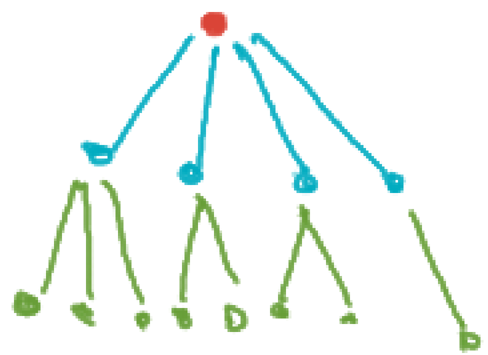
Camino más corto

- Dijkstra (BFS)
- Bellman-Ford
- Floyd-Warshall
- A*

Árbol de Expansión mínima:

- Prim-Jarník (BFS)
- Kruskal

BFS: (por ejemplo, Dijkstra)



$h(s_0) = 6$

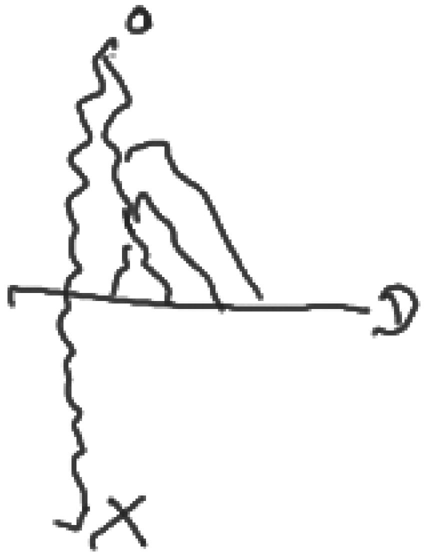
¿Cómo evaluar el desempeño de un alg. de búsqueda?

- <u>Completitud</u>: ¿El algoritmo garantiza encontrar una solución <u>óptima</u>, cuando hay o reporta falla cuando <u>no</u> <u>hay solución</u>?

- <u>Costo optimal</u>: ¿El algoritmo produce una solución de costo mínimo?

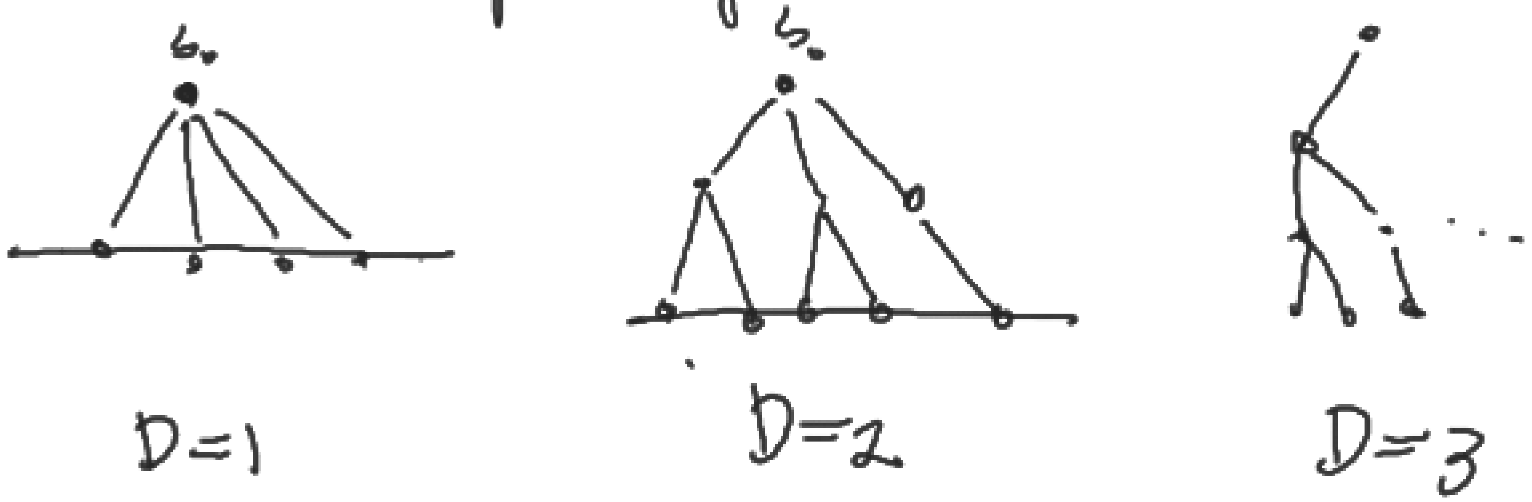
  Soluciones sub-optimales = "cerca" de ser las sol. de costo mínimo.

- <u>Complejidad</u> <u>Temporal</u>: $O(f(n))$
  Tiempo de ejecución



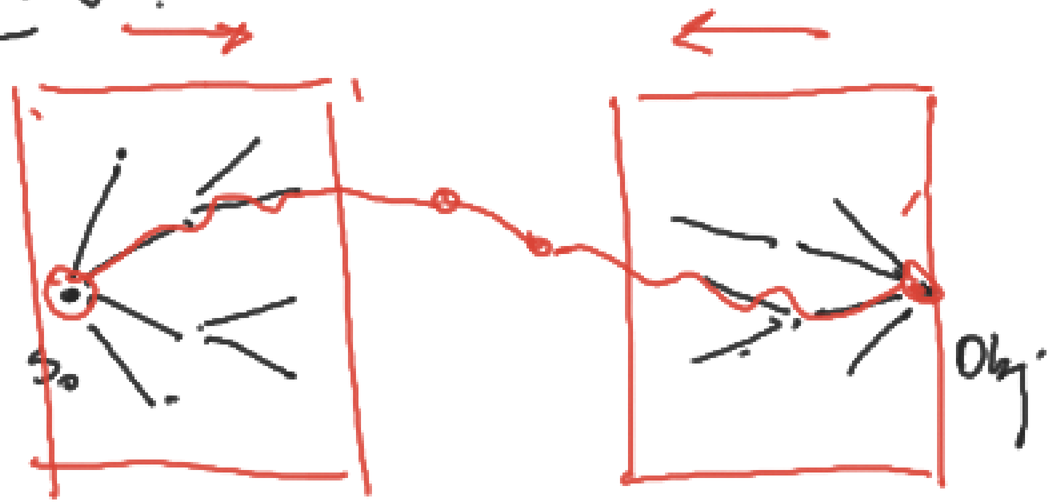
- <u>Complejidad</u> <u>Espacial</u>: Tamaño de <u>memoria</u>

Iterative Deepening

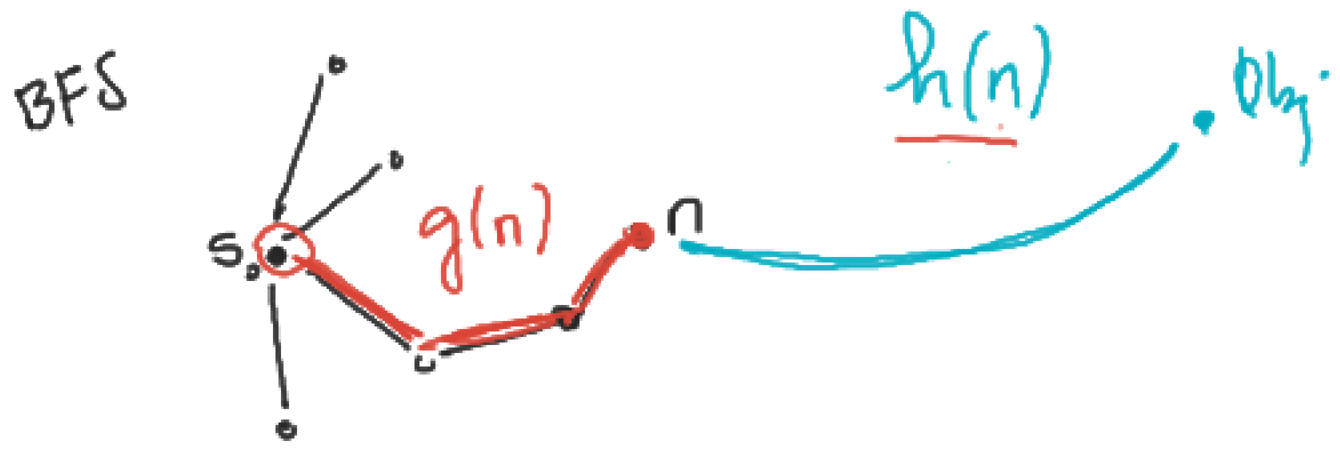


~~BFS~~

Bidireccional:



Heurísticas:

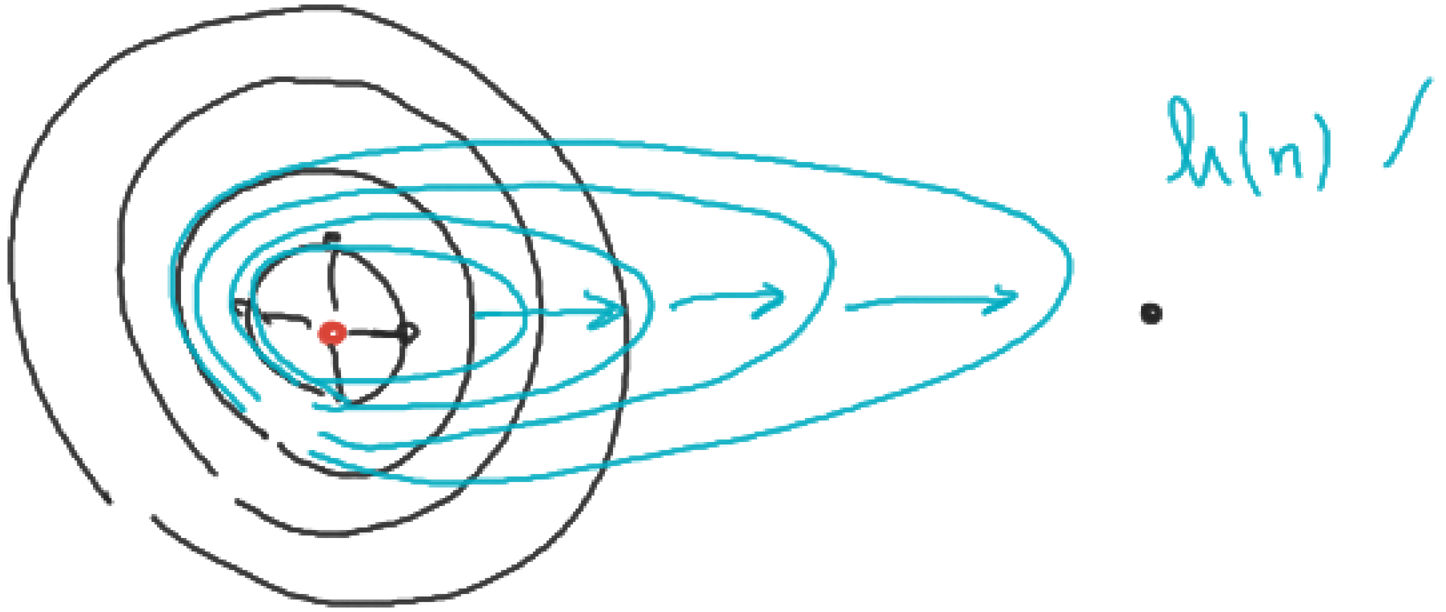


$$f(n) = \underbrace{\text{costo total del camino desde } s_0 \text{ hasta } n}_{g(n)}$$

Problema: $\boxed{\min_{r} f(r)}$ r es camino de s_0 a Obj.

Heurística $h(n)$ = estima el costo para llegar desde el nodo actual n hasta el objetivo.

$$h: V \longrightarrow \mathbb{R}$$



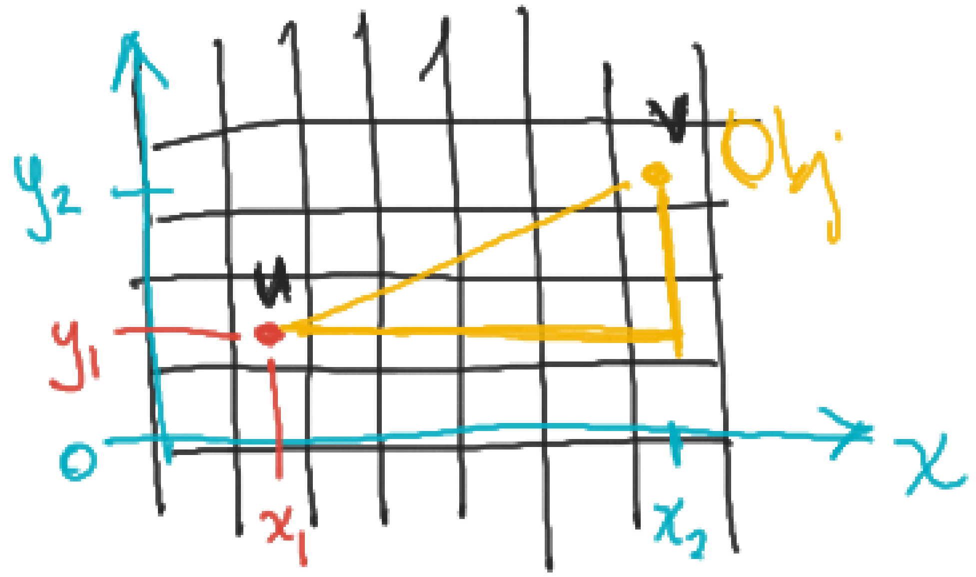
Al introducir una heurística, modificamos la función objetivo.

informado.

$$f(n) = \underbrace{g(n)}_{\text{costo de } s_0 \text{ a } n} + \underbrace{h(n)}_{\text{costo estimado de } n \text{ a obj}}$$

$$\min_{r} f(r) = \text{costo estimado de } s_0 \text{ a obj. (pasando por } n\text{).}$$

Ejemplo:



- Distancia Euclideana

$$h(u) = \|u - v\|_2 = \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$$

- Distancia Manhattan $(\|\cdot\|_1)$

$$h(u) = \|u-v\|_1 = |x_2-x_1| + |y_2-y_1|$$

- Distancia de Chebyshev $(\|\cdot\|_\infty)$

$$h(u) = \|u-v\|_\infty = \text{máx}\{|x_1-x_2|, |y_1-y_2|\}$$

- Distancia Octile

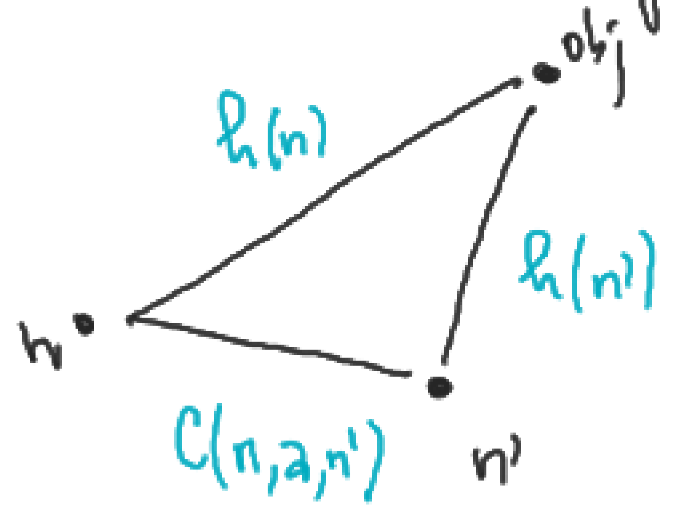
$$1 \cdot \text{máx}\{dx, dy\} + (\sqrt{2}-1)\,\text{mín}\{dx, dy\}$$

Usualmente requerimos que una heurística sea:

- <u>admisible</u>: h nunca sobre-estima el costo ($h(n) \leq$ costo real de n a obj).

- <u>consistente</u>: si para cada nodo n, y cada nodo sucesor n' de n generado por la acción a,

$$h(n) \leq C(n, a, n') + h(n').$$



<u>Prop</u>: h admisible $\Rightarrow$ A^* es costo optimal

h consistente $\Rightarrow$ la primera vez que alcanzamos un nodo <u>n</u> n está en un trayecto óptimo.

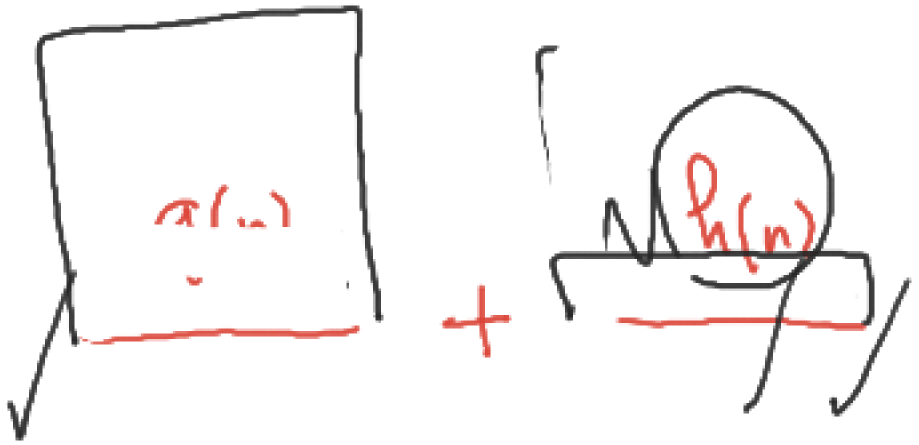
Algoritmo A^*: (A star, Ástar, A estrella)

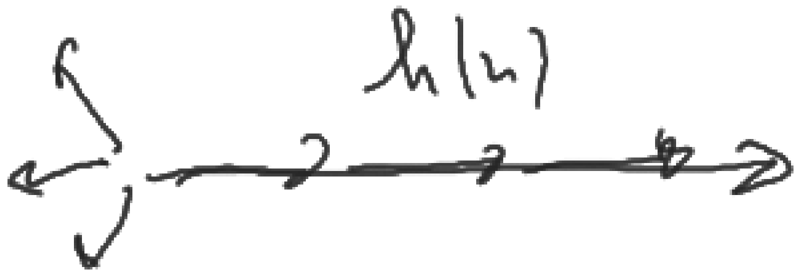
BFS con la función objetivo $f(n) = \underline{g(n) + h(n)}$

$w_1 g(n) + w_2 h(n)$

Casos:

	weighted A^*:	$f(n) = g(n) + W h(n)$	$W > 0$, $W \neq 1$
	A^*:	$\boxed{= g(n) + h(n)}$	$\underline{W = 1}$
$W \in \mathbb{R}$	Dijkstra:	$= g(n)$	$W = 0$
	Greedy:	$= h(n)$	"$W = \infty$"





$\underline{W = 0.1}$


s_0


s_0
$W > 0$
(g)
no es
opt en distancia
gth