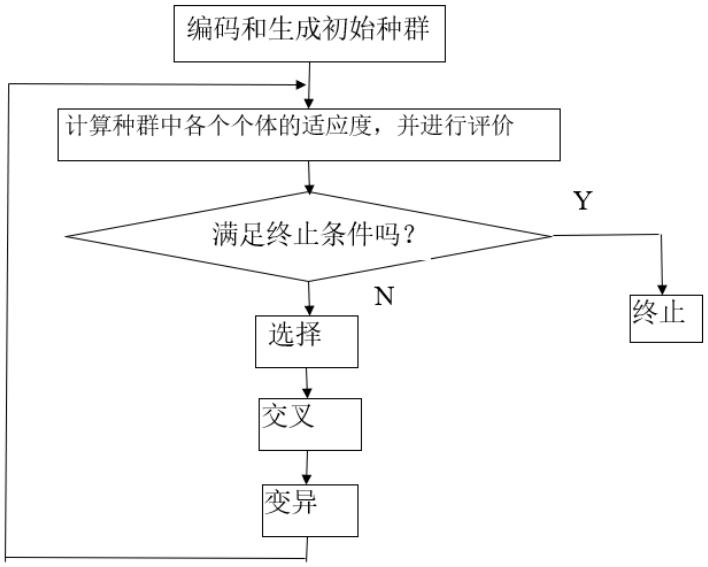
遗传算法



- 遗传编码

常用的遗传编码算法有霍兰德二进制码、格雷码、实数编码和字符编码等。我们这里只讲解二进制编码：

二进制编码是将原问题的结构变换为染色体的位串结构。在二进制编码中，首先要确定二进制字符串的长度l，该长度与变量的定义域和所求问题的计算精度有关。

例：假设变量x的定义域为$[5,10]$，要求的计算精度为10^{-5}，则需要将$[5,10]$至少分为600000个等长小区间，每个小区间用一个二进制串表示。于是，串长至少等于20，原因是：

$$524288 = 2^{19} < 600000 < 2^{20} = 1048576$$

这样，对应于区间$[5,10]$内满足精度要求的每个值x，都可用一个20位编码的二进制串$< b_{19}, b_{18}, \cdots, b_0 >$来表示。

- 适应度函数

适应度函数是一个用于对个体的适应性进行度量的函数。通常，一个个体的

适应度值越大，它被遗传到下一代种群中的概率也就越大。

（一）常用的适应度函数

在遗传算吗中，有许多计算适应度的方法，其中最常用的适应度函数有以下两种：

1. 原始适应度函数

它是直接将待求解问题的目标函数 $f(x)$ 定义为遗传算法的适应度函数。例如，在求解极值问题 $\max_{x\in[a,b]} f(x)$ 时， $f(x)$ 即为 x 的原始适应度函数。

采用原始适应度函数的优点是能够直接反映出待求解问题的最初求解目标，其缺点是有可能出现适应度值为负的情况。

2. 标准适应度函数

在遗传算法中，一般要求适应度函数非负，并其适应度值越大越好。这就往往需要对原始适应函数进行某种变换，将其转换为标准的度量方式，以满足进化操作的要求，这样所得到的适应度函数被称为标准适应度函数 $f_{Normal}(x)$ 。

（1）极小化问题：

对极小化问题，其标准适应度函数可定义为：

$$f(x)=\begin{cases} f_{\max}(x)-f(x) \\ 0 \end{cases}$$

当 $f(x)<f_{\max}(x)$ 时，有 $f(x)=f_{\max}(x)-f(x)$ ；否则有 $f(x)=0$ 。

其中， $f_{\max}(x)$ 是原始适应函数 $f(x)$ 的一个上界。如果 $f_{\max}(x)$ 未知，则可用当前代或到目前为止各演化代中的 $f(x)$ 的最大值来代替。可见， $f_{\max}(x)$ 是会随着进化代数的增加而不断变化的。

（2）极大化问题

对极大化问题，其标准适应度函数可定义为：

$$f(x)=\begin{cases} f(x)-f_{\min}(x) \\ 0 \end{cases}$$

当 $f(x)>f_{\min}(x)$ 时，有 $f(x)=f(x)-f_{\min}(x)$，否则有 $f(x)=0$。

其中，$f_{\min}(x)$ 是原始适应函数 $f(x)$ 的一个下界。如果 $f_{\min}(x)$ 未知，则可用当前代或到目前为止各演化代中的 $f(x)$ 的最小值来代替。

- 基本遗传操作

遗传算法中的基本遗传操作包括选择、交叉和变异3种，而每种操作又包括多种不同的方法，下面分别对它们进行介绍。

（一）选择操作

选择操作是指根据选择概率按照某种策略从当前种群中挑选出一定数目的个体，使它们能够有更多的机会被遗传到下一代中。常用的选择策略可分为比例选择、排序选择和竞技选择三种类型。

（1） 比例选择

比例选择方法的基本思想是：各个个体被选中的概率与其适应度大小成正比。

常用的比例选择策略包括轮盘赌选择和繁殖池选择。

（2） 轮盘赌选择

轮盘赌选择法又被称为转盘赌选择法或轮盘选择法。在这种方法中，个体被选中的概率取决于该个体的相对适应度。而相对适应度的定义为：

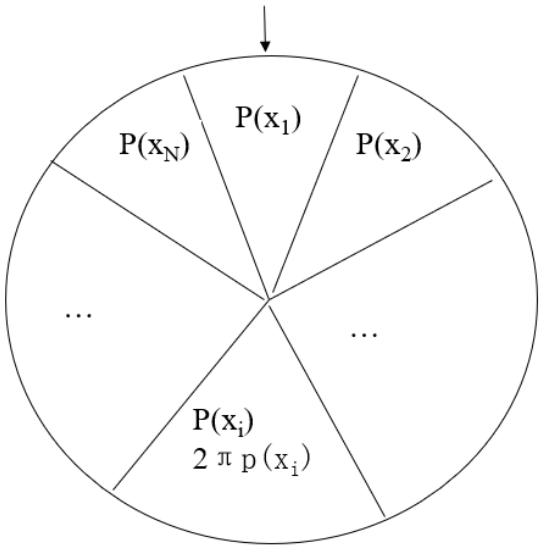
$$P(x_i)=\frac{f(x_i)}{\sum_{j=1}^{N} f(x_j)}$$

其中，$P(x_i)$ 是个体 x_i 的相对适应度，即个体 x_i 被选中的概率：$f(x_i)$ 是个体 x_i 的原始适应度；是种群的累加适应度。

轮盘赌选择算法的基本思想是：根据每个个体的选择概率$P(x_i)$将一个圆盘分成N个扇区，其中第i个扇区的中心角为：

$$2\pi \frac{f(x_i)}{\sum_{j=1}^{N} f_i(x_j)} = 2\pi p(x_i)$$

并再设立一个固定的指针。当进行选择时，可以假想转动圆盘，若圆盘静止时指针指向第i个扇区，则选择个体i，其物理意义如下：



从统计角度看，个体的适应度值越大，其对应的扇区的面积越大，被选中的可能性也越大。这种方法有点类似于发放奖品使用的轮盘，并带有某种赌博的意思，因此亦被称为轮盘赌选择。

（二）交叉操作

交叉操作是指按照某种方式对选择的父代个体的染色体的部分基因进行交配重组，从而形成新的个体。交配重组是自然界中生物遗传进化的一个主要环节，也是遗传算法中产生新的个体的最主要方法。根据个体编码方法的不同，遗传算法中的交叉操作可分为二进制交叉和实值交叉两种类型。

（1） 二进制交叉

二进制交叉是指二进制编码情况下所采用的交叉操作，它主要包括单点交叉、两点交叉、多点交叉和均匀交叉等方法。

1. 单点交叉

单点交叉也称简单交叉，它是先在两个父代个体的编码串中随机设定一个交叉点，然后对这两个父代个体交叉点前面或后面部分的基因进行交换，并生成子代中的两个新的个体。假设两个父代的个体串分别是：

$$X = x_1, x_2, \cdots, x_k, x_{k+1}, \cdots, x_n$$
$$Y = y_1, y_2, \cdots, y_k, y_{k+1}, \cdots, y_n$$

随机选择第 k 位为交叉点，若采用对交叉点后面的基因进行交换的方法，但点交叉是将 X 中的 x_{k+1} 到 x_n 部分与 Y 中的 y_{k+1} 到 y_n 部分进行交叉，交叉后生成的两个新的个体是：

$$X' = x_1, x_2, \cdots, x_k, y_{k+1}, \cdots, y_n$$
$$Y' = y_1, y_2, \cdots, y_k, x_{k+1}, \cdots, x_n$$

例：设有两个父代的个体串 A=0 0 1 1 ⋮0 1 和 B=1 1 0 0 ⋮1 0，若随机交叉点为 4，则交叉后生成的两个新的个体是：

A'=0 0 1 1⋮ 1 0

B'=1 1 0 0⋮ 0 1

2. 两点交叉

两点交叉是指先在两个父代个体的编码串中随机设定两个交叉点，然后再按这两个交叉点进行部分基因交换，生成子代中的两个新的个体。

假设两个父代的个体串分别是：

$$X = x_1, x_2, \cdots, x_i, \cdots, x_j, \cdots, x_n$$
$$Y = y_1, y_2, \cdots, y_i, \cdots, y_j, \cdots, y_n$$

随机设定第 i 和第 j 位为两个交叉点（其中 $i < j < n$），两点交叉是将 X 中的

x_{i+1}到x_j部分与Y中的y_{i+1}到y_j部分进行交换，交叉后生成的两个新的个体是：

$$X' = x_1, x_2, \cdots, x_i, y_{i+1}, \cdots, y_j, x_{j+1}, \cdots, x_n$$
$$Y' = y_1, y_2, \cdots, y_i, x_{i+1}, \cdots, x_j, y_{j+1}, \cdots, y_n$$

例：设有两个父代的个体串A=0 0 1 ⋮1 0 ⋮1和B=1 1 0 ⋮0 1 ⋮0，若随机交叉点为3和5，则交叉后的两个新的个体是：

A'=0 0 1 ⋮0 1⋮ 1

B'=1 1 0 ⋮1 0⋮ 0

3. 多点交叉

多点交叉是指先随机生成多个交叉点，然后再按这些交叉点分段地进行部分基因交换，生成子代中的两个新的个体。

这里不再赘述！

（三）变异操作

变异是指对选中个体的染色体中的某些基因进行变动，以形成新的个体。变异也是生物遗传和自然进化中的一种基本现象，它可增强种群的多样性。遗传算法中的变异操作增加了算法的局部随机搜索能力，从而可以维持种群的多样性。根据个体编码方式的不同，变异操作可分为二进制变异和实值变异两种类型。

（1） 二进制变异

当个体的染色体采用二进制编码表示时，其变异操作应采用二进制变异方法。该变异方法是先随机地产生一个变异位，然后将该变异位置上的基因值由“0”变为“1”，或由“1”变为“0”，产生一个新的个体。

例：设变异前的个体为A=0 0 1 1 0 1，若随机产生的变异位置是2，则该个体的第2位由“0”变为“1”。

变异后的新的个体是：$A' = 0\ 1\ 1\ 1\ 0\ 1$

- 遗传算法示例讲解

例题：用遗传算法求函数 $f(x) = x^2$ 的最大值，其中 x 为 $[0,31]$ 间的整数。

解：这个问题本身比较简单，其最大值很显然是在 $x = 31$ 初。但作为一个例子，它有着较好的示范性和可理解性。

按照遗传算法，求求解过程如下：

1. 编码

由于 x 的定义域是区间 $[0,31]$ 上的整数，由 5 位二进制数即可全部表示。因此，可采用二进制编码方法，其编码串的长度为 5。

例如，用二进制串 00000 来表示 $x = 0$，11111 来表示 $x = 31$ 等。其中的 0 和 1 为基因值。

2. 生成初始种群

若假设给定的种群规模 $N = 4$，则可用 4 个随机生成的长度为 5 的二进制串作为初始种群，再假设随机生成的初始种群（即第 0 代种群）为：

$S_{01} = 0\ 1\ 1\ 0\ 1$　　$S_{02} = 1\ 1\ 0\ 0\ 1$

$S_{03} = 0\ 1\ 0\ 0\ 0$　　$S_{04} = 1\ 0\ 0\ 1\ 0$

3. 适应度计算

遗传算法中以个体适应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。

要计算个体的适应度，首先应该定义适应度函数。由于本例是求 $f(x)$ 的最大值，因此可直接用 $f(x)$ 来作为适应度函数。即：

$$f(s) = f(x)$$

其中的二进制串 s 对应着变量 x 的值。根据此函数，初始种群中各个个体的适应值及其所占的比例如下表所示：

编号	个体串（染色体）	x	适应值	百分比%	累计百分比%	选中次数
s_{01}	01101	13	169	14.30%	14.30%	1
s_{02}	11001	25	625	52.88%	67.18%	2
s_{03}	01000	8	64	5.41%	72.59%	0
s_{04}	10010	18	324	27.41%	100%	1

可以看出，在 4 个个体中，s_{02} 的适应度最大，是当前最佳个体。

我总结为以下的几步：

1） 计算适应值，对应于本例子的 $f(x)$

2） 计算适应度，也就是适应值/总和，就是占总数的百分比

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

4. 选择操作

假设采用轮盘赌的方式选择个体，且依次生成的 4 个随机数（相当于轮盘上指针所指的数）为 0.85、0.32、0.12、0.46，经选择后得到的新的种群为：

$s_{01}=10010$

$s_{02}=11001$

$s_{03}=01101$

$s_{04}=11001$

其中，染色体 11001 在种群中出现了 2 次，而原染色体 01000 则因适应值太小而被淘汰。

5. 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。

我总结为以下几步：

1） 先对群体进行随机配对，其次随机设置交叉点位置

2） 最后再相互交换配对染色体之间的部分基因，产生新的个体

假设交叉概率P_i为50%，则种群中只有$\frac{1}{2}$的染色体参与交叉。若规定种群中的染色体按顺序两两配对交叉，且有s_{01}和s_{02}交叉，s_{03}和s_{04}不交叉，则交叉情况如下表所示：

编号	个体串(染色体)	交叉对象	交叉位	子代	适应值
s_{01}	10010	s_{02}	3	10001	289
s_{02}	11001	s_{01}	3	11010	676
s_{03}	01101	s_{04}	N	01101	169
s_{04}	11001	s_{03}	N	11001	625

可见，经交叉后得到的新的种群为：

$s_{01}=10001$

$s_{02}=11010$

$s_{03}=01101$

$s_{04}=11001$

6. 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。

总结：

1） 首先确定出各个个体的基因变异位置，随机产生的变异点位置。

2） 然后依照某一概率将变异点的原有基因值取反。

变异概率 P_m 一般都很小，假设本次循环中没有发生变异，则变异前的种群即为进化后所得到的第 1 代种群，即：

$s_{11}=10001$

$s_{12}=11010$

$s_{13}=01101$

$s_{14}=11001$

然后，对第 1 代种群重复上述（4）—（6）的操作。

对第 1 代种群，同样重复上述（4）—（6）的操作，其选择情况如下所示：

编号	个体串（染色体）	x	适应值	百分比%	累计百分比%	选中次数
s_{11}	10001	27	289	16.43%	16.43%	1
s_{12}	11010	28	676	38.43%	54.86%	2
s_{13}	01101	13	169	9.61%	64.47%	0
s_{14}	11001	25	625	35.53%	100%	1

其中若假设按轮盘赌选择时依次生成的 4 个随机数为 0.14、0.51、0.24 和 0.84，经选择后得到的新的种群为：

$s_{11}=10001$

$s_{12}=11010$

$s_{13}=11010$

$s_{14}=11001$

可以看出，染色体 11010 被选择了 2 次，而原染色体 01101 则因适应值太小而被淘汰。

对于第 1 代种群，其交叉情况如下表所示：

编号	个体串(染色体)	交叉对象	交叉位	子代	适应值
s_{11}	10001	s_{12}	3	10010	324
s_{12}	11010	s_{11}	3	11001	625
s_{13}	11010	s_{14}	2	11001	625
s_{14}	11001	s_{13}	2	11010	675

可见，经杂交后得到的新的种群为：

$s_{11}=10010$

$s_{12}=11001$

$s_{13}=11001$

$s_{14}=11010$

可以看出，第 3 位基因均为 0，已经不可能通过交配达到最优解。这种过早陷入局部最优解的现象称为早熟。为解决这一问题，需要采用变异操作。

对于第一代种群，其变异情况如下表所示：

编号	个体串（染色体）	是否变异	变异位	子代	适应值
s_{11}	10010	N		10010	324
s_{12}	11001	N		11001	625
s_{13}	11001	N		11001	625
s_{14}	11010	Y	3	11110	900

它是通过对s_{14}的第3位的变异来实现的。变异后所得到的第2代的种群为：

$s_{21}=10010$

$s_{22}=11001$

$s_{23}=11001$

$s_{24}=11110$

接着，再对第2代种群同样重复上述（4）—（6）的操作：

对第2代种群，同样重复上述(4)-(6)的操作。其选择情况如表所示：

编号	个体串（染色体）	x	适应值	百分比%	累计百分比%	选中次数
s_{21}	10010	18	324	23.92%	23.92%	1
s_{22}	11001	25	625	22.12%	46.04%	1
s_{23}	11001	25	625	22.12%	68.16%	1
s_{24}	11110	30	900	31.84%	100%	1

其中若假设按轮盘赌选择时依次生成的4个随机数为0.42、0.15、0.59和0.91，经选择后得到的新的种群为：

$s_{21}=11001$

$s_{22}=10010$

$s_{23}=11001$

$s_{24}=11110$

对第二代种群，其交叉情况如下表：

编号	个体串(染色体)	交叉对象	交叉位	子代	适应值
s_{21}	11001	s_{22}	3	11010	676
s_{22}	10010	s_{21}	3	10001	289

s_{23}	11001	s_{24}	4	11000	576
s_{24}	11110	s_{23}	4	11111	961

这时，函数的最大值已经出现，其对应的染色体为 11111，经解码后可知问题的最优解是在点 $x=31$ 处。