**人工智能原理实验**

**实验三 遗传算法求TSP问题**

**组长：**周俊峰

**组员：**周俊峰

1. **实验目的**

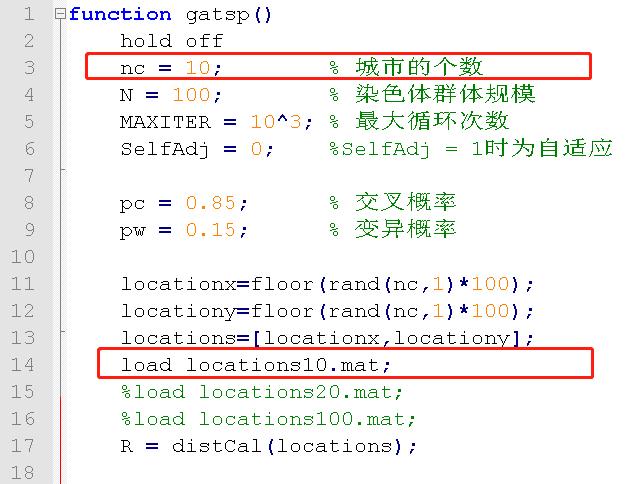
熟悉和掌握遗传算法的原理，流程和编码策略，理解求解TSP问题的流程并测试主要参数对结果的影响

1. **实验内容**

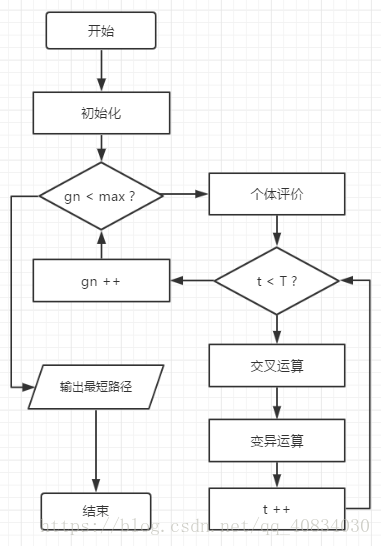
用遗传算法求解不同规模（如10个城市、20个城市、100个城市）的TSP问题。注意，切换到不同城市规模的时候需要改变变量nc的数值，并加载（load）对应的城市位置文件。

**摘要：**

TSP问题是指假设有一个旅行商人要拜访n个城市，他必须选择所要走的路径，路径的限制是每个城市只能拜访一次，而且最后要回到原来出发的城市。路径的选择目标是要求得的路径路程为所有路径之中的最小值。本文使用遗传算法解决att48问题，即48个城市的旅行商问题，该问题目前的最优解是10628，受个别参数影响以及数据样本数量的限制，本文所得到的最优结果是10648，相对误差为0.18818216%。



遗传算法的基本流程图如下图所示。gn对应当前代数，max为最大代数，T种群大小。



1. **实验要求**

1. 在主函数中有几个函数（包括calFitness( )、chooseNewP( )、crossPop( )、mutPop( )）需要重新定义，请搜索到相应的位置重新编写代码实现上述函数的功能

2. 编写完上述函数

3. 执行命令 gatsp 或 gatsp4student 得到一个TSP问题的解

4.用编写好的遗传算法程序求解不同规模（如10个城市、20个城市、100个城市）的TSP问题，同一种情况重复5次，把最短长度（即5次结果中最小数值）、最长长度（即5次结果中最大数值）及平均长度和运行时间填入表格一。（概率变量保持不变，即pc = 0.85;pw = 0.15;）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表格一** | | | | |
| **城市规模** | **最短长度** | **最长长度** | **平均长度** | **平均运行时间（秒）** |
| **10** |  |  |  |  |
| **20** |  |  |  |  |
| **100** |  |  |  |  |

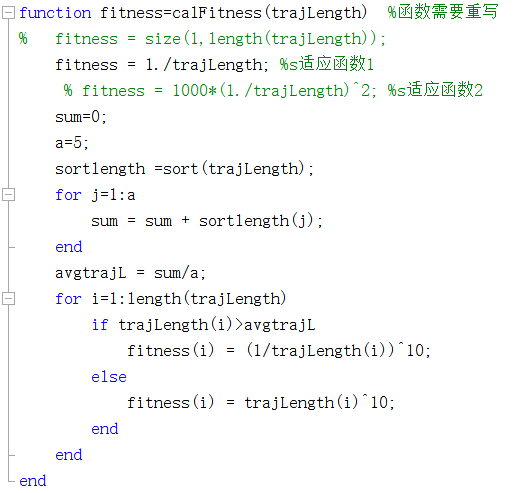
5.对于同一个TSP问题，设置不同的种群规模（如把代码中的染色体群体规模N设置成10/20/100），交叉概率（代码中的变量pc设置成 0/0.5/0.85/1 ）和变异概率（代码中的变量pw设置成 0/0.15/0.5/1 ），把结果填入表格二。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **表格二** | | | | | | |
| **城市规模** | **交叉概率** | **变异概率** | **最短长度** | **最长长度** | **平均长度** | **平均运行时间（秒）** |
| 10 | 0.85 | 0.15 |  |  |  |  |
| 20 | 0.85 | 0.15 |  |  |  |  |
| 100 | 0.85 | 0.15 |  |  |  |  |
| 100 | 0 | 0.15 |  |  |  |  |
| 100 | 0.5 | 0.15 |  |  |  |  |
| 100 | 1 | 0.15 |  |  |  |  |
| 100 | 0.85 | 0 |  |  |  |  |
| 100 | 0.85 | 0.5 |  |  |  |  |
| 100 | 0.85 | 1 |  |  |  |  |

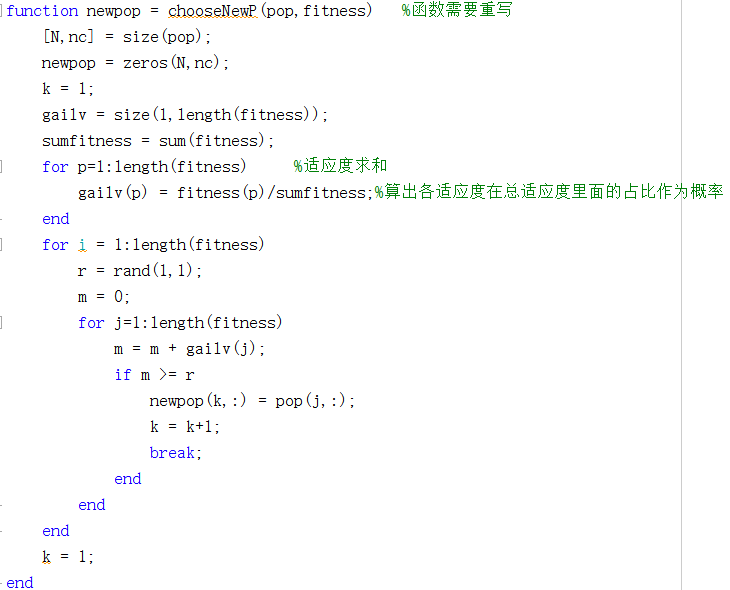
**四，实验结果**

以下为补充的四个函数代码：

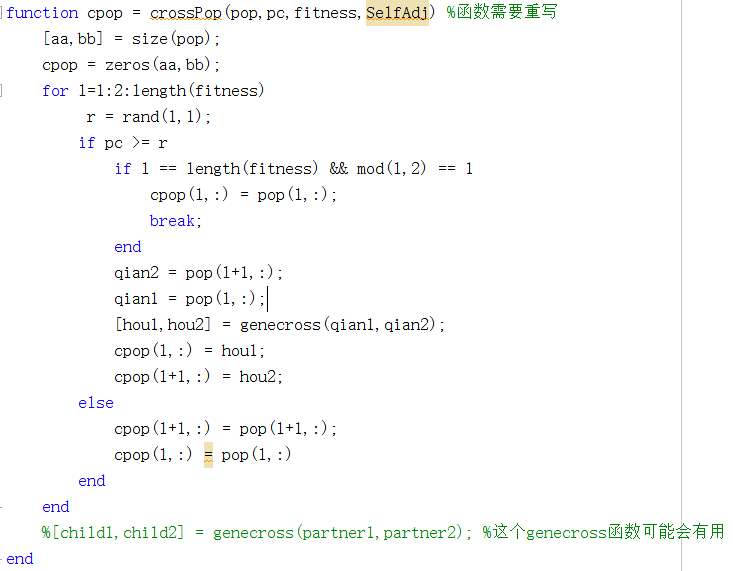
calFitness( )



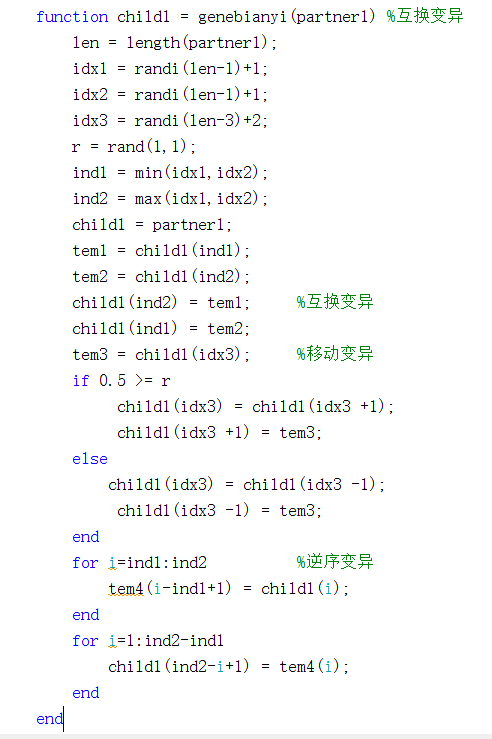
chooseNewP( )



crossPop( )



mutPop( )



根据上述代码运行结果如下：

声明：我们程序的最大循环次数是定值1000，以下结果都由最大循环次数1000得出。

2. 表格一

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **表格一** | | | | |
| **城市规模** | **最短长度** | **最长长度** | **平均长度** | **平均运行时间（秒）** |
| **10** | **324.7267** | **361.4761** | **343.1471** | **6.875** |
| **20** | **473.325** | **578.2221** | **531.6458** | **11.639** |
| **100** | **1284.591** | **1499.5082** | **1390.7246** | **62.5734** |

从上面的表格我们看到，随着城市规模的加大，长度越来越大，运行时间也越来越多。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **表格二** | | | | | | |
| **城市规模** | **交叉概率** | **变异概率** | **最短长度** | **最长长度** | **平均长度** | **平均运行时间（秒）** |
| 10 | 0.85 | 0.15 | **324.7264** | **334.5528** | **331.2773** | **6.885** |
| 20 | 0.85 | 0.15 | **519.7242** | **591.0299** | **548.1102** | **16.4797** |
| 100 | 0.85 | 0.15 | **1225.7331** | **1593.0364** | **1409.9019** | **102.622** |
| 100 | 0 | 0.15 | **1306.9721** | **1477.2917** | **1358.5241** | **22.3154** |
| 100 | 0.5 | 0.15 | **1333.953** | **1536.9373** | **1421.8562** | **67.352** |
| 100 | 1 | 0.15 | **1354.263** | **1498.8756** | **1439.2685** | **81.241** |
| 100 | 0.85 | 0 | **1210.654** | **1339.2515** | **1273.6225** | **66.251** |
| 100 | 0.85 | 0.5 | **1425.2641** | **1459.3642** | **1452.1978** | **103.5214** |
| 100 | 0.85 | 1 | **4251.6354** | **4425.7592** | **4289.3692** | **108.3621** |

**实验总结**

用遗传算法解决TSP问题不一定得到最优解，而且解的优劣情况与循环次数、染色体个数、适应值函数的优劣、变异概率、突变概率有关。对于TSP问题，我们都设计出相应的函数来使应对不同种群规模，不同交叉概率，不同变异概率对算法结果造成的影响。这次实验让我们对遗传算法以及TSP问题有了更进一步的了解。