

UNIVERSITAT POLITÈCNICA DE CATALUNYA – BARCELONATECH  
OPE – ORGANIZACIÓN DE LA PRODUCCIÓN Y DE EMPRESA (ASPECTOS TÉCNICOS, JURÍDICOS  
Y ECONÓMICOS EN PRODUCCIÓN )

## Dirección de Operaciones. Proyectos singulares - II

DIRECCIÓN DE OPERACIONES 240EO024 – Máster Universitario en Ingeniería de Organización  
(240MUEO) - ETSEIB

Joaquín Bautista · Rocío Alfaro

OPE-PROTHIUS – OPE-MSc.2017/04 240EO024 (20170214) - <http://futur.upc.edu/OPE> - [www.prothius.com](http://www.prothius.com) -  
Departamento de Organización de Empresas – ETSEIB · UPC



**PROTHIUS**  
Càtedra Organització Industrial

DO' 17 – Pro (II) 0  
J. Bautista, R. Alfaro

# Contenido

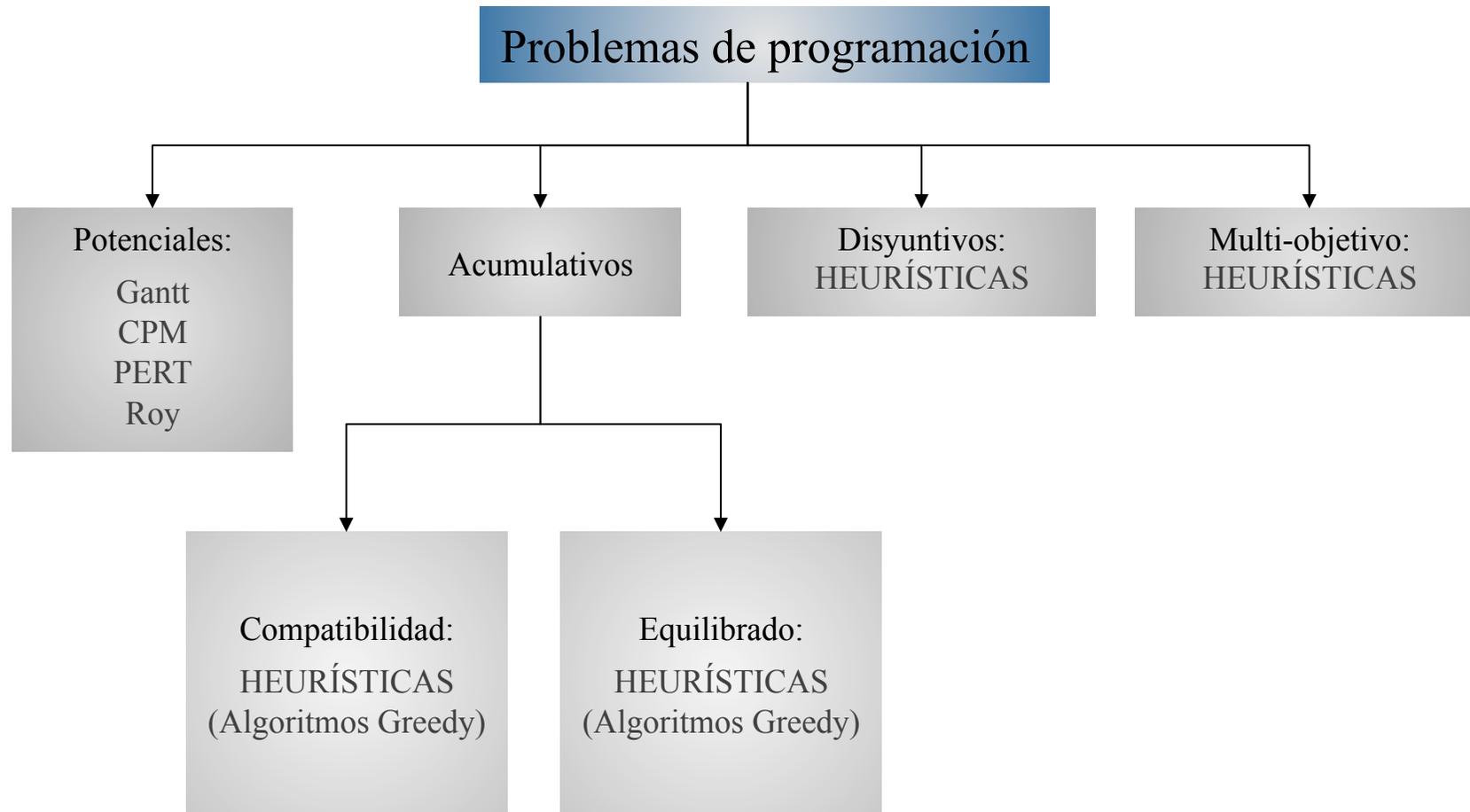
- Problemas. Tipología
- Procedimientos. Tipología
- Problemas acumulativos
  - Definición y categorías
  - Parámetros, variables y relaciones
  - Restricciones
  - Cotas inferiores
- Ejemplo 1. Datos y cotas inferiores
- Ejemplo 2. Enunciado y cotas inferiores
- Problemas de compatibilidad. Algoritmos Greedy
- Ejemplo 2. Problemas de compatibilidad. Algoritmos Greedy
- Problemas de equilibrado. Algoritmos Greedy
- Ejemplo 2. Problemas de equilibrado. Algoritmos Greedy

- Bautista, J.; Companys, R.; Corominas, A. (1998) Gestió de projectes. Editorial UOC, BCN. ISBN: 9788495131003

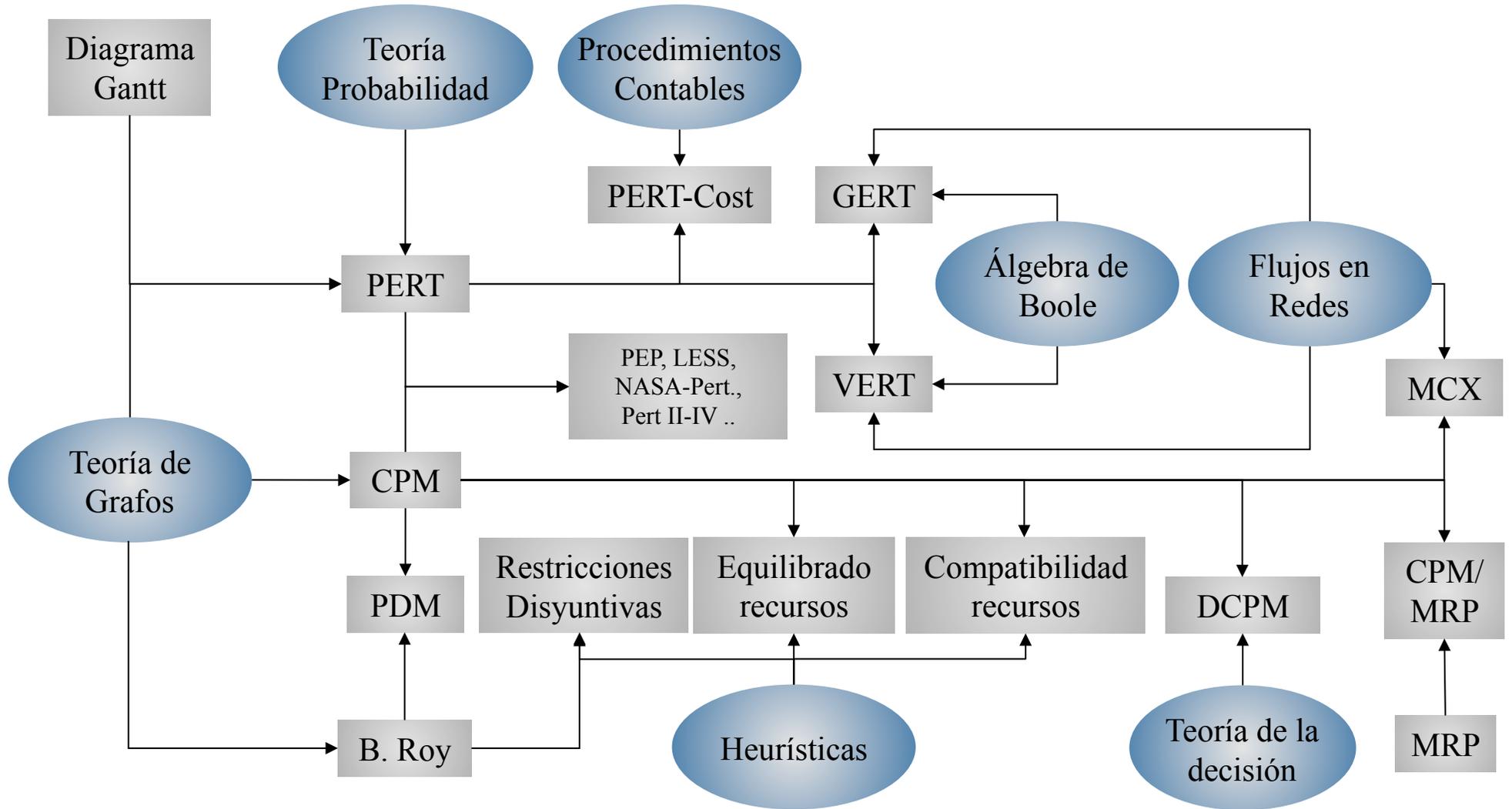
- Companys, R.; Corominas, A. (1998) Organización de la producción I. Diseño de sistemas productivos 2. Capítulo 7. Edicions UPC. BCN



# Problemas. Tipología



# Procedimientos. Tipología



# Problemas Acumulativos. Definición y categorías

## *Definición:*

- Problemas de programación de proyectos en los que se tratan a la vez las restricciones potenciales y las acumulativas.

## *Categorías de problemas:*

- *Problemas de compatibilidad de recursos:*
  - Dados unos niveles de disponibilidad de recursos dependientes del tiempo, *Establecer un programa compatible* del proyecto que satisfaga: (1) las restricciones potenciales y (2) que, en todo momento, los niveles de utilización de los recursos sean menores o iguales que sus niveles de disponibilidad.
- *Problemas de equilibrado de recursos:*
  - Dados una duración límite del proyecto y unos niveles de referencia de utilización de los recursos, *Establecer un programa* que satisfaga: (1) las restricciones potenciales y (2) que, en todo momento, los niveles de utilización y de disponibilidad de los recursos se ajusten lo mejor posible.



# Problemas Acumulativos. Parámetros, variables y relaciones

## *Parámetros y variables adicionales:*

- $T$  Duración del proyecto.
- $K(t)$  Conjunto de recursos renovables en el instante  $t$  ( $t=1, \dots, T$ ).  $K = \bigcup_{t=1}^T K(t)$
- $R_k(t)$  Nivel de disponibilidad en el instante  $t$  ( $t=1, \dots, T$ ) del recurso renovable  $k \in K(t)$
- $r_k(t)$  Tasa de utilización del recurso renovable  $k$  en el instante  $t$  ( $t=1, \dots, T$ ).
- $r_{j,k}$  Tasa de utilización del recurso renovable  $k$  por parte de la actividad  $j \in J$
- $A(t)$  Conjunto de actividades que se están ejecutando en el instante  $t$  ( $t=1, \dots, T$ ),  $A(t) \subseteq J$
- $w_k(t)$  Sobrecarga del recurso  $k$  en el instante  $t$ .
- $W_k$  Sobrecarga global del recurso  $k$  a lo largo del proyecto.
- $P_j$  Duración de la tarea  $j \in J$

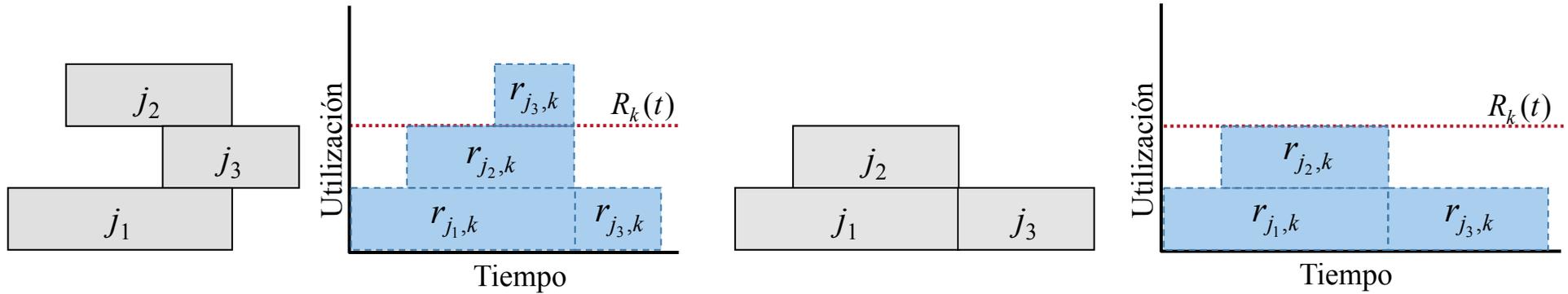
## *Relaciones:*

$$r_k(t) = \sum_{j \in A(t)} r_{j,k} ; w_k(t) = \max\{0, r_k(t) - R_k(t)\} \quad \forall t = 1, \dots, T \wedge \forall k \in K; W_k = \sum_{t=1}^T w_k(t)$$

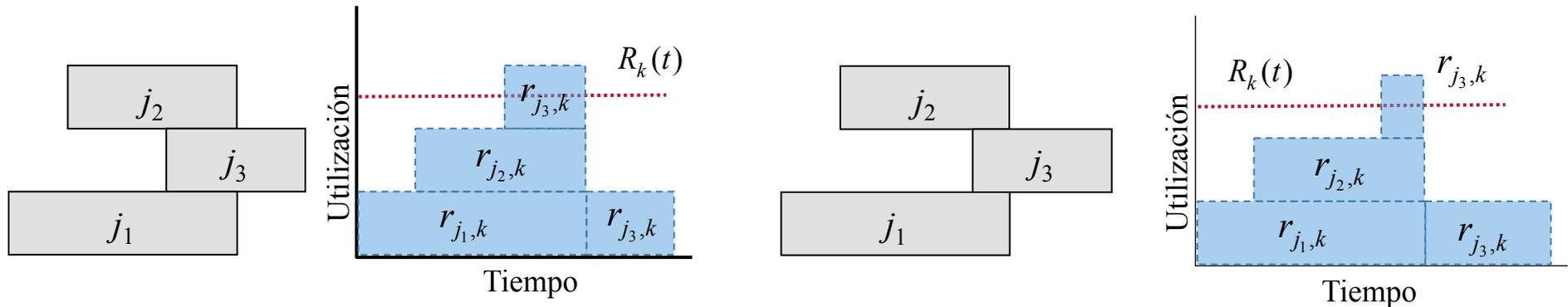


# Problemas Acumulativos. Restricciones

*Restricciones de compatibilidad:*  $\sum_{j \in A(t)} r_{j,k} \leq R_k(t) \Rightarrow w_k(t) = 0 \quad \forall t = 1, \dots, T, \forall k \in K(t)$



*Restricciones de equilibrado:*  $\min f(W_k)$



# Problemas Acumulativos. Cotas inferiores

## Concepto:

Una cota inferior es una aproximación por defecto de la mejor solución que puede obtenerse para un problema.

## Método:

Resolver de forma exacta el problema, descartando una parte de su complejidad.

1. Considerando las restricciones potenciales, pero no las acumulativas.

- *Problema potencial*:  $LB_1 = T^* \equiv T^{\min} = \max_{j \in J} \{e_j^{\min}\} = s_{\omega}^{\min}$

2. Considerando los recursos, pero no las restricciones potenciales.

- *Problema acumulativo*:  $LB(k) = T \frac{\sum_{j=1}^{|J|} p_j \cdot r_{j,k}}{\sum_{t=1}^T R_k(t)} \quad \forall k \in K \Rightarrow LB_2 = \max_{k \in K} \{LB(k)\}$



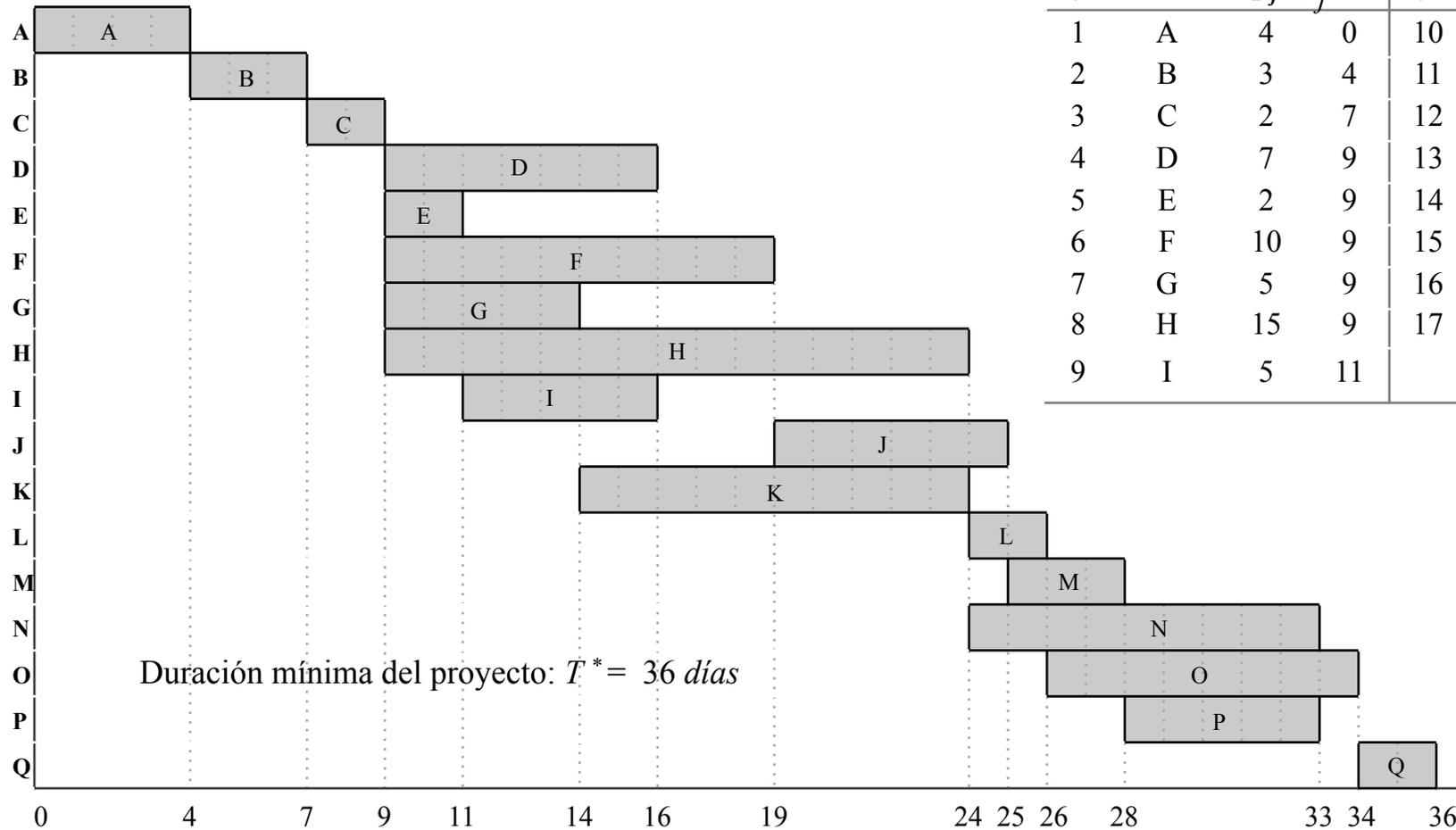
## Ejemplo 1. Datos y cotas inferiores (1)

$j$	CÓDIGO	DESCRIPCIÓN	$p_j$ (días)	$P_j$	$F_j$	$r_{j,A}$	$r_{j,B}$
1	A	Despejar emplazamiento	4	-	B	1	1
2	B	Medición y replanteo	3	A	C	2	3
3	C	Explanación	2	B	D, E, G, H, F	4	-
4	D	Preparación acometida eléctrica	7	C	P	2	4
5	E	Excavación conducciones eléctricas	2	C	I	1	1
6	F	Excavación desagües	10	C	J	2	4
7	G	Cimientos depósito agua	5	C	K	1	3
8	H	Perforación pozo	15	C	L	1	2
9	I	Instalación conducciones eléctricas	5	E	M	2	4
10	J	Instalación tuberías desagües	6	F	M	1	7
11	K	Construcción depósito agua	10	G	N	3	7
12	L	Instalación Bomba	2	H	O	1	1
13	M	Instalación estación transformadora	3	I, J	P	2	8
14	N	Instalación tuberías depósito	9	K	Q	2	4
15	O	Instalación conducciones subterráneas	8	L	Q	1	8
16	P	Conexión red general	5	D, M	-	1	4
17	Q	Conexión tuberías	2	N, O	-	2	2



# Ejemplo 1. Datos y cotas inferiores (2)

*Fechas mínimas de inicio*

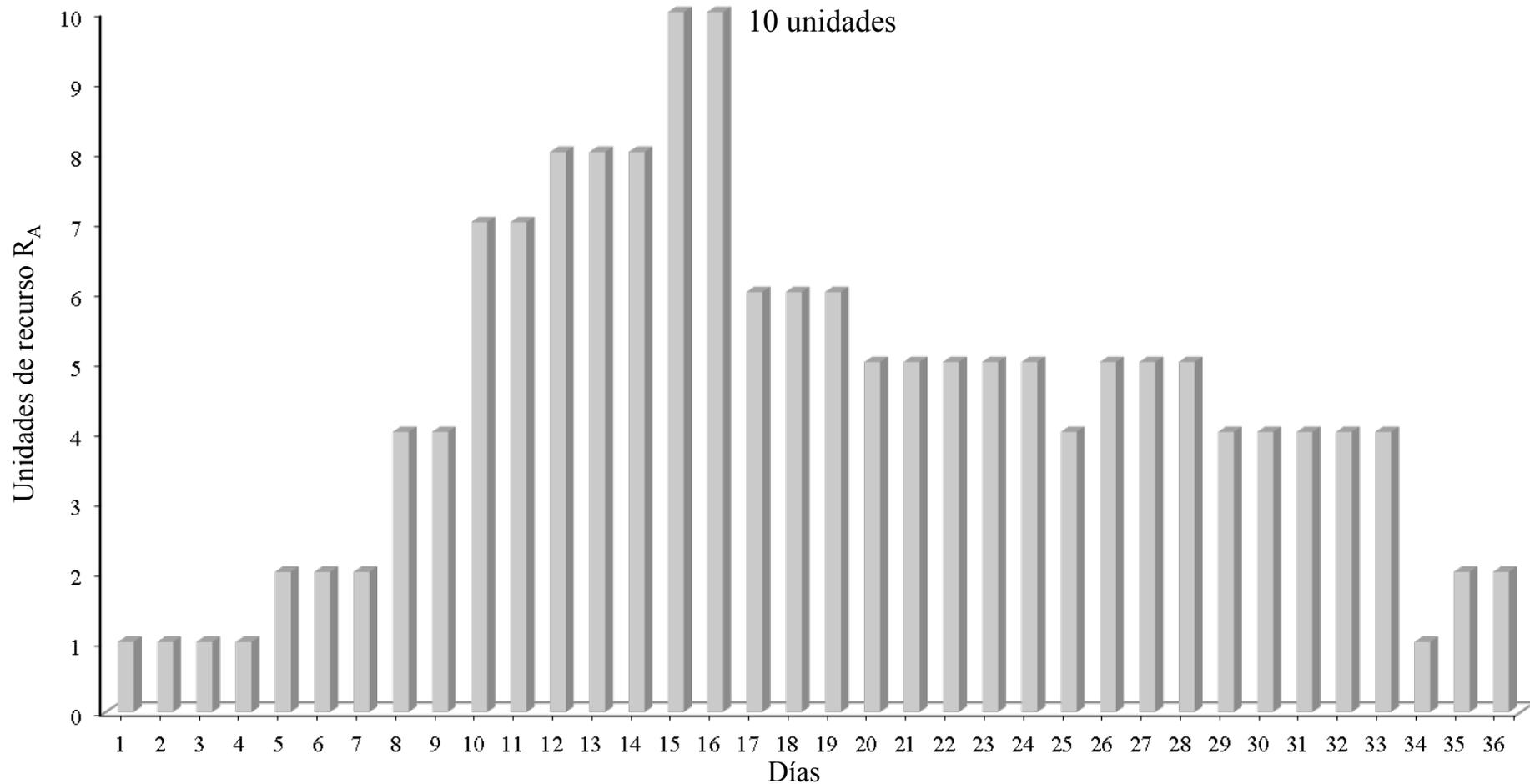


$j$	<i>Cod.</i>	$p_j$	$s_j^{\min}$	$j$	<i>Cod.</i>	$p_j$	$s_j^{\min}$
1	A	4	0	10	J	6	19
2	B	3	4	11	K	10	14
3	C	2	7	12	L	2	24
4	D	7	9	13	M	3	25
5	E	2	9	14	N	9	24
6	F	10	9	15	O	8	26
7	G	5	9	16	P	5	28
8	H	15	9	17	Q	2	34
9	I	5	11				



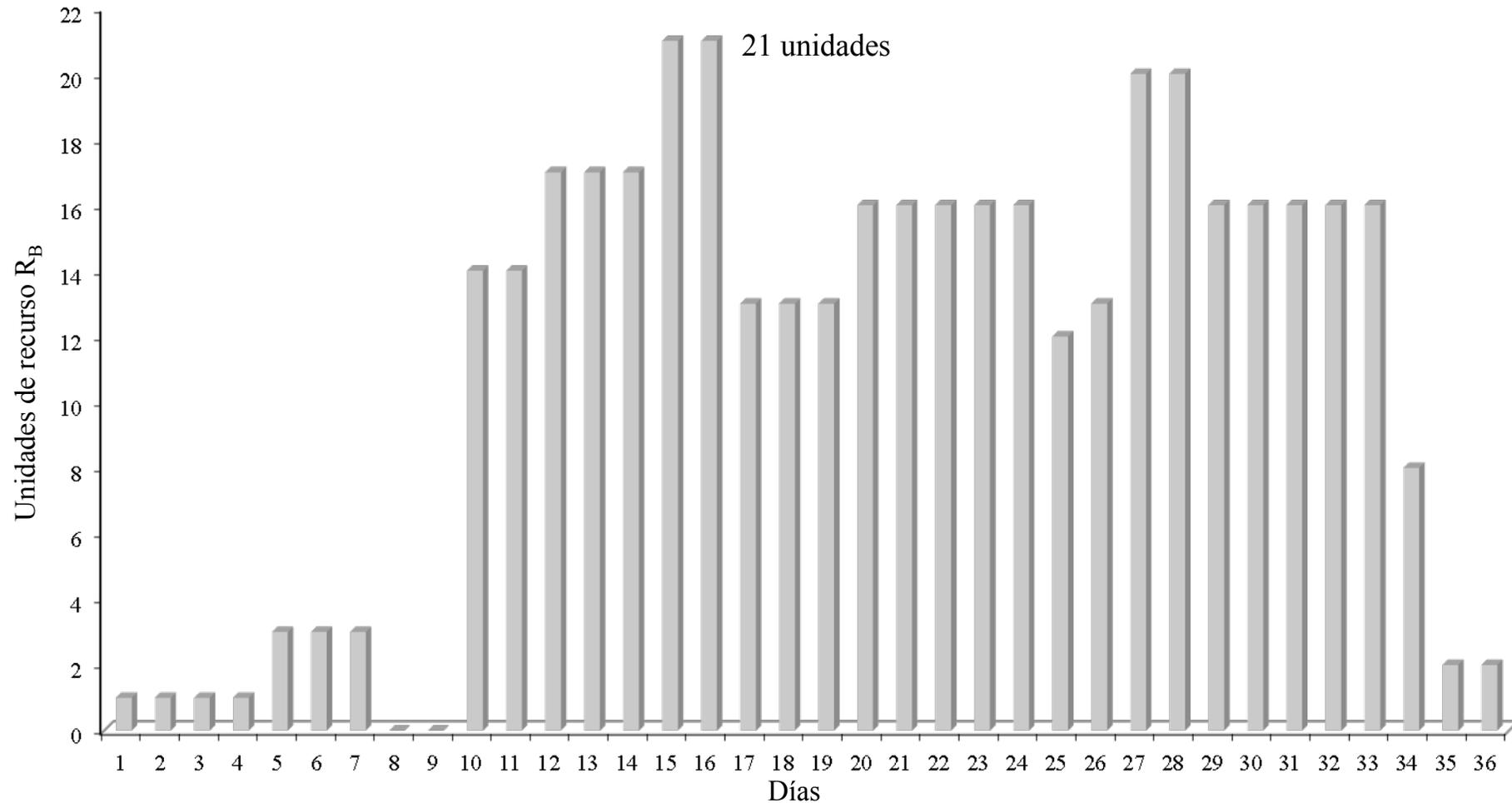
## Ejemplo 1. Datos y cotas inferiores (3)

*Curva de carga del recurso  $R_A$  con fechas mínimas de inicio. Utilización de  $R_A$*



## Ejemplo 1. Datos y cotas inferiores (4)

*Curva de carga del recurso  $R_B$  con fechas mínimas de inicio. Utilización de  $R_B$*



## Ejemplo 1. Datos y cotas inferiores (5)

$j$	Código	Descripción	$p_j$ (ut)	$r_{j,A}$	$r_{j,B}$
1	A	Despejar emplazamiento	4	1	1
2	B	Medición y replanteo	3	2	3
3	C	Explanación	2	4	-
4	D	Preparación acometida eléctrica	7	2	4
5	E	Excavación conducciones eléctricas	2	1	1
6	F	Excavación desagües	10	2	4
7	G	Cimientos depósito agua	5	1	3
8	H	Perforación pozo	15	1	2
9	I	Instalación conducciones eléctricas	5	2	4
10	J	Instalación tuberías desagües	6	1	7
11	K	Construcción de depósito agua	10	3	7
12	L	Instalación Bomba	2	1	1
13	M	Instalación estación transformadora	3	2	8
14	N	Instalación tuberías depósito	9	2	4
15	O	Instalación conducciones subterráneas	8	1	8
16	P	Conexión red general	5	1	4
17	Q	Conexión tuberías	2	2	2

$$LB_1 = T^{\min} = 36 \text{ ut}$$

$$R_A(t) = 5, R_B(t) = 15 \forall t$$

$$LB(A) = \frac{163}{5} = 32.6 \text{ ut}$$

$$LB(B) = \frac{410}{15} = 27.3 \text{ ut}$$

$$LB_2 = \max\{32.6, 27.3\}$$

$$LB = \max\{36, 32.6\}$$

$$LB = 36 \text{ ut}$$



## Ejemplo 2. Enunciado

Un taller dispone de 2 máquinas idénticas para pulir tres discos distintos, A, B y C por ambas caras. Una pulidora procesar una sola cara de un solo disco a la vez. Los tiempos de preparación son despreciables, mientras que los de proceso son: 7, 10 y 15 minutos, por cada cara de un disco A, B y C, respectivamente.

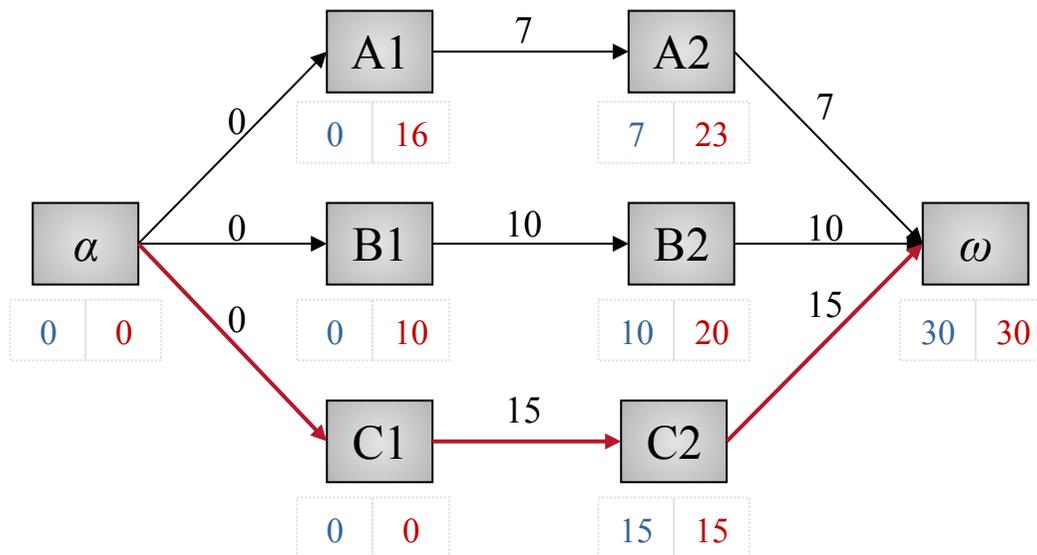
$j$	Código	Descripción	$p_j$ (ut)	$P_j$
1	A1	Pulir cara_1 del disco A	7	-
2	A2	Pulir cara_2 del disco A	7	1 (A1)
3	B1	Pulir cara_1 del disco B	10	-
4	B2	Pulir cara_2 del disco B	10	3 (B1)
5	C1	Pulir cara_1 del disco C	15	-
6	C2	Pulir cara_2 del disco C	15	5 (C1)

Establezca un programa de producción compatible con mínima ocupación del taller.



## Ejemplo 2. Cotas inferiores

- Diagrama de Roy



$$LB_1 = T^{\min} = 30 \text{ ut}$$

$$LB_2 = LB(Pul) = \frac{2 \cdot (7 + 10 + 15)}{2} = 32 \text{ ut}$$

$$LB = \max\{30, 32\}$$

$$LB = 32 \text{ ut}$$

# Problemas de compatibilidad. Algoritmos Greedy (1)

## *Concepto:*

Algoritmo constructivo orientado a resolver problemas con más de un recurso.

- Se parte de una LISTA ORDENADA de las tareas según un ÍNDICE DE PRIORIDAD.
- Se secuencian y temporizan las tareas en esquema serie o paralelo.

## *Esquema serie :*

- Se seleccionan las tareas ORDENADAMENTE y se localizan temporalmente en los instantes más tempranos respetando las restricciones de precedencia y la limitación de recursos.

## *Esquema paralelo:*

- En un instante de reloj, se lanzan ORDENADAMENTE las tareas cuyas precedentes han finalizado y con niveles de requerimiento de recursos que no superen a los disponibles.
- Se avanza el reloj hasta el menor instante en que pueden lanzarse nuevas tareas.

Cuando todas las actividades han sido programadas, se tiene una solución: PROGRAMA

- Las heurísticas anteriores no garantizan solución óptima.



## Problemas de compatibilidad. Algoritmos Greedy (2)

### Nomenclatura básica:

$J$	Conjunto de tareas
$K$	Conjunto de tipos de recurso
$p_j$	Tiempo de proceso de la tarea $j \in J$
$r_{j,k}$	Tasa de utilización del recurso $k \in K$ por parte de la actividad $j \in J$
$R_k(t)$	Nivel de disponibilidad del recurso $k \in K$ en $t \leq T$ ( $R_k = R_k(t) \forall t$ )
$F_j, F_j^*$	Conjuntos de tareas siguientes de $j \in J$ inmediatas y transitivas
$i \rightarrow j$	Indica: $i \in J$ es precedente inmediata de $j \in J$
$i \Rightarrow j$	Indica: $i \in J$ es precedente transitiva de $j \in J$
$s_j^{\min}$	Fecha mínima de inicio de $j \in J$ (Earliest Start Time: $EST_j$ )
$s_j^{\max}$	Fecha máxima de inicio de $j \in J$ (Latest Start Time: $LST_j$ )
$e_j^{\min}$	Fecha mínima de finalización de $j \in J$ (Earliest Finish Time: $EFT_j$ ) $\Leftrightarrow e_j^{\min} = s_j^{\min} + p_j$
$e_j^{\max}$	Fecha máxima de finalización de $j \in J$ (Latest Finish Time: $LFT_j$ ) $\Leftrightarrow e_j^{\max} = s_j^{\max} + p_j$
$\Delta s_j$	Margen total de $j \in J$ (compartido) $\Leftrightarrow \Delta s_j = s_j^{\max} - s_j^{\min}$
$T$	Duración del proyecto



# Problemas de compatibilidad. Algoritmos Greedy (3)

*Reglas de prioridad:*

Priorizar:  $j^* = \operatorname{argmin}_{j \in J} [v_\rho(j)]$  con  $\rho \in P$  (conjunto de reglas)

Regla-1 · Menor fecha mínima de inicio ·  $v_1(j) = s_j^{\min} \forall j \in J$ :

$$[(v_1(i) \leq v_1(j)) \Rightarrow (i \prec j)]$$

Regla-2 · Menor fecha máxima de inicio ·  $v_2(j) = s_j^{\max} \forall j \in J$ :

$$[(v_2(i) \leq v_2(j)) \Rightarrow (i \prec j)]$$

Regla-3 · Menor fecha mínima de finalización ·  $v_3(j) = e_j^{\min} \forall j \in J$ :

$$[(v_3(i) \leq v_3(j)) \Rightarrow (i \prec j)]$$

Regla-4 · Menor fecha máxima de finalización ·  $v_4(j) = e_j^{\max} \forall j \in J$ :

$$[(v_4(i) \leq v_4(j)) \Rightarrow (i \prec j)]$$

Regla-5 · Menor tiempo de proceso ·  $v_5(j) = p_j \forall j \in J$ :

$$[(v_5(i) \leq v_5(j)) \Rightarrow (i \prec j)]$$

Regla-6 · Mayor tiempo de proceso ·  $v_6(j) = -p_j \forall j \in J$ :

$$[(v_6(i) \leq v_6(j)) \Rightarrow (i \prec j)]$$

Regla-7 · Mayor número de siguientes inmediatas ·  $v_7(j) = -|F_j| \forall j \in J$ :

$$[(v_7(i) \leq v_7(j)) \Rightarrow (i \prec j)]$$

Regla-8 · Mayor número de siguientes transitivas ·  $v_8(j) = -|F_j^*| \forall j \in J$ :

$$[(v_8(i) \leq v_8(j)) \Rightarrow (i \prec j)]$$

Regla-9 · Menor utilización de recursos ·  $v_9(j) = p_j \sum_{k \in K} r_{j,k} R_k^{-1} \forall j \in J$ :

$$[(v_9(i) \leq v_9(j)) \Rightarrow (i \prec j)]$$

Regla-10 · Mayor utilización de recursos ·  $v_{10}(j) = -p_j \sum_{k \in K} r_{j,k} R_k^{-1} \forall j \in J$ :

$$[(v_{10}(i) \leq v_{10}(j)) \Rightarrow (i \prec j)]$$

Regla-11 · Hibridación ·  $v_{11}(j) = \sum_{\rho \in P} \alpha_\rho v_\rho(j) \forall j \in J$ . Ej.  $v_{11}(j) = \Delta s_j$

$$[(v_{11}(i) \leq v_{11}(j)) \Rightarrow (i \prec j)]$$

Regla-12 · Randomización ·  $v_{12}(j) \leftarrow RND(\ ) \forall j \in J$ :

$$[(v_{12}(i) \leq v_{12}(j)) \Rightarrow (i \prec j)]$$

Regla-13 · Regla de reglas ·  $\vec{\rho} = (\rho_1, \dots, \rho_{|J|})$  con  $\rho_n \in P \forall n = 1, \dots, |J|$

$$\pi(\vec{\rho}) = \operatorname{argmin}_{j \in J} [\vec{\rho}]$$



## Ejemplo 2. Problemas de compatibilidad. Reglas de prioridad

		5/-6	1	2	11	-7	3	4	9
$j$	Código	$p_j$	$s_j^{\min}$	$s_j^{\max}$	$\Delta s_j$	$ F_j $	$e_j^{\min}$	$e_j^{\max}$	$v_9(j)$
1	A1	7	0	16	16	1	7	23	3.5
2	A2	7	7	23	16	0	14	30	3.5
3	B1	10	0	10	10	1	10	20	5
4	B2	10	10	20	10	0	20	30	5
5	C1	15	0	0	0	1	15	15	7.5
6	C2	15	15	15	0	0	30	30	7.5

Priorizar:  $j^* = \operatorname{argmin}_{j \in J} [v_\rho(j)]$  con  $\rho \in P$  (conjunto de reglas)

Regla  $\cdot v_1(j) \triangleleft v_5(j)$      $s_j^{\min} \triangleleft p_j$  :     $\pi_1 = (A1, B1, C1, A2, B2, C2)$

Regla  $\cdot v_1(j) \triangleleft v_6(j)$      $s_j^{\min} \triangleleft -p_j$  :     $\pi_2 = (C1, B1, A1, A2, B2, C2)$

Regla  $\cdot v_2(j)$      $s_j^{\max}$  :     $\pi_3 = (C1, B1, C2, A1, B2, A2)$

Regla  $\cdot v_4(j) \triangleleft v_6(j)$      $e_j^{\max} \triangleleft -p_j$  :     $\pi_4 = (C1, B1, A1, C2, B2, A2)$

Regla  $\cdot v_{11}(j) \triangleleft v_7(j)$      $\Delta s_j \triangleleft -|F_j|$  :     $\pi_5 = (C1, C2, B1, B2, A1, A2)$

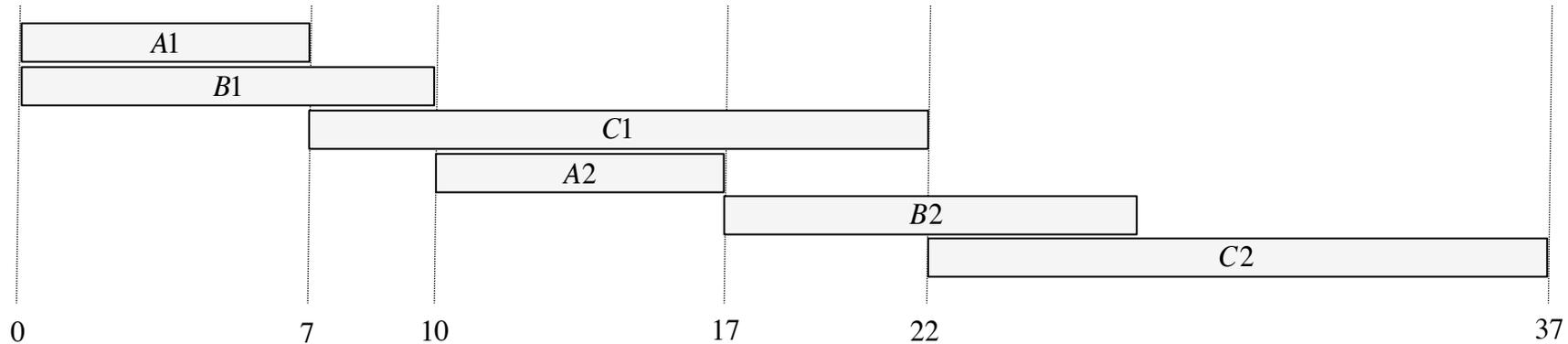
Regla  $\cdot v_{12}(j)$      $RND()$  :     $\pi_6 = (C1, A1, B1, A2, C2, B2)$

Vector-regla  $\cdot \vec{\rho} = (s_j^{\max}, s_j^{\max}, p_j, RND, \Delta s_j, p_j)$  :     $\pi_7 = (C1, B1, A1, B2, C2, A2)$

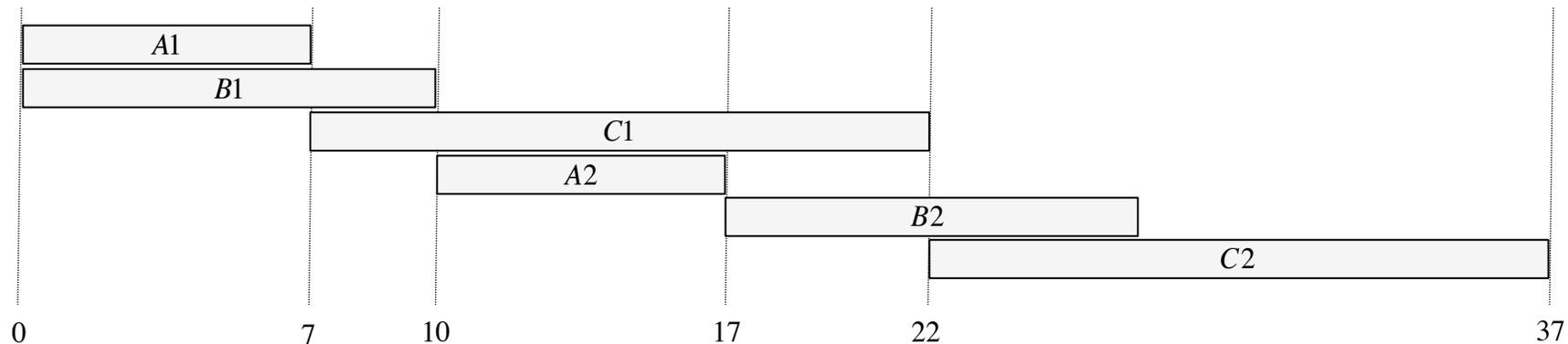


## Ejemplo 2. Problemas de compatibilidad. Algoritmos Greedy (1)

*Heurística serie:* Regla ·  $v_1(j) \triangleleft v_5(j)$   $s_j^{\min} \triangleleft p_j$  :  $\pi_1 = (A1, B1, C1, A2, B2, C2)$

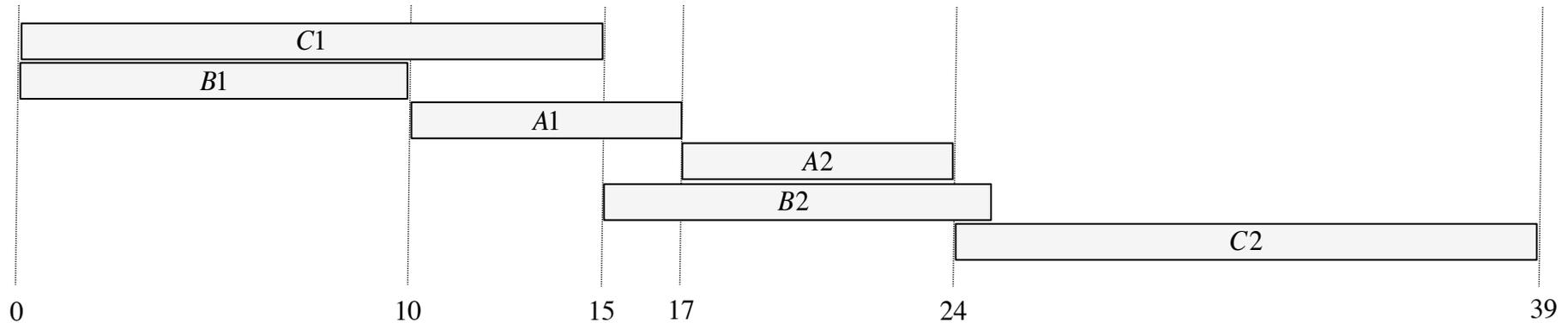


*Heurística paralelo:* Regla ·  $v_1(j) \triangleleft v_5(j)$   $s_j^{\min} \triangleleft p_j$  :  $\pi_1 = (A1, B1, C1, A2, B2, C2)$

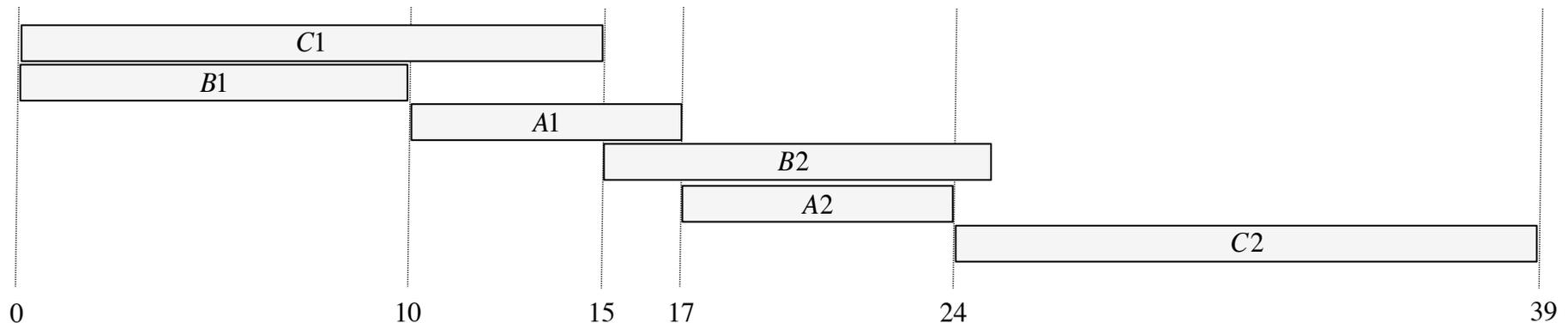


## Ejemplo 2. Problemas de compatibilidad. Algoritmos Greedy (2)

*Heurística serie:* Regla ·  $v_1(j) \triangleleft v_6(j)$   $s_j^{\min} \triangleleft -p_j$ :  $\pi_2 = (C1, B1, A1, A2, B2, C2)$

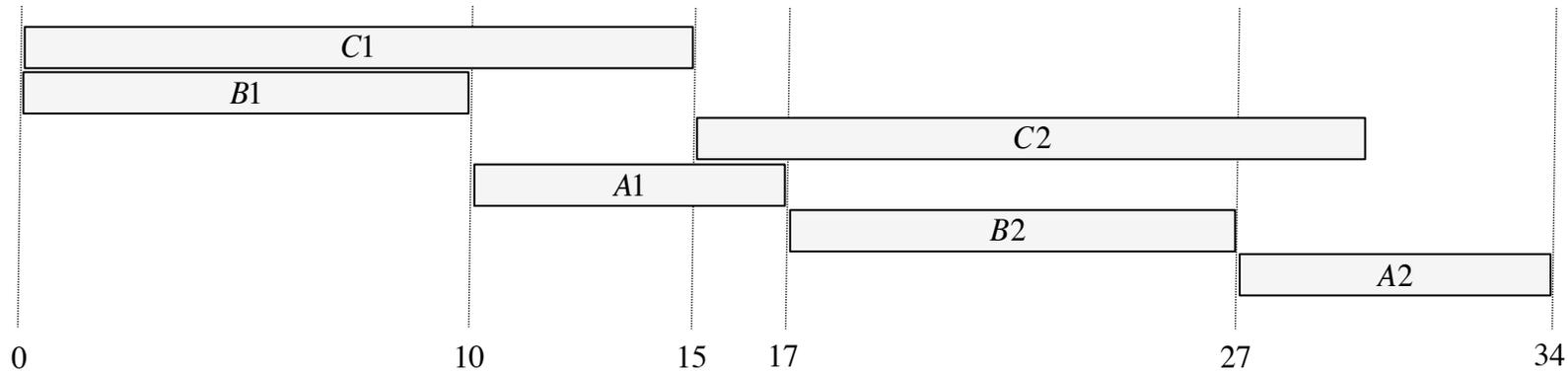


*Heurística paralelo:* Regla ·  $v_1(j) \triangleleft v_6(j)$   $s_j^{\min} \triangleleft -p_j$ :  $\pi_2 = (C1, B1, A1, A2, B2, C2)$

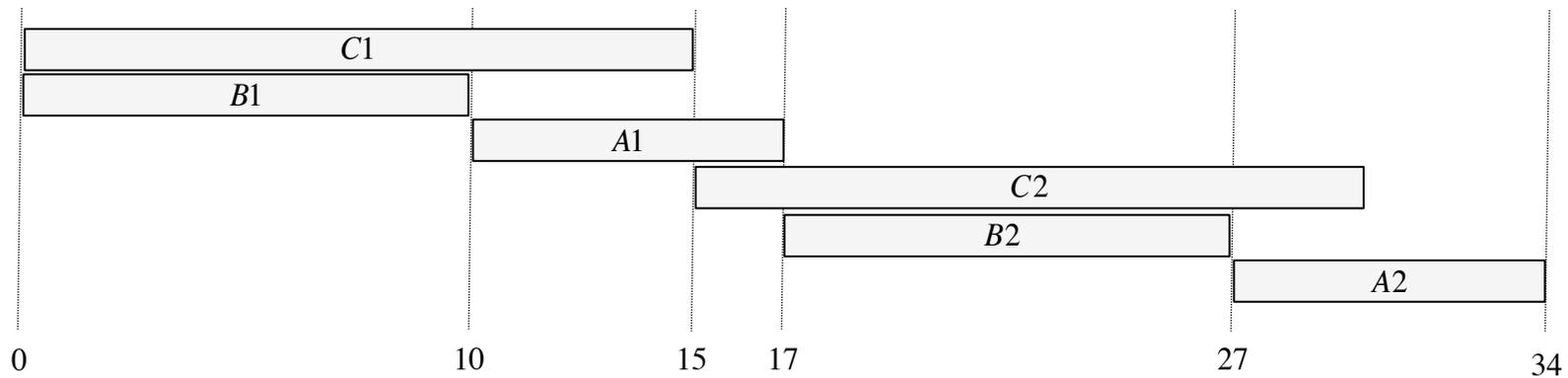


## Ejemplo 2. Problemas de compatibilidad. Algoritmos Greedy (3)

*Heurística serie:* Regla  $\cdot v_2(j)$   $s_j^{\max}$  :  $\pi_3 = (C1, B1, C2, A1, B2, A2)$

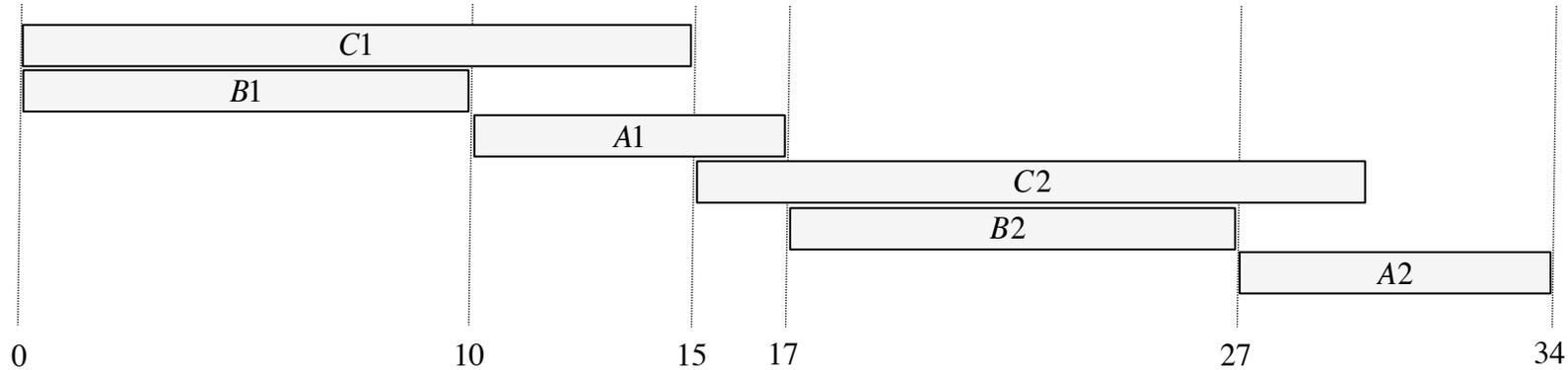


*Heurística paralelo:* Regla  $\cdot v_2(j)$   $s_j^{\max}$  :  $\pi_3 = (C1, B1, C2, A1, B2, A2)$

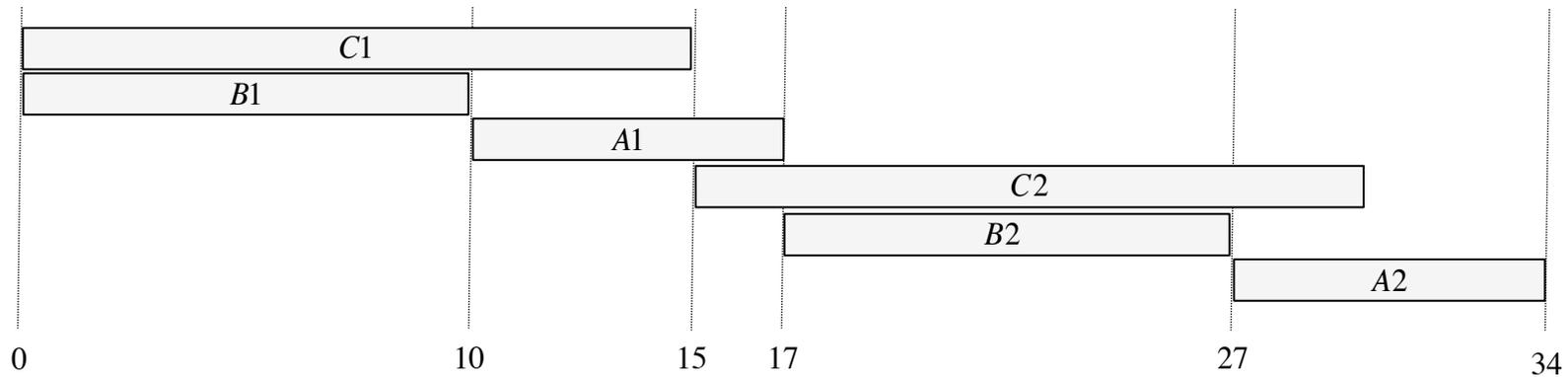


## Ejemplo 2. Problemas de compatibilidad. Algoritmos Greedy (4)

*Heurística serie:* Regla ·  $v_4(j) \triangleleft v_6(j)$   $e_j^{\max} \triangleleft -p_j$ :  $\pi_4 = (C1, B1, A1, C2, B2, A2)$

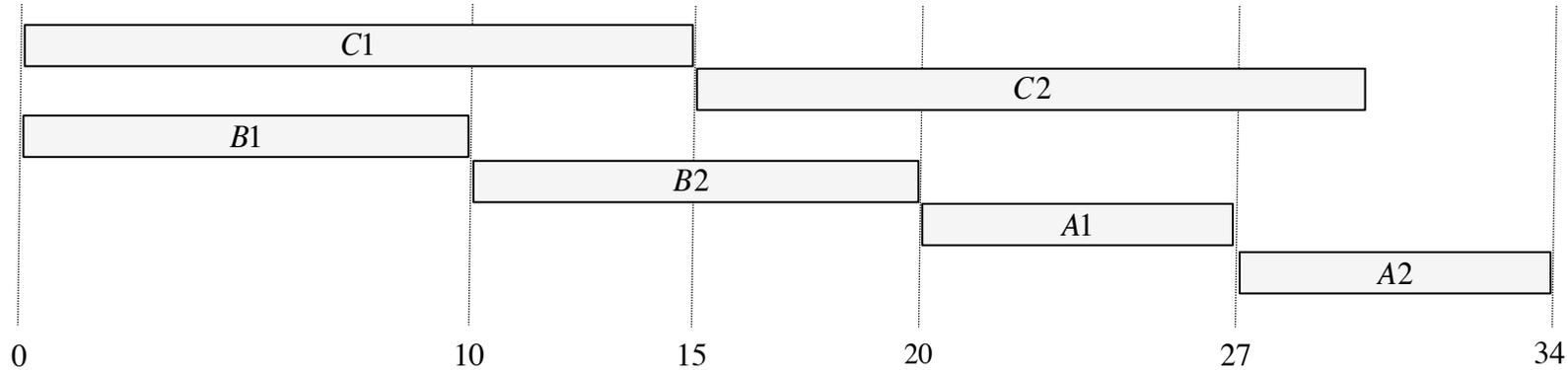


*Heurística paralelo:* Regla ·  $v_4(j) \triangleleft v_6(j)$   $e_j^{\max} \triangleleft -p_j$ :  $\pi_4 = (C1, B1, A1, C2, B2, A2)$

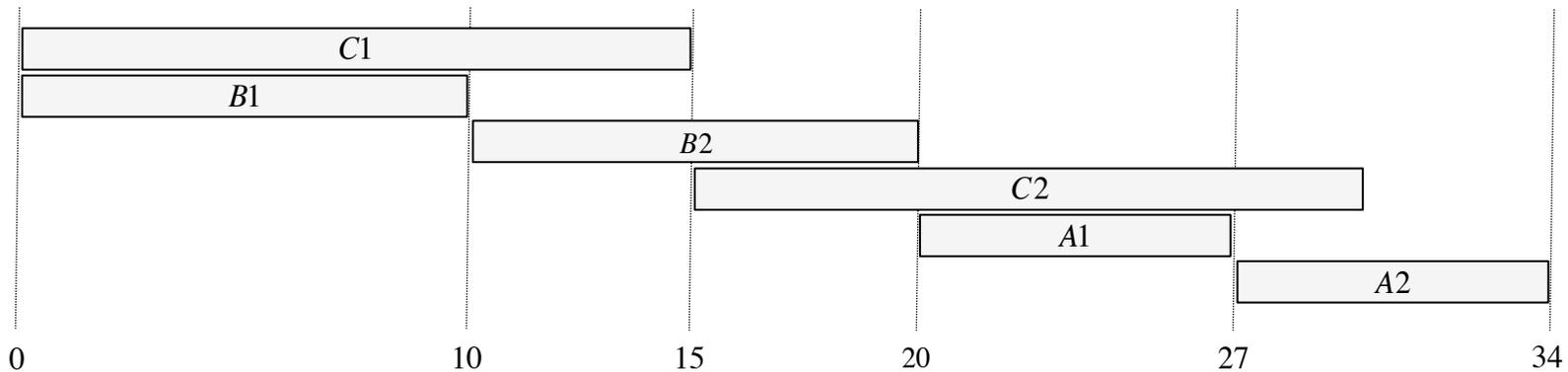


## Ejemplo 2. Problemas de compatibilidad. Algoritmos Greedy (5)

*Heurística serie:* Regla ·  $v_{11}(j) \triangleleft v_7(j)$   $\Delta s_j \triangleleft -|F_j|$ :  $\pi_5 = (C1, C2, B1, B2, A1, A2)$

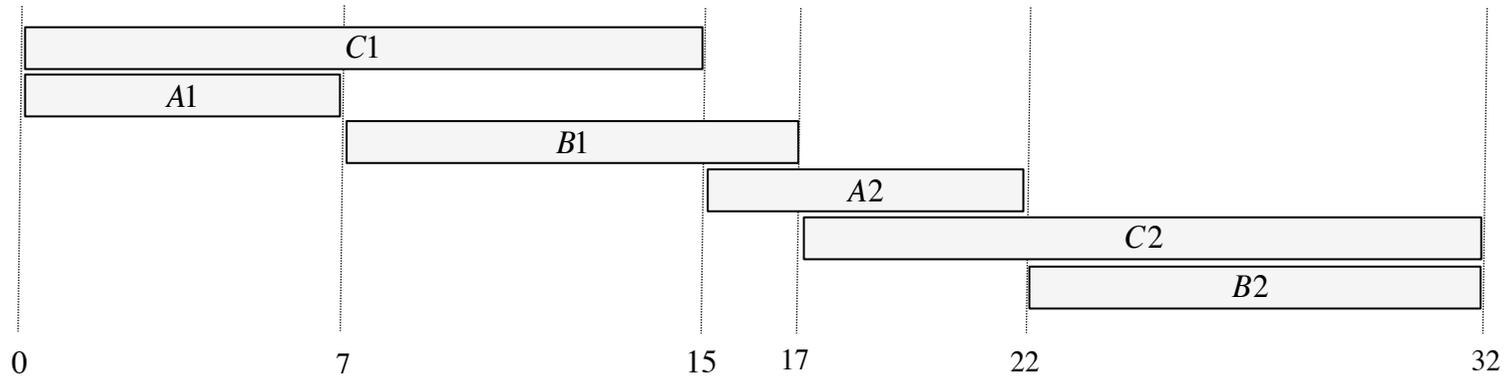


*Heurística paralelo:* Regla ·  $v_{11}(j) \triangleleft v_7(j)$   $\Delta s_j \triangleleft -|F_j|$ :  $\pi_5 = (C1, C2, B1, B2, A1, A2)$

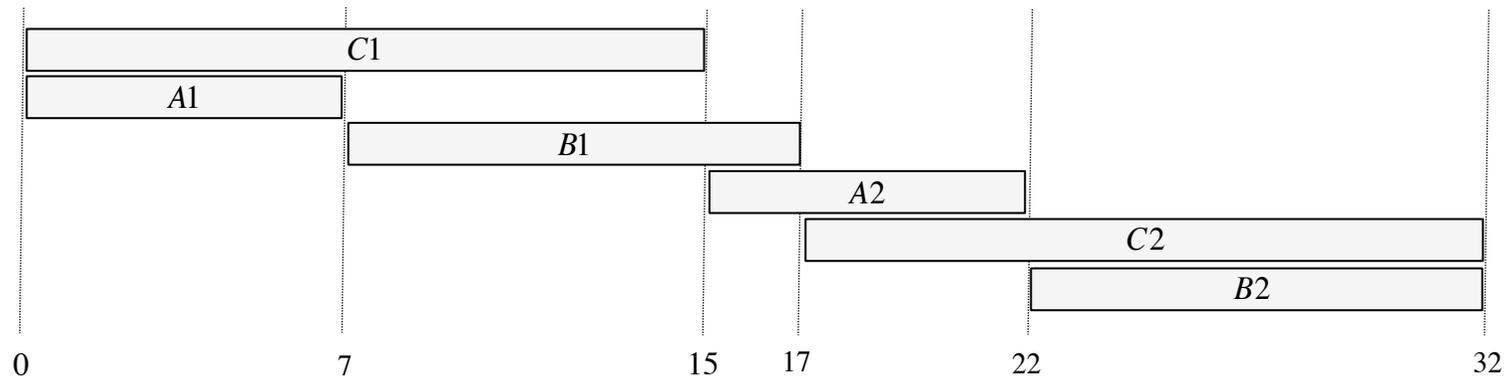


## Ejemplo 2. Problemas de compatibilidad. Algoritmos Greedy (6)

*Heurística serie:* Regla  $\cdot v_{12}(j)$   $RND()$ :  $\pi_6 = (C1, A1, B1, A2, C2, B2)$

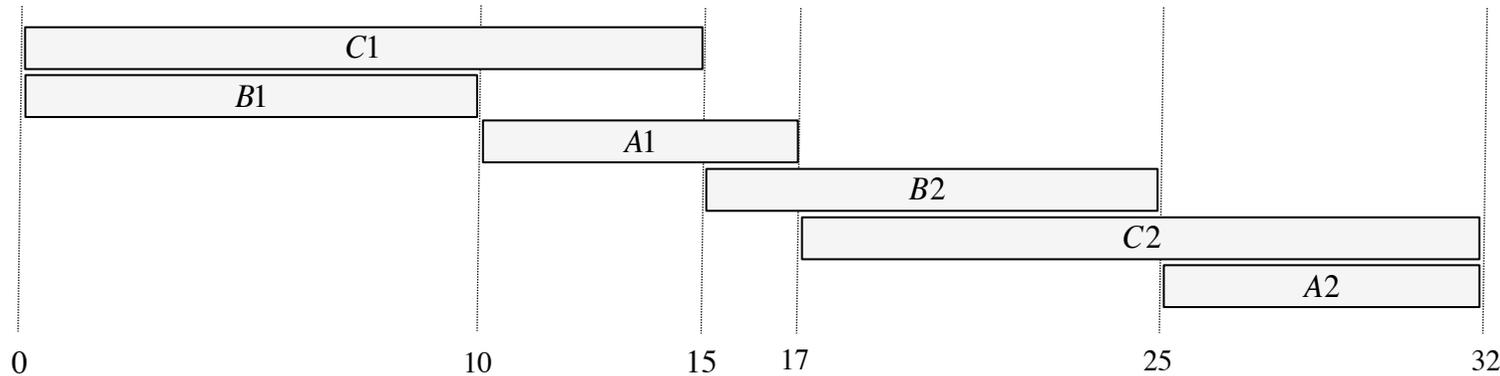


*Heurística paralelo:* Regla  $\cdot v_{12}(j)$   $RND()$ :  $\pi_6 = (C1, A1, B1, A2, C2, B2)$

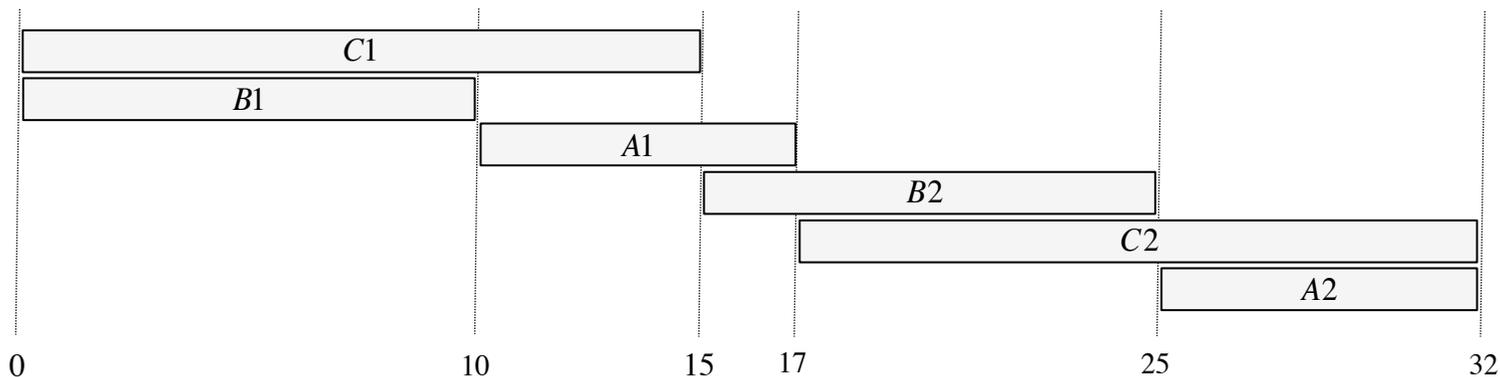


## Ejemplo 2. Problemas de compatibilidad. Algoritmos Greedy (7)

*Heurística serie:* Vector-regla ·  $\vec{\rho} = (s_j^{\max}, s_j^{\max}, p_j, RND, \Delta s_j, p_j)$ :  $\pi_7 = (C1, B1, A1, B2, C2, A2)$



*Heurística paralelo:* Vector-regla ·  $\vec{\rho} = (s_j^{\max}, s_j^{\max}, p_j, RND, \Delta s_j, p_j)$ :  $\pi_7 = (C1, B1, A1, B2, C2, A2)$



# Problemas de equilibrado. Algoritmos Greedy

## *Concepto:*

Algoritmos de exploración de vecindarios orientados a resolver problemas con más de un recurso y con el objetivo de reducir la sobrecarga de recursos en un horizonte prefijado.

- Se parte de una LISTA CÍCLICA de las tareas ordenadas según un ÍNDICE DE PRIORIDAD.
- Se localiza el inicio de cada tarea en los periodos del INTERVALO ACOTADO por su fecha mínima y máxima de inicio y las restricciones de precedencia, siguiendo un esquema serie.

## *Esquema serie básico :*

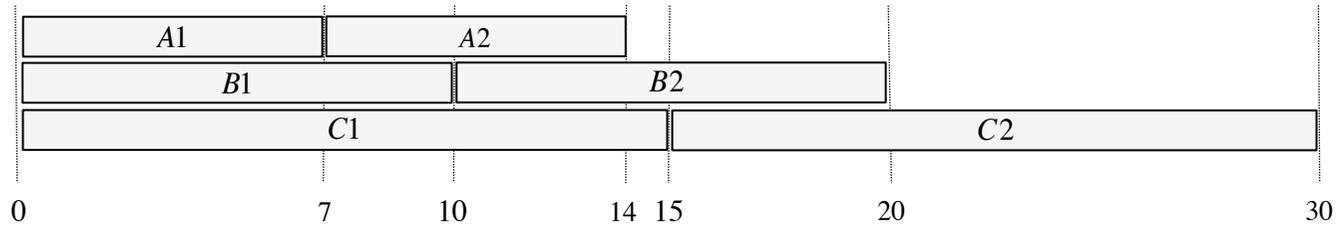
1. Seleccionar tarea de LISTA CÍCLICA y localizar su inicio en los periodos del INTERVALO ACOTADO, evaluando la sobrecarga de recursos en dichos periodos.
2. Fijar inicio de tarea en curso en el periodo del INTERVALO ACOTADO con máxima reducción de sobrecarga; en caso de empate, desplazar inicio de tarea a extremo del INTERVALO ACOTADO.
3. Actualizar: sobrecarga total y extremos del INTERVALO ACOTADO de cada tarea.
4. Si hay cambios en Paso\_3 (o ciclo abierto en LISTA CÍCLICA), Ir a Paso\_1; Si \_no, Finalizar

Nota: Las heurísticas anteriores no garantizan solución óptima.



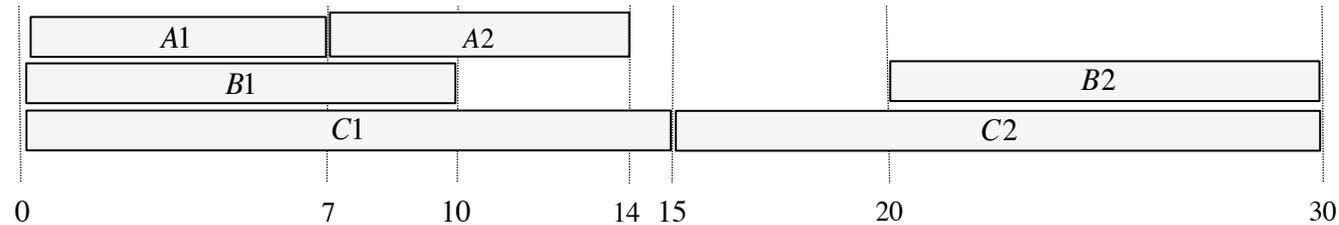
## Ejemplo 2. Problemas de equilibrado. Algoritmos Greedy (1)

Heurística serie:  $T = 30$ . Regla  $\cdot v_{11}(j) \triangleleft v_7(j) : \Delta s_j \triangleleft -|F_j| \Rightarrow LC = \{C1, C2, B1, B2, A1, A2\}$



$$LC = \{C1, C2, B1, B2, A1, A2\}$$

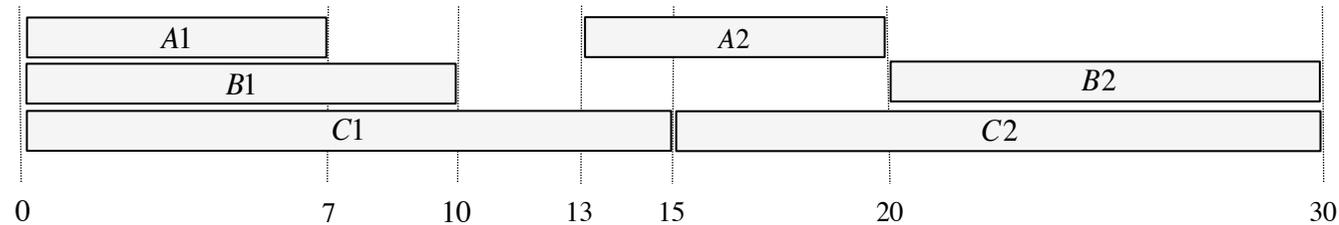
$$W_{pul} = 14$$



$C1 \wedge C2 \wedge B1$  (Block)

$$B2(s_{B2} \rightarrow 20)$$

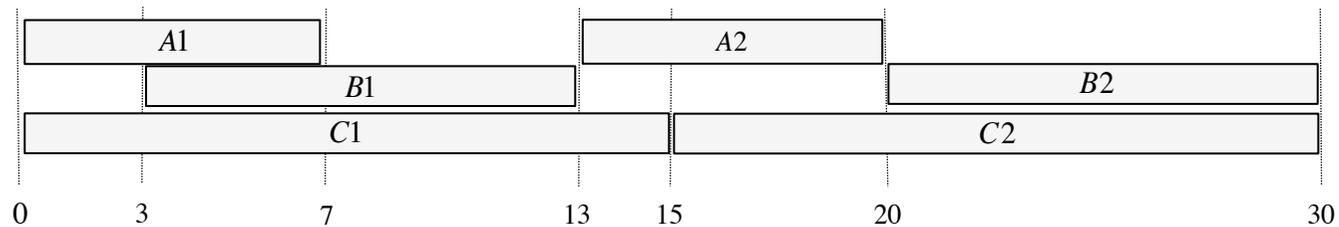
$$W_{pul} = 10$$



$A1$  (Block)

$$A2(s_{A2} \rightarrow 13)$$

$$W_{pul} = 7$$



$C1 \wedge C2$  (Block)

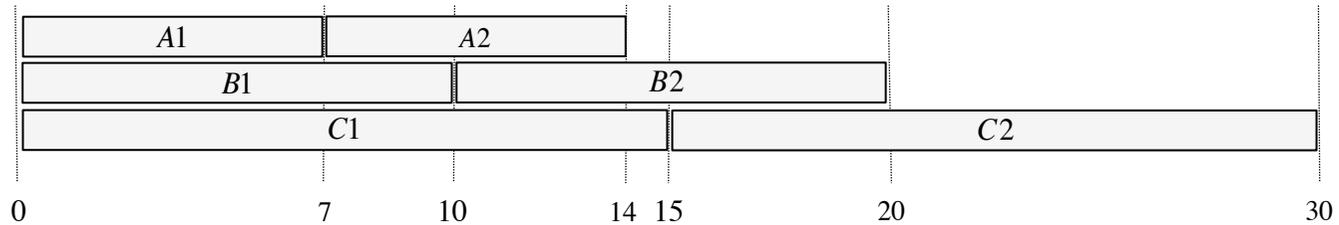
$$B1(s_{B1} \rightarrow 3)$$

$$W_{pul} = 4$$



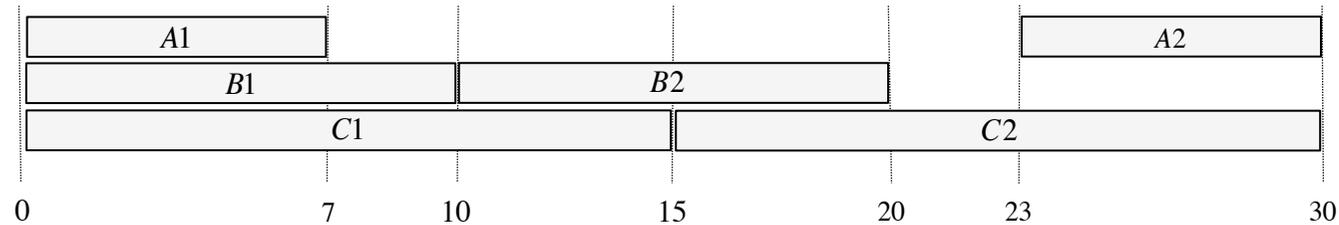
## Ejemplo 2. Problemas de equilibrado. Algoritmos Greedy (2)

Heurística serie:  $T = 30$ . Regla  $\cdot -v_7(j) \triangleleft -v_{11}(j) : |F_j| \triangleleft -\Delta s_j \Rightarrow LC = \{A2, B2, C2, A1, B1, C1\}$



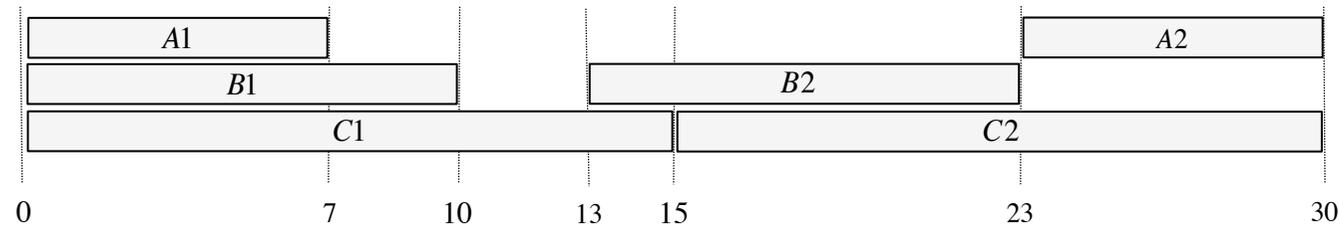
$$LC = \{A2, B2, C2, A1, B1, C1\}$$

$$W_{pul} = 14$$



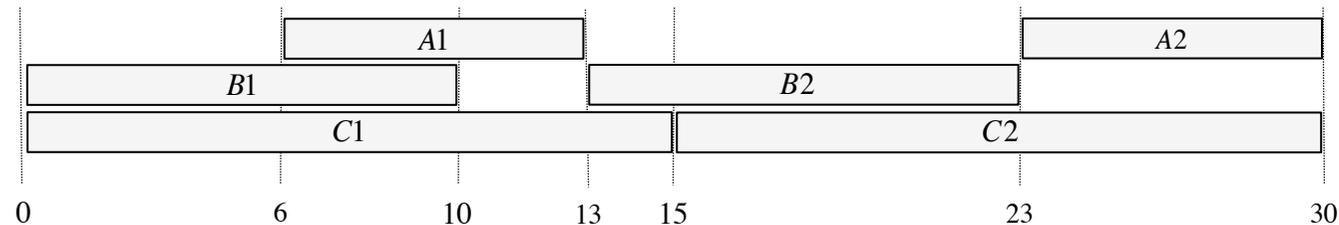
$$A2(s_{A2} \rightarrow 23)$$

$$W_{pul} = 7$$



$$B2(s_{B2} \rightarrow 13)$$

$$W_{pul} = 7$$



$$C2(block)$$

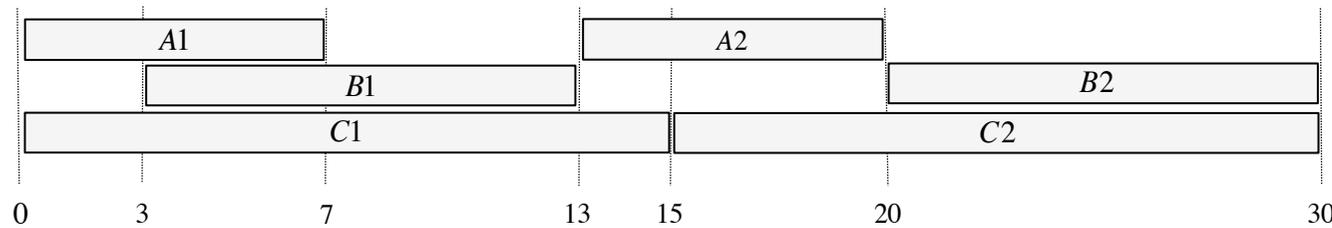
$$A1(s_{A1} \rightarrow 6)$$

$$W_{pul} = 4$$



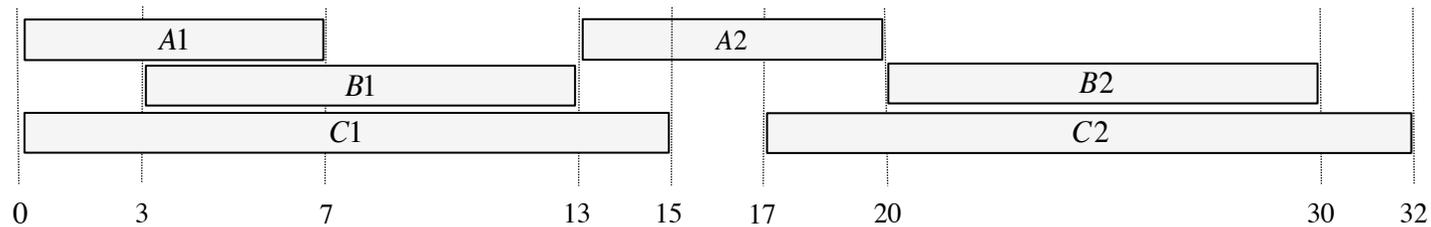
## Ejemplo 2. Problemas de equilibrado. Algoritmos Greedy (3)

*Heurística serie:*  $T = 32$ . Regla  $\cdot v_{11}(j) \triangleleft v_7(j) : \Delta s_j \triangleleft -|F_j| \Rightarrow LC = \{C1, C2, B1, B2, A1, A2\}$



$$LC = \{C1, C2, B1, B2, A1, A2\}$$

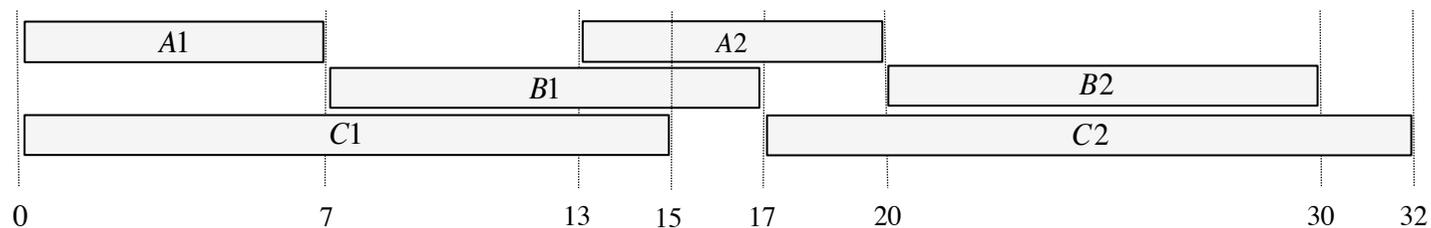
$$W_{pul} = 4$$



$C1$  (Block)

$C2(s_{C2} \rightarrow 17)$

$$W_{pul} = 4$$



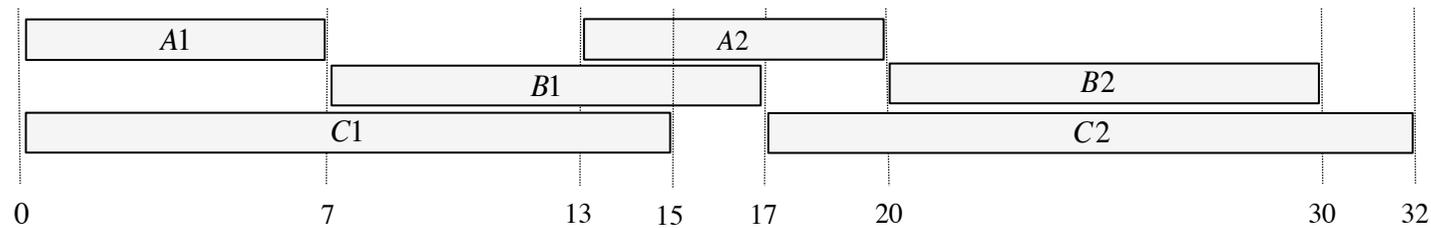
$B1(s_{B1} \rightarrow 7)$

$$W_{pul} = 2$$

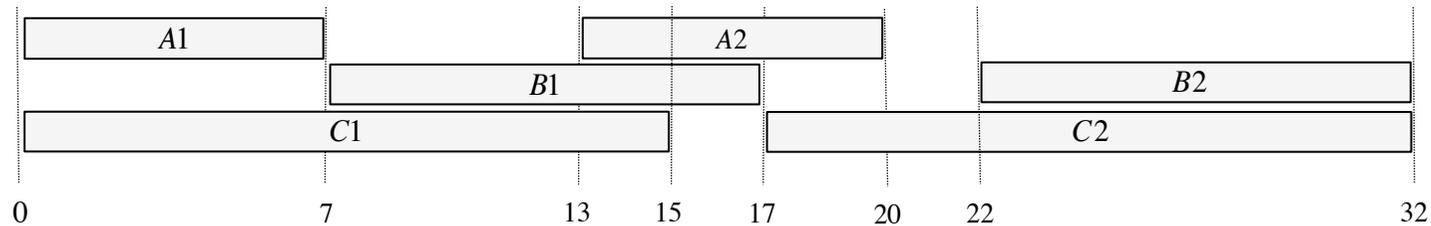


## Ejemplo 2. Problemas de equilibrado. Algoritmos Greedy (4)

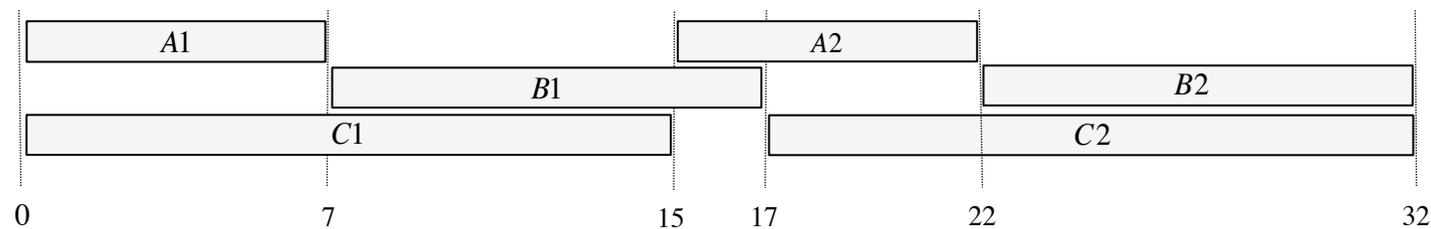
*Heurística serie:*  $T = 32$ . Regla  $\cdot v_{11}(j) \triangleleft v_7(j) : \Delta s_j \triangleleft -|F_j| \Rightarrow LC = \{C1, C2, B1, B2, A1, A2\}$



$B1(s_{B1} \rightarrow 7)$   
 $W_{pul} = 2$



$B2(s_{B2} \rightarrow 22)$   
 $W_{pul} = 2$



$A1(W_{pul} \uparrow)$   
 $A2(s_{A2} \rightarrow 15)$   
 $W_{pul} = 0$

## En la Atenas de Pericles

“Fidias se centró en la decoración escultórica del conjunto, mientras de la construcción se encargaban dos arquitectos, Ictino y Calícrates; el ingeniero romano Vitrubio, que escribió cuatro siglos más tarde, menciona a un tercer arquitecto llamado Carpión del que no tenemos más noticias.

No sabemos el tipo de relación que mantenían los arquitectos y la forma en que se ocupaban de los trabajos. Las obras necesitaron, además, gentes dedicadas a los más variados oficios: canteros, albañiles, carpinteros, doradores, pintores, escultores, herreros, modeladores de cera, transportistas y operadores de poleas.

Sabemos por las inscripciones que los trabajadores eran ciudadanos de Atenas, metecos (extranjeros con carta de residencia) y esclavos; y que todos cobraban lo mismo por el mismo trabajo. Las labores especializadas se retribuían a razón de un dracma por día. Por sorprendente que nos parezca, los arquitectos cobraban un dracma también, a pesar de su responsabilidad.”

Murcia, F.J. (2012) La construcción del Partenón. Partenón, el gran templo de Atenea.  
*Historia - National Geographic - España*, n. 104.

