



Búsqueda Informada (Heurística)

Sistemas Inteligentes

Dr. Víctor de la Cueva

vcueva@itesm.mx

Búsqueda Heurística

- Utiliza el conocimiento específico del problema más allá de la definición del mismo.
- Puede encontrar soluciones de una forma más eficiente que las estrategias no informadas.
- Su forma general es el algoritmo llamado Búsqueda Primero el Mejor (BFS).

Búsqueda Primero el Mejor

- La selección del nodo para expansión se hace de acuerdo a una función de evaluación $f(n)$.
- Tradicionalmente, $f(n)$ mide la distancia al objetivo por lo que el mejor es el del valor más bajo.
- Se implementa fácilmente con una cola con prioridades, estructura que mantiene la frontera en orden ascendente de valores f .
- Nombre venerable pero inexacto.
- Hay una familia entera de algoritmos BFS con evaluaciones diferentes.
- El componente clave de estos algoritmos es una función heurística, denotada por $h(n)$.

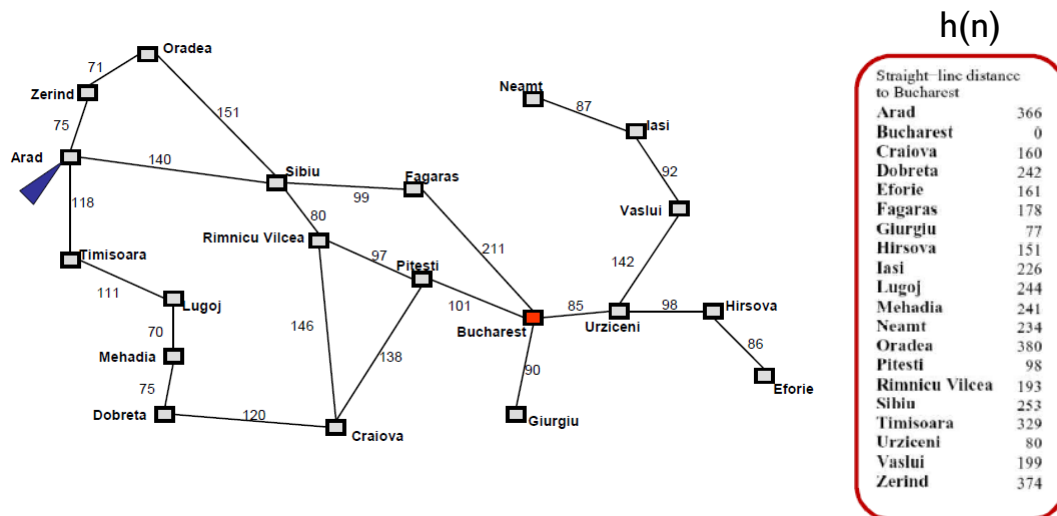
Heurística

- $h(n)$ = costo estimado del camino más barato desde el nodo n hasta un nodo objetivo.
- Las funciones heurísticas son la forma más común de transmitir el conocimiento adicional del problema al algoritmo de búsqueda.
- Por ahora se considerarán funciones arbitrarias, no negativas, con una restricción: si n es un estado objetivo, $h(n) = 0$.
- Hablando en términos más coloquiales, una heurística es una regla de dedo que ha demostrado ser efectiva pero que no se puede demostrar que lo es.

Búsqueda (Voraz) Primero el Mejor

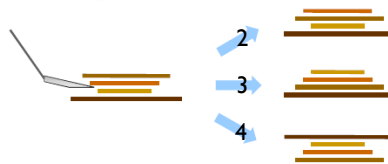
- Trata de expandir el nodo más cercano al objetivo, con la idea de que probablemente conduzca rápidamente a una solución.
- Evalúa los nodos usando solamente la función heurística, es decir, $f(n) = h(n)$.
- Se parece a DFS y tiene los mismos problemas pero se pueden minimizar con una buena heurística.

Un tour por Rumania con heurística

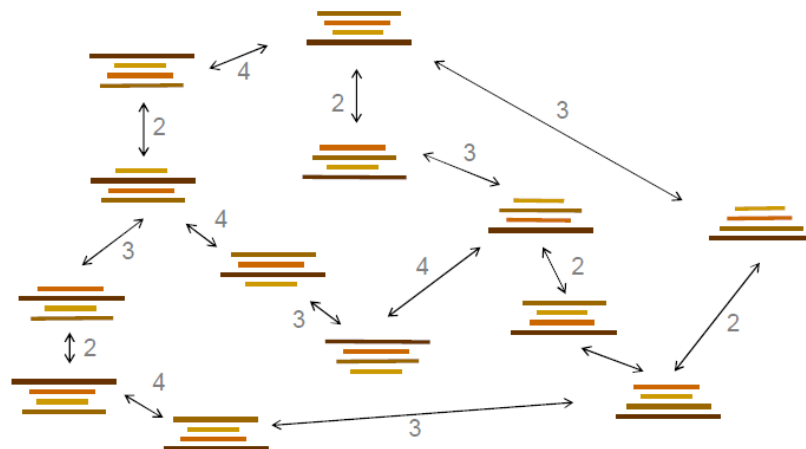


Problema de los hot cakes

- Se tienen 4 hot-cakes, uno más grande que el siguiente. ¿Cuál es el menor número de hot-cakes que se deben voltear para que queden ordenados de mayor (abajo) a menos (arriba)?
- En cada vuelta se pueden tomar desde 2 hasta 4 hot-cakes (1 no tiene sentido).
- Dicho número se puede tomar como el costo.



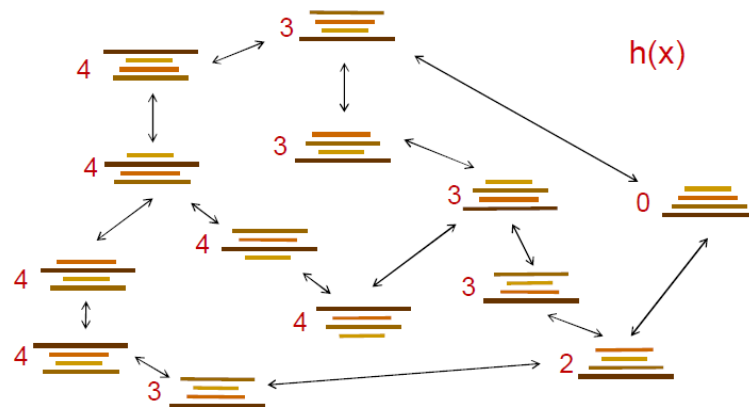
Grafo de estados con costo como peso



Fuente: MOOC AI, EdX, Dan Klein. UC Berkeley (2013)

¿Heurística?

- El número del hot-cake más grande que está fuera de su lugar.



Fuente: MOOC AI, EdX, Dan Klein. UC Berkeley (2013)

Ejemplo de Heurísticas para 8-puzzle

¿Cuál sería una buena heurística para el problema del 8-puzzle? Es decir, un número que me indique si me falta mucho o poco para llegar a la meta.

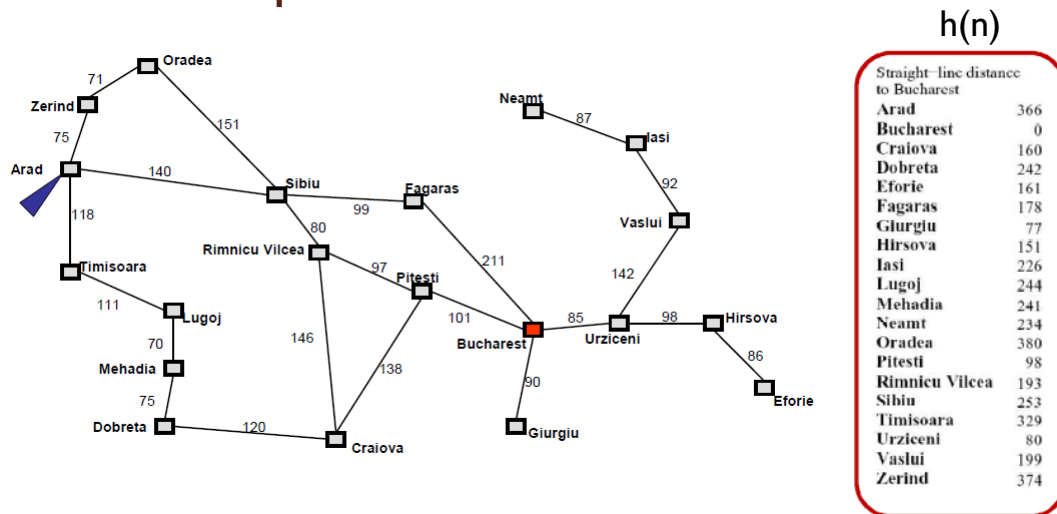
H1: Número de cuadritos fuera de su lugar.

H2: Suma de distancias Manhattan de los números a su lugar correcto.

A*

- Es el algoritmo BFS más conocido.
- Propuesto por Peter Hart, [Nils Nilsson](#) and Bertram Raphael del *Stanford Research Institute*, en 1968
- Evalúa los nodos con una función $f(n) = g(n) + h(n)$, donde $g(n)$ es el costo de alcanzar el nodo n desde la raíz y $h(n)$ es el costo estimado para ir del nodo n al nodo meta (heurística).
- Es decir, $f(n)$ estima el costo de la mejor solución de la raíz a la meta.
- En otras palabras, está combinando UCS con BFS.
- Un buen algoritmo en pseudocódigo lo pueden encontrar en Wikipedia: https://en.wikipedia.org/wiki/A*_search_algorithm

Un tour por Rumania con A*



Proyecto 2

Buena heurística

- Para que A* funcione correctamente, la clave es tener una buena heurística.
- Las características de una buena heurística son:
 - Admisible: Son heurísticas optimistas que piensan que el costo de resolver un problema es MENOR que el que realmente es (e.g. distancia recta en Tour por Rumania).
 - Consistente (o monotónica): si para cada nodo n y para cada sucesor n' , el costo estimado para alcanzar la meta desde n es menor que el costo del paso para ir de n a n' , más el costo estimado para ir de n' a la meta ($h(n) < c(n,a,n') + h(n')$). Sólo se requiere para búsqueda en grafos.

Optimalidad de A*

- A* en su versión de TREE-SEARCH, si la heurística es admisible.
- A* en su versión de GRAPH-SEARCH, si la heurística es admisible y consistente.
- La característica más importante de las dos es la admisibilidad.
- Desde luego que entre mejor sea la heurística, es decir, entre mejor estime el valor real, mejor va a ser la búsqueda, es decir, se entrará el camino más corto.

demo

¿Cómo crear heurísticas admisibles?

- Es complicado formar una heurística admisible para los problemas, sin embargo, existe un método que garantiza una heurística admisible y se trata de usar un problema relajado.
- Un problema relajado es una versión del problema original que tienen menos restricciones.
- El costo de la solución óptima de un problema relajado es una heurística admisible para el problema original.

8-puzzle

- Problema Original. Un cuadrado A se puede mover a un cuadrado B si:
 1. A es horizontal o verticalmente adyacente a B.
 2. B es el cuadro blanco.
- Ese problema tiene dos restricciones por lo que se pueden construir 3 problemas relajados:
 - Quitando las dos (Problema relajado 1)
 - Un cuadrado A se puede mover a un cuadrado B.
 - Quitando la restricción 2 (Problema relajado 2)
 - Un cuadrado A se puede mover a un cuadrado B si A es horizontal o verticalmente adyacente a B.
 - Quitando la restricción 1 (Problema relajado 3)
 - Un cuadrado A se puede mover a un cuadrado B si B es el blanco.

Referencia

- S. Russel and P. Norvig. Inteligencia Artificial un enfoque moderno. 2ª edición, Pearson, España (2004).