

演化搜索算法 (1)

——遗传算法





6.1 遗传算法的基本概念

- **基本思想**：使用模拟生物和人类进化的方法求解复杂的优化问题，因而也称为**模拟进化优化算法**。
- 遗传算法将**择优**与**随机信息交换**结合在一起，在每一代中，使用上代中最好的，即最适应环境的位或片段，形成新的人工生物集。



6.1 遗传算法的基本概念

- 遗传算法是一个**迭代过程**，在每次迭代中都保留一组候选解，并按某种优劣指标进行排序，然后按某种指标从中选出一些解，利用遗传算子（遗传操作），对其进行运算以产生新一代的一组解。重复上述过程，直到满足指定的收敛要求为止。

6.1.1 遗传算法的基本定义

定义6.1 个体:

个体是一个数据结构，用来描述基本的遗传结构。

定义6.2 适应性:

每个个体有一对应的适应值。在优化问题中，适应值来自于一个估计函数。



6.1.1 遗传算法的基本定义

定义6.3 群体：

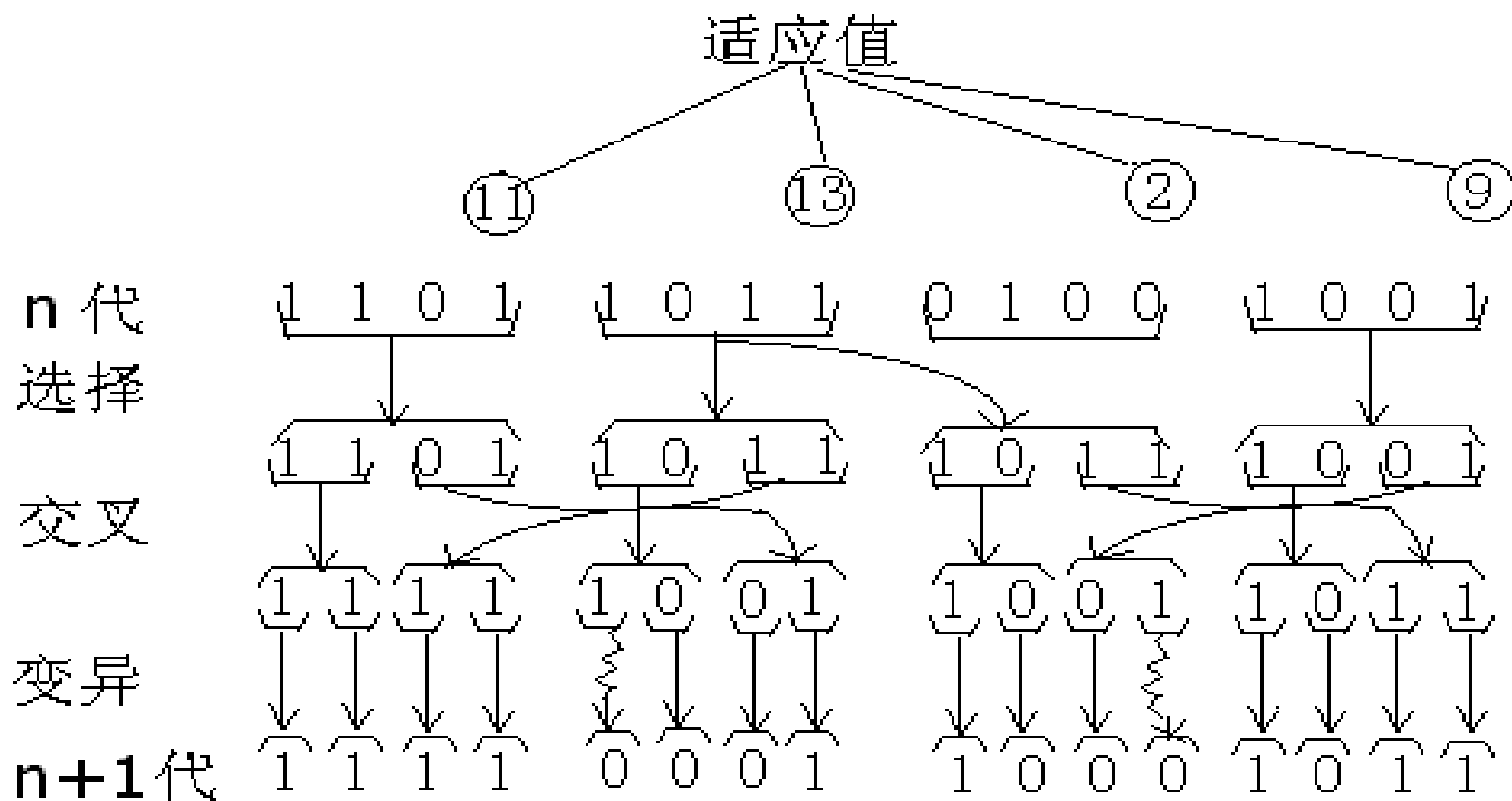
由个体组成的集合。

定义6.4 遗传操作：

遗传操作作用于**群体**而产生新的群体。

标准的代遗传操作一般包括**选择**(或复制)，**交叉**(或重组)和**变异**三种基本形式。

一个简单的遗传操作实例



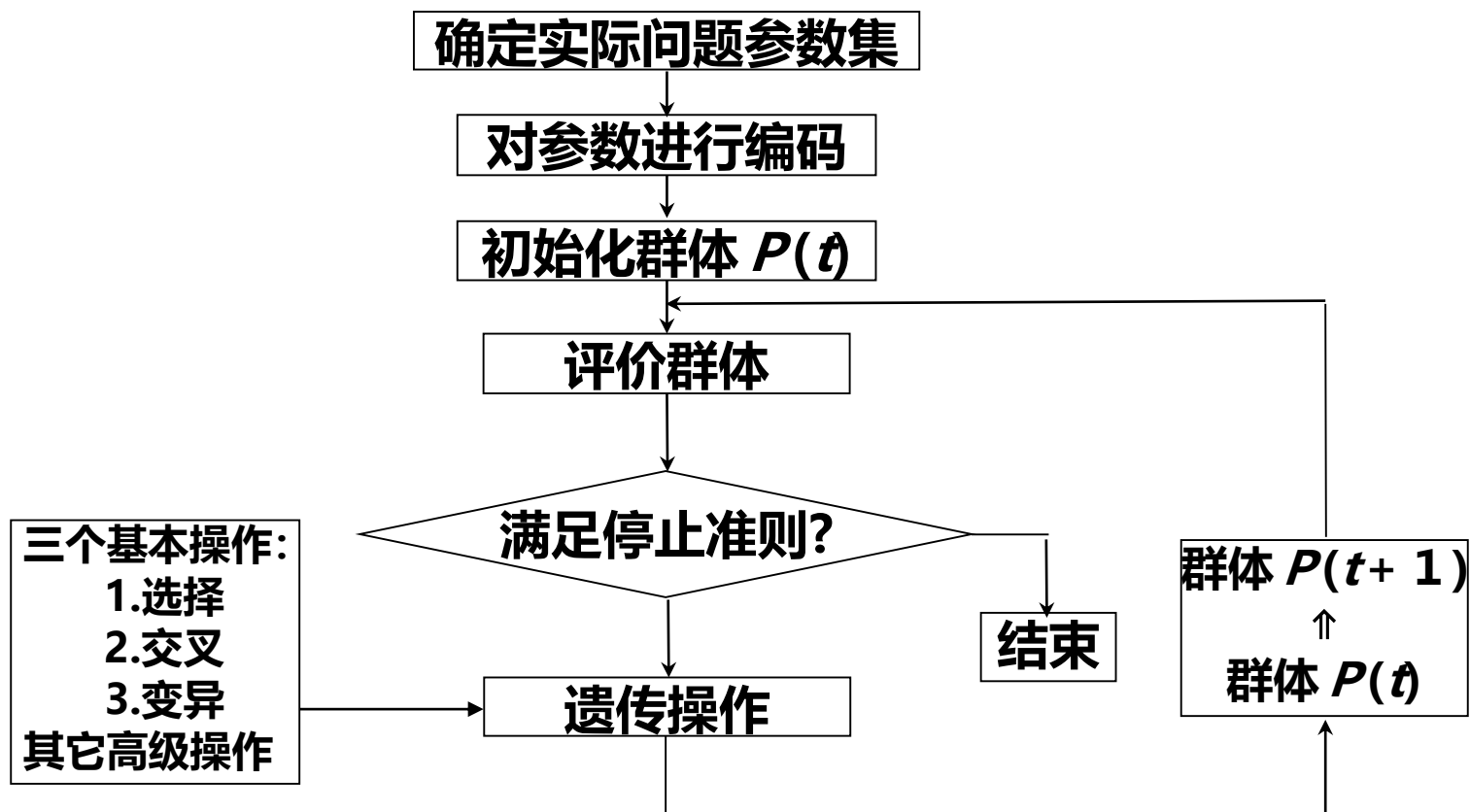


6.1.2 遗传算法的基本流程

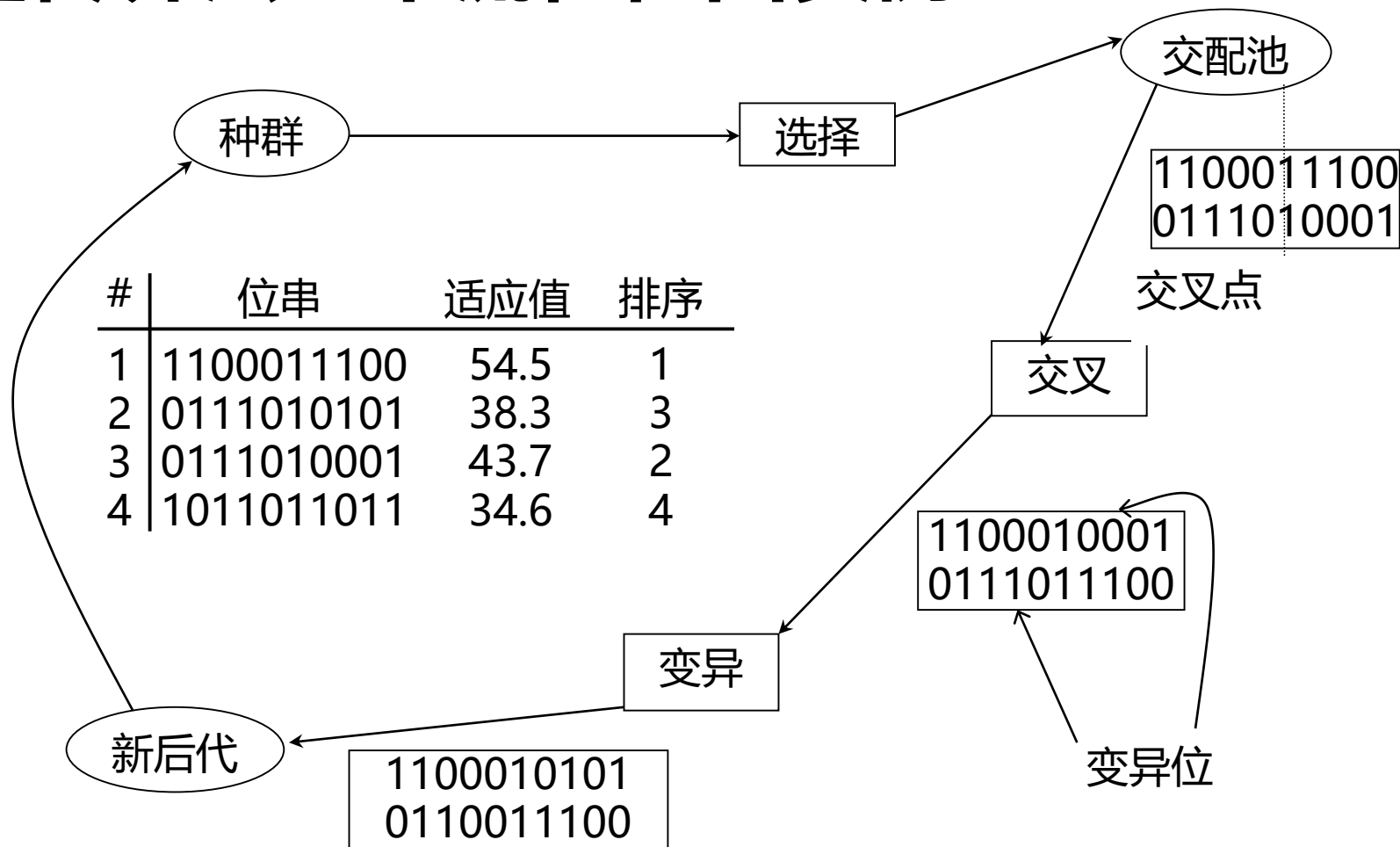
- **遗传算法涉及五大要素：**
 1. 参数编码
 2. 初始群体设定
 3. 适应度函数设计
 4. 遗传操作设计
 5. 控制参数设定

6.1.2 遗传算法的基本流程

标准遗传算法 基本流程框图



标准遗传算法基本流程框图实例





遗传算法的运行过程

1. 选择编码策略，把参数集合 X 和域转换为相应编码空间 S ;
2. 定义适应值函数 $f(X)$;
3. 定义遗传策略，包括选择群体大小、选择、交叉、变异方法以及确定交叉概率 P_c 、变异概率 P_m 等遗传参数;
4. 随机初始化生成群体 $P(t)$;
5. 计算群体中个体的适应值 $f(X)$;
6. 按照遗传策略，运用选择、交叉和变异操作作用于群体，形成下一代群体;
7. 判断群体性能是否满足某一指标，或者已完成预定迭代次数，不满足则返回步骤6)，或者修改遗传策略再返回步骤6)。



6.2 遗传编码

- 二进制编码
- Gray编码
- 实数编码
- 有序编码
- 结构式编码



6.2.1 二进制编码

在二进制编码过程中，首先要确定二进制串的长度 l ，串长 l 取决于变量的定义域及计算所需的精度。

例6.2 变量 x 的定义域为 $[-2, 5]$ ，要求精度为 10^{-6} ，则我们需要将 $[-2, 5]$ 分成至少 7 000 000 个等长小区域，而每个小区域用一个二进制串表示。于是串长至少等于 23，这是因为：

$$4194304 = 2^{22} < 7000000 < 2^{23} = 8388608$$

这样，计算中的任何一个二进制串 $(b_{22}b_{21}\dots b_0)$ 都对应 $[-2, 5]$ 中的一个点。

6.2.2 Gray编码

Gray编码即是将二进制码通过如下变换进行转换得到的码。

设有二进制串 $(\beta_1\beta_2\cdots\beta_n)$, 对应的Gray串为 $(\gamma_1\gamma_2\cdots\gamma_n)$, 则从二进制码到Gray码的变换为

$$\gamma_k = \begin{cases} \beta_1, & \text{如果 } k = 1 \\ \beta_{k-1} \oplus \beta_k, & \text{如果 } k > 1 \end{cases}$$

其中 \oplus 表示模2加法。

从一个Gray码到二进制串的变换为 $\beta_k = \sum_{i=1}^k \gamma_i (\text{mod } 2)$

例6.3 二进制串1101011对应的Gray串为1011110。



6.2.3 实数编码

为了克服二进制编码的缺点，对于问题的变量是实向量的情形，直接可以采用十进制进行编码，这样可以直接在解的表现形式上进行遗传操作，从而便于引入与问题领域相关的启发式信息以增加系统的搜索能力。

6.2.3 实数编码

例3 作业调度问题 (JSP) 的种群个体编码常用 $m \times n$ 的矩阵 $Y=[y_{ij}]$, $i=1, 2, \dots, m$, $j=1, 2, \dots, n$ (n 为从加工开始的天数, m 为工件的优先顺序)。 y_{ij} 表示工件 i 在第 j 日的加工时间。下表是一个随机生成的个体所示。

| | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 |
|-----|----|----|----|----|----|----|----|----|----|
| 工件1 | 0 | 2 | 1 | 2 | 2 | 1 | 2 | | |
| 工件2 | 0 | 1 | 2 | 0 | 0 | 4 | | | |
| 工件3 | 1 | 0 | 0 | 1 | 3 | 1 | | | |
| 工件4 | 0 | 2 | 3 | 0 | 0 | 1 | | | |
| 工件5 | 0 | 2 | 0 | 3 | 0 | 0 | 0 | 1 | 1 |



6.2.4 有序编码

对很多组合优化问题，目标函数的值不仅与表示解的字符串中各字符的值有关，而且与其所在字符串中的位置有关。这样的问题称为有序问题。

若目标函数的值只与表示解的字符串中各字符的位置有关而与其具体的字符值无关，则称为纯有序问题，如采用顶点排列的旅行商问题。

6.2.4 有序编码

例6.4 有10个城市的TSP问题，城市序号为{1, 2, ..., 10}，则编码位串：

1 2 3 4 5 6 7 8 9 10

表示对城市采用按序号升序方法访问行走路线。

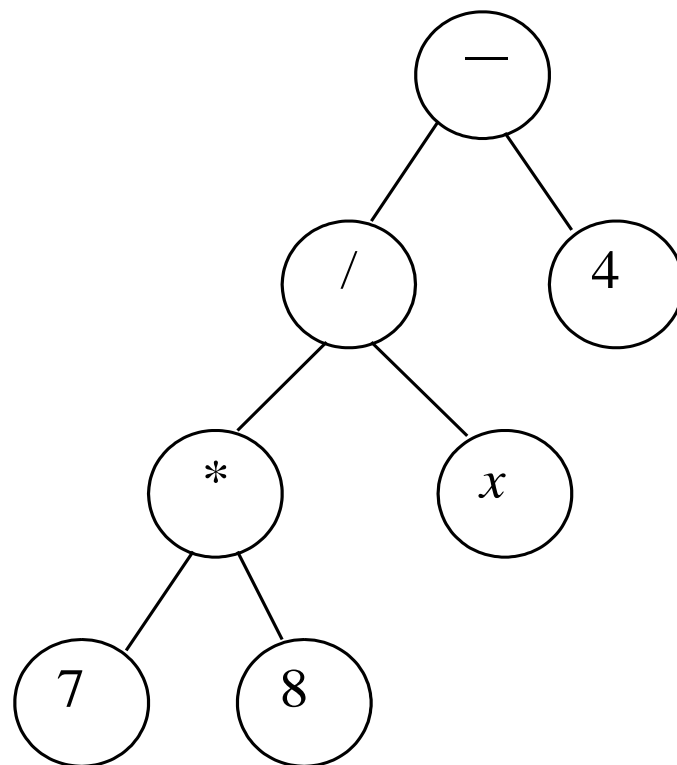
1 3 5 7 9 2 4 6 8 10

表示按特定 “1 → 3 → 5 → 7 → 9 → 2 → 4 → 6 → 8 → 10 → 1”依次访问各个城市。



6.2.5 结构式编码

对很多问题其更自然的表示是树或图的形式，这时采用其它形式的变可能很困难。这种将问题的解表达成树或图的形式编码称为结构式编码。





6.3 适应值函数

- 适应值函数构成了个体的生成环境。根据个体的适应值可以决定它在此环境下的生存能力。
- 若 S^L 表示位串空间, S^L 上的适应值函数可表示为
 $f(\cdot): S^L \rightarrow R^+$, f 为实值函数, R^+ 表示非负实数集合。
- 针对进化过程中关于遗传操作的控制的需要, 选择函数变换 $T: g \rightarrow f$, 使得对于最优解 x^* ,

$$\max f(x^*) = \text{opt } g(x^*) \quad (x^* \in [u, v])$$



6.4 遗传操作

- 选择(selection)
- 交叉操作(crossover)
- 变异操作(mutation)

6.4.1 选择(selection)

选择即从当前群体中选出个体以生成交配池 (mating pool) 的过程。所选出的这些个体具有良好的特征，以便产生优良的后代。

1. 基于适应值比例的选择
2. 基于排名的选择
3. 基于局部竞争机制的选择

1. 基于适应值比例的选择

(1) 繁殖池选择

相对适应值:

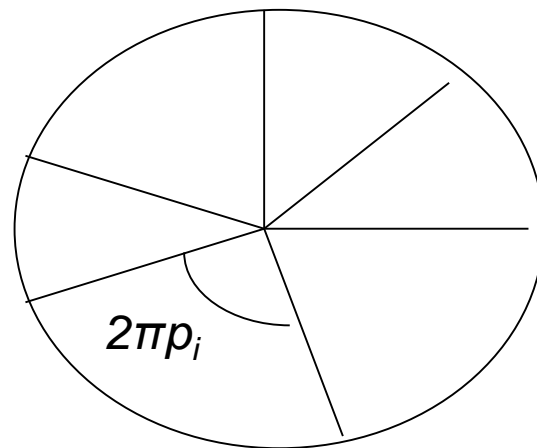
$$rel_i = \frac{f_i}{\sum_{i=1}^N f_i}$$

每个个体的繁殖量: $N_i = \text{round}(rel_i \cdot N)$

1. 基于适应值比例的选择

(2) 转盘赌选择

$$p_i = \frac{f_i}{\sum f_i}$$



现生成一个 $[0,1]$ 内的随机数 r ,

若 $p_1+p_2+\dots+p_{i-1} < r \leq p_1+p_2+\dots+p_i$, 则选择个体 i 。



1. 基于适应值比例的选择

(3) Boltzmann选择

利用函数 $\delta(f_i) = \exp(f_i/T)$ 将适应值进行变换，以改变原始的选择压力。其中 T 是一个控制参数， T 取得较大（小）值时选择具有较小（大）的选择压力，即适应值的相对比例变小（大），通过这个变换之后再按照前面的选择方法进行父体的选择。

2. 基于排名的选择

(1) 线性排名选择

首先假设群体成员按适应值大小从好到坏依次排列为 x_1, x_2, \dots, x_N , 然后根据一个线性函数分配选择概率 p_i 。

设线性函数 $p_i = (a - b \cdot i / (N + 1)) / N$, $i = 1, 2, \dots, N$, 其中 a, b 为常数。由于 $\sum_{i=1}^N p_i = 1$, 易得, $b = 2(a - 1)$ 。又要求对任意 $i = 1, 2, \dots, N$, 有 $p_i > 0$, 且 $p_1 \geq p_2 \geq \dots \geq p_N$, 故限定 $1 \leq a \leq 2$ 。通常使用的值为 $a = 1.1$ 。

2. 基于排名的选择

(2) 非线性排名选择

将群体成员按适应值从好到坏依次排列，并按下式进行分

配选择概率：

$$p_i = \begin{cases} q(1-q)^{i-1}, & i = 1, 2, \dots, N-1 \\ (1-q)^{N-1}, & i = N \end{cases}$$

其中 q 是常数，表示最好的个体的选择概率。



3. 基于局部竞争机制的选择

(1) 锦标赛选择(tournament selection)

选择时，先随机地选择在群体中选择 k 个个体（放回或不放回）进行比较，适应值最好的个体将被选择作为生成下一代的父体。反复执行该过程，直到下一代个体数量达到预定的群体规模。参数 k 称为竞赛规模，一般取 $k=2$ 。



3. 基于局部竞争机制的选择

(2) (μ, λ) 和 $\mu+\lambda$ 选择

(μ, λ) 选择是先从规模为 μ 种群中随机选取个体通过交叉和变异生成 λ ($\geq \mu$) 个后代, 然后再从这些后代中选取 μ 个最优的后代作为新的一代种群。 $\mu+\lambda$ 选择则是从这些后代与其父体共 $\mu+\lambda$ 个后代中选取 μ 个最优的后代。



6.4.2 交叉操作(crossover)

交叉的具体步骤为：

1. 从交配池中随机取出要交配的一对个体；
2. 根据位串长度 L ，对要交配的一对个体，随机选取 $[1, L - 1]$ 中一个或多个的整数 k 作为交叉点；
3. 根据交叉概率 p_c ($0 < p_c \leq 1$) 实施交叉操作，配对个体在交叉点处，相互交换各自的部分内容，从而形成新的一对个体。

对二进制编码常用的交叉算子有**单点交叉**、**多点交叉**和**均匀交叉**。



1. 单点交叉

对于从交配池中随机选择的两个串

$$s_1 = a_{11}a_{12}\dots a_{1l_1} \mathbf{a_{1l_2}\dots a_{1L}}$$

$$s_2 = a_{21}a_{22}\dots a_{2l_1} \mathbf{a_{2l_2}\dots a_{2L}}$$

随机选择一个交叉位 $x \in [1, 2, \dots, L - 1]$, 不妨设 $l_1 \leq x \leq l_2$, 对两个位串中该位置右侧部分的染色体位串进行交换, 产生两个子位串个体为:

$$s'_1 = a_{11}a_{12}\dots a_{1l_1} \mathbf{a_{2l_2}\dots a_{2L}}$$

$$s'_2 = a_{21}a_{22}\dots a_{2l_1} \mathbf{a_{1l_2}\dots a_{1L}}$$

1. 单点交叉

例6.5 考虑如下两个11位变量的父个体：

父个体1: 0 1 1 1 0 0 1 1 0 1 0

父个体2: 1 0 1 0 1 1 0 0 1 0 1

交叉点在位置5，交叉后生成两个子个体：

子个体1: 0 1 1 1 0 1 0 0 1 0 1

子个体2: 1 0 1 0 1 0 1 1 0 1 0

2. 多点交叉

对于选定的两个个体位串，随机选择多个交叉点，构成交叉点集合：

$$x_1, x_2, \dots, x_K \in [1, 2, \dots, L-1], x_k \leq x_{k+1}, k=1, 2, \dots, K-1$$

将 L 个基因划分为 $K+1$ 个基因位集合：

$$Q_k = \{l_k, l_k+1, \dots, l_{k+1}-1\}, k=1, 2, \dots, K+1, l_1=1, l_{K+2}=L+1$$

算子形式为

$$O(p_c, k): a'_{1i} = \begin{cases} a_{2i}, & i \in Q_k, k \text{ 为偶数} \\ a_{1i}, & \text{否则} \end{cases} \quad a'_{2i} = \begin{cases} a_{1i}, & i \in Q_k, k \text{ 为偶数} \\ a_{2i}, & \text{否则} \end{cases}$$

生成的新个体为 $s'_1 = a'_{11}a'_{12}\dots a'_{1L}, s'_2 = a'_{21}a'_{22}\dots a'_{2L}$

2. 多点交叉

例6.6 考虑如下两个11位变量的父个体：

父个体1: 0 1 1 1 0 0 1 1 0 1 0

父个体2: 1 0 1 0 1 1 0 0 1 0 1

交叉点在位置2, 6, 10, 交叉后生成两个子个体：

子个体1: 0 1 1 0 1 1 1 1 0 1 1

子个体2: 1 0 1 1 0 0 0 0 1 0 0

3. 均匀交叉

将位串上的每一位按相同概率随机进行均匀交叉。均匀交叉算子生成的新个体为：

$$s'_1 = a'_{11}a'_{12}\dots a'_{1L}, \quad s'_2 = a'_{21}a'_{22}\dots a'_{2L},$$

其操作描述如下：

$$O(p_c, K) : a'_{1i} = \begin{cases} a_{1i}, & x < 1/2 \\ a_{2i}, & x \geq 1/2 \end{cases}$$

$$a'_{2i} = \begin{cases} a_{2i}, & x < 1/2 \\ a_{1i}, & x \geq 1/2 \end{cases}$$

x 是取值在 $[0, 1]$ 上符合
均匀分布的随机变量

3. 均匀交叉

例6.7

父个体1: 0 1 1 1 0 0 1 1 0 1 0

父个体2: 1 0 1 0 1 1 0 0 1 0 1

模板: 0 1 1 0 0 0 1 1 0 1 0

子个体1: 0 0 1 1 0 0 0 0 0 0 0

子个体2: 1 1 1 0 1 1 1 1 1 1 1

交叉操作产生非法个体的问题

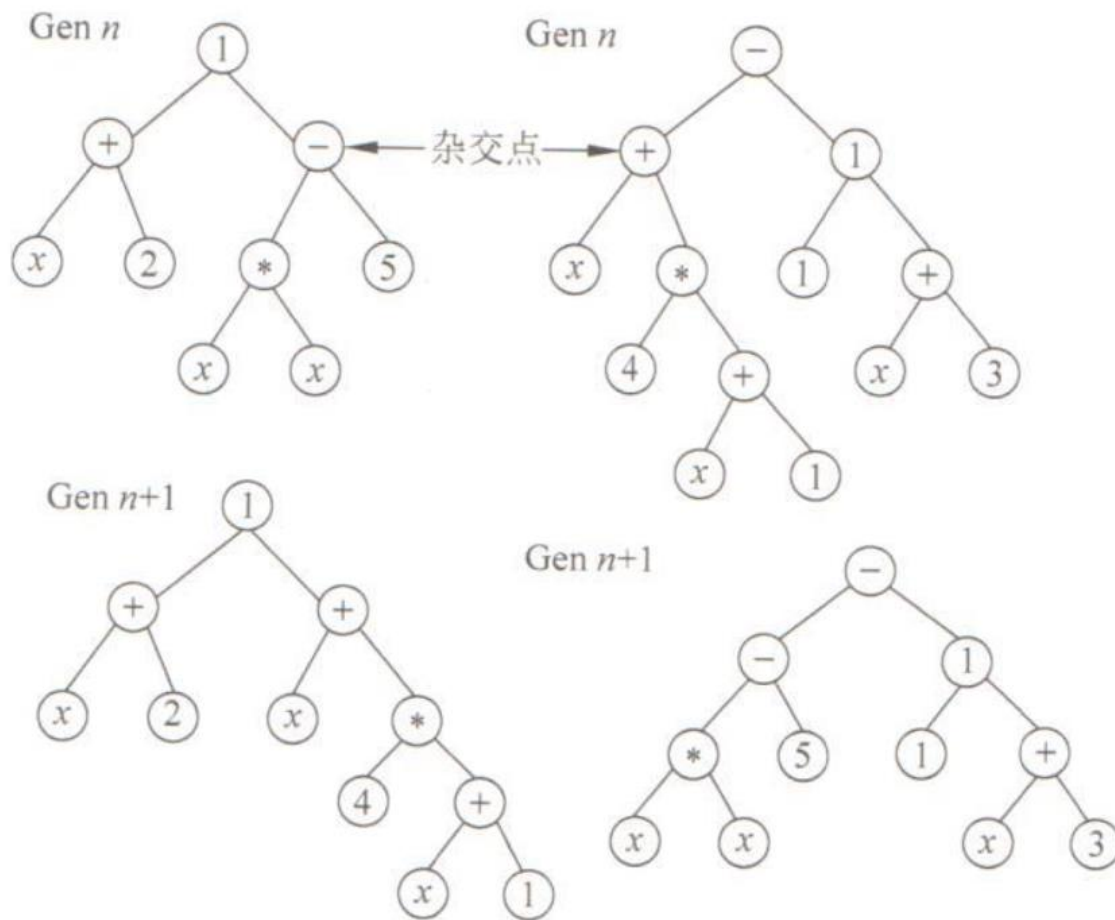
例6.8 设城市数的旅行商问题，对如下的两个个体进行交叉，中间得竖线表示交叉点。

$$\begin{array}{l}
 T_1 = 134 \mid 275 \mid 68 \\
 T_2 = 236 \mid 751 \mid 84
 \end{array}
 \longrightarrow
 \begin{array}{l}
 \tilde{T}_1 = 134 \mid 751 \mid 68 \\
 \tilde{T}_2 = 236 \mid 275 \mid 84
 \end{array}$$

下一代个体都不是合法的个体。怎样保证所产生的个体仍然合法？一种方法是为参与交换的数增加一个映射，将其应用于未交换的等位基因：

$$\begin{array}{l}
 \boxed{1 \rightarrow 5 \rightarrow 7 \rightarrow 2} \longrightarrow
 \begin{array}{l}
 \tilde{T}_1 = 234 \mid 751 \mid 68 \\
 \tilde{T}_2 = 136 \mid 275 \mid 84
 \end{array}
 \end{array}$$

树形编码的交叉算子





6.4.3 变异操作

变异的具体操作为：对种群中任一个体，随机产生一实数，如果大于事先定义的变异概率的阈值，就对该个体进行变异。

1. 实值变异
2. 二进制变异
3. 树形编码变异



6.5 初始化群体

- 初始群体中的个体一般是随机产生的。
- 我们往往希望在问题解空间均匀采样，随机生成一定数目的个体（为群体规模的2倍，即 $2n$ ），然后从中挑出较好的个体构成初始群体。
- 对于二进制编码，染色体位串上的每一位基因在 $\{0, 1\}$ 上随机均匀选择，所以群体初始化至少需要 $L \times n$ 次随机取值。
- 可以证明初始群体的位串译码到问题实空间中也是均匀分布的。

6.6 控制参数的选取

- 主要的参数包括：位串长度 L ，群体规模 n ，交叉概率 p_c ，变异概率 p_m 。
- 参数的最佳建议：
 - $n=20 \sim 200$
 - $p_c=0.6 \sim 1.0$
 - $p_m=0.005 \sim 0.01$



6.7 算法的终止准则

- (1) 预先规定最大演化代数;
- (2) 连续多代后解的适应值没有明显改进, 则终止;
- (3) 达到明确的解目标, 则终止。



6.9 遗传算法简例

例6.9 用GA求解一元函数最大值的优化问题：

$$f(x) = x \sin(10\pi \cdot x) + 2.0 \quad x \in [-1, 2]$$

(1) **编码** 变量 x 作为实数，可以视为遗传算法的表现型形式。现采用二进制编码形式。如果设定求解精度精确到6位小数，由于区间长度为 $2 - (-1) = 3$ ，因此将闭区间 $[-1, 2]$ 分为 3×10^6 等份。因为 $2\,097\,152 = 2^{21} < 3 \times 10^6 < 2^{22} = 4\,194\,304$ 所以编码的二进制串长至少需要22位。

6.9 遗传算法简例

现在采用22位二进制编码，将一个二进制串 $(b_{21}b_{20}\dots b_0)$ 与区间 $[-1, 2]$ 内对应的实数值的建立对应：

$$(b_{21}b_{20}\dots b_0)_2 = \left(\sum_{i=0}^{21} b_i \cdot 2^i \right)_{10} = x'$$
$$x = -1.0 + x' \cdot \frac{2 - (-1)}{2^{22} - 1}$$

6.9 遗传算法简例

例如:一个二进制串 $s_1 = \langle 1000101110110101000111 \rangle$ 表示实数0.637 197

$$x' = (1000101110110101000111)_2 = 2\,288\,967$$

$$x = -1.0 + 2288967 \cdot \frac{3}{2^{22} - 1} = 0.637\,197$$



6.9 遗传算法简例

(2) **产生初始种群** 一个个体由串长为22的随机产生的二进制串组成染色体的基因码，我们可以产生一定数目的个体组成的种群。设产生的4个初始个体如下：

$$s_1 = \langle 1000101110110101000111 \rangle$$

$$s_2 = \langle 000000111000000010000 \rangle$$

$$s_3 = \langle 1110000000111111000101 \rangle$$

$$s_4 = \langle 001000100011011101000 \rangle$$

6.9 遗传算法简例

(3) **计算适应度** 对于个体的适应度的计算，考虑到本例目标函数在定义域内均大于0，而且是求函数的最大值，所以直接引用目标函数作为适应值函数： $f(s)=f(x)$ (这里二进制串 s 对应变量的值)

| 编号 | 个体串 | x | 适应值 | 百分比% | 累计百分比% |
|-------|------------------------|------------|-----------|------|--------|
| s_1 | 1000101110110101000111 | 0.637 197 | 2.586 345 | 29.1 | 29.1 |
| s_2 | 000000111000000010000 | -0.958 973 | 1.078 878 | 12.1 | 41.2 |
| s_3 | 1110000000111111000101 | 1.627 888 | 3.250 650 | 36.5 | 77.7 |
| s_4 | 0010001000110111010010 | -0.599 032 | 1.981 785 | 22.3 | 100 |



6.9 遗传算法简例

(4) **遗传操作** 设按转盘赌方式选择子个体，生成的随机数为0.35, 0.72。则选中的个体为 s_2 和 s_3 。

对 s_2 和 s_3 进行交叉操作，随机选择一个交叉点，例如第5位与第6位之间的位置，交叉后产生新的子个体：

$$s'_2 = \langle 00000 \mid 00000111111000101 \rangle$$

$$s'_3 = \langle 11100 \mid 01110000000010000 \rangle$$

这两个子个体的适应值分别为：

$$f(s'_2) = f(-0.998 \ 113) = 1.940 \ 865$$

$$f(s'_3) = f(1.666 \ 028) = 3.459245$$

交叉后个体 s'_3 的适应值比其父个体的适应值高。

6.9 遗传算法简例

下面考察变异操作。假设已经以一小概率选择了 s_3 的第5个遗传因子（即第5位）变异，遗传因子由原来的0变成1，产生新的个体为

$$s'_3 = \langle 1110100000111111000101 \rangle$$

计算该个体的适应值： $f(s'_3) = f(1.721\ 638) = 0.917\ 743$ ，发现个体的适应值比其父个体的适应值减少了，但是如果选择第10个遗传因子变异，产生的新个体为

$$s''_3 = \langle 1110000001111111000101 \rangle$$

$$f(s''_3) = f(1.630\ 818) = 3.343555$$

显然，这个个体的适应值比其父个体的适应值提高了。这说明变异操作有“扰动”作用。

6.9 遗传算法简例

(5) **模拟结果** 设定种群大小为50, 交叉概率 $p_c=0.25$, 变异概率 $p_m=0.01$, 按照标准的遗传算法SGA, 在运行到第89代时获得最佳个体:

$$s_{\max} = \langle 1111001100111111001011 \rangle$$

$$x_{\max} = 1.850\ 549, \quad f(x_{\max}) = 3.850\ 274$$

这个个体对应的解与微分方程预计的最优解得情况吻合。表6.3列出了模拟一部分代的种群中最佳个体的演变情况 (150代终止)。



模拟世代的种群中最佳个体的演变情况

| 代数 | 个体的二进制串 | x | 适应值 |
|-----|------------------------|-----------|-----------|
| 1 | 1111000110100001110000 | 1.831 624 | 3.534 806 |
| 4 | 1111001010001101100000 | 1.842 416 | 3.790 362 |
| 7 | 1111001110011101010110 | 1.854 860 | 3.833 280 |
| 11 | 1111001110011101010110 | 1.854 860 | 3.833 286 |
| 17 | 1111001011111101010110 | 1.847 536 | 3.842 004 |
| 18 | 1111001011111101110000 | 1.847 554 | 3.842 102 |
| 34 | 1111001101111011000011 | 1.853 290 | 3.843 402 |
| 40 | 1111001100010001001011 | 1.848 443 | 3.846 232 |
| 54 | 1111001100010110110000 | 1.848 699 | 3.847 155 |
| 71 | 1111001101000110110001 | 1.850 897 | 3.850 162 |
| 89 | 1111001100111111001011 | 1.850 549 | 3.850 274 |
| 150 | 1111001100111111001011 | 1.850 549 | 3.850 274 |



6.10遗传算法的应用领域

1. 天然气管道的最优控制
2. 喷气式飞机涡轮机的设计
3. 旅行商问题
4. 作业调度问题
5. 遗传学习
6. 自动控制领域
7. 人工智能与计算机科学
8. 社会和经济领域