

UNIVERSIDAD MAYOR DE SAN SIMON
FACULTAD DE CIENCIAS Y TECNOLOGIA

BUSQUEDA

AVARA

DOCENTE	Lic. Garcia Perez Carmen Rosa
MATERIA	Inteligencia Artificial
ALUMNOS	Flores Montero Danny Sergio Caceres Quiroga Sergio Medina Rodríguez Yamil Alexis Ugarte Mamani Andres Carlos
FECHA	07/ abril / 2014

Cochabamba – Bolivia

BUSQUEDA AVARA

Muchos algoritmos en la inteligencia artificial son heurísticos por naturaleza, o usan reglas heurísticas, cualquiera de las reglas usadas de forma independiente puede llevar a errores de clasificación pero cuando se unen múltiples reglas la solución es más robusta y creíble.

Por heurística entendemos una estrategia, método, criterio o truco usado para hacer más sencilla la solución de problemas difíciles. El conocimiento heurístico es un tipo especial de conocimiento usado por los humanos para resolver problemas complejos. En este caso el adjetivo heurístico significa medio para descubrir.

Debido a la existencia de algunos problemas importantes con un gran interés práctico difíciles de resolver, comienzan a surgir algoritmos capaces de ofrecer posibles soluciones que aunque no consiguen el resultado óptimo, si que se acercan en un tiempo de cálculo razonable. Por lo general, estos algoritmos heurísticos encuentran buenas soluciones, aunque a veces no hay pruebas de que la solución pueda hallarse en un tiempo razonablemente corto o incluso de que no pueda ser errónea. Frecuentemente pueden encontrarse casos particulares del problema en los que la heurística obtendrá resultados muy malos o que tarde demasiado en encontrar una solución.

Un método heurístico es un conjunto de pasos que deben realizarse para identificar en el menor tiempo posible una solución de alta calidad para un determinado problema.

En computación, dos objetivos fundamentales son encontrar algoritmos con buenos tiempos de ejecución y buenas soluciones, usualmente las óptimas. Una heurística es un algoritmo que abandona uno o ambos objetivos; por ejemplo, normalmente encuentran buenas soluciones, aunque no hay pruebas de que la solución no pueda ser arbitrariamente errónea en algunos casos; o se ejecuta razonablemente rápido, aunque no existe tampoco prueba de que siempre será así. Las heurísticas generalmente son usadas cuando no existe una solución óptima bajo las restricciones dadas (tiempo, espacio, etc.), o cuando no existe del todo.

A menudo, pueden encontrarse instancias concretas del problema donde la heurística producirá resultados muy malos o se ejecutará muy lentamente. Aun así, estas instancias concretas pueden ser ignoradas porque no deberían ocurrir nunca en la práctica por ser de origen teórico. Por tanto, el uso de heurísticas es muy común en el mundo real.

Para problemas de búsqueda del camino más corto el término tiene un significado más específico. En este caso una heurística es una función matemática, $h(n)$ definida en los nodos de un árbol de búsqueda, que sirve como una estimación del coste del camino más económico de un nodo dado hasta el nodo objetivo.

A la vez la búsqueda heurística se divide en:

- Búsqueda preferente por lo mejor.
- Búsqueda limitada por la capacidad de la memoria
- Búsqueda de mejoramiento iterativo

La búsqueda avara es una búsqueda preferente por lo mejor que consiste en expandir primero aquel nodo con mejor evaluación. Dicha evaluación al nodo la cual devuelve un número que sirve para representar lo deseable que sería la expansión de un nodo.

La búsqueda avara es una búsqueda heurística informada la cual consiste en que utiliza el conocimiento específico del problema más allá de la definición del problema en sí mismo, puede encontrar soluciones de una manera más óptima, es una de las más sencillas estrategias consiste en reducir al mínimo el costo estimado para una meta, este costo no se puede calcular con precisión.

Para estimar el costo de la búsqueda avara se utiliza una función llamada función heurística simbolizada por h .

$h(n)$ es una función *heurística*, estima el menor costo desde el nodo n hasta la meta.

h puede ser cualquier función. El único requisito es que $h(n)=0$ cuando n es una meta, cuando los problemas son de determinación de rutas en el mundo real, una buena función heurística es la distancia en línea recta a la meta. Es plausible que si la distancia área es menor entonces el nodo esté más cerca de la meta y por lo tanto conviene expandirlo.

La calidad de la función $h(n)$ determinará el grado de disminución de la complejidad temporal y de memoria con respecto a la búsqueda profundidad primero. Aunque si la heurística es buena uno puede esperar que el camino hallado sea lo suficientemente aceptable. Se implementa en nuestro algoritmo en la función *Obtener_Nodo* para que devuelva el nodo de menor heurística (o puede también insertarse en forma ordenada en la lista ABIERTA).

Una función heurística óptima sería por ejemplo para encontrar una ruta la distancia en línea recta hasta un nodo meta.

Para calcular por ejemplo la mejor ruta entre nodos se requiere datos heurísticos que nos ayuden a encontrar la ruta menos costosa, los nodos a expandirse se van ordenando en la lista según el valor de $h(n)$, se va verificando si el nodo en el que estamos contiene un estado objetivo, si no entonces se expande a sus nodos adyacentes.

Búsqueda avara:

Inglés: greedy search

Idea:

minimizar el coste estimado *para llegar a la meta*

Estrategia:

Entre las hojas del árbol de búsqueda, seleccionar el nodo que minimice $h^*(n)$

Algoritmo:

mantener la lista *abierta* ordenada por valores crecientes de h^*

insertar nuevos nodos en *abierta* según sus valores h^*

{búsqueda avara}

abierta \leftarrow s0

Repetir

Si vacía?(abierta) **entonces**

devolver(negativo)

nodo \leftarrow primero(abierta)

Si meta?(nodo) **entonces**

devolver(nodo)

sucesores \leftarrow expandir(nodo)

Para cada n que pertenece sucesores **hacer**

 n.padre \leftarrow nodo

ordInsertar(n,abierta,h*)

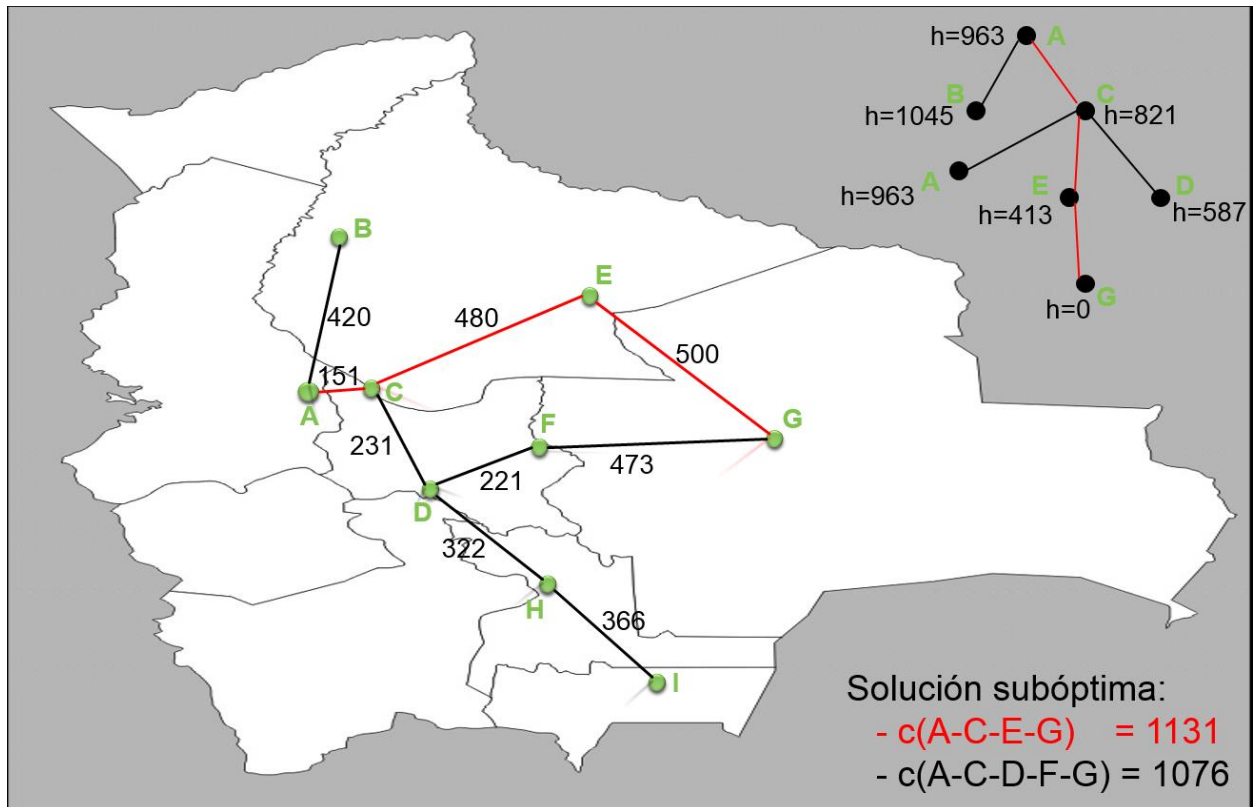
Fin {repetir}

Solución = 58

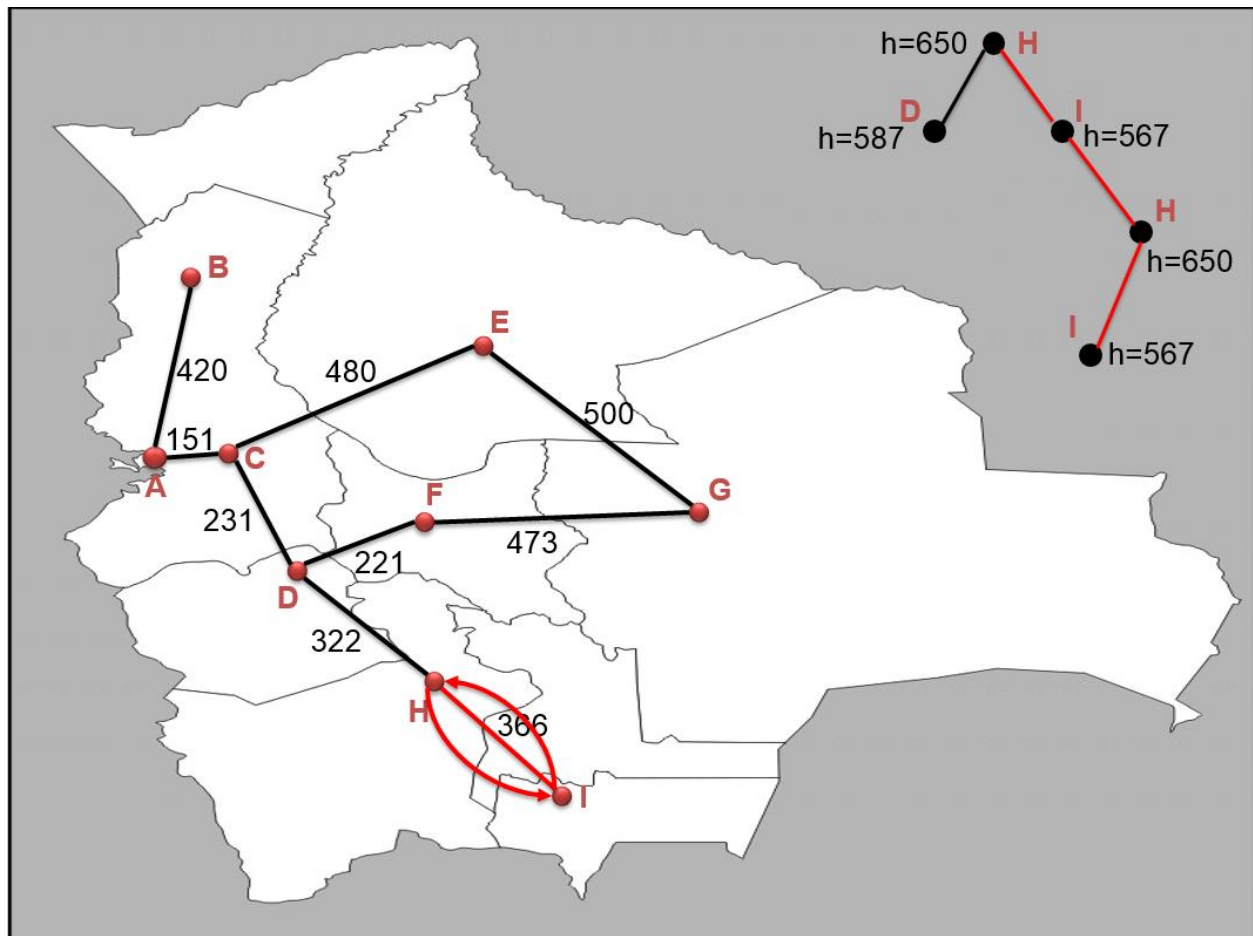
Ejemplo

8	1	5	7	9	6	3	5	4	7
9		Meta N=0							4
2		4	8	9	6	3	2		4
5		1	5	1		8	3		2
1		7	6	8		1	5	8	8
7						6	4	9	2
8	4	1	4	6	7	5	7	9	8
1	3	5	8	8	Inicio	9	8	9	8

NO OPTIMA

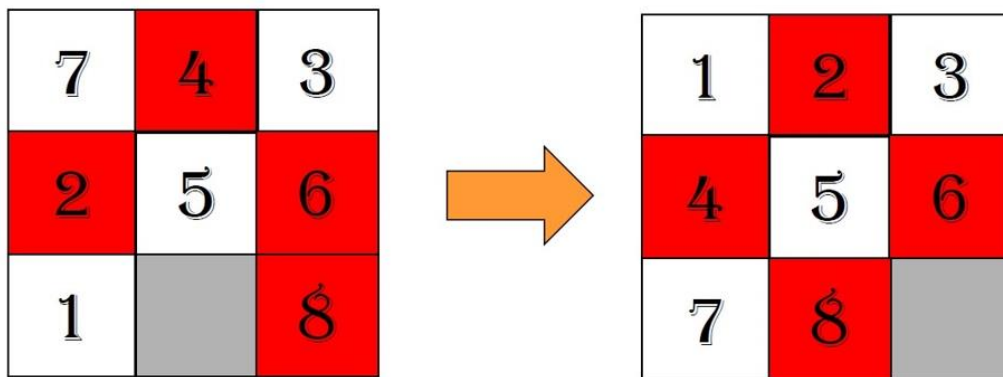


NO COMPLETA

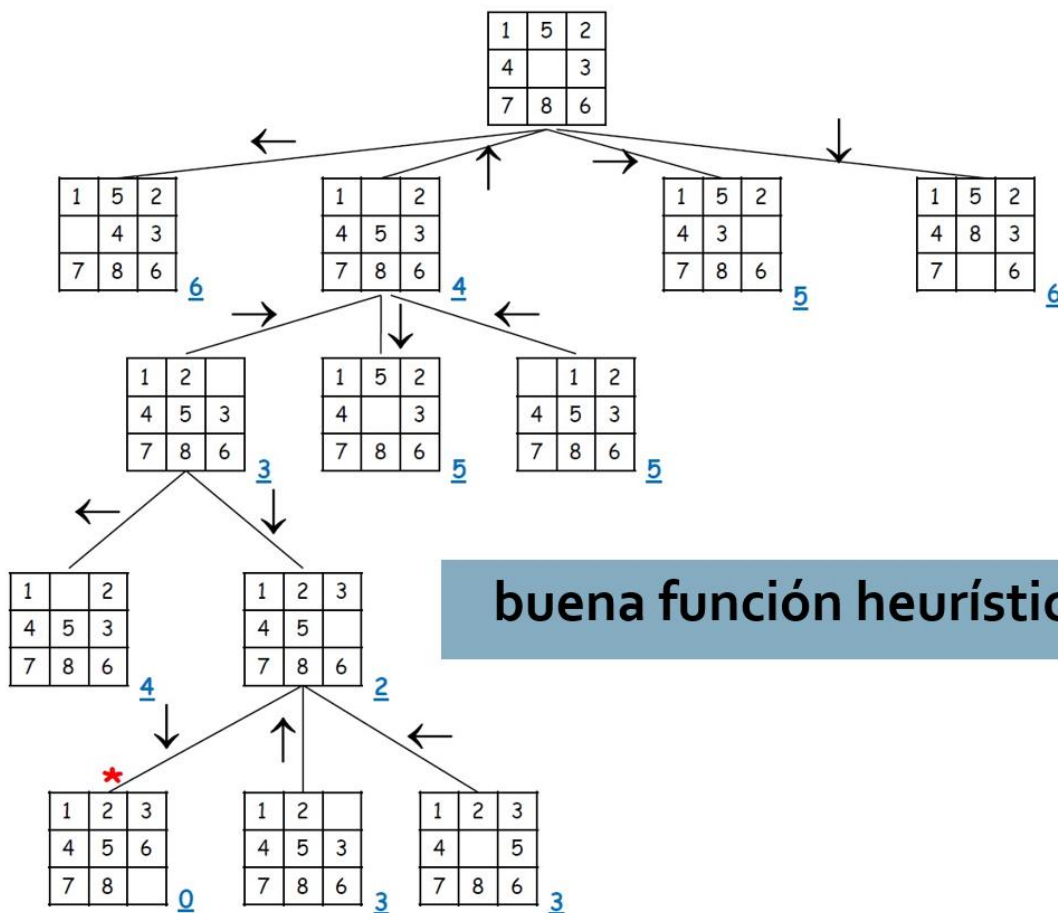


EJEMPLO 2

8-puzzle



$h(n)$: cantidad de piezas ubicadas en el lugar incorrecto (teniendo en cuenta la placa vacía)



Esta búsqueda usualmente produce resultados buenos, tienden a producir soluciones rápidamente, aunque no siempre la solución encontrada es la óptima.

La búsqueda avara no es **óptima** por que el tiempo y la cantidad de memoria crece exponencialmente con respecto a la profundidad pero la elección de una buena función heurística permite disminuir notablemente la complejidad tanto en tiempo como espacio, es decir solo considera el coste para llegar al nodo actual.

La búsqueda avara no es **completa** ya que se “atora” al toparse con un callejón sin salida, también puede recorrer una ruta infinita.

Sin embargo suele encontrar una solución aceptable de forma rápida.

Conclusiones

Consiste en reducir al mínimo el costo estimado para alcanzar una meta.

Para ello se utiliza una función llamada heurística, la cual estima el costo que implica llegar a una meta desde un estado determinado, y elige cual es el siguiente nodo que se va a expandir aplicando esta función a cada nodo.

En esta búsqueda el tiempo y la cantidad de memoria necesaria crece exponencialmente con respecto a la profundidad. Pero la elección de una buena función heurística permite disminuir notablemente la complejidad tanto en tiempo como en espacio. En el peor de los casos se almacenan todos los nodos en memoria.

No es optima ni completa.

Para asegurar la completitud habría que evitar todos los estados repetidos.

El método es optimo solo en aquellos espacios de estados en los que el coste de un nodo n es independiente del camino por el que se llega hasta el.