Practica 4a
Mecanismo de búsqueda informada
Primero el Mejor y Escalamiento de la colina

Introducción

Los algoritmos de búsqueda se clasifican en: *a*) *Informados* y *b*) *No informados*. Dentro de los algoritmos informados se encuentran, entre otros: *i*) Búsqueda Primero el Mejor y *ii*) Búsqueda de escalamiento de la colina.

Los métodos no informados o "ciegos" realizan una búsqueda exhaustiva sin "ver" hacia donde se dirigen. Si se contara con información general que pudiera guiar la búsqueda, podría utilizarse para dirigirla desde su mismo inicio. Esto es precisamente lo que proponen los mecanismos de búsqueda informados. Se utiliza una función *f(N)* que mapea cada nodo *N* con un número real que sirve para *estimar* el costo o beneficio relativo de continuar la búsqueda en dirección de ese nodo. Esa función se conoce como *función de evaluación*. La función de evaluación se define en base a la experiencia, conocimiento del problema, sentido común o intuición de quien implementa el algoritmo y por tanto es una *función heurística*. Por ejemplo, para encontrar el recorrido más corto entre dos nodos podríamos usar una función *f(N)* que estime la distancia entre cada y el nodo meta. La función se utiliza para decidir el orden en el cual se visitarán los nodos, visitando primero los nodos que se estima tienen un costo menor o un mayor beneficio. Con la ayuda de una función de evaluación heurística *h(X)* se busca encontrar un nodo meta *g* partiendo de un nodo inicial *S*.

Suponga que se tiene el espacio de estados mostrado en la Ilustración 1, donde se saben las coordenadas de cada nodo en el espacio de estados. Una manera de estimar la conveniencia (una heurística) de cada nodo es calcular la distancia que hay en línea recta desde dicho nodo al nodo meta. Nótese que se trata sólo una estimación, no necesariamente indica siempre la conveniencia real de cada nodo para acercarnos más rápido al nodo meta. Recuerde que la distancia entre dos puntos (x_1,y_1) y (x_2,y_2) en un plano cartesiano está dada por:

$$\text{distancia-en-linea-recta} = \text{sqrt}((x_1-x_2)^2 + (y_1 -y_2)^2)$$

En esta práctica usarán un **programa recursivo de búsqueda primero el mejor**, donde la heurística supone que el nodo sucesor con menor distancia en línea recta hasta el nodo meta es el más conveniente y posteriormente lo modificarán para implementar la estrategia **escalamiento de colina**.

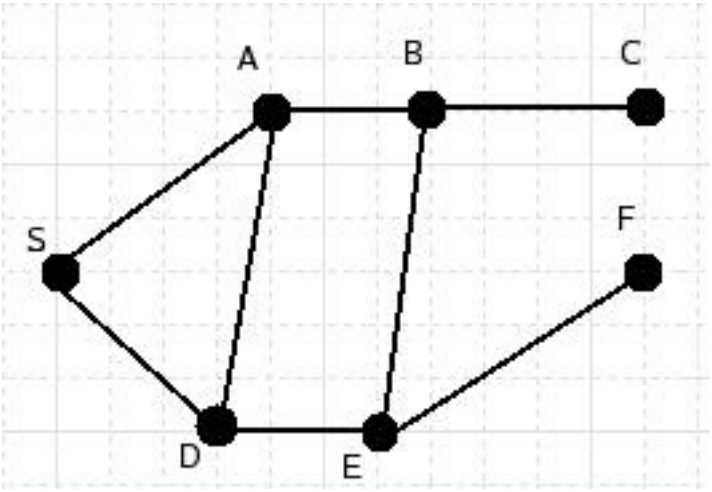

Ilustración 1: Espacio de estados con coordenadas

Objetivo

Conocer y aplicar los algoritmos de búsqueda informada primero el mejor y escalamiento de colina en un espacio de estados explícito.

Actividades

1.- Captura y prueba del programa recursivo.

1.1.- Captura el siguiente programa en Common Lisp en un archivo llamado `busqinf.lsp` :

```
;;; -------------------------------------------------------
;;; Archivo: busqinf.lsp
;;; Implementa la busqueda informada primero el mejor
;;; en un espacio de estado explicito.
;;; -------------------------------------------------------
(defun espacio ()
     (setf (get 's 'vecinos) '(a d)
          (get 'a 'vecinos) '(s b d)
          (get 'b 'vecinos) '(a c e)
          (get 'c 'vecinos) '(b)
          (get 'd 'vecinos) '(s a e)
          (get 'e 'vecinos) '(b d f)
          (get 'f 'vecinos) '(e)
     )

     (setf (get 's 'coordenadas) '(0 3)
          (get 'a 'coordenadas) '(4 6)
          (get 'b 'coordenadas) '(7 6)
          (get 'c 'coordenadas) '(11 6)
          (get 'd 'coordenadas) '(3 0)
          (get 'e 'coordenadas) '(6 0)
          (get 'f 'coordenadas) '(11 3) )
)

(defun extiende (trayectoria)
     (mapcar #'(lambda (nuevo-nodo) (cons nuevo-nodo trayectoria))
               (remove-if #'(lambda (vecino) (member vecino trayectoria))
                    (get (first trayectoria) 'vecinos))
     )
)

(defun DLR (n1 n2)
     (let ((x1 (first (get n1 'coordenadas)))
           (y1 (second (get n1 'coordenadas)))
           (x2 (first (get n2 'coordenadas)))
           (y2 (second (get n2 'coordenadas))) )
               (sqrt (+ (expt (- x2 x1) 2) (expt (- y2 y1) 2)))
     )
)

(defun mas-cercap (trayectoria1 trayectoria2 nodo-destino)
     (<   (DLR (first trayectoria1) nodo-destino)
          (DLR (first trayectoria2) nodo-destino))
)

(defun primero-el-mejor (inicial final &optional (cola (list (list
inicial))))
     (cond ((endp cola) nil)    ; ¿esta vacia la cola?
           ((eq final (first (first cola))) ; trayectoria completa?
               (reverse (first cola)) )    ; Devuelve trayectoria.
```

```
            (t (primero-el-mejor inicial final
                   (sort (append (extiende (first cola)) (rest cola))
                      #'(lambda (t1 t2) (mas-cercap t1 t2 final)))))
      )
)
```

1.2.- Realiza las siguientes búsquedas ejecutando la funcion `(primero-el-mejor 'x 'y)` donde `x` es el nodo inicial y `y` el nodo final y reporta el resultado de cada una de ellas.

Nodo inicial	Nodo Final	Trayectoria
S	F	_____
F	S	_____
A	F	_____
F	A	_____
S	B	_____
B	S	_____

2.- Cambio de la función de evaluación.

Suponga ahora que la función de estimación heurística *h(N)* se basa en la distancia horizontal entre el nodo *N* y el nodo meta, esto es

$f(N) = ABS(x_1-x_2)$

en lugar de

$f(N) = sqrt((x_1-x_2)^2 + (y_1 -y_2)^2)$

2.1.- Modifica la función **`DLR( )`** para que refleje esta nueva heurística.
Reporta el código de la nueva función `DLR()`.

2.2.- Comprobación de la nueva heurística.

Repite las búsquedas del inciso 1.2 con el programa modificado y reporta los resultados:

Nodo inicial	Nodo Final	Trayectoria
S	F	_____
F	S	_____
A	F	_____
F	A	_____
S	B	_____
B	S	_____

2.3.- ¿Hubo cambios en las trayectorias exploradas? Sí: _____ No: _____
Si los hubo, explica porqué crees que los hubo: _____
_____
_____
_____
_____

2.4.- ¿Cuál heurística funcionó mejor para este problema? ¿Porqué?

_____
_____
_____
_____

3.- Estrategia Escalamiento de colinas

La estrategia de ***búsqueda por escalamiento de colinas*** difiere de ***primero el mejor*** en que la nueva cola se hace ordenando los nuevos vecinos y colocándolos a la cabeza de ella, en lugar de ordenar toda la cola. Modifica el procedimeinto `PRIMERO-EL-MEJOR` para implementar la estrategia `ESCALAMIENTO-DE-COLINAS`. **Utiliza como función heurística la distancia en línea recta entre dos puntos.**

Reporta el código completo de `ESCALAMIENTO-DE-COLINAS.`

3.3.- Ejecuta las mismas búsquedas de los incisos 1.2 y comenta sobre las diferencias en las trayectorias exploradas, si las hubiera y explica las diferencias si es el caso.

Nodo inicial	Nodo Final	Trayectoria
S	F	_____
F	S	_____
A	F	_____
F	A	_____
S	B	_____
B	S	_____

¿Hubo cambios en las trayectorias exploradas? Sí: _____ No: _____
Si los hubo, explica porqué crees que los hubo: _____

_____
_____
_____
_____

4.- Comentarios y conclusiones

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____