

Diseño de un modelo de optimización¹

Ángel Luna Ángel Rubén

Instituto Tecnológico y de Estudios Superiores de Monterrey

15 de febrero de 2026

¹<https://github.com/Lunaaaalj/Diseno-de-un-modelo-de-optimizacion>

Problema

Un país pequeño se encuentra en guerra y actualmente cuenta únicamente con dos fábricas activas para abastecer al ejército. Las prioridades estratégicas establecidas por el alto mando militar son la producción de armas, alimentos y municiones.

Los recursos disponibles permiten producir como máximo 150 armas, 600 unidades de alimento y 1200 municiones. La fábrica 1 produce por cada hora de operación 3 armas, 2 unidades de alimento y 1 munición, mientras que la fábrica 2 produce 1 arma, 3 unidades de alimento y 2 municiones por hora.

El costo de operación por hora es de 10 000 USD para la fábrica 1 y 7 000 USD para la fábrica 2. Para poder sostener el esfuerzo bélico, se requiere producir al menos 20 armas, 200 unidades de alimento y 150 municiones.

Además, por razones logísticas y estratégicas, se ha determinado que la fábrica 2 debe operar al menos el doble de horas que la fábrica 1.

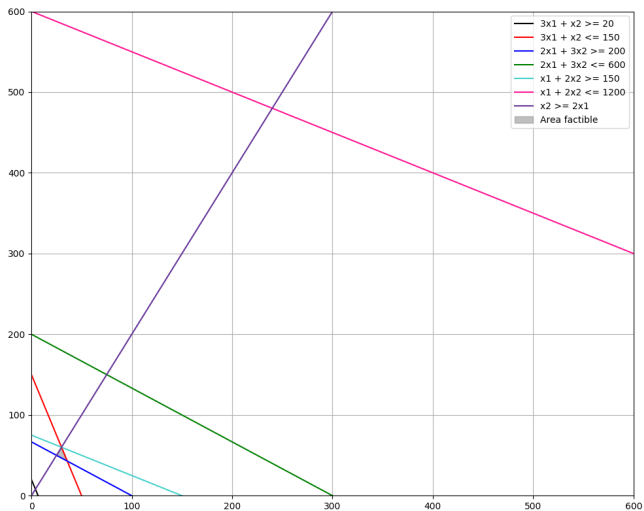
El objetivo es determinar cuántas horas debe operar cada fábrica de forma que se minimice el costo total de producción, cumpliendo todas las restricciones establecidas.

Modelación del problema

$$\text{mín } z = 10000x_1 + 7000x_2$$

$$\text{s.a. } \left\{ \begin{array}{ll} 3x_1 + x_2 \geq 20 & (\text{armas mínimas}) \\ 3x_1 + x_2 \leq 150 & (\text{armas máximas}) \\ 2x_1 + 3x_2 \geq 200 & (\text{alimento mínimo}) \\ 2x_1 + 3x_2 \leq 600 & (\text{alimento máximo}) \\ x_1 + 2x_2 \geq 150 & (\text{municiones mínimas}) \\ x_1 + 2x_2 \leq 1200 & (\text{municiones máximas}) \\ x_2 \geq 2x_1 & (\text{restricción operativa}) \\ x_1, x_2 \geq 0 & (\text{no negatividad}) \end{array} \right.$$

Graficación



Solución

```
model = pyo.ConcreteModel()

model.x1 = pyo.Var(within = pyo.NonNegativeIntegers)
model.x2 = pyo.Var(within = pyo.NonNegativeIntegers)

model.obj = pyo.Objective(expr = 10000 * model.x1 + 7000 * model.x2,
    sense = pyo.minimize)

model.con1 = pyo.Constraint(expr = 3 * model.x1 + model.x2 >= 20)
model.con2 = pyo.Constraint(expr = 3 * model.x1 + model.x2 <= 150)
model.con3 = pyo.Constraint(expr = 2 * model.x1 + 3 * model.x2 >= 200)
model.con4 = pyo.Constraint(expr = 2 * model.x1 + 3 * model.x2 <= 600)
model.con5 = pyo.Constraint(expr = model.x1 + 2 * model.x2 >= 150)
model.con6 = pyo.Constraint(expr = model.x1 + 2 * model.x2 <= 1200)
model.con7 = pyo.Constraint(expr = model.x2 >= 2 * model.x1)

opt = pyo.SolverFactory("glpk")
results = opt.solve(model)
model.display()
```

Resultados

Variables:

x1 : Size=1, Index=None

Key : Lower : Value : Upper : Fixed : Stale : Domain

None : 0 : 0.0 : None : False : False : NonNegativeIntegers

x2 : Size=1, Index=None

Key : Lower : Value : Upper : Fixed : Stale : Domain

None : 0 : 75.0 : None : False : False : NonNegativeIntegers

Objectives:

obj : Size=1, Index=None, Active=True

Key : Active : Value

None : True : 525000.0