

# 第四章 遗传算法

## Genetic Algorithms

邢文训

清华大学数学科学系

62787945

Email: [wxing@tsinghua.edu.cn](mailto:wxing@tsinghua.edu.cn)

答疑时间：周四下午4：00-5：00

# 主要内容

- 遗传算法的基本框架及一些基本性质
- 理论结果之一——模板定理
- 理论结果之二——马尔可夫收敛性
- 遗传算法实现技术
- 遗传算法同模拟退火算法的结合
- 支撑树的解表示



# 解决的优化问题

$$\begin{aligned} z &= \max f(x) \\ s.t. & g(x) \geq 0, \\ & x \in D, \end{aligned}$$

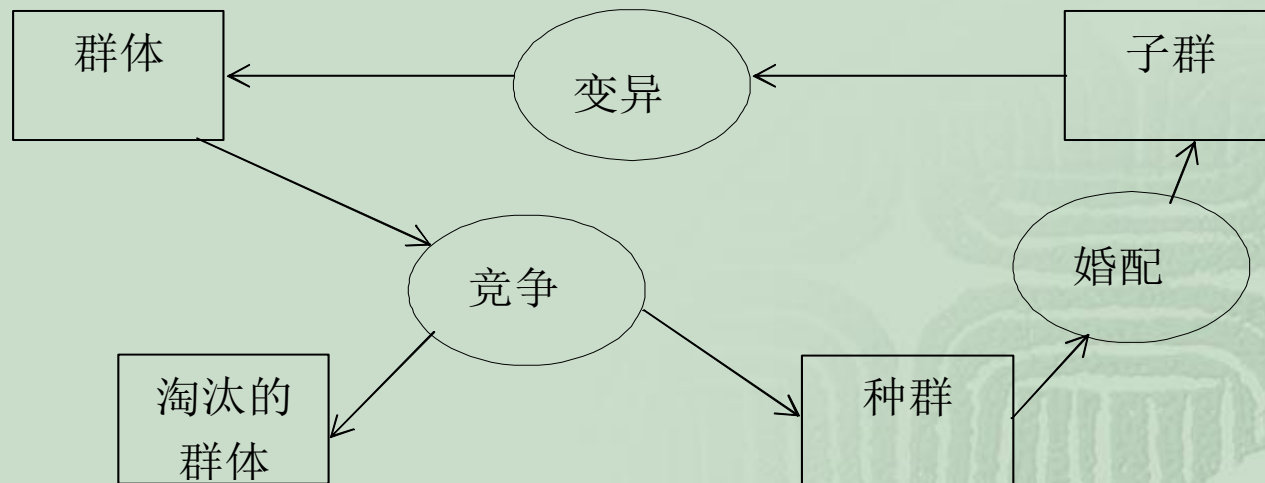


## 4.1 遗传算法

- 中国鲤鱼横行美国
- 美国白蛾威胁中国森林
- 荷兰郁金香在中国自然繁殖不超过2代
- 草坪野草泛滥
- 雾霾天是否戴口罩？
- 新冠的自身免疫性，放开还是严管？
- 适者生存

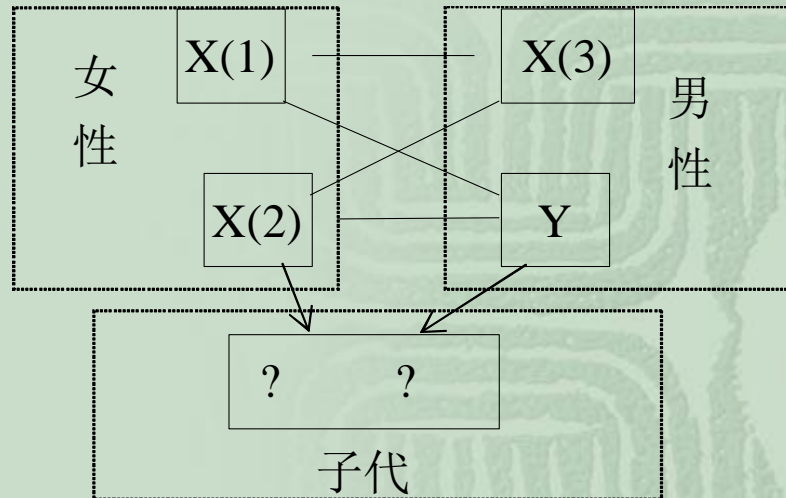


# 进化链



- 每个人包含有**46**条基因的染色体(chromosome)，组成**23**对同源染色体。男女性的结合使得对应的**23**对基因的染色体优胜劣汰再产生**23**对基因的染色体，因而形成一个新的生命。

染色体中  
性基因交  
叉规则





# 生物进化的一些特征

- 染色体通过基因表现
- 进化发生在染色体的基因上。
- 自然选择规律决定哪些染色体产生超过平均数的后代。
- 当染色体的结合时，双亲的遗传基因的结合使得子女保持父母的特征。
- 当染色体结合后，随机的变异会造成子代与父代的基因不同。
- 优化问题的解通过编码实现遗传过程，一切性质都通过编码来研究。编码和解码是遗传算法的一个主题。
- 遗传算法中，通过优化问题的目标而人为地构造适应函数以达到好的染色体产生超过平均数的后代。

# 生物遗传概念在遗传算法的对应关系

生物遗传概念	遗传算法中的作用
适者生存	在算法停止时，最优目标值的解有最大的可能性被留住
个体(individual)	解
染色体(chromosome)	解的编码（字符串，向量等）
基因(gene)	解中每一分量的特征（如各分量的值）
适应性(fitness)	适应函数值
群体(population)	选定的一组解（其中解的个数为群体的规模）
种群(reproduction)	根据适应函数值选取的一组解
交配(crossover)	通过交配原则产生一组新解的过程
变异(mutation)	编码的某一个分量发生变化的过程



# 一个模拟例子

用遗传算法求解  $f(x) = x^2, 0 \leq x \leq 31, x$  为整数的最大值。

第一步：编码

二进制编码，即0,1字符串。由于变量的最大值是31，因此可以采用5位数的二进制码。

10000→16      11111→31      01001→9      00010→2,

第二步：产生一个初始群体

$x_1=(00000), x_2=(11001), x_3=(01111), x_4=(01000)$

第三步：通过适应度选择一个交叉群体

$$p(x_i) = \frac{fitness(x_i)}{\sum_j fitness(x_j)}$$

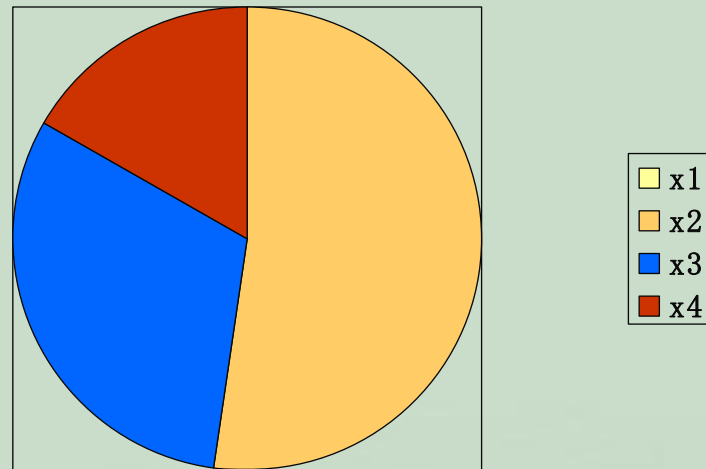
其中， $fitness(x)=x^2$



# 轮盘赌选种群

产生四个(0,1)区间的随机数，  
按落入的区域选取种群。

$C_1=X_2=(11001)$ ， $C_2=X_2=(11001)$ ，  
 $C_3=X_3=(01111)$ ， $C_4=X_4=(01000)$ 。



第四步：交叉。产生（1，2，3，4）的一个随机排列（1324），所以（13）（24）分别成对

产生1-4的一个随机整数：2  $\begin{matrix} x_2=(11|001) \\ x_3=(01|111) \end{matrix} \longrightarrow \begin{matrix} y_1=(11|111) \\ y_2=(01|001) \end{matrix}$

产生1-4的一个随机整数：3  $\begin{matrix} x_2=(110|01) \\ x_4=(010|00) \end{matrix} \longrightarrow \begin{matrix} y_3=(110|00) \\ y_4=(010|01) \end{matrix}$

第五步：变异，变异概率 $p_m$

产生1-4的一个随机整数：4。产生1-5的一个随机数：5。产生(0,1)区间一个随机数 $r$ ，当 $r < p_m$ 时， $y_4=(01000)$ ，否则，不变。  
再以 $y_1, y_2, y_3, y_4$ 为新的群体

## 遗传算法

STEP1 选择问题的一个编码；给出一个有  $N$  个染色体的初始群体  $POP(1)$ ， $t:=1$ ；

STEP2 对群体  $POP(t)$  中的每一个染色体  $pop_i(t)$  计算它的适应函数

$$f_i = fitness(pop_i(t));$$

STEP3 若停止规则满足，则算法停止；否则，计算概率

$$p_i = \frac{f_i}{\sum_{j=1}^N f_j}, i = 1, 2, \dots, N, \quad (1)$$

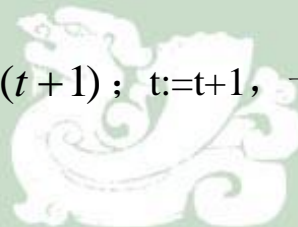
并以概率分布(1)从  $POP(t)$  中随机选一些染色体构成一个种群

$$NewPOP(t+1) = \{pop_j(t) \mid j = 1, 2, \dots, M\};$$

[注]  $NewPOP(t+1)$  集合中可能重复选  $POP(t)$  中的一个元素。

STEP4 通过交配，得到一个有  $N$  个染色体的  $CrossPOP(t+1)$ ；

STEP5 以一个较小的概率  $p$ ，使得一个染色体的一个基因发生变异，形成  $MutPOP(t+1)$ ； $t:=t+1$ ，一个新的群体  $POP(t) = MutPOP(t)$ ；返回 STEP2。



$$\max f(x) = 1 - x^2, x \in [0,1]$$

假设解的误差要求是 $1/16$ ，则可以采用四位二进制编码。对应关系为

$$(abcd) \leftrightarrow \frac{a}{2} + \frac{b}{4} + \frac{c}{8} + \frac{d}{16}$$

[illegible]

# 有待研究主要因素

- 解的编码和解码。
- 初始群体的选取和计算中群体的大小。
  - ∞ 一般采用随机产生初始群体或通过其他方法先构造一个初始群体。通过其他方法构造的初始群体可能会节省进化的代数，但也可能过早地陷入局部最优群体中。群体中个体的个数称为群体的维数。群体的维数越大，其代表性越广泛，最终进化到最优解的可能性越大。群体的维数越大，造成计算时间越长。
- 适应函数的确定
- 三个算子。
  - ∞ 种群选取、交配和变异或称突变。
  - ∞ 交配使得算法能够搜索更多的解，但受基因的限制，只能搜索已有的所有基因组合。
  - ∞ 变异是扩大染色体选择范围的一个手段，可以得到一些新的基因。

# 4.2 模板理论(schema theorem)

## ——数学解释现象

- 现象
  - ∞ 强势群体有很多相同的遗传基因特征
- 一个模板：一个给定的向量结构，包括关心的分量所在位置和值。
  - ∞ 基因和结构  $1^*1^*0^{**}$
  - ∞ 1010001 和 1111010 具有这样的模版
- 具有相同模板的两个染色体预示着具有一些共同的性质。
- 对于  $n$  维的向量，如果采用二进制编码，共有  $3^n$  个模板。



- 模板位置：关心的位置，一个模板中0或1所占的位置。
- 模板的长度(**length**)：从第一个模板位置到最后一个模板位置的所有分量个数减1，空隙的个数。
  - ∞ 如 $H=1*1*0^{**}$ 的长度为4。
  - ∞ 模板长度记为 $\delta(H)$ 。
- 模板的阶数(**order**)：模板位置对应的分量个数。
  - ∞ 如 $1*1*0^{**}$ 的阶数为3。
  - ∞ 记成 $o(H)$ 。
- 若染色体Y在H的模板位置上对应的分量相同，则称Y具有H模板。



# 基本结论

- 第一个算子：种群选取
- 引理 在生殖过程中，若个体以轮盘赌的概率被选入种群NewPOP( $t+1$ ), 且种群的规模同群体相同，则模板H所包含的染色体在 $t+1$ 时刻的期望数为

$$E_1(H, t+1) = f(H, t)N(H, t)$$

其中 $N(H, t)$ 为 $t$ 时刻 $T(H, POP(t)) = \{Y \in POP(t), Y \text{ 具有模板 } H\}$ 所包含的染色体数。

$$f(H, t) = \frac{\sum_{Y \in T(H, POP(t))} \frac{fitness(Y)}{|T(H, POP(t))|}}{\sum_{Y \in POP(t)} \frac{fitness(Y)}{|POP(t)|}}$$



证明：第*i*个体被选入种群的概率

$$p_i = \frac{f_i}{\sum_{j=1}^N f_j}, i = 1, 2, \dots, N,$$

选一个染色体，具有H模板被选中的概率为

$$\sum_{Y \in T(H, POP(t))} \frac{fitness(Y)}{\sum_{X \in POP(t)} fitness(X)}$$

种群NewPOP(t+1)的个体数是|POP(t)|，含H模板的染色体数为

$$|T(H, POP(t))| \frac{|POP(t)|}{|T(H, POP(t))|} \sum_{Y \in T(H, POP(t))} \frac{fitness(Y)}{\sum_{X \in POP(t)} fitness(X)}$$

## ■ 交叉算子

**引理** 若在  $t$  时刻, 模板  $H$  的长度为  $\delta(H)$ , 采用简单交配方法, 即随机选一个交配位, 交换位后基因, 交配的概率是  $p_c$ , 则在  $t+1$  时刻模板  $H$  保留下来的概率为

$$p_1(H, t+1) \geq 1 - \frac{p_c \delta(H)}{n-1} (1 - p(H, t)), \text{ 其中, } p(H, t) \text{ 表示 } t \text{ 时刻模板 } H \text{ 出现的概率。}$$

证明: 模板  $H$  的变化是由交配产生的。由于交配位后的基因交换, 因此选择第一个模板位到最后一个模板位中的任何一个位置为交配位都可能产生模板的变化, 所以, 模板发生改变的最大概率是  $p_c \frac{\delta(H)}{n-1}$ 。假设交配的双方有相同的模板, 则交配后模板不变。一方不具有同  $H$

相同模板的概率为  $1 - p(H, t)$ 。使模板  $H$  改变的最大概率为

$$p_c \frac{\delta(H)}{n-1} (1 - p(H, t)),$$

在交配的过程中, 会出现交配的双方不具有  $H$  模板, 但交配后却具有  $H$  模板, 故有

$$p_1(H, t+1) \geq 1 - p_c \frac{\delta(H)}{n-1} (1 - p(H, t))。$$



# 变异算子

**引理** 假设模板  $H$  在  $t$  时刻存在的概率为  $p(H,t)$ ，经过简单变异，则有  $p_2(H,t+1) \geq 1 - p_m o(H)$ ，其中， $p_m$  为每个基因变异概率， $o(H)$  为  $H$  的阶。

**证明：** 变异没有改变  $H$  模板的概率为

$$(1 - p_m)^{o(H)} \geq 1 - p_m o(H)$$



**定理**（模板定理----Schema theorem）假设群体在  $t$  时刻时含有模板  $H$  的染色体个数为  $N(H,t)$ ，经过种群选取、以概率  $p_c$  的交配和以概率  $p_m$  的变异，则在  $t+1$  时刻，群体中具有  $H$  模板的染色体数的期望值为

$$E(H, t + 1) \geq \left\{ 1 - \frac{p_c \delta(H)}{n - 1} (1 - p(H, t)) - p_m o(H) \right\} f(H, t) N(H, t)。$$

如果 
$$f(H, t) \geq \left\{ 1 - \frac{p_c \delta(H)}{n - 1} (1 - p(H, t)) - p_m o(H) \right\}^{-1},$$

则从概率意义来说，具有  $H$  模板的染色体个数将随代数  $t$  的增加而增加。



# 直观结论

$$E(H, t + 1) \geq \left\{ 1 - \frac{p_c \delta(H)}{n - 1} (1 - p(H, t)) - p_m o(H) \right\} f(H, t) N(H, t)$$

$$f(H, t) \geq \left\{ 1 - \frac{p_c \delta(H)}{n - 1} (1 - p(H, t)) - p_m o(H) \right\}^{-1}$$

- 要使具有H模板的染色体个数随代数t增加，必须要求f(H,t)不小于1。
  - ☞ 若要具有Y模板的染色体个数代代增加，必须使得它们的平均适应性不小于1。
- 要使模版的遗传可能性尽量的大，可以控制的参数有：使 $\delta(H)$ 尽量小，使 $o(H)$ 尽量小。
- 在平均适应性相同的前提下，新一代群体中，具有低阶、长度短模板的染色体期望数不低于那些具有高阶、长度长模板的染色体期望数。
- 低阶、长度短且平均适应性不低于1的模板一般称之为“积木块”(building block)

## 4.3 马尔可夫链收敛分析

- 遗传算法可以看成是一个马氏链
  - ∞ 优化问题的一个解
    - $X \in \Omega$
  - ∞ 状态
    - 一代群体  $(X_1, X_2, \dots, X_N)(k) \in \Omega^N, q = |\Omega^N|$
  - ∞ 经过种群选取、交叉和变异三个算子
    - 一个新的群体  $(X_1, X_2, \dots, X_N)(k+1)$
  - ∞ 三个算子有随机性
    - $Y(k) = (X_1, X_2, \dots, X_N)(k)$  为随机变量。
  - ∞ 确定  $Y(k)$  到  $Y(k+1)$  的一步转移概率。
  - ∞ 从  $Y(k)$  到  $Y(k+1)$  的转移概率不依赖于  $k$ 。
    - 时齐的马氏链



# 收敛标准

定义 假设优化问题要求目标函数  $f(x)$  达到最大。令

$$Z_t = \max\{f(X_j(t)) | j = 1, 2, \dots, N\},$$

其中  $Y(t) = (X_1(t), X_2(t), \dots, X_N(t))$  是第  $t$  代遗传后得到的一个种群。一个遗传算法收敛到全局最优，当且仅当

$$\lim_{t \rightarrow \infty} \Pr(Z_t = f^*) = 1,$$

其中，  $f^* = \max\{f(X) | X \in \Omega\}$ 。



交叉算子的概率矩阵

$$C = (c_{ij})_{q \times q}$$

性质

全概公式:  $\sum_{j=1}^q c_{ij} = 1$ 。

变异算子的概率矩阵

$$M = (m_{ij})_{q \times q}$$

性质

相同基因的位置总数记为  $H_{ij}$ 。由于采用染色体的

每个基因有相同的变异概率  $p_m > 0$ ，状态  $i$  变到状

态  $j$  的概率  $m_{ij} = p_m^{Nn-H_{ij}} (1-p_m)^{H_{ij}} > 0$ ，其中  $n$

为个体的编码长度， $N$  为群体的规模。



# 种群选取算子的概率矩阵

$$S = (s_{ij})_{q \times q}$$

仅考虑一个状态  $i$  经过种群选取不变的概率  $s_{ii}$ 。

假设状态  $i$  由  $N$  个染色体  $\{X_1, X_2, \dots, X_N\}$  组成，由种群选取的轮盘赌方法，

一个染色体  $X_j$  被选中的概率为

$$\frac{fitness(X_j)}{\sum_{k=1}^N fitness(X_k)} > 0,$$

状态  $i$  经过种群选取不变的概率

$$s_{ii} \geq \prod_{j=1}^N \left( \frac{fitness(X_j)}{\sum_{k=1}^N fitness(X_k)} \right) > 0。$$



定义一个方阵  $A \in R^{n \times n}$  称为:

(1)非负 ( $A \geq 0$ ), 若  $a_{ij} \geq 0, \forall i, j \in \{1, 2, \dots, n\}$  成立;

(2)全正 ( $A > 0$ ), 若  $a_{ij} > 0, \forall i, j \in \{1, 2, \dots, n\}$  成立;

(3)本原, 如果 A 非负且存在一个整数  $k \geq 1$  使得  $A^k > 0$ ;

(4)可约, 若 A 非负且经过相同的行和列初等变换得到如下形式

$$\begin{pmatrix} C & 0 \\ R & T \end{pmatrix}$$

其中, C 和 T 皆为方阵;

(5)不可约, 如果 A 非负且不可约;

(6)随机, 若 A 非负且  $\sum_{j=1}^n a_{ij} = 1, \forall i \in \{1, 2, \dots, n\}$ ;

(7)稳定, 若 A 是一个随机阵且所有行相同, 即每一列中的元素全部相同;

(8)列容, 若 A 是一个随机阵且每一列中至少有一个正数。





- 引理 若C、M和S是随机阵，其中M>0和S列容，则CMS为随机阵且CMS>0。

证明：记 A=CM 和 B=AS。因为 C 随机，则 C 的每一行中至少有一个正元素。由此

$$a_{ij} = \sum_{k=1}^n c_{ik} m_{kj} > 0, \forall i, j \in \{1, 2, \dots, n\},$$

即得 A>0。类似因为 S 是列容，得到

$$b_{ij} = \sum_{k=1}^n a_{ik} s_{kj} > 0, \forall i, j \in \{1, 2, \dots, n\}。$$

请证明 CMS 为随机阵。



引理 若  $P$  是本原随机阵, 则  $P^k$  收敛到一个全正稳定随机阵

$$P^\infty = \lim_{k \rightarrow \infty} P^k = (1, 1, \dots, 1)^T (p_1, p_2, \dots, p_n),$$

其中  $(p_1, p_2, \dots, p_n)^T$  唯一满足,

$$(p_1, p_2, \dots, p_n)P = (p_1, p_2, \dots, p_n), \quad \sum_{i=1}^n p_i = 1 \text{ 和 } p_j = \lim_{k \rightarrow \infty} p_{ij}^{(k)} > 0.$$

即  $(p_1, p_2, \dots, p_n)^T$  是矩阵  $P^T$  的特征值为 1 且每一个分量为正数的特征向量, 且

满足  $\sum_{i=1}^n p_i = 1$ 。



**定理** 若变异概率  $0 < p_m < 1$ ，交配概率  $0 \leq p_c \leq 1$ ，则简单遗传算法不收敛到全局最优值。

证 明： 设 状 态  $Y = (X_1, X_2, \dots, X_N)$  满 足

$\max\{f(X_j) | j = 1, 2, \dots, N\} < f^*$ ， $p_Y^t$  是遗传算法第  $t$  代在  $Y$  状态的

概率，则有  $\Pr\{Z_t = f^*\} \leq 1 - p_Y^t$ 。由交配矩阵  $C$ 、变异矩阵  $M$  和种群

选取矩阵  $S$  的性质，知  $CMS$  为全正矩阵， $\lim_{t \rightarrow \infty} p_Y^t = p_Y^\infty > 0$ ，因此，

$$\lim_{t \rightarrow \infty} \Pr(Z_t = f^*) \leq 1 - p_Y^\infty < 1。$$



- 记录前面各代遗传的最优解并存放在群体的第一位，这个染色体只起一个记录的功能而不参与遗传运算。第 $t$ 代进化时，马氏链状态的形式改进为

$$(X_{best}(t-1), X_1(t), \dots, X_N(t))$$

## 遗传算法的四个运算

交叉、变异、种群选取、最好解更新

马氏链一步转移矩阵形式：状态以目标非减序排列

$$\begin{array}{ccc}
 & (Y_1, |\Omega|^N) & \dots & (Y_{|\Omega|}, |\Omega|^N) \\
 (Y_1, |\Omega|^N) & [ & & ] \\
 \vdots & \vdots & & \vdots \\
 (Y_{|\Omega|}, |\Omega|^N) & [ & & ]
 \end{array}$$



## 改进简单遗传算法的交配、变异和种群选择转移阵

$$M^+ = \text{diag}(M, M, \dots, M)$$

$$C^+ = \text{diag}(C, C, \dots, C) \quad S^+ = \text{diag}(S, S, \dots, S)$$

$$W_1 = (Z_1, X_1, \dots, X_N) \quad W_2 = (Z_2, X_1, \dots, X_N)$$

$$\Pr(W_2 | W_1) = \begin{cases} 1, & Z_2 = \arg \max \{f(X_1), \dots, f(X_N)\} \text{ 且} \\ & f(Z_2) > f(Z_1), \\ 1, & \max \{f(X_1), \dots, f(X_N)\} \leq f(Z_1), Z_2 = Z_1, \\ 0, & \text{其它.} \end{cases}$$

$\arg \max \{f(X_1), \dots, f(X_N)\}$  最大函数值对应的第一个染色体。



$$U = \begin{pmatrix} U_{11} & & & \\ U_{21} & U_{22} & & \\ \vdots & \vdots & \ddots & \\ U_{|W|,1} & U_{|W|,2} & \cdots & U_{|W|,|W|} \end{pmatrix}$$

$U_{11}$  的每一列至少有一个 1。

**定理** 如果改进简单遗传算法按交配、变异、种群选取之后更新当前最优染色体（解）的进化循环过程，则收敛于全局最优。

证明思路：如果将第一位的染色体按目标值从好到坏的顺序排列，这个进化过程马氏链的一步转移概率为

$$\begin{aligned} P^+ = C^+ M^+ S^+ U &= \begin{pmatrix} CMS & & & \\ & CMS & & \\ & & \ddots & \\ & & & CMS \end{pmatrix} \begin{pmatrix} U_{11} & & & \\ U_{21} & U_{22} & & \\ \vdots & \vdots & \ddots & \\ U_{|W|,1} & U_{|W|,2} & \cdots & U_{|W|,|W|} \end{pmatrix} \\ &= \begin{pmatrix} CMSU_{11} & & & \\ CMSU_{21} & CMSU_{22} & & \\ \vdots & \vdots & \ddots & \\ CMSU_{|W|,1} & CMSU_{|W|,2} & \cdots & CMSU_{|W|,|W|} \end{pmatrix}. \end{aligned}$$





$$P^+ = \begin{bmatrix} Q & 0 \\ R & T \end{bmatrix} \quad Q = CMSU_{11}$$

$$R = \begin{pmatrix} CMSU_{21} \\ CMSU_{31} \\ \vdots \\ CMSU_{|\Omega|,1} \end{pmatrix} \quad T = \begin{pmatrix} CMSU_{22} & & \\ & \ddots & \\ CMSU_{|\Omega|,2} & & CMSU_{|\Omega|,|\Omega|} \end{pmatrix}$$

基本结论：本证明中假设只有一个最优解。当最优解不唯一时，同样可以得到相同的结论。

**Q全正**

**$U_{i1}$ 是一个非零的矩阵， $R$ 每一行中至少有一个非零元素。**

分2种情况讨论 $U_{i1}$ 是一个非零矩阵：（1）两个状态的第一个染色体目标值相同，（2）两个状态的第一个染色体目标值不同。

引理 若  $P = \begin{pmatrix} C & 0 \\ R & T \end{pmatrix}$  的随机阵，其中可约阵中  $C$  为一个  $m \times m$  全正随机阵， $R$  每一行中至少有一个非零元素，则

$$P^\infty = \lim_{k \rightarrow \infty} P^k = \lim_{k \rightarrow \infty} \begin{pmatrix} C^k & 0 \\ \sum_{i=0}^{k-1} T^i R C^{k-i} & T^k \end{pmatrix} = \begin{pmatrix} C^\infty & 0 \\ R_\infty & 0 \end{pmatrix},$$

是一个稳定的随机矩阵，满足

$$P^\infty = \lim_{k \rightarrow \infty} P^k = (1, 1, \dots, 1)^T (p_1, p_2, \dots, p_n),$$

$$\sum_{i=1}^n p_i = 1 \text{ 和 } p_j = \lim_{k \rightarrow \infty} p_{ij}^{(k)} \geq 0,$$

其中， $p_j > 0 (1 \leq j \leq m)$ ,  $p_j = 0 (m+1 \leq j \leq n)$ ,  $R_\infty$  表示  $\sum_{i=0}^{k-1} T^i R C^{k-i}$  的极限。



- **定理** 如果改进简单遗传算法按交配、变异后就更新当前最优染色体，之后再继续进行种群选取的进化循环过程，则收敛于全局最优。



## 4.4 实现的技术问题

- 常规0-1码
  - ∞ 背包问题的0-1码
  - ∞ TSP的0-1码
  - ∞ 划分问题的0-1码
- 非常规码：0-1以外的编码
  - ∞ TSP的城市排列序



# 约束机器排序问题的编码

## ■ 0-1码和非常规码。

$$\min T$$

$$s.t. \sum_{t=1}^T x_{it} = 1, i = 1, 2, \dots, n,$$

$$\sum_{i=1}^n d_i x_{it} \leq c_t, t = 1, 2, \dots, T,$$

$$x_{it} \in \{0, 1\}, i = 1, 2, \dots, n, t = 1, 2, \dots, T.$$

0-1编码形式，每个解用一个  $n \times T$  的向量表示。

非常规码：(1, 2, ..., n) 一个加工序。



例  $d_1 = 5, d_2 = 3, d_3 = 10, d_4 = 4$

前五个时段可提供能力为(3,9,10,5,20)，若按(1,2,3,4)顺序加工，各时段加工的产品为：第一个时段不加工，第二个时段加工产品1和2，第三个时段加工产品3，第四个时段加工产品4。





# 进化效能监测

- 在线(on-line)比较法。用进化过程中的每一个解来了解进化趋势。 $T$ 为到目前出现的染色体

$$v^{on-line}(T) = \frac{1}{T} \sum_{t=1}^T v(t)$$

- 离线(off-line)比较法。用进化过程中的每一代的解来了解进化趋势。

$$v^{off-line}(T) = \frac{1}{T} \sum_{t=1}^T v^*(t)$$



# ■ 初始参数的选取和停止原则

## ∞ 群体的规模

- 为编码长度数的多项式关系

## ∞ 初始群体的选取

- 随机选取 —— 加大了进化的代数
- 用其他的一些启发式算法或经验选择一些比较好的染色体（种子） —— 早熟

## ∞ 终止规则

- 给定一个最大的遗传代数 **MAXGEN**
- 给定问题一个上界 **UB**，达到偏差度  $\varepsilon$ ，算法终止。
- 自适应性：算法再进化已无法改进解的性能。



# 进化过程中的技术问题

## ■ 适应函数

例 已知优化问题为:  $z = \max_{x \in [0.5, 1]} f(x) = \max_{x \in [0.5, 1]} \{1 + \log x\}$  。

若采取四位编码且适应函数为优化问题的目标函数。

x	群体	fitness(x)	概率分布
1/2	0000	0.689	0.214
5/8	0100	0.796	0.248
21/32	0101	0.817	0.254
23/32	0111	0.857	0.272

由于它们四个的概率分布值相差很小, 因此哪一个染色体占先也不是非常显著。



# 非线性加速适应函数

$$fitness(x) = \begin{cases} \frac{1}{1-x}, & 0.5 \leq x < 1, \\ M > 0, & 1 \leq x, \end{cases}$$

x	群体	fitness(x)	概率分布
1/2	0000	3.322	0.161
5/8	0100	4.902	0.237
21/32	0101	5.464	0.264
23/32	0111	6.993	0.338

$$fitness(x) = \begin{cases} \frac{1}{f_{\max} - f(x)}, & f(x) < f_{\max}, \\ M > 0, & f(x) = f_{\max}, \end{cases}$$

x	群体	fitness(x)	概率分布
1/2	0000	5.376	5.376/(46.769+M)
5/8	0100	16.393	16.393/(46.769+M)
21/32	0101	25.000	25/(46.769+M)
23/32	0111	M	M/(46.769+M)

# 两种模型比较

- 后一种易于实现，不需要知道原有函数的最优解。
- 选取M的策略是：初始迭代时，M同第一大与第二大目标差值的倒数尽量接近以避免早熟，后期迭代中逐步扩大差距。也可以在早期迭代中用简单的适应函数，而在后期用这类加速的方法。



# 线性加速适应函数

$$fitness(x) = \alpha f(x) + \beta$$

$$\left\{ \begin{array}{l} \alpha \frac{\sum_{i=1}^m f(x_i)}{m} + \beta = \frac{\sum_{i=1}^m f(x_i)}{m} \\ \alpha \max_{1 \leq i \leq m} \{f(x_i)\} + \beta = M \frac{\sum_{i=1}^m f(x_i)}{m} \end{array} \right.,$$

$$\alpha = \frac{(M-1) \frac{\sum_{i=1}^m f(x_i)}{m}}{\max_{1 \leq i \leq m} \{f(x_i)\} - \frac{\sum_{i=1}^m f(x_i)}{m}},$$

$$\beta = \frac{\sum_{i=1}^m f(x_i)}{m} \left[ \frac{\max_{1 \leq i \leq m} \{f(x_i)\} - M \frac{\sum_{i=1}^m f(x_i)}{m}}{\max_{1 \leq i \leq m} \{f(x_i)\} - \frac{\sum_{i=1}^m f(x_i)}{m}} \right]$$



# 排序适应函数

- 将同一代群体中的 $m$ 个染色体按目标函数值从小到大排列，直接取分布概率为

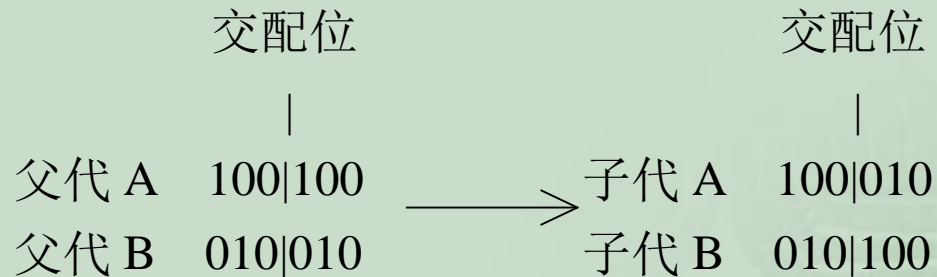
$$p(i) = \frac{2i}{m(m+1)}, \quad 1 \leq i \leq m$$



# 交配规则

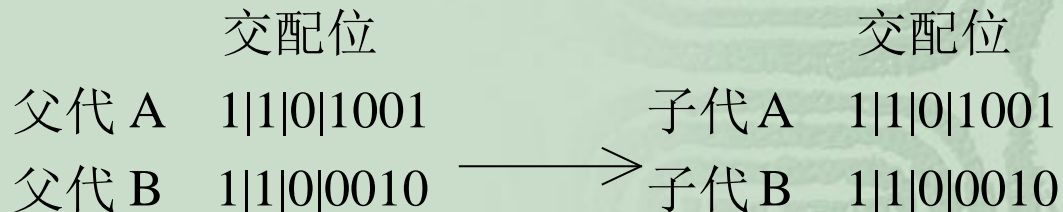
## ■ 常用方法——双亲双子

∞ 以一个随机位进行位之后的所有基因对换。对换后形成两个后代。



## ■ 变化交配法(string-of-change crossover)

∞ 从不同基因位置按常规方法随机选交配位。



## ■ 多交配位法(multi-point crossover)

	交配位		交配位	
父代 A	11 01 001	→	子代 A	11 00 001
父代 B	11 00 010		子代 B	11 01 010

## ■ 显性遗传法(dominance)

父代 A	1101001	→	子代	1101011
父代 B	0000010			

## ■ 非常规码的交配方法

	交配位		交配位	
父代 A	213 4567	→	子代 A	213 4567
父代 B	431 2567		子代 B	431 2567



# 种群的选取和交配后群体的确定

- 种群的选取、交配和变异用常规的方法，只是在 $\text{MutPOP}(t+1)$ 中选最优的 $L$ 个染色体替换 $\text{POP}(t)$ 中最差的 $L$ 个染色体；
- 选择种群中染色体的个数只是群体的一个比例，此时采用常规的交配方法，交配概率为1，交配后的子代同 $\text{POP}(t)$ 中的染色体通过筛选组成 $\text{POP}(t+1)$ ；
- 采用一些非常用的交配方法，用交配、变异后的子代同 $\text{POP}(t)$ 通过筛选组成 $\text{POP}(t+1)$ 。

## ■ 控制替换率

$$G(t+1) = 1 - \frac{POP(t) \cap POP(t+1)}{m}$$

- 100% ( $G=1$ ) 的替换使得新群体和旧群体的染色体相重率降低。直观看计算效果应该好。但有可能使得当前最优解无法遗传下去。
- 低替换率产生过多地重复计算适应函数值, 使得搜索的范围扩展较慢。它的优点是使得某些关心的染色体得以保留。
- 在群体或种群中有时会出现相同的染色体。相同的染色体造成适应函数的重复计算, 但同时也是适应能力的一种表现, 有扩大下一代相同染色体生存的可能性。对这个问题不必作太多的处理。
- 100%的替换会出现当前最优解的遗失, 一种保持的策略是使得上一代的当前最优解强行遗传到下一代。



## 4.5 遗传模拟退火算法

遗传模拟退火算法

STEP1 给定群体规模 MAXPOP,  $k:=0$ ; 初始温度  $t_k := t_0$ , 群体 POP(k);

STEP2 若满足停止规则, 停止计算; 否则, 对群体 POP(k)中每一个染色体  $i \in POP(k)$  的邻域中随机选状态  $j \in N(i)$ , 按模拟退火中的接受概率

$$A_{ij}(t_k) = \min\{1, \exp(-\frac{f(j) - f(i)}{t_k})\},$$

接受或拒绝  $j$ , 其中  $f(i)$  为状态  $i$  的目标值; 这一阶段共需 MAXPOP 次迭代选出新群体 NewPOP1(k+1);

STEP3 在 NewPOP1(k+1)中计算适应函数

$$f_i(t_k) = \exp\{-\frac{f(i) - f_{\min}}{t_k}\},$$

其中,  $f_{\min}$  是 NewPOP1(k+1)中的最小值; 由适应函数决定的概率分布从 NewPOP1(k+1)中随机选 MAXPOP 个染色体形成种群 NewPOP2(k+1);

STEP4 按遗传算法的常规方法进行交配得到 CrossPOP(k+1); 再变异得到 MutPOP(k+1);

STEP5  $t_{k+1} := d(t_k)$ ,  $k := k + 1$ , POP(k)=MutPOP(k), 返回 STEP2。



# 案例：最小支撑树解表示

$$G = (V, E)$$

$$V = \{v_1, v_2, \dots, v_n\}$$

$$E = \{e_{ij} \mid e_{ij} = (v_i, v_j)\}$$

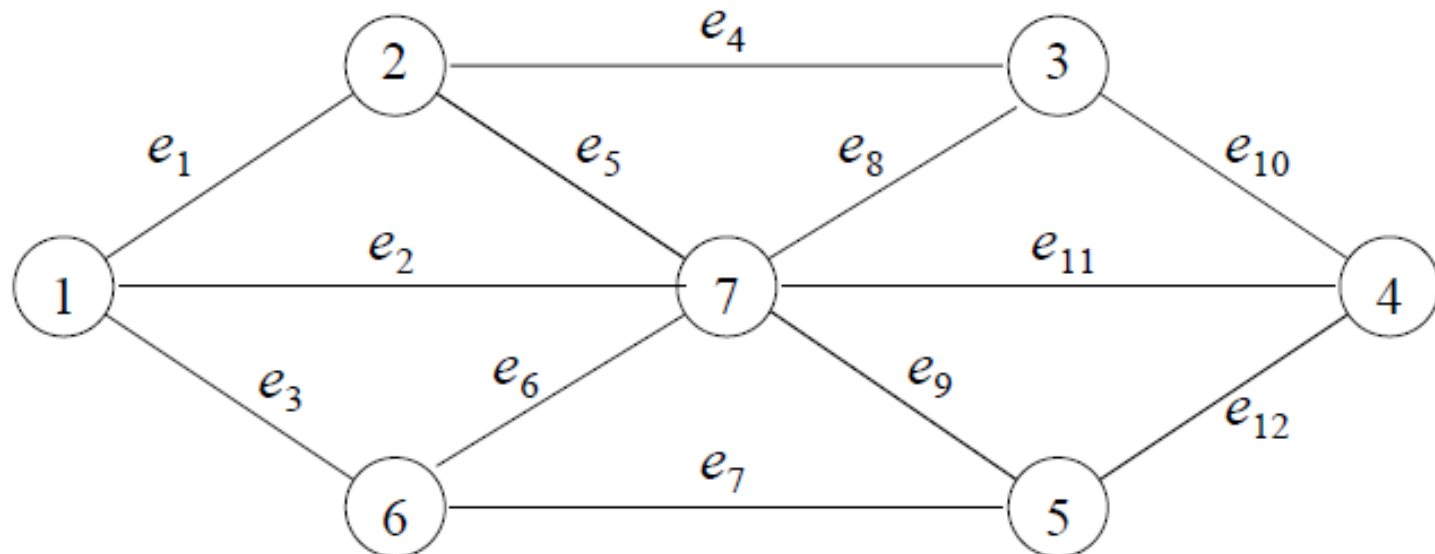
$$W = \{w_{ij} \mid w_{ij} = w(e_{ij}) > 0\}$$

A spanning tree is a minimal set of edges from  $E$  that connects all the vertices in  $V$  and therefore at least one spanning tree can be found in graph  $G$ . The minimum spanning tree, denoted as  $T^*$ , is the spanning tree whose total weight of all edges is minimal, i.e.,

$$T^* = \min_T \sum_{e_{ij} \in E \cap T} w_{ij}$$

- Encoding

(1) Edge encoding



$e_1$	$e_2$	$e_3$	$e_4$	$e_5$	$e_6$	$e_7$	$e_8$	$e_9$	$e_{10}$	$e_{11}$	$e_{12}$
0	1	0	1	1	0	1	0	1	0	0	1

## Problems:

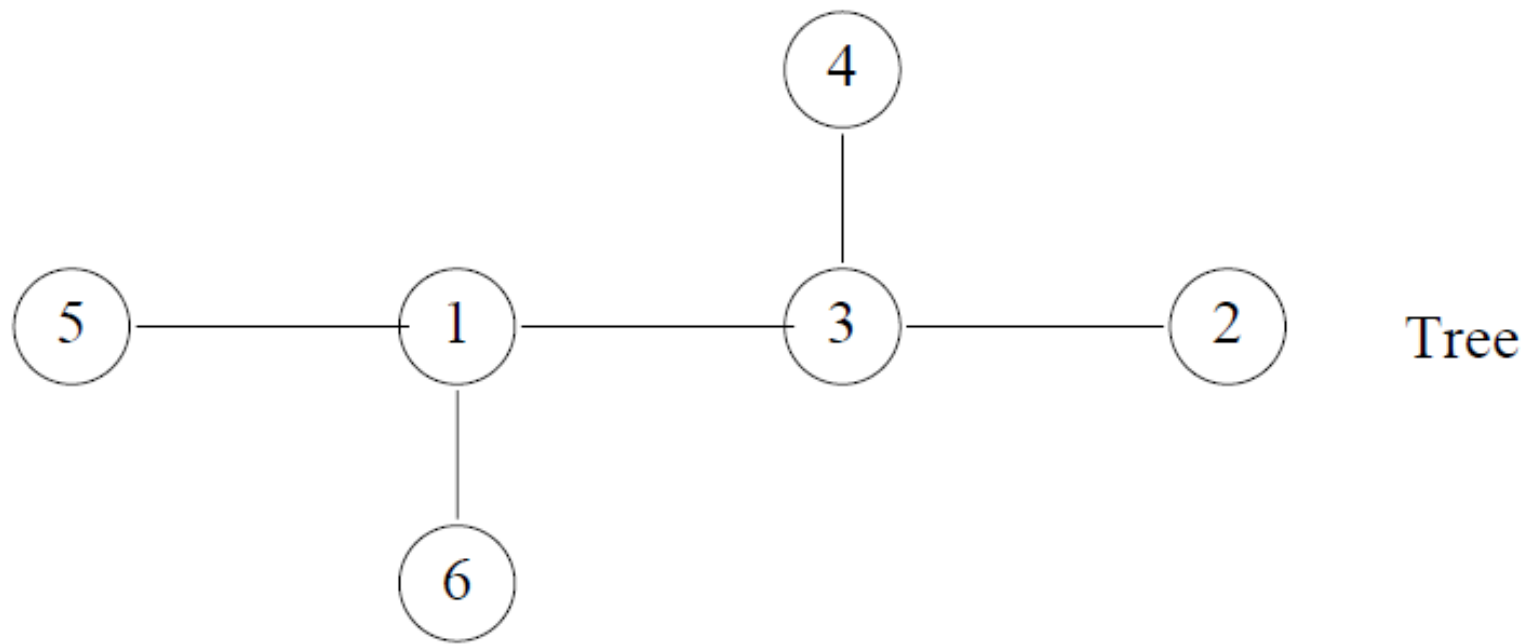
- Representation space is too large.
- Probability of having a tree is too low.

### (2) Prüfer number encoding:

Cayley's Theorem: There are  $n^{n-2}$  distinct labeled tree on a complete graph with  $n$  vertices.

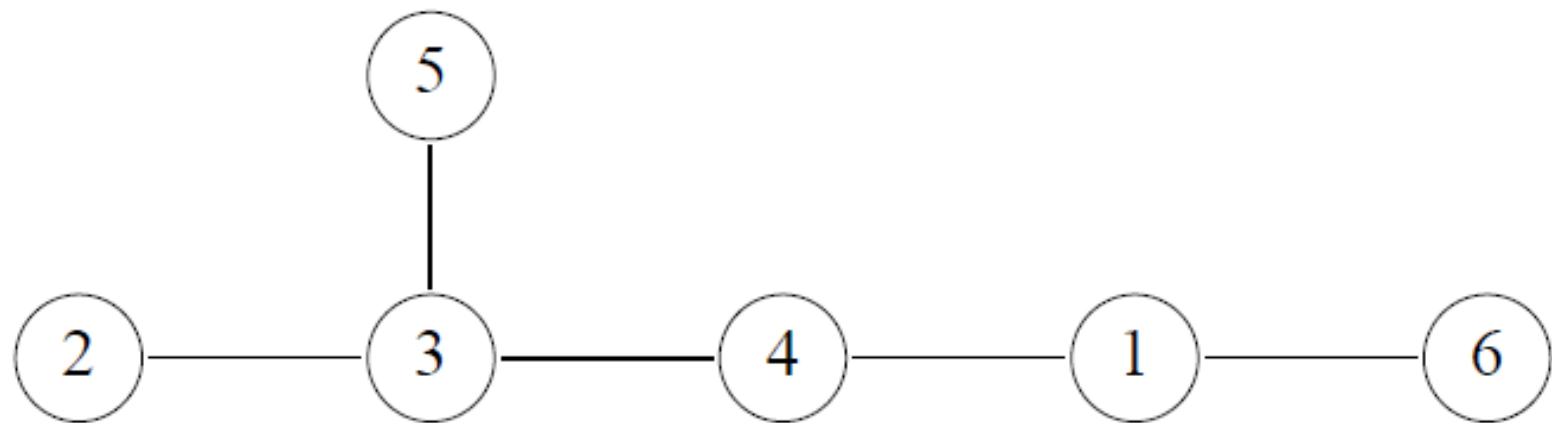
(Prüfer established a one-to-one correspondence between such trees and the set of all strings of  $n - 2$  digits.)

## Examples



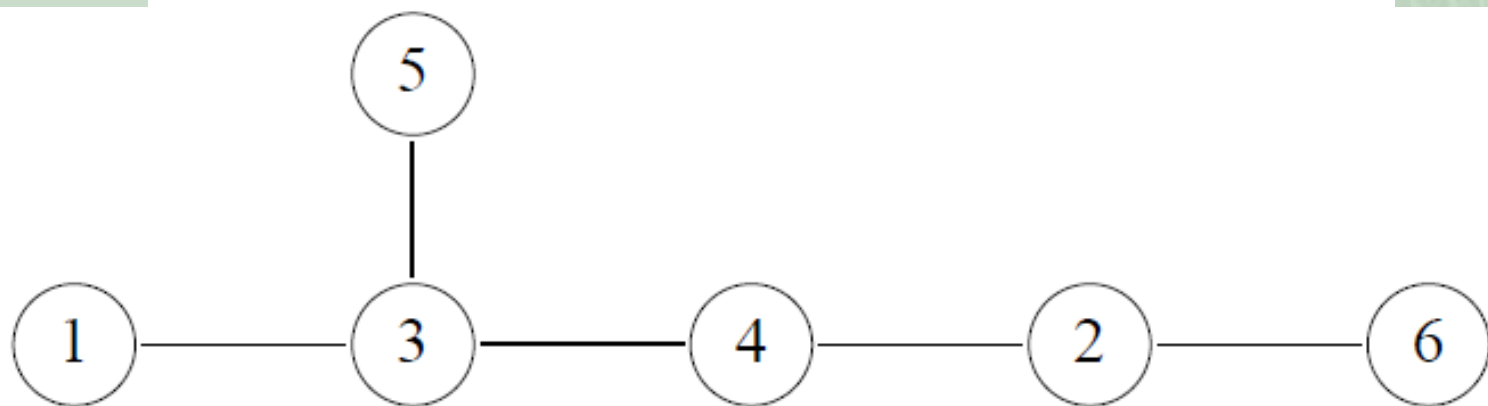
3	3	1	1
---	---	---	---

Prufer number



3	3	4	1
---	---	---	---

Prufer number



3	3	4	2
---	---	---	---

Prufer number

# 编码 (encoding)

- Step 1.** Let vertex  $i$  be the smallest labeled leaf vertex in a labeled tree  $T$ .
- Step 2.** Let  $j$  be the first digit in the encoding as the vertex  $j$  incident to vertex  $i$  is uniquely determined. Here we build the encoding by appending digits to the right, and thus the encoding is built and read from left to right.
- Step 3.** Remove vertex  $i$  and the edge from  $i$  to  $j$ ; thus we have a tree with  $n - 1$  vertices.
- Step 4.** Repeat the above steps until one edge is left. We produce a Prüfer number or an encoding with  $n - 2$  digits between 1 and  $n$  inclusive.



# 解码 (decoding)

- Step 1.** Let  $P$  be the original Prüfer number and let  $\bar{P}$  be the set of all vertices not included in  $P$ .  $P$  designates vertices eligible for consideration.
- Step 2.** Let  $i$  be the eligible vertex in  $\bar{P}$  with the smallest label. Let  $j$  be the leftmost digit of  $P$ . Add the edge from  $i$  to  $j$  into the tree. Remove  $i$  from  $\bar{P}$  and  $j$  from  $P$ . If  $j$  does not occur anywhere in  $P$ , put it into  $\bar{P}$ . Repeat the process until no digits are left in  $P$ .
- Step 3.** If no digits remain in  $P$ , there are exactly two vertices,  $r$  and  $s$ , in  $\bar{P}$  and thus still eligible for consideration. Add edge from  $r$  to  $s$  into the tree to form a tree with  $n - 1$  edges.

- Crossover

cut-point



parent 1

2	3	4	7	1	8	9
---	---	---	---	---	---	---

parent 2

8	4	6	2	8	9	7
---	---	---	---	---	---	---



offspring 1

2	3	4	7	8	9	7
---	---	---	---	---	---	---

offspring 2

8	4	6	2	1	8	9
---	---	---	---	---	---	---

- Mutation

select a position at random



parent

6	2	7	8	3	2	9
---	---	---	---	---	---	---

replace with a digit at random



offspring

6	2	7	8	5	2	9
---	---	---	---	---	---	---

- Evaluation

**Step 1.** Convert a chromosome into a tree.

**Step 2.** Calculate the total weight of the tree.

- Let  $P$  be a chromosome, and let  $\bar{P}$  be the set of eligible vertices.

## Procedure: Evaluation

**begin**

$T \leftarrow \{\phi\};$

$eval(T) \leftarrow 0;$

define  $\bar{P}$  according to  $P$ ;

**repeat**

    select the leftmost digit from  $P$ , say  $i$ ;

    select the eligible vertex with the smallest label  
        from  $\bar{P}$ , say  $j$ ;

$T \leftarrow T \cup \{e_{ij}\};$

$eval(T) \leftarrow eval(T) + w_{ij};$

    remove  $i$  from  $P$ ;

    remove  $j$  from  $\bar{P}$ ;

**if**  $i$  does not occur anywhere in remaining  $P$  **then**

    put  $i$  into  $P$ ;

**end**

$k \leftarrow k + 1$ ;

**until**  $k \leq n - 2$ ;

$T \leftarrow T \cup \{e_{rs}\}, r, s \in \bar{P}$ ;

$eval(T) \leftarrow eval(T) + w_{rs}$ ;

**end**





- Selection

- a mixed strategy with  $(\mu + \lambda)$  selection and *roulette wheel* selection is used.



## Procedure: Selection

begin

select  $\mu'$  best different chromosomes;

if  $\mu' < \mu$  then

select  $\mu - \mu'$  chromosomes by *roulette wheel* selection;

end

end



- Algorithm

Procedure: Genetic Algorithm for dc-MST

begin

$t \leftarrow 0$ ;

    initialize  $P(t)$ ;

    evaluate  $P(t)$ ;

**while** (not termination condition) **do**

        recombine  $P(t)$  to yield  $C(t)$ ;

        evaluate  $C(t)$ ;

        select  $P(t + 1)$  from  $P(t)$  and  $C(t)$ ;

$t \leftarrow t + 1$ ;

**end**

**end**

## Parameters:

population size  $pop\_size = 50$ ;

crossover probability  $p_c = 0.5$ ;

mutation probability  $p_m = 0.01$ ;

maximum generation  $max\_gen = 500$ ;



# 计算结果

