



Heurística y Optimización

PRÁCTICA DE PROGRAMACIÓN LINEAL

Curso 2024/2025



|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre** | **NIA** | **Correo Electrónico** | **Grupo** |
| Javier Rosales Lozano | 100495802 | 100495802@alumnos.uc3m.es | 81 |
| Alonso Rios Guerra | 100495821 | 100495821@alumnos.uc3m.es | 81 |

Fecha de entrega: 31/10/2024

Índice de contenidos

[1. Introducción. 2](#_Toc180657192)

[2. Parte 1: Modelo Básico en Calc 3](#_Toc180657193)

[2.1. Modelado del problema 3](#_Toc180657194)

[2.2. Implementación del modelo en LibreOffice Calc 5](#_Toc180657195)

[3. Parte 2: Modelo Avanzado en GLPK 6](#_Toc180657196)

[3.1. Modelado del problema 6](#_Toc180657197)

[3.2. Implementación de la Parte 1 en MathProg 6](#_Toc180657198)

[3.3. Implementación de la Parte 2 en MathProg 6](#_Toc180657199)

[3.4. Implementación de ambos modelos (Parte 1 y Parte 2) en MathProg 7](#_Toc180657200)

[4. Conclusiones 7](#_Toc180657201)

[4.1. Interpretación de los resultados y decisiones de diseño generales 7](#_Toc180657202)

[4.2. Conclusiones de la práctica 7](#_Toc180657203)

# Introducción.

La primera práctica del curso de Heurística y Optimización propone un problema de programación lineal sobre el problema de una importante compañía aérea. El objetivo de la práctica se centra en el planteamiento, desarrollo y resolución del problema mediante las técnicas aprendidas en clase puestas en práctica en dos entornos de resolución: LibreOffice Calc y el lenguaje MathProg.

El contenido de la memoria se dividirá en diversas secciones, atendiendo a la estructura de la práctica, donde se explicará el planteamiento del problema y la resolución de éste. Finalmente, analizaremos los resultados obtenidos y daremos nuestras conclusiones sobre ellos.

Primeramente, analizaremos el primer enunciado del problema, y estableceremos un modelo que represente el problema de programación lineal presentado. Además, responderemos a la solución obtenida de este mediante el resultado hecho por una hoja de cálculo de LibreOffice Calc. Después, interpretaremos los resultados obtenidos.

Seguidamente, utilizaremos el mismo método para el segundo enunciado del problema, el cual será un modelo más avanzado. Este modelado vendrá seguido por la implementación del problema anterior en un script de MathProg, el cual nos servirá para compararlo con las soluciones obtenidas en la primera parte de la práctica, y para modelar más fácilmente el segundo problema.

Finalmente, concluiremos con la valoración de los resultados obtenidos en ambas partes y consideraremos nuestras respectivas opiniones acerca del trabajo realizado.

# Parte 1: Modelo Básico en Calc

## Modelado del problema

Una vez interpretado el enunciado, podemos concluir que tendremos una **matriz de variables de decisión**, en la que cada columna representará el número de billetes vendidos para cada avión, y cada fila representará número de billetes vendidos de cada tipo. Asignaremos esta matriz como de dimensión (); para diferenciar el tipo de billetes en la matriz, asignaremos las letras (donde representa el avión) para los billetes Estándar, Leisure+ y Business+, respectivamente.

Podemos representar el **número de billetes vendidos en cada avión** como un vector columna (); cada fila se compone de la suma de los elementos de la matriz ; por otro lado, podemos representar el **total de billetes vendidos de cada tipo entre todos los aviones** como un vector columna (); cada fila se compone de la suma de los elementos de cada fila de la matriz ;

Donde ​ representa el elemento en la fila y la columna de la matriz .

Una vez interpretadas las variables de decisión, debemos representar la función objetivo. Ya hemos dicho que el objetivo será maximizar los beneficios de la empresa, por lo que es obvio que la función objetivo sea de tipo maximización. También sabemos los precios de cada tipo de billete, por lo que podemos interpretar la función objetivo como la suma del número de billetes vendidos de cada tipo en cada avión por su respectivo precio. Como los tipos de billete los hemos dividido por filas, podemos sumar los elementos de cada fila entre sí y multiplicarlos por dichos precios. Entonces, la **función objetivo**, quedaría de la siguiente manera:

En este caso, cada sumatorio representa la suma total de billetes de cada tipo vendidos. Para entenderlo de una mejor forma, la función objetivo podría interpretarse como la suma de todos los elementos de cada fila multiplicada por su respectivo precio. Podemos representar los **precios de cada billete (en €)** como un vector fila () al cual le asignaremos la letra :

También representaremos la **capacidad permitida por cada billete (en )** como un vector columna () de la siguiente manera:

Para terminar el modelado del problema, nos queda representar las **restricciones**. Estas restricciones vienen dadas por:

* **Número de asientos de cada avión ()**: cada avión presenta un número de asientos limitado, por lo que la suma de los elementos de cada columna de la matriz debería ser menor o igual a los asientos totales del avión que representa dicha columna. Podemos representar el **número de asientos totales de cada avión** como un vector columna () al cual denominaremos ; debemos considerar que este dato debe componerse de números enteros.
* **Capacidad máxima de cada avión ()**: podríamos interpretarlo como la multiplicación de cada elemento de una columna por el respectivo equipaje máximo dado por el tipo de billete. Podemos representar la **capacidad máxima de cada avión (en )** como un vector columna () que denominaremos , donde cada fila representa la capacidad de un avión.
* **Mínimo de billetes vendidos del tipo Leisure+ para cada avión**: el hecho de que sea para cada avión cobra importancia en el modelado; como hemos dividido los tipos de billetes por filas, todos los elementos pertenecientes a la segunda fila de la matriz deben ser individualmente mayor o igual a este parámetro.
* **El mínimo de billetes vendidos del tipo Business+**: del mismo modo que la anterior restricción, solo que considerando los elementos de la tercera fila de la matriz .
* **Porcentaje de billetes Estándar vendidos respecto al total:** finalmente, debido a que la compañía es de bajo coste, el número de billetes Estándar vendidos debe conformar mínimo el 60% del total de billetes vendidos entre todos los aviones. Esto sería fácil de representar considerando la suma de los elementos de la primera fila como el total de los billetes Estándar vendidos y la suma de todos los elementos de la matriz como el número total de billetes vendidos.

Una vez hemos establecido los vectores y las matrices, problema de programación lineal que modela el problema especificado se representaría de la siguiente manera:

Donde recordemos que:

* es el vector fila de los costes de cada billete
* es el vector columna de la suma de los billetes de cada tipo entre todos los aviones ( será la suma de los billetes vendidos del tipo )
* es la suma de billetes vendidos en el avión
* es el número máximo de asientos en el avión
* es la matriz de variables de decisión, que especifica el número de billetes de cada tipo en cada avión
* es el vector fila de la capacidad máxima permitida para cada tipo de billete
* es el número de billetes Leisure+ vendidos en el avión
* es el número de billetes Business+ vendidos en el avión

## Implementación del modelo en LibreOffice Calc

La resolución de éste modelo se hace en una hoja de cálculo de LibreOffice, y la resolución del problema mediante la función **Solver** (usando el algoritmo lineal de LibreOffice).

En la hoja de cálculo podemos encontrar diferentes matrices y vectores, representando los datos ya especificados en el apartado anterior. Podemos diferenciar también los datos mediante el color de su celda, **estableciendo en rojo los datos constantes del problema y en amarillo las variables que muestran la solución óptima del problema**.

Como decisiones de diseño podemos declarar la creación de una matriz de costes en vez de un vector de costes, ya que supondría una facilidad a la hora de calcular la función objetivo; por otro lado, hemos asignado para cada tipo de billete una letra de forma que podamos distinguir los tipos de billetes en cada parte del proyecto; de igual manera, el modelado en la hoja de cálculo se ha diseñado de tal manera que se puedan modificar los datos otorgados del problema, pudiendo cambiar el problema de programación lineal. De esta forma cualquier restricción nueva que queramos añadir, o cualquier dato que queramos modificar, es posible. El resultado del problema modelado en la hoja de cálculo de LibreOffice es el siguiente:

Según lo observado en la hoja de cálculo, podemos establecer que la solución es correcta, ya que cumple todas las restricciones impuestas y todas las variables son enteras y positivas. Por otro lado, interpretando las restricciones individualmente, podemos observar que la solución que nos otorga el Solver de LibreOffice es una solución donde, para cada avión, se venden todos los asientos que dispone; la capacidad de cada avión no es superada nunca; la restricción del número mínimo de billetes vendidos Leisure+ y Business+ en cada avión se cumple; y el total de billetes Estándar vendidos representa exactamente el 60% del total de billetes vendidos entre todos los aviones.

# Parte 2: Modelo Avanzado en GLPK

## Modelado del problema

## Implementación de la Parte 1 en MathProg

Para la implementación del problema permite practicar con la sintaxis de este lenguaje antes de abordar la implementación del modelo de la parte 2. Este modelo es similar al modelo implementado en la hoja de cálculo de LibreOffice, con la única diferencia de que dividimos el modelado en tres módulos.

En el módulo p1.dat podemos observar las variables constantes del problema, como el precio de cada billete, la capacidad de cada avión, el número de asientos, el peso permitido para cada billete, el número mínimo de billetes Leisure+ y Business+ vendidos para cada avión, y el porcentaje mínimo de billetes estándar vendidos en total. Todos estos datos son modificables, y su cambio permitiría el modelado de diferentes problemas de programación lineal distintos.

En el módulo p1.mod establecemos todo lo necesario para la resolución del problema: la matriz de las variables de decisión, la función objetivo y las restricciones del problema. Este módulo es intocable, ya que representa el problema especificado en el enunciado.

Finalmente, si ejecutamos el comando glpsol -m p1.mod -d p1.dat -o p1.txt obtenemos un fichero de texto p1.txt donde podremos observar los resultados obtenidos. En el fichero primeramente vemos que el resultado de la función objetivo es el mismo que al ejecutar el Solver de LibreOffice, lo cual ya nos da indicios de que el resultado de la parte 1 es correcto. Se muestra a su vez una tabla donde vemos que cada restricción del problema se cumple. Finalmente, podemos observar una última tabla, donde podemos observar el valor de todas las variables de decisión:

Sin embargo, podemos observar que estos valores no coinciden con los marcados en la hoja de cálculo anterior; esto podría dar indicios de que **no hay una única combinación de valores fijos para las variables de decisión que otorguen la solución óptima del problema**.

## Implementación de la Parte 2 en MathProg

## Implementación de ambos modelos (Parte 1 y Parte 2) en MathProg

# Conclusiones

## Interpretación de los resultados y decisiones de diseño generales

## Conclusiones de la práctica