## 实验三 TSP问题的遗传算法实现

**一 实验目的**

1 熟悉和掌握遗传算法的基本概念和基本思想；

2 理解和掌握遗传算法的各个操作算子，能够用选定的编程语言设计简单的遗传优化系统；

3 通过实验培养学生利用遗传算法进行问题求解的基本技能。

**二 实验预习内容**

1 旅行商问题的表示；

2 遗传算法的基本概念和基本算子。

**三 实验内容**

以N个节点的TSP（旅行商问题）问题为例，应用遗传算法进行求解，求出问题的最优解。

1 旅行商问题

旅行商问题（Traveling Salesman Problem, TSP），又译为旅行推销员问题、货担郎问题，简称为TSP问题，是最基本的路线问题。假设有n个可直达的城市，一销售商从其中的某一城市出发，不重复地走完其余n-1个城市并回到原出发点，在所有可能的路径中求出路径长度最短的一条。

TSP问题是组合数学中一个古老而又困难的问题，也是一个典型的组合优化问题，现已归入NP完备问题类。NP问题用穷举法不能在有效时间内求解，所以只能使用启发式搜索。遗传算法是求解此类问题比较实用、有效的方法之一。

下面给出30个城市的位置信息：

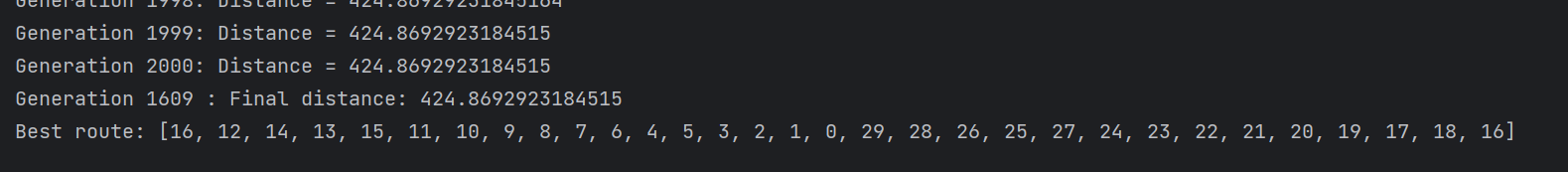
表1 Oliver TSP问题的30个城市位置坐标

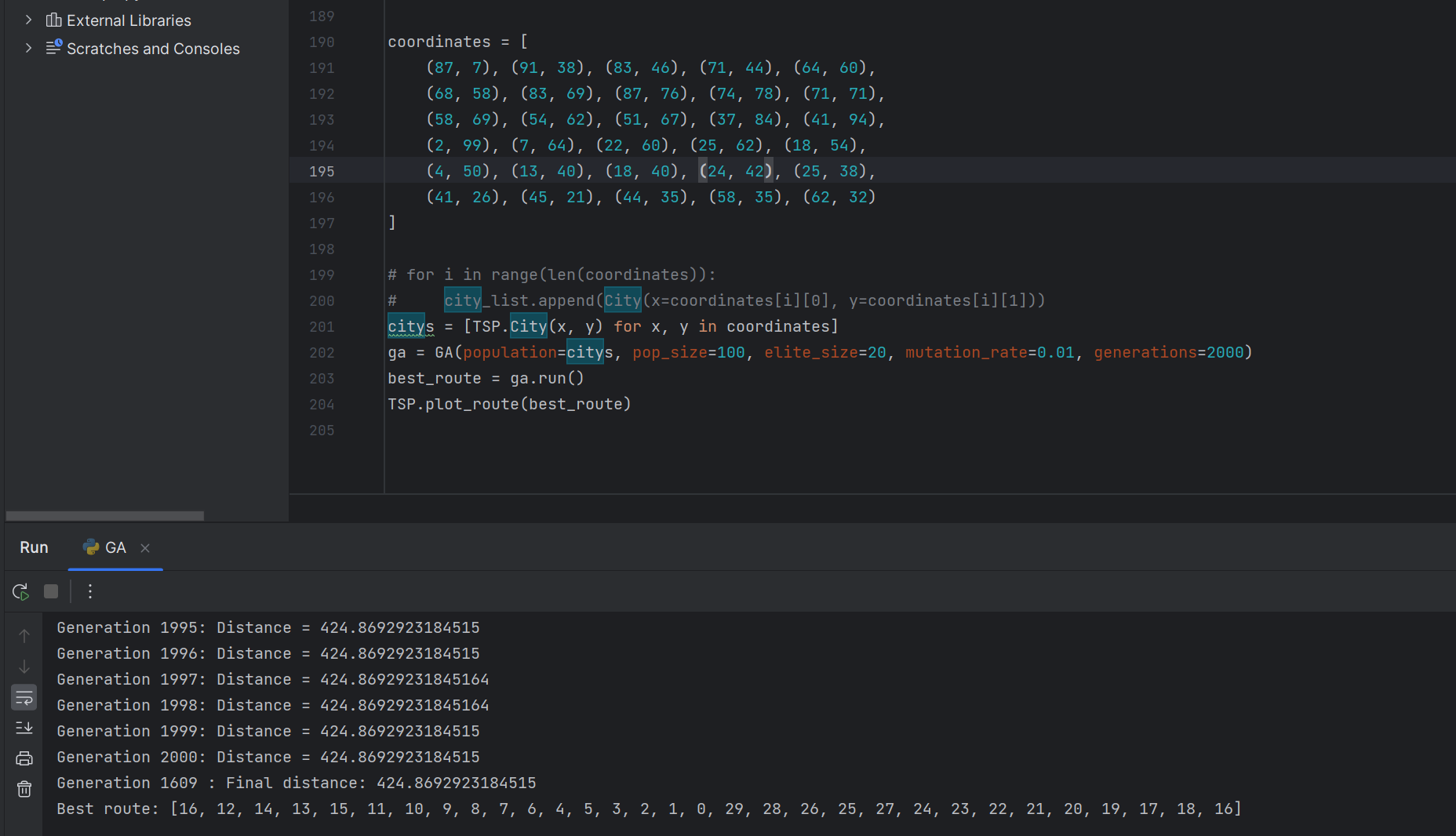
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 城市编号 | 坐标 | 城市编号 | 坐标 | 城市编号 | 坐标 |
| 1 | (87，7) | 11 | (58，69) | 21 | (4，50) |
| 2 | (91，38) | 12 | (54，62) | 22 | (13，40) |
| 3 | (83，46) | 13 | (51，67) | 23 | (18，40) |
| 4 | (71，44) | 14 | (37，84) | 24 | (24，42) |
| 5 | (64，60) | 15 | (41，94) | 25 | (25，38) |
| 6 | (68，58) | 16 | (2，99) | 26 | (41，26) |
| 7 | (83，69) | 17 | (7，64) | 27 | (45，21) |
| 8 | (87，76) | 18 | (22，60) | 28 | (44，35) |
| 9 | (74，78) | 19 | (25，62) | 29 | (58，35) |
| 10 | (71，71) | 20 | (18，54) | 30 | (62，32) |

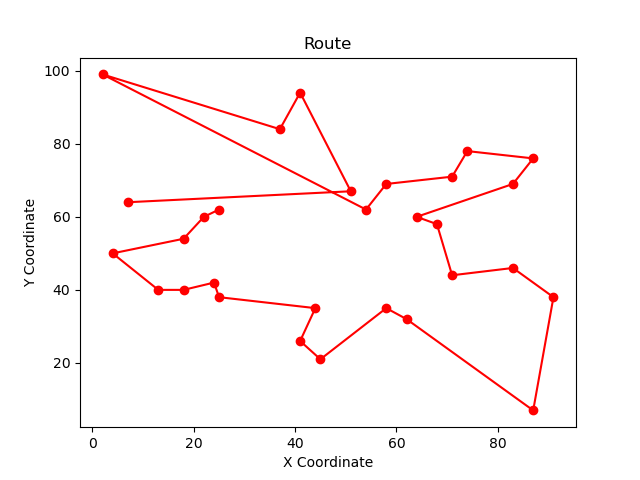
最优路径为：1 2 3 4 6 5 7 8 9 10 11 12 13 14 15 16 17 19 18 20 21 22 23 24 25 28 26 27 29 30

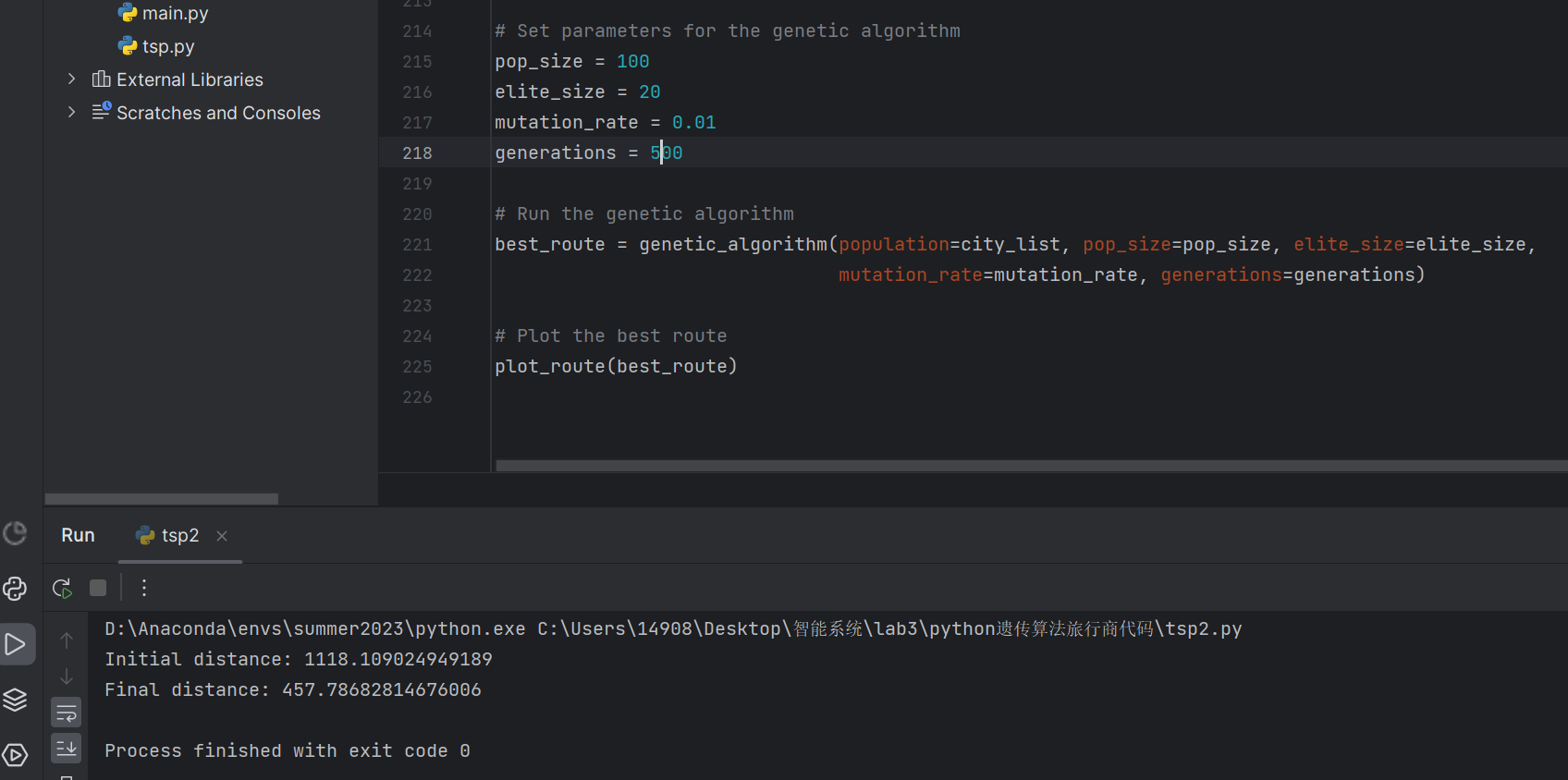
其路径长度为：424.869292

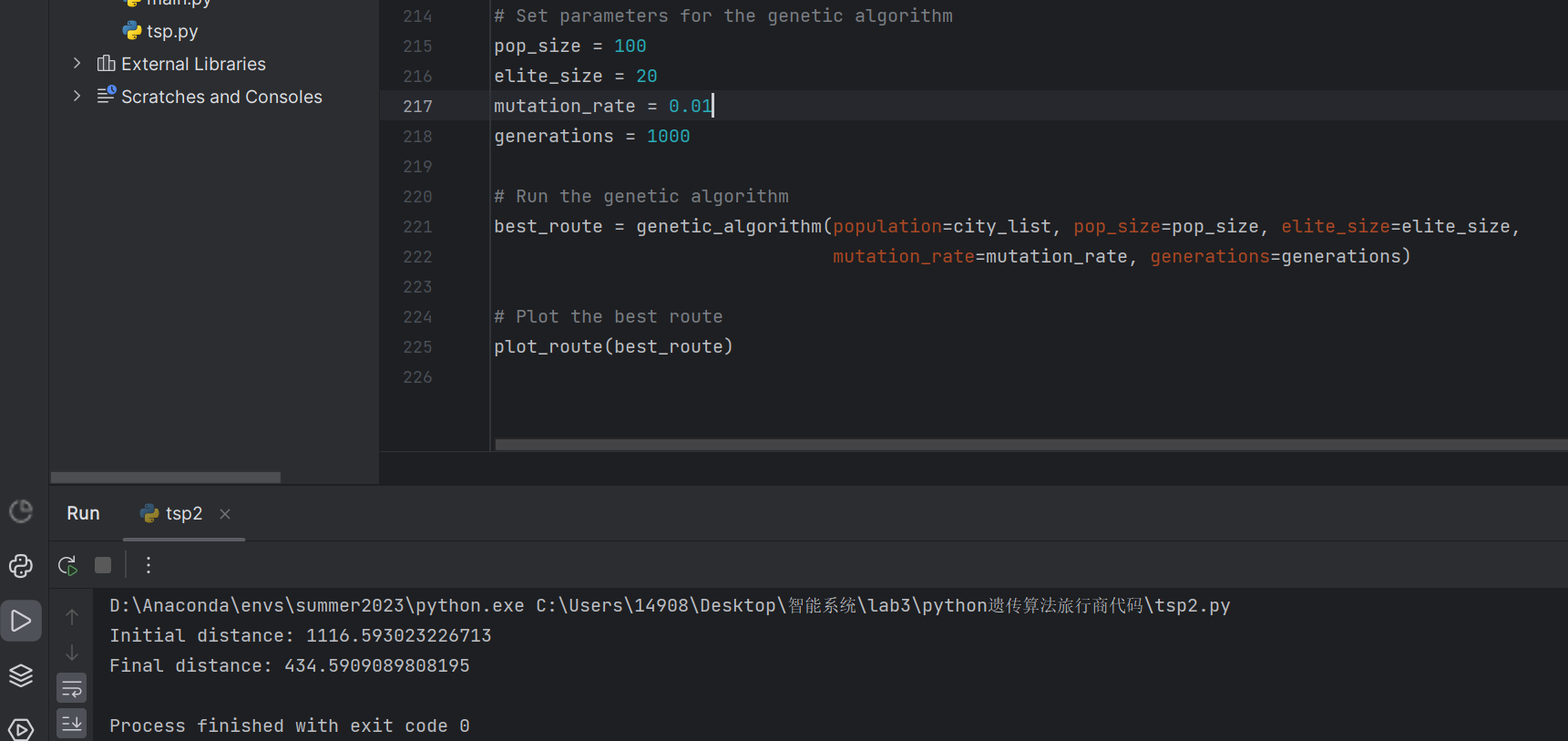
500次迭代，没有到最优

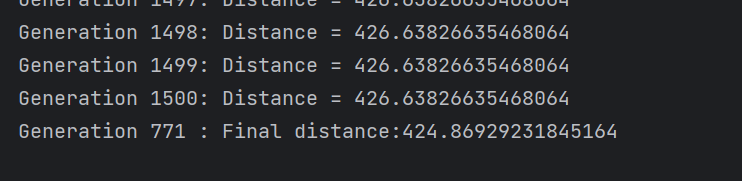












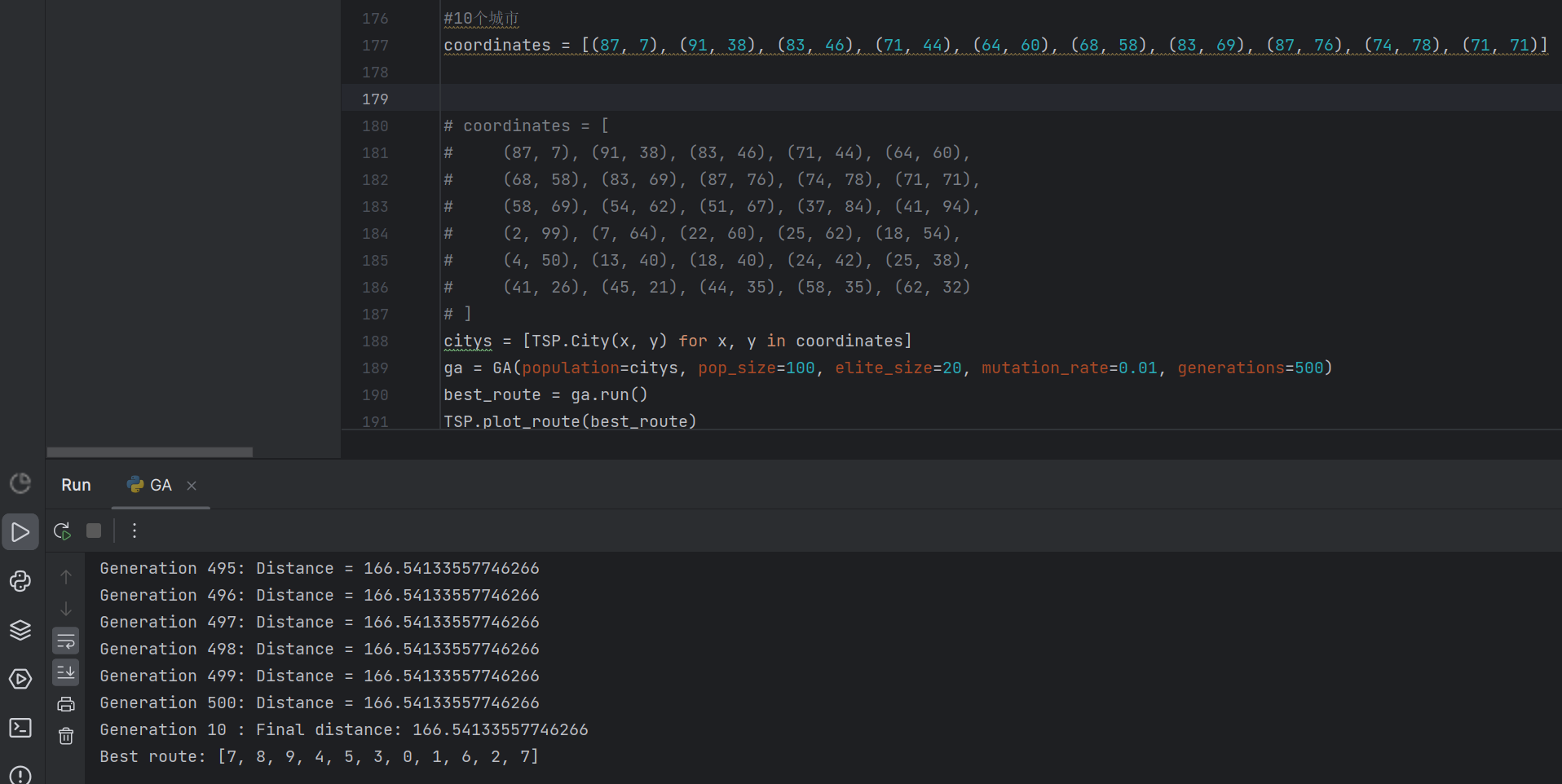
也可取前10个城市的坐标（如表2）进行测试：

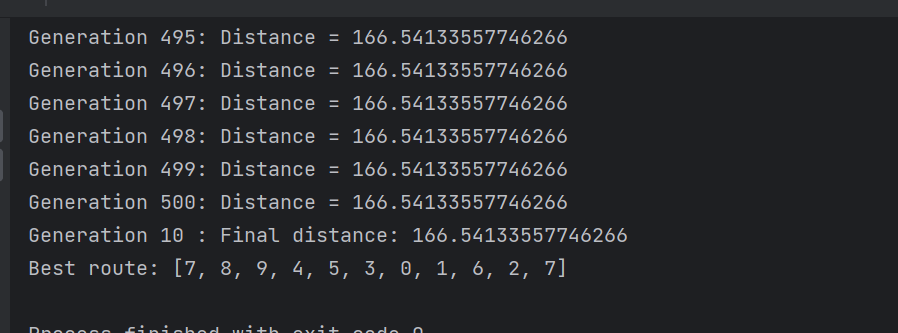
表2  Oliver TSP问题的10个城市位置坐标

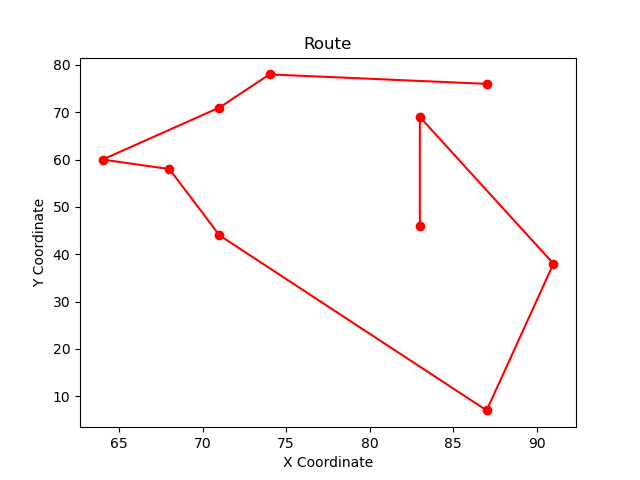
|  |  |
| --- | --- |
| 城市编号 | 坐标 |
| 1 | (87，7) |
| 2 | (91，38) |
| 3 | (83，46) |
| 4 | (71，44) |
| 5 | (64，60) |
| 6 | (68，58) |
| 7 | (83，69) |
| 8 | (87，76) |
| 9 | (74，78) |
| 10 | (71，71) |

有人求得的最优路径为： 0 3 5 4 9 8 7 6 2 1 0

路径长度是166.541336







上述10个城市的求解中编号从0开始，把所有路径搜索完又返回到出发节点。

2 问题描述

应用遗传算法求解30/10个节点的TSP（旅行商问题）问题，求问题的最优解。

**四 实验要求**

1 掌握遗传算法的基本原理、各个遗传操作和算法步骤；

2 要求求出问题最优解，若得不出最优解，请分析原因；

3 对实验中的几个算法控制参数进行仔细定义，并能通过实验选择参数的最佳值；

4 要求界面显示每次迭代求出的局部最优解和最终求出的全局最优解。

**五 实验背景知识**

遗传算法是模仿生物遗传学和自然选择机理，通过人工方式构造的一类优化搜索算法，是对生物进化过程的一种数学仿真，是进化计算的一种最重要形式。遗传算法为那些难以找到传统数学模型的难题找出了一个解决方法。自从Holland于1975年在其著作《Adaptation in Natural and Artificial Systems》中首次提出遗传算法以来，经过近30年的研究，现在已发展到一个比较成熟的阶段，并且在实际中已经得到了很好的应用。

**1 遗传算法基本步骤**

(1) 初始化群体;

(2) 计算群体上每个个体的适应度值;

(3) 按由个体适应度值所决定的某个规则选择将进入下一代的个体;

(4) 按概率Pc进行交叉操作;

(5) 按概率Pm进行变异操作;

(6) 没有满足某种停止条件，则转第(2)步，否则进入(7)；

(7) 输出种群中适应度值最优的染色体作为问题的满意解或最优解。

**2 参数编码**

把待求解问题的解空间中的每个可行解看作一个染色体，并用编码的方式来表示，通常编码中的每一位都看作是一个组成该染色体的基因。在TSP问题中，多采用以遍历城市的次序排列进行编码。如对于8个城市的TSP问题，123456781就表示一个可行解。还有其它编码方法如Grefenstette编码，其他可以查阅相关参考文献。

**3 初始群体**

初始群体是指问题的一组初始可行解，可行解的数量（可定义为变量popsize）和分布对于遗传算法的运行有着很大的影响。实际求解中，初始群体往往采用随机生成的方法。在TSP问题中，随机生成popsize条可行路径序列。

**4 评价函数**

评价函数即适应度函数，在遗传算法中用来计算一个染色体优劣的函数。在进行遗传操作和种群进化的时候，每个染色体的适应值是决定它是否进入下一轮种群进化的关键因素。适应值高的函数被选作新一代个体的可能性就会大。

TSP问题中适应度函数常取路径长度的倒数（或倒数的相关函数），如：



其中，N是个调节参数，根据实验情况进行确定。

**5 选择算子**

赌轮算法是选择算子中常用的一种方法。它的名称来源于赌博中的轮盘赌，轮盘赌是一种随机性赌博游戏，我们这里就是由它的随机性来选择出某些个体，这些个体相对来说具有较优良的适应性。

我们定义f(xi)为第i（i=1,2,3.....popsize）个染色体的适应度，则每个个体被选中的概率

是： 



图 10 赌轮盘示意图

在算法中赌轮选择法可用下面的子过程来模拟：

1. 在[0,1]区间内产生一个均匀分布的伪随机数r。
2. 若r<=q1，则染色体x1被选中。
3. 若，则染色体xk被选中。

其中qi称为染色体xi（i=1，2，...,popsize)的积累概率，其计算公式为：



赌轮选择算子在个体数不太多时，有可能出现不正确反映个体适应度的选择过程，也就是说适应度高的个体有可能反而被淘汰了。为了改进赌轮选择算子的这种缺点，有很多改进的交叉选择算子，如：最佳个体保存法、期望值方法、排序选择方法、联赛选择方法、排挤方法等。

**6 交叉算子**

在自然界生物进化过程中，起核心作用的是生物遗传基因的重组(加上变异)。同样，遗传算法中，起核心作用的是遗传操作的交叉算子。所谓交叉算子就是把两个父代个体的部分结构加以替换重组而生成新个体的操作。通过交叉，遗传算法的搜索能力得以飞跃提高。

交叉算子设计一般与所求解的具体问题有关。下面列举几种在TSP问题中常见的交叉方法：

1. 部分匹配交叉（PMX，partially mapped crossover）

由Goldberg于1985年提出。在PMX操作时，先依据均匀随机分布产生两个位串交叉点，定义这两点之间的区域为匹配区域，并交换两父串的匹配区域。如父串及匹配区域为：

A=9 8 4 | 5 6 7 | 1 3 2 0

B=8 7 1 | 2 3 0 | 9 5 4 6

首先交换A、B的匹配区域，得：

A’=9 8 4 | 2 3 0 | 1 3 2 0

B’=8 7 1 | 5 6 7 | 9 5 4 6

再对A’、 B’两子串匹配区域以外的地方出现的遍历重复，依据匹配区域内的位置映射关系，逐一交换。如A’区域外的2，3，0分别以5，6，7交换，得：

A‘’=9 8 4 | 2 3 0 | 1 6 5 7

B‘’=8 0 1 | 5 6 7 | 9 2 4 3

(2) 顺序交叉（OX，order crossover）

Davis在1985年提出。此方法开始也是选择一个匹配区域：

A=9 8 4 | 5 6 7 | 1 3 2 0

B=8 7 1 | 2 3 0 | 9 5 4 6

根据匹配区域的映射关系，在其匹配区域外的相应位置标记H

A‘=9 8 4 | 5 6 7 | 1 H H H

B’=8 H 1 | 2 3 0 | 9 H 4 H

再移动匹配区域至起点位置，且在其后预留相应于匹配区域的空间（H数目），然后将其余的码按相对次序排列在预留区后面

A‘’=5 6 7 | H H H | 1 9 8 4

B‘’=2 3 0 | H H H | 9 4 8 1

最后将父串A、B的匹配区域互换，并放置到A‘’、B‘’的预留区域，得到子代：

A‘‘’=5 6 7 | 2 3 0 | 1 9 8 4

B‘’’=2 3 0 | 5 6 7 | 9 4 8 1

（3）改进的启发式顺序交叉

与OX法有点类似。

随机在串中选择一个交配区域，如两父串及交配区域选定为：

A = 1 2 | 3 4 5 6 | 7 8 9

B = 9 8 | 7 6 5 4 | 3 2 1

将B的交配区域加到A的前面或后面，A的交配区域加到B的前面或后面得到：

A’=7 6 5 4 |1 2 3 4 5 6 7 8 9

B’=3 4 5 6 |9 8 7 6 5 4 3 2 1

在A’中自交配区域后依次删除与交配区相同的城市码，得到最终的两子串为：

A‘’=7 6 5 4 1 2 3 8 9

B‘’=3 4 5 6 9 8 7 2 1

**7 变异算子**

TSP问题中，经常采取的变异操作主要有：

（1）位点变异

变异仅以一定的概率（通常较小）对串的某些位作值的变异。

（2）逆转变异

在串中，随机选择两点，再将这两点内的子串按反序插入到原位置中，如选择A的你转点为3，6，则经逆转后，变为A’。如

A= 1 2 3 | 4 5 6 |7 8 9

A’=1 2 3 | 6 5 4 |7 8 9

这种变异操作对于TSP问题，就调整前后引起的TSP圈的长度变化而言属于最细微的调整，因而局部优化的精度较高；但码串绝对位置所呈现的“模式”变化较大。

（3）对换变异

随机选择串中的两点，交换其值（码）。对于串A

A= 1 2 3 4 | 5 6 7 | 8 9

若对换点位4，7，则经对换后，A’为：

A’=1 2 3 7|5 6 4 | 8 9

这种变异操作在求解TSP问题优化算法中常被采用。在遗传算法中，对换变异操作对码串绝对位置所呈现的“模式”变化影响较大，所需的计算也简单一些，但局部优化精度稍差一点。

1. 插入变异

从串中随机选择1个码，将此码插入随机选择的插入点中间，对于上述A而言，若取插入码为5，选取插入点位2~3之间，则

A’=1 2 5 3 4 6 7 8 9

此外，还有一些有关上述变异操作的变体形式，如引入连续逆转，进化变异（爬山法）和混合变异等。

**8 Grefenstette编码**

在TSP问题中，以遍历城市的次序进行编码是最自然的一种方式，但是这种编码方法所对应的交叉运算和变异运算实现起来比较困难。Grefenstette等人提出了一种新编回路线。对于一个城市列表V，假定对各个城市的一个访问顺序为T=(t1, t2,…，tn, tn+1）。规定每访问完一个城市，就从未访问城市列表W=V-{t1, t2,…，ti-1}(i=1,2,3,…，n)中将该城市去掉。然后用第i个所访问城市ti在未访问城市列表W中的对应位置序号gi(1≤gi≤n-i+1)表示具体访问哪个城市。如此进行一直到处理完V中所有的城市。将全部gi顺序排列在一起所得到的一个列表G=(g1 g2 g3 …gn）就表示一条巡回路线。

设有7个城市分别为V=(a,b,c,d,e,f,g)，对于如下两条巡回路线：

Tx=(a,d,b,f,g,e,c,a)

Ty=(b,c,a,d,e,f,g,b)

用Grefenstette等人所提出的编码方法，其编码为：

Gx=(1 3 1 3 3 2 1)

Gy=(2 2 1 1 1 1 1)

对于TSP使用Grefenstette编码时，个体基因型和个体表现型之间具有一一对应的关系，也就是它使得经过遗传运算后得到的任意的编码串都对应于一条合法的TSP路径。所以我们就可以用基本遗传算法来求解TSP。于是交叉算子可以使用通常的单点或者多点交叉算子；变异运算也可使用常规的一些变异算子，只是基因座gi(i=1,2,3,…,n)所对应的等位基因值应从{1，2，3，…,n-i+1}中选取。

例如将上面的两个TSP个体编码经过单点交叉（交叉点为5）之后可得两个新个体：

Gx=(1 3 1 3 3 2 1) 单点交叉 G，x=(1 3 1 3 1 1 1)

Gy=(2 2 1 1 1 1 1) G’y=(2 2 1 1 3 2 1)

对它们进行解码处理后，可得到两条新的巡回路线：

T’x=(a,d,b,f,c,e,g,a)

T’y=(b,c,a,d,g,f,e,b)

在设计遗传算子时，一般希望它能够有效遗传个体的重要表现性状。对于TSP使用Grefenstette编码时，编码串中前面基因座上的基因值改变，会对后面基因座上的基因值产生不同解释。所以这里使用单点交叉算子，个体在交叉点之前的性状能够被完全继承下来，而在交叉点之后的性状就改变得相当大。

**9 算法控制参数设定**

（1）N 群体大小，即群体中所含个体的数量，根据具体问题来选择，本实验可取20~100；

（2）T 遗传算法的终止进化代数，本实验可取 100~500；

（3）Pc 交叉概率，它体现了被选择出来进行杂交的个体的比例，一般取0.4~0.9；

（4）Pm 变异概率，它体现了发生变异的个体的比例。一般取0.001~0.1 。

**六 实验关键技术**

遗传算法求解问题时操作算子比较多，所以在求解时要注意分模块分层次地进行编码。程序实现中的几个关键点：

（1）种群规模、进化代数以及交叉和变异概率的定义。

（2）染色体编码方法；

（3）适应度函数定义方法；

（4）选择、交叉和变异这三个遗传操作的定义方法；

为了帮助大家快速掌握该方法，附件中给出一个应用遗传算法求解TSP问题的程序模板。该模板中染色体编码采用的是城市序列的自然排列编码方法。

**七 实验检查要求**

1 界面显示要求

（1）显示求出的最优解；

（2）显示迭代次数及每次迭代求出的局部最优值。

2 代码要求：

要求提供选择、交叉和变异算子的核心代码。

3 讲解要求

要求学生讲解自己设计代码的构架，主要有以下几个要点：（1）染色体编码方法；（2）适应度函数定义方法；（3）选择、交叉和变异这三个遗传操作的定义方法；（4）种群规模、进化代数以及交叉和变异概率的定义；（5）实际求得的最优解，若不是最优解，自己分析可能的原因。