

# Introducción al uso de software de optimización

## Sesión 1

Dr. Norberto Castillo García

25 de octubre de 2023

# Contenido

1. Introducción a la optimización

2. Aplicaciones de la optimización

3. Software de optimización

4. Ejercicios

# Introducción a la optimización

*Norberto*  
Castillo García

# Definiciones

## Optimización

Es un proceso que consiste en **seleccionar el mejor elemento** de un conjunto de elementos disponibles con respecto a algún criterio.

# Definiciones (continuación)

## Problema de optimización

Es un problema en el que hay **muchas posibles soluciones** y alguna forma clara de **comparación** entre ellas.

# Definiciones (continuación)

## Programación lineal

Es el campo de la optimización dedicado a **maximizar** o **minimizar** una función lineal, denominada **función objetivo**, de tal forma que las **variables** de dicha función están sujetas a una serie de **restricciones** expresadas mediante un sistema de ecuaciones o inecuaciones (o una combinación de ambas) también lineales.

# Modelos de Programación Lineal

## Modelo general:

Maximizar o minimizar **Función objetivo**  
sujeto a:  
**Restricciones**

## Ejemplo:

minimizar:  $z = 3x_1 - 2x_2$

sujeto a:

$$5x_1 + 4x_2 \leq 40$$

$$5x_1 + 2x_2 \geq 20$$

$$x_1 + 5x_2 \geq 10$$

$$x_1, x_2 \geq 0$$

# Variables de Decisión

Fundamentalmente, las variables de decisión se pueden dividir en tres tipos:

- ▶ **Continuas.** Pueden tomar cualquier valor real (e.g., 1.5,  $-10.84$  o  $\pi \approx 3.1416$ ).
- ▶ **Enteras.** Pueden tomar valores no fraccionarios (e.g., 1 o 2).
- ▶ **Binarias.** Pueden tomar valores de cero o uno.

**NOTA:** Por lo general, las variables de decisión se denotan con la letra  $x$  (en itálica) y un subíndice, por ejemplo:  $x_1$ .



# Función Objetivo

En programación lineal, las funciones objetivo son **funciones reales de varias variables** y éstas deben ser **lineales**.

► Ejemplos de funciones objetivo **lineales**:

»  $f(x_1, x_2) = x_1 + x_2$

»  $f(x_1, x_2, x_3) = 3x_1 - 2x_2 + 7x_3$

► Ejemplos de funciones objetivo **no lineales**:

»  $f(x_1, x_2) = 3x_1^2 - 7\sqrt{x_2}$

»  $f(x_1, x_2) = \ln(x_1 + x_2)$

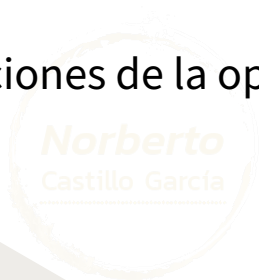
**Nota:** Una función objetivo se desea maximizar o minimizar.

# Restricciones

Las restricciones pueden ser de tres tipos:

- ▶ **Menor o igual.** Por ejemplo:  $x_1 + x_2 \leq 30$
- ▶ **Mayor o igual.** Por ejemplo:  $x_1 + x_2 \geq 0$
- ▶ **Igual.** Por ejemplo:  $x_1 + x_2 = 1$

# Aplicaciones de la optimización



# ¿En donde se aplica la optimización?

- ▶ Toma de decisiones en proyectos de inversión.
- ▶ Programación de horarios en escuelas.
- ▶ Programación de quirófanos (cirugías) en hospitales.
- ▶ Secuenciación de cortado de cables UTP para cableado estructurado.
- ▶ Programación de recursos materiales y humanos en una terminal portuaria.
- ▶ Encontrar la mejor ruta para las empresas de reparto (p.ej., DHL o estafeta).
- ▶ Reducir los costos de producción y las emisiones de contaminantes en la generación de energía eléctrica.
- ▶ ¡Y muchas más!
- ▶ ¿Conoces alguna otra?

# Ejemplo de aplicación

- ▶ Una empresa se dedica a la fabricación de ropa. El próximo mes planea fabricar camisas y pantalones. Cada camisa se vende en \$135 y cada pantalón en \$290. Para fabricar una camisa se requiere de  $2.5 \text{ m}^2$  de tela y 1.5 horas-máquina. Para fabricar un pantalón se requiere de  $4 \text{ m}^2$  de tela y 2.5 horas-máquina. En total se dispone de  $300 \text{ m}^2$  de tela y 120 horas-máquina al mes.
- ▶ ¿Cuál es la cantidad de camisas y pantalones que se deben fabricar al mes para que la utilidad esperada sea máxima?

# Ejemplo de aplicación (continuación)

## Variables de decisión:

- ▶  $x_1$  cantidad de **camisas** a producir mensualmente.
- ▶  $x_2$  cantidad de **pantalones** a producir mensualmente.

## Modelo de PL:

maximizar:

$$z = 135x_1 + 290x_2$$

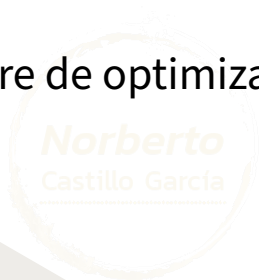
sujeto a:

$$2.5x_1 + 4x_2 \leq 300$$

$$1.5x_1 + 2.5x_2 \leq 120$$

$$x_1, x_2 \geq 0 \text{ y enteras}$$

# Software de optimización



# Diferentes motores de optimización

## ► Software libre (GNU):

- » GLPK ([www.gnu.org/software/glpk/](http://www.gnu.org/software/glpk/))
- » Lpsolve ([www.sourceforge.net/projects/lpsolve/](http://www.sourceforge.net/projects/lpsolve/))
- » COIN-OR ([www.coin-or.org/](http://www.coin-or.org/))

## ► Software propietario:

- » Lindo ([www.lindo.com/](http://www.lindo.com/))
- » Gurobi ([www.gurobi.com/](http://www.gurobi.com/))
- » CPLEX ([www.ibm.com/mx-es/analytics/cplex-optimizer](http://www.ibm.com/mx-es/analytics/cplex-optimizer))



# ¿Qué es CPLEX?

- ▶ Es un software de optimización.
- ▶ Resuelve problemas de programación lineal, programación entera y programación cuadrática, entre otros.
- ▶ Ofrece interfaces a otros lenguajes de programación como C++, C# y Java (a través de API's).
- ▶ Desafortunadamente, es un software comercial (de paga).
- ▶ Sin embargo, para fines académicos es completamente gratuito, gracias a la iniciativa académica de IBM.

# Declaración del modelo

En la API de CPLEX para **Java**, un modelo se declara de la siguiente manera:

```
IloCplex modelo = new IloCplex();
```

Donde `modelo` es el nombre que le asignamos a nuestro modelo.

# Declaración de variables

Para declarar una variable se puede utilizar la siguiente instrucción:

```
IloNumVar x = modelo.numVar(0.0, 10.0, IloNumVarType.Float, "x");
```

Cota inferior de la variable

Cota superior de la variable

Tipo de variable

Nombre de la variable

# Declaración de expresiones

- ▶ Las expresiones se componen uno o más términos algebraicos.
- ▶ Si en una expresión todas las variables están elevadas a la potencia 1, entonces se denomina **expresión lineal**.
- ▶ Una expresión lineal es del tipo:  $a_1x_1 \pm a_2x_2 \pm \dots \pm a_nx_n$ .

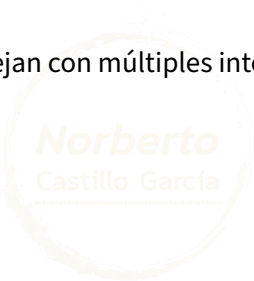
## Ejemplos:

- »  $3x_1 - 2x_2$
- »  $135x_1 + 290x_2$
- »  $x_1 - 8x_2 + 4x_3$
- ▶ Si en una expresión hay una multiplicación de dos variables, entonces es una **expresión cuadrática**, por ejemplo:  $5x_1^2 - 3x_1x_2 + 11x_2^2$ .
- ▶ Las expresiones son importantes ya que con ellas se construyen tanto en la función objetivo como las restricciones.

# Declaración de expresiones (continuación)

En CPLEX, las expresiones se manejan con múltiples interfaces y subinterfaces. Algunos ejemplos son:

- ▶ `IloNumExpr`
- ▶ `IloLinearNumExpr`
- ▶ `IloIntExpr`



# Declaración de expresiones (continuación)

Una expresión se puede declarar utilizando **IloLinearNumExpr** y el método **addTerm** de la siguiente manera:

```
IloLinearNumExpr exprLineal = modelo.linearNumExpr();  
exprLineal.addTerm(c, x);
```

variable tipo double

Objeto de tipo IloNumVar

**Nota:** El método **addTerm** agrega un sólo término a la expresión.

# Declaración de expresiones (continuación)

La expresión anterior también puede ser ingresada utilizando el método **scalProd** (producto escalar):

```
modelo.scalProd(c, x);
```

Array de tipo double (coeficientes)

Array de la clase IloNumVar (variables)

# Declaración de la función objetivo

Para declarar una función objetivo podemos usar cualquiera de las siguientes declaraciones:

► Para funciones objetivo de **maximización**:

- » `modelo.addMaximize(IloNumExpr ine);`
- » `modelo.add(modelo.maximize(IloNumExpr ine));`

► Para funciones objetivo de **minimización**:

- » `modelo.addMinimize(IloNumExpr ine);`
- » `modelo.add(modelo.minimize(IloNumExpr ine));`



# Declaración de las restricciones

- ▶ Para declarar restricciones de tipo **menor o igual** ( $\leq$ ):
  - » `modelo.addLe(IloNumExpr ine, double d, String str)`
- ▶ Para declarar restricciones de tipo **mayor o igual** ( $\geq$ ):
  - » `modelo.addGe(IloNumExpr ine, double d, String str)`
- ▶ Para declarar restricciones de tipo **igual** ( $=$ ):
  - » `modelo.addEq(IloNumExpr ine, double d, String str)`

# Resolución del modelo

- ▶ Para resolver el modelo de Programación Lineal, se utiliza el método **solve** de la siguiente manera:
  - » `modelo.solve()`;
- ▶ Este método retorna un valor booleano.
- ▶ Si el valor de retorno es **true**, entonces el modelo pudo resolverse.
- ▶ Si el valor de retorno es **false**, entonces no fue posible resolver el modelo.

# Recuperación de los valores de interés

- ▶ Para obtener el valor objetivo:

- » `modelo.getObjValue()`;

- ▶ Para obtener el valor de la variable  $x$  (de tipo `IloNumVar`):

- » `modelo.getValue(x)`;

# Ejercicios



# Ejercicio 1

Resolver el siguiente modelo de Programación Lineal utilizando la API de CPLEX para Java:

maximizar:  $z = 3x_1 - 2x_2$

sujeto a:

$$5x_1 + 4x_2 \leq 40$$

$$5x_1 + 2x_2 \geq 20$$

$$x_1 + 5x_2 \geq 10$$

$$x_1, x_2 \geq 0$$

## Ejercicio 2

Resolver el siguiente modelo de Programación Lineal utilizando la API de CPLEX para Java:

maximizar:  $z = 135x_1 + 290x_2$

sujeto a:

$$2.5x_1 + 4x_2 \leq 300$$

$$1.5x_1 + 2.5x_2 \leq 120$$

$$x_1, x_2 \geq 0 \quad \text{y enteras}$$

## Ejercicio 3

Resolver el siguiente modelo de Programación Lineal utilizando la API de CPLEX para Java:

maximizar:  $z = 325x_1 + 400x_2 + 100x_3 + 540x_4 + 270x_5$

sujeto a:

$$118x_1 + 232x_2 + 63x_3 + 325x_4 + 78x_5 \leq 500$$

$$x_1, x_2, x_3, x_4, x_5 \text{ binarias}$$

# Ejercicio 4

Resolver el siguiente modelo de Programación Cuadrática utilizando la API de CPLEX para Java:

maximizar:  $z = 4x_1 + 6x_2 - 2x_1^2 - 2x_1x_2 - 2x_2^2$

sujeto a:

$$x_1 + 2x_2 \leq 2$$
$$x_1, x_2 \geq 0$$



# Referencias I



J. P. Ignizio and T. M. Cavalier.

*Linear programming.*

Prentice-Hall, Inc., 1994.



A. D. Muñoz.

*Metaheurísticas*, volume 22.

Librería-Editorial Dykinson, 2007.



H. A. Taha.

*Investigación de operaciones.*

Pearson Educación, 2004.

