

Best-First Search y A*

Jorge Baier

Departamento de Ciencia de la Computación
Pontificia Universidad Católica de Chile

Santiago, Chile

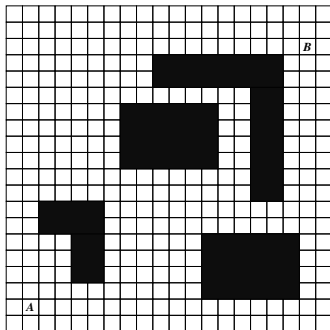


- Comprender el algoritmo de búsqueda Best-First Search
- Conocer el concepto de Función Heurística
- Introducir el Algoritmo A*



Busqueda Informada

¿Qué podemos hacer para mejorar la búsqueda en estos casos?



Problema: ir de *A* a *B*

1	2	3
	5	4
6	7	8



1	2	3
8		4
7	6	5



Búsqueda el Mejor Primero (*Best-First Search*)

El algoritmo *el mejor primero*, intuitivamente:

- Mantiene una lista de *Open* y *Closed*.
- Funciona como DFS, pero:
- Los nodos en *Open* tienen asociados una calidad.
- Siempre extrae de *Open* el nodo de mejor calidad.
- Un estado sucesor es descartado si está en *Closed* con mejor o igual calidad.



Función Heurística

- En búsqueda informada, usamos una función de estimación del costo de un nodo del árbol de búsqueda a una solución. La denotamos como

$$h(n)$$

- En el problema de navegación, si

$$\Delta x = |x_{obj} - x|, \quad \Delta y = |y_{obj} - y|,$$

donde (x, y) es la posición actual y (x_{obj}, y_{obj}) es el objetivo. La siguiente es una posible heurística (también llamada *distancia octile*)

$$h(x, y) = |\Delta x - \Delta y| + \sqrt{2} \min\{\Delta x, \Delta y\}$$



Incorporando el Costo

- Como vimos en el ejemplo, usar sólo h conduce a soluciones no óptimas.
- Es posible encontrar soluciones óptimas al incorporar el *costo* incurrido hasta llegar a un nodo n .
- Denotamos este costo como $g(n)$.
- Luego, podemos ordenar la frontera de búsqueda por la siguiente función:

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



Algoritmo A*

Input: Un problema de búsqueda (S, A, s_0, G)

Output: Un nodo objetivo

- 1 **for each** $s \in S$ **do** $g(s) \leftarrow \infty$
- 2 $Open \leftarrow \{s_0\}$
- 3 $g(s_0) \leftarrow 0$; $f(s_0) \leftarrow h(s_0)$
- 4 **while** $Open \neq \emptyset$
- 5 Extrae un u desde $Open$ con menor valor- f
- 6 **if** u es objetivo **return** u
- 7 **for each** $v \in Succ(u)$ **do**
- 8 Insertar v

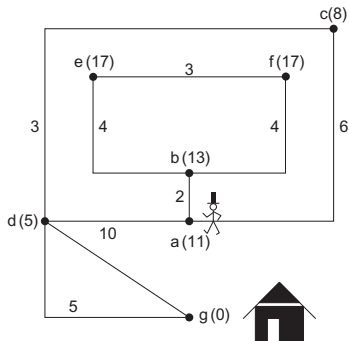


Insertar v en *Open*

- 1 $cost_v = g(u) + c(u, v)$ // el costo de llegar a v por u
- 2 **if** $cost_v \geq g(v)$ **return** // seguimos solo si $cost_v < g(v)$
- 3 $parent(v) \leftarrow u$
- 4 $g(v) \leftarrow cost_v$
- 5 $f(v) \leftarrow g(v) + h(v)$
- 6 **if** $v \in Open$ **then** Reordenar *Open* // depende de la impl.
- 7 **else** Insertar v en *Open*



Un ejemplo



Entre paréntesis, $h(n)$.



- Comprender el algoritmo de búsqueda Best-First Search
- Conocer el concepto de Función Heurística
- Introducir el Algoritmo A*

