

Inteligência artificial

Busca Informada

Adaptação dos slides de **Stuart Russel** e **Peter Norvig**, disponíveis em
aima.cs.berkeley.edu.

Apresentação

- Estratégias de Busca Informada
- Função Heurística

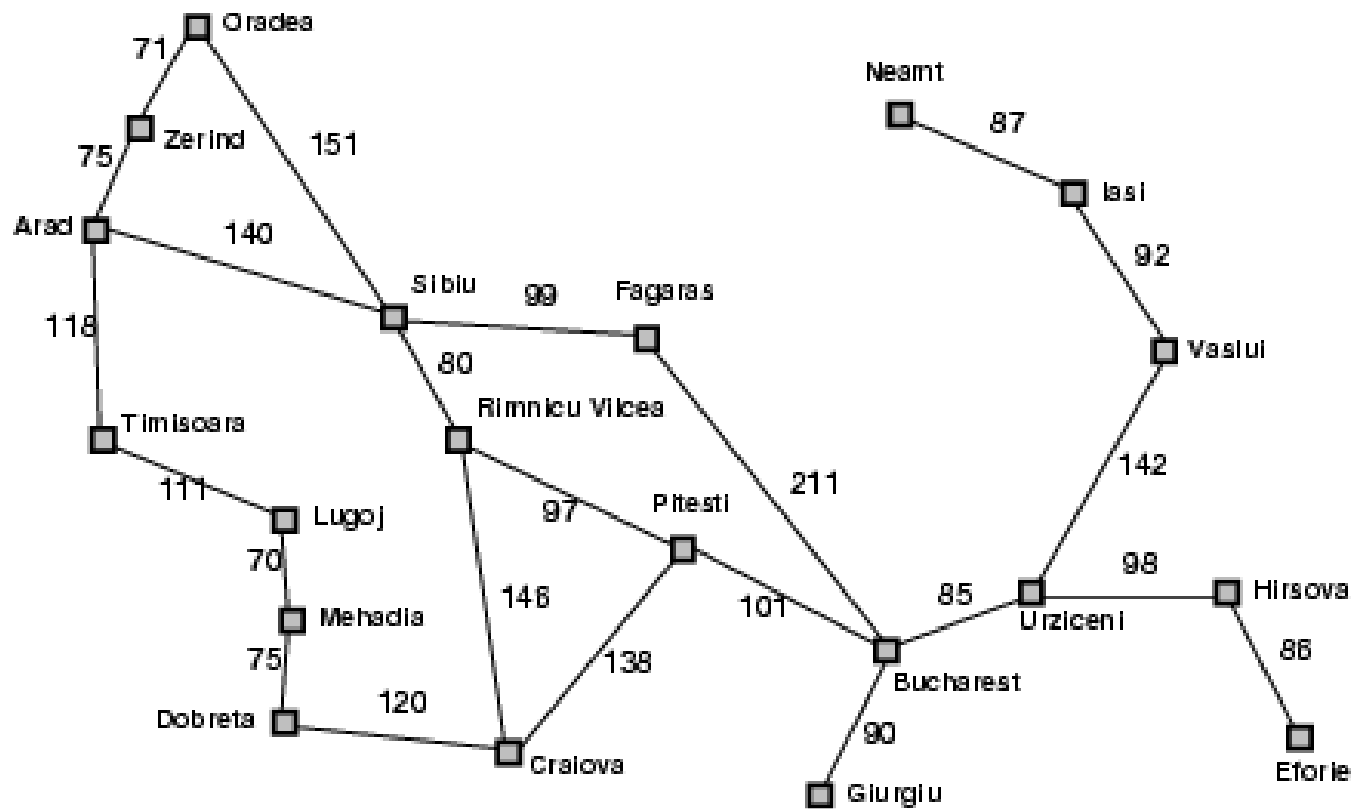
Busca Best First

- Abordagem Geral de Busca Informada:
 - “Informada” = Baseada em conhecimento de domínio.
 - Busca *Best-First*: nodo é selecionado para expansão baseado em uma *função de avaliação* $f(n)$
- Idéia: Estimar a distância até a meta.
 - Escolhe o nodo que *aparenta* ser o melhor - segundo $f(n)$.
- Casos especiais
 - Busca Gulosa
 - Busca A^*

Função Heurística

- ❑ $h(n)$ = custo estimado do caminho mais curto do nodo n até o nodo meta.
- ❑ Condição Básica: Se n é a meta então $h(n) = 0$.

Exemplo: Mapa da Romênia



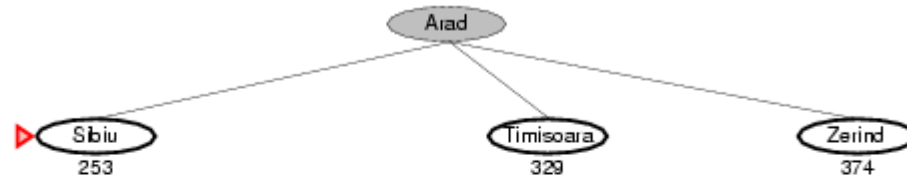
Busca Gulosa

- Função de avaliação $f(n) = h(n)$ (heurística) = estimativa do custo de n até o objetivo
- Ex., $h_{DLR}(n)$ = Distância em linha reta de n até Bucharest
- Expande o nó que “parece” ser o mais próximo ao objetivo

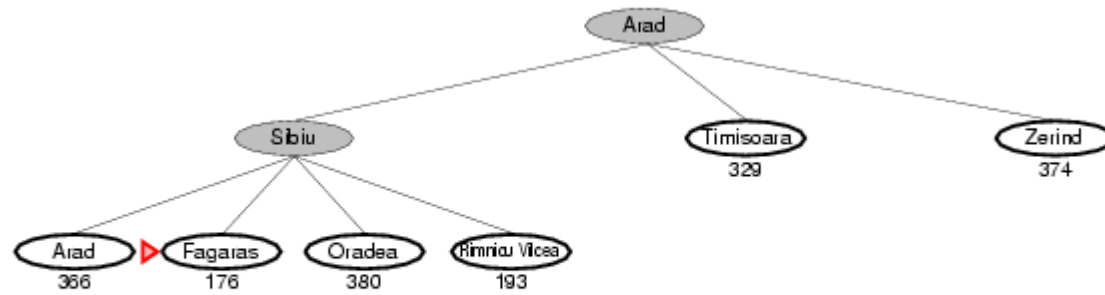
Busca Gulosa



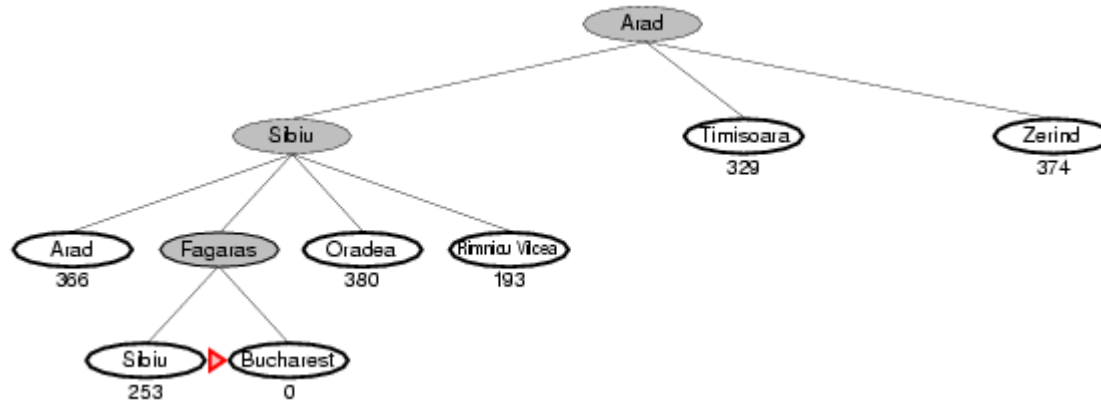
Busca Gulosa



Busca Gulosa



Busca Gulosa

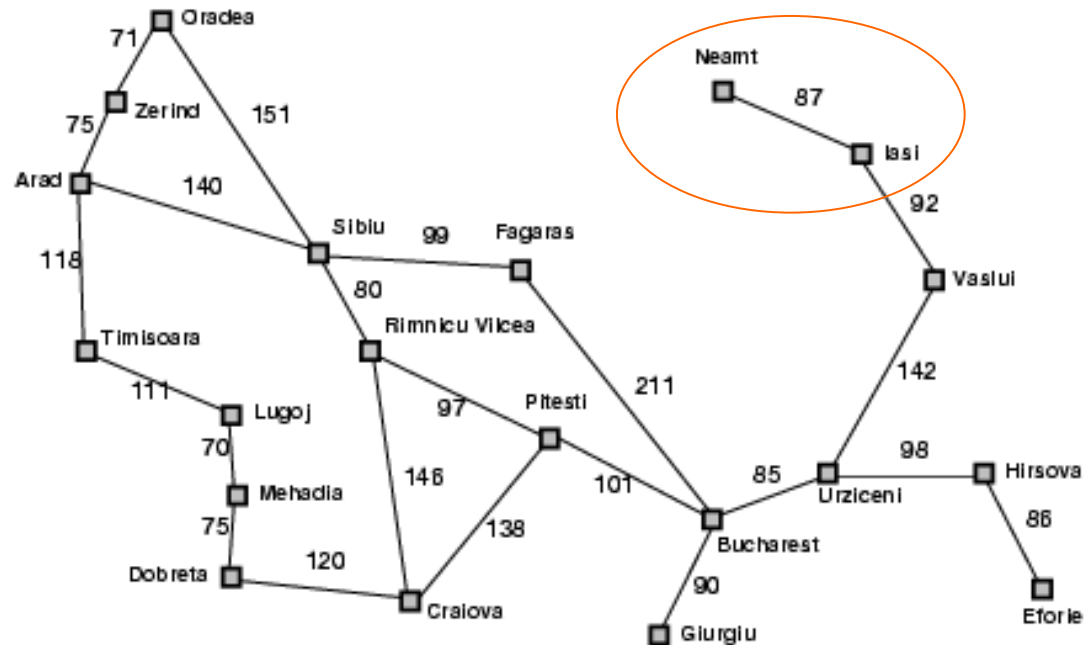


- Meta Alcançada
 - Porém não ótima (veja Arad, Sibiu, Rimnicu Vilcea, Pitesti).

Propriedades da Busca Gulosa

(Problema de ir de Iasi a Fagaras)

- Completa? Não – pode ficar preso em *loops*. Por exemplo, Iasi \rightarrow Neamt \rightarrow Iasi \rightarrow Neamt \rightarrow



Propriedades da Busca Gulosa

- Complexidade (tempo)? $O(b^m)$, mas uma boa heurística pode reduzir muito a quantidade de nós gerados
- Complexidade(espço)? $O(b^m)$ mantém todos os nós na memória
- Ótima? Não

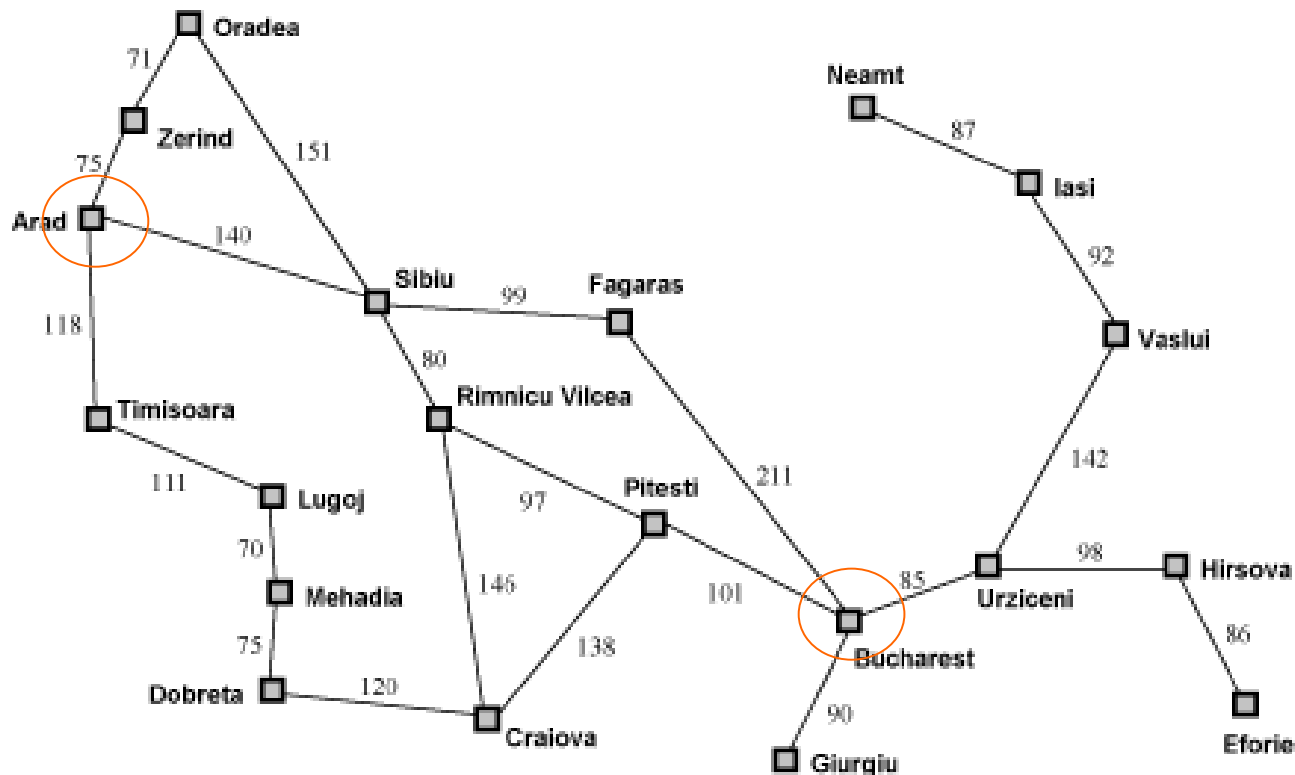
Busca A^*

- Ideia: considerar também o custo de alcançar cada nodo candidato n a partir do nodo (estado) inicial.
- Função de Avaliação $f(n) = g(n) + h(n)$
 - $g(n)$ Custo (já conhecido) de alcançar o nodo n .
 - $h(n)$ Custo estimado do nodo n até o nodo meta.
 - $f(n)$ Custo estimado total do caminho através de n até a meta.
- OBS.
 - $g(n) = 0$ Busca Gulosa
 - $h(n) = 0$ Busca Não Informada de Custo Uniforme

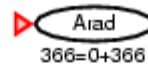
Busca A^*

- Busca A^* utiliza uma heurística **admissível**
 - Uma heurística é admissível se *nunca superestima* o custo para alcançar a meta.
 - Ou seja, heurísticas admissíveis são otimistas.
- Formalmente:
 - 1. $h(n) \leq b^*(n)$ em que $b^*(n)$ é o custo verdadeiro de n até a meta.
 - 2. Dado que $h(n) \geq 0$, tem-se $h(G) = 0$ para qualquer meta G .
- Ex. $h_{DLR}(n)$ nunca superestima a distância real via estrada.

Busca A*



Busca A^*



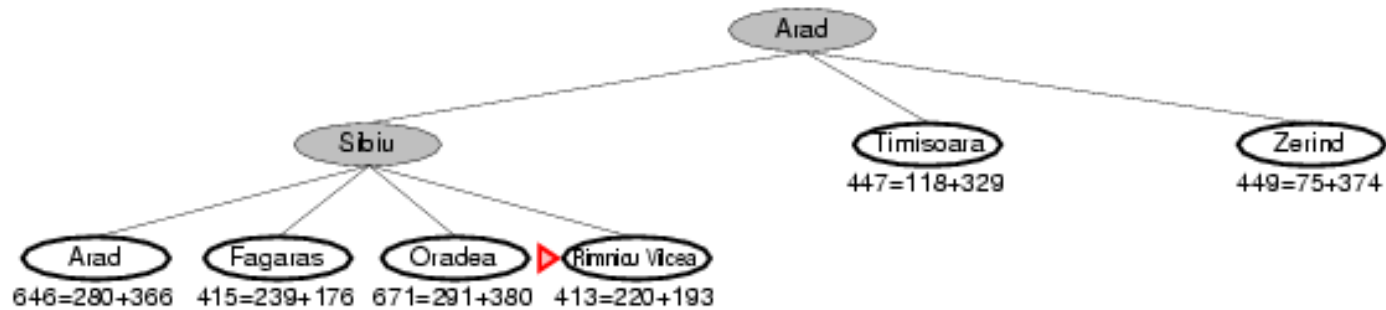
- Encontrar Bucarest partindo de Arad:
 - $f(\text{Arad}) = 0 + h(\text{Arad}) = 0 + 366 = 366$

Busca A*



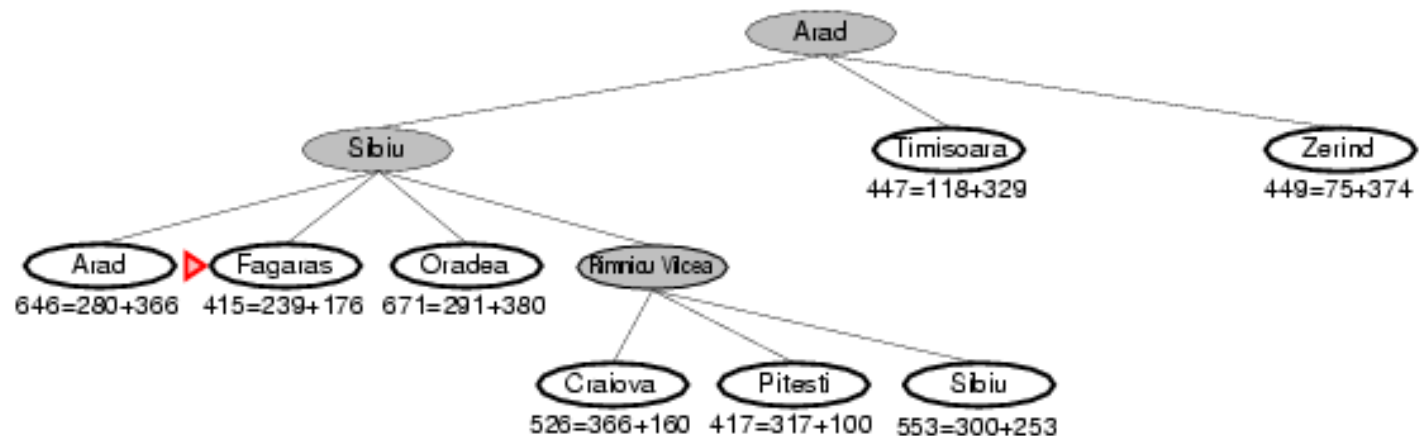
- Expandindo Arad e determinando $f(n)$ para cada nodo:
 - $f(\text{Sibiu}) = c(\text{Arad}, \text{Sibiu}) + h(\text{Sibiu}) = 140 + 253 = 393$.
 - $f(\text{Timisoara}) = c(\text{Arad}, \text{Timisoara}) + h(\text{Timisoara}) = 118 + 329 = 447$.
 - $f(\text{Zerind}) = c(\text{Arad}, \text{Zerind}) + h(\text{Zerind}) = 75 + 374 = 449$.
- Melhor escolha é Sibiu.

Busca A*



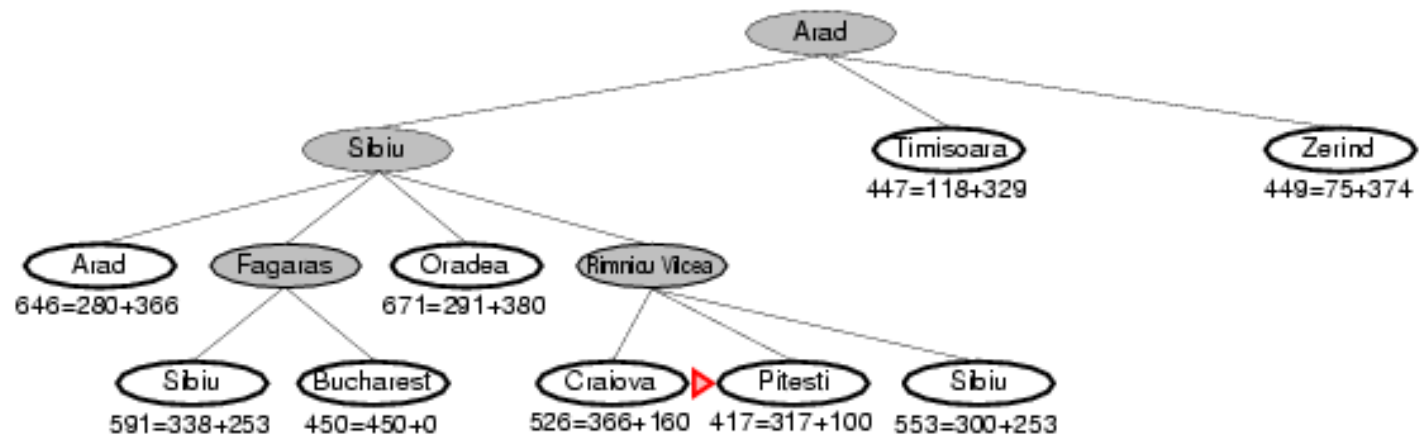
- Expandindo Sibiu e determinando $f(n)$ para cada nodo:
 - $f(\text{Arad}) = g(\text{Arad}) + h(\text{Arad}) = 280 + 366 = 646.$
 - $f(\text{Fagaras}) = g(\text{Fagaras}) + h(\text{Fagaras}) = 239 + 176 = 415.$
 - $f(\text{Oradea}) = g(\text{Oradea}) + h(\text{Oradea}) = 291 + 380 = 671.$
 - $f(\text{Rimnicu Vilcea}) = g(\text{Rimnicu Vilcea}) + h(\text{Rimnicu Vilcea}) = 220 + 193 = 413.$
- Melhor escolha é Rimnicu Vilcea.

Busca A*



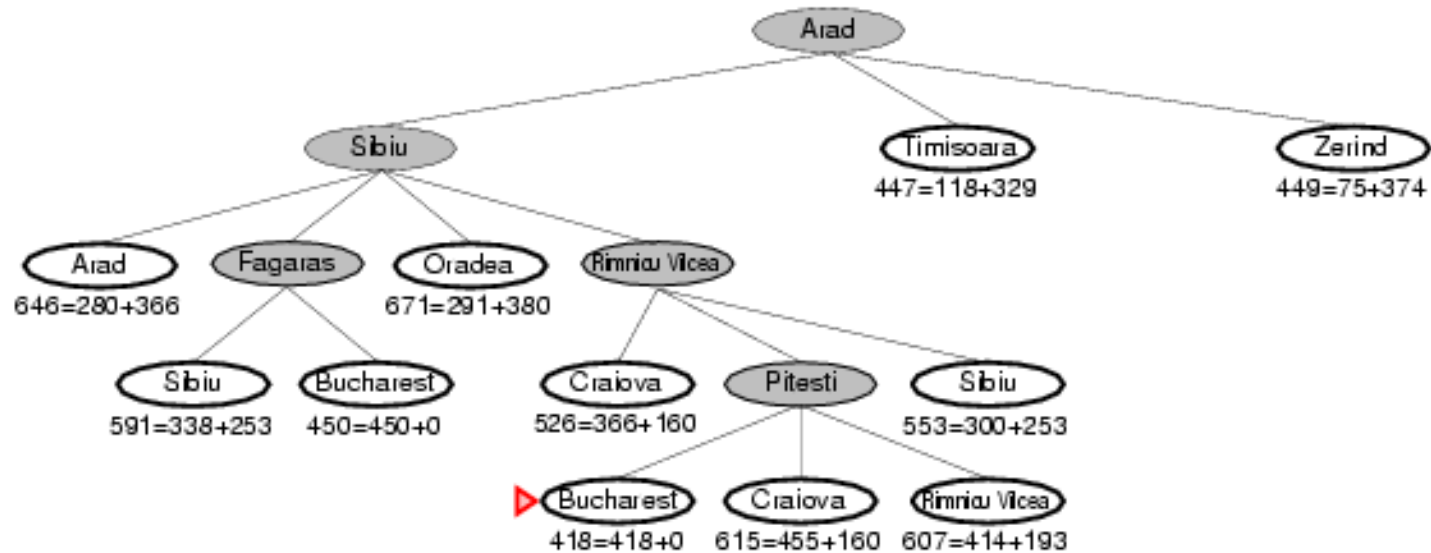
- Expandindo Rimnicu Vilcea:
 - $f(\text{Craiova}) = g(\text{Craiova}) + h(\text{Craiova}) = 366 + 160 = 526.$
 - $f(\text{Pitesti}) = g(\text{Pitesti}) + h(\text{Pitesti}) = 317 + 100 = 417.$
 - $f(\text{Sibiu}) = g(\text{Sibiu}) + h(\text{Sibiu}) = 300 + 253 = 553.$
- Melhor escolha parece ser Fagaras.

Busca A*



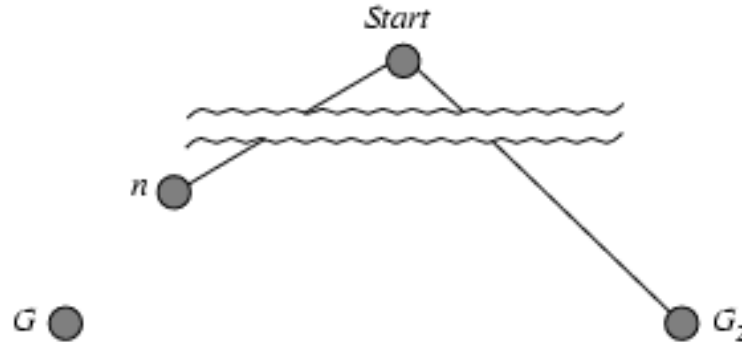
- Expandindo Fagaras:
 - $f(\text{Sibiu}) = g(\text{Sibiu}) + h(\text{Sibiu}) = 338 + 253 = 591$.
 - $f(\text{Bucareste}) = g(\text{Bucareste}) + h(\text{Bucareste}) = 450 + 0 = 450$.
- Melhor escolha é Pitesti.

Busca A*



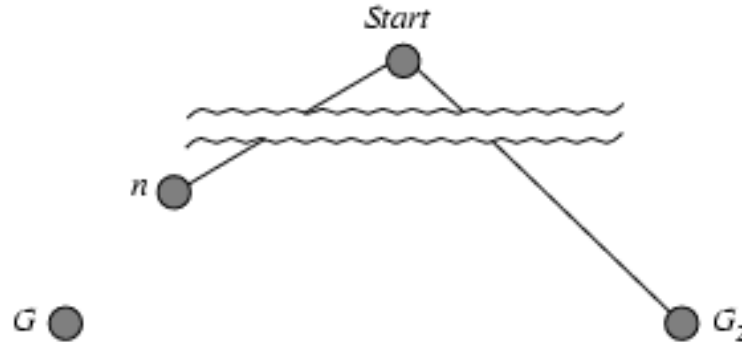
- Expandindo Pitesti:
 - $f(\text{Bucareste}) = g(\text{Bucareste}) + h(\text{Bucareste}) = 418 + 0 = 418$.
 - Melhor escolha é Bucareste.
- Solução ótima (dado que $h(n)$ é **admissível**).

Prova que A^* é ótimo (para busca em árvore)



- Suponha um nó objetivo não ótimo G_2 apareça na borda e seja C^* o custo da solução ótima.
- Como G_2 não é ótimo e $h(G_2)=0$, então
$$f(G_2)=g(G_2)+h(G_2)>C^*$$

Prova que A^* é ótimo (para busca em árvore)



- Considere um nó na borda n que está no caminho da solução ótima.
- Se $h(n)$ for admissível, então, $f(n)=g(n)+h(n)\leq C^*$
- Desse modo, $f(n)\leq C^*<f(G_2)$, e assim G_2 não será expandido e A^* retorna uma solução ótima

Propriedades da Busca A^*

- Completa? Sim (a menos que haja um número infinito de nós com $f \leq f(G)$)
- Complexidade(tempo)? exponencial
- Complexidade(espço)? Guarda todos os nós na memória
- Ótima? Sim

Heurísticas Admissíveis

Ex., para o 8-puzzle:

- $h_1(n)$ = número de blocos deslocados
- $h_2(n)$ = distância total Manhattan

7	2	4
5		6
8	3	1

Start State

	1	2
3	4	5
6	7	8

Goal State

- $h_1(S) = ?$
- $h_2(S) = ?$

Heurísticas Admissíveis

Ex., para o 8-puzzle:

- $h_1(n)$ = número de blocos em posições erradas
- $h_2(n)$ = distância total Manhattan

7	2	4
5		6
8	3	1

Start State

	1	2
3	4	5
6	7	8

Goal State

- $h_1(S) = 8$
- $h_2(S) = 3+1+2+2+2+3+3+2 = 18$

Dominância

- Se $h_2(n) \geq h_1(n)$ para todo n (ambos admissíveis) então h_2 **domina** h_1
- h_2 é melhor para a busca
- Custo de busca típicos (número médio de nós expandidos):

$d=12$ IDS = 3,644,035 nós

$A^*(h_1) = 227$ nós

$A^*(h_2) = 73$ nós

$d=24$ IDS = um número muito grande

$A^*(h_1) = 39,135$ nós

$A^*(h_2) = 1,641$ nós

Problema relaxado

- Um problema com poucas restrições sobre as ações é chamado **problema relaxado**
- O custo de uma solução ótima para um problema relaxado é uma heurística admissível para o problema original
- Se as regras do 8-puzzle são relaxadas de tal modo que um bloco pode se mover **para qualquer lugar**, então $h_1(n)$ dá a *solução mais curta*
- Se as regras são relaxadas de modo que um bloco pode se mover **para qualquer quadro adjacente**, então $h_2(n)$ dá a *solução mais curta*

Referências

- Stuart Russel e Peter Norvig, **Inteligência Artificial**, 2ª edição, Editora Campus, 2004.