

组合优化理论

第11章 启发式算法

主讲人：陈安龙

第11章 启发式算法

§ 1 模拟退火算法

§ 2 模拟遗传算法

本章介绍的启发式算法也称**智能算法、现代优化算法**，自20世纪80年代初以来已得到深入研究和广泛应用，主要包括**模拟退火算法、神经网络算法、遗传算法、蚁群优化算法**等等。

它们的共性是基于客观世界中的一些自然现象，通过与组合优化求解进行类比，找出它们的一些共性，建立相应的算法。

生命科学与工程科学的相互交叉、相互渗透、相互促进是近代科学技术发展的显著特点之一。

这些算法的目标是希望求解 **NP 难问题** 的全局最优解，有一定的普适性，可用于解决大量的实际应用问题。

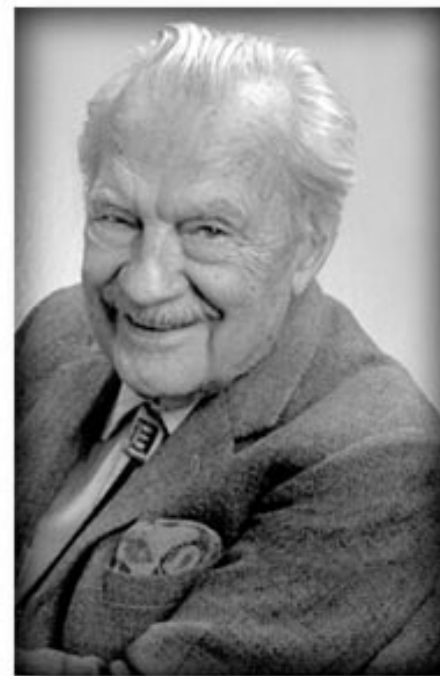
§ 1 模拟退火算法

◆ 算法的提出

模拟退火算法最早的思想由**Metropolis**等（1953）提出，**1983年Kirkpatrick**等将其应用于组合优化。

◆ 算法的目的

解决NP复杂性问题；
克服优化过程陷入局部极小；
克服初值依赖性。



Nick Metropolis

§ 1 模拟退火算法

◆ 物理退火过程

什么是退火：

退火是指将固体加热到**足够高的温度**，使分子呈随机排列状态，然后逐步**降温使之冷却**，最后分子以低能状态排列，固体达到**某种稳定状态**。



§ 1 模拟退火算法

◆ 物理退火过程

加温过程——增强粒子的热运动，消除系统原先可能存在的非均匀态；

等温过程——对于与环境换热而温度不变的封闭系统，系统状态的自发变化总是朝自由能减少的方向进行，当自由能达到最小时，系统达到平衡态；

冷却过程——使粒子热运动减弱并渐趋有序，系统能量逐渐下降，从而得到低能的晶体结构。

◆ 热力学中的退火现象：

温度越低，物体的能量状态越低，到达足够的低点时，液体开始冷凝与结晶，在结晶状态时，系统的能量状态最低。**缓慢降温（退火，annealing）**时，可达到**最低能量状态**；但如果**快速降温（淬火，quenching）**，会导致不是最低能态的**非晶形**。

◆ 大自然知道慢工出细活：

缓缓降温，使得物体分子在每一温度时，能够有足够时间找到安顿位置，则逐渐地，到最后可得到**最低能态，系统最稳定**。

◆ 研究退火现象的目的

模仿自然界退火现象而得，利用了物理中固体物质的退火过程与一般优化问题的相似性

从某一初始温度开始，伴随温度的不断下降，结合概率突跳特性在解空间中随机寻找全局最优解

一、原理与步骤

由统计力学的研究表明，在温度 T ，分子停留在状态 r 满足 *Boltzmann* 概率分布

$$P\{\bar{E} = E(r)\} = \frac{1}{Z(T)} \exp\left(-\frac{E(r)}{k_B T}\right)$$

其中， $E(r)$ 表示状态 r 的能量， $k_B > 0$ 为 *Boltzmann* 常数，表示分子能量的一个随机变量。 $Z(T)$ 为概率分布的标准化因子

$$Z(T) = \sum_{s \in D} \exp\left(\frac{E(s)}{k_B T}\right)$$

退火过程的数学表达

- 在同一个温度 T ，选定两个能量 $E_1 < E_2$ ，有

$$P\{\bar{E} = E_1\} - P\{\bar{E} = E_2\} = \frac{1}{Z(T)} \exp\left(-\frac{E_1}{k_B T}\right) \left[1 - \exp\left(-\frac{E_2 - E_1}{k_B T}\right) \right]$$

> 0
 < 1

模拟退火算法基本思想：在一定温度下，搜索从一个状态随机地变化到另一个状态；随着温度的不断下降直到最低温度，搜索过程以概率1停留在最优解

Boltzman概率分布揭示

- (1) 在同一个温度，分子停留在**能量小状态的概率**大于停留在能量大状态的概率
- (2) 温度越高，不同能量状态对应的概率相差越小；
温度足够高时，各状态对应概率基本相同。
- (3) 随着温度的下降，能量最低状态对应概率越来越大；**温度趋于0时，其状态趋于1**

◆ **Metropolis准则（1953）** ——以概率接受新状态

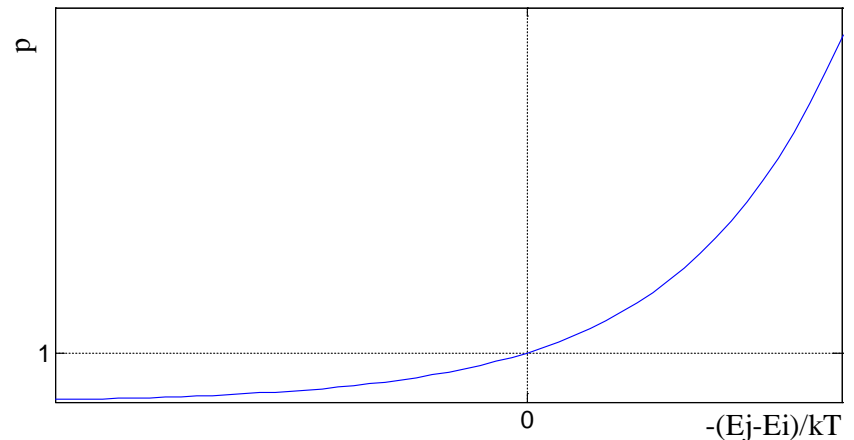
固体在恒定温度下达到热平衡的过程可以用**Monte Carlo方法**（计算机随机模拟方法）加以模拟，虽然该方法简单，但必须大量采样才能得到比较精确的结果，计算量很大。

◆ Metropolis准则（1953）——以概率接受新状态

若在温度 T ，当前状态 $i \rightarrow$ 新状态 j

若 $E_j < E_i$ ，则接受 j 为当前状态；

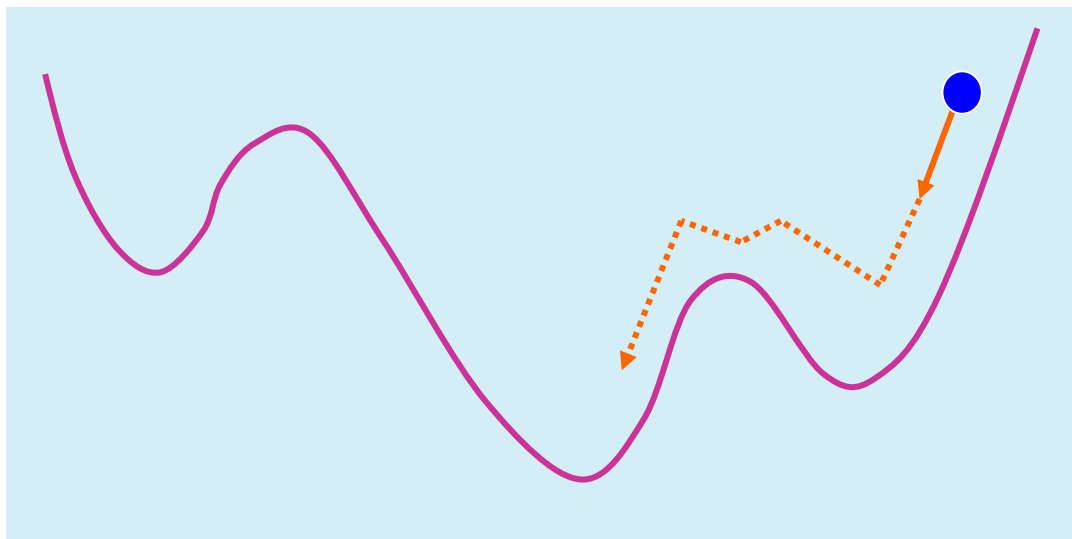
否则，若概率 $p = \exp[-(E_j - E_i)/k_B T]$ 大于 $[0,1)$ 区间的随机数，则仍接受状态 j 为当前状态；若不成立则保留状态 i 为当前状态。



◆ Metropolis准则（1953）——以概率接受新状态

$$p = \exp[-(E_j - E_i)/k_B T]$$

在高温下，可接受与当前状态能量差较大的新状态；
在低温下，只接受与当前状态能量差较小的新状态。



组合优化与物理退火的相似性

组合优化问题	金属物体
解	粒子状态
最优解	能量最低的状态
设定初温	熔解过程
Metropolis抽样过程	等温过程
控制参数的下降	冷却
目标函数	能量

模拟退火算法的基本思想和步骤

◆ 基本步骤

给定初温 $t=t_0$ ，随机产生初始状态 $s=s_0$ ，令 $k=0$ ；

Repeat

Repeat

产生新状态 $s_j=\text{Genete}(s)$ ；

if $\min\{1, \exp[-(C(s_j)-C(s))/t_k]\} \geq \text{randrom}[0,1]$ $s=s_j$ ；

Until 抽样稳定准则满足；

退温 $t_{k+1}=\text{update}(t_k)$ 并令 $k=k+1$ ；

Until 算法终止准则满足；

输出算法搜索结果。

模拟退火算法的基本思想和步骤

◆ 影响优化结果的主要因素

给定初温 $t=t_0$ ，随机产生初始状态 $s=s_0$ ，令 $k=0$ ；

Repeat

Repeat

产生新状态 $s_j = \text{Genete}(s)$;

if $\min\{1, \exp[-(C(s_j) - C(s))/t_k]\} \geq \text{randrom}[0,1]$ $s=s_j$;

Until 抽样稳定准则满足;

退温 $t_{k+1} = \text{update}(t_k)$ 并令 $k=k+1$;

Until 算法终止准则满足;

输出算法搜索结果。

三函数两准则

初始温度

简单的模拟退火算法的步骤:

初始温度

Step 1 任选一初始解 x_{i_0} ; $x_i = x_{i_0}$; $k = 0$; $t_0 = t_{max}$.

Step 2 若在该温度达到内循环停止条件, 则到 *Step 3*;
否则, 从邻域 $N(x_i)$ 中随机选取一 x_j , 计算

$\Delta C_{ij} = C(x_j) - C(x_i)$; 若 $\Delta C_{ij} \leq 0$, 则 $x_i = x_j$, 否则

若 $\exp(-\Delta C_{ij} / t_k) \geq \text{random}(0,1)$ 时, 则 $x_i = x_j$; 重复

Step 2.

Step 3 $t_{k+1} = d(t_k)$; $k = k+1$; 若满足停止条件, 终止计算; 否则, 回到 *Step 2*.

在 (0,1) 之间均匀分布的伪随机数

温度下降规则
(衰减函数)

温度下降规则；迭代
步数增加；停止条件.

在上述的 SA 算法中，包含一个内循环和一个外循环。内循环是 *Step 2*，它表明在同一个温度 t_k 时，在一些状态随机搜索。外循环主要包括 *Step 3*。

SA 算法的每一次迭代都体现集中和扩散两个策略的平衡。对遇到的下一个迭代解，如果这个解更好，则采用集中策略，选择这个解为新解。不然的话，采用扩散策略，以某一概率选择这个解为新解。

由于在给定领域结构后，模拟退火过程是从一个状态到另一个不断地随机游动，所以可以用 **Markov** 链来描述。

马尔科夫链

设 $\{X(k)\}_{k=0,1,2,\dots}$ 为随机变量序列， $X(k) = i$ 称在时刻 k 处于状态 i ， $i \in D$ 。 D 称为状态空间。

当 D 中的状态数有限时，称为有限状态空间。

若对任意的正整数 n ,

$$\begin{aligned} &P\{X(n) = j \mid X(1) = i_1, X(2) = i_2, \dots, X(n-1) = i\} \\ &= P\{X(n) = j \mid X(n-1) = i\} \quad i_1, i_2, \dots, i, j \in D. \end{aligned}$$

则称 $\{X(k)\}_{k=0,1,2,\dots}$ Markov 链。 Markov 链具有记忆遗忘特性，它只记忆前一时刻的状态。

模拟退火算法主要可以分为两类：

第一类为**时齐算法**，对每一个固定的 t ，计算对应的 Markov 链，直至**达到一个稳定状态**，然后再让**温度下降**；

第二类为**非时齐算法**，基本计算框架是由一个 Markov 链组成，要求在**两个相邻的转移中**，**温度 t 是下降的**。

马尔科夫链

◆ 定义

一步转移概率:

$$p_{i,j}(n-1) = \Pr\{X(n) = j \mid X(n-1) = i\}$$

n 步转移概率:

$$p_{i,j}^{(n)} = \Pr\{X(n) = j \mid X(0) = i\}$$

若解空间有限, 称马尔可夫链为有限状态;

若 $\forall n \in \mathbb{Z}^+, p_{i,j}(n) = p_{i,j}(n-1)$ 称**马尔可夫链为时齐的**。

模拟退火算法与马尔科夫链

◆ 模拟退火算法对应了一个马尔可夫链

模拟退火算法：新状态接受概率仅依赖于新状态和当前状态，并由温度加以控制。

若固定每一温度，算法均计算马氏链的变化直至平稳分布，然后下降温度，则称为**时齐算法**；

若无需各温度下算法均达到平稳分布，但温度需按一定速率下降，则称为**非时齐算法**。

◆ 分析收敛性

模拟退火算法收敛性分析

收敛性分析的数学基础-马尔可夫随机过程

举例说明：盲人一维游走、醉汉或青蛙在3块石头上随机跳动，这3中状况可用来说明这个问题，他们行动的共同特点是无记忆性。

状态：

处于系统中的一种特定状态表达。

状态转移概率：

从状态 i 转移到状态 j 的可能性。

无后效性：

到一个状态后，决策只与本状态有关，与以前的历史状态无关。

收敛性分析

-确定模拟退火算法中的转移概率矩阵

考虑定义在由**N**个元素组成的有限集**S**上的优化问题

$$\min f(x), x \in S$$

对所有状态下的目标值按从小到大排序，有

$$f(x_1) \leq f(x_2) \leq \cdots \leq f(x_N),$$

收敛性分析

-确定模拟退火算法中的转移概率矩阵

设 p_{ij} 为 x_i 选 x_j 为邻域点时，从状态 x_i 到状态 x_j 的转移概率

$$P_{ij}(T_k) = \begin{cases} \alpha_{ij}(T_k) \hat{P}_{ij}, i \neq j \\ 1 - \sum_{k \neq i} \alpha_{ik}(T_k) \hat{P}_{ik}, i = j \end{cases}$$

$\hat{P}_{ij} = 1/N(i)$ ， $N(i)$ 为状态 i 的邻域点的个数

其中 $\alpha_{ij}(T_k)$ 为在温度 T_k 时的条件转移因子

$$\begin{cases} \alpha_{ij}(T_k) = 1, & \text{if } f(x_j) \leq f(x_i) \\ 0 \leq \alpha_{ij}(T_k) < 1, & \text{if } f(x_j) > f(x_i) \end{cases}$$

收敛性分析结果

设 $P_{ij}=P_{ji}$, 若在温度 T_k 达到热平衡(状态平稳分布), 此时各状态的平稳概率分布为:

$$\pi(T_k) = \pi_1(T_k) [1, a_{12}(T_k), a_{13}(T_k), \dots, a_{1N}(T_k)]$$

作为推论, 当在温度 $T_k=0$ 时到热平衡(状态平稳分布), 此时各状态的平稳概率分布为:

$$\begin{aligned} \pi(0) &= \pi_1(0) [1, \alpha_{12}(0), \alpha_{13}(0), \dots, \alpha_{1N}(0)] \\ &= [1, 0, 0, \dots, 0] \end{aligned}$$

这说明算法以概率1转移到使得目标函数最优的状态 x_1

状态产生函数

◆ 原则

产生的候选解应遍布全部解空间

◆ 方法

在当前状态的邻域结构内以一定概率方式（均匀分布、正态分布、指数分布等）产生

状态接受函数

◆ 原则

- ① 在固定温度下，接受使目标函数下降的候选解的概率要大于使目标函数上升的候选解概率；
- ② 随温度的下降，接受使目标函数上升的解的概率要逐渐减小；
- ③ 当温度趋于零时，只能接受目标函数下降的解。

◆ 方法

具体形式对算法影响不大，一般采用 $\min[1, \exp(-\Delta C/t)]$

初始温度

◆ 收敛性分析

通过理论分析可以得到初温的解析式，但解决实际问题时难以得到精确的参数；

初温应充分大；

◆ 实验表明

初温越大，获得高质量解的机率越大，但花费较多的计算时间；

◆ 方法

- ① 均匀抽样一组状态，以各状态目标值得方差为初温；
- ② 随机产生一组状态，确定两两状态间的最大目标值差，根据差值，利用一定的函数确定初温；
- ③ 利用经验公式。

从理论上说，起始温度 t_0 应保证平稳分布中每一状态的概率相等，即 $\exp\left(-\frac{\Delta f_{ij}}{t_0}\right) \approx 1$

其中 $\Delta f_{ij} = f(j) - f(i)$ 很容易估计一个值为 $t_0 = K\Delta_0$

其中 K 为充分大的数， $\Delta_0 = \max\{f(i) | i \in D\} - \min\{f(i) | i \in D\}$

实际计算中，可选取 $K = 10, 20, 100, \dots$ 的试验值。

对 TSP ，记对应的距离矩阵为 $(d_{ij})_{n \times n}$ 则

$$\Delta_1 = \sum_{i=1}^n \max\{d_{ij} | j \neq i, j = 1, 2, \dots, n\} - \sum_{i=1}^n \min\{d_{ij} | j \neq i, j = 1, 2, \dots, n\}$$

温度更新函数

在 SA 中，**降温的方式对算法有很大影响**。如果**温度下降过快，可能会丢失极值点**；如果温度**下降过慢**，算法的收敛**速度大大降低**。

实际应用中，常采用以下方法：

a 越接近 1
温度下降越慢

a . 每一步温度以相同比率下降，即 $t_{k+1} = a t_k$

其中 $k \geq 0$ 为降温次数， $0 < a < 1$ ， a 为降温系数。

b . 每一步温度以相同长度下降，即 $t_k = t_0(K-k) / K$

其中 t_0 为起始温度， K 为温度下降的总次数。

c . 经典退火方式： $t_k = t_0 / \ln(1+k)$

特点： 温度下降很缓慢，因此算法的收敛速度很慢。

d . 快速退火方式： $t_k = t_0 / (1+ak)$

特点： 在高温区降温比较快，而在低温区降温较慢。

内循环终止准则

- ◆ 非时齐模拟退火算法

每个温度下只产生一个或少量候选解

- ◆ 时齐算法

- (1) 检验目标函数的均值是否稳定;
- (2) 连续若干步的目标值变化较小;
- (3) 按一定的步数抽样。

外循环终止准则

◆ 常用方法

- (1) 设置终止温度的阈值;
- (2) 设置外循环迭代次数;
- (3) 算法搜索到的最优值连续若干步保持不变;
- (4) 概率分析方法。

模拟退火算法的优缺点

◆ 模拟退火算法的优点

质量高；

初值鲁棒性强；

简单、通用、易实现。

◆ 模拟退火算法的缺点

由于要求较高的初始温度、较慢的降温速率、较低的终止温度，以及各温度下足够多次的抽样，因此优化过程较长。

模拟退火算法改进内容

◆ 改进的可行方案

- (1) 设计合适的状态产生函数;
- (2) 设计高效的退火历程;
- (3) 避免状态的迂回搜索;
- (4) 采用并行搜索结构;
- (5) 避免陷入局部极小, 改进对温度的控制方式;
- (6) 选择合适的初始状态;
- (7) 设计合适的算法终止准则。

模拟退火算法改进内容

◆ 改进的方式

- (1) 增加升温或重升温过程，避免陷入局部极小；
- (2) 增加记忆功能（记忆 “**Best so far**” 状态）；
- (3) 增加补充搜索过程（以最优结果为初始解）；
- (4) 对每一当前状态，采用多次搜索策略，以概率接受区域内的最优状态；
- (5) 结合其它搜索机制的算法；
- (6) 上述各方法的综合。

模拟退火算法的一种改进方法

◆ 改进的思路

- (1) 记录 “Best so far” 状态，并即时更新；
- (2) 设置双阈值，使得在**尽量保持最优性**的前提下减少计算量，即在各温度下当前**状态连续 m_1 步**保持不变则认为 Metropolis 抽样稳定，若**连续 m_2 次退温过程中所得最优解不变**则认为算法收敛。

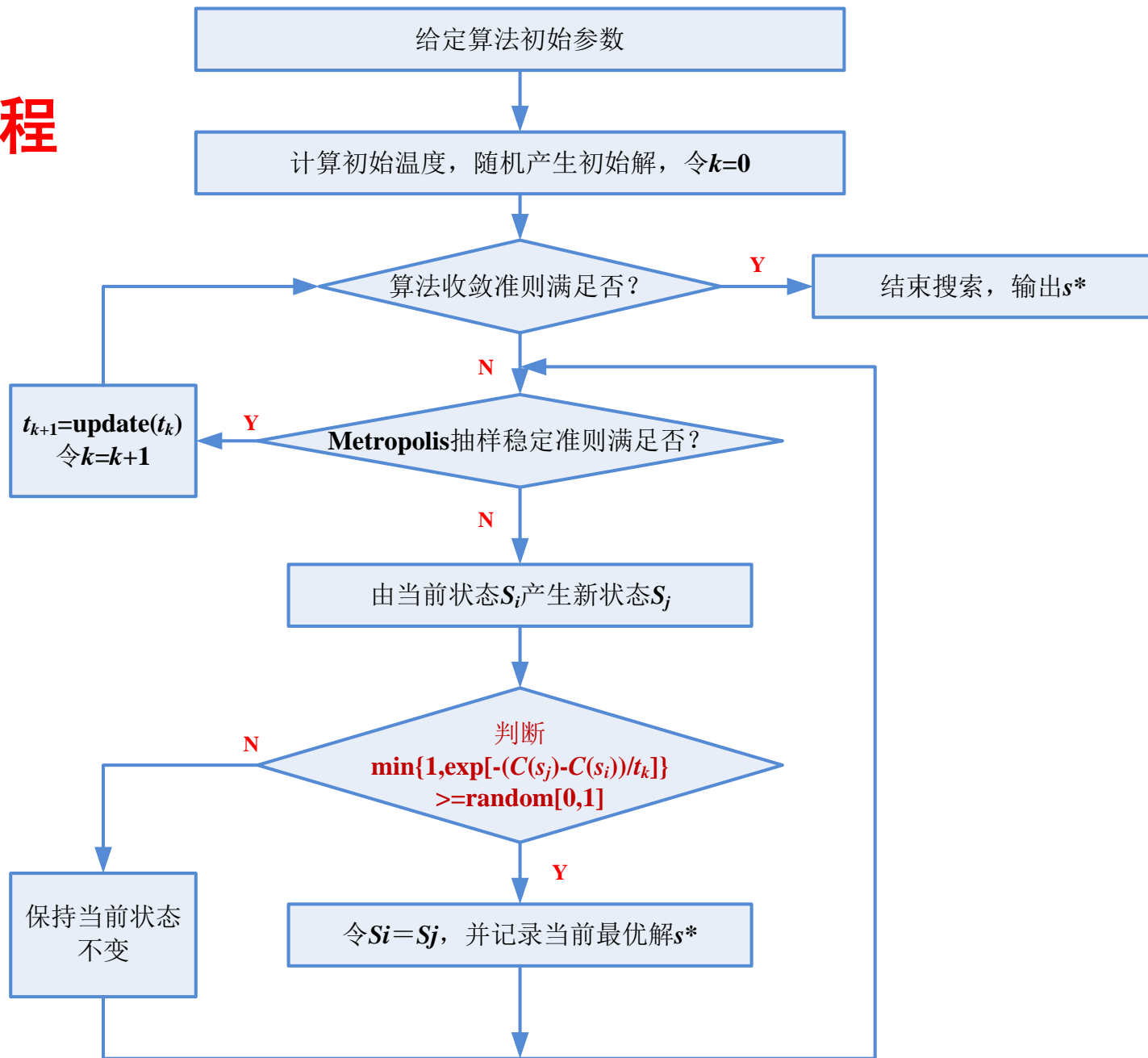
◆ 改进的抽样过程

- ① 令 $k=0$ 时的初始当前状态为 $s'(0)=s(i)$, $q=0$;
- ② 由状态 s 通过状态产生函数产生新状态 s' , 计算增量 $\Delta C' = C(s') - C(s)$;
- ③ 若 $\Delta C' < 0$, 则接受 s' 作为当前解, 并判断 $C(s^*) > C(s')$?
若是, 则令 $s^* = s'$, $q=0$; 否则, 令 $q=q+1$ 。若 $\Delta C' > 0$, 则以概率 $\exp(-\Delta C' / t)$ 接受 s' 作为下一当前状态;
- ④ 令 $k=k+1$, 判断 $q > m_1$? 若是, 则转第(5)步; 否则, 返回第(2)步;
- ⑤ 将当前最优解 s^* 和当前状态 $s'(k)$ 返回改进退火过程。

◆ 改进的退火过程

- ① 给定初温 t_0 ，随机产生初始状态 s ，令初始最优解 $s^*=s$ ，当前状态为 $s(0)=s$ ， $i=p=0$ ；
- ② 令 $t=t_i$ ，以 t ， s^* 和 $s(i)$ 调用改进的抽样过程，返回其所得最优解 $s^{*'}$ 和当前状态 $s'(k)$ ，令当前状态 $s(i)=s'(k)$ ；
- ③ 判断 $C(s^*) < C(s^{*'})$ ？若是，则令 $p=p+1$ ；否则，令 $s^*=s^{*'}$ ， $p=0$ ；
- ④ 退温 $t_{i+1}=\text{update}(t_i)$ ，令 $i=i+1$ ；
- ⑤ 判断 $p > m_2$ ？若是，则转第(6)步；否则，返回第(2)步；
- ⑥ 以最优解 s^* 作为最终解输出，停止算法。

退火算法流程



三、应用案例 —— *TSP*

设完全连通图 G 有 n 个顶点 x_1, x_2, \dots, x_n . 已知任意两点之间的距离为 $d_{ij} = d(x_i, x_j)$, 其中 $1 \leq i, j \leq n$.

解的表现形式: (解空间)

$$S = \left\{ (k_1, k_2, \dots, k_n) \mid (k_1, k_2, \dots, k_n) \text{ 为 } 1, 2, \dots, n \text{ 的一个排列} \right\}$$

$$\text{目标函数: } f(x_{k_1}, x_{k_2}, \dots, x_{k_n}) = \sum_{m=1}^n d_{k_m k_{m+1}}$$

定义 $k_{m+1} = k_1$

SA 算法的求解过程:

(1) 定义相关参数

初始解 s_0 (随机得到); 初始温度 T_0 (开始可以高一些); **Markov** 链的长度 L ; 温度下降方法 $t_{k+1} = a t_k$, a 为降温系数; 终止原则.

(2) 产生新解

新解可通过分别或交替使用以下两种方法来产生:

a、2变换法: 随机产生 1 和 n 之间的 2 个相异数 h 和 m , 不妨设 $h < m$, 则将原路径

$k_1, k_2, \dots, k_h, k_{h+1}, \dots, k_m, k_{m+1}, \dots, k_n$

变为新路径

$(k_1, \dots, k_{h-1}, k_m, k_{m-1}, \dots, k_{h+1}, k_h, k_{m+1}, \dots, k_n)$

b、3变换法：随机产生 1 和 n 之间的 3 个数 h 、 m 和 p ，不妨设 $h \leq m < p$ ，则将原路径中 h 和 m 之间的路径插到 p 之后，得到新路径

$$(k_1, \dots, k_{h-1}, k_{m+1}, \dots, k_{p-1}, k_p, k_h, \dots, k_m, k_{p+1}, \dots, k_n)$$

(3) 计算目标函数差

$$\Delta f_1 = (d_{k_{h-1}k_m} + d_{k_hk_{m+1}} + \sum_{i=h+1}^m d_{k_i k_{i-1}}) - (d_{k_{h-1}k_h} + d_{k_mk_{m+1}} + \sum_{i=h+1}^m d_{k_{i-1}k_i})$$

Δf_2 可类似计算。 若 $d_{ij} = d_{ji}$ ，则

$$\Delta f_1 = (d_{k_{h-1}k_m} + d_{k_hk_{m+1}}) - (d_{k_{h-1}k_h} + d_{k_mk_{m+1}})$$

(4) 接受准则

$$\Delta f < 0 \text{ 或 } \exp(-\Delta f / t_k) \geq \text{random}(0,1)$$

§ 2 遗传算法

遗传算法是20世纪60~70年代主要由美国 *Michigan* 大学 *John Holland* 教授提出。其内涵哲理启迪于自然界生物从低级、简单到高级、复杂，乃至人类这样一个漫长而绝妙的进化过程。借鉴 *Darwin* 的物竞天择、优胜劣汰、适者生存的自然选择和自然遗传的机理。其本质是一种求解问题的**高效并行全局**搜索方法，它能在搜索过程中自动获取和积累有关搜索空间的知识，并自适应地控制搜索过程以求得最优解。

一、基本概念

GA 的基本思想:

从一初始化的群体出发， 通过随机选择(**复制**)（使群体中优秀的个体有更多的机会传给下一代）， **交叉**（体现了自然界中群体内个体之间的信息交换）， 和**变异**（在群体中引入新的变种确保群体中信息的多样性）等遗传操作， 使最**具有生存能力的染色体以最大可能生存**, 群体一代一代地进化到搜索空间中越来越好的区域。

生物遗传概念在遗传算法中的对应关系

生物遗传概念

遗传算法中的作用

适者生存

最优目标值的解有最大可能留住

个 体

解

染 色 体

解的编码（字符串、向量等）

基 因

解中每一分量的特征（如分量的值）

适 应 性

适应函数值

群 体

选定的一组解

种 群

根据适应函数值选取的一组解

交叉（基因重组）

通过交叉原则产生一组新解的过程

变 异

编码的某一个分量发生变化的过程

Example 1 某快餐店要作出以下三项决定：

价格 汉堡包的价格应定在 1 美元还是 50 美分；

饮料 和汉堡包一起供应的应该是酒还是可乐；

服务速度 饭店应提供慢的还是快的服务速度。

Solution : 编码 有 3 个决策变量共有 $2^3 = 8$ 种方案

用三位 0-1 数串，表示一个方案 $(a_1 \ a_2 \ a_3)$ $a_i = 0, 1$

a_1 表示价格 0 —— 高价格 1 —— 低价格

a_2 表示饮料 0 —— 酒 1 —— 可乐

a_3 表示速度 0 —— 慢 1 —— 快

基因

适应函数 即目标函数就取每种方案实行后的利润，
为简单起，每种方案所对应的利润（**适应值**）恰为
这三位二进制所对应的十进制数值。

确定群体规模 N **取 $N=4$** ，在 8 个方案中随机抽取 4
个方案作为初始群体（第 0 代）。

x_i	011	001	110	010
$f(x_i)$	3	1	6	2

总和 12 最小值 1 平均值 3 最大值 6

GA 不是一个个的考虑方案，而是同时**考虑 N 个方案**，体现了并行性。

选择算子 把当前群体中的个体按与适应值成比例的概率 $\left(\frac{f(x_i)}{\sum f(x_i)} \right)$ 复制到新的群体中（适者生存）。

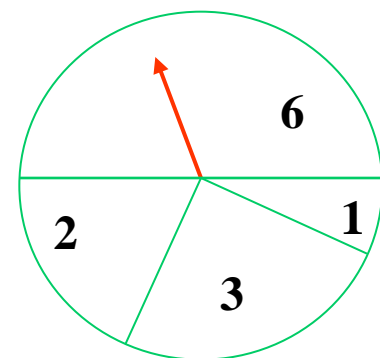
希望 110 在新的群体中出现 $4 \times \frac{6}{12} = 2$ 次, 类似希望 011、010 各出现 1 次, 而 001 消失。因 GA 是随机的, 按概率随机选择, 以上只是期望。

轮盘赌选择:

- 步骤:
- 1、求群体中所有个体的适应值的总和 S ;
 - 2、产生一个在 0 与 S 之间的随机数 m ;
 - 3、从群体中编号为 1 的个体开始, 将其适应值与后继个体的适应值相加, 直到累加和等于或大于 m , 则停止. 其中那个最后加进去的个体即为选择的个体.

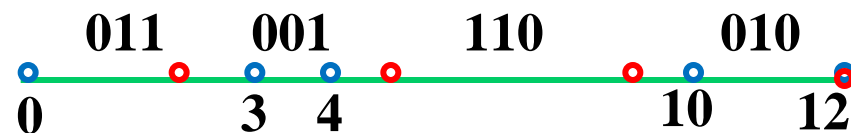
如 随机数 5 2 12 9

选择的个体 110 011 010 110



由选择算子产生的新群体 (可能有重复)

称为种群, 其规模仍为 N .



x_i	110	011	010	110			
$f(x_i)$	6	3	2	6			
总和	17	最小值	2	平均值	4.25	最大值	6

选择算子作用的效果是提高了群体的平均适应值及最差的适应值，低适应值的个体趋于被淘汰，高适应值的个体趋于被复制。但是以损失群体的多样性为代价，选择算子并没有产生新的个体，当然群体中最好个体的适应值不会改进。

交叉（杂交）算子

交叉算子每次作用在种群中随机选取的两个个体上，产生两个不同的子个体，它们一般与父个体不同，但又包含两个父个体的遗传物质。介绍一个最简单的方法。

首先，产生一个 1 到 $l-1$ （其中 l 为染色体分量的个数）的随机数 i （交叉点），然后配对的两个个体相互交换从 $i+1$ 到 l 的位子。如对 x_1, x_2 配对且交叉点选

在 2 ，则

$$\begin{array}{c|c} 11 & 0 \\ 01 & 1 \end{array} \rightarrow \begin{array}{c} 111 \\ 010 \end{array}$$

对种群要确定交叉概率 P_c ，

随机选择 $N \times P_c$ 个个体进行交叉，其余不变。

假设 $P_c = 50\%$

交叉（杂交）算子

x_i	111	010	010	110
$f(x_i)$	7	2	2	6

总和 17 最小值 2 平均值 4.25 最大值 7

显然，利用选择、交叉算子可以产生具有更高平均适应值和更好个体的群体。但如果仅仅如此，很容易近亲繁殖，会早熟（局部最优解）。

变异算子

以一个很小的变异概率 $P_m (= 0.02)$ 随机地改变染色体上的某个基因 $\begin{pmatrix} 0 \rightarrow 1 \\ 1 \rightarrow 0 \end{pmatrix}$ ，具有增加群体多样性的效果。
如：选择 x_3 第 2 位，则 $010 \rightarrow 000$ 得到新的群体。
称为第 1 代，再进行选择、交叉、变异....

Example 2 用 GA 求

Solution : 对连续变量求解, 要解决如何编码问题.

假设对解的误差要求是 $\frac{1}{16}$, 则可采用 4 位二进制编码, 对应关系 $(a \ b \ c \ d) \leftrightarrow \frac{a}{2} + \frac{b}{4} + \frac{c}{8} + \frac{d}{16}$

一次迭代的结果. (交叉概率 $P_c = 1$, 变异概率 $P_m = 0.02$)

x 值	群体 x_i	$f(x_i)$	概率分布	种群	交叉位	交叉结果	变异?	新群体	x 值	$f(x_i)$
1/16	0001	0.99	0.318	0001	00 01	0000	N	0000	0	1
1/4	0100	0.93	0.299	0100	01 00	0101	Y	1101	13/16	0.34
3/16	0011	0.96	0.308	0001	000 1	0001	N	0001	1/16	0.99
7/8	1110	0.23	0.075	0011	001 1	0011	N	0011	3/16	0.96

第 0 代 总和 3.133 最小值 0.234 平均值 0.7833 最大值 0.996;

第 1 代 总和 3.301 最小值 0.340 平均值 0.8253 最大值 1.

遗传算法的步骤：

step 1 选择问题解的一个编码，给出一个有 N 个染色体的初始群体 $pop(1)$, $t = 1$;

step 2 对群体中每一个染色体 $pop_i(t)$ 计算它的适应函数值 $f_i = fitness(pop_i(t))$;

step 3 若停止规则满足，则算法停止，否则计算概率

$$p_i = \frac{f_i}{\sum_{j=1}^N f_j} \quad i = 1 \sim N$$


并以此概率分布，从 $pop(t)$ 中随机选 N 个染色体构成一个种群， $newpop(t)$;

step 4 通过交叉(交叉概率)为 P_c , 得到有 N 个染色体的 $crosspop(t+1)$;

step 5 以较小的概率 (变异概率) P_m 使得某染色体的一个基因发生变异, 形成新的群体 $mutpop(t+1)$

令 $t = t+1$ $pop(t) = nutpop(t)$ go to step 2 .

遗传算法的优越性:

- 1、作为数值求解方法，具有普适性。对目标函数几乎没有要求，总能以极大的概率找到全局最优解；
- 2、**GA** 在求解很多组合优化问题时，不需要很高的技巧和对问题有非常深入的了解，在给问题的决策变量**编码**后，其计算过程是比较简单的，且可较快地得到一个满意解；
这是关键。
- 3、与其他启发式算法有较好的兼容性，易与别的技术相结合，形成性能更优的问题求解方法。

二、实现的技术问题

GA 的主要问题之一是算法如何实现的技术问题，以下的一些技术方法，有的借助于直观，有的则有一定的理论，在此仅给出技术细节。

（一）编码

编码是 GA 中的基础工作之一，GA 不能直接处理解空间的解数据，必须通过编码表成遗传空间的基因型数据。比较直观和常规的方法是 0、1 二进制编码，称为常规码。这种编码方法使算法的三个算子构造比较简单。

如 **0-1** 背包问题的解是一个 **0-1** 向量，可以按 (x_1, x_2, \dots, x_n) 的取值形成一个自然编码。

$$\begin{aligned} \max \quad & z = \sum_{j=1}^n c_j x_j \\ \text{s.t.} \quad & \sum_{j=1}^n a_j x_j \leq b \\ & x_j = 0 \quad \text{or} \quad 1 \end{aligned}$$

➤ 采用 **0-1** 编码可以精确地表示整数。

如 $\max f(x) = x^2 \quad 0 \leq x \leq 31 \quad x \in \mathbb{Z}$

则可用 **5** 位二进制编码。

精确表示 a 到 b 整数的 **0-1** 编码长度 n 满足：

$$\frac{b-a}{2^n} < 1 \quad \text{即} \quad n > \log_2(b-a)$$

➤ 连续变量也可采用二进制编码，但要考虑精度。

对给定的区间 $[a, b]$ ， 设采用二进制编码长为 n ， 则

任何一个变量 $x = a + x_1 \frac{b-a}{2} + x_2 \frac{b-a}{2^2} + \cdots + x_n \frac{b-a}{2^n}$

对应一个二进制码 (x_1, x_2, \dots, x_n) ， 二进制码与实

际变量的最大误差为 $\frac{b-a}{2^n}$ 。

编码长为 n ($n-1$) 所
有解的个数 $2^{n(n-1)}$
可行解的个数 $(n-1)!$

➤ 常规码在表示有些组合优化问题会显得无效或不方便

如 *TSP* 问题 n 个城市二进制编码 $(x_1, x_2, \dots, x_{n(n-1)})$

可行解占有所有解的极小比例，大量的计算浪费，且

对非可行解通过约束限制也增加计算量。

对 **TSP** 的一个自然的表示方法是 n 个城市的序号的排列. $n=10$ 9408712536、3802617945 etc .

理论上而言, 编码应该适合要解决的问题, 而不是简单的描述问题. 编码是一个很值得研究的问题.

(二) 评价 GA 的常用方法

GA 的最优性效果的理论分析是一种趋势讨论, 或无穷代的概率结果. 用一个方法监察计算过程中解的变化趋势, 可以了解进化的程度以便决定是否继续下去. GA 求解最优化问题的一个目的是得到最优解, 于是需要了解 GA 求最优解的功效.

1、当前最好解 (*best-so-far*) 方法

在每一代的进化中，记录下最好的解，通过这个最好解展示算法的效果。这个最好解可以用于不同算法的横向比较；如果知道问题的界，也可以自身比较它的功效。

2、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其计算公式：

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

其中 T 为当前计算群体中出现的染色体总数, $v(t)$ 为第 t 个染色体的目标值。

3、离线 (*off-line*) 比较法

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

其中 T 为当前计算中出现的遗传代数, $v^*(t)$ 为第 t 代前 (包括第 t 代) 时, 所得最优值. 即

$$v^*(t) = \max \{v^*(1), v^*(2), \dots, v^*(t-1), v^*(pop(t))\}$$

$v^*(pop(t))$ 表示 $pop(t)$ 中的最优目标值.

Note (*) 与 (**) 中 t 的区别: 在(*)中 t 表示第 t 个染色体; 而 (**) 中 t 表示第 t 代.

(**) 的另一种理解方法是 $v^*(t)$ 为第 t 代时所得的

最优目标值, 即 $v^*(t) = \max_{i \in pop(t)} \{v(i)\} \quad (***)$

这与 *best-so-far* 方法类似.

从直观可以看出，当优化问题是最大化目标函数时，随着 T 的增加 (**) 应该具有上升趋势，因为适应函数取决于目标函数，随着一代代进化使适应能力强的染色体生存下来，(**) 的平均值上升。

(*) 总体是上升的，但某些代可能会有波动。

通过 (*), (**), (***) 的监控，可以掌握遗传算法计算的进展情况，以决定如何改进？何时停止算法。

(三) 初始参数的选取和停止原则

1、群体的规模

群体规模的确定，根据模式定理，受遗传操作中
选择操作的影响很大。一般来说，群体规模越大，
群体中个体的多样性越高，算法陷入局部解的危险就
越小。

但太大从计算效率着眼，群体越大其适应度评估
次数增加,所以计算量增加，从而影响计算效能；

太小会使 GA 的搜索空间中分布范围有限，因而
搜索有可能停止在未成熟阶段，引起早熟收敛现象。

① 群体的规模取个体编码长度数的一个线性倍数是实际应用时经常采用的方法之一。

如： m 在 n 和 $2n$ 之间的一个数。

② 群体规模的选择也可以变化的。

如：当多个进化代没有改变解的性能，保持现有的群体规模已很难改进解，此时可扩大群体规模；反之，若解的改进非常好，则可以减少群体的规模以便加快计算的速度。

2、初始群体的选取

① 用随机方法产生

只有这样选取才能达到所有状态的遍历，使最优解在 GA 的进化中最终得以生存。

② 用其他优化方法或启发式方法选取 使初始群体更优良。

3、终止规则

- ① 给定一个最大的遗传代数 *Maxgen*
- ② 给定问题的一个上界 *UB*，当进化中达到要求的偏差度时，算法终止；
- ③ 按前面的（*）、（**）、（***）式的评价算法规则，当由它们监控得到算法再进化已无法改进解的性能，则终止计算；
- ④ 以上多种规则的组合。

一般适应函数取正的

目标函数

(四) 进化过程中的技术问题

1、适应函数。

① 简单适应函数

优化目标为最大时, $fitness(x) = f(x)$

优化目标为最小时, $fitness(x) = M - f(x)$ $M > f_{max}(x)$

优点: 简单、与目标函数直接相关。

缺点: 有时使适应度很接近。

如: $z = \max_{x \in [0.5, 1]} f(x) = \max_{x \in [0.5, 1]} (1 + \log x)$

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	0.221
5/8	0100	0.796	0.252
21/32	0101	0.817	0.259
23/32	0111	0.857	0.271

此时，若从群体中以概率分布选取种群，由于概率分布值相差很小，因此，很难区别哪一个染色体占先。

决定当前最优解
的继承性

② 非线性加速适应函数

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

M 一个充分大的数， **f_{\max}** 是当前的最优目标值。

Note **M** 选取的策略：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。

如前例

$$fitness(x) = \begin{cases} \frac{1}{0.857 - f(x)} & f(x) < 0.857 \\ 28 & f(x) \geq 0.857 \end{cases}$$

x	群体	$fitness(x)$	概率分布
1/2	0000	5.95	0.079
5/8	0100	16.39	0.218
21/32	0101	25	0.332
23/32	0111	28	0.372

△ 适应函数的设计应满足的条件:

- ① 单值、连续、非负、最大化 ② 合理、一致性
- ③ 计算量小 ④ 通用性强

△ 遗传算法的欺骗问题:

① 在遗传进化的初期，通常会产生一些超常个体，按比例选择，这些个体竞争力太强，而控制了选择过程，影响算法的全局优化性能；

② 在遗传进化的后期，即算法接近收敛时，由于种群中个体适应度差异较小时，继续优化的潜能降低，可能获得某个局部最优解。

2、选择策略

选择——从群体中选择优胜的个体，淘汰劣质个体，得到种群的操作。

选择算子是遗传算法的关键，体现了自然界中适者生存的思想，基于适应函数值的计算。

① 基于适应函数值比例的策略

计算各个体的相对适应度 $p_i = \frac{f_i}{\sum f_j}$ ，

以 p_i 为选择概率，用论盘赌或 *MonteCarlo* 选择种群。

比 ① 表现出更好的鲁棒性

② 基于排名的策略

不计算各个体的相对适应度，而根据个体适应度在群体中的排名来确定其选择概率。

将同一代群体中的 m 个染色体按适应函数值，从小到大排列，记为 $1, 2, \dots, m$ 。取选择概率

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

再用 ① 的方法选择。

这可避开非线性加速可能产生的早熟现象。

③ 基于局部竞争机制的策略

在群体中**随机选择 k 个个体**（常取 $k = 2$ ）进行比较，其中适应度最好的个体被确定为生成下一代的父体。

④ 最优保存选择策略 **强行遗传**

优点：阶段最优不被随机性破坏；

缺点：影响全局搜索能力。

双亲双子法—
常用方法

3、交叉（杂交）规则

交叉操作是基因算法获得优良个体的重要手段。

内容包括：

- a*、从种群中对个体随机配对，并按预定的交叉概率来决定是否进行交叉操作；
- b*、设定个体的交叉点，并对这些点前后的配对个体的基因相互交换。

① 单交叉位法

父代 A 100|100

父代 B 010|010



子代 A 100|010

子代 B 010|100

② 变化交叉法

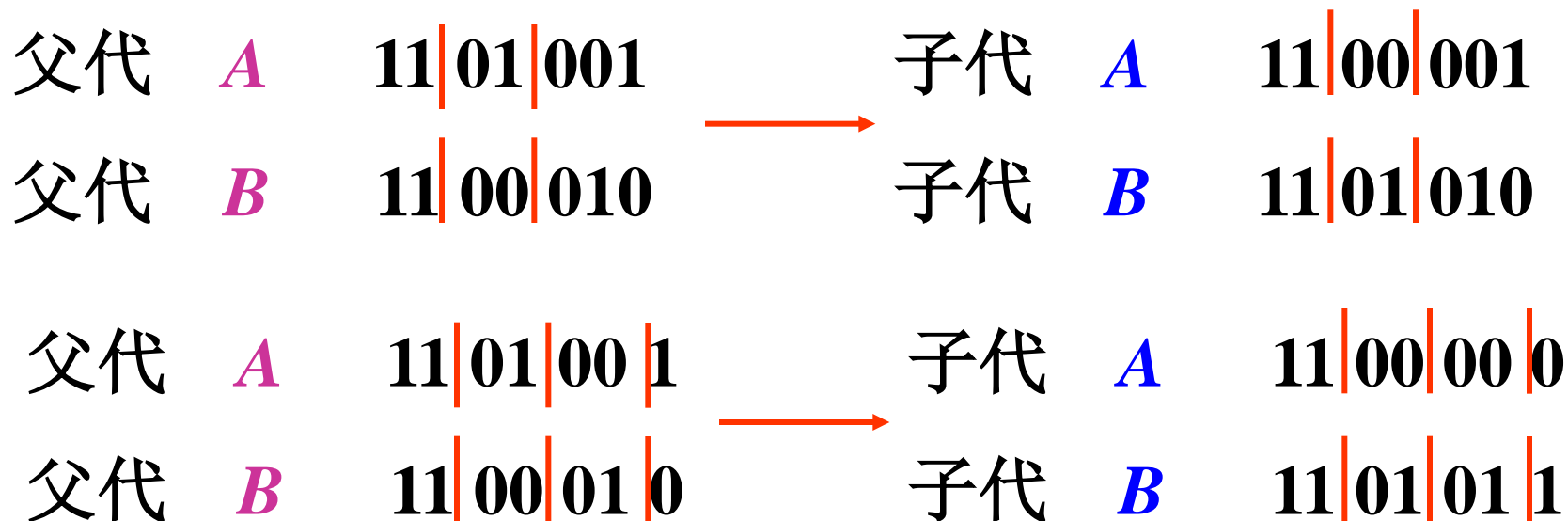
父代	A	1 1 0 1001	→	子代	A	1 1 0 0010
父代	B	1 1 0 0010		子代	B	1 1 0 1001

若交叉点在 1、2、3，则交叉后无任何变化。因此交叉点需避开这三个位置，采用方法：从左开始，先比较它们的相同的基因，从不同基因位置按 ① 进行交叉。

父代	A	1101 001	→	子代	A	1101 010
父代	B	1100 010		子代	B	1100 001

③ 多交叉位法

随机选择多个交叉点，双亲以一个交叉点到下一个交叉点基因相互替代和下一个交叉点到再下一个交叉点不变这样交叉形成两个新的后代。



④ 双亲单子法

- a、从前述方法得到双子后，随机选一个；
- b、从前述方法得到双子后，选一个好的。

⑤ 显性遗传法

对双亲中的基因，有些是具有优越关系的，这些基因必将遗传给下一代。

如：

则 1 永远比 0 好，所以

父代 A 1 0 1 0 1

父代 B 1 0 0 1 0



子代 1 0 1 1 1

⑥ 单亲遗传法

单亲遗传的一个特点是只有一个父代，下一代的产生通过单亲自身的基因变化。

如选定一个单亲，随机选两个基因位置，将两个位置的元素进行交换：

父代 1 1 0 1 0 0 1 \longrightarrow 子代 1 0 0 1 0 1 1

或选定两个交叉位，通过交叉位之内的基因倒排得到子代。

父代 1 \vdots 1 0 1 0 \vdots 0 1 \longrightarrow 子代 1 \vdots 0 1 0 1 \vdots 0 1

⑥ 单亲遗传法

单亲遗传法使得染色体中的基因取值受到限制，如例中的 1 的个数为 4 个，0 的个数为 3 因此限制搜索的范围。在使用单亲遗传时应加大变异的概率。单亲遗传法可以同双亲遗传法结合使用。

⑦ 基于 “与/或” 交叉法

对父代 A 、 B 按位 “与” 逻辑运算产生一子代 A ;
按位 “或” 逻辑运算产生另一子代 B .

父代 A 0100101101

子代 A 0100100100



父代 B 1101110100

子代 B 1101111101

该交叉策略在解背包问题中效果较好 .

⑧ 非常规码的交叉方法 (仅讨论 $1\sim n$ 排列的非常规码)

α 、非常规码的常规交叉法

随机选一个交叉位，两个后代交叉位之前的基因分别继承双亲的交叉位之前基因，交叉位之后的基因分别按异方基因顺序选取不重基因。

如：

父代 A	87213 09546	→	子代 A	87213 95640
父代 B	98356 71420		子代 B	98356 72104

b、非常规码的不变位法

随机产生一个同染色体有相同维数的 0、1 向量。

1 表示不变 0 表示变。 变化方式按方法 *a*。

如： (1 0 0 1 1 0 0)

父代 <i>A</i>	2 1 3 7 6 5 4	→	子代 <i>A</i>	2 3 1 7 6 4 5
父代 <i>B</i>	4 3 1 2 5 6 7		子代 <i>B</i>	4 1 3 2 5 7 6

c、部分匹配交叉 (PMX) 法

先随机产生两个交叉点，定义这两点间的区域为匹配区域，并用位置交换操作交换两个父代的匹配区域。如：交叉点为 3、6

父代 A	872	130	9546	→	A'	872	567	9546
父代 B	983	567	1420		B'	983	130	1420

对于 A'、B' 中匹配区域以外出现的数码重复，要依据匹配区域内的位置逐一进行交换。

5 → 1, 6 → 3, 7 → 0

子代 A	8025679143	子代 B	9861305427
------	------------	------	------------

d、循环交叉 (*CX*) 法

与 *PMX* 法不同, *CX* 法以父亲特征为参考, 按一定的规则进行基因重组.

从左开始, 对应的原位不变, 形成循环, 余下的按另一父代次序.

父代 *A* 9 8 2 1 7 4 5 0 6 3 子代 *A* **9 2 3 1 5 4 7 8 6 0**

父代 *B* 1 2 3 4 5 6 7 8 9 0 子代 *B* **1 8 2 4 7 6 5 0 9 3**

e、顺序交叉法

父代 A 1 2 3 | 4 5 6 7 | 8 9

父代 B 4 5 2 | 1 8 7 6 | 9 3

将父代 B 从第2个交叉位排列为 B' : 9 3 4 5 2 1 8 7 6 .

对父代 A 保留中间 4 5 6 7, 将它们从父代 B' 中删去, 得 9 3 2 1 8, 再按从第2个交叉位开始得

子代 A 2 1 8 4 5 6 7 9 3

同理可得 子代 B 3 4 5 1 8 7 6 9 2

4、变异算子

变异算子能使个体发生突然变异，**导入新的遗传信息**，**使寻优有可能指向未探知区域**，是提高全局最优搜索能力的有效步骤，也是保持群体差异、防止过早出现收敛现象的重要手段。

在 GA 中，交叉和变异这对相互配合又相互竞争的操作，而使其具备兼顾全局和局部的均衡搜索能力。

所谓相互配合是指当群体在进化中陷入某个局部而仅靠交叉不能摆脱时，通过变异有助于这种摆脱；所谓相互竞争是指当通过交叉已形成所期望的群体，变异有可能破坏这个群体。

变异概率 P_m 不能太小，这样降低全局搜索能力；也不能太大， P_m 大于 0.5，这时 GA 退化为随机搜索。

② 逆转变异算子

如 个体 A **1011011** **1110011** 个体 B

基本操作：在个体中随机挑选两个逆转点，再将两个逆转点间的基因反序插入原位置。

如 个体 A 10 **11010** 00 10 **01011** 00 个体 B

个体 A 982 ^o174^o5 063 9825471063 个体 B

③ 插入变异算子

基本操作：从个体中随机选择一个码（基因），将此码插入随机选择的插入点中间。

如 个体 A 11 0^o100110 \rightarrow 1100100110 个体 B

个体 A 98 2^o1745063 \rightarrow 9842175063 个体 B

④ 对换变异算子

随机选 3、7 为对换点

随机选择个体中的两个点，并交换其基因。

如 个体 A 982^o1745063 9851742063 个体 B

个体 A 1101001101 1111000101 个体 B

通过选择、交叉、变异得到新的群体其替换率是值得研究的。主要涉及交叉概率 P_c 和变异概率 P_m 的选择，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，但优点是使某些关心的染色体得以保留。100% 的替换率也并不是最好的，会使当前最优解遗失。一种策略是使上一代的当前最优解强行遗传到下一代。以上方法如何最佳搭配，因问题而言。“无免费午餐”。

自适应遗传算法

遗传算法开创了在解空间中从多点出发搜索问题最优解的先河，可应用于组合优化、机器学习、智能控制、人工生命、图象处理、模式识别等领域。由于 GA 的结构是开放的，与问题无关，很容易与其他算法综合运用。

遗传算法的求解特点可以概括成如下的四点：

- 1、利用变量的编码方式，而不使用变量本身；
- 2、在解空间中，从多个点出发搜索问题的解，而不是像某些传统的搜索方法从一个点出发搜索问题的解；
- 3、直接利用目标函数值的信息，而不使用函数的导数或其他辅助信息；
- 4、使用概率转移规则，而不是采用确定性的转移规则。

三、 在组合优化中的应用

(一) 背包问题 (*Knapsack Problem*)

$$\max \quad z = \sum_{j=1}^n p_j x_j$$

$$s.t. \quad \sum_{j=1}^n w_j x_j \leq C$$

$$x_j = 0 \quad or \quad 1 \quad j \in N = \{1, 2, \dots, n\}$$

设

$$\frac{p_1}{w_1} \geq \frac{p_2}{w_2} \geq \dots \geq \frac{p_n}{w_n}$$

1、用一般编码方法的遗传算法

a、编码

最常用的是用 n 个二进制编码表示一个个体的遗传基因。如 8 个变量的 KP ，则 $(1, 0, 0, 1, 1, 0, 0, 1)$ 表示该个体对应选择了物品 1、4、5、8。

b、适应度函数（罚函数）

为处理约束条件，适应度函数的计算用

$$f_i = \begin{cases} \sum_{j=1}^n p_j x_j & \sum_{j=1}^n w_j x_j \leq C \\ 0 & \sum_{j=1}^n w_j x_j > C \end{cases}$$

群体规模为 $m = n$

c、3 个基本算子

- * 正比于适应度的轮盘赌随机选择方式;
- * 单交叉位法, 交叉概率为 $P_c = 0.8$;
- * 基本位变异法, 变异概率为 $P_m = 0.001$.

Note : 用罚函数处理约束条件, 当问题规模很大时, 搜索的效率很低. 可结合贪婪算法 (称为**混合遗传算法**) 对那些不满足约束条件的个体, 优先装入价值密度较大且编码值为 1 的物品, 直至背包容量限制装不下为止, 并将未装入的物品的编码值修正为 0 , 形成个体新的染色体编码. 这一运算可在选择、交叉、变异各环节前后.

2、二重结构编码的遗传算法

个体的染色体的二重结构编码由变量码和附加码两行组成。上行 $s(i)$ 为变量 x_j 的附加码 $s(i) = j$ ，下行为变量 $x_{s(i)}$ 的取值。

附加码	$s(1)$	$s(2)$	\dots	$s(i)$	\dots	$s(n)$
变量码	$x_{s(1)}$	$x_{s(2)}$	\dots	$x_{s(i)}$	\dots	$x_{s(n)}$

对某个个体编码时, 首先随机产生附加码 $s(i)$ $i=1\sim n$ 。再随机产生变量码值(0 or 1)，这为个体的二重结构码。

为考虑约束条件，进行解码。从左到右按顺序考虑附加码为 $s(i)$ 的物品，若 $x_{s(i)} = 0$ ，则 $p_{s(i)} = 0$ ；
若 $x_{s(i)} = 1$ 违背约束条件，则 $p_{s(i)} = 0$ ，否则 $p_{s(i)} = 1$ 。
直到所有物品处理完。

附加码	$s(1)$	$s(2)$	$s(3)$	$s(4)$	$s(5)$	$s(6)$	$s(7)$
变量解码值	$p_{s(1)}$	$p_{s(2)}$	$p_{s(3)}$	$p_{s(4)}$	$p_{s(5)}$	$p_{s(6)}$	$p_{s(7)}$

附加码	4	3	8	1	6	2	5	7
变量解码值	1	1	0	1	1	0	0	0

用部分匹配交叉 (PMX) 算子进行交叉操作

Note: PMX 操作只是针对个体的附加码, **变量码值** 仍根据其父代个体中附加码与变量码的关系来确定.

A	5	7	2	8	1	6	4	3
	1	0	0	1	1	0	1	0
B	7	1	6	2	4	8	3	5
	1	1	0	1	0	0	1	1

A'	5	7	6	2	4	8	1	3
	1	0	0	0	1	1	1	0
B'	7	4	2	8	1	6	3	5
	1	0	1	0	1	0	1	1

用逆转变异算子进行变异操作，对父代个体**随机选择两个变异点**，两点间的附加码按相反顺序重新排列，而变量码顺序不变。

7	2	5	8	4	1	6	3
1	0	0	1	1	0	1	0

→

7	2	6	1	4	8	5	3
1	0	0	1	1	0	1	0

基于“与 / 或”交叉方法求解背包问题

交叉算子:

由父代 A 、 B 按位进行“与”逻辑运算产生一子代 A' ;

由父代 A 、 B 按位进行“或”逻辑运算产生一子代 B' .

如 父代

A	0100101101	→	0100100100	A'	子代
B	1101110100				

如 父代

A	0100101101	→	1101111101	B'	子代
B	1101110100				

显然，这一交叉方法使子代继承了双亲的同型基因,对于双亲的异型基因，“与 / 或”交叉方法采取了两种不同的支配方式，“与”运算是0支配1；而“或”运算是1支配0的方式。

模式定理

◆ 模式

将种群中的个体即基因串中的相似样板称为模式。

在二进制编码的串中，模式是基于三个字符集（0，1，*）的字符串，符号*称作通配符，代表任意字符，即0或1。

如模式 *1* 描述了一个四个元的子集{010，011，110，111}。

◆ 模式阶

模式 H 中确定位置的个数（即 H 中0或1的个数）称为模式 H 的模式阶，记作 $O(H)$ ，如

$$O(0\ 1\ 1\ * \ 1\ *) = 4$$

模式阶用来反映不同模式间确定性的差异，模式阶越高，模式的确定性就越高，所匹配的样本个数就越少。

◆ 定义距

模式 H 中第一个确定位置和最后一个确定位置之间的距离称为模式的定义距，记作 $\delta(H)$ ，如

$$\delta(0 \ 1 \ 1 \ * \ 1 \ * \ *) = 5 - 1 = 4。$$

阶数相同的模式会有不同的性质，定义距就反映了这种性质的差异。

◆ 平均适应度

模式 H 的平均适应度是指在一个确定的群体内，与 H 相匹配的所有个体之平均适应度，记做 $f(H)$ 。

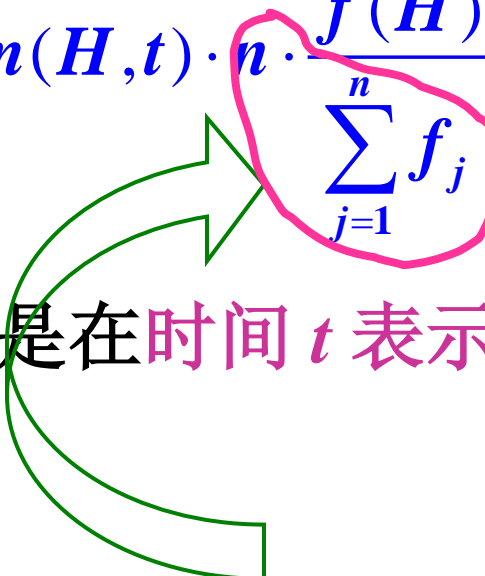
◆ 再生概率

在给定时间步 t , 一个特定模式 H 有 m 个代表串包含在种群 $A(t)$ 中, 记为 $m=m(H,t)$, 在再生阶段, 每个串根据它的适应值 f_i 进行复制, 一个串 A_i 的再生概率为

$$p_i = \frac{f_i}{\sum_{j=1}^n f_j}$$

◆ 模式定理 (schema theorem)

当采用非重叠的 n 个串的种群替代种群 $A(t)$, 可以得到:

$$m(H, t+1) = m(H, t) \cdot n \cdot \frac{f(H)}{\sum_{j=1}^n f_j} \Rightarrow m(H, t+1) = m(H, t) \frac{f(H)}{\bar{f}}$$


其中 $f(H)$ 是在时间 t 表示模式 H 的串的平均适应度。

整个种群的平均适应度为 $\bar{f} = \frac{\sum_{j=1}^n f_j}{n}$

◆ 模式选择生长方程

假设从 $t=0$ 开始，某一特定模式适应度值保持在种群平均适应度以上一个 $c\bar{f}$ （ c 为一常数），则模式的选择生长方程为

$$m(H, t+1) = m(H, t) \frac{(\bar{f} + c\bar{f})}{\bar{f}} = (1+c) \cdot m(H, t) = (1+c)^t \cdot m(H, 0)$$

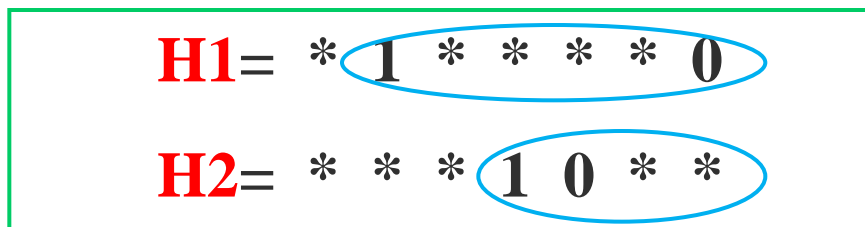
◆ 生存概率计算模型

考虑交叉操作，模式 H 被破坏的概率为 $\delta(H)/(l-1)$ ，模式 H 生存概率为 $1 - \delta(H)/(l-1)$ ，若交叉操作发生的概率为 p_c ，因此对于模式 H 的生存概率计算为：

$$p_s \geq 1 - p_c \frac{\delta(H)}{l-1}$$

同时考虑选择、交叉操作对模式的影响，可得：

$$m(H, t+1) \geq m(H, t) \cdot \frac{f(H)}{\bar{f}} \left[1 - p_c \frac{\delta(H)}{l-1} \right]$$



◆ 模式定理 (schema theorem)

考虑变异操作，单个等位基因存活的概率为 p_m ，当模式 H 中 $O(H)$ 个确定位都存活时，模式 H 才被保留，存活概率为：

$$(1 - p_m)^{O(H)} \approx 1 - O(H) \cdot p_m \quad (p_m \ll 1)$$

同时考虑选择、交叉和变异操作对模式的影响，可得：

$$m(H, t+1) \geq m(H, t) \cdot \frac{f(H)}{\bar{f}} \left[1 - p_c \frac{\delta(H)}{l-1} - O(H)p_m \right]$$

H1= * 1 * * * * 0

H2= * * * 1 0 * 0

本章结束