



Universidad de  
Oviedo



# Técnicas de Inteligencia Artificial para la Optimización y Programación de Recursos

## Tema 6: Aplicaciones de Scheduling en la vida real

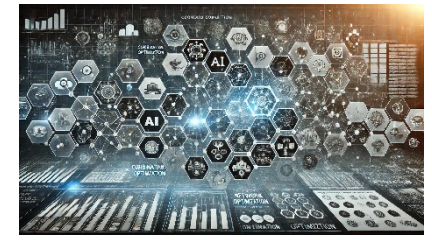
Sezin Afsar  
[afsarsezin@uniovi.es](mailto:afsarsezin@uniovi.es)

Ciencia de la Computación e Inteligencia Artificial  
Departamento de Informática

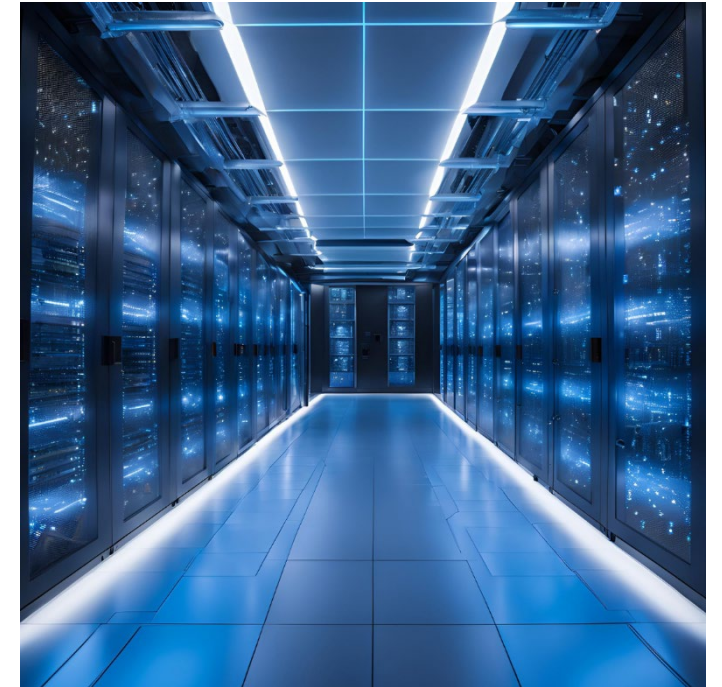




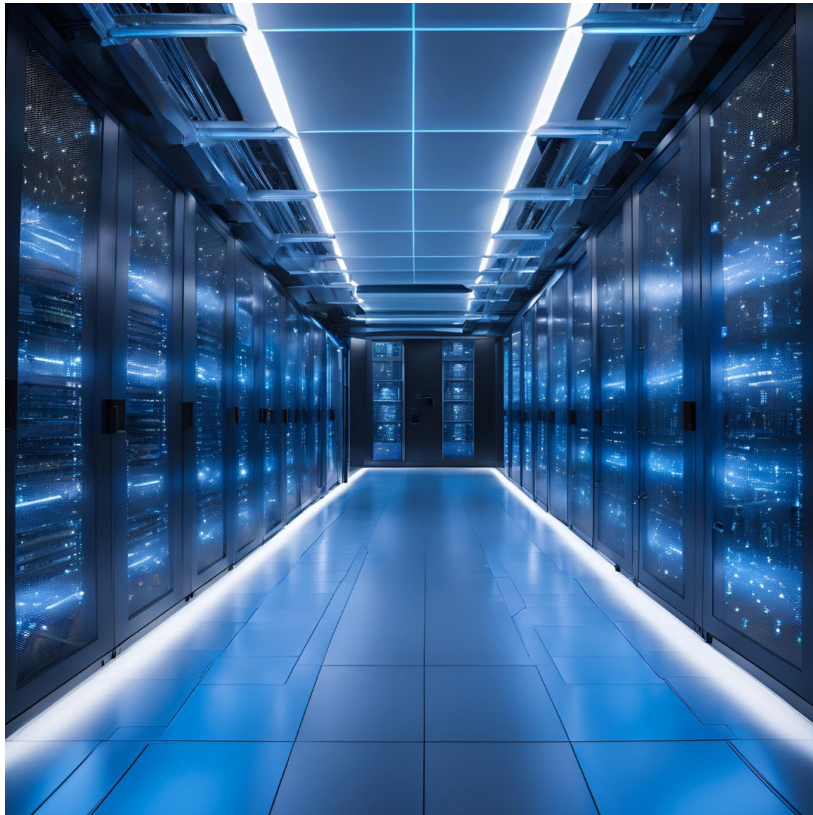
# Scheduling “Verde”: ¿Por qué?



Cuando hablamos de **la computación en la nube** o de **la inteligencia artificial**,  
¿qué imaginamos?



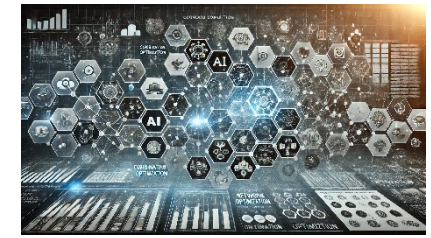
# Scheduling “Verde”: ¿Por qué?



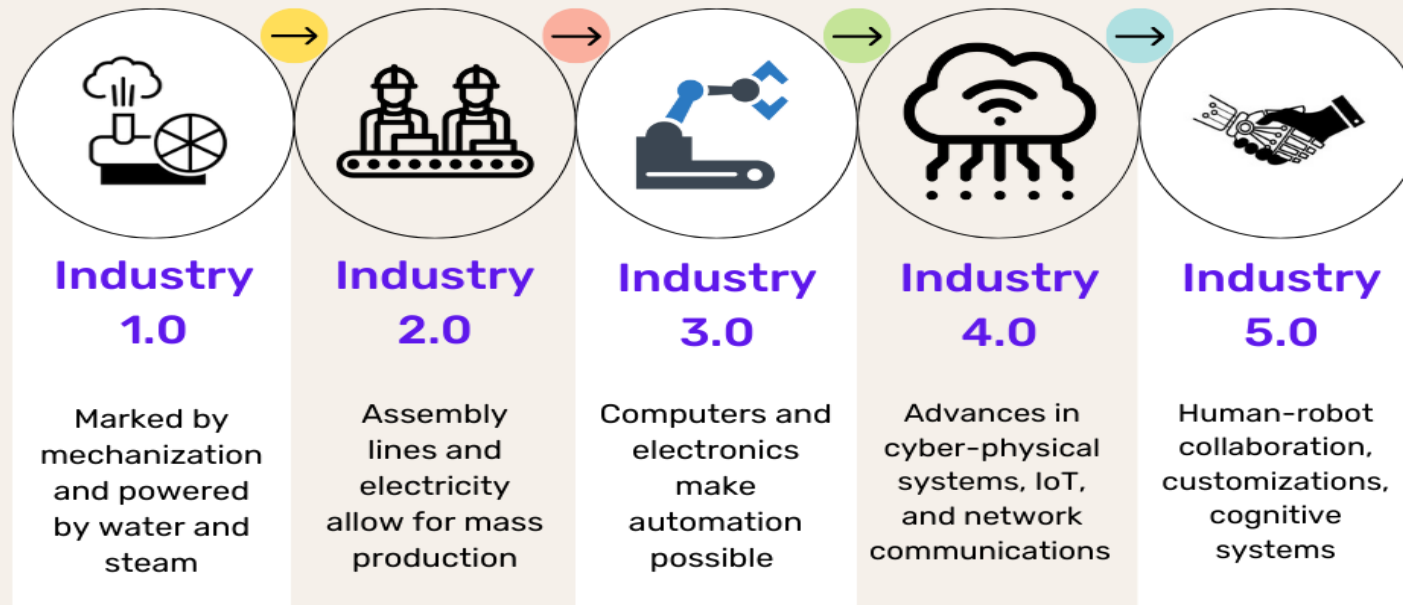
- Procesar una consulta en ChatGPT consume en promedio 10 veces **más energía** que una búsqueda estándar en Google. [Goldman Sachs, 2024]
- En España:
  - La demanda de energía de la IA y los centros de datos crecerá hasta casi 26 teravatio hora (TWh) en 2050 (**más del 300%** que en la actualidad)
  - Lo que representará el **44% de la demanda** de energía de los electrodomésticos comerciales y el **15% de la demanda total** de electricidad de los edificios.



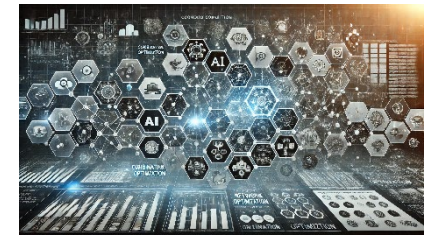
# Scheduling “Verde”: ¿Por qué?



## Industrial Revolutions



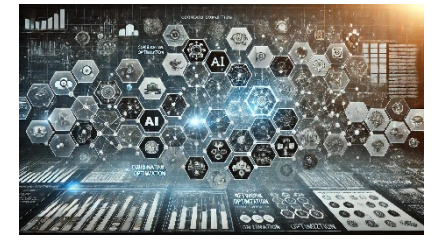
# Scheduling “Verde”: ¿Por qué?



Según la definición que desarrolla la propia Comisión Europea en su informe, **la industria 5.0** es un nuevo paradigma que se centra en tres elementos clave:

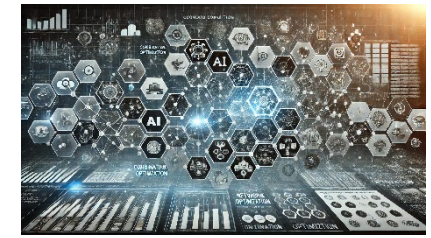
- El ser humano
- La resiliencia
- **La sostenibilidad medioambiental**

# Scheduling “Verde”: ¿Por qué?

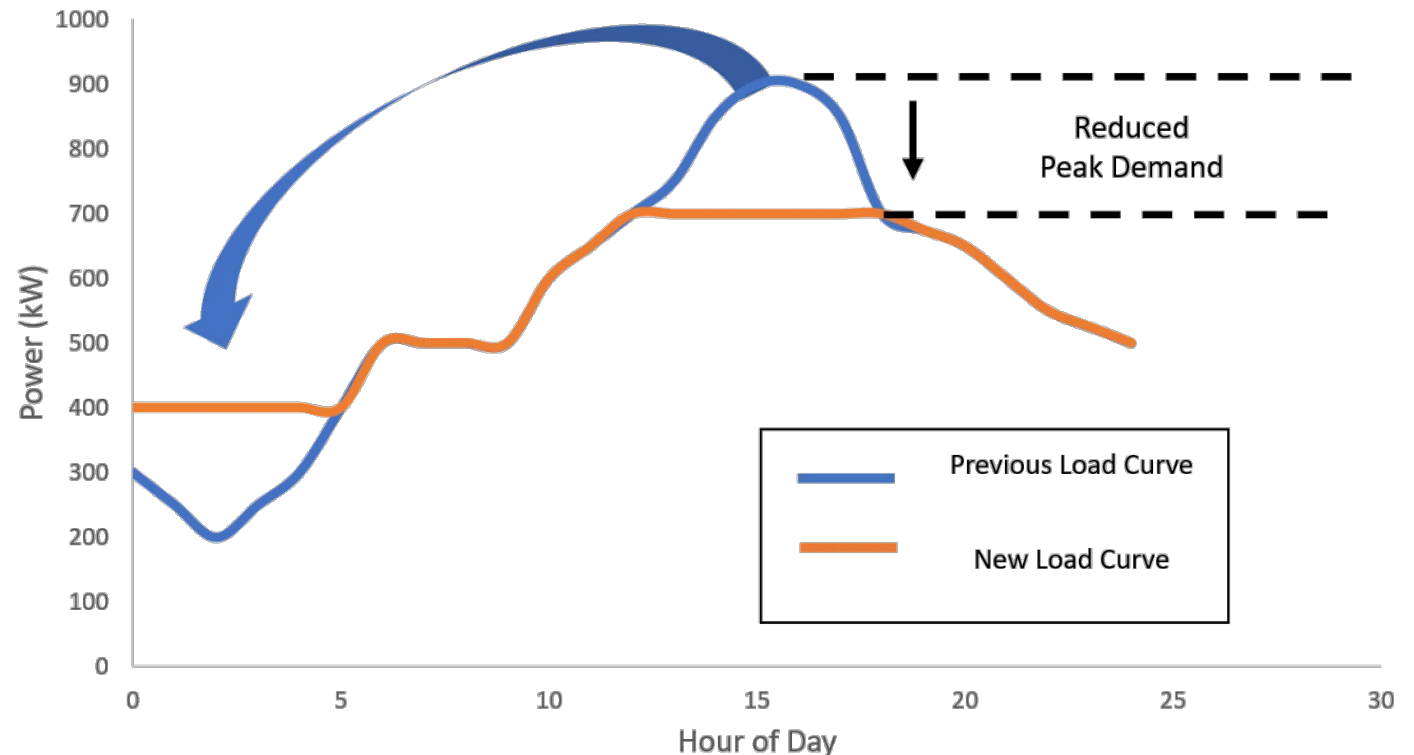


- La sostenibilidad medioambiental es un factor clave debido a **la crisis climática**
- Una mayor demanda de energía significa consumir **más** combustibles fósiles, **más** emisiones de gases de efecto invernadero, lo que **dificulta** la lucha contra la crisis climática.

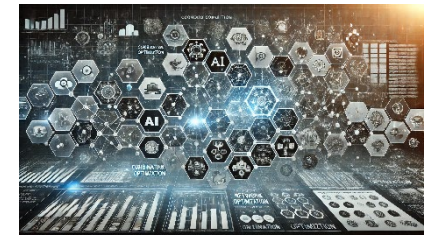
# Scheduling “Verde”: ¿Qué se puede hacer?



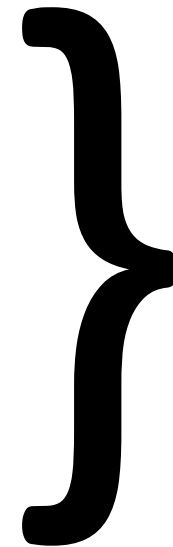
- Minimizar el consumo total de energía
- Desplazar el consumo a horas de menor demanda cuando sea posible



# Scheduling “Verde”: ¿Qué se puede hacer?



- Minimizar el consumo total de energía
- Desplazar el consumo a horas de menor demanda cuando sea posible

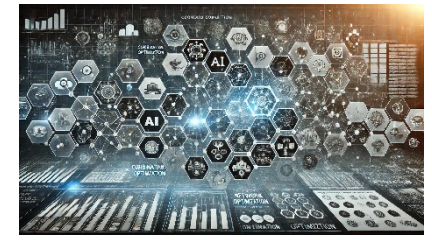


Minimizar  
el **coste** total  
de energía

$$\min \sum_h \text{precio}_h * \text{carga}_h$$



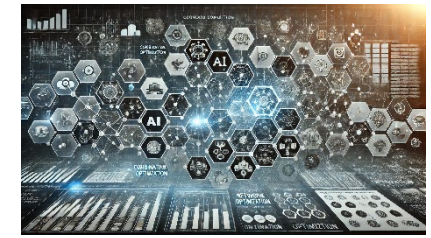
# Scheduling “Verde”: ¿Qué se puede hacer?



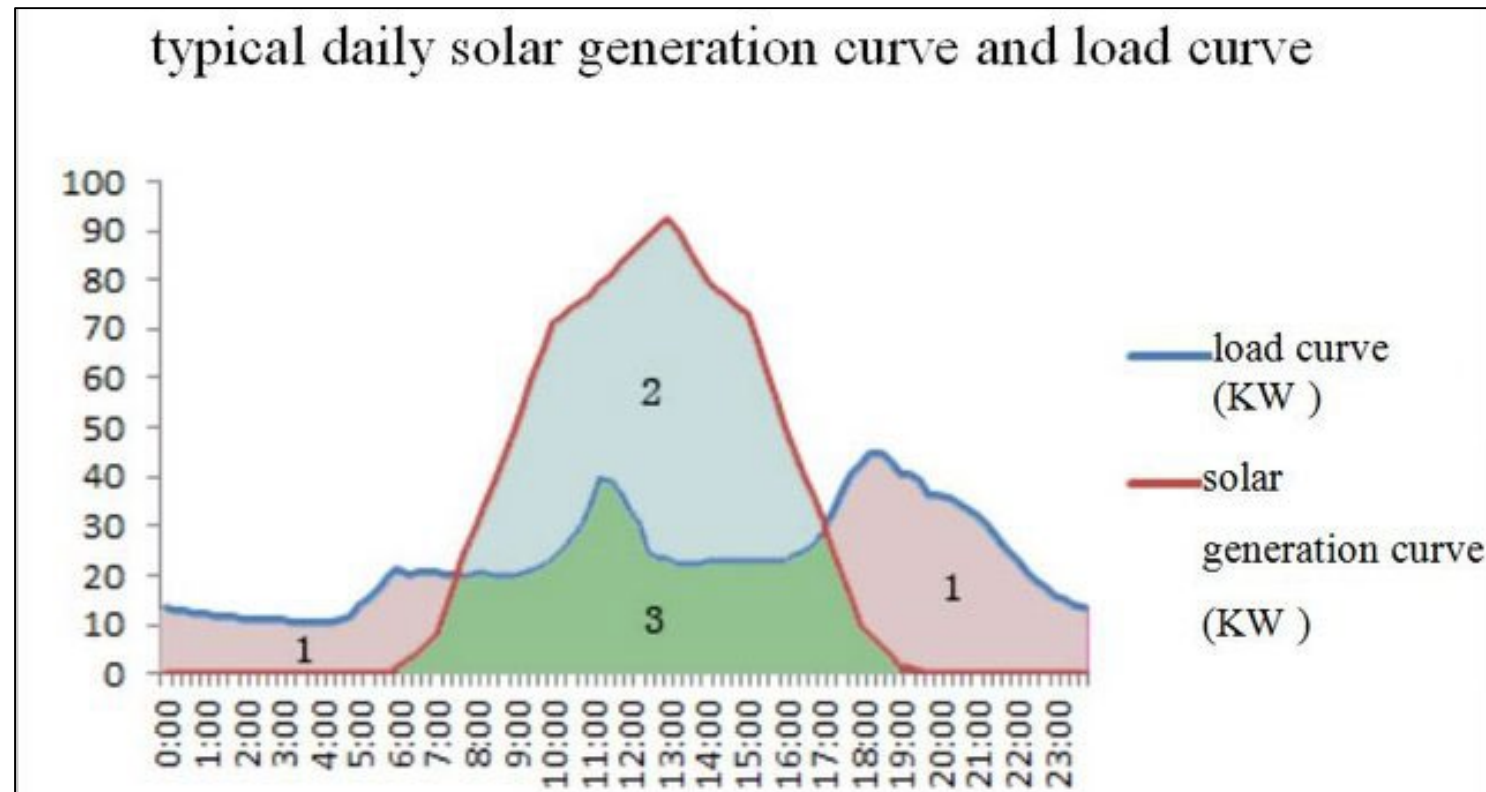
El mismo modelo se puede utilizar para escenarios distintos:

- Para **explotar** la capacidad existente de generación de energía renovable
- Para **adaptarse** a los precios del mercado spot

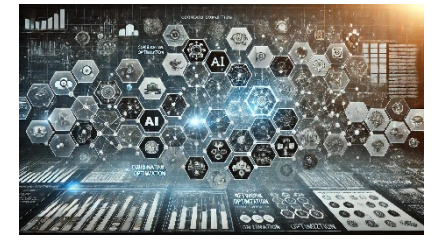
# Scheduling “Verde”: ¿Qué se puede hacer?



- Para explotar la capacidad existente de generación de energía renovable



# Scheduling “Verde”: ¿Qué se puede hacer?



El mismo modelo se puede utilizar para escenarios distintos:

- Para explotar la capacidad existente de generación de energía renovable
- Para adaptarse a los precios del mercado spot



# Scheduling “Verde”: ¿Qué se puede hacer?



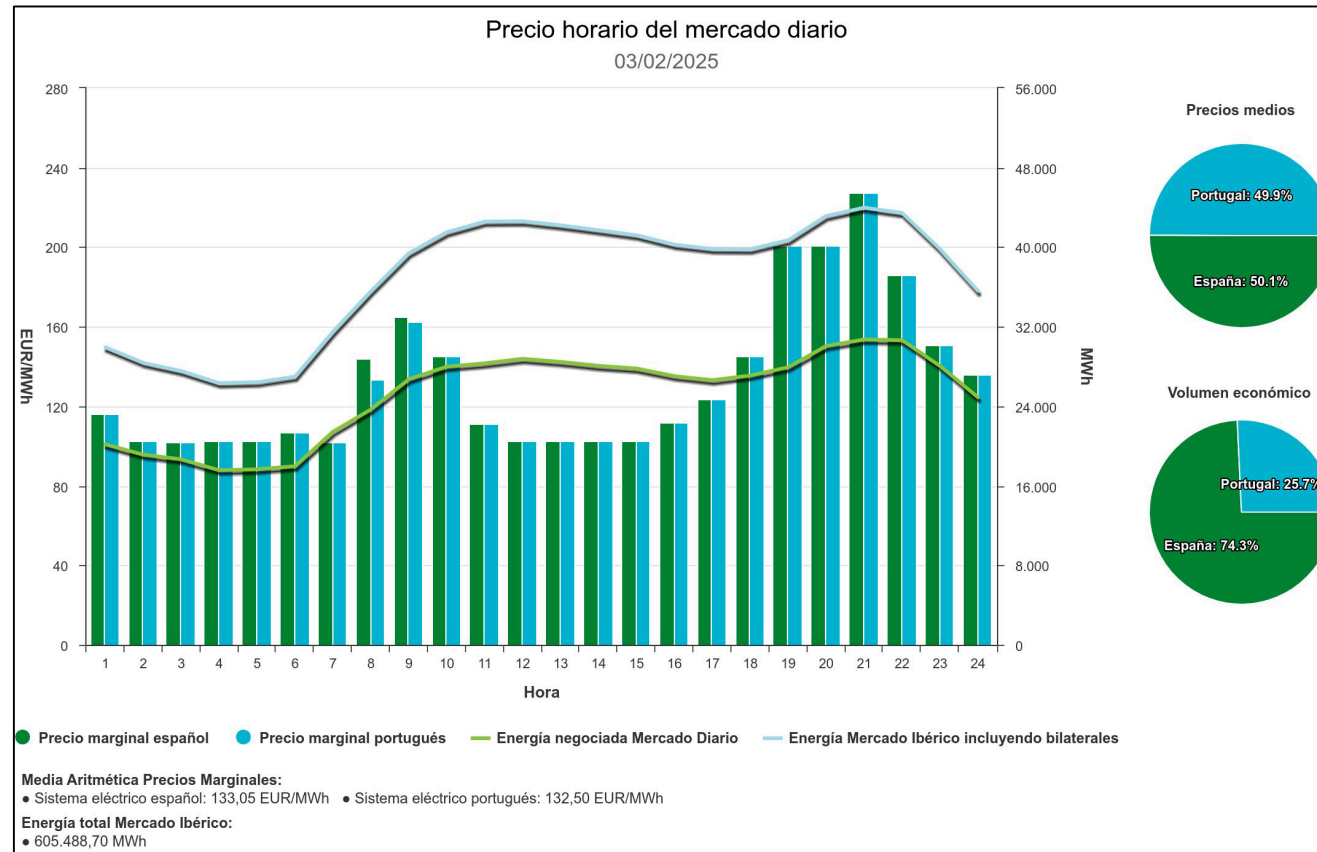
El mercado spot diario (Day-ahead market)

- Los generadores presentan ofertas (para vender)
- Los compradores estiman demandas (para comprar) la electricidad que se entregará al día siguiente
- Los generadores con precios más bajos se despachan primero
- El precio final se establece donde la curva de oferta y demanda se cruzan (precio marginal).
- El mercado "cleara" (cierra) cuando la oferta y la demanda coinciden a un precio determinado

# Scheduling “Verde”: ¿Qué se puede hacer?



- Para adaptarse a los precios del mercado spot



[omie.es](http://omie.es)

# Scheduling “Verde”: ¿Qué se puede hacer?



El mismo día en Francia:



[epexspot.com](http://epexspot.com)

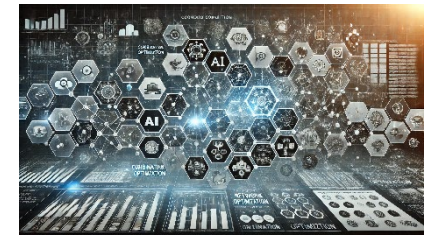


# Scheduling “Verde”: ¿Como se puede modelar?



- Minimizar el coste energético  $\longrightarrow \min \sum_{h \in \text{horas}} \text{precio}_h * \text{carga}_h$
- Optimizar el objetivo operacional (min makespan, min tardiness...)  $\longrightarrow \min \sum_{j \in \text{jobs}} \text{tardiness}_j$

# Scheduling “Verde”: ¿Como se puede modelar?



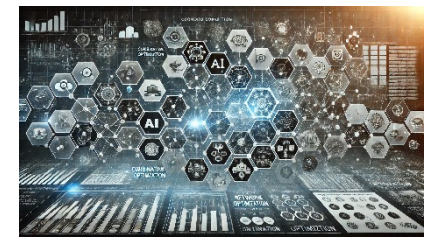
Minimizar el costo energético  $\longrightarrow \min \sum_{h \in \text{horas}} \text{precio}_h * \text{carga}_h$

Optimizar el objetivo operacional  
(min makespan, min tardiness...)  $\longrightarrow \min \sum_{j \in \text{jobs}} \text{tardiness}_j$

sujeto a

## Las restricciones del problema de scheduling

# Scheduling “Verde”: ¿Como se puede modelar?



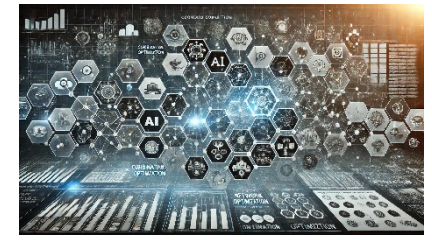
## Problemas de scheduling

- Hay un conjunto de trabajos y un conjunto de máquinas
- Trabajos consisten en tareas

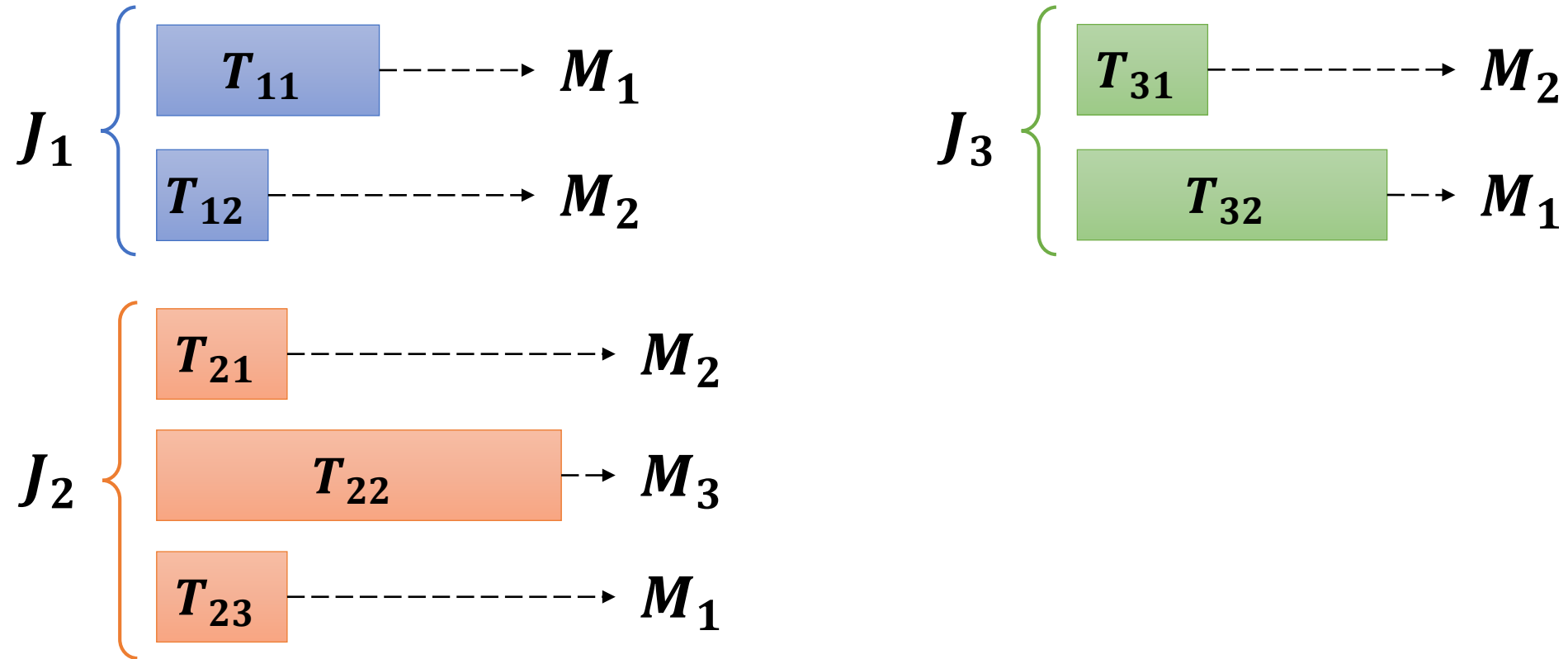
Tipo	Asig. de máq-tareas prefijada	Orden de tareas prefijado	Mismo orden de tareas	Todas máquinas
Flow shop	X	X	X	X
Open shop	X			X
Job shop clásico	X	X		
Job shop flexible		X		



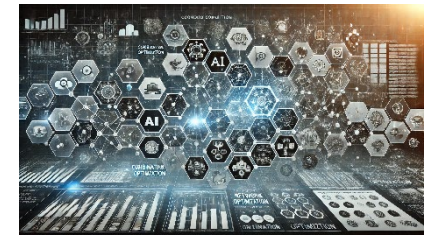
# Scheduling “Verde”: ¿Como se puede modelar?



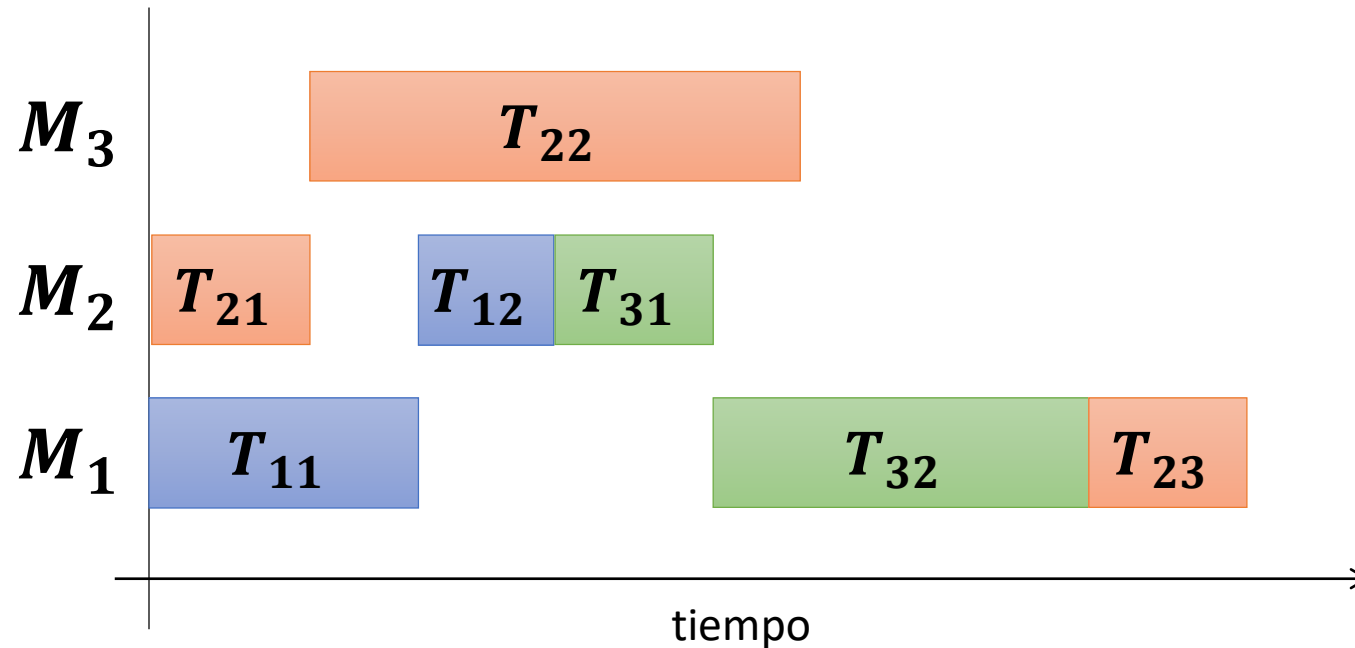
## Job Shop Clásico



# Scheduling “Verde”: ¿Como se puede modelar?



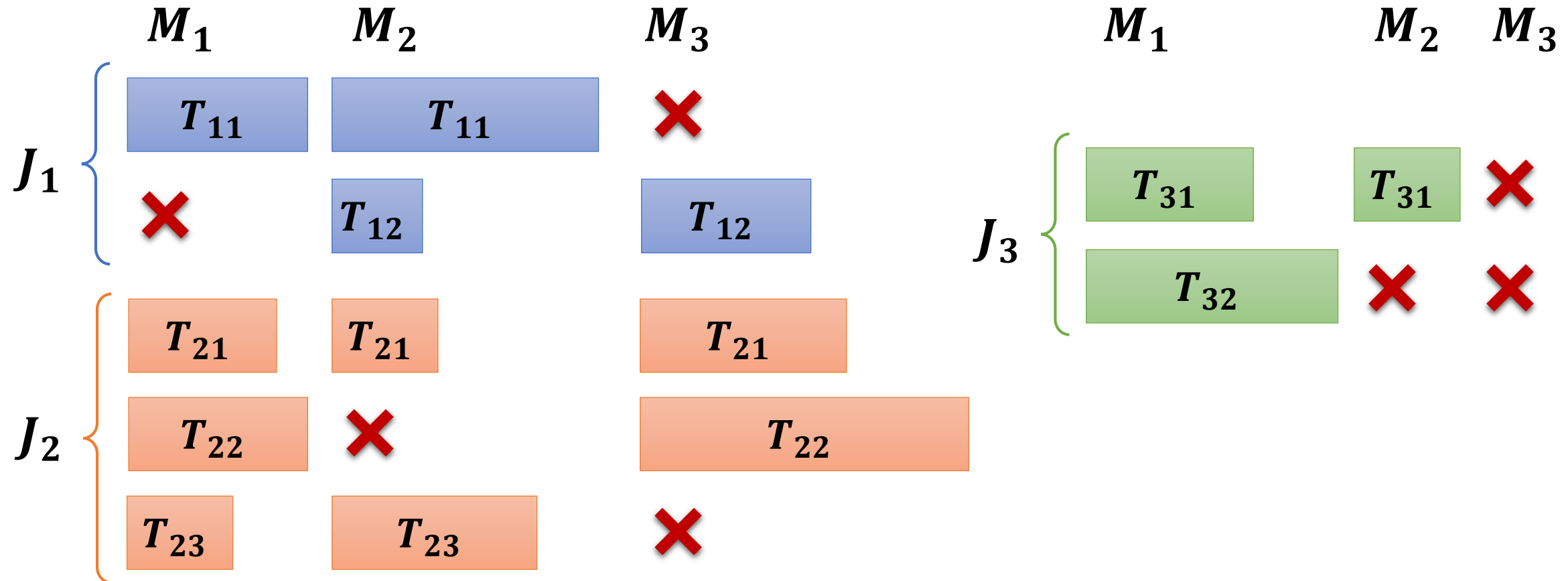
## Job Shop Clásico



# Scheduling “Verde”: ¿Como se puede modelar?

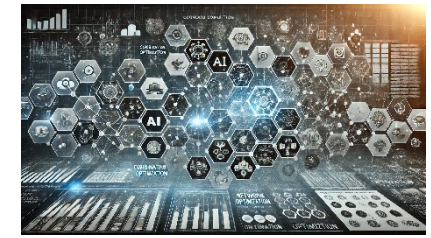


## Job Shop Flexible



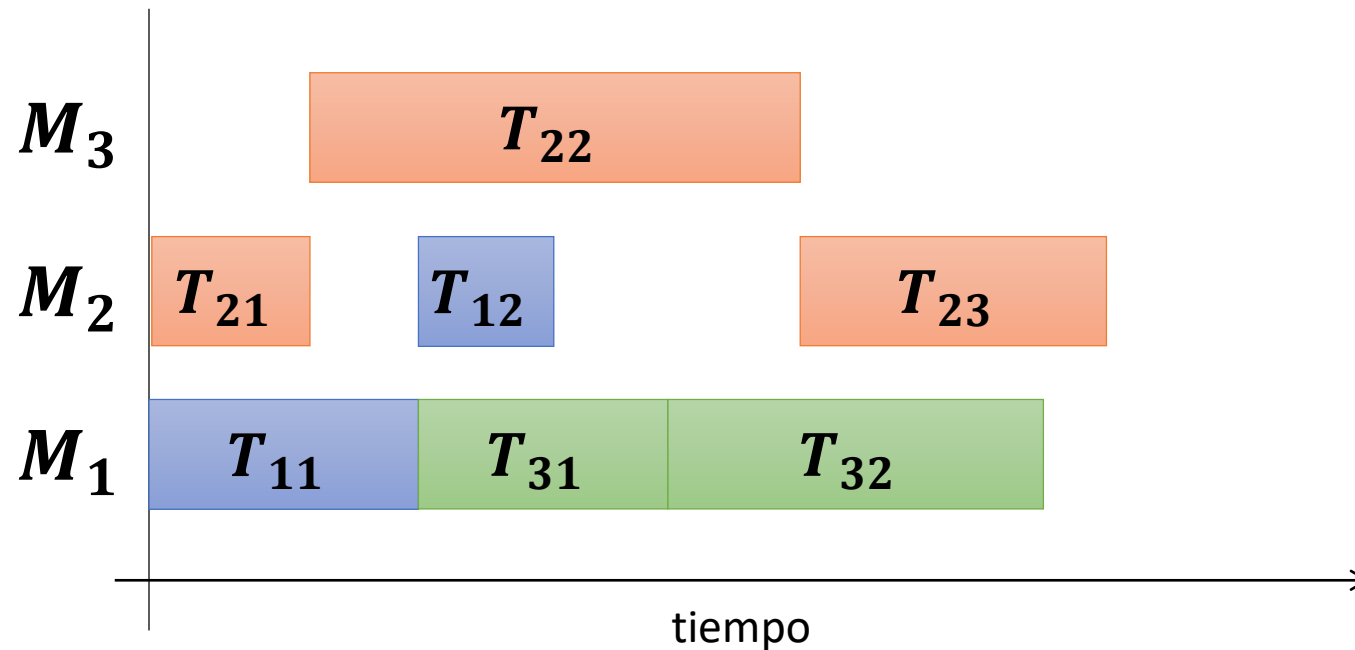


# Scheduling “Verde”: ¿Como se puede modelar?



## Job Shop Flexible

1. Decidir a qué máquina asignar cada tarea
2. Hacer la planificación



# Scheduling “Verde”: ¿Como se puede modelar?



Minimizar {el tardiness total, el coste energético}

objetivos

Asignar las tareas a las maquinas  
Asignar las horas de inicio de todas las tareas

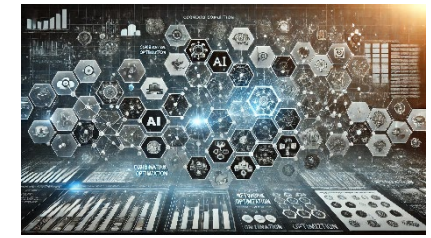
variables de  
decisión

Sujeto a

El orden de las tareas de los trabajos prefijado  
Sin solapamiento en las máquinas  
Sin interrupción de tareas

restricciones

# Scheduling “Verde”: ¿Como se puede modelar?



El método más común es modelarlo como un **MILP**

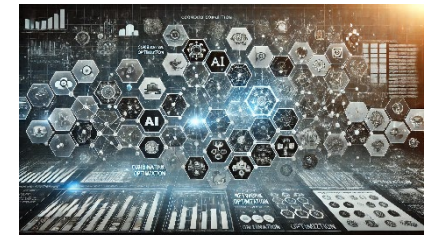
## Conjuntos:

$J$	conjunto de los trabajos
$O_j$	conjunto de las operaciones/tareas del trabajo $j$
$M$	conjunto de las máquinas
$M_{(j,o)}$	conjunto de las máquinas que puede procesar la operación $o$ del trabajo $j$
$H$	conjunto de franjas horarias

## Parametros:

$q_{m,h}^{(j,o)}$	el coste energético de la operación $o$ del trabajo $j$ si empieza en la máquina $m$ a las $h$
$d_j$	la fecha de entrega del trabajo $j$
$p_m^{(j,o)}$	el tiempo de procesamiento de la operación $o$ del trabajo $j$ en la máquina $m$
$n_j$	la última tarea del trabajo $j$

# Scheduling “Verde”: ¿Como se puede modelar?



El método más común es modelarlo como un MILP

Variables:

$TT$	el tardiness total
$CE$	el coste energético total
$X_{m,h}^{(j,o)}$	variable binaria, es igual a 1 si la operación $o$ del trabajo $j$ empieza en la máquina $m$ a las $h$
$C_j$	el tiempo de finalización del trabajo $j$
$T_j$	el tardiness del trabajo $j$



# Scheduling “Verde”: ¿Como se puede modelar?



$$\min\{TT, CE\}$$

s.t.

$$CE = \sum_{j \in J} \sum_{o \in O_j} \sum_{m \in M_{(j,o)}} \sum_{h \in H} q_{m,h}^{(j,o)} X_{m,h}^{(j,o)} \quad (1)$$

$$TT = \sum_{j \in J} T_j \quad (2)$$

$$T_j \geq C_j - d_j \quad \forall j \in J \quad (3)$$

$$C_j \geq (h + p_m^{(j,n_j)}) \times X_{m,h}^{(j,n_j)} \quad \forall h \in H, j \in J \quad (4)$$

$$\sum_{m \in M_{(j,o)}} \sum_{h \in H} X_{m,h}^{(j,o)} = 1 \quad \forall j \in J, o \in O_j \quad (5)$$

# Scheduling “Verde”: ¿Como se puede modelar?



$$\sum_{m \in M_{(j,o)}} \sum_{h \in H} X_{m,h}^{(j,o)} (h + p_m^{(j,o)}) \leq \sum_{h \in H} X_{m,h}^{(j,o+1)} h \quad \forall j \in J, o \in O_j \setminus \{n_j\} \quad (6)$$

$$\sum_{j \in J} \sum_{o \in O_j} \sum_{\tau = h - p_m^{(j,o)} + 1} X_{m,\tau}^{(j,o)} \leq 1 \quad \forall m \in M, h \in H \quad (7)$$

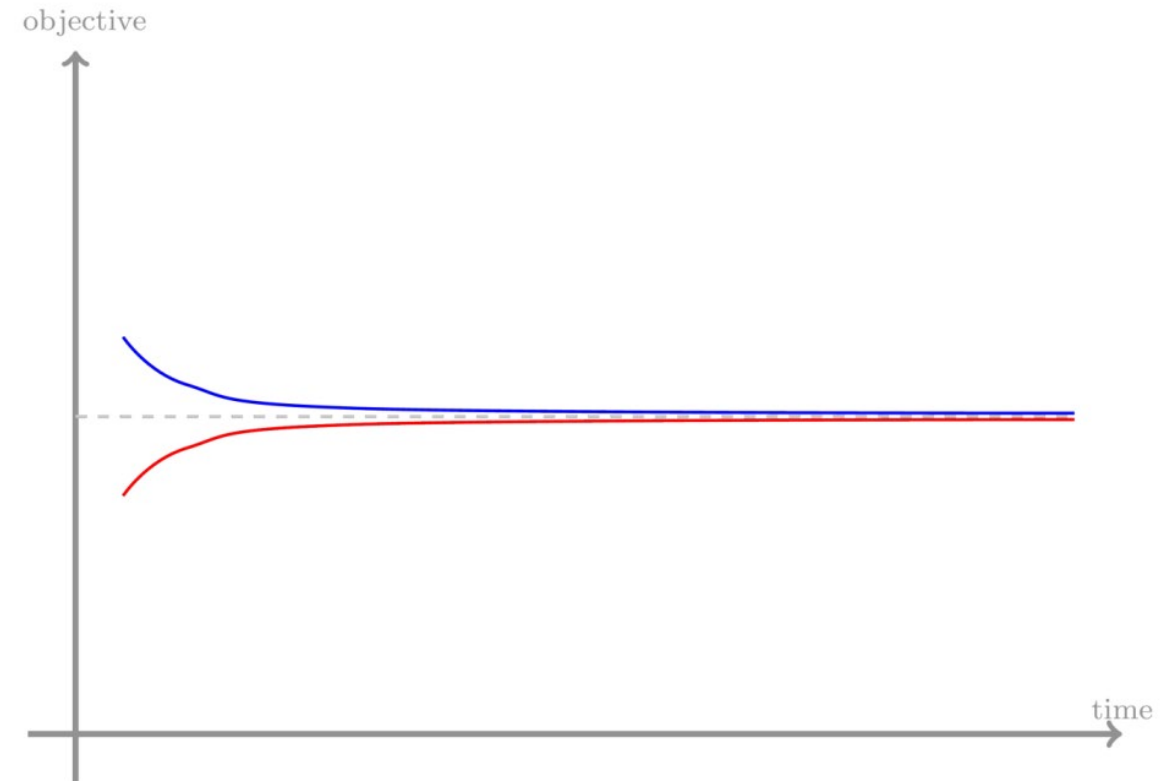
$$X_{m,h}^{(j,o)} \in \{0, 1\} \quad \forall j \in J, o \in O_j, m \in M_{(j,o)}, h \in H \quad (8)$$

$$T_j, C_j \geq 0 \quad \forall j \in J \quad (9)$$

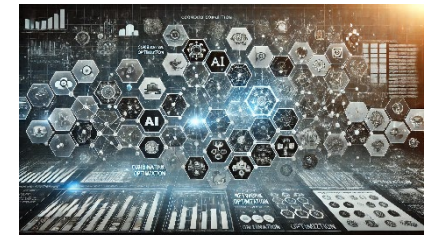
# Scheduling “Verde”: ¿Como se puede resolver?



- Se puede resolver los modelos de MILP con solvers comerciales (CPLEX, Gurobi) o solvers de tipo open-source (SCIP, JuMP)
- El efecto “tail-off” suele ocurrir y aumenta el tiempo de ejecución
- No logran encontrar una solución factible para instancias más grandes



# Scheduling “Verde”: ¿Como se puede resolver?



## IBM ILOG CP Optimizer



- Forma parte del CPLEX Optimization Studio
- Inicialmente desarrollado para resolver problemas de scheduling
- Facilita la construcción de modelos y abordar la resolución de problemas de scheduling y enrutamiento de vehículos
- Es un solver **muy** potente



# Scheduling “Verde”: ¿Como se puede resolver?



## IBM ILOG CP Optimizer

- Tiene su propio “lenguaje” de modelización
- La curva de aprendizaje suele ser pronunciada

```
from docplex.cp.model import *
```

# Scheduling “Verde”: ¿Como se puede resolver?



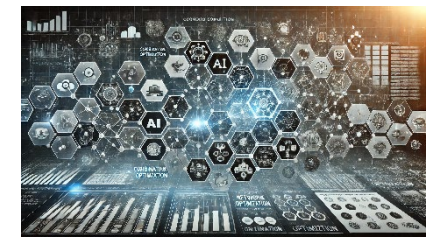
- CP Optimizer introduce 2 nuevos tipos de variables: *intervalos* y *secuencias*
- **Variable de intervalo**
  - Para modelizar las tareas
  - Definida por su duración

```
CPmodel = CpoModel()  
task1 = CPmodel.interval_var(name='task1', size=10)
```

- **Variable de secuencia:**
  - Para modelizar las máquinas
  - Es un conjunto de variables de intervalo sin orden

```
list_tasks = [task1, task2]  
machine = CPmodel.sequence_var(list_tasks)
```

# Scheduling “Verde”: ¿Como se puede resolver?



Job shop clásico (JSP)  
en CP Optimizer:

$$\min \sum_{j \in Jobs} Tardiness_j$$

s.t.

$$Tardiness_j = \max\{0, completud_j - due_date_j\} \quad \forall j \in Jobs$$

$$noOverlap(máquina_m)$$

$$\forall m \in Máquinas$$

$$endBeforeStart(tarea_{jk}, tarea_{j,k+1})$$

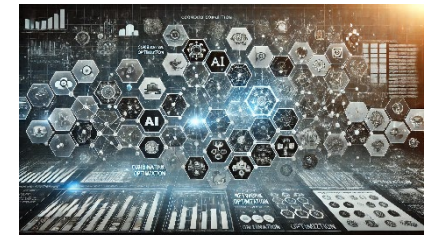
$$\forall j \in Jobs, k \in Tareas_j$$

```
total_tardiness = CPmodel.integer_var(name = 'Total Tardiness')
CPmodel.add(total_tardiness ==
             sum(max(0, completud[j] - due_dates[j]) for j in Jobs))
```

```
CPmodel.add(CPmodel.no_overlap(machine[m]) for m in Maquinas)
```

```
CPmodel.add(end_before_start(tasks[j,k], tasks[j,k+1])
             for j in Jobs, k in Tareas[j])
```

# Scheduling “Verde”: ¿Como se puede resolver?



- JSP flexible verde trae nuevos retos:  
**flexibilidad**

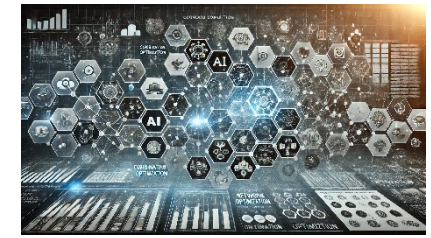
- CP Optimizer tiene variables de intervalos **opcionales** para modelizarlo

```
task1_alt1 = interval_var(size=15, optional=True)
task1_alt2 = interval_var(size=12, optional=True)
task1_alts = [task1_alt1, task1_alt2]
CPmodel.add(alternative(task1, [task1_alts[a]
                             for a in task1_alts]))
```

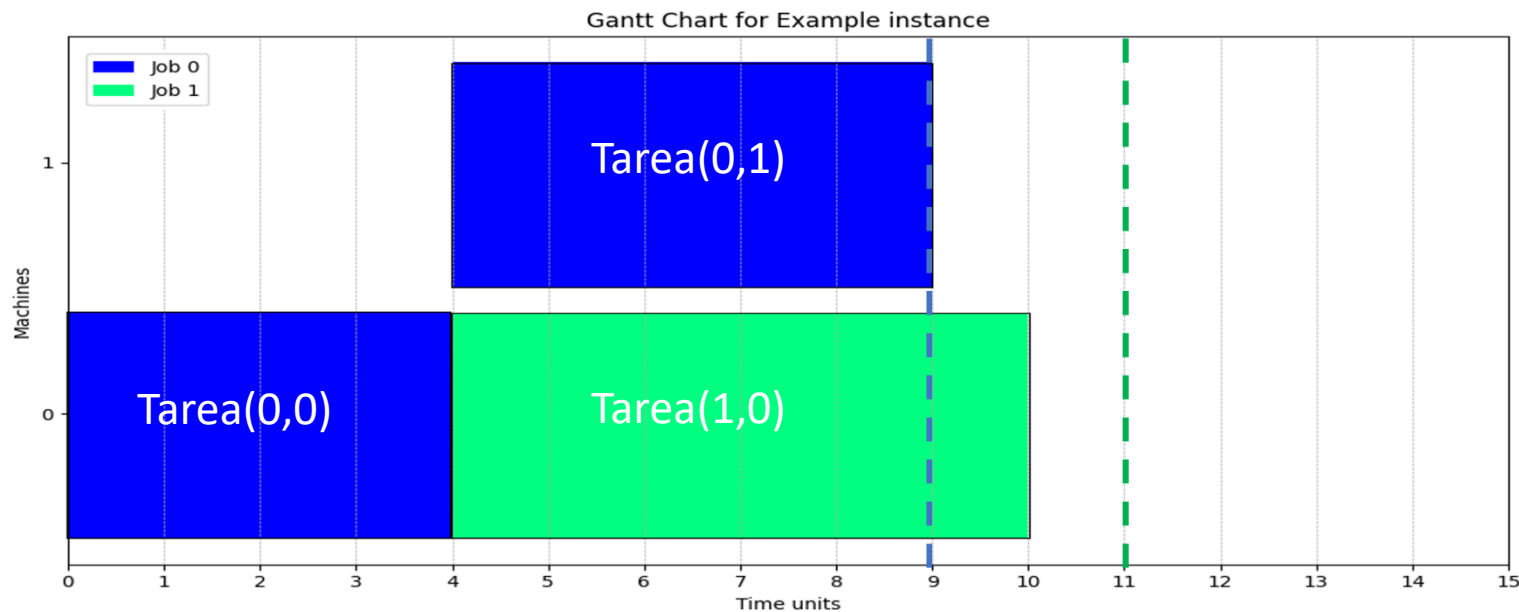
- Hay más de una máquina que puede procesar la misma tarea
  - Por ej. puede haber una **más rápida** que las demás
  - Pero consumiendo **más energía**
  - Si esta máquina procesa todas las tareas, ya no será tan rápido.



# Scheduling “Verde”: ¿Como se puede resolver?



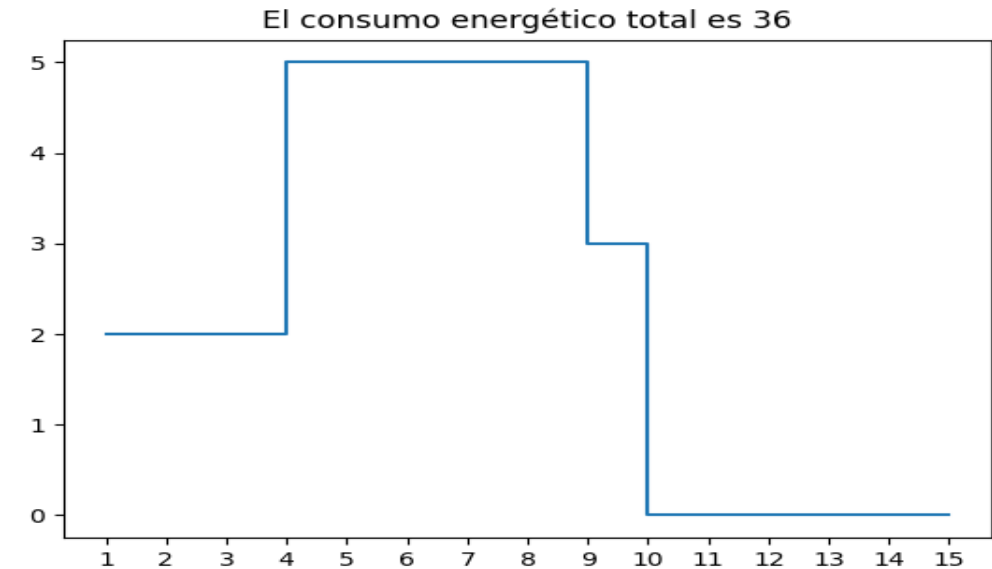
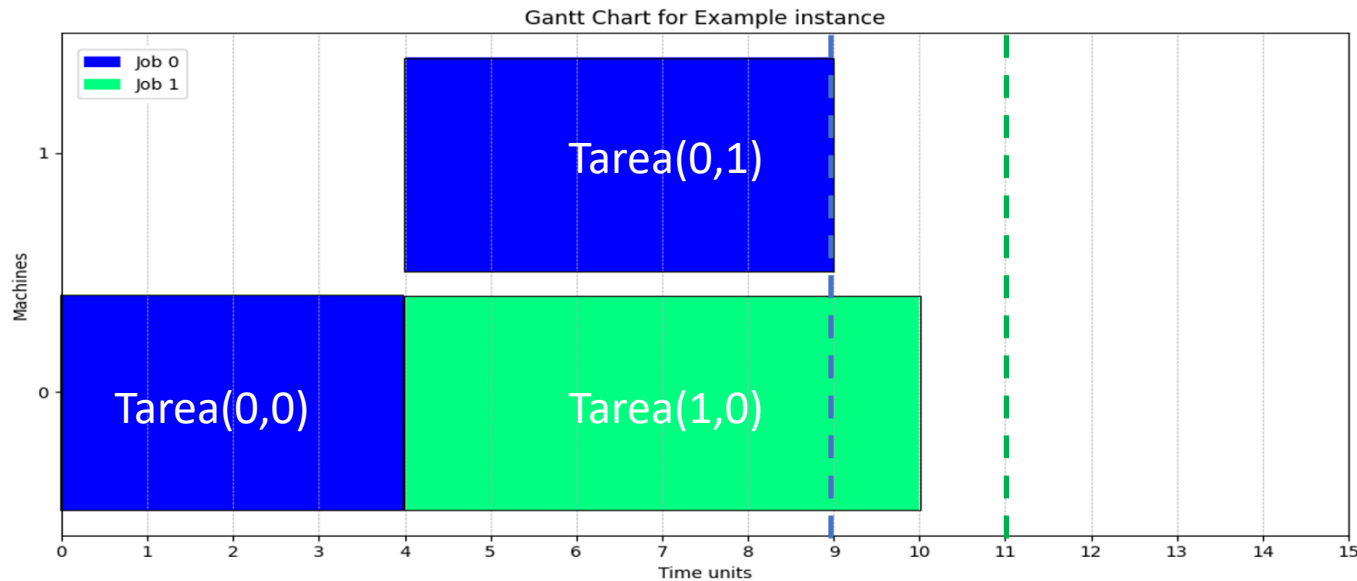
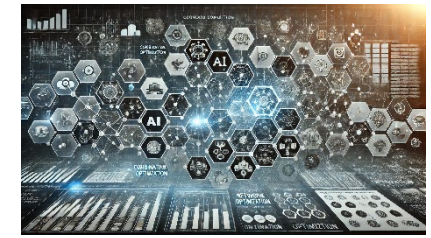
Por ejemplo, si tenemos 2 máquinas y 2 trabajos con 2 tareas cada uno



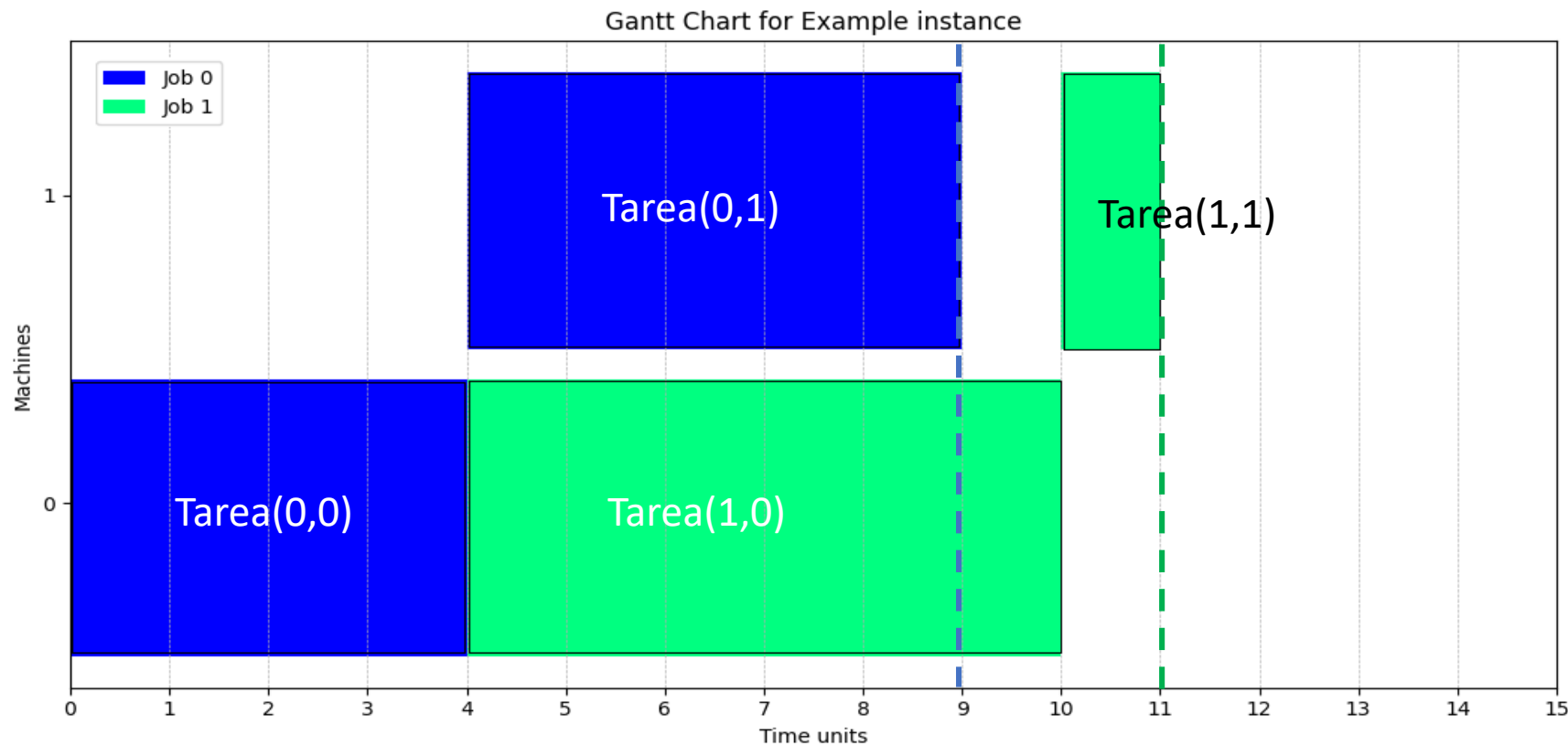
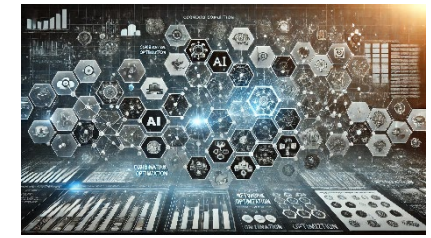
Imaginemos que, además, solo una tarea, Tarea(1,1), se puede ejecutar en las dos máquinas:

- Duración en la M0 es 4h
- Duración en la M1 es 1h

# Scheduling “Verde”: ¿Cuál es la solución óptima?

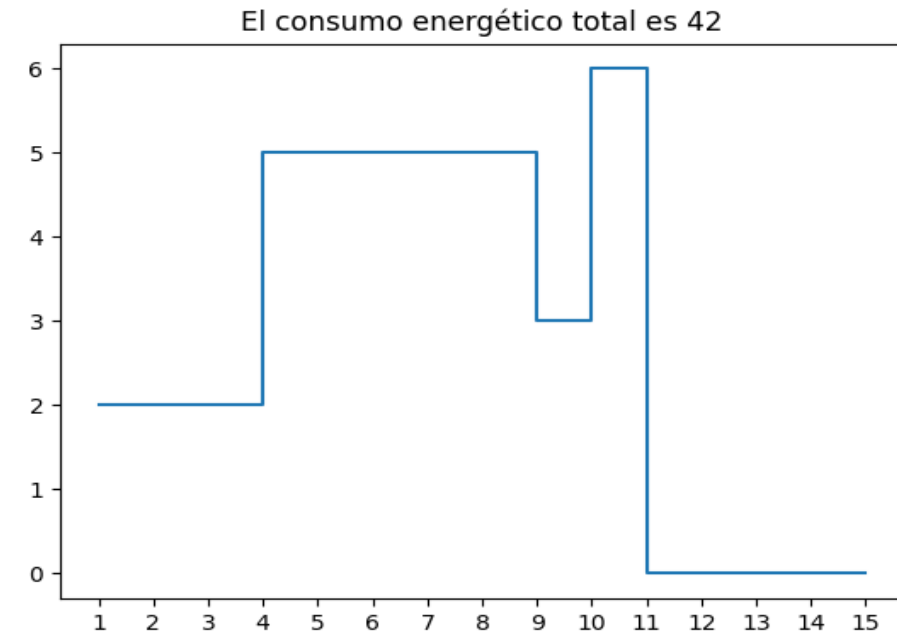
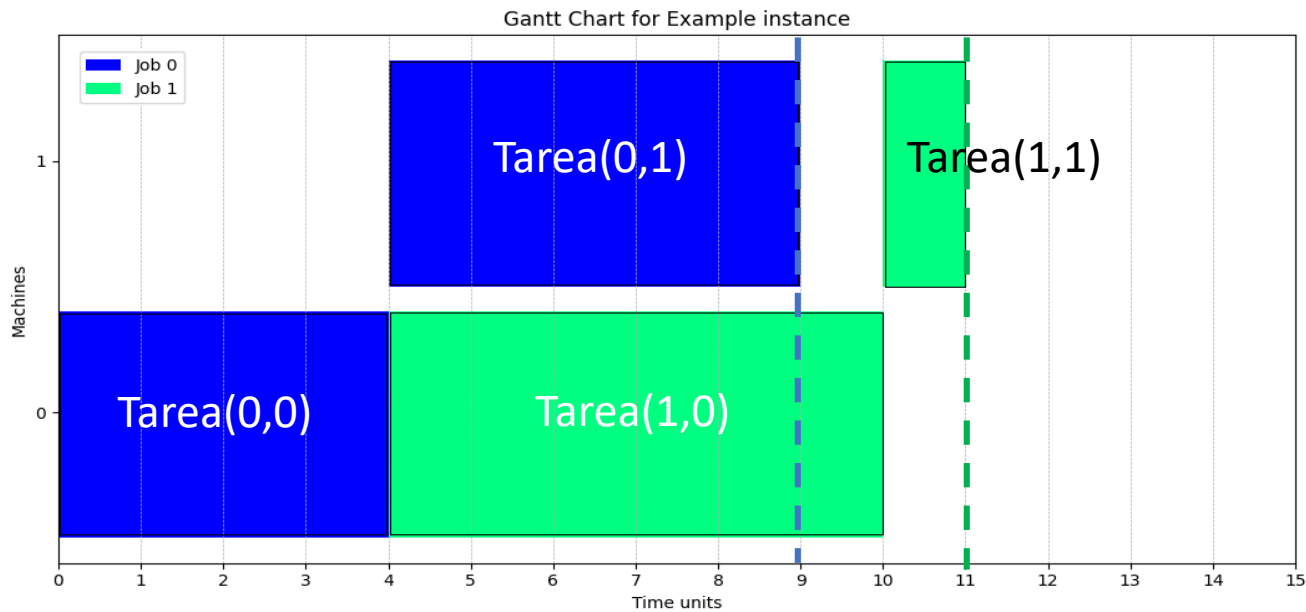
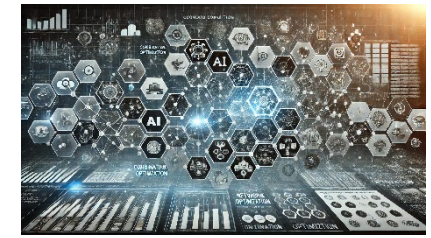


# Scheduling “Verde”: ¿Cuál es la solución óptima?

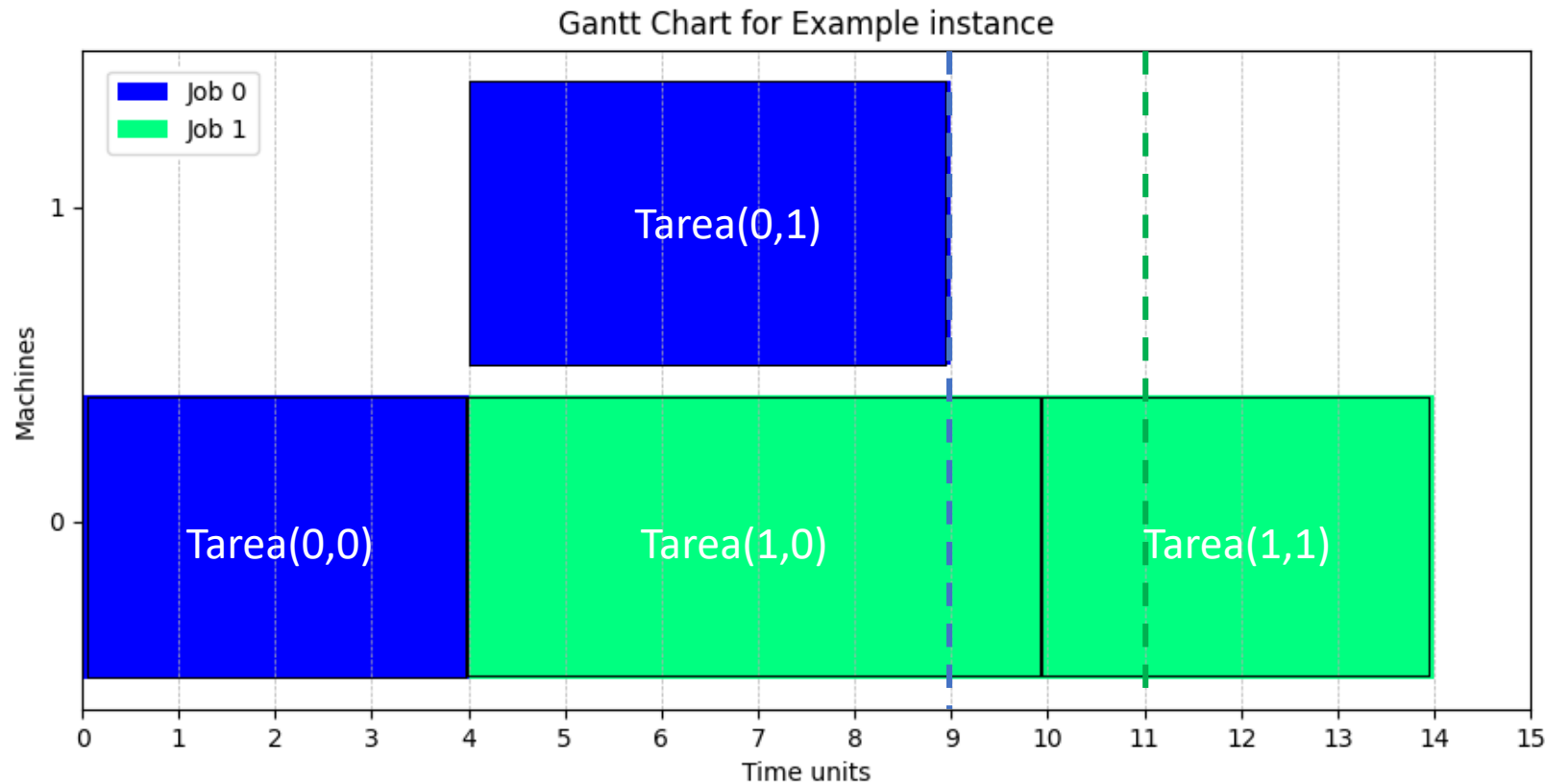
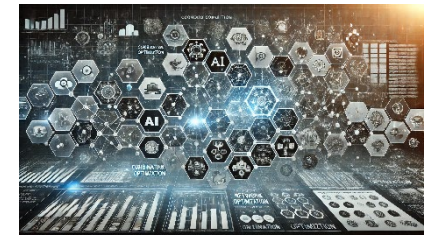


- Si minimizamos el tardiness, este schedule es óptimo con el valor objetivo 0...
- Pero ¿qué ocurre con el consumo energético?

# Scheduling “Verde”: ¿Cuál es la solución óptima?



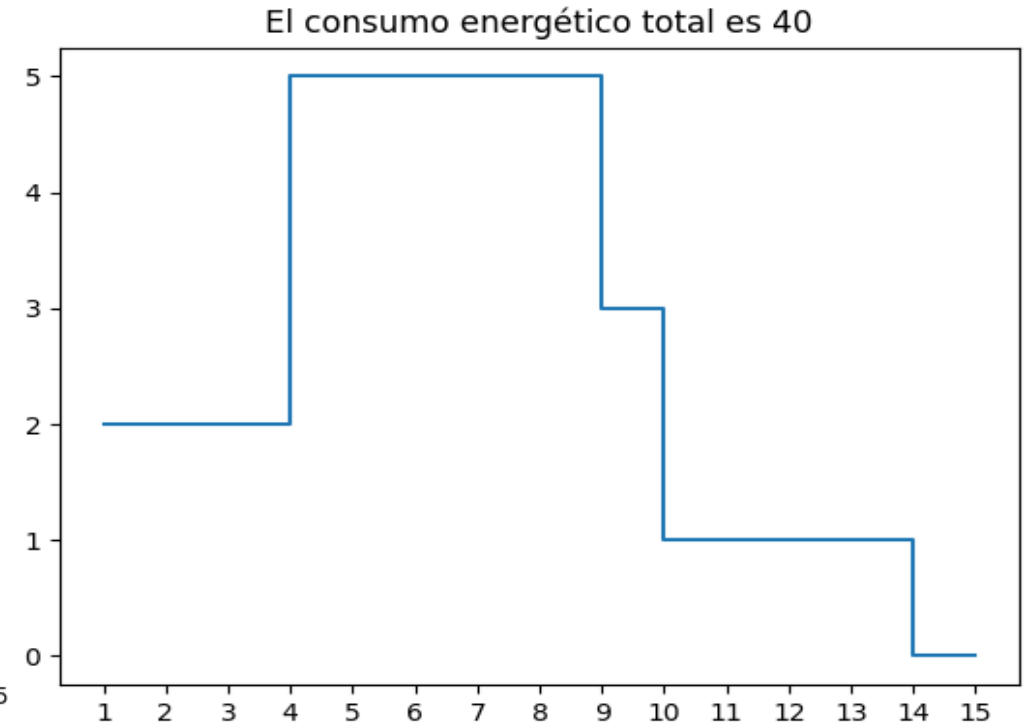
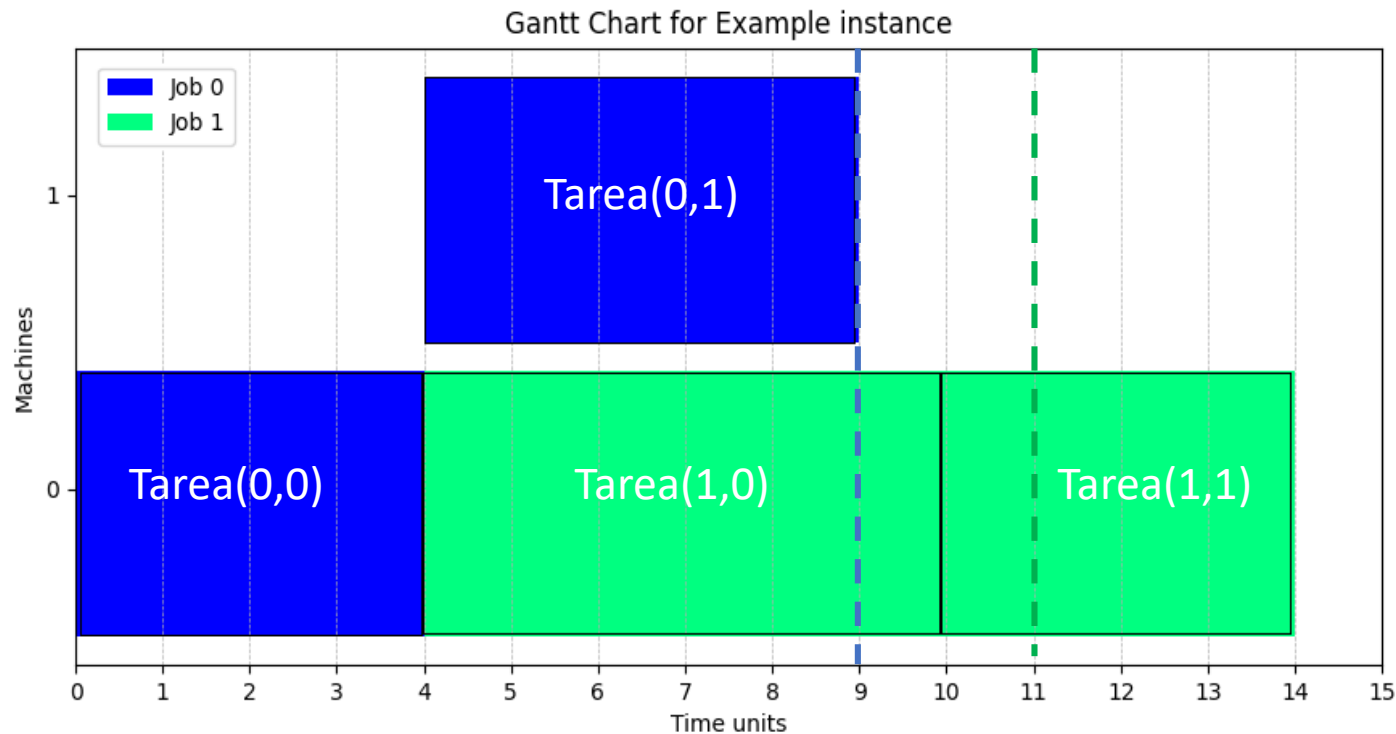
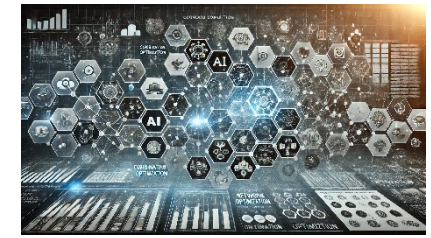
# Scheduling “Verde”: ¿Cuál es la solución óptima?



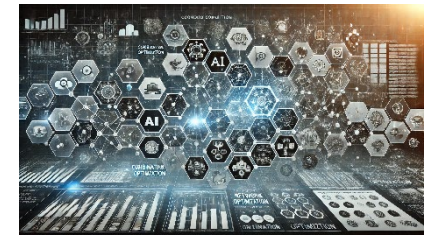
- Pero si minimizamos la energía total, esta solución es mejor



# Scheduling “Verde”: ¿Cuál es la solución óptima?

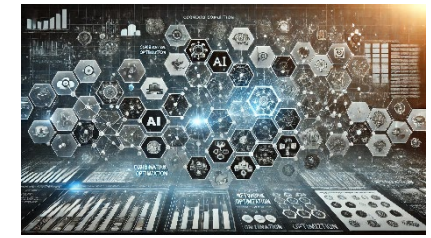


# Scheduling “Verde”: ¿Cuál es la solución óptima?

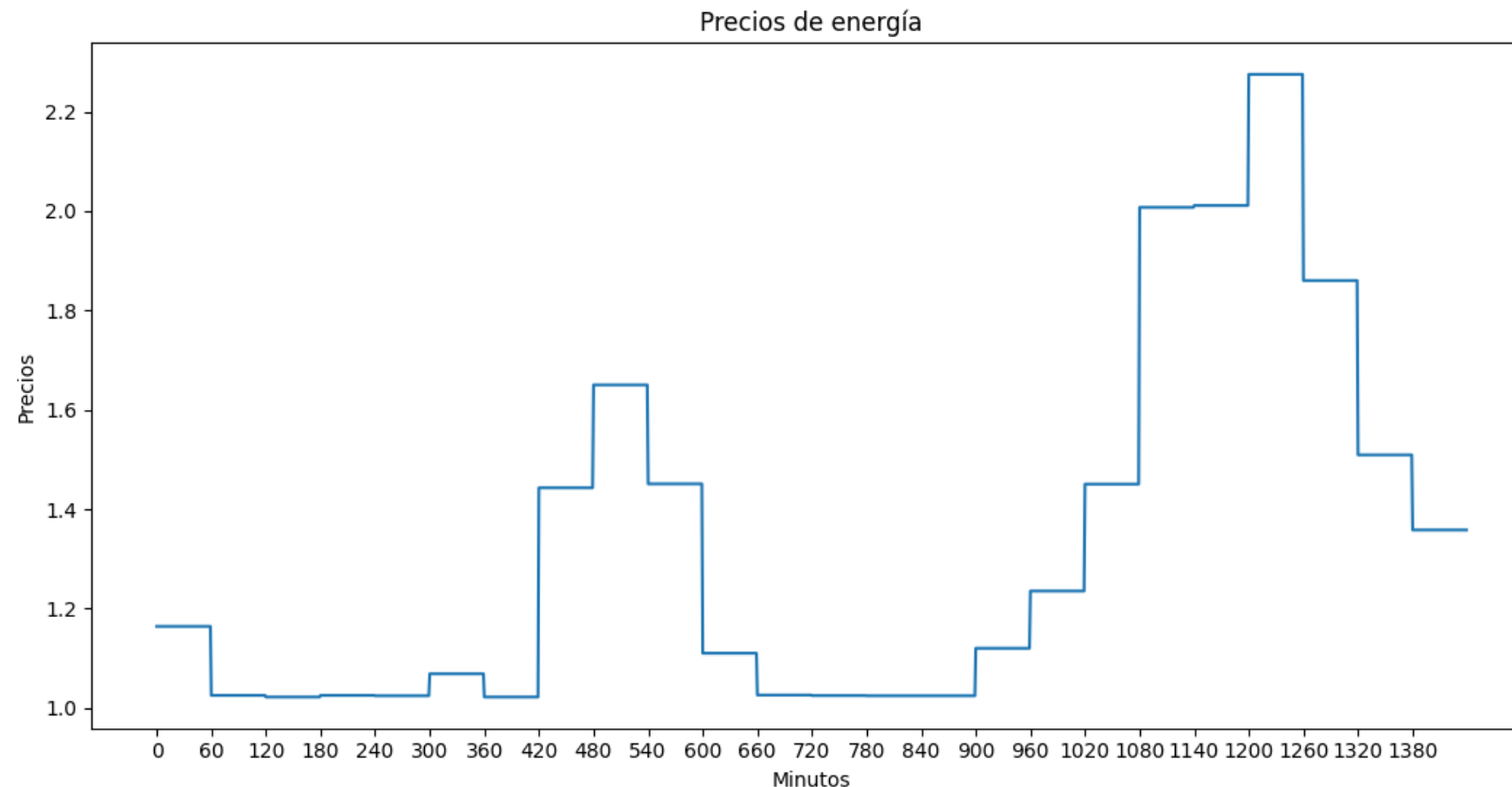


- Job shop clásico es un problema NP-duro
  - Ya lo hemos visto estos días
- Job shop flexible presenta más alternativas, más soluciones
  - Además de lo anterior, hay que decidir en qué máquina procesar cada tarea
- Job shop flexible verde introduce aún más retos...
  - Reducir los costes energéticos de las soluciones adoptadas

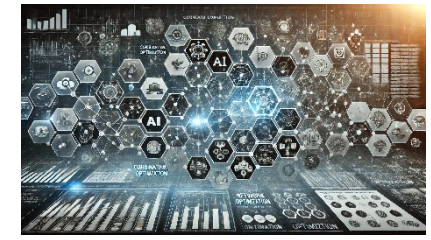
# Scheduling “Verde”: Tardiness vs Coste energético



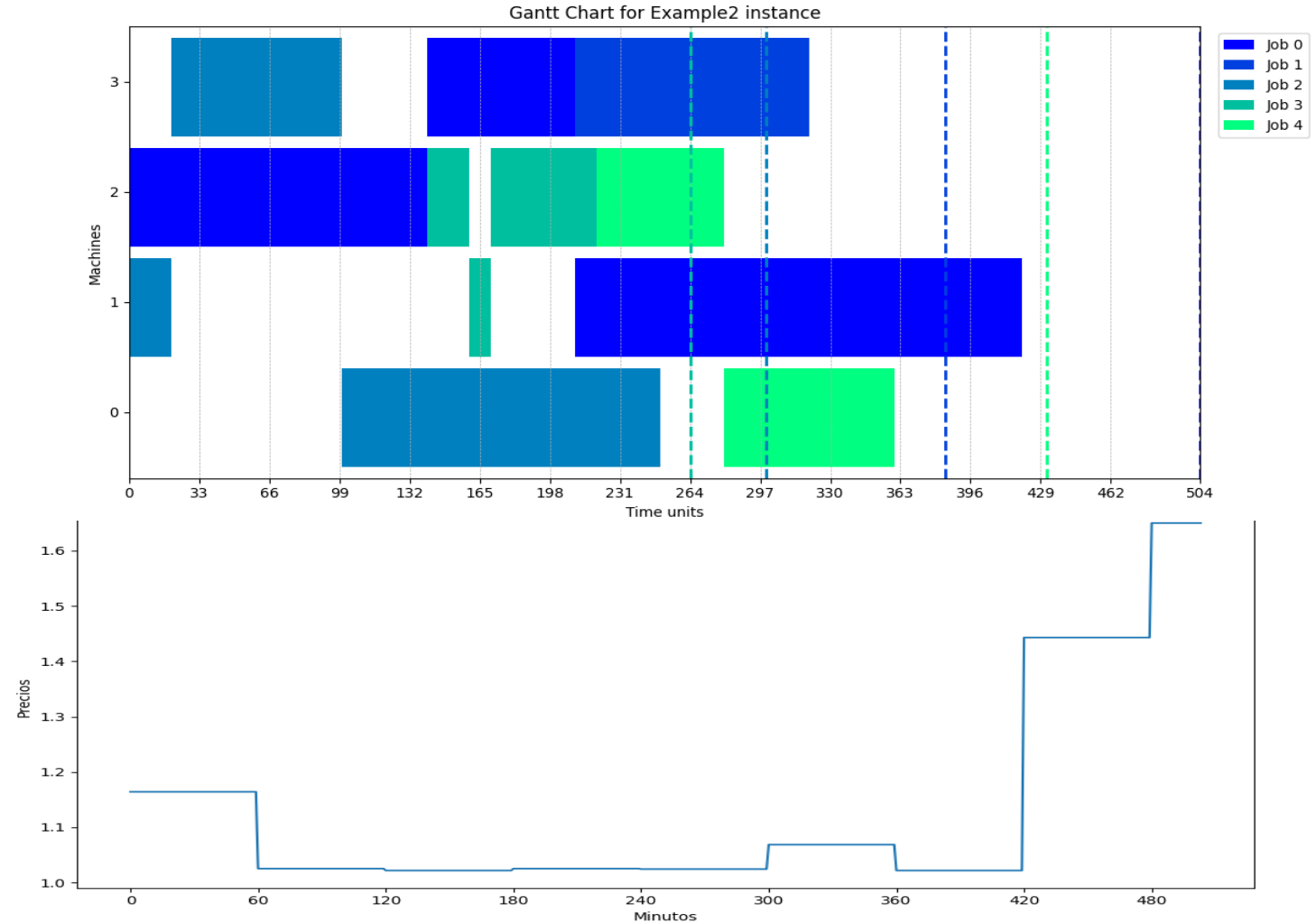
- Incorporamos los precios de energía por hora:



# Scheduling “Verde”: ¿Cuál es la solución óptima?



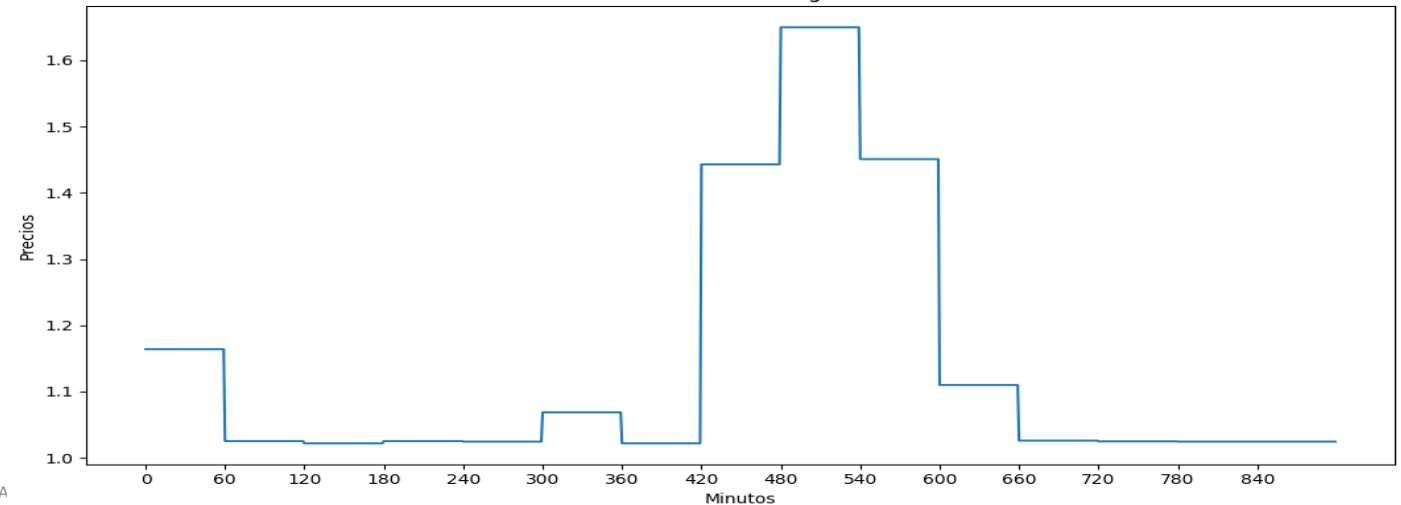
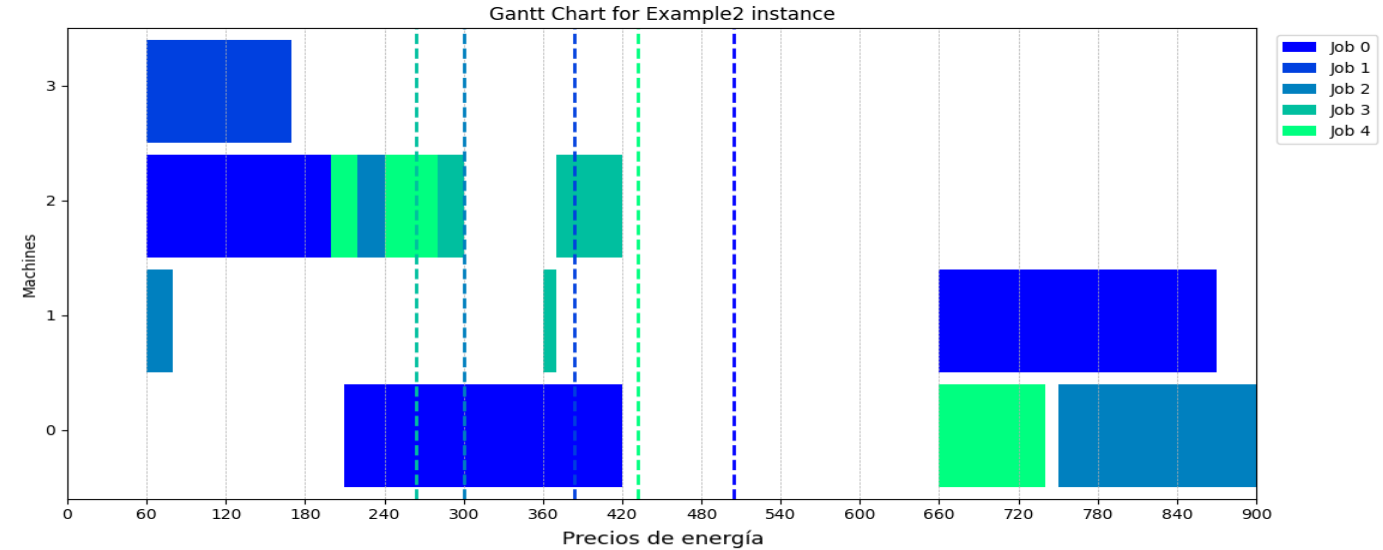
- Tenemos una instancia con 5 trabajos, 4 máquinas y 13 tareas en total.
- ¿Qué ocurre si solo minimizamos **el tardiness**?
- El tardiness total = 0
- El coste de energía = 615.1



# Scheduling “Verde”: ¿Cuál es la solución óptima?

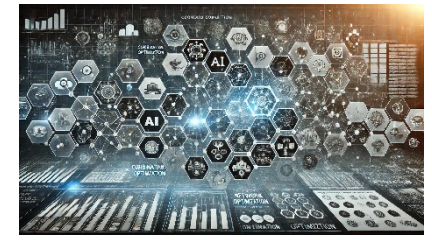


- Tenemos una instancia con 5 trabajos, 4 máquinas y 13 tareas en total.
- ¿Qué ocurre si solo minimizamos **el coste de energía**?
- El tardiness total = 1430
- El coste de energía = 550.7



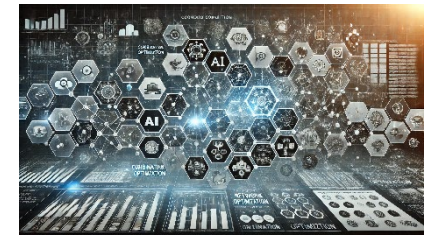


# Scheduling “Verde”: ¿Cuál es la solución óptima?



- La optimización monoobjetivo puede no corresponder a las realidades del sector
- Existen enfoques alternativos para abordarlo:
  - Optimización multi-objetivo: lexicográfico, objetivos combinados...
  - Programación con objetivos (goal programming)
  - Restricciones presupuestarias

# Scheduling “Verde”: ¿Cuál es la solución óptima?



- Optimización lexicográfica/jerárquica:
  - Se utiliza cuando el experto tiene un orden de prioridad entre los objetivos
  - Por ejemplo, puede querer lograr el mínimo tardiness posible y luego centrarse en minimizar el coste energético

```
my_obj = [total_tardiness, total_energy_cost]  
CPmodel.add(minimize_static_lex(my_obj))
```

# Scheduling “Verde”: Optimización lexicográfica

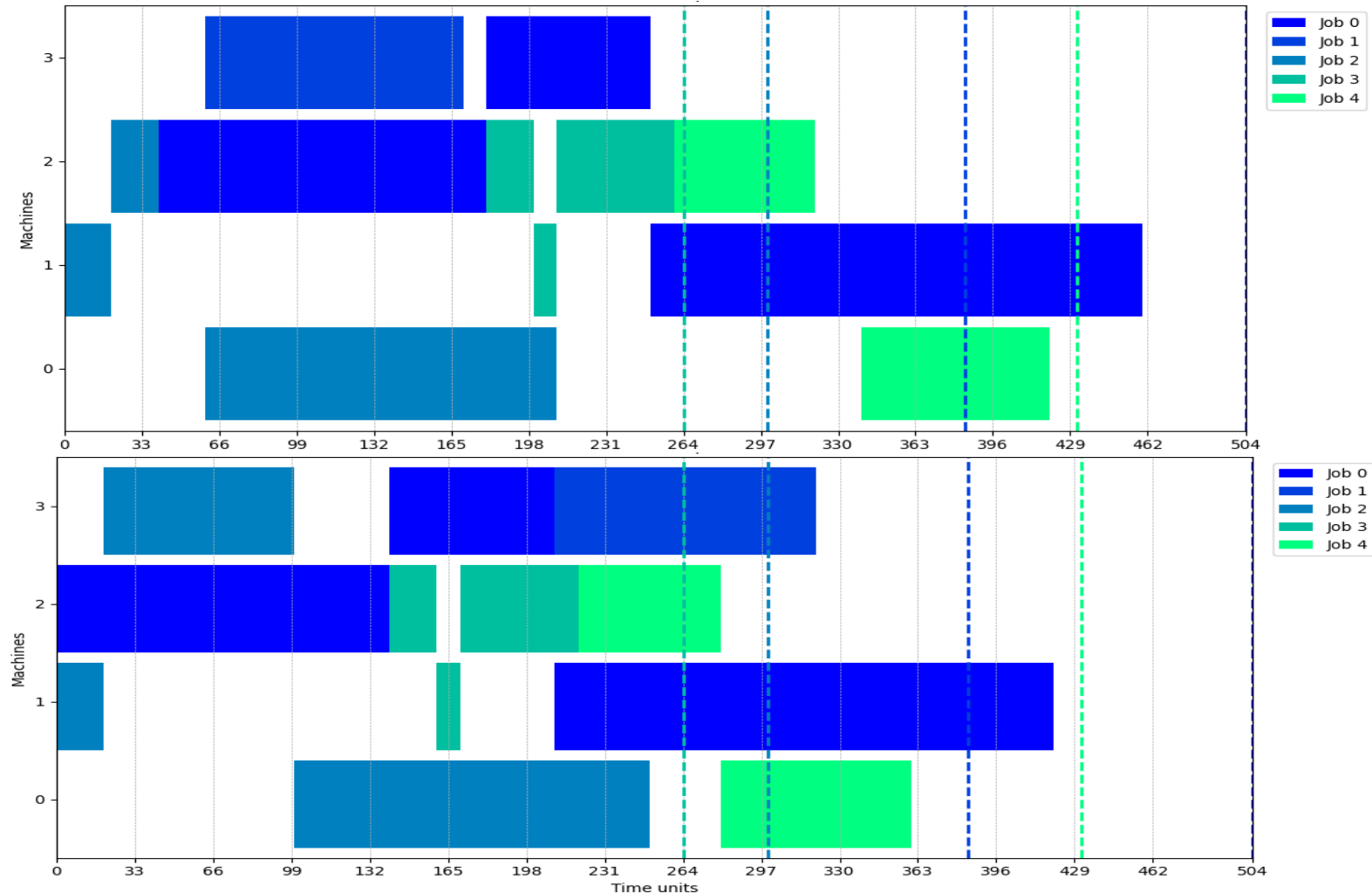


$\text{lex min } \{ \textit{tardiness total}, \textit{el coste energetico} \}$

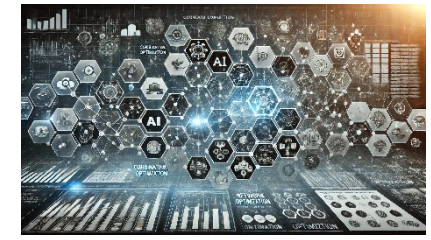
- El tardiness total = 0
- El coste de energía = 565.37

$\min\{\textit{tardiness total}\}$

- El tardiness total = 0
- El coste de energía = 615.1



# Scheduling “Verde”: Optimización lexicográfica

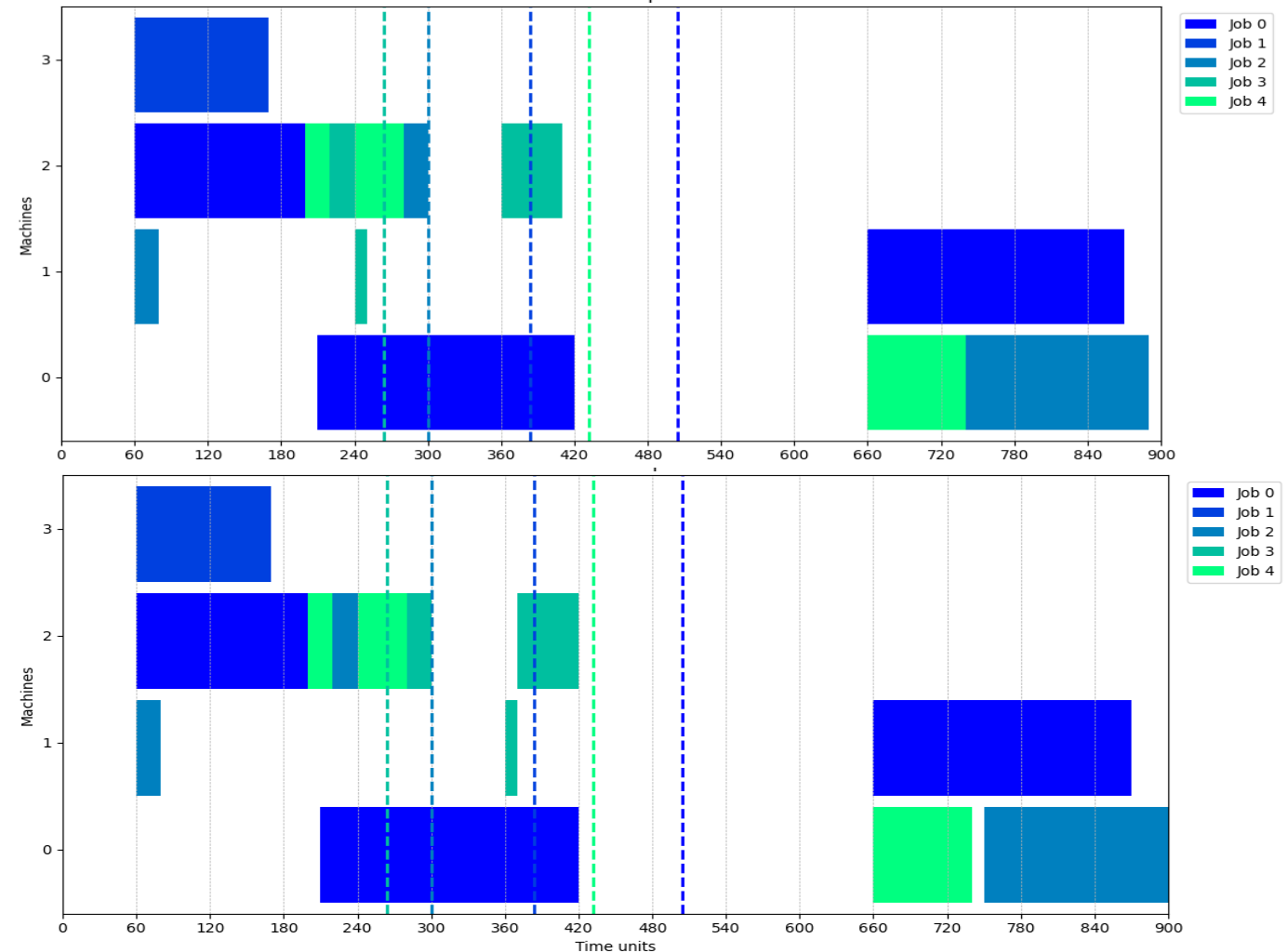


$\text{lex min } \{ \text{el coste energetico, tardiness total} \}$

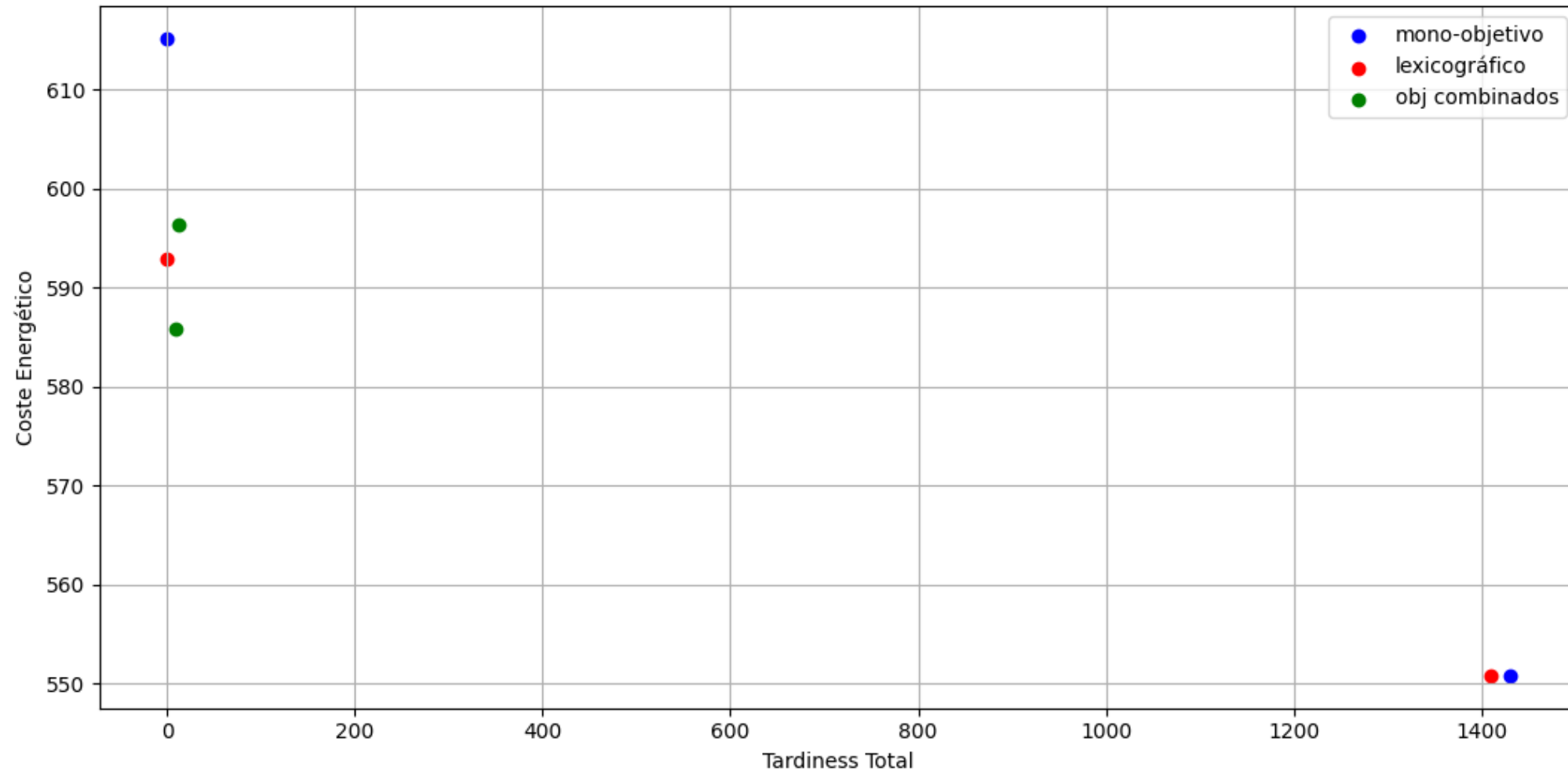
- El tardiness total = 1410
- El coste de energía = 550.7

$\text{min} \{ \text{el coste energetico} \}$

- El tardiness total = 1430
- El coste de energía = 550.7

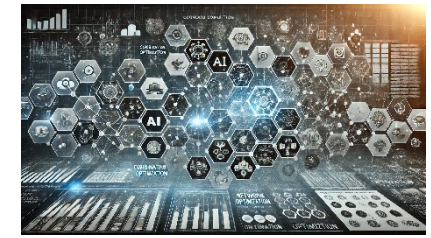


# Scheduling “Verde”: Objetivos combinados

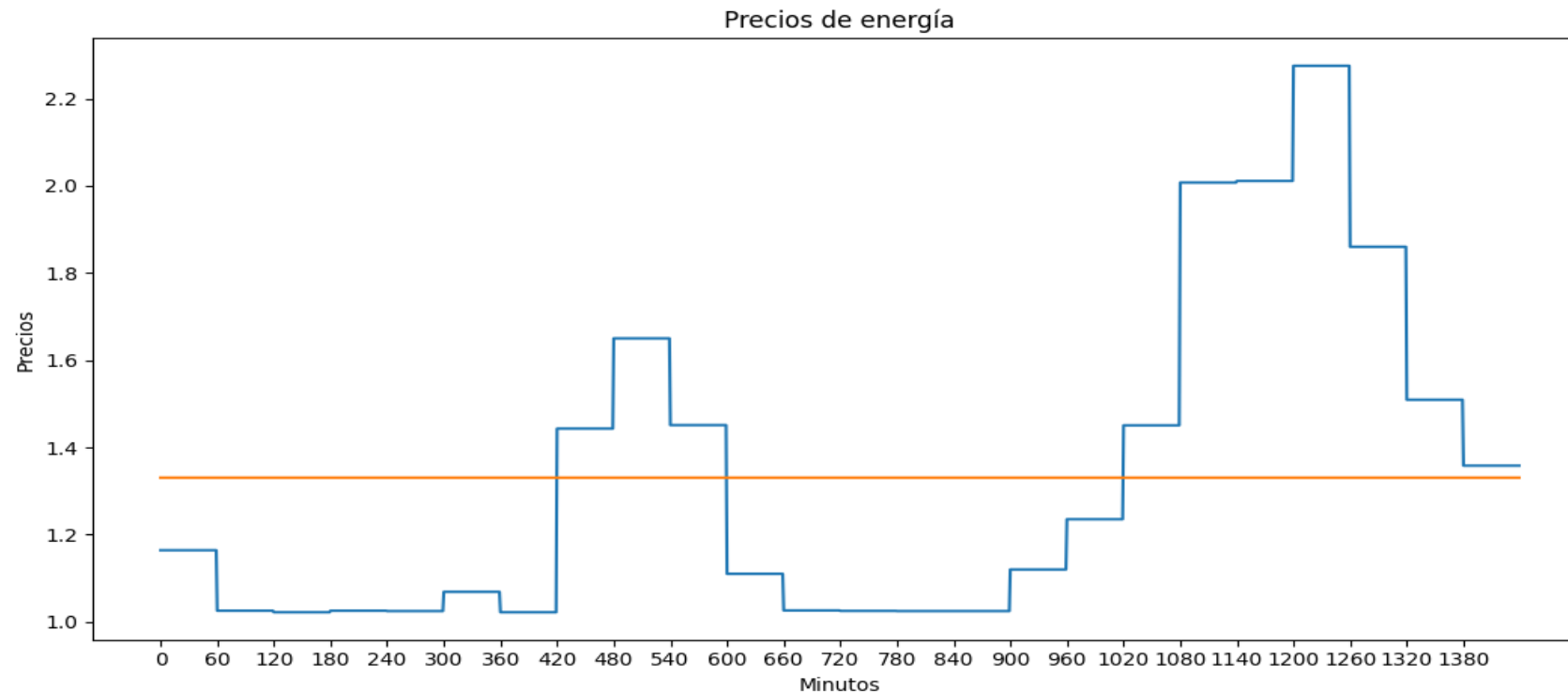




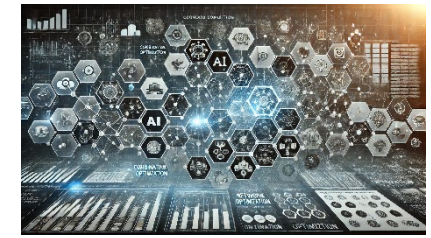
# Scheduling “Verde”: El coste vs el consumo de energía



Si tenemos las **tarifas fijas**, minimizar el coste de energía es equivalente a minimizar el consumo de energía



# Scheduling “Verde”: El coste vs el consumo de energía

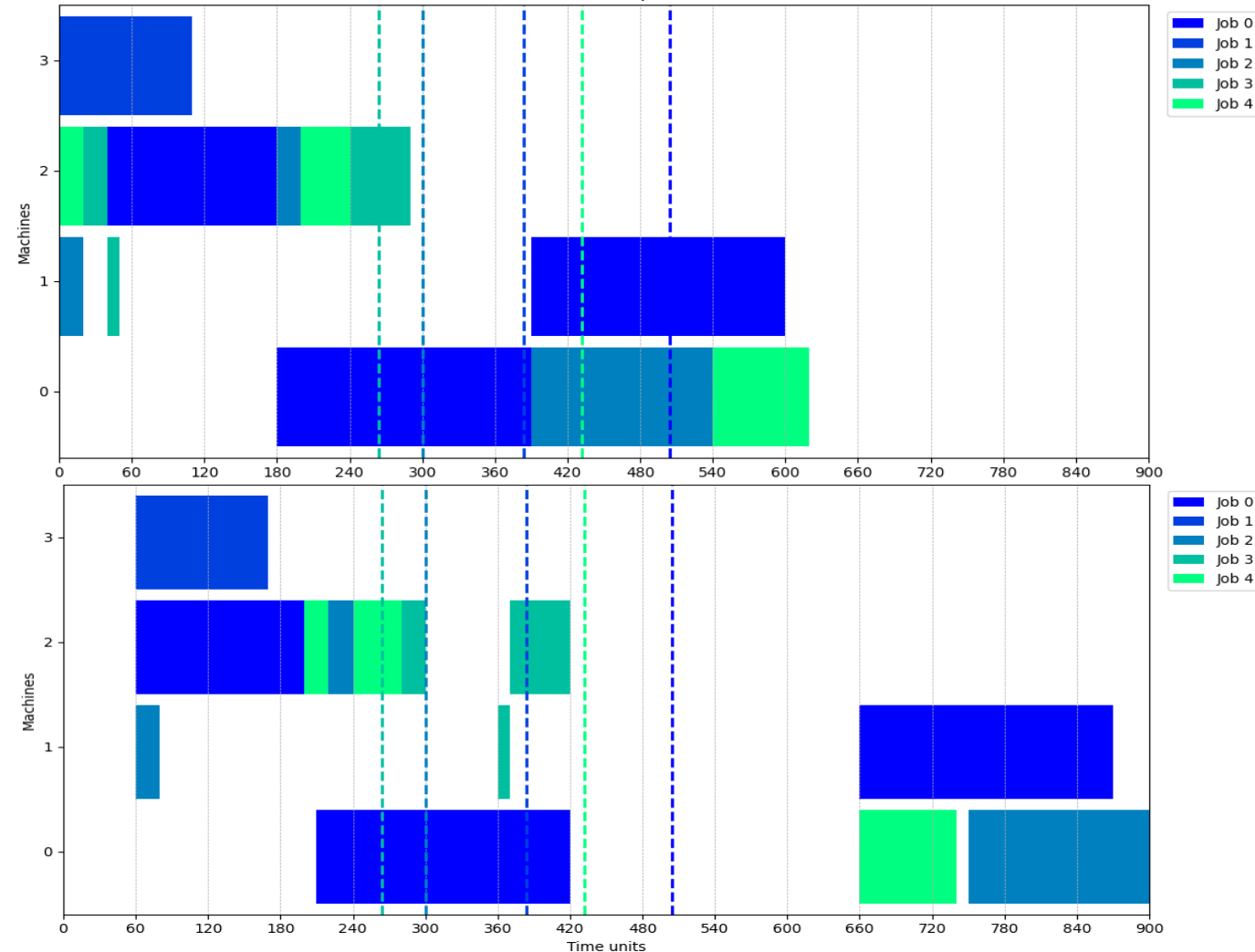


$\min \{ \text{el coste energetico} \}$   
 $s.a. \text{ tarifa fija}$

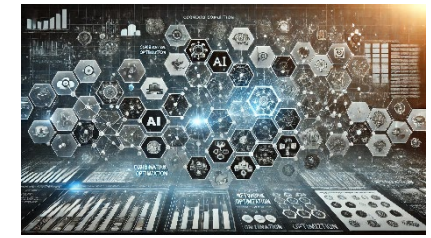
- El tardiness total = 550
- El coste de energía = 717.6
- El consumo de energía = 53960

$\min \{ \text{el coste energetico} \}$   
 $s.a. \text{ precios por hora}$

- El tardiness total = 1430
- El coste de energía = 550.7
- El consumo de energía = 53960



# Scheduling “Verde”: El coste vs el consumo de energía

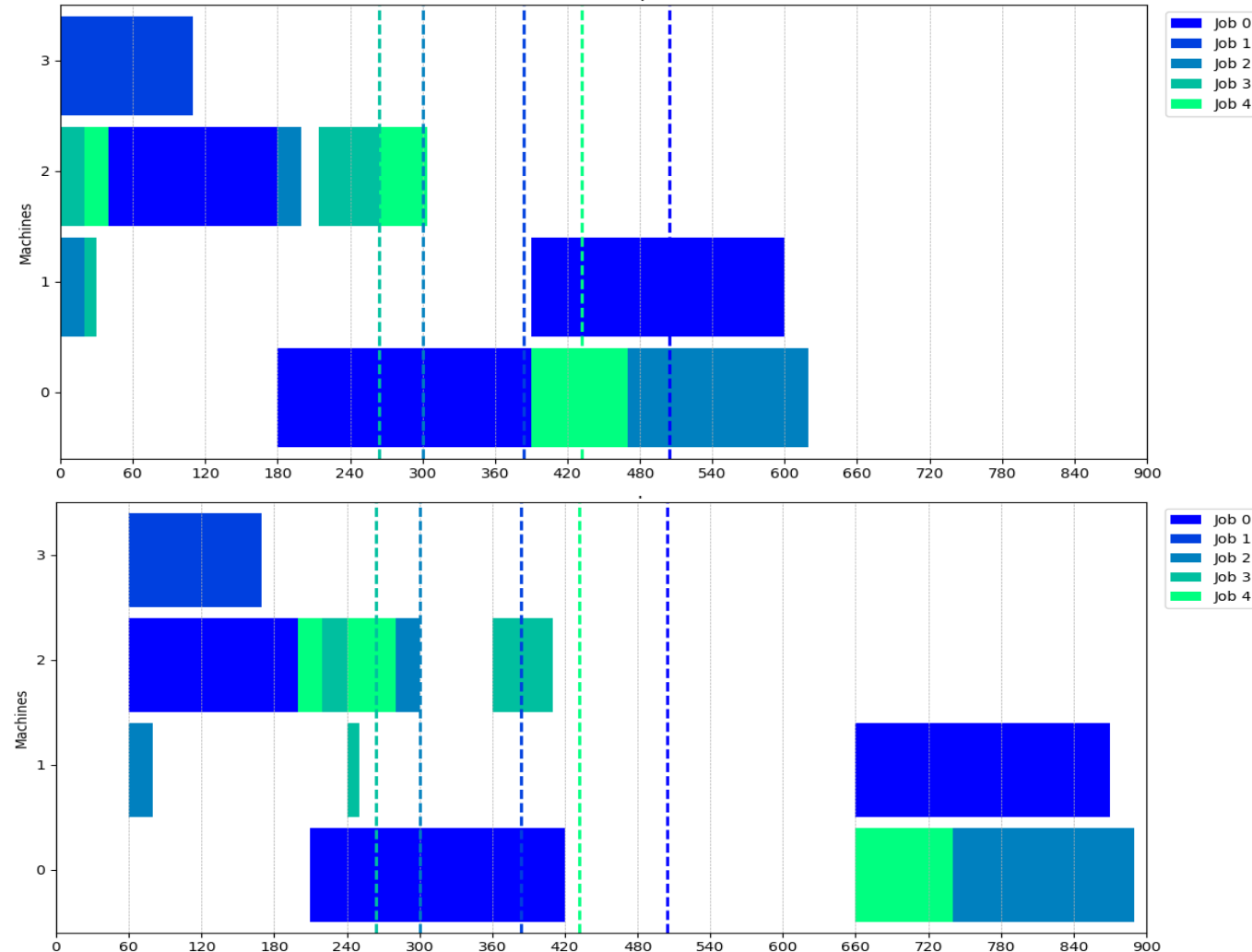


*{el coste energetico,  
lex min tardiness total}  
s.a. tarifa fija*

- El tardiness total = 454
- El coste de energía = 717.6
- El consumo de energía = 53960

*{el coste energetico,  
lex min tardiness total}  
s.a. precios por hora*

- El tardiness total = 1410
- El coste de energía = 550.7
- El consumo de energía = 53960

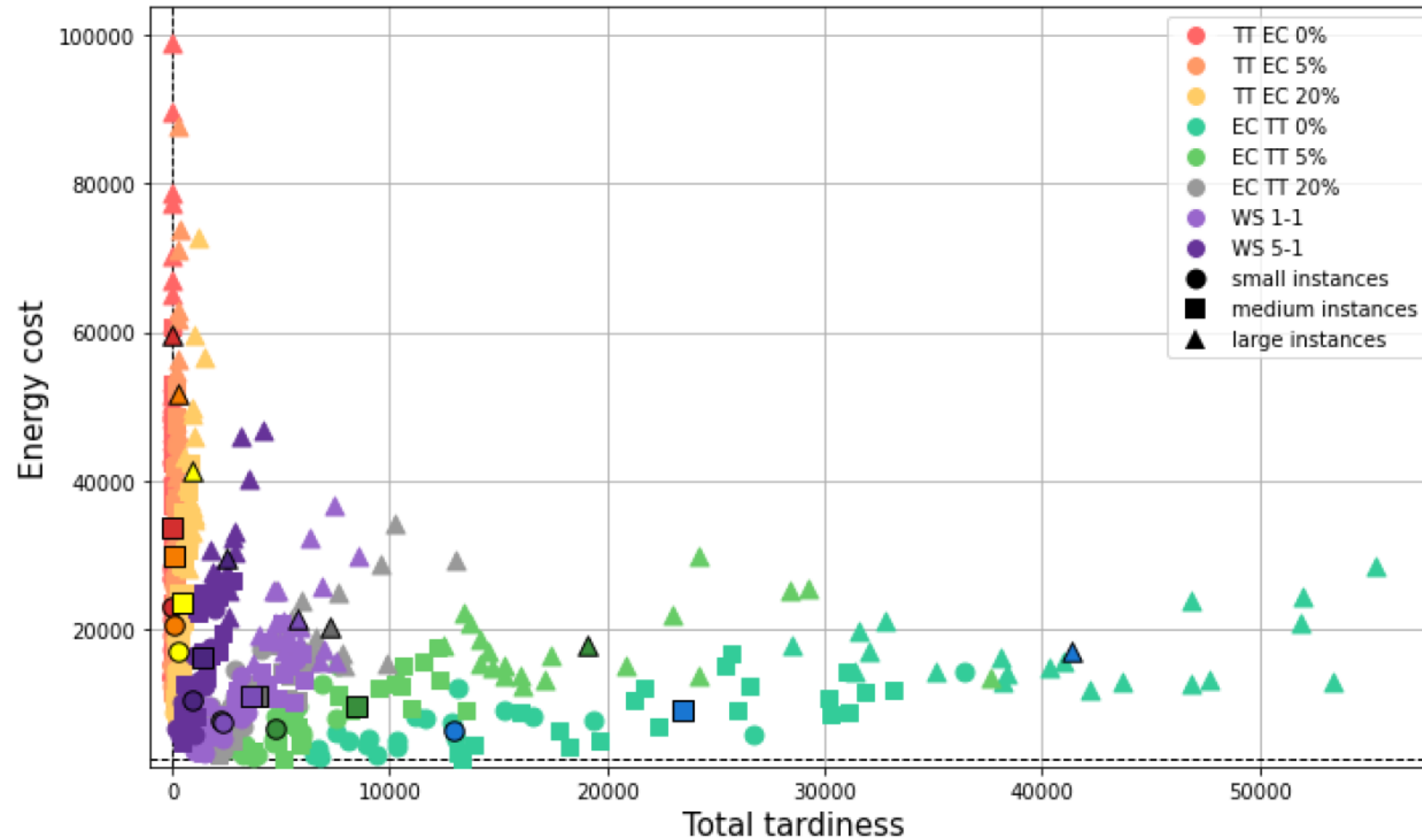


# Scheduling “Verde”: Optimización multi-objetivo



- Tenemos un proyecto en progreso con la Universidad de Groningen, Holanda
- Trabajamos sobre un variante del problema de JSP flexible verde
- Intentamos descubrir el espacio factible usando optimización lexicográfica, optimización con restricciones y objetivos combinados
- Los resultados de 60 instancias de distintos tamaños demuestran la relación conflictiva entre los objetivos

# Scheduling “Verde”: Optimización multi-objetivo





# Scheduling “Verde”: Conclusiones



- El consumo energético y los costes asociados son preocupaciones fundamentales en el mundo real
- Es crucial integrarlos a los problemas de scheduling clásicos
- Hay nuevos métodos para modelar y resolver esos problemas
- Es posible obtener schedules más eficientes energéticamente sin comprometer la calidad del servicio al cliente



Universidad de  
Oviedo



# Técnicas de Inteligencia Artificial para la Optimización y Programación de Recursos

## Tema 6: Aplicaciones de Scheduling en la vida real

Sezin Afsar  
[afsarsezin@uniovi.es](mailto:afsarsezin@uniovi.es)

Ciencia de la Computación e Inteligencia Artificial  
Departamento de Informática

