Universidade Federal do Sul e Sudeste do Pará
Faculdade de Computação e Engenharia Elétrica
Inteligência Artificial

Capítulo 2 – Metodologias de Busca em IA (Parte 2)

Prof. Dr. Elton Alves

Propriedades dos Métodos de Buscas

1. **Complexidade:**
   - **Quanto tempo e memória o método usa.**
2. **Completude:**
   - **Um método completo é aquele que garante encontrar uma meta, se existir.**
   - **Busca em largura – SIM**
   - **Busca em profundidade - NÃO**
3. **Quanto a ser ótimo:**
   - **Um método é ideal (ou admissível) se for garantido que você encontrará o melhor caminho para a meta.**

Propriedades dos Métodos de Buscas

4. **Irrevogabilidade:**

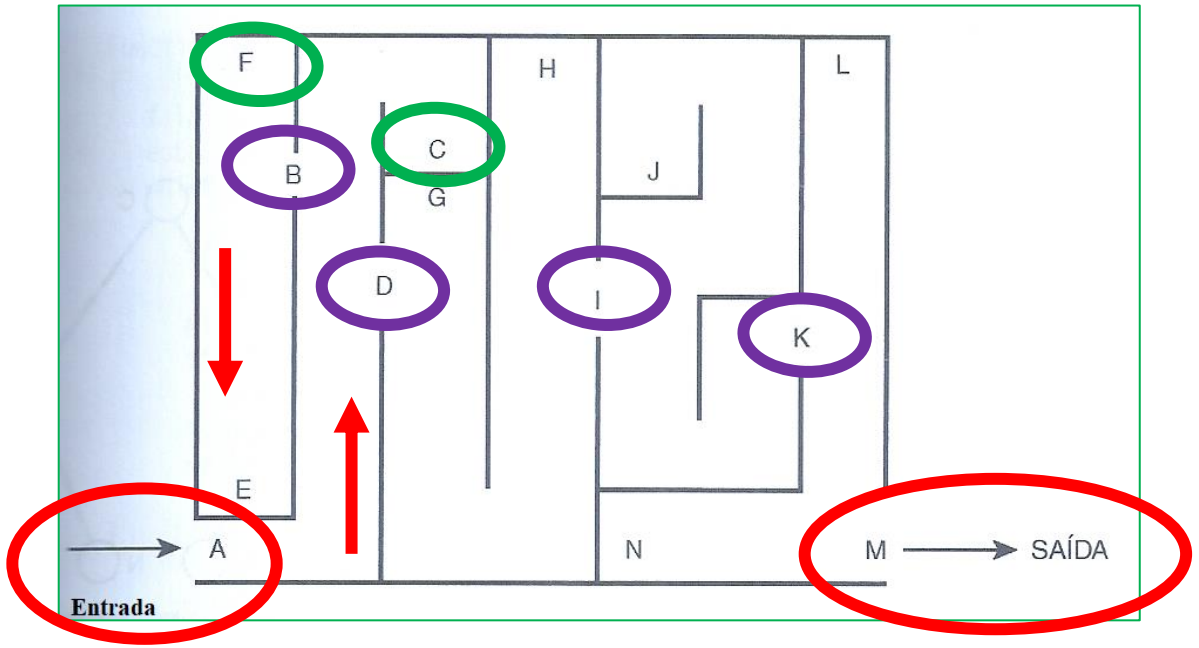
- **Um método é irrevogável se, como escalar montanhas, nunca voltar atrás.**

5. Admissibilidade:

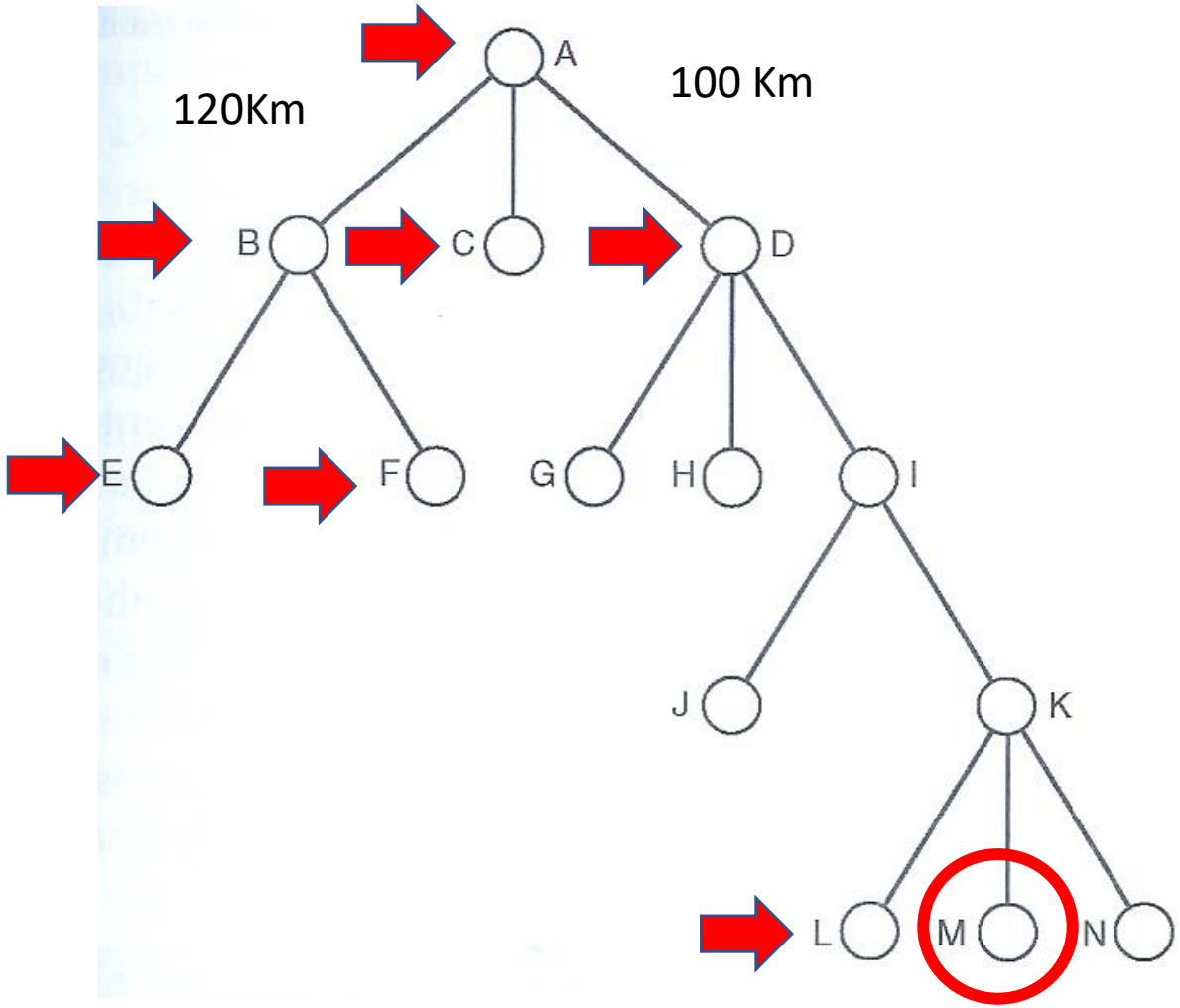
- **Algoritmo encontre a melhor solução.**

Exemplo 1: Percorrendo um Labirinto

- **Pode ser usado a busca em profundidade (seguir com a mão pela esquerda)**
- **A é entrada.**
- **M é a saída.**
- **C, E, F, G, H, J, L e N são sem saída**
- **B, D, I e K são pontos no labirinto onde se pode escolher qual próxima direção seguir.**
- **OBS: caminho com a mão pelo lado esquerdo: A, B, E, F, C, D, G, H, I, J K, L, M.**



Exemplo 1: Percorrendo um Labirinto



Representação em árvore de busca do labirinto

Exemplo 2 – Comprando um presente

- **SITUAÇÃO: Procurar um presente para a namorada em várias lojas, cada uma com vários andares e cada andar com várias lojas de departamentos.**
- **Como seria o emprego da busca em largura e busca em profundidade?**
- **Qual seria a abordagem mais natural de emprego?**

Usando Heurística na Busca

- **Busca em largura e profundidade são conhecidos como métodos de força bruta.**
- **Não empregam nenhum conhecimento especial sobre as árvores de busca que estão examinando.**
- **Não há informação adicional disponível que possa ser utilizada para direcionar a busca de maneira melhor.**
- **Informações heurísticas auxiliam na solução de problemas.**

• ***Função heurística*** **(*f*): função de avaliação para cada nó para estimar a distância entre nó e o objetivo.**

$f, se\ f(m) < f(n)$ **Caminho ótimo: menor *f***

Métodos Informados e Não Informados

➢ **Método heurístico informado:**

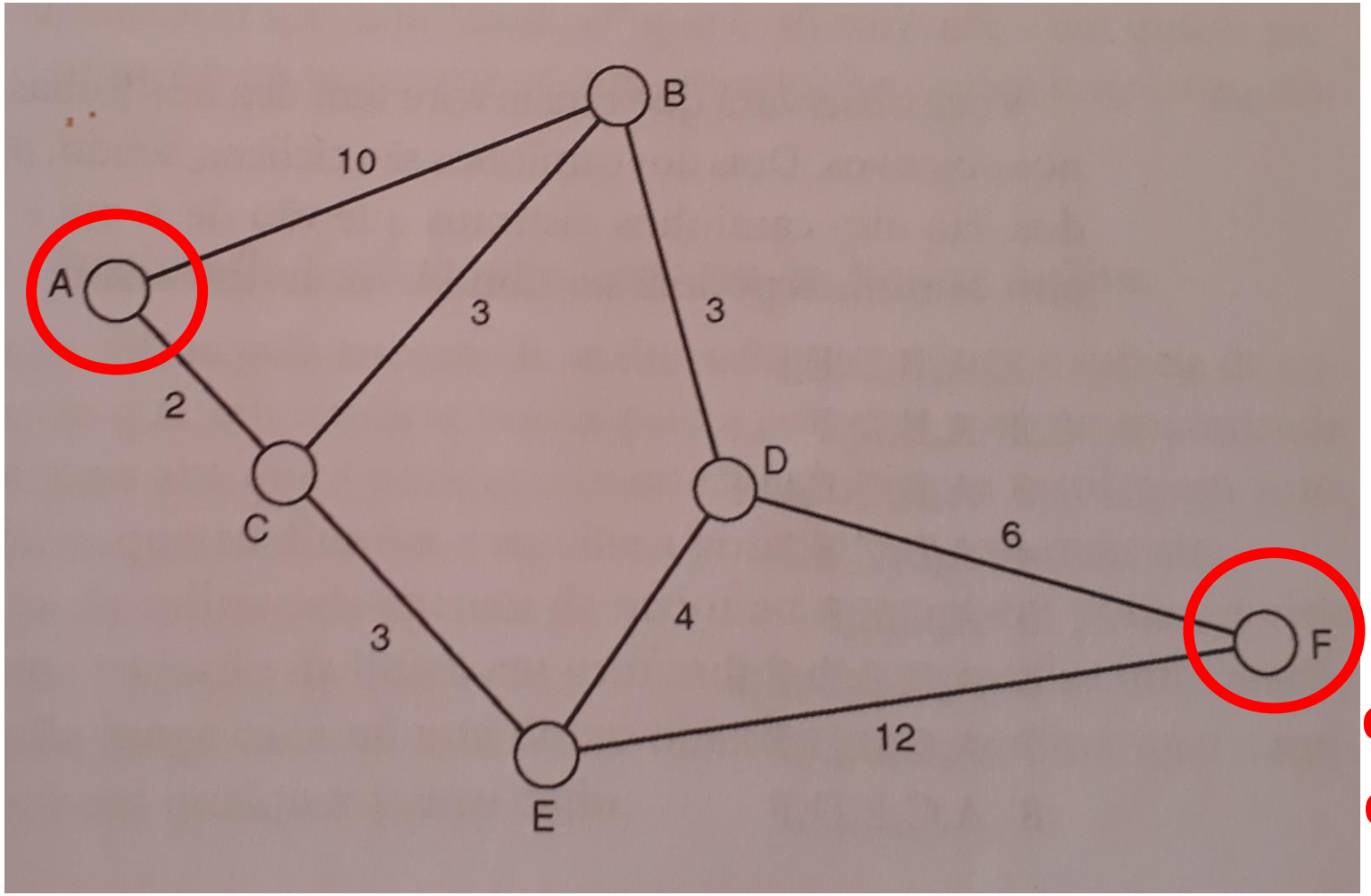
- **Usa informação adicional sobre o nó que ainda não tenham sido explorados para decidir quais nós examinar a seguir.**
- **Métodos mais eficientes.**
- **Exemplo: Busca pelo primeiro melhor.**

➢ **Método heurístico não informado ou cego:**

- **Método que não usa informação adicional.**
- **Exemplo: Busca em largura e profunda.**

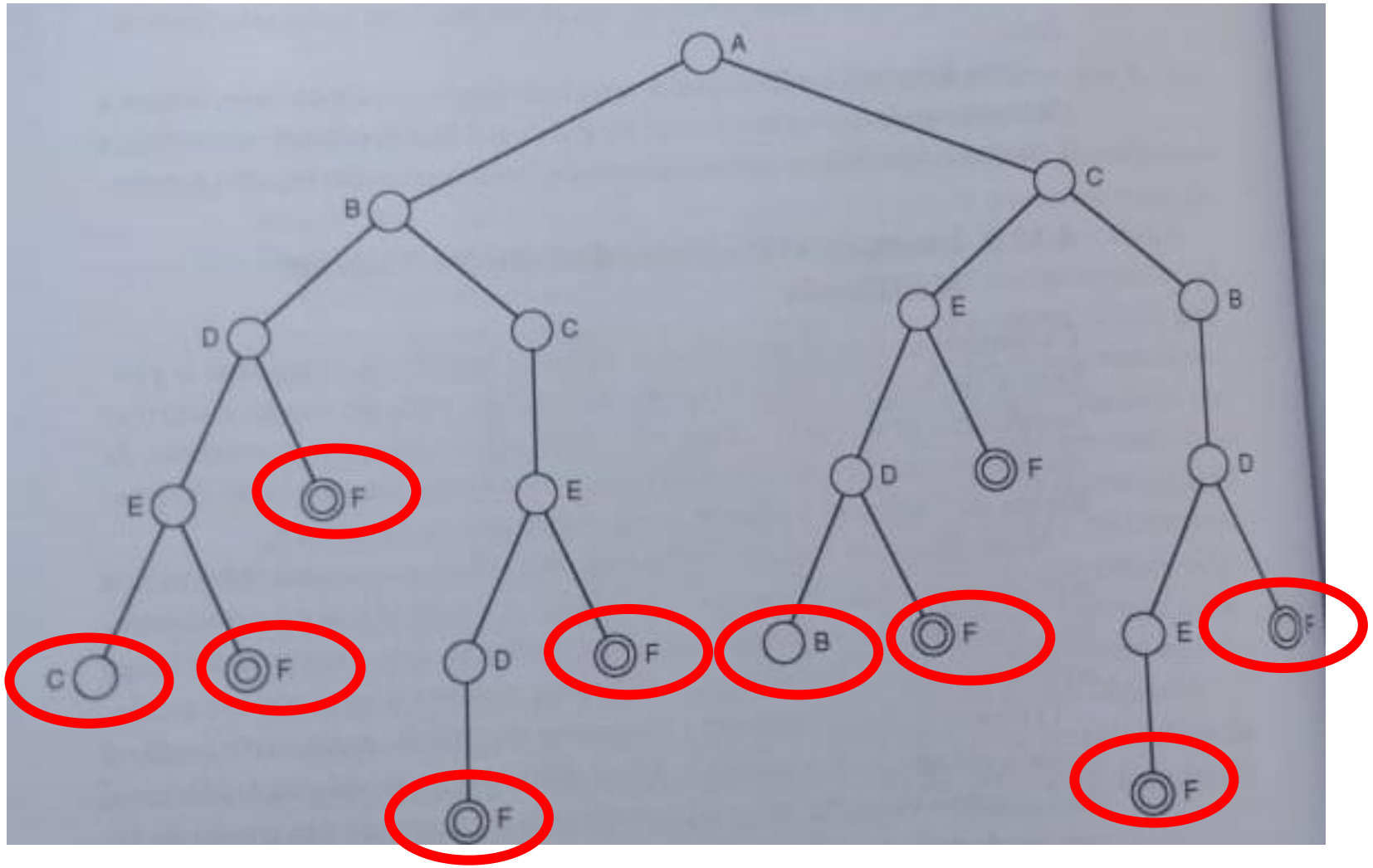
➢ **OBS: quanto mais informado um método de busca for, mais eficiente será a busca por ele realizada.**

Exemplo de heurística – O problema do caixeiro viajante modificado



Objetivo: encontrar o caminho mais curto possível

Espaço de busca



1. A, B, D, E, F
2. A, B, D, F
3. A, B, C, E, D, F
4. A, B, C, E, F
5. A, C, E, F
6. A, C, B, D, E, F
7. A, C, B, D, F
8. A, C, E, D, F

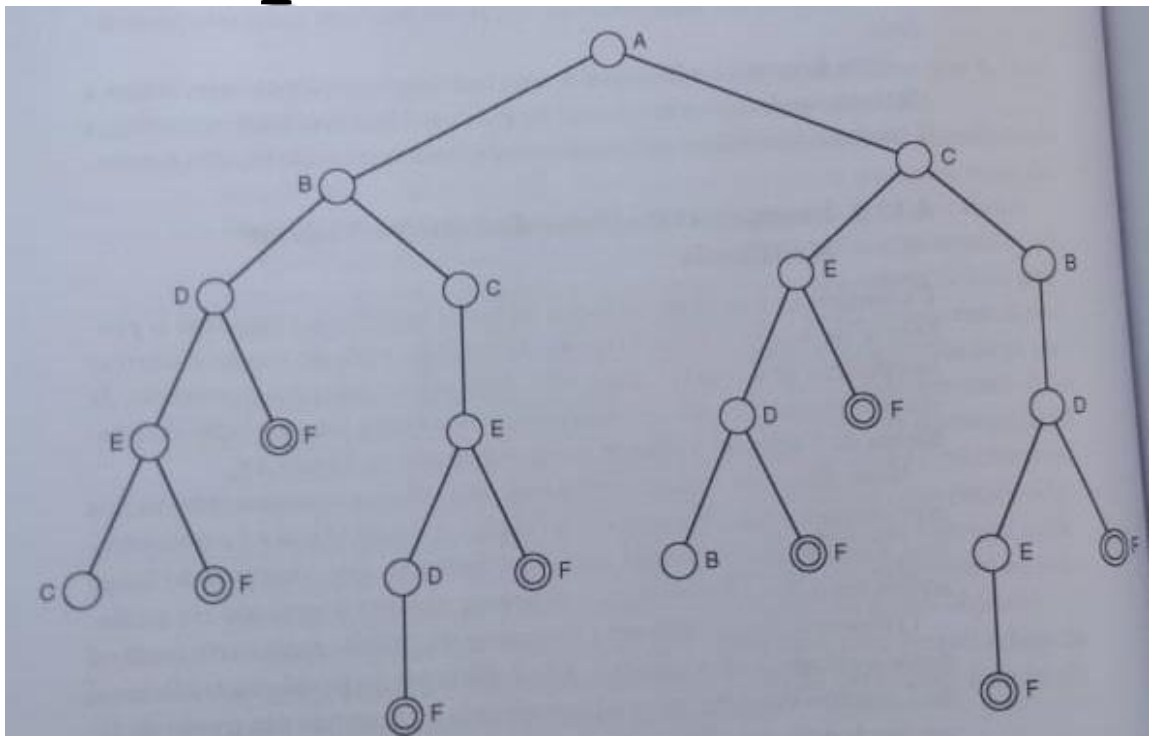
Subida da Colina

- É um exemplo de método informado.
- **Exemplo:**
  - Escalada de uma montanha, em dia de neblina, com um altímetro, mas sem mapa (gerar e testar).
  - Achar o topo (altura)
  - Movimento para primeira posição encontrada que seja mais alta que a posição corrente.

Busca pelo Primeiro Melhor

- **Funciona como escalar montanhas (subida da colina).**
- **A fila inteira é ordenada após receber a inserção de novos caminhos, em vez de inserir um conjunto de caminhos ordenados.**
- **A busca pelo primeiro melhor segue o melhor caminho disponível na árvore.**

Busca pelo Primeiro Melhor



PASSO	Estado	Fila	Observações
1	A	(Vazia)	A fila começa vazia e estado inicial é A.
2	A	B,C	Os sucessores do estado corrente, B e C, são inseridos na fila.
	A	B,C	A fila é ordenada, deixando B a frente de C, por ele estar mais perto de F.
3	B	C	
4	B	D,C,C	Os filhos do nó B são inseridos na frente da fila
5	B	D,C,C	A fila é ordenada, ficando D na frente, por estar mais próximo de F.

Identificando caminhos ótimos

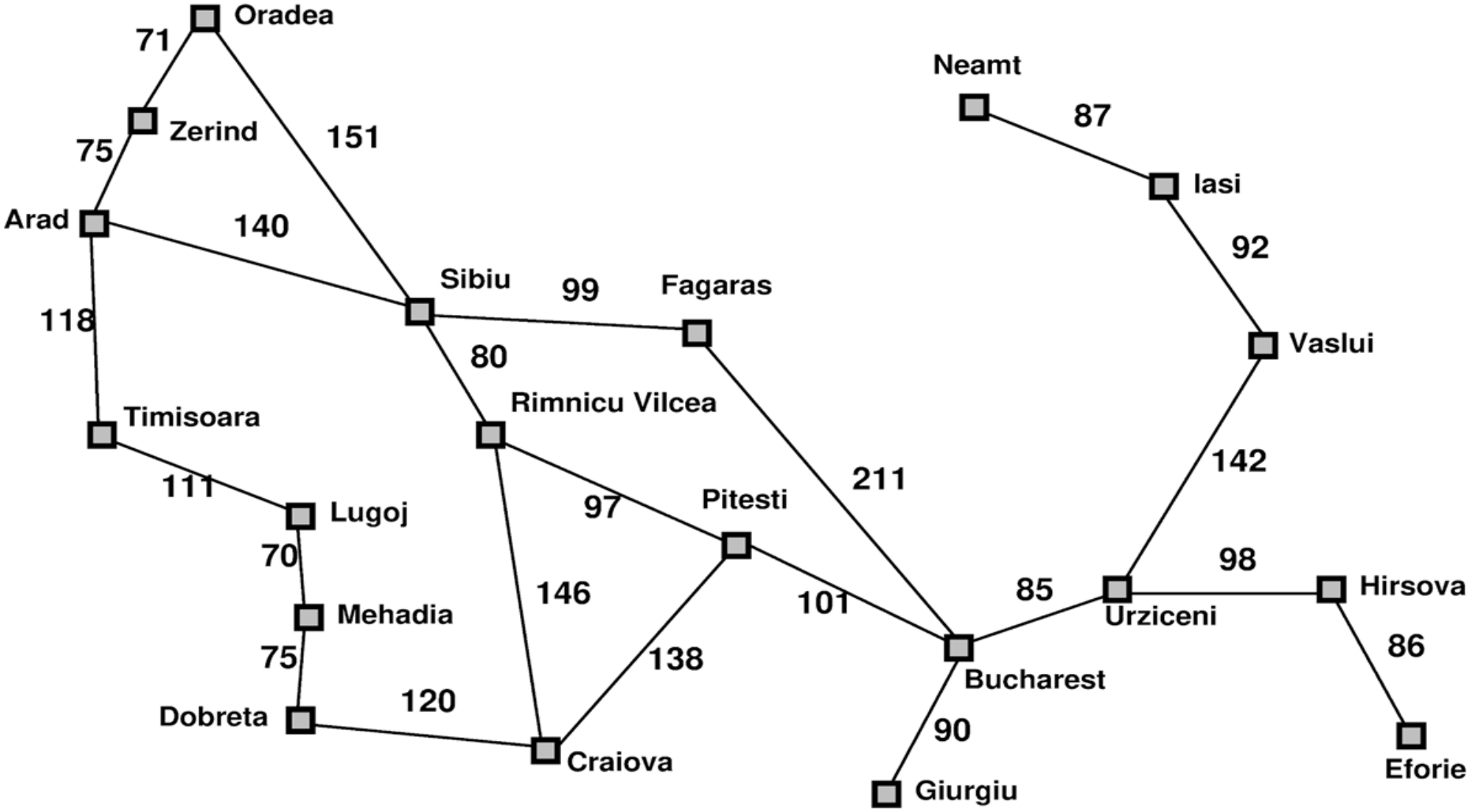
- **O caminho ótimo é aquele que tem o menor custo ou envolve percorrer a distância mais curta.**
- **Exemplos de técnicas utilizadas para identificar o caminho ótimo:**

1. **A***
2. **Busca uniforme**
3. **Busca gulosa**

Algoritmos A*

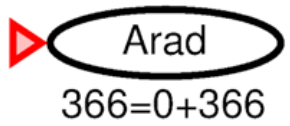
- **Função de avaliação**: $f(n) = g(n) + h(n)$
  - $g(n)$ = custo para alcançar n
  - $h(n)$ = custo estimado de n até o objetivo
  - $f(n)$ = custo total estimado do caminho através de n até o objetivo.

Romênia (Distância Linha Reta-Função heurística)

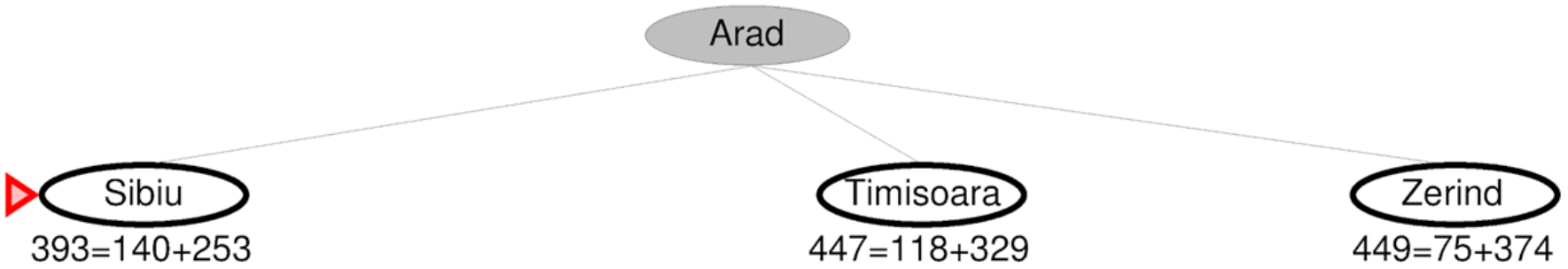


Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

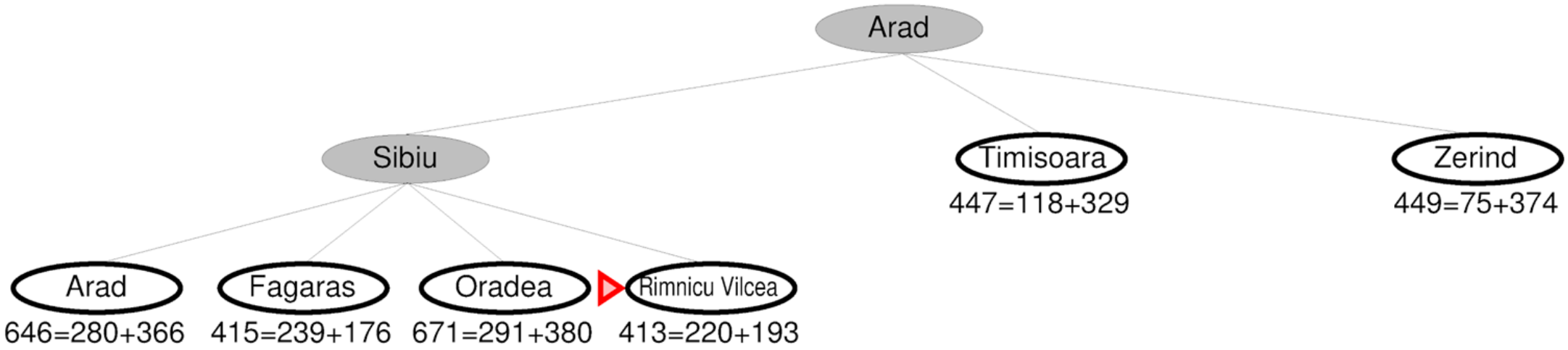
Algoritmos A* (Exemplo)



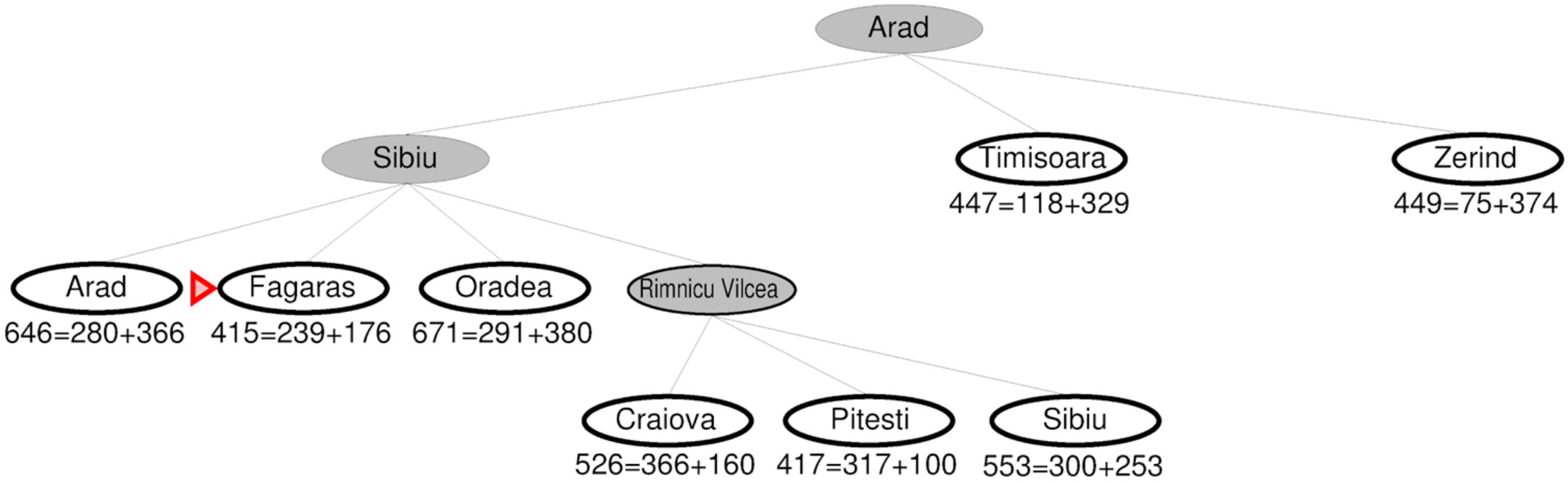
Algoritmos A* (Exemplo)



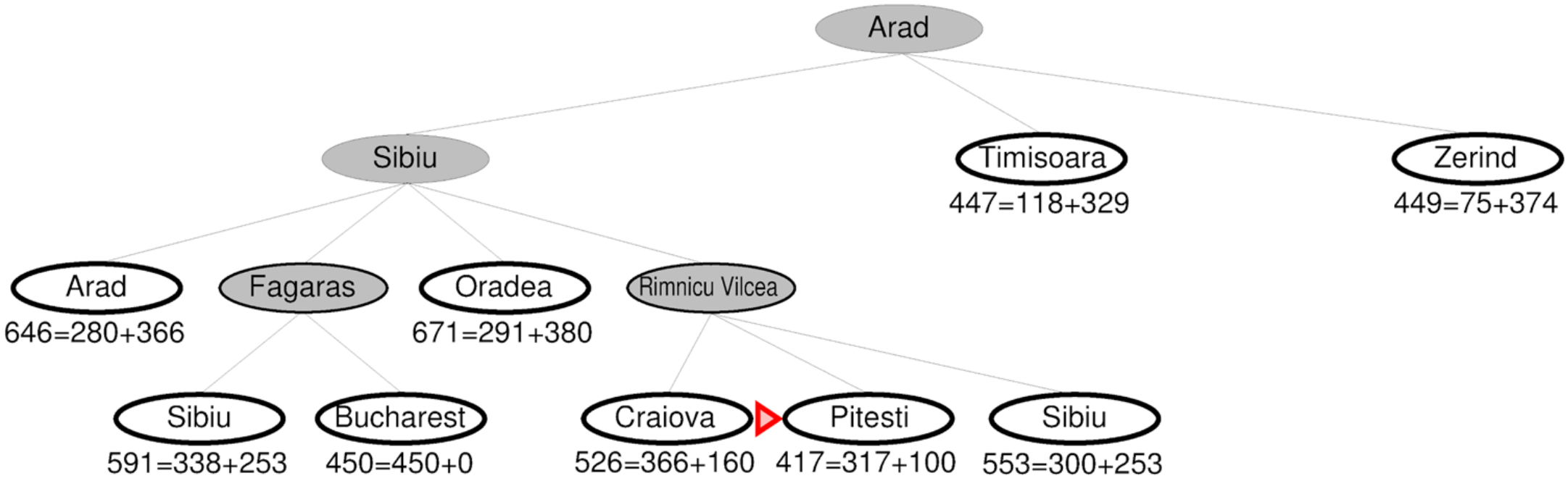
Algoritmos A* (Exemplo)



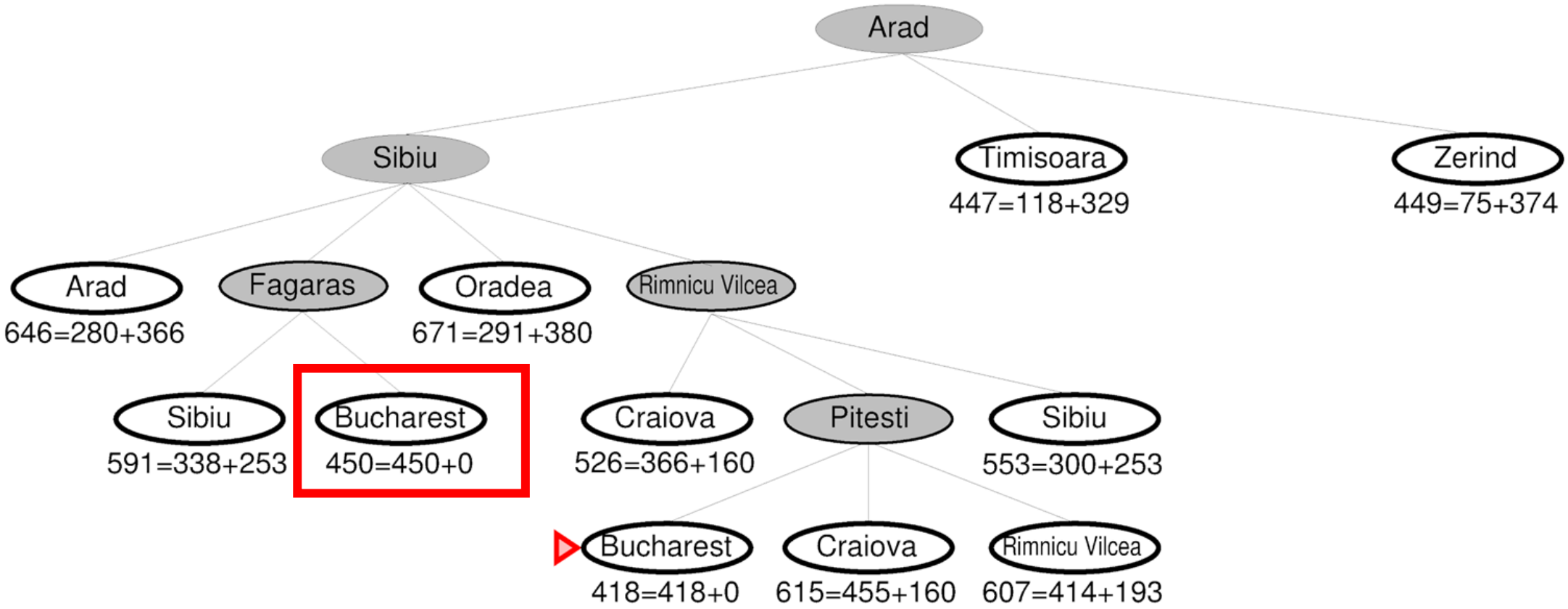
Algoritmos A* (Exemplo)



Algoritmos A* (Exemplo)



Algoritmos A* (Exemplo)



Busca Gulosa

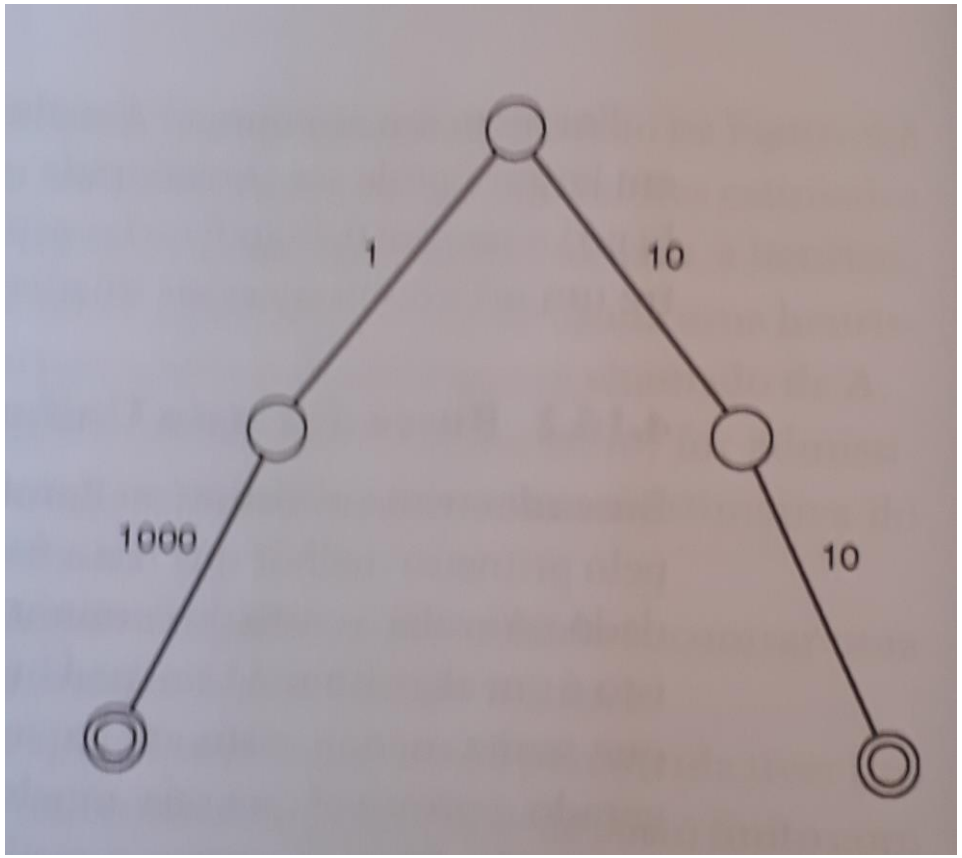
- **Como A *, mas usa:**

-f(node) = h(node) (heurística) = estimativa do custo de *n* até o objetivo.

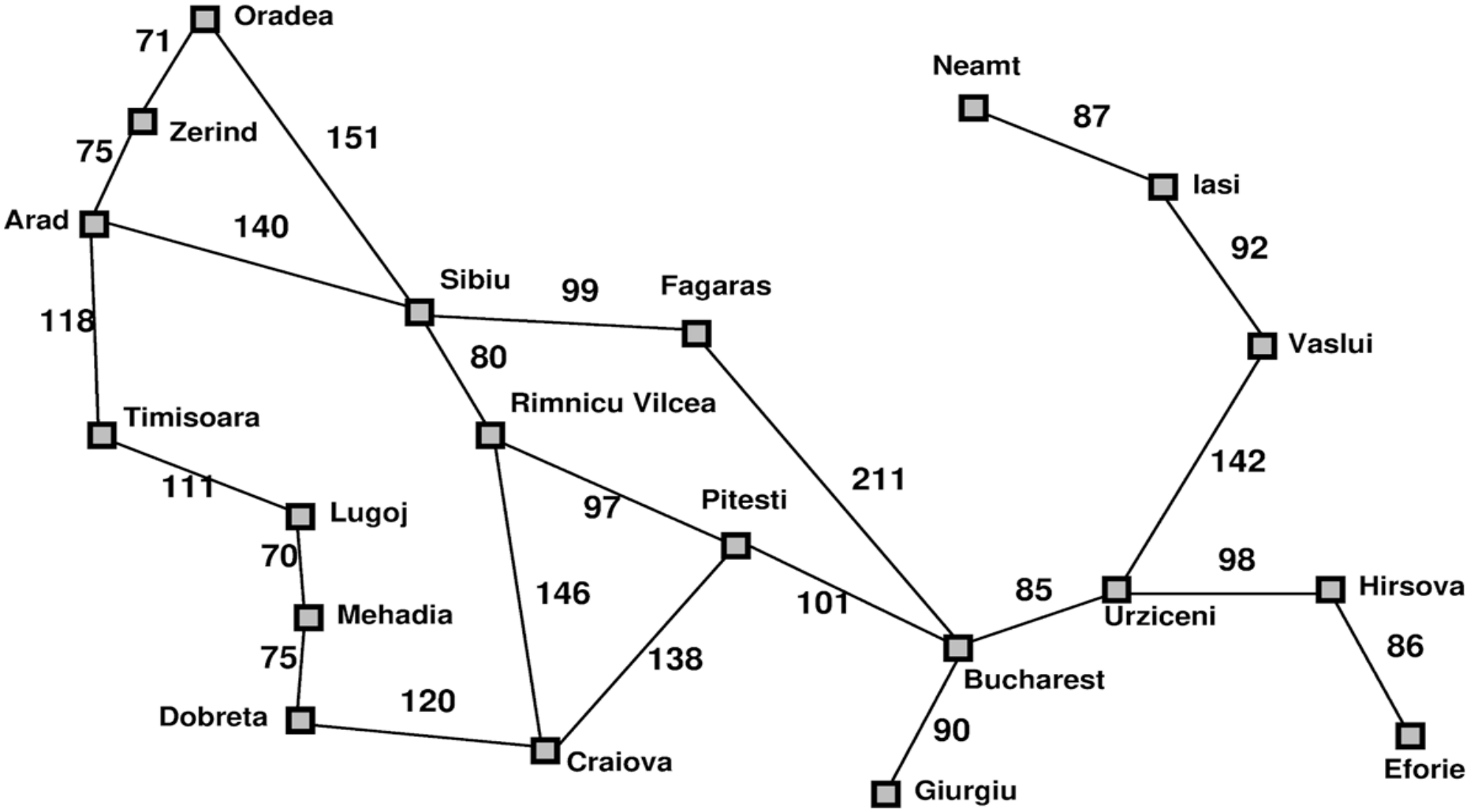
- **Portanto, sempre expande o nó que parece estar mais próximo de uma meta (objetivo), independentemente do que ocorreu antes.**
- **Não é o ideal e não há garantia de encontrar uma solução.**
- **Pode facilmente ser enganado em seguir caminhos ruins.**

Busca Gulosa

- **Uma árvore de busca na qual o método de busca guloso não encontra a melhor solução.**



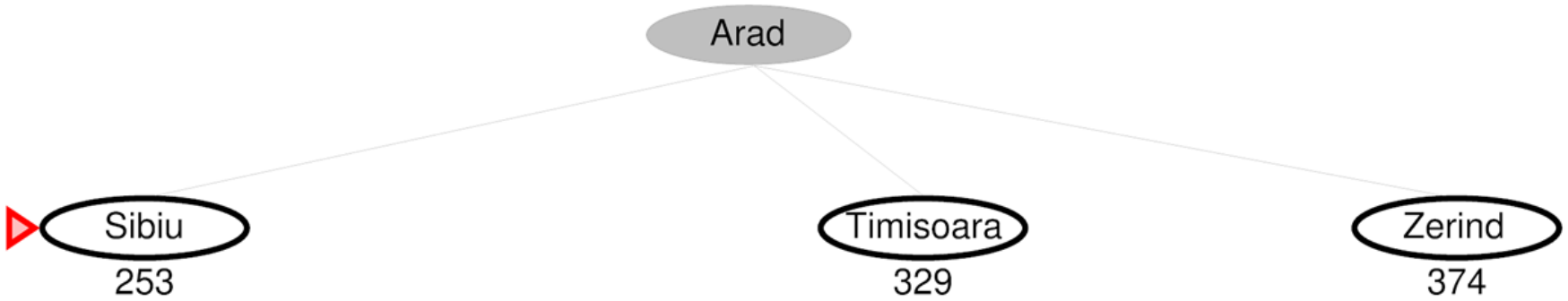
Romênia (Distância Linha Reta-Função heurística)



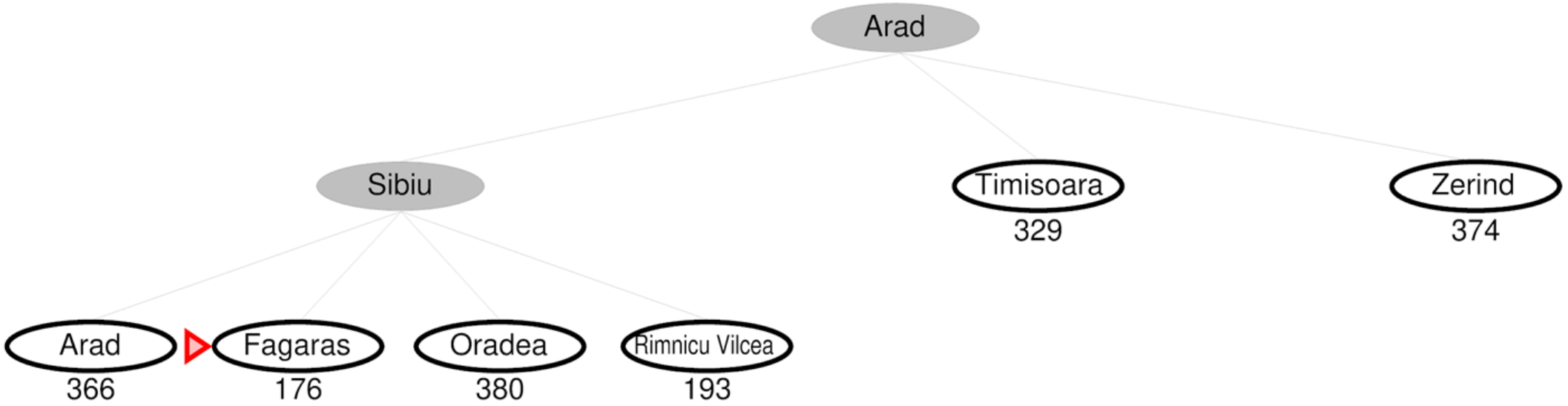
Arad	366
Bucharest	0
Craiova	160
Dobreta	242
Eforie	161
Fagaras	178
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	98
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Busca Gulosa (Exemplo)

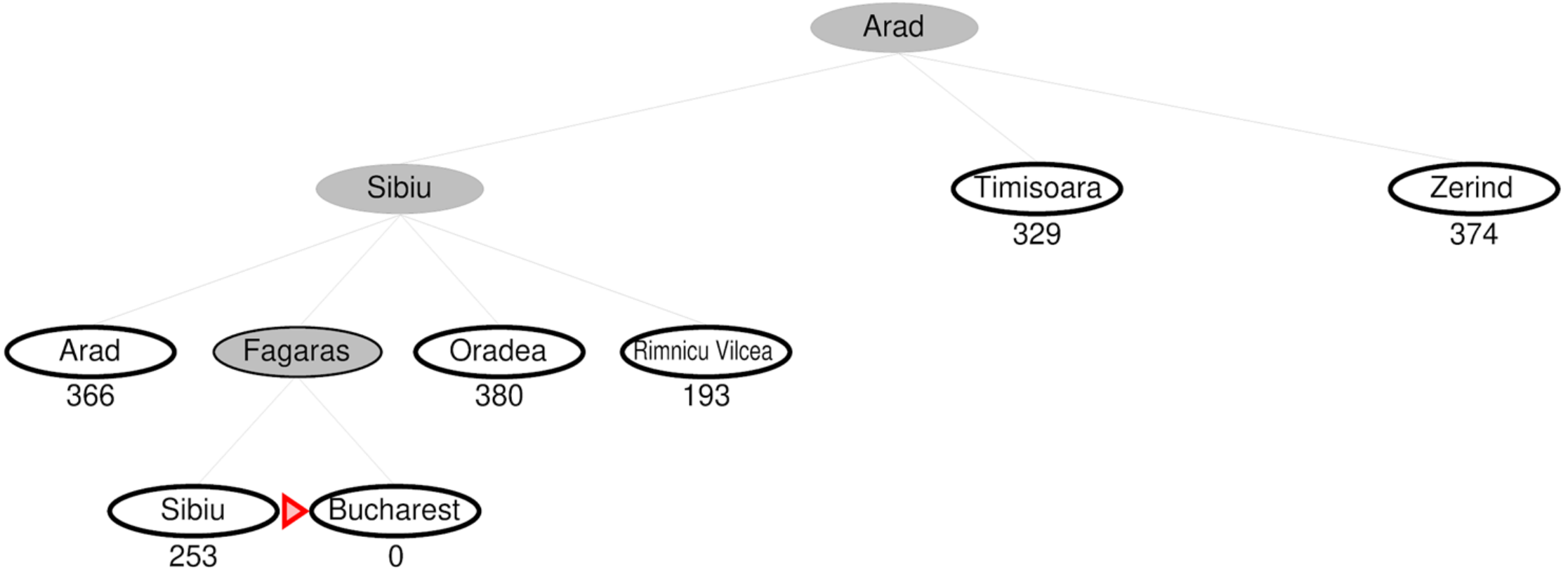
Busca Gulosa (Exemplo)



Busca Gulosa (Exemplo)



Busca Gulosa (Exemplo)



Busca de Custo Uniforme – Ramificação Linear

- **Expande o nó não-expandido que tenha o caminho de custo mais baixo**
- **Como A *, mas usa:**
  - **f(node) = g(node).**
  - **g(nó) é o custo do caminho que leva ao nó.**
- **A cada estágio o caminho com menor custo até então é expandido.**

Busca de Custo Uniforme