**TEMA 3: BÚSQUEDA INFORMADA (O HEURÍSTICA)**

1.- Funciones Heurísticas. Búsqueda informada.

2.- Métodos voraces.

2.- Algoritmo ***A\****.

3.- Propiedades de las heurísticas: comparación de heurísticas, admisibilidad, consistencia, monotonía,

4.- Relajación y diseño de heurísticas.

5.- Búsqueda heurística con memoria limitada: algoritmos ***IDA\**** y ***SMA\****

6.- Mejora iterativa y búsqueda local.

En los métodos de búsqueda no informada del tema anterior, la búsqueda se realizaba explorando sistemáticamente los nodos del grafo, a veces teniendo en cuenta lo ya gastado en la exploración de la parte del camino recorrido (coste uniforme), pero sin utilizar ninguna información disponible (que en ocasiones puede existir) sobre preferencia de unas vías sobre otras o sobre proximidad de las metas en las partes de los caminos aún por recorrer.

Cuando se dispone de alguna clase de información en este segundo sentido se habla de ***heurística***s y los métodos correspondientes son los ***métodos heurísticos.***

En ellos, básicamente lo que se hace es utilizar dicha información para elegir el desarrollo de unos estados frente a otros, tomando para desarrollar de la lista de abiertos antes aquellos que aparenten estar más cerca de la meta.

EJEMPLOS

En el problema del 8-*puzzle*: el número de piezas por colocar o la suma de las distancias de cada pieza descolocada a su posición definitiva son indicadores de la “distancia” de cada estado hasta la solución.

En problemas de laberintos o de mapas: distancias (aérea o en cuadrícula) desde cada estado a la meta.

**1. FUNCIONES HEURÍSTICAS. BÚSQUEDA HEURÍSTICA**

La información sobre proximidad a la meta consiste en una función numérica ***h*** del espacio de estados a ***[0, ∞ [ = R +U {0}***, que estime la “distancia” de cada estado al objetivo más próximo, valiendo **0** en las metas.

Esta estimación es información incierta (de no serlo, desaparecería el problema), y puede conducir por sí sola a engaños. Al usarla se sacrifica certeza para obtener economía, “podando” el árbol de búsqueda para centrarse en las ramas o direcciones más prometedoras.

Combinando adecuadamente la información heurística (sobre el futuro, incierta) con la del gasto realizado (sobre el pasado, conocida), se puede establecer métodos seguros y económicos, que combinan útilmente las ventajas respectivas de ambas informaciones.

**2. MÉTODOS VORACES**

**Búsqueda en *Escalada (o irrevocable, ingl. Hill Climbing)***

Esta búsqueda, rudimentaria, sin memoria, consiste en seguir siempre sólo el camino hacia el que, según lo apuntado por la función heurística, parezca mejor sucesor, mientras éste mejore al estado en que se esté:

1. Empezar con *ACTUAL* = lista formada por el estado inicial
2. Hasta que *ACTUAL* = meta o no haya cambios en *ACTUAL*, hacer:
3. Presentar la lista de *ACTUAL* si su estado final es una meta y parar
4. Tomar los sucesores de *ACTUAL* y usar *h* para puntuar cada uno de ellos.
5. Si uno de los sucesores tiene mejor (o sea, menor valor de *h*) puntuación que *ACTUAL*, hacer *ACTUAL* igual a la lista anterior aumentada en el sucesor.
6. Presentar FALLO y terminar

CUESTIONES.

1. Comprobar mostrando ejemplos adecuado que esta búsqueda no explora más que parte de un único camino de los existentes, que puede no seguir dicho camino hasta el final, que puede entrar en ciclos infinitos, que no explora sistemáticamente el espacio de estados y, en suma, que no es completa ni óptima, aunque la heurística sea totalmente fidedigna.
2. Estudiar la complejidad en tiempo y espacio de este “método”
3. Relacionar la primera deficiencia de (a) con la existencia de mínimos locales de la función *h* en el grafo de búsqueda y pensar posibles modificaciones del algoritmo para tratar de subsanarla.

**Búsqueda *Primero el mejor***

Como la anterior, usa la heurística para elegir siempre al nodo sucesor aparentemente mejor situado, pero usa memoria (manteniendo abiertas las posibilidades alternativas mediante una lista de ABIERTOS) y así varios caminos posibles, no sólo uno sin alternativas.

Funciona de modo análogo al de la búsqueda (desinformada) optimal o de coste uniforme gestionando la lista ABIERTOS como una cola de prioridad, pero usando sólo la función heurística *h* para ordenarla y priorizar, examinando antes los nodos que tengan valor heurístico más bajo (que, según la información aportada por *h* estén “más cerca” del objetivo).

Por supuesto, la obtención de resultado y la calidad del algoritmo, dependerá de lo fidedigna que sea la función heurística.

1. Empezar con *ABIERTOS* = lista formada por el estado inicial
2. Mientras *ABIERTOS* no esté vacío hacer:
3. Quitar de *ABIERTOS* la lista con el mejor nodo terminal y ponerla en *ACTUAL*
4. Si el nodo terminal de *ACTUAL* es meta, presentar su lista y terminar.
5. En otro caso, tomar los sucesores de dicho nodo, usar *h* para puntuar cada uno de ellos e incorporar las listas resultantes de ampliar *ACTUAL* con dichos nodos a la lista *ABIERTOS* en su orden.

3. Presentar FALLO y terminar.

Si se dispone de una buena función heurística, la búsqueda *primero el mejor* puede lograr en la práctica sustanciales descensos en el coste computacional de su ejecución, si bien no queda garantizado que el camino obtenido sea óptimo. Hay que tener además en cuenta el coste suplementario que la evaluación de *h* en cada nodo suponga.

CUESTIONES:

1. Comprobar mostrando ejemplos adecuados que esta búsqueda puede no ser óptima.

**2. ALGORITMO A\***

La idea en la que se basa la búsqueda **A\*** consiste en intentar combinar las ventajas de la búsqueda desinformada de coste uniforme (completitud y optimalidad) con las de la búsqueda heurística primero el mejor (eficiencia, por la poda del árbol de búsqueda).

Ello se consigue priorizando en la lista de abiertos aquellos caminos en que menor resulte una **combinación** de los valores del **gasto hecho** en la parte del camino recorrido (suma de los costes de sus aristas etapa) con la **distancia** restante hasta la meta según la información dada por *h* en el trozo de camino por recorrer:

Para hacerlo, en el paso 2.3 del algoritmo anterior se substituye la función ***h***por la ***f*** dada por

***f(nodo)=g(nodo)+h(nodo)***

lo que ordena a los nodos de *ABIERTOS* según la media aritmética de los valores de ***g*** y de ***h***  o por alguna otra combinación adecuada que agregue ambos valores.

Si la heurística es adecuada, con este método se consigue un algoritmo completo, óptimo y con un coste computacional menor, equivalente en la práctica una disminución del factor de ramificación.

**3. PROPIEDADES DE *A\** Y DE LAS HEURÍSTICAS**

Si se denotan con ***h\**** a la heurística ideal, que daría la información perfecta sobre el coste óptimo, por el mejor camino, desde cada nodo hasta la meta más cercana, y con ***f\*(n)=g(n)+h\*(n)*** al coste total de la mejor solución que pase por n, dada una posible función heurística ***h***, puede ocurrir:

a) Que ***h(n)=0*** para todo ***n***, es decir, que la heurística no dé información. En este caso, ***A\**** se convierte en la búsqueda optimal.

b) Que ***h(n)=h\*(n)*** para cada ***n***, entonces no habría problema, pues en cada paso se sabría qué dirección hay que seguir para llegar a la meta más cercana del inicio por el mejor camino.

c) Que ***h(n) > h\*(n)*** para algún ***n.*** Entonces ***A\**** puede resultar no óptimo.

d) Que ***h(n) ≤ h\*(n)*** para cada ***n.*** Entonces ***A\**** termina y resulta ser completo, óptimo y con coste computacional tanto más reducido cuanto mayor sea ***h*** (es decir, cuanto más parecida sea ***h*** a ***h\****), oscilando entre la búsqueda – a ciegas- de coste uniforme del caso (a) y la ausencia se búsqueda del caso (b).

Las heurísticas consideradas en (d) se llaman ***admisibles*.**

Una heurística ***h*** es ***más informativa*** que otra ***h´*** si cumplen ***0 ≤ h´(n) ≤ h(n) ≤ h(n)\**** para cada ***n***. Cuanto más informativa sea una heurística, más económica resultará la búsqueda mediante ella con ***A\****, es decir, menos nodos habrá que examinar para encontrar la solución.

Una heurística es ***consistente c***uando para cada par de nodos adyacentes ***n*** y ***n´*** con el segundo sucesor del primero cumple que ***h(n)-h(n´) ≤ C(n,n´).*** Es ***monótona***  si se cumple que siempre que ***n´*** sea un sucesor de ***n*** se tendrá que ***f(n´) ≥ f(n)*** . Cuando una heurística es consistente, entonces también es monótona y se cumple que la primera vez que un nodo sea escogido de la lista *ABIERTOS* para ser examinado, se habrá llegado desde el inicial hasta él por el camino más corto posible (es decir, que en la aplicación de ***A\**** no habrá que “rectificar” ningún camino)

Nota: las pruebas de estas afirmaciones, véase Nilsson 9.2.1-9.2.4 o Poole-Mackworth 3

CUESTIONES:

1. Mostrar con ejemplos la necesidad de las condiciones exigidas para garantizar las conclusiones correspondientes.
2. Describir funciones numéricas del espacio de estados que, usadas en vez de la función *f=g + h* en *A\** ocasiones las búsquedas en anchura, profundidad, de coste uniforme y primero el mejor.
3. Dadas dos heurísticas admisibles *h1*y *h2*, describir para qué valores de los coeficientes α y β resultaría admisible la función αh1 + βh2 como heurística.
4. Compruebe que toda heurística consistente es admisible, pero no toda admisible es consistente.
5. ¿Tiene sentido considerar búsquedas heurísticas bidireccionales?
6. ¿Puede plantearse algún tipo de búsqueda en profundidad iterativa heurística?

**Ejercicios 3.1**

1. Comprobar que la unión de una cantidad finita de conjuntos finitos es un conjunto finito.
2. Comprobar que el producto de una cantidad finita de conjuntos finitos es un conjunto finito.
3. Comprobar que la potencia finita de conjuntos finitos es un conjunto finito.
4. Comprobar que todo conjunto finito de números reales tiene un máximo y un mínimo.
5. Comprobar que en un problema en el que cada estado tenga una cantidad finita de sucesores (que puede ser distinta para estados diferentes) y en el que haya una cota inferior positiva para el coste de las transiciones, se cumple que la cantidad de estados existente a distancia menor o igual que una dada es finita.
6. Comprobar que en un problema que cumpla las dos condiciones del anterior, en el que además exista alguna meta accesible desde el estado inicial, se cumple que hay alguna solución óptima.
7. Comprobar que en un problema que cumpla las tres condiciones del problema anterior y en el que se use una heurística admisible, la ejecución de A\* termina.
8. Encontrar un ejemplo de problema en el que A\* presente una solución que no sea óptima.
9. Encontrar un ejemplo de problema en el que A\*, ejecutándose con una heurística admisible, no encuentre el óptimo como primer camino a una de las soluciones más cercanas.
10. Comprobar que cuando una heurística es más informativa que otra, siendo ambas admisibles, todo estado examinado durante la ejecución de A\* por la primera lo será por la segunda. ¿Es necesaria la condición de admisibilidad para garantizar lo anterior?
11. Razonar si tiene sentido la realización de una búsqueda heurística bidireccional.

**APUNTES**

**Admisibilidad**

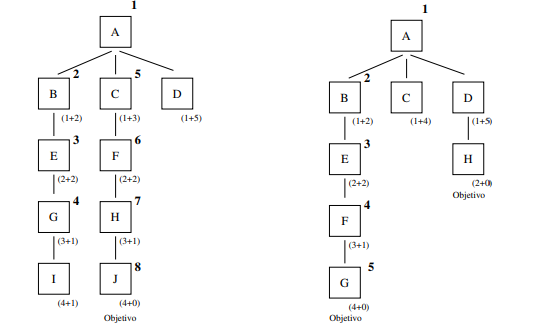
La propiedad clave que nos garantizará el hallar la solución óptima es la que denominaremos **admisibilidad**. Diremos que una función heurística h es admisible siempre que se cumpla que su valor en cada nodo sea menor o igual que el valor del coste real del camino que nos falta por recorrer hasta la solución:

∀n 0 ≤ h(n) ≤ h\*(n)

Esto quiere decir que la función heurística ha de ser un estimador optimista del coste que falta para llegar a la solución.

**Ejemplo**

En la figura podemos ver en este ejemplo dos grafos de búsqueda, uno con una función admisible y otra que no lo es. En este primer caso, la función heurística siempre es admisible, pero en la primera rama el coste es más pequeño que el real, por lo que pierde cierto tiempo explorándola (el efecto en profundidad que hemos comentado), pero al final el algoritmo pasa a la segunda rama hallando la solución. En el segundo caso el algoritmo acabaría al encontrar la solución en G, a pesar de que haya una solución con un coste más pequeño en H. Esto es así porque el coste estimado que da en el nodo D es superior al real y eso le relega en la cola de nodos abiertos.



**EXAMEN**

**T1. Razone si son admisibles y si son consistentes las heurísticas de la distancia de Hamming (o Manhattan) y la Euclídea (o Aérea) en problemas de laberintos o mapas.**

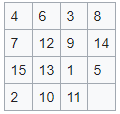
La distancia de Hamming viene definida por el número de elementos que distinguen a un elemento de otro. Por ejemplo, la distancia de Hamming entre “tener” y “reses” es de 3.

Por ejemplo, si se quiere resolver el 15-puzzle en la menor cantidad de pasos posible usando A\*, es necesario emplear una heurística admisible. Las dos heurísticas más comúnmente usadas son:

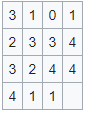
* **Distancia de Hamming**: el número de fichas que no están en su lugar.
* **Distancia Manhattan**: la suma de las distancias desde la posición actual de cada ficha hasta su posición original.

Queda claro que la **Distancia de Hamming** es admisible, dado que el número total de movimientos para ordenar las fichas correctamente es al menos el número de fichas que no están en su lugar (si cada ficha no está en su posición original deberá ser movida al menos una vez).

La **Distancia Manhattan** también será una heurística admisible porque cada ficha será movida al menos la cantidad de pasos entre ella misma y su posición original. Considere el siguiente rompecabezas:



la distribución de las distancias Manhattan para cada ficha quedaría así:



La distancia total de Manhattan para el puzle quedaría:

h(n) = 3 + 1 + 0 + 1 + 2 + 3 + 3 + 4 + 3 + 2 + 4 + 4 + 4 + 1 + 1 = 36

La distancia de Hamming para el puzle sería: 14

Todas las heurísticas consistentes son admisibles, sin embargo, no todas las heurísticas admisibles son consistentes.

La heurística euclídea consiste en la información del coste entre dos nodos mediante una única línea recta. Es admisible ya que la distancia aérea es siempre una estimación optimista de la distancia real.

**T1.**

**a) Defina “heurística admisible” y “heurística consistente”.**

Una heurística es admisible cuando se cumple la siguiente condición:

0 <= h(n) <= h\*(n)

es decir, se tiene que cumplir que la estimación del coste desde un nodo n hasta el nodo meta sea menor o igual al coste real.

Si se diese el caso en el que la estimación del coste fuese mayor al coste real entonces no se podría llegar a una solución óptima y, por tanto, la heurística no sería admisible.

Una heurística es consistente cuando se cumple la siguiente condición:

h(n)-h(n’) <= C (n,n’)

es decir, se tiene que cumplir que para un nodo n y un sucesor de n llamado n’, el coste para ir de n a su sucesor tiene que ser mayor obligatoriamente que la resta de las heurísticas de n menos n’.

**b) Explique su utilidad y diferencia.**

La utilidad de una heurística admisible es encontrar la solución óptima.

La utilidad de una heurística consistente sería decidir si la solución óptima encontrada es válida o no.

**c) Dadas dos heurísticas h1 y h2 admisibles y consistentes, se definen h3 = h1+h2 / 2 y h4 = max (h1, h2). Estudie si h3 es consistente, si h4 es admisible y si algunas de las cuatro pueden compararse entre sí por ser más o menos informadas.**

**Describa brevemente las características que sirven para clasificar los juegos e indique para qué tipo de juegos se ha aplicado el método de poda alfa-beta.**

Las características existentes para clasificar los juegos son: número de jugadores, información completa o incompleta, intervención del azar, existencia o no de coaliciones, turnos de juego y cantidad de jugadas posibles en cada momento para un jugador.

Para aplicar el método de poda alfa-beta se necesita que un juego cumpla las siguientes condiciones: existan dos jugadores que jueguen alternativamente, disponer de una información completa, es decir, sin que intervenga el azar y que cada turno de juego tenga una cantidad de jugadas posibles finitas.

**Condiciones suficientes del grafo de búsqueda y de la función heurística para garantizar la completitud y la optimalidad en los métodos Optimal (o de Coste Uniforme) y A\***