Programación Lineal
Análisis de sensibilidad
Clase 18

Investigación Operativa UTN FRBA 2021

Curso: I4051

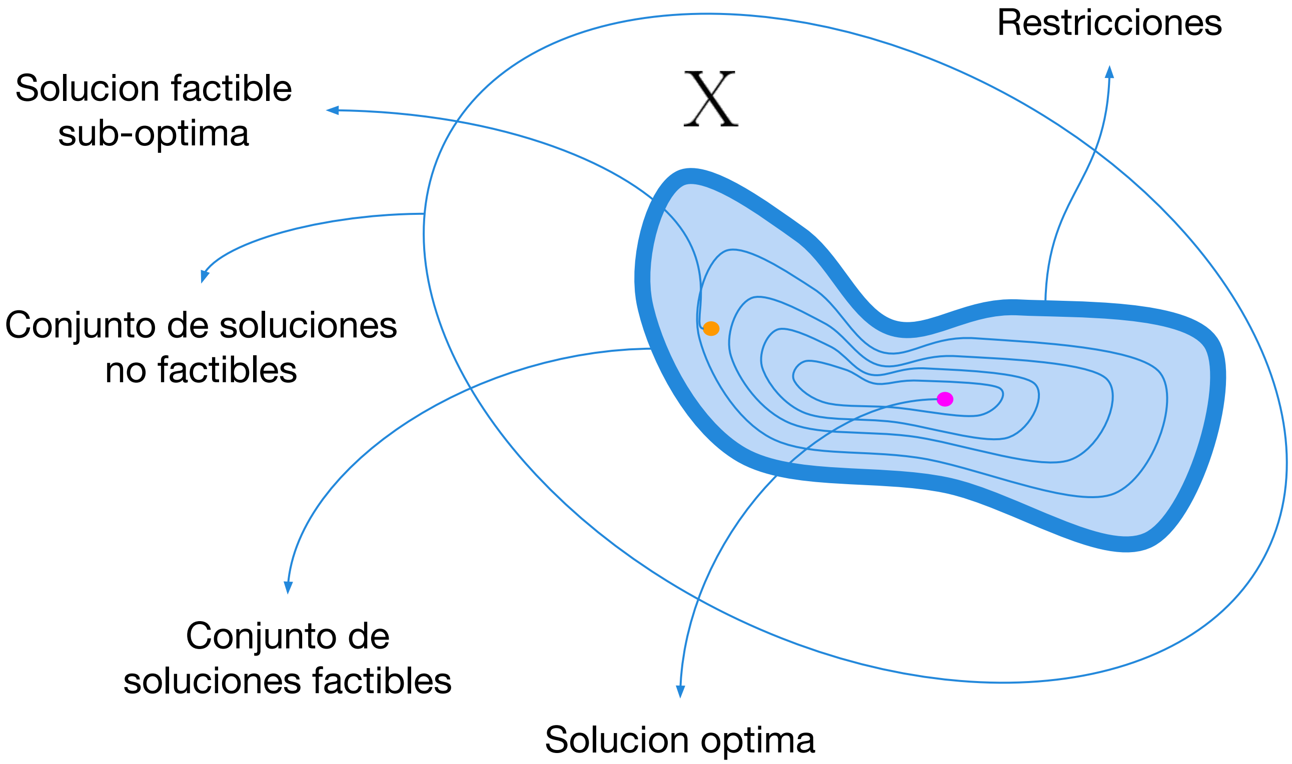
Docente: Martín Palazzo

Elementos centrales de un problema de optimización

Soluciones factibles: valores de la variable de decisión que satisfacen las restricciones.

Restricciones: conjunto de todas las soluciones factibles.

Solucion optima: conjunto de soluciones factibles que generan el valor máximo/mínimo de la función objetivo.



Elementos centrales de un problema de optimización

$$x_1, x_2, \dots, x_i, \dots, x_n$$

Variables de decision

$$f(x) = \quad c_1 x_1 + \dots + c_n x_n$$

Función objetivo a minimizar o maximizar

$$\begin{aligned}
g_1(x) &= \quad a_{11}x_1 \dots + a_{1n}x_n \quad &\le b_1 \\
g_2(x) &= \quad a_{21}x_1 \dots + a_{2n}x_n \quad &\le b_2 \\
\dots & \qquad\qquad \dots &\dots \\
g_m(x) &= \quad a_{m1}x_1 \dots + a_{mn}x_n \quad &\le b_m
\end{aligned}$$

Restricciones con desigualdad

$$q(x) = \quad q_1 x_1 + \dots + q_n x_n = d$$

Restricciones con igualdad

Programación Lineal

Un problema genérico de optimización a ser resuelto por programación lineal se estructura matricial y vectorialmente de la siguiente forma:

$$
\begin{aligned}
\mathbf{c}^T\mathbf{x} &= \mathbf{z} \longrightarrow \text{Max/min} \\
A\mathbf{x} &\leq \mathbf{b} \\
\mathbf{x} &\geq 0
\end{aligned}
$$

Donde **c** es un vector [n,1] de 'n' coeficientes, **x** es un vector [n,1] de 'n' variables de decisión, **A** es una matriz [m,n] de coeficientes (tecnológicos) sobre las 'n' variables de decisión **x** en las 'm' restricciones **b**. Tanto **c**, **b** y **A** son conocidos mientras que **x** es la incógnita multidimensional a averiguar. Notar que cuando n>3 nos encontramos con un problema de alta dimensión que no puede ser resuelto por el método gráfico.

Estructura de un problema de Programación Lineal

Función objetivo f(x)

$$\boxed{c_1x_1 + c2x_2 \quad = z}$$

Restricciones g(x)

$$\boxed{\begin{aligned} a_{11}x_1 + a_{12}x_2 &\leq b_1 \\ a_{21}x_1 + a_{22}x_2 &\leq b_2 \\ a_{31}x_1 + a_{32}x_2 &\geq b_3 \\ a_{41}x_1 + a_{42}x_2 &= b_4 \\ x_1, x_2 &\geq 0 \end{aligned}}$$

Programación Lineal

En caso que n = 2 y r = 2 el planteo del problema tendría la siguiente forma:

$$
\begin{aligned}
c_1x_1 + c_2x_2 &= \mathbf{z} \\
a_{11}x_1 + a_{12}x_2 &\leq b_1 \\
a_{21}x_1 + a_{22}x_2 &\leq b_2 \\
x_1, x_2 &\geq 0
\end{aligned}
$$

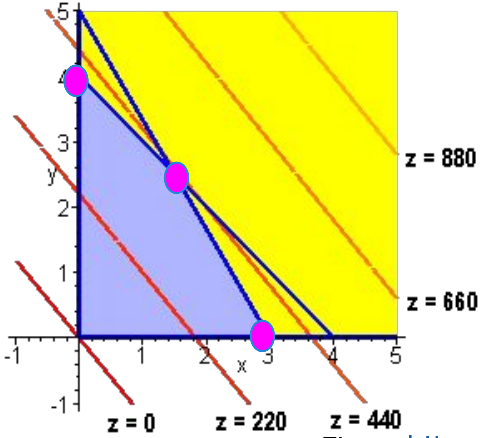


Figura https://services.math.duke.edu/

Donde el área azul contiene las soluciones básicas factibles (SBF) limitadas por las 2 restricciones **ax** =< **b** y por las condiciones de no-negatividad. La recta roja representa a la función objetivo **z** a optimizar y su valor estará en función de cuàn lejos del origen se encuentre. Los puntos en fucsia representan la soluciones en los vértices del área de SBF.

Análisis de Sensibilidad

Uno de los principales supuestos a la hora de solucionar un problema de programación lineal es considerar que el valor de los recursos b = [b1 , b2...bn] y los coeficientes del funcional c = [c1, c2...cm] son constantes.

El supuesto anterior permite preguntarse cuánto se modificaría la solución del funcional si se modifican los coeficientes mencionados. Para responder a esa pregunta podemos hacer un **Análisis de Sensibilidad** y determinar cuánto podemos modificar estos coeficientes ahora vistos como hiper-parámetros que el usuario podría ingresar externamente. Así podemos ver cómo responde nuestro problema, si la solución cambia o no, etc.

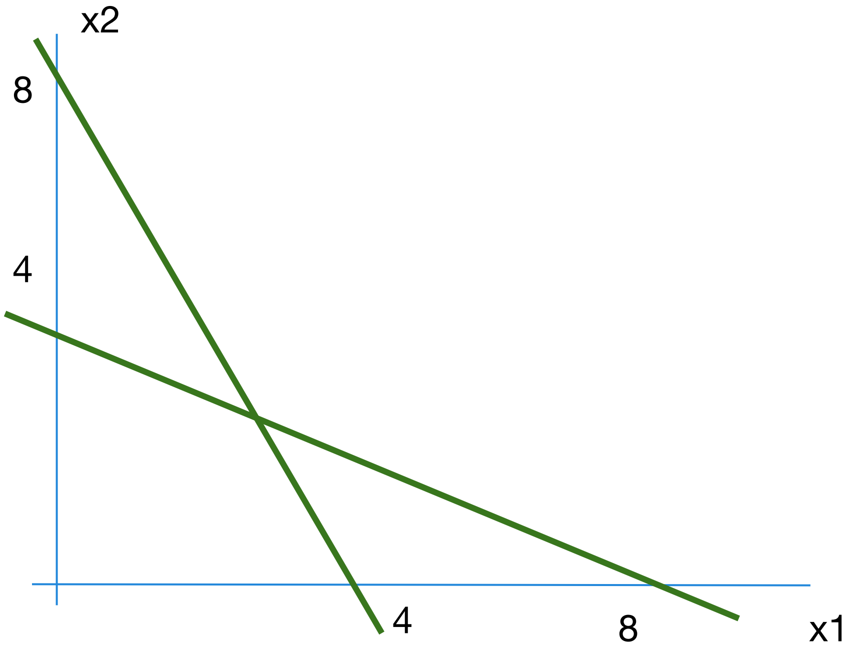
Análisis de Sensibilidad: ejemplo

Supongamos el siguiente problema de PL:

	Producto 1	Producto 2	Disponibilidad
Recurso 1	2	1	8
Recurso 2	1	2	8
Utilidad	300	200	

$$\begin{aligned} 300x_1 + 200x_2 &= z \\ 2x_1 + x_2 &\leq 8 \\ x_1 + 2x_2 &\leq 8 \end{aligned}$$

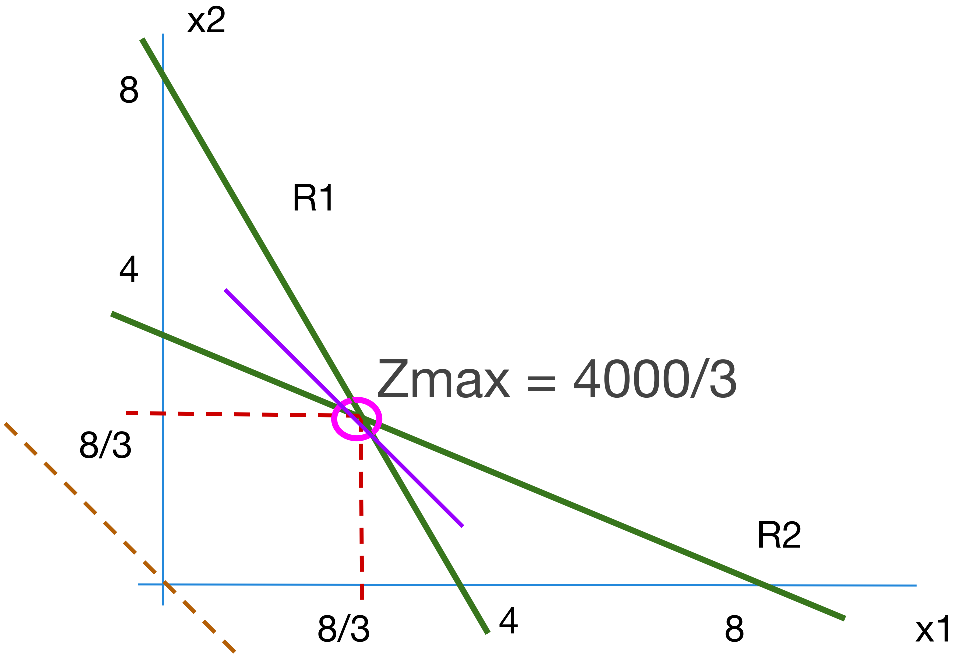
Análisis de Sensibilidad: ejemplo



$$
\begin{aligned}
300x_1 + 200x_2 &= z \\
2x_1 + x_2 &\leq 8 \\
x_1 + 2x_2 &\leq 8
\end{aligned}
$$

Podemos graficar la restricciones.

Análisis de Sensibilidad: ejemplo



$$
\begin{aligned}
300x_1 + 200x_2 &= z \\
2x_1 + x_2 &\leq 8 \\
x_1 + 2x_2 &\leq 8
\end{aligned}
$$

Sencillamente podríamos encontrar el máximo global del problema.

Solucion en Python

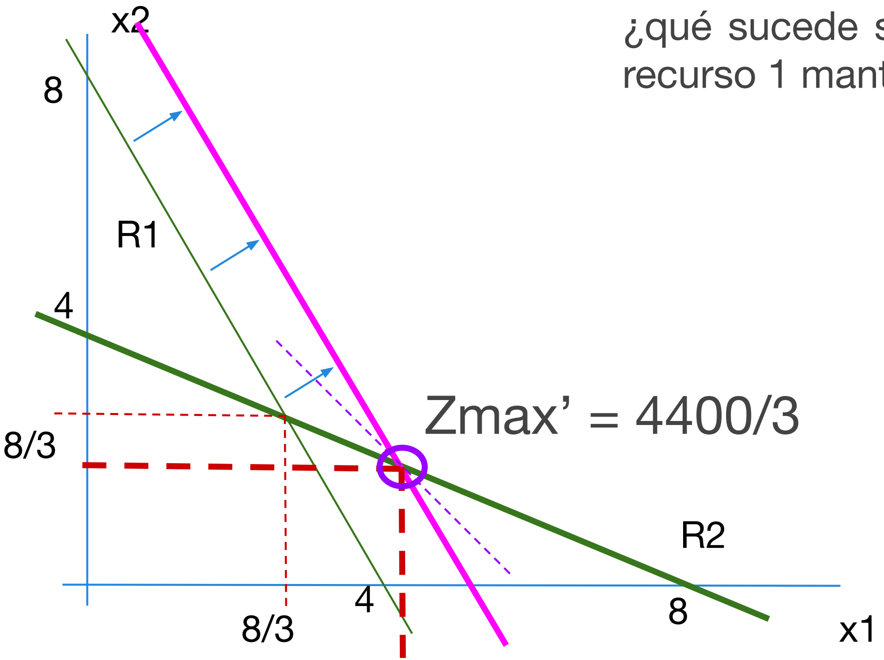
```python
# definimos si es un problema de minimizacion o maximizacion
prob1 = LpProblem("Toy example 1", LpMaximize)

# definimos las variables de decision
x1 = LpVariable('x1', lowBound=0, cat='Continuous')
x2 = LpVariable('x2', lowBound=0, cat='Continuous')

# primero agregamos la funcion objetivo
prob1 += 300 * x1 + 200 * x2, "Funcion objetivo"

prob1 += 2*x1 + 1*x2 <= 8, "restriccion 1"
prob1 += 1 * x1 + 2 * x2 <= 8, "restriccion 2"

# Resolver el problema con el solver de PULP
prob1.solve()
```

$$
\begin{aligned}
300x_1 + 200x_2 &= z \\
2x_1 + x_2 &\leq 8 \\
x_1 + 2x_2 &\leq 8
\end{aligned}
$$

Análisis de sensibilidad en programación lineal: shadow price

Modificando los parámetros b1, b2 de tope de restricciones

Análisis de Sensibilidad: precio sombra (shadow price)



¿qué sucede si aumentamos una unidad la disponibilidad del recurso 1 manteniendo los demás parámetros constantes?

$$\begin{aligned} 300x_1 + 200x_2 &= z \\ 2x_1 + x_2 &\leq 9 \\ x_1 + 2x_2 &\leq 8 \end{aligned}$$

Análisis de Sensibilidad: precio sombra (shadow price)

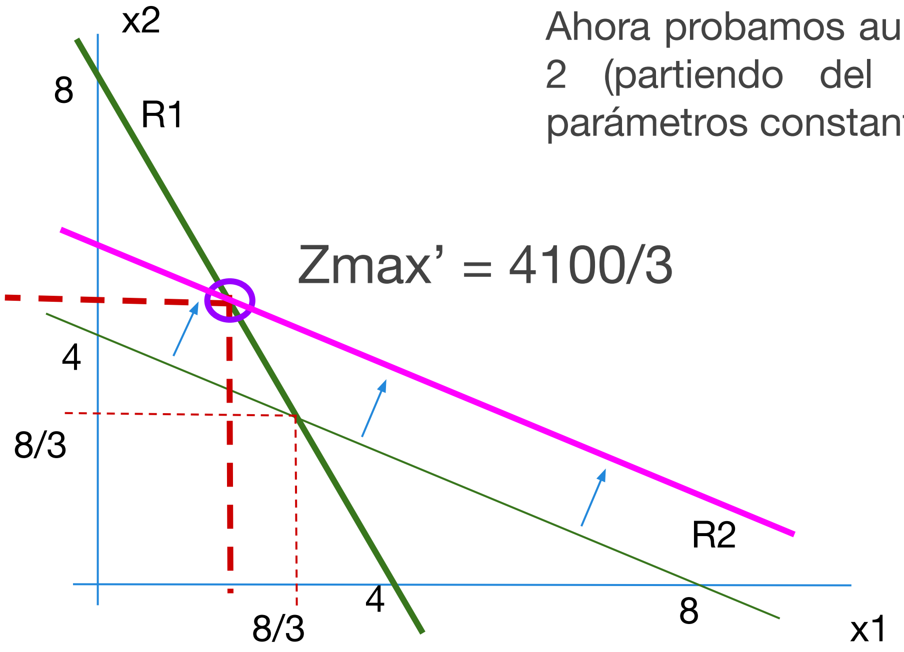
Introducimos así el precio sombra (shadow price) para cada **recurso**. Representa el incremento marginal del funcional Z por cada unidad adicional de recurso que se encuentre disponible.

$$\text{Shadow Price } b_i = \frac{\Delta Z}{\Delta b_i}$$

Entonces para el recurso 1 (b1) calculamos su precio sombra correspondiente

$$\text{Shadow Price } b_1 = \frac{\Delta Z}{\Delta b_1} = \frac{4400/3 - 4000/3}{9 - 8} = 133\$/u$$

Análisis de Sensibilidad: precio sombra (shadow price)



Ahora probamos aumentar una unidad la disponibilidad del recurso 2 (partiendo del problema original) manteniendo los demás parámetros constantes.

$$
\begin{aligned}
300x_1 + 200x_2 &= z \\
2x_1 + x_2 &\leq 8 \\
x_1 + 2x_2 &\leq 9
\end{aligned}
$$

Análisis de Sensibilidad: precio sombra (shadow price)

Incrementado el recurso 2 en una unidad podemos calcular su precio sombra correspondiente.

$$\text{Shadow Price } b_2 = \frac{\Delta Z}{\Delta b_2} = \frac{4100/3 - 4000/3}{9 - 8} = 33\$/u$$

Análisis de Sensibilidad: precio sombra (shadow price)

Dados los precios sombras de ambos recursos (ejemplo, máquinas), si tuviéramos que incrementar la capacidad de uno, ¿a cuál priorizamos?

$$\text{Shadow Price } b_1 \quad = 133\$/u$$
$$\text{Shadow Price } b_2 \quad = 33\$/u$$

Si el costo de incrementar la disponibilidad de cualquier recurso en una unidad es:

- $200 -> ?
- $40 -> ?
- $30 -> ?

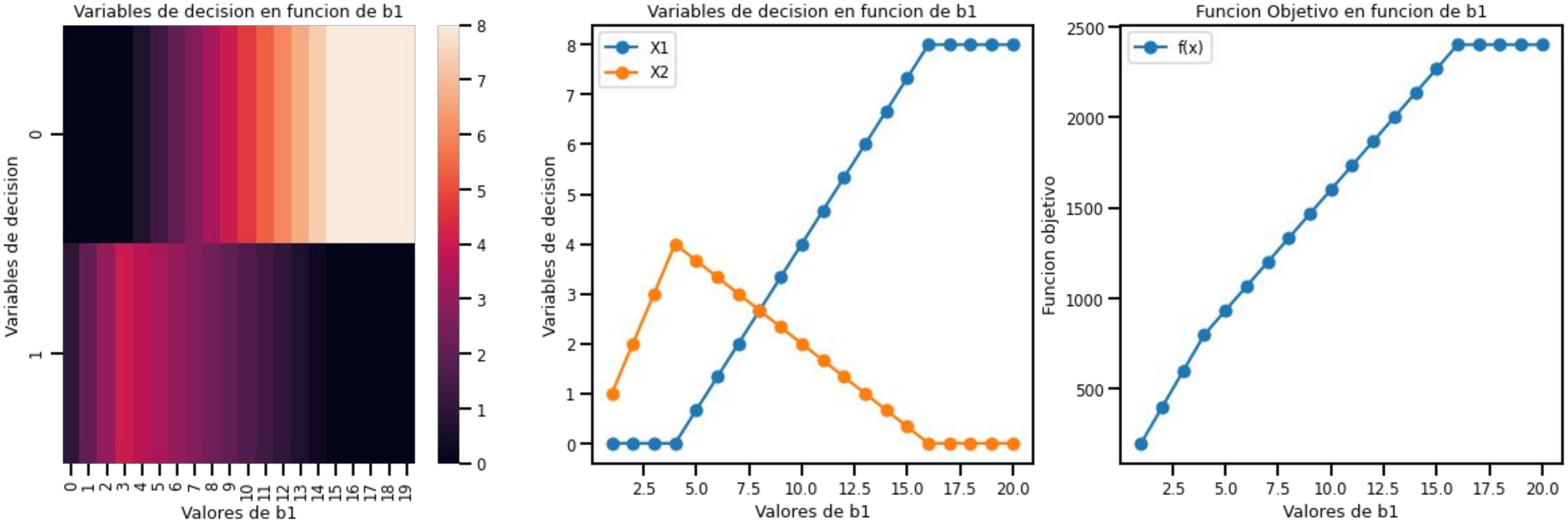
¿que decisión deberíamos tomar en cada caso?

Análisis de sensibilidad con b1 en python

```python
# creo una matriz vacia para ir guardando las variables de decision en cada prueba parametrica
x_param_analysis_b1 = np.zeros((20,2))

# creo un vector vacio para ir guardando la funcion objetivo en cada prueba parametrica
sol_param_b1 = np.zeros((20))

# contador para ir contando cuantas iteraciones hice
counter = 0
for i in range(1,21):

  # definimos si es un problema de minimizacion o maximizacion
  prob_sens_b1 = LpProblem("Sensitivity analysis", LpMaximize)

  # definimos las variables de decision
  x1 = LpVariable('x1', lowBound=0, cat='Continuous')
  x2 = LpVariable('x2', lowBound=0, cat='Continuous')

  # primero agregamos la funcion objetivo
  prob_sens_b1 += 300 * x1 + 200 * x2, "Funcion objetivo"

  # definimos restricciones
  prob_sens_b1 += 2*x1 + 1*x2 <= i, "restriccion 1"
  prob_sens_b1 += 1 * x1 + 2 * x2 <= 8, "restriccion 2"

  # Resolver el problema con el solver de PULP
  prob_sens_b1.solve()

  # valor de la funcion objetivo
  sol_param_b1[counter] = value(prob_sens_b1.objective)

  # obtenemos el valor de la variable de decision X1 en el punto optimo
  x_param_analysis_b1[counter,0] = prob_sens_b1.variables()[0].varValue
  x_param_analysis_b1[counter,1] = prob_sens_b1.variables()[1].varValue

  counter += 1
  print("solve iteration " + str(i))
```

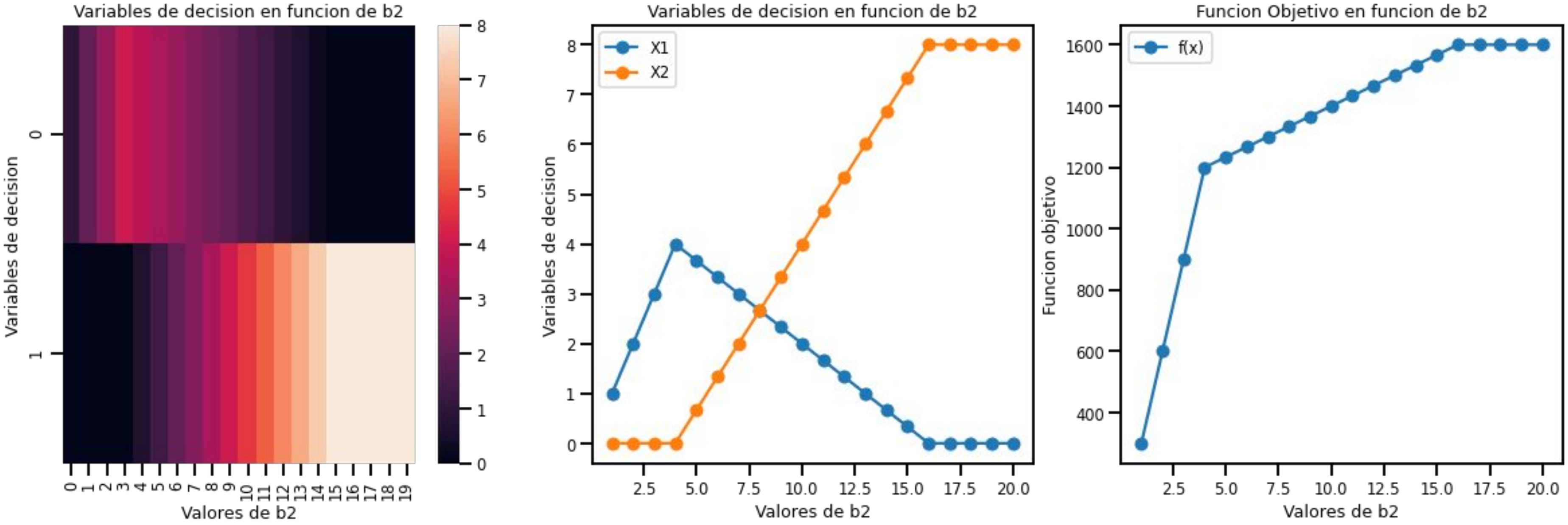
$$\begin{aligned} 300x_1 + 200x_2 \quad &= z \\ 2x_1 + 1x_2 \quad &\leq b_1 = 1, 2, ..., 10 \\ 1x_1 + 2x_2 \quad &\leq b_2 = 1, 2, ..., 10 \end{aligned}$$

Análisis de sensibilidad con b1 en python



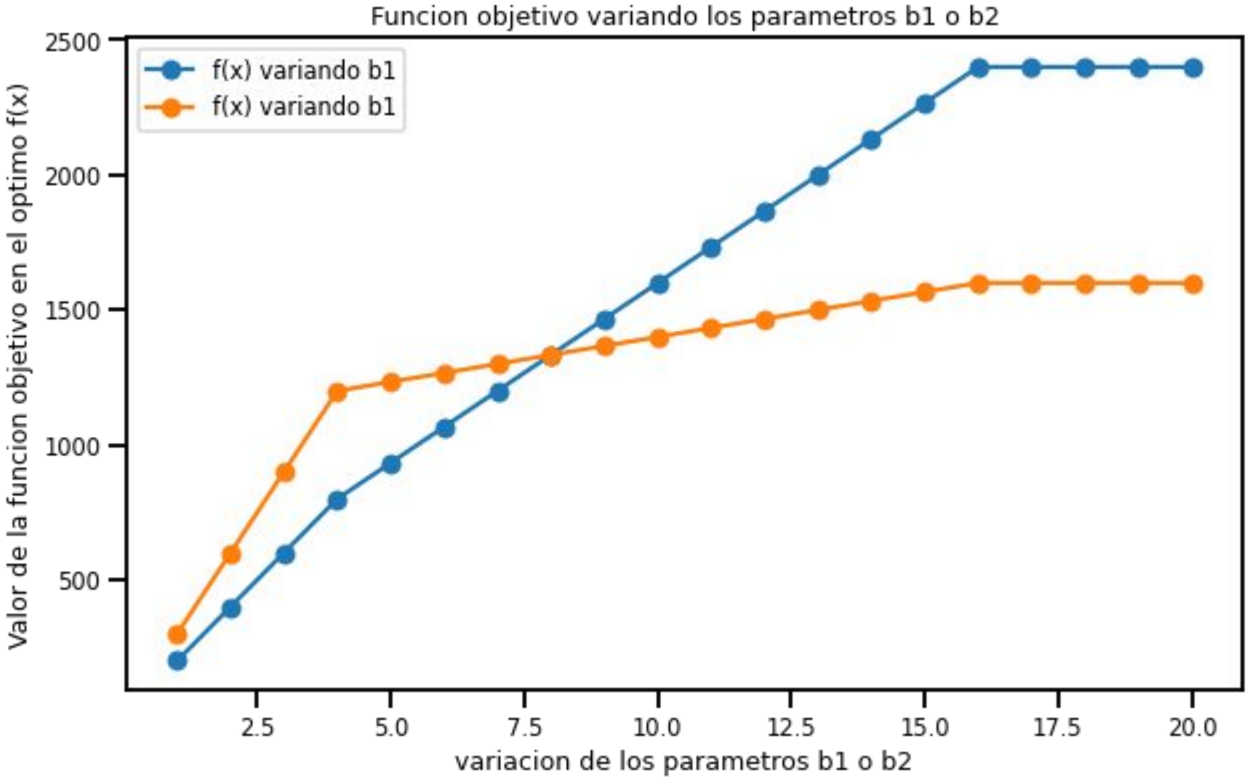
Analisis de sensibilidad al hacer un barrido parametrico del parametro b1 (limite de un recurso/restricciones)

Análisis de sensibilidad con b2 en python



Analisis de sensibilidad al hacer un barrido parametrico del parametro b2 (limite de un recurso/restricciones)

Funcion objetivo variando cada uno de los parametros b1 b2



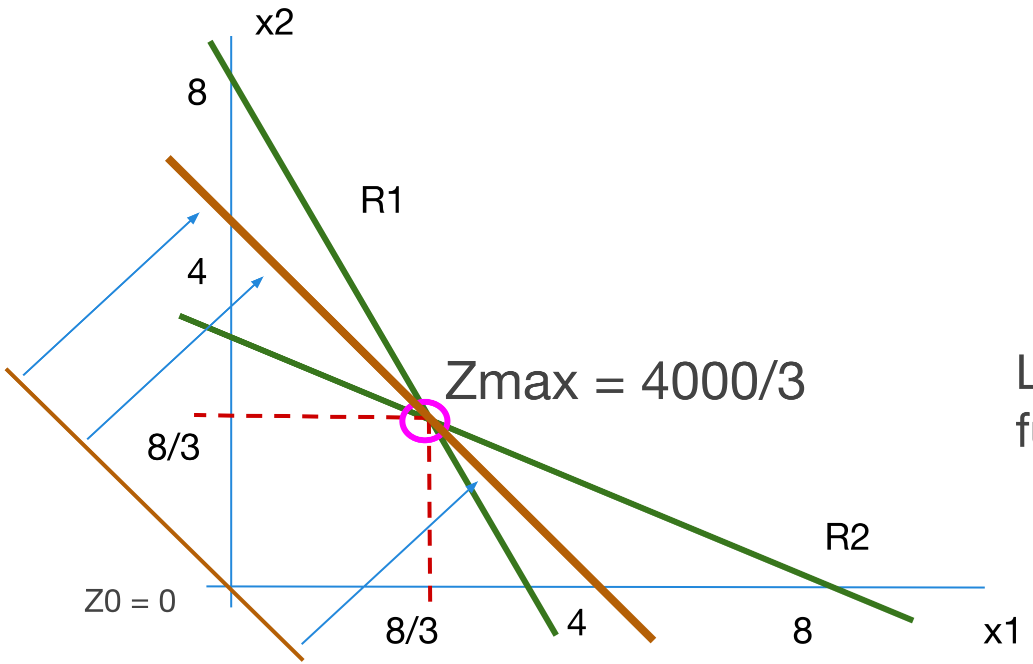
Analisis de sensibilidad al hacer un barrido paramétrico del parametro b1 o b2 (límite de un recurso/restricciones) en la funcion objetivo.

Análisis de sensibilidad en programación lineal

Modificando los parámetros c_1, c_2 de la función objetivo

Análisis de Sensibilidad: coeficientes de Z

Otro tipo de análisis de sensibilidad se centra en el valor de los coeficientes del funcional, lo que comúnmente representa la utilidad de cada unidad de producto.



$$
\begin{aligned}
c_1x_1 + c_2x_2 &= z \\
a_{11}x_1 + a_{12}x_2 &\leq b_1 \\
a_{21}x_1 + a_{22}x_2 &\leq b_2
\end{aligned}
$$

La pendiente de la recta que define al funcional estará determinada por:

$$m_z = \frac{-c_1}{c_2}$$

Análisis de Sensibilidad: coeficientes de Z

Lo que interesa saber es si al modificar los coeficientes del funcional manteniendo los demás parámetros constantes la solución óptima (en valores de X1 y X2) se modifica o no. Para eso debemos saber en que rango de valores puede variar la pendiente mz de la función objetivo.

$$a_{11}x_1 + a_{12}x_2 \leq b_1 \mapsto x_2 = \frac{b_1}{a_{12}} - \boxed{\frac{a_{11}}{a_{21}}}x_1$$

$$a_{21}x_1 + a_{22}x_2 \leq b_2 \mapsto x_2 = \frac{b_2}{a_{22}} - \boxed{\frac{a_{21}}{a_{22}}}x_1$$

Lo que determina el rango de valores de mz que mantiene la función objetivo igual son las pendientes de las restricciones.

Análisis de Sensibilidad: coeficientes de Z

Volviendo al ejercicio original obtenemos las pendientes de las restricciones

$$2x_1 + 1x_2 \ \leq 8 \ \mapsto x_2 = \frac{8}{1} - \frac{2}{1}x_1$$

$$1x_1 + 2x_2 \ \leq 8 \ \mapsto x_2 = \frac{8}{2} - \frac{1}{2}x_1$$

Y obtenemos el rango de valores en que la pendiente del funcional puede variar sin cambiar el resultado óptimo.

$$-2 \leq \frac{-c_1}{c_2} \leq -1/2$$

Análisis de Sensibilidad: coeficientes de Z

¿que pasaría si las utilidades de los productos 1 y 2 cambian a $c1 = 350$ y $c2 = 250$?

$$300x_1 + 200x_2 = z \longrightarrow z = 350x_1 + 250x_2$$

¿la solución en cantidades de x1 y x2 es la misma?

$$-2 \leq -\frac{350}{250} = -\frac{7}{5} \leq -1/2$$

Como la pendiente del nuevo funcional se mantiene en el rango de pendientes calculado desde las restricciones la solución óptima será la misma.

Análisis de Sensibilidad: coeficientes de Z

¿que pasaría si c2 = 200 queda fijo según el problema original y se quiere conocer el rango de valores de utilidad de c1 que no modifican la solución óptima?

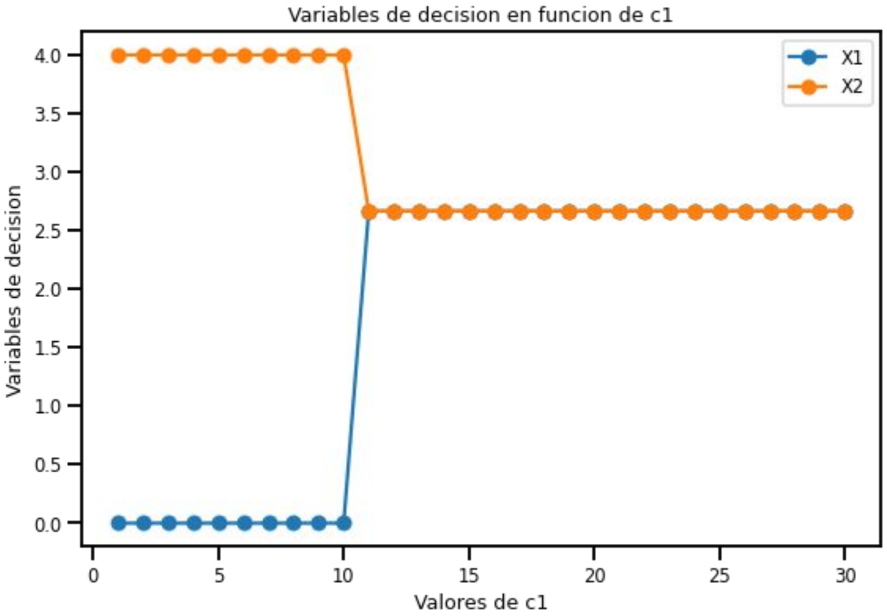
$$300x_1 + 200x_2 = z \longrightarrow z = c_1x_1 + 200x_2$$

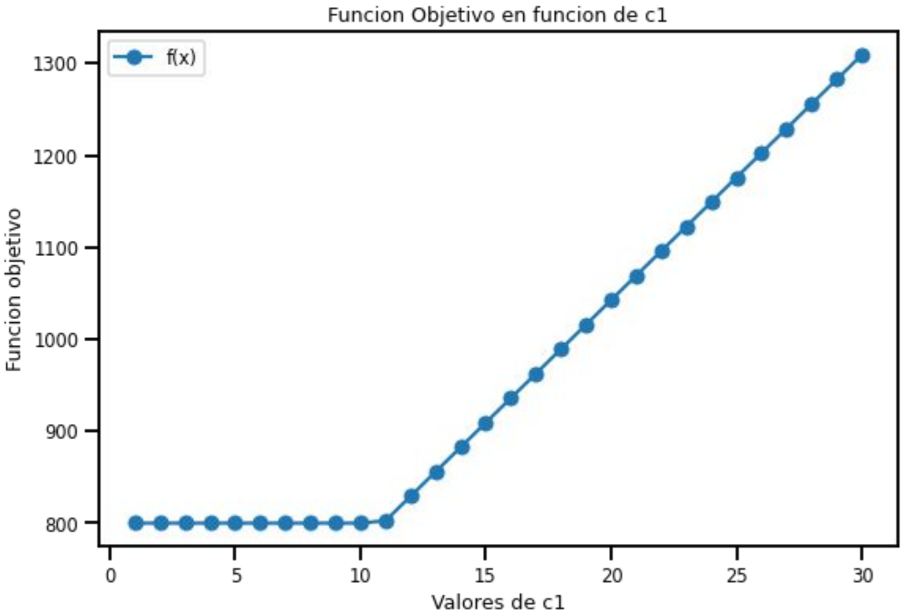
El rango de valores de c1 quedará determinado

$$-2 \leq -\frac{c_1}{c_2} = -\frac{c_1}{200} \leq -1/2 \mapsto c_1 = ?$$

$$\boxed{100 \leq c_1 \leq 400}$$

Analisis de sensibilidad barriendo C1





Analisis de sensibilidad al hacer un barrido paramétrico del parametro c1 (coeficiente de la funcion objetivo) en la función objetivo.

Analisis de sensibilidad: caso nutricion

Nutrition problem

Consideremos el problema de elegir comida preparada para alcanzar ciertos requisitos nutricionales. Supongamos que las cenas preparadas están disponibles al siguiente precio:

Carne: $390 , Pollo: $290, Pescado: $300, Roll vegetariano: $250, Tarta de queso: $230, Pasta: $100

Estos platos estan detallan porcentajes por plato de la cantidad de vitaminas diarias que aporta cada porcion. Cada semana se necesita cubrir un 700% de cada vitamina.

	Vita. A	Vita. C	Vita. B1	Vita. B2
Carne	40%	15%	10%	15%
Pollo	20	0	20	20
Pescado	20	10	15	10
Roll Vegetariano	5	20	10	10
Tarta de queso	15	5	15	20
Pasta	5	5	10	10

Utilizando programación matemática modelar el problema para determinar que combinación de alimentos debe incluir el plan de nutrición de manera de satisfacer todas las necesidades de vitaminas al menor costo.

A modelar el problema

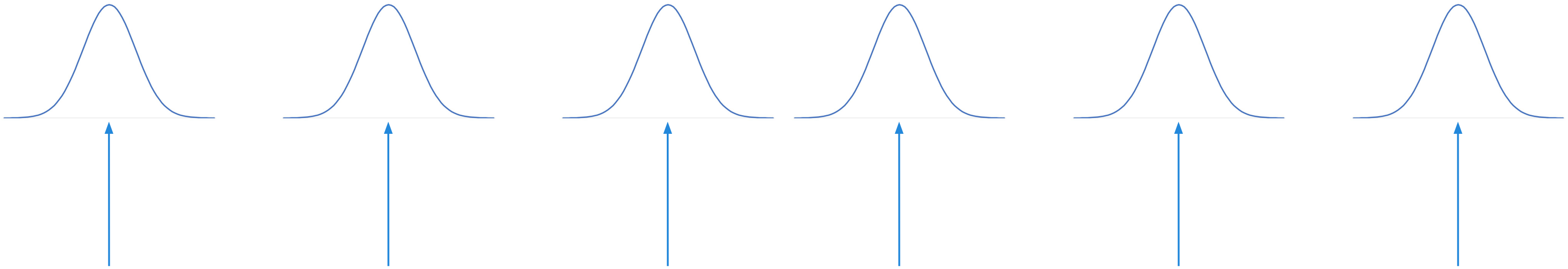
- Variables de decision?
- Función objetivo?
- Restricciones?
- Tipo de problema de optimización?

Modelización del problema

$$z = 390x_1 + 290x_2 + 300x_3 + 250x_4 + 230x_5 + 100x_6$$

$$40x_1 + 20x_2 + 20x_3 + 5x_4 + 15x_5 + 5x_6 \geq 700$$
$$15x_1 + 0x_2 + 10x_3 + 20x_4 + 5x_5 + 5x_6 \geq 700$$
$$10x_1 + 20x_2 + 15x_3 + 10x_4 + 15x_5 + 10x_6 \geq 700$$
$$15x_1 + 20x_2 + 10x_3 + 10x_4 + 20x_5 + 10x_6 \geq 700$$

Analisis de sensibilidad perturbando coeficientes



$$z = 390x_1 + 290x_2 + 300x_3 + 250x_4 + 230x_5 + 100x_6$$

Es posible tambien afectar con ruido generado desde una distribucion normal a los parametros c1….c6 y observar que sucede con la funcion objetivo si simultaneamente perturbamos estos coeficientes.

Análisis de sensibilidad en Python

El analisis de sensibilidad contempla

- Modificacion de parametros de tope de restricciones b1..bn
- Modificacion de coeficientes de la funcion objetivo c1….cn

https://github.com/investigacion-operativa/pyOperativ/blob/main/07_programacion_matematica/casos_codigo/ioperativ_clase18_linprog_analisis_sensibilidad.ipynb