

BUSQUEDA HEURISTICA

Todos los algoritmos de fuerza bruta son poco eficientes debido a que son búsquedas a ciegas, es decir, no usan conocimiento sobre el dominio para guiarse en busca del siguiente nodo mejor situado.

La búsqueda heurística se basa en que la mayor parte de los **"espacios problemas"** proporcionan información adicional con un pequeño coste computacional. Esta información adicional nos ayuda a elegir los nodos que estén más cerca de la meta. Esta información se llama **Heurística**.

Se llama **Función de Evaluación** a una función **f** que relaciona cada nodo con un número real y que sirve para estimar el costo relativo de continuar la búsqueda desde ese nodo. Por lo general, **f(N)** se considera que es la distancia estimada que queda entre **N** y el **Nodo Meta**. Otras veces se dice que **f(N)** es la longitud estimada desde el **Nodo Raíz** hasta el **Nodo Meta**, pasando por el nodo **N**, etc... Esta función de evaluación se utilizará para decidir el orden en que se explorarán los nodos durante la búsqueda. Hay varios algoritmos de búsqueda que usan una de estas funciones de evaluación heurística. Entre otros veremos los siguientes:

- Escalar colinas
- Mejor primero
- A*

ESCALAR COLINAS

El método de escalar colinas es el más simple de los métodos heurísticos. El nombre viene de la analogía existente entre el espacio del problema y un plano bidimensional donde la función heurística define la elevación de cada punto y la meta es alcanzar el punto más alto del terreno.

Escalar colinas funciona efectuando movimientos en la dirección que tenga la subida más pronunciada localmente. Lo único que se memoriza es el estado en curso y el que originalmente era el nodo raíz. En cada ciclo se expande el nodo en curso y se aplica la función heurística a cada uno de sus sucesores y el que tenga el mejor valor se convierte en el nuevo estado en curso.

Es semejante a la búsqueda en profundidad, excepto que el método de escalar colinas ordena los posibles estados a elegir en cada paso de la búsqueda de acuerdo con alguna medida 1 función heurística de la distancia que queda desde cada uno de ellos hasta la meta. Cuanto mejor sea la medida heurística mejores resultados dará el método de escalar colinas con respecto al de profundidad.

BUSQUEDA EL MEJOR-PRIMERO

En la búsqueda "el mejor primero" el movimiento hacia adelante se hace desde el nodo mejor situado con respecto a la meta, sin tener en cuenta su situación en el árbol.

Los caminos que se encuentren utilizando este método de búsqueda serán más cortos que los que se encuentren buscando con otros métodos, ya que con esta estrategia se parte del nodo mejor situado.

Igual que con el método de escalar colinas este método requiere clasificar los nodos de acuerdo con su situación. En este caso, sin embargo, se clasifica toda la estructura de datos.

Por lo tanto, en cada paso del proceso de búsqueda se selecciona el nodo más prometedor de **todos** los que se han generado hasta el momento. Esto se consigue aplicando una función heurística apropiada a cada nodo generado expandiendo, a continuación, aquel que ha dado el mejor resultado al evaluar dicha función heurística. Si uno de los sucesores (hijo) del nodo expandido es la meta, la búsqueda termina. En caso contrario, estos sucesores se suman al conjunto de nodos generados hasta el momento y se repite el proceso de selección del nodo más prometedor

El algoritmo que se presenta a continuación realiza la búsqueda a través de un grafo dirigido en el cual cada nodo representa un punto, o estado, en el espacio del problema. Cada nodo es, entonces, una estructura que contiene, además de una descripción del estado del problema que representa, una indicación de lo prometedor que es respecto a los demás nodos, un enlace-padre que apunta al mejor nodo de los nodos-padre que cuelga, y por último, una lista de sus nodos sucesores (aquellos que se generan a partir de él). El enlace-padre permite recobrar el camino hasta la meta una vez se ha llegado a ella. La lista de sucesores permite, cuando se encuentra un camino mejor hasta un nodo ya generado, propagar esta mejora hacia sus sucesores. Para implementar este procedimiento se necesitan dos listas de nodos:

- **OPEN:** Contiene los nodos que se han generado y a los cuales se ha aplicado la función heurística pero todavía no han sido examinados (no se han generado sus sucesores). OPEN es, en realidad, una cola de prioridad en la que los elementos con mayor prioridad son aquellos con valor más prometedor para la función heurística.
- **CLOSED:** Contiene los nodos que ya han sido examinados. Es necesario mantener estos nodos en memoria ya que cuando se genera un nuevo nodo hay que comprobar si ya ha sido generado antes.

DESCRIPCION GENERAL DEL ALGORITMO "EL MEJOR PRIMERO"

Se comienza insertando el nodo **S** (nodo de comienzo) en la lista **OPEN**. A continuación se van procesando sucesivamente los nodos en **OPEN**: se comprueba si son nodos meta, y en caso de que no lo sean, se meten en la lista **CLOSED**, se introducen sus sucesores en **OPEN** y se actualiza la función heurística **f(N)** para cada nodo **N** procesado. Los nodos en **OPEN** son tratados según un orden que da la mayor prioridad al nodo con menor **f(N)**.

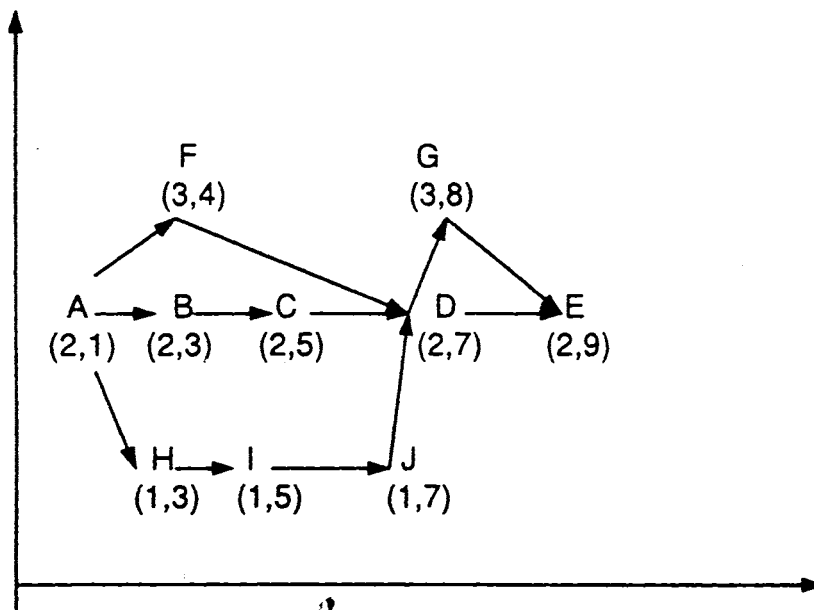
ALGORITMO

- 1.- Insertar el nodo de comienzo **S** en **OPEN**, evaluar **f(S)** y asociar este valor con **S**. Poner el enlace-padre a **NULL**.
- 2.- Si **OPEN** está vacío, devolver **FALLO** y parar.
- 3.- Elegir un nodo **N** de **OPEN** tal que **f(N) < f(M)** para cada **M** en **OPEN**, tal que **N** es un nodo meta algún nodo meta tiene ese valor mínimo para **f**.
- 4.- Poner **N** en **CLOSED** y quitar **N** de **OPEN**.
- 5.- Si **N** es un nodo meta devolver el camino desde **S** hasta **N**. Este camino se obtiene recorriendo **f** hacia atrás los enlaces-padre desde **N** hasta **S**. Luego parar.
- 6.- Para cada sucesor **J** de **N** que no está aún en **OPEN** o en **CLOSED**:
 - a.- Evaluar **f(J)** y asociarlo con **J**.
 - b.- Poner **J** en **OPEN**.
 - c.- Poner el enlace-padre de **J** apuntando hacia **N**.
- 7.- Para cada sucesor **J** que ya está en **OPEN**, reevaluar **f(J)** y comparar este valor con el anterior. Si el nuevo valor es menor, asociarlo con **J** y volver a colocar **J** en **OPEN**.
- 8.- Ir al paso 2

La eficiencia de este método de búsqueda depende de la función heurística **f** que se utilice.

EJEMPLO:

Resolver utilizando el método de **mejor-primero**:



SOLUCION

1.- $OPEN \leftarrow A$

$f(A) = 8$

$P(A) = \text{Null}$

Luego al final de esta primera etapa tenemos:

$OPEN = A$

$CLOSE = \text{NULL}$

2.- $A \neq \text{Meta}$ y además es el único nodo en OPEN, luego

$CLOSE \leftarrow A$

Generamos los sucesores de A

$B, F, H \notin OPEN \vee CLOSED$

$f(B) = 6, P(B) = A$

$f(F) = \sqrt{37}, P(F) = A$

$f(H) = \sqrt{37}, P(H) = A$

quedando entonces:

$$\text{OPEN} = B, F, H$$

$$\text{CLOSED} = A$$

3.- $f(B) < f(F)$. por lo tanto metemos B en CLOSED

$$\text{CLOSED} \leftarrow B$$

generamos los sucesores de B

$$C \notin \text{OPEN} \vee \text{CLOSED}$$

$$\text{OPEN} \leftarrow C$$

$$f(C) = 4, P(C) = B$$

quedando entonces:

$$\text{OPEN} = F, H, C$$

$$\text{CLOSED} = A, B$$

4.- $f(C) < f(F)$. por lo tanto metemos C en CLOSED

$$\text{CLOSED} \leftarrow C$$

generamos los sucesores de C

$$D \notin \text{OPEN} \vee \text{CLOSED}$$

$$f(D) = 2, P(D) = C$$

quedando entonces:

$$\text{OPEN} = F, H, D$$

$$\text{CLOSED} = A, B, C$$

5.- $f(D) < f(F)$. por lo tanto metemos D en CLOSED

$$\text{CLOSED} \leftarrow D$$

generamos los sucesores de D

$$E, G \notin \text{OPEN} \vee \text{CLOSED}$$

$$f(E) = 0, P(E) = D$$

$$f(G) = 6, P(G) = D$$

quedamos entonces:

OPEN = F, H, E, G

CLOSE = A, B, C, D

6.- E es la meta, luego devolverá el camino desde el origen hasta la meta.

METODO DE BUSQUEDA A*

Este método de búsqueda examina los nodos en un orden que da la mayor prioridad a los nodos que parecen pertenecer al camino más corto desde el nodo inicial hasta la meta. Es por tanto, similar al método "El Mejor Primero" pero siendo la diferencia fundamental la función heurística **f**, ya que ésta no es, simplemente, la estimación de la distancia que queda desde el nodo, en cuestión, hasta la meta, sino que se suma un factor **g**, que representa el coste del mejor camino encontrado hasta el momento, desde el nodo inicial hasta el nodo actual.

Así, la función heurística **f** utilizada en la búsqueda **A*** queda de la siguiente forma:

$$f = g + h$$

donde las funciones **g** y **h** reflejan respectivamente:

g = el coste del mejor camino desde el nodo inicial hasta el actual.

h = estimación del coste involucrado en llegar desde el nodo actual a la meta. En esta función **h** es donde se explota, realmente, el conocimiento que se tiene sobre el dominio del problema.

La función combinada **f**, para un nodo **N**, representa, por tanto, una estimación del coste involucrado en llegar desde el nodo inicial hasta un nodo meta a través del camino que genera el presente nodo **N**. Si más de un camino generan el nodo **N**, el algoritmo **A*** registra el mejor.

Asumiendo que la estimación dada por **h** nunca excede la distancia real entre el nodo actual y la meta, el método **A*** siempre encuentra el camino más corto entre el nodo inicial y la meta. Esto se conoce como la **admisibilidad** del método **A***.

Para implementar este *método*, cada nodo contendrá la misma información que para el método "El Mejor Primero" y se utilizarán las mismas listas **OPEN** y **CLOSED**.

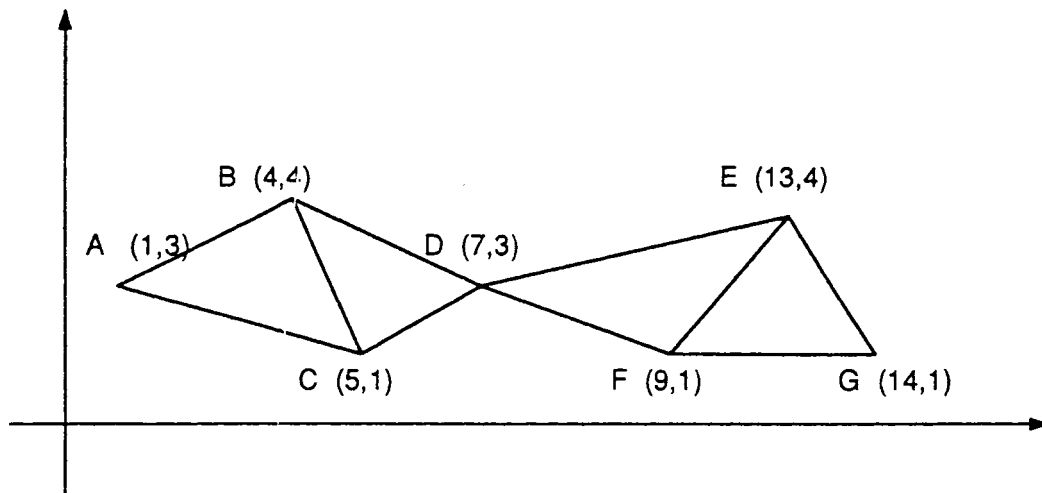
ALGORITMO DE BÚSQUEDA A*

- 1.-Meter en **OPEN** el nodo inicial. Inicializar el valor **g** de este nodo a 0, el valor **h** a lo que corresponda, y el valor **f** a **(0 + h)**. Inicializar **CLOSED** con la lista vacía.
- 2.-Hasta que se encuentre un nodo meta, repetir el siguiente procedimiento: Si no hay ningún nodo en **OPEN**, devolver fallo. En caso contrario coger el nodo de **OPEN** que tenga el menor valor para **f**. Llamar a este nodo **MEJOR-NODO**. Eliminarlo de **OPEN**. Ponerlo en **CLOSED**. Ver si **MEJOR-NODO** es un nodo meta. Si lo es devolver la solución (bien **MEJOR-NODO** o el camino creado entre el estado inicial y **MEJOR-NODO**, según lo que se quiera). En caso contrario, generar los sucesores de **MEJOR-NODO**. Para cada **SUCESOR** hacer lo siguiente:
 - a.- Poner el enlace-padre de **SUCESOR** apuntando a **MEJOR-NODO**.
 - b.- Evaluar $g(SUCESOR) = g(MEJOR-NODO) + \text{coste de ir desde MEJOR-NODO a SUCESOR}$.
 - c.- Ver si **SUCESOR** está ya en **OPEN** (es decir, ya ha sido generado pero no procesado):
 - i.- Si **SUCESOR** está ya en **OPEN**: llamar **VIEJO** a ese nodo. Hay que decidir si el enlace-padre de **VIEJO** debe cambiarse para apuntar a **MEJOR-NODO**. Este debe ser el caso cuando el camino que acabamos de encontrar hasta **SUCESOR** es más "barato" que el camino ya existente hasta **VIEJO** (**SUCESOR** y **VIEJO** son el mismo nodo en el grafo). Para ello se comparan los valores **g** de **SUCESOR** y **VIEJO**. Si **VIEJO** es más barato (o igual), no hacer nada. Si, por el contrario, **SUCESOR** es más barato, cambiar el enlace-padre de **VIEJO** para apuntar a **MEJOR-NODO**, registrar el nuevo camino más barato en $g(viejo)$ y actualizar $f(VIEJO)$. Añadir **VIEJO** a la lista de sucesores de **MEJOR-NODO**.

ii.- Si **SUCESOR** no está ya en **OPEN**, ver si está en **CLOSED**. Si es así, llamar **VIEJO** a este nodo. Ver cual de los dos caminos, el nuevo o el viejo, es más barato (como en el paso anterior), y ajustar el enlace-padre, y los valores de g y f para **VIEJO**, según se indica en el apartado c.i.. Si se ha encontrado un camino mejor a **VIEJO**, esta mejora se debe propagar a los sucesores de **VIEJO**. Para conseguir esto, hay que tener en cuenta que **VIEJO** contiene la lista de sus sucesores. A su vez, cada uno de ellos tiene, respectivamente, la lista de sus sucesores, y así sucesivamente, hasta que cada rama termina con un nodo que, o bien está en **OPEN**, o no tiene sucesores. Por tanto, para propagar el nuevo coste hacia abajo, hay que hacer una búsqueda en profundidad del subarbol que comienza en **VIEJO**; cambiando el valor g de cada nodo (y luego el de f), y terminando cada rama cuando se llega, bien a un nodo sin sucesores o a un nodo que ya tenía un camino equivalente o mejor (esto garantiza que el algoritmo termina aún cuando se encuentren bucles en el grafo). Para comprobar esta última condición, hay que recordar que el enlace-padre de cada nodo apunta hacia el mejor padre de generados hasta el momento. Añadir **VIEJO** a la lista de sucesores de **MEJOR-MODO**.

d.- Si **SUCESOR** no está ya en **OPEN** o en **CLOSED**: Poner **SUCESOR** EN **OPEN** y sumarlo a la lista de sucesores de **MEJOR-NODO**. Evaluar: $f(SUCESOR) = g(SUCESOR) + h(SUCESOR)$.

Ejemplo:



¿Es posible aplicar el método de búsqueda A* en este "mapa de carreteras"?

Si tenemos la función $f(N) = g(N) + h(N)$ donde:

$g(N)$ = menor distancia real desde el inicio hasta N.

$h(N)$ = menor distancia estimada desde N hasta la meta.

Mediante las coordenadas X, Y de cada nodo se puede calcular la distancia entre cada par de nodos, luego sí podemos hallar la $g(N)$. Igualmente podemos estimar la "distancia en línea recta" desde N hasta la meta, por lo tanto, también se puede hallar $h(N)$. Por consiguiente, podemos aplicar el método de A*.

Solución:

1.- Hay un único nodo de OPEN, el origen A.

OPEN \leftarrow A

P(A) = NULL

$g(A) = 0$

$h(A) = \sqrt{173}$

$f(A) = \sqrt{173}$

quedando al final de esta primera etapa:

OPEN = A

CLOSED = NIL

2.- A ≠ Meta y además es el único nodo en OPEN luego,

CLOSED ← A

Generamos los sucesores de A

B, C ∉ OPEN ∨ CLOSED

P(B) = A

P(C) = A

$g(B) = g(A) + \text{dist}(A - B)$
 $= 0 + \sqrt{10} = \sqrt{10}$

$g(C) = g(A) + \text{dist}(A - C)$
 $= 0 + \sqrt{20}$

$h(B) = \sqrt{109}$

$h(C) = \sqrt{81}$

$f(B) = \sqrt{10} + \sqrt{109} = 13.6$

$f(C) = \sqrt{20} + \sqrt{81} = 13.47$

OPEN ← B

OPEN ← C

quedando al final de esta etapa:

OPEN ← B, C

CLOSED ← A

3.- $f(C) < f(B)$ luego metemos C a CLOSED

CLOSED ← C

generamos los sucesores de C, que en este caso es únicamente D

D ∉ OPEN ∨ CLOSED

P(D) = C

$g(D) = g(C) + \text{dist}(C - D) = \sqrt{20} + \sqrt{8} = 7.3$

$h(D) = \sqrt{53}$

$f(D) = 7.3 + \sqrt{53} = 14.58$

OPEN ← D

quedando al final de esta etapa:

OPEN \leftarrow B, D
CLOSED \leftarrow A, C

4.- $f(B) < f(D)$

CLOSED \leftarrow B

generamos los sucesores de B que en este caso serían D y C

el sucesor D \in OPEN

$P'(D) = B$

$$\begin{aligned} g'(D) &= g(B) + \text{dist}(13 - D) \\ &= \sqrt{10} + \sqrt{10} = 6.32 \end{aligned}$$

$$g'(D) = 6.32 < g(D) = 7.3$$

luego hay que modificar

$P(D) = B$

$g(D) = 6.32$

$$\begin{aligned} f(D) &= g'(D) + h(D) \\ &= 6.32 + \sqrt{53} \\ &= 13.6 \end{aligned}$$

quedando al final de esta etapa:

OPEN \leftarrow D
CLOSED \leftarrow A, B

el sucesor C \in CLOSED

$P'(C) = B$

$$\begin{aligned} g'(C) &= g(B) + \text{dist}(B - C) \\ &= \sqrt{10} + \sqrt{10} = 6.32 \end{aligned}$$

$$g'(C) > g(C)$$

luego no cambia nada

$P(C) = A$

$g(C) = 4.47$

$$f(C) = 13.47$$

5.- Solo hay un nodo en OPEN, luego no hay que compararlo

CLOSED \leftarrow D

generamos los sucesores de D, y en este caso son E y F

$E \notin \text{OPEN} \vee \text{CLOSED}$

$$P(E) = D$$

$$\begin{aligned} g(E) &= g(D) + \text{dist}(D - E) \\ &= 6.32 + \sqrt{37} = 12.4 \end{aligned}$$

$$h(E) = \sqrt{10}$$

$$f(E) = 12.4 + \sqrt{10} = 15.56$$

$F \notin \text{OPEN} \vee \text{CLOSED}$

$$P(F) = D$$

$$\begin{aligned} g(F) &= g(D) + \text{dist}(D - F) \\ &= 6.32 + \sqrt{8} = 9.15 \end{aligned}$$

$$h(F) = \sqrt{25} = 5$$

$$f(F) = 9.15 + 5$$

$\text{OPEN} \leftarrow E$

$\text{OPEN} \leftarrow F$

quedando al final de esta etapa:

$\text{OPEN} \leftarrow E, F$
 $\text{CLOSED} \leftarrow A, B, D$

6.- $f(F) < f(E)$, por lo tanto, metemos F en CLOSED

$\text{CLOSED} \leftarrow F$

generamos los sucesores de F, siendo en este caso solo G

$G \notin \text{OPEN} \vee \text{CLOSED}$

$$P(G) = F$$

$$\begin{aligned} g(G) &= g(F) + \text{dist}(F - G) \\ &= 9.15 + 5 = 14.15 \end{aligned}$$

$$h(G) = 0$$

$$f(G) = 14.15 + 0 = 14.15$$

$\text{OPEN} \leftarrow G$

quedando al final de esta etapa:

$\text{OPEN} \leftarrow E, G$
 $\text{CLOSED} \leftarrow A, B, D, F$

7.- $f(G) < f(E)$ y además G es la meta luego metemos G a CLOSED

$\text{CLOSED} \leftarrow G$

Quedando al final de esta etapa:

OPEN \leftarrow E
CLOSED \leftarrow A, B, D, F, G

Para, por fin, obtener el camino solución:

G
P(G) = F
P(F) = D
P(D) = B
P(B) = A