

# Exámen Parcial Investigación Operativa

Segundo Semestre 2025

Universidad de San Andrés

¡RESUELTO PARA USTED!

**Nombre:** Vincent Vega

## **Pautas de Examen:**

- Cada problema se puntua sobre un total de 10.
- La nota es el promedio de las notas de los 3 problemas. Este parcial es el 50 % de su nota práctica.
- Es condición necesaria para aprobar tener al menos 4 puntos en cada problema.
- Para todo ejercicio escriba claramente cuales son las variables de decisión, la función objetivo y las restricciones.
- Para todo ejercicio escriba el código de Python correspondiente en su computadora y reporte el resultado de la ejecución (valores de las variables y el valor de la función objetivo).

## **1. Programación Lineal**

El empleado LG de cierto emprendimiento botánico debe coordinar la ubicación de productos a tres distintos clientes. Tiene dos proveedores del producto, uno en Escobar y el otro en Pueyrredón, que le venden el producto a distintos valores por unidad. Además, el costo de transporte de cada uno de estos proveedores a los clientes es distinto. Se pueden comprar hasta 21 kilos en Escobar y hasta 13 kilos en Pueyrredón. LG necesita llevar 4, 20 y 6 kilos a cada uno de los clientes Ezequiel, Ignacio y Felipe. Los costos de compra y transporte se resumen en la siguiente tabla.

	\$ de transporte (por kg)			\$ por kg
	Ezequiel	Ignacio	Felipe	
Escobar	\$100	\$190	\$160	\$300
Pueyrredón	\$180	\$110	\$140	\$420

## 1.1. Planteo

Para resolverlo vamos a definir variables  $x_{ij}$  que nos dirán la cantidad de producto que se compra a cada proveedor  $i$  y se manda al cliente  $j$ . La función objetivo es de la forma:

$$\text{mín } Z = \sum_{i=1}^2 \sum_{j=1}^3 (c_{ij} + t_{ij}) x_{ij}$$

donde  $c_{ij}$  es el costo de compra del producto al proveedor  $i$  para el cliente  $j$  y  $t_{ij}$  es el costo de transporte del producto desde el proveedor  $i$  al cliente  $j$ . Las restricciones son las siguientes:

$$\sum_{i=1}^2 x_{ij} = d_j \quad \forall j \in \{1, 2, 3\} \quad (\text{Demanda})$$
$$\sum_{j=1}^3 x_{ij} \leq C_i \quad \forall i \in \{1, 2\} \quad (\text{Capacidad})$$

donde  $d_j$  es la demanda del cliente  $j$  y  $C_i$  es la capacidad del proveedor  $i$ .

## 1.2. Python

```
1 import picos
2 import numpy as np
3
4 P = picos.Problem()
5
6 # Variables de decision
7 x = picos.RealVariable('x', (2, 3), lower=0)
8
9 # Costos de compra por proveedor y cliente
10 c = np.array([
11     [300, 300, 300],
12     [420, 420, 420]
13 ])
14 # Costos de transporte por proveedor y cliente
15 t = np.array([
16     [100, 190, 160],
17     [180, 110, 140]
18 ])
19
20 # Demanda de cada cliente
21 d = [4, 20, 6]
22 # Capacidad de cada proveedor
23 C = [21, 13]
24
25 # Funcion objetivo: minimizar costos totales
26 P.set_objective('min', picos.sum([
27     (c[i, j] + t[i, j]) * x[i, j]
28     for i in range(2)
```

```

29     for j in range(3)
30 ]))
31
32 # Restricciones de demanda
33 for j in range(3):
34     P.add_constraint(picos.sum(x[:, j]) == d[j])
35
36 # Restricciones de capacidad
37 for i in range(2):
38     P.add_constraint(picos.sum(x[i, :]) <= C[i])
39
40 P.solve(solver='glpk')
41 print(f"x = {x.value}")
42 print(f"Z = {P.value}")

```

```

x = [ 4.00e+00  1.10e+01  6.00e+00]
[ 0.00e+00  9.00e+00  0.00e+00]

Z = 14520.0

```

### 1.3. Respuesta

El costo total mínimo es de \$14520. Se deben comprar 4 kilos en Escobar para Ezequiel, 11 kilos en Escobar con 9 kilos en Pueyrredón para Ignacio y 6 kilos en Escobar para Felipe.

### 1.4. Rúbrica de Corrección

El 50 % del punto se obtiene planteando bien el problema. Dentro de ese 50 % la mitad se obtiene planteando bien las variables y la función objetivo. La otra mitad se obtiene planteando bien las restricciones. El otro 25 % del punto se obtiene con el código en Python. Dentro de ese 25 %, la mitad se obtiene teniendo el código correcto y el otro 50 % teniendo el código claro, ordenado y legible. El 25 % final se obtiene habiendo llegado al resultado exacto y correcto del problema.

Criterio	Puntos	Bien	Regular	Mal	Obtenido
Variables y función objetivo	2.5				____/2.5
Restricciones	2.5				____/2.5
Código correcto	1.25				____/1.25
Código claro y ordenado	1.25				____/1.25
Resultado correcto	2.5				____/2.5
<b>NOTA FINAL</b>	<b>10</b>				____/10

## 2. Programación Entera Mixta

La planta de botellas PET recicladas (B) debe planificar su producción para el próximo mes (4 semanas). Cada semana dispone de una capacidad de línea limitada de  $H = 18$  horas.

Siempre que en una semana se decida producir es necesario realizar un setup que consume  $s = 1.5$  horas de línea y genera un costo fijo de  $f = \$75$ , aun cuando el volumen producido sea pequeño. Las demandas semanales (medidas en miles de botellas) son:

$$D_1 = 20, D_2 = 15, D_3 = 22, D_4 = 13.$$

El proceso de fabricación insume 0.6 horas por cada mil botellas producidas y tiene un costo variable de 1.8 por mil botellas. Se permite producir por encima de la demanda semanal y almacenar el excedente pagando un costo de inventario de 0.15 por mil botellas y por semana. A fin de mes se quiere que el inventario se encuentre en las mismas condiciones que al comienzo, es decir, nulo. Escriba un problema de programación mixta que permita minimizar los costos totales de la planta.

## 2.1. Planteo

Para resolver este problema de programación entera mixta, definimos las siguientes variables de decisión:

- $x_t$ : cantidad de miles de botellas producidas en la semana  $t$
- $y_t$ : variable binaria que indica si se produce en la semana  $t$  (1 si se produce, 0 si no)
- $I_t$ : inventario al final de la semana  $t$

La función objetivo es minimizar los costos totales:

$$\text{mín } Z = \sum_{t=1}^4 (75y_t + 1,8x_t + 0,15I_t)$$

Las restricciones son:

$I_t = I_{t-1} + x_t - d_t$	$\forall t \in \{1, 2, 3, 4\}$	(Balance de inventario)
$0,6x_t + 1,5y_t \leq 18$	$\forall t \in \{1, 2, 3, 4\}$	(Capacidad de línea)
$x_t \leq My_t$	$\forall t \in \{1, 2, 3, 4\}$	(Setup obligatorio)
$I_0 = 0$		(Inventario inicial)
$I_4 = 0$		(Inventario final)
$x_t \geq 0, I_t \geq 0$	$\forall t \in \{1, 2, 3, 4\}$	(No negatividad)
$y_t \in \{0, 1\}$	$\forall t \in \{1, 2, 3, 4\}$	(Variables binarias)

donde  $d_t$  son las demandas semanales  $[20, 15, 22, 13]$  y  $M$  es una constante suficientemente grande.

## 2.2. Python

```
1 import picos
2 import numpy as np
3
4 P = picos.Problem()
5
6 # Variables
7 x = picos.RealVariable('x', 4, lower=0)
8 y = picos.BinaryVariable('y', 4)
9 I = picos.RealVariable('I', 4, lower=0)
10
11 # Parametros
12 d = [20, 15, 22, 13] # demandas semanales
13 M = 100 # constante grande
14
15 # Funcion objetivo
16 P.set_objective('min', picos.sum([
17     75 * y[i] + 1.8 * x[i] + 0.15 * I[i]
18     for i in range(4)
19 ]))
20
21 # Restricciones de balance de inventario
22 I_prev = 0 # inventario inicial
23 for t in range(4):
24     P.add_constraint(I[t] == I_prev + x[t] - d[t])
25     I_prev = I[t]
26
27 # Restricciones de capacidad
28 for t in range(4):
29     P.add_constraint(0.6 * x[t] + 1.5 * y[t] <= 18)
30
31 # Restricciones de setup obligatorio
32 for t in range(4):
33     P.add_constraint(x[t] <= M * y[t])
34
35 # Inventario final nulo
36 P.add_constraint(I[3] == 0)
37
38 P.solve(solver='glpk')
39 print(f"x = {x.value}")
40 print(f"y = {y.value}")
41 print(f"I = {I.value}")
42 print(f"Z = {P.value}")
```

```
x = [ 2.00e+01]
[ 2.25e+01]
[ 2.75e+01]
[ 1.78e-15]

y = [ 1.00e+00]
[ 1.00e+00]
[ 1.00e+00]
```

```
[ 0.00e+00]

I = [ 0.00e+00]
[ 7.50e+00]
[ 1.30e+01]
[ 0.00e+00]

Z = 354.075
```

## 2.3. Respuesta

El plan óptimo de producción tiene un costo total de \$354.075. Se debe producir en las semanas 1, 2 y 3, evitando la semana 4. La producción es de 20, 22.5 y 27.5 miles de botellas respectivamente. El inventario al final de cada semana es de 0, 7.5 y 13 miles de botellas respectivamente.

## 2.4. Rúbrica de Corrección

El 50 % del punto se obtiene planteando bien el problema. Dentro de ese 50 % la mitad se obtiene planteando bien las variables y la función objetivo. La otra mitad se obtiene planteando bien las restricciones. El otro 25 % del punto se obtiene con el código en Python. Dentro de ese 25 %, la mitad se obtiene teniendo el código correcto y el otro 50 % teniendo el código claro, ordenado y legible. El 25 % final se obtiene habiendo llegado al resultado exacto y correcto del problema.

Criterio	Puntos	Bien	Regular	Mal	Obtenido
Variables y función objetivo	2.5				____/2.5
Restricciones	2.5				____/2.5
Código correcto	1.25				____/1.25
Código claro y ordenado	1.25				____/1.25
Resultado correcto	2.5				____/2.5
<b>NOTA FINAL</b>	<b>10</b>				____/10

## 3. Programación No Lineal

En mercados donde la demanda de un producto depende del precio de forma no lineal, la fijación del precio se convierte en un problema de programación cuadrática. Considere que tiene que ponerle precio a tres productos y se sabe que la demanda para uno es decreciente de forma lineal con el precio:

$$d(p) = a - Bp \quad p^\top = (p_1, p_2, p_3)$$

$$\text{donde } a^\top = (100, 80, 90)^\top \text{ y } B = \begin{pmatrix} 8 & 1,5 & 1 \\ 1,5 & 6 & 1,5 \\ 1 & 1,5 & 7 \end{pmatrix}$$

Los costos por producto se escriben como  $c^\top = (20, 18, 22)$ . Los ingresos se pueden calcular como  $I(p) = p^\top d(p)$  y los costos como  $C(p) = c^\top d(p)$ . Escriba un código en Python que:

- 1) Determine si la función beneficios  $B(p) = I(p) - C(p)$  es cóncava o convexa clasificando el Hessiano de  $B(p)$ .
- 2) Optimice el problema usando SciPy.

*Ayuda: Si  $A$ ,  $B$ ,  $C$  son matrices entonces*

### 3.1. Resolución

Sea  $p^\top = (p_1, p_2, p_3)$  y  $d(p) = a - Bp$ . Entonces

$$Bp = \begin{pmatrix} 8 & 1,5 & 1 \\ 1,5 & 6 & 1,5 \\ 1 & 1,5 & 7 \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} = \begin{pmatrix} 8p_1 + 1,5p_2 + p_3 \\ 1,5p_1 + 6p_2 + 1,5p_3 \\ p_1 + 1,5p_2 + 7p_3 \end{pmatrix}.$$

Con  $a^\top = (100, 80, 90)$ , se sigue que

$$\alpha(p) = a - Bp = \begin{pmatrix} 100 - 8p_1 - 1,5p_2 - p_3 \\ 80 - 1,5p_1 - 6p_2 - 1,5p_3 \\ 90 - p_1 - 1,5p_2 - 7p_3 \end{pmatrix}.$$

Los ingresos son  $I(p) = p^\top d(p) = (p_1 \ p_2 \ p_3) \alpha(p)$ . Al expandir término a término:

$$\begin{aligned} I(p) &= p_1 (100 - 8p_1 - 1,5p_2 - p_3) + p_2 (80 - 1,5p_1 - 6p_2 - 1,5p_3) + p_3 (90 - p_1 - 1,5p_2 - 7p_3) \\ &= 100p_1 - 8p_1^2 - 1,5p_1p_2 - p_1p_3 + 80p_2 - 1,5p_1p_2 - 6p_2^2 - 1,5p_2p_3 \\ &\quad + 90p_3 - p_1p_3 - 1,5p_2p_3 - 7p_3^2 \\ &= \boxed{-8p_1^2 - 6p_2^2 - 7p_3^2 - 3p_1p_2 - 2p_1p_3 - 3p_2p_3 + 100p_1 + 80p_2 + 90p_3}. \end{aligned}$$

$$C(p) = c^\top d(p) = (20, 18, 22) \begin{pmatrix} 100 - 8p_1 - 1,5p_2 - p_3 \\ 80 - 1,5p_1 - 6p_2 - 1,5p_3 \\ 90 - p_1 - 1,5p_2 - 7p_3 \end{pmatrix}$$

$$C(p) = (2000 - 160p_1 - 30p_2 - 20p_3) + (1440 - 29p_1 - 108p_2 - 27p_3) + (1980 - 22p_1 - 33p_2 - 154p_3)$$

$$C(p) = \boxed{5420 - 49p_1 - 141p_2 - 181p_3}.$$

El beneficio resulta

$$\begin{aligned} B(p) &= I(p) - C(p) \\ &= -8p_1^2 - 6p_2^2 - 7p_3^2 - 3p_1p_2 - 2p_1p_3 - 3p_2p_3 \\ &\quad + (100 + 209)p_1 + (80 + 171)p_2 + (90 + 201)p_3 - 5420 \\ &= \boxed{-8p_1^2 - 6p_2^2 - 7p_3^2 - 3p_1p_2 - 2p_1p_3 - 3p_2p_3 + 149p_1 + 221p_2 + 271p_3 - 5420}. \end{aligned}$$

Gradiente y Hessiano de  $B(p)$ :

$$\nabla B(p) = \begin{pmatrix} \partial B / \partial p_1 \\ \partial B / \partial p_2 \\ \partial B / \partial p_3 \end{pmatrix} = \begin{pmatrix} -16p_1 - 3p_2 - 2p_3 + 149 \\ -3p_1 - 12p_2 - 3p_3 + 221 \\ -2p_1 - 3p_2 - 14p_3 + 271 \end{pmatrix}.$$

$$\nabla^2 B = \begin{pmatrix} -16 & -3 & -2 \\ -3 & -12 & -3 \\ -2 & -3 & -14 \end{pmatrix}.$$

$$D_1 = \det(-16) = -16 < 0, \quad D_2 = \det \begin{pmatrix} -16 & -3 \\ -3 & -12 \end{pmatrix} = 192 - 9 = 183 > 0,$$

$$D_3 = \det \begin{pmatrix} -16 & -3 & -2 \\ -3 & -12 & -3 \\ -2 & -3 & -14 \end{pmatrix} = -2406 < 0.$$

Los signos  $D_1 < 0$ ,  $D_2 > 0$ ,  $D_3 < 0$  alternan, por lo tanto el Hessiano es definido negativo. En consecuencia,  $B(p)$  es cóncava y el problema admite un máximo global único.