

UNIVERSIDAD DE SALAMANCA
Grado en Ingeniería Informática
Fundamentos de Sistemas Inteligentes
Práctica de búsqueda

Es posible que hayáis oído hablar de Amazon y alguna vez habréis pensado en las características de sus almacenes (por ejemplo tienen una superficie de más de 100.000 m^2). Si nos planteamos cómo se organiza su almacén, alguno podría pensar que tendría algún parecido con unos gran superficie (Media Markt, Ikea, ...) con secciones, subsecciones, etc. La respuesta es que el sistema de almacenamiento de Amazon es un ejemplo de “Almacenamiento caótico” (<http://twistedsifter.com/2012/12/inside-amazons-chaotic-storage-warehouses/>). En la Figura 1 se puede observar el aspecto general de dicho almacén. Se buscan ventajas como: flexibilidad (se puede dejar cualquier cosa en cualquier sitio), simplicidad (los operarios no tienen que conocer las secciones, no existen!!!) y, sobre todo, la optimización.



Figura 1.- Almacén caótico. El operario va a buscar su pedido

En este momento es cuando se debe empezar a hablar de Inteligencia Artificial. El esquema general del almacén es el que se presenta en la Figura 2. El operario parte de una posición de inicio (p_s). Se va a suponer que se dispone de n estanterías (s_i) ubicadas en posiciones (ps_k) que se conectan entre sí con pasillos (en los que se detectan cruces dados por pc_j). En cada estantería tenemos colocadas una serie de objetos (artículos) en una determinada cantidad. Por otra parte, un pedido es una serie de artículos que se han de llevar a la zona de expediciones p_T en un tiempo mínimo.

Parte I. Solución en Prolog (1/3 del valor de la práctica)

Desarrollar un programa en Prolog que resuelva este problema.

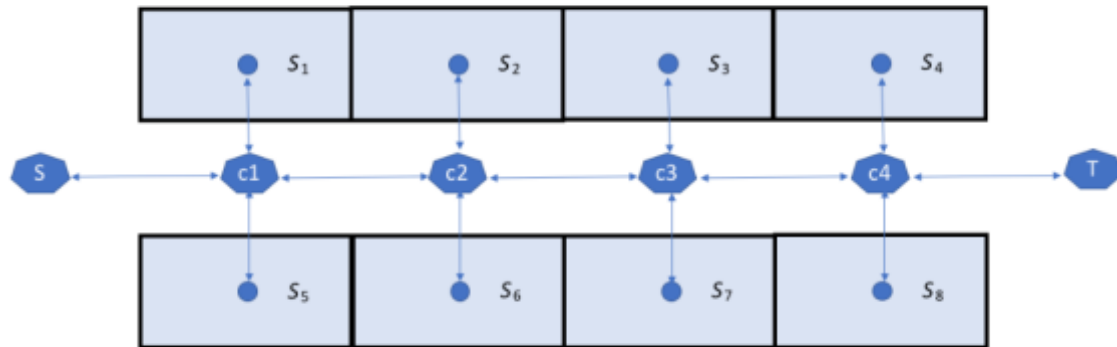


Figura 2.- Plano de estanterías del almacén caótico

Así, podremos considerar que existe una estantería mediante el predicado:

estanteria(s1,ps1)

siendo *s1* y *ps1* la identificación y posición de la estantería 1. La conectividad se establece mediante los predicados

conectado(ps1,pc1,5) que indica que *ps1* y *pc1* están conectados (ojo en las dos direcciones) y que la distancia es de 5 metros.

conectado(pci,pcj,10) que indica que *pci* y *pcj* están conectados (ojo en las dos direcciones) y que la distancia es de 10 metros. *pci* y *pcj* representan el inicio (S) los 4 centros y el final (T)

Los contenidos de cada estantería se reflejan mediante el predicado

ubicado(o11,s1,300) que indica que el objeto *o11* está en la estantería *s1* con 300 unidades.

Para la práctica considerar el almacén definido por los siguientes predicados:

```
/* Contenido del almacen */
ubicacion(patatas,s1,200).
ubicacion(melones,s1,100).
ubicacion(boligrafos,s2,500).
ubicacion(boligrafos,s3,400).

ubicacion(melocotones,s4,200).
ubicacion(berzas,s4,100).
```

```
ubicacion(papeles,s5,500) .  
ubicacion(boligrafos,s6,400) .
```

```
ubicacion(patatas,s1,200) .  
ubicacion(melones,s1,100) .  
ubicacion(plumas,s7,500) .  
ubicacion(plumas,s8,400) .
```

```
ubicacion(colonias,s3,150) .  
ubicacion(ratones,s4,210) .
```

El problema es determinar el viaje (una secuencia de posiciones) que hay que realizar para recoger un pedido de una serie objetos en una cantidad cada uno

```
lineapedido=linea(objeto,unidades)
```

y un pedido lógicamente es una lista de *lineapedido*.

En Prolog nos vamos a conformar con que sea capaz de realizar el pedido:

```
[linea(patatas,40),linea(boligrafos,40),linea(plumas,1  
0)]
```

Debemos conocer la secuencia de viajes necesaria así como la distancia recorrida.

Parte II. Solución óptima (2/3 del valor de la práctica)

Utilizando un algoritmo A* encontrar la secuencia óptima en lo que se refiere a la distancia recorrida. Se podrá utilizar el lenguaje de programación que libremente decida el alumno. El algoritmo NO es LIBRE