

2. MÉTODO DE RECOZIMENTO SIMULADO (SIMULATED ANNEALING)

- 2.1. [Analogia Física: o recozimento de metais \(annealing\)](#)
- 2.2. [A Tradução Matemática: o algoritmo básico](#)
- 2.3. [A Programação de Recozimento](#)
- 2.4. [Critério de Aceitação de Soluções de Maior Energia](#)
- 2.5. [A Estrutura de Vizinhaça](#)
- 2.6. [Aspectos de Implementação do Algoritmo](#)
- 2.7. [Exemplo Ilustrativo](#)

2. MÉTODO DE RECOZIMENTO SIMULADO

2.1. Analogia Física: o recozimento de metais (*annealing*)

O recozimento de metais (também chamado de recristalização) é um processo de alteração de propriedades de um material metálico por aquecimento e resfriamento lento.

Quando um material é submetido a um processamento a frio, sua estrutura cristalina é deformada e surgem vários pontos de tensão. Os cristais deformados têm mais energia que os não deformados, por causa da desorganização da estrutura cristalina nas interfaces entre os grãos. Estas estruturas são estáveis nesta temperatura, já que os átomos não têm mobilidade suficiente para alterar a sua organização cristalina. Havendo oportunidade, os átomos irão se deslocar visando um arranjo mais perfeito e regular, de menor energia.

Para fazer cristais perfeitos a partir da técnica de recozimento, a temperatura do sistema é aumentada até pouco abaixo do ponto de fusão do material. A altas temperaturas, os átomos se movimentarão, e eventualmente chegarão à configuração mais estável que pode ser alcançada nesta temperatura. A temperatura é reduzida, e é dado algum tempo para o material se acomodar a cada nova temperatura, tentando minimizar sua energia. A seqüência de alterações de temperatura e o tempo gasto em cada etapa configura uma programação de recozimento (“annealing schedule”).

Quanto mais organizada a estrutura cristalina de um material, menor a sua energia. Para qualquer material, existe pelo menos uma estrutura atômica que minimiza sua energia na temperatura de zero absoluto. Esta energia “mínima global” corresponde ao arranjo perfeitamente regular do material, sem qualquer interface entre grãos de organização cristalina diferente. Há estruturas que são estáveis a certas temperaturas, já que os átomos não têm mobilidade suficiente na rede cristalina para buscar novas estruturas. Estes estados correspondem a mínimos locais de energia.

A Figura 1 mostra a estrutura cristalina do material antes e depois do processo de recozimento.

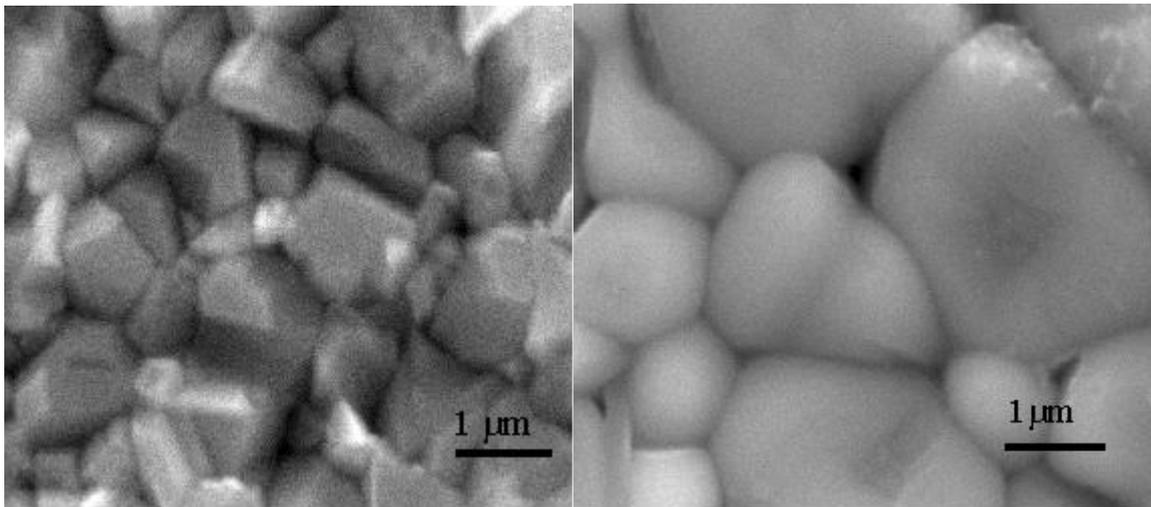


Figura 1 - Morfologia de um filme de CdTe depositado pela técnica CSS: a) sem tratamento térmico; b) tratado a 430°C por 30 min (CRUZ et al.)

[voltar para RECOZIMENTO SIMULADO](#)

2.2. A Tradução Matemática: o algoritmo básico

O método de Recozimento Simulado (*simulated annealing*) é um tipo de método de busca local de implementação extremamente simples proposto por METROPOLIS *et al.* (1953), que perceberam que a natureza faz na verdade a minimização da energia da estrutura cristalina quando o material é recozido para remover defeitos de sua estrutura atômica. KIRKPATRICK *et al.* (1983) estenderam o método de otimização termodinâmica de Metropolis para o problema de otimização combinatorial.

Na analogia matemática, o estado do sistema termodinâmico corresponde à solução atual do problema combinatorial, a equação de energia para o sistema termodinâmico é a função objetivo e o estado de referência é análogo ao mínimo global da função. A maior dificuldade na implementação do algoritmo é o fato de não haver parâmetro análogo no problema combinatorial à temperatura. Além disto, o que evita o algoritmo de ficar preso em um mínimo local (conhecido como “quenching” ou têmpera, na metalurgia) é a seleção adequada da programação do recozimento. Escolher uma programação significa definir os seguintes fatores: a temperatura inicial, o número de iterações do algoritmo com a mesma temperatura e a estratégia de redução de temperatura ao longo da execução do algoritmo. Os átomos de um cristal se movem pouco a pouco, passando de uma estrutura para uma estrutura muito próxima da anterior, chamada de *estrutura vizinha*. A definição de uma estrutura de vizinhança pode ser muito simples ou bastante complexa, a depender da aplicação considerada.

No algoritmo básico, a partir de um ponto no espaço de soluções calcula-se um novo ponto *vizinho* do atual. Se a energia (o valor da função objetivo) é menor neste novo ponto, este passa a ser o nosso ponto atual: um ponto seu vizinho é calculado e o algoritmo continua. Se a energia é

maior neste novo ponto, ele não é automaticamente descartado. Há uma certa probabilidade de ele ser aceito como o novo ponto atual, e esta probabilidade é tão maior quanto maior for o parâmetro temperatura ou quanto menor for a diferença de energia entre os dois pontos.

Queremos reduzir o valor da energia, mas ainda assim aceitamos como nova solução um ponto que seja pior que o anterior? O motivo para esta aparente contradição é simples: para que se vá de um mínimo local a um mínimo global da estrutura acessando apenas pontos vizinhos, é necessário que, pelo menos em alguns pontos ao longo desta trajetória, a energia do sistema aumente. Em outras palavras, a única maneira de ir de um mínimo local para o mínimo global (ou para outro mínimo local) é aumentando a energia do sistema. A Figura 2 ilustra este fato.

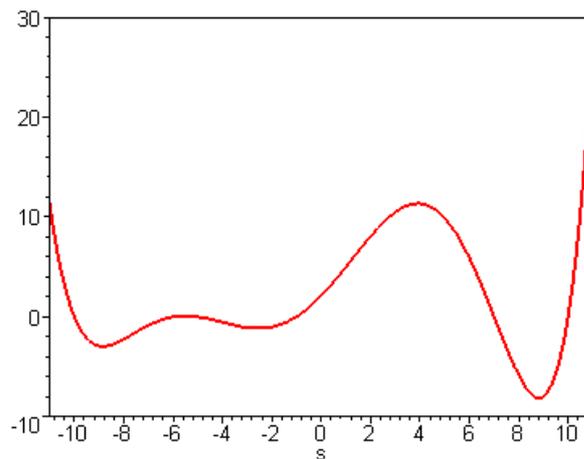


Figura 2 – Para ir de um mínimo local (em $x=-2$) para o mínimo global (em $x=9$) é necessário passar por regiões de máximo (em torno de $x=4$).

Nas primeiras iterações do algoritmo, a temperatura está alta. A probabilidade de aceitar uma transição para um ponto de maior energia é alta. Com isto, o método consegue sair de bacias de atração de mínimos locais e varrer bem o espaço de soluções. Lentamente, com o passar das iterações, a temperatura vai sendo reduzida segundo um esquema de resfriamento pré-definido. As transições para pontos de maior energia vão ficando cada vez menos prováveis, e o método passa a fazer mais exploração local do que exploração global no espaço de soluções. Ao final do algoritmo, a temperaturas muito baixas, apenas são aceitas transições para soluções melhores, e o método eventualmente levará ao mínimo de uma bacia de atração.

O algoritmo básico está representado na Figura 3. As etapas mais sensíveis do método são: a seleção da temperatura inicial, a definição do esquema de resfriamento, a determinação do número de iterações em cada temperatura ($n_{X_{max}}$) e o número de ciclos de resfriamento ($n_{T_{max}}$). Estes aspectos serão detalhados nas próximas seções.

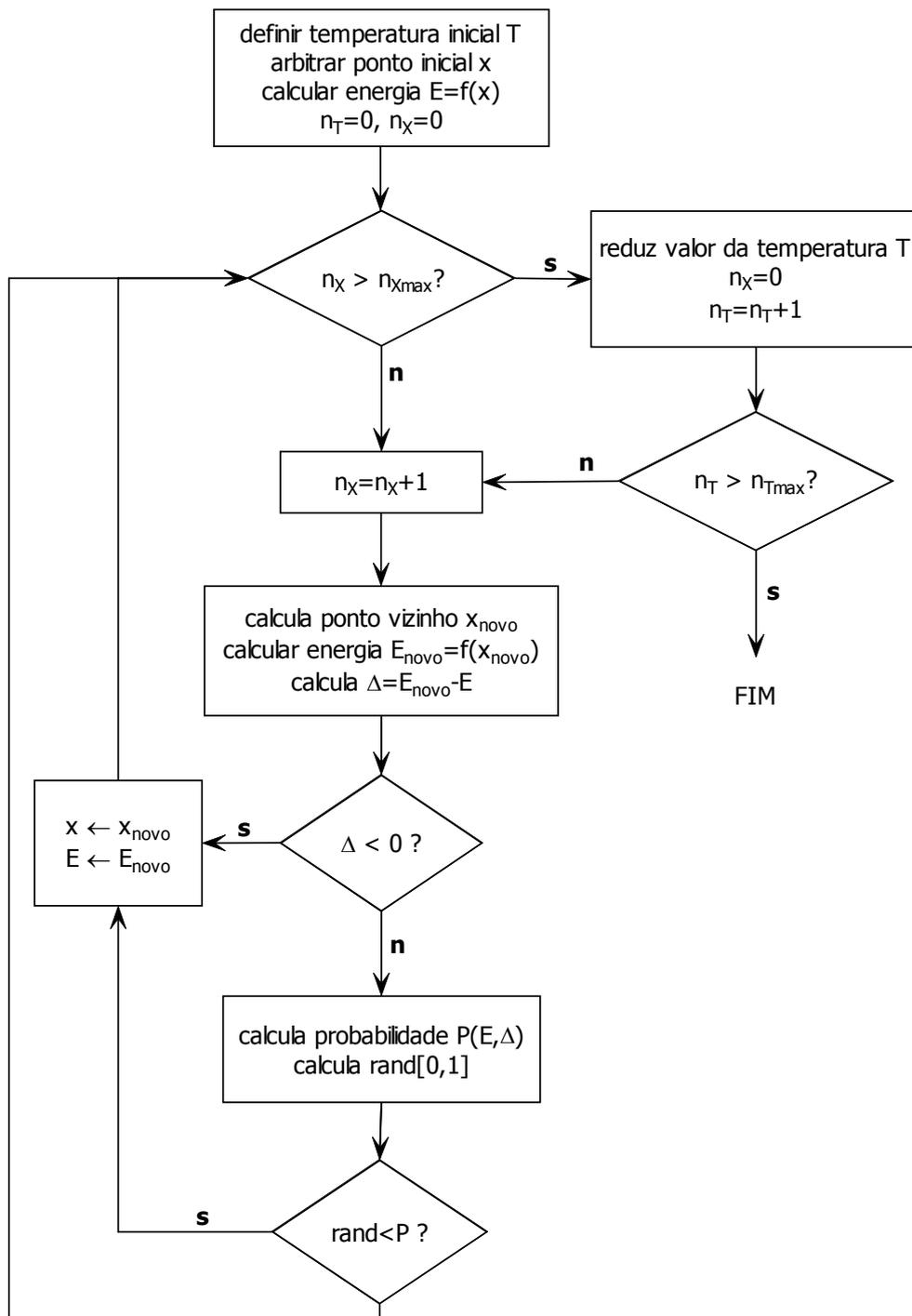


Figura 3 – Algoritmo básico do Recozimento Simulado

[voltar para RECOZIMENTO SIMULADO](#)

2.3. A Programação de Recozimento

Quando a temperatura é muito baixa, o algoritmo aceita apenas transições que melhoram a solução atual. Este algoritmo é conhecido como um *hill climber* (“escalador de morros”, em tradução literal). O sistema rapidamente encontrará um mínimo local e nenhuma melhora adicional será conseguida, independente do número de iterações permitidas. Por outro lado, se a temperatura

for muito grande, praticamente qualquer transição é aceita, e o algoritmo se comportará como um algoritmo de busca aleatória simples.

Pode ser demonstrado que, se a temperatura for reduzida de forma lenta o suficiente (o inverso do logaritmo do tempo) a probabilidade de o método levar ao mínimo global em $T=0$ tende a um. Este esquema de resfriamento tão lento torna a minimização de qualquer função inviável, e o interesse nesta prova de convergência é mais científico que prático.

Qualquer função monótona decrescente pode ser usada como função de resfriamento. Uma escolha muito usual é a redução linear da temperatura, fazendo $T^{k+1} = \alpha T^k$. O parâmetro α varia tipicamente entre 0,7 e 0,95. Funções exponenciais de resfriamento são também utilizadas na literatura da área.

Como definir os parâmetros do algoritmo? Como uma regra geral, o número de temperaturas empregadas (n_1) deve ser pequeno – algo entre 10 e 20 costuma ser suficiente. Já o número de iterações com a mesma temperatura (n_2) deve ser um pouco maior, para que se dê tempo suficiente ao algoritmo para buscar o espaço de soluções com aquela temperatura. Se este número for grande demais, não só o esforço computacional será desnecessariamente alto com o algoritmo poderá ficar vagando pelo espaço, sem se concentrar em uma região mais promissora. Algo entre 20 e 100 iterações são empregadas, a depender da complexidade do espaço de soluções e dos recursos computacionais disponíveis.

Apenas em uma coisa todos os trabalhos da área concordam: experimentação é fundamental para que se definam os parâmetros do método.

[voltar para RECOZIMENTO SIMULADO](#)

2.4. Critério de Aceitação de Soluções de Maior Energia

O critério proposto por Metropolis consiste em avaliar a diferença de energia entre a solução atual e a nova solução e calcular $\Delta \equiv E^{k+1} - E^k$. Se Δ é negativo, isto significa que a solução atual é melhor que a anterior, e esta última é substituída. Se Δ é positivo, a probabilidade de esta solução de maior energia substituir a anterior é dada por $p = \exp(-\Delta/T)$.

Transições muito ruins (ou seja, com Δ grande) são menos prováveis que as transições “pouco ruins”. Quanto menor a temperatura, menos provável é que uma transição ruim seja aceita. As figuras abaixo mostram o comportamento da função probabilidade. Observe que para um valor de Δ/T igual a 5 (Δ 5 vezes maior que a temperatura) a probabilidade de aceitação é praticamente nula, e que a probabilidade de aceitação de 50% corresponde a $\Delta/T = \ln(2) = 0,69$.

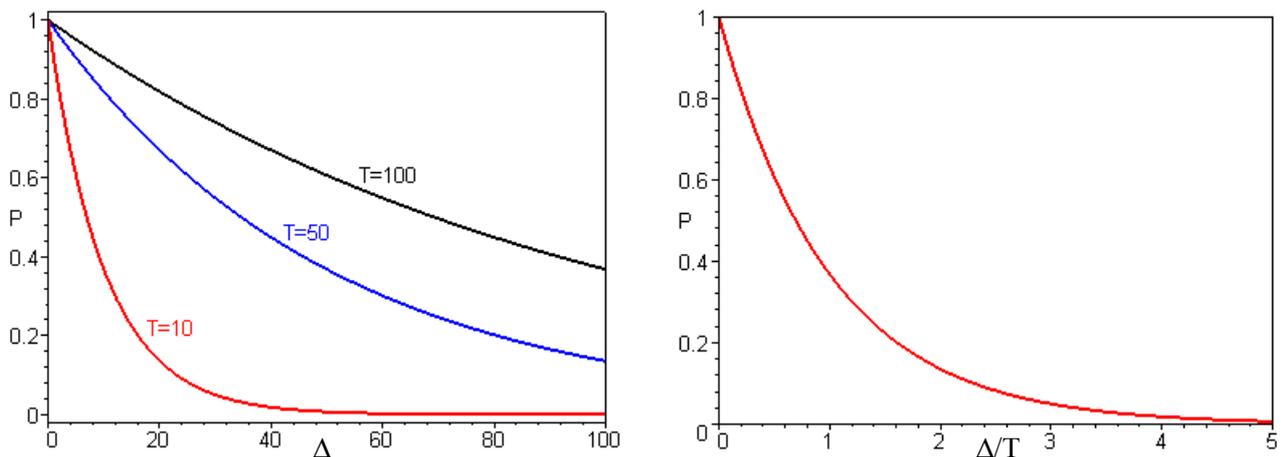


Figura 4 – Probabilidade de aceitação de uma transição de maior energia:
 (a) $P \times \Delta$ para vários valores de temperatura T ; (b) $P \times (\Delta/T)$.

Como implementar este critério probabilístico de aceitação? Primeiramente é sorteado um número randômico r a partir de uma distribuição uniforme no intervalo $[0,1]$. Considere que p representa a probabilidade de a transição para este novo ponto ser aceita. A probabilidade do número sorteado r ser menor ou igual do que p , também contido no intervalo $[0,1]$, é $(100p)\%$. Esta é uma característica da distribuição uniforme que foi discutida na Aula 1. Logo, se $r < p$ a transição é aceita. Do contrário, este ponto é descartado e um novo vizinho do ponto atual é calculado.

Os valores inicial e final da temperatura dependem da escala da sua função. Lembre-se que há 50% de chance de aceitar uma transição para uma solução pior quando $\Delta/T = 0,69$. Alguns trabalhos na literatura sugerem que sejam definidos os parâmetros de resfriamento tais que 80% das transições pobres sejam aceitas no início do algoritmo. No entanto, este critério é difícil de ser aplicado na prática, e nem sempre corresponde à melhor escolha de parâmetros.

[voltar para RECOZIMENTO SIMULADO](#)

2.5. A Estrutura de Vizinhança

A definição da estrutura de vizinhança é fundamental para a implementação do algoritmo. Uma estrutura de vizinhança nada mais é do que uma maneira de calcular uma solução próxima à solução anterior. Para um mesmo problema, várias estruturas de vizinhança são cabíveis. A vizinhança de um ponto corresponde ao conjunto de todos os pontos que lhe são vizinhos.

Por exemplo, imagine o problema de minimização de uma função com domínio no \mathbb{R}^2 . Podemos calcular um ponto “vizinho” no plano real de várias maneiras. Poderíamos: (i) somar (ou subtrair) uma quantidade fixa δ em uma ou mais direções; (ii) dar um passo de tamanho δ em alguma direção arbitrária; (iii) selecionar qualquer ponto que diste não mais que δ do ponto anterior. Estas possíveis vizinhanças de tamanho δ estão representadas na Figura 5, respectivamente nas cores vermelha, azul e verde.

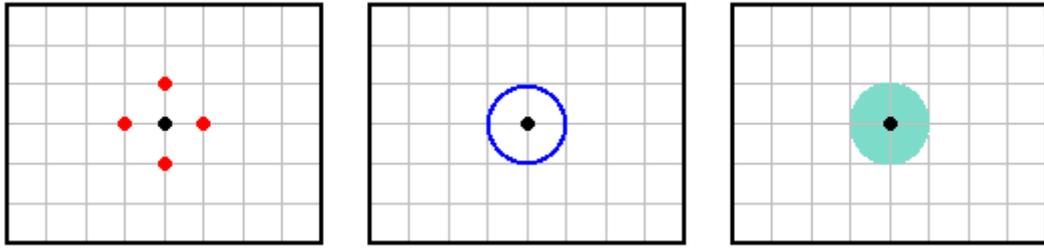


Figura 5- Possíveis estruturas de vizinhança no plano real (quadriculado de $\delta \times \delta$).

A definição da estrutura de vizinhança em problemas de otimização de estruturas de moléculas é menos intuitiva, uma possibilidade seria definir uma molécula vizinha como uma molécula em que todos os grupos funcionais são iguais exceto um. A figura abaixo mostra dois pares de moléculas vizinhas.

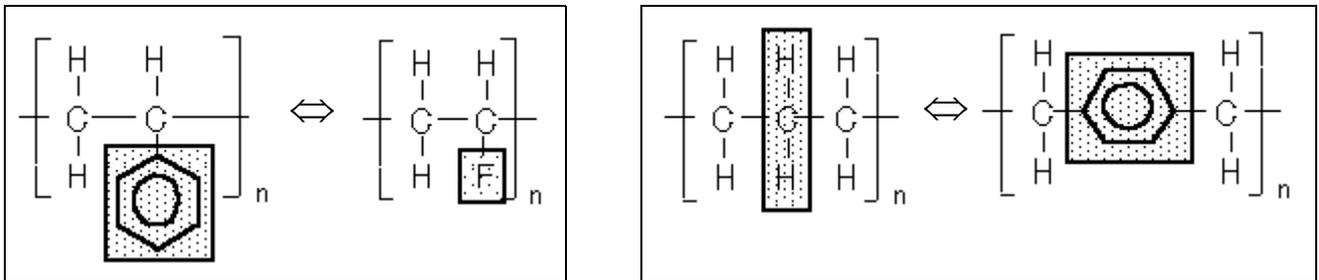


Figura 6- Pares de moléculas vizinhas.

[voltar para RECOZIMENTO SIMULADO](#)

2.6. Aspectos de Implementação do Algoritmo

Uma característica do recozimento simulado: não há qualquer garantia de que o último ponto visitado será o melhor ponto já visitado pelo algoritmo. Uma modificação simples e efetiva no algoritmo original consiste em registrar em uma variável à parte este melhor resultado encontrado. Assim, é possível que o seu problema seja resolvido mesmo com uma programação de recozimento pobre.

A figura abaixo mostra o resultado de execuções do código onde, por uma escolha inadequada de parâmetros, o mínimo global da solução foi visitado mas abandonado em seguida. Para evitar este problema, a melhor solução encontrada ao longo da execução do algoritmo foi registrada à parte. Observe que o ótimo global foi encontrado nas duas situações.

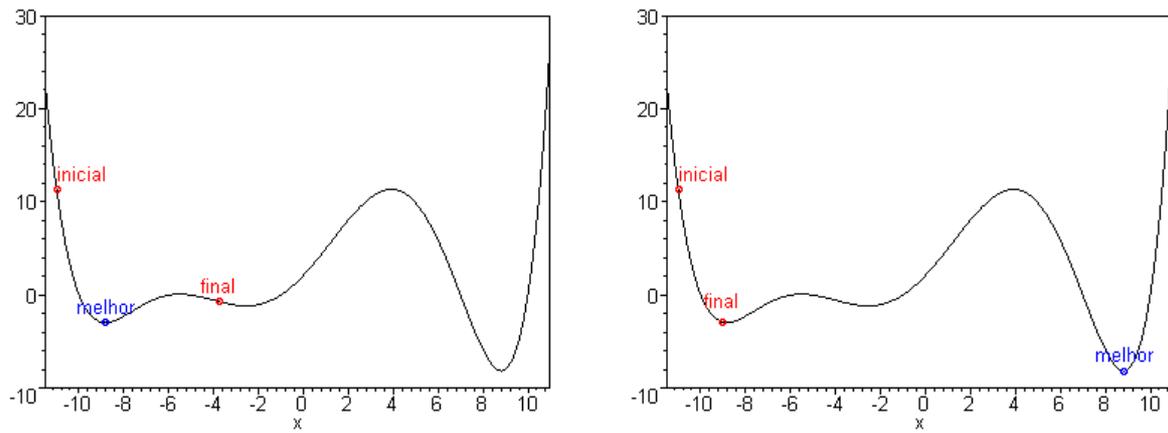


Figura 7 – Possíveis resultados de otimização onde o ótimo foi visitado e abandonado

Uma outra sugestão freqüente na literatura consiste em executar várias vezes o algoritmo com o mesmo grupo de parâmetros, partindo de diversas condições iniciais, seqüencialmente ou em paralelo. Assim, um possível erro na seleção do chute inicial, do gerador de números aleatórios ou da programação de resfriamento tem seu efeito suavizado. O preço a pagar é o drástico aumento no custo computacional.

[voltar para RECOZIMENTO SIMULADO](#)

2.7. Exemplo Ilustrativo

Para entender o significado dos parâmetros do método, considere o problema de minimização da função $F(x)$ apresentada abaixo.

$$F(x) = \frac{1}{10000} (x + 10) (x + 6) (x + 5) (x + 1) (x - 7) (x - 10)$$

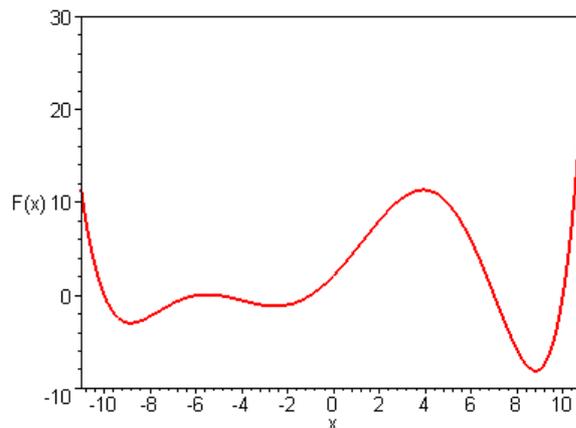


Figura 8 – Espaço de busca da função teste

Observe que existem mínimos locais em $x = -8,834$, $x = -2,546$ e $x = 8,817$, sendo este último o mínimo global. A função apresenta máximos locais em $x = -5,518$ e $x = 3,914$. Um método

tipo gradiente não convergiria para o mínimo global desejado para qualquer condição inicial:

- se arbitrado $x < -5,518$, o método convergiria para $x = -8,834$;
- se arbitrado $-5,518 < x < 3,914$, o método convergiria para $x = -2,546$;
- se arbitrado $x > 3,914$, o método convergiria para $x = 8,817$.

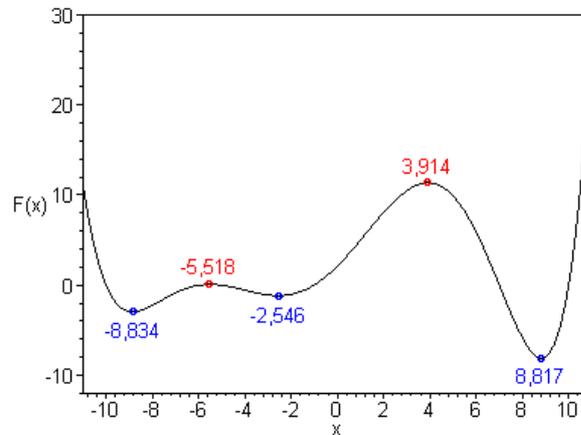


Figura 9 – Extremos da função teste

Foi implementado o algoritmo de Recozimento Simulado apresentado na Figura 3. Foi empregado um esquema simples de resfriamento, tal que $T^{k+1} = \alpha T^k$. Os parâmetros empregados foram $\alpha=0,85$, $T_0=30$, $n_1 = 15$ e $n_2 = 90$. O código implementado em Maple está apresentado na *homepage* do curso.

Os resultados abaixo correspondem a algumas execuções do algoritmo. Cabe lembrar que a cada execução o código levará a uma seqüência de tentativas diferente por conta dos números aleatórios envolvidos. Isto justifica a diversidade dos gráficos. Foi implementada a modificação comentada no texto: o melhor ponto visitado ao longo da execução do algoritmo é armazenado. Assim, estão mostrados nos gráficos abaixo o ponto de partida do algoritmo, o resultado final fornecido (ponto de parada do algoritmo) e o melhor ponto visitado ao longo da execução.

Apenas como um teste, foi alterado o parâmetro α que controla a velocidade de resfriamento. Observe que um resfriamento rápido demais levou o método a ficar preso em um mínimo local (Figura 10a). O resfriamento na taxa adequada levou o algoritmo ao mínimo global desejado (Figura 10b).

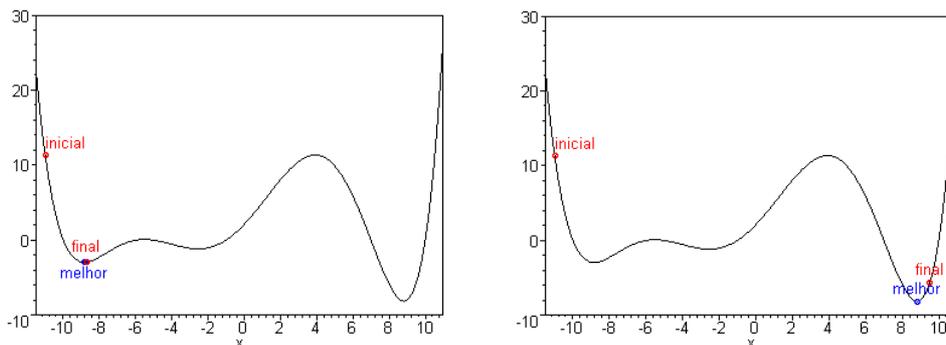


Figura 10 – Possíveis resultados de otimização empregando Recozimento Simulado: (a) $\alpha=0,75$; (b) $\alpha=0,80$.

O resfriamento lento demais, mantidos todos os demais parâmetros, fez com que o algoritmo ficasse sempre com a temperatura muito alta. Com isto, foi feita pouca exploração local, e o algoritmo passou a maior parte do tempo se comportando como um algoritmo de busca aleatória exaustiva. Encontrar o mínimo é questão de sorte: o método pode ou não passar por perto do mínimo local, como se observa nas Figuras 11a e 11b.

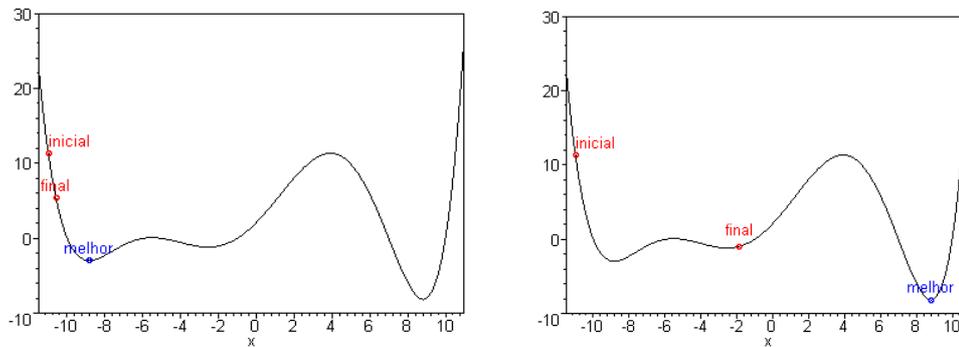


Figura 11 – Possíveis resultados de otimização empregando Recozimento Simulado com $\alpha=0,90$

Não há um conjunto de parâmetros reconhecidamente bom para o algoritmo, que se aplique a qualquer sistema sendo otimizado. Experimentação é fundamental para o sucesso da otimização.

[voltar para RECOZIMENTO SIMULADO](#)