

TP- 6 Inteligencia Artificial
Carlota Moncasi Gosá
839841

El problema del lobo, la cabra y la col consiste en un pastor que necesita trasladar a los 3 personajes de una orilla del río a la otra. Pero si el lobo coincide con la cabra en la misma orilla, se la come y si la cabra coincide con la col en la misma orilla, también se la come. Hay que tener en cuenta que en la barca solo caben 2 seres: el pastor y otro personaje.

De esta manera, es necesario tener en cuenta las restricciones de concurrencia entre los personajes para evitar que coincidan uno con otro.

El problema se puede solucionar llevando la cabra a la orilla final en primer lugar, ya que es la causante común de las dos restricciones mencionadas. Después, se dejaría también el lobo o la col en la orilla final pero trayendo de vuelta a la cabra a la orilla inicial, puesto que, si no, tendríamos otra vez el peligro de que la cabra y col o el lobo y cabra se comiesen mutuamente. Así, al mismo tiempo que se deja la cabra en la orilla inicial, se lleva al personaje que quede en la orilla inicial, hasta la orilla final. Por último, se trae a la cabra a la orilla final y ya estarían los 3 personajes en la orilla final.

En conclusión, las condiciones para poder transportar la col: que no coincidan el lobo y la cabra en la misma orilla, porque se comerían. Y la condición análoga para transportar el lobo sería que no coincidan la cabra y la col en la misma posición, por la misma razón. Al contrario, la cabra puede desplazarse siempre sin ninguna precondition.

Representación de estados con los personajes:

orilla (0)		orilla (1)
lobo,cabra,col		
lobo,col		cabra
	<-	
col		lobo,cabra
	<- cabra	
cabra		lobo, col
	<-	
		lobo,cabra,col

El vector state para este trabajo, tiene 3 posiciones:

state[0] = lobo en orilla inicial (0), o en orilla final (1),
state[1] = cabra en orilla inicial (0), o en orilla final (1),
state[2] = col en orilla inicial (0), o en orilla final (1),
state[3] = barca en orilla inicial (0), o en orilla final (1).

El resultado obtenido por el programa es el siguiente:

El lobo, la cabra y la col BFS -->

pathCost: 5

nodesExpanded: 7

Tiempo: 7ms

El lobo, la cabra y la col DLS(11) -->

pathCost: 11

nodesExpanded: 52

Tiempo: 3ms

El lobo, la cabra y la col IDLS -->

pathCost: 5

nodesExpanded: 85

Tiempo: 2ms

Mediante las trazas de los tres algoritmos, se pueden observar las diferencias en la ejecución del programa, tales como:

- El tiempo. El algoritmo de IDLS o Iterative Deepening Search es el más rápido, ya que combina el BFS con el DLS, uniendo todas sus ventajas. En concreto, se limita la memoria como en DLS pero se va ampliando conforme se exploran los nodos, iterativamente. Así, realiza una búsqueda profunda y encuentra la solución en un periodo corto de tiempo, como el DLS.
- El número de nodos recorridos. El algoritmo BFS o Breadth First Search es el más corto en nodos explorados ya que mira primero los nodos vecinos, en vez de los nodos sucesores en profundidad.
- El espacio que gasta en memoria. El algoritmo DLS o Deepening Limited Search tiene un coste de camino de 11 ya que se limita a esa profundidad, DLS(11).