# Tarea 2.

# Resolviendo problemas de programación para el ajuste de flujos de tesorería, gestión de activos y obligaciones de pago

Alejandro Martínez Casado
Paula Pérez Fernández
Rodrigo Salort Antón
Antón Salvadores Muñiz
Alfonso San Miguel Villar

15 de Octubre, 2024

**Asignatura:** Matemáticas y Estadística
**Profesora:** María Araceli Garín Martín

# Introducción

A lo largo de este documento se resolverán diferentes problemas de programación lineal (LP), los cuales serán resueltos de forma rápida y eficiente por medio de un SOLVER. Entre otras ventajas de la utilización de métodos computacionales, podremos realizar de forma sencilla análisis de sensibilidad y observar de forma rápida cómo varían nuestros resultados ante variaciones en nuestros modelos. Para llevarlo a cabo se realizarán los siguientes pasos:

→ Realización de un modelo para el problema en cuestión.

→ Optimización del modelo utilizando como SOLVER *Gurobi*.

→ Realización del análisis de sensibilidad y exportación de datos para poder mostrarlos y tratar con ellos.

Aunque cada ejercicio tiene características diferentes, este procedimiento será utilizado de forma general para la resolución de todos ellos.

Los códigos que se han ejecutado en *Python* se presentan al final de este documento en el Anexo. Para su construcción ha sido necesario la utilización de las librerías de *GurobiPy* y *Pandas*.

A menudo, los problemas que se plantean en un primer momento después de resueltos son modificados, bien porque se ha detectado que falta alguna variable o restricción, o bien porque ha cambiado algún dato (parámetro). En la resolución del nuevo problema conviene intentar aprovechar la solución ya obtenida, quizás porque no está muy lejos de la nueva solución óptima.

Además, aunque no haya modificaciones, puede resultar importante para el decisor disponer de alguna medida de cuán sensible es la solución óptima propuesta a los datos que definen el problema, es decir, cómo cambiarían tanto la solución óptima de su problema como el beneficio o coste asociado ante pequeños cambios en los parámetros del problema.

Sería útil poder responder a estas preguntas sin tener que resolver el problema en cuestión una vez por cada cambio que se quiera evaluar. El estudio de este tipo de cuestiones se conoce como análisis de la sensibilidad o post-optimalidad y, afortunadamente, en el caso de la programación lineal se puede hacer muy eficientemente, a partir de los datos aportados en el óptimo al resolver el problema con el método símplex.

A continuación, se resolverán una serie de ejemplos empleando el método símplex, realizando un analisis de la sensibilidad sobre algunos de ellos.

# Problema 1: Planificación Financiera de una Empresa (Ejercicios 3.4 y 3.14)

El Ejercicio 3.2 presenta un problema de planificación financiera que involucra la gestión de flujos de caja de una empresa a lo largo de 2 años divididos en ocho trimestres.

| Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 |
|---|---|---|---|---|---|---|---|
| 100 | 500 | 100 | -600 | -500 | 200 | 600 | -900 |

Table 1: Flujo de la empresa medido en miles de dólares. Las entradas negativas corresponden a exceso de fondos, las positivas a necesidades de fondos.

La empresa enfrenta fluctuaciones en sus necesidades de efectivo, lo que requiere una estrategia para tomar decisiones óptimas de endeudamiento e inversión. El objetivo es maximizar la riqueza disponible al inicio del noveno trimestre, $v$, aprovechando diferentes opciones de préstamos y oportunidades de inversión disponibles, a la vez que se cubren los requerimientos de efectivo en cada trimestre.

La empresa dispone de las siguientes opciones de financiación:

- Un préstamo a 2 años con un interés trimestral de un 1% disponible únicamente en el primer trimestre. Corresponderá a la variable $x_i$.
- Un préstamo a 6 meses con un interés trimestral del 1.8% disponibles todos los trimestres. Se verá representado por la variable $y_i$.
- Un préstamo trimestral con la misma disponibilidad que el anterior y con un interés del 2.5% por trimestre, representado por $z_i$.
- Los excesos de fondos ($k_i$) pueden ser invertidos con un interés del 0.5% por trimestre.

En nuestro caso la función objetivo maximizar es $v \equiv k_8$ que se corresponde con la riqueza o el excedente de la compañía en el noveno trimestre. Planteamos el problema de maximización con su función objetivo y las restricciones trimestrales, imponiendo también que las variables sean no negativas:

$$
\begin{array}{lllllllllll}
\max\ v & & & & & & & & & & \\
 & x_1 & & + y_1 & & + z_1 & & - k_1 & & & = 100 \\
 & & - 0.01x_1 & + y_2 & - 0.018y_1 & + z_2 & - 1.025z_1 & - k_2 & +1.005k_1 & & = 500 \\
 & & - 0.01x_1 & + y_3 & - 1.018y_1 - 0.018y_2 & + z_3 & - 1.025z_2 & - k_3 & +1.005k_2 & & = 100 \\
 & & - 0.01x_1 & + y_4 & - 1.018y_2 - 0.018y_3 & + z_4 & - 1.025z_3 & - k_4 & +1.005k_3 & & = -600 \\
 & & - 0.01x_1 & + y_5 & - 1.018y_3 - 0.018y_4 & + z_5 & - 1.025z_4 & - k_5 & +1.005k_4 & & = -500 \\
 & & - 0.01x_1 & + y_6 & - 1.018y_4 - 0.018y_5 & + z_6 & - 1.025z_5 & - k_6 & +1.005k_5 & & = 200 \\
 & & - 0.01x_1 & & - 1.018y_5 - 0.018y_6 & + z_7 & - 1.025z_6 & - k_7 & +1.005k_6 & & = 600 \\
 & & - 1.01x_1 & & - 1.018y_6 & & - 1.025z_7 & & +1.005k_7 & - v & = -900 \\
 & x_1, y_i, z_i, k_i \geq 0\ \forall i & & & & & & & & &
\end{array}
$$

En la formulación de este modelo, ya que este dato no se especificaba claramente, se ha hecho el siguiente supuesto: los préstamos generan intereses constantes cada trimestre y al vencimiento se devuelve la totalidad del préstamo inicial.

Por ello, del préstamo a 2 años, $x_1$, el cual solamente está disponible en el primer trimestre, se pagan intereses trimestralmente hasta devolver la cantidad completa del préstamo en el octavo trimestre (2 años). De igual modo, el préstamo de seis meses $y_i$ está disponible al inicio de cada trimestre, por lo que se pedirán 6 préstamos $y_i$, con $i = 1, 2, \ldots 6$. Su vencimiento es de 6 meses. Esto significa que en el primer trimestre se pagan intereses solamente y en el segundo trimestre se pagan los intereses junto con la devolución del total del préstamo.

## Interpretación de resultados:

Al ejecutar el código en *Python*, el SOLVER nos devuelve un valor óptimo de $v = 471.35$K€. Esto refleja la riqueza generada por la empresa a finales del octavo trimestre o, en otras palabras, el excedente de fondos acumulado después de haber cubierto las necesidades de efectivo en cada trimestre, habiendo pagado los intereses de todos los préstamos vencidos y después de haber reinvertido los beneficios generados con los excedentes de capital disponibles.

El valor de $x_1$ que da lugar a tal óptimo es de 424.410K€, que es la cantidad de dinero que la empresa debe pedir prestada en el préstamo a 2 años.

El SOLVER también devuelve el valor $y_2 = 178.21$K€, lo que refleja la cantidad prestada con vencimiento de 6 meses en el comienzo del segundo trimestre, así como un valor de 107.45K€en la variable $z_3$, correspondiente al préstamo trimestral del tercer trimestre. observemos que en el óptimo no se hace uso de los demás préstamos ofrecidos.

Esto nos deja un conjunto de valores de excedentes de capital representados por la variable $k_i$ hasta terminar en el octavo trimestre con un valor de $k_8 = 471.35$K€.

$$\boxed{v = 471.35\text{K €}}$$

## Análisis de sensibilidad

La ejecución del código formulado para este problema nos devuelve los datos relativos al análisis de sensibilidad que pueden encontrarse en las Tabla 5 y Tabla 6.

Veamos como podemos utilizar estos datos para algunos requerimientos:

(i) Suponga que el requerimiento de capital en $Q2$ es 300K€ (en lugar de 500K€). ¿Cómo afectaría esto a la riqueza en $Q9$?

De los datos mostrados en la tabla anterior podemos observar que el precio sombra para la restricción del segundo trimestre es

$$C_2 = -1.06675.$$

Veamos cómo se modifica el problema si en el segundo trimestre el requerimiento de capital disminuye. Esta diferencia viene dada por $\Delta = 300 - 500 = -200$K€, lo cual se encuentra dentro del rango de optimalidad para dicha restricción, pues el nuevo dato es mayor que el rango inferior, 192.85. De este modo, el cambio en el óptimo de la función objetivo en ese trimestre vendrá dado por el producto del precio sombra por la diferencia

$$\Delta_v = (-1.06675) \cdot (-200) = 213.35\text{K€},$$

indicando que aumenta en 213.348K€. Por tanto, la riqueza de la empresa en el noveno trimestre será

$$\boxed{v_N = 471.35 + 213.35 = 684.70\text{K€}}.$$

Podemos comprobar que el resultado es el mismo que se obtendría en caso de que modificáramos el código por encontrarse dentro del rango de optimalidad la modificación.

Observemos que, al relajar las condiciones de requerimiento de capital, la empresa no únicamente necesita pedir prestado menos dinero, sinó que puede destinar una mayor cantidad a inversiones en ese período. Como resultado se obtiene una reducción en el capital destinado al pago de intereses y unos mayores retornos derivados de la inversión.

Si la necesidad de capital en $Q2$ es alta (500 en el caso original), la empresa deberá pedir prestado más, aumentando sus costos financieros. Al reducir esta necesidad a 300, se liberan recursos que pueden ser utilizados de manera más eficiente.

Además, la mayor riqueza acumulada podría indicar que la empresa ha podido aprovechar mejor las oportunidades de inversión en otros trimestres.

(ii) Suponga que el requerimiento de capital en $Q2$ es 100K€ (en lugar de 500K€). ¿Puede usarse el análisis de sensibilidad anterior para calcular la riqueza en $Q8$?

En este caso la diferencia es $\Delta = -400$K€. Podemos ver que el límite inferior en este caso se encuentra por debajo del rango permitido en el análisis (el límite inferior del análisis es 192.85K€). Como la diferencia se encuentra fuera de nuestro rango de optimalidad no podemos usar el análisis de sensibilidad para calcularlo. Veamos lo que sucedería en dicho caso; según el análisis de sensibilidad, la riqueza ahora sería

$$v_N = -400 \cdot 1.06675 + v = 44.65\text{K€}$$

Mientras que si modificados los datos del problema y volvemos a optimizar podemos ver que el óptimo para l riqueza en el último trimestre es

$$\boxed{v = 897.17\text{K€}}$$

Lo cual nos confirma que en efecto el análisis de sensibilidad en este caso no es válido.

(iii) Uno de los proveedores de la empresa podría permitir pagos diferidos de 50K€ del $Q3$ al $Q4$. ¿Cuál sería el valor de esto?

Esta condición conllevaría que el flujo de $Q3$ disminuyera, debido a que se libera dicha cantidad ese trimestre al aceptar pagos en diferido, mientras que en $Q4$ aumenta en la misma cantidad, ya que en ese momento se hace frente al pago, 50K€.

$$Q3 = 50\text{K€} \qquad Q4 = -550\text{K€}$$

Como ambos valores están dentro del intervalo permitido, vamos a utilizar los precios sombra para calcular como se modifica. Para $Q3$

$$\Delta v_3 = -1.05496 \cdot -50 = 52.73\text{K€}$$

En el caso de $Q4$

$$\Delta v_4 = -1.02923 \cdot 50 = -51.46\text{K€}$$

Por lo tanto el valor óptimo se vería afectado por la diferencia de ambas siendo

$$\boxed{v_N = 471.35 + (52.73 - 51.46) = 472.62\text{K €}}$$

El cual podría comprobarse que se trata del mismo resultado que si volviéramos a optimizar el modelo modificando dichos trimestres en el problema.

# Problema 2: Planificación Financiera de una Empresa (Ejercicio 3.5)

En este caso, únicamente se nos solicita resolver el modelo que fue realizado para una empresa en el anterior documento. Dicha empresa contaba con las siguientes formas de financiación:

- Una línea de crédito de 100K€ con un interés del 1% mensual.
- Un instrumento financiero que recibe el nombre de *papel comercial*. Este únicamente podrá ser emitido en cualquiera los tres primeros meses a dos meses con un interés del 2% para el trimestre.
- La inversión del exceso de fondos a una tasa de interés del 0.3% anual.

Además, el flujo de la empresa en los siguientes seis meses era el siguiente:

| Mes | Enero | Febrero | Marzo | Abril | Mayo | Junio |
|---|---|---|---|---|---|---|
| Flujo Neto | -150 | -100 | 200 | -200 | 50 | 300 |

Table 2: Flujo de la empresa medido en miles de euros.

Con estos datos fue posible construir el siguiente modelo

$$
\begin{array}{lllllllll}
\max \ v & & & & & & & & \\
& x_1 & + y_1 & & & & - z_1 & = & 150 \\
& x_2 & + y_2 & & - 1.01x_1 & + 1.003z_1 & - z_2 & = & 100 \\
& x_3 & + y_3 & - 1.02y_1 & - 1.01x_2 & + 1.003z_2 & - z_3 & = & -200 \\
& x_4 & & - 1.02y_2 & - 1.01x_3 & + 1.003z_3 & - z_4 & = & 200 \\
& x_5 & & - 1.02y_3 & - 1.01x_4 & + 1.003z_4 & - z_5 & = & -50 \\
& & & & - 1.01x_5 & + 1.003z_5 & - v & = & -300 \\
& x_1 \leq 100 & & & & & & & \\
& x_2 \leq 100 & & & & & & & \\
& x_3 \leq 100 & & & & & & & \\
& x_4 \leq 100 & & & & & & & \\
& x_5 \leq 100 & & & & & & & \\
& x_i, y_i, z_i \geq 0 \ \forall i & & & & & & &
\end{array}
$$

Una vez hemos construido el modelo, se procedió a resolver de forma computacional. Para ello, utilizamos como SOLVER *Gurobi* implementado en *Python* por medio de la librería *GurobiPy*. El código puede encontrarse en el **Anexo**.

El resultado que nos devolvió fue que el problema era no acotado, correspondiente a que devuelva el valor 4 en *model.status*. Esto significa que la función objetivo en este caso puede tomar valores infinitos, por lo que no obtendremos una solución óptima. Veámoslo con un sencillo ejemplo:

$$
\begin{aligned}
\max \ v =& x \\
x \geq& 0
\end{aligned}
$$

Esta claro que en este problema el valor máximo de $x$ tiende a infinito, por lo que será necesario añadir más restricciones para poder lograr una solución óptima. No es posible que la riqueza de una empresa diverja.

# Problema 3: Gestor de Carteras (Ejercicios 3.6 y 3.13)

Se plantea la siguiente situación:

Un gestor de cartera de bonos dispone de 100000€ para asignar entre dos bonos diferentes: uno corporativo y uno gubernamental.

- Bono corporativo: tiene un rendimiento del 4%, vencimiento de 3 años y una calificación de **A** (calificación numérica de 2). Corresponde a la variable $x_1$.
- Bono gubernamental: tiene un rendimiento del 3%, vencimiento de 4 años y una calificación de **Aaa** (calificación numérica de 1). Lo denotamos como $x_2$.

La función objetivo del problema es maximizar $Z$. Cuando $Z$ agrupa la información de la cantidad de euros que el gestor debe asignar a cada bono para obtener el máximo beneficio en la operación. El problema debe cumplir con estas restricciones:

- El gestor desea asignar sus recursos a una cartera cuya calificación promedio sea de **Aa** (equivalente numérico 1.5).
- El gestor quiere que el vencimiento promedio de la cartera sea como máximo de 3.6 años.
- Además, se debe cumplir la no negatividad de las variables.

Dada la información se obtiene la siguiente formulación para el problema del gestor de cartera:

$$\begin{aligned}
&\text{Max } Z = 4x_1 + 3x_2 \\
&\text{sujeto a:} \\
&\qquad x_1 + x_2 \leq 100 \\
&\qquad \frac{2x_1 + x_2}{100} \leq 1.5 \\
&\qquad \frac{3x_1 + 4x_2}{100} \leq 3.6 \\
&\qquad x_1, x_2 \geq 0
\end{aligned}$$

Donde $x_1$ y $x_2$ representan las cantidades (en miles de euros) asignadas al bono corporativo y al bono gubernamental, respectivamente.

## Optimización del problema y resultados

Al ejecutar el SOLVER, Python nos devuelve un óptimo de 350K€ de valor en la función objetivo a maximizar. Con unos valores de $x_1$ y $x_2$ de 50K€ en cada una. El código puede encontrarse en el **Anexo**.

Esto significa que el gestor deberá asignar una cantidad de 50000 euros al bono corporativo y 50000 euros al bono gubernamental para maximizar el rendimiento de la inversión, en nuestro caso de 350000 euros respetando las restricciones de riesgo que nos planta el problema y de madurez.

## Análisis de sensibilidad

Queremos ver como se modifica el valor óptimo de nuestra función objetivo si modificamos la madurez promedio de la cartera a 3.3 en vez de 3.6. Los resultados obtenidos del análisis de sensibilidad pueden encontrarse en la Tabla 7 y Tabla 8.

Si reducimos la madurez a 3.3 tenemos una diferencia, $\Delta = -0.3$, la cual se encuentra fuera del rango de optimalidad (el mínimo permitido es 3.5) por lo que hemos de volver a optimizarlo. Obtenemos los siguientes valores óptimos para nuestras variables

$$x_1 = 54\text{K€} \qquad x_2 = 42\text{K€}$$

Mientras que el valor óptimo de la función objetivo sería

$$\boxed{4x_1 + 3x_2 = 342\text{K €}}$$

En cambio, si hubiéramos utilizado los datos proporcionados por el análisis de sensibilidad, al ser el precio sombra nulo, nuestro resultado no habría variado. Podemos comprobar que al encontrarse fuera del rango de optimalidad este procedimiento no es válido.

## <u>Problema 4:</u> Fondo de Pensiones (Ejercicios 3.12 y 3.18)

El siguiente problema de optimización lineal trata de minimizar el coste de una cartera de bonos dedicada a financiar un fondo de pensiones.

| **Año 1** | **Año 2** | **Año 3** | **Año 4** | **Año 5** | **Año 6** | **Año 7** | **Año 8** | **Año 9** |
|---|---|---|---|---|---|---|---|---|
| 24 | 26 | 28 | 28 | 26 | 29 | 32 | 33 | 34 |

Table 3: Obligaciones del fondo de pensiones (en millones de euros).

Los bonos disponibles son los siguientes:

| **Bonos** | **Precio** | **Cupón** | **Vencimiento (Años)** |
|---|---|---|---|
| 1 | 102.44 | 5.625 | 1 |
| 2 | 99.95 | 4.75 | 2 |
| 3 | 100.02 | 4.25 | 2 |
| 4 | 102.66 | 5.25 | 3 |
| 5 | 87.90 | 0.00 | 3 |
| 6 | 85.43 | 0.00 | 4 |
| 7 | 83.42 | 0.00 | 5 |
| 8 | 103.82 | 5.75 | 5 |
| 9 | 110.29 | 6.875 | 6 |
| 10 | 108.85 | 6.5 | 6 |
| 11 | 109.95 | 6.625 | 7 |
| 12 | 107.36 | 6.125 | 7 |
| 13 | 104.62 | 5.625 | 8 |
| 14 | 99.07 | 4.75 | 8 |
| 15 | 103.78 | 5.5 | 9 |
| 16 | 64.66 | 0.00 | 9 |

Table 4: Bonos disponibles, además del precio, cupón y vencimiento de los mismos.

Además, tenemos una tasa de reinversión del 2% del capital restante. Partiendo de estos datos nuestro modelo consistirá en minimizar el coste del porfolio. Por tanto, la función objetivo corresponderá al producto de número de bonos y precio para cada uno añadiendo el capital inicial:

$$u(x_i, p_i) = \sum_{i}^{i=16} x_i \cdot p_i + s$$

Donde la variable $x_i$ corresponde a la variable número de bonos y $p_i$ es el parámetro que representa precio del bono $i = 1, \ldots, 16$. Junto a estas variables tenemos que tener en cuenta otra variable, denotada como $k_i$, que corresponde a los fondos restantes. Además, por comodidad hemos denotado como $s$ los fondos iniciales disponibles, los cuales estarán disponibles antes de la compra de bonos.

Las restricciones además de incluir que todas las variables anteriores han de ser no negativas, corresponde a que la suma de los cupones y la reinversión en cada mes tiene que ser igual a los pasivos del fondo de pensiones. En el caso, de que nos encontremos en la fecha de vencimiento del bono hemos de tener en cuenta el valor nominal del bono que se ha supuesto como 100€.

## 0.1 Optimización del problema y resultados

El modelo final se ha implementado en el código que puede encontrarse en el **Anexo**. El valor obtenido para nuestra cartera en el óptimo fue

$$\boxed{u^* = 204.62 \cdot 10^6 €}$$

### Análisis de Sensibilidad

Los datos obtenidos tras realizar el análisis de sensibilidad se pueden encontrar en las Tablas **9** y **10**.

Veamos como cambiarían nuestros resultados bajo ciertos supuestos:

(i) Suponemos que los pasivos en el año 3 son 29 (en vez de 28). ¿Cuánto será el aumento en la cartera?

La diferencia, $\Delta = 1$, se encuentra dentro del rango de optimalidad. Como el precio sombra es 0.91020 el resultado será

$$Z_N = 204.62 \cdot 10^6 + 10^6 \cdot 0.9120 = 205.53 \cdot 10^6 €$$

(ii) Dibuja un gráfico de la estructura a plazo de las tasas de interés implícitas por los precios sombra.

Para realizar dicho gráfico tenemos que igualar los precios sombra a $\frac{1}{(1+r_t)^t}$, debido a que el precio sombra en un año $t$ corresponde al coste de la cartera de bonos que puede atribuirse a un euro de pasivo en dicho año. Despejando podemos obtener la tasa de interés

$$r_t = \frac{1}{S_t^{1/t}} - 1$$

Donde $S_t$ corresponde al precio sombra del año $t$. Con la implementación de la librería *matploblib* hemos obtenido la siguiente figura

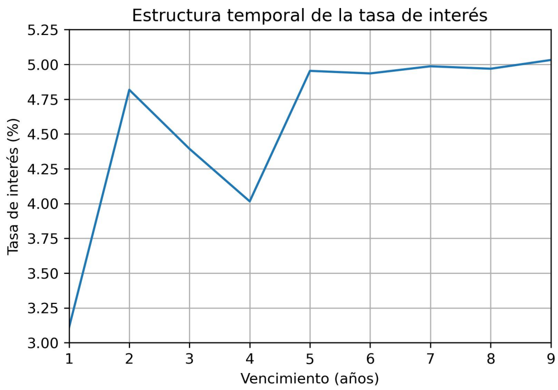


Figure 1: Estructura a plazo de las tasas de interés implícitas por los precios sombra

(iii) El Bono 4 no está incluido en la cartera óptima. ¿En cuánto tendría que disminuir el precio del Bono 4 para que el Bono 4 se convirtiera en parte de la cartera óptima?

El coste reducido para este bono es 0.27502. Conociendo su precio actual de 102.66, podemos calcular cuanto tendría que ser su precio para que formara parte de la cartera óptima:

$$\boxed{p_{B4_N} = (102.66 - 0.27502) \cdot 10^6 = 102.38 \cdot 10^6 €}$$

(iv) El gestor del fondo le gustaría tener 10000 unidades del Bono 3 en la cartera. ¿En cuánto aumentaría esto el costo de la cartera?

Podríamos calcular el nuevo precio de la cartera teniendo en cuenta que el coste reducido del bono 3 es 1.01002€, de este modo, teniendo en cuenta que el precio de este bono es 100.02€ y queremos tener 10000

$$\Delta_{coste} = 10000 \cdot (100.02 + 1.01002) = 1010300.2$$

Por lo tanto el nuevo coste de la cartera será:

$$\boxed{u_N^* = (204.62 + 1.01) \cdot 10^6 = 205.63 \cdot 10^6 €}$$

(v) ¿Hay algún bono que parezca estar mal valorado?

Tenemos varios bonos que parecen mal valorados (3, 4, 7, 10, 12, 14, 16). Reduciendo el precio de dichos bonos por el coste reducido en el análisis de sensibilidad podrían forman parte de la cartera óptima, es decir, podríamos revalorarlos de forma que entren a formar parte de nuestra cartera.

(vi) ¿Qué tasa de interés sobre el efectivo haría óptimo incluir el efectivo como parte de la cartera óptima?

El efectivo viene denotado como $s$, si queremos incluir dicho efectivo en la cartera tendríamos que observar el coste reducido del análisis de sensibilidad. El valor de dicha variable es 0.011, por lo que si aumentáramos la tasa de interés en dicha cantidad el efectivo se incluiría en la cartera óptima, es decir, si la tasa de interés $r$ fuera

$$\boxed{r = 1.031}$$

# 1 Anexo: Códigos de *Python* Utilizados

## 1.1 Problema 1

```
import pandas as pd
from gurobipy import *

# Crear el modelo
model = Model("Ejercicio_3_4")

# Flujo
flujo = [100, 500, 100, -600, -500, 200, 600, -900]
tiempo = len(flujo) - 1

# Variables de decisión para préstamos e inversiones
x = model.addVar(name="x1", lb=0, vtype=GRB.CONTINUOUS)
y = [model.addVar(name=f"y{q+1}", lb=0, vtype=GRB.CONTINUOUS) for q in range(0, tiempo)]
z = [model.addVar(name=f"z{q+1}", lb=0, vtype=GRB.CONTINUOUS) for q in range(0, tiempo+1)]
k = [model.addVar(name=f"k{q+1}", lb=0, vtype=GRB.CONTINUOUS) for q in range(0, tiempo+1)]

# Intereses
rx = 0.01
ry = 0.018
rz = 0.025
rk = 0.005

# Función objetivo (maximizar)
model.setObjective(k[tiempo], GRB.MAXIMIZE)

# Restricciones para cada periodo
for q in range(0, tiempo+1):
    if q == 0:
        model.addConstr(flujo[q] == x + y[q] + z[q] - k[q], f"Q{q+1}")
    elif q == 1:
        model.addConstr(flujo[q] == -rx*x + y[q] - ry*y[q-1] + z[q]
        - (1+rz)*z[q-1] - k[q] + (1+rk)*k[q-1], f"Q{q+1}")
    elif q == tiempo:
        model.addConstr(flujo[q] == -(1+rx)*x - (1+ry)*y[q-2]
        - (1+rz)*z[q-1] - k[q] + (1+rk)*k[q-1], f"Q{q+1}")
    elif q == tiempo - 1:
        model.addConstr(flujo[q] == -rx*x - (ry*y[q-1] + (1+ry)*y[q-2])
        + z[q] - (1+rz)*z[q-1] - k[q] + (1+rk)*k[q-1], f"Q{q+1}")
    else:
        model.addConstr(flujo[q] == -rx*x + y[q] - (ry*y[q-1] + (1+ry)*y[q-2])
        + z[q] - (1+rz)*z[q-1] - k[q] + (1+rk)*k[q-1], f"Q{q+1}")

# Resolver el modelo
model.optimize()

# Informe de sensibilidad

var_names = []
var_final = []
coste_reducido = []
coef_optimo = []
```

```python
obj_low = []
obj_up = []
constr_names = []
precio_sombra = []
rhs_low = []
rhs_up = []

for v in model.getVars():
    var_names.append(v.VarName)
    var_final.append(v.x)
    coste_reducido.append(v.RC)
    coef_optimo.append(v.Obj)
    obj_low.append(v.SAObjLow)
    obj_up.append(v.SAObjUp)

for c in model.getConstrs():
    constr_names.append(c.ConstrName)
    precio_sombra.append(c.Pi)
    rhs_low.append(c.SARHSLow)
    rhs_up.append(c.SARHSUp)

df_vars = pd.DataFrame({
    'Variable': var_names,
    'Valor final': var_final,
    'Coste reducido': coste_reducido,
    'Coeficiente óptimo': coef_optimo,
    'Disminución permitido': obj_low,
    'Crecimiento permitida': obj_up
})

df_constrs = pd.DataFrame({
    'Restricción': constr_names,
    'Precio Sombra': precio_sombra,
    'Valor mínimo permitido': rhs_low,
    'Valor máximo permitido': rhs_up
})

# Exportar a Excel
with pd.ExcelWriter('ejercicio34.xlsx') as writer:
    df_vars.to_excel(writer, sheet_name='Sensibilidad_Variables', index=False)
    df_constrs.to_excel(writer, sheet_name='Sensibilidad_Restricciones', index=False)
```

## 1.2 Problema 2

```
from gurobipy import *


model = Model("Ejercicio_3_5")

# Flujo
flujo = [150, 100, -200, 200, -50, -300]
tiempo = len(flujo) - 1

#Variables
x = [model.addVar(lb=0.0, ub=100, vtype=GRB.CONTINUOUS,
name=f"x{q+1}") for q in range(0, tiempo+1)]
y = [model.addVar(lb=0, vtype=GRB.CONTINUOUS,
name= f"y{q+1}") for q in range(0, tiempo+1)]
z = [model.addVar(lb=0, vtype=GRB.CONTINUOUS,
name= f"z{q+1}", ) for q in range(0, tiempo+1)]

# Intereses
rx = 0.01
ry = 0.02
rz = 0.003

#Funcion objetivo
model.setObjective(z[tiempo], GRB.MAXIMIZE)

# Restricciones
for q in range(0, tiempo+1):
    if q == 0:
        model.addConstr(flujo[q] == x[q] + y[q] - z[q], f"Q{q+1}")
    elif q == 1:
        model.addConstr(flujo[q] == x[q] - 1.01*x[q-1] + y[q]
        - z[q] + (1+rz)*z[q-1], f"Q{q+1}")
    elif q == 3:
        model.addConstr(flujo[q] == x[q] - 1.01*x[q-1] + y[q] - (1+ry)*y[q-2]
        - z[q] + (1+rz)*z[q-1], f"Q{q+1}")
    elif q == tiempo:
        model.addConstr(flujo[q] == - 1.01*x[q-1] - z[q] + (1+rz)*z[q-1], f"Q{q+1}")
    else:
        model.addConstr(flujo[q] == x[q] - 1.01*x[q] - (1+ry)*y[q-2]
        - z[q] + (1+rz)*z[q-1], f"Q{q+1}")

# Resolver el modelo
model.optimize()

print(f"{model.status}")
```

## 1.3 Problema 3

```
import pandas as pd
from gurobipy import *


model = Model("Ejercicio_3_6")

x = [model.addVar(name=f"x{q+1}", lb=0, vtype=GRB.CONTINUOUS) for q in range(0, 2)]

model.setObjective(4*x[0] + 3*x[1], GRB.MAXIMIZE)


#Restricciones

model.addConstr(x[0] + x[1] <= 100)
model.addConstr((2*x[0] + x[1])/100  <= 1.5)
model.addConstr((3*x[0] + 4*x[1])/100  <= 3.6)

model.optimize()


# Informe de sensibilidad

var_names = []
var_final = []
coste_reducido = []
coef_optimo = []
obj_low = []
obj_up = []
constr_names = []
precio_sombra = []
rhs_low = []
rhs_up = []

for v in model.getVars():
    var_names.append(v.VarName)
    var_final.append(v.x)
    coste_reducido.append(v.RC)
    coef_optimo.append(v.Obj)
    obj_low.append(v.SAObjLow)
    obj_up.append(v.SAObjUp)

for c in model.getConstrs():
    constr_names.append(c.ConstrName)
    precio_sombra.append(c.Pi)
    rhs_low.append(c.SARHSLow)
    rhs_up.append(c.SARHSUp)

df_vars = pd.DataFrame({
    'Variable': var_names,
    'Valor final': var_final,
    'Coste reducido': coste_reducido,
    'Coeficiente óptimo': coef_optimo,
```

```
    'Disminución permitido': obj_low,
    'Crecimiento permitida': obj_up
})

df_constrs = pd.DataFrame({
    'Restricción': constr_names,
    'Precio Sombra': precio_sombra,
    'Valor mínimo permitido': rhs_low,
    'Valor máximo permitido': rhs_up
})

# Exportar a Excel
with pd.ExcelWriter('ejercicio36.xlsx') as writer:
    df_vars.to_excel(writer, sheet_name='Sensibilidad_Variables', index=False)
    df_constrs.to_excel(writer, sheet_name='Sensibilidad_Restricciones', index=False)
```

Si lo formulamos en formato estándar

```
import pandas as pd

from gurobipy import *


model = Model("Ejercicio_3_6")

x = [model.addVar(name=f"x{q+1}", lb=0, vtype=GRB.CONTINUOUS) for q in range(0, 5)]

model.setObjective(4*x[0] + 3*x[1], GRB.MAXIMIZE)


#Restricciones

model.addConstr(x[0] + x[1] + x[2] == 100)
model.addConstr(2*x[0] + x[1] + x[3]  == 150)
model.addConstr(3*x[0] + 4*x[1] + x[4]  == 360)

model.optimize()


# Informe de sensibilidad

var_names = []
var_final = []
coste_reducido = []
coef_optimo = []
obj_low = []
obj_up = []
constr_names = []
precio_sombra = []
rhs_low = []
rhs_up = []

for v in model.getVars():
    var_names.append(v.VarName)
```

```
    var_final.append(v.x)
    coste_reducido.append(v.RC)
    coef_optimo.append(v.Obj)
    obj_low.append(v.SAObjLow)
    obj_up.append(v.SAObjUp)

for c in model.getConstrs():
    constr_names.append(c.ConstrName)
    precio_sombra.append(c.Pi)
    rhs_low.append(c.SARHSLow)
    rhs_up.append(c.SARHSUp)

df_vars = pd.DataFrame({
    'Variable': var_names,
    'Valor final': var_final,
    'Coste reducido': coste_reducido,
    'Coeficiente óptimo': coef_optimo,
    'Disminución permitido': obj_low,
    'Crecimiento permitida': obj_up
})

df_constrs = pd.DataFrame({
    'Restricción': constr_names,
    'Precio Sombra': precio_sombra,
    'Valor mínimo permitido': rhs_low,
    'Valor máximo permitido': rhs_up
})

# Exportar a Excel
with pd.ExcelWriter('ejercicio36.xlsx') as writer:
    df_vars.to_excel(writer, sheet_name='Sensibilidad_Variables', index=False)
    df_constrs.to_excel(writer, sheet_name='Sensibilidad_Restricciones', index=False)
```

## 1.4 Problema 4

```
import pandas as pd
from gurobipy import *
import matplotlib.pyplot as plt
import numpy as np

model = Model("Ejercicio_3_12")

#Datos

precio = [102.44, 99.95, 100.02, 102.66, 87.90, 85.43, 83.42, 103.82,
           110.29, 108.85, 109.95, 107.36, 104.62, 99.07, 103.78, 64.66]

cupon = [5.625, 4.75, 4.25, 5.25, 0, 0, 0, 5.75, 6.875, 6.5, 6.625,
           6.125, 5.625, 4.75, 5.5, 0]

vencimiento = [1, 2, 2, 3, 3, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9]

pasivos = [24* 10**6, 26* 10**6, 28* 10**6, 28* 10**6,
           26* 10**6, 29* 10**6, 32* 10**6, 33* 10**6, 34* 10**6]

rk = 0.02

nbonos = len(precio)
tiempo = max(vencimiento)

# Variables

x = [model.addVar(name=f"x{q+1}", lb=0, vtype=GRB.CONTINUOUS) for q in range(nbonos)]
k = [model.addVar(name=f"k{q+1}", lb=0, vtype=GRB.CONTINUOUS) for q in range(tiempo)]
s = model.addVar(name=f"s", lb=0, vtype=GRB.CONTINUOUS)

#Objetivo

model.setObjective(quicksum([x[q] * precio[q]
+ s/nbonos for q in range(nbonos)]), GRB.MINIMIZE)

#Restricciones

for t in range(tiempo):
   cupon_t = quicksum([cupon[q] * x[q] for q in range(nbonos) if vencimiento[q] >= t+1])

   vencimiento_t = quicksum([100 * x[q] for q in range(nbonos) if vencimiento[q] == t+1])

   if t == tiempo-1:
        reinversion_t = (1+rk) * k[t-1]
   elif t == 0:
        reinversion_t = -k[t] + (1+rk)*s
   else:
        reinversion_t = (1+rk) * k[t-1] - k[t]

   model.addConstr(cupon_t + vencimiento_t + reinversion_t == pasivos[t], f"Y{t+1}")

model.optimize()
```

```python
# Informe de sensibilidad

var_names = []
var_final = []
coste_reducido = []
coef_optimo = []
obj_low = []
obj_up = []
constr_names = []
precio_sombra = []
rhs_low = []
rhs_up = []

for v in model.getVars():
    var_names.append(v.VarName)
    var_final.append(v.x)
    coste_reducido.append(v.RC)
    coef_optimo.append(v.Obj)
    obj_low.append(v.SAObjLow)
    obj_up.append(v.SAObjUp)

for c in model.getConstrs():
    constr_names.append(c.ConstrName)
    precio_sombra.append(c.Pi)
    rhs_low.append(c.SARHSLow)
    rhs_up.append(c.SARHSUp)

df_vars = pd.DataFrame({
    'Variable': var_names,
    'Valor final': var_final,
    'Coste reducido': coste_reducido,
    'Coeficiente óptimo': coef_optimo,
    'Disminución permitido': obj_low,
    'Crecimiento permitida': obj_up
})

df_constrs = pd.DataFrame({
    'Restricción': constr_names,
    'Precio Sombra': precio_sombra,
    'Valor mínimo permitido': rhs_low,
    'Valor máximo permitido': rhs_up
})

# Exportar a Excel
with pd.ExcelWriter('ejercicio3_12.xlsx') as writer:
    df_vars.to_excel(writer, sheet_name='Sensibilidad_Variables', index=False)
    df_constrs.to_excel(writer, sheet_name='Sensibilidad_Restricciones', index=False)

#Grafico

x =  np.linspace(1, tiempo, tiempo)
y = [(1/(precio_sombra[t])**(1/(t+1))-1)*100 for t in range(tiempo)]

plt.plot(x, y)
```

```
plt.title('Estructura temporal de la tasa de interés')
plt.xlabel('Vencimiento (años)')
plt.ylabel('Tasa de interés (%)')
plt.xlim(1, tiempo)     # Limitar el eje x de 0 a 10
plt.ylim(3, 5.25) # Limitar el eje y de -1.5 a 1.5

plt.grid()

plt.savefig('tasainteres.jpg', dpi=300, bbox_inches='tight')

plt.show()
```

# Anexo: Tablas de Parámetros de los Análisis de Sensibilidad

## Problema 1

| Variable | Valor final | Coste reducido | Coeficiente óptimo | Disminución permitida | Crecimiento permitido |
|---|---|---|---|---|---|
| x1 | 424,4099 | 0 | 0 | -0,011011437 | 0,008910737 |
| y1 | 0 | -0,021072319 | 0 | -inf | 0,021072319 |
| y2 | 178,2121 | 0 | 0 | -0,008955515 | 0,006551904 |
| y3 | 0 | -0,006108818 | 0 | -inf | 0,006108818 |
| y4 | 0 | -0,026560633 | 0 | -inf | 0,026560633 |
| y5 | 0 | -0,026428491 | 0 | -inf | 0,026428491 |
| y6 | 0 | -0,017235745 | 0 | -inf | 0,017235745 |
| y7 | 0 | 0 | 0 | -inf | 0 |
| z1 | 0 | -0,021334927 | 0 | -inf | 0,021334927 |
| z2 | 0 | -0,014589348 | 0 | -inf | 0,014589348 |
| z3 | 107,4519 | 0 | 0 | -0,006108526 | 0,014489577 |
| z4 | 0 | -0,020482207 | 0 | -inf | 0,020482207 |
| z5 | 0 | -0,020380305 | 0 | -inf | 0,020380305 |
| z6 | 0 | -0,020278911 | 0 | -inf | 0,020278911 |
| z7 | 0 | -0,011054453 | 0 | -inf | 0,011054453 |
| z8 | 0 | 0 | 0 | -inf | 0 |
| k1 | 324,4099 | 0 | 0 | -0,011011437 | 0,008910737 |
| k2 | 0 | -0,006509885 | 0 | -inf | 0,006509885 |
| k3 | 0 | -0,020584618 | 0 | -inf | 0,020584618 |
| k4 | 304,1977 | 0 | 0 | -0,010891438 | 0,006029738 |
| k5 | 801,4746 | 0 | 0 | -0,01094497 | 0,008856951 |
| k6 | 601,2379 | 0 | 0 | -0,010999302 | 0,008900918 |
| k7 | 0 | -0,008945547 | 0 | -inf | 0,008945547 |
| k8 | 471,346 | 0 | 1 | 0 | inf |

Table 5: Parámetros del análisis de sensibilidad de las variables.

| Restricción | Precio Sombra | Valor mínimo permitido | Valor máximo permitido |
|---|---|---|---|
| Q1 | -1,07208 | -299,834 | 539,6556 |
| Q2 | -1,06675 | 192,8473 | 941,8538 |
| Q3 | -1,05496 | -8,34709 | 271,4742 |
| Q4 | -1,02923 | -918,348 | -424,239 |
| Q5 | -1,02411 | -796,737 | -323,36 |
| Q6 | -1,01902 | -98,2208 | 377,5231 |
| Q7 | -1,01395 | 300,2881 | 778,4107 |
| Q8 | -1 | -inf | -428,654 |

Table 6: Parámetros del análisis de sensibilidad de las restricciones.

## Problema 3

| Variable | Valor final | Coste reducido | Coeficiente óptimo | Disminución permitida | Crecimiento permitido |
|---|---|---|---|---|---|
| x1 | 50 | 0 | 4 | 3 | 6 |
| x2 | 50 | 0 | 3 | 2 | 4 |

Table 7: Parámetros del análisis de sensibilidad de las variables.

| Restricción | Precio Sombra | Valor mínimo permitido | Valor máximo permitido |
|---|---|---|---|
| R0 | 2 | 75 | 102 |
| R1 | 1 | 140 | 200 |
| R2 | 0 | 350 | inf |

Table 8: Parámetros del análisis de sensibilidad de las restricciones.

## Problema 4

| Variable | Valor final | Coste reducido | Coeficiente óptimo | Disminución permitida | Crecimiento permitido |
|---|---|---|---|---|---|
| x1 | 145669 | 0 | 102,44 | 98,25597 | 103,5539 |
| x2 | 173862,9 | 0 | 99,95 | 98,52352 | 100,9649 |
| x3 | 0 | 1,010022 | 100,02 | 99,00998 | inf |
| x4 | 0 | 0,275019 | 102,66 | 102,385 | inf |
| x5 | 202121,4 | 0 | 87,9 | 87,1386 | 88,1613 |
| x6 | 202121,4 | 0 | 85,43 | 80,37863 | 86,17647 |
| x7 | 0 | 4,892084 | 83,42 | 78,52792 | inf |
| x8 | 182121,4 | 0 | 103,82 | 101,7064 | 108,9934 |
| x9 | 222593,4 | 0 | 110,29 | 107,9351 | 110,7821 |
| x10 | 0 | 0,490359 | 108,85 | 108,3596 | inf |
| x11 | 267896,7 | 0 | 109,95 | 107,9977 | 110,2911 |
| x12 | 0 | 0,339484 | 107,36 | 107,0205 | inf |
| x13 | 295644,9 | 0 | 104,62 | 102,3405 | 104,7917 |
| x14 | 0 | 0,170257 | 99,07 | 98,89974 | inf |
| x15 | 322274,9 | 0 | 103,78 | -inf | 104,1722 |
| x16 | 0 | 0,371755 | 64,66 | 64,28824 | inf |
| k1 | 0 | 0,041444 | 0 | -0,04144 | inf |
| k2 | 0 | 0,013618 | 0 | -0,01362 | inf |
| k3 | 0 | 0,007614 | 0 | -0,00761 | inf |
| k4 | 0 | 0,053315 | 0 | -0,05332 | inf |
| k5 | 0 | 0,021298 | 0 | -0,0213 | inf |
| k6 | 0 | 0,023431 | 0 | -0,02343 | inf |
| k7 | 0 | 0,019305 | 0 | -0,0193 | inf |
| k8 | 0 | 0,022729 | 0 | -0,02273 | inf |
| k9 | 0 | 0 | 0 | 0 | inf |
| s | 0 | 0,010757 | 1 | 0,989243 | inf |

Table 9: Parámetros del análisis de sensibilidad de las variables.

| Restricción | Precio Sombra | Valor mínimo permitido | Valor máximo permitido |
|---|---|---|---|
| Y1 | 0,969846154 | 8613706,606 | inf |
| Y2 | 0,910197907 | 7787857,669 | 365308259,6 |
| Y3 | 0,879 | 7787857,669 | inf |
| Y4 | 0,8543 | 7787857,669 | inf |
| Y5 | 0,785279165 | 6740659,485 | 322415269,8 |
| Y6 | 0,749000845 | 5210329,825 | 293954831 |
| Y7 | 0,711343405 | 3435514,176 | 325168771,1 |
| Y8 | 0,678469188 | 1772511,848 | 397710094,8 |
| Y9 | 0,642882449 | 0 | 427513903,4 |

Table 10: Parámetros del análisis de sensibilidad de las restricciones.