

# Heurísticas

Jorge Baier

Departamento de Ciencia de la Computación  
Pontificia Universidad Católica de Chile

Santiago, Chile



- Comprender los conceptos de heurísticas consistentes y admisibles
- Introducir el algoritmo IDA\*



## Definición (Heurísticas Consistentes)

Una heurística se dice *consistente* ssi

- $h(s) = 0$ , para todo  $s \in G$ .
- $h(s) \leq c(s, s') + h(s')$ , para todo vecino  $s'$  de  $s$ .

## Teorema

Si  $h$  es consistente, entonces  $h$  es admisible.

## Teorema

Cuando  $A^*$  es usado con una heurística admisible, cuando  $A^*$  expande un nodo  $v$ ,  $g(v)$  contiene el costo del camino óptimo desde  $s_0$  a  $v$ .

El anterior teorema tiene un potencial impacto en la forma de implementar  $A^*$ .



## Teorema

Si  $h_1$  y  $h_2$  son consistentes y  $h_1 \geq h_2$ , entonces  $A^*$ , usado con  $h_2$ , expande todos los nodos que  $A^*$  expande cuando es usado con  $h_1$ .

Como conclusión tenemos que  $h_1$  es “mejor” que  $h_2$  en la práctica.



# Encontrando Heurísticas Admisibles

- Una estrategia simple: *relajar* el problema.
- La heurística es el costo de resolver el problema relajado.
- Ejemplo:

2		3
1	8	6
7	5	4

Estado Inicial

1	2	3
8		4
7	6	5

Objetivo

- Los operadores respetan las siguientes restricciones:
  - 1 Un azulejo sólo se puede mover a un cuadrado vecino.
  - 2 Un azulejo sólo se puede mover a un cuadrado desocupado.



# Heurísticas en Nuestro Ejemplo

- Si relajamos ambas restricciones:

$h_1$  = “número de azulejos en la posición incorrecta”

- Si relajamos la restricción 2:

$h_2$  = “suma de la distancia *manhattan* de cada azulejo”

*¿cuál es mejor?*



# Heurísticas en Nuestro Ejemplo

- Si relajamos ambas restricciones:

$h_1$  = “número de azulejos en la posición incorrecta”

- Si relajamos la restricción 2:

$h_2$  = “suma de la distancia *manhattan* de cada azulejo”

*¿cuál es mejor?*

	Search Cost		
$d$	IDS	$A^*(h_1)$	$A^*(h_2)$
2	10	6	6
4	112	13	12
6	680	20	18
8	6384	39	25
10	47127	93	39
12	364404	227	73
14	3473941	539	113
16	—	1301	211
18	—	3056	363
20	—	7276	676
22	—	18094	1219
24	—	39135	1641



# Sacrificando Optimalidad Gradualmente

- $A^*$  con pesos (*weighted  $A^*$* ) es una buena opción cuando se está dispuesto a **sacrificar** optimalidad para obtener un mejor rendimiento.
- Consiste en usar  $A^*$  con la siguiente función de evaluación

$$f(n) = g(n) + w \cdot h(n),$$

con  $w \geq 1$ .

## Teorema

Si  $h$  es admisible, *weighted  $A^*$*  encuentra una solución cuyo costo es a lo más  $w$  veces el óptimo.

En la práctica encuentra soluciones mejores.



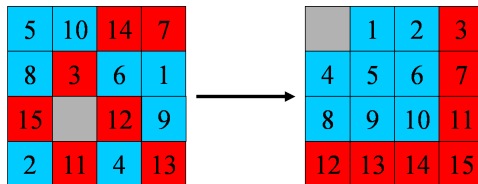


# Iterative Deepening A\* – IDA\*

- Algoritmo similar a A\* pero mucho más eficiente en memoria
- Realiza una serie de búsquedas usando DFS.
- Se poda una rama cuando se excede un límite (threshold) de costo.
- El threshold inicial es el valor- $h$  del nodo raíz.



- Técnica para computar heurísticas admisibles a problemas difíciles.
- Consiste en construir una *abstracción* del problema de búsqueda .
- Se pre-computan soluciones óptimas para las abstracciones.



- En tiempo de búsqueda se usa el costo de esta solución como heurística.

