

minimum minimum

Prof. Felipe Andery Reis

<u>fandery@inatel.br</u>

Material adaptado do prof. Edielson

mar/2018



Introdução

Agenda

Parte 2

- Estratégias de busca heurística:
 - Busca gulosa
 - Busca A*
- Exercícios práticos envolvendo estratégias de busca heurística

Introdução

Objetivos

- Promover a compreensão dos algoritmos de busca gulosa e A* que fazem parte dos algoritmos de busca informada (heurística)

Busca Informada

- Busca Informada (Heurística):
- ➤ Uma busca heurística é uma busca que utiliza uma função h(n) onde, para cada nó n do espaço de busca, fornece uma avaliação do custo para atingir o estado final.

A função h(n) é chamada função heurística.

- Função heurística h => estima o custo do menor caminho do estado atual até o estado final mais próximo;
- Funções heurísticas são específicas para cada problema
 - Roteamento = rota mais barata de Corumbá a Campo Grande;
 - > Arvore = distância direta entre n e o nó final.

- Os problemas de IA empregam heurísticas, basicamente, em duas situações:
- Um problema pode não ter uma solução exata por causa das ambiguidades inerentes a sua formulação ou pela disponibilidade dos dados.

Exemplos: Diagnóstico médico, Sistemas de visão.

2. Um problema pode ter uma solução exata, mas o custo computacional para encontrá-la pode ser proibitivo.

Exemplo: Jogo de xadrez, Caixeiro viajante.

 \triangleright **Exemplo:** Definição da função h(n) dependente do problema.





> Exemplo: Heurísticas possíveis para o problema.



 $> h_1 = n^0$ de elementos em posições erradas.



Neste exemplo, os números 2, 8, 6 e 1 estão na posição errada, sendo assim, $h_1 = 4$.

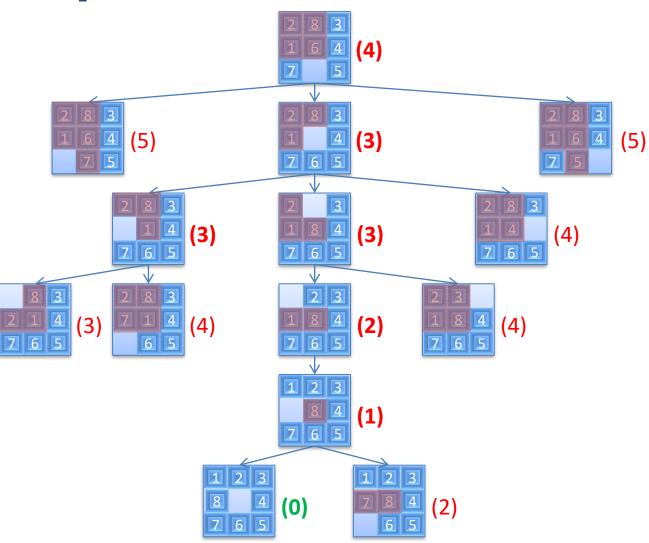
 h_2 = soma das distâncias d_n de cada elemento à posição final — objetivo (city block distance — Manhattan distance).



- \triangleright O 2 deve andar uma casa para a direita, assim d_2 =1;
- \triangleright O 8 deve andar uma casa para a esquerda e uma para baixo, assim d_8 =2;
- \triangleright O 3, 4, 5 e 7 já estão em seus lugares, assim, d_3 =0 , d_4 =0 , d_5 =0 e d_7 =0;
- \triangleright O 6 deve andar uma casa para a baixo, assim d_6 =1;
- \triangleright O 1 deve andar uma casa para a cima, assim $d_1=1$;
- \triangleright Portanto, o valor de h_2 vale:

$$h_2 = \sum_{i=1}^{8} d_i = 1 + 2 + 0 + 0 + 0 + 0 + 1 + 1 = 5$$

 \triangleright **Exemplo:** $h_1 = n^0$ de elementos em posições erradas



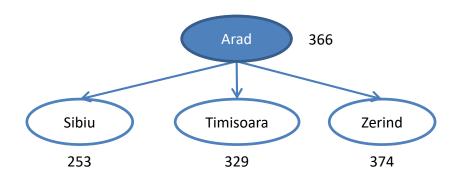
- Busca Gulosa
- > Também conhecida como busca de melhor escolha;
- > Tenta expandir o nó que está mais próximo do objetivo;
- ightharpoonup Cada nó é avaliado de acordo com uma função heurística f(n)=h(n);

- Busca Gulosa: Exemplo de roteamento na Romênia.
 - \blacktriangleright A heurística representa a distância em linha reta (DLR) para uma cidade, chamada h_{DLR} .
 - Se o objetivo for Bucarest, precisa-se conhecer a distância em linha reta de todas as cidades para a mesma.

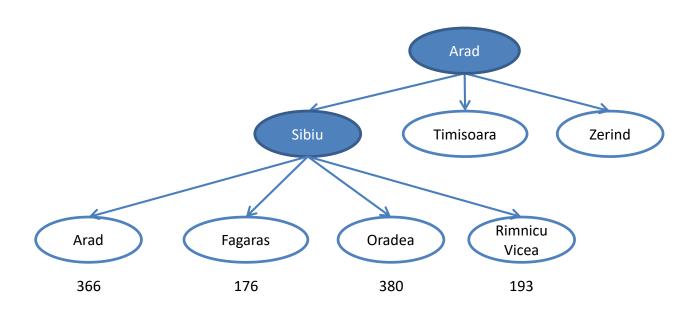
Tabela 1 - h_{DLR} para todas as cidades

Arad	366	Mehadia	241
Bucarest	0	Neamt	234
Craiova	160	Oradea	380
Drobeta	242	Pitesti	100
Eforie	161	Rimnicu Vilcea	193
Fagaras	176	Sibiu	253
Giurgiu	77	Timisoara	329
Hirsova	151	Urziceni	80
lasi	226	Vaslui	199
Lugoj	244	Zerind	374

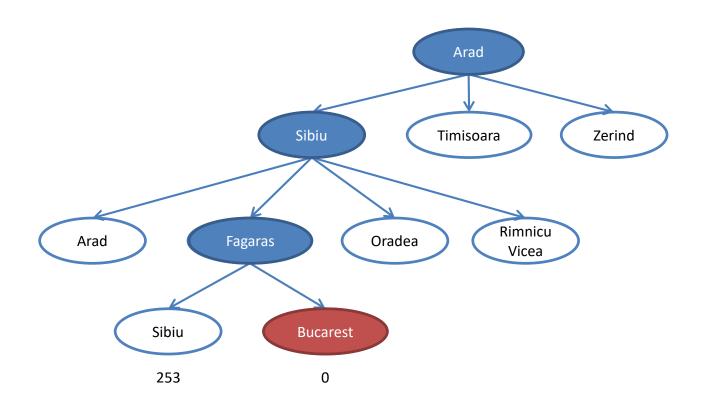
- Busca Gulosa: Exemplo de roteamento da Romênia, solução.
 - > Partindo da cidade Arad.
 - Neste caso, deve-se expandir Arad.



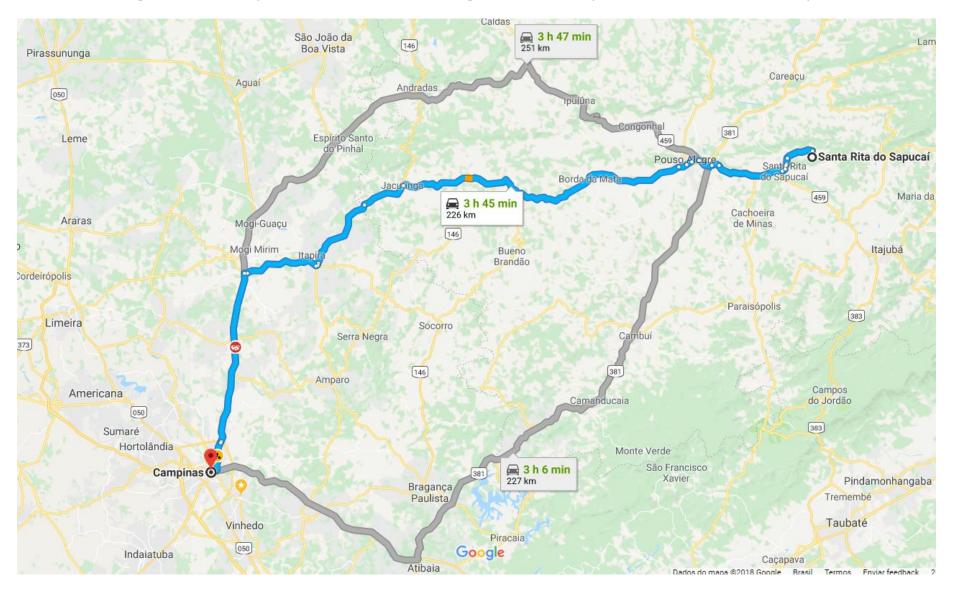
- Busca Gulosa: Exemplo do caixeiro viajante.
 - Partindo da cidade Arad.
 - Neste caso, deve-se expandir Arad.
 - Neste caso, o menor caminho leva a **Sibiu**, que deve ser expandida.



- Busca Gulosa: Exemplo do caixeiro viajante.
 - Partindo da cidade Arad.
 - Neste caso, deve-se expandir Arad.
 - Neste caso, o menor caminho leva a **Sibiu**, que deve ser expandida.
 - Neste caso, o menor caminho leva a Fagaras, que deve ser expandida.



Busca gulosa: Mapa das cidades (origem: S.R.Sapucaí, destino: Campinas)



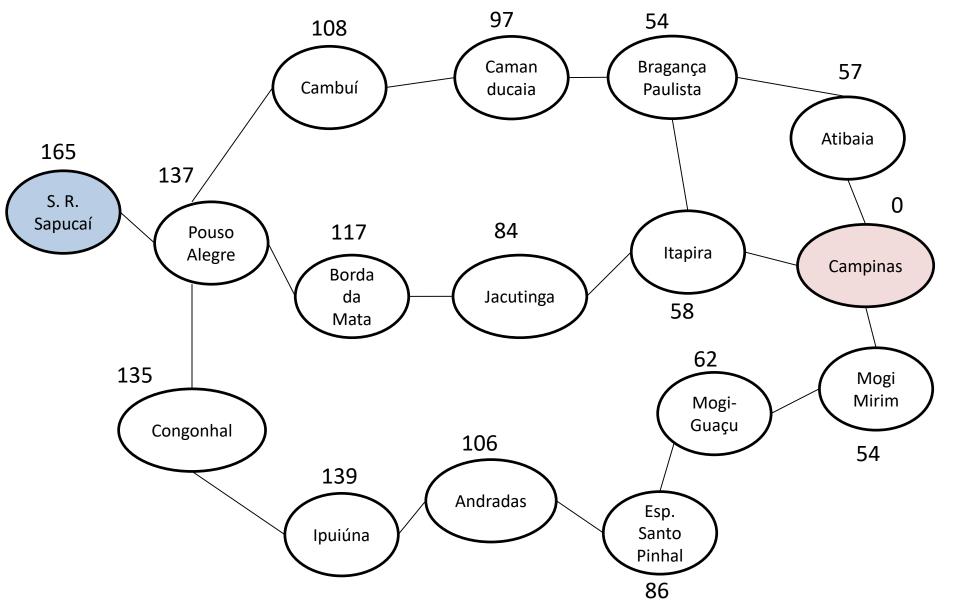
- Busca gulosa: Mapa das cidades (origem: S.R.Sapucaí, destino: Campinas)
 - \triangleright A heurística representa a distância em linha reta (DLR) para uma cidade, chamada h_{DLR} .
 - ➤ Se o objetivo for **Campinas**, precisa-se conhecer a distância em linha reta de todas as cidades para a mesma.

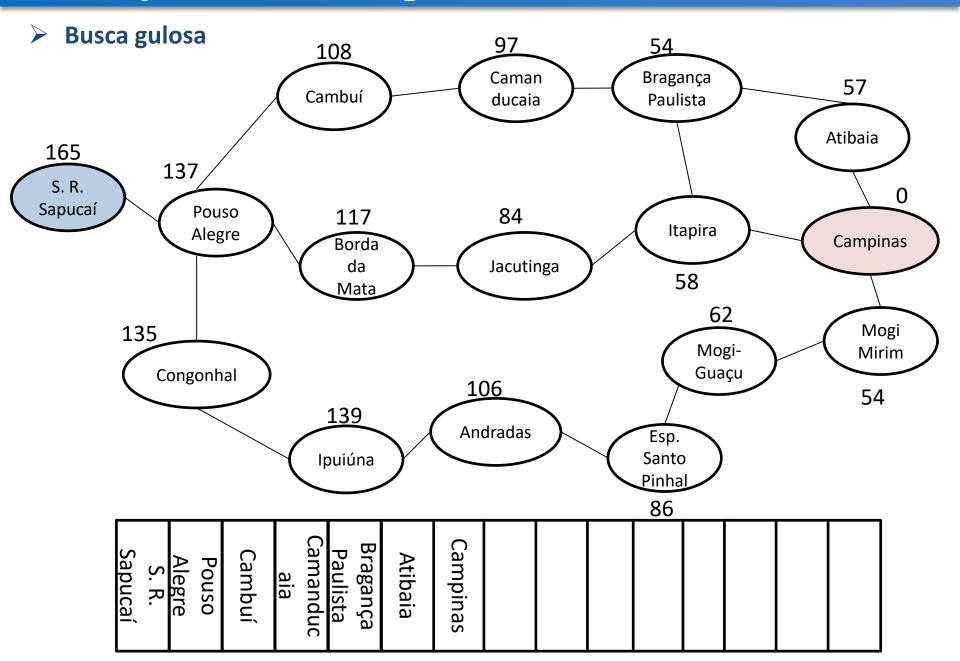
Tabela 2 - h_{DLR} para todas as cidades

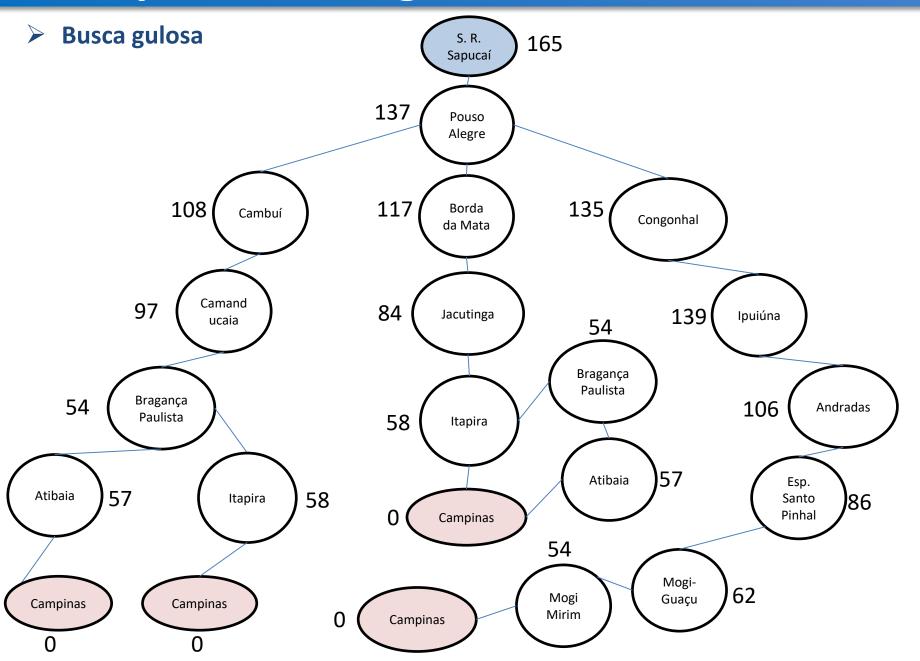
S. R. Sapucaí	165	Jacutinga	84
Pouso Alegre	137	Itapira	58
Cambuí	108	Congonhal	135
Camanducaia	97	Ipuiúna	139
Bragança Paulista	54	Andradas	106
Atibaia	57	Esp. Santo Pinhal	86
Campinas	0	Mogi-Guaçu	62
Borda da Mata	117	Mogi Mirim	54

Fonte: https://www.adistanciaentre.com

Busca gulosa

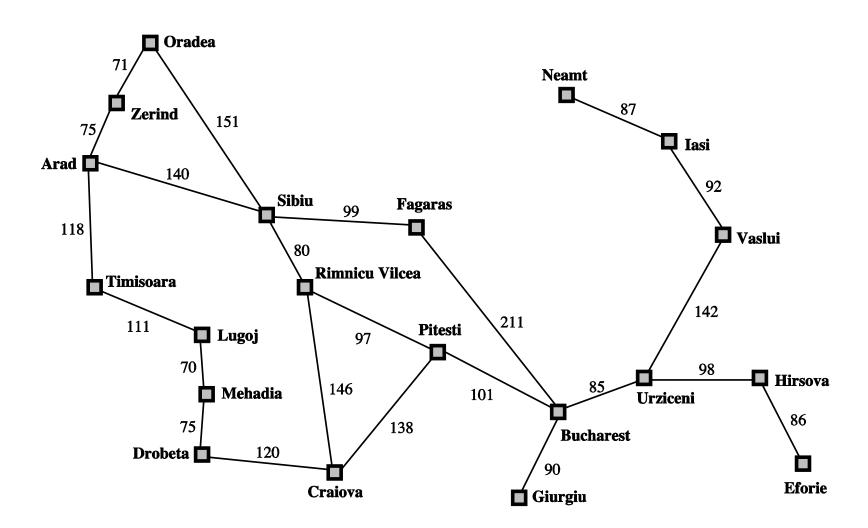




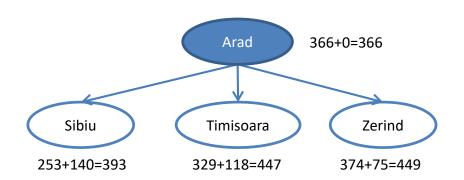


- Busca A*: Minimização do custo total estimado da solução.
 - Melhoria da busca gulosa;
 - \triangleright Além do custo para ir do nó ao objetivo, h(n), adiciona-se o custo g(n), que representa o custo para alcançar o nó;
 - \triangleright O objetivo da busca A* é minimizar o custo total h(n)+g(n) o que a torna uma solução ótima e completa, em certas condições;

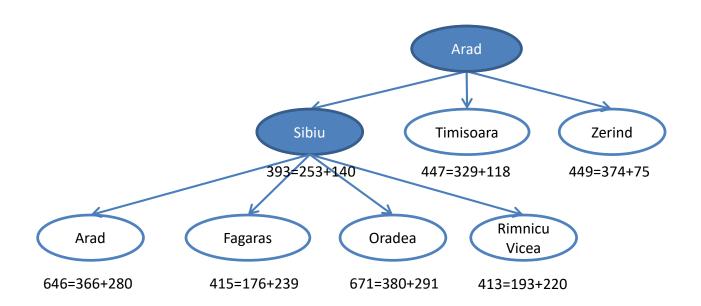
- Busca A*: Exemplo de roteamento da Romênia.
 - ➤ Para o mesmo objetivo, Bucarest, além da distância em linha reta para todas as cidades (Tabela 1), precisa-se considerar o custo de cada caminho.



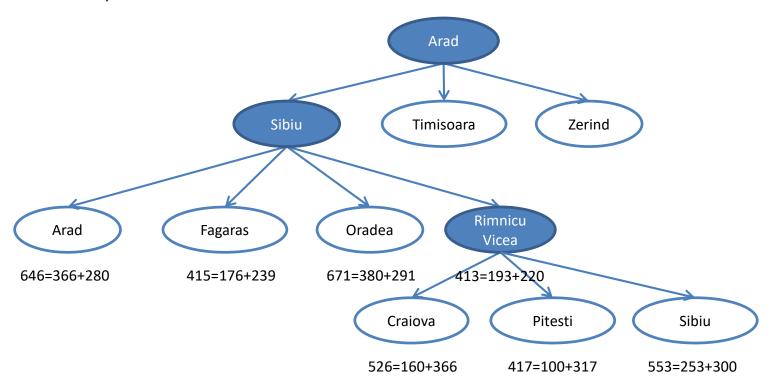
- Busca A*: Exemplo de roteamento da Romênia, solução.
 - Partindo da cidade Arad.
 - Neste caso, deve-se expandir Arad.



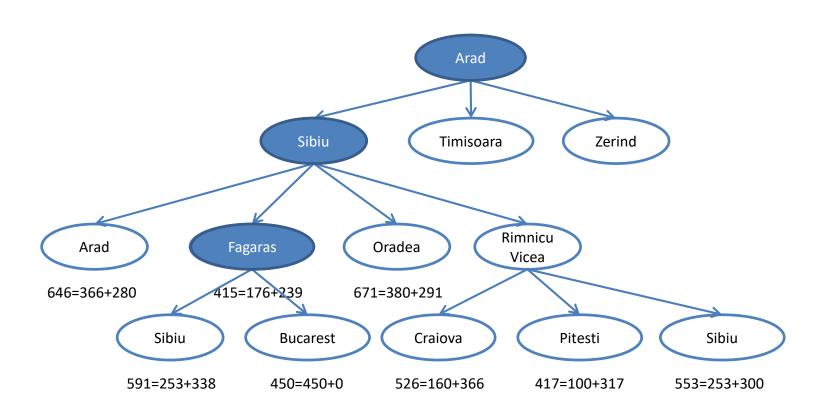
- Busca A*: Exemplo de roteamento da Romênia, solução.
 - > Partindo da cidade Arad.
 - Neste caso, deve-se expandir Arad.
 - Neste caso, o menor caminho leva a Sibiu, que deve ser expandida.



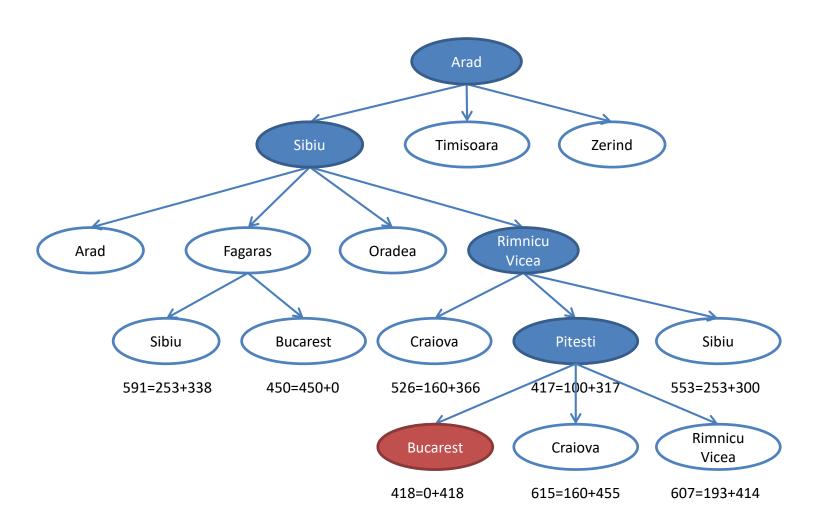
- Busca A*: Exemplo de roteamento da Romênia, solução.
 - Partindo da cidade Arad.
 - Neste caso, deve-se expandir Arad.
 - Neste caso, o menor caminho leva a **Sibiu**, que deve ser expandida.
 - ➤ Neste caso, o menor caminho leva a **Rimnicu Vicea**, que deve ser expandida.



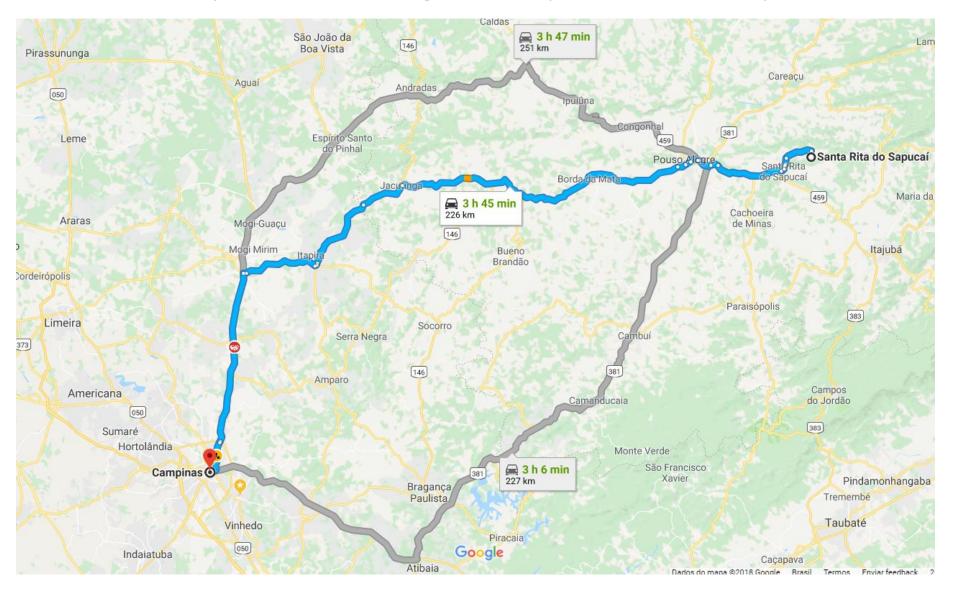
- Busca A*: Exemplo de roteamento da Romênia, solução.
 - Neste caso, o menor caminho leva a **Rimnicu Vicea**, que deve ser expandida.
 - Neste caso, o menor caminho leva a Fagaras, que deve ser expandida.



- Busca A*: Exemplo de roteamento da Romênia, solução.
 - > Neste caso, o menor caminho leva a Pitesi, que deve ser expandida.
 - Neste caso, a melhor solução seria.



Busca A*: Mapa das cidades (origem: S.R.Sapucaí, destino: Campinas)



- Busca A*: Mapa das cidades (origem: S.R.Sapucaí, destino: Campinas)
 - \blacktriangleright Além do custo para ir do nó ao objetivo, h(n), adiciona-se o custo g(n), que representa o custo para alcançar o nó;
 - Se o objetivo for **Campinas**, precisa-se conhecer a distância em linha reta de todas as cidades para a mesma e também o custo real que interliga todas as cidades do mapa.

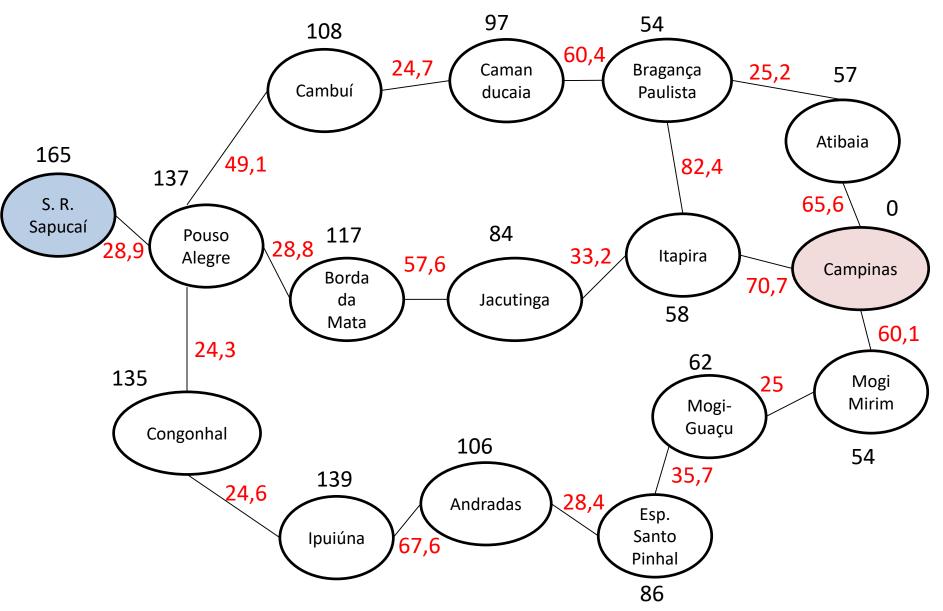
Tabela 2 - h_{DLR} para todas as cidades

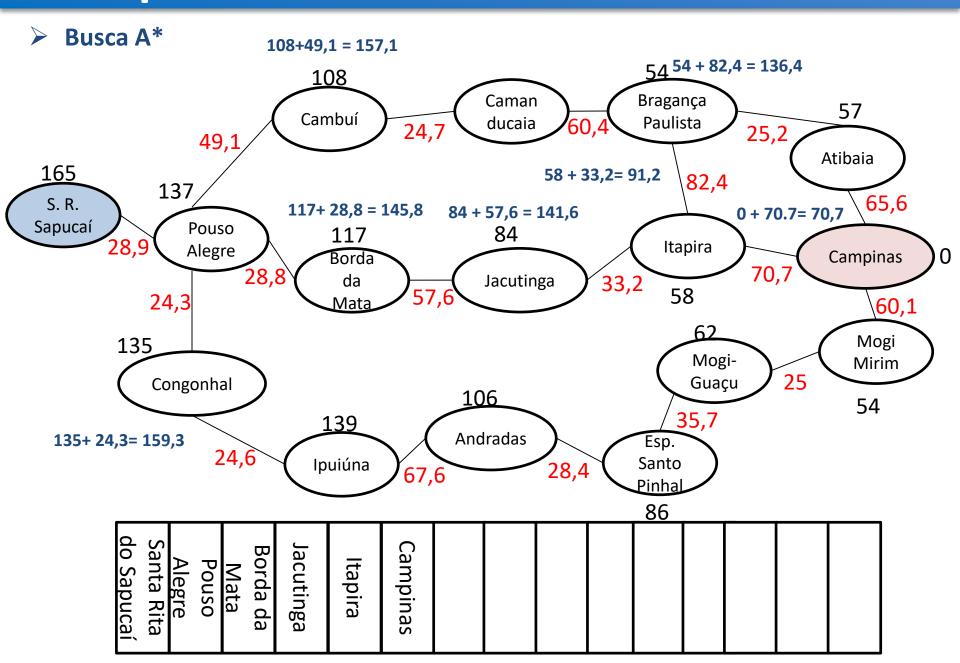
S. R. Sapucaí	165	Jacutinga	84
Pouso Alegre	137	Itapira	58
Cambuí	108	Congonhal	135
Camanducaia	97	Ipuiúna	139
Bragança Paulista	54	Andradas	106
Atibaia	57	Esp. Santo Pinhal	86
Campinas	0	Mogi-Guaçu	62
Borda da Mata	117	Mogi Mirim	54

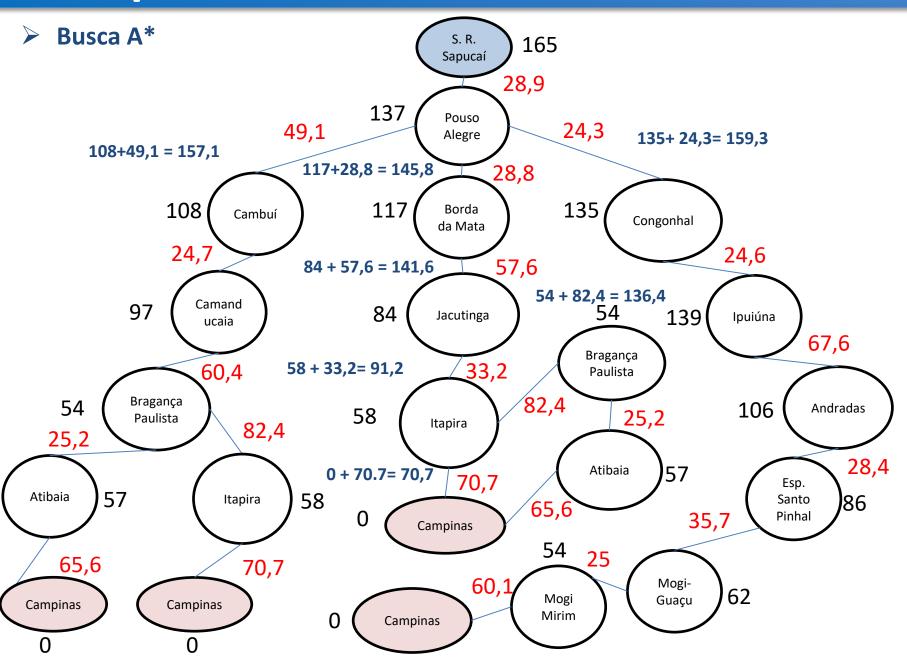
Tabela 3 – Distância real entre as cidades

Pouso Alegre	28,9
Cambuí	49,1
Congonhal	24,3
Borda da Mata	28,8
Camanducaia	24,7
Bragança Paulista	60,4
Atibaia	25,2
Itapira	82,4
Campinas	65,6
Campinas	70,7
Jacutinga	57,6
Itapira	33,2
Ipuiúna	24,6
Andradas	67,5
Esp. Santo Pinhal	28,4
Mogi-Guaçu	35,7
Mogi Mirim	25
Campinas	60,1
	Congonhal Borda da Mata Camanducaia Bragança Paulista Atibaia Itapira Campinas Campinas Jacutinga Itapira Ipuiúna Andradas Esp. Santo Pinhal Mogi-Guaçu Mogi Mirim









Referências

[RUSSELL, S.; NORVIG, P.; Inteligência Artificial, 2ºed. – Cap. 1 e 2]

[RUSSELL, S.; NORVIG, P.; Inteligência Artificial, 2ºed. – Cap. 3]

