UNCUYO
UNIVERSIDAD NACIONAL DE CUYO

FACULTAD DE INGENIERÍA

UNIDAD 3

PROGRAMACIÓN DE SISTEMAS DE FLUJO INTERMITENTE

Expositor: **Ing. Nicolás De Simone**

Institución: **Facultad de Ingeniería – U.N.Cuyo**

PROGRAMACIÓN DE SISTEMAS INTERMITENTES

Objetivos de la presentación

- ✓ Analizar la importancia de los diferentes Sistemas de Producción.
- ✓ Definir los Tipos de Estrategia de Procesos.
- ✓ Estudiar los Procesos Intermitentes: estrategia, ventajas y desventajas.
- ✓ Programación de Sistemas de Flujo Intermitente: introducción y definición de objetivos.
- ✓ Secuenciamiento de múltiples trabajos “n” en 1 máquina.
- ✓ Programación de múltiples trabajos “n” en “m” máquinas.

  Algoritmo Johnson: para “n” trabajos y 2 máquinas.

  Algoritmo CDS: para “n” trabajos y 3 máquinas.

  Método de Bera: para “n” trabajos y “m” máquinas.
- ✓ Programación de un número de trabajos en un mismo número de máquinas correctas (matriz nxn o mxm).
- ✓ Programación mediante el uso de software de Simulación.

SISTEMAS DE PRODUCCIÓN

Por qué son tan importante.....

La importancia de los sistemas de producción, y lógicamente de su selección, radica en cómo afectan:

- Costos.
- Calidad.
- Tiempos.
- Confiabilidad.
- Flexibilidad de las operaciones y productos obtenidos.

"¿Qué son las Prioridades Competitivas Estratégicas?"

Representan el énfasis o foco estratégico que una empresa le otorga a sus procesos dentro de su cadena de valor.

SISTEMAS DE PRODUCCIÓN

1. **Costo**
2. **Calidad**
3. **Tiempo**
4. **Flexibilidad**
5. **Innovación**

1. **Costo**
2. **Calidad**
3. **Tiempo**
4. **Flexibilidad**
5. **Innovación**

Las empresas hacen foco en 1 o 2 dimensiones claves

SISTEMAS DE PRODUCCIÓN

Estrategias de Operaciones

Implica la determinación de cómo producir un producto o cómo proporcionar un servicio.

- Objetivos:
  - Hallar el camino para satisfacer los requerimientos de los clientes.
  - Satisfacer los objetivos de producción y gestión según las prioridades competitivas.

Las estrategias de operaciones tienen efectos a largo plazo dado que las mismas se generan definiendo los siguientes parámetros:

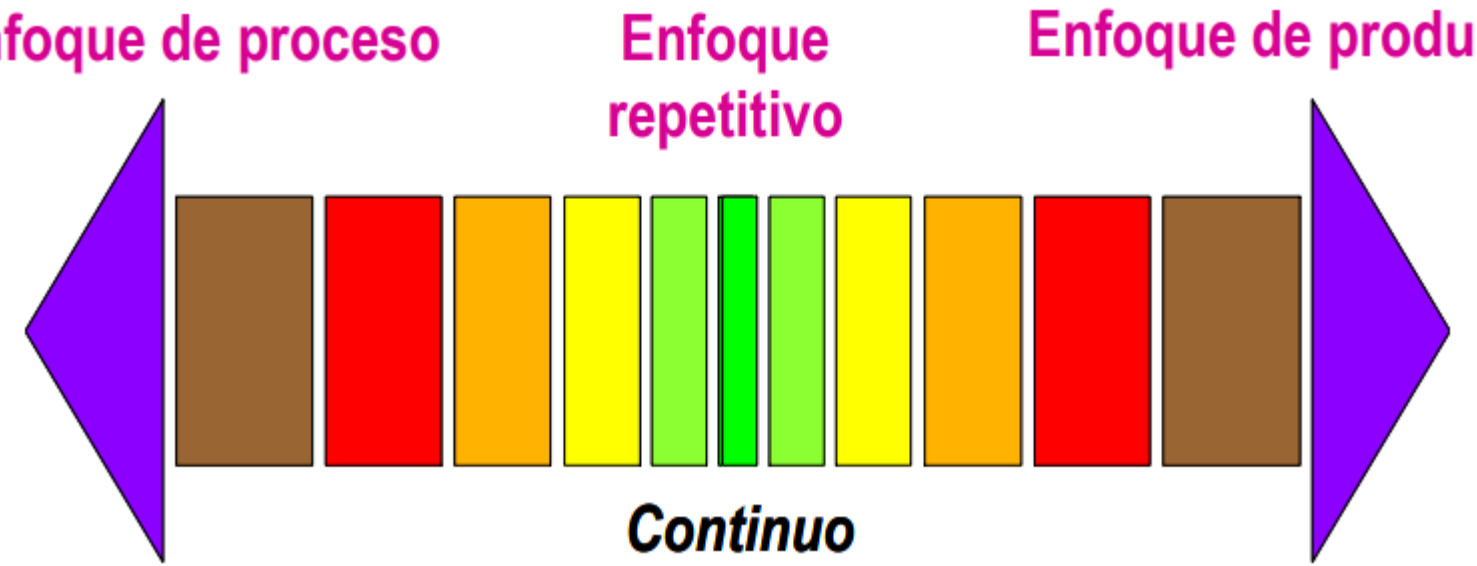
- ✓ Volumen de producción.
- ✓ Eficiencia de procesos.
- ✓ Flexibilidad del producto/servicio.
- ✓ Costo y calidad.

SISTEMAS DE PRODUCCIÓN

Tipos de Estrategias de Procesos

Dentro de las Estrategias de Operaciones, se deben definir las Estrategias de Procesos.

En el interior de una determinada instalación, se pueden utilizar varias estrategias. Las mismas pueden ser clasificadas de la siguiente forma:

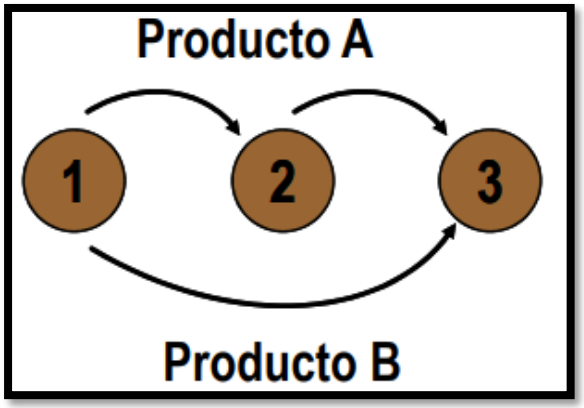


Cabe destacar que, además de los enfoques arriba mencionados, en la bibliografía encontrarán también un Enfoque por Proyectos o de Flujo Nulo.

PROCESOS INTERMITENTES

Estrategias de Enfoque de Proceso

- Esta estrategia es también conocida como Procesos Intermitentes.
- Las instalaciones se organizan para realizar un proceso.
- Los centros de trabajo similares suelen estar juntos. Por ejemplo, los tornos suelen estar cercanos unos de otros; las máquinas agujereadoras también, etc.
- La fabricación consiste en poca cantidad con mucha variedad.
- Ejemplos: IMPSA, talleres metalúrgicos, bancos, hospitales, etc.



PROCESOS INTERMITENTES

- **Estrategias de Enfoque de Proceso**

PROCESOS INTERMITENTES

Pros y Contras de la Estrategias de Enfoque de Proceso

- Ventajas:
  - Mayor flexibilidad del producto.
  - Equipamiento con utilidad más general.
  - Baja inversión de capital inicial.
- Desventajas:
  - Mayor formación de sus empleados.
  - Compleja planificación y control de las operaciones.
  - Escasa utilización de los equipos (del 5% al 25%).

PROGRAMACIÓN DE SISTEMAS INTERMITENTES

Introducción

En los sistemas de flujo intermitente cada producto que fluye de un centro de trabajo a otro sufre detenciones, por lo que el movimiento es irregular, no continuo.

Los inventarios de productos en proceso se acumulan, por lo que la programación se vuelve más compleja.

Se forman colas de inventarios de productos en proceso (WIP) en cada centro de trabajo, y los productos esperan hasta que las instalaciones estén disponibles.

La programación de sistemas de flujo intermitente consiste en decidir qué productos deben procesarse en cada momento en cada centro, para reducir las colas.

Objetivos de la Programación

Los objetivos de la programación de sistemas de flujo intermitente son:

- Cumplir los plazos de entrega de los productos.
- Minimizar el tiempo muerto o de demora en el uso de los recursos.
- Minimizar tiempos o costos de preparación de los recursos.
- Minimizar el inventario de los trabajos sin terminar.
- Maximizar el aprovechamiento de máquinas y trabajadores.

Es muy probable, que todos los objetivos arriba mencionados **NO** puedan cumplirse simultáneamente. Es por ello, que es fundamental mantener una perspectiva de sistemas para asegurarse de que los objetivos de los diferentes centros de trabajo estén sincronizados con la estrategia de operaciones de la organización.

PROGRAMACIÓN DE SISTEMAS INTERMITENTES

➢ Secuenciamiento de múltiples trabajos “n” en 1 máquina

El proceso de determinar el pedido en una máquina o en un centro de trabajo se llama **secuenciación** o secuenciación por prioridades.

Las *reglas de prioridad* son reglas usadas para obtener una secuenciación de los trabajos. Las mismas pueden ser estáticas o dinámicas (si es que dependen del factor tiempo o no).

1. FCFS (*first-come, first-served*, primero en entrar, primero en trabajarse) Los pedidos se ejecutan en el orden en que llegan al departamento.
2. SOT (*shortest operating time*, tiempo de operación más breve) Ejecutar primero el trabajo con el tiempo de terminación más breve, luego el siguiente más breve, etc. Se llama también SPT (*shortest processing time*, tiempo de procesamiento más breve). A veces la regla se combina con una regla de retardo para evitar que los trabajos con tiempos más demorados se atrasen demasiado.
3. EDD (*earliest due date first*, primero el plazo más próximo) Se ejecuta primero el trabajo que antes se venza.
4. STR (*slack time remaining*, tiempo ocioso restante) Se calcula como el tiempo que queda antes de que se venza el plazo menos el tiempo restante de procesamiento. Los pedidos con menor tiempo ocioso restante (STR) se ejecutan primero.

STR = Tiempo restante antes de la fecha de vencimiento – tiempo de procesamiento restante

PROGRAMACIÓN DE SISTEMAS INTERMITENTES

➢ Secuenciamiento de múltiples trabajos “n” en 1 máquina

5. STR/OP (*slack time remaining per operation*, tiempo ocioso restante por operación) Se ejecutan primero los pedidos con el menor tiempo ocioso por número de operaciones.

   STR/OP = STR/Número de operaciones restantes

6. CR (*proporción crítica*) Se calcula como la diferencia entre la fecha de vencimiento y la fecha actual, dividida entre el número de días hábiles que quedan. Se ejecutan primero los pedidos con la menor CR.
7. LCFS (*last-come, first-served*, último en llegar, primero en trabajarse). Esta regla se aplica a menudo automáticamente. Cuando llegan los pedidos, de ordinario se colocan arriba de la pila; el operador toma primero el que esté más alto.
8. Orden aleatorio o a capricho. Los supervisores u operadores escogen el trabajo que quieran ejecutar.

Las siguientes medidas de desempeño de los programas se usan para evaluar las reglas de prioridad:

- ✓ Cumplir las fechas de los clientes.
- ✓ Minimizar el tiempo de tránsito (el tiempo que pasa un trabajo en el proceso).
- ✓ Minimizar el inventario de los trabajos sin terminar.
- ✓ Minimizar el tiempo ocioso de máquinas y trabajadores.

➢ Programación de múltiples trabajos “n” en “m” máquinas

Algoritmo de Johnson

El algoritmo de Johnson consiste en una serie de pasos para establecer una secuenciación de ”n” trabajos en 2 máquinas. Tomando como base los trabajos que tienen menos tiempo de realización y así, irlos efectuando según sea la máquina o la prioridad.

Finaliza con una *gráfica de Gantt* para tener una visión más clara de la secuencia en cada máquina y los tiempos requeridos totales.

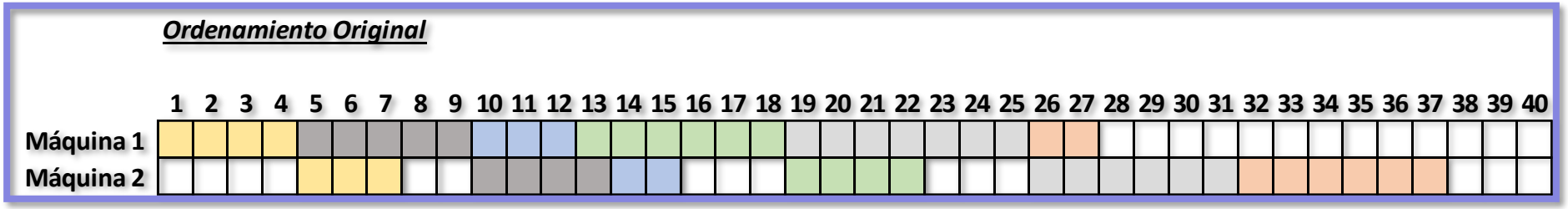
Analicemos un ejemplo para mostrar las bondades de la aplicación de este algoritmo:

PROGRAMACIÓN DE SISTEMAS INTERMITENTES

Sin aplicar el algoritmo de Johnson:

Trabajos	Máquina 1	Máquina 2	Reorden Trabajos
Trab N°1	4h	3h	(1)
Trab N°2	5h	4h	(2)
Trab N°3	3h	2h	(3)
Trab N°4	6h	4h	(4)
Trab N°5	7h	6h	(5)
Trab N°6	2h	6h	(6)

Ordenamiento Original



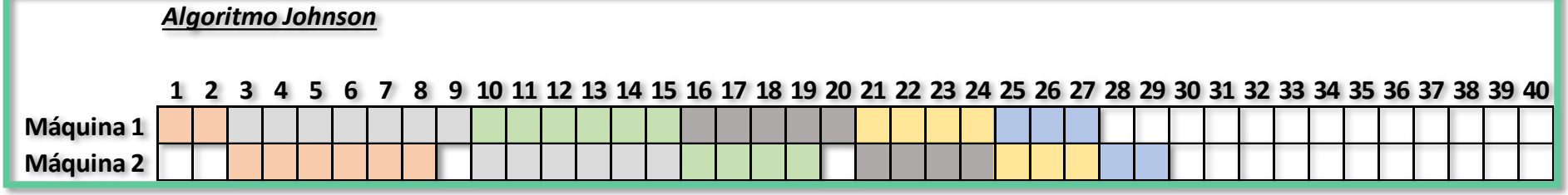
- Tiempo de Fabricación de la secuencia: 37 hs.
- Tiempo Muerto total de los recursos (máquinas): 12 hs.

PROGRAMACIÓN DE SISTEMAS INTERMITENTES

Aplicando el algoritmo de Johnson:

Trabajos	Máquina 1	Máquina 2	Reorden Trabajos
Trab N°1	4h	3h	(6)
Trab N°2	5h	4h	(5)
Trab N°3	3h	2h	(4)
Trab N°4	6h	4h	(2)
Trab N°5	7h	6h	(1)
Trab N°6	2h	6h	(3)

Algoritmo Johnson



- Tiempo de Fabricación de la secuencia: 29 hs.
- Tiempo Muerto total de los recursos (máquinas): 4 hs.

➢ Programación de múltiples trabajos “n” en “m” máquinas

Algoritmo Heurístico CDS

Campbeel, Dudek, y Smith (CDS) usan una múltiple aplicación del algoritmo de Johnson para dos máquinas para tratar de obtener una buena programación en el problema de programación del Flowshop. Esencialmente, ellos crean (m-1) problemas de programación, los cuales resuelven con el algoritmo de Johnson.

Para ser precisos los k problemas (k=1,2,..(m-1)) son formados como sigue. Los tiempos de proceso para la primera máquina para el i-ésimo trabajo es:

$$a_i^{(k)} = \sum_{j=1}^{k} P_{ij}$$

Esto es la suma de los tiempos de proceso para la i-ésima tarea en el primer k de máquinas actuales. De la misma manera, se construye el tiempo para la segunda máquina que es la suma de los tiempos de proceso para la i-ésima tarea en la última k de las maquinas actuales.

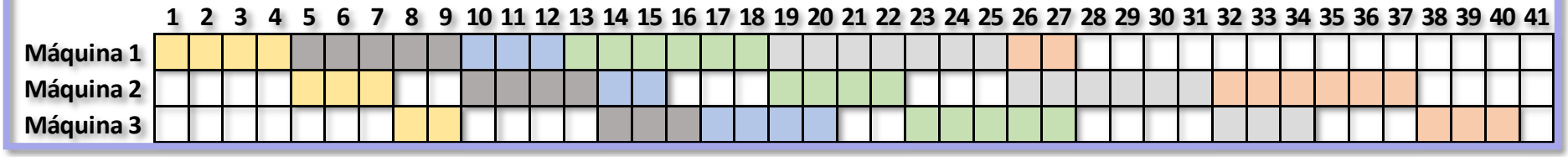
$$b_i^{(k)} = \sum_{j=m-k+1}^{m} P_{ij}$$

PROGRAMACIÓN DE SISTEMAS INTERMITENTES

Sin aplicar el algoritmo CDS:

Trabajos	Máquina 1	Máquina 2	Máquina 3	Reorden Trabajos
Trab N°1	4h	3h	2h	(1)
Trab N°2	5h	4h	3h	(2)
Trab N°3	3h	2h	4h	(3)
Trab N°4	6h	4h	5h	(4)
Trab N°5	7h	6h	3h	(5)
Trab N°6	2h	6h	3h	(6)

Ordenamiento Original



- Tiempo de Fabricación de la secuencia: 40 hs.
- Tiempo Muerto total de los recursos (máquinas): 32 hs.

PROGRAMACIÓN DE SISTEMAS INTERMITENTES

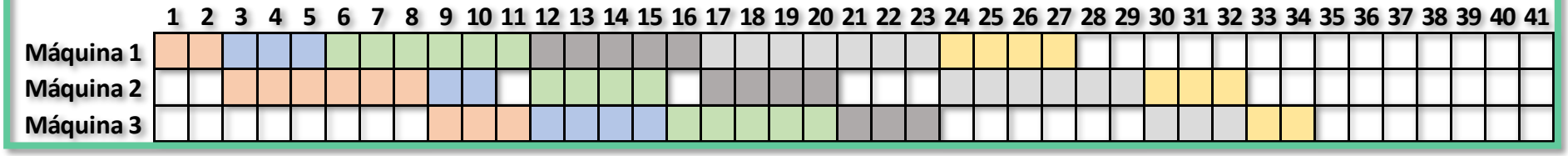
Aplicando el algoritmo CDS (Algoritmo Johnson N° 1):

Trabajos	Máquina 1	Máquina 2	Máquina 3
Trab N°1	4h	3h	2h
Trab N°2	5h	4h	3h
Trab N°3	3h	2h	4h
Trab N°4	6h	4h	5h
Trab N°5	7h	6h	3h
Trab N°6	2h	6h	3h

Trabajos	Máquina 1	Máquina 3
Trab N°1	4h	2h
Trab N°2	5h	3h
Trab N°3	3h	4h
Trab N°4	6h	5h
Trab N°5	7h	3h
Trab N°6	2h	3h

Reorden Trabajos
(6)
(3)
(4)
(2)
(5)
(1)

Método CDS - Algoritmo Johnson N°1



- Tiempo de Fabricación de la secuencia: 34 hs.
- Tiempo Muerto total de los recursos (máquinas): 21 hs.

PROGRAMACIÓN DE SISTEMAS INTERMITENTES

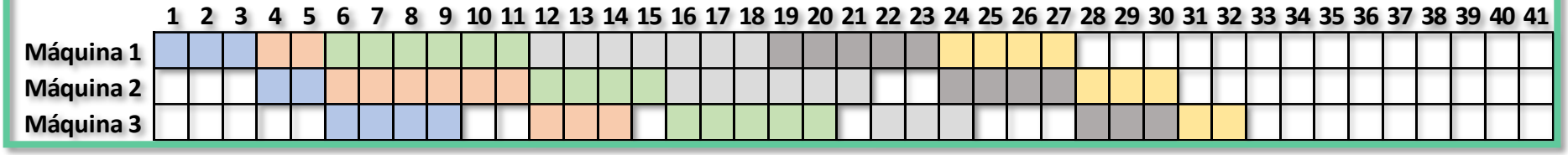
Aplicando el algoritmo CDS (Algoritmo Johnson N°2):

Trabajos	Máquina 1	Máquina 2	Máquina 3
Trab N°1	4h	3h	2h
Trab N°2	5h	4h	3h
Trab N°3	3h	2h	4h
Trab N°4	6h	4h	5h
Trab N°5	7h	6h	3h
Trab N°6	2h	6h	3h

Trabajos	M1+M2	M3+M2
Trab N°1	7h	5h
Trab N°2	9h	7h
Trab N°3	5h	6h
Trab N°4	10h	9h
Trab N°5	13h	9h
Trab N°6	8h	9h

Reorden Trabajos
(3)
(6)
(4)
(5)
(2)
(1)

Método CDS - Algoritmo Johnson N°2



- Tiempo de Fabricación de la secuencia: 32 hs.
- Tiempo Muerto total de los recursos (máquinas): 17 hs.

➢ Programación de múltiples trabajos “n” en “m” máquinas

Método de Bera

El método de Bera, desarrollado por H. Bera en 1996, mediante un indicador numérico permite determinar el orden de entrada de los pedidos en el sistema.

A diferencia del algoritmo Johnson, este método nos permite programar múltiples trabajos en múltiples máquinas “sin tener limitaciones”.

Sumando los tiempos de procesamiento entre máquinas consecutivas, de a dos máquinas (M1+M2, M2+M3,…, Mn-1+Mn), y el resultado menor de las diferentes sumas será el denominador para calcular al factor de Bera.

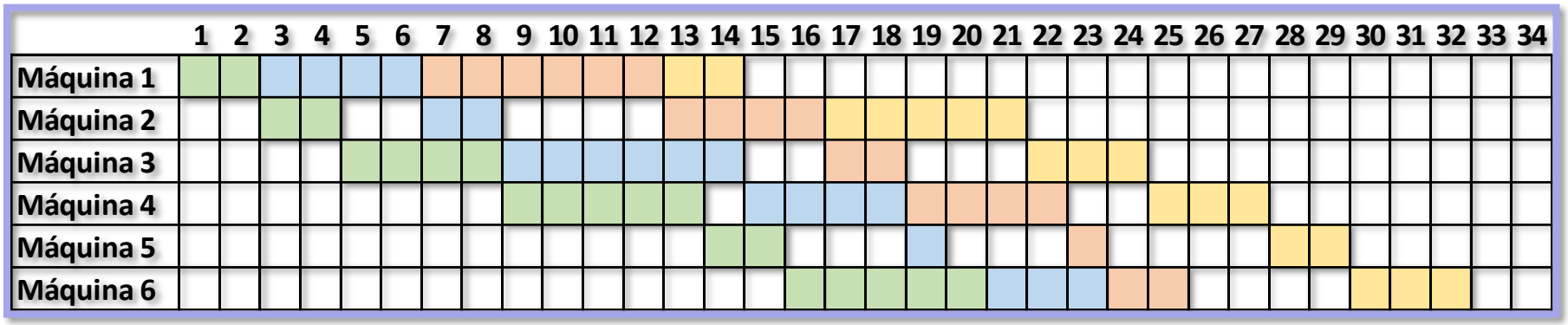
El numerador siempre será uno, pero su signo dependerá del tiempo entre la primera máquina y la última, donde, si el tiempo de la primera máquina es menor a la última máquina, el numerador será *negativo*, de ser mayor o igual el numerador será *positivo*.

A continuación, se muestra un ejemplo de secuenciación utilizando el método de Bera.

PROGRAMACIÓN DE SISTEMAS INTERMITENTES

Sin aplicar el método de Bera:

	Trabajo 1	Trabajo 2	Trabajo 3	Trabajo 4
Máquina 1	**2**	**4**	**6**	**2**
Máquina 2	2	2	4	5
Máquina 3	4	6	2	3
Máquina 4	5	4	4	3
Máquina 5	2	1	1	2
Máquina 6	**5**	**3**	**2**	**3**



- Tiempo de Fabricación de la secuencia: 32 hs.
- Tiempo Muerto total de los recursos (máquinas): 70 hs.

PROGRAMACIÓN DE SISTEMAS INTERMITENTES

Aplicando el método de Bera:

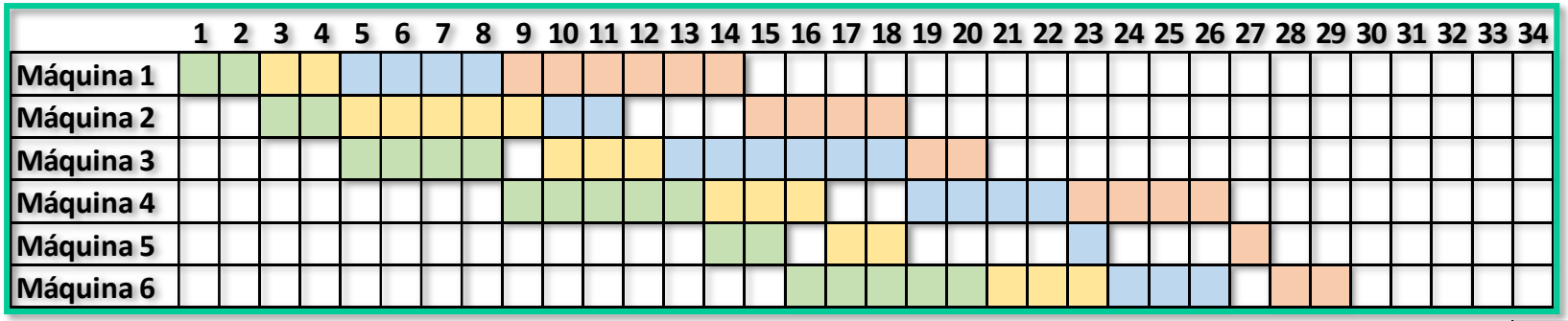
	Trabajo 1	Trabajo 2	Trabajo 3	Trabajo 4
Máquina 1	2	4	6	2
Máquina 2	2	2	4	5
Máquina 3	4	6	2	3
Máquina 4	5	4	4	3
Máquina 5	2	1	1	2
Máquina 6	5	3	2	3

	Trabajo 1	Trabajo 2	Trabajo 3	Trabajo 4
	4	6	10	7
	6	8	6	8
	9	10	6	6
	7	5	5	5
	7	4	3	5
Mínimo:	4	4	3	5

Numerador:	-1	1	1	-1
Denominador:	4	4	3	5
Cociente:	-0,250	0,250	0,333	-0,200

Orden: | Trabajo 1 | Trabajo 4 | Trabajo 2 | Trabajo 3 |

- Tiempo de la secuencia: 29 hs.
- Tiempo Muerto total: 57 hs.



PROGRAMACIÓN DE SISTEMAS INTERMITENTES

➢ Programación de un número de trabajos en un mismo número de máquinas (matriz nxn o mxm)

Algunos centros de trabajo tienen *suficientes máquinas correctas* para iniciar todos los trabajos al mismo tiempo. Aquí el problema **NO** es qué trabajo hacer primero, sino qué asignación de trabajos a cuáles máquinas dará el mejor programa general. En estos casos, se aplica el método de asignación.

*El **método de asignación** es un caso especial del método de transporte de programación lineal.* Puede aplicarse a situaciones en las que hay *n fuentes de oferta y n usos de la demanda* (como cinco trabajos en cinco máquinas) y el objetivo es minimizar o maximizar alguna medida de eficacia. El método de asignación es apropiado para resolver problemas que tienen las características siguientes:

1. **Hay n "cosas" que se distribuyen a n "destinos".**
2. **Cada cosa debe asignarse a un, y sólo un, destino.**
3. **Sólo puede aplicarse un criterio (por ejemplo, costo mínimo, utilidad máxima o tiempo mínimo de terminación).**

EJEMPLO 19.3: Método de asignación

Supóngase que un programador tiene cinco trabajos que pueden realizarse en cinco máquinas ($n = 5$). El costo de terminar cada combinación de trabajos y máquinas se muestra en la ilustración 19.5. El programador quisiera diseñar una asignación de costo mínimo (¡hay 5!, o 120 posibles asignaciones).

SOLUCIÓN

Este problema puede resolverse con el método de asignación, que consiste en cuatro pasos (observe que también puede resolverse con el Solver de Excel):

1. Se resta el número menor de cada *hilera* del mismo número y de todos los números de la hilera (por tanto, habrá al menos un cero en cada hilera).
2. Se resta el número menor de cada *columna* de todos los demás números de la columna (por tanto, habrá por lo menos un cero en cada columna).
3. Se determina si el número *mínimo* de rectas necesarias para cubrir todos los ceros es igual a n. En tal caso, se encontró una solución óptima, porque las asignaciones de trabajos a las máquinas deben hacerse en las entradas cero y esta prueba demuestra que es posible. Si el número mínimo de rectas necesario es menor que n, se va al paso 4.
4. Se traza el mínimo número de rectas por todos los ceros (pueden ser las mismas líneas usadas en el paso 3). Se resta el número mínimo descubierto por las rectas del mismo número y de todos los otros números descubiertos y se suma al número de cada intersección de las rectas. Se repite el paso 3.

PROGRAMACIÓN DE SISTEMAS INTERMITENTES

Matriz de asignación con los costos de procesamiento de las máquinas para cada trabajo

	Máquina				
Trabajo	A	B	C	D	E
I	$5	$6	$4	$8	$3
II	6	4	9	8	5
III	4	3	2	5	4
IV	7	2	4	5	3
V	3	6	4	5	5

Paso 1: Reducción de hileras: el número menor se resta de cada hilera.

	Máquina				
Trabajo	A	B	C	D	E
I	2	3	1	5	0
II	2	0	5	4	1
III	2	1	0	3	2
IV	5	0	2	3	1
V	0	3	1	2	2

Paso 2: Reducción de columnas: el número menor se resta de cada columna.

	Máquina				
Trabajo	A	B	C	D	E
I	2	3	1	3	0
II	2	0	5	2	1
III	2	1	0	1	2
IV	5	0	2	1	1
V	0	3	1	0	2

PROGRAMACIÓN DE SISTEMAS INTERMITENTES

Paso 3: Se aplica la prueba de la recta: el número de rectas que tachen todos los ceros es 4; como se requieren 5, se avanza al paso 4.

Trabajo	Máquina A	B	C	D	E
I	2	3	1	3	0
II	2	0	5	2	1
III	2	1	0	1	2
IV	5	0	2	1	1
V	0	3	1	0	2

Paso 4: Se resta el menor número descubierto y se suma a la intersección de las rectas. Con las rectas trazadas en el paso 3, el menor número descubierto es 1.

Trabajo	Máquina A	B	C	D	E
I	1	3	0	2	0
II	1	0	4	1	1
III	2	1	0	1	2
IV	4	0	1	0	1
V	0	4	1	0	2

Solución óptima, según la "prueba de las rectas".

Trabajo	Máquina A	B	C	D	E
I	1	3	0	2	0
II	1	0	4	1	1
III	2	2	0	1	3
IV	4	0	1	0	1
V	0	4	1	0	3

Asignación óptima y sus costos.

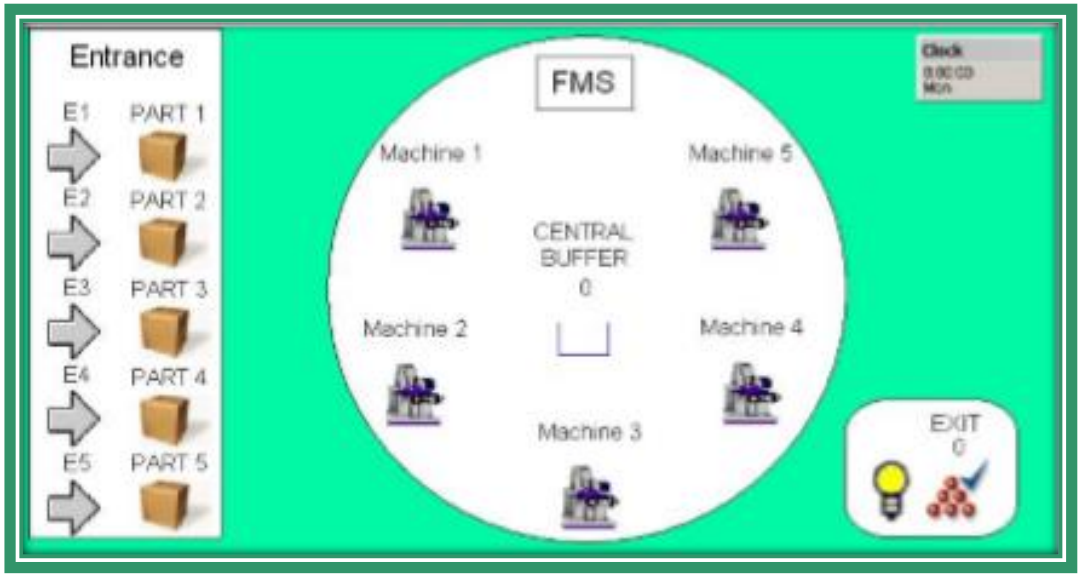
Trabajo I a la máquina E	$3
Trabajo II a la máquina B	4
Trabajo III a la máquina C	2
Trabajo IV a la máquina D	5
Trabajo V a la máquina A	3
Costo total	$17

PROGRAMACIÓN DE SISTEMAS INTERMITENTES

➢ Programación mediante el uso de software de Simulación

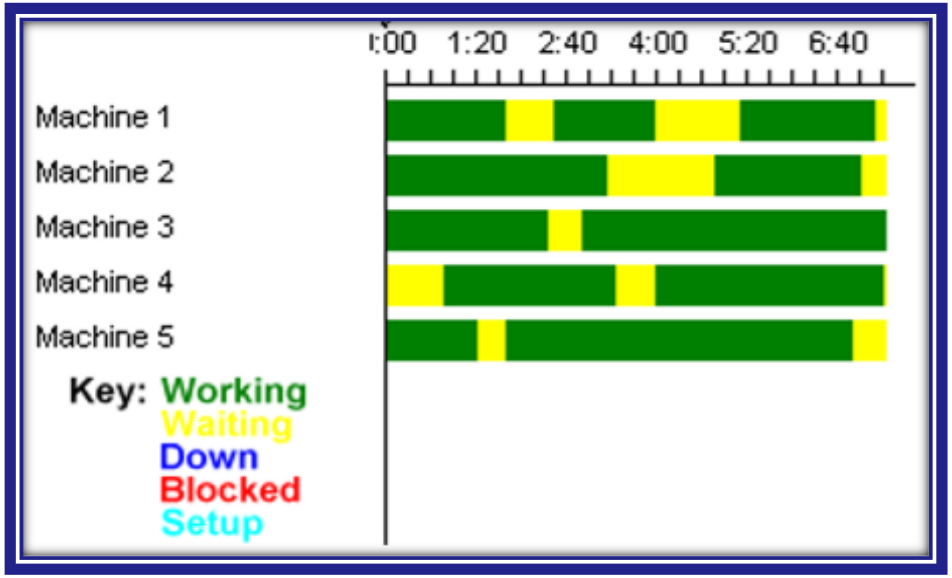
La utilidad de lograr un modelo de simulación en *"sistemas complejos"* es:

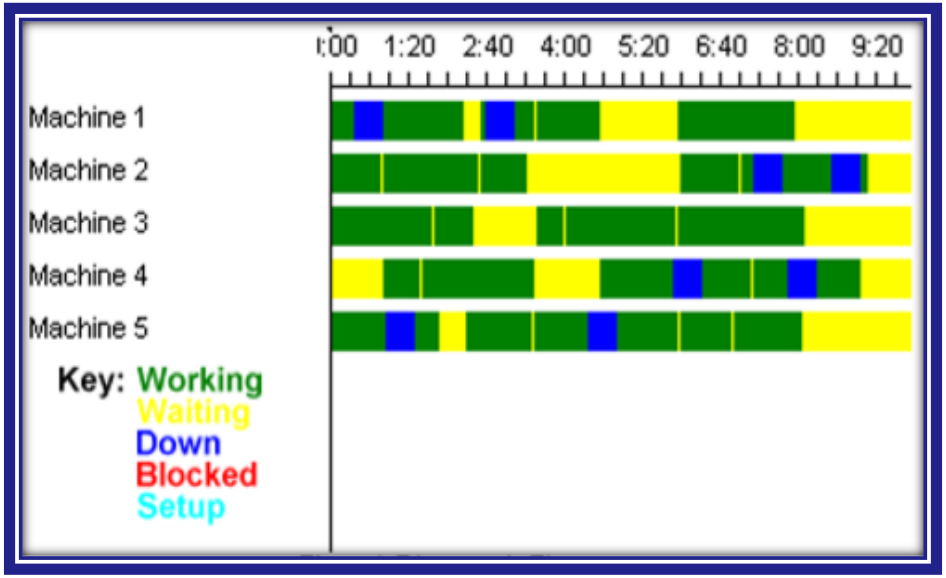
- ✓ Optimizar los recursos y reducir al mínimo los tiempos muertos de los mismos.
- ✓ Realizar análisis de sensibilidad y determinar en qué medida influye la variación de ciertas variables del sistema.
- ✓ Plantear nuevos escenarios modificando algunas variables del sistema.



PROGRAMACIÓN DE SISTEMAS INTERMITENTES

- **Programación mediante el uso de software de Simulación**





PROGRAMACIÓN DE SISTEMAS INTERMITENTES

- **Programación mediante el uso de software de Simulación**

Objeto Simulado	Medida de Desempeño	Resultado de la corrida
Máquina 1	Espera %	29.12
	Procesando %	70.88
Máquina 2	Espera %	26.86
	Procesando %	73.14
Máquina 3	Espera %	6.77
	Procesando %	93.23
Máquina 4	Espera %	19.64
	Procesando %	80.36
Máquina 5	Espera %	12.19
	Procesando %	87.81
SALIDA	Número de Partes Completadas	5
	Tiempo Promedio en el Sistema (Parte 1 en Máquina 2)	420
	Tiempo Promedio en el Sistema (Parte 2 en Máquina 1)	433
	Tiempo Promedio en el Sistema (Parte 3 en Máquina 3)	443
	Tiempo Promedio en el Sistema (Parte 4 en Máquina 5)	413
	Tiempo Promedio en el Sistema (Parte 5 en Máquina 4)	440

PROGRAMACIÓN DE SISTEMAS INTERMITENTES

- **Programación mediante el uso de software de Simulación**

Muchas Gracias por su atención!!!