实验 1：利用遗传算法解决 TSP 问题

一、实验目的

- 掌握遗传算法在实际问题中的应用方法
- 理解遗传算法中染色体编码和适应度函数的定义方法
- 熟悉遗传算法中选择、交叉、变异操作在实际问题中的设计方法
- 熟悉遗传算法中的控制参数设置对性能的影响

二、实验原理

2.1 旅行商问题

旅行商问题也叫旅行推销员问题（Travelling salesman problem, TSP）：假设有一个旅行商人要拜访 n 个城市，每个城市只能拜访一次，而且最后要回到原来出发的城市。那么旅行商在何种路径选择的情况下，最后总路程最小。在数学中其被定义为 NP 困难问题，在运筹学和理论计算机科学中非常重要。

最早的旅行商问题的数学规划是由 Dantzig（1959）等人提出，并且是在最优化领域中进行了深入研究。许多优化方法都用它作为一个测试基准。尽管问题在计算上很困难，但已经有了大量的启发式和精确方法来求解数量上万的实例，并且能将误差控制在 1%内。

对于 TSP 这类 NP 困难问题，目前比较主流的方法是采用启发式的搜索算法，比如遗传算法、蚁群算法、模拟退火算法等。在本次实验中，要求通过遗传算法来解决 TSP 问题。

2.2 遗传算法

遗传算法（Genetic Algorithm）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，通过模拟自然进化过程搜索最优解。遗传算法是从一组候选解构成的一个种群（population）开始的，初代种群产生之后，按照适者生存和优胜劣汰的原理，逐代（generation）演化产生出越来越好的近似解，

在每一代，根据问题域中个体的适应度（fitness）大小选择个体，并借助于的遗传算子进行组合交叉（crossover）和变异（mutation），产生出代表新的解集的种群。这个过程将导致种群按照自然进化一样产生后生代种群，并且比前代更加适应于环境，末代种群中的最优个体经过解码（decoding），可以作为问题近似最优解。

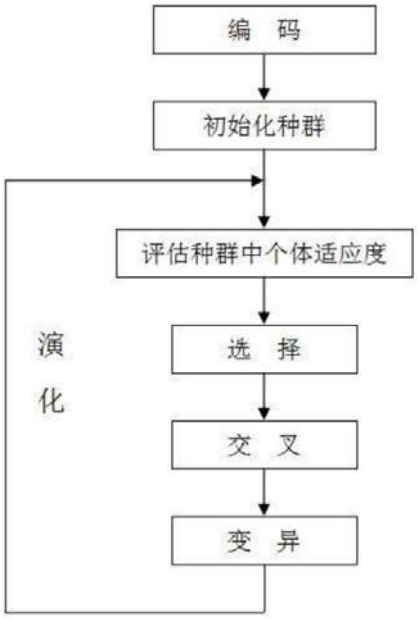

图 1.遗传算法基本流程

如图 1 所示，在遗传算法中，一共有如下几个步骤：

1. 初始化：设置进化代数计数器 t=0、设置最大进化代数 T、交叉概率、变异概率、随机生成 M 个个体作为初始种群 P

2. 个体评价：计算种群 P 中各个个体的适应度

3. 选择运算：将选择算子作用于群体。以个体适应度为基础，选择最优个体直接遗传到下一代或通过配对交叉产生新的个体再遗传到下一代

4. 交叉运算：在交叉概率的控制下，对群体中的个体两两进行交叉

5. 变异运算：在变异概率的控制下，对群体中的个体两两进行变异，即对某一个体的基因进行随机调整

6. 经过选择、交叉、变异运算之后得到下一代群体 P’。

2.3 测试数据

在本次实验中，我们给定了中国 34 个省会城市的二维坐标，其中部分数据截图如下：

重庆,106.54,29.59
拉萨,91.11,29.97
乌鲁木齐,87.68,43.77
银川,106.27,38.47
呼和浩特,111.65,40.82
南宁,108.33,22.84
哈尔滨,126.63,45.75
长春,125.35,43.88
沈阳,123.38,41.8
石家庄,114.48,38.03
太原,112.53,37.87
西宁,101.74,36.56
济南,117,36.65
郑州,113.6,34.76
南京,118.78,32.04
合肥,117.27,31.86
杭州,120.19,30.26
福州,119.3,26.08
南昌,115.89,28.68
长沙,113,28.21
武汉,114.31,30.52
广州,113.23,23.16
台北,121.5,25.05

数据总共由 34 行组成，每一行代表一个城市名字以及对应坐标。两个城市之间的距离可以通过对应坐标求欧氏距离得到。

三、实验要求

对于给定数据，要求选择始发城市和剩余 33 个城市中的全部城市或部分城市作为需要遍历的城市，通过编写相应的遗传算法代码，求解 TSP 问题中回到始发城市的路径，并且找到路径总长度最短的解。

3.1 算法实现要求

1) 保证染色体编码中的路径是合法的（即不出现重复城市）;

2) 设计遗传算法中的初始化、交叉、变异操作，保证进化中种群中每个染色体都是合法的。如果有能力的话，可实现多种的交叉、突变方式，并对比不同方法的优劣性;

3) 在旅行商需要经过所有 34 个城市的情况下，能够在较短的时间内得到一个最优解。

4）分析遗传算法中的主要控制参数对求解问题性能的影响

3.2 其他要求

可从多个方面进一步提高算法设计。比如设定在经过 A 城市之后，必须经过 B 城市，如何设计满足该要求的遗传算法，或者研究如何采用其他方法提高算法

的收敛速度。（该部分的完成情况占一定实验成绩比重）

四、实验步骤

4.1 染色体编码

设计 TSP 问题的染色体编码方式并进行编程实现。设计种群初始化方法并确保不出现“非法”染色体。

4.2 评估

设计 TSP 问题的适应度函数并进行编程实现。

4.3 选择

设计合适的个体选择方法，如轮盘赌、竞标赛选择，并进行编程实现。

4.4 交叉

设计合理有效的交叉操作方案，保证在交叉操作中不出现“非法”的染色体，并进行编程实现。

4.5 突变

设计合理有效的突变操作方案，保证在交叉操作中不出现“非法”的染色体，并进行编程实现。

4.6 结束程序

设计遗传进化终止条件，当条件满足时结束程序，返回最好结果。

五、其他说明

建议用 Python 或者 Java 实现本实验中的算法

实验 2：TSP 问题的可视化程序设计

一、实验目的

- 对实验 1 中的遗传算法为后台程序，建立相应的可视化和用户交互前端程序
- 掌握程序设计中的事件监听方法并实现一定的用户交互操作
- 熟悉前端程序调用后台算法的基本方法

二、实验原理

在本次实验中，我们给定了中国 34 个省会城市的二维坐标，其中部分数据截图如下：

```
重庆,106.54,29.59
拉萨,91.11,29.97
乌鲁木齐,87.68,43.77
银川,106.27,38.47
呼和浩特,111.65,40.82
南宁,108.33,22.84
哈尔滨,126.63,45.75
长春,125.35,43.88
沈阳,123.38,41.8
石家庄,114.48,38.03
太原,112.53,37.87
西宁,101.74,36.56
济南,117,36.65
郑州,113.6,34.76
南京,118.78,32.04
合肥,117.27,31.86
杭州,120.19,30.26
福州,119.3,26.08
南昌,115.89,28.68
长沙,113,28.21
武汉,114.31,30.52
广州,113.23,23.16
台北,121.5,25.05
```

数据总共由 34 行组成，每一行代表一个城市名字以及对应坐标。利用实验 1 中已经实现的遗传算法作为后台执行程序，设计实现相应的前端可视化程序。例如，34 个城市的可视化结果如图 2 左图所示：

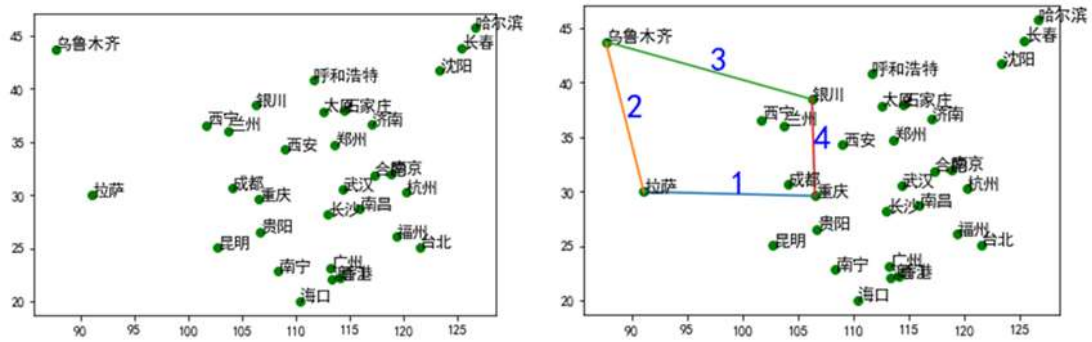

图 2. 地图可视化

两个城市之间的距离可以通过两个城市的坐标来求出，假设在旅行商问题中，旅行商希望经过的城市为乌鲁木齐、银川、拉萨、重庆这四个城市并且起点为重

庆，那么一个可行的解的可视化结果如图 2 右图所示。设计的软件程序中包含类似的地图可视化。

三、实验要求

1) **界面设计：**可视化程序设计简洁、用户交互易操作，并具有一定美观性；

2） **地图显示：**能够在可视化程序的地图中正确显示 34 个城市的位置及城市之间的位置关系；

3）**用户设置：**用户可以通过一定的交互界面选择所有 34 个城市作为搜索集合，也可以自行选择其中的部分城市作为搜索集合，程序需要给出对应搜索集合下 TSP 问题的求解方案，用户可以自行设定起始城市（起始城市需在搜索集合中）；

3）**输出方案：**可视化界面中，可以让用户自行设定输出可行方案的个数（如果采用遗传算法求解，可以规定输出可行方案个数不得超过个体总数）；对于输出的不同的解决方案，需要有明显的区分（比如颜色，形状等）；对于一种解决方案，在输出路径时候，需要标注路径的起始点，以及路径遍历的顺序；

4）**统计分析：**能够使用户直观的了解到不同解决方案的优劣性。