



Ajust de models  
d'optimització

# **Programació de tasques eficients en centres de dades per a estalvi energètic**

Carlota Fernández, Emma Juanico, Aránzazu Miguélez i Lucía Revaliente



# Índex

- Repàs model matemàtic **03**
- Generació de dades **06**
- Implementació **08**
- Resultats **10**
- Conclusions **12**

# Repàs model matemàtic



# Modelització matemàtica

## Funció objectiu

$$\text{Minimitzar: Energy} = \sum_{i=1}^n \sum_{j=1}^m x_{ij} \times \text{power}(P_j) \times \frac{\text{MI}(T_i)}{\text{speed}(P_j)}$$

- $\text{MI}(T_i)$ : Longitud de la tasca  $T_i$  en Milions d'Instruccions (MI).
- $\text{speed}(P_j)$ : Velocitat del processador  $P_j$ , en MIPS (Million Instructions Per Second).
- $\text{power}(P_j)$ : Consum de potència del processador  $P_j$ , en kilowatts (kW).

# Modelització matemàtica

## Restriccions

Limitació del temps d'execució

$$\sum_{i=1}^n \sum_{j=1}^m x_{ij} \times \frac{\text{MI}(T_i)}{\text{speed}(P_j)} \leq \text{deadline}$$

Capacitat del processador

$$\sum_{i=1}^n x_{ij} \times \text{MI}(T_i) \leq \text{Capacitat}(P_j), \quad \forall j = 1, 2, \dots, m$$

Assignació de tasques única

$$\sum_{j=1}^m x_{ij} = 1, \quad \forall i = 1, 2, \dots, n$$

$$x_{ij} \in \{0, 1\}$$

# Generació de dades



# Generació de dades: Pandas

## Process\_dataset

Task Id  
Process Id  
Processor Id  
MI: Milions d'instruccions  
velocitat: MiPS  
Potència: Kw

```
Task_ID,Process_ID,Processor_ID,MI,Speed,Power
T1,T1_P1,P7,1402,1457,1.26
T1,T1_P2,P8,2913,1285,0.54
T1,T1_P3,P5,526,1563,0.54
T1,T1_P4,P3,2231,1051,0.63
T1,T1_P5,P6,1638,1501,0.83
T2,T2_P1,P4,1878,2518,1.13
```

## Task\_deadlines

Task Id  
Deadline: MI / velocitat

```
Task_ID,Deadline
T1,2
T2,1
T3,2
T4,0
T5,1
T6,2
```

## Processor\_capacities

Processor Id  
Capacitat: MI

```
Processor_ID,Capacity
P1,46180
P2,24560
P3,21020
P4,50360
P5,31260
```

# Implementació





# Implementació: PuLP

1	<b>Carrega datasets:</b> Dades generades prèviament (MI, velocitats, potències). <code>pd.read_csv()</code>
2	<b>Formulació de la funció objectiu:</b> Modela la funció que calcula el consum energètic total, per minimitzar-lo. <code>problem = LpProblem("Minimitzar_Energia", LpMinimize)</code>
3	<b>Definició variables de decisió:</b> Dinàmiques, ja que són les que el model ha de canviar fins trobar la solució òptima. Declaració de $x_{ij}$ --> <code>LpVariable( <math>x_{ij}</math>, cat="Binary" )</code>
4	<b>Aplicació de les restriccions:</b> Temps d'execució, capacitat dels processadors, assignació única. Per cada restricció: <code>lpSum(...)</code>
5	<b>Resolució del problema:</b> Solver PuLP troba la solució òptima. <code>problem.solve()</code> <code>print("Estat de la solució:", problem.status)</code>

# Resultats i discrepàncies



# Resultats

## Prova 1

---

25 tasques  
9 processadors  
20.76 kWh

**NO FACTIBLE**

## Prova 2

---

50 tasques  
10 processadors  
31.52 kWh

**NO FACTIBLE**

## Prova 3

---

500 tasques  
200 processadors  
109.28 kWh

**FACTIBLE**

# Conclusions



# Conclusions

- El model minimitza el consum energètic en l'assignació de tasques.
- Resultats satisfactoris: robustesa i escalabilitat comprovades.

## Treball futur:

- Ampliar el model: incloure latència de comunicació.
- Treballar amb dades reals en lloc de sintètiques.
- Explorar mètodes heurístics per escales majors.



**Gràcies per la vostra atenció!**

**Alguna pregunta?**