遗传算法

1. 实验目的

使用遗传算法来解决TSP实际问题。

1. 实验内容

用遗传算法求解31个城市的TSP问题（从第一个城市出发最后再回来）

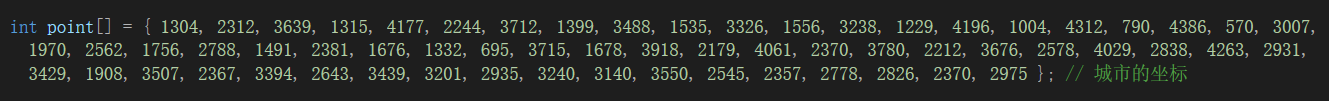
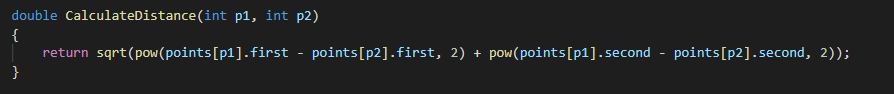
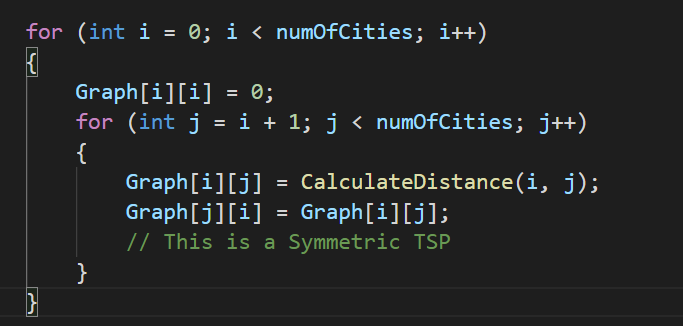
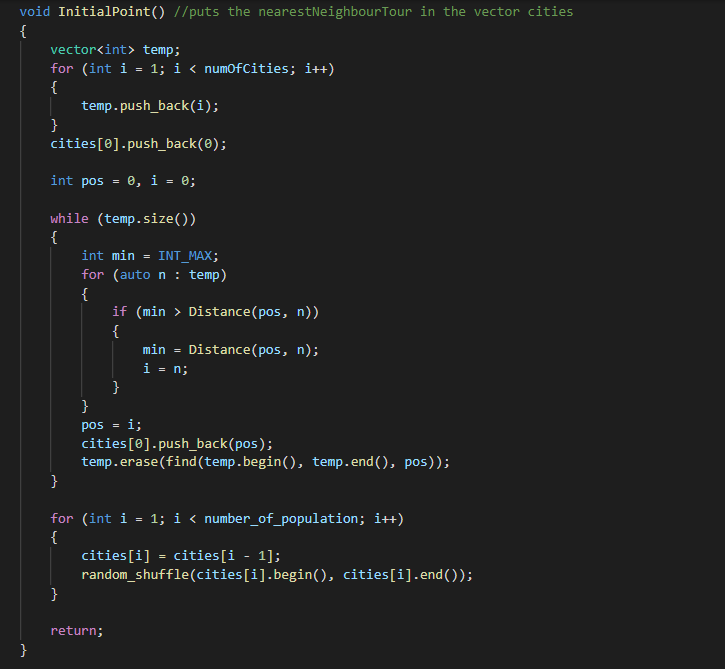
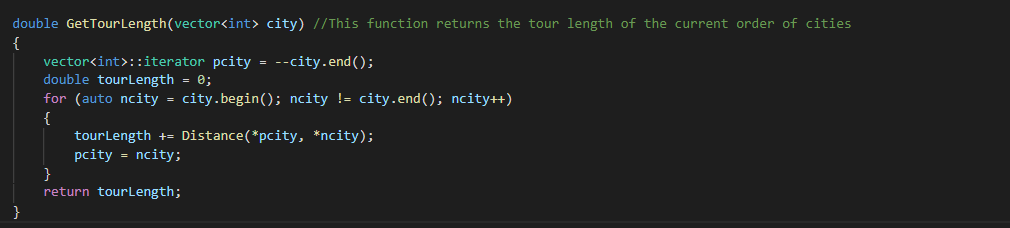
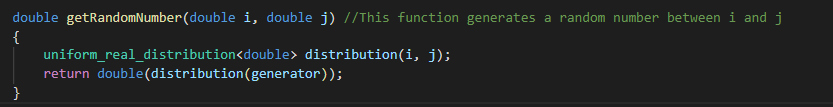
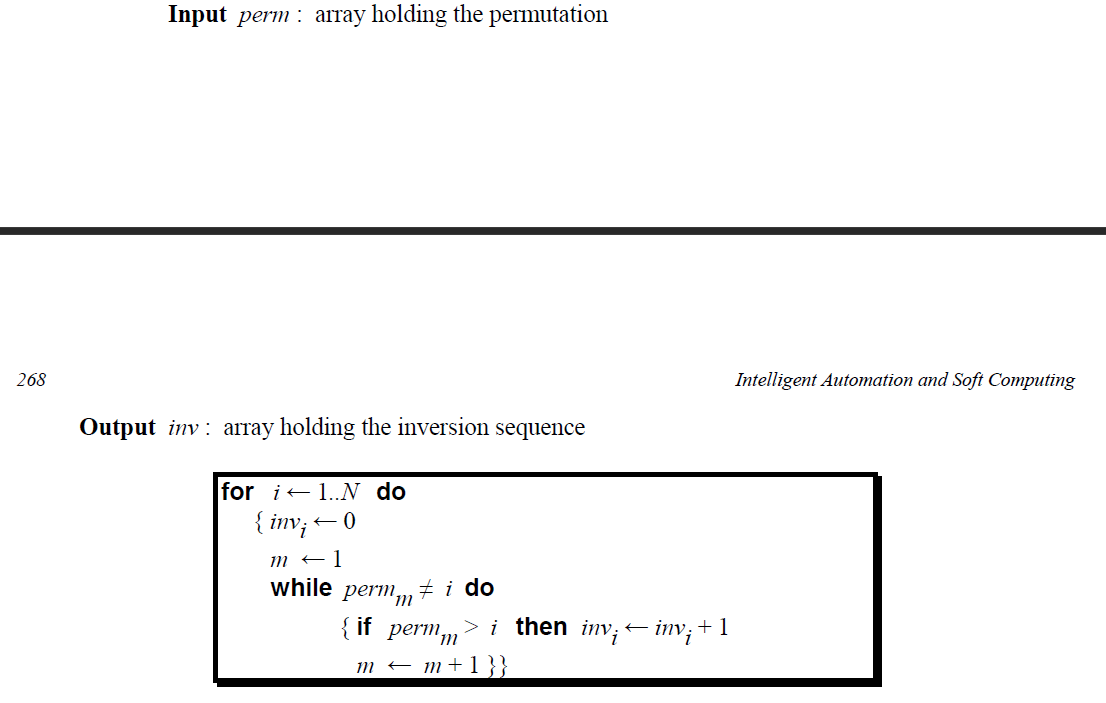
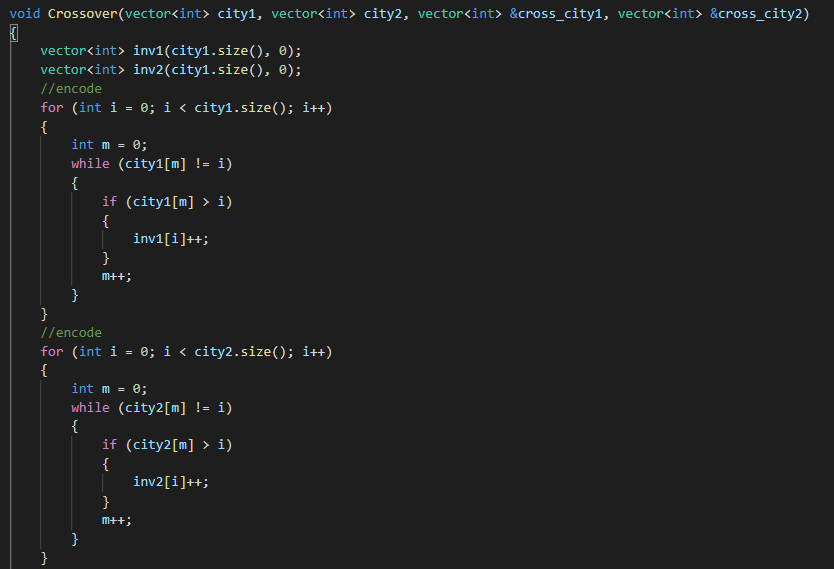
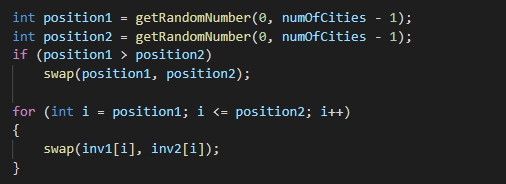
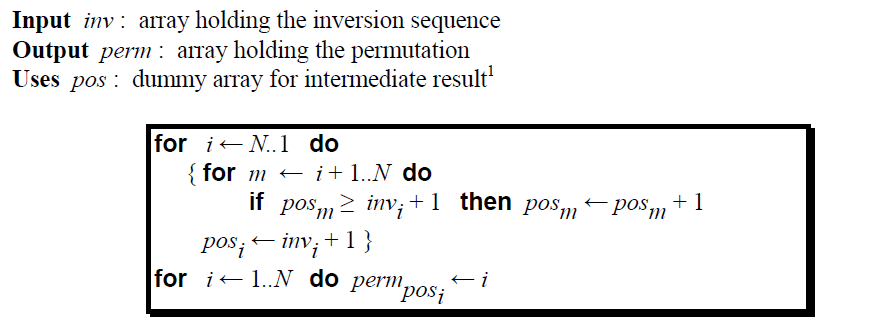
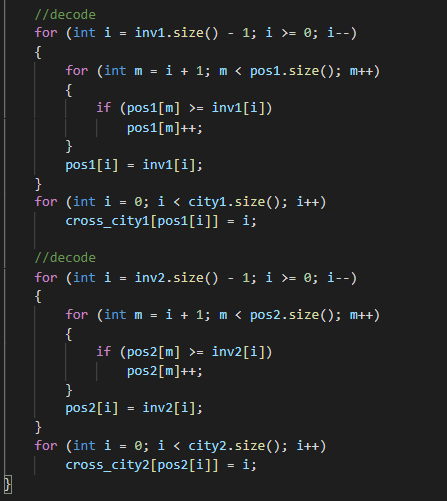
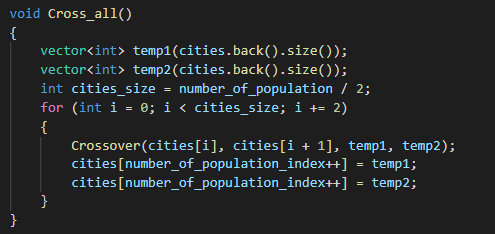
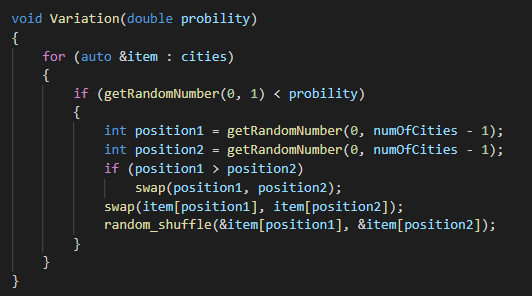
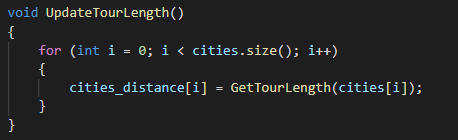
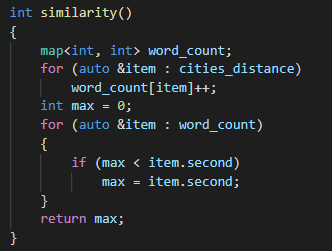
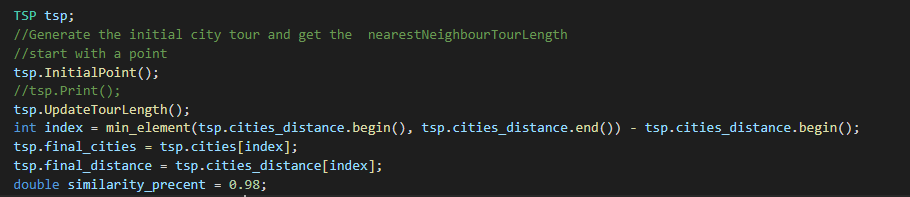
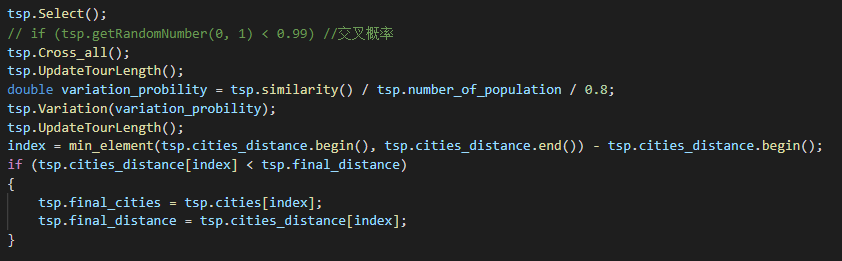
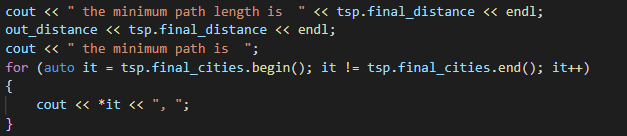
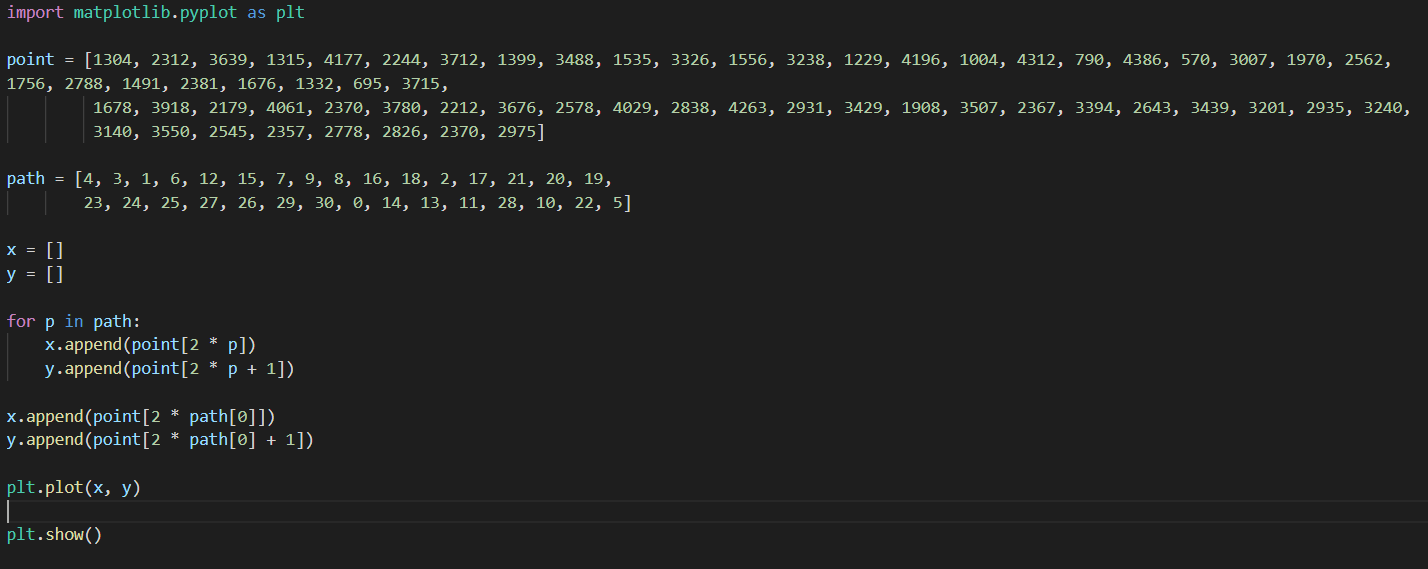
全国31个省会城市的坐标为[1304 2312; 3639 1315; 4177 2244; 3712 1399; 3488 1535; 3326 1556; 3238 1229; 4196 1004; 4312 790; 4386 570; 3007 1970; 2562 1756; 2788 1491; 2381 1676; 1332 695; 3715 1678; 3918 2179; 4061 2370; 3780 2212; 3676 2578; 4029 2838; 4263 2931; 3429 1908; 3507 2367; 3394 2643; 3439 3201; 2935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

1. 实验环境

Visual Studio code

本地进行测试，服务器调节数据跑实验

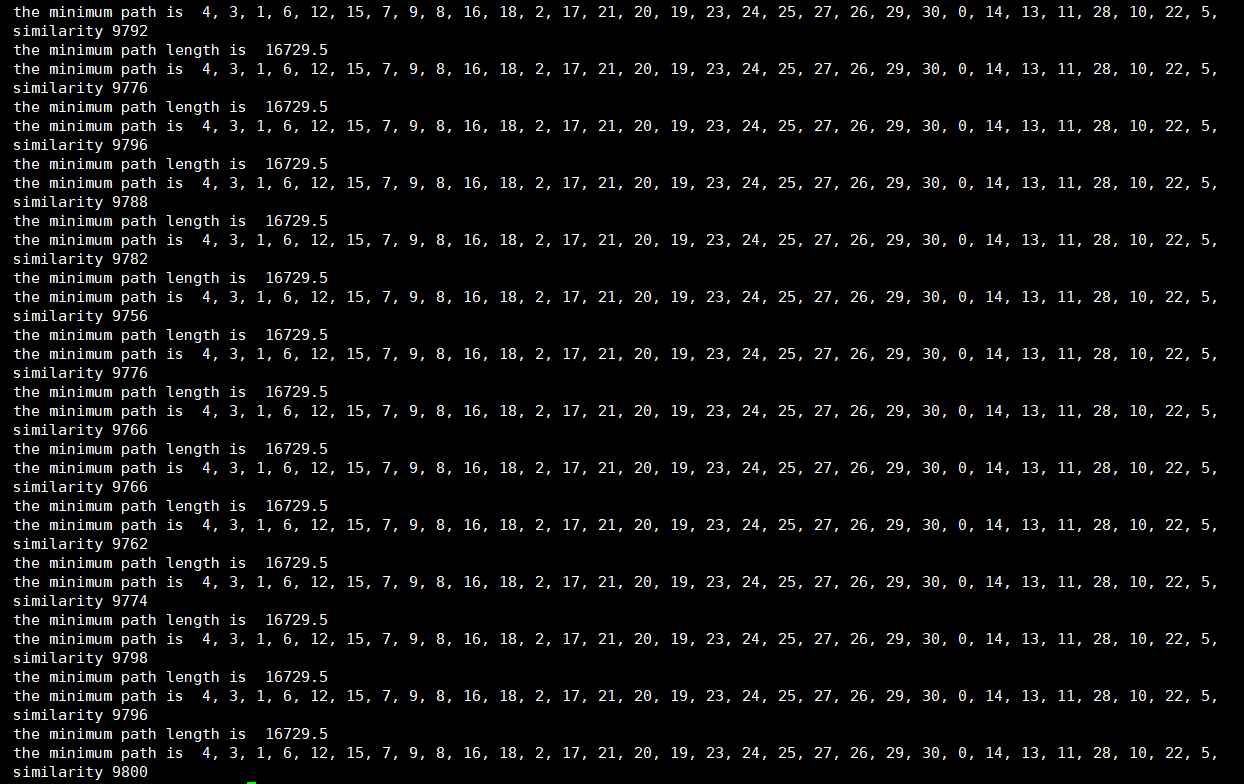
C++语言进行开发，python进行画图

1. 实验步骤
2. 首先初始化每个城市的坐标。
3. 使用欧氏距离来计算两个点之间的距离。
4. 创建一个记录每个点之间的距离的矩阵，并计算出相应的值放入矩阵。
5. 使用贪心算法初始化一个路径并随机产生其他更多的路径，直到达到设置的种群数量最大值。
6. 通过该函数计算一条路径的总长度。
7. 使用该函数得到一个i，j之间的随机数。
8. 选择阶段的算法。使用轮盘赌的思想，使用y=100000/x函数来计算。y是想要的适应度，x是该路径的长度。这样就能使路径越短的适应度更高。生成种群数量一半的后代。
9. 两个交叉。  
   首先对2个父母亲进行编码  
    。之后就可直接进行交叉  
   采用片段交叉。   
   然后进行解码，交叉就完成了。 
10. 总群体交叉。打乱顺序后相邻的进行交叉。
11. 变异。随机选择一个片段打乱。
12. 更新所有路径的长度。 
13. 计算整个种群的相似度。 
14. 调用函数生成城市，并找到最小值保存。设置结束时的种群相似度。
15. 停止条件。当种群相似度大于设置的值时。
16. 调用之前的算法，进行整个算法的流程。先进行选择，然后进行交叉，更新路径长度。变异概率我通过相似性来动态调整，当相似性越大时，变异的概率就越大。然后进行变异，更新路径长度，计算出最短路径并保存。
17. 最终输出答案。
18. 结果可视化。
19. 实验结果

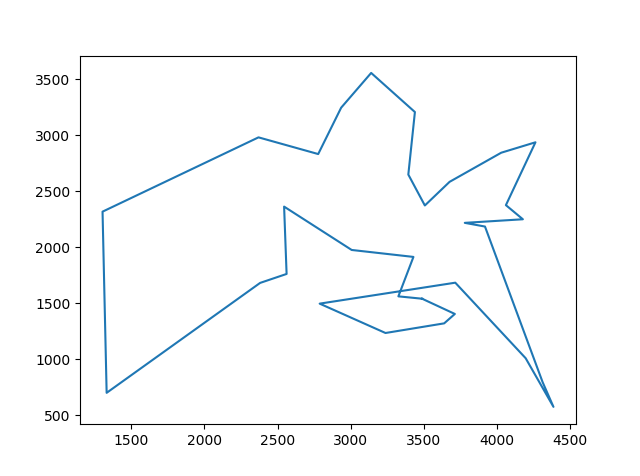
在这里，我简单的将程序进行了并行，以将CPU多核充分利用并提高效率。

根据我计算机的配置，我选择了如下的参数：种群相似性阈值为0.98，变异概率我使用的是如下的函数：



下面是使用该参数得到的结果。

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，通过多次运行，该程序的鲁棒性和计算速度不是太好，容易陷入局部最优解并且收敛较慢。

这是迭代时的种群相似度的变化。

这是迭代时的最优结果的距离的变化。

1. 实验总结

遗传算法（Genetic Algorithm，GA）是根据大自然中生物体进化规律而设计提出的。是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型，是一种通过模拟自然进化过程搜索最优解的方法。该算法通过数学的方式,利用计算机仿真运算,将问题的求解过程转换成类似生物进化中的染色体基因的交叉、变异等过程。在求解较为复杂的组合优化问题时,相对一些常规的优化算法,通常能够较快地获得较好的优化结果。遗传算法已被人们广泛地应用于组合优化、机器学习、信号处理、自适应控制和人工生命等领域。

由于该实验的结果受初始种群和变异概率影响较大。通过我进行的多次实验，我发现，  
（1） 想要降低初始值的影响，就需要增加算法的执行次数并记录最好的结果；  
（2） 想要得到越好的结果，这变异概率需要随着种群相似度的增大而增大，避免陷入局部最优解，但同时算法又需要更多的时间。