

# Aula 5

## Algoritmos Metaheurísticos

### Busca Tabu e Simulated Annealing

**Prof. Dr. Peterson A. Belan**

# Busca Tabu (BT)

A Busca Tabu é um método iterativo de otimização local que admite soluções de piora para escapar de ótimos locais. A cada iteração seleciona-se o melhor vizinho  $s'$  do estado corrente  $s$ . Independentemente de  $s'$  ser melhor ou pior que  $s$ ,  $s'$  será o novo estado corrente. Se  $s'$  for melhor que a melhor solução encontrada até o momento  $s^*$ , então esta é substituída por  $s'$ .

Como esse mecanismo não é suficiente para escapar de ótimos locais, uma vez que poderia haver retorno a uma solução previamente gerada, o algoritmo usa o conceito de lista tabu, a qual registra os estados que já foram visitados. O algoritmo chega ao fim quando alcança um certo critério de parada, geralmente número de iterações sem melhoras.

# Busca Tabu (BT)

## Início

Seja  $s_0$  a solução inicial (gerada aleatoriamente),  $s$  a solução representada pelo estado corrente,  $s'$  a solução representada pelo melhor vizinho do estado corrente,  $s^*$  a melhor solução obtida até então, Iter o contador do número de iterações, Melhor\_Iter a iteração mais recente que forneceu  $s^*$ , Max\_Iter\_sm o número máximo de iterações sem melhoria de  $s^*$  e  $T$  a lista tabu

$T \leftarrow \emptyset$

$s^* \leftarrow s_0$

$s \leftarrow s_0$

Iter  $\leftarrow 1$

Melhor\_Iter  $\leftarrow 1$

Max\_Iter\_sm  $\leftarrow 50$

$T \leftarrow T \cup s_0$

Enquanto (Iter – Melhor\_Iter  $\leq$  Max\_Iter\_sm) Faça

    Seleciona sucessor  $s' \{s' \notin T\}$

$s \leftarrow s'$

$T \leftarrow T \cup s$

    Se  $f(s) > f(s^*)$  Então

$s^* \leftarrow s$

        Melhor\_Iter  $\leftarrow$  Iter

Fim-se

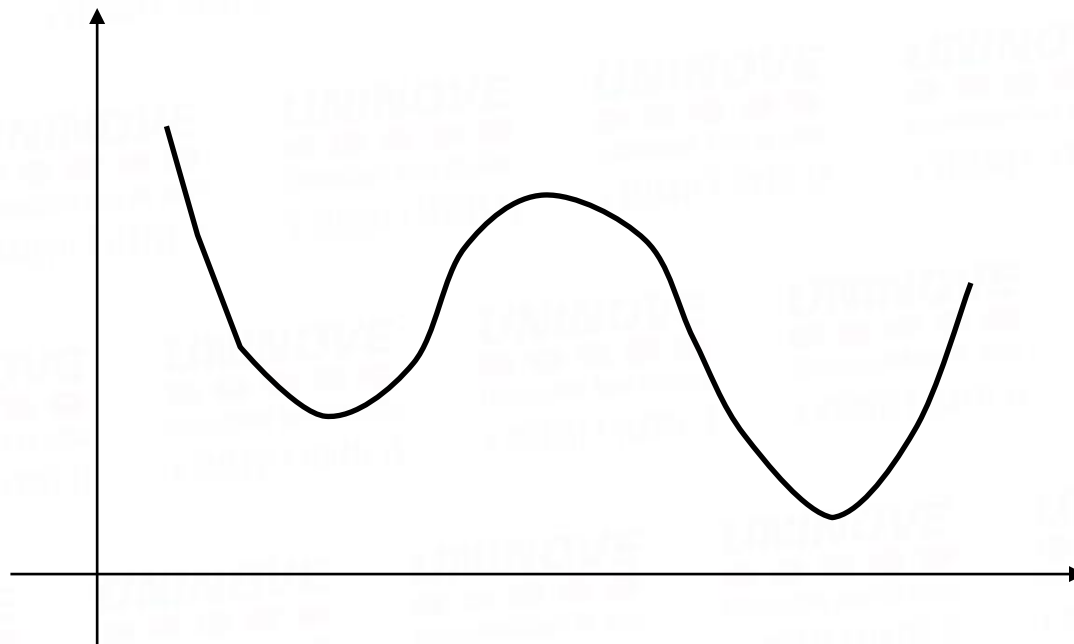
    Iter  $\leftarrow$  Iter + 1

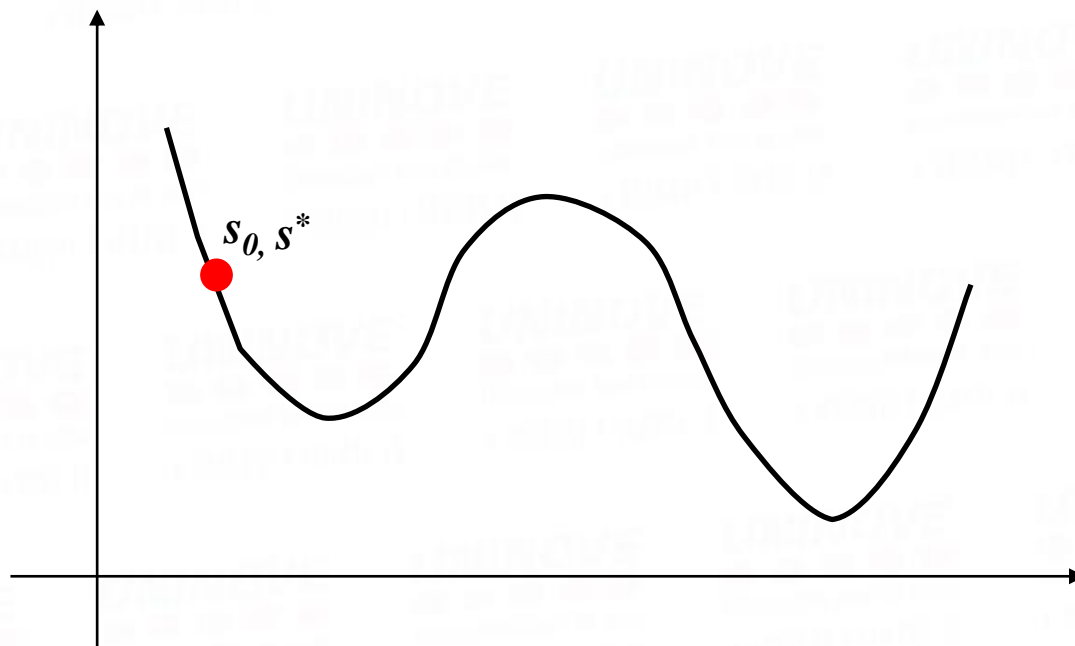
Fim-enquanto

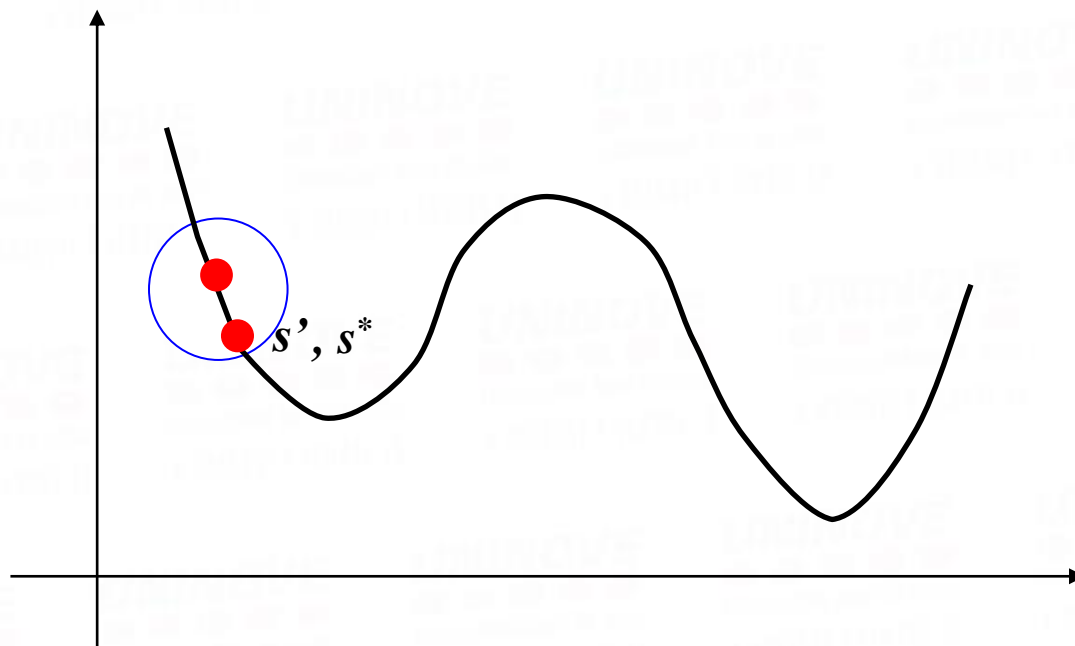
Retorne  $s^*$

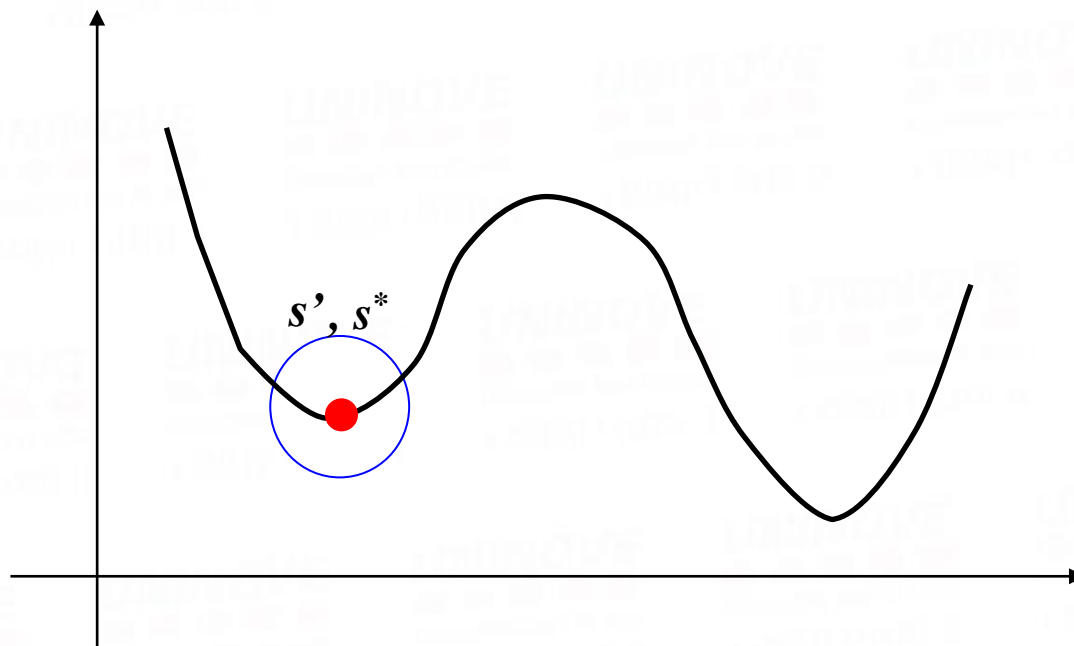
Fim

# Busca Tabu - simulação

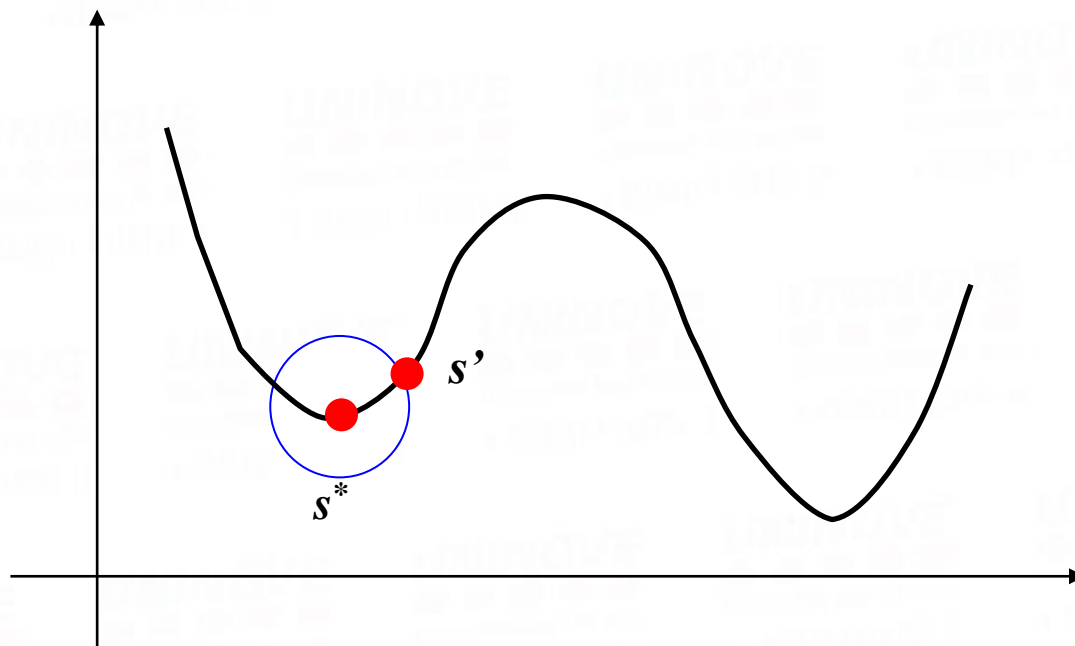








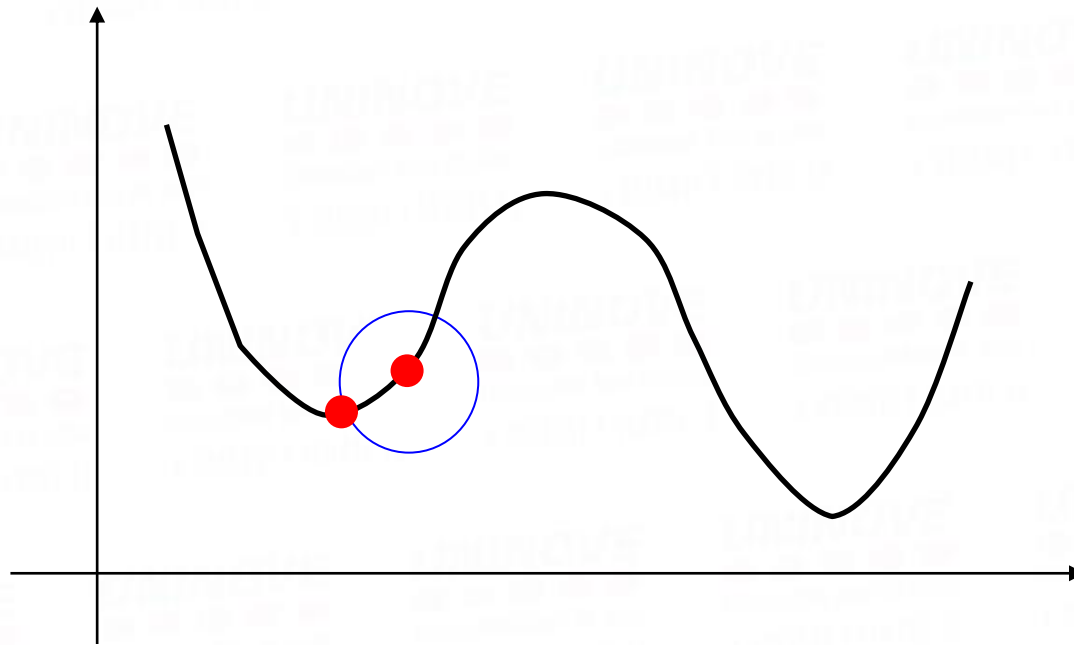
**Problema: Fica-se preso no primeiro ótimo local**



**O melhor vizinho pode ser de piora!**

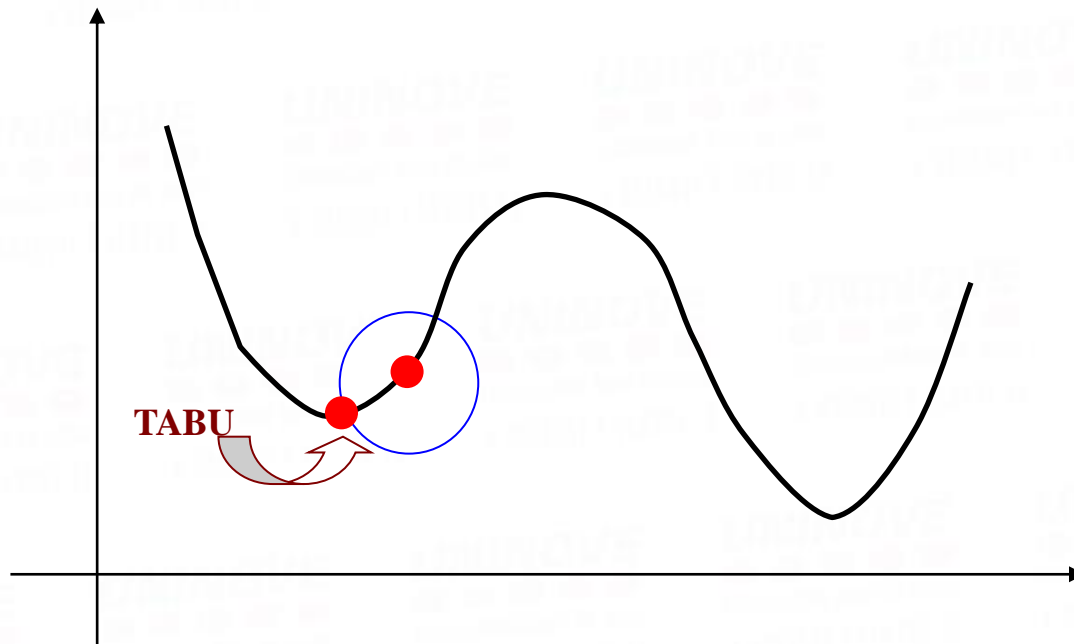


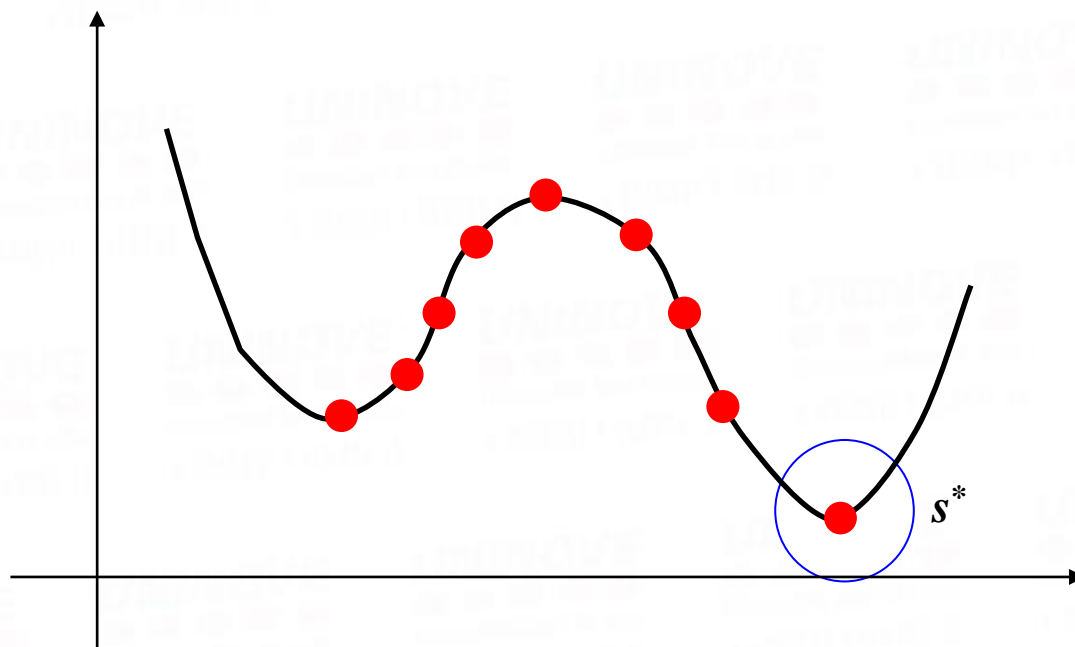
# Mover para o melhor vizinho



**Problema: Ciclagem**

# Ideia: Lista Tabu





# Simulated Annealing (SA)

O algoritmo Simulated Annealing ou algoritmo de arrefecimento simulado, baseado no processo de recozimento de metais, é uma versão estocástica do conhecido algoritmo Subida de Encosta.

Neste algoritmo, movimentos para estados melhores que o estado corrente são sempre aceitos. Caso o movimento seja para um estado de piora, ele pode ser aceito com uma determinada probabilidade que diminui com o tempo. Desta forma, nas iterações finais do algoritmo, somente soluções melhores são aceitas já que a probabilidade de aceitação de uma solução pior é quase nula (RICH; KNIGHT, 1994; RUSSEL; NORVIG, 1995).

# Simulated Annealing (SA)

- Proposto originalmente por Kirkpatrick et al. (1983), fundamenta-se na analogia com a termodinâmica, ao simular o resfriamento de um conjunto de átomos aquecidos.
- O algoritmo SA foi inspirado no algoritmo de Metropolis para a resolução de problemas de otimização.
- Metropolis et al. (1953) propuseram um algoritmo para a simulação de um conjunto de átomos em equilíbrio a uma determinada temperatura.

# Simulated Annealing (SA)

- ✓ SA é um método de busca que aceita movimentos de piora, dada uma certa probabilidade, para escapar de ótimos locais.
- ✓ A probabilidade  $P$  de aceitação de soluções piores depende do parâmetro temperatura ( $Temp$ ) que é um dos principais parâmetros do SA.
- ✓ Deve-se iniciar ( $Temp$ ) com valor alto para evitar convergência precoce para um mínimo local, e decair gradativamente a cada iteração (uma dentre outras alternativas).

# Simulated Annealing (SA)

Início

Seja  $s_0$  a solução inicial (gerada aleatoriamente),  $s$  a solução representada pelo estado corrente,  $s'$  a solução representada pelo sucessor do estado corrente,  $s^*$  a melhor solução obtida até então,  $Iter$  o contador do número de iterações,  $Max\_Iter$  o número máximo de iterações,  $\Delta E$  a variação entre  $f(s)$  e  $f(s')$ ,  $P$  a probabilidade de aceitar soluções piores que a atual,  $Temp$  uma temperatura que controla a probabilidade  $P$

$s \leftarrow s_0$

$s^* \leftarrow s_0$

$Iter \leftarrow 1$

$Max\_Iter \leftarrow Qtd\_Maxima\_Iterações$

Enquanto ( $Iter \leq Max\_Iter$ ) Faça

$Temp \leftarrow Max\_Iter - Iter$

    Se  $Temp=0$  Então

        Retorne  $s^*$

    Fim-se

    Gera sucessor  $s'$  {aleatoriamente}

$\Delta E \leftarrow f(s') - f(s)$

    Se  $\Delta E > 0$  Então

$s \leftarrow s'$

    Senão

$P \leftarrow e^{\Delta E/Temp}$

$s \leftarrow s'$  {somente com uma certa probabilidade  $P$ }

Fim-se

    Se  $f(s) > f(s^*)$  Então

$s^* \leftarrow s$

Fim-se

$Iter \leftarrow Iter + 1$

Fim-enquanto

Retorne  $s^*$

Fim

# Simulated Annealing (SA)

- Para um problema de maximização:
- $\Delta E = \text{fo\_solução sucessora} - \text{fo\_solução atual}$
- Se  $\Delta E > 0$ , a solução sucessora é aceita e passa a ser a solução atual.
- Se  $\Delta E < 0$ , a solução candidata sucessora poderá ser aceita com uma determinada probabilidade  $P$ , onde:

$$P = e^{(\Delta E / T)}$$

$T = \text{Temp}$



# Simulated Annealing (SA)

- Para um problema de minimização:
- $\Delta E = \text{fo\_solução sucessora} - \text{fo\_solução atual}$
- Se  $\Delta E \leq 0$ , a solução sucessora é aceita e passa a ser a solução atual.
- Se  $\Delta E > 0$ , a solução candidata sucessora poderá ser aceita com uma determinada probabilidade  $P$ , onde:

$$P = e^{(-\Delta E / T)}$$

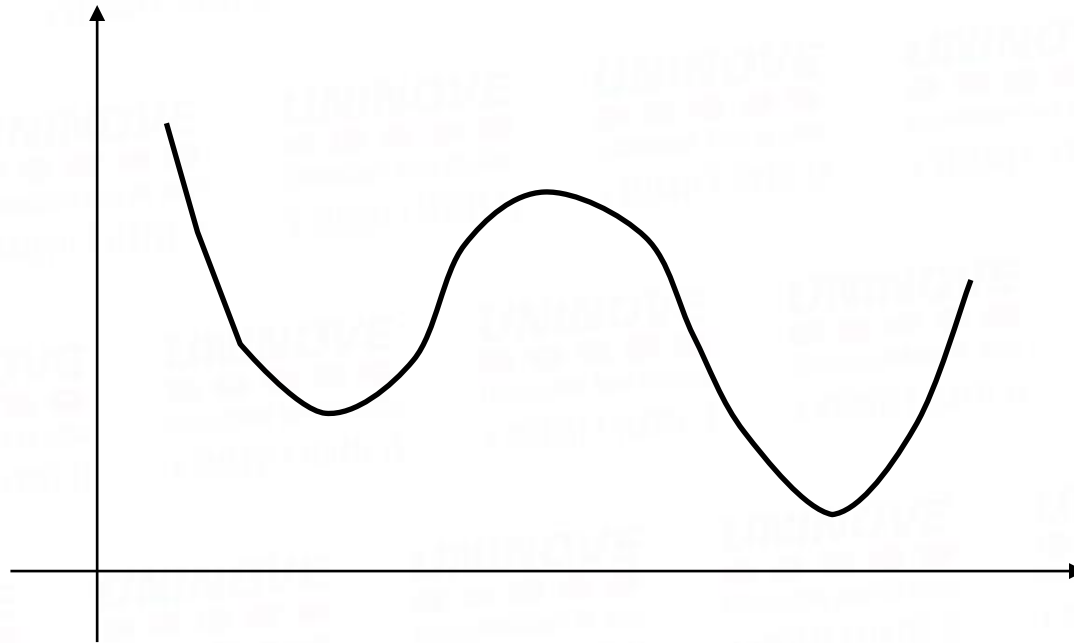
$T = \text{Temp}$

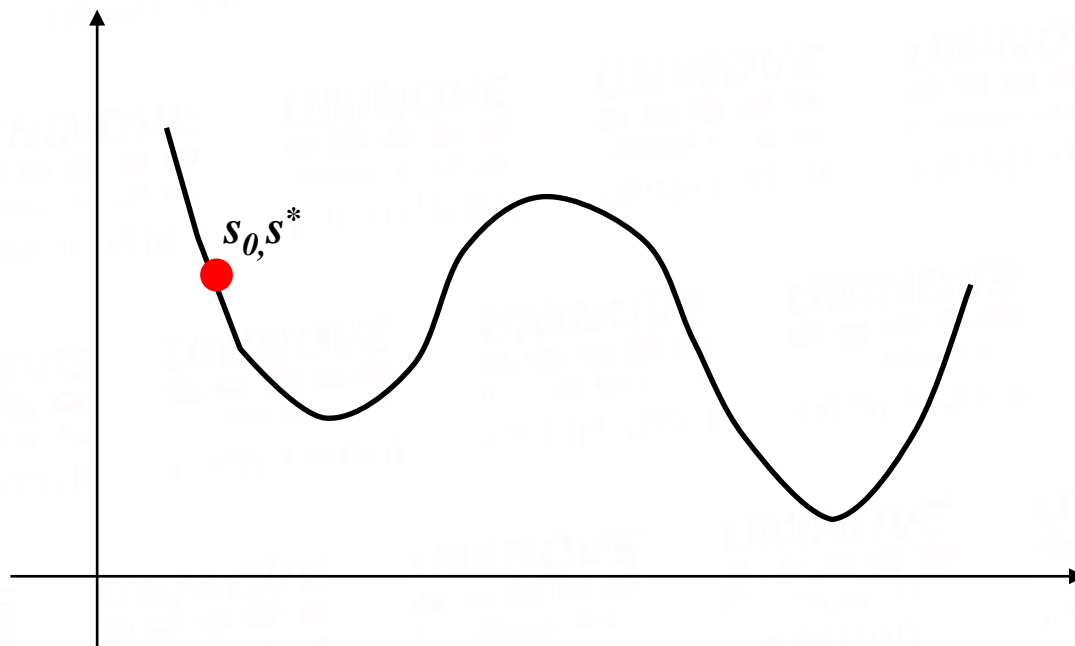
# Simulated Annealing (SA)

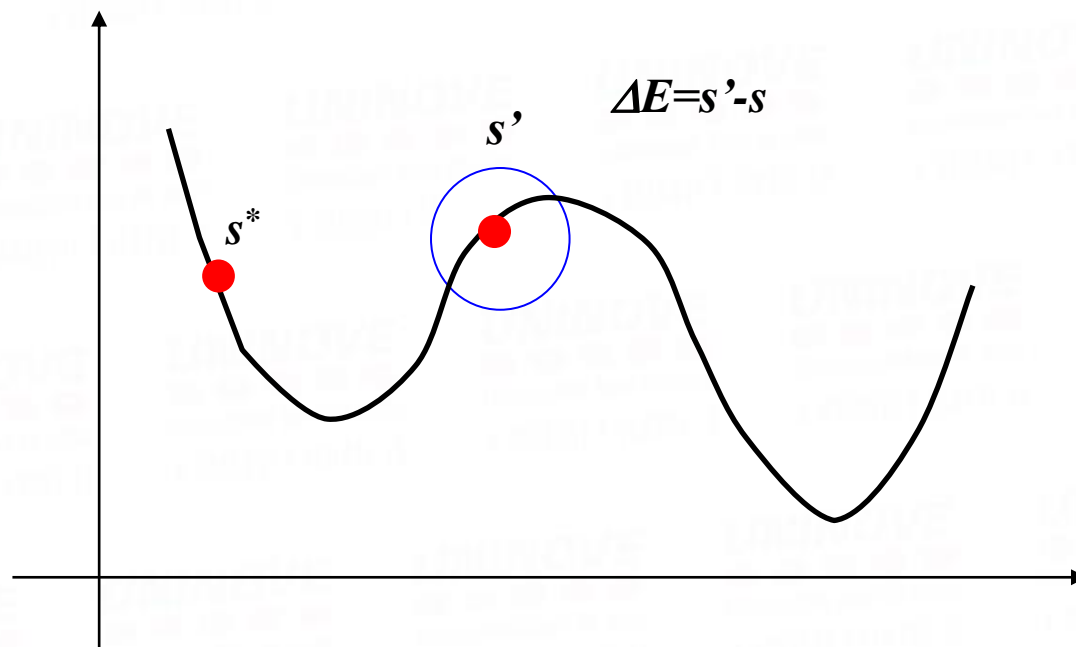
Maximização				
			s' - s	
T	s'	s	DE	$P=e^{(DE/T)}$
100	8	9	-1	0,990
90	8	9	-1	0,989
70	8	9	-1	0,986
50	8	9	-1	0,980
30	8	9	-1	0,967
15	8	9	-1	0,936
5	8	9	-1	0,819
2	8	9	-1	0,607
1	7	9	-2	0,135

Minimização				
			s' - s	
T	s'	s	DE	$P=e^{(-DE/T)}$
100	9	8	1	0,990
90	9	8	1	0,989
70	9	8	1	0,986
50	9	8	1	0,980
30	9	8	1	0,967
15	10	8	2	0,875
5	10	8	2	0,670
2	11	8	3	0,223
1	11	8	3	0,050

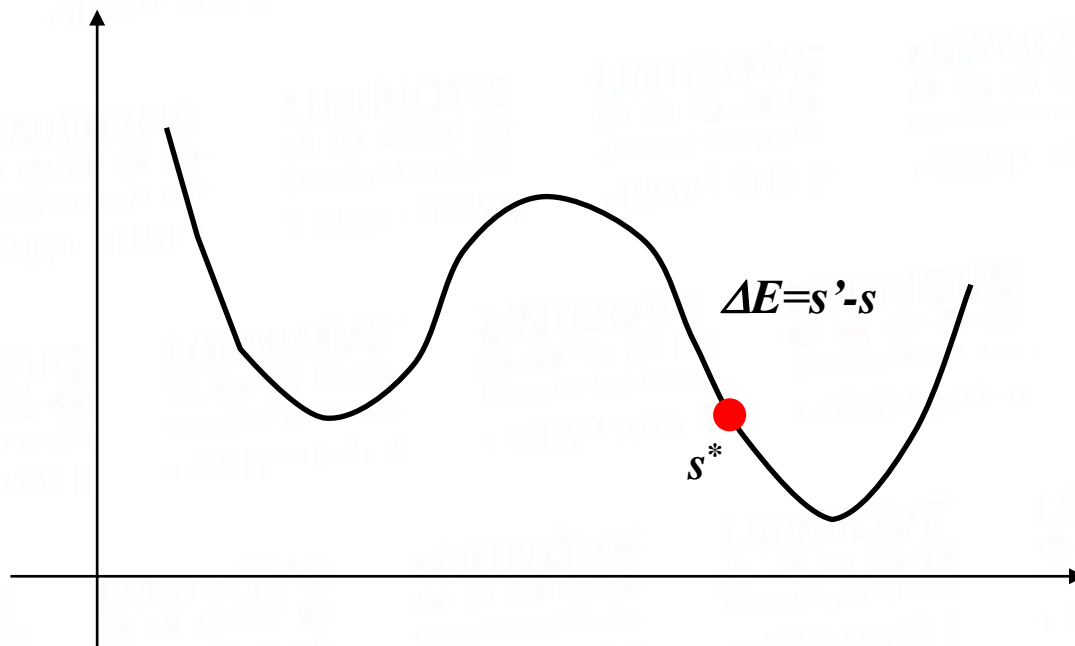
# Simulated Annealing - Simulação

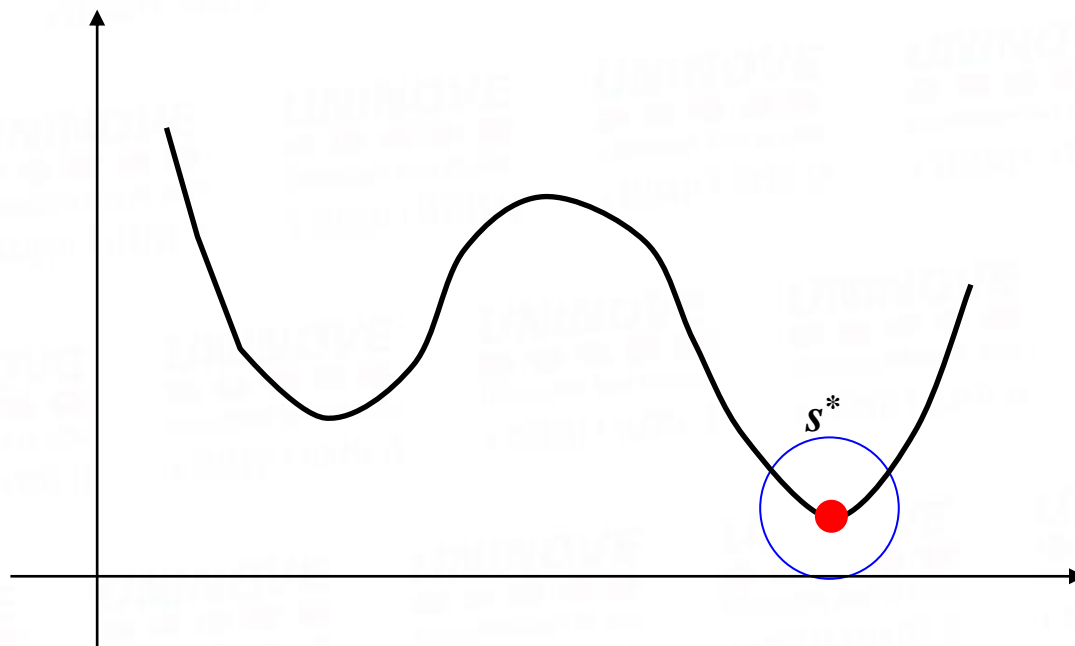






**$s'$  vizinho pode ser de piora!**





# Referências

KIRKPATRICK, S.; GELATTI, C. D.; VECCHI, M. P. Optimization by Simulated Annealing. Science, New Series, v.220, n.4598, p. 671-680, May. 1983.

LUKE, S. Essentials of metaheuristics. 2. ed. Raleigh: Lulu, 2013.

METROPOLIS, N. et al. Equation of state calculations by fast computing machines. Journal of Chemical Physics. v.21, p.1087-1092, 1953.

RUSSELL, S. J.; NORVIG, P. Inteligência artificial . 2. ed. Rio de Janeiro: Campus, 2004.

SOUZA, M. J. F. Inteligência Computacional para Otimização, Notas de aula, Departamento de Computação, Universidade Federal de Ouro Preto, disponível em: <http://www.decom.ufop.br/prof/marcone>.