《计算机模拟》

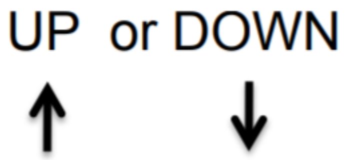
第9讲 – Ising模型与模拟退火

胡贤良
浙江大学数学科学学院

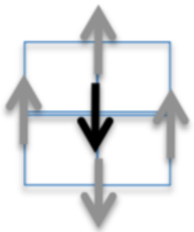
1. 统计物理与Ising模型

微观粒子的相互作用

- 自旋(Spin)



- 近临相互作用(Interaction)



- 相互作用势

$$E_i = -J\sigma_i\sigma_j$$

其中，J 是相互作用的强度系数，

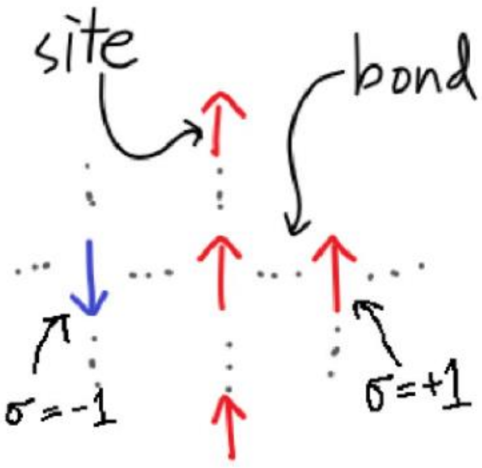
$\sigma_i, \sigma_j = \{-1↓, +1↑\}$表示自旋状态。

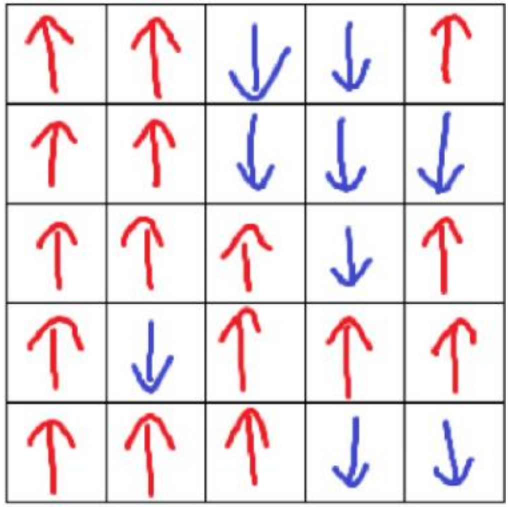
- 粒子i与周围环境的交互能

外部磁场h

$$\epsilon_i = -J \sum_{j=\{nn\}} \sigma_i\sigma_j \boxed{- h\sigma_i}$$

- （几个例子）





正则系综(Canonical Ensemble)

1. 系综(Ensemble): 处于相同宏观条件下的大量结构完全相同的系统的集合。能量

$$E=\sum_{i=1}^{n}\epsilon_i\,.$$

2. 系综是统计力学的一种假想工具。系综的微观态数K也是由各个部分的微观态K_i确定的:

$$K=\prod_{i=1,2,\cdots,n}K_i$$

- 满足上述两个条件的系综被称为：正则(Canonical)系综。正如人们用理想气体状态方程研究空气动力学问题，正则系综也是一种理想状态假设：与环境只交换能量，不交换物质。
- 熵

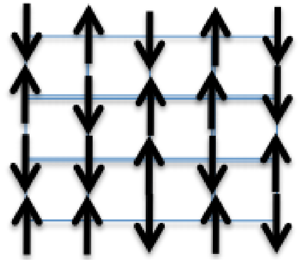
$$S=k_B\ln(K_1K_2)=k_B(\ln K_1+\ln K_2)\coloneqq S_1+S_2$$

给定了正则系综后，假设我们只知道系统总能量为 E，并设共有 m 个微观状态：ψ_1，$\psi_2, \cdots, \psi_m$。这里 m 是一非常大的数，所以这种离散的情况甚至允许数学上被近似地看成是连续的。但我们并不知道哪些系统处在微观态 ψ_i，应该问的问题是：一个系统处在微观态 ψ_i 上的概率是多少？

令 n_i 表示处在微观态 ψ_i 上的系统数，E_i 表示第 i 微观态 ψ_i 的能量。于是，$\{n_1, n_2, \cdots, n_m\}$ 就是系综的构型 (即频数)。当然，它们满足下面关系式：

$$\sum_{i=1}^{m} n_i = N \qquad \text{(系综的系统总数)}, \tag{6.19}$$

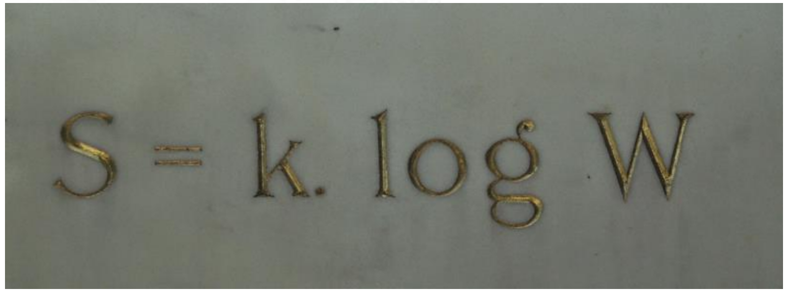
$$\sum_{i=1}^{m} n_i E_i = E \qquad \text{(系综的总能量)。} \tag{6.20}$$



$$m = (N_\uparrow - N_\downarrow)/N$$

即使我们知道了总能量 E 和系统总数 N，并且给定了构型 $\{n_i\}$，但是各系统的状态仍然没有完全确定。例如，已知有 2 个系统在 ψ_1 态，3 个系统在 ψ_2 态，但是到底哪 2 个系统在 ψ_1 态，哪 3 个系统在 ψ_2 态，还是不确定的。

$$\frac{-S}{\boxed{k_B N}} = f_\uparrow \log f_\uparrow + (1 - f_\uparrow) \log(1 - f_\uparrow)$$

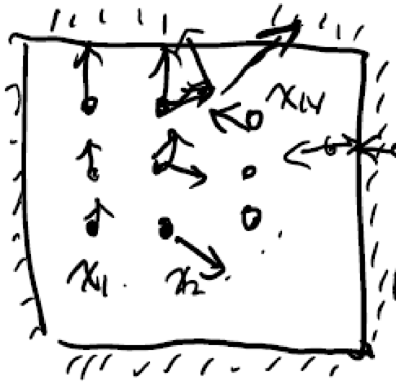


N: $n_1, n_2, \ldots, n_m$

(1) $K = \dfrac{N!}{n_1!\, n_2! \cdots n_m!}$

(2). stirling公式: $\ln n! \approx n \ln n - n$

$$\boxed{\begin{aligned} &\max_{n_i} S = k_B \ln K \\ &\text{s.t.} \quad \sum_{i=1}^{m} n_i = N \\ &\qquad\ \ \sum_{i=1}^{m} n_i E_i = E \end{aligned}}$$



微观状态.

N个粒子无交换

$$\mathcal{L}(\{n_i\}_{i=1}^{m}, \alpha, \beta) = -k_B \ln K + \alpha\left(\sum_{i=1}^{m} n_i - N\right) + \beta\left(\sum_{i=1}^{m} n_i E_i - E\right)$$

$$\frac{\partial \mathcal{L}}{\partial n_i} = 0 \Rightarrow k_B \ln n_i - k_B - \alpha - \beta E_i = 0$$

$$\therefore\ n_i = e^{\left(-1-\frac{\alpha}{k_B}\right) - \frac{\beta}{k_B} E_i}$$

$$= e^{-1-\frac{\alpha}{k_B}} \cdot e^{-\frac{\beta}{k_B} E_i}$$

$$\boxed{\text{故}\quad n_i \sim e^{-\frac{\beta}{k_B} E_i}}$$

可验证: $\beta = \dfrac{1}{T} \triangleq \dfrac{\partial S}{\partial E}$.

E.

故: $\boxed{P(x) = \dfrac{1}{Z} e^{\frac{-E(x)}{k_B T}}}$

玻尔兹曼分布 - 统计物理学的理论基石

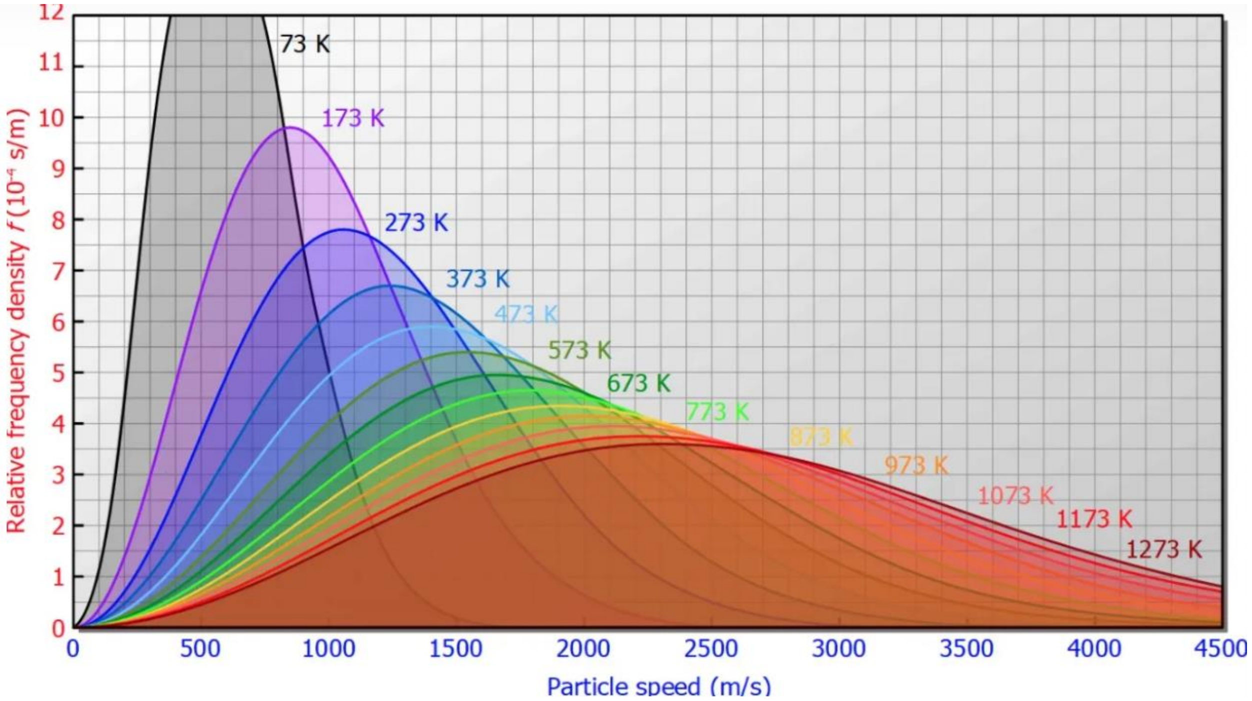
- 在正则系综(系统总数N和总能量E固定)中，系统处在第i微观态ψ_i上的概率（可由统计力学的最大熵假设和函数极值的必要/KKT条件演算得到）可表示为

$$p_i = \frac{1}{Z} e^{-\frac{E_i}{k_B T}},$$

其中，Z是归一化因子，称为配分函数:

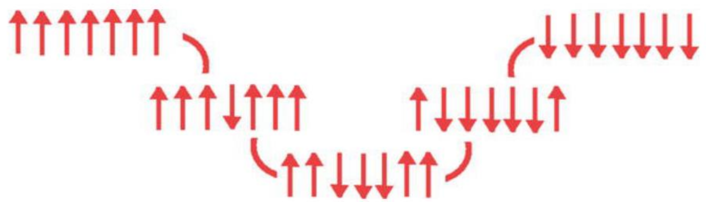
$$Z = \sum_{i=1}^{n} e^{-\frac{E_i}{k_B T}}.$$

- 事实上，p_i已被归一化！它是一个概率分布，即所谓的玻尔兹曼(Boltzmann)分布。



宏观可测量

- 各个粒子的自旋状态经过大量随机转换之后达到一个平衡态。 Ising模型所描述的铁磁体的磁矩量，可由各个粒子自旋方向的统计量之平均



- 统计力学中，统计平均量(宏观可测的物理量A，如温度、磁矩等)即为微观状态A_x在状态空间Ω内的Boltzmann分布期望：

$$\langle A \rangle_T = \sum_{x \in \Omega} A_x \frac{e^{-\frac{E_x}{k_B T}}}{Z}$$

- 对于连续物质系统， 考察某个Hamiltonian量H（如能量E或者磁场M），计算宏观量

$$\langle A \rangle_T = \frac{1}{Z} \int_{\Omega} A(x)\, e^{-\frac{H(x)}{k_B T}}\, dx,$$

这里，$Z = \int_{\Omega} e^{-\frac{H(x)}{k_B T}} dx$是归一化因子，A(x)是连续的微观量函数。

玻尔兹曼(Ludwig Edward Boltzmann)

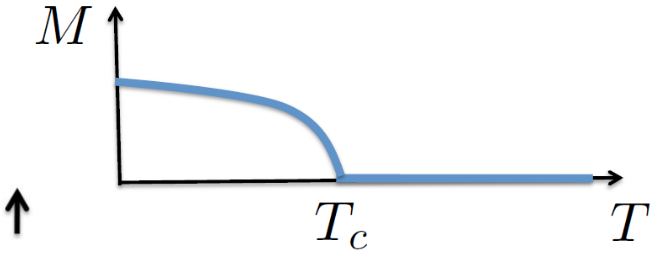
热力学和**统计物理学**的奠基人之一。

- 1844：出生于奥地利**维也纳**
- 1866：获得**维也纳大学**博士学位
- 1869年：将麦克斯韦速度分布律推广到保守力场作用下的情况，得到了**玻尔兹曼分布律**
- 1872年：玻尔兹曼建立了**玻尔兹曼方程**（又称**输运方程**）
- 1877年：提出了著名的**玻尔兹曼熵**
- 1906年：自杀身亡，维也纳中央公墓。

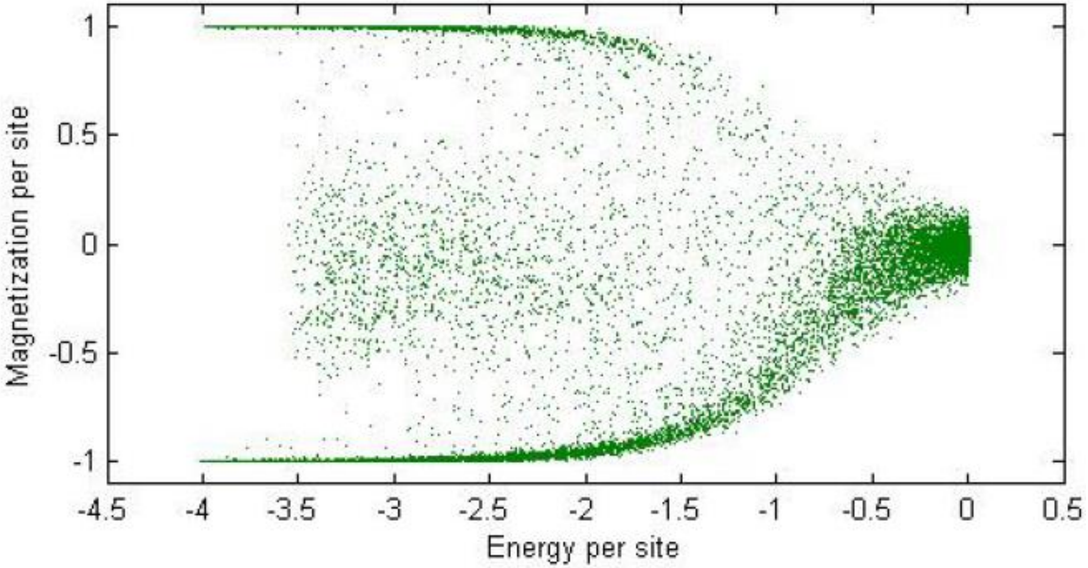
磁性(Magnetism)与相变理论

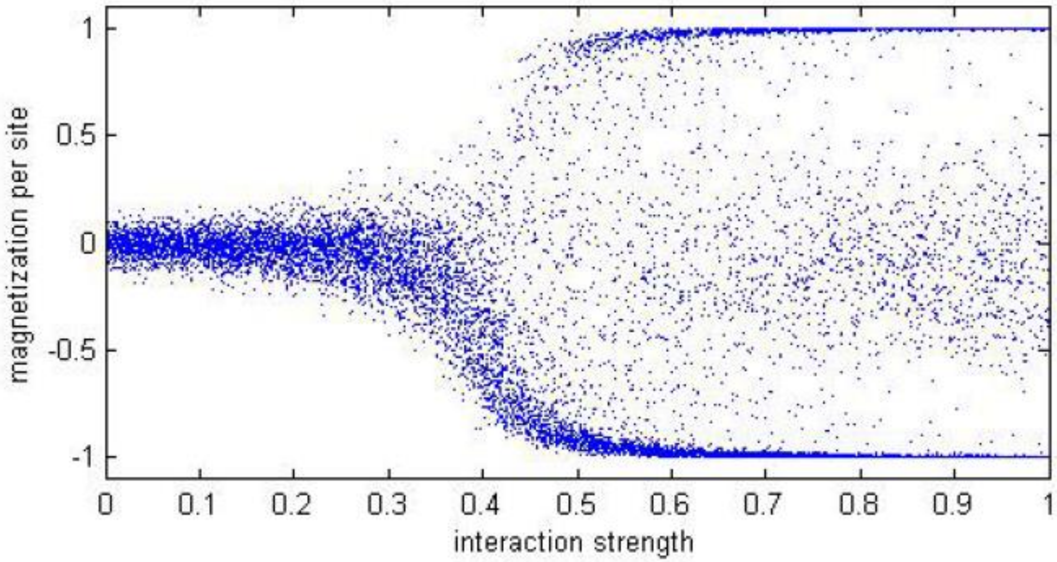
- 考虑

$$J = \begin{cases} > 0 & \text{ferromagnetic} \quad \uparrow\uparrow\uparrow\uparrow\uparrow\uparrow \\ < 0 & \text{antiferromagnetic} \quad \downarrow\uparrow\downarrow\uparrow\downarrow\uparrow \end{cases}$$



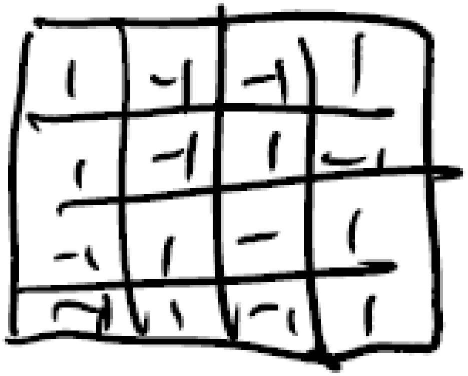
- 希望研究磁性（$J > 0,\ -\frac{E}{kT}$）在新材料研究方面的应用



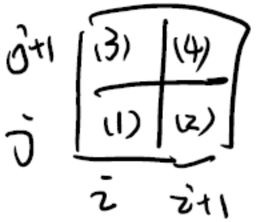


Ising Model： Mean-Field Theory

- 是对相变现象的数学建模。只有二维以上才能模拟相变!



定义: 令邻域:



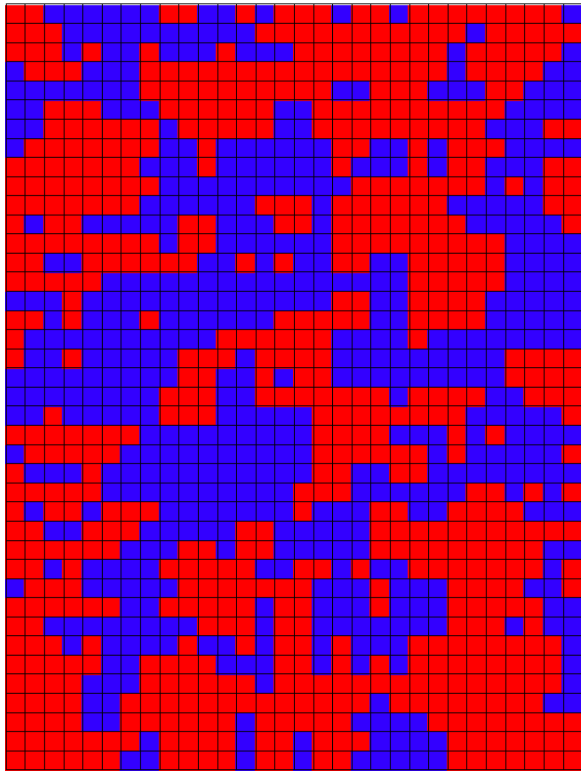
$$\Lambda(i,j)=\{(i,j),\ (i+1,j),\ (i,j+1),\ (i+1,j+1)\}.$$

定义能量: $$E(S)=-\frac{1}{2}J\sum_{i,j}\underbrace{\left(S_iS_j+S_{i+1}S_j+S_iS_{j+1}+S_{i+1}S_{j+1}\right)}_{\Lambda(i,j)\text{自旋相互作用势能. 卷积.}} \quad (1)$$

定义磁性: $$M(S)=\frac{1}{N^2}\sum_{i,j}S_{i,j}$$

总磁性 $$\langle M\rangle=\sum_{S\in\Omega}M(S)\cdot p(S)=\underbrace{\boxed{\frac{1}{Z}}}_{\text{无法求出}}\sum_{S\in\Omega}M(S)\,e^{-\beta E(S)}$$

记 $\beta=\frac{1}{k_BT}$



Metropolis算法

状态 S_0 给定， T， N， $M=0$

$S = S_0$.

for $k = 1, \cdots, NMAX$.

  计算 $E(S)$ with (1)

  随机选择 (i,j). $S(i,j) = -S(i,j)$； $Y = S$

  计算 $E(Y)$ with (1)

  $$h(S,Y) = \min\left\{1, \frac{p(Y)}{p(S)}\right\} := e^{-\beta(E(Y)-E(S))}$$

  if (uniform rand < $h(S,Y)$). 接受Y.

    $S = Y$； $E(S) = E(Y)$

  end.

  $M = M + M(S)/NMAX$.

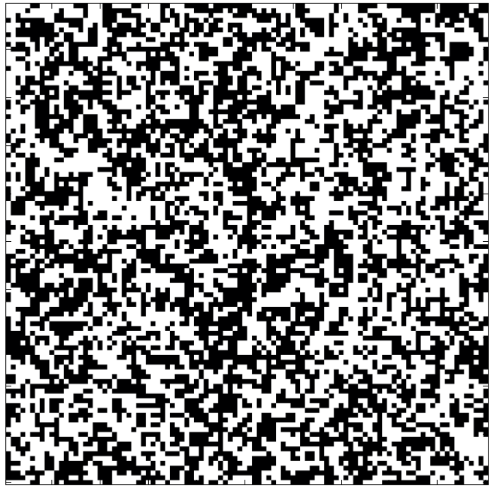
end.

```
function M = Ising(T)      % T is the tempure.  +-+-+   Ising area.
    nTrials = 100000;      % for stat.          |3|4|   N = 2.
    startup = 10000;      % for steady.         +-+-+   free boundary.
    d = zeros(1, nTrials);    % results.        |1|2|
    beta = 1/T;                 %               +-+-+
    M = 0;
    s = 2 * round(rand(1, 4)) - 1;    % randomly set 1 or -1.
    Es = s(1) * s(2) + s(3) * s(4);
    Es = Es + s(1) * s(3) + s(2) * s(4);
    Es = -Es;
    for t = 1 : (startup + nTrials)
        k = fix(1 + 4 * rand);  % randomly pick 1 ~ 4.
        y = s;              % y is the suggestion dist in s' neighbour.
        y(k) = -s(k);     % randomly flip the status on one point.
        Ey = y(1) * y(2) + y(3) * y(4);
        Ey = Ey + y(1) * y(3) + y(2) * y(4);
        Ey = -Ey;
        h = min(1, exp(-beta * (Ey - Es)));
        if (rand < h)
            s = y;
        end
        if (t > startup)
            Ms = s(1) + s(2) + s(3) + s(4);    % cheating!
            M = M + Ms;                        % this is not Boltzmanr
            d(t - startup) = Ms;
        end
        if(mod(t, 1000)==0) fprintf('Round %d ...\n', t);
    end
    x = -4:1:4;
    hist(d, x);
    M = (M / nTrials) / 4;
end
```

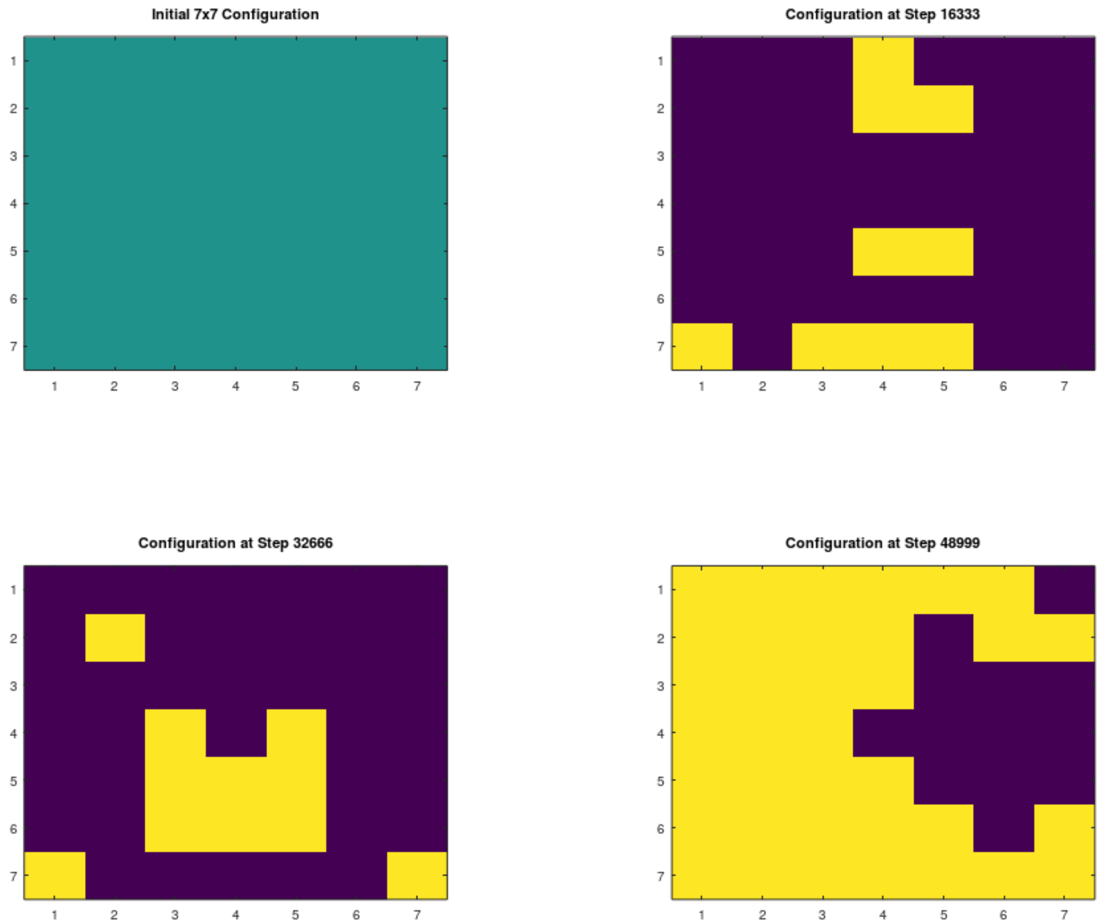
- 即应用Metropolis算法产生渐进收敛于Boltzmann分布的抽样！

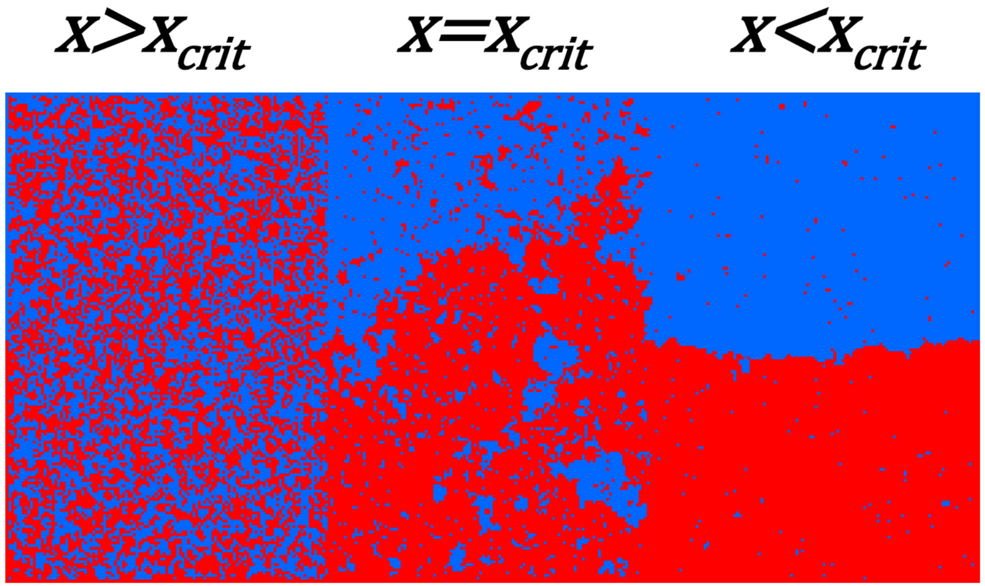
一些模拟结果参考(demo_Ising.html)

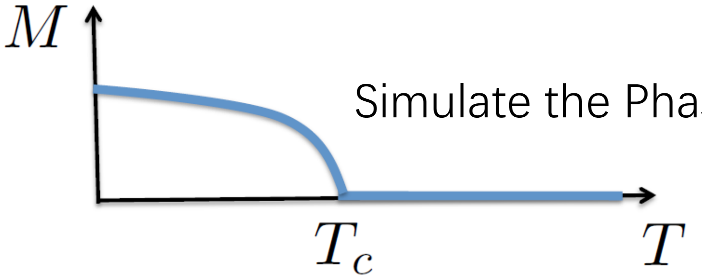
T = 1.00, M = −0.04, E = −3.61



T = 5.00, M = 0.03, E = −0.81



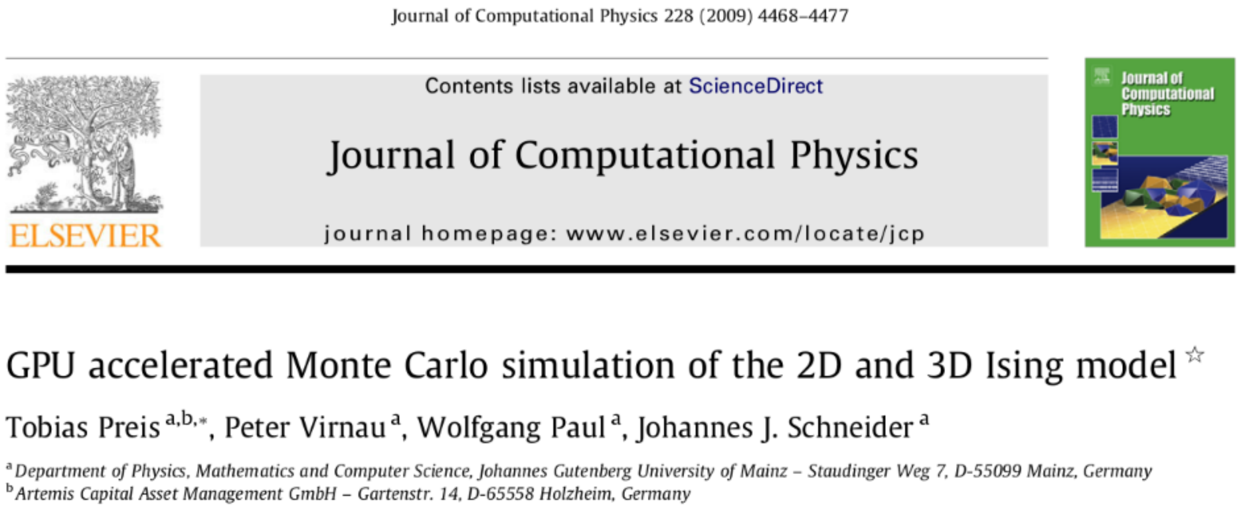




Simulate the Phase Transition via Ising Model

GPU 加速

1. 论文


Journal of Computational Physics 228 (2009) 4468–4477

Contents lists available at ScienceDirect

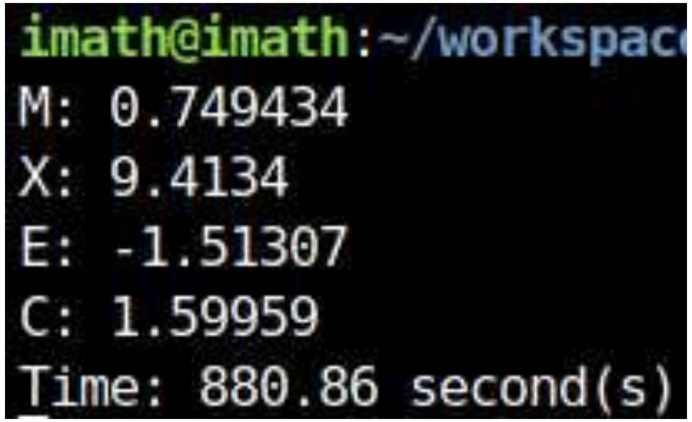
Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

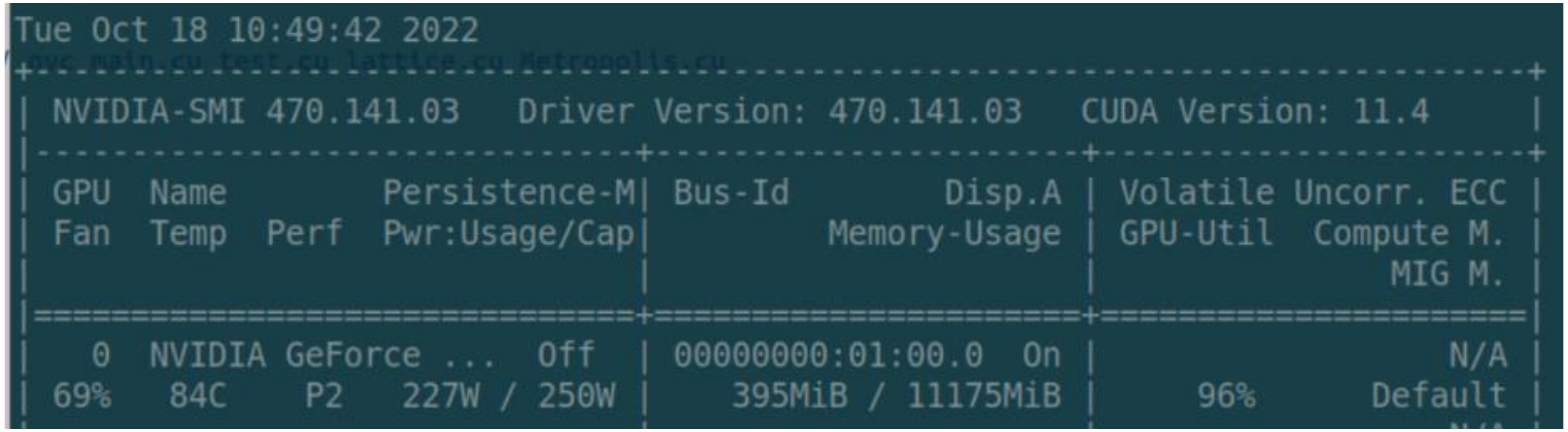
GPU accelerated Monte Carlo simulation of the 2D and 3D Ising model ☆

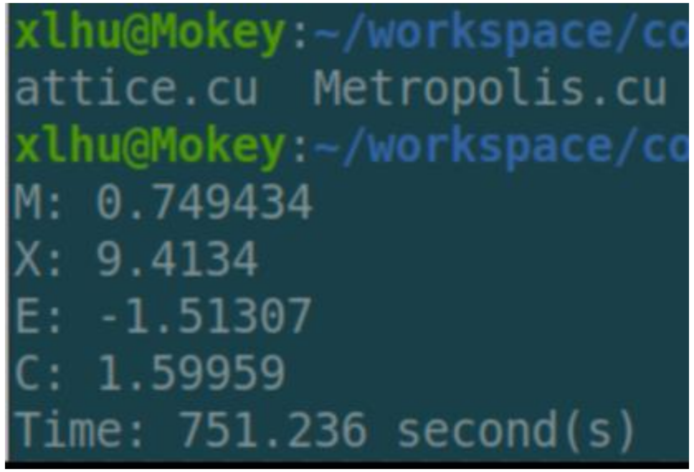
Tobias Preis [a,b,*], Peter Virnau [a], Wolfgang Paul [a], Johannes J. Schneider [a]

[a] Department of Physics, Mathematics and Computer Science, Johannes Gutenberg University of Mainz – Staudinger Weg 7, D-55099 Mainz, Germany
[b] Artemis Capital Asset Management GmbH – Gartenstr. 14, D-65558 Holzheim, Germany



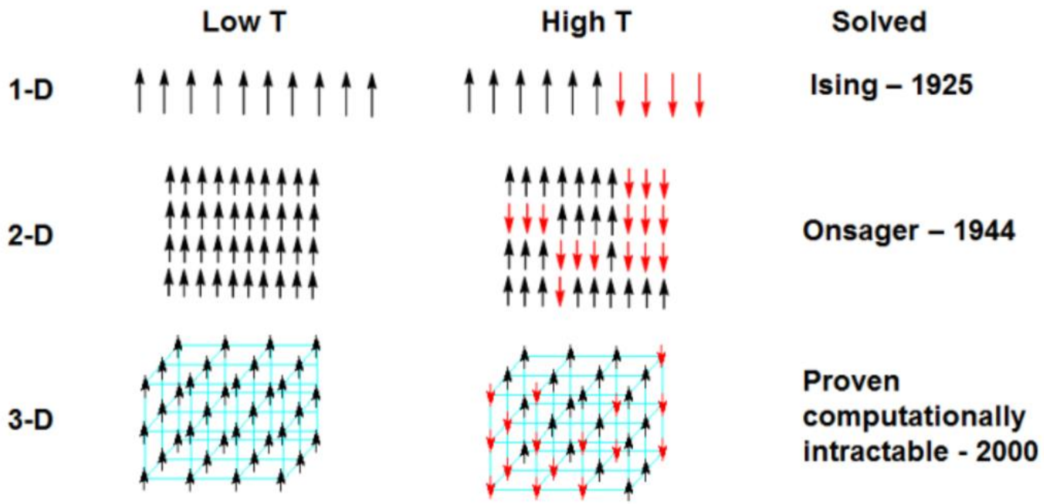
2. https://github.com/megumi-ovo/Ising-GPU

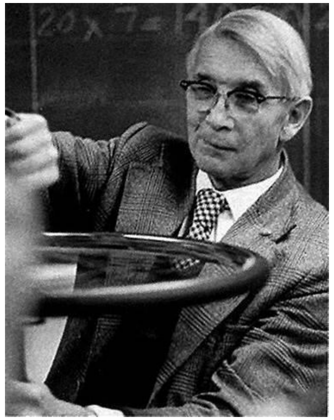




History : the Ising Model

- Pierre Curie 1895年发现磁性的”居里点”
- 1920年由德国物理学家Wilhelm Lenz提出，用于描述铁磁性物质的内部的原子自旋状态及其与宏观磁矩的关系。
- 1925年，他的学生Ernst Ising求解了不包含相变的一维伊辛模型
- 1944年，美国物理学家**Lars Onsager**得到了二维伊辛模型在没有外磁场时的解析解，即Onsager解。
- 三维…





Lars Onsager

- A series of papers 1944-1950

PHYSICAL REVIEW VOLUME 65, NUMBERS 3 AND 4 FEBRUARY 1 AND 15, 1944

Crystal Statistics. I. A Two-Dimensional Model with an Order-Disorder Transition

LARS ONSAGER
Sterling Chemistry Laboratory, Yale University, New Haven, Connecticut
(Received October 4, 1943)

PHYSICAL REVIEW VOLUME 76, NUMBER 8 OCTOBER 15, 1949

Crystal Statistics. II. Partition Function Evaluated by Spinor Analysis

BRURIA KAUFMAN*
Columbia University, New York City, New York
(Received May 11, 1949)

- 1931年，昂萨格在《物理学评论》杂志发表了“不可逆过程的倒易关系”的著名论文，为不可逆过程热力学的建立做出了卓越贡献，因此文昂萨格荣获1968年诺贝尔化学奖
- 1932年，昂萨格将此工作提交挪威诺尔格斯工学院，申请授予博士学位，但该校认为此文不符合学位论文的要求，而不同意授予博士学位
- 1933年被耶鲁大学批准担任博士后研究人员。提交了一篇以数学为主题的论文。耶鲁大学的化学和物理教授均感到无法对此论文做出评价。最后在数学系教授的竭力推荐下，耶鲁大学化学系决定授予昂萨格博士学位。

REVIEW ARTICLE

Front. Phys., 12 February 2014 | https://doi.org/10.3389/fphy.2014.00005

Ising formulations of many NP problems

Andrew Lucas*
Lyman Laboratory of Physics, Department of Physics, Harvard University, Cambridge, MA, USA

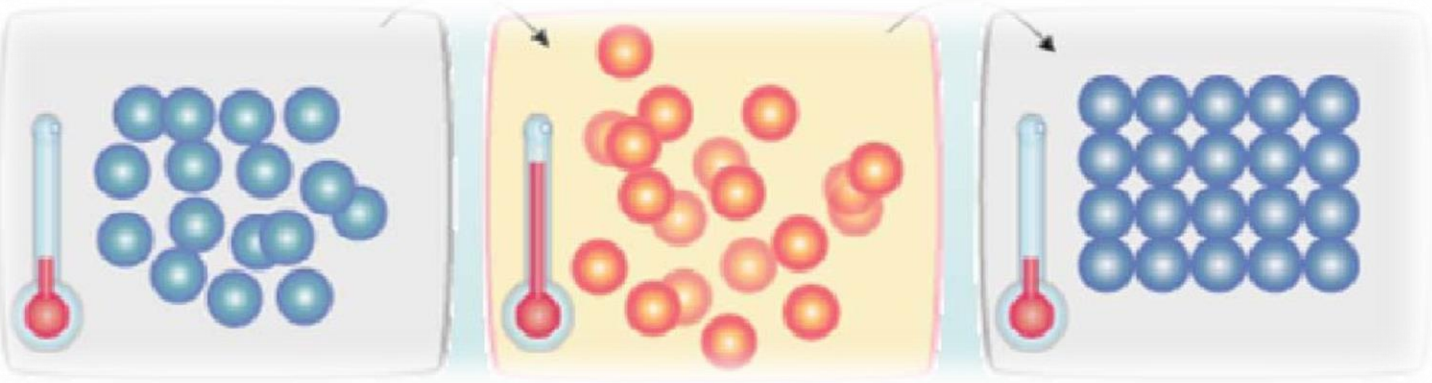
We provide Ising formulations for many NP-complete and NP-hard problems, includin

Monte Carlo Simulation in Statistical Physics

直接追踪粒子，物理思路清晰，易于理解；程序结构清晰简单；其它优点包括：

1. 采用随机抽样的方法，较真切的模拟粒子输运的过程，反映了统计涨落的规律。
2. 不受系统多维、多因素等复杂性限制，是解决复杂系统粒子输运问题的好方法。
3. 研究人员采用MC方法编写程序来解决粒子输运问题，比较容易得到自己想得到的任意中间结果，应用灵活性强。
4. MC方法主要弱点是收敛速度较慢和误差的概率性质，如果单纯以增大抽样粒子个数N来减小误差，就要增加很大的计算量。

2. 模拟退火



模拟退火法(Simulation Annealing)

Scott Kirkpatrick

School of Computer Science and Engineering

The Hebrew University of Jerusalem （希伯来大学）

- 1959-1963 普林斯顿大学 物理学学士
- 1963-1968 哈佛大学 物理学博士
- S. Kirkpatrick, C. Gelatt, M. Vecchi: **Optimization by simulated annealing**， Science, 220 (1983), pp. 671-680.

➢ Independently by Cerny : **Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm**, Journal of Optimization Theory and Applications, 45 (1985), pp. 41-51.

例1：Shekel函数的最大值问题

Shekel函数： $f(x) = \sum_{i=1}^{m} \frac{1}{a_i(x-b_i)^2+c_i}$

```
function y = shekel(x)
   y = 1 ./(10.295*(x-7.382).^2 + 0.1613) + 1 ./(4.722*(x-1.972).^2+0.2327);
end
```

其中a_i, b_i, c_i为常数且要求$a_i c_i > 0, i = 1, \ldots, m$，而函数中的m不宜超过30。当$a_i, c_i > 0$时，该函数$f > 0$,它的各个峰值分别位于点$b_i, i = 1,2, \ldots, m$，并且$a_i$是相应的衰变率。

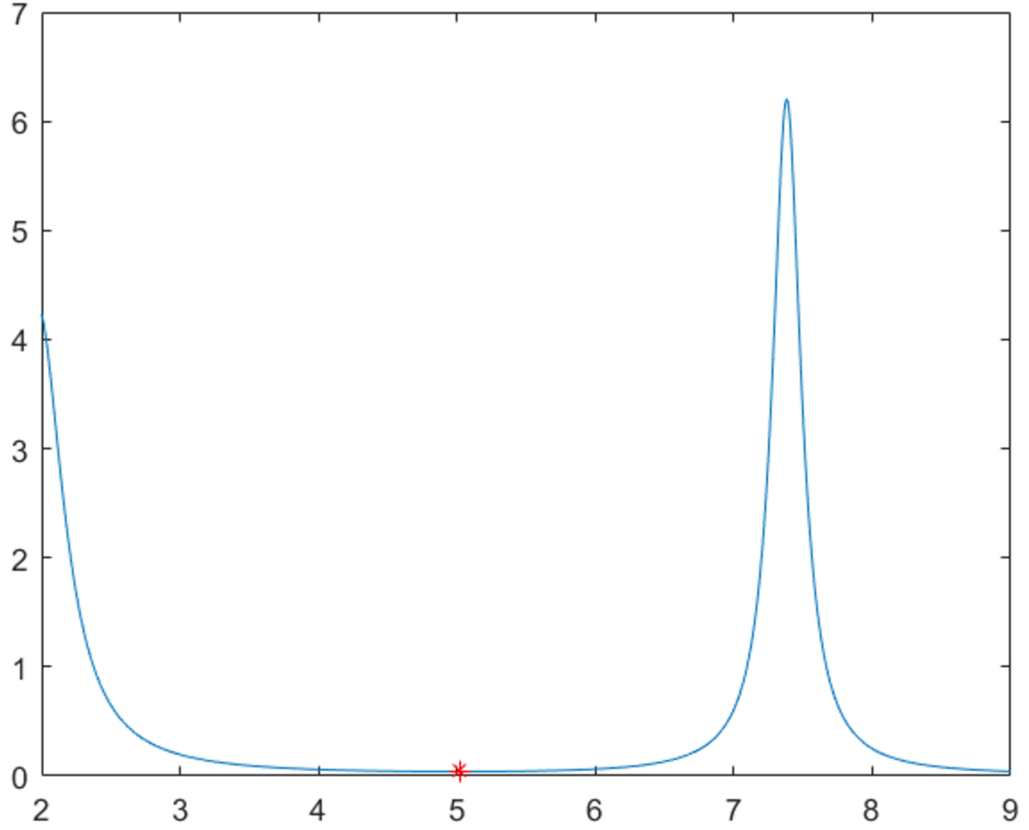
利用MATLAB工具箱得模拟退火法:

```
minx = 2;   maxx = 9;
xx = minx:0.01:maxx;
yy = shekel(xx);

x = simulannealbnd(@shekel, 2.1, 2, 9);

plot(xx, yy, ‘b-’, x, shekel(x), 'r*');
```

求得如右图的最小值。



模拟退火与Metropolis准则

源自于统计力学(Statistical mechanics):

假设优化问题的能量函数为$f(x)$,其状态概率服从玻尔兹曼分布，即概率密度为

$$\frac{1}{Z}e^{-\frac{f(x)}{k_B T}},$$

其中，Z是归一化因子（配分函数）。考虑用随机的方式产生一个建议状态y。那么，从当前x到新状态y的接受概率由下式给出：

$$h(x,y)=\min\left\{1, e^{-\frac{f(y)-f(x)}{k_B T}}\right\}$$

➢ 实际是导入了一种状态转移机制。

SA

Choose, at random, an initial solution s for the system to be optimized
Initialize the temperature T
while the stopping criterion is not satisfied **do**
    repeat
        Randomly select $s' \in N(s)$
        if $f(s') \leq f(s)$ **then**
            $s \leftarrow s'$
        else
            $s \leftarrow s'$ with a probability $p(T, f(s'), f(s))$
        end
    until the "thermodynamic equilibrium" of the system is reached
    Decrease T
end
return the best solution met

The algorithm starts by generating an initial solution (either randomly or constructed using an heuristic) and by initializing the temperature parameter T. Then, at each iteration, a solution s' is randomly selected in the neighborhood $N(s)$ of the current solution s. The solution s' is accepted as new current solution depending on T and on the values of the objective function for s' and s, denoted by $f(s')$ and $f(s)$, respectively. If $f(s') \leqslant f(s)$, then s' is accepted and it replaces s. On the other hand, if $f(s') > f(s)$, s' can also be accepted, with a probability $p(T,f(s'),f(s)) = \exp\left(-\frac{f(s')-f(s)}{T}\right)$. The temperature T is decreased during the search process, thus at the beginning of the search, the probability of accepting deteriorating moves is high and it gradually decreases. The high level SA algorithm

https://www.sciencedirect.com/science/article/pii/S0020025513001588

关于模拟退火不得不说的几个要素

模拟退火法的基本思想，来源于统计力学：一个热系统可通过退火过程达到内部能量最小的状态

核心技术 - Metropolis算法

$$h(x,y) = \min\left\{1, e^{-\frac{f(y)-f(x)}{k_B T}}\right\}$$

关键点 - 冷却进度表(即T的选择)

序列成为马尔可夫链的两个条件

不可约转移性：对于Ω中任意两个不同的状态x和y，只需经过有限次总能从x到达y。

对称性：转移概率g满足 $g(x|y) = g(y|x), \forall x, y$。

- 冷却进度表：降低算法中的虚拟温度T，使系统运行在在有效时间内逼近最优解(平衡

设置开始冷却的初始温度T_0；

给出温度T的衰减函数；确定每次冷却下降多少温度；

在每一温度T下需要迭代的次数：即等温过程下的马尔可夫链长度L_k(如果温度下降足够慢，则这一步可以省略)；

给定冷却的终止温度T_f:冷却的停止点.

冷却的四个原则

1. 开始温度T_0要足够大，通常可以取100至2000

2. 衰减的速率要小，采用较多的有:

   (a)几何冷却方式的衰减函数:

$$T_{k+1} = \alpha T_k, k = 0,1,2, \ldots,$$

   其中，常数$0 < \alpha < 1$; α控制降温的速率，它越大则速率越小，取值范围为$0.5 \sim 0.99$。

   (b)逆对数的冷却方式:

$$T = \frac{c}{a + \ln t}$$

   其中， a和c是常数，这里c通常是一个很大的正数。

3. 合适的链长度

4. 停止温度T_f应设为足够小，经验上为0.01~5的范围内。

收敛性【哈耶克定理】

在不可约及对称转移条件下，如果温度衰减被取为迭代t的函数：

$$\mathrm{T}_t = \frac{c}{\ln(1+t)}, t = 1,2, \dots,$$

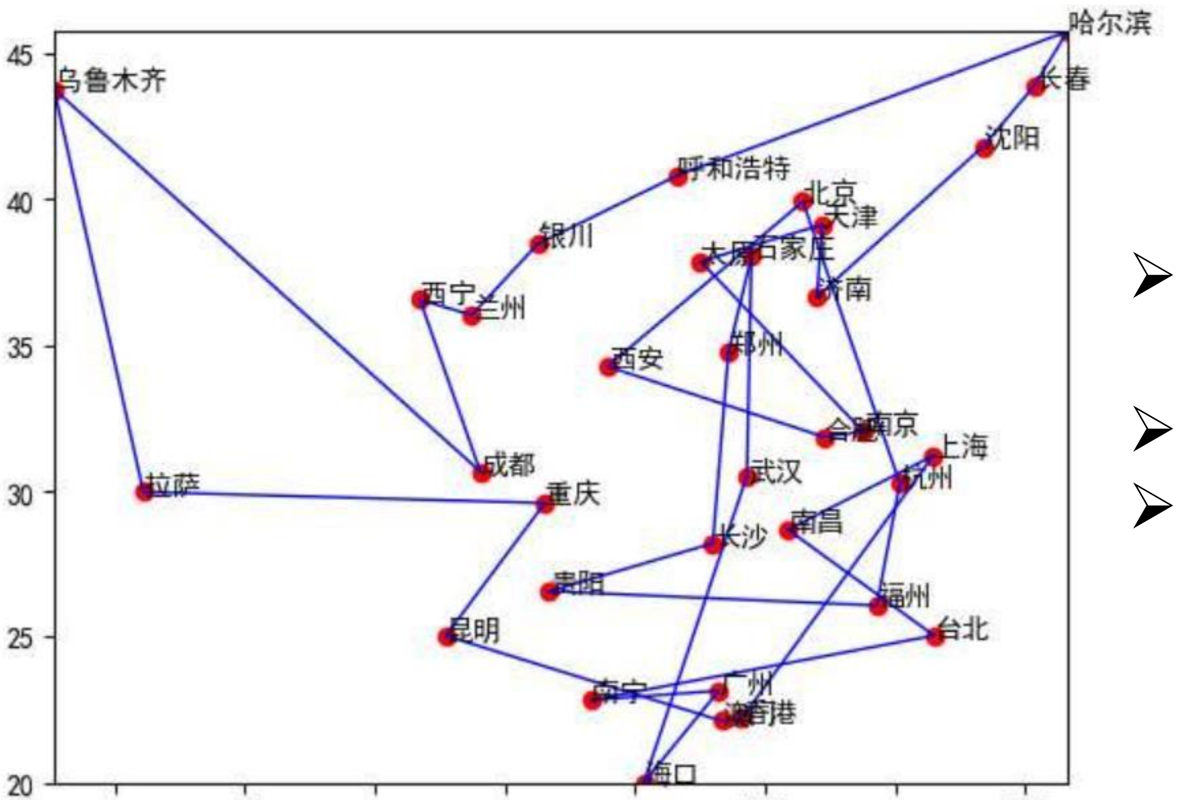
其中c是足够大的常数。则当$t \to \infty$时，模拟退火产生的马尔可夫链X_t将以概率1趋于全局最小解，即

$$\lim_{t \to \infty} P(X_t \in G) = 1$$

Simulated Annealing研究工作概览

- 1983, Kirkpatrick S. etc., Optimization by simulated annealing，**Science**, 220 (1983), 671-680.
- 1984, German S. & German D. 给出**退火率与退火时间的对数成反比**的版本(题名 Equation of state calculation by Fast Computing Machines)
- 1985, Cerny V.运用SA求解Traveling Salesman Problem(**TSP**)
- 1986, Szu H.，提出一种**退火率与时间成反比**的Fast Simulated Annealing(**FSA**)
- 1987, Corana A. 求解多目标连续变量优化问题;
- 同年，Laarhoven P. & Aarts E.出版《模拟退火的理论和应用》
- 1989, Charnes A. & Wolfe M. : 收敛性研究；**Aarts E.结合SA和ANN分析Boltzmann机**
- 1990, Gunte D. & Tobias S.研究SA中**初始温度临界值**的确定方法
- 1992, Laarhoven P. 运用SA解决(NP完全) Jobshop问题; Ingber L.等人提出Very Fast SA
- 1993, Kirkpatrick S. 运用SA到一般优化问题
- 1995, Tarek M. 等人提出并行模拟退火算法，更适用于解决复杂的开学和工程问题
- 1997, Ali & Storey 研究连续变量优化问题，提出**A**spiration based **S**imulated **A**nnealing(**ASA**)以及 **D**irect search **S**imulated **A**nnealing(**DSA**)
- 2001,都志辉等人提出混合SPMD模拟退火算法，取得Scalable并行结果。

例2： 旅行商问题(TSP)



https://zhuanlan.zhihu.com/p/144896129

- 给定n个城市，其地理坐标表示为$C_i, i = 1,2,\ldots,n$,旅行开始城市是C_1,然后按照路径从一个城市到下一个城市，最后回到C_1,每个城市恰好去游玩一次。
- 线路表示为$x = (i_1, i_2, .., i_{n-1})$是{2，3，⋯,n}的置换，其中$i_0 = 1$。
- Ω是所有旅行线路的全体。
- 旅行线路长度为路径的长度，即

$$E(x) = \sum_{k=1}^{n-1} d(C_{i_{k-1}}, C_{i_k}) + d(C_{i_{n-1}}, C_{i_0})$$

其中$d(C_i, C_j)$表示城市C_i到C_j的距离。

- “旅行”的路径选择 ~ $\frac{1}{2}(n+1)!$ 种选择

部分路径逆转策略(Metropolis抽样)

基本想法：随机选择路线序列$x=(i_1,\dots,i_{n-1})$中的两个城市i_{k_1}和i_{k_2}，逆转连接线路的方向得到建议线路y。如将

3-> 5 -> 2 -> 8 -> 4 -> 6-> 7

中随机选中的2和6进行逆转，得到：

3-> 5 -> 6 -> 4 -> 8 -> 2-> 7

如此往复，总是记录路径短的那个!

Initial: 产生$\{2,3,\dots,n\}$的一个随机置换x并计算能量$E(x)$

$$T \leftarrow T(0), B \leftarrow E(x), t \leftarrow 0$$

Loop until $T = T_f$

y←被部分逆转的x的路径

计算$E(y)$，得到 $\Delta E \leftarrow E(y) - E(x)$

$h \leftarrow \min\{1, e^{-\Delta E/T}\}$

产生均匀分布的随机数$r\sim U(0,1)$

if $r < h$ **then**

$x \leftarrow y$; $E(x) \leftarrow E(y)$

end if

$\boldsymbol{B \leftarrow \min\{B, E(x)\}}$

$t \leftarrow t+1$; $T \leftarrow T(t)$

End loop

```
function cityXY = swapcities(cityXY, m)
% cityXY:城市的坐标
% m:被逆转的城市数
S = cityXY;
for i = 1 : m
    city_1 = round(length(cityXY)*rand(1));
    if(city_1 < 1) city_1 = 1; end;
    city_2 = round(length(cityXY)*rand(1));
    if(city_2 < 1) city_2 = 1; end;
    temp = S(:, city_1)
    S(:,city_1) = S(:,city_2)
    S(:,city_2) = temp;
end
```

```
function TSPSA(cityXY, initialTemp, coolRate, maxIter, nSwap)
global iterations;
temp = initialTemp;
iterations = 1;
final_temp_iterations = 0;
while iterations < maxIter
    previous_distance = distance(cityXY);
    pos_cities = swapcities(cityXY, nSwap);
    current_distance = distance(pos_cities);
    diff = abs(current_distance - previous_distance);
    if current_distance < previous_distance
        cityXY = pos_cities;
        plotcities(cityXY);
        if final_temp_iterations >= 10
            temp = coolRate*temp;
            final_temp_iterations = 0;
        end
        nSwap = round(nSwap*exp(-diff/(iterations*temp)));
        if nSwap == 0
            nSwap = 1;
        end
    iterations = iterations + 1;
     final_temp_iterations = final_temp_iterations
else
        if rand(1) < exp(-diff/temp)
            cityXY = pos_cities;
            plotcities(cityXY);
            nSwap = round(nSwap * exp(-diff/(iterations*temp)));
            if nSwap == 0
                nSwap = 1;
            end
            final_temp_iterations = final_temp_iterations + 1;
            iterations = iterations + 1;
        end
    end
    clc
    fprintf('迭代次数= %d', iterations);
    fprintf('最终温度 = %3.8f', temp);
end
```

旅行商问题测试2例

Coord = [66.83 61.95 40 24.39 17.07 22.93 51.71 87.32 68.78 84.88 50 40 25 ;

25.36 26.34 44.39 14.63 22.93 76.1 94.14 65.36 52.19 36.09 30 20 26] ;

initialTemp = 1;   coolRate = 0.99;   nSwap = 3;
maxIter = 2000;

TSPSA(Coord, initialTemp, coolRate, maxIter, nSwap);

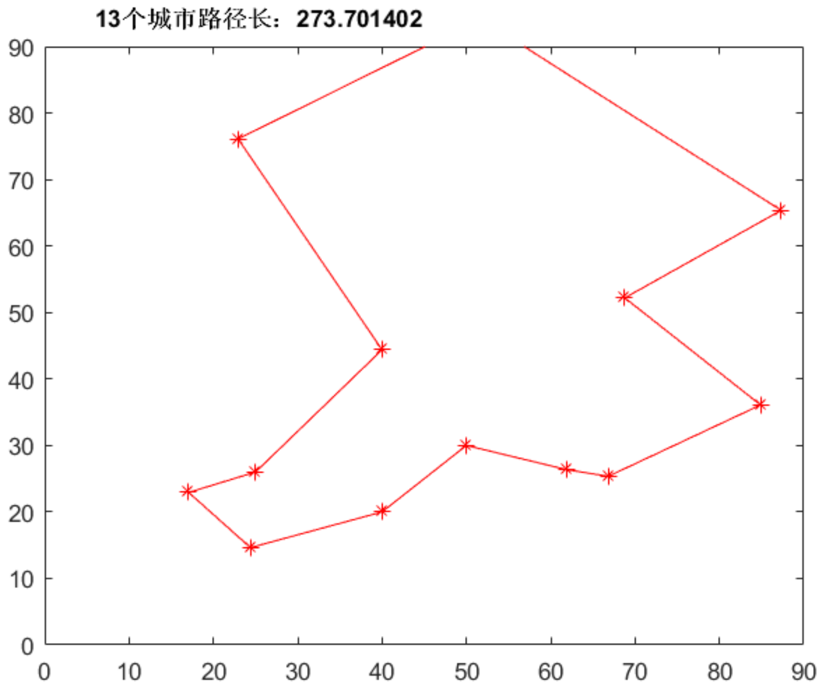
Coord = ···   % city.csv

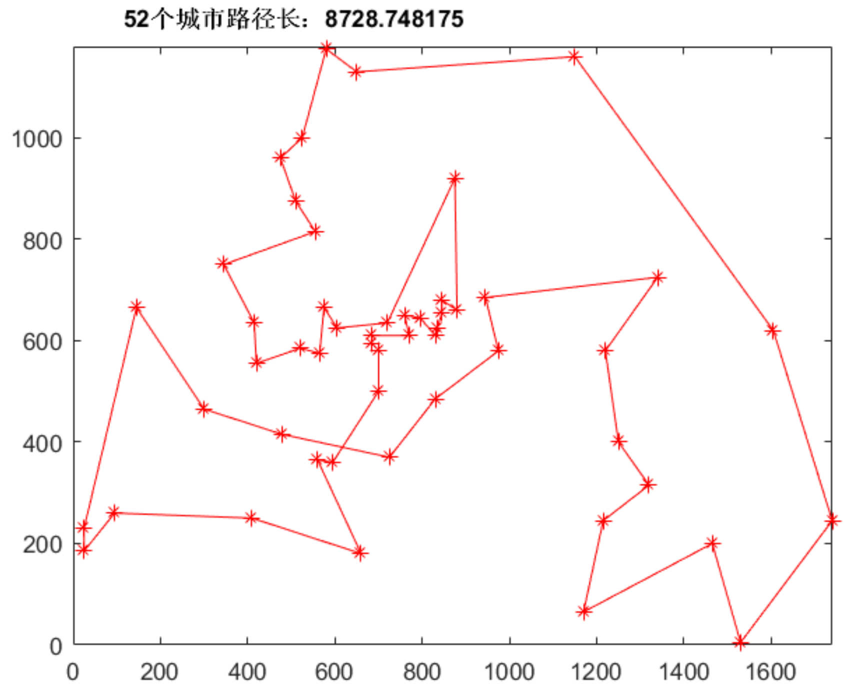
initialTemp = 5000; % 初 温

coolRate = 0.95;   nSwap = 3;

maxIter = 2000;

TSPSA(Coord, initialTemp, coolRate, maxIter, nSwap);





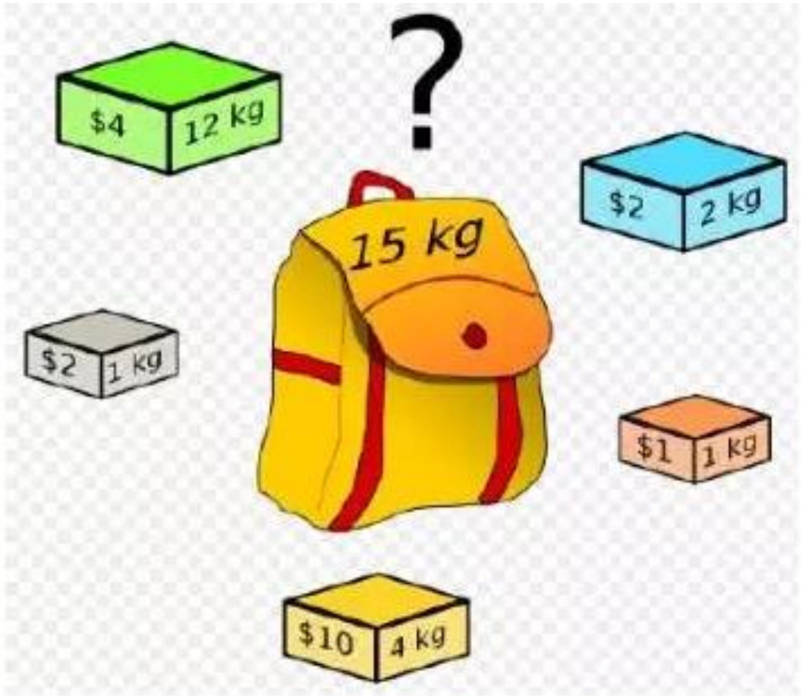
例3：背包问题

物品$j(j=1,2,\cdots,n)$的重量$w_j \geq 0$,价值$p_j \geq 0$.背包能承受的最大重量为W,求背包能装物品和的最大价值。如果限定每种物品只能选择0个或1个，则称为0-1背包问题. 数学模型:

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

$$s.t. \sum_{j=1}^{n} w_j x_j \leq W, x_j \in \{0,1\}$$

其中, x_i表示物品的取舍,0为舍,1为取.



模拟表示方案：对于n个物品的0-1背包问题，解空间为一维数组$(x_1, x_2, \dots, x_n)$。

模拟采样方案：类似TSP问题，采用部分逆转法，每次随机逆转某个x_i的值。

背包问题求解

```
%% 核心代码:
while iterations < maxIter
        %产生随机扰动
        tmp = ceil(rand*num);
        sol_new(1,tmp) = ~sol_new(1,tmp);
        %检查是否满足约束
        while 1
            q = (sol_new * W <= restriction);
            if ~q
               %如果不满足约束随机丢弃一个物品
                temp2=find(sol_new==1);
                temp3=ceil(rand*length(temp2));
    sol_new(temp2(temp3))=~sol_new(temp2(temp3));
            else
                break
            end
        end
end
```

```
%% 测试用例
P = [5;10;13;4;3;11;13;10;8;16;7;4];   %价值
W = [2;5;18;3;2;5;10;4;11;7;14;6];    %重量
restriction = 44;  % 重量限制
initialTemp = 97;
coolRate = 0.95;
endTemp = 3;
maxIter = 1000;
nSwap = 12;

beibao_SA(P,W,restriction,initialTemp,
endTemp,coolRate, maxIter, nSwap)
```

结果:

迭代次数 = 1000当前解为:

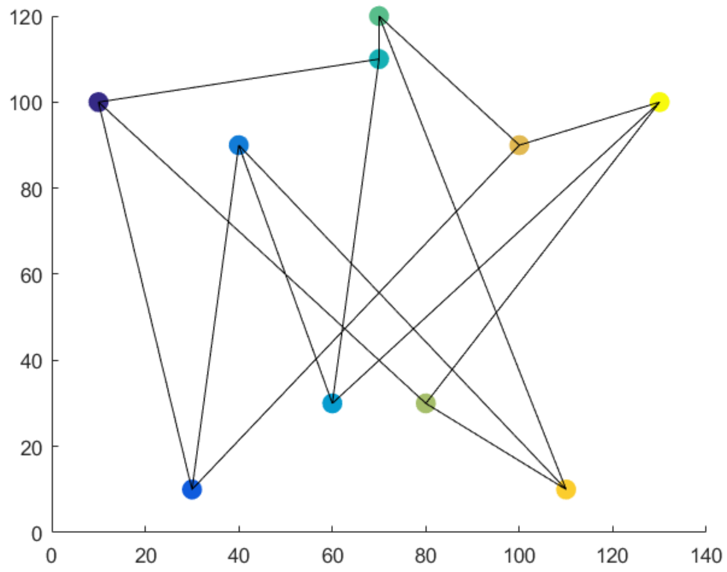
[1 1 0 1 1 1 1 1 0 1 0 1]

物品总价值等于: 76

背包中物品重量是: 44

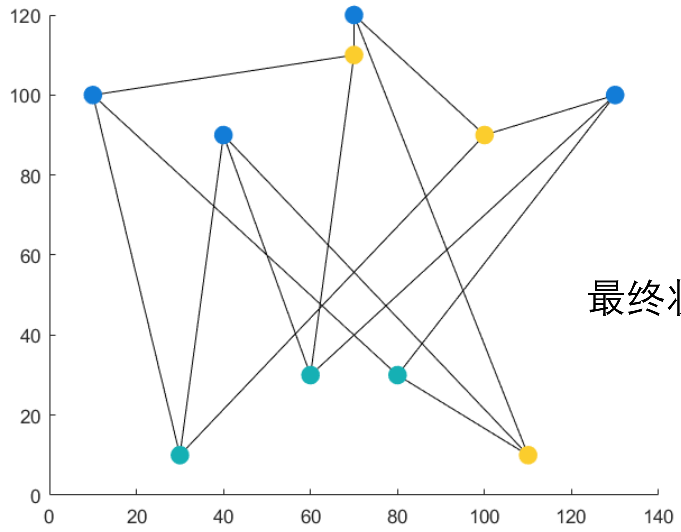
例4：顶点着色问题

➢ 给定一个无向图 G = (V, E)， 其中V为顶点集合，E为边集合，图着色问题即为将V分为K个颜色组，每个组形成一个独立集，即其中没有相邻的顶点。目标是求得最小的K值。 如下图每个点为一个独立集：



初始状态:10种颜色

➢ 模拟退火求解顶点着色问题，类似部分逆转法，每次都随机的将图中的某种颜色替换掉。



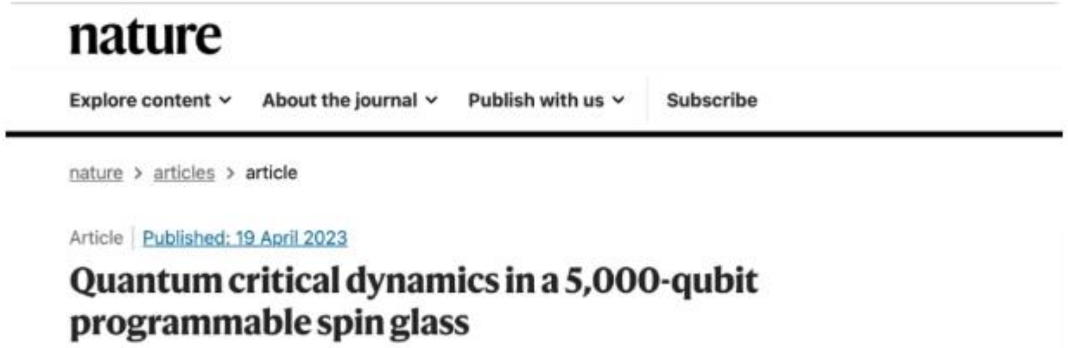
最终状态：3种颜色

```
while 1
      sol_new(ceil(rand*num)) =  sol_current(ceil(rand*num));
      if detect(graph, X, Y, sol_new)
         break
      else
         sol_new = sol_current;
      end
end
```

前沿：基于退火原理的量子计算机

1. D-Wave自1999年成立以来，一直专注于退火量子技术，其量子退火系统用户：**洛克希德马丁**、**谷歌**、NASA、**洛斯阿拉莫斯国家实验室**等。
2. **IBM** 发表一篇学术论文认为： D-Wave 的量子退火系统不是真正的量子计算机 https://arxiv.org/pdf/1401.7087v1.pdf
3. D-Wave 也表示，其退火量子计算系统是专为优化而设计，但却不能在模拟量子系统方面发挥作用。

- **2021年10月05日**宣布：将开始构建**门型量子计算机**，并预计将**于2023年或 2024 年将60-Qubit的量子系统推向市场**。将采用**超导**材料为其通用量子计算机构建量子比特。
- 2023年，相关研究以“Quantum critical dynamics in a 5,000-qubit programmable spin glass”为题发表于《Nature》



nature

Explore content ⌄ About the journal ⌄ Publish with us ⌄ Subscribe

nature > articles > article

Article | Published: 19 April 2023

Quantum critical dynamics in a 5,000-qubit programmable spin glass

D-Wave： 5000量子比特计算平台(2020) Advantage。唯一商用**量子退火机** (quantum annealing machine)

Homework 09

1. 课本P178，第2，3，6题

2. 实践和修改本讲附件相关程序，尝试

   ① 调整相关参数，给出一个尺寸为32x16的二维Ising相变模拟程序和有意义结果

   ② 运用计算加速技术，你能获得更大尺度的模拟结果吗？请给出相关解释和说明。