3. 遗传算法的变种

堵威

华东理工大学 自动化系

2021.3.18

回顾

- **遗传学的历史、遗传学**
  - 达尔文、孟德尔；基因（显性、隐性）、交配、变异
- **遗传算法的历史**
  - John H. Holland，1975年《自然和人工系统中的适应性》
- **二进制遗传算法**
  - 定义适应度函数
  - 编码
  - 初始化
  - 选择
  - 交叉
  - 变异
  - 评估

Parents $\rightarrow$ {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 $\leftarrow \emptyset$
    While |Children| < |Parents|
        用适应度根据概率选择出一对交叉的父代
        父代交叉生成子代c_1和c_2
        Children $\leftarrow$ Children $\cup \{c_1, c_2\}$
    Loop
    一些子代随机变异
    Parents $\leftarrow$ Children
下一代

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

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

初始化

- 种群初始化
  - –初始化对遗传算法/智能优化算法有着重要的影响
  - –随机初始化：最容易也最常用的种群初始化方法
  - –其他一些初始化方法：假设种群规模为N

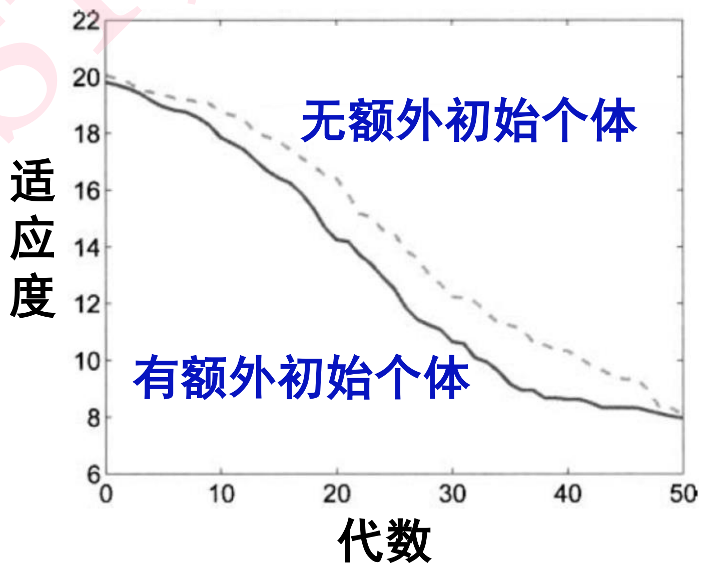
1. 初始化时生成比N更多的个体，但只留下最好的N个作为初始种群。如：初始生成5N个，留下最好的N个。
2. 初始种群后，对每个个体做局部优化。如：初始生成**N**个个体，对这些个体进行梯度下降优化。
3. 定向初始化：结合问题的信息进行种群的初始化。

初始化

- 不同初始化方法对比
  - 在对10维的Rosenbrock函数的优化中，使用两种不同的初始化方法：

  1）无额外初始个体：随机初始化10个个体；

  2）有额外初始个体：随机生成20个个体，取适应度最好的10个个体作为初始种群。



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

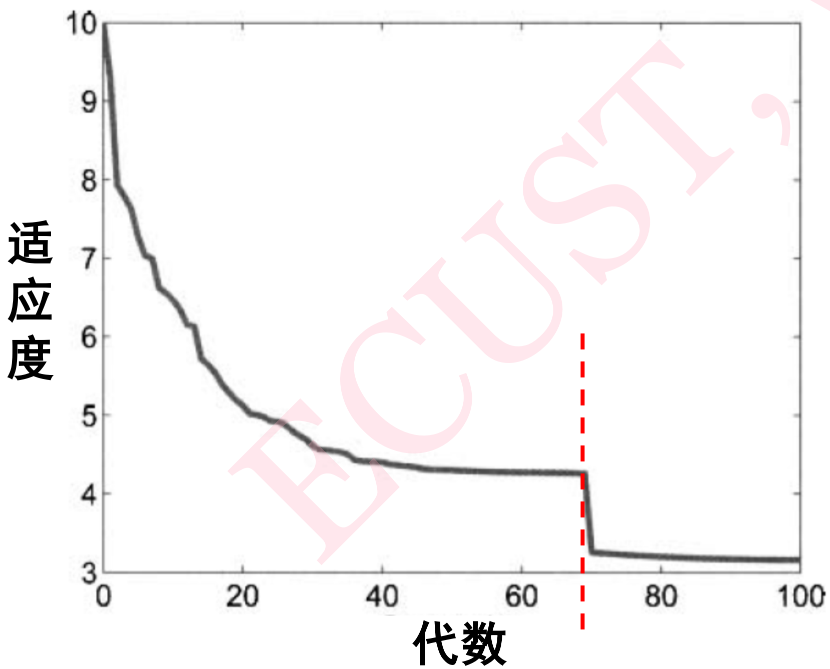
收敛准则

- 优化算法何时停止？
  - While not (终止准则)

1. 在预定的代数之后停止优化算法：简单、常用；注意对于不同算法，预定的函数评价次数
2. 在解“足够好”之后停止优化算法：定义“足够好”，很多时候双倍运行时间只能得到微小的改进
3. 种群不再改进时停止优化算法：一定代数中，最好的个体适应度无明显改变；种群平均适应度无明显改变；种群适应度的标准差不再减小...

收敛准则

- 优化算法何时停止？
  - 最后一条终止条件**“种群不再改进时停止优化算法”**可能会使算法错过更好的解



左图：某个智能优化算法仿真的最好或平均适应度值与代数的关系

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

编码方式

- 二进制编码的不足
  - 考虑0~7这些数的二进制编码：

    000 = 0，001 = 1，010 = 2，**011 = 3**，

    100 = 4，101 = 5，110 = 6，111 = 7.

    3和4相差一个数，但是编码的三位都不相同

    1和5编码只有一位不同，但是表示的数的范围却相差很远

  - 二进制编码中，相邻的数会有不止一位不同；有时表示离得很远的两个数时编码却很相似

编码方式

- 格雷编码
  - 相邻的数只有一个位不同：

二进制编码：

000 = 0, 001 = 1, 010 = 2, 011 = 3,

100 = 4, 101 = 5, 110 = 6, 111 = 7.

格雷编码：

000 = 0, 001 = 1, **011 = 2, 010 = 3,**

110 = 4, 111 = 5, 101 = 6, 100 = 7.

  - 相邻数字的格雷编码总是只有一位不同，格雷编码消除了Hamming悬崖（只有一位不同的编码表示的整数缺大不相同）

编码方式

- 格雷编码举例
  - 例 1：考虑问题：$\min_x f(x)$， 其中 $f(x) = x^4 + 5x^3 + 4x^2 - 4x + 1.$

分别利用4位二进制编码和格雷编码对x∈[-4, 1]的16个均匀间隔的数值编码。

二进制编码：

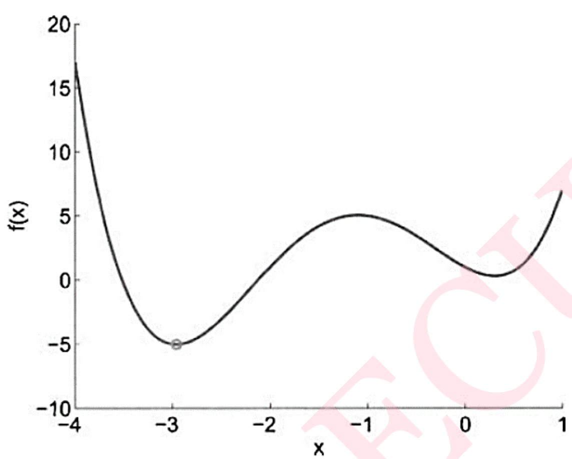
0000 = -4.00, 0001 = -3.67, 0010 = -3.33, 0011 = -3.00,
0100 = -2.67, 0101 = -2.33, 0110 = -2.00, 0111 = -1.67,
1000 = -1.33, 1001 = -1.00, 1010 = -0.67, 1011 = -0.33,
1100 = +0.00, 1101 = +0.33, 1110 = +0.67, 1111 = +1.00.

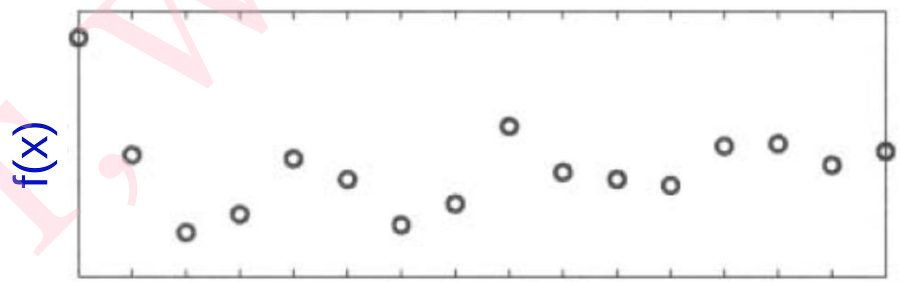
格雷编码：

0000 = -4.00, 0001 = -3.67, 0011 = -3.33, 0010 = -3.00,
0110 = -2.67, 0111 = -2.33, 0101 = -2.00, 0100 = -1.67,
1100 = -1.33, 1101 = -1.00, 1111 = -0.67, 1110 = -0.33,
1010 = +0.00, 1011 = +0.33, 1001 = +0.67, 1000 = +1.00.

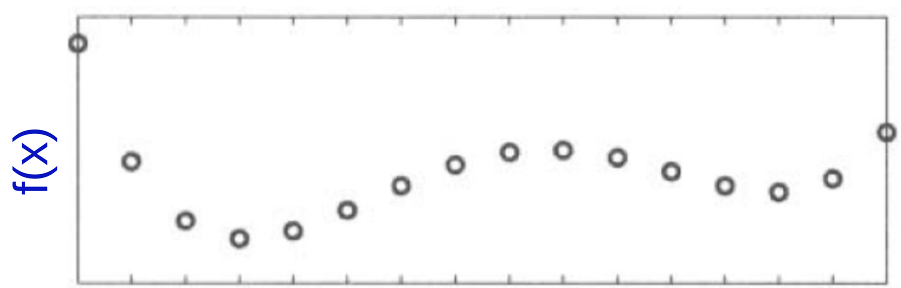
编码方式

- 格雷编码举例
  - 绘制x与f(x)的图，让相邻的x的值的二进制编码/格雷编码只有一位不同





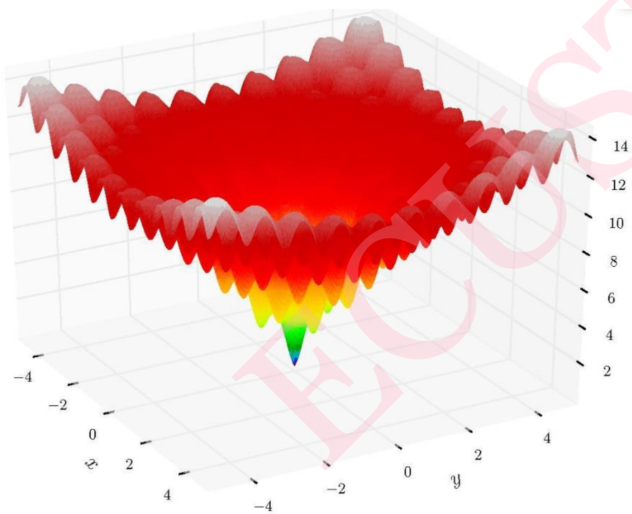
二进制编码x

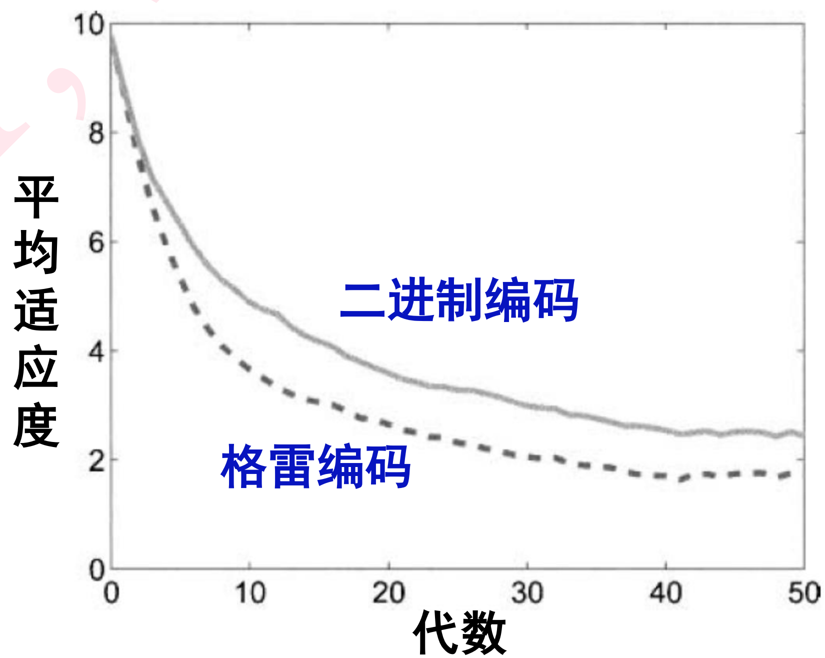


格雷编码x

编码方式

- 格雷编码举例
  - **例 2：** 测试格雷编码和二进制编码对遗传算法性能的影响。用遗传算法优化二维Ackley函数，每一维用6位编码，种群规模为20，每一代每位的变异率为2%。





编码方式

- 格雷编码举例
  - 在最坏情况问题（离散问题，搜索空间中有一半的点都是局部最小点）中，二进制编码的效果更好
  - **例 3：** 最小化一个最坏情况问题，该问题：当二进制编码的值为偶数时，适应度为1；值为奇数时，适应度为2。

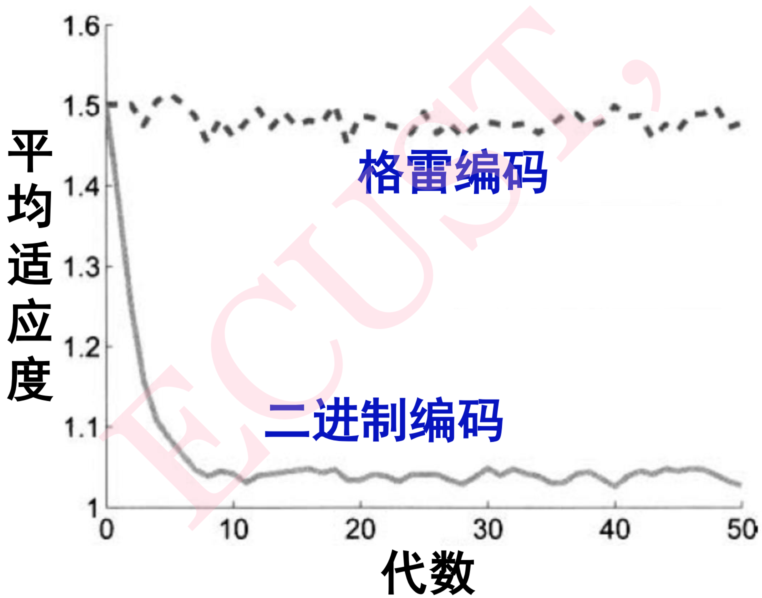
二进制编码： f(000)= 1, f(001) = 2, f(010) = 1, f(011) = 2,
f(100)= 1, f(101) = 2, f(110) = 1, f(111) = 2.

格雷编码： f(000)= 1, f(001) = 2, f(011) = 1, f(010) = 2,
f(110)= 1, f(111) = 2, f(101) = 1, f(100) = 2.

二进制编码：适应度高的个体之间交叉仍可以得到适应度高的子代

编码方式

- 格雷编码举例
  - **例 4：** 测试遗传算法在20位最坏情况问题上的性能。当二进制编码的值为偶数时，适应度为1；值为奇数时，适应度为2；种群规模为20，每一代每位的变异率为2%。



二进制编码比格雷编码更擅于找到多个局部最小

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

精英策略

- 经典遗传算法
  - 经典遗传算法：不能保证第i代的优秀个体 x_e 进化到第i+1代
  - 精英/精英策略（elitism）：确保优化算法中最好的个体从一代到下一代能保留在种群中的方式

假设种群规模为N，精英个数为E

1. 每一代只生成N-E个子代 + 前一代种群中最好的E个个体
2. 每一代生成N个子代，将其中最差的E个用前一代最好的E个个体替换
3. 其他方案：被替换的E个个体通过反轮盘赌选择方式挑选

精英策略

- 精英策略伪代码
  - **精英方案1：** N是种群规模，E是精英个数，每一代生成N-E个子代（基于遗传算法）

假设种群规模为N，精英个数为E，每一代生成N-E个子代

```
Parents ← {随机生成的种群}
While not (终止准则)
    计算在种群中每一个父代的适应度
    Elites ← 最好的E个父代
    Children ← Ø
    While |Children| < |Parents| - E
        利用适应度根据概率选择一对父代
        父代交叉生成子代c1和c2
        Children ← Children ∪ {c1, c2}
    Loop
    随机变异一些子代
    Parents ← Children ∪ Elites
下一代
```

精英策略

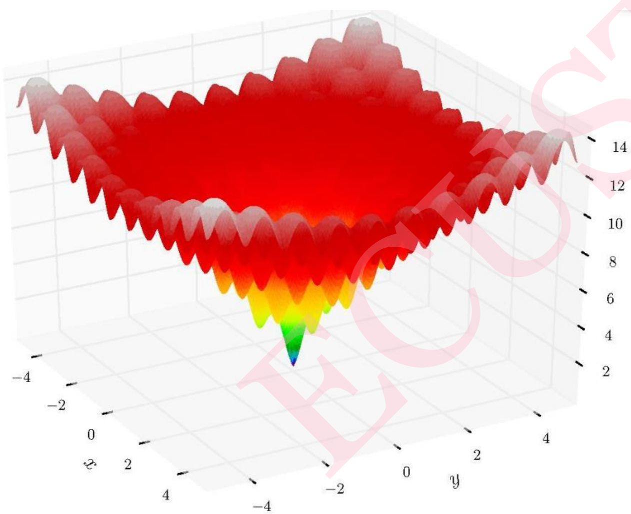
- 精英策略伪代码
  - **精英方案2：**N是种群规模，E是精英个数，每一代生成N个子代（基于遗传算法）

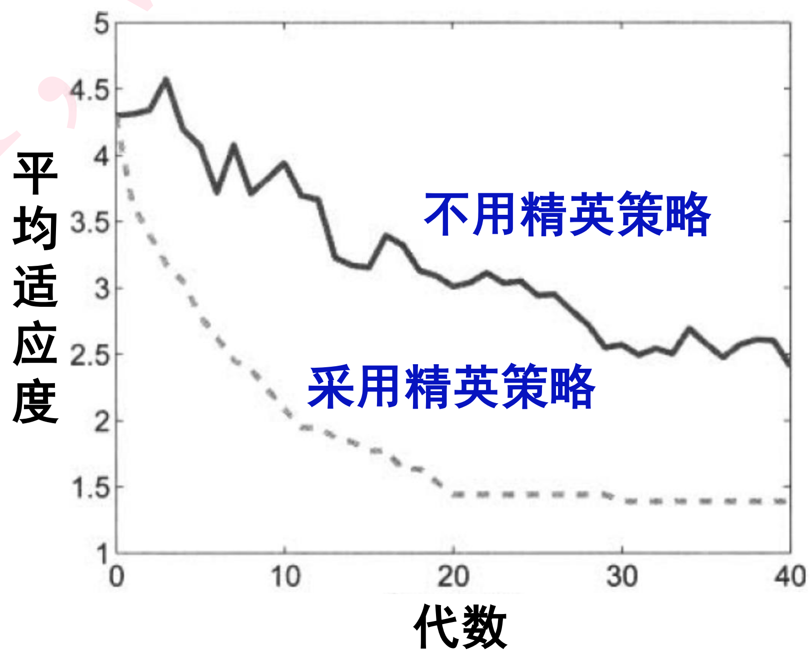
假设种群规模为N，精英个数为E，每一代生成N个子代

Parents ← {随机生成的种群}
While not (终止准则)
    计算在种群中每一个父代的适应度
    Elites ← 最好的E个父代
    Children ← Ø
    While |Children| < |Parents|
        利用适应度根据概率选择一对父代
        父代交叉生成子代c_1和c_2
        Children ← Children ∪ $\{c_1, c_2\}$
    Loop
    随机变异一些子代
    Parents ← Children ∪ Elites
    Parents ← 最好的N个父代
下一代

精英策略

- 精英策略举例
  - **例：** 用遗传算法优化二维Ackley函数，每一维用6位二进制编码，每一代每位的变异率为2%。采用精英方案1，种群规模为N=20，精英个数E=2。



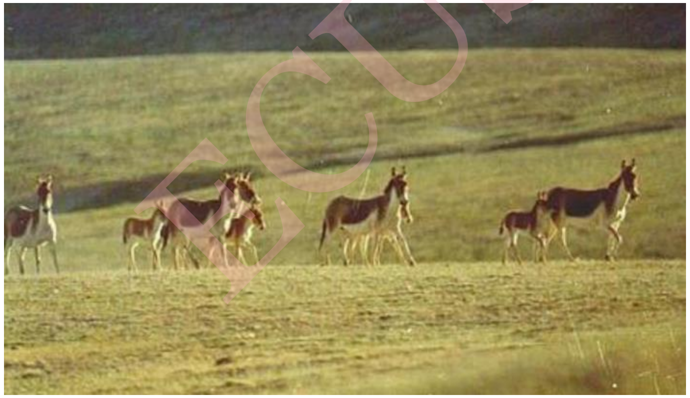
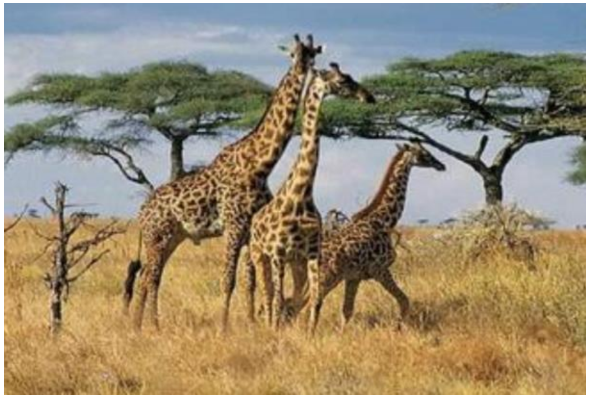


本章内容

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

稳态与代际算法

- 基本概念
  - 代际算法(generational)：算法中每一代整个种群都会被替代；如经典遗传算法，但是精英策略是例外
  - 自然进化：自然中的代会相互交错，死亡与新生连续发生（稳态的进化）
  - 稳态算法(steady-state)：算法中每一代若干个体被替代

稳态与代际算法

- 稳态算法
  - 稳态算法举例：假设每一代只生成两个子代，并用这两个子代替换它们在种群中的两个父代

伪代码：

Parents ← {随机生成的种群}

计算种群中每一个父代的适应度

While not (终止准则)

　　利用适应度根据概率选择一对父代p_1和p_2进行交叉

　　父代交叉生成子代c_1和c_2

　　随机变异c_1和c_2

　　计算c_1和c_2的适应度

　　$p_1 \leftarrow c_1$, $p_2 \leftarrow c_2$　仅当c_1/ c_2比p_2/p_2好时替换

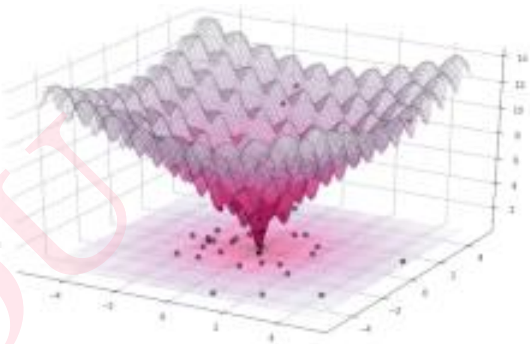
下一代

每一代替换个体数0~N

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

种群多样性



- 基本概念
  - 多样性(diversity)：描述种群的差异性（相似个体，重复个体），是影响算法性能（避免早熟）的重要因素
  - 智能优化算法中，一代代的反复交叉，种群均一性，限制了算法对搜索空间的深入搜索
  - 过早收敛：在找到优化问题的一个令人满意的解之前就出现均一性
  - 防止过早收敛：提高变异率（双刃剑？）

种群多样性

- 重复个体
  - 常用的方法：持续搜索种群中重复的个体并替换它们

1. 每生成一个子代，扫描种群以确保生成的不是一个复制品（再进行选择、交叉，或进行变异）
2. 每当个体变异时，扫描种群以确保生成的不是一个复制品（再进行变异）
3. 每一代结束时，扫描种群找出复制品（随机生成，或再变异）
4. 在种群中允许一些复制品存在，但不能超过设定的阈值（保证高适应性的个体）

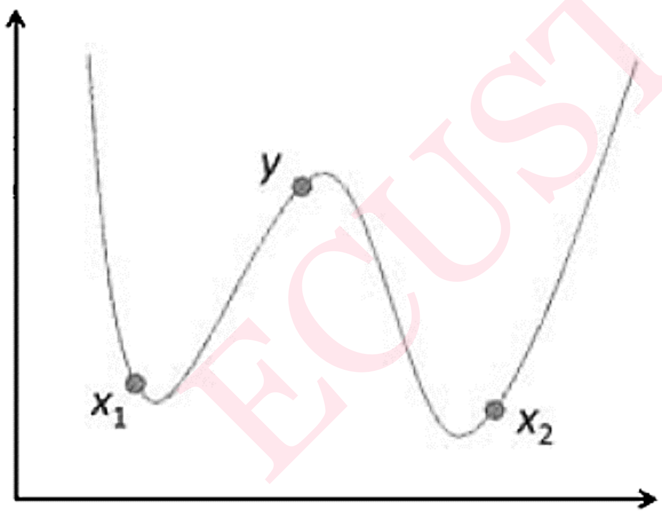
计算复杂度：$O(N^2)$，N为种群规模，实际问题？测试问题？

隔G代扫描

种群多样性

- 不同物种间/近亲间的交配
  - 生物进化中，很少看到不同物种间的交配/近亲间的交配
  - 典型的智能优化算法中，并没有特别考虑父代间的差异/相似性

父代间的差异性



左图：多峰最小化问题

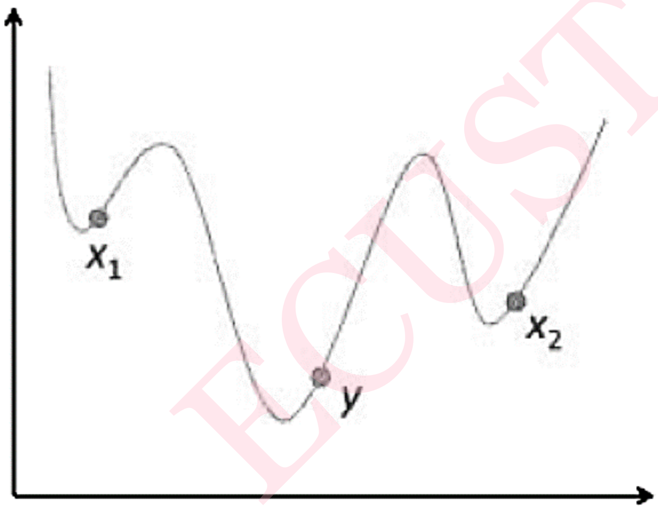
两个适应度强且非常不同的个体x_1和x_2交叉后出现的问题

交叉后得到的点可能是一个差的解，会失去问题解的重要遗传信息

种群多样性

- 不同物种间/近亲间的交配
  - 生物进化中，很少看到不同物种间的交配/近亲间的交配
  - 典型的智能优化算法中，并没有特别考虑父代间的差异/相似性

父代间的相似性



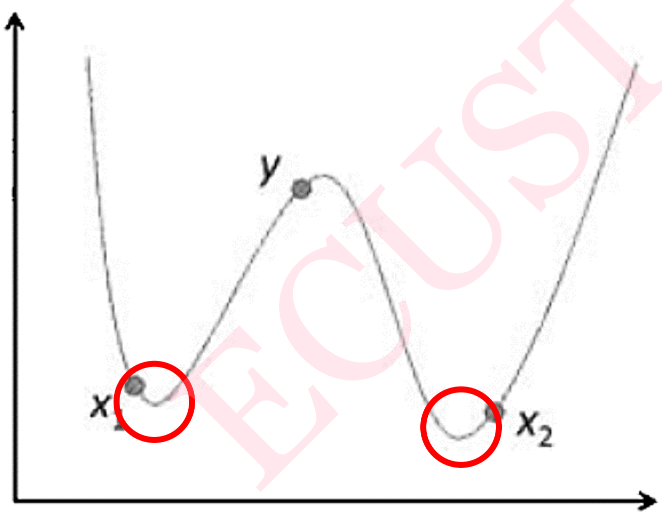
左图：多峰最小化问题

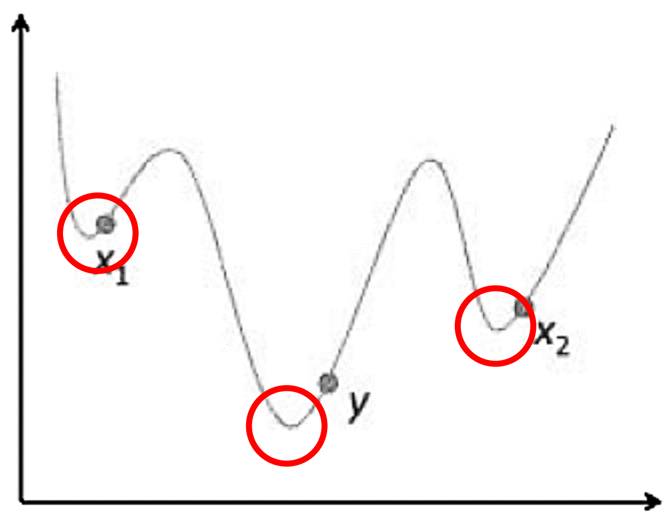
不允许差别很大的个体x_1和x_2交叉会出现的问题

很难找到y附近的全局最小值，进化搜索过程可能会原地踏步

种群多样性

- 多峰问题
  - 有不止一个局部最优解的优化问题
  - 要让靠近局部最优解的个体保留下来：希望相似的个体聚集在一起



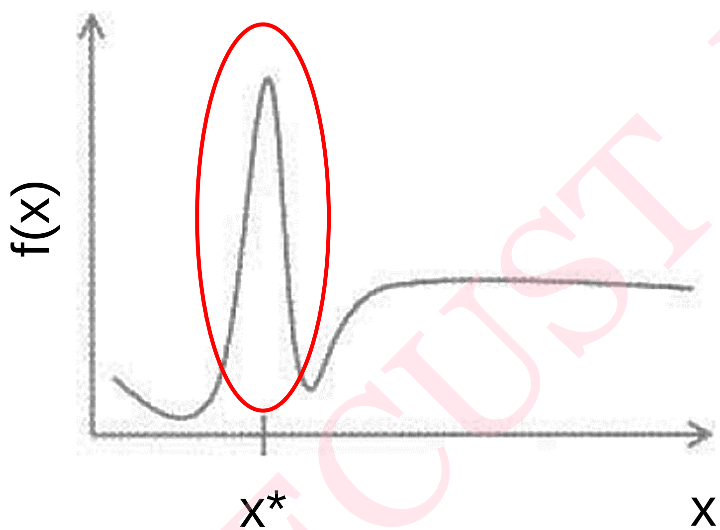


种群多样性

- 小生境 (niching)
  - –小生境是解决多峰问题的有效方法
  - –生物学上，小生境指特定环境下的一种组织结构：在自然界中，生物总是喜欢与自己特征、性状相似的生物生活在一起
  - –对于多峰问题，利用小生境思想将每一代种群划分为若干类，每个类中选出若干适应度较大的个体作为一个类的优秀代表
  - –求解多峰问题的主要小生境策略：适应度共享、清理、排挤

种群多样性

- 适应度共享 (fitness sharing)
  - 适应度好的个体，过早收敛，多样性



- x*是全局最大值，按理x*附近的个体应该多被选中参与交叉等操作
- x*附近的个体适应度值较低
- 右侧多样性低

为鼓励多样性，可以人为地让较特别的个体适应度值增大，让较普通的个体适应度值减小

种群多样性

- 适应度共享 (fitness sharing)
  - 适应度共享会让搜索空间相互接近的个体适应度值减小（生物学上，相同的地理区域内，相似的个体会争夺相似的资源）

假设智能优化算法的种群有N个个体$\{x_i\}$，f_i是x_i的适应度

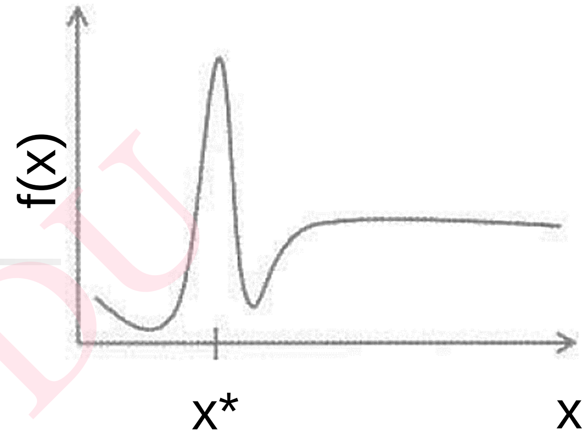
1. 定义共享函数s(·):

$$s(d) = \begin{cases} 1 - (d/\sigma)^{\alpha} & \text{if } d < \sigma \\ 0 & \text{otherwise} \end{cases}$$

d_{ij}度量个体x_i和x_j的距离，σ是自定义参数，称为共享半径，α为共享参数，通常取1

共享函数：体现种群两个个体的密切关系程度

种群多样性



- 适应度共享 (fitness sharing)

假设智能优化算法的种群有N个个体{x_i}，f_i是x_i的适应度

2. 定义共享度m(·)：

$$m_i = \sum_{j=1}^{N} s(d_{ij})$$

共享度：种群中某个个体在群体中共享程度的一种度量，表征与x_i相似的个体的个数

3. 适应度值修改：

$$f_i' = f_i / m_i$$

适应度共享：在同一小生境中的个体分享适应度值

种群多样性

- 清理 (clearing)
  - 清理：减小某些个体的适应度（与适应度共享类似）

1. 定义种群中每一个个体x_i的小生境集合D_i：

$$D_i = \{x_j : d_{ij} < \sigma\}$$

d_{ij}：度量个体x_i和x_j的距离，σ：自定义参数；

2. 根据每一个小生境个体的适应度，确定个体的排名：

r_{ki} = x_k在D_i中的排名

每个小生境中最好的个体排名为1，第二好的为2，以此类推

种群多样性

- 清理 (clearing)

3. 定义R为每一个小生境中需要存活的个体数，并利用下面的算法得到修改后的适应度值：

For $i = 1$ to N
    For $k = 1$ to $|D_i|$
        If $r_{ki} \leq R$ then
            $f'_k \leftarrow f_k$
        else
            $f'_k \leftarrow -\infty$
        End if
    下一个小生境
下一个个体

- 算法保证每一个小生境中，适应性最差的个体不会被选中或参与进化
- 但不能保证选出适应度最强的个体（一个个体可能不止属于一个小生境？）

种群多样性

- 排挤 (crowding)
  - 模仿自然界中的资源竞争：由交叉产生的相似个体替换种群中的个体
  - 三种主要类型：标准排挤、确定性排挤、受限锦标赛选择
  - **标准排挤：**
    - 与稳态算法结合
    - 每一代产生M个子代，然后将这些子代与随机选出的C_f个父代比较，每一个子代替换这C_f个父代中与其最相似的个体（C_f：排挤因子）
    - $M = N/10$, $C_f = 3$（N为种群规模）

种群多样性

- 排挤 (crowding)

标准排挤的稳态遗传算法

父代$\{p_k\} \leftarrow$ {随机生成N个个体的种群}

计算每一个父代p_k，$k \in [1,N]$ 的适应度

While not (终止准则)

  利用适应度依概率选择M个父代交叉

  父代交叉生成M个子代c_i, $i \in [1,M]$

  随机变异每一个子代c_i, $i \in [1,M]$

  计算每一个子代c_i, $i \in [1,M]$ 的适应度

  For i = 1 to M

    从父代种群$\{p_k\}$随机挑选出C_f个个体I

    $p_{min} = \underset{p}{\operatorname{argmin}} \|p - c_i\| : p \in I$ $\|p-c_i\|$：定义的距离函数

    $p_{min} \leftarrow c_i$

  下一个子代

下一代

种群多样性

- 排挤 (crowding)
  - **确定性排挤：**父代交叉生成子代，每个子代替换与之最相似的父代（仅在子代的适应度优于父代时）

Parents ← {随机生成的种群}
计算种群中每一个父代的适应度
While not (终止准则)
    利用适应度依概率选择一对父代$\mathbf{p}_1$和$\mathbf{p}_2$
    父代交叉生成两个子代$\mathbf{c}_1$和$\mathbf{c}_2$
    随机变异子代$\mathbf{c}_1$和$\mathbf{c}_2$
    计算子代$\mathbf{c}_1$和$\mathbf{c}_2$的适应度
    For i = 1 to 2
        If $\|\mathbf{p}_1-\mathbf{c}_i\| < \|\mathbf{p}_2-\mathbf{c}_i\|$ 并且 fitness($\mathbf{c}_i$)＞fitness($\mathbf{p}_1$) then
            $\mathbf{p}_1 \leftarrow \mathbf{c}_i$
        Elseif $\|\mathbf{p}_2-\mathbf{c}_i\| < \|\mathbf{p}_1-\mathbf{c}_i\|$ 并且 fitness($\mathbf{c}_i$)＞fitness($\mathbf{p}_2$) then
            $\mathbf{p}_2 \leftarrow \mathbf{c}_i$
        Endif
    下一个子代
下一代

$\|\mathbf{p}-\mathbf{c}_i\|$：定义的距离函数

最大化问题

种群多样性

- 排挤 (crowding)
  - **受锦标赛选择：** M个父代生成M个子代，将子代与随机选出的C_f个个体比较，每一个子代替换这些个体中与之最相似的个体（仅在子代的适应度优于父代时）

Parents ← {随机生成的种群}
计算种群中每一个父代的适应度
While not (终止准则)
    利用适应度依概率选择M个父代
    父代交叉生成M个子代$\mathbf{c}_i$, $i\in[1,M]$
    随机变异每一个子代$\mathbf{c}_i$, $i\in[1,M]$
    计算每一个子代$\mathbf{c}_i$, $i\in[1,M]$ 的适应度
    从父代种群{$\mathbf{p}_k$}随机挑选出C_f个个体I
    For i = 1 to M
        $\mathbf{p}_{min}=\arg\min_{\mathbf{p}}\|\mathbf{p}-\mathbf{c}_i\|: \mathbf{p}\in I$
        If fitness($\mathbf{c}_i$) > fitness($\mathbf{p}_{min}$) then
            $\mathbf{p}_{min} \leftarrow \mathbf{c}_i$
        Endif
    下一个子代
下一代

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

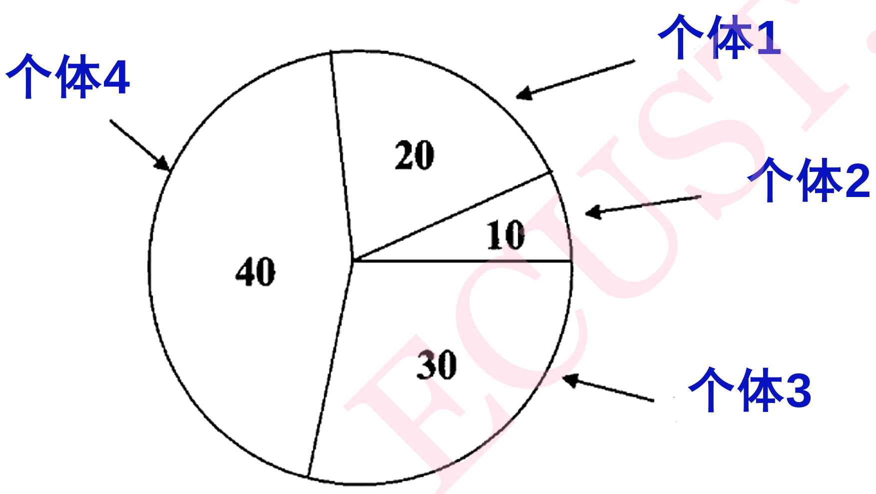
- 基本概念
  - 在生成子代前，需要进行选择父代的操作
  - 除轮盘赌选择外，还有很多选择方法，共同点都是：适应性较强的个体总是比适应性较弱的个体有更大的可能性被选中
  - 选择方法过分偏向适应性强的个体、不能足够地偏向适应性强的个体
  - 选择压力：量化不同选择方法之间的差别

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

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

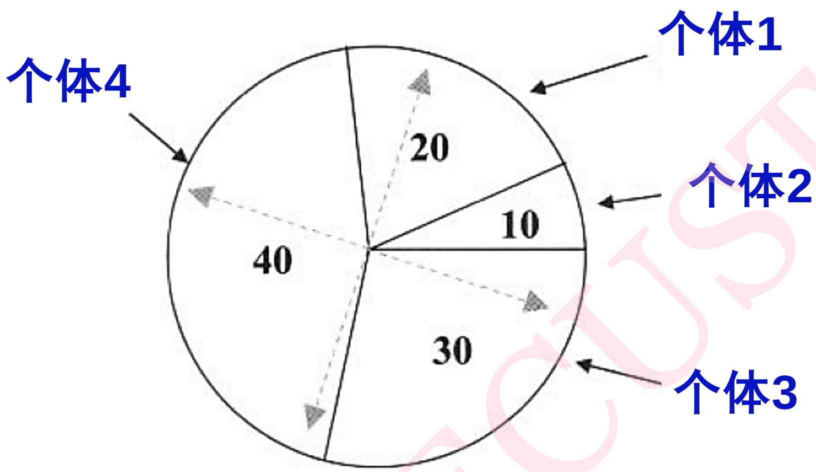


个体4在4次选择中都没有被选中的概率为：0.6^4=13%

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



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

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

- 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

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