

# Inteligencia Artificial

PhD Carola Figueroa Flores  
Académica del DCCTI-UBB

Noviembre 2021

# Búsqueda Informada

## Parte II



# Implementación: búsqueda genérica basada en árbol

- Un **estado** es una representación de una configuración física
- Un **nodo** es una estructura de datos que constituye parte de una búsqueda vía árboles incluyendo **estado, nodo padre, acción, costo de ruta  $g(x)$ , profundidad**
- **Fringe** es una cola que tiene nodos sucesores no explorados
- Una función **Expand** crea nuevos nodos, llenando los varios campos y usando la función **SuccessorFn** del problema para crear los estados correspondientes.



# Implementación: búsqueda genérica basada en árbol (cont)

```
function TREE-SEARCH(problem, fringe) returns a solution, or failure  
  fringe  $\leftarrow$  INSERT(MAKE-NODE(INITIAL-STATE[problem]), fringe)  
  loop do  
    if fringe is empty then return failure  
    node  $\leftarrow$  REMOVE-FRONT(fringe)  
    if GOAL-TEST[problem](STATE[node]) then return SOLUTION(node)  
    fringe  $\leftarrow$  INSERTALL(EXPAND(node, problem), fringe)
```

---

```
function EXPAND(node, problem) returns a set of nodes  
  successors  $\leftarrow$  the empty set  
  for each action, result in SUCCESSOR-FN[problem](STATE[node]) do  
    s  $\leftarrow$  a new NODE  
    PARENT-NODE[s]  $\leftarrow$  node; ACTION[s]  $\leftarrow$  action; STATE[s]  $\leftarrow$  result  
    PATH-COST[s]  $\leftarrow$  PATH-COST[node] + STEP-COST(node, action, s)  
    DEPTH[s]  $\leftarrow$  DEPTH[node] + 1  
    add s to successors  
  return successors
```

# Estrategias de búsqueda

- Una estrategia de búsqueda se define al seleccionar un **orden de expansión de nodos**
- Las estrategias se evalúan de acuerdo a :
  - **completitud**: siempre encuentra una solución si alguna existe?
  - **complejidad temporal**: numero de nodos generados
  - **complejidad espacial**: numero máximo de nodos en memoria
  - **optimalidad**: siempre encuentra una solución de mínimo costo?
- Complejidad de tiempo y espacio se mide en termino de
  - ***b***: máximo factor del numero de ramas del árbol de búsqueda
  - ***d***: profundidad de solución de mínimo costo
  - ***m***: profundidad máxima del espacio de estados (puede ser  $\infty$ )

# Estrategias de búsqueda no informadas

- Estrategias de búsqueda no informadas usan solo la información disponible en la definición del problema
  - Búsqueda al ancho primero (i.e. Breadth-first)
  - Búsqueda de costo uniforme
  - Búsqueda en profundidad primero
  - Búsqueda en profundidad limitada
  - Búsqueda iterativa en profundidad

# Búsqueda informada o heurísticas

---

- ❑ La búsqueda no informada es ineficiente en la mayoría de los casos
- ❑ El propósito de la búsqueda informada es utilizar conocimiento específico del problema para alcanzar el objetivo de manera más eficiente
- ❑ La idea es ser capaces de medir la “calidad” de un estado
- ❑ Eso nos permitirá dirigir la búsqueda por los estados mejores que estarán “más cerca” del objetivo y no seguir estrategias en anchura o profundidad que no tienen en cuenta la calidad de los estados
- ❑ Las estrategias de búsqueda informada son mucho más eficientes que las no informadas

# Función de evaluación

---

- ❑ Función de evaluación  $f(n)$ , mide la calidad de  $n$
- ❑ Un nodo tendrá calidad cuanto menor sea la distancia al objetivo
- ❑  $f(n)$  estima la distancia desde ese nodo  $n$  a un nodo objetivo
- ❑ Las búsquedas informadas expanden primero los nodos que están más cerca del objetivo; i.e. aquellos en los que la función  $f(n)$  asigna un menor valor



# Búsqueda primero el mejor

---

- ❑ Idea: usar la función de evaluación  $f(n)$  para cada nodo, de modo que se pueda estimar su “deseabilidad” y sea el nodo más deseable que no ha sido expandido el que se elija para expandir
- ❑ Implementación: los nodos en la frontera deben ordenarse de forma decreciente con respecto a su deseabilidad
- ❑ Casos especiales
  - Búsqueda voraz primero el mejor
  - $A^*$

# Búsqueda voraz primero el mejor

---

- ❑ Expande el nodo más cercano al objetivo, asumiendo que probablemente conduzca más rápidamente a la solución.
- ❑ La función de evaluación  $f(n)$  es la función heurística  $h(n)$

$$f(n) = h(n)$$

donde  $h(n)$  = costo estimado del camino más barato desde el nodo  $n$  hasta el objetivo

# Búsqueda voraz primero el mejor

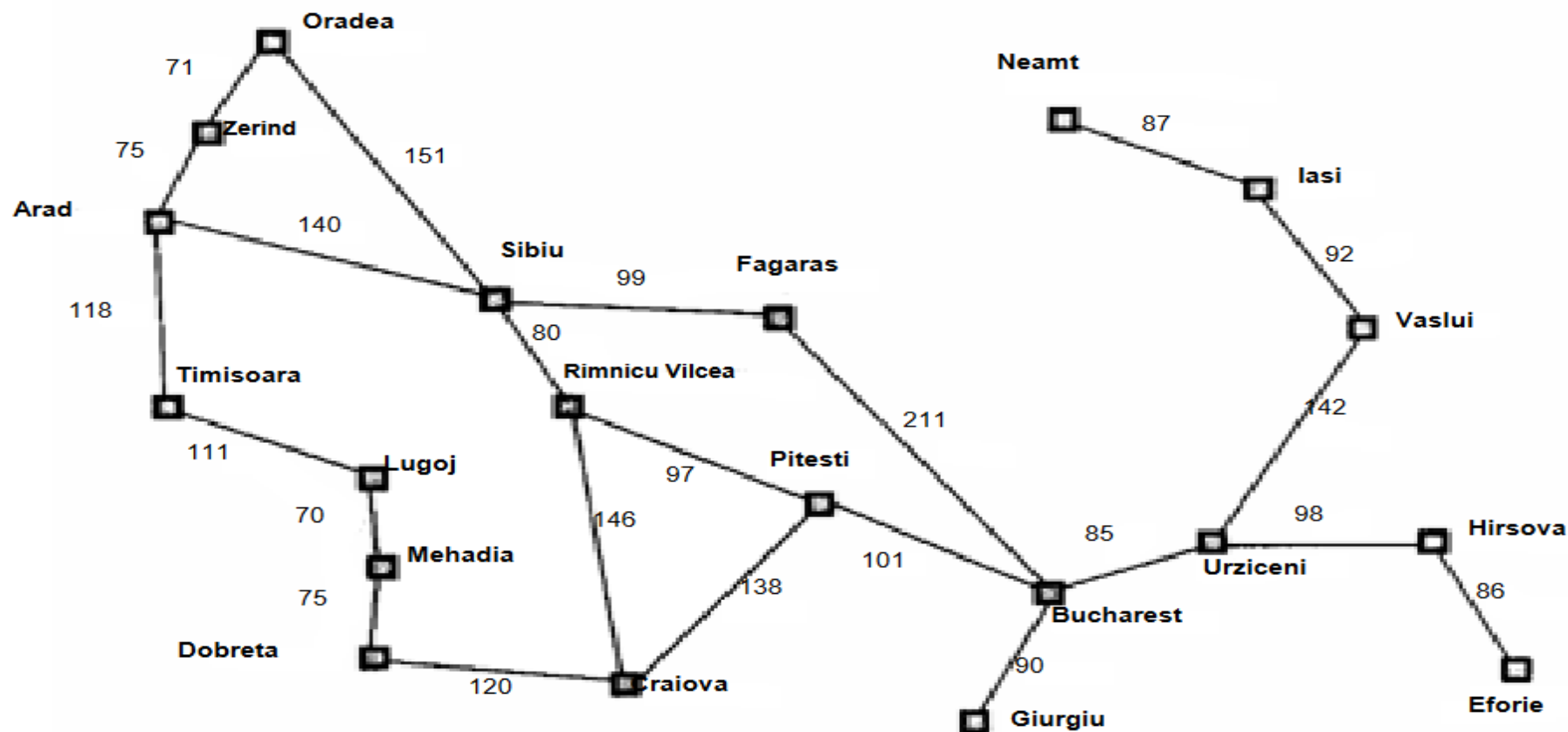
---

- ❑ El término Voraz (Greedy) ó Avaro es porque en cada paso trata de situarse tan cerca del objetivo como pueda, seleccionando el nodo con menor función de evaluación  $f(n)$
- ❑ No necesariamente brinda la solución óptima
- ❑ Al igual que los otros métodos estudiados es necesario verificar los “callejones sin salidas” (no expandir estados repetidos)

# Ejemplo

## Objetivo: Ir a Bucarest

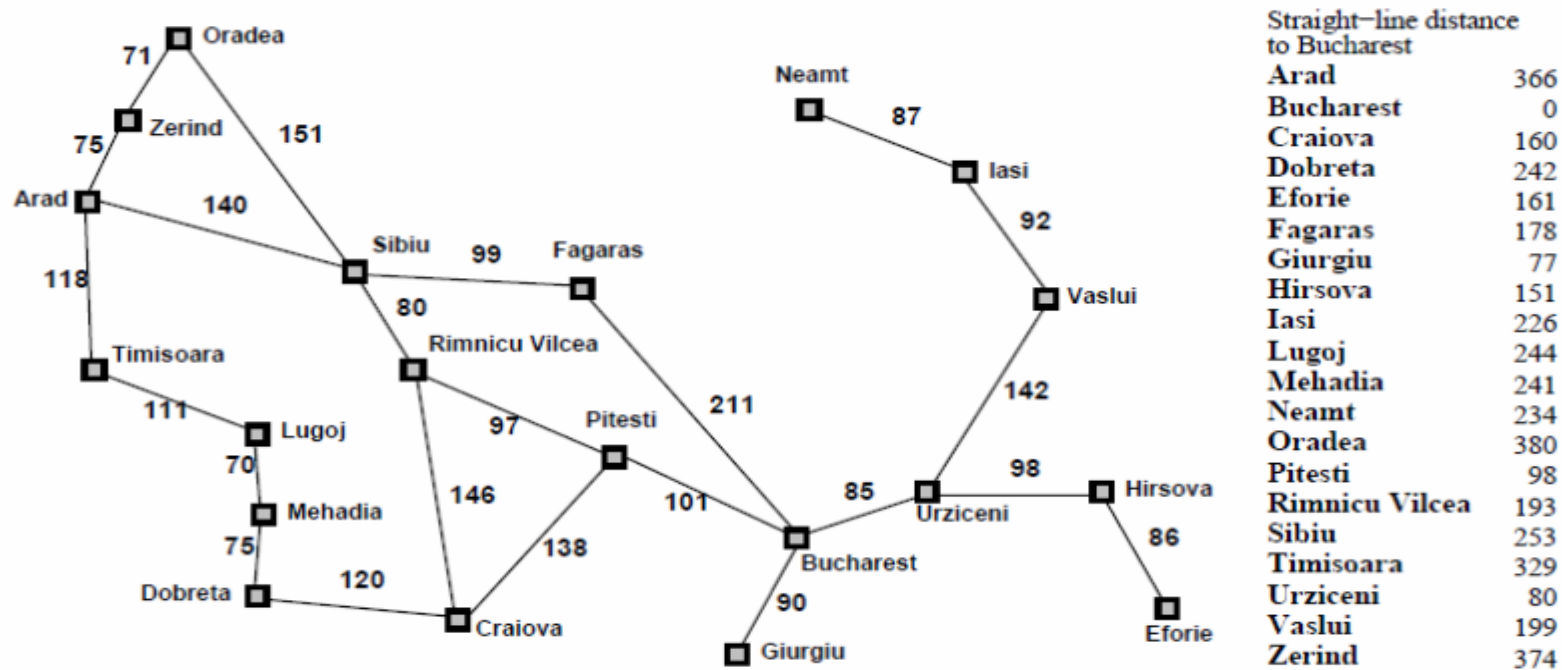
---



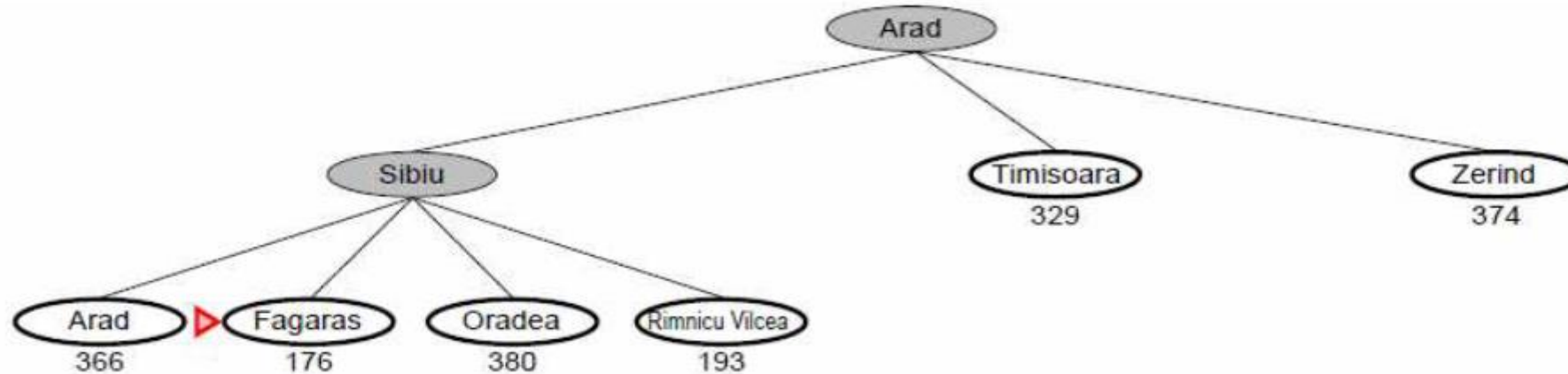
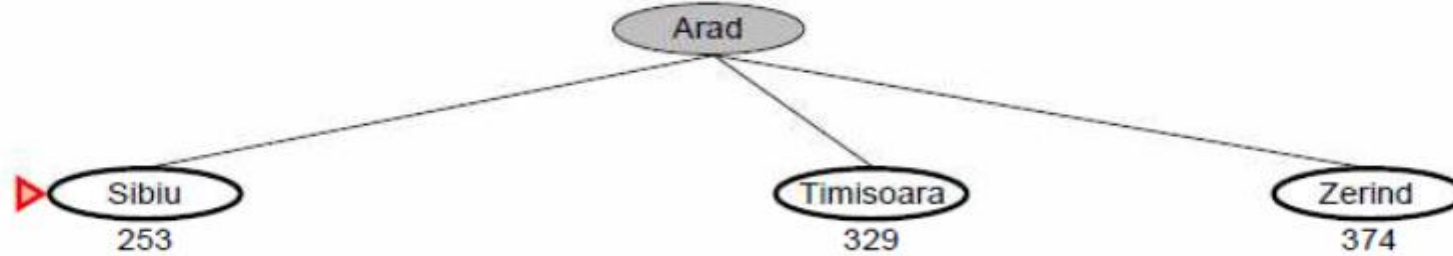


# Heurística

- ❑ Consideraremos como función de evaluación (función heurística) a  $hDLR(n)$  = Distancia en Línea Recta desde la ciudad n hasta Bucharest

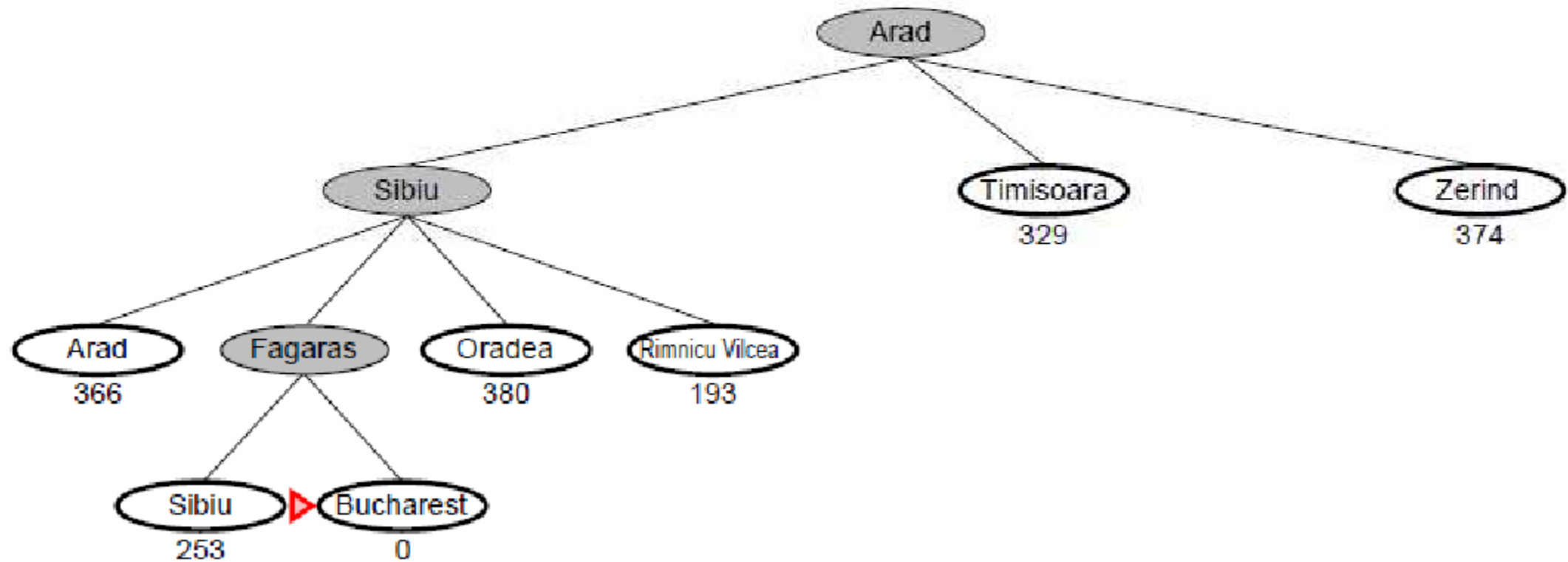


# Búsqueda voraz primero el mejor



# Búsqueda voraz primero el mejor

---



# Ejemplo (cont.)

---

- ❑ Solución de Búsqueda voraz primero el mejor:
  - Arad – Sibiu – Fagaras – Bucharest
  - Costo total:  $(140+99+211) = 450$
- ❑ Sin embargo:
  - Arad – Sibiu – Rimnicu – Pitesti – Bucharest
  - Costo total:  $(140+80+97+101) = 418$



# Búsqueda A\*

---

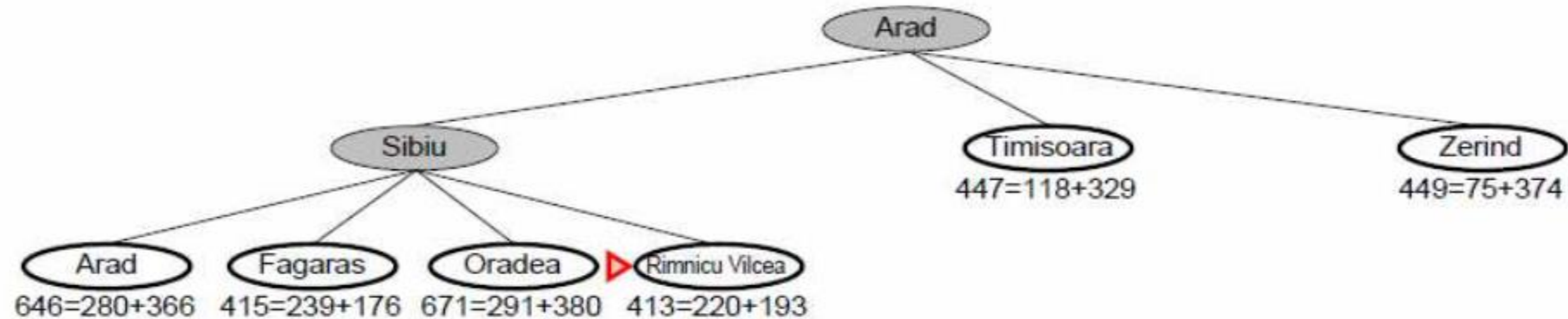
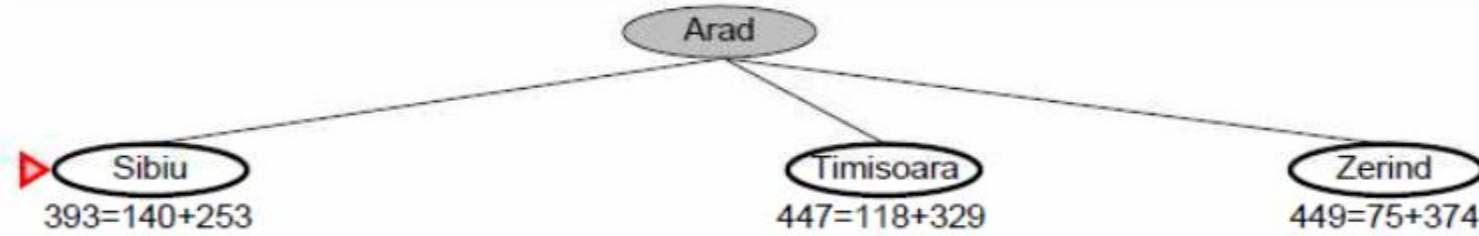
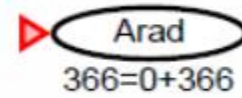
- ❑ Minimizar el costo estimado total de la solución
- ❑ Evalúa los nodos combinando  $g(n)$  y  $h(n)$ 
  - $g(n)$ : costo de haber alcanzado  $n$
  - $h(n)$ : costo para llegar desde  $n$  hasta el objetivo
  - $f(n) = g(n) + h(n)$  -> costo más barato estimado de la solución a través de  $n$

# Búsqueda $A^*$

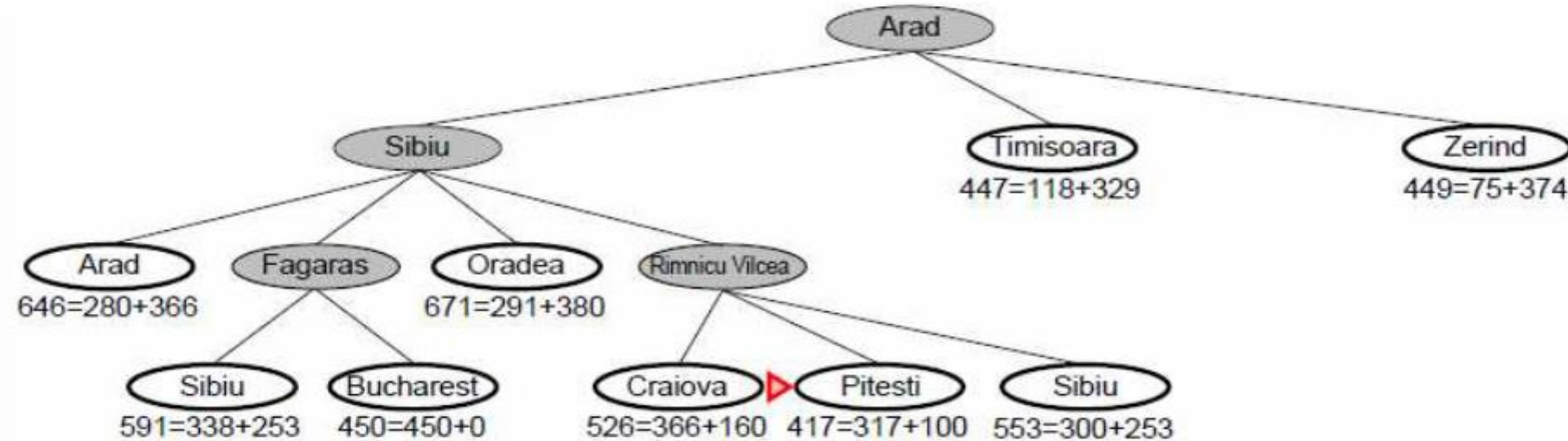
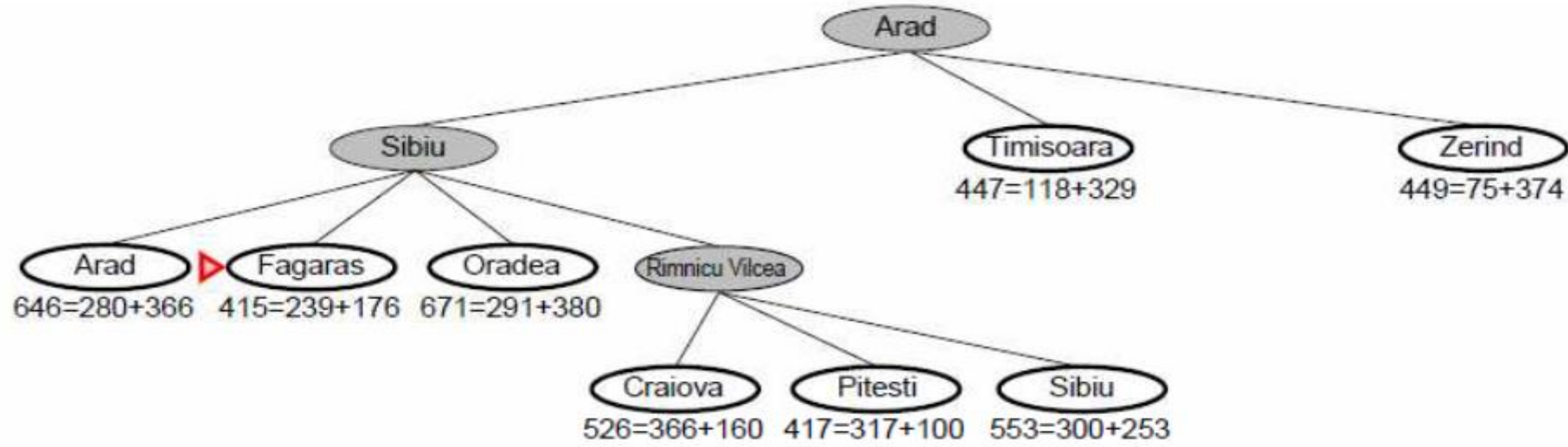
---

- ❑ En cada paso se expande el nodo con el valor más bajo de  $f(n)$ , ó sea, de  $g(n)+h(n)$
- ❑ La búsqueda  $A^*$  es óptima siempre y cuando la función heurística  $h(n)$  sea una heurística *admisible*, i.e. nunca sobreestime el costo de alcanzar el objetivo
- ❑ Son funciones optimistas
- ❑ En el ejemplo hDLR es admisible ya que la distancia en línea recta entre dos puntos es el camino más corto.

# Búsqueda A\*

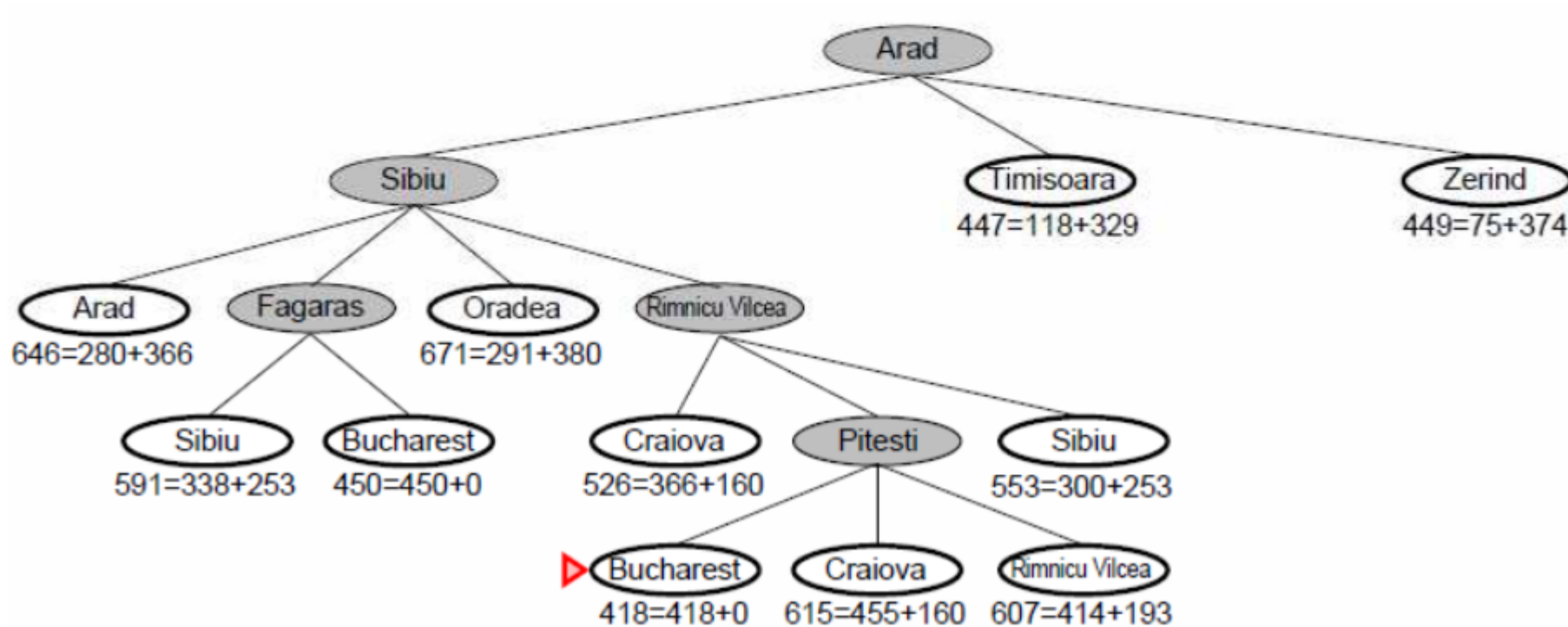


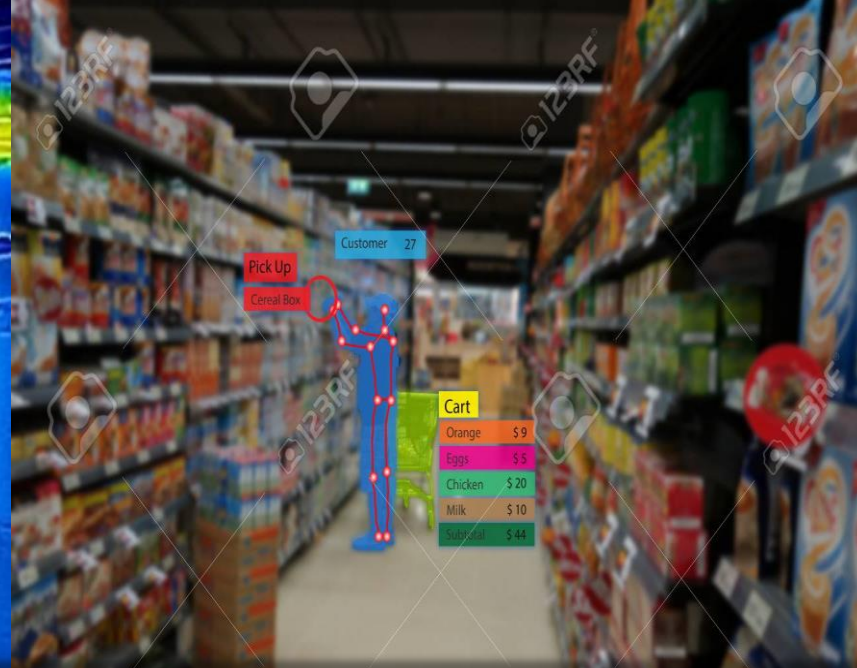
# Búsqueda A\*





# Búsqueda A\*





# Preguntas ¿? Inteligencia Artificial

PhD Carola Figueroa Flores  
Académica del DCCTI-UBB

Noviembre 2021