

# Simulated annealing

Origem: Wikipédia, a enciclopédia livre.

**Recozimento simulado** (ou ***Simulated Annealing***) é uma meta-heurística para otimização que consiste numa técnica de busca local probabilística, e se fundamenta numa analogia com a termodinâmica.

A metaheurística usada é uma metáfora de um processo térmico, dito recozimento ou *annealing*, utilizado em metalurgia para obtenção de estados de baixa energia num sólido. O processo consiste de duas etapas: na primeira, a temperatura do sólido é aumentada para um valor próximo de 1100°C, que é a temperatura de início de transformação da fase perlítica em austenita; na segunda, o resfriamento deve ser realizado lentamente até que o material se solidifique, sendo acompanhado e controlado esse arrefecimento. Nesta segunda fase, executada lentamente, os átomos que compõem o material organizam-se numa estrutura uniforme com energia mínima. Isto resulta em que os átomos desse material ganhem energia para se movimentarem livremente e, ao arrefecer de forma controlada, dar-lhes uma melhor hipótese de se organizarem numa configuração com menor energia interna, para ter uma redução dos defeitos do material, como resultado prático.

De forma análoga, o algoritmo de Arrefecimento Simulado substitui a solução atual por uma solução próxima (i.e., na sua vizinhança no espaço de soluções), escolhida de acordo com uma função objetivo e com uma variável ***T*** (dita *Temperatura*, por analogia). Quanto maior for ***T***, maior a componente aleatória que será incluída na próxima solução escolhida. À medida que o algoritmo progride, o valor de ***T*** é decrementado, começando o algoritmo a convergir para uma solução ótima, necessariamente local.

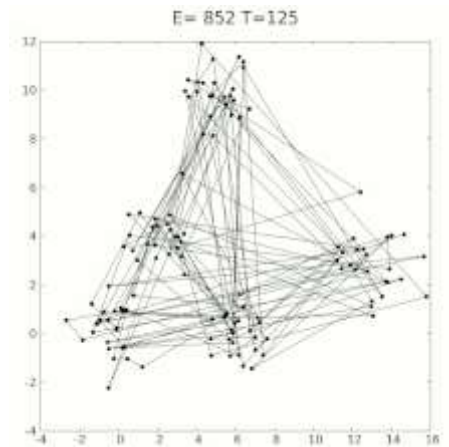
Uma das principais vantagens deste algoritmo é permitir testar soluções mais distantes da solução atual e dar mais independência do ponto inicial da pesquisa.

## Descrição

Esta técnica começa sua busca a partir de uma solução inicial qualquer. O procedimento principal consiste em um loop ou laço que a cada iteração, gera aleatoriamente um único vizinho *s'* da solução corrente *s*.

A cada geração de um novo vizinho *s'* de *s*, é testada a variação  $\Delta$  do valor da função objetivo, isto é,  $\Delta = f(s') - f(s)$ , onde temos as seguintes situações:

- $\Delta < 0$ : Há uma redução de energia, a qual implica que a nova solução é melhor que a anterior. O método aceita a solução e *s'* passa a ser a nova solução corrente.
- $\Delta = 0$ : Caso de estabilidade, não havendo redução de energia. Na verdade, situação pouco provável de acontecer na prática. A aceitação da solução é, portanto, indiferente.



O Simulated Annealing é aplicado ao problema do Caixeiro Viajante. Existem 125 nós em 5 blocos. O número de combinações possíveis é cerca de  $1,5e+207$ .

- $\Delta > 0$ : Houve um aumento do estado de energia. A aceitação desse tipo de solução é mais provável a altas temperaturas e bastante improvável a temperaturas reduzidas. Para reproduzir essas características, geralmente usa-se, para calcular a probabilidade de se aceitar a nova solução, uma função conhecida por fator de Boltzmann, que é dada por  $e^{-\Delta/T}$ , onde  $T$  é um parâmetro do método, chamado de temperatura e que regula a probabilidade de soluções com pior custo. Por exemplo, esta poderá ser:
  - Gera-se um número aleatório retirado de uma distribuição uniforme no intervalo  $[0, 1]$ .
  - Se este número for menor ou igual a “ $p$ ”, aceita-se a solução.
  - Se for maior que “ $p$ ”, rejeita-se a solução.

A temperatura  $T$  assume inicialmente um valor elevado,  $T_0$ . Após um número fixo de iterações (o qual representa o número de iterações para o sistema atingir o equilíbrio térmico em uma dada temperatura), a temperatura é gradativamente diminuída por uma razão de resfriamento  $\alpha$ , tal que  $T_n \leftarrow \alpha * T_{n-1}$ , sendo  $0 < \alpha < 1$ . Como esse procedimento se dá no início, há uma chance maior de se escapar de mínimos locais e, à medida que  $T$  se aproxima de zero, o algoritmo se comporta como o método de descida, uma vez que diminui a probabilidade de se aceitar movimentos que possa piorar ( $T \rightarrow 0 \Rightarrow e^{-\Delta/T} \rightarrow 0$ ).

O procedimento é finalizado quando a temperatura chega a um valor próximo de zero e nenhuma solução que piore o valor da melhor solução seja mais aceita, ou seja, quando o sistema estiver estável. A solução obtida quando o sistema encontra-se nesta situação evidencia o encontro de um mínimo local.

Algoritmos baseados em Simulated Annealing geralmente incluem reaquecimento seguido de um novo processo de resfriamento, utilizado quando a quantidade de movimentos consecutivamente rejeitados é alta. É também comum trabalhar nas temperaturas mais altas com taxa de resfriamento menor e aumentá-la quando a temperatura reduzir.

## Pseudo-código

---

Antes de apresentar o algoritmo, vejam-se os identificadores neles utilizados:

- $S_0 \rightarrow$  Configuração Inicial (Entrada);
- $S_i \rightarrow$  Configuração da Iteração  $i$ ;
- $S \rightarrow$  Configuração Final;
- $T_0 \rightarrow$  Temperatura Inicial;
- $T_i \rightarrow$  Temperatura na Iteração  $i$ ;
- $M \rightarrow$  Número máximo de iterações (Entrada);
- $P \rightarrow$  Número máximo de Perturbações por iteração (Entrada);
- $L \rightarrow$  Número máximo de sucessos por iteração (Entrada);
- $\alpha \rightarrow$  Factor de redução da temperatura (Entrada);
- $f(S_i) \rightarrow$  Valor da função objetivo correspondente á configuração  $S_i$ ;
- $nSucesso \rightarrow$  Contador de sucesso em uma iteração;
- $i$  e  $j \rightarrow$  Variáveis de controle de Loops.

Além dos indicadores acima, consideremos as seguintes funções:

- $Perturba(S) \rightarrow$  Função que realiza uma perturbação na Solução  $S$ ;
- $Randomiza() \rightarrow$  Função que gera um número aleatório no intervalo  $[0,1]$ ;
- $Templnicial() \rightarrow$  Função que calcula a temperatura inicial;

1.

## Pseudo-Código Simulated Annealing

Início

```
/* Entradas do Algoritmo */
```

```
Ler (S0, M, P, L, α)
```

```
/* Inicialização das variáveis */
```

```
S = S0
```

```
T0 = TempInicial()
```

```
T = T0
```

```
j = 1
```

```
/*Loop principal - Verifica se foram atendidas as condições de termino do algoritmo*/
```

```
Repita
```

```
    i = 1
```

```
    nSucesso = 0
```

```
    /*Loop Interno – Realização de perturbação em uma iteração*/
```

```
    Repita
```

```
        Si = Perturba(S)
```

```
        ΔFi = f(Si) – f(S)
```

```
        /*Teste de aceitação de uma nova solução*/
```

```
        Se (Δfi ≤ 0) ou (exp(-Δfi/T) > Randomiza()) então
```

```
            S= Si
```

```
            nSucesso = nSucesso + 1
```

```
        Fim-se
```

```
        i = i + 1
```

```
Até (nSucesso ≥ L) ou (i > P)
```

```
/*Actualização da Temperatura*/
```

```
T = α.T
```

```
/*Actualização do Contador de iterações*/
```

```
j = j + 1
```

```
Até (nSucesso = 0) ou (j > M)
```

```
/*Saída do Algoritmo*/
```

```
Imprima(S)
```

## Veja também

---

- Método de Monte Carlo
  - Pesquisa tabu
  - Cadeias de Markov
  - Problema do caixeiro viajante
- 

Obtida de "[https://pt.wikipedia.org/w/index.php?title=Simulated\\_annealing&oldid=64086997](https://pt.wikipedia.org/w/index.php?title=Simulated_annealing&oldid=64086997)"