Memoria Práctica 1

Práctica Programación Lineal. Heurística y Optimización

- **Pablo Pino Castillo, 100522129**
- **Alejandro Ros Quesada, 100522331**

Parte 1. Modelización

Introducción

Se modela un problema básico de asignación entre autobuses y talleres, con el objetivo de minimizar la distancia total recorrida.
Cada autobús debe ser asignado a un único taller, y cada taller sólo puede recibir un autobús.

Decisión de diseño:

Se definieron variables binarias x_{ij} para representar la asignación directa entre autobuses y talleres.
Las restricciones se plantearon de forma simétrica para garantizar la exclusividad en ambas direcciones (un autobús $\leftrightarrow$ un taller).
La función objetivo se formuló como una suma ponderada de distancias, lo que permite minimizar el recorrido total de todos los autobuses.

Parámetros

Los datos que se deben introducir por el usuario en nuestro modelado serán los siguientes:

- $c_{i,j}$ será la distancia de los buses con cada taller, siendo el autobús j y el taller i

Variables de decisión

$$x_{ij}(i, j \in 1, \ldots, 5)$$

Estas variables son **binarias**, es decir:

- $x_{i,j} = 1$ si el autobús j es asignado al taller i
- $x_{i,j} = 0$ en caso contrario

En total hay $5 \cdot 5 = 25$ variables de decisión

Restricciones

1. Cada autobús debe tener un taller asignado

$$\sum_{i=1}^{5} x_{ij} = 1, \ \forall j$$

> No puede haber un autobús sin un taller asignado, al ser alguno de ellos asignados la variable xij que sea asignada se pondrá como 1, lo que obligará a que ese autobús j se asigne a cualquier otro taller i

2. Cada taller solo puede recibir un autobús

$$\sum_{j=1}^{5} x_{ij} = 1, \ \forall i$$

> Solamente puede estar un autobús asignado a cada taller, al ser algún autobús asignado a un taller se pondrá la pertinente $x_{ij} = 1$, lo que bloqueará a los demás autobuses y no dejara que en ese taller (i) haya ningún otro autobús asignado.

En total hay 10 restricciones.

Función objetivo

Minimizamos el coste total:

$$\min z = \sum_{i=1}^{5} \sum_{j=1}^{5} c_{ij}\ x_{ij}$$

- Si $x_{ij} = 1$, entonces c_{ij} se suma al total
- Si $x_{ij} = 0$, entonces $c_{ij} \cdot 0 = 0$, por lo que no contribuye a la función objetivo

---

Parte 2.1. Modelización

Introducción

Ampliamos el modelo anterior considerando un único taller que dispone de varias franjas horarias.
El objetivo es minimizar el coste total, combinando el coste de desplazamiento de los autobuses asignados con la penalización por los que no pueden ser atendidos.

Decisión de diseño:

Se añadieron variables auxiliares y_j para representar los autobuses que no son atendidos por falta de franjas, lo que permite incluir penalizaciones por pasajero no asignado.

Parámetros

Los datos que se deben introducir por el usuario en nuestro modelado serán los siguientes:

- k_p: coste por kilómetro de asignar un autobús a una franja (valor único).
- p_j: número de pasajeros del autobús j.
- k_d: penalización por cada pasajero de un autobús que no se asigna a ninguna franja (valor único).
- d_j: distancia del autobús j.
- $Franja$: Número de franjas.
- $Autobús$: Número de autobuses.

Variables de decisión

1. Variable de decisión principal

$$x_{j,s}(j \in 1 \dots m,\ s \in 1 \dots n)$$

2. Variable de decisión auxiliar (utilizada para conocer el número de autobuses sin asignar)

$$y_j(j \in 1 \dots m)$$

Estas variables son **binarias**, es decir:

- $x_{j,s} = 1$ si el autobús j es asignado a la franja s // $x_{j,s} = 0$ en caso contrario
- $y_j = 1$ si el autobús j es asignado a alguna franja // $y_j = 0$ en caso contrario

---

Restricciones

1. Cada franja sólo puede tener un autobús

$$\sum_{j=1}^{m} x_{j,s} \leq 1,\ \forall s \in (1, \dots, n)$$

> Cómo máximo, solo se puede asignar un único autobús a cada franja (1 franja, 1 autobús), por lo que en una parte del sumatorio lsi a franja s contiene ya un autobús , entonces se bloqueará y no podra tener más autobuses en esa franja

2. Cada autobús sólo puede estar en una franja

$$\sum_{s=1}^{n} x_{j,s} + y_j = 1,\ \forall j \in (1, \dots, m)$$

> Un autobús solo puede estar asignado a una franja como máximo (1 autobús, 1 franja), por lo que si la variable $x_{j,s}$ no está asignada en ninguna franja (el autobús no es asignado) la variable y_j será obligatoriamente 1

Función objetivo

Minimizamos el coste:

$$\min z = \sum_{j=1}^{m} k_p \cdot p_j \cdot (1 - \sum_{s=1}^{n} x_{j,s}) + k_d \cdot d_j \cdot \sum_{s=1}^{n} x_{j,s}$$

Que simplificando nos queda:

$$\sum_{j=1}^{m} \sum_{s=1}^{n} (k_d \cdot d_j \cdot x_{j,s}) + \sum_{j=1}^{m} k_p \cdot p_j \cdot y_j$$

> El coste de los autobuses asignados ($\sum_{j=1}^{m}\sum_{s=1}^{n}(k_d \cdot d_j \cdot x_{j,s})$) más el coste de los autobuses no asignados ($\sum_{j=1}^{m} k_p \cdot p_j \cdot y_j$)

Parte 2.2. Modelización

Introducción

Ahora abordamos un modelo más general con varios talleres, cada uno con diferentes franjas de disponibilidad.
El objetivo es minimizar el impacto sobre los pasajeros cuando dos autobuses coinciden en la misma franja, incluso si se encuentran en talleres distintos.

Decisión de diseño:

Se incorporaron variables adicionales $z_{i,s}$ y $y_{i,j,s}$ para modelar coincidencias de autobuses en la misma franja, incluso si están en talleres distintos.
Las relaciones entre estas variables se linealizaron mediante desigualdades.
Este enfoque garantiza la compatibilidad con la programación lineal entera y permite cuantificar el impacto de los conflictos entre autobuses de forma precisa.

Parámetros

Los datos que se deben introducir por el usuario en nuestro modelado serán los siguientes:

- $O_{s,t}$ disponibilidad de franja s en taller el t: 1 si la franja está disponible, 0 si no está disponible.
- $c_{i,j}$ número de pasajeros que utilizan ambos autobuses i y j.
- $Franjas$: Número de franjas.
- $Autobuses$: Número de autobuses.
- $Taller$: Número de autobuses.

Variables de decisión

1. Autobús en un taller y en una franja horaria

$$x_{i,t,s}(i \in 1 \dots m,\ t \in 1 \dots u,\ s \in 1 \dots n)$$

2. Dos autobuses en la misma franja horaria (aun siendo de distintos talleres)

$$y_{i,j,s}(i,j \in 1 \dots m,\ s \in 1 \dots n)$$

3. Autobús en una franja horaria

$$z_{i,s}(i \in 1 \dots m,\ s \in 1 \dots n)$$

- $i,\ j \rightarrow$ Autobús
- $t \rightarrow$ Taller
- $s \rightarrow$ Franja horaria

Restricciones

1. Disponibilidad del taller en una franja horaria

$$x_{i,t,s} \leq O_{s,t},\ \forall t, s$$

> Un autobús sólo puede estar en una franja horaria de un taller si dicha franja está disponible, por lo cual si $O_{s,t}$ es 0 (no disponible) entonces $x_{i,t,s}$ no podrá asignarse

2. Cada autobús solo puede estar asignado una vez

$$\sum_{i=1}^{m} x_{i,t,s} \leq 1,\ \forall t, s$$

> Un autobús sólo puede estar asignado a una franja horaria de un taller, por lo que si algún autobús se asigna en alguna franja de algún taller se bloqueará y no podrá asignarse más veces

3. Cada autobús debe tener un taller asignado

$$\sum_{t=1}^{u} \sum_{s=1}^{n} x_{i,t,s} = 1,\ \forall i$$

> Todos los autobuses deben ser asignados a una franja (y solo una) de algún taller, por lo que una vez que se asigne a una franja de un taller se bloqueará a las demás, pero obligatoriamente a una

Para evitar operadores no lineales, las condiciones del tipo “dos autobuses coinciden en la misma franja” se representaron mediante un conjunto de tres restricciones lineales (acotaciones superior e inferior), garantizando que el modelo cumpla con los requisitos de la programación lineal entera. (conjunción lógica AND):
4. Acotación superior 1

$$z_{i,s} \geq y_{i,j,s}$$

> Si $y_{i,j,s} = 1$ (el conflicto se activa), obliga a que el autobús i esté asignado a la franja s

5. Acotación superior 2

$$z_{j,s} \geq y_{i,j,s}$$

> Si $y_{i,j,s} = 1$ (el conflicto se activa), obliga a que el autobús j esté asignado a la franja s

6. Acotación inferior

$$y_{i,j,s} \geq z_{i,s} + z_{j,s} - 1$$

> Si ambos autobuses (i, j) están en la franja s, entonces $y_{i,j,s} \geq 2 - 1$, forzando a que $y_{i,j,s}$ sea 1.

Función objetivo

Minimizamos el impacto que tiene que dos autobuses coincidan en la misma franja:

$$\sum_{i=1}^{m} \sum_{j=1}^{m} \sum_{s=1}^{n} c_{i,j}\ y_{i,j,s},\ (j > i)$$

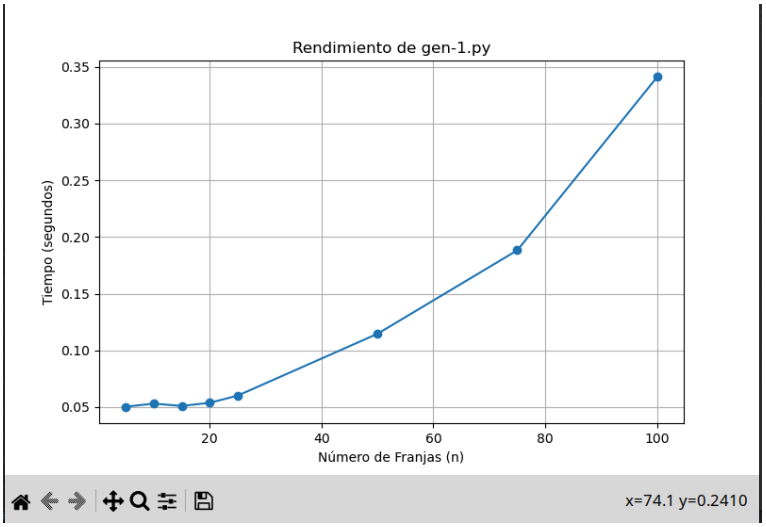
> Si dos autobuses coinciden en la misma franja aun estando en distintos talleres, esta función busca minimizar el número de pasajeros que han contratado los buses i-ésimos y j-ésimos.

Análisis de resultados

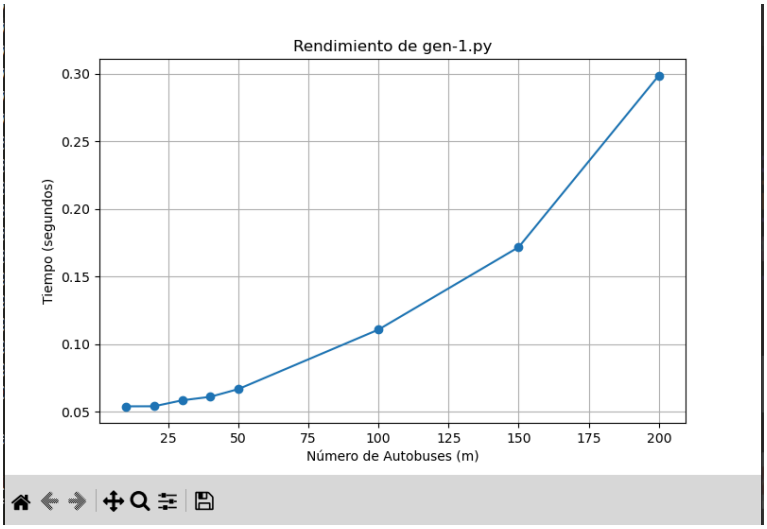
Para analizar correctamente los modelos que hemos propuesto para solucionar los distintos problemas, hemos creado dos scripts que nos permiten generar archivos de entrada aleatorios, permitiéndonos variar los valores de los parámetros especificados en el apartado anterior, cambiando así las dimensiones del problema.

Pruebas Parte 2.1

1. Al aumentar considerablemente el número de franjas ofertadas, se observa como hay un crecimiento en el tiempo de ejecución pero no es una diferencia realmente preocupante, esto se debe a que la modelización del problema que hemos hecho no es restrictiva en cuanto a que no busca que todos los autobuses tengan que estar asignados.



2. Incrementando el número de autobuses ocurre lo mismo que en el caso anterior, los tiempos aumentan pero de una manera razonable, no hay ningún pico muy grande ya que no es un problema especialmente restrictivo.



3. En este caso, hemos probado a introducir un número mucho mayor de autobuses que de franjas, por lo que forzamos a una gran cantidad de estos autobuses a quedarse sin asignar. Esto provoca que la penalización en la función objetivo que supongan sea mucho mayor, resultando en un valor objetivo increíblemente alto.

```
Duración de la resolución: 0.0662 segundos

--- DETALLES DEL CASO (n=100, m=300) ---
Valor Objetivo: 631813.5 | Variables: 30300 | Restricciones: 401

Asignaciones (Autobús -> Franja):
  - Autobús 1 asignado a franja 60
  - Autobús 2 asignado a franja 44
  - Autobús 4 asignado a franja 29
  - Autobús 7 asignado a franja 85
  - Autobús 8 asignado a franja 80
  - Autobús 14 asignado a franja 36
  - Autobús 19 asignado a franja 12
  - Autobús 21 asignado a franja 7
  - Autobús 24 asignado a franja 76
  - Autobús 28 asignado a franja 67
  - Autobús 29 asignado a franja 17
```

Conclusiones

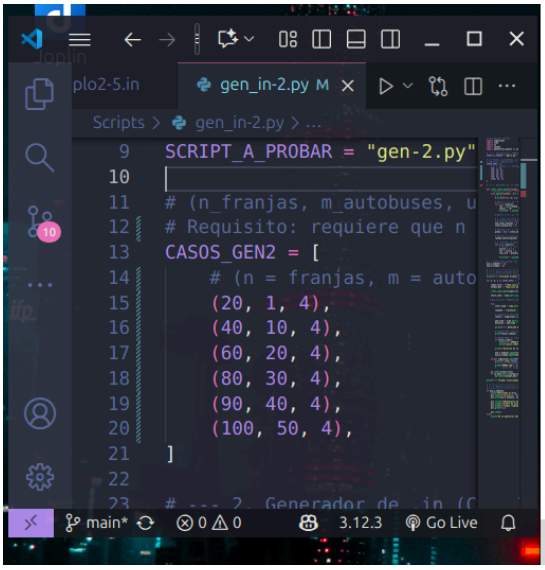
Este modelo logra minimizar eficazmente el coste de desplazamiento y penalizaciones por no ser asignado.
Podemos observar que el número de franjas y autobuses influye directamente en el tiempo de

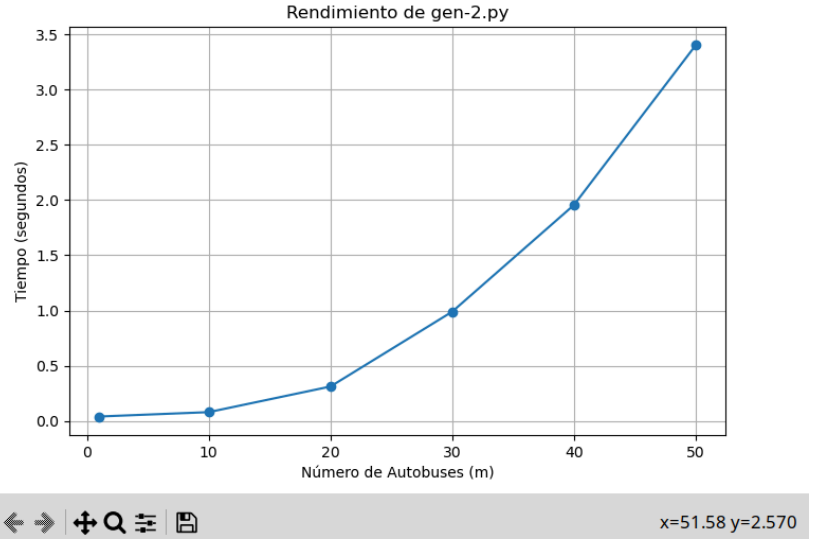
resolución, pero el solver aun asi maneja bien el crecimiento del problema.
Cuando el número de autobuses supera al número de franjas, el modelo responde de forma correcta aumentando la penalización total, lo que refleja el impacto de los buses no asignados en el coste_total.
Por lo que el modelo es robusto y escalable en distintos escenarios

---

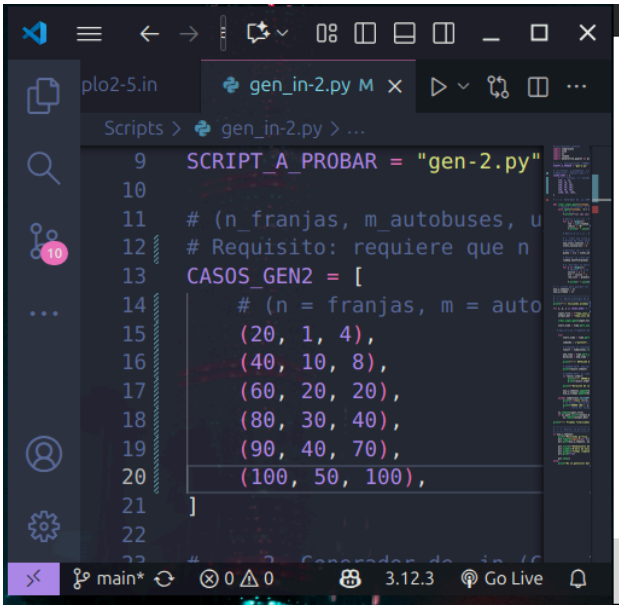
Pruebas Parte 2.2

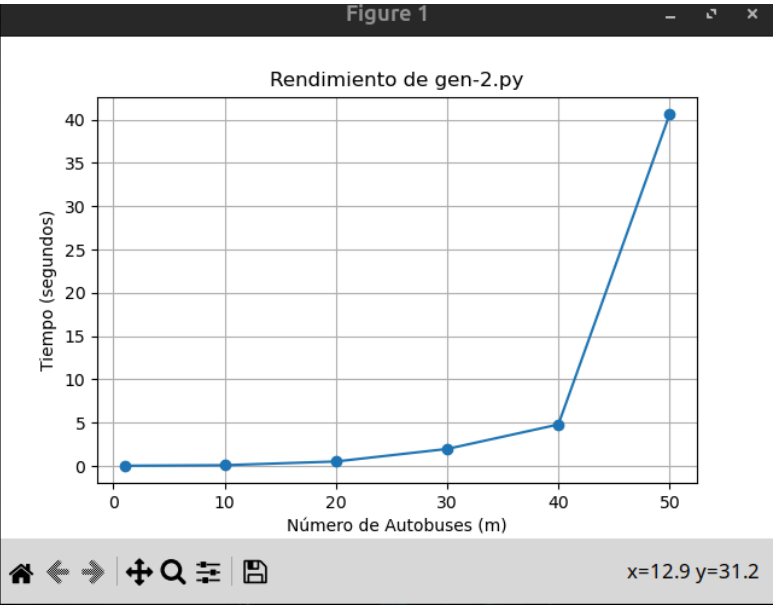
1. Como primera prueba hemos decidido aumentar las franjas junto con el número de autobuses, pero vemos que pese a aumentar en gran cantidad el número de autobuses, si el número de franjas sigue acorde, al solucionador le va a costar bastante poco trabajo encontrar la solución óptima a estos problemas (ya que el valor óptimo va a ser 0).





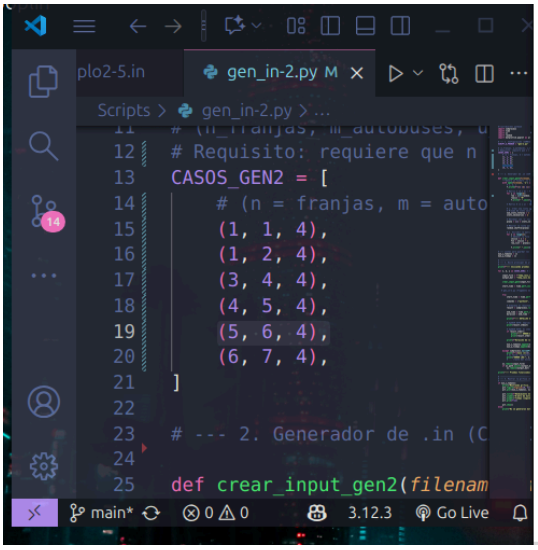
2. Si al aumentar las franjas y los autobuses, ahora también aumentamos los talleres, observamos que hay una mayor diferencia de tiempo respecto al caso anterior, ya que con 50 autobuses pasamos de una ejecución en $3,5\ s$, a una ejecución de unos $40\ s$

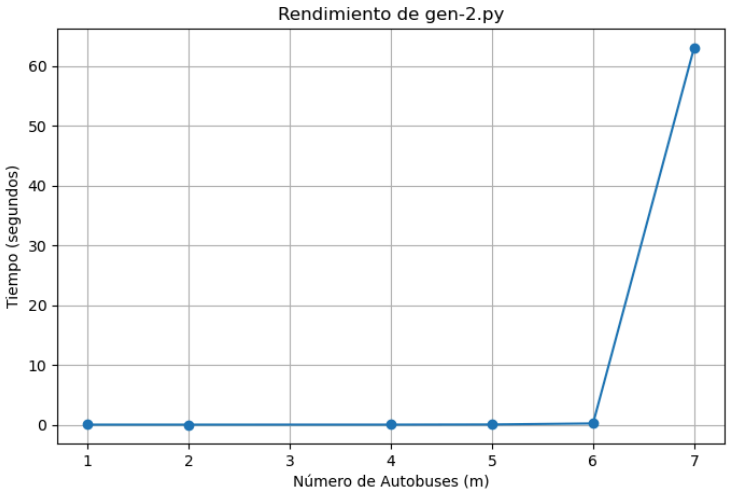




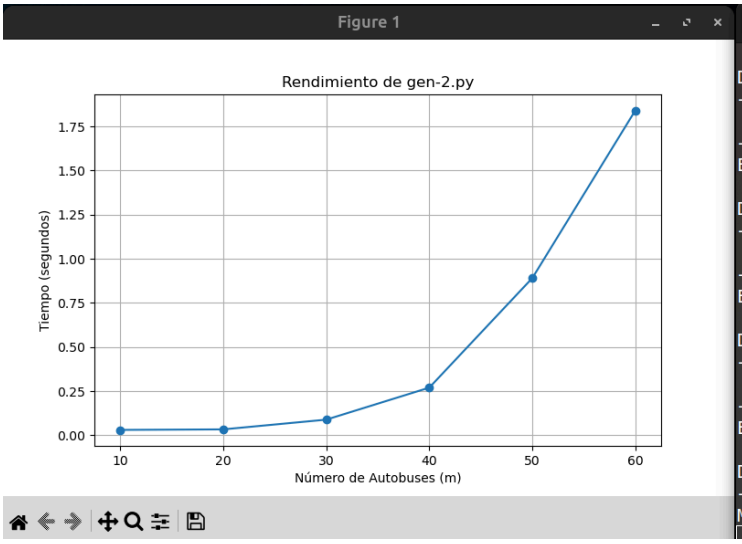
3. Este caso es el más impactante, ya que aun habiendo un número no muy elevado de autobuses, vemos que el incremento en tiempo de ejecución tiene un pico muy elevado. Esto se debe a que en nuestro generador de scripts, hemos forzado que el número de pasajeros $c_{i,j}$, sea un número mayor que 0, y además, hemos puesto un número de franjas menor al número de autobuses.

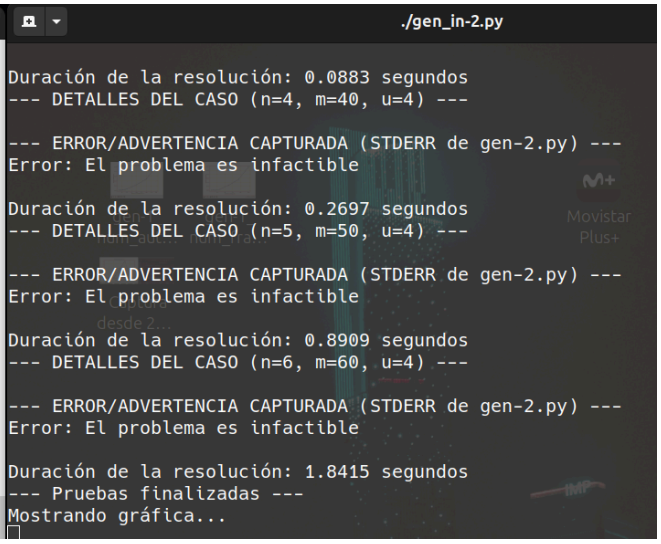
   El conjunto de esto, provoca que al solucionador le cueste mucho más encontrar el camino correcto cuando se incrementa el número de autobuses, como se observa en la gráfica:





4. Por último, hemos probado a realizar unos casos de problemas infactibles, donde las franjas ofertadas son mucho menores al número de autobuses, por lo que podemos ver que a diferencia de los casos anteriores, al detectar la infactibilidad del problema mucho antes, el tiempo de ejecución no crece de esa manera tan brusca pese a que el número de autobuses es mucho mayor.





Conclusiones

El modelo demuestra un correcto funcionamiento al minimizar el impacto de autobuses con pasajeros compartidos en diferentes talleres y franjas.
Los resultados nos muestran que el tiempo de ejecución crece significativamente al aumentar simultáneamente autobuses, talleres y franjas (debido al crecimiento de las restricciones y variables).
Lo que hemos comprobado es que el solver no es muy esclable ya que crece de forma exponencial y al ejecutar un gran número de autobuses su tiempo de ejecución se dispara.
También nos podemos dar cuenta que cuando el problema tiene datos que lo convierten en infactible este tiempo no crece tanto al aumentar los autobuses, y lo mismo le pasa a los problemas de coste 0 (las misma franjas o más que autobuses)