

Búsqueda Informada

Redes Neuronales (y algo más)

- ▶ ¿A qué llamamos inteligencia?
- ▶ ¿A qué llamamos inteligencia artificial?
- ▶ ¿Cómo podemos representar conocimiento?
- ▶ ¿Cómo resolvemos problemas basados en conocimiento?

State Explosion Problem

- ▶ ¿Cuántos estados es necesario representar?
- ▶ ¿Cuántos estados es necesario explorar?
- ▶ ¿Podemos codificar información que acelere la búsqueda?

State Explosion Problem

- ▶ ¿Cuántos estados es necesario representar?
Todos, salvo que distingamos un error catastrófico.
Idealmente queremos generarlos bajo demanda.
- ▶ ¿Cuántos estados es necesario explorar?
- ▶ ¿Podemos codificar información que acelere la búsqueda?

State Explosion Problem

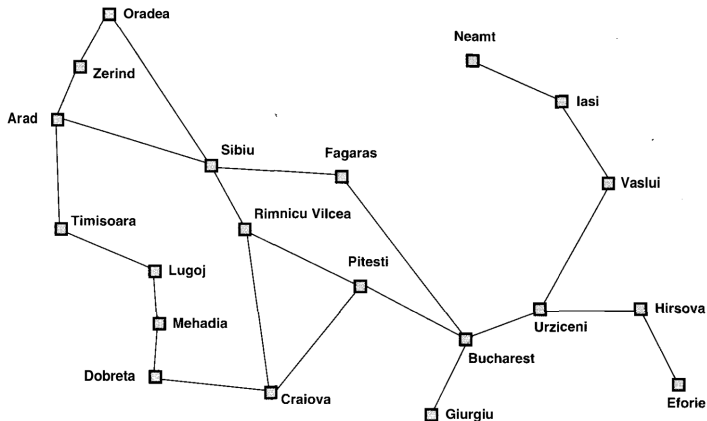
- ▶ ¿Cuántos estados es necesario representar?
*Todos, salvo que distingamos un error catastrófico.
Idealmente queremos generarlos bajo demanda.*
- ▶ ¿Cuántos estados es necesario explorar?
En mejor caso sólo es necesario explorar la rama entre el estado inicial y el objetivo.
- ▶ ¿Podemos codificar información que acelere la búsqueda?

State Explosion Problem

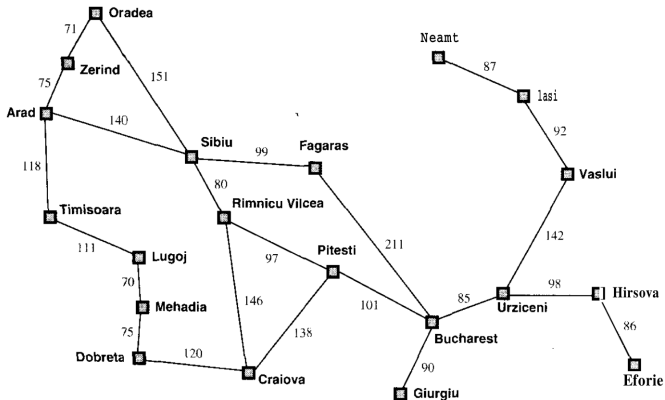
- ▶ ¿Cuántos estados es necesario representar?
*Todos, salvo que distingamos un error catastrófico.
Idealmente queremos generarlos bajo demanda.*
- ▶ ¿Cuántos estados es necesario explorar?
En mejor caso sólo es necesario explorar la rama entre el estado inicial y el objetivo.
- ▶ ¿Podemos codificar información que acelere la búsqueda?
Sí, en particular priorizando la selección del siguiente estado a expandir.

From Arad to Bucharest

From Arad to Bucharest



From Arad to Bucharest (considering distances)



Straight-line distance
to 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

¿Cómo guiamos la búsqueda?

Heurística

Función que “estima” la distancia al objetivo.

- ▶ Puede mejorar la performance en caso promedio.
- ▶ No necesariamente mejora la performance en peor caso.

¿Cómo guiamos la búsqueda?

Heurística

Función que “estima” la distancia al objetivo.

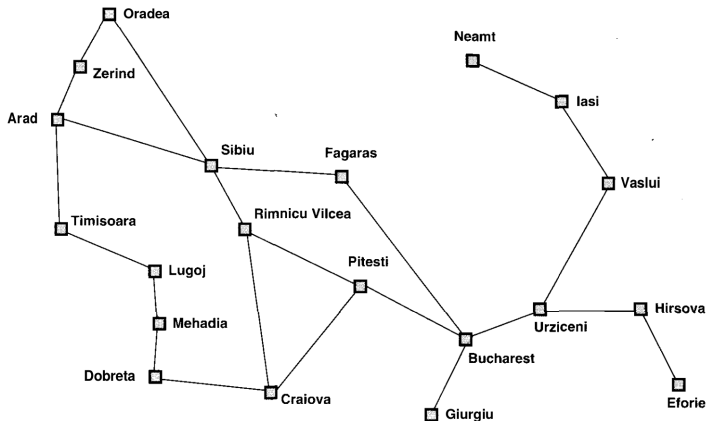
- ▶ Puede mejorar la performance en caso promedio.
- ▶ No necesariamente mejora la performance en peor caso.

Pedimos que $h(n) = 0$, si n es un estado objetivo.

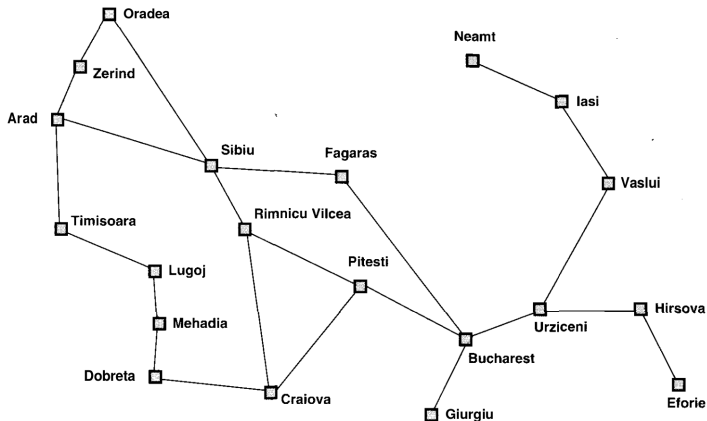
Best First Search or Greedy Search

```
def gs(graph, initial, h):
    pqueue = PriorityQueue()
    visited = set()
    pqueue.put((h(initial), initial, []))
    visited.add(initial)
    while not pqueue.empty():
        priority, node, path = pqueue.get_nowait()
        if priority == 0:
            return node, path
        for edge_label, target_node in graph[node].iteritems():
            if target_node not in visited:
                visited.add(target_node)
                pqueue.put((
                    h(target_node),
                    target_node,
                    path + [edge_label]
                ))
    return None
```

From Iasi to Fagaras



From Iasi to Fagaras



La heurística puede guiarnos mal, la búsqueda debe ser robusta.

¿Se puede buscar el camino mas corto?

¿Se puede buscar el camino mas corto?

Algoritmo: A^*

Considera el costo acumulado hasta el nodo n desde el estado inicial ($g(n)$) más lo que estima la heurística ($h(n)$).

$$f(n) = g(n) + h(n)$$

¿Se puede buscar el camino mas corto?

Algoritmo: A^*

Considera el costo acumulado hasta el nodo n desde el estado inicial ($g(n)$) más lo que estima la heurística ($h(n)$).

$$f(n) = g(n) + h(n)$$

Si h es admisible entonces el resultado es el de “costo” mínimo.

Admisibilidad

Una función heurística es admisible cuando nunca sobre-estima la distancia al objetivo.

Corolario: Si $h(n) = \infty$, entonces el objetivo es inalcanzable.

- ¿Suelen ser eficientes GS y A^* ?

	Exploración	Solución
GS		
A^*		

- ¿Suelen ser eficientes GS y A^* ?

	Exploración	Solución
GS	Sí	No
A^*	No	Sí

¿Cómo medimos la calidad de una heurística?

¿Cómo medimos la calidad de una heurística?

Effective Branching Factor

El *EBF* mide la diferencia entre los nodos considerados por A^* en comparación con GS.

¿Cómo medimos la calidad de una heurística?

Effective Branching Factor

El *EBF* mide la diferencia entre los nodos considerados por A^* en comparación con GS.

Decimos que h_2 domina a h_1 si $h_2(n) \geq h_1(n)$ para todo n .

Lo ideal es usar la función que retorne los valores más altos en la medida que no sobre-estime la distancia al objetivo.

¿Cómo inventar heurísticas?

Acercamientos clásicos:

- ▶ Aprendizaje (sistemas expertos)
- ▶ Problema relajado
- ▶ Most constrained variable

- ▶ Búsqueda informada mediante GS y A^* .
- ▶ Representación de “intuición” mediante heurísticas.
- ▶ Heurísticas admisibles y dominantes.

- ▶ Búsqueda informada mediante GS y A^* .
- ▶ Representación de “intuición” mediante heurísticas.
- ▶ Heurísticas admisibles y dominantes.

Ejercicio:

- ▶ 8-Square Problem (on-the-fly)
- ▶ Implemente dos funciones heurísticas:
 - ▶ h_1 : NumberOfWrongNumbers
 - ▶ h_2 : SumOfManhattanDistances
- ▶ Indique si las heurísticas son admisibles y si una domina a la otra.
- ▶ Compare el proceso de exploración y la calidad del resultado para GS y A^* usando ambas heurísticas.