

**Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Estatística**

**Algoritmos de Swendsen-Wang: um exemplo de
algoritmos de Monte Carlo com atualização coletiva**

Sabino J. Ferreira Neto

**Relatório Técnico
RTE-04/2004
Série Ensino**

1. Introdução

O objetivo deste trabalho é apresentar em detalhes um exemplo particular (denominado algoritmo de Swendsen-Wang) de algoritmos de Monte Carlo denominados algoritmos de atualização coletiva ou de *clusters*. O pré-requisito básico para o material contido neste trabalho é o conhecimento dos conceitos elementares da teoria de campos aleatórios e de cadeias de Markov [4]. De uma maneira geral podemos dizer que o objetivo básico de todo algoritmo de Monte Carlo é: dado um campo aleatório de Markov com uma distribuição de probabilidade associada gostaríamos de um algoritmo que gerasse amostras aleatórias deste campo de acordo com a distribuição de probabilidade associada. Nos algoritmos de Monte Carlo mais conhecidos como o de Metrópolis [4,7,9,10] ou o do amostrador de Gibbs [4,7] um passo elementar do algoritmo atualiza apenas uma componente do campo aleatório de Markov de cada vez (que denominamos atualização local). Como iremos mostrar aqui, um passo elementar dos chamados algoritmos de atualização coletiva atualiza um aglomerado ou *cluster* de componentes do campo aleatório. Em um trabalho anterior [1] mostramos que a variância das estimativas obtidas por simulações de Monte Carlo é uma função crescente de uma determinada medida da correlação estatística entre as sucessivas amostras do campo aleatório geradas pelo algoritmo. Para determinados valores (denominados valores críticos) dos parâmetros que entram na distribuição de probabilidade do campo simulado esta correlação pode ficar muito alta nos caso dos algoritmos de atualizações locais prejudicando consideravelmente a precisão das estimativas obtidas. Como vamos discutir neste trabalho, devido à natureza não local de seu mecanismo de atualização a correlação estatística entre as sucessivas amostras geradas do campo aleatório é sensivelmente reduzida e em alguns casos eliminada nos casos de simulações com algoritmos de *clusters* com os mesmos valores

críticos para os parâmetros da distribuição de probabilidade. É importante observar que se os valores dos parâmetros da distribuição de probabilidade forem diferentes dos valores críticos este problema da correlação deixa de ser relevante e o uso dos algoritmos locais se torna mais vantajoso devido aos seus menores custos computacionais (típicamente 10 vezes menores) se comparados aos dos algoritmos de atualização coletiva.

A seguir apresentamos formalmente conceitos, notações e definições que serão usados ao longo deste trabalho.

Seja $S = \{1, \dots, n\}$ um conjunto discreto de n índices que indexam um conjunto de n sítios. Um sítio representa um ponto ou um região (neste caso representada pelo seu centroide) de um espaço Euclidiano. Por exemplo uma imagem digital é representada por uma rede bidimensional retangular de n quadrados elementares (denominados pixels), neste caso cada pixel representado pelas coordenadas espaciais (k, l) de seu centroide seria um dos sítios do conjunto de sítios S que representa a imagem. No exemplo da imagem os sítios tem uma distribuição espacial *regular* denominada rede mas de uma maneira mais geral os sítios podem ter uma distribuição espacial irregular. Para a imagem poderíamos reescrever o conjunto S como: $S = Z_m^2 = \{(k, l) \in Z^2 \mid 1 \leq k, l \leq m\}$ onde $Z^2 = Z \times Z$ é o produto cartesiano de dois espaços Z dos números reais inteiros. Cada um dos $m \times m$ pixels (k, l) da imagem é indexado por um único índice $i \in \{1, \dots, n\}$ desde que $n = m \times m$. Vamos usar a notação de representar um sítio por um único índice ao longo deste trabalho. O relacionamento entre os sítios de S será estabelecido por um sistema de vizinhança N definido como:

$$N = \{N_i \mid \forall i \in S\}$$

onde N_i representa o conjunto de sítios que são vizinhos do sítio i . A relação de vizinhança tem as seguintes propriedades:

1. um sítio não é vizinho de si mesmo, ou seja $i \notin N_i$.
2. a relação de vizinhança é mútua, ou seja $i \in N_j \Leftrightarrow j \in N_i$.

Para conjunto regular de sítios S definimos a vizinhança N_i de um sítio i como:

$$N_i = \{j \in S \mid |\text{dist}(i, j)|^2 \leq r\} \quad (1)$$

onde $\text{dist}(i, j)$ representa a distância euclideana entre os sítios i e j . Denotamos esta relação de vizinhança por $i \sim j$. Neste trabalho consideramos um sistema particular de vizinhança onde $r = 1$ (ver a Figura 1 onde j_1, j_2, j_3 e j_4 são os vizinhos do sítio i).

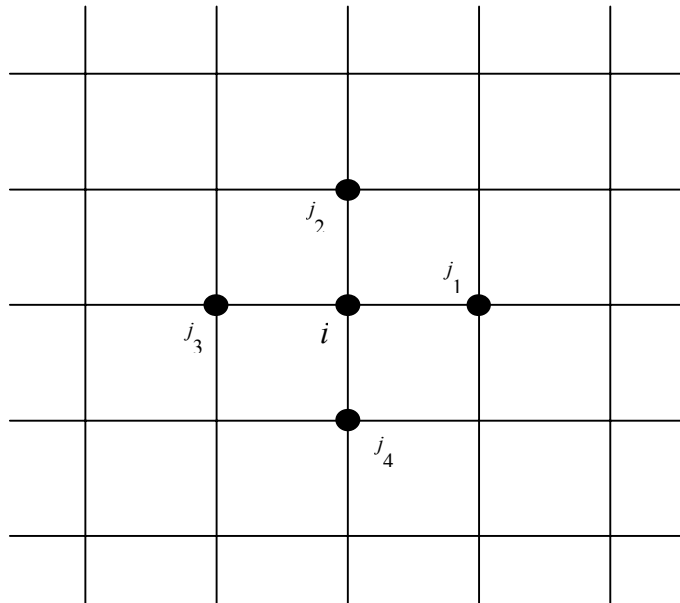


Figura 1

Definimos o campo aleatório X sobre o conjunto de sítios S como uma família de variáveis aleatórias denotada por $X = \{X_1, X_2, \dots, X_n\}$. Para cada sítio $i \in S$ temos uma variável aleatória associada X_i que assume valores x_i em um espaço de estados Λ_i (neste trabalho e na maioria das aplicações considera-se que todas as componentes X_i do campo aleatório X assumam valores em um mesmo espaço de estados Λ). Usamos a notação $X_i = x_i$ para denotar o evento onde a componente X_i assume o valor x_i e a notação $X = x$ onde $x = \{x_1, x_2, \dots, x_n\}$ para denotar o evento conjunto $(X_1 = x_1, X_2 = x_2, \dots, X_n = x_n)$ que é um dos estados ou configurações possíveis do espaço de estados conjuntos $\Lambda^n = \Lambda \times \Lambda \times \dots \times \Lambda$ que é o produto cartesiano de n espaços Λ . Ao campo aleatório X associamos uma medida de probabilidade que denominamos distribuição de probabilidade de Gibbs [4], denotada por $\pi(x)$ e definida como:

$$\pi(x) = (1/Z) e^{-U(x)}, \quad Z = \sum_{\{x\} \in \Lambda^n} e^{-U(x)}, \quad U(x) = \sum_{i \sim j} V(x_i, x_j) \quad (2)$$

onde

- Z – Constante de Normalização (Função Partição)
- $U(x)$ – Função Energia . Neste trabalho onde estamos considerando um sistema de vizinhança tal que um par de sítios i e j são considerados vizinhos quando satisfazem a condição (1) com $r = 1$ o somatório na expressão da função energia engloba todos os pares distintos de vizinhos do conjunto de sítios S . A função V denominada função potencial é uma função somente dos estados x_i e x_j assumidos pelas componentes associadas aos sítios i e j do campo aleatório X e de um parâmetro J_{ij} que é um número real representando uma medida da intensidade da

interação entre os dois sítios vizinhos i e j . É a forma específica da função V que caracteriza um determinado modelo como veremos a seguir na Seção 2 quando introduzirmos o modelo de Potts e de Ising. Observamos também que para definições de vizinhança com valores de $r > 1$ em (1) a expressão da função energia em (2) pode incluir potenciais que sejam funções de 3 ou mais componentes do campo aleatório X . Detalhes destes casos mais gerais podem ser vistos em [4].

Considerando as definições e notações apresentadas podemos dizer que toda simulação de Monte Carlo tem uma estrutura geral: dada uma medida de probabilidade $\pi(x)$ sobre um espaço de todos estados possíveis $\Lambda^n = \{x\}$ de um campo aleatório X gostaríamos de ter um algoritmo que nos permitisse gerar sucessivas amostras aleatórias $x^{(i)}$ do campo X com a probabilidade correta $\pi(x^{(i)})$ para, por exemplo, estimar médias do tipo $\sum_{\{x\}} f(x) \pi(x)$ (vamos considerar que estamos tratando de um espaço de estados finito). Este algoritmo com as propriedades adequadas [1] é a implementação computacional de uma cadeia de Markov que tem como distribuição de probabilidade de equilíbrio a medida π . Como mencionamos anteriormente, um problema que ocorre nestas simulações é que para determinados valores (denominados valores críticos) dos parâmetros que entram na distribuição de probabilidade do campo simulado os sucessivos estados gerados não são estatisticamente independentes. Esta correlação entre os sucessivos estados faz com que as variâncias da estimativas obtidas sejam maiores do que no caso onde os sucessivos estados gerados são independentes. Na verdade, a variância da estimativa obtida no caso correlacionado é a variância que seria obtida caso não houvesse correlação multiplicada por duas vezes o tempo de autocorrelação τ (o chamado tempo de autocorrelação τ em unidades de passos da cadeia é justamente uma medida da

correlação entre os sucessivos estados da cadeia). Todos estes conceitos são tratados em detalhes em [1].

Como mostraremos brevemente na Seção 3 no passo mais elementar de um algoritmo como o de Metropolis fazemos a atualização da componente do campo aleatório X_i associada ao um sítio $i \in S$. Um passo da cadeia associada ao algoritmo é obtido quando todos os sítios de S são visitados e as respectivas componentes do campo aleatório X são atualizadas. É justamente esta atualização sítio a sítio que dá o carácter local a algoritmos como o de Metropolis e o do amostrador de Gibbs. Neste trabalho vamos mostrar um exemplo de algoritmo de Monte Carlo denominado algoritmos de Swendsen-Wang onde, ao contrário dos algoritmos de atualização local como o de Metropolis ou do amostrador de Gibbs, em vez de atualizarmos o campo aleatório sítio a sítio em um passo elementar do algoritmo atualizamos simultaneamente um determinado aglomerado ou *cluster* de sítios de S . Estes algoritmos apesar de terem maior complexidade e custo computacional (típicamente 10 vezes maiores) se comparados aos algoritmos de atualização local (ou sítio a sítio) reduzem significativamente [11] e em alguns casos eliminam [17] a correlação entre os estados ao longo da cadeia. A presença desta correlação é especialmente significativa quando utilizamos estes algoritmos para simular sistemas em regime crítico (significando que o sistema passa por uma transição de fase). Nesta situação pode-se mostrar que dependendo da natureza da transição de fase o tempo de autocorrelação τ se comporta como

$$\tau \sim \min(L, \xi)^z \quad (3)$$

sendo que L o comprimento linear do conjunto de sítios S e ξ representa o comprimento de correlação espacial entre duas componentes quaisquer do campo aleatório x . A definição de ξ vem do comportamento típico da função de correlação espacial :

$$C(x_i, x_j) = \sum_{\{x\}} x_i x_j \pi(x) \sim \exp(-|i-j|/\xi) \quad (4)$$

O expoente dinâmico crítico z , que aparece na expressão (3) determina o comportamento do tempo de auto-correlação τ e portanto a dinâmica das correlações entre os sucessivos estados gerados ao longo da cadeia, é característico de uma classe de algoritmos. Para os algoritmos locais e para a maioria dos modelos de interesse $z \geq 2$, quando é maior do que 2 o valor do expoente z está sempre próximo a 2 [16]. O algoritmo de Swendsen Wang que é um tipo de algoritmo de Monte Carlo de atualização coletiva, quando utilizado no estudo das propriedades críticas do modelo de Ising (que definimos na próxima seção) apresenta um comportamento dinâmico de acordo com valor estimado de 0.35 para o expoente z [11].

É exatamente este comportamento crítico do tempo de autocorrelação ($\tau \sim \min(L, \xi)^z$), que faz com que as estimativas obtidas por simulações de Monte Carlo no estudo de modelos no regime crítico, apresentem o denominado *desaceleramento crítico*. Como discutimos em detalhes em [1] a variância das estimativas de Monte Carlo são proporcionais à τ/n onde n representa o número de iterações do algoritmo, ou seja, para obtermos estimativas com precisão de 1% por exemplo temos que fazer um número n de passos de Monte Carlo tal que $n \approx 10^4 \tau$. Porém, como vamos mostrar na Seção 4 em um único passo de um algoritmo de atualização coletiva como o de Swendsen Wang já é possível obter um estado radicalmente diferente do estado original.

Novamente é importante salientar que fora do regime crítico é mais conveniente usar os chamados algoritmos locais por sua simplicidade computacional se comparados aos algoritmos de cluster para se fazer as estimativas de Monte Carlo.

O objetivo deste trabalho é apresentar em detalhes a mecânica de um algoritmo de atualização coletiva, o chamado algoritmo de Swendsen Wang [11] aplicado ao modelo de Potts e de Ising. Na Seção 2 junto com as notações e definições necessárias apresentamos o modelo de Potts e de Ising. Na Seção 3 apresentamos de uma maneira breve o algoritmo de atualização local de Metropolis. Finalmente, na Seção 4 discutimos em detalhes o algoritmo de Swendsen Wang aplicado ao modelo de Potts.

2. O modelo de Ising e de Potts: definição e notação

Considerando as definições e notações da Seção 1 definimos o modelo de Potts [2,3] sobre o conjunto de sítios S como o campo aleatório $X = \{X_1, X_2, \dots, X_n\}$ com espaço de estados Λ^n onde $\Lambda = \{1, 2, \dots, q\}$, q é um número inteiro positivo e uma distribuição de probabilidade de Gibbs $\pi(x)$ dada por:

$$\pi(x) = (1/Z) e^{-U(x)}, \quad Z = \sum_{\{x\} \in \Lambda^n} e^{-U(x)}, \quad U(x) = \sum_{i \sim j} J_{ij} (1 - \delta(x_i, x_j)) \quad (5)$$

onde

- Z – Constante de Normalização (Função Partição);
- J_{ij} – Parâmetro que mede a intensidade da interação entre dois sítios vizinhos i e j ;

- $U(x)$ – Função Energia .O somatório envolve todos os pares distintos de vizinhos i e j ;
- $\delta(x_i, x_j) = \begin{cases} 1 & \text{se } x_i = x_j \\ 0 & \text{se } x_i \neq x_j. \end{cases}$

O modelo de Ising é um caso particular do modelo de Potts com o número de estados $q = 2$ e as componentes $x_i = \pm 1$. Toda a terminologia acima vem da Física onde estes modelos foram originalmente introduzidos para, no caso do modelo de Ising, modelar materiais ferromagnéticos que, dependendo do valor que o parâmetro que representa a temperatura assume, sofrem uma transição de fase entre uma fase magnetizada de baixa temperatura e uma fase de magnetização nula a altas temperaturas [5,6]. Os dois modelos hoje são empregados em diversas áreas do conhecimento, em particular, o modelo de Potts é utilizado na representação probabilística de imagens digitais [7] onde os sítios são os pixels (elementary picture) da imagem e os q estados das componentes x_i são os possíveis tons de cinza dentro de uma escala que vai do preto ao branco que um pixel pode assumir. Um exemplo de aplicação recente do modelo da Ising é na área de vigilância [8].

3. Algoritmo de Metropolis: um exemplo de algoritmo de Monte Carlo com atualização local.

Tentando ser o mais completo possível vamos apresentar brevemente o algoritmo de Metropolis onde as atualizações são realizadas localmente, sítio a sítio. Existe uma vasta literatura sobre o assunto, aqui apresentamos apenas as referências originais [9,10] e de um livro texto [4]. Uma iteração do algoritmo é completada quando visitamos sistematicamente

todos os sítios de S ou então, quando fazemos um número k de visitas a sítios escolhidos aleatoriamente sendo k igual ao número de sítios de S . Vale a pena repetir a idéia básica de todo algoritmo de Monte Carlo:

O método de Monte Carlo para geração de amostras aleatórias de uma distribuição de probabilidade π pode ser considerado como uma prescrição para a realização em computadores de uma cadeia de Markov cuja distribuição estacionária seja π .

Algoritmo de Metropolis

- I. Inicialize o campo $\{x\}$ aleatoriamente (para cada sítio i sorteie uniformemente um dos q estados possíveis para x_i).
- II. Para cada um dos sítios $i \in S$
 - a. Gere aleatória e uniformemente um dos q estados possíveis como o novo possível estado x'_i
 - b. Calcule $\Delta U = U(x') - U(x)$, o campo x' é o mesmo campo x exceto pela componente do sítio i que agora está no estado x'_i .
 - c. Calcule $p = \min \{1, e^{-\Delta(U)}\}$
 - d. Com probabilidade p faça $x_i = x'_i$
- III. Repita o passo II $N = \text{PMC}$ (Passos de Monte Carlo) vezes

A visita a cada um dos sítios $i \in S$ pode ser feita de diversas maneiras: 1) uma varredura seguindo a ordem crescente dos índices que identificam os sítios, 2) uma varredura alternada estilo tabuleiro de xadrez onde primeiro as casa brancas são visitadas e em seguida as casa

pretas ou então 3) escolhendo-se aleatoriamente o sítio a ser visitado desde que o número de visitas seja igual ao número de sítios do conjunto S . A natureza local do algoritmo é justamente devida ao passo II onde a componente do campo x associada a cada um dos sítios é atualizada individualmente.

4. Algoritmos de Swendsen Wang: um exemplo de algoritmo de Monte Carlo com atualizações coletivas

O algoritmo que vamos apresentar aqui foi introduzido originalmente por Swendsen e Wang [11] para estudar o modelo de Potts. Vamos começar reescrevendo a expressão para o fator $e^{-U(x)}$ que aparece na expressão (5) para o modelo de Potts como:

$$e^{-U(x)} = \exp\left[\sum_{i \sim j} J_{ij} (\delta(x_i, x_j) - 1)\right] = \prod_{i \sim j} [(1 - p_{ij}) + p_{ij} \delta(x_i, x_j)] \quad (6)$$

onde $p_{ij} = 1 - \exp(-J_{ij})$. Neste caso a medida de Gibbs $\pi_{Potts}(x)$ pode ser escrita como:

$$\pi_{Potts}(x) = Z^{-1} \exp\left[\sum_{i \sim j} J_{ij} (\delta(x_i, x_j) - 1)\right] = Z^{-1} \prod_{i \sim j} [(1 - p_{ij}) + p_{ij} \delta(x_i, x_j)] \quad (7)$$

e a função partição Z como:

$$Z = \sum_{\{x\} \in \Lambda^n} e^{-U(x)} = \sum_{\{x\}} \exp\left[\sum_{i \sim j} J_{ij} (\delta(x_i, x_j) - 1)\right] = \sum_{\{x\}} \prod_{i \sim j} [(1 - p_{ij}) + p_{ij} \delta(x_i, x_j)] \quad (8)$$

Sendo a e b dois números reais quaisquer podemos escrever:

$$a + b = \sum_{k=0}^1 [a \delta(k,0) + b \delta(k,1)] \quad (9)$$

onde $\delta(i, j) = \begin{cases} 1 & \text{se } i = j \\ 0 & \text{se } i \neq j \end{cases}$. Vamos aplicar a identidade (9) na expressão que aparece após a terceira igualdade na equação (8) para função partição fazendo as seguintes associações:

$$\begin{aligned} a &\rightarrow (1 - p_{ij}) \\ b &\rightarrow p_{ij} \delta(x_i, x_j) \end{aligned} \quad (10)$$

A cada ligação ou elo $i \sim j$ vamos associar uma variável auxiliar denotada por n_{ij} que assume os valores 0 e 1. Neste caso fazendo as associações em (10) e considerando (9) escrevemos a função partição Z como:

$$Z = \sum_{\{x\}} \sum_{\{n\}} \prod_{i \sim j} [(1 - p_{ij}) \delta(n_{ij}, 0) + p_{ij} \delta(n_{ij}, 1) \delta(x_i, x_j)] \quad (11)$$

onde o simbolo $\{n\}$ representa o conjunto de todas variáveis n_{ij} , uma para cada par distinto de sítios vizinhos i e j e o simbolo $\sum_{\{n\}}$ é uma notação compacta significando que para cada

uma das ligações $i \sim j$ associada a um par de sítios vizinhos i e j temos o somatório $\sum_{n_{ij}=0}^1$.

Agora vamos considerar o conjunto das variáveis $\{n\}$, que inicialmente tratamos simplesmente como um conjunto de números reais, como o conjunto de todas realizações n de um campo aleatório N cujas componentes N_{ij} são associadas a cada um dos pares distintos de sítios vizinhos i e j e assumem os valores n_{ij} que indicam o estado de ocupação do elo $i \sim j$:

$$n_{ij} = \begin{cases} 1 & \text{se o elo } i \sim j \text{ está ocupado} \\ 0 & \text{se o elo } i \sim j \text{ está vazio} \end{cases}$$

Definimos então o chamado modelo de Fortuin-Kasteleyn Swendsen Wang (FKSW) como um modelo conjunto para o campo de Potts X cujas componentes X_i são definidas nos sítios i e o campo N cujas componentes N_{ij} são definidas nos elos $i \sim j$ associados aos pares distintos de vizinhos i e j de S que tem uma distribuição conjunta de probabilidade dada por:

$$\pi_{FKSW}(x, n) = Z^{-1} \prod_{i \sim j} [(1 - p_{ij}) \delta(n_{ij}, 0) + p_{ij} \delta(n_{ij}, 1) \delta(x_i, x_j)] \quad (12)$$

sendo Z a função partição conjunta dada pela equação (11). Vamos considerar agora o que acontece com a função partição (11) quando fazemos a soma sobre todos os estados possíveis $\{x\}$ do campo X mantendo o campo N fixo em um dado estado n . Neste caso, cada elo $i \sim j$ ocupado estabelece a restrição de que os estados x_i e x_j sejam iguais. De outro modo, ou

seja se o elo $i\sim j$ não estiver ocupado, não existe nenhuma restrição para os estados x_i e x_j .

Inicialmente agrupamos os sítios em *clusters* conexos (um *cluster* é denominado conexo se ao considerarmos dois quaisquer de seus sítios existir entre eles pelo menos um caminho composto de elos ocupados formados por sítios pertencentes ao *cluster*). Todas as componentes X_i associadas aos sítios do *cluster* devem estar em um mesmo estado considerando que todos os q estados são igualmente prováveis. *Clusters* distintos são considerados independentes. Com isto obtemos a seguinte expressão para a função partição :

$$Z = \sum_{\{n\}} \left(\prod_{i\sim j : n_{ij}=1} p_{ij} \right) \left(\prod_{i\sim j : n_{ij}=0} (1-p_{ij}) \right) q^{C(n)} \quad (13)$$

onde $C(n)$ representa o número de *clusters* conexos associados ao estado do campo n (incluindo os *clusters* formados de um único sítio denominados *singletons*) no grafo cujas arestas são formadas pelos elos $i\sim j$ ocupados (para os quais $n_{ij}=1$). A função partição em (13) é a constante de normalização da medida de probabilidade $\pi_{RC}(n)$ de um modelo de *clusters* aleatórios com parâmetro q associados ao campo N [12], a qual assume a seguinte forma :

$$\pi_{RC}(n) = Z^{-1} \left(\prod_{i\sim j : n_{ij}=1} p_{ij} \right) \left(\prod_{i\sim j : n_{ij}=0} (1-p_{ij}) \right) q^{C(n)} \quad (14)$$

onde o sub-índice *RC* significa “*random cluster*”.

Resumindo os resultados obtidos:

- a) $Z_{Potts} = Z_{FKSW} = Z_{RC}$
- b) a distribuição marginal de π_{FKSW} em relação as variáveis $\{x\}$ é precisamente o modelo de Potts $\pi_{Potts}(x)$ (ver equação (7)).
- c) a distribuição marginal de π_{FKSW} em relação as variáveis de ocupação dos elos $\{n\}$ é justamente o modelo de clusters aleatórios $\pi_{RC}(n)$ (ver equação (14)).

Podemos também estabelecer as distribuições condicionais de π_{FKSW} :

- d) a distribuição condicional do campo $\{n\}$ dado um estado do campo de Potts $\{x\}$ é assim construída: *independentemente*, para cada elo $i \sim j$, fazemos $n_{ij} = 0$ com probabilidade 1 quando $x_i \neq x_j$. Caso contrário, quando $x_i = x_j$, fazemos $n_{ij} = 0, 1$ com probabilidade $1 - p_{ij}, p_{ij}$ respectivamente.
- e) a distribuição condicional do campo $\{x\}$ dado um estado do campo $\{n\}$ é construída da seguinte forma: *independentemente*, para cada *cluster* conexo, fazemos as variáveis x_i associadas aos sítios i do *cluster* iguais a um mesmo valor escolhido equiprovavelmente entre os valores do espaço de estados $\Lambda = \{1, 2, \dots, q\}$.

O algoritmo de Swendsen Wang [11] gera estados de acordo com a distribuição de probabilidade conjunta (12) usando alternadamente as duas distribuições condicionais d) e e) acima.

O Algoritmo de Swendsen Wang

- I. Inicialize o campo $\{x\}$ aleatoriamente, ou seja, para cada sítio i sorteie uniformemente um dos q estados possíveis para x_i .
- II. gere um estado para o campo $\{n\}$ dado o campo $\{x\}$ de acordo com a distribuição condicional apresentada no item d) acima.
- III. condicionado ao estado do campo $\{n\}$ que foi gerado no passo II acima, gere um novo estado para o campo $\{x\}$ de acordo com a distribuição condicional apresentada no item e) acima.

O passo II do algoritmo é trivial. O passo III depende de um algoritmo que identifique as componentes conexas de um grafo não-direcionado que é um problema clássico em ciência da computação [13,14,15]. Se considerarmos o problema do desaceleramento crítico podemos perceber através de um exemplo porque é plausível que o desaceleramento crítico seja menor no algoritmo de Swendsen Wang se comparado aos algoritmos locais. O exemplo seguinte mostra que um movimento de atualização local pode ter efeitos não-locais significativos em um único passo do algoritmo de Swendsen-Wang. Na Figura 2 temos um cluster conexo de sítios onde as variáveis x_i estão no mesmo estado que é composto de dois subconjuntos de sítios C_1 e C_2 ligados por um único elo ocupado $i \sim j$. Se no passo II a variável n_{ij} associada a este elo assumir o valor zero os dois subconjuntos de sítios C_1 e C_2 serão desconectados e no passo III do algoritmo o novo valor que as variáveis x_i do subconjunto C_1 assumirão será

completamente independente do novo valor que as variáveis x_i do subconjunto C_2 assumirão. Ou seja uma atualização local, no caso $n_{ij} = 0$ gerou um efeito não-local significativo no passo seguinte. Dada uma configuração do campo estes efeitos não-locais de um algoritmo de *cluster* fazem com que em poucos passos do algoritmo se possa obter uma nova configuração

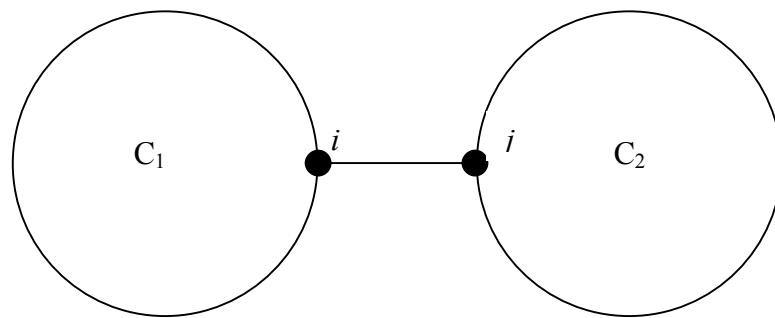


Figura 2

Na equação (3) para o comportamento do tempo de auto-correlação τ em uma simulação de Monte Carlo feita com os parâmetros do modelo assumindo valores críticos o expoente dinâmico crítico z é sempre igual ou maior do que 2 para os algoritmos locais [16] enquanto, por exemplo, para o modelo de Ising o algoritmo de Swendsen-Wang apresenta $z \approx 0.35$ [11] e em alguns casos o desaceleramento crítico pode ser eliminado, $z \approx 0$ [17] (neste trabalho foi utilizado uma pequena variação do algoritmo de Swendsen-Wang no estudo das propriedades críticas do modelo de Potts anti-ferromagnético com $q = 3$ estados). Como a variância das estimativas obtidas pelas simulações de Monte Carlo dependem linearmente do tempo de autocorrelação τ [1] mesmo considerando o maior custo computacional dos algoritmos de *clusters* comparados aos algoritmos locais (típicamente 10 vezes maiores) os menores valores

dos expoentes z torna os algoritmos de *clusters* os mais indicados no uso de simulações de Monte Carlo para o estudo de propriedades críticas de modelos como o de Potts e o de Ising.

Devido ao fato da escolha do estado do campo que inicializa o algoritmo ser completamente arbitrário não refletindo a distribuição de probabilidade de equilíbrio ou estacionária segundo a qual se deseja que o algoritmo gere amostras aleatórias, o procedimento comum nas simulações de Monte Carlo é o de descartar um determinado número das configurações iniciais geradas pelo algoritmo para se evitar o chamado vício de inicialização para as estimativas obtidas. Um critério é o de descartar as primeiras $n/5$ configurações geradas onde n é o número total de iterações a serem feitas do algoritmo. Este critério é usado tanto para os algoritmos de atualização local quanto para os de atualização coletiva.

5. Referências

- [1] – S. J. Ferreira, *Um estudo de Cadeias Markovianas: Algoritmos Dinâmicos de Monte Carlo*, Relatório Técnico Série Ensino RTE-02/2000, Departamento de Estatística, UFMG (2000).
- [2] – R. B. Potts, Proc. Camb. Phil. Soc **48**, 106 (1952).
- [3] – F. Y. Wu, Rev. Mod. Phys **54**, 235 (1982).
- [4] – P. Bremaud, *Markov Chains Gibbs Fields, Monte Carlo Simulation and Queues*. Springer-Verlag New York (1999).

- [5] – C. J. Thompson, *Mathematical Statistical Mechanics*, Princeton University Press
Princeton, NY (1972).
- [6] – R. J. Baxter, *Exactly Solved Models in Statistical Mechanics*, Academic Press Limited
London (1982).
- [7] – G. Wrinkler, *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods:
A Mathematical Introduction*, 2nd edition, Applications of Mathematics: Stochastic
Modelling and Applied Probability Springer-Verlag Berlin (2003).
- [8] – E. Järpe, Surveillance of the interaction parameter of the Ising model, *Communs. Statist.
Theory Meth.*, **28**, 3009 (1999).
- [9] – N. Metropolis, M. N. Rosenbluth, A. W. Rosenbluth, A. H. Teller e E. Teller; *Equations
of state calculations by fast computing machines*, *J. Chem. Phys.*, **21**, 1087, (1953).
- [10] – W. K. Hastings, Monte Carlo sampling methods using Markov chains and their appli-
cations, *Biometrika*, **57**, 97 (1970).
- [11] – R. H. Swendsen e J. S. Wang, *Phys. Rev. Lett.* **58**, 86 (1987); J. S. Wang e
R. H. Swendsen, *Physica* **A167**, 565 (1990).
- [12] – P. W. Kasteleyn e C. M. Fortuin, *J. Phys. Soc. Japan* **26** Suppl., 11 (1969); C. M.
Fortuin e P. W. Kasteleyn, *Physica* **57**, 536 (1972); C. M. Fortuin, *Physica* **58**, 393
(1972); C. M. Fortuin, *Physica* **59**, 545 (1972).
- [13] – A. Gibbons, *Algorithmic Graph Theory*, Chapter 1, Cambridge University Press
(1985).
- [14] – S. Even, *Graph Algorithms*, Chapter 3, Computer Science Press, Potomac Maryland
(1979).
- [15] – D. E. Knuth, *The Art of Computer Science*, vol 1 2nd ed., pags 353-355, 360, 572,
Addison-Wesley, Reading, Massachusetts (1973).

- [16] – G. F. Mazenko e O. T. Valls, Phys. Rev. **B24**, 1419 (1981); C. Kalle, J. Phys. **A17**, L801 (1984); J. K. Williams, J. Phys. **A18**, 49 (1985); S. Wansleben e D. P. Landau, J. Appl. Phys. **61**, 3968 (1987).
- [17] – S. J. Ferreira e A. D. Sokal, J. Stat. Phys. **Vol 96 Nos. 3/4**, 461 (1999).