



Inteligência Artificial



Unidade IV – Buscas Heurísticas

Profa. Vlândia Pinheiro

Adaptação de conteúdo dos Prof. André Coelho e Prof. Vasco Furtado

Motivação

- Relembrando:
 - **Busca cega**: qualquer estratégia de busca que **não** recebe informação adicional acerca do problema (somente a sua especificação em espaço de estados) e que tenta encontrar soluções (ótimas) pela geração sistemática de novos nós (estados), comparando-os ao estado-objetivo
- **Problema**:
 - Tais estratégias são **ineficientes** e **ineficazes** na maioria dos casos, pois só “olham para o presente” e não estimam quão longe ainda estão do alvo!
 - Apenas a BCU é capaz de calcular o custo de caminho do nó inicial ao atual ($g(n)$), mas essa informação não necessariamente conduz a busca na direção do objetivo

Buscas Heurísticas (Informed search)

■ Heurística:

- **Estratégias** que ajudam a encontrar uma solução orientando o processo de busca, sem a garantia de encontrar a solução ótima.
- É como um guia de turismo.
 - válida porque aponta para direções normalmente interessantes;
 - imprópria porque pode deixar de fora pontos de interesse sob determinados aspectos.
- Melhora eficiência mas sacrifica perfeição
- Exige conhecimento do problema.
 - uma heurística de sucesso em um dado problema pode não se aplicar a outro
- Sem a heurística estaríamos irremediavelmente presos a análises combinatórias

Buscas Heurísticas (Informed search)

■ Como aplicar heurística numa busca:

- Decidir sobre qual nó será feita a próxima expansão no espaço de busca, ao invés de expandir numa ordem de Profundidade ou Amplitude.
- Decidir que certos nós deveriam ser descartados ou podados da busca.
- Definir uma função heurística $h(n)$ a ser aplicada a cada nó n da fronteira atual da árvore de busca
 - Esta função estima o custo de caminho mais econômico desde o nó n até o **nó-objetivo**

Funções Heurísticas

- Exemplos:
 - Problema do viajante romeno:
 - $h_{DLR}(n)$: distância em linha reta entre o nó n e o destino
 - Jogo da velha:
 - $h_{MNV}(n)$: número de “linhas, colunas ou diagonais vencedoras em potencial” de uma config. de peças
- Uma heurística pode levar um algoritmo de busca a se perder, não encontrando soluções
 - Isso dependerá justamente da qualidade da heurística
- **Pergunta-chave:** O que seria uma boa heurística?
 - Resposta: p/ este estudo, ela nunca deve superestimar o custo real da solução → deve ser **admissível**!
 - $h_{DLR}(n)$ **é admissível**, pois o caminho mais curto entre dois pontos é sempre uma reta!
 - $h_{MNV}(n)$ **não é admissível**, pois o no. de passos que faltam p/ um nó-objetivo pode ser menor que tal valor

Buscas Heurísticas (Informed search)

■ Tipos de Buscas Heurísticas:

■ Buscas Locais

- Hill Climbing ou Subida da Encosta

■ Buscas pela Melhor Escolha

- Best First (melhor escolha primeiro)

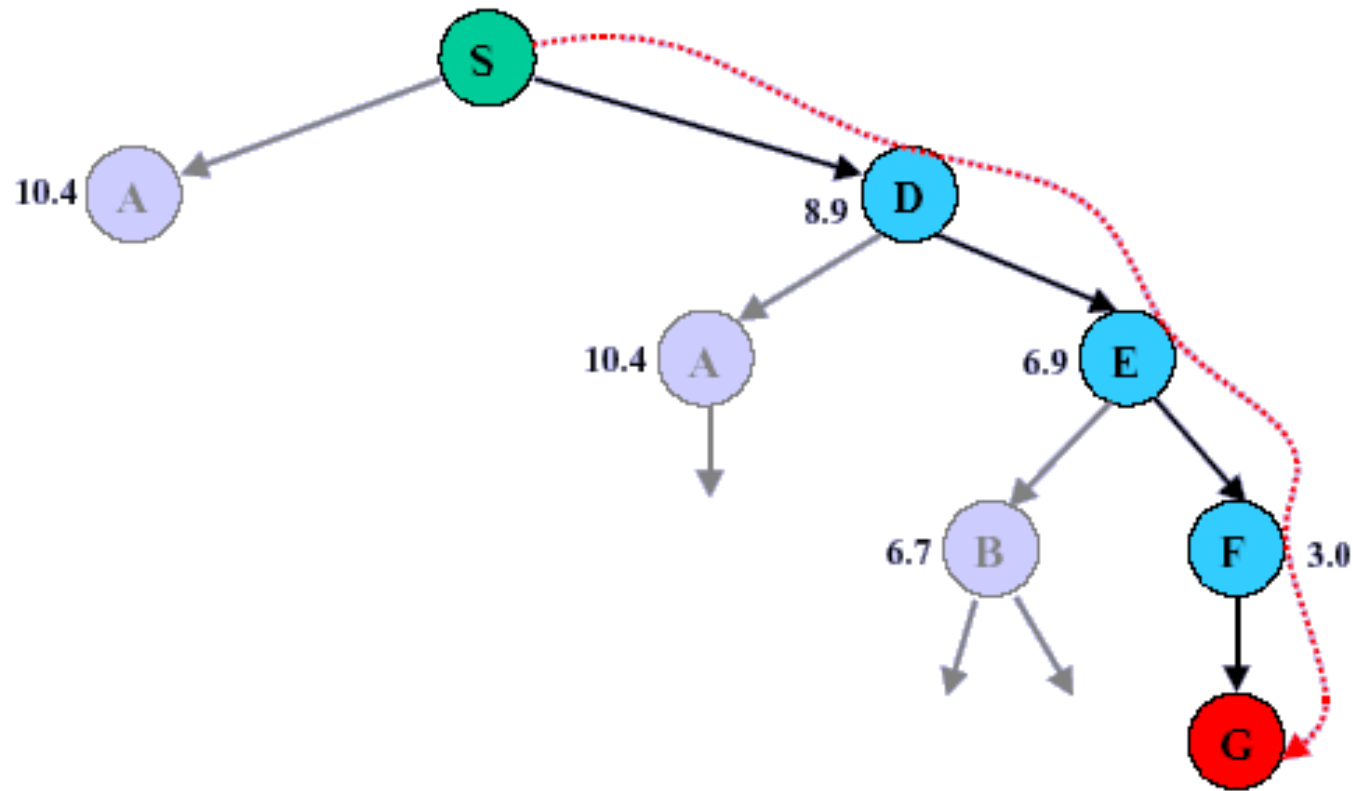
- A*

Busca Hill Climbing (subida da encosta)

- É similar à Busca em Profundidade, mas a seleção não é arbitrária.
- É um método de busca local que usa a ideia de que o objetivo deve ser atingido com o menor número de passos.
- A seleção é baseada numa estimativa de proximidade com o estado objetivo (custo estimado de *n* ao nó objetivo).
- O algoritmo *não* mantém uma árvore de busca:
 - Guarda apenas o estado atual e sua avaliação.
 - É simplesmente um “loop” que se move na direção *crescente* (para maximizar) ou *decrecente* (para minimizar) da função de avaliação.

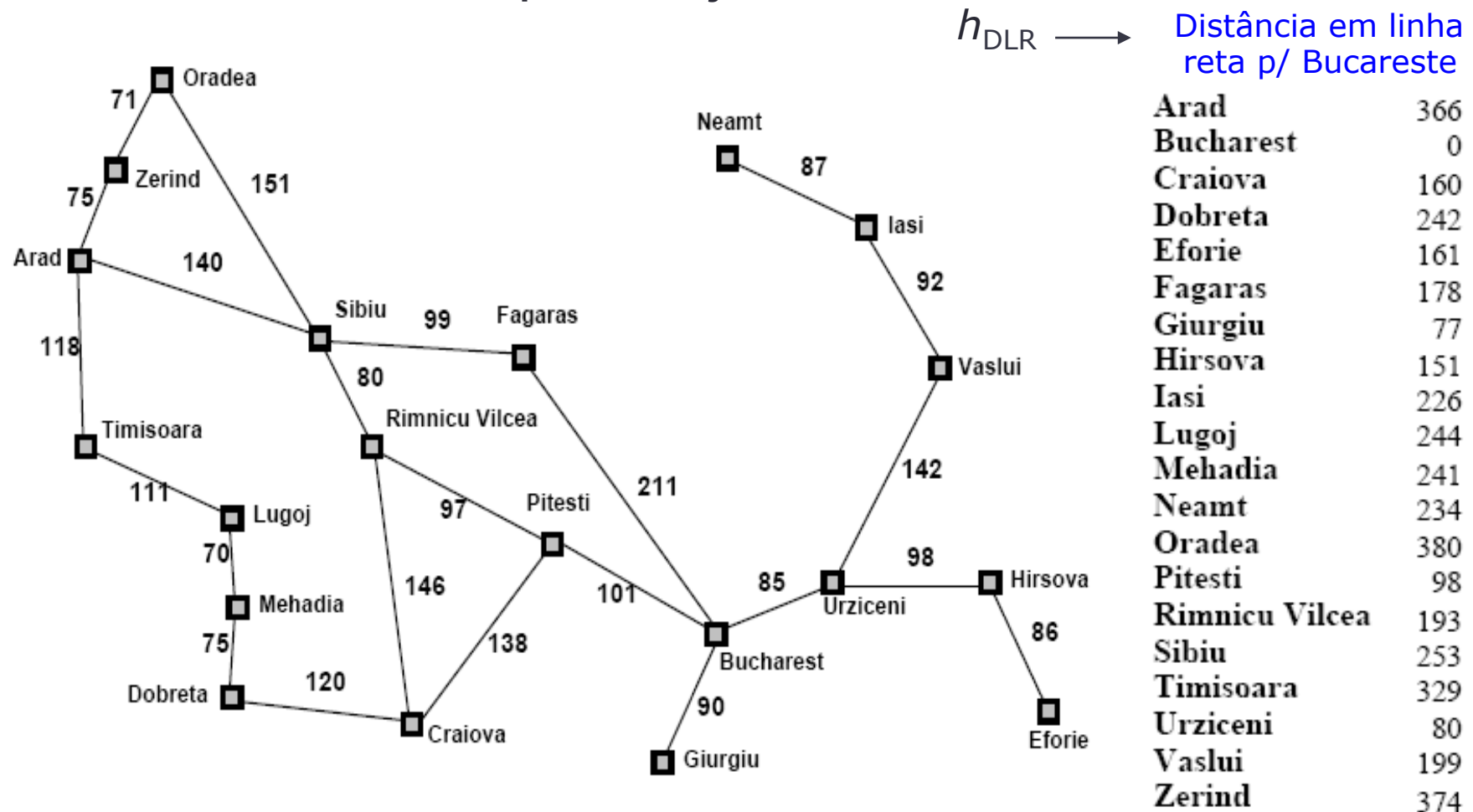
Busca Hill Climbing (subida da encosta)

■ Exemplo



Busca Hill Climbing (subida da encosta)

Exemplo: Viajante da Romênia



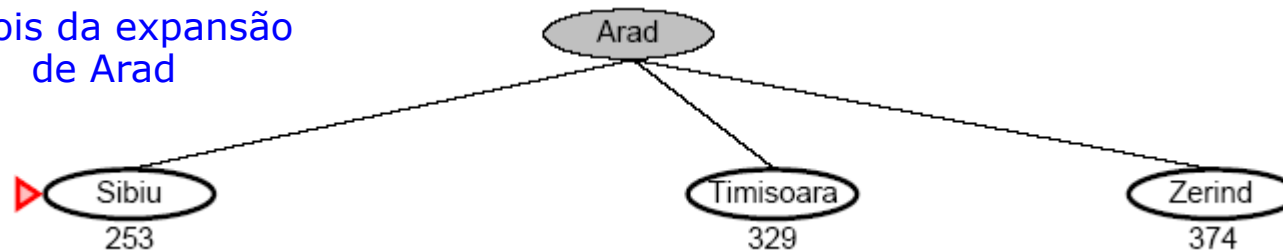
Busca Hill Climbing (subida da encosta)

Exemplo: Viajante da Romênia

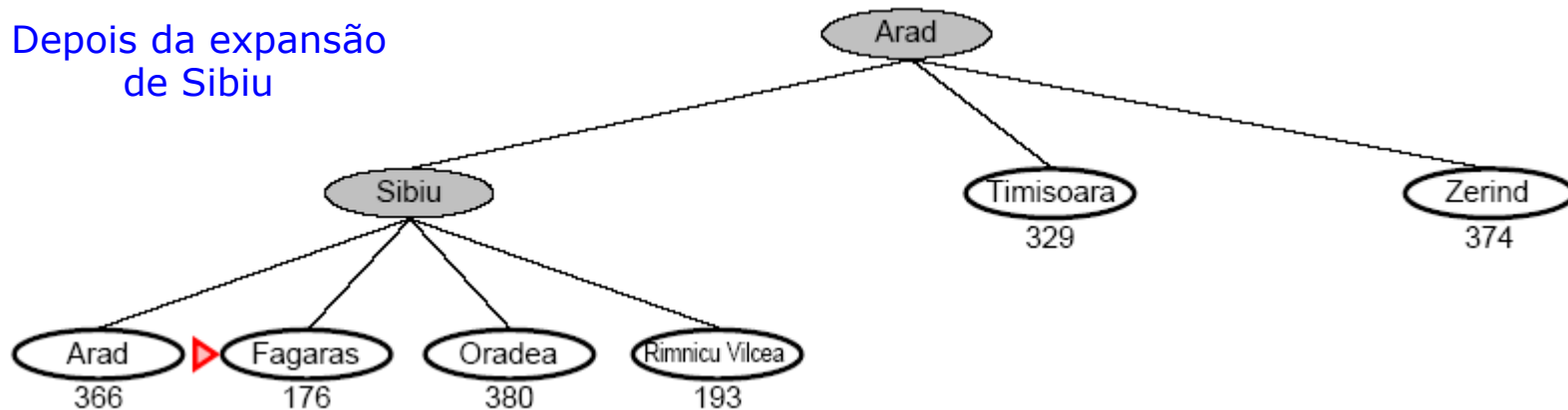
Estado inicial



Depois da expansão
de Arad



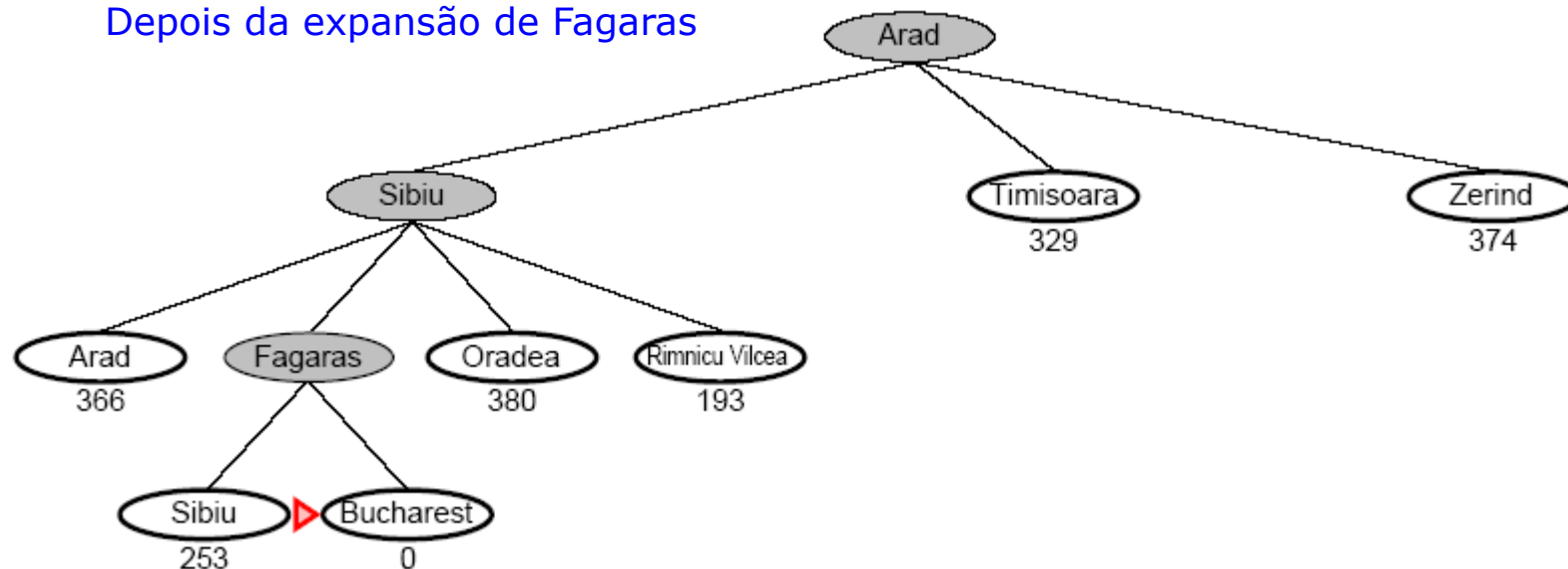
Depois da expansão
de Sibiu



Busca Hill Climbing (subida da encosta)

Exemplo: Viajante da Romênia

Depois da expansão de Fagaras



1. **custo temporal/espacial da busca foi mínimo**, pois não encontrou uma solução sem testar nenhum nó que não esteja no caminho da solução.
2. **não é ótima**, pois o caminho até Bucareste passando por Sibiu e Fagaras é 32 Km mais longo que o caminho por Rimnicu, Vilcea e Pitesti.

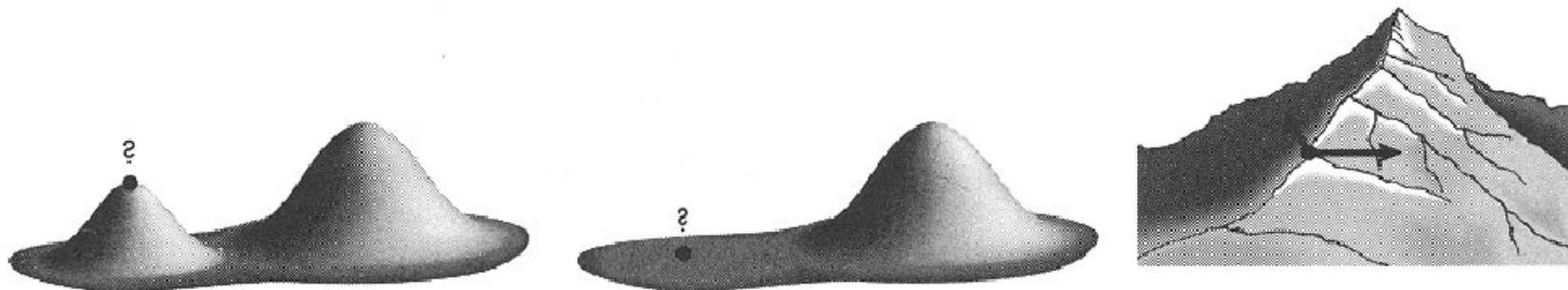
Busca Hill Climbing (subida da encosta)

- A “visão” desta estratégia é de curto alcance!
 - Só está “preocupada” c/ o próximo passo e não c/ o impacto dessa decisão nas próximas decisões
- Mesmos defeitos da Busca em Profundidade
 - Embora geralmente eficiente e com bom desempenho, **não** garante encontrar a solução ótima!
 - **Não** é nem completa, pois pode entrar em um caminho infinito (*loop*), e nunca retornar p/ experimentar outras possibilidades
 - P. ex.: p/ ir de Iasi até Fagaras, a heurística sugere expandir Neamt primeiro, mas lá é um beco sem saída

Busca Hill Climbing (subida da encosta)

■ Este algoritmo pode apresentar três problemas:

- **Contrafortes (ou máximos locais):** Em contraste com máximos globais, são picos mais baixos do que o pico mais alto no espaço de estados (solução ótima).
- **Platôs (planícies):** Uma região do espaço de estados onde a função de avaliação dá o mesmo resultado.
- **Cristas:** Apesar de existir uma direção que leva ao pico, nenhum dos operadores válidos sozinho conduz o algoritmo nessa direção.

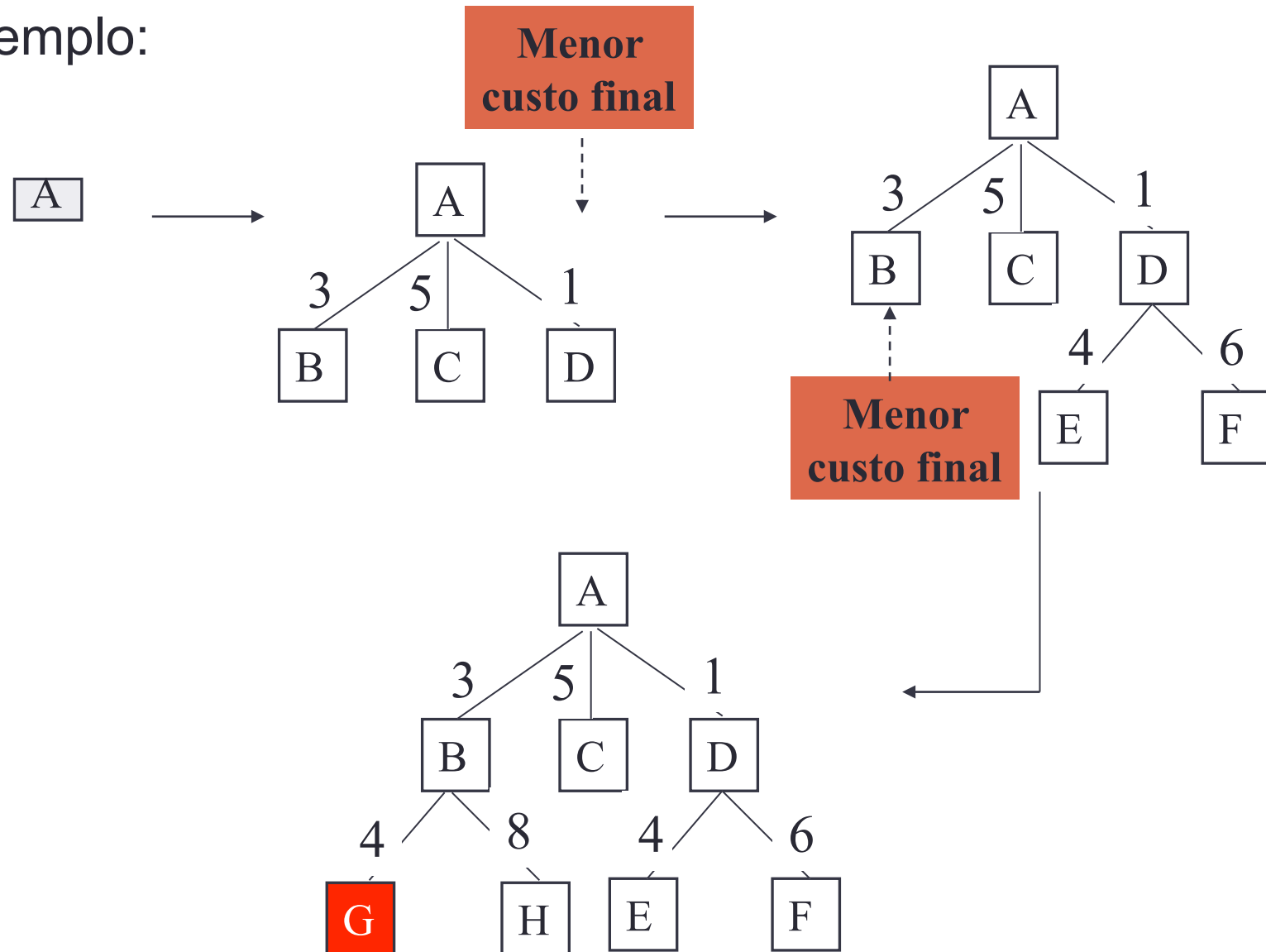


Busca da Melhor Escolha Primeiro (Best First)

- Busca genérica onde o nó de *menor custo estimado* na fronteira do espaço de estados é expandido primeiro.
- Todos os estados abertos até determinado instante são candidatos à expansão.
- Os nós abertos são ordenados em ordem crescente da função de custo $f(n)=h(n)$, de forma que sempre o caminho de menor custo seja expandido.
- Tenta evitar o Contrafortes da busca Hill Climbing.

Busca Best First

Exemplo:



Busca Best First

- Observação importante:
 - $f(n)$: qualquer função de avaliação que indique quão desejável é um nó
 - Tanto $g(n)$ como $h(n)$ podem ser ou integrar $f(n)$
- Assim, tanto algoritmos de busca cega como de busca heurística podem ser vistos como subcasos da BME!!!
- Contrastando $g(n)$ com $h(n)$
 - $g(n)$: função de avaliação de um nó que considera somente o custo de caminho da raiz até ele
 - Adotada pela estratégia BCU para escolha do próximo nó → nesse caso: $f(n) = g(n)$
 - A Busca em Largura (BL) pode ser vista como equivalente à BCU (quando os custos de passo são iguais) → BL é também subcaso da BME!

Busca Best First

- Contrastando $g(n)$ com $h(n)$ (cont.)
 - $h(n)$: função de avaliação de um nó que leva em consideração a estimativa do custo de caminho mais econômico desde o estado representado por esse nó até o objetivo mais próximo
 - Como apenas provê uma **estimativa**, **não** há garantia de que o próximo nó a ser testado e expandido seja realmente o melhor nó!
 - No caso do viajante, a próxima cidade escolhida pode na verdade levar a um caminho mais longo!
 - Casos especiais de busca heurística derivados de BME:
 - Busca Melhor Escolha (*Best First*): $f(n) = h(n)$
 - Algoritmo A*: $f(n) = h(n) + g(n)$



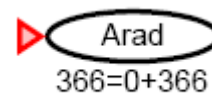
Busca A*

- A ideia do algoritmo é evitar expandir nós com custo total elevado:
 - $g(n)$ = **custo acumulado** da distância de n ao nó inicial.
 - $h(n)$ = **custo estimado** de n ao nó objetivo
 - $f(n)$ é a função de avaliação de **custo total** (custo acumulado + custo estimado)
 - $f(n) = g(n) + h(n)$

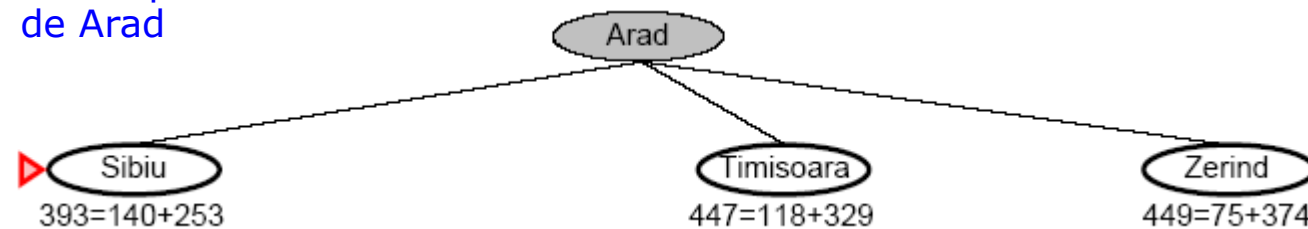
Algoritmo A*

Exemplo: Viajante da Romênia

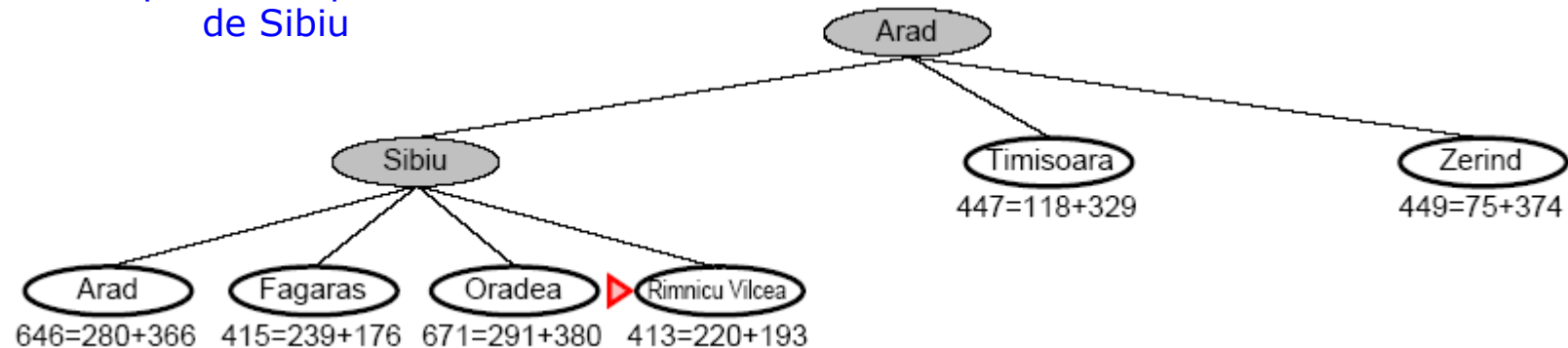
Estado inicial



Depois da expansão
de Arad



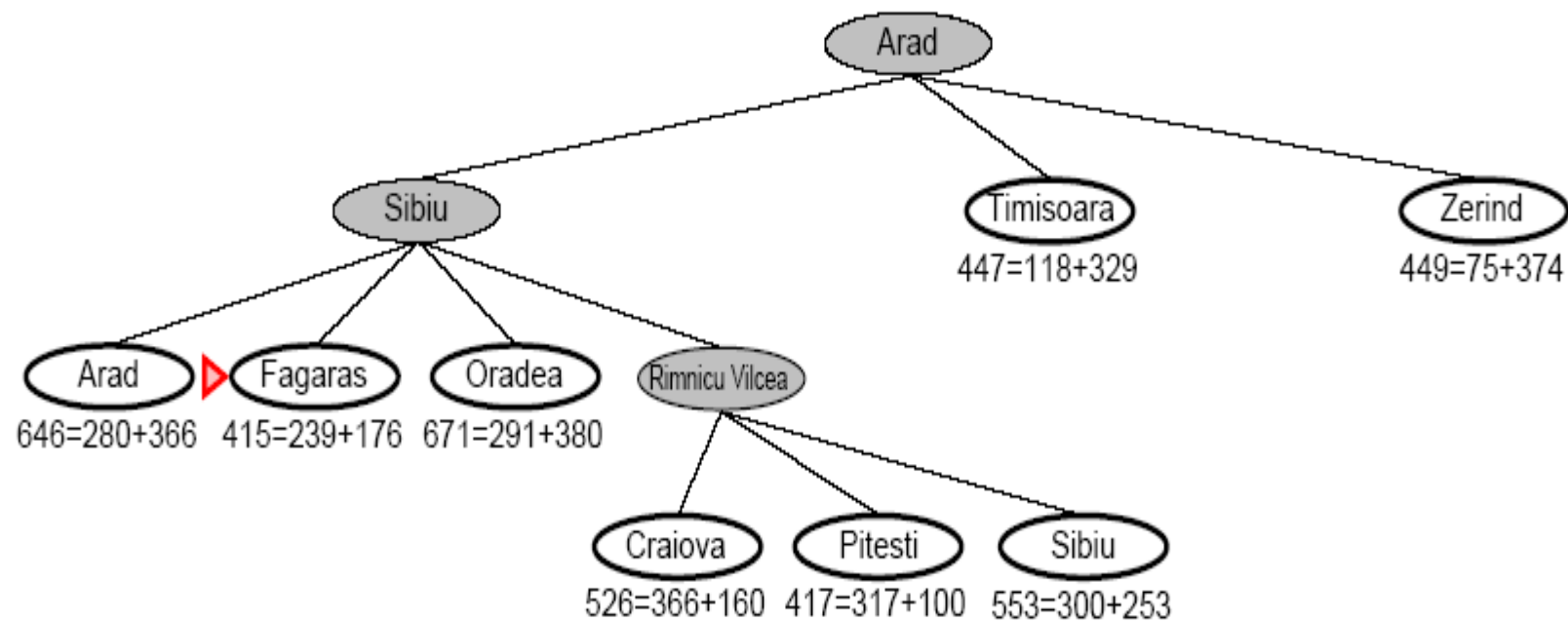
Depois da expansão
de Sibiu



Algoritmo A*

Exemplo: Viajante da Romênia

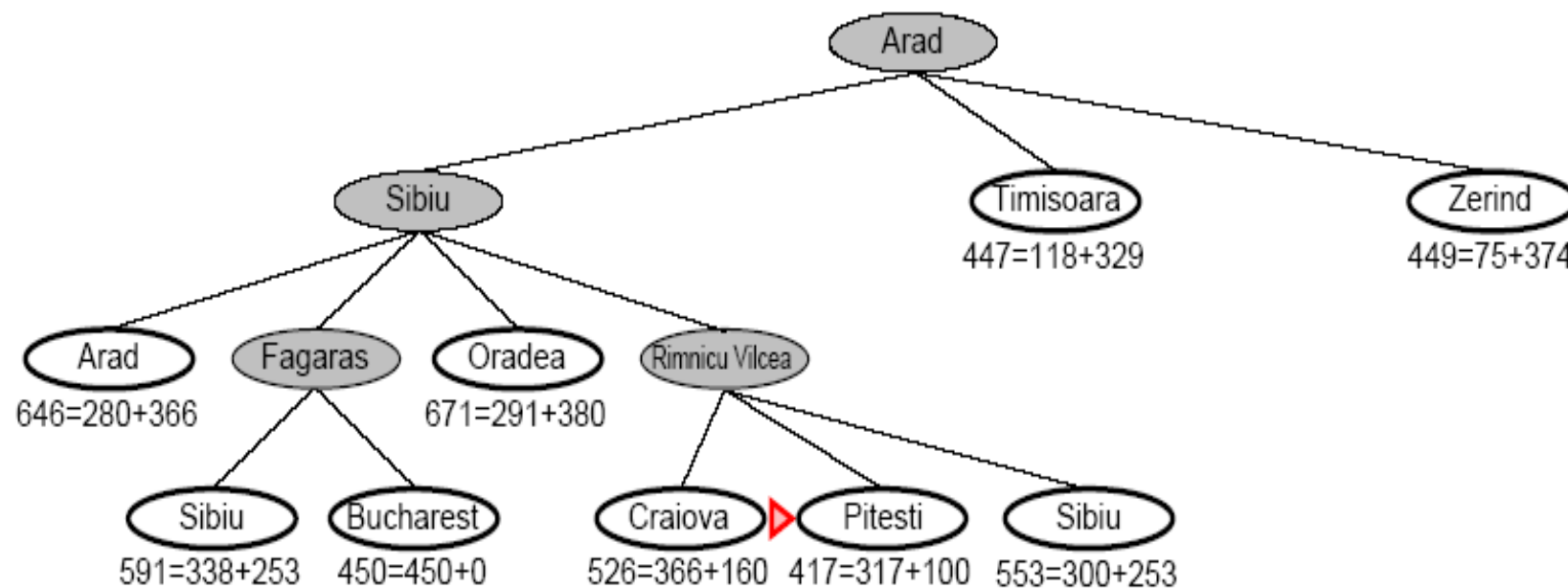
Depois da expansão de Rimnieu Vilcea



Algoritmo A*

Exemplo: Viajante da Romênia

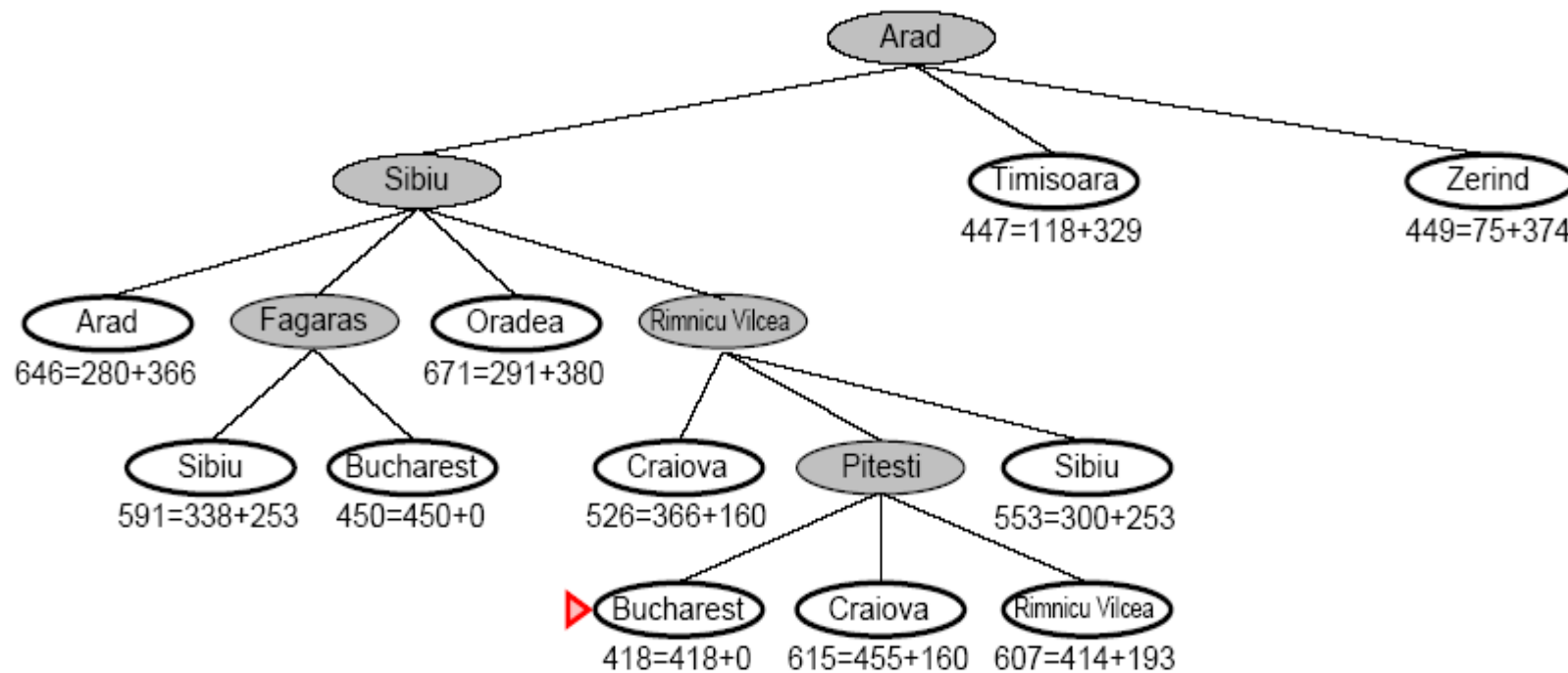
Depois da expansão de Fagaras



Algoritmo A*

Exemplo: Viajante da Romênia

Depois da expansão de Pitesti



Busca On-Line

- Nos algoritmos visto até aqui, a solução é calculada antes do agente atuar no mundo real (busca off-line)
- Algoritmos de busca on-line intercalam ação e computação do próximo passo.
 - 1o) executa uma ação
 - 2o) percebe o ambiente como resultado da ação
 - 3o) calcula próximo passo
- É adequado para:
 - problemas em domínios dinâmicos (onde se perderia tempo calculando todas as ações de uma só vez)
 - problemas em que os estados e as ações são desconhecidos, por exemplo, problemas de exploração.
- Encontre exemplos de situações onde é aplicável algoritmos de busca on-line.

Exercício 1)

- Forneça o estado inicial, o teste objetivo, a função sucessor e a função de custo para cada um dos itens a seguir. Esta definição do problema deve ser precisa o bastante para ser implementada.
 - Um macaco com 1m de altura está em uma sala em que algumas bananas estão suspensas no teto a 2,5m de altura. A sala contém duas caixas empilháveis, móveis e escaláveis com 1m de altura cada. Problema: o macaco quer alcançar as bananas.
 - Têm-se três jarros, com capacidade para 12, 8 e 3 litros e uma torneira de água. É possível encher os jarros e esvaziá-los derramando a água no chão ou em outro jarro. Problema: Você precisa medir exatamente 1 litro.

Exercício 2)

- Determine uma heurística para o problema 8-Puzzle e verifique se é admissível.

5	4	
6	1	8
7	3	2

Start State

1	2	3
8		4
7	6	5

Goal State