

## 1. Implementación computacional

### 1.1. Descripción del problema

Considere el problema de la pregunta 1 de la I1. Lo que se pide a grandes rasgos es la implementación computacional, usando la interfaz Python-Gurobi, del modelo del problema. Debe utilizar el modelo de la pauta que está subido en Canvas. A continuación se indicarán los elementos técnicos a considerar en esta pregunta.

### 1.2. Base de datos:

La implementación computacional de este problema debe poder recibir cualquier instancia de parámetros, independiente de su tamaño y valores de los datos. No puede asumir valores ni cardinalidades de ningún parámetro o conjunto. Tanto los cuadrantes como los tipos de semillas serán trabajados de manera abstracta, sólo considerándolos por sus índices. Toda la información de las instancias de datos vendrá en 7 archivos con extensión .csv. Una instancia de estos archivos se puede descargar desde Canvas, junto al enunciado de la tarea. Es fuertemente recomendado testear la robustez del código con estas instancias, ya que para la evaluación se utilizarán unos de idéntico formato pero con diferentes valores y dimensiones. Cabe aclarar que los tamaños de los archivos csv indicarán los tamaños de los conjuntos  $K$ ,  $J$  y  $T$  de la pauta. Puede asumir que todos los valores dentro de los archivos son valores enteros. Deben saber que en la pauta todos esos conjuntos de índices parten en 1 y terminan en la cardinalidad del conjunto. A continuación se detalla el formato de cada uno de estos archivos.

#### 1.2.1. capacidad\_por\_saco.csv

Este archivo csv tiene solo 1 columna y  $|J|$  filas. Contiene la información del parámetro  $\alpha_j$  de cada semilla  $j \in J$ . En la primera fila, está el valor de  $\alpha_1$ , en la segunda está el valor de  $\alpha_2$ , y así sucesivamente.

#### 1.2.2. costo\_saco.csv

Este archivo csv tiene una matriz de  $|J|$  filas y  $|T|$  columnas. Contiene la información del parámetro  $c_{jt}$  de cada semilla  $j \in J$  para cada mes  $t \in T$ . En la primera fila, en la primera columna está el valor de  $c_{1,1}$  y están ordenados de manera creciente los datos.

#### 1.2.3. tiempo\_demora.csv

Este archivo csv tiene solo 1 columna y  $|J|$  filas. Contiene la información del parámetro  $\theta_j$  de cada semilla  $j \in J$ . En la primera fila, está el valor de  $\theta_1$  y están ordenados de manera creciente.

#### 1.2.4. kilos\_fruta.csv

Este archivo csv tiene solo 1 columna y  $|J|$  filas. Contiene la información del parámetro  $\lambda_j$  de cada semilla  $j \in J$ . En la primera fila, está el valor de  $\lambda_1$  y están ordenados de manera creciente.

#### 1.2.5. precio\_venta.csv

Este archivo csv tiene una matriz de  $|J|$  filas y  $|T|$  columnas. Contiene la información del parámetro  $\beta_{jt}$  de cada semilla  $j \in J$  para cada mes  $t \in T$ . En la primera fila, en la primera columna está el valor de  $\beta_{1,1}$  y están ordenados de manera creciente los datos.

### 1.2.6. `capital_inicial.csv`

Este archivo csv tiene solo 1 columna y 1 fila. Contiene la información del parámetro  $\gamma$ . Es sólo un valor.

### 1.2.7. `cantidad_cuadrantes.csv`

Este archivo cvs tiene solo 1 columna y 1 fila. Contiene la información de la cantidad de cuadrantes. Es sólo un valor.

## 1.3. Código `main.py`

Se debe elaborar y entregar, junto con el desarrollo de la tarea, un código llamado `main.py`. Ese código debe cumplir una serie de requisitos:

- 1) Que al ser ejecutado en una carpeta que contenga los 7 archivos de la base de datos, logre abrir los archivos y extraer su información. Puede asumir que los archivos seguirán el formato mencionado de manera exacta.
- 2) Se debe resolver el modelo utilizando la librería Gurobi. Cada parte del código debe estar comentada de manera brevísima y clara. Además, debe asumir que la solución óptima existe.
- 3) El código debe imprimir en consola de manera clara el valor óptimo y su unidad de medida. Además, debe imprimir en consola de manera clara la cantidad de veces que se plantó (no importa qué semilla) en cada terreno  $k \in K$  en el marco de  $T$  meses.
- 4) En su código debe elaborar una tabla que permita visualizar la información de la variable  $X$  a modo de calendario. En cada fila debe ir cada terreno y en las columnas los meses. El interior de la tabla se debe rellenar con un cero cuando no se plante nada, y con un valor entero (representando el tipo de semilla  $j \in J$ ) cuando se plante. Debe ser una tabla perfectamente entendible y que no esté impresa en la consola. Tiene permitido usar cualquier librería externa.