Memoria Práctica 1 Heurística y Optimización

Álvaro Guerrero Espinosa - 100472294
Adrian Cortázar - 100475860

Índice

1 Introducción

El objetivo de la realización de esta práctica consiste en:

- Modelar un problema y resolverlo utilizando SCP
- Diseñar un problema y resolverlo usando búsqueda heurística (A*).

2 Descripción de los modelos

2.1 Problema 1

Representación del problema

Según el enunciado del primer problema, se nos pide modelar un parking con cantidad variable de plazas reservadas para vehículos eléctricos o con congelador. Para resolver el problema en sí, se representan los siguientes conceptos:

- Parking: se modelará como una matriz de n filas y m columnas. Dichos parámetros se elegirán por el usuario.
- Plaza $\in \mathbb{N}^2$: Una plaza del aparcamiento anterior estará representada por una tupla con sus coordenadas. Por ejemplo, para la plaza en la primera fila y columna, se representará como $(1, 1)$ en el problema.
- Vehículo: Un vehículo quedará definido por una cadena formada por la concatenación de su número identificador, su tipo (TSU si es urgente y TNU si no) y si tiene congelador o no. Ej. "1-TSU-X".

Las variables del problema indispensables para su resolución serán representadas por todos y cada uno de los vehículos que deseen ingresar al parking (V). Sus dominios serán el conjunto de posiciones asignables a dicho vehículo en el parking. Para este problema, se ha diferenciado entre dos tipos de variables:

- Vehículos equipados con congelador: Estos vehículos solo podrán ser asignados plazas eléctricas en el parking.
- Vehículos sin congelador: Estos vehículos podrán tener asignada cualquier plaza en el parking.

De esta forma, se garantiza que se cumplen las restricciones sobre asignar a todos los vehículos a una plaza del parking, y que los vehículos con congelador tengan asignada una plaza eléctrica del parking

Modelado de Restricciones

Para definir las restricciones, se ha dividido el conjunto de vehículos (V) en 2 subconjuntos disjuntos:

- V_U: conjunto de vehículos urgentes (TSU)
- V_N: conjunto de vehículos no urgentes (TNU)

La descripción de las restricciones dada la representación anterior del problema queda así:

$$a \neq b \quad \forall a, b \in V \tag{1}$$

$$(a_y \neq b_y) \vee (a_x > b_x) \quad \forall a \in V_U, b \in V_N \tag{2}$$

$$\neg(a_x = b_x = c_x) \vee \min(a_y, b_y, c_y) + 2 \neq \max(a_y, b_y, c_y) \quad \forall a, b, c \in V, a \neq b \neq c, a \neq c \tag{3}$$

$$a_x \neq b_x \vee \min(a_y, b_y) + 1 \neq \max(a_y, b_y) \vee 1 \neq \min(a_y, b_y) \neq n - 1 \quad \forall a, b \in V, a \neq b \tag{4}$$

$$n \geq 2 \vee \|V\| = 0 \tag{5}$$

(1) Todos los vehículos se tienen que aparcar en plazas distintas del aparcamiento

(2) Un vehículo de tipo urgente (TSU) no puede tener aparcado uno no urgente (TNU) en todas las posiciones de su derecha en su misma fila.
(3) Ningún trio de vehículos puede ocupar 3 plazas consecutivas en una columna, ya que esto no cumpliría con la restricción de maniobrabilidad para el vehículo central
(4) Igual a la restricción (3), pero aplicado al caso de un vehículo aparcado en el bordes del parking. En este caso, un vehículo aparcado en el borde inferior o superior del parking no puede tener a otro aparcado en el lado opuesto al borde.
(5) Restricción (4) aplicada al caso de un parking con una única fila. Ninguna de las plazas del parking tiene un hueco libre arriba o abajo, luego el problema no tiene solución en este caso

2.2 Problema 2

Parámetros globales

Para resolver este problema, se han necesitado los siguientes parámetros globales (constantes):

- $P_T \in \mathbb{N}$: plazas totales de la ambulancia
- $P_{T_C} \in \mathbb{N}$: plazas reservadas para pacientes contagiosos
- $E_0 \in \mathbb{N}$: energía inicial de la ambulancia, y valor al que se reinicia al pasar por el parking
- Casillas $= \{1, 2, X, N, C, CN, CC, P\}$: conjunto de posibles contenidos de una casilla
- $M_{ij} \in$ Casillas $\quad (i, j \in \mathbb{N}, i < N, j < M)$: mapa del problema, donde cada elemento indica el contenido de la casilla correspondiente
- energía : Casillas $\to \mathbb{N}$: función que devuelve el coste de energía de pasar por una casilla

$$\text{energía}(c) = \begin{cases} 2 & c = 2 \\ 1 & X \neq c \neq 2 \end{cases}$$

Estado

Los posibles estados de la ambulancia se han representado con una tupla con los siguientes valores:

- $P_N \in \mathbb{N}$: número de plazas actualmente ocupadas por pacientes no contagiosos
- $P_C \in \mathbb{N}$: número de plazas actualmente ocupadas por pacientes contagiosos
- $E \in \mathbb{N}$: energía actual de la ambulancia
- Pos $\in \mathbb{N}^{2\times 1}$: posición actual de la ambulancia
- Visitados: campo de bits que codifica los pacientes que han sido recogidos, donde la posición i indica si el paciente con ID i ha sido recogido o no. Este ID se obtiene a partir de posición del paciente. Esto se eligió para reducir la cantidad de memoria necesaria para codificar cada estado

Con esto, el estado inicial sería el siguiente:

- $P_N = P_C = 0$
- $E = E_0$
- Pos = posición del parking
- Visitados = 0

El estado final sería cualquier estado que cumpla las siguientes condiciones:

- $P_N = P_C = 0$
- Pos = posición del parking
- Visitados = campo de bits con todos los bits a 1

Operadores

Este problema cuenta con un operador: move(x, y). Este operador mueve la ambulancia según el desplazamiento (x, y). Para cada estado, sus sucesores serán los resultantes de aplicar este operador con los desplazamientos $(-1, 0)$, $(1, 0)$, $(0, -1)$, y $(0, 1)$, los cuales se corresponden con los movimientos horizontales y verticales permitidos.

Las precondiciones son las siguientes:

$$0 \leq \text{Pos}_x + x < N \tag{6}$$
$$0 \leq \text{Pos}_y + y < M \tag{7}$$
$$M[\text{Pos} + (x, y)] \neq X \tag{8}$$
$$E \geq \text{energía}(M[\text{Pos} + (x, y)]) \tag{9}$$

(1)(2) La nueva posición $(\text{Pos} + (x, y))$ está dentro del mapa

(3) La nueva posición $(\text{Pos} + (x, y))$ se puede transitar

(4) La ambulancia tiene la suficiente energía para atravesar la casilla

Los efectos son los siguientes:

- Se copian los valores del estado antes de aplicar el operador
- $\text{Pos} = \text{Pos} + (x, y)$
- $E = E - \text{energía}(M[\text{Pos} + (x, y)])$
- Si $M[\text{Pos} + (x, y)] = CN \Rightarrow P_N = 0$
- Si $M[\text{Pos} + (x, y)] = CC \Rightarrow P_C = 0$
- Si $M[\text{Pos} + (x, y)] = P \Rightarrow E = E_0$
- Si $M[\text{Pos} + (x, y)] = N$, el paciente se puede recoger $(P_C = 0, P_N \leq P_T)$, y no está marcado en Visitados $\Rightarrow P_N = P_N + 1$ y se marca el paciente en Visitados
- Si $M[\text{Pos} + (x, y)] = C$, el paciente se puede recoger $(P_C < P_{T_C}, P_N \leq P_T - P_{T_C})$, y no está marcado en Visitados, $\Rightarrow P_C = P_C + 1$ y se marca el paciente en Visitados

El coste del operador es $\text{energía}(M[\text{Pos} + (x, y)])$

Heurísticas

Para las heurísticas definidas se ha usado la distancia de Manhattan para estimar la distancia entre dos posiciones dadas. Además, ambas heurísticas relajan las precondiciones (3) y (4), y las condiciones sobre cuando se puede recoger un paciente.

La primera heurística es el coste total de recoger al paciente no recogido más lejano (restringido a los pacientes contagiosos si la ambulancia ya tiene pacientes contagiosos), ir al centro de pacientes contagiosos (si es necesario), ir al centro de pacientes no contagiosos (si es necesario), y finalmente volver al parking.

Esta heurística es admisible porque estos pasos siempre se tendrán que hacer en orden al menos una vez al final. Además, las condiciones para ir a los centros de pacientes garantizan que el coste de sus respectivos pasos solo se añade si el paso realmente es necesario, garantizando que nunca se sobrestima el coste real.

La segunda heurística es una modificación de la primera. En el primer paso, si la ambulancia sí tiene algún paciente contagioso, además de recoger al paciente contagioso no recogido más lejano, también se añade el coste de ir al centro de pacientes contagiosos y recoger al paciente no contagioso no recogido

más lejano. Además, en este caso no se volvería a ir al centro de pacientes contagiosos, asumiendo que ya se han entregado todos los pacientes contagiosos en el primer viaje.

Si la ambulancia tiene algún paciente contagioso, tendrá que pasar por el centro de pacientes contagiosos antes de recoger a algún otro paciente no contagioso. Por lo tanto, añadir en estos casos a la heurística original una estimación de este coste que siempre subestima el real no hace que deje de ser admisible. Además, como esta heurística es la primera con un sumando extra, está más informada.

3 Análisis de resultados

3.1 Problema 1

Resultado Obtenido

Según nuestra implementación en Python con la librería `Python-Constraints` de este problema, al ejecutarlo con un archivo de ejemplo (ejemplo del enunciado), se obtienen 2175288 posibles soluciones.

Este dato es imposible de comprobar ya que se tardaría mucho tiempo en realizar las combinaciones de todas las casillas a mano. Para demostrar que las soluciones que arroja nuestro programa son las necesarias, se realizarán una serie de tests desarrollados en `bash`.

Definición de casos de prueba

A la hora de realización de los tests mencionados, se han tenido en cuenta los siguientes casos de uso para los diferentes inputs del programa:

1) Parking con mismo número de posiciones eléctricas que vehículos con congelador (Posiciones eléctricas consecutivas).
2) Parking con más plazas eléctricas de las necesarias. Entre las soluciones disponibles de este caso, se encontrarán algunas en las que un vehículo sin congelador por lo menos haya ocupado una plaza eléctrica.
3) Parking con mismo número de plazas eléctricas que de vehículos con congelador (Posiciones eléctricas no consecutivas).
4) Parking con grupos de tres plazas eléctricas consecutivas. La finalidad es testear la restricción que verifica la maniobrabilidad en el parking.
5) Parking con posiciones consecutivas en los bordes. Testeo de restricciones en las filas primera y última.

Planteamiento de tests

Los archivos generados como *"Inputs"* de nuestro programa `CSPParking.py` se crean de la siguiente manera:

1) Para el primer caso de prueba, se definirá un parking de tamaño **2x1**. En dicho parking las dos casillas son eléctricas y solo entrarán al parking 2 vehículos con congelador. En este problema, la restricción de tipos se ve relajada debido a que no hace efecto en un parking de una columna. La restricción de maniobrabilidad también lo está debido a que se necesita al menos un trio para poder tener influencia. La restricción que se prueba entonces es la de consecutividad en los bordes. Es por eso que, al no poder salir los coches del parking fácilmente al haber aparcado, no existen soluciones para este problema.

2) En el segundo caso de prueba, se creará un parking de tamaño **5x6**. En este, se habilitarán cuatro casillas eléctricas no consecutivas en los bordes de las treinta disponibles. También las

ambulancias que necesitarán aparcar en dicho parking serán tres con congelador y de distintos tipos de urgencia. Las restricciones que tienen efecto en las soluciones son las de los bordes (en las soluciones, no saldrán las que tengan una casilla contigua a las eléctricas ocupadas), la de posiciones consecutivas (no puede haber un trio de posiciones consecutivas en ninguna solución) y la restricción de tipos (siempre los vehículos urgentes tendrán a los no urgentes detrás suya y no al revés).

3) Para el tercer caso de prueba, se construirá un parking de dimensiones **3x3**. Para este, las posiciones eléctricas serán las cuatro esquinas del cuadrado. Cuatro vehículos eléctricos querrían ocupar su plaza en el aparcamiento (uno de tipo urgente y tres no urgentes). Ya que las posiciones eléctricas no son consecutivas mutuamente y no hay más ambulancias que las eléctricas, la única restricción que tiene efecto sobre las soluciones es la de los tipos (urgente y no urgente). Entre las soluciones, se debería de observar que no existen soluciones que no tengan al vehículo urgente (vehículo con id de 1) en la esquina superior derecha o inferior derecha.

4) En este caso de prueba, se edificará un parking de tamaño **6x4**. Para probar la maniobrabilidad de las posiciones consecutivas, se habilitarán seis posiciones eléctricas organizadas en trios consecutivos (concretamente en la columna 2 y 4). Para este escenario, seis ambulancias eléctricas de tipo no importante van a ingresar al parking. Dado que todos los vehículos son eléctricos y obligatoriamente tienen que estar en plazas eléctricas consecutivas, el algoritmo no dará soluciones válidas ya que también se restringen las posiciones consecutivas por otro lado.

5) Para el último caso de prueba, se definirá un aparcamiento de tamaño **3x3**. En él, no habrá trios de posiciones consecutivas pero si posiciones eléctricas consecutivas en los bordes de dicho cuadrado (seis posiciones). Todos los seis vehículos que desean aparcar son eléctricos y forzosamente deben estar en posiciones eléctricas, que a la vez están consecutivas, luego el algoritmo de resolución no dará soluciones válidas al igual que en el anterior caso de prueba.

Comprobación de soluciones

Para el método de comprobación de las soluciones obtenidas, los resultados siguientes son los esperados:

1) Los casos de prueba 1, 4 y 5 no deben tener soluciones válidas para los archivos generados.

2) El caso de prueba 2 debería dar $4! - 4 * 2 = 16$ soluciones. Esto se puede comprobar ya que 4! soluciones son las posibles combinaciones de 4 elementos en 4 plazas de parking. Dado que hay dos vehículos que son de tipo urgente, cada uno resta al total de soluciones 4 que no se pueden dar debido a la restricción de tipos.

3) El caso de prueba 3 debería arrojarnos $\frac{4!}{2}$ soluciones debido a que hay 4! posibles maneras de permutar 4 vehículos en 4 posibles plazas del mismo. Como la mitad de las soluciones no respetan la restricción de tipo dejando a la ambulancia urgente en las esquinas superior izquierda e inferior izquierda, dichas soluciones se descartan de las finales (cumpliendo así todas las restricciones).

Conclusiones de los tests

Dado que los resultados obtenidos por los tests son los esperados, los tests se han ejecutado correctamente, probando así que nuestro programa es correcto.

3.2 Problema 2

Resultado obtenido

Para el problema dado, el programa encuentra 2 soluciones óptimas con coste 88 función de la heurística usada. Para la primera heurística tiene una longitud del plan de 85 pasos, mientras que para la segunda tiene una longitud de 83 pasos

Casos de prueba

Los casos de prueba implementados son los siguientes:

1) Mapa lineal con un único paciente contagioso entre el parking y el centro de pacientes contagiosos. La solución óptima será ir al centro de pacientes contagiosos y volver
2) Mapa lineal con un único paciente no contagioso entre el parking y el centro de pacientes contagiosos. La solución óptima será ir al centro de pacientes no contagiosos y volver
3) Mapa lineal con un paciente no contagioso seguido de un paciente contagioso, seguido del centro de pacientes contagiosos y de pacientes no contagiosos. La solución óptima será ir al centro de pacientes no contagiosos y volver
4) Mapa lineal con un paciente no contagioso seguido de un paciente contagioso, seguido del centro de pacientes no contagiosos y de pacientes contagiosos. La solución óptima será ir al centro de pacientes contagiosos y volver, y se entregará al paciente no contagioso a la vuelta
5) Mapa lineal con un paciente contagioso seguido de un paciente no contagioso, seguido del centro de pacientes no contagiosos y de pacientes contagiosos. La solución óptima será ir al centro de pacientes contagiosos, volver a por el paciente no contagioso, ir al centro de pacientes no contagiosos, y volver al parking
6) Mapa lineal con un paciente contagioso seguido de un paciente no contagioso, seguido del centro de pacientes contagiosos y de pacientes no contagiosos. La solución óptima será ir al centro de pacientes contagiosos, volver a por el paciente no contagioso, ir al centro de pacientes no contagiosos, y volver al parking
7) Mapa en el que los pacientes/centros de pacientes están separados del parking por una casilla no transitable. No hay solución
8) Mapa con un paciente no contagioso seguido de 2 caminos con diferente longitud al centro de pacientes no contagiosos. La solución óptima será ir y volver del centro por el camino más corto
9) Mapa lineal con un único paciente no contagioso entre el parking y su centro, en el cual hay la energía justa para ir a su centro y volver. La solución óptima es ir al centro del paciente y volver
10) Mapa lineal en el cual el parking está entre un paciente no contagioso y su centro, y solo hay energía suficiente para llegar al centro desde el paciente si se recarga en el parking. La solución óptima es recoger al paciente, entregarlo en su centro, y volver al parking
11) Mapa con un único paciente no contagioso en el cual hay energía suficiente para ir y volver desde el parking al paciente o a su centro, pero no para realizar ambos en un único trayecto. La solución óptima es recoger al paciente no contagioso, volver al parking, ir a su centro, y volver al parking
12) Mapa lineal con más pacientes no contagiosos de los que puede llevar la ambulancia entre el parking y su centro. La solución óptima es ir a su centro, volver a por los restantes, volver a ir a su centro, y volver al parking
13) Mapa lineal con suficientes pacientes no contagiosos como para tener que usar las plazas de pacientes contagiosos pero no como para llenar la ambulancia, seguidos de un paciente contagioso y los centros de los pacientes. La solución óptima es ir al centro de pacientes no contagiosos, volver a por el paciente contagioso, ir al centro de pacientes contagiosos, y volver al parking
14) Mapa lineal con más pacientes contagiosos de los que puede llevar la ambulancia entre el parking y su centro. La solución óptima es ir a su centro, volver a por los restantes, volver a ir a su centro, y volver al parking

Tras ejecutar el programa con todos los casos de prueba, este obtiene siempre la solución óptima si existe

Rendimiento

Debido a la similitud de las heurísticas, en muchos de los casos de prueba el resultado es el mismo. Sin embargo, en los casos en los que la modificación de la segunda heurística se usa, esta es capaz de expandir significativamente menos nodos.

Para un caso complejo como el problema dado, la primera heurística necesita expandir ~ 174 millones de estados, mientras que la segunda es capaz de expandir únicamente ~ 87 millones

4 Conclusión

Para esta práctica, hemos aprendido los diferentes conceptos:

- Representar problemas en `Python`
- Modelado de problemas reales
- Representado de las restricciones del problema (funciones lambda)
- Resolver problemas de satisfacción de restricciones con la librería `Python-Constraints`
- Extraer información de archivos
- Volcar resultados de algoritmos en archivos de salida
- Implementación de A^* en Rust
- Comparación de distintas heurísticas admisibles y sus efectos en la resolución del problema
- Resolución de problemas reales con algoritmos de búsqueda

Se puede concluir que en el tiempo de realización de la práctica, se ha seguido una curva de aprendizaje bastante razonable y se han aprovechado conceptos del lenguaje `Python` adquiridos en cursos anteriores.