

Inteligência Artificial

BCC35G

Diego Bertolini

diegobertolini@utfpr.edu.br

<http://www.inf.ufpr.br/diegob/>

Aula 005

- **Aula Anterior:**
 - Busca com Informação;
- **Aula de Hoje:**
 - Busca Local;

Objetivo

O que vocês devem saber ao final da aula:

Discutiremos como as informações de estados podem evitar que os algoritmos se percam na escuridão .

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.

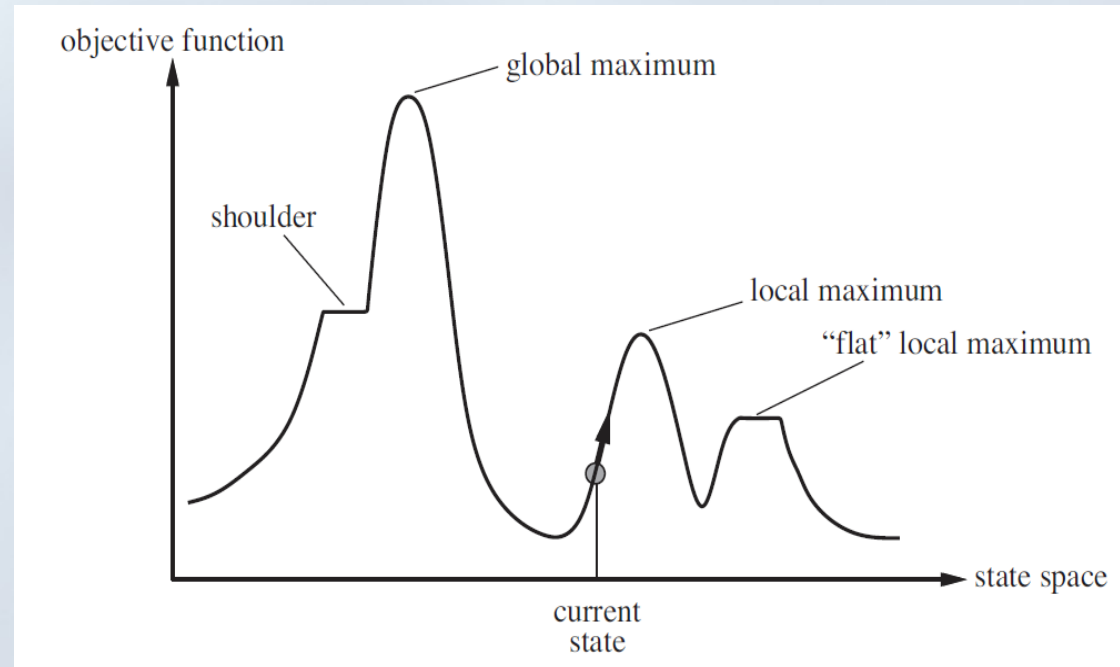
São uteis para resolver problemas de otimização.

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:

- 1) Hill Climbing (Existem dezenas de evoluções deste algoritmo) (Subida da Encosta)
- 2) Simulated Annealing (Têmpera Simulada)
- 3) Genetic Algorithms

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:

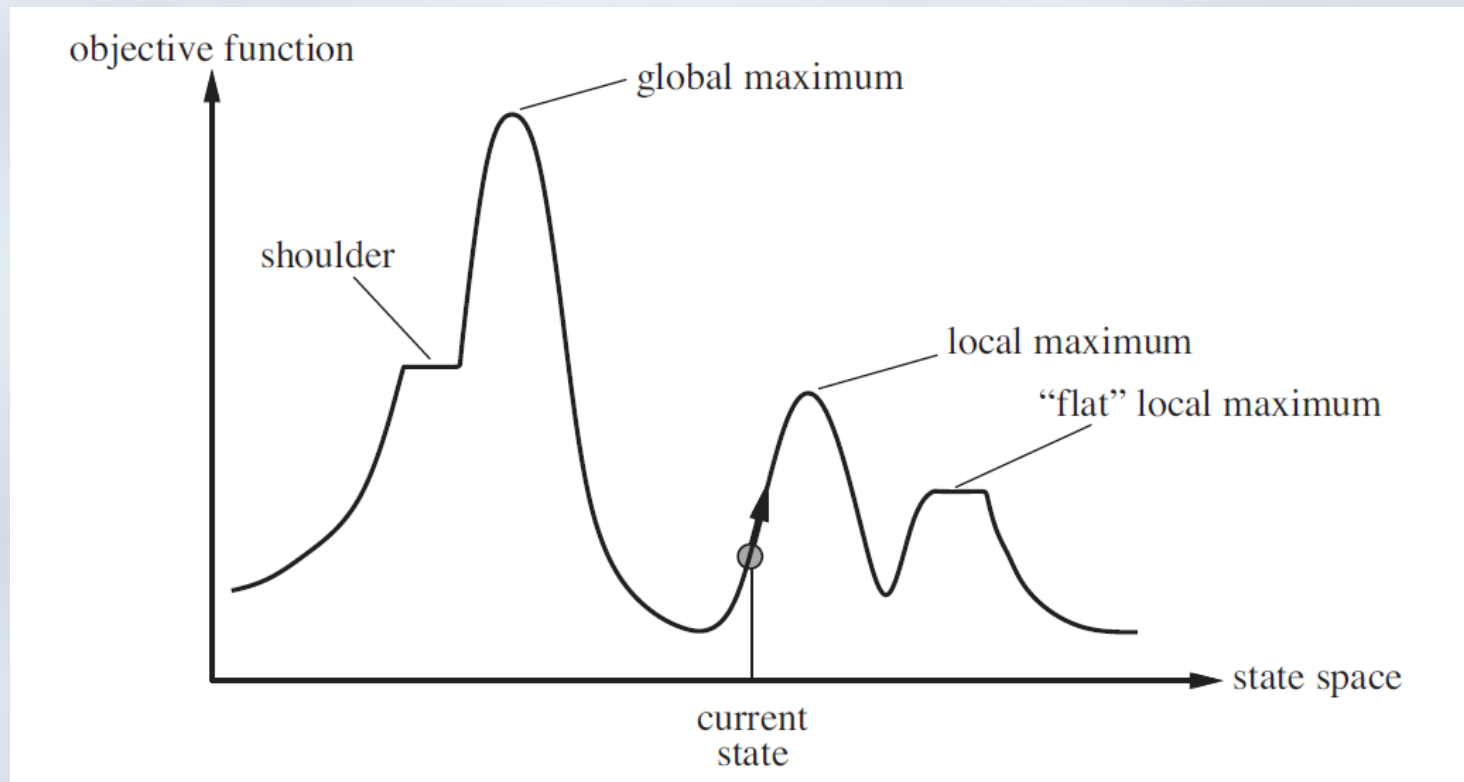
- Máximos locais

- Planícies

- Encostas e Picos

Hill Climbing

Máximos Locais
Planícies



Hill Climbing

- *Se move de forma contínua no sentido do valor crescente*
- *Termina quando alcança um pico, em que nenhum vizinho tem valor mais alto ;*
- *Não mantém árvore de busca, somente o estado e o valor da função objetivo ;*
- *Não examina antecipadamente valores de estados além de seus vizinhos imediatos (busca gulosa local) ;*
- *É como subir o Everest em meio a um nevoeiro e sofrendo de Amnésia.*

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

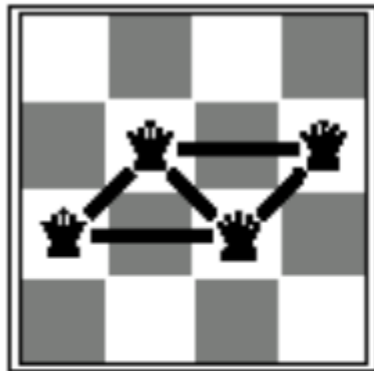
Fim

- 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.

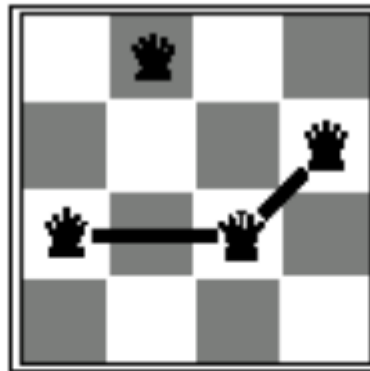
Hill Climbing – Exemplo

- Algoritmos de busca local utilizam uma formulação de estados completos ;
- Função sucessora gera todos os estados possíveis ;
- Cada estado tem n rainhas, 1 por coluna ;
- Gerados pela movimentação de uma única rainha para outro lugar na mesma coluna;
- A função heurística é o números de pares de rainhas que estão se atacando umas às outras ;
- O mínimo global dessa função é zero, que só ocorre em soluções perfeitas

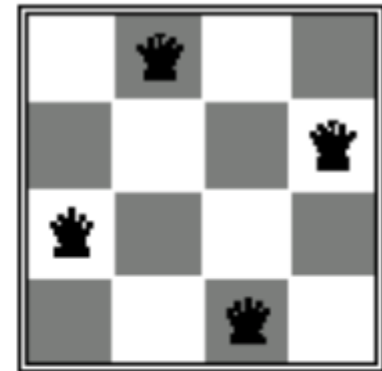
Hill Climbing – Exemplo



$h = 5$



$h = 2$



$h = 0$

Hill Climbing – Exemplo

Problema: 8 Rainhas (estados completos)

Em cada estado: 8 rainhas no tabuleiro, uma em cada coluna.

Ações:

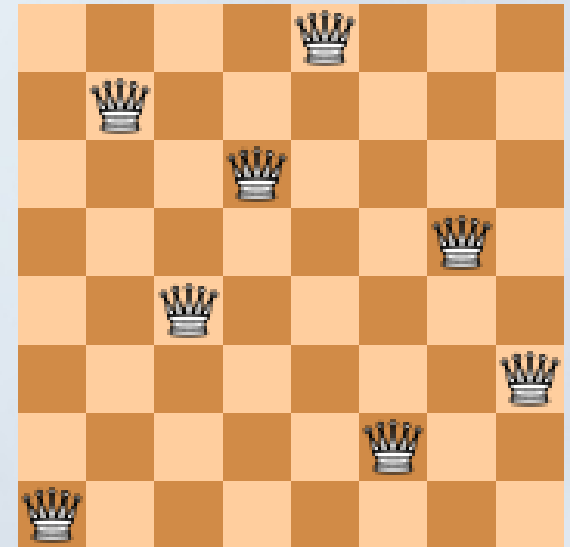
Mover uma rainha para outro quadrado na mesma coluna.

Função Heurística (h):

Número de rainhas sendo atacadas.

Objetivo:

$h = 0$ (nenhuma rainha sendo atacada)



Hill Climbing – Exemplo

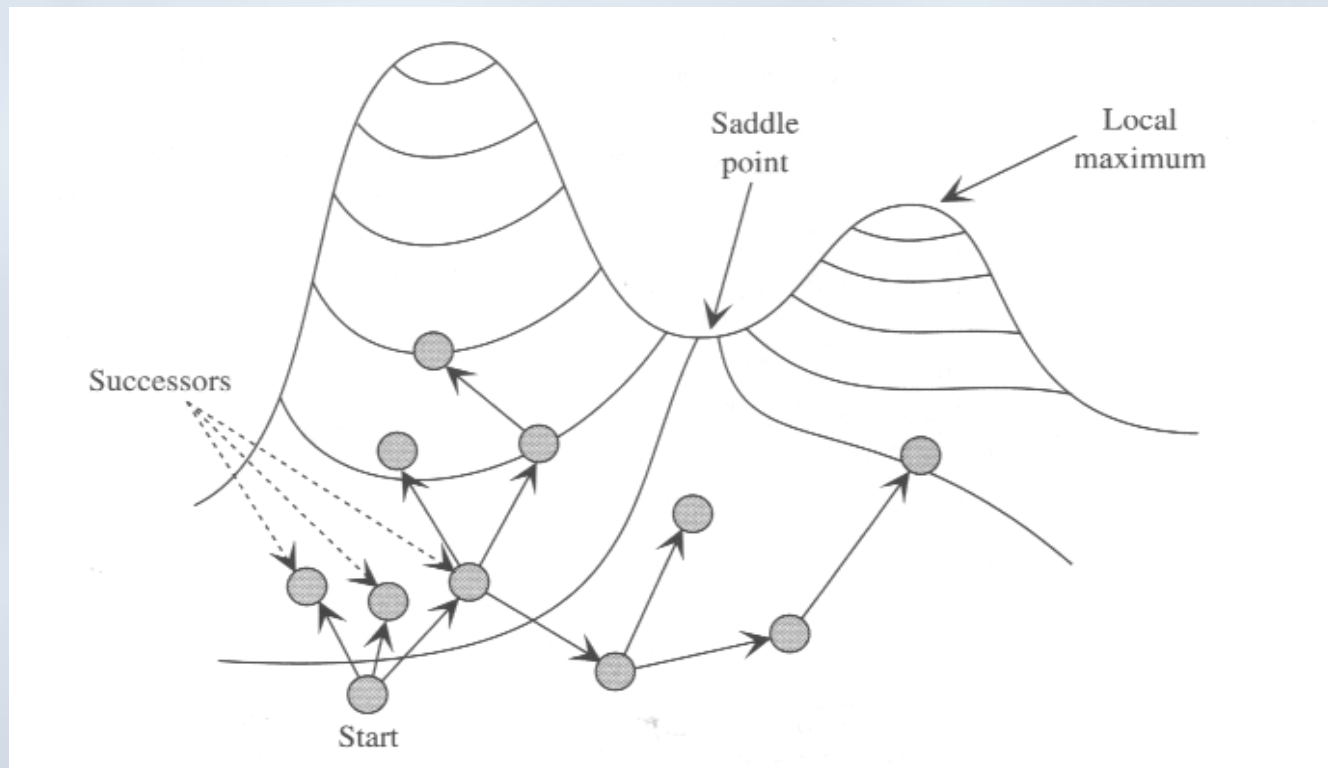
$h = 17$

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

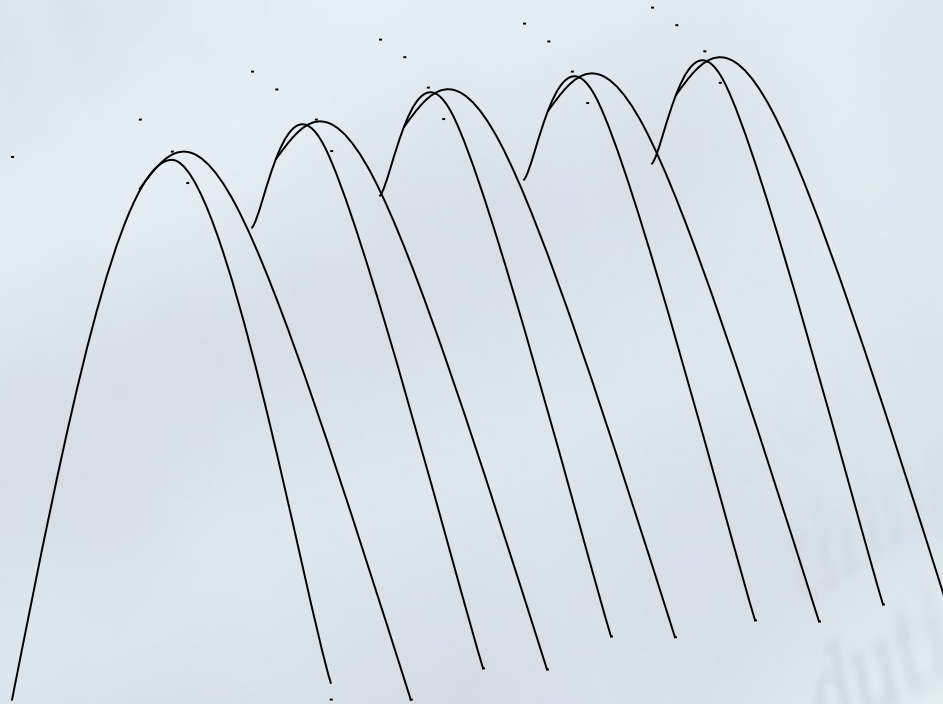
Hill Climbing – Exemplo

- Dependendo do estado inicial, pode ficar preso em **um máximo local**.



Hill Climbing

Podem existir **picos** que fazem com que a função de qualidade oscile entre vários máximos locais.



Hill Climbing – Exemplo

Problema: 8 Rainhas

Inicializando aleatoriamente o estado inicial:

O algoritmo fica **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

Subida da encosta estocástica:

- Escolhe **ao acaso** um estado vizinho com valor melhor do que o estado corrente.
- Converge mais lentamente, mas encontra soluções melhores.

Subida da encosta pela primeira escolha:

- Gera sucessores **ao acaso**, até gerar um sucessor melhor do que o estado corrente.
- É uma boa estratégia quando um estado tem muitos sucessores.

Subida da encosta com reinício **aleatório**:

- Conduz uma série de buscas a partir de estados iniciais gerados aleatoriamente.

Hill Climbing

O sucesso deste tipo de busca depende muito da topologia do espaço de estados.

- Muitos problemas reais tem uma topologia mais parecida com uma família de ouriços em um piso plano, com ouriços em miniatura vivendo na ponta de cada espinho de um ouriço, *ad infinitum*.

Problemas NP-completos têm um número exponencial de máximos locais em que ficam paralisados.

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.

Simulated Annealing (Têmpera simulada)

Têmpera: processo usado para temperar ou endurecer metais e vidro aquecendo-os a alta temperatura e depois resfriando gradualmente.

- Existem várias técnicas em computação que são inspiradas na natureza – computação natural.

Ideia: Fugir do máximo local permitindo alguns movimentos “ruins” para fora do máximo, mas gradualmente decrescendo seu tamanho e frequência.

A temperatura diminui em função do tempo diminuindo a probabilidade de se escolher um estado pior.

Amplamente utilizado para layout de VLSI, planejamento de linhas aéreas, etc.

Propriedades do Simulated Annealing

- Se estivermos interessados em encontrar o mínimo de uma função, podemos pensar no processo de colocar uma bola de pingue-pongue na fenda mais profunda da superfície (espaço de estados).
- A ideia é agitar a superfície com força suficiente para tirar a bola dos mínimos locais, mas não o bastante para tirá-la do mínimo global.
- No início da busca, agitamos com força (temperatura alta).
- Durante a busca, a força decresce conforme a temperatura diminui.

Bibliografia e Materiais.

Estes slides foram adaptados do Livro:

Russell, S. and Norvig, P. Artificial Intelligence: a Modern Approach, 2nd Edition, Prentice-Hall, 2003. Capítulo 3: Informed Search and Exploration

Solving Problems by Searching;

Adaptado das Aulas do Professor: Ederley – PUC

