

1, 完成Django全部课程的学习

完成后端框架Django全部课程的学习, 学习了Form组件、Ajax验证、文件上传、iframe、jsonp等知识点, 并动手编写相应的代码作为练习。

2, 系统化学习python中面对对象的部分

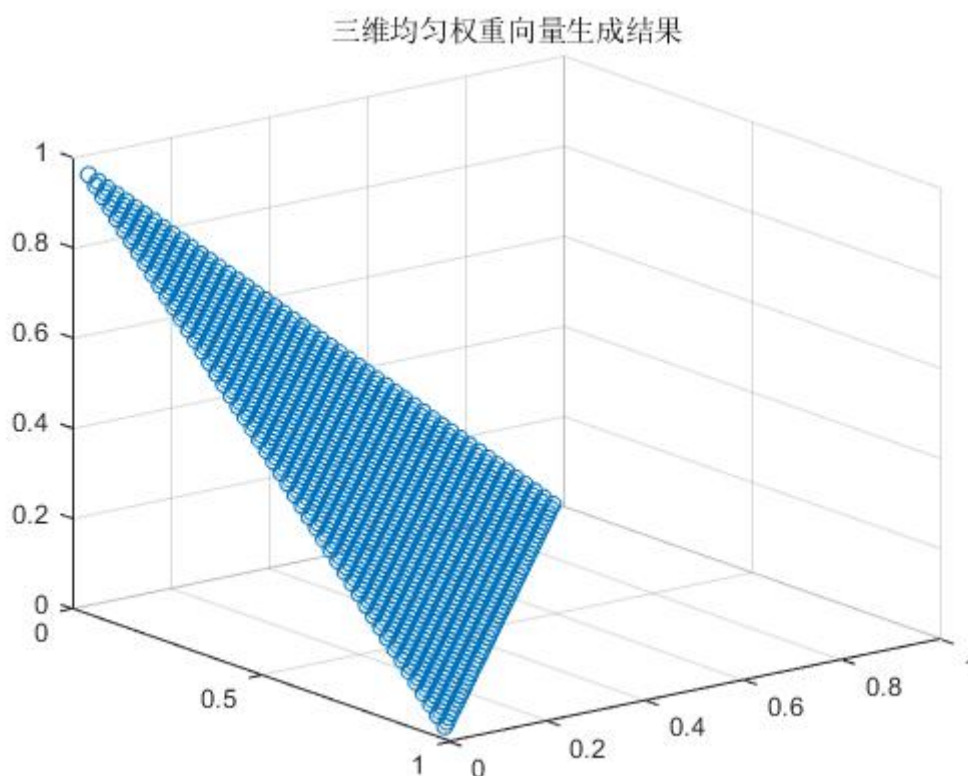
进行python面向对象编程的系统化学习, 学习了静态方法、反射、双下划线方法、动态创建类、异常处理、assert等知识点, 并编写对应的练习题, 保存在“面向对象教程”文件夹中。

3, 学习深度学习算法

学习吴恩达深度学习教程前三周的课程, 并进行了logistics回归的推导。

4, 学习均匀权重向量生成的算法

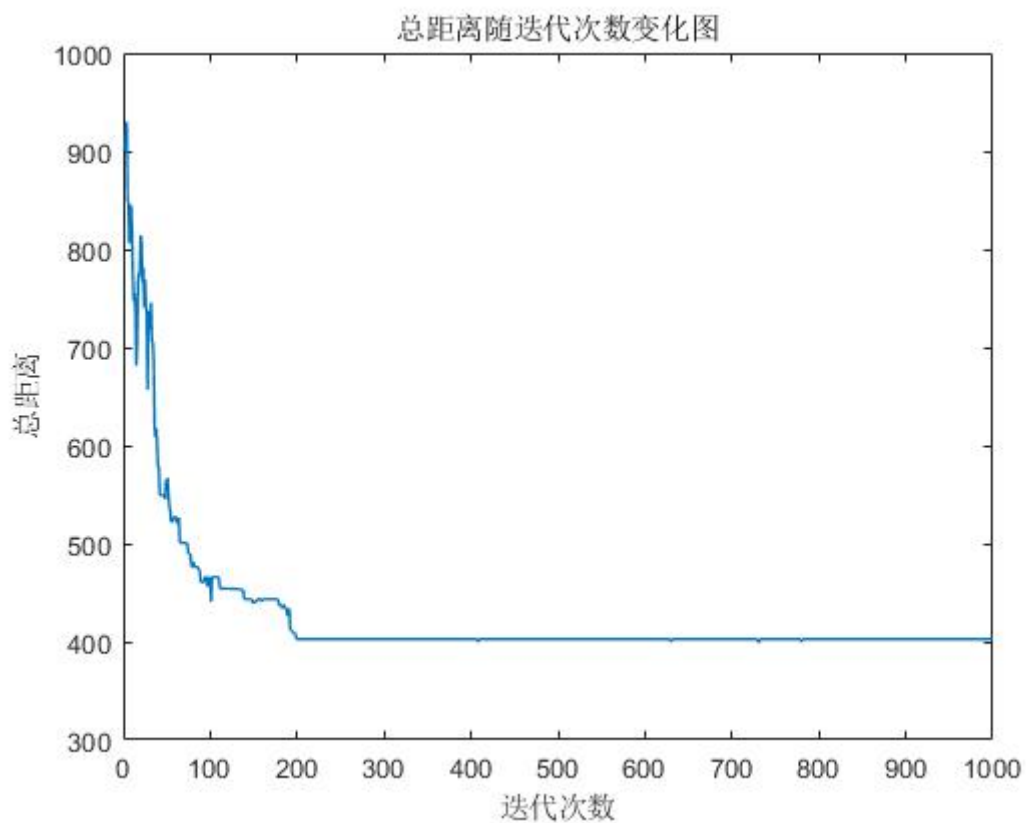
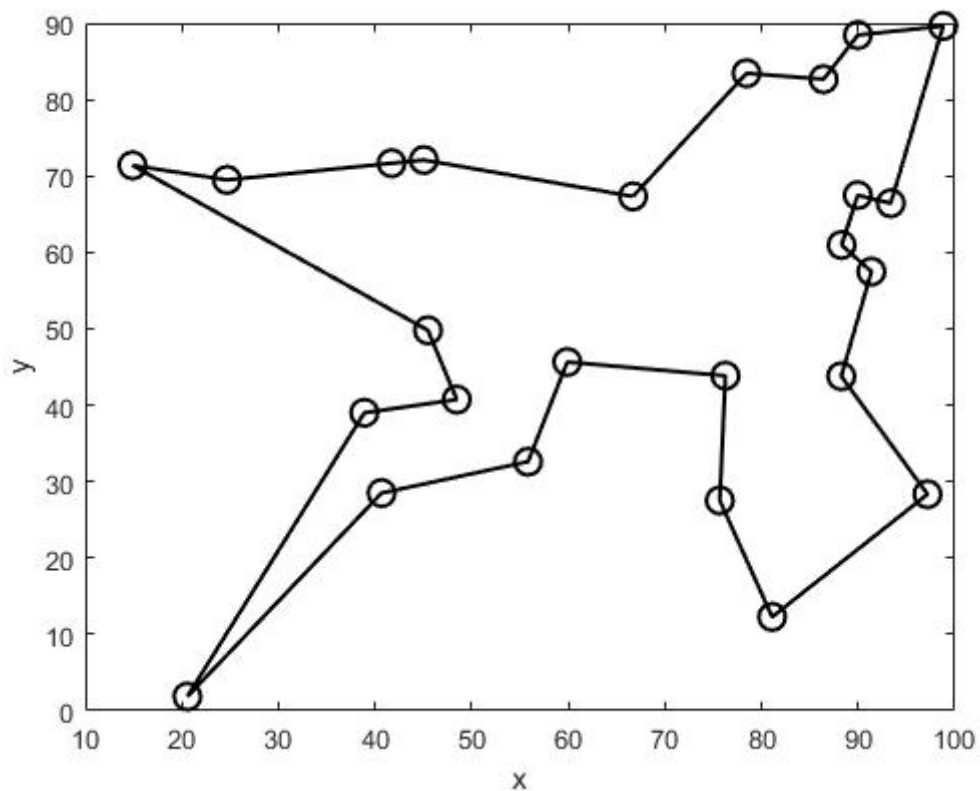
参考论文[1]提出的方法, 学习多维度均匀权重向量的生成算法, 为MOEA/D等多目标算法解决大于二维的问题打下基础。编写代码, 保存在“均匀权重向量集合的生成”文件夹中, 执行main函数, 可生成任意维度的权重向量。以三维为例:



[1] Das I, Dennis J E. Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems[J]. SIAM Journal on Optimization, 1998, 8(3): 631-657.

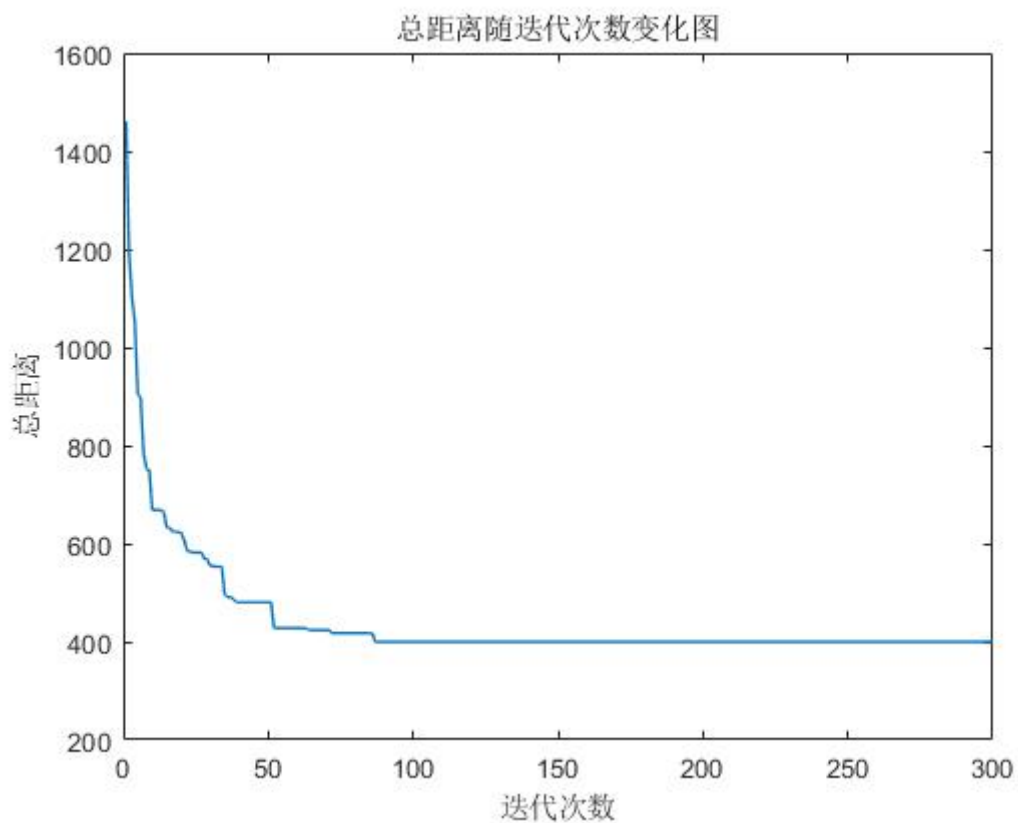
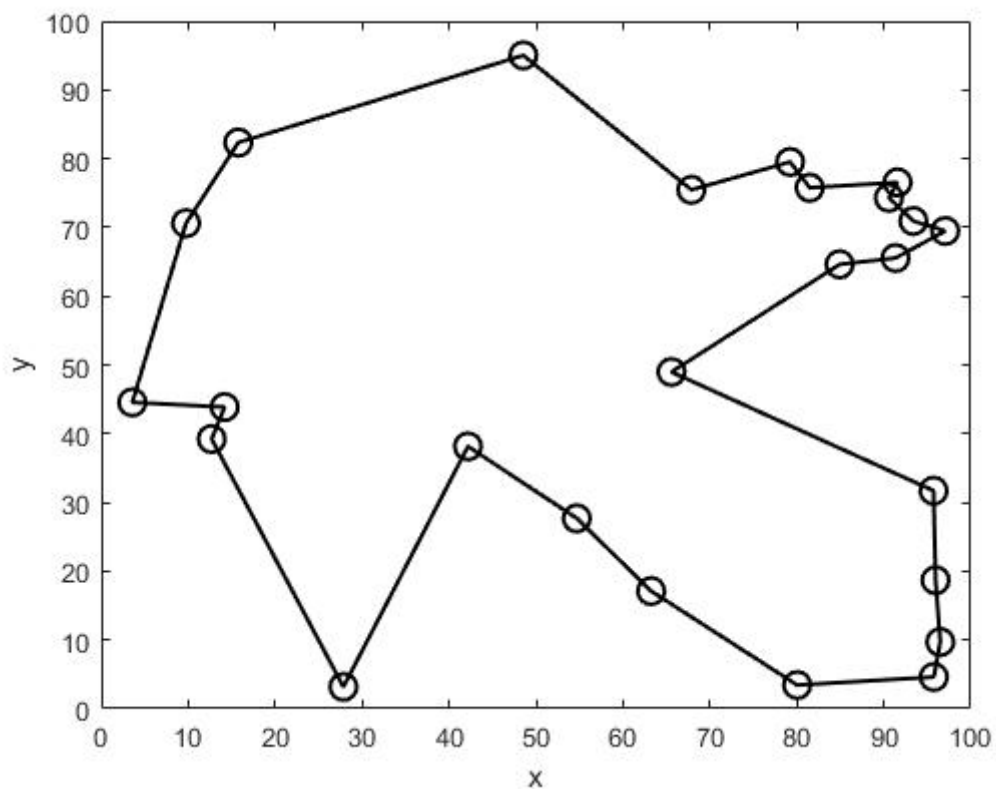
5, 遗传算法解决TSP问题

学习遗传算法关于TSP的编码, 并学习了变异的swap, insertion, reversion操作, 交叉的OX交叉、PBX交叉、OBX交叉操作, 完成代码的编写, 保存到“遗传算法解决TSP”文件夹中, 执行main函数, 可观察到优化过程。运行结果如下, 可看出在200代左右时算法收敛到最优解:



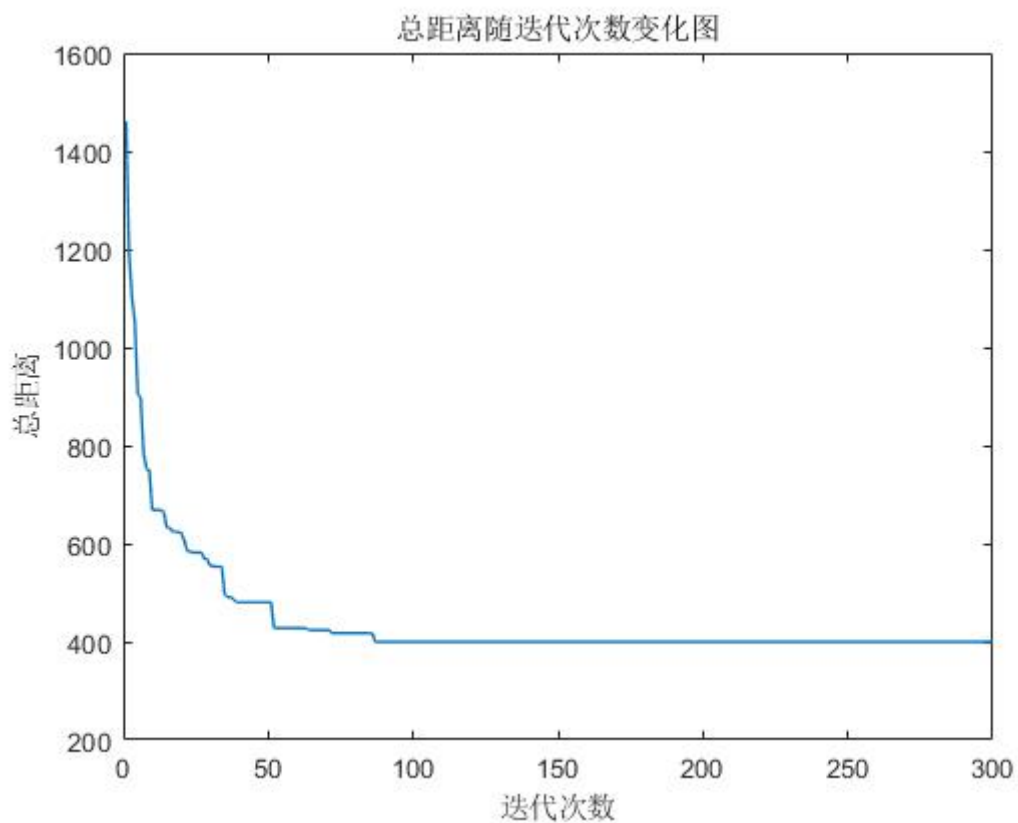
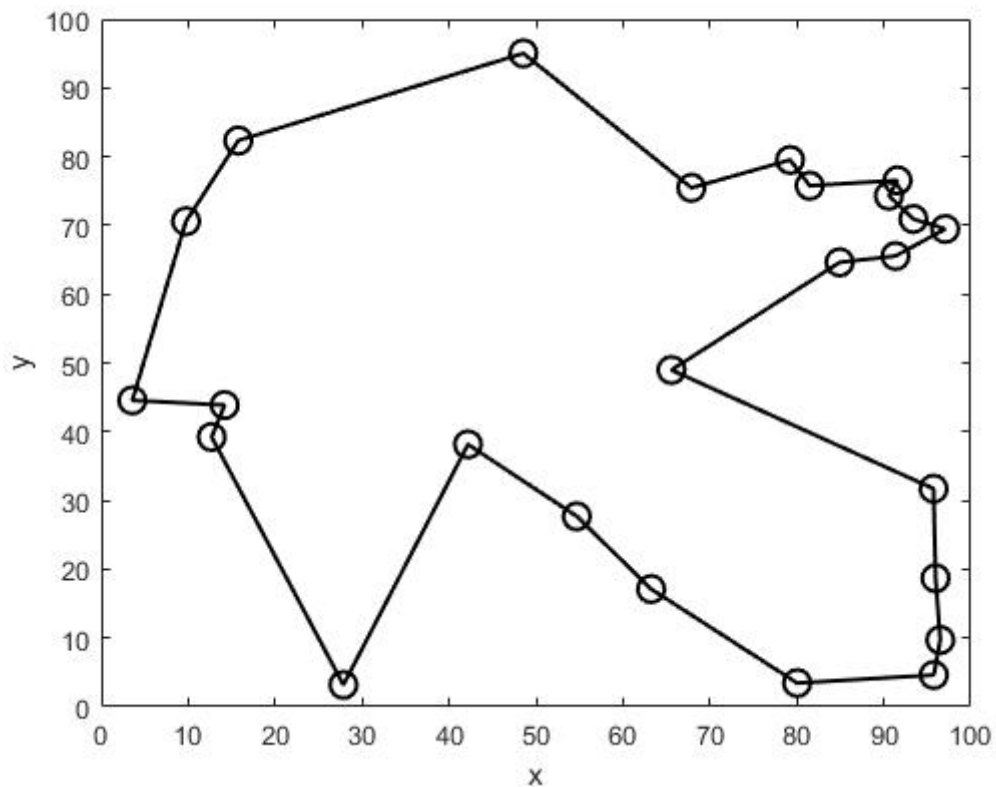
6, 模拟退火算法解决TSP问题

使用模拟退火算法解决TSP问题，完成代码的编写，保存到“模拟退火解决TSP”文件夹中，执行main函数，可观察出优化过程。运行结果如下，可看出在80代左右时算法收敛到最优解：



7, 学习遗传模拟退火算法, 并应用在解决TSP问题上

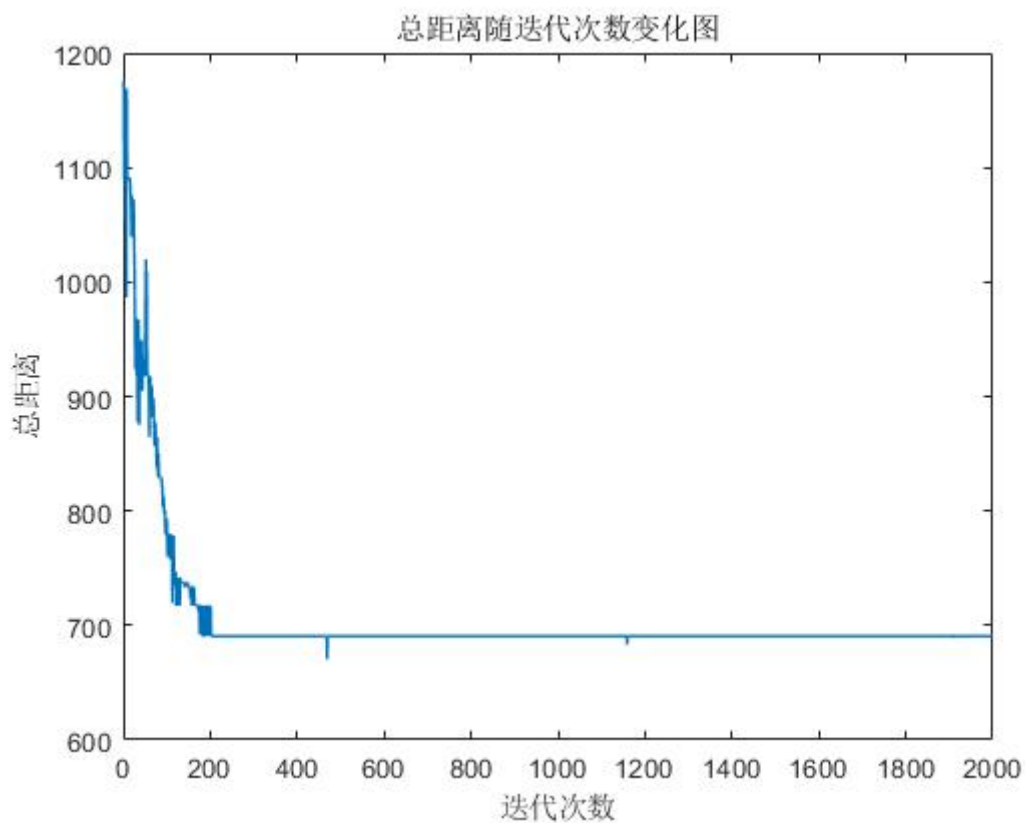
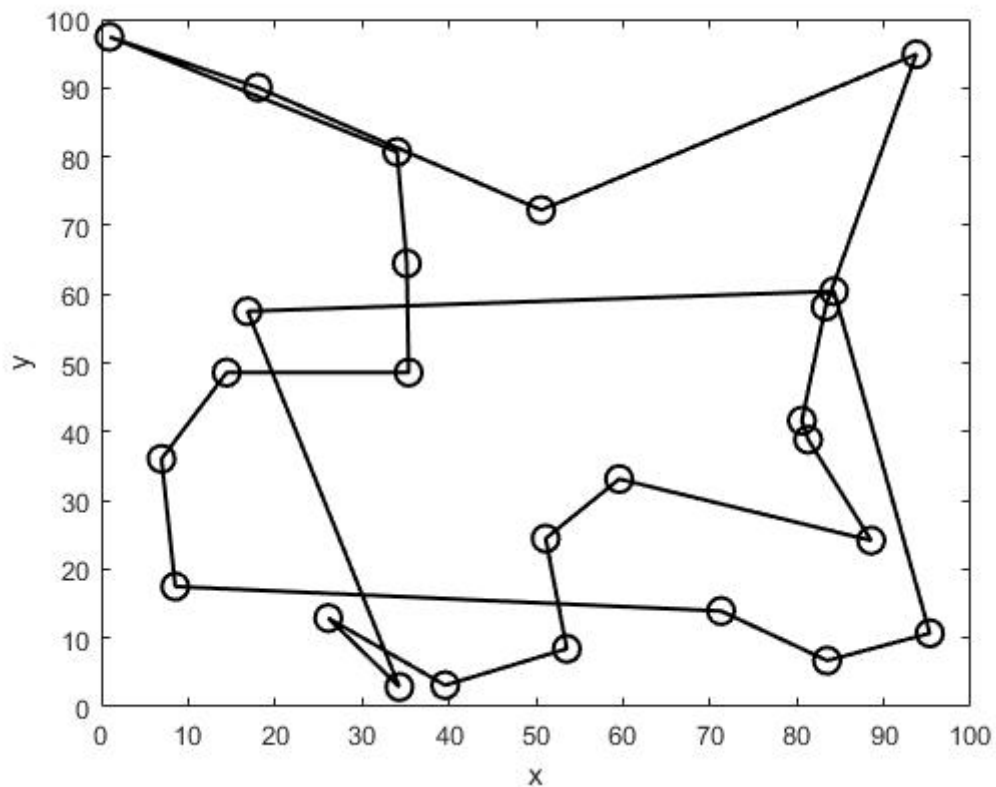
参考文献[2], 学习混合遗传模拟退火算法, 完成代码编写并保存到“混合遗传模拟退火算法”文件夹中, 执行main函数, 可观察出优化过程。运行结果如下, 可看出在120代左右时算法收敛到最优解:



[2] 刘怀亮, 刘淼. 一种混合遗传模拟退火算法及其应用[J]. 广州大学学报:自然科学版, 2005(02):141-145.

8, 学习Binary-Representation-Based Genetic Algorithm, 并应用在解决TSP问题上

参考论文[3], 学习Binary-Representation-Based Genetic Algorithm, 学习染色体结构及相应的变异、交叉操作, 完成代码编写并保存到“Binary-Representation-Based Genetic Algorithm 解决TSP”文件夹中, 执行main函数, 可观察出优化过程。运行结果如下, 可看出该算法陷入局部最优:

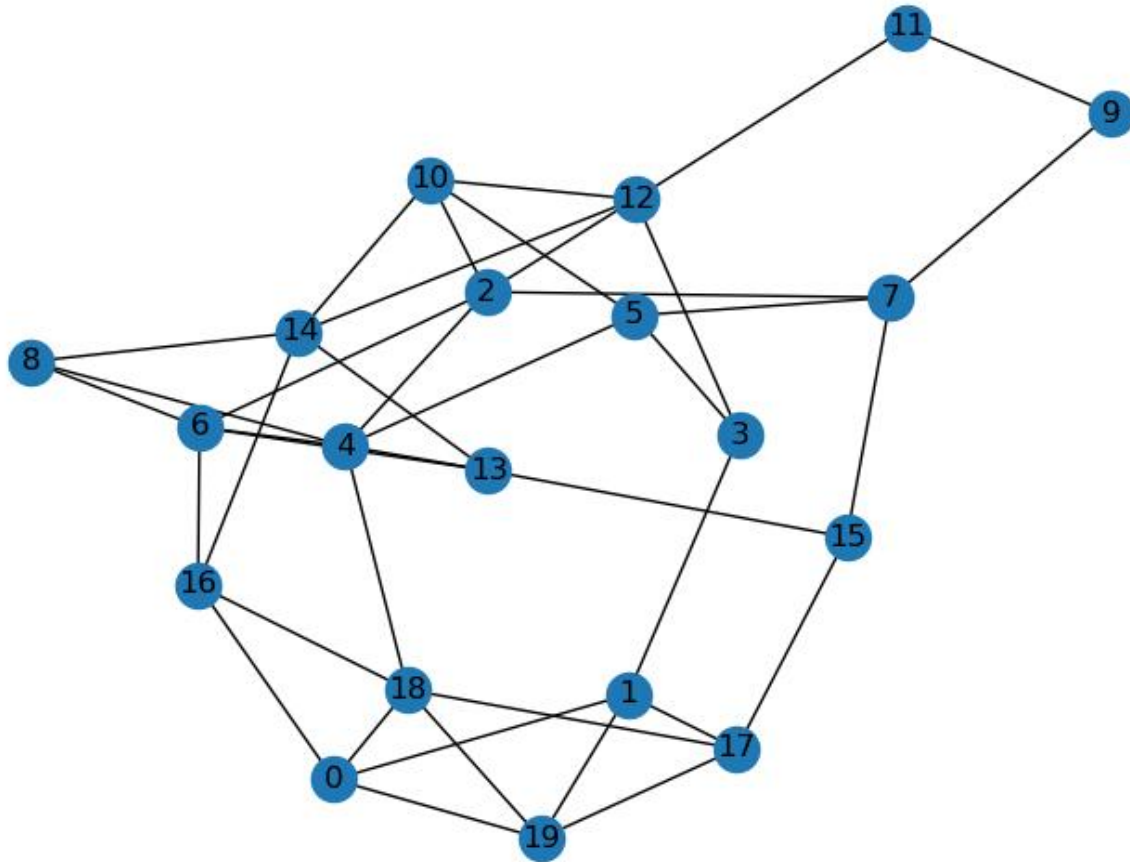


说明该基因编码结构不适用于解决TSP问题，由于TSP问题是一个环路，考虑将染色体进行改造为每行、每列只有一个位置是1的二维结构，该部分代码编写与测试工作将在下周完成。

[3] Hu X B , Di Paolo E . Binary-Representation-Based Genetic Algorithm for Aircraft Arrival Sequencing and Scheduling[J]. IEEE Transactions on Intelligent Transportation Systems, 2008, 9(2):301-310.

9, 学习经典网络生成算法

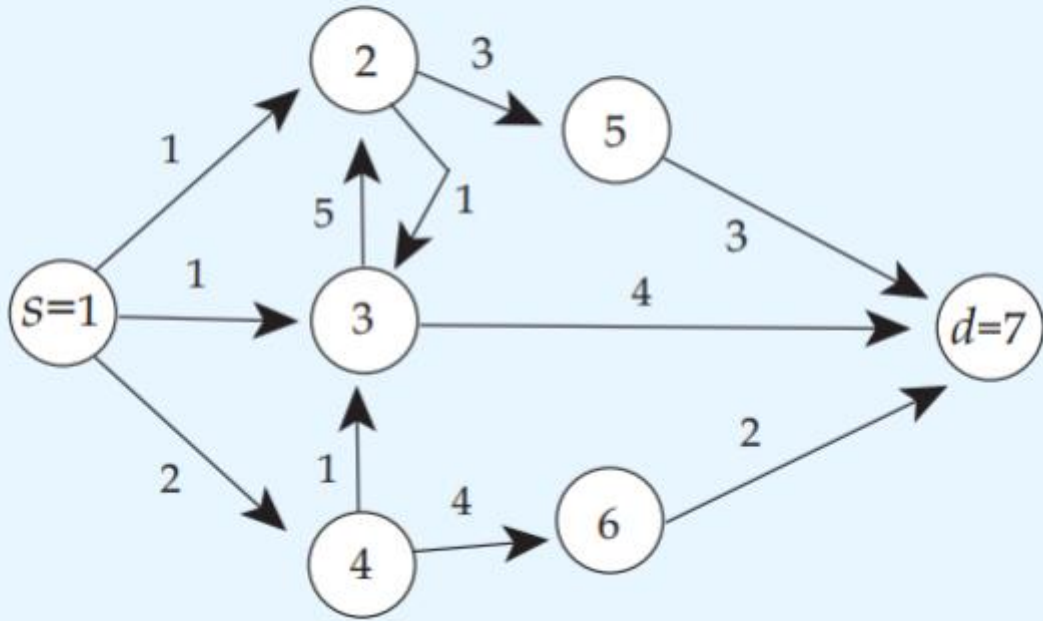
参考论文[4], 学习 BA Scale Free Network, ER Random Network Generator, WS Small World Network的生成算法, 并完成程序编写保存在“图生成器”文件夹中, 便于SPTP在大规模网络上进行测试。



[4] Holme P , Kim B J . Growing scale-free networks with tunable clustering[J]. Physical Review E, 2002, 65(2):026107.

10, 完成SPTP的编写, 并在大规模网络上进行测试

$N = 4;$
 $T_1 = \{s = 1\}, \quad T_2 = \{3\}, \quad T_3 = \{2, 4\}, \quad T_4 = \{d = 7\}.$



编写考虑顺序的SPTP问题，并和论文[5]中提出的方法进行比较，编写程序保存至“SPTP”文件夹中，并在大规模图中进行测试，发现涟漪扩散解决考虑顺序的SPTP问题的速度比论文[5]中提出的算法慢。

```
[0, 58, 5, 11, 50, 99]
((([1, 59, 106, 112, 251, 400], 67, 7), {'route': [1, 59, 6, 12, 51, 100], 'cost': 67}))
程序1运行时间9558ms
程序2运行时间41ms
```

重新检查程序，由于使用了面向对象的编程，每次生成新的涟漪都会初始化一个变量并存放在内存中，可能会影响程序的执行速度。并考虑新的数据结构重写程序，然后再作比较。

[5] Festa P, Guerriero F, D. Laganà, et al. Solving the shortest path tour problem[J]. European Journal of Operational Research, 2013, 230(3):464-474.

11, 自己动手编写kNN，并在手写图片数据集中进行算法测试

用numpy实现kNN算法，并在手写图片数据集中进行测试，该部分代码保存在“KNN”文件夹中，执行kNN，可得到在测试集上手写图片识别结果：

```
The classifier came back with: 9, the real answer is: 9
The classifier came back with: 9, the real answer is: 9
The classifier came back with: 9, the real answer is: 9
The classifier came back with: 9, the real answer is: 9
The classifier came back with: 9, the real answer is: 9
The classifier came back with: 9, the real answer is: 9

The total number of errors is: 10

The total error rate is: 0.010571
```

可看出错误率仅为0.01，可得出kNN算法编写无误。

下周计划:

- 1, 完成适用于TSP问题的二维染色体想法的编程实现, 并写测试函数, 横向比较遗传算法、模拟退火算法、混合遗传模拟退火算法解决TSP问题的运算速度、收敛速度。
- 2, 编写蚁群算法。
- 3, 编写logistics回归, 并选择测试集进行测试。
- 4, 阅读车辆路径问题相关论文。
- 5, 阅读SPTP问题的相关论文。
- 6, 编写新版SPTP程序, 并重新在大规模网络上进行测试。