禁忌搜索算法Tabu search

1. 实验目的

使用禁忌算法来解决TSP实际问题。

1. 实验内容

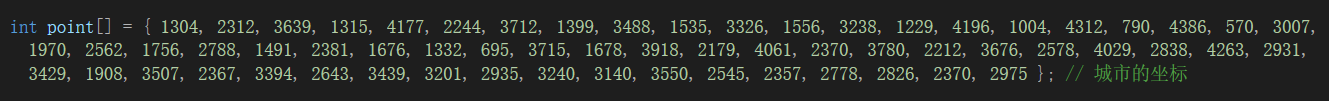
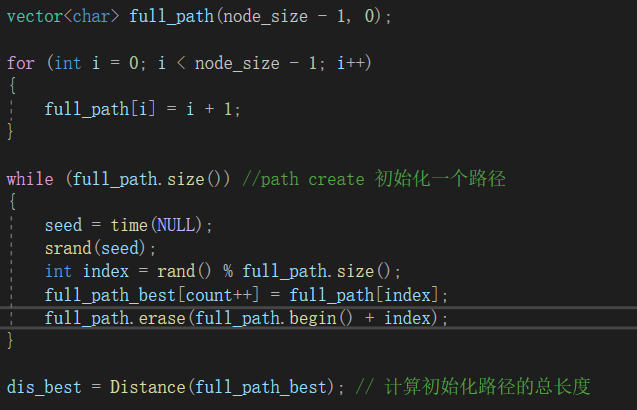
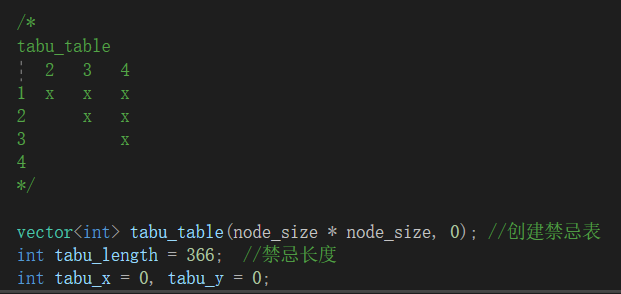
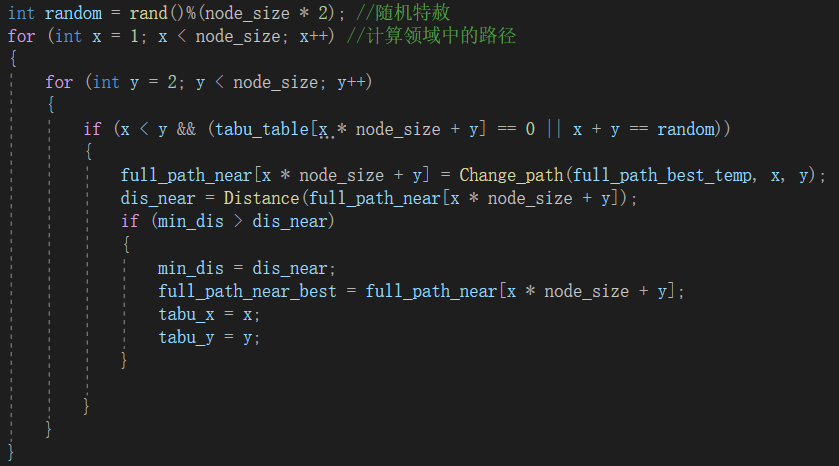
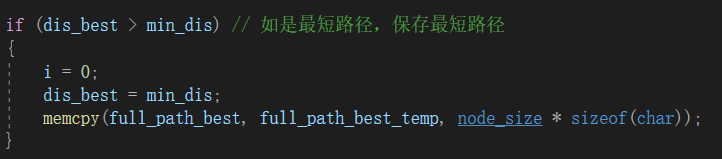
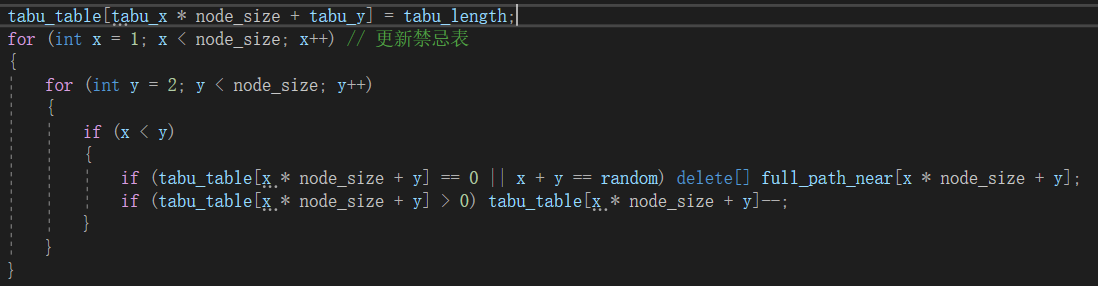
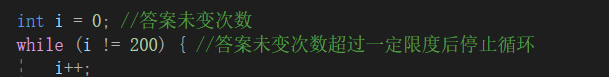
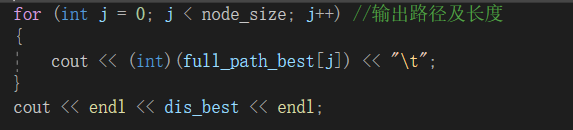
用禁忌搜素算法求解31个城市的TSP问题（从第一个城市出发最后再回来）

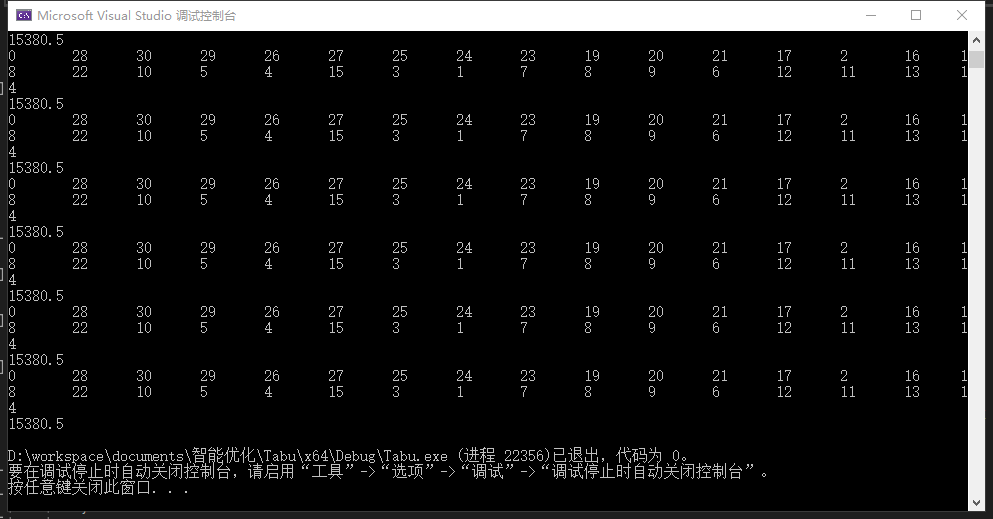
全国31个省会城市的坐标为[1304 2312; 3639 1315; 4177 2244; 3712 1399; 3488 1535; 3326 1556; 3238 1229; 4196 1004; 4312 790; 4386 570; 3007 1970; 2562 1756; 2788 1491; 2381 1676; 1332 695; 3715 1678; 3918 2179; 4061 2370; 3780 2212; 3676 2578; 4029 2838; 4263 2931; 3429 1908; 3507 2367; 3394 2643; 3439 3201; 2935 3240; 3140 3550; 2545 2357; 2778 2826; 2370 2975]

1. 实验环境

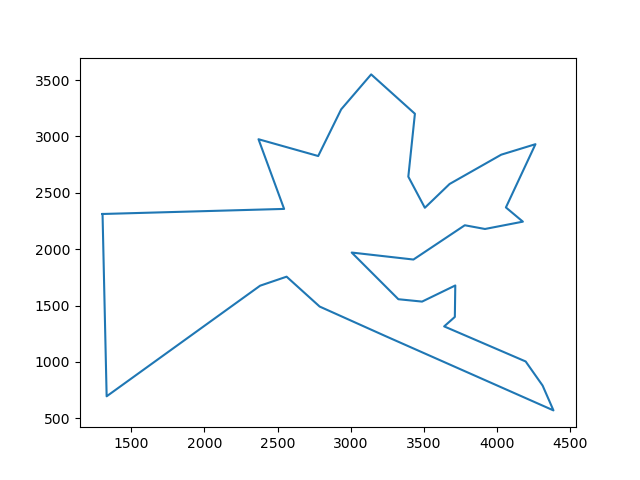
Visual Studio 2019

C++语言进行开发

1. 实验步骤
2. 首先初始化每个城市的坐标。
3. 随机初始化一个路径并计算该路径的长度。
4. 创建禁忌表并设定一个禁忌长度。
5. 每次随机设置特赦哪一个，然后计算2-opt领域中的路径长度，并选出一个路径最短的作为当前解。
6. 如果当前路径是最短路径，就进行保存。
7. 更新禁忌表。
8. 设置终止条件。
9. 最终输出答案。
10. 实验结果

由于该实验受初始值和禁忌表长度影响较大，如下是通过将禁忌长度设置为366的时候，得到的最好的结果。

下面是将该路程可视化后得到的结果。



可以大致看出没有交叉路线。

1. 实验总结

禁忌算法是一种启发式随机搜索算法，它从一个初始可行解出发，选择一系列的特定搜索方向作为试探，选择让目标函数值最优的移动。为了避免陷入局部最优解，TS搜索中采用了一种灵活的“记忆”技术，对已经进行的优化过程进行记录和选择，指导下一步的搜索方向，就是Tabu表的建立。Tabu表有几个很重要的参数，禁忌对象和禁忌长度，在本实验中我选择的禁忌对象是交换的位置，亦“解向量分量的变化”，禁忌长度选择的366，禁忌长度是进过多次实验后的结果，如果设计的长度太短，就难以跳出局部最优解，设计的太长又会难以收敛到最优解。标记已经解得的局部最优解或求解过程，并在进一步的迭代中避开这些局部最优解或求解过程。特赦原则我是每次随机选取一个解空间进行特赦。终止规则我是使用的目标控制原则，如果在一个给定步数内，当前最优值没有变化，终止计算。在该实验中受初始值和禁忌表长度影响较大。不同的初始值和禁忌长度都会造成不同的解，有时结果可能会有很大的差距。