

Planificación temporal y otras cosas

Planificación Automática
Sergio Santamaria Carrasco
NIA: 100425411

¿Qué es la Planificación Temporal?



- En la planificación temporal en las que las acciones no se suceden necesariamente de forma secuencial, tienen una duración y pueden solaparse. La planificación temporal puede expresar factores como plazos, efectos condicionales, condiciones durante la aplicación de las acciones o efectos que se producen en puntos temporales arbitrarios (Fox y Long 2003).



Dominio de pruebas: Cadena de ensamble



- **Componentes:**

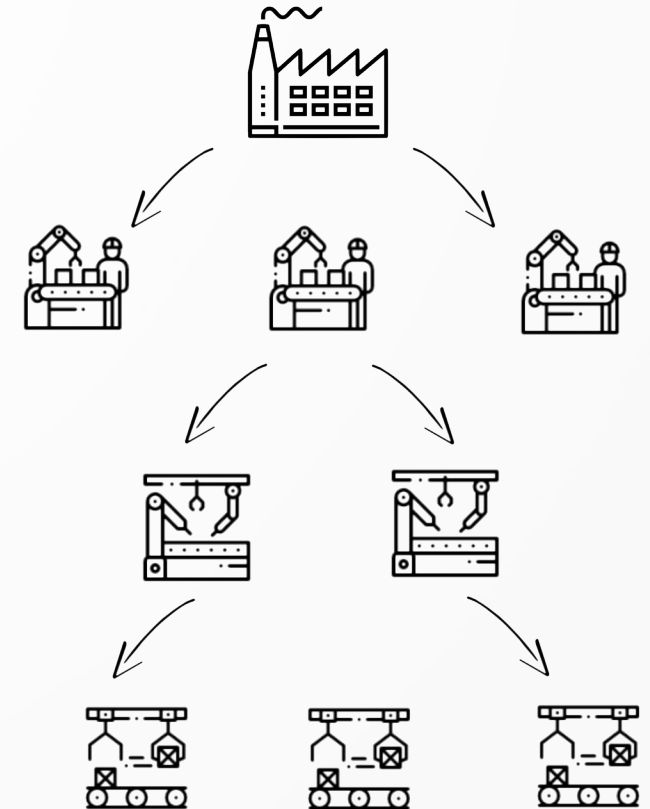
- P piezas
- F procesos que aplicar a las piezas para obtener la pieza final
- M máquinas capaces de realizar los diferentes procesos en un tiempo t

- **Objetivo:**

- Obtener P piezas finales

- **Métrica:**

- Minimizar el tiempo necesario



Modelo I: PDDL 2.2



- **Tipos:**

- piece
- machine
- proccess

- **Functions:**

- (time-needed ?p - proccess ?m - machine)

- **Predicados:**

- (is-done ?p -proccess ?x - piece)
- (is-avaible ?m - machine)
- (is-first ?p1 ?p2 - proccess)
- (support-procces ?m - machine ?p - proccess)

- **Acciones:**

```
(:durative-action apply
  :parameters
    (?p1 ?p2 - proccess
     ?m - machine
     ?p - piece)

  :duration
    (= ?duration (time-needed ?p2 ?m))

  :condition
    (and
      (at start (is-done ?p1 ?p))
      (at start (is-first ?p1 ?p2))
      (at start (is-avaible ?m))
      (at start (support-procces ?m ?p2)))

  :effect
    (and
      (at start (not (is-avaible ?m)))
      (at end (is-avaible ?m))
      (at end (is-done ?p2 ?p)))
)
```

- **Factor de ramificación:**

$$PMF^2$$

- P número de piezas
- M número de máquinas
- F número de procesos

- **Ejemplo de ejecución:**

- 3 piezas
- 3 máquinas
- 4 procesos

PLAN	SOLVE TIME	COST TIME	EVALUATED STATES
1	0	6	10
2	424.36	5	156613
3	951.68	5	334163

Modelo II: PDDL 2.2



- **Tipos:**

- piece
- machine
- proccess

- **Functions:**

- (time-needed ?p - proccess ?m - machine)
- (num-pieces ?p - proccess)

- **Predicados:**

- (is-done ?p - proccess ?x - piece)
- (is-avaible ?m - machine)
- (is-first ?p1 ?p2 - proccess)
- (support-procces ?m - machine ?p - proccess)

- **Acciones:**

```
(:durative-action apply
  :parameters
    (?p1 ?p2 - proccess
     ?m - machine)

  :duration
    (= ?duration (time-needed ?p2 ?m))

  :condition
    (and
      (at start (is-first ?p1 ?p2))
      (at start (is-avaible ?m))
      (at start (support-procces ?m ?p2))
      (at start (>= (num-pieces ?p1) 1))
    )

  :effect
    (and
      (at start (not (is-avaible ?m)))
      (at start (decrease (num-pieces ?p1) 1))
      (at end (is-avaible ?m))
      (at end (increase (num-pieces ?p2) 1))
    )
)
```

- **Factor de ramificación:**

$$MF^2$$

- M número de máquinas
- F número de procesos

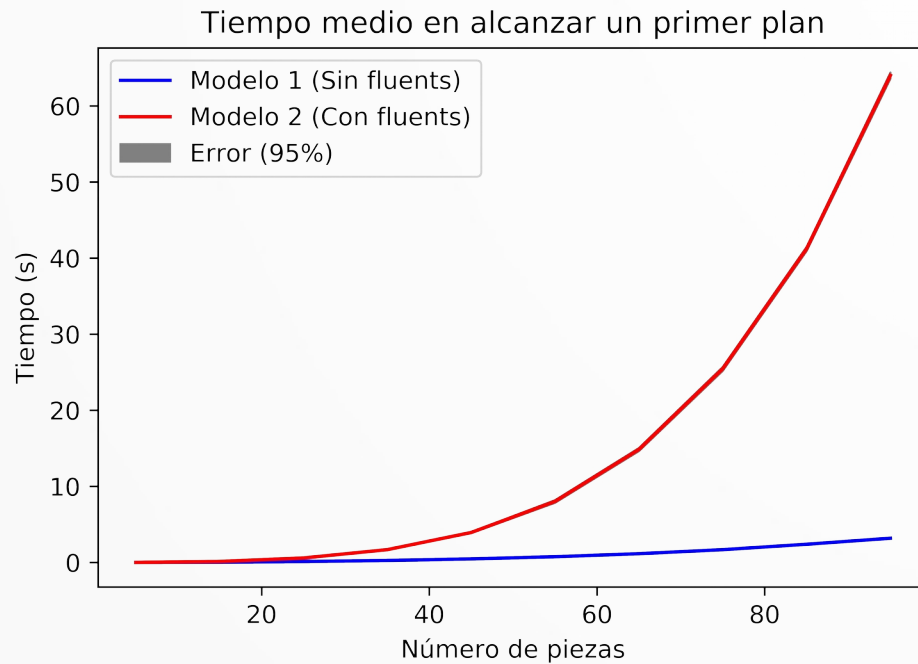
- **Ejemplo de ejecución:**

- 3 máquinas
- 4 procesos

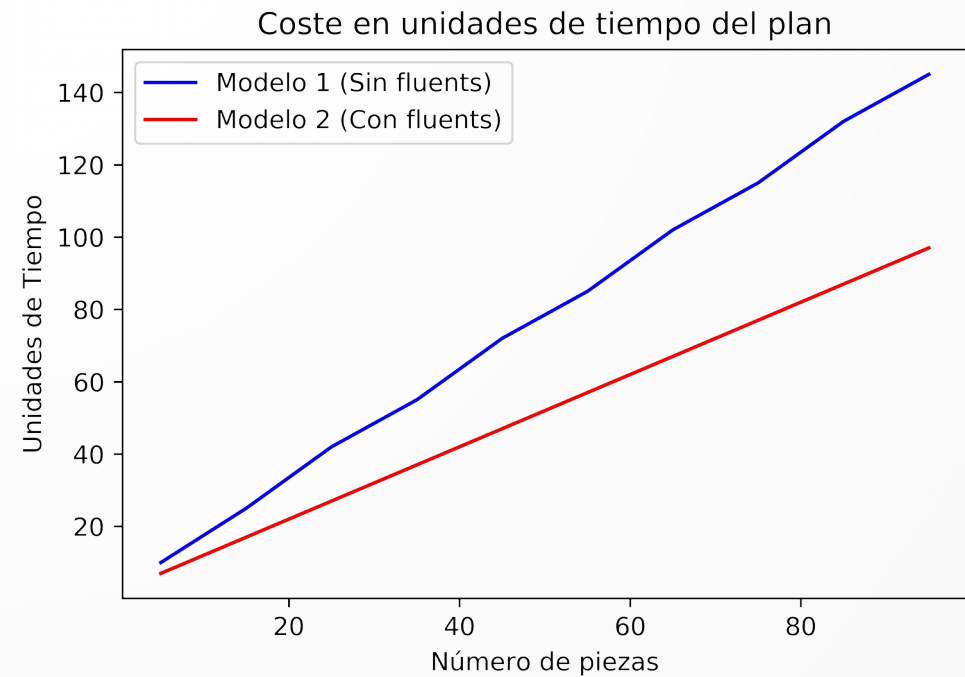
	NUM PIEZAS	SOLVE TIME	COST TIME	EVALUATED STATES	FINISH
MODELO ₁	3	951.68	5	334163	TRUE
MODELO ₂	3	2.35	5	1581	TRUE
MODELO ₁	4	-	-	-	FALSE
MODELO ₂	4	20.68	6	11870	TRUE
MODELO ₁	5	-	-	-	FALSE
MODELO ₂	5	114.54	7	62632	TRUE
MODELO ₁	6	-	-	-	FALSE
MODELO ₂	6	-	-	-	FALSE

Comparación entre modelos I y II

- Tiempo medio en alcanzar un plan inicial**




- Coste en unidades de tiempo del plan**




Ventajas y desventajas



Modelo 1

Más rápido en obtener un plan inicial	Duración del plan obtenido lejos del óptimo
	Mayor coste computacional (Más nodos evaluados)

Modelo 2

Duración del plan obtenido más cerca del óptimo	Más lento obteniendo un primer plan
Menor coste computacional	

Modelo III: Programación Lineal

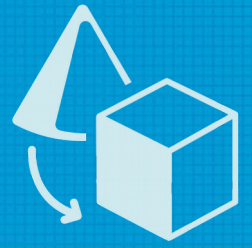


- La programación lineal es una técnica para la optimización de una función objetivo lineal, sujeta a restricciones de igualdad y desigualdad lineal.

$$\begin{array}{ll} \text{Maximize} & \mathbf{c}^T \mathbf{x} \\ \text{subject to} & A\mathbf{x} \leq \mathbf{b} \\ \text{and} & \mathbf{x} \geq \mathbf{0} \end{array}$$

- Traducción:
 - Tipos → Conjuntos
 - Estado meta → Restricciones
 - Acciones → Variables
 - Precondiciones → Restricciones

Modelo III: Traducción



- Tipos → Conjuntos

- machine → \mathcal{M} conjunto de máquinas
- process → \mathcal{F} conjunto de procesos

- Acciones → Variables

- apply → $\alpha_{f,m}^t \begin{cases} 1 \text{ si se aplica el proceso } f \text{ en la máquina } m \\ \text{durante el periodo } t \\ 0 \text{ en otro caso} \end{cases} \quad \forall m \in \mathcal{M}, f \in \mathcal{F}, t \in \mathcal{T}$

- Precondiciones → Restricciones

- (support-process ?m ?f) → $\alpha_{m,f}^t \leq c_m^f$
 $\forall m \in \mathcal{M}, f \in \mathcal{F}, t \in \mathcal{T}$

- Estado meta → Restricciones

- ```
(:goal
 (and
 (= (num-pieces process_3) 6)
))
```

 →  $\sum_{m'}^{\mathcal{M}} \sum_{t'=1}^{|T|-t_{m'}^f} \alpha_{m',f}^{t'} = d^f$   
 $\forall f \in \mathcal{F}$

- También se añaden nuevos elementos

$t_m^f$  duración del proceso  $f$  en  $m$ ,  $f \in \mathcal{F}, m \in \mathcal{M}$ .

$d^f$  demanda de piezas con el proceso  $f$  aplicado,  $f \in \mathcal{F}^{final}$ .

$c_m^f$  compatibilidad del proceso  $f$  en la máquina  $m$ ,  $f \in \mathcal{F}, m \in \mathcal{M}$ .

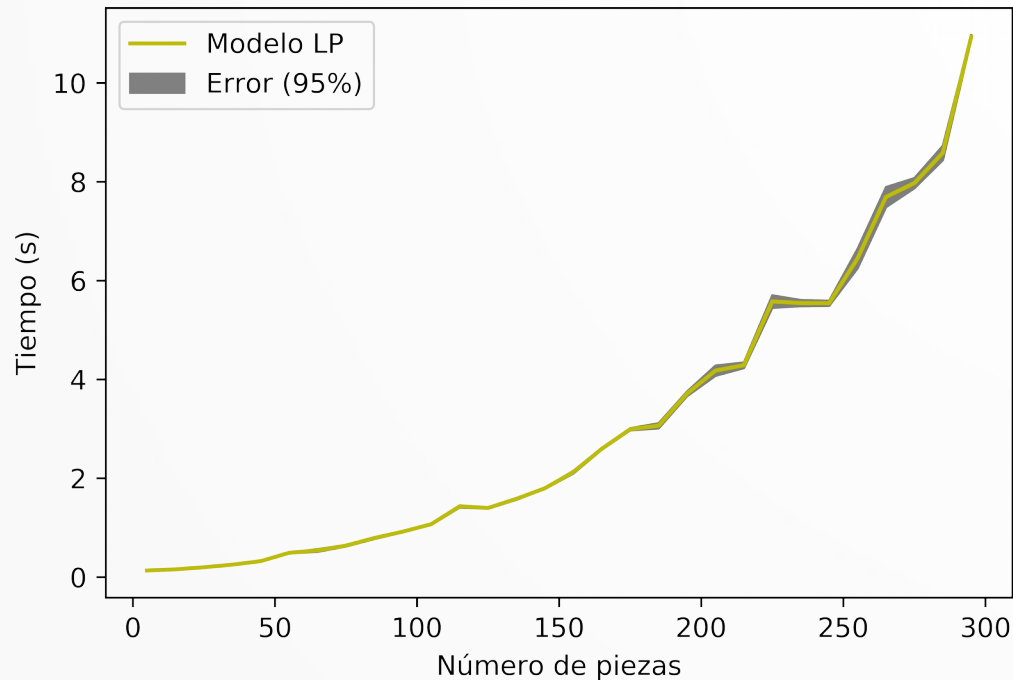
$c_m^f = 1$  si el proceso  $f$  se puede aplicar en la máquina  $m$ .

$c_m^f = 0$  en otro caso.

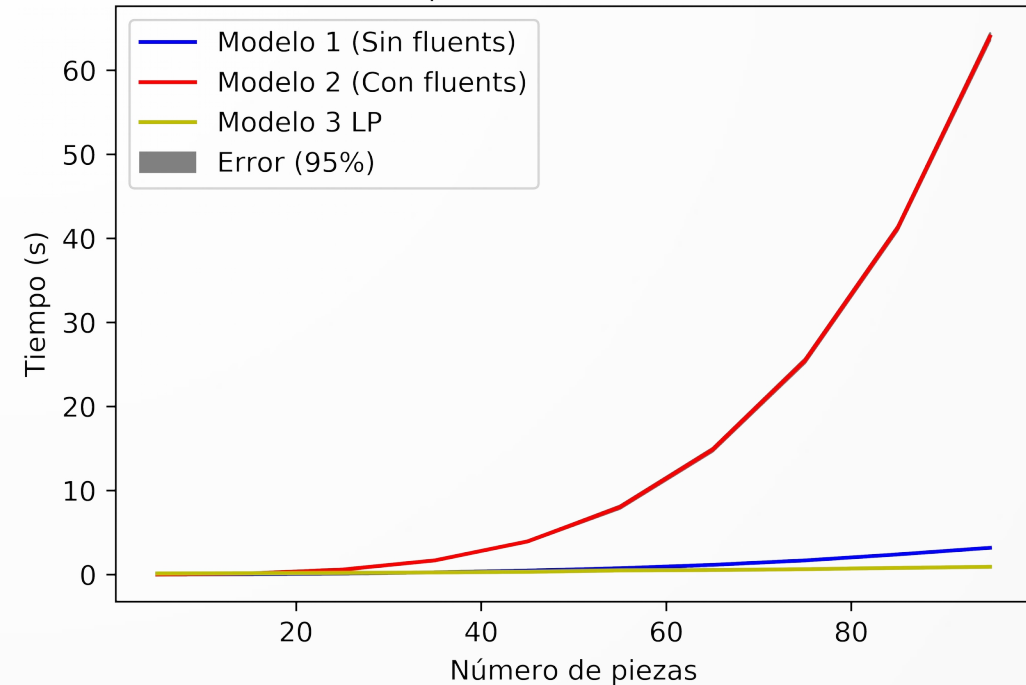
$\mathcal{T}$ , conjunto de periodos de tiempo en el horizonte de planificación.

# Modelo III: Experimentación

Tiempo en alcanzar el plan óptimo el modelo 3



Comparación entre modelos

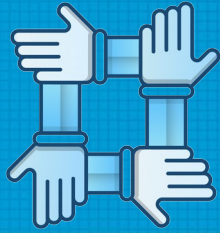


- Ventajas
  - Mayor escalabilidad
  - Mucho más rápido
- Desventajas
  - Requiere un horizonte de planificación como entrada

Heurística + Modelo III



# Modelo IV: Portfolio

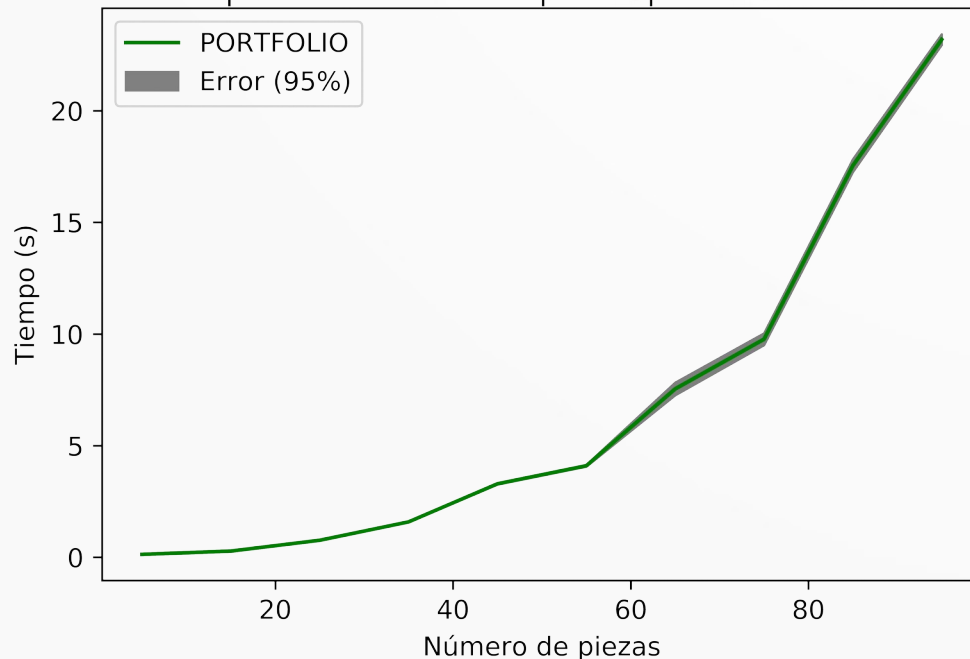


- Funcionamiento:

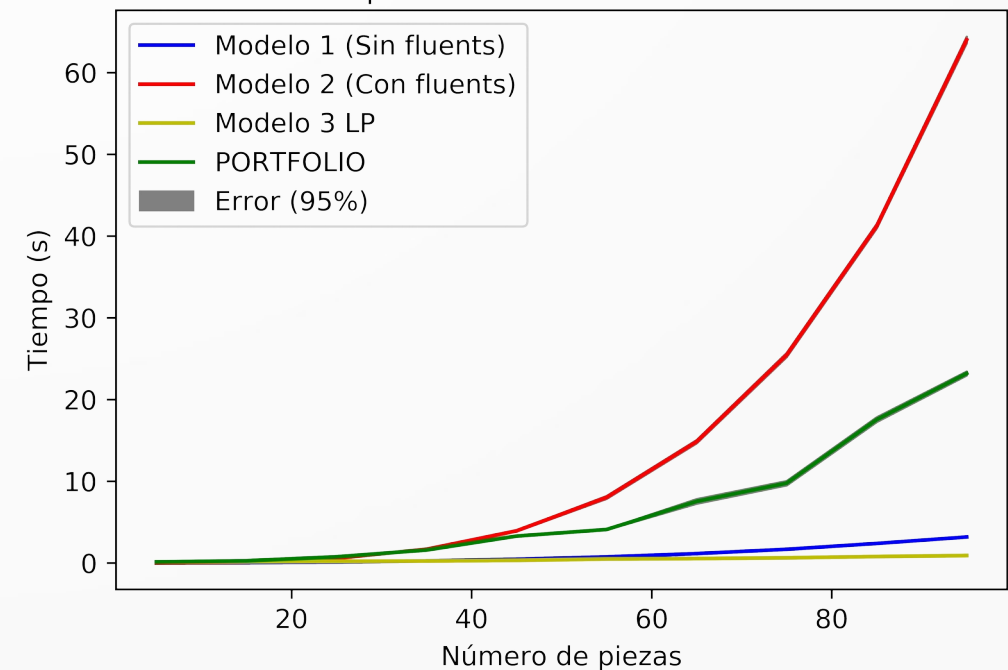
- Modelo I nos da un plan inicial con una aproximación del horizonte de planificación
- Modelo III refina el plan y minimiza el horizonte de planificación

- Resultados:

Tiempo en alcanzar el plan óptimo el modelo 4



Comparación entre modelos



# Conclusiones



- Resultados muy dependientes de la implementación
- Cada modelo presenta sus ventajas
- La unión de los modelos obtiene el mejor resultado

# Bibliografía

- Edelkamp, S., & Hoffmann, J. (2004). PDDL 2.2: The Language for the Classical Part of the 4th International Planning Competition, Albert Ludwigs Universität Institut für Informatik, Freiburg. freiburg. Tech. rep., Germany, Technical Report.
- Murty, K. G. (1983). Linear programming. Chichester.
- Gurobi Optimization, I. (2015). Gurobi optimizer reference manual. URL <http://www.gurobi.com>.
- Hart, W. E., Laird, C. D., Watson, J. P., Woodruff, D. L., Hackebeit, G. A., Nicholson, B. L., & Sirola, J. D. (2017). Pyomo-optimization modeling in python (Vol. 67). Berlin: Springer.
- Merkel, D. (2014). Docker: lightweight linux containers for consistent development and deployment. Linux journal, 2014(239), 2.
- <https://nms.kcl.ac.uk/planning/software/optic.html>
- <https://github.com/ssantamaria94/planificacion>