

WUOLAH



vrnk98

www.wuolah.com/student/vrnk98



12559

Relacion2 .pdf

Relacion 2 IA ejercicios Resueltos



2º Inteligencia Artificial



Grado en Ingeniería Informática



Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
UGR - Universidad de Granada

1. Tenemos un tablero de 3×3 casillas como el siguiente:

Deseamos intercambiar los caballos negros con los blancos.

(a) Define que elementos forman:

- el estado inicial
- el estado final
- los operadores

(b) ¿Importa el camino o sólo el estado final?

(c) ¿La solución ha de ser óptima?

a) Estado inicial

1-Representación

matriz $M(i,j) \in \{L,N,B\}$

2-Estado inicial

NLN

LLL

BLB

3-Estado final

BLB

LLL

NLN

4-Acciones: total 8 acciones

IZQ-ARR $\rightarrow \{M(i-1,j-2)\}$

ARR-IZQ $\rightarrow \{M(i-2,j-1)\}$

DCHA-ABA $\rightarrow \{M(i+1,j+2)\}$

ABA-DCHA $\rightarrow \{M(i+2,j+1)\}$

5-Modelo de búsqueda

Búsqueda en profundidad(BP){

PLAN=BP(Estado s,Arbol A)

Si S es final{

Plan=Construir_plan(s,A)

Devolver Plan

}

En otro caso{

Para cada posible acción Act{

Estado nuevo=Generar_Estado(s,Act)

Si(A no contiene nuevo){

Insertar nuevo Como hijo de s en A;

Plan=BP(nuevo,A);

Si plan!=0{

Devolver plan

}

}

}

}

Devolver plan vacio

}

b) -----

c) No tiene por qué

CONGRESO INTERNACIONAL DESAFÍOS A LA SEGURIDAD GLOBAL

INTELIGENCIA, TERRORISMO Y
AMENAZAS HÍBRIDAS

17 - 18 JUNIO

LUGAR

Centro Superior de Estudios de la
Defensa Nacional CESEDEN

Pº de la Castellana, 61 Madrid

MÁS INFORMACIÓN

eventos@iniseg.es

912 141 926

www.iniseg.es

Organiza:



INISEG

Instituto Internacional de
Estudios en Seguridad Global



INSTITUCIONES COLABORADORAS



OTROS COLABORADORES



Sigue las novedades del congreso

#CIS_INISEG

2. El grafo que se muestra a continuación determina un problema de búsqueda. Cada nodo representa un estado, mientras que los arcos modelan la aplicación de los operadores. Si A es el estado inicial y K y E son los estados meta:

- (a) Desarrolle el árbol de búsqueda en anchura
- (b) ¿Cuál de los nodos meta se expande primero?
- (c) La búsqueda en profundidad se diferencia de la búsqueda en anchura en que al expandir un nodo, los nodos hijo se insertan al inicio de la lista. Resuelva el problema usando este algoritmo.

- a) ADFGHCEBKZW
- b) Se expande primero el nodo E
- c) ADHBHDCCKBKCDADFCKBKCFEZEWEFAGA

3. Liste el orden en el que son visitados los nodos del árbol para cada una de las siguientes estrategias de búsqueda (1) . ¿Cuál es el número máximo de nodos que se ha requerido almacenar simultáneamente en memoria en cada caso?

- (a) Búsqueda primero en anchura.
- (b) Búsqueda en profundidad.
- (c) Búsqueda retroactiva.
- (d) Descenso iterativo.

- | | |
|------------------------------|-------------------------------|
| a) ABCDEFGHIJKLM | 13 nodos |
| b) ABEBFJFKFBACGLGMGCADHDIDA | 25 nodos |
| c) ABEFJKCGLMDHI | 13 nodos |
| d) ABCD | Depth bound = 1 4 nodos |
| ABEFCGDHI | Depth bound = 2 9 nodos |
| ABEFJKCGLMDHI | Depth bound = 3 13 nodos |

4. Dar el orden de búsqueda para los nodos mostrados en la siguiente figura para:

- (a) Búsqueda en anchura.
- (b) Búsqueda retroactiva con profundidad limitada ($d=2$).
- (c) Descenso iterativo (profundidad inicial = 1).

- a) ABCDEFGHIJKLMN
- b) ABEBFBACGCHCADIDJDA
- c) ABCD Depth bound = 1
- ABEFCGHDIJ Depth bound = 2
- ABEFKLCGHDIMNJ Depth bound = 3

5. Ejecutar el algoritmo de búsqueda retroactiva sobre el grafo de la figura:

ABEKSKELTLEBFLTLFMFBACGNGCHOHPUPHCADIPUPIQIDJRJDA

6. En la orilla de un río hay 3 misioneros y 3 caníbales. Todos ellos pretenden cruzar al otro lado. La barca que se utiliza para cruzarlo solo tiene capacidad para dos personas, con lo que alguien ha de estar volviendo siempre a la orilla inicial mientras quede gente sin cruzar. Además, si en alguna ocasión y en cualquiera de las orillas se encuentran un número mayor de caníbales que de misioneros, los primeros se comerían a los segundos.

- (a) ¿Cómo representarías los estados?
- (b) ¿Cuáles serían los operadores?
- (c) Dibujar el espacio de estados

1-Representación

(M,C) O1 (M,C) B

2-Estado Inicial

$(3,3)$ $(0,0)$

3-Estado Final

$(0,0)$ $(0,0)$

4-Acciones

Montar desde orilla1 en barca $M(x,y,2) \rightarrow S'$ Misioneros, caníbales, estado actual

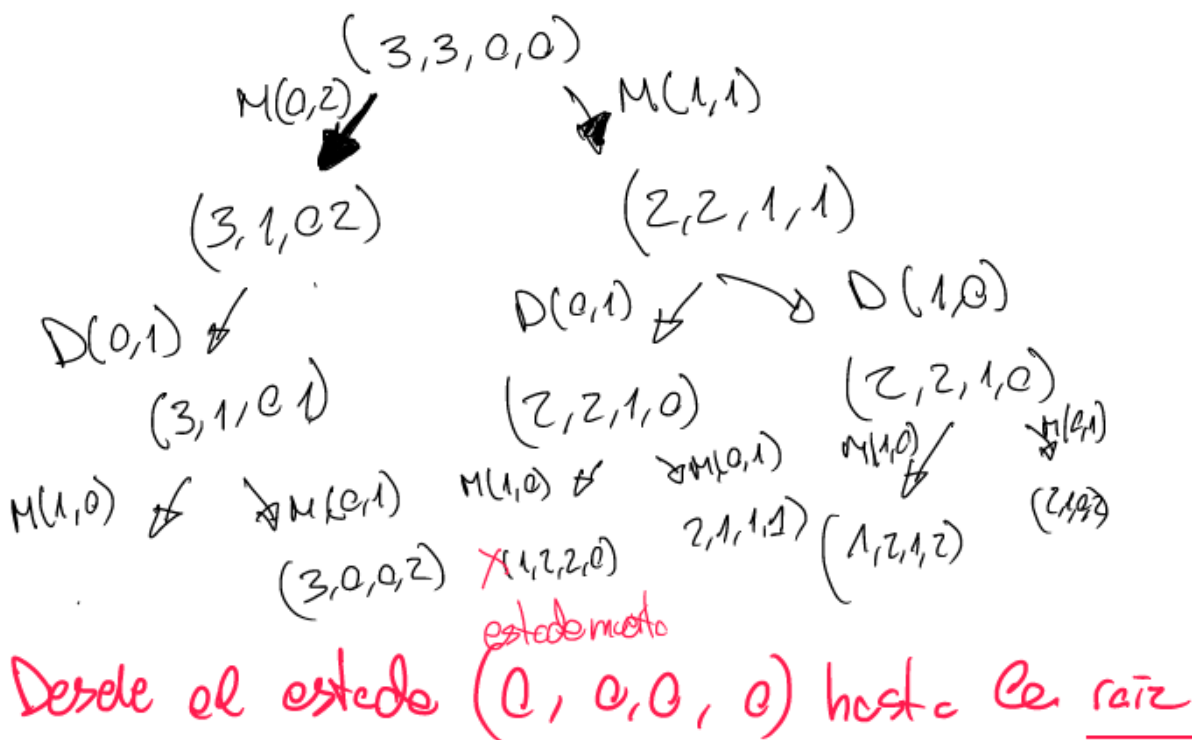
$S' \rightarrow (M1-x, C1-y), (M2+x, C2+y)$

Descargar en orilla2 desde barca

$D(x,y,2) \rightarrow S'=(M1,C1), (M2-x, C2-y)$

Problema de búsqueda explícita sobre grafos que se conoce

5-Búsqueda Grafo Explicito



7. En una mesa se encuentran dos jarras, una con capacidad para 3 litros (llamada Tres), y la otra con capacidad para 4 litros (llamada Cuatro). Inicialmente, Tres y Cuatro están vacías. Cualquiera de ellas puede llenarse con agua del grifo G.

Asimismo, el contenido de las jarras se puede vaciar en una pila P. También es posible verter el agua de una jarra en la otra. No se dispone de dispositivos de medición adicionales. Se trata de encontrar una secuencia de operadores que deje exactamente dos litros de agua en Cuatro.

(a) Representar este problema como un problema de búsqueda

(b) Definir un estado inicial, el conjunto de estados meta, los operadores, así como el coste de cada operador.

(c) Encontrar una solución al problema utilizando un algoritmo de búsqueda.

1-Representación

(T,C) C= cuatro(puede tener valor 1,2,3,4), T= tres(puede tener valor 0,1,2,3)

2-Estado inicial

$(T,C) = 0$

3-Estado final

$$(T,C) = (x,2)$$

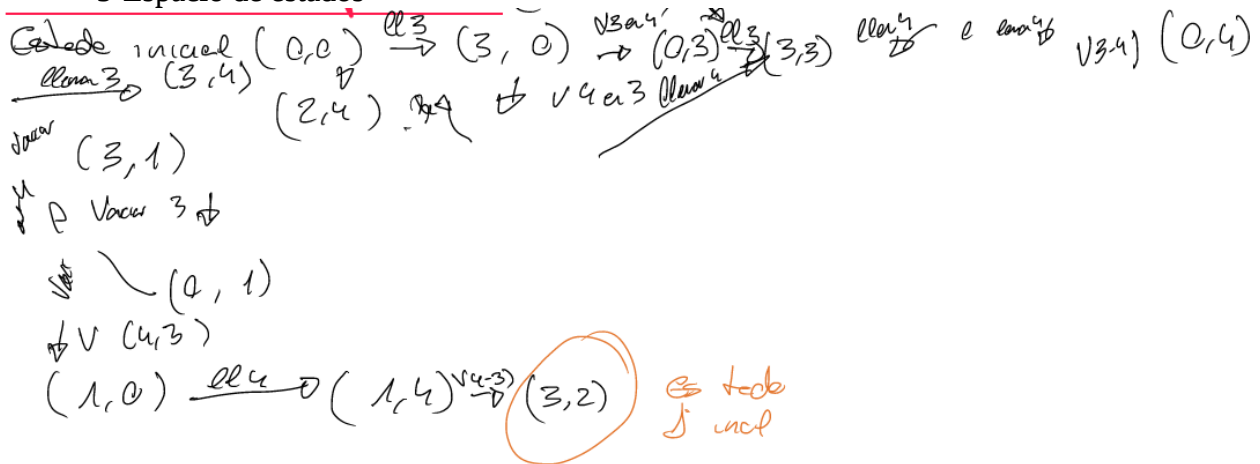
4-Acciones

llenar(G) sobre un estado $(x,y) \Rightarrow (3,y)$ si $G=T$
 $\Rightarrow (x,4)$ si $G=C$

vaciar(G) sobre el estado $(x,y) \Rightarrow (0,y)$ si $G=T$
 $\Rightarrow (x,0)$ si $G=C$

volcar(A,B) sobre el estado $(x,y) \Rightarrow (1,3)$
 $\Rightarrow (0,4)$

5-Espacio de estados



8. “Las torres de Hanoi” es un juego matemático ideado en el siglo XVIII. Este juego consiste en pasar 64 discos de diámetro decreciente, de un poste a otro poste, utilizando un tercer poste auxiliar para los pasos intermedios.

Cada vez solo se puede mover un disco, los discos siempre deben estar en algún poste y no se puede colocar un disco sobre otro de menor tamaño.

(a) ¿Cómo representarías los estados?

(b) ¿Cuáles serían los operadores?

1. Representación

3 vectores: $P1[]$, $P2[]$, $P3[]$

2. Estado inicial

$P1[64, 63, \dots, 3, 2, 1]$

$P2[]$

$P3[]$

3. Estado final

$P1[]$

$P2[]$

$P3[64, 63, \dots, 3, 2, 1]$

4. Acciones

Se puede mover si el poste está vacío

Si $x < y$, siendo y el disco de encima $\rightarrow x.push_back$:

Mover x de $P1$ a $P2$

Mover x de $P2$ a $P3$

Mover x de $P1$ a $P3$

Mover x de $P2$ a $P1$

Mover x de $P3$ a $P2$

Mover x de $P3$ a $P1$

9. Un acertijo consiste en dados 4 números y un resultado, determinar las operaciones de suma o resta que hay que realizar sobre los números para obtener ese resultado.

Por ejemplo:

- Números: 1, 4, 3, 2
- Resultado: 0
- Solución: $4 - 3 - 2 + 1$

Suponiendo que resolvemos el acertijo como un problema de búsqueda, responde las siguientes cuestiones:

- Propón una representación de los estados y explica cómo se generarían los estados sucesores.
- ¿Cuál sería el tamaño del espacio de estados?
- ¿Qué tipo de algoritmo de búsqueda no informada sería mejor utilizar y por qué?

1. Representación

Vector de 4 números

2. Estado inicial

$v[0,0,0,0]$

3. Estado final

$s \rightarrow$ resultado de las operaciones

4. Acciones

Sumar (si es un hijo derecha)

Restar (si es un hijo izquierda)

El tamaño del espacio de estado sería 16.

Usaría búsqueda en profundidad

10. Aplica la búsqueda de coste uniforme para encontrar la ruta más corta de Pitesti a Fagaras.
 (a) Desarrolla el árbol de búsqueda generado por el algoritmo, asumiendo que se evitan ciclos simples.
 (b) Indica el valor g de cada nodo, así como el orden en que se expanden.

10. ABIERTO

~~(P, P, 0)~~
~~(E, P, 99)~~
~~(B, P, 101)~~
~~(C, P, 138)~~
 IGNORE \rightarrow (C, P, 243)
~~(S, E, 172)~~
 (F, B, 312) \rightarrow (F, S, 276)
~~(U, B, 186)~~
~~(G, B, 191)~~
~~(D, C, 258)~~
 (A, S, 3187)
 (O, S, 328)
 (U, V, 348)
 (H, U, 284)
 (M, D, 333)
 {

CERRADO

(P, P, 0)
 (E, P, 99)
 (B, P, 101)
 (C, P, 138)
 (S, E, 172)
 (U, B, 186)
 (G, B, 191)
 (D, C, 258)
 (F, S, 276)

* COSTE: EL COSTE = G
 ACUMULADO.

13. El algoritmo A* no termina mientras no se selecciona un nodo objetivo para su expansión. Sin embargo, es posible encontrar un camino al objetivo mucho antes.

- (a) ¿Por qué no se termina en el momento en que se encuentra un nodo objetivo?
 (b) Ilustre la respuesta anterior con un ejemplo

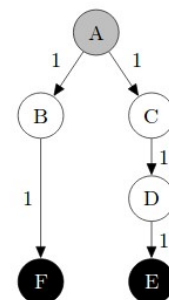
Porque el algoritmo A* termina cuando la lista de abiertos se vacía.

ABIERTOS

(A,B)
 (A,C)
 (B,F)
 (C,D)
 (D,E)

CERRADOS

(A,B)
 (B,F) \rightarrow Objetivo
 (A,C)
 (C,D)
 (D,E) \rightarrow Objetivo



14. 4. El grafo que se muestra en la figura describe un problema de búsqueda. Suponga que A es el estado inicial y que F y E son estados meta. Los arcos están etiquetados con el coste real de los operadores y h es una función heurística cuyos valores son:

	A	B	C	D	E	F
H	8	6	6	5	0	0

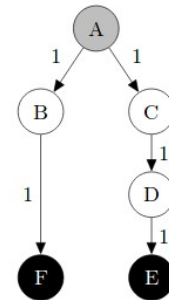
- Desarrolle el árbol de búsqueda que genera el algoritmo A*.
- Indique el orden en que se expanden los nodos.
- ¿Cuál de los nodos meta se encuentra primero?
- ¿La función heurística es admisible? Argumente su respuesta.

ABIERTOS

A(\emptyset ,0,8,8)
 B(A,1,6,7)
 C(A,1,6,7)
 F(B,2,0,2)
 D(C,2,5,7)
 E(D,3,0,3)

CERRADOS

A(8)
 B(7)
 C(7)
 F(2)
 D(7)
 E(3)



Se encuentra primero el nodo F

No, porque no se cumple: $h(n) \leq h^*(n)$

15. Considere el problema de búsqueda del ejercicio precedente.

- Asigne los valores del coste real de los operadores y de la función heurística h, de modo que ésta resulte ser admisible.
- Desarrolle el árbol de búsqueda que genera el algoritmo A*.
- ¿Se ha obtenido la solución óptima?

Coste operadores	B	C	D	E	F
A	9	9			
B					7
C			7		
D				6	

Abiertos

A(\emptyset ,0,8,8)
 B(A,9,6,15)
 C(A,9,6,15)
 F(B,7,0,7)

Cerrados

A(8)
 B(15)
 F(7)

La solución es óptima $h(n) \leq h^*(n)$ ya que $5 \leq 6$

16. Tenemos cinco monedas dispuestas de la siguiente forma:

A R A R A

El anverso de la moneda está representado por A y el reverso por R. Se considera un movimiento (de coste 1) el dar la vuelta a dos monedas contiguas. Deseamos obtener la situación final siguiente:

R R R A R

Dada la función heurística $h(n)$ = número de monedas mal colocadas.

- Resolver el problema aplicando A*. Indica claramente el orden de expansión de los nodos, el tratamiento de nodos duplicados, los valores de las funciones, el camino obtenido y su coste.
- ¿La heurística usada es admisible? ¿Por qué?

1-Representación

vector de 5 caracteres

2-Estado inicial

monedas={A,R,A,R,A}

3-Estado final

monedas={R,R,R,A,R}

4-Acciones total

Voltear

5-A*

$g(S)$ =profundidad(S)

$h(S)$ =nº monedas mal colocados

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

17. Dado el siguiente mapa de carreteras en el que los caminos entre cada dos ciudades están etiquetados con sus distancias en kilómetros:

El objetivo es encontrar el camino más corto entre Palencia y Barcelona. Para ello:

- Aplica el algoritmo A* utilizando como estimación del coste del camino entre la ciudad actual y el objetivo el valor 0 para todos los nodos.
- Aplica el algoritmo A* utilizando ahora como estimación del coste del camino entre la ciudad actual y el objetivo las distancias siguientes:

	Bilbao	Cáceres	Madrid	Palencia	Santander	Valencia	Zaragoza
Barcelona	502	850	550	580	605	303	275

Abiertos

Palencia(\emptyset ,0,580,580)

Cáceres(Palencia, 368, 850, 1218)

Madrid(Palencia, 239, 550, 789)

Santander(Palencia, 203, 605, 808)

Zaragoza(Madrid, 561, 275, 836)

Valencia(Madrid, 589, 303, 892)

Bilbao(Santander, 314, 502, 816)

Barcelona(Zaragoza, 860, 0, 860)

Cerrados

Palencia

Madrid

Santander

Bilbao

Zaragoza

Barcelona

18. Considere el 8-puzzle cuyo estado inicial y estado meta se muestran en la siguiente figura:

1	2	3
6	4	
8	7	5

Estado Inicial

1	2	3
8		4
7	6	5

Estado Meta

Desarrolle el árbol de búsqueda que expande el algoritmo A* utilizando las siguientes heurísticas. Evite ciclos generales, indique el orden de expansión de los estados y muestre en cada paso los valores de f, g y h. Suponga que el coste de cada operador es 1.

(a) Heurística $h_1(n)$ = número de piezas mal colocadas.

(b) Heurística $h_2(n)$ = suma de distancia Manhattan. La distancia Manhattan de una pieza es la suma de las distancias vertical y horizontal a su posición final.

(c) ¿Cuál de las heurísticas expande menos nodos y por qué?

(d) ¿Puede sacar una conclusión general con respecto a la calidad de las funciones heurísticas?

a)

Abiertos

A(0, 4, 4)

C(1, 3, 4)

D(1, 5, 6)

B(1, 5, 6)

E(2, 3, 5)

F(2, 3, 5)

G(2, 4, 6)

H(3, 2, 5)

I(3, 4, 7)

J(3, 3, 6)

K(3, 4, 7)

L(4, 1, 5)

M(5, 0, 5)

Cerrados

A(4)

C(4)

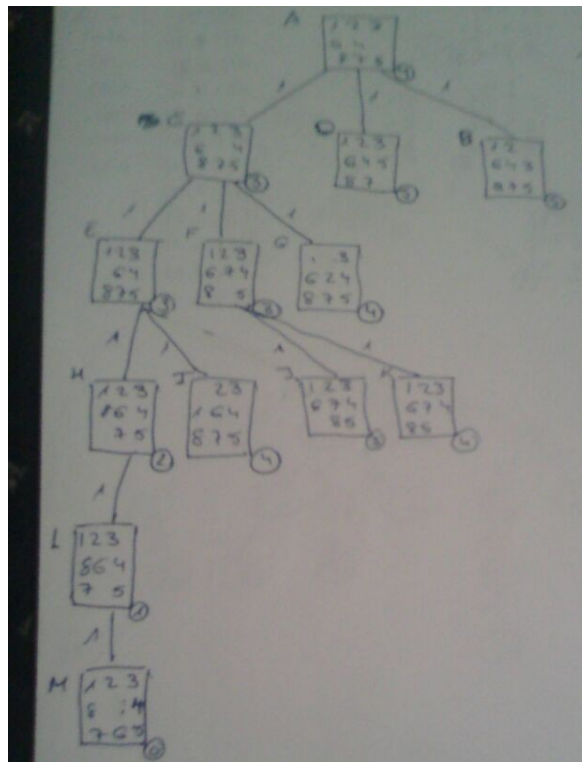
E(5)

F(5)

H(5)

L(5)

M(5)



b)

Abiertos

A(\emptyset , 5, 5)

B(1, 6, 7)

C(1, 4, 5)

D(1, 6, 7)

E(2, 3, 5)

F(2, 5, 7)

G(2, 5, 7)

H(3, 2, 5)

I(3, 4, 7)

J(4, 1, 5)

K(5, 0, 5)

Cerrados

A(5)

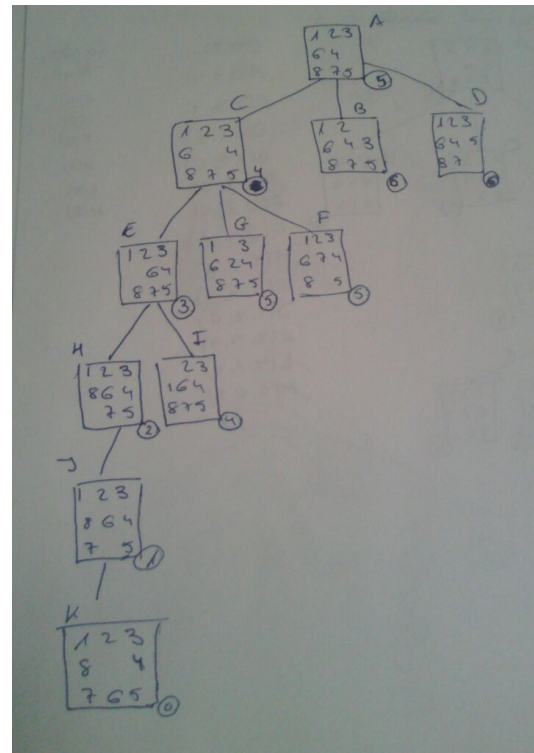
C(5)

E(5)

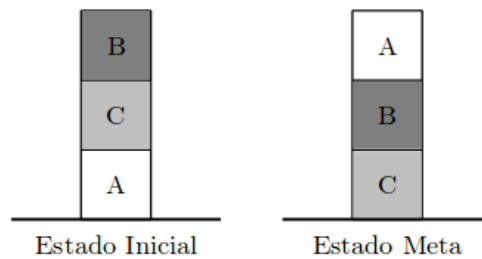
H(5)

J(5)

K(5)



19. Considere el problema de los bloques cuyo estado inicial y estado meta se muestran en la siguiente figura:



Desarrolle el árbol de búsqueda que expande el algoritmo A* utilizando la siguiente heurística:
 $h(n)$ = número de bloques mal colocados

Filtre los ciclos simples, indique el orden de expansión de los estados y muestre en cada paso los valores de f, g y h. Suponga que el coste de cada operador es 1.

Abiertos

$\alpha(0, 3, 3)$

$\beta(1, 3, 4)$

$\gamma(1, 1, 2)$

$\delta(2, 0, 2)$

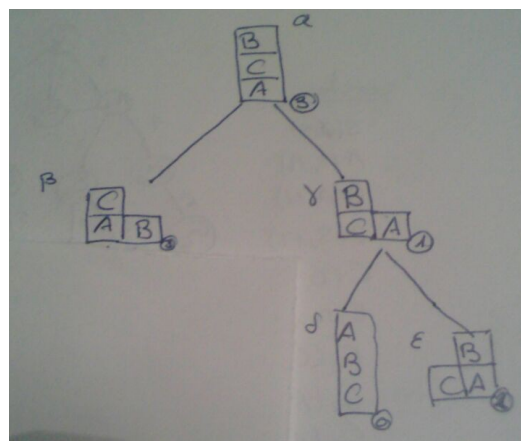
$\epsilon(2, 2, 4)$

Cerrados

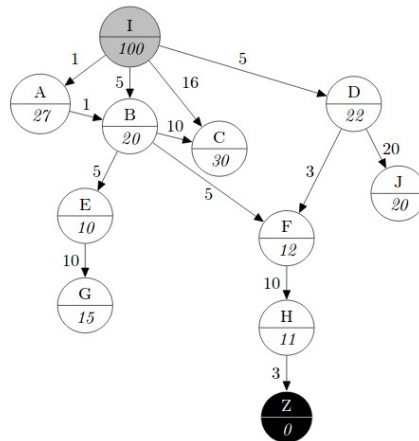
$\alpha(3)$

$\gamma(2)$

$\delta(2)$



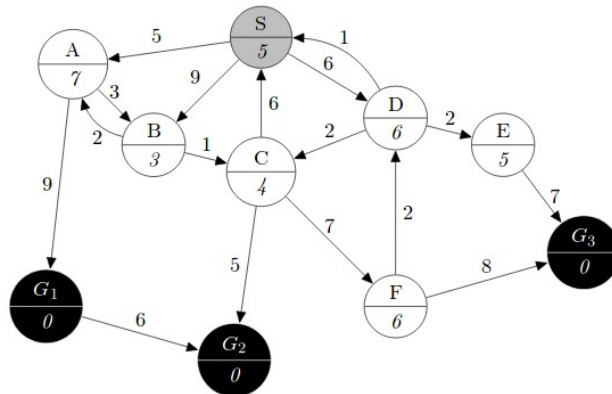
20. Dado el siguiente grafo representando un espacio de estados, utilizar el algoritmo A* para obtener la solución al problema. En dicho grafo considerar como función h la que se indica dentro de cada nodo, y para el cálculo de la función g considerar el coste que aparece en el arco. El nodo inicial es I y hay un solo nodo meta Z.



- (a) Mostrar el gráfico final con la solución obtenida, los valores g y h finales para cada nodo, junto con los enlaces a los hijos y enlace al mejor padre.
 (b) ¿Es la función heurística utilizada en el problema admisible?

Abiertos	Cerrados
I(0, 100, 100)	I(100)
A(1, 27, 28)	B(22)
B(2, 20, 22)	E(17)
C(12, 30, 42)	D(27)
D(5, 22, 27)	F(19)
E(7, 10, 17)	A(28)
F(7, 12, 19)	H(28)
G(17, 15, 32)	Z(21)
H(17, 11, 28)	
J(20, 20, 40)	
Z(21, 0, 21)	

21. Supongamos que el siguiente grafo representa un espacio de estados de un problema real. El número dentro de cada nodo representa el valor de la función h del nodo y el número sobre el arco el coste del operador.



S es el nodo de inicio y G1, G2 y G3 son nodos que satisfacen la condición objetivo. En todos los casos ABIERTOS representará el conjunto de nodos que puede ser explorados en la siguiente iteración del algoritmo de búsqueda. Utilizar distintos algoritmos de búsqueda para resolver el problema, para cada algoritmo obtener los siguientes datos:

- El conjunto de nodos expandidos en el orden que salen de ABIERTOS.
- El objetivo alcanzado.
- El camino solución encontrado.

Utilizar los siguientes algoritmos:

- Búsqueda en anchura.
- Profundizaje iterativo
- Método de escalada simple (usando solo la función h).
- Método de escalada por la máxima pendiente (usando solo la función h).
- Algoritmo A* (usando la función $f=g+h$).

a)
Abiertos {S, A, B, D, G1}
Cerrados {S, A, B, D, G1}

b)
Abiertos {S, A, G1}
Cerrados {S, A, G1}

c)
Abiertos {S, B, A, C}
Cerrados {S, B}

d)
Abiertos {S, A, B, D, C}
Cerrados {S, B}

e)

Abiertos

S(\emptyset , 5)

A(S, 12)

B(S, 12)

D(S, 12)

B(A, 11)

G1(A, 14)

C(B, 13)

C(D, 12)

E(D, 13)

G2(C, 13)

F(C, 21)

G3(E, 15)

Cerrados

S(\emptyset , 5)

A(S, 12)

B(A, 11)

D(S, 12)

C(D, 12)

E(D, 13)

G2(C, 13)

G1(A, 14)

G3(E, 15)

F(C, 21)

23. Aplicar el algoritmo A* para hallar el camino con menor distancia que una las ciudades 1 y 8. Las distancias por carretera entre las distintas ciudades vienen especificada por la siguiente tabla:

	1	2	3	4	5	6	7	8
1		200						
2	200		150	250	450			
3		150			400	225		
4		350			300			
5		450	400	300			250	
6			225				450	
7					250	450		125
8							125	

Además se dispone de la distancia aérea en línea recta que existe entre todas las ciudades con la ciudad de destino:

1	2	3	4	5	6	7
800	650	500	650	325	375	125

Abiertos

A(\emptyset , 0, 800, 800)

B(A, 200, 600, 800)

C(B, 350, 500, 850)

D(B, 450, 650, 1100)

E(B, 650, 325, 975)

F(C, 575, 375, 950)

G(F, 1025, 125, 1150)

G(E, 900, 125, 1025)

H(G, 1025, 0, 1025)

Cerrados

A(800)

B(800)

C(850)

F(950)

E(975)

G(1025)

H(1025)