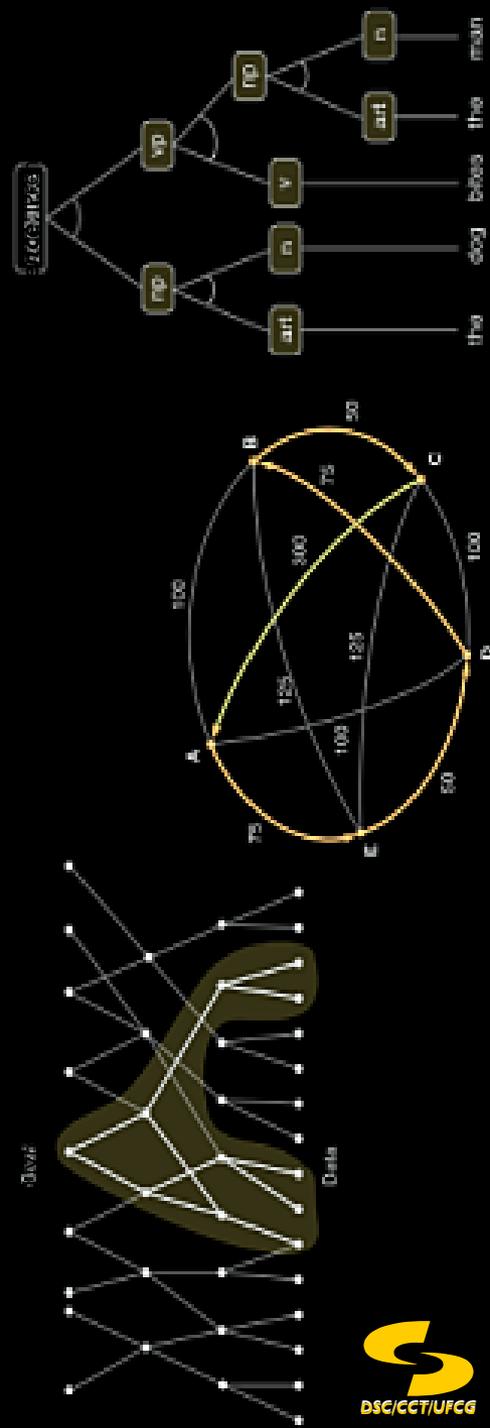


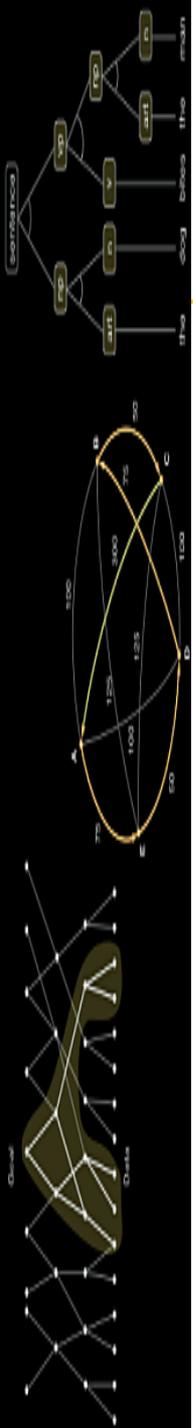
Universidade Federal de Campina Grande
Departamento de Sistemas e Computação
Curso de Pós-Graduação em Ciência da Computação

Inteligência Artificial

Resolução de Problemas (Parte III)

Prof.^a Joseana Macêdo Fachine Régis de Araújo
joseana@computacao.ufcg.edu.br

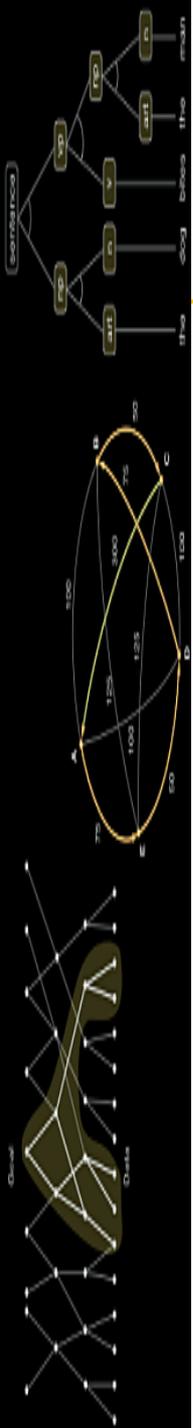




Em Busca de Soluções

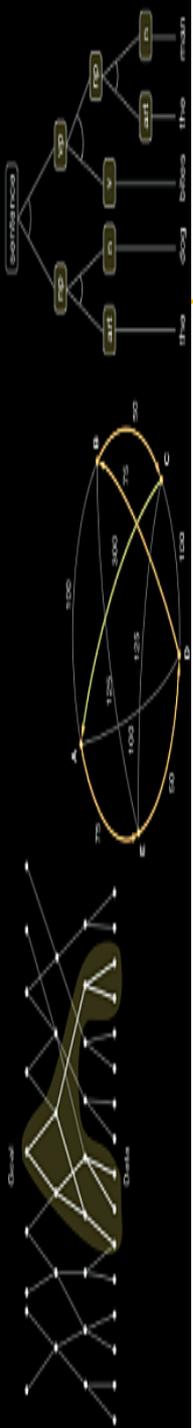
Tópico

- Busca Heurística



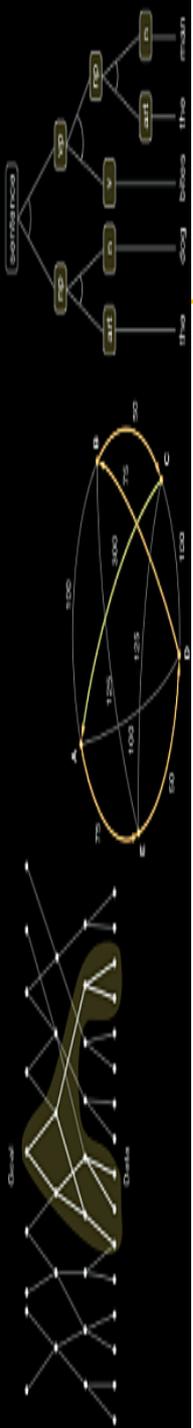
Busca Heurística

- ❑ **Heurística** - Informação específica do domínio que pode ser usada para guiar o processo de busca.
- ❑ Em muitos casos uma heurística envolve a aplicação de uma função que avalia um nó particular e prediz a qualidade dos seus nós sucessores.
- ❑ Uma função heurística de avaliação no jogo-da-velha poderia ser o número de linhas, colunas e diagonais ainda disponíveis, quanto maior este número maior a chance de vitória.



Busca Heurística

- Os problemas de IA empregam heurísticas, basicamente, em duas situações:
 1. Um problema pode não ter uma solução exata por causa das ambiguidades inerentes na 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.



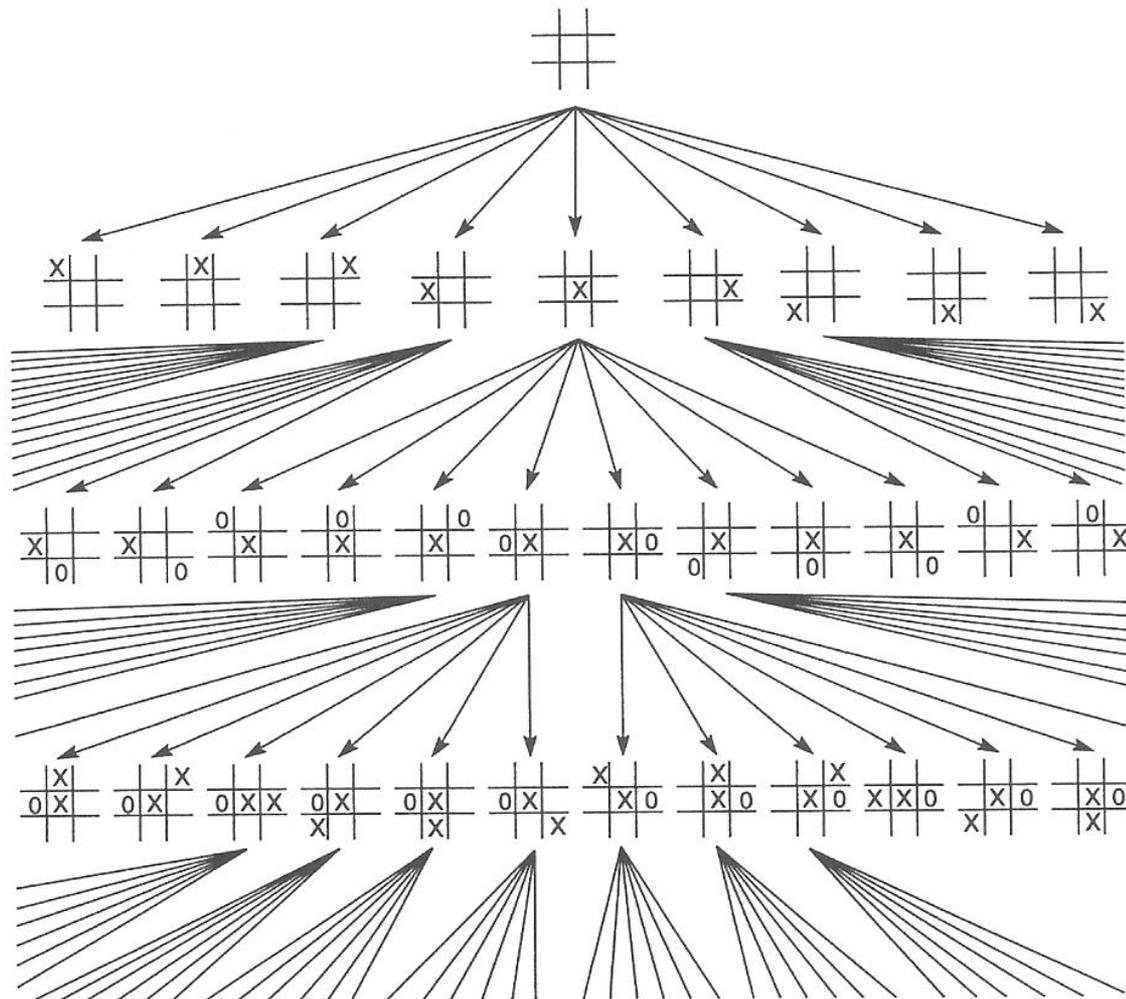
Busca Heurística

- ❑ As heurísticas podem falhar.
- ❑ Uma heurística é apenas uma conjectura informada sobre o próximo passo a ser tomado na solução de um problema.
- ❑ A heurística é baseada na experiência e na intuição.
- ❑ Uma heurística pode levar um algoritmo de busca a uma solução subótima ou, inclusive, levá-lo a não conseguir encontrar uma solução.

George Polya define heurística como “o estudo dos métodos e das regras de descoberta e invenção” (Polya, 1945) – relacionada com o termo grego original, o verbo eurisco (“Eu descubro”). Quando Arquimedes emergiu de seu famoso banho segurando a coroa de ouro, ele gritou “Eureka!” (“Eu descobri!”).

Busca Heurística – Exemplo ...

Porção do espaço de estados para o jogo-da-velha



9

8

7

⋮

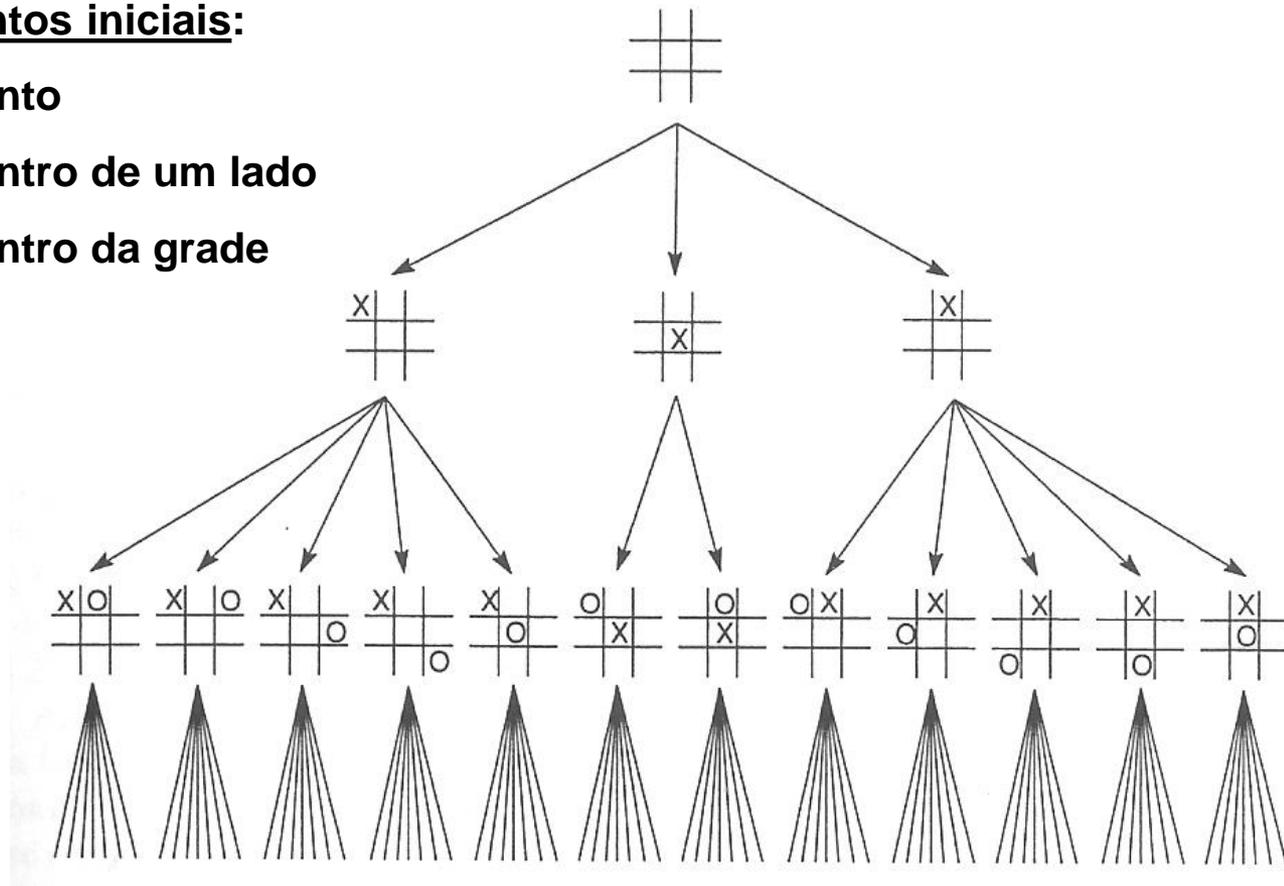
Nº de
caminhos
= 9!

Busca Heurística – Exemplo ...

Os primeiros três níveis do espaço de estados do jogo-da-velha reduzidos por simetria.

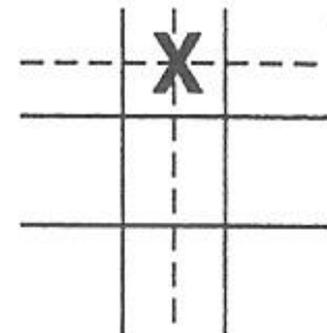
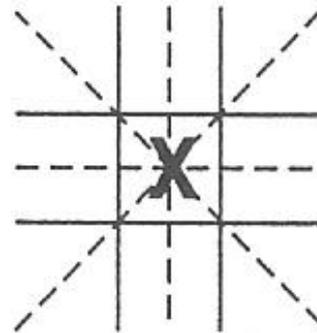
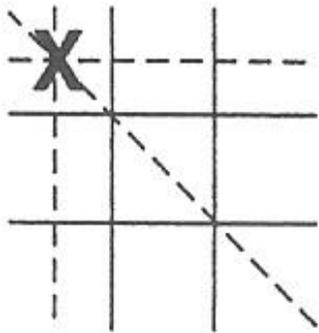
3 movimentos iniciais:

- Para o canto
- Para o centro de um lado
- Para o centro da grade



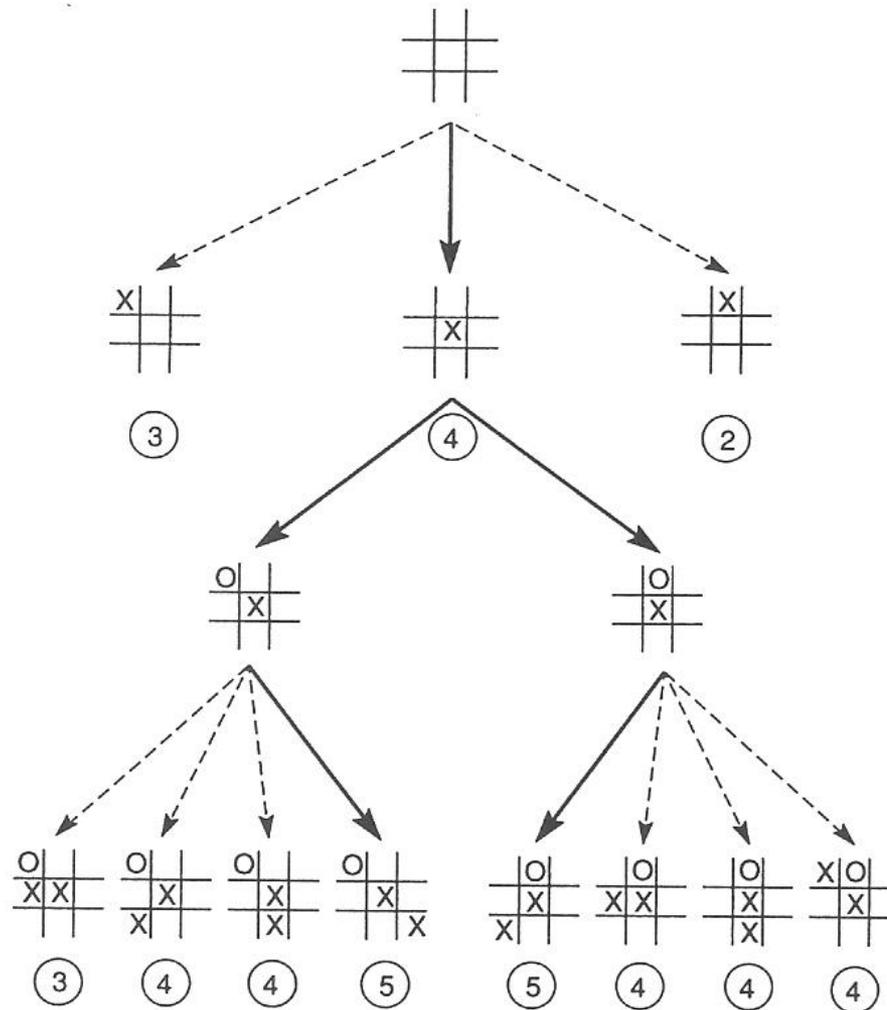
Busca Heurística – Exemplo ...

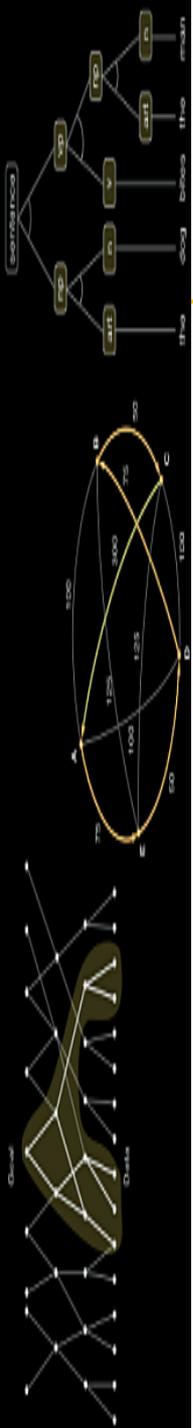
A heurística do “maior número de vitórias” aplicada aos primeiros filhos do jogo-da-velha.



Busca Heurística – Exemplo

Espaço de estados reduzido heurísticamente para o jogo-da-velha.

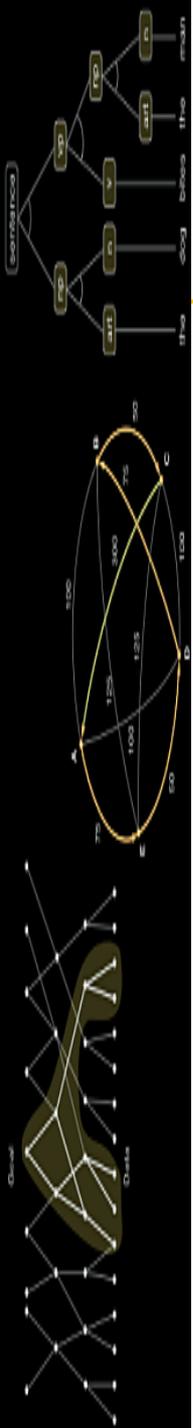




Busca Heurística

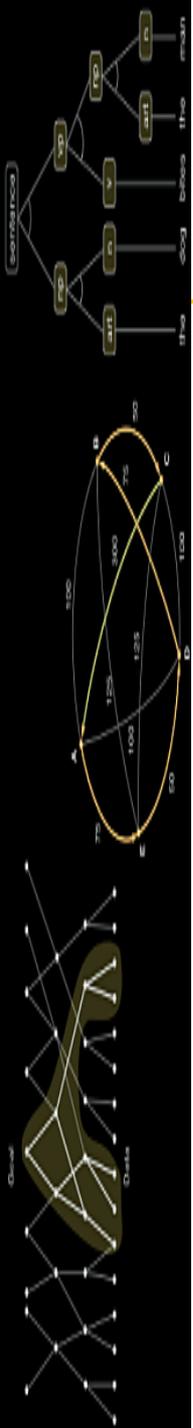
Estratégias de Busca Heurística

- Usam *conhecimento específico* do problema na busca da solução.
- Mais eficientes do que a busca não informada.
- Algoritmo geral: Busca pela Melhor Escolha - BME (*Best-first search*)
 - Seleciona para expansão o nó que tiver o menor custo estimado até a meta (objetivo), segundo uma *função de avaliação* $f(n)$.
 - Tipicamente $f(n)$ usa uma *função heurística* $h(n) =$ custo estimado do caminho mais econômico do nó n até um nó objetivo (Restrição inicial: se n é um nó objetivo, $h(n)=0$).



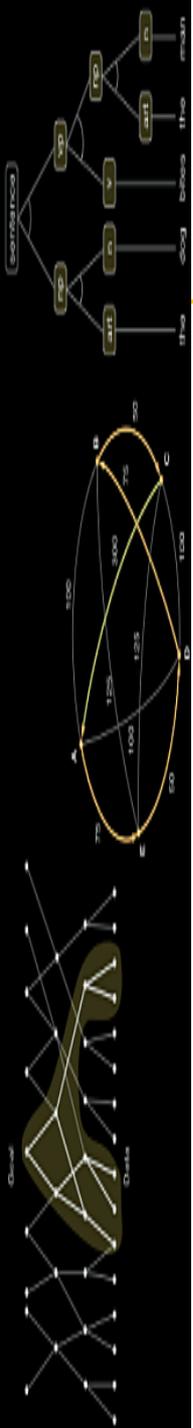
Busca Heurística

- Uma forma de uso da informação heurística sobre um problema consiste em computar estimativas numéricas para os nós no espaço de estados;
- Uma estimativa indica o quanto um nó é promissor com relação ao alcance de um nó-objetivo;
- A idéia é continuar a busca sempre a partir do nó mais promissor no conjunto de candidatos;
- O programa de busca do melhor caminho (escolha) é baseado neste princípio.



Busca Heurística

- ❑ **Busca do melhor caminho** - pode ser derivada de um refinamento da busca em largura.
- ❑ **Busca em largura** - sempre escolhe para expansão os menores caminhos-candidatos (isto é, os nós extremos menos profundos da busca).
- ❑ **Busca do melhor caminho** - refina este princípio calculando uma estimativa heurística para cada candidato e escolhe para expansão o melhor candidato de acordo com esta estimativa.



Busca Heurística

Greedy best-first search

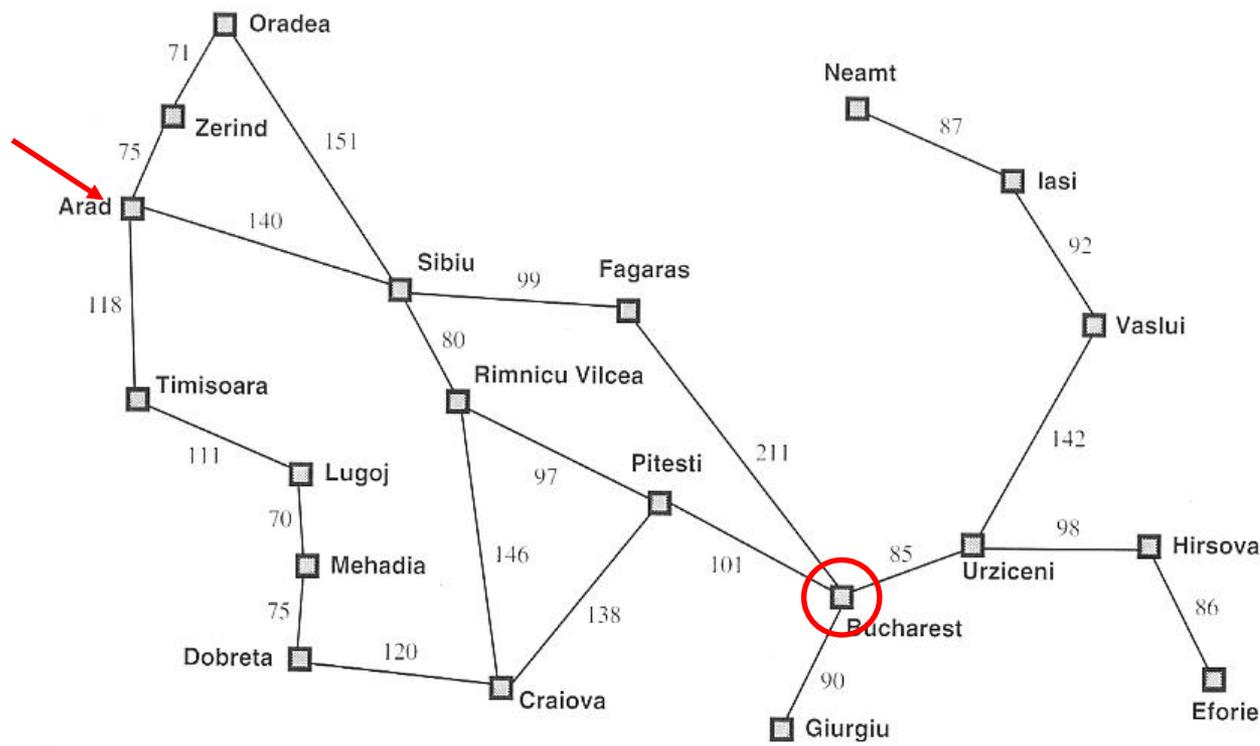
(Busca gulosa pela melhor escolha)

- ❑ Tenta expandir o nó mais próximo à meta, na suposição de que isso provavelmente levará a uma solução rápida.
- ❑ Avalia nós para expandir com base unicamente na função heurística: $f(n) = h(n)$
- ❑ **Exemplo:** encontrar a melhor rota (rota mais curta) de uma cidade a outra, num mapa.
 - $h(n)$ = distância em linha reta entre as cidades e a cidade-meta.

Busca Heurística

Exemplo: Localização de rotas na Romênia, usando a heurística de **distância em linha reta (h_{DLR})**

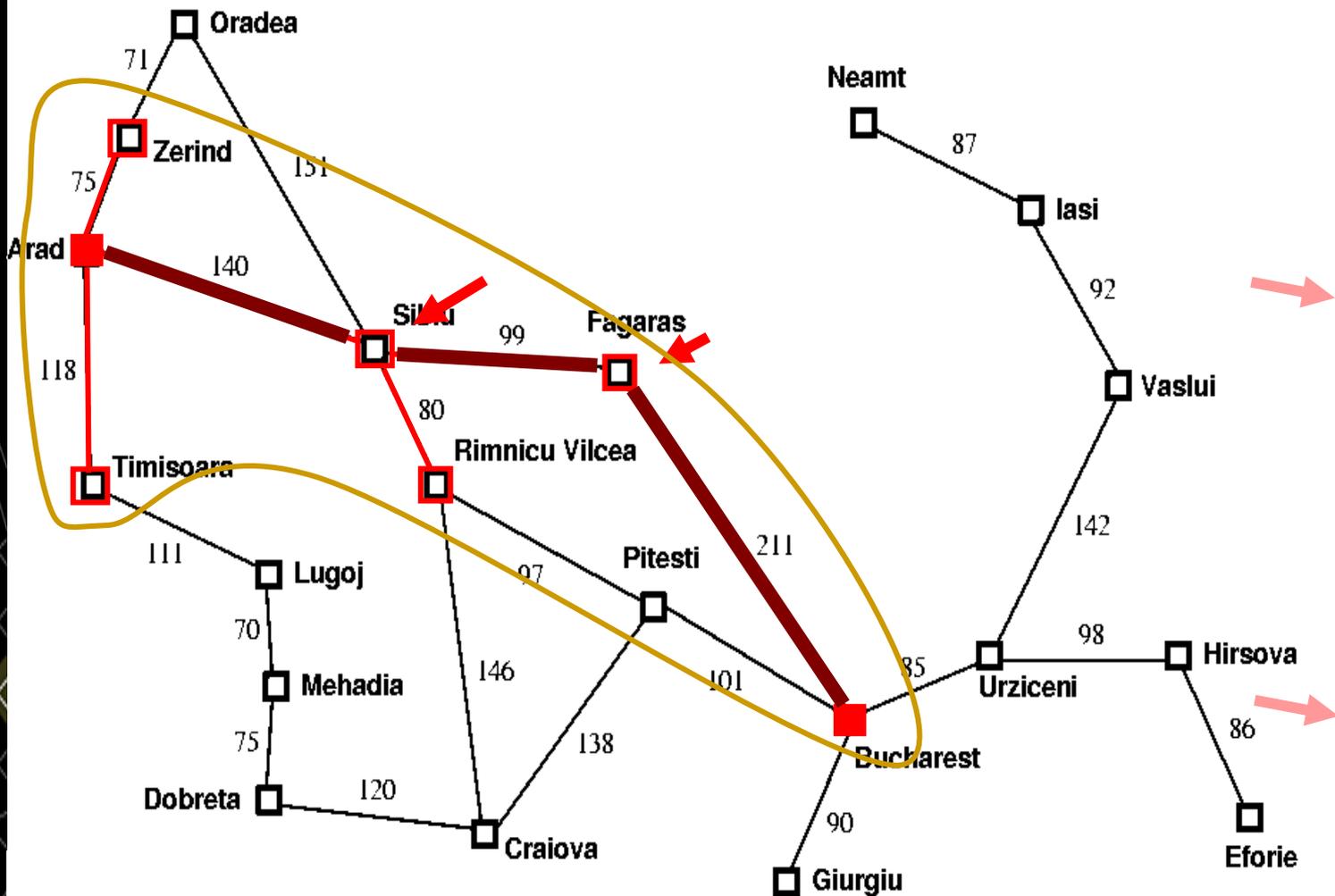
Objetivo: Bucharest (Bucareste)



Um mapa rodoviário simplificado de parte da Romênia.

Busca Heurística

Busca pela melhor escolha - Busca Gulosa



Distância em linha
reta para 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

Exemplo – Passo a Passo ...

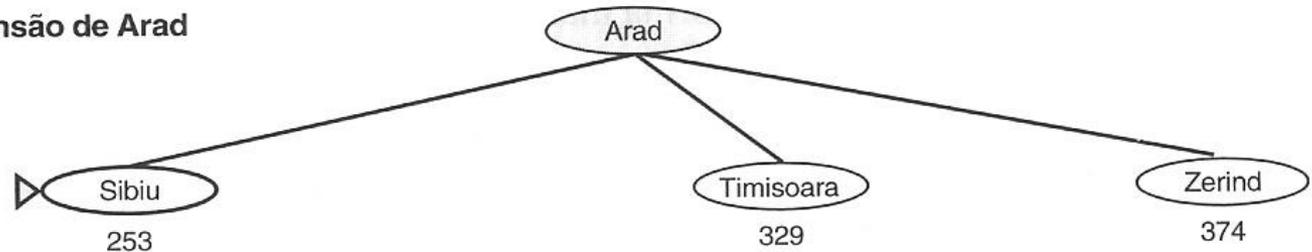
(a) O estado inicial



Fases de uma busca gulosa pela melhor escolha para Bucareste, usando-se a heurística de distância em linha reta h_{DLR} . Os nós são identificados por seus valores de h .

Exemplo – Passo a Passo ...

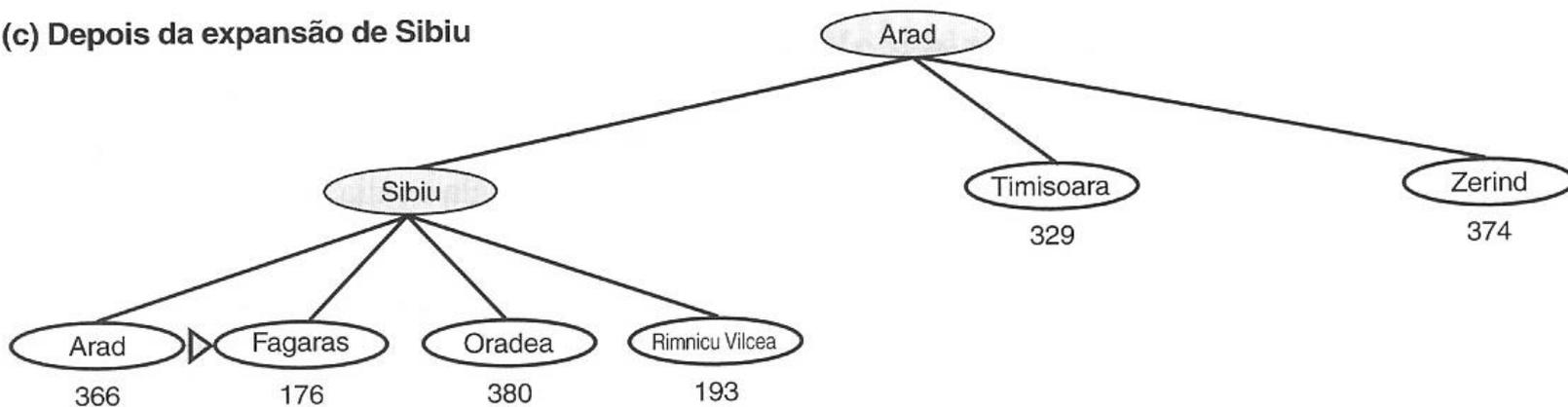
(b) Depois da expansão de Arad



Fases de uma busca gulosa pela melhor escolha para Bucareste, usando-se a heurística de distância em linha reta h_{DLR} . Os nós são identificados por seus valores de h .

Exemplo – Passo a Passo ...

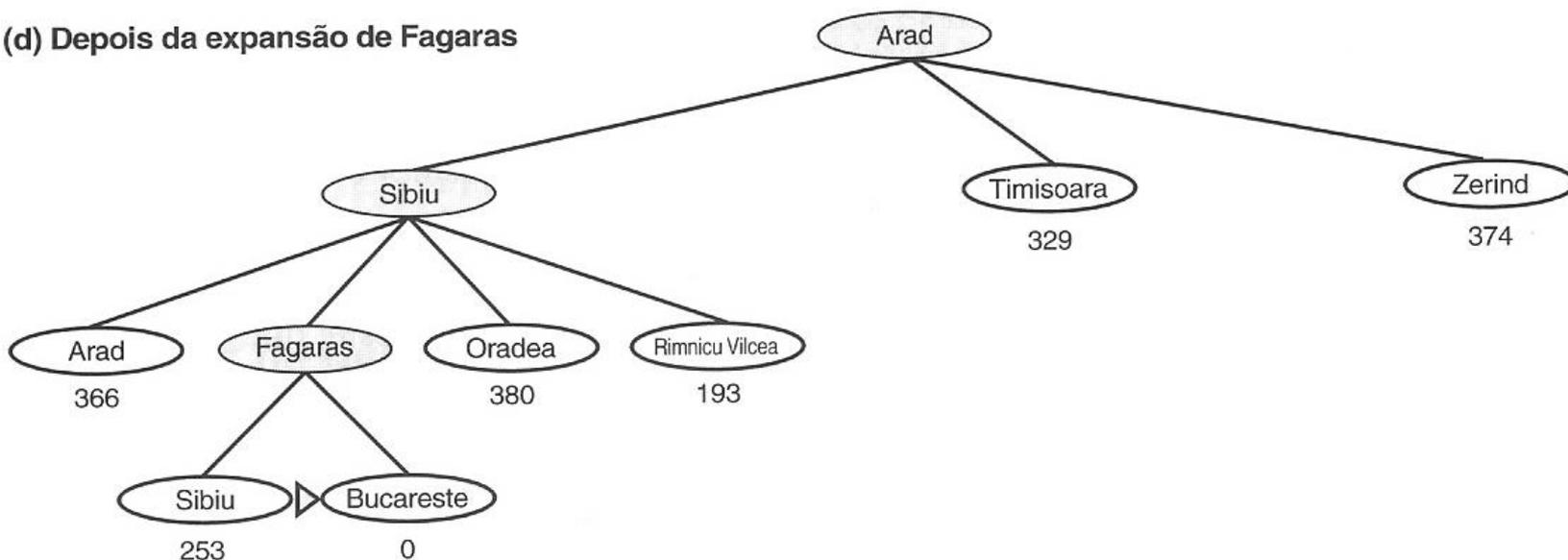
(c) Depois da expansão de Sibiu



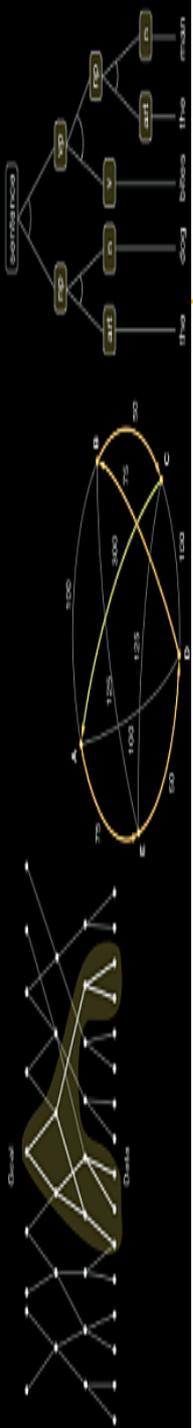
Fases de uma busca gulosa pela melhor escolha para Bucareste, usando-se a heurística de distância em linha reta h_{DLR} . Os nós são identificados por seus valores de h .

Exemplo – Passo a Passo ...

(d) Depois da expansão de Fagaras



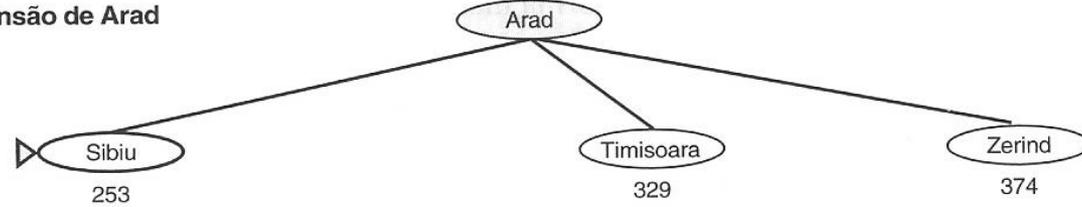
Fases de uma busca gulosa pela melhor escolha para Bucarest, usando-se a heurística de distância em linha reta h_{DLR} . Os nós são identificados por seus valores de h.



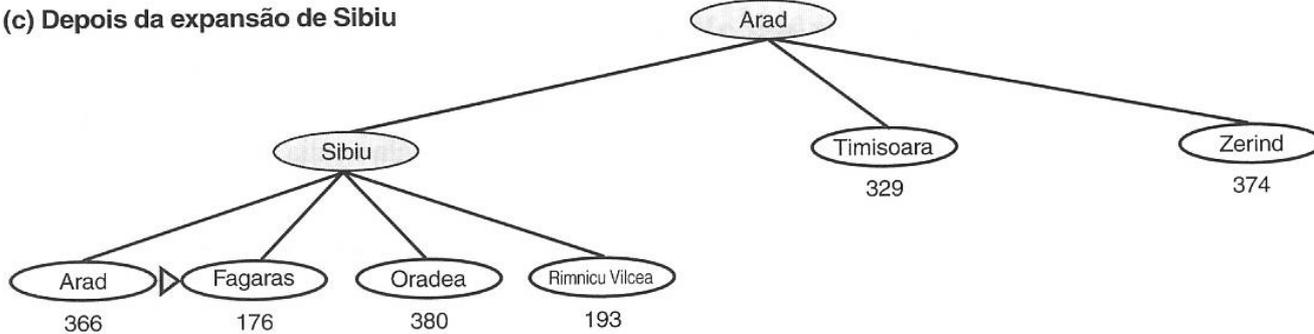
(a) O estado inicial



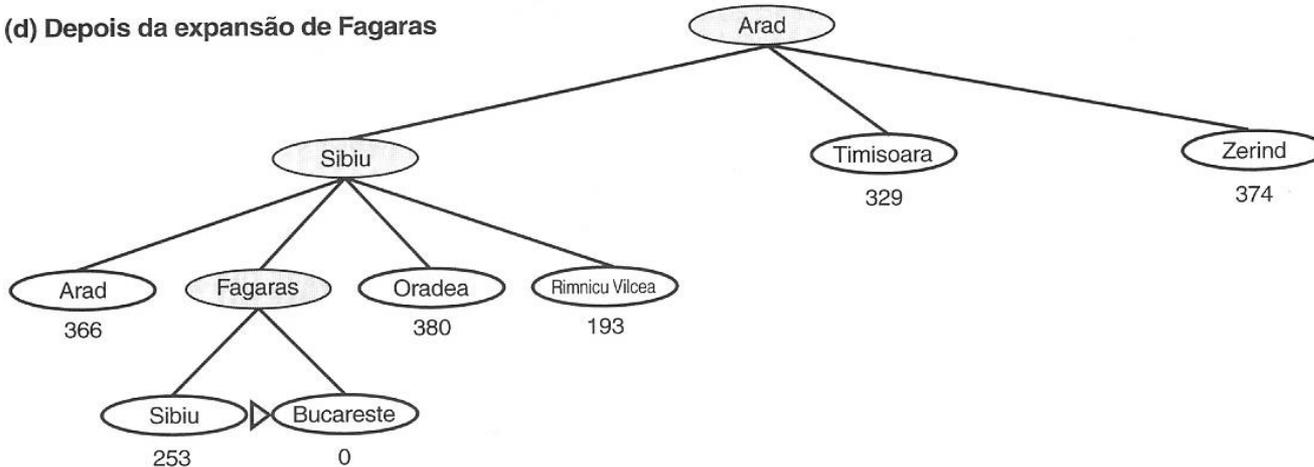
(b) Depois da expansão de Arad



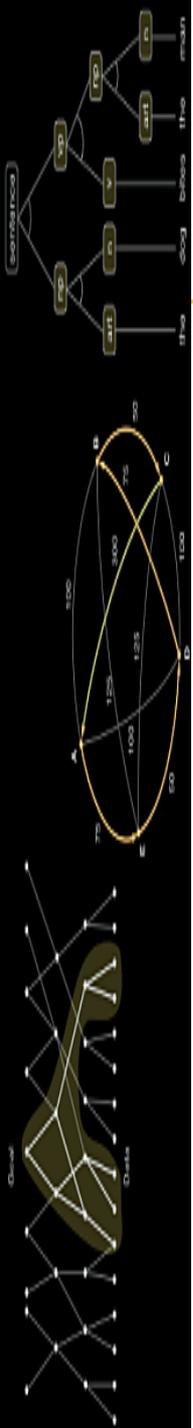
(c) Depois da expansão de Sibiu



(d) Depois da expansão de Fagaras



Fases de uma busca gulosa pela melhor escolha para Bucareste, usando-se a heurística de distância em linha reta h_{DLR} . Os nós são identificados por seus valores de h .



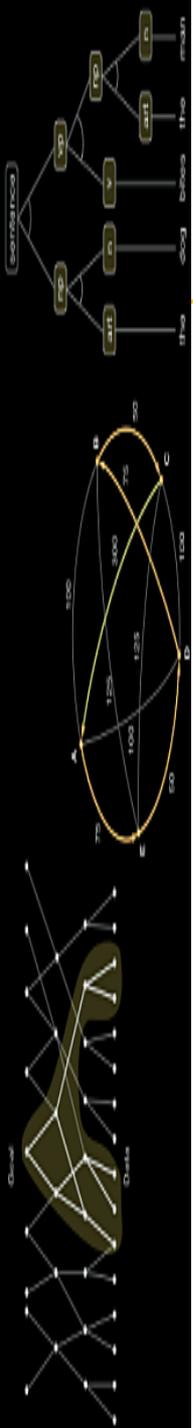
Busca Heurística

Não é completa

- ❑ pode entrar em ciclos e não encontrar a solução se não detectar estados repetidos;
- ❑ pode se perder em um caminho infinito e nunca retroceder para tentar outras opções.

Não é ótima

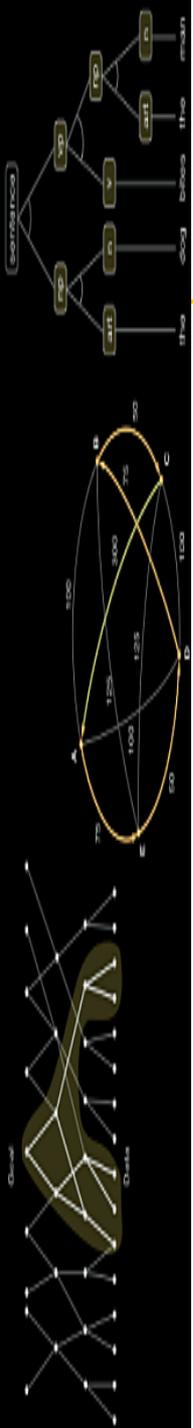
- ❑ No exemplo encontrou caminho (Arad, Sibiu, Fagaras, Bucharest) que é 32km maior que (Arad, Sibiu, Rimnicu Vilcea, Pitesti, Bucharest)
- ❑ Dependendo do problema e da qualidade da heurística a complexidade pode ter uma redução substancial.



Busca Heurística

BME mais “famoso”: Busca A*

- ❑ **Objetivo:** Minimizar o custo total estimado da solução.
- ❑ **Função de avaliação:** $f(n) = g(n) + h(n)$
 - $g(n)$ = distância (custo) do nó inicial ao nó n
 - $h(n)$ = distância (custo) estimada de n ao nó final
 - Assim, $f(n)$ estima o custo da melhor solução que passa por n .
- ❑ **A*** expande o nó de menor valor de f na fronteira do espaço de estados.



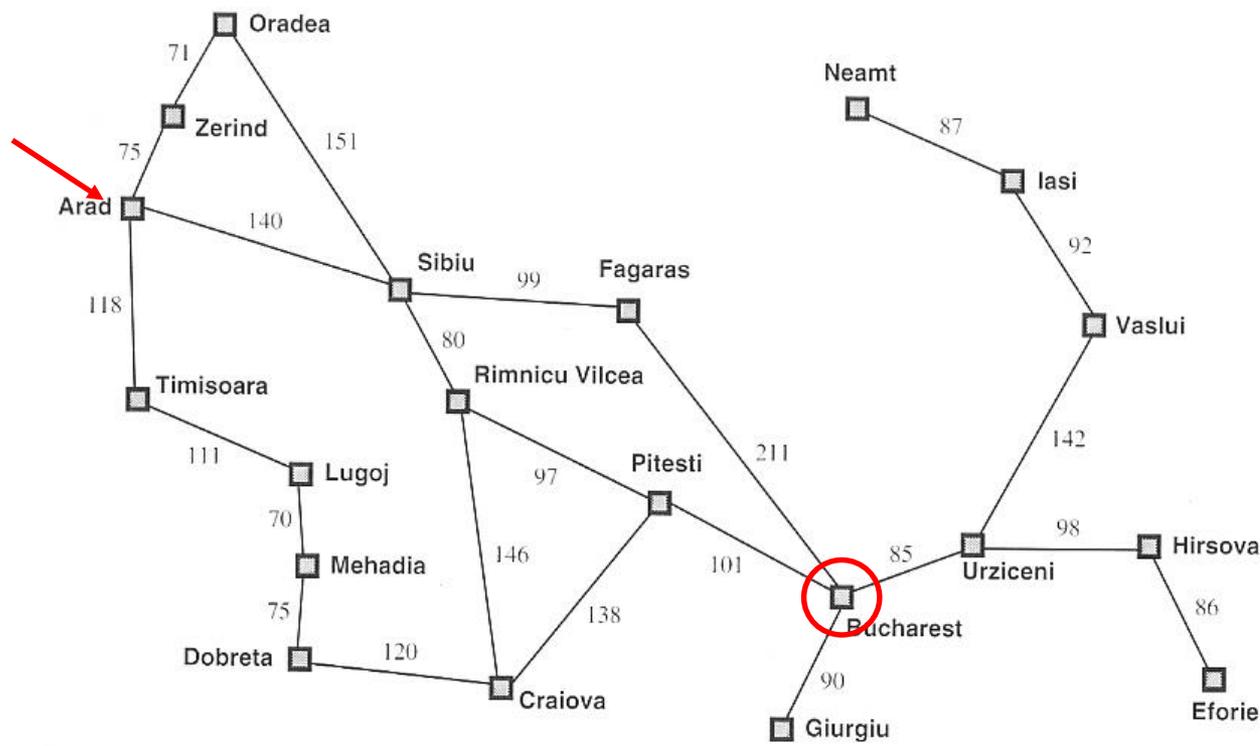
Busca Heurística

- Quando n é encontrado pelo processo de busca, tem-se a seguinte situação:
 - Um caminho de i para n já deve ter sido encontrado e o seu custo pode ser calculado como a soma dos custos dos arcos no caminho, e pode servir como uma estimativa $g(n)$ do custo mínimo de i para n .
 - $h(n)$ é mais problemático porque o espaço entre n e t ainda não foi explorado, e portanto $h(n)$ é meramente um palpite baseado no conhecimento geral do algoritmo sobre o problema particular.
 - Não existe um método universal para construção de h , pois depende do domínio do problema.

Busca Heurística

Exemplo: Localização de rotas na Romênia, usando a **Busca A***

Objetivo: Bucharest (Bucareste)

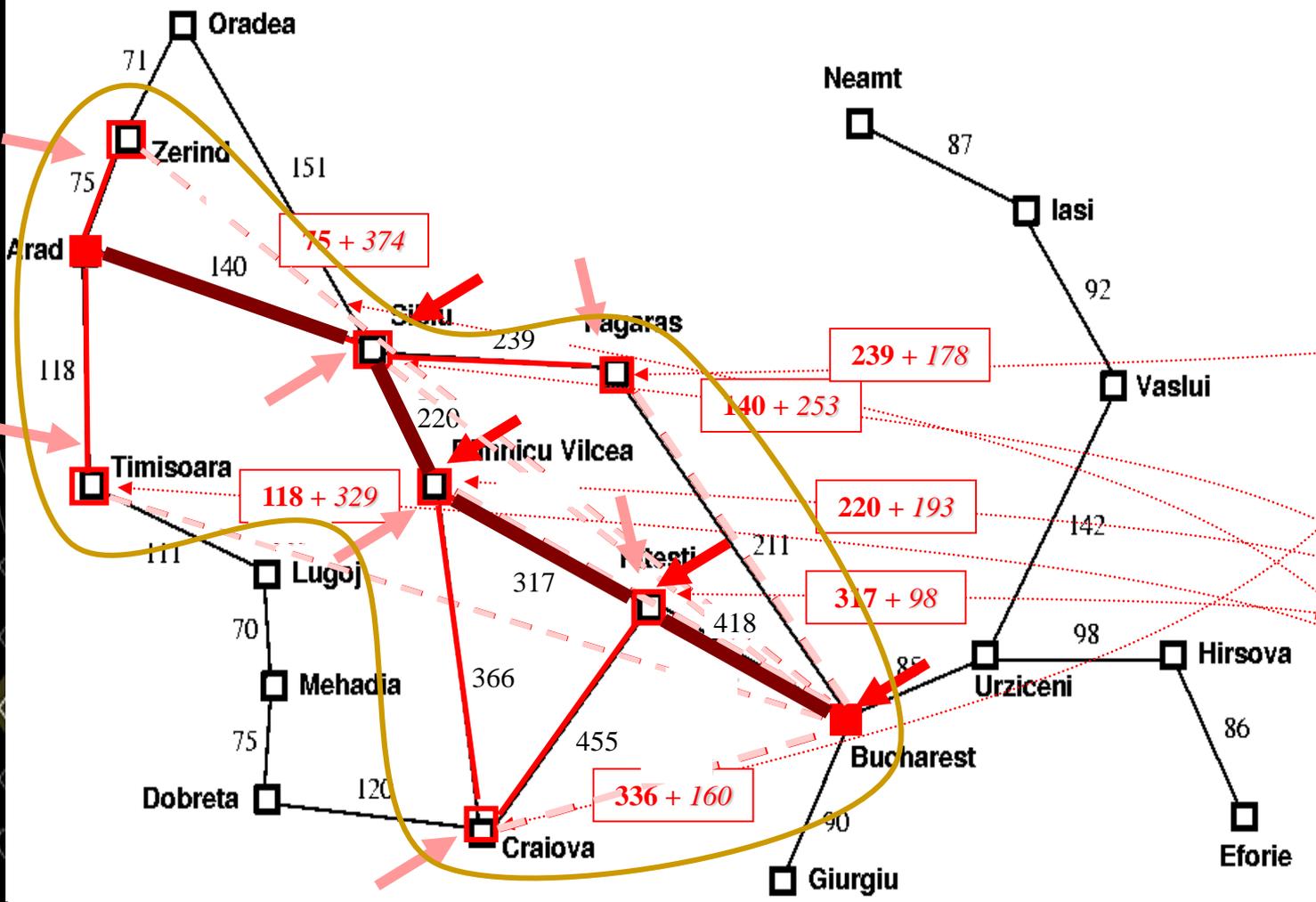


	Straight-line distance to Bucharest
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Um mapa rodoviário simplificado de parte da Romênia.

Busca Heurística

Busca pela melhor escolha - Algoritmo A*

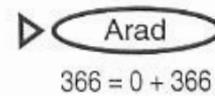


Distância em linha
reta para Bucharest:

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

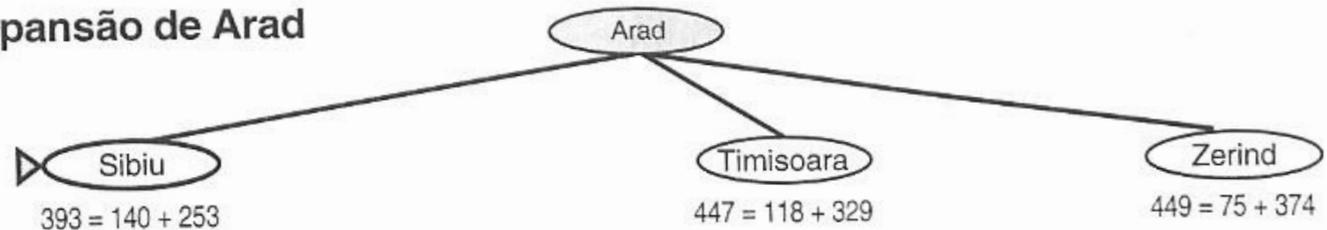
Exemplo – Passo a Passo ...

(a) O estado inicial



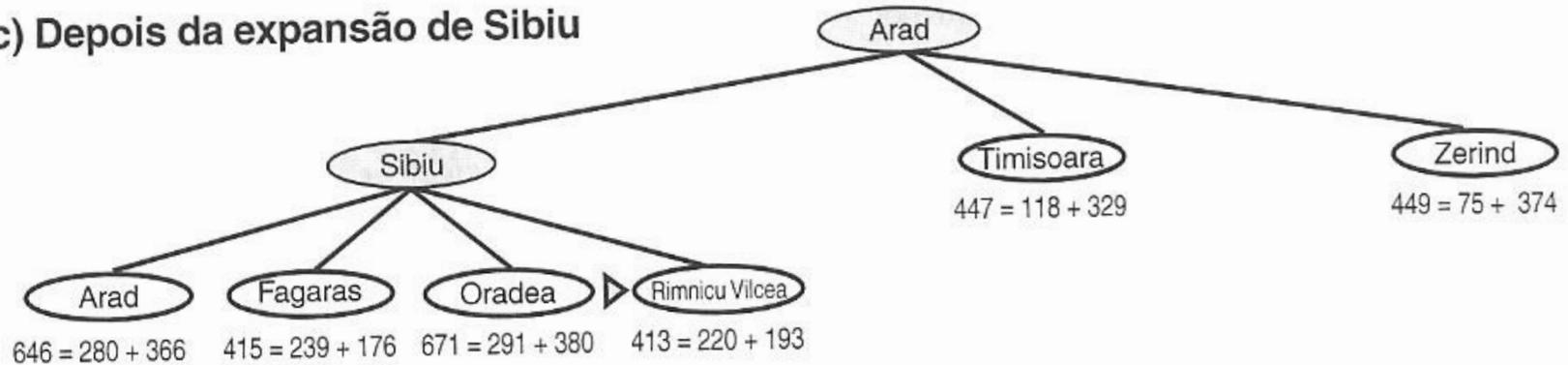
Exemplo – Passo a Passo ...

(b) Depois da expansão de Arad



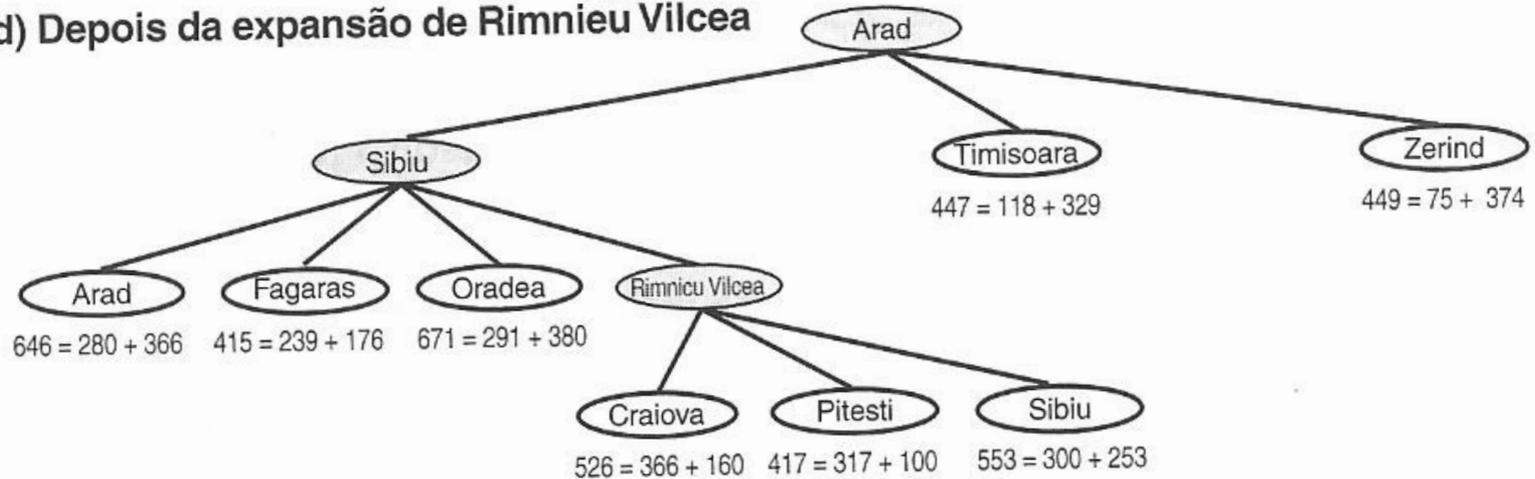
Exemplo – Passo a Passo ...

(c) Depois da expansão de Sibiu



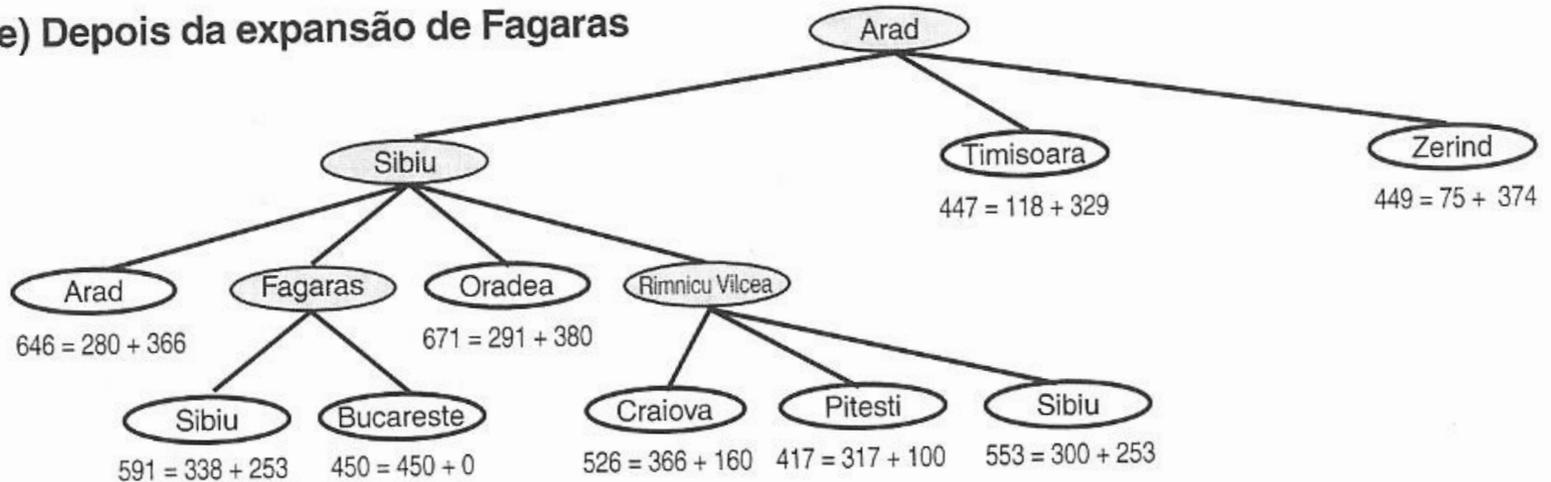
Exemplo – Passo a Passo ...

(d) Depois da expansão de Rimnieu Vilcea



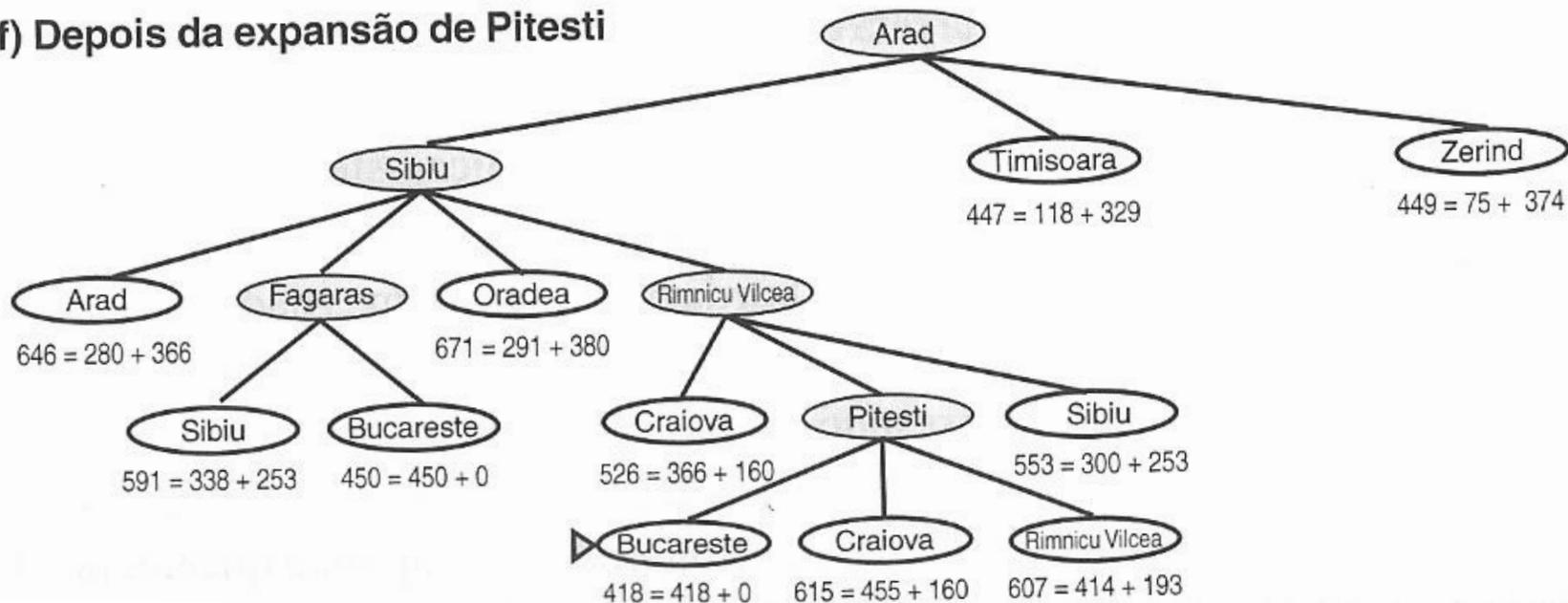
Exemplo – Passo a Passo ...

(e) Depois da expansão de Fagaras



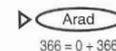
Exemplo – Passo a Passo

(f) Depois da expansão de Pitesti

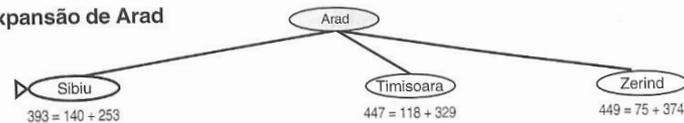


Estágios em uma busca A* por Bucareste. Os nós estão rotulados $f = g + h$. Os valores de h são distâncias em linha reta para Bucareste.

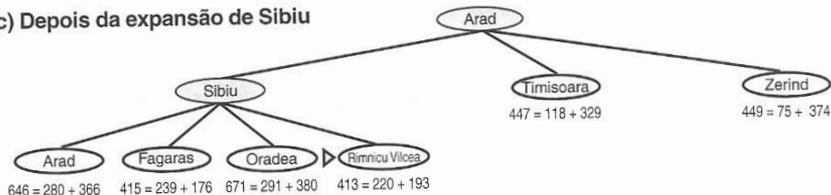
(a) O estado inicial



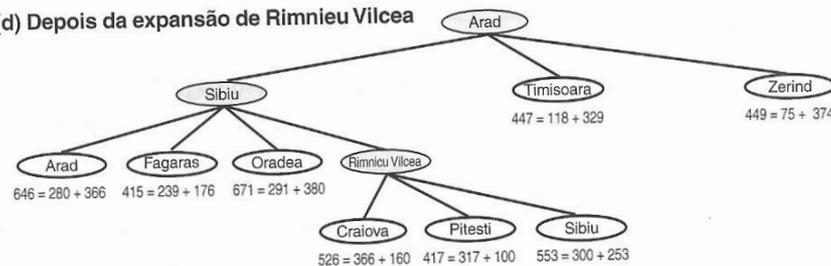
(b) Depois da expansão de Arad



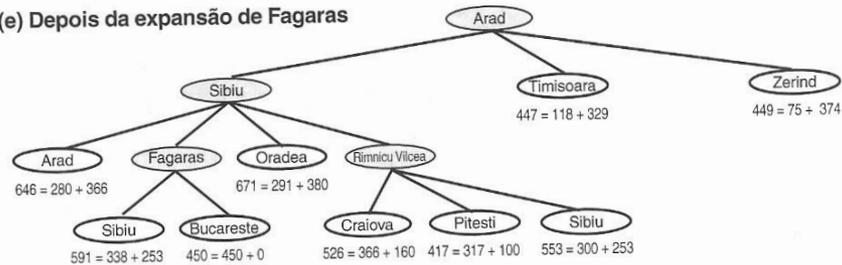
(c) Depois da expansão de Sibiu



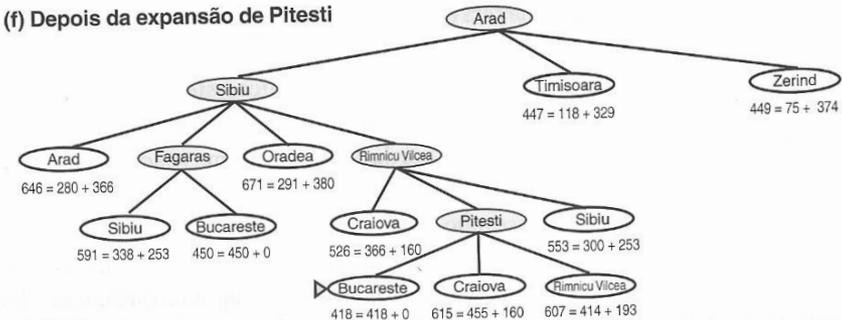
(d) Depois da expansão de Rimnicu Vilcea

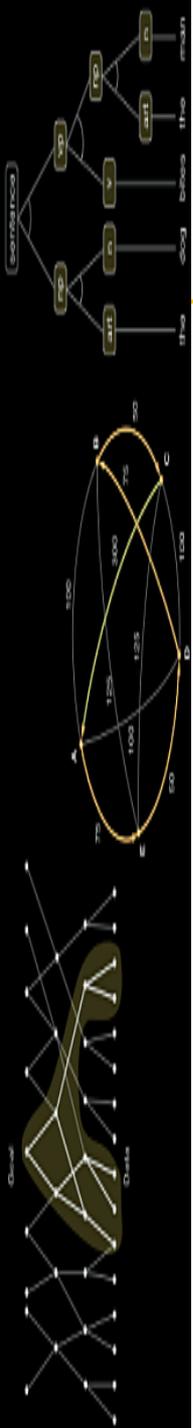


(e) Depois da expansão de Fagaras



(f) Depois da expansão de Pitesti

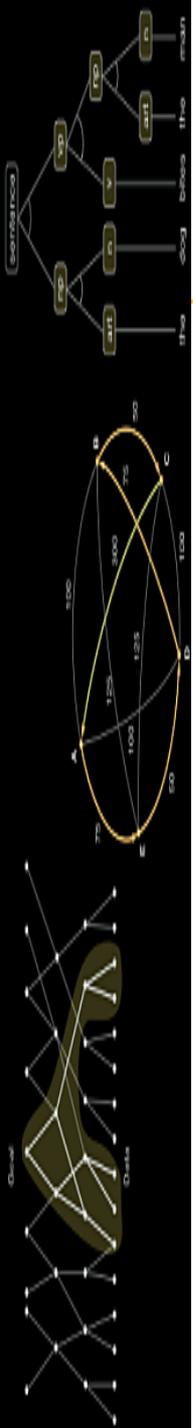




Busca Heurística

Desempenho do A^*

- A análise do caráter ótimo de A^* é direta se for usada com BUSCA-EM-ÁRVORE: A^* será ótima se $h(n)$ for uma **heurística admissível**.
- Conseqüência mais importante da **consistência** (também chamada **monotonicidade**) é: A^* usando BUSCA-EM-GRAFO é ótima se $h(n)$ é **consistente**.



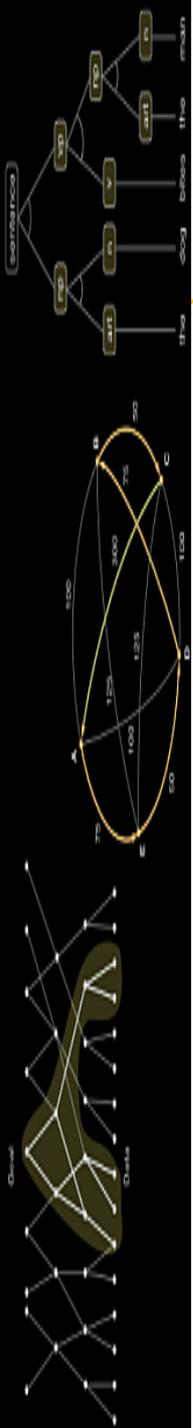
Busca Heurística

Desempenho do A*

- **A* é completa e ótima se $h(n)$ for admissível ou consistente**
 - **h admissível:** nunca superestima o custo de atingir a meta
 - **h consistente (ou monotônica):**

$$h(n) \leq c(n, a, n') + h(n'), \forall n, n'$$

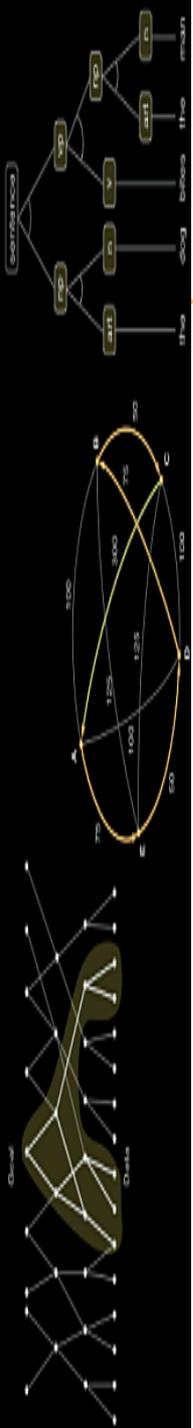
- n' é sucessor de n , gerado pela ação a ;
- $c(n, a, n')$ é o custo de sair de n e atingir n' .
- Se h é consistente, os valores de $f(n)$ ao longo de qualquer caminho são não-decrescentes.



Busca Heurística

Desempenho do A*

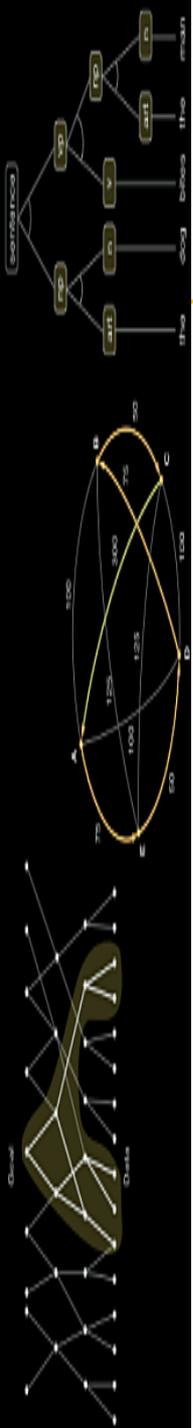
- ❑ **A* é otimamente eficiente:** nenhum outro algoritmo ótimo garante expandir menos nós que A*.
- ❑ Infelizmente há, na maioria das vezes, crescimento exponencial do número de nós com o comprimento da solução (complexidade temporal).
- ❑ O maior problema é a complexidade espacial: **A* armazena todos os nós gerados!**
- ❑ Assim, A* não é aplicável em muitos problemas de grande escala. Usa-se variantes que encontram soluções **subótimas**.



Busca Heurística

Com Memória Limitada

- **IDA* (Iterative Deepening A*)**
 - igual ao aprofundamento iterativo, porém seu limite é dado pela função de avaliação (f), e não pela profundidade (d).
 - necessita de menos memória do que A*
- **SMA* (Simplified Memory-Bounded A*)**
 - O número de nós guardados em memória é fixado previamente.



Busca Heurística

- ❑ **Solução de problemas usando técnicas de *busca heurística*:**
 - dificuldades em definir e usar a *função de avaliação*
 - não consideram conhecimento genérico do mundo (ou “*senso comum*”)
- ❑ **Função de avaliação: compromisso (conflito) entre**
 - tempo gasto na seleção de um nó e
 - redução do espaço de busca
- ❑ **Achar o melhor nó a ser expandido a cada passo pode ser tão difícil quanto o problema da busca em geral.**