Tema 5

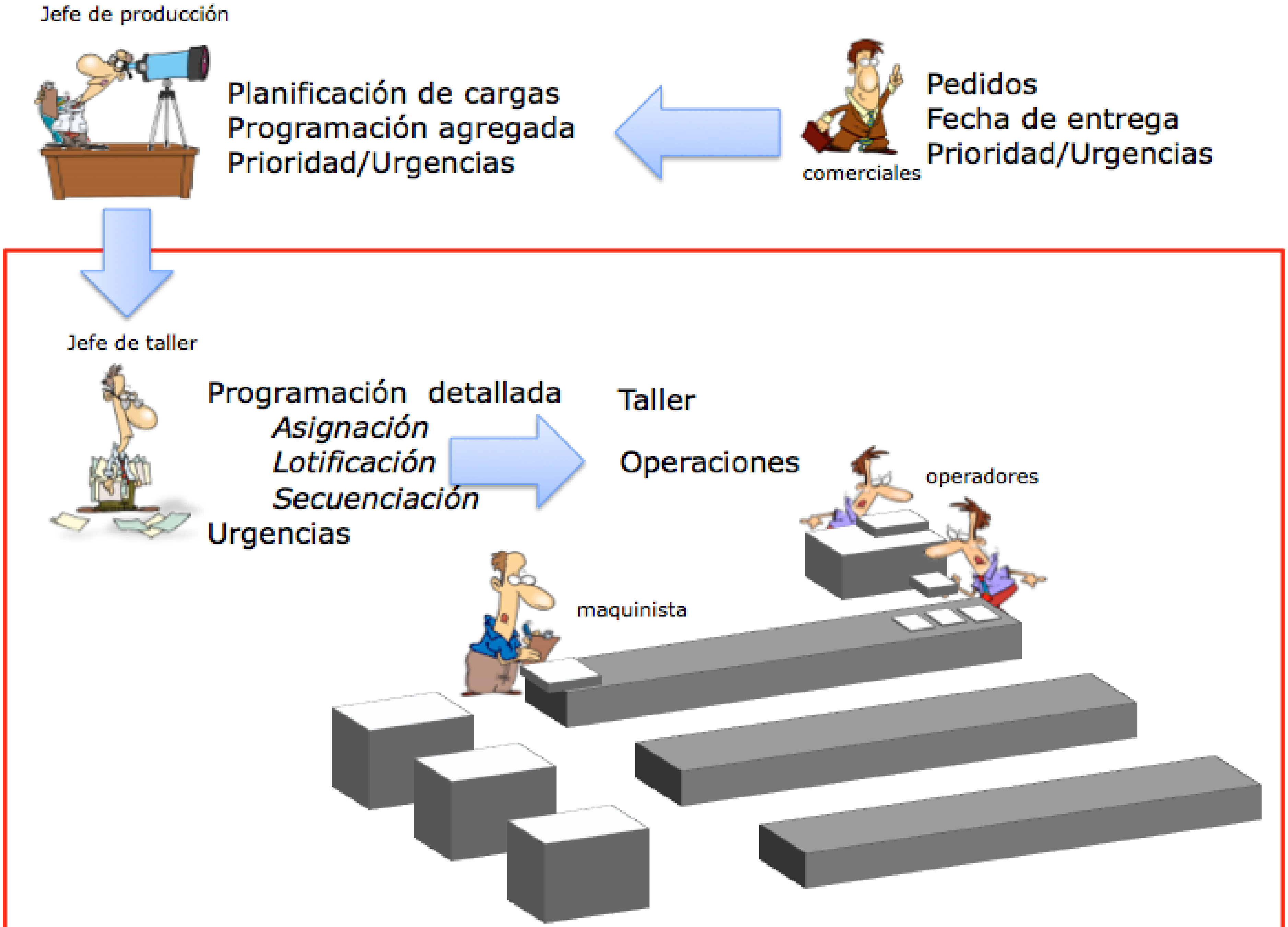
Planificación y Asignación Optimizada de Recursos

CONTENTS

Scheduling

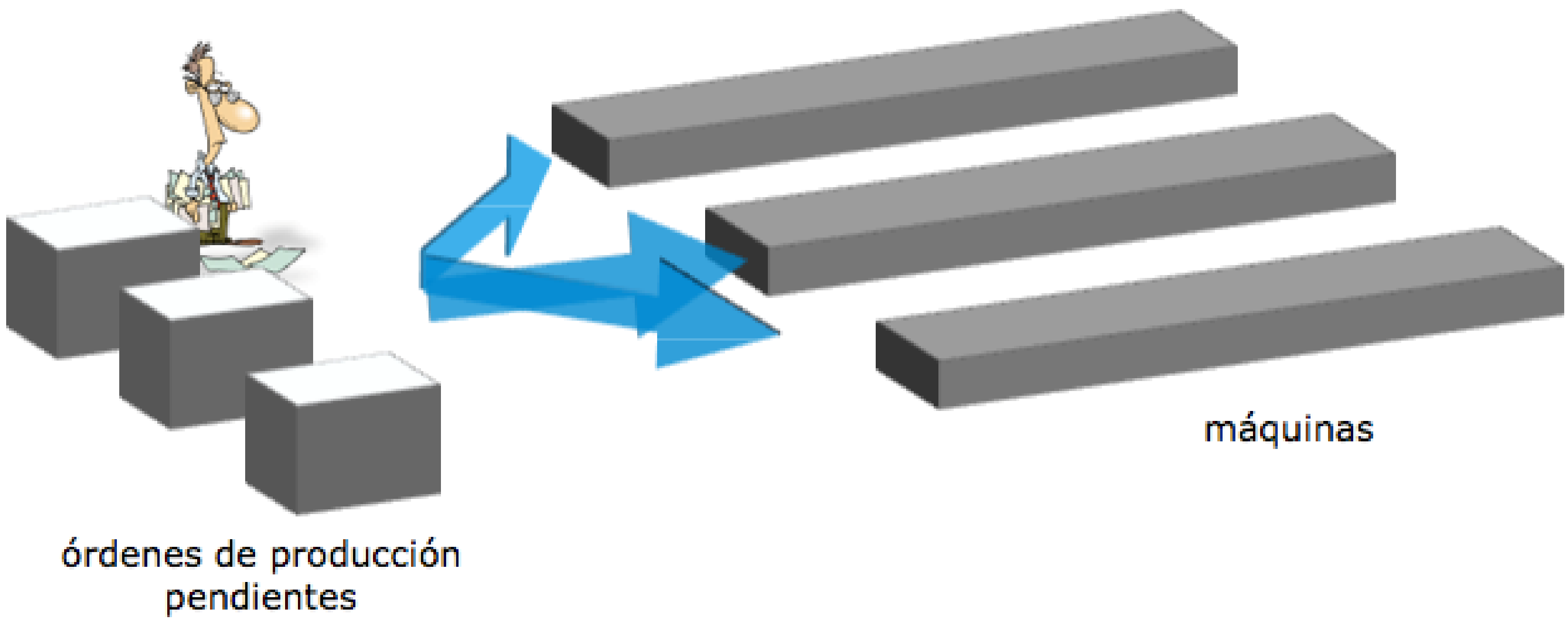
- Asignación de recursos a tareas a lo largo del tiempo

- Proceso de toma de decisiones, para optimizar un objetivo (o múltiples objetivos)


Jefe de producción
Planificación de cargas
Programación agregada
Prioridad/Urgencias
Pedidos
Fecha de entrega
Prioridad/Urgencias
comerciales
Jefe de taller
Programación detallada
Asignación
Lotificación
Secuenciación
Urgencias
Taller
Operaciones
operadores
maquinista

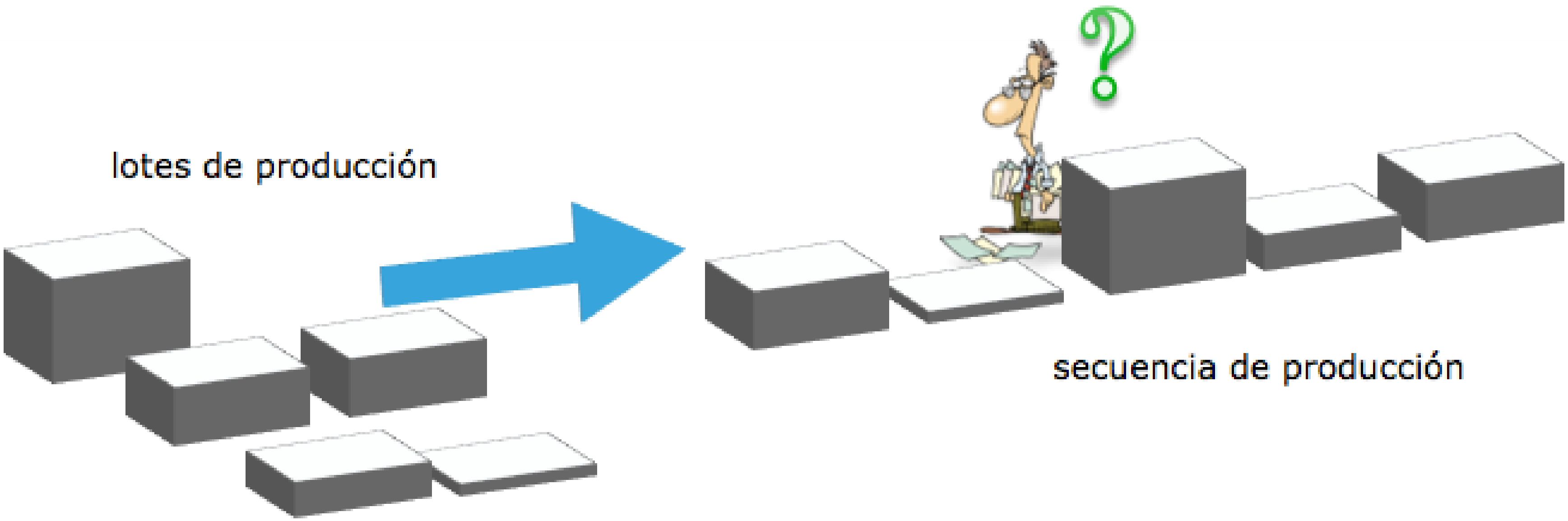
Problema de asignación

En el problema de asignación, el planificador intentará determinar la correspondencia óptima entre las órdenes pendientes de fabricación, y los recursos que serán utilizados para su procesamiento (máquinas y operarios).



Problema de secuenciación

La secuenciación consiste en ordenar según un criterio de prioridad (o temporizar) los lotes a fabricar. Los lotes secuenciados en primer lugar serán acabados antes, y podrán ser entregados antes al cliente



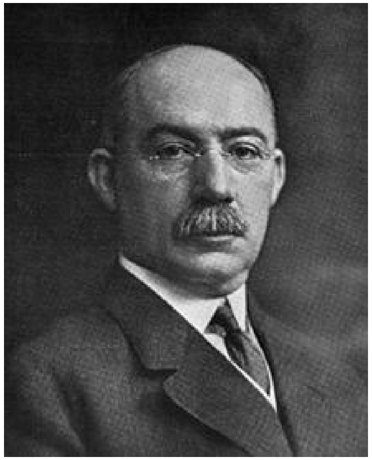
Elementos del problema

- **Recursos:**
  - Máquinas en un taller
  - Pistas en un aeropuerto
  - Procesadores en un ordenador

- **Tareas:**
  - Operaciones en una pieza
  - Aterrizajes de un avión
  - Programas a ejecutar

- **Objetivos:**
  - Cumplir las fechas de entrega
  - Minimizar el tiempo total de completación

Diagramas de Gantt



THE GANTT CHART

A WORKING TOOL OF MANAGEMENT

By

WALLACE CLARK

WITH APPENDICES

by

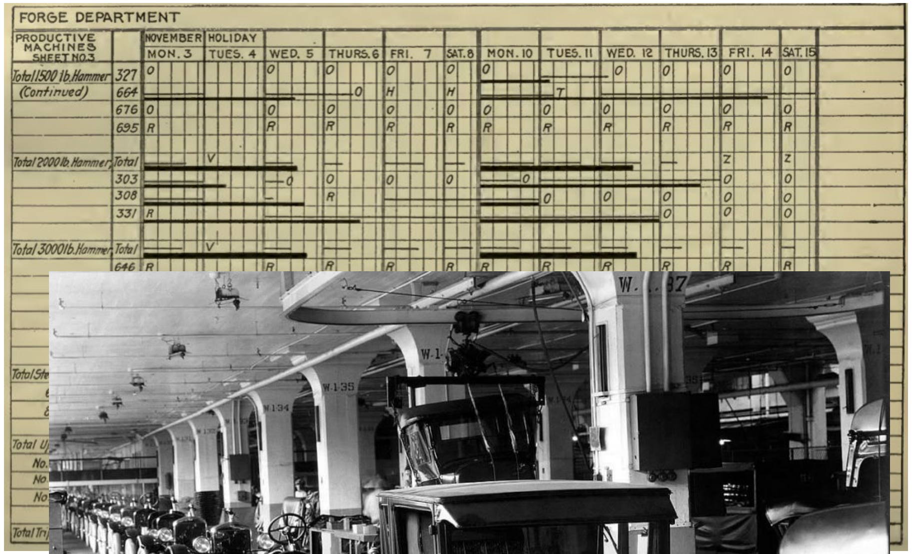
WALTER N. POLAKOV

and

FRANK W. TRABOLD

NEW YORK

THE RONALD PRESS COMPANY

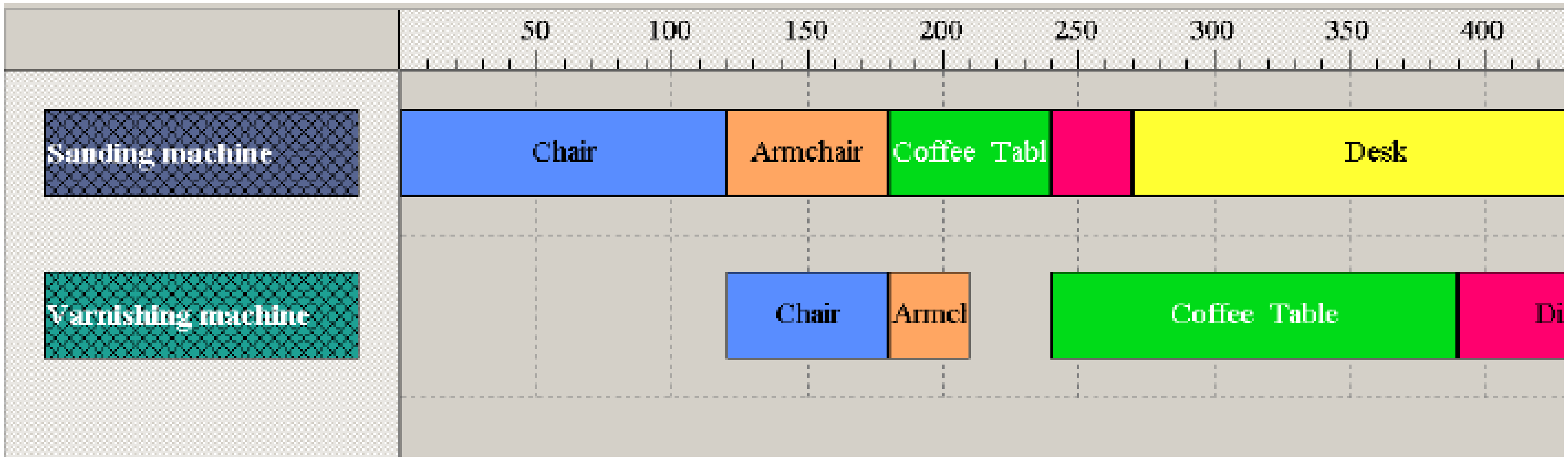


Henry Lawrence Gantt

1861-1919)

Diagramas de Gantt

Un diagrama de Gantt es un diagrama de barras horizontales, en el que el eje x representa el tiempo y el eje y las máquinas. Las operaciones de la misma tarea se indican mediante el color o la textura



Tipos de problemas de Scheduling

- **Secuenciación de tareas en máquinas**
  (Machine scheduling)

- **Secuenciación de proyectos**
  (Project Scheduling)

- **Turnos de trabajadores**
  (Labour Scheduling)

- **Problemas de horarios**
  (Timetabling)

Organización de la producción en una fábrica de bolsas de papel

- Se producen bolsas de papel para cemento, carbón, comida para perros…
- La **materia prima** son rollos de papel
- El proceso de producción tiene **3 fases**:
  - Impresión del logotipo
  - Pegado los lados
  - Cosido de los extremos
- En cada fase hay varias **máquinas**, con diferentes características:
  - Velocidad
  - Tipo de bolsas que pueden procesar
  - Colores que puede utilizar
- Cada **pedido** comprende cierto número de bolsas de cada tipo, con su fecha de entrega

Organización de la producción en una fábrica de bolsas de papel

- Entregar un pedido después de su **fecha de entrega** supone una penalización

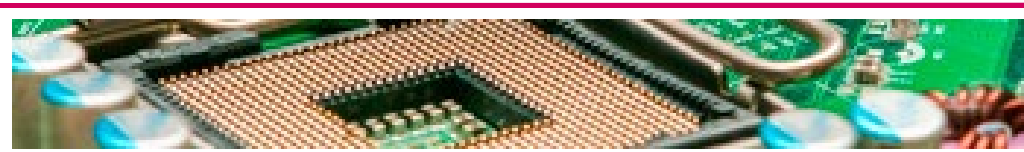
↓

Un objetivo es minimizar la suma ponderada de los retrasos

- Cada vez que una máquina cambia de tipo de bolsa, hay un **tiempo de preparación**, que depende de los tipos de bolsa (tamaño, colores,…)

↓

Otro objetivo es minimizar los tiempos de preparación



Asignación de puertas de embarque en un aeropuerto

- En un gran aeropuerto hay decenas de **puertas de embarque** y centenares de **vuelos de salida y llegada** cada día
- No todas las puertas son iguales, ni tampoco lo son los aviones
- Los aviones siguen un calendario conocido, pero sujeto a incertidumbre (clima, retrasos en otros aeropuertos,...)
- La hora de salida de un vuelo puede ser vista como su fecha de entrega y las compañías aéreas se miden por su cumplimiento
- Si se sabe que un avión no puede aterrizar en destino a su hora prevista, no despega y continúa ocupando la puerta de embarque
- Hay que asignar puertas de embarque a los vuelos, minimizando el personal necesario y los retrasos.

◆**Puerta = Recurso** ◆**Vuelo = Tarea** ◆**Salida= Completación**

¿Por qué necesitamos programar la producción?

- Ciclos de vida de los productos cada vez más cortos
- Creciente variedad de productos
- Producción bajo demanda
- Respuesta rápida a los clientes
- Operaciones cada vez más complejas han de secuenciarse en tiempos cada vez más cortos, con menos margen de error!

Niveles de planificación

ERP

▪ Planificación a largo plazo	2-5 años
▪ Planificación a medio plazo	1-2 años
▪ Planificación a corto plazo	3-6 meses
▪ **Secuenciación (scheduling)**	2-6 semanas
▪ Resecuenciación	1-3 días (ya!)

Características importantes de las técnicas de resolución

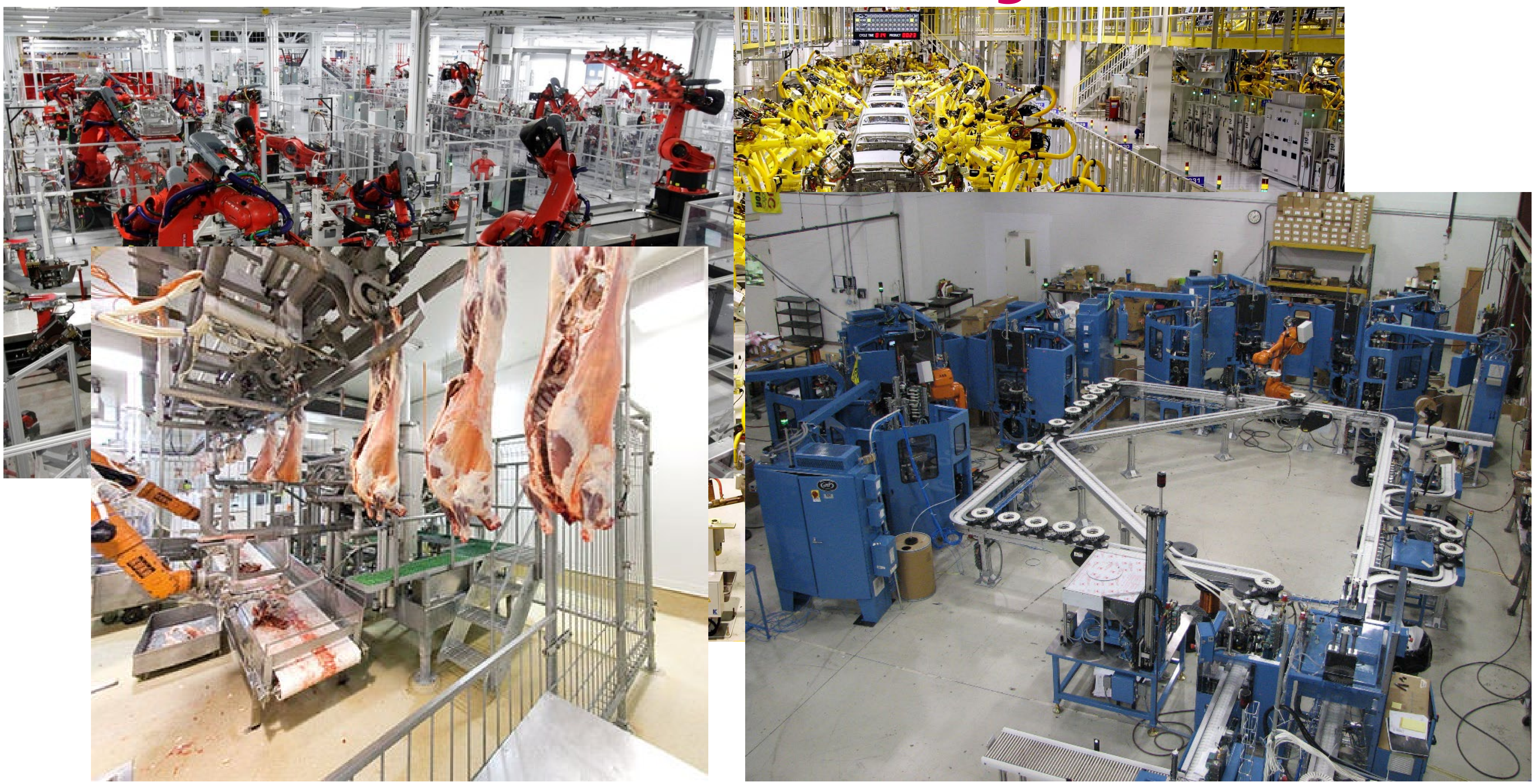
- **Calidad** de las soluciones obtenidas
  (¿cómo de cerca de la solución óptima?)

- **Tiempo** de CPU requerido

- **Simplicidad** de desarrollo e implementación
  (¿cuánto tiempo se necesita para programarlo,
  probarlo, ajustarlo, modificarlo?)

- **Costes** de implementación
  (¿se necesita un código comercial de Programación Lineal?)

Clasificación de problemas de Machine Scheduling



Clasificación de problemas de Machine Scheduling

- Clasificación: $\alpha \mid \beta \mid \gamma$

Máquinas

Tareas

Objetivos

α : Características de las máquinas

Sistemas de una etapa:

- Una máquina (m=1): cada job se ha de procesar en esa máquina
- Máquinas en paralelo: cada job se procesa en una de las máquinas del conjunto $\{M_1, M_2, \ldots, M_m\}$

Sistemas multi-etapa:

- Cada job se ha de procesar en cada una de las máquinas del conjunto $\{M_1, M_2, \ldots, M_m\}$
- Cada máquina sólo puede procesar un job en cada momento.
- Cada job se procesa en una máquina en cada momento.

Sistemas de una etapa

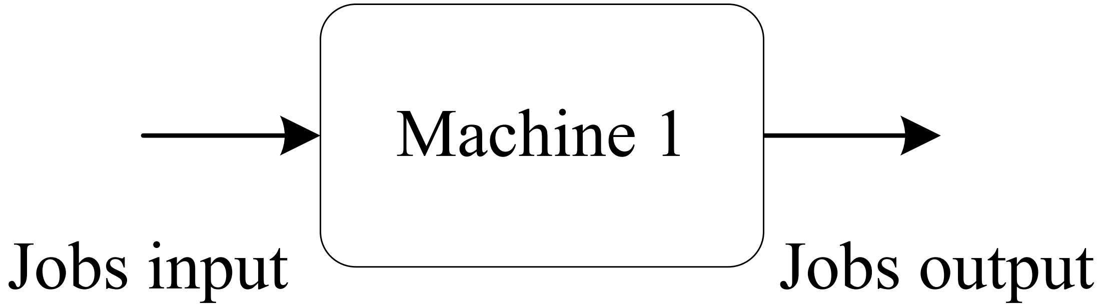
- **Una máquina:** **1**

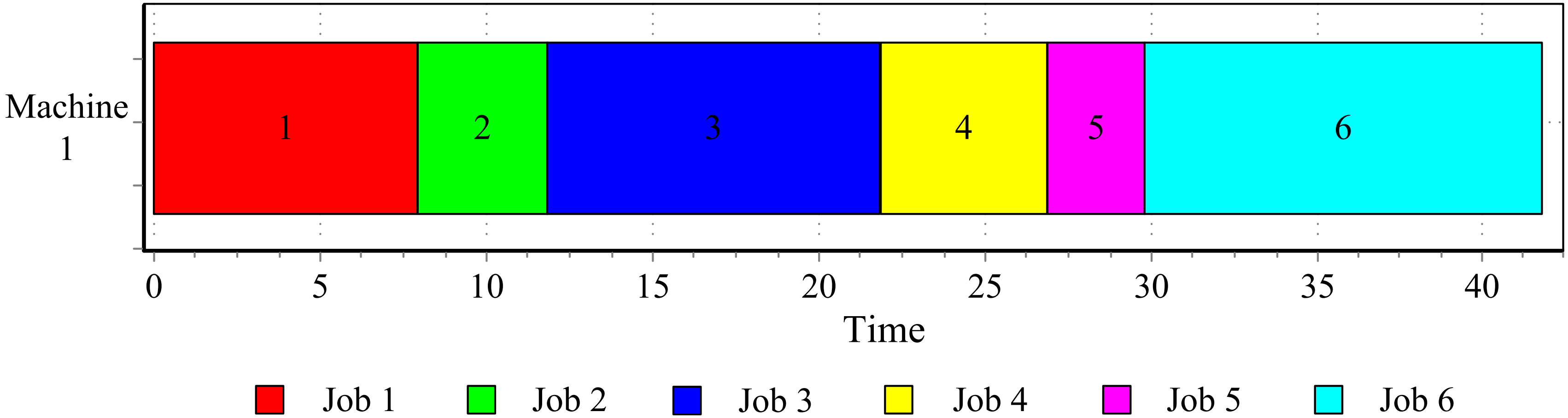
- **Máquinas en paralelo**
  - Máquinas idénticas: **P**
    (mismo tiempo de proceso de todas las tareas
    en todas las máquinas)

  - Máquinas con diferentes velocidades: **Q**
    (pero velocidad uniforme para todas las tareas)

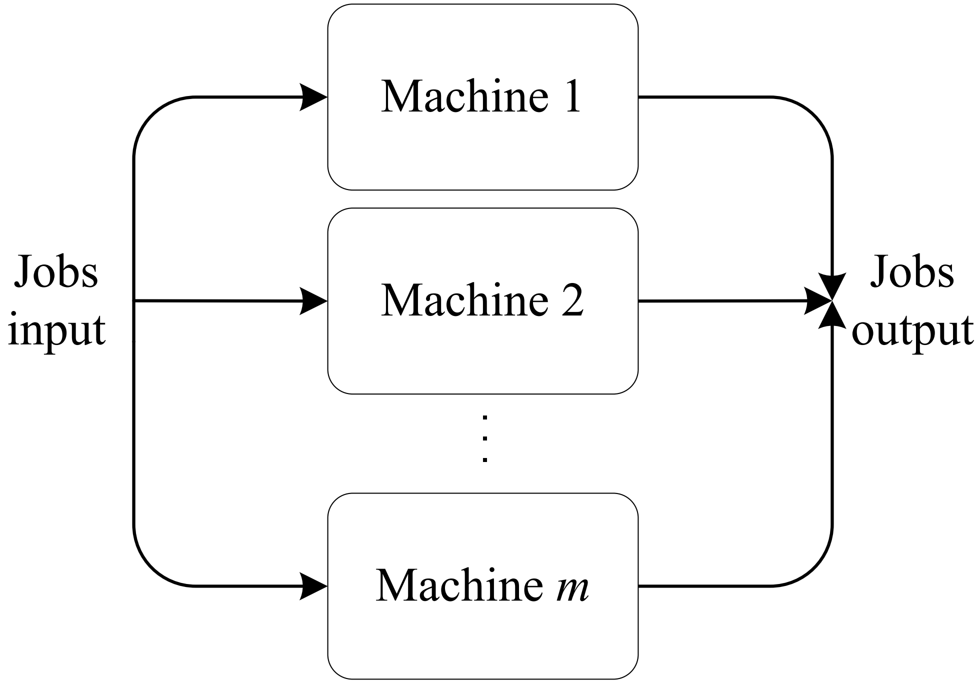
  - Máquinas con velocidad dependiente de la tarea **R**

Sistemas de una etapa

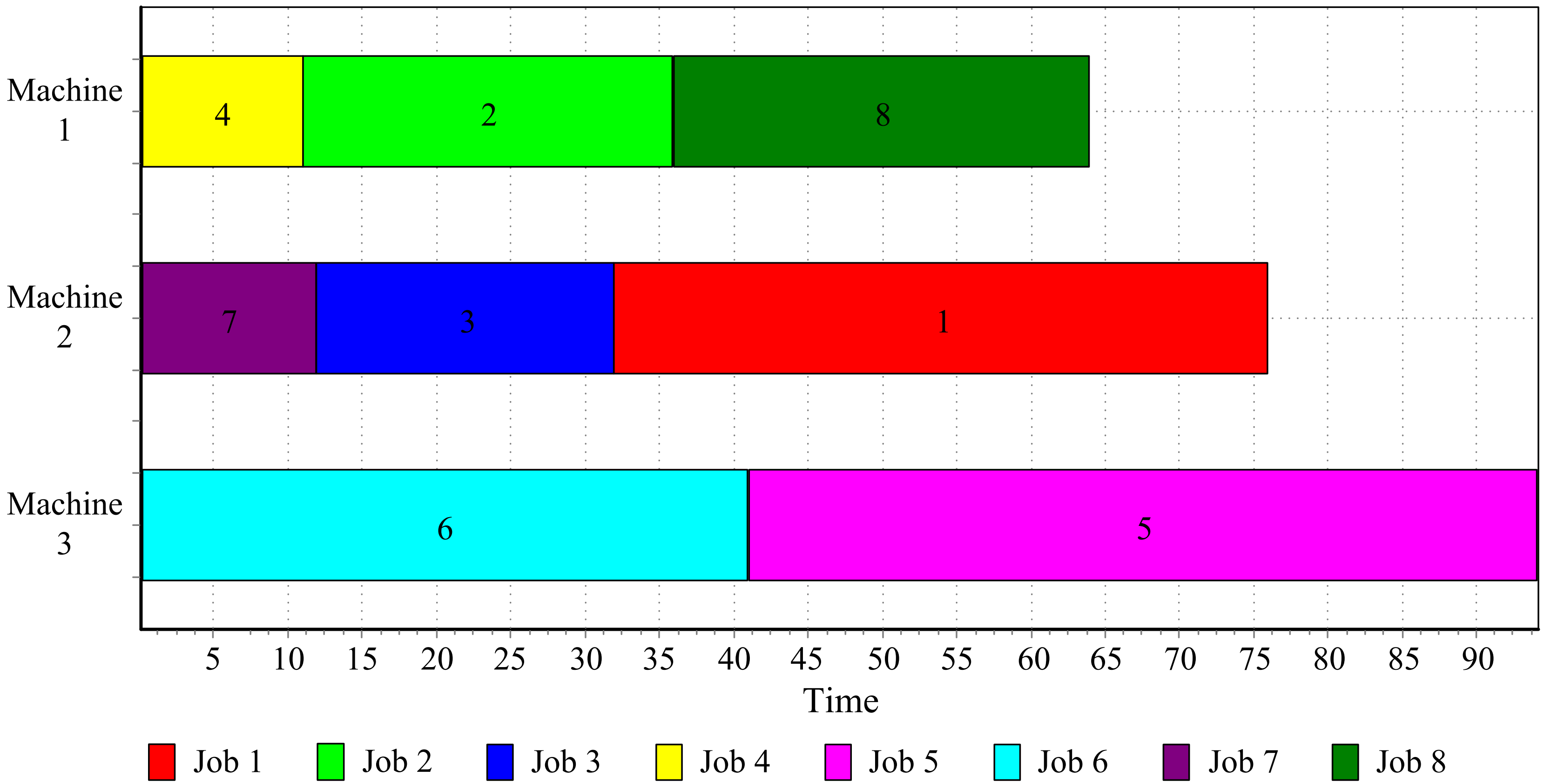




Sistemas de una etapa

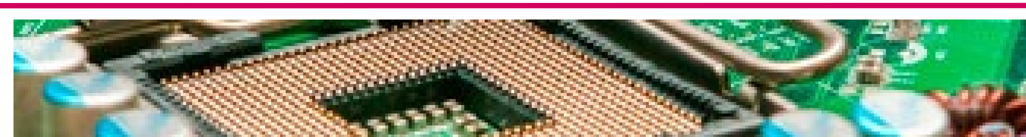


Sistemas de una etapa

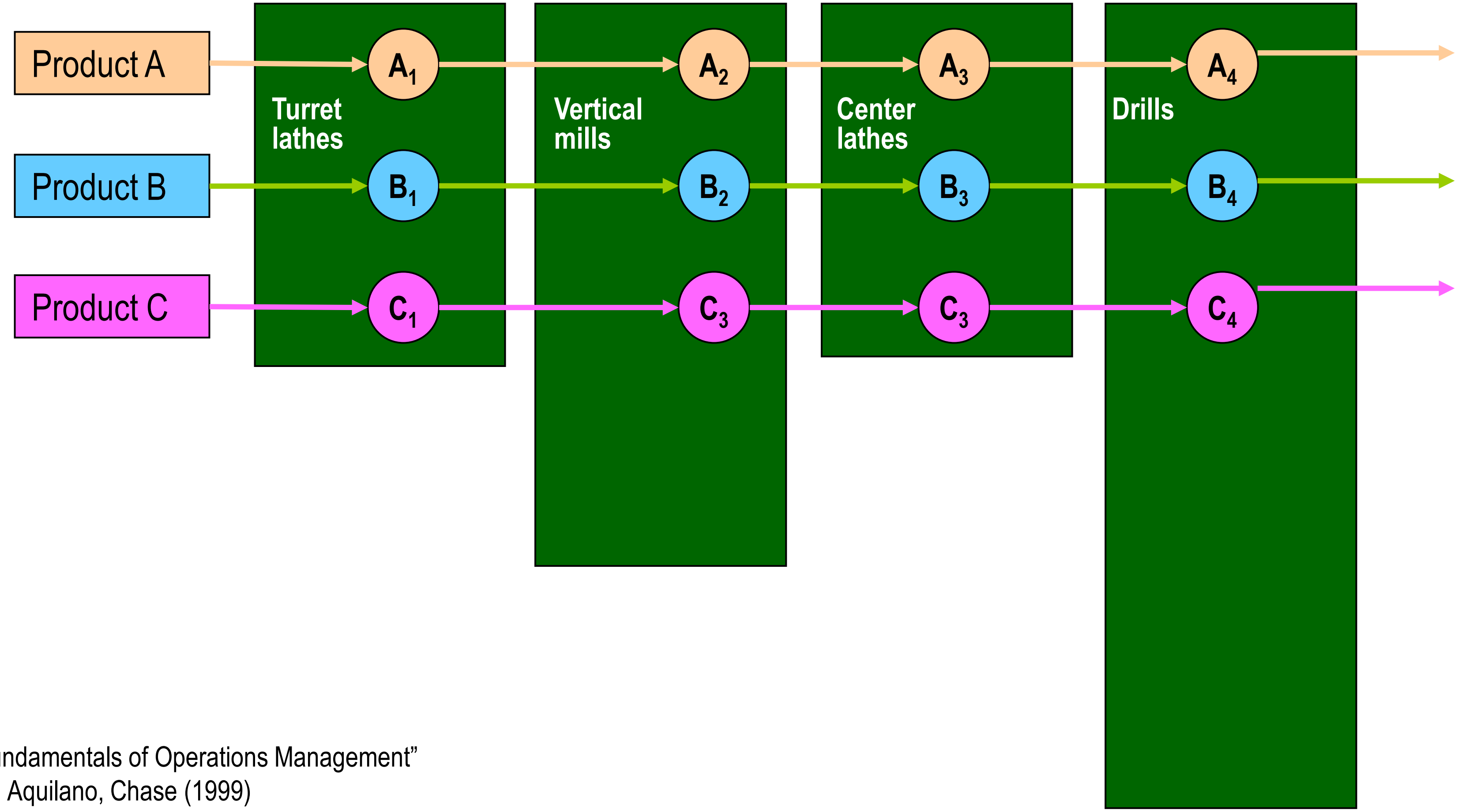


Sistemas multi-etapa

- **Flow-shop** F
  (cada tarea pasa por las m máquinas, con la misma ruta)
- **Job – shop** J
  (cada tarea tiene su propia ruta)
- **Open – shop** O
  (las tareas no tienen ruta predefinida)

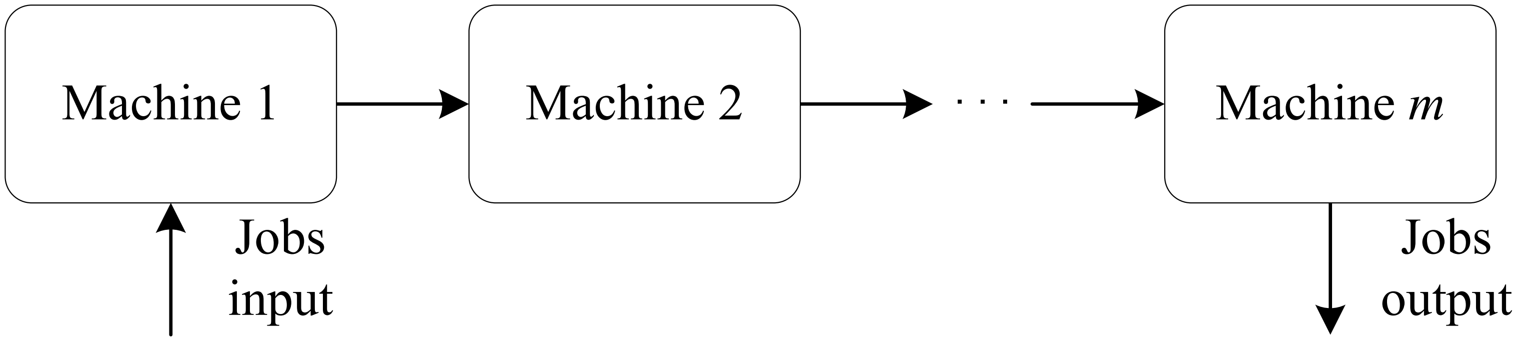


Flow Shop

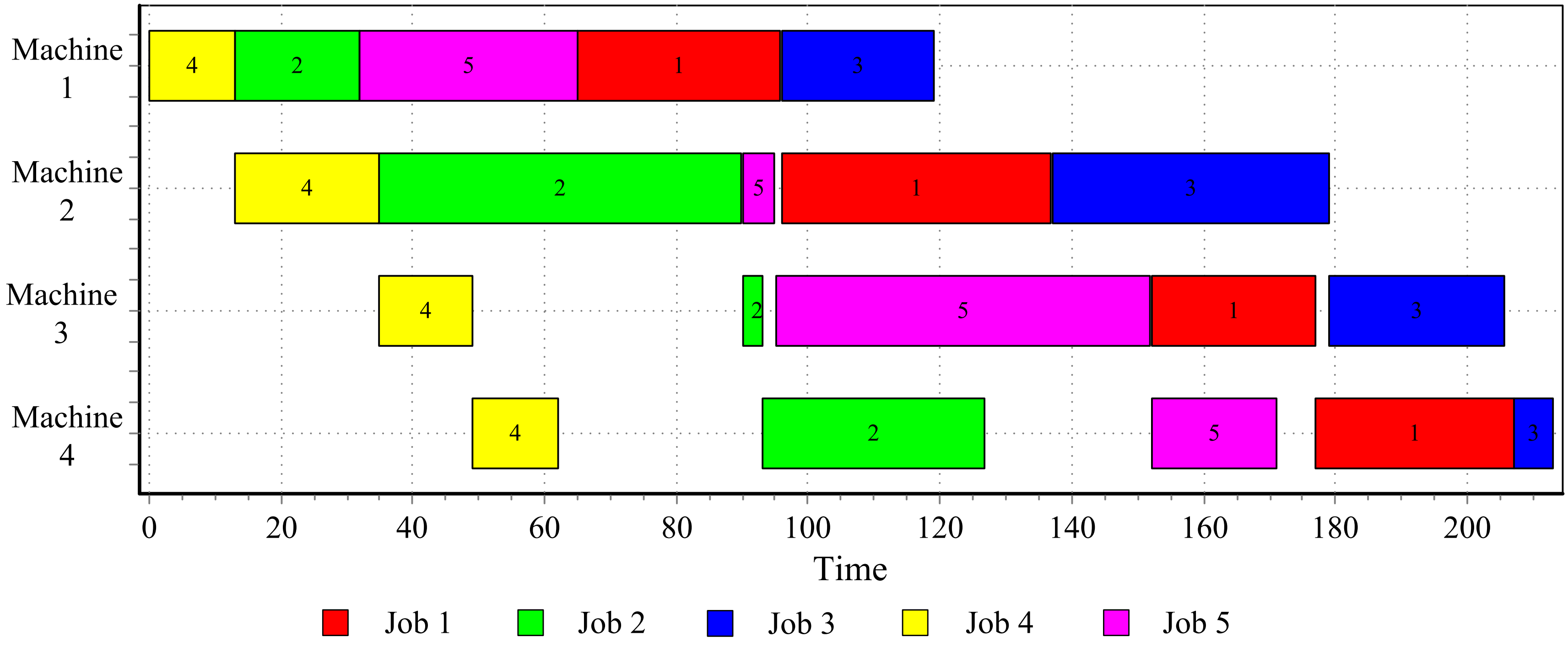


From "Fundamentals of Operations Management" by Davis, Aquilano, Chase (1999)

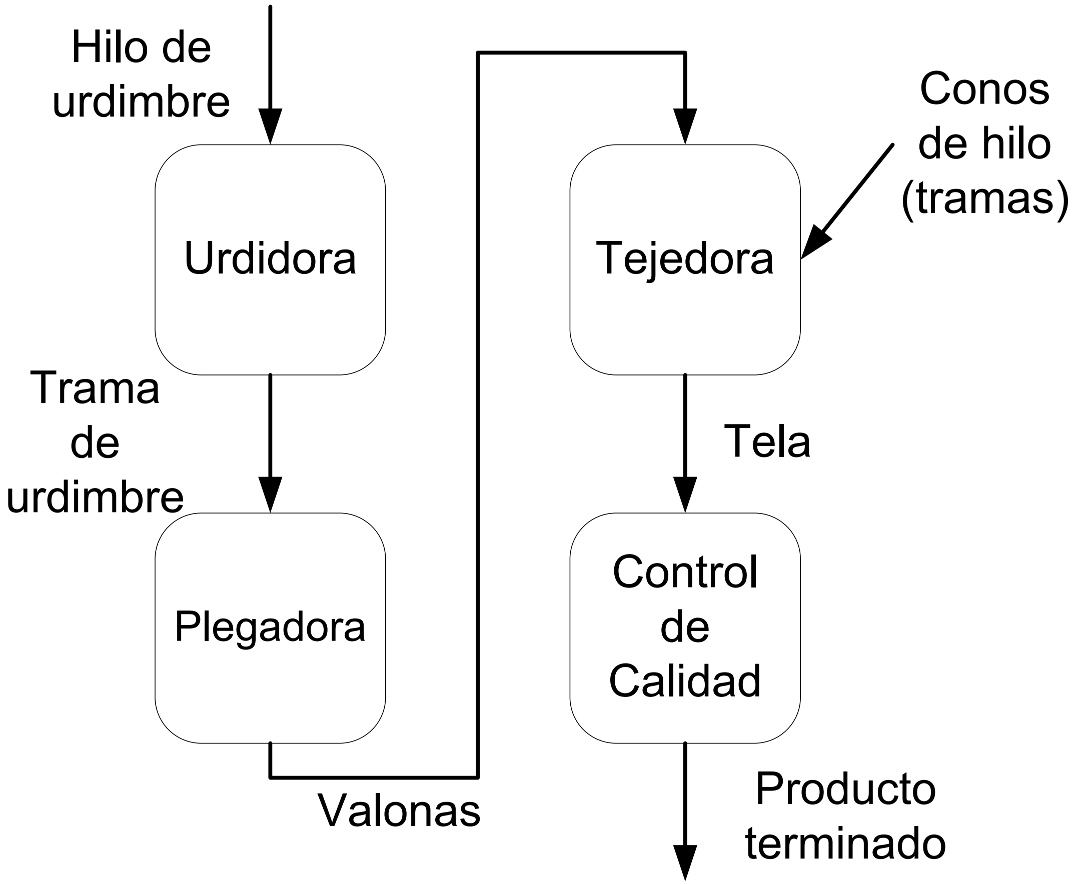
Sistemas multi-etapa



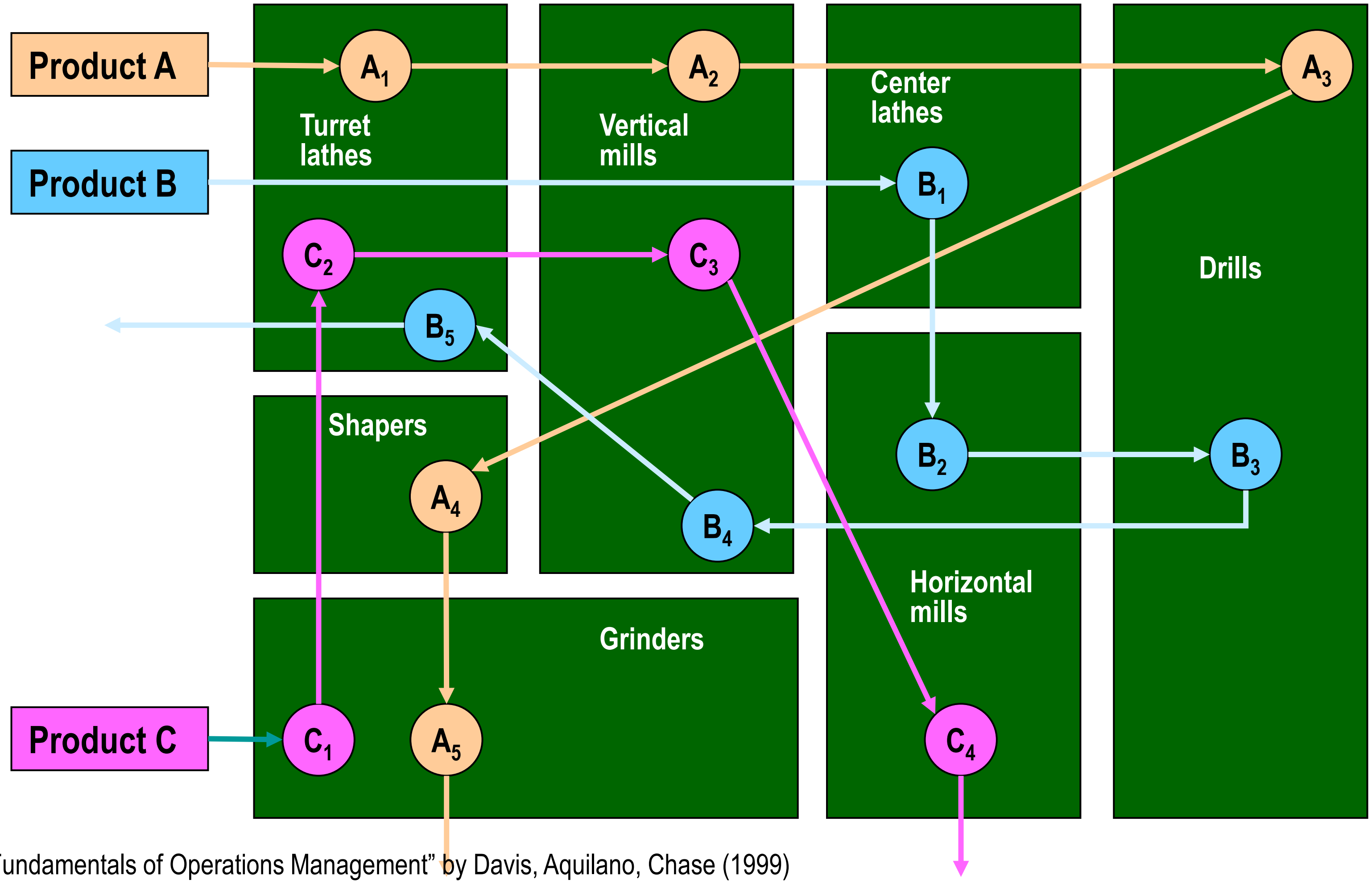
Sistemas multi-etapa



- Ejemplo de taller de flujo en el sector textil:

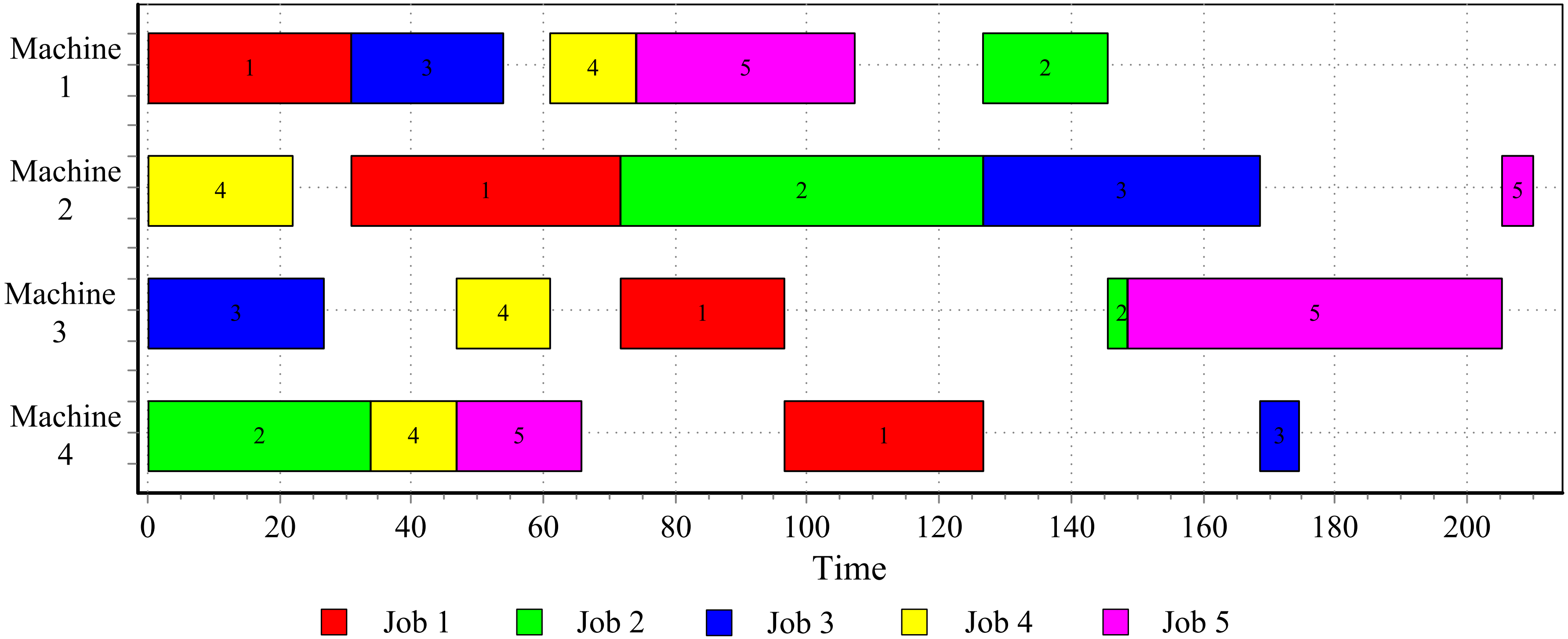


Job Shop



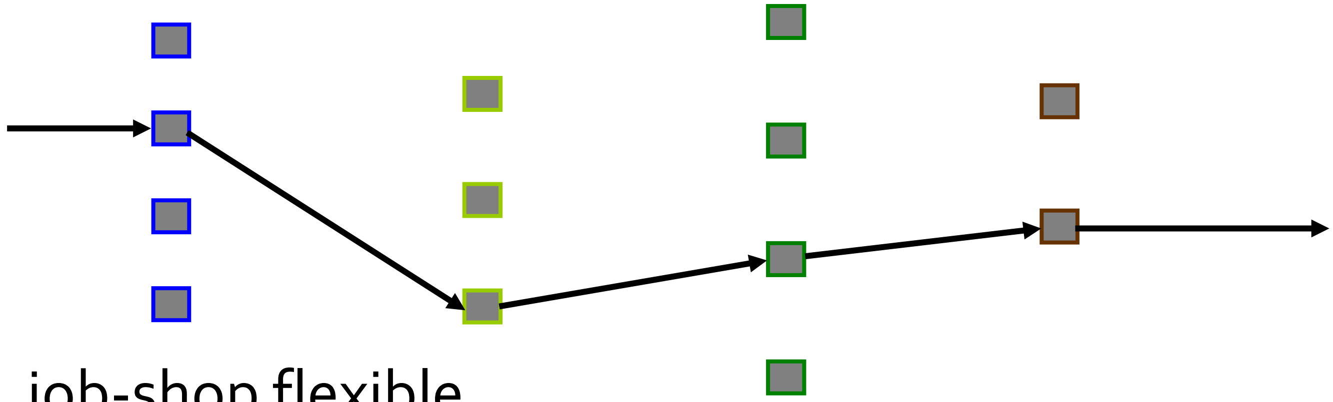
From "Fundamentals of Operations Management" by Davis, Aquilano, Chase (1999)

Sistemas multi-etapa



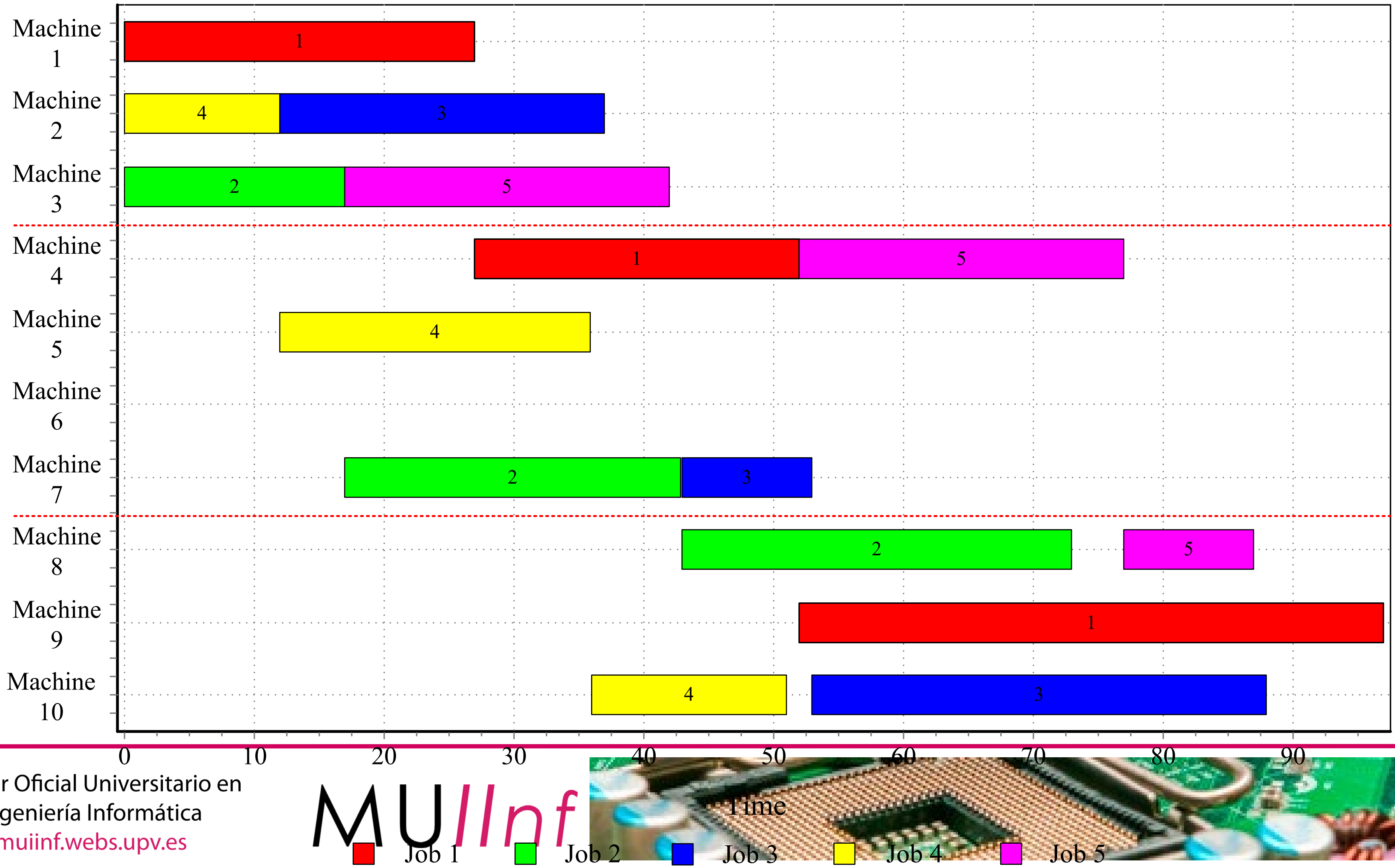
Sistemas multi-etapa flexibles

- **FF** flow-shop flexible
  (en vez de una máquina en cada fase,
  un conjunto de máquinas para elegir una)



- **FJ** job-shop flexible

Sistemas multi-etapa flexibles



Datos de entrada

- p_{ij} duración (processing time) de tarea j en máquina i
- r_j fecha de disponibilidad (release time)
- w_j peso (importancia)

- d_j fecha de entrega (due date)
- fecha límite (deadline)

$\bar{d}_j$

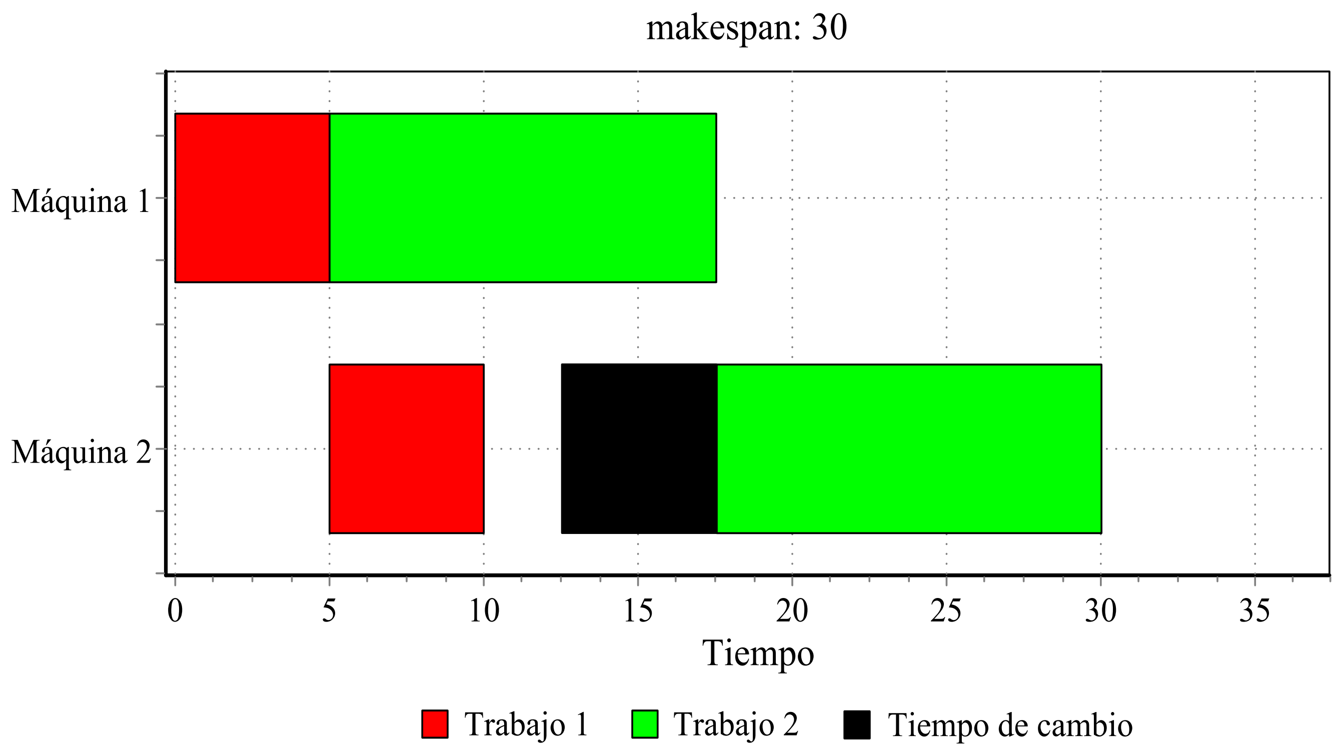
- s_{jk} tiempo de preparación (setup time) entre tareas j y k (s_{ijk})

β : Características de las tareas

- *prmp* posibilidad de interrupción (preemption)
- *prec* relaciones de precedencia
- *brkdwn* máquinas no disponibles en algunos momentos
- M_j máquinas en las que se puede procesar
- *prmu* mismo orden todas las tareas (permutation)
- *block* si se llena el buffer intermedio, se bloquea la máquina
- *nwt* las tareas se ejecutan sin parar entre máquinas (no-wait)
- *recrc* una tarea pasa más de una vez por una máquina

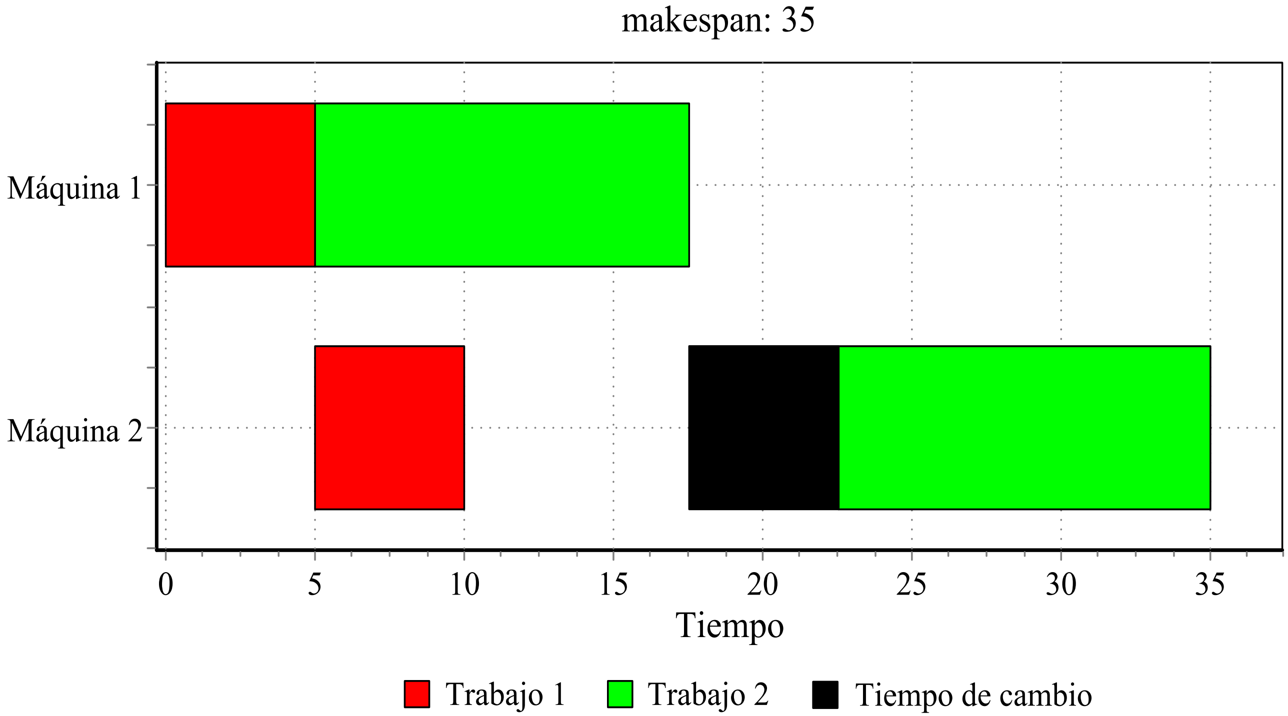
Tiempos de cambio dependientes de la secuencia: *setups*

- Tiempo de cambio anticipativo

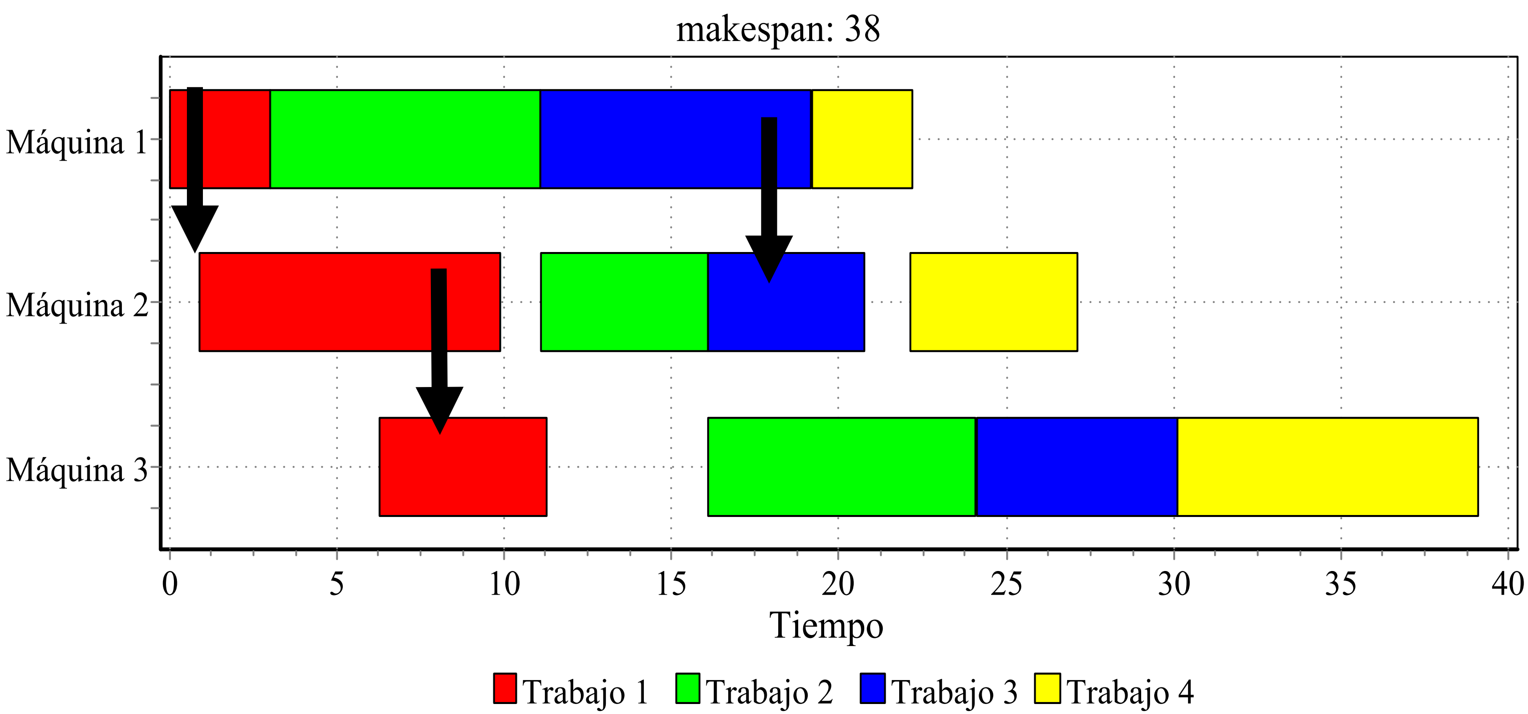


Tiempos de cambio dependientes de la secuencia: *setups*

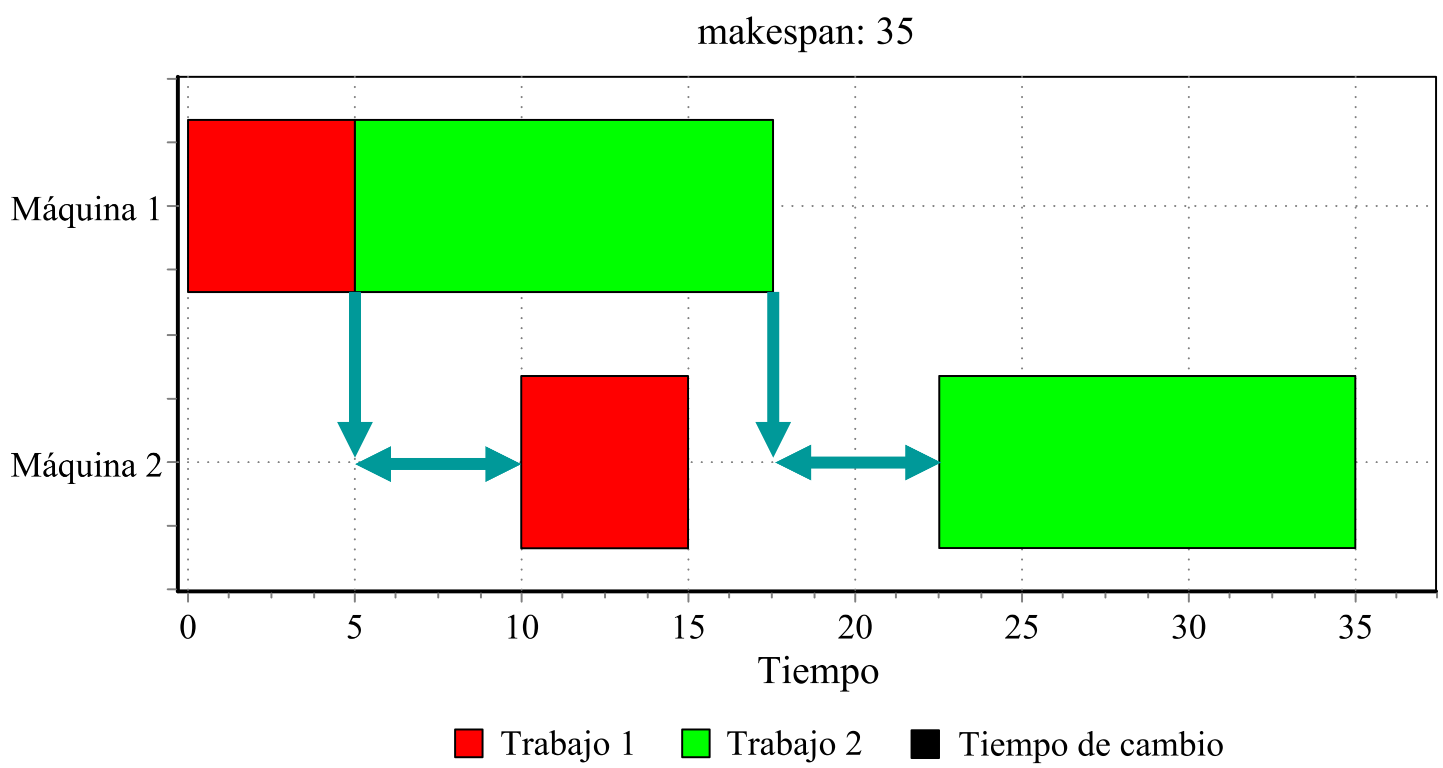
- Tiempo de cambio NO anticipativo:



Solapamiento entre operaciones

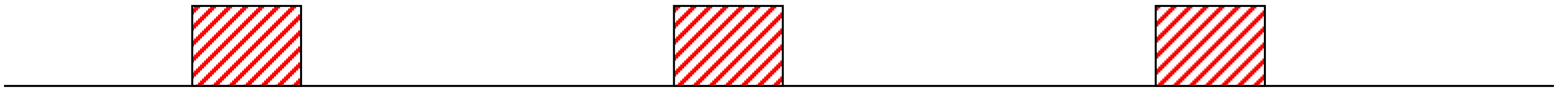


Tiempos de espera entre operaciones



Máquinas no disponibles continuamente: *brkdwn*

- Mantenimiento programado



- Programaciones previas de otras tareas

No todas las tareas en todas las máquinas: *Mj*

Sea el problema P3 | M_j | C_{max} con 8 jobs

Jobs	1	2	3	4	5	6	7	8
p_j	10	10	7	7	7	7	7	7

$M_1=\{1,3\}$

$M_2=\{2,3\}$

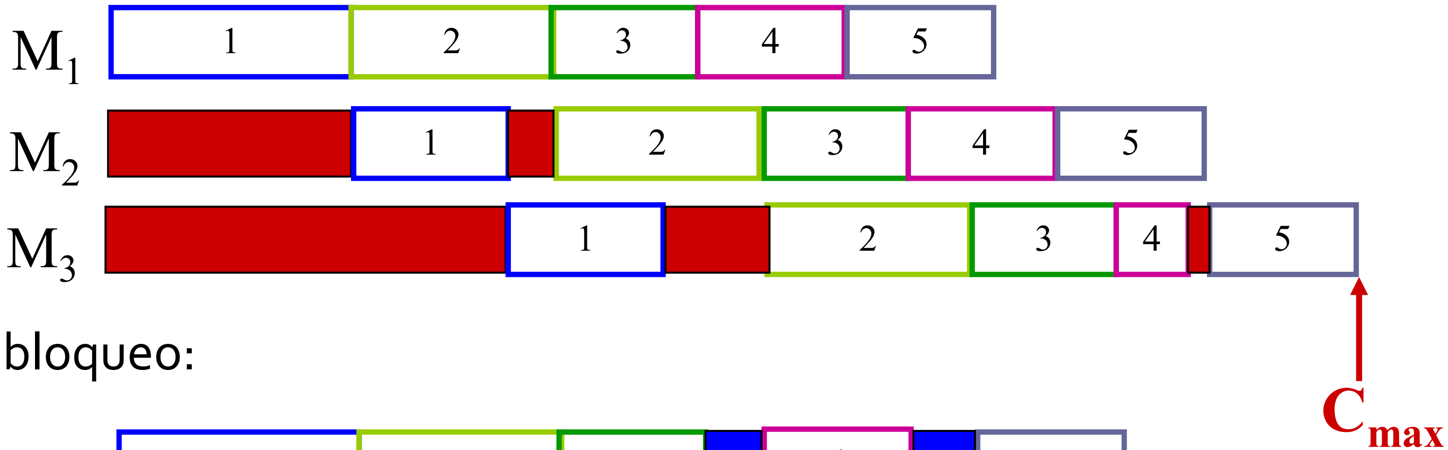
$M_3= M_4= M_5=\{1\}$

$M_6= M_7= M_8=\{2\}$

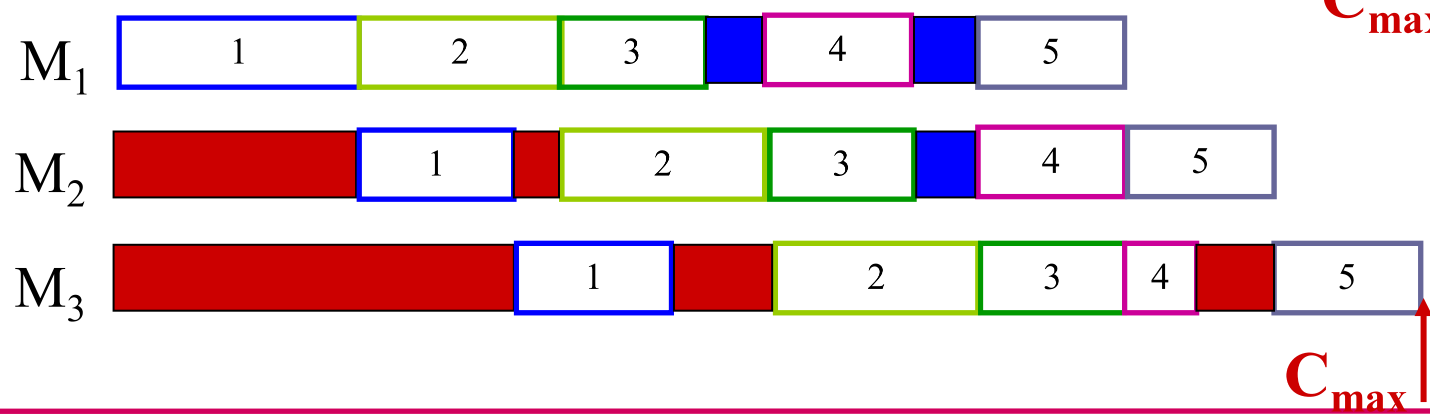
Estrategia: regla LFJ (que da prioridad al job menos flexible)

Sin almacenamiento intermedio: *block*

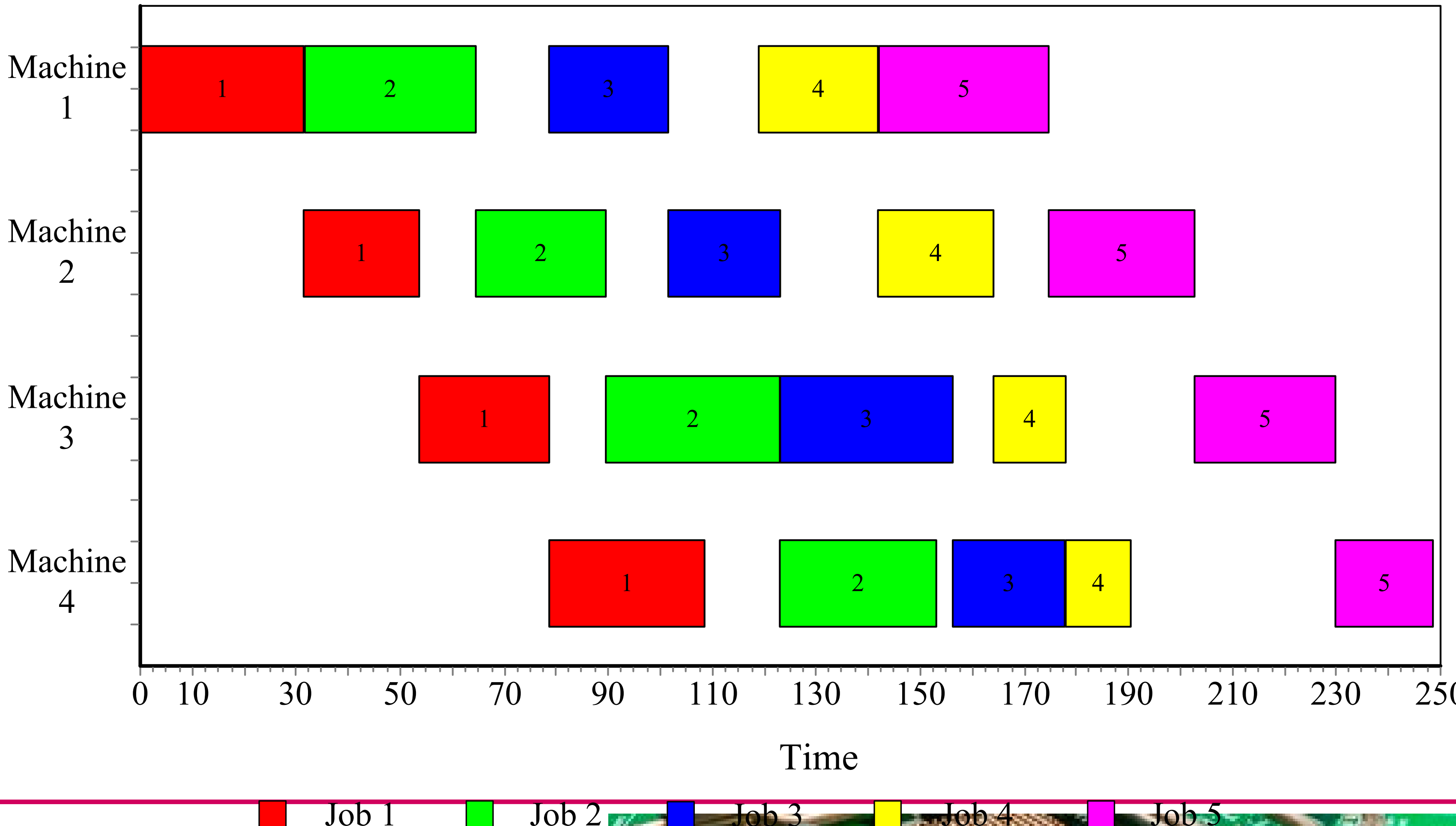
- Si no hay bloqueo:



- Si hay bloqueo:



Operaciones sin posibilidad de espera: *no-wait*



Otras restricciones

- Transporte de piezas entre máquinas
- Necesidad de otros recursos: moldes,….
- Restricciones de personal

γ: Características de los objetivos:
1.- Objetivos relativos a la completación

- C_j tiempo de completación del job j
  C_{ij} tiempo de completación de la operación *j* en la máquina *i*

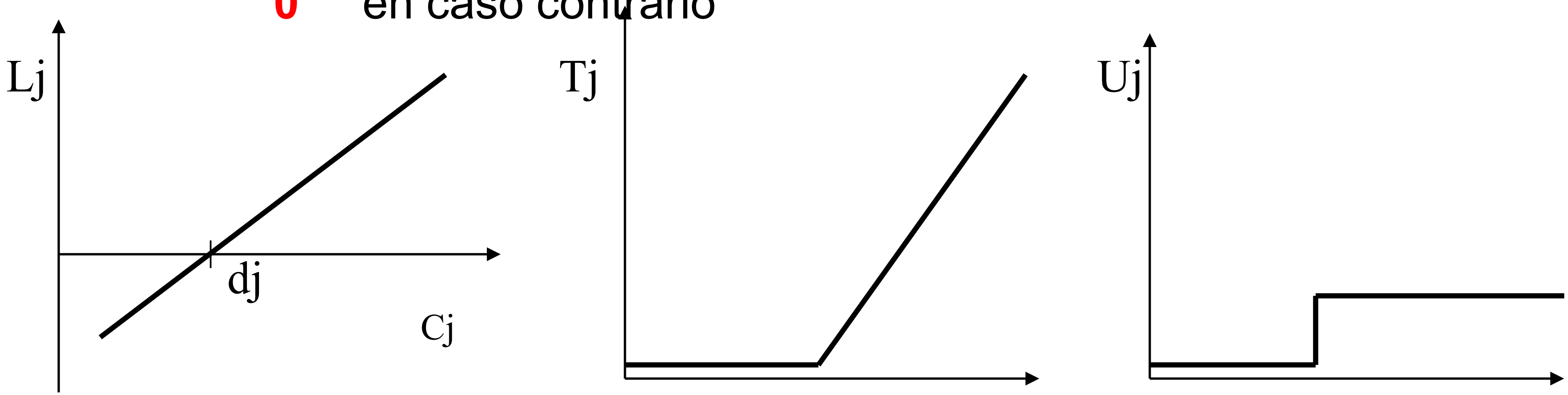
- C_{max} tiempo máximo de completación (makespan)

- ΣC_j tiempo total de completación

- $\Sigma w_j C_j$ tiempo total de completación ponderada

γ: Características de los objetivos:
2.- Objetivos relativos a las fechas de entrega

- $L_j = C_j - d_j$ retraso (lateness)
- $T_j = \max\{L_j, 0\}$ tardanza (tardiness)
- $U_j =$ 1 si $C_j > d_j$ job retrasado
  0 en caso contrario



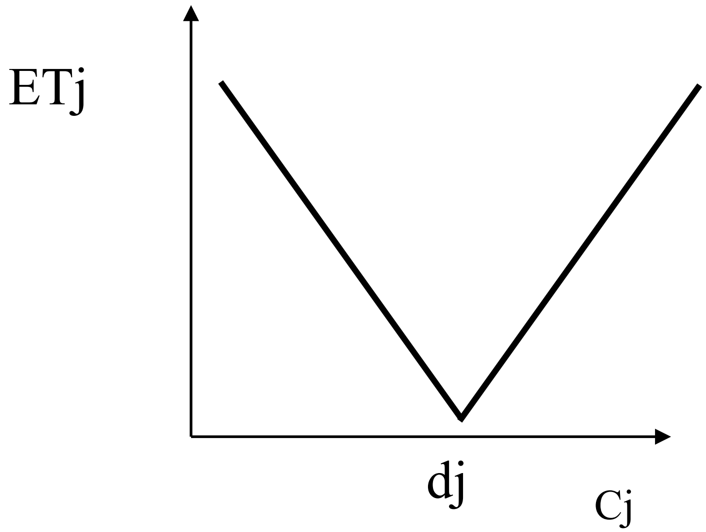
γ: Características de los objetivos:
2.- Objetivos relativos a las fechas de entrega

- L_{max} máximo retraso
- $\Sigma w_j T_j$ tardanza total ponderada
- ΣU_j número de jobs retrasados
- Minimización de la suma de adelantos y retrasos

$$ET = \sum_{j=1}^{n} \left(E_j + T_j \right)$$

- $E_j = max\ \{dj - Cj\ ,\ 0\}$

γ: Características de los objetivos:
2.- Objetivos relativos a las fechas de entrega



$$ET_w = \sum_{j=1}^{n} \left(w_j \cdot E_j + w'_j \cdot T_j \right)$$

Complejidad algorítmica

- **Problema**: problema genérico **P||Cmax**
- **Instancia :** caso particular, con datos

  m=2, n=7, pj = 2, 5, 2, 3, 5, 5, 8
- **Tamaño de la instancia:** longitud de la secuencia de datos que la definen
- En la práctica el tamaño será **n = número de tareas**

- **Eficiencia de un algoritmo**
  - número **máximo** de pasos de computación necesarios
  - para obtener la solución óptima,
  - en función del tamaño de la instancia

- **Paso de computación** : operación (comparación, operación aritmética,.....)

- **Cálculo del número de pasos**: $1500 + 100\,n^2 + 5\,n^3$
  - Sólo interesa el término mayor: n^3
  - No es importante el coeficiente: $O(n^3)$

$$[f(n)=O(g(n)) \text{ si } \; c>0 \,/\, f(n)\leq c\, g(n)]$$

Cálculo del número de pasos

- Calcular el máximo de un conjunto de n números positivos p_i:

  max = 0
  For (i=1; i=n; i++)
  {
  if (*max* < p_i)
  max= p_i
  }

- Número máximo de pasos: 2n+1

- Complejidad: O(n)

Una clasificación sencilla

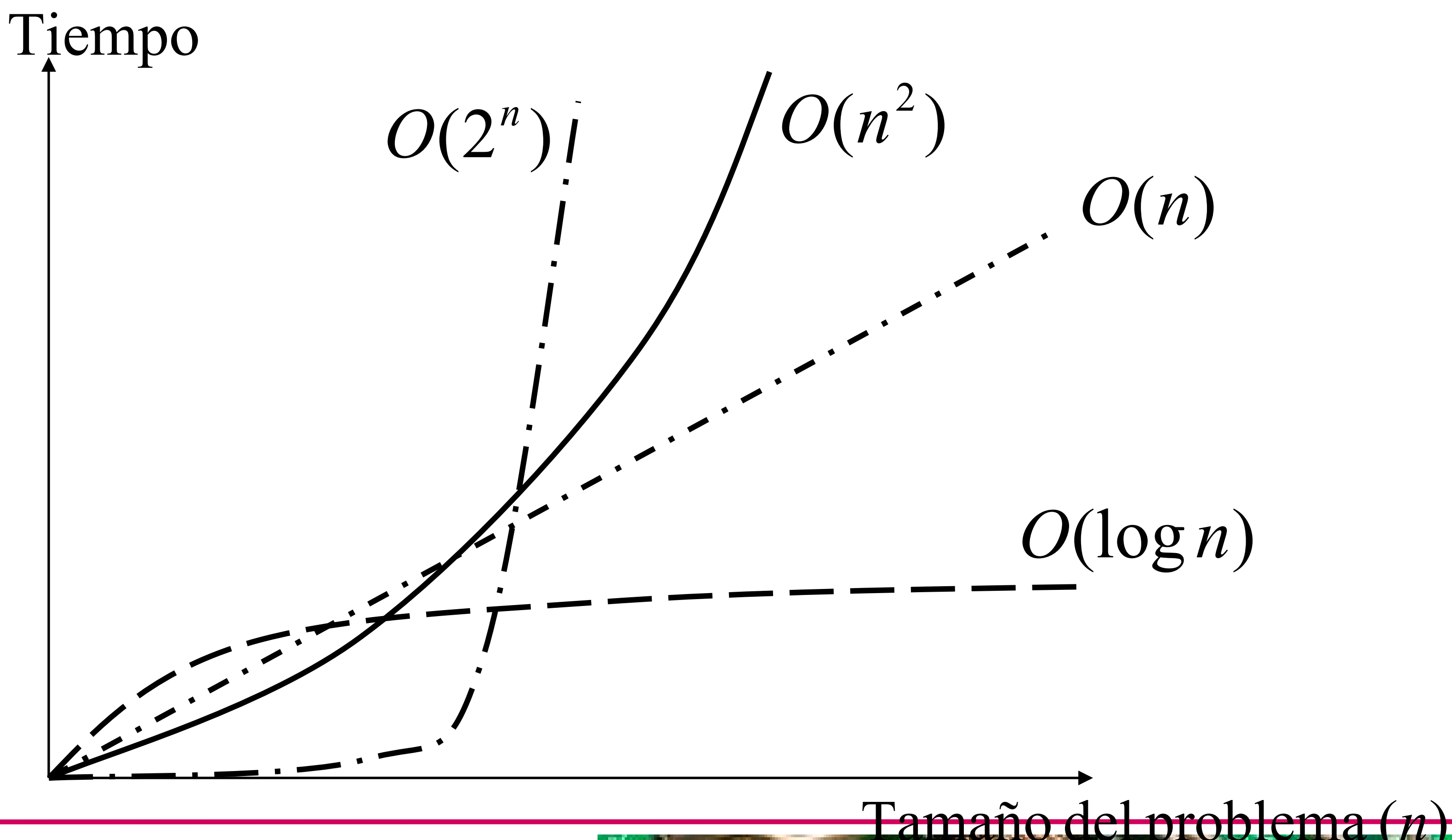
- **Problemas "fáciles": existe un Algoritmo de complejidad polinómica $O(n^k)$**

- **Problemas difíciles (NP-hard): Sólo existen algoritmos de complejidad exponencial O(2n)**

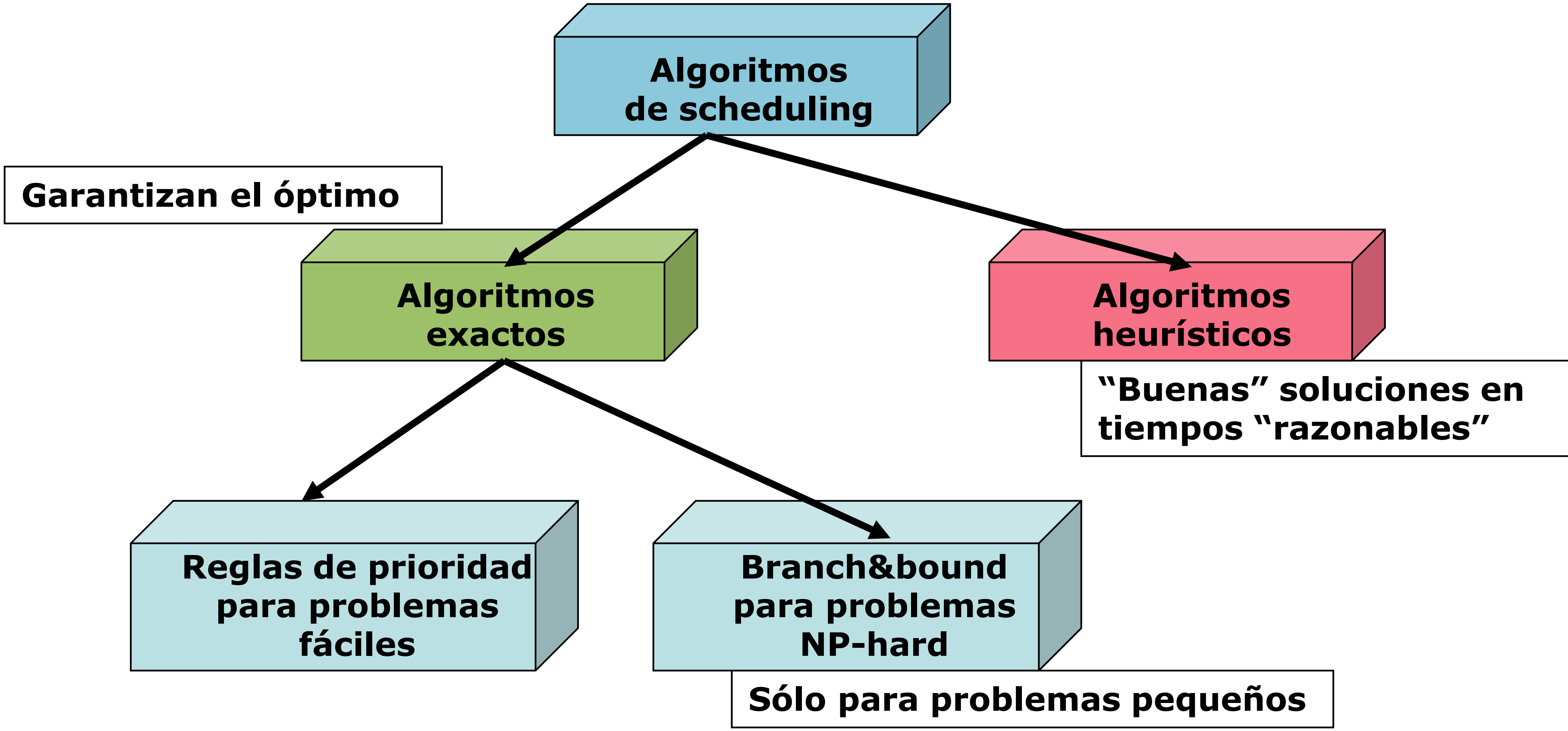
Algoritmos polinómicos y exponenciales

Complejidad	Tamaño					
	10	20	30	40	50	60
n	0.00001 seg	0.00002 seg	0.00003 seg	0.00004 seg	0.00005 seg	0.00006 seg
n^2	.0001 seg	.0004 seg	.0009 seg	.0016 seg	.0025 seg	.0036 seg
n^5	.1 seg	3.2 seg	24.3 seg	1.7 min	5.2 min	13 min
2^n	.001 seg	1.0 seg	17.9 min	12.7 días	35.7 años	366 siglos

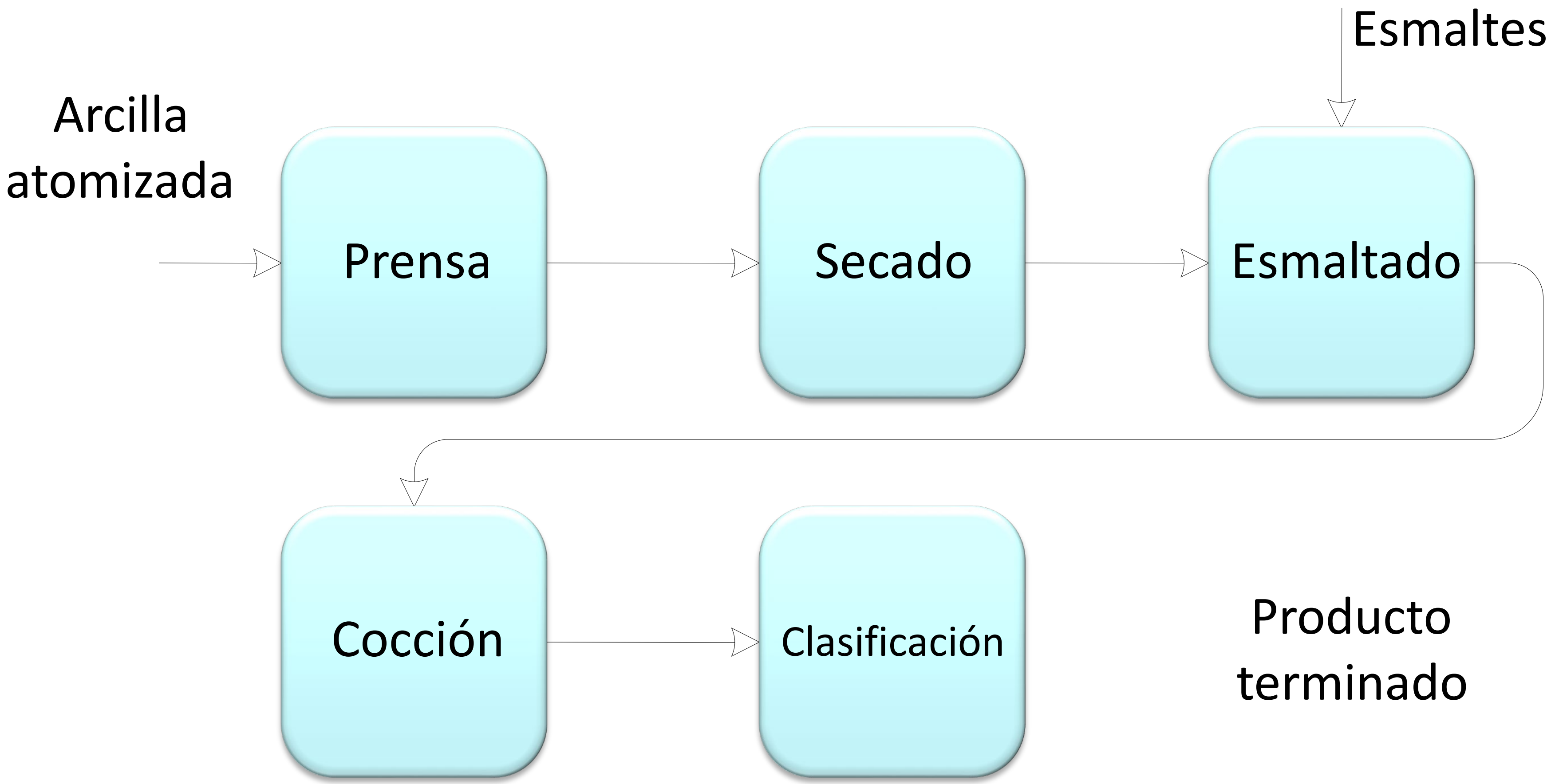
Algoritmos polinómicos y exponenciales



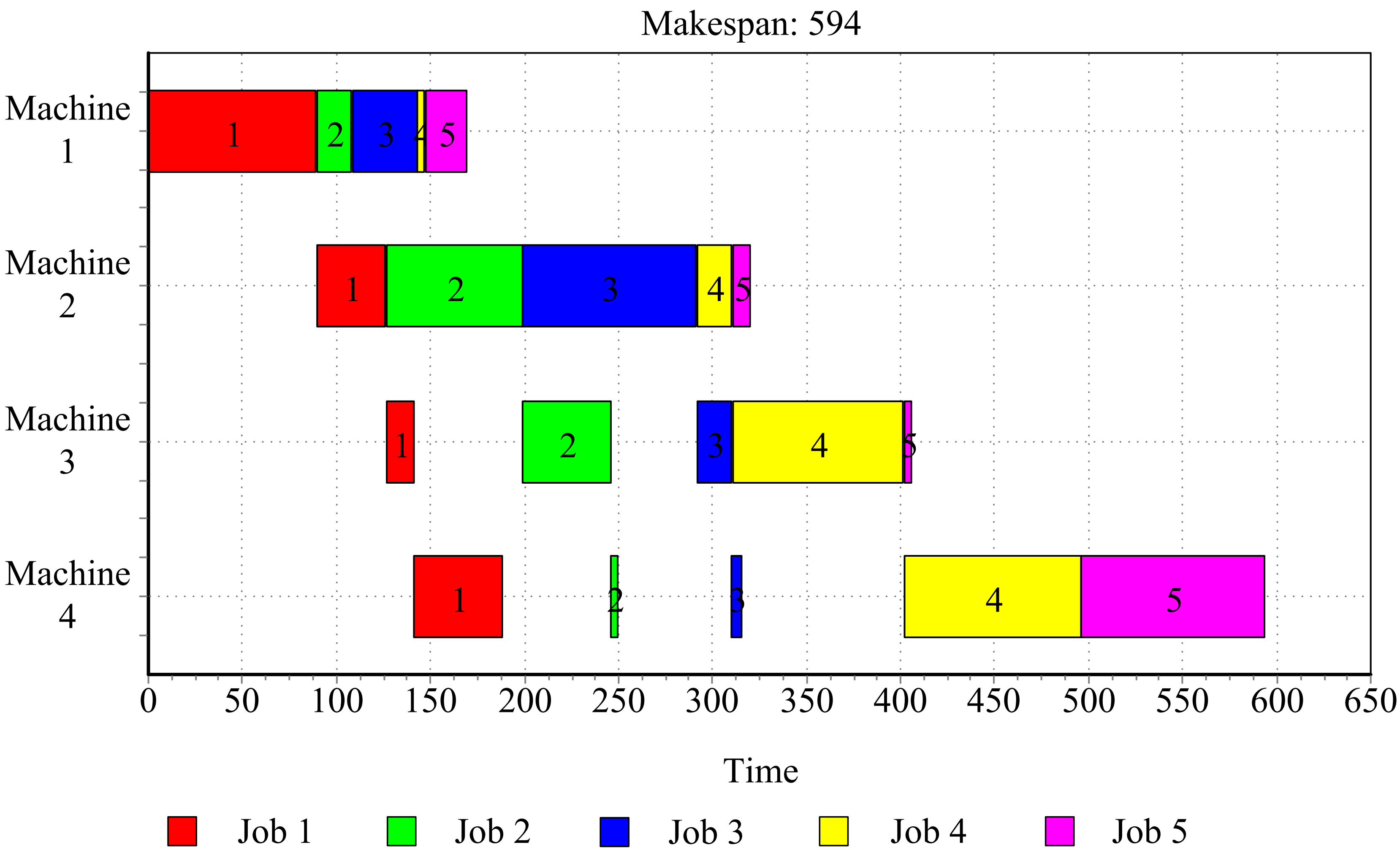
Estrategias de solución



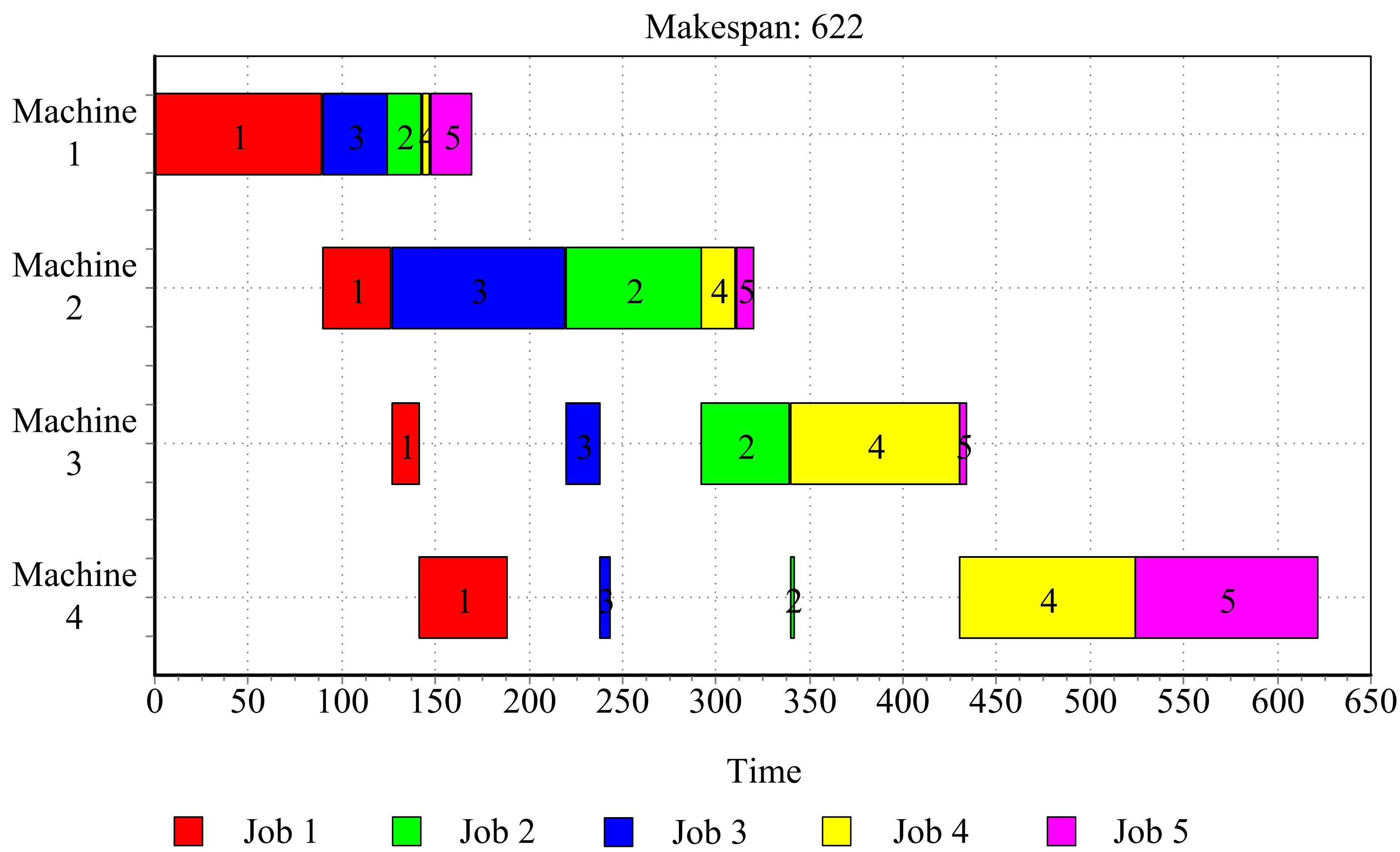
Por qué la programación es importante



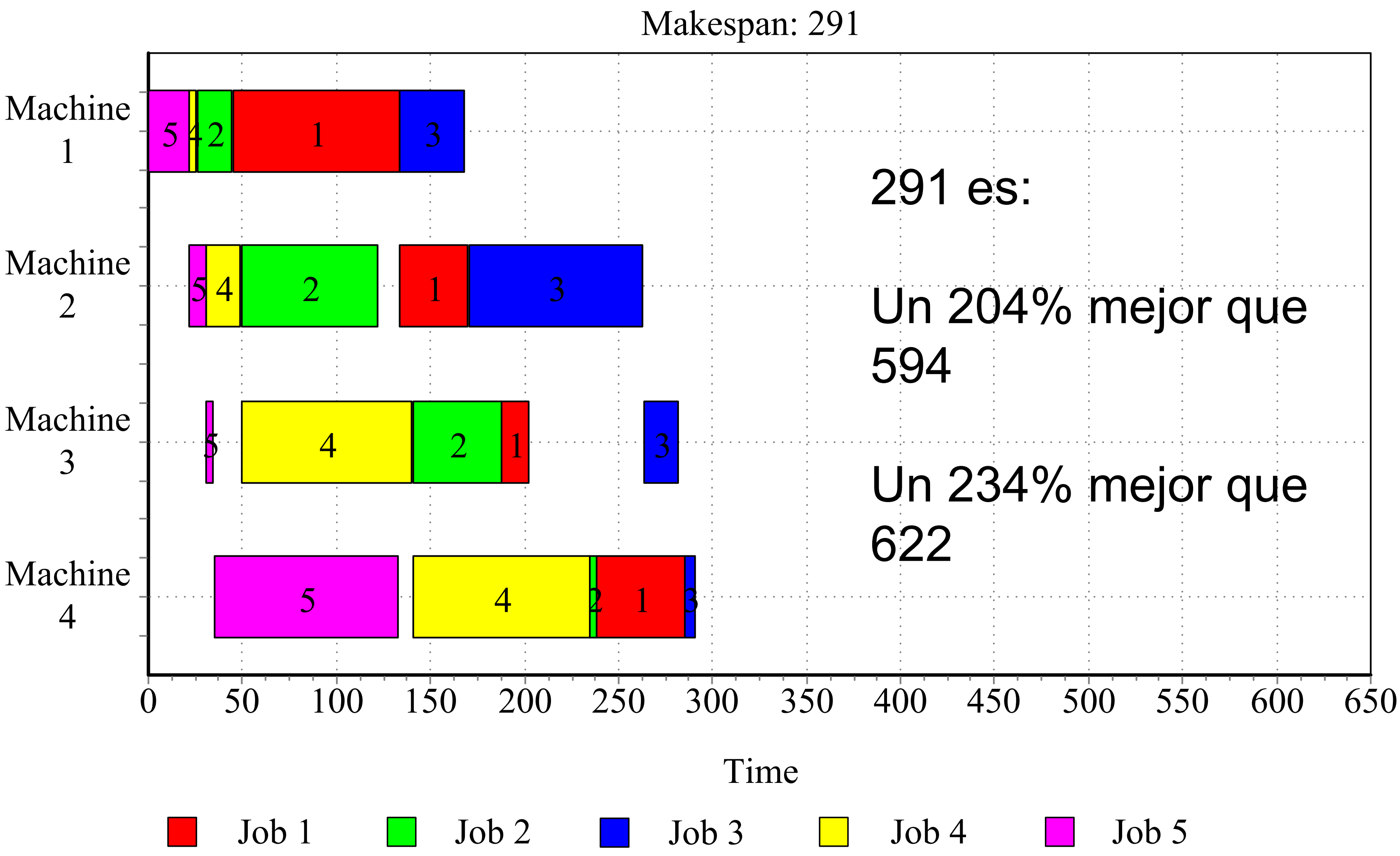
Por qué la programación es importante



Por qué la programación es importante



Por qué la programación es importante



291 es:

Un 204% mejor que 594

Un 234% mejor que 622

Por qué la programación es difícil:

Secuencias posibles para *n* trabajos en problemas de una máquina sencillos: $n!$

Secuencias posibles para n trabajos en un taller de flujo: $(n!)^m$

Vamos a suponer un sencillo ejemplo de 10 trabajos y 5 máquinas ($n=10$, $m=5$)

$(10!)^5 = 6{,}29 \cdot 10^{32}$ secuencias

Supongamos que podemos calcular una secuencia en 1^{-9} segundos (1000 millones de secuencias por segundo)

Por qué la programación es difícil:

Calcular todas las secuencias llevaría $1{,}99 \cdot 10^{16}$ años

Aproximadamente “un millón y medio de VU”

VU=Vidas del Universo

Para un caso más realista (n=75, m=20) no es ni siquiera posible hacer el cálculo de cuántas VU llevaría

LA MAYORÍA DE LOS PROBLEMAS REALES DE SCHEDULING SON TREMENDAMENTE COMPLEJOS DE RESOLVER

Bibliografía

- Framinan, J. M., Leisten, R. y Ruiz, R. (2014) "Manufacturing Scheduling Systems. An Integrated View on Models, Methods and Tools" Springer
- Michael Pinedo (2016) "Scheduling: Theory, Algorithms, and Systems" Springer, quinta edición.
- Peter Brucker (2007) "Scheduling Algorithms" Springer, quinta edición.
- Scheduling Computer and Manufacturing Processes
  J. Blazewicz, K.H. Ecker, E. Pesch, G. Schmidt, J. Weglarz, Springer Verlag; 2nd edition, 2002
- Heuristic Scheduling Systems : With Applications to Production Systems and Project Management
  T. Morton and David W. Pentico. John Wiley and Sons, 1993
- Michael Pinedo (2009) "Planning and Scheduling in Manufacturing and Services" Springer, segunda edición

Bibliografía

- Pinedo, M. L., 2005. Planning and scheduling in manufacturing and services. Springer series in operations research. Springer, New York, USA.
- Edis, E. B., Oguz, C., Ozkarahan, I., 2013. Parallel machine scheduling with additional resources: Notation, classification, models and solution methods. European Journal of Operational Research 230, 449–463.
- Alberto Caprara, Michele Monaci, Paolo Toth, 2003. Models and algorithms for a staff scheduling problem. Mathematical Programming, Series B 98, 445-476.
- Luis Fanjul, Federico Perea, Rubén Ruiz, 2015. Algorithms for the Unspecified Unrelated Parallel Machine Scheduling Problem with additional Resources. (presented at the 6th IESM Conference, October 2015, Seville, Spain)