

第四章 模拟退火

一. 导言

0. 知识回顾

➤ 组合优化问题

- ✓ “最优化问题似乎自然地分成两类：一类是连续变量的问题，另一类是离散变量的问题。具有离散变量的问题，我们称它为组合的。”——来自百度百科
- ✓ “A combinatorial optimization problem can be modeled as that of minimizing a cost function or maximizing a profit function over a finite set of discrete variables.”

一. 导言

0. 知识回顾

➤ 解空间

对于一个组合优化问题来说，**解空间**定义为：任一可能（可行）**解可以访问的搜索空间**

举例：对于 n 个城市的旅行商问题来说，一个解可以用自然数序列 $\{1, 2, \dots, n\}$ 的某种排列来表达，于是解空间则是由该自然数序列的所有排列组合构成。

$\{1, 2, 3, 4\}$ 可以表示4个城市旅行商问题的一个解

一. 导言

0. 知识回顾

➤ 邻域

令 S 表示组合优化问题的一个解，则其邻域 $N(S)$ 可以定义为： S 执行一次邻域移动可以访问的所有解的集合

举例：对于4个城市的旅行商问题来说，采用上述的基于排列的解表达方法，邻域移动定义为 *2-opt*，即对 S 中2个元素进行对换

$S=(1,2,3,4)$ ，则邻域 $N(S)=\{(2,1,3,4), (3,2,1,4), (4,2,3,1), (1,3,2,4), (1,4,3,2), (1,2,4,3)\}$

一. 导言

0. 知识回顾

- 局域搜索：从一个初始解开始，每次迭代总是遍历其邻域，若不存在比当前最优解更好的邻域解，则返回当前最好解，算法停止；否则用最好的邻域解替换当前解。如此循环直至算法停止。类似于模拟爬山的过程，因此也被称之为爬山算法。

一. 导言

1. 基本思想

- 原始算法是由Metropolis等(1953)提出，但未引起反响；1983年Kirkpatrick等将其应用于组合优化，才得到广泛的应用
- 目的是为了克服优化过程中陷入局优和初值依赖等弊端
- 基本思想是模拟热力学中的退火过程
- **Simulated Annealing**，简称SA

一. 导言

2. 金属退火过程

➤ 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



一. 导言

2. 金属退火过程

➤ 什么是退火

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**

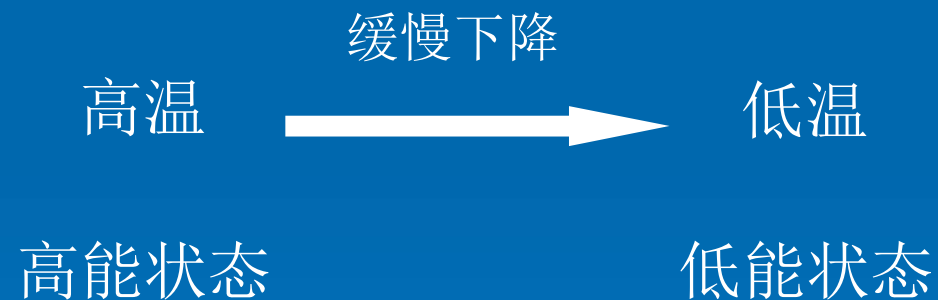
等温过程——对于与环境换热而温度不变的封闭系统，系统状态的自发变化总是朝自由能减少的方向进行，当自由能达到最小时，系统**达到平衡态**

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**

一. 导言

2. 金属退火过程

➤ 什么是退火



一. 导言

2. 金属退火过程

➤ 数学描述

设热力学系统 S 中有 n 个状态（有限且离散的），其中状态 i 的能量为 E_i ，在温度 T_k 下，经一段时间达到热平衡，此时处在状态 i 的概率为：

$$P_i(T_k) = C_k \exp\left(-\frac{E_i}{T_k}\right)$$

则有如下关系： $E_i \downarrow \rightarrow P_i \uparrow$ ， $T_k \downarrow \rightarrow P_i \downarrow$

如何确定 C_k ？

一. 导言

2. 金属退火过程

➤ 数学描述

$$\sum_{j=1}^n P_j(T_k) = 1 \Rightarrow \sum_{j=1}^n C_k \cdot \exp\left(\frac{-E_j}{T_k}\right) = 1$$

$$\Rightarrow C_k \cdot \sum_{j=1}^n \exp\left(\frac{-E_j}{T_k}\right) = 1 \Rightarrow C_k = \frac{1}{\sum_{j=1}^n \exp\left(\frac{-E_j}{T_k}\right)}$$

一. 导言

2. 金属退火过程

➤ 数学描述

通过求 C_k ，获得Bolzman方程：

$$P_i(T_k) = \frac{\exp\left(-E_i/T_k\right)}{\sum_{j=1}^n \exp\left(-E_j/T_k\right)}$$

一. 导言

2. 金属退火过程

➤ Boltzman方程

①同一温度下 S 处在能量小的状态要比处在能量大的状态概率大

若存在 $E_1 < E_2$ ，则在同一温度 T_k 下，有

$$\frac{P_1(T_k)}{P_2(T_k)} = \frac{C_k \exp\left(\frac{-E_1}{T_k}\right)}{C_k \exp\left(\frac{-E_2}{T_k}\right)} = \exp\left(\frac{E_2 - E_1}{T_k}\right) > 1$$

故 $P_1(T_k) > P_2(T_k)$

一. 导言

2. 金属退火过程

➤ Boltzman方程

②若 i^* 表示 S 中唯一的最低能量的状态, $P_{i^*}(T_k)$ 是关于温度 T_k 单调递减

对 $P_i(T_k)$ 求对温度的导数, 则

$$\frac{\partial P_i(T_k)}{\partial T_k} = \frac{\exp(\frac{-E_i}{T_k})}{T_k^2 \cdot \left[\sum_{j=1}^n \exp(\frac{-E_j}{T_k}) \right]^2} \left[\sum_{j=1}^n (E_i - E_j) \cdot \exp(\frac{-E_j}{T_k}) \right]$$

一. 导言

2. 金属退火过程

➤ Boltzman方程

由于 $\exp(\frac{-E_i}{T_k}) > 0, T_k^2 \cdot \left[\sum_{j=1}^n \exp(\frac{-E_j}{T_k}) \right]^2 > 0, \exp(\frac{-E_j}{T_k}) > 0$

且存在当 $j \neq i^*$ 时, $E_{i^*} - E_j < 0$

故 $\frac{\partial P_{i^*}(T_k)}{\partial T_k} < 0$

一. 导言

2. 金属退火过程

➤ Boltzman方程

③当 $T_k \rightarrow 0$ 时, $P_{i^*}(T_k) \rightarrow 1$

$$P_{i^*}(T_k) = \frac{\exp\left(\frac{-E_{i^*}}{T_k}\right)}{\sum_{j=1}^n \exp\left(\frac{-E_j}{T_k}\right)} = \frac{1}{\sum_{j=1}^n \exp\left(\frac{-(E_j - E_{i^*})}{T_k}\right)}$$

同理, 当 $T_k \rightarrow 0$ 时, $\bar{E}(T_k) \rightarrow E_{i^*}$

一. 导言

2. 金属退火过程

➤ 温度 T_k 对 $P_i(T_k)$ 的影响

①当 T_k 很大时, $E_i/T_k \rightarrow 0$, $P_i(T_k) \approx 1/n$, 这意味着各状态的概率几乎相等。

②当 $T_k \rightarrow 0$ 时, $E_i/T_k \rightarrow \infty$, 这意味着 E_i 与 E_j 的小差别带来 $P_i(T_k)$ 和 $P_j(T_k)$ 的巨大差别

一. 导言

2. 金属退火过程

➤ 温度 T_k 对 $P_i(T_k)$ 的影响

例: $E_i=90$, $E_j=100$

当 $T_k=100$ 时

$$\frac{P_i(T_k)}{P_j(T_k)} = \frac{C_k \cdot e^{-\frac{90}{100}}}{C_k \cdot e^{-\frac{100}{100}}} = \frac{0.406 \cdot C_k}{0.367 \cdot C_k} = \frac{0.406}{0.367} \approx 1$$

此时 $P_i(T_k) \approx P_j(T_k)$

一. 导言

2. 金属退火过程

➤ 温度 T_k 对 $P_i(T_k)$ 的影响

例: $E_i=90$, $E_j=100$

当 $T_k=1$ 时

$$\frac{P_i(T_k)}{P_j(T_k)} = \frac{8.194 \times 10^{-40} C_k}{3.72 \times 10^{-44} C_k} \approx 20000$$

此时 $P_i(T_k)+P_j(T_k) \approx P_i(T_k)$

一. 导言

3. 组合优化与退火

➤ Metropolis准则—以概率接受新状态

若在温度 T_k ，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 $p = \exp[-(E_j - E_i) / T_k]$ 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

一. 导言

3. 组合优化与退火

组合优化问题	物理退火
解	状态
目标函数	能量函数
最优解	最低能量状态
设定初始高温	加热过程
基于Metropolis准则的搜索	等温过程
温度参数的下降	冷却过程

二. 基本算法

1. 构成要素

➤ 解的表达与邻域移动方式同TS

➤ 邻域解的产生

在当前解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

➤ 选择策略

一般采用 $\min[1, \exp(-\Delta f/t)]$

二. 基本算法

1. 构成要素

➤ 初始温度

- ① 均匀抽样一组状态，以各状态目标值的方差为初温
- 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

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

二. 基本算法

1. 构成要素

➤ 降温函数

① $T_{k+1} = T_k \cdot r$, 其中 $r \in (0.95 - 0.99)$

优点: 简单易行

② $T_{k+1} = T_k - \Delta T$

二. 基本算法

1. 构成要素

➤ 热平衡条件

Metropolis抽样稳定准则

- ① 设置内循环迭代次数
- ② 检验目标函数的均值是否稳定
- ③ 连续若干步的目标值变化较小

➤ 停止准则

- ① 设置外循环迭代次数
- ② 设置终止温度的阈值
- ③ 算法搜索到的最优值连续若干步保持不变

二. 基本算法

2. 算法流程

Step 1

选一个初始点 i ($i \in S$), 给定初始温度 T_0 , 终止温度 T_f , 令迭代指标 $k = 0$, $T_k = T_0$;

注: 选择 T_0 时, 要足够高, 使 $E_i/T_k \rightarrow 0$

Step 2

随机产生一个邻域解 $j \in N(i)$, 计算目标值增量 $\Delta f = f(j) - f(i)$;

二. 基本算法

2. 算法流程

Step 3

若 $\Delta f < 0$, 令 $i=j$; (j 比 i 好, 无条件转移)

否则产生 $\xi \in U(0,1)$, 若 $\exp(-\Delta f / T_k) > \xi$,
则令 $i=j$; (i 比 j 好, 有条件转移)

注: T_k 高时, 广域搜索; T_k 低时, 局域搜索

Step 4

若达到热平衡 (内循环停止准则) , 转Step 5;
否则, 转Step 2;

二. 基本算法

2. 算法流程

Step 5

若 $k=k+1$ ，更新 T_k ，若 $T_k < T_f$ （外循环停止准则），则停止；否则，转Step 2。

二. 基本算法

3. 算法伪码图

```
Generate an initial solution  $s$ ;  
Initialize the temperature  $T$ ;  
while the stop condition is not met do  
  repeat  
    Select  $s' \in N(s)$  randomly;  
    if  $f(s') \leq f(s)$  then  
       $s \leftarrow s'$ ;  
    else  
       $s \leftarrow s'$  with a probability  $p(T, f(s'), f(s))$ ;  
    endif  
  until the “thermodynamic equilibrium” condition is met  
  Decrease  $T$ ;  
endwhile
```

三. 算法举例

1. 问题提出

单机极小化总流水时间的排序问题
四个工作：

$$P_1 = 8, P_2 = 18, P_3 = 5, P_4 = 15$$

求使 $\min \sum_{i=1}^4 F_i$ 的最优顺序

三. 算法举例

1. 问题提出

预备知识：按SPT准则，最优顺序为3-1-4-2

$$F_{[1]} = P_{[1]}$$

$$F_{[2]} = P_{[1]} + P_{[2]}$$

$$F_{[3]} = P_{[1]} + P_{[2]} + P_{[3]}$$

$$F_{[4]} = P_{[1]} + P_{[2]} + P_{[3]} + P_{[4]}$$

三. 算法举例

1. 问题提出

预备知识：按SPT准则，最优顺序为3-1-4-2

$$\begin{aligned} F &= [P_{[1]} \quad P_{[2]} \quad P_{[3]} \quad P_{[4]}] \begin{bmatrix} 4 \\ 3 \\ 2 \\ 1 \end{bmatrix} \\ &= 4 \times P_{[1]} + 3 \times P_{[2]} + 2 \times P_{[3]} + 1 \times P_{[4]} \\ &= 4 \times 5 + 3 \times 8 + 2 \times 16 + 1 \times 18 \\ &= 92 \end{aligned}$$

三. 算法举例

2. 算法设计

- 编码方式：顺序编码
- 初始解的产生：随机产生，如1-4-2-3
- 邻域移动方式：2-OPT，即两两交换
- 初始温度： $T_0 = 100$
- 终止温度： $T_f = 60$
- 降温函数： $T_{k+1} = T_k - 20$
- 内循环最大迭代次数： $n(T_k) = 3$

三. 算法举例

(1) 当前解: $i=1-4-2-3$, $f(i) = 118$, $T_k = 100$

内循环次数	i	j	$f(j)$	Δf	$\exp\left(-\frac{\Delta f}{T_k}\right)$	ξ	是否执行
①	1-4-2-3	1-3-2-4	98	-20	null	null	$i \leftarrow j$
②	1-3-2-4	4-3-2-1	119	21	0.8106	0.7414	$i \leftarrow j$
③	4-3-2-1	4-2-3-1	132	13	0.8781	0.3991	$i \leftarrow j$

注释:

- j 从 i 邻域内随机产生, ξ 是随机产生的 $U(0,1)$;
- ①为无条件转移;
- ②③为有条件转移;
- 在②③中, 虽然目标值变坏, 但搜索范围变大

三. 算法举例

(2) 当前解: $i=4-2-3-1$, $f(i) = 132$, $T_k = 80$

内循环次数	i	j	$f(j)$	Δf	$\exp\left(-\frac{\Delta f}{T_k}\right)$	ξ	是否执行
①	4-2-3-1	4-2-1-3	135	3	0.9632	0.3413	$i \leftarrow j$
②	4-2-1-3	4-3-1-2	109	-26	null	null	$i \leftarrow j$
③	4-3-1-2	4-3-2-1	119	10	0.8825	0.9286	$i \leftarrow i$

注释:

- 当前温度降低至80;
- ①为有条件转移;
- ②为无条件转移;
- 在③中, 停在4-3-1-2状态, 目标值仍为109

三. 算法举例

(3) 当前解: $i=4-3-1-2$, $f(i) = 109$, $T_k = 60$

内循环次数	i	j	$f(j)$	Δf	$\exp\left(-\frac{\Delta f}{T_k}\right)$	ξ	是否执行
①	4-3-1-2	1-3-4-2	95	-24	null	null	$i \leftarrow j$
②	1-3-4-2	3-1-4-2	92	-3	null	null	$i \leftarrow j$
③	3-1-4-2	2-1-4-3	131	39	0.5220	0.7105	$i \leftarrow i$

注释:

- 当前温度降低至60;
- ① ②为无条件转移;
- 在③中, 停在3-1-4-2状态, 目标值仍为92

三. 算法举例

若继续降温，则当前温度低于终止温度，故算法停止。

上述算法所获得的最终解为：**3-1-4-2**，目标值为：**92**。

注意：上述算法可能不会终止在最优解，故一般SA算法一定要保持历史最优

四. 算法应用实例

1. 基础数据

➤ 全国31个省会城市的坐标数据如下：

$C = [1304\ 2312; 3639\ 1315; 4177\ 2244; 3712\ 1399; 3488\ 1535; 3326\ 1556;$
 $3238\ 1229; 4196\ 1044; 4312\ 790; 4386\ 570; 3007\ 1970; 2562\ 1756;$
 $2788\ 1491; 2381\ 1676; 1332\ 695; 3715\ 1678; 3918\ 2179; 4061\ 2370;$
 $3780\ 2212; 3676\ 2578; 4029\ 2838; 4263\ 2931; 3429\ 1908; 3507\ 2376;$
 $3394\ 2643; 3439\ 3201; 2935\ 3240; 3140\ 3550; 2545\ 2357; 2778\ 2826;$
 $2370\ 2975]$

四. 算法应用实例

2. SA算法设置

- 顺序编码，编码长度 $N=31$
- 初始解随机产生
- 邻域移动方式为2-OPT
- 初始温度 $T=100*N$ ，内循环次数 $L=100$
- 衰减参数 $K=0.99$

五. 算法应用实例

3. Matlab代码

- 参见附件文档

编程作业

作业要求：

- 根据参考代码，调试程序。
- 分析内循环次数和衰减参数对算法性能的影响。
- 分析初始解对算法性能的影响。

提示：

1. 目前代码中内循环次数设为100，尝试将其分别设置为50、150和200时，分析算法性能会发生什么样的变化？
2. 目前代码中衰减参数设为0.99，尝试将其分别设置为0.9、0.8和0.5时，分析算法性能会发生什么样的变化？
3. 当前代码中初始解随机产生，尝试利用贪婪启发式产生初始解，分析是否有助于改善算法性能？