

# Intel·ligència Artificial

## Col·lecció de Problemes

Departament de Ciències de la Computació  
Grau en Enginyeria Informàtica  
Curs 2019/2020 1Q



**FIB**

Facultat d'Informàtica  
de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA



Copyleft © 2010-2020 UPC

UNIVERSITAT POLITÈCNICA DE CATALUNYA

*Edició Setembre 2019*

Esta obra está bajo una licencia  
Reconocimiento-NoComercial-CompartirIgual de Creative Commons.

Para ver una copia de esta licencia, visite  
<http://creativecommons.org/licenses/by-nc-sa/2.5/es/>  
o envíe una carta a

Creative Commons,  
559 Nathan Abbott Way, Stanford,  
California 94305,  
USA.

En l'elaboració de la col·lecció de problemes d'IA han participat els professors:

M<sup>a</sup> Teresa Abad Soriano  
Javier Béjar Alonso  
Núria Castell Ariño  
Miquel Sànchez i Marré  
Jordi Turmo Borrás  
Javier Vázquez Salceda

Responsable de la publicació: Javier Béjar (bejar@cs.upc.edu)

# Índex

I	<b>Cerca</b>	
1	Representació de Problemes de Cerca .....	1
2	Cerca Heurística .....	5
3	Cerca Local .....	15
4	Jocs .....	33
5	Satisfacció de restriccions .....	43
6	Anàlisi de mètodes de cerca .....	53
II	<b>Sistemes Basats en el Coneixement</b>	
7	Sistemes de producció .....	67
8	Enginyeria del Coneixement .....	77
9	Xarxes Bayesianes .....	107
III	<b>Planificació</b>	
10	Planificació .....	115





# Cerca

1	Representació de Problemes de Cerca ...	1
2	Cerca Heurística .....	5
3	Cerca Local .....	15
4	Jocs .....	33
5	Satisfacció de restriccions .....	43
6	Anàlisi de mètodes de cerca .....	53





# 1. Representació de Problemes de Cerca

1. Tenemos un tablero de 3x3 casillas como el de la figura

N		N
B		B

En cada esquina tenemos un caballo de ajedrez, dos caballos negros y dos blancos. Deseamos intercambiar los caballos negros con los blancos.

- (a) Define que elementos forman el estado, el estado inicial y cual es estado final o que propiedades ha de cumplir. Estima el tamaño del espacio de estados.
  - (b) Define las características de los operadores para realizar la búsqueda (condiciones de aplicabilidad y función de transformación), evalúa cual sería el factor de ramificación (aproximadamente).
  - (c) ¿Importa el camino o sólo el estado final? ¿la solución ha de ser óptima?
2. Dispones de dos jarras de agua, una de 4 litros y otra de 3 litros. Tiene un grifo que te permite llenar totalmente las jarras de agua, necesitas obtener exactamente 2 litros en la jarra de cuatro litros
    - (a) Define que elementos forman el estado, el estado inicial y cual es estado final o que propiedades ha de cumplir. Estima el tamaño del espacio de estados.
    - (b) Define las características de los operadores para realizar la búsqueda (condiciones de aplicabilidad y función de transformación), evalúa cual sería el factor de ramificación (aproximadamente).
    - (c) ¿Importa el camino o sólo el estado final? ¿la solución ha de ser óptima?
  3. Si recuerdas el procedimiento de validación por resolución en lógica de enunciados, puedes validar cualquier razonamiento a partir de la transformación de sus premisas y la negación de la conclusión a forma normal conjuntiva. El algoritmo mas sencillo (pero no muy eficiente) de validar un razonamiento es aplicar la regla de la resolución (desempolva tus apuntes de ILO) sistemáticamente entre las cláusulas hasta conseguir derivar una contradicción.
    - (a) Define que elementos forman el estado, el estado inicial y cual es estado final o que propiedades ha de cumplir. Estima el tamaño del espacio de estados.

- (b) Define las características de los operadores para realizar la búsqueda (condiciones de aplicabilidad y función de transformación), evalúa cual sería el factor de ramificación (aproximadamente).
  - (c) ¿Importa el camino o sólo el estado final? ¿la solución ha de ser óptima?
  - (d) Piensa en el mecanismo de resolución lineal y vuelve a responder a los apartados a y b
4. Si piensas en el juego del tetrís, este consiste en cubrir la máxima área de un rectángulo de  $N \times M$  sin dejar huecos (o dejando los mínimos posibles) utilizando un conjunto ordenado de piezas con todas las posibles formas construibles utilizando cuatro cuadrados (siete piezas distintas)
- (a) Define que elementos forman el estado, el estado inicial y cual es estado final o que propiedades ha de cumplir. Estima el tamaño del espacio de estados.
  - (b) Define las características de los operadores para realizar la búsqueda (condiciones de aplicabilidad y función de transformación), evalúa cual sería el factor de ramificación (aproximadamente).
  - (c) ¿Importa el camino o sólo el estado final? ¿la solución ha de ser óptima?
5. Continuando con el juego del tetrís este utiliza como piezas todas las formas posibles usando cuatro cuadrados de manera que cada cuadrado tenga al menos un lado contiguo a otro cuadrado. Supón que quieres calcular todas las formas posibles que se pueden construir con  $N$  cuadrados.
- (a) Plantéalo como un problema de búsqueda en espacio de estados definiendo los estados y los operadores necesarios para realizar la búsqueda (evidentemente no hay una única manera de plantear el problema, define las que se te ocurran y evalúa sus diferencias)
6. Dada una permutación de números de 1 a  $n$  queremos ordenarla utilizando el menor número de operaciones de inversión de un intervalo, donde una inversión de intervalo se define a partir de un par de posiciones  $(i, j)$  y su efecto es invertir el orden en el que están todos los números entre esas posiciones.
- (a) Define que elementos forman el estado, el estado inicial y cual es estado final o que propiedades ha de cumplir. Estima el tamaño del espacio de estados.
  - (b) Define las características de los operadores para realizar la búsqueda (condiciones de aplicabilidad y función de transformación), evalúa cual sería el factor de ramificación (aproximadamente).
7. Existen múltiples problemas clásicos sobre grafos que se pueden plantear como una búsqueda en espacio de estados. La gracia de estos problemas es que muchos problemas reales se pueden transformar a éstos. Intenta plantear los siguientes:

El viajante de comercio: Un viajante de comercio desea visitar un conjunto de ciudades partiendo de una dada y acabando en ésta, sin repetir ninguna y recorriendo el mínimo de distancia (se supone que dispone de un mapa que indica las conexiones entre ciudades y sus distancias)

El k-viajante de comercio: Ahora el viajante quiere obtener  $k$  caminos de mínima longitud que comiencen y terminen en una ciudad y que en los  $k$  caminos cada ciudad aparezca como mínimo en un camino (es decir, entre los  $k$  caminos recorreremos todas las ciudades)

El Cartero chino: Un cartero (chino) desea poder repartir el correo por su zona en el mínimo tiempo posible, para ello necesita obtener un recorrido que pase por todas las calles al menos una vez (se supone que tenemos el mapa de todas las calles de la zona) (también se puede plantear la misma variante de  $k$  caminos)

Máximo Clique: Un clique es un grafo en el que cada vértice está conectado con el resto de vértices del grafo (o sea que es un grafo completo). El problema consiste en encontrar para un grafo el mayor subgrafo que sea un clique.

Mínimo k-árbol de expansión mínima: Dado un grafo en el que cada arista tiene un peso se trata de encontrar el subgrafo sin ciclos (árbol) que contenga  $k$  aristas y que tenga el mínimo peso.

Mínimo árbol de Steiner: Dado un grafo completo (todos los vértices conectados con todos), donde cada arco tiene un peso, y un subconjunto de vértices del grafo, obtener el grafo de coste mínimo

que contenga el conjunto de vértices (el grafo puede contener mas vértices, la gracia es conectar unos vértices específicos con el mínimo coste)

Coloreado de grafos: Dado un grafo y  $k$  colores, asignar a cada vértice del grafo un color de manera que dos vértices conectados no tengan el mismo color.

- (a) Define que elementos forman el estado, el estado inicial y cual es estado final o que propiedades ha de cumplir, estima el tamaño del espacio de estados. Piensa que puede haber diferentes formas de plantear el problema.
- (b) Define las características de los operadores para realizar la búsqueda (condiciones de aplicabilidad y función de transformación), evalúa cual sería el factor de ramificación (aproximadamente).

8. Otros problemas típicos que se pueden plantear como búsqueda en espacio de estados son los que involucran horarios, piensa en los siguientes problemas:

Tienes que distribuir un conjunto de cursos de diferentes niveles ( $m$  cursos por cada nivel,  $n$  niveles) en un horario semanal (5 días) en un aula sabiendo que todos los cursos son de una hora, pero que no quieres que haya mas de  $k$  cursos de un nivel cada día Generaliza el problema suponiendo que tienes  $g$  grupos de cada curso y  $a$  aulas y no quieres que haya dos cursos iguales a la misma hora Supón que tienes que hacer tu horario cuatrimestral y dispones de  $m$  asignaturas, cada una de ellas con  $n$  grupos y quieres elegir un subconjunto  $k$  de ellas ( $k < m$ ) sin que se solapen sus horarios

- (a) Define que elementos forman el estado, el estado inicial y cual es estado final o que propiedades ha de cumplir, estima el tamaño del espacio de estados. Piensa que puede haber diferentes formas de plantear el problema.
- (b) Define las características de los operadores para realizar la búsqueda (condiciones de aplicabilidad y función de transformación), evalúa cual sería el factor de ramificación (aproximadamente).

9. El problema del club de golf. Tenemos un club de golf con 32 jugadores, todos ellos juegan una vez a la semana en grupos de 4 personas. Queremos obtener los emparejamientos para el mayor numero de semanas de manera que un jugador no este en el mismo grupo que otro jugador mas de una vez.

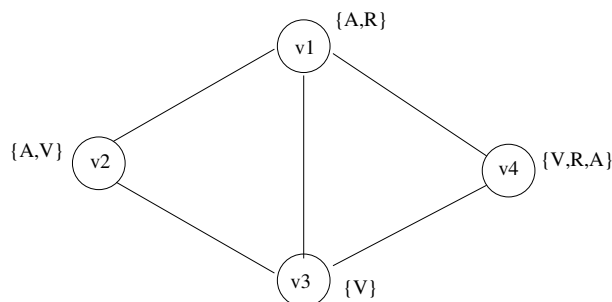
- (a) Define que elementos forman el estado, el estado inicial y cual es estado final o que propiedades ha de cumplir, estima el tamaño del espacio de estados. Piensa que puede haber diferentes formas de plantear el problema.
- (b) Define las características de los operadores para realizar la búsqueda (condiciones de aplicabilidad y función de transformación), evalúa cual sería el factor de ramificación (aproximadamente).



## 2. Cerca Heurística

1. El problema del coloreado de grafos consiste en etiquetar con un color cada vértice del grafo, de forma que no haya dos vértices adyacentes con el mismo color. Este problema se puede resolver por medio de búsqueda.

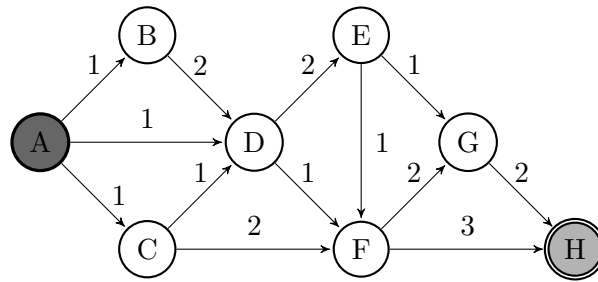
- (a) Describe los pasos que seguiría un algoritmo  $A^*$  para encontrar una solución al coloreado del siguiente grafo:



en donde los vértices son  $V1$ ,  $V2$ ,  $V3$  y  $V4$  y los colores posibles para cada vértice se indican entre llaves (A:azul, R:rojo, V:verde). El estado inicial tiene todos los vértices sin color asignado. Un estado  $S'$  es sucesor de otro  $S$  si  $S'$  contiene la asignación de colores de  $S$  más la asignación de un color al vértice más pequeño sin colorear de  $S$  (es decir, los vértices se instancian en el orden  $V1 \rightarrow V4$ ). La función heurística es:

$h(s) = \text{número de vértices sin color en } s + 2 \times \text{número de vértices adyacentes con el mismo color en } s$

- (b) ¿Es admisible la función heurística? Razona la respuesta.
  - (c) Sugiere otro algoritmo que solucione este problema de forma más eficiente que  $A^*$
2. Dado el siguiente grafo, donde cada arco indica su coste, y la tabla que indica la estimación del coste  $h$  hasta la solución, indica cual sería el árbol de búsqueda que se obtendría mediante el algoritmo de  $A^*$  e  $IDA^*$  para encontrar el camino entre el nodo A y el nodo H. Haz la generación de los nodos siguiendo el orden alfabético e indica claramente las reexpansiones de los nodos y los cambios de coste que aparezcan. ¿Es la función heurística admisible?



Nodo	A	B	C	D	E	F	G	H
h(nodo)	4	4	3	3	2	3	2	0

3. Aplicar l'algoritme A\* per a trobar un camí solució del graf descrit a la taula:

	b	c	d	e	h
a	10	4			3
b			10		4
c			12	2	12
d					0
e	2		4		4

on els nodes de les files són els nodes-pares i els nodes de les columnes els nodes-fills. El node a és l'inicial i el final és d. Cada casella de la taula indica el cost de l'arc que uneix els dos nodes corresponents. La darrera columna indica el valor de la funció heurística (h).

- Mostrar l'arbre de cerca generat per l'algoritme, indicant l'ordre d'expansió dels nodes, reopertures de nodes tancats, etc. Quin és el camí trobat? Quin és el seu cost? és l'òptim?
- Es verifiquen les condicions d'admissibilitat? Justifiquen la resposta.
- Fer el mateix amb l'algoritme IDA\*.

4. Disponemos de un casillero con cuatro monedas colocadas de la siguiente forma:

A	R	A	R	
---	---	---	---	--

El anverso de la moneda está representado por A y el reverso por R. Son posibles los siguientes movimientos:

- Desplazamiento (coste=1): Una moneda puede ser desplazada a la casilla contigua si ésta se encuentra vacía.
- Giro (coste=1): Cualquier moneda puede ser girada sin ninguna condición adicional. Sólo una cada vez.
- Salto (coste=2): Una moneda puede saltar sobre su vecina si a continuación hay una casilla vacía, es decir, sólo es posible saltar por encima de una moneda. Cuando una moneda salta, cae realizando un giro. Un ejemplo de salto (coste=2) es pasar del estado AR\_RA al estado ARRR\_

Deseamos obtener la situación final siguiente:

	R	A	R	A
--	---	---	---	---

Dada la función heurística  $h(n) = p_2 + p_3 + p_4 + p_5 + dv$

donde  $p_i$  vale 0 si la casilla  $i$  contiene la asignación correcta respecto del estado final y vale 1 en caso contrario y  $dv$  es la distancia del blanco respecto a la posición final (casilla 1).

Por ejemplo,  $h(\text{estado inicial}) = 1 + 4 = 5$

- (a) 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.
- (b) El camino obtenido es de coste óptimo? El heurístico usado es admisible? Por qué?
- (c) Resolver el problema aplicando IDA\*. Indica claramente el proceso en cada nivel: orden de expansión de los nodos, nodos duplicados... Se obtiene un camino de coste óptimo?
5. 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,

- (a) 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.
- (b) ¿El camino obtenido es de coste óptimo? ¿El heurístico usado es admisible? ¿Por qué?
- (c) Resolver el problema aplicando IDA\*. Indica claramente el proceso en cada nivel: orden de expansión de los nodos, nodos duplicados ¿Se obtiene un camino de coste óptimo?
6. Supón que tenemos un computador en el que varias instrucciones se pueden ejecutar a la vez. La unidad que ejecuta las instrucciones puede ejecutar en paralelo instrucciones de tres tipos: A, B y C. La instrucción A tarda 3 ciclos de reloj, la instrucción B tarda 2 ciclos y la instrucción C tarda 1 ciclo. Esta unidad es capaz de ejecutar en un paso tantas instrucciones como quepan dentro de una ventana de 3 ciclos de reloj, de manera que se puede utilizar para ejecutar a la vez varios procesos (hilos de ejecución). Por ejemplo, si tenemos estos 3 hilos de instrucciones H1(ABC) H2(BCA) H3(CCC) como estado inicial, la unidad podría ejecutar las instrucciones de la manera que aparecen en la tabla:

	A	B	C
1 <sup>er</sup> paso	H1	H2	H3
	H1	H2	H3
	H1		H2

Se ejecuta la instrucción A del primer hilo en paralelo con la B del segundo hilo, las dos primeras C del tercer hilo y finalmente la C del segundo hilo respetando así la precedencia de instrucciones del segundo hilo.

Estado actual: H1(BC), H2(A), H3(C)

	A	B	C
2 <sup>o</sup> paso	H2	H1	H3
	H2	H1	
	H2		H1

Se ejecuta la instrucción A del segundo hilo en paralelo con la B del primer hilo, la tercera C del tercer hilo y finalmente la C del primer hilo respetando así la precedencia de instrucciones del primer hilo.

La secuencia de ejecución sería: 1er paso : [(A,H1),(B,H2),(C,H3),(C,H3),(C,H2)],

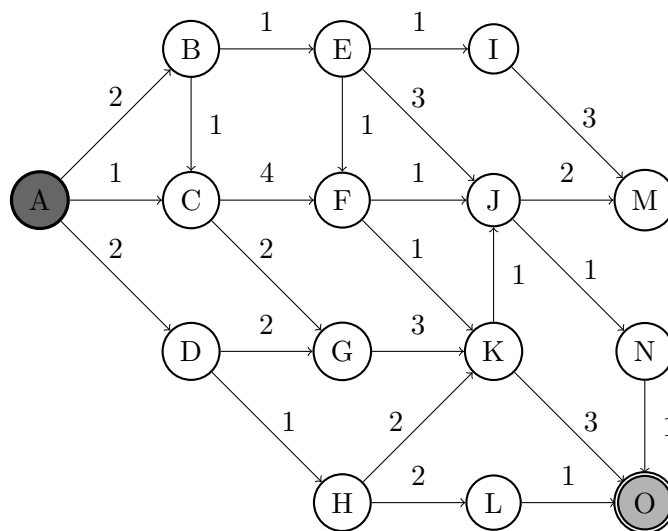
2o paso: [(A,H2),(B,H1),(C,H3),(C,H2)]

La idea es secuenciar varios hilos en paralelo en tiempo de compilación, de manera que el número de pasos de ejecución sea el mínimo. La búsqueda consiste en determinar qué conjunto de instrucciones y de que hilos se pueden ejecutar en cada paso de la unidad de ejecución. En caso de que se pueda elegir entre varias instrucciones iguales en un paso, el orden de exploración se seguirá escogiendo la instrucción según el orden de los hilos.

Los hilos a compilar serán: H1(ABAC) H2 (CABA) H3(ACBA)

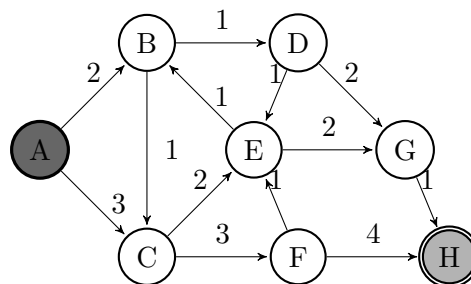
- (a) Utiliza el algoritmo del A\* para encontrar la secuencia de menos pasos para estos hilos de ejecución (indica claramente el orden de expansión de los nodos). Considera que el coste de cada paso es 1. Como heurístico para la búsqueda utiliza el siguiente:  
 $h_1(n)$  = número de instrucciones del hilo con más instrucciones pendientes
- (b) El heurístico no es admisible ¿Por qué? ¿La solución es óptima?
- (c) Supón el heurístico:  
 $h_2(n)$  = número de instrucciones A que quedan por ejecutar  
 ¿Sería admisible este heurístico?
- (d) Utiliza ahora el algoritmo del IDA\* con este nuevo heurístico para hacer la exploración (realiza cada iteración por separado indicando el orden de expansión de los nodos)

7. Dado el siguiente grafo donde cada arco indica su coste y la tabla que indica la estimación del coste  $h$  hasta la solución, indica cual sería el árbol de búsqueda que se obtendría mediante el algoritmo de A\* e IDA\* para encontrar el camino entre el nodo A y el nodo O. Haz la generación de los nodos siguiendo el orden alfabético e indica claramente las repeticiones de los nodos y los cambios de coste que aparezcan. ¿Es la función heurística admisible?



Nodo	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
$h(\text{nodo})$	6	5	6	6	3	5	5	4	8	3	2	1	5	1	0

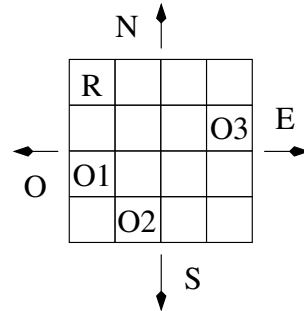
8. Dado el siguiente grafo, donde cada arco indica su coste, y la tabla que indica la estimación del coste  $h$  hasta la solución, indica cual sería el árbol de búsqueda que se obtendría mediante el algoritmo de A\* e IDA\* para encontrar el camino entre el nodo A y el nodo H. Haz la generación de los nodos siguiendo el orden alfabético e indica claramente las reexpansiones de los nodos y los cambios de coste que aparezcan. ¿Es la función heurística admisible?



Nodo	A	B	C	D	E	F	G	H
$h(\text{nodo})$	6	4	3	3	2	3	2	0



9. Deseamos hallar la ruta que ha de seguir un robot para recoger un conjunto de objetos en un orden específico en una habitación. La situación es la que se presenta en la figura. El robot sólo se puede desplazar en vertical y en horizontal. En cada movimiento sólo se desplaza una casilla. Cada vez que el robot llega a la casilla donde hay un objeto lo recoge y lo lleva consigo.



- (a) Utiliza el algoritmo del A\* para hallar el camino que permite al robot recoger los 3 objetos en orden usando la función heurística:

$$h_1(n) = \sum d_{rj}$$

Donde  $d_{rj}$  es la distancia en movimientos del robot al objeto j. Considera que, cuando el robot recoge un objeto, su distancia pasa a ser 0 y que el objeto se mueve con el robot una vez recogido. Para hacer la expansión de los nodos utiliza el orden N-S-E-O tal como se indica en la figura. Indica el orden de expansión de los nodos. ¿El camino encontrado es óptimo?

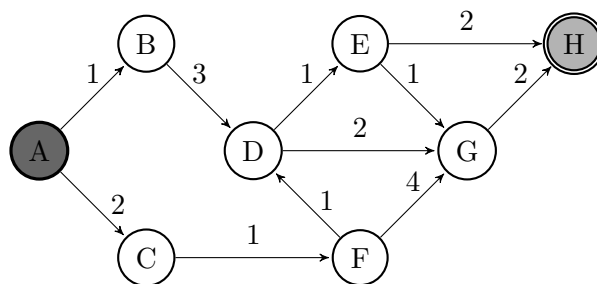
- (b) Supón que el heurístico es:

$$h_2(n) = d_{r1} + d_{12} + d_{23}$$

Donde  $d_{r1}$  es la distancia en movimientos del robot al objeto O1,  $d_{12}$  es la distancia en pasos del objeto O1 al objeto O2 y  $d_{23}$  es la distancia del objeto O2 al objeto O3. Utiliza de nuevo el algoritmo del A\* para hallar la solución. Para hacer la expansión de los nodos utiliza el orden N-S-E-O tal como se indica en la figura. Indica el orden de expansión de los nodos. ¿El camino encontrado es óptimo?

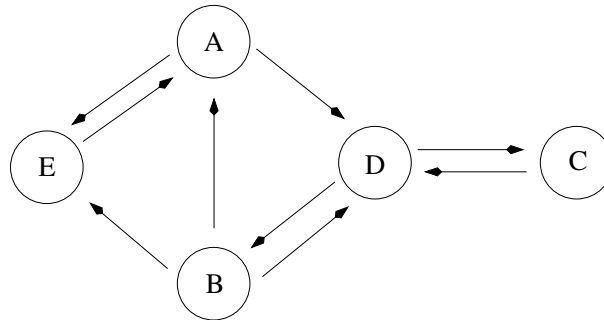
- (c) Los heurísticos usados son admisibles? Podemos afirmar que uno es más informado que el otro? Razona las respuestas.
- (d) Haz de nuevo la búsqueda utilizando el algoritmo del IDA\* y el heurístico  $h_1$ . Indica el orden de expansión de los nodos. ¿El camino encontrado es óptimo?

10. Dado el siguiente grafo, donde cada arco indica su coste, y la tabla que indica la estimación del coste  $h$  hasta la solución, indica cual sería el árbol de búsqueda que se obtendría mediante el algoritmo de A\* e IDA\* para encontrar el camino entre el nodo A y el nodo H. Haz la generación de los nodos siguiendo el orden alfabético e indica claramente las reexpansiones de los nodos y los cambios de coste que aparezcan. ¿Es la función heurística admisible?



Nodo	A	B	C	D	E	F	G	H
h(nodo)	4	4	3	3	2	3	2	0

11. Dado el siguiente grafo



- Buscar, aplicando  $A^*$ , un recorrido que pase por todos los nodos empezando y acabando por el nodo A. Sólo es posible pasar de un nodo X a un nodo Y si existe el arco  $X \rightarrow Y$ . Se permite pasar más de una vez por el mismo nodo. En cada iteración indica el orden de expansión de los nodos. Utiliza como función heurística el número de nodos nuevos pendientes de visitar. Orden de generación de sucesores: orden lexicográfico.
- ¿El camino encontrado es óptimo? Justifica formalmente la respuesta.
- Repite la búsqueda utilizando el algoritmo  $IDA^*$

12. Tenemos el siguiente rompecabezas con la configuración de la figura

N	N	N	B	B	B
---	---	---	---	---	---

y disponemos de una única regla, dos fichas se pueden intercambiar si están adyacentes, considerando que los dos extremos son adyacentes (es un tablero circular).

El objetivo es obtener la siguiente configuración:

B	B	N	B	N	N
---	---	---	---	---	---

- Aplica el algoritmo  $A^*$  para resolver el puzzle usando la siguiente función heurística  $h(n)$  = número de posiciones diferentes entre el estado actual y la configuración final.  
ORDEN: Al expandir los nodos considera los movimientos secuencialmente, empezando por el de más a la izquierda.  
Indica el orden de expansión de los nodos y la gestión de duplicados.
- Resuelve el mismo problema aplicando el algoritmo del  $IDA^*$ . Indica el orden de expansión de los nodos y la gestión de duplicados.
- ¿Es admisible el heurístico? ¿Por qué?

13. Un problema en genética es averiguar si dos cadenas de ADN son similares o no. Para ello se intentan alinear los símbolos de las secuencias y asignar una puntuación al alineamiento. El problema es que hay muchas maneras de alinear dos secuencias y sólo interesa la mejor.

Para simplificar el problema, vamos a suponer que tenemos secuencias con solamente dos símbolos: A y B. El alineamiento de dos secuencias se hace símbolo a símbolo de izquierda a derecha. Como es posible que las dos secuencias sean de longitud diferente, se pueden introducir símbolos blancos durante el alineamiento para compensarlo, de manera que al final del alineamiento las dos cadenas tengan la misma longitud. Podemos haber introducido símbolos blancos en ambas cadenas.

Tenemos dos operaciones:

- Crear un emparejamiento entre dos símbolos de la secuencia. El coste de esta operación es cero si los dos símbolos coinciden y uno, si no coinciden.
- Añadir un blanco en una secuencia, de manera que avanzamos un símbolo en una de las secuencias y la otra la alargamos con este blanco. El coste de esta operación es dos.

Por ejemplo, si tenemos las secuencias AABB y BAA, podríamos alinear los dos primeros símbolos, de manera que estaríamos en el estado (A|ABB, B|AA) con coste uno ya que los símbolos no coinciden (la barra vertical | indica hasta donde llega el alineamiento que llevamos), o podríamos añadir un hueco en la primera secuencia, de manera que estaríamos en el estado (\_|AABB, B|AA), con coste dos.

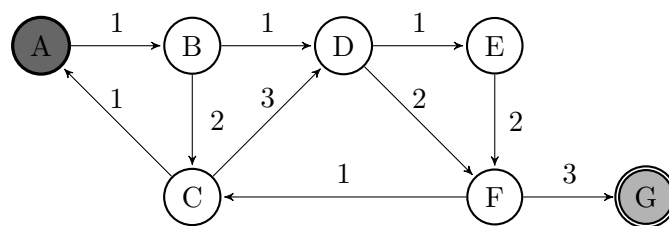
- (a) Aplicar el algoritmo A\* para alinear las secuencias AB y BAAB, usando como función heurística  $h(n) = |\text{longitud del trozo de la secuencia 1 por alinear} - \text{longitud del trozo de la secuencia 2 por alinear}|$

Al expandir los nodos, en el caso de añadir blancos, primero se añadirá a la primera secuencia y después a la segunda. Indica el orden de expansión de los nodos y la gestión de duplicados, si procede.

- (b) Resuelve el mismo problema aplicando el algoritmo del IDA\*. Indica el orden de expansión de los nodos y la gestión de duplicados, si procede.

- (c) ¿Es admisible el heurístico? ¿Por qué?

14. Dado el siguiente grafo donde cada arco indica su coste y la tabla que indica la estimación del coste  $h$  hasta la solución, indica cual sería el árbol de búsqueda que se obtendría mediante el algoritmo de A\* e IDA\* para encontrar el camino entre el nodo A y el nodo G. Haz la generación de los nodos siguiendo el orden alfabético e indica claramente las reexpansiones de los nodos y los cambios de coste que aparezcan. ¿Es la función heurística admisible?



Nodo	A	B	C	D	E	F	G
$h(\text{nodo})$	3	4	3	4	3	3	0

15. Dada la siguiente configuración inicial del tablero:

N	N	B	B	
---	---	---	---	--

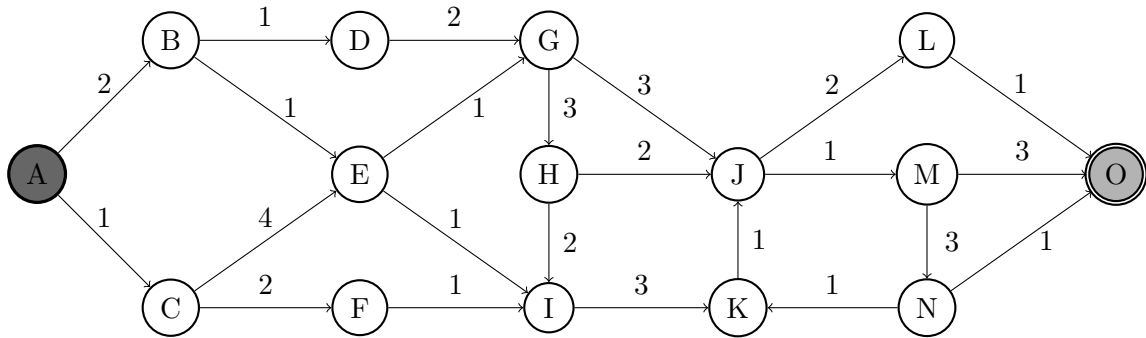
donde hay dos piezas negras (N), dos blancas (B) y una posición vacía, y teniendo en cuenta las siguientes reglas:

- Una pieza puede moverse a la posición adyacente vacía con coste 1.
- Una pieza puede saltar por encima de otra (sólo una) para colocarse en la posición vacía con coste 1. Si la pieza que salta es de distinto color que la saltada, esta última cambia de color.

Queremos conseguir que todas las piezas del tablero sean negras. La posición final de la casilla vacía es indiferente.

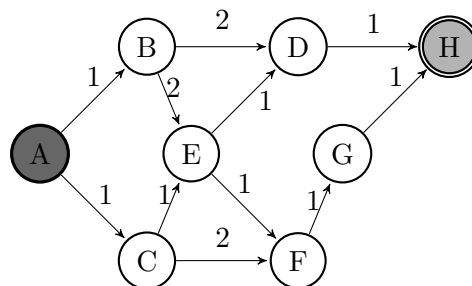
- Muestra el árbol de búsqueda generado por el algoritmo A\* indicando el orden de expansión de los nodos, el tratamiento de los duplicados, la solución encontrada y el coste de la misma. Utiliza como heurístico:  $h(n) = \text{número de piezas blancas}$
- Justifica la admisibilidad o no del heurístico utilizado.
- Considera ahora el siguiente heurístico:  $h_2(n) = \text{número de piezas blancas} - c$  donde  $c$  vale 1 si el espacio es adyacente a una pieza blanca y vale 0 en caso contrario. Justifica su admisibilidad o no.
- Razona si  $h_2(n)$  es más informado que  $h(n)$ .

16. Dado el siguiente grafo donde cada arco indica su coste y la tabla que indica la estimación del coste  $h$  hasta la solución, indica cual sería el árbol de búsqueda que se obtendría mediante el algoritmo de A\* e IDA\* para encontrar el camino entre el nodo A y el nodo O. Haz la generación de los nodos siguiendo el orden alfabético e indica claramente las reexpansiones de los nodos y los cambios de coste que aparezcan. ¿Es la función heurística admisible?



Nodo	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
$h(\text{nodo})$	6	5	6	6	3	5	5	4	8	3	2	1	5	1	0

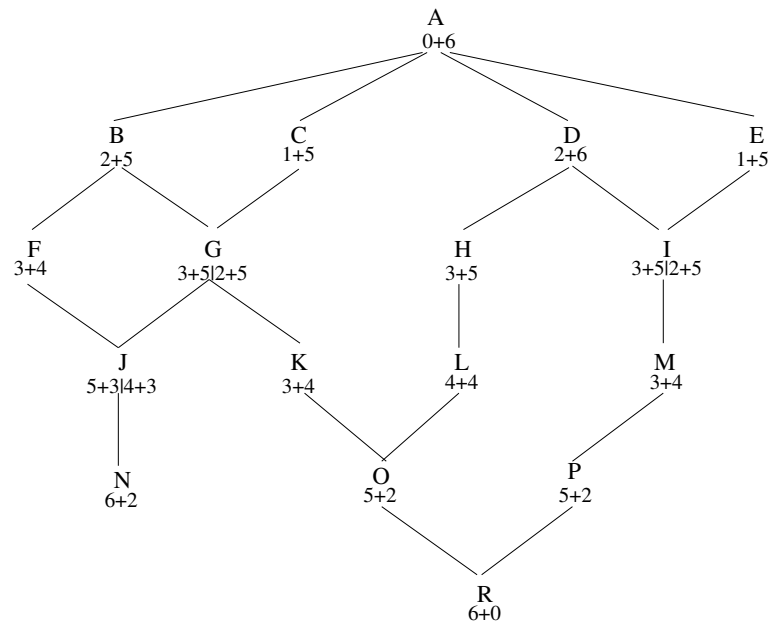
17. Dado el siguiente grafo, donde cada arco indica su coste, y la tabla que indica la estimación del coste  $h$  hasta la solución, indica cual sería el árbol de búsqueda que se obtendría mediante el algoritmo de A\* e IDA\* para encontrar el camino entre el nodo A y el nodo H. Haz la generación de los nodos siguiendo el orden alfabético e indica claramente las reexpansiones de los nodos y los cambios de coste que aparezcan. ¿Es la función heurística admisible?



Nodo	A	B	C	D	E	F	G	H
$h(\text{nodo})$	6	4	4	4	4	3	3	0

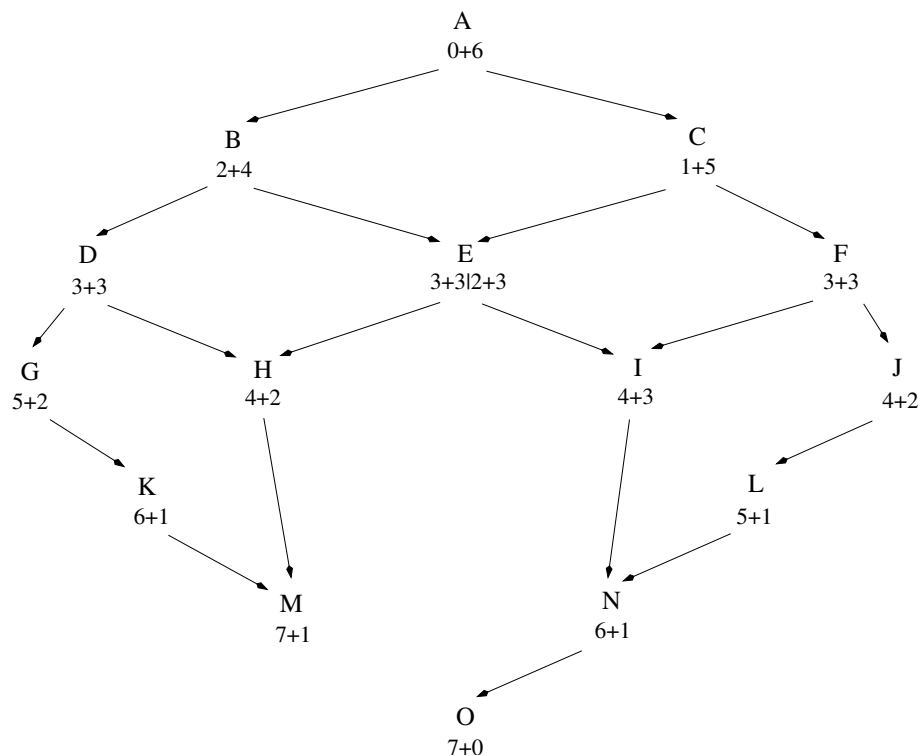
18. Dado el árbol de búsqueda que aparece en la figura indica cual sería el recorrido que haría el algoritmo A\* e IDA\*. ¿La función heurística es admisible?

En el árbol la función heurística está descompuesta en la  $g$  y la  $h$ . Cuando en un nodo aparecen dos valores el orden corresponde a la rama por la que se ha llegado al nodo.



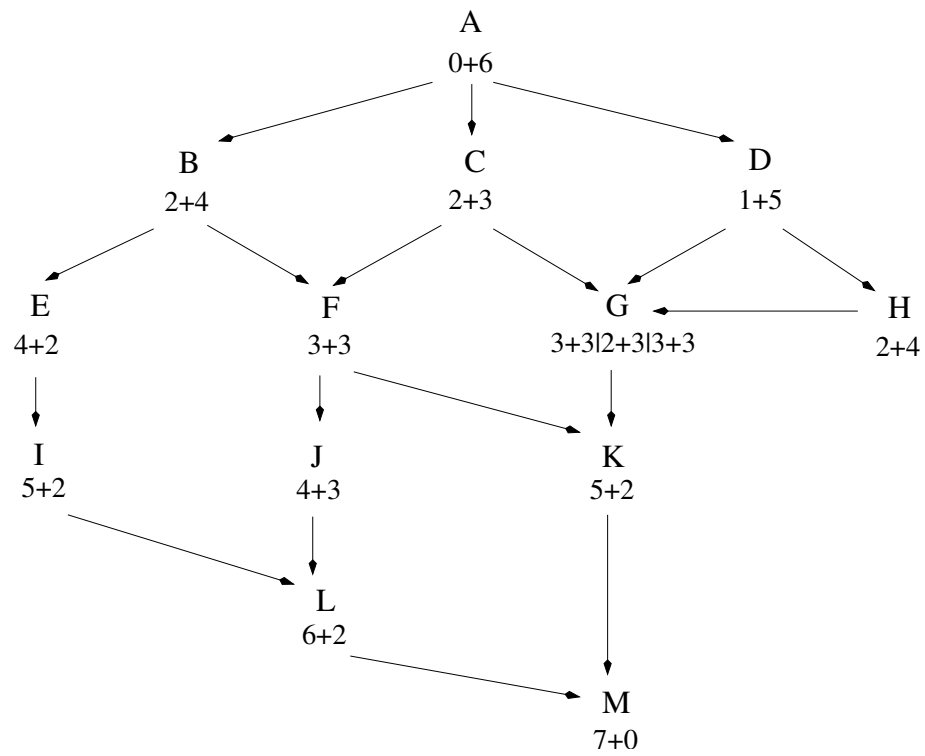
19. Dado el árbol de búsqueda que aparece en la figura indica cual sería el recorrido que haría el algoritmo A\* e IDA\* ¿la función heurística es admisible?

En el árbol la función heurística esta descompuesta en la g y la h. Cuando en un nodo aparecen dos valores el orden corresponde a la rama por la que se ha llegado al nodo.



20. Dado el árbol de búsqueda que aparece en la figura indica cual sería el recorrido que haría el algoritmo A\* e IDA\* ¿la función heurística es admisible?

En el árbol la función heurística esta descompuesta en la g y la h. Cuando en un nodo aparecen dos valores el orden corresponde a la rama por la que se ha llegado al nodo.



### 3. Cerca Local

1. Existen problemas en los que dado el tamaño de su espacio de búsqueda es imposible plantearse encontrar la solución óptima. Para este tipo de problemas se han diseñado algoritmos de búsqueda local (hill-climbing, simulated annealing, algoritmos genéticos)

Estos son ejemplos de este tipo de problemas:

- (a) El viajante de comercio:

Un viajante de comercio desea visitar un conjunto de ciudades partiendo de una dada y acabando en ésta, sin repetir ninguna y recorriendo el mínimo de distancia (se supone que se dispone de un mapa que indica las conexiones entre ciudades y sus distancias)

- (b) Mínimo k-árbol de expansión mínima:

Dado un grafo en el que cada arista tiene un peso, se trata de encontrar el subgrafo sin ciclos (árbol) que contenga k aristas y que tenga el mínimo peso

- (c) Mínimo árbol de Steiner:

Dado un grafo completo (todos los vértices están conectados con todos), donde cada arco tiene un peso, y un subconjunto de vértices del grafo, obtener el grafo de coste mínimo que contenga el conjunto de vértices (el grafo puede contener mas vértices, la gracia es conectar unos vértices específicos con el mínimo coste)

- (d) Bin packing en una dimensión:

Disponemos de un conjunto de contenedores y un conjunto de paquetes que colocar en ellos. Los contenedores son todos iguales y se pueden llenar hasta cierta altura. Los paquetes tienen las mismas dimensiones en su base que los contenedores, pero diferentes alturas, siempre por debajo de la altura de los contenedores. El problema consiste en: dado un conjunto de paquetes saber cual es el mínimo conjunto de contenedores necesario para transportarlos.

- (e) Knapsack problem:

También conocido como el problema de la mochila. Dado un conjunto de objetos con cierto volumen y cierto valor y un contenedor con una capacidad fija, hallar el conjunto de objetos que nos debemos llevar para maximizar la ganancia.

- (f) Weighted max-cut:

Dado un grafo con pesos en los arcos, partirlo en dos grafos de manera que se maximice la suma total de los arcos de los dos subgrafos

Responde para cada problema los siguientes apartados:

- Determina que elementos forman el estado y estima el tamaño del espacio de búsqueda
  - Dado que los algoritmos que se pueden utilizar suelen partir de una solución, busca métodos con los que se puede encontrar la solución de partida (no tiene por que ser buena)
  - Define lo necesario para solucionar el problema mediante el algoritmo de hill-climbing (función heurística, operadores de cambio de estado)
  - Define lo necesario para solucionar el problema mediante algoritmos genéticos (representación del estado, función de calidad, operadores de cruce y mutación)
2. Los propietarios del hotel “Bienestar” trabajan habitualmente con  $N$  tour-operadores para los cuales deben reservar siempre un conjunto de habitaciones. A principios de año, cada operador realiza su petición de habitaciones para todo el año. Las peticiones son consideradas como el número máximo a reservar para cada tour-operador, pero los propietarios saben que no tienen habitaciones suficientes para cubrir todas las peticiones de todos los tour-operadores. Por esta razón, internamente tienen definido un número mínimo a asignar a cada tour-operador. Este número les permite mantener las buenas relaciones. A partir de aquí asignarán a los tour-operadores un número de habitaciones que estará entre el mínimo interno y la petición real recibida.

Para asignar las habitaciones, los propietarios tienen varias restricciones a respetar. Por un lado, hay un número mínimo de habitaciones que queda siempre bajo la gestión directa del hotel. Por otro lado, se quiere maximizar los beneficios obtenidos de las habitaciones reservadas, teniendo en cuenta que los beneficios  $B_i$  son distintos dependiendo del tour-operador y el beneficio por habitación gestionada por el hotel es  $B_H$ . Adicionalmente, se quiere maximizar la “calidad de la ocupación” para lo cual tienen asignado a cada tour-operador un índice  $Q_i$  que cuantifica la calidad de los turistas que suelen venir a través del operador en cuestión. Este índice se asociará a cada habitación reservada para ese operador. El propio hotel tiene también un índice  $Q_H$ .

En los siguientes apartados se proponen diferentes alternativas para algunos de los elementos necesarios para plantear la búsqueda (solución inicial, operadores, función heurística, ...). El objetivo es comentar la solución que se propone respecto a si es correcta, es eficiente, o es mejor o peor respecto a otras alternativas posibles. Justifica tu respuesta.

- (a) Se plantea aplicar Hill-climbing usando como solución inicial asignar el mínimo de habitaciones a cada tour-operador. Se define como operador asignar una habitación a alguien distinto del que la tiene asignada (tour-operador/hotel), siempre y cuando el nuevo tour-operador tenga mejor índice  $Q$  y se respeten todos los mínimos. Se plantea como función de evaluación la suma total de los beneficios obtenidos de cada habitación del hotel.
- (b) Se plantea aplicar Hill-climbing usando como solución inicial asignar el mínimo de habitaciones a cada tour-operador y del conjunto de habitaciones no asignadas apartar el mínimo para el hotel y asignar el resto al tour-operador que nos da mayor beneficio. Se define como operador asignar una habitación a un tour-operador distinto del que la tiene asignada. Se plantea como función de evaluación la suma total de los beneficios obtenidos por habitación más la suma del índice  $Q$  de cada una de ellas.
- (c) Se plantea usar algoritmos genéticos donde la representación de la solución es una tira de bits donde hay  $(N + 1)$  secuencias bits. Para cada tour-operador y el hotel tenemos asociadas una secuencia de bits que representa el número de habitaciones asignadas. Cada secuencia de bits ha de ser suficiente para codificar en binario el número total de habitaciones del hotel. Como solución inicial se asigna el mínimo de habitaciones a cada tour-operador y el resto al hotel. Los operadores a usar son los habituales de cruce y mutación. La función de evaluación valdrá infinito cuando no se cumplan todos los mínimos, y si se cumplen tendrá la siguiente fórmula:

$$h(n) = num\_habitaciones_H \frac{B_H}{Q_H} + \sum_{i=1}^N num\_habitaciones_i \frac{B_i}{Q_i}$$



3. Una empresa de transporte marítimo desea decidir la configuración de la carga de su próximo barco. La empresa recibe un conjunto de peticiones de envío de entre las que escoger. Cada petición tiene asociado un precio de transporte.

Cada petición va dentro de un tipo contenedor que está fijado por el tamaño, el peso del envío y las características de la carga. Existen solamente  $K$  tipos de contenedor. Los contenedores son propiedad de la empresa y su uso tiene un coste asociado que depende del tipo de contenedor.

Existen diferentes restricciones para la carga: la suma total de pesos de los contenedores no ha de sobrepasar la capacidad de carga del barco  $P_{max}$ , ni ha de ser inferior a cierto valor  $P_{min}$ . Cada contenedor tiene un peso asociado que depende de la carga que contiene. Las posibilidades de colocar los contenedores en el barco también imponen que haya un mínimo de  $C_{min}$  contenedores y un máximo de  $C_{max}$  contenedores de cada tipo.

El objetivo es encontrar la combinación de peticiones que maximicen el precio de transporte y minimicen el coste y que esté dentro de las restricciones impuestas.

En los siguientes apartados se proponen diferentes alternativas para algunos de los elementos necesarios para plantear la búsqueda (solución inicial, operadores, función heurística, ...). El objetivo es comentar la solución que se propone respecto a si es correcta, es eficiente, o es mejor o peor respecto a otras alternativas posibles. Justifica tu respuesta.

- (a) Se plantea solucionarlo mediante Hill-climbing utilizando como solución inicial el barco vacío y como operadores añadir y quitar contenedores del barco. Queremos usar como función de evaluación de las soluciones lo siguiente:

$$h(n) = \sum_{i=0}^{Ncont} Precio_i \times \sum_{i=0}^{Ncont} Coste_i$$

Donde  $Ncont$  es el número de contenedores que hay en una solución,  $Precio_i$  es el precio de transporte del contenedor  $i$  de la solución y  $Coste_i$  es el coste de un contenedor  $i$  de la solución.

- (b) Se plantea solucionarlo mediante Hill-climbing utilizando como solución inicial escoger al azar  $C_{min}$  contenedores de cada tipo. Como operador utilizamos intercambiar un contenedor del barco por otro que no esté en él. Queremos usar como función de evaluación de las soluciones lo siguiente:

$$h(n) = \sum_{i=0}^{Ncont} \frac{Precio_i}{Coste_i}$$

- (c) Se plantea utilizar algoritmos genéticos donde la representación de la solución es una tira de bits con tantos bits como peticiones haya, donde cada bit significa si la petición está o no en el barco. Para generar la población inicial usamos la misma técnica que en el apartado anterior. Usamos los operadores habituales de cruce y mutación y como función de evaluación usamos la del apartado anterior pero haciendo que valga cero si la solución incumple las restricciones del problema.

4. Una empresa del mundo del espectáculo gestiona los contratos de  $N$  grupos musicales que se dedican a actuar en las fiestas de los pueblos. Cada grupo musical tiene una tarifa por actuación, de la que una parte se la queda el grupo y otra la empresa. Por otro lado, los municipios celebran sus fiestas patronales en unas ciertas fechas y dedican una parte de su presupuesto festivo,  $X$  euros, a pagar las actuaciones de los grupos musicales. El problema es organizar las giras de los grupos de modo que se cubran las peticiones de los pueblos, en total  $P$  actuaciones (puede haber más de una actuación en un pueblo en días consecutivos), sin pasarse del presupuesto municipal, se maximice la recaudación de la empresa y se minimicen los kilómetros que al cabo de la temporada ha de realizar cada grupo. Además, el mismo grupo no puede actuar más de un día en el mismo pueblo y cuando acaban en un pueblo, se van al siguiente en el que tienen actuación.

En los siguientes apartados se proponen diferentes alternativas para algunos de los elementos necesarios para plantear la búsqueda (solución inicial, operadores, función heurística, ...). El objetivo es comentar la solución que se propone respecto a si es correcta, es eficiente, o es mejor o peor respecto a otras alternativas posibles. Justifica tu respuesta.

- (a) Se propone usar Hill-Climbing. La solución de partida coloca, en orden creciente de fecha, un grupo diferente en cada una de las fechas previstas de actuación y cada grupo se asigna una sola vez. El operador disponible es **poner\_grupo** y la función heurística es:

$$h'(n) = \frac{\sum_{\forall grupo_i} Kilometros\_recorridos(grupo_i)}{\sum_{\forall grupo_i} Dinero\_recaudado\_empresa(grupo_i)}$$

- (b) Se propone usar Hill-Climbing. La solución de partida se calcula de la siguiente manera: se ordenan los grupos en orden decreciente de beneficio para la empresa y se empiezan a asignar siguiendo ese orden a cada una de las fechas disponibles. Si cuando ya se han colocado todos los grupos una vez todavía quedan fechas sin cubrir, se repite el proceso las veces que sea necesario. Se respeta que el mismo grupo no actúe más de una vez en el mismo pueblo. El operador es **intercambiar\_grupo** siempre y cuando no suponga que se supera el presupuesto del municipio y se respeta una única actuación por grupo en el mismo pueblo. La función heurística es:

$$h'(n) = \sum_{\forall grupo_i} Kilometros\_recorridos(grupo_i) - \sum_{\forall grupo_i} Dinero\_recaudado\_empresa(grupo_i) * 100$$

- (c) Se plantea resolverlo con algoritmos genéticos. Se asigna un identificador binario de longitud  $b$  a cada uno de los grupos, la solución se representa como una tira de  $P \times b$  bits. La solución de partida se calcula de la siguiente forma: se ordenan los grupos en orden decreciente de beneficio para la empresa, se toma el primer grupo y se le asigna la primera fecha libre sin pasarse del presupuesto del municipio, luego se busca la siguiente fecha libre en la que pueda actuar el mismo grupo, y tal que suponga un desplazamiento mínimo para el grupo y que no supere el presupuesto del municipio, se le asigna el grupo y se repite el proceso hasta que no queden más fechas posibles para ese primer grupo. Se repite exactamente el mismo proceso para los grupos sucesivos hasta que ya no queden fechas por cubrir o no haya más grupos.

Como operadores genéticos se utiliza solamente el operador de cruce, donde los puntos de cruce no pueden estar dentro de la identificación de un grupo. El heurístico es el siguiente:

$$h'(n) = \sum_{\forall grupo_i} Kilometros\_recorridos(grupo_i) \times \frac{Dinero\_cobrado(grupo_i)}{Dinero\_recaudado\_empresa(grupo_i)}$$

5. Una empresa chocolatera desea comercializar una nueva caja de bombones que cumpla cierto conjunto de características: el precio ha de ser el mínimo posible, el beneficio que se obtenga ha de ser el máximo posible, ha de tener un peso limitado (no más de 500 g, ni menos de 400 g) y ha de estar compuesta de una combinación de los diferentes tipos de bombones que fabrica, estos pertenecen a  $K$  tipos diferentes, cada uno de ellos con un precio, un peso y un beneficio por bombón diferentes. La caja ha de contener no más de  $max_k$  y no menos de  $min_k$  bombones de cada tipo. Es posible que poniendo el mínimo de bombones de cada tipo no se llegue a los 400g pero nunca pasará de 500g.

En los siguientes apartados se proponen diferentes alternativas para algunos de los elementos necesarios para plantear la búsqueda (solución inicial, operadores, función heurística, ...). El objetivo es comentar la solución que se propone respecto a si es correcta, es eficiente, o es mejor o peor respecto a otras alternativas posibles. Justifica tu respuesta.

- (a) Se plantea solucionarlo mediante Hill-climbing utilizando como solución inicial una caja vacía y como operadores añadir o quitar  $n$  bombones de un tipo  $k$  en la caja. Como heurística usamos la suma de precios de los bombones en la caja.
- (b) Se plantea usar Hill-climbing. Como solución inicial se añaden  $min_k$  bombones de cada tipo  $k$ , luego se ordenan los tipos de bombones por beneficio, y se usa un greedy para completar la caja, añadiendo tantos bombones como haga falta (empezando por los de mayor beneficio) hasta llegar a un peso intermedio de 450g aprox., sin superar el  $max_k$  por tipo de bombón. Los operadores son

añadir un bombón sin pasar del peso máximo y quitar un bombón en la caja sin bajar del peso mínimo. Como heurística usamos la suma, para cada bombón en la caja, de su precio dividido por su beneficio.

- (c) Se plantea usar algoritmos genéticos, donde la representación de la solución es una tira de bits en la que hay  $K \cdot N$  bits ( $N$  se calcula como el máximo de los  $max_k$ ). Cada bit determina si el bombón está (1) o no (0) en la caja. Como generador de cada solución inicial se usa el mismo generador del apartado anterior. Usaremos los operadores habituales de cruce y mutación. La función de evaluación es la suma de los precios penalizando la solución con el valor  $+\infty$  cuando el peso está por encima o por debajo de los límites de peso.

6. Se han descubierto  $A$  fuentes de contaminación en un parque natural y se quieren colocar  $B$  (donde  $B < A$ ) aparatos de descontaminación para mejorar la situación. Para ello se dispone de un mapa del parque que indica la posición de la estación de trenes donde se han almacenado todos los aparatos y de los  $A$  lugares donde se necesita colocar los aparatos de descontaminación. Además también se dispone del nivel de contaminación que hay alrededor de cada fuente, de un mapa de los desplazamientos (dirigidos) posibles de los aparatos en el territorio y del coste de cada desplazamiento. Cada aparato puede eliminar por completo la contaminación de una fuente, independientemente de su nivel.

El objetivo es colocar los aparatos de manera que se minimice la contaminación total en el parque y el coste del recorrido (suma de desplazamientos) que harán los aparatos en el sentido “estación  $\rightarrow$  fuente de contaminación”.

En los siguientes apartados se proponen diferentes alternativas para algunos de los elementos necesarios para plantear la búsqueda (solución inicial, operadores, función heurística). Hay que comentar la solución que se propone respecto a si es correcta, es eficiente, y es mejor o peor en comparación con otras alternativas. Y hay que justificar todas las respuestas.

- (a) Se plantea solucionar el problema mediante Hill-Climbing, partiendo de una solución inicial sin ningún aparato y con un operador que coloca un aparato en una fuente de contaminación determinada, controlando que el número de los aparatos colocados sea como máximo  $B$ .
- (b) Se plantea solucionarlo mediante Hill-Climbing, partiendo de una solución inicial con  $B$  aparatos colocados aleatoriamente, y utilizando como función heurística la suma de los costes de desplazamiento de la estación a cada una de las  $B$  fuentes.
- (c) Se plantea solucionarlo mediante Hill-Climbing, partiendo de una solución inicial con  $B$  aparatos colocados aleatoriamente, y utilizando como función heurística la suma de los costes mínimos de los recorridos “estación  $\rightarrow$  fuente de contaminación” multiplicada por la suma de los niveles de contaminación en correspondientes a los  $B$  aparatos.
- (d) Se plantea solucionarlo mediante Hill-Climbing, partiendo de una solución inicial alcanzada colocando los  $B$  aparatos ordenadamente según el coste mínimo “estación  $\rightarrow$  fuente de contaminación” y empezando con el que tiene coste menor. Se plantea como operador mover un aparato a cualquier fuente cuyo producto de “coste mínimo estación  $\rightarrow$  fuente” por “nivel de contaminación” sea menor que el actual.
- (e) Se plantea solucionarlo mediante algoritmos genéticos: se usan individuos de  $A$  bits y como población inicial se generan  $n$  individuos donde en cada uno hay exactamente  $B$  bits a 1. La función de idoneidad es la suma de los costes mínimos “estación  $\rightarrow$  fuente de contaminación” más la contaminación total residual del parque multiplicada por una constante. Como operadores se usan los habituales de cruce y mutación.

7. Tenemos que planificar la topología de interconexión de un conjunto de routers que están distribuidos por el Campus Nord, de manera que podamos canalizar todo su tráfico hacia el equipo de comunicaciones que lo reenvía a internet. Para cada router  $r_i$  sabemos su localización  $l_i$  en el campus (supondremos que tenemos un sistema de coordenadas en el que podemos situar cada router) y el ancho de banda  $bw_i$  del tráfico directo que debe distribuir (el que no le llega de otros routers). También conocemos las coordenadas del equipo de comunicaciones ( $l_{EC}$ ). Cada router puede estar conectado a otro router o al

equipo de comunicaciones exterior directamente (la topología ha de ser un árbol). Disponemos de tres tipos de router (tipos A, B y C) capaces de distribuir hasta cierto ancho de banda máximo cada uno ( $bw_A < bw_B < bw_C$ ). Para distribuir el tráfico directo hay suficiente con un router de tipo A. El coste de cada router es proporcional al tipo (el tipo B cuesta el doble que el A y el C el doble que el B). Para conectar cada router  $r_i$  necesitamos instalar un cable  $c_i$  de cierta longitud (supondremos que es la distancia euclídea entre las coordenadas de los dos equipos conectados), el coste de este cable es proporcional a la longitud. El coste del equipo de comunicaciones exterior es fijo y no lo tendremos en cuenta.

El objetivo es decidir el tipo de los routers dependiendo del ancho de banda que deben soportar y la forma de interconectarlos, de manera que se que minimice el coste de la instalación.

En los siguientes apartados se proponen diferentes alternativas para algunos de los elementos necesarios para plantear la búsqueda (solución inicial, operadores, función heurística,...). Comenta muy brevemente la solución que se propone respecto a si es correcta, eficiente o mejor/peor respecto a otras alternativas posibles. Justifica tus respuestas.

- (a) Usar Hill-climbing. Como solución inicial conectamos directamente todos los routers al equipo de comunicaciones exterior. Como operadores usamos conectar un router a otro y desconectar un router de otro. La función heurística es el sumatorio, para todos los routers, del coste del router más el coste del cable que lo conecta (a otro router o al equipo de comunicaciones).
  - (b) Usar Hill-climbing. Como solución inicial conectamos cada router con el router más cercano y conectamos aleatoriamente uno de los routers al equipo exterior, asignamos a cada router el tipo A. Como operadores usamos cambiar la conexión de un router a otro router o equipo exterior por otra distinta si no superamos su capacidad y cambiar el tipo de un router. La función heurística quiere reducir al máximo la longitud de las conexiones y por ello es el sumatorio, para todos los routers, del coste del router más el coste del cable que lo conecta a otro router o al equipo de comunicaciones dividido por la longitud de esa conexión.
  - (c) Usar Algoritmos Genéticos. Supondremos que los routers y el equipo exterior están numerados consecutivamente, supondremos que para codificar ese número hacen falta  $n$  bits. La codificación es una tira  $r \cdot n$  de bits, donde  $r$  es el número de routers, cada grupo de  $n$  bits corresponden a un router siguiendo el orden de la numeración (es importante tener en cuenta que  $r + 1$  puede ser menor que  $2^n$ ). Como operadores usamos los operadores de cruce y mutación habituales. Para generar la población inicial usamos el método del apartado b) . La función heurística es la suma ponderada del coste de cada router (con peso 0,7) y el coste de su cable de conexión (con peso 0,3) para darle mayor relevancia al coste del router (que es lo más caro).
8. La International Telecommunications Union (ITU) es un agencia de Naciones Unidas que regula el uso de las tecnologías de la telecomunicación y entre otras cosas coopera en la asignación de órbitas para satélites. El incremento del uso de satélites de todo tipo ha llevado a buscar métodos automáticos para asignar posiciones en órbitas que mantengan la seguridad de los satélites y la calidad de su funcionamiento.

Para solucionar el problema, se ha dividido la órbita geosincrónica en regiones donde se asignarán las posiciones orbitales de los satélites. Obviamente, una región del espacio tiene un volumen. Se quiere resolver el problema para cada región.

Hay  $S$  diversos tipos de satélites especializados que se desean poner en órbita (televisión, telefonía, científicos, militares, ...). Para cada satélite se tienen dos informaciones, el espacio de seguridad (volumen necesario para que el satélite pueda navegar con el mínimo riesgo de colisión y pueda alimentar sus paneles solares) y el coste de alquiler de la órbita que cobra la ITU al dueño del satélite (este precio depende de cada satélite y no es una función del espacio de seguridad).

Las restricciones que tiene la ITU son que la suma del volumen que han de ocupar los espacios de seguridad de los satélites ha de cubrir como mínimo  $1/3$  del volumen de la región si se quieren cubrir las necesidades existentes, pero no puede superar los  $2/3$  si no se quieren tener problemas con otros lanzamientos que deben atravesar la órbita geosincrónica o con la basura espacial. También se ha

de cubrir un cupo mínimo  $m_i$  para cada tipo de satélites. El objetivo del problema es maximizar la ganancia que obtiene la ITU con el alquiler de órbitas.

En los siguientes apartados se proponen diferentes alternativas para algunos de los elementos necesarios para plantear la búsqueda (solución inicial, operadores, función heurística, ...). El objetivo es comentar la solución que se propone respecto a si es correcta, es eficiente, o es mejor o peor respecto a otras alternativas posibles. Justifica tu respuesta.

- (a) Se plantea usar Hill-climbing, como solución inicial consideramos que asignamos aleatoriamente al menos la mínima cantidad de satélites para cubrir la cuota de cada tipo hasta ocupar 1/3 del volumen de la región. Disponemos del operador `añadir-satélite()` que comprueba el posible exceso de ocupación en la región y del operador `quitar-satélite()` que comprueba la posible infra-ocupación de la región. Como heurística usamos la suma para todos los satélites en la solución del producto entre el espacio de seguridad del satélite y el coste de alquiler del satélite.
- (b) Se plantea usar Hill-climbing, como solución inicial ordenamos los satélites descendientemente por el coste del alquiler y vamos añadiendo los satélites en ese orden hasta llenar 2/3 del volumen. Como operador utilizamos cambiar un satélite de un tipo por otro del mismo tipo que no esté ya en la solución. Como función heurística usamos la suma del espacio de seguridad de todos los satélites en la solución multiplicado por la suma del coste del alquiler de todos los satélites.
- (c) Se plantea usar algoritmos genéticos. Un individuo es una tira de bits cuya longitud es el número total de satélites que tenemos. Si el bit está a 1 significa que el satélite está en la solución y si está a 0 es que no lo está. La población inicial se genera creando individuos que tengan a 1 los bits de los  $m_i$  satélites con mayor coste de alquiler de cada tipo. Como operadores se usan los habituales de cruce y mutación. Como función heurística usamos la suma del coste de alquiler de todos los satélites en la solución.

9. Tenemos una pequeña flota de  $C$  camiones que utilizamos para repartir mercancías y cada día tenemos que determinar qué ruta ha de seguir cada camión para abastecer un conjunto de ciudades por todo el país. El objetivo es que todos los camiones acaben la jornada aproximadamente a la misma hora, por lo que el número de kilómetros que ha de recorrer cada camión ha de ser muy parecido, y que recorran en total el mínimo número de kilómetros.

Disponemos de un mapa de carreteras que nos dice la distancia en kilómetros entre cada ciudad donde hemos de dejar nuestra mercancía, suponemos que todos los camiones parten de la misma ciudad y han de volver a ella al final del día, cargan al principio de la jornada todo lo que han de repartir, han de pasar sólo una vez por cada ciudad.

Una posible solución a éste problema se puede obtener mediante el uso de algoritmos de búsqueda local. En los siguientes apartados se proponen diferentes alternativas para algunos de los elementos necesarios para plantear la búsqueda (solución inicial, operadores, función heurística,...). Comenta muy brevemente la solución que se propone respecto a si es correcta y si es mejor/peor respecto a otras alternativas posibles. Justifica tus respuestas.

- (a) Usar Hill Climbing. Como solución inicial asignamos al azar a cada camión un número aproximadamente igual de ciudades, recorriéndolas en orden también al azar. Como operadores usamos el intercambiar dos ciudades entre los recorridos de dos camiones e intercambiar las posiciones de dos ciudades en el recorrido de un camión. La función heurística es la siguiente:

$$h(n) = \sum_{i=1}^C \left( \frac{LR_i}{\sum_{j=1}^C LR_j} - \frac{1}{C} \right)$$

donde  $LR_i$  es la longitud del recorrido del camión  $i$ .

- (b) Usar Hill Climbing. Como solución inicial asignamos todas las ciudades a un camión, estableciendo el recorrido inicial mediante una estrategia avariciosa de manera que intentemos minimizarlo. Como operadores usamos el mover una ciudad del recorrido de un camión a otro e intercambiar las posiciones de dos ciudades en el recorrido de un camión. La función heurística es la siguiente:

$$h(n) = \prod_{i=1}^C LR_i$$

- (c) Usar Hill climbing. Como solución inicial escogemos las  $C$  ciudades más cercanas a la ciudad origen como la primera ciudad a visitar por cada camión, como segunda ciudad en el recorrido de cada camión escogemos la más cercana a la primera que no esté ya asignada, y así sucesivamente hasta asignar todas las ciudades. Como operadores usamos el mover una ciudad del recorrido de un camión a otro, intercambiar las posiciones de dos ciudades en el recorrido de un camión e intercambiar dos ciudades entre los recorridos de dos camiones. La función heurística es la siguiente:

$$h(n) = \frac{C \cdot (C-1)}{2} - \sum_{i=1}^C \sum_{j=i+1}^C \frac{LR_i}{LR_j}$$

- (d) Usar algoritmos genéticos. Donde cada ciudad está representada por tantos bits como sean necesarios para codificar el valor  $C$ , la tira de bits contiene concatenados los bits de todas las ciudades, es decir, representamos en la tira de bits el número del camión que ha de recorrerla. Como operadores utilizamos los operadores habituales de cruce y mutación.

10. La sala de cine “Lo nunca visto” desea realizar una planificación estratégica de las películas a proyectar durante un año (52 semanas) siguiendo varios criterios comerciales. Cada película tiene asociada varias informaciones relevantes para este problema: tipo (infantil/adulto), previsión de beneficio semanal, índice de calidad. La programación anual debe contener al menos un 15% de películas infantiles y un 40% de películas para adultos. Se desea maximizar el beneficio anual de proyección de películas, pero sin dejar de lado la calidad. Por este motivo, el índice global de calidad no debe ser inferior a una cota predeterminada ( $Q$ ). En el caso de que una película deba proyectarse más de una semana, dichas semanas deberán ser consecutivas. Las estrategias comerciales exigen que una película no se proyecte más de 8 semanas.

Teniendo en cuenta el escenario descrito, comenta brevemente las diversas propuestas que se describen en los apartados siguientes. Valora si son correctas o no, si son eficientes o no, si son mejores o peores que otras alternativas. Justifica tus respuestas.

- (a) Usar un Hill-climbing. Como estado inicial asignamos aleatoriamente una película distinta a cada una de las 52 semanas. Como operador usamos Asignar-película (título, num-semana), que sustituye la película asignada una semana por otra nueva o por alguna de las ya asignadas a otras semanas.
- (b) Usar un Hill-climbing. Como estado inicial asignamos dos películas infantiles (4 semanas cada una) y seis películas para adultos (4 semanas cada una), el resto de las semanas se asigna aleatoriamente. Como operador usamos el mismo del apartado a.
- (c) Usar un Hill-climbing. Partimos del mismo estado inicial del apartado b con el control adicional de que el índice de calidad no sea inferior a  $Q$ . Al operador de asignación le añadimos como condiciones de aplicabilidad que mantenga las proporciones mínimas de películas infantiles y para adultos, que ninguna película supere las 8 semanas y que se respete la cota inferior  $Q$ . Como función heurística usaremos  $h1'(n)$ : la suma de la previsión de beneficio semanal para cada película proyectada multiplicada por el número de semanas asignado.
- (d) Usar un Hill-climbing. Como heurístico usaremos  $h2'(n)$ :  $h1'(n) +$  suma del índice de calidad de cada película proyectada multiplicado por el número de semanas asignado.
- (e) Usar algoritmos genéticos. Cada individuo se representa mediante 52 tiras de bits. La longitud de la tira de bits es suficiente para codificar el número de identificación de todas las películas. Como población inicial se generan  $n$  individuos con los criterios de estado inicial descritos en b y c. Como función de fitness usamos  $h3'(n)$ :  $h1'(n) +$  (suma del índice de calidad de cada película proyectada multiplicado por el número de semanas asignado -  $Q$ ). Como operadores usamos el

operador de cruce habitual, pero trabajando siempre con tiras completas de bits (la identificación de una película), es decir, un punto de cruce nunca parte el identificador de una película.

11. Los dueños del hotel “Bed, bugs and beyond” quieren reducir su presupuesto de limpieza para maximizar sus ganancias. Para hacerlo han decidido tener tres tipos diferentes de precios para sus habitaciones. El precio A incluye una limpieza diaria de la habitación y tiene un precio de  $pa$  euros por día, el precio B incluye una limpieza de la habitación en días alternos y tiene un precio de  $pb$  euros por día y el precio C solo incluye una limpieza el primer y último día y tiene un precio de  $pc$  por día. El mínimo número de días que se pueden reservar para una habitación es de dos, y asumiremos que la limpieza de las habitaciones de precio B esta organizada de manera que la última limpieza se hace el día en el que el cliente deja su habitación. Tenemos un total de  $R$  habitaciones en el hotel.

Limpiar una habitación de precio A cuesta  $ca$  euros por día de limpieza, una habitación de precio B cuesta  $cb$  por día de limpieza y una habitación de precio C cuesta  $ca$  más el número de días que la habitación ha sido reservada por  $cc$ . Las habitaciones que están vacías no se limpian.

El departamento de sanidad impone como restricción que toda habitación (ocupada o no) ha de limpiarse al menos 10 veces al mes.

El hotel recibe peticiones de reserva para un mes completo, cada reserva incluye el precio que el cliente quiere, el día en el que la reserva comienza y el número de días a reservar. Asumiremos que todas las peticiones comienzan y finalizan dentro del mes. El objetivo es decidir qué reservas aceptar, de manera que los gastos de limpieza se minimicen y las ganancias se maximicen.

En los siguientes apartados se proponen diferentes alternativas para algunos de los elementos necesarios para plantear la búsqueda (solución inicial, operadores, función heurística, ...). El objetivo es comentar la solución que se propone respecto a si es correcta, es eficiente, o es mejor o peor respecto a otras alternativas posibles. Justifica tu respuesta.

- (a) Queremos usar Hill-climbing, la solución inicial es la solución vacía. Como operadores de búsqueda usamos **añadir-reserva** que añade una reserva de una habitación, **cambiar-reserva** que cambia la reserva de una habitación por otra reserva que no esté ya en la solución. La función heurística es la suma de todos los costes de limpieza de las habitaciones multiplicado por las ganancias obtenidas por las reservas.
  - (b) Queremos usar Hill-climbing, la solución inicial se obtiene de la siguiente manera: para una habitación elegimos una de las reservas que tenga la fecha más temprana, la siguiente reserva será la que tenga la fecha más temprana que empiece después del final de la última reserva asignada a la habitación, repetimos este procedimiento hasta que no podamos asignar más reservas a la habitación. Hacemos esto para todas las habitaciones. Como operadores de búsqueda usamos **cambiar-reserva** que cambia una reserva de una habitación por otra reserva que no este en la solución. La función heurística es la suma de todos los costes de limpieza de las habitaciones menos las ganancias obtenidas por las reservas.
  - (c) Queremos usar algoritmos genéticos, decidimos asignar a cada reserva un número de 0 a  $R$ , la concatenación de estos números para todas las reservas en binario es la codificación de una solución. El número 0 significa que la reserva no esta asignada a ninguna habitación, otro número representa el número de habitación asignada a la reserva. Para generar la población inicial obtenemos una solución asignando aleatoriamente una reserva a cada habitación (solo  $R$  reservas tienen un número diferente de 0). Como operadores genéticos usamos los operadores habituales de cruce y mutación. La función heurística es la suma para todas las reservas en la solución del cociente entre las ganancias obtenidas por la reserva y los costes de limpieza de la reserva.
12. Una empresa de distribución de electricidad necesita calcular cada día qué centrales de producción ha de tener en marcha para abastecer la demanda de sus clientes. La empresa dispone de  $C$  centrales de producción eléctrica ubicadas en diferentes puntos de la geografía, con una capacidad de producción  $CA_i$  cada una. El coste diario de tener en marcha cada central es  $CO_i$ .
- El contrato que tiene la empresa con cada uno de sus  $CL$  clientes indica la cantidad de electricidad ( $E_i$ ) que se les debe suministrar cada día. La distancia entre los clientes y las centrales ( $d_{ij}$ ) supone

una pérdida en la eficiencia del suministro, esta pérdida se calcula como un factor fijo  $P$  multiplicado por esa distancia.

El objetivo es determinar el conjunto de centrales que se han de tener en marcha y a qué clientes han de servir, de manera que se minimice el coste de tenerlas en marcha y la pérdida debida a la distancia, teniendo en cuenta que la cantidad de electricidad que reciben los clientes de una central no ha de sobrepasar su capacidad de producción y que se ha de servir toda la demanda de los clientes. Asumiremos que los valores asignados al problema siempre permiten hallar una solución.

Una posible solución a este problema se puede obtener mediante el uso de algoritmos de búsqueda local. En los siguientes apartados se proponen diferentes alternativas para algunos de los elementos necesarios para plantear la búsqueda (solución inicial, operadores, función heurística,...). Comenta muy brevemente la solución que se propone respecto a si es correcta y si es mejor/peor respecto a otras alternativas posibles. Justifica tus respuestas.

- (a) Se plantea solucionarlo mediante Hill-climbing utilizando como solución inicial asignar a cada cliente al azar una central. Como operadores se utiliza el mover un cliente de una central a otra e intercambiar dos clientes entre dos centrales. La función heurística es la siguiente:

$$h(n) = \sum_{i=1}^C (CO_i \times marcha(i)) - \sum_{i=1}^C \sum_{j=1}^{CL} asig(i,j) \times d_{ij}$$

donde  $asig(i,j)$  es una función que retorna uno si el cliente  $i$  esta asignado a la central  $j$  y cero en otro caso y  $marcha(i)$  retorna uno si la central tiene clientes asignados y cero si no.

- (b) Se plantea solucionarlo mediante Hill-climbing utilizando como solución inicial un algoritmo avaricioso que ordena ascendentemente las centrales según su coste y los clientes según su demanda, y va asignando clientes a las centrales según ese orden. Cuando se sobrepasa la capacidad de una central se pasa a la siguiente. El operador utilizado es intercambiar dos clientes entre dos centrales siempre que no se supere la capacidad de suministro de alguna de ellas. La función heurística es:

$$h(n) = \left( \sum_{i=1}^C CO_i \right) \times \left( \sum_{i=1}^C \sum_{j=1}^{CL} P \times asig(i,j) \times d_{i,j} \right)$$

- (c) Se plantea utilizar algoritmos genéticos donde la representación de la solución es una tira de bits. Esta está compuesta por la concatenación de  $C$  grupos de  $CL$  bits, donde cada grupo representa la asignación o no de cada cliente a la central como un 1 o un 0. Las población inicial la obtenemos mediante el mismo mecanismo que en el apartado anterior. Se utilizan los operadores habituales de cruce y mutación. La función heurística es:

$$h(n) = \left( \sum_{i=1}^C CO_i \times marcha(i) \right) + \left( P \times \sum_{i=1}^C \sum_{j=1}^{CL} asig(i,j) \times d_{i,j} \right)$$

13. Después de los incendios ocurridos en el verano se nos plantea el problema de buscar una forma efectiva de repoblar las áreas afectadas. Dada un área concreta de cierto número de hectáreas a repoblar, dividimos el área en una cuadrícula de  $N \times M$  áreas y nos planteamos cuanto plantar en cada área.

Cada área  $i$  de la cuadrícula tiene asociado un factor de repoblación ( $Fr(i)$ ) que va de 0 a 3 (0 significa ninguna repoblación, 1 significa plantar  $A$  árboles, 3 significa plantar  $3 \times A$  árboles). Disponemos un máximo de  $K \times A$  árboles para plantar, pero no queremos plantar menos de  $I \times A$  árboles.

Además queremos minimizar el riesgo de incendio del área. Este es una función del factor de repoblación de un elemento de la cuadrícula y todos los elementos adyacentes, y se calcula de la siguiente manera:



$$R(i) = \sum_{\forall j \text{ adyacente a } i} Fr(i) \times Fr(j)$$

El objetivo es encontrar una solución que plante el máximo número de árboles (entre los límites  $I \times A$  y  $K \times A$ ) y con el mínimo riesgo de incendio.

Una posible solución a éste problema se puede obtener mediante el uso de algoritmos de búsqueda local. En los siguientes apartados se proponen diferentes alternativas para algunos de los elementos necesarios para plantear la búsqueda (solución inicial, operadores, función heurística,...). Comenta muy brevemente la solución que se propone respecto a si es correcta y si es mejor/peor respecto a otras alternativas posibles. Justifica tus respuestas.

- (a) Usar Hill Climbing. Para ello generamos una solución inicial en la que todas las áreas tienen factor de repoblación 0. Los operadores son aumentar o disminuir el factor de repoblación de un área. La función heurística que queremos optimizar es:

$$h'(n) = \sum_{i=1}^{M \times N} R(i) - \sum_{i=1}^{M \times N} Fr(i)$$

- (b) Usar Hill Climbing. Para ello generamos una solución inicial en la que asignamos secuencialmente factor de repoblación +1 a cada elemento de la cuadrícula hasta llegar a la cantidad mínima de árboles que hemos de plantar. Los operadores son aumentar el factor de repoblación siempre que este no sea 3 o disminuir el factor de repoblación de un área siempre que no sea 0. La función heurística que queremos optimizar es:

$$h'(n) = \left( \sum_{i=1}^{M \times N} Fr(i) - K \right) - \frac{1}{\sum_{i=1}^{M \times N} R(i)}$$

- (c) Usar algoritmos genéticos. Para representar una solución generamos una tira de  $2 \times M \times N$  bits (con 2 bits codificamos el factor de repoblación). Como solución inicial utilizamos la misma que en el apartado anterior. Como operadores utilizamos los operadores habituales de cruce y mutación. La función heurística que queremos optimizar es:

$$h'(n) = \sum_{i=1}^{M \times N} R(i) \times \sum_{i=1}^{M \times N} Fr(i)$$

14. Una de las labores de los equipos de comunicaciones en una red es distribuir el flujo de paquetes que le llegan a un equipo entre los diferentes equipos con los que está conectado, optimizando la calidad de la distribución de paquetes de forma local.

Una posible estrategia de distribución de paquetes sería decidir a priori cuantos de los paquetes que se reciben en cierto segundo se envían por cada conexión, sin preocuparnos exactamente de a donde deben ir.

Un equipo de comunicaciones recibe cierto número de paquetes por segundo ( $P$ ) que tiene que distribuir entre sus conexiones de salida. Para cada una de las  $N$  conexiones de salida del equipo se conocen tres informaciones: su capacidad (en número de paquetes por segundo), el tiempo de retraso que introduce la conexión (tiempo medio adicional que añade el nodo de salida al tiempo de llegada a destino de cada paquete) y el número medio de saltos que hará cada paquete hasta llegar a su destino.

Deseamos calcular el número de paquetes que debemos enviar por cada conexión de salida para optimizar la distribución de paquetes de manera que el retraso y número medio de saltos de los paquetes sea el mínimo posible. Evidentemente se han de enviar todos los paquetes que llegan.

En los siguientes apartados se proponen diferentes alternativas para algunos de los elementos necesarios para plantear la búsqueda (solución inicial, operadores, función heurística, ...). El objetivo es comentar la solución que se propone respecto a si es correcta, es eficiente, o es mejor o peor respecto a otras alternativas posibles. Justifica tu respuesta.

- (a) Se plantea solucionarlo mediante Hill-Climbing usando como solución inicial el repartir todos los paquetes a partes iguales entre todas las conexiones de salida del equipo. Como operador utilizamos mover cierta cantidad de paquetes de una conexión a otra. Como función heurística usamos el sumatorio, para todas las conexiones con paquetes asignados, del producto entre el retraso de la conexión y el número de saltos.
- (b) Se plantea solucionarlo mediante Hill-Climbing tomando como solución inicial el repartir los paquetes entre las conexiones de salida del equipo asignando aleatoriamente a una conexión el máximo de su capacidad, repitiendo el proceso hasta haber repartido todos los paquetes. Como operador se utilizaría mover cualquiera de los paquetes de una conexión a otra siempre que la operación sea válida. Como función heurística usamos el sumatorio para todas las conexiones con paquetes asignados del producto entre el número de paquetes asignados a la conexión y el retraso de la conexión más el sumatorio, para todas las conexiones con paquetes asignados, del producto entre el número de paquetes asignados a la conexión y el número de saltos medio de esa conexión.
- (c) Se plantea solucionarlo mediante Algoritmos Genéticos, donde la representación del problema es una tira de bits compuesta por la concatenación de la representación en binario del número de paquetes asignados a cada conexión (evidentemente usando el mismo número de bits para cada conexión). Como solución inicial ordenamos las conexiones por su retraso (de menor a mayor) y asignamos paquetes en ese orden llenando la capacidad de cada conexión hasta haber asignado todos los paquetes. Usamos los operadores de cruce y mutación habituales. Usamos como función heurística el sumatorio, para todas las conexiones, del producto entre el número de paquetes asignados a cada conexión y el retraso de la conexión.

15. La empresa *Cloud For U* se dedica a revender tiempo de cómputo para cloud computing y para ello ha montado un servicio que recibe solicitudes para usar ese tiempo. Este servicio es un sistema de pre-reserva que nos permite saber las solicitudes para un mes completo. Cada petición nos indica un día y hora de inicio y un tiempo de uso en horas, indicando además la prioridad de la petición (de 1 a 3, siendo 1 la mínima prioridad). Una petición ha de ejecutarse completamente, sin interrupciones y usando exclusivamente una CPU. Nosotros compramos el tiempo de cómputo en una CPU durante un mes a  $M$  y cobramos el tiempo de cómputo a  $prioridad \times P$  por hora, con el compromiso de que si no servimos la petición, ésta se servirá seguro el mes siguiente a solo  $P$  por hora independientemente de su prioridad inicial. Nos planteamos saber cuantas CPUs debemos alquilar nosotros para poder obtener el mayor beneficio posible.

En los siguientes apartados se proponen diferentes alternativas para algunos de los elementos necesarios para plantear la búsqueda (solución inicial, operadores, función heurística,...). Comentar cada apartado indicando si los elementos de la propuesta son correctos, eficientes en coste, mejor/peor que otras alternativas propuestas o si hay alternativas mejores. Justifica tus respuestas.

- (a) Se plantea aplicar Hill-climbing, para crear la solución inicial suponemos que necesitamos tantas CPUs como peticiones nos han llegado, de manera que asignamos cada petición a una CPU diferente. Como operador usamos fusionar dos CPUs, que consiste en combinar las peticiones de dos CPUs en una sola eliminando la CPU sobrante. Como función heurística consideraremos que alquilaremos las CPUs en la solución que tengan más del 90% de ocupación ( $CPU_{90}$ ), por lo tanto minimizaremos la función:

$$h(n) = (N_{CPU} \times M) - \sum_{\forall p_i \in CPU_{90}} prio(p_i) \times P \times long(p_i) + \sum_{\forall p_i \in \overline{CPU_{90}}} 3 \times P \times long(p_i)$$

donde  $N_{CPU}$  es el número de CPUs usadas en la solución,  $CPU90$  es el conjunto de CPUs ocupadas a más del 90% y  $\overline{CPU90}$  es su conjunto complementario.  $long(p_i)$  es la longitud en horas de la petición.

- (b) Se plantea usar algoritmos genéticos, como representación escogemos un orden arbitrario y representamos cada petición con un número en binario que es la CPU que tiene asignada, suponemos que ese número es un valor entre el número de peticiones de prioridad 3 y el numero de peticiones total, añadimos un valor adicional para representar una CPU ficticia. Para generar una solución se van generando al azar valores entre los límites mencionados y se asignan consecutivamente a las peticiones de la siguiente manera, con un 50% de probabilidad se le asigna el valor de una CPU real y con un 50% de probabilidad se le asigna la CPU ficticia. Utilizamos los operadores habituales de cruce y mutación. Como función heurística consideraremos que alquilaremos las CPUs en la solución que no sean la ficticia, por lo tanto minimizaremos la función:

$$h(n) = (N_{CPU} \times M) - \sum_{\forall p_i \notin CPU_f} prio(p_i) \times P \times long(p_i) + \sum_{\forall p_i \in CPU_f} prio(p_i) \times P \times long(p_i)$$

donde  $N_{CPU}$  es el número de CPUs que no son la ficticia,  $CPU_f$  es el conjunto de peticiones que están en la CPU ficticia.

16. Dado el interés creciente que despierta el juego del GO, la junta directiva de la Asociación de GO necesita establecer un calendario para la final del campeonato anual de GO, que este año tendrá muchos más participantes. En la final del campeonato hay  $C$  diferentes categorías dependiendo de la edad y del nivel de los jugadores (30 Kyu .. 1 Kyu, 1 dan .. 9 dan). En cada categoría participan un número  $J_c$  de jugadores, lo cual significa un número  $R_c$  de rondas eliminatorias (obviamente  $R_c = \log_2(J_c)$ ). La final ha de durar un máximo de 15 días, para cada día de la final se disponen de  $F$  franjas horarias, en una franja solo se puede jugar una partida de GO. Obviamente se han de jugar todas las partidas de la final del campeonato.

El calendario ha de cumplir un conjunto de restricciones:

- El número de días necesarios para realizar la final del campeonato ha de ser el mínimo posible.
- El conjunto de rondas correspondientes a cada categoría se ha de celebrar en el mínimo número de días posible.
- En un día se pueden jugar como máximo  $P$  partidas de una categoría.
- Entre dos partidas de una misma categoría en un mismo día debe haber como mínimo dos franjas horarias de separación y como máximo cinco.
- Debe haber como mínimo un día de separación entre dos rondas de una categoría.

En los siguientes apartados se proponen diferentes alternativas para algunos de los elementos necesarios para plantear la búsqueda (solución inicial, operadores, función heurística,...). Comenta muy brevemente la solución que se propone respecto a si es correcta, eficiente o mejor/peor respecto a otras alternativas posibles. Justifica tus respuestas.

- (a) Usar Hill-climbing. Como solución inicial se utiliza el calendario vacío. Como operadores usamos asignar y desasignar una partida de una ronda de una categoría a una franja de un día. Como función heurística usamos el número total de días en los que hay partidas asignadas más la suma del número de días que ocupan las rondas de una categoría.
- (b) Usar Hill-climbing. Como solución inicial se usa el calendario resultante de añadir de manera secuencial las partidas de cada ronda de la primera categoría manteniendo una distancia de cinco franjas horarias entre cada partida, sin superar las  $P$  partidas diarias y manteniendo un día de separación entre cada ronda. Con el mismo procedimiento se van introduciendo las partidas del resto de categorías. Como operadores utilizamos mover una partida a una franja no ocupada e intercambiar dos partidas. Como función heurística usamos el número de días total en los que hay

partidas asignadas, más la suma del número de días que ocupan las rondas de una categoría, más una constante por el número de franjas que separan en un día las partidas de una misma ronda y categoría.

- (c) Usar Algoritmos Genéticos. Asignamos a cada partida a jugar un código binario con la longitud necesaria para representarlas todas, añadimos a la codificación un código con todos los bits a cero. Representamos un calendario como una cadena de bits de longitud igual al producto de  $F$  franjas por 15 días por el número de bits necesarios para representar las partidas. Si una franja no tiene una partida asignada contendrá la codificación a cero, si no tendrá el código de una partida.

Para generar la población inicial usamos el método del segundo apartado cambiando el orden en el que escogemos las categorías par generar variabilidad. Como operadores genéticos usamos solamente el de cruce por un punto, donde el punto de cruce no puede estar en medio de la codificación de una partida. Como función heurística minimizamos el número de franjas vacías entre dos partidas consecutivas.

17. Se ha de celebrar un concurso nacional de canto coral que reúne a varias corales no profesionales. Con el objetivo de reducir el desembolso que han de realizar los miembros de las corales, la organización ha solicitado y ha obtenido el patrocinio de una empresa para sufragar los costes de alojamiento de los participantes. Para llevar mejor el control del gasto en alojamiento, los organizadores del evento han de planificar las reservas de alojamiento de los 950 asistentes en los hostales de la zona. Disponen de una lista de hostales y para cada uno de ellos saben:

- A qué distancia se encuentra del lugar de la celebración del concurso,
- Cuantas camas tiene disponibles.
- El precio de la cama por noche.

Su problema es encontrar una asignación de las 950 personas a hostales de modo que no se supere el tope de camas disponibles en cada hostel, se minimice la suma de las distancias recorridas por todos los asistentes y se minimice la cantidad de dinero que la organización va a dedicar al pago del alojamiento. Se puede suponer que sólo hay que planificar una noche, que en cada hostel todas las camas cuestan lo mismo y que solo hemos de asignar personas a hostales (la asignación de personas a camas dentro del hostel la gestionará el propio hostel con los clientes por orden de llegada).

Una posible solución a éste problema se puede obtener mediante el uso de algoritmos de búsqueda local. En los siguientes apartados se proponen diferentes alternativas para algunos de los elementos necesarios para plantear la búsqueda (solución inicial, operadores, función heurística,...). Comenta muy brevemente la solución que se propone respecto a si es correcta y si es mejor/peor respecto a otras alternativas posibles. Justifica tus respuestas.

- (a) Usar Hill Climbing. Como solución inicial se distribuyen aleatoriamente todas las personas entre los hostales. Como operador se utiliza mover un grupo de personas de un hostel a otro, moviendo tantas como quepan en el hostel destino. La función heurística es la suma de las distancias recorridas por todas las personas multiplicada por una constante  $D$  más la suma de los precios de las camas multiplicada por una constante  $P$ .
- (b) Usar Hill climbing. Como solución inicial se utiliza una estrategia avariciosa poniendo a las personas en los hostales con el cociente precio/distancia más bajo. Como operador se utiliza mover un grupo de personas de un hostel a otro, moviendo tantas como quepan en el hostel destino. La función heurística es la suma, para todas las personas, del producto entre la distancia recorrida por una persona y el precio de su cama.
- (c) Usar algoritmos genéticos. Representaremos la solución como una tira de bits donde asignamos a cada hostel tantos bits como se necesiten para representar el número máximo de personas que caben. Utilizamos los operadores habituales de cruce y mutación. La función heurística es la suma de las distancias de todos los hostales que tienen alguna persona asignada, dividida por la suma de los precios de las camas de todas las personas.

18. Tras numerosas críticas por la lentitud de la justicia en España, el Ministerio de Justicia quiere mejorar la eficiencia del sistema con un nuevo procedimiento de asignación de plazas de funcionarios a los juzgados. Cada funcionario  $f_i$  tiene un rol (juez, fiscal, abogado defensor, administrativo...), acepta un cierto sueldo mínimo ( $Pf_i$ ) y acepta una distancia máxima a recorrer entre su casa y su trabajo ( $Df_i$ ). Cada juzgado  $j_k$  dispone de  $x_r$  plazas nuevas por cada rol  $r$ . El ministerio paga cada plaza del rol  $r$  a un precio máximo determinado ( $P_r^k$ ) que es diferente para cada juzgado  $j_k$ , y dispone de una tabla de distancias entre las viviendas de los funcionarios y los juzgados.

Se quiere asignar funcionarios a juzgados de forma que se maximice el ahorro económico respecto a lo que inicialmente está dispuesto a pagar el ministerio, se minimicen las distancias que los funcionarios deberán recorrer a su trabajo, y se cubra el máximo número de plazas nuevas, priorizando este último factor respecto a los dos anteriores.

En los siguientes apartados se proponen diferentes alternativas para algunos de los elementos necesarios para plantear la búsqueda (solución inicial, operadores, función heurística). Hay que comentar la solución que se propone respecto a si es correcta, es eficiente, y es mejor o peor en comparación con otras alternativas. Justifica todas las respuestas.

- (a) Usar Hill-climbing a partir del estado en el que no se ha asignado ninguna plaza. Utilizar los operadores de asignar/desasignar funcionario a plaza si los roles son iguales y el sueldo que piden es menor que el precio que se quiere pagar ( $P_r^k$ ). Usar como función heurística la suma de sueldos mínimos que aceptan los funcionarios asignados, más la suma de distancias que deberán recorrer dichos funcionarios para ir a trabajar, todo ello dividido por el número de plazas asignadas.
- (b) Usar algoritmos genéticos. Cada estado se representa con una cadena de bits de longitud equivalente al producto de número de plazas por el número de funcionarios. Un bit representa la asignación o no de un funcionario a una plaza. Los operadores son los típicos de cruce y mutación. La función de fitness es:

$$h(n) = \frac{\sum_{\forall i \in FAsig} Pf_i}{\sum_{\forall i \in F} Pf_i} + \frac{\sum_{\forall i \in FAsig} Df_i}{MaxD * \#F}$$

Donde  $F$  es el conjunto de todos los funcionarios,  $FAsig$  es el conjunto los funcionarios asignados a una plaza,  $\#F$  es la cardinalidad de  $F$  y  $MaxD$  es la distancia mayor entre la vivienda de un funcionario y un juzgado.

19. Tras el proceso de reestructuración bancaria el potente Banco Nacional de Crédito (BNC) ha comprado la Caja de Ahorros Popular (CAP) con todas sus oficinas. El problema es que, tras la fusión de ambas entidades, hay un exceso de oficinas bancarias en una de las ciudades donde ambas entidades operaban antes de la fusión. Nos piden un sistema inteligente que les ayude a decidir que oficinas cerrar. Tenemos un mapa de la ciudad con las  $O$  oficinas disponibles tras la fusión. Para cada oficina  $o_i$  podemos obtener el coste anual de su alquiler ( $alquiler(o_i)$ ). Tenemos también un listado de todos los  $C$  clientes que tenemos tras la fusión del BNC y la CAP. Para cada cliente  $c_j$  tenemos su dirección (geo-localizada en el mapa) y podemos obtener el importe total de los depósitos que tiene en la entidad ( $depositos(c_j)$ ).

El banco le da mucha importancia a tener oficinas cerca de sus clientes. Por sus estudios de mercado han visto que sus mejores clientes suelen estar en un radio de 600 metros de sus oficinas. Nos dan una función `Oficina oficina_cercana(Cliente c)` que dado un cliente  $c$  usa el mapa de la ciudad para devolvernos la oficina o más cercana, o devuelve `null` si no hay ninguna oficina a menos de 600 metros del cliente. También tienen una función `float beneficio_deposito(Cliente c)` que dado un cliente  $c$  hace una previsión de los beneficios que obtendrá el banco durante un año operando en los mercados financieros con el dinero de los depósitos de ese cliente. A partir de estas funciones definen dos más: la primera es `float estim_benef_depositos(Oficina o)` que calcula la suma de beneficios obtenidos por los depósitos de todos los clientes cuya oficina más cercana es  $o$ ; la segunda es `LClientes clientes_sin_oficina()` que nos devuelve la lista de clientes que no tienen ninguna oficina a menos de 600 metros de donde viven. Todas estas funciones se pueden invocar varias veces durante la ejecución y nos darán diferentes resultados dependiendo de las oficinas que se quieran cerrar y de cual quede más cerca de cada cliente en cada momento.

El objetivo es reducir el número de oficinas que se mantienen abiertas tras la fusión de ambas entidades, minimizando el coste total anual en alquileres de las oficinas que se mantienen abiertas y maximizando el beneficio total obtenido a partir de los depósitos de los clientes que tienen una oficina a menos de 600 metros de su vivienda. La idea es cerrar oficinas que no salen a cuenta mantener porque no tienen suficientes clientes para cubrir gastos con los beneficios obtenidos, o porque hay otra oficina en la cercanía que puede atender a esos clientes. Aunque no gusta perder clientes, es posible dejar algunos clientes sin oficina cercana si mantener dicha oficina genera más costes que beneficios.

En los siguientes apartados se proponen diferentes alternativas para algunos de los elementos necesarios para plantear la búsqueda (solución inicial, operadores, función heurística, ...). El objetivo es comentar la solución que se propone respecto a si es correcta, es eficiente, o es mejor o peor respecto a otras alternativas posibles. Justifica tu respuesta.

- (a) Se plantea solucionarlo mediante Hill-Climbing. Como solución inicial se parte del conjunto completo de todas las  $O$  oficinas disponibles y de los  $C$  clientes. Como operador único se plantea el eliminar una oficina de la solución. Como función heurística se usa

$$h_1 = \sum_{\forall o_i \in OS} estim\_benef\_depositos(o_i) - alquiler(o_i)$$

siendo  $OS$  el conjunto de oficinas supervivientes (no eliminadas).

- (b) Se plantea solucionarlo mediante  $A^*$ . Para ello se parte del estado en el que no hay ninguna oficina en la ciudad. Tenemos dos operadores: colocar oficina, que añade una oficina  $o_i \in O$ , su coste es  $estim\_benef\_depositos(o_i) - alquiler(o_i)$ ; eliminar oficina, que elimina una oficina  $o_i$  que hubiera sido colocada con el operador anterior, su coste es  $-(estim\_benef\_depositos(o_i) - alquiler(o_i))$ . Como heurístico  $h$  se usa:

$$h_2 = \sum_{\forall c_j \in CN} beneficio\_deposito(c_j)$$

siendo  $CN$  el conjunto de clientes que quedan por servir (los que no tienen ninguna oficina cercana), que se obtiene con la función `clientes_sin_oficina()`.

- (c) Se plantea resolverlo mediante algoritmos genéticos. Para representar el problema utilizamos una tira de  $O$  bits, donde un bit a 1 indica que esa oficina se va a mantener abierta y un bit a 0 indica que esa oficina se va a cerrar. Como población inicial generamos aleatoriamente  $n$  individuos donde en cada uno hay  $O/2$  bits a 1. Como operadores usamos un operador de cruce en un punto y un operador de mutación que cambia el valor de un bit de la cadena (de 0 a 1 o de 1 a 0). La función heurística es:

$$h_3 = \sum_{\forall o_i \in OS} \frac{estim\_benef\_depositos(o_i)}{alquiler(o_i)}$$

siendo  $OS$  el conjunto de oficinas supervivientes (no eliminadas), es decir, las que tienen el bit a 1.

20. Como plan de emergencia ante una posible impugnación del proceso del 9-N nos piden crear un programa que, de forma automática, asigne nuevas localizaciones para las urnas en locales que no dependen de la Generalitat. Para ello nos proporcionan un mapa del territorio con  $L$  posibles locales alternativos donde podemos colocar un total de  $C$  colegios electorales ( $L > C$ ). Tenemos una tabla con las distancias entre los locales. Además también disponemos del número de voluntarios disponibles en cada local en el que se puede ubicar un colegio electoral.

El objetivo es colocar todos los colegios electorales de manera que se maximice la suma de las distancias entre ellos (para obtener así la cobertura máxima del territorio) y se maximice también el número de voluntarios disponibles que se movilizan para ayudar en las votaciones.

En los siguientes apartados se proponen diferentes alternativas para algunos de los elementos necesarios para plantear la búsqueda (solución inicial, operadores, función heurística, ...). El objetivo es comentar la solución que se propone respecto a si es correcta, es eficiente, o es mejor o peor respecto a otras alternativas posibles. Justifica tu respuesta.

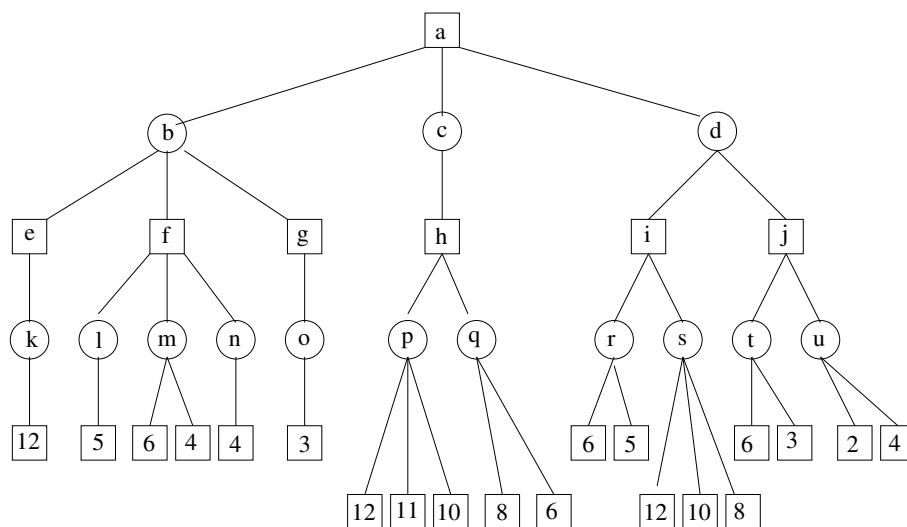
- (a) Se plantea solucionarlo mediante Hill-Climbing. Como solución inicial se parte de la solución vacía. Como operadores se usa el colocar un colegio electoral en un local vacío y quitar un colegio electoral de un local. Como función heurística se usa la suma de las distancias de los locales de cada uno de los  $C$  colegios electorales al resto, multiplicado por la suma de los voluntarios disponibles en cada local asignado.
- (b) Se plantea solucionarlo mediante Hill-Climbing utilizando como solución inicial el colocar aleatoriamente los  $C$  colegios electorales y usando como operadores de búsqueda mover un colegio electoral al local vacío más cercano respecto a su posición actual. Como función heurística se usa la suma de las distancias de los locales de cada uno de los  $C$  colegios electorales al resto más la suma de los voluntarios disponibles en cada local asignado.
- (c) Se plantea resolverlo mediante algoritmos genéticos. Para representar el problema utilizamos una tira de  $L$  bits, donde un bit a 0 indica que ese local no tiene colegio electoral asignado y un bit a 1 indica que si se le ha asignado un colegio electoral. Como población inicial generamos aleatoriamente  $n$  individuos donde en cada uno hay exactamente  $C$  bits a 1. La función heurística es la suma de distancias de cada colegio electoral al resto más una constante  $k$  por la suma de los voluntarios disponibles de cada local asignado. Como operadores usamos un operador de cruce que intercambia aleatoriamente la mitad de los bits de cada individuo de la pareja cruzada y un operador de mutación que intercambia aleatoriamente dos posiciones de la tira de bits cumpliendo que en una sea un bit a 1 y en la otra un bit a 0.





## 4. Jocs

1. La figura següent mostra l'arbre d'un determinat joc, generat fins a una profunditat màxima de 4. Els números que acompanyen als nodes del nivell 4 indiquen el valor de la funció d'avaluació estàtica per cadascun d'ells.



Contesta les següents qüestions:

- (a) Seguint el mètode del Minimax, quin seria el valor assignat a cada node de l'arbre?
- (b) Aplica el mètode Minimax amb poda Alfa-Beta, usant els valors donats a la figura. Mostra el procés seguit, indicant quines són les branques podades i els valors assignats progressivament als nodes (tant els provisionals com el final).
2. Donat aquest arbre on els nodes que mostren un número corresponen a l'avaluació estàtica de l'estat corresponent



5. Donada la següent posició del joc de dames en un tauler de 4 per 4 on nosaltres juguem amb les peces rodones i toca jugar al contrari.

	3		4
	1		
		2	

Indica quina seria la valoració que min-max donaria a la jugada del contrari consistent en avançar la peça 3 cap a l'esquerra (en relació a la nostra posició davant el tauler), tenint en compte que:

- la valoració de cada posició es fa sumant peces pròpies i restant peces contràries, en base als següents valors individuals de les peces: 1 peça = 1 punt, 1 dama = 5 punts
- només cal cercar fins a una fondària de 4 jugades (a comptar des de la posició del dibuix). Cal que:
- utilitzis l'algorisme min-max amb poda alfa-beta
- generis els successors d'un estat seleccionant :
- primer les peces etiquetades amb un nombre més baix
- primer el moviment cap a l'esquerra (en relació a la nostra posició davant el tauler).
- dibuixis l'arbre generat indicant, per cada node, l'ordre de visita, el valor min-max calculat, l'evolució dels valors de alfa i beta i els valors propagats a d'altres nodes.

Nota per a aquells que NO saben jugar a dames:

- Moviment: Una peça es pot moure a qualsevol de les caselles adjacents, diagonals del davant (en relació a la posició pròpia davant del tauler).
  - Una peça pot matar una peça contrària si aquesta està al davant i pot saltar pel damunt en diagonal (o sigui la posició on va a parar està lliure).
  - Coronació: Una peça esdevé dama si es situa en una casella de la darrera fila (en relació a la posició pròpia davant del tauler).
  - Moviment de dama: Una dama es pot desplaçar en qualsevol sentit de qualsevol diagonal i en un nombre no restringit de caselles.
6. Supongamos la siguiente posición en un juego de damas en un tablero 4 x 4. Las piezas propias son 1 y 2, las del contrario son 3 y 4. Acabo de jugar y la posición es:

4			
	3		
		2	
			1

Valorar la posición utilizando min-max con profundidad de búsqueda 4 (dos jugadas del contrario y dos propias) a partir de la posición del dibujo. La valoración de una posición se obtiene sumando los puntos de las piezas propias y restando las del adversario. Cada pieza vale 1 punto. Al coronar las piezas se convierten en damas y valen 5 puntos.

El orden de generación de los sucesores es:

- la pieza 1 (3) tiene prioridad sobre la 2 (4)
- Si una pieza tiene más de un movimiento, tiene prioridad el de menor abscisa.

NO se aplicará la regla de la "bufada".

Se pide:

- Utilizar el algoritmo min-max SIN poda dibujando el árbol e indicando para cada nodo el orden de visita y el valor de la posición.
  - Hacer lo mismo con el algoritmo min-max CON poda alfa-beta indicando además la evolución de los valores alfa/beta.
  - Comparar los resultados, ¿ha habido ahorro en el número de nodos visitados? Nota per a aquells que NO saben jugar a dames veure regles al problema 5.
7. Supongamos la siguiente posición en un juego de damas en un tablero 4 x 4. Las piezas propias son 1 y 2, las del contrario son 3 y 4. Acabo de jugar y la posición es:

4			
	3		
		1	
	2		

Valorar la posición utilizando min-max con profundidad de búsqueda 4 (dos jugadas del contrario y dos propias) a partir de la posición del dibujo. La valoración de una posición se obtiene sumando los puntos de las piezas propias y restando las del adversario. Cada pieza vale 1 punto. Al coronar las piezas se convierten en damas y valen 5 puntos.

El orden de generación de los sucesores es:

- la pieza que pueda comer. Si hubiera más de una, el orden sería (1, 2) o (3, 4)
- la pieza 1 (3) tiene prioridad sobre la 2 (4)
- Si una pieza tiene más de un movimiento, tiene prioridad el de menor abscisa.

NO se aplicará la regla de la "bufada".

Se pide:

- Aplicar el algoritmo min-max SIN poda, dibujando el árbol e indicando para cada nodo el orden de visita y el valor de la posición.
  - Hacer lo mismo con el algoritmo min-max CON poda alfa-beta indicando además la evolución de los valores alfa/beta.
  - Comparar los resultados, ¿ha habido ahorro en el número de nodos visitados? Nota per a aquells que NO saben jugar a dames veure regles al problema 5.
8. Tenemos el tablero que aparece en la figura, en la que el jugador MAX tiene las fichas marcadas como X y el min las fichas marcadas como O. La única ficha que pueden mover los jugadores es la marcada como +. Esta ficha se puede mover hacia la derecha y hacia la izquierda y tiene el efecto de cambiarse con la ficha que ocupa la posición a la que se desplaza y cambiar el signo de esa ficha y de la que pasa a ser su contigua. Por ejemplo, en la configuración XOXO+XOXO, si desplazamos la ficha + hacia la derecha obtenemos la configuración XOXOO+XXO. El objetivo de cada jugador es tener 5 fichas del mismo tipo seguidas.

X	O	X	O	+	X	O	X	O
---	---	---	---	---	---	---	---	---

Para la evaluación de las configuraciones utilizaremos la siguiente función:

$f'(n) = \text{Suma de los tamaños de los grupos de X mayores que 1} - \text{Suma de los tamaños de los grupos de O mayores que 1}.$

Consideraremos que si la ficha + está entre dos fichas iguales, éstas no forman un grupo. Por ejemplo la evaluación de la configuración XOXOO+XXO sería  $(2 - 2) = 0$ , la evaluación de la configuración XOOOXO+OO sería  $(0 - 3 - 2) = -5$ .

- (a) Utiliza el algoritmo minimax con poda alfa-beta para evaluar el primer movimiento que debería hacer el jugador MAX, haz la exploración hasta el nivel 4 (dos jugadas de MAX y dos de MIN). Aplica siempre el mismo orden: primero el movimiento hacia la izquierda y después el movimiento hacia la derecha. Para cada nodo muestra claramente la evolución de los valores alfa y beta. ¿Cuál es el movimiento que debería escoger MAX?
- (b) ¿Cuántos nodos nos ahorramos respecto al uso del algoritmo sin poda?
9. El juego del Othelo se juega en un tablero de NxN como el de la figura y consiste en cubrir todo el tablero con fichas, colocándolas alternativamente, de manera que gane el que más fichas de su color consiga. Las reglas del juego son las siguientes: Para colocar una ficha hay que tener en esa fila, columna o diagonal otra ficha propia que encierre fichas contrarias con la que colocamos. Al colocar la ficha, todas las fichas de color contrario que queden atrapadas entre dos fichas propias cambian su color al nuestro.

El tablero comienza con las fichas que aparecen en la figura.

	N	B	
	B	N	
		B	N

- (a) Utiliza el algoritmo de minimax con poda alfa-beta para averiguar qué movimiento deberían hacer las blancas desde la posición inicial. Explora hasta profundidad 2 (un movimiento de las blancas y otro de las negras). Para la evaluación de los estados utiliza la siguiente tabla para asignar valor a cada ficha, el valor de un estado se obtiene sumando los valores de las fichas propias y restando las del contrario.

4	2	2	4
2	1	1	2
2	1	1	2
4	2	2	4

#### Orden de generación de sucesores:

Para expandir los nodos supón que el tablero está numerado desde la esquina superior izquierda siguiendo el orden por filas. Expande los nodos siguiendo el orden de esta numeración.

- (b) Utiliza el algoritmo MiniMax con poda alfa-beta para averiguar qué movimiento deberían hacer las negras desde la posición escogida por las blancas en el apartado anterior (en caso de empate escoge el primer movimiento que desarrollaste). Explora hasta profundidad 2 (un movimiento de las negras y otro de las blancas). Para la evaluación de las posiciones utiliza la tabla anterior.
10. Tenemos un tablero de 2x2 casillas. A cada casilla le asignamos un valor booleano (cierto/falso). El juego consiste en lo siguiente: dos jugadores se turnan en seleccionar una casilla del tablero. La casilla seleccionada queda eliminada de la partida y modifica el valor de la otra casilla de la misma fila y el valor de la otra casilla de la misma columna. A la casilla de la misma fila se le asigna el resultado de hacer la O lógica entre el valor que contiene y el de la casilla seleccionada. A la casilla de la misma columna se le asigna el resultado de hacer la Y lógica entre el valor que contiene y el de la casilla seleccionada. El juego termina cuando sólo queda una casilla. Empezando moviendo nosotros, ganamos la partida si la última casilla sin seleccionar tiene valor Falso y perdemos si queda con valor Cierto. A partir del siguiente tablero:

C	F
F	C

- (a) Aplica el algoritmo MiniMax para decidir qué casilla debemos seleccionar primero.
- (b) Repite la exploración aplicando poda alfa-beta.

Orden de generación de sucesores: empieza por la casilla de la esquina superior izquierda y sigue el sentido de las agujas del reloj.

11. El juego del reversi se juega en un tablero de  $N \times N$  como el de la figura y consiste en cubrir todo el tablero con fichas, colocándolas alternativamente, de manera que gane el que más fichas de su color consiga. Las reglas del juego son las siguientes: Para colocar una ficha hay que tener en esa fila, columna o diagonal otra ficha propia que encierre fichas contrarias con la que colocamos. Al colocar la ficha, todas las fichas de color contrario que queden atrapadas entre dos fichas propias cambian su color al nuestro. El tablero comienza con las fichas que aparecen en la figura.

	N	B	
	B	N	

- (a) Utiliza el algoritmo de minimax con poda alfa-beta para averiguar qué movimiento deberían hacer las blancas desde la posición inicial. Explora hasta profundidad 2 (un movimiento de las blancas y otro de las negras). Para la evaluación de los estados utiliza la siguiente tabla para asignar valor a cada ficha, el valor de un estado se obtiene sumando los valores de las fichas propias y restando las del contrario.

4	2	2	4
2	1	1	2
2	1	1	2
4	2	2	4

Orden de generación de sucesores:

Para expandir los nodos supón que el tablero está numerado desde la esquina superior izquierda siguiendo el orden por filas. Expande los nodos siguiendo el orden de esta numeración.

- (b) Utiliza el algoritmo MiniMax con poda alfa-beta para averiguar qué movimiento deberían hacer las negras desde la posición escogida por las blancas en el apartado anterior (en caso de empate escoge el primer movimiento que desarrollaste). Explora hasta profundidad 2 (un movimiento de las negras y otro de las blancas). Para la evaluación de las posiciones utiliza la tabla anterior.

12. Tenemos el juego que muestra la figura, que consiste en lo siguiente:

1	2
2	1
1	2
2	1

Cada jugador posee una mitad del tablero, el jugador MAX tiene la mitad superior y el jugador MIN la mitad inferior. Cada jugador escoge en cada turno una de las celdas de su mitad. Cuando se escoge una celda, ésta pasa su valor a sus celdas adyacentes de la siguiente manera: 2 puntos (o 1 si la celda sólo contiene 1) pasan a la celda inferior en el caso de MAX o superior en el caso de MIN, el resto de puntos que queden se reparten, primero un punto a la celda de la izquierda y después un punto a la celda de la derecha, si aun quedan puntos, uno pasa a la celda superior en el caso de MAX o inferior en el caso de MIN. Los puntos que queden se mantienen en la celda elegida. Si se reparten todos los puntos la celda queda con un valor de 0. Se gana cuando se consigue que en la zona del contrario la suma de los valores de las celdas sea igual o superior a 9.

- (a) Aplicar el algoritmo minimax con poda alfa-beta dos niveles (una jugada de MAX y otra de MIN) para averiguar cual sería la jugada de MAX a partir de la posición inicial. Para evaluar las posiciones se ha de utilizar la siguiente función:

$$V_{\text{posición}} = V_{\text{MAX}} - V_{\text{MIN}}$$

$V_{\text{MAX}}$  = suma de valores de la zona MIN - suma de diferencias de la zona MAX (valor de celda mayor de 2 - 2)

$V_{\text{MIN}}$  = suma de valores de la zona MAX - suma de diferencias de la zona MIN (valor de celda mayor de 2 - 2)

Por ejemplo:

0	2
3	3
1	0
2	1

$$V_{\text{MAX}} = 4 - (1+1) = 2, V_{\text{MIN}} = 8 - 0 = 8, V_{\text{posición}} = -6$$

Para elegir el orden de expansión de los nodos, comenzar por la celda de mayor valor. En caso de empate se escoge la que esté más cerca de la mitad y, en caso de nuevo empate, de izquierda a derecha.

- (b) A partir de la jugada elegida por MAX, hacer una nueva búsqueda para averiguar cual será la respuesta de MIN.

13. El siguiente juego sobre un tablero 3x4 empieza con las fichas 'X' en la fila 1 y con las fichas 'O' en la fila 4. Un jugador gana cuando consigue colocar todas sus fichas en la fila del contrincante. Dada la siguiente situación del juego, donde le toca jugar al contrincante (fichas 'O'),

	1	2	3
1	X		
2	X		
3	O	X	O
4			O

teniendo en cuenta que:

- Las fichas 'X' avanzan sólo hacia abajo, ya sea en diagonal o en vertical.
- Las fichas 'O' sólo avanzan hacia arriba, de la misma manera.
- Las fichas sólo pueden avanzar una posición en cada turno y siempre a una posición libre. Sin embargo, una ficha puede saltar sobre una o dos fichas contrarias contiguas.

y aplicando los movimientos Estrictamente en el siguiente orden. Selección de fichas: mover primero la ficha más cercana al objetivo que esté más a la izquierda, seleccionar la segunda con el mismo criterio y luego mover la tercera. Orden de selección de movimientos: 1) Saltar. 2) Mover a diagonal izquierda. 3) Mover en vertical. 4) Mover a diagonal derecha.

- (a) Aplicar el algoritmo minimax con una profundidad máxima 2 para obtener la valoración de la configuración de la figura. Utiliza la siguiente función de evaluación de estados:  $F_{AE} = V('O') - V('X')$   
donde  $V('*')$  es el número de movimientos posibles para las fichas \*. Si no hay posibilidad de movimientos para esas fichas, el valor de  $V('*')$  será  $\infty$ .
- (b) Aplicar el algoritmo minimax  $\alpha\beta$  bajo las mismas circunstancias para el mismo propósito.
- (c) ¿Cuántos nodos se han podado?

(d) ¿Existe algún orden de aplicación de movimientos diferente al exigido que podría provocar más podas?

14. El siguiente juego sobre un tablero de 3x4 empieza con las fichas 'A' en la fila 1 y con las fichas 'B' en la fila 4. Un jugador gana cuando consigue colocar todas sus fichas en la fila del contrincante. Dada la siguiente situación del juego, donde el contrincante acaba de jugar (fichas 'B'),

	1	2	3
1	A		
2	A	B	A
3			B
4			B

teniendo en cuenta que:

- Las fichas 'A' avanzan sólo hacia abajo, ya sea en diagonal o en vertical.
- Las fichas 'B' sólo avanzan hacia arriba, de la misma manera.
- Las fichas sólo pueden avanzar una posición en cada turno y siempre a una posición libre. Sin embargo, una ficha puede saltar una o dos fichas contrarias contiguas.

y aplicando los movimientos **ESTRICTAMENTE** en el siguiente orden. Selección de fichas: mover primero la ficha más cercana al objetivo que esté más a la izquierda, seleccionar la segunda con el mismo criterio y luego mover la tercera. Orden de selección de movimientos: 1) Saltar. 2) Mover a diagonal izquierda. 3) Mover en vertical. 4) Mover a diagonal derecha.

- (a) Aplicar el algoritmo minimax con una profundidad máxima 2 para obtener la siguiente jugada a realizar. Utiliza la siguiente función de evaluación de estados:

$$FAE = V('B') - V('A')$$

donde  $V(*)$  es el número de movimientos posibles para las fichas \*. Si no hay posibilidad de movimientos para esas fichas, el valor de  $V(*)$  será  $\infty$ .

- (b) Aplicar el algoritmo minimax  $\alpha\beta$  bajo las mismas circunstancias para el mismo propósito. ¿Qué jugada te recomienda ahora?
- (c) ¿Cuántos nodos se han podado?
- (d) ¿Existe algún orden de aplicación de movimientos diferente al exigido que podría provocar más podas?

15. Tenemos el juego que muestra la figura, que consiste en lo siguiente:

$$1111+1111$$

la única pieza móvil es el + y se puede mover a la derecha o a la izquierda, el efecto que tiene es intercambiarse con el número de la dirección a la que se desplaza y sumar a todas las posiciones que están en el lado del movimiento el número con el que se intercambia. Por ejemplo:

$$1111+1111$$

(Desplazamos + a la izquierda)

$$222+1111$$

(Desplazamos + a la izquierda)

$$44+21111$$

Tomando como referencia todo el tablero y no importando donde está el +, el jugador MAX posee los cuatro números de la izquierda y el MIN los cuatro números de la derecha. Gana el jugador que consigue que sus números sumen más de 50. Aplicar el algoritmo minimax con poda alfa-beta explorando cuatro niveles (dos jugadas de MAX y otras dos de MIN) para averiguar cual sería la jugada de MAX a partir de la posición inicial. Para evaluar las posiciones se ha de utilizar la siguiente función:

$$V_{\text{posición}} = \text{Suma de los números de MAX} - \text{suma de los números de MIN}$$

El orden de expansión es primero los movimientos de la izquierda y después los de la derecha.



16. Una persona está jugando al Conecta-4 en su ordenador. Desde el punto de vista de la máquina, su contrincante es la persona. El juego consta de un tablero de 4 pilas (columnas) de altura 4 (filas). Cada jugador dispone de 8 fichas ('X' para el ordenador y 'O' para la persona). En cada turno de un jugador, éste puede colocar una de sus fichas en una de las 4 pilas siempre y cuando la pila contenga menos de 4 fichas. Una vez colocada una ficha, ésta no se puede volver a mover.

Un jugador gana cuando consigue que 4 de sus fichas queden alineadas en horizontal, en vertical o en diagonal. Suponiendo el siguiente estado del juego en el que le toca jugar al ordenador (X):

O			
X		O	X
X	O	X	O
1	2	3	4

- (a) a ¿Cuál sería el movimiento que haría la máquina si el juego estuviera implementado con un algoritmo minimax de profundidad 2? Representa los pasos seguidos por dicho algoritmo suponiendo que se está utilizando la siguiente FAE:

FAE = posibles alineaciones para el ordenador - posibles alineaciones para la persona

donde una posible alineación significa la existencia de una posibilidad de llegar a conseguir una alineación en lo que queda de juego. Por ejemplo, en el estado dibujado, la persona (O) tiene 4 posibles alineaciones (una en diagonal, una vertical y dos horizontales) mientras que el ordenador (X) tiene dos posibilidades (una en diagonal y una horizontal).

**ORDEN Estricto:** Considera las posibles jugadas recorriendo las columnas de izquierda a derecha.

- (b) ¿Cuál sería dicho movimiento si la implementación incorporase podas  $\alpha\beta$ ? Representa los pasos seguidos por dicho algoritmo de forma separada del apartado (a). Para ello etiqueta con letras mayúsculas (A, ... , Z, AA, ...) cada estado del apartado (a) y utilízalas aquí. ¿Cuántos nodos se han podado?

17. El juego del Nim se juega a partir de varias filas de palillos de las cuales un jugador puede retirar, en un turno, el número que desee de palillos pero de una única fila. El jugador que retira el último palillo pierde.

Podemos representar la configuración de las filas mediante secuencias de enteros. Por ejemplo, (1, 3, 5) indica que hay tres filas con uno, tres y cinco palillos respectivamente.

Se pide:

- (a) A partir de la configuración (1, 2, 2), utiliza el algoritmo minimax para averiguar qué movimiento deberíamos realizar primero.
- (b) Repite la exploración utilizando la poda alfa-beta. ¿Qué movimiento nos aconseja? ¿Cuántos nodos se han podado?

Orden de generación de sucesores: de izquierda a derecha según el orden indicado en la secuencia de enteros y aplicando todos los movimientos posibles para cada fila antes de pasar a la siguiente. Dentro de cada fila, se comenzará por retirar un palillo, luego dos, y así sucesivamente. Si el estado resultante de la aplicación de un movimiento es una permutación de otro estado hermano anterior, no hay que incluirlo en el árbol (por ejemplo, de (1,2,2) se puede pasar a (1,1,2) y también a (1,2,1), pero este último no se incluirá en el árbol). Estados finales: En el árbol, son estados finales tanto los estados con cero palillos como los estados con un palillo ya que en ambos casos podemos marcar quien es el ganador.

18. Tenemos el tablero que aparece en la figura, en la que el jugador MAX tiene las fichas marcadas como A y el MIN las fichas marcadas como B. Los jugadores pueden mover sus fichas y las del contrario. Los movimientos posibles son dos:

- desplazamiento de una ficha a la casilla libre contigua
- salto de una ficha sobre otra contraria (sólo una) cambiando de signo la ficha contraria, siempre y cuando la siguiente casilla esté libre. Ejemplo: ABA\_BBAB  $\Rightarrow$  A\_BBAB

Restricción: una jugada de desplazamiento no es válida si simplemente está deshaciendo la jugada anterior. Por ejemplo, si un jugador pasa de AB\_BBAB al estado ABB\_BBAB, el otro jugador no puede volver a pasar a AB\_BBAB. El objetivo de cada jugador es tener 4 fichas del mismo tipo estrictamente consecutivas.

A	B	A		B	A	B
---	---	---	--	---	---	---

Para la evaluación de las configuraciones utilizaremos la siguiente función:

$f'(n)$  = tamaño del mayor grupo de As consecutivas - tamaño del mayor grupo de Bs consecutivas.

Ejemplos:

$$f'(ABA\_BAB) = 1 - 1 = 0$$

$$f'(ABB\_BAB) = 1 - 2 = -1$$

- Utiliza el algoritmo minimax con poda alfa-beta para evaluar cual debería ser el primer movimiento del jugador MAX. Haz la exploración hasta el nivel 3 (dos jugadas de MAX y una de MIN). Aplica siempre el mismo orden: movimientos posibles recorriendo el tablero de izquierda a derecha. Para cada nodo muestra claramente la evolución de los valores alfa y beta. ¿Cuál es el movimiento que debería escoger MAX?
- ¿Cuántos nodos nos ahorramos respecto al uso del algoritmo sin poda?

## 5. Satisfacció de restriccions

1. Una empresa de alquiler de vehículos dispone de una flota con las siguientes características:

	Marca	Color	Precio/día
Coche1	Ford	Blanco	30 euros
Coche2	Citroen	Azul	35 euros
Coche3	Mercedes	Gris	60 euros
Coche4	Citroen	Verde	30 euros
Coche5	Seat	Rojo	35 euros
Coche6	Opel	Blanco	35 euros
Coche7	Mercedes	Rojo	35 euros
Coche8	BMW	Negro	60 euros
Coche9	Citroen	Gris	40 euros
Coche10	Seat	Azul	40 euros

La empresa ha recibido la siguiente serie de peticiones que ha de intentar satisfacer:

	Precio/día	Color
Petición1	$\leq 40$ euros	NO Azul
Petición2	$\leq 30$ euros	Azul O Blanco
Petición3	$\leq 60$ euros	NO Gris y NO Negro
Petición4	$\leq 40$ euros	NO Rojo y NO Azul
Petición5	$\leq 60$ euros	Blanco

Tenemos la restricción adicional de que no puede haber dos coches de la misma marca. Considera las peticiones como variables, en el mismo orden en que figuran, y los coches de la flota como valores, también en el mismo orden en que se listan.

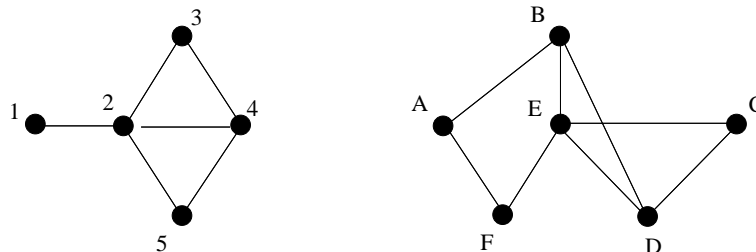
- (a) Aplica forward-checking para resolver el problema. Desarrolla el proceso hasta el punto en el que encuentres una solución. Indica en cada paso sólo los dominios que se modifican.
- (b) Aplica backtracking cronológico hasta el segundo backtracking a la variable petición 1.

2. El director de la coral infantil "Veus suaus" tiene que decidir el orden de colocación de los ocho cantores. Debe distribuirlos en dos filas (A, B) de forma que en cada fila queden en orden decreciente de altura, colocando el más alto a la izquierda (posición 1). Además, la altura de cada niño de la fila trasera (A) debe ser superior o igual a la del que tenga delante. Finalmente, no quiere colocar dos hermanos seguid@s en la misma fila ni un@ delante del otr@. La relación de niños y sus alturas es la siguiente:

Nombre	Altura
Esteva Blanco (EB)	1,40
Pedro Costa (PC)	1,60
Ana Costa (AC)	1,50
Juan Costa (JC)	1,30
Oriol Pi (OP)	1,40
María Ruiz (MR)	1,60
Rosa Sánchez (RS)	1,50
Carla Sánchez (CS)	1,30

Considera como variables las posiciones de izquierda a derecha en las filas (A1, A2, A3, A4, B1, B2, B3, B4) y como valores las iniciales de los nombres de los cantores. El orden de recorrido de las variables es el indicado entre paréntesis. El orden de recorrido de los valores es el de la lista anterior.

- Aplica forward-checking para resolver este problema. Desarrolla el proceso hasta el punto en el que hay que realizar el primer backtracking a B1.
  - Aplica búsqueda en profundidad y backtracking cronológico hasta el primer backtracking a A2.
3. Deseamos utilizar la técnica de satisfacción de restricciones para resolver el problema de comprobar si un grafo está contenido en otro. Dados los dos grafos siguientes:



Deseamos saber si el primer grafo está contenido en el segundo.

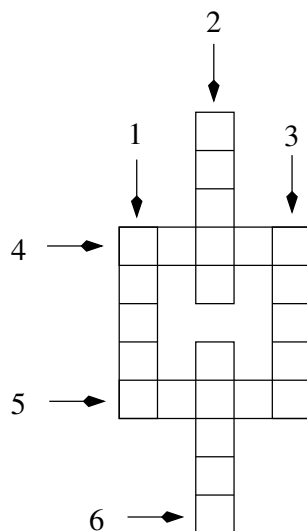
Para resolver el problema debemos asignar a cada nodo del primer grafo un nodo del segundo grafo de manera que las conexiones entre los nodos del primer grafo se respeten en el segundo grafo.

- Define los dominios de cada variable (nodos del primer grafo) y aplica las restricciones que creas posibles en cada variable para reducir el número de valores posibles (las que sean evidentes por las características del problema, no apliques arco consistencia).
  - Resolver el problema aplicando el algoritmo del forward checking
  - Resolver el problema aplicando el algoritmo de backtracking cronológico hasta que la variable 1 tome el valor C
  - Si tomamos como variables los nodos del segundo grafo ¿cuales serían los valores de los dominios de estas variables?
4. Dado un vector de cinco posiciones se desea obtener una asignación de letras tal que no haya dos letras consecutivas iguales y que el conjunto sea capicúa.
- Los valores posibles para la posición 1 son A,B,C,D,E.
  - Los valores posibles para las posiciones 2 y 3 son A,B,C.

- Los valores posibles para la posición 4 son C,D,E.
- Los valores posibles para la posición 5 son B,C,D,E.

Ejemplos de asignaciones válidas: E-C-A-C-E      D-C-B-C-D

- Resolver el problema mediante backtracking cronológico.
  - Resolver el problema mediante forward checking.
5. Como cada año, los protagonistas de los más famosos culebrones se reúnen para celebrar la realización de 100 nuevos capítulos de su teleserie. Lamentablemente la rivalidad entre ellos es tal que algunos no se pueden sentar al lado de otros durante la cena de celebración. Nuestros protagonistas son: Carlos Miguel, David Miguel, Carlos David, Juan Miguel, Pedro Miguel, Juan Luis y Juan Carlos. Las restricciones son tales que un Carlos no se puede sentar junto a un Pedro, ni un Juan se puede sentar al lado de otro Juan.
- Utiliza el algoritmo de forward checking para dar una solución a la ubicación de estos 7 personajes en una mesa circular (cada comensal sólo tiene dos vecinos, uno a la derecha y otro a la izquierda). Usa para la exploración de los valores el orden en el que están en el enunciado.
  - Resuelve el mismo problema aplicando el algoritmo de backtracking cronológico.
6. Deseamos resolver el crucigrama de la figura colocando las siguientes palabras: ROLLO, SALVO, SOLAR, ROCAS, OCIOS, SILOS.
- Para resolverlo aplica el algoritmo de forward checking utilizando las posiciones para las palabras como variables y las palabras como valores utilizando la numeración para las variables que se indica la figura y explorando las palabras en el orden en el que se dan en el enunciado.
  - Resuelve ahora el problema aplicando el backtracking cronológico.



NOTA: las palabras horizontales se colocan de izquierda a derecha y las palabras verticales de arriba a abajo. No hay palabras escritas al revés.

7. Una petita empresa ha de comprar telèfons mòbils pels seus directius. Han estat mirant preus de companyies telefòniques i es troben amb la següent informació:

Companyia	Tarifa
MultiStaf (MS)	5000
AguaTel (AT)	6000
FunkyTel (FT)	4500
TopeVisión (TV)	3500

Aquesta petita companyia té cinc executius i vol assignar-los un telèfon mòbil a cadascun sota les següents restriccions:

- No pot haver-hi més de dos executius amb la mateixa companyia.
- El cost total de l'assignació de companyies a executius no ha de superar les 21.000 ptes.

Es demana:

- Resoldre completament el problema aplicant forward checking.
- Desenvolupar l'aplicació del backtracking cronològic fins al punt en que el backtracking ens porta a revisar la primera assignació feta a la segona variable.

NOTA: Per ambdós apartats considereu que els executius es representen per E1, E2, E3, E4 i E5. Els valors possibles són MS, AT, FT, TV i cal considerar-los sempre en aquest ordre.

- La compañía de aviación “Air Vostrum” debe realizar habitualmente la tarea de configurar la tripulación de los vuelos. El problema actual consiste en organizar parejas de comandante y piloto para cubrir cuatro vuelos: París, Roma, Beijing y Tokio. El personal disponible es:

Comandantes			Pilotos		
C1	Pérez	45 años	P1	Asensio	32 años
C2	Benitez	43 años	P2	Martín	35 años
C3	Almansa	40 años	P3	Marín	38 años
C4	Morales	47 años	P4	Casales	40 años

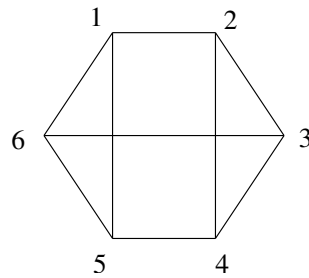
Las normas de la compañía, para vuelos fuera de Europa, impiden que la suma de edades del comandante y el piloto exceda de 75 años. Adicionalmente, los comandantes son muy supersticiosos y no admiten que su piloto tenga como inicial de apellido la misma que ellos. Aplica el algoritmo de forward checking para configurar las cuatro tripulaciones necesarias. Explicita los valores posibles de cada variable en cada paso. Utiliza variables y valores en el orden en que aparecen en el enunciado.

- El senyors Grífol tenen el problema habitual de cada Nadal i que consisteix en comprar regals per als seus nebots/nebodes. Han de comprar regals per als germans Joan i Maria, per als germans Pere, Ana i Oriol i per en Ramon i en Xavier, ambdós fills únics. Han decidit que compraran dos jocs iguals d'escacs, dos jocs iguals d'experiments de química i tres llibres didàctics iguals. Per a repartir aquests regals entre els nebots han de tenir en compte que, per raons òbvies, no poden donar el mateix regal entre germans, que ni el Joan ni el Xavier poden tenir el mateix regal que l'Oriol i que el Ramon no pot tenir el mateix regal que l'Ana.

NOTA: Utilitza els noms dels nebots en l'ordre en que apareixen a l'enunciat. Els regals s'identifiquen per E1, E2, Q1, Q2, L1, L2, L3 i també cal respectar aquest ordre.

- Dibuixa el graf de restriccions entre les variables.
- Resol el problema aplicant forward checking. Indica clarament a l'inici i a cada pas el domini de les variables.
- Inicia el procés de resolució aplicant cerca en profunditat i backtracking cronològic. Desenvolupa el procés fins al moment en que es produeix el segon backtracking cap a Oriol.
- Creus que el backtracking cronològic trobarà la solució? Justifica la resposta.

- Una empresa de telefonía móvil quiere colocar seis antenas en una ciudad cuyas posibles ubicaciones están reflejadas por este grafo:



Donde cada arco indica que existe visión directa entre dos posiciones. Las antenas tienen las características siguientes:

Antena	Frecuencia	Potencia
A	20 GHz	1 Mw
B	1.8 GHz	2 Mw
C	20 GHz	3 Mw
D	1.8 GHz	1 Mw
E	20 GHz	1 Mw
F	1.8 GHz	2 Mw

Las restricciones de colocación de las antenas son las siguientes: No se pueden colocar dos antenas consecutivas de la misma frecuencia No puede haber dos antenas con visibilidad directa cuya suma de potencia sea superior a 4 Mw

- (a) Considerando las posiciones de las antenas como variables y las antenas como los dominios de estas variables, utiliza el algoritmo del forward checking para buscar una solución a la colocación de las antenas usando para la exploración el orden en el que aparecen las posiciones y las antenas en el enunciado. Desarrolla el algoritmo hasta el primer backtracking a la primera variable.
  - (b) Desarrolla el algoritmo de backtracking cronológico hasta el punto en que deba realizarse el primer backtracking. Para cada asignación que falle indica brevemente el motivo.
11. El responsable del periódico "Noticias frescas" debe confeccionar la primera página de la edición de mañana que consta de cuatro posiciones tal como se ve en la figura

Noticias Frescas 8-junio-2002	
P1	P2
P3	P4

Dispone de siete noticias con las siguientes características:

	Ámbito	Tema
N1:	Nacional	Política
N2:	Nacional	Sucesos
N3:	Internacional	Política
N4:	Nacional	Deportes
N5:	Nacional	Política
N6:	Internacional	Sociedad
N7:	Internacional	Sucesos

Las normas de redacción del periódico imponen las siguientes restricciones:

- Una noticia internacional sólo puede estar en una posición inferior (P3 o P4) si la de justo encima es también internacional.
- Sólo puede haber dos noticias del mismo tema si están en diagonal.
- Una noticia de política y otra de sucesos no pueden estar en la misma horizontal.

Considerando las variables P1..P4 y los valores N1..N7 y respetando el orden numérico, se pide:

- resolver el problema de la confección de la primera página aplicando forward-checking
- resolver el mismo problema mediante backtracking cronológico.

12. Dos cadenas de televisión desean coordinar sus franjas horarias de manera que el telespectador tenga más donde elegir. Supondremos que cada cadena divide su horario en cuatro franjas, y que en cada franja podemos tener uno de estos tres tipos de programa: Fútbol, Concurso o Película. Tenemos como restricciones:

- No puede haber simultáneamente el mismo tipo de programa en las dos cadenas.
- No puede haber en una misma cadena dos programas seguidos del mismo tipo.
- No puede haber en total más de dos partidos de fútbol o concursos.

Utilizando las franjas horarias como variables y siguiendo el orden: Fr1.1, Fr1.2, Fr1.3, Fr1.4, Fr2.1, Fr2.2, Fr2.3, Fr2.4 (donde el primer número es la cadena y el segundo la franja horaria) y usando para los valores posibles el orden fútbol, concurso, película, utiliza el algoritmo de forward checking para encontrar una solución a este problema.

13. Deseamos construir un circuito ubicando los componentes de manera que las distancias que los componentes permitan su correcto funcionamiento. El circuito sobre el que queremos trabajar es una cuadrícula de 3x3, de manera que cada componente tiene asignado unas coordenadas.

Tenemos que ubicar 4 componentes (A,B,C,D), las restricciones entre ellos son:

- La distancia entre el componente A y B ha de ser mayor o igual a 2
- La distancia entre el componente A y C ha de ser 1
- La distancia entre el componente B y C ha de ser 2
- La distancia entre el componente B y D ha de ser 1
- La distancia entre el componente C y D ha de ser 1

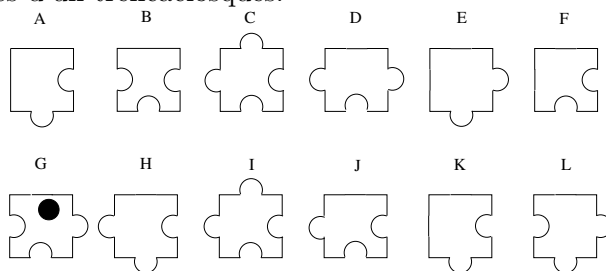
Para calcular la distancia entre dos componentes se usa la función:  $d(a,b)=|ax-bx|+|ay-by|-1$ , donde  $ax$  es el valor de la coordenada  $x$  del componente  $a$ ,  $ay$  es el valor de la coordenada  $y$  del componente  $a$ , ídem para el componente  $b$ . Considera los componentes como variables y las componentes como los valores de estas variables. Como orden de exploración, empieza por la esquina superior izquierda y sigue el orden de izquierda a derecha y de arriba a abajo.

- Utiliza el algoritmo de forward checking para explorar el problema hasta llegar al tercer valor de la variable A
- Utiliza el algoritmo de backtracking cronológico para explorar el problema hasta el primer backtracking a la variable B
- ¿Se puede simplificar el primer problema eliminando valores del dominio de las variables propagando las restricciones? ¿Porqué?
- ¿Se puede aprovechar alguna característica del problema para simplificarlo y probar menos valores en la búsqueda?

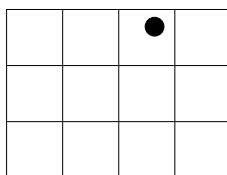


- (e) En muchas ocasiones se pueden intercambiar variables y valores en los problemas de satisfacción de restricciones. ¿Cuáles serían los dominios de valores si usáramos como variables las posiciones en lugar de los componentes?

14. Tenim les següents peces d'un trencaclosques:



Les 12 peces han de formar la següent imatge:



Fent servir les següents posicions del trencaclosques com a variables i les peces com a valors (poden efectuar-se rotacions sobre les peces, excepte per la C i la I):

1	2	3	4
10	11	12	5
9	8	7	6

el domini de cada variable queda inicialment restringit, donades les característiques del puzzle.

- Utilitza backtracking cronològic per completar el trencaclosques.
- Fes-ho ara amb forward checking.

**NOTA:** En ambdós casos, feu servir l'ordre numèric per les variables i l'ordre alfabètic pels valors. Deixa clar quins són els passos donats!!

15. Una empresa de consultoria tiene que organizar el trabajo de tres proyectos durante tres días y dispone de cuatro consultores para llevarlos a cabo. Cada consultor puede dedicar un conjunto de horas a cada proyecto, este conjunto de horas está dividido en fracciones (un consultor sólo tiene dos fracciones de dedicación al día), las horas de cada consultor en cada fracción de dedicación son las siguientes:

Consultor	Disponibilidad
C1	2.5
C2	4
C3	3
C4	1.5

Existen las siguientes restricciones de asignación:

- Un consultor no puede trabajar dos días seguidos en el mismo proyecto
- Un consultor sólo puede dedicar dos fracciones de dedicación al día
- El número total de horas dedicadas a un proyecto ha de ser inferior o igual a 8
- Todos los días ha de haber algún consultor asignado a un proyecto

Utilizar los proyectos combinados con los días como variables, asignando valores para todos los proyectos para el primer día, para el segundo día y para el tercer día (P1d1, P2d1, P3d1, P1d2, ...), usa los consultores como valores (C1-C4)

- Resolver el problema aplicando forward checking. En cada paso indica solamente las variables en las que hay algún cambio en su dominio.
- Describe que otras maneras hay de escoger variables y valores para hacer la exploración y que ventajas o inconvenientes tienen.

16. El club de natació “Dofins mulars” ha d’inscriure per a les properes competicions dos equips de relleus (A i B). Els nedadors a inscriure (amb les seves corresponents edats) són: Manel(10), Oriol(10), Adrià(11), Enric(11), Xavier(12), Pere(12), Joan(13) i Ramon(13). Cada equip està format per quatre nedadors. L’entrenador ha de tenir en compte les següents restriccions:

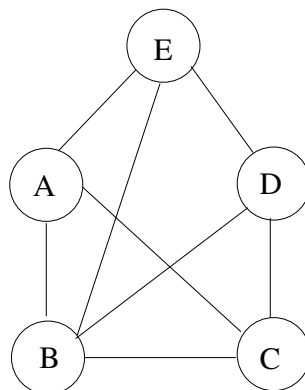
- La suma d’edats de cada equip no pot ser superior a 46.
- No poden haver-hi en el mateix equip dos nedadors de 10 anys.
- No poden haver-hi en el mateix equip dos nedadors de 13 anys.
- El Pere i el Joan no poden estar en el mateix equip.
- En Xavier i l’Adrià no poden estar en el mateix equip.

Considera els nedadors com les variables del problema i tracta-les en l’ordre en que apareixen els noms a l’enunciat.

- Representa les restriccions de la 2 a la 5 mitjançant un graf.
- Resol el problema aplicant forward checking. Indica a cada pas només els dominis que es modifiquen. Quan correspongui, indica breument per què cal assignar un nou valor o fer backtracking.
- Resol el problema aplicant cerca en profunditat amb backtracking cronològic.

17. Dado el siguiente grafo de restricciones donde cada restricción es una condición de desigualdad y los siguientes dominios para las variables:

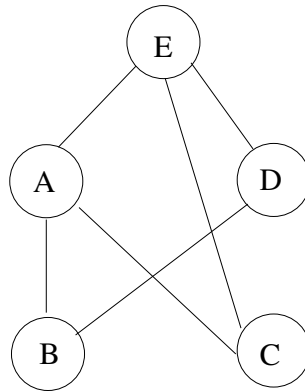
$A = \{1, 2, 3\}$   
 $B = \{1, 2\}$   
 $C = \{2, 3\}$   
 $D = \{2, 3\}$   
 $E = \{1, 2, 3\}$



Haz la ejecución del backtracking cronológico hasta el primer backtracking a la variable A y del forward checking hasta encontrar la primera solución

18. Dado el siguiente grafo de restricciones donde cada restricción es una condición de desigualdad y los siguientes dominios para las variables:

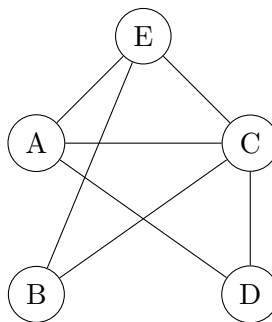
$$\begin{aligned} A &= \{1, 2\} \\ B &= \{2, 3\} \\ C &= \{1, 3\} \\ D &= \{2, 3\} \\ E &= \{1, 3\} \end{aligned}$$



Haz la ejecución del forward checking hasta encontrar la primera solución

19. Dado el siguiente grafo de restricciones donde cada restricción es una condición de desigualdad y los siguientes dominios para las variables:

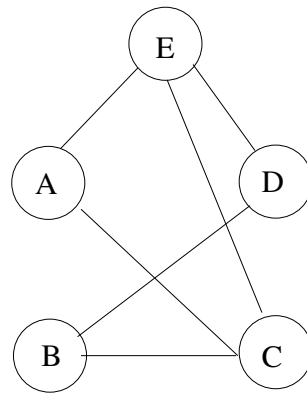
$$\begin{aligned} A &= \{1, 2\} \\ B &= \{2, 3\} \\ C &= \{1, 2\} \\ D &= \{1, 2, 3\} \\ E &= \{1, 2, 3\} \end{aligned}$$



Haz la ejecución del forward checking hasta encontrar la primera solución

20. Dado el siguiente grafo de restricciones donde cada restricción es una condición de desigualdad y los siguientes dominios para las variables:

$$\begin{aligned} A &= \{1, 2\} \\ B &= \{1, 3\} \\ C &= \{1, 2\} \\ D &= \{1, 3\} \\ E &= \{1, 2\} \end{aligned}$$



Haz la ejecución del forward checking hasta encontrar la primera solución

## 6. Anàlisi de mètodes de cerca

1. Queremos conectar  $N$  ciudades con  $B$  autobuses. Cada autobús ha de comenzar su trayecto en la ciudad  $C_1$  y finalizarlo en la ciudad  $C_N$ . Conocemos las distancias entre cada par de ciudades y queremos que el camino recorrido por cada autobús no exceda  $K$  kilómetros y que el número de paradas que haga cada autobús no sea mayor que  $P$ .

Buscamos la solución que conecte todas las ciudades sin que dos autobuses pasen por la misma ciudad (exceptuando  $C_1$  y  $C_N$ ) y recorriendo el mínimo número de kilómetros.

- (a) Queremos utilizar  $A^*$ . Definimos el estado como la asignación de ciudades a autobuses. El estado inicial consiste en asignar las ciudades  $C_1$  y  $C_N$  a cada autobús. Tenemos un operador que asigna una ciudad a un autobús y cuyo coste es la suma de las distancias entre la ciudad asignada con  $C_1$  y  $C_N$ , sólo podemos asignar una ciudad a un autobús si no excedemos el valor  $P$ . Como función heurística usamos la suma de las distancias de todas las ciudades por asignar con  $C_1$  y  $C_N$ .
- (b) Aplicar un algoritmo de satisfacción de restricciones. Las variables son las ciudades, los dominios son los autobuses donde podemos asignarlas. Supondremos que no tenemos en cuenta las ciudades  $C_1$  y  $C_N$  ya que estarán asignadas a todos los autobuses. Como restricciones imponemos que un autobús no esté asignado a más de  $P$  ciudades y que la longitud del camino mínimo que recorre las ciudades asignadas a un autobús no sea mayor que  $K$ .

Comenta cada una de las posibilidades indicando si resuelven o no el problema, qué errores te parece que tiene cada solución y cómo se podrían corregir, y qué ventajas e inconvenientes tienen cada una de ellas. Justifica la respuesta.

2. Tenemos un problema de bioinformática que consiste en lo siguiente: Tenemos una cadena de ADN de una longitud fija y un conjunto de subcadenas que tenemos que encajar en cualquier posición de la cadena donde coincidan las letras. Cada subcadena tiene una longitud ( $l_i$ ). El objetivo es buscar el mínimo subconjunto de las subcadenas que permita cubrir la mayor longitud posible de la cadena, sin que haya ningún solapamiento entre las posiciones que cubren las subcadenas escogidas.
  - (a) Queremos utilizar  $A^*$ , consideramos el estado como las subcadenas que tenemos asignadas. El estado inicial es no tener ninguna cadena asignada. Tenemos un operador que asigna una subcadena a alguna de las posiciones en las que coincide con la cadena, el coste de este operador es uno. Como función heurística usamos la suma de las longitudes de los trozos de la cadena que no están cubiertos.

- (b) Queremos utilizar satisfacción de restricciones. Consideramos que las subcadenas son las variables y los dominios son todas las posiciones donde se pueden encajar las subcadenas en la cadena. Como restricciones tenemos que para cada par de cadenas no puede haber solapamiento entre las posiciones que corresponden a sus asignaciones y que la suma de las longitudes de las subcadenas que estén asignadas sea en la solución mayor que un valor  $L$ .

Comenta cada una de las posibilidades indicando si resuelven o no el problema, qué errores te parece que tiene cada solución y cómo se podrían corregir, y qué ventajas e inconvenientes tienen cada una de ellas. Justifica la respuesta.

3. Queremos planificar cómo debemos integrar diferentes procesos individuales de fabricación de diversos productos en un único proceso. Cada producto necesita utilizar un conjunto de máquinas en un orden específico que se puede describir mediante una secuencia de proceso. El uso de una máquina es un paso del proceso de fabricación de un producto. Para simplificar supondremos que cada paso tiene la misma duración. Hay que tener en cuenta que un producto puede necesitar un mismo tipo de máquina en diferentes pasos de su proceso de fabricación.

Disponemos de un número fijo de máquinas de cada tipo ( $m_i$ ). Supondremos que necesitamos fabricar una unidad de cada producto y solo podemos volver a iniciar el proceso una vez que hemos acabado todos los productos, es decir, en el proceso integrado sólo se fabrica una unidad de cada producto a la vez. Obviamente, un paso del proceso integrado se compone de un paso de todos o un subconjunto de procesos individuales. Queremos minimizar el número de pasos del proceso integrado.

Se nos plantean las siguientes alternativas:

- (a) Queremos utilizar  $A^*$  para minimizar el número total de pasos del proceso integrado. El estado lo forman los pasos de los procesos individuales que hemos encajado en cada paso del proceso integrado. El operador de cambio de estado consiste en colocar el máximo número de procesos individuales (un paso de cada uno) en el paso actual del proceso integrado, el coste del operador es el número de pasos integrados. La función heurística es la suma de pasos de los procesos individuales que nos quedan por integrar.
- (b) Queremos utilizar satisfacción de restricciones, para ello creamos un grafo de restricciones que tiene  $MAX$  variables para cada producto, considerando que este número es el máximo número de pasos que tendremos. El dominio de cada variable es la máquina que se ha asignado a ese paso. Para cada variable de un producto, ponemos como restricción que todas las variables anteriores en la secuencia deben respetar el orden de la secuencia de fabricación. Para las variables de todos los productos en un paso de fabricación, ponemos la restricción de que el número de máquinas asignadas a los productos en ese paso no supere los máximos de máquinas de los que se dispone.

Comenta cada una de las posibilidades indicando si resuelven o no el problema y qué ventajas e inconvenientes tiene cada una de ellas. Justifica la respuesta.

4. El área de Parques y Jardines del Ayuntamiento de Barcelona quiere modernizar su flota de mini-furgonetas con una nueva, experimental, impulsada por hidrógeno, que tiene la ventaja de ser altamente ecológica y fácil de mantener pero que tiene una autonomía muy limitada. Para resolver el problema de la autonomía se han colocado 3 surtidores de hidrógeno por diferentes partes de la ciudad para que la mini-furgoneta pueda repostar.

Queremos planificar el recorrido diario de esta mini-furgoneta por los diferentes parques de la ciudad de forma que pueda pasar por todos los parques una vez, tardando el mínimo tiempo posible y pasando por uno de los surtidores cada vez que se le esté acabando el hidrógeno. Como datos para esta planificación disponemos de una tabla que nos indica las posiciones de todos los parques y jardines a visitar y de los tres surtidores, así como la distancia entre cada uno de estos puntos en la ciudad. Sabemos también que la mini-furgoneta experimental tiene un tanque de  $n$  litros de hidrógeno y que su consumo es de  $x$  litros de hidrógeno por kilómetro.

- (a) Queremos utilizar  $A^*$  para minimizar el número de kilómetros recorridos por la furgoneta y reducir el número de repostajes al mínimo. Utilizaremos como función de coste la longitud del camino (usando la tabla de distancias antes mencionada), donde la función heurística vale infinito si la mini-furgoneta no tiene suficiente combustible para ir del punto actual a un surtidor, y en caso contrario es la suma de las distancias de los puntos por recorrer al punto actual. El operador aplicable es pasar del punto actual a otro punto.
- (b) Queremos utilizar búsqueda local para minimizar el número de kilómetros recorridos por la mini-furgoneta y reducir el número de repostajes al mínimo. Partimos de una solución inicial en la que intercalamos un paso por el surtidor de hidrógeno entre parque y parque. Se dispone de dos operadores de modificación de la solución: uno para eliminar del camino un paso por el surtidor, y otro para modificar el orden en el que visitamos alguno de los puntos del camino (ya sea parque o surtidor). La función heurística es la longitud del camino recorrido + 2 kilómetros extra de penalización por cada paso por un surtidor.

Comenta cada una de las posibilidades indicando si resuelven o no el problema, qué errores te parece que tiene cada solución y cómo se podrían corregir, y qué ventajas e inconvenientes tienen cada una de ellas. Justifica la respuesta.

5. Los dueños de cierto hipódromo desean configurar la lista de participantes de cada carrera de una jornada. Cada día se realizan 10 carreras con 10 caballos cada una. Para confeccionar las carreras pueden usar caballos de 8 caballerizas profesionales que ponen 7 caballos cada una a su disposición. Un caballo puede correr en varias carreras siempre que tenga como mínimo tres carreras de descanso. Tampoco pueden correr más de dos caballos de la misma caballeriza en una carrera. Que un caballo corra en una competición tiene un coste para el hipódromo que depende del ranking del caballo. Buscamos minimizar el coste total de las carreras respetando las restricciones.

Tras un *brainstorming* se plantean dos estrategias distintas para resolver el problema:

- (a) Usar  $A^*$  tomando como estado la asignación de caballos a carreras. El estado inicial sería la asignación vacía, el estado final sería la asignación completa. Usaríamos como operador de cambio de estado el asignar un caballo a una carrera, su coste sería el coste del caballo asignado. Como función heurística usaríamos el número de caballos que faltan por asignar.
- (b) Usar un algoritmo de satisfacción de restricciones. Supondríamos que un caballo solo puede estar en un máximo de tres carreras. Crearíamos un grafo de restricciones cuyas variables serían las tres posibles carreras de cada caballo. El dominio de cada variable sería la carrera en la que corre el caballo o vacío si el caballo no corre. Tendríamos una restricción entre las carreras consecutivas de un caballo que no permitiera que su diferencia fuera menor que tres y una restricción entre todas las carreras de los caballos de una caballeriza para que no pudiera haber más de dos caballos en la misma carrera.

Comenta cada una de las posibilidades indicando si resuelven o no el problema y qué ventajas e inconvenientes tiene cada una de ellas. Justifica la respuesta.

6. Dada la sequía persistente de los últimos años y el aumento de la gravedad de los incendios, se nos ha planteado rediseñar la ubicación de los parques de bomberos en el territorio. Para solucionar el problema dividimos el territorio en una cuadrícula de  $N \times M$  posiciones, cada posición tiene asignado un factor de accesibilidad  $A$  que toma un valor entero entre 1 y 3 (1 es una zona de fácil acceso, 3 es una zona de difícil acceso).

Deseamos ubicar un total de  $P$  parques de bomberos, cada parque de bomberos puede tener entre uno y tres camiones. Tenemos un total de  $C$  camiones para repartir (obviamente  $C > P$ ).

Definimos el factor de seguridad de una posición como la suma para esa posición y todas las que la rodean del cociente entre el número de camiones de bomberos que hay en esa posición y el factor de accesibilidad de cada posición.

El objetivo es ubicar todos los parques de bomberos y repartir entre ellos todos los camiones de manera que el factor de seguridad global (la suma para todas las posiciones) sea el máximo posible. Estas podrían ser soluciones a este problema:

- (a) Se plantea solucionarlo mediante  $A^*$ , recorriendo la cuadrícula de la esquina superior izquierda a la inferior derecha. El estado es la asignación que hemos hecho de parques de bomberos y camiones a las posiciones recorridas. Utilizamos como operador poner un parque de bomberos, asignando uno, dos o tres camiones en el caso de ponerlo, el coste del operador es el número de camiones asignados más uno, o no ponerlo con coste uno. La función heurística es el número de posiciones que nos quedan por visitar y vale infinito si ya hemos asignado más de  $C$  camiones o  $P$  parques de bomberos.
- (b) Se plantea resolverlo como un problema de satisfacción de restricciones. Consideramos que los  $P$  parques de bomberos son las variables, y el dominio de cada una de ellas está compuesto por pares  $\langle \text{posición, camiones} \rangle$ , tantos como combinaciones posibles hay entre las  $N \times M$  posiciones de la cuadrícula y el número de camiones asignados al parque (entre 1 y 3). Tenemos un grafo de restricciones (binarias) que conecta cada parque con todos los demás para controlar (de dos en dos) que las posiciones asignadas a los parques son diferentes. Tenemos una restricción global que controla, en todo momento, que el número de camiones asignados a parques no exceda el número  $C$ , y otra restricción global que controla que el factor de seguridad global sea superior a un valor  $FSG_{min}$  (asumimos que durante la ejecución del algoritmo PSR existe una función auxiliar que es capaz de calcular el factor de seguridad de todas las posiciones teniendo en cuenta solo los parques de bomberos que tienen posición y camiones asignados, asumiendo que el resto de parques de bomberos y camiones no existen a la hora de hacer el cálculo).

Comenta cada una de las posibilidades indicando si resuelven o no el problema, qué errores te parece que tiene cada solución y cómo se podrían corregir, y qué ventajas e inconvenientes tienen cada una de ellas. Justifica la respuesta.

7. En todo proyecto de desarrollo de una aplicación informática es necesario determinar cuanto tiempo y recursos va a ser necesario invertir. Una vez decididas las tareas, su duración y la dependencia entre ellas podemos determinar automáticamente cuando se van a desarrollar y qué personas van a hacer cada tarea.

Supondremos que hemos dividido el problema en tareas ( $t_1$  a  $t_m$ ) que tienen todas la misma duración y hemos determinado su grafo de dependencias, de manera que para cada una de ellas sabemos qué tareas han de estar terminadas para poder empezarla. Disponemos también de un conjunto de programadores ( $prog_1$  a  $prog_n$ ) que podemos asignar a cada una de las tareas. Obviamente si asignamos el mismo programador a dos tareas que pueden realizarse simultáneamente tardaremos más que si asignamos programadores diferentes, pero siempre nos saldrá más barato usar un número menor de programadores.

El objetivo del problema es encontrar la menor asignación de programadores a tareas que respete el orden de dependencia de estas.

- (a) Queremos usar  $A^*$  definiendo el estado como la asignación de programadores a tareas. El estado inicial sería la asignación vacía y el estado final sería la asignación completa de programadores a tareas. Como operadores usaremos dos: asignar un programador cualquiera a una tarea cualquiera, el coste del operador sería una unidad de duración; y desasignarle a un programador una tarea, el coste de este operador sería una unidad negativa de duración. La función heurística es el número de tareas que quedan por asignar que tengan alguna tarea de la que dependan que no esté ya asignada.
- (b) Queremos usar Satisfacción de Restricciones, donde las variables son las tareas a asignar y los programadores los dominios de las variables. Las restricciones son las relaciones de dependencia entre las diferentes tareas, estableciendo que un programador no puede estar asignado a 2 tareas entre las que haya una dependencia. Sobre estas restricciones imponemos además que no pueda haber más de tres programadores diferentes en tareas que pueden realizarse en paralelo.



Comenta cada una de las posibilidades indicando si resuelven o no el problema, qué errores te parece que tiene cada solución y cómo se podrían corregir, y qué ventajas e inconvenientes tienen cada una de ellas. Justifica la respuesta.

8. El ayuntamiento de una pequeña ciudad quiere ofrecer a sus ciudadanos un servicio similar al Bicing barcelonés, pero a menor escala. Uno de los problemas a resolver es como mover las bicicletas de un punto de recogida a otro para intentar que en toda estación haya algunos puestos libres para dejar bicicletas, y que en toda estación haya alguna bicicleta disponible. El ayuntamiento ya tiene un sistema que decide que bicicletas hay que mover de una estación a otra, y ha subcontratado los servicios de un camión, que puede llevar un máximo de  $B$  bicicletas a la vez. Cada hora el camión ha de recoger y dejar bicicletas en diferentes estaciones del servicio distribuidas por la ciudad, siguiendo un listado del ayuntamiento de pares (estación\_origen, estación\_destino), y ha de hacerlo realizando el recorrido más corto posible, sin que se sobrepase en ningún momento el máximo de bicicletas que el camión puede cargar. Cada hora partimos de cierto punto de origen y volvemos a él, habiendo movido todas las bicicletas que el ayuntamiento ha solicitado de su estación origen a su estación destino. Para obtener el recorrido se dispone de un mapa de la ciudad que indica la distancia mínima entre cada par de estaciones por las que ha de pasar el camión.

Puedes resolverlo mediante:

- (a) El algoritmo de  $A^*$ . El estado es el camino recorrido. Utilizamos como coste la longitud del camino actual. La función heurística vale infinito si el camión en el estado actual supera el número  $B$  de bicicletas que puede transportar y, en caso contrario, es la suma de las distancias de las estaciones por recorrer al origen. El operador aplicable es pasar de la estación actual a otra no visitada. Para evitar la necesidad de otro operador, el punto de origen y final del camión se modela como una estación más.
- (b) Satisfacción de restricciones, donde las variables son todas las aristas del grafo de conexiones entre las estaciones a recorrer, éstas son variables booleanas e indican si pertenecen al camino a recorrer o no. Las restricciones son que debe haber exactamente dos aristas de un mismo vértice en la solución y que no se sobrepase el número  $B$  de bicicletas que el camión puede llevar a la vez en el recorrido formado por las aristas.

Comenta cada una de las posibilidades indicando si resuelven o no el problema, qué errores te parece que tiene cada solución y cómo se podrían corregir, y qué ventajas e inconvenientes tienen cada una de ellas. Justifica la respuesta.

9. Los organizadores de la pasarela Gaudí nos piden un sistema inteligente que ayude a planificar los desfiles de la próxima edición. El sistema ha de organizar diez desfiles para diez diseñadores, dos por día en un total de cinco días, sabiendo que algunos diseñadores se niegan a desfilarse el mismo día que algún otro. Se han de asignar modelos a los desfiles ( $m$  por desfile) sin que la misma persona desfile más de tres veces en total durante la semana. Cada diseñador tiene también unas restricciones de altura y peso para sus modelos. Los organizadores de la pasarela han de pagar a los modelos y cada uno tiene su tarifa por participar en un desfile. Obviamente, los organizadores quieren pagar lo menos posible. Tras un análisis inicial del problema un compañero de nuestra empresa nos plantea dos estrategias distintas para resolverlo:

- (a) Usar el algoritmo de  $A^*$ . El estado es una asignación parcial de diseñadores y modelos a desfiles. Se definen dos operadores: **añadir\_diseñador** a un desfile concreto, que comprueba que el diseñador no se haya añadido antes y que en caso de ser el segundo diseñador asignado del día, no haya incompatibilidad entre ambos diseñadores, el coste de este operador es siempre 1; y **añadir\_modelo** a un desfile concreto, que comprueba que el desfile tenga ya diseñador asignado, que no haya más de  $m$  modelos en el desfile y que se cumplan las restricciones del diseñador, el coste de este operador es la tarifa del modelo asignado. Como función heurística usamos el número de diseñadores que falta incluir en los desfiles más el número de modelos que falta incluir en los desfiles multiplicado por la tarifa del modelo más caro.

- (b) Usar un algoritmo de satisfacción de restricciones. El grafo de restricciones tendría tres variables por cada uno de los modelos que representan los tres posibles desfiles que se le pueden asignar a cada modelo ( $desfile_1modelo_x$ ,  $desfile_2modelo_x$ ,  $desfile_3modelo_x$ ) y una variable por cada uno de los 10 diseñadores ( $diseñador_y$ ). El dominio de cada una de esas variables sería el identificador del desfile a asignar (un número entero entre el 1 y el 10, asumiendo que 1 y 2 son los desfiles del primer día, 3 y 4 son los desfiles del segundo día, y así sucesivamente). Como restricciones binarias habría: desigualdad entre todo par de variables de un mismo modelo  $x$  ( $desfile_zmodelo_x$  y  $desfile_wmodelo_x$ ) de modo que tengan asignado un desfile diferente; desigualdad entre todo par de variables  $diseñador_z$  y  $diseñador_w$  (cada diseñador ha de tener asignado un desfile diferente), y para cada modelo  $x$  que no cumple las restricciones de altura y peso de un diseñador  $y$ , establecer también una restricción binaria de desigualdad entre la variable  $diseñador_y$  y las tres variables del modelo  $x$  (el modelo  $x$  no puede coincidir en ninguno de sus desfiles asignados con el diseñador  $y$ ).

Comenta cada una de las posibilidades indicando si resuelven o no el problema, qué errores te parece que tiene cada solución y cómo se podrían corregir, y qué ventajas e inconvenientes tienen cada una de ellas. Justifica la respuesta.

10. Para hacer presión ante RENFE, los sindicatos de conductores de Cercanías en Barcelona quieren encontrar la forma de cumplir los requisitos mínimos que se les imponen sobre número de trenes, pero afectando al mayor número de usuarios, para que sus reclamaciones les den a ellos mayor poder en las negociaciones. Para ello han construido una tabla que, para cada hora y línea de cercanías, les dice el número de usuarios que viajan a esa hora por esa línea (es decir, la demanda), y el número mínimo de trenes que según RENFE han de pasar durante esa hora por esa línea para cumplir los servicios mínimos. Sabemos también que cada tren puede llevar como máximo  $P$  pasajeros. Hay además otras reglas que han de cumplir, y es que RENFE impone un número mínimo  $L_i$  total de trenes que han de circular por la línea  $i$  cada día, y un número mínimo  $H_h$  total de trenes que han de circular en una hora  $h$ , siendo estos números algo mayores que la suma de los trenes por líneas o por horas de los servicios mínimos antes mencionados.

Se nos plantean las siguientes alternativas:

- (a) Queremos utilizar satisfacción de restricciones donde tenemos una variable por cada línea de cercanías y cada hora, y los valores son el número de trenes asignados a cada línea y cada hora. Las restricciones son el número mínimo de trenes para cada línea y hora, el número mínimo de trenes que han de circular para cada línea durante el día, el número mínimo de trenes que han de circular para cada hora y que el número total de usuarios que se queden sin tren sea mayor que un cierto valor  $U$ .
- (b) Queremos utilizar búsqueda local, donde se genera una solución inicial colocando suficientes trenes para cubrir los servicios mínimos de cada línea y hora, y asignando aleatoriamente los trenes restantes entre todas las líneas. Los operadores de modificación de la solución consisten en mover un tren de una hora a otra en la misma línea, y mover un tren de una línea a otra. Queremos maximizar el número de usuarios que se verán afectados por la falta de trenes.

Comenta cada una de las posibilidades indicando si resuelven o no el problema y qué ventajas e inconvenientes tiene cada una de ellas. Justifica la respuesta.

11. Los organizadores de un congreso internacional nos piden un sistema inteligente de planificación de los diferentes seminarios que se celebrarán en el evento. El objetivo del sistema es asignar un conjunto de seminarios a una lista de aulas. Para cada seminario tenemos la fecha tentativa en la que debería realizarse, pero sabemos que tenemos un margen de hasta tres días que podemos usar para retrasar su inicio. También disponemos de la duración del seminario (1, 2 o 3 horas).

Para cada aula sabemos en qué fecha está disponible y durante cuántas horas (1, 2 o 3 horas). Un aula sólo se puede reservar para todo el periodo durante el que está disponible, por lo que si el seminario dura menos estaremos perdiendo horas.

El objetivo sería asignar los seminarios a las aulas de manera que se minimice el número de horas desperdiciadas y que los seminarios se retrasen lo mínimo posible. Supondremos que tenemos aulas suficientes para asignar todos los seminarios y que hay muchas más aulas que seminarios. Se nos plantean las siguientes formas de solucionar el problema:

- (a) Queremos utilizar  $A^*$  de manera que ordenamos las aulas según la fecha en las que están disponibles (en caso de estar disponibles en la misma fecha se ordenan poniendo primero las aulas con menos horas disponibles ese día). Procedemos a asignar los seminarios siguiendo el orden establecido utilizando dos operadores: asignar un seminario que quepa en las horas disponibles y que no viole las restricciones de fecha de inicio (el coste sería las horas disponibles del aula) o no asignar nada al aula (el coste sería cero). Como función heurística utilizaremos la suma de horas de seminario que quedan por asignar.
- (b) Queremos usar satisfacción de restricciones para resolver el problema, para ello elegimos como variables los seminarios y para cada seminario calculamos su dominio seleccionando todas las aulas en las que puede ubicarse cumpliendo las restricciones de fechas y de horas necesarias. Añadimos dos restricciones adicionales al problema, la primera es que la suma de retrasos de los seminarios ha de ser menor que el valor  $R$  y que la diferencia entre las horas que dura cada seminario y las horas libres del aula en el que es ubicado ha de ser menor o igual que uno.

Comenta cada una de las posibilidades indicando si resuelven o no el problema, qué errores te parece que tiene cada solución y cómo se podrían corregir, y qué ventajas e inconvenientes tienen cada una de ellas. Justifica la respuesta.

12. Con la introducción de la telefonía 4G es necesario hacer un uso más inteligente del limitado rango de frecuencias en el que operan. Para ello se necesita de un sistema dinámico que asigne a los teléfonos que están llamando en cierto momento la frecuencia en la que tienen que operar y la antena a través de la que tienen que comunicarse. Tenemos un conjunto de  $A$  antenas de telefonía, cada antena tiene asignadas  $f$  frecuencias que puede utilizar. Dos teléfonos móviles pueden operar en la misma frecuencia si la distancia que los separa es mayor que  $d$ . Una antena no puede manejar más de  $t$  teléfonos o  $tf$  teléfonos en la misma frecuencia, ni tampoco teléfonos que estén a una distancia de más de  $da$  de la antena. Estas podrían ser soluciones a este problema:

- (a) Se plantea solucionarlo mediante  $A^*$ . Consideramos que el estado es una asignación total o parcial de antena y frecuencia a los teléfonos móviles. Buscamos la secuencia de asignaciones que da un valor a todos los teléfonos móviles. Como operador usamos asignar una antena y una frecuencia a un teléfono siempre que la asignación no provoque interferencia por su distancia con otro teléfono ya asignado y no supere los límites de número de teléfonos por antena y máximo de teléfonos con la misma frecuencia de la antena. El coste es la distancia del teléfono a la antena asignada. La función heurística es la suma de las distancias de los teléfonos por asignar a su antena más cercana.
- (b) Se plantea resolverlo como un problema de satisfacción de restricciones. Consideramos que los teléfonos móviles son variables y la frecuencia y la antena que se les ha de asignar son los dominios. Tenemos un grafo de restricciones que conecta a los teléfonos de manera que dos teléfonos no pueden tener la misma frecuencia si su distancia es menor que  $d$ . Tenemos una restricción global que impide que asignemos la misma antena a más de  $t$  teléfonos y antes de iniciar la solución del problema eliminamos del dominio de cada teléfono todas las antenas que están a una distancia de más de  $da$ .

Comenta cada una de las posibilidades indicando si resuelven o no el problema, qué errores te parece que tiene cada solución y cómo se podrían corregir, y qué ventajas e inconvenientes tienen cada una de ellas. Justifica la respuesta.

13. Un problema logístico en el 9-N es asegurar transporte para los voluntarios que no disponen de vehículo propio y que han de desplazarse por zonas del territorio con pocas opciones de transporte público. Por

ello varias asociaciones del Alt Urgell han alquilado de forma conjunta un autocar con 35 plazas, y nos han pedido crear un programa que, dada la información sobre el punto de recogida de cada voluntario y su punto de destino (ambos dentro de la comarca) calcule el recorrido más corto posible que debe hacer el camión para recoger y dejar a todos los voluntarios, sin que se sobrepase en ningún momento el número de plazas disponibles en el autocar.

El autocar parte desde cierto punto de origen y ha de volver a él al final del recorrido, habiendo dejado todos los voluntarios en sus puntos de destino. En ningún punto de recogida habrá más de 35 voluntarios por recoger. Para obtener el recorrido se dispone de un mapa de la comarca que indica la longitud del camino mínimo entre cada dos puntos por los que ha de pasar el autocar.

- (a) Queremos utilizar  $A^*$ . Usaremos como coste la longitud del camino. La función heurística vale infinito si el autocar supera el número máximo de pasajeros y en caso contrario, es la suma de las distancias de los puntos por recorrer al origen. El operador aplicable es pasar del punto actual a otro no visitado (se supone que en cada punto visitado se recoge o deja a todos los voluntarios que toca, no se puede recoger solo a una parte de los voluntarios y volver más tarde a por el resto).
- (b) Aplicar un algoritmo de satisfacción de restricciones. Las variables son todas las aristas del grafo de conexiones entre los puntos a recorrer, éstas son booleanas e indican si pertenecen al camino a recorrer o no. Las restricciones son que debe haber exactamente dos aristas de un mismo vértice en la solución y que no se sobrepase el número máximo de pasajeros del autocar en el recorrido.

Comenta cada una de las posibilidades indicando si resuelven o no el problema, qué errores te parece que tiene cada solución y cómo se podrían corregir, y qué ventajas e inconvenientes tienen cada una de ellas. Justifica la respuesta.

14. Tras las numerosas críticas y movilizaciones de los taxistas por el mecanismo de turnos que ha implantado recientemente el *Ajuntament de Barcelona* para el sector, un grupo de trabajo formado por representantes de los taxistas y del *Ajuntament* nos ha pedido crear un sistema que adapte el número de taxis circulando por el área metropolitana a la demanda estimada y al nivel de contaminación diaria (medida en gr. de  $\text{CO}_2$  emitidos a la atmósfera).

Según nos cuentan, en estos momentos hay un número  $T$  de taxis registrados en el área metropolitana. Nos han contado que en vez de controlar el número de pasajeros que lleva cada taxi, el sistema controlará el número de servicios (un servicio es un viaje del taxi que recoge a un cierto número de pasajeros en un lugar de origen y los deja en un lugar de destino). Para simplificar el problema consideraremos que cada servicio dura unos 15 minutos de media. Se establece que el sistema asignará a cada taxi como mínimo  $NS$  servicios a la semana. Para cada taxi  $t$  disponemos del valor  $CO_t$  que mide el grado de contaminación que produce (medido en gr. de  $\text{CO}_2$  emitidos por minuto). El *Ajuntament* nos dice que en ningún momento los taxis en servicio han de superar la cota máxima  $MAX\_CO\_H$  de gramos de  $\text{CO}_2$  generados en una hora, y que en un día la cantidad de gramos de  $\text{CO}_2$  emitidos no puede superar la cota  $MAX\_CO\_D$ . Con el número  $T$  de taxis disponibles es fácil superar ambas cotas. zona, tenemos una tabla semanal en la que, para cada zona  $z$ , día  $d$  y hora  $h$  se tiene una estimación del número de servicios ( $S_{z,d,h}$ ) que se suelen originar en esa zona, para cada hora del día. Para modelar la demanda esperada de taxis tenemos una tabla semanal en la que, para cada día  $d$  y hora  $h$  se tiene una estimación del número de servicios ( $S_{d,h}$ ) que se suelen realizar.

El objetivo es decidir de forma dinámica los taxis que han de circular por el área metropolitana, de forma que se minimicen los niveles de contaminación diaria (nunca se pueden superar los límites) y se maximice el número de servicios realizados (pero podemos dejar usuarios sin servir, si añadir servicios implica superar los límites de contaminación). Se nos plantean las siguientes alternativas:

- (a) Queremos utilizar  $A^*$ . Definimos el estado como la asignación de taxis a horas del día. El estado inicial consiste en asignar de forma ordenada a cada taxi un número mínimo de servicios  $NS$  distribuidos de forma aleatoria en las horas de la semana. Tenemos un operador que asigna un nuevo servicio a un taxi  $t$  en un día  $d$  y hora  $h$  determinados, el coste serán los gramos  $CO_t$  emitidos en los 15 minutos del servicio. Como función heurística usamos la suma del número de

servicios estimados en la tabla de demanda y que aun no hemos servido, o infinito si no se supera la cota  $S_{d,h}$ .

- (b) Queremos utilizar Hill Climbing, donde se genera una solución inicial asignando al azar suficientes taxis a servicios hasta llegar a servir toda la demanda estimada en cada hora. Los operadores de modificación de la solución consisten en intercambiar dos taxis entre servicios que pertenezcan a horas diferentes, y eliminar un servicio asignado a un taxi (siempre que no quedemos por debajo del límite  $NS$  de servicios mínimos asignados a ese taxi). Como función de evaluación usaremos el sumatorio de los gramos de  $CO_2$  emitidos por todos los taxis asignados a servicios en la solución actual.

Comenta cada una de las posibilidades indicando si resuelven o no el problema y qué ventajas e inconvenientes tiene cada una de ellas. Justifica la respuesta.

15. La cadena de comida rápida a domicilio ComidApp ha decidido renovar su flota de motos de reparto por motos 100% eléctricas pero después de unas pruebas ha visto que necesita hacer una mejor gestión de los trayectos de entrega para reducir la distancia recorrida y sacar el máximo partido de la limitada autonomía de las motos. Para ello nos piden un sistema que decida que moto y en que orden ha de servir los pedidos realizados por los clientes desde sus casas con una app. Esta app ya envía directamente el pedido (con su localización GPS) a la tienda más cercana al cliente.

Cada tienda tiene 4 motos eléctricas para entregar los pedidos asignados a su tienda. Cada moto puede llevar un máximo de 5 pedidos a la vez. Dado que no pueden tener las motos paradas durante horas cargando sus baterías al máximo, después de hacer pruebas han visto que lo mejor es hacer cargas rápidas de 10 minutos cada vez que la moto vuelve a la tienda, dando así una autonomía máxima de 8 km. Disponemos de una función `distancia(loc_1, loc_2)` que es capaz de calcular la longitud (en km) de la ruta más corta entre dos localizaciones GPS (tanto de clientes como de la tienda). Nos piden un sistema que, cada 30 minutos, asigne los pedidos recibidos por la tienda a las motos. Cada moto recibe una lista de pedidos a servir con el orden de entrega, y debe tener en cuenta que la longitud de la ruta (de la tienda al primer cliente, del primer cliente al segundo, del segundo al tercero... y del último a la tienda) no supere la autonomía máxima. Toda solución propuesta ha de entregar todos los pedidos.

- (a) Queremos utilizar  $A^*$  para minimizar el número de kilómetros recorridos por las motos para asegurar que entregan todos los pedidos sin superar su autonomía máxima. El operador es asignar un pedido de la tienda a una moto (el orden de asignación determina el orden de entrega), el coste de ese operador es la distancia desde la localización donde se encuentra la moto a la localización de entrega de ese pedido (usando la función `distancia(loc_1, loc_2)` antes mencionada). La función heurística vale infinito si a alguna de las motos se le asigna más de 5 pedidos o si alguna de las motos no tiene energía suficiente para volver desde la localización actual a la tienda, y en caso contrario es la suma de las distancias de las localizaciones de los pedidos por entregar a la localización de la moto más cercana en cada caso.
- (b) Queremos utilizar satisfacción de restricciones para encontrar una asignación correcta de pedidos a motos. Asumiendo que cada moto puede llevar como máximo 5 pedidos, tenemos, para cada moto, cinco variables que representan el primer/segundo/tercer/cuarto/quinto pedido a entregar, el dominio para todas esas variables es la lista de identificadores de todos los pedidos recibidos por la tienda. Tenemos una restricción binaria  $\neq$  entre todo par de variables que impide que el mismo identificador de pedido se asigne a 2 o más variables a la vez. Tenemos una restricción  $n$ -aria entre las 5 variables de cada moto que impide que la longitud de la ruta necesaria para entregar los pedidos exceda la autonomía máxima de esa moto.

Comenta cada una de las posibilidades indicando si resuelven o no el problema, qué errores te parece que tiene cada solución y cómo se podrían corregir, y qué ventajas e inconvenientes tienen cada una de ellas. Justifica la respuesta.

16. Tenemos un sistema P2P que utiliza un mecanismo centralizado para asignar a cada cliente qué otros clientes son los que le envían las partes del fichero que le faltan. Cada cliente calcula una lista con los

retardos medios de transmisión a cada uno de los clientes que conoce (en milisegundos). El mecanismo centralizado conoce el ancho de banda disponible de cada cliente tanto de subida como de bajada (en Kb/s) para el fichero que se quiere transmitir. Cada cierto tiempo el mecanismo centralizado distribuye a los clientes con qué otros clientes debe conectarse para recibir partes del fichero y qué ancho de banda dedicar. Para cada cliente conocemos qué partes del fichero tiene, por lo que podemos saber si puede enviar o no a un cliente. La idea es que minimicemos el tiempo de retardo total de las transmisiones y utilicemos el máximo ancho de banda de bajada disponible de cada cliente.

Comenta cada una de las posibilidades indicando si resuelven o no el problema, qué errores te parece que tiene cada solución y cómo se podrían corregir, y qué ventajas e inconvenientes tienen cada una de ellas. Justifica la respuesta.

- (a) Queremos utilizar  $A^*$  de manera que recorremos la lista de clientes en un orden preestablecido. El estado es la asignación que hemos hecho de clientes y sus anchos de banda a los clientes recorridos. Utilizamos como operador asignar a un cliente uno de los que conoce (siempre que tenga partes del fichero que el cliente actual no tenga) y su máximo ancho de banda de subida al cliente actual, cuando el ancho de banda de bajada del cliente actual es superado por la suma de los anchos de banda de subida de los clientes asignados pasamos al siguiente cliente. Evidentemente una vez asignado un cliente para transmitir partes del fichero no lo podemos asignar más veces. El coste del operador es el retardo del cliente asignado. La función heurística es la suma para los clientes que quedan por recorrer de los retardos a los clientes que conocen.
  - (b) Queremos utilizar búsqueda local generando una solución inicial en la que cada cliente recibe de todos los clientes que conoce que tienen partes del fichero que le faltan con un ancho de banda de 1 Kb/s. Como operadores tenemos aumentar o disminuir el ancho de banda de un cliente que transmite a otro en 1 Kb/s. La función heurística es la suma para cada cliente de los retardos de los clientes que le transmiten con un ancho de banda superior a 0 Kb/s.
17. Se quiere planificar cómo componer  $S$  servicios Web en un único servicio de orden superior (meta-servicio). Cada servicio Web usa un conjunto de agentes informáticos que deben ejecutarse en un orden específico para cumplir la tarea que realiza el servicio, estos agentes pueden trabajar en paralelo. Se supone que la acción que realiza cada agente tiene la misma duración (un paso) y hay que tener en cuenta que un servicio puede necesitar un mismo agente en diferentes pasos de su ejecución. Se dispone de un agente de cada tipo, teniendo un total de  $A$  agentes. El meta-servicio se considera completo cuando se haya completado cada servicio que lo compone. Se plantean las siguientes alternativas para minimizar el número total de pasos de ejecución del meta-servicio:
- (a) Queremos utilizar  $A^*$ . Definiremos el estado como la asignación de pasos de los  $S$  servicios individuales a uno de los  $A$  agentes en cada paso del meta-servicio. El estado inicial es tener un único paso del meta-servicio donde ninguno de los agentes tiene un servicio asignado. Los operadores de cambio de estado consisten en:
    - i. Asignar el primer paso no ejecutado de alguno de los servicios a un agente libre en el paso actual del meta-servicio, con coste uno
    - ii. Añadir un paso nuevo al meta-servicio, con coste uno
 La función heurística es la suma de pasos de los servicios individuales que nos quedan por ejecutar dividida por el número de agentes.
  - (b) Queremos utilizar satisfacción de restricciones. Suponemos que el número máximo de pasos del meta servicio ( $MP$ ) es el número de veces que aparece el agente más utilizado, de manera que usamos  $S \cdot MP$  variables para representar qué agente ejecuta un paso de un servicio en la secuencia de pasos del metaservicio. El dominio de cada variable son los  $A$  agentes, mas un valor que indica que la variable no esta asignada. Las restricciones son las siguientes:
    - i. Para las variables de servicio en un paso, estas no pueden tener el mismo agente.
    - ii. Para las variables de un mismo servicio, estas no pueden violar la secuencia de acciones del servicio

Comenta cada una de las posibilidades indicando si resuelven o no el problema, qué errores te parece que tiene cada solución y cómo se podrían corregir, y qué ventajas e inconvenientes tienen cada una de ellas. Justifica la respuesta.

18. El Servei Català de la Salut (CatSalut) quiere mejorar el sistema de asignación de plazas de médicos a centros hospitalarios catalanes. Cada médico  $m_i$  tiene una especialidad (medicina general, pediatría, traumatología, dermatología, odontología, cardiología, oncología, ...), acepta un sueldo mínimo determinado ( $Sm_i$ ) y acepta una distancia máxima a recorrer entre su casa y su trabajo ( $Dm_i$ ). Cada centro  $c_j$  dispone de  $x_e$  plazas nuevas por cada especialidad  $e$ . Para cada plaza de especialidad  $e$  el CatSalut determina un sueldo máximo ( $S_e^j$ ) que es diferente para cada centro  $c_j$ , y dispone de una función `float distancia(Medico m, Centro c)` que devuelve la distancia que hay (en km) entre la vivienda de un médico  $m$  y un centro  $c$  dados.

Se quiere asignar médicos a centros de manera que se minimicen las distancias que los médicos deban recorrer cada día (para mejorar su rendimiento), se maximice el ahorro económico respecto a lo que inicialmente está dispuesto a pagar CatSalut por plaza ( $S_e^j - Sm_i$ ) y cubra el máximo número de plazas nuevas, priorizando este último criterio respecto a los dos anteriores. Supondremos que el número de médicos que solicitan plaza es mucho mayor que el número de plazas, pero las restricciones de especialidad, sueldo y distancia pueden dejar plazas nuevas sin asignar.

Se nos plantean las siguientes formas de solucionar automáticamente este problema:

- Queremos usar  $A^*$  definiendo el estado como la asignación de médicos a plazas nuevas. El estado inicial es la asignación vacía. Como operadores tenemos el de asignar un médico a una plaza (si las especialidades coinciden y si el sueldo y la distancia son convenientes), el coste de este operador es  $Sm_i$  (el sueldo mínimo del médico asignado), y desasignar un médico de una plaza, cuyo coste es  $-Sm_i$ . La función heurística  $h$  que se pretende usar es el sumatorio de los sueldos máximos determinados ( $S_e^j$ ) para las plazas que quedan por asignar.
- Queremos usar satisfacción de restricciones donde las variables son las plazas nuevas por cubrir, y sus dominios son el conjunto de médicos solicitantes  $m_i$  que tienen la especialidad adecuada para cada una de las plazas. Tenemos un grafo de restricciones que conecta las plazas de manera que 2 plazas no puedan tener el mismo médico asignado. También se añaden como restricciones que el sueldo máximo determinado para la plaza sea mayor que el sueldo mínimo del médico asignado, y que la distancia a recorrer por el médico asignado sea menor que su distancia máxima. Tenemos además una restricción global que impone que el ahorro económico sea mayor que un valor  $minA$  dado.

Comenta cada una de las posibilidades indicando si resuelven o no el problema, qué errores te parece que tiene cada solución y cómo se podrían corregir, y qué ventajas e inconvenientes tienen cada una de ellas. Justifica la respuesta.

19. Queremos transmitir  $X$  Kbits entre dos máquinas  $M_I$  y  $M_F$ , para ello hemos de establecer  $C$  canales de comunicación que han de atravesar  $N$  máquinas ( $M_1, M_2, \dots, M_N$ ) que hacen de puntos intermedios ( $C \leq N$ ), los  $X$  Kbits los dividimos a partes iguales por cada canal. Conocemos la velocidad en *kbits* del canal que se puede establecer entre cada par de máquinas (incluidas  $M_I$  y  $M_F$ ), queremos que la velocidad media de un canal sea como mínimo  $V$  Kbps y que un canal no atraviere más de  $P$  máquinas.

Buscamos la solución que conecte todas las máquinas sin que dos canales pasen por la misma máquina (exceptuando  $M_I$  y  $M_F$ ) tardando lo menos posible en transmitir los  $X$  Kbits.

- Queremos utilizar  $A^*$ . Definimos el estado como la asignación de máquinas a canales. El estado inicial consiste en asignar las máquinas  $M_I$  y  $M_F$  a cada canal. Tenemos un operador que asigna una máquina a un canal y cuyo coste es la velocidad media del canal que empieza en  $M_I$ , pasa por la máquina y acaba en  $M_F$ , solo podemos asignar una máquina a un canal si no excedemos el valor  $P$ . Como función heurística usamos la suma de las velocidades medias de todos los canales que se obtienen poniendo cada máquina que queda por conectar como único nodo intermedio entre  $M_I$  y  $M_F$ .

- (b) Aplicar un algoritmo de satisfacció de restriccions. Las variables son las máquinas, los dominios son los canales donde podemos asignarlas. Supondremos que no tenemos en cuenta las máquinas  $M_I$  y  $M_F$  ya que estarán asignadas a todos los canales. Como restricciones imponemos que un canal no esté asignado a más de  $P$  máquinas y que la velocidad media mínima del canal que incluye las máquinas asignadas sea mayor que  $V$ .

Comenta cada una de las posibilidades indicando si resuelven o no el problema, qué errores te parece que tiene cada solución y cómo se podrían corregir, y qué ventajas e inconvenientes tienen cada una de ellas. Justifica la respuesta.

20. Una empresa pesquera y conservera envía sus  $N$  barcos  $\{b_i\}$  con capacidades para  $C(b_i)$  pescadores a sus  $D$  destinos internacionales  $\{d_i\}$  ( $N > D$ ) para 180 días de pesca. Algunos de sus  $M$  pescadores  $\{p_i\}$  ( $M > \sum C(b_i)$ ) son asignados a los barcos y son enviados a los destinos en el barco asignado. El coste por marinero del viaje en el barco  $b_i$  a un destino  $d_j$  es  $K_b(d_j, b_i)$ . Cada pescador  $p_i$  proporciona a la empresa una ganancia de  $g(p_i)$  euros/día en media y cuesta a la empresa  $k(p_i)$  euros/día en media. La empresa requiere una planificación que cubra todos sus destinos sin pérdidas.

En los siguientes apartados se proponen diferentes alternativas para algunos de los elementos necesarios para plantear la búsqueda (solución inicial, operadores, función heurística). Comenta la solución que se propone respecto a si es correcta, es eficiente, y es mejor o peor en comparación con otras alternativas. Justifica todas las respuestas.

- (a) Aplicar búsqueda local. La solución inicial consiste en: 1) asignar secuencialmente  $N \text{ div } D$  barcos a cada destino, repartiendo los  $N \text{ mod } D$  restantes equitativamente entre los primeros  $N \text{ mod } D$  destinos; 2) asignar aleatoriamente  $C(b_i)$  pescadores a cada barco  $b_i$ . Los operadores son: cambiar el destino de un barco (y el de todos sus pescadores) e intercambiar dos pescadores entre dos barcos de destinos diferentes. Como función heurística usamos:

$$h'(n) = \sum_{i=1}^D \sum_{b_j \in \text{Asig}(d_i)} K_b(d_i, b_j) + 180 \times \sum_{p_i \in O(b_j)} (k(p_i) - g(p_i))$$

donde  $O(b_j)$  es la asignación de marineros del barco  $b_j$  ( $|O(b_j)| \leq C(b_j)$ ) y  $\text{Asig}(d_i)$  son los barcos asignados al destino  $d_i$ .

- (b) Aplicar satisfacción de restricciones. Las variables son todos los posibles pares  $\langle p_i, d_j \rangle$  (pescador, destino). Todas las variables tienen como dominio el conjunto  $\{b_i, \text{undef}\}$ . El primer valor denota que el barco  $b_i$  se asigna al destino y que el pescador llegará en él. El segundo valor denota que el pescador no viaja al destino. Las restricciones son:

$$R1 : \forall_i |O(b_i)| \leq C(b_i) \qquad R2 : \sum_{j=1}^N \sum_{p_i \in O(b_j)} (k(p_i) - g(p_i)) \geq 0$$





# Sistemes Basats en el Coneixement

7	Sistemes de producció .....	67
8	Enginyeria del Coneixement .....	77
9	Xarxes Bayesianes .....	107



## 7. Sistemes de producció

- Utilitzant les regles i fets que apareixen a continuació, mostreu com un sistema de producció amb encadenament cap enrera (seguint una estratègia seqüencial en la selecció tant de fets i regles com d'objectius), intentaria demostrar  $R(a)$ .

Base de hechos		Base de reglas
F1. $P(b)$	F5. $S(d,b)$	R1. $P(x) \wedge P(y) \wedge S(x,y) \rightarrow R(y)$
F2. $U(b)$	F6. $Q(a,c)$	R2. $U(x) \wedge T(y) \rightarrow Q(x,y)$
F3. $Q(c,d)$		R3. $Q(x,y) \rightarrow S(y,x)$
F4. $P(a)$		R4. $S(z,x) \wedge S(z,y) \rightarrow Q(x,y)$

- Donades les següents bases de regles i de fets:

Base de hechos		Base de reglas
F1. $P(b)$	F4. $P(c)$	R1. $S(x,y) \rightarrow Q(y,x)$
F2. $S(a,b)$	F5. $Q(a,d)$	R2. $Q(x,z) \wedge Q(z,y) \rightarrow S(x,y)$
F3. $Q(a,c)$		R3. $P(x) \wedge P(y) \wedge Q(x,y) \rightarrow R(x)$

- Seguint encadenament cap endavant, mostreu com un sistema de producció amb estratègia seqüencial en la selecció de les regles arribaria a  $R(c)$ .
  - Fer el mateix considerant que el criteri de resolució de conflictes escull la regla amb més condicions.
- Considera la siguiente base de hechos (donde cada hecho viene acompañado por el momento de su creación) y la siguiente base de reglas:

Base de hechos		Base de reglas
H1. $A(e)$	H4. $P(n)$	R1. $A(x) \wedge A(y) \wedge C(x,y) \rightarrow L(x,y)$
H2. $D(e)$	H5. $A(s)$	R2. $L(x,y) \wedge L(y,z) \rightarrow L(x,z)$
H3. $A(n)$	H6. $C(n,s)$	R3. $D(x) \wedge P(y) \rightarrow C(x,y)$

- Considera el objetivo  $L(e,s)$  y comprueba si es deducible aplicando razonamiento hacia atrás. Utiliza la estrategia de resolución de conflictos: 1º) subobjetivo más reciente, 2º) primer subobjetivo por la izquierda, 3º) regla más específica 4º) hechos en secuencia. Indica claramente para cada iteración el conjunto conflicto, la selección realizada y el resultado de la aplicación de la regla.

- (b) Considera ahora el objetivo  $L(n,e)$  y comprueba si es deducible aplicando razonamiento hacia delante. Utiliza la estrategia de resolución de conflictos: 1º) regla más general, 2º) instanciación más antigua, 3º) hecho más antiguo en la primera condición. Indica claramente en cada iteración el conjunto de conflicto, la selección realizada y el resultado de la aplicación de la regla. ¿Una estrategia distinta daría mejores resultados?

4. Tenemos la siguiente base de reglas y la siguiente base de hechos:

Base de hechos	Base de reglas
H1. $P(a)$	R1. $P(x) \wedge R(y) \rightarrow Q(x,y)$
H2. $P(b)$	R2. $Q(x,y) \rightarrow Q(y,x)$
H3. $R(a)$	R3. $P(x) \wedge P(y) \rightarrow R(y)$
	R4. $Q(x,y) \wedge R(y) \rightarrow P(x)$

- (a) Realiza los 4 primeros pasos del razonamiento hacia delante con la estrategia de resolución de conflictos 1) primera regla en secuencia 2) instanciación con hechos más antiguos 3) condición de más a la izquierda con hecho más antiguo. Especifica claramente para cada iteración el conjunto de conflicto, la selección realizada y el resultado de la aplicación de la regla
- (b) Realiza los 4 primeros pasos del razonamiento hacia delante con la estrategia de resolución de conflictos 1) regla más específica 2) instanciación con hechos más recientes 3) condición de más a la izquierda con hecho más reciente. Especifica claramente para cada iteración el conjunto de conflicto, la selección realizada y el resultado de la aplicación de la regla
- (c) Realiza los 4 primeros pasos del razonamiento hacia delante con la estrategia de resolución de conflictos 1) regla más general 2) instanciación con hechos más antiguos 3) condición de más a la izquierda con hecho más antiguo. Especifica claramente para cada iteración el conjunto de conflicto, la selección realizada y el resultado de la aplicación de la regla
- (d) Realiza los 10 primeros pasos del razonamiento hacia atrás para resolver el objetivo  $Q(b,c)$  utilizando como estrategia de resolución de conflictos 1) Objetivos en secuencia 2) reglas en secuencia 3) hechos más recientes. Especifica claramente para cada iteración el objetivo, el conjunto de conflicto, la selección realizada y el resultado de la aplicación de la regla

5. Tenemos la siguiente base de reglas y la siguiente base de hechos:

Base de hechos	Base de reglas
H1. $P(a,c)$	R1. $P(x,y) \wedge R(y) \rightarrow Q(x)$
H2. $Q(c)$	R2. $Q(x) \rightarrow R(x)$
H3. $R(d)$	R3. $Q(x) \rightarrow P(x,y)$
H4. $P(c,d)$	R4. $Q(x) \wedge P(x,y) \rightarrow R(y)$
	R5. $P(x,y) \wedge P(y,z) \rightarrow R(y)$

- (a) Realiza los 5 primeros pasos del razonamiento hacia delante con la estrategia de resolución de conflictos 1) primera regla en secuencia 2) instanciación con hechos más recientes 3) condición de mas a la izquierda con hecho más antiguo. Especifica claramente para cada iteración el conjunto de conflicto, la selección realizada y el resultado de la aplicación de la regla
- (b) Realiza los 5 primeros pasos del razonamiento hacia delante con la estrategia de resolución de conflictos 1) regla más específica 2) instanciación con hechos más antiguos 3) condición de mas a la izquierda con hecho más antiguo 4) primera regla en secuencia. Especifica claramente para cada iteración el conjunto de conflicto, la selección realizada y el resultado de la aplicación de la regla
- (c) Realiza los 5 primeros pasos del razonamiento hacia delante con la estrategia de resolución de conflictos 1) regla más general 2) instanciación con hechos más recientes 3) condición de mas a la izquierda con hecho más recientes 4) primera regla en secuencia. Especifica claramente para cada iteración el conjunto de conflicto, la selección realizada y el resultado de la aplicación de la regla

- (d) Realiza la traza del razonamiento hacia atrás para intentar resolver el objetivo R(b) utilizando como estrategia de resolución de conflictos 1) objetivos en secuencia 2) hechos mas recientes 3) regla más general 4) primera regla en secuencia. Especifica claramente para cada iteración el objetivo, el conjunto de conflicto, la selección realizada y el resultado de la aplicación de la regla
6. Dados la siguiente base de hechos (cada hecho viene precedido por el tiempo en que ha sido establecido) y base de reglas:

Base de hechos	Base de reglas
H1. P(a)	R1. $P(a) \wedge Q(e) \rightarrow Q(b)$
H2. Q(c)	R2. $P(c) \wedge R(x) \wedge (x < 15) \wedge (y = x - 5) \rightarrow R(y)$
H3. R(10)	R3. $Q(b) \wedge R(x) \wedge (x < 5) \rightarrow S(e)$
H4. P(c)	R4. $Q(x) \rightarrow Q(e)$

- (a) Dado el objetivo S(e) y suponiendo un mecanismo de razonamiento hacia delante con estrategia de resolución de conflictos: 1º) regla más específica y 2º) instanciación más reciente,
- Indica las 3 primeras iteraciones en la aplicación de las reglas, mostrando para cada una de ellas el conjunto de conflictos, la selección realizada y el resultado de la aplicación.
  - Si detectas algún problema, propón alguna solución.
- (b) Dado el objetivo R(0) y suponiendo un mecanismo de encadenamiento hacia atrás con estrategia de resolución de conflictos: 1º) regla más específica, 2º) subobjetivo más simple y 3º) hechos más antiguos,
- Indica las 3 primeras iteraciones mostrando para cada una de ellas el conjunto de conflictos, la selección realizada y el resultado de la aplicación.
  - ¿Qué pasaría si el objetivo fuera R(20)?
7. Supposeu que tenim la següent de fets (fets precedits per l'identificador temporal del moment de creació) i Base de regles:

Base de hechos	Base de reglas
F1. P(a)	R1. $P(x) \wedge C(x,y) \rightarrow S(x)$
F2. Q(c)	R2. $S(x) \wedge C(x,y) \rightarrow R(x)$
F3. S(a)	R3. $S(x) \rightarrow C(y,x)$
F4. P(c)	R4. $P(x) \rightarrow C(x,y)$

- (a) Donat l'objectiu C(a,x) i suposant estratègia de raonament cap endavant amb els criteris de resolució de conflictes: 1er.) regla més específica i 2n.) instanciació més recent, mostra les quatre primeres iteracions d'aplicació de les regles. Mostra explícitament cadascun dels conjunts de conflictes que hi ha, la regla finalment seleccionada a cada pas i el resultat d'aplicar-la.
- (b) Donat l'objectiu  $(C(x,b) \wedge R(a))$  i suposant estratègia de raonament cap enrera amb criteris de resolució de conflictes: 1er.) suobjectiu (predicat) amb menys paràmetres i 2n.) ordre seqüencial de les regles, mostra les cinc primeres iteracions d'aplicació de les regles. Mostra explícitament cadascun dels conjunts de conflictes que hi ha, els subobjectius pendents a cada iteració, la regla finalment seleccionada a cada pas i el resultat d'aplicar-la.
8. Es frecuente encontrar en las secciones de pasatiempos de los periódicos cosas como "dado un número, normalmente alto, ¿cómo puede expresarse a través de una expresión aritmética sobre un conjunto de números básico?". Este ejercicio se basa en la misma idea: Considerése el siguiente conjunto de hechos y reglas:

Base de hechos	Base de reglas
H1. par(0)	R1. $\text{par}(x) \rightarrow \text{par}(x+2)$
H2. num(1)	R2. $\text{par}(x) \wedge x > 0 \rightarrow \text{par}(x-2)$
H3. num(17)	R3. $\text{num}(x) \wedge \text{num}(y) \rightarrow \text{num}(x+y)$
	R4. $\text{num}(x) \wedge \text{num}(y) \rightarrow \text{num}(x*y)$
	R5. $\text{num}(x) \wedge \text{par}(x) \rightarrow \text{num}(x/2)$
	R6. $\text{num}(x) \rightarrow \text{num}(x*x)$
	R7. $\text{num}(x) \wedge \text{num}(y) \wedge x > y \rightarrow \text{num}(x-y)$

Suponemos que los valores que pueden recibir las variables son números naturales menores que 100. El objetivo es num(9). Se pide:

- (a) Considerar un sistema de razonamiento dirigido por hechos en el que la resolución de conflictos se lleve a cabo de acuerdo a los siguientes criterios:
  - i. Tendrá prioridad la regla más compleja.
  - ii. A igualdad de a) tendrá prioridad la instanciación con hechos más antiguos en sus premisas, considerando las condiciones de las premisas de izquierda a derecha.
  - iii. A igualdad de b) tendrá prioridad la primera regla en secuencia.

Efectuar las 3 primeras iteraciones del proceso detallando en cada iteración explícitamente y completamente el conjunto de conflicto, la regla escogida y el resultado de su aplicación.

- (b) ¿Se ha llegado a la solución? En caso contrario, a la vista de como se van aplicando las reglas, realizar una estimación razonada de en cuántas iteraciones se llegaría a la solución (obviamente sin necesidad de efectuar la traza de las iteraciones).
- (c) Suponiendo que el coste de aplicación de cada regla fuera 1, ¿se llegaría a una solución óptima (de coste mínimo)?
- (d) Considerar ahora un sistema de razonamiento dirigido por objetivos en el que la resolución de conflictos se lleve a cabo de acuerdo a los siguientes criterios:
  - i. El primer objetivo en secuencia
  - ii. La regla más simple
  - iii. La última regla en secuencia

Efectuar las 3 primeras iteraciones del proceso detallando en cada iteración explícitamente el conjunto de conflicto, el objetivo a resolver, la regla escogida y el resultado de su aplicación.

- (e) ¿Se ha llegado a la solución? En caso contrario, justificar si continuando el procedimiento se llegaría a una solución óptima. ¿Qué conclusiones se pueden extraer de las dos estrategias presentadas? ¿Qué estrategia se podría utilizar para garantizar que la solución obtenida fuera óptima?

9. Dadas las siguientes bases de hechos y de conocimientos aritméticos:

Base de hechos	Base de reglas
H1. num(13)	R1. $\text{num}(x) \wedge \text{num}(y) \wedge \text{divexacta}(x,y) \rightarrow \text{múltiplo}(x,y)$
H2. num(4)	R2. $\text{num}(x) \wedge \text{divexacta}(x,2) \rightarrow \text{par}(x)$
H3. num(2)	R3. $\text{num}(x) \wedge \text{no divexacta}(x,2) \rightarrow \text{impar}(x)$
H4. divexacta(4,2)	R4. $\text{par}(x) \wedge (x > 2) \rightarrow \text{par}(x-2)$
H5. divexacta(2,2)	R5. $\text{par}(x) \rightarrow \text{par}(x+2)$
	R6. $\text{num}(x) \wedge \text{múltiplo}(x,2) \rightarrow \text{par}(x)$
	R7. $\text{par}(x) \wedge \text{impar}(y) \rightarrow \text{impar}(x+y)$

Divexacta(x,y) expresa que x dividido por y da resto cero.

Múltiplo(x,y) expresa que x es múltiplo de y.

- (a) Aplica razonamiento hacia delante y realiza las cinco primeras iteraciones. En cada iteración indica el conjunto conflicto, la selección realizada y el resultado de la aplicación de la regla. Estrategia de resolución de conflictos: 1) Regla con más condiciones, 2) Instanciación más reciente, 3) Instanciación con hecho más antiguo que satisfaga la primera condición.
- (b) Considera el objetivo impar(17) y comprueba si es deducible aplicando razonamiento hacia atrás. Expresa claramente, ya sea mediante árbol o por pasos, para cada iteración el conjunto conflicto, la selección realizada y el resultado. Aplica una estrategia secuencial para la selección de subobjetivos y como estrategia de resolución de conflictos aplica: 1º) selección secuencial de reglas, 2º) preferencia a hechos más recientes.

10. Dadas las siguientes bases de hechos y de conocimientos relativos a personas y concesiones de créditos bancarios:

Base de Hechos			
H1.	cliente(P)	H8.	salario(P,100.000)
H2.	cliente(Q)	H9.	posee(R,C1)
H3.	cliente(R)	H10.	valor(C1,600.000)
H4.	pariente(Q,R)	H11.	trabajador(S)
H5.	rico(Q)	H12.	salario(S,200.000)
H6.	trabajador(P)	H13.	posee(S,C2)
H7.	especial(R)	H14.	valor(C2,1.000.000)

Base de reglas	
R1.	$\text{cliente}(x) \wedge \text{rico}(x) \rightarrow \text{crédito}(x,y)$
R2.	$\text{cliente}(x) \wedge \text{pariente}(x,y) \wedge \text{rico}(y) \rightarrow \text{rico}(x)$
R3.	$\text{pariente}(x,y) \rightarrow \text{pariente}(y,x)$
R4.	$\text{cliente}(x) \wedge \text{posee}(x,y) \wedge \text{valor}(y,z) \wedge (z > t) \rightarrow \text{crédito}(x,t)$
R5.	$\text{cliente}(x) \wedge \text{trabajador}(x) \wedge \text{salario}(x,y) \wedge (y \cdot 3 > z) \rightarrow \text{crédito}(x,z)$
R6.	$\text{cliente}(x) \wedge \text{especial}(x) \wedge \text{crédito}(x,y) \rightarrow \text{crédito}(x, 2 \cdot y)$

- (a) Considera el objetivo crédito (R,2.000.000) y comprueba si es deducible aplicando razonamiento hacia atrás. Expresa claramente, ya sea mediante árbol o por pasos, para cada iteración el conjunto conflicto, la selección realizada y el resultado. Para la selección de subobjetivos da preferencia al subobjetivo más simple (menos argumentos), y como estrategia de resolución de conflictos aplica: 1º) preferencia a la regla con más condiciones, 2º) primera regla en secuencia.
- (b) La regla 6 desea expresar que a los clientes especiales se les puede conceder el doble de crédito que a un cliente no especial que tenga las mismas características. Crees que está bien diseñada y su funcionamiento se ajustará a esta definición.
- (c) Considera ahora la siguiente base de reglas:

Base de reglas	
R1.	$\text{cliente}(x) \wedge \text{rico}(x) \rightarrow \text{crédito}(x,y)$
R2.	$\text{cliente}(x) \wedge \text{pariente}(x,y) \wedge \text{rico}(y) \rightarrow \text{rico}(x)$
R3.	$\text{pariente}(x,y) \rightarrow \text{pariente}(y,x)$
R4.	$\text{cliente}(x) \wedge \text{posee}(x,y) \wedge \text{valor}(y,z) \rightarrow \text{crédito}(x,z)$
R5.	$\text{cliente}(x) \wedge \text{trabajador}(x) \wedge \text{salario}(x,y) \rightarrow \text{crédito}(x,3 \cdot y)$

Aplica razonamiento hacia delante para saber el crédito máximo para las cuatro personas (P,Q,R,S). En cada iteración indica el conjunto conflicto, la selección realizada y el resultado de la aplicación de la regla. Estrategia de resolución de conflictos: 1) Regla con más condiciones, 2) Instanciación más antigua.

11. Considera la siguiente base de hechos (donde cada hecho viene acompañado por el momento de su creación) y la siguiente base de reglas:

Base de hechos	Base de reglas
H1. $P(a,b)$	R1. $P(x,y) \rightarrow P(y,x)$
H2. $E(a,b)$	R2. $P(x,y) \rightarrow H(x,y)$
H3. $E(b,c)$	R3. $C(x) \rightarrow V(x)$
H4. $C(a)$	R4. $H(x,y) \rightarrow S(x,y)$
H5. $C(b)$	R5. $V(x) \wedge E(x,y) \rightarrow S(x,y)$
H6. $C(c)$	R6. $M(x,y) \rightarrow S(x,y)$

- (a) Resuelve el objetivo  $S(b,a)$  suponiendo el mecanismo de inferencia dirigido por hechos y una estrategia de resolución de conflictos con los criterios: 1º) regla más específica, 2º) primera regla en secuencia, 3º) instanciación con hechos más antiguos. Indica en cada iteración el conjunto de conflicto, la selección realizada y el resultado de la aplicación.
- (b) Considera ahora el objetivo  $S(a,c)$ . Suponiendo el mecanismo de inferencia dirigido por objetivos y una estrategia de resolución de conflictos con los criterios: 1º) subobjetivo más simple (menos parámetros), 2º) regla más específica, 3º) primera regla en secuencia, realiza las iteraciones necesarias. Indica en cada iteración el conjunto de conflicto, la selección realizada y el resultado de la aplicación

12. Dadas las siguientes bases de hechos y conocimiento donde los hechos están ordenados por antigüedad,  $x$ ,  $y$  son variables y  $a$ ,  $b$ ,  $c$  son constantes:

Base de hechos	Base de reglas
H1. $T(b,c)$	R1. $T(c,x) \wedge Q(x,c) \wedge U(a) \rightarrow R(c,x)$
H2. $P(a)$	R2. $T(b,x) \wedge S(a) \rightarrow P(x)$
H3. $Q(b,c)$	R3. $P(x) \wedge U(y) \rightarrow R(x,y)$
H4. $T(c,b)$	R4. $S(x) \rightarrow R(x,b)$
H5. $P(c)$	R5. $P(x) \wedge Q(a,x) \rightarrow S(x)$
H6. $Q(a,a)$	R6. $S(x) \wedge P(x) \rightarrow U(x)$
H7. $Q(a,b)$	R7. $R(x,c) \wedge U(y) \rightarrow Q(x,y)$

- (a) Considera el objetivo  $R(c,b)$  y comprueba si es deducible aplicando razonamiento hacia atrás. Utiliza las estrategias de resolución de conflictos: 1º) subobjetivo con menos parámetros, 2º) el subobjetivo más reciente, 3º) la regla más general y 4º) la primera regla en orden secuencial. Especifica claramente para cada iteración el conjunto conflicto, la selección realizada y el resultado de la aplicación de la regla.
- (b) Considera ahora el objetivo  $R(a,b)$  y aplica razonamiento hacia delante para intentar demostrarlo. Utiliza las estrategias de resolución de conflictos: 1º) la regla más específica, 2º) la primera regla en orden secuencial y 3º) la instanciación con hechos más recientes. Especifica claramente para cada iteración el conjunto conflicto, la selección realizada y el resultado de la aplicación de la regla. ¿Es óptima la solución? En caso de que no, sugiere una estrategia/s de resolución de conflictos alternativa para que lo sea.

13. Dados los siguientes conjuntos de reglas de producción y hechos:

Base de hechos	Base de reglas
H1. $U(b,c)$	R1. $U(x,c) \rightarrow S(x,e)$
H2. $P(a,f)$	R2. $P(x,f) \rightarrow S(x,e)$
H3. $P(f,f)$	R3. $S(x,e) \wedge U(x,h) \rightarrow Q(x,e)$
H4. $P(b,a)$	R4. $S(x,e) \wedge U(x,c) \wedge Q(x,b) \rightarrow Q(x,e)$
H5. $U(f,c)$	R5. $U(x,c) \rightarrow R(x,d)$
H6. $R(a,h)$	R6. $P(b,x) \wedge R(x,h) \rightarrow U(x,h)$
	R7. $R(x,d) \rightarrow U(x,c)$



- (a) Considera el objetivo  $Q(a,e)$  y comprueba si es demostrable aplicando razonamiento hacia atrás. Usa las estrategias de resolución de conflictos 1) el subobjetivo más reciente, 2) la regla más específica y 3) la primera regla en orden secuencial.
- (b) Muestra las cinco primeras iteraciones resultantes de aplicar razonamiento hacia delante. Utiliza las estrategias de resolución de conflictos 1) la regla más simple y 2) la instanciación con hechos más recientes, 3) la primera regla en orden secuencial.

14. Donats els següents conjunts de regles de producció i fets:

Base de hechos	Base de reglas
F1. $Q(d,b)$	R1. $Q(x,y) \rightarrow S(x,y)$
F2. $Q(a,a)$	R2. $P(x,y,z) \wedge Q(z,a) \rightarrow S(z,y)$
F3. $P(a,b,d)$	R3. $R(x,y) \wedge Q(y,x) \wedge R(y,z) \rightarrow S(x,z)$
F4. $R(a,d)$	R4. $R(x,y) \wedge S(x,y) \rightarrow P(x,d,y)$
F5. $P(a,c,d)$	R5. $R(x,y) \wedge S(x,t) \wedge Q(y,b) \rightarrow P(x,t,z)$
F6. $R(a,c)$	R6. $R(x,y) \wedge P(x,z,y) \rightarrow Q(x,z)$
F7. $S(a,b)$	

- (a) Veure si el fet  $s(a,c)$  es deduïble per raonament cap enrera aplicant la seqüent estratègia de selecció de regles, fets i objectius: 1) objectius en ordre seqüencial dels més recentment creats, 2) regla més específica, 3) regla en ordre seqüencial, 4) fets en ordre seqüencial.
- (b) Realitzar les 5 primeres iteracions de raonament cap endavant aplicant la seqüent estratègia de selecció: 1) regla més específica, 2) instanciació més recent (comparació feta per ordre de predicats).

15. Considera la siguiente base de hechos (donde cada hecho viene acompañado por el momento de su creación) y la siguiente base de reglas:

Base de hechos	Base de reglas
H1. $P(a,b)$	R1. $T(x) \wedge P(x,y) \rightarrow S(x)$
H2. $P(b,c)$	R2. $R(x,y) \rightarrow P(y,x)$
H3. $Q(b)$	R3. $P(x,y) \wedge S(y) \rightarrow R(y,x)$
H4. $T(c)$	R4. $P(x,y) \wedge Q(x) \rightarrow R(x,y)$
H5. $S(b)$	

- (a) Comprueba si el objetivo  $R(a,c)$  se puede obtener suponiendo un mecanismo de inferencia dirigido por los hechos y una estrategia de resolución de conflictos con los criterios: 1) regla mas general, 2) instanciación con hechos globalmente mas recientes. Indica en cada iteración el conjunto de conflicto, la selección realizada y el resultado de la aplicación.
- (b) Considera ahora el objetivo  $R(c,a) \vee S(c)$ . Suponiendo un mecanismo de inferencia dirigido por los objetivos y una estrategia de resolución de conflicto con los criterios: 1) Subobjetivo mas complejo (mas parámetros), 2) Primera regla en secuencia. Realiza las iteraciones necesarias. Indica en cada iteración el conjunto de conflicto, la selección realizada y el resultado de la aplicación.

16. Tenemos la siguiente base de reglas y la siguiente base de hechos:

Base de hechos	Base de reglas
H1. $R(a)$	R1. $R(x) \wedge R(y) \rightarrow P(x,y)$
H2. $R(b)$	R2. $R(x) \rightarrow P(x,x)$
H3. $Q(b)$	R3. $P(x,y) \wedge Q(x) \rightarrow R(x)$
	R4. $P(x,y) \wedge R(y) \rightarrow Q(y)$

- (a) Realiza los 3 primeros pasos del razonamiento hacia adelante con la estrategia de resolución de conflictos 1) primera regla en secuencia, 2) instanciación con hechos mas recientes. Especifica claramente para cada iteración el conjunto de conflicto, la selección realizada y el resultado de la aplicación de la regla.
- (b) Realiza los 3 primeros pasos del razonamiento hacia adelante con la estrategia de resolución de conflictos 1) regla mas general, 2) instanciación con hechos mas recientes. Especifica claramente para cada iteración el conjunto de conflicto, la selección realizada y el resultado de la aplicación de la regla.
- (c) Realiza los 3 primeros pasos del razonamiento hacia adelante con la estrategia de resolución de conflictos 1) regla mas específica, 2) instanciación con hechos mas antiguos. Especifica claramente para cada iteración el conjunto de conflicto, la selección realizada y el resultado de la aplicación de la regla.
- (d) Utiliza el razonamiento hacia atrás para resolver el objetivo Q(c) utilizando como estrategia de resolución de conflictos 1) objetivos en secuencia 2) reglas en secuencia 3) hechos en secuencia. Especifica claramente para cada iteración el conjunto de conflicto, la selección realizada y el resultado de la aplicación de la regla.

17. Considera la siguiente base de hechos (donde cada hecho viene acompañado de su momento de creación) y de reglas:

Base de hechos	Base de reglas	
H1. R(a)	R1. $P(x,y) \wedge Q(x) \rightarrow P(x,x)$	R3. $P(x,y) \rightarrow Q(y)$
H2. Q(b)	R2. $Q(x) \wedge Q(y) \rightarrow P(x,y)$	R4. $R(x) \wedge Q(y) \rightarrow Q(x)$

- (a) Realiza los 4 primeros pasos del razonamiento hacia adelante con la estrategia de resolución de conflictos 1) primera regla en secuencia 2) instanciación con hechos mas recientes 3) condición de mas a la izquierda con hecho mas reciente. Especifica claramente para cada iteración el conjunto de conflicto, la selección realizada y el resultado de la aplicación de la regla.
- (b) Realiza los 4 primeros pasos del razonamiento hacia adelante con la estrategia de resolución de conflictos 1) regla mas específica 2) hechos mas antiguos 3) condición de mas a la izquierda con hecho mas antiguo. Especifica claramente para cada iteración el conjunto de conflicto, la selección realizada y el resultado de la aplicación de la regla.
- (c) Utiliza el razonamiento hacia atrás para resolver el objetivo P(b,a) utilizando como estrategia de resolución de conflictos 1) objetivos en secuencia 2) reglas en secuencia 3) hechos en secuencia. Especifica claramente para cada iteración el conjunto de conflicto, la selección realizada y el resultado de la aplicación de la regla.
- (d) Utiliza el razonamiento hacia atrás para resolver el objetivo Q(c) utilizando como estrategia de resolución de conflictos 1) objetivos en secuencia 2) reglas en secuencia 3) hechos en secuencia. Especifica claramente para cada iteración el conjunto de conflicto, la selección realizada y el resultado de la aplicación de la regla.

18. Considera la siguiente base de hechos, en la que cada hecho viene acompañado de su momento de creación, y la de reglas:

Base de hechos	Base de reglas	
H1. Q(b)	R1. $Q(x) \wedge Q(y) \rightarrow P(x,y)$	R3. $Q(y) \wedge P(x,y) \rightarrow Q(x)$
H2. R(a)	R2. $Q(x) \rightarrow P(x,x)$	R4. $R(x) \rightarrow Q(x)$

- (a) Aplica 5 iteraciones de razonamiento hacia adelante con la estrategia de resolución de conflictos 1. primera regla en secuencia, 2. instanciación con hechos más antiguos y 3. condición de más a la izquierda con hecho más antiguo.

- 
- (b) Realiza 5 iteraciones de razonamiento hacia delante con la estrategia de resolución de conflictos 1. regla más general, 2. instanciación con hechos más recientes y 3. condición de más a la izquierda con hecho más reciente.
  - (c) Aplica razonamiento hacia atrás para resolver el objetivo  $P(b,a)$  y usa como estrategia de resolución de conflictos 1. objetivos en secuencia, 2. reglas en secuencia, 3. hechos en secuencia.

En cada uno de los apartados anteriores especifica, para cada iteración, el conjunto conflicto, la selección realizada y por qué, y el resultado de la aplicación de la regla seleccionada.



## 8. Enginyeria del Coneixement

1. Para los siguientes problemas indica si son problemas de síntesis o de análisis justificando tu respuesta:

- Determinar la mejor ubicación para un hospital dentro de una ciudad dadas un conjunto de posibles ubicaciones y sus características.
- Determinar la mejor forma de asignar los horarios del personal de ambulancias de una red de hospitales para que se puedan atender un mínimo de urgencias optimizando los desplazamientos de las ambulancias.
- Determinar para una lista de operaciones quirúrgicas a realizar y una lista de médicos y personal de quirófano, en qué orden se han de realizar las operaciones y qué personal ha de intervenir, dadas las restricciones de horarios de quirófanos y la urgencia de las intervenciones.

2. El departamento de planificación urbana de la ciudad de Urbópolis ha decidido utilizar un SBC para la planificación de los diferentes elementos de la ciudad.

Para este departamento la ciudad está compuesta por unidades urbanas de tres tipos distintos. El primer tipo son las unidades comerciales que pueden corresponder a comercios mayoristas o comercios minoristas. El segundo tipo son las unidades de servicio al ciudadano tales como colegios públicos, bibliotecas, zonas verdes, polideportivos, hospitales, centros de asistencia primaria, plazas públicas, farmacias y oficinas de servicios públicos. Finalmente, el tercer tipo son las unidades residenciales que pueden corresponder a viviendas unifamiliares, edificios de pisos y manzanas de viviendas.

Las unidades urbanas se pueden ubicar en espacios. Hay espacios de dos tipos. El primero son los solares que pueden clasificarse en bloques (capacidad para un solo edificio), manzanas (capacidad para varios edificios) y grandes áreas (capacidad para grandes instalaciones). El segundo tipo son los locales.

El departamento de planificación tiene la información de todos los espacios que existen en la ciudad. También dispone de la información de qué espacios están vacantes en la actualidad y qué unidades urbanas se han asignado a los espacios ya ocupados. A cada espacio solo se le puede asociar una unidad urbana.

Las unidades urbanas se encuentran dentro de barrios, que es la división más fina en la que el ayuntamiento agrupa las zonas de la ciudad. El barrio tiene información sobre su población, incluyendo su distribución por edades (niños, adultos, ancianos). Los barrios a su vez están agrupados en distritos.

Otro elemento de la ciudad son las vías. Estas se pueden clasificar en grandes vías, avenidas, vías fluidas y vías comunes. Todo espacio tiene una serie de vías con las que está relacionado (aquellas

que lo bordean y/o cruzan). Cada vía tiene información de la cantidad de coches por minuto que la recorren. Estas vías estarán relacionadas también con barrios y distritos.

- (a) Propón una representación estructurada del dominio descrito e identifica los atributos más relevantes.
- (b) El primer objetivo del departamento de planificación es hacer una clasificación de los espacios vacantes de manera que se pueda tener un catálogo que indique qué tipos de usos son adecuados para cada una de ellos. Suponemos que a cada espacio solo se le puede asociar una unidad urbana. Por el conocimiento de los ingenieros urbanísticos, se sabe que las unidades comerciales solo pueden ser ubicadas en locales. Lo mismo sucede con las oficinas de servicios públicos y farmacias que solo pueden ubicarse en locales. Las bibliotecas se pueden colocar en locales si la población del barrio no es muy grande o ya existen otras bibliotecas en el barrio. En caso contrario se ubican en bloques.

Respecto a las unidades residenciales, las viviendas unifamiliares y los edificios de pisos solo se pueden ubicar en bloques y las manzanas de viviendas solo se pueden ubicar en manzanas.

Para el resto unidades de servicio al ciudadano, los colegios, centros de asistencia primaria y plazas solo pueden ubicarse en manzanas y los polideportivos y parques solo pueden hacerlo en grandes áreas.

Para decidir el uso de un espacio se tienen en cuenta además otras características de la zona donde está ubicado ese espacio. Por ejemplo, se tiene en cuenta la densidad de población del barrio (*alta*, *media*, *baja*) globalmente o por tipos de población. También se estima la densidad del tráfico del barrio (*alta*, *media*, *baja*) respecto al número de vías que pasan por él y su densidad.

También se tienen en cuenta las diferentes proximidades (*al lado*, *cerca*, *media*, *lejos*) de las unidades urbanas ya ubicadas en los espacios, tanto a nivel de barrio como a nivel de distrito. Finalmente también se tienen en cuenta el tipo de vías con las que está relacionado el espacio y la densidad de su tráfico.

Por ejemplo, los colegios se deben ubicar en zonas de *alta* densidad de población infantil, donde haya al menos *cerca* bibliotecas y zonas verdes como máximo a *media* distancia del colegio. La densidad del tráfico de las vías *cercanas* ha de ser baja. Todo colegio debe estar a distancia *media* o *lejos* de cualquier otro colegio.

Los centros de asistencia primaria deben ubicarse preferiblemente en zonas de *alta* densidad de población infantil y/o anciana, *cerca* de alguna farmacia, con una densidad de tráfico alrededor *media*, *cerca* de alguna vía fluida y a una distancia *lejos* de otros centros de asistencia primaria.

El problema que se plantea es un problema de análisis y se puede resolver mediante clasificación heurística. Determina cómo se ubicarían los pasos de resolución de este problema en cada una de las fases de esta metodología. Da diferentes ejemplos de reglas para cada una de las fases que muestren como se llegaría a asociar un colegio a un espacio.

- (c) Una alternativa para representar el conocimiento de los ingenieros y las normas urbanísticas que permite obtener una propuesta de usos de los espacios vacantes son las redes bayesianas. Como ya se ha mencionado en el apartado anterior, en una manzana podría por ejemplo ubicarse una manzana de viviendas, un colegio, una plaza o un centro de asistencia primaria. Adicionalmente las evidencias recolectadas pueden hacer que alguno de los usos tenga más fuerza que otros. La red bayesiana permitiría representar el proceso de asignación de unidades y sus pesos.  
Comenta cómo modelarías el problema con una red bayesiana (qué escogerías como nodos, qué valores tendrían y qué dependencias habría entre los nodos) y da un ejemplo simple de red bayesiana para ilustrarlo.
- (d) Determinar los posibles usos de cada uno de los espacios solo es el primer paso del proceso que sigue el departamento de planificación. El siguiente paso es encontrar la mejor forma de ocuparlos. En el caso particular de las unidades de interés ciudadano, se ha de tener en cuenta las unidades urbanas de ese tipo ya existentes, las diferentes restricciones de ubicación que impone el propio departamento y los recursos de que se disponen.

La ciudad determina que debe haber al menos un centro de atención primaria por barrio, pero no más de diez por distrito. Los centros de atención primaria han de estar lo más *lejos* posible entre sí dentro del barrio. Debe haber al menos dos farmacias *cercanas* a cada centro de atención primaria, pero las farmacias del barrio deben estar a una distancia *media* entre ellas. Debe haber al menos un hospital por distrito y deben estar a distancia *media* de los centros de atención primaria.

Debe haber un colegio en un barrio por cada 300 niños, pero éste debe estar a una distancia *media* de otros colegios del barrio y debe estar *cerca* de al menos dos bibliotecas y al menos tres plazas en el barrio. Debe haber a distancia *media* del colegio al menos una zona verde y un polideportivo en el barrio.

El barrio debe tener al menos una zona verde y un polideportivo pero las áreas verdes y polideportivos de cada barrio deben estar a distancia *lejana* entre sí. Cada barrio debe tener al menos 5 plazas que deben estar a distancia *media* entre sí pero no más de 30 por distrito. No debe haber grandes vías *cerca* de colegios, zonas verdes o polideportivos.

Las oficinas de atención ciudadana deben estar *cerca* entre sí en el barrio y no debe haber más de 5 por barrio. Debe haber como mínimo una biblioteca por barrio y debe estar *cerca* de alguna oficina de atención ciudadana.

Este problema se puede plantear como un problema de satisfacción de restricciones. Explica cómo se puede resolver con este enfoque. Identifica las variables adecuadas, sus dominios, los diferentes tipos de restricciones entre variables que se necesitan y la estructura de la red de restricciones. Justifica tus respuestas.

3. El siguiente problema es un problema de análisis:

“Una labor de los equipos médicos de urgencias en situaciones de emergencia es determinar la prioridad con la que se ha de tratar a los pacientes. Este proceso es denominado *triage*. Los médicos están entrenados para analizar el estado del paciente y clasificarlo en tres grupos: Situación vital, grave pero no vital y heridas menores. Para realizar esta clasificación se utilizan diferentes criterios, pero podríamos reducirlos a tres: cardiovasculares, traumatismos y respiratorios. Cada uno de estos incluye toda una serie de síntomas, como por ejemplo, paro cardíaco, arritmias, tensión arterial, ... para los cardiovasculares, hemorragias, heridas abiertas, cortes, fracturas, ... para traumatismos, obstrucción de vías, perforación pulmonar, colapso pulmonar, ritmo respiratorio, ... para respiratorios”

Explica cómo lo resolverías usando clasificación heurística. Da algún ejemplo de regla para cada una de las fases de esta metodología.

4. Una importante marca de móviles y tablets desea ampliar los servicios que da a sus clientes creando su propia tienda de productos multimedia (a la que llamará simplemente *Media Store*). La parte innovadora es que quiere que la tienda genere recomendaciones personalizadas a los usuarios de sus móviles y tablets sobre que contenidos podría comprar a partir de las compras que ya ha realizado y la información que ha recolectado de los usuarios que han hecho compras similares a las suyas. Los productos que se venden en la *Media Store* son libros electrónicos, películas y música (canciones individuales). También se ofrecen packs de productos, que pueden ser todos del mismo tipo (por ejemplo, un album entero de un cantante o grupo musical es un pack con las canciones de ese album) o una mezcla de diferentes tipos (por ejemplo, un pack compuesto por una película, su banda sonora y el libro en el que se basa dicha película).

Se ha decidido utilizar un sistema basado en el conocimiento capaz de determinar el perfil al que pertenece un usuario usando las características de todas las compras realizadas de cada tipo de producto. De todas formas, para poder refinar más las recomendaciones guardamos también la cantidad de productos de cada tipo que ha comprado el usuario a lo largo del tiempo.

De los libros electrónicos se guarda información como el título, el autor, la fecha de publicación y el precio. Los libros están categorizados según su temática en libros científico-técnicos (matemáticas, física, biología, informática, ...), literatura (clasificada como clásica o contemporánea o bien como novela, teatro o poesía), juvenil, policíaca, cocina, etc.

De las películas se guarda información como el título, el director, la fecha de publicación, su duración, el número de premios recibidos y el precio. Estas están categorizadas según su género en acción, comedia,

drama, documental, ciencia-ficción, animación, cine clásico, etc. En el caso de que la película esté basada en un libro dentro del catálogo de la *Media Store* esto está también guardado.

De las canciones se guarda información como el título, intérprete, fecha de publicación, duración y precio. Estas están categorizadas por su género musical en jazz, pop-rock, country, techno, urbano, música étnica, música clásica, bandas sonoras, etc. En el caso de las bandas sonoras, se especifica a que película pertenecen del catálogo de la *Media Store*

Para simplificar las cosas se ha determinado un conjunto de características que permiten categorizar mejor a los usuarios de la *Media Store* y poder guiar el proceso de recomendación:

- el **tipo de usuario**, que puede ser *ocasional*, *un-poco-de-todo*, *melómano*, *cinéfilo* o *lector*. Los usuarios *ocasionales* son los que han tenido menos de diez compras en cada tipo de producto, y a estos solo se les puede recomendar los productos más vendidos del catálogo de la Media Store. Los usuarios *un-poco-de-todo* son los que no tienen unas preferencias definidas y compran productos de todas las categorías, a estos se les pueden recomendar solo los productos más vendidos del catálogo dada su indefinición. Los usuarios de tipo *cinéfilo* son aquellos usuarios que compran mayoritariamente cine, y se les recomendará según su perfil de cine (ver a continuación). Los usuarios de tipo *melómano* son aquellos usuarios que compran mayoritariamente canciones, y sus recomendaciones se construirán a partir de su perfil musical (ver a continuación). Finalmente los usuarios de tipo *lector* son aquellos usuarios que compran mayoritariamente libros electrónicos, y a los que se les recomendará según su perfil lector (ver a continuación).
- el **perfil de cine**, que puede ser *nulo* (no compra cine), *indefinido* (compra cine de cualquier género), o bien alguno de los géneros (*acción*, *comedia*, *drama*, *documental*, *ciencia-ficción*, *animación*, *cine clásico*, etc.) si sus compras son mayoritariamente de ese género. En el caso de perfil de cine *nulo* o *indeterminado* las recomendaciones se construirán según el tipo de usuario (que será *ocasional* o *un poco de todo*, ver arriba). En el caso de perfil de cine *acción* se les recomendarán las películas más vendidas del género acción. En el caso de perfil de cine *drama* se le recomendarán las películas más premiadas del género drama. Para el resto de valores del perfil de cine (*comedia*, *documental*, *ciencia-ficción*, *animación*, etc.) se le recomendarán las novedades de cine del catálogo relacionadas con ese género.
- el **perfil musical**, que puede ser *nulo* (no compra canciones), *indefinido* (compra canciones de cualquier género), o bien alguno de los géneros (*jazz*, *pop-rock*, *country*, *techno*, *urbano*, *música étnica*, *música clásica*, *bandas sonoras*, etc.) si sus compras son mayoritariamente de ese género. En el caso de perfil musical *nulo* o *indeterminado* las recomendaciones se construirán según el tipo de usuario (que será *ocasional* o *un poco de todo*, ver arriba). En el caso de perfil de música *jazz* o *música clásica* se les recomendarán las canciones más vendidas entre los melómanos de esos géneros. Para el resto de valores del perfil musical (*pop-rock*, *country*, *techno*, *urbano*, etc.) se le recomendarán las novedades de canciones del catálogo relacionadas con ese género.
- el **perfil lector**, que puede ser *nulo* (no compra libros electrónicos), *indefinido* (compra libros electrónicos de cualquier temática), o bien alguno de los temas (*matemáticas*, *física*, *biología*, *informática*, *novela clásica*, *novela contemporánea*, *teatro clásico*, *teatro contemporáneo*, *poesía clásica*, *poesía contemporánea*, *juvenil*, *policíaca*, *cocina*, etc.) si sus compras son mayoritariamente de esa temática. En el caso de perfil lector *nulo* o *indeterminado* las recomendaciones se construirán según el tipo de usuario (que será *ocasional* o *un poco de todo*, ver arriba). En el caso de perfil lector *novela clásica* o *teatro clásico* o *poesía clásica* se les recomendarán los libros más vendidos entre los lectores de esos géneros. Para el resto de valores del perfil lector se le recomendarán las novedades de libros del catálogo relacionadas a esa temática.
- el **gasto mensual** en productos de la *Media Store*, que puede ser *bajo* (por debajo de 20 euros/mes en el último año), *normal* (entre 21 euros/mes y 100 euros/mes en el último año) o bien *elevado* (por encima de 100 euros/mes en el último año). Para los usuarios con gasto mensual *bajo* se construirán las recomendaciones a partir de las ofertas del mes. Para los usuarios con gasto mensual *normal* se construirán las recomendaciones a partir de las características listadas arriba. Para los usuarios con gasto mensual *elevado* se añadirán a las recomendaciones packs de productos relacionados con su perfil.



Evidentemente un usuario no puede ser clasificado siempre en una única categoría, dadas sus últimas compras puede que cumpla características de varios tipos de compradores, por lo que la recomendación final debería ser una combinación de las recomendaciones adecuadas (los elementos subrayados arriba) para cada una de las categorías de comprador a las que pertenezca.

El objetivo es construir un sistema capaz de generar una recomendación de productos del catálogo de la *Media Store* adecuados a cada usuario del dispositivo (móvil o tablet).

- (a) Diseña la ontología del dominio descrito, incluyendo todos los conceptos que aparecen en la descripción e identificando los atributos más relevantes. Lista que conceptos forman parte de los datos de entrada del problema y que conceptos forman parte de la solución. (Nota: tened en cuenta que la ontología puede necesitar modificaciones para adaptarla al apartado siguiente).
- (b) El problema descrito es un problema de análisis. Explica cómo lo resolverías usando clasificación heurística, usando los conceptos de la ontología desarrollada en el apartado anterior. Da al menos 4 ejemplos de reglas para cada una de las fases de esta metodología.
- (c) Las características que se usan durante el proceso de recomendación de productos de la *Media Store* no son independientes entre sí. El tipo de usuario suele influir en el gasto mensual (el usuario ocasional suele tener un gasto bajo, y el usuario un-poco-de-todo suele tener un gasto normal). El perfil musical, el perfil lector y el perfil de cine suele influir en el tipo de usuario (usuarios con perfiles nulos suelen ser de tipo ocasional, mientras que los de perfiles indefinidos suelen ser de tipo un-poco-de-todo; usuarios con uno de los perfiles mucho más definidos que el resto suelen ser melómanos, cinéfilos o lectores, según corresponda). Finalmente el perfil lector, si tiene una temática muy definida, suele influenciar el género del perfil de cine (por ejemplo, los lectores de libros juveniles suelen preferir cine de animación, y los lectores de novelas contemporáneas suelen tener una predilección por películas de genero drama). Define el problema de asociación heurística como una red bayesiana expresando en ella al menos las relaciones indicadas en el enunciado, de forma que todas las características abstractas del problema que hayas definido en el apartado anterior tengan algun tipo de influencia en la solución. Separa bien en el diagrama qué variables describen características de problema y cuáles describen soluciones. Lista de forma clara los diferentes valores que puede tomar cada variable. Da un ejemplo de tabla de probabilidad de algún nodo, inventándote las probabilidades, pero expresando como influyen los valores de los nodos padre en las probabilidades de los valores de los nodos hijo.

5. La compañía de mudanzas *Smart Move* quiere desarrollar una herramienta web capaz de asesorar a sus clientes respecto a cual es la forma más adecuada de trasladar sus pertenencias de su lugar actual al destino de la mudanza. Esta compañía es capaz de realizar diferentes tipos de mudanzas desde diferentes tipos de ubicaciones, ya sean viviendas (tanto pisos como casas unifamiliares), u oficinas (tanto un planta de oficinas como un edificio completo de oficinas).

Una ubicación se compone de habitáculos, que se pueden clasificar, si son de viviendas, en dormitorio, salón, cocina y baño, o si son de oficinas en oficina, sala de reuniones, habitación de material y habitación multiusos. Respecto a las ubicaciones origen y destino de la mudanza, aparte de la dirección completa, es importante saber si tiene ascensor (incluidas sus dimensiones interiores) o solo hay un acceso de escaleras (de las que queremos saber el ancho y alto del lugar más estrecho). También son importantes los accesos al lugar, por ejemplo saber si hay una area de estacionamiento dentro del edificio, o si podemos aparcar cerca al estar junto a una calle con poco trafico, o si no se puede cargar delante del edificio sin cortar el tráfico.

Respecto a las cosas a trasladar, se puede estimar el volumen (en litros) a partir del número de habitáculos que tiene el lugar y las características y número de los objetos que contienen (muebles desmontables, muebles no desmontables, objetos de pequeño tamaño, objetos voluminosos, objetos frágiles). Los muebles no desmontables y los objetos voluminosos no se pueden empaquetar y se debe indicar su peso y su dimensión más larga (en cm). Los muebles desmontables y los objetos pequeños se pueden empaquetar y se conoce el volumen aproximado que ocupan empaquetados. Para los objetos frágiles se indica si se pueden empaquetar o no (indicando el volumen en el caso de que sean empaquetables) y si hace falta personal especializado para trasladarlos.

La empresa dispone de diferentes medios de mudanza (furgonetas, camiones grandes, contenedores de tren y grúas) y de cada uno sabemos el peso máximo de la carga (en kg). De las furgonetas, camiones grandes y trenes sabemos además el precio por kilómetro y el volumen de carga (en litros), mientras que de las grúas sabemos el precio de montaje de la grúa y el precio por hora. La empresa también dispone de personal para realizar la mudanza (empaquetadores, cargadores, desmontadores, personal especializado, conductor y operador de grúa), y de cada uno se tiene el NIF, el nombre y apellidos, un teléfono de contacto y el precio por hora. Una mudanza se compone de un conjunto de medios de mudanza y del personal necesario.

Para poder obtener la recomendación de la mudanza el usuario se ha de registrar en la web aportando su NIF, nombre y apellidos, teléfono de contacto, email y escogiendo un nombre de usuario (por razones de seguridad el password no lo guardamos en la ontología). A la hora de solicitar una mudanza el usuario introduce los datos de la ubicación de origen y la de destino (y, apartir de ellos, se calcula la distancia al lugar de la mudanza), cual es el tiempo máximo de carga y descarga de los objetos a trasladar (en horas) y el precio máximo que quiere pagar por la mudanza.

A partir de esta información y de las características de los objetos que hay que trasladar, el tipo de ubicación y las características de los accesos se quiere obtener una solución que indique si la mudanza debe hacerse mediante camiones grandes y/o furgonetas y cuántas hacen falta de cada tipo, si se ha de utilizar también el tren, si se ha de montar alguna grúa, el tipo de personal que es necesario y el número aproximado de personas de cada tipo. También dará una estimación del coste total de la mudanza.

El experto en logística de la empresa nos ha contado que ellos usan una serie de características durante el proceso de diseño de una mudanza. Hay características que son sencillas de obtener, como por ejemplo:

- el **tipo de mudanza**: *local* (menos de 10 Km), *regional* (menos de 100 Km) o *larga distancia*;
- el **presupuesto**: *bajo* (menos de 1000 euros), *medio* (menos de 3000 euros) o *alto*;
- la **duración de la mudanza**: *corta* (menos de 5 horas), *media* (menos de 10 horas), *larga*.

Otras requieren cierto razonamiento, como

- la **complejidad de la mudanza**: *sencilla* en la que la mayor parte de los objetos se pueden empaquetar, no hay que desmontar muchos muebles y no hay objetos voluminosos y frágiles, *normal* en la que hay que hacer bastante desmontaje, hay algunos objetos voluminosos pero no se necesitan grúas y hay algunos objetos frágiles y *difícil* en la que se necesitan grúas y hay bastantes objetos frágiles;
- el **volumen de la mudanza**: *pequeño* si el número de habitáculos es inferior a 5 y no hay objetos voluminosos o muebles no desmontables, *medio* si es una casa unifamiliar o una planta de oficinas y hay pocos objetos no desmontables, *grande* si es una planta de oficinas y hay bastantes objetos no desmontables, *extremo* si es un edificio de oficinas o hay una gran cantidad de objetos no desmontables;
- la **accesibilidad de la mudanza**: *accesible* si se pueden ubicar los medios de mudanza cerca de la ubicación y hay ascensor, *medianamente accesible* si hay ascensor, pero no caben todos los objetos voluminosos, *poco accesible* si no se pueden ubicar los medios de mudanza cerca de la ubicación y el ascensor no se puede utilizar para la mayor parte de los objetos voluminosos.

A partir de estas características deciden qué tipo de medios de mudanza necesitamos y los tipos de personal, por ejemplo para una mudanza de volumen grande harán falta camiones, si la mudanza es de larga distancia y de volumen grande hará falta usar el tren, si la ubicación es poco accesible es mejor usar furgonetas, ... Si la complejidad de la mudanza es difícil y es de volumen grande harán falta empaquetadores, cargadores y desmontadores, si el volumen es pequeño con cargadores podría haber suficiente, ...

El número específico de medios de mudanza y personal de cada tipo se puede obtener también razonando a partir de las características definidas, por ejemplo a mayor volumen más personal será necesario, a más objetos frágiles o voluminosos más cargadores y personal especializado hará falta, si la duración de la mudanza ha de ser corta se habrá de incrementar el personal

- (a) Diseña la ontología del dominio descrito, incluyendo todos los conceptos que aparecen en la descripción e identificando los atributos más relevantes. Lista que conceptos forman parte de los datos de entrada del problema y que conceptos forman parte de la solución. (Nota: tened en cuenta que la ontología puede necesitar modificaciones para adaptarla al apartado siguiente).
- (b) El problema descrito es un problema de análisis. Explica cómo lo resolverías usando clasificación heurística, usando los conceptos de la ontología desarrollada en el apartado anterior. Da al menos 3 ejemplos de reglas para cada una de las fases de esta metodología.
- (c) La parte de asociación heurística se podría resolver mediante el formalismo de redes bayesianas. Las características de una solución se podrían definir de manera que se obtuvieran una serie de valores a partir de los cuales se pudiera hacer mejor la especialización. Por ejemplo se podría definir la necesidad de camiones grandes o furgonetas en tres valores *ninguno o uno*, *dos o tres* y *más de tres*, y las necesidades de los diferentes tipos de personal (empaquetadores, cargadores, desmontadores, personal especializado...) en *ninguno*, *hasta 5 personas*, *hasta 10 personas* y *más de 10 personas*, y así con el resto de variables de la solución abstracta. Define el problema de asociación heurística como una red bayesiana expresando en ella al menos las relaciones indicadas en el enunciado, de forma que todas las características abstractas del problema que hayas definido en el apartado anterior tengan algún tipo de influencia en la solución. Separa bien en el diagrama que variables describen características de problema y cuales describen soluciones. Lista de forma clara los diferentes valores que puede tomar cada variable. Da un ejemplo de tabla de probabilidad de algún nodo, inventándote las probabilidades, pero expresando como influyen los valores de los nodos padre en las probabilidades de los valores de los nodos hijo.

6. Una compañía aérea desea organizar las asignaciones de puestos que han de desempeñar sus empleados de manera que se puedan cubrir todos los puestos que necesitan para atender los vuelos de la compañía que llegan y salen de un aeropuerto concreto.

El personal de que dispone la compañía se puede clasificar en personal de vuelo y personal de tierra. Para el personal de vuelo hay puestos de piloto y auxiliar de vuelo. Para el personal de tierra hay puestos en facturación de equipajes, venta de billetes, atención al cliente, embarque y reclamación de equipajes. Cualquier persona del personal de tierra puede ocupar cualquiera de los puestos posibles. Por razones evidentes no ocurre lo mismo con el personal de vuelo. Para el personal de tierra cada puesto se asigna a una persona durante una semana en cualquiera de los tres horarios de trabajo (mañana, tarde y noche). El personal de vuelo es móvil, ya que se desplaza en los vuelos y por lo tanto sólo puede ser asignado a un nuevo vuelo cuando ha regresado del último. Los puestos de tierra están asignados a las diferentes terminales del aeropuerto. En cada terminal hay facturación de equipajes, embarque y atención al cliente, pero solo hay venta de billetes y reclamación de equipajes en una terminal.

Para poder hacer la asignación de personal nos hará falta cierta información para cada persona como por ejemplo cual fue el último puesto que ocupó, cuándo lo hizo, en qué horario, etc.

La compañía ha de cubrir un número específico de puestos, tanto para vuelos como en tierra. Esta asignación deberá cumplir también ciertas restricciones, como por ejemplo que el personal de tierra no pase más de dos semanas en el mismo puesto, salvo si está en reclamación de equipajes, puesto en el que nadie puede estar más de una semana seguida en un mes. Tampoco puede pasar una persona más de un mes en el mismo horario. El personal de vuelo no puede realizar más de un vuelo internacional al mes y no debe ser asignado a más de 7 vuelos al mes.

El objetivo es construir un sistema capaz de obtener una asignación de personas a puestos en el aeropuerto durante un mes para cada uno de los horarios cumpliendo las restricciones comentadas.

- (a) Identifica qué conceptos, características, objetivos y soluciones forman el problema. Representa gráficamente los conceptos que has identificado mediante una red de frames y sus relaciones taxonómicas. Incluye en cada concepto al menos dos atributos que sean necesarios para su descripción. Describe completamente mediante el lenguaje de representación de frames vista en clase al menos tres relaciones entre los conceptos que has representado. Indica qué demonios podrían ser necesarios en la representación y qué comprobarían (no hace falta que los implementes).

- (b) El problema que se pretende resolver ¿es un problema de análisis o de síntesis? ¿Porqué?
- (c) ¿Si lo implementáramos mediante un sistema de producción que tipo de razonamiento te parecería el más adecuado para resolverlo? ¿Porqué?
- (d) ¿Como descompondrías la solución del problema? Identifica y especifica los subproblemas a resolver y como se encadenarían para construir la solución. Indica si los subproblemas se corresponden con alguna de las metodologías de resolución de problemas que conoces. Escribe para cada problema que identifiques algún ejemplo de las reglas de producción que harían falta para resolverlo.

7. Con el objetivo de aumentar la seguridad vial y controlar la contaminación ambiental, el Servei Català de Transit no solo quiere extender el sistema de velocidad variable a toda la red catalana de carreteras y a las vías principales de las ciudades, sino que además quiere implantar un sistema totalmente automatizado y mucho más inteligente que el que se ha desplegado hasta el momento, uno que sea capaz de analizar las condiciones de cada tramo de vía, la climatología y la contaminación atmosférica y decida la velocidad a la que han de circular los coches, velocidad que será mostrada en los paneles luminosos de esa vía. Se quiere aprovechar también la prueba piloto para probar el efecto de subir la velocidad máxima en autopista y autovía a 130 km/h (pero solo bajo buenas condiciones de la vía, del clima y de la contaminación).

Disponemos de datos de todos los tramos de vías que se quieren controlar. Se consideran dos tipos de vía: urbanas e interurbanas. Hay tres tipos de vía urbana a controlar: las internas de 4 o más carriles (máximo 80 km/h), las internas de 3 carriles (máximo 60 km/h) y las internas de 2 carriles (máximo 50 km/h). Hay tres tipos de vías interurbanas: las vías rápidas (autopistas y autovías, máximo 130 km/h), las carreteras de 3 o 4 carriles (2 carriles en un sentido, 1 o 2 carriles en el sentido contrario, máximo 100 km/h), y las carreteras de 2 carriles (1 por sentido, máximo 90 km/h). Las vías urbanas se componen de una serie de tramos de vía que se identifican por los números de la calle en los que empieza y acaba el tramo (por ejemplo, *Sepúlveda 1-49*), y que tienen asociado el número de carriles del tramo, la existencia de carril bus, la existencia de carril bici y la velocidad máxima en ese tramo. Las vías interurbanas se componen de tramos de vía que se identifican con el kilómetro inicial y final del tramo (por ejemplo *C-58 km.20-40*) y que tienen asociado el número de carriles del tramo y la velocidad máxima en ese tramo. Cada tramo de vía tiene asociados los tramos de vía contiguos con los que conecta.

De cada tramo de vía se dispone también de información sobre la temperatura y la climatología en el momento actual. Hay cinco tipos de condición meteorológica que el sistema monitoriza de forma especial por su impacto en el tránsito: viento (medido en km/h), lluvia (medido en litros por metro cuadrado y hora), nieve (medido en centímetros de espesor por hora), granizo (medido por dos parámetros, diámetro medio en cm del granizo, y gramos de granizo por metro cuadrado y hora) y niebla (medido por metros de visibilidad). En un tramo puede ocurrir más de una condición climática al mismo tiempo (puede llover y hacer viento con fuerza a la vez). Otra información asociada a cada tramo de vía es la calidad del aire, que se cuantifica con tres parámetros medidos en microgramos por metro cúbico ( $\mu\text{g}/\text{m}^3$ ): las concentraciones máximas de dióxido de nitrógeno ( $\text{NO}_2$ ) y ozono ( $\text{O}_3$ ) en la última hora, y la media diaria de las partículas en suspensión menores de 10 micrómetros ( $\text{PM}_{10}$ ).

Aprovechando el despliegue del sistema de control de la velocidad por tramos, se tiene información de cada uno de vehículos que circulan por un tramo de vía en un cierto instante. El sistema es capaz de leer la matrícula del vehículo y acceder a la información del mismo en la base de datos de Transit para obtener el tipo de vehículo (turismo, moto, autobús y transporte) y el nivel de emisiones de  $\text{CO}_2$  de su ficha de la ITV (medidas en gramos por kilómetro). Además las bases de datos de Transit también tienen, para cada tramo de vía, una lista de accidentes que han ocurrido en el pasado reciente, y para cada uno de ellos se guarda la fecha, los vehículos implicados, el número de heridos y el número de muertos.

Hemos querido saber cómo los operadores humanos del actual sistema de velocidad variable procesan toda la información disponible para decidir la velocidad de cada tramo de vía y nos han contado que tienen en cuenta los siguientes parámetros durante su proceso de toma de decisiones:

- La tipología de la vía, distinguiendo entre vías rápidas, carreteras y vías internas.

- La velocidad media del tramo en este momento, distinguiendo entre la velocidad máxima (los vehículos circulan a la velocidad máxima de la vía, todo va bien), moderada, lenta y congestión (tenemos un problema serio de tránsito)
- La velocidad media de los tramos contiguos (una sola media para todos, con los mismos 4 niveles: máxima, moderada, lenta y congestión). Si la velocidad en los tramos contiguos es mayor o menor que en el tramo actual se optará por recomendar una velocidad algo mayor o algo menor en el tramo actual para ajustarlo a sus contiguos.
- El tipo de vehículos transitando por el tramo, con tres posibles valores: mayoría ligeros (la mayoría de vehículos en el tramo son turismos y motos), mayoría pesados (una mayoría de autobuses y camiones de transporte), o mezcla de ligeros y pesados. Cuando se tiene una mayoría de vehículos pesados se tenderá a reducir algo la velocidad recomendada para aumentar la seguridad y para reducir también la posible polución que pueden causar.
- Nivel de polución del aire, que puede ser baja, media, alta o nivel de alerta. A mayor polución más se recomendará bajar la velocidad recomendada para intentar reducir la contaminación.
- Siniestralidad del tramo, basada en el historial de accidentes, y que se mide con cuatro valores: baja (menos de 2 accidentes por año, ningún muerto), media (menos de 5 accidentes por año, máximo un muerto al año), alta (menos de 10 accidentes al año, máximo 10 muertos) y punto negro (más de 10 accidentes al año o más de 10 muertos al año). A mayor siniestralidad se recomendará una reducción mayor de la velocidad de la vía.
- Agarre del asfalto, que indica la posibilidad de que los neumáticos de los vehículos pierdan agarre. Se describen cuatro niveles: óptimo (asfalto seco), reducido (asfalto está algo mojado por lluvia o nieve, precaución), peligroso (lluvias fuertes, nieve cuajando en el suelo) y nulo (lluvias torrenciales que provocan aquaplaning, placas de hielo o nieve). Cuanto peor sea el agarre del asfalto mayor será la reducción en la velocidad recomendada en ese tramo.
- Temperatura actual, que se describe con cuatro valores: bajo cero, frío, templado y caliente. En el caso de temperaturas bajo cero se recomienda siempre una reducción de velocidad, en el caso de frío solo se recomienda bajar velocidad si el asfalto tiene un agarre reducido.
- Visibilidad en el tramo, que puede ser óptima (aire limpio y despejado, día luminoso), normal (aire algo polucionado, nubes bajas o poca luminosidad) y pobre (polución muy alta, niebla, lluvias torrenciales o tormenta de nieve). A menor visibilidad mayor reducción en la velocidad recomendada.

El sistema ha de dar como resultado final (para cada tramo de vía) el valor de la velocidad recomendada que se proyectará en las pantallas instaladas en el tramo y que es un valor discreto entre los siguientes: 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130. El sistema calculará la velocidad de los tramos de uno en uno, de forma independiente.

- (a) Diseña la ontología del dominio descrito, incluyendo todos los conceptos que aparecen en la descripción e identificando los atributos más relevantes. Lista que conceptos forman parte de los datos de entrada del problema y que conceptos forman parte de la solución. (Nota: tened en cuenta que la ontología puede necesitar modificaciones para adaptarla al apartado siguiente).
- (b) El problema descrito es un problema de análisis. Explica cómo lo resolverías usando clasificación heurística, usando los conceptos de la ontología desarrollada en el apartado anterior. Da al menos 3 ejemplos de reglas para cada una de las fases de esta metodología.
- (c) Las características que se usan durante el proceso de toma de decisiones no son independientes entre sí. La velocidad media de los tramos contiguos influencia la velocidad media del tramo actual. El tipo de vehículos que transitan por el tramo (ligeros, pesados) influencia la polución del tramo, y ésta a su vez influencia la visibilidad. Y las temperaturas bajas suelen provocar menor agarre en el asfalto.

Define el problema de asociación heurística como una red bayesiana expresando en ella al menos las relaciones indicadas en el enunciado, de forma que todas las características abstractas del

problema que hayas definido en el apartado anterior tengan algun tipo de influencia en la solución. Separa bien en el diagrama que variables describen características de problema y cuales describen soluciones. Lista de forma clara los diferentes valores que puede tomar cada variable. Da un ejemplo de tabla de probabilidad de algún nodo, inventándote las probabilidades, pero expresando como influyen los valores de los nodos padre en las probabilidades de los valores de los nodos hijo.

8. El restaurant *Aprofita-ho* és un restaurant de prestigi internacional. S'ha guanyat el seu prestigi perquè elabora els seus plats tenint en compte les preferències del client i els ingredients dels que disposa. Així doncs, els menús resultants són al gust del client. el xef té en compte si el client prefereix la carn crua o cuïta, si li agrada el peix, si és vegetarià, si li agraden les verdures poc cuites, si li agraden les amanides, si prefereix les amanides a l'inici o al final, si li agrada la barreja entre dolç i salat, etc. També té en compte els ingredients dels que disposa: ous, pollastre, conill, enciam, endívies, col, patates, llobarro, rap, calamars, tomàquet, ceba, alls, etc. Amb tota aquesta informació, elabora els plats i confecciona el menú més adient per al client.

El xef pertany a la nova fornada de cuiners tecnòcrates, i està força convençut de que es pot informatitzar el procés d'elaboració dels plats i confecció del menú. Encara més, creu que podria ser útil un sistema expert, segons el que va aprendre en un curs de postgrau en Intel·ligència Artificial que va fer a la UPC. Responen a les següents qüestions:

- Identifica qué conceptes formen les dades d'entrada del problema. Representa gràficament aquests conceptes mitjançant una ontologia incloent els atributs mes rellevants i les relacions, tant les taxonòmiques como les no taxonòmiques que creguis que son necessaries.
  - Identifica els conceptes que formen la solució del problema. Representa gràficament aquests conceptes mitjançant una ontologia i les relacions, tant les taxonòmiques, como les no taxonòmiques, que creguis que son necessaries.
  - ¿Com descompondries el problema en subproblemes? Identifica i especifica els subproblemes a resoldre i com s'encadenarien per construir la solució.
  - Indica per cada subproblema que has identificat si es un problema d'anàlisis o de síntesi. Justifica-ho.
  - ¿Quin tipus de raonament et sembla mes adequat para cada subproblema? Justifica-ho.
9. Una empresa del sector de los videojuegos quiere ir mucho más allá en la introducción de la Inteligencia Artificial en sus productos. Para su próximo videojuego quieren que la IA del juego adapte la historia, el ritmo y el tipo de juego al tipo de jugador que lo usa, de forma que tenga más o menos acción, más o menos aventura, más o menos misterio, centrar la historia en el jugador individual o modificarla para que sea un juego de estrategia por equipos, todo ello dentro del mismo juego. Para hacer esta adaptación el sistema dispone del historial de jugador, tanto dentro del juego como en otros juegos de la misma compañía, extrayendo de ahí el perfil de jugador.

De cada juego del catálogo de esta empresa tiene asociado un título, un año de publicación, la edad recomendada (+0 años, +3 años, +5 años, +8 años, +12 años, +18 años), si es un juego on-line o no, si es un juego multi-jugador y la saga de juegos a la que pertenece (de la que solo se guarda el título y los juegos que la componen). La empresa solo ofrece en estos momentos ocho tipos de juego: arcade, juegos de estrategia en tiempo real (más conocidos por las siglas RTS en inglés), juegos de lucha, de deportes, aventuras gráficas, puzzles, simuladores de mundo y simuladores de vehículos, y de todos ellos se guarda la misma información excepto en el caso de los simuladores de vehículos, en los que se guarda además el conjunto de vehículos que los jugadores pueden pilotar. Cada juego esta compuesto por uno o más niveles, y de cada nivel se guarda su nombre, la dificultad (de 1 a 10), el conjunto de niveles desde los que se puede acceder a ese nivel, y el tiempo máximo para poder superar el nivel (en segundos, si vale 0 significa que no hay límite de tiempo). Cada nivel está compuesto por uno o más espacios, y de cada uno se guarda su nombre, si es un espacio interior o exterior, el conjunto de espacios desde los que se puede acceder a ese espacio y el conjunto de objetos que estan situados dentro de ese espacio. De cada objeto se guarda su nombre y su precio en créditos (en moneda ficticia,

el usuario puede gastar esos créditos en el mismo juego o en otros juegos de la misma compañía). Hay cinco tipos de objetos: las llaves, de las que se guarda además el espacio al que pueden dar acceso; las armas, de las que se guarda nivel de daño que pueden generar por ataque y la energía disponible (en %); los vehículos, que pueden ser aéreos, espaciales, marinos, anfibios o terrestres, y de los que se guarda el nombre, el nivel de energía y el número de plazas; las herramientas, de las que se guarda solo el nombre y el precio, y los contenedores, de los que se guarda el conjunto de objetos que tienen dentro.

La participación de los jugadores en los juegos se guarda por partidas, y de cada partida se guarda la fecha y hora de inicio, la duración (en minutos), los niveles del juego completados por el usuario, los personajes que han participado y los equipos que han participado. En un juego hay uno o varios personajes, que pueden ser personajes jugador (personajes que corresponden a usuarios reales) y personajes no jugador (más conocidos por el acrónimo NPC, personajes que corresponden a jugadores virtuales que crea el juego). De los personajes jugador se guarda el nombre del personaje (texto), el usuario real asociado a este personaje su rol (texto), su salud (en %), los puntos obtenidos (entero), los créditos obtenidos (en la moneda ficticia) y el conjunto de objetos del juego que tiene (ya sea porque los encontró o porque los compró). De los personajes no jugador se guarda el nombre del personaje (texto), su rol (texto), su salud (en %), si es enemigo, y el nivel de inteligencia artificial (de 1 a 10). Tanto los personajes jugador como los no jugador pueden formar un equipo, del que se guarda su nombre y todos sus integrantes. De cada usuario se guarda el nombre de usuario, todas las partidas que ha jugado y el crédito total acumulado (en la moneda ficticia).

Los personajes pueden realizar acciones, y cada acción tiene asociado un nombre, la fecha y hora en la que se completó la acción, el tiempo de reacción del personaje a la hora de ejecutar la acción (en milisegundos), el conjunto de acciones de las que esta acción depende para poder ejecutarse, el espacio en el que se ha realizado la acción y el conjunto de objetos que requiere la acción para poder hacerla. Los personajes también pueden participar en un diálogo, y de los diálogos se guarda el identificador de diálogo y el conjunto de personajes que ha participado. Todos los juegos (incluso los puzzles de habilidad más simples) tienen una historia asociada, ligada a lo que el jugador va haciendo en el juego. La historia está compuesta por hitos argumentales, es decir puntos importantes dentro de la historia del juego a los que el jugador ha de llegar. Cada hito argumental ocurre en un nivel del juego (un nivel puede tener varios hitos asociados) y tiene asociada una acción objetivo que el jugador ha de conseguir realizar. La información que se guarda para cada hito incluye el tema (una misión individual, una misión grupal, una relación personal, un misterio), si el hito es central (es decir, imprescindible para el juego) o si es opcional (se puede añadir o eliminar sin que el juego pierda sentido), la acción a realizar, la fecha y hora límite para realizar la acción (todo a ceros si no hay límite temporal). Todos los hitos argumentales tienen asociadas tres escenas pre-grabadas de animación 3D: la escena de inicio (un cortometraje donde se narra parte de la historia y le dan pistas al jugador de la acción objetivo que ha de realizar) la escena de objetivo cumplido (se confirma al jugador que ha conseguido el objetivo del hito, y se cuenta un trozo de historia que es consecuencia del éxito) y la de objetivo no cumplido (un trozo de historia consecuencia del fracaso), de estas escenas solo se guarda el identificador del video y la duración de la escena. Además un hito argumental puede incluir uno o más diálogos (que se insertan en el juego para guiar al usuario en cierta dirección, o darle pistas).

Los servidores de la empresa recopilan toda esta información y la complementan con otros estadísticos generados de forma automática, como el tiempo medio de reacción del jugador (en milisegundos) en las acciones de la partida actual / en el día actual / en la última semana / en el último mes; o el tiempo extra (en segundos) que el usuario dedica a explorar los espacios en la partida actual / en el día actual / en la última semana / en el último mes, para las escenas animadas el porcentaje de video (en media) que ve el usuario en la partida actual / en el día actual / en la última semana / en el último mes (hay usuarios que verán normalmente los videos enteros, otros que se saltarán el video para ir a la acción, pero esto puede variar por días, según si el usuario tiene tiempo para verlos), etc.

Por lo que nos dicen los expertos se suelen usar varias características para construir un perfil de jugador:

- La **salud** del personaje en la partida actual, que puede ser muy baja ( $< 10\%$ ) , baja ( $< 40\%$ ), normal ( $< 80\%$ ) o elevada ( $\geq 80\%$ ).
- El **tiempo de reacción** medio de las acciones en la partida actual, que puede ser rápido ( $< 0.4$

seg), normal ( $< 0.9$  seg) o lento ( $\geq 0.9$  seg).

- El **tiempo de exploración** medio en las partidas del día actual, que puede ser mínimo ( $< 5$  seg), normal ( $< 50$  seg) o elevado ( $\geq 50$  seg).
- El **tiempo de narración** medio que admite el usuario en las escenas animadas del día actual, que puede ser mínimo ( $< 3\%$ ), bajo ( $< 20\%$ ), normal ( $< 80\%$ ) o elevado ( $\geq 80\%$ ).
- El **nivel de trabajo en equipo**, que está asociado al porcentaje de partidas jugadas e el último mes en las que el usuario ha participado dentro de un equipo, y que puede ser muy bajo ( $< 10\%$ ), bajo ( $< 40\%$ ), normal ( $< 80\%$ ) o elevado ( $\geq 80\%$ ).
- La **interacción con otros personajes**, que está asociada al número de diálogos en los que el usuario ha participado en el último mes, y que puede ser muy baja ( $< 5$ ), baja ( $< 20$ ), normal ( $< 30$ ) o elevada ( $\geq 30$ ).
- La **tipología de juegos** que el usuario ha jugado en el último año, que puede ser variada (un poco de todos los tipos de juego) mayormente de acción (si la mayoría de partidas jugadas son de juegos de tipo arcade o lucha), mayormente de ingenio (si la mayoría de juegos son de tipo puzzle, aventura gráfica o simulador de mundo), o mayormente competitivo (si la mayoría de juegos son de tipo deporte o simulador de vehículo).

A partir de estas características queremos adaptar la partida actual modificando algunos de sus parámetros para ajustar la experiencia de juego a lo que el usuario parece necesitar o preferir:

- *Ajustar el número de enemigos*: si la salud del personaje es alta o el tiempo de reacción es rápido y la tipología de juegos es mayormente de acción o mayormente competitivo entonces hay que incrementar el número de NPCs enemigos; si la salud es normal y el tiempo de reacción es normal o alto y la tipología de juegos es mayormente de acción entonces hay que incrementar el número de NPCs enemigos; si la salud es muy baja o si el tiempo de reacción es lento entonces hay que reducir el número de NPCs enemigos; en los otros casos no modificar los enemigos.
- *Ajustar el nivel de la IA de los enemigos*: si la salud del personaje es alta o el tiempo de reacción es rápido y la tipología de juegos es mayormente de aventuras o mayormente competitivo entonces incrementar el nivel de la IA de los NPCs enemigos; si la salud es normal y el tiempo de reacción es normal o alto y la tipología de juegos es mayormente competitivo entonces hay que incrementar el nivel de la IA de los NPCs enemigos; si la salud es muy baja o si el tiempo de reacción es lento entonces hay que reducir el nivel de la IA de los NPCs enemigos; en los otros casos no modificar la IA de los enemigos.
- *Ajustar misiones de equipo*: si el nivel de trabajo en equipo es bajo y el tiempo de narración es normal o elevado entonces introducir hitos argumentales con tema de misión grupal; si el nivel de trabajo en equipo es elevado y el tiempo de narración es elevado entonces introducir hitos argumentales con tema misión individual; si el tiempo de narración es mínimo entonces eliminar todos los hitos argumentales con tema misión individual o misión grupal que no sean centrales; en el resto de casos no variar.
- *Ajustar misterios*: si la tipología de juegos es mayormente de ingenio y el tiempo de narración es elevado o el tiempo de exploración es elevado entonces introducir hitos argumentales con tema misterio; si el tiempo de narración es bajo o el tiempo de exploración es bajo entonces reducir hitos argumentales con tema misterio; si el tiempo de narración es mínimo entonces eliminar todos los hitos argumentales con tema misterio que no sean centrales; en el resto de casos no variar.
- *Ajustar relaciones*: si el nivel de interacción con otros personajes es bajo o muy bajo y el tiempo de narración es normal o elevado y el tiempo de exploración es normal o elevado entonces introducir hitos argumentales con tema relación personajes; si el nivel de interacción con otros personajes es normal y el tiempo de narración es bajo entonces reducir hitos argumentales con tema relación personajes; si el tiempo de narración es mínimo entonces eliminar todos los hitos argumentales con tema relación personajes que no sean centrales; en el resto de casos no variar.

La salida de este sistema de adaptación ha de sugerir exactamente que modificaciones hay que introducir en el juego (número de enemigos a añadir/eliminar, niveles de IA que aumentar/disminuir, que hitos hay que añadir o eliminar, etc...).

- a. Diseña la ontología del dominio descrito, incluyendo todos los conceptos que aparecen en la



descripción e identificando los atributos más relevantes. Lista que conceptos forman parte de los datos de entrada del problema y que conceptos forman parte de la solución. (Nota: tened en cuenta que la ontología puede necesitar modificaciones para adaptarla al apartado siguiente).

- b. El problema descrito es un problema de análisis. Explica cómo lo resolverías usando clasificación heurística, usando los conceptos de la ontología desarrollada en el apartado anterior. Da al menos 4 ejemplos de reglas para cada una de las fases de esta metodología.
- c. Las características que se usan durante el proceso de adaptación del videojuego no son independientes entre sí. Los tiempos de reacción suelen influir en la salud del personaje (con tiempos de reacción lentos la salud suele empeorar); el tiempo de exploración suele influir en el tiempo de narración (un usuario que dedica mayor tiempo a explorar el entorno suele tener más paciencia o curiosidad por ver los videos enteros, y por ello el tiempo de narración es mayor, mientras que un tiempo de exploración menor suele asociarse a un jugador con menos tiempo disponible o menos curiosidad, y por lo tanto tener un tiempo de narración menor); el nivel de trabajo en equipo influye en la interacción con otros personajes (a mayor trabajo en equipo, mayor interacción con otros personajes). Define el problema de asociación heurística como una red bayesiana expresando en ella al menos las relaciones indicadas no solo en este apartado sino en el resto del enunciado, de forma que todas las características abstractas del problema que hayas definido en el apartado anterior tengan algún tipo de influencia en la solución. Separa bien en el diagrama que variables describen características de problema y cuales describen soluciones. Lista de forma clara los diferentes valores que puede tomar cada variable. Da un ejemplo de tabla de probabilidad de algún nodo, inventándote las probabilidades, pero expresando como influyen los valores de los nodos padre en las probabilidades de los valores de los nodos hijo.

10. Un gran banco quiere ofrecer a sus clientes preferentes un nuevo servicio de asesor personal, un sistema capaz de aconsejar al cliente como invertir de la mejor manera sus ahorros. Para ello el sistema puede combinar toda la información que dispone sobre el cliente y su operativa dentro de la entidad con otra información que se le pregunta al cliente para poder modelar su perfil de ahorrador/inversor y a partir de él darle unas prioridades sobre qué productos podrán ser más interesantes para él o ella. La información que podemos obtener directamente sobre el cliente dentro de las bases de datos de la entidad incluye:

- **Información personal** como su nombre, dirección, fecha de nacimiento, estado civil y profesión;
- El **listado de ingresos en cuenta** (incluido el sueldo) y **gastos regulares de los últimos 3 años**, para cada uno de los movimientos se tiene su fecha y su valor (negativo en el caso de gastos);
- Los **créditos** (incluidas las hipotecas) que tiene con la entidad, de los que se tiene la fecha de la contratación, el importe del préstamo, el plazo de devolución y el interés que se le aplica;
- Los diferentes **productos de ahorro e inversión** que ha contratado con la entidad en el pasado (fondos de pensiones, acciones, fondos de inversión en acciones, fondos de inversión en renta fija, fondos de inversión mixtos, depósitos a corto plazo, depósitos a largo plazo). En todos los casos se tiene la fecha de contratación del producto, la fecha y hora de acceso del cliente en la plataforma web para consultar el estado del producto, y la fecha de cancelación del producto. En el caso de las acciones se tiene además un histórico de compras y ventas que incluye fecha, hora, operación (compra o venta) y el número de acciones involucradas.

La información derivada que obtenemos de la información anterior (o de preguntas al usuario, en caso de que falte información sobre ello en las bases de datos del banco) incluye:

- Su **edad**.
- Si su **perspectiva de ahorro futuro** (la capacidad de ahorro) es favorable o desfavorable .
- Si tiene deudas, cuanta **proporción de su salario necesita para pagar las deudas**.
- Su **actitud respecto a fluctuaciones negativas interanuales** de sus inversiones de un 5%, un 10%, un 20% o más de un 20% (moderada, negativa, muy negativa).
- Su **actitud respecto a la oportunidad de ganancia contra la posibilidad de pérdida** (seguridad ante todo, ganar un poco más compensa si la posibilidad de pérdida es pequeña, la ganancia ante todo).

- **El objetivo del ahorro** (garantizar el capital, mantener la inflación, obtener ganancias sobre la inflación, obtener grandes ganancias).
- **Experiencia en la administración de inversiones** (nada, poca, bastante).
- **El ahorro de impuestos** (indiferente, importante o muy importante).

A partir de ahí se realizará un perfilado del usuario teniendo en cuenta los siguientes parámetros:

- **Riesgo**, que mide el riesgo que puede o quiere asumir el cliente, valorado como bajo, medio, alto y muy alto. El riesgo está asociado a la edad (cuanto más viejo, menos riesgo se puede asumir), a la actitud a las fluctuaciones (cuanto peor, menos riesgo), a la proporción que suponen las deudas respecto al salario (cuantas más deudas, menos riesgo), a la actitud respecto a la oportunidad de ganancia, el objetivo de ahorro (cuantas más ganancias más riesgo) y la experiencia en la administración de inversiones (cuanta más experiencia, más riesgo).
- **Diversificación**, que mide la variedad de valores en los que invertir, valorado como poca, media y mucha. La diversificación está asociada a la actitud respecto a la oportunidad de ganancia, (cuanta más seguridad se quiera respecto a la ganancia, mayor ha de ser la diversificación), la actitud respecto a las fluctuaciones (cuanto más negativa, más diversificación) y la experiencia en la administración de inversiones (cuanta menos experiencia, más diversificación).
- **Plazo**, que mide el tiempo que se debe mantener la inversión, valorado como corto, medio o largo. El plazo depende de la actitud respecto las fluctuaciones (cuanto menos negativa, el plazo puede ser más largo), la perspectiva del futuro ahorro (cuanto mejor sea, más plazo), el objetivo de ahorro (cuantas más ganancias, mayor ha de ser el plazo) y el ahorro de impuestos (se pagan más impuestos cuanto más corto sea el periodo de la inversión).
- **Administración activa**, que mide lo involucrado que debe estar el cliente en la evolución y manejo de sus inversiones, valorada en constante, moderada y poca. Esta característica depende de la experiencia del cliente (cuanta menos experiencia, menos involucrado), el objetivo del ahorro (cuanta mayor la ganancia, más deberá estar involucrado) y la actitud respecto a las fluctuaciones (cuanto más negativa sea a mayores pérdidas, más deberá estar involucrado).
- **Liquidez de la inversión**, que mide la facilidad para convertir la inversión en dinero a corto plazo, valorada como alta, moderada y baja. La liquidez dependerá de la proporción de la deudas (cuantas más deudas, más liquidez), la experiencia en administración de inversiones (cuanto menos experiencia, menos liquidez), la actitud respecto a la pérdida (cuanto menos negativa, menos liquidez es necesaria) y el objetivo de ahorro (cuanto menos ganancia, menos liquidez).

A partir de evaluar estos criterios queremos obtener una prioridad para los diferentes tipos de productos que se pueden ofrecer al cliente: fondos de pensiones, acciones, fondos de inversión (en acciones, renta fija o mixtos) y depósitos (a corto plazo o a largo plazo). Esta prioridad deberá obtenerse como un valor entero entre 0 y 5, indicando la importancia de incluir ese tipo de valor en la cartera de inversión.

La experiencia del dominio nos dice que:

- Los fondos de pensiones son más convenientes cuanto menos riesgo se quiera, son mejores cuando se quiere una buena diversificación, tienen mejor resultado cuanto más largo es el plazo, no requieren una administración activa y no tienen liquidez.
  - Las acciones son más recomendables para inversiones con más riesgo, siempre que se pueda hacer una administración activa, permiten menos diversificación, se puede invertir en ellas a cualquier plazo, pero son más recomendables a corto y tienen total liquidez.
  - Los fondos de inversión permiten una mayor diversificación, requieren poca administración y suelen tener liquidez alta o moderada. Los de acciones son más recomendables a largo plazo cuando se quiere menos riesgo, los de renta fija son más recomendables cuanto menos riesgo y los mixtos equilibran el riesgo cuando el plazo es más corto.
  - Los depósitos tienen menor liquidez cuanto mayor sea el plazo, no requieren ninguna administración, son recomendables cuanto menos riesgo y diversificación se quiera
- (a) Diseña la ontología del dominio descrito, incluyendo todos los conceptos que aparecen en la descripción e identificando los atributos más relevantes. Lista que conceptos forman parte de los

datos de entrada del problema y que conceptos forman parte de la solución. (Nota: tened en cuenta que la ontología puede necesitar modificaciones para adaptarla al apartado siguiente).

- (b) El problema descrito es un problema de análisis. Explica cómo lo resolverías usando clasificación heurística, usando los conceptos de la ontología desarrollada en el apartado anterior. Da al menos 3 ejemplos de reglas para cada una de las fases de esta metodología.
  - (c) Las características del perfil que se usan durante el proceso de toma de decisiones no son independientes entre sí. El plazo de la inversión influencia su liquidez (a mayor plazo, menor liquidez), y el riesgo influencia la diversificación (a menor riesgo, mayor diversificación). Define el problema de asociación heurística como una red bayesiana expresando en ella al menos las relaciones indicadas en el enunciado, de forma que todas las características abstractas del problema que hayas definido en el apartado anterior tengan algún tipo de influencia en la solución. Separa bien en el diagrama que variables describen características de problema y cuales describen soluciones. Lista de forma clara los diferentes valores que puede tomar cada variable. Da un ejemplo de tabla de probabilidad de algún nodo, inventándote las probabilidades, pero expresando como influyen los valores de los nodos padre en las probabilidades de los valores de los nodos hijo.
11. El elegir una tarifa de telefonía móvil es una tarea realmente complicada dado el gran número de diferentes tarifas que los operadores de telefonía ofrecen. Para facilitarlo el portal *SuperComparador.com* quiere desarrollar un sistema que permita recomendar la tarifa más adecuada a cada usuario de las disponibles en el mercado.

Para desarrollar el sistema el portal hace rellenar al usuario un formulario sobre su uso mensual del móvil, como por ejemplo cuantas llamadas tienen como destinatario a amigos y familiares, cuantas tienen como destinatario a contactos del trabajo y cual es más o menos la duración de cada tipo, cuantas llamadas hace en horario de mañana, tarde y noche, cuantas llamadas hace en días laborables y cuantas en fin de semana, si usa internet en el móvil, y en caso positivo el número de GB de datos que en media suele consumir al mes, cuantos mensajes SMS/MMS envía, a cuantos operadores diferentes son sus llamadas, cuantas tarifas diferentes de móvil ha tenido en los últimos tres años, las características de cada una de esas tarifas y a qué operador pertenecían, ...

A partir de esta información se puede elaborar un perfil de las características del usuario basándonos en un conjunto de atributos: perfil particular (bajo, medio, alto), perfil laboral (bajo, medio, alto), uso de teléfono (bajo, moderado, alto, intensivo), flexibilidad horaria (baja, media, alta), destino de llamadas (al operador, a varios operadores), uso de mensajería (bajo, medio, alto), uso de internet (nulo, bajo, medio, intensivo), fidelidad (alta, baja).

Las diferentes tarifas de móvil también se pueden describir mediante una serie de características que nos permitirán identificar mejor que conjunto de tarifas son las más adecuadas para el usuario. Una tarifa pertenece a un operador de telefonía y puede ser descrita por su cuota mensual (ninguna, baja, media, alta), la exigencia de un consumo mínimo (no, bajo, medio, alto), el coste de establecimiento de llamada (ninguno, bajo, medio, alto), el coste de la llamada por minuto al operador (ninguno, bajo, medio, alto), el coste por minuto a otros operadores (ninguno, bajo, medio, alto), descuento por volumen de llamadas (si, no), minutos de llamada incluidos en la cuota mensual (medio, alto, muy alto), flexibilidad de horario con coste reducido (ninguna, baja, media, alta), velocidad de internet (no tiene, 2G, 3G, 4G), los GB de datos incluidos en la tarifa (ninguno, menos de 2 GB, entre 3 y 4 GB, más de 4 GB), coste de los mensajes (bajo, medio, alto), exigencia de permanencia (alta, media, baja).

A partir del perfil del usuario podemos asociar los valores de las características de la tarifa más adecuada, por ejemplo, si el perfil particular del usuario es alto y hace un uso alto, puede convenirle una tarifa que tenga una exigencia de consumo mínimo alta y que tenga una cuota mensual media. Si el perfil particular es alto, el uso del teléfono es bajo y con una flexibilidad horaria baja puede convenirle una tarifa con ninguna cuota mensual y una exigencia de consumo baja. Si el uso del teléfono es intensivo necesitará una flexibilidad de horario alta y con los minutos de llamada incluidos en la tarifa altos o muy altos o bien con un descuento por volumen de llamadas. Si el perfil laboral es alto le conviene una tarifa con un coste de llamada a otros operadores bajo o medio. Si la fidelidad del usuario es alta y su

uso es bajo podrá tener una tarifa con una cuota mensual baja y con una exigencia de permanencia alta.

Una vez determinados los valores de las características de la tarifa más adecuada se puede identificar la tarifa concreta a partir de los valores concretos de consumo del usuario y los precios y características específicas de las tarifas ofrecidas por los operadores de telefonía (que tiene el portal en su base de datos).

- (a) Diseña la ontología del dominio descrito, incluyendo todos los conceptos que aparecen en la descripción e identificando los atributos más relevantes. Lista que conceptos forman parte de los datos de entrada del problema y que conceptos forman parte de la solución. (Nota: tened en cuenta que la ontología puede necesitar modificaciones para adaptarla al apartado siguiente).
- (b) El problema descrito es un problema de análisis. Explica cómo lo resolverías usando clasificación heurística, usando los conceptos de la ontología desarrollada en el apartado anterior. Da al menos 3 ejemplos de reglas para cada una de las fases de esta metodología. Inventa algún ejemplo concreto de tarifa si te hace falta.
- (c) Las características que se utilizan para caracterizar a los usuarios y los tipos de tarifas no son independientes entre si. Por ejemplo: un uso alto de mensajería suele implicar la recomendación de una tarifa con baja velocidad de internet y viceversa, un uso intensivo de internet suele implicar la recomendación de tarifas con altas velocidades de internet y con más de 2 GB de datos incluidos, una flexibilidad horaria alta implica la recomendación de tarifas con una flexibilidad de horario de coste reducido media o alta, etc.

Define el problema de asociación heurística como una red bayesiana expresando en ella al menos las relaciones indicadas en el enunciado, en este apartado y alguna más, de forma que todas las características abstractas del problema que hayas definido en el apartado anterior tengan algún tipo de influencia en la solución. Separa bien en el diagrama que variables describen características de problema y cuales describen soluciones. Da un ejemplo de tabla de probabilidad de algún nodo, inventándote las probabilidades, pero expresando como influyen los valores de los nodos padre en las probabilidades de los valores de los nodos hijo.

12. La agencia de viajes *Viajando Voy* quiere renovar su página de web y su app con un sistema inteligente de recomendación que ayude a sus usuarios registrados a escoger entre su extenso catálogo de viajes. Para hacer esta recomendación el sistema usa no solo la petición de viaje que realiza un usuario (que se identifica con un nombre de usuario y password) sino también el historial de viajes anteriores que ha hecho el usuario en esa agencia.

El catálogo de la agencia consta de destinos, y cada uno tiene asociado el nombre de la población, el país en el que se encuentra, el nivel de riesgo (del 1 a 10, tiene en cuenta la posibilidad de robos, de accidentes, de atentados), la moneda de pago, los medios de transporte disponibles (avión, tren, barco, autobús, taxi, vehículo de alquiler, etc.) los lugares de transporte (aeropuertos, puertos, estaciones de tren, estaciones de autobús, agencias de alquiler de coches, etc.) los alojamientos disponibles (por ahora solo hoteles, hostales y albergues) y las actividades (de ocio, cultura, deporte o experiencias) que se pueden hacer en ese destino.

De los lugares de transporte se guarda su nombre, dirección completa, y las coordenadas GPS, y en el caso de los aeropuertos se guarda también el código internacional de tres letras (por ej "BCN" para Barcelona). De los aviones se guarda la compañía aérea, el identificador de vuelo, el aeropuerto de origen y el aeropuerto final, las escalas (aeropuertos en los que se hace una parada intermedia en el viaje), la hora de salida y la hora de llegada, y el precio del trayecto (en euros); en el caso de los trenes se guarda la compañía, la estación de tren de origen y la estación de final, la hora de salida y la hora de llegada, y el precio del trayecto (en euros); en el caso de los autobuses se guarda la compañía, la estación de autobuses de origen y la estación de final, la hora de salida y la hora de llegada, y el precio del trayecto (en euros); en el caso de los barcos se guarda la compañía, el puerto de origen y el puerto final, las escalas (puertos en los que se hace una parada intermedia en el viaje), la hora de salida y la hora de llegada, y el precio del trayecto (en euros); en el caso de los taxis se guarda la dirección de

origen, la dirección de llegada y el precio por kilómetro (en euros); en el caso de los vehículos de alquiler se guarda la oficina de alquiler de recogida, la oficina de alquiler de entrega, la fecha de recogida, la fecha de entrega, la marca y modelo del vehículo, el número de plazas, el precio base de alquiler por día, el precio por kilómetros extra recorridos y el precio del seguro de accidentes.

Cada alojamiento (hotel, hostel o albergue) tiene asociado un nombre, el número de estrellas que tiene (de 1 a 5 estrellas) y cada una de las habitaciones. Para cada habitación se guarda el identificador de habitación, el número de camas individuales, el número de camas dobles, el número máximo de personas, si tiene baño propio o no, si tiene televisión o no, si tiene wifi o no y si tiene mini-bar o no. Hay un tipo especial de alojamiento: los cruceros, que disponen de las características de Hotel y Barco a la vez, ya que son un alojamiento y también un medio de transporte.

De las actividades se tiene el nombre, la dirección, una descripción corta, el nivel de riesgo (de 1 a 10), y si es una actividad para hacer de forma individual, en pareja o en grupos. Hay cuatro tipos de actividades: de ocio, culturales, deporte y experiencias. Como actividades de ocio se tiene la playa y los parques temáticos (de estos se guarda hora de apertura, hora de cierre y precio); de actividades culturales se tienen los museos y los monumentos (y de ambos se guarda hora de apertura, hora de cierre y precio); de actividades de deporte se tiene el trekking y el esquí (y en ambas se tiene asociado el nivel técnico necesario, de 1 a 10); y se tiene tres tipos de experiencias: románticas, culinarias o safari (en todas ellas se guarda la hora de inicio, la duración y el precio).

Una petición de viaje introducida por el usuario incluye un precio máximo, un número mínimo de días, un número máximo de días, un intervalo de fechas entre las que se puede realizar ese viaje y cada uno de los viajeros que irán en el viaje (de los que se guarda nombre, edad, el estado físico -de 1 a 10-, si tienen alguna alergia o no, y si tienen movilidad reducida o no). El usuario tiene asociados todos los viajes que ha hecho en el pasado, y de cada viaje se guarda la fecha de inicio, la fecha de fin, el precio total del viaje, los viajeros que fueron en el viaje, los destinos, los medios de transporte, los alojamientos y las actividades realizadas.

Para recomendar un viaje esta empresa usa siete características:

- el **presupuesto**, que tiene en cuenta el ratio entre el precio máximo que se quiere gastar el usuario en el viaje y el número de viajeros, y que puede ser bajo ( $< 200$  euros/persona), medio (entre 200 y 500 euros/persona) y alto (más de 500 euros/persona);
- la **tipología de viajeros**, distinguiendo entre individuo, grupo sin niños, pareja o grupo con niños;
- la **duración** del viaje, distinguiendo entre cuatro clases: de 1 a 3 días, de 4 a 7 días, de 7 a 14 días o más de 14 días;
- el **nivel de actividad** que analiza las actividades que ha realizado el usuario en sus viajes anteriores y el estado físico de los viajeros del nuevo viaje y que toma tres valores: baja (la mayoría de actividades en viajes pasados fueron calmadas o hay viajeros en la petición con movilidad reducida o menores de 6 años o mayores de 65 años), media (la mayoría de actividades en viajes pasados fueron de un nivel intermedio o hay viajeros en la petición entre los 7 y los 13 años o entre los 55 y los 65 años) o alta (en caso contrario);
- el **nivel de riesgo** que analiza el nivel de riesgo de los destinos y las actividades realizadas por el usuario en sus viajes anteriores y que toma tres valores: bajo (la media del nivel de riesgo de la mayoría de actividades y destinos anteriores está entre 1 y 3), medio (la media está entre 4 y 7) o alto (la media está entre 8 y 10);
- la **curiosidad cultural** que analiza la proporción de actividades culturales realizadas en viajes pasados respecto al total y que puede tomar tres valores: alta (más de un 70% de las actividades anteriores fueron culturales), media (entre un 30% y un 70% de actividades culturales) y baja (por debajo del 30% de actividades);
- la **temporada** del viaje, que puede ser primavera, verano, otoño o invierno, y que se obtiene a partir del rango de fechas de viaje en la petición del viajero;

A partir de estas características se determinan dos características más:

- el **arquetipo de viaje**, distinguiendo entre ocho arquetipos (entre paréntesis las características relevantes): "Viaje con niños"(para grupos con niños, actividad media o alta, riesgo bajo);

"Monumentos y museos"(curiosidad cultural alta, actividad baja o media); "Sol y playa"(actividad baja o media, riesgo bajo, en cualquier temporada ya que hay destinos de playa en el mundo todo el año); "Experiencia Culinaria"(curiosidad cultural media o alta, actividad baja, riesgo bajo); "Aventura Trekking"(actividad alta, riesgo medio, temporada de primavera u otoño); "Aventura Safari"(individuo o grupo sin niños, actividad alta, riesgo alto); "Disfruta de la Nieve"(actividad alta, riesgo medio, temporada de invierno o verano, ya que será invierno en el hemisferio sur); "Viaje romántico"(una pareja de viajeros, actividad baja, riesgo bajo);

- el **destino recomendado**, distinguiendo entre 11 destinos (entre paréntesis las características relevantes): "Viaje de proximidad"(1 a 3 días o presupuesto bajo); "Viaje por Europa Central"(4 a 7 días, riesgo bajo, presupuesto medio); "Viaje por el Mediterraneo"(4 a 7 días, riesgo bajo, presupuesto medio o alto); "Viaje por el Magreb"(4 a 7 días, riesgo medio, presupuesto medio); "Viaje por Africa"(7 a 14 días o >14 días, presupuesto alto, riesgo medio o alto); "Viaje por Oriente Medio"(7 a 14 días, presupuesto alto, riesgo alto); "Viaje por Asia"(7 a 14 días o >14 días, presupuesto alto, riesgo medio, temporada no sea verano para evitar los monzones); "Viaje por Oceania o Polinesia"(7 a 14 días o >14 días, presupuesto alto, riesgo medio); "Viaje por América del Norte"(7 a 14 días o >14 días, presupuesto alto, riesgo medio o bajo); "Viaje por América Central o del Sur"(7 a 14 días o >14 días, presupuesto alto, riesgo medio o alto); "Viaje al Caribe"(7 a 14 días, presupuesto alto, riesgo medio o bajo);

La salida del sistema es una lista de viajes recomendados que se genera llamando a la función `Viaje listar_viajes_recomendados(Arquetipos_Viaje, Destinos_recomendados, precio_maximo, min_dias; max_dias)` y se muestra por pantalla para que el usuario escoja una opción. Al obtener el viaje escogido se llama a una segunda función `Actividades listar_actividades_extra(Viaje, Arquetipos_Viaje, precio_maximo)` que muestra al usuario actividades extra que puede añadir a su viaje para finalizar su reserva.

- Diseña la ontología del dominio descrito, incluyendo todos los conceptos que aparecen en la descripción e identificando los atributos más relevantes. Lista que conceptos forman parte de los datos de entrada del problema y que conceptos forman parte de la solución. (Nota: tened en cuenta que la ontología puede necesitar modificaciones para adaptarla al apartado siguiente).
  - El problema descrito es un problema de análisis. Explica cómo lo resolverías usando clasificación heurística, usando los conceptos de la ontología desarrollada en el apartado anterior. Da al menos 4 ejemplos de reglas para cada una de las fases de esta metodología.
  - Las características que se usan durante la recomendación de viajes no son independientes entre sí. El presupuesto influye en la duración del viaje, y la tipología de viajeros influye en el nivel de actividad y el nivel de riesgo. Define el problema de asociación heurística como una red bayesiana expresando en ella al menos las relaciones indicadas en este apartado y el resto del enunciado, de forma que todas las características abstractas del problema que hayas definido en el apartado anterior tengan algún tipo de influencia en la solución. Separa bien en el diagrama que variables describen características de problema y cuales describen soluciones. Lista de forma clara los diferentes valores que puede tomar cada variable. Da un ejemplo de tabla de probabilidad de algún nodo, inventándote las probabilidades, pero expresando como influyen los valores de los nodos padre en las probabilidades de los valores de los nodos hijo.
- La cadena de supermercados *HIPERFUR* nos pide que diseñemos un SBC orientado a la creación de folletos de ofertas de productos personalizados para cada cliente. Cada folleto consta de 8 páginas y en cada página se pueden colocar 9 ofertas diferentes, que provienen de una base de ofertas a la que tendrá acceso el SBC. Toda oferta tiene una fecha de inicio y de fin, y un producto o productos asociados. Hay dos tipos de ofertas:
    - *descuentos*, que tienen entre otros datos el porcentaje de descuento, e información sobre si son descuentos directos (el cliente paga el precio rebajado) o indirectos (el cliente recibe un vale de compra).
    - *promociones*, que pueden ser packs de productos diferentes (por ejemplo, un queso y un jamón) u ofertas al llevarse varias unidades del mismo producto (2x1, 3x2, 4x3).

Los productos pueden ser de diferente tipología: alimentación (leche, huevos, pescado, carne, bebidas...), belleza (cremas, maquillaje, after-shave...), menaje (sartenes, ollas, vasos, cubertería...), limpieza del hogar (jabón de ropa, lavavajillas, quitamanchas, mocho, escoba...). Además los productos pueden pertenecer a una marca blanca (precio muy bajo), a una marca generalista (precio medio) o a una marca premium (precios muy altos). Este sistema obtendrá la información de los hábitos de compra de un cliente (una lista de las ofertas y los productos individuales que ha comprado en los últimos 6 meses donde, para cada oferta y para cada producto individual, se dice cuantas unidades ha comprado). Además los clientes pueden (a través de la página web) añadir una lista de productos preferidos, y/o una lista de restricciones que se pueden aplicar a un producto concreto (ej: no quiero el jabón "Pixan plus") o a un subtipo de producto (ej: no me gustan las leches de soja). Se quiere construir este SBC de manera que, en base a toda la información disponible sobre el cliente, seleccione un conjunto de ofertas adecuadas para ese tipo de cliente y las asigne a alguna de las 8 páginas del folleto. El director de marketing de *HIPERFUR* nos recomienda que incluyamos las siguientes características abstractas como base de la selección: el nivel de **gasto mensual** (alto, medio, bajo) que servirá para calcular el importe total de los productos que se coloquen en el folleto; la **calidad** de los productos (mayoría premium, mayoría normal, mayoría marca blanca) que servirá para seleccionar, dentro de un (sub)tipo de producto, el producto adecuado al patrón de compra del cliente; la **cantidad\_alimentación** (exagerada, mucha, normal, poca, limitada), que servirá para decidir que porcentaje del folleto se dedica a productos alimenticios (y en los casos extremos -exagerada, limitada- se colocarán preferentemente productos de alto valor nutricional); la **cantidad\_belleza** (narcicista, normal, dejado/a) que servirá para decidir que porcentaje del folleto se dedica a productos de belleza; la **cantidad\_menaje** (exagerado/a, normal, minimalista) que servirá para decidir que porcentaje del folleto se dedica a productos de menaje; y la **cantidad\_limpieza** (obseso/a, normal, sucio/a) que servirá para decidir que porcentaje del folleto se dedica a productos de limpieza.

- (a) Diseña la ontología del dominio descrito, incluyendo todos los conceptos que aparecen en la descripción e identificando los atributos más relevantes. Lista qué conceptos forman parte de los datos de entrada del problema y qué conceptos forman parte de la solución. (Nota: tened en cuenta que la ontología puede necesitar modificaciones para adaptarla al apartado siguiente).
- (b) El problema descrito es un problema de análisis. Explica cómo lo resolverías usando clasificación heurística, usando los conceptos de la ontología desarrollada en el apartado anterior. Da al menos 3 ejemplos de reglas para cada una de las fases de esta metodología.
- (c) Las características que se utilizan para determinar las ofertas a incluir en el folleto no son independientes entre sí. Por ejemplo: la cantidad de productos de belleza suele ser inversamente proporcional a la cantidad de alimentación (cuanto más cuida su aspecto, menos come), la calidad está influida por el nivel de gasto mensual y las cantidades de cada tipo de producto. El tipo de ofertas a incluir viene influenciado por el nivel de gasto y la cantidad (por ejemplo, en caso de niveles de gasto bajos o cantidades bajas suele ser mejor incluir descuentos, mientras que para clientes que compren grandes cantidades o con niveles de gasto elevado lo mejor es recomendar promociones multi-producto, etc).

Define el problema de asociación heurística como una red bayesiana expresando en ella al menos las relaciones indicadas y alguna más. Da un ejemplo de tabla de probabilidad de algún nodo, inventándote las probabilidades, pero expresando como influyen los valores de los nodos padre en las probabilidades de los valores de los nodos hijo.

14. La empresa de pirotécnica FantasIA quiere diseñar un sistema inteligente de gestión de pedidos de material a través de su página web que ayude al cliente a escoger el material pirotécnico que desea recomendándole el más adecuado según sus preferencias, el evento y el lugar donde se celebrará. Coincidiendo con las fiestas de San Juan de este año quiere probar un primer prototipo del sistema y ver si gusta a sus clientes.

Los usuarios de este sistema podran ser empresas o ayuntamientos. De las empresas podemos obtener información sobre si es una empresa *pequeña*, *mediana* o *grande*, y de los ayuntamiento podemos saber si se trata de una *metrópoli* (una ciudad grande), una *ciudad pequeña*, una *villa* o un *pueblo* (cuanto más grande es la empresa o el ayuntamiento más puede invertir en material pirotécnico). Tanto en

el caso de las empresas como de los ayuntamientos el sistema puede acceder al histórico de pedidos para saber que tipo de material han solicitado en el pasado y usarlo en la recomendación. Del evento obtendremos información sobre el número de asistentes que se preveen y el tipo de espacio en el que se va a usar el material (*espacio abierto, espacio forestal, parque, plaza urbana, calle...*). Para todos los espacios sabremos los m<sup>2</sup> de superficie libre de los que se dispone, y en el caso de los espacios forestales y los parques sabremos además el *grado de humedad* de la vegetación. También interesa que el usuario haga un listado de los objetos y estructuras en las cercanías del lugar que podrían recibir algún daño (*edificios, naves industriales, mobiliario urbano, coches*) dando el número de objetos de cada tipo, su distancia al lugar en el que se usará el material y, en el caso de las naves industriales, si contienen productos altamente inflamables en su interior. El sistema preguntará si se tiene contratada una *dotación de bomberos* o no, y si la persona que va a usar el material es un *especialista pirotécnico* certificado o no. Accediendo a un servicio web el sistema obtendrá además una *previsión meteorológica* de la zona en la que se celebrará el evento (temperatura máxima y mínima, humedad relativa, velocidad del viento, probabilidad de lluvia). Y obviamente en todo pedido tendremos el *presupuesto máximo* que quiere gastar al cliente.

A partir de toda esta información el sistema ha de recomendar una serie de productos de su catálogo de material pirotécnico: *material básico* (pequeños cohetes y petardo que puede manipular cualquier persona), *truenos* (petardos de una cierta potencia), *cohetes* (solo suben y explotan), *luces de color* y *luces fantasía* (lo más avanzado, al abrirse dibujan formas en el aire). El material básico es lo más barato y menos peligroso, mientras que las luces fantasía son lo más caro, y ambos tipos de luces son el material más peligroso. Para facilitar la recomendación, la empresa ya tiene preparados unos surtidos o packs de productos: el **pack fantasía** (donde predominan las luces fantasía y de colores), el **pack color** (donde predominan las luces de color y los cohetes), el **pack de cohetes** (con cohetes que hacen diferentes tipos de explosión, muy usados en las fiestas de poblaciones pequeñas), el **pack de truenos** (con mucho éxito en la Comunidad Valenciana) y el **pack básico** (muy solicitado en fiestas organizadas por pequeñas empresas donde se deja que los trabajadores ‘quemen’ material). También tienen un **surtido variado** con un poco de todo (excepto luces fantasía y, según el presupuesto, luces de color).

Además, los expertos del departamento de ventas nos cuentan que, desde siempre, para realizar sus recomendaciones a los clientes se evalúa en cada pedido cada una de las siguientes características:

- **presupuesto**, que se basa en el tipo de cliente y en el número de asistentes para clasificar el presupuesto en *generoso, correcto, justito, rácano*. Por poner algunos ejemplos, un presupuesto de 4000 euros será considerado generoso para un pueblo de 20 habitantes, correcto para una villa de 2000 habitantes, justito para una ciudad de 100000 habitantes y rácano para una metrópoli de 1200000 habitantes.
- **preferencias**, que basándose en el histórico de pedidos indica el tipo de producto que el cliente pide más: *de todo un poco, más básico, más cohetes, más truenos, más color, más fantasía*.
- **peligro incendio**, que mide el grado de peligro (*extremo, importante o menor*) de que se produzca un incendio en base a la proximidad de bosques, coches, estaciones de servicio o naves industriales con productos muy inflamables. El grado de peligro se incrementa con la presencia de viento en la zona y grados de humedad bajos, y se decrementa con la presencia en el evento de una dotación de bomberos.
- **peligro daños materiales**, que mide el grado de peligro (*extremo, importante o menor*) de que se produzcan daños a objetos u estructuras en base a la proximidad de edificios, coches ... al lugar donde se usará el material pirotécnico. Este grado de peligro disminuye si se ha contratado a un especialista pirotécnico.
- **peligro daños personales**, que mide el grado de peligro (*extremo, importante o menor*) de que se produzcan daños a personas en base al número de personas que asistirán al evento y a los m<sup>2</sup> de espacio de los que se dispone. Este grado de peligro disminuye si se ha contratado a un especialista pirotécnico.

Es importante aclarar que los niveles de peligro antes mencionados no dependen del producto que



finalmente se va a recomendar al cliente. Se trata de niveles de peligro *a priori* computados únicamente a partir de las características del cliente, el evento y el lugar.

A partir de la evaluación de estos criterios queremos obtener una recomendación que diga al usuario de la web que tipo de pack es el más adecuado para su evento y la cantidad de productos de cada tipo que habrá finalmente en el pack.

Sabemos que las características abstractas tienen diferentes relaciones con estos productos. Por ejemplo, las preferencias marcan de una forma abstracta el tipo de pack que suele escoger el cliente, los pack color y fantasía suelen estar contraindicados si hay niveles extremos de peligro en alguno de los criterios, niveles generosos de presupuesto suelen llevar la recomendación hasta el máximo nivel de pack que sea adecuado para las características del evento, ...

- (a) Diseña la ontología del dominio descrito, incluyendo todos los conceptos que aparecen en la descripción e identificando los atributos más relevantes. Lista que conceptos forman parte de los datos de entrada del problema y que conceptos forman parte de la solución. (Nota: tened en cuenta que la ontología puede necesitar modificaciones para adaptarla al apartado siguiente).
- (b) El problema descrito es un problema de análisis. Explica cómo lo resolverías usando clasificación heurística, usando los conceptos de la ontología desarrollada en el apartado anterior. Da al menos 3 ejemplos de reglas para cada una de las fases de esta metodología.
- (c) Las características que se utilizan para caracterizar el pedido y el pack recomendado no son independientes entre si. Por ejemplo: el peligro de incendio influye sobre el peligro de daños materiales, y éste sobre el peligro de daños personales; las preferencias influyen directamente sobre la elección de cada tipo de pack, etc.

Define el problema de asociación heurística como una red bayesiana expresando en ella al menos las relaciones indicadas y alguna más, de forma que todas las características tengan algún tipo de influencia en la solución. Da un ejemplo de tabla de probabilidad de algún nodo, inventándote las probabilidades, pero expresando como influyen los valores de los nodos padre en las probabilidades de los valores de los nodos hijo.

15. El telescopio Kepler<sup>1</sup> es un telescopio orbital usado para el descubrimiento de planetas extrasolares. Los datos de este telescopio y los de otros observatorios tanto terrestres como orbitales se pueden utilizar para estimar la habitabilidad de los planetas descubiertos. Una organización que prefiere mantenerse anónima nos ha encargado un SBC que permita planificar la próxima emigración masiva de toda la población terrestre a los planetas descubiertos.

Los datos de los que disponemos incluyen información sobre el planeta candidato, el resto de planetas y la estrella del sistema planetario y otros astros y sistemas candidatos en la cercanía (unos 100 parsecs). Cada sistema planetario tiene asociada su distancia a la Tierra (en años-luz) y está compuesto por una estrella y varios planetas (se descartan sistemas de más de una estrella, suelen ser inhabitables). De cada estrella guardamos su radio (medido en soles: por debajo de 0.4 soles es una estrella enana, por encima de 20 soles es una estrella gigante). Los planetas pueden ser rocosos o gigantes gaseosos. De los planetas rocosos se tiene su tamaño relativo a la Tierra (medido en tierras) y la distancia máxima y mínima de la órbita del planeta a su estrella (medida en unidades astronómicas -UA-). Del planeta candidato (siempre rocoso) se guarda otra información adicional (como el porcentaje de oxígeno y de hidrocarburos en la atmósfera), tal y como veremos a continuación. De los gigantes gaseosos se tiene su tamaño relativo a Júpiter (medido en júpiteres) y la distancia máxima y mínima de la órbita del planeta a su estrella (medida en unidades astronómicas -UA-). También tenemos información de las supernovas cercanas (a máximo 10 parsecs de distancia), de ellas guardamos su tipo (Ia, Ib, II-b, II-L, II-P, II-n) y su distancia (en parsecs) al sistema planetario. Tenemos información de las estrellas gigantes cercanas (a máximo 10 parsecs de distancia) ya que son estrellas con alta probabilidad de convertirse en supernovas en el futuro, de ellas guardamos su masa (medida en masas solares) y su distancia (en parsecs) al sistema planetario candidato. Finalmente tenemos información de otros

<sup>1</sup><http://kepler.nasa.gov/>

sistemas planetarios en la cercanía que puedan ser candidatos para ser habitables, de ellos guardamos la misma información que para el sistema planetario candidato.

De toda esa cantidad de datos los expertos nos indican que hay información que es especialmente relevante a la hora de evaluar un planeta candidato:

- El tamaño del planeta candidato relativo al tamaño de la Tierra (entre 0.8 y 0.9 tierras es un planeta ligero, hasta 1.25 tierras es normal, entre 1.25 y 1.5 es pesado)
- La distancia máxima y mínima de la órbita del planeta candidato a su estrella (considerando la distancia medida en unidades astronómicas (UA), entre 0.7 y 0.8 UA es considerado caluroso, hasta 1.3 UA es considerado normal y entre de 1.3 y 1.5 es considerado frío)
- Porcentaje de oxígeno en la atmósfera del planeta candidato (un valor entre el 12% y el 15% se considera aceptable, un porcentaje superior al 25% se considera una atmósfera reactiva, en otro caso es respirable)
- El porcentaje de hidrocarburos en la atmósfera del planeta candidato (con un porcentaje por debajo del 1% es una atmósfera ligera, hasta el 2% se considera normal y hasta el 3% es una atmósfera densa)
- El radio de la estrella del sistema planetario candidato relativo al sol (entre 0.7 y 0.8 sol es una estrella fría, hasta 1.2 sol es una estrella normal, hasta 1.5 sol es una estrella caliente)
- Número de gigantes gaseosos en el sistema (el rango normal es entre 4 y 7 gigantes gaseosos, por debajo es bajo, por encima es alto)
- Tamaño del mayor gigante gaseoso relativo a Júpiter (un tamaño por debajo de 2 Júpiter es ideal, hasta 5 Júpiter es bueno y por encima molesto)
- La distancia mínima entre las órbitas de los planetas por encima y debajo del candidato (para el planeta en la órbita superior, una distancia por debajo de las 2 UA se considera malo, hasta las 3 UA se considera aceptable y por encima ideal, para el planeta en la órbita inferior una distancia por debajo de 0.3 UA se considera malo, hasta las 0.5 UA se considera aceptable y por encima es ideal)
- Número de supernovas en un radio de 10 parsecs (cero o una es seguro, hasta dos es peligroso, más es muy peligroso)
- Número de estrellas gigantes en un radio de 10 parsecs (cero o una es seguro, hasta dos es peligroso, más es muy peligroso)
- Cercanía a otros sistemas solares candidatos (si hay alguno más cerca de 20 parsecs es ideal, hasta 40 parsecs aceptable, mas lejos es aislado)

Nuestro análisis nos ha llevado a determinar que existen tres criterios a partir de los cuales podemos dar un diagnóstico:

- Hospitalidad planetaria, que nos califica lo hospitalario que es el planeta descubierto, valorado en inhabitable, muy dura, dura, soportable e ideal. Está asociada al tamaño del planeta, a la órbita, al porcentaje de oxígeno y al de hidrocarburos.
- Hospitalidad del sistema, que nos califica lo hospitalario que es el sistema en el que está el planeta, valorado en hostil, inestable, soportable y segura. Está asociada al radio de la estrella, al número de gigantes gaseosos, el tamaño del mayor gigante gaseoso y las distancias a las órbitas de los planetas por encima y por debajo.
- Hospitalidad de la región, que nos califica lo hospitalaria que es la región que rodea al sistema planetario, valorada en peligrosa, aceptable y segura. Está asociada al número de supernovas, el número de estrellas gigantes y la cercanía de otros sistemas candidatos.

La valoración de estos criterios será mejor cuantas más circunstancias favorables haya y cuanto mejor valoración tengan. Por ejemplo, un tamaño de planeta pesado con una distancia de órbita máxima y

mínima calurosa, una atmósfera de hidrocarburos densa y una atmósfera de oxígeno reactiva hará un planeta inhabitable, pero si la atmósfera de oxígeno es respirable lo hará muy duro.

A partir de estas características queremos determinar la supervivencia de la población que se traslade a ese planeta en una escala de enteros del 0 al 10, donde 0 significa misión suicida y 10 un viaje al paraíso.

La *experiencia* en el dominio nos dice que:

- La hospitalidad de la región tiene un peso moderado en el éxito, e incluso una zona segura no aportaría más de tres puntos de la valoración. Lo que si sabemos es que una zona peligrosa no permite dar una valoración total superior a cinco.
- La hospitalidad del sistema tiene un peso superior y aportaría hasta cinco puntos de la valoración. Sabemos que una hospitalidad segura permite una valoración total de al menos cinco si la región es por lo menos aceptable.
- La hospitalidad planetaria aportaría hasta seis puntos de la valoración. Lo que sabemos es que un planeta inhabitable no permite una valoración total mayor que uno. Una valoración ideal aportaría el máximo de valoración posible para este criterio, si la hospitalidad del sistema es al menos soportable y la de la región es al menos aceptable.

El valor final de la supervivencia se calcula a partir de las puntuaciones obtenidas para los tres criterios, combinándolas según las restricciones que imponen para obtener un valor en la escala entera de 0 a 10.

- (a) Diseña la ontología del dominio descrito, incluyendo todos los conceptos que aparecen en la descripción e identificando los atributos más relevantes. Lista que conceptos forman parte de los datos de entrada del problema y que conceptos forman parte de la solución. (Nota: tened en cuenta que la ontología puede necesitar modificaciones para adaptarla al apartado siguiente).
- (b) El problema descrito es un problema de análisis. Explica cómo lo resolverías usando clasificación heurística, usando los conceptos de la ontología desarrollada en el apartado anterior. Da al menos 3 ejemplos de reglas para cada una de las fases de esta metodología.
- (c) Para aumentar la capacidad de trabajar con datos incompletos la empresa nos pide sustituir las reglas de asociación heurística por una red bayesiana que, en vez de calcular la supervivencia potencial de los humanos en el planeta candidato como un número entre 1 y 10, lo haga calculando la probabilidad de supervivencia o de no supervivencia en ese planeta.

Define el problema de asociación heurística como una red bayesiana expresando en ella al menos las relaciones indicadas en el enunciado, de forma que todas las características abstractas del problema que hayas definido en el apartado anterior tengan algún tipo de influencia en la solución. Separa bien en el diagrama que variables describen características de problema y cuales describen soluciones. Lista de forma clara los diferentes valores que puede tomar cada variable. Da un ejemplo de tabla de probabilidad de algún nodo, inventándote las probabilidades, pero expresando como influyen los valores de los nodos padre en las probabilidades de los valores de los nodos hijo.

16. Una gran operadora de telecomunicaciones (telefonía, televisión a la carta e internet) quiere recomendar nuevos productos personalizados para sus clientes, ya sean de 1) telefonía móvil (compuesto por los servicios de telefonía, internet móvil y mensajería SMS/MMS), 2) internet+TV por cable (compuesto por internet por cable y televisión a la carta), o 3) ambos. Para ello quiere construir un sistema que recoge un montón de información del usuario (que ya tienen en sus sistemas informáticos):

- sobre su uso del móvil: se guarda toda la información relevante de las llamadas (destinatario, fecha, hora, duración) para poder extraer información como por ejemplo cuantas llamadas hace en horario de mañana, tarde y noche, cuantas llamadas hace en días laborables y cuantas en fin de semana, a cuantos operadores diferentes son sus llamadas, y cual es más o menos la duración de cada tipo de llamada. Del destinatario de la llamada se guarda su número de teléfono y la operadora de telecomunicaciones a la que pertenece. También se guarda la información sobre si usa internet en el móvil, y en caso positivo el número de GB de datos que en media suele

consumir al mes. También se guarda información sobre cuantos mensajes SMS/MMS envía al mes. Además se guarda información sobre la tarifa actual de telefonía móvil que tiene el cliente (nombre comercial de la tarifa, precio mensual, precio de consumo mínimo, precio de establecimiento de llamada, número de minutos incluidos, velocidad de conexión de datos, precio por SMS, precio por MMS, los meses de permanencia) y también de las tarifas anteriores que haya tenido con esta operadora en los últimos 10 años.

- sobre su uso de internet por cable: el número de meses que ha contratado el internet con esta operadora (solo cuentan los últimos 10 años), el número de GB de datos de bajada que en media suele consumir al mes, el número de GB de datos de subida que en media suele consumir al mes, si requiere una IP fija, si dispone de modem wifi, si ha contratado servicio de firewall, si ha contratado servicio de asistencia telefónica.
- sobre su uso de la televisión por cable: se guarda toda la información sobre los canales que puede ver (de cada uno se tiene nombre y número de canal), los programas que se ofrecen en cada canal, y los que finalmente el usuario ha visto (para cada programa se guarda un sumatorio de los minutos que ha visto). De todos los programas se tiene también el nombre, la duración media (en minutos), la clasificación por edades (apto, mayores de 3, mayores de 7, mayores de 12, mayores de 18), el año de la primera emisión y si se ha de pagar aparte para verlo (pay-per-view). Los programas se clasifican a su vez en informativos, documentales, culturales, entretenimiento, deportes, corazón, series de ficción, infantiles, películas y musicales. En el caso de programas del tipo serie de ficción se guarda también el número de temporada y el número de episodio. En el caso de las películas se guarda el nombre del director y el género (cine clásico, animación, ciencia ficción, comedia, drama, acción, etc.)

A partir de toda esta información se puede elaborar un perfil de las características del usuario basándonos en un conjunto de atributos: fidelidad del usuario (alta si lleva más de 3 años contratando nuestros productos, baja en caso contrario), uso de teléfono (bajo, moderado, alto, intensivo), destino de llamadas (al operador, a varios operadores), uso de mensajería (bajo, medio, alto), uso de internet móvil (nulo, bajo, medio, intensivo), uso de internet fijo (nulo, bajo, medio, intensivo), uso de televisión (nulo, ocasional, normal, elevado) y perfil de visionado (ninguno, variado, mayormente informativo, mayormente cultural, mayormente ficción, mayormente infantil, mayormente corazón).

Las diferentes tarifas de telefonía móvil a recomendar también se pueden describir mediante una serie de características que nos permitirán identificar mejor que conjunto de tarifas son las más adecuadas para el usuario. Una tarifa puede ser descrita por su cuota mensual (ninguna, baja, media, alta), la exigencia de un consumo mínimo (no, bajo, medio, alto), el coste de establecimiento de llamada (ninguno, bajo, medio, alto), minutos de llamada incluidos en la cuota mensual (medio, alto, muy alto), velocidad de internet móvil (no dispone, 3G, 4G, 5G), los GB de datos incluidos en la tarifa (ninguno, menos de 2 GB, entre 3 y 4 GB, más de 4 GB), coste de los mensajes (bajo, medio, alto), exigencia de permanencia (alta, media, baja).

Las diferentes tarifas de internet+TV por cable también se pueden describir por una serie de características: cuota mensual (baja, media, alta), velocidad de bajada de datos (básica, media, alta), la velocidad de subida de datos (básica, media, alta), oferta de canales (básica, ampliada, máxima), target TV (infantil, juvenil, toda la familia, informativo, cultural, deportes, corazón).

A partir del perfil del usuario podemos asociar los valores de las características de las tarifas de telefonía e internet+TV más adecuada. Por ejemplo, si la fidelidad del usuario es baja y hace un uso alto de telefonía móvil, puede convenirle una tarifa que tenga una exigencia de consumo mínimo alta y que tenga una cuota mensual media. Si la fidelidad del usuario es alta y su uso de teléfono es bajo podrá tener una tarifa telefónica con una cuota mensual baja y con una exigencia de permanencia alta. Si el uso del teléfono es intensivo necesitará un coste de establecimiento de llamada bajo, sin consumo mínimo y con los minutos de llamada incluidos en la tarifa altos o muy altos. Si el uso de internet fijo es intensivo y el uso de televisión es nulo entonces le recomendará una velocidad de subida de datos alta, una velocidad de bajada alta, una oferta de canales básica con un target TV para toda la familia. Si el uso de internet es medio o bajo y el uso de televisión es elevado entonces le recomendará una velocidad

de subida de datos baja, una velocidad de bajada alta, y una oferta de canales ampliada con un target TV adecuado a su perfil de visionado.

Una vez determinados los valores de las características de la tarifa más adecuada se puede identificar la tarifa concreta a partir de los valores concretos de consumo del usuario y los precios y características específicas de las tarifas ofrecidas de telefonía y de Internet+TV ofrecidas por el operador.

- (a) (2,5 puntos) Diseña la ontología del dominio descrito, incluyendo todos los conceptos que aparecen en la descripción e identificando los atributos más relevantes. Lista que conceptos forman parte de los datos de entrada del problema y que conceptos forman parte de la solución. (Nota: tened en cuenta que la ontología puede necesitar modificaciones para adaptarla al apartado siguiente, de forma que todos los predicados hagan relación a conceptos y atributos de la ontología).
- (b) (2,5 puntos) El problema descrito es un problema de análisis. Explica cómo lo resolverías usando clasificación heurística, identificando las fases necesarias y listando los predicados que se usan para ir de una fase a la siguiente. Da al menos 4 ejemplos de reglas para cada una de las fases de esta metodología, usando los conceptos de la ontología desarrollada en el apartado anterior. Inventa algún ejemplo concreto de tarifa si te hace falta.

17. Dado el preocupante aumento de los niveles de contaminación en las grandes ciudades europeas se quiere incentivar la compra de vehículos híbridos y eléctricos. Para ello nos piden diseñar un programa que, a partir del tipo de conductor, le recomiende uno o varios vehículos de emisiones reducidas del mercado. En esta primera fase el programa solo recomendará vehículos para particulares y autónomos, dejando los coches de empresa y las flotas para futuras versiones de la herramienta.

Al usuario (el conductor habitual del vehículo) se le pregunta por el nombre, la edad, el código postal en el que reside (para saber, por ejemplo, si vive en una área urbana especialmente contaminada) y el presupuesto total (cantidad máxima en euros que se quiere gastar en el vehículo). Y se le pide rellenar toda una serie de datos para conocer sus hábitos de desplazamiento en un año. De cada tipo de desplazamiento que suele realizar se le pregunta si es por ocio o por trabajo, los otros ocupantes del vehículo (de cada uno pregunta nombre y edad), la cantidad de equipaje que suele llevar (en litros), cuantos kilómetros hace al día en ese tipo de desplazamiento, el número de días al mes que se realiza, y el lugar de origen y destino de ese desplazamiento. De cada lugar de desplazamiento se quiere saber su tipología (urbano, interurbano asfaltado, tierra, abrupto) y si dispone de una estación de carga para vehículos eléctricos.

Sobre los vehículos que queremos recomendar (coches y motos) tenemos información almacenada sobre la marca, el modelo, el número máximo de plazas (entre 2 y 7), el tipo de motorización (eléctrico, híbrido o híbrido enchufable), la potencia fiscal (en caballos), el tiempo de carga total de las baterías (en minutos, 0 para los híbridos), el consumo homologado (en litros/100 km), la autonomía homologada (en km), las emisiones homologadas de CO<sub>2</sub> (en gr/km) y el precio base (en euros). En el caso de las motos tenemos también el tipo de moto (scooter, triciclo, motocicleta) y su velocidad máxima (en km/h). En el caso de los coches solo realizará recomendaciones en 6 categorías: micro-coche, turismo, monovolumen, todocamino-compacto, todocamino o todoterreno. En el caso de los micro-coches, turismos, monovolumenes y todocaminos-compactos tendremos también la longitud del coche (en metros, con decimales), al ser un dato relevante en la compra. En los micro-coches tendremos también el ancho del vehículo (en metros, con decimales), y en turismos y todocaminos-compactos sabemos si se abaten los asientos traseros o no. Otra información relevante disponible para todocaminos-compactos, todocaminos y todoterrenos es el tipo de tracción (4x2, 4x4 o 4x4 desconectable), la altura libre de la carrocería al suelo (en cm) y si dispone de asistente para pendientes. Los compradores de todoterrenos son un público muy exigente, y por ello en este caso tenemos además información sobre si la caja de cambios dispone de reductora, la profundidad de vadeo (en cm), el ángulo de ataque, ángulo ventral y ángulo de salida (los tres en grados).

Los expertos en movilidad nos han indicado una serie de criterios importantes a la hora de escoger un vehículo:

- El **tipo de presupuesto** disponible, que se clasifica en bajo (por debajo de los 10000 euros), medio (entre 10000 y 25000 euros) y elevado (por encima de 25000 euros);

- El **número de pasajeros** medio de los desplazamientos, que se clasifica en tres grupos: de 1 a 2 pasajeros, de 3 a 5 pasajeros y de 6 a 7 pasajeros;
- La **necesidad de equipaje** media, que se clasifica en cuatro grupos: muy baja (por debajo de los 10 litros), baja (entre 10 y 100 litros en la mayoría de desplazamientos), media (entre 100 y 250 litros de equipaje medio, o si se lleva un menor de 10 años abordo), y elevada (más de 250 litros de equipaje, o si se lleva dos o más menores de 10 años abordo).
- El **trayecto medio**, que se clasifica en corto (una mayoría de desplazamientos por debajo de 50 km), medio (desplazamientos con una media entre 50 y 200 km) y largo (con una media de más de 200 km por desplazamiento)
- Los **kilómetros al año**, en que se distinguen tres grupos: por debajo de 2000 km/año, entre 2000 y 10000 km/año, y por encima de los 10000 km/año.
- El tipo de **terreno** en el que se suele conducir principalmente, distinguiendo entre cuatro grupos: conductores mayormente urbanos, o mayormente interurbanos, o mixtos (conductores que ocasionalmente salen de vías asfaltadas) o bien off-road (conductores que conducen más del 25% de sus kilómetros en tierra o que conducen en algún momento por terreno abrupto);
- La **disponibilidad de cargadores**, que estima la facilidad para encontrar una estación de carga en los lugares de origen y destino de los desplazamientos. Se considera una disponibilidad alta si al menos el 90% de los lugares de origen y destino de los desplazamientos de trabajo y el 75% de los lugares en desplazamientos de ocio disponen de estaciones de carga, disponibilidad media si los porcentajes son del 75% y 50% respectivamente, y disponibilidad baja en caso contrario.

Los expertos nos han dado también ejemplos de otros criterios para seleccionar los vehículos a recomendar:

- Podemos determinar que la prioridad principal del usuario es un consumo reducido si sus trayectos suelen ser cortos o medios y si suele hacer más de 10000 kilómetros al año. Pero su prioridad será la autonomía si sus trayectos son medios o largos y si suele hacer menos de 10000 km/año.
- Para escoger el tipo de motorización más adecuada para el usuario se combinan varios factores. Los vehículos eléctricos suelen ser más adecuados cuando la principal prioridad es el consumo, si el trayecto medio es corto y si hay una disponibilidad de cargadores media o elevada. Los vehículos híbridos suelen ser más adecuados cuando la principal prioridad es la autonomía, si el trayecto medio es largo y si la disponibilidad de cargadores es baja. Los vehículos híbridos enchufables combinan lo mejor de los vehículos eléctricos y los híbridos, son aptos para usuarios cuya prioridad sea la autonomía o el consumo y si la disponibilidad de cargadores es media, pero no son aptos para presupuestos bajos.
- Para escoger el tipo de vehículo adecuado para el usuario se tienen en cuenta otros factores. Las motos son recomendables para tipos de presupuesto bajos, cuando el terreno no es off-road, se llevan de 1 a 2 pasajeros y la necesidad de equipaje es muy baja. Los micro-coches son más adecuados para presupuestos bajos o medios, cuando el terreno es mayormente urbano, se llevan 5 o menos pasajeros y la necesidad de equipaje es baja. Los turismos son más adecuados para presupuestos medios, cuando el terreno no es mixto ni off-road, se llevan 5 o menos pasajeros y la necesidad de equipaje es media. Los monovolúmenes son más adecuados para presupuestos altos, cuando el terreno no es mixto ni off-road, se llevan entre 3 y 7 pasajeros y la necesidad de equipaje es elevada. Los todocaminos compactos son más adecuados para presupuestos medios, uso en terrenos mixtos, cuando se llevan 5 o menos pasajeros y la necesidad de equipaje es media. Los todocaminos son más adecuados para presupuestos altos, uso en terrenos mixtos, cuando se llevan entre 3 y 7 pasajeros y la necesidad de equipaje es elevada. Finalmente los todoterrenos son aptos para terrenos off-road.

El sistema dará dos tipos de salidas:

- Para aquellos usuarios con la mayoría de trayectos por debajo de los 200 km y que hacen menos de 2000 km al año el sistema intentará concienciarlos de que no necesitan tener un coche en propiedad, y que lo mejor es que combinen el uso del transporte público con el uso esporádico de un vehículo alquilado en una empresa de renting. Eso sí, se le aconsejará el tipo de vehículo ecológico (moto, micro-coche, turismo, monovolumen, todocamino-compacto, todocamino o todoterreno) y el tipo

de motorización (eléctrico, híbrido o híbrido enchufable) más adecuado a sus necesidades.

- Para el resto de usuarios recomendará la compra de un vehículo ecológico y mostrará la lista de modelos disponibles en el mercado, incluyendo el precio y los impuestos. La aplicación dispone de una función `mostrar_lista_vehículos(tipo_vehículo, tipo_motorización, presupuesto_total, código_postal)` que dado el tipo de vehículo escogido, el tipo de motorización, el presupuesto total (en euros) y el código postal del usuario muestra por pantalla la lista de vehículos disponibles en el mercado que corresponden al tipo de motorización y vehículo recomendados y cuyo precio base más el importe del impuesto de matriculación que ha de pagar en su zona de residencia están por debajo del presupuesto total.

- (a) Diseña la ontología del dominio descrito, incluyendo todos los conceptos que aparecen en la descripción e identificando los atributos más relevantes. Lista que conceptos forman parte de los datos de entrada del problema y que conceptos forman parte de la solución. (Nota: tened en cuenta que la ontología puede necesitar modificaciones para adaptarla al apartado siguiente).
- (b) El problema descrito es un problema de análisis. Explica cómo lo resolverías usando clasificación heurística, usando los conceptos de la ontología desarrollada en el apartado anterior. Da al menos 4 ejemplos de reglas para cada una de las fases de esta metodología.
- (c) Las características que se usan durante el proceso de recomendación de vehículos ecológicos no son independientes entre sí. El número de ocupantes en el vehículo suele influir en la necesidad de equipaje (a mayor número de pasajeros, más equipaje) y la disponibilidad de cargadores suele depender del tipo de terreno por el que circula normalmente (mayor disponibilidad en terrenos urbanos, disponibilidad mínima en terrenos off-road). Define el problema de asociación heurística como una red bayesiana expresando en ella al menos las relaciones indicadas en el enunciado, de forma que todas las características abstractas del problema que hayas definido en el apartado anterior tengan algún tipo de influencia en la solución. Separa bien en el diagrama que variables describen características de problema y cuales describen soluciones. Lista de forma clara los diferentes valores que puede tomar cada variable. Da un ejemplo de tabla de probabilidad de algún nodo, inventándote las probabilidades, pero expresando como influyen los valores de los nodos padre en las probabilidades de los valores de los nodos hijo.

18. La manera habitual de poner precio a bienes cuyo valor no es objetivo es la tasación. Este tipo de valoración es la que se utiliza por ejemplo para las viviendas. Existen muchos sistemas de tasación, pero habitualmente se basan en un conjunto de características cualitativas que se obtienen de la valoración de las características de la vivienda y su ubicación.

En el caso de las viviendas se recopila información sobre el piso en el que está (bajos, entresuelo, primero, segundo...) la superficie total y la superficie habitable (ambas en  $m^2$ ), el ancho y el largo de la planta de la vivienda (en  $m$ ), si tiene certificado energético ( $A, B, C, D, E, F, G$  ó *ninguno*), si tiene calefacción (central, a gas, eléctrica), las frigorías del sistema de aire acondicionado (0 si no tiene), si tiene instalación de fibra óptica, la orientación de la mayoría de ventanas (norte, noreste, este, sureste...) el tiempo que lleva deshabitada (en semanas) y el total de impuestos y tasas que se han de pagar al año (en euros). Para cada una de las estancias de la vivienda (baño, cocina, habitación...) se guarda la superficie (en  $m^2$ ), el material del suelo (madera, gres, acrílico), el material de las paredes (gres, pintura, papel pintado), la calidad de los acabados (superior, alta, media, baja), si hay aparato calefactor en la estancia, si hay aparato de aire acondicionado, la anchura de la puerta más estrecha en la estancia (en cm, para saber si puede pasar una silla de ruedas) y si necesita reforma. Aparte de esa información, en el caso del baño se guarda el tipo de ducha (bañera o plato de ducha), si tiene armarios y si hay bidet. En el caso de la cocina se guarda el tipo (cerrada, semicerrada en un segundo ambiente o abierta estilo americano), la superficie útil de trabajo (en  $cm^2$ ), el material de las superficies de trabajo (marmol, madera, composite) y si tiene una isla central con los fogones. Del lavadero se guarda si es una estancia independiente o forma parte de la cocina. Del comedor se guarda si tiene una chimenea y el número de columnas que tiene en su interior (0 si no hay). De cada habitación se guarda también si tiene armarios empotrados y si tiene baño propio. De los pasillos y distribuidores se guarda el largo, el ancho máximo y el ancho mínimo (los tres datos en metros). En los balcones se

guarda el largo y el ancho (en metros) y la proporción de terraza que está cubierta por techo (en %). De trasteros y altillos se guarda el volumen total (en  $m^3$ ).

En el caso de los edificios se guarda el año de construcción, la planta del edificio (en  $m^2$ ), el tipo de construcción (vivienda unifamiliar aislada, vivienda adosada, edificio aislado o edificio adosado), la calle y el número, el código postal, el número de vecinos, el gasto comunitario mensual (en euros), si tiene portero o no, el ancho mínimo de las rampas de acceso (0 si no tiene), el tamaño del ascensor (largo y ancho dentro de la cabina), la altura total del edificio (en metros), y el año de la última reforma de fachada. De la calle se guarda también el tipo de vía (avenida, calle, paseo, calle peatonal, Zona 30, ...), a qué barrio pertenece, la anchura de aceras y la distancia mínima entre las fachadas de los edificios de lado a lado de la calle (ambos en metros). El edificio puede tener una serie de elementos comunitarios (placas solares, parking, trasteros, terraza comunitaria, piscina, gimnasio comunitario). De la piscina se guarda el largo y el ancho (en metros) y si tiene vestidores o taquillas, de las placas solares se guarda la potencia media de generación mensual (en vatios), del parking se guarda el número de plazas, de los trasteros su número y el tamaño medio ( $m^3$ ) y de la terraza comunitaria su superficie (en  $m^2$ ) y si los vecinos pueden colgar ropa allí.

De los barrios se guarda el nombre, el tipo de zona (centro, periferia, turística, industrial...), el precio medio del metro cuadrado de vivienda, el tiempo medio que se tarda en llegar al centro en transporte público (en minutos) y el tiempo medio que se tarda en salir desde allí a las vías rápidas fuera de la ciudad (en minutos). Cada barrio tiene asociado una serie de equipamientos/servicios (locales de ocio, comercios, centros de barrio, parques y jardines, centros de culto, farmacias, oficina de correos, comisaría de policía, hospitales, centros de atención primaria...). De todos ellos se guarda el nombre, la calle, el número y el horario de apertura (un string del estilo "lunes a viernes de 9 a 13 y de 15 a 18 horas" o "24/7" en el caso de que esté abierto las 24 horas del día los 7 días de la semana). De los locales de ocio queremos guardar el tipo (bares, restaurantes, discotecas, pubs musicales, salas de baile, salones de fiestas, ...) y el aforo (en número máximo de personas). De los centros de culto queremos saber la religión (un string) y el aforo máximo. De los centros educativos se guarda el tipo de estudios (pre-escolar, primaria, secundaria, universitarios, academia, ...). También se guarda un listado de las paradas de bus, metro, tranvía, tren y taxi dentro de cada barrio. Para todas ellas se tiene las coordenadas GPS (un string) y una descripción textual (string) de las líneas que pasan.

Por lo que nos dicen los expertos se suelen usar cuatro características para tasar una vivienda:

- El **entorno**, con una valoración de 1 a 5, que tendría en cuenta el número, tipo y proximidad de los equipamientos/servicios de la zona del inmueble (jardines, bibliotecas, centro de atención primaria, hospitales, colegios, áreas comerciales, ...), las comunicaciones de la zona (transporte público, vías rápidas, ...) y la zona en la que está ubicada (centro, periferia, zona turística, zona industrial...).
- La **calidad del edificio**, con una valoración de 1 a 5, que tendría en cuenta su antigüedad, el tipo de construcción, la superficie de su planta, su altura, la existencia de elementos de accesibilidad (rampas, barandillas, ascensor, anchura de la entrada, de los rellanos, ...), el número de vecinos, la anchura de la calle, la existencia de aparcamiento en el edificio, los gastos de la comunidad, ...
- La **calidad de la vivienda**, con una valoración de 1 a 5, que tendría en cuenta su superficie total, su superficie habitable, la relación entre la anchura y la longitud de la vivienda, si necesita reformas, la calidad de los acabados, el tiempo que lleva deshabitada, ...
- La **habitabilidad/funcionalidades** con una valoración de 1 a 5, que tendría en cuenta el número de habitaciones, el número de baños, la existencia de balcones y terrazas, elementos extra (suelo de madera, calefacción, trasteros, armarios empotrados, ...), ...

A partir de estas características queremos determinar si la vivienda se ha de tasar a precio de mercado, por debajo del precio de mercado o por encima del precio de mercado. El precio de mercado se obtiene a partir del precio medio del  $m^2$  de la zona. Para determinar la variación respecto al precio de mercado, usaremos las características particulares de la vivienda para cuantificar la variación del precio. Por ejemplo, en el caso de que se clasifique la vivienda por debajo del precio de mercado, si la vivienda se ha de reformar el precio de mercado se rebajaría en un 20%, o si hay pocos servicios en el entorno se rebajaría un 5%. En el caso de que se clasifique la vivienda por encima del precio de mercado, si



la vivienda tiene muchos elementos extra de habitabilidad como suelo de madera, o calefacción se le añadirán 3000 euros por cada elemento extra, o si el edificio tiene elementos de accesibilidad, como por ejemplo rampas, se añadirán 2000 euros al precio.

- (a) Diseña la ontología del dominio descrito, incluyendo todos los conceptos que aparecen en la descripción e identificando los atributos más relevantes. Lista que conceptos forman parte de los datos de entrada del problema y que conceptos forman parte de la solución. (Nota: tened en cuenta que la ontología puede necesitar modificaciones para adaptarla al apartado siguiente).
- (b) El problema descrito es un problema de análisis. Explica cómo lo resolverías usando clasificación heurística, usando los conceptos de la ontología desarrollada en el apartado anterior. Da al menos 3 ejemplos de reglas para cada una de las fases de esta metodología.
- (c) Las características que se usan durante la tasación de la vivienda no son independientes entre sí. Tanto el entorno como la calidad del edificio influyen en la calidad de la vivienda, (a mayores servicios y mayor calidad del edificio, mayor calidad de la vivienda) y ésta a su vez influye en sus funcionalidades (a mayor calidad de la vivienda, aumenta la probabilidad de una mayor habitabilidad y funcionalidad). Define el problema de asociación heurística como una red bayesiana expresando en ella al menos las relaciones indicadas en el enunciado, de forma que todas las características abstractas del problema que hayas definido en el apartado anterior tengan algún tipo de influencia en la solución. Separa bien en el diagrama que variables describen características de problema y cuales describen soluciones. Lista de forma clara los diferentes valores que puede tomar cada variable. Da un ejemplo de tabla de probabilidad de algún nodo, inventándote las probabilidades, pero expresando como influyen los valores de los nodos padre en las probabilidades de los valores de los nodos hijo.

19. Un despacho de arquitectura especializado en la construcción de viviendas unifamiliares en parcelas individuales (no hacen adosados), desea introducir las tecnologías de la información en el diseño de las mismas construyendo un sistema basado en el conocimiento. Aunque el diseño de viviendas unifamiliares es una tarea muy creativa existen ciertos aspectos estructurales (como la selección de los volúmenes que conforman la vivienda) que se pueden sistematizar. Hay una serie de características que se han de tener en cuenta:

- La orografía de la parcela, su tamaño, su orientación y la climatología de la zona.
- Las normativas urbanísticas municipales (altura máxima, distancia mínima desde la fachada principal a la calle, aspecto exterior de la fachada, etc.).
- Número máximo de plantas y distribución mínima requerida por el cliente (número mínimo de habitaciones dobles, tamaño mínimo del salón-comedor, garaje, número mínimo de cuartos de baño/aseo, ...).
- Los materiales de construcción disponibles en la zona, de diferentes tipos, calidades y precios.
- Las necesidades (rampa para inválidos, sala insonorizada, ...) y gustos del cliente.
- Número y perfil de personas que han de ocupar la casa.
- El presupuesto del cliente.
- El coste de mantenimiento.

Dada su larga experiencia en el área, la empresa ha creado un catálogo de tipos básicos de viviendas unifamiliares: UNI (una planta), DUO (dos plantas), TRI (tres plantas) que pueden tener más o menos habitaciones en función del tamaño de la parcela y las necesidades del cliente. Las viviendas UNI son más baratas de construir por  $m^2$  (necesitan menos estructura de soporte y los materiales pueden ser más sencillos) y son especialmente recomendables para familias con personas de movilidad reducida (ancianos, personas en silla de ruedas, etc.) pero necesitan más terreno edificable y no son adecuadas en lugares muy húmedos. Las viviendas DUO son algo más caras de construir por  $m^2$  al tener que reforzar la estructura por la planta extra y necesitan materiales de mejor calidad. Independientemente de los gustos del cliente, las DUO son especialmente recomendables en terrenos algo inclinados para

tener luz natural en la mayoría de habitaciones. Las TRI son las más caras de construir por  $m^2$  pero son recomendables en terrenos con mucho desnivel o con poca superficie edificable. Las TRI no son recomendables para familias en las que haya gente con tendencia a problemas musculares o articulatorios o en zonas con vientos fuertes ocasionales.

Se desea construir un sistema basado en el conocimiento que, dadas las preferencias y restricciones del cliente, pero teniendo en cuenta las características de la parcela y las normativas municipales, obtenga el mejor diseño de la vivienda (tipo básico escogido, número de habitaciones de cada tipo y, para cada una de las habitaciones o estancias, en que planta está y que tamaño tiene).

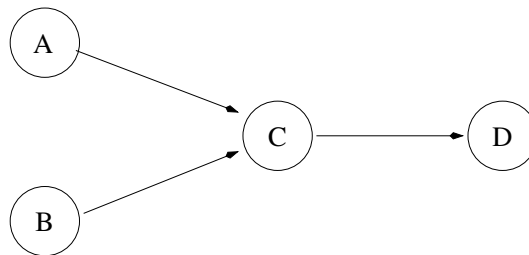
- (a) Diseña la ontología del dominio descrito, incluyendo todos los conceptos que aparecen en la descripción e identificando los atributos más relevantes. Lista que conceptos forman parte de los datos de entrada del problema y que conceptos forman parte de la solución. (Nota: tened en cuenta que la ontología puede necesitar modificaciones para adaptarla al apartado siguiente).
- (b) El problema descrito es un problema de análisis. Explica cómo lo resolverías usando clasificación heurística, usando los conceptos de la ontología desarrollada en el apartado anterior. Da al menos 3 ejemplos de reglas para cada una de las fases de esta metodología.
- (c) Las características que se utilizan para caracterizar el pedido y los tipos básicos de viviendas escogidos no son independientes entre si. Por ejemplo: la movilidad de los habitantes de la casa, el presupuesto, la orografía y el espacio disponible afectan al número de plantas, el presupuesto también influye en la calidad de los materiales, que a su vez tiene un efecto sobre el número de plantas, etc.

Define el problema de asociación heurística como una red bayesiana expresando en ella al menos las

relaciones indicadas en el enunciado, en este apartado y alguna más, de forma que todas las características abstractas del problema que hayas definido en el apartado anterior tengan algún tipo de influencia en la solución. Separa bien en el diagrama que variables describen características de problema y cuales describen soluciones. Da un ejemplo de tabla de probabilidad de algún nodo, inventándote las probabilidades, pero expresando como influyen los valores de los nodos padre en las probabilidades de los valores de los nodos hijo.

## 9. Xarxes Bayesianes

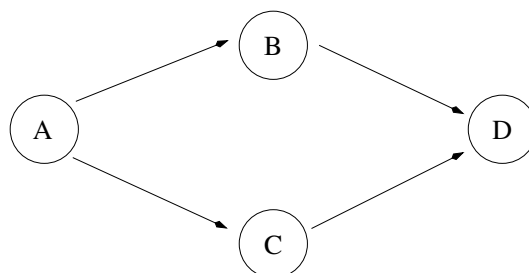
1. Dada la siguiente red bayesiana y las probabilidades asociadas a cada nodo



A	P(A)	B	P(B)	A	B	P(C A,B)		C	P(D C)	
						C	F		C	F
C	0.25	C	0.4	C	C	0.75	0.25	C	0.3	0.7
F	0.75	F	0.6	C	F	0.5	0.5	F	0.4	0.6
				F	C	0.75	0.25			
				F	F	0.1	0.9			

- Da la expresión de la distribución de probabilidad conjunta que representa la red
- Calcula las siguientes probabilidades mediante el algoritmo de eliminación de variables
  - $P(C|a, \neg b)$
  - $P(D|b, \neg c)$
  - $P(B|a)$
  - $P(D|a)$

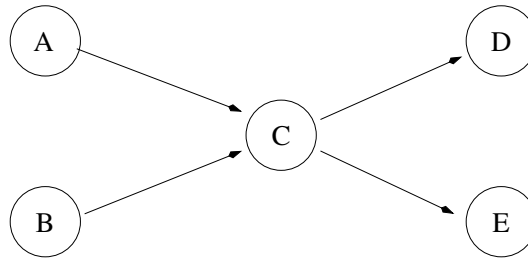
2. Dada la siguiente red bayesiana y las probabilidades asociadas a cada nodo



A	P(A)	A	P(B A)		A	P(C A)		B	C	P(D B,C)	
			C	F		C	F			C	F
C	0.3	C	0.1	0.9	C	0.1	0.9	C	C	0.7	0.3
F	0.7	F	0.4	0.6	F	0.6	0.4	C	F	0.2	0.8
								F	C	0.1	0.9
								F	F	0.4	0.6

- (a) Da la expresión de la distribución de probabilidad conjunta que representa la red
- (b) Calcula las siguientes probabilidades mediante el algoritmo de eliminación de variables
- $P(B|a, \neg c)$
  - $P(D|b)$
  - $P(C|a, \neg d)$

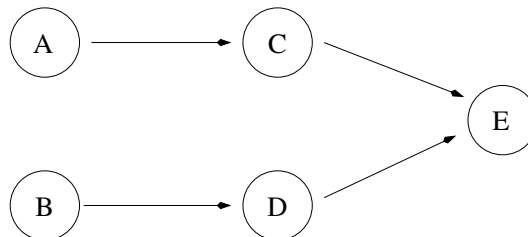
3. Dada la siguiente red bayesiana y las probabilidades asociadas a cada nodo



A	P(A)	B	P(B)	A	B	P(C A,B)		C	P(D C)		C	P(E C)	
						C	F		C	F		C	F
C	0.3	C	0.2	C	C	0.3	0.7	C	0.1	0.9	C	0.4	0.6
F	0.7	F	0.8	C	F	0.1	0.9	F	0.7	0.3	F	0.7	0.3
				F	C	0.7	0.3						
				F	F	0.4	0.6						

- (a) Da la expresión de la distribución de probabilidad conjunta que representa la red
- (b) Calcula las siguientes probabilidades mediante el algoritmo de eliminación de variables
- $P(D|e, \neg b)$
  - $P(E|c)$
  - $P(E|\neg a)$

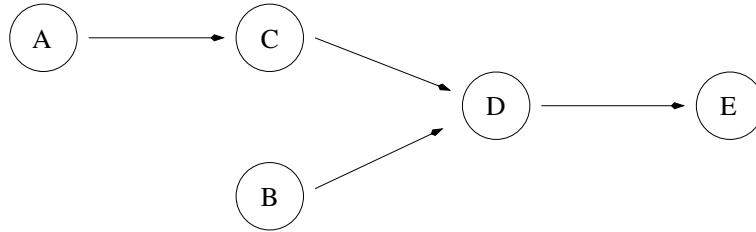
4. Dada la siguiente red bayesiana y las probabilidades asociadas a cada nodo



A	P(A)	B	P(B)	A	P(C A)		B	P(D A)		C	D	P(E C,D)	
					C	F		C	F			C	F
C	0.2	C	0.6	C	0.3	0.7	C	0.6	0.4	C	C	0.1	0.9
F	0.8	F	0.4	F	0.2	0.8	F	0.9	0.1	C	F	0.7	0.3
										F	C	0.2	0.8
										F	F	0.4	0.6

- (a) Da la expresión de la distribución de probabilidad conjunta que representa la red
- (b) Calcula las siguientes probabilidades mediante el algoritmo de eliminación de variables
- $P(E|a)$
  - $P(E|\neg d)$
  - $P(D|c, e)$

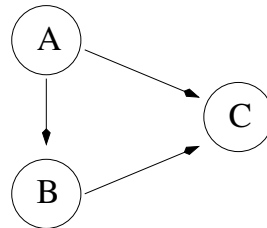
5. Dada la siguiente red bayesiana y las probabilidades asociadas a cada nodo



A	P(A)	B	P(B)	A	P(C A)		B	C	P(D B,C)		D	P(E D)	
					C	F			C	F		C	F
C	0.1	C	0.3	C	0.2	0.8	C	C	0.1	0.9	C	0.2	0.8
F	0.9	F	0.7	F	0.6	0.4	C	F	0.7	0.3	F	0.7	0.3
							F	C	0.8	0.2			
							F	F	0.4	0.6			

- (a) Da la expresión de la distribución de probabilidad conjunta que representa la red
- (b) Calcula las siguientes probabilidades mediante el algoritmo de eliminación de variables
- $P(E|a, \neg c)$
  - $P(E|c)$
  - $P(D|e, \neg a)$

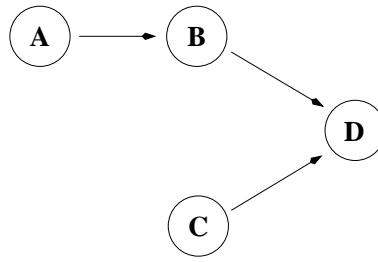
6. Dada la siguiente red bayesiana y las probabilidades asociadas a cada nodo



A	P(A)	A	P(B A)		A	B	P(C A,B)	
			C	F			C	F
C	0.3	C	0.1	0.9	C	C	0.1	0.9
F	0.7	F	0.6	0.4	C	F	0.7	0.3
					F	C	0.4	0.6
					F	F	0.8	0.2

- (a) Da la expresión de la distribución de probabilidad conjunta que representa la red
- (b) Calcula la probabilidad de  $P(C|a)$  mediante el algoritmo de eliminación de variables

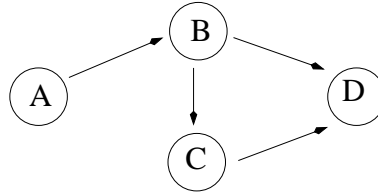
7. Dada la siguiente red bayesiana y las probabilidades asociadas a cada nodo



$A$	$P(A)$	$C$	$P(C)$	$A$	$P(B A)$		$B$	$C$	$P(D B,C)$	
					$C$	$F$			$C$	$F$
$C$	0.3	$C$	0.6	$C$	0.2	0.8	$C$	$C$	0.3	0.7
$F$	0.7	$F$	0.4	$F$	0.4	0.6	$C$	$F$	0.9	0.1
							$F$	$C$	0.5	0.5
							$F$	$F$	0.6	0.4

- (a) Da la expresión de la distribución de probabilidad conjunta que representa la red  
 (b) Calcula la probabilidad de  $P(D|\neg a)$  mediante el algoritmo de eliminación de variables

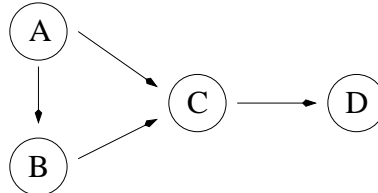
8. Dada la siguiente red bayesiana y las probabilidades asociadas a cada nodo



$A$	$P(A)$	$A$	$P(B A)$		$B$	$P(C A,B)$		$B$	$C$	$P(D B,C)$	
			$C$	$F$		$C$	$F$			$C$	$F$
$C$	0.7	$C$	0.1	0.9	$C$	0.5	0.5	$C$	$C$	0.1	0.9
$F$	0.3	$F$	0.4	0.6	$F$	0.2	0.8	$C$	$F$	0.4	0.6
								$F$	$C$	0.3	0.7
								$F$	$F$	0.8	0.2

- (a) Calcula la probabilidad de  $P(D|\neg a)$  mediante el algoritmo de eliminación de variables

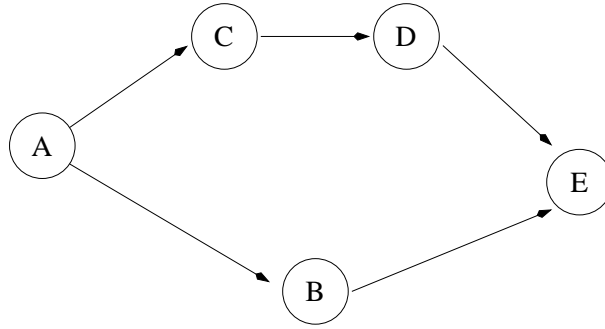
9. Dada la siguiente red bayesiana y las probabilidades asociadas a cada nodo



$A$	$P(A)$	$A$	$P(B A)$		$A$	$B$	$P(C A,B)$		$C$	$P(D C)$	
			$C$	$F$			$C$	$F$		$C$	$F$
$C$	0.6	$C$	0.2	0.8	$C$	$C$	0.1	0.9	$C$	0.1	0.9
$F$	0.4	$F$	0.7	0.3	$C$	$F$	0.5	0.5	$F$	0.7	0.3
					$F$	$C$	0.3	0.7			
					$F$	$F$	0.4	0.6			

Calcula la probabilidad de  $P(D|b)$  mediante el algoritmo de eliminación de variables

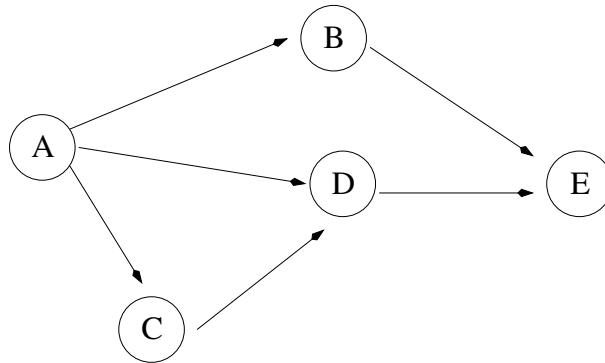
10. Dada la siguiente red bayesiana y las probabilidades asociadas a cada nodo



A	P(A)	A	P(B A)		A	P(C A)		C	P(D C)		B	D	P(E B,D)	
			C	F		C	F		C	F			C	F
C	0.3	C	0.5	0.5	C	0.4	0.6	C	0.1	0.9	C	C	0.4	0.6
F	0.4	F	0.1	0.9	F	0.2	0.8	F	0.6	0.4	C	F	0.1	0.9
											F	C	0.2	0.8
											F	F	0.5	0.5

Calcula la probabilidad de  $P(E|\neg a, b)$  mediante el algoritmo de eliminación de variables

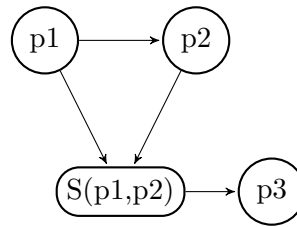
11. Dada la siguiente red bayesiana y las probabilidades asociadas a cada nodo



A	P(A)	A	P(B A)		A	P(C A)		A	C	P(D A,C)		B	D	P(E B,D)	
			C	F		C	F			C	F			C	F
C	0.7	C	0.6	0.4	C	0.6	0.4	C	C	0.4	0.6	C	C	0.1	0.9
F	0.3	F	0.8	0.2	F	0.3	0.7	C	F	0.8	0.2	C	F	0.6	0.4
								F	C	0.3	0.7	F	C	0.7	0.3
								F	F	0.5	0.5	F	F	0.2	0.8

Calcula la probabilidad de  $P(E|\neg b, c)$  mediante el algoritmo de eliminación de variables

12. Las redes bayesianas se pueden usar en cualquier problema de decisión, por ejemplo, en el área de lenguaje natural, podemos usar este formalismo para determinar la probabilidad de los diferentes roles sintácticos de las palabras. Supongamos que tenemos tres palabras consecutivas  $p_1$ ,  $p_2$  y  $p_3$  y también tenemos el grupo sintáctico al que pertenecen las dos primeras palabras  $S(p_1, p_2)$ , la siguiente red bayesiana describiría la probabilidad de la categoría sintáctica de la palabra  $p_3$  en función del resto de variables:



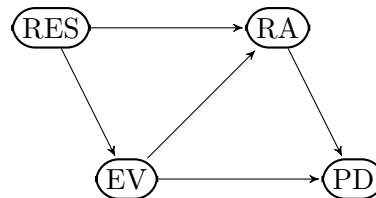
$p1$	$P(p1)$	$p1$	$P(p2 p1)$		$p1$	$p2$	$P(S(p1,p2) p1,p2)$		$S(p1,p2)$	$P(p3 S(p1,p2))$	
			nom	verb			SN	nSN		nom	adj
adj	0.8	adj	0.7	0.3	adj	nom	0.9	0.1	SN nSN	0.7	0.3
adv	0.2	adv	0.1	0.9	adj	ver	0.5	0.5		0.2	0.8
					adv	nom	0.8	0.2			
					adv	ver	0.3	0.7			

Define la distribución de probabilidad conjunta de  $P(p1,p2,p3,S(p1,p2))$  que define la red y usa el algoritmo de eliminación de variables para calcular la probabilidad de  $P(p3|p1 = adj)$ .

13. Una aplicación bastante difundida de las redes bayesianas es la implementación de asistentes de usuario, como por ejemplo en un sistema operativo. Estos asistentes pueden recolectar información del funcionamiento del sistema y presentar recomendaciones y avisos al usuario para que atienda a posibles problemas en su sistema.

Por ejemplo, podemos tener un asistente que cronometra la respuesta de la aplicación en primer plano (RA), el rendimiento de entrada/salida de los discos duros (RES) y el tamaño de la cola de eventos del interfaz del sistema operativo (EV) y usar estas informaciones para lanzar un mensaje sobre un posible problema en la configuración de drivers (PD).

Esta red bayesiana representaría las relaciones entre estos conceptos.



Y estas tablas de probabilidades representarían las influencias de unos nodos sobre otros.

$RES$	$P(RES)$	$RES$	$P(EV)$		$RES$	$EV$	$P(RA)$		$EV$	$RA$	$P(PD)$	
			vacía	llena			$< 500ms$	$> 500ms$			Sí	No
$> 12Mbps$	0.8	$> 12Mbps$	0.7	0.3	$> 12Mbps$	vacía	0.9	0.1	vacía	$< 500ms$	0.1	0.9
$< 12Mbps$	0.2	$< 12Mbps$	0.1	0.9	$> 12Mbps$	llena	0.5	0.5	vacía	$> 500ms$	0.3	0.7
					$< 12Mbps$	vacía	0.6	0.4	llena	$< 500ms$	0.4	0.6
					$< 12Mbps$	llena	0.1	0.9	llena	$> 500ms$	0.9	0.1

Define la distribución de probabilidad conjunta de  $P(RES, EV, RA, PD)$  que representa la red y usa el algoritmo de eliminación de variables para calcular la probabilidad de  $P(RA|EV = llena, PD = Sí)$ .





# Planificació



## 10. Planificació

1. Traducir el siguiente modelo del problema del mundo de los bloques de una codificación STRIPS a una codificación en PDDL.

### Operadores:

OP: pick-up(x):

PRE: clear(x), ontable(x), handempty().

DEL: clear(x), ontable(x), handempty().

ADD: holding(x).

OP: put-down(x):

PRE: holding(x).

DEL: holding(x).

ADD: clear(x), handempty(), ontable(x).

OP: stack(x,y):

PRE: holding(x), clear(y).

DEL: holding(x), clear(y).

ADD: clear(x), handempty(), on(x,y).

OP: unstack(x,y):

PRE: on(x,y), clear(x), handempty().

DEL: on(x,y), clear(x), handempty().

ADD: holding(x), clear(y).

### Problema:

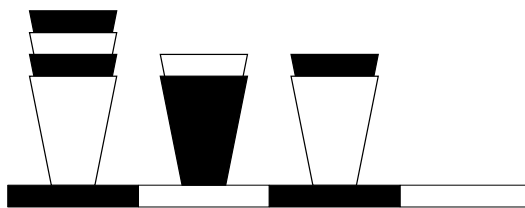
clear(A), clear(B), clear(E), clear(C), clear(D), ontable(F), ontable(B),  
ontable(E), ontable(C), ontable(D), on(A,F), handempty().

$\text{on}(E,F) , \text{on}(F,C) , \text{on}(C,B) , \text{on}(B,A) , \text{on}(A,D) .$

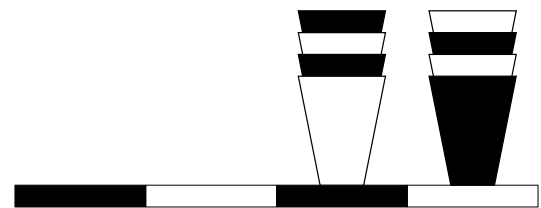
2. Tenemos un vector de tres posiciones. Queremos crear un modelo de dominio que permita darle a un planificador una configuración inicial de valores del vector (ej:  $|a|b|c|$ ) y una configuración final de esos valores en el vector (ej:  $|c|b|a|$ ) y que un planificador estilo Fast Forward encuentre los movimientos necesarios a realizar.
3. Un equipo de investigadores en robótica necesita dotar a su robot de la capacidad de planificar acciones para poder probar su habilidad de reconocimiento de objetos y colores. Para empezar las pruebas quieren hacer un planificador simple que le diga al robot cómo apilar y desapilar vasos en una mesa.

Los vasos pueden ser de dos colores: blanco y negro. En la mesa se han pintado cuatro casillas, dos blancas y dos negras. Las restricciones que se imponen son las siguientes:

- El robot solo puede usar uno de sus brazos, y coger un vaso a la vez.
  - Un vaso negro solo se puede poner encima de un vaso blanco o una casilla blanca.
  - Un vaso blanco solo se puede poner encima de un vaso negro o una casilla negra.
- (a) Describir el dominio (incluyendo predicados, acciones, etc...) usando PDDL. Dad una explicación razonada de los elementos que habéis escogido. Tened en cuenta que el modelo del dominio ha de funcionar para tamaños de problema grandes (como el descrito en el apartado c)
  - (b) Describir el problema siguiente usando PDDL. Dad una breve explicación de cómo modeláis el problema.



Estado inicial



Estado final

- (c) Ahora supongamos que hemos pintado en la mesa un tablero de  $24 \times 24$  casillas, la mitad blancas y la mitad negras, y que tenemos 600 vasos blancos y 600 vasos negros. ¿Cuál sería el factor de ramificación de vuestros operadores? ¿Se os ocurre una forma de mejorar el modelo que habéis hecho? Razonad vuestra respuesta (no hace falta que cambiéis vuestro modelo, sólo que indiquéis que cambios habría que hacer.).
4. Como parte de la resolución de un desastre natural provocado por un monzón en China, hemos de resolver el problema de logística producido por el derrumbe de todos los puentes en la confluencia de los rios Xi y Beijang. Tenemos varios camiones que han de transportar víveres y material diverso entre 3 ciudades (Zhaoging, Gaoming y Foshan). Disponemos de una flota limitada de ferries que han sobrevivido al monzón, para llevar los camiones de una ciudad a otra. Pero estos ferries solo puede llevar cada vez como máximo dos camiones a bordo. Nos piden hacer un planificador simple que genere un plan para mover los camiones de su ciudad origen a su ciudad destino por el rio mediante los ferries disponibles.
  - (a) Describe el dominio (incluyendo predicados, acciones, etc...) usando PDDL. Da una explicación razonada de los elementos que has escogido. Ten en cuenta que el modelo del dominio ha de poderse extender a más o menos ciudades y más o menos ferries.

- (b) El jefe de logística nos ha proporcionado una tabla con los camiones que hay que transportar entre Zhaoging (Zh), Gaoming (Gm) y Foshan (Fs). También nos dice que en este momento tenemos sólo un ferry disponible atracado en el puerto de Foshan. Nos pide que el planificador genere un plan en el que el ferry acabe llevando todos los camiones a su ciudad de destino.

camion	origen	destino
c1	Zhaoging	Gaoming
c2	Zhaoging	Foshan
c3	Zhaoging	Foshan
c4	Gaoming	Zhaoging
c5	Gaoming	Zhaoging
c6	Gaoming	Zhaoging
c7	Foshan	Gaoming
c8	Foshan	Zhaoging

Describe este problema usando PDDL. Da una breve explicación de cómo modelas el problema.

5. La FIB nos ha pedido una herramienta sencilla en que los alumnos puedan decir que asignaturas quieren cursar en los últimos semestres de la carrera y les haga un plan de matriculación que, basándose en las asignaturas que ya tienen cursadas, les diga de que asignaturas se han de matricular como mínimo los próximos semestres para poder llegar a matricularse de las asignaturas que no se desean perder.

El sistema tendrá conocimiento sobre las asignaturas del plan de estudios, sus pre-requisitos, sus pre-corequisitos (si los hay), las asignaturas que el alumno ya ha cursado y las asignaturas que el alumno quiere cursar. El resultado es un plan de matriculación que 1) refleja las asignaturas mínimas que el alumno ha de matricular, 2) para cada asignatura, indica en que semestre ha de hacerla, 3) para todas las asignaturas del plan se cumplen en todo momento los pre-requisitos y/o pre-corequisitos necesarios para cursarlas

Para simplificar el problema, 1) no pondremos un límite de créditos matriculables por semestre, 2) del sistema de pre/co/pre-corequisitos solo modelaremos dos tipos de requisitos entre asignaturas

- pre-requisito: Si una asignatura A es **prerequisito** de una asignatura B, para poder matricularse de B se ha de haber cursado la asignatura A en un semestre anterior. Ejemplos de prerequisitos: PRO1⇒BD, PRO1⇒EDA, PRO2⇒BD, PRO2⇒EDA, BD⇒IES, IES⇒AS, AS⇒ECSDI, M1⇒PE, M2⇒PE, EC⇒SO, IC⇒SO, A⇒AA, EDA⇒IDI, EDA⇒IA, EDA⇒PROP, EDA⇒LI, EDA⇒TC, EDA⇒PAR, SO⇒PAR
- pre-corequisito: Si una asignatura A es **pre-corequisito** de una asignatura B, para poder matricularse de B se ha de haber cursado la asignatura A en un semestre anterior o matricularla en el mismo semestre que A (es decir, matricular A y B a la vez). Ejemplos de pre-corequisitos: BD↦PROP, PE↦A, PROP↦IA, PROP↦LI, PROP↦TC, PROP↦A, PE↦A

- (a) Describe el dominio (incluyendo predicados, acciones, etc.) usando PDDL. Da una explicación razonada de los elementos que has escogido.
- (b) Como primera prueba del sistema nos dan un alumno de la FIB que solo ha aprobado PRO1, PRO2, EC, IES y EDA. Nos pide que el planificador genere un plan de matriculación en el que pueda llegar a cursar las asignaturas ECSDI, AA y PAR. Describe este problema usando PDDL. Da una breve explicación de cómo modelas el problema.

6. Hace años la burbuja inmobiliaria y de infraestructuras junto a los delirios de grandeza de algunos políticos provocó la creación de múltiples aeropuertos distribuidos por la geografía española. Sin embargo la crisis económica ha obligado ahora al Gobierno a tomar medidas y exigir a estos aeropuertos un plan de viabilidad para evitar su cierre. Por eso los gestores de un pequeño aeropuerto de la costa Mediterránea nos piden participar en un proyecto piloto de automatización de la torre de control del aeropuerto usando técnicas de inteligencia artificial. En concreto nos piden un sistema automático que, cada media hora, recibe la lista de vuelos que están pendientes de aterrizar o despegar del aeropuerto (y en el caso de los pendientes de despegue, la puerta de embarque que ocupan) y ha de generar un

plan que diga en que orden han de despegar o aterrizar los aviones (y en el caso de los que aterrizan, en que puerta de embarque han de desembarcar).

El aeropuerto dispone de dos pequeñas terminales (*A* y *B*) con cinco puertas de embarque (*A1*, *A2*, *B1*, *B2* y *B3*). Dispone también de dos pistas para el despegue y aterrizaje de aviones (*ps1* y *ps2*), pero los fuertes vientos de la zona hacen que a veces no se pueda usar una de ellas. Nunca puede haber dos aviones a la vez en la misma pista (por ejemplo un avión no puede aterrizar si hay otro avión que acaba de aterrizar o un avión a punto de despegar; un avión no puede entrar en la pista para despegar si hay otro a punto de despegar o un avión aterrizando que no ha salido de la pista). Como las pistas están muy cerca de las puertas de embarque y hay poco espacio para maniobrar, por motivos de seguridad no se permite a un avión en tierra desconectarse de una puerta de embarque e ir a una pista si la pista no está libre de aviones, ni a un avión que acaba de aterrizar dirigirse a una puerta de embarque si esta tiene un avión).

Para simplificar el problema y no tener que modelar tiempos de despegue, aterrizaje o desplazamiento entre pistas y puertas de embarque, aprovechando que el aeropuerto es tan pequeño (y que se han de observar unas distancias de seguridad entre aviones) consideraremos que un avión que ha de despegar solo puede estar o bien en la puerta de embarque, o bien en la pista a punto de despegar o bien ya ha despegado. De igual forma un avión que ha de aterrizar en el aeropuerto sólo puede estar o bien pendiente de aterrizar, aterrizado ya en la pista o en la puerta de embarque para desembarcar. No hace falta modelar los estados intermedios y solo se puede pasar de un estado al siguiente si la pista o la puerta de embarque a la que se ha de mover el avión no tiene otro avión ocupándola.

- (a) Describe el dominio (incluyendo predicados, acciones, etc...) usando PDDL. Da una explicación razonada de los elementos que has escogido. Ten en cuenta que el modelo del dominio ha de poderse extender no sólo a más o menos aviones sino también a más o menos pistas y a más o menos puertas de embarque.
- (b) El aeropuerto nos ha proporcionado el listado de salidas y llegadas programadas para la próxima media hora. También nos avisa que sólo tenemos en estos momentos la pista *ps2* operativa (un fuerte viento lateral hace impracticable usar la otra pista en los próximos minutos). Nos pide que el planificador genere un plan de salidas y llegadas para la próxima media hora en la que se asigne un orden entre los despegues y aterrizajes (no ha de seguir el orden en el que aparecen en la tabla) y se asigne una puerta de embarque para los vuelos que aterrizan.

SALIDAS		
<i>vuelo</i>	<i>destino</i>	<i>puerta</i>
IB0051	Madrid	A1
BA6136	Londres	B1
AF0700	Lyon	B2
AL8860	Milán	B3

LLEGADAS		
<i>vuelo</i>	<i>procedente de</i>	<i>puerta</i>
IB0121	Málaga	—
VY0256	Barcelona	—
KL0333	Amsterdam	—
LH4044	Berlín	—

Describe este problema usando PDDL. Da una breve explicación de cómo modelas el problema.

7. Una asociación de fans de la serie británica *Doctor Who* nos pide una aplicación para móvil que ayude a los nuevos miembros de la asociación a visionar los capítulos de las diferentes temporadas de las series *Doctor Who* (DW), *Torchwood* (TW) y *The Sarah Jane Adventures* (SJ) en el orden adecuado, teniendo en cuenta que a menudo cruzan historias y subtramas entre ellas y que a veces las series se cruzan por completo en capítulos *crossover* (por ejemplo, en el final de la 4ª temporada de *Doctor Who* aparecen todos los personajes principales de las tres series).

La app ha de tener en cuenta, por un lado, los capítulos de las tres series que el usuario ha visto ya, y por el otro un listado de capítulos que quiere ver en los próximos días, haciendo un plan de visionado diario en el que se le recomiende en que orden ha de ver los capítulos teniendo en cuenta sus dependencias:

- *dependencias dentro de la misma serie*: para cada temporada  $i$  de cada una de las series  $Sk$ , cada capítulo  $Sk : i \times j, (j > 1)$  tiene como predecesor el capítulo previo  $Sk : i \times j - 1$  dentro de la temporada (por ejemplo,  $DW:1 \times 4 \rightarrow DW:1 \times 5$ ) y el primer capítulo tiene como predecesor el último capítulo de la temporada anterior (por ejemplo,  $TW:2 \times 13 \rightarrow TW:3 \times 1$ ). Cada temporada de *Doctor Who* tiene 14 episodios (excepto la 4ª temporada, que tiene 18). Cada temporada de *Torchwood* tiene 13 episodios (excepto la 3ª temporada, que tiene 5, y la 4ª temporada, que tiene 10). Las 4 primeras temporadas de *The Sarah Jane Adventures* tienen 6 capítulos y la 5ª temporada tiene solo 3 capítulos.
- *dependencias entre series*: son dependencias entre pares de capítulos de series diferentes. Por ejemplo:

DW:2×3→SJ:1×1	DW:2×13→TW:1×1	TW:1×13→DW:3×11	DW:3×13→TW:2×1
SJ:1×6→DW:4×12	TW:2×13→DW:4×12	DW:4×13→SJ:2×1	DW:4×13→TW:3×1
SJ:2×6→DW:4×17	TW:3×5→DW:4×17	DW:4×18→SJ:3×1	DW:4×18→TW:4×1
SJ:3×6→DW:5×1	DW:5×14→SJ:4×1	SJ:4×6→DW:6×11	TW:4×10→DW:6×11
DW:6×13→SJ:5×1			

visto los capítulos de la 1a y 2a temporada de la serie clásica de Star Trek, el planificador ha de incluir en el plan de visionado todos los capítulos de la 3a temporada (en el orden correcto) y las dos primeras películas de Star Trek. universos ficticios donde las historias no solo pasan una después de la otra, sino también en paralelo, por lo que es bueno ver las historias más o menos en el orden en el que ocurren. Este fenómeno es muy típico de algunas series que han tenido varias subseries emitiéndose a la vez (Stargate, Star Trek, CSI, NCIS, Doctor Who/Torchwood...) y que a veces incluso cruzan historias y subtramas entre ellas.

El resultado es un plan de visionado que lista los capítulos mínimos que el usuario ha de ver para llegar a ver los que quiere en las mejores condiciones, indicando para cada capítulo en que día ha de verlo. El sistema recomendará un máximo de tres capítulos al día, y para todos los capítulos dentro del plan se ha de cumplir que sus predecesores se ven en el mismo día (antes del capítulo en cuestión) o en días anteriores. El plan de visionado no tiene porque llenar los tres slots de visionado cada día, solo ha de intentar hacer un plan factible. Si el usuario tiene pocos capítulos como objetivo el plan puede llegar a ser un solo contenido al día unos pocos días.

- Describe el dominio (incluyendo predicados, acciones, etc...) usando PDDL. Da una explicación razonada de los elementos que has escogido. Ten en cuenta que el modelo del dominio ha de poderse extender no sólo a más o menos capítulos y temporadas sino también a más o menos series (por si en el futuro se crean más series paralelas) y a más o menos días de visionado.
- Para probar el sistema nos piden que nos centremos en las temporadas 3, 4 y 5 de *Doctor Who*, las temporadas 2, 3 y 4 de *Torchwood* y las temporadas 1, 2 y 3 de *The Sarah Jane Adventures*, que es donde hay más lío de cruce de historias entre las series. Queremos simular a un usuario que ha visto ya la 2ª temporada de *Torchwood*\*, la 1ª temporada de *The Sarah Jane Adventures*, la 3ª temporada de *Doctor Who*\* y los primeros 11 capítulos de la 4ª. Durante la semana que viene (7 días) quiere llegar a ver los tres primeros capítulos de la 4ª temporada de *Torchwood*. Describe este problema usando PDDL, incluyendo las dependencias entre capítulos que menciona el enunciado. Da una breve explicación de cómo modelas el problema.

\*No hace falta modelar las temporadas 1 y 2 de *Doctor Who* o la temporada 1 de *Torchwood*, asumimos que si ha visto una temporada también ha visto las anteriores.

- Una coctelería de gran éxito ha decidido ampliar el local para poder colocar muchas más mesas y varias barras en las que preparar los cócteles. Para poder controlar los pedidos en un local tan grande, han decidido crear un sistema informático que conecta a todo el personal en todo momento. Los clientes, en vez de pedir en una de las barras, piden directamente en la mesa, donde un empleado entra el pedido en una tablet. A partir de ahí el sistema ha de planificar las tareas de barmans y camareros. Como ya se ha dicho el local tiene más de una barra, y en cada barra puede haber uno o más barmans. Cada

cóctel pedido por un cliente en una mesa es luego creado por cualquiera de los barmans (pero solo uno, se debe evitar que dos barmans creen el mismo coctel para el mismo pedido). Supondremos que un mismo cliente no pedirá dos veces el mismo coctel. Una vez el barman ha creado un cóctel lo deja en su barra. En el local hay varios camareros, y cualquiera puede venir a coger el cóctel de la barra, ponerlo en su bandeja, llevarlo a la mesa y servirlo (cuidado, ha de saber a que persona de la mesa lo ha de servir). Cada camarero puede llevar en todo momento como máximo tres cócteles en la bandeja con la que se mueven por el local, y en un mismo viaje puede coger cócteles en más de una barra y servirlos en mesas diferentes. El objetivo del planificador es crear el plan en el que todos los pedidos realizados han sido servidos.

- (a) Describe el dominio (incluyendo predicados, acciones, etc...) usando PDDL. Da una explicación razonada de los elementos que has escogido. Ten en cuenta que el modelo del dominio ha de poderse extender no sólo a más o menos personal (barmans y camareros) sino también a más o menos barras y mesas (por si en el futuro se vuelve a ampliar el local).
- (b) Para probar el sistema nos piden que modelemos un local con tres barmans (Manuel, Jorge y Alfonso), dos barras (Manuel y Jorge están juntos en una de las barras), dos camareros (Rosita y Camilo) y diez mesas (M1 a M10). Como ejemplo del tipo de pedidos nos dan la siguiente tabla:

<i>mesa</i>	<i>cliente</i>	<i>coctel</i>
M1	Irene	Mojito
M2	Oscar	SexOnTheBeach
M6	Aida	Margarita
M8	Berto	Daiquiri
M3	Pau	Daiquiri

<i>mesa</i>	<i>cliente</i>	<i>coctel</i>
M1	Pep	TequilaSunrise
M5	Marta	PiñaColada
M7	Montse	SanFrancisco
M8	Alex	BloodyMary
M8	Carla	Mojito

Describe este problema en PDDL. Por favor incluye una corta explicación sobre como modelas el problema.

9. La empresa de aplicaciones para movil *SmartApps* necesita nuestra ayuda para desarrollar el módulo de IA de su nueva app, *ShareBath*. En los pisos pequeños con un solo cuarto de baño suele haber muchos conflictos sobre el uso del baño, especialmente por las mañanas cuando todos tienen que lavarse y arreglarse en un espacio pequeño. Por ello han pensado en una app para familias y estudiantes que comparten piso de un solo baño que les ayude a planificar el orden en el que cada persona ha de hacer sus tareas.

ducha, una delante del lavamanos con espejo y una en el sanitario. Cada persona se ha de duchar, peinar y usar el sanitario. Otras acciones opcionales que dependen de la persona son afeitarse, secar el pelo y maquillarse (las tres requieren el uso del espejo). Las acciones de peinar, secar el pelo y maquillarse se deben hacer siempre después de ducharse (si hay que secar el pelo, peinar es una acción previa para desenredar el pelo y el peinado se acaba durante el secado). Cada persona tiene su propio peine, toalla, maquina de afeitar y maquillaje, pero otras cosas se han de compartir como el sanitario, la ducha, el lavamanos con espejo y el secador de pelo. Para las cosas que se comparten nos piden que se modele explícitamente las acciones de coger y dejar ese recurso, evitando que la app recomiende cosas incompatibles como dos personas secándose el pelo con el mismo secador o una persona afeitándose y otra maquillándose delante del mismo lavamanos con espejo.

En el momento de registrarse en la app el usuario ha de hacer un inventario de los elementos del lavabo que se comparten (sanitario, ducha, lavamanos con espejo, secador) y para cada individuo ha de decir que tareas ha de realizar cada mañana (pensad que no todos los hombres necesitan afeitarse, y que el secador no lo usan solo mujeres, sino también hombres con el pelo largo). El resultado es un plan que lista en que orden han de realizar sus tareas cada uno de los individuos dentro del baño compartido, evitando conflictos. Solo se pide un orden de las tareas, no se tiene que modelar su duración o hacer una planificación minuto a minuto.

- (a) Describe el dominio (incluyendo predicados, acciones, etc...) usando PDDL. Da una explicación razonada de los elementos que has escogido. Ten en cuenta que el modelo del dominio ha de



poderse extender no sólo a más o menos usuarios del baño sino también a más o menos cosas compartidas (por ejemplo baños con dos lavamanos o con dos secadores de pelo).

- (b) Para probar el sistema nos dan el caso particular de una familia numerosa de 7 personas que comparte un baño con un sanitario, una ducha, dos lavamanos con espejo y un secador de pelo. segundo secador que tenían se ha estropeado). En el momento del registro la familia nos ha pasado la siguiente lista de tareas para cada individuo (cuidado, la lista no está ordenada).

<i>persona</i>	<i>edad</i>	<i>tareas</i>
Juan	48	afeitarse, ducharse, peinarse, usar sanitario
Montse	45	ducharse, maquillarse, secar el pelo, peinarse, usar sanitario
Pablo	22	afeitarse, ducharse, secar el pelo, peinarse, usar sanitario
Miguel	19	afeitarse, ducharse, peinarse, usar sanitario
Jesus	17	ducharse, peinarse, usar sanitario
Martina	14	ducharse, secar el pelo, peinarse, usar sanitario
Oscar	12	ducharse, peinarse, usar sanitario

Describe este problema usando PDDL. Da una breve explicación de cómo modelas el problema.

10. Cada día los dispositivos móviles (smartphones, tablets) son más potentes y complejos y se parecen más y más a los ordenadores. Por eso una empresa está empezando a crear un nuevo sistema operativo para la próxima generación de dispositivos. Una de las novedades es un sistema inteligente de gestión de la descarga e instalación de paquetes de software. Inspirado en el sistema de Linux, quieren que su sistema operativo sea capaz de gestionar él solo las dependencias entre paquetes de forma que, cuando un usuario indique que desea instalar una aplicación, se genere automáticamente un plan de descargas de todos los paquetes necesarios y se instalen en el orden adecuado para que funcionen.

Un plan de instalación está formado por una secuencia de pasos. En cada paso o bien se baja un paquete a instalar en el dispositivo o bien se instala un paquete previamente bajado. No hace falta que un paquete se instale inmediatamente después de bajarlo al dispositivo, pero un paquete solo se puede instalar si se ha bajado en algún paso anterior.

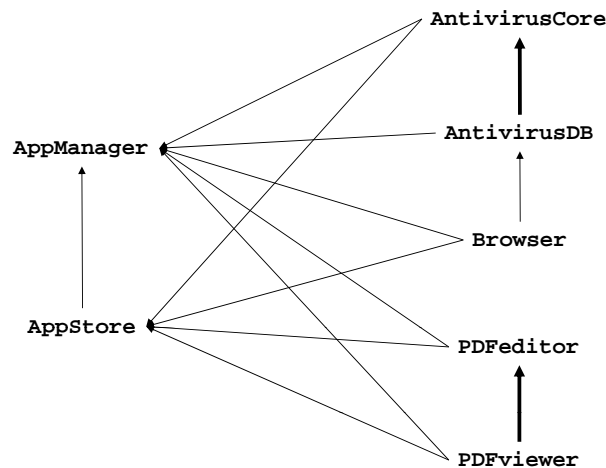
Cada paquete (una app o una librería) tiene asociada una lista de paquetes de los que depende (paquetes que han de estar instalados en el dispositivo antes del paquete que se quiere instalar). Si un paquete A depende de un paquete B para funcionar, para instalar el paquete A el paquete B debe haber sido instalado en el dispositivo en algún paso anterior (no tiene porque ser justo en el paso anterior a la instalación de A). El orden de bajada de los paquetes al dispositivo no ha de contemplar las dependencias (no hace falta bajar el paquete B antes del paquete A).

Existen casos especiales de dependencia, la dependencia absoluta. Si el paquete A depende absolutamente del paquete B, el paquete A ha de ser instalado justo en el paso siguiente tras la instalación del paquete B, y no puede haber ninguna otra acción insertada entre la instalación de B y la instalación de A. Eso significa que la bajada del paquete A no se puede hacer entre la instalación de B y A, y por ello el paquete A debe bajarse antes de la instalación de B. Un paquete A puede depender de varios paquetes, pero por razones obvias solo puede tener un paquete del que dependa de forma absoluta.

El planificador recibe una lista de los paquetes de software (apps y librerías) que ya están instalados en el dispositivo y la lista de paquetes de software que el usuario quiere instalar. El planificador tiene además una lista de dependencias normales entre paquetes y una lista de dependencias absolutas entre paquetes que han de instalarse el uno a continuación del otro. Con todo esto ha de crear el plan de instalación, indicando en cada paso del plan que acción se ha de realizar (bajar un paquete, instalar un paquete).

- (a) Describe el dominio (incluyendo predicados, acciones, etc...) usando PDDL. Da una explicación razonada de los elementos que has escogido. Ten en cuenta que el modelo del dominio ha de poderse extender a más o menos paquetes y a más o menos dependencias entre paquetes.

- (b) Para probar el sistema nos piden que modelemos una petición particular de instalación en la que el usuario ya tiene instalados dos paquetes (el **AppManager** y la **AppStore**) y quiere instalar el **Browser** y un **PDFviewer**. A continuación se muestran las dependencias entre los paquetes que se han de observar ( $A \rightarrow B$  significa que A depende de B, en **negrita** las dependencias absolutas que requieren la instalación en dos pasos consecutivos):



Describe este problema en PDDL. Por favor incluye una corta explicación sobre como modelas el problema.

11. La cirugía ha avanzado tanto en los últimos años que, para las intervenciones menores, los quirófanos han adaptado su funcionamiento al sistema de producción en cadena, pudiendo tratar muchos pacientes cada hora. Un ejemplo de ello son las operaciones de cataratas. El paciente llega al hospital, se le hace pasar a una sala donde se quita la ropa y se pone la bata, luego de allí pasa a una camilla donde se le aplica anestesia local en la superficie del ojo, se opera en tan solo unos minutos, se deja al paciente descansar unos minutos para que se recupere de la anestesia, y después el paciente vuelve a vestirse y sale por su propio pie del hospital, con un vendaje protegiendo el ojo.

Un centro de referencia en cirugía ocular nos ha pedido un sistema planificador que establezca un orden correcto para los diferentes pasos del proceso durante una hora, evitando bloqueos. El sistema recibe la lista de pacientes a tratar e irá simulando como van pasando de un paso a otro. Para simplificar el modelo, todos los pasos tienen la misma duración (entrada y registro, desvestirse, anestesia y operación, recuperación, vestirse y salir), y solo se han de modelar cuatro espacios: la sala de entrada, los vestidores, el quirófano con sus camillas y la sala de salida. Tanto la sala de entrada como la de salida tienen capacidad de sobra para todos los pacientes de una hora. En el caso de los vestidores son pequeñas habitaciones donde el paciente se quita la ropa y se pone la bata de quirófano, antes de la operación, y tras ella se quita la bata y vuelve a vestirse con la ropa de calle. Cada vestidor tiene capacidad para una sola persona a la vez y normalmente hay menos vestidores que pacientes, por lo que se ha de controlar que en cada momento haya como máximo una sola persona por vestidor. Tras desvestirse el paciente solo podrá entrar a quirófano cuando haya una camilla libre. En esa camilla se mantendrá durante la anestesia, operación y recuperación (son los médicos los que se van moviendo de una camilla a otra para operar cuando el paciente está listo). Cuando el paciente se recupera de la anestesia, ha de esperar a que haya un vestidor libre para volver a vestirse y marchar. Hay que tener cuidado en este punto, ya que se podrían crear bloqueos si tenemos todos los vestidores ocupados con pacientes esperando operación y todas las camas del quirófano ocupadas con pacientes que querrán volver a

vestirse.

- (a) Describe el dominio (incluyendo predicados, acciones, etc...) usando PDDL. Da una explicación razonada de los elementos que has escogido. Ten en cuenta que el modelo del dominio ha de poderse extender a más o menos vestidores y a más o menos camillas dentro del quirófano. Supondremos que siempre hay un solo quirófano.
- (b) Para probar el sistema nos piden que modelemos un centro con una sala de entrada, tres vestidores, un quirófano con cuatro camillas y una sala de salida. Tenemos 7 pacientes por operar: Joan, Luis, Mireia, Alba, Ester, Pau i Quim. No hay orden de entrada prefijado entre estos pacientes, el planificador nos dirá quienes entran primero en los vestidores para desvestirse y los pasos posteriores hasta que todos los pacientes hayan sido operados y hayan salido del centro.  
Describe este problema en PDDL. Por favor incluye una corta explicación sobre como modelas el problema.

12. Una de las cosas que crea más quebraderos de cabeza en los preparativos de una boda es el colocar correctamente a los invitados en las diferentes mesas del banquete de forma que todo el mundo esté contento. Por ello un grupo de emprendedores quiere crear una app que ayude a las parejas a planificar la colocación de sus invitados en el banquete.

Para cada una de las mesas que hay en la sala del banquete, la app recibe el número máximo de sillas que se pueden colocar en la mesa. También recibe un listado de los invitados al banquete. Para evitar problemas, la app permite introducir dos tipos de restricciones:

- **La persona A necesita estar al lado de la persona B:** esta restricción indica que la persona B debe estar sentada o bien en la silla inmediatamente a la izquierda de A o en la silla inmediatamente a la derecha de A, y está pensada para que se pueda indicar que, por ejemplo, un niño muy pequeño tiene que estar en la silla al lado su madre, que una persona mayor tiene que estar al lado de alguien que se hará cargo de ella, o que una pareja de hermanos/as, amigos/as o casados/as no quieren estar separados durante el banquete;
- **La persona A no soporta a la persona B:** a la app no le importa saber la naturaleza o los motivos de esa enemistad, pero lo que hará es evitar que A y B estén en la misma mesa.

En ambos casos, si la relación de necesidad o de rechazo es en ambos sentidos, la app solo guarda uno de los sentidos para ahorrar memoria. Por ahora otros criterios de colocación de invitados en el banquete, como intentar intercalar a las personas por género dentro de una mesa, o la cercanía de las mesas de invitados entre sí o a la mesa de los novios NO serán soportados por la app en su primera versión.

Este grupo de emprendedores nos pide que desarrollemos un planificador simple que genere una asignación de invitados a mesas de forma que se cumplan las dos restricciones descritas, o que nos diga que no existe asignación posible.

- (a) Describe el dominio (incluyendo predicados, acciones, etc...) usando PDDL. Da una explicación razonada de los elementos que has escogido. Ten en cuenta que el modelo del dominio ha de poderse extender a más o menos mesas, a mesas con diferente número máximo de sillas y a más o menos invitados.
- (b) Para probar nuestro planificador el jefe del proyecto nos ha proporcionado los datos de una boda reciente, en concreto dos mesas de amigos de los novios. Una mesa puede tener como máximo 7 sillas y la otra máximo 6 sillas. En ellas se ha de colocar a 12 amigos de los novios:
  - *Antonio y Ainhoa* son recién casados y, por lo tanto, inseparables,
  - *Lola y Lolo* son mellizos y van juntos a todas partes,
  - *Esperanza* pidió a los novios que la colocaran al lado de *Enrique*,
  - *Susana y Manu* son pareja y tienen dos hijos, *Manolito* y *Susanita*, que no comen bien, y por ello necesitan estar los dos padres en la misma mesa de los niños para tenerlos controlados,
  - *Jesus y Quico* vienen solos al banquete y no necesitan de nadie.

Desafortunadamente hay desavinencias entre los amigos: *Ainhoa* no soporta a *Susana*, y *Quico* terminó de forma unilateral su relación con *Esperanza* hace unas semanas.

Describe este problema usando PDDL. Da una breve explicación de cómo modelas el problema.

13. Una empresa española ha decidido entrar en el sector de los coches autónomos poco a poco. Bajo el nombre de *GoAuto* ha desarrollado tres tipos de vehículo eléctrico totalmente autónomos que no tienen volante ni otro tipo de mecanismos de manejo manual: el *GoAuto2*, vehículo de 2 plazas similar al de Google, el *GoAuto4* con similares características y 4 plazas, y el *GoAuto7* con 7 plazas. Para ayudar al desarrollo de la tecnología la empresa ha creado un proyecto piloto de movilidad urbana autónoma personalizada en las calles de una pequeña ciudad para substituir a los taxis y a los micro-buses. Se ha creado una red de 15 *GoPuntos*, puntos de recogida que substituyen a las antiguas paradas de bus y taxi de la población, con la idea de crear más puntos en el futuro. Los ciudadanos pueden pedir a través de una app que uno de estos vehículos los lleve desde un punto de recogida a otro punto en la ciudad. Alguno de los vehículos autónomos que tenga plazas libres recogerá al ciudadano en el punto de recogida y lo llevará al punto de destino, pudiendo parar por el camino en otros puntos para recoger o descargar pasajeros. La empresa nos pide que desarrollemos un planificador simple que genere un plan para servir las peticiones de movilidad de los ciudadanos con los vehículos autónomos disponibles en cada momento (al ser prototipos a veces falla alguno y se han de servir las peticiones con el resto).
- Describe el dominio (incluyendo predicados, acciones, etc...) usando PDDL. Da una explicación razonada de los elementos que has escogido. Ten en cuenta que el modelo del dominio ha de poderse extender a más o menos *GoPuntos* y más o menos *GoAutos*.
  - El jefe del proyecto nos ha proporcionado una tabla como ejemplo de las peticiones de movilidad que el sistema recibe cada 15 minutos, con una serie de ciudadanos que se han de transportar entre los 15 puntos de la red (*GoPunto1*, *GoPunto2* ... *GoPunto15*). También nos dice que al inicio de ese intervalo de 15 minutos tenemos un vehículo *GoAuto2* de 2 pasajeros vacío estacionado en *GoPunto3* y un vehículo *GoAuto4* de 4 pasajeros vacío estacionado en *GoPunto7*, y que el vehículo *GoAuto7* no está disponible por avería. Nos pide que el planificador genere un plan en el que los vehículos disponibles acaben llevando todos los ciudadanos a sus puntos de destino.

ciudadano	GoPunto origen	GoPunto destino
Julian	GoPunto1	GoPunto3
Paula	GoPunto7	GoPunto12
David	GoPunto14	GoPunto10
Eva	GoPunto3	GoPunto5
Laura	GoPunto3	GoPunto7
Felipe	GoPunto3	GoPunto8
Amelia	GoPunto1	GoPunto5
Ivan	GoPunto13	GoPunto8
Daniela	GoPunto6	GoPunto11
Hugo	GoPunto12	GoPunto10

Describe este problema usando PDDL. Da una breve explicación de cómo modelas el problema.

14. En un intento de atraer más visitantes, la red de albergues *Future Hostel* está buscando nuevos servicios que ofrecer a sus clientes. Tras un acuerdo firmado con una start-up que construye robots humanoides, los albergues de la red ofrecerán un servicio de habitaciones robotizado por las noches, cuando no hay personal humano en el albergue. Los clientes podrán solicitar a través de una app algunos productos (mantas, almohadas, toallas, papel de baño, adaptadores de enchufe, snacks, refrescos, cerveza, agua mineral y agua con gas) que les serán llevados a su dormitorio. Además, los clientes podrán pedir también algunas preparaciones recién hechas (sandwiches, omelettes y zumo de naranja).

Un planificador recibe las peticiones, cada una de ellas incluye el cliente, el dormitorio compartido en el que duerme, y el producto o preparación que solicitan, y crea un plan para servir las peticiones con los robots disponibles. En el caso de los productos, estos se encuentran almacenados en diferentes lugares del

albergue (la cocina, almacenes), y los robots asistentes se usarán para llevar esos productos desde el lugar de almacenaje hasta el dormitorio del cliente. Para transportar los productos estos robots asistentes tienen contenedores especiales (normalmente 2) instalados en sus cuerpos. Un robot asistente solo puede llevar un producto o preparación por contenedor, y no puede coger más productos/preparaciones si tiene todos sus contenedores llenos. Para los productos almacenados consideramos que hay más que suficientes para satisfacer todas las demandas de una noche, y por lo tanto un robot que vaya a buscar un producto a su lugar de almacenamiento siempre encontrará uno disponible. En el caso de preparaciones recién hechas los robots asistentes deben esperar a que la preparación sea realizada por un tipo especial de robot, el robot cocinero. Los robots cocineros solo son capaces de realizar las preparaciones, y solo las concinarán bajo demanda, es decir, que solo cocinarán una preparación cuando aparece una petición del cliente. Los robots asistentes solo irán a buscar las preparaciones a la cocina cuando estén listas, y será entonces cuando las lleven en sus contenedores al dormitorio correspondiente.

La compañía quiere que el planificador modele todos los pasos de la solución: el robot cocinero cocina las preparaciones, el robot asistente va al lugar donde el producto/preparación está disponible, el robot asistente recoge el producto/preparación del lugar en el que se encuentra, el robot asistente va a un dormitorio, el robot asistente entrega el/los producto(s)/preparación(es) que corresponde(n) a ese dormitorio al cliente correcto.

- (a) Describe el dominio (incluyendo predicados, acciones, etc...) usando PDDL. Da una explicación razonada de los elementos que has escogido. Ten en cuenta que el modelo del dominio ha de poderse extender a más o menos dormitorios, clientes, robots cocineros, robots asistentes (con más o menos contenedores).
- (b) El jefe de Future Hostels nos ha proporcionado una tabla como ejemplo de las peticiones que el sistema recibe cada 30 minutos, con una serie de productos/preparaciones que se han de transportar desde dos almacenes (almacen1 para las mantas, almohadas y toallas, almacen2 para el papel de baño y los adaptadores de enchufe) o desde la única cocina en el hostel (snacks, refrescos, cerveza, agua mineral, agua con gas, sandwiches\*, omelettes\* y zumo de naranja\*), donde el \* indica que son productos frescos recién hechos. Cada uno de esos productos/preparaciones debe ser transportada a uno de los 6 dormitorios en el albergue (H101, H102, H103, H201, H202, H203). También nos dice que al inicio de ese intervalo de 30 minutos tenemos solo el robot cocinero *R1C1* en la cocina, y dos robots asistentes, *r2W1* and *r2W2* que esperan los pedidos desde *Almacen1* y *Almacen2* respectivamente. Se nos pide que el planificador genere un plan en el que los robots disponibles sirvan todos los pedidos, transportando cada producto/preparación al dormitorio correspondiente.

cliente	dormitorio	producto o preparación*
Virginia	H102	manta
Sam	H101	almohada
Rick	H101	sandwich*
Catalina	H201	toalla
Ana	H201	zumoNaranja*
Paul	H103	omelette*
Agnes	H203	papelWC
Manuel	H202	sandwich*
Virginia	H102	adaptador
Frank	H202	cerveza

Describe este problema usando PDDL. Da una breve explicación de cómo modelas el problema.

15. Queremos desarrollar una app capaz de recomendar programas de entrenamiento físico al usuario del dispositivo. La app posee un conjunto de ejercicios que se pueden realizar dentro o fuera de un gimnasio, como ejercicios de suelo, ejercicios de estiramientos/flexibilidad, ejercicios con aparatos (bicicleta estática, cinta de andar, remo, stepper, pesos...), ejercicios con o sin pesas para los diferentes grupos musculares, correr, patinar, nadar, etc. Cada ejercicio tiene diez niveles de dificultad (del

1 al 10), donde el aumento de dificultad en un mismo ejercicio se consigue alargando su duración, incrementando su intensidad, el peso a mover, el número de series o reduciendo el tiempo de descanso entre series.

La interfaz de la app permite al usuario indicar los ejercicios que el usuario está realizando en este momento y su nivel de dificultad inicial (del 1 al 10), y el nivel de dificultad que quiere llegar a alcanzar para algunos de ellos, y la app le generará un plan de entrenamiento físico semanal especificando los ejercicios a realizar cada día y su orden, con el objetivo de conseguir subir de nivel en el conjunto de ejercicios escogidos. El plan de entrenamiento ha de tener en cuenta las siguientes dependencias y restricciones:

- ejercicios preparadores: son ejercicios que se han de hacer antes de un cierto ejercicio en el mismo día de entrenamiento pero no tienen porque ir inmediatamente antes (un ejemplo son los estiramientos, que preparan para varios ejercicios de la sesión de entrenamiento). Es una restricción estricta que no se puede romper.
- ejercicios nivel +1: sirve para modelar la evolución en el nivel de dificultad de un ejercicio según pasan los días. Para cada ejercicio podremos subir un nivel  $N + 1$  solo si ese ejercicio se ha realizado en algún día anterior en el nivel  $N$ .
- no repetir ejercicio el mismo día: en un día de entrenamiento no se debe repetir el mismo ejercicio
- límite de ejercicios por día: no se pueden colocar más de 6 ejercicios en un día de entrenamiento.

La app ya tiene cargada la lista completa de ejercicios, sus niveles y los ejercicios preparadores de un ejercicio. Del usuario recibe la lista de los ejercicios que ya está haciendo (y en que nivel los hace) y los niveles incrementados de algunos de los ejercicios a los que quiere llegar al final de la semana. El resultado es un plan de entrenamiento que refleja, para cada día, la lista ordenada de los ejercicios que el usuario ha de hacer para llegar a los niveles objetivo con cumpliendo las dependencias y restricciones, o un mensaje del planificador diciendo que no se puede llegar a esos niveles en solo 7 días. El planificador no tiene que rellenar el plan con ejercicios extra, solo los necesarios para llegar al día 7 a los niveles deseados. Si sobran huecos se entiende que se le da al usuario libertad para hacer otros ejercicios suplementarios al plan de entrenamiento. En caso de que un usuario no especifique nivel inicial para un ejercicio el sistema asume que ha de empezar ese ejercicio en el nivel 1.

- Describe el dominio (incluyendo predicados, acciones, etc...) usando PDDL. Da una explicación razo-nada de los elementos que has escogido. Ten en cuenta que el modelo del dominio ha de poderse extender a más o menos días y más o menos ejercicios por día.
- El jefe del proyecto nos ha proporcionado un ejemplo de uso del sistema, con los ejercicios que está haciendo y sus niveles objetivo.

PUNTO DE PARTIDA ejercicio (nivel actual)	OBJETIVOS DEL ENTRENAMIENTO: ejercicio (nivel final)
estiramientos (nivel 1)	
correr (nivel 1)	correr (nivel 4)
nadar (nivel 1)	nadar (nivel 4)
cuádriceps (nivel 1)	cuádriceps (nivel 4)
isquios (nivel 2)	
gemelos (nivel 3)	
abductores (nivel 2)	abductores (nivel 6)
remo (nivel 2)	remo (nivel 3)
dominadas (nivel 3)	dominadas (nivel 6)
pectoral (nivel 3)	pectoral (nivel 6)
bíceps (nivel 3)	bíceps (nivel 7)
tríceps (nivel 3)	tríceps (nivel 6)
dorsal (nivel 2)	dorsal (nivel 4)

También nos dan la siguientes listas de ejercicios preparadores:

- los estiramientos son ejercicio preparador de todos los demás;
- las sentadillas son ejercicio preparador de cuádriceps y correr;
- las dominadas son ejercicio preparador de pectoral;
- cuádriceps ejercicio preparador del de remo.

Describe este problema usando PDDL. Da una breve explicación de cómo modelas el problema.

16. La empresa *TraeMeLo* quiere convertirse en la líder dentro del sector emergente de empresas que reciben pedidos de sus clientes para ir a recoger algún producto/compra en uno de los comercios registrados en el sistema (normalmente una tienda, una farmacia, un restaurante) y traérselo a sus casas, siempre que sea dentro de la misma ciudad, no exceda los 10kg de peso y con un volumen de máximo 40x40x30cm. Los pedidos los llevan unos mensajeros, que pueden ir en bici (con un cajón trasero en el que pueden llevar un máximo de 2 pedidos a la vez) o en scooter (con un cajón trasero en el que pueden llevar un máximo de 4 pedidos a la vez).

Los clientes pueden solicitar a través de una app la recogida de un pedido en un comercio registrado (la app envía un identificador de comercio, y el sistema tiene un método auxiliar para convertir ese string en coordenadas de GPS) y la entrega en la dirección registrada del cliente (hay que tener en cuenta que un cliente puede realizar más de un pedido, y en una misma dirección puede haber más de un cliente). El sistema entonces queda a la espera a que el comercio envíe un aviso de que el pedido está listo para ser recogido, y en ese momento asigna alguno de los mensajeros que tenga espacio dentro del cajón trasero para recoger el pedido en el comercio y llevarlo a su destino, pudiendo parar por el camino en otros puntos para recoger o entregar pedidos. La empresa nos pide que desarrollemos un planificador simple que genere un plan para recoger y entregar los pedidos recibidos en un periodo de tiempo con los mensajeros disponibles en cada momento (a veces una avería o una baja por enfermedad hace que ese número varíe). En el modelo deben quedar explícitos los pedidos a recoger, el aviso del comercio de que el pedido está listo, la recogida del pedido por parte del mensajero y su entrega al cliente.

- (3.5 puntos) Describe el dominio (incluyendo predicados, acciones, etc...) usando PDDL. Da una explicación razonada de los elementos que has escogido. Ten en cuenta que el modelo del dominio ha de poderse extender a más o menos mensajeros, más o menos comercios, más o menos clientes y a futuros vehículos capaces de llevar más pedidos a la vez.
- (1.5 puntos) El jefe del proyecto nos ha proporcionado una tabla como ejemplo de las peticiones que el sistema recibirá en un periodo de 20 minutos, con una serie de pedidos que se han de transportar desde un comercio registrado hasta la dirección del cliente. También nos dice que al inicio de ese intervalo de 20 minutos tenemos disponibles dos motos (capacidad máxima de 4 pedidos) y una bici (capacidad máxima de 2 pedidos) estacionadas en el parking central de *TraeMeLo*, y que tras entregar todos los pedidos del periodo las motos y la bici han de volver a ese parking.

pedido	dirección recogida	dirección entrega	cliente
p1	PizzaHotBalmes88	PauClaris55	Mario
p2	TodoPastaAragon167	Meridiana131	Aitana
p3	SupermercadoBruc31	Villaroel21	Pau
p4	frankfurtCasp13	ManuelGirona86	Amalia
p5	MercahoyValencia216	RamblaPrim12	Jaume
p6	Farmacia24hDiagonal340	Meridiana131	Aitana
p7	TodoPastaAragon167	Lepant95	Nuria
p8	FarmaciaGarciaSardenya501	Lepant95	David
p9	PizzaHotBalmes88	RocBoronat01	Esther
p10	BurgerFastCiencies62	AvCarrilet73	Miquel
p11	Farmacia24hDiagonal340	AvCarrilet73	Miquel
p12	MercahoyValencia216	RamblaPrim12	Laura

Describe este problema usando PDDL. Da una breve explicación de cómo modelas el problema.

17. La empresa *ComeBien* quiere un sistema inteligente que ayude a programar los menús semanales en su cadena de restaurantes.

Un menú está formado por un primero y un segundo para cada día laborable (de lunes a viernes). Asumiremos que el cocinero es capaz de preparar una serie de primeros y segundos que podremos combinar en los menús. Cada plato tendrá asociado un tipo (sopa, crema, ensalada, carne, pescado, ...).

Se ha de tener en cuenta las incompatibilidades entre los primeros y segundos de manera que no pueden aparecer el mismo día del menú (por ejemplo ensaladilla de primero y fabada asturiana de segundo). También se quiere que no se utilice un mismo plato más de una vez a la semana.

También podemos tener restricciones en el tipo de plato (sopa, crema, ensalada, carne, pescado, ...) entre días consecutivos de manera que este no pueda repetirse. La restricción la observaremos para primeros y segundos separadamente de manera que no podemos poner dos días seguidos dos primeros platos de pescado (salmón un día y lubina el siguiente), pero puede haber un primero de pescado un día (paella) y un segundo de pescado al día siguiente (salmón).

Nos piden que el planificador genere el menú para una sola semana (primero y segundo, de lunes a viernes) sin repetir platos, sin incompatibilidades en el mismo día y cumpliendo las restricciones de tipo de plato en días consecutivos. Ha de quedar claro para cada plato en que día se sirve y si es primero o segundo.

- (a) Describe el dominio (incluyendo predicados, acciones, etc...) usando PDDL. Da una explicación razonada de los elementos que has escogido. Ten en cuenta que el modelo del dominio ha de poderse extender a más o menos platos, y más o menos incompatibilidades.
- (b) El director de la área TIC de *ComeBien* nos ha proporcionado una lista de platos como ejemplo de la entrada que recibiría el planificador, y un listado de las incompatibilidades entre ellos. Se nos pide que el planificador genere un plan usando esta información.

PRIMEROS		SEGUNDOS	
<i>plato</i>	<i>tipo</i>	<i>plato</i>	<i>tipo</i>
ensalada_tomate	ensalada	fabada	carne
crema_calabaza	crema	filete_ternera	carne
ensaladilla	ensalada	plato_salmon	pescado
paella	pescado	plato_dorada	pescado
escalivada	ensalada	plato_pollo	carne
sopa_pollo	sopa	butifarra	carne

incompatibilidades	
<i>PRIMERO</i>	<i>SEGUNDO</i>
sopa_pollo	butifarra
paella	plato_salmon
crema_calabaza	fabada
ensaladilla	fabada
crema_calabaza	filete_ternera
crema_calabaza	plato_pollo

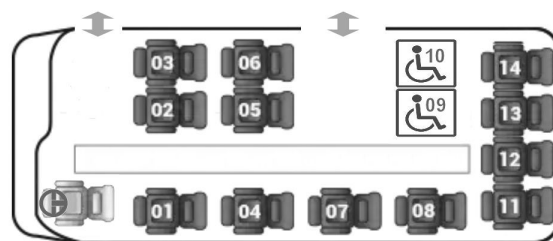
Describe este problema usando PDDL. Da una breve explicación de cómo modelas el problema.

18. Una ciudad del este de Europa quiere posicionarse en el top 15 de las Smart Cities europeas incorporando la digitalización a la movilidad urbana. Después de reorganizar el transporte en el denso núcleo urbano, quieren solucionar ahora el problema de movilidad para todos los barrios periféricos (compuestos de pequeñas casas dispersas entre colinas, y donde no tiene sentido poner líneas de autobuses de recorrido fijo). Quieren crear el SmartBus, líneas de autobús de barrio donde el recorrido se adapta a los usuarios que haya en cada momento.



Cada barrio tendrá asignado uno (o más) autobuses de 14 plazas (12 asientos y 2 espacios reservados para personas de movilidad reducida). Por cada barrio se distribuyen tantas paradas del SmartBus como haga falta, y una o más paradas les permiten transbordar hacia otros medios de transporte del resto de la ciudad. Los vecinos del barrio pueden pedir, a través de una app, que un SmartBus los lleve desde una parada del barrio a otra. Uno de los autobuses que tenga plazas libres recogerá al viajero en la parada de origen y lo llevará a la parada de destino, pudiendo parar por el camino en otras paradas de SmartBus para recoger o descargar pasajeros.

La empresa nos pide que desarrollemos un planificador simple que genere un plan para servir las peticiones de movilidad de un barrio con los autobuses disponibles en cada momento (según la hora del día).



El resultado del plan ha de mostrar como va cada autobús de parada en parada, y para cada una de sus paradas 1) quien baja del autobús y 2) quien sube al autobús y que asiento tiene asignado.

- (3,5 puntos) Describe el dominio (incluyendo predicados, acciones, etc...) usando PDDL. Da una explicación razonada de los elementos que has escogido. Ten en cuenta que el modelo del dominio ha de poderse extender a más o menos paradas del SmartBus y más o menos autobuses.
- (1,5 puntos) El jefe del proyecto nos ha proporcionado una tabla como ejemplo de las peticiones de movilidad que el sistema recibe cada 15 minutos, con una serie de ciudadanos que se han de transportar entre las 14 paradas del barrio (p2701, p2702 ... p2714) (los pasajeros con \* son de movilidad reducida y solo pueden usar las plazas 09 o 10). También nos dice que al inicio de ese intervalo de 15 minutos tenemos solo el autobus *SBus027*, vacío y estacionado en la parada p2701. Nos pide que el planificador genere un plan en el que el vehículo disponible acabe llevando todos los ciudadanos a sus paradas de destino.

pasajero	parada origen	parada destino	pasajero	parada origen	parada destino	pasajero	parada origen	parada destino
Esteban	p2701	p2703	Cristina	p2707	p2712	Alba*	p2703	p2707
Daniel	p2714	p2710	Luis	p2711	p2705	Beatriz	p2703	p2708
Lorena	p2703	p2707	Jesus	p2703	p2708	Jordi	p2706	p2705
Raul	p2701	p2705	Maria	p2713	p2708	Kilian*	p2713	p2708
Estela	p2706	p2711	Sebas*	p2712	p2710	Fabiola*	p2706	p2710
Martina	p2701	p2711	Noelia	p2707	p2705	Susana	p2714	p2710
Pau	p2704	p2710	Xavier*	p2703	p2705	Ruth	p2706	p2710

Describe este problema usando PDDL. Da una breve explicación de cómo modelas el problema. No hace falta incluir en el modelo toda la tabla de pasajeros con sus paradas de origen y destino, es suficiente modelar algunos. En vuestro modelo 1) debe haber al menos un ejemplo de uso de todos los predicados que deban ser inicializados en el fichero del problema, 2) la condición de fin si se ha de estar completamente modelada.

