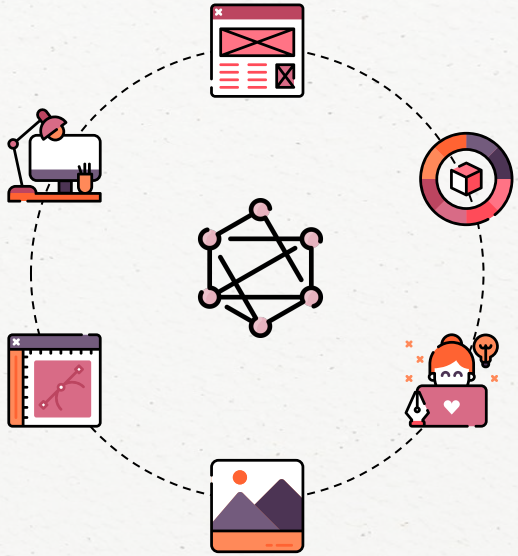
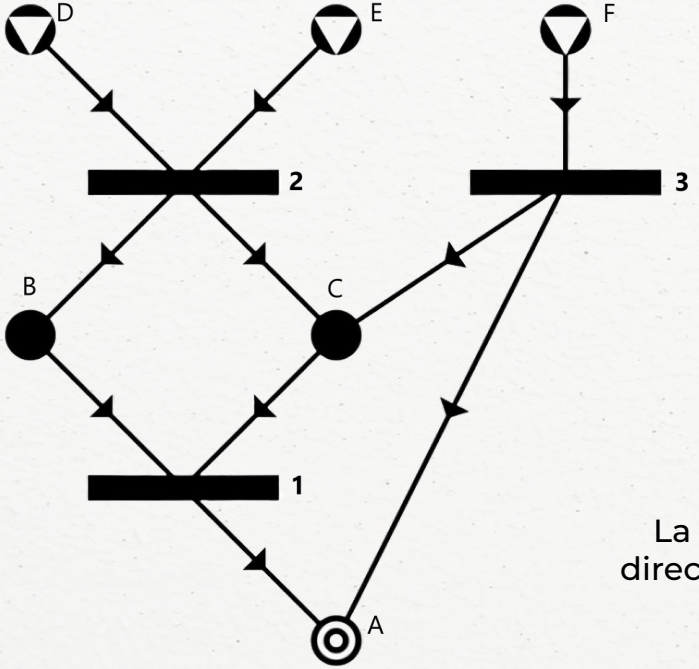
P-Graphs y MILP

Pablo Hernández & Juan Camilo Narváez



P-Graphs



Son representaciones gráficas de un proceso sistemático

Las unidades operativas se representan como rectángulos y los materiales como círculos

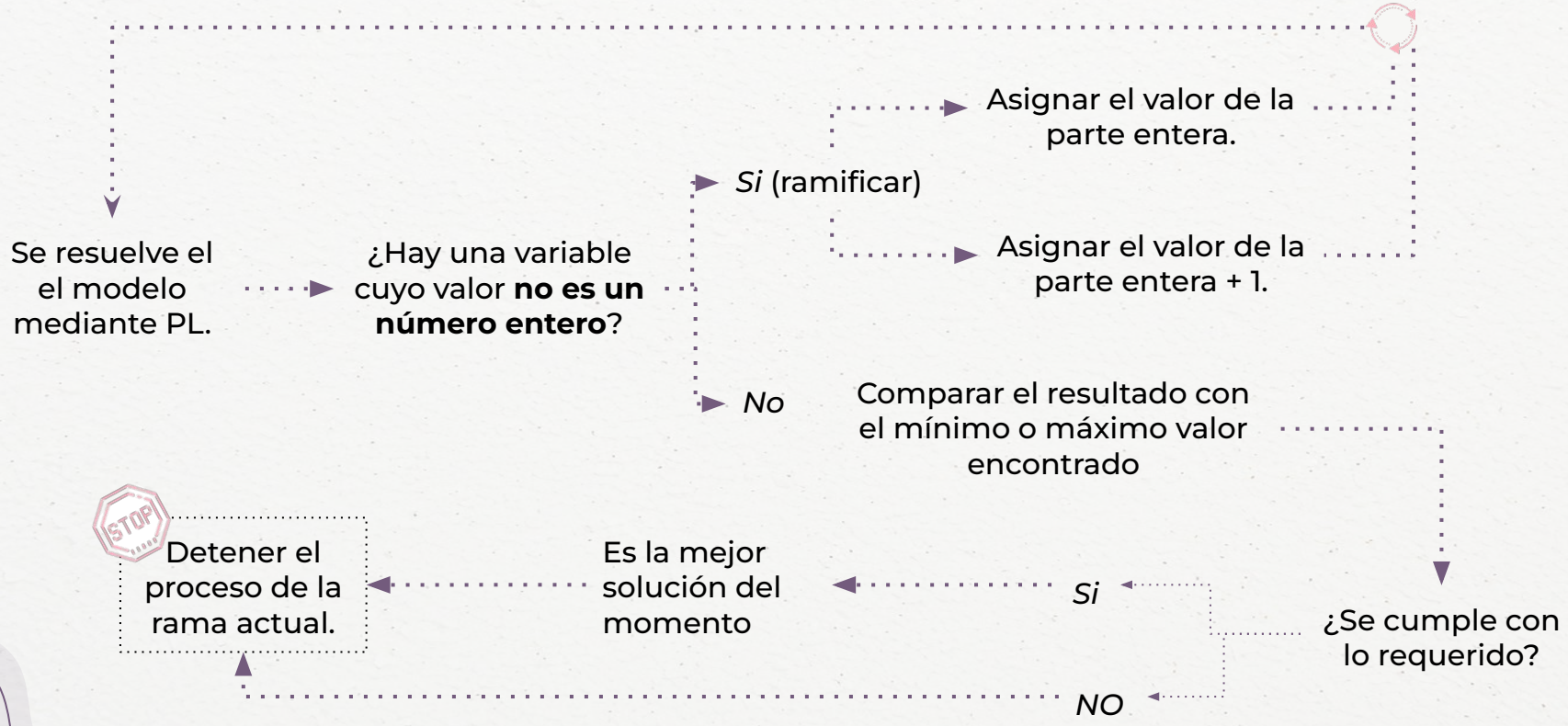
La dirección los arcos es la dirección del **flujo de material** en el sistema.

MILP

La MILP es una extensión de la Programación lineal que incorpora la condición adicional de que algunas o todas las variables deben tomar **valores enteros** (por ejemplo, 0, 1, 2, etc.).

Esta característica la hace ideal para modelar problemas que involucran decisiones discretas, como la asignación de recursos, la planificación de producción o la programación de itinerarios.

Método Branch and Bound



Caso de estudio




P-Graph

Identificar las **variables** y **restricciones** del sistema.

Crear el modelo PL

Asocia las variables y restricciones al modelo.

Branch And Bound

Ejecuta el método de branch and bound.

Visualización y resultados

Graficar el arbol generado a partir del branch and bound.

Entidades del sistema

Unidad operativa

- Nombre
- Costo fijo (Cf)
- Costo proporcional (Cp).
- Material importado.
- Material exportado.

Material

- Tipo
- Limite inferior (Li)

Objetivo y restricciones

Función Objetivo

$$Minimize\,(Cf_1 \times Y_1 + Cp_1 \times X_1 + \ldots + Cf_n \times Y_n + Cp_n \times X_n)$$

Restricciones

- Toda Y es binaria, de modo que $O_u \rightarrow Y_u : 0, 1$
- Todas las variables deben ser mayor o igual que 0

  $X_1, X_2, ..., X_{n-1}, X_n, \geq 0$
- La entrada menos la salida de material debe ser mayor o igual que 0

  $X_{entrada} - X_{salida} \geq 0$
- El flujo de la unidad debe ser menor o igual a su límite superior

  $O_u \rightarrow Y_u : \{0, 1\}$

X representa el f**lujo de material** que pasa a través de una unidad operativa.

Formato de archivo de entrada

Archivo.txt

```
materials:
A: raw_material
B: intermediate
C: intermediate
D: product, flow_rate_lower_bound=10


operating_units:
O1: capacity_upper_bound=1000, fix_cost=4, proportional_cost=4
O2: capacity_upper_bound=1000, fix_cost=2, proportional_cost=1
O3: capacity_upper_bound=1000, fix_cost=3, proportional_cost=2
O4: capacity_upper_bound=1000, fix_cost=3, proportional_cost=2
O5: capacity_upper_bound=1000, fix_cost=2, proportional_cost=4
O6: capacity_upper_bound=1000, fix_cost=2, proportional_cost=1


material_to_operating_unit_flow_rates:
O1: A => B
O2: A => C
O3: A => C
O4: B => D
O5: C => D
O6: C => D
```

Herramientas

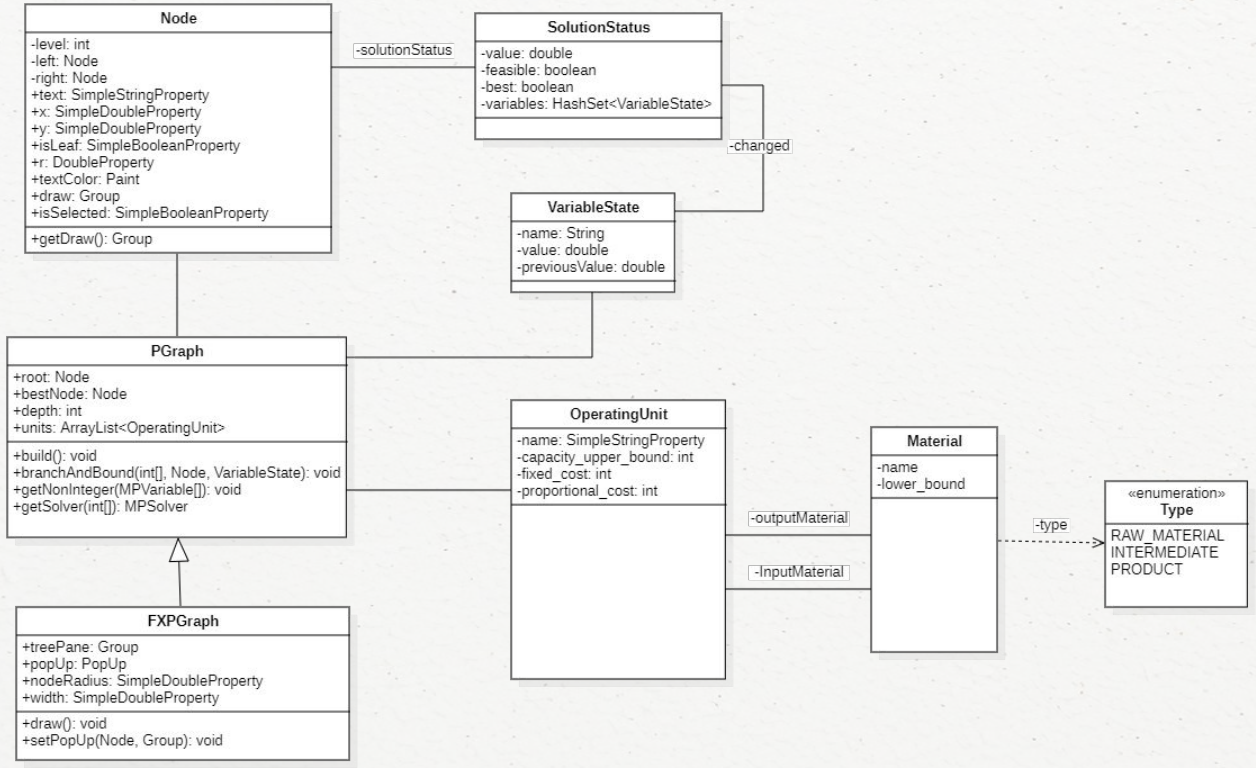
Entorno de desarrollo

- Java 21 (Zulu JDK 21.34.19)
- Gradle
- Netbeans & Intellij Idea
- JavaFx (22.0.1)

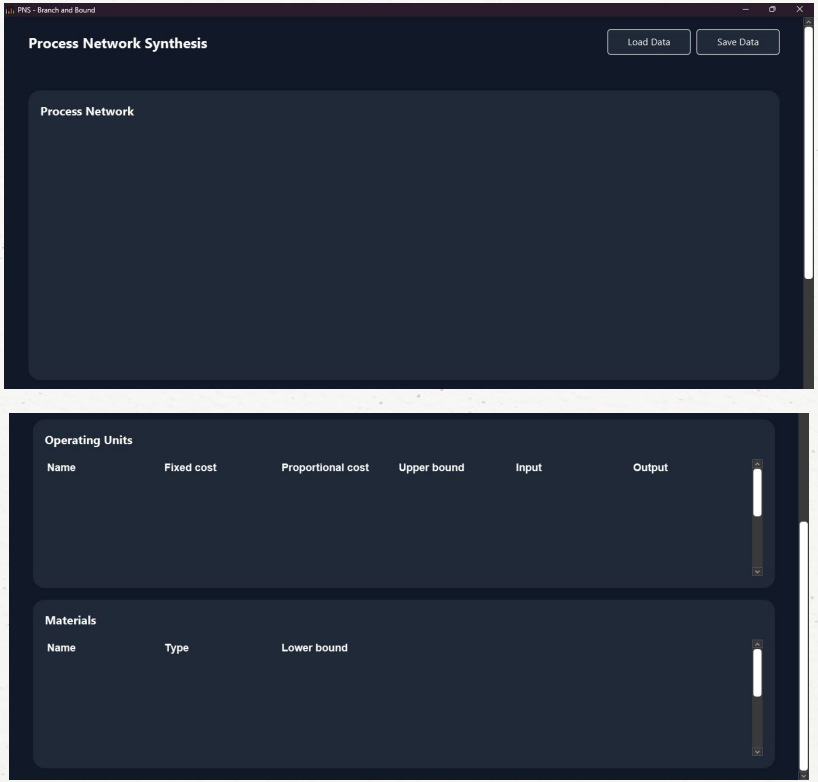
Librería de PL

Google OR-Tools (solucionador de problemas de GLOP).

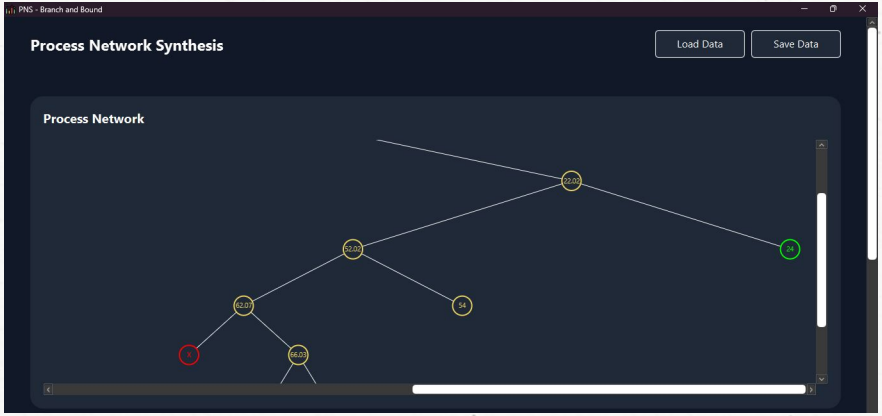
Diagrama de clases(Simplificado**)**

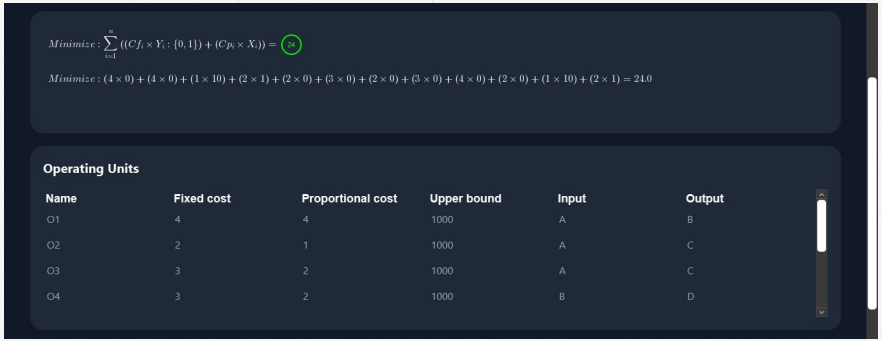


Interfaz (Sin datos**)**

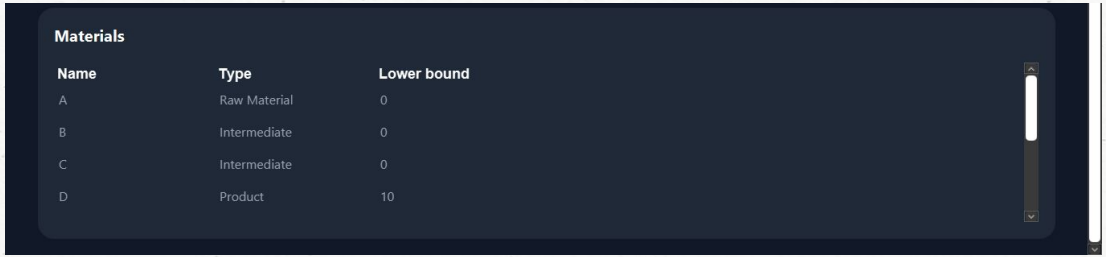


Interfaz

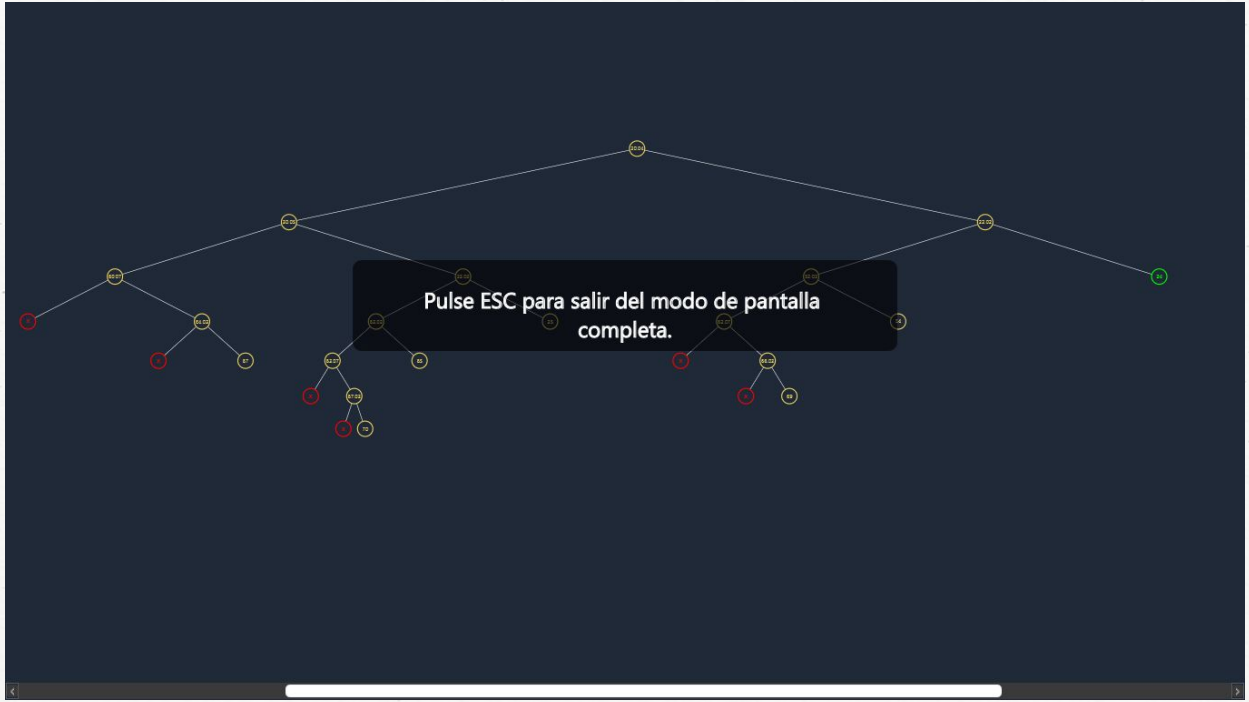

PNS - Branch and Bound
Process Network Synthesis
Load Data
Save Data
Process Network
72.02
62.02
24
62.07
54
66.03
0


$Minimize: \sum_{i=1}^{n} ((Cf_i \times Y_i : \{0,1\}) + (Cp_i \times X_i)) =$ 24
$Minimize: (4 \times 0) + (4 \times 0) + (1 \times 10) + (2 \times 1) + (2 \times 0) + (3 \times 0) + (2 \times 0) + (3 \times 0) + (4 \times 0) + (2 \times 0) + (1 \times 10) + (2 \times 1) = 24.0$
Operating Units
Name
Fixed cost
Proportional cost
Upper bound
Input
Output
O1
4
4
1000
A
B
O2
2
1
1000
A
C
O3
3
2
1000
A
C
O4
3
2
1000
B
D

Interfaz


Materials
Name
Type
Lower bound
A
Raw Material
0
B
Intermediate
0
C
Intermediate
0
D
Product
10

Interfaz(Pantalla completa)



Gracias