A*

Jorge Baier

Departamento de Ciencia de la Computación Pontificia Universidad Católica de Chile Santiago, Chile



Objetivos

- Comprender el algoritmo A*
- Entender la optimalidad de A*



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 Principal

Algoritmo A*

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

Output: Un nodo objetivo

- **1** for each $s \in \mathcal{S}$ do $g(s) \leftarrow \infty$
- 2 $Open \leftarrow \{s_0\}$
- $g(s_0) \leftarrow 0; \ f(s_0) \leftarrow h(s_0)$
- **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



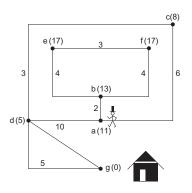
El Procedimiento *Insertar*

Insertar v en Open

- **2** if $cost_v \ge g(v)$ return // seguimos solo si $cost_v < g(v)$
- \square parent $(v) \leftarrow u$
- **4** $g(v) \leftarrow cost_v$
- $f(v) \leftarrow g(v) + h(v)$
- **6** if $v \in Open$ then Reordenar Open // depende de la impl.
- **7 else** Insertar *v* en *Open*



Un ejemplo

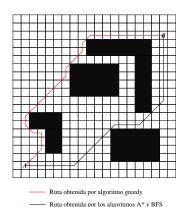


Entre paréntesis, h(n).



A* y Greedy

- Si usamos f(n) = h(n) en A*, entonces el algoritmo resultante es *greedy best-first search* (ambicioso).
- Los algoritmos ambiciosos encuentran soluciones más rápidamente, sacrificando la calidad de la solución.





Optimalidad de A*

Partiremos con algunas definiciones

Definición

Para un estado s, denotamos por $h^*(s)$ al costo de un camino óptimo desde s a un estado objetivo.

Definición (Admisibilidad)

Una función heurística h se dice admisible, si para todo s:

$$h(s) \leq h^*(s)$$

Teorema (Optimalidad de A*)

Si h es admisible, entonces A^* , usado con h, encuentra una solución óptima si esta existe.

Demostración: Pizarra.

Objetivos

- Comprender el algoritmo A*
- Entender la optimalidad de A*

