Algoritmos de pesquisa informada

Capítulo 4, Secções 1–2

Sumário

◊ Pesquisa o melhor primeiro (Best-first search)

◊ Pesquisa A* (A* search)

◊ Heurísticas

Pesquisa em árvores

function TREE-SEARCH(*problem*, *fringe*) **returns** a solution, or failure
  fringe ← INSERT(MAKE-NODE(INITIAL-STATE[*problem*]), *fringe*)
  loop do
    if *fringe* is empty **then return** failure
    node ← REMOVE-FRONT(*fringe*)
    if GOAL-TEST[*problem*] applied to STATE(*node*) succeeds **return** *node*
    fringe ← INSERTALL(EXPAND(*node*, *problem*), *fringe*)

Define-se uma estratégia escolhendo a **ordem de expansão dos nós**

Pesquisa o melhor primeiro (Best-first search)

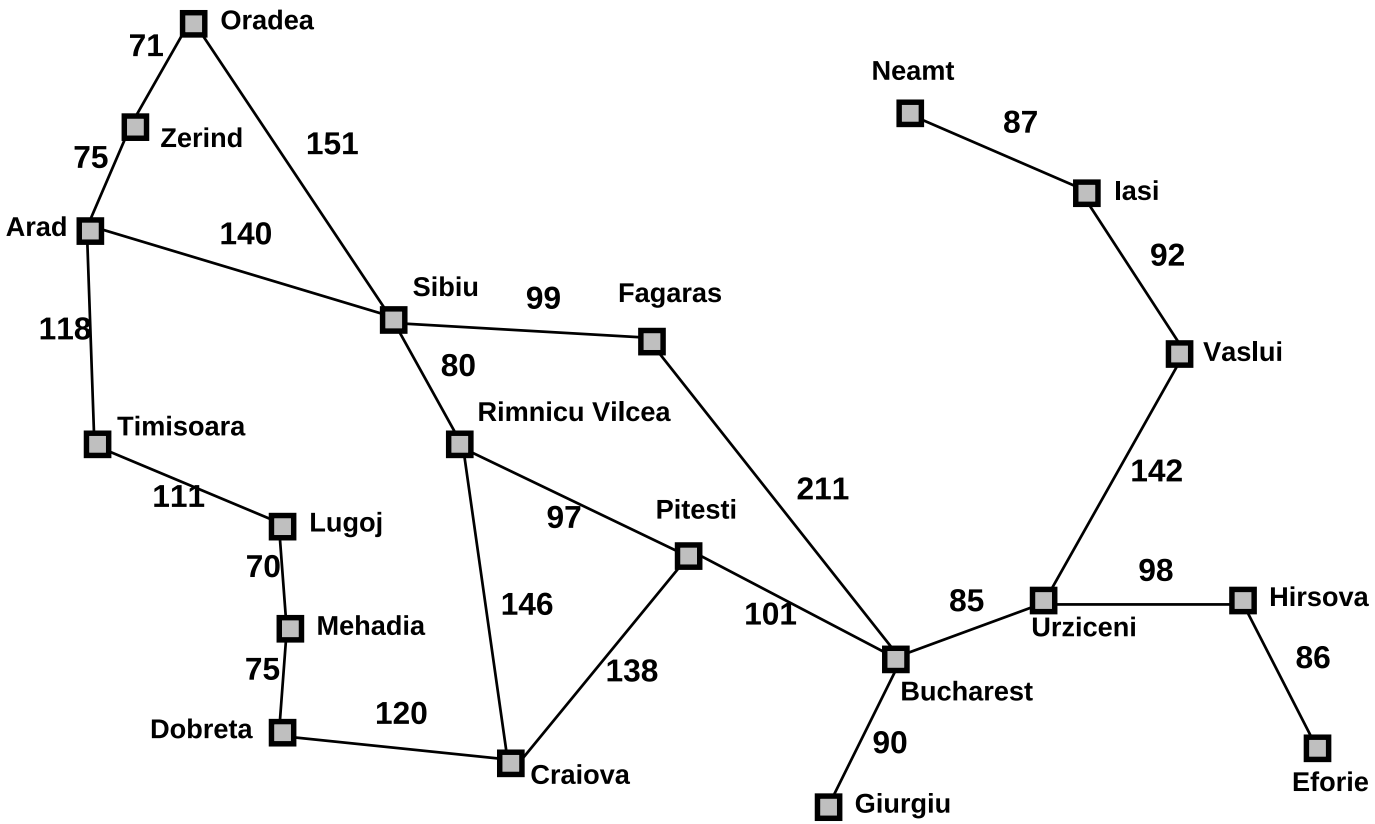
Ideia: usar um função de avaliação para cada nó
– estimando a "adequação" ("desirability")

$\Rightarrow$ Expandir o nó mais adequado

Implementação:
fringe é uma fila ordenada por ordem decrescente de adequação dos nós

Casos especiais:
pesquisa ansiosa ou gananciosa (greedy search)
pesquisa A* (A^* search)

Roménia com custo em km



Straight−line distance
to Bucharest

Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Pesquisa ansiosa — Greedy search

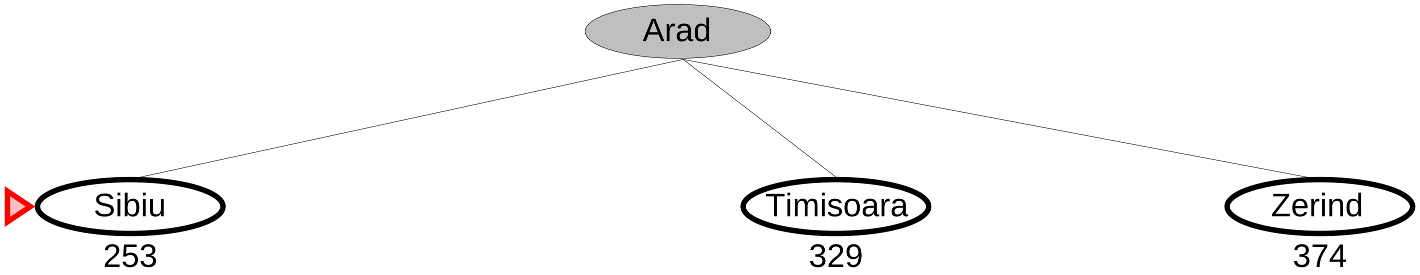
Função de avaliação $h(n)$ (**h**eurística)
$= $ estimativa do custo de n ao objectivo mais próximo

E.g., $h_{\mathrm{SLD}}(n) = $ distância em linha recta de n a Bucharest

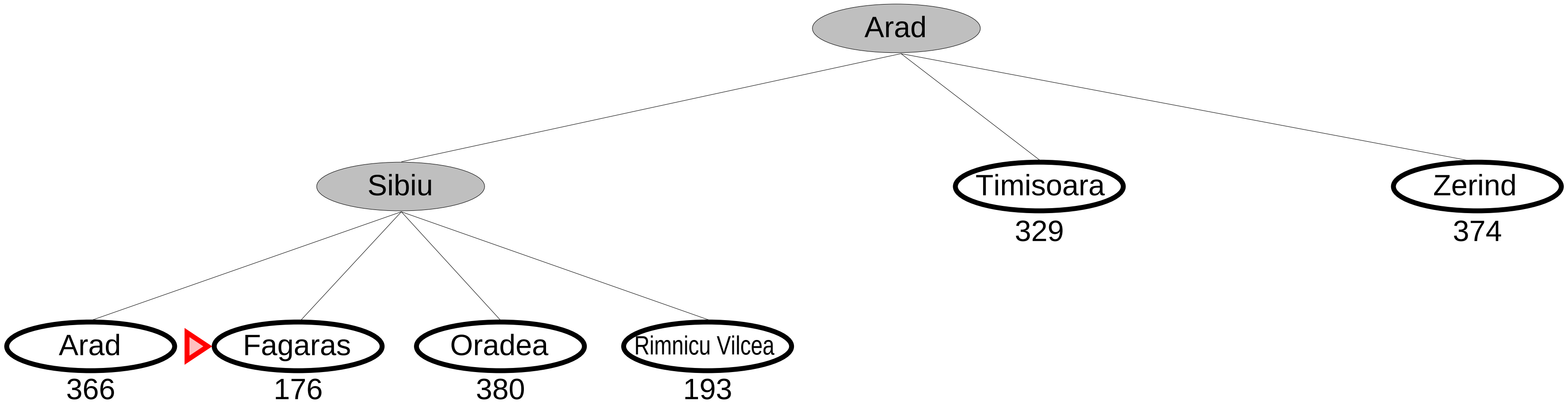
A pesquisa ansiosa expande o nó que **aparenta** estar mais próximo do objectivo

Exemplo da pesquisa ansiosa

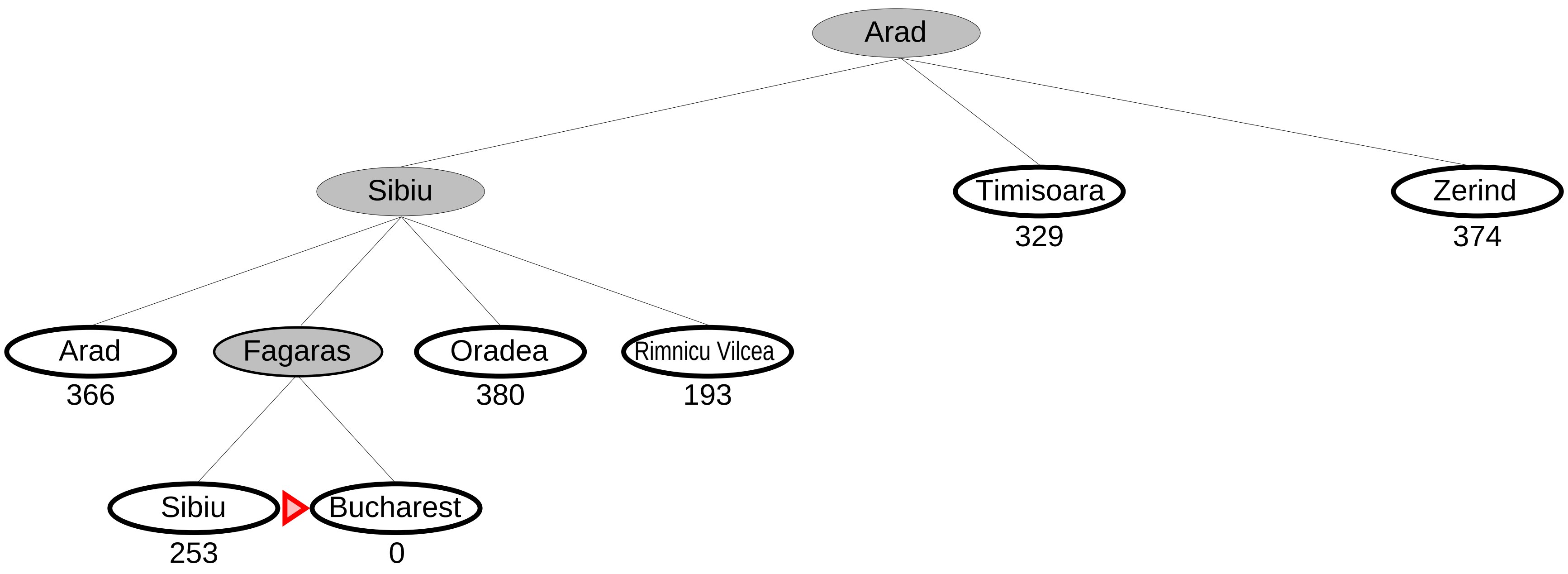
Exemplo da pesquisa ansiosa



Exemplo da pesquisa ansiosa



Exemplo da pesquisa ansiosa



Propriedades da pesquisa ansiosa

Completo??

Propriedades da pesquisa ansiosa

Completo?? Não– pode ficar preso em ciclos, e.g., com Oradea como objectivo,

Iasi $\rightarrow$ Neamt $\rightarrow$ Iasi $\rightarrow$ Neamt $\rightarrow$

Completo em espaços finitos com verificação de repetição de estados

Tempo??

Propriedades da pesquisa ansiosa

Completo?? Não– pode ficar preso em ciclos, e.g., com Oradea como objectivo,

Iasi $\rightarrow$ Neamt $\rightarrow$ Iasi $\rightarrow$ Neamt $\rightarrow$

Completo em espaços finitos com verificação de repetição de estados

Tempo?? $O(b^m)$, mas uma boa heurística pode melhorar bastante o tempo

Espaço??

Propriedades da pesquisa ansiosa

Completo?? Não– pode ficar preso em ciclos, e.g., com Oradea como objectivo,

Iasi $\rightarrow$ Neamt $\rightarrow$ Iasi $\rightarrow$ Neamt $\rightarrow$

Completo em espaços finitos com verificação de repetição de estados

Tempo?? $O(b^m)$, mas uma boa heurística pode melhorar bastante o tempo

Espaço?? $O(b^m)$— guarda todos os nós em memória

Óptima??

Propriedades da pesquisa ansiosa

Completo?? Não– pode ficar preso em ciclos, e.g., com Oradea como objectivo,

Iasi $\rightarrow$ Neamt $\rightarrow$ Iasi $\rightarrow$ Neamt $\rightarrow$

Completo em espaços finitos com verificação de repetição de estados

Tempo?? $O(b^m)$, mas uma boa heurística pode melhorar bastante o tempo

Espaço?? $O(b^m)$— guarda todos os nós em memória

Óptima?? Não

Pesquisa A*

Ideia: evitar expandir os caminhos caros

Função de avaliação $f(n) = g(n) + h(n)$

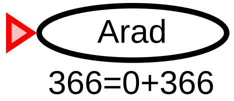
$g(n)$ = custo até atingir n
$h(n)$ = custo estimado para atingir o objectivo a partir de n
$f(n)$ = custo total estimado do caminho que passa por n até ao objectivo

A pesquisa A* usa uma heurística admissível
i.e., $h(n) \leq h^*(n)$ onde $h^*(n)$ é o **verdadeiro** custo de n.
(Também é necessário que $h(n) \geq 0$, com $h(G) = 0$ para qualquer objectivo G.)

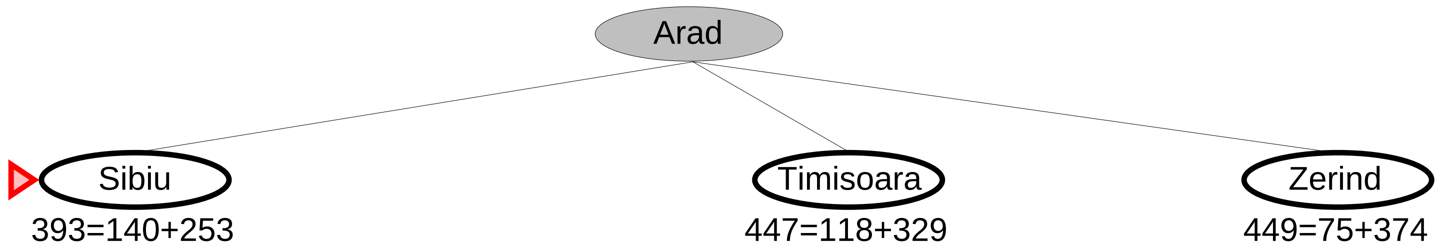
E.g., $h_{\mathrm{SLD}}(n)$ nunca sobre-estima a distância da estrada.

Teorema: a pesquisa A* é óptima.

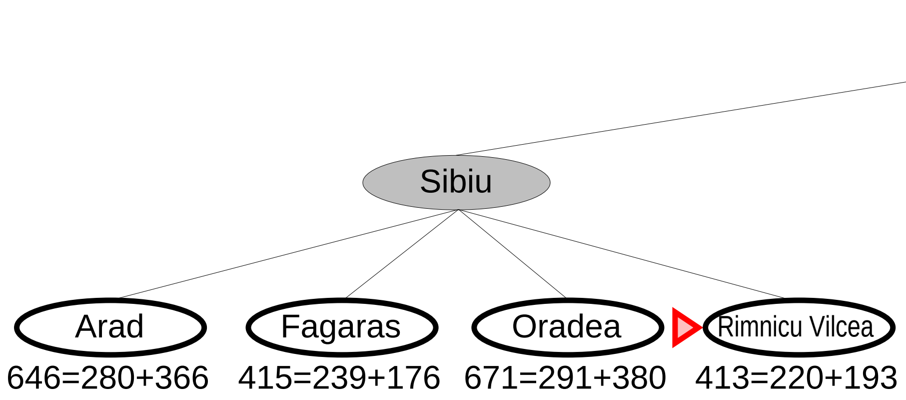
Exemplo da pesquisa A*



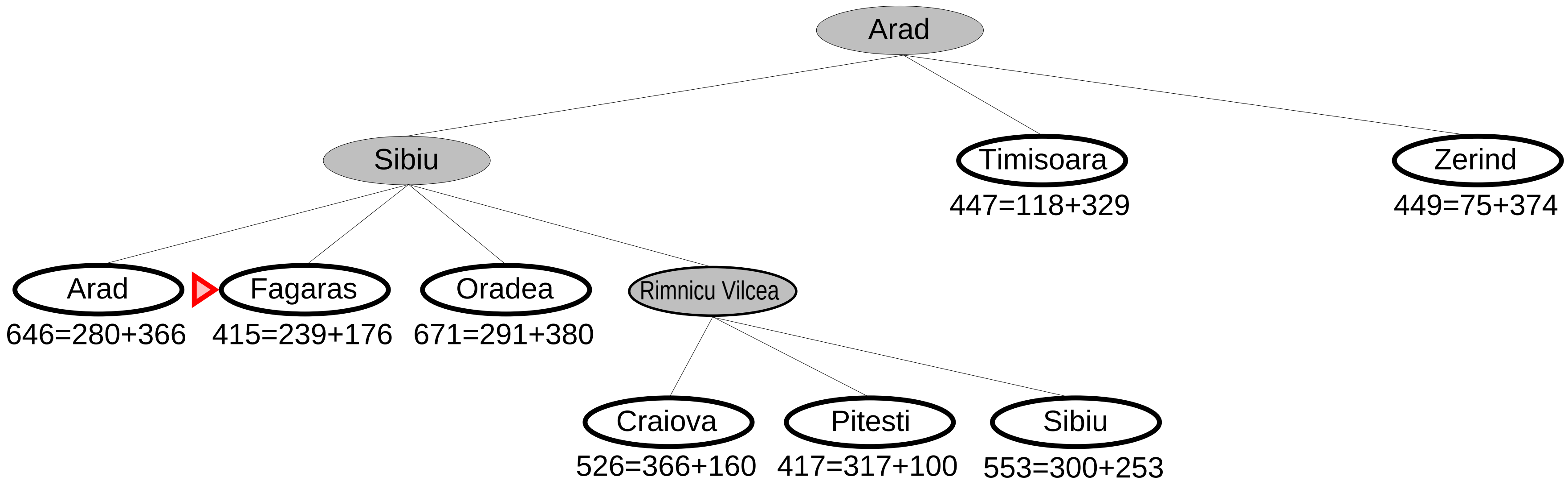
Exemplo da pesquisa A*



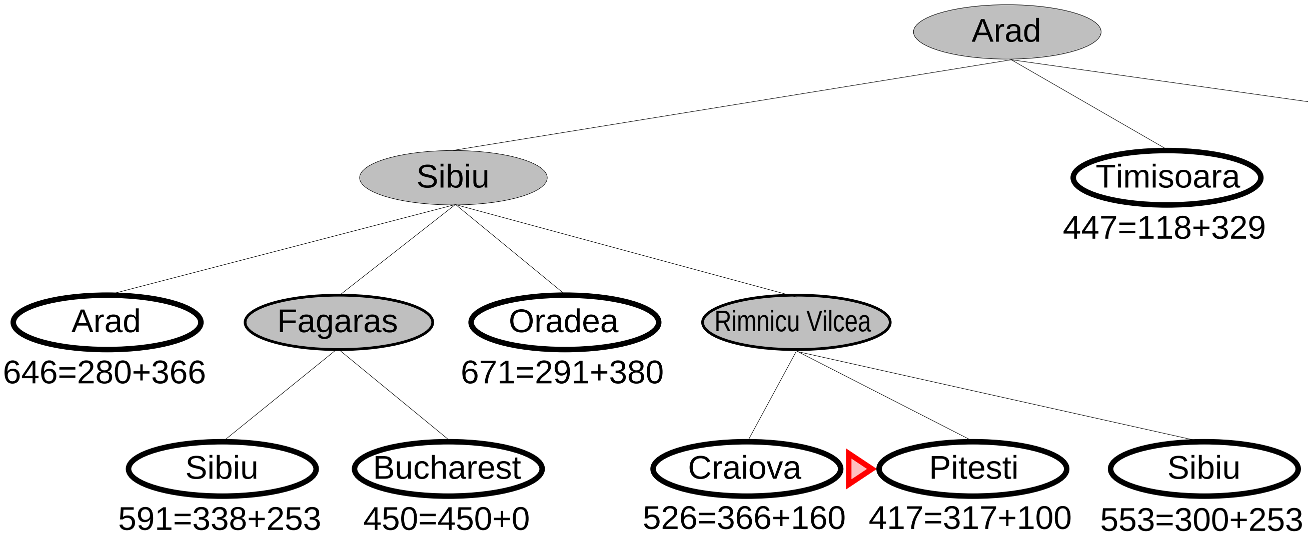
Exemplo da pesquisa A*



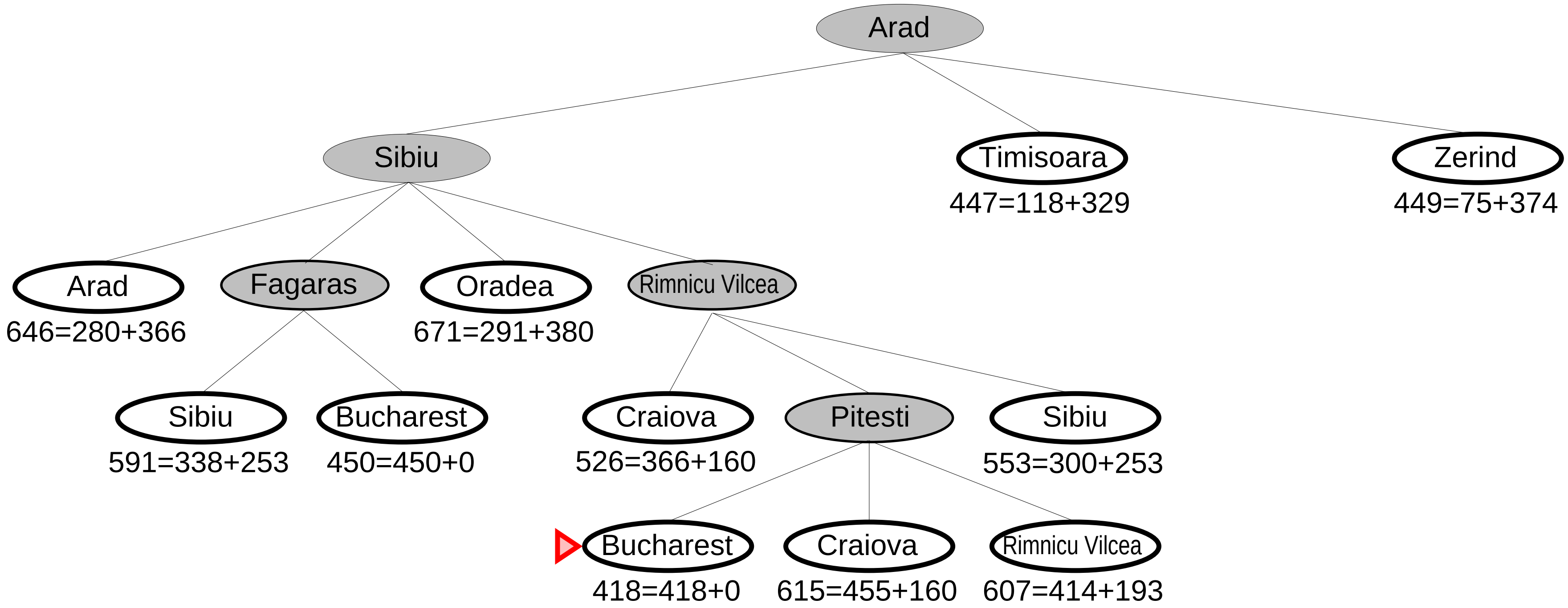
Exemplo da pesquisa A*



Exemplo da pesquisa A*

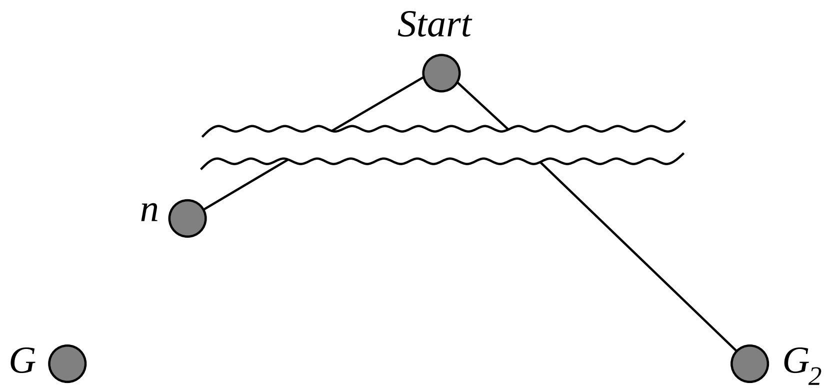


Exemplo da pesquisa A*



Optimalidade do A* (prova)

Suponha que algum objectivo sub-óptimo G_2 foi gerado e está na fila. Seja n um nó não expandido num caminho mais curto para o objectivo óptimo G_1.



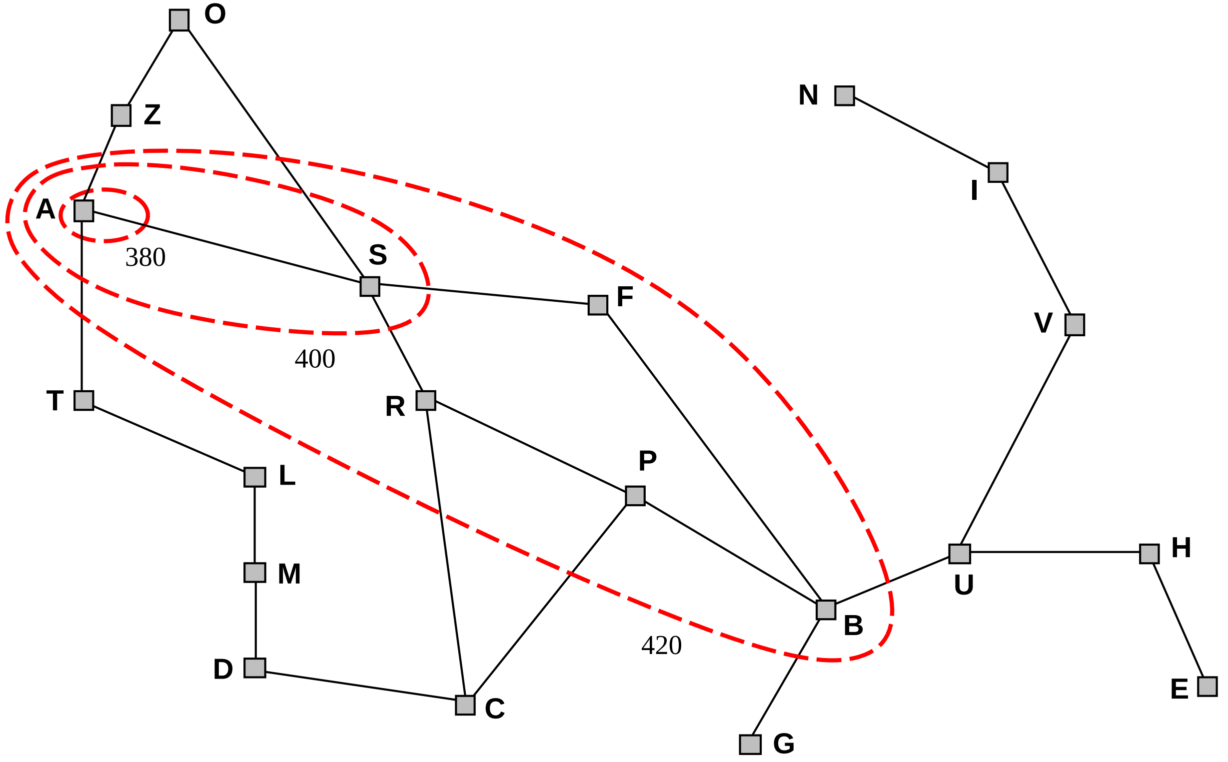
$$
\begin{aligned}
f(G_2) &= g(G_2) && \text{pois } h(G_2) = 0 \\
&> g(G_1) && \text{pois } G_2 \text{ e' subo'ptimo} \\
&\geq f(n) && \text{pois } h \text{ e' admissivel}
\end{aligned}
$$

Como $f(G_2) > f(n)$, A* nunca selecciona G_2 para expandir

Optimalidade do A* (mais útil)

Lema: A* expande os nós por ordem crescente de f valor*

Gradualmente adiciona "f-contornos" de nós
(cf. a pesquisa em largura adiciona camadas)
O contorno i tem todos os nós com $f = f_i$, onde $f_i < f_{i+1}$



Propriedades do A*

Completo??

Propriedades do A*

Completo?? Sim, a não ser infinitos nós com $f \leq f(G)$

Tempo??

Propriedades do A*

Completo?? Sim, a não ser que existam infinitos nós com $f \leq f(G)$

Tempo?? Exponencial Exponencial em [erro relativo h $\times$ comprimento de soln.]

Espaço??

Propriedades do A*

Completo?? Sim, a não ser que existam infinitos nós com $f \leq f(G)$

Tempo?? Exponencial Exponencial em [erro relativo h $\times$ comprimento de soln.]

Espaço?? mantém todos os nós em memória

Óptima??

Propriedades do A*

Completo?? Sim, a não ser que existam infinitos nós com $f \leq f(G)$

Tempo?? Exponencial Exponencial em [erro relativo h $\times$ comprimento de soln.]

Espaço?? mantém todos os nós em memória

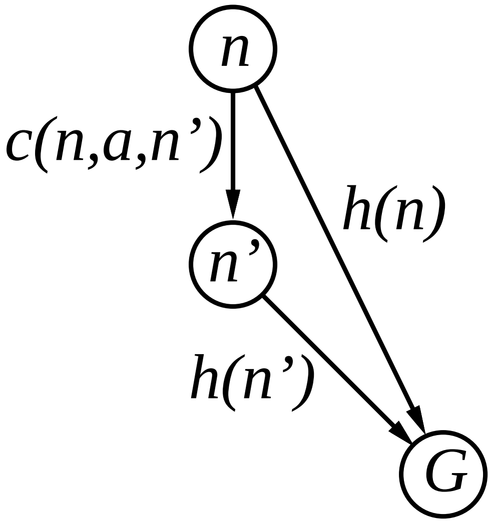
Óptima?? Sim—não expande f_{i+1} até f_i terminar

A* expande todos os nós com $f(n) < C^*$
A* expande alguns nós com $f(n) = C^*$
A* não expande nós com $f(n) > C^*$

Prova do lema: Consistência

Uma heurística é consistente se

$$h(n) \le c(n,a,n') + h(n')$$



Se h é consistente, temos

$$\begin{aligned} f(n') &= g(n') + h(n') \\ &= g(n) + c(n,a,n') + h(n') \\ &\ge g(n) + h(n) \\ &= f(n) \end{aligned}$$

I.e., $f(n)$ é não decrescente ao longo do caminho.

Heurística Admissível

E.g., para o 8-puzzle:

$h_1(n)$ = numero de peças mal colocadas
$h_2(n)$ = total da distância de Manhattan
(i.e., numero de quadrados até ao local certo de cada peça)

7	2	4
5		6
8	3	1

Start State

1	2	3
4	5	6
7	8	

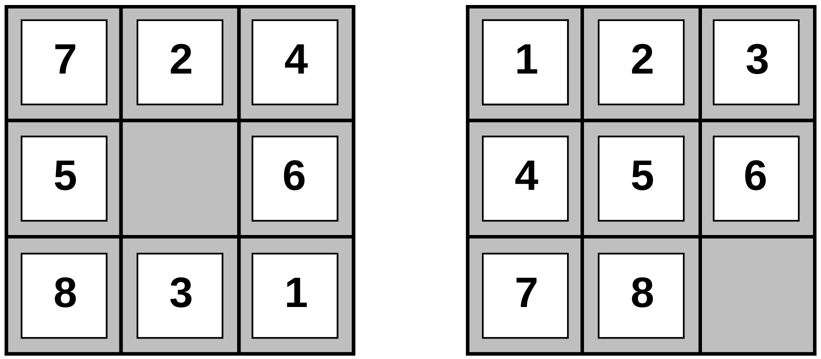
Goal State

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

Heurística Admissível

E.g., para o 8-puzzle:

$h_1(n)$ = numero de peças mal colocadas
$h_2(n)$ = total da distância de Manhattan
(i.e., numero de quadrados até ao local certo de cada peça)



Start State **Goal State**

$h_1(S) =$?? 6
$h_2(S) =$?? 4+0+3+3+1+0+2+1 = 14

Dominância

Se $h_2(n) \geq h_1(n)$ para todo o n (ambas são admissíveis)
então h_2 **domina** h_1 e é melhor para a pesquisa

Custos de pesquisas:

$d = 14$	IDS = 3,473,941 nós
	A$^*(h_1)$ = 539 nós
	A$^*(h_2)$ = 113 nós
$d = 24$	IDS $\approx$ 54,000,000,000 nós
	A$^*(h_1)$ = 39,135 nós
	A$^*(h_2)$ = 1,641 nós

Dadas duas heurísticas admissíveis h_a, h_b,

$$h(n) = \max(h_a(n), h_b(n))$$

também admissível e domina h_a, h_b

Problemas simplificados

Podem-se derivar heurísticas admissíveis do custo de soluções **exactas** de versões **simplificadas** do problema

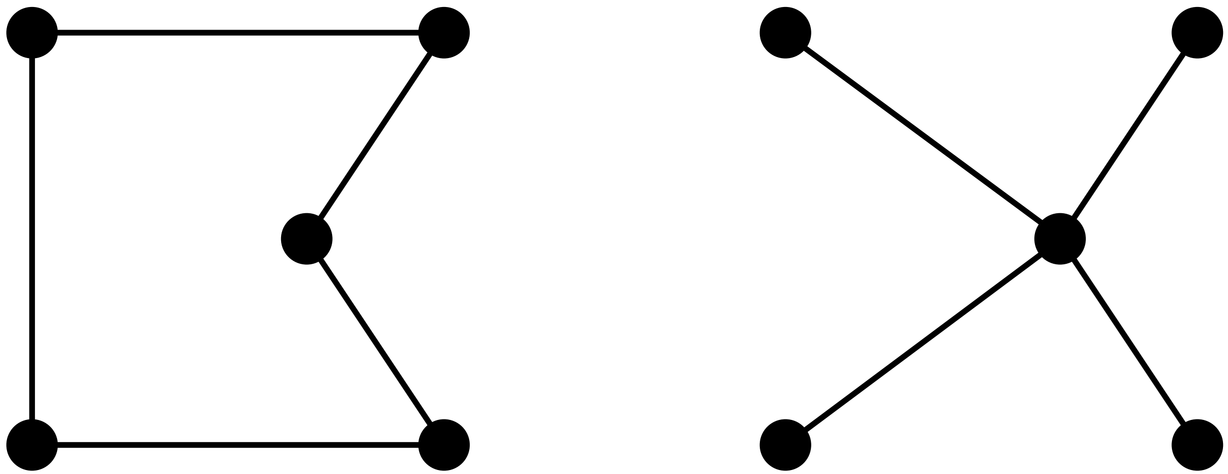
Se as regras do 8-puzzle fossem simplificadas de forma a que uma peça se possa mover para **qualquer lugar**, então $h_1(n)$ é o custo da solução mais curta desta simplificação

Se as regras forem simplificadas de forma a que uma peça se possa mover para **qq casa adjacente**, então $h_2(n)$ é o custo da solução mais curta desta simplificação

Importante: O custo da solução óptima de um problema simplificado não é maior que o custo da solução óptima do problema real.

Problemas simplificados cont.

O problema do caixeiro viajante (travelling salesperson problem, TSP)
Encontrar o caminho mais curto que passa por todas as cidades uma só vez



Árvore mínima que cobre todos os pontos Minimum spanning tree pode ser calculada em $O(n^2)$
e é um limite inferior da menor volta (aberta)

Resumo

As funções heurísticas estimam o custo dos caminhos mais curtos

Uma boa heurística pode reduzir bastante o custo da pesquisa.

O algoritmo Ansioso melhor primeiro (Greedy best-first search) expande o menor h
– é incompleto e nem sempre óptimo

A pesquisa A* expande o menor $g + h$
– é completo e óptimo

Heurísticas admissíveis podem ser obtidas a partir de soluções exactas para o problema simplificado