## 退火算法

模拟退火算法来源于固体退火原理，是一种基于概率的算法，将固体加温至充分高，再让其徐徐冷却，加温时，固体内部[粒子](https://baike.baidu.com/item/%E7%B2%92%E5%AD%90" \t "_blank)随[温升](https://baike.baidu.com/item/%E6%B8%A9%E5%8D%87" \t "_blank)变为无序状，内能增大，而徐徐冷却时粒子渐趋有序，在每个温度都达到[平衡态](https://baike.baidu.com/item/%E5%B9%B3%E8%A1%A1%E6%80%81" \t "_blank)，最后在[常温](https://baike.baidu.com/item/%E5%B8%B8%E6%B8%A9" \t "_blank)时达到基态，内能减为最小。

模拟退火的基本思想:

(1) 初始化：初始温度T([充分大](https://baike.baidu.com/item/%E5%85%85%E5%88%86%E5%A4%A7" \t "_blank))，初始解状态S(是算法迭代的起点)，每个T值的迭代次数L

(2) 对k=1, …, L做第(3)至第6步：

(3) 产生新解S′

(4) 计算增量ΔT=C(S′)-C(S)，其中C(S)为[评价函数](https://baike.baidu.com/item/%E8%AF%84%E4%BB%B7%E5%87%BD%E6%95%B0" \t "_blank)

(5) 若ΔT<0则接受S′作为新的当前解，否则以概率exp(-ΔT/T)接受S′作为新的当前解.

(6) 如果满足终止条件则输出当前解作为[最优解](https://baike.baidu.com/item/%E6%9C%80%E4%BC%98%E8%A7%A3" \t "_blank)，结束程序。

终止条件通常取为连续若干个新解都没有被接受时终止算法。

(7) T逐渐减少，且T->0，然后转第2步。

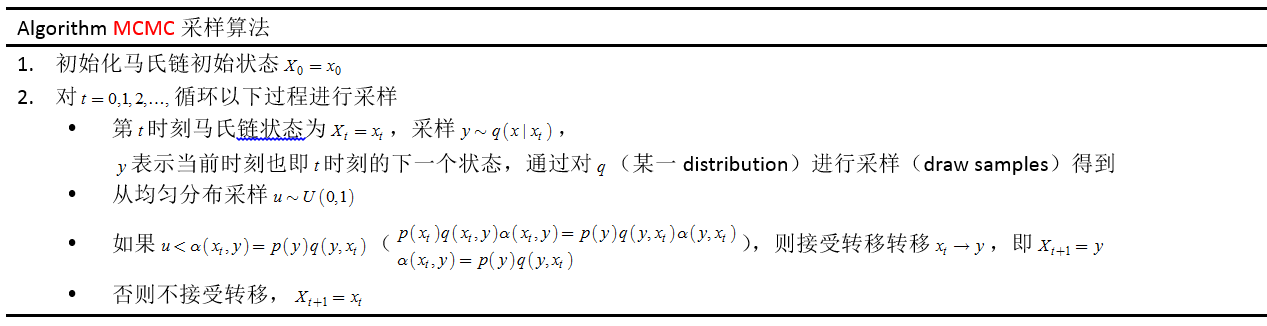
模拟退火算法具有跳出局部最优陷阱的能力。在Boltzmann机中，即使系统落入了局部最优的陷阱，经过一段时间后，它还能再跳出来，再系统最终将往全局最优值的方向收敛。

## MCMC方法

MCMC就是一种通过在概率空间中随机采样来近似感兴趣参数的后验分布的方法

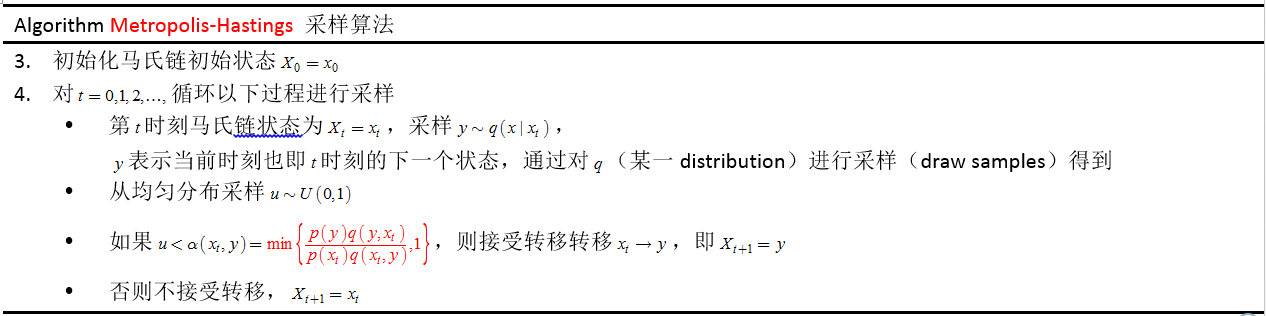
MCMC基本思想：

既然马尔科夫链可以收敛于一个平稳分布，如果这个分布恰好是我们需要采样的分布，那么当马尔科夫链在第n步收敛之后，其后续不断生成的序列X(n), X(n+1), ... 就可以当做是采样的样本。



**Metropolis Hastings算法**:

接受率a如果偏小，那么马尔科夫链很容易拒绝跳转，导致收敛速度慢。实际上我们可以把a放大，只要保证对任意i,j，max(a(i,j), a(j,i))=1即可。



**Gibbs采样：**

应用在至少二维的数据上，也就是高维数据的MCMC采样。

**注意：**

要将状态转移矩阵的每一行都设计成与输入概率分布相一致，这样该算法才能得到更好的结果。

Mcmc从均匀分布中采样，每次都是独立的随机过程。

## 弱人工智能和强人工智能

弱人工智能是指不能制造出真正地推理（Reasoning）和解决问（Problem\_solving）的智能机器，这些机器只不过看起来像是智能的，但是并不真正拥有智能，也不会有自主意识。

强人工智能观点认为有可能制造出真正能推理和解决问题的智能机器，并且，这样的机器能将被认为是有知觉的，有自我意识的。强人工智能可以有两类：类人的人工智能，即机器的思考和推理就像人的思维一样。非类人的人工智能，即机器产生了和人完全不一样的知觉和意识，使用和人完全不一样的推理方式。

## JGAP java的遗传算法库

官网：<https://sourceforge.net/projects/jgap/>

## GAFT python遗传算法库

官网：<https://github.com/PytLab/gaft>