Inteligencia Artificial 1/2003

Práctica 1 – Resolución de Problemas

I- Formulación de problemas

a) **SAT. Satisfacibilidad**. En el problema de k-SAT tenemos un número de cláusulas, y cada cláusula contiene cuanto más k literales. Un literal es una variable booleana o su complemento. Se dice que una cláusula está satisfecha si al menos un literal es verdadero en la cláusula. El objetivo en el k-SAT es, dado un conjunto de cláusulas, determinar si existe una asignación a las variables que satisface todas las cláusulas. Se sabe que k-SAT para $k \geq 3$ es NP-completo, mientras que 2-SAT es resoluble en tiempo lineal en el número de cláusulas y el número de variables.

b) TSP. Problema del viajante. Un viajante debe visitar cada ciudad en su territorio exactamente una vez y luego volver a su punto de partida habiendo recorrido la distancia más corta.

c) NLP. Problema de programación no lineal. Maximizar la función:

$$G2(\mathbf{x}) = \left| \frac{\sum_{i=1}^{n} \cos^4(x_i) - 2\prod_{i=1}^{n} \cos^2(x_i)}{\sqrt{\sum_{i=1}^{n} i x_i^2}} \right|,$$

d) Una fábrica produce autos de n colores distintos. La tarea es encontrar un programa (schedule) óptimo de producción que minimice el costo total de la tarea de pintado de los autos. Cada máquina involucrada en la línea de producción debe cambiar de color entre tareas y el costo del cambio depende de los colores involucrados y del orden. El costo de pasar del amarillo al negro podría ser de 30 unidades. Esto podría medirse en pesos, minutos o cualquier otro estándar. El costo de cambiar del negro al amarillo podría ser de 80 unidades. El costo de pasar del amarillo al verde podría ser de 35 unidades, etcétera. Para minimizar costos se debe encontrar una secuencia de tareas que satisfaga todos los requerimientos de producción para el número de autos de cada color, de forma oportuna, manteniendo los costos de operación tan bajos como sea posible.

e) Una compañía tiene n depósitos que almacenan resmas de papel. Estas se envían a k centros de distribución. Los depósitos y centros de distribución pueden considerarse fuentes y destinos respectivamente. Cada ruta de distribución posible entre un depósito i y un centro de distribución j tiene un costo mensurable de transporte que está determinado por una función f_{ij}. La forma de esta función depende de una variedad de factores que incluye la distancia entre el depósito y el centro de distribución, la calidad de la ruta, la densidad del tráfico, el número de paradas necesarias, el límite de velocidad promedio, etcétera. Por ejemplo, la función de costo de transporte entre el depósito 2 y el centro de distribución 3 podría definirse así:

$$f_{23}(x) = \begin{cases} 0 & \text{if } x = 0 \\ 4 + 3.33x & \text{if } 0 < x \leq 3 \\ 19.5 & \text{if } 3 < x \leq 6 \\ 0.5 + 10\sqrt{x} & \text{if } 6 < x, \end{cases}$$

donde x refiere a la cantidad de resmas transportadas de i a j. Debe encontrarse una distribución óptima.

f) Horario cuatrimestral de materias. En el próximo cuatrimestre se dictarán en el departamento de Computación n materias y se cuenta con m aulas. Se conoce la lista de los alumnos inscriptos en cada una de esas materias, la capacidad de cada aula y los recursos tecnológicos de la que disponen. La tarea es construir dicho horario.

g) Sea el siguiente juego de palabras. Dadas dos palabras de la misma longitud, debe cambiar la primera palabra una letra por vez, transformándola en una palabra significativa, hasta llegar a obtener la segunda palabra. Por ejemplo dado mapa y tina, podríamos tener:

mapa $\rightarrow$ tapa $\rightarrow$ tipa $\rightarrow$ tina.
mapa $\rightarrow$ capa $\rightarrow$ cepa $\rightarrow$ cena $\rightarrow$ cuna $\rightarrow$ tuna $\rightarrow$ tina

h) Demostración lógica. Demostrar que cualquier poliedro debe tener por lo menos dos caras con el mínimo número de aristas.

i) Problema de ingenio. En un tablero de 3 x 3 se encuentran dos caballos blancos y dos caballos negros. La tarea es hace que los dos caballos blancos intercambien lugares con los caballos negros en tan pocas movidas como sea posible. Los caballos deben moverse del mismo modo que en el juego del ajedrez.

j) Hay seis fósforos en una mesa y la tarea es construir cuatro triángulos equiláteros donde la longitud de cada lado es igual a la longitud de un fósforo.

k) Encontrar valores enteros para x, y, z y t, tal que

x + y + z + t = 711, y
x * y * z * t = 711

l) Problema de ingenio. Sea un tablero de ajedrez con una dimensión de 2^m (consiste de 2^{2m} cuadrados) con un agujero, es decir, un cuadrado arbitrario que fue eliminado. Tenemos fichas con forma de L de tres cuadrados y la tarea es cubrir el tablero con esas fichas.

m) Minimizar f(x), donde $x \in D \subseteq R^n$.

n) Supóngase que se quiere construir una ruta entre la ciudad A y la ciudad B, pero estas ciudades están separadas por un rio. Nos gustaría minimizar la longitud de la ruta entre estas ciudades. El puente debe construirse perpendicular a las márgenes del rio.

ñ) Sean cuatro ciudades localizadas en los vértices de un cuadrado. La tarea es diseñar una red de rutas tal que (1) cada ciudad está conectada con cualquier otra ciudad y (2) la longitud total de las rutas es mínima. Las rutas pueden tener intersecciones.

o) Problema de Steiner. Dados n puntos en el plano, diseñar una red de conexiones de modo que la longitud total de toda la red sea tan corta como sea posible.

p) Problema de la moneda falsa.

q) Misioneros y caníbales: Tres misioneros y tres caníbales llegan a un río. En su orilla hay un bote que puede ser usado por una o dos personas. El objetivo es pasar a misioneros y caníbales a la orilla opuesta de tal modo que el número de caníbales nunca supere al de misioneros en ninguna orilla.

r) **8 puzzle**.

s) Jarras de 4 y 3 litros. Se tienen dos jarras de 4 y 3 litros y se desea dejar en esta última un litro de agua. Para eso se puede llenar o vaciar totalmente una jarra y trasvasar el contenido de una en la otra.

t) Criptoaritmética: El problema consiste en asignar a cada letra de un cierto conjunto, un único valor numérico entre 0 y 9 tal que la expresión:

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

respete la suma aritmética.

u) Problema de Bloques. Dada la siguiente configuración de bloques:

C
A B

llegar al estado

A
B
C

mediante las siguientes acciones:

Poner un bloque X sobre otro bloque Y si no hay nada sobre X ni sobre Y
Poner un bloque sobre la mesa.

v) Demostración de teoremas. Demostrar que Marco está muerto a partir del siguiente conjunto de axiomas:
1. Marco era un hombre.
2. Marco habitaba en Pompeya.
3. Marco nació en el año 40 d.C.
4. Todos los hombres son mortales.
5. Todos los habitantes de Pompeya murieron cuando el volcán hizo erupción en el año 79 d.C.
6. Ningún mortal vive más de 150 años.
7. La fecha actual es 1999 d.C.

w) Sea el siguiente juego. Dos jugadores tienen frente a ellos una sola pila de 7 monedas. El primer jugador divide la pila original en dos pilas que deben ser desiguales. Cada jugador alternativamente divide una pila de monedas cuando es su turno. El juego continúa hasta que todas las pilas tienen solo una o dos monedas, en cuyo punto se hace imposible continuar el juego. El jugador que sea el primero en no poder jugar pierde.

x) Problema de construir crucigramas. Colocar palabras en una grilla con cuadrados horizontales y verticales que se intersecan. Suponga que posee una lista de palabras (esto es, un diccionario) y que la tarea es llenar los cuadrados usando cualquier subconjunto de esta lista.

y) Dos archivos ordenados que contienen **n** y **m** registros respectivamente, pueden ser mezclados para obtener otro archivo ordenado realizando aproximadamente **n+m** operaciones. Cuando la operación se realiza entre más de dos archivos las operaciones se realizan de a pares. El problema es dado un grupo de cinco archivos de 20, 30, 10, 5 y 30 registros, encontrar una secuencia de aplicaciones de la operación de mezclado que minimice el número total de operaciones.

z) Analice el problema de generar sentencias bien formadas y de analizarlas utilizando una gramática de clase 2.

aa) Dada una situación donde Juan afirma ser descendiente directo de Newton y contando con el árbol genealógico de ambos. El problema es probar que Newton es antecesor de Juan.

bb) Problema de la cadena. Existen varios trozos de cadena que deben ser reconfigurados en una nueva organización. Los operadores pueden abrir un eslabón y cerrar un eslabón. En la forma estándar del problema, el estado inicial contiene cuatro cadenas, cada una con tres eslabones. El objetivo consiste en lograr una sola cadena de 12 eslabones en un círculo con la aplicación del menor número de operadores posible.

cc) Resolución de ecuaciones algebraicas (por ejemplo, “resolver $x^2y^3 = 3 - xy$ para x”).

dd) Encontrar un camino en el plano con obstáculos rectangulares.

ee) Resolver problemas de laberintos.

ff) Supóngase que está diseñando un compilador para una máquina con un solo registro y las siguientes instrucciones:

UNO	Poner en el registro el valor 1
DOBLE	Duplicar el contenido del registro
SUMAR_UNO	Sumar uno al contenido del registro
RESTAR_UNO	Restar uno al contenido del registro

Estas instrucciones solo pueden ejecutarse cuando el contenido del registro no es divisible por 3. Cuando esto ocurre, la única instrucción disponible es:

DIVIDIR	Dividir el valor del registro por 3

El problema es cómo poner el número 7 en el registro. Puede suponerse que el registro está inicializado en 1.

Supóngase que los costos de los operadores sobre un registro que contiene el número n son:

UNO	1
DOBLE	n
SUMAR_UNO	1
RESTAR_UNO	1
DIVIDIR	2n/3

(esto es, cada operador tiene un costo equivalente a la diferencia que produce respecto del valor inicial del registro) y la función heurística $h(n) = |7 - n|$.

gg) Considere el problema de encontrar un camino en una grilla desde la posición S a la posición F. El agente puede moverse horizontal o verticalmente en la grilla, un cuadrado por vez. No se pueden hacer pasos sobre el área marcada. Sea la siguiente grilla:

		F					
		X	X	X	X	X	
		X				X	
		X		**S**		X	

hh) Considere un juego de cartas en el cual dos jugadores retiran alternativamente una carta desde uno de los extremos de una línea de ellas. Inicialmente la línea contiene cinco cartas cada una etiquetada con un 1 o un 0 como muestra la siguiente figura:

[1] [0] [1] [0] [1]

El jugador que remueve la última carta gana si esta está marcada con un 1. Pierde en caso contrario. El problema consiste en mostrar que el segundo jugador tiene estrategia ganadora (esto es, siempre puede ganar si hace las apropiadas elecciones)

ii) Considere una variante del juego anterior donde ahora las cartas están etiquetadas con algún dígito (i.e. un número entre 0 y 9) y donde hay un único jugador que retira sucesivamente cartas desde los extremos de la línea hasta que la misma quede vacía. El puntaje final corresponde a la suma de los números de las etiquetas de las cartas extraídas en las jugadas impares. El problema es encontrar una estrategia que maximice el puntaje.

jj) Suponga un lenguaje cuyo vocabulario es V= {ε, A, B, C} y con las siguientes reglas de formación:

$$S_0 \Rightarrow AS \;/\; BBS_1 \;/\; CAS_1$$
$$S_1 \Rightarrow AS_1 \;/\; BBS_1 \;/\; ACS_1 \;/\; \varepsilon$$

A partir de estos datos se desea determinar cual es la palabra palíndromo de longitud 6 construible a partir del no terminal S_0 teniendo en cuenta la siguiente función de costo C(vi | vj):

vi \ vj	ε	A	B	C
ε	----	1/4	1/4	1/4
A	1/2	1	3/4	1/4
B	7/8	1/2	7/8	1/4
C	3/4	7/8	1/2	1/4

kk) Considere la siguiente variación del juego de cartas, jugado por un solo jugador. En cada movida el jugador retira un elemento de alguno de los dos extremos de una lista determinada de números. Su costo es igual a la diferencia entre el número más alto y el más bajo retirado en las movidas impares.

ll) Problema típico de scheduling (aunque muy simple, la mayoría no son tan sencillos). Suponga que tiene un conjunto de tareas, cada una con una duración especificada. Existen restricciones de precedencia entre las mismas, de modo que si la tarea A precede a la tarea B, entonces:

inicio A + duración A $\leq$ inicio B

Tarea	Duración	Precede
A	3	B,C
B	2	D
C	4	D
D	2	

mm) Considere la tarea de hacer un cronograma para los exámenes finales de n materias en un turno de exámenes de nuestro departamento. Para cada examen, se conoce la lista de los estudiantes que lo tomarán y la duración del examen. Puede programarse más de un examen en el mismo período de tiempo, pero existen dos restricciones: ningún estudiante puede realizar dos exámenes al mismo tiempo y pueden programarse cuanto más n exámenes en un período determinado (solo hay n aulas disponibles). La tarea es entonces producir un cronograma de exámenes que satisfaga las restricciones.

nn) Sea el problema de descomponer el número 6 en una cadena de igual cantidad de unos (1's) según las siguientes reglas:

$6 \Rightarrow 3, 3$	$6 \Rightarrow 3, 2, 1$	$4 \Rightarrow 2, 2$
$4 \Rightarrow 3, 1$	$3 \Rightarrow 1, 2$	$2 \Rightarrow 1, 1$

Suponga que costo de aplicación de una regla de descomposición es igual a la cantidad de números resultantes.

oo) Consideremos un puzzle consistente en una vaina con fichas, con la siguiente configuración inicial:

N	N	N	B	B	B	V

Hay tres fichas negras (N), tres blancas (B) y una casilla vacía (V). El puzzle tiene los siguientes movimientos:

Una ficha puede moverse a una casilla vacía adjunta, con costo unidad.
Una ficha puede saltar sobre otras dos como máximo hasta una celda vacía, con costo igual al número de fichas saltadas.

El objetivo del problema es llegar a tener todas las fichas blancas a la izquierda de todas las fichas negras (sin importar la posición de la ficha vacía).

pp) Problema del Cuadrado Latino. Consiste en llenar un cuadrado de 3 x 3 con un elemento del conjunto {1,2,3} de forma tal que en cada fila y columna no haya elementos repetidos.

qq) Se debe ejecutar un conjunto de n procesos $\{J_k\}_{1 \le k \le n}$ tal que cada uno tiene estimado un tiempo de ejecución $a_k \geq 0$ y un tiempo de preparación s_{ik} dependiendo de cual sea el proceso Ji que lo precedió inmediatamente en la ejecución (en nuestro caso $s_{0k} = 0$ para todo J_k). El costo de ejecución de un proceso es una función lineal del tiempo total de procesamiento, i.e. $G(J_k)=c_k *(a_k+ s_{ik})$. El procesador sólo puede ejecutar un sólo trabajo a la vez. El objetivo es minimizar el costo de ejecución.

trabajo	a_k	c_k	s_{k1}	s_{k2}	s_{k3}	s_{k4}
1	1	1	-------------	1	1	10
2	4	1	1	-------------	3	10
3	3	1	5	5	-------------	10
4	10	1	10	6	8	-------------

rr) Dado el siguiente problema consistente en partir de una configuración [0X0X0X0X0X] pasar a otra [0000XXXX], moviendo de a dos letras adyacentes a la vez (cualquier par) colocándolo en algunos de los extremos y eliminando el hueco dejado (si lo hubiera).

ss) Dado un tablero de tres por tres donde en cada casilla se pide encontrar uno diferente de los primeros nueve decimales, se desea, a través de permutaciones de cualquier longitud de contenidos de casillas, obtener una configuración final del tablero tal que la suma de filas, columnas y diagonales dé como resultado 15. El costo de una permutación es uno más la distancia entre las posiciones permutadas (Así por ejemplo si las casillas cuyos contenidos se intercambia están en los extremos de la misma diagonal, el costo de la operación es 5 y si las casillas son adyacentes a través de una arista, el costo de la permutación es 2).

tt) Compresión de textos. Hay dos enfoques básicos para la compresión de textos: estadístico y con diccionario. La codificación con diccionario logra compresión reemplazando grupos de caracteres consecutivos (frases) por índices en algún diccionario. El diccionario es una lista de frases que se espera que ocurran frecuentemente. Los índices se eligen de modo que en promedio ocupen menos espacio que la frase a la que codifican, logrando así compresión.

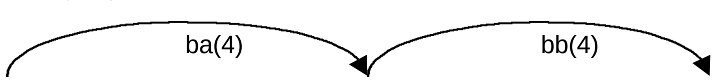
Formalmente, un diccionario D = (M, C) es un conjunto finito de frases M y una función C que aplica M en un conjunto de códigos. Las frases en M se generan a partir de un alfabeto de entrada A. Sin pérdida de generalidad, puede suponerse que los códigos de salida son strings sobre el alfabeto binario {0,1}.

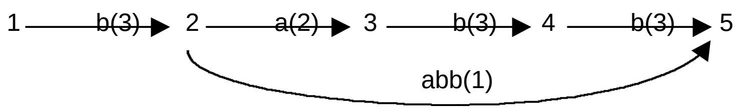
Una vez que se eligió un diccionario, hay más de un modo de elegir qué frases en el texto de entrada serán reemplazadas por índices del diccionario. La tarea de dividir el texto en frases para codificar se llama parsing. El enfoque más práctico es el parsing goloso, donde en cada paso el codificador busca el string más largo $m \in M$ que aparea los siguientes caracteres en el texto y utiliza C(m) para codificarlos. Por ejemplo, si M = {a, b, ba, bb, abb} y C(a) = 00, C(b) = 010, C(ba) = 0110, C(bb) = 0111 y C(abbb) = 1, entonces “babb” se codifica en 8 bits como C(ba).C(bb) = 0110.0111.

Desgraciadamente, el parsing goloso no es necesariamente óptimo. Por ejemplo, el string “babb” podría haber sido codificado en solo 4 bits como C(b).C(abb). Determinar un parsing óptimo puede ser difícil en la práctica, porque no hay ningún límite en cuánto más adelante el codificador tiene que mirar.

La tarea de parsing óptimo puede transformarse en un problema de camino más corto que entonces puede resolverse mediante los algoritmos existentes. La transformación se realiza para un string x[1..n] como sigue. Se construye un grafo que contiene n + 1 nodos, numerados de 1 a n + 1. Para todo par de nodos i y j, se coloca un arco dirigido entre ellos si el subsintagma x[i..j - 1] está en el diccionario. El arco recibe un peso igual a la longitud del código C(x[i..j-1]. El camino más corto desde el nodo 1 al nodo n + 1 representa entonces la secuencia parseada óptima para x.

Por ejemplo

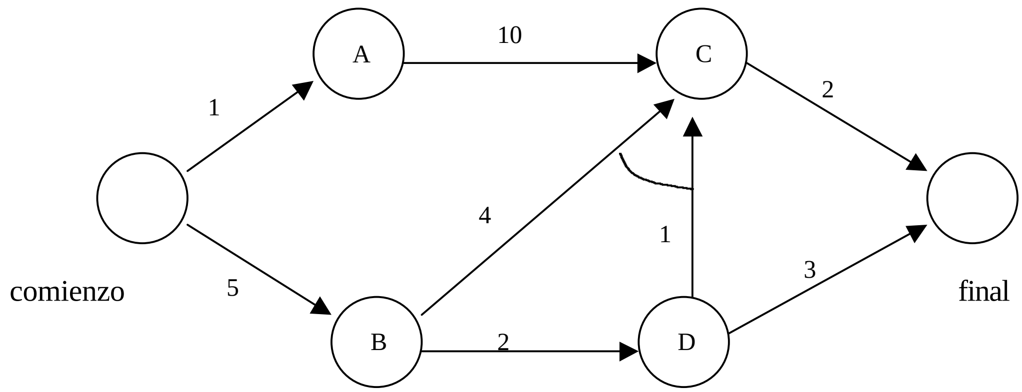




m	C(m)	L(m)
a	00	2
b	010	3
ba	0110	4
bb	0111	4
abb	1	1

L(m): longitud del código del string m

Resolver el problema de obtener la codificación óptima para el string “abbbaba”.



uu) La planificación de un proyecto consiste de una colección de trabajos que deben ser realizados y las interdependencias entre ellos (por ejemplo el resultado de un trabajo es necesario para el inicio de otro). La típica representación consiste en un grafo AND-OR.
En general con un proyecto se busca obtener un resultado final partiendo de ciertas condiciones iniciales (materias primas, datos originales, etc.) y como en la mayoría de los casos suele haber formas alternativas de lograr los mismos resultados. El problema de optimización en este tipo de planificaciones consiste en encontrar el camino crítico, esto es, la secuencia de tareas que condicionan la duración total del proyecto. Este problema puede ser convertido en uno de encontrar el camino más corto en el grafo AND-OR donde los nodos están etiquetados por el tiempo necesario para que los resultados de una etapa puedan ser aplicados a la siguiente y donde un arco OR indica que una tarea tiene proveedores alternativos y un nodo AND corresponderá a proveedores concurrentes. Así el tiempo de iniciar (costo) una tarea será el mínimo del tiempo de completamiento de las tareas proveedoras alternativas y el máximo del tiempo de terminación de las tareas proveedoras concurrentes.

Sea el siguiente grafo y los valores heurísticos para cada nodo.

h(A)=10, h(B)=2, h(C)=2, h(D)=1

vv) Cuadrado mágico. Un cuadrado mágico de 3x3 consiste en una tabla de 3x3 que tiene en cada posición un dígito (1,2,3,...,9) sin repetirse ninguno, y de forma tal que la suma de cada fila, cada diagonal y cada columna coinciden. Un ejemplo de un cuadrado mágico es el que sigue:

4	9	2
3	5	7
8	1	6

En este ejemplo, cada fila, columna y diagonal suman 15.

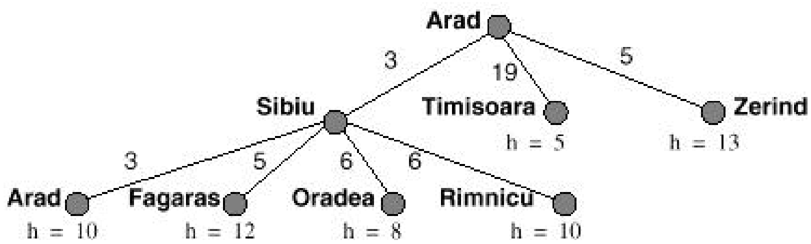
Problema: Dado un número N y un número r, enunciar un cuadrado mágico de NxN tal que sus filas, columnas y diagonales suman r.

ww. Problema de rutas áreas. Una línea aérea debe diseñar un conjunto de rutas áreas que les permita optimizar algunas variables que tienen que ver con su costo operativo y la conveniencia para la aerolínea y los pasajeros. Si miramos un mapa de las rutas aéreas de una aerolínea serán evidentes algunas “cabeceras” (hubs). Si un par de estas cabeceras es el origen y destino de un pasajero esto resulta conveniente porque es probable que haya muchos vuelos diarios entre las dos ciudades. En cambio, si el destino o el origen es otro, es probable que no exista un vuelo directo. El pasajero entonces deberá viajar hasta alguna cabecera y luego tomar otros vuelos para llegar a su destino. Para las líneas áreas redes modulares de rutas son más fáciles de mantener y coordinar, ya que solo muchos de los servicios esenciales pueden concentrarse en las cabeceras en lugar de distribuirse en todos los aeropuertos. Esta solución es ideal desde el punto de vista de la línea aárea, pero no tan conveniente para el pasajero. Sin embargo, al reducir sus costos de operación, la aerolínea está en mejores condiciones de ofrecer boletos más baratos.

xx. Juego Buscaminas (Windows)

II- Ejercicios

1) Determine para cada problema las características relevantes para su resolución: clases de problemas a las que pertenece, número de estados iniciales, número de estados finales, número de soluciones posibles, mérito de las soluciones, restricciones duras y blandas, descomponibilidad, etc.
2) Determine para cada problema cuál o cuáles serían las representaciones más adecuadas y justifique.
3) Represente cada problema según el o los enfoques de representación considerados más convenientes.
4) Determine para cada problema qué enfoques o algoritmos de búsqueda serían apropiados. Justifique en cada caso.
5) De acuerdo con el análisis del problema 4, resuelva los problemas con los métodos indicados.
6) El siguiente diagrama muestra un árbol de búsqueda parcialmente expandido. Cada arco está rotulado con el costo correspondiente del paso, y las hojas están rotuladas con el valor h.

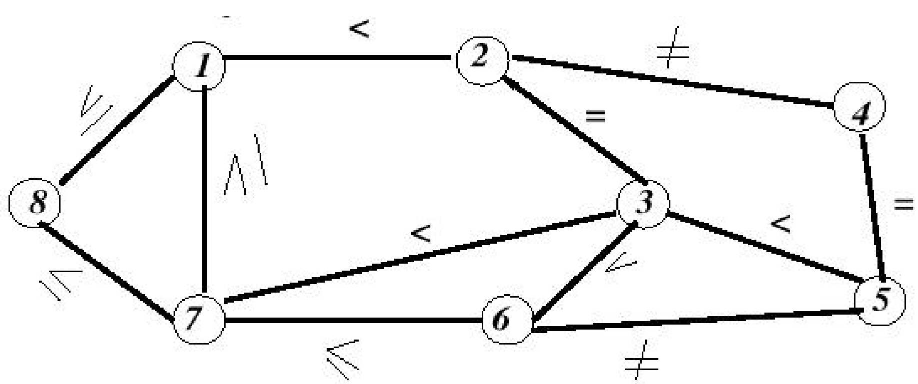


i. ¿Qué nodo expandirá una búsqueda golosa?
ii. ¿Qué nodo expandirá una búsqueda de costo uniforme?
iii. ¿Qué nodo expandirá una búsqueda por A*?
En cada caso justifique brevemente su respuesta.
7. En el siguiente cuadro marque cuáles métodos de búsqueda poseen las propiedades indicadas.

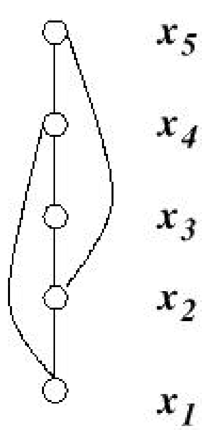
	Completo en espacio finito	Completo en espacio infinito	Admisible	Heurístico
Best-first				
A*				
SMA*				
Hill climbing				
Hill climbing con reinicio aleatorio				
Depth-first con profundidad iterativa				
IDA*				
AO*				

8) a. Supóngase que se fija una cota inferior negativa c sobre el costo de cualquier paso dado, es decir, se permiten costos negativos, pero el costo de un paso no puede ser menor que c. ¿Permite esto que la búsqueda de costo uniforme evite buscar en todo el árbol?
b. Supóngase que existe un conjunto de operadores que forman un ciclo, de modo que ejecutando el conjunto en algún orden no resulta en ningún cambio de estado. Si todos estos operadores tienen costo negativo, qué implica esto de la conducta óptima de un agente en un entorno de este tipo?
c. ¿Puede pensar en un dominio real en el que los costos de los pasos son tales que causan ciclos?
9) Suponga que utilizamos un algoritmo de búsqueda goloso con h(n) = -g(n). ¿Qué tipo de búsqueda emulará la búsqueda golosa?
10) A veces no existe una buena función de evaluación para un problema, pero hay un buen método de comparación: un modo de decir si un nodo es mejor que otro, sin asignar valores numéricos a ninguno. Muestre que esto es suficiente para hacer búsqueda best-first. ¿Qué propiedades de la búsqueda best-first abandonamos si solo tenemos un método de comparación?

11) Suponga que una regla **R** de un sistema de producción conmutativo se aplica al estado **I** produciendo el estado **F**. Pruebe que si **R** tiene inversa, el conjunto de reglas aplicables a **F** es el mismo que las aplicables a **I**.
12) Una función de evaluación *f* se dice que preserva orden si para cualesquiera dos caminos alternativos *P1* y *P2* que van desde el nodo inicial *s* y un nodo actual *n*, y cualquier extensión de esos caminos por otro *P3* sucede:
si f(P1) ³ f(P2) entonces f(P1º P3) ³ f(P2º P3)
Pruebe que la función *f* de A* cumple esta propiedad.
13) Reformule AO* para que sea ejecutado en grafos OR con el objetivo de encontrar el camino de costo (suma) inferior desde s a un conjunto de nodos meta, dado que h es optimista. ¿Cómo diferirá de A*? Analice las ventajas del modo en que cada algoritmo maneja los nodos que son generados y que ya están en Cerrados.
14) Supóngase que tenemos una función heurística que solo es aproximadamente admisible. Es decir, existe algún e > 0 tal que para todo nodo n en el árbol, tenemos h(n) ≤ h*(n) + e. (Nótese que en este caso es posible que h(n) para un nodo meta n sea mayor que cero). Sea C* el costo del nodo meta óptimo n*.
i. Demuestre que si A* retorna un nodo meta n_g, entonces el costo $g(n_g)$ es cuanto más C* + e. Suponga que h es monotónica.
ii. ¿Cómo debiéramos cambiar el algoritmo A* para encontrar la meta óptima n* en este caso? Supóngase que el valor de e está dado. (Pista: Agregar una segunda fase al algoritmo que trabaje sobre el resultado del punto anterior). Explique por qué su algoritmo es correcto.
15) Para cada uno de los problemas de la primera parte intente encontrar 3 heurísticas (no nulas). Analice las propiedades de cada una (admisibilidad, monotonía, existencia de máximos/mínimos locales, etc.) y compare sus ventajas y desventajas.
16) Busque un problema real que pueda formularse como CSP y hágalo (en una primera aproximación por supuesto; no sobrepase una página en su formulación).
17) Sea la siguiente red de restricciones.



El dominio de cada variable es {1..4}. Encuentre una red equivalente arco consistente. Determine si es mínima. Determine si es “libre de backtracking”.
18) Sea G un grafo de restricciones como el siguiente:



y un orden $d_1 = x_1, x_2, x_3, x_4, x_5$. Determine si las siguientes oraciones son verdaderas o falsas y justifique su respuesta:

i. El backjumping basado en el grafo se comportará exactamente como backtracking.
ii. El backjumping de Gaschnig se comportará como backtracking.
iii. El backjumping dirigido por conflictos y el backjumping de Gaschnig son idénticos en G.
iv. El aprendizaje basado en grafos sobre G y con orden d1 nunca registrará restricciones de un tamaño mayor que 2.
v. Si ocurre una hoja sin salida en x_5, ¿cuál es el conjunto de ancestros inducidos $I_5(\{x_5\})$?
vi. Si ocurre una hoja sin salida en x_5 y otra sin salida interna en x_4, ¿cuál es el conjunto de conflicto registrado por el aprendizaje basado en grafos?

b) ¿Cómo habría respondido las preguntas de arriba si se omitiese el arco (x_2, x_3)?

19) Sea una restricción entre tres variables X_i, X_j y X_k, con dominios D_i, D_j, y D_k respectivamente.

a) Describa formalmente un método simple para propagar esta restricción con el fin de reducir el espacio de valores D_i.

(b) Muestre cómo podemos extender el algoritmo básico de propagación de restricciones para utilizar información acerca de pares de países para reducir el conjunto de colores posibles de un tercer país. Más precisamente, si el país i tiene como vecinos a j y k, muestre cómo podemos eliminar del conjunto D_i los colores que no son consistentes con los conjuntos D_j y D_k (Solo se debe especificar cómo puede utilizarse la información de D_j y D_k para eliminar los valores de D_i. Recuerde que hay que diseñar un algoritmo que utilice la información de D_j y D_k **simultáneamente** para eliminar valores de D_i. ¿El algoritmo extendido elimina muchos coloreos posibles en comparación con el algoritmo estándar? De un ejemplo o muestre por qué no.

(c) En el problema anterior ayudaría extender su algoritmo para utilizar 3 países vecinos en lugar de 2? ¿Y 4 países o más? En particular, cómo depende el tiempo de corrida de su algoritmo del número m de vecinos considerados simultáneamente?

III- Ejercicios prácticos a entregar

1) Considere el 8 puzzle con nodo inicial s = [2 8 3, 1 6 4, 7 _ 5] y nodo meta γ = [1 2 3, 8 _ 4, 7 6 5]. Ejecute el algoritmo A* utilizando las siguientes heurísticas:
a. h = 0
b. h_1 = número de fichas mal colocadas
c. h_2 = suma de distancias de Manhattan
d. h_3 = 3 * seq(n) donde seq(n) cuenta 1 para una ficha central y 2 para cada ficha en el perímetro del tablero que no esté seguida (en el orden definido por las agujas del reloj) por su sucesor apropiado.
En cada uno de los casos precedentes cuente el número de nodos expandidos y muestre el árbol de búsqueda explícito en la terminación.
e. Evaluar la performance de las estrategias de trepada de colina utilizando las heurísticas en (a), (b) y (c).
f. Compare la performance de A*, trepada de colina, IDA* y SMA*.

2) Un lenguaje L usa un vocabulario compuesto por 4 símbolos V={ε, A, B, C} donde ε es el símbolo vacío. La probabilidad de que un carácter v_i esté seguido por v_j se representa en la siguiente matriz:

P(vj/vi)=

	vj			
vi	ε	A	B	C
ε	¼	¼	¼	¼
A	½	0	¼	¼
B	1/8	½	1/8	¼
C	1/4	1/8	½	1/8

Utilizando costo uniforme,
a. Encuentre la cadena más probable de longitud 5 que comience con ε.
b. Encuentre la cadena más probable de longitud 5 que comience y termine con ε.

Al transmitir mensajes de L algunos caracteres pueden corromperse por ruido y ser confundidos por otros. La probabilidad de que un carácter vj pueda ser interpretado como otro vk se representa en la siguiente matriz:

$P(v_k/v_j)$=

	v_k			
v_j	ε	A	B	C
ε	0.9	0.1	0	0
A	0.1	0.8	0.1	0
B	0	0.1	0.8	0.1
C	0	0.1	0.1	0.8

c. Usando costo uniforme encuentre el mensaje que más probablemente fue transmitido dado que la cadena ε ε B C A A ε ε fue recibida, y sabiendo que todos los mensajes deben comenzar y terminar con ε.
d. Repita las partes (a), (b), y (c) usando backtracking. Compare el número de nodos generados y el almacenamiento requerido por los dos algoritmos.
e. ¿Puede encontrar una función heurística razonable h a utilizar con A* sobre la parte c)?

3) Descripción de una clase empírica de problemas. Debe presentar un informe en el que se describan las siguientes cosas: nombre de la clase y breve descripción; breve historia de su estudio y automatización; subclases y relación con otras clases de problemas; complejidad del problema; formas de representarlo y algoritmos de búsqueda apropiados.

4) Cuadro de métodos de búsqueda. Debe realizar un cuadro que represente los diversos métodos de búsqueda estudiado. Los métodos deben estar agrupados en clases y a cada método se debe asociar el conjunto de propiedades (completitud, correctitud, optimalidad, etcétera) estudiados. Un eje del cuadro debe estar conformado por las variables de “diseño” (independientes) que caracterizan a los algoritmos (p. ej. tipo de espacio de búsqueda que recorren, informados vs. no informados, exhaustivos vs. no exhaustivos,

etcétera). Estas variables deben permitir agrupar en clases a los algoritmos. El otro eje del cuadro debe estar conformado por las propiedades asociadas a los métodos (variables dependientes). El cuadro debe ser lo más completo posible en los métodos y propiedades contenidos. También debe ser autocontenido, es decir, debe tener todas las aclaraciones necesarias (en la forma de llamadas) para que sea perfectamente comprensible.

5) Cuadro de métodos de abstracción (transformación de problemas). Debe realizar un cuadro que describa los métodos de abstracción estudiados y las propiedades de estos. Los criterios de diseño del cuadro son los mismos del punto 3).

6) Definir criterios para la selección de los algoritmos de búsqueda en función de características del problema. Estos criterios pueden estar planteados en un cuadro, en un conjunto de reglas, en un árbol de decisiones o en cualquier otro formato que le permita representar claramente una forma de seleccionar los algoritmos apropiados o convenientes para un problema (o clase de problemas) dado. El objetivo general es poder ofrecerle una "guía rápida" a una persona que pueda querer emplear los métodos de búsqueda de IA sobre cuándo usar cada método. Por supuesto, puede "reutilizar" el cuadro sobre algoritmos del punto 3. Debe ser explícito y dentro de lo posible justificar las cosas. Debe ser lo más completo posible respecto de la bibliografía pero también me gustaría que ponga un poco de pensamiento propio e independiente, tanto en lo que a crítica de la bibliografía se refiere como a intentar completar los "gaps" que encuentre en ella.

7) Desarrollo de una experiencia de comparación de algoritmos. Esta experiencia tiene cuatro etapas: a) diseño, b) realización, c) análisis y d) informe. En la etapa de diseño debe primero seleccionar los dos algoritmos, el problema y sus instancias, y debe definir cómo será la realización y análisis de la experiencia. En la etapa de realización debe implementar los algoritmos e instancias de problemas, realizar las corridas y recoger los datos obtenidos. En la etapa de análisis debe analizar los resultados obtenidos utilizando técnicas estadísticas apropiadas. En la etapa de informe debe realizar un informe escrito apropiado de la experiencia. Se debe explicitar clara y detalladamente cómo se realizaron las tres etapas anteriores y deben presentarse los datos obtenidos, el análisis estadístico hecho, su interpretación y conclusiones. Los dos algoritmos y el problema elegido deben tener sentido y ser razonables. Toda la experiencia debe hacerse siguiendo los lineamientos metodológicos contenidos en la bibliografía correspondiente.