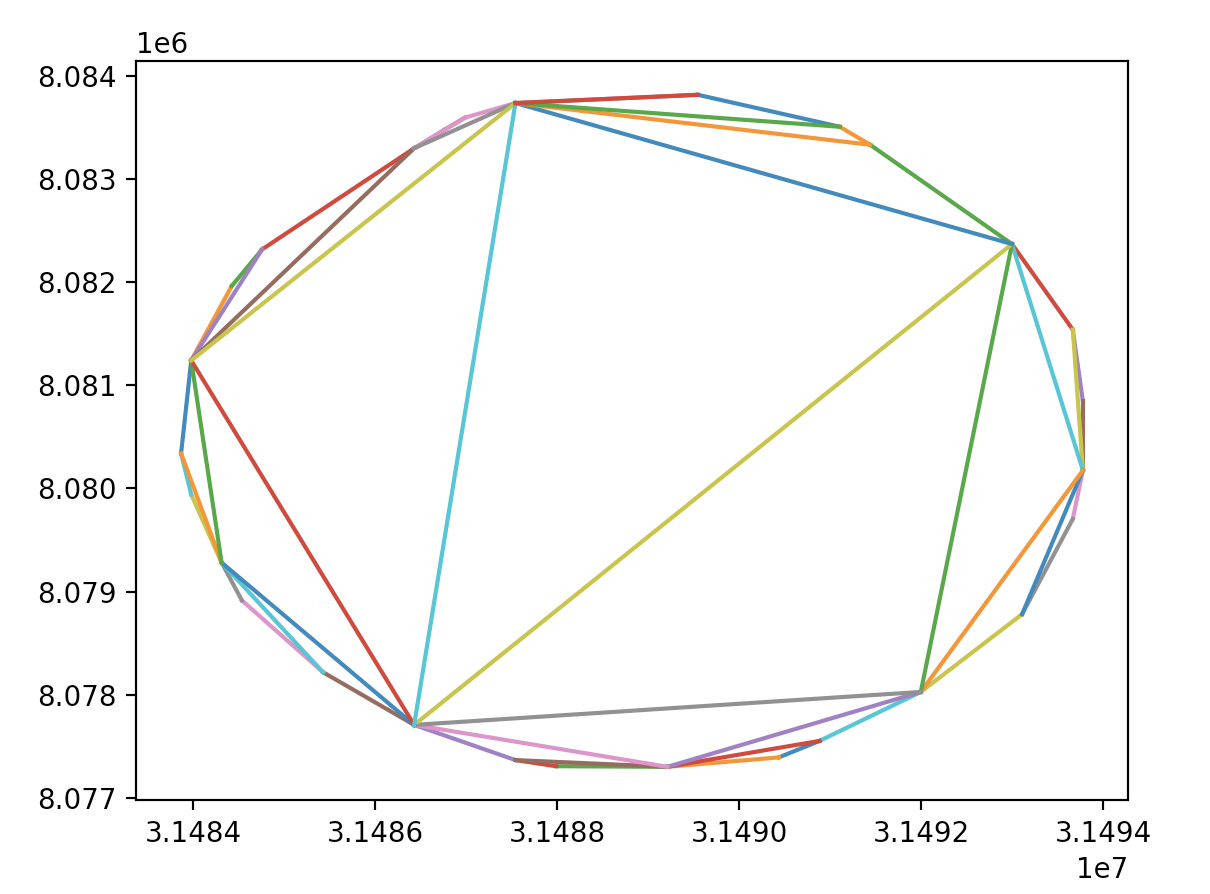




算法二：

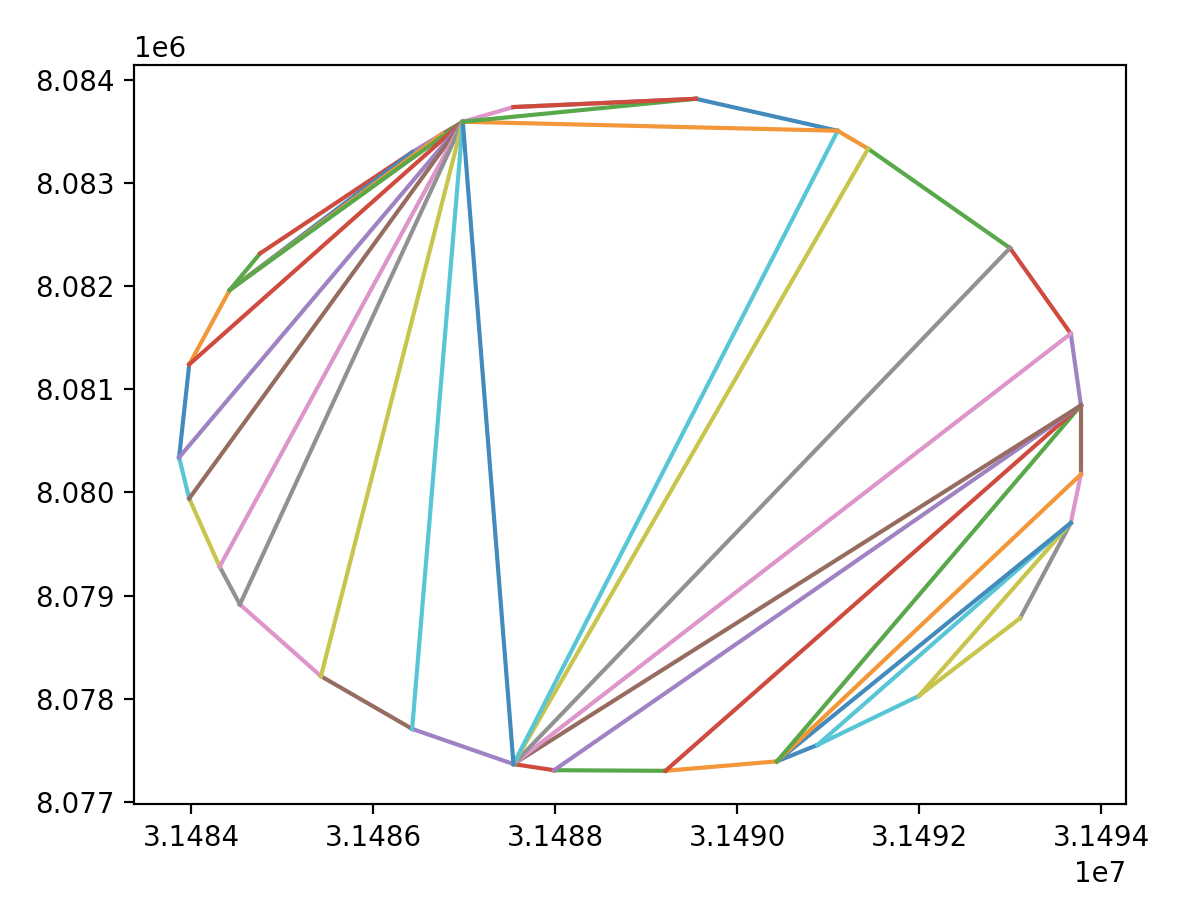
29个基站：

最优：



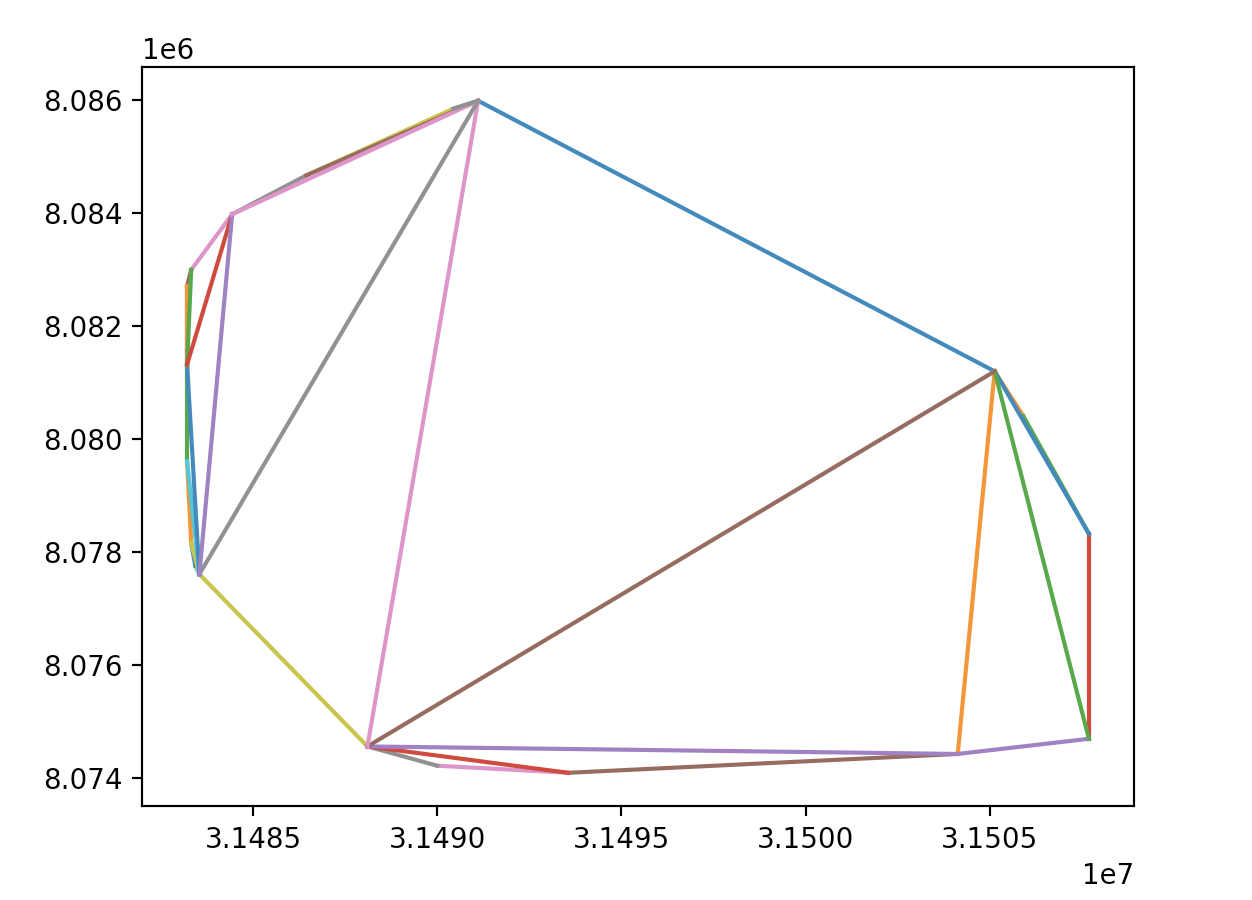


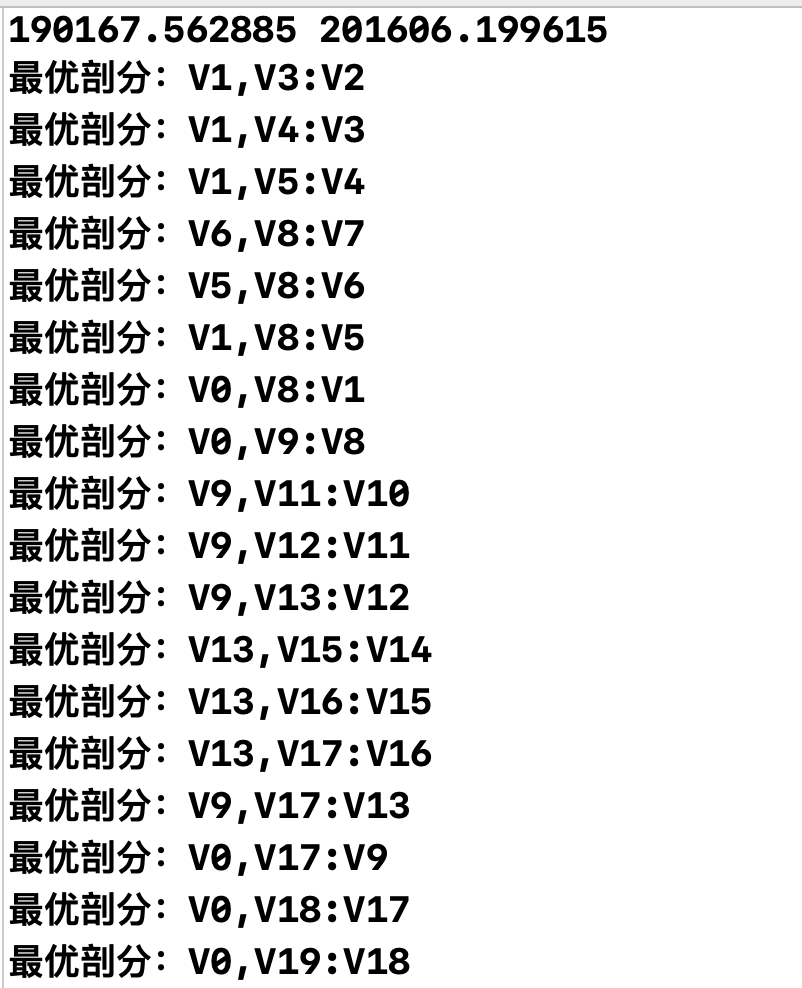
次优：



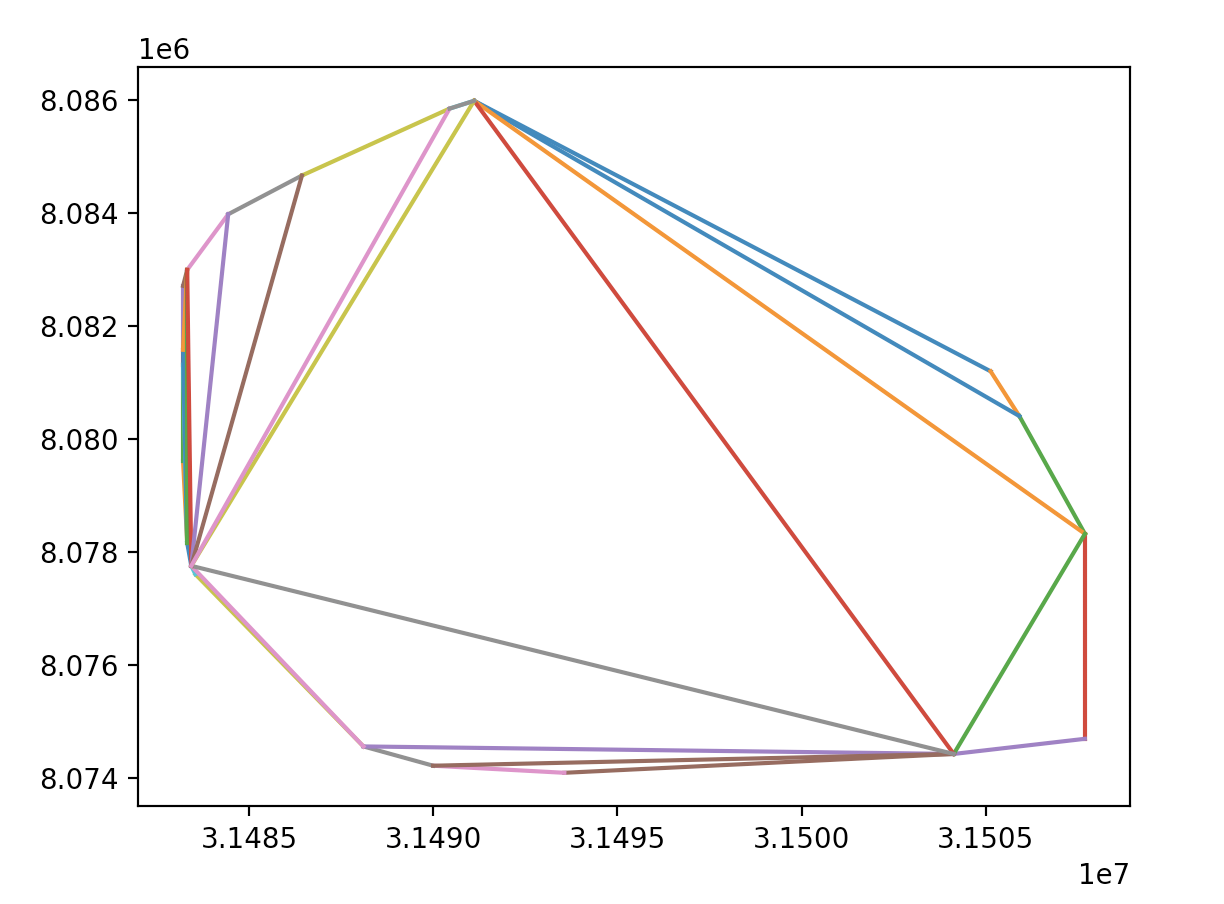


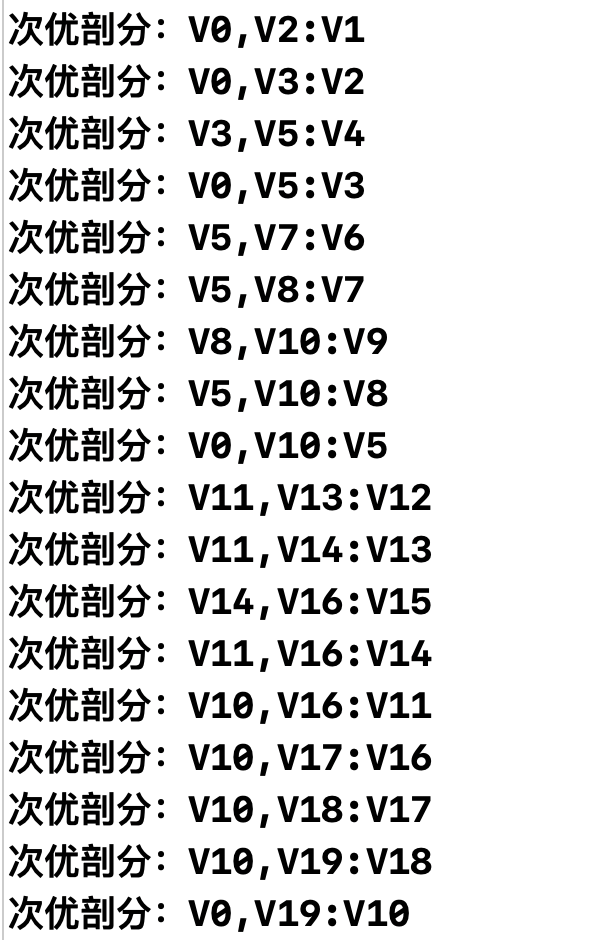
21个基站：





次优：





4、算法时间空间复杂度分析

算法一：

因为进行全遍历时间复杂度为o(n^3)，空间复杂度为o(n^2)，时间复杂性较高o(n^3)，所以当n较大时，算法运行时间较长。

算法二：

只选择三个固定位置，作为断点K的备选，时间复杂度为o(n^3)，空间复杂度为o(n^2)。

5、算法改进：

对于算法二，选取固定位置的点，对于贪心性质有很大的局限性，可以采用随机数的方式在可能断点中随机选取三个点，进行比较计算求出最优划分。

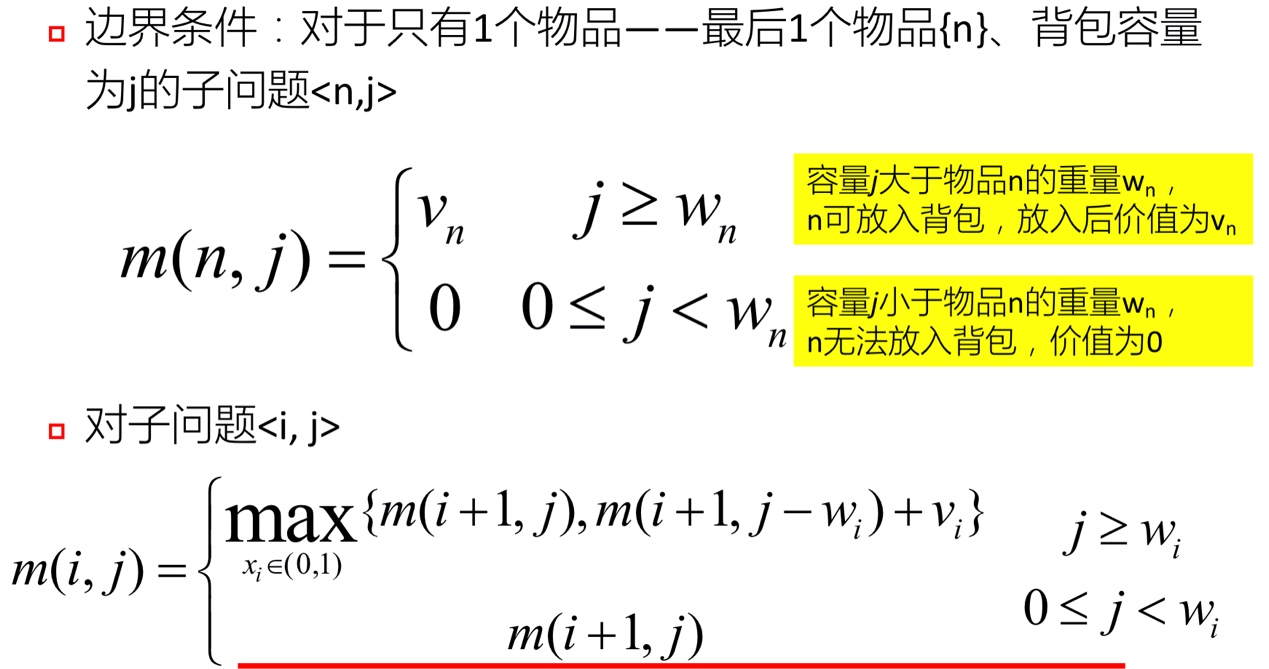
二、0-1背包问题：

1、问题描述：

给定n种物品{1, 2, 3, ...,n}和一背包。物品i的重量是wi，其价值为vi，背包的容量为C问:应如何选择装入背包的物品，使得装入背包中物品在其总重量不超过背包容量C的前提下，背包中物品的总价值最大?

2、算法实现思路：

可证明背包问题符合最优子结构性质，建立计算m(i, j)的递归式



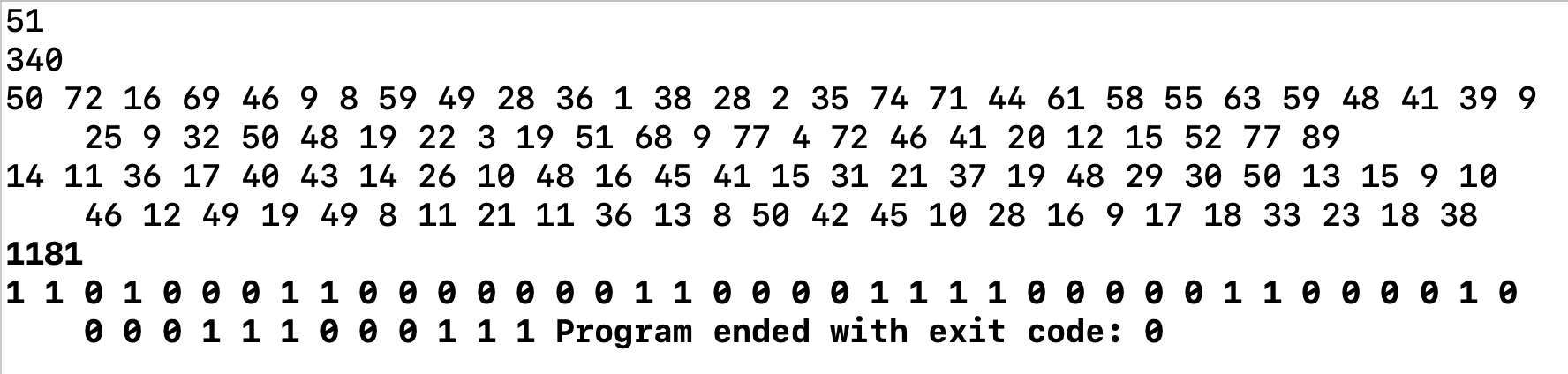
如果第i种物品重量wi大于容量j，第i种物品无法放入，可以不考虑，xi=0, 向后考虑其它物品，子问题<i, j>缩减为规模更 小的<i+1,j>

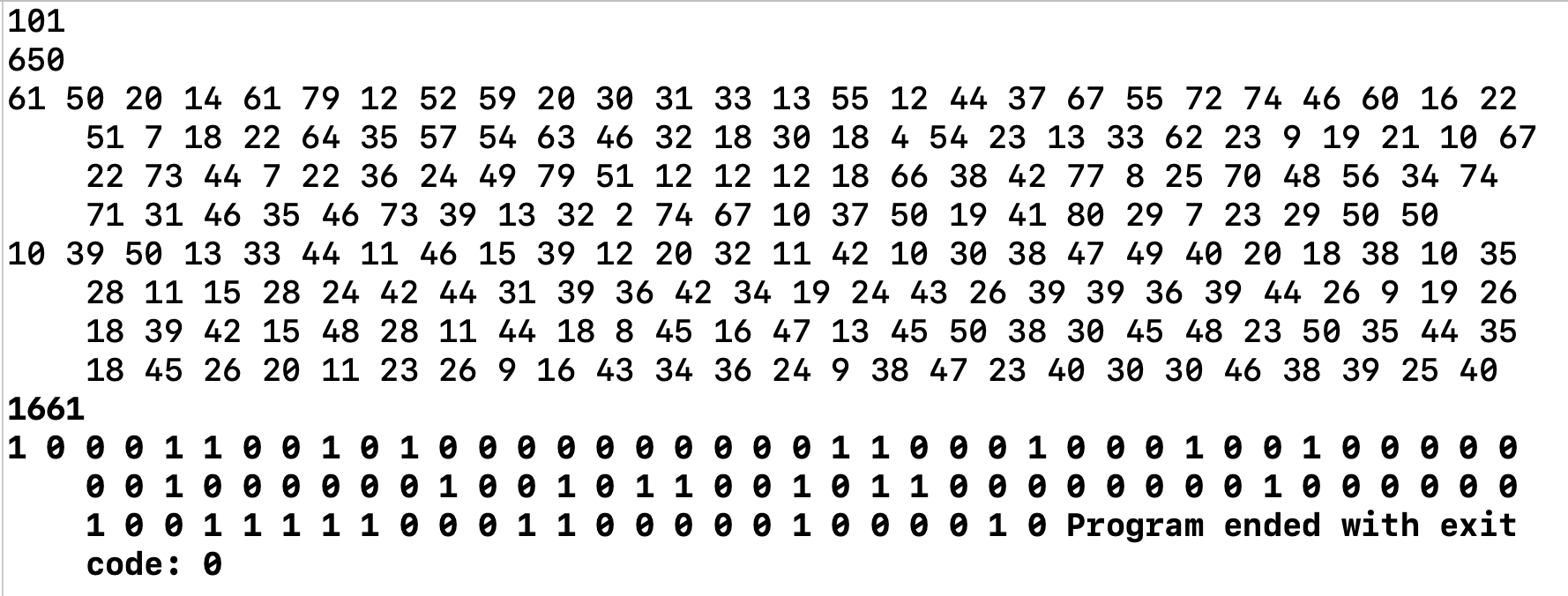
ii)如果第i种物品重量wi小于容量j，第i种物品可以考虑放入，分为2种情况

-情况1:将第i种物品放入，对总价值的贡献为vi，剩余容量为j-wi ，子问题<i, j>缩减为规模更小的<i+1, j-wi>

-情况2:第i种物品不放入，背包容量仍为j ，子问题<i, j>缩减为规模更小的<i+1,j >

3、运行结果：





4、算法复杂度分析

时间复杂度为o(nc)，空间复杂度为o(nc)。

5、算法改进

若不需知道哪些物品被装进了背包，只需要求最大价值，空间复杂度可下降到o(c)。