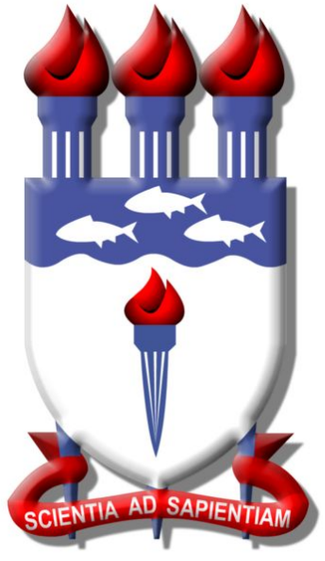


A*

Marcelo Lima e Thiago Miranda

https://github.com/m4rcelolima/P2_Huffman

Motivação

- Pathfinding.

- Expandir Dijkstra.
  - Heurística!

- Qual o melhor caminho entre duas cidades?

Definições

- Lista aberta e fechada.
- Peso nas arestas.
- Valor heurístico.

- f(n) = g(n) + h(n)
  - f(n) = custo estimado da solução de custo mais baixo passando por “n”.
  - g(n) = custo do caminho desde o nó inicial até o nó “n”.
  - h(n) = custo estimado do caminho de custo mais baixo desde “n” até o objetivo.

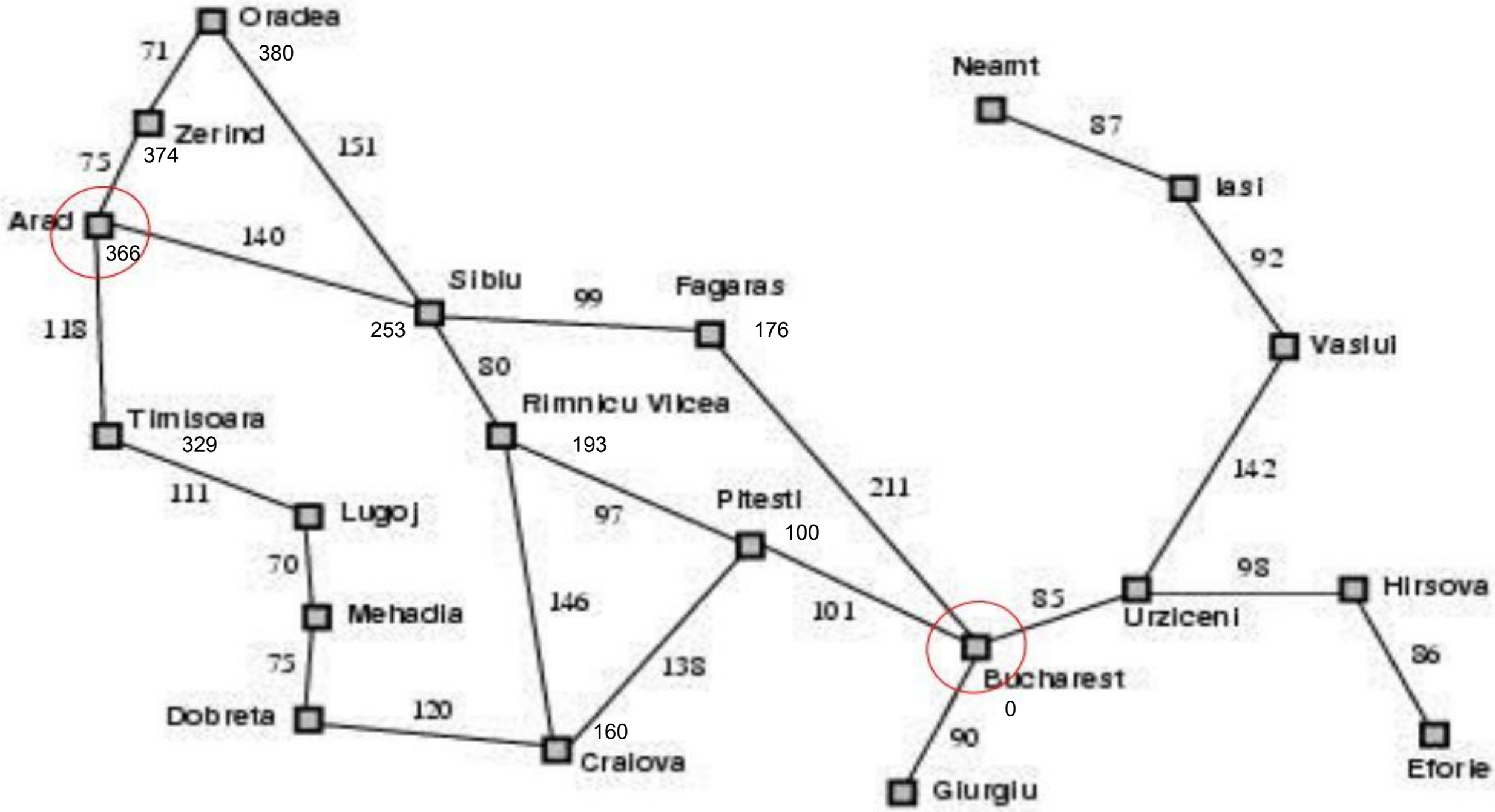
Pseudo-Código

```
ListaAberta; // Lista de nós que serão analisados
ListaFechada; // Lista de nós que já foram analisados
// Adicione o nó INÍCIO na ListaAberta;
insere(ListaAberta, INÍCIO);
while(1){
    ATUAL = getMin(ListaAberta);//nó da ListaAberta de menor  F;
        remove(ListaAberta, ATUAL);// remova ATUAL da ListaAberta;
    insere(ListaFechada, ATUAL);insere ATUAL na ListaFechada;

    if(ATUAL == FIM){ // Encontramos um caminho
        printf("Caminho encontrado");
        break;
}
    if(isEmpty(ListaAberta)){ // Caminho não encontrado
        printf("Caminho não encontrado");
    break;
}
```

```
  for( i = 0; i < Nvizinhos; i++){
      if(isListaFechada(VIZINHO) || isBlock(VIZINHO))
      break;
      } else if(!isListaAberta(VIZINHO) ||
newPathMin(ATUAL,VIZINHO)){ //se não estiver na lista ou o
novo caminho para o vizinho for menor
            calcule(G, H, F, VIZINHO);
            VIZINHO->parent = ATUAL;
            if(!isListaAberta(VIZINHO)){
                insere(ListaAberta, VIZINHO);
            }
        }
} // Fim do While
```

Exemplo: Distância entre as cidades

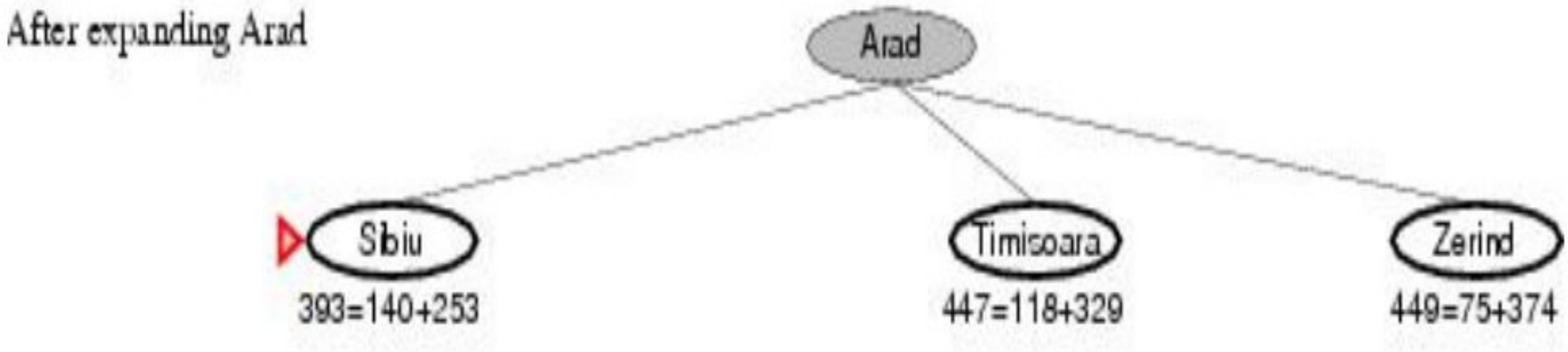


Exemplo: Distância entre as cidades

(a) The initial state

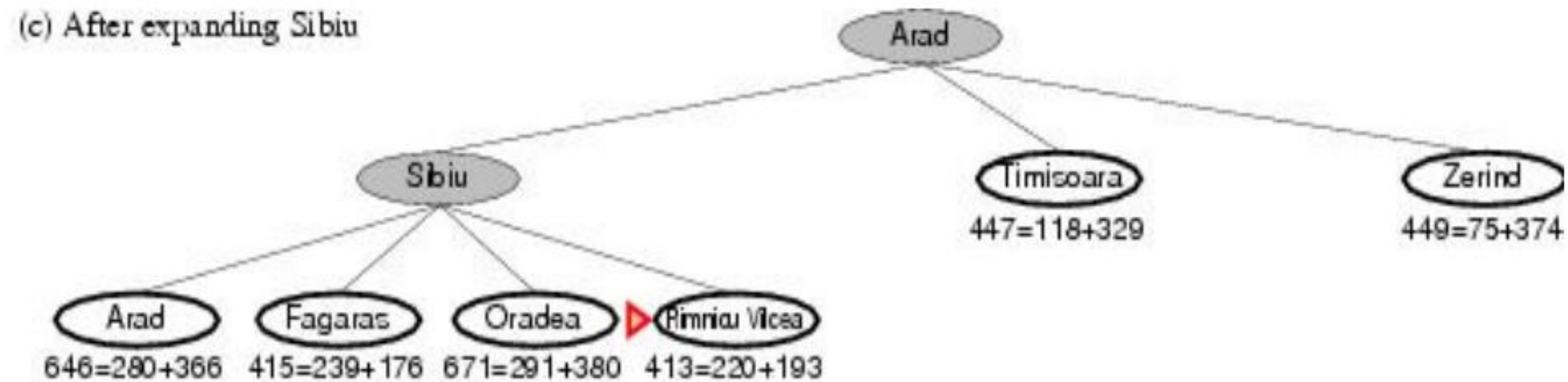
- Encontrar Bucareste partindo de Arad:
  - f(arad)= 0 + h(arad)= 0 + 366 = 366
    - Onde o valor da heurística é dado pelo caminho em linha reta entre as duas cidades

Exemplo: Distância entre as cidades



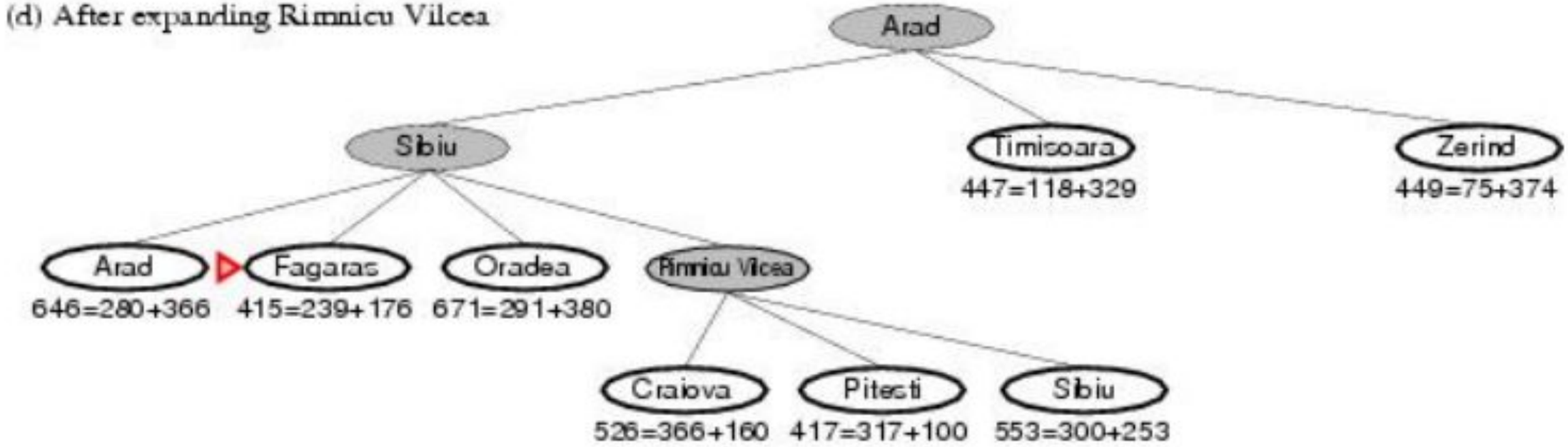
- Expandindo Arad e determinando f(n) para cada nó:
  - f(Sibiu) = c(Arad,Sibiu) + h(sibiu) = 140 + 253 = 393
  - f(Timisoara) = c(Arad,Timisoara) + h(Timisoara) = 118 + 329 = 447
  - f(Zerind) = c(Arad,Zerind) + h(Zerind) = 75 + 374 = 449
- Melhor escolha é Sibiu

Exemplo: Distância entre as cidades



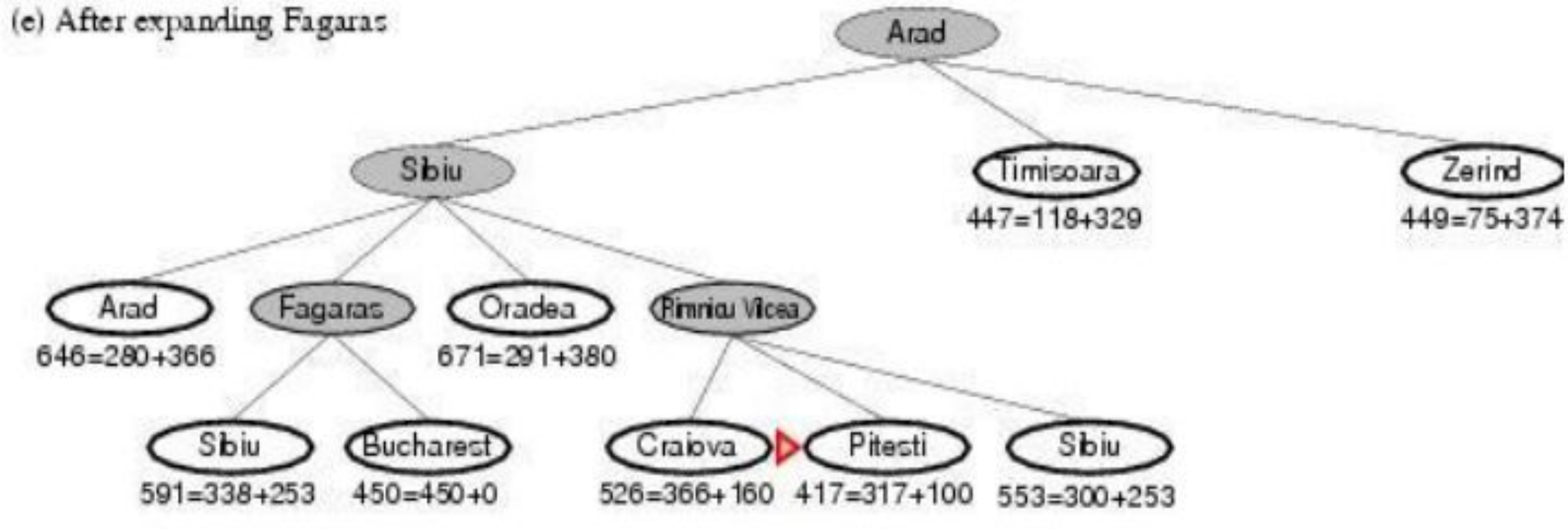
- Expandindo Sibiu e determinado f(n) para cada nó:
  - f(Arad) = g(Arad) + h(Arad) = 280 + 366 = 646
  - f(Fagaras) = g(Fagaras) + h(Fagaras) = 239 + 176 = 415
  - f(Oradea) = g(Oradea) + h(Oradea) = 291 + 380 = 671
  - f(Riminicu Vilcea) = g(Rimnicu Vilcea) + h(Rimnicu Vilcea) = 220 + 193 = 413
- Melhor escolha é Rimnicu Vilcea

Exemplo: Distância entre as cidades



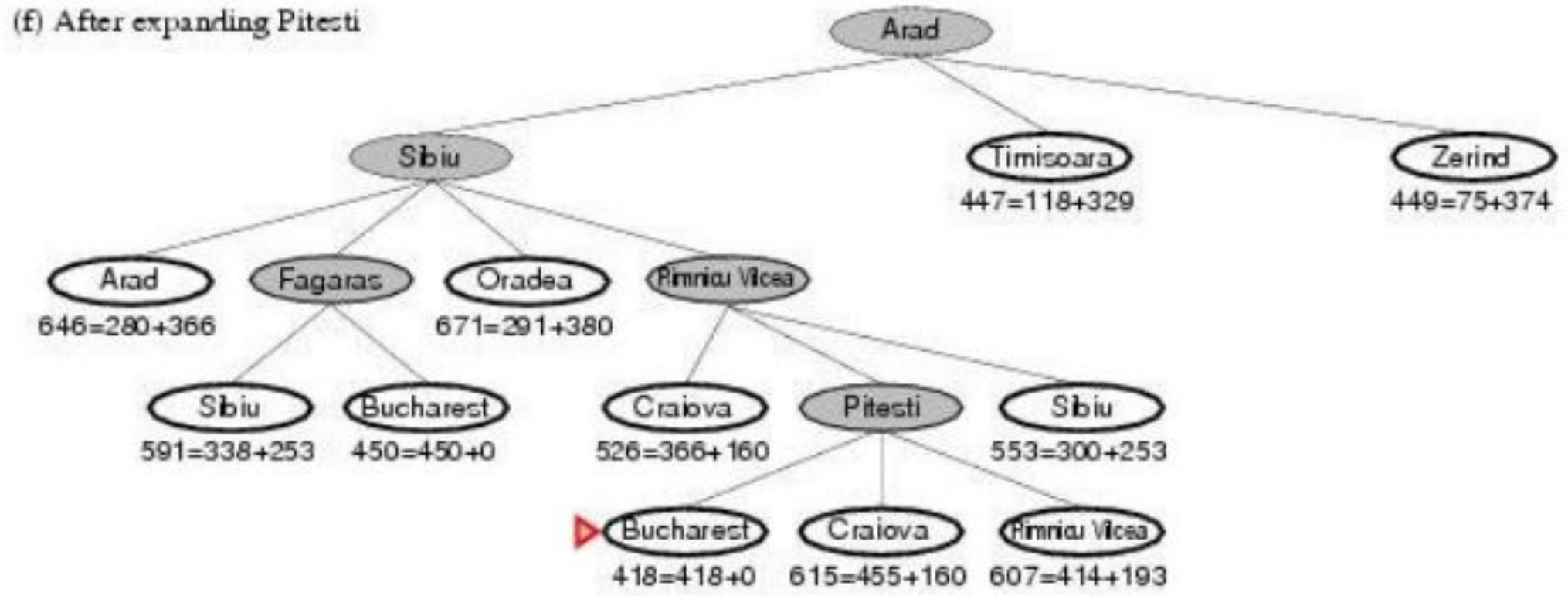
- Expandindo Rimnicu Vilcea:
  - f(Craiova) = g(Craiova) + h(Craiova) = 366 + 160 = 526
  - f(Pitesti) = g(Pitesti) + h(Pitesti) = 317 + 100 = 417
  - f(Sibiu) = g(Sibiu) + h(Sibiu) = 300 + 253 = 553
- Melhor escolha é Fagaras.

Exemplo: Distância entre as cidades

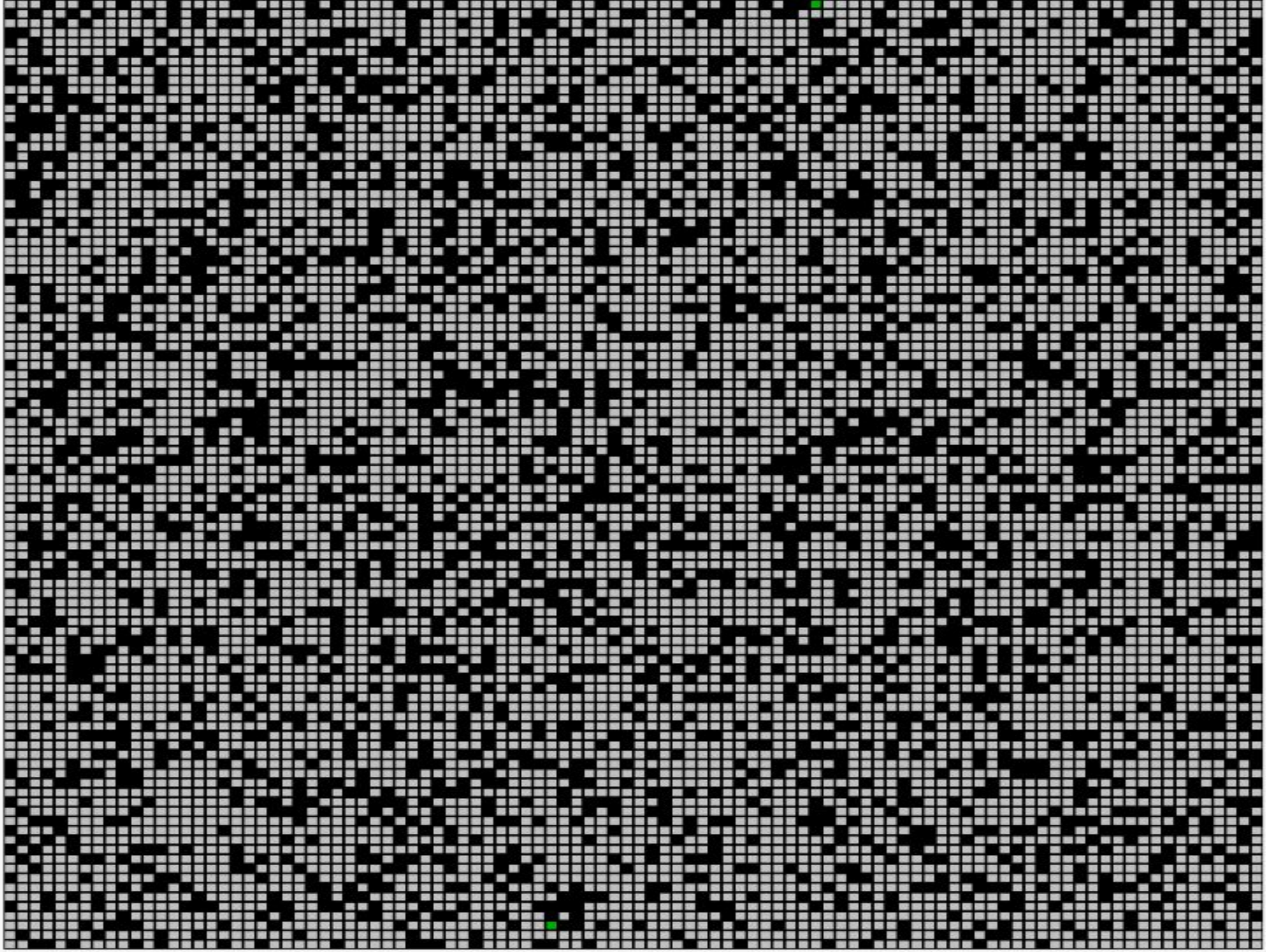


- Expandindo Fagaras:
  - f(Sibiu) = g(Sibiu) + h(Sibiu) = 338 +253 = 591
  - f(Bucareste) = g(Bucareste) + h(Bucareste) = 450 + 0 = 450
- Melhor escolha é Pitesti

Exemplo: Distância entre as cidades



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



Complexidade

- Depende da Heurística.
- O(b^n)
  - b = fator de ramificação.
  - n = número de nós expandidos.
  - Exponencial!
  - Melhor a heurística, menor b.

- |h(x) - h*(x)| = O(log(h*(x)))
  - Espaço de procura é uma árvore.
  - Polinomial!

De volta à Motivação…

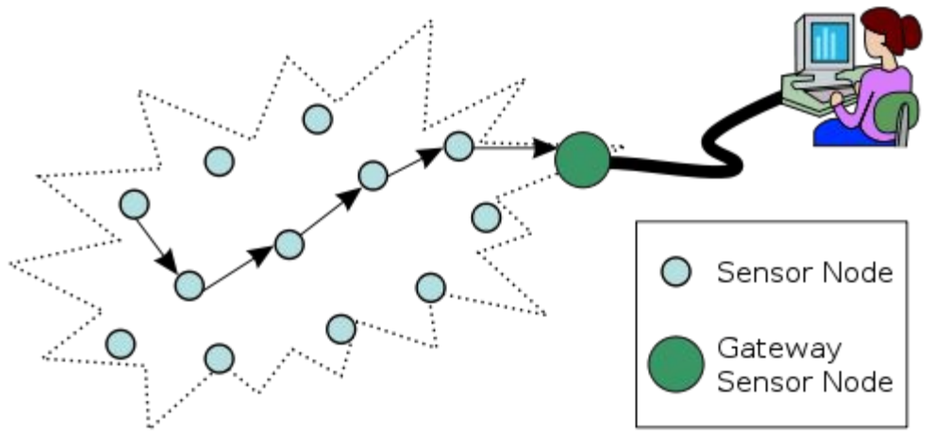
- O algoritmo encontrou uma solução admissível, pois h(n)>=h*(n).

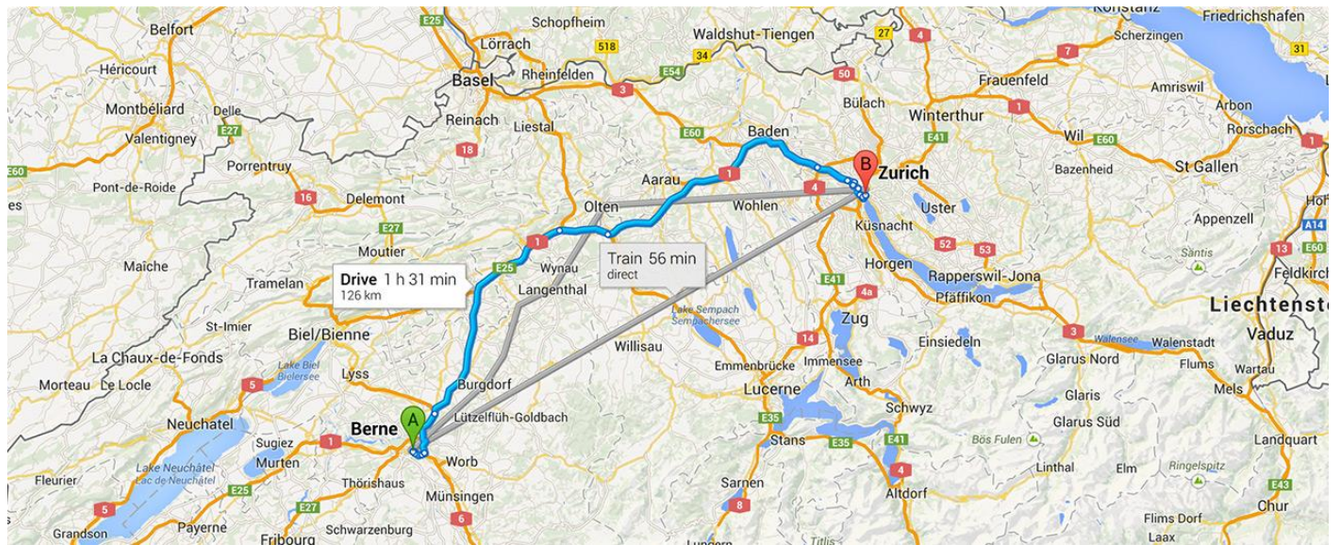
- Heurística adicionada a Dijkstra

- Solução:
  - Completa.
  - Ótima.

Aplicações

- https://www.youtube.com/watch?v=hmx1sx6ezQA
- Many others!





Limitações

- Depende do cálculo da heurística.
- Não é prática para problemas de grande escala.
- Em geral, esgota o espaço antes de esgotar o tempo.
- Usualmente perde para algoritmos que pré-processam o grafo

Referências

- https://en.wikipedia.org/wiki/A*_search_algorithm

- http://theory.stanford.edu/~amitp/GameProgramming/AStarComparison.html

- https://www.youtube.com/watch?v=-L-WgKMFuhE

- https://www.youtube.com/watch?v=KNXfSOx4eEE