UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Búsqueda informada [1]

Alfons Juan
Albert Sanchis
Jorge Civera

DSIC
Departamento de Sistemas Informáticos y Computación

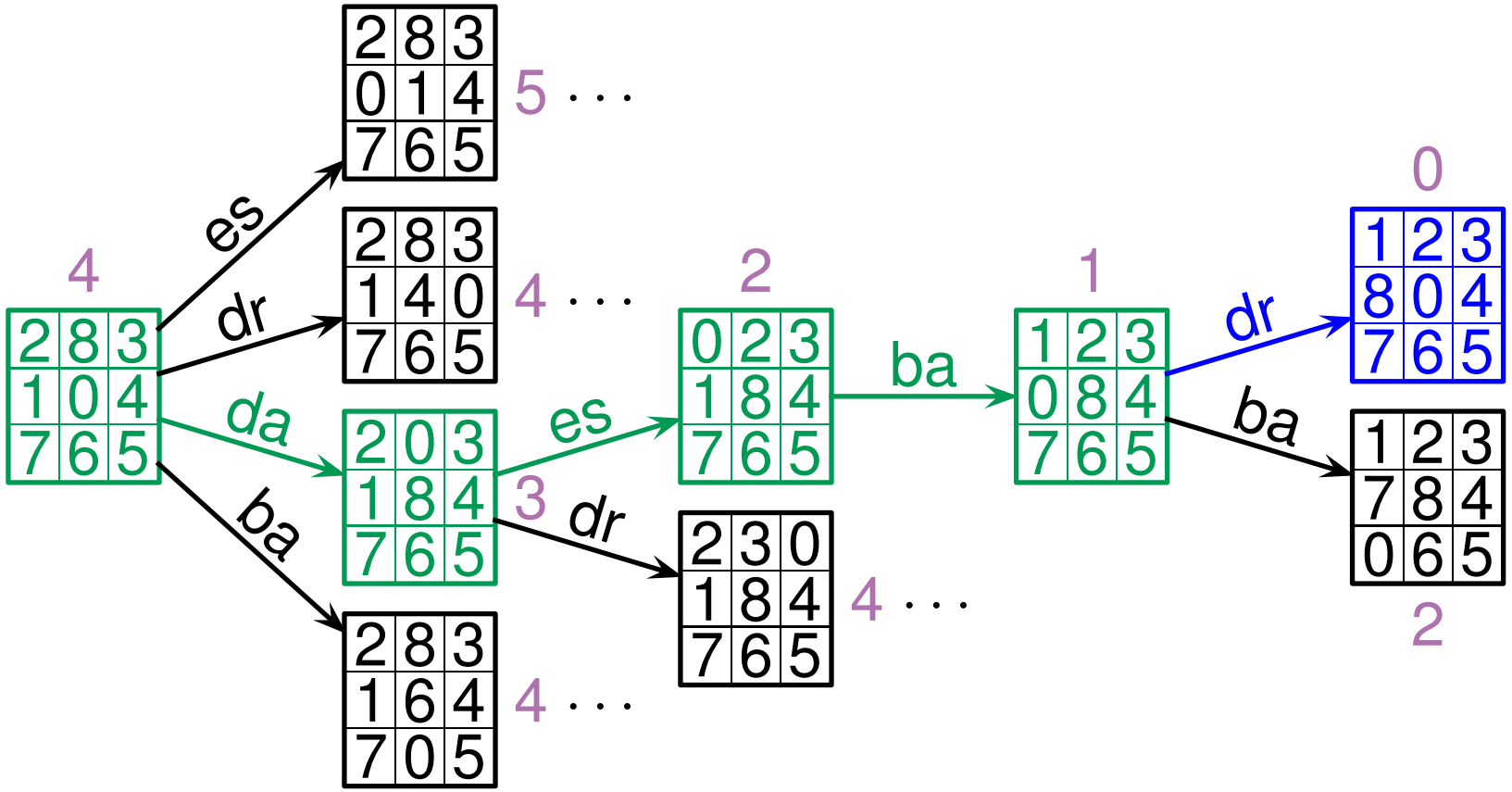
[1] Para una correcta visualización, se requiere Acrobat Reader v. 7.0 o superior

Índice

1. Introducción

Dado un problema de búsqueda representado con un grafo de estados G, una ***heurística*** es cualquier función h que estima, ***eficientemente***, el coste mínimo h^* de llegar a una solución a partir de cualquier nodo.

Ejemplo: ***suma de distancias Manhattan*** en 8-puzle



2. Búsqueda en árbol y grafo

EnÁrbolGrafo(n_0, L**)** // nodo inicial y límite de profundidad

$OPEN = \{n_0\}$ // inicialización de la frontera

$CLOSED = \emptyset$ // inicialización del conjunto explorado

bucle

si $OPEN = \emptyset$ **devuelve** NULL // solución no encontrada

$s = \underset{n \in OPEN}{\arg\min}\ f(n)$ // selecciona un nodo de mínima. f

si $Objetivo(s)$ **devuelve** s // ¡solución encontrada!

$OPEN = OPEN - \{s\}$ // elimínalo de la frontera

$CLOSED = CLOSED \cup \{s\}$ // añade al conjunto explorado

si $Profundidad(s) < L$ **para todo** $n \in Hijos(s)$:

sí $n \notin CLOSED$

sí $n \notin OPEN$: $OPEN = OPEN \cup \{n\}$

si no deja en $OPEN$ el de menor f

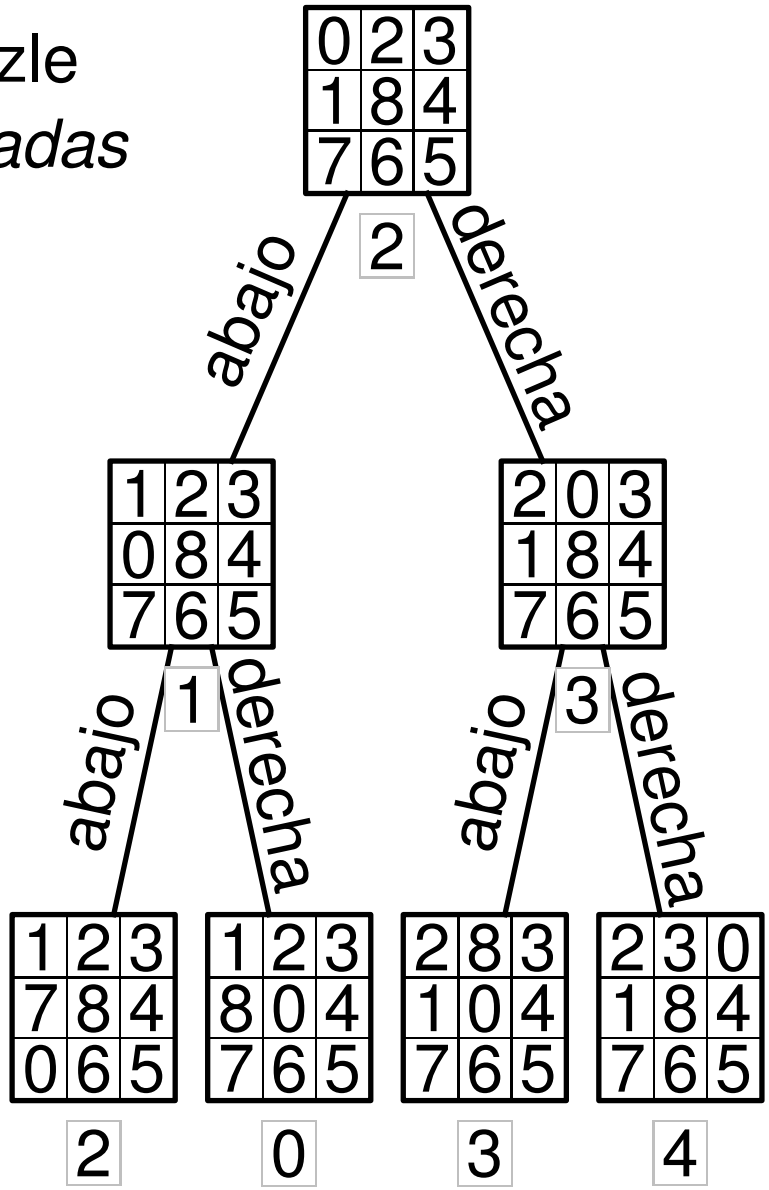
si no si tiene menor f que el de CLOSED:

borra el de CLOSED e inserta n en OPEN

3. Heurística

$h(n)$: coste estimado del camino óptimo desde n a una solución

Ejemplo: 8-puzzle
fichas descolocadas



4. Funciones de evaluación

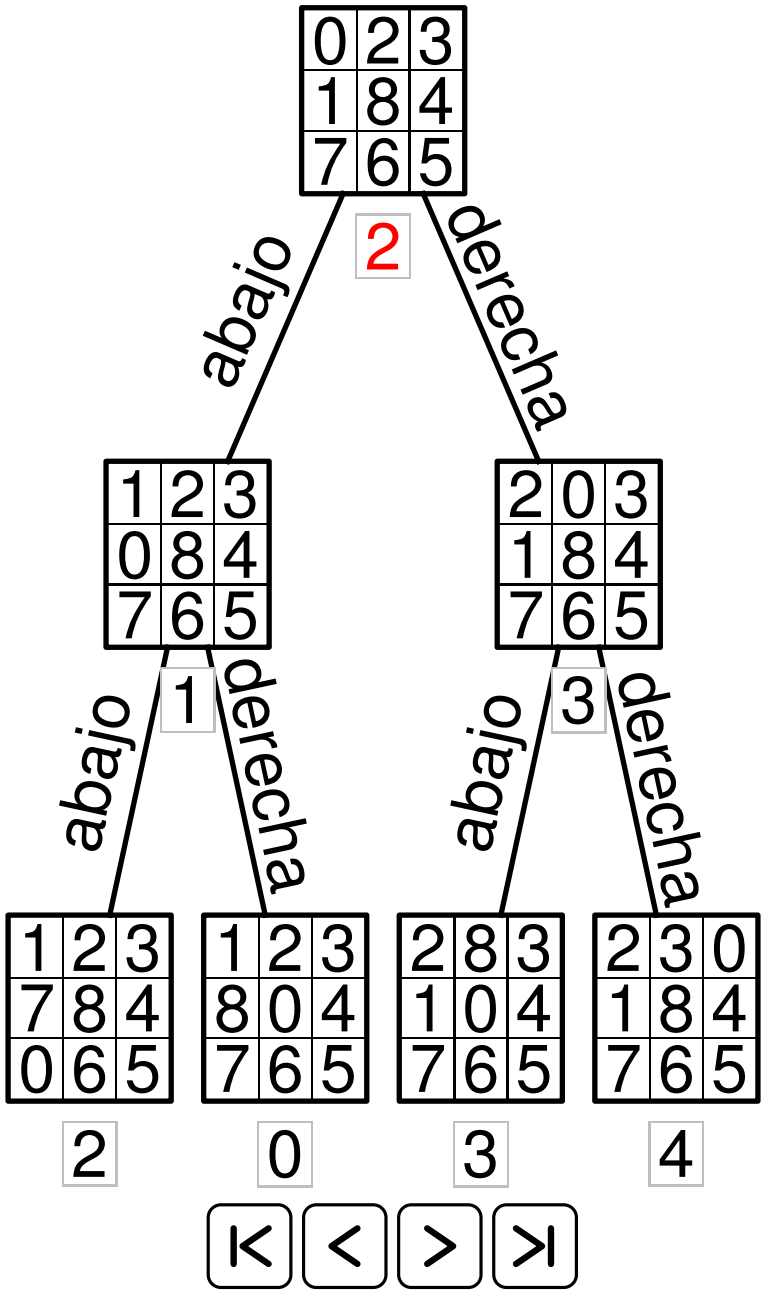
- ***Voraz (primero el mejor):*** la frontera es una ***cola de prioridad (heap)***

$$f(n) = h(n)$$

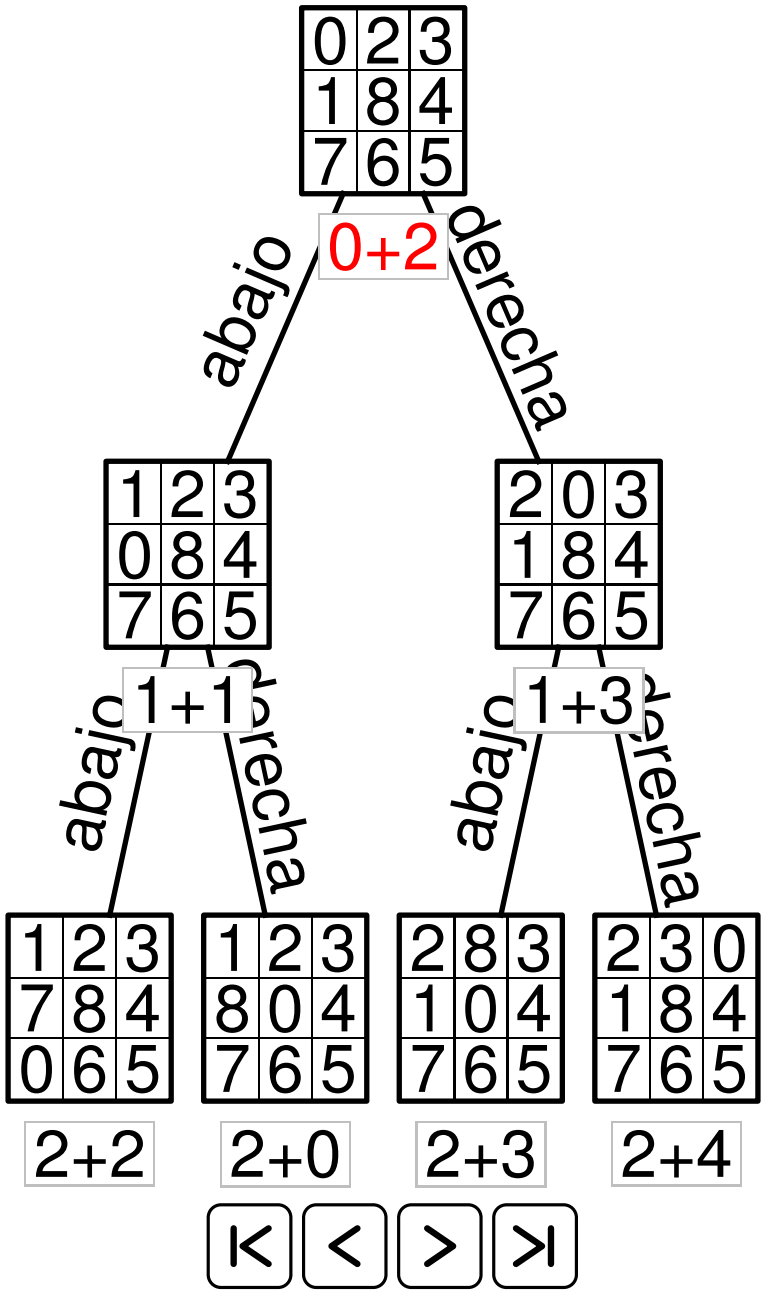
- ***A:*** la frontera es una ***cola de prioridad (heap)***

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

5. Búsqueda voraz: $f(n) = h(n)$



6. Búsqueda A: $f(n) = g(n) + h(n)$



7. Propiedades

De la heurística:

- ***Admisibilidad:*** $h(n) \leq h^*(n)$ para todo n
  - ***Algoritmo A****: búsqueda A con heurística admisible
- ***Consistencia:*** $h(n) \leq c(n, a, n') + h(n')$ para todo n, a y n'
  - Consistencia implica admisibilidad
- ***Dominancia:*** $h_1(n)$ domina $h_2(n)$ si $h_1(n) \geq h_2(n)$ para todo n

Asumiendo acciones de coste positivo y $L = \infty$:

- ***Completitud:*** voraz con búsqueda en grafo y A*
- ***Optimalidad:***
  - A*con búsqueda en árbol o grafo, y $h(n)$ admisible.
  - A*con búsqueda en grafo sin re-expandir y $h(n)$ consistente.
- ***Complejidad:*** $O(b^d)$ temporal; $O(b \cdot d)$ o $O(b^d)$ espacial.