

UNIVERSIDAD
NACIONAL
DE COLOMBIA

Técnicas de Búsqueda Heurística

3007855 - Inteligencia Artificial

3010476 - Introducción a la Inteligencia Artificial

Semestre: 02/2021

Prof. Demetrio Arturo Ovalle Carranza
Departamento de Ciencias de la Computación
y de la Decisión
Facultad de Minas

Noviembre 9 de 2021

LMS: <https://minaslap.net/user/index.php?id=560>

Link Clases: meet.google.com/quy-okvi-ugq



Facultad de Minas
Sede Medellín

Gidia
Grupo de I+D
en Inteligencia Artificial



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Técnicas de Búsqueda Heurística

Las **técnicas de búsqueda heurísticas** (del griego heuriskein, que significa encontrar) están orientados a reducir en árboles y grafos la cantidad de **búsqueda** requerida para encontrar una solución. Dichas técnicas disponen de alguna información sobre la proximidad de cada estado a un **estado objetivo**, lo que permite explorar en primer lugar los caminos más prometedores.

Técnicas de Búsqueda Heurística

- ➔ ♣ Búsqueda el primero mejor, A^*
- ♣ Escalada simple y máxima pendiente
- ♣ Verificación de restricciones

INTRODUCCIÓN

Estrategias de búsqueda simple (sin utilizar heurísticas):

- Profundidad
- Amplitud (Anchura)

El problema de los dos baldes

Tenemos 2 baldes de agua vacíos, uno con capacidad de 6 gal. y otro con capacidad de 8 gal. Teniendo en cuenta que podemos llenar cada uno de los baldes como queramos, ¿cómo podríamos llenar el balde de los 8 gal. exactamente hasta la mitad? (supongamos que no existe ninguna marca de medidas en ninguno de los baldes).

8 gal

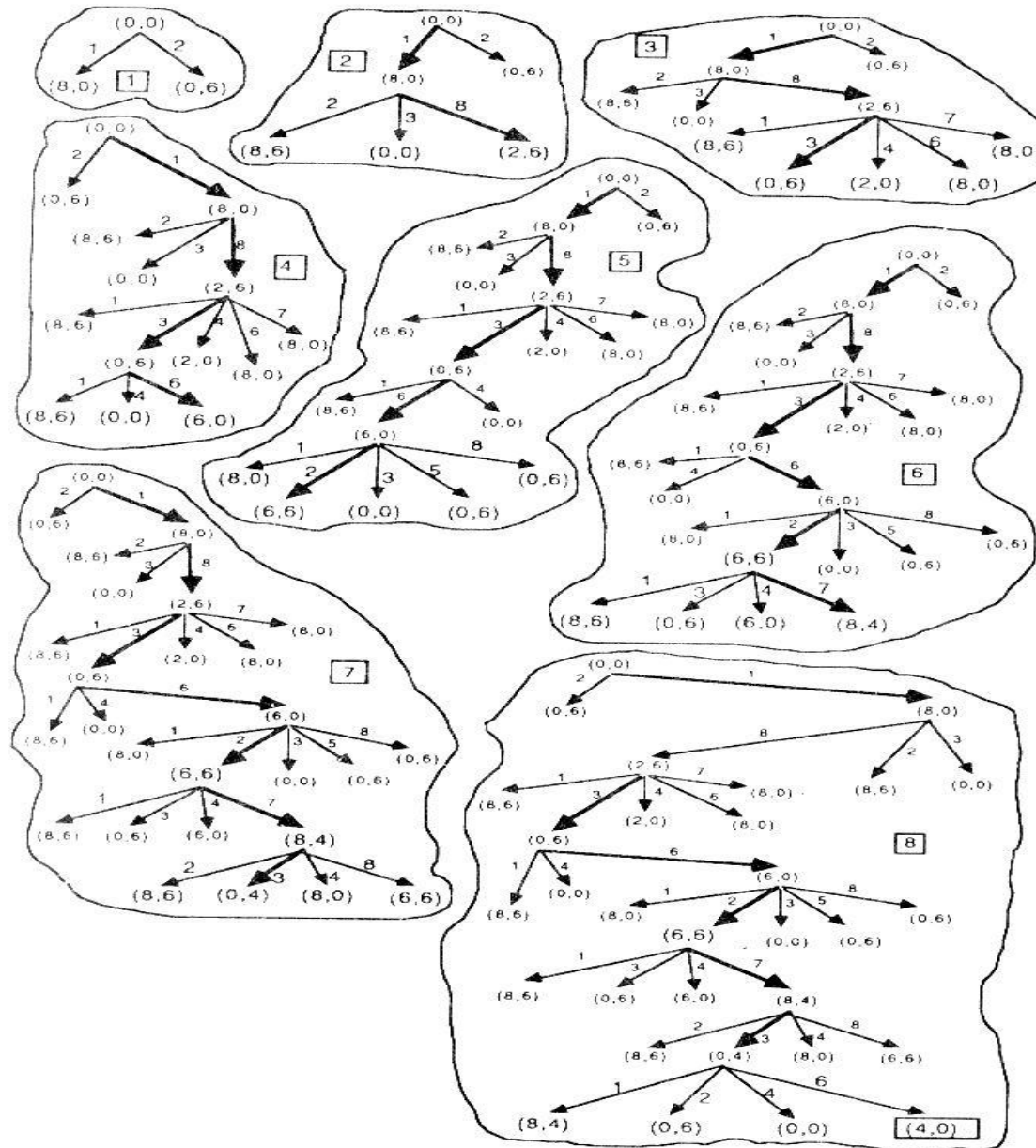


6 gal

El problema de los dos baldes

# Acción	Descripción	Condiciones para su aplicación
1	Llenar el balde de 8 gal.	El balde de 8 gal. no está lleno
2	Llenar el balde de 6 gal.	El balde de 6 gal. no está lleno
3	Descargue el balde de 8 gal.	El balde de 8 gal. no está vacío
4	Descargue el balde de 6 gal	El balde de 6 gal. no está vacío
5	Vacíe el balde de 8 gal. en el de 6 gal.	El balde de 6 gal. no está lleno y el de 8 gal. no está vacío. El contenido conjunto es ≤ 6 gal.
6	Vacíe el balde de 6 gal. en el de 8 gal.	El balde de 8 gal. no está lleno y el de 6 gal. no está vacío. El contenido conjunto es ≤ 8 gal
7	Llene el balde de 8 gal. con el de 6 gal.	El balde de 8 gal. no está lleno y el de 6 gal. no está vacío. El contenido conjunto es ≥ 8 gal
8	Llene el balde de 6 gal. con el de 8 gal.	El balde de 6 gal. no está lleno y el de 8 gal. no está vacío. El contenido conjunto es ≥ 8 gal

Búsqueda Primero en Profundidad



Solución:

$E_0 = (0,0)$

$(8,0)$

$(2,6)$

$(0,6)$

$(6,0)$

$(6,6)$

$(8,4)$

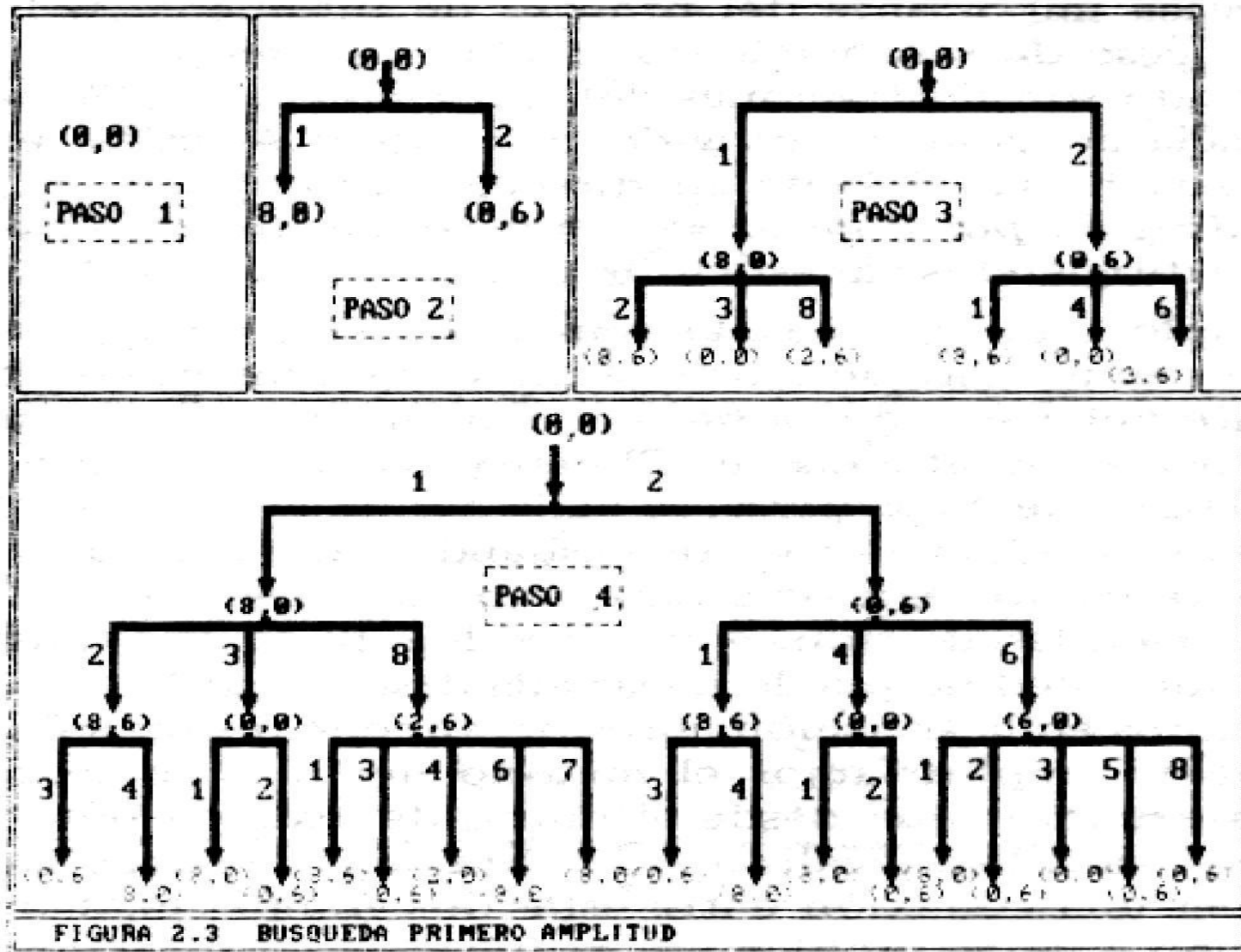
$(0,4)$

$E_f = (4,0)$



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Búsqueda Primero en Amplitud



BÚSQUEDA EL PRIMERO MEJOR (*Best First Search*) A^*

- Los grafos O
- La función
heurística f



CÁLCULO DE LA FUNCIÓN f PARA A^*

$$f = g + h'$$

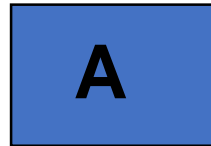
Donde:

g = costo de ir desde el estado inicial hasta el nodo actual.

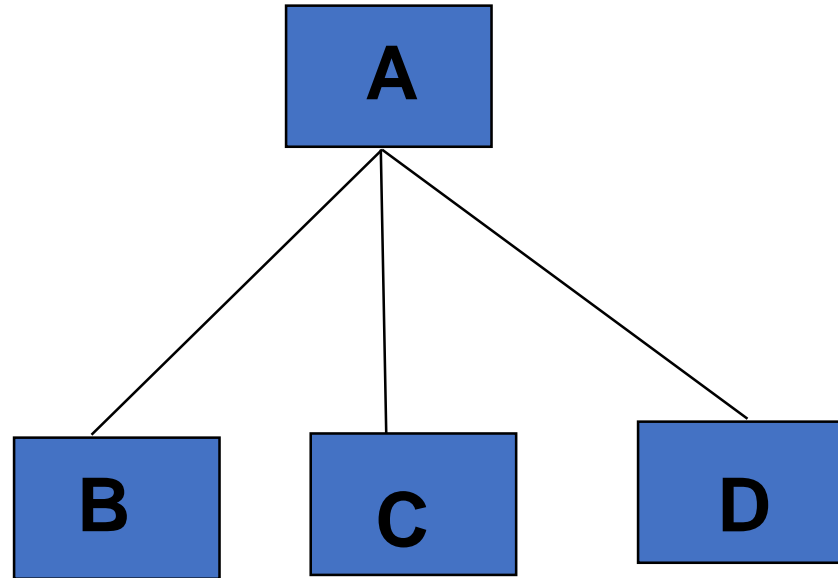
h' = estimación del costo adicional necesario para alcanzar un nodo objetivo a partir del nodo actual.

f = la función combinada del costo necesario para alcanzar un estado objetivo a partir del estado inicial.

Paso 1



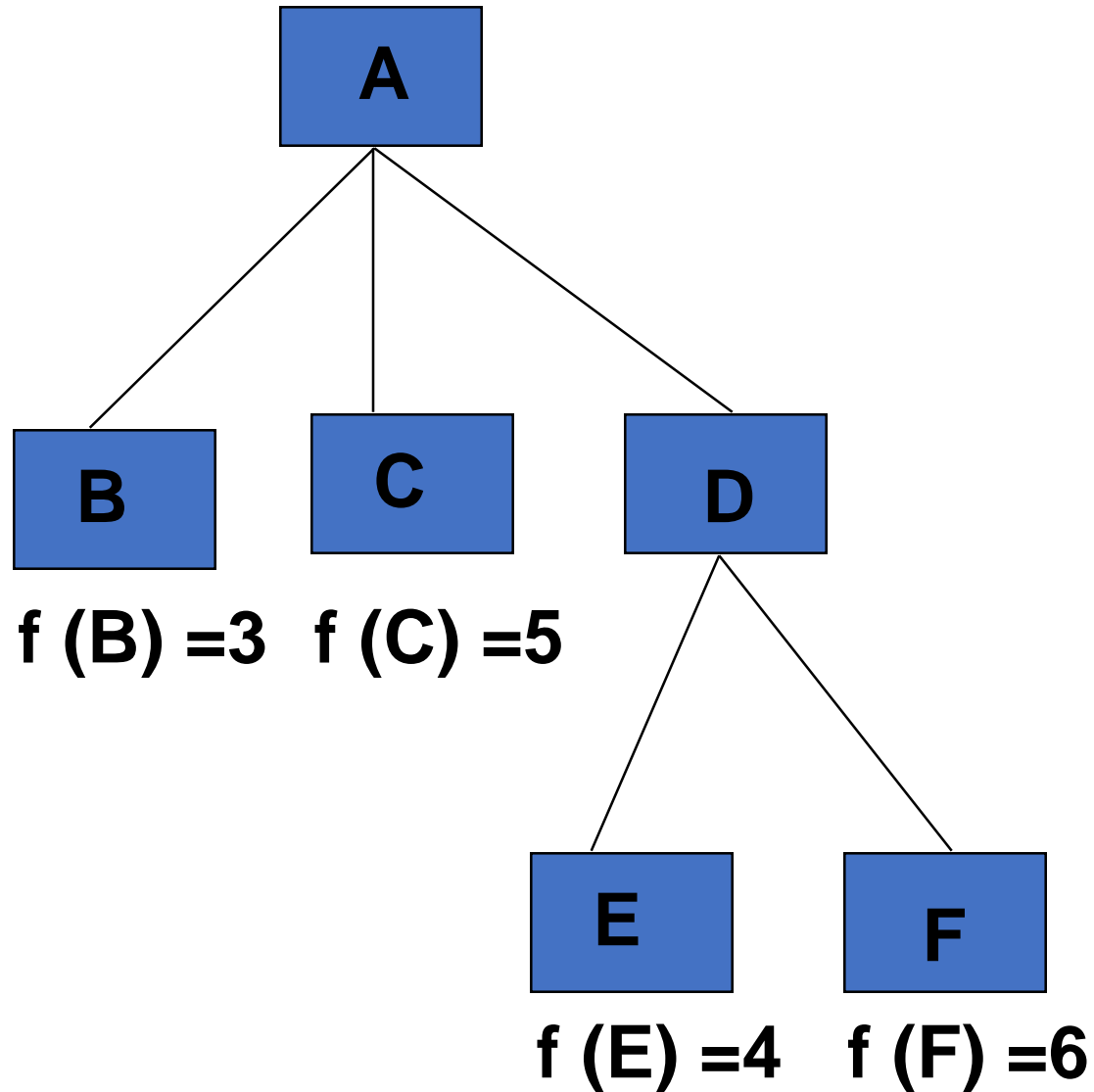
Paso 2



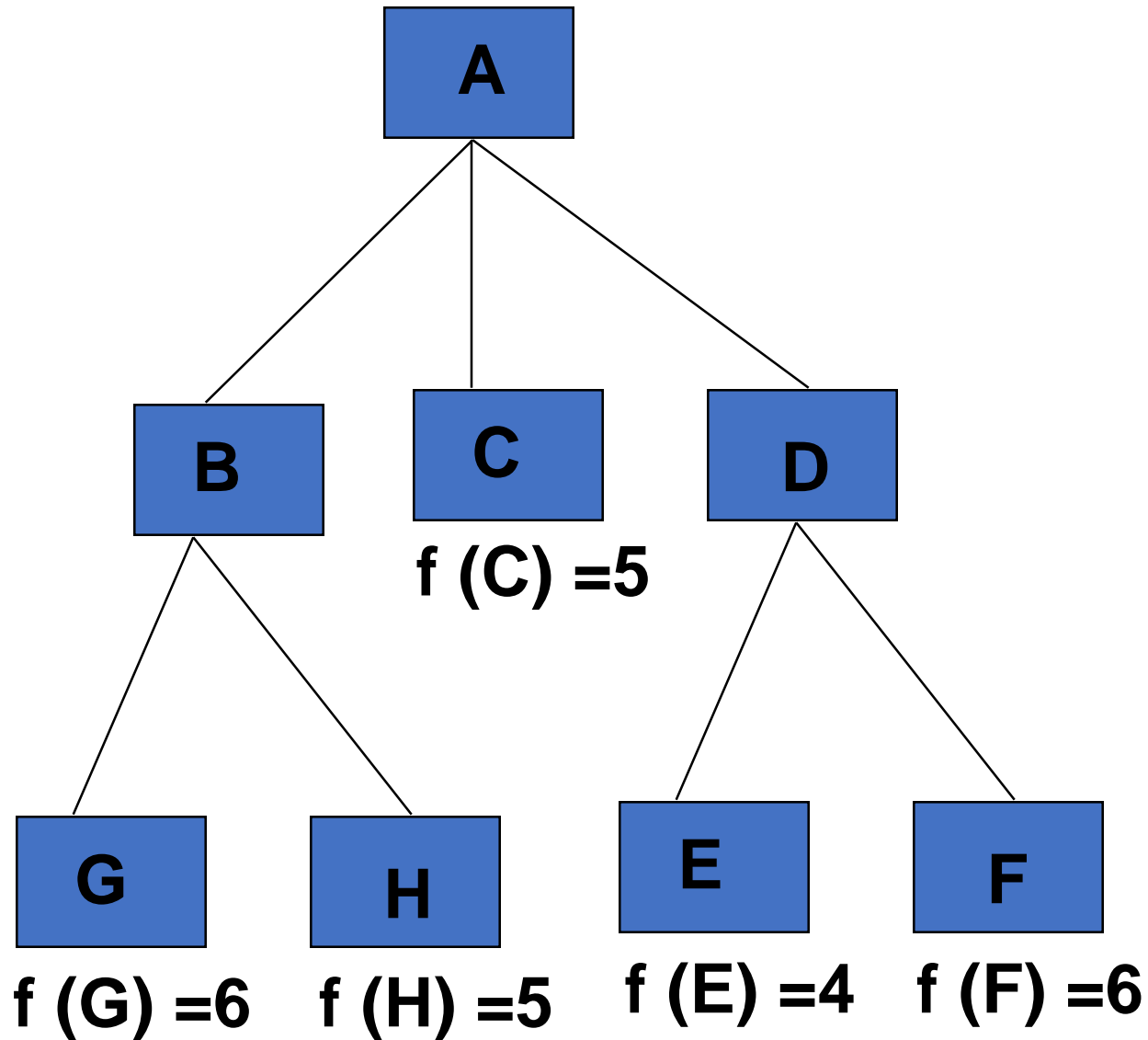
$$f(B) = 3 \quad f(C) = 5 \quad f(D) = 1$$

**f es la función heurística que
representa una estimación del costo
de llegar a una solución a partir de
un Nodo Dado**

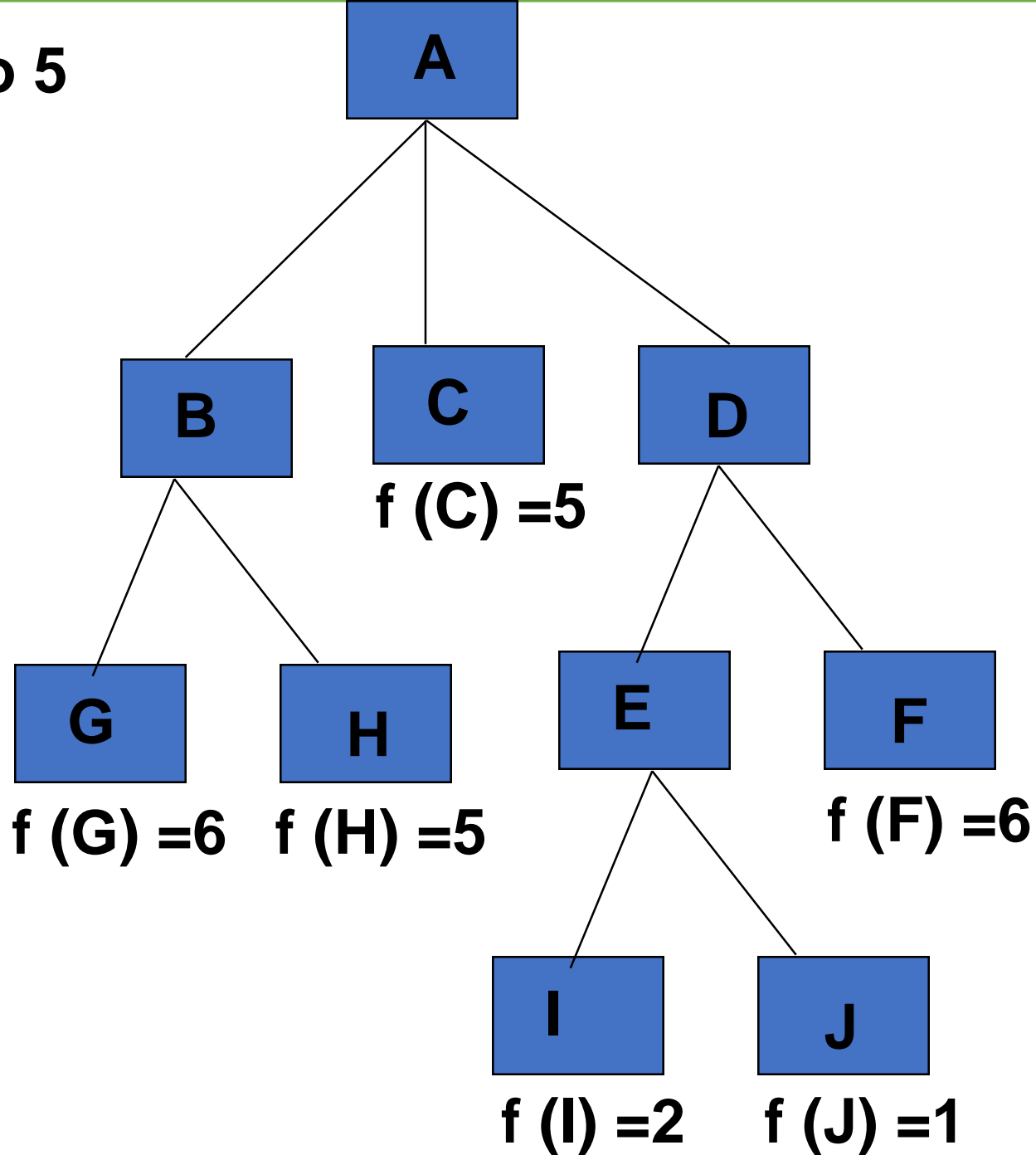
Paso 3



Paso 4

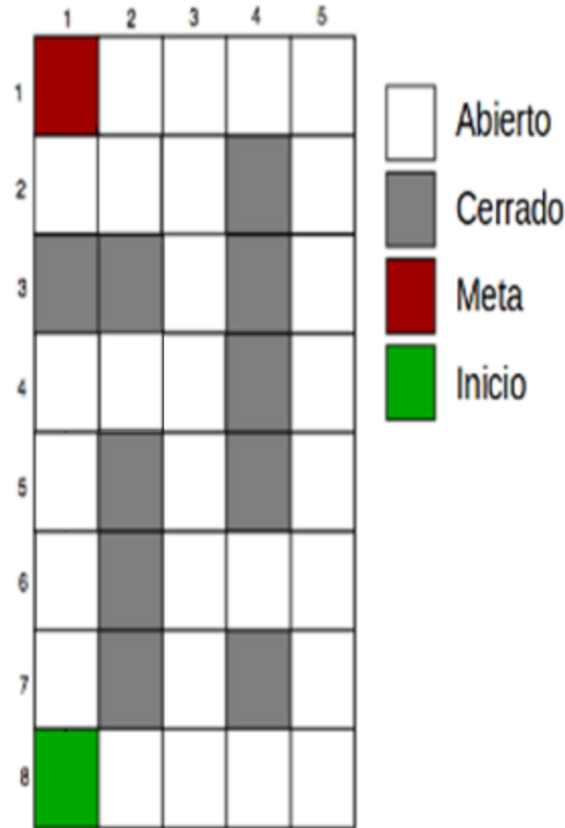


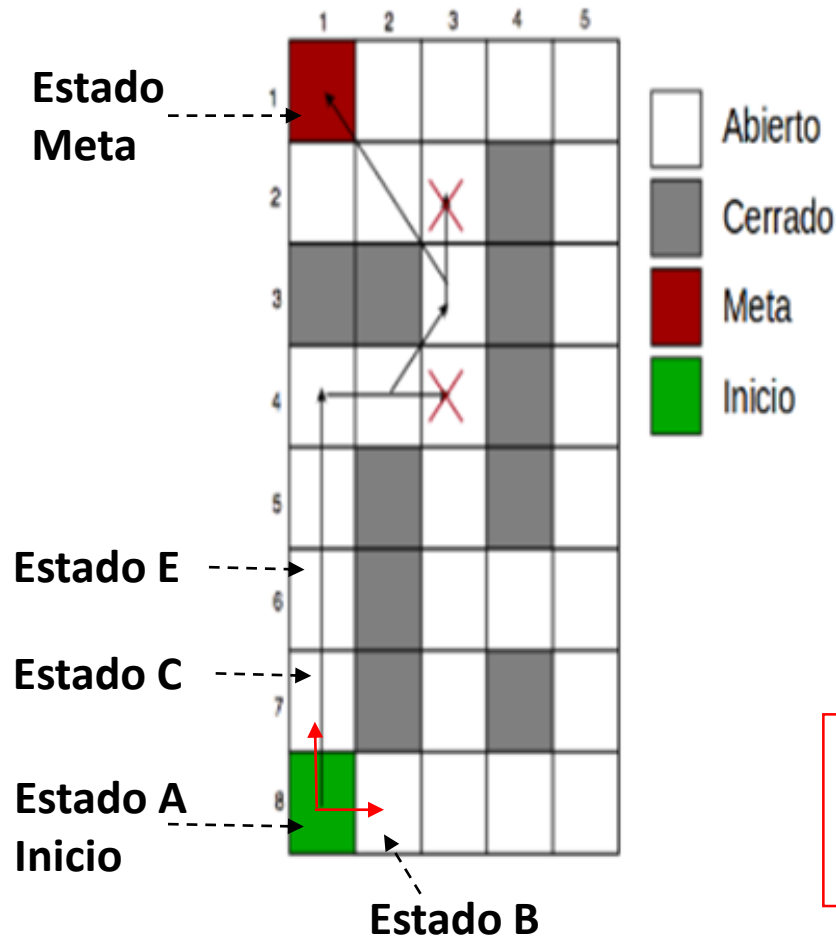
Paso 5



Planificación de trayectorias

*Recorrido
por menor
distancia
Algo. A**





$$f = g + h'$$

Inicio (8,1) = A , Meta (1,1)

(8,2) = B

$g = 1$

$h' = 7$

$f = 8$

(7,1) = C

$g = 1$

$h' = 6$

$f = 7$

(6,1) = E

$f = ?$

(8,1) = A

$f = ?$

(8,2) = B

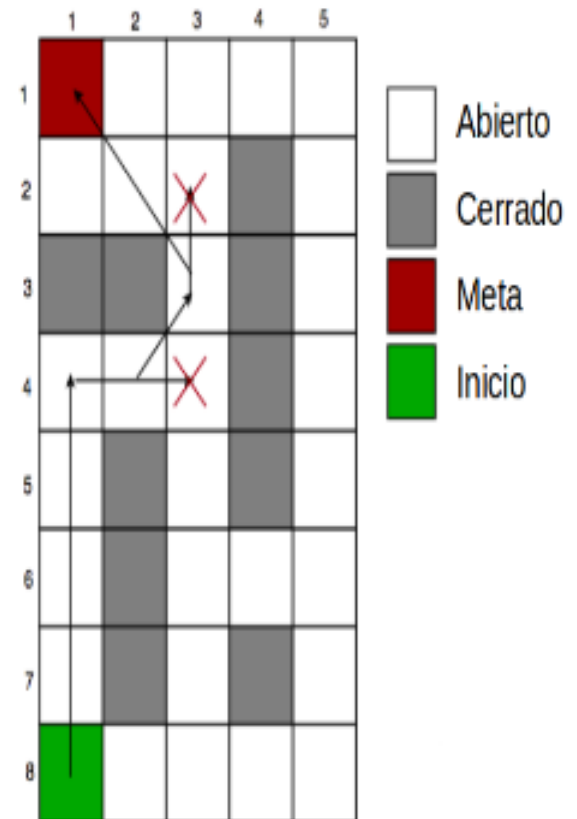
$f = ?$

ALGORITMO DE BÚSQUEDA A*

Según la figura de la derecha, si se requiere llegar a la celda Meta desde la celda Inicio, entonces el algoritmo A*, seguirá el camino como se muestra en la figura.

El algoritmo toma la decisión más inteligente en cada paso, por lo tanto, se puede ver que el algoritmo va desde la celda (4,1) a la celda (3,3) y no a la celda (4,3) marcada con una x.

Igualmente el algoritmo va desde (3,3) a (2,2) y no a (2,3) y así hasta alcanzar la meta.



ESTRUCTURAS PARA GRAFOS O

- **LISTA DE NODOS ABIERTOS :**

Nodos generados, no expandidos a los que se les ha aplicado la función heurística, los cuales, no tienen aún sucesores.

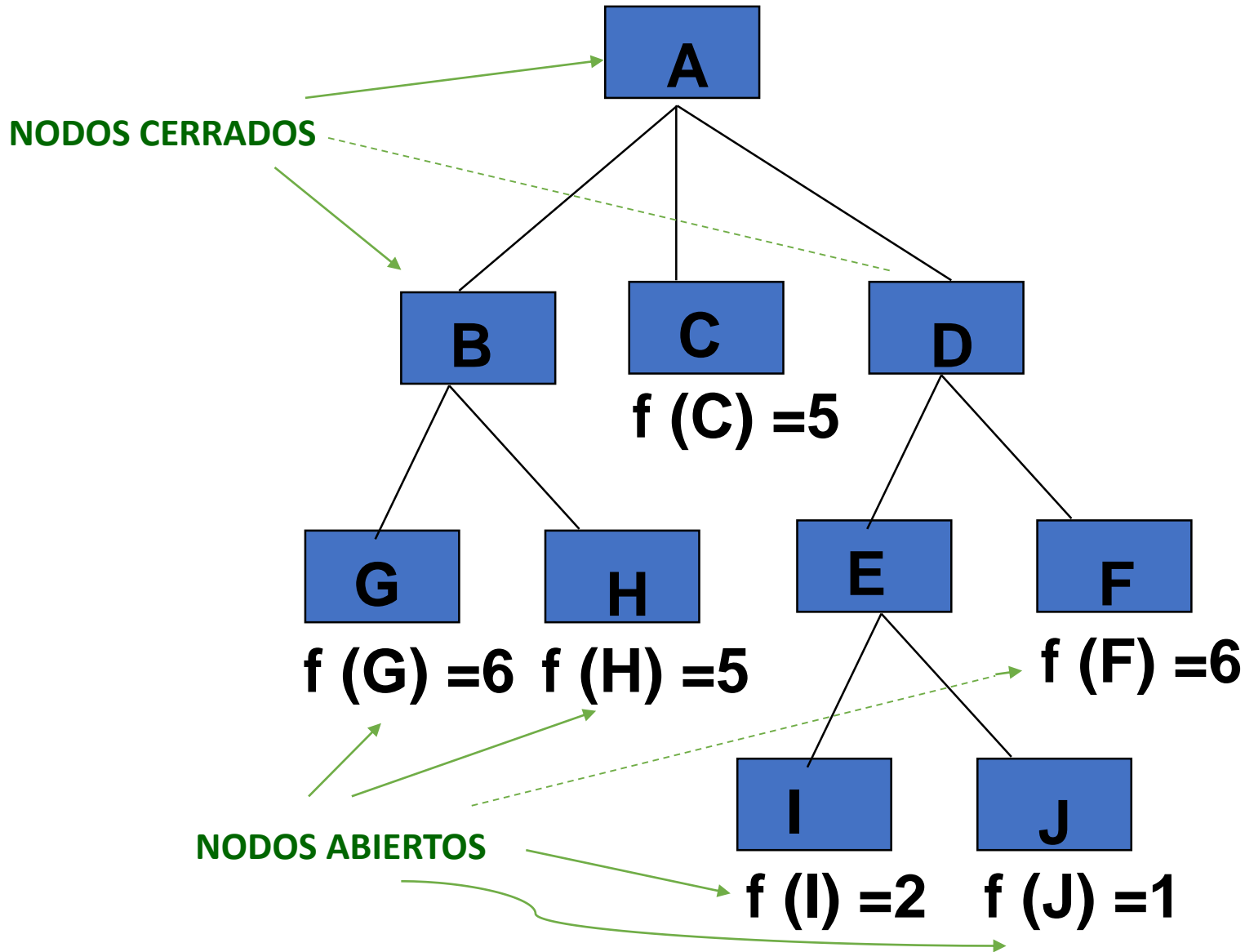
Estructura: cola con prioridad relativa al valor de la func. heurística

- **LISTA DE NODOS CERRADOS:**

Nodos ya examinados.

Guardar en Memoria, en caso de tener nodos repetidos ya generados.

NODOS ABIERTOS Y NODOS CERRADOS



ALGORITMO

ABIERTOS \leq Estado inicial

Hasta Objetivo o ABIERTOS = Vacío

a) Tomar el mejor de ABIERTOS

b) Generar sus sucesores

c) Para cada sucesor hacer:

i. Si no se había generado, evaluar \Rightarrow ABIERTO
almacenar a su padre en CERRADOS

ii. Si ya generado \Rightarrow cambiar al padre (si el nuevo camino es mejor que el anterior).

Actualizar el costo para alcanzar el nodo y los sucesores

CÁLCULO DE LA FUNCIÓN f PARA A^*

$$f = g + h'$$

Donde:

g = costo de ir desde el estado inicial hasta el nodo actual.

h' = estimación del costo adicional necesario para alcanzar un nodo objetivo a partir del nodo actual.

CÁLCULO DE LA FUNCIÓN f PARA A^*

Ejemplo :

SAMAP. Sistema adaptativo multi-agente de planificación dependiente del contexto

DANIEL BORRAJO MILLÁN



Universidad Carlos III de Madrid
Avda. de la Universidad, 30
28911 Madrid, SPAIN

Teléfono:

+(34 91) 624 9459

Email:

dborrajo@ia.uc3m.es

Web:

<http://scalab.uc3m.es/~dborrajo>



Proyecto SAMAP (Borrajo et al.)

El SAMAP (*Sistema Adaptativo Multi-Agente de Planificación dependiente del Contexto*) genera un plan que contiene los siguientes elementos:

- Una selección de los lugares más interesantes para un usuario de acuerdo a su perfil (gustos, intereses, preferencias, etc).
- Indicaciones sobre cuál medio de transporte debería tomar para desplazarse de un lugar a otro, incluyendo desplazamientos a pié.
- Recomendaciones sobre dónde ir a almorzar o cenar (restaurantes, bares, etc.).
- Propuesta de lugares de entretenimiento tales como cines o teatros.

Proyecto SAMAP (Borrajo et al.)

SAMAP utiliza una ontología la cual representa la información necesaria para generar planes de actividades turísticas a un usuario específico:

- Información sobre la ciudad, o sea el contexto que rodea la visita (es decir, monumentos y lugares de interés en la ciudad, medios de transporte, etc)
- Datos personales, intereses y preferencias del usuario, o sea, tipo de actividades que el usuario le gusta cuando visita otras ciudades (modelo del usuario).
- Lugares que personas similares al usuario (perfiles de usuario afines) le gusta visitar.

Proyecto SAMAP (Borrajo et al.)

Planificación de visitas

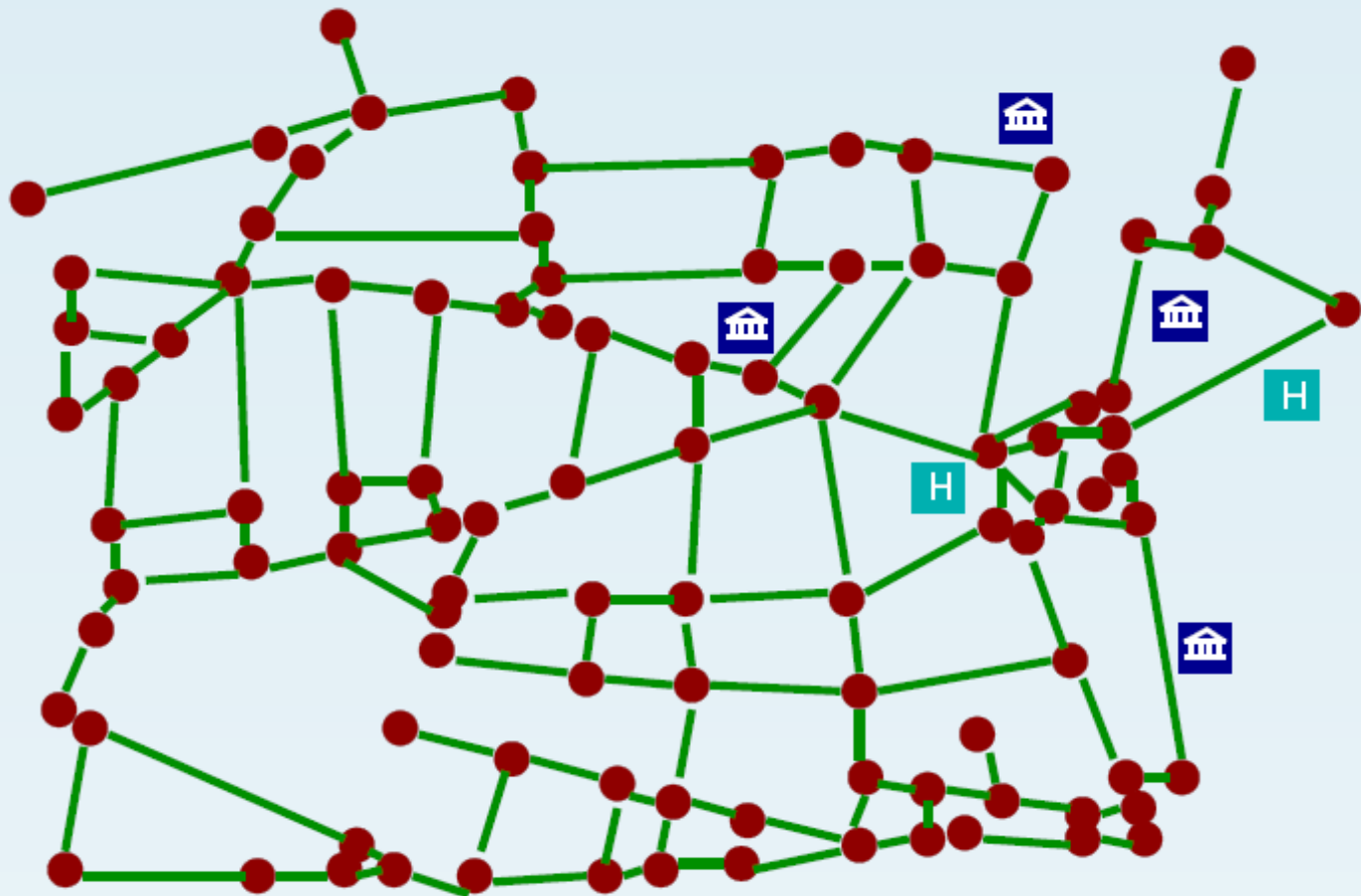
Ejemplo de grafo en Madrid



Proyecto SAMAP (Borrajo et al)

Planificación de visitas

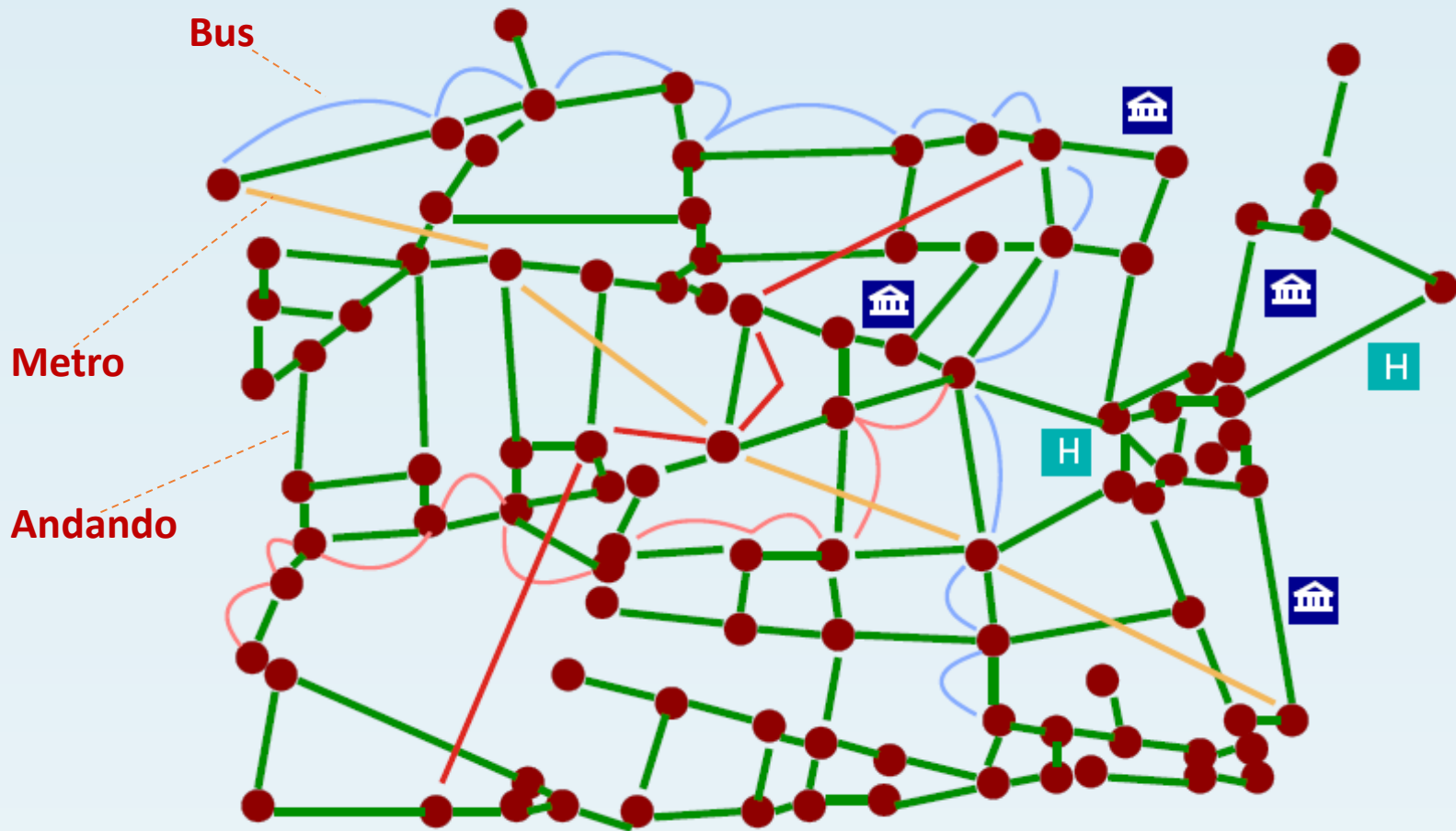
Ejemplo de grafo en Madrid



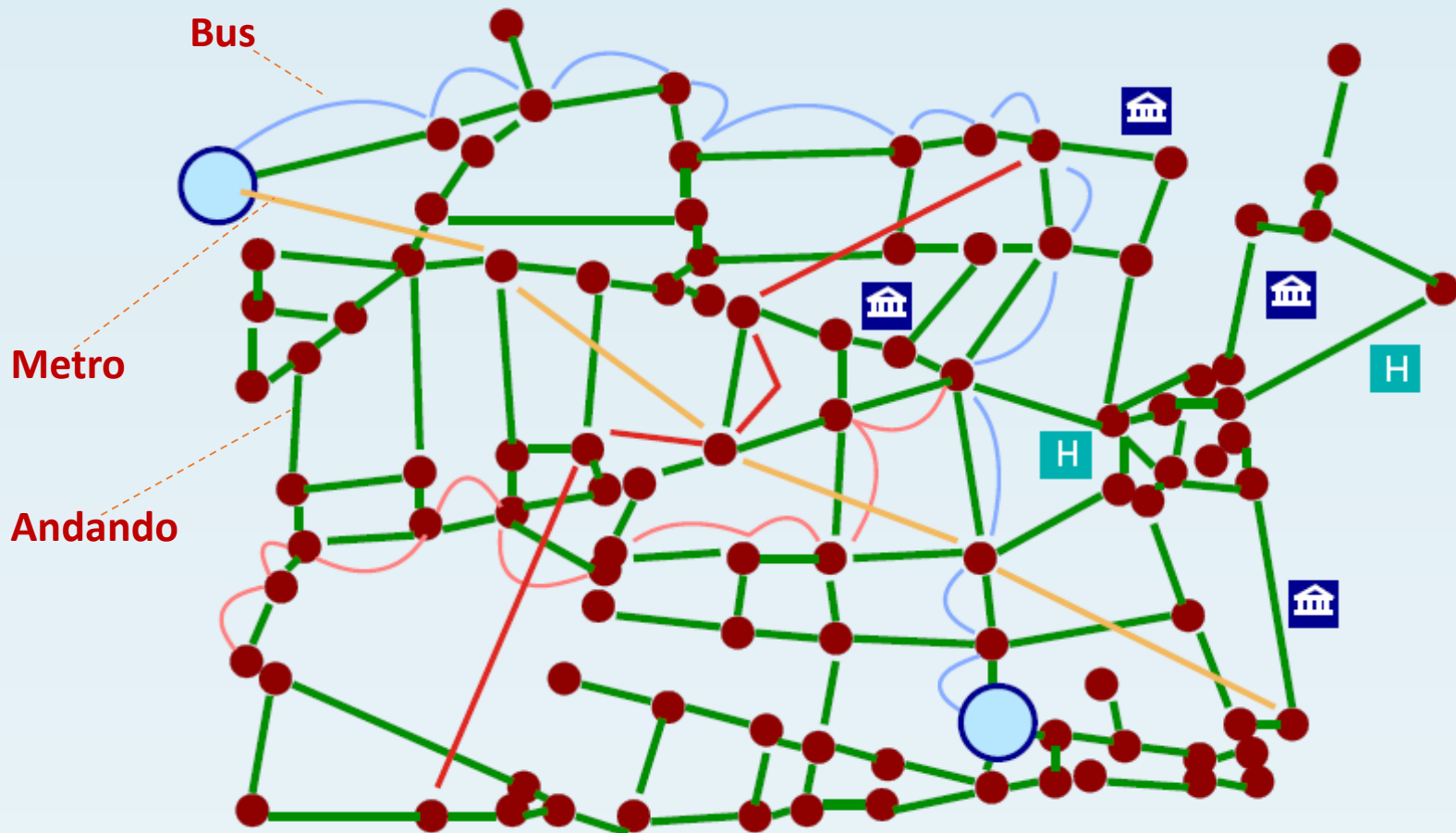
Proyecto SAMAP (Borrajo et al.)

Planificación de visitas

Ejemplo de grafo en Madrid



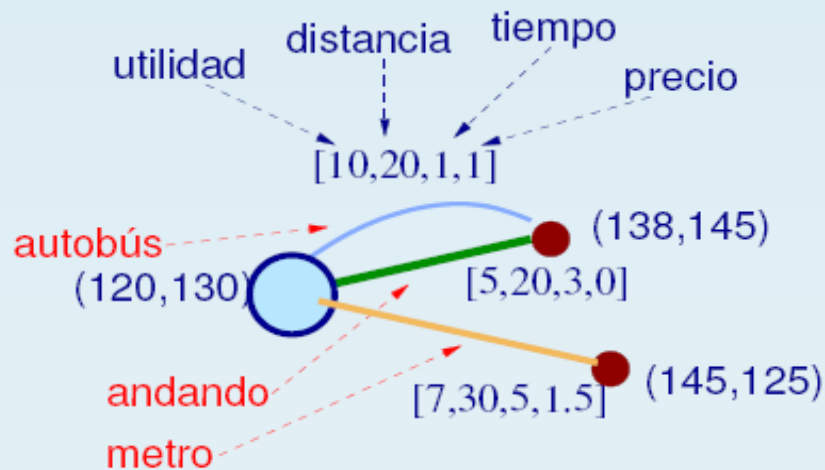
Ejemplo de problema



Proyecto SAMAP (Borrajó et al)

Planificación de visitas

A*

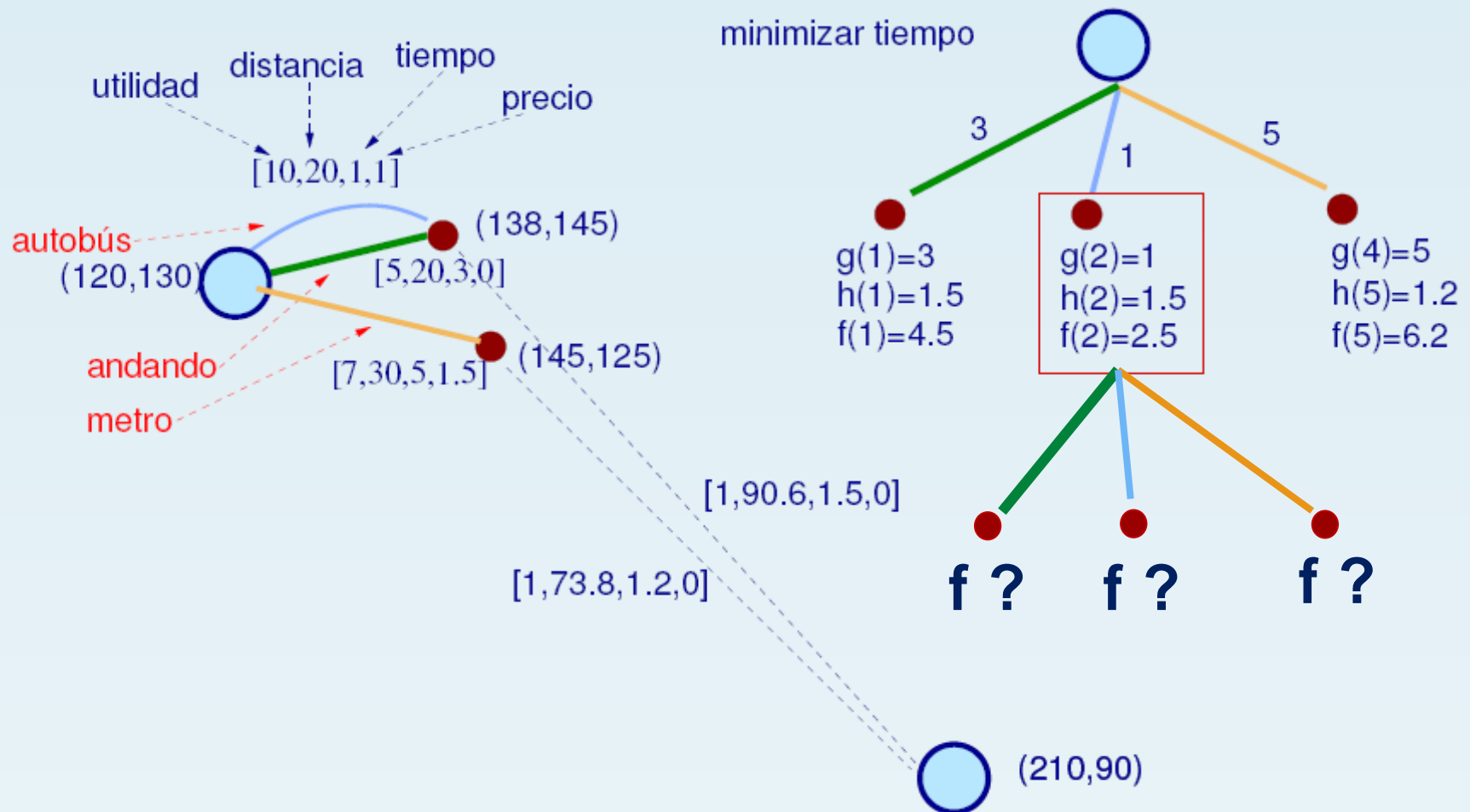


(210,90)

Proyecto SAMAP (Borrajó et al.)

Planificación de visitas

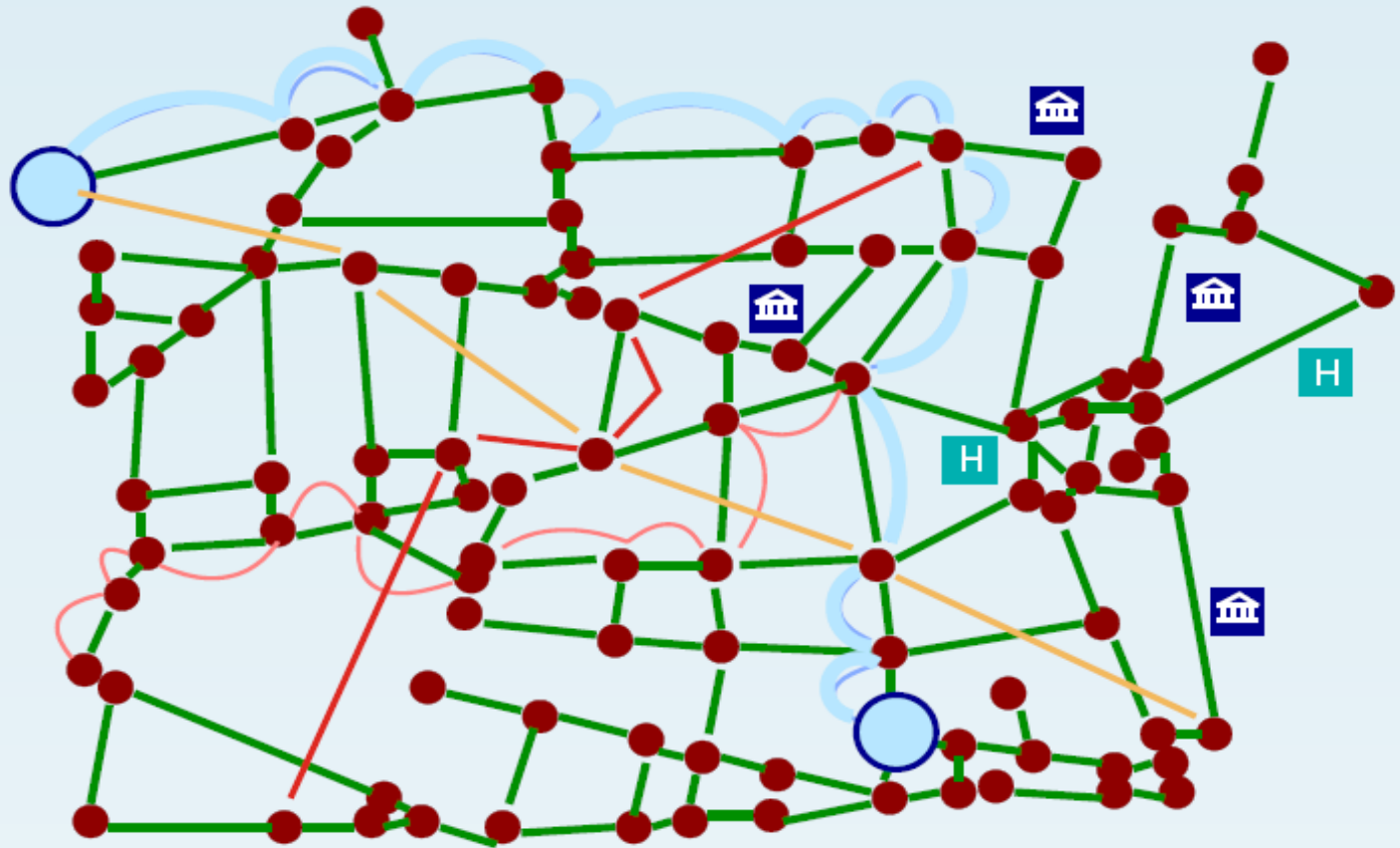
A*



Proyecto SAMAP (Borrajo et al.)

Planificación de visitas

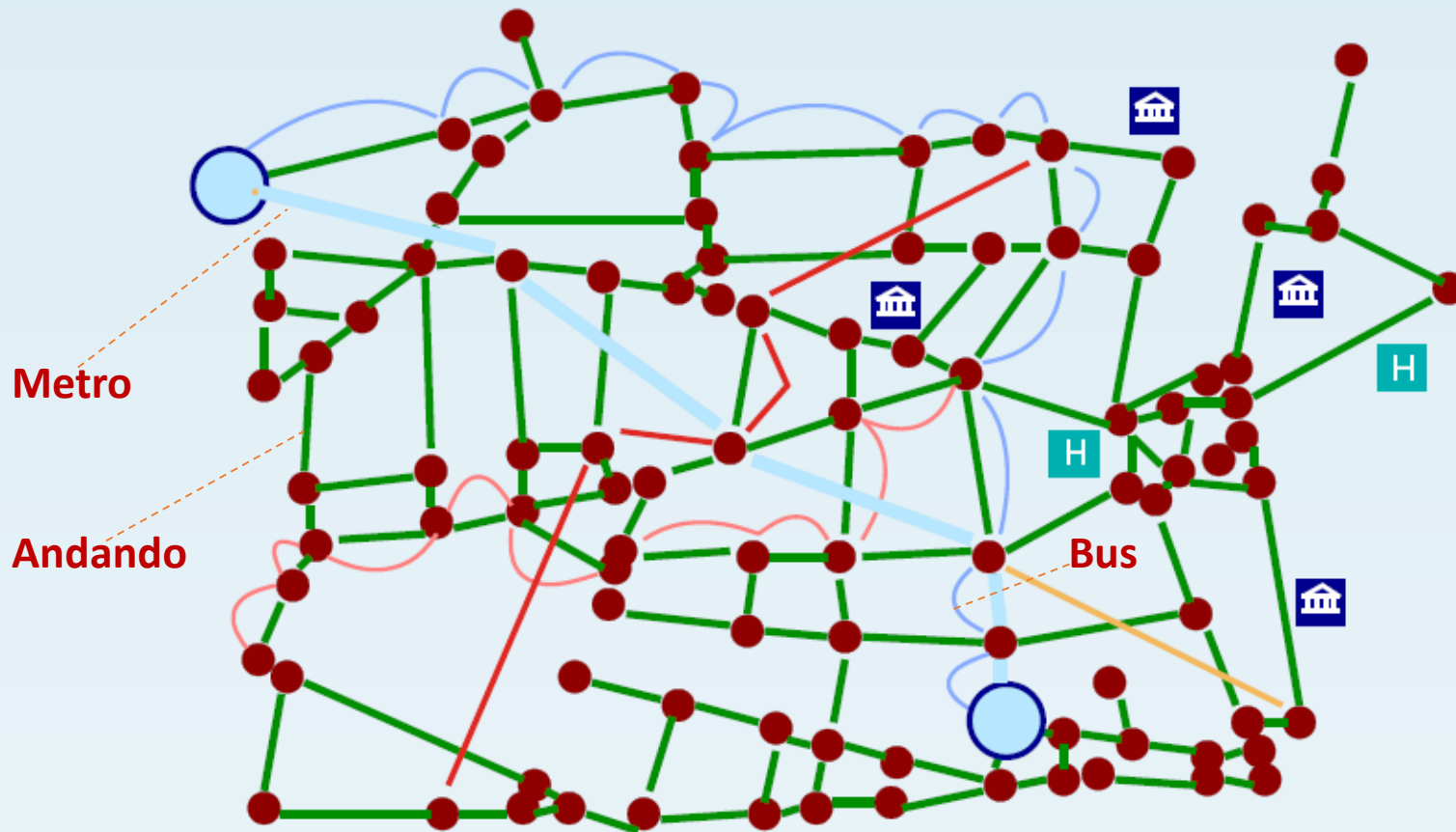
Ejemplo de solución



Proyecto SAMAP (Borrajo et al.)

Planificación de visitas

Ejemplo de solución



Técnicas de Búsqueda Heurística

- ♣ Búsqueda el primero mejor, A^*



- ♣ Escalada simple y máxima pendiente

- ♣ Verificación de restricciones

Escalada (*Hill Climbing*)



- Simple (*Hill climbing*)
- Máxima pendiente
(*Steepest ascent hill climbing or
Gradient search*)

Escalada Simple

Hill Climbing



- Función heurística de proximidad.
- Dirigirse siempre a un estado *mejor* que el actual.
- No se mantiene reporte de estados anteriores.
- Es un método local. Sus movimientos están determinados por ser mejores que los previos.



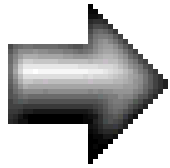
Búsqueda

- Si existe un sucesor s del estado actual n mejor que n , entonces hacer a s como estado actual. De lo contrario, detener.
- Mirar un paso hacia adelante para determinar si algún sucesor es mejor que el estado actual; si lo hay, moverse al mejor sucesor.

Escalada por la Máxima Pendiente

(Steepest ascent hill climbing or

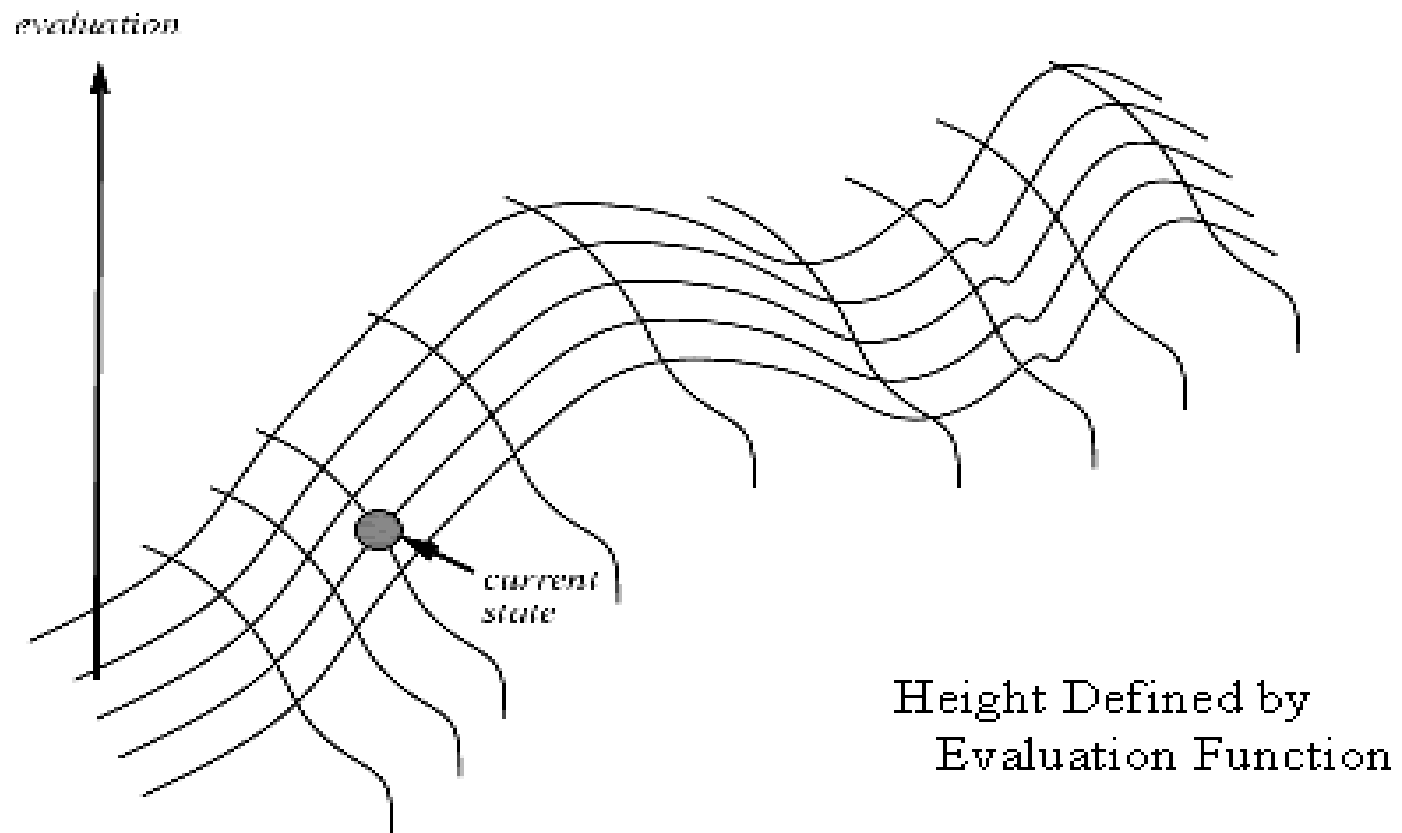
Gradient search)



Buscar no solamente un estado mejor que el actual, sino el mejor de todos estos estados posibles (Máxima pendiente).

Escalada por la Máxima Pendiente

Hill Climbing on a Surface of States

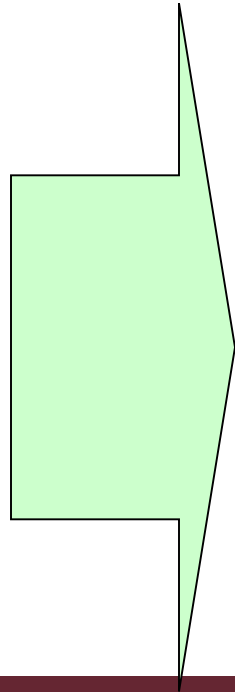




Dificultades de la Escalada

Dificultad

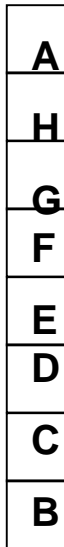
- Máximo local.
- Mesetas.
- Crestas.



Posible Solución

- Backtrack.
- Saltar.
- Moverse en varias direcciones a la vez.

Ejemplo: El mundo de los bloques



Estado Inicial (Eo)



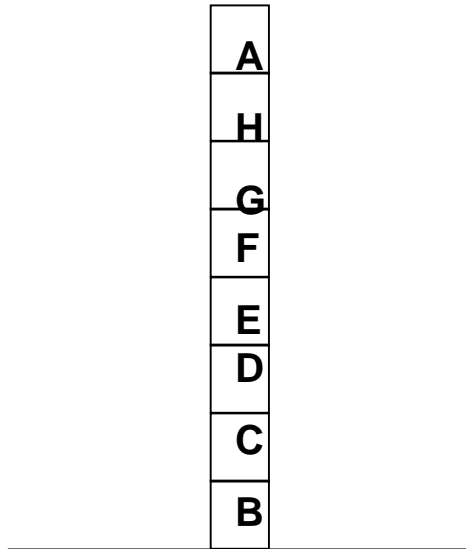
Estado Final (Ef)

¿Cuánto vale $H(Eo)$?

¿Cuánto vale $H(Ef)$?

Función Heurística H : Añadir un punto por cada bloque que esté sobre un bloque o piso correcto. Restar un punto por cada bloque que esté situado en un lugar incorrecto.

Ejemplo: El mundo de los bloques



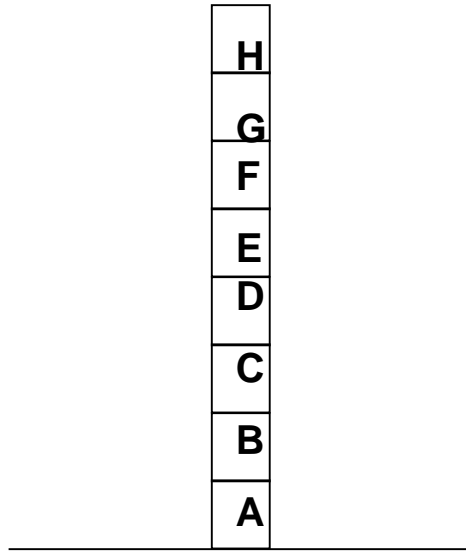
Los bloques A y B
están situados en
lugares incorrectos
entonces:

$$H(E_0) = -1 + 6 - 1 = 4$$

Estado Inicial (E_0)

Función Heurística H : Añadir un punto por cada bloque que esté sobre un bloque o piso correcto. Restar un punto por cada bloque que esté situado en un lugar incorrecto.

Ejemplo: El mundo de los bloques



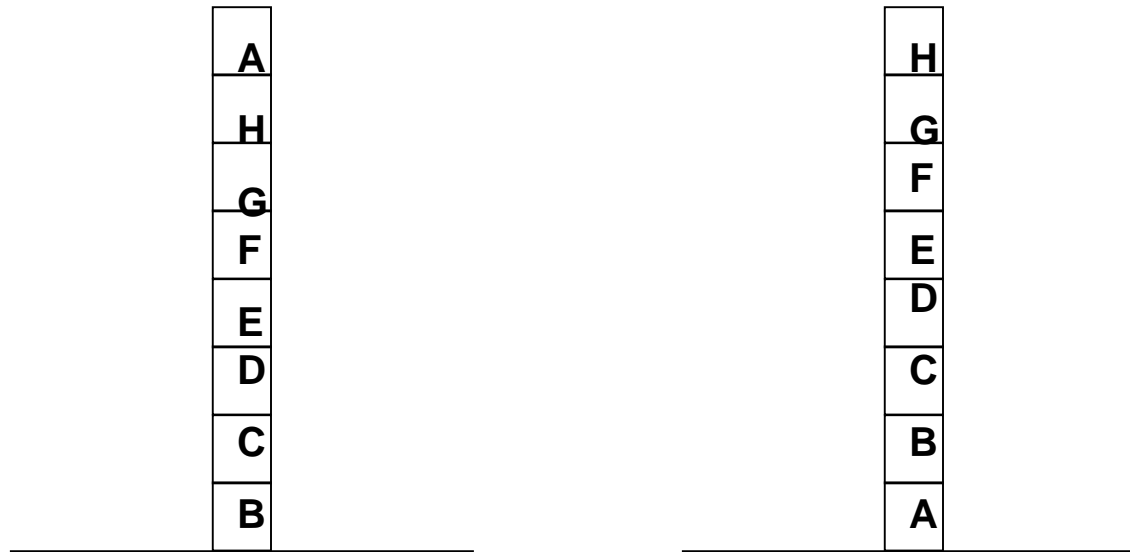
Los bloques A y B
están situados en
lugares incorrectos
entonces:

$$H(Ef) = 8$$

Estado Final (Ef)

Función Heurística H : Añadir un punto por cada bloque que esté sobre un bloque o piso correcto. Restar un punto por cada bloque que esté situado en un lugar incorrecto.

Ejemplo: El mundo de los bloques



Estado Inicial (Eo)--- $H(Eo)=4$ Estado Final (Ef) -- $H(Ef)=8$

Función Heurística H : Añadir un punto por cada bloque que esté sobre un bloque o piso correcto. Restar un punto por cada bloque que esté situado en un lugar incorrecto.

¿Cuánto vale $H(E1)$?

E1

H
G
F
E
D
C
B A

¿Cuánto vale $H(E2)$?

E2

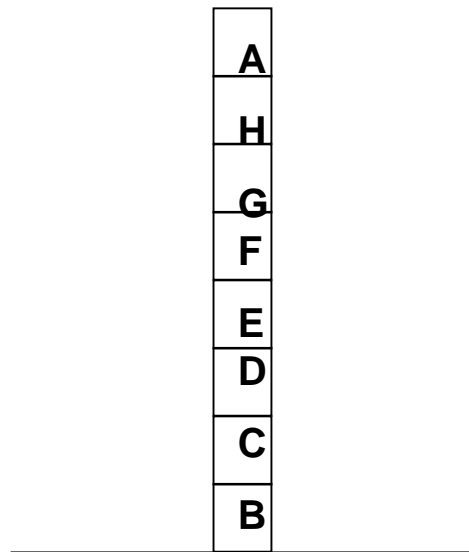
G
F
E
D
C H
B A

¿Cuánto vale $H(E3)$?

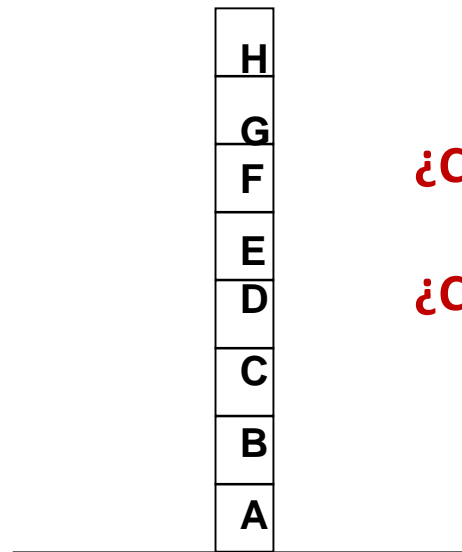
E3

G
F
E
D
C
B A H

Tarea nov 9: El mundo de los bloques, función heurística G.



Estado Inicial (Eo)



Estado Final (Ef)

¿Cuánto vale $G(Eo)$?

¿Cuánto vale $G(Ef)$?

- ❖ Debido a que la función heurística **H** no funciona con la técnica escalada máxima pendiente, definir una función heurística **G** que funcione y muestre el valor obtenido para cada uno de los estados sucesores desde el estado inicial (Eo) hasta llegar al estado Final (Ef).

Técnicas de Búsqueda Heurística

- ♣ Búsqueda el primero mejor, A^*
- ♣ Escalada simple y máxima pendiente
- ➔ ♣ Verificación de restricciones

Verificación de Restricciones

En los problemas de verificación de restricciones el objetivo consiste en descubrir algún estado del problema que satisfaga un conjunto dado de restricciones.

Coloreado de Mapas



Sudoku

	1		4	2				5
		2		7	1		3	9
							4	
2		7	1					6
				4				
6					7	4		3
	7							
1	2		7	3		5		
3				8	2		7	

Criptoaritmética

$$\begin{array}{r}
 \text{S E N D} \\
 + \text{M O R E} \\
 \hline
 \text{M O N E Y}
 \end{array}$$

Ejemplos

Problemas de coloreado de mapas, sudoku, rompecabezas y criptoaritméticos.

El diseño de tareas (tiempo, costo y materiales como limitantes).

¿Cómo funciona?

Es un procedimiento de búsqueda que funciona en un espacio de conjuntos de restricciones. Se realizan restricciones iniciales y se plantean nuevas restricciones, las cuales se van reduciendo conforme la búsqueda se va restringiendo.

Pasos para la búsqueda

La propagación se hace necesaria por el hecho de que normalmente existen dependencias entre las restricciones.

También debido a la presencia de reglas de inferencia que permiten la inferencia de otras restricciones adicionales.

❖ Realizar nuevas hipótesis

- En el caso de que con las restricciones iniciales no se llegue a una solución.
- Después comenzar de nuevo la propagación de restricciones a partir de ese nuevo estado.
- Si se encuentra una solución se muestra.
- Si se detecta alguna contradicción puede usarse vuelta atrás (**backtracking**).

Ejemplo de verificación de restricciones

Criptoaritmética

$$\begin{array}{r} \text{S E N D} \\ + \text{M O R E} \\ \hline \text{M O N E Y} \end{array}$$

Ejemplo de verificación de restricciones

Restricciones Iniciales

- a) $0 \leq \text{letra} \leq 9$
- b) $\text{letra}_i \neq \text{letra}_j$

$$\begin{array}{rccccccc} & C_3 & C_2 & C_1 & \text{---} & & \\ & S & E & N & D & & \\ + & M & O & R & E & & \\ \hline & M & O & N & E & Y & \end{array}$$

Ejemplo de verificación de restricciones

Se puede deducir $M=1$, Suponiendo $S=9 \Rightarrow C2=0, O=0, C1=1$
Suponiendo $E=8, E=7$ y $E=6$ se llega a contradicciones al desarrollar cada rama y se debe hacer entonces backtracking cada vez (vuelta atrás)
También se debe tener en cuenta que $E + 1 = N$

$$M=1, S=?$$

Solución en PROLOG

% Ecuaciones para SEND+MORE=MONEY.

```
solution(S,E,N,D,M,O,R,Y,C1,C2,C3) :-  
  different_digits(S,E,N,D,M,O,R,Y),S>0,M>0,  
  Y is (D+E) mod 10, C1 is (D+E) // 10,  
  E is (N+R+C1) mod 10, C2 is (N+R+C1) // 10,  
  N is (E+O+C2) mod 10, C3 is (E+O+C2) // 10,  
  O is (S+M+C3) mod 10, M is (S+M+C3) // 10.
```

% Restricciones iniciales

```
different_digits(S,E,N,D,M,O,R,Y) :-  
  dig(S),dig(M),dig(O),dig(N),dig(R),dig(D),dig(E),dig(Y),  
  \+ S=M, \+ S=O, \+ S=N, \+ S=R, \+ S=D, \+ S=E, \+ S=Y,  
  \+ M=O, \+ M=N, \+ M=R, \+ M=D, \+ M=E, \+ M=Y,  
  \+ O=N, \+ O=R, \+ O=D, \+ O=E, \+ O=Y,  
  \+ N=R, \+ N=D, \+ N=E, \+ N=Y,  
  \+ R=D, \+ R=E, \+ R=Y,  
  \+ D=E, \+ D=Y,  
  \+ E=Y.
```

% Restricciones iniciales: dígitos 0 a 9

```
dig(0). dig(1). dig(2). dig(3). dig(4).  
dig(5). dig(6). dig(7). dig(8). dig(9).
```

	C3	C2	C1		
	S	E	N	D	
+	M	O	R	E	

	M	O	N	E	Y

solution(S,E,N,D,M,O,R,Y).

42.668 seconds cpu time

D = 7,
E = 5,
M = 1,
N = 6,
O = 0,
R = 8,
S = 9,
Y = 2

C1 = 1,
C2 = 1,
C3 = 0

solution(9,E,N,D,1,O,R,Y).

4.256 seconds cpu time

D = 7,
E = 5,
M = 1,
N = 6,
O = 0,
R = 8,
Y = 2

C1 = 1,
C2 = 1,
C3 = 0

Solución posible

Restricciones Iniciales:

- a) $0 \leq \text{letra} \leq 9$
- b) $\text{letra}_i \neq \text{letra}_j$

D = 7,
E = 5,
M = 1,
N = 6,
O = 0,
R = 8,
S = 9,
Y = 2

$$\begin{array}{r} \textcolor{blue}{1} \quad \textcolor{blue}{0} \quad \textcolor{blue}{1} \quad \textcolor{blue}{1} \quad \textcolor{blue}{\rule{0.5em}{0.4pt}} \\ 9 \quad 5 \quad 6 \quad 7 \\ + \quad 1 \quad 0 \quad 8 \quad 5 \\ \hline 1 \quad 0 \quad 6 \quad 5 \quad 2 \end{array}$$