

第五章 遗传算法

一. 前言

1. 进化算法的产生与发展

- 早在50年代，一些生物学家开始研究运用数字计算机模拟生物的自然遗传与自然进化过程
- **1963年**，德国柏林技术大学I. Rechenberg和H. P. Schwefel做风洞实验时，产生了**进化策略**的初步思想
- 60年代，美国学者L.J. Fogel在设计有限态自动机时提出进化规划的思想。**1966年**，Fogel等出版了《基于模拟进化的人工智能》，系统阐述了**进化规划**的思想

一. 前言

1. 进化算法的产生与发展

- **1975年**，美国学者Holland出版了著名的《Adaptation in Natural and Artificial Systems》，标志**遗传算法**的诞生
- **1992年**，Koza提出了**遗传规划**算法
- **1995年**，Price和Storn提出了**差分进化**算法



一. 前言

1. 进化算法的产生与发展

- 由于遗传算法、进化规划和进化策略是不同领域的研究人员分别独立提出的，在相当长的时期里相互之间没有正式沟通。直到90年代，才有所交流。他们发现彼此的基本思想具有惊人的相似之处，于是提出将这类方法统称为“进化算法” (**Evolutionary Algorithm, EA**)。后来，遗传规划和差分进化也被归纳到范畴之内。

一. 前言

1. 进化算法的产生与发展

- **1989年**, Holland的学生**D. J. Goldberg**出版了《**Genetic Algorithms in Search, Optimization, and Machine Learning**》, 对遗传算法及其应用作了全面而系统的论述
- **1991年**, **L. Davis**编辑出版了《**Handbook of Genetic Algorithm**》, 其中包括了遗传算法在工程技术和社会生活中大量的应用实例
- **2000年**, 汪定伟教授翻译出版了《**遗传算法与工程优化**》

一. 前言

2. 生物学原理

- 达尔文进化论——**Survival of the Fittest**
 - 所有环境资源都是有限的，只能维持有限的物种生存需要
 - 任何物种都有生存繁衍（ **production** ）本能
 - 自然选择是无法避免的
 - 能够最有效地争夺资源的物种无疑具有更大的生存机会

一. 前言

2. 生物学原理

- 达尔文进化论——**Diversity Drives Change**
 - 物种的表型性状（**phenotypic traits**）是指物种对外界环境反应的行为或者物理差异
 - 部分由继承决定，部分由发展因素决定
 - 对于每个个体来说都是独一无二的，部分是由随机变化产生的
 - 如果某种表型性状能够使得物种具有更高的生存机会，并且被后代继承，则这样的表型性状在后代中会增加
 - 新的性状组合出现...

一. 前言

2. 生物学原理

➤ 基本术语

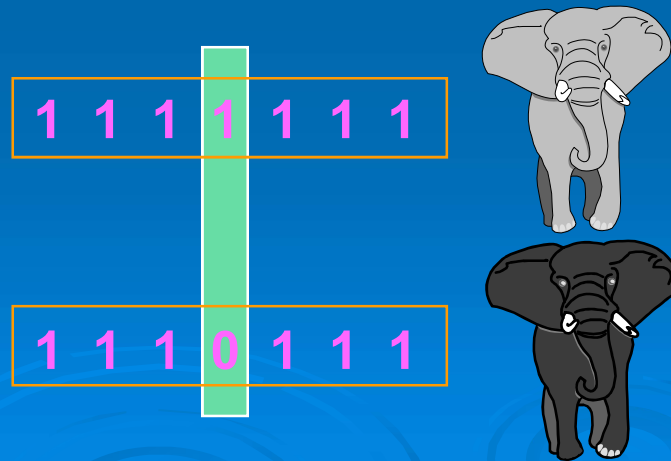
- 染色体 (**chromosome**) : 遗传物质的载体
- 遗传因子 (**gene**) : **DNA**长链结构中占有一定位置的基本遗传单位

一. 前言

2. 生物学原理

➤ 基本术语

- 基因型 (**genotype**) : 遗传因子组合的模型
- 表现型 (**phenotype**) : 由染色体决定性状的
外部表现



一. 前言

2. 生物学原理

➤ 基本术语

- 复制（**reproduction**）：细胞在分裂时，遗传物质**DNA**通过复制而转移到新产生的细胞中，新的细胞就继承了旧细胞的基因
- 交叉（**crossover**）：在两个染色体的某一相同位置处**DNA**被切断，其前后两串分别交叉组合形成两个新的染色体。又称基因重组，俗称“杂交”

一. 前言

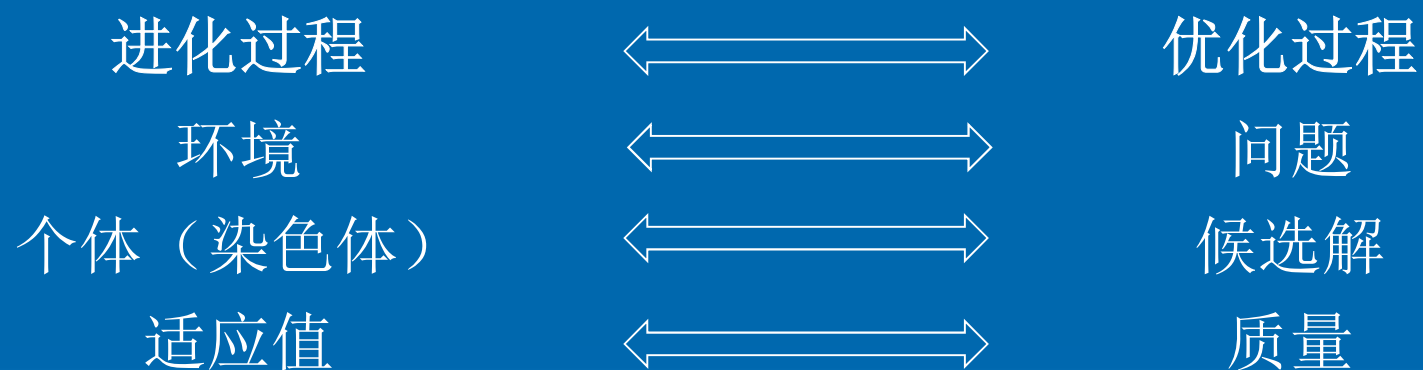
2. 生物学原理

➤ 基本术语

- 变异（**mutation**）：在细胞进行复制时可能以很小的概率产生某些复制差错，从而使DNA发生某种变异，产生出新的染色体，这些新的染色体表现出新的性状
- 适应度（**fitness**）：度量某个物种对于生存环境的适应程度。对生存环境适应程度较高的物种将获得更多的繁殖机会，而对生存环境适应程度较低的物种，其繁殖机会就会相对较少，甚至逐渐灭绝

一. 前言

3. 基本思想



一. 前言

3. 基本思想

- 遗传算法（**Genetic Algorithm, GA**）是一种基于种群的迭代的元启发式优化算法
 - “黑箱”工具
 - 利用种群进行寻优
 - 借鉴遗传进化思想来设计迭代机制

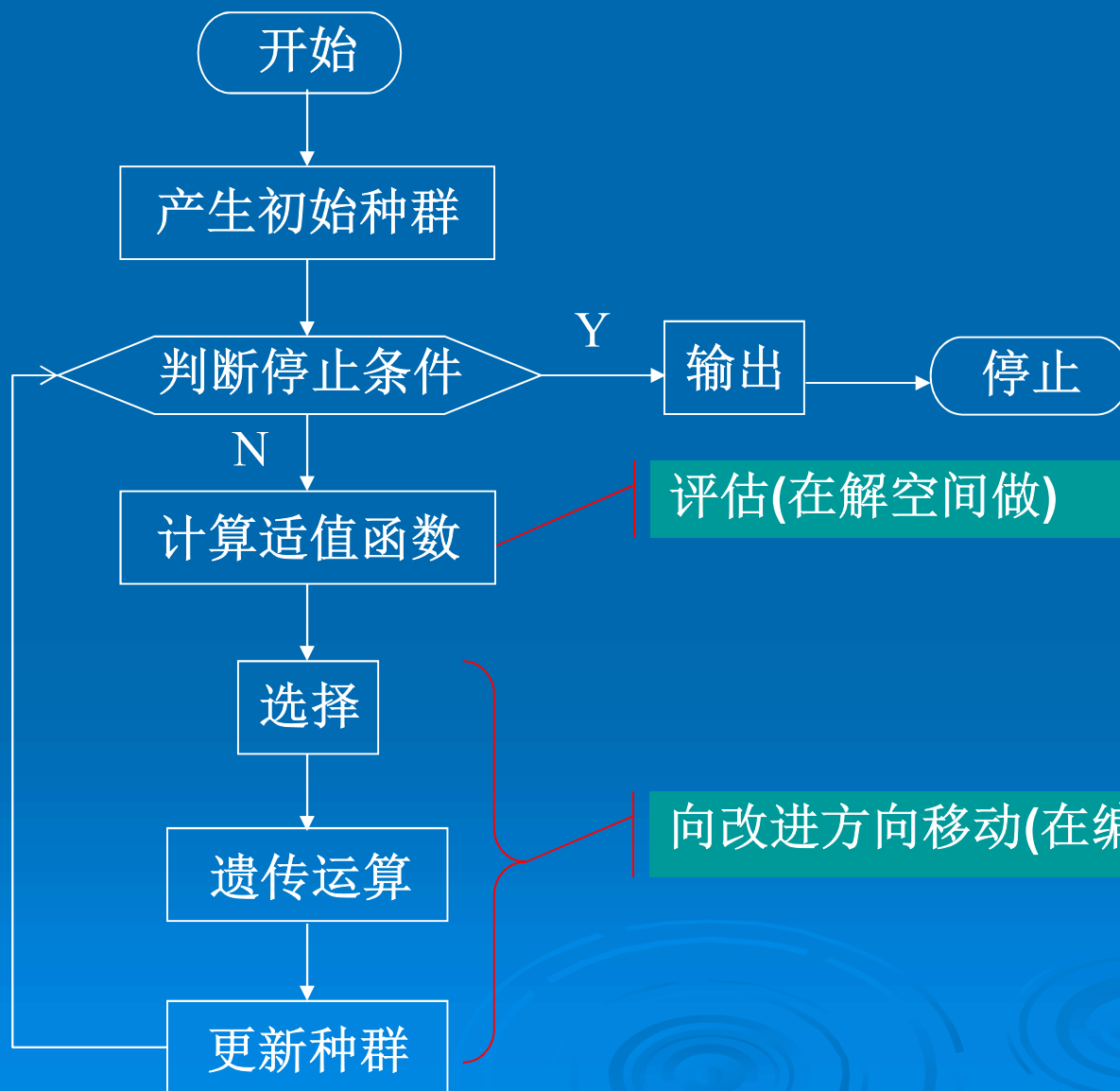


二. 基本算法

1. 基本流程

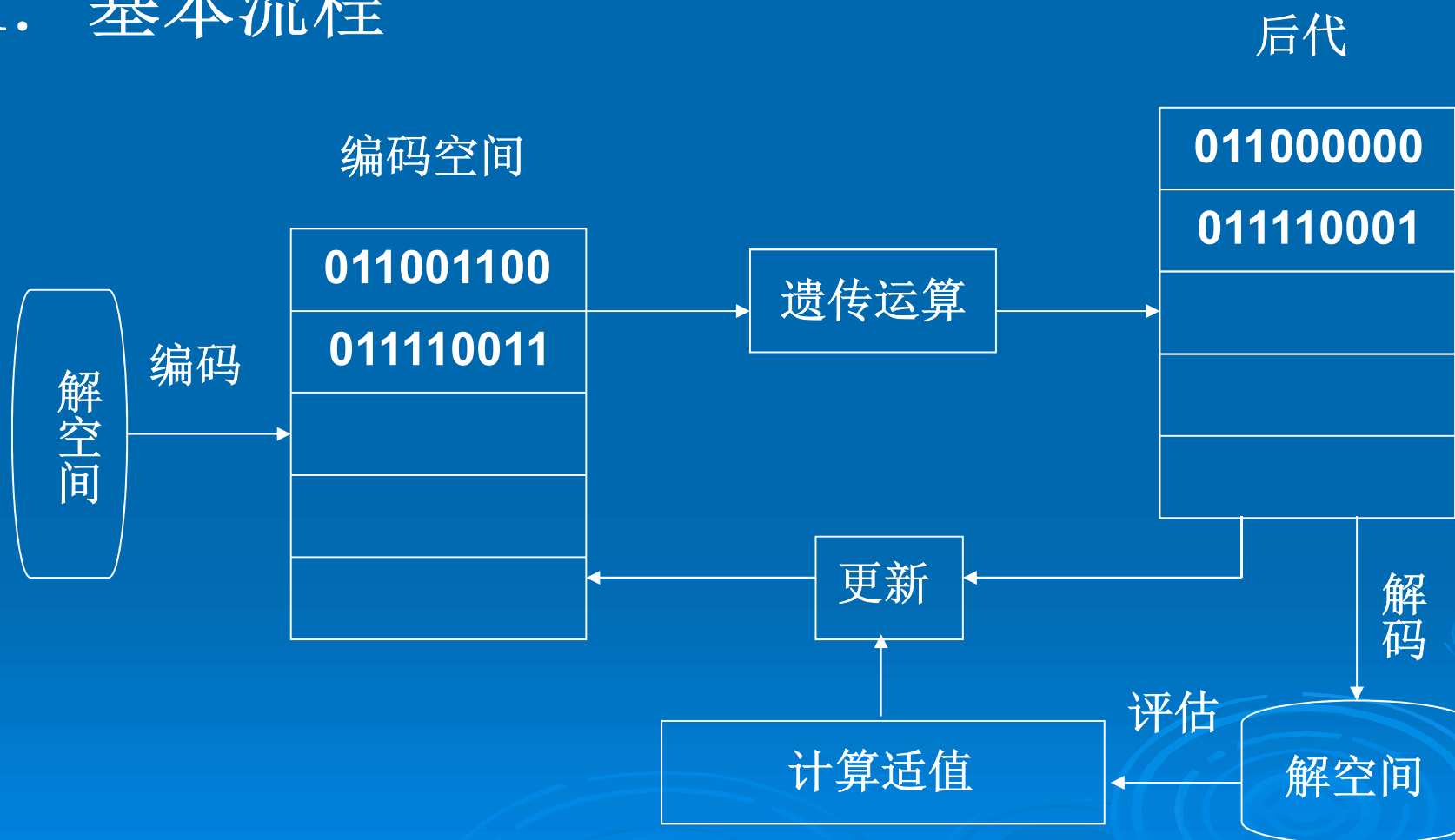
见下页

二. 基本算法



二. 基本算法

1. 基本流程



Holland提出的Simple GA

二. 基本算法

2. 构成要素

➤ 个体

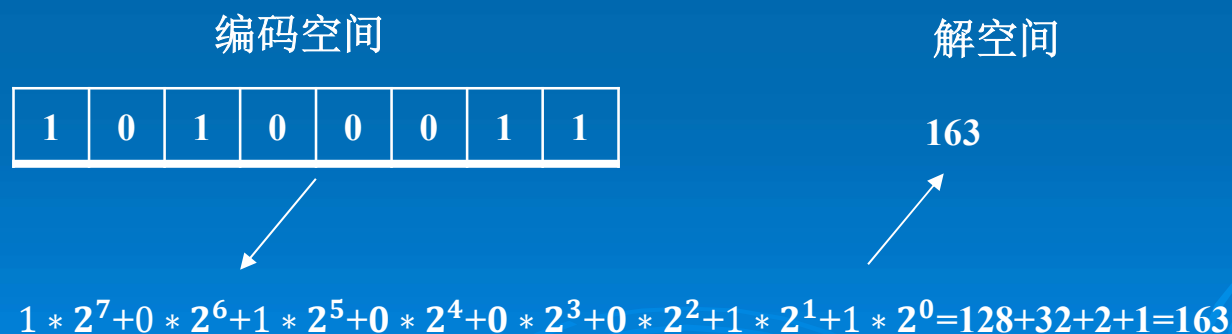
- 编码方法：二进制、整数编码、顺序编码、实数编码等，问题依赖的
- 适值函数：根据目标函数设计（适值函数的标定），可以直接将目标函数作为适值函数

二. 基本算法

1. 基本流程

➤ 解空间与编码空间的转换

- 适值计算需要在解空间内进行，而遗传运算是
对编码空间操作的，所以要进行两个空间的
转换



二. 基本算法

2. 构成要素

➤ 种群

- 种群大小：依赖于计算机的计算能力和计算复杂度，通常设为常数
- 初始种群的产生：依赖于编码方法，随机或者依据一定的启发式产生

二. 基本算法

2. 构成要素

➤ 选择

- 选择概率的计算方法：按比例的适应度函数、基于排序的适应度计算等
- 选择算法：轮盘赌选择、随机遍历抽样、截断选择、锦标赛选择等

二. 基本算法

2. 构成要素

➤ 交叉

- 交叉率：一对父代个体进行交叉的概率，用 p_c 表示，通常设为0.8-0.9
- 二进制及整数编码：单切点交叉、双切点交叉、均匀交叉
- 顺序编码：部分映射交叉、顺序交叉、循环交叉
- 实数编码：离散交叉、线性重组

二. 基本算法

2. 构成要素

➤ 变异

- 变异率：染色体上基因发生变异的概率，用 pm 表示，一般设定得比较小，在 **0.05** 以下
- 二进制及整数编码：位变异
- 顺序编码：插入变异，交换变异，颠倒变异
- 实数编码：值变异

二. 基本算法

3. 计算举例

➤ 产生初始种群

0001100000	0101111001	0000000101	1001110100	1010101010
(8)	(5)	(2)	(10)	(7)
1110010110	1001011011	1100000001	1001110100	0001010011
(12)	(5)	(19)	(10)	(14)

➤ 计算适值

二. 基本算法

3. 计算举例

➤ 选择

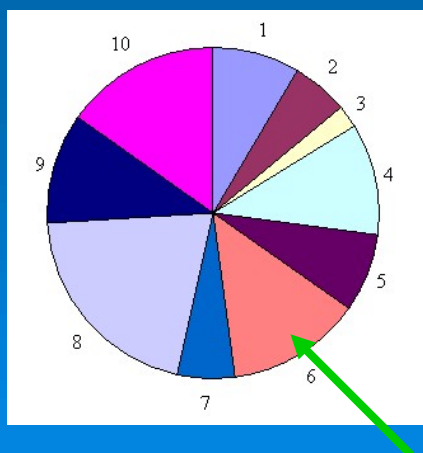
个体	染色体	适值	选择概率	累积概率
1	0001100000	8	0.086957	
2	0101111001	5	0.054348	
3	0000000100	2	0.021739	
4	1001010011	10	0.108696	
5	0100101010	7	0.076087	
6	1000100101	12	0.130435	
7	0001010101	5	0.054348	
8	1100000001	19	0.206522	
9	1001110100	10	0.108696	
10	0001010011	14	0.152174	

8+5+2+10+7+12+5+19+10+14

二. 基本算法

3. 计算举例

➤ 选择



个体	染色体	适值	选择概率	累积概率
1	0001100000	8	0.086957	0.086957
2	0101111001	5	0.054348	0.141304
3	0000000101	2	0.021739	0.163043
4	1001110100	10	0.108696	0.271739
5	1010101010	7	0.076087	0.347826
6	1110010110	12	0.130435	0.478261
7	1001011011	5	0.054348	0.532609
8	1100000001	19	0.206522	0.739130
9	1001110100	10	0.108696	0.847826
10	0001010011	14	0.152174	1.000000

二. 基本算法

3. 计算举例

➤ 选择

在0-1之间产生随机数:

0.070
0.545929
0.784567
0.446930
0.507893
0.291198
0.716340
0.270901
0.371435
0.854641

淘汰!

淘汰!

个体	染色体	适应度	选择概率	累积概率
1	0001100000	8	0.086957	0.086957
2	0101111001	5	0.054348	0.141304
3	0000000101	2	0.021739	0.163043
4	1001110100	10	0.108696	0.271739
5	1010101010	7	0.076087	0.347826
6	1110010110	12	0.130435	0.478261
7	1001011011	5	0.054348	0.532609
8	1100000001	19	0.206522	0.739130
9	1001110100	10	0.108696	0.847826
10	0001010011	14	0.152174	1.000000

二. 基本算法

3. 计算举例

➤ 交叉 ($pc = 0.9$)

在0-1之间产生随机数

0001100000	1110010110	1100000001	1001110100	1010101010
1110010110	1001011011	1001110100	1100000001	0001010011



0001010110	1111011011	1100000100	1001110100	1010101011
1110100000	1000010110	1001110001	1100000001	0001010010

0.755361

0.321546

0.568922

0.925140

0.154232

二. 基本算法

3. 计算举例

➤ 变异 ($pm = 0.05$)

在0-1之间产生随机数

0001010110	1111011011	1100000100	10011 <u>1</u> 0100	1010101011
1110100000	1000010110	1001 <u>1</u> 10001	1100000001	0 <u>0</u> 010100 <u>1</u> 0



0001010110	1111011011	1100000100	10011 <u>0</u> 0100	1010101011
1110100000	1000010110	1001 <u>0</u> 10001	1100000001	0 <u>1</u> 010100 <u>0</u> 0

二. 基本算法

3. 计算举例

- 至下一代，适应度计算→选择→交叉→变异，直至满足终止条件

三. 模板理论

1. 基本概念

- 模板：若干位确定，若干位不确定的一类个体的总称，用 S 表示，如 $0****$ 或 $1****$
- 模板的长度 $l(s)$ ：模板第一个确定位与最后一个确定位之间的长度
- 模板的阶数 $K(s)$ ：模板中确定位的个数

例如

若 $S : **0*1*1*$ ，则 $l(s)=4$ ， $K(s)=3$

三. 模板理论

1. 基本概念

➤ 常识

- n 位编码总长 $n-1$
- 阶数为 $K(s)$ 的模板 S 中的个体总数为 $2^{n-K(s)}$
- 对于一个 n 位二进制表达, 染色体长度为 n , 则模板数 $>$ 个体数($3^n > 2^n$), 即分类方法数 $>$ 个体总数, 因模板因子、个体因子分别为 $(0,1,*)$ 、 $(0,1)$ 。

三. 模板理论

2. 模板理论

- 引理1：在正比选择下，模板在第 $t+1$ 代的期望个体数为： $E(s, t + 1) = f(s, t) \cdot N(s, t)$ 。

其中， $f(s, t)$ 是第 t 代模板 s 中所有个体的适值均值与种群中所有个体的适值均值的比， $N(s, t)$ 是第 t 代 s 的个体数。

三. 模板理论

2. 模板理论

➤ 引理1

证明:

$$E(s, t + 1) = [P_1 + P_2 + \cdots + P_{NS}] * NP$$

$$= \frac{F_1 + F_2 + \cdots + F_{NS}}{Sum} * NP$$

$$= \frac{\frac{F_1 + F_2 + \cdots + F_{NS}}{N(s, t)} * N(s, t)}{\frac{Sum}{NP}} = \frac{\bar{f}_s}{\bar{f}} * N(s, t)$$

$$= f(s, t) * N(s, t)$$

三. 模板理论

2. 模板理论

➤ 引理1

注释

种群的适值和 $Sum = \sum_{i=1}^{NP} F_i$,

则选择概率 $P_i = \frac{F_i}{Sum} = \frac{F_i}{\sum_{i=1}^{NP} F_i}$

三. 模板理论

2. 模板理论

➤ 引理1

注释

\bar{f}_s 为 s 模板中所有个体的适值均值, \bar{f} 为种群的适值均值

只要均值 $f(s, t) > 1$, 则好模板的个体会越来越多

以上证明没有考虑交叉变异, 那么交叉变异会不会破坏种群模板 s ? 概率有多大?



三. 模板理论

2. 模板理论

- 引理2：第 t 代以概率 pc 做交叉，对长度为 $l(s)$ 的模板 s 中的个体，则在第 $t+1$ 代中该个体仍在模板 s 中的概率下界为：

$$P(s, t + 1) \geq 1 - \frac{pc \cdot l(s)}{n - 1} [1 - P(s, t)]$$

其中， $P(s, t)$ 为第 t 代个体属于模板 s 的概率。

三. 模板理论

2. 模板理论

➤ 引理2

证明:

交叉破坏 s 的条件

做了交叉: pc

交叉点在 s 内: $l(s)/(n-1)$

配偶不在 s 中: $1 - P(s, t)$

则不被破坏的概率为:

$$P(s, t + 1) \geq 1 - pc \cdot \frac{l(s)}{n-1} \cdot [1 - P(s, t)]$$

若配偶不属于模板 s , 是否能产生后代为模板 s ?



三. 模板理论

2. 模板理论

- 引理3: 若第 t 代以 pm 做变异, 对于一个阶数为 $K(s)$ 的模板 s 中的个体, 则在第 $t+1$ 代仍在 s 中的概率下界为: $P(s, t + 1) \geq 1 - pm \cdot K(s)$

三. 模板理论

2. 模板理论

➤ 引理3

证明:

对于 S 来说

当 $K(s) = 1$ 时, 不被破坏的概率为: $1 - pm$

当 $K(s) > 1$ 时, 不被破坏的概率为: $(1 - pm)^{K(s)}$

取其泰勒展开式的第一项: $1 - pm \cdot K(s)$

三. 模板理论

2. 模板理论

- 主定理（模板定理）：第 t 代以概率 pc 和 pm 做交叉和变异时，长度为 $l(s)$ ，阶数为 $K(s)$ ，适值均值为 $f(s,t)$ 的模板 s 在第 $t+1$ 代的期望个体数的下界为：

$$\begin{aligned} E(s, t + 1) \\ \geq \left\{ 1 - \frac{pc \cdot l(s)}{n - 1} (1 - P(s, t)) - pm \cdot K(s) \right\} \cdot f(s, t) \\ \cdot N(s, t) \end{aligned}$$

三. 模板理论

2. 模板理论

- 主定理（模板定理）：第 t 代以概率 pc 和 pm 做交叉和变异时，长度为 $l(s)$ ，阶数为 $K(s)$ ，适值均值为 $f(s,t)$ 的模板 s 在第 $t+1$ 代的期望个体数的下界为：

$$\begin{aligned} E(s, t+1) \\ \geq \left\{ 1 - \frac{pc \cdot l(s)}{n-1} (1 - P(s, t)) - pm \cdot K(s) \right\} \cdot f(s, t) \\ \cdot N(s, t) \end{aligned}$$

当 $f(s, t) > [1 - \frac{pc \cdot l(s)}{n-1} (1 - P(s, t)) - pm \cdot K(s)]^{-1}$ 时，

$E(s, t)$ 随代数增加而增加

四. 算法设计

1. 编码方法

➤ 编码原则

- 完备性：所有解都能表示为染色体
- 健全性：染色体上任何基因取值都尽可能存在
- 非冗余性：解空间和编码空间尽可能一一对应

四. 算法设计

1. 编码方法

➤ 编码方式

- 二进制编码：适用于背包问题、实优化问题等
- 整数编码：适用于时间优化，伙伴挑选等
- 顺序编码：适用于指派、旅行商问题，单机调度等，合法性问题
- 浮点数编码：适用于实优化等，简单运算方便

编码方法示例（1）

背包问题：有 n 个物品，对物品 i ，价值为 p_i ，体积为 w_i ，背包容量限制为 W ，如何选取物品装入背包，使背包中的价值最大？

二进制编码策略

物品编号：	1	2	3	4	5	6	7
染色体：	0	1	1	0	0	1	1
解：	不装入	装入	装入	不装入	不装入	装入	装入

思考：

是否可以满足完备性原则？

是否满足健全性原则？

非冗余性原则满足程度如何？

编码方法示例（2）

实优化问题：

$$\begin{aligned} \max x^2 \\ \text{s.t. } 0 \leq x \leq 31 \end{aligned}$$

二进制编码策略

染色体：	0	1	1	0	1	0	1	1	1	0
解：	$0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4} + 0 \times 2^{-5} = 13.4375$									

思考：

是否可以满足完备性原则？

是否满足健全性原则？

非冗余性原则满足程度如何？

编码方法示例（3）

整数优化问题：

$$\min \sum_{i=1}^6 x_i^2$$

s. t. $x_i \in \{0, 1, \dots, 10\}, i = 1, 2, \dots, 6$

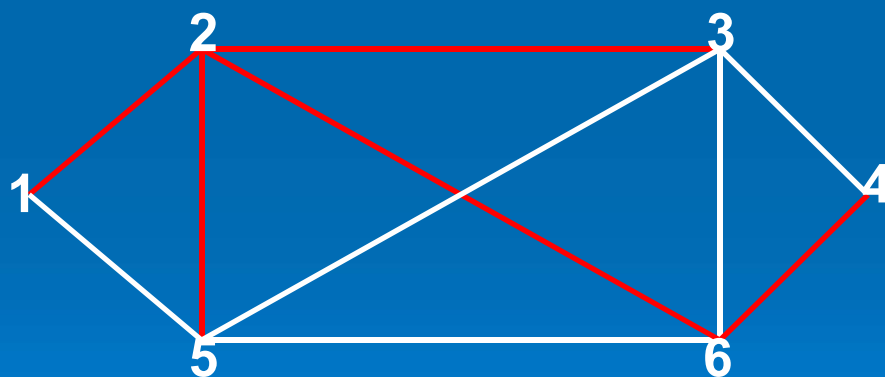
整数编码策略

编号 <i>i</i> :	1	2	3	4	5	6
染色体:	2	5	3	1	0	7
解 x_i :	2	5	3	1	0	7

编码方法示例（4）

生成树问题：在 n 个节点的无向图上找到满足如下条件的连通子图，即为一颗 n 个节点的生成树

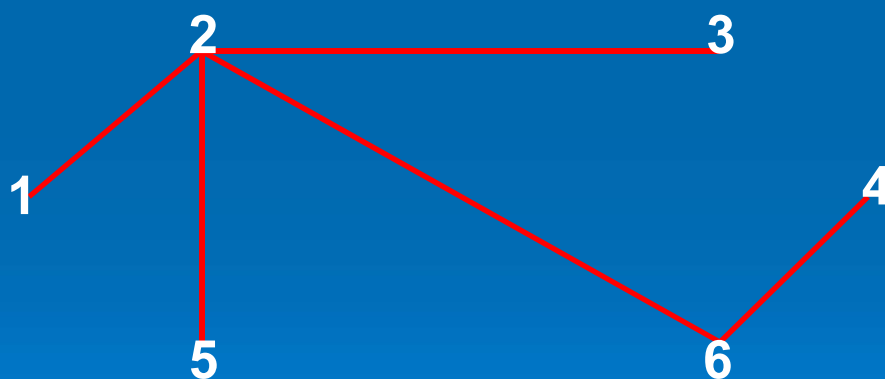
- 覆盖所有节点
- 连通的
- 没有回路



思考：如何对图中所示的生成树问题进行编码？

编码方法示例（4）

一种特殊的**Prüfer**数编码：用 **$n-2$** 位自然数唯一的表达出一棵 **n** 个节点的生成树，其中每个数字在**1**和 **n** 之间



叶子：树中度数为1的节点
图中节点1、3、4、5

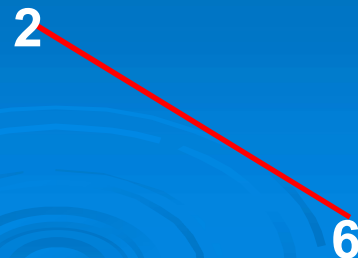
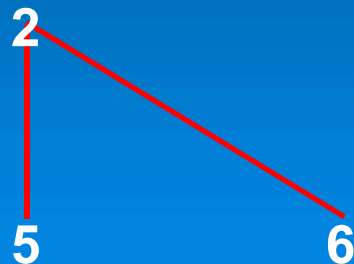
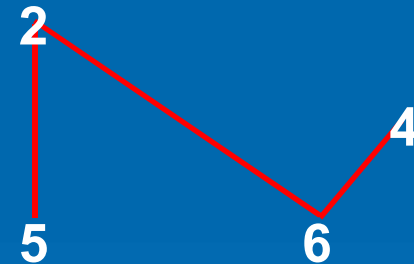
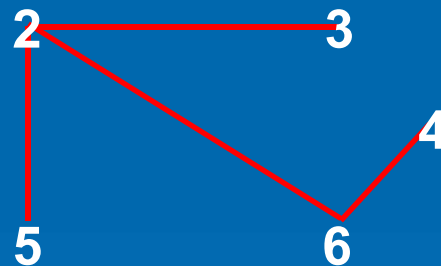
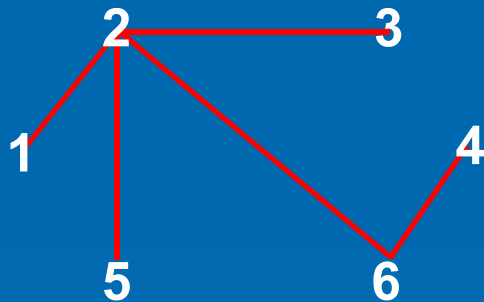
编码方法示例（4）

如何将一个生成树编码为一个**Prüfer**数，具体步骤如下：

- a. 设节点*i*是标号最小的叶子；
- b. 若*i*与*j*相连，令*j*是编码中的第一个数字；
- c. 删去边(*i*, *j*)；
- d. 转a，直到剩下一条边为止。

图解: $i=1, (i, j)=(1,2), j=2;$
 $i=3, (i, j)=(3,2), j=2;$
 $i=4, (i, j)=(4,6), j=6;$
 $i=5, (i, j)=(5,2), j=2。$

编码: 2 2 6 2

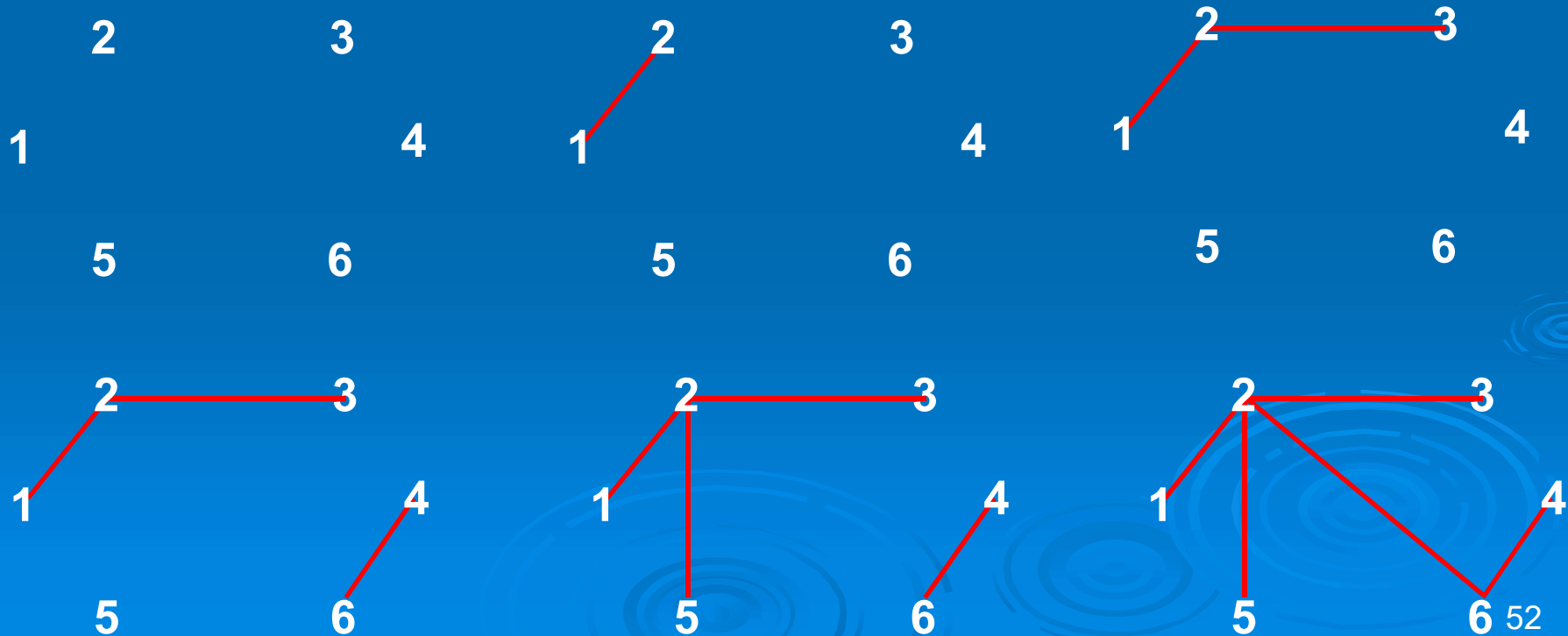


编码方法示例（4）

如何将一个**Prüfer**数解码为一个生成树，具体步骤如下：

- a. 令**Prüfer**数中的节点集为 P ，不包含在 P 中的节点集为 P' ；
- b. 若 i 为 P' 中最小标号的节点， j 为 P 上最左边数字连接边 (i, j) ，并从 P' 中去掉 i ，从 P 中去掉 j ，若 j 不再在 P 中，将 j 加入 P' 中；
- c. 重复b，直到 P 中没有节点（即为空）， P' 中剩下 (s, r) ；
- d. 连接 (s, r) 。

图解: $P = \{ 2, 2, 6, 2 \}$ $P' = \{ 1, 3, 4, 5 \}$ (1, 2)
 $P = \{ 2, 6, 2 \}$ $P' = \{ 3, 4, 5 \}$ (3, 2)
 $P = \{ 6, 2 \}$ $P' = \{ 4, 5 \}$ (4, 6)
 $P = \{ 2 \}$ $P' = \{ 5, 6 \}$ (5, 2)
 $P = \{ \Phi \}$ $P' = \{ 6, 2 \}$ (6, 2)



编码方法示例（4）

最小生成树

- 在 n 个节点的无向图中找到弧权重和最小的生成树
- 对于 n 个节点的无向图来说，生成树的个数是 n^{n-2} ，而**Prüfer**数的个数也为 n^{n-2}
- 利用遗传算法解决最小生成树问题时，采用**Prüfer**数编码方法能够实现解空间和编码空间的一一对应，同时交叉变异不破坏合法性

一个好的编码方法对遗传算法至关重要

四. 算法设计

2. 适值函数的标定

- 适应度函数的重要性：适应度函数的选取直接影响遗传算法的收敛速度以及能否找到最优解；一般而言，适应度函数是由目标函数变换而成的，对目标函数值域的某种映射变换称为适应度的标定

四. 算法设计

2. 适值函数的标定

➤ 标定的目的

$$\bar{f}_1 = f_1 = 1001$$

$$\bar{f}_2 = f_2 = 1002$$

$$\bar{f}_3 = f_3 = 999$$

$$\bar{f}_4 = f_4 = 997$$

标定

$$\bar{f}_1 = f_1 - f_4 = 4$$

$$\bar{f}_2 = f_2 - f_4 = 5$$

$$\bar{f}_3 = f_3 - f_4 = 2$$

$$\bar{f}_4 = f_4 - f_4 = 0$$

选择压力小，差别小，选优功能弱化了

选择压力大，差别放大，选优功能强化了

四. 算法设计

2. 适值函数的标定

➤ 标定方法

- 线性标定

函数表达式: $F = af + b$

F 为适值函数, f 为目标函数

a. 对 $\max f(x)$, 令 $a = 1$, $b = -f_{\min} + \xi$

函数表达式: $F = f - f_{\min} + \xi$

b. 对 $\min f(x)$, 令 $a = -1$, $b = f_{\max} + \xi$

函数表达式: $F = -f + f_{\max} + \xi$

四. 算法设计

2. 适值函数的标定

➤ 标定方法

- 动态线性标定（最常用）

函数表达式: $F = a^k f + b^k$, k 为迭代指标

优点: 计算容易不占用时间

a. 对 $\max f(x)$, 令 $a^k = 1$, $b^k = -f_{min}^k + \xi^k$

函数表达式: $F = f - f_{min}^k + \xi^k$

b. 对 $\min f(x)$, 令 $a^k = 1$, $b^k = f_{max}^k + \xi^k$

函数表达式: $F = -f + f_{max}^k + \xi^k$

四. 算法设计

2. 适值函数的标定

➤ 标定方法

- 动态线性标定（最常用）

ξ^k 的作用： ξ^k 的加入使最坏个体仍有繁殖的可能，它的值随 k 的增大而减小

ξ^k 的取值：

$$\xi^0 = M, \quad \xi^k = \xi^{k-1} \cdot r, \quad r \in [0.9, 0.999]$$

调节 M 和 r ，从而来调节 ξ^k

四. 算法设计

2. 适值函数的标定

➤ 标定方法

- 幂率标定

函数表达式: $F = f^a$

a 的取值, $a > 1$ 时加大差别

$a < 1$ 时减小差别

- 对数标定

函数表达式: $F = a \cdot \ln f + b$

对数标定的作用: 缩小差别

四. 算法设计

2. 适值函数的标定

➤ 标定方法

- 指数标定

函数表达式: $F = a \cdot e^{bf} + c$

指数标定的作用: 扩大差别

- 窗口技术

函数表达式: $F = af - f_w$

f_w 为前 W 代中的最小目标值, 它考虑了各代 f_{min} 的波动, 这样 f_w 具有记忆性

四. 算法设计

2. 适值函数的标定

➤ 标定方法

- 正规化技术

函数表达式: $F = \frac{f - f_{\min} + r}{f_{\max} - f_{\min} + r}$

正规化技术的作用: 将 F 映射到 $(0,1)$ 区间, 抑制超级染色体

实质: 特殊的动态标定 $F = a^k f + b^k$

$$a^k = \frac{1}{f_{\max} - f_{\min} + r}, \quad b^k = \frac{-f_{\min} + r}{f_{\max} - f_{\min} + r}$$

适值函数标定应用示例

0-1背包问题

- 求解难点：如何处理约束来保持解的可行性
 - 拒绝策略：可行解不易达到时，很难达到一个初始种群
 - 修复策略：将不可行解修复为可行的，但将失去多样性
 - 惩罚策略：设计惩罚函数进行适值标定，但设计不好会掩盖目标函数的优化

适值函数标定应用示例

0-1背包问题

➤ 惩罚函数法

将适值函数标定为 $F(x) = f(x)P(x)$ ，其中 $f(x)$ 是目标函数

$$\text{令 } P(x) = \frac{|\sum_{i=1}^n w_i x_i - W|}{\delta},$$

$$\text{其中 } \delta = \max\{W, |\sum_{i=1}^n w_i - W|\}$$

注： W 与 $|\sum_{i=1}^n w_i - W|$ 是 $|\sum_{i=1}^n w_i x_i - W|$ 的两个端点

↓ ↓

$$x = [0, 0, \dots, 0]^T \quad x = [1, 1, \dots, 1]^T$$

适值函数标定应用示例

0-1背包问题

➤ 惩罚函数法

- 函数式的意义

a. δ 的作用是使 $0 \leq |\sum_{i=1}^n w_i x_i - W| \leq \delta$, 保证 $0 \leq P(x) \leq 1$

b. $P(x)$ 可行也罚, 只有当 $|\sum_{i=1}^n w_i x_i - W| = 0$ 时不罚

c. 罚函数法目的是把解拉向边界, 尽量装满

四. 算法设计

3. 遗传操作——选择

➤ 基本概念

- 选择压力：最佳个体选中的概率与平均个体选中概率的比值
- 多样性损失：在选择阶段未选中个体数目占种群的比例

四. 算法设计

3. 遗传操作——选择

➤ 个体选择概率的计算方法

- 按比例的比例度分配

某个体 i ，其适应度为 f_i ，则其被选取的概率 P_i 为：

$$P_i = \frac{f_i}{\sum_{i=1}^{NP} f_i}$$

四. 算法设计

3. 遗传操作——选择

➤ 个体选择概率的计算方法

- 基于排序的适应度分配步骤:

a. 从好到坏排序所有个体

b. 定义最好个体的选择概率为 q , 则个体 i 的选择概率为: $P_i = q(1 - q)^{i-1}$

c. 由于

$$\sum_{i=1}^{NP} q(1 - q)^{i-1} \xrightarrow{NP \rightarrow \infty} q \frac{1}{1 - (1 - q)} = 1$$

四. 算法设计

3. 遗传操作——选择

- 个体选择概率的计算方法
 - 基于排序的适应度分配
- 步骤:

c. NP 有限时要归一化, 则有下面的两个公式:

$$\bar{q} = \frac{q}{1 - (1 - q)^{NP}}$$

$$p_j = \bar{q}(1 - q)^{j-1}$$

个体选择概率示例

染色体	适应值	排序	按比例计算 P_i	按排序计算 $P_i(q=0.5)$
A	1	3	0.10	0.14
B	5	1	0.50	0.57
C	4	2	0.40	0.29
Sum	10		1.0	1.0

染色体	适应值	排序	按比例计算 P_i	按排序计算 $P_i(q=0.5)$
A	1	3	0.05	0.14
B	15	1	0.75	0.57
C	4	2	0.20	0.29
Sum	20		1.0	1.0

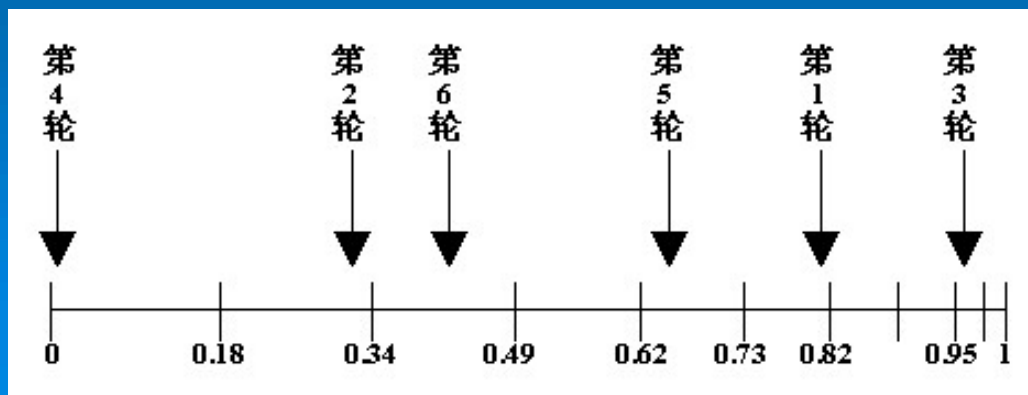
四. 算法设计

3. 遗传操作——选择

➤ 常用选择方法

- 轮盘赌选择法

个体	1	2	3	4	5	6	7	8	9	10
适应度	2.0	1.8	1.6	1.4	1.2	1.0	0.8	0.6	0.4	0.2
选择概率	0.18	0.16	0.15	0.13	0.11	0.09	0.07	0.06	0.03	0.02
累计概率	0.18	0.34	0.49	0.62	0.73	0.82	0.89	0.95	0.98	1.00



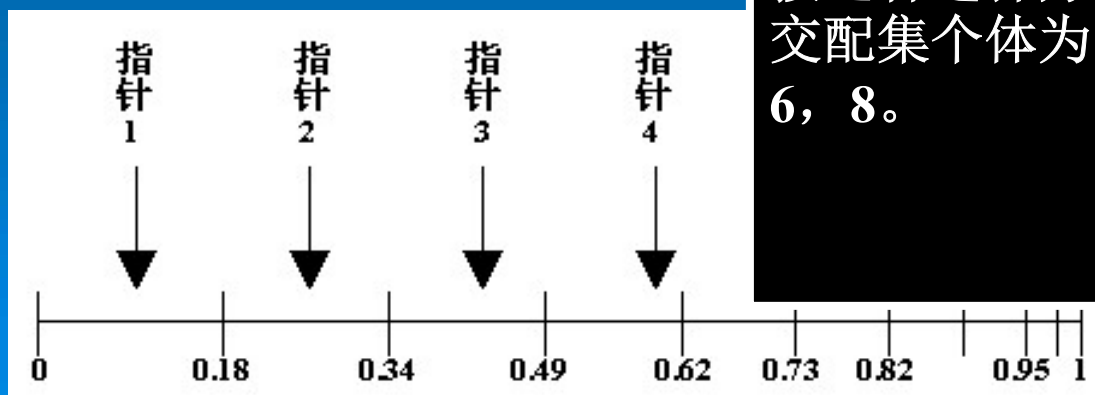
四. 算法设计

3. 遗传操作——选择

➤ 常用选择方法

- 随机遍历抽样法

个体	1	2	3	4	5
适应度	2.0	1.8	1.6	1.4	1.2
选择概率	0.18	0.16	0.15	0.13	0.11
累计概率	0.18	0.34	0.49	0.62	0.73



设定 n 为需要选择的个体数目，等距离选择个体，选择指针的距离为 $1/n$ 。第一个指针的位置由 $[0, 1/n]$ 区间的均匀随机数决定。如图所示，需要选择6个个体，指针间的距离为 $1/6=0.167$ ，第一个指针的随机位置为0.1，按这种选择方法被选中作为交配集个体为：1，2，3，4，6，8。

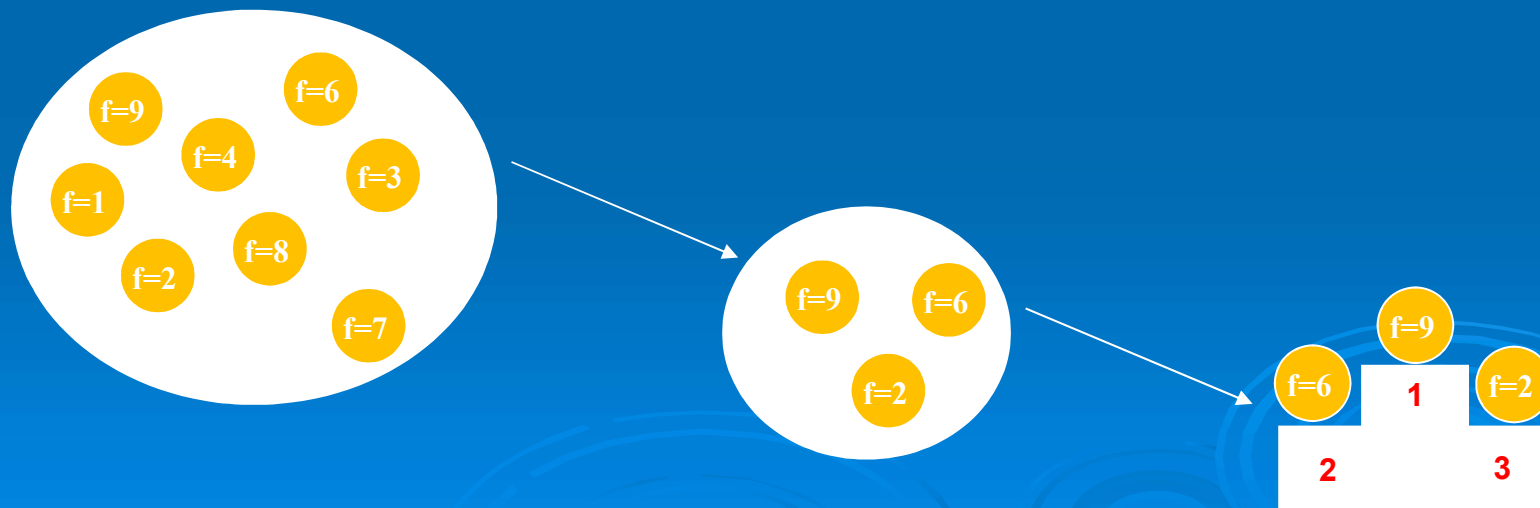
四. 算法设计

3. 遗传操作——选择

➤ 常用选择方法

- 锦标赛选择法

随机从种群中挑选一定数目个体，其中最好的个体作为父个体，此过程重复进行完成个体的选择



四. 算法设计

3. 遗传操作——交叉

➤ 二进制交叉

- 单切点交叉

parents

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

children

1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1	1	1	1	1	1

四. 算法设计

3. 遗传操作——交叉

➤ 二进制交叉

- 多切点交叉

parents

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

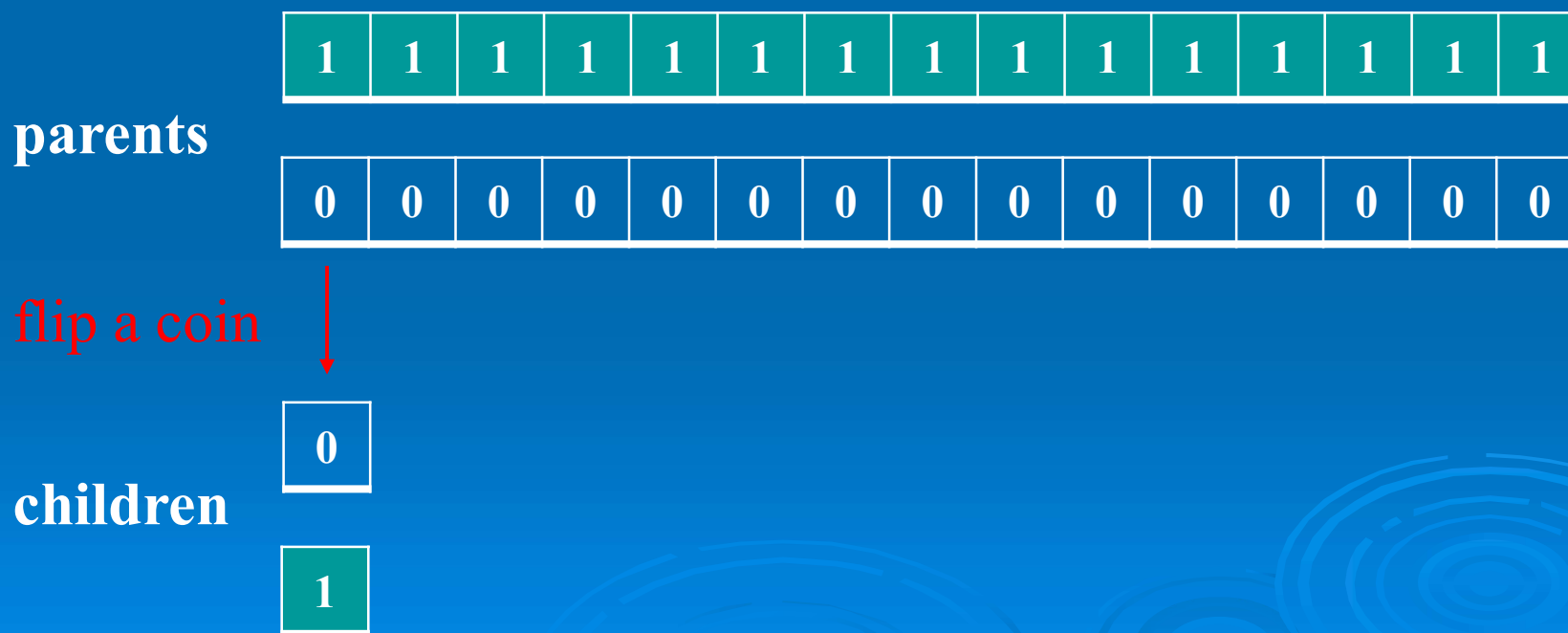
children

1	1	1	1	1	0	0	0	1	1	1	1	0	0	0
0	0	0	0	0	1	1	1	0	0	0	0	1	1	1

四. 算法设计

3. 遗传操作——交叉

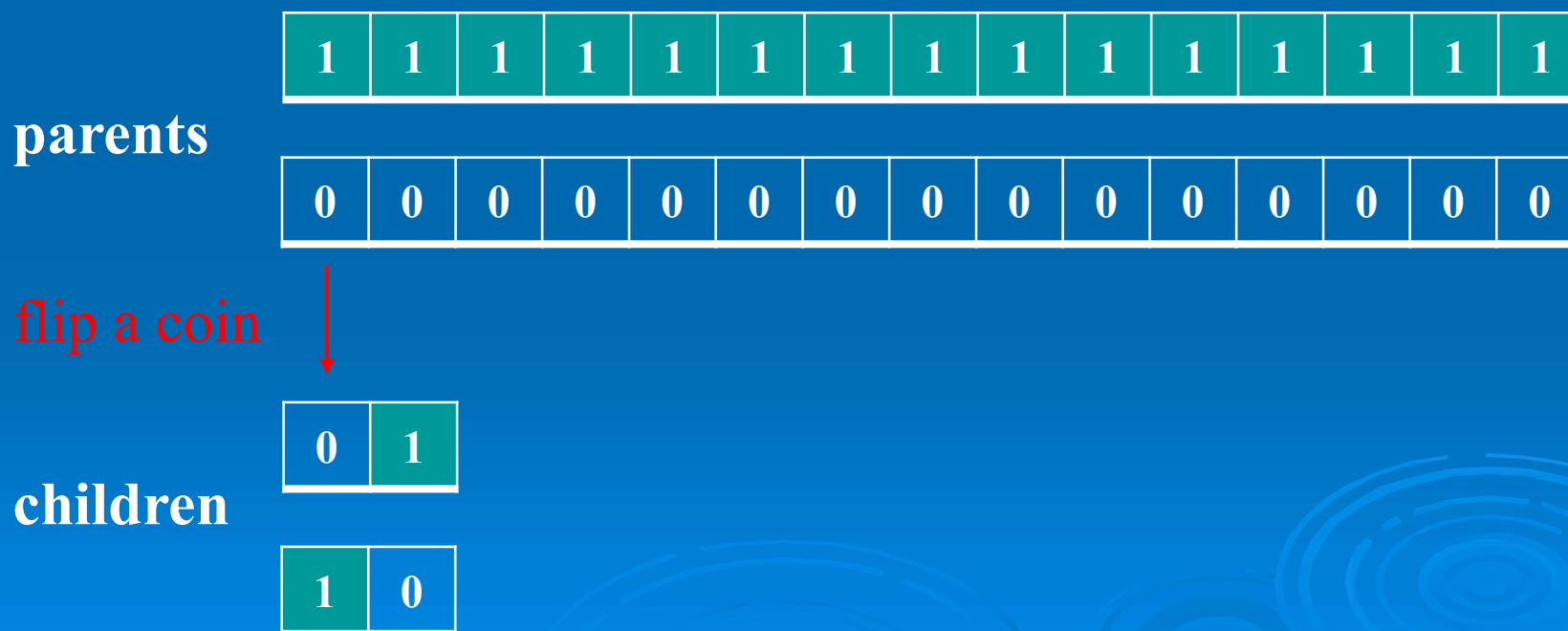
- 二进制交叉
 - 均匀交叉



四. 算法设计

3. 遗传操作——交叉

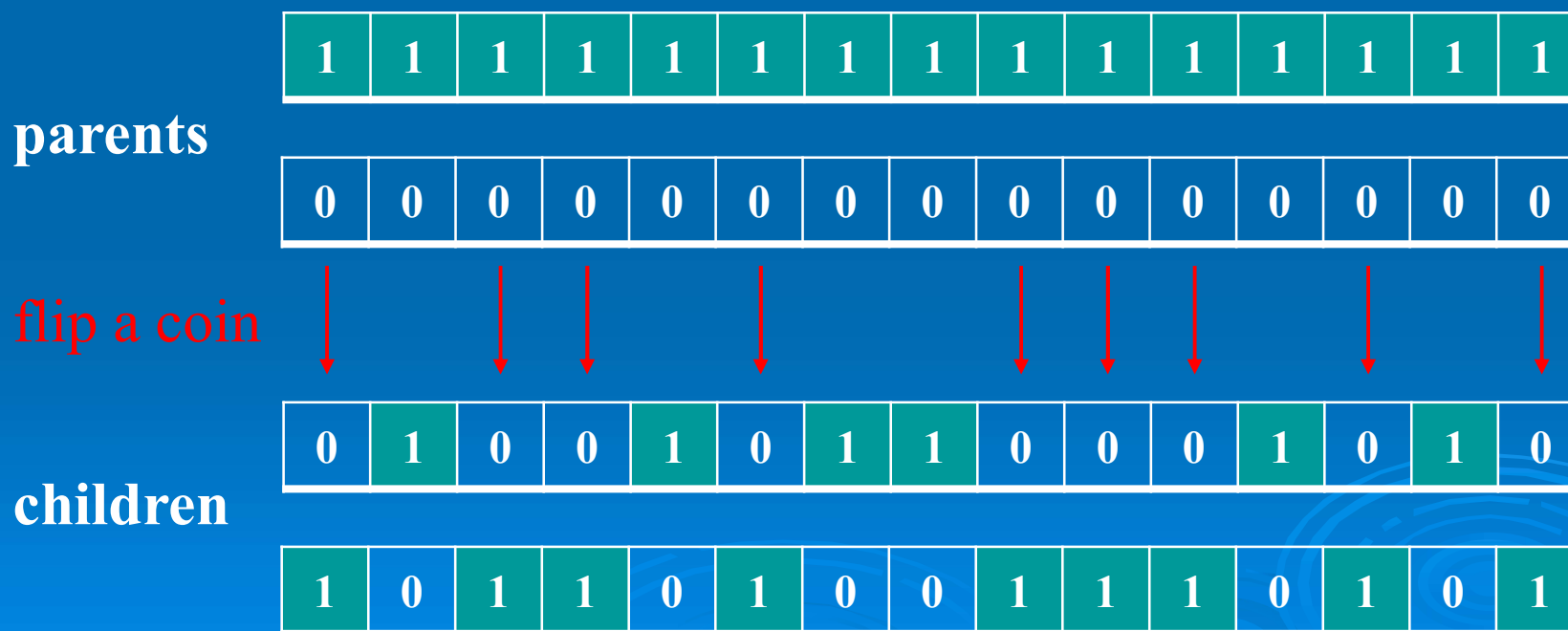
- 二进制交叉
 - 均匀交叉



四. 算法设计

3. 遗传操作——交叉

- 二进制交叉
 - 均匀交叉



四. 算法设计

3. 遗传操作——交叉

➤ 顺序编码交叉

在顺序编码遗传运算的过程中会遇见不合法的编码，应对策略有两种：拒绝或修复

例如：经交叉后，后代编码不合法

P1: 21 | 345 | 67 C1: 21 | 125 | 67

P2: 43 | 125 | 76 C2: 43 | 345 | 76

四. 算法设计

3. 遗传操作——交叉

➤ 顺序编码交叉

- 部分映射交叉(PMX)

步骤:

- a. 选切点X,Y;
- b. 交换中间部分;
- c. 确定映射关系;
- d. 将未换部分按映射关系恢复合法性。

四. 算法设计

3. 遗传操作——交叉

➤ 顺序编码交叉

- 部分映射交叉(PMX)

P1: 2 1 3 4 5 6 7	— — 1 2 5 — —
P2: 4 3 1 2 5 7 6	— — 3 4 5 — —

映射关系: 3-1, 4-2, 5-5

C1: 4 3 1 2 5 6 7
C2: 2 1 3 4 5 7 6

四. 算法设计

3. 遗传操作——交叉

➤ 顺序编码交叉

- 顺序交叉(OX)

步骤:

a. 选切点X,Y;

b. 交换中间部分;

c. 从切点Y后第一个基因起列出原序, 去掉已有基因;

d. Y后第一个位置起填入。

四. 算法设计

3. 遗传操作——交叉

➤ 顺序编码交叉

• 顺序交叉(OX)

P1: 2 1 3 4 5 6 7	— — 1 2 5 — —
P2: 4 3 1 2 5 7 6	— — 3 4 5 — —

列出基因: 6 7 ~~2~~ ~~1~~ 3 4 ~~5~~ 7 6 ~~4~~ ~~3~~ 1 2 ~~5~~

C1: 3 4 1 2 5 6 7
C2: 1 2 3 4 5 7 6

四. 算法设计

3. 遗传操作——交叉

➤ 顺序编码交叉

- 循环交叉(CX)

步骤:

- a. 选P1的第一个元素作为C1的第一位，选P2的第一个元素作为 C2的第一位；
- b. 到P1中找P2的第一个元素赋给C1的相对位置，重复此过程，直到P2上得到P1的第一个元素为止，称为一个循环；

四. 算法设计

3. 遗传操作——交叉

➤ 顺序编码交叉

- 循环交叉(CX)

步骤:

c. 对最前的基因按P1、P2基因轮替原则重复以上过程;

d. 重复以上过程, 直到所有位都完成。

循环交叉示例

P1	2	4	5	3	8	9	6	1	7
----	---	---	---	---	---	---	---	---	---

P2	3	9	8	6	5	4	2	7	1
----	---	---	---	---	---	---	---	---	---

循环1

P1	2	4	5	3	8	9	6	1	7
----	---	---	---	---	---	---	---	---	---

P2	3	9	8	6	5	4	2	7	1
----	---	---	---	---	---	---	---	---	---

循环2

P1	2	4	5	3	8	9	6	1	7
----	---	---	---	---	---	---	---	---	---

P2	3	9	8	6	5	4	2	7	1
----	---	---	---	---	---	---	---	---	---

循环3

P1	2	4	5	3	8	9	6	1	7
----	---	---	---	---	---	---	---	---	---

P2	3	9	8	6	5	4	2	7	1
----	---	---	---	---	---	---	---	---	---

循环4

循环交叉示例



交叉
规则

奇数循环: C1保留P1信息, C2保留P2信息

偶数循环: C1保留P2信息, C2保留P1信息



四. 算法设计

3. 遗传操作——交叉

➤ 顺序编码交叉

- 循环交叉(CX)

与OX的特点不同的是，CX较好的保留了位值特征，适合指派问题；而OX较好的保留了相邻关系、先后关系满足了TSP问题的需要。

四. 算法设计

3. 遗传操作——交叉

➤ 实数编码交叉

- 离散交叉: $z_i = x_i \text{ or } y_i$

采用多切点交叉或者均匀交叉方式

- 线性交叉: $z_i = a \cdot x_i + (1 - a) \cdot y_i$

其中, $0 \leq a \leq 1$

采用单点重组（随机选择一个基因位）或者完全重组（所有基因位均参与）

四. 算法设计

3. 遗传操作——变异

➤ 二进制编码变异：位变异

变异前

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

flip a coin



变异后

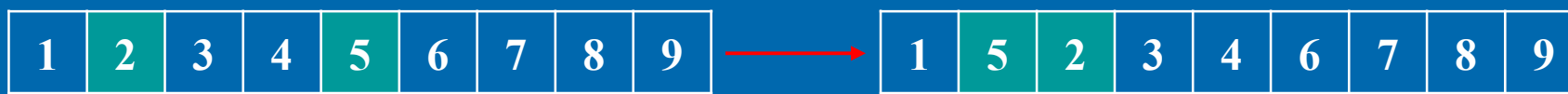
1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

四. 算法设计

3. 遗传操作——变异

➤ 顺序编码变异

- 插入变异：随机选择两个基因，将后面的基因移到前面的基因之前，其余基因依次顺延



- 交换变异：随机选择两个基因，交换它们的位置

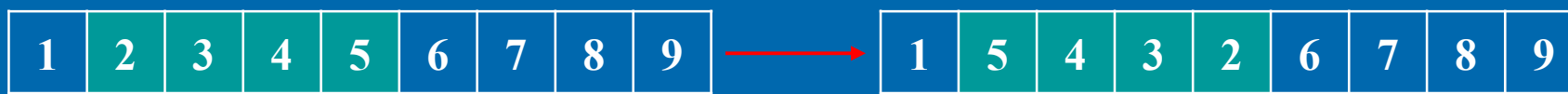


四. 算法设计

3. 遗传操作——变异

➤ 顺序编码变异

- 翻转变异：随机选择两个基因，将它们中间子串翻转，其余基因保持不变



- 随机变异：随机选择若干个基因，将它们随机地重新安排在这些位置



四. 算法设计

3. 遗传操作——变异

➤ 实数编码变异

- 随机变异：任选一位用一个随机数取代
- 位值变异：任选一位加 Δ ， Δ 服从均匀分布、指数分布或者正态分布
- 梯度方向变异

四. 算法设计

3. 遗传操作——更新种群

➤ 更新策略

- 基于Age的更新策略：保留最新的个体
- 基于适值的更新策略：采用上述介绍的选择策略来保留个体
- 精英保留策略：至少保留一个最优父代个体
- **GENITOR**：合并所有父代和子代个体，保留最好的