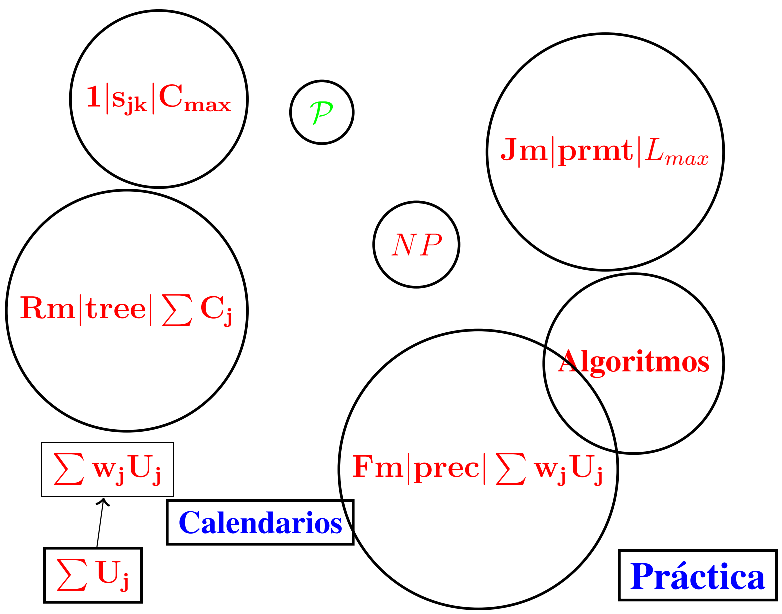
MODELOS DE PLANIFICACIÓN

Santiago de Compostela, Octubre 2006



CONTENIDO.

Contents

1 Descripción del problema.

"**SCHEDULING**", **ELABORACIÓN DE CALENDARIOS o PLANIFICACIÓN**: asignación de recursos escasos a *tareas* a lo largo del tiempo para alcanzar un objetivo.

♦ **Fabricación de etiquetas**.

- ▶ Impresión del logotipo, precio, etc.
- ▶ Recorte.
- ▶ Empaquetado.
- ♣ Minimizar el retraso en la entrega.

♦ **Construcción de una línea férrea de alta velocidad**.

- ▶ Elaboración del proyecto.
- ▶ Trámites administrativos.
- ▶ Ejecución de la obra.
- ♣ Minimizar la duración del proceso.

RECURSOS	$\longleftrightarrow$	MÁQUINAS , m	M_1, M_2, M_3
TAREAS	$\longleftrightarrow$	*TRABAJOS*, n	J_1, J_2, J_3, J_4
Objetivos	$\longleftrightarrow$	FUNCIONES	

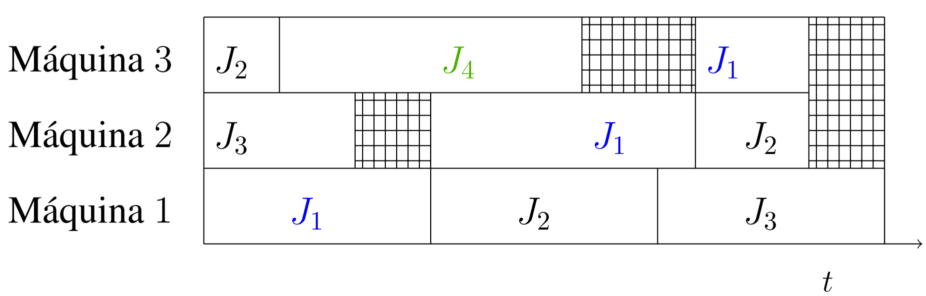


Diagrama de Gantt. Planificación factible

RECURSOS	$\longleftrightarrow$	MÁQUINAS , m
TAREAS	$\longleftrightarrow$	*TRABAJOS*, n
Objetivos	$\longleftrightarrow$	FUNCIONES

Parámetros asociados a cada *TRABAJO* j:

- Operaciones ($\{O_{1j}, \ldots, O_{m_j j}\}$).
- Tiempo de proceso (p_{ij}).
- "Ready o release date" (r_{ij}).
- Fecha de entrega "due date" (d_j).
- Peso o ponderación (w_j).

- Función de coste ($f_j(t)$).

RECURSOS	$\longleftrightarrow$	MÁQUINAS , m
TAREAS	$\longleftrightarrow$	*TRABAJOS*, n
Objetivos	$\longleftrightarrow$	FUNCIONES

Descripción de un problema determinístico $\alpha|\beta|\gamma$

♦ $\alpha = \alpha_1; \alpha_2$, donde $\alpha_1 \in \{\circ, \text{P, Q, R, J, F, O, X}\}$ y $\alpha_2 \in \{\circ\} \cup \mathbb{N}$.

- $\alpha_1 \in \{\circ, \text{P, Q, R}\}$. Cada trabajo consta de una única operación que puede procesarse en cualquier máquina.
  - $\alpha_1 = \circ$: una única máquina; $p_{ij} = p_j$.
  - $\alpha_1 =$P: Máquinas paralelas idénticas.
  - $\alpha_1 =$Q: Máquinas paralelas uniformes.
  - $\alpha_1 =$R: Máquinas paralelas no relacionadas.
- $\alpha_1 =$J: Problema "Job Shop".

$$O_{1j} \to O_{2j} \to \ldots \to O_{m_j j}.$$

  O_{ij} será procesada en la máquina μ_{ij} durante p_{ij} unidades de tiempo. En general, se supone que $\mu_{ij} \neq \mu_{i+1,j}$ para cada $i = 1, \ldots, m_j$.

- $\alpha_1 =$F: Problema "Flow Shop".

  Es un caso especial del problema "Job Shop" tomando $m_j = m$, para cualquier $j = 1, \ldots, n$ y $\mu_{ij} = M_i$, para cualquier $j = 1, \ldots, n$.

- $\alpha_1 =$O: Problema "Open Shop".

  Todos los trabajos tienen el mismo número de operaciones, cada operación se realiza en una máquina específica, pero no existe ninguna relación de precedencia entre las operaciones.

- $\alpha_1 =$X: Problema "Mixed Shop".

  Es una combinación de "Job Shop" y de "Open Shop".

- $\alpha_2 = m$.

- $\alpha_2 = \circ$.

RECURSOS	$\longleftrightarrow$	MÁQUINAS , m
TAREAS	$\longleftrightarrow$	*TRABAJOS*, n
Objetivos	$\longleftrightarrow$	FUNCIONES

Descripción de un problema determinístico $\alpha|\beta|\gamma$

♦ β con β un subconjunto de $\{\beta_1, \beta_2, \beta_3, \beta_4\}$.

- $\beta_1 \in \{pmtn, \circ\}$:
  - β_1 =*pmtn*: cualquier operación puede interrumpirse y reiniciarse más tarde.
  - $\beta_1 = \circ$: No se permite interrupción.
- $\beta_2 \in \{prec, tree, \circ\}$:
  - β_2 =*prec*: Se especifica una relación de precedencia entre los trabajos.
  - β_2 =*tree*: el grafo G es un árbol con raíz, donde de cada nodo sale a lo sumo una rama o llega a él una única rama.
  - $\beta_2 = \circ$: No hay relaciones de precedencia.
- $\beta_3 \in \{r_j, s_{jk}, \circ\}$.
  - $\beta_3 = r_j$.
  - $\beta_3 = s_{jk}$.
  - $\beta_3 = \circ$: Todos los valores son cero.
- $\beta_4 \in \{p_j = 1, p_{ij} = 1, \circ\}$.
  - $\beta_4 = p_j = 1$: Cada trabajo necesita un tiempo de proceso de una unidad.
  - $\beta_4 = p_{ij} = 1$: Cada operación requiere una unidad de procesamiento.
  - $\beta_4 = \circ$: Los valores de p_j o de p_{ij} pueden ser cualquier entero no negativo.

RECURSOS	$\longleftrightarrow$	MÁQUINAS , m
TAREAS	$\longleftrightarrow$	*TRABAJOS*, n
Objetivos	$\longleftrightarrow$	FUNCIONES

Descripción de un problema determinístico $\alpha|\beta|\gamma$

♦ γ con $\gamma \in \{f_{max}, \sum f_j\}$.

- **Makespan** (alcance o duración total)

$$\mathbf{C_{max}} = \max\{C_1, \ldots, C_n\}$$

  C_j es el tiempo total del trabajo j en el sistema.

- **Máximo retraso, $\mathbf{L_{max}}$.**

$$\mathbf{L_{max}} = \max\{L_1, \ldots, L_n\}$$

  C_j es el tiempo total del trabajo j en el sistema.

  $L_j = C_j - d_j$ es el retraso ("lateness") que sufre un trabajo respecto a su fecha de entrega.

- **Tiempo total de finalización ponderado**

$$\sum \mathbf{w_j C_j}$$

C_j es el tiempo total del trabajo j en el sistema.

- **Tardanza total ponderada**,

$$\sum \mathbf{w_j T_j}$$

C_j es el tiempo total del trabajo j en el sistema.

$T_j = \max\{C_j - d_j, 0\} = \max\{L_j, 0\}$ es la tardanza ("tardiness") del trabajo j respecto a su fecha de entrega.

- **Número total ponderado de trabajos con retraso**,

$$\sum \mathbf{w_j U_j}$$

C_j es el tiempo total del trabajo j en el sistema.

$$U_j = \begin{cases} 1 & \text{si} \quad C_j > d_j \\ 0 & \text{en el resto} \end{cases}$$

RECURSOS $\longleftrightarrow$ MÁQUINAS , m

TAREAS $\longleftrightarrow$ *TRABAJOS*, n

Objetivos $\longleftrightarrow$ FUNCIONES

Descripción de un problema determinístico $\alpha|\beta|\gamma$

- $Pm|r_j, M_j|\sum w_j T_j$. Asignación de puertas de embarque.
- $1|s_{jk}|C_{max}$. Problema del viajante.
- $P\infty|prec|C_{max}$. Planificación de proyectos.
- $Jm||C_{max}$. Job Shop con m máquinas.

$T_j = \max\{C_j - d_j, 0\}$

$C_{max} = \max\{C_1, \ldots, C_n\}$ Calendario sin retraso. Ninguna máquina está libre si hay alguna operación lista para ser procesada.

Calendario activo. No es posible construir otro intercambiando el orden de los trabajos donde alguna operación finalice antes y ninguna finalice más tarde.

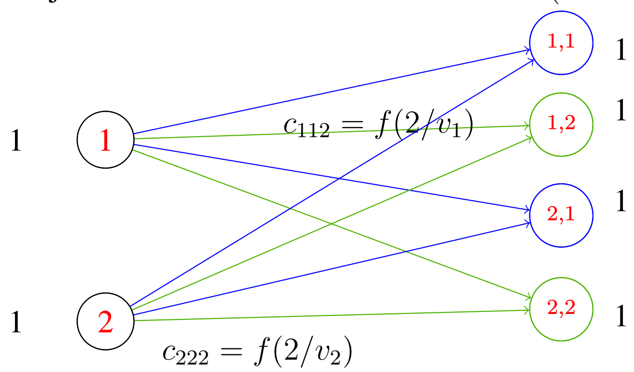
Calendario semi-activo. Ninguna operación finaliza antes sin cambiar el orden de procesamiento en alguna máquina.

2 Resolución de problemas de planificación.

2.1 Determinísticos

$Qm|p_j = 1|\sum f(C_j)$. Modelo del transporte asociado.

Trabajos n Posiciones $(m \times n)$



$x_{ijk} = 1$ el trabajo j es el trabajo k en la máquina i; $x_{ijk} = 0$

$$c_{ijk} = h_j(C_j) = f_j(k/v_i).$$

$\mathbf{Qm|p_j = 1|\sum f(C_j)}$ ↔Problema de asignación ponderada

N=conjunto de trabajos M=conjunto de posiciones (i, k)

$(i, 1)$ posición del último trabajo en la máquina i.
$(i, 2)$ posición del penúltimo trabajo en la máquina i

$\vdots$

(i, k) posición del trabajo $k-$ésimo en la máquina i empezando por el final.

$x_{ijk} = 1$ el trabajo j es el trabajo k, empezando por el final, en la máquina i
$x_{ijk} = 0$, en otro caso. $w_{ijk} = kp_{ij}$.

$\mathbf{1||\sum w_j C_j}$

Trabajos	J_1	$\ldots$	J_n
w_j	w_1	$\ldots$	w_n
p_j	p_1	$\ldots$	p_n
u_j	u_1	$\ldots$	u_n

Indice de urgencia: $u_j = w_j/p_j$, $j = 1, \ldots, n$

Calendario óptimo (regla WSPT):

el trabajo j se ejecuta, sin interrupción, antes que el trabajo k si y sólo si

$$u_j \geq u_k$$

Este problema puede resolverse en tiempo polinomial.

$\mathbf{1|r_j|L_{max}}$

$L_j = C_j - d_j$, $j = 1, \ldots, n$

$\mathbf{L_{max}} = \max\{L_1, \ldots, L_n\}$

fuertemente NP-hard!!!!

$\mathbf{1}||\sum \mathbf{w_j U_j}$

$$U_j = \begin{cases} 1 & \text{si} \quad C_j > d_j \\ 0 & \text{en el resto} \end{cases}$$

NP-hard!!!!

2.2 Estocásticos

▶X_{ij}: Tiempo de proceso del trabajo j en la máquina i.

▶$1/\lambda_{ij}$: valor esperado o media de la variable X_{ij}.

▶R_j: Instante en que está listo el trabajo j para ser procesado.

▶D_j: Fecha de entrega del trabajo j.

▶w_j: la ponderación del trabajo j.

Tasa de finalización: <u>V. a. continua</u>

$c(t) = \frac{f(t)}{1-F(t)}, t > 0$ <u>V. a. discreta</u>

$c(t) = \frac{P(X=t)}{P(X \geq t)}, t = 0, 1, 2, \ldots$

ICR: $c(t)$ es creciente.

DCR: $c(t)$ es decreciente.

♦ **Régimen de lista estática sin interrupción (Nonpreemptive Static List Policy)**

El decisor ordena los trabajos en el instante cero de acuerdo con una lista de prioridades. Esta lista de prioridades se mantiene durante el proceso.

♦ **Régimen de lista estática con interrupción (Preemptive Static List Policy)** Un trabajo está preparado en un cierto instante. El trabajo que está siendo procesado posee una prioridad más baja, entonces éste trabajo deja de ser procesado y se sustituye por el que está listo con prioridad más alta.

♦ **Régimen dinámico sin interrupción (Nonpreemptive Dynamic Policy)** Cada vez que una máquina está libre el decisor puede seleccionar el siguiente trabajo que va a ser procesado. No está permitido interrumpir la ejecución de un trabajo para iniciar el procesado de otro trabajo.

♦ **Régimen dinámico con interrupción (Preemptive Dynamic Policy)**

En cada instante el decisor puede escoger el trabajo que va a procesarse en cada máquina. Para ello tendrá en cuenta toda la información disponible y también que es posible la interrupción.

$\mathbf{1}|\,|\sum \mathbf{w_j C_j}$

▶X_j v.a. con $E[X_j]$ finita.

▶Función objetivo: $E[\sum w_j C_j]$

Calendario óptimo bajo lista estática sin interrupción y lista dinámica sin interrupción:

el trabajo j se ejecuta antes que el trabajo k si y sólo si

$$w_j/E[X_j] \geq w_k/E[X_k]$$

Calendario óptimo bajo lista dinámica con interrupción:
WSEPT sin interrupción
si todas las distribuciones X_j son ICR.

$\mathbf{1| \,| \sum w_j C_j}$

▶X_j v.a. con $Exp(\lambda_j)$.

▶R_j v.a. con cualquier distribución conjunta.

▶Función objetivo: $E[\sum w_j C_j]$

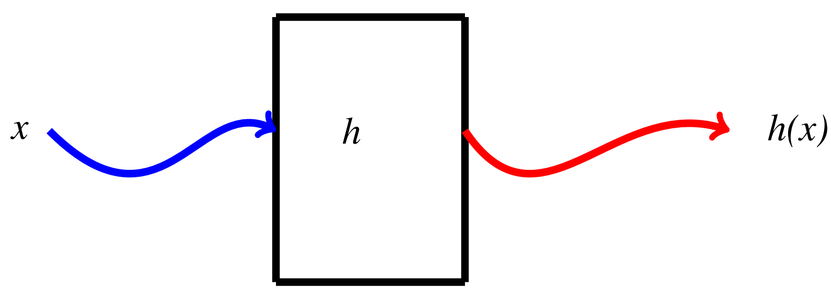
Calendario óptimo: WSEPT con interrupción

La versión determinística es *NP-hard*.

Problema	Determínistico	Estocástico
$\mathbf{Pm|p_j = 1}, \text{tree}\mathbf{|C_{max}}$	Regla CP	*???*
$\mathbf{O2| \,|C_{max}}$	Regla LAPT	*???*

2.3 Complejidad

Representación de un problema computacional.



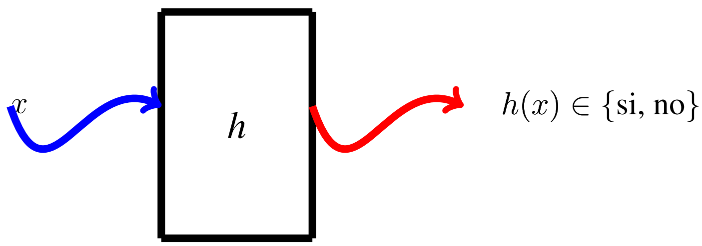
Complejidad del algoritmo M: $T(n) = \max\{time_h(x) \mid |x| = n\}$,

$time_h(x)$: número de etapas realizadas por h

$|x|$: longitud de x respecto a un sistema de codificación.

Resoluble de modo polinomial: $T(|x|) \in O(|x|^k)$.

Problema de decisión



$L \in \mathcal{P}$ si y sólo si $T(n) \in O(n^k)$

$L \in \mathcal{NP}$ si y sólo si para cada x existe y con $|y| \leq q(|x|)$
tal que y es un *certificado* para la respuesta “si” y se puede
comprobar en tiempo polinomial la certificación

$1||\sum w_jU_j$, $\qquad \mathcal{P} \subset \mathcal{NP}$

$$U_j = \begin{cases} 1 & \text{si} \quad C_j > d_j \\ 0 & \text{en el resto} \end{cases}$$

- Problema de optimización.

  Caso: para cada trabajo $j = 1, \ldots, n$, (p_j, d_j, w_j).

  Respuesta: una planificación de los trabajos.

- Problema de decisión.

  Caso: para cada trabajo $j = 1, \ldots, n$, (p_j, d_j, w_j) y un entero positivo K.

  Respuesta: “si” cuando el número ponderado de trabajos con retraso es menor o igual a K.

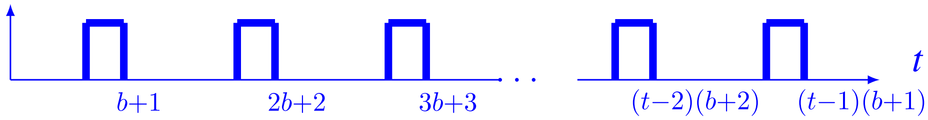
CERTIFICADO: Calendario S con $\sum w_jU_{j|S} \leq K$
L_1, L_2, decimos que $L_1 \propto L_2$, si existe una función g tal que

i) g se puede calcular en tiempo polinomial.

ii) $x \in L_1$ si y sólo si $g(x) \in L_2$ para cualquier x.

iii) x lleva a la respuesta ”si” en L_1 si y sólo si $g(x)$ conduce a la respuesta ”si” en L_2.

$\mathbf{1||\sum w_jT_j}$ es fuertemente NP-hard. ($T_j{=}\max\{C_j - d_j, 0\}$)
3-PARTICIÓN reduce a $1||\sum w_jT_j$
$t, a_1, a_2, \ldots, a_{3t}, b \in \mathbb{N}$, con $\frac{b}{4} < a_j < \frac{b}{2}$, $j = 1, \ldots, 3t$, y $\sum_{j=1}^{3t} a_j = tb$,
¿$A_1, \ldots, A_t$, de $\{1, \ldots, 3t\}$, tales que $A_r \cap A_s = \emptyset$ con $r \neq s$ y $\sum_{j\in A_i} a_j = b$ para cualquier $i = 1, \ldots, t$?



$$n=4t-1,\quad d_j=0,\qquad\qquad p_j=a_j,\quad w_j=a_j,\quad j=1,\ldots,3t$$
$$d_j=(j-3t)(b+1),\quad p_j=1,\quad w_j=2,\quad j=3t+1,\ldots,4t-1.$$

$$z=\sum_{1\le j\le k\le 3t} a_j a_k+\tfrac{1}{2}(t-1)tb$$

2.4 Algoritmos

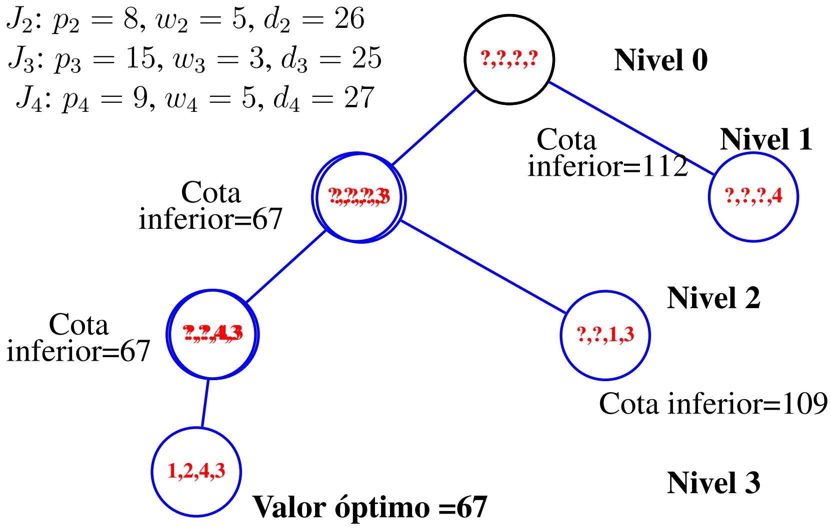
¿Cómo encontrar soluciones en $\mathcal{NP}$?

- Búsqueda local
- ”Simulated annealing”
- Búsqueda Tabú
- Ramificación y acotación (”branch and bound”)
- Algoritmos genéticos
- Basados en mercados y agentes

  ⋮

$\mathbf{1}||\sum \mathbf{w_j T_j}$ $\qquad d_j\le d_k,\ p_j\le p_k,\ w_j\ge w_k \Rightarrow j$ antes k

J_1: $p_1=12$, $w_1=4$, $d_1=16$
J_2: $p_2=8$, $w_2=5$, $d_2=26$
J_3: $p_3=15$, $w_3=3$, $d_3=25$
J_4: $p_4=9$, $w_4=5$, $d_4=27$



$\mathbf{Fm}||\mathbf{C_{max}}$ es NP-hard.

Estructura: una secuencia de trabajos en el problema.

Población inicial:

- $m-1$ calendarios producidos por el método Campbell, Dudek y Smith (1970).
- Calendario producido por el método de Dannenbring (1977).
- El resto de los miembros se obtienen mediante mutaciones.

Función de ajuste:

- Se calcula, para cada calendario, el alcance o makespan.
- Se calcula el máximo de los alcances, **CMAX**.
- La función de ajuste es la desviación del alcance de cada miembro de la población respecto a **CMAX**.

$\mathbf{Fm||C_{max}}$ es NP-hard.
Operador genético: cruce (operador PMX de Golberg)

A	2	5	4	1	6	9	8	3	7
B	3	7	9	2	5	4	1	6	8

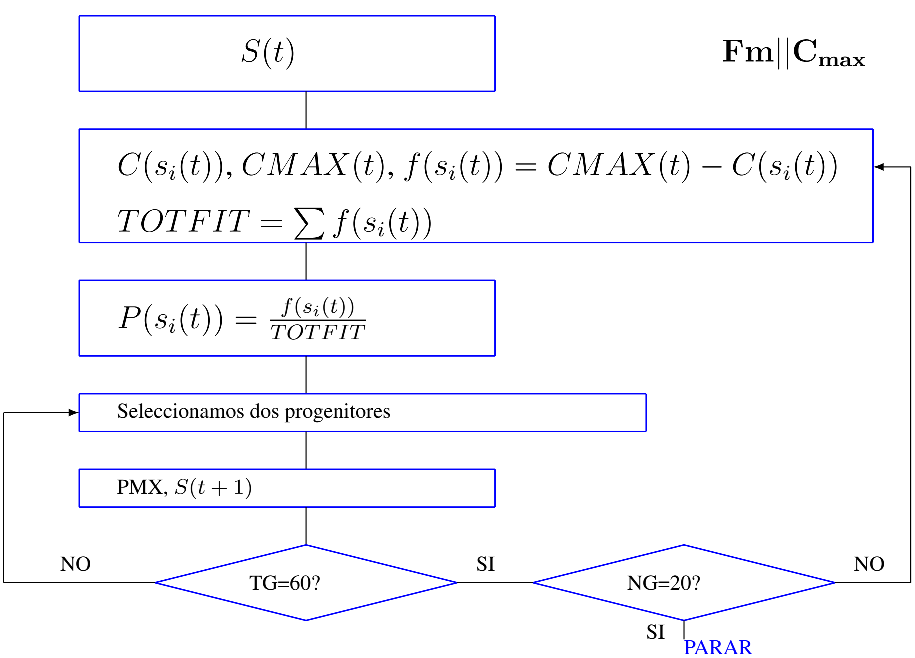
Seleccionamos aleatoriamente las posiciones $[3, 5]$.

A	2	5	9	2	5	9	8	3	7
B	3	7	4	1	6	4	1	6	8

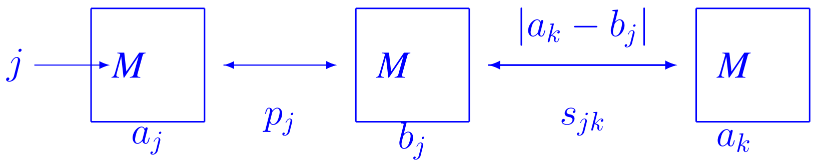
A	*2*	*5*	9	2	5	*9*	8	3	7
B	3	7	4	1	6	*4*	*1*	*6*	8

A	*1*	*6*	9	2	5	*4*	8	3	7
B	3	7	4	1	6	*9*	*2*	*5*	8

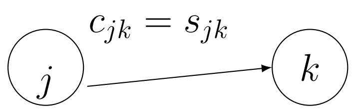
Criterio de finalización: Número de generaciones.



$\mathbf{1|s_{jk}|C_{max}}$ es fuertemente NP-hard.
Caso particular: Trabajo $j \longrightarrow (a_j, b_j)$ y $s_{jk} = |a_k - b_j|$.



TSP:



existe algoritmo en tiempo polinomial

3 Planificación en la práctica

Reglas de prioridad o "dispatching rules"

- ♦ Según su dependencia respecto al instante en que se aplican.
  - ▶Estáticas: WSPT, EDD, ...
  - ▶Dinámicas: MS

    En el instante t, $\max\{d_j - p_j - t, 0\}$.
- ♦ Según la información que utilizan.
  - ▶Locales: WSPT, EDD, MS, ...
  - ▶Globales: MS

$\mathbf{1}| \,| \sum \mathbf{w_j T_j}$

Regla ATC ("Aparent Tardiness Cost")

$$I_j(t) = \frac{w_j}{p_j} exp(-\frac{\max\{d_j - p_j - t, 0\}}{K\bar{p}})$$

▶$\bar{p}$: valor medio de los tiempos de proceso de los trabajos pendientes
▶K: factor de escala determinado empíricamente.

j se realiza antes que k si y sólo si $I_j(t) \geq I_k(t)$.

K grande	$\Longrightarrow$	ATC=WSPT.	
K pequeño	$\Longrightarrow$	ATC=MS para trabajos retrasados.	K?
		ATC=WSPT para trabajos sin retraso.	

▶Prontitud de las fechas de entrega ("due date tightness"):

$$\tau = 1 - \frac{\sum d_j}{nC_{max}}.$$

▶Rango de las fechas de entrega, R:

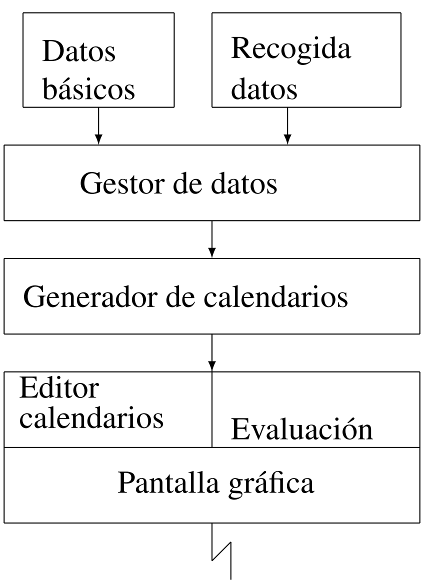
$$R = \frac{d_{max} - d_{min}}{C_{max}}.$$

Modelos y Aplicaciones.

- Estáticos $\leftrightarrow$ Dinámica.
- Funcionamiento de las máquinas más complejo.
- Existencia de preferencias.
- Disponibilidad de las máquinas.
- Linealidad de las funciones de penalización.
- Multiplicidad de objetivos.
- Asignación de turnos.
- Distribución del tiempo de proceso no exponencial.
- Correlación entre tiempos de proceso.
- Aprendizaje y desgaste.

¡¡ LAS EMPRESAS DISEÑAN E IMPLEMENTAN SISTEMAS DE PLANIFICACIÓN !!

. Arquitectura de un sistema

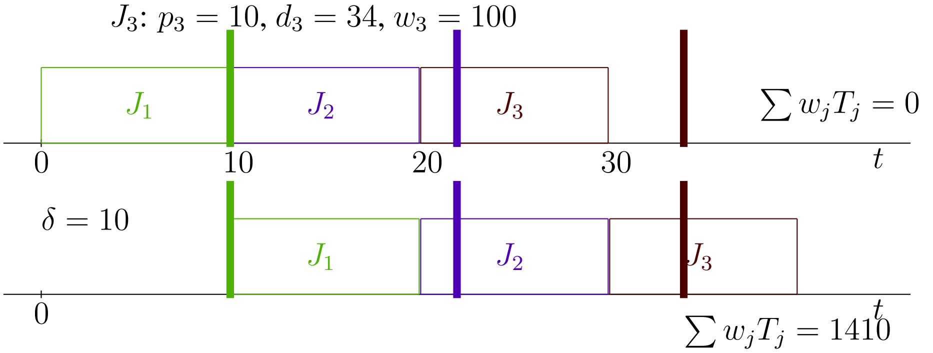


. Calendarios robustos

Ocurrencia de hechos inesperados

$\mathbf{1 | \,| \sum w_j T_j}$

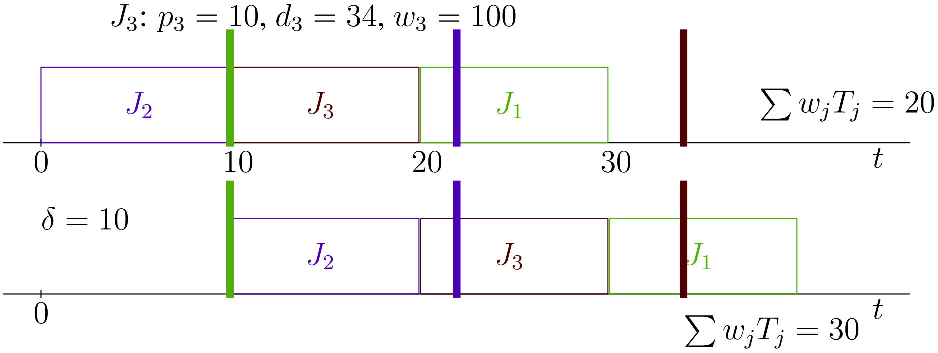
J_1: $p_1 = 10$, $d_1 = 10$, $w_1 = 1$
J_2: $p_2 = 10$, $d_2 = 22$, $w_2 = 100$
J_3: $p_3 = 10$, $d_3 = 34$, $w_3 = 100$



Ocurrencia de hechos inesperados

$\mathbf{1 | \,| \sum w_j T_j}$

J_1: $p_1 = 10$, $d_1 = 10$, $w_1 = 1$
J_2: $p_2 = 10$, $d_2 = 22$, $w_2 = 100$
J_3: $p_3 = 10$, $d_3 = 34$, $w_3 = 100$



Ocurrencia de hechos inesperados

$\mathbf{1 | \,| \sum w_j T_j}$

J_1: $p_1 = 10$, $d_1 = 10$, $w_1 = 1$
J_2: $p_2 = 10$, $d_2 = 22$, $w_2 = 100$
J_3: $p_3 = 10$, $d_3 = 34$, $w_3 = 100$

	Sin retraso	$\delta = 0$
Orden $1, 2, 3$	$\sum w_j T_j = 0$	$\sum w_j T_j = 1410$
Orden $2, 3, 1$	$\sum w_j T_j = 20$	$\sum w_j T_j = 30$

Tiempo de finalización: C_j, Función objetivo: Z

Retraso $\delta \longrightarrow$ Tiempo de finalización: $C_j(\delta)$
Función objetivo: $Z'(\delta)$

Medidas de robustez:

$$\frac{Z'(\delta) - Z}{\delta}$$

$$\sum_{\delta=0}^{\infty}(Z'(\delta) - Z)P(\Delta = \delta), \qquad \int_{\delta=0}^{\infty}(Z'(\delta) - Z)P(\Delta = \delta)$$

$$\mathcal{R}(\mathcal{S}) = \frac{\sum_{j=1}^{n} w_j(d_j - C_j)}{\sum_{j=1}^{n} w_j d_j}$$

. Mecanismos de aprendizaje

♦ Aprendizaje de memoria

♦ Basado en el estudio de casos

♦ Inducción y redes neuronales

♦ Sistemas de clasificación

¿Qué queda por hacer?

♦ Métodos de búsqueda de soluciones.

♦ Secuenciación ”on-line”.

♦ El campo de la resecuenciación.

♦ Combinar características determinísticas y estocásticas.

♦ Evaluación de los heurísticos.

♦ Combinar programación en máquinas con otros aspectos como personal, control del inventario, mantenimiento, etc.

♦ Secuenciación distribuida.

McKay, K., Pinedo, M. y Webster, S. (2001) A practice-focused agenda for production scheduling research. *Production and Operations Management* 11, 249-258.

4 Bibliografía

Brucker, P. (2004) Scheduling algorithms. Springer.
Pinedo, M. (2002) Scheduling. Theory, Algorithms, and Systems. Prentice-Hall.
⋮

European Journal of Operational Research.
IEEE Transactions on Robotics and Automation.
Interfaces.
Journal of Scheduling.
Mathematical Programming.
Managenement Science.
Operations Research.

$\vdots$

http://www.mathematik.uni-osnabrueck.de/research/OR/class/
http://www.lix.polytechnique.fr/ durr/query/
http://www.stern.nyu.edu/om/software/lekin/