

Examen Parcial de IA

(19 de abril de 2021)

Duración: 90 min

1. (6 puntos) El mago Rincewind debe componer un hechizo a partir de conocimiento arcano que se encuentra en diversas torres esparcidas por el mundo, cada una con una parte única y indispensable. Hay un total de T torres, y desde cada torre se puede viajar a cualquier otra torre, pero la distancia puede variar y a Rincewind no le gusta andar sin sentido, así que quiere hacer el recorrido con menor distancia posible.

Sin embargo, el problema no acaba aquí. **Una proporción bastante alta** de estas torres se encuentra cerrada bajo sellos arcanos que requieren de llaves (en forma de hechizos). Estas llaves se encuentran en algunas de las otras torres que Rincewind tiene que visitar. Por ejemplo, para visitar(y acceder a) la torre A Rincewind requiere de haber visitado la torre B o la torre C (ambas contienen el hechizo para abrir la torre A). Generalmente, el hechizo de cada torre se puede encontrar en una o dos torres, no en muchas más. En consecuencia, el recorrido de Rincewind debe tener en cuenta un conjunto de preordenes entre las torres. El sistema está montado de forma que todas las torres son accesibles. Es decir, no puede darse que haya un ciclo entre los preordenes como que el hechizo que abre la torre A está solo en la torre B y el que abre la torre B, solo en A (este problema no tendría solución).

Rincewind dispone de una serie de propuestas para organizar la ruta, que puede programar para que Hex (el superordenador de la universidad) saque como salida el orden de visita a las torres, minimizando la distancia.

- a) Usar el algoritmo de A^* . Definimos el estado como la asignación de torres a la ruta. El estado inicial es la ruta vacía. Tenemos un operador `visitar_torre` que asigna la siguiente torre a la ruta, el coste es la distancia entre la torre actual y la siguiente escogida. Como función heurística usamos $h(n) = M * P + k * L$, donde M es la distancia media entre dos torres, P es el número de torres a las que todavía no hemos ido (*pending*), k es un factor de ponderación y L es el número de torres todavía selladas (*locked*, es decir, el número de llaves por recoger).
- b) Usar el algoritmo de Hill Climbing. La solución inicial se contruye añadiendo todas las torres en orden alfabético. Como operador de búsqueda usamos `swap(torre1, torre2)`, que comprueba que el intercambio de torres en la ruta cumpla los pre-ordenes impuestos por los sellos. La función heurística es la suma de las distancias entre torres consecutivas en la solución.
- c) Usar un algoritmo de satisfacción de restricciones. El grafo de restricciones tendría como variables las torres, los dominios son un número natural (de 1 a T) indicando el orden de cada torre en la ruta. Como restricciones se propone una restricción n -aria tal que se conserven los preordenes establecidos por los sellos arcanos, y otra restricción n -aria para comprobar que la distancia de la ruta sea menor que una distancia D (que iremos disminuyendo y re-ejecutando a medida que encontramos soluciones hasta encontrar la óptima).

Comenta cada una de las soluciones que se proponen, analizando si la técnica escogida es adecuada para este problema, si cada uno de los elementos de la solución son correctos o no (cada uno por separado y en conjunción los unos con los otros). Incluye un análisis de los costes algorítmicos y/o factores de ramificación allá donde sea necesario. Justifica tu respuesta.

2. (4 puntos) Como una de sus medidas de reactivación económica para salir de la crisis provocada por la COVID-19, y aprovechando la gran proporción de población jubilada que ya está vacunada, el Ministerio de Sanidad, Consumo y Bienestar Social quiere proporcionar viajes gratuitos a los jubilados a través del IMSERSO. Para ello, el Ministerio intenta repartir a los J jubilados que se apuntan a la iniciativa anualmente entre los H hoteles que se han presentado al plan de ayudas del sector por la COVID-19 (no hay restricciones sobre J y H , pero claramente $J \gg H$). Cada jubilado j puede tener o no una discapacidad de tipo K(cognitiva), M(medicalización continua) o R(movilidad reducida), y puede presentar al IMSERSO un máximo de tres lugares a los que prefiere viajar. Por otra parte, cada hotel h está asociado a un lugar, dispone de un número de plazas PL_h a un precio único (cada hotel tiene un precio por plaza PR_h) y puede estar habilitado para discapacidades de tipo K, M y/o R, o no estarlo para ninguna.

El reparto debe:

- maximizar el número de jubilados con plaza asignada,
- minimizar el precio total que se deberá gastar el Ministerio,
- priorizar la asignación de jubilados que han viajado menos a través, del IMSERSO.
- intentar respetar al máximo las preferencias de los jubilados,
- respetar las discapacidades de los jubilados.

En los siguientes apartados se proponen diferentes alternativas para algunos de los elementos necesarios para plantear la búsqueda (solución inicial, operadores, función heurística, ...). El objetivo es comentar la solución que se propone, analizando si la técnica escogida es adecuada para este problema, si cada uno de los elementos de la solución son correctos o no (cada uno por separado y en conjunción los unos con los otros). Incluye un análisis de los costes algorítmicos y/o factores de ramificación allá donde sea necesario. Justifica tu respuesta.

Sea:

A = #jubilados asignados a plazas,
 B = #jubilados con preferencias satisfechas,
 C = suma de frecuencias de viajes de los jubilados asignados,
 D = precio total de la asignación,
 E = suma de los precios de todas las plazas de todos los hoteles,
 F = numero total de plazas de los H hoteles.

- a) Queremos usar Hill-climbing, tomando como solución inicial la asignación de F jubilados con menos frecuencia de viajes, intentando llenar primero los hoteles más baratos. Como operador de búsqueda usamos `intercambiar(plaza, jubilado1, jubilado2)` donde *jubilado1* está asignado a la plaza en la solución actual, *jubilado2* no está asignado a la solución actual, y la *plaza* cumple las posibles discapacidades del *jubilado2*. La función heurística será:

$$h'(n) = A + B + C + D$$

- b) Se plantea utilizar algoritmos genéticos. Asignamos a cada plaza de hotel un número de 0 a F , y representamos una solución como una secuencia de $J \cdot \log_2(F + 1)$ bits, representando para cada jubilado j la plaza de hotel asignada, o el valor 0 si no tiene plaza asignada. Para generar la población inicial obtenemos una solución asignando aleatoriamente un jubilado a cada plaza (solo F jubilados tienen un número diferente de 0). Como operadores genéticos usamos los operadores habituales de cruce y mutación. La función heurística es:

$$h'(n) = \frac{A + B}{\frac{C}{10 \cdot J} + \frac{D}{E}}$$

Las notas se publicarán el día **10 de mayo**.