



UNIVERSIDAD SIMÓN BOLÍVAR

LACONGA-PHYSICS

d) Algoritmo de Swenden-Wang
Algoritmo de Wolf

Profesora
GLORIA BUENDÍA

27 de abril de 2023

Contenido General

1	Algoritmo de Swendsen-Wang SW (1987)	1
1.1	Descripción del algoritmo de SW	1
1.2	Probar que el algoritmo de SW satisface balance detallado	2
2	Algoritmo de Wolff (1989)	3
2.1	Descripción del algoritmo de Woff para el modelo de Ising	3



1. Algoritmo de Swendsen-Wang SW (1987)

Primer algoritmo no local para simulaciones de MC. Diseñado para eliminar el problema del "critical slowing down" (a medida que nos acercamos a la T_c necesitamos un gran número de pasos para alcanzar el equilibrio térmico, es muy difícil general configuraciones diferentes). El tiempo de correlación se comporta como $\tau = \xi^z$, donde ξ es la longitud de correlación. Cerca de T_c el tiempo de correlación y por consiguiente el tiempo de simulación se vuelven enormes. Por ejemplo para el Modelo de Ising standard

Algoritmo de Metropolis $z=2$ (3d) $z=2.125$ (2d)

Algoritmo SW $z=0.75$ (3d) $z=0.35$ (2d)

SW es un algoritmo no local.

1.1. Descripción del algoritmo de SW

Comienzo con una distribución de espines al azar. A cada par de vecinos cercanos se le asocia un $b_{k,l}=0$ ó 1 de acuerdo con

—Si k y l son distintos (antiparalelos)

$b_{kl}=0$

—Si k y l son iguales (paralelos)

$b_{kl}=0$ con $P=e^{-2\beta J_{kl}}$

$b_{kl}=1$ con $P=1-e^{-2\beta J_{kl}}$

b_{kl} es un índice de conectividad, dos espines solo pueden estar conectados si son iguales, pero no todos los espines iguales están conectados. Después de asignados los links identificamos los clusters de espines (conjunto de espines paralelos conectados entre sí por vecinos cercanos) y los volteamos con probabilidad $1/2$.

Justificación del modelo (ejemplo para el modelo de Ising)

$$H = -\sum_{\langle ij \rangle} J_{ij} S_i S_j$$

donde

$$Z = \sum_C e^{-\beta H}; \text{ Función de partición canónica}$$

considere la interacción entre un par de vecinos cercanos k, l y restesela a H

$$H_{kl} = -\sum_{\langle ij \rangle} J_{ij} S_i S_j + J_{kl} S_k S_l$$

defina ahora las funciones de partición restringidas

$$Z_{kl}^I = \sum_C e^{-\beta H_{kl}} \delta_{S_k S_l} \neq 0 \quad S_k = S_l$$

$$Z_{kl}^D = \sum_C e^{-\beta H_{kl}} [1 - \delta_{S_k S_l}] \neq 0 \quad S_k = -S_l$$

La función de partición Z ahora puede escribirse como



$$Z = \sum_C e^{-\beta[H_{kl} - J_{lk}S_kS_l]} = e^{\beta J_{kl}} Z_{kl}^I + e^{-\beta J_{kl}} Z_{kl}^D$$

si definimos

$$Z_{kl}^{ID} = Z_{kl}^I + Z_{kl}^D$$

entonces

$$Z = (e^{\beta J_{kl}} - e^{-\beta J_{kl}}) Z_{kl}^I + e^{-\beta J_{kl}} Z_{kl}^{ID}$$

$$Z = e^{\beta J_{lk}} [(1 - e^{-2\beta J_{kl}}) Z_{kl}^I + e^{-2\beta J_{kl}} Z_{kl}^{ID}]$$

el primer término tiene la restricción de que los espines k y l sean iguales (paralelos), en el segundo término los espines pueden ser iguales o no. Asociamos cada factor con la probabilidad de que los espines sean iguales o no

$$(1 - e^{-2\beta J_{kl}}) \text{Probabilidad de que k y l sean paralelos}$$

$$e^{-2\beta J_{kl}} \text{Probabilidad de que k y l sean antiparalelos}$$

Despues de asignados los links, identificamos los clusters de espines (conjunto de espines paralelos conectados entre sí, y los volteamos con probabilidad 1/2.

1.2. Probar que el algoritmo de SW satisface balance detallado

$$\frac{P_{C \rightarrow C'}}{P_{C' \rightarrow C}} = \frac{P_{C \rightarrow CC} P_{CC \rightarrow C'}}{P_{C' \rightarrow CC} P_{CC \rightarrow C}}$$

CC es una configuración intermedia cualquiera, tal que $P_{CC \rightarrow C'} = P_{C' \rightarrow CC}$.

$$\frac{P_{C \rightarrow C'}}{P_{C' \rightarrow C}} = \frac{e^{-\beta \sum_{\langle kl \rangle} J_{kl} \delta_{S_k S_l}}}{e^{-\beta \sum_{\langle kl \rangle} J_{kl} \delta_{S_k' S_l'}}$$

La diferencia de energía, ΔE entre la configuración C' y la C es

$$\Delta E = - \sum_{kl} J_{kl} [S_k' S_l' - S_k S_l] = - \sum_{kl} J_{kl} [\delta_{S_k' S_l'} - (1 - \delta_{S_k' S_l'}) - \delta_{S_k S_l} + (1 - \delta_{S_k S_l})] = -2 \sum_{kl} J_{kl} [\delta_{S_k' S_l'} - \delta_{S_k S_l}]$$

por lo que

$$\frac{P_{C \rightarrow C'}}{P_{C' \rightarrow C}} = e^{-\beta \Delta E}$$

el algoritmo satisface balance detallado!!.

Es facil ver porque este algoritmo tiene un parámetro dinámico mas bajo que el algoritmo de Metropolis, por lo que funciona muy bien cerca de la T_c . Como hemos mencionado cerca del punto crítico se forman grandes clusters de espines paralelos y es muy difícil que cambien con el algoritmo de Metropolis, pero con este algoritmo se pueden voltear todos a la vez.



2. Algoritmo de Wolff (1989)

Este algoritmo elimina completamente el problema del "critical slowing down", en este caso $\tau = \xi^z$ con $z=1$ (recordemos que τ es tiempo de correlación y ξ es la longitud de correlación).

2.1. Descripción del algoritmo de Wolff para el modelo de Ising

1. Seleccione al azar un sitio i de la red
2. Visite todos los vecinos, sitios j , y añada j a un cluster al que pertenece i con probabilidad

$$P_+(S_i S_j) = 1 - e^{-\max(0, 4\beta J S_i S_j)}$$

Para un sistema ferromagnético, $J>0$, P_+ es cero si i, j son antiparalelos, y $1 - e^{-4\beta J}$ si son paralelos.

3. Repita el segundo paso pero ahora sustituyendo i por cada uno de los j que se unieron al cluster.

4. Repita el paso 3 hasta que no se añadan sitios nuevos

5. Voltee todos los espines del cluster

Este algoritmo cumple ergodicidad, siempre hay una probabilidad distinta de cero, de que el cluster tenga un solo espin, y volteando una secuencia de espines individuales siempre se pueden generar todas las configuraciones posibles.

También cumple balance detallado:

Dos configuraciones C y C' con espines $[S_i]$ y $[S_i']$ respectivamente, pueden convertirse la una en la otra invirtiendo un cluster de espines. El ratio entre las probabilidades de transición es

$$\begin{aligned} \frac{P_{C \rightarrow C'}}{P_{C' \rightarrow C}} &= \prod \frac{1 - P_+(S_i S_j)}{1 - P_+(S_i' S_j')} \\ &= e^{4\beta J \sum_{i \in C; j \notin C'} S_i S_j} = e^{-\beta(E_{C'} - E_C)} \end{aligned}$$

Se utilizó la invariancia del Hamiltoniano bajo la inversión de dos espines para cancelar las probabilidades de añadir los espines internos del cluster para las dos configuraciones. Pero las probabilidades de añadir espines en el borde del cluster no se cancelan, dan un factor de $e^{4\beta J}$ para cada par (ij) alineados en la superficie y un factor de $e^{-4\beta J}$ para cada par no alineado de (ij) . A medida que nos acercamos a T_c los clusters crecen más y más y el número de operaciones requeridas crece.