

Manual de Usuario: Modelos de Optimización Lineal con Gurobi y Python

1. Introducción

Este manual proporciona una guía práctica para construir, resolver e interpretar modelos de optimización lineal utilizando Python y el solver Gurobi. Está dirigido a estudiantes y docentes que deseen desarrollar modelos como los empleados en la producción de jugos o mezcla de productos con restricciones de capacidad, insumos y demanda.

2. Estructura Básica de un Modelo de Optimización

2.1 Importación de librerías

from gurobipy import Model, GRB
import pandas as pd (si se usan datos desde Excel)

```
!pip install gurobipy  
from gurobipy import Model, GRB  
import pandas as pd
```

2.1 Creación del modelo

Esto crea un objeto de modelo de Gurobi, el cual contendrá todas las variables, restricciones y la función objetivo.

```
model = Model("Nombre_Del_Modelo")
```

2.2 Definición de conjuntos e índices

Los conjuntos representan los grupos de elementos sobre los cuales se construye el modelo, como productos, plantas, ingredientes, etc.

```
productos = ['A', 'B', 'C']  
fabricas = ['X', 'Y']
```

2.3 Parámetros

Variables fijas que no cambian durante la ejecución del modelo, como precios, capacidades, demandas.

```
precio = {'A': 100, 'B': 150, 'C': 200}
capacidad = {'X': 500, 'Y': 600}
```

2.4 Variables de decisión

Se crean con `model.addVars(...)`. Representan las cantidades a determinar, como cuánto producir de cada jugo. Puede definirse el limite inferior (`lb=n`) y definir

```
A = model.addVar(name="A", lb=0)
B = model.addVar(name="B", lb=0)
```

2.5 Función objetivo

Se define con `model.setObjective(...)`. Puede ser de maximización (`GRB.MAXIMIZE`) o minimización (`GRB.MINIMIZE`).

```
model.setObjective(30 * A + 50 * B, GRB.MAXIMIZE)
```

2.6 Tipo de variables numéricas

Se define con `vtype=`. Puede ser Continuos (`GRB.CONTINUOUS`), Enteros (`GRB.INTEGER`) y Binarias (`GRB.BINARY`).

```
x = model_electro.addVars(plantas, centros, vtype=GRB.CONTINUOUS, name="x", lb=0)
```

2.7 Restricciones

Se definen con `model.addConstr(...)` o `model.addConstrs(...)`. Limitan las decisiones posibles según recursos, demanda, etc.

```
model.addConstr(2 * A + 4 * B <= 100, name="Trabajo")
model.addConstr(3 * A + 2 * B <= 90, name="Material")
```

2.8 Solución del modelo

`model.optimize()`: Esta instrucción resuelve el modelo usando el solver de Gurobi.

```
model.optimize()
```

3. Interpretación de Resultados

`X[f, p].x`: Devuelve el valor óptimo de la variable de decisión 'X[f, p]' tras resolver el modelo.

`model.objVal`: Devuelve el valor óptimo de la función objetivo.

`v.VarName`: Nombre de la variable, útil para imprimir resultados o escribirlos a archivos.

`model.getConstrByName('Nombre')`: Permite obtener una restricción para acceder a sus propiedades.

c.pi: Precio sombra de la restricción 'c'. Muestra cuánto mejoraría la función objetivo si se relaja una unidad esta restricción.

v.SAObjLow, v.SAObjUp: Límite inferior y superior del intervalo de sensibilidad para el coeficiente objetivo de una variable.

Constantes de solución de resultados:

Constante	Significado
GRB.OPTIMAL	✓ Se encontró una solución óptima
GRB.INFEASIBLE	✗ El modelo no tiene solución viable
GRB.UNBOUNDED	⚠ El modelo no tiene límite superior o inferior (solución infinita)
GRB.INF_OR_UNBD	📦 Gurobi no pudo determinar si es infactible o no acotado
GRB.TIME_LIMIT	🕒 Tiempo máximo alcanzado sin resolver

4. Lectura de Datos desde Excel

Se usa pandas: `pd.read_excel('archivo.xlsx', sheet_name='Hoja')` para importar conjuntos, parámetros o consumos.

```
# Asegúrese de subir el archivo 'datos.xlsx' en tu entorno de Colab
archivo = 'datos.xlsx'

productos_df = pd.read_excel(archivo, sheet_name="Productos")
recursos_df = pd.read_excel(archivo, sheet_name="Recursos")
consumo_df = pd.read_excel(archivo, sheet_name="Consumo")
```

5. Recomendaciones Finales

- Verifica que todos los parámetros y conjuntos estén correctamente definidos.
- Usa nombres explícitos para variables y restricciones.
- Interpreta los resultados óptimos y los precios sombra para obtener valor del análisis de sensibilidad.