《算法分析—第五章实验》

实验报告

姓　　名　　袁洁　　胡敏臻

学　　号　　2019211426　2019211424

班　　级　　2019211307　2019211307

专　　业　　计算机科学与技术

1 实验内容

1.1 旅行商问题

针对昆明 LTE 网络，选取部分基站，计算基站间的距离，在部分基站间引入边，得到

1） n=15 个基站顶点组成的图，以图中基站顶点作为城市，去除 7 个位置相邻的基站：2,15; 14,1,4, 6,18

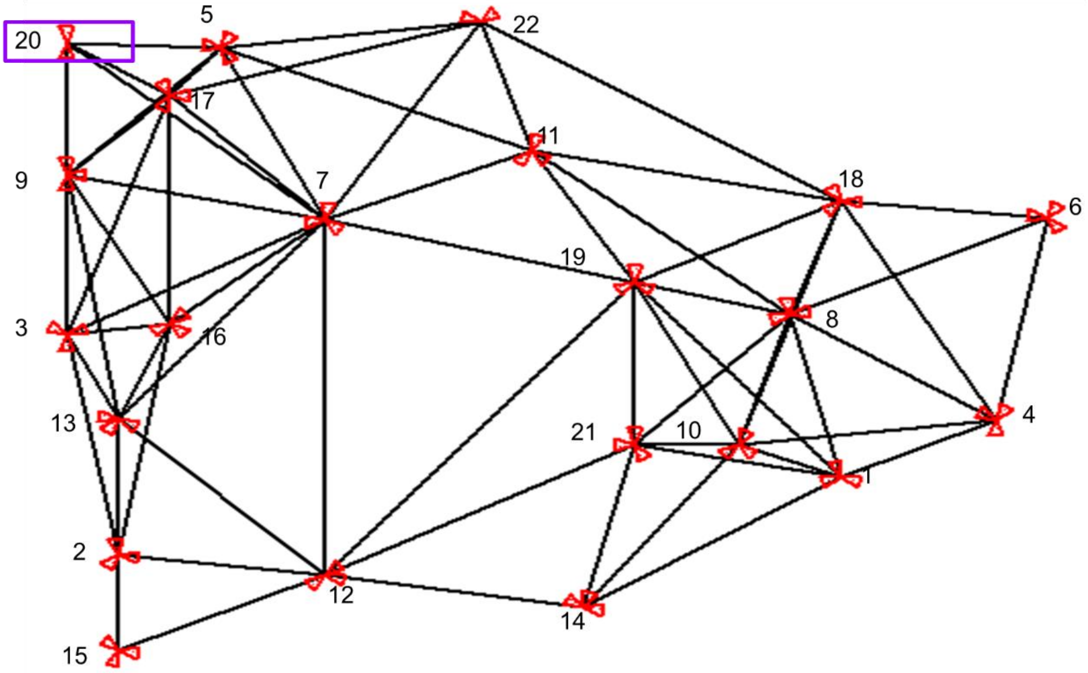

图 1：15 个基站组成的无向图

2） n=20 个基站顶点组成的图，以图中基站顶点作为城市，去除 2 个位置相邻的基站：4，6

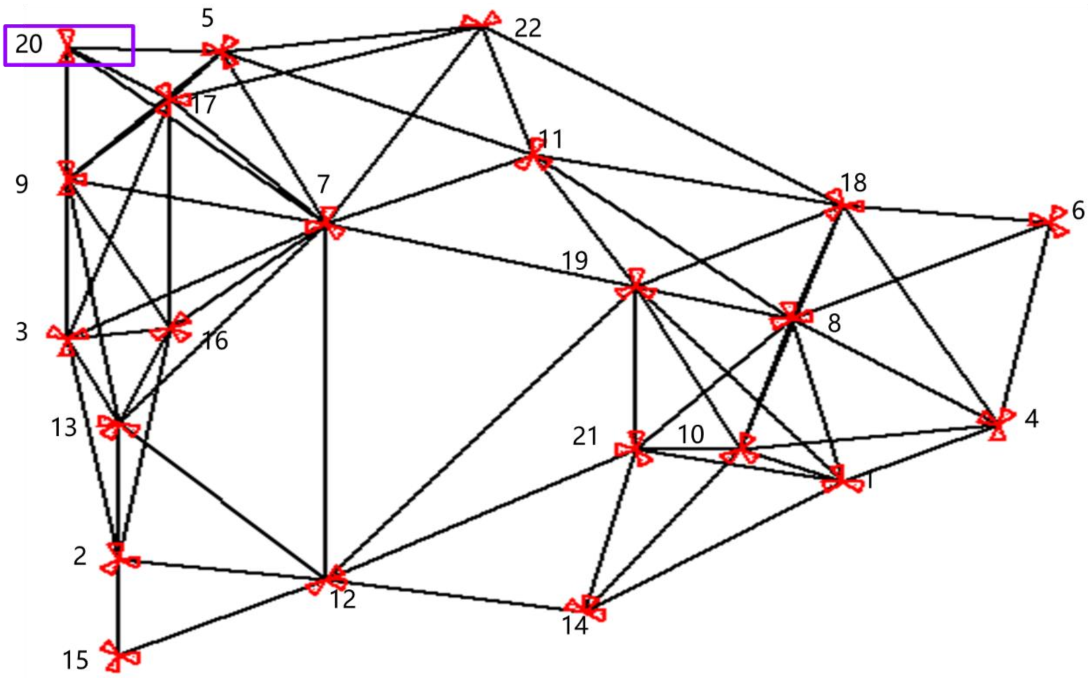

图 2：20 个基站组成的无向图

3） n=22 的基站图

要求：

1. 修改完善程序，统计搜索过程中扫描过的搜索树结点总数 L

2.修改完善程序，记录程序运行时间 T

3. 针对图 1、图 2，输出采用回溯法、分支限界法得到的

1）从起始城市出发的最短旅行路径

2）路径总长度

3）扫描过的搜索树结点总数 L

4）程序运行时间 T

结果记录在表格中

1.1.2 回溯法

解向量 n 维，表示旅行商依次经过的城市，通过递归函数 backtrack 遍历排列数，依次搜索旅行商可能经过的路线，通过两个城市间有路径，以及当前花费小于已求出的最小花费进行剪枝。最后得出结果。

1.1.3 分支限界法

解向量 n 维，扩展内部节点，通过计算当前部分解可以得到的最小花费，与分支限界上界比较进行剪枝，并通过求出的解来更新上界，若符合条件将其加入队列，当队列中没有节点结束循环。

1.2 图的 m 着色问题

参照教科书，编程实现回溯法。针对图 1、图 3，给定颜色总数 m 后，运行程序，为图中各个基站结点，分配颜色。

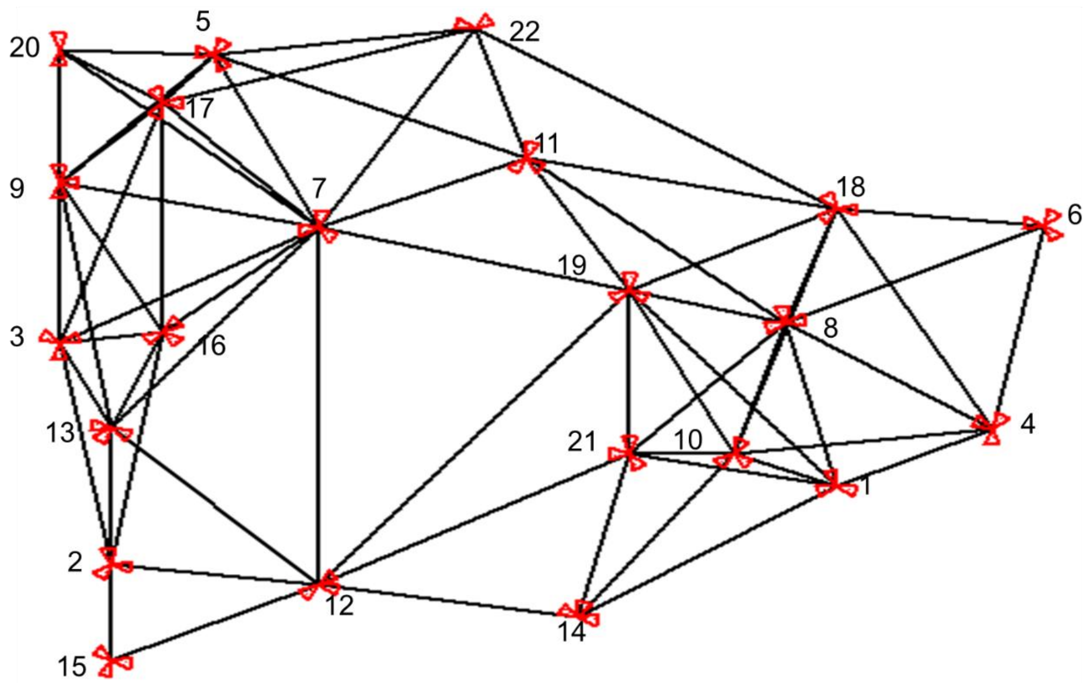

图 1，m≤m0=22 22 个基站组成的无向图

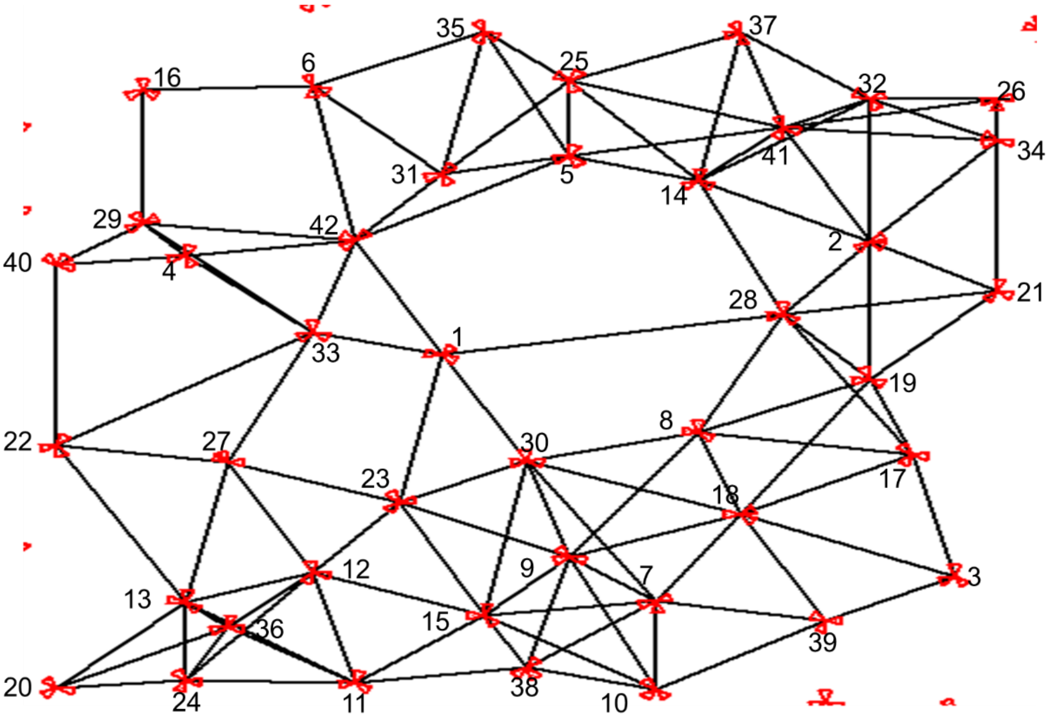

图 3，m≤m0=42 42 个基站组成的无向图

要求：

1. 修改完善程序，采用尽可能少的 m≤m0 种颜色，为图中各个结点着色
2. 修改完善程序，统计搜索过程中扫描过的搜索树的结点总数 L
3. 记录运行时间 T
4. 输出图中各个顶点的着色方案、用到的颜色总数 m、搜索过的结点总数 L、算法运行时间 T，结果记录在表中

2 旅行商问题

2.1 问题描述

n 个城市组成的带权无向图 G=(V,E)，顶点 V 对应于城市，边 E 对应于城市间路径，要求找出一条最短旅行线路，每个城市只经历一次

- 旅行回路$\{v_1,v_2,v_3,\ldots,v_n,v_1\}$总费用极小化

$$\min\{\underbrace{\sum_{i=1}^{n-1} w(v_i,v_{i+1})}_{\text{从起点出发，遍历其它n-1个城市，到达第n个城市}} + \underbrace{w(v_n,v_1)}_{\text{从第n个城市回到起点城市}}\}$$

图中，权$w(v_i,v_j)$（或w(i, j)）表示城市i与j间的直接距离，$w(v_i,v_j)=\infty$表示城市i与j间无直接路径。

将图中n个顶点编号为1, 2,…, n;

以顶点1为起点，旅行回路描述为1, x_2, .., x_n, 1;

其中，x_2, .., x_n为顶点2, 3, 4,…, n的1个排列。

因此，**解空间大小为(n-1)!**

2.2 回溯法

剪枝条件/约束 1：

如果当前正在考虑的顶点 j 与当前已经走过的部分路径中)末端结点 i（代表没有走过的其它城市）没有边相连，即 w[i, j]= ∞，则放弃搜索 j 所在分支

剪枝/约束条件 2：

令到第 i 层结点为止，构造的部分解路径为<1, x[2], x[3],…, x[i-1], x[i], ?, ? , ?>，如果 B(i) ≥ bestw——x[i]没有希望达到更优的路径，则停止搜索 x[i]分支及其下面的层，其中 bestw 代表到目前为止，在前面的搜索中，从其它已经搜索过的路径中，找到的最佳完整回路的权和（总长度）

2.2 分支限界法

利用界限，进行剪枝

对 TSP 最小化问题，在问题求解过程中，如果 1 个部分解的目标函数 dist 下界 lb，超出问题解的上界，则该部分解对应了死结点，可剪枝

1. 采用贪心法，计算问题上界*up*——用于后续结点剪枝
   根据目标函数公式，计算根结点/问题下界***down***（用处？）
2. 将活结点表ANT初始化为空
3. for（i=1; i <= n; i++) x[i]=0;　　/*解向量初始化
4. k=1; x[1]=1;　　/*第一步，从顶点1出发
5. while (k>=1)　　/*k: 遍历步骤，第k步，x[k]:第k步走的城市
   - 5.1 i=k+1;　　/* 第k步已选定城市，考虑第k+1步的城市
   - 5.2 x[i]=1;　　/*第k+1步，按照城市序号,首先选择第1个城市1
   - 5.3 while (x[i] <= n)　　/*宽度优先，生成x[k]的子结点x[i]=1,2,..,n
     - 5.3.1 如果路径上城市顶点不重复，则：　/*x[i]满足硬约束
       - 5.3.1.1 计算x[i]的下界lb
       - 5.3.1.2 if (lb up),　　/*当前结点无需剪枝，可进一步扩展
         将路径上的顶点和lb值存入活结点表ANT
     - 5.3.2 x[i] = x[i] + 1　　/*依次生成x[i]=1的各个兄弟结点

5.4 如果

(1) i==n, //到达叶结点，选定了最后一步所走城市

(2) 并且，该叶子结点的目标函数值与表ANT中所有叶结点、非叶结点的评估值lb相比，是最小的

，则将该叶结点对应的最优解输出，算法结束!!!

5.5 否则，若i==n /*该结点目标函数值lb并不是最小

该结点是叶结点，对应一条可能并非最优的完整回路，

，则：

从ANT中，找出具有最小lb值*minlb*的叶子结点，

(1) 令**up=*minlb***, **//更新问题上界，提高后续剪枝效率**

(2) 删除ANT表中目标函数值lb超出up的结点

/*利用更新后的上界，剪枝

5.6 k= 表ANT中*lb*最小的路径上的顶点个数

选*lb*最小的非叶结点（或叶结点），作为第k+1步的扩展结点，

e.g. 上例中Step5, 选结点3, k=2, 对应部分路径<1,3>, 下一步考虑第k+1=3步

2.3 实验结果

问题	求解算法	最短回路	路径总长度（单位：m）	搜索过的结点总数	程序运行时间（单位：s）
15个基站	回溯	20 9 7 16 3 13 12 21 10 8 19 11 22 5 17	5506.88	255112	0.045297
	分支	20 17 5 22 11 19 8 10 21 12 13 3 16 7	5506.88	3025	0.067702

	限界	9			
20个基站	回溯	20 9 7 16 3 13 2 15 12 14 21 10 1 8 18 19 11 22 5 17	6987.51	76206613	7.10864
	分支限界	20 17 5 22 11 19 18 8 1 10 21 14 12 15 2 13 3 16 7 9	6987.51	15163	0.255324
22个基	回溯	20 9 7 16 3 13 2 15 12 14 21 10 1 4 6 18 8 19 11 22 5 17	7690.8	486674749	46.7224
	分支限界	20 17 5 22 11 19 8 18 6 4 1 10 21 14 12 15 2 13 3 16 7 9	7690.8	6453	0.158245

3 图的 m 着色问题

3.1 问题描述

图顶点着色问题：

给定向连通图 G=(V, E)和 m 种同的颜色， 这些颜色为图 G 的各顶点着色，每个顶点着一种颜色

约束：相邻边不同着色

问题解空间

给定n个结点的图G=(V, E)，m种颜色，解向量

$$x[1:n]=(x_1, x_2, \dots , x_n)$$

, x[i]表示顶点i所着颜色, 顶点i的颜色取值范围为1, 2,…, m, 即$x[i] \in \{1, 2,\dots, m\}$

图的色数：
如果一个图最少需要 m 种颜色才能使图中每条边连接的 2 个顶点着不同颜色，m 为该图的色数

3.2 实验结果

问题	用到的颜色总数 m（色数）	搜索过的结点总数 L	程序运行的时间 T（单位：s）
22 个基站	5	215644	0.065
42 个基站	5	15407962835	6637.26

22 个基站着色方案：
0 0 1 1 1 0 0 2 2 3 3 2 3 1 1 4 3 4 1 4 4 2

42 个基站着色方案：
0 0 0 0 0 0 0 0 1 2 0 1 2 1 3 1 1 2 3 0 1 0 2 3 2 0 3 2 2 4 1 2 1 3 3 4 0 4 1 1 4 3

截图：
22 个基站

```
图的色数为5
搜索过的结点总数215644
递归所用时间0.065
```

42 个基站

```
图的色数为5
搜索过的结点总数15407962835
递归所用时间6637.26
```

4 时间、空间复杂度分析

4.1 旅行商问题

4.1.2 回溯法：

最坏：时间 O((n-1)!)

空间 O(n)

4.1.2 分支限界法：

时间：O(2^n*n^2)

空间：O((n-1)2^(n-2))

4.2 图的 m 着色问题

最坏：时间 O(nm^n)

空间 O(n)

5 实验总结

在本次实验中，每个任务我们都共同付出了较多的努力。每个人的贡献度为50%+50%。

本次实验的难点在于分支限界法，因为对分支限界的掌握不够熟练，我们花费了较多的时间与精力去解决分支限界问题。最后通过查阅各种资料解决了。本次实验跑代码的时间较长。在图着色问题中，因为要找到最小色数，逐步遍历各个结点的时候花了将近 2 个小时的时间。同时我们也采用了不同的回溯方案。在旅行商问题中我们采用的是递归回溯方案，在图着色问题中我们采用的是迭代递归的方案，这对于我们对回溯方案有了更深的了解。

改进思路：

1、图着色问题中在范围较大时所需时间比较长，可以考虑对算法做优化，因为不同颜色他们的地位是等价的，我们可以尝试确定选中的颜色顺序，不断深入，若找到可行解则可以视为解的上界，如果有更优解则继续更新。

2、在分支限界法中，本程序采用先进先出的队列，如果采用优先级队列，边历的节点会减少许多