

Inteligência Artificial

Busca Local

Prof. Dr^a. Andreza Sartori
asartori@furb.br

Documentos Consultados/Recomendados

- KLEIN, Dan; ABBEEL, Pieter. **Intro to AI**. UC Berkeley. Disponível em: <http://ai.berkeley.edu>.
- LIMA, Edirlei Soares. **Inteligência Artificial**. PUC-Rio, 2015.
- NOGUEIRA, Bruno Magalhães. **Algoritmos de busca - Parte 4: Busca local e algoritmo Hill Climbing (subida de encosta)**. LIA (Artificial Intelligence Laboratory) - UFMS. Acesso em: <https://www.youtube.com/watch?v=OD012QkZpCA>
- RUSSELL, Stuart J. (Stuart Jonathan); NORVIG, Peter. **Inteligência artificial**. Rio de Janeiro: Campus, 2013. 1021 p, il.
- ZADROZNY, Bianca. **Inteligência Artificial**. Instituto de Computação. Universidade Federal Fluminense (UFF) <http://www.ic.uff.br/~bianca/ia>

Conteúdo Programático

Unidade 1: Fundamentos de Inteligência Artificial

Unidade 2: Busca

Unidade 3: Sistemas Baseados em Conhecimento

Unidade 4: Redes Neurais Artificiais

Unidade 5: Aplicações de Inteligência Artificial



Conteúdo Programático

Unidade 1: Fundamentos de Inteligência Artificial

Unidade 2: Busca

Unidade 3: Sistemas Baseados em Conhecimento

Unidade 4: Redes Neurais Artificiais

Unidade 5: Aplicações de Inteligência Artificial



Conteúdo Programático

Unidade 1: Fundamentos de Inteligência Artificial

Unidade 2: Busca

2.1. Resolução de Problemas por meio de busca

2.2. Busca Cega ou Exaustiva

2.3. Busca Heurística

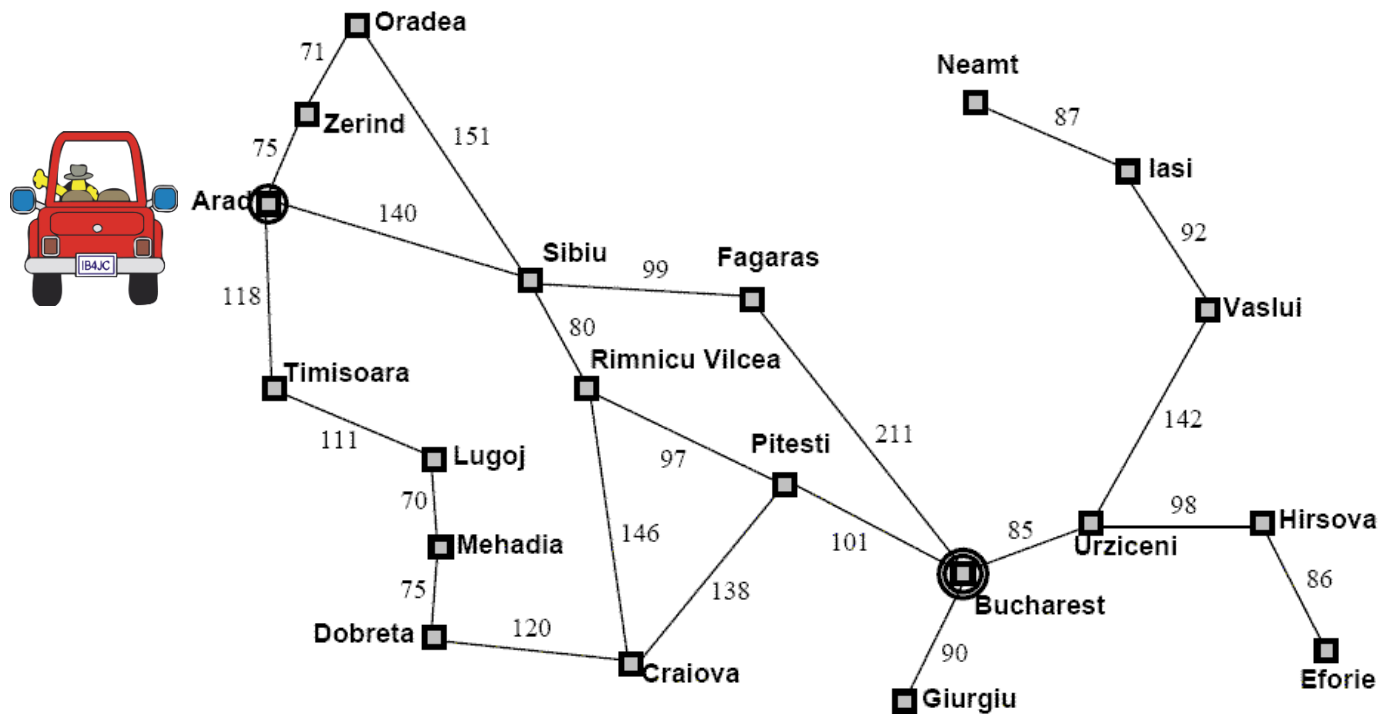
2.4. Busca Competitiva

2.5. Busca Local

2.5.1 Algoritmos Genéticos (AG)



Relembrando: Problema de Busca



Relembrando: Definição de um Problema

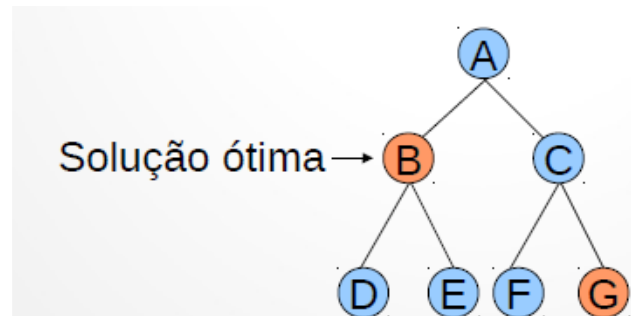
- **Estado Inicial:** Estado inicial do agente.
 - Ex: Em(Arad)
- **Estado Objetivo (Estado Final):** Estado buscado pelo agente.
 - Ex: Em(Bucharest)
- **Ações Possíveis (Função Sucessor):** Conjunto de ações que o agente pode executar.
 - Ex: Ir(Cidade, PróximaCidade)
- **Espaço de Estados:** Conjunto de estados que podem ser atingidos a partir do estado inicial.
 - Ex: Mapa da Romênia.
- **Custo de Caminho:** Custo numérico de cada caminho.
 - Ex: Distância em KM entre as cidades.

Relembrando: Solução para um Problema

- A **solução** para um problema é um caminho desde o estado inicial até o estado objetivo (estado final).
- A qualidade da solução é medida pela função de custo de caminho, isto é, a **solução que tiver menor custo** de caminho entre todas as soluções.

Medida de Desempenho do Algoritmo de Busca

- Uma estratégia de busca é definida pela escolha da **ordem da expansão de nós**
- Estratégias são avaliadas de acordo com os seguintes critérios:
 - **Completeza:** o algoritmo sempre encontra a solução se ela existe?
 - **Otimização (Custo de Caminho):** a estratégia encontra a solução ótima? - Qualidade da solução
 - Para passos com igual custo, é aquela em menor profundidade na árvore de busca



Medida de Desempenho do Algoritmo de Busca

- Uma estratégia de busca é definida pela escolha da **ordem da expansão de nós**
- Estratégias são avaliadas de acordo com os seguintes critérios:
 - **Complexidade De Tempo (Custo de Busca):** quanto tempo ele leva para encontrar a solução? - Número de nós gerados
 - **Complexidade De Espaço (Custo de Busca):** quanta memória é necessária para executar a busca? - Número máximo de nós na memória.

Custo Total

Custo do Caminho + Custo de Busca.

Métodos de Busca

- **Busca Cega ou Exaustiva:**

- Não tem nenhuma informação adicional sobre os estados, isto é, não sabe qual o melhor nó da fronteira a ser expandido. Apenas distingue o estado objetivo dos não objetivos.

- **Busca Heurística:**

- Ou busca com informação, estima qual o melhor nó da fronteira a ser expandido baseado em funções heurísticas.

- **Busca Competitiva:**

- Considera que há oponentes hostis e imprevisíveis. Ex: Jogos

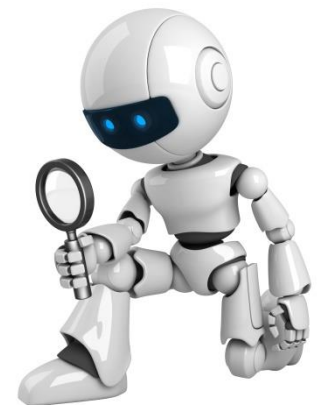
- **Busca Local:**

- Operam em um único estado e movem-se para a vizinhança deste estado.
- **Algoritmos Genéticos:**
 - Variante de Busca Local em que é mantida uma grande população de estados. Novos estados são gerados por mutação e por crossover, que combina pares de estados da população.

Algoritmos de Busca Cega ou Exaustiva

As estratégias de busca sem informação se distinguem pela **ordem** em que os nós são expandidos.

1. Busca em extensão/largura;
2. Busca em profundidade;
3. Busca por aprofundamento iterativo;
4. Busca de custo uniforme.



Métodos de Busca

- **Busca Cega ou Exaustiva:**

- Não tem nenhuma informação adicional sobre os estados, isto é, não sabe qual o melhor nó da fronteira a ser expandido. Apenas distingue o estado objetivo dos não objetivos.

- **Busca Heurística:**

- Ou busca com informação, estima qual o melhor nó da fronteira a ser expandido baseado em funções heurísticas.

- **Busca Competitiva:**

- Considera que há oponentes hostis e imprevisíveis. Ex: Jogos

- **Busca Local:**

- Operam em um único estado e movem-se para a vizinhança deste estado.
- **Algoritmos Genéticos:**
 - Variante de Busca Local em que é mantida uma grande população de estados. Novos estados são gerados por mutação e por crossover, que combina pares de estados da população.

Busca Heurística

- Algoritmos de Busca Heurística

- Busca Gulosa

- Avalia os nós usando apenas a função heurística:

$$f(n) = h(n)$$



- A*

- Combina o custo do caminho $g(n)$ com o valor da heurística $h(n)$

$$f(n) = g(n) + h(n)$$



Até aqui...

- Problemas sem interação com outro agente.
- O agente possui total controle sobre suas ações e sobre o efeito de suas ações.
- Muitas vezes encontrar a solução ótima é factível.

Jogos X Busca

O oponente é “imprevisível”

O agente tem que levar em consideração todos os movimentos possíveis do oponente.

Métodos de Busca

- **Busca Cega ou Exaustiva:**

- Não tem nenhuma informação adicional sobre os estados, isto é, não sabe qual o melhor nó da fronteira a ser expandido. Apenas distingue o estado objetivo dos não objetivos.

- **Busca Heurística:**

- Ou busca com informação, estima qual o melhor nó da fronteira a ser expandido baseado em funções heurísticas.

- **Busca Competitiva:**

- Considera que há oponentes hostis e imprevisíveis. Ex: Jogos

- **Busca Local:**

- Operam em um único estado e movem-se para a vizinhança deste estado.

- **Algoritmos Genéticos:**

- Variante de Busca Local em que é mantida uma grande população de estados. Novos estados são gerados por mutação e por crossover, que combina pares de estados da população.

Decisões Ótimas

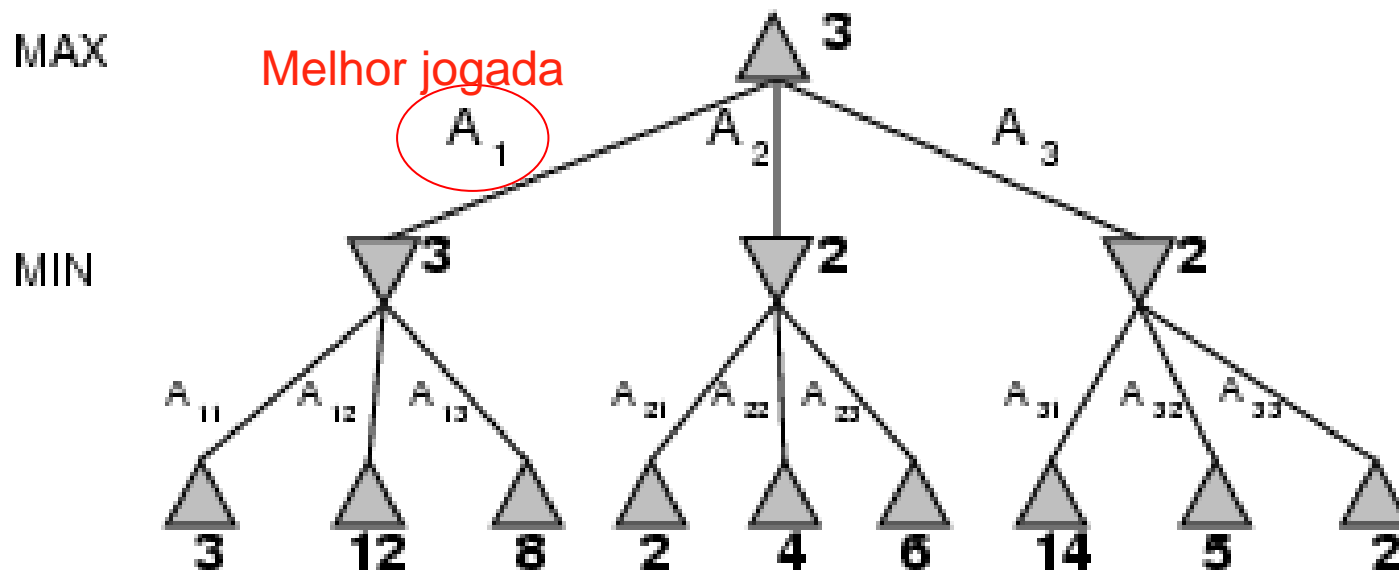
- Considerando jogos com 2 jogadores:
- MAX e MIN
 - MAX faz o primeiro movimento, tentando maximizar, escolher os melhores valores para ele mesmo.
 - O MIN (oponente) vai tentar escolher o pior valor para aquele jogador MAX.

Busca Competitiva

- Algoritmos de Busca Competitiva
 - Algoritmo Minimax
 - Poda alfa-beta
 - Jogos Estocásticos

Minimax

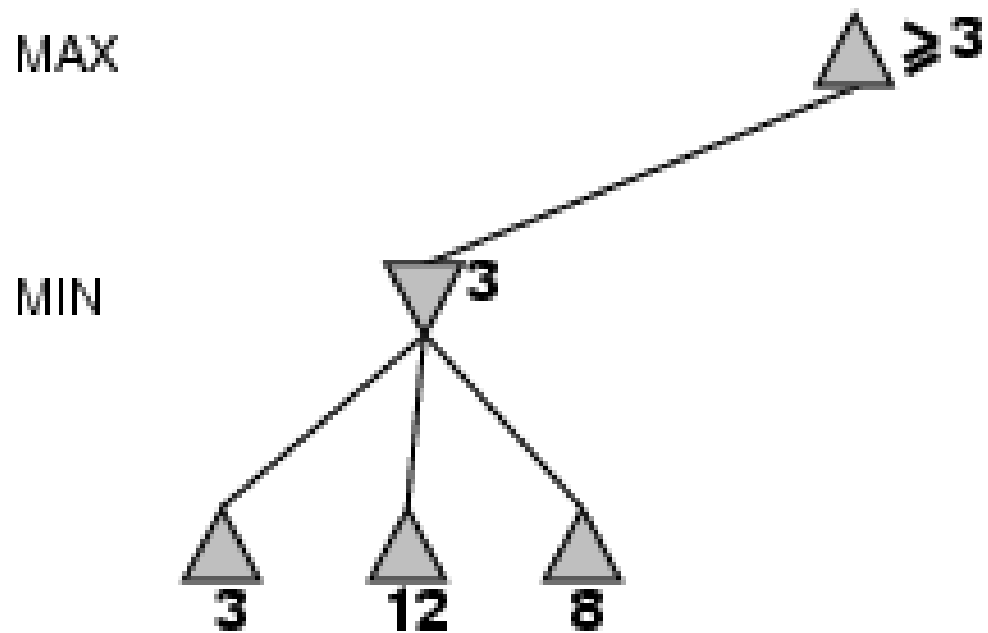
- Melhor estratégia para jogos determinísticos
- Ideia: escolher a jogada com o melhor retorno possível supondo que o oponente também vai fazer a melhor jogada possível
- Ex: Jogo simples, cada jogador faz um movimento



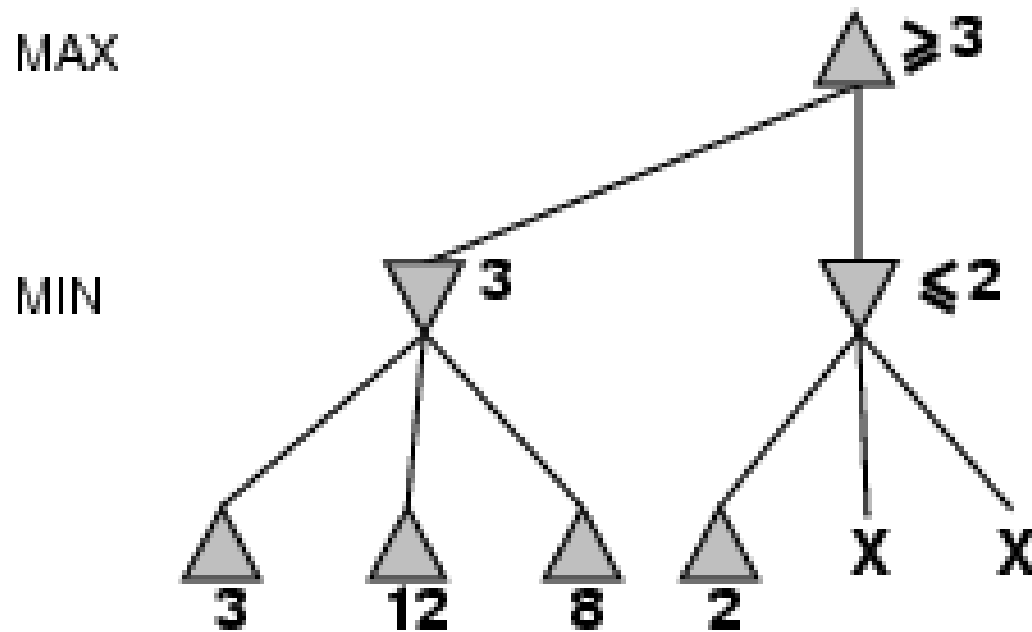
Poda α - β

- Algoritmo minimax: número de estados do jogo é exponencial em relação ao número de movimentos
- Poda α - β :
 - calcular a decisão correta sem examinar todos os nós da árvore,
 - retorna o mesmo que minimax, porém sem percorrer todos os estados.

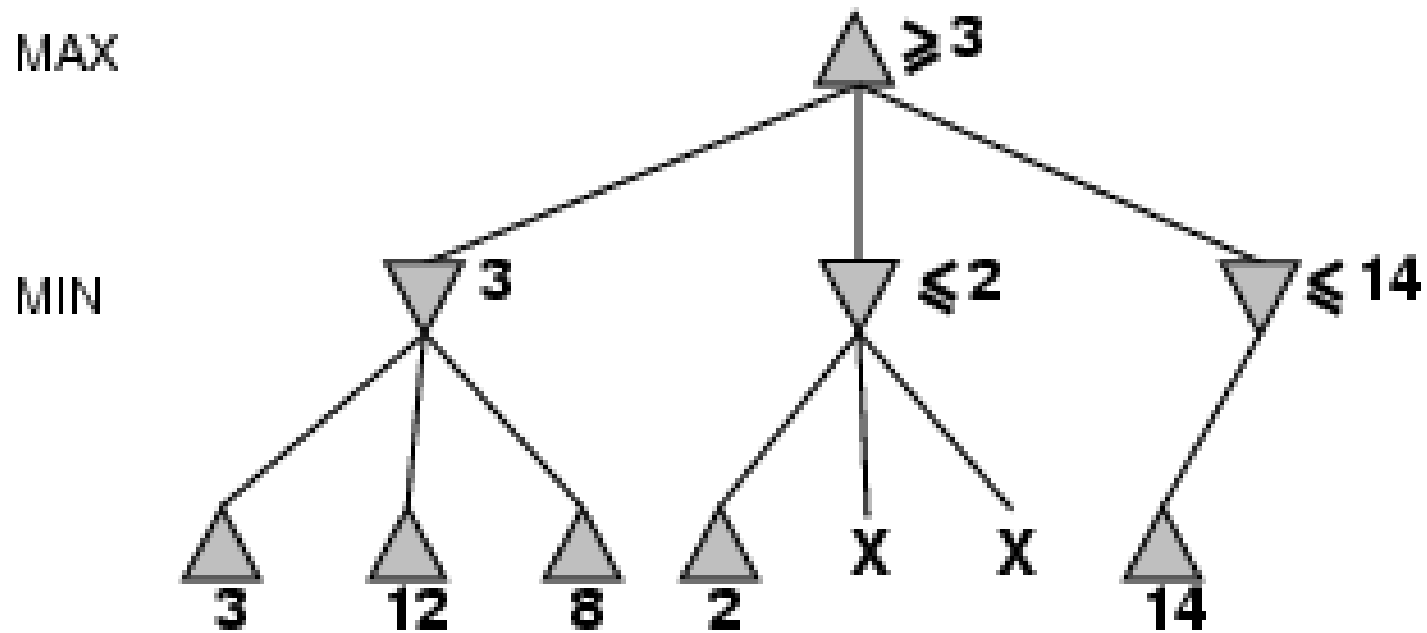
Poda α - β



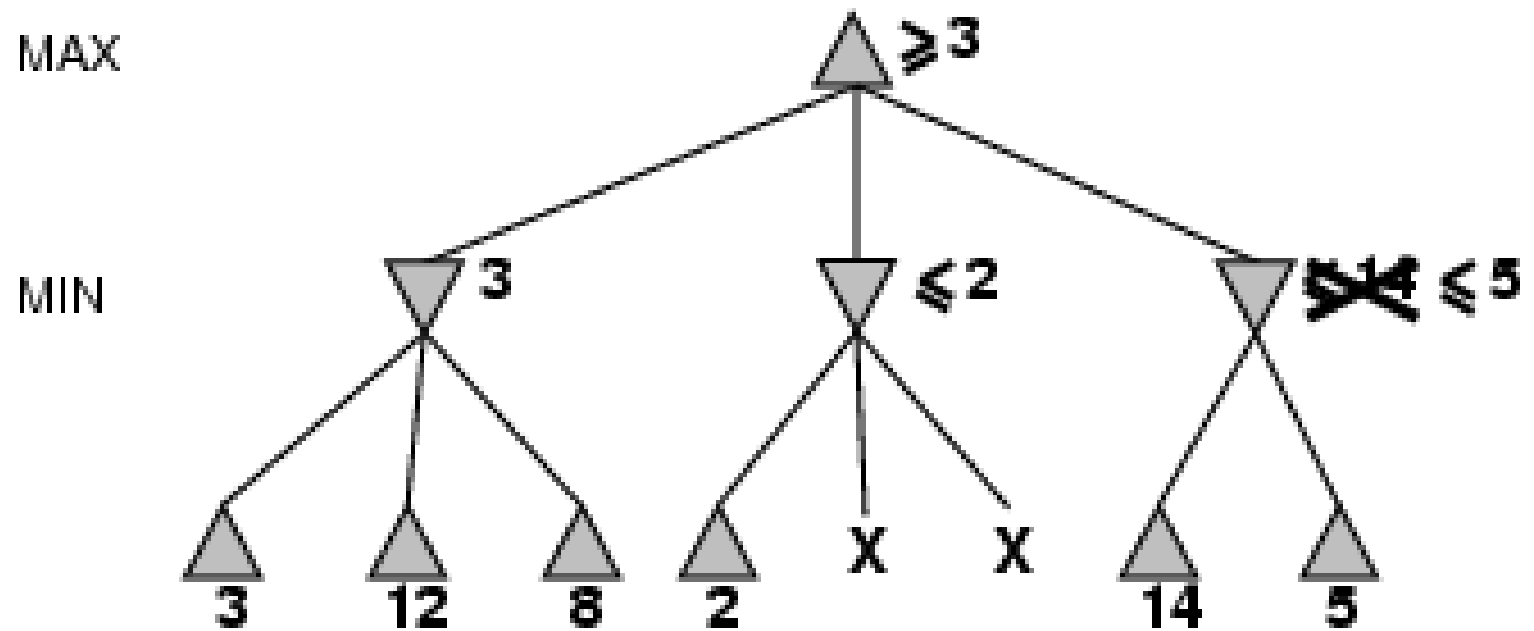
Poda α - β



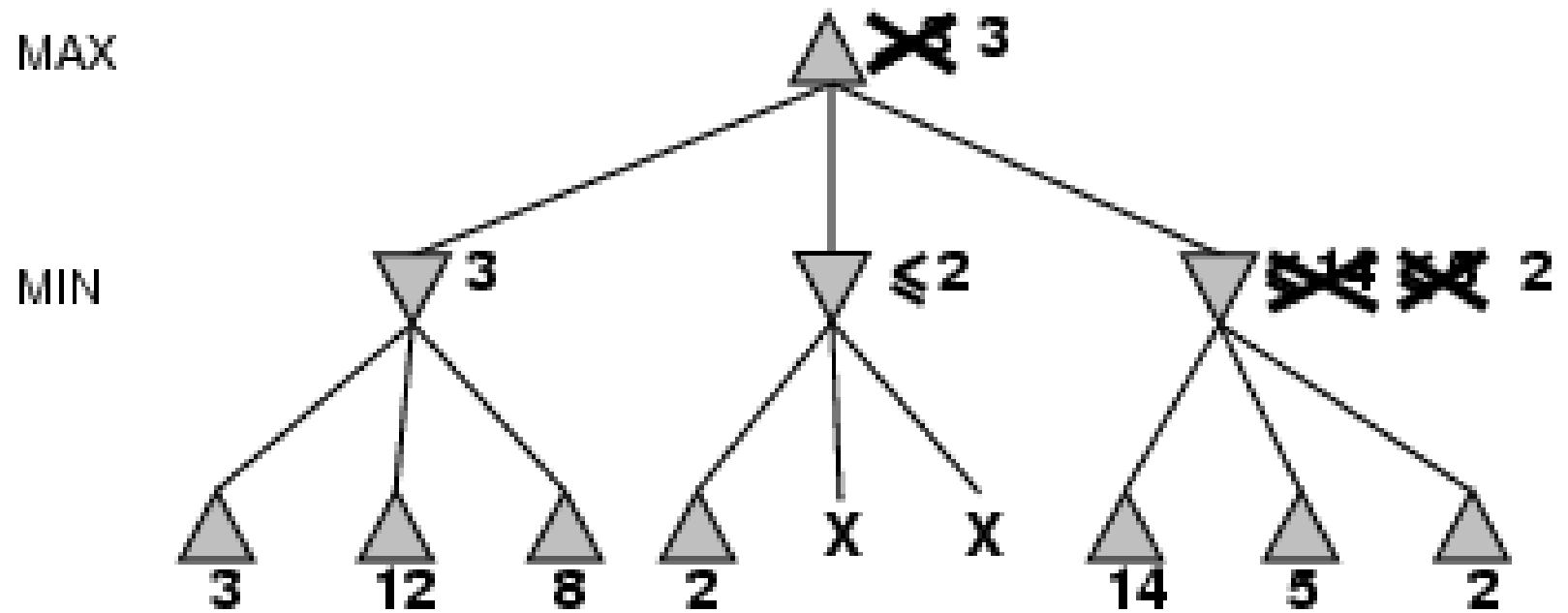
Poda α - β



Poda α - β



Poda α - β

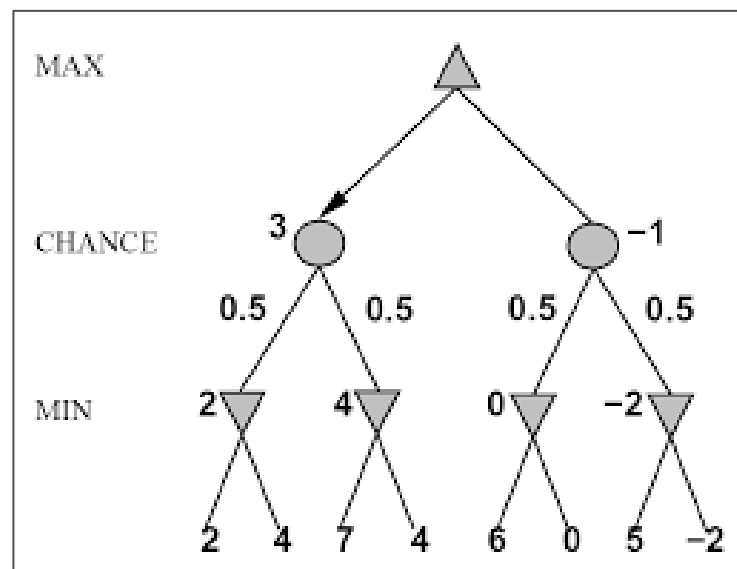
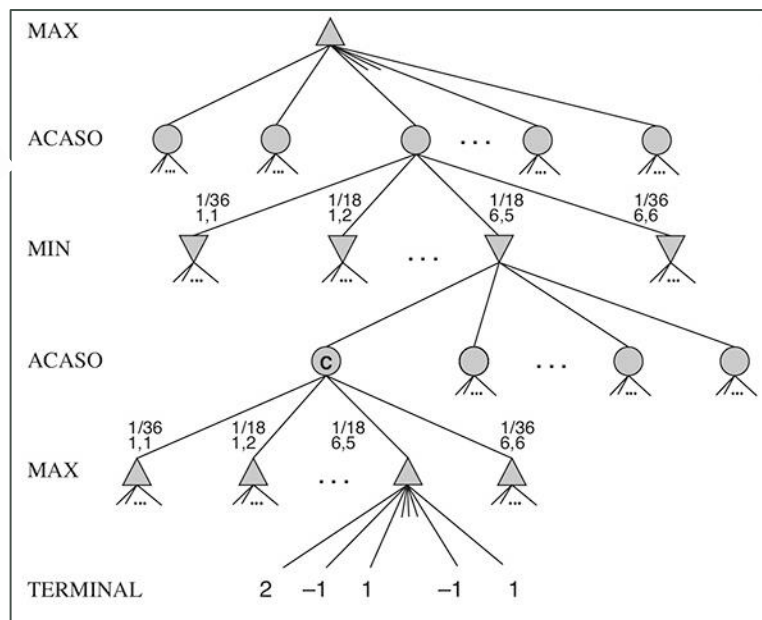


Jogos Estocásticos

- Muitos jogos refletem imprevisibilidade, incluindo o elemento aleatório proveniente de jogo de dados, sorteio de cartas, etc. (Jogos de azar)
- Não-determinismo é inerente em ambientes reais.
 - O estudo de algoritmos para jogos com elemento aleatório é um passo em direção a algoritmos que podem ser aplicados no mundo real.
- Uma árvore de um jogo não-determinístico deve incluir **nós de acaso** além de nós minimax.

Jogos Estocásticos: EXPECTMINMAX

- Ramificações que levam a **nós de acaso** denotam “jogadas de dados possíveis” (anotadas com a **probabilidade** de cada mudança de estado).
 - Se faz a média ponderada dos valores ao acaso para escolher o valor de max.



Métodos de Busca

- **Busca Cega ou Exaustiva:**

- Não tem nenhuma informação adicional sobre os estados, isto é, não sabe qual o melhor nó da fronteira a ser expandido. Apenas distingue o estado objetivo dos não objetivos.

- **Busca Heurística:**

- Ou busca com informação, estima qual o melhor nó da fronteira a ser expandido baseado em funções heurísticas.

- **Busca Competitiva:**

- Considera que há oponentes hostis e imprevisíveis. Ex: Jogos

- **Busca Local:**

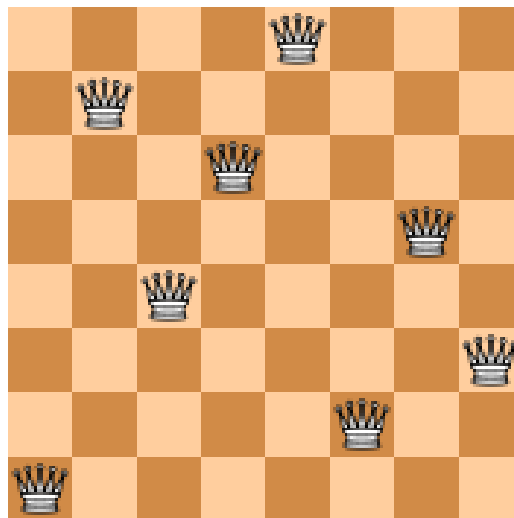
- Operam em um único estado e movem-se para a vizinhança deste estado.
- **Algoritmos Genéticos:**
 - Variante de Busca Local em que é mantida uma grande população de estados. Novos estados são gerados por mutação e por crossover, que combina pares de estados da população.

Busca Local

- Em muitos problemas o **caminho para a solução é irrelevante**.
 - Queremos apenas encontrar o estado objetivo, não importando a sequência de ações.
 - Espaço de estados = conjunto de configurações completas.
 - Queremos encontrar a melhor configuração.
 - Neste caso podemos usar algoritmos de busca local.
 - Mantêm apenas o estado atual, sem a necessidade de manter a árvore de busca.
- Se o caminho para a solução não importa, podemos utilizar um algoritmo de **busca local**.

Busca Local: Exemplo

- 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 posicionadas.



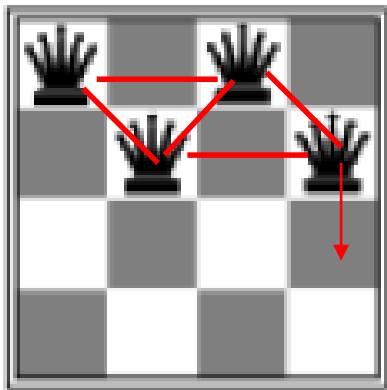
Busca Local: Exemplo

- 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 posicionadas.
- **Outros exemplos:**
 - Projeto de Circuitos eletrônicos;
 - Layout de instalações industriais;
 - Escalonamento de jornadas de trabalho;
 - Otimização de redes de telecomunicações.

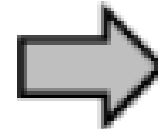
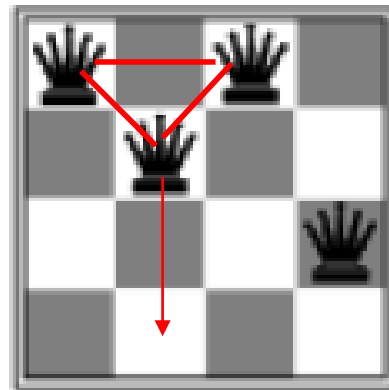
Exemplo: n -rainhas

- Colocar n rainhas em um tabuleiro $n \times n$, sendo que cada linha coluna ou diagonal pode ter apenas uma rainha.

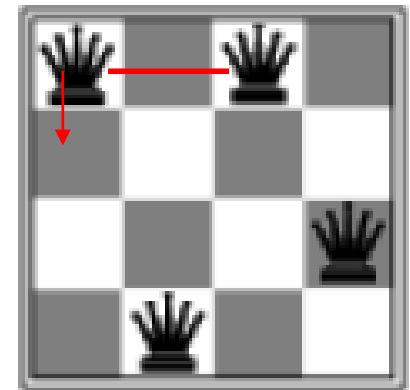
4 rainhas podem se atacar



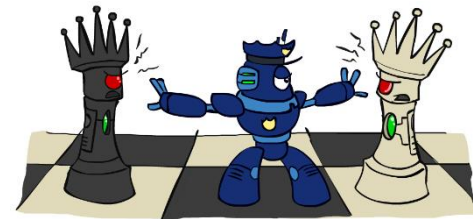
3 rainhas podem se atacar



2 rainhas podem se atacar



estado filho



Busca Local

- **Principais Algoritmos:**

- Hill Climbing (Busca de Subida de Encosta)
- Simulated Annealing (Busca de Têmpera Simulada)
- Local Beam (Busca de Feixe Local)
- Genetic Algorithms (Algoritmos Genéticos)

Busca de Subida de Encosta

- “É como tentar alcançar o cume do Monte Everest em meio a um nevoeiro denso durante uma crise de amnésia” (Russel, 2013 p. 107)



Busca de Subida de Encosta

função SUBIDA-DE-ENCOSTA(*problema*) **retorna** um estado que é um máximo local

corrente \leftarrow CRIAR-NÓ(ESTADO-INICIAL[*problema*])
repita

vizinho \leftarrow um sucessor de *corrente* com valor mais alto

se VALOR[*vizinho*] \leq VALOR[*corrente*] então retornar ESTADO[*corrente*]
corrente \leftarrow *vizinho*

Se há mais de um vizinho com a melhor qualidade:

- Escolhe o primeiro melhor
- Escolhe um entre todos de forma aleatória

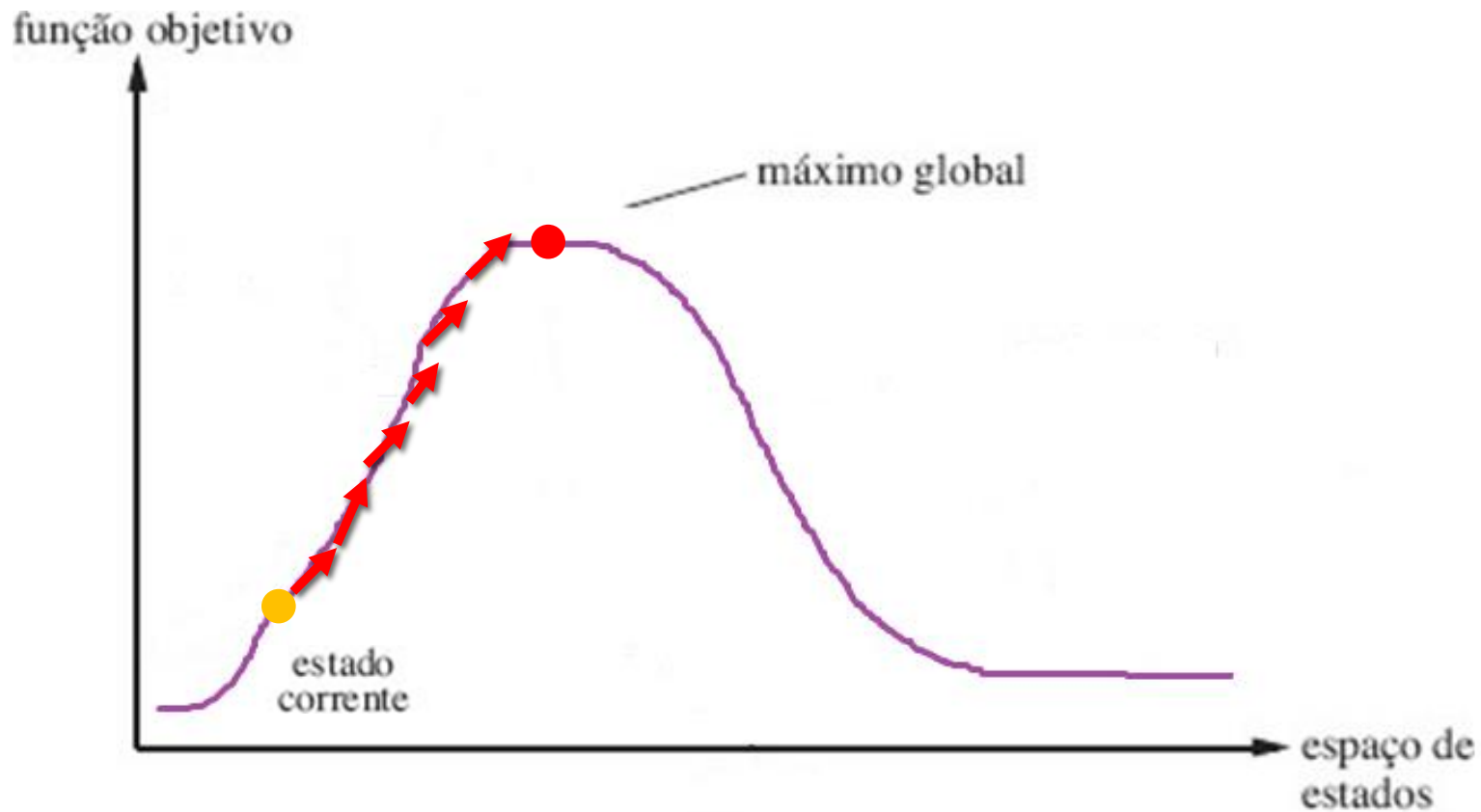
- Consiste de um loop que continuamente move-se para os estados que aumentam o valor em sua **função de avaliação** (i.e., busco o estado mais promissor que o estado atual).
- Termina quando encontra um "pico" (ou vale) em que nenhum vizinho tem valor maior.
- Também é conhecido como Gradiente Descendente e a Função de Avaliação é vista como custo



Busca de Subida de Encosta

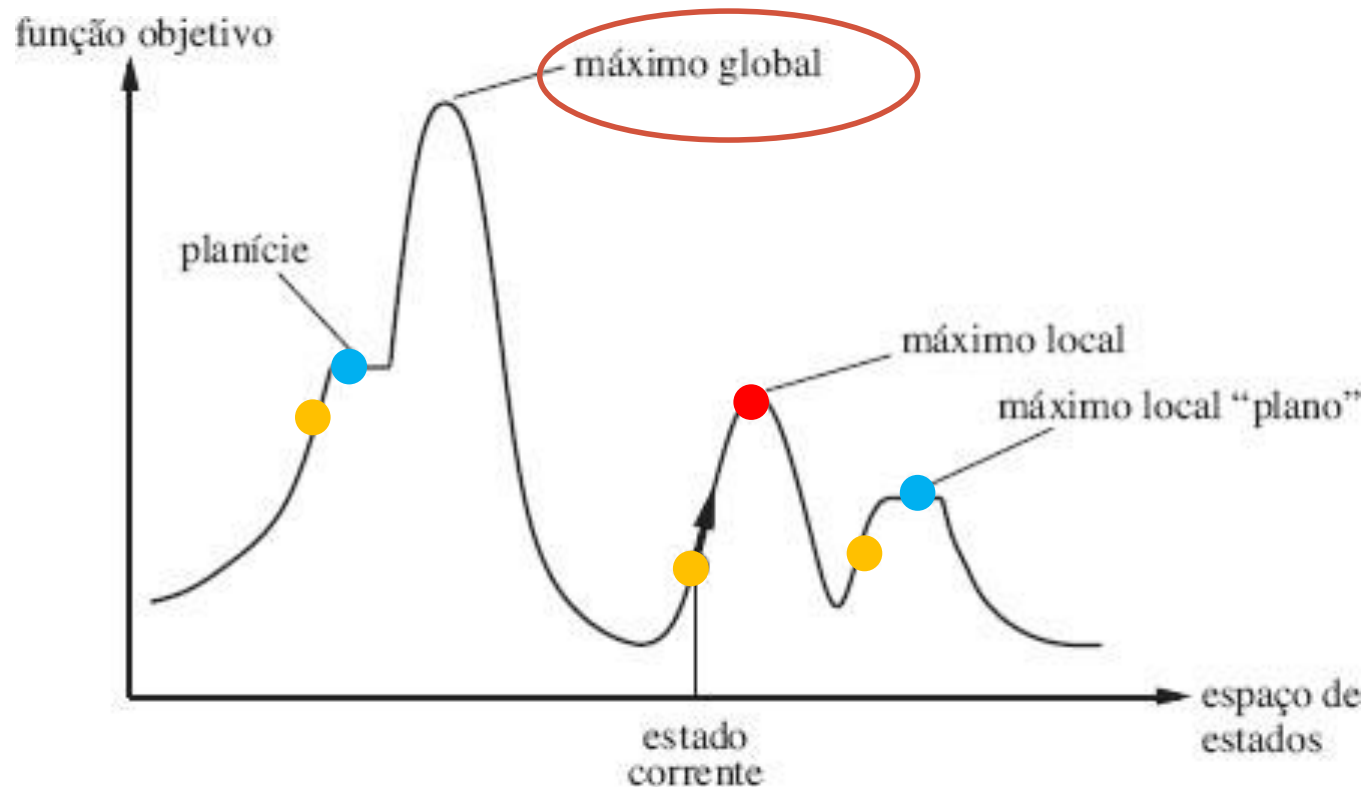
- Se a elevação corresponder ao custo, o objetivo é encontrar o **vale mais baixo = mínimo global**.
- Se a elevação corresponder a uma função objetivo, então o objetivo é encontrar o **vale mais alto = máximo global**.
- O algoritmo consiste em uma repetição que percorre o espaço de estados no sentido do valor crescente (ou decrescente).

Busca de Subida de Encosta



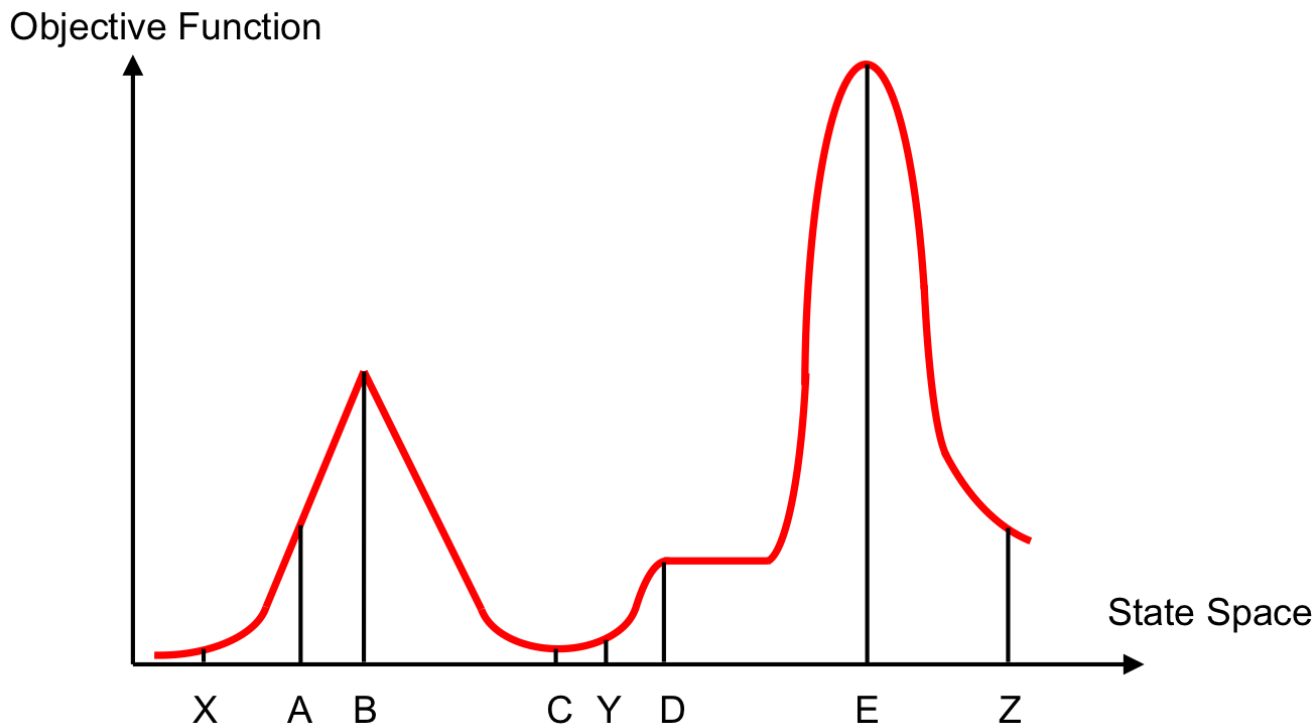
Busca de Subida de Encosta

- Problema: dependendo do estado inicial pode ficar presa em máximos (ou mínimos) locais.



Resposta Rápido

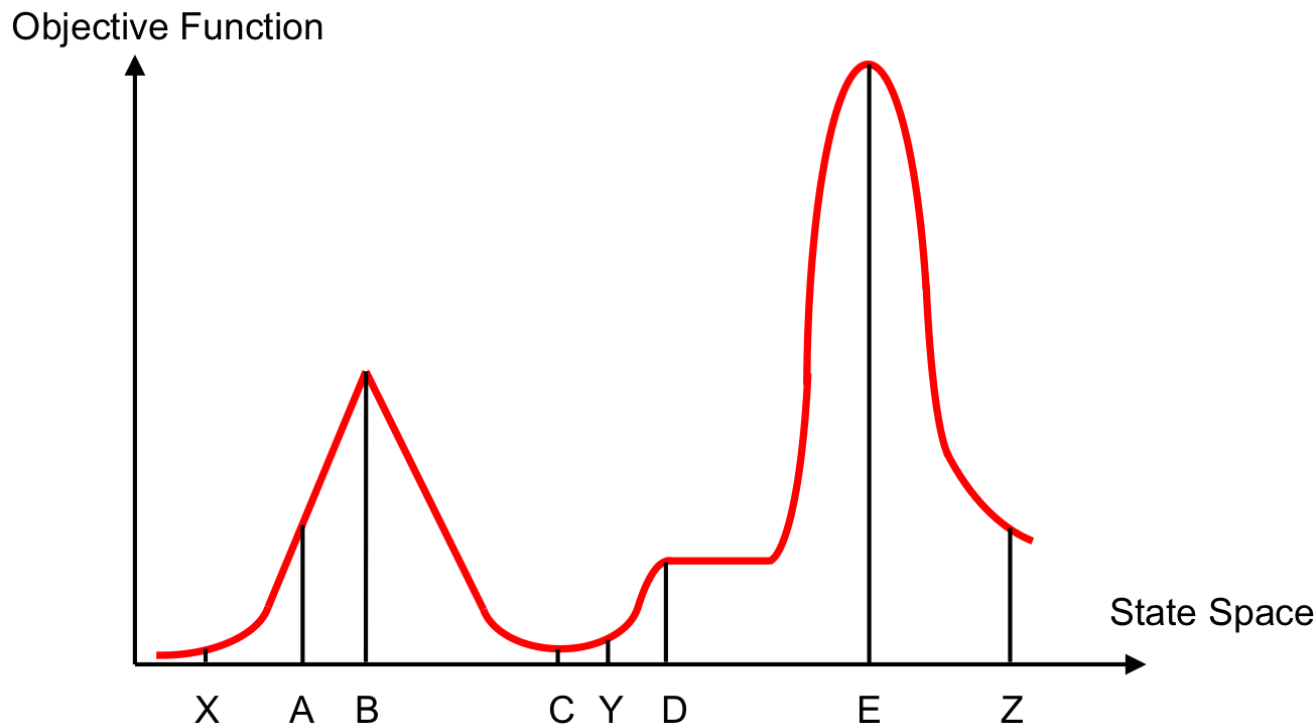
Considere o Algoritmo de Busca de Subida de Encosta. Se iniciar a partir da posição **X**, terminaremos em:



A
B
C
D
E

Resposta Rápido: Resposta

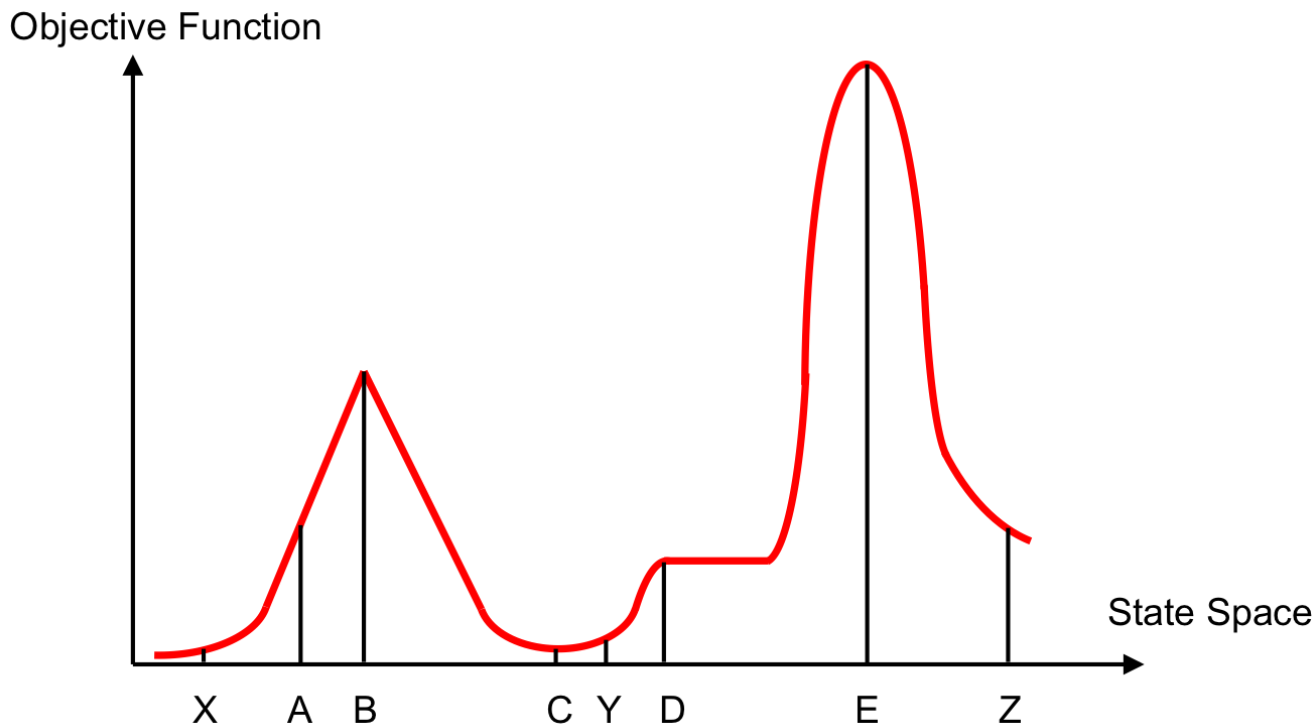
Considere o Algoritmo de Busca de Subida de Encosta. Se iniciar a partir da posição **X**, terminaremos em:



A
B
C
D
E

Resposta Rápido

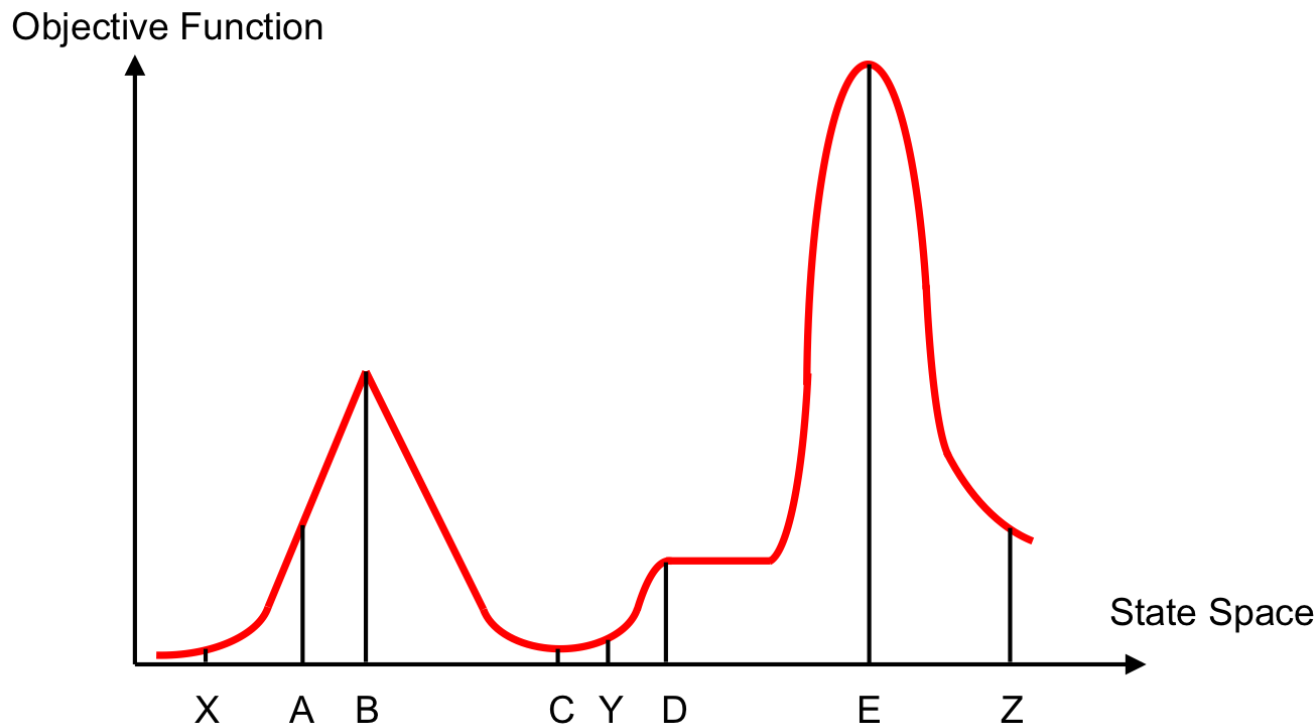
Considere o Algoritmo de Busca de Subida de Encosta. Se iniciar a partir da posição **Y**, terminaremos em:



A
B
C
D
E

Resposta Rápido: Resposta

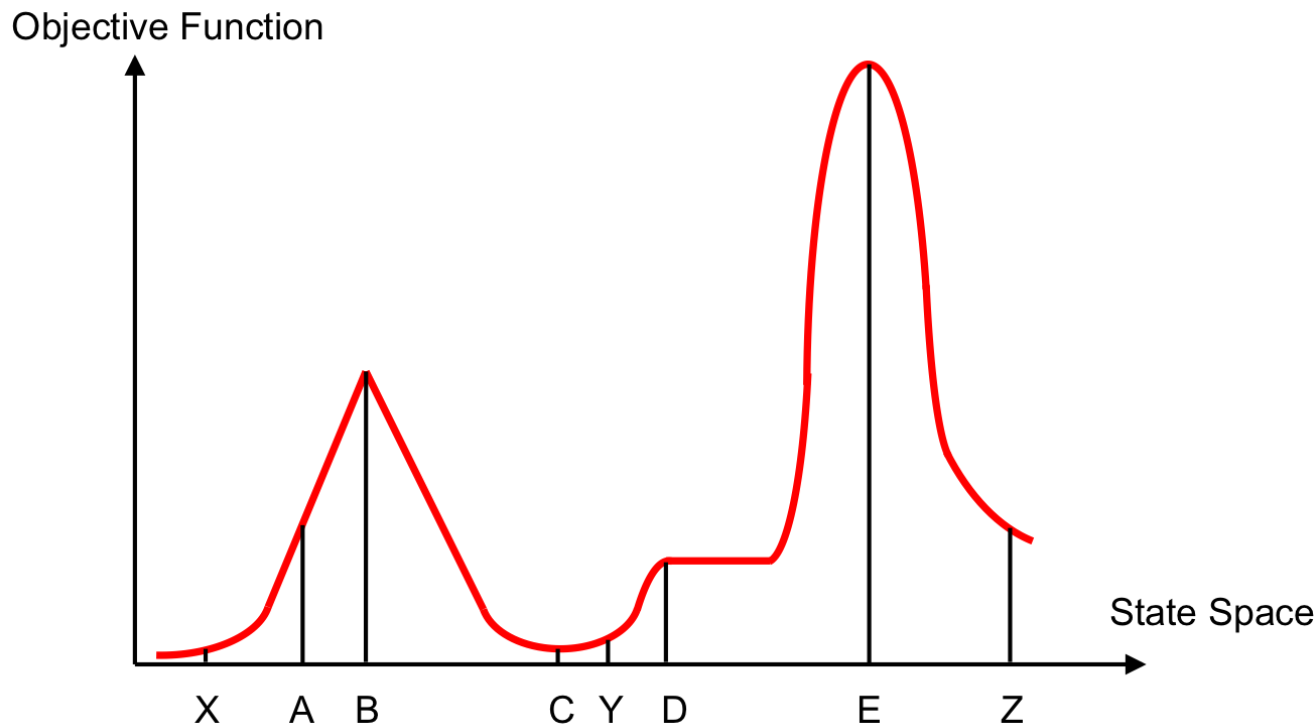
Considere o Algoritmo de Busca de Subida de Encosta. Se iniciar a partir da posição **Y**, terminaremos em:



A
B
C
D
E

Resposta Rápido

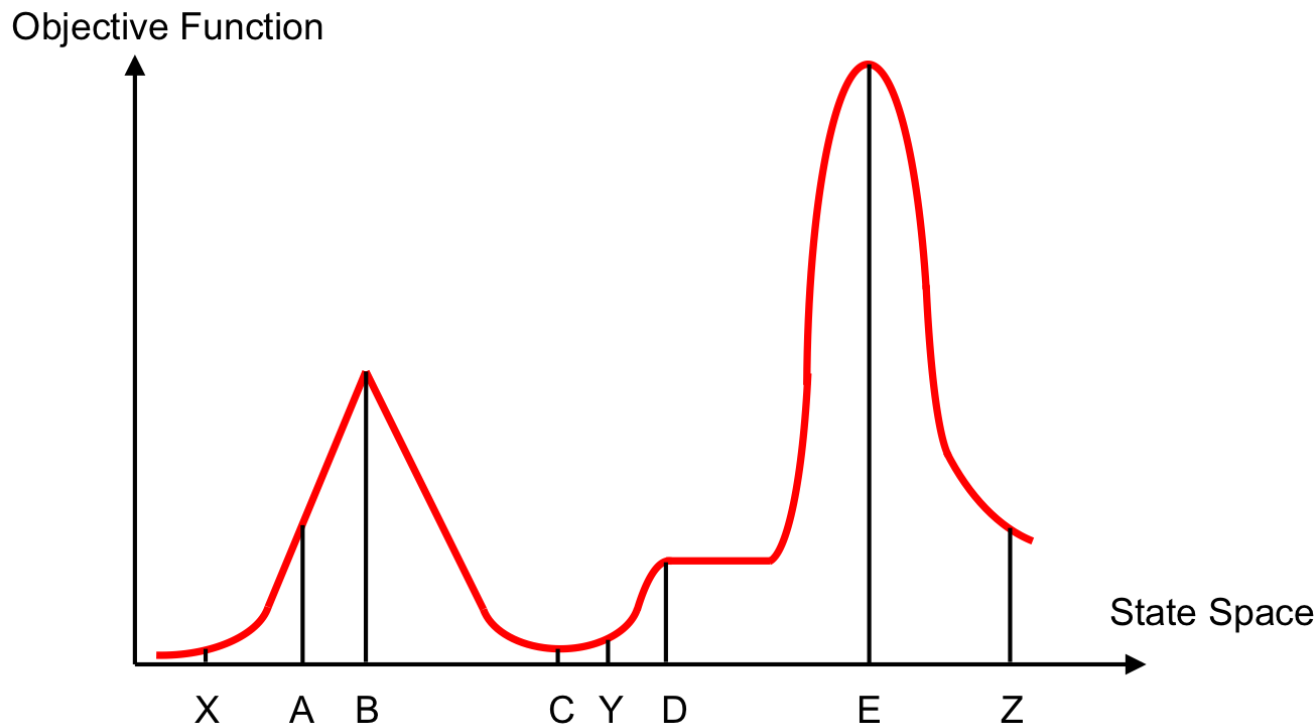
Considere o Algoritmo de Busca de Subida de Encosta. Se iniciar a partir da posição **Z**, terminaremos em:



A
B
C
D
E

Resposta Rápido: Resposta

Considere o Algoritmo de Busca de Subida de Encosta. Se iniciar a partir da posição **Z**, terminaremos em:



A
B
C
D
E

Busca de Subida de Encosta

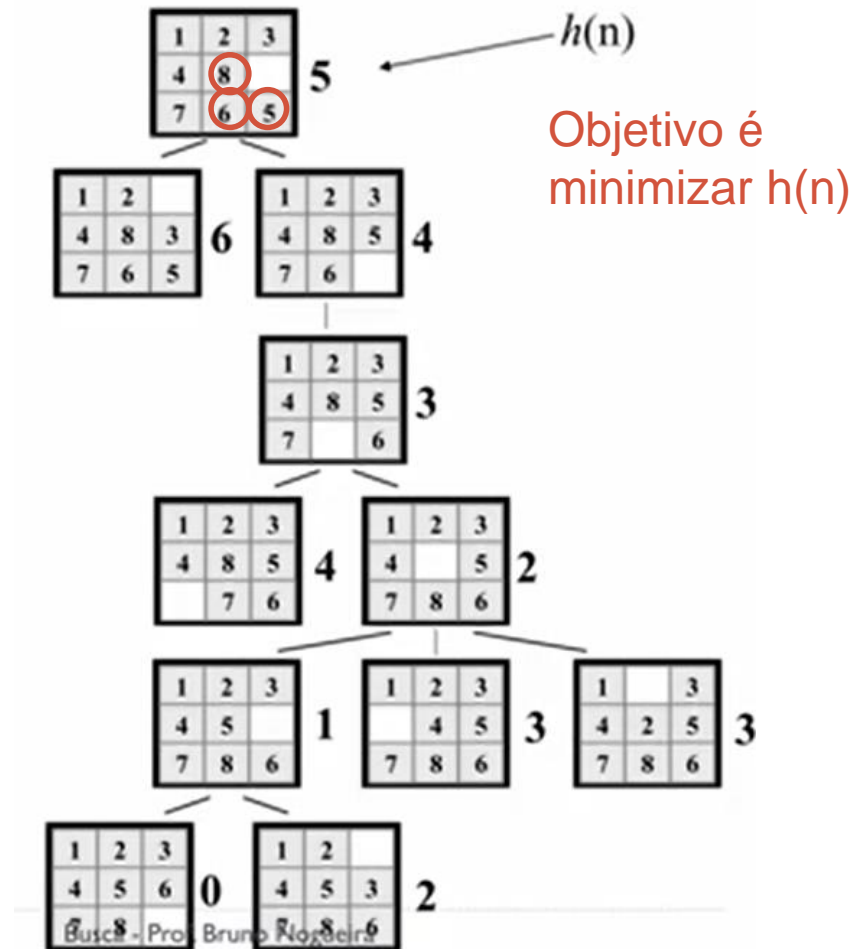
- Não mantém uma árvore de busca, o nó atual só registra o estado atual e o valor da função objetivo.
- Não examina antecipadamente valores de estados além dos valores dos vizinhos imediatos do estado atual.
- É um **algoritmo guloso** – escolhe sempre o primeiro melhor vizinho para progredir na busca.
 - Busca em profundidade + função de avaliação/heurística $h(n)$ – não considera a função de custo $g(n)$
- Essa abordagem pode ter **bons resultados em alguns problemas**. Sendo capaz de progredir rapidamente para a solução problema.

Busca de Subida de Encosta

Configuração final

1	2	3
4	5	6
7	8	

Cálculo da distância Manhattan

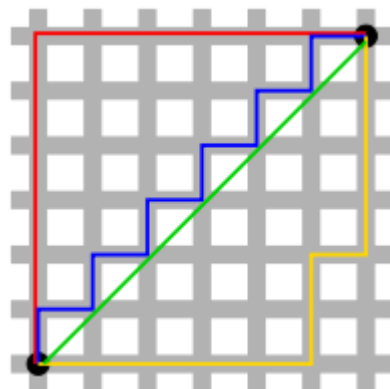


Distância Manhattan ou máxima, “city block”

Dado dois vetores X e Y , esta métrica é definida como o somatória dos módulos das diferenças:

$$d(X, Y) = \left(\sum_{i=1}^p |X_i - Y_i| \right)^1 = \\ |X_1 - Y_1| + |X_2 - Y_2| + \cdots + |X_p - Y_p|$$

$$d(x,y) = (|linha(x) - linha(y)| + |coluna(x) - coluna(y)|)$$



Os caminhos vermelho, amarelo e azul têm o mesmo menor comprimento de caminho igual a 12.

Na distância euclidiana, a linha verde tem comprimento aproximado de 8.49.

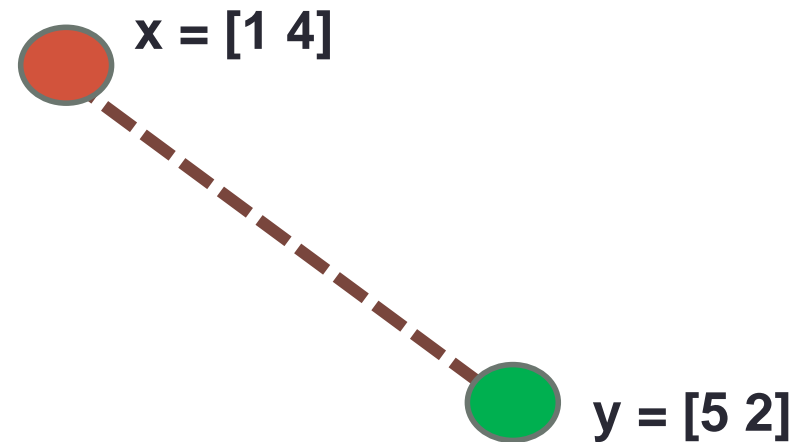
Distância Manhattan ou máxima, “city block”

$$d(X, Y) = (\sum_{i=1}^p |X_i - Y_i|)^1 = |X_1 - Y_1| + |X_2 - Y_2| + \cdots + |X_p - Y_p|$$

$$d(x,y) = (|linha(x) - linha(y)| + |coluna(x) - coluna(y)|)$$

Exemplo:

- $d(x,y) = |1-5| + |4-2|$
- $d(x,y) = 4 + 2$
- $d(x,y) = 6$



Busca de Subida de Encosta

Configuração final

1	2	3
4	5	6
7	8	

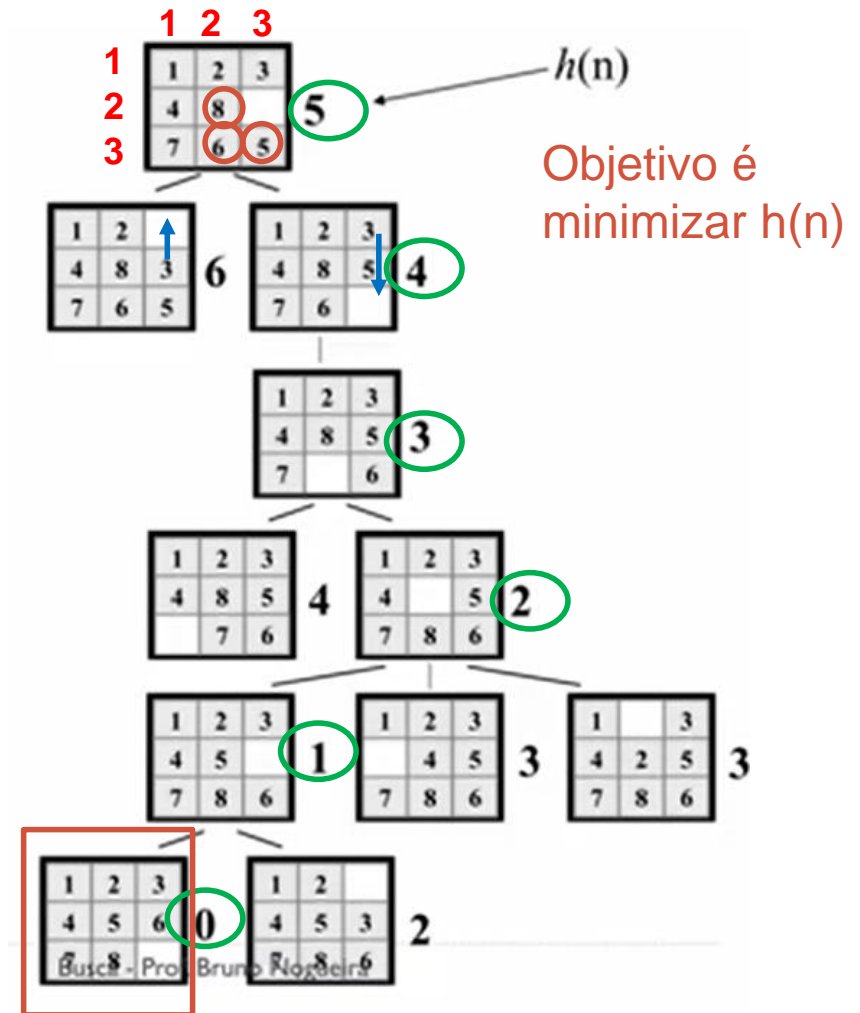
**Exemplo - Cálculo da distância
Manhattan do 8:**

Posição esperada = (2,3)

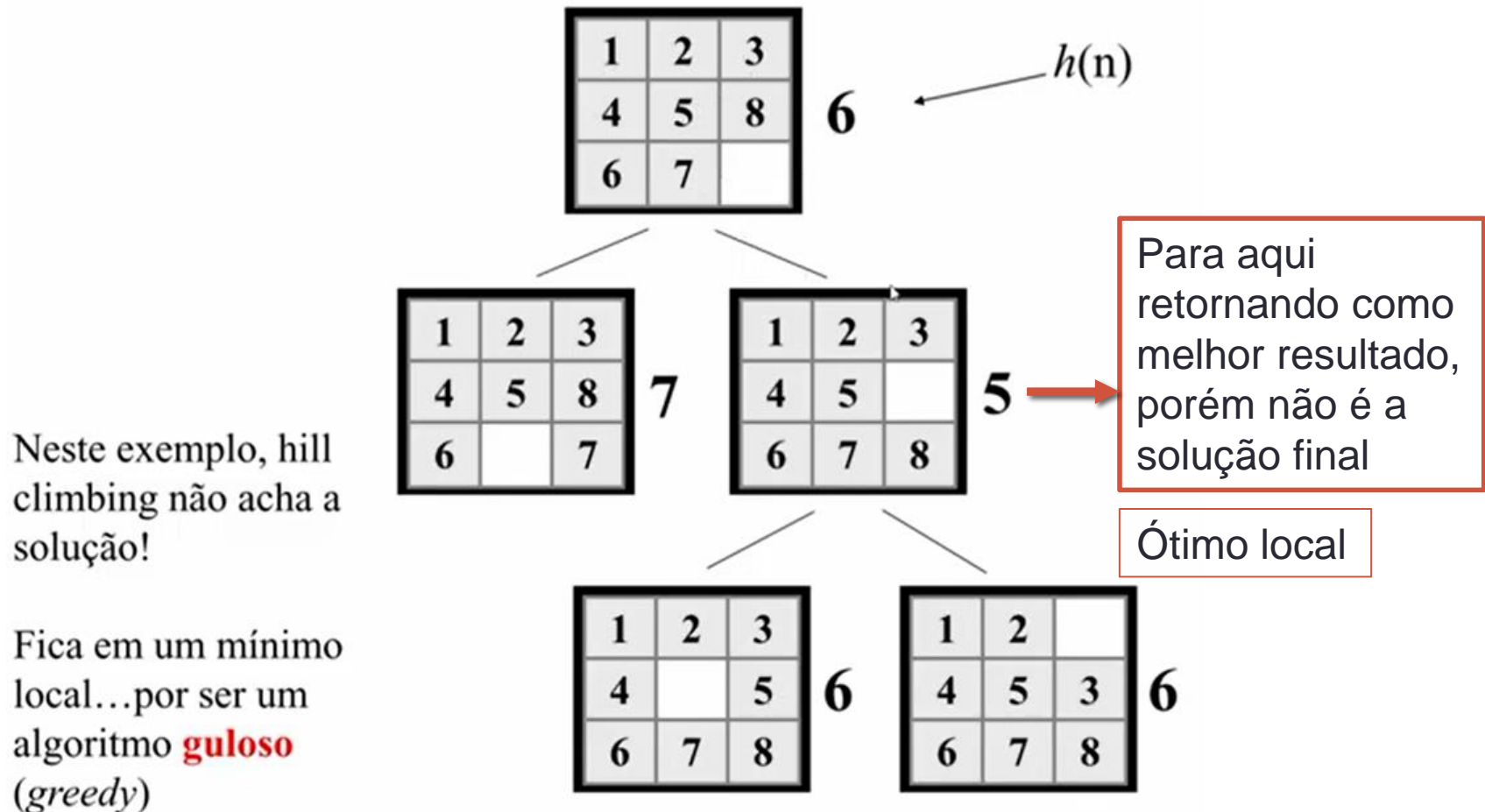
Posição atual = (2,2)

$$|2-2| + |3-2|$$

$$= 0+1=1$$



Busca de Subida de Encosta

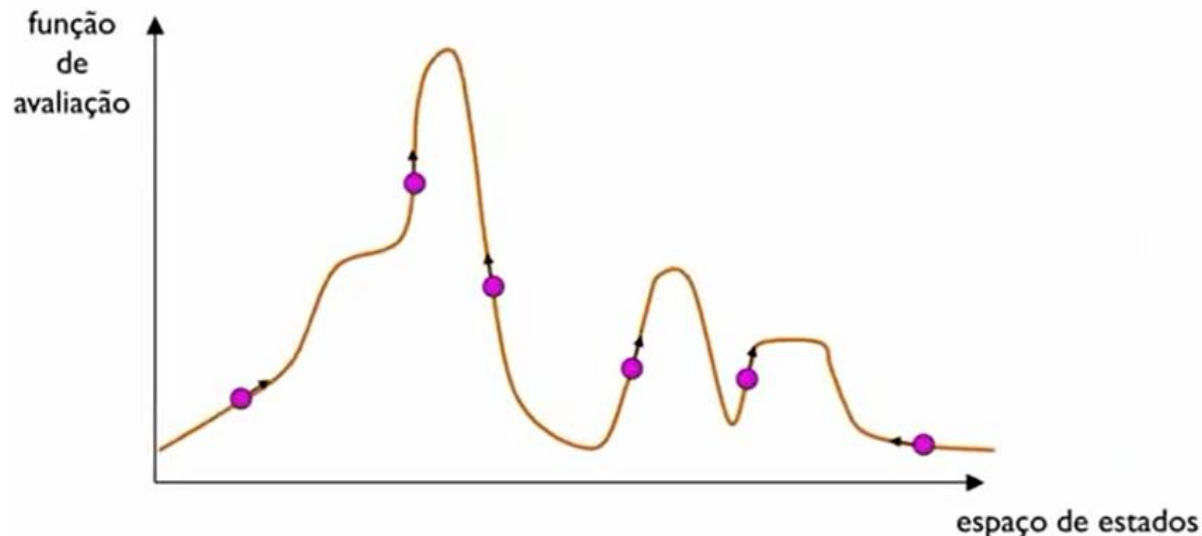


Subida de encosta: Variantes

- Subida de encosta estocástica:
 - **nem sempre escolhe o melhor vizinho** (ou o filho mais promissor);
 - escolhe de forma aleatória os movimentos.
- Subida de encosta pela primeira escolha:
 - **escolhe o primeiro bom vizinho que encontrar**;
 - implementa a subida de encosta estocástica gerando sucessores ao acaso até ser gerado um sucessor melhor que o estado corrente.
 - essa é uma boa estratégia quando um estado tem muitos sucessores (por exemplo, milhares).
- Os algoritmos de subida de encosta descritos até agora são incompletos — com frequência, eles deixam de encontrar um objetivo que existe porque ficam presos em máximos locais.

Subida de encosta: Variantes

- Subida de encosta com reinício aleatório
 - adota o ditado: “Se não tiver sucesso na primeira vez, continue tentando.”
 - conduz uma série de buscas de subida de encosta a partir de estados iniciais gerados (k) de forma aleatória, até encontrar um objetivo ou até que não exista progresso significativo.
 - o melhor de todos os resultados da busca é armazenado.



Busca Local

- **Vantagens:**

- Ocupam pouquíssima memória (normalmente constante).
- Podem encontrar soluções razoáveis em grandes ou infinitos espaços de estados.

- **São úteis para resolver problemas de otimização.**

- Nos quais o objetivo é encontrar o melhor estado de acordo com uma **função objetivo**.