



# 智能控制理论 (Intelligent Control Theories)

## 第八章 遗传算法理论

主讲教师：段朝霞（能电院自动化系）

办公室：勤学楼1212

邮 箱：[duanzx1989@163.com](mailto:duanzx1989@163.com)



遗传算法（Genetic Algorithm，简称GA）作为一种解决复杂问题的优化搜索方法，是由密歇根大学的John Holland教授首先提出来的（Holland，1975）。遗传算法是以达尔文的生物进化论为启发而创建的，是一种基于进化论中优胜劣汰、自然选择、适者生存和物种遗传思想的优化算法。

遗传算法广泛应用于人工智能、机器学习、知识工程、函数优化、自动控制、模式识别、图像处理、生物工程等众多领域。目前，遗传算法正在向其它学科和领域渗透，正在形成遗传算法、神经网络和模糊控制相结合，从而构成一种新型的智能控制系统整体优化的结构形式。



## 8.1 遗传算法的基本原理

遗传算法简称 GA (Genetic Algorithms) 是1962年由美国 Michigan 大学的 Holland 教授提出的模拟自然界遗传机制和生物进化论而成的一种并行随机搜索最优化方法。



GA 的基本思想来源于 Darwin 的进化论和 Mendel 的遗传学说。Darwin 的进化论认为每一物种在不断的发展过程中都是越来越适应环境。物种的每个个体的基本特征被后代所继承，但后代又不完全同于父代，这些新的变化，若适应环境，则被保留下来。在某一环境中也是那些更能适应环境的个体特征能被保留下来，这就是适者生存的原理。

## 相关术语:

**基因型 (genotype):** 性状染色体的内部表现。

**表现型 (phenotype):** 染色体决定的性状的外部表现，或者说，根据基因型形成的个体的外部表现。

**进化 (evolution):** 种群逐渐适应生存环境，品质不断得到改良。生物的进化是以种群的形式进行的。

**适应度 (fitness):** 度量某个物种对于生存环境的适应程度。

**选择 (selection):** 以一定的概率从种群中选择若干个个体。一般，选择过程是一种基于适应度的优胜劣汰的过程。

**复制 (reproduction):** 细胞分裂时，遗传物质DNA通过复制而转移到新产生的细胞中，新细胞就继承了旧细胞的基因。



## 相关术语（续一）：

**交叉(crossover):** 两个染色体的某一相同位置处DNA被切断，前后两串分别交叉组合形成两个新的染色体。也称基因重组或杂交。

**变异(mutation):** 复制时可能（很小的概率）产生某些复制差错，变异产生新的染色体，表现出新的性状。

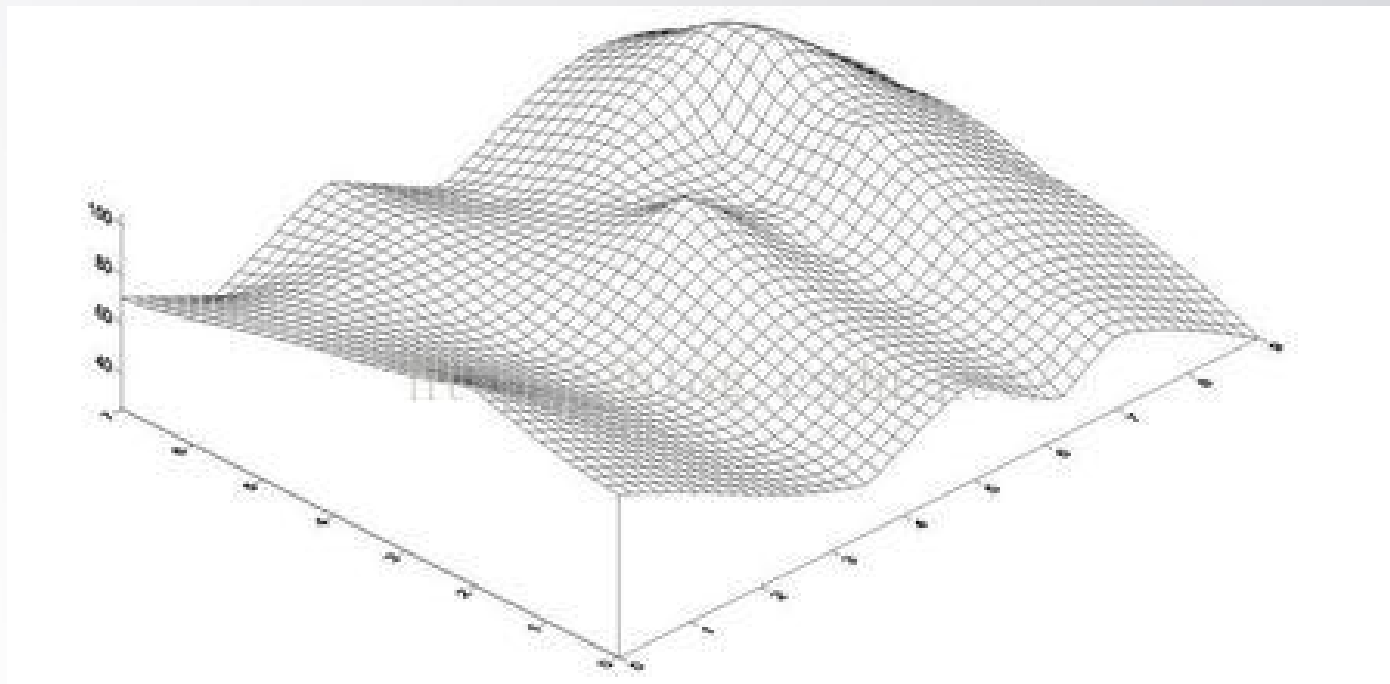
**编码(coding):** DNA中遗传信息在一个长链上按一定的模式排列。遗传编码可看作从表现型到基因型的映射。

**解码(decoding):** 基因型到表现型的映射。

**个体(individual)：** 指染色体带有特征的实体。

**种群(population)：** 个体的集合，该集合内个体数称为种群的大小。

可以把遗传算法的过程看作是一个在多元函数里面求最优解的过程。可以这样想象，这个多维曲面里面有数不清的“山峰”，而这些山峰所对应的就是局部最优解。而其中也会有一个“山峰”的海拔最高的，那么这个就是全局最优解。而遗传算法的任务就是尽量爬到最高峰，而不是陷落在一些小山峰。



# “袋鼠跳”问题

既然我们把函数曲线理解成一个一个山峰和山谷组成的山脉。那么我们可以设想所得到的每一个解就是一只袋鼠，我们希望它们不断的向着更高处跳去，直到跳到最高的山峰。所以求最大值的过程就转化成一个“袋鼠跳”的过程。

## 1. 爬山法（最速上升爬山法）：

从搜索空间中随机产生邻近的点，从中选择对应解最优的个体，替换原来的个体，不断重复上述过程。

因为爬山法只对“邻近”的点作比较，所以目光比较“短浅”，常常只能收敛到离开初始位置比较近的局部最优解上面。对于存在很多局部最优点的问题，通过一个简单的迭代找出全局最优解的机会非常渺茫。



## 2. 遗传算法

模拟物竞天择的生物进化过程，通过维护一个潜在解的群体执行了多方向的搜索，并支持这些方向上的信息构成和交换。是以面为单位的搜索，比以点为单位的搜索，更能发现全局最优解。

在遗传算法中，有很多袋鼠，它们降落到喜马拉雅山脉的任意地方。这些袋鼠并不知道它们的任务是寻找珠穆朗玛峰。但每过几年，就在一些海拔高度较低的地方射杀一些袋鼠，并希望存活下来的袋鼠是多产的，在它们所处的地方生儿育女。

# 遗传算法的实现过程

遗传算法的实现过程实际上就像自然界的进化过程那样。

首先，寻找一种对问题潜在解进行“数字化”编码的方案。（建立表现型和基因型的映射关系）

然后，用随机数初始化一个种群（那么第一批袋鼠就被随意地分散在山脉上），种群里面的个体就是这些数字化的编码。

接下来，通过适当的解码过程之后（得到袋鼠的位置坐标），用适应性函数对每一个基因个体作一次适应度评估（袋鼠爬得越高，越是受我们的喜爱，所以适应度相应越高）。

# 遗传算法的实现过程

**选择：** 用选择函数按照某种规定择优选择（我们要每隔一段时间，在山上射杀一些所在海拔较低的袋鼠，以保证袋鼠总体数目持平。）。

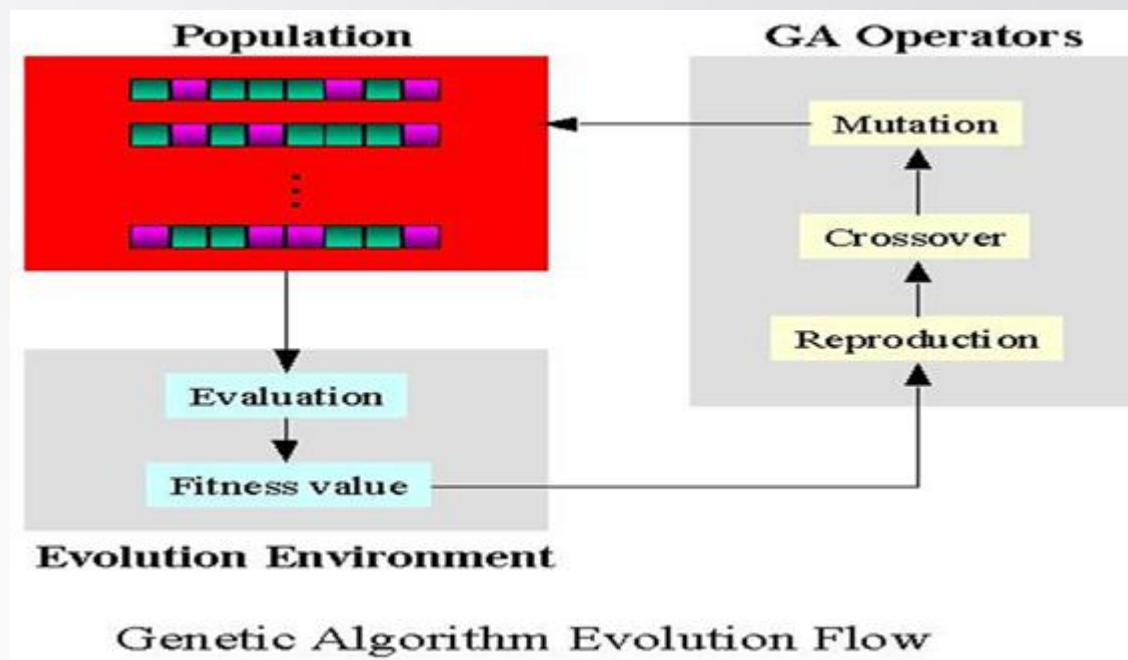
**变异：** 让个体基因变异（让袋鼠随机地跳一跳）。然后产生子代（希望存活下来的袋鼠是多产的，并在那里生儿育女）。

遗传算法并不保证你能获得问题的最优解，但是使用遗传算法的最大优点在于你不必去了解 and 操心如何去“找”最优解。（你不必去指导袋鼠向那边跳，跳多远。）而只要简单的“否定”一些表现不好的个体就行了。

**把那些总是爱走下坡路的袋鼠射杀，这就是遗传算法的精粹！**

## 遗传算法的步骤

1. 评估每条染色体所对应个体的适应度。
2. 遵照适应度越高，选择概率越大的原则，从种群中选择两个个体作为父方和母方。
3. 抽取父母双方的染色体，进行交叉，产生子代。
4. 对子代的染色体进行变异。
5. 重复2，3，4步骤，直到新种群的产生。



## 编制袋鼠的染色体-----基因的编码方式

如何为袋鼠的染色体编码呢？因为编码的目的是建立表现型到基因型的映射关系，而表现型一般就被理解为个体的特征。所以我们要想为“袋鼠”的染色体编码，我们必须先来考虑“袋鼠”的“个体特征”是什么。

我们由始至终都只关心一件事情：袋鼠在哪里。因为只要我们知道袋鼠在那里，我们就能做两件必须去做的事情：

- （1）通过查阅喜马拉雅山脉的地图来得知袋鼠所在的海拔高度（通过自变量求适应函数的值。）以判断我们有没有必要把它射杀。
- （2）知道袋鼠跳一跳（交叉和变异）后去到哪个新位置。



既然确定了袋鼠的位置作为个体特征，具体来说位置就是横坐标。那么接下来，我们就要建立表现型到基因型的映射关系。就是说如何用编码来表现出袋鼠所在的横坐标。由于横坐标是一个实数，要对这个实数编码。

- 二进制编码
- 浮点数编码
- 符号编码



## 物竞天择——适应性评分与及选择函数。

### 物竞——适应度函数 (*fitness function*)

自然界生物竞争过程往往包含两个方面：**生物相互间的搏斗、生物与客观环境的搏斗过程。**

直接用袋鼠的海拔高度作为它们的适应性评分。即适应度函数直接返回函数值就行了。

### 天择——选择函数 (*selection*)

自然界中，越适应的个体就越有可能繁殖后代。但是也不能说适应度越高的就肯定后代越多，只能是从概率上来说更多。（毕竟有些所处海拔高度较低的袋鼠很幸运，逃过了你的眼睛。）那么我们怎么来建立这种概率关系呢？

## 选择概率

例 8-1： 有5条染色体，他们所对应的适应度评分分别为：  
**5,7,10,13,15**

所以累计总适应度为：

$$F = \sum_{i=1}^n f_i = 5 + 7 + 10 + 13 + 15 = 50$$

所以各个个体被选中的概率分别为：

$$P_1 = \frac{f_1}{F} = 10\%, P_2 = \frac{f_2}{F} = 14\%,$$

$$P_3 = \frac{f_3}{F} = 20\%, P_4 = \frac{f_4}{F} = 26\%,$$

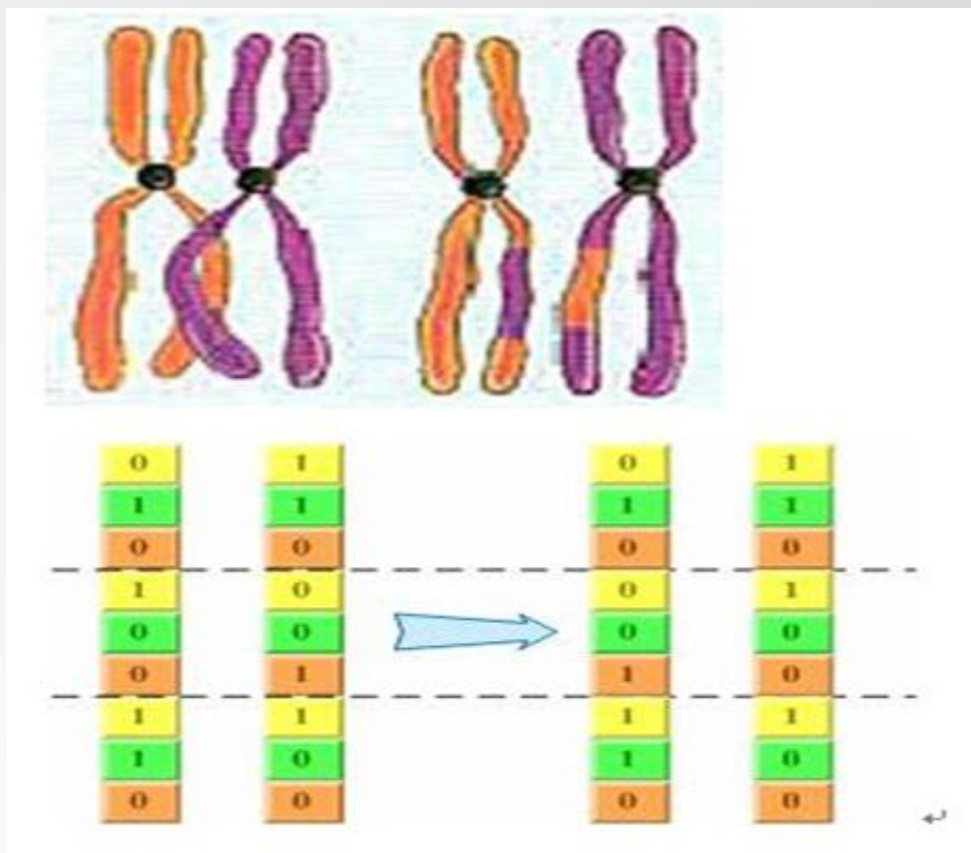
$$P_5 = \frac{f_5}{F} = 30\%$$



# 遗传变异——基因重组（交叉）与基因突变

这两个步骤就是使得子代不同于父代的根本原因（注意，并不一定子代优于父代，只有经过自然的选择后，才会出现子代优于父代的倾向。）。

**交叉：**随机把其中几个位于同一位置的编码进行交换，产生新的个体。





**基因突变过程：**基因突变是染色体的某一个位点上基因的改变。基因突变使一个基因变成它的等位基因，并且通常会引起一定的表现型变化。

二进制编码的遗传操作过程和生物学中的过程非常相类似，基因串上的“0”或“1”有一定几率变成与之相反的“1”或“0”。

例如下面这串二进制编码：

**101101001011001**

经过基因突变后，可能变成以下这串新的编码：

**001101011011001**



遗传算法将“优胜劣汰，适者生存”的生物进化原理引入优化参数形成的编码串联群体中，按所选择的适应度函数并通过遗传中的复制、交叉及变异对个体进行筛选，使适应度高的个体被保留下来，组成新的群体，新的群体既继承了上一代的信息，又优于上一代。这样周而复始，群体中个体适应度不断提高，直到满足一定的条件。遗传算法的算法简单，可并行处理，并能到全局最优解。



# 遗传算法的基本操作为：

## (1) 复制 (Reproduction Operator)

复制是从一个旧种群中选择生命力强的个体位串产生新种群的过程。具有高适应度的位串更有可能在下一代中产生一个或多个子孙。

复制操作可以通过随机方法来实现。首先产生 $0 \sim 1$ 之间均匀分布的随机数，若某串的复制概率为40%，则当产生的随机数在 $0.40 \sim 1.0$ 之间时，该串被复制，否则被淘汰。



## (2) 交叉 (Crossover Operator)

复制操作能从旧种群中选择出优秀者，但不能创造新的染色体。而交叉模拟了生物进化过程中的繁殖现象，通过两个染色体的交换组合，来产生新的优良品种。

交叉的过程为：在匹配池中任选两个染色体，随机选择一点或多点交换点位置；交换双亲染色体交换点右边的部分，即可得到两个新的染色体数字串。



交换体现了自然界中信息交换的思想。交叉有一点交叉、多点交叉、还有一致交叉、顺序交叉和周期交叉。一点交叉是最基本的方法，应用较广。它是指染色体切断点有一处，例：

$$A: 101100 \ 1110 \rightarrow 101100 \ 0101$$
$$B: 001010 \ 0101 \rightarrow 001010 \ 1110$$



### (3) 变异 (Mutation Operator)

变异运算用来模拟生物在自然的遗传环境中由于各种偶然因素引起的基因突变，它以很小的概率随机地改变遗传基因（表示染色体的符号串的某一位）的值。在染色体以二进制编码的系统中，它随机地将染色体的某一个基因由1变为0，或由0变为1。



若只有选择和交叉，而没有变异，则无法在初始基因组合以外的空间进行搜索，使进化过程在早期就陷入局部解而进入终止过程，从而影响解的质量。为了在尽可能大的空间中获得质量较高的优化解，必须采用变异操作。






1. 选择的作用：优胜劣汰，适者生存；
2. 交叉的作用：保证种群的稳定性，朝着最优解的方向进化；
3. 变异的作用：保证种群的多样性，避免交叉可能产生的局部收敛。

## 8.2 遗传算法的特点

(1) 遗传算法是对参数的编码进行操作，而非对参数本身，这就是使得我们在优化计算过程中可以借鉴生物学中染色体和基因等概念，模仿自然界中生物的遗传和进化等机理；

(2) 遗传算法同时使用多个搜索点的搜索信息。传统的优化方法往往是从解空间的单个初始点开始最优解的迭代搜索过程，单个搜索点所提供的信息不多，搜索效率不高，有时甚至使搜索过程局限于局部最优解而停滞不前。



遗传算法从由很多个体组成的一个初始群体开始最优解的搜索过程，而不是从一个单一的个体开始搜索，这是遗传算法所特有的一种隐含并行性，因此遗传算法的搜索效率较高。

(3) 遗传算法直接以目标函数作为搜索信息。传统的优化算法不仅需要利用目标函数值，而且需要目标函数的导数值等辅助信息才能确定搜索方向。而遗传算法仅使用由目标函数值变换来的适应度函数值，就可以确定进一步的搜索方向和搜索范围，无需目标函数的导数值等其他一些辅助信息。



遗传算法可应用于目标函数无法求导数或导数不存在的函数的优化问题，以及组合优化问题等。

(4) 遗传算法使用概率搜索技术。遗传算法的选择、交叉、变异等运算都是以一种概率的方式来进行的，因而遗传算法的搜索过程具有很好的灵活性。随着进化过程的进行，遗传算法新的群体会更多地产生出许多新的优良的个体。



(5) 遗传算法在解空间进行高效启发式搜索，而非盲目地穷举或完全随机搜索；

(6) 遗传算法对于待寻优的函数基本无限制，它既不要求函数连续，也不要求函数可微，既可以是数学解析式所表示的显函数，又可以是映射矩阵甚至是神经网络的隐函数，因而应用范围较广；

(7) 遗传算法具有并行计算的特点，因而可通过大规模并行计算来提高计算速度，适合大规模复杂问题的优化。

## 8.3 遗传算法的应用

### (1) 函数优化

函数优化是遗传算法的经典应用领域，也是遗传算法进行性能评价的常用算例。尤其是对非线性、多模型、多目标的函数优化问题，采用其他优化方法较难求解，而遗传算法却可以得到较好的结果。

### (2) 组合优化

随着问题的增大，组合优化问题的搜索空间也急剧扩大，采用传统的优化方法很难得到最优解。遗传算法是寻求这种满意解的最佳工具。例如，遗传算法已经在求解旅行商问题、背包问题、装箱问题、图形划分问题等方面得到成功的应用。



### (3) 生产调度问题

在很多情况下，采用建立数学模型的方法难以对生产调度问题进行精确求解。在现实生产中多采用一些经验进行调度。遗传算法是解决复杂调度问题的有效工具，在单件生产车间调度、流水线生产车间调度、生产规划、任务分配等方面遗传算法都得到了有效的应用。



#### (4) 自动控制

在自动控制领域中有很多与优化相关的问题需要求解，遗传算法已经在其中得到了初步的应用。例如，利用遗传算法进行控制器参数的优化、基于遗传算法的模糊控制规则的学习、基于遗传算法的参数辨识、基于遗传算法的神经网络结构的优化和权值学习等。





## (5) 机器人

例如，遗传算法已经在移动机器人路径规划、关节机器人运动轨迹规划、机器人结构优化和行为协调等方面得到研究和应用。

## (6) 图像处理

遗传算法可用于图像处理过程中的扫描、特征提取、图像分割等的优化计算。目前遗传算法已经在模式识别、图像恢复、图像边缘特征提取等方面得到了应用。



## (7) 人工生命

人工生命是用计算机、机械等人工媒体模拟或构造出的具有生物系统特有行为的人造系统。人工生命与遗传算法有着密切的联系，基于遗传算法的进化模型是研究人工生命现象的重要基础理论。遗传算法为人工生命的研究提供了一个有效的工具。

## (8) 遗传编程

遗传算法已成功地应用于人工智能、机器学习等领域的编程。



## (9) 机器学习

基于遗传算法的机器学习在很多领域都得到了应用。例如，采用遗传算法实现模糊控制规则的优化，可以改进模糊系统的性能；遗传算法可用于神经网络连接权的调整和结构的优化；采用遗传算法设计的分类器系统可用于学习式多机器人路径规划。

## 8.4 遗传算法的设计

### 8.4.1 遗传算法的构成要素

#### (1) 染色体编码方法


基本遗传算法使用固定长度的二进制符号来表示群体中的个体，其等位基因是由二值符号集{0,1}所组成。初始个体基因值可用均匀分布的随机值生成，如

$$x = 100111001000101101$$

就可表示一个个体，该个体的染色体长度是18。



(2) 个体适应度评价：基本遗传算法与个体适应度成正比的概率来决定当前群体中每个个体遗传到下一代群体中的概率多少。为正确计算这个概率，要求所有个体的适应度必须为正数或零。因此，必须先确定由目标函数值  $J$  到个体适应度  $f$  之间的转换规则。



(3) 遗传算子：基本遗传算法使用下述三种遗传算子：

① 选择运算：使用比例选择算子；

② 交叉运算：使用单点交叉算子；

③ 变异运算：使用基本位变异算子或均匀变异算子。



#### (4) 基本遗传算法的运行参数

有下述4个运行参数需要提前设定：

**M:** 群体大小，即群体中所含个体的数量，一般取为20~100；

**G:** 遗传算法的终止进化代数，一般取为100~500；

**P<sub>c</sub>:** 交叉概率，一般取为0.4~0.99；

**P<sub>m</sub>:** 变异概率，一般取为0.0001~0.1。

## 8.4.2 遗传算法的应用步骤



对于一个需要进行优化的实际问题，一般可按下述步骤构造遗传算法：

**第一步：**确定决策变量及各种约束条件，即确定出个体的表现型 $X$ 和问题的解空间；

**第二步：**建立优化模型，即确定出目标函数的类型及数学描述形式或量化方法；





第三步：确定表示可行解的染色体编码方法，即确定出个体的基因型 $x$ 及遗传算法的搜索空间；

第四步：确定个体适应度的量化评价方法，即确定出由目标函数值 $J(x)$ 到个体适应度函数 $F(x)$ 的转换规则；

第五步：设计遗传算子，即确定选择运算、交叉运算、变异运算等遗传算子的具体操作方法；

第六步：确定遗传算法的有关运行参数，即  $M, G, P_c, P_m$  等参数；

第七步：确定解码方法，即确定出由个体表现型 $X$ 到个体基因型 $x$ 的对应关系或转换方法。



以上操作过程可以用图8-1来表示。

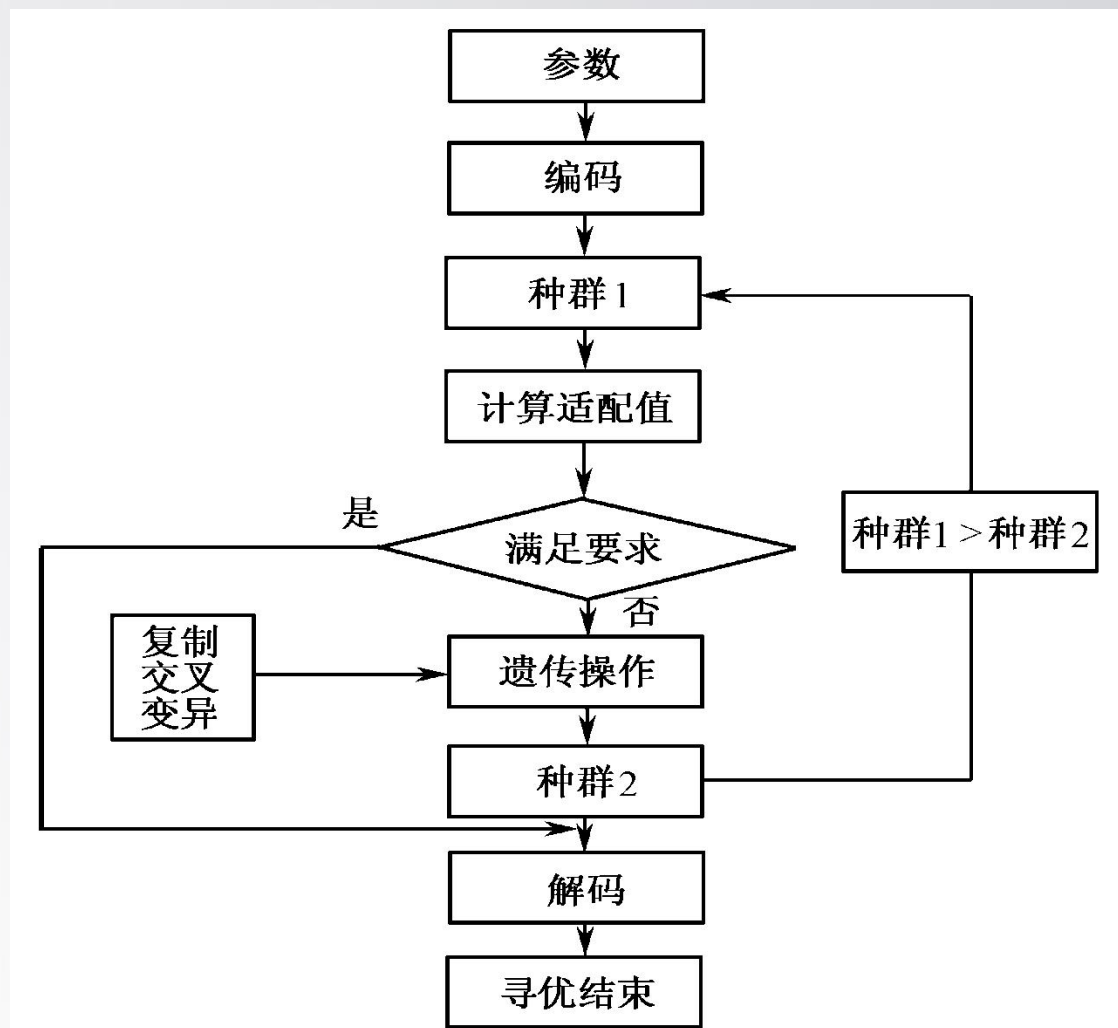


图8-1 遗传算法流程图

## 8.5 遗传算法求函数极大值

例8-2： 利用遗传算法求 **Rosenbrock** 函数的极大值

$$\begin{cases} f(x_1, x_2) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2 \\ -2.048 \leq x_i \leq 2.048 \quad (i = 1, 2) \end{cases}$$



函数  $f(x_1, x_2)$  的三维图如图8-2所示，可以发现该函数在指定的定义域上有两个接近的极点，即一个全局极大值和一个局部极大值。因此，采用寻优算法求极大值时，需要避免陷入局部最优解。

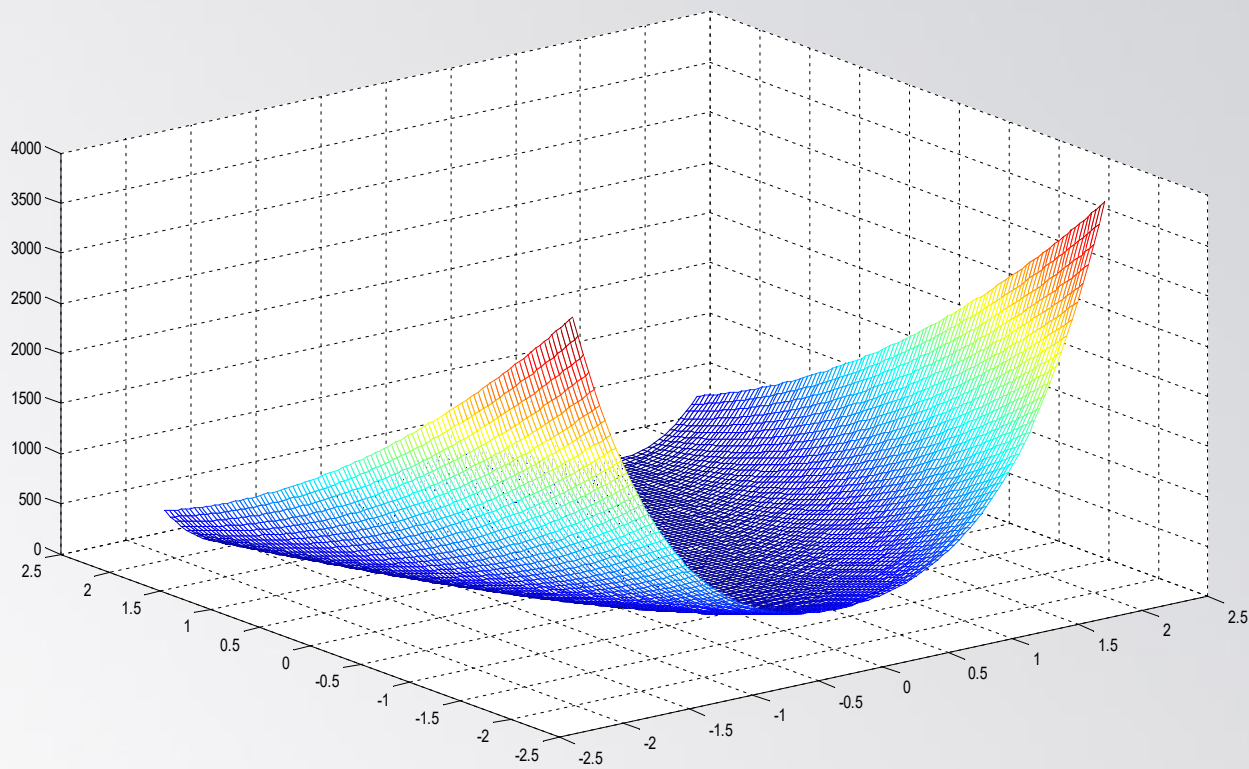



图 8-2  $f(x_1, x_2)$  的三维图



采用二进制编码遗传算法求函数极大值求解该问题遗传算法的构造过程：

- (1) 确定决策变量和约束条件；
- (2) 建立优化模型；
- (3) 确定编码方法



用长度为10位的二进制编码串来分别表示两个决策变量 $x_1, x_2$ 。10位二进制编码串可以表示从0到1023之间的1024个不同的数，故将 $x_1, x_2$ 的定义域离散化为1023个均等的区域，包括两个端点在内共有1024个不同的离散点。

从离散点-2.048到离散点2.048，分别对应于从0000000000(0)到1111111111(1023)之间的二进制编码。



将 $x_1$ ,  $x_2$ 分别表示的两个10位长的二进制编码串连接在一起, 组成一个20位长的二进制编码串, 它就构成了这个函数优化问题的染色体编码方法。使用这种编码方法, 解空间和遗传算法的搜索空间就具有一一对应的关系。

例如:  $x$ : 0000110111 1101110001 表示一个个体的基因型, 其中前10位表示 $x_1$ , 后10位表示 $x_2$ 。






(4) 确定解码方法：解码时需要将20位长的二进制编码串切断为两个10位长的二进制编码串，然后分别将它们转换为对应的十进制整数代码，分别记为 $y_1$ 和 $y_2$ 。

依据个体编码方法和对定义域的离散化方法可知，将代码 $y_i$ 转换为变量 $x_i$ 的解码公式为

$$x_i = 4.096 \times \frac{y_i}{1023} - 2.048 \quad (i = 1, 2) \quad (8-1)$$

例如，对个体  $x: 0000110111 \quad 1101110001$



它由两个代码所组成

$$y_1 = 55, y_2 = 881$$

上述两个代码经过解码后，可得到两个实际的值

$$x_1 = -1.828, x_2 = 1.476$$

(5) 确定个体评价方法：由于Rosenbrock函数的值域总是非负的，并且优化目标是求函数的最大值，故可将个体的适应度直接取为对应的目标函数值，即

$$F(x) = f(x_1, x_2) \quad (8-2)$$

选个体适应度的倒数作为目标函数

$$J(x) = \frac{1}{F(x)} \quad (8-3)$$

(6) 设计遗传算子：选择运算使用比例选择算子，交叉运算使用单点交叉算子，变异运算使用基本位变异算子。

(7) 确定遗传算法的运行参数：群体大小 $M=80$ ，终止进化代数 $G=100$ ，交叉概率 $P_c=0.60$ ，变异概率 $P_m=0.10$ 。

上述七个步骤构成了用于求函数极大值的优化计算基本遗传算法。

采用上述方法进行仿真，经过**100**步迭代，最佳样本为

$$\text{Best}S = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$$

即当  $x_1 = -2.0480$  ,  $x_2 = -2.0480$  时，**Rosenbrock** 函数具有极大值，极大值为**3905.9**。

仿真程序：**chap8\_1.m**



遗传算法的优化过程是目标函数 $J$ 和适应度函数 $F$ 的变化过程。

由仿真结果可知，随着进化过程的进行，群体中适应度较低的一些个体被逐渐淘汰掉，而适应度较高的一些个体会越来越多，并且它们都集中在所求问题的最优点附近，从而搜索到问题的最优解。



## 8.6 基于遗传算法的TSP问题优化

**Hopfield**网络已经对旅行商问题进行了描述。遗传算法由于其全局搜索的特点，在解决**TSP**问题中有明显的优势。

### 8.6.1 TSP问题的编码

设  $\mathbf{D} = \{d_{ij}\}$  是由城市*i*和城市*j*之间的距离组成的距离矩阵，旅行商问题就是求出一条通过所有城市且每个城市只通过一次的具有最短距离的回路。



在旅行商问题的各种求解方法中，描述旅行路线的方法主要有如下两种：

- (1) 巡回旅行路线经过的连接两个城市的路线的顺序排列；
- (2) 巡回旅行路线所经过的各个城市的顺序排列。

大多数求解旅行商问题的遗传算法是以后者为描述方法的，它们大多采用所遍历城市的顺序来表示各个个体的编码串，其等位基因为N个整数值或N个记号。

以城市的遍历次序作为遗传算法的编码，目标函数取路径长度。在群体初始化、交叉操作和变异操作中考虑TSP问题的合法性约束条件（即对所有的城市做到不重不漏）。



## 8.6.2 TSP问题的遗传算法设计

采用遗传算法进行路径优化，分为以下几步：

### 第一步：参数编码和初始群体设定

一般来说遗传算法对解空间的编码大多采用二进制编码形式，但对于 **TSP** 一类排序问题，采用对访问城市序列进行排列组合的方法编码，即某个巡回路径的染色体个体是该巡回路径的城市序列。

针对 **TSP** 问题，编码规则通常是 **N** 进制编码，即每个基因仅从 **1** 到 **N** 的整数里面取一个值，每个个体的长度为 **N**，**N** 为城市总数。定义一个 **s** 行 **t** 列的 **pop** 矩阵来表示群体，**t** 为城市个数 **N+1**，即 **N+1**，**s** 为样本中个体数目。针对 **30** 个城市的 **TSP** 问题，**t** 取值 **31**，





即矩阵每一行的前30个元素表示经过的城市编号，最后一个元素表示经过这些城市要走的距离。

参数编码和初始群体设定程序为：

- `pop=zeros(s,t);`
- `for i=1:s`
- `pop(i,1:t-1)=randperm(t-1);`
- `end`

## 第二步：计算路径长度的函数设计

在 TSP 的求解中，用距离的总和作为适应度函数，来衡量求解结果是否最优。将 POP 矩阵中每一行表示经过的距离的最后一个元素作为路径长度。



两个城市  $m$  和  $n$  间的距离为：

$$d_{mn} = \sqrt{(x_m - x_n)^2 + (y_m - y_n)^2} \quad (8-4)$$

用于计算路径长度的程序为chap8\_2dis.m。

通过样本的路径长度可以得到目标函数和自适应度函数。根据 $t$ 的定义，两两城市组合数共有  $t-2$  组，则目标函数为：

$$J(t) = \sum_{j=1}^{t-2} d(j) \quad (8-5)$$

自适应度函数取目标函数的倒数，即：

$$f(t) = \frac{1}{J(t)} \quad (8-6)$$



### 第三步： 计算选择算子

选择就是从群体中选择优胜个体、淘汰劣质个体的操作，它是建立在群体中个体适应度评估基础上。仿真中采用最优保存方法，即将群体中适应度最大的 **c** 个个体直接替换适应度最小的 **c** 个个体。选择算子函数为**chap8\_2select.m**。

### 第四步： 计算交叉算子

交叉算子在遗传算法中起着核心的作用，它是指将个体进行两两配对，并以交叉概率将配对的父代个体加以替换重组而生成新个体的操作。如果当前随机值大于  $p_c$ ，则随机选择两个个体进行交叉。

有序交叉法实现的步骤是：



有序交叉法实现的步骤是:

步骤 1 随机选取两个交叉点crosspoint(1)和crosspoint(2);

步骤 2 两后代  $X'_1$  和  $X'_2$  先分别按对应位置复制双亲X1和X2匹配段中的两个子串A1和B1;

步骤 3 在对应位置交换X1和X2双亲匹配段A1和B1以外的城市, 如果交换后, 后代  $X'_1$  中的某一城市a与  $X'_1$  子串中A1的城市重复, 则在子串B1中找到与子串A1中城市a对应位置处的城市b,并用城市b取代城市a。如果城市b与  $X'_1$  子串A1中的城市还重复, 则在子串B1中找到与子串A1中b处对应位置处的城市c, 并用城市c取代城市b, 直到  $X'_1$  中的城市均不重复为止, 对后代  $X'_2$  也采用同样方法, 如图8-4所示:



图8-4 有序交叉算子




以图8-4例，具体的有序交叉法实现步骤为：

- (1) 确定两个交叉点的位置分别为2和7；
- (2) 两后代 $X_1'$ 和 $X_2'$ 分别按对应位置复制双亲 $X_1$ 和 $X_2$ 匹配段中两个子串 $A_1$ 和 $B_1$ ，在对应位置交换 $X_1$ 和 $X_2$ 双亲匹配段 $A_1$ 和 $B_1$ 以外的城市，得到

$$X_1' : 8\ 7\ |\ 4\ 5\ 6\ 7\ 1\ |\ 9\ 6\ 5。$$

- (3) 可以看到 $X_1'$ 中的位置1处的元素8与 $A_1$ 中没有重复，故位置1为8，位置2处的元素7与 $A_1$ 中的第4个元素有重复，则用 $B_1$ 中的第4个元素3来代替，且元素3与 $A_1$ 中的元素没有重复，故 $X_1'$ 中的位置2处的元素为城市3，此时 $X_1' : 8\ 3\ |\ 4\ 5\ 6\ 7\ 1\ |\ 9\ 6\ 5。$



(4)  $X_1'$ 中第8个元素为9，此元素与A1中的元素没有重复，故 $X_1'$ 中的位置8处为城市9， $X_1'$ 中的第9个元素为6，与A1中的第3个元素重复，则用B1中的第3个元素0取代，且0不与A1中的元素重复，则 $X_1'$ 中第9个元素为城市0，此时  $X_1'$  : 8 3 | 4 5 6 7 1 | 9 0 5。

(5)  $X_1'$ 中第10个元素为5，此元素与A1中的位置2处的元素重复，则用B1中的位置2处的元素4取代，元素4仍与A1中的位置1处的元素重复，则用B1中位置1处的元素1取代，元素1仍与A1中的位置5处的元素重复，则用B1中位置5处的元素2取代，此时不与A1中的元素重复，故  $X_1'$ 中的第10个元素为城市2。

重复检查完成后，得到后代  $X_1'$ ，即

$X_1'$  : 8 3 | 4 5 6 7 1 | 9 0 2。至此  $X_1'$  交叉完毕。同理可得后代  $X_2'$ ，即 $X_2'$  : 9 8 | 1 4 0 3 2 | 7 5 6。



从图8-4可知，有序交叉算子能够有效地继承双亲的部分基因成分，达到了进化过程中的遗传功能，使该算法并不是盲目搜索，而是趋向于使群体具有更多的优良基因，最后实现寻优的目的。交叉算子函数为

chap8\_2corss.m



## 第五步：计算变异算子

变异操作是以变异概率 $P_m$ 对群体中个体串某些基因位上的基因值作变动，若变异后子代的适应度值更加优异，则保留子代染色体，否则，仍保留父代染色体。

这里采用倒置变异法：假设当前个体 $X$ 为(1 3 7 4 8 0 5 9 6 2)，如果当前随机概率值大于  $P_m$ ，则随机选择来自同一个体的两个点 **mutatepoint(1)**和**mutatepoint(2)**，然后倒置该两个点的中间部分，产生新的个体。

- 例如，假设随机选择个体 $X$ 的两个点“7”和“9”，则倒置该两个点的中间部分，即将“4805”变为“5084”，产生新的个体 $X$ 为(1 3 7 5 0 8 4 9 6 2)。变异算子函数为**chap8\_2mutate.m**。



### 8.6.3 仿真实例

分别以8个城市和30个城市的路径优化为例，其城市路径坐标保存在当前路径的文件cities8.txt和cities30.txt中。

8个城市优化时，遗传算法参数设定为：群体中个体数目 $S=30$ ，交叉概率  $P_c=0.10$ ，变异概率 $P_m=0.80$ 。通过改变进化代数为 $k$ ，观察不同进化代数下路径的优化情况，经过50次进化，城市组合路径达到最小。最短路程为2.8937，如图8-5所示。仿真过程表明，在100次仿真实验中，有98次以上可收敛到最优解。

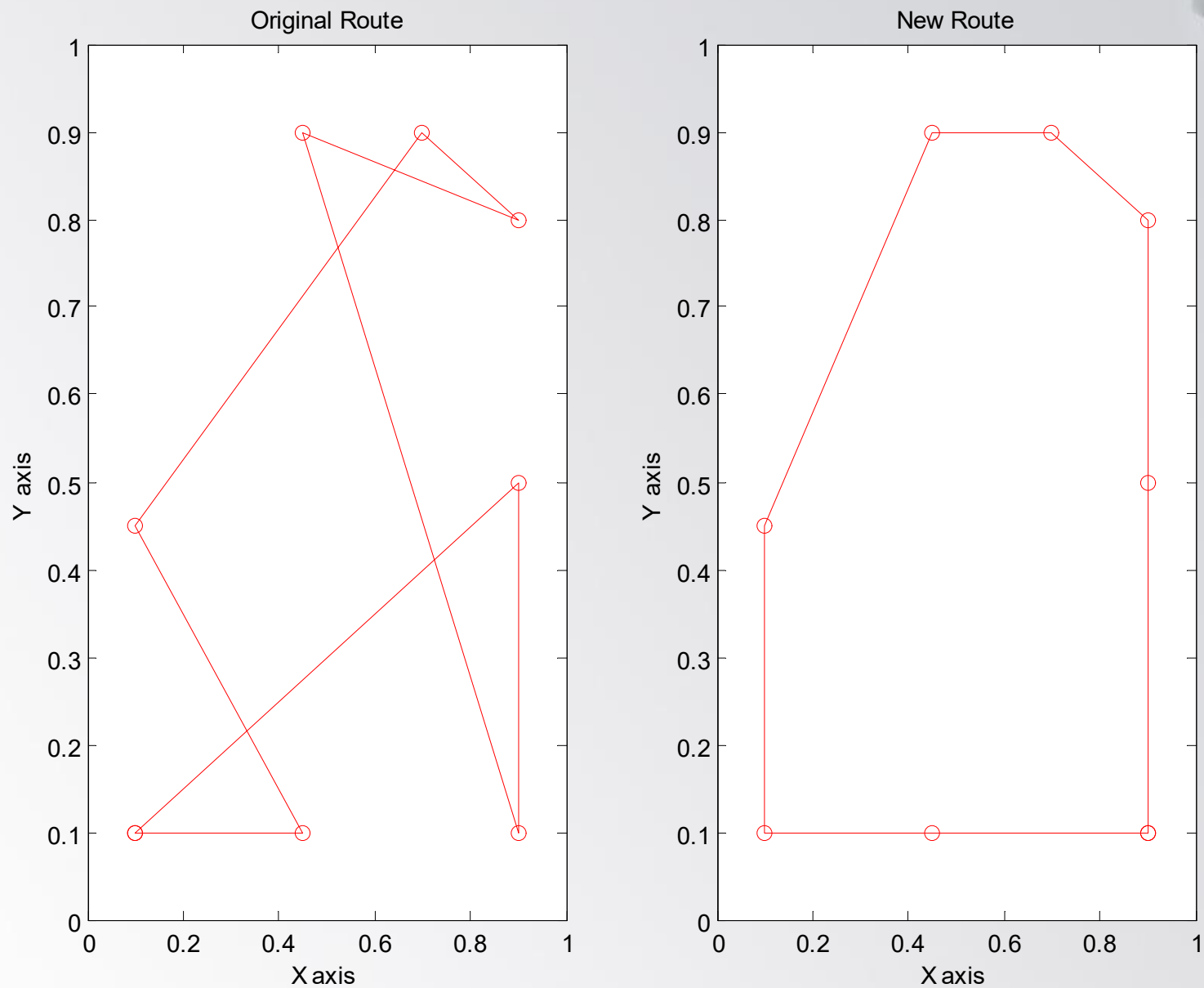


图8-5 8城市进化次数为50时的优化效果，距离 $L=2.8937(M=1)$



30个城市优化时，遗传算法参数设定为：群体中个体数目  $s=1500$ ，交叉概率  $P_c=0.10$ ，变异概率  $P_m=0.80$ 。取  $c=25$ ，经过300次进化，城市组合路径达到最小。最短路程为424.8693，如图8-6所示。

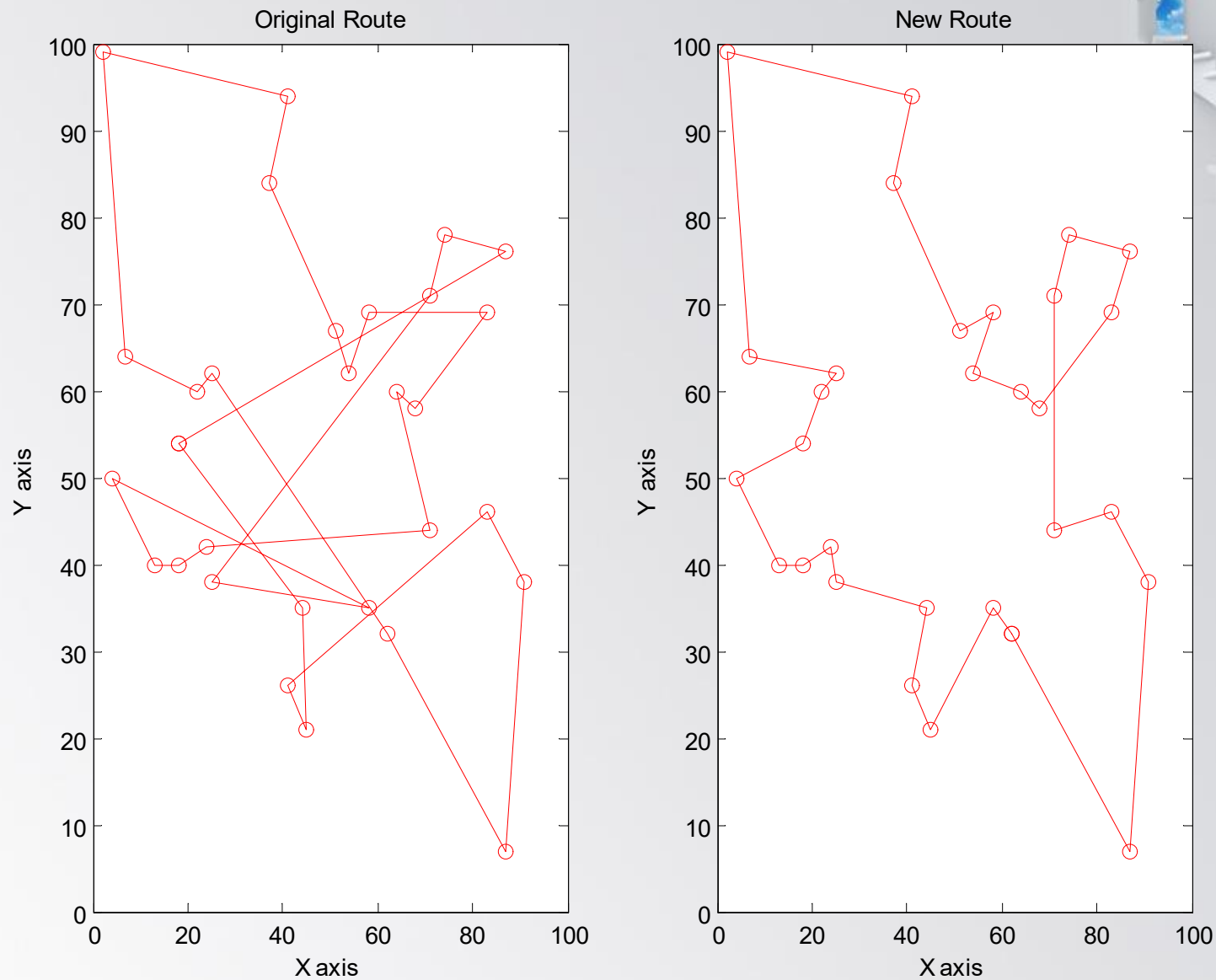


图8-6 30城市进化次数为300时的优化效果，  
距离  $L=424.8693$  ( $M=2$ )

# 作业

1. 在Rosenbrock 函数极大值遗传算法仿真程序chap8\_1.m中，通过改变群体大小M、终止进化代数G、交叉概率P<sub>c</sub>和变异概率P<sub>m</sub>,分析群体大小、终止进化代数、交叉概率对进化效果的影响。
2. 采用二进制编码方法，利用遗传算法求函数  $f(x_1, x_2, x_3)$  的极小值。

$$\begin{cases} f(x_1, x_2, x_3) = \sum_{i=1}^3 x_i^2 \\ -5.12 \leq x_i \leq 5.12 \end{cases} \quad i = 1, 2, 3$$



谢 谢!