

# Examen Parcial de IA

(3 de mayo de 2018)

grupo 10

Duración: 55 min

1. (6 puntos) Tenemos que planificar la topología de interconexión de un conjunto de routers que están distribuidos por el Campus Nord, de manera que podamos canalizar todo su tráfico hacia el equipo de comunicaciones que lo reenvía a internet. Para cada router  $r_i$  sabemos su localización  $l_i$  en el campus (supondremos que tenemos un sistema de coordenadas en el que podemos situar cada router) y el ancho de banda  $bw_i$  del tráfico directo que debe distribuir (el que no le llega de otros routers). También conocemos las coordenadas del equipo de comunicaciones ( $l_{EC}$ ). Cada router puede estar conectado a otro router o al equipo de comunicaciones exterior directamente (la topología ha de ser un árbol). Disponemos de tres tipos de router (tipos A, B y C) capaces de distribuir hasta cierto ancho de banda máximo cada uno ( $bw_A < bw_B < bw_C$ ). Para distribuir el tráfico directo hay suficiente con un router de tipo A. El coste de cada router es proporcional al tipo (el tipo B cuesta el doble que el A y el C el doble que el B). Para conectar cada router  $r_i$  necesitamos instalar un cable  $c_i$  de cierta longitud (supondremos que es la distancia euclídea entre las coordenadas de los dos equipos conectados), el coste de este cable es proporcional a la longitud. El coste del equipo de comunicaciones exterior es fijo y no lo tendremos en cuenta.

El objetivo es decidir el tipo de los routers dependiendo del ancho de banda que deben soportar y la forma de interconectarlos, de manera que se que minimice el coste de la instalación.

En los siguientes apartados se proponen diferentes alternativas para algunos de los elementos necesarios para plantear la búsqueda (solución inicial, operadores, función heurística,...). Comenta muy brevemente la solución que se propone respecto a si es correcta, eficiente o mejor/peor respecto a otras alternativas posibles. Justifica tus respuestas.

- a) Usar Hill-climbing. Como solución inicial conectamos directamente todos los routers al equipo de comunicaciones exterior. Como operadores usamos conectar un router a otro y desconectar un router de otro. La función heurística es el sumatorio, para todos los routers, del coste del router más el coste del cable que lo conecta (a otro router o al equipo de comunicaciones).
- b) Usar Hill-climbing. Como solución inicial conectamos cada router con el router más cercano y conectamos aleatoriamente uno de los routers al equipo exterior, asignamos a cada router el tipo A. Como operadores usamos cambiar la conexión de un router a otro router o equipo exterior por otra distinta si no superamos su capacidad y cambiar el tipo de un router. La función heurística quiere reducir al máximo la longitud de las conexiones y por ello es el sumatorio, para todos los routers, del coste del router más el coste del cable que lo conecta a otro router o al equipo de comunicaciones dividido por la longitud de esa conexión.
- c) Usar Algoritmos Genéticos. Supondremos que los routers y el equipo exterior están numerados consecutivamente, supondremos que para codificar ese número hacen falta  $n$  bits. La codificación es una tira  $r \cdot n$  de bits, donde  $r$  es el número de routers, cada grupo de  $n$  bits corresponden a un router siguiendo el orden de la numeración (es importante tener en cuenta que  $r + 1$  puede ser menor que  $2^n$ ). Como operadores usamos los operadores de cruce y mutación habituales. Para generar la población inicial usamos el método del apartado b) . La función heurística es la suma ponderada del coste de cada router (con peso 0,7) y el coste de su cable de conexión (con peso 0,3) para darle mayor relevancia al coste del router (que es lo más caro).

2. (4 puntos) La cadena de comida rápida a domicilio ComidApp ha decidido renovar su flota de motos de reparto por motos 100% eléctricas pero después de unas pruebas ha visto que necesita hacer una mejor gestión de los trayectos de entrega para reducir la distancia recorrida y sacar el máximo partido de la limitada autonomía de las motos. Para ello nos piden un sistema que decida que moto y en que orden ha de servir los pedidos realizados por los clientes desde sus casas con una app. Esta app ya envía directamente el pedido (con su localización GPS) a la tienda más cercana al cliente.

Cada tienda tiene 4 motos eléctricas para entregar los pedidos asignados a su tienda. Cada moto puede llevar un máximo de 5 pedidos a la vez. Dado que no pueden tener las motos paradas durante horas cargando sus baterías al máximo, después de hacer pruebas han visto que lo mejor es hacer cargas rápidas de 10 minutos cada vez que la moto vuelve a la tienda, dando así una autonomía máxima de 8 km. Disponemos de una función `distancia(loc_1, loc_2)` que es capaz de calcular la longitud (en km) de la ruta más corta entre dos localizaciones GPS (tanto de clientes como de la tienda). Nos piden un sistema que, cada 30 minutos, asigne los pedidos recibidos por la tienda a las motos. Cada moto recibe una lista de pedidos a servir con el orden de entrega, y debe tener en cuenta que la longitud de la ruta (de la tienda al primer cliente, del primer cliente al segundo, del segundo al tercero... y del último a la tienda) no supere la autonomía máxima. Toda solución propuesta ha de entregar todos los pedidos.

- a) Queremos utilizar  $A^*$  para minimizar el número de kilómetros recorridos por las motos para asegurar que entregan todos los pedidos sin superar su autonomía máxima. El operador es asignar un pedido de la tienda a una moto (el orden de asignación determina el orden de entrega), el coste de ese operador es la distancia desde la localización donde se encuentra la moto a la localización de entrega de ese pedido (usando la función `distancia(loc_1, loc_2)` antes mencionada). La función heurística vale infinito si a alguna de las motos se le asigna más de 5 pedidos o si alguna de las motos no tiene energía suficiente para volver desde la localización actual a la tienda, y en caso contrario es la suma de las distancias de las localizaciones de los pedidos por entregar a la localización de la moto más cercana en cada caso.
- b) Queremos utilizar satisfacción de restricciones para encontrar una asignación correcta de pedidos a motos. Asumiendo que cada moto puede llevar como máximo 5 pedidos, tenemos, para cada moto, cinco variables que representan el primer/segundo/tercer/cuarto/quinto pedido a entregar, el dominio para todas esas variables es la lista de identificadores de todos los pedidos recibidos por la tienda. Tenemos una restricción binaria  $\neq$  entre todo par de variables que impide que el mismo identificador de pedido se asigne a 2 o más variables a la vez. Tenemos una restricción n-aria entre las 5 variables de cada moto que impide que la longitud de la ruta necesaria para entregar los pedidos exceda la autonomía máxima de esa moto.

Comenta cada una de las posibilidades indicando si resuelven o no el problema, qué errores te parece que tiene cada solución y cómo se podrían corregir, y qué ventajas e inconvenientes tienen cada una de ellas. Justifica la respuesta.