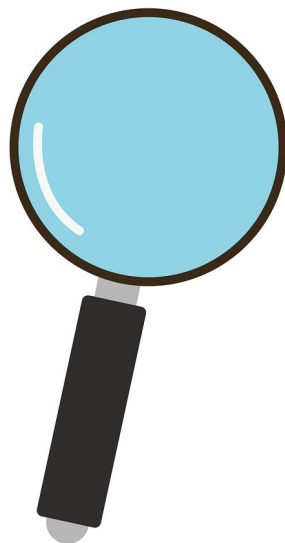


Busca Local



Busca Local



- O caminho da solução não é importante.
- A solução é importante.

Métodos de Busca

Busca Heurística: Subida da encosta



Busca Local



Subida da encosta

- Na busca gulosa, a cada momento escolhemos o próximo estado que aparenta levar a uma solução maximizando (ou minimizando) algum valor.
- No entanto, é mantida uma lista de estados não visitados que podem, em determinado passo, se tornarem mais interessantes heurísticamente.
- Além disso, é preciso guardar os caminhos até cada nó visitado.
- Na subida da encosta, apenas o estado atual é analisado. Não é mantida uma lista de nós abertos, tampouco os caminhos até cada nó visitado.

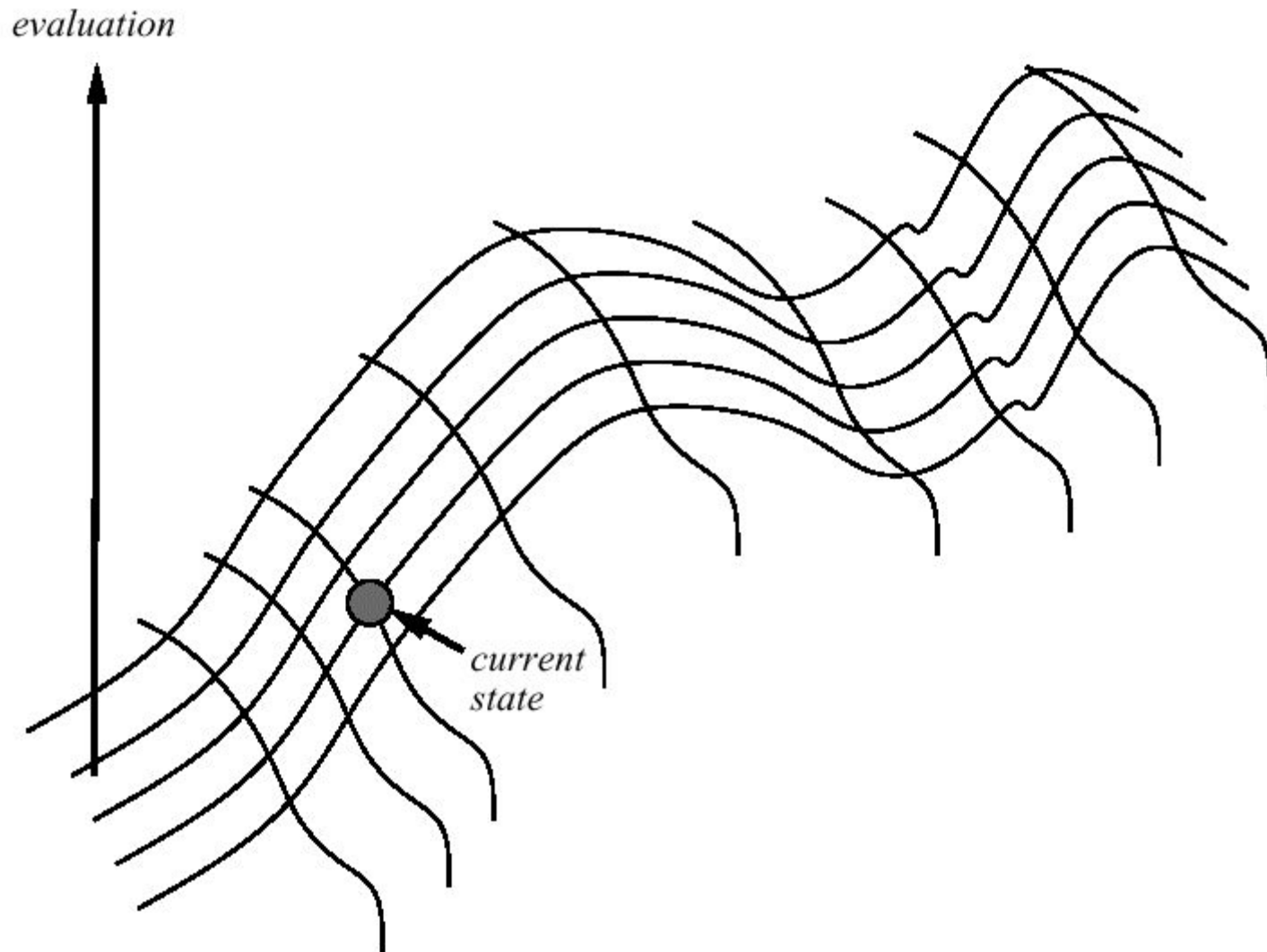
Busca Local

Subida da encosta

- Formalmente, o algoritmo da subida da encosta escolhe, como próximo passo, o nó que parece estar mais próximo do objetivo.
- O nome do algoritmo deriva da analogia com o andarilho que está perdido no escuro em uma montanha e deseja chegar até o topo da montanha. Mesmo no escuro, o andarilho sabe que cada passo para cima é um passo na direção certa.
- No caso da seleção de uma rota para viagem, podemos incorporar a seguinte heurística: Escolha a cidade que seja a mais distante possível da posição corrente

Busca Local

Subida da encosta



Busca Local

Subida da encosta

function HILL-CLIMBING(*problem*) **returns** a solution state

inputs: *problem*, a problem

static: *current*, a node

next, a node

current \leftarrow MAKE-NODE(INITIAL-STATE[*problem*])

loop do

next \leftarrow a highest-valued successor of *current*

if VALUE[*next*] < VALUE[*current*] **then return** *current*

current \leftarrow *next*

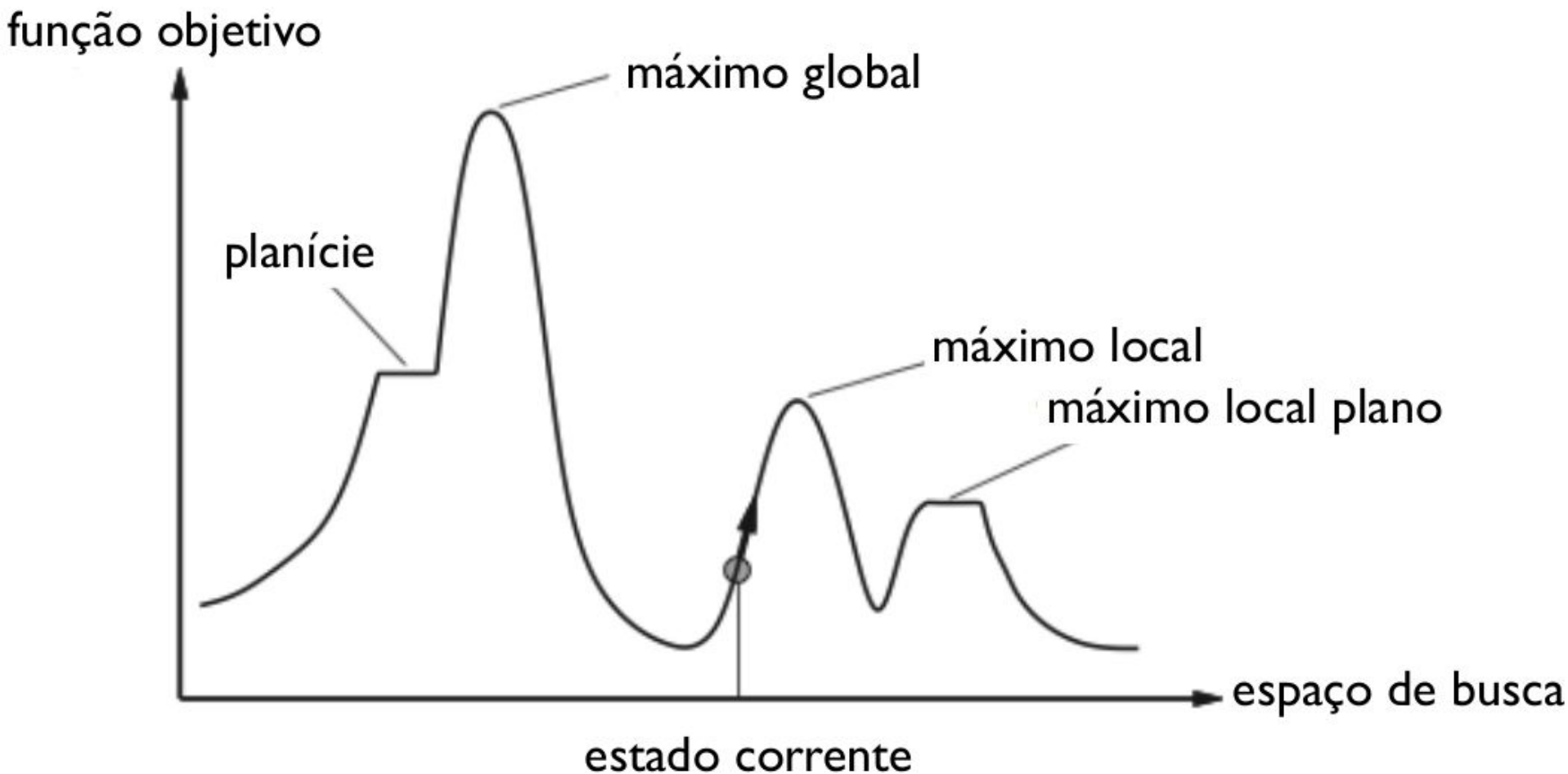
end

Subida da encosta (problemas)

- Falsas montanhas.
- Planaltos — no qual todos os passos possíveis, a partir do nó corrente, parecem igualmente bons (ou ruins). Nesse caso, a subida da encosta não é melhor do que a busca em profundidade.
- Crista — ocorre quando existe uma área no espaço de busca mais alta do que as áreas circunjacentes, mas que não pode ser atravessada por movimentos singulares numa direção qualquer.

Busca Local

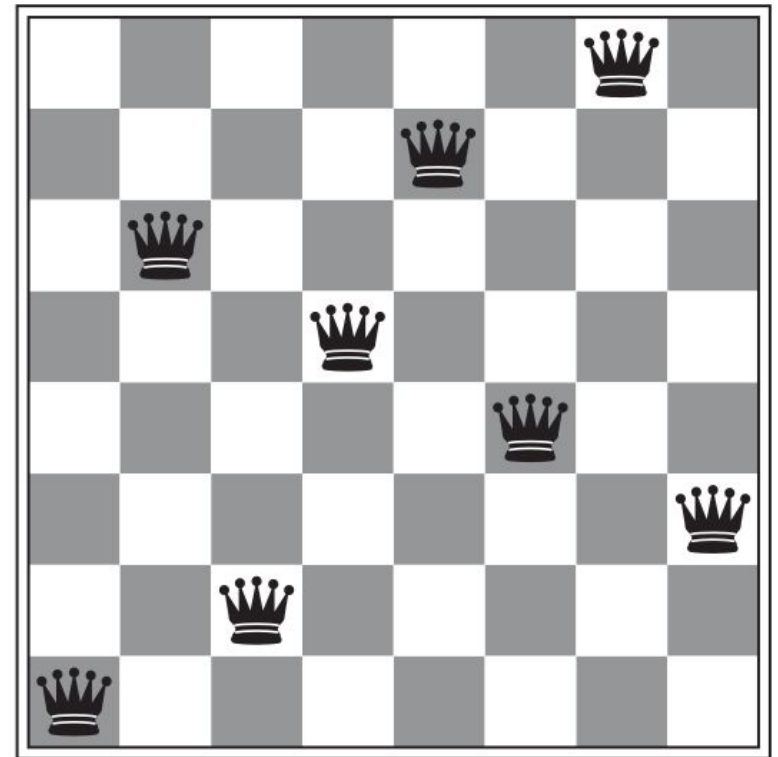
Subida da encosta (problemas)



Busca Local

Subida da encosta (problemas)

18	12	14	13	13	12	14	14
14	16	13	15	12	14	12	16
14	12	18	13	15	12	14	14
15	14	14	♔	13	16	13	16
♔	14	17	15	♔	14	16	16
17	♔	16	18	15	♔	15	♔
18	14	♔	15	15	14	♔	16
14	14	13	17	12	14	12	18



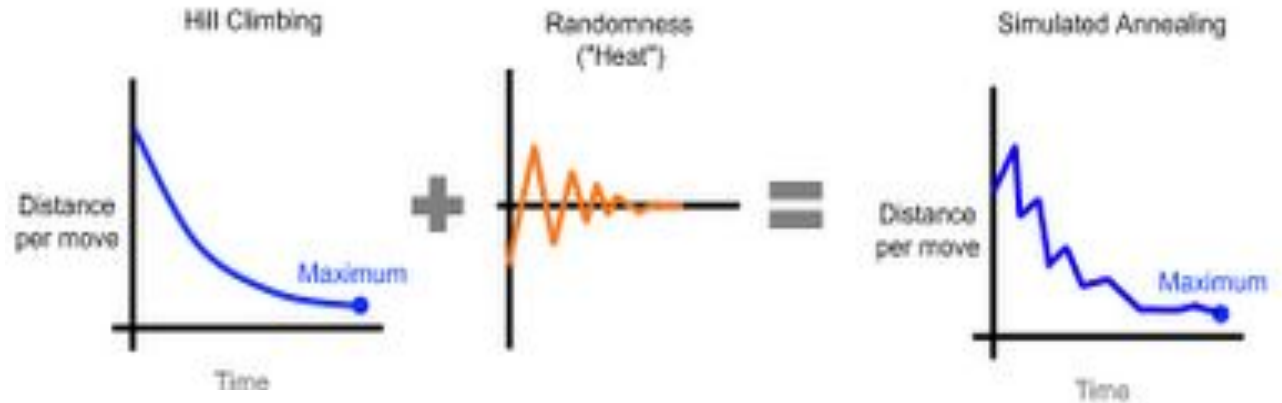
Busca Local

A thick, horizontal yellow brushstroke with a textured, painterly appearance, spanning the width of the slide below the title.

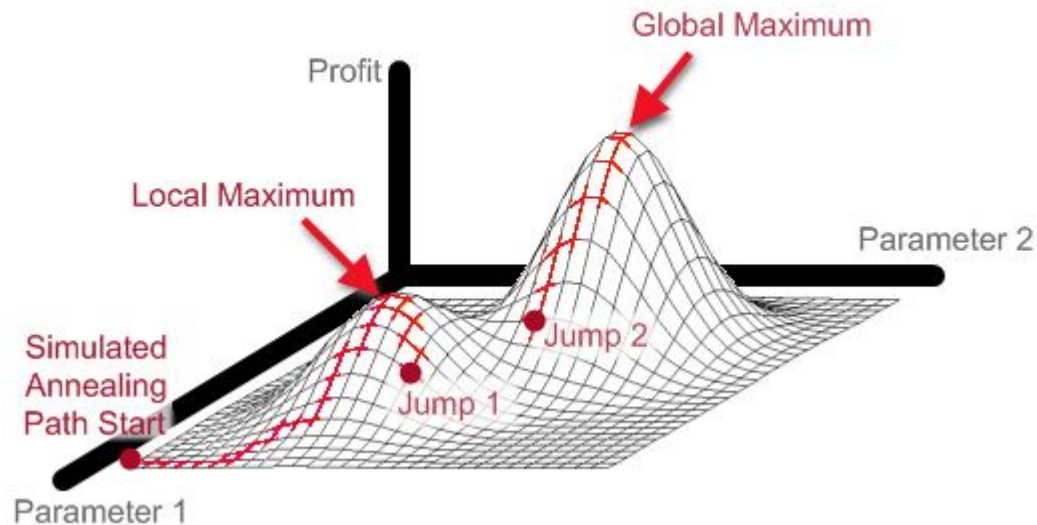
Têmpera simulada

Busca Local

Têmpera simulada



Simulated Annealing can escape local minima with chaotic jumps



Busca Local

Busca Heurística: Têmpera simulada

- Ao invés de ficar preso em um máximo local podemos permitir que a pesquisa faça alguns movimentos para baixo para escapar dos máximos locais.
- Esta é a idéia da têmpera simulada. Ao invés de sempre pegar o melhor caminho o algoritmo escolhe um caminho aleatório.
- Se o caminho melhorar a situação ele é escolhido. Caso contrário, existe probabilidade do caminho ser escolhido.
- A probabilidade de escolher um caminho ruim tende a decrescer com o tempo.

Busca Local

Busca Heurística: Têmpera simulada

- O nome do método vem da técnica de se esfriar um material gradualmente de modo a ter um melhor arranjo das moléculas.
- Esta técnica tem sido usada com sucesso em vários problemas, como projeto de circuitos VLSI e otimização de linhas de produção.

Busca Local

Busca Heurística: Têmpera simulada

function SIMULATED-ANNEALING(*problem*, *schedule*) **returns** a solution state

inputs: *problem*, a problem

schedule, a mapping from time to “temperature”

static: *current*, a node

next, a node

T, a “temperature” controlling the probability of downward steps

current \leftarrow MAKE-NODE(INITIAL-STATE[*problem*])

for *t* \leftarrow 1 **to** ∞ **do**

T \leftarrow *schedule*[*t*]

if *T*=0 **then return** *current*

next \leftarrow a randomly selected successor of *current*

$\Delta E \leftarrow$ VALUE[*next*] – VALUE[*current*]

if $\Delta E > 0$ **then** *current* \leftarrow *next*

else *current* \leftarrow *next* only with probability $e^{\Delta E/T}$



FIM