



INTELIGENCIA ARTIFICIAL

Trabajo Práctico N° 3
Búsqueda Ciega y Heurística
Segundo Cuatrimestre de 2008

1. Búsqueda en la resolución de problemas.

- ¿Cuáles son los principales elementos presentes en un problema de búsqueda?
- ¿Qué estructuras de datos resultan adecuadas para representar un problema de búsqueda?
- ¿Qué implica tener un método o una estrategia para recorrer un espacio de búsqueda?

2. Búsqueda ciega.

- ¿Qué significa que una estrategia de búsqueda sea ciega?
- Considere el siguiente algoritmo general de búsqueda:

```
buscarEn(Frontera) ← seleccionar(Nodo, Frontera, F),  
                    esMeta(Nodo).  
buscarEn(Frontera) ← seleccionar(Nodo, Frontera, F),  
                    vecinos(Nodo, Vs).  
                    agregar(F, Vs, F1).  
                    buscarEn(F1).
```

- ¿En qué parte del algoritmo se determina la estrategia de búsqueda utilizada?
- ¿De qué forma modificaría el algoritmo para obtener la solución de la búsqueda, es decir, el camino desde el estado inicial a la meta hallada? Analice distintas formas en las que se podría obtener o reconstruir este camino.

3. Búsqueda en profundidad (DFS) y Búsqueda a lo ancho (BFS)

- Describir los métodos de búsqueda DFS y BFS
- Modifique el algoritmo del ejercicio 2 para realizar DFS y luego para realizar BFS.
- Al realizar búsqueda DFS la frontera se manipula/comporta como una conocida estructura de datos. ¿A qué estructura de datos nos referimos? ¿Y al realizar búsqueda BFS?
- Contrastar estos métodos entre sí e indicar en qué circunstancias resulta preferible cada uno de ellos.
- ¿Qué problemas se originan si el espacio de búsqueda considerado define un grafo con ciclos? ¿Cómo se pueden solucionar esos problemas?

4. Búsqueda *bi-direccional*.

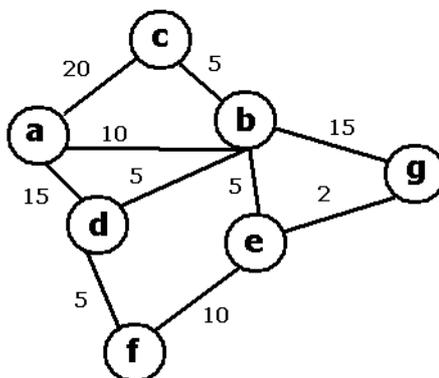
Describa esta estrategia de búsqueda y analice bajo qué condiciones resulta aplicable. Dar un ejemplo en el cuál resulte adecuado utilizar esta estrategia y otro en el cual no sea aplicable.

5. Método de búsqueda *iterative deepening*.
 - a) Escribir un algoritmo para el método de búsqueda *iterative deepening*.
 - b) Se dice que *iterative deepening* combina las ventajas de dos métodos de búsqueda. ¿Cuáles son estos métodos? Justificar adecuadamente.
6. Compare las distintas estrategias de búsqueda enunciadas en base a los siguiente criterios:
 - Completitud
 - Optimalidad
7. Evaluar la complejidad temporal y la complejidad espacial (para el peor caso) de los métodos de búsqueda *depth first*, *breadth first* e *iterative deepening*, asumiendo que la búsqueda se realiza sobre un árbol con factor de ramificación r , profundidad p y que tiene un solo nodo meta a profundidad m .
8. En una habitación hay juguetes sobre el piso. Un agente debe encontrar un juguete cualquiera. El agente avanza a pasos discretos (de a una baldosa). Sus movimientos son restringidos, puede avanzar hacia adelante o hacia la derecha cada vez. Cuando se encuentra con una pared no puede avanzar hacia esa dirección. El agente tiene la capacidad de *ver* las baldosas adyacentes hacia las cuales puede avanzar. Si por alguna razón no puede seguir avanzando, el agente detiene su búsqueda.
 - a) Defina la tarea que tiene este agente como un problema de búsqueda. Determine cuáles son los estados del mundo, cuál es la meta, cuáles son las acciones, etc.
 - b) Representar a través de un grafo, el espacio de búsqueda completo si se considera una habitación de 3x4 baldosas y el agente está inicialmente en al esquina inferior izquierda. Marque sobre este grafo los estados (nodos) que se visitan realizando una búsqueda en un recorrido DFS para resolver el problema.
 - c) Considere ahora que también son válidos movimientos hacia atrás y hacia la izquierda. ¿Cómo cambia el espacio de búsqueda? ¿Cómo influye este cambio en la implementación de una estrategia de búsqueda?
9. ¿Qué es una heurística? ¿Qué diferencia fundamental existe entre la búsqueda ciega y la búsqueda heurística?
10. Métodos Best first y Hill Climbing
 - a) Escribir algoritmos para los métodos de búsqueda heurística *hill climbing* y *best first*.
 - b) ¿Qué diferencias existen entre estos dos métodos? Hallar, de ser posible, un espacio de búsqueda en el cual se comporten de manera diferente uno del otro.
 - c) Enumerar los principales inconvenientes que presenta cada uno de estos métodos.
11. Método Primero en profundidad heurístico
 - a) Escribir un algoritmo para el método de búsqueda *primero en profundidad heurístico*.
 - b) En [PMG98] se dice que este método es una variante de método *hill climbing*. ¿En qué difieren?
 - c) Enumerar las principales desventajas de este método.

12. Método de búsqueda A^*

- Escribir un algoritmo para el método de búsqueda A^* .
- ¿Bajo qué condiciones se dice que una función heurística $h'(n)$ es admisible?
- ¿Garantiza A^* que hallará la mejor solución? ¿Cuándo puede garantizarse esto?
- Sea h' una función heurística tal que $h'(n) = 0$, para todo nodo n . ¿Qué método de búsqueda resulta de emplear A^* con esta heurística?
- Refinar el método de búsqueda A^* para considerar la realización de podas cuando se llega por un camino alternativo a un nodo ya descubierto previamente.
- ¿Se pueden aplicar las ideas de la técnica de *iterative deepening* para mejorar el desempeño de A^* ?
- De acuerdo al algoritmo A^* , la búsqueda no finaliza hasta que un nodo meta es seleccionado para ser visitado. Sin embargo, cabe destacarse que el nodo meta pudo haber sido generado mucho antes. Considerando esto, se podría pensar en modificar A^* para que la búsqueda finalice ni bien se genere un nodo meta. ¿Por qué razón esta modificación no es considerada? Ilustrar la respuesta mediante un ejemplo.
- Enunciar y demostrar la propiedad de admisibilidad del método de búsqueda A^* .
- Dar un ejemplo en el cual A^* no encuentra una solución óptimal (por supuesto, consecuencia de violar alguna de las condiciones de *admisibilidad*).

13. Considere la siguiente figura:



Suponga que cada uno de los nodos representa una ciudad y los arcos representan caminos (ida y vuelta) entre pares de ciudades. La etiqueta de cada arco representa la distancia entre ciudades. Se desea encontrar un camino desde una la ciudad a a cualquiera de las *metas*. Las metas son las ciudades f y g . Considere la siguiente función heurística h definida sobre cada uno de los nodos:

$$h(a) = 20, h(b) = 10, h(c) = 15, h(d) = 7, h(e) = 5, h(f) = h(g) = 0$$

- Muestre cómo se recorre el espacio de búsqueda si se utiliza el método *Best first search*.
- Muestre cómo se recorre el espacio de búsqueda si se utiliza el método A^* .

Mostrar claramente en ambos casos cómo evoluciona la *frontera* y el conjunto de *nodos explorados*.

14. PROLOG

- a) Escribir un programa PROLOG que represente el grafo del ejercicio 13.
- b) Siguiendo el algoritmo general de búsqueda dado en el ejercicio 2, implementar en PROLOG un algoritmo `buscar(Frontera)` que implemente búsqueda a lo ancho utilizando la representación elegida en el ejercicio anterior. Considerar que junto con la representación del grafo, existe un hecho `esMeta/1` por cada nodo que es meta. En este caso,

```
esMeta(f).  
esMeta(g).
```

Implementar el predicado de forma tal que se muestre por pantalla los nodos que son seleccionados a medida que avanza la búsqueda. (Utilizar los predicados `write/1` y `nl/0`).

- c) La *solución de una búsqueda* es el camino que lleva del estado inicial a la meta encontrada. Modificar el predicado implementado para que, además de mostrar por pantalla los nodos seleccionados, retorne en un segundo parámetro la solución de la búsqueda.
- d) Modificar el predicado implementado para hacer búsqueda en profundidad. Considerar que el grafo tiene ciclos, por lo tanto es necesario evitarlos en el proceso de búsqueda. Para ello se deberá mantener, además de la *frontera* de la búsqueda, una lista con los nodos ya *explorados*.
- e) Modificar la representación del grafo adoptada en el inciso *a* para representar los costos de los arcos y el valor de la función heurística en cada nodo.
- f) Utilizando la representación del ejercicio anterior, definir un predicado que implemente el algoritmo de búsqueda heurística A^* . Para almacenar los nodos de la *frontera* y los nodos ya *explorados*, mantener, respectivamente, un conjunto de hechos dinámicos de la forma:

```
open(Nodo).  
closed(Nodo).
```

donde, `Nodo` es un término PROLOG que representa toda la información que debe mantenerse en un nodo para llevar adelante la búsqueda: etiqueta (estado), heurística, camino desde el estado inicial, etc.

En el proceso de búsqueda, al generar los vecinos de un nodo `N`, se debe agregar dinámicamente un hecho de la forma `open(V)`, por cada vecino `V` de `N` que haya sido generado.

Cuando un nodo `N` termina de ser explorado (*i.e.* sus vecinos han sido generados) dicho nodo debe ser eliminado de la frontera. Para ello, se eliminará dinámicamente el hecho `open(N)` y se agregará el hecho dinámico `closed(N)` para indicar que dicho nodo ya ha sido explorado.

Importante: La implementación del predicado **debe** tener en cuenta que, durante la búsqueda, un nodo ya generado o ya explorado podría ser descubierto nuevamente sobre un camino más promisorio. En ese caso, el nodo debe ser explorado nuevamente.

15. Misioneros y caníbales.

En una costa de un río hay tres misioneros y tres caníbales. Cuentan con un bote que puede transportar hasta dos personas y lo usarán para cruzar el río. Desafortunadamente, si en un determinado instante los caníbales en una orilla de río superan en número a los misioneros, entonces se los comerán. Se desea encontrar una serie de movimientos para que los misioneros y los caníbales se trasladen desde una orilla del río a la otra sin tener que lamentar víctima alguna.

- a) Reformular el problema anterior como un problema de búsqueda. ¿Cuál sería una representación adecuada para los estados?
- b) Haciendo uso del planteo anterior, escribir en PROLOG un programa que solucione el problema de los misioneros y los caníbales utilizando alguna de las estrategias de búsqueda vistas.

Referencias

- [PMG98] POOLE, D., MACKWORTH, A., AND GOEBEL, R. *Computational Intelligence: A Logical Approach*. Oxford University Press, 1998.
- [RN95] RUSSEL, S., AND NORVIG, P. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1995.
- [Sho94] SHOHAM, Y. *Artificial Intelligence Techniques in Prolog*. Morgan Kaufmann Publishers, 1994.