

41° Jornadas Argentinas de Informática
1° Simposio Argentino de Informática Industrial

Planificación de la producción de corto plazo en plantas “batch” multiproducto, multietapa: Un enfoque CP novedoso

Franco M. Novara - Juan M. Novas - Gabriela P. Henning
INTEC (UNL, CONICET) – Santa Fe, Argentina



La Plata, 28 de Agosto de 2012

➤ Motivación

➤ Problema

- Definición del problema
- Complejidad matemática
- Por qué un modelo CP?

➤ Modelo

- Modelo básico
- Extensión al modelo básico I: recursos renovables
- Extensión al modelo básico II: equipos de limpieza limitados

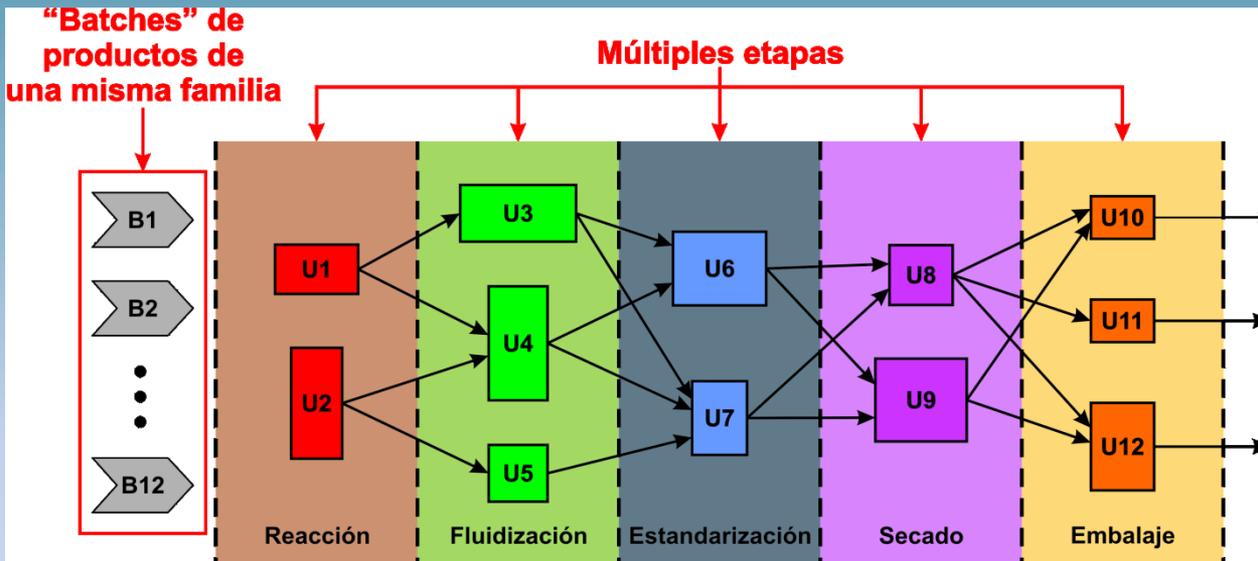
➤ Resultados

- Ejemplos abordados
- Resumen de resultados

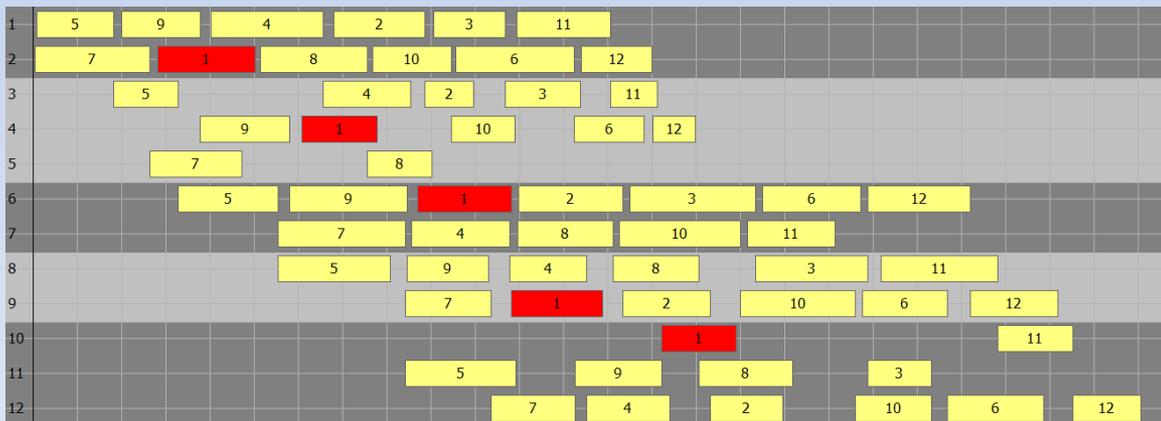
➤ Conclusiones - Trabajos futuros

Ambientes productivos “batch” multiproducto, multietapa

Diversas industrias operan en estos ambientes: química fina, farmacéutica, alimenticia, etc.



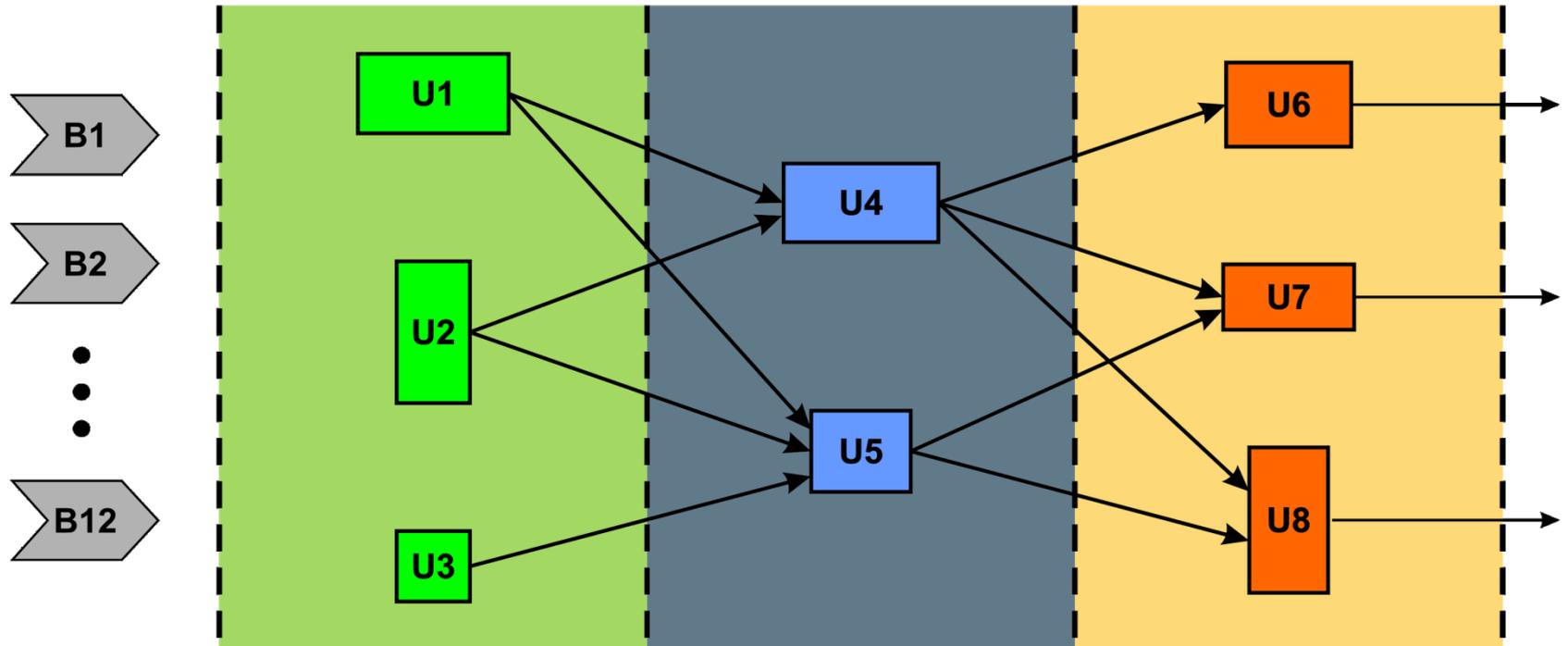
- Cada “batch” debe **atravesar** secuencialmente **todas las etapas**
- Cada “batch” se **asigna** a una unidad por etapa, lo que origina una **tarea**
- Las tareas deben secuenciarse y determinarse su tiempo de inicio y fin

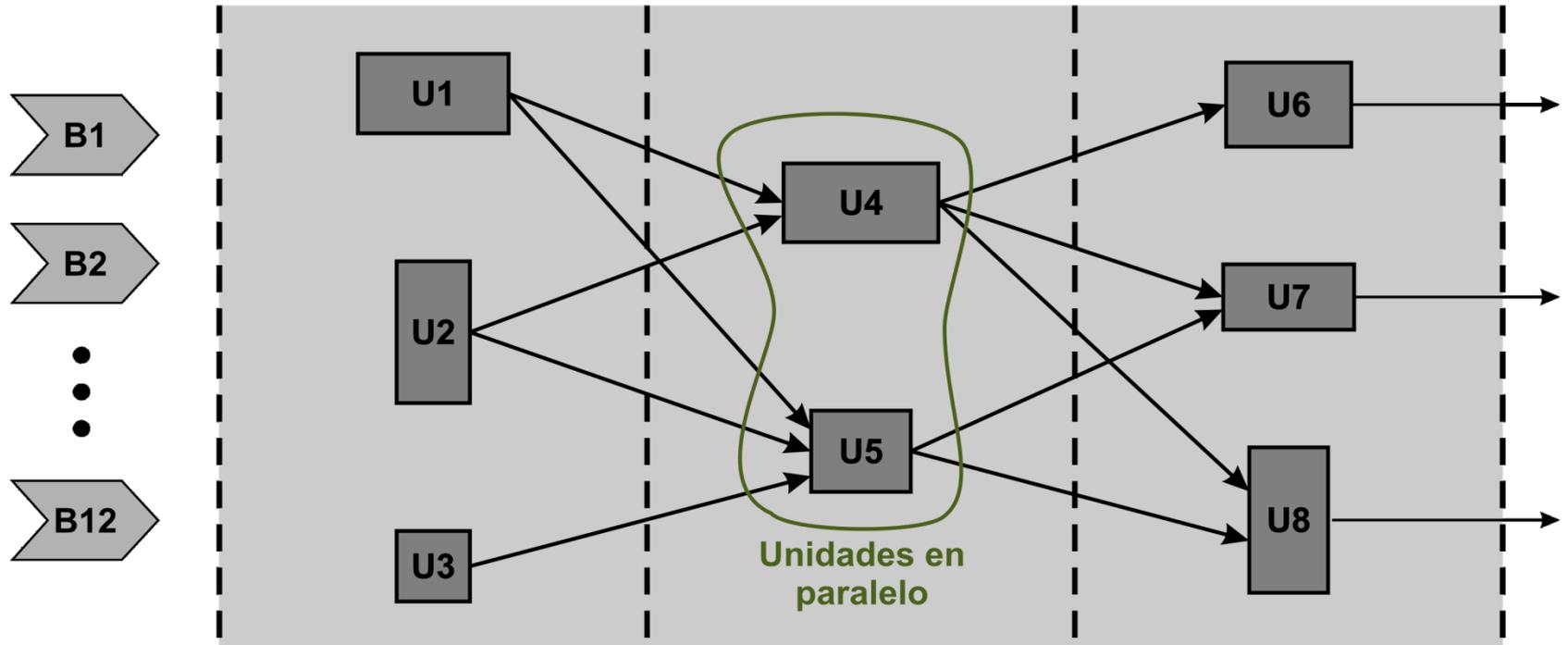


Objetivo:

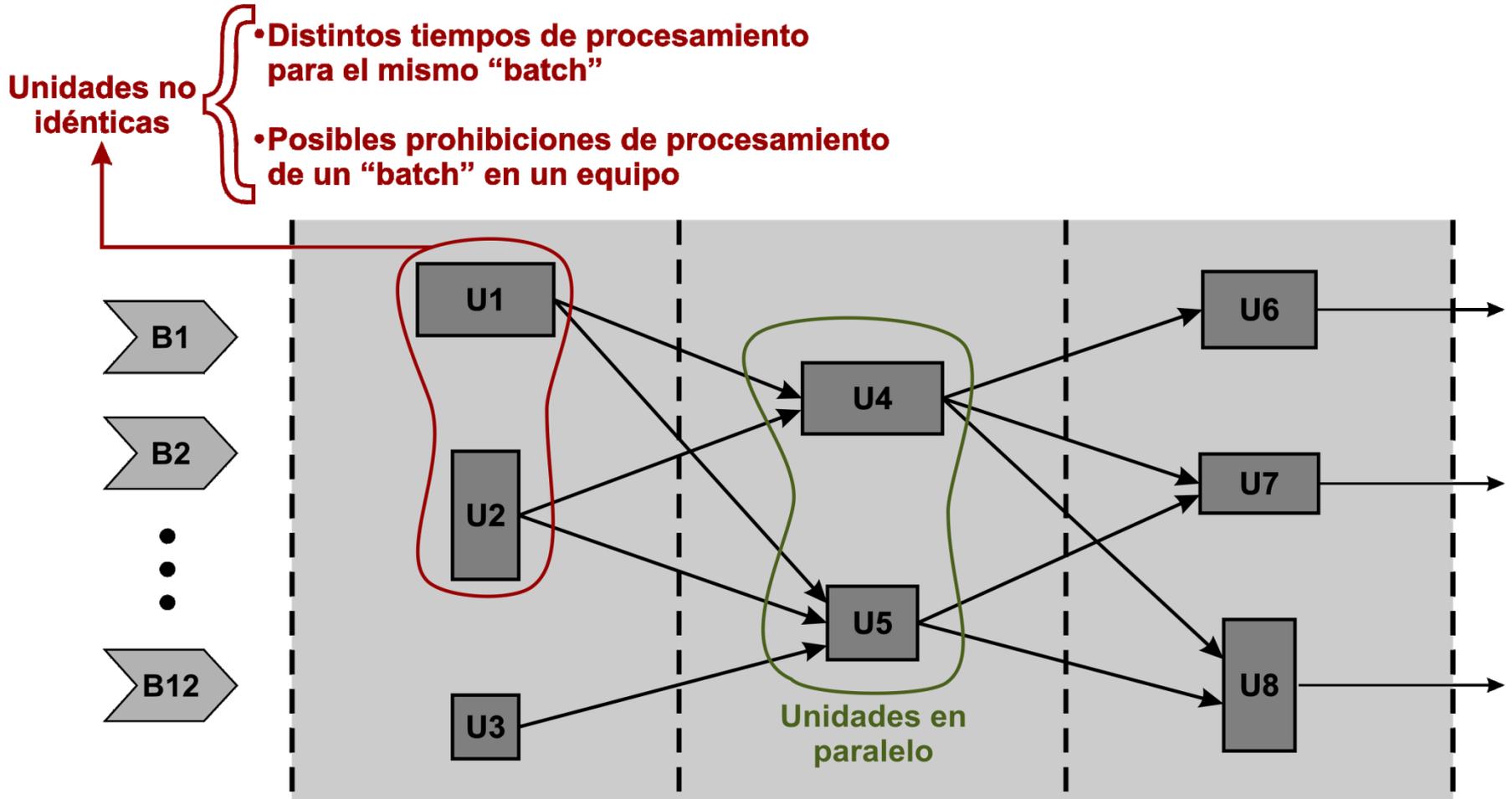
Crear una **agenda** de producción que **optimice una medida de desempeño**

Esquema de una planta "batch" multiproducto, multietapa

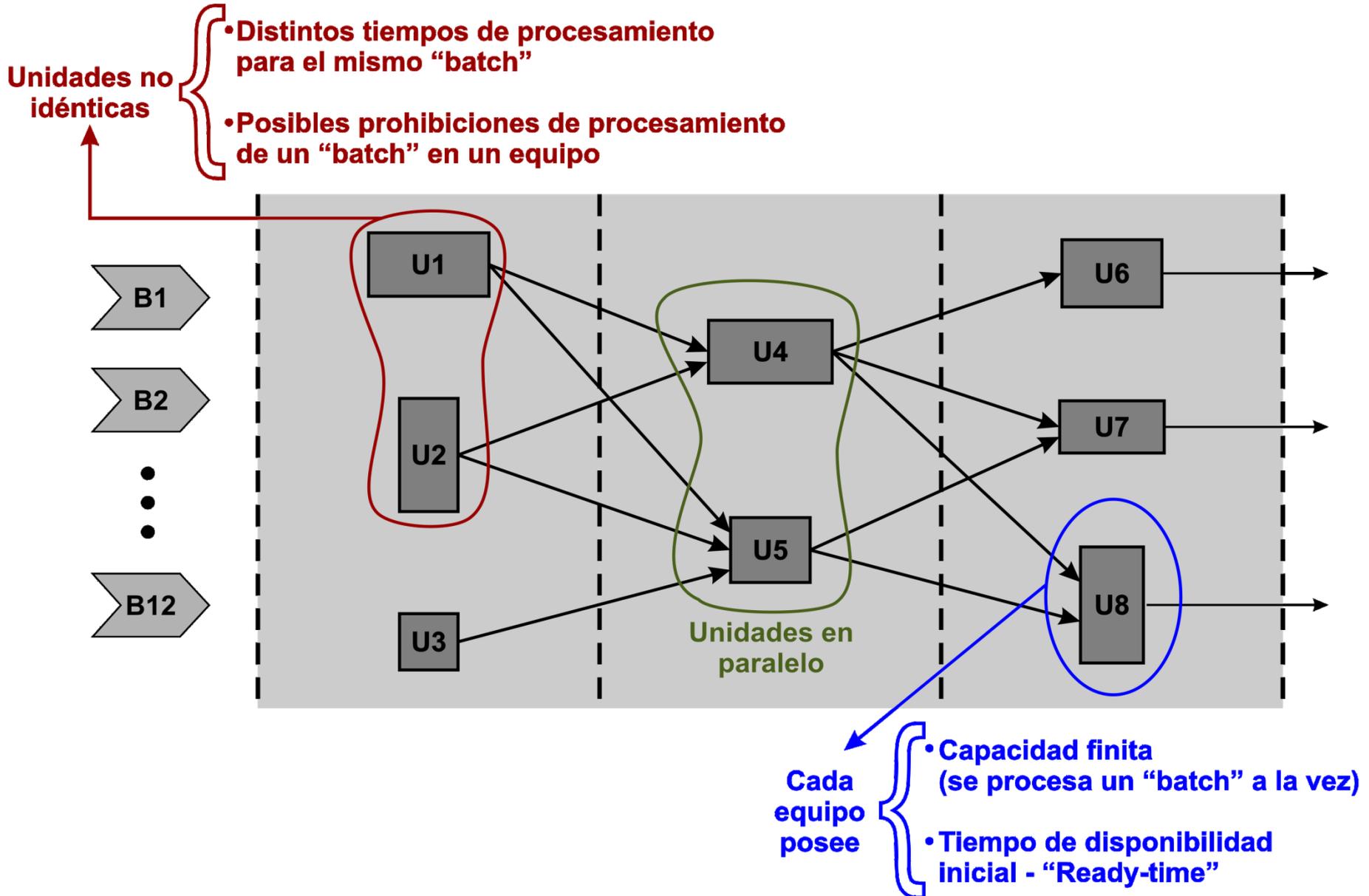




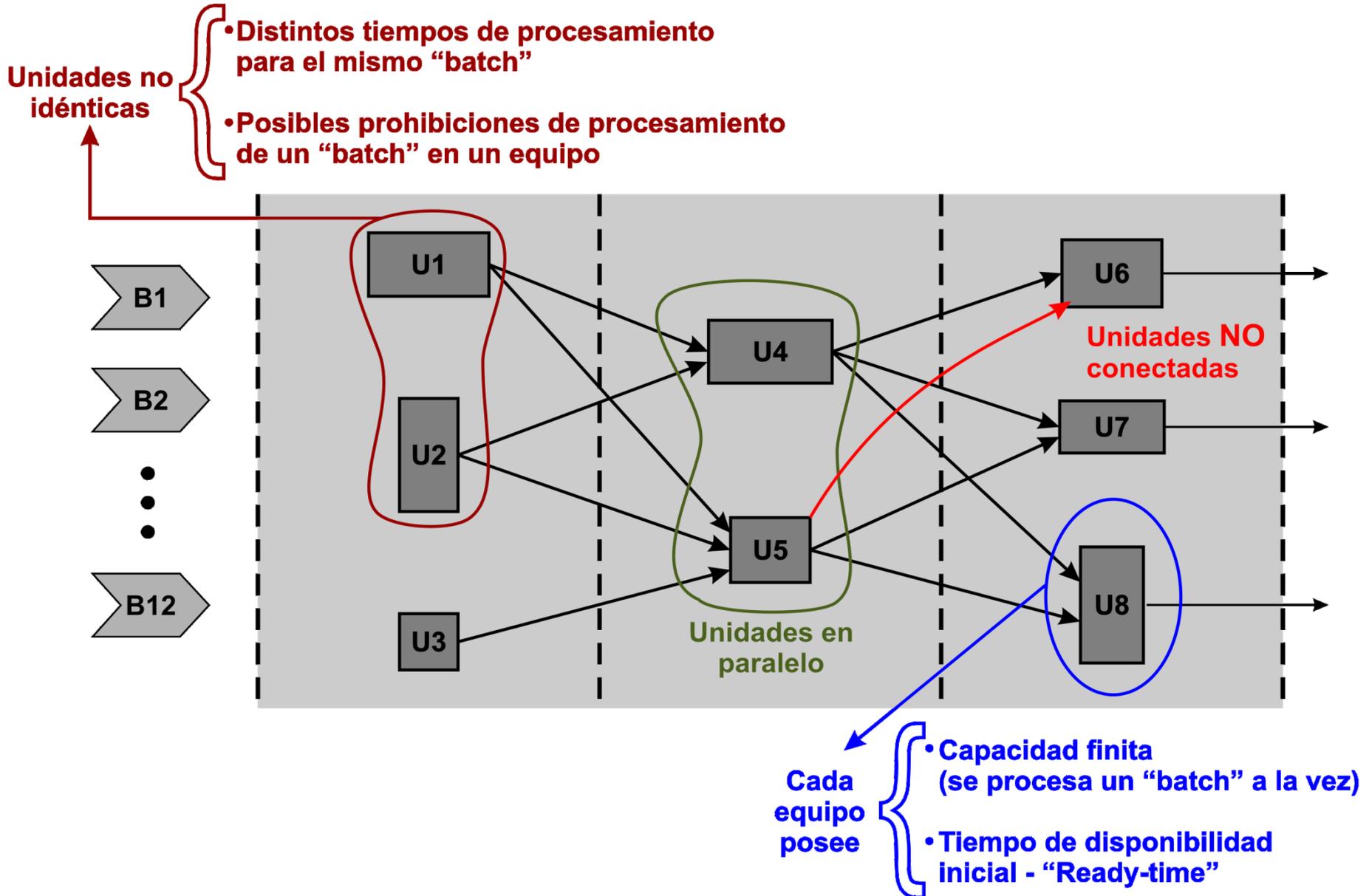
Definición del problema



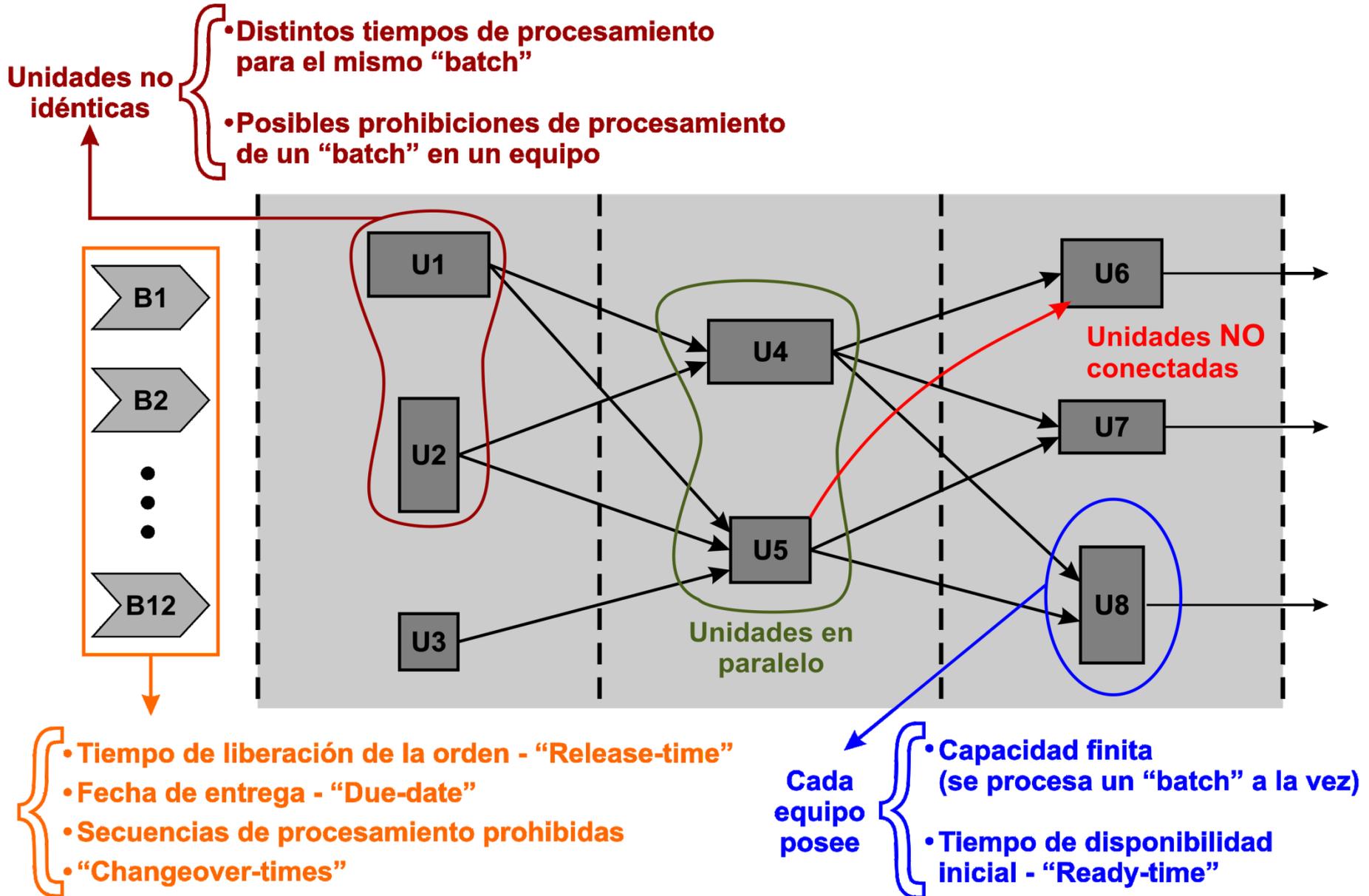
Definición del problema



Definición del problema

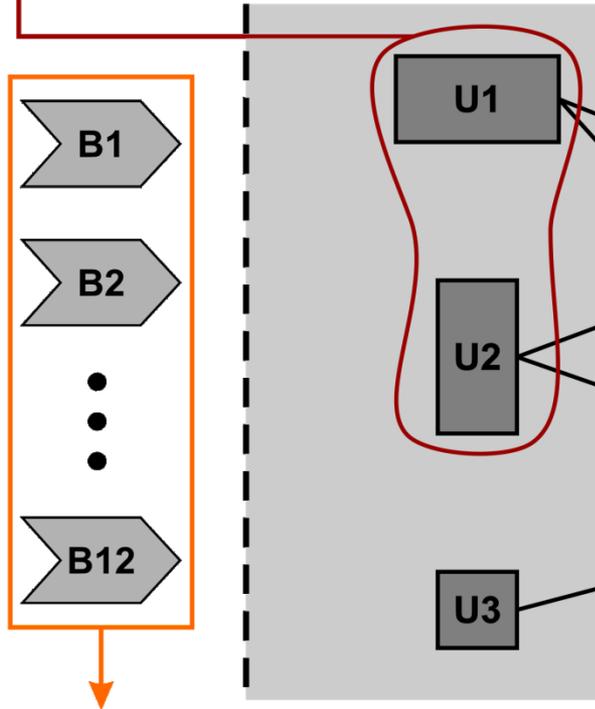


Definición del problema

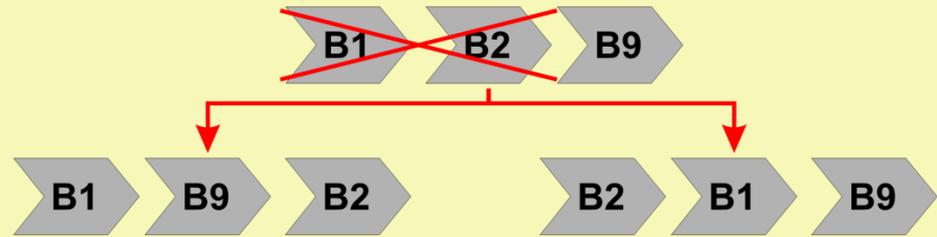


Definición del problema

- Unidades no idénticas
- Distintos tiempos de procesamiento para el mismo "batch"
 - Posibles prohibiciones de un "batch" en un equipo

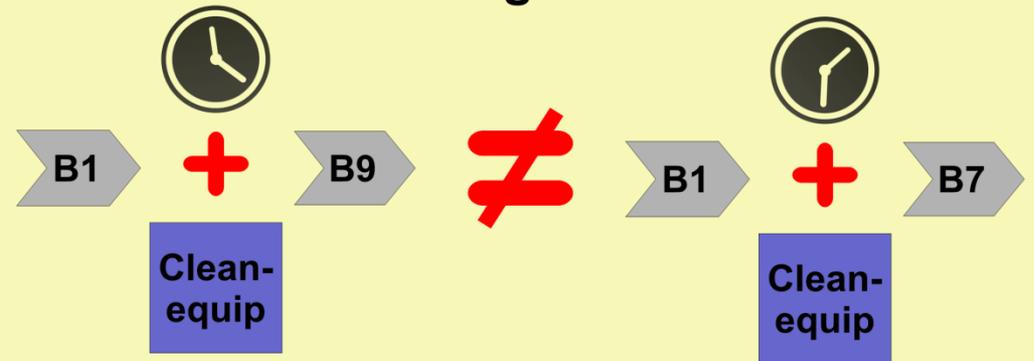


Secuencias de procesamiento prohibidas



Ej.: secuencia prohibida "batch 1"- "batch 2"

"Changeover"

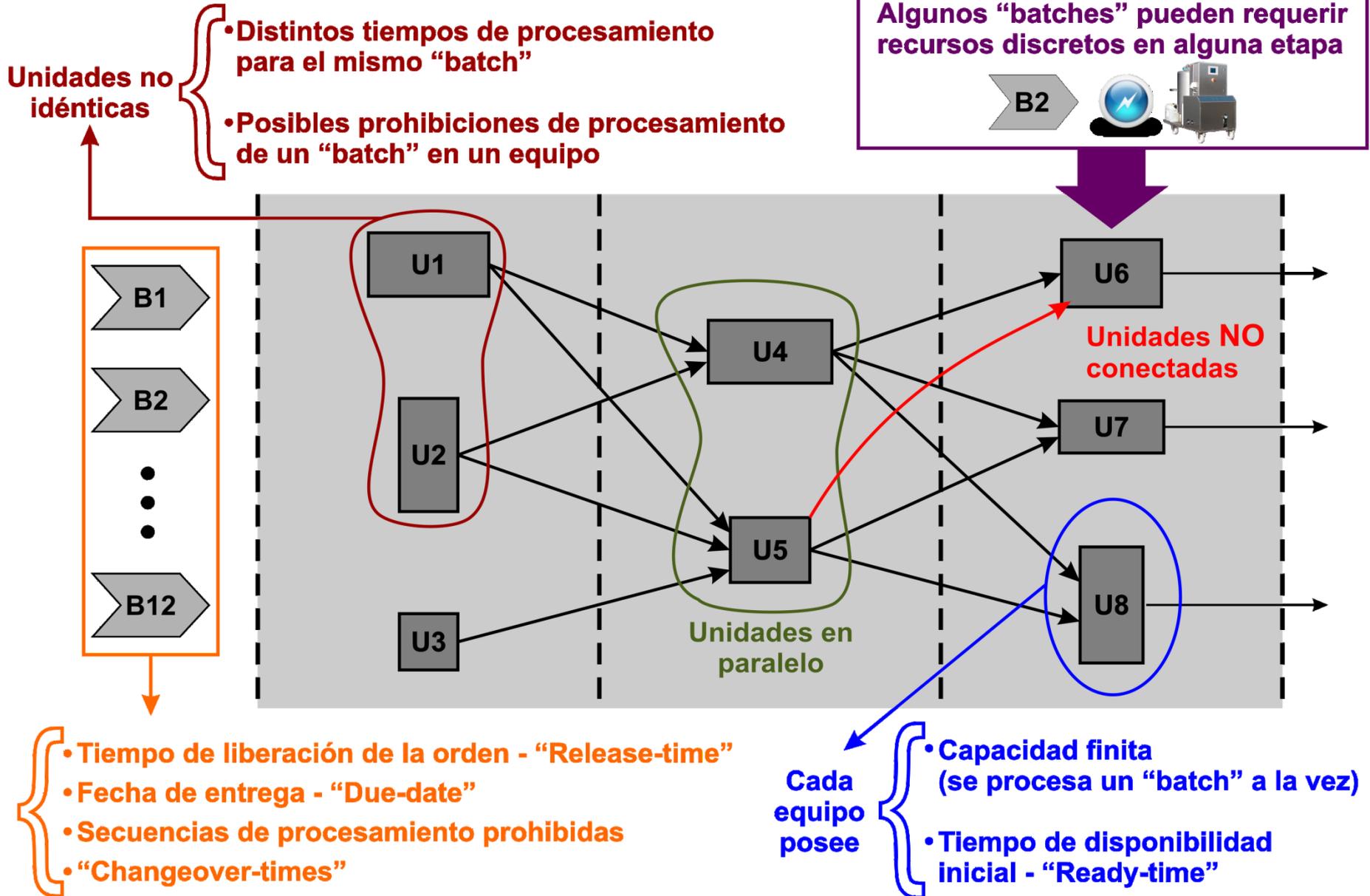


equipo posee

- Tiempo de disponibilidad inicial - "Ready-time"

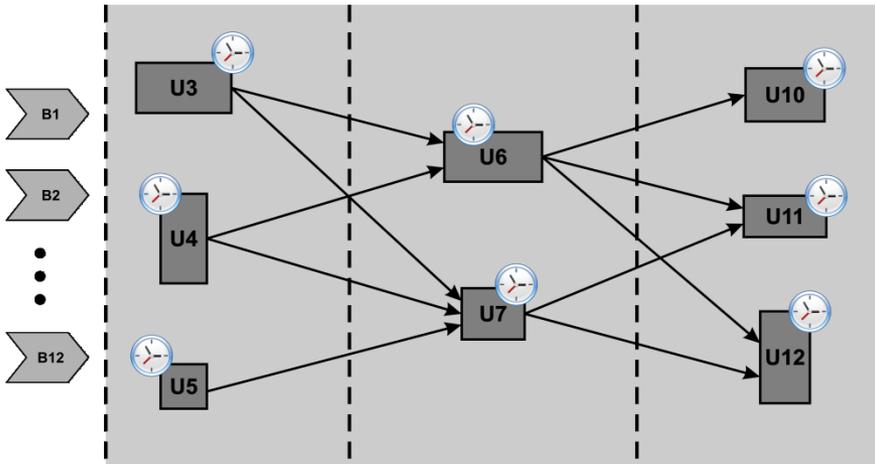
- Tiempo de liberación de la orden
- Fecha de entrega - "Due-date"
- Secuencias de procesamiento prohibidas
- "Changeover-times"

Definición del problema

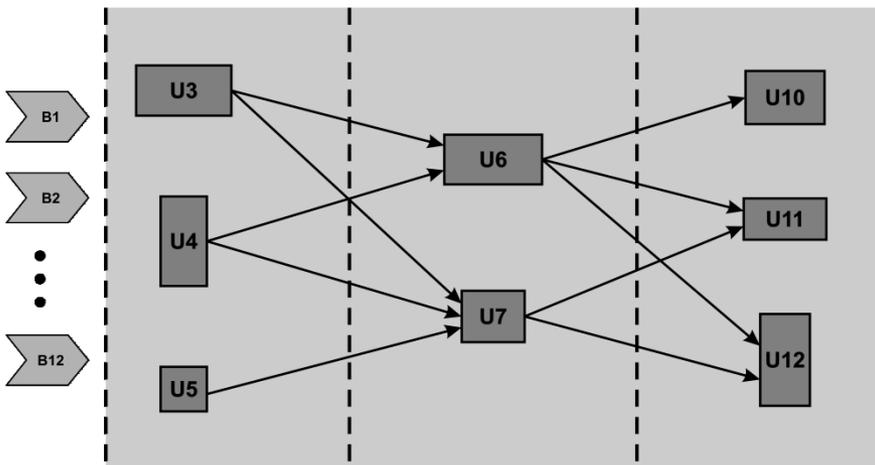


Políticas de almacenamiento/operación

Con restricciones de almacenamiento

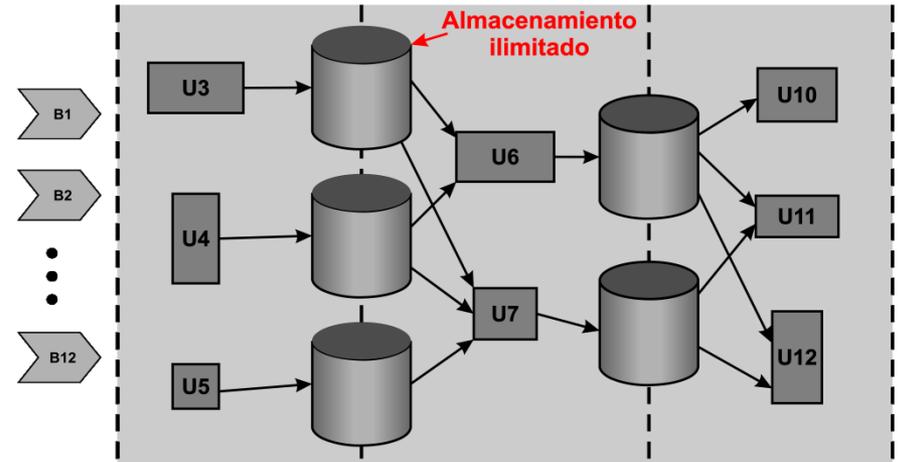


“Nonintermediate storage, unlimited wait”, NIS/UW



“Nonintermediate storage, zero wait”, NIS/ZW

Sin restricciones de almacenamiento



“Unlimited intermediate storage”, UIS

- El modelado matemático del problema de “scheduling” en estos ambientes lleva a problemas combinatorios de tipo NP-hard
- La resolución de casos reales, en general, implica elevados tiempos de CPU. En la práctica se dispone de poco tiempo.
- La representación de algunas características del ambiente lleva a expresiones complejas.

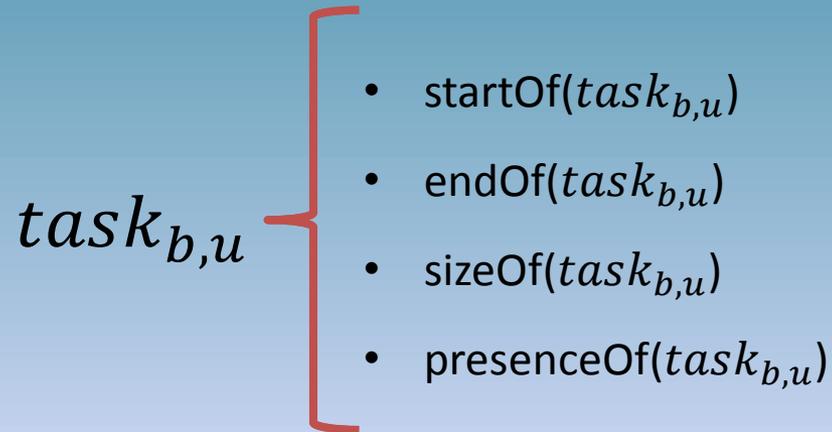
- El modelado matemático del problema de “scheduling” en estos ambientes lleva a problemas combinatorios de tipo NP-hard
- La resolución de casos reales, en general, implica elevados tiempos de CPU. En la práctica se dispone de poco tiempo.
- La representación de algunas características del ambiente lleva a expresiones complejas.

Construcción de un modelo de Programación con Restricciones (CP)

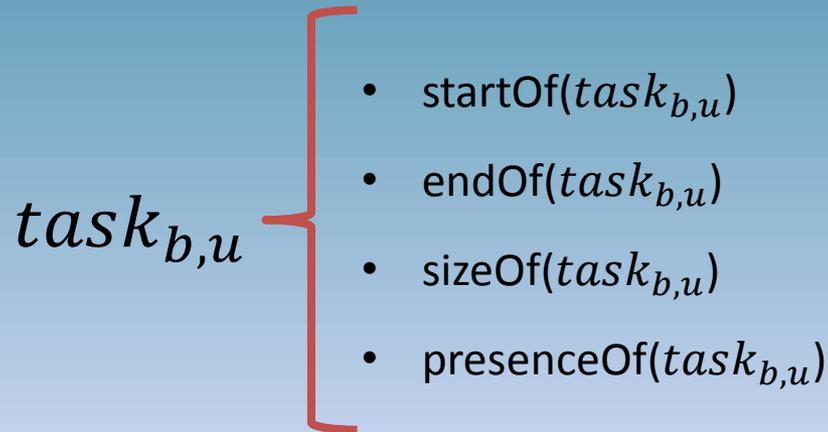
Lenguaje OPL - Entorno IBM ILOG CPLEX Optimization Studio 12.4

- Permite instanciar soluciones en bajos tiempos de CPU.
- Es posible establecer estrategias de búsqueda utilizando el conocimiento del dominio de trabajo.
- Posibilita construir restricciones matemático-lógicas de gran poder expresivo.
- Pueden emplearse constructores de alto nivel que facilitan el modelado.
- Utiliza algoritmos de búsqueda muy eficientes para los problemas de “scheduling”.

- Las **tareas productivas** se representan utilizando **variables de intervalo**



- Las **tareas productivas** se representan utilizando **variables de intervalo**



- Similarmente, para las **tareas de limpieza** se emplea

$cleanTask_{b1,b2,u}$

- Cada “batch” debe ser **asignado** a una unidad por etapa

$$\sum_{u \in Units_s} presenceOf(task_{b,u}) = 1, \forall s \in Stages, \forall b \in Batches$$

- Cada tarea posee una **duración**

$$sizeOf(task_{b,u}) = pt_{b,u} \cdot presenceOf(task_{b,u}), \forall b \in Batches, \forall u \in Units$$

- Se debe respetar la relación de **precedencia entre las tareas** que se realizan sobre un mismo “batch”

$$endBeforeStart(task_{b,u1}, task_{b,u2}), \quad \forall b \in Batches, \\ \forall u1 \in Units_{s1}, \forall u2 \in Units_{s2}, \forall s1, s2 \in Stages, s1 + 1 = s2$$

- Cada “batch” debe ser **asignado** a una unidad por etapa

$$\sum_{u \in Units_s} presenceOf(task_{b,u}) = 1, \forall s \in Stages, \forall b \in Batches$$

- Cada tarea posee una **duración**

$$sizeOf(task_{b,u}) \geq p_{t_{b,u}} \cdot presenceOf(task_{b,u}), \forall b \in Batches, \forall u \in Units$$

Política NIS/UW

- Se debe respetar la relación de **precedencia entre las tareas** que se realizan sobre un mismo “batch”

$$endBeforeStart(task_{b,u1}, task_{b,u2}), \quad \forall b \in Batches, \\ \forall u1 \in Units_{s1}, \forall u2 \in Units_{s2}, \forall s1, s2 \in Stages, s1 + 1 = s2$$

- Cada “batch” debe ser **asignado** a una unidad por etapa

$$\sum_{u \in Units_s} presenceOf(task_{b,u}) = 1, \forall s \in Stages, \forall b \in Batches$$

- Cada tarea posee una **duración**

$$sizeOf(task_{b,u}) = pt_{b,u} \cdot presenceOf(task_{b,u}), \forall b \in Batches, \forall u \in Units$$

- Se debe respetar la relación de **precedencia entre las tareas** que se realizan sobre un mismo “batch”

$$endBeforeStart(task_{b,u1}, task_{b,u2}), \quad \forall b \in Batches, \\ \forall u1 \in Units_{s1}, \forall u2 \in Units_{s2}, \forall s1, s2 \in Stages, s1 + 1 = s2$$

- Cada “batch” debe ser **asignado** a una unidad por etapa

$$\sum_{u \in Units_s} presenceOf(task_{b,u}) = 1, \forall s \in Stages, \forall b \in Batches$$

- Cada tarea posee una **duración**

$$sizeOf(task_{b,u}) = pt_{b,u} \cdot presenceOf(task_{b,u}), \forall b \in Batches, \forall u \in Units$$

- Se debe respetar la relación de **precedencia entre las tareas** que se realizan sobre un mismo “batch”

Política
NIS/ZW

$$endAtStart(task_{b,u_1}, task_{b,u_2}), \quad \forall b \in Batches, \\ \forall u_1 \in Units_{s_1}, \forall u_2 \in Units_{s_2}, \forall s_1, s_2 \in Stages, s_1 + 1 = s_2$$

- Las **unidades** son **recursos renovables**, pueden procesar sólo un “batch” a la vez

$$unitUsage_u \leq 1, \forall u \in Units$$

“**unitUsage**” es una “**cumul function**” definida como:

$$unitUsage_u = \sum_{b \in Batches} pulse(task_{b,u}, 1), \forall u \in Units$$

- La planta presenta **restricciones topológicas**, correspondientes a unidades de etapas sucesivas no comunicadas entre sí, que deben respetarse

$$presenceOf(task_{b,u1}) \Rightarrow presenceOf(task_{b,u2}) = 0 \\ \forall b \in Batches, \forall u1 \in Units, \forall u2 \in F_{u1}$$

- Las **unidades** son procesadas en un "batch" a la vez

cesar sólo un

"unitUsage" es

no:

unitUsage

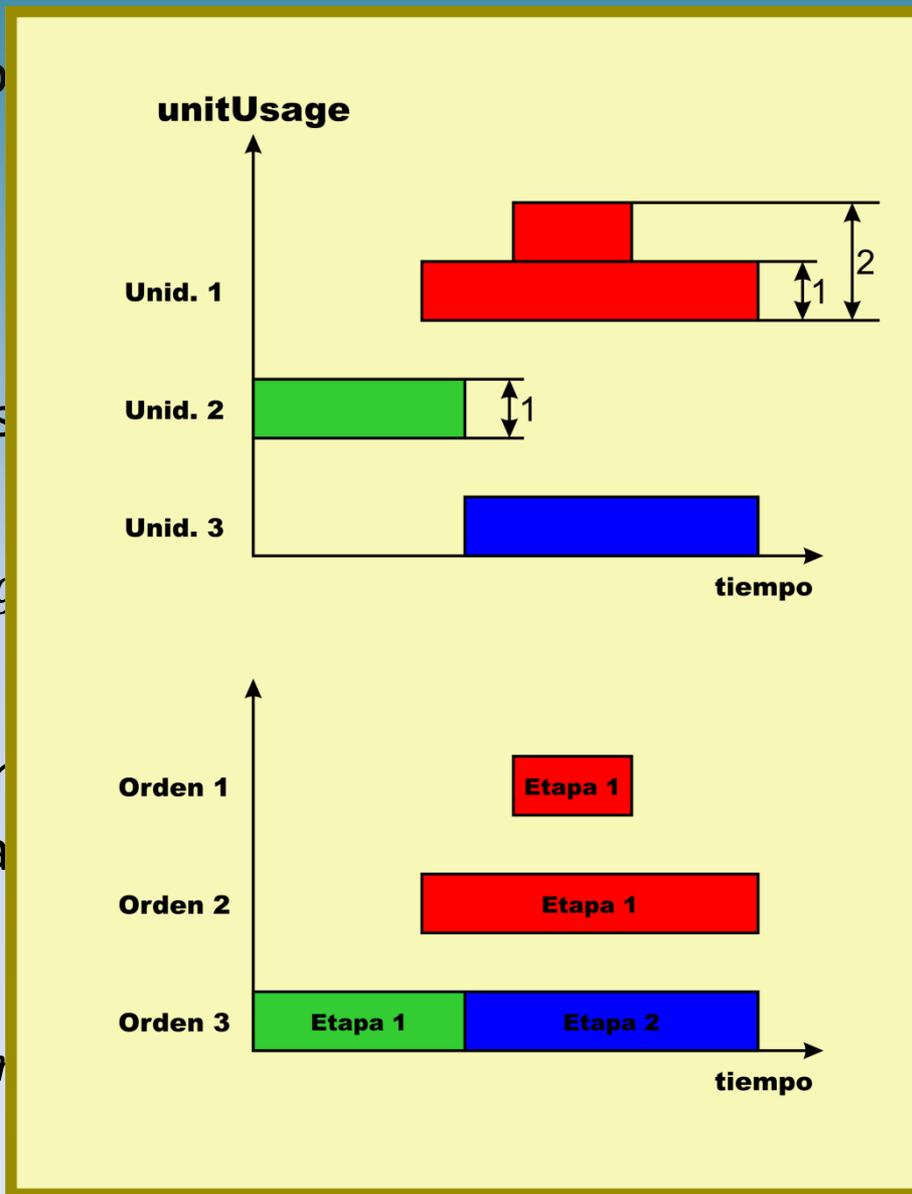
Units

- La planta presenta unidades de etapa que deben respetarse

correspondientes a
de sí, que deben

presen

$2) = 0$



- Las **unidades** son **recursos renovables**, pueden procesar sólo un “batch” a la vez

$$unitUsage_u \leq 1, \forall u \in Units$$

“**unitUsage**” es una “**cumul function**” definida como:

$$unitUsage_u = \sum_{b \in Batches} pulse(task_{b,u}, 1), \forall u \in Units$$

- La planta presenta **restricciones topológicas**, correspondientes a unidades de etapas sucesivas no comunicadas entre sí, que deben respetarse

$$presenceOf(task_{b,u1}) \Rightarrow presenceOf(task_{b,u2}) = 0 \\ \forall b \in Batches, \forall u1 \in Units, \forall u2 \in F_{u1}$$

- Entre dos “batches” procesados consecutivamente en una misma unidad existen **tiempos de “changeover”**

$$\begin{aligned}
 & \text{presenceOf}(\text{task}_{b_1,u}) \cdot \text{presenceOf}(\text{task}_{b_2,u}) \cdot \\
 & \min(\text{endOf}(\text{task}_{b_1,u}) + co_{b_1,b_2} + su_u, \text{endOf}(\text{task}_{b_2,u}) + co_{b_2,b_1} + su_u) \\
 & \leq \max(\text{startOf}(\text{task}_{b_1,u}), \text{startOf}(\text{task}_{b_2,u})) \\
 & \quad \forall b_1, b_2 \in \text{Batches}, \forall u \in \text{Units}
 \end{aligned}$$

- Una tarea no puede iniciarse antes de realizar el “**set-up**” de la unidad, ni antes del “**ready-time**” de dicha unidad. Tampoco puede comenzar antes del “**release-time**” del “batch”

$$\begin{aligned}
 \text{startOf}(\text{task}_{b,u}) & \geq \max(rd_u + su_u, rt_b) \cdot \text{presenceOf}(\text{task}_{b,u}) \\
 & \quad \forall b \in \text{Batches}, \forall u \in \text{Units}
 \end{aligned}$$

- Existen **secuencias de procesamiento prohibidas** que deben ser respetadas

$$\text{alwaysIn}(\text{forbiddenChangeOver}_{f.b2,u}, \text{task}_{f.b2,u}, 0), \quad \forall f \in Fb, \forall u \in Units$$

“**forbiddenChangeOver**” es una “**cumul function**” definida como:

$$\text{forbiddenChangeOver}_{f,u} = \text{stepAtStart}(\text{task}_{f.b1,u}, 1, 1) - \sum_{b \notin f} \text{stepAtStart}(\text{task}_{b,u}, 0, 1), \quad \forall f \in Fb, b \in Batches, \forall u \in Units$$

*Se incorpora el modelado de recursos discretos distintos de los equipos.
Un recurso puede ser demandado por un “batch” en una dada etapa.*

- Cada recurso tiene una **disponibilidad limitada**

$$resourceUsage_r \leq avail_r \quad \forall r \in Resources$$

“**resourceUsage**” es una “**cumul function**” definida como:

$$resourceUsage_r = \sum_{\forall b \in Batches} \sum_{\forall u \in Units_s} pulse(task_{b,u}, requir_{b,s,r}),$$

$$\forall r \in Resources$$

“**resourceUsage_r**” refleja el perfil de utilización del recurso “r”.

Los equipos de limpieza son recursos discretos.

Cada “changeover” demanda un equipo de limpieza.

Las tareas de limpieza se modelan con las variables “cleanTask”

- Cada **tarea de limpieza** tiene una **duración** determinada

$$\begin{aligned} \text{sizeOf}(\text{cleanTask}_{b_1,b_2,u}) &= \text{co}_{b_1,b_2} \cdot \text{presenceOf}(\text{cleanTask}_{b_1,b_2,u}) \\ &\forall b_1, b_2 \in \text{Batches}, b_1 \neq b_2, \forall u \in \text{Units} \end{aligned}$$

- Una tarea de limpieza correspondiente a un cierto “changeover” entre dos tareas de procesamiento se lleva a cabo sólo si se ejecutan esas tareas consecutivamente

$$\begin{aligned} \text{presenceOf}(\text{cleanTask}_{b_1,b_2,u}) &= 0 \\ &\Rightarrow \text{typeOfNext}(\text{taskSequence}_u, \text{task}_{b_1,u}) \neq b_2 \\ &\forall b_1, b_2 \in \text{Batches}, \forall u \in \text{Units} \end{aligned}$$

- Las **tareas de limpieza y producción** deben **sincronizarse**

$$\text{endOf}(\text{task}_{b_1,u}) \cdot \text{presenceOf}(\text{cleanTask}_{b_1,b_2,u}) \leq \text{startOf}(\text{cleanTask}_{b_1,b_2,u}), \forall b_1, b_2 \in \text{Batches}, b_1 \neq b_2, \forall u \in \text{Units}$$

$$\text{startOf}(\text{task}_{b_2,u}) \cdot \text{presenceOf}(\text{task}_{b_1,u}) \geq \text{endOf}(\text{cleanTask}_{b_1,b_2,u}), \forall b_1, b_2 \in \text{Batches}, b_1 \neq b_2, \forall u \in \text{Units}$$

- Los **equipos de limpieza** son **limitados**

$$\text{cleaningResourceUsage} = \sum_{u \in \text{Units}} \sum_{b_1, b_2} \text{pulse}(\text{cleanTask}_{b_1,b_2,u}, 1)$$

“**cleaningResourceUsage**” es una “**cumul function**” definida como:

$$\text{cleaningResourceUsage} \leq Q_{cr}, \forall u \in \text{Units}$$

- Minimizar el **makespan, Mk**

$$\text{endOf}(\text{task}_{b,u}) \leq Mk, \quad \forall b \in \text{Batches}, \forall u \in \text{Units}$$

- Minimizar la **tardanza total, T**

$$T = \sum_{\substack{u \in \text{Units}_s \\ : s = \text{numberStages}}} \sum_{b \in \text{Batches}} \max(0, \text{endOf}(\text{task}_{b,u}) - dd_b)$$

- Minimizar **costo total, CT**

$$CT = \sum_{u \in \text{Units}} \sum_{b \in \text{Batches}} \text{variableCost} \cdot \text{presenceOf}(\text{task}_{b,u}) + \sum_{u \in \text{Units}} Z_u \cdot \text{FixedCost}$$

Descripción del problema

- 5 etapas productivas
- 25 unidades
- 22 “batches” (órdenes)
- Política NIS/UW
- Secuencias prohibidas
- “Changeover-times”
- “Ready-times”
- “Release-times”

Función objetivo:
Minimizar makespan

$$endOf(Task_{b,u}) \leq Mk$$

$$\forall b \in Batches, \forall u \in Units$$



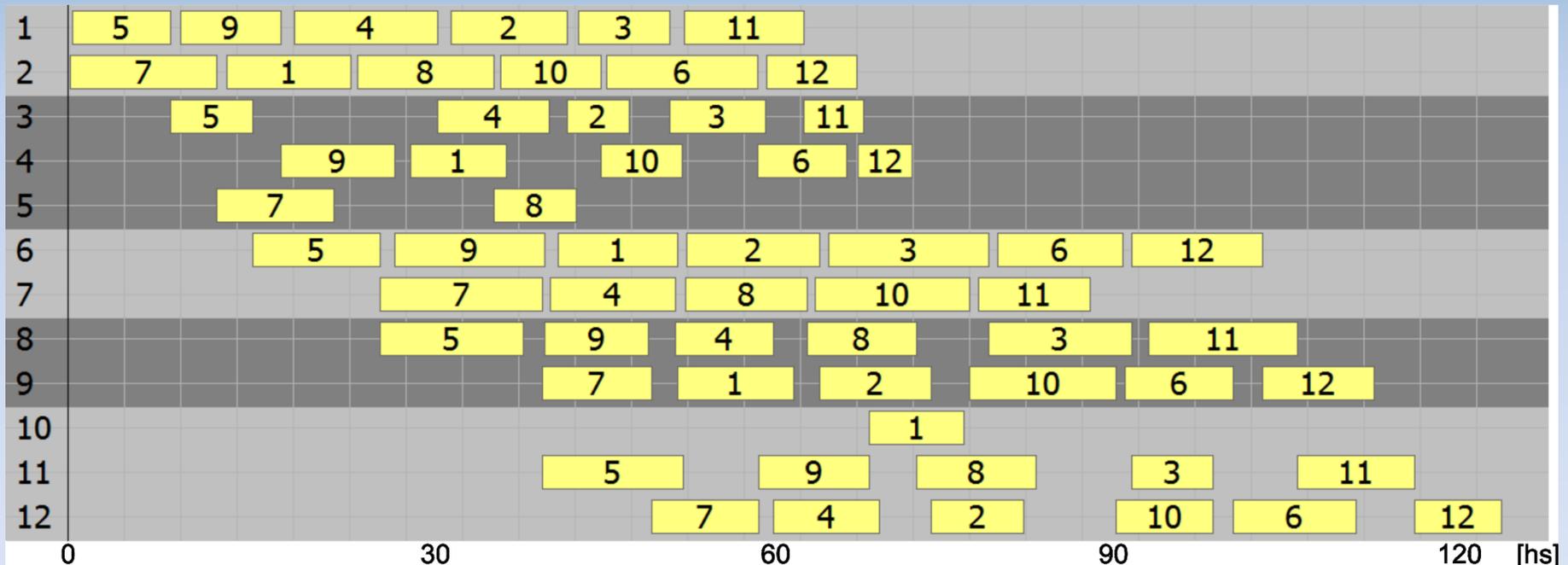
Zeballos y colab., 2011

Descripción del problema

- 5 etapas productivas
- 12 unidades
- 12 “batches” (órdenes)
- Política UIS
- “Setup-times”
- “Changeover-times”
- “Due-dates”
- 3 tipos de recursos renovables

Función objetivo: Minimizar tardanza total

$$T = \sum_{\substack{u \in Units_s: \\ s=numberStages}} \sum_{b \in Batches} \max(0, endOf(Task_{b,u}) - dd_b)$$



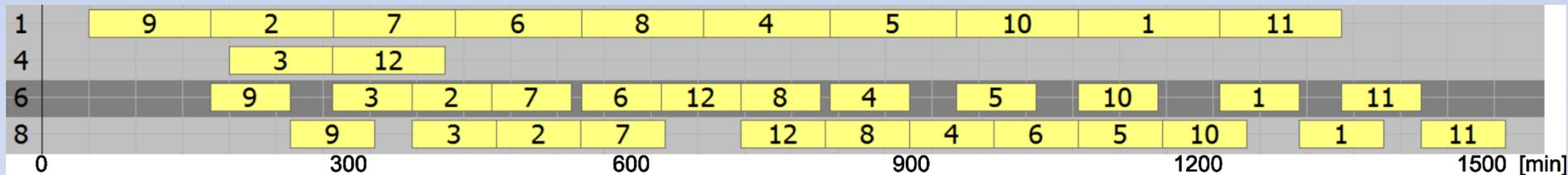
Marchetti y Cerdá, 2009

Descripción del problema

- 3 etapas productivas
- 8 unidades
- 12 “batches” (órdenes)
- Política UIS
- “Setup-times”
- Costo fijo de operación de equipos
- “Due-dates”
- Costo variable de producción, por “batch” y equipo

Función objetivo: Minimizar coto total

$$CT = \sum_{u \in Units} \sum_{b \in Batches} variableCost \cdot presenceOf(Task_{b,u}) + \sum_{u \in Units} Z_u \cdot FixedCost$$



Jain y colab., 2001

Tabla resumen casos abordados

Problema	Resultados de otros autores				Resultados del presente trabajo			
	Primera solución		Mejor solución		Primera solución		Mejor solución ^a	
	F.O.	CPU	F.O.	CPU	F.O.	CPU	F.O.	CPU
5-UIS	228.00	0.03	199.00	8.70	199.00	0.34	199.00	0.51
5-NIS/ZW	228.00	0.03	199.00	9.31	200.00	0.43	199.00	0.59
5-NIS/UW	228.00	0.03	199.00	10.31	206.00	0.35	199.00	0.54
12-UIS	354.00	0.11	293.10 ^b	42.88	264.00	3.40	200.00 ^b	309.78
12-NIS/ZW	381.00	0.13	311.20 ^b	168.06	245.00	1.80	199.00 ^b	132.57
12-NIS/UW	370.10	0.13	301.20 ^b	378.00	245.00	1.54	199.00 ^b	127.42
22-UIS	534.40	0.33	509.40 ^b	4.47	273.00	3.38	291.00 ^b	833.63
22-NIS/ZW	-	-	-	-	377.00	24.72	311.30 ^b	578.43
22-NIS/UW	592.40	1.84	550.40 ^b	26.63	352.90	45.36	352.90 ^b	45.36
Caso 2	-	-	31.6	114.05	93.40	2.24	31.60 ^b	47.00
Caso 3.a	-	-	111 ^b	c	125	0.18	100 ^b	1.32
Caso 3.b	-	-	704 ^b	c	768	0.18	704 ^b	4.18

Variantes de 5-UIS	Número de batches por producto	Cantidad de equipos de limpieza	Primera solución		Mejor solución ^a	
			F.O.	CPU	F.O.	CPU
V1	[1,1,1,1,1]	1	344.30	19.67	199.00	2.12
V2	[2,1,1,1,1]	1	200.00	2.90	199.00	5.88
V3	[4,1,1,6,1]	No se contempla	297.50	0.56	212.60 ^b	47.31
V4	[4,1,1,6,1]	1	212.60	138.31	212.60 ^b	138.31
V5	[4,1,1,6,1]	2	243.60	38.3	212.60 ^b	364.51

a. Solución obtenida en un tiempo máximo de 900 segundos de CPU.

b. Solución sub-óptima /solución óptima no comprobada.

c. Tiempos no reportados por diferencias tecnológicas en los equipos de cómputo.

- Se desarrolló un modelo CP para ambientes productivos “batch” multiproducto, multietapa.
- En todos los ejemplos abordados se logró encontrar soluciones iniciales en pocos segundos de CPU, así como obtener soluciones finales óptimas o de alta calidad.
- El modelo se probó en diferentes variantes de este tipo de ambiente, demostrando ser flexible y robusto.
- Resta validar este modelo, así como las nuevas versiones desarrolladas, utilizando casos de mayor tamaño y diferentes funciones objetivo.
- Se está trabajando y pretende validar un modelo que permita representar aquellos ambientes productivos que operan en “modo campaña”.

Muchas gracias por su atención!!!

**Franco M. Novara - Juan M. Novas - Gabriela P. Henning
INTEC (UNL, CONICET) – Santa Fe, Argentina**



1. Maravelias, C.T.: General Framework and Modeling Approach Classification for Chemical Production Scheduling. AIChE Journal, In press (2012)
2. Castro, P.M., Grossmann, I.E.: New Continuous-Time MILP Model for the Short-Term Scheduling of Multistage Batch Plants. Ind. Eng. Chem. 44, 9175-9190 (2005)
3. Marchetti, P.A., Cerdá, J.: A general resource-constrained scheduling framework for multistage batch facilities with sequence-dependent changeovers. Computers and Chemical Engineering 33, 871-886 (2009)
4. Zeballos, L.J., Novas, J.M., Henning, G.P.: A CP formulation for scheduling multiproduct multistage batch plants. Computers and Chemical Engineering 35, 2973-2989 (2011)
5. Jain, V., Grossmann, I. E. Algorithms for hybrid MILP/CP models for a class of optimization problems. INFORMS Journal of Computing, 13, 258–276. (2001)
6. Harjunkski, I., Grossmann, I.E.: Decomposition techniques for multistage scheduling problems using mixed-integer and constraint programming methods. Computers and Chemical Engineering. 26, 1533-1552 (2002)
7. Méndez, C.A., Cerdá, J., Grossman, I.E., Harjunkski, I., Fahl, M.: State-of-the-art review of optimization methods for short-term scheduling of batch processes. Computers and Chemical Engineering 30, 913-946 (2006)
8. IBM ILOG CPLEX Optimization Studio <http://www-01.ibm.com/software/integration/optimization/cplex-optimization-studio/>