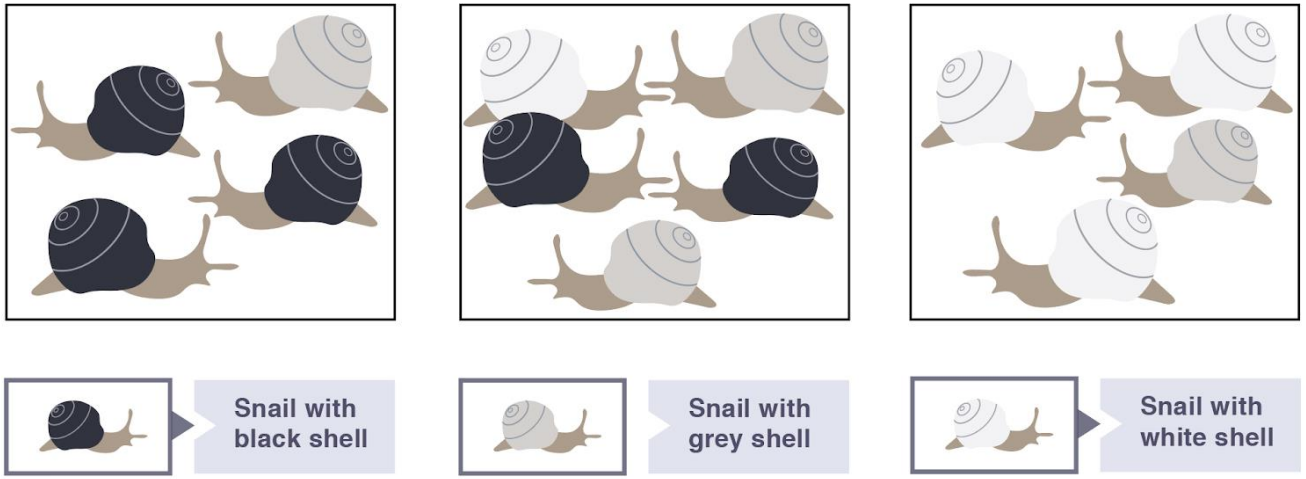
第六章 遗传算法

遗传算法

- 遗传算法（**Genetic Algorithm，GA**）最早是由美国的 John holland于20世纪70年代提出,该算法是**根据大自然中生物体进化规律**而设计提出的。
- 受到达尔文生物进化论的**自然选择**学说和孟德尔的**遗传学定律**而启发。



- 该算法通过数学的方式,利用计算机仿真运算,将**问题的求解过程转换成类似生物进化中的染色体基因的交叉、变异等过程**。
- 遗传算法已被人们广泛地应用于组合优化、机器学习、信号处理、自适应控制和人工生命等领域。

遗传算法的发展历史

- 1962年，Fraser提出了自然遗传算法。
- 1965年，Holland首次提出了人工遗传操作的重要性。
- 1967年，Bagley首次提出了**遗传算法**这一术语。
- 1970年，Cavicchio把遗传算法应用于**模式识别**中。
- 1971年，Hollstien在论文《计算机控制系统中人工遗传自适应方法》中阐述了遗传算法用于数字反馈控制的方法。
- 1975年，美国J. Holland出版了《自然系统和人工系统的适配》；DeJong完成了重要论文《遗传自适应系统的行为分析》。
- **20世纪80年代以后，遗传算法进入兴盛发展时期。**

生物学术语

- **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（根据基因型形成的个体的外部表现）
- **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。是一种优胜劣汰的过程。
- **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

遗传算法的思想

通过:

- 种群中个体的**选择**
- 种群中的**交叉**繁殖
- 种群中个体**变异**

上述操作逐代进行，
个体（解）逐渐优化

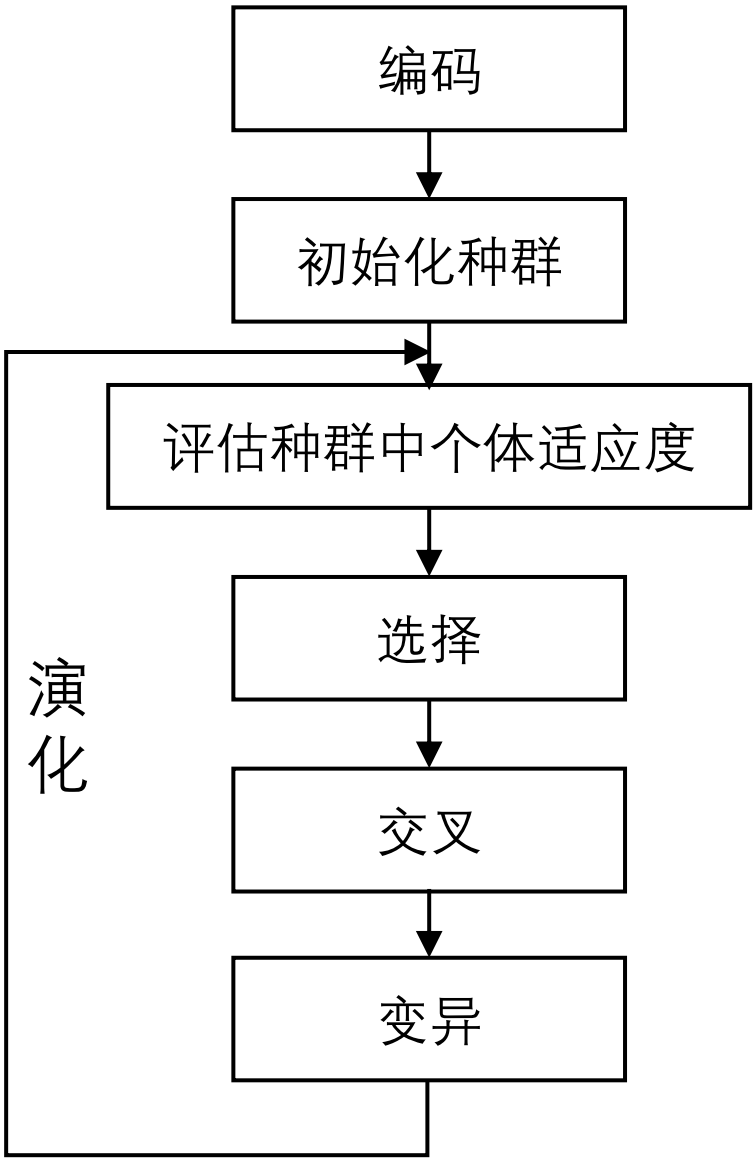
- Darwin的进化论
  ----“自然选择，适者生存”

个体：解空间当中某一候选解

遗传算法过程图解

在一个遗传算法中需要考虑如下**5个阶段**:

- **初始化种群（Initial population）**
- **适应度函数（Fitness function）**
- **选择(Selection)**
- **交叉(Crossover)**
- **变异(Mutation)**



遗传算法步骤

例： $\max f(x_1, x_2) = {x_1}^2 + {x_2}^2$

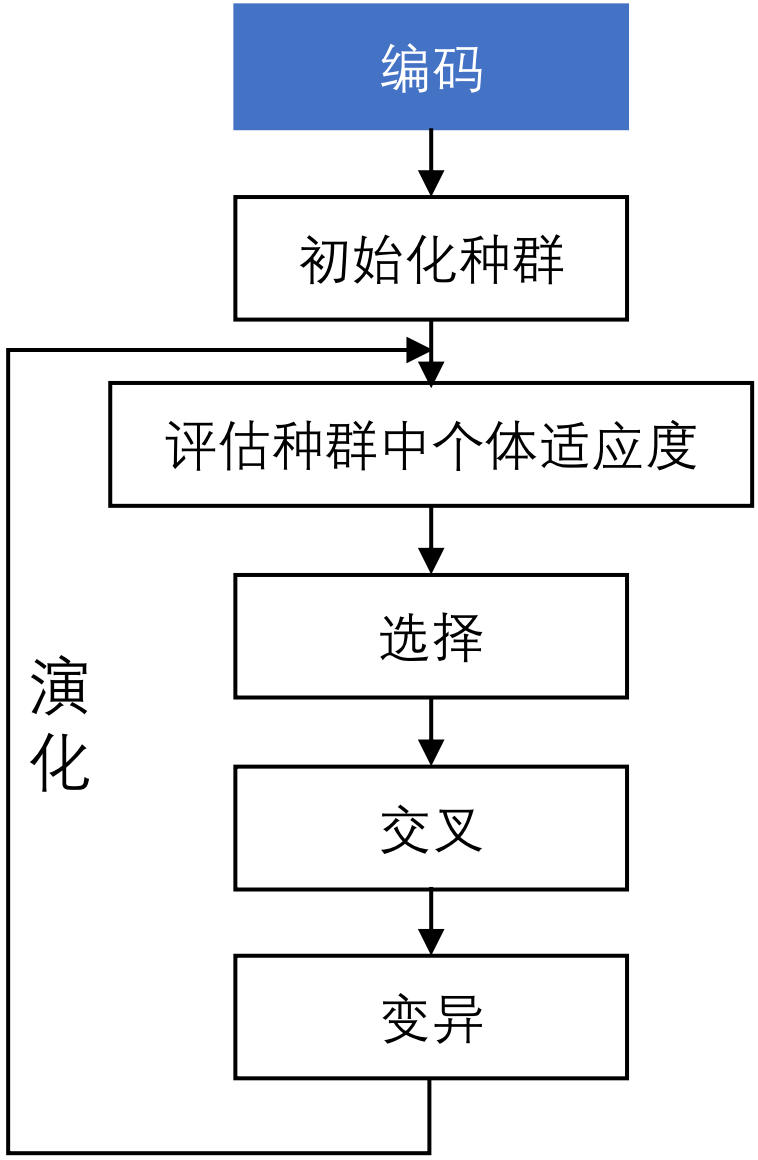
$$s.t.\ \ x_1 \in \{1,2,3,4,5,6,7\}$$
$$x_2 \in \{1,2,3,4,5,6,7\}$$

a) 编码

由于遗传算法不能直接处理问题空间的解，因此必须通过编码将问题空间的解转化为遗传空间中的基因型。这一转换操作就叫做**编码（encoding）**。

本例中，我们可以用无符号二进制整数对x_1, x_2进行编码。x_1, x_2是0-7的整数，分别可以用3位无符号二进制来表示，连接在一起便是6位无符号二进制数。它代表了遗传空间中个体的基因型。

例：个体基因$X = 101110$，所对应的表现型是$x_1 = 5, x_2 = 6$



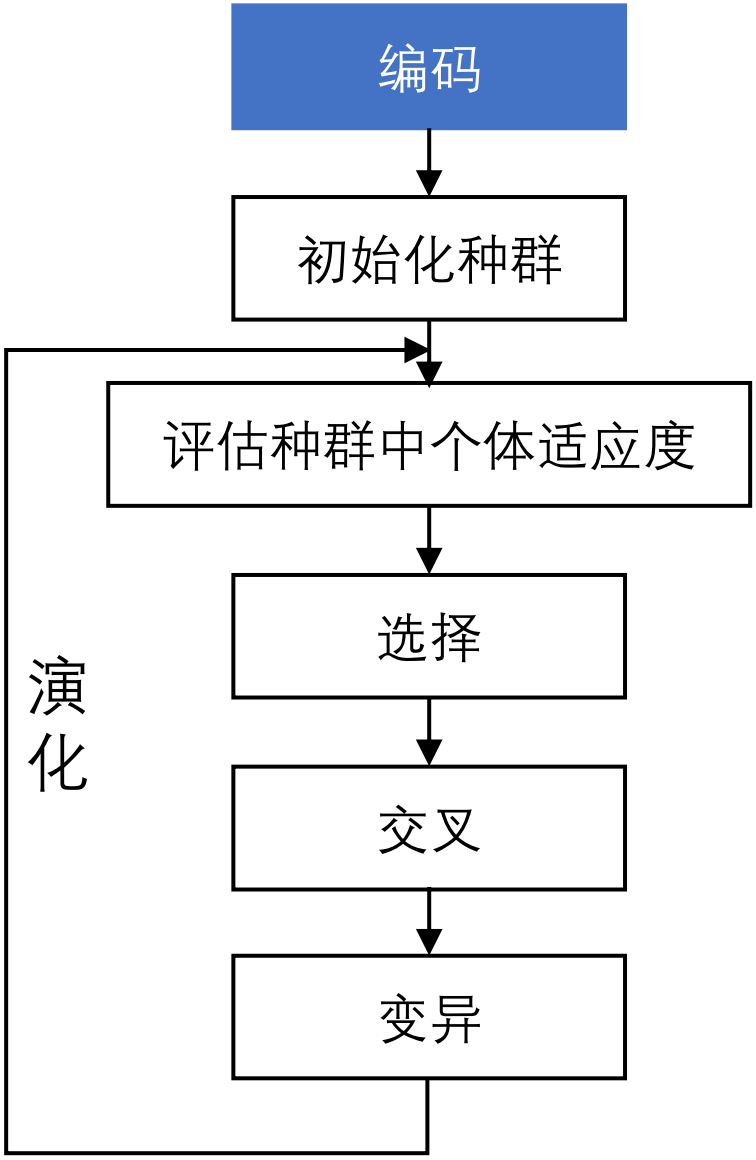
遗传算法步骤

a) 编码

编码策略（方法）需要遵循:

- **完备性**(completeness):问题空间中的所有点(候选解)都能表示为遗传空间中的点(染色体)。
- **健全性**(soundness): 遗传空间中的点能对应所有问题空间中的候选解。
- 即遗传空间中的点和问题空间的候选解要**一一对应**。

个体的表现型和基因型之间可通过编码和解码程序相互转换。



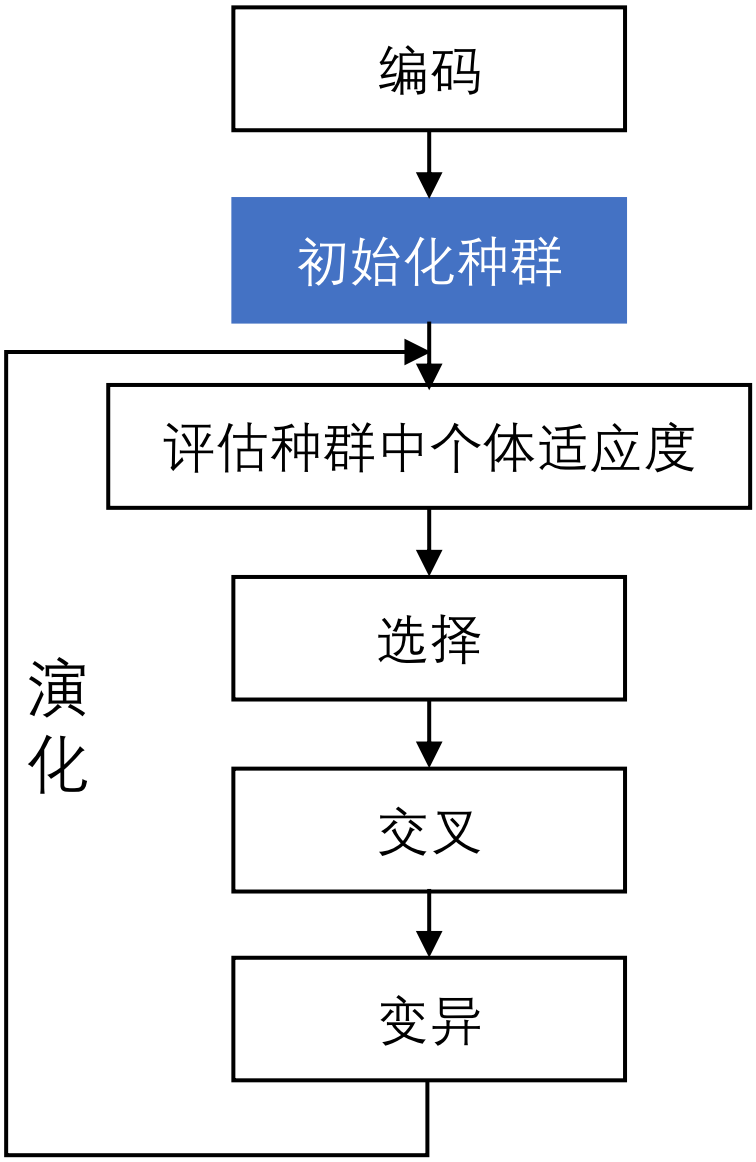
遗传算法步骤

b) 初始化种群

遗传算法是对**群体**进行的进化操作，需要准备**初始种群**以进行“进化”操作：

- 初始化种群的大小取决于要解决的问题，但通常包含成百上千的个体。
- 通常，初始群体的选取是随机的，是在整个搜索空间内的随机选取。
- 有先验知识的前提下，也可以在搜索空间的一部分区域内随机选取。

在本例中，我们设定群体规模的大小为4，即群体由4个个体组成，每个个体可通过随机方法产生。如：011101（$x_1 = 3, x_2 = 5$），101011（$x_1 = 5, x_2 = 3$），011100（$x_1 = 3, x_2 = 4$），111001（$x_1 = 7, x_2 = 1$）



遗传算法步骤

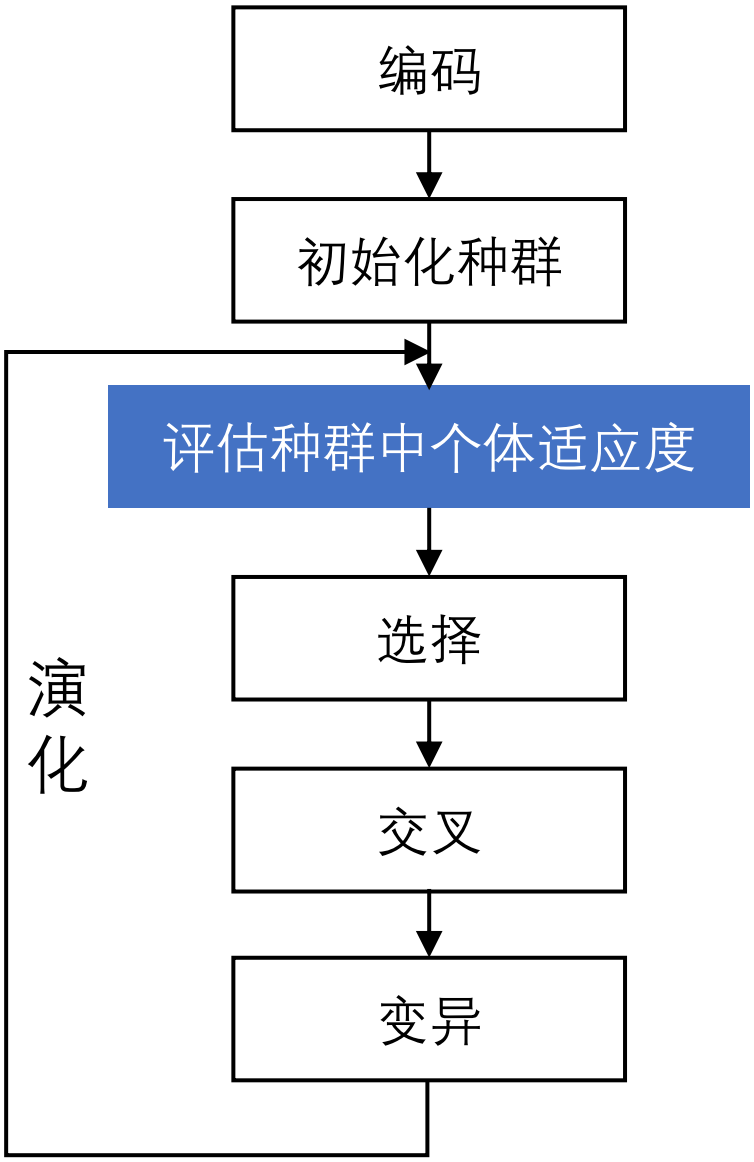
c) 评估种群中个体适应度

例：$\max f(x_1, x_2) = {x_1}^2 + {x_2}^2$

$$s.t.\ \ x_1 \in \{1,2,3,4,5,6,7\}$$
$$x_2 \in \{1,2,3,4,5,6,7\}$$

遗传算法中以**个体适应度**（对环境的适应程度）的大小来评定各个个体的优劣程度，从而决定其遗传机会的大小。

本例是以求函数最大值为目标，故可直接利用函数值作为个体的适应度。（因为如果一个个体（解）能够使函数取得更大的值，我们就说这个个体（解）的适应度更好。）



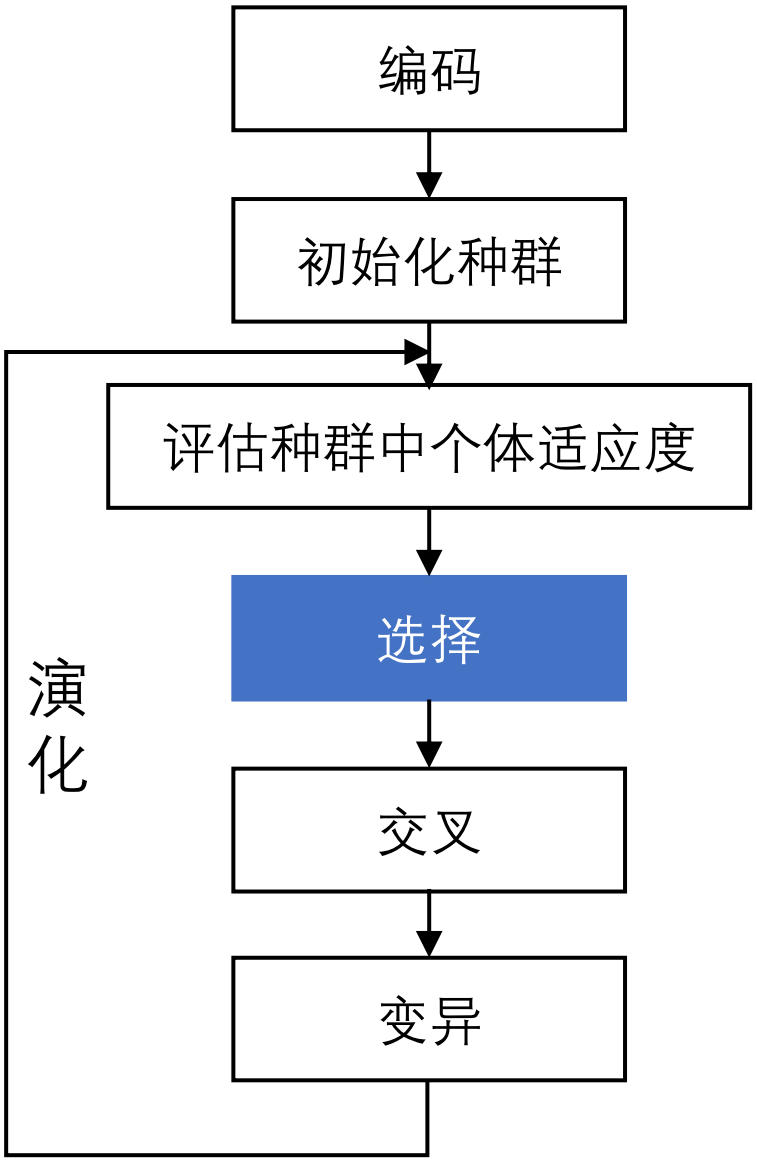
遗传算法步骤

d) 选择

选择运算把当前群体中适应度较高的个体按某种规则或模型遗传到下一代群体中。一般要求适应度较高的个体将有更多的机会遗传到下一代群体中。

- **适应度高**就有更大的概率被遗传到下一代。
- **适应度低**就容易被淘汰，有很小的概率被遗传到下一代。

本例中，我们采用与适应度成正比的概率来确定各个个体复制到下一代群体中的数量。



遗传算法步骤

d) 选择

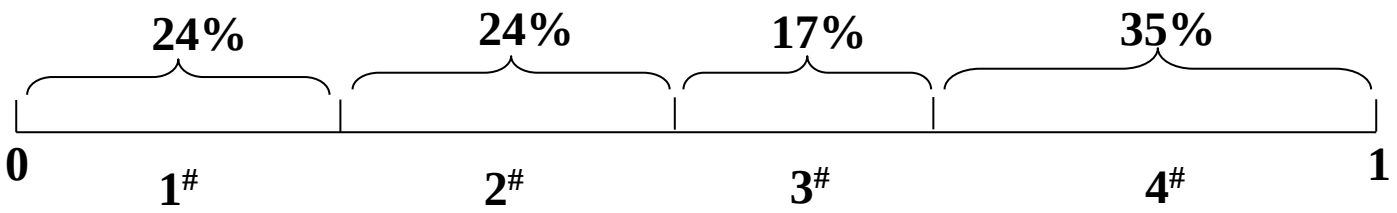
① 先计算出群体中所有个体的适应度的总和 Σf_i (i=1,2,…,M);

② 其次计算出每个个体的相对适应度的大小 $f_i / \Sigma f_i$ ，它即为每个个体被遗传到下一代群体中的概率

③ 每个概率值组成一个区域，全部概率值之和为1;

④ 最后再产生一个0到1之间的随机数，依据该随机数出现在上述哪一个概率区域内来确定各个个体被选中的次数。

个体编号	初始群体p(0)	x_1	x_2	适应度	占总数的百分比	选择次数	选择结果
1	011101	3	5	34	0.24	1	011101
2	101011	5	3	34	0.24	1	111001
3	011100	3	4	25	0.17	0	101011
4	111001	7	1	50	0.35	2	111001
总和				143	1		



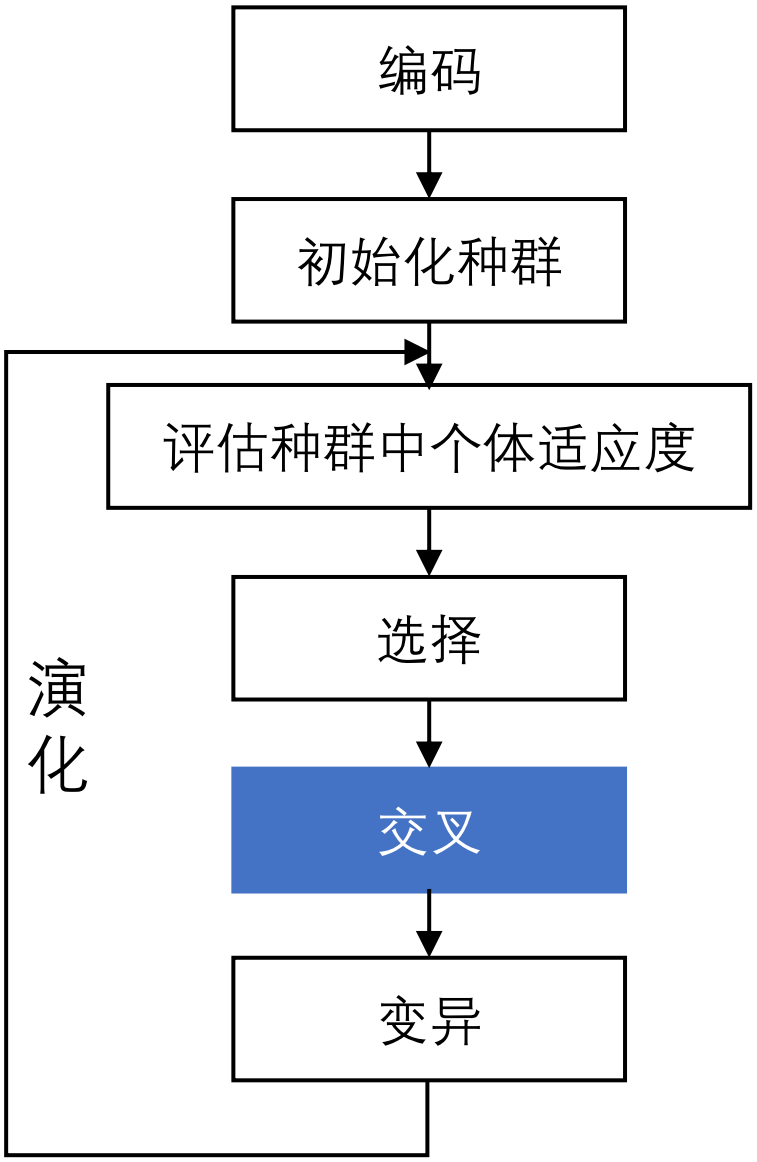
遗传算法步骤

e) 交叉

- 交叉运算是遗传算法中产生新个体的主要操作过程，它以某一概率相互交换某两个个体之间的部分染色体。
- 交叉从父代个体对的相同位置选择长度相同的子串，替换它们，并生成一个新的个体。交叉点可以随机选择。

个体X	1	0	0	1	1	0	0	1
个体Y	1	1	1	1	0	0	0	1

个体P	1	1	1	1	1	0	0	1
个体Q	1	0	0	1	0	0	0	1



遗传算法步骤

e) 交叉

本例采用**单点交叉**的方法，其具体操作过程是:

- 先对群体进行随机配对;
- 其次随机设置交叉点位置;
- 最后再相互交换配对染色体之间的部分基因。

个体编号	选择结果	配对情况	交叉点位置	交叉结果
1 2 3 4	01 1101 11 1001 1010 11 1110 01	1-2 3-4	1-2：3，4 3-4：5，6	011001 111101 101001 111011

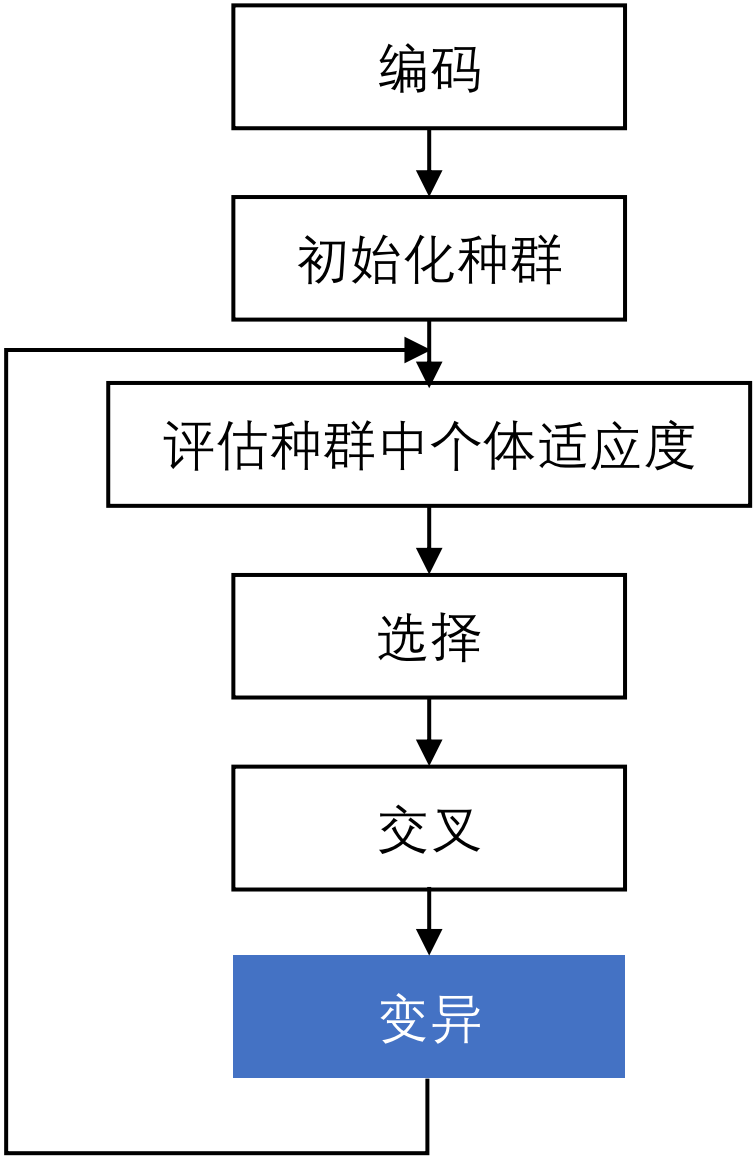
可以看出，其中新产生的个体“111101”、“111011”的适应度（74，58）较原来两个个体的适应度都要高。

遗传算法步骤

f) 变异

- 变异运算是对个体的某一个或某一些基因座上的基因值按某一较小的概率进行改变，它也是产生新个体的一种操作方法。
- 一般地，变异是一个翻转操作。随机选择0～1之间的数字，如果随机数小于变异概率$p_{Mutation}$，则变异在当前位完成，否则该位不会改变。
- 对于二进制编码，变异意味着翻转一个或多个随机选择位，将1变成0或将0变成1.

个体X	1	0	0	1	1	0	0	1
新个体X	1	0	1	1	0	0	0	1



遗传算法步骤

f) 变异

本例中，我们采用基本位变异的方法来进行变异运算，其具体操作过程是:

- 首先确定出各个个体的基因变异位置。下表所示为随机产生的变异点位置，其中的数字表示变异点设置在该基因座处;
- 然后依照某一概率将变异点的原有基因值取反。

个体编号	交叉结果	变异点	变异结果	子代群体p(1)
1	011001	4	011101	011101
2	111101	5	111111	111111
3	101001	2	111001	111001
4	111011	6	111010	111010

对群体P(t)进行一轮选择、交叉、变异运算之后可得到新一代的群体p(t+1)。

遗传算法步骤

g) 下一代进化结果

个体编号	子群体p(1)	x_1 x_2	适值	占总数的百分比
1	011101	3 5	34	0.14
2	111111	7 7	98	0.42
3	111001	7 1	50	0.21
4	111010	7 2	53	0.23
总和			235	1

从上表中可以看出，群体经过一代进化之后，其适应度的最大值、平均值都得到了明显的改进。事实上，这里已经找到了最佳个体“111111”。

[注意]

需要说明的是：这里为了更好地说明问题，我们特意选择了一些较好的数值以便能够得到较好的结果，而在实际运算过程中有可能需要一定的循环次数才能达到这个最优结果。

各步骤评价

- 选择 *--- 优胜劣汰*

  选择操作为种群提供了**演进的方向**

- 交叉 *--- 优优组合*

  交叉操作的作用在于**汇集散布于不同个体间的局部优势模式**

- 变异 *--- 寻找新模式*

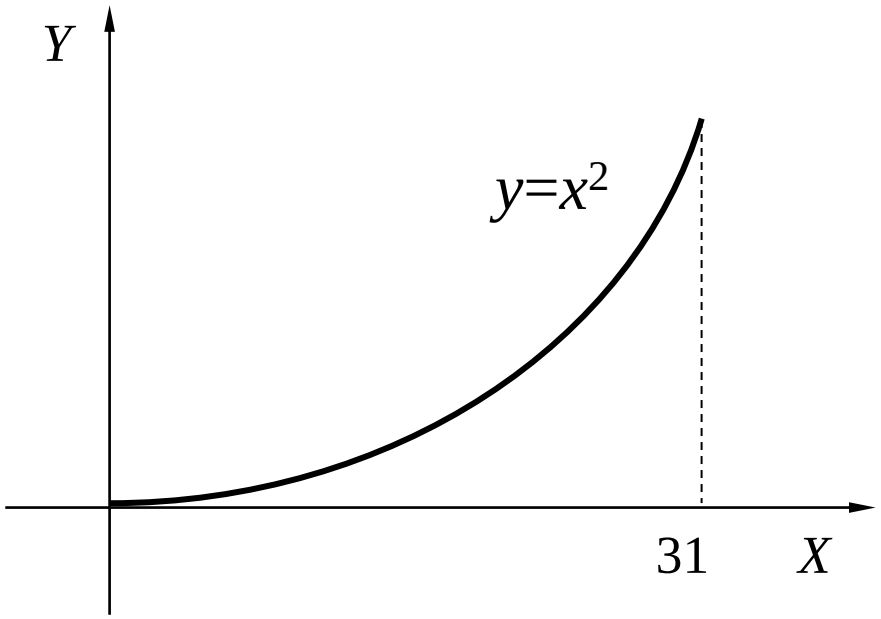
  变异操作是**种群向外扩展的触角**(随机)

  好的变异将保留，坏的淘汰

例子2

问题

利用遗传算法求解区间［0,31］上的二次函数$y=x^2$的最大值。



例子2

编码，初始化种群，定义种群个体适应度

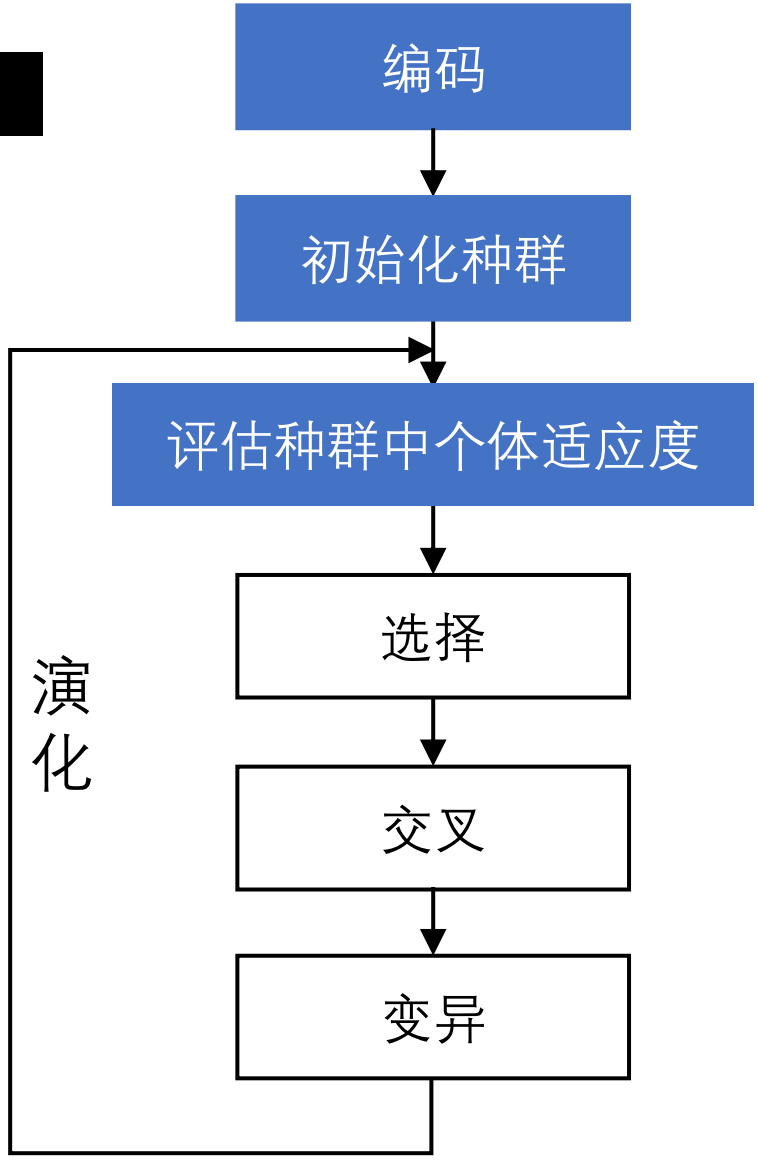
- 设定种群规模,编码染色体，产生初始种群。
- 将种群规模设定为4；用5位二进制数编码染色体；取下列个体组成初始种群S1:

  s1= 13 (01101), s2= 24 (11000)

  s3= 8 (01000), s4= 19 (10011)

- 定义适应度函数:

  取适应度函数：$f(x) = x^2$

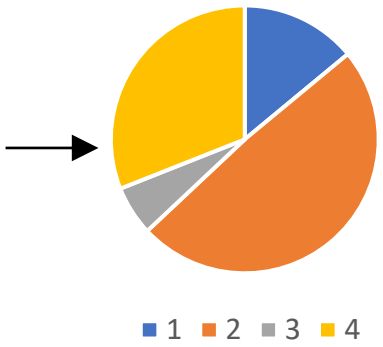


例子2

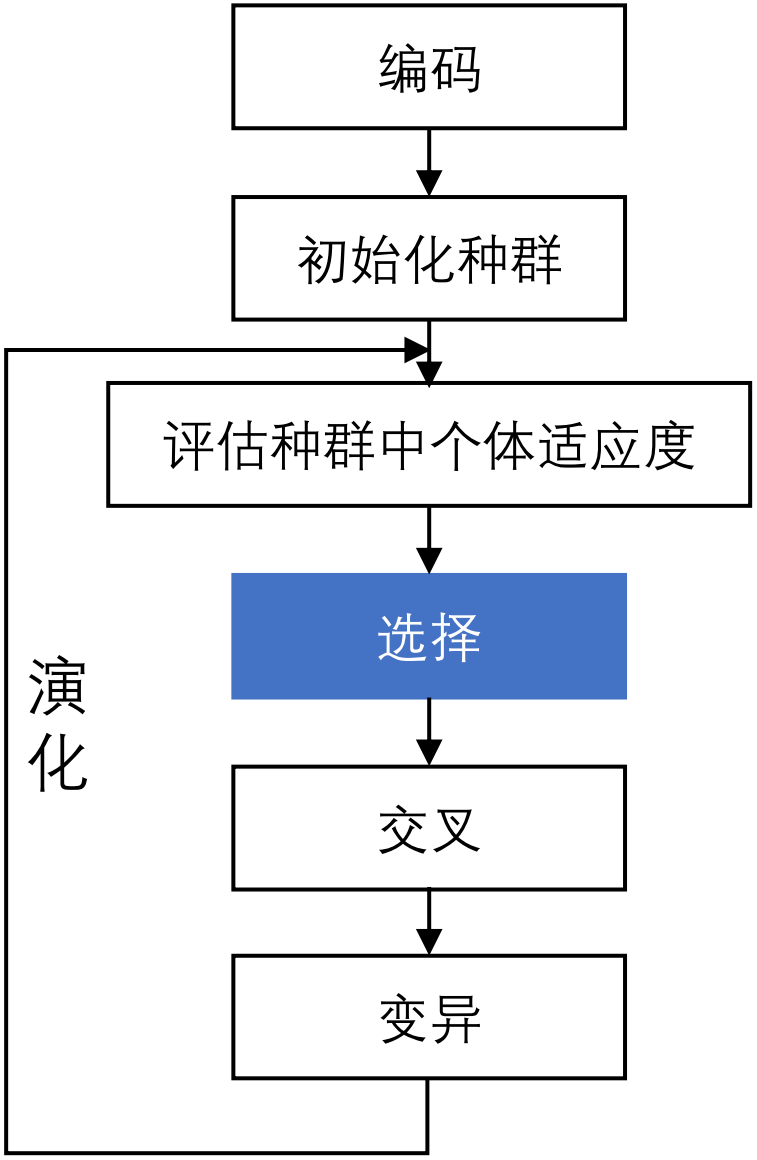
选择

- 选择那些适应度比较好的，
  淘汰掉那些适应度比较差的

个体编号	初始群体p(0)	适应度	占总数的百分比
1	01101 (13)	169	0.14
2	11000 (24)	576	0.49
3	01000 (8)	64	0.06
4	10011 (19)	361	0.31
总和		1170	1



个体1被选中了1次，
个体2被选中了2次，
个体4被选中了1次



例子2

交叉

于是，经过**选择**得到如下个体：

s_1'=11000（24），s_2'=01101（13）

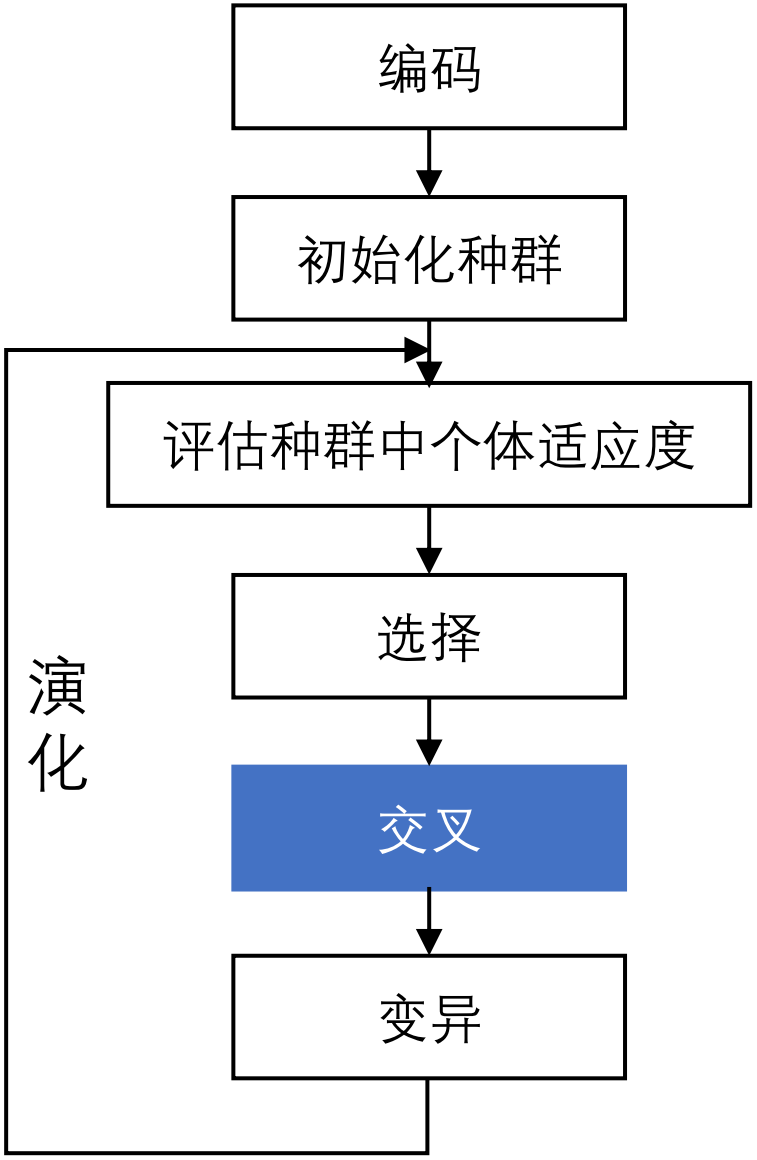
s_3'=11000（24），s_4'=10011（19）

设交叉率p_c=100%，即S_1中的全体染色体都参加交叉运算。

设s_1'与s_2'配对，s_3'与s_4'配对。分别交换后两位基因，得新染色体：

s_1''=11001（25），s_2''=01100（12）

s_3''=11011（27），s_4''=10000（16）



例子2

变异

设变异率p_m=0.1（变异率定义了变异的基因个数）。这样，群体S_1中共有5×4×0.1=2 位基因可以变异。

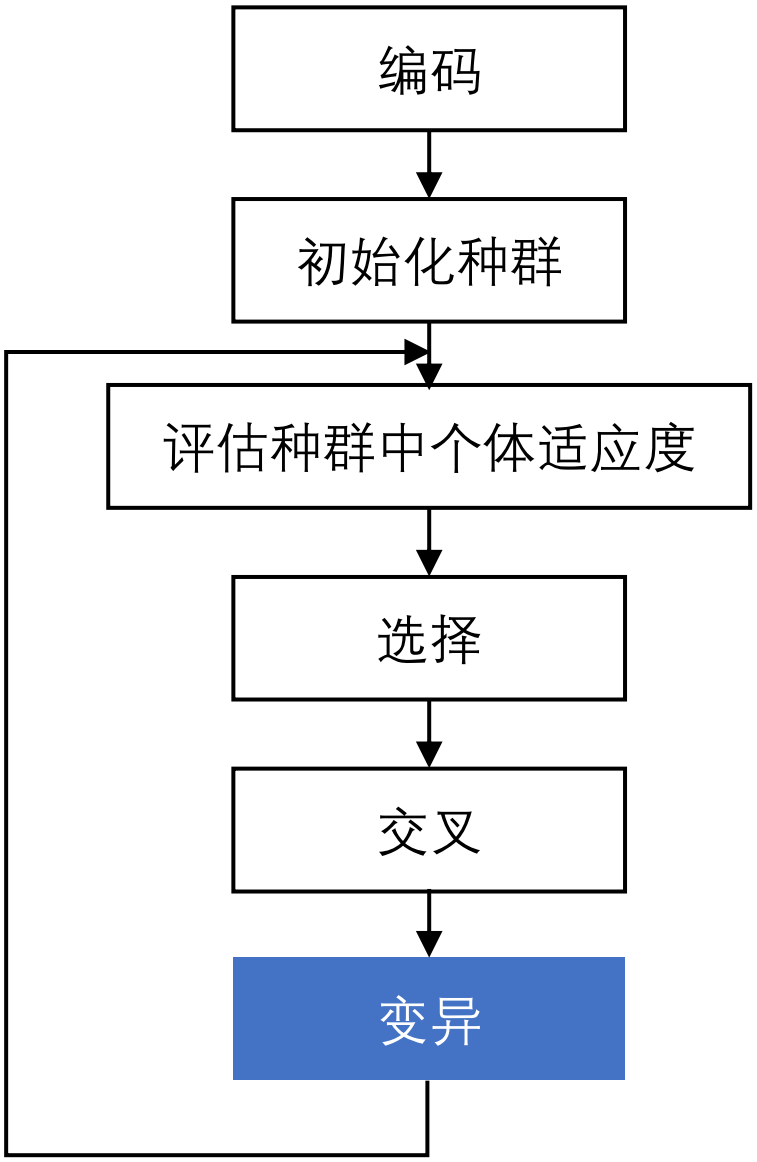
s_1'''=11001（25），s_2'''=01110（14）

s_3'''=11011（27），s_4'''=10001（17）

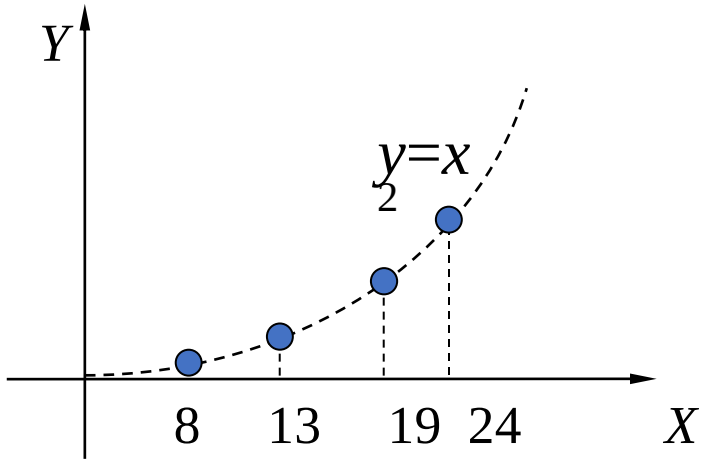
于是，得到第二代种群S_2:

s_1=11001（25），s_2=01110（14）

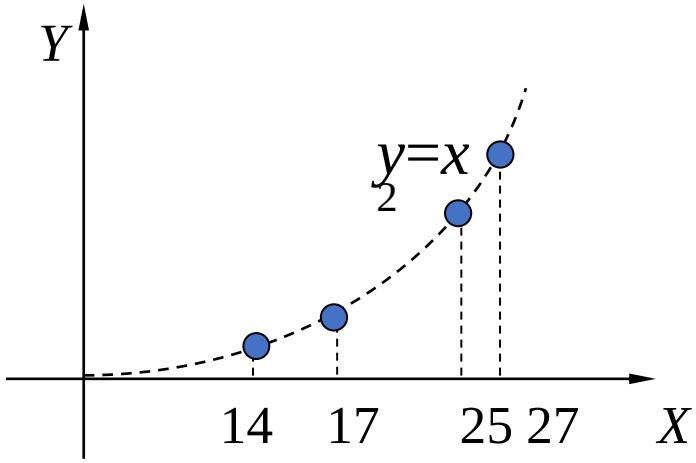
s_3=11011（27），s_4=10001（17）



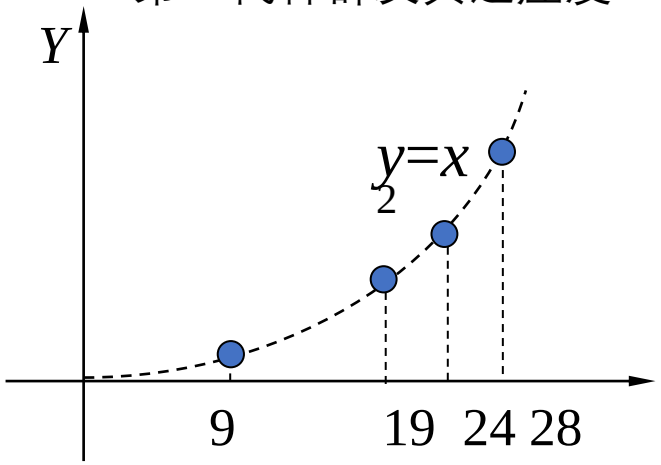
例子2



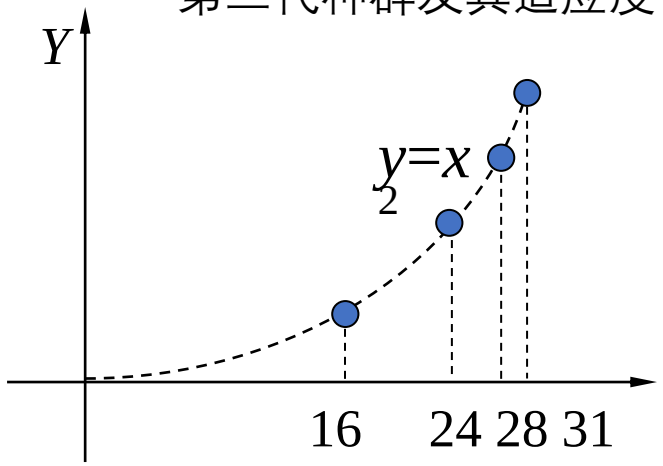
第一代种群及其适应度



第二代种群及其适应度



第三代种群及其适应度



第四代种群及其适应度

随着种群的进化，种群的适应度越来越高（适应度越高就代表着函数能够取得更大的值）

课堂思考

1. 遗传算法是一种基于生物进化原理的优化算法。
2. 在遗传算法中，适应度函数用于评估个体解的优劣。
3. 遗传算法的选择操作是随机的。
4. 在遗传算法中，变异操作通过随机改变个体的部分基因来产生新的解。
5. 遗传算法总是能够找到问题的全局最优解。
6. 遗传算法不适合解决具有离散解空间的优化问题。