实验二：遗传算法

**【实验目的】**

遗传算法(GeneticAlgorthm,GA）起源于对生物系统所进行的计算机模拟研究。其本质是一种高效、并行、全局搜索的方法，能在搜索过程中自动获取和积累有关搜索空间的知识，并自适应地控制搜索过程以求得最佳解。本实验通过解决旅行商问题，帮助学生更好的熟悉和掌握遗传算法。

【**实验内容**】

**利用遗传算法解决旅行商问题**

旅行商问题即TSP问题（Traveling Salesman Problem）又译为旅行推销员问题、货郎担问题，是数学领域中著名问题之一。假设有一个旅行商人要拜访n个城市，每两座城市之间的距离是不同的，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。

【实验原理】

一、进化论知识

**种群(Population)**：生物的进化以群体的形式进行，这样的一个群体称为种群。

**个体**：组成种群的单个生物。

**基因 ( Gene )** ：一个遗传因子。

**染色体 ( Chromosome )** ：包含一组的基因。

**生存竞争，适者生存**：对环境适应度高的、牛B的个体参与繁殖的机会比较多，后代就会越来越多。适应度低的个体参与繁殖的机会比较少，后代就会越来越少。

**遗传与变异：**新个体会遗传父母双方各一部分的基因，同时有一定的概率发生基因变异。

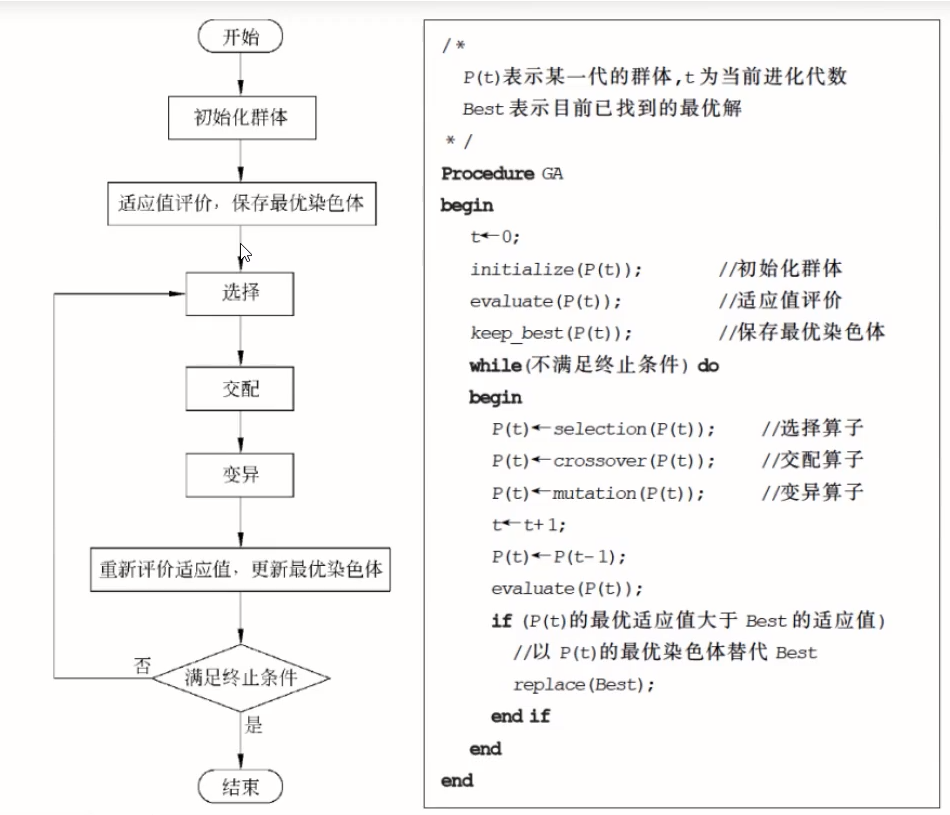
简单说来就是：繁殖过程，会发生基因交叉( Crossover ) ，基因突变 ( Mutation ) ，适应度( Fitness )低的个体会被逐步淘汰，而适应度高的个体会越来越多。那么经过N代的自然选择后，保存下来的个体都是适应度很高的，其中很可能包含史上产生的适应度最高的那个个体。

二.遗传算法思想

　　借鉴生物进化论，遗传算法将要解决的问题模拟成一个生物进化的过程，通过复制、交叉、突变等操作产生下一代的解，并逐步淘汰掉适应度函数值低的解，增加适应度函数值高的解。这样进化N代后就很有可能会进化出适应度函数值很高的个体。

【例】

使用遗传算法解决“0-1背包问题”的思路：



0-1背包的解可以编码为一串0-1字符串（0：不取，1：取） ；首先，随机产生M个0-1字符串，然后评价这些0-1字符串作为0-1背包问题的解的优劣；然后，随机选择一些字符串通过交叉、突变等操作产生下一代的M个字符串，而且较优的解被选中的概率要比较高。这样经过G代的进化后就可能会产生出0-1背包问题的一个“近似最优解”。

**编码：**需要将问题的解编码成字符串的形式才能使用遗传算法。最简单的一种编码方式是二进制编码，即将问题的解编码成二进制位数组的形式。例如，问题的解是整数，那么可以将其编码成二进制位数组的形式。将0-1字符串作为0-1背包问题的解就属于二进制编码。

　　遗传算法有3个最基本的操作：**选择，交叉，变异**。

**选择**：选择一些染色体来产生下一代。一种常用的选择策略是 “比例选择”，也就是个体被选中的概率与其适应度函数值成正比。假设群体的个体总数是M，那么那么一个体Xi被选中的概率为f(Xi)/( f(X1) + f(X2) + …….. + f(Xn) ) 。比例选择实现算法就是所谓的“轮盘赌算法”( Roulette Wheel Selection ) 。

**交叉(Crossover)**：2条染色体交换部分基因，来构造下一代的2条新的染色体。例如：

交叉前：

00000|011100000000|10000

11100|000001111110|00101

交叉后：

00000|000001111110|10000

11100|011100000000|00101

染色体交叉是以一定的概率发生的，这个概率记为Pc 。

**变异(Mutation)**：在繁殖过程，新产生的染色体中的基因会以一定的概率出错，称为变异。变异发生的概率记为Pm 。例如：

变异前：

000001110000000010000

变异后：

000001110000100010000

**适应度函数 ( Fitness Function )**：用于评价某个染色体的适应度，用f(x)表示。有时需要区分染色体的适应度函数与问题的目标函数。例如：0-1背包问题的目标函数是所取得物品价值，但将物品价值作为染色体的适应度函数可能并不一定适合。适应度函数与目标函数是正相关的，可对目标函数作一些变形来得到适应度函数。

【**实验要求**】

1. 选择测试用的目标函数，设计有效的遗传算子，分别编写初始化函数、适应度函数、复制函数、交换函数、变异函数以及主函数。

2.编程解决旅行商问题

3.撰写实验报告