

Inteligência Artificial

Além da Busca Clássica Parte 1

Hill Climbing

Prof. Jefferson Morais

Introdução

- Até agora, analisamos algoritmos cujas soluções eram uma **sequência de ações**
 - Mantendo um ou mais caminhos na memória
 - Registrando alternativas exploradas e não exploradas
- Entretanto, em muitos problemas, o **caminho** até o objetivo é **irrelevante**, importando apenas a solução encontrada
 - Ex.: problema da oito rainhas, otimização de redes de telecomunicações, roteamento de veículos, etc
- Analisaremos agora algoritmos que se importam apenas com a **solução** e não com o custo do caminho para alcançá-lo

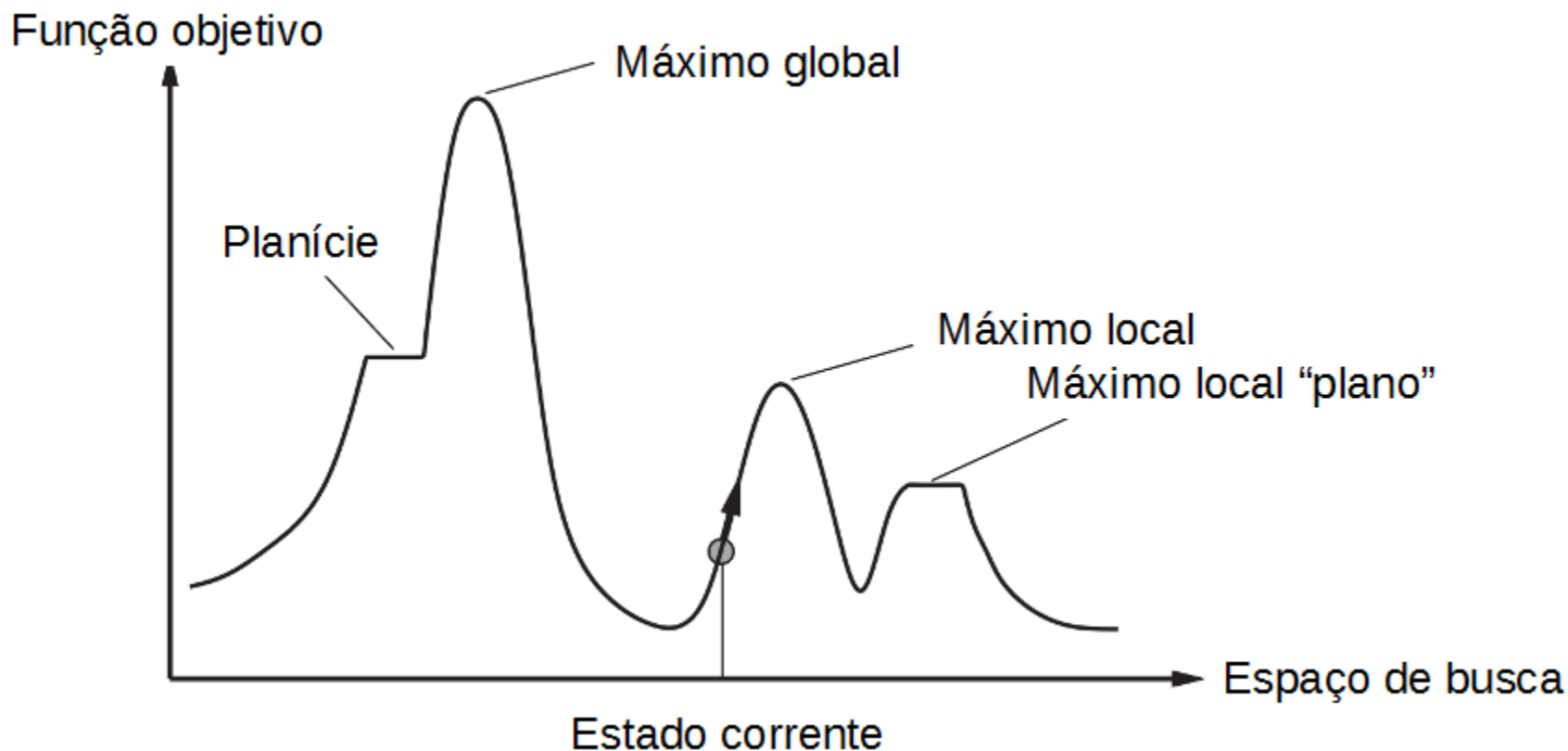
Algoritmos de Busca Local

- São aqueles que operam usando um **único estado atual** (em vez de vários caminhos)
- Em geral, movem-se apenas para os **vizinhos** desse estado
- Normalmente, os caminhos seguidos pela busca não são guardados (sem memória)
- Duas vantagens, embora não sejam sistemáticos
 - Usam pouca memória: normalmente um valor constante
 - Frequentemente encontram soluções razoáveis em grandes ou infinitos (contínuos) espaços de busca para os quais os algoritmos sistemáticos são inadequados

Algoritmos de Busca Local

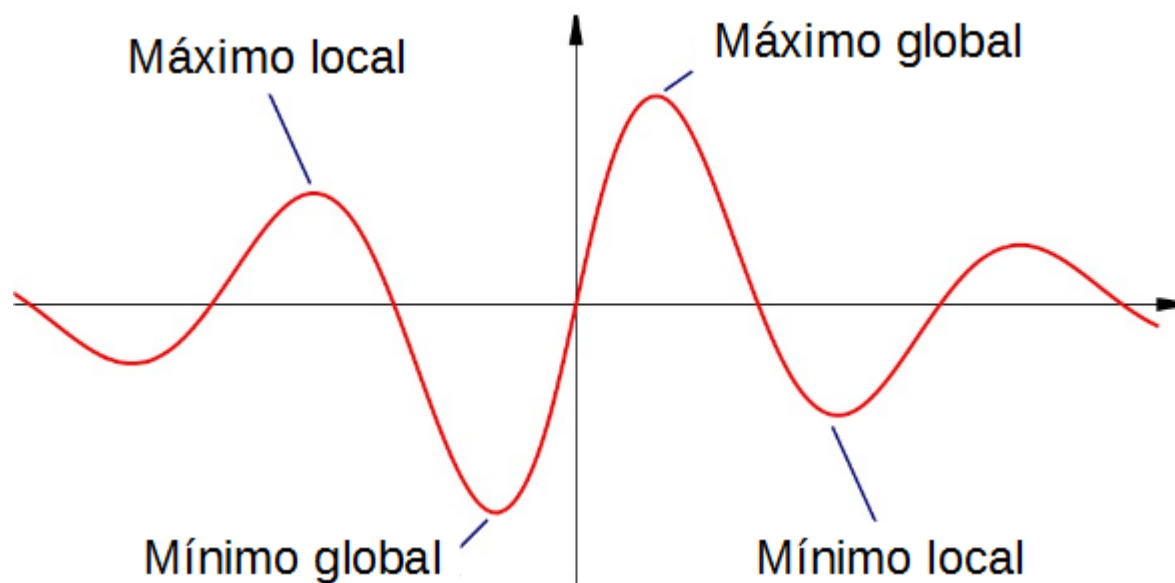
- **Topologias de espaço de estados**

- **Espaço de busca (estado)**: contém todos os estados possíveis
- **Função objetivo**: modela o problema e o que se quer otimizar

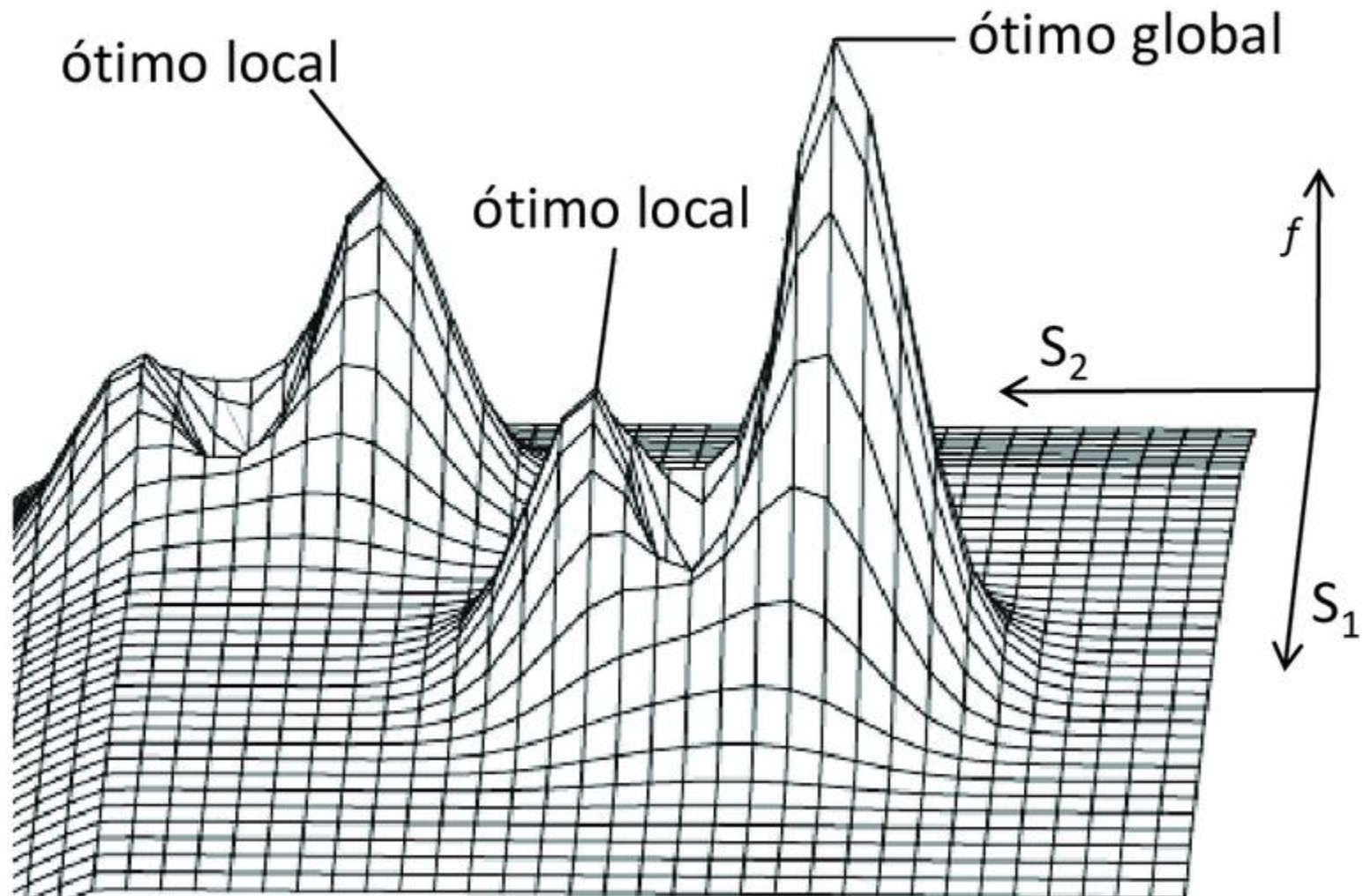


Algoritmos de Busca Local

- **Ótimo local**: melhor solução entre os estados vizinhos
 - Problemas de minimização, diz-se **mínimo local**
 - Problemas de maximização, diz-se **máximo local**
- **Ótimo global**: melhor solução entre todos os estados possíveis
 - Problemas de minimização, diz-se **mínimo global**
 - Problemas de maximização, diz-se **máximo global**

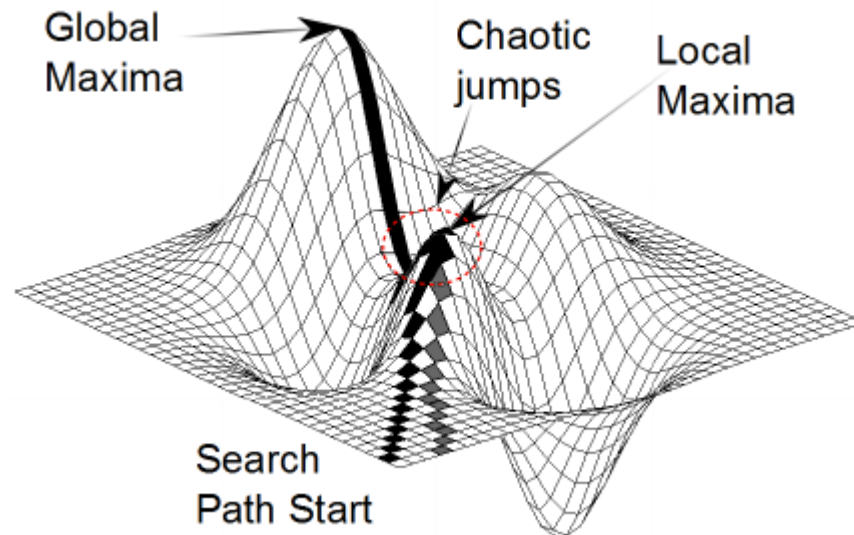


Algoritmos de Busca Local



Hill Climbing (subida da encosta)

- **Versão encosta mais íngreme (steepest ascent)**
 - É um laço repetitivo que se move de forma contínua no sentido do **valor crescente** (encosta acima)
 - O algoritmo termina quando alcança um “pico” (problema de maximização) em que nenhum vizinho tem valor mais alto



Hill Climbing

Versão: steepest ascent

function HILL-CLIMBING(*problem*) **returns** a state that is a local maximum

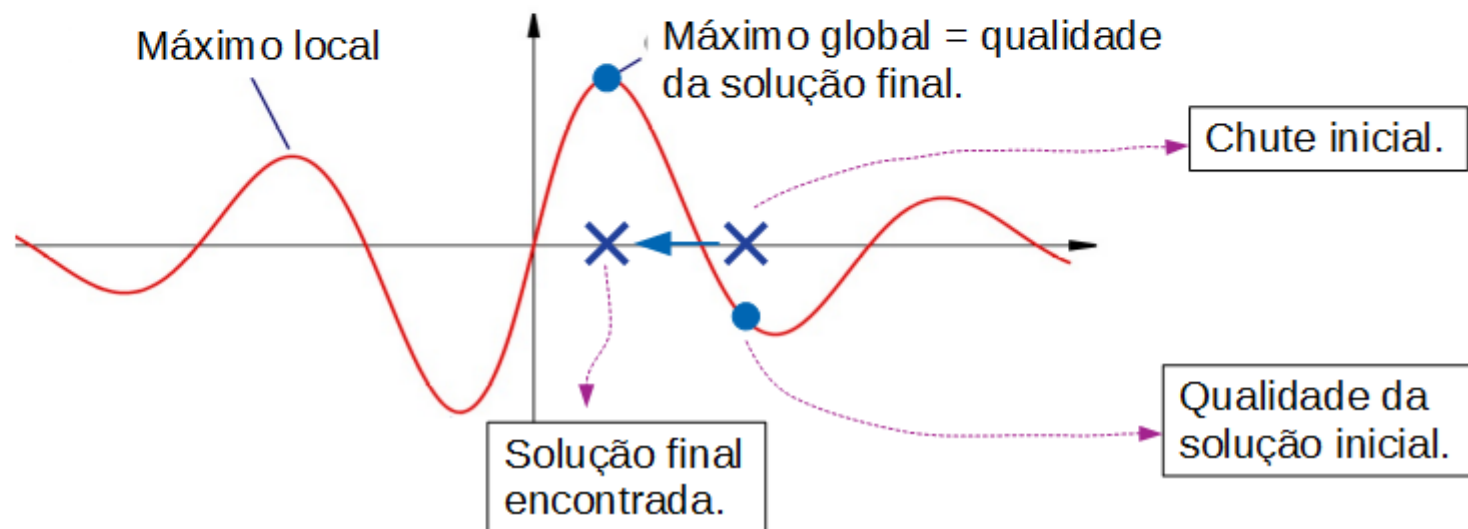
current \leftarrow MAKE-NODE(*problem*.INITIAL-STATE)

loop do

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

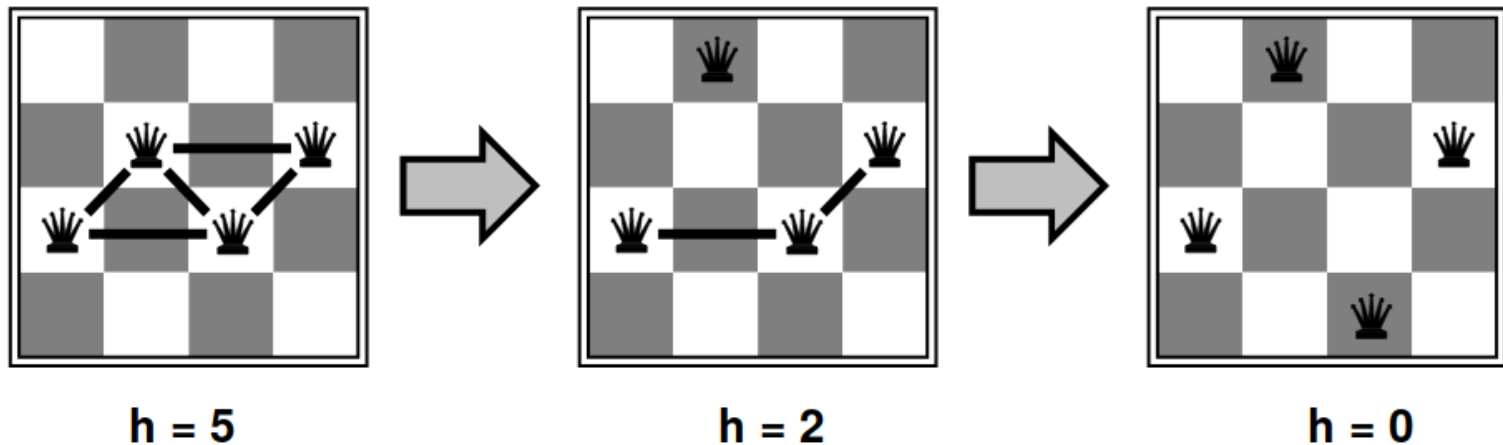
if *neighbor*.VALUE \leq *current*.VALUE **then return** *current*.STATE

current \leftarrow *neighbor*



Hill Climbing - Problema das n Rainhas

- O problema consiste em
 - Posicionar n rainhas em um tabuleiro $n \times n$, sendo que nenhum par de rainhas deve ficar na mesma linha, coluna ou diagonal
 - Sucessores de um estado são todos os estados possíveis a partir do movimento de uma única rainha
- Deve-se mover as rainhas para **reduzir** o número de conflitos. Uma possível **função de custo h** : número de pares de rainhas em conflito
- O mínimo global desta função é zero

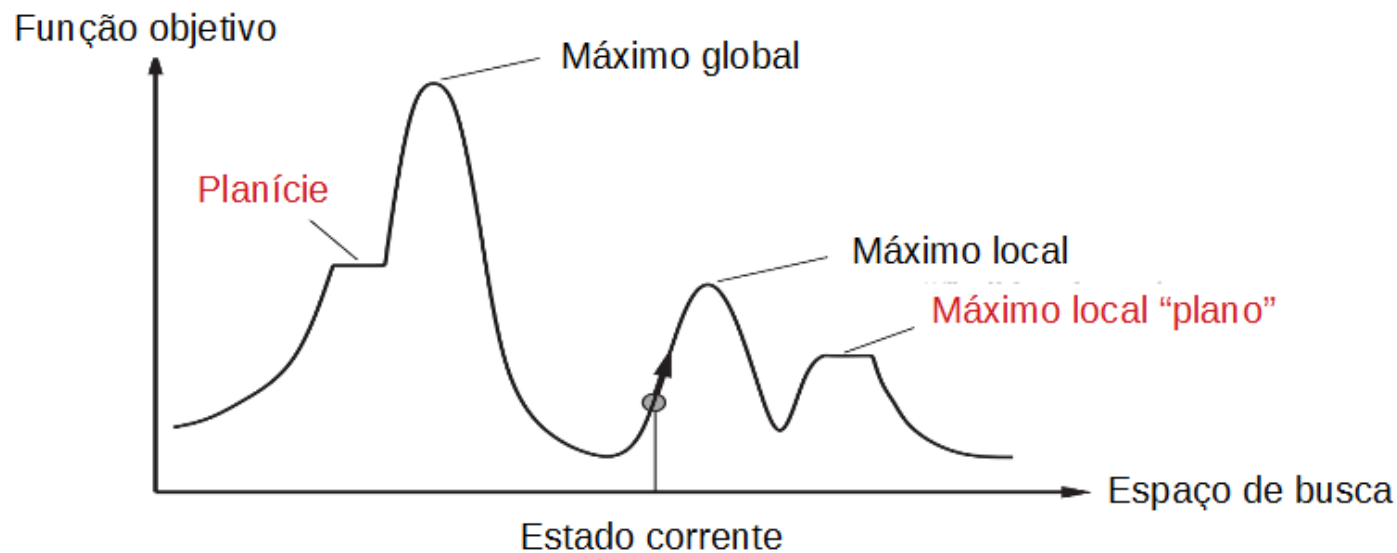


Hill Climbing

- O Hill Climbing também é conhecido como **busca gulosa local**
 - Captura um bom estado vizinho sem decidir com antecedência para onde irá em seguida
- Quando **n** é pequeno, o algoritmo progride ao ótimo global rapidamente
- Em caso de platôs (áreas planas na topologia de espaço de estados)
 - Normalmente, o algoritmo faz uma **escolha aleatória entre o conjunto de melhores sucessores**

Hill Climbing

- O algoritmo Hill Climbing pode ficar **paralisado**
 - **Máximos locais**: pico mais alto entre seus vizinhos e mais baixo que o máximo global
 - **Platôs**: é uma área plana da topologia de espaço de estados
 - **Máximo local plano**: não existe saída encosta acima
 - **Planície**: é possível progredir

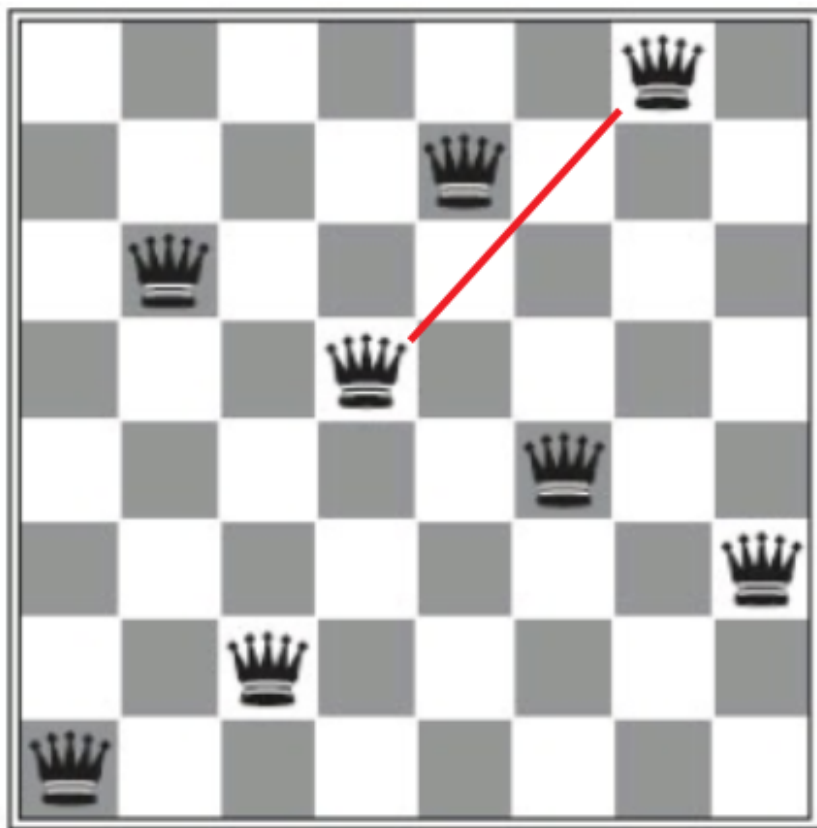


Hill Climbing

- A partir de um estado do problema de **oito rainhas** gerado aleatoriamente
 - 86% do tempo ficará paralisada
 - Resolve 14% de instâncias do problema
- Funciona com rapidez, em média
 - 4 passos quando tem sucesso
 - 3 passos quando fica paralisada
- Considerando o espaço de busca com $8^8 \approx 17$ milhões de estados, o resultado possível com Hill Climbing é satisfatório

Hill Climbing

- **Mínimo local:** todos os estados vizinhos ao atual são piores que ele
 - Ex.: atual $h = 1$, vizinhos $h > 1$



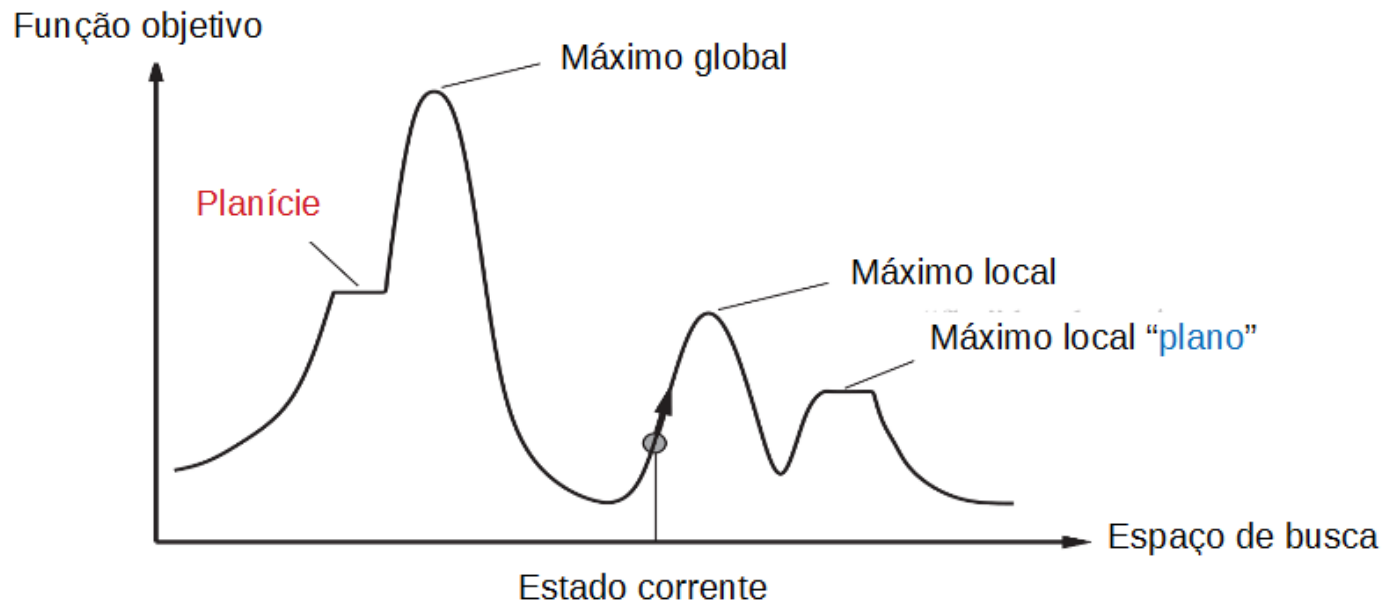
Formulação do problema com $8 \times 7 = 56$ estados sucessores.

Considerando que cada rainha deve ocupar uma coluna (heurística).

Mover a rainha da coluna 4 ou 7, na mesma coluna, piorará a função objetivo.

Hill Climbing

- **Platôs:** uma região do espaço de estados onde a função de avaliação retorna o mesmo resultado
 - Pode funcionar se o platô for uma **planície**
 - Se for uma um máximo local **plano**, ocorrerá uma repetição infinita



Hill Climbing

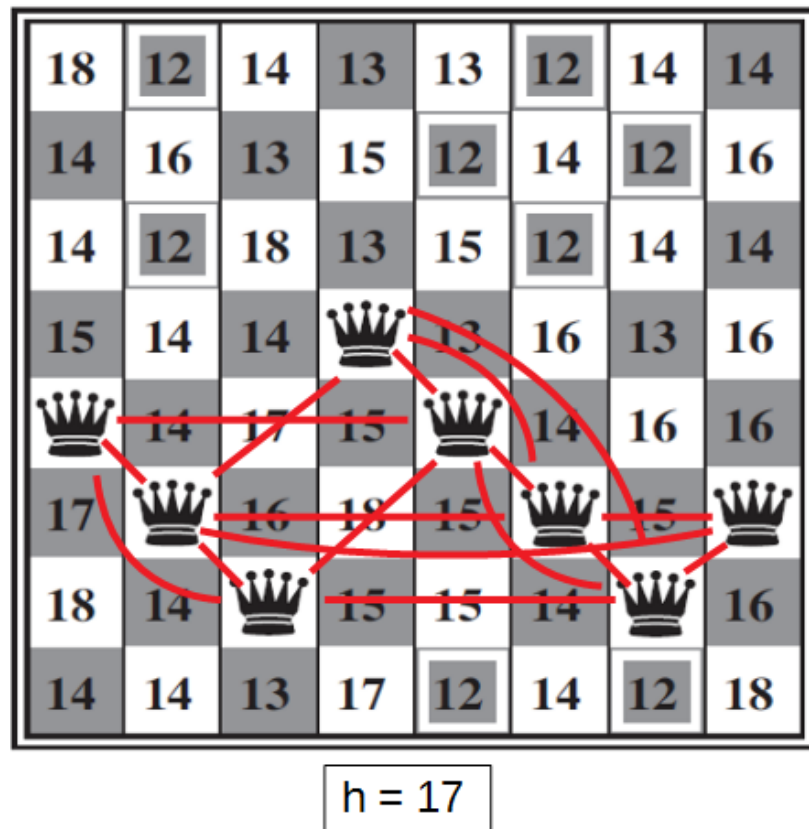
- Possível solução para os problemas das rainhas
 - Impor um limite para o número de movimentos laterais consecutivos permitidos
 - 100 movimentos laterais consecutivos para $n = 8$
 - Aumenta a porcentagem de instâncias resolvidas
 - De 14% para 94%
 - Custo em média
 - 22 passos para instâncias bem sucedidas
 - 64 passos para cada falha

Hill Climbing

- **Subida da encosta mais íngreme (Steepest ascent hill climbing):** todos os sucessores são avaliados e o melhor é selecionado como o próximo movimento
- **Subida da encosta estocástica (Stochastic hill climbing):** escolhe de forma aleatória os movimentos encosta acima (converge mais lentamente que a subida mais íngreme)
 - **Implementação: subida da encosta pela primeira escolha (First-choice hill climbing)** que gera sucessores ao acaso até encontrar um sucessor melhor que o estado corrente. É uma boa estratégia quando um estado tem muitos sucessos (milhares)
- **Subida da encosta com reinício aleatório (Random-restart hill climbing):** inicia em um estado aleatório e aplica o algoritmo. Se o algoritmo não encontrar uma solução na primeira execução, reinicie-o

Hill Climbing

- **Subida da encosta mais íngreme**
 - Avaliar todos os sucessores e escolhe o melhor (uma das opções onde $h = 12$)
- **Subida da encosta pela primeira escolha**
 - Escolhe aleatoriamente um movimento com que leve a $h < 17$
- **Subida da encosta com reinício aleatório**
 - Executa várias vezes o algoritmo com o estado inicial aleatório



Hill Climbing

- **Próxima Aula:**

Simulated Annealing