



## Memoria Práctica 1 Heurística y Optimización

Álvaro Guerrero Espinosa - 100472294  
Adrian Cortázar - 100475860

# Índice

1	Introducción . . . . .	2
2	Descripción de los modelos . . . . .	2
2.1	Problema 1 . . . . .	2
	Representación del problema . . . . .	2
	Modelado de Restricciones . . . . .	2
2.2	Problema 2 . . . . .	3
	Parámetros globales . . . . .	3
	Estado . . . . .	3
	Operadores . . . . .	4
	Heurísticas . . . . .	4
3	Análisis de resultados . . . . .	6
3.1	Problema 1 . . . . .	6
	Resultado Obtenido . . . . .	6
	Casos de prueba . . . . .	6
3.2	Problema 2 . . . . .	7
	Resultado obtenido . . . . .	7
	Casos de prueba . . . . .	7
	Rendimiento . . . . .	8
4	Conclusión . . . . .	9

# 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:

$$x_a \neq x_b \quad \forall a, b \in V, a \neq b \quad (1)$$

$$(x_{ay} \neq x_{by}) \vee (x_{ax} > x_{bx}) \quad \forall a \in V_U, b \in V_N \quad (2)$$

$$\neg(x_{ax} = x_{bx} = x_{cx}) \vee \min(x_{ay}, x_{by}, x_{cy}) + 2 \neq \max(x_{ay}, x_{by}, x_{cy}) \quad \forall a, b, c \in V, a \neq b \neq c, a \neq c \quad (3)$$

$$x_{ax} \neq x_{bx} \vee \min(x_{ay}, x_{by}) + 1 \neq \max(x_{ay}, x_{by}) \quad \forall a, b \in V, a \neq b \quad (4)$$

$$\vee 1 \neq \min(x_{ay}, x_{by}) \neq n - 1 \quad (5)$$

- (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. La expresión  $\min(N \text{ valores}) + (N-1) = \max(N \text{ valores})$  se usa para comprobar si estos valores son consecutivos o no suponiendo que son distintos, lo cual siempre ocurre en los casos en los que se evalúa debido a la restricción (1) y la comparación de igualdad anterior
- (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 si hay algún vehículo

## 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.
- $P_{TC} \in \mathbb{N}$ : 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.
- $M_{ij} \in \text{Casillas}$  ( $i, j \in \mathbb{N}, i < N, j < M$ ): mapa del problema, donde cada elemento indica el contenido de la casilla correspondiente.
- energía : Casillas  $\rightarrow \mathbb{N}$ : 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:  $\text{move}(x, y)$ . 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:

$$0 \leq \text{Pos}_x + x < N \quad (1)$$

$$0 \leq \text{Pos}_y + y < M \quad (2)$$

$$M[\text{Pos} + (x, y)] \neq X \quad (3)$$

$$E \geq \text{energía}(M[\text{Pos} + (x, y)]) \quad (4)$$

(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 ( $P_C < P_{TC}, P_N \leq P_T - P_{TC}$ ), 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.

La primera heurística consiste en el coste máximo de la distancia estimada entre cada paciente no recogido y su centro, y este centro y el parking. Además, en el caso de no quedar pacientes sin recoger, usa la distancia al centro de pacientes contagiosos, de este al centro de pacientes no contagiosos, y de ahí al parking (saltándose los viajes a los centros si la ambulancia no tiene pacientes de su respectivo tipo). Esto relaja las precondiciones (3) y (4), las restricciones de la capacidad del vehículo, y la restricción de recoger a todos los pacientes (ya que asume que el paciente con la distancia calculada máxima es el único paciente restante). Como esta heurística se obtiene por relajación de restricciones, es *admisibile*.

La segunda heurística es una modificación de la primera. En esta, se le añade a la distancia calculada de cada paciente la distancia a este desde la posición actual de la ambulancia. Esta heurística también es *admisibile* por el mismo motivo que la anterior, ya que relaja las mismas restricciones. Además, como

la distancia estimada entre la posición de la ambulancia y la posición de cualquier paciente no puede ser negativa, esta heurística es *más informada* que la primera.

## 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`.

#### Casos de prueba

Los casos de prueba implementados son los siguientes:

- 1) Parking de **1x2** con una única ambulancia. No hay solución ya que no se puede cumplir la restricción (5)
- 2) Parking de **2x1** con 2 vehículos eléctricos y 2 plazas eléctricas en la misma columna. No hay solución ya que no se puede cumplir la restricción (4)
- 3) Parking de **3x1** con 2 vehículos eléctricos y 2 plazas eléctricas en la misma columna al lado del borde superior del parking. No hay solución ya que no se puede cumplir la restricción (4)
- 4) Parking de **3x1** con 2 vehículos eléctricos y 2 plazas eléctricas en la misma columna al lado del borde inferior del parking. No hay solución ya que no se puede cumplir la restricción (4)
- 5) Parking de **4x1** con 3 vehículos eléctricos y 3 plazas eléctricas seguidas en la misma columna. No hay solución ya que no se puede cumplir la restricción (3)
- 6) Parking de **3x1** con 4 vehículos. No hay solución ya que no se puede cumplir la restricción (1)
- 7) Parking de **3x1** filas con 3 vehículos eléctricos y 2 plazas eléctricas. No hay solución ya que no se puede asignar a cada vehículo eléctrico una plaza eléctrica distinta.
- 8) Parking de **2x2** con un vehículo eléctrico TSU, un vehículo no eléctrico TNU, y 1 plaza eléctrica en la primera columna. Hay una única solución en la que el vehículo TNU está en la esquina opuesta al TSU debido a la restricción (2)
- 9) Parking de **2x2** con un vehículo eléctrico TSU, un vehículo no eléctrico TSU, y 1 plaza eléctrica en la primera columna. Hay 2 soluciones en la cual el vehículo eléctrico está en la plaza eléctrica y el no eléctrico está en la segunda columna debido a la restricción (4)
- 10) Parking de **2x2** con un vehículo eléctrico TSU, un vehículo no eléctrico TNU, y 1 plaza eléctrica en la segunda columna. Hay 2 soluciones en las cuales el vehículo eléctrico está en la plaza eléctrica y el no eléctrico está en la primera columna debido a la restricción (4)
- 11) Parking de **3x3** con 2 vehículos eléctricos TNU y 3 plazas eléctricas en la diagonal principal. Hay 6 soluciones correspondientes a cada asignación de los vehículos a las plazas eléctricas.
- 12) Parking de **2x2** con 2 vehículos no eléctricos TNU y 2 plazas eléctricas en la diagonal principal. Hay 8 soluciones correspondientes a cada asignación de vehículos a las plazas con los vehículos en columnas distintas.
- 13) Parking de **3x2** con 1 vehículo eléctrico TSU, 2 vehículos no eléctricos TNU, y 2 plazas eléctricas en primera columna. Hay 6 soluciones en total: 2 correspondientes al vehículo eléctrico en la plaza eléctrica central y los vehículos no eléctricos en las esquinas de la segunda columna, y 4 correspondientes al vehículo eléctrico en la plaza eléctrica del borde, y uno de los vehículos no eléctricos en el borde opuesto de la primera columna y el otro en una de las 2 plazas de la segunda columna cuya fila es distinta a la del vehículo eléctrico.

Tras ejecutar el programa con todos los casos de prueba, este obtiene siempre las soluciones esperadas, luego el programa es correcto.

## 3.2 Problema 2

### Resultado obtenido

Para el problema dado con un mapa de **10x10**, el programa no termina su ejecución por problemas de memoria. Por esto, se ha decidido usar una versión reducida de este mapa con un tamaño de **8x10** (caso de prueba 17). Para este problema, el programa encuentra una solución óptima de coste 79 y longitud del plan 73.

### 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 con un único paciente no contagioso en el cual hay energía suficiente para ir al paciente y llegar a su hospital, pero no para volver al parking desde el paciente ni desde su centro. No hay solución ya que no se puede volver al parking.
- 13) 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.
- 14) 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.



- 15) 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.
- 16) Mapa cuadrado de **5x5** con el parking en el centro, 8 pacientes no contagiosos a la misma distancia del parking (formando un rombo), y su centro en una esquina. La solución óptima es ir a uno de los pacientes en el lado más cercano a este, dar una vuelta para recoger a todos los pacientes, ir a su centro, y volver al parking.
- 17) Mapa reducido del problema dado, con un tamaño de **8x10** en vez de **10x10**. La cantidad de pacientes de cada tipo es la misma al original, pero se han eliminado las 2 últimas filas y movido los pacientes de estas a los huecos restantes.

Tras ejecutar el programa con todos los casos de prueba, se ha verificado que este obtiene siempre la solución óptima (en los casos en los que existe una solución), luego el programa es correcto. Cabe destacar que, para el caso de prueba 17, como este es un problema complejo, no se ha podido verificar que la solución obtenida realmente es óptima. Sin embargo, al analizar la solución no hemos encontrado ninguna forma de mejorarla, y teniendo en cuenta que la heurística es admisible sí que debería ser óptima.

## Rendimiento

Debido a la similitud de las heurísticas y simpleza de los casos de prueba, en muchos de estos el rendimiento (medido como la cantidad de estados expandidos) es el mismo o casi el mismo, con una diferencia de unos pocos estados como mucho. Sin embargo, a medida que el coste de la solución óptima aumenta (particularmente en los casos de prueba 10 y 11), la primera heurística pasa a expandir aproximadamente un 50% menos de estados que la segunda heurística, lo cual es esperable dado que la segunda heurística está más informada.

El caso extremo es el caso de prueba 16, en el cual la primera heurística necesita expandir 15565 estados mientras que la segunda solo expande 2765. Para un caso más complejo como el caso de prueba 17, la primera heurística necesita expandir  $\sim 109$  millones de estados para encontrar la solución óptima, mientras que la segunda es capaz de expandir únicamente  $\sim 44.2$  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 y rendimiento del problema.
- Dificultad de encontrar buenas heurísticas admisibles, y de verificar si una heurística es admisible cuando no se obtiene exclusivamente por relajación de restricciones.
- 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.