



# INF 1771 – Inteligência Artificial

## Aula 05 – Busca Local

Edirlei Soares de Lima  
<elima@inf.puc-rio.br>

# Métodos de Busca

- **Busca Cega ou Exaustiva:**
  - Não sabe qual o melhor nó da fronteira a ser expandido. Apenas distingue o estado objetivo dos não objetivos.
- **Busca Heurística:**
  - Estima qual o melhor nó da fronteira a ser expandido com base em funções heurísticas.
- **Busca Local:**
  - Operam em um único estado e movem-se para a vizinhança deste estado.

# Busca Local

- Em muitos problemas o **caminho para a solução é irrelevante**.
  - **Jogo das n-rainhas:** o que importa é a configuração final e não a ordem em que as rainhas foram acrescentadas.
  - **Outros exemplos:**
    - Projeto de Circuitos eletrônicos;
    - Layout de instalações industriais;
    - Escalonamento de salas de aula;
    - Otimização de redes;
- Se o caminho para a solução não importa, podemos utilizar um algoritmo de **busca local**.

# Busca Local

- Algoritmos de busca local operam sobre um **único estado corrente**, ao invés de vários caminhos.
- Em geral se movem apenas para os **vizinhos** desse estado.
- O caminho seguido pelo algoritmo não é guardado.

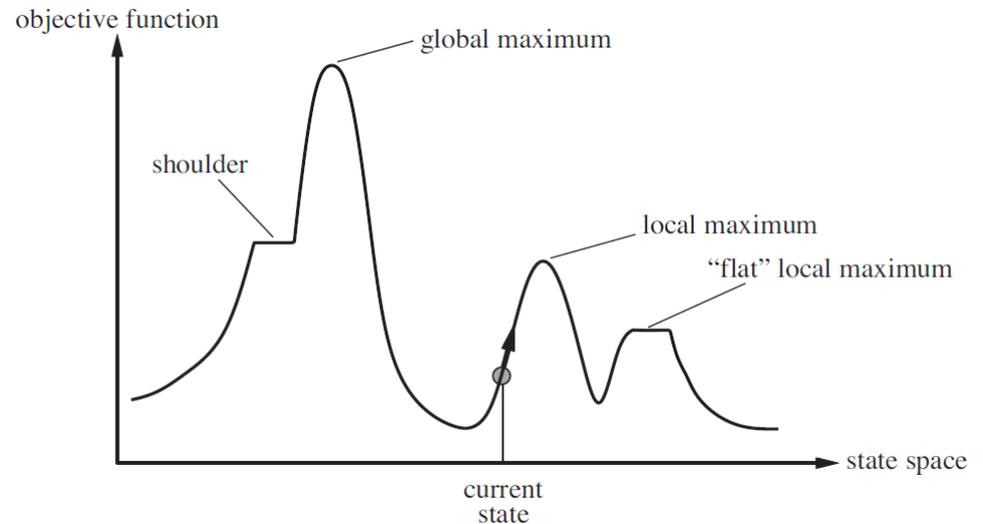
# Busca Local

- **Vantagens:**
  - Ocupam pouquíssima memória (normalmente constante).
  - Podem encontrar soluções razoáveis em grandes ou infinitos espaços de estados.
- **São uteis para resolver problemas de otimização.**
  - Buscar por estados que atendam a uma função objetivo.

# Busca Local

- **Panorama do Espaço de Estados:**

- **Local** = Estado;
- **Elevação** = Valor de custo da função heurística;
- Busca-se o máximo ou mínimo global;



# Busca Local

- **Principais Algoritmos:**
  - Hill Climbing
  - Simulated Annealing
  - Local Beam
  - Genetic Algorithms

# Pseudocódigo – Hill Climbing

**Função** Hill-Climbing(*Problema*) **retorna** um estado que é o máximo local

**Início**

EstadoAtual ← FazNó(*Problema*[EstadoInicial])

**loop do**

Vizinho ← SucessorDeMaiorValor(EstadoAtual)

**se** Vizinho[Valor] for menor ou igual EstadoAtual[Valor] **então**

**retorna** EstadoAtual

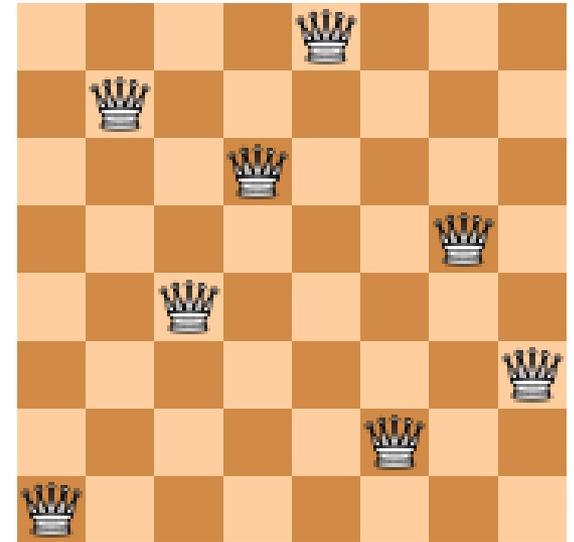
EstadoAtual ← Vizinho

**Fim**

- Consiste de de um loop que continuamente move-se para os estados que aumentam o valor em sua função de avaliação.
- Termina quando atinge um "pico" onde nenhum vizinho tem um valor maior.
- Não mantém uma árvores de busca.

# Hill Climbing – Exemplo

- **Problema:** 8 Rainhas (estados completos)
  - Em cada estado: 8 rainhas no tabuleiro, uma em cada coluna.
- **Ações:**
  - Mover uma rainha para outro quadrado na mesma coluna.
- **Função Heurística (h):**
  - Número de rainhas sendo atacadas.
- **Objetivo:**
  - $h = 0$  (nenhuma rainha sendo atacada)



# Hill Climbing – Exemplo

- $h = 17$
- Melhor movimento: 12

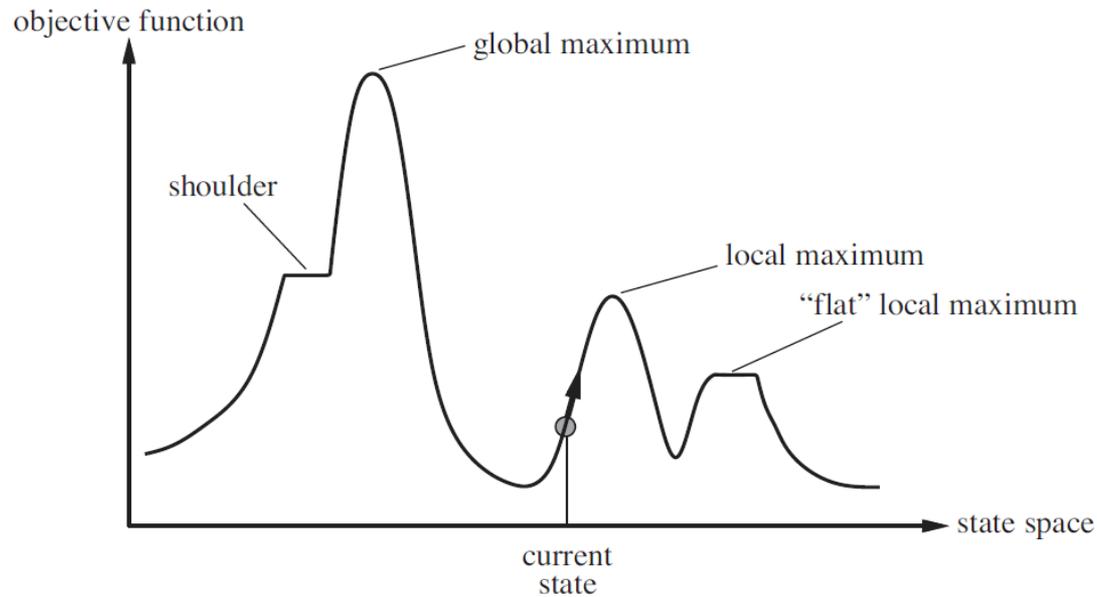
18	12	14	13	13	12	14	14
14	16	13	15	12	14	12	16
14	12	18	13	15	12	14	14
15	14	14	👑	13	16	13	16
👑	14	17	15	👑	14	16	16
17	👑	16	18	15	👑	15	👑
18	14	👑	15	15	14	👑	16
14	14	13	17	12	14	12	18

# Hill Climbing

- É um **algoritmo guloso** – escolhe sempre o primeiro melhor vizinho para progredir na busca.
- Essa abordagem pode ter **bons resultados em alguns problemas**. Sendo capaz de progredir rapidamente para a solução problema.
- Mas, sofre de três sérios **problemas**:
  - Máximos locais
  - Planícies
  - Encostas e Picos

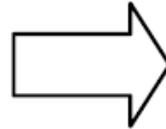
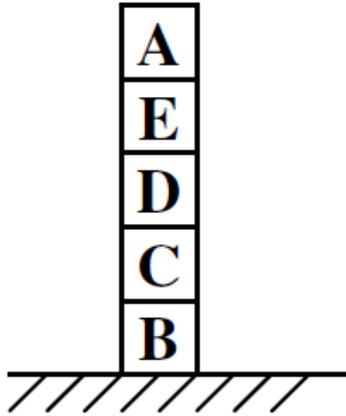
# Hill Climbing

- Máximos Locais
- Planícies

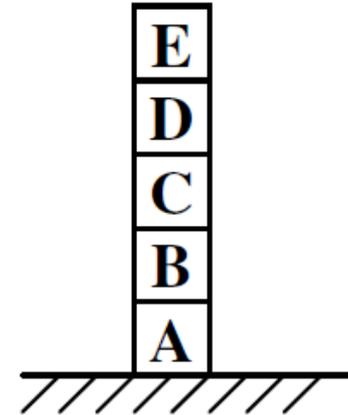


# Hill Climbing – Exemplo

**Estado Inicial**

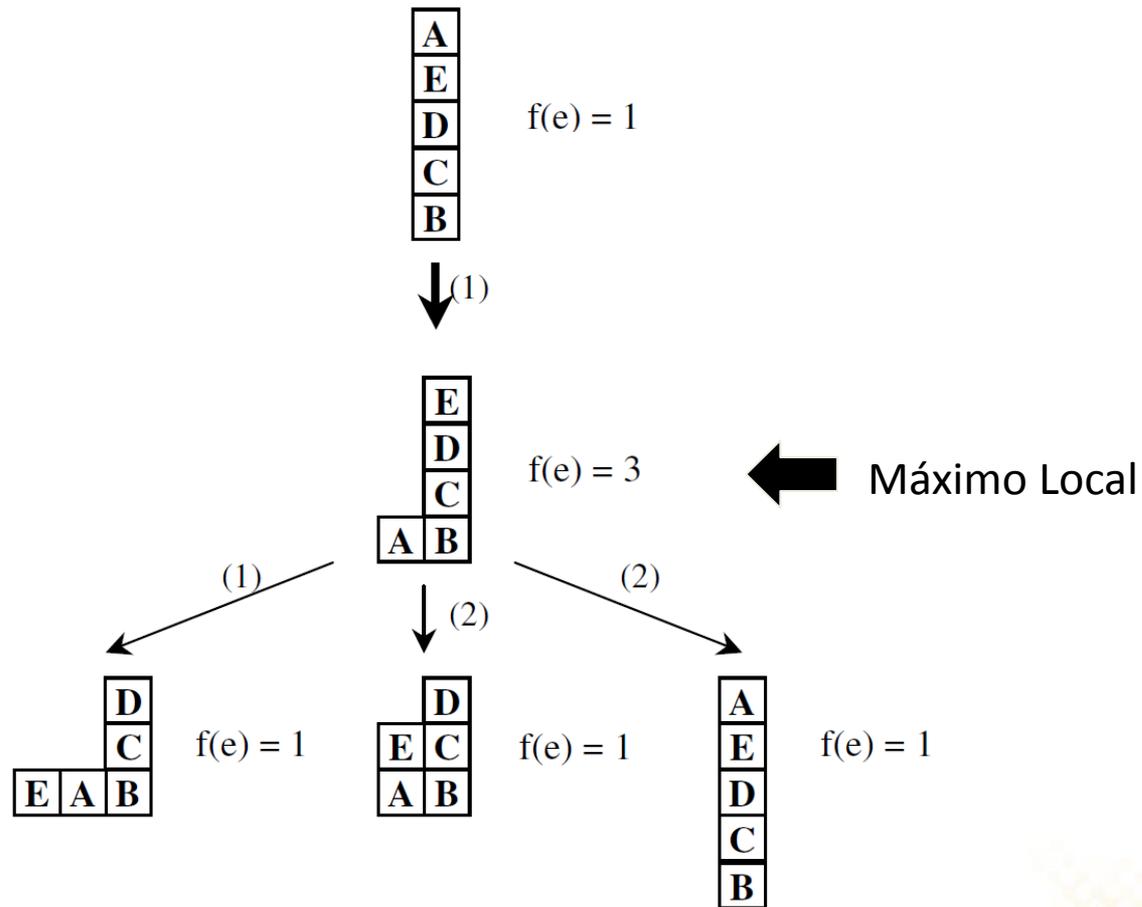


**Estado Meta**



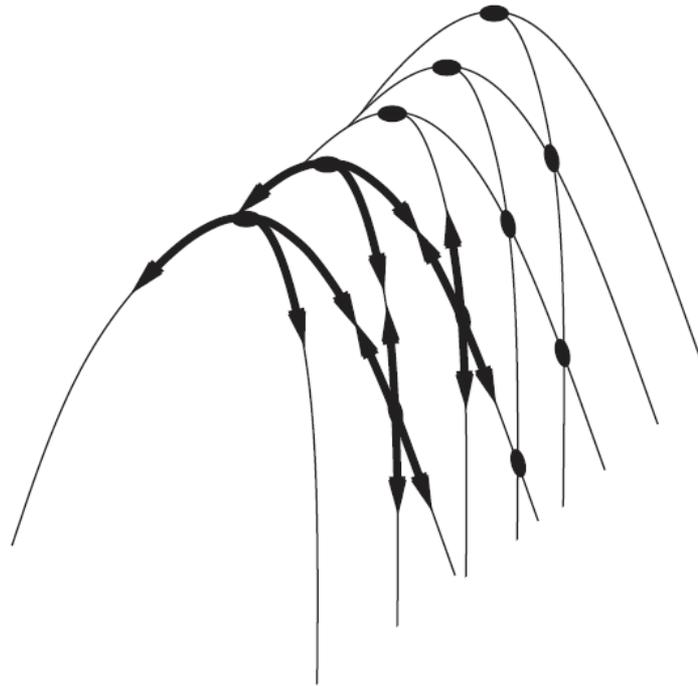
- **Ações Possíveis:**
  - Pegar um bloco e colocar ele sobre a mesa.
  - Pegar um bloco e colocar ele sobre outro bloco.
- **Heurística:**
  - +1 para cada bloco em cima do bloco onde ele deve estar.
  - -1 para cada bloco em cima do bloco errado.

# Hill Climbing – Exemplo



# Hill Climbing

- Encostas e Picos



# Hill Climbing – Exemplo

- **Problema: 8 Rainhas**
- **Inicializando aleatoriamente o estado inicial:**
  - O algoritmo fica em **preso em um máximo local** em 86% das vezes.
  - Resolve apenas 14% das instancias.
- Quando tem sucesso, resolve o problema em **aproximadamente 4 passos** – nada mal para um espaço de estados com 17 milhões de estados.

# Hill Climbing

- **Variações:**
  - Random-Restart Hill Climbing;
- **Não é ótimo e não é completo.**
- O desempenho do Hill Climbing depende muito do **formato do panorama** do espaço de estados.

# Busca Online

- Todos os métodos de busca vistos até o momento são **offline**, ou seja, o agente calcula todos antes passos antes de agir.
- Agentes de busca online devem **intercalar entre busca e execução das ações**.
- Apropriado para **ambientes dinâmicos, desconhecidos** e também para **espaços de busca muito grandes**.

# Busca Online - Exemplo

- **Informações:**

- **Ações(s)**

- Lista de ações possíveis no estado  $s$ .

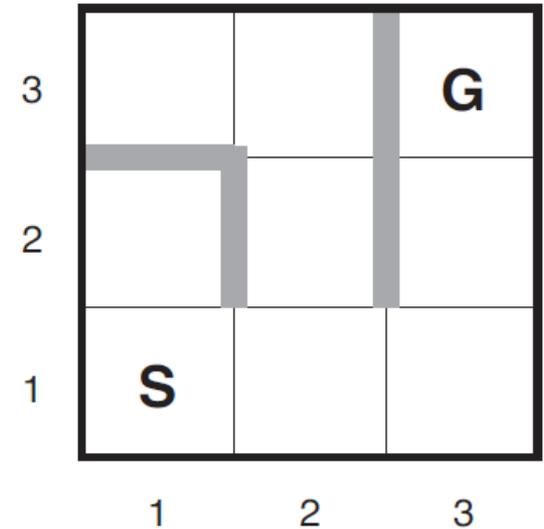
- **$C(s, a, s')$**

- Custo da execução da ação  $a$  no estado  $s$  resultando no estado  $s'$ .

- **TesteObjetivo(s)**

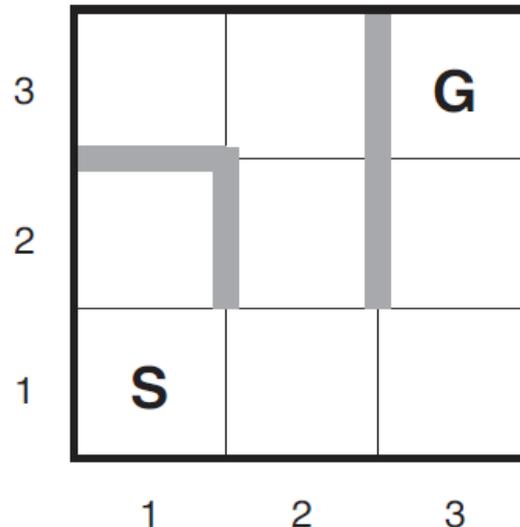
- Testa se  $s$  é o estado objetivo.

- O agente somente tem acesso aos estados sucessores após executar as ações.



# Busca em Profundidade Online

- Realiza uma busca em profundidade executando as ações fisicamente durante o processo até chegar no estado objetivo.
- Se todos os vizinhos de um estado já foram visitados é necessário retornar fisicamente para a posição anterior.

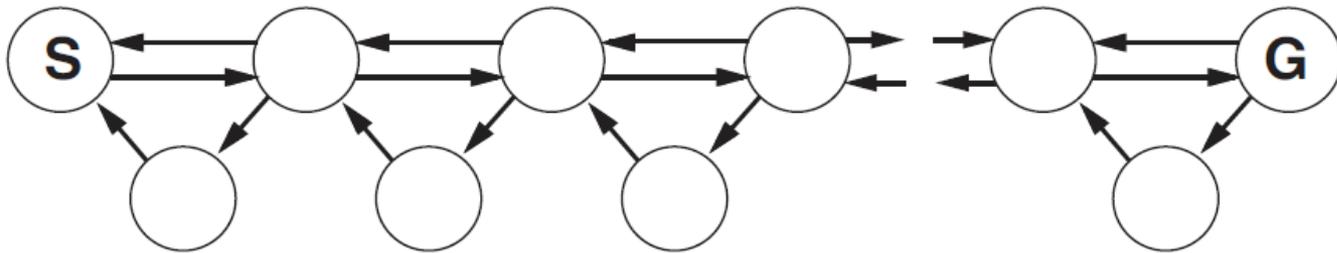


# Busca Local Online

- O **hill climbing** é naturalmente um algoritmo de busca online.
- **Problema:**
  - Na sua forma mais simples, pode deixar o agente parado em um **máximo local** sem ter para onde ir.
  - Não é possível **reiniciar o processo randomicamente** porque o agente não pode teletransportar.
- **Solução:**
  - Caminhada aleatória ao chegar em uma máximo local.

# Busca Local Online

- A caminhada aleatória garante que o agente eventualmente vai encontrar o objeto ou completar o processo de exploração.
- Esse processo pode ser lento.



# Qual Algoritmo Usar?



# Leitura Complementar

- Russell, S. and Norvig, P. **Artificial Intelligence: a Modern Approach**, 2nd Edition, Prentice-Hall, 2003.
- **Capítulo 4: Informed Search and Exploration**

