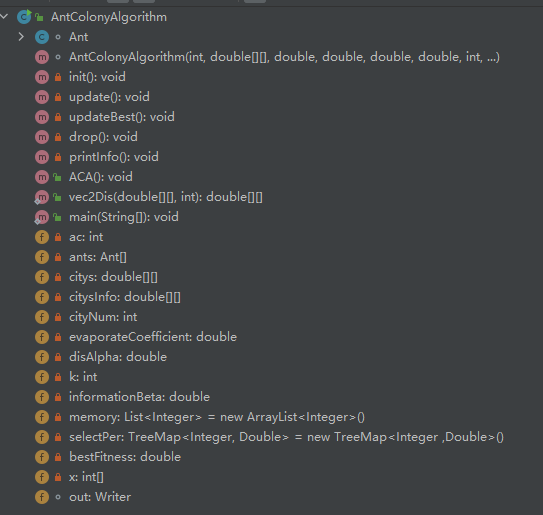
蚁群算法

1． 算法思想  
 蚁群算法是一种用来寻找优化路径的概率型算法。它由Marco Dorigo于1992年在他的博士论文中提出，其灵感来源于蚂蚁在寻找食物过程中发现路径的行为。这种算法具有分布计算、信息正反馈和启发式搜索的特征，本质上是进化算法中的一种启发式全局优化算法。蚁群系统(Ant System或Ant Colony System)是由意大利学者Dorigo、Maniezzo等人于20世纪90年代首先提出来的。他们在研究蚂蚁觅食的过程中，发现单个蚂蚁的行为比较简单，但是蚁群整体却可以体现一些智能的行为。例如蚁群可以在不同的环境下，寻找最短到达食物源的路径。这是因为蚁群内的蚂蚁可以通过某种信息机制实现信息的传递。后又经进一步研究发现，蚂蚁会在其经过的路径上释放一种可以称之为“信息素”的物质，蚁群内的蚂蚁对“信息素”具有感知能力，它们会沿着“信息素”浓度较高路径行走，而每只路过的蚂蚁都会在路上留下“信息素”，这就形成一种类似正反馈的机制，这样经过一段时间后，整个蚁群就会沿着最短路径到达食物源了。  
2． 算法设计

设蚁群中所有蚂蚁的数量为m，所有城市之间的信息素用矩阵citysInfo表示，城市间的距离用矩阵citys表示。最短路径为bestFitness，最佳路径为bestX。为蚂蚁单独创建一个类，每个蚂蚁都是类的一个实例化对象，每只蚂蚁都有自己的内存，采用一个List表示可达城市，每次更新都会更新当前可达城市的列表，蚂蚁会记录自己的路径以及适应度，并以此为依据更新城市的信息素矩阵。

3．代码设计



设计了一个主类,AntColonyAlgorithm，以及一个内部类Ant，由于Ant不会被别的类所调用，所以设计成内部类。将所得结果输出到文件中，并借用同学的Matlab处理数据并进行画图。

private void update() {

for (Ant a : ants) {

a.update();

}

}

private void updateBest() {

for (Ant a : ants) {

if (a.fitness < bestFitness) {

bestFitness = a.fitness;

x = a.getX();

System.out.println("updated " + bestFitness);

}

}

}

public void ACA() throws IOException {

init();

for (int i = 0; i < k; i++) {

update();

updateBest();

drop();

}

printInfo();

}

4．源代码

见附件。

5．实验测试

50个城市的坐标

88.6263755109159 51.7081936156540

18.9157463178892 2.97314404105127

5.20146297892062 93.9741545550866

85.7133569969534 10.5845731067004

4.79338042087024 53.5884676642844

98.0218037568871 68.5480752238465

38.1369015448087 94.9957382660233

51.5104680079221 72.5413512140760

13.6995731024090 6.53922304429371

8.32737787734371 45.4766235226785

81.1522672151172 18.8450781588384

84.8077200304583 21.0847505301893

76.3791145997499 21.7607874326927

52.9455598295550 51.5038603389182

39.0886493901097 35.1416171460595

40.9116278351014 69.3841499584880

95.5520660098299 66.3530707433996

55.5097548489196 12.9445328950385

66.8708986373368 58.2468210096325

19.1144824565057 0.737316707446822

67.4934426414759 50.2023940878336

96.5193862212747 77.8876934470199

59.8133700521287 94.8298574742081

92.2404540736888 92.0256288036293

23.9760687788337 83.1099316276181

49.7105375713187 12.1916395650402

2.31249552451155 95.6211007910860

62.7950180249898 65.8972125566090

28.4773466198276 83.2487867436172

24.6673993936470 47.1856910456068

85.0373829334244 75.2949044557633

93.0571013105838 6.17445676247925

56.6015971370545 45.5694387806052

2.42479936300664 93.0143173720299

54.0313304409630 11.5309347177546

42.4862173793135 85.4809300684227

50.7880342045245 85.7650895976390

83.1833937028049 51.6046208232739

47.9095305708964 79.2245462115565

24.7458373540089 99.3812720343682

23.3043459783892 48.7270826892021

45.2812405892299 68.8163892061884

72.4582198751308 31.2571191297137

22.6456426195076 28.7278945843072

96.8319064802370 45.5609221472125

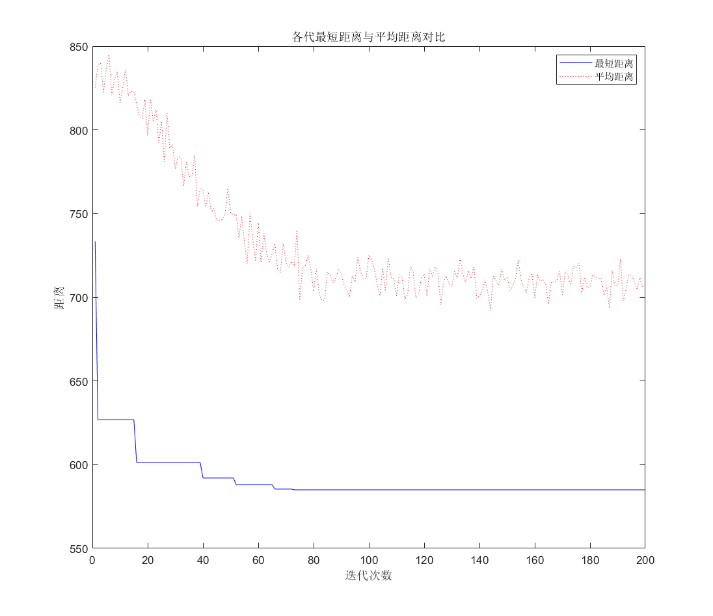
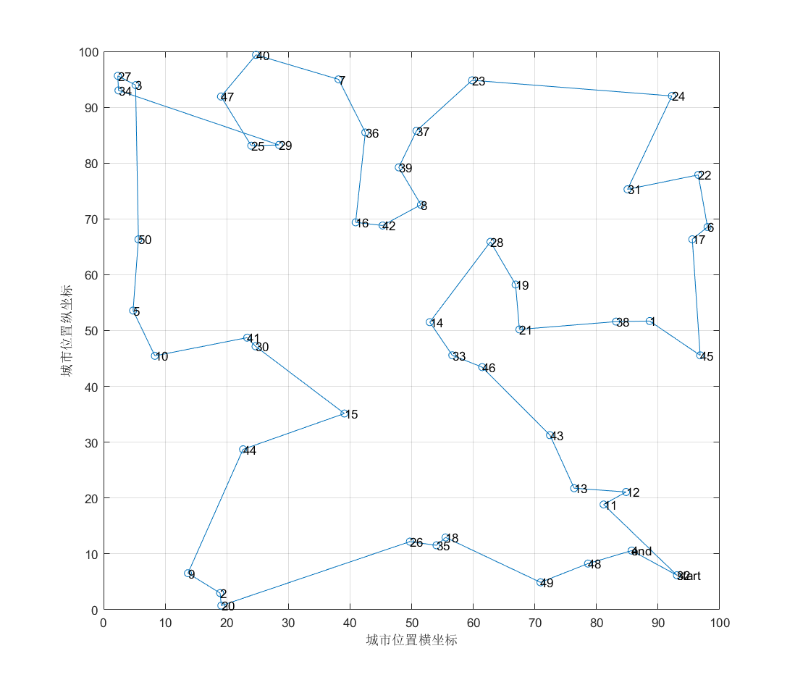
61.4503406194567 43.4490156570623

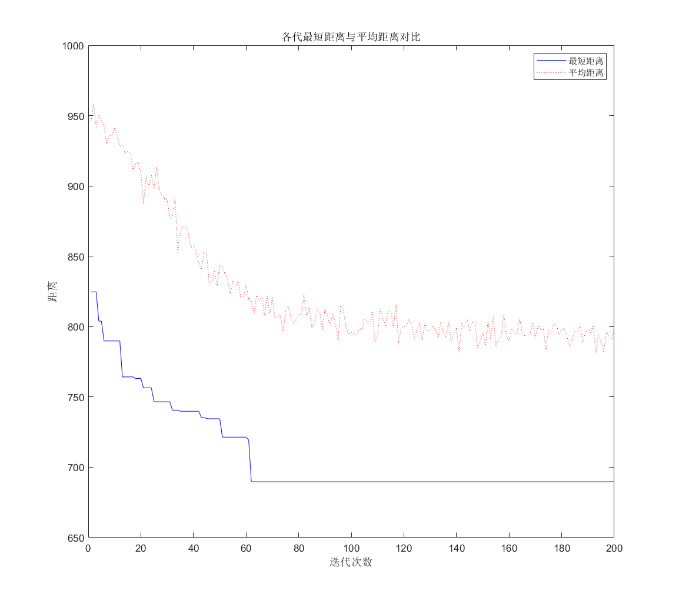
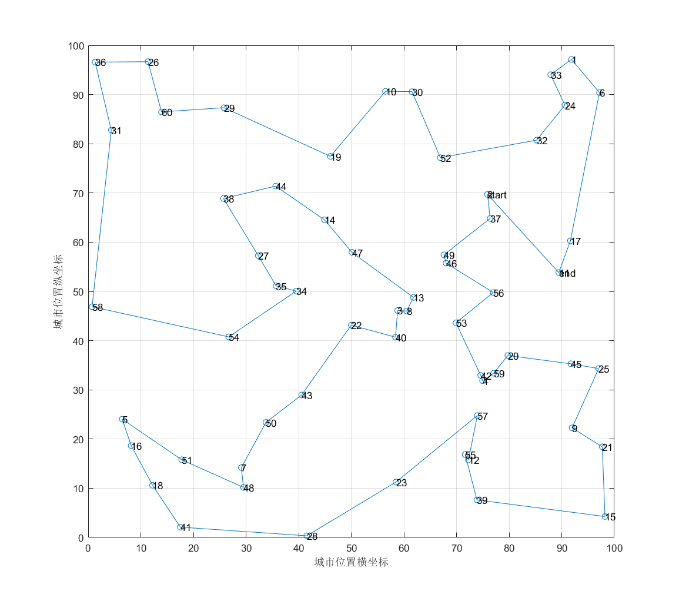
19.0637954527892 91.9181244543605

78.6225060897038 8.25952255631520

70.9149343026344 4.90039557850884

5.66169434981597 66.3472542108740





可以看到，参数调整的合适，一般而言可以在迭代迭代六十次的时候算法就趋于收敛了，也可以从城市的连接情况可以看出，城市的路径选择也是非常好的。

6. 结论

使用正反馈机制，使得算法不断地趋于收敛。每只蚂蚁在经过的路径上释放信息素，更改地图的信息，并通过地图间接影响下一代蚂蚁的路径选择。每个蚂蚁个体都可以独立的进行计算，提高了算法的执行效率。不容易陷入局部最优，易于寻找到全局最优解。