Inteligencia Artificial
Hoja 2: Búsqueda

Publicación: 2013/01/19
Entrega: 2013/02/26/10:00
Resolución en clase: 2013/02/26

---

1. Utiliza algoritmos de búsqueda para ayudar a completar el cuadrado mágico que el pintor alemán Durero quiere incluir en su grabado *Melancholia I.*

16	3	2	13
5	10		8
9			12
4	15	14	1

Para diseñar este cuadrado mágico se han de colocar los 16 primeros enteros en una matriz cuadrada de forma que todas las filas, columnas y las dos diagonales principales sumen a 34.

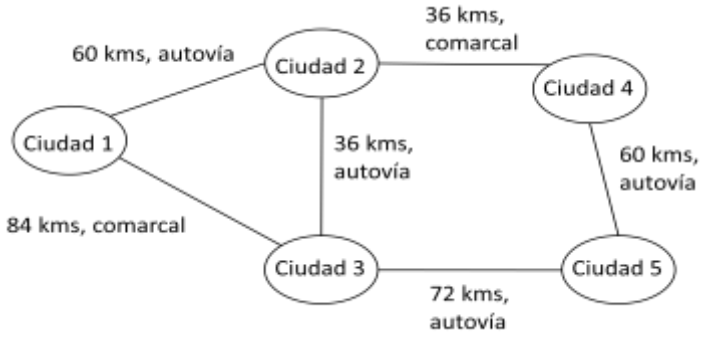
1.1 Formaliza el problema especificando
(i) ¿Cuál es la formalización del estado inicial?
(ii) ¿Cuál la formalización de los estados de búsqueda?
(iii) Formaliza los operadores que se utilizan para generar sucesores
(iv) Describe el test que determina si un estado de búsqueda no puede conducir a la solución porque no cumple alguna de las restricciones del problema.
(v) Describe el test que determina si el estado de búsqueda cumple el objetivo de la búsqueda.

1.2 Dibuja el árbol de búsqueda, utilizando búsqueda en profundidad, indicando el orden de expansión de los nodos. Además de indicar el orden, etiqueta como inviables aquellos nodos que no pueden conducir a la solución.
Recuerda que los tests diseñados sólo pueden ser aplicados al intentar expandir un nodo, no cuando éste es generado.

Inteligencia Artificial
Hoja 2: Búsqueda

Publicación:	2013/01/19
Entrega:	2013/02/26/10:00
Resolución en clase:	2013/02/26

2. Consideremos el siguiente mapa de ciudades, donde cada nodo representa una ciudad, y cada arco una carretera que conecta directamente dos ciudades (asociados a cada arco tenemos la longitud de dicha carretera y el tipo de carretera):



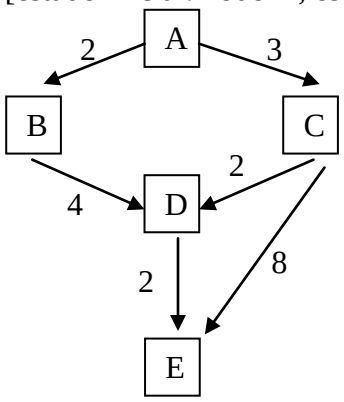
Hay dos tipos de carreteras: comarcales y autovías. En cada una de estas se estima que la velocidad promedio va a ser diferente (60 km/h y 100 km/h respectivamente). Nuestro objetivo es llegar desde la ciudad 1 a la ciudad 5 por el camino por el que tardemos menos tiempo en nuestro coche.
Como información adicional disponemos de la distancia en línea recta de cada una de las ciudades a la objetivo:

Ciudad	Distancia a ciudad 5 (kms)
1	120
2	90
3	60
4	42
5	0

Responde a las siguientes cuestiones:

- Define una heurística admisible para este problema y muestra los valores de esta h
- Determina, para cada estado, el coste para ir de dicho estado a cada uno de sus posibles sucesores.
- ¿A qué solución llegaría búsqueda-en-árbol (es decir, sin eliminar estados repetidos) usando A* y esa heurística? Muestra paso a paso qué nodos expandiría el algoritmo, cuáles genera, y en qué orden. ¿Qué solución encuentra? ¿es la óptima? ¿cuántas horas tardaría nuestro coche en llegar a la ciudad objetivo por esta trayectoria?

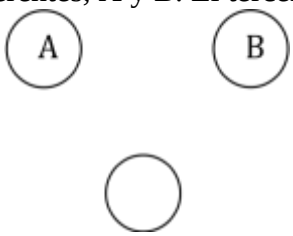
3. Proporciona un ejemplo de una heurística que sea admisible pero no monótona para la búsqueda en el siguiente grafo [estado inicial: nodo A; estado final: nodo E].



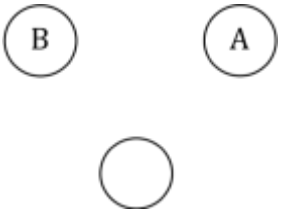
Inteligencia Artificial
Hoja 2: Búsqueda

Publicación: 2013/01/19
Entrega: 2013/02/26/10:00
Resolución en clase: 2013/02/26

4. En una fábrica disponemos de tres contenedores. Dos de ellos están inicialmente llenos de dos substancias diferentes, A y B. El tercero está vacío:



El objetivo es el siguiente estado



No se pueden mezclar las sustancias A y B. Las tres acciones posibles son:

- Mover en un paso el contenido de los tres contenedores **en el sentido de las agujas del reloj** (coste = 3), por ejemplo:

- Mover A al contenedor vacío (coste = 1)
- Mover B al contenedor vacío (coste = 1)

  El orden en el que se aplican estos operadores para generar sucesores es este mismo (1º, mover en el sentido de las agujas del reloj; 2º, mover A al contenedor vacío; 3º, mover B al contenedor vacío).

  En caso de que haya varios estados con la misma valoración, se explorará antes el estado que se haya generado antes.

4.1 Define una heurística para este problema utilizando el método de relajación, y que no sea trivial (h=0 en todos los estados).
4.2 ¿Es dicha heurística admisible? ¿Por qué?
4.3 ¿Cuál es la solución encontrada por A* con búsqueda en árbol (sin eliminar estados repetidos)? ¿Es la solución óptima?

Inteligencia Artificial
Hoja 2: Búsqueda

Publicación: 2013/01/19
Entrega: 2013/02/26/10:00
Resolución en clase: 2013/02/26

---

5. Consideremos el problema de búsqueda en un espacio con estados A, B, C, D, E. Las acciones posibles en este espacio son:

origen	destino	Coste
A	B	3
A	C	2
B	C	5
B	D	3
C	D	5
C	B	1
C	E	11
D	E	5
E	B	8

Consideremos las heurísticas

nodo	h_1 (nodo)
A	8
B	6
C	6
D	4
E	0

nodo	h_2 (nodo)
A	7
B	8
C	5
D	5
E	0

Cuestiones: (justificar las respuestas

a) ¿El algoritmo A* es óptimo con h_1?
b) ¿El algoritmo A* es óptimo con h_2?
c) ¿Son admisibles las siguientes heurísticas? (para cada nodo n se toma como valor de la heurística)
   i. $h_1(n)$
   ii. $h_2(n)$
   iii. $\min(h_1(n), h_2(n))$
   iv. $h_1(n) + h_2(n)$
   v. $(h_1(n) + h_2(n))/2$
   vi. $\max(h_1(n), h_2(n))$
   vii. $h_1(n) \cdot h_2(n)$
   viii. $(h_1(n) \cdot h_2(n))^{1/2}$
d) ¿De entre las heurísticas admisibles, hay alguna dominante?

Inteligencia Artificial
Hoja 2: Búsqueda

Publicación: 2013/01/19
Entrega: 2013/02/26/10:00
Resolución en clase: 2013/02/26

---

1. Utiliza búsqueda para ayudar a completar el cuadrado mágico que el pintor alemán Durero quiere incluir en su grabado *Melancholia I.*

16	3	2	13
5	10		8
9			12
4	15	14	1

Para diseñar este cuadrado mágico se han de colocar los 16 primeros enteros en una matriz cuadrada de forma que todas las filas, columnas y las dos diagonales principales sumen a 34.

1.1 Formaliza el problema especificando

(i) ¿Cuál es la formalización del estado inicial?

16	3	2	13
5	10	x_{23}	8
9	x_{32}	x_{33}	12
4	15	14	1

Con $x_{23}, x_{32}, x_{33} \in \{6,7,11\}$

(ii) ¿Cuál la formalización de los estados de búsqueda?

16	3	2	13
5	10	x_{23}	8
9	x_{32}	x_{33}	12
4	15	14	1

con algunas o todas las variables x_{23}, x_{32}, x_{33} instanciadas con valores distintos elegidos de entre los del conjunto {6,7,11}.

(iii) Formaliza los operadores que se utilizan para generar sucesores

Instanciar alguna de las variables aún no instanciadas con valores dentro del conjunto {6,7,11} que aún no hayan sido utilizados en la instanciación de otra variable.

En cada nivel se instancia una de las variables en el orden x_{23}, x_{32}, x_{33}.
Para cada variable se inicializarán con valores en orden creciente.

(iv) Describe el test que determina si un estado de búsqueda no puede conducir a la solución porque no cumple alguna de las restricciones del problema.

Los elementos de alguna de las filas, columnas y diagonales principales completas no suman 34.

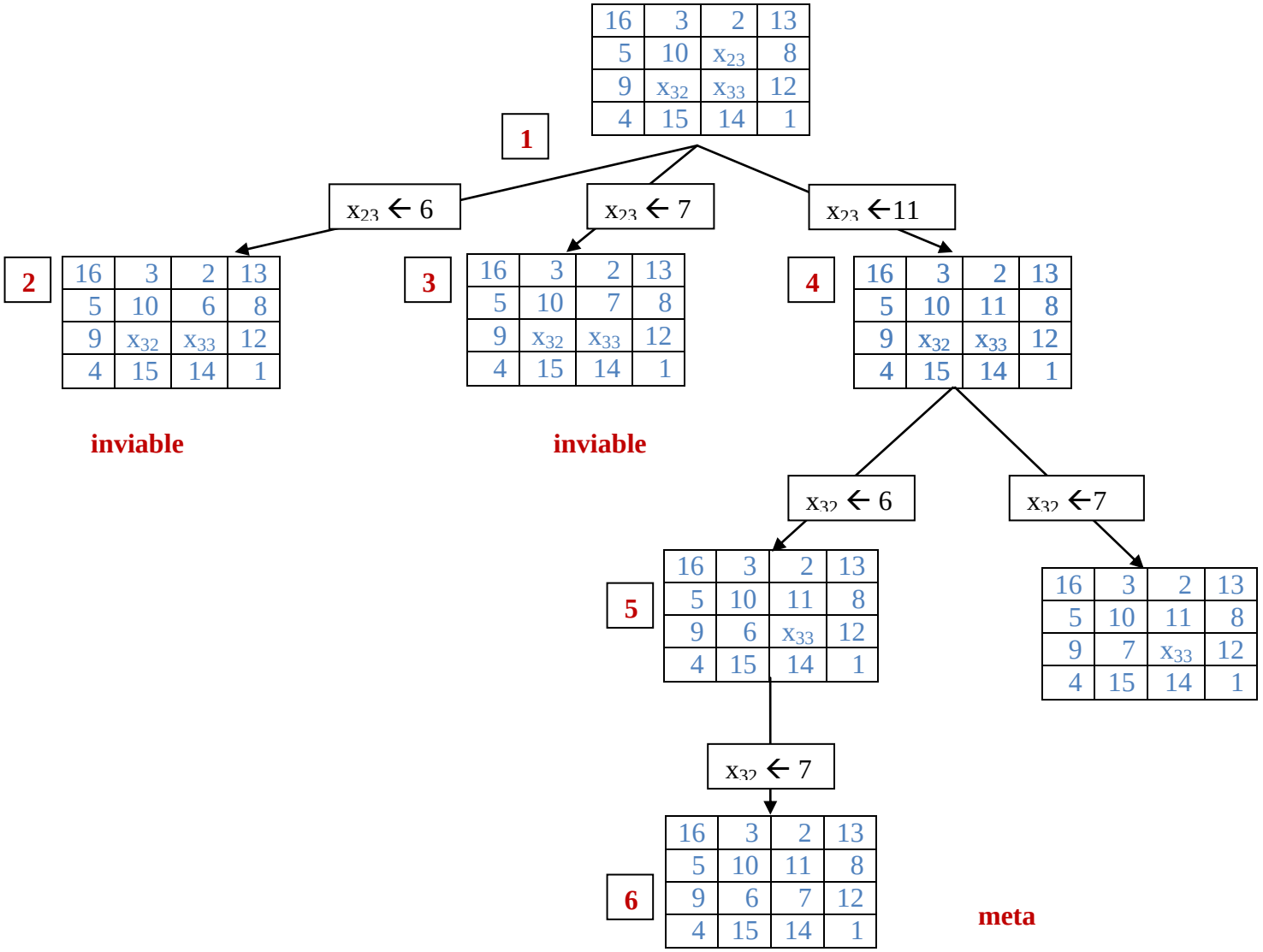
(v) Describe el test que determina si el estado de búsqueda cumple el objetivo de la búsqueda.

La matriz está completa.
Los elementos de todas las filas, columnas y diagonales principales suman 34.

1.2 Utiliza el árbol de búsqueda, utilizando búsqueda en profundidad, indicando el orden de expansión de los nodos. Además de indicar el orden, etiqueta como inviables aquellos nodos que no pueden conducir a la solución.
Recuerda que los tests diseñados sólo pueden ser aplicados al intentar expandir un nodo, no cuando éste es generado.

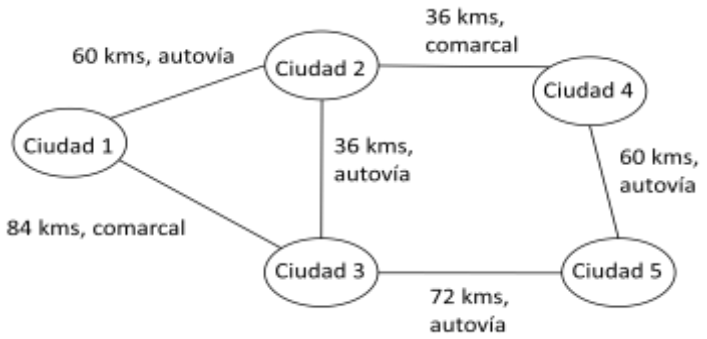
Inteligencia Artificial
Hoja 2: Búsqueda

Publicación: 2013/01/19
Entrega: 2013/02/26/10:00
Resolución en clase: 2013/02/26



Inteligencia Artificial Hoja 1: Lógica	Publicación: 2013/01/19 Entrega: 2013/02/26/10:00 Resolución en clase: 2013/02/26

2. Consideremos el siguiente mapa de ciudades, donde cada nodo representa una ciudad, y cada arco una carretera que conecta directamente dos ciudades (asociados a cada arco tenemos la longitud de dicha carretera y el tipo de carretera):



Hay dos tipos de carreteras: comarcales y autovías. En cada una de estas se estima que la velocidad promedio va a ser diferente (60 km/h y 100 km/h respectivamente). Nuestro objetivo es llegar desde la ciudad 1 a la ciudad 5 por el camino por el que tardemos menos tiempo en nuestro coche.

Como información adicional disponemos de la distancia en línea recta de cada una de las ciudades a la objetivo:

Ciudad	Distancia a ciudad 5 (kms)
1	120
2	90
3	60
4	42
5	0

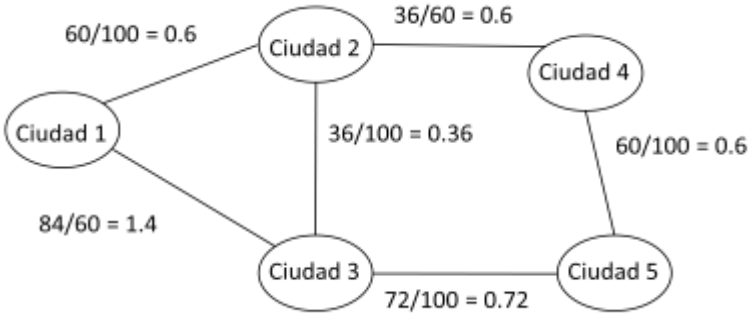
Responde a las siguientes cuestiones:

- Define una heurística admisible para este problema y muestra los valores de esta h
- Determina, para cada estado, el coste para ir de dicho estado a cada uno de sus posibles sucesores.
- ¿A qué solución llegaría búsqueda-en-árbol (es decir, sin eliminar estados repetidos) usando A* y esa heurística? Muestra paso a paso qué nodos expandiría el algoritmo, cuáles genera, y en qué orden. ¿Qué solución encuentra? ¿es la óptima? ¿cuántas horas tardaría nuestro coche en llegar a la ciudad objetivo por esta trayectoria?

h(n) = (distancia a 5) / máximo(60, 100) = (distancia a 5)/100

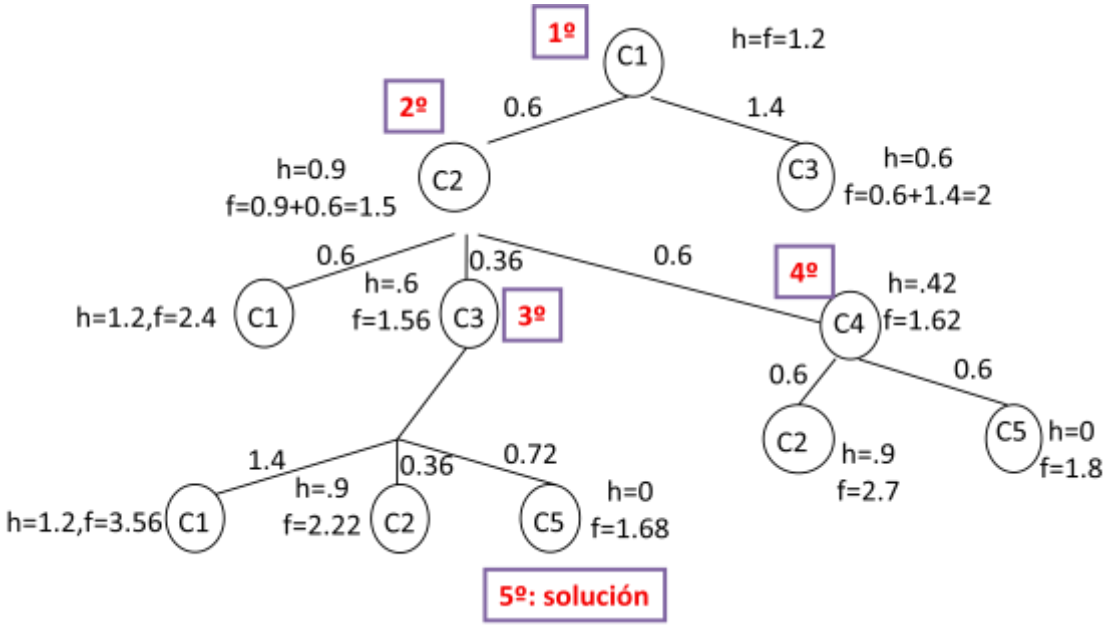
Ciudad	Distancia a 5 en línea recta (kms)	h
1	120	120/100 = 1.2
2	90	90/100 = 0.9
3	60	60/100 = 0.6
4	42	42/100 = 0.42
5	0	0

Costes con los que debemos trabajar (en horas):



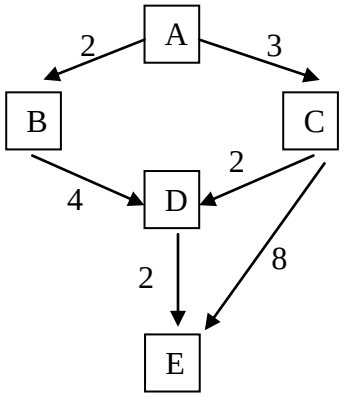
Resultado de aplicar A*:

Inteligencia Artificial Hoja 1: Lógica	Publicación: 2013/01/19 Entrega: 2013/02/26/10:00 Resolución en clase: 2013/02/26



Inteligencia Artificial Hoja 1: Lógica	Publicación: 2013/01/19 Entrega: 2013/02/26/10:00 Resolución en clase: 2013/02/26

3. Proporciona un ejemplo de una heurística que sea admisible pero no monótona para la búsqueda en el siguiente grafo [estado inicial: nodo A; estado final: nodo E].



SOLUCIÓN:
Condición de monotonicidad ∀n,n'(vecinos) $h(n) \leq coste(n \rightarrow n') + h(n')$
Una posible solución es

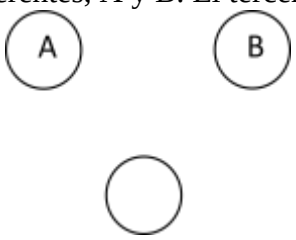
	h*(n)	h(n)
A	7	7
B	6	0
C	4	4
D	2	2
E	0	0

$\forall n\ h(n) \leq h^*(n)$ [admisible]

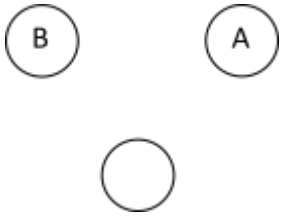
$h(B) + \Gamma(A \rightarrow B) < h(A)$ [no monótona]

Inteligencia Artificial Hoja 1: Lógica	Publicación: 2013/01/19 Entrega: 2013/02/26/10:00 Resolución en clase: 2013/02/26

4. En una fábrica disponemos de tres contenedores. Dos de ellos están inicialmente llenos de dos substancias diferentes, A y B. El tercero está vacío:



El objetivo es el siguiente estado



No se pueden mezclar las sustancias A y B. Las tres acciones posibles son:

- Mover en un paso el contenido de los tres contenedores **en el sentido de las agujas del reloj** (coste = 3), por ejemplo:

- Mover A al contenedor vacío (coste = 1)
- Mover B al contenedor vacío (coste = 1)

  El orden en el que se aplican estos operadores para generar sucesores es este mismo (1º, mover en el sentido de las agujas del reloj; 2º, mover A al contenedor vacío; 3º, mover B al contenedor vacío).

  En caso de que haya varios estados con la misma valoración, se explorará antes el estado que se haya generado antes.

4.1 Define una heurística para este problema utilizando el método de relajación, y que no sea trivial (h=0 en todos los estados).

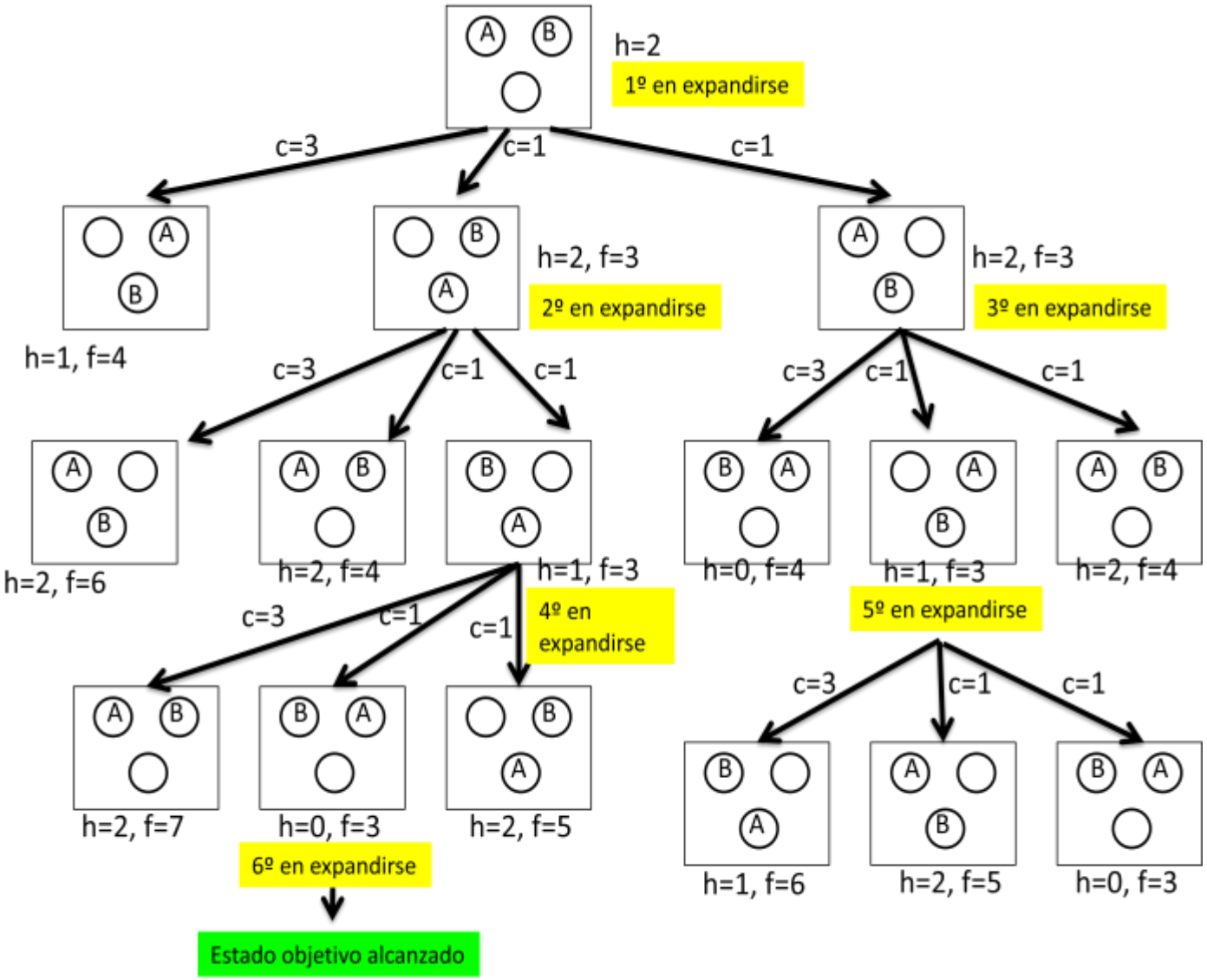
4.2 ¿Es dicha heurística admisible? ¿Por qué?
4.3 ¿Cuál es la solución encontrada por A* con búsqueda en árbol (sin eliminar estados repetidos)? ¿Es la solución óptima?

SOLUCIÓN:

Solución a la pregunta de búsqueda:

Heurística: relajando el problema (asumiendo que se pueden mezclar y volver a separar completamente las sustancias) el número de pasos hacia la solución no puede ser mayor que colocar cada solución directamente en su situación objetivo. Como cada paso tiene como poco un coste de 1, esta heurística es admisible (no puede haber caminos a la solución con un coste menor): la heurística la definimos entonces como el número de contenedores que no tienen el contenido objetivo (sin tener en cuenta el contenedor vacío).

Inteligencia Artificial Hoja 1: Lógica	Publicación: 2013/01/19 Entrega: 2013/02/26/10:00 Resolución en clase: 2013/02/26



5. Consideremos el problema de búsqueda en un espacio con estados A, B, C, D, E. Las acciones posibles en este espacio son:

origen	destino	Coste
A	B	3
A	C	2
B	C	5
B	D	3
C	D	5
C	B	1
C	E	11
D	E	5
E	B	8

Consideremos las heurísticas

nodo	h_1 (nodo)
A	8
B	6
C	6
D	4
E	0

nodo	h_2 (nodo)
A	7
B	8
C	5
D	5
E	0

Cuestiones: (justificar las respuestas

e) ¿El algoritmo A* es óptimo con h_1?
f) ¿El algoritmo A* es óptimo con h_2?
g) ¿Son admisibles las siguientes heurísticas? (para cada nodo n se toma como valor de la heurística)
   ix. $h_1(n)$
   x. $h_2(n)$
   xi. $\min(h_1(n), h_2(n))$
   xii. $h_1(n) + h_2(n)$
   xiii. $(h_1(n) + h_2(n))/2$
   xiv. $\max(h_1(n), h_2(n))$
   xv. $h_1(n) \cdot h_2(n)$
   xvi. $(h_1(n) \cdot h_2(n))^{1/2}$
h) ¿De entre las heurísticas admisibles, hay alguna dominante?

Respuesta:

Una heurística h es monótona si cumple, para todos los pares de nodos n, n', donde n' es sucesor de n,

$$h(n) \leq h(n') + \text{coste}(n \rightarrow n')$$

Vamos a demostrar que $h_1(n)$ es monótona

origen	destino	Coste	h(n)	h(n') +	h(n)≤h(n') +

Inteligencia Artificial
Hoja 1: Lógica

Publicación: 2013/01/19
Entrega: 2013/02/26/10:00
Resolución en clase: 2013/02/26

nodo	h_1 (nodo)
A	8
B	6
C	6
D	4
E	0

n	n'	coste(n→n')		coste(n→n')	coste(n→n')
A (8)	B (6)	3	8	3 + 6 = 9	True
A (8)	C (6)	2	8	2 + 6 = 8	True
B (6)	C (6)	5	6	5 + 6 = 11	True
B (6)	D (4)	3	6	3 + 4 = 7	True
C (6)	D (4)	5	6	5 + 4 = 9	True
C (6)	B (6)	1	6	1 + 6 = 7	True
C (6)	E (0)	11	6	11 + 0 = 11	True
D (4)	E (0)	5	4	5 + 0 = 5	True
E (0)	B (6)	8	0	8 + 6 = 14	True

Vamos a demostrar que $h_2(n)$ es monótona

nodo	h_1 (nodo)
A	7
B	8
C	5
D	5
E	0

origen n	destino n'	Coste coste(n→n')	h(n)	h(n') + coste(n→n')	h(n)≤h(n') + coste(n→n')
A (7)	B (8)	3	7	3 + 8 = 11	True
A (7)	C (5)	2	7	2 + 5 = 7	True
B (8)	C (5)	5	8	5 + 5 = 10	True
B (8)	D (5)	3	8	3 + 5 = 8	True
C (5)	D (5)	5	5	5 + 5 = 10	True
C (5)	B (8)	1	5	1 + 8 = 9	True
C (5)	E (0)	11	5	11 + 0 = 11	True
D (5)	E (0)	5	5	5 + 0 = 5	True
E (0)	B (8)	8	0	8 + 8 = 16	True

a) y b) A* es óptimo con heurísticas monótonas, luego A* es óptimo con h_1 y h_2

c) h_1 y h_2 son monótonas y por lo tanto admisibles, por lo que utilizando los resultados del apartado anterior

i. $h_1(n)$ es admisible
ii. $h_2(n)$ es admisible
iii. $\min(h_1(n), h_2(n))$ es admisible
iv. $h_1(n) + h_2(n)$ no es admisible
v. $(h_1(n) + h_2(n))/2$ es admisible
vi. $\max(h_1(n), h_2(n))$ es admisible
vii. $h_1(n) \cdot h_2(n)$ no es admisible
viii. $(h_1(n) \cdot h_2(n))^{1/2}$ es admisible.

c) Para todos los nodos n

$$\min(h_1(n), h_2(n)) \leq (h_1(n) \cdot h_2(n))^{1/2} \leq (h_1(n) + h_2(n))/2 \leq \max(h_1(n), h_2(n))$$

Por lo que

(vi) domina a (v) domina a (viii) domina a (iii)