3. 遗传算法的变种

堵威

华东理工大学 自动化系

2021.4.1

回顾

- **种群初始化**
  - 生成比N多的个体；局部优化；定向初始化
- **终止准则**
  - 预定代数、足够好、种群不再改进
- **编码方式**
  - 二进制编码、格雷编码
- **精英策略**
  - 每一代生成N-E个子代，每一代生成N个子代
- **稳态与代际算法**
- **种群多样性**
  - 重复个体，多峰问题，小生境，适应度共享，清理，排挤

本章内容

1. 初始化
2. 收敛准则
3. 编码方式
4. 精英策略
5. 稳态与代际算法
6. 种群多样性
7. **选择方案**
8. 交叉
9. 变异

选择方案

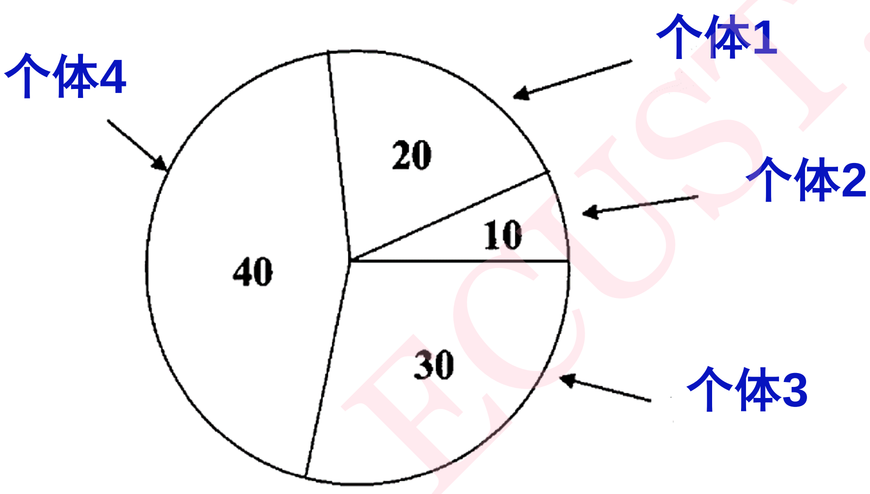
- 基本概念
  - 在生成子代前，需要进行选择父代的操作
  - 除轮盘赌选择外，还有很多选择方法，共同点都是：适应性较强的个体总是比适应性较弱的个体有更大的可能性被选中
  - 选择方法过分偏向适应性强的个体、不能足够地偏向适应性强的个体
  - 选择压力：量化不同选择方法之间的差别

$$\varphi = \mathrm{Pr}(\text{选择适应性最强的个体}) / \mathrm{Pr}(\text{选择平均个体})$$

选择方案

- 随机遍历采样 (Stochastic Universal Sampling)
  - 轮盘赌选择：可能会漏掉最好的个体

例：假设选择4个个体进行交叉操作

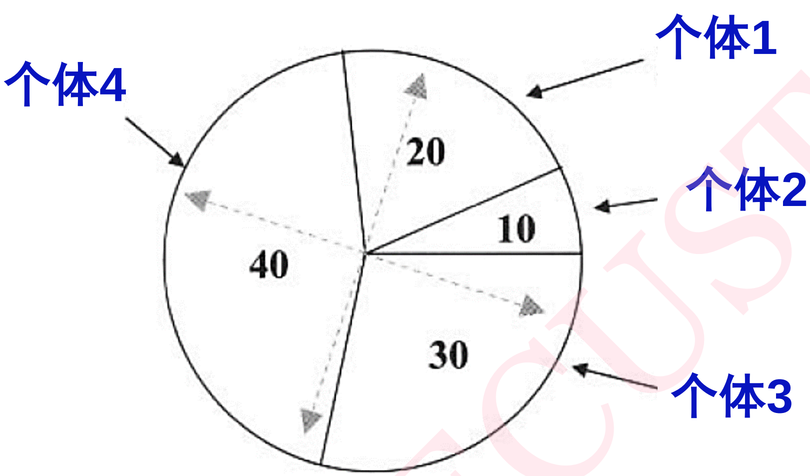


个体4在4次选择中都没有被选中的概率为：0.6^4=13%

即有1/7的概率会失去种群中最好个体的信息

选择方案

- 随机遍历采样
  - 随机遍历采样仍采用轮盘赌方法，但是能解决轮盘赌选择的问题



- 个体1,2,3和4
- 个体1,3,3和4
- 个体2,3,4和4
- 个体1,3,4和4

例：假设选择4个个体进行交叉操作

- 采用带有均匀分布的4个指针的旋转器，旋转一次就能得到4个父代
- 能保证父代中至少一个是个体4，一个是个体3
- 随机遍历采样保证个体x_i被选择的次数在$N_{i,min}$和$N_{i,max}$之间

$$N_{i,\min} = \left\lfloor \frac{N f_i}{f_{\text{sum}}} \right\rfloor \qquad N_{i,\max} = \left\lceil \frac{N f_i}{f_{\text{sum}}} \right\rceil$$

选择方案

- 随机遍历采样
  - 算法伪代码：从N个个体中随机遍历采样选择N个父代

$\mathbf{x}_i$ = 种群中第i个个体，$i\in[1,N]$

$f_i \leftarrow \mathbf{x}_i$的适应度，$i\in[1,N]$

$f_{sum} \leftarrow \sum_{i=1}^{N} f_i$

生成均匀分布随机数 $r\in[0, f_{sum}/N]$

$f_{accum} \leftarrow 0$

$Parents \leftarrow \emptyset$

$k \leftarrow 0$

While $|Parents| < N$

    $k \leftarrow k+1$

    $f_{accum} \leftarrow f_{accum} + f_k$

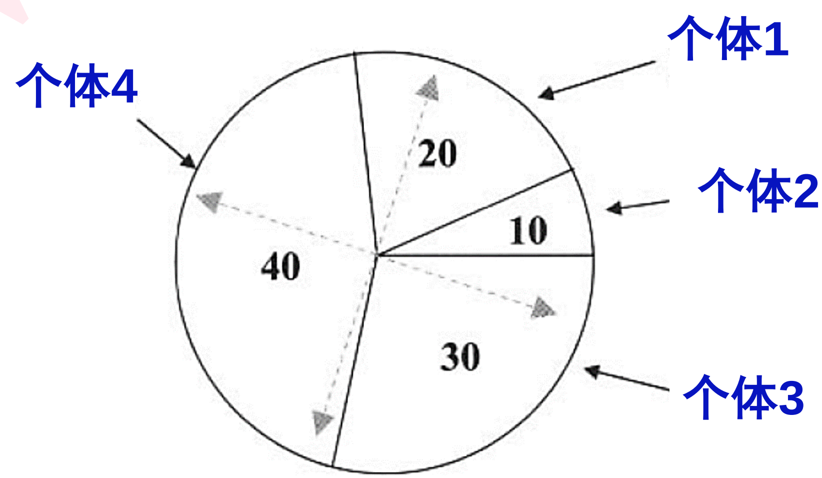
    While $f_{accum} > r$

        $Parents \leftarrow Parents \cup \mathbf{x}_k$

        $r \leftarrow r + f_{sum}/N$

    End while

下一个父代



选择方案

- 超比例选择
  - John Koza在遗传规划的背景下1992年提出
  - 在轮盘赌选择的基础上，通过对高适应性个体的适应度值进行不成比例地加权修改
  - 核心思想：适应度缩放，适应性强的个体具有与其适应度不成比例的更大的概率被选中（精确的百分比不那么重要！）
  - Koza的版本中：最好的32%→80%，最差的68%→20%
  - 进化早期，种群适应度值方差大，选择压力可能会过大；进化后期，种群适应度值方差小，额外的选择压力会有好处

选择方案

- Sigma缩放
  - 让差别较大的适应度值的选择概率更均衡，让聚在一起的适应度值的选择概率更分散

将适应度值相对于整个种群的适应度标准差做归一化

$$f'(x_i) = \begin{cases} \max\left[1 + (f(x_i) - \bar{f})/(2\sigma), \epsilon\right] & \text{if } \sigma \neq 0 \\ 1 & \text{if } \sigma = 0 \end{cases}$$

$\bar{f}$: 适应度的均值，σ: 适应度值的标准差

$$\sigma = \left(\frac{1}{N}\sum_{i=1}^{N}(f(x_i) - \bar{f})^2\right)^{1/2} \qquad \sigma = \left(\frac{1}{N-1}\sum_{i=1}^{N}(f(x_i) - \bar{f})^2\right)^{1/2}$$

ϵ: 用户定义的适应度值在缩放后可允许的非负最小值

选择方案

- Sigma缩放
  - 例：假设种群中有4个个体，适应度值为

$$f(\mathbf{x}_1) = 10,\ f(\mathbf{x}_2) = 5,\ f(\mathbf{x}_3) = 40,\ f(\mathbf{x}_4) = 15$$

轮盘赌选择中个体的选择概率如下：

$$Pr(\mathbf{x}_1) = 14\%,\ Pr(\mathbf{x}_2) = 7\%,\ Pr(\mathbf{x}_3) = 57\%,\ Pr(\mathbf{x}_4) = 22\%$$

计算得到适应度值的均值为：17.5，标准差为：15.5，得到缩放后的适应度值为：

$$f'(\mathbf{x}_1) = 0.76,\ f'(\mathbf{x}_2) = 0.60,\ f'(\mathbf{x}_3) = 1.72,\ f'(\mathbf{x}_4) = 0.92$$

轮盘赌选择中个体的选择概率如下：

$$Pr(\mathbf{x}_1) = 19\%,\ Pr(\mathbf{x}_2) = 15\%,\ Pr(\mathbf{x}_3) = 43\%,\ Pr(\mathbf{x}_4) = 23\%$$

选择方案

- Sigma缩放
  - 例：假设种群中有4个个体，适应度值为

$$f(\mathbf{x}_1) = 15,\ f(\mathbf{x}_2) = 25,\ f(\mathbf{x}_3) = 20,\ f(\mathbf{x}_4) = 10$$

轮盘赌选择中个体的选择概率如下：

$$Pr(\mathbf{x}_1) = 21\%,\ Pr(\mathbf{x}_2) = 36\%,\ Pr(\mathbf{x}_3) = 29\%,\ Pr(\mathbf{x}_4) = 14\%$$

计算得到适应度值的均值为：17.5，标准差为：6.5，得到缩放后的适应度值为：

$$f'(\mathbf{x}_1) = 0.81,\ f'(\mathbf{x}_2) = 1.58,\ f'(\mathbf{x}_3) = 1.19,\ f'(\mathbf{x}_4) = 0.42$$

轮盘赌选择中个体的选择概率如下：

$$Pr(\mathbf{x}_1) = 20\%,\ Pr(\mathbf{x}_2) = 40\%,\ Pr(\mathbf{x}_3) = 30\%,\ Pr(\mathbf{x}_4) = 10\%$$

选择方案

- 基于排名选择
  - 基于适应度值的排名选择，不是基于适应度值选择
  - 将种群中的个体按适应度值从最好到最差排序，最好的个体排名为N（种群规模），最差的个体排名为1
  - 例：某最大化问题中，种群适应度值为

$$f(\mathbf{x}_1) = 15,\ f(\mathbf{x}_2) = 25,\ f(\mathbf{x}_3) = 20,\ f(\mathbf{x}_4) = 10$$

则排名为：

$$R(\mathbf{x}_1) = 2,\ R(\mathbf{x}_2) = 4,\ R(\mathbf{x}_3) = 3,\ R(\mathbf{x}_4) = 1$$

选择方案

- 基于排名选择
  - 例：某最小化问题中，种群适应度值为

$$f(\mathbf{x}_1) = 15,\ f(\mathbf{x}_2) = 25,\ f(\mathbf{x}_3) = 20,\ f(\mathbf{x}_4) = 10$$

基于排名的选择概率如下：

$Pr(\mathbf{x}_1) = 20\%$, $Pr(\mathbf{x}_2) = 40\%$, $Pr(\mathbf{x}_3) = 30\%$, $Pr(\mathbf{x}_4) = 10\%$

轮盘赌选择中个体的选择概率如下：

$Pr(\mathbf{x}_1) = 21\%$, $Pr(\mathbf{x}_2) = 36\%$, $Pr(\mathbf{x}_3) = 29\%$, $Pr(\mathbf{x}_4) = 14\%$

当适应度值聚在一起时，基于排名选择会分散选择概率，在进化算法运行的后期，种群开始收敛时，相似的个体之间会有较大的区别

选择方案

- 基于排名选择
  - 也可以用非线性函数让排名变换，调整选择概率的分散程度，如取排名的平方：

$$R(\mathbf{x}_1) = 4,\ R(\mathbf{x}_2) = 16,\ R(\mathbf{x}_3) = 9,\ R(\mathbf{x}_4) = 1$$

则选择概率为：

$$Pr(\mathbf{x}_1) = 13\%,\ Pr(\mathbf{x}_2) = 53\%,\ Pr(\mathbf{x}_3) = 40\%,\ Pr(\mathbf{x}_4) = 4\%$$

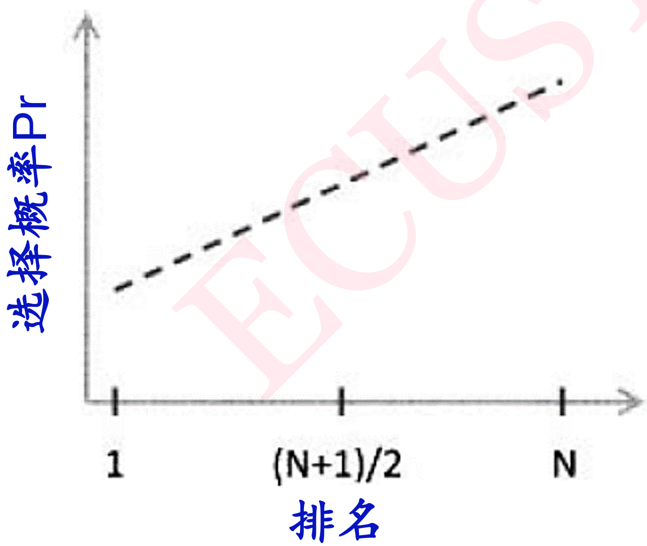
  - 排名的平方：分散
  - 排名的平方根：均匀

选择方案

- 线性排名
  - 是基于排名选择的一般化，在线性排名中将选择个体x_i的概率设置为：

$$\Pr(x_i) = \alpha + \beta R(x_i)$$

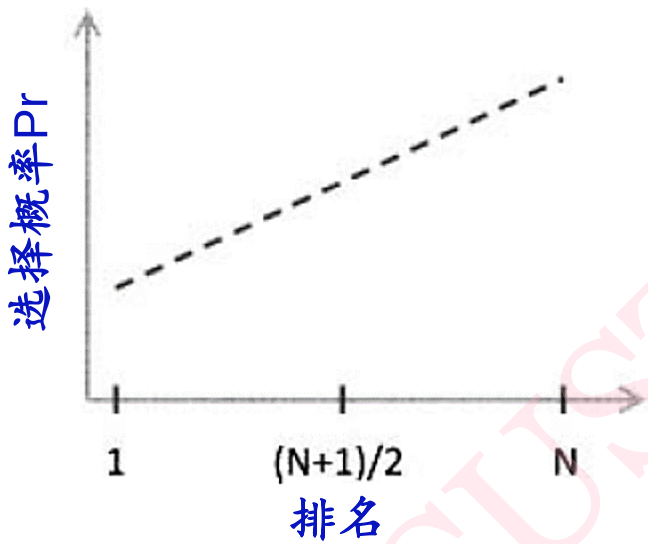
$R(x_i)$是x_i的排名，α和β是自定义参数。



左图中，种群规模为N的进化算法中选择的线性排名方法（最差个体排名为1，最好个体排名为N）

选择方案

- 线性排名



左图：最差个体排名为1，最好个体排名为N，平均个体排名为(N+1)/2，选择压力为：

$$\phi = \frac{\alpha + \beta N}{\alpha + \beta (N+1)/2}$$

将选择概率做归一化处理，让它们的总和为1，得到：

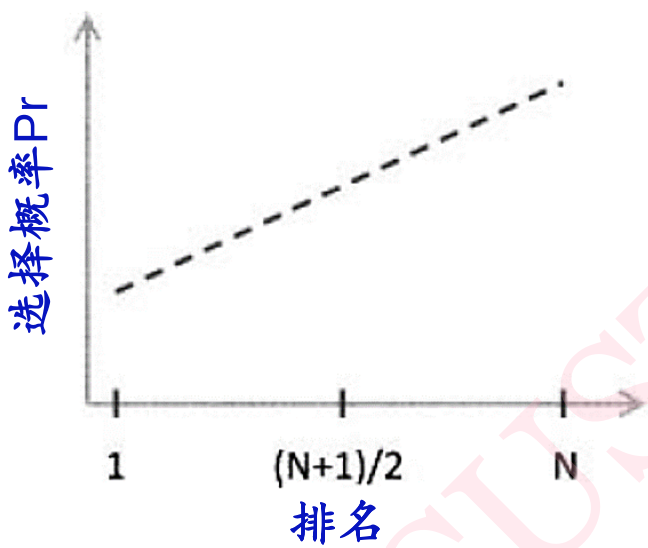
$$\sum_{i=1}^{N} (\alpha + \beta i) = \alpha N + \beta N (N+1)/2 = 1$$

如果想让选择压力为Φ，可以求得：

$$\alpha = \frac{2N - \phi(N+1)}{N(N-1)} \qquad \beta = \frac{2(\phi - 1)}{N(N-1)}$$

选择方案

- 线性排名



Pr(x_i)是R(x_i)的线性函数，有：

Pr(平均的x)=1/2[Pr(最差的x)+ Pr(最好的x)] ≥ 1/2Pr(最好的x)

则：

Φ= Pr(最好的x) / Pr(平均的x) ≤ 2

如果想让Φ > 2，则Pr(最差的x)为负

在进化算法早期，通常想让选择压力较小以避免过早收敛，在后期让选择压力较大以便利用适应性强的个体。

选择方案

- 锦标赛选择
  - 每次从种群中取出一定数量个体，然后选择其中最好的一个进入父代种群。重复该操作，直到新的种群规模达到原来的种群规模。

1. 确定每次选择的个体数量，一般选择2个；
2. 从种群中随机选择2个个体(每个个体入选概率相同) 构成组，根据每个个体的适应度值，选择其中适应度值最好的个体进入父代种群；
3. 重复步骤(2)，直至得到的个体构成新一代种群。

选择的个体越多，选择压力越大/越小？

选择方案

- 种马遗传算法
  - 经典遗传算法选择交叉的父代：依据概率
  - 种马遗传算法：每次交叉操作始终选择每一代最好的个体（种马：每一代最好的个体）
  - 个体1：种马

    个体2：正常的方式选择（基于适应度选择、基于排名选择、锦标赛选择、…）

选择方案

- 种马遗传算法

种马遗传算法的伪代码：

Parents → {随机生成的种群}
While not (终止准则)
    计算种群中每个父代的适应度
    Children ← Ø
    x_1 ← 适应性最强的父代
    While |Children| < |Parents|
        用适应度根据概率**选择**出第二个父代**x_2, $x_2 \neq x_1$**
        父代**交叉**生成子代c_1和c_2
        Children ← Children ∪ {c_1, c_2}
    Loop
    一些子代随机**变异**
    Parents←Children
下一代

选择方案

- 种马遗传算法
  - 例：选出一组20维的测试问题，测试采用种马方案和不采用种马方案的连续遗传算法。种群规模为50，代数为50，变异率为1%，使用参数为2的精英策略

测试函数	非种马	种马算法
Ackley	1.44	1
Fletcher	3.26	1
Griewank	3.96	1
Penalty #1	1.05×10^5	1
Penalty #2	160.8	1
Quartic	9.14	1
Rastrigin	1.92	1
Rosenbrock	3.89	1
Schwefel 1.2	1.24	1
Schwefel 2.21	1.65	1
Schwefel 2.22	3.70	1
Schwefel 2.26	2.56	1
Sphere	4.47	1
Step	4.23	1

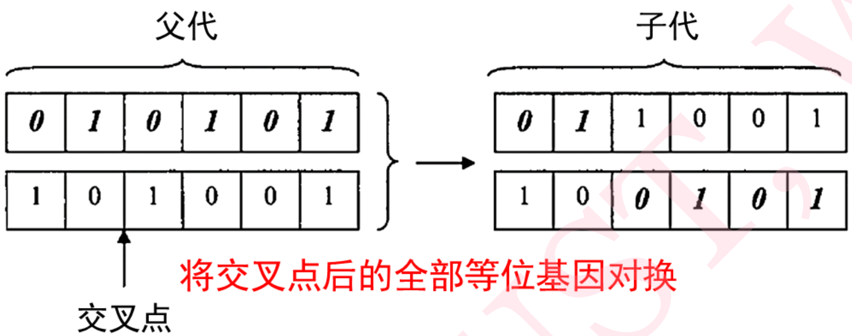
对于这一组基准，种马遗传算法明显优于标准遗传算法

本章内容

1. 初始化
2. 收敛准则
3. 编码方式
4. 精英策略
5. 稳态与代际算法
6. 种群多样性
7. 选择方案
8. **交叉**
9. 变异

交叉

- 基本概念
  - 简单的遗传算法采用单点交叉



交叉点

两个父代	**4.72**	**0.68**	**3.73**	**6.69**	**1.54**
	5.82	1.10	9.22	3.61	8.30

两个子代	**4.72**	**0.68**	**3.73**	3.61	8.30
	5.82	1.10	9.22	**6.69**	**1.54**

  - 还有其他交叉方法
  - 假设种群为 $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$，父代个体 $\mathbf{x}_1 = [x_i(1), x_i(2), \ldots, x_i(n)]$，子代 $y = [y(1), y(2), \ldots, y(n)]$

交叉

- 单点交叉（二进制或连续算法）Single-point crossover
  - 假设有两个父代 $\mathbf{x}_a$ 和$\mathbf{x}_b$

$$\mathbf{x}_a = [x_a(1), x_a(2), ..., x_a(n)]$$

$$\mathbf{x}_b = [x_b(1), x_b(2), ..., x_b(n)]$$

  - m是随机选出的交叉点

$$\begin{array}{lcl} y_1(k) & \leftarrow & \left[\begin{array}{ccc:ccc} x_a(1) & \cdots & x_a(m) & x_b(m+1) & \cdots & x_b(n) \end{array} \right] \\ y_2(k) & \leftarrow & \left[\begin{array}{ccc:ccc} x_b(1) & \cdots & x_b(m) & x_a(m+1) & \cdots & x_a(n) \end{array} \right] \end{array}$$

  - 如果m=0，子代**y**是$\mathbf{x}_b$的克隆；如果m=n，子代子代**y**是$\mathbf{x}_a$的克隆

交叉

- 多点交叉（二进制或连续算法）Multi-point crossover
  - 多个交叉点
  - 假设有两个父代 $\mathbf{x}_a$ 和$\mathbf{x}_b$

$$\mathbf{x}_a = [x_a(1), x_a(2), \ldots, x_a(n)]$$

$$\mathbf{x}_b = [x_b(1), x_b(2), \ldots, x_b(n)]$$

  - 交叉点为m_1和m_2的两点交叉：

$$y(k) \leftarrow \left[\; x_a(1) \quad \cdots \quad x_a(m_1) \;\vdots\; x_b(m_1+1) \quad \cdots \quad x_b(m_2) \;\vdots\; x_a(m_2+1) \quad \cdots \quad x_a(n) \;\right]$$

交叉

- 分段交叉（二进制或连续算法）Segmented crossover
  - 分段交叉可以看成是多点交叉的一般化
  - 子代1从父代1获得它的第一个特征；然后以概率ρ切换到父代2取得子代1的第二个特征，以1-ρ的概率从父代1取得子代1的第二个特征
  - 每次得到子代1的一个特征，就从概率ρ切换到另一个父代取得下一个特征
  - 如果子代1从父代1得到特征k，则子代2从父代2得到特征k
  - 切换概率ρ通常取0.2

交叉

- 分段交叉（二进制或连续算法） Segmented crossover

$S \leftarrow$ true
For $k = 1$ to n
    If S then
        $c_1(k) \leftarrow p_1(k)$
        $c_2(k) \leftarrow p_2(k)$
    else
        $c_1(k) \leftarrow p_2(k)$
        $c_2(k) \leftarrow p_1(k)$
    End if
    $r \leftarrow U[0, 1]$
    If $r < \rho$ then $S \leftarrow$ not S
Next solution feature

交叉

- 均匀交叉（二进制或连续算法）Uniform crossover
  - 假设有两个父代$\mathbf{x}_a$和$\mathbf{x}_b$，均匀交叉得到子代y，其中对每个k∈[1,n], y的第k个特征为：

$$y(k) \leftarrow x_{i(k)}(k)$$

  这里从集合{a, b}随机地选择i(k)，即随机地以50%的概率从它的两个父代选择每一个子代的特征

- 多父代交叉（二进制或连续算法）Multi-parent crossover
  - 又被称为基因库重组、扫描交叉、多性交叉；是均匀交叉的一般化

交叉

- 多父代交叉（二进制或连续算法）Multi-parent crossover
  - 在多父代交叉中，当父代的个数大于2时，从父代随机选择每一个子代的特征
  - 对于每一个k∈[1,n]，由多父代交叉得到：

$$y(k) \leftarrow x_{i(k)}(k)$$

说明：在这里，随机地从[1, N]的子集中选择i(k)，在实施多父代交叉时，需注意：

在潜在父代的池子（**pool**）里应该有多少个体？

应该如何为这个池子选择个体？

池子确定后，如何从池子中选父代？

交叉

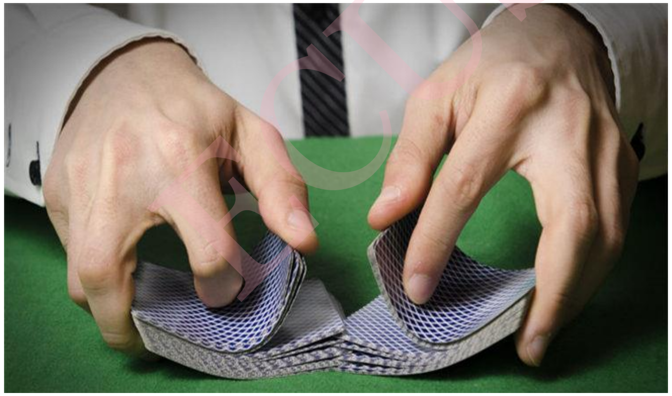
- 全局均匀交叉（二进制或连续算法）Global uniform crossover
  - 多父代交叉的一个特例，此时的父代池为整个种群
  - 对于每一个k∈[1,n]，由全局均匀交叉得到：

$$y(k) \leftarrow x_{i(k)}(k)$$

其中对每一个k随机选择在[1, N]上均匀分布的i(k)；也可以基于适应度来选i(k)，即对所有 k∈[1, n] 和 m∈[1, N]，i(k)=m的概率可以与$\mathbf{x}_m$的适应度成正比

交叉

- 洗牌交叉（二进制或连续算法）Shuffle crossover
  - 洗牌交叉将父代的解的特征随机重新排列
  - 在为给定的子代提供解的特征的所有父代中，解的特征用相同的方式重排，然后采用前面的交叉方法（通常单点交叉）得到子代，再撤除对子代解的特征的重排



交叉

- 洗牌交叉（二进制或连续算法）Shuffle crossover
  - 算法：n维父代与单点交叉结合的洗牌交叉算法

$\{r_1, r_2, ..., r_n\} \leftarrow \{1,2,..., n\}$的一个随机排列
交叉点$m \leftarrow U[1, n-1]$
For $k = 1$ to m
$t_1(k) \leftarrow p_1(r_k)$
$t_2(k) \leftarrow p_2(r_k)$
下一个k
For $k = m+1$ to n
$t_1(k) \leftarrow p_2(r_k)$
$t_2(k) \leftarrow p_1(r_k)$
下一个k
For $k = 1$ to n
$c_1(r_k) \leftarrow t_1(k)$
$c_2(r_k) \leftarrow t_2(k)$
下一个 k

$\mathbf{p}_1$, $\mathbf{p}_2$：两个父代

$\mathbf{t}_1$, $\mathbf{t}_2$：撤除洗牌前的两个子代

$\mathbf{c}_1$, $\mathbf{c}_2$：撤除洗牌后的两个子代

交叉

- 平交叉/算术交叉（连续算法）Flat/Arithmetic crossover
  - 平交叉，也称为算术交叉，描述如下：

$$y(k) \leftarrow U[x_a(k), x_b(k)] = \alpha x_a(k) + (1-\alpha)x_b(k)$$

其中，$\alpha \sim U[0,1]$，即y(k)是来自两个父代第k个特征之间均匀分布的随机数（两个父代特征的线性组合）

  - 平交叉和算术交叉的区别所在：

平交叉得到一个子代，算术交叉得到两个子代

$$y(k) = \alpha x_a(k) + (1-\alpha)x_b(k)$$

$$y_1(k) = \alpha x_a(k) + (1-\alpha)x_b(k)$$
$$y_2(k) = (1-\alpha) x_a(k) + \alpha x_b(k)$$

交叉

- 混合交叉（连续算法）Blended crossover
  - 混合交叉，也称为BLX-α交叉和启发式交叉，描述如下：

$$x_{min}(k) \leftarrow \min(x_a(k), x_b(k)),\ x_{max}(k) \leftarrow \max(x_a(k), x_b(k))$$

$$\triangle x(k) \leftarrow x_{max}(k) - x_{min}(k),\ y(k) \leftarrow U[x_{min}(k)-\alpha\triangle x(k),\ x_{max}(k)+\alpha\triangle x(k)]$$

α是用户定义的参数

若α=0，混合交叉等价于平交叉

若α<0，混合交叉会让搜索域缩小，利于种群开发

若α>0，混合交叉会让搜索域扩大，利于探索

建议α=0.5

交叉

- 线性交叉（连续算法）Linear crossover
  - 线性交叉由父代$\mathbf{x}_a$和$\mathbf{x}_b$生成3个后代：

$$y_1(k) \leftarrow (1/2)x_a(k)+(1/2)x_b(k)$$

$$y_2(k) \leftarrow (3/2)x_a(k)-(1/2)x_b(k)$$

$$y_3(k) \leftarrow (-1/2)x_a(k)+(3/2)x_b(k)$$

为下一代保留这3个后代中适应性最强的一个，或适应性最强的两个

交叉

- 模拟二进制交叉（连续算法）Simulated binary crossover (SBX)
  - SBX由父代$\mathbf{x}_a$和$\mathbf{x}_b$生成2个后代：

$$y_1(k) \leftarrow (1/2)[(1-\beta_k)x_a(k)+(1+\beta_k)x_b(k)]$$

$$y_2(k) \leftarrow (1/2)[(1+\beta_k)x_a(k)+(1-\beta_k)x_b(k)]$$

β_k为由下面的概率密度函数生成的随机数：

$$PDF(\beta) = (1/2)(\eta+1)\,\beta^{\eta}，\text{如果}0\le\beta\le1$$

$$PDF(\beta) = (1/2)(\eta+1)\,\beta^{-(\eta+2)}，\text{如果}\beta>1$$

η是任意的非负实数，推荐η取0~5.

交叉

- 小结
  - 二进制/连续算法：单点交叉、多点交叉、分段交叉、均匀交叉、多父代交叉、全局均匀交叉、洗牌交叉
  - 连续算法：平交叉/算术交叉，混合交叉，线性交叉，模拟二进制交叉
  - 原本都是针对遗传算法提出的，但是也可以用于别的进化算法
  - 这些算法没有绝对的最优者：某个交叉方法可能在一个问题上最好，而另一个交叉方法在另一个问题上最好 ......
  - 单点交叉的性能可以认为最差

本章内容

1. 初始化
2. 收敛准则
3. 编码方式
4. 精英策略
5. 稳态与代际算法
6. 种群多样性
7. 选择方案
8. 交叉
9. **变异**

变异

- 背景知识
  - 二进制算法中，变异是一个简单的操作，每一个个体有n位，变异率为ρ（以概率ρ翻转每一个个体的每一位）
  - 连续算法中，对每一个i和每一个k以概率ρ修改$x_i(k)$，如何修改？
  - 方式1：采用以搜索域中心为均值的均匀分布或高斯分布生成$x_i(k)$
  - 方式2：以非变异的$x_i(k)$的值为均值的均匀分布和高斯分布生成xi(k)

变异

- 以$x_i(k)$为中心的均匀变异
  - 对于i∈[1, N] 和 k∈[1, n]，以$x_i(k)$为中心的均匀变异可以写成：

$$
\begin{aligned}
r &\leftarrow U[0,1] \\
x_i(k) &\leftarrow \begin{cases} x_i(k) & \text{if } r \geq \rho \\ U[x_i(k)-\alpha_i(k),\, x_i(k)+\alpha_i(k)] & \text{if } r < \rho \end{cases}
\end{aligned}
$$

其中$\alpha_i(k)$是用户定义的参数，它确定变异的大小：通常选择尽可能大的$\alpha_i(k)$同时保证变异仍在搜索域内：

$$\alpha_i(k) = \min(x_i(k) - x_{\min}(k),\, x_{\max}(k) - x_i(k))$$

变异

- 以搜索域的中央为中心的均匀变异
  - 对于i∈[1, N] 和 k∈[1, n]，以搜索域的中央为均匀变异可以写成：

$$
\begin{aligned}
r &\leftarrow U[0,1] \\
x_i(k) &\leftarrow \begin{cases} x_i(k) & \text{if } r \geq \rho \\ U[x_{\min}(k), x_{\max}(k)] & \text{if } r < \rho \end{cases}
\end{aligned}
$$

变异

- 以$x_i(k)$为中心的高斯变异
  - 对于i∈[1, N] 和 k∈[1, n]，以$x_i(k)$为中心的高斯变异可以写成：

$$
\begin{aligned}
r &\leftarrow U[0,1] \\
x_i(k) &\leftarrow \begin{cases} x_i(k) & \text{if } r \geq \rho \\ \max\left[\min(x_{\max}(k), N(x_i(k), \sigma_i^2(k)), x_{\min}(k)\right] & \text{if } r < \rho \end{cases}
\end{aligned}
$$

其中，$\sigma_i(k)$是用户定义的参数，与变异的大小成正比。min和max的运算保证$x_i(k)$变异后的值留在搜索域中。

变异

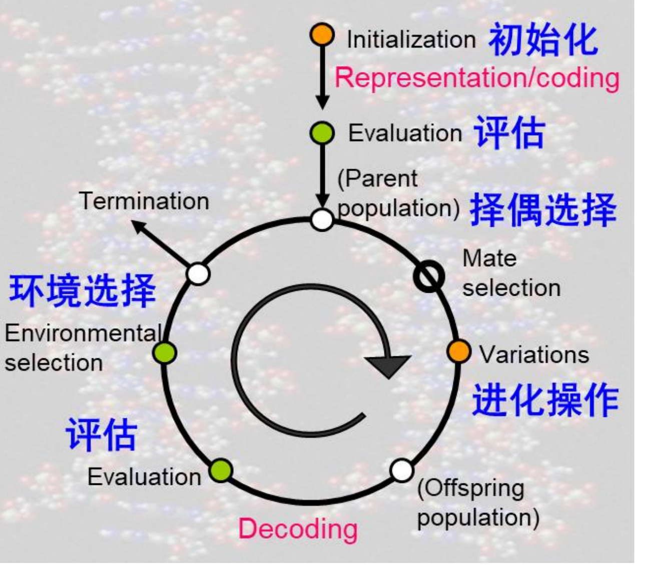
- 以搜索域的中央为中心的高斯变异
  - 对于i∈[1, N] 和 k∈[1, n]，以搜索域的中央为中心的高斯变异可以写成：

$$
\begin{aligned}
r &\leftarrow U[0,1] \\
x_i(k) &\leftarrow \begin{cases} x_i(k) & \text{if } r \geq \rho \\ \max\left[\min(x_{\max}(k), N(c_i(k), \sigma_i^2(k)), x_{\min}(k)\right] & \text{if } r < \rho \end{cases}
\end{aligned}
$$

其中，$c_i(k)=(x_{min}(k)+x_{max}(k))/2$是搜索域的中央， $\sigma_i(k)$是用户定义的参数，与变异的大小成正比。min和max的运算保证$x_i(k)$变异后的值留在搜索域中。

变异

- 本章总结
  - 讨论了遗传算法/进化算法的常用变种
  - 初始化，收敛准则，格雷编码，精英策略，稳态与代际算法，种群多样性，选择，交叉，变异
  - 还有其他变种：种群规模、交互的子种群 ......
  - 进化算法是一个通用的框架，定义优化方法，包括候选解的种群、选择、交叉和变异



结束