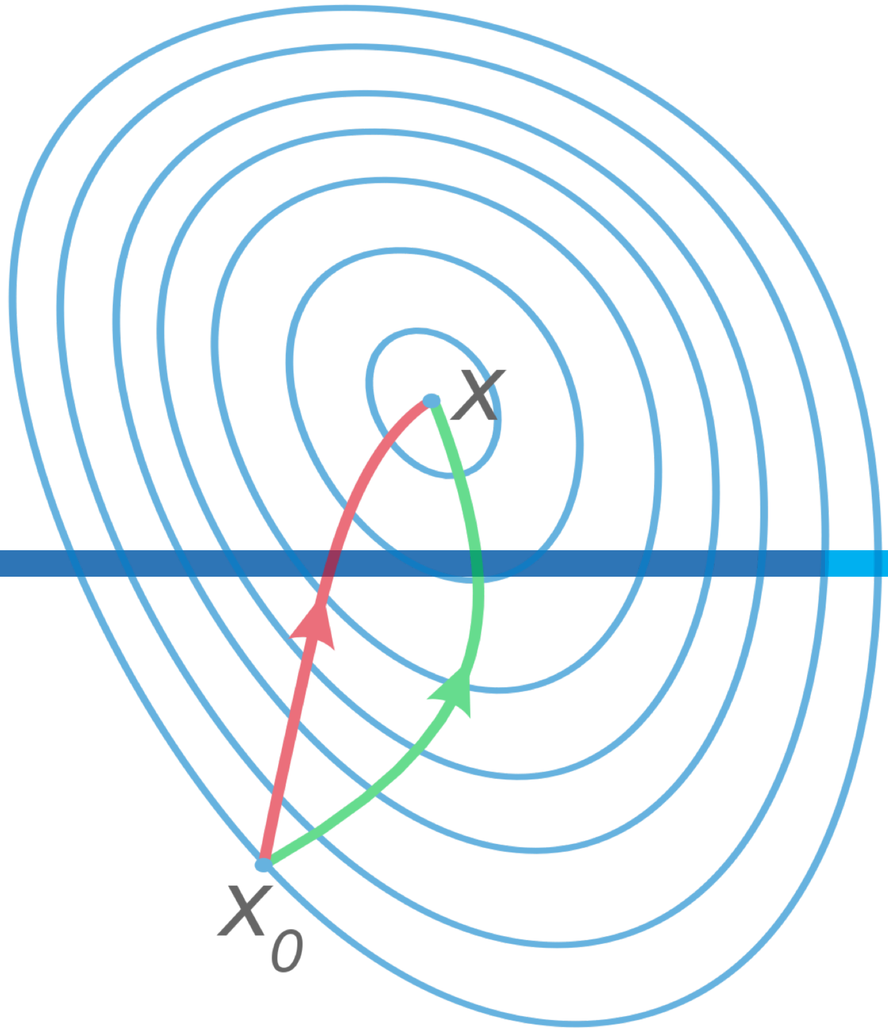
最优化方法

第五周



计算机学院
余皓然

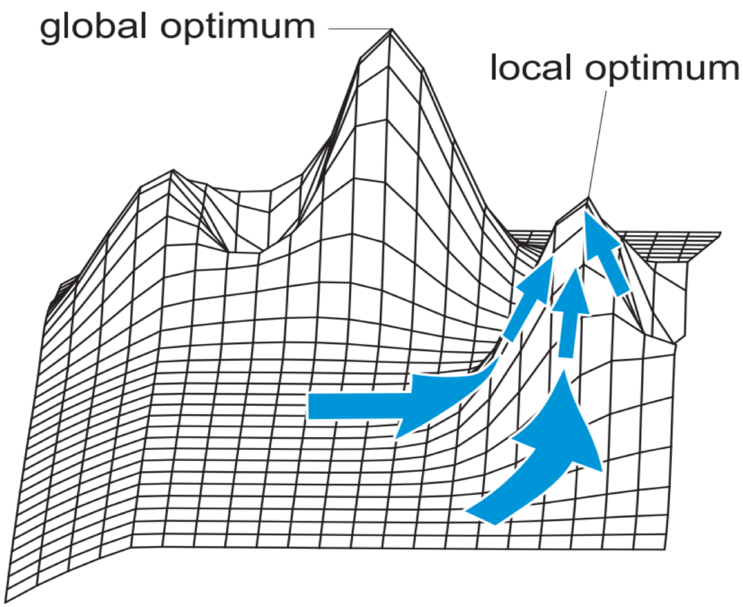
2023/3/23

模拟退火算法

Part1 如何从复杂度的角度衡量算法好坏？
Part2 针对一些有特定形式的最优化问题，如何找到最优解？
Part3 针对较复杂的最优化问题（如NP难问题），如何找到较好解？
局部搜索、模拟退火、遗传算法、差分进化算法、蚁群算法、粒子群优化算法、人工蜂群算法
Part4 针对较复杂的多目标最优化问题，如何找到较好解？

登山搜索缺陷

（原始）登山搜索只考虑最优化目标函数的方向，易陷入局部最优

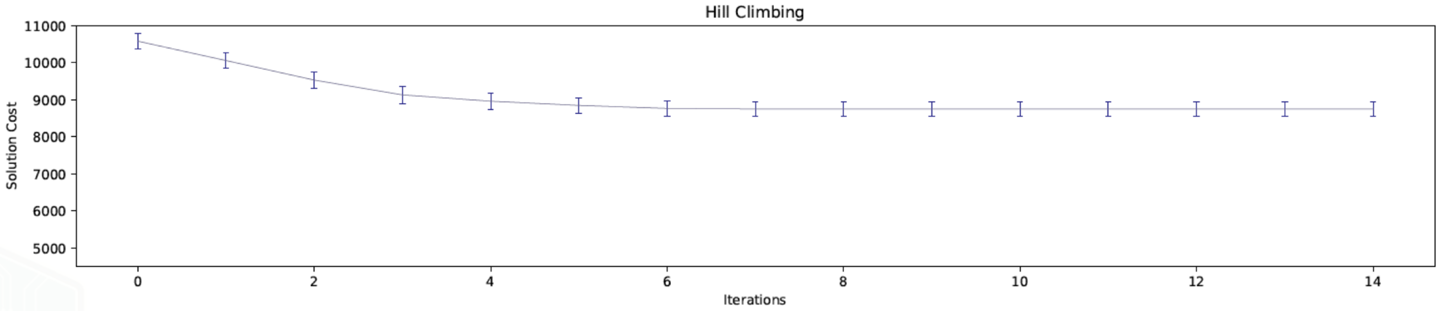


登山搜索缺陷

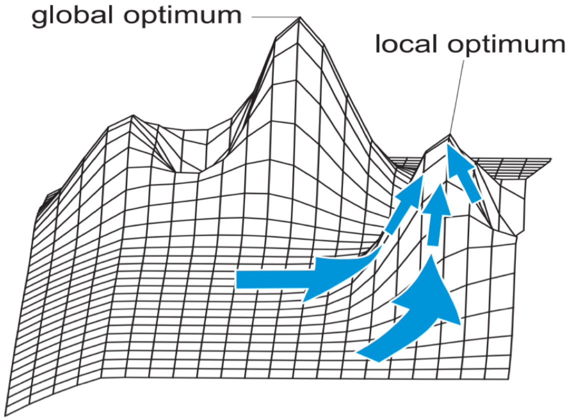
- （原始）登山搜索只考虑最优化目标函数的方向，易陷入局部最优

a）15 cities

前述例子中15个城市旅行商问题（最小化问题）



解决思路



允许朝下山方向移动

引入随机性——随机登山搜索

随机登山搜索

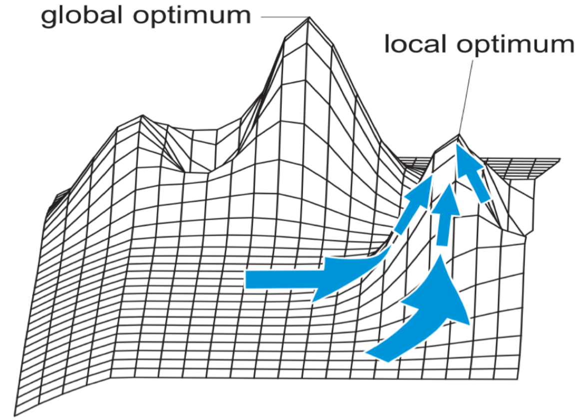
0 初始化当前解$\boldsymbol{x}_{\text{current}}$

1 随机选择一个邻域解$\boldsymbol{x}_{\text{neighbor}}$

2 以概率$\frac{1}{1+e^{\frac{f(x_{\text{current}})-f(x_{\text{neighbor}})}{T}}}$令$\boldsymbol{x}_{\text{current}} \leftarrow \boldsymbol{x}_{\text{neighbor}}$

3 若算法终止条件不满足，返回1

解决思路



允许朝下山方向移动

引入随机性——随机登山搜索

随机登山搜索

0 初始化当前解$\boldsymbol{x}_{\text{current}}$

1 随机选择一个邻域解$\boldsymbol{x}_{\text{neighbor}}$

2 以概率$\frac{1}{1+e^{\frac{f(x_{\text{current}})-f(x_{\text{neighbor}})}{T}}}$令$\boldsymbol{x}_{\text{current}} \leftarrow \boldsymbol{x}_{\text{neighbor}}$

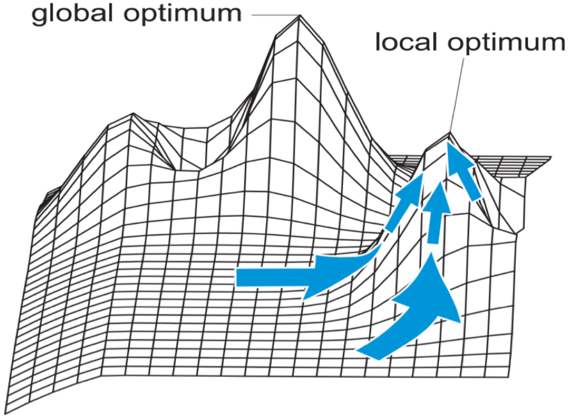
3 若算法终止条件不满足，返回1

仍旧存在问题?

如果已经抵达全局最优可能会跳出来

解决思路？

解决思路



允许朝下山方向移动

要动态降低下山移动概率

引入随机性——随机登山搜索

随机登山搜索

0 初始化当前解$\boldsymbol{x}_{\text{current}}$

1 随机选择一个邻域解$\boldsymbol{x}_{\text{neighbor}}$

2 以概率$\frac{1}{1+e^{\frac{f(x_{\text{current}})-f(x_{\text{neighbor}})}{T}}}$令$\boldsymbol{x}_{\text{current}} \leftarrow \boldsymbol{x}_{\text{neighbor}}$

3 若算法终止条件不满足，返回1

仍旧存在问题?

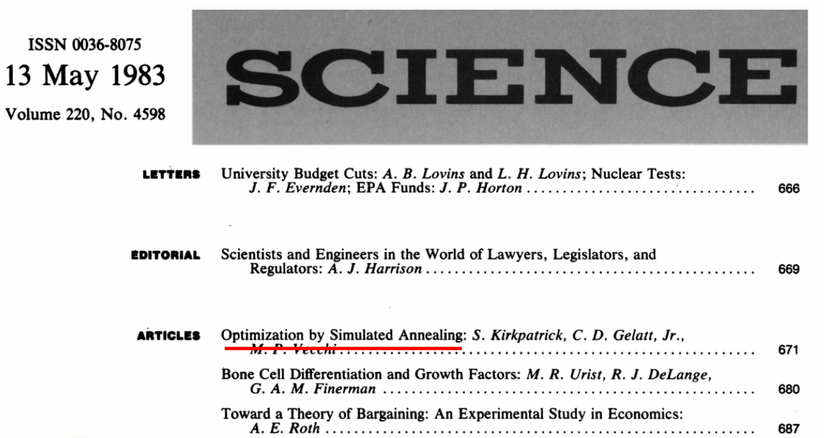
如果已经抵达全局最优可能会跳出来

解决思路？ 随着迭代次数变化，动态调整选择邻域解的概率

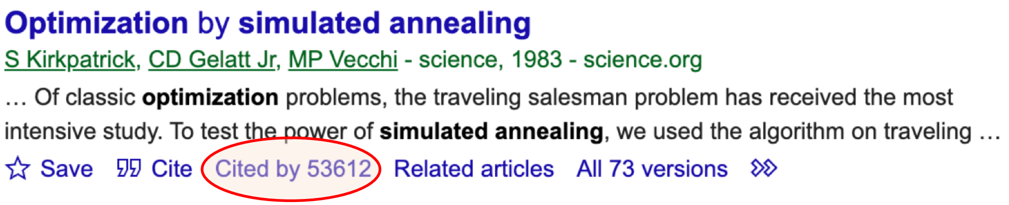
模拟退火算法

模拟退火算法（Simulated Annealing, SA）

- 在引入随机性的同时，随着迭代次数变化动态调整随机性


ISSN 0036-8075
13 May 1983
Volume 220, No. 4598

SCIENCE


Optimization by **simulated annealing**

S Kirkpatrick, CD Gelatt Jr, MP Vecchi - science, 1983 - science.org

… Of classic **optimization** problems, the traveling salesman problem has received the most intensive study. To test the power of **simulated annealing**, we used the algorithm on traveling …

1983年提出用于解决旅行商问题等

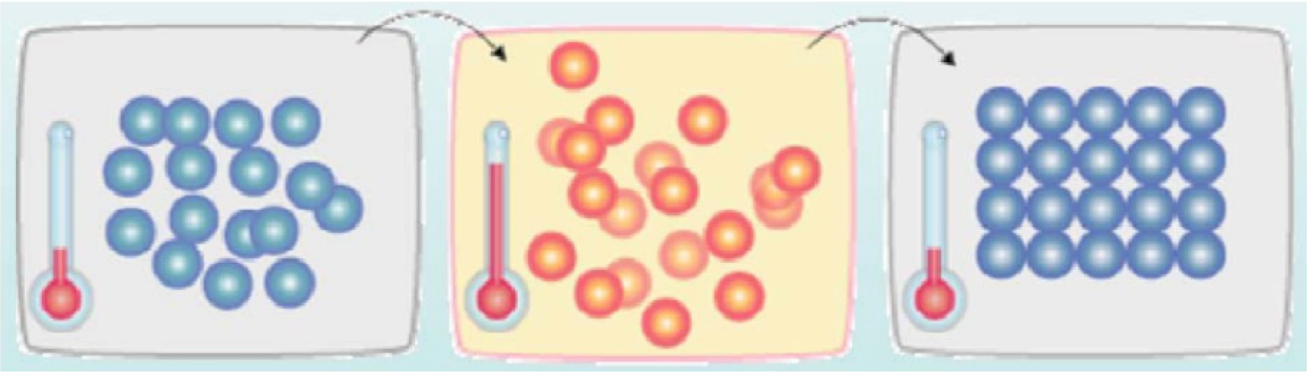
模拟退火算法

工业热处理：退火、正火、淬火、回火

“回火、淬火、正火、退火，这4把火终于整明白了！”

https://www.bilibili.com/video/BV1M3411q7DC?spm_id_from=333.337.search-card.all.click

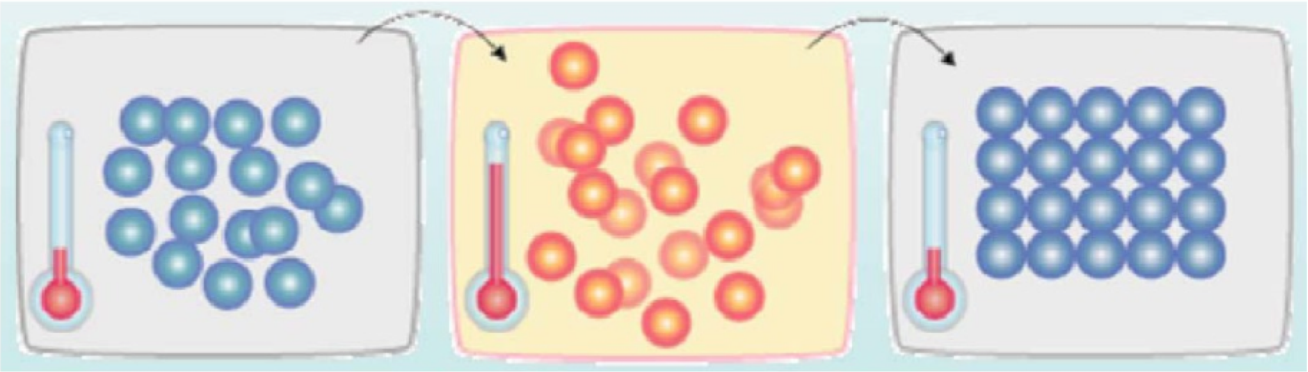
模拟退火算法



物理退火过程：将金属加热到高温，再将其缓慢冷却

- 退火过程中，温度高时，原子活跃度高；温度低时，原子活跃度低
- 经过退火过程，原子排列将改变
- 目的是恢复金属因冷加工而降低的性质，增加柔软性、延展性、韧性

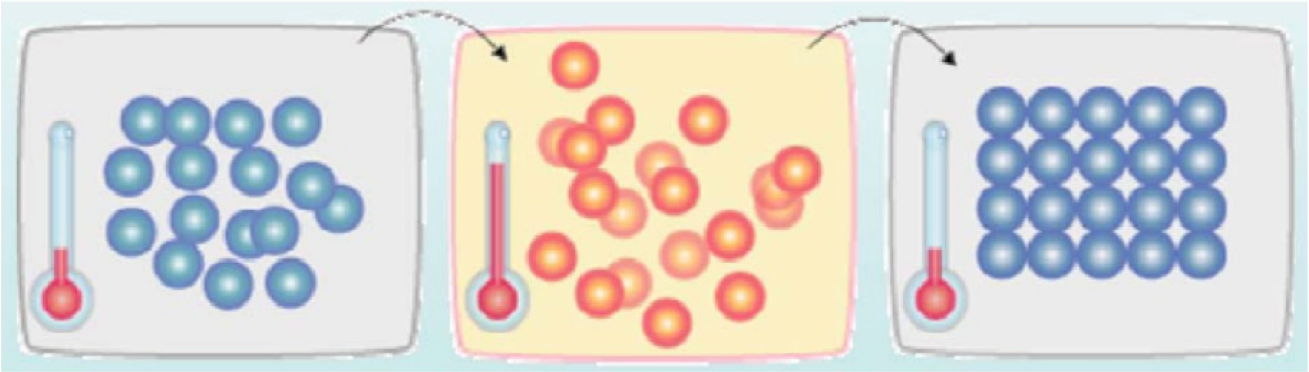
模拟退火算法



模拟退火算法：模仿物理退火过程

➢根据温度的变化调整用邻域解代替当前解的概率

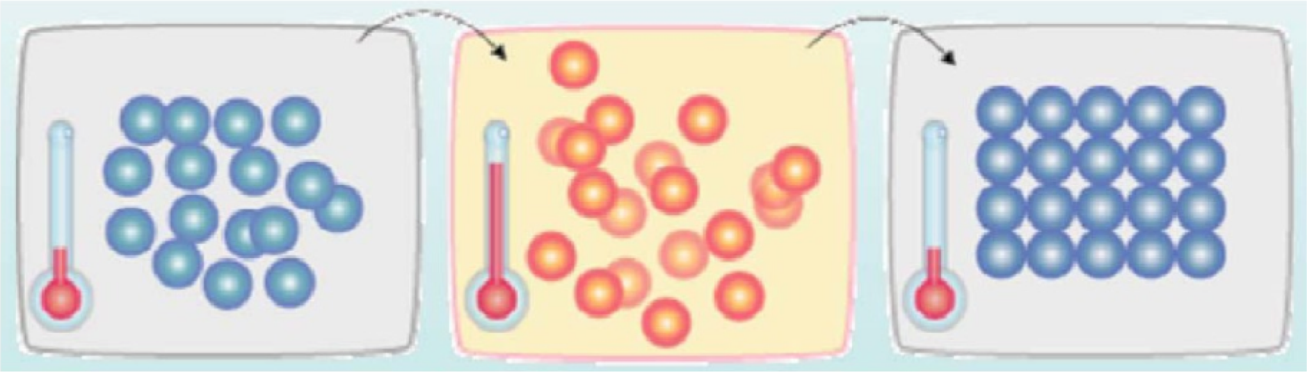
模拟退火算法



模拟退火算法：模仿物理退火过程

- 根据温度的变化调整用邻域解代替当前解的概率
    - ❑ 温度高时，随机性高：更接纳邻域解，即便质量较差
    - ❑ 温度低时，随机性低：趋向于只接纳质量较好的邻域解
    - ❑ 温度降为0时，停止搜索，结束算法

模拟退火算法



模拟退火算法与物理退火过程完整的对偶关系更为复杂（略）

当前解 —— 当前系统物理状态

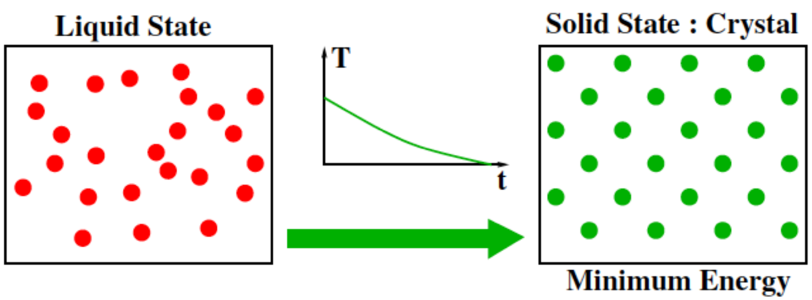
解的质量 —— 物理状态的能量

最优目标函数 —— 能量最小状态（基态）

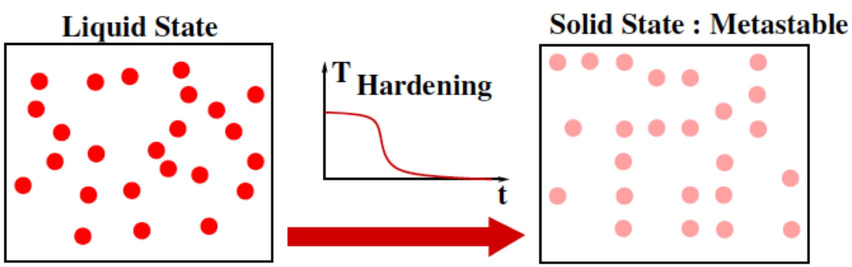
. . .

模拟退火算法

退火需要温度缓慢下降



退火：最终固体状态能量最小、原子对称排列



淬火：最终固体状态非能量最小状态

图片取自Delahaye, et al, “Simulated annealing: From basics to applications”

模拟退火算法

模拟退火算法（求解最大化问题）

初始化当前解$\boldsymbol{x}_{\text{current}}$
for t=1 to ∞:
    根据温度下降规则和t，确定当前温度T
    若T=0，终止算法，输出$\boldsymbol{x}_{\text{current}}$
    随机选择一个邻域解$\boldsymbol{x}_{\text{neighbor}}$
    计算$\Delta E = f(\boldsymbol{x}_{\text{neighbor}}) - f(\boldsymbol{x}_{\text{current}})$
    若$\Delta E > 0$，则令$\boldsymbol{x}_{\text{current}} \leftarrow \boldsymbol{x}_{\text{neighbor}}$
    若$\Delta E \leq 0$，则以$e^{\Delta E/T}$的概率令$\boldsymbol{x}_{\text{current}} \leftarrow \boldsymbol{x}_{\text{neighbor}}$

模拟退火算法

模拟退火算法（求解最大化问题）

初始化当前解$\boldsymbol{x}_{\text{current}}$

for t=1 to ∞:

    根据温度下降规则和t，确定当前温度T　温度下降规则说明了t与T的函数关系

    若T=0，终止算法，输出$\boldsymbol{x}_{\text{current}}$

    随机选择一个邻域解$\boldsymbol{x}_{\text{neighbor}}$　选中的不一定是邻域解中最优的

    计算$\Delta E = f\left(\boldsymbol{x}_{\text{neighbor}}\right) - f\left(\boldsymbol{x}_{\text{current}}\right)$

    若$\Delta E > 0$，则令$\boldsymbol{x}_{\text{current}} \leftarrow \boldsymbol{x}_{\text{neighbor}}$

    若$\Delta E \le 0$，则以$e^{\Delta E / T}$的概率令$\boldsymbol{x}_{\text{current}} \leftarrow \boldsymbol{x}_{\text{neighbor}}$

模拟退火算法

模拟退火算法（求解最大化问题）

```
初始化当前解 x_current
for t=1 to ∞:
    根据温度下降规则和t，确定当前温度T
    若T=0，终止算法，输出 x_current
    随机选择一个邻域解 x_neighbor
    计算ΔE = f(x_neighbor) − f(x_current)
    若ΔE > 0，则令 x_current ← x_neighbor
    若ΔE ≤ 0，则以 e^(ΔE/T) 的概率令 x_current ← x_neighbor
```

关于T

当T为∞时（注意不是t为∞），概率为1，即无条件接纳邻域解

随着T减小，概率减小

当T趋近于0时，概率趋近于0，即仅接纳比当前解更优的邻域解

模拟退火算法

模拟退火算法（求解最大化问题）

初始化当前解$\boldsymbol{x}_{\text{current}}$
for t=1 to ∞:
  根据**温度下降规则**和t，确定当前温度T
  若T=0，终止算法，输出$\boldsymbol{x}_{\text{current}}$
  随机选择一个邻域解$\boldsymbol{x}_{\text{neighbor}}$
  计算$\Delta E = f\left(\boldsymbol{x}_{\text{neighbor}}\right) - f\left(\boldsymbol{x}_{\text{current}}\right)$
  若$\Delta E > 0$，则令$\boldsymbol{x}_{\text{current}} \leftarrow \boldsymbol{x}_{\text{neighbor}}$
  若$\Delta E \leq 0$，则**以$e^{\Delta E / T}$的概率**令$\boldsymbol{x}_{\text{current}} \leftarrow \boldsymbol{x}_{\text{neighbor}}$ 注意ΔE小于等于0

关于ΔE

当ΔE为0时，概率为1，即接纳邻域解

随着ΔE减小，概率减小

当ΔE为-∞时，概率趋近于0，即仅接纳比当前解更优的邻域解

模拟退火算法

模拟退火算法（求解最大化问题）

初始化当前解$\boldsymbol{x}_{\text{current}}$
for t=1 to ∞:
    根据温度下降规则和t，确定当前温度T
    若T=0，终止算法，输出$\boldsymbol{x}_{\text{current}}$
    随机选择一个邻域解$\boldsymbol{x}_{\text{neighbor}}$
    计算$\Delta E = f\left(\boldsymbol{x}_{\text{neighbor}}\right) - f\left(\boldsymbol{x}_{\text{current}}\right)$
    若$\Delta E > 0$，则令$\boldsymbol{x}_{\text{current}} \leftarrow \boldsymbol{x}_{\text{neighbor}}$
    若$\Delta E \leq 0$，则以$e^{\Delta E/T}$的概率令$\boldsymbol{x}_{\text{current}} \leftarrow \boldsymbol{x}_{\text{neighbor}}$

可以理论证明，在满足一系列条件时，当T趋近于0时，$\boldsymbol{x}_{\text{current}}$能收敛到全局最优解

证明思路

第一步：分析在一个固定的温度值T下，当算法执行时间足够长时，解空间中的每个解成为$\boldsymbol{x}_{\text{current}}$的概率

例如，解空间中有4个解。如果固定$T=10$，调用模拟退火算法执行10^5次（t从1增长到10^5）对这4个解分别求其成为$\boldsymbol{x}_{\text{current}}$的概率？

$f=10$ 1	$f=15$ 2
$f=4$ 3	$f=6$ 4

证明思路

第一步：分析在一个固定的温度值T下，当算法执行时间足够长时，解空间中的每个解成为$\boldsymbol{x}_{\text{current}}$的概率

例如，解空间中有4个解。如果固定$T = 10$，调用模拟退火算法执行10^5次（t从1增长到10^5）对这4个解分别求其成为$\boldsymbol{x}_{\text{current}}$的概率？

$f = 10$ 1	$f = 15$ 2
$f = 4$ 3	$f = 6$ 4

第二步：求证当温度值T趋近于0时，最优解成为$\boldsymbol{x}_{\text{current}}$的概率趋近于1

证明思路

例如，解空间中有4个解。如果固定$T=10$，调用模拟退火算法执行10^5次（t从1增长到10^5）对这4个解分别求其成为$\boldsymbol{x}_{\mathrm{current}}$的概率？

$f=10$ 1	$f=15$ 2
$f=4$ 3	$f=6$ 4

假设选择上/下/左/右一格范围为邻域

当前解是1号解时，下一个回合的当前解是2/3/4/1号解的条件概率：

$$\Pr\left(\boldsymbol{x}_{\mathrm{current}}^{t+1}=2\middle|\boldsymbol{x}_{\mathrm{current}}^{t}=1\right)=\frac{1}{2}$$

$$\Pr\left(\boldsymbol{x}_{\mathrm{current}}^{t+1}=3\middle|\boldsymbol{x}_{\mathrm{current}}^{t}=1\right)=\frac{1}{2}e^{-\frac{6}{10}}$$

$$\Pr\left(\boldsymbol{x}_{\mathrm{current}}^{t+1}=4\middle|\boldsymbol{x}_{\mathrm{current}}^{t}=1\right)=0$$

$$\Pr\left(\boldsymbol{x}_{\mathrm{current}}^{t+1}=1\middle|\boldsymbol{x}_{\mathrm{current}}^{t}=1\right)=1-\frac{1}{2}-\frac{1}{2}e^{-\frac{6}{10}}$$

证明思路

例如，解空间中有4个解。如果固定$T = 10$，调用模拟退火算法执行10^5次（t从1增长到10^5）对这4个解分别求其成为$\boldsymbol{x}_{\text{current}}$的概率？

$f = 10$ ①	$f = 15$ ②
$f = 4$ ③	$f = 6$ ④

当前解是2号解时，下一个回合的当前解是1/3/4/2号解的条件概率：

$$\Pr\left(\boldsymbol{x}_{\text{current}}^{t+1} = 1 \middle| \boldsymbol{x}_{\text{current}}^{t} = 2\right) = \frac{1}{2} e^{-\frac{5}{10}}$$

$$\Pr\left(\boldsymbol{x}_{\text{current}}^{t+1} = 3 \middle| \boldsymbol{x}_{\text{current}}^{t} = 2\right) = 0$$

$$\Pr\left(\boldsymbol{x}_{\text{current}}^{t+1} = 4 \middle| \boldsymbol{x}_{\text{current}}^{t} = 2\right) = \frac{1}{2} e^{-\frac{9}{10}}$$

$$\Pr\left(\boldsymbol{x}_{\text{current}}^{t+1} = 2 \middle| \boldsymbol{x}_{\text{current}}^{t} = 2\right) = 1 - \frac{1}{2} e^{-\frac{5}{10}} - \frac{1}{2} e^{-\frac{9}{10}}$$

证明思路

例如，解空间中有4个解。如果固定$T=10$，调用模拟退火算法执行10^5次（t从1增长到10^5）对这4个解分别求其成为$\boldsymbol{x}_{\text{current}}$的概率？

$f=10$ 1	$f=15$ 2
$f=4$ 3	$f=6$ 4

当前解是3号解时，下一个回合的当前解是1/2/4/3号解的条件概率：

$$\Pr\left(\boldsymbol{x}_{\text{current}}^{t+1}=1 \middle| \boldsymbol{x}_{\text{current}}^{t}=3\right)=\frac{1}{2}$$
$$\Pr\left(\boldsymbol{x}_{\text{current}}^{t+1}=2 \middle| \boldsymbol{x}_{\text{current}}^{t}=3\right)=0$$
$$\Pr\left(\boldsymbol{x}_{\text{current}}^{t+1}=4 \middle| \boldsymbol{x}_{\text{current}}^{t}=3\right)=\frac{1}{2}$$
$$\Pr\left(\boldsymbol{x}_{\text{current}}^{t+1}=3 \middle| \boldsymbol{x}_{\text{current}}^{t}=3\right)=0$$

证明思路

例如，解空间中有4个解。如果固定$T=10$，调用模拟退火算法执行10^5次（t从1增长到10^5）对这4个解分别求其成为$\boldsymbol{x}_{\text{current}}$的概率？

$f=10$ 1	$f=15$ 2
$f=4$ 3	$f=6$ 4

当前解是4号解时，下一个回合的当前解是1/2/3/4号解的条件概率：

$$\Pr\left(\boldsymbol{x}_{\text{current}}^{t+1}=1 \middle| \boldsymbol{x}_{\text{current}}^{t}=4\right)=0$$

$$\Pr\left(\boldsymbol{x}_{\text{current}}^{t+1}=2 \middle| \boldsymbol{x}_{\text{current}}^{t}=4\right)=\frac{1}{2}$$

$$\Pr\left(\boldsymbol{x}_{\text{current}}^{t+1}=3 \middle| \boldsymbol{x}_{\text{current}}^{t}=4\right)=\frac{1}{2}e^{-\frac{2}{10}}$$

$$\Pr\left(\boldsymbol{x}_{\text{current}}^{t+1}=4 \middle| \boldsymbol{x}_{\text{current}}^{t}=4\right)=1-\frac{1}{2}-\frac{1}{2}e^{-\frac{2}{10}}$$

证明思路

例如，解空间中有4个解。如果固定$T=10$，调用模拟退火算法执行10^5次（t从1增长到10^5）对这4个解分别求其成为$\boldsymbol{x}_{\text{current}}$的概率？

$f=10$ 1	$f=15$ 2
$f=4$ 3	$f=6$ 4

$$\mathbf{P}=\begin{bmatrix} \boxed{\begin{matrix}\Pr(\boldsymbol{x}_{\text{current}}^{t+1}=1\mid\boldsymbol{x}_{\text{current}}^{t}=1)\\ \Pr(\boldsymbol{x}_{\text{current}}^{t+1}=2\mid\boldsymbol{x}_{\text{current}}^{t}=1)\\ \Pr(\boldsymbol{x}_{\text{current}}^{t+1}=3\mid\boldsymbol{x}_{\text{current}}^{t}=1)\\ \Pr(\boldsymbol{x}_{\text{current}}^{t+1}=4\mid\boldsymbol{x}_{\text{current}}^{t}=1)\end{matrix}} & \begin{matrix}\Pr(\boldsymbol{x}_{\text{current}}^{t+1}=1\mid\boldsymbol{x}_{\text{current}}^{t}=2)\\ \Pr(\boldsymbol{x}_{\text{current}}^{t+1}=2\mid\boldsymbol{x}_{\text{current}}^{t}=2)\\ \Pr(\boldsymbol{x}_{\text{current}}^{t+1}=3\mid\boldsymbol{x}_{\text{current}}^{t}=2)\\ \Pr(\boldsymbol{x}_{\text{current}}^{t+1}=4\mid\boldsymbol{x}_{\text{current}}^{t}=2)\end{matrix} & \begin{matrix}\Pr(\boldsymbol{x}_{\text{current}}^{t+1}=1\mid\boldsymbol{x}_{\text{current}}^{t}=3)\\ \Pr(\boldsymbol{x}_{\text{current}}^{t+1}=2\mid\boldsymbol{x}_{\text{current}}^{t}=3)\\ \Pr(\boldsymbol{x}_{\text{current}}^{t+1}=3\mid\boldsymbol{x}_{\text{current}}^{t}=3)\\ \Pr(\boldsymbol{x}_{\text{current}}^{t+1}=4\mid\boldsymbol{x}_{\text{current}}^{t}=3)\end{matrix} & \begin{matrix}\Pr(\boldsymbol{x}_{\text{current}}^{t+1}=1\mid\boldsymbol{x}_{\text{current}}^{t}=4)\\ \Pr(\boldsymbol{x}_{\text{current}}^{t+1}=2\mid\boldsymbol{x}_{\text{current}}^{t}=4)\\ \Pr(\boldsymbol{x}_{\text{current}}^{t+1}=3\mid\boldsymbol{x}_{\text{current}}^{t}=4)\\ \Pr(\boldsymbol{x}_{\text{current}}^{t+1}=4\mid\boldsymbol{x}_{\text{current}}^{t}=4)\end{matrix} \end{bmatrix}$$

$$\begin{bmatrix}\Pr(\boldsymbol{x}_{\text{current}}^{t+1}=1)\\ \Pr(\boldsymbol{x}_{\text{current}}^{t+1}=2)\\ \Pr(\boldsymbol{x}_{\text{current}}^{t+1}=3)\\ \Pr(\boldsymbol{x}_{\text{current}}^{t+1}=4)\end{bmatrix}=\mathbf{P}\begin{bmatrix}\Pr(\boldsymbol{x}_{\text{current}}^{t}=1)\\ \Pr(\boldsymbol{x}_{\text{current}}^{t}=2)\\ \Pr(\boldsymbol{x}_{\text{current}}^{t}=3)\\ \Pr(\boldsymbol{x}_{\text{current}}^{t}=4)\end{bmatrix}$$

证明思路

例如，解空间中有4个解。如果固定$T=10$，调用模拟退火算法执行10^5次（t从1增长到10^5）对这4个解分别求其成为$\boldsymbol{x}_{\text{current}}$的概率？

$f=10$ ①	$f=15$ ②
$f=4$ ③	$f=6$ ④

$$\mathbf{P}=\begin{bmatrix}
\Pr(\boldsymbol{x}_{\text{current}}^{t+1}=1|\boldsymbol{x}_{\text{current}}^{t}=1) & \Pr(\boldsymbol{x}_{\text{current}}^{t+1}=1|\boldsymbol{x}_{\text{current}}^{t}=2) & \Pr(\boldsymbol{x}_{\text{current}}^{t+1}=1|\boldsymbol{x}_{\text{current}}^{t}=3) & \Pr(\boldsymbol{x}_{\text{current}}^{t+1}=1|\boldsymbol{x}_{\text{current}}^{t}=4)\\
\Pr(\boldsymbol{x}_{\text{current}}^{t+1}=2|\boldsymbol{x}_{\text{current}}^{t}=1) & \Pr(\boldsymbol{x}_{\text{current}}^{t+1}=2|\boldsymbol{x}_{\text{current}}^{t}=2) & \Pr(\boldsymbol{x}_{\text{current}}^{t+1}=2|\boldsymbol{x}_{\text{current}}^{t}=3) & \Pr(\boldsymbol{x}_{\text{current}}^{t+1}=2|\boldsymbol{x}_{\text{current}}^{t}=4)\\
\Pr(\boldsymbol{x}_{\text{current}}^{t+1}=3|\boldsymbol{x}_{\text{current}}^{t}=1) & \Pr(\boldsymbol{x}_{\text{current}}^{t+1}=3|\boldsymbol{x}_{\text{current}}^{t}=2) & \Pr(\boldsymbol{x}_{\text{current}}^{t+1}=3|\boldsymbol{x}_{\text{current}}^{t}=3) & \Pr(\boldsymbol{x}_{\text{current}}^{t+1}=3|\boldsymbol{x}_{\text{current}}^{t}=4)\\
\Pr(\boldsymbol{x}_{\text{current}}^{t+1}=4|\boldsymbol{x}_{\text{current}}^{t}=1) & \Pr(\boldsymbol{x}_{\text{current}}^{t+1}=4|\boldsymbol{x}_{\text{current}}^{t}=2) & \Pr(\boldsymbol{x}_{\text{current}}^{t+1}=4|\boldsymbol{x}_{\text{current}}^{t}=3) & \Pr(\boldsymbol{x}_{\text{current}}^{t+1}=4|\boldsymbol{x}_{\text{current}}^{t}=4)
\end{bmatrix}$$

$$\begin{bmatrix}
\Pr(\boldsymbol{x}_{\text{current}}^{100000}=1)\\
\Pr(\boldsymbol{x}_{\text{current}}^{100000}=2)\\
\Pr(\boldsymbol{x}_{\text{current}}^{100000}=3)\\
\Pr(\boldsymbol{x}_{\text{current}}^{100000}=4)
\end{bmatrix}=\mathbf{P}^{100000}\begin{bmatrix}
\Pr(\boldsymbol{x}_{\text{current}}^{0}=1)\\
\Pr(\boldsymbol{x}_{\text{current}}^{0}=2)\\
\Pr(\boldsymbol{x}_{\text{current}}^{0}=3)\\
\Pr(\boldsymbol{x}_{\text{current}}^{0}=4)
\end{bmatrix}$$

证明思路

例如，解空间中有4个解。如果固定$T=10$，调用模拟退火算法执行10^5次（t从1增长到10^5）对这4个解分别求其成为$\boldsymbol{x}_{\text{current}}$的概率？

$f=10$ 1	$f=15$ 2
$f=4$ 3	$f=6$ 4

Users ▸ haoranyu ▸ Documents ▸ MATLAB ▸

Command Window

```
>> clear all;
close all;

T=10;
p11=1-0.5-0.5*exp(-6/T);
p21=0.5;
p31=0.5*exp(-6/T);
p41=0;

p12=0.5*exp(-5/T);
p22=1-0.5*exp(-5/T)-0.5*exp(-9/T);
p32=0;
p42=0.5*exp(-9/T);

p13=0.5;
p23=0;
p33=0;
p43=0.5;

p14=0;
p24=0.5;
p34=0.5*exp(-2/10);
p44=1-0.5-0.5*exp(-2/10);

P=[p11, p12, p13, p14; p21, p22, p23, p24; p31, p32, p33, p34; p41, p42, p43, p44];

distribution=P^100000*[1/4; 1/4; 1/4; 1/4]

distribution =

    0.2585
    0.4263
    0.1419
    0.1733
```

$$\begin{bmatrix} \Pr(\boldsymbol{x}_{\text{current}}^{100000}=1) \\ \Pr(\boldsymbol{x}_{\text{current}}^{100000}=2) \\ \Pr(\boldsymbol{x}_{\text{current}}^{100000}=3) \\ \Pr(\boldsymbol{x}_{\text{current}}^{100000}=4) \end{bmatrix} = \begin{bmatrix} 0.2585 \\ 0.4263 \\ 0.1419 \\ 0.1733 \end{bmatrix}$$

证明思路

例如，解空间中有4个解。如果固定$T=10$，调用模拟退火算法执行10^5次（t从1增长到10^5）对这4个解分别求其成为$\boldsymbol{x}_{\text{current}}$的概率？

$f=10$ ①	$f=15$ ②
$f=4$ ③	$f=6$ ④

```
>> q1=exp(1);
>> q2=exp(15/10);
>> q3=exp(4/10);
>> q4=exp(6/10);
>> redvector=1/(q1+q2+q3+q4)*[q1;q2;q3;q4]

redvector =

    0.2585
    0.4263
    0.1419
    0.1733
```

Users ▸ haoranyu ▸ Documents ▸ MATLAB ▸

Command Window

```
>> clear all;
close all;

T=10;
p11=1-0.5-0.5*exp(-6/T);
p21=0.5;
p31=0.5*exp(-6/T);
p41=0;

p12=0.5*exp(-5/T);
p22=1-0.5*exp(-5/T)-0.5*exp(-9/T);
p32=0;
p42=0.5*exp(-9/T);

p13=0.5;
p23=0;
p33=0;
p43=0.5;

p14=0;
p24=0.5;
p34=0.5*exp(-2/10);
p44=1-0.5-0.5*exp(-2/10);

P=[p11, p12, p13, p14; p21, p22, p23, p24; p31, p32, p33, p34; p41, p42, p43, p44];

distribution=P^100000*[1/4; 1/4; 1/4; 1/4]

distribution =

    0.2585
    0.4263
    0.1419
    0.1733
```

$$\begin{bmatrix}\Pr(\boldsymbol{x}_{\text{current}}^{100000}=1)\\ \Pr(\boldsymbol{x}_{\text{current}}^{100000}=2)\\ \Pr(\boldsymbol{x}_{\text{current}}^{100000}=3)\\ \Pr(\boldsymbol{x}_{\text{current}}^{100000}=4)\end{bmatrix}=\begin{bmatrix}0.2585\\0.4263\\0.1419\\0.1733\end{bmatrix}\approx\frac{1}{\text{归一化因子}}\begin{bmatrix}e^{\frac{10}{10}}\\ e^{\frac{15}{10}}\\ e^{\frac{4}{10}}\\ e^{\frac{6}{10}}\end{bmatrix}$$

证明思路

在给定T时，在各解之间跳变的过程可以由如下转移概率表征：

$$P_{ij}(T)=\begin{cases} G_{ij}(T)A_{ij}(T), & i\neq j, \\ 1-\sum_{l\in S, l\neq i} G_{il}(T)A_{il}(T), & i=j. \end{cases}$$

解i到解j的转移概率

$$G_{ij}(T)=\begin{cases} \frac{1}{|N(i)|} & j\in N(i), \\ 0, & otherwise. \end{cases}$$

当前解是i时，解j被选中的概率

$$A_{ij}(T)=\begin{cases} 1, & Val(j)>Val(i), \\ e^{(Val(j)-Val(i))/T}, & otherwise. \end{cases}$$

当前解是i且解j被选中时，

i转移到j的概率

图片取自Kathryn A. Dowsland, Jonathan M. Thompson, "Simulated Annealing," Handbook of Natural Computing

证明思路

$$P_{ij}(T)=\begin{cases} G_{ij}(T)A_{ij}(T), & i\neq j, \\ 1-\sum\limits_{l\in S, l\neq i} G_{il}(T)A_{il}(T), & i=j. \end{cases}$$

解i到解j的转移概率

利用马尔可夫链的相关知识，可证明在一定条件下*，各个解成为当前解的概率存在唯一稳态分布

* 解空间中各个解之间是强互联关系。即给定一个解，从任意解可以不断沿着邻域解组成的路径达到该解

证明思路

$$P_{ij}(T)=\begin{cases} G_{ij}(T)A_{ij}(T), & i\neq j, \\ 1-\sum\limits_{l\in S, l\neq i} G_{il}(T)A_{il}(T), & i=j. \end{cases}$$

解i到解j的转移概率

利用马尔可夫链的相关知识，可证明在一定条件下*，各个解成为当前解的概率存在唯一稳态分布

$$Q_T(i)=\frac{\exp\left(\frac{Val(i)}{T}\right)}{\sum\limits_{j\in S}\exp\left(\frac{Val(j)}{T}\right)}$$

* 解空间中各个解之间是强互联关系。即给定一个解，从任意解可以不断沿着邻域解组成的路径达到该解

证明思路

$$P_{ij}(T)=\begin{cases} G_{ij}(T)A_{ij}(T), & i\neq j, \\ 1-\sum\limits_{l\in S, l\neq i} G_{il}(T)A_{il}(T), & i=j. \end{cases}$$

解i到解j的转移概率

利用马尔可夫链的相关知识，可证明在一定条件下*，各个解成为当前解的概率存在唯一稳态分布

$$Q_T(i)=\frac{\exp\left(\frac{Val(i)}{T}\right)}{\sum\limits_{j\in S}\exp\left(\frac{Val(j)}{T}\right)}$$

$f=10$ 1	$f=15$ 2
$f=4$ 3	$f=6$ 4

$$\begin{bmatrix} \Pr(\boldsymbol{x}_{\mathtt{current}}^{100000}=1) \\ \Pr(\boldsymbol{x}_{\mathtt{current}}^{100000}=2) \\ \Pr(\boldsymbol{x}_{\mathtt{current}}^{100000}=3) \\ \Pr(\boldsymbol{x}_{\mathtt{current}}^{100000}=4) \end{bmatrix} = \begin{bmatrix} 0.2585 \\ 0.4263 \\ 0.1419 \\ 0.1733 \end{bmatrix} \approx \frac{1}{\text{归一化因子}} \begin{bmatrix} e^{\frac{10}{10}} \\ e^{\frac{15}{10}} \\ e^{\frac{4}{10}} \\ e^{\frac{6}{10}} \end{bmatrix}$$

* 解空间中各个解之间是强互联关系。即给定一个解，从任意解可以不断沿着邻域解组成的路径达到该解

证明思路

$$Q_T(i)=\frac{\exp\left(\frac{Val(i)}{T}\right)}{\sum_{j\in S}\exp\left(\frac{Val(j)}{T}\right)}$$

当温度值T趋近于0时？

证明思路

$$Q_T(i) = \frac{\exp\left(\frac{Val(i)}{T}\right)}{\sum_{j \in S} \exp\left(\frac{Val(j)}{T}\right)}$$

当温度值T趋近于0时？

$T = 10$

$f = 10$ 1	$f = 15$ 2
$f = 4$ 3	$f = 6$ 4

$$\begin{bmatrix} \Pr(\boldsymbol{x}_{\text{current}}^{100000} = 1) \\ \Pr(\boldsymbol{x}_{\text{current}}^{100000} = 2) \\ \Pr(\boldsymbol{x}_{\text{current}}^{100000} = 3) \\ \Pr(\boldsymbol{x}_{\text{current}}^{100000} = 4) \end{bmatrix} = \begin{bmatrix} 0.2585 \\ 0.4263 \\ 0.1419 \\ 0.1733 \end{bmatrix} \approx \frac{1}{\text{归一化因子}} \begin{bmatrix} e^{\frac{10}{10}} \\ e^{\frac{15}{10}} \\ e^{\frac{4}{10}} \\ e^{\frac{6}{10}} \end{bmatrix}$$

$T = 1$

$f = 10$ 1	$f = 15$ 2
$f = 4$ 3	$f = 6$ 4

$$\begin{bmatrix} \Pr(\boldsymbol{x}_{\text{current}}^{100000} = 1) \\ \Pr(\boldsymbol{x}_{\text{current}}^{100000} = 2) \\ \Pr(\boldsymbol{x}_{\text{current}}^{100000} = 3) \\ \Pr(\boldsymbol{x}_{\text{current}}^{100000} = 4) \end{bmatrix} = \begin{bmatrix} 0.0067 \\ 0.9931 \\ 0.0000 \\ 0.0001 \end{bmatrix} \approx \frac{1}{\text{归一化因子}} \begin{bmatrix} e^{\frac{10}{1}} \\ e^{\frac{15}{1}} \\ e^{\frac{4}{1}} \\ e^{\frac{6}{1}} \end{bmatrix}$$

证明思路

$$Q_T(i)=\frac{\exp\left(\frac{Val(i)}{T}\right)}{\sum_{j\in S}\exp\left(\frac{Val(j)}{T}\right)}$$

当温度值T趋近于0时？

当在每个T温度下，在各种解之间的跳变都到达到稳态分布时，算法将逐渐趋近于最优解（注意是在一定理论条件满足的情况下达到全局最优解）

证明思路

第一步：分析在一个固定的温度值T下，当算法执行时间足够长时，解空间中的每个解成为x_{current}的概率

第二步：求证当温度值T趋近于0时，最优解成为x_{current}的概率趋近于1

完整证明需要考虑T逐步减小的过程，即非齐次马尔可夫链

T随时间变化需要满足一定条件（T不能下降得太快）

[BOOK] **Simulated annealing** and **Boltzmann machines**: a stochastic approach to combinatorial optimization and neural computing

E Aarts, J Korst - 1989 - dl.acm.org

Large combinatorial optimization problems (the most famous example is the traveling salesman problem) are notably difficult because the number of feasible configurations grows exponentially with the size of the problem. Algorithms exist that require fewer calculations and find the optimum solution most of the time or a good suboptimal solution, but they are hard to design and depend strongly upon the problem. A stochastic (ie, Monte Carlo) approach is an attractive alternative, for the same reasons that such an approach is applied …

模拟退火算法

模拟退火算法（求解最大化问题）

初始化当前解$\boldsymbol{x}_{\text{current}}$

for t=1 to ∞:

    根据温度下降规则和t，确定当前温度T

    若T=0，终止算法，输出$\boldsymbol{x}_{\text{current}}$

    随机选择一个邻域解$\boldsymbol{x}_{\text{neighbor}}$

    计算$\Delta E = f(\boldsymbol{x}_{\text{neighbor}}) - f(\boldsymbol{x}_{\text{current}})$

    若$\Delta E > 0$，则令$\boldsymbol{x}_{\text{current}} \leftarrow \boldsymbol{x}_{\text{neighbor}}$

    若$\Delta E \leq 0$，则以$e^{\Delta E/T}$的概率令$\boldsymbol{x}_{\text{current}} \leftarrow \boldsymbol{x}_{\text{neighbor}}$

哪些需要人为设置的？

模拟退火算法

模拟退火算法（求解最大化问题）

0 初始化当前解$\boldsymbol{x}_{\text{current}}$

1 for t=1 to ∞:

2 　根据温度下降规则和t，确定当前温度T

3 　若T=0，终止算法，输出$\boldsymbol{x}_{\text{current}}$

4 　随机选择一个邻域解$\boldsymbol{x}_{\text{neighbor}}$

5 　计算$\Delta E = f\left(\boldsymbol{x}_{\text{neighbor}}\right) - f\left(\boldsymbol{x}_{\text{current}}\right)$

6 　若$\Delta E > 0$，则令$\boldsymbol{x}_{\text{current}} \leftarrow \boldsymbol{x}_{\text{neighbor}}$

7 　若$\Delta E \leq 0$，则以$e^{\Delta E/T}$的概率令$\boldsymbol{x}_{\text{current}} \leftarrow \boldsymbol{x}_{\text{neighbor}}$

一、**起始温度：**需要足够高以使得所有的比当前解差的邻域解都能以一个足够高的概率（如0.98）被接纳

模拟退火算法

一、**起始温度：**需要足够高以使得所有的**比当前解差的**邻域解都能以一个足够高的概率被接纳

若$\Delta E \leq 0$，则以$e^{\Delta E/T}$的概率令$\boldsymbol{x}_{\text{current}} \leftarrow \boldsymbol{x}_{\text{neighbor}}$

模拟退火算法

一、起始温度：需要足够高以使得所有的比当前解差的邻域解都能以一个足够高的概率被接纳

若$\Delta E \leq 0$，则以$e^{\Delta E/T}$的概率令$\boldsymbol{x}_{\text{current}} \leftarrow \boldsymbol{x}_{\text{neighbor}}$

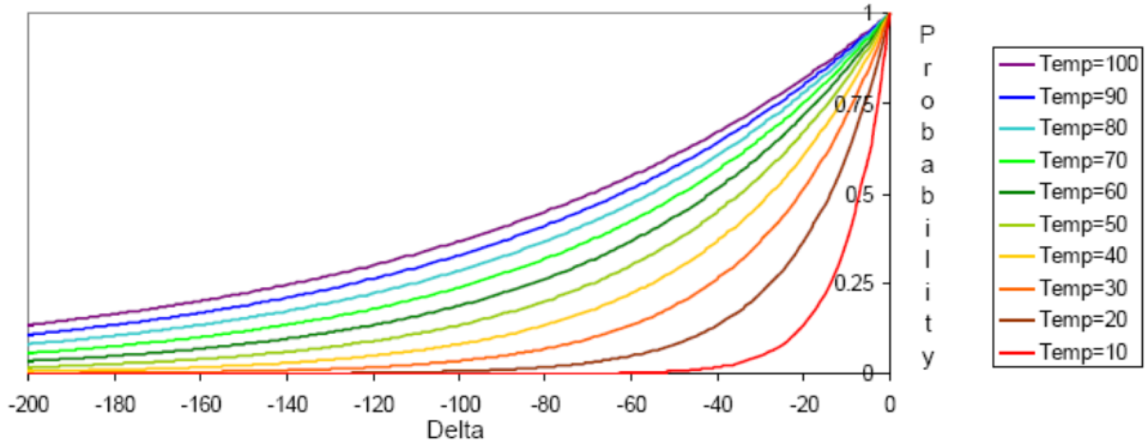
T_0=(-所有“下山”跳变的质量下降程度的平均值)/ln(需要达到的概率)

在实际中用在解空间上的随机游走估测该项目

或者可以（从小到大）尝试不同起始值，观察在起始温度下邻域解被接纳的比例，选择使接纳比例较大（比如接近于1）的初始值

模拟退火算法

一、起始温度：需要足够高以使得所有的比当前解差的邻域解都能以一个足够高的概率被接纳



Initially temperature is very high (most bad moves accepted)

Temp slowly goes to 0, with multiple moves attempted at each temperature

Final runs with temp=0 (always reject bad moves) greedily “quench” the system

图片取自Scott Hauck <UW EE 541: Automated Layout of Integrated Circuits>

模拟退火算法

模拟退火算法（求解最大化问题）

初始化当前解$\boldsymbol{x}_{\text{current}}$
for t=1 to ∞:
    根据温度下降规则和t，确定当前温度T
    若T=0，终止算法，输出$\boldsymbol{x}_{\text{current}}$
    随机选择一个邻域解$\boldsymbol{x}_{\text{neighbor}}$
    计算$\Delta E = f\left(\boldsymbol{x}_{\text{neighbor}}\right) - f\left(\boldsymbol{x}_{\text{current}}\right)$
    若$\Delta E > 0$，则令$\boldsymbol{x}_{\text{current}} \leftarrow \boldsymbol{x}_{\text{neighbor}}$
    若$\Delta E \leq 0$，则以$e^{\Delta E/T}$的概率令$\boldsymbol{x}_{\text{current}} \leftarrow \boldsymbol{x}_{\text{neighbor}}$

二、降温过程：需要相对缓慢

模拟退火算法

二、降温过程：

（1）温度取值

常见几何递减：$T_k = T_0\alpha^k$（α一般取0.8到0.99之间）

另外还有线性、对数递减等不同选择

（2）每个温度下执行的次数（每个温度下不止一次跳变）

尽量确保在每个温度下，当前解的邻域解大都能被随机选择到

The value of *nrep* is often related to the size of the neighborhood, and may vary from temperature to temperature. For example, it is important to spend enough time at low temperatures to ensure that the regions around a local optimum have been fully explored.

模拟退火算法

模拟退火算法（求解最大化问题）

```
初始化当前解 x_current
for t=1 to ∞:
    根据温度下降规则和t，确定当前温度T
    若T=0，终止算法，输出 x_current
    随机选择一个邻域解 x_neighbor
    计算 ΔE = f(x_neighbor) − f(x_current)
    若 ΔE > 0，则令 x_current ← x_neighbor
    若 ΔE ≤ 0，则以 e^(ΔE/T) 的概率令 x_current ← x_neighbor
```

三、**结束条件**：理论上等温度降为0，但实际可能需要时间过长、且早已达到理想的解

模拟退火算法

三、结束条件：

—— 如果连续不接纳邻域解的次数超过限制

—— 如果最近若干解的质量的平均值几乎不再上升

—— acceptance ratio（即接纳邻域解次数/总尝试次数）低于一定值

……

旅行商问题

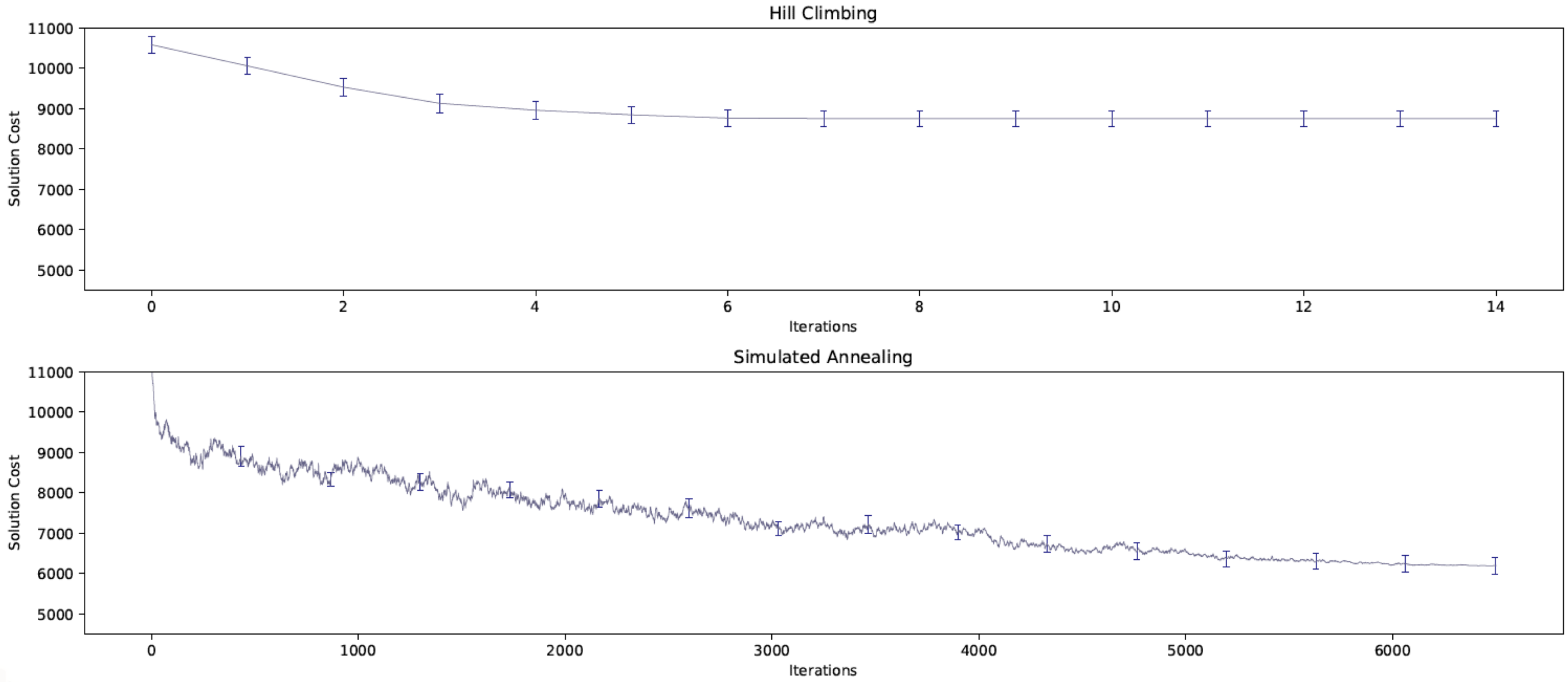
用模拟退火算法求解旅行商问题

- *proposal mechanism:* uniform random choice from 2-exchange neighbourhood; 定义邻域
- *acceptance criterion:* Metropolis condition (always accept improving steps, accept worsening steps with probability $\exp[(f(s) - f(s'))/T]$); 注意旅行商问题是最小化问题
- *annealing schedule*: geometric cooling $T := 0.95 \cdot T$ with $n \cdot (n-1)$ steps at each temperature (n = number of vertices in given graph), T_0 chosen such that 97% of proposed steps are accepted; 起始温度、降温过程
- *termination:* when for five successive temperature values no improvement in solution quality and acceptance ratio $< 2\%$. 结束条件

例子取自Thomas Stutzle, Heuristic Optimization 2014

旅行商问题

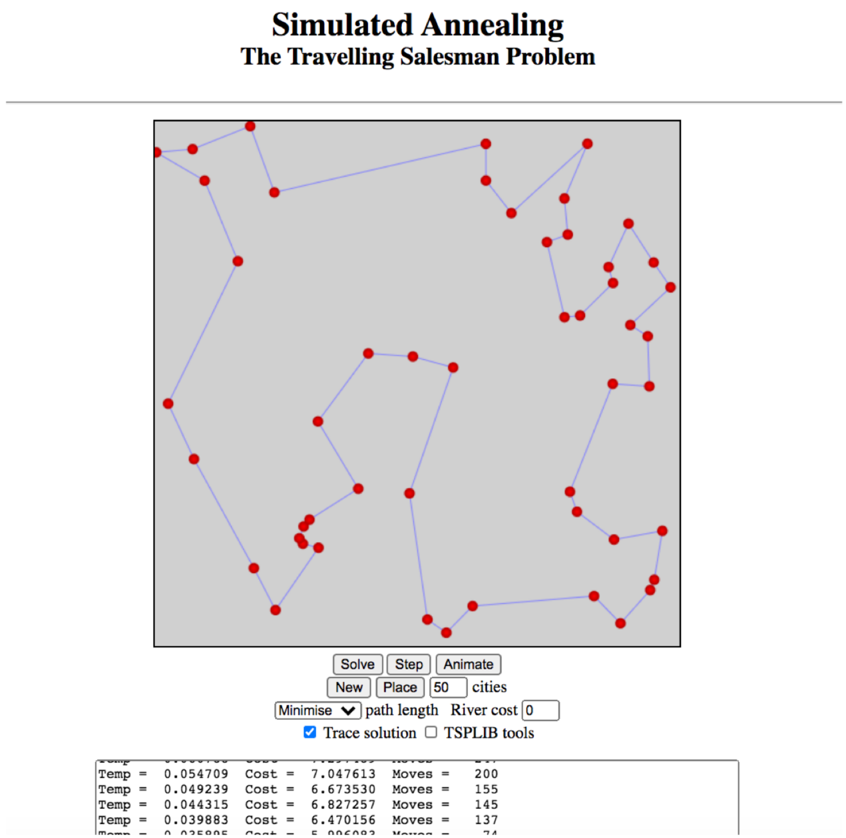
上讲中15个城市旅行商问题



旅行商问题

求解旅行商问题演示

https://www.fourmilab.ch/documents/travelling/anneal/



0-1背包问题

如何考虑约束条件？

$$\max f(x) = \sum_{i=1}^{n} v_i x_i$$

$$\sum_{i=1}^{n} w_i . x_i . \leq P.$$

$$x_i \in \{0,1\}, i = 1,2,\ldots,n$$

0-1背包问题

如何考虑约束条件？

$$\max f(x) = \sum_{i=1}^{n} v_i x_i$$

$$\sum_{i=1}^{n} w_i.x_i. \leq P.$$

$$x_i \in \{0, 1\}, i = 1, 2, \ldots, n$$

将解的质量定义为

$$\sum_{i=1}^{n} v_i x_i - \mu \frac{\max\left\{0, \left(\sum_{i=1}^{n} w_i x_i\right) - P\right\}}{P}$$

0-1背包问题

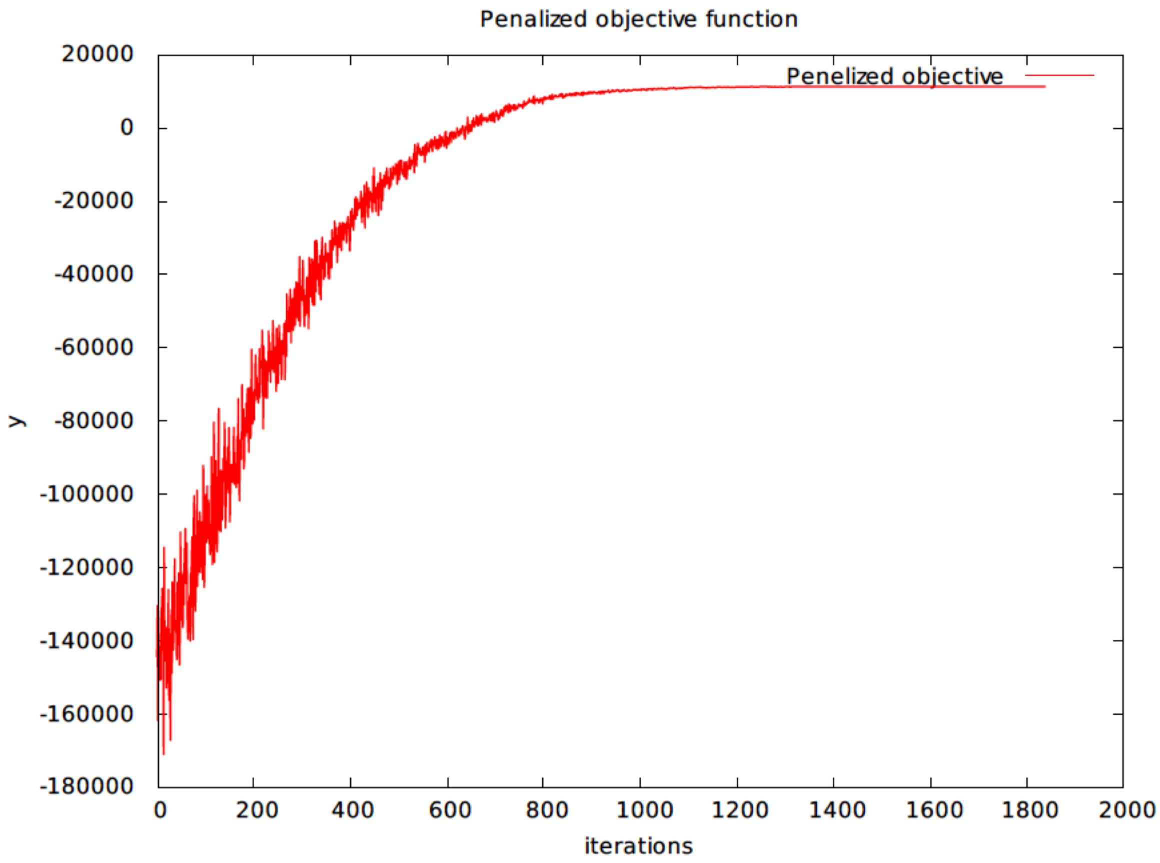
$$\sum_{i=1}^{n} v_i x_i - \mu \frac{\max\left\{0, \left(\sum_{i=1}^{n} w_i x_i\right) - P\right\}}{P}$$

In order to test the simulated annealing algorithm on this problem, we first build an instance of the problem by randomly generating 100 objects for which the weights have also been selected randomly between 1 and 100 with a uniform probability density function. For this instance, the capacity of the bag is set to $P = 2000$. We choose $\mu = 1$ for the penalty parameter and we apply the basic SA algorithm with the initial temperature set to a value of c_0 such that $\chi(c) = 0.8$, a geometric cooling schedule with $\alpha = 0.995$, and $L_k = 1{,}000$ for every iteration k. The algorithm is stopped when the temperature reaches $\frac{c_o}{1{,}000}$.

例子取自Delahaye, et al, "Simulated annealing: From basics to applications"

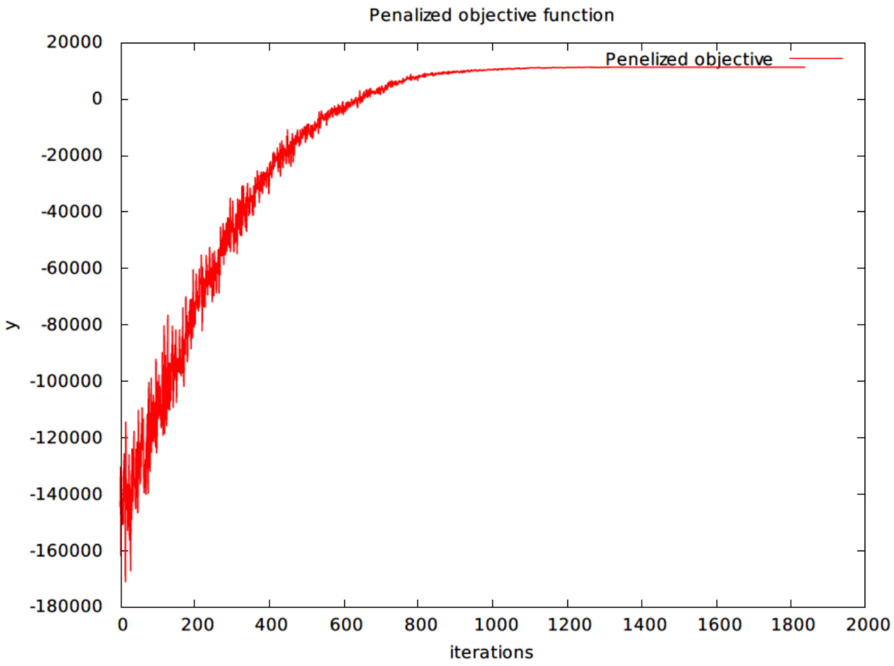
0-1背包问题

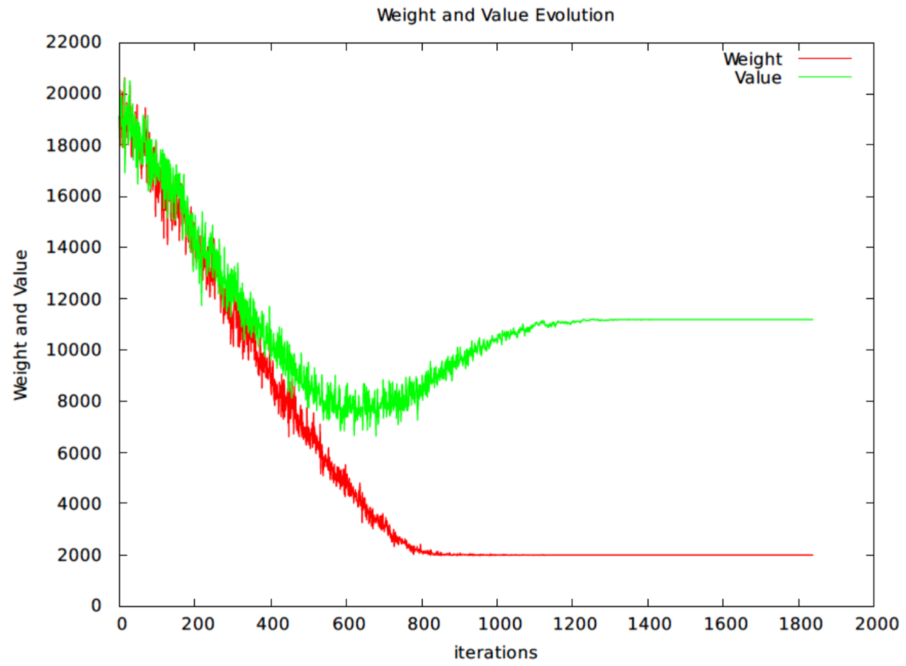
$$\sum_{i=1}^{n} v_i x_i - \mu \frac{\max\left\{0, \left(\sum_{i=1}^{n} w_i x_i\right) - P\right\}}{P}$$



0-1背包问题

$$\sum_{i=1}^{n} v_i x_i - \mu \frac{\max\left\{0, \left(\sum_{i=1}^{n} w_i x_i\right) - P\right\}}{P}$$





背包承重上限是2000

本讲小结

- 模拟退火算法

主要参考资料

为汉科技《回火、淬火、正火、退火》视频

Delahaye, et al <Simulated annealing: From basics to applications>

K. A. Dowsland, J. M. Thompson <Simulated Annealing - Handbook of Natural Computing>

Scott Hauck <UW EE 541: Automated Layout of Integrated Circuits>

Thomas Stutzle <Heuristic Optimization>

谢谢！