Capítulo 7 Diseño y equilibrado de líneas de producción y montaje

7.1 Conceptos

En la producción en masa de muchas unidades de productos con un alto grado de homogeneidad, un dispositivo comúnmente utilizado, en la fabricación y/o en el montaje, es el de la cadena o línea de puestos de trabajo, que en cierta forma lleva a sus últimas consecuencias los conceptos de división del trabajo y de especialización. Aunque este dispositivo era empleado desde tiempos lejanos (por ejemplo en el Arsenal de Venecia fundado en el siglo XII) fue popularizado por Henry Ford tras su implantación en la fabricación de su vehículo Ford T en 1914. Las unidades de producto circulan de un puesto de trabajo al siguiente, en forma continua o intermitente, para recibir en cada uno de ellos, atendido, en principio, por un solo operario, una cierta cantidad de trabajo. La suma de todos los elementos de trabajo recibidos corresponde al total de las operaciones precisas para que la unidad pueda convertirse en un producto terminado.

Como se ha indicado en el capítulo 6 del tomo I, la cadena corresponde a una configuración de la distribución en planta orientada a producto; puede tener una longitud de varios kilómetros, estar subdividida en centenares de estaciones o puestos de trabajo y una unidad puede tardar más de un día laborable en recorrerla. Un problema no siempre fácilmente soluble que presenta el diseño de la cadena es la asignación de las tareas a las estaciones, que es función de la tasa de producción que desea obtenerse de la misma y que se traduce en la velocidad de circulación de las unidades por la línea. Dicha tasa de producción define el tiempo que cada unidad está disponible en cada estación para efectuar en la misma las tareas correspondientes, tiempo que se denomina tiempo ciclo o simplemente ciclo. Aunque dicho tiempo posee un cierto grado de flexibilidad, ésta no es ilimitada, lo que impone restricciones a la secuencia de unidades circulantes. Dos unidades sucesivas que exijan un esfuerzo considerable en una de las estaciones (superior al tiempo ciclo) pueden crear en la misma un problema insoluble; el operario de la estación puede tener dificultades en la realización de todas las tareas requeridas por la segunda unidad. Este fenómeno fue ilustrado por Charles Chaplin en la película "Tiempos

modernos", aunque en la mayoría de los casos la consecuencia no será la invasión por el operario de la estación del territorio de la estación siguiente, sino que la segunda unidad continuará su camino sin que se hayan realizado en ella todas las tareas, por lo que en algún momento deberá ser retirada de la línea para su finalización, con el incremento de coste y pérdida de calidad que ello implica.

Dado que los productos y los procesos se modifican a lo largo del tiempo y que las tasas de producción son así mismo cambiantes, debido a las oscilaciones de la demanda, el equilibrado de la cadena (asignación de tareas a estaciones) es un problema recurrente.

Existe mucha bibliografia relativa al efecto alienante de la línea de montaje sobre los operarios obligados a realizar indefinidamente las mismas operaciones que constituyen una mínima parte del trabajo necesario hasta ver un producto terminado. Sin dejar de reconocer algunos aspectos perniciosos consideramos que existen diferentes tipologías que definen la posición respecto al trabajo de los operarios, y algunas de ellas pueden hallar satisfactorio dedicar algunas horas a una labor alimenticia con pocas complicaciones a cambio de disponer de un remanente de tiempo para abordar otras actividades más gratificantes aunque no remuneradas. Consideramos muy significativo el fracaso de algunos modelos alternativos de producción ("islas" de fabricación, ver apartado 8.1.1.3 del tomo I).

La filosofía productiva JIT conduce a considerar, aun a falta de arrastre mecánico de las unidades, un establecimiento formal de estaciones de trabajo, entre las que se reparten las tareas, para las que también es preciso proceder a un equilibrado, lo que ha dado un nuevo impulso a las técnicas adecuadas para realizarlo.

En el presente capítulo trataremos en al apartado 7.1.1 de las definiciones iniciales para plantear el problema de equilibrado así como de modelos analíticos. En 7.1.2 describiremos la heurística de Helgeson & Birnie, ilustrada con varios ejemplos. En el apartado 7.1.3 analizaremos los procedimientos de simulación y en 7.1.4 otras heurísticas que se han propuesto recientemente, dejando para 7.1.5 las conclusiones.

En 7.1.6 introduciremos el problema de la secuenciación de unidades, del que analizaremos diversas variantes según los objetivos perseguidos. En 7.1.7 analizaremos la relativa a la regularización de las tasas de productos (problema de Miltenburg) y en 7.1.8 la modificación propuesta por Inman & Bulfin. El apartado 7.1.9 está orientado a la regularización de las tasas de consumo de recursos (problema de Monden).

7.1.1 Líneas de producción y montaje

Una primera clasificación de las líneas es la siguiente:

- Línea monomodelo (*single model line*), es una línea diseñada para fabricar un sólo producto o modelo,

- Línea multimodelo (*multimodel line*), es una línea en la que dos o más modelos se producen por lotes sucesivamente,

- Línea mixta (*mixed model line*), es una línea en la que simultáneamente se producen dos o más modelos,

En la línea multimodelo el paso de un modelo a otro exige habitualmente una adaptación de la línea (tiempo de preparación), lo que justifica la adopción de lotes de cada modelo, mientras que en la línea mixta las unidades de los distintos modelos se suceden en una secuencia (preestablecida o no) sin tiempos de preparación, por lo que las estaciones deben estar dispuestas para atender a cualquiera de los modelos. Inicialmente nos centraremos en las líneas monomodelo.

Llamaremos tarea o elemento de trabajo al nivel de división más fino respecto a la asignación a puestos de trabajo. El elemento, i ($1 \leq i \leq n$), representa una cantidad de trabajo que no puede distribuirse entre dos estaciones (ni entre dos operarios). El tiempo asignado a este elemento de trabajo lo llamaremos p_i.

Los elementos se agrupan formando una operación que se asigna a una estación o puesto de trabajo. Una característica del funcionamiento de una cadena es el tiempo ciclo o simplemente ciclo, C; corresponde al tiempo de que dispone cada estación para efectuar sobre una unidad las tareas que tiene asignadas. Salvo la existencia de estaciones en paralelo, podemos acotar superior e inferiormente C mediante las siguientes expresiones:

$$\max p_i \leq C \leq \sum p_i$$

El ciclo es una característica fundamental, pues define la tasa de producción de la cadena. En efecto, si cada C unidades de tiempo una unidad de producto pasa de una estación o puesto de trabajo al siguiente, también pasarán C unidades de tiempo entre la salida de la cadena de dos unidades terminadas sucesivas. La tasa de producción (número de unidades producidas por unidad de tiempo) es la inversa del ciclo:

$$P = \frac{1}{C}$$

Por tanto si C = 10 minutos, y trabajamos a dos turnos de 7 horas 40 minutos efectivos, la producción será:

$$P = \frac{2 \times 460}{10} = 92 \text{ unidades/día}$$

Dada una estación j, sea ET_j el conjunto de elementos de trabajo asignados a ella. Llamaremos ocupación S_j de la estación a:

$$S_j = \sum_{i \in ET_j} p_i$$

Debe cumplirse:

$$S_j = \sum_{i \in ET_j} p_i \leq C$$

de lo contrario el operario de dicha estación no podría realizar todas sus tareas dentro del tiempo que la unidad está en su demarcación. Por otra parte el tiempo muerto u ocioso planificado para dicha estación será:

$$TM_j = C - S_j$$

Interesará, en general, que dicho valor no sea grande, ni haya amplias diferencias en el mismo de una estación a otra. De hecho, un objetivo razonable es intentar que dicho tiempo muerto sea nulo para todas las estaciones. Tiempos muertos importantes redundarán en la necesidad de mayor número de estaciones y operarios que los estrictamente necesarios en teoría. Una cota del número de estaciones correspondientes a un ciclo, *C*, dado es:

NME **= número mínimo de estaciones = entero por exceso de** $\frac{\sum p_i}{C}$

Consideremos los datos del ejemplo E-1 (figura 7.1.1.1), para los que:

$$\max p_i = 6 \; ; \; \sum p_i = 39$$

para un ciclo $C = 10$ $\quad NME = 4$
para un ciclo $C = 13$ $\quad NME = 3$

La indivisibilidad de los p_i puede impedir que se alcance dicho número mínimo (con tres tareas de duración 5, 7 y 8 es imposible alcanzar el número mínimo de 2 estaciones con los ciclos 10 u 11). La existencia de ligaduras o restricciones (por ejemplo, precedencias) es otro factor que puede conducir a dicha imposibilidad; así ocurre en el ejemplo E-1 con el ciclo $C = 13$. Finalmente la no disponibilidad de un procedimiento exacto viable de asignación de tareas a las estaciones que conduzca al número mínimo posible de éstas, puede llevarnos a un número *N* de estaciones superior a *NME*. La eficiencia para un número de estaciones *N* podemos determinarla de la forma siguiente:

$$E = \frac{\sum p_i}{N \mathbf{x} C}$$

el numerador representa el trabajo efectivo a realizar, el denominador el trabajo realmente disponible. La diferencia entre ambos es el tiempo muerto existente en la cadena:

$$TM = N \times C - \sum p_i$$

El número de estaciones *NME* **conduce a la eficiencia máxima ideal y a un tiempo muerto mínimo:**

$$EMI = \frac{\sum p_i}{NME \times C}$$

$$TMM = NME \times C - \sum p_i$$

Como ya se ha indicado, habitualmente existirán restricciones o ligaduras que condicionarán la asignación de las tareas a las estaciones. Pueden ser de varios tipos:

- Ligaduras de precedencia (*precedence constraints***), son las más habituales y simples de tratar, vienen impuestas generalmente por consideraciones tecnológicas; son en todo semejantes a las ligaduras potenciales de los proyectos. Su forma es "la tarea** *i* **no puede realizarse si antes no se ha efectuado la** *h***", lo que implica, ordenando las estaciones en el sentido de avance de la línea, que** *h* **deberá estar en la misma estación que** *i* **o en una anterior, pero jamás en una siguiente.**

- Ligaduras de zona (*zoning constraints***), que a su vez pueden adoptar tres formas:**

- zonificación positiva, que obliga a situar una tarea en la misma estación que otra tarea (por precisar la misma herramienta o habilidad, por ejemplo),

- zonificación negativa, que exige que una tarea no esté en la misma estación que otra tarea; es similar a una ligadura disyuntiva de los proyectos (las causas de la prohibición pueden residir en la disposición de la unidad en la línea: tareas a realizar por la derecha incompatibles con las similares a realizar por la izquierda, en peculiaridades del operario: tipo de especialización, tareas que ensucian las manos incompatibles con otras que exigen las manos limpias, etc.)

- zonificación límite, que exige que ciertas tareas se asignen a estaciones anteriores o posteriores a una dada (tareas a realizar debajo del producto sólo posibles después que se le haya dado la vuelta en una posición determinada, tareas que precisan una herramienta especial fija sólo posibles en ciertas estaciones proximas o incluso sólo en una, etc.)

Inicialmente sólo consideraremos las ligaduras de precedencia. Se llama equilibrado de la línea (*line balancing***) al proceso de asignar tareas a las estaciones a lo largo de la línea**

satisfaciendo las restricciones y procurando que las cantidades de trabajo en cada una de las estaciones sea lo más parecido posible. Según las circunstancias deberá considerarse:

- fijado el tiempo ciclo, lo que introduce una limitación en el trabajo asignado a cada estación y un objetivo obvio, la minimización del número de estaciones;

- fijado el número de estaciones, lo que define un tiempo ciclo mínimo y un objetivo obvio, la minimización del tiempo ciclo real;

- límites del tiempo ciclo (inferior) y del número de estaciones (superior);

Tarea *i*	**Duración** p_i	**Precedentes inmediatas**
a	**5**	**-**
b	**4**	**-**
c	**5**	*a*
d	**6**	*b*
e	**2**	*c,d*
f	**4**	*e*
g	**3**	*e*
h	**5**	*f,g*
i	**2**	*h*
j	**3**	*h*

Fig. 7.1.1.1 Datos del ejemplo E-1

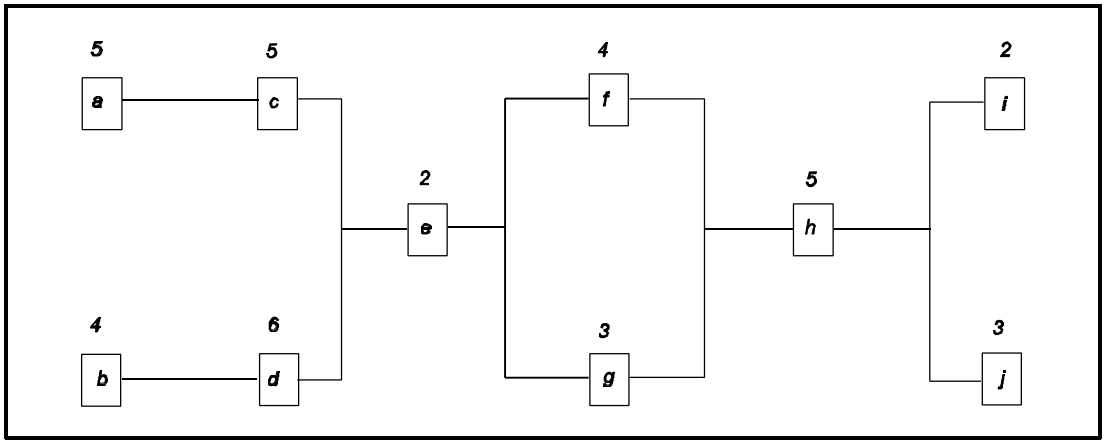


Fig. 7.1.1.2 Grafo de precedencias del ejemplo E-1

Dadas las tareas, sus duraciones y las precedencias podemos establecer un grafo que las represente (semejante a la representación ROY de los proyectos; en este caso pueden existir más de una tarea sin precedentes y más de una sin siguientes). Respecto a los grafos de los proyectos, los correspondientes a las precedencias de montaje serán usualmente más "cortos" y "anchos". No existirán ni bucles y circuitos, lo que permite establecer funciones ordinales y clasificar las tareas por niveles (como en la lista de materiales). La explotación de la adscripción de las tareas a niveles es la idea que se encuentra en la base del método debido a Kilbridge & Wester para establecer la asignación de las mismas a las estaciones. En la figura 7.1.1.2 hemos representado el grafo de precedencias del ejemplo E-1.

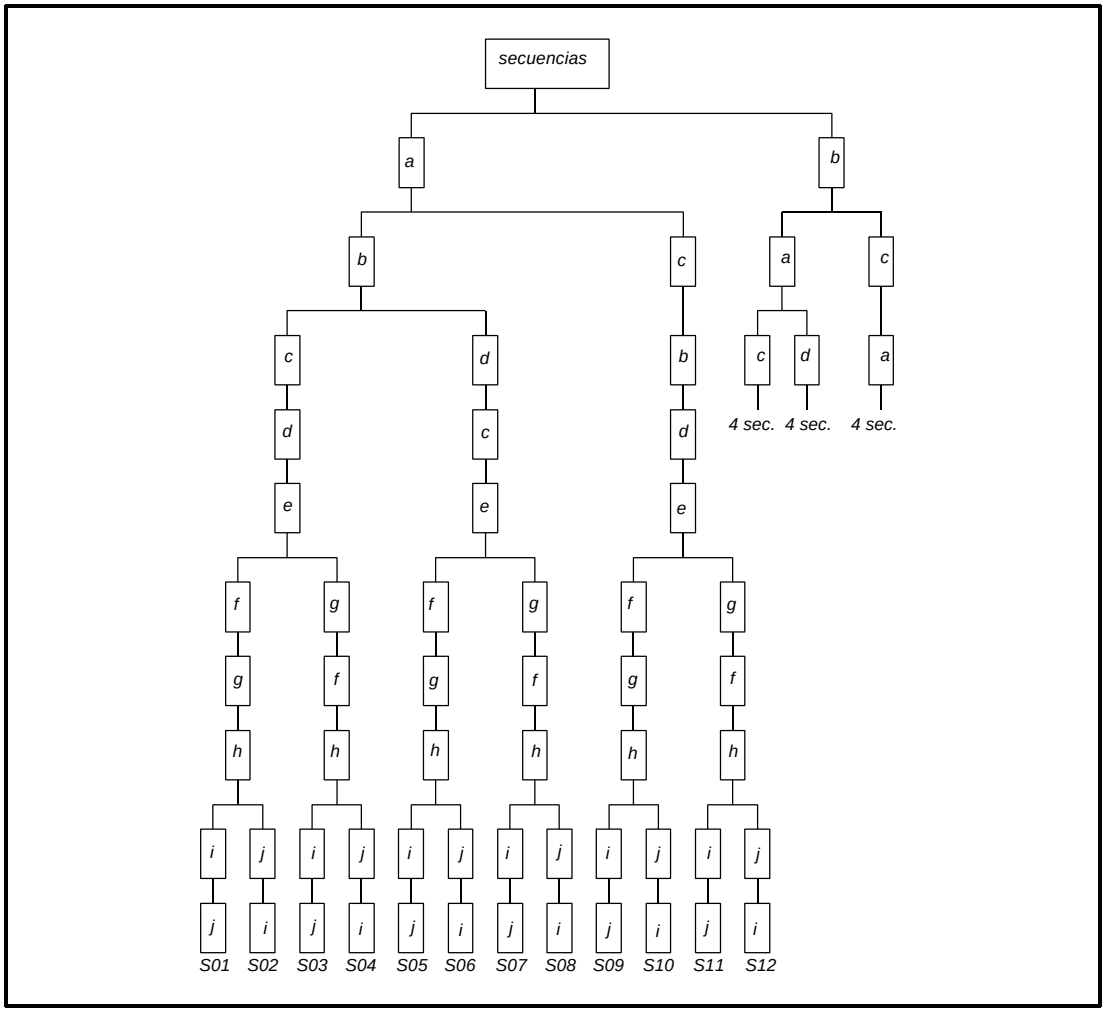


Fig. 7.1.1.3 Secuencias de tareas en los órdenes inducidos por las precedencias

Las precedencias introducen limitaciones en el orden de realización de la tareas. En el ejemplo E-1 el número de posibles secuencias de las tareas compatibles con las precedencias son 24 (véase la figura 7.1.1.3) en lugar de las 10! = 3.628.800 iniciales. Una reducción tan drástica no se realizará en casos industriales con menor proporción de ligaduras de este tipo. Cada secuencia puede traducirse en una asignación de tareas a estaciones: se asignarán en el orden de la secuencia las tareas a una estación mientras la suma de sus duraciones no supere el tiempo ciclo y cuando una tarea no "quepa" en la estación en curso se pasará a la estación siguiente. Actuando de esta forma con las 24 secuencias posibles encontraríamos una o varias asignaciones óptimas con el número mínimo de estaciones. Este procedimiento resulta inviable cuando el número de secuencias posibles es demasiado elevado, aunque el concepto de secuencia está implícito en la mayoría de procedimientos especialmente en aquellos con *backtracking* **como el MALB.**

7.1.1.1 Modelos analíticos

Dos modelos muy simples son el de White (1961) y el de Thangavelu & Shetty (1971). El primero se formula de la siguiente forma:

$$[MIN]\, z = \sum_{i \in F} \sum_{j = NME+1}^{NMAX} k_j \cdot x_{i,j} \qquad (1)$$

$$\sum_{j=1}^{NMAX} x_{i,j} = 1 \qquad 1 \le i \le n \qquad (2)$$

$$\sum_{i=1}^{n} p_i \cdot x_{i,j} \le C \qquad 1 \le j \le NMAX \qquad (3)$$

$$x_{i,l} \le \sum_{j=1}^{l} x_{h,j} \qquad \begin{matrix} 1 \le i \le n \\ 1 \le l \le NMAX \\ h \in G^{-1} i \end{matrix} \qquad (4)$$

$$x_{i,j} \in \{0, 1\} \qquad \text{a todo i, j} \qquad (5)$$

donde:

$x_{i,j}$ **es una variable que vale 1 si la tarea i se asigna a la estación** j **y 0 en caso contrario.**

F **es el conjunto de las tareas finales (sin siguientes).**

$NMAX$ **es el número máximo de estaciones consideradas en el modelo (que puede**

coincidir, por ejemplo, con las obtenidas mediante un método heurístico), con $NMAX > NME$**.**

k_j **es el coeficiente de penalización de la estación adicional, respecto a** NME**,** j**; debe garantizarse a través de estos coeficientes que la estación** $j+1$ **sólo se empleará si con las estaciones de 1 a** j **no ha sido suficiente para asignar todas las tareas, por lo que** $k_{j+1} = M.k_j$**, donde** M **es un número positivo elevado.**

$G^{-1}i$ **es el conjunto de los precedentes inmediatos de** i**.**

El significado de la función económica y de las restricciones es muy simple:

(5) implica que las $x_{i,j}$ **sólo adoptarán como valores 0 o 1.**

(2) obliga a que toda tarea sea asignada a una estación y sólo a una,

(3) explicita que la suma de las duraciones de las tareas asignadas a una estación no debe superar el tiempo ciclo,

(4) impone que una tarea no sea asignada a una estación anterior a las estaciones a que están asignadas sus precedentes,

(1) penaliza la utilización de más estaciones que el mínimo teórico NME**.**

Thangavelu & Shetti proponen un cambio de los coeficientes de la función económica pues los del modelo de White crecen tanto que pueden ocasionar inestabilidad numérica. Los coeficientes propuestos son.

$$k_{i,j} = p_i \cdot \left(1 + \sum_{h \in F} p_h\right)^{(j - NME - 1)} \qquad i \in F;\ NME + 1 \le j \le NMAX$$

$$k_{i,j} = 0 \qquad \text{en caso contrario}$$

También substituyen las relaciones (4) por la más compacta:

$$\sum_{j=1}^{NMAX} (NMAX - j + 1) \cdot (x_{i,j} - x_{h,j}) \le 0 \qquad 1 \le i \le n,\ h \in G^{-1}i \tag{6}$$

Siendo la nueva función económica:

$$z = \sum_{i=1}^{n} \sum_{j=1}^{NMAX} k_{i,j} \cdot x_{i,j} \qquad (7)$$

Existen otras formulaciones más sofisticadas.

7.1.2 Procedimientos heurísticos

Dada la reducida dimensión de los problemas abordables mediante los modelos analíticos, es preciso recurrir en la mayoría de los casos industriales a procedimientos heurísticos.

7.1.2.1 Método de Helgeson & Birnie

El procedimiento de Helgeson & Birnie, como otros muchos, pretende, mediante unos pesos o índices de prioridad, elegir una de las secuencias, permitiendo en la asignación ligeras modificaciones de la misma (paso a secuencias vecinas). El peso establecido por Helgeson & Birnie para una tarea *i* **es la suma de su duración más la de todas las tareas que la siguen. En nuestro ejemplo E-1, la tarea** *e* **tiene como siguientes (inmediatas o no)** *f, g, h, i, j***; por tanto, su peso será**

$$w_e = 2 + 4 + 3 + 5 + 2 + 3 = 19$$

En la figura 7.1.2.1 hemos resumido los datos relativos al ejemplo, incluyendo los pesos, habiendo ordenado las tareas en orden decreciente de los mismos. La secuencia deducida del orden de pesos decreciente es la que hemos designado como S06 en la figura 7.1.1.3 (o la que designaríamos por S18 si resolvieramos el empate entre *a* **y** *b* **a favor de esta última).**

i	p_i	w_i	prec. inmed.
a	**5**	**29**	**-**
b	**4**	**29**	**-**
d	**6**	**25**	*b*
c	**5**	**24**	*a*
e	**2**	**19**	*a,b*
f	**4**	**14**	*e*
g	**3**	**13**	*e*
h	**5**	**10**	*g,h*
j	**3**	**3**	*h*
i	**2**	**2**	*h*

Fig. 7.1.2.1 Datos del ejemplo E-1 ordenados por pesos

El algoritmo de Helgeson & Birnie consta de los siguientes pasos:

Paso 1. *Inicialización*. **Se abre la estación 1 y se le asigna el ciclo como tiempo disponible (**$TD = C$**).**

Paso 2. *Busca de candidatos*. **Sea** j **la estación abierta, y** TD **el tiempo disponible. Se establece una lista de tareas candidatos a ser asignados a la estación** j**. Para ello la tarea debe cumplir las tres condiciones siguientes:**

condición 1 : no haber sido asignada todavía,
condición 2 : tener todas sus precedentes inmediatas asignadas a una estación (la j **o anteriores),**
condición 3 : tener una duración inferior o igual a TD**.**

Paso 3. *Test de cierre*. **Si la lista de candidatos es vacía,** *ir al paso 6*.

Paso 4. *Asignación de tareas*. **Si hay una sola tarea candidato asignarla directamente a la estación** j**; si hay varias asignar a la estación** j **la tarea** i **de mayor peso** w_i **de la lista de candidatos.**

Paso 5. *Actualización*. **Reducir el tiempo disponible** TD **en** p_i**; si** TD **es nulo (o inferior al menor valor** p_i **existente),** *ir al paso 6*; **en caso contrario,** *ir al paso 2*.

Paso 6. *Cierre de estación*. **Cerrar la estación** j**; el tiempo disponible restante después de cerrar la estación es el tiempo muerto de la misma, (que iremos acumulando para obtener el tiempo muerto total),**

Paso 7. *Bucle*. **Si todas las tareas están asignadas** *fin del algoritmo*; **en caso contrario abrir la estación** $j + 1$ **y asignarle como tiempo disponible el ciclo,** *ir al paso 2*.

La aplicación del algoritmo anterior al ejemplo E-1 para el ciclo $C = 10$ **se realiza a través del desarrollo de la figura 7.1.2.2, los resultados son:**

$N = 5$ **estaciones**	**ESTACIÓN 1 {** a , b **}**	**ocupación 9**
	ESTACIÓN 2 { d **}**	**ocupación 6**
	ESTACIÓN 3 { c , e , g **}**	**ocupación 10**
	ESTACIÓN 4 { f , h **}**	**ocupación 9**
	ESTACIÓN 5 { j , i **}**	**ocupación 5**

Tiempo muerto total = 11

Las tareas en las estaciones se han indicado en el orden de asignación, su secuencia corresponde a la designada por S08 en la figura 7.1.1.3

Hemos obtenido un número de estaciones superior al número mínimo (5 frente a 4); por tanto, la solución es potencialmente mejorable. Helgeson y Birnie recomiendan la mejora del equilibrado mediante observación visual y permutación de tareas entre estaciones. En este caso es fácil, dado que es obvio que la dificultad fundamental se centra en la estación 2, cuyo tiempo muerto es excesivo (ya que la estación 5, que tiene mayor tiempo muerto, aparece como consecuencia de las asignaciones anteriores). Para mejorar la solución deberemos corregir la asignación realizada en la estación 1. La secuencia S10 nos conduciría a una solución óptima:

N **= 4 estaciones** **ESTACIÓN 1 {** *a , c* **} ocupación 10**
ESTACIÓN 2 { *b , d* **} ocupación 10**
ESTACIÓN 3 { *e , f , g* **} ocupación 9**
ESTACIÓN 4 { *h , j , i* **} ocupación 10**

Tiempo muerto total = 1

Estación	*TD*	Candida-tos	Elegido	p_i	*TM*	Nº línea
1	10 5 1	*a,b* *b,c* -	*a* *b*	5 4	 1	1 2 3
2	10 4	*d,c* -	*d*	6	 4	4 5
3	10 5 3 0	*c* *e* *g*	*c* *e* *g*	5 2 3	 0	6 7 8 9
4	10 6 1	*f* *h* -	*f* *h*	4 5	 1	10 11 12
5	10 7 5	*j,i* *i* -	*j* *i*	3 2	 5	13 14 15

Fig. 7.1.2.2 Aplicación del algoritmo de Helgeson & Birnie al ejemplo E-1

Otro ejemplo (E-2), con 20 tareas, de duraciones, precedencias y pesos reproducidos en la tabla de la figura 7.1.2.3, y grafo de precedencias en la 7.1.2.4 permitirá profundizar en el algoritmo. Vamos a proceder a determinar el equilibrado de la línea de montaje para un tiempo ciclo C = 12 minutos.

i	p_i	w_i	prec.	i	p_i	w_i	prec.	i	p_i	w_i	prec.
2	5	39	-	9	3	19	5	12	4	9	8
1	4	38	-	10	4	16	6	15	3	5	11
5	4	34	1,2	13	6	16	9	16	2	5	11,12
4	5	26	-	11	5	15	7	20	4	4	17
3	6	25	-	14	2	12	10	19	3	3	16
7	6	21	4	8	2	11	5	18	2	2	15
6	3	19	3	17	6	10	13,14				

Fig. 7.1.2.3 Tareas del ejemplo E-2 ordenadas por pesos crecientes

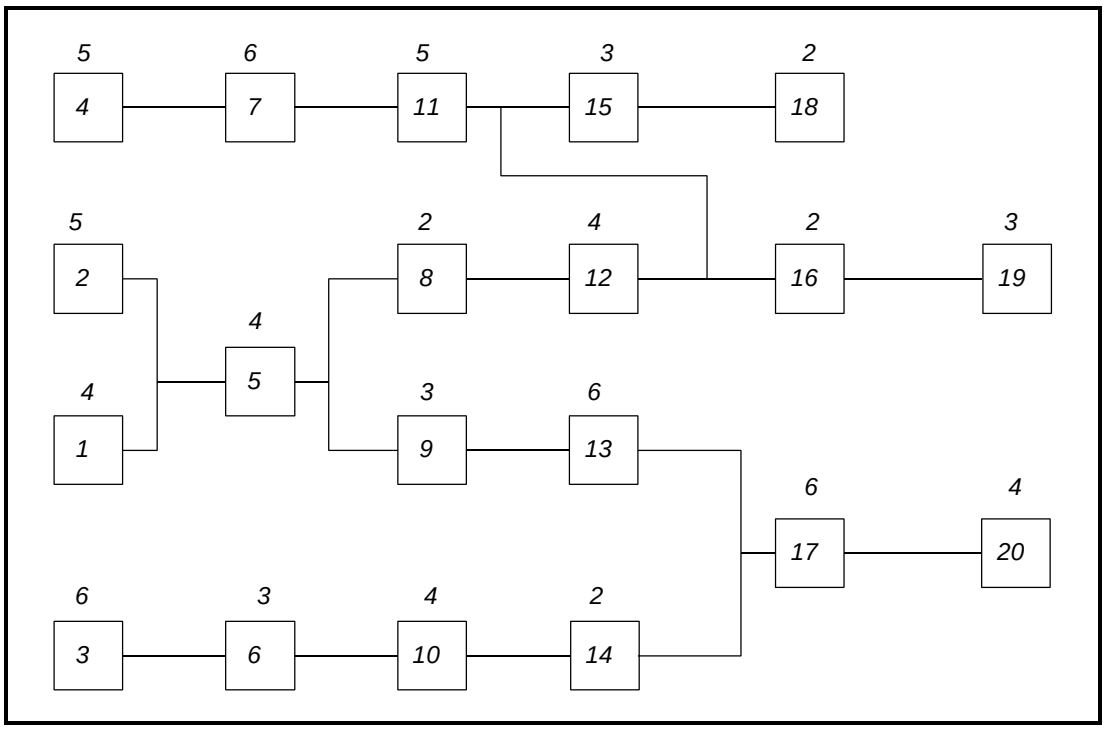


Fig. 7.1.2.4 Grafo de precedencias correspondiente a las tareas del ejemplo E-2

La suma de duraciones es $\sum p_i$ = 79 minutos, y la duración mayor es max { p_i } = 6

minutos; en consecuencia, salvo estaciones con varios trabajadores, los ciclos posibles a considerar están entre 6 y 79 minutos, lo que cumple el indicado.

En nuestro caso:

$$\frac{\sum p_i}{C} = \frac{79}{12} = 6{,}58 \rightarrow NME = 7$$

$$EMI = \frac{79}{7 \times 12} = 0{,}95$$

Estación	*TD*	Candidatos	Elegido	p_i	*TM*	Nº línea
1	12 7 3	2,1,3,4 1,4,3 -	2 1	5 4	 3	1 2 3
2	12 8 3 0	5,4,3 4,3,9,8 9,8	5 4 9	4 5 3	 0	4 5 6 7
3	12 6 0	3,7,13,8 7,6,13,8	3 7	6 6	 0	8 9 10
4	12 9 5 0	6,13,11,8 10,13,11,8 11,14,8	6 10 11	3 4 5	 0	11 12 13 14
5	12 6 4 2	13,14,8,15 14,8,15 8,15	13 14 8	6 2 2	 2	15 16 17 18
6	12 6 2 0	17,12,15 12,15,20 16 -	17 12 16	6 2 2	 0	15 16 17 18
7	12 9 5 2 0	15,20,19 20,19,18 19,18 18	15 20 19 18	3 4 3 2	 0	23 24 25 26 27

Fig. 7.1.2.5 Aplicación del algoritmo de Helgeson & Birnie al ejemplo E-2 con C = 12

Por tanto, una asignación que nos conduzca a 7 estaciones será óptima en cuanto a la minimización del tiempo muerto. Una asignación con 7 estaciones la hemos alcanzado aplicando el algoritmo tal como se detalla en la figura 7.1.2.5.

N = 7 estaciones	**ESTACIÓN 1 { 2 , 1 }**	**ocupación 9**
	ESTACIÓN 2 { 5 , 4 , 9 }	**ocupación 12**
	ESTACIÓN 3 { 3 , 7 }	**ocupación 12**
	ESTACIÓN 4 { 6 , 10 , 11 }	**ocupación 12**
	ESTACIÓN 5 { 13 , 14 , 8 }	**ocupación 10**
	ESTACIÓN 6 { 17 , 12 , 16 }	**ocupación 12**
	ESTACIÓN 7 { 15 , 20 , 19 , 18 }	**ocupación 12**

Tiempo muerto total = 5

En el mismo ejemplo E-2 la aplicación del algoritmo con el ciclo C = 10 no proporciona el mínimo de estaciones:

$$\frac{\sum p_i}{C} = \frac{79}{10} = 7{,}9 \rightarrow NME = 8$$

$$EMI = \frac{79}{8 \times 10} = 0{,}987$$

Los cálculos se han desarrollado en la figura 7.1.2.6 habiendo sido necesaria una estación más. Como veremos más adelante existen asignaciones con ciclo 10 y 8 estaciones, por tanto el algoritmo no ha obtenido el óptimo.

N = 9 estaciones	**ESTACION 1 { 2 , 1 }**	**ocupación 9**
	ESTACION 2 { 5 , 4 }	**ocupación 9**
	ESTACION 3 { 3 , 6 }	**ocupación 9**
	ESTACION 4 { 7 , 9 }	**ocupación 9**
	ESTACION 5 { 10 , 13 }	**ocupación 10**
	ESTACION 6 { 11 , 14 , 8 }	**ocupación 9**
	ESTACION 7 { 17 , 12 }	**ocupación 10**
	ESTACION 8 { 15 , 16 , 20 }	**ocupación 9**
	ESTACION 9 { 19 , 18 }	**ocupación 5**

Tiempo muerto total : 11

Eficiencia : 0,8777

Estación	*TD*	Candidatos	Elegido	p_i	*TM*	Nº paso
1	10	2,1,3,4	2	5		1
	5	1,4,3	1	4		2
	1	-			1	3
2	10	5,4,3	5	4		4
	6	4,3,9,8	4	5		5
	1	-			1	6
3	10	3,7,9,8	3	6		7
	4	6,9,8	6	3		8
	1	-			1	9
4	10	7,9,10,8	7	6		10
	4	9,10,11,8	9	3		11
	1	-			1	12
5	10	10,13,11,8	10	4		13
	6	13,11,14,8	13	6		14
	0				0	15
6	10	11,14,8	11	5		16
	5	14,8,15	14	2		17
	3	8,15	8	2		18
	1	-			1	19
7	10	17,12,15	17	6		20
	4	12,15,20	12	4		21
	0				0	22
8	10	15,16,20	15	3		23
	7	16,20,18	16	2		24
	5	20,19,18	20	4		25
	1	-			1	26
9	10	19,18	19	3		27
	7	18	18	2		28
	5	-			5	29

Fig. 7.1.2.6 Aplicación del algoritmo de Helgeson & Birnie al ejemplo E-2 con c = 10

Podríamos haber formalizado el método de Helgeson & Birnie de una manera más simple, pero hemos preferido establecer un esquema de alcance más general. La sucesión de los siete pasos indicados es adaptable a otros procedimientos. Si en lugar de utilizar un

índice de prioridad basado en w_i queremos emplear uno distinto (basado en el número de siguientes de cada tarea, su duración, etc.) o bien un índice resultante de la ponderación de varios índices simples, bastará modificar en consecuencia el paso 4.

El comportamiento del algoritmo de Helgeson & Birnie en algunos ejemplos simples nos sugiere que w_i es un índice de prioridad adecuado para la asignación de las primeras tareas a una estación pero no tanto para la asignación de las últimas. Por consiguiente, podrían ser interesantes procedimientos que establecieran las reglas de prioridad a utilizar en la próxima asignación de una tarea a una estación en función del estado de carga (proporción del ciclo ya asignado) de la misma. Una variante de esta idea la veremos más adelante en el algoritmo de Boctor.

7.1.3 Mejora del equilibrado obtenido mediante simulación

Es fácil adaptar el procedimiento anterior a una búsqueda de soluciones por simulación, guardando la mejor hallada hasta el momento. En esencia basta cambiar el paso 4:

Paso 0. *Inicio proceso*. **Se fija el número de simulaciones y se establece como solución incumbente una determinada previamente con un procedimiento heurístico (eventualmente la hallada en la primera simulación).**

Paso 1. *Inicialización*. **Se abre la estación 1 y se le asigna el ciclo como tiempo disponible (**$TD = C$**).**

Paso 2. *Busca de candidatos*. **Sea** j **la estación abierta, y** TD **el tiempo disponible. Se establece una lista de tareas candidatas a ser asignadas a la estación** j**. Para ello la tarea debe cumplir las tres condiciones siguientes:**

condición 1 : no haber sido asignada todavía,
condición 2 : tener todas sus precedentes inmediatas asignadas a una estación (la j o anteriores),
condición 3 : tener una duración inferior o igual a TD**.**

Paso 3. *Test de cierre*. **Si la lista de candidatos es vacía,** *ir al paso 6*.

Paso 4. *Asignación de tareas*. **Si hay una sola tarea candidata asignarla directamente a la estación** j**; si hay varias asignar a la estación** j **la tarea** i **elegida al azar de la lista de candidatos.**

Paso 5. *Actualización*. **Reducir el tiempo disponible** TD **en** p_i**; si** TD **es nulo (o inferior al**

menor valor p_i **existente),** *ir al paso 6***; en caso contrario,** *ir al paso 2***.**
Paso 6. *Cierre de estación***. Cerrar la estación** *j***, el tiempo disponible restante después de cerrar la estación es el tiempo muerto de la misma, (que iremos acumulando para obtener el tiempo muerto total). Si el tiempo muerto agregado supera o iguala el total del tiempo muerto de la solución incumbente cancelar la simulación en curso,** *ir al paso 8***.**

Paso 7. *Bucle***. Si todas las tareas están asignadas,** *ir al paso 8***; en caso contrario, abrir la estación** *j + 1* **y asignarle como tiempo disponible el ciclo,** *ir al paso 2***.**

Paso 8. *Iteración***. Si la solución hallada es mejor que la incumbente substituirla; si la solución incumbente tiene el número mínimo teórico de estaciones o se han realizado todas las simulaciones prescritas** *fin del algoritmo***, en caso contrario,** *ir al paso 1***.**

Para el ejemplo E-2 y utilizando un sencillo programa escrito en BASIC para un microordenador PC hemos obtenido varias soluciones óptimas (en explotaciones diferentes). Las soluciones indican en cada estación las tareas por orden de asignación.

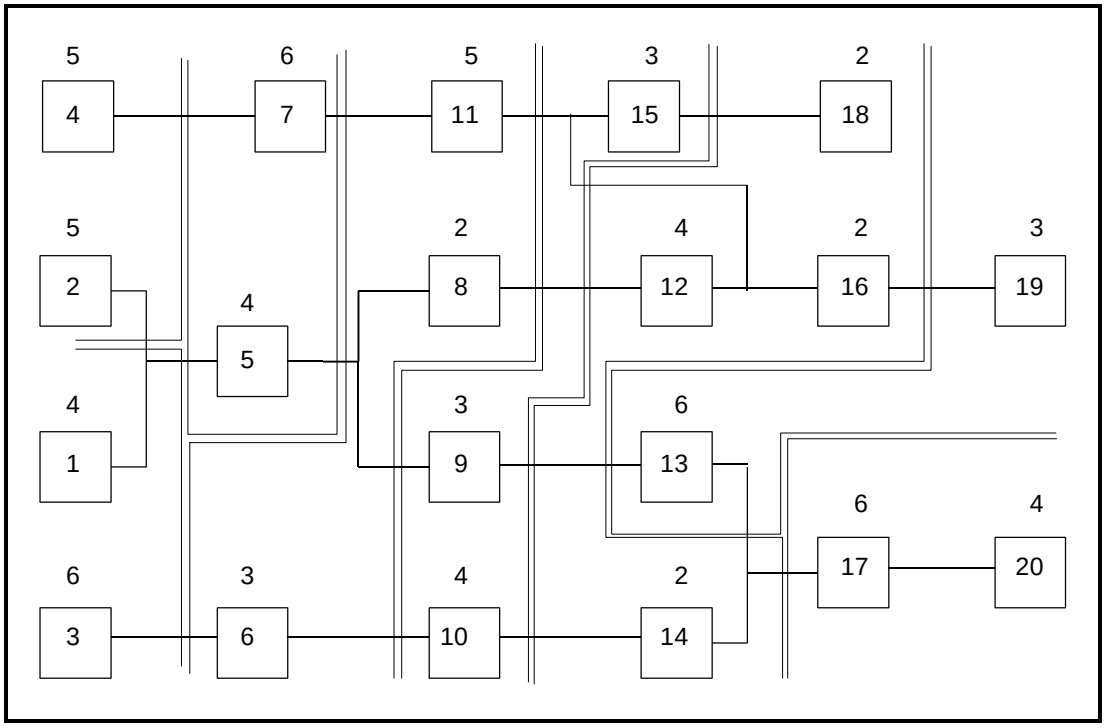


Fig. 7.1.3.1 Agrupación, para el ejemplo E-1, de tareas en estaciones correspondiente a la solución óptima 1

Solución óptima 1.

N = 8 estaciones	**ESTACIÓN 1 : { 4 , 2 }**	**ocupación 10**
	ESTACION 2 : { 3 , 1 }	**ocupación 10**
	ESTACIÓN 3 : { 7 , 5 }	**ocupación 10**
	ESTACIÓN 4 : { 11 , 8 , 6 }	**ocupación 10**
	ESTACIÓN 5 : { 9 , 10 , 15 }	**ocupación 10**
	ESTACIÓN 6 : { 18 , 12 , 14 , 16 }	**ocupación 10**
	ESTACIÓN 7 : { 13 , 19 }	**ocupación 9**
	ESTACIÓN 8 : { 17 , 20 }	**ocupación 10**

Tiempo muerto total : 1 minuto
Eficiencia : 0,9875

La elección al azar, cuando hay varios candidatos, puede efectuarse dando a cada uno idéntica probabilidad (como se ha realizado en la aplicación real), o bien una probabilidad proporcional a cierto peso (w_i o alguno de los descritos anteriormente).

7.1.4 Otros algoritmo heurísticos

Otra familia de algoritmos razonablemente eficientes lo constituyen los de exploración arborescente (semejantes a los utilizados en inteligencia artificial) tales como el MALB (E. M. Mansoor, "Assembly Line Balancing: An Improvement on the Ranked Positional Weight Technique", *The Journal of Industrial Engineering***, vol. 15, n. 2, 1964), pero su exposición detallada excede los límites del presente trabajo. Nos limitaremos a describir el algoritmo de Bedworth y el algoritmo de Boctor.**

7.1.4.1 Algoritmo de Bedworth

La simplificación desarrollada por D. D. Bedworth utiliza además de un principio de exploración (*backtracking* **reducido) ideas tomadas de Kilbridge & Wester y consiste en:**

Paso 1. Desarrollar el grafo de precedencias en la forma habitual.

Paso 2. Asignar niveles a las tareas de acuerdo con las precedencias de forma que en el último nivel se sitúen las tareas que no preceden a ninguna otra; en caso de indeterminación asignar las tareas al nivel más alto posible. Esto favorecerá que las tareas con pocas siguientes serán consideradas con posterioridad a las que tengan muchas.

Paso 3. Dentro de cada nivel ordenar las tares por duración decreciente. Esto favorecerá que las tareas largas se consideren antes que las cortas, permitiendo un mejor aprovechamiento del tiempo ciclo.

Paso 4. Asignar las tareas a las estaciones, de acuerdo a las precedencias y al remanente de tiempo ciclo en el orden indicado:

a) primero el nivel más bajo,
b) dentro de un nivel primero la tarea de mayor duración,

Paso 5. Cuando se haya terminado la asignación de tareas a una estación considerar si su ocupación es aceptable. Si no es así comprobar todas las tareas cuyas relaciones de precedencia se han satisfecho. Determinar si es posible substituir una o varias de las tareas asignadas a la estación por las consideradas (cumpliendo las precedencias) aumentando la utilización de la estación. Si es así, se realiza la substitución. Cuando no sea posible efectuar ninguna substitución que aumente la utilización de la estación la asignación se da como definitiva y se pasa a la estación siguiente.

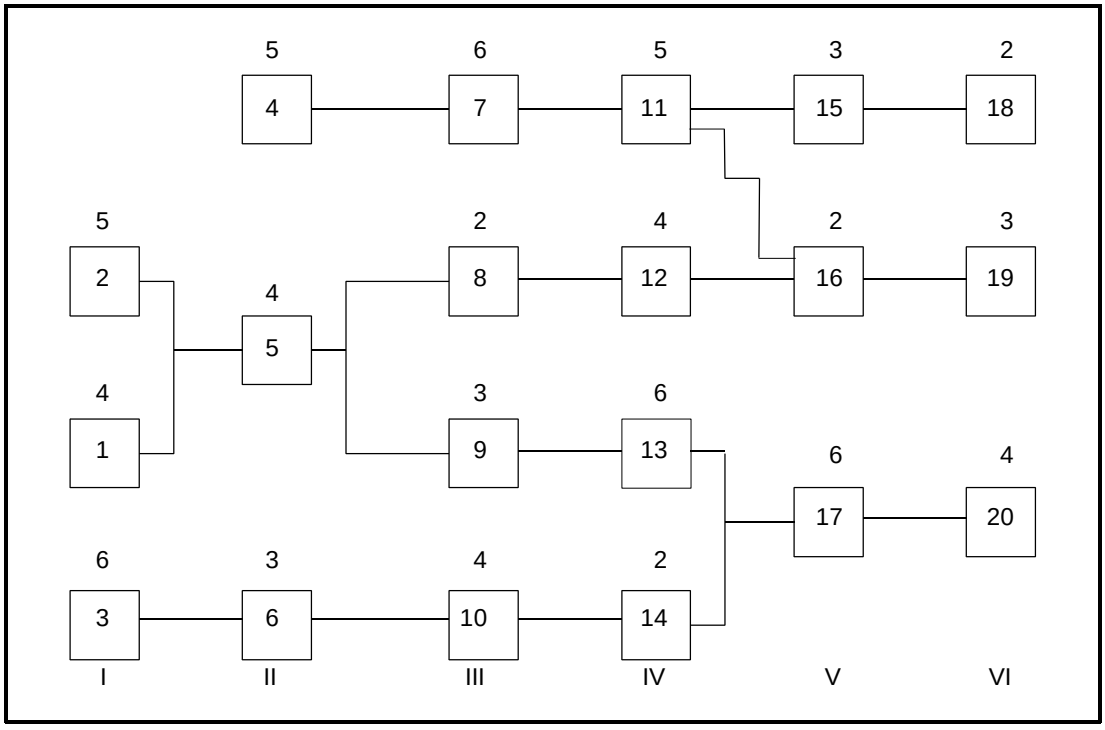


Fig. 7.1.4.1 Grafo de precedencias correspondiente a las tareas del ejemplo E-2, ordenadas por niveles

La aplicación del algoritmo conduce a:

NIVEL I	**3 (6)**	**2 (5)**	**1 (4)**	
NIVEL II	**4 (5)**	**5 (4)**	**6 (3)**	
NIVEL III	**7 (6)**	**10 (4)**	**9 (3)**	**8 (2)**
NIVEL IV	**13 (6)**	**11 (5)**	**12 (4)**	**14 (2)**
NIVEL V	**17 (6)**	**15 (3)**	**16 (2)**	
NIVEL VI	**20 (4)**	**19 (3)**	**18 (2)**	

Fig. 7.1.4.2 Tareas del ejemplo E-2 ordenadas por niveles y duración

ESTACIÓN 1: { 1 , 3 } **ocupación 10**
ESTACIÓN 2: { 2 , 4 } **ocupación 10**
ESTACIÓN 3: { 5 , 6 , 9 } **ocupación 10**
ESTACIÓN 4: { 7 , 10 } **ocupación 10**
ESTACIÓN 5: { 8 , 13 , 14 } **ocupación 10**
ESTACIÓN 6: { 11 , 12 } **ocupación 9**

quedan disponibles 15, 16 y 17; [1] **cambiamos 11 por 17**

{ 17 , 12 } **ocupación 10**

ESTACIÓN 7: { 11 , 15 , 16 } **ocupación 10**
ESTACIÓN 8: { 20 , 19 , 18 } **ocupación 9**

no queda ninguna tarea por asignar

[1] **15 está disponible bajo la condición de asignar 11 a la estación 6, 16 de asignar 11 y 12 (por tanto no es candidata al intercambio), 17 no está condicionada a la asignación a la estación 6 ni de 11 ni de 12.**

Esta solución es óptima, aunque diferente a la indicada en 7.1.3.

7.1.4.2 Algoritmo de Boctor

Fayez F. Boctor ha propuesto recientemente un algoritmo ("A multiple rule heuristic for assembly line balancing", *Journal of the Operational Research Society***, vol. 46, nº 1, pág. 62-69, 1995) que combina varias reglas simples. Inicialmente presentaremos dos definiciones que serán utilizadas en la formulación de las reglas:**

- una *tarea dura* **es una tarea cuya duración es igual o mayor a la mitad del tiempo ciclo,**

- una tarea se llama *candidato condicionado por la tarea i* **si se convierte o permanece como candidato después de asignar la tarea** *i*; **si el asignar** *i* **a la estación en curso reduce el tiempo ciclo restante a 0, se llama** *candidato condicionado* **a una tarea candidato para la siguiente estación.**

El algoritmo utiliza los pasos descritos en 7.1.2.1 modificando el cuarto que tomará la forma:

Paso 4. *Asignación de tareas*. **Si hay una sola tarea candidato asignarla directamente a la estación** *j*; **si hay varias asignar a la estación** *j* **la tarea** *i* **de de acuerdo a las siguientes reglas:**

R1: una tarea cuya duración sea igual al tiempo ciclo restante *TD*. **Si no existe ninguna ir a la siguiente regla. Para deshacer los empates asignar la tarea con más candidatos condicionados.**

R2: una tarea dura con el mayor número de candidatos condicionados. Si no existe ninguna ir a la siguiente regla. Para deshacer los empates elegir la tarea con mayor duración.

R3: una combinación de dos tareas con duración igual al ciclo restante TD. Si no existe esta combinación ir a la siguiente regla. Para deshacer los empates elegir la pareja con mayor número de candidatos condicionados.

R4: una tarea con el mayor número de candidatos condicionados. Para deshacer los empates elegir la tarea con el mayor número de siguientes inmediatos duros y si persiste la de mayor duración.

La aplicación del algoritmo al ejemplo E-2 para el ciclo 10 se detalla en la figura 7.1.4.3; entre paréntesis se indica si la tarea es dura y el número de candidatos condicionados. Se alcanza la siguiente solución óptima (que no coincide con ninguna de las anteriores):

N = 8 estaciones	**ESTACIÓN 1 : { 3 , 1 }**	**ocupación 10**
	ESTACIÓN 2 : { 2 , 4 }	**ocupación 10**
	ESTACIÓN 3 : { 7 , 5 }	**ocupación 10**
	ESTACIÓN 4 : { 11 , 6 , 8 }	**ocupación 10**
	ESTACIÓN 5 : { 9 , 13 }	**ocupación 9**
	ESTACIÓN 6 : { 10 , 12 , 14 }	**ocupación 10**
	ESTACIÓN 7 : { 17 , 20 }	**ocupación 10**
	ESTACIÓN 8 : { 15 , 16 , 19 , 18}	**ocupación 10**

Tiempo muerto total : 1 minuto
Eficiencia : 0,9875

Estación	*TD*	Candidatos	Regla	Elegidos	p_i	*TM*
1	10	1(3), 2(D,2), 3(D,2), 4(D,2)	R2	3	6	
	4	1(3), 6(0)	R1	1	4	0
2	10	2(D,3), 4(D,2), 6(3)	R2	2	5	
	5	4(D,3), 5(0), 6(0)	R1	4	5	0
3	10	5(4), 6(3), 7(D,2)	R2	7	6	
	4	5(4), 6(0)	R1	5	4	0
4	10	6(4), 8(4), 9(4), 11(D,4)	R2	11	5	
	5	6(1), 8(3), 9(1), 15(1)	R3 (1)	6,8	3 + 2	0
5	10	9(4), 10(4), 12(4), 15(4)	R4	9	3	
	7	10(2), 12(2), 13(D,0), 15(3)	R2	13	6	
	1					1
6	10	10(3), 12(3), 15(3)	R4	10	4	
	6	12(2), 14(2), 15(2)	R3 (2)	12,14	4 + 2	0
7	10	15(3), 16(3), 17(D,2)	R2	17	6	
	4	15(0), 16(0), 20(2)	R1	20	4	0
8	10	15(2), 16(2)	R4	15	3	
	7	16(2), 18(1)	R4	16	2	
	5	18(1), 19(1)	R4	19	3	
	2	18(0)	R1	18	2	0

(1) **Parejas candidato** **6,8 (4)**
8,9 (4)
8,15 (4)
15,18 (3)

(2) **Parejas candidato** **12,14 (3)**
12,16 (3)

Fig. 7.1.4.3 Aplicación del algoritmo de Boctor al ejemplo E-2 con ciclo 10

7.1.5 Líneas mixtas

En las líneas mixtas, en las que circulan unidades de diferentes productos o modelos, ocurrirá usualmente que las duraciones de las tareas dependan del producto concreto sobre el que deban aplicarse. Sea *I* el número de productos distintos asociados a la línea y α_i ($i = 1,2,..,I$) la proporción de unidades del producto *i* contenidas en el programa de producción. Llamemos $p_{k,i}$ la duración de la tarea *k* aplicada al producto *i*. Una manera de reducir el problema de equilibrado a la situación estudiada exigirá la obtención de un valor equivalente único para cada tarea, que, normalmente, podrá oscilar entre los dos valores extremos siguientes:

$$\max_i \{p_{k,i}\}$$

y

$$\sum_{i=1}^{I} \alpha_i \cdot p_{k,i}$$

En la medida en que el valor equivalente único esté más cerca del segundo, más importante será la secuenciación adecuada de las unidades que circulan en la línea.

No hemos tratado las ligaduras de zona en este texto. Los métodos analíticos, incluidos los similares a los heurísticos que hemos descrito, no son suficientemente eficientes para el tratamiento de este tipo de restricciones, aunque en ciertos casos pueden serlo la construcción de soluciones mediante simulación, la exploración arborescente con *backtracking* **y ciertos procesos interactivos.**

7.1.6 Secuenciación de unidades en una línea

Una vez equilibrada la línea, si todas las unidades que circulan por ella son idénticas, no existe ninguna dificultad adicional. Sin embargo lo más habitual es que dichas unidades sean similares, pero que posean algunas características distintivas: no exijan exactamente la misma carga de trabajo en todas las estaciones o los componentes a incorporar en las distintas estaciones sean distintos en calidad y/o en número. Si la línea se ha equilibrado teniendo en cuenta los valores medios de dichas características, será importante secuenciar adecuadamente las unidades con la finalidad de que no se produzcan grandes divergencias puntuales entre los dichos valores medios y los reales.

Si varias unidades "ricas" respecto a la carga de trabajo en cierta estación se encuentran muy próximas en la secuencia, en dicha estación el operario no tendrá tiempo de atenderlas a todas (aunque a lo largo de la jornada se compense el defase en carga) por lo que en las últimas de dicho tramo de la secuencia no se efectuarán todos los elementos de trabajo y se deberá proceder a su terminación fuera de línea con los problemas de coste y calidad que ello comporta.

Por otra parte en un contexto JIT interesa regularizar el flujo de componentes, que es la condición obligada para la reducción de stocks. Si la secuencia de unidades introduce en forma puntual grandes divergencias entre el consumo medio y el real sólo podrán soportarse mediante un incremento de los stocks a pie de línea.

Podemos formular dos enfoques diferentes de para alcanzar la regularidad de la secuencia:

- equibrado en función de las tasas de los productos secuenciados,
- equilibrado en función de tasas de las necesidades de los productos secuenciados (cargas o componentes),

7.1.7 Equilibrado de las tasas de los productos

Deseamos secuenciar en forma regular cierto número de unidades de diferentes tipos. Utilizaremos la siguiente nomenclatura:

I	**número de productos (tipos) diferentes a secuenciar;**
i	**producto genérico (**$i = 1, 2, \ldots, I$**);**
u_i	**número de unidades del producto i a secuenciar;**
$K = \sum u_i$	**total de unidades a secuenciar.**

Para un artículo i**, la tasa de producción (o montaje) es:**

$$r_i = \frac{u_i}{K}$$

En una posición k **de la secuencia el número real de unidades secuenciadas (desde la posición 1 a la** k**) será** $X_{i,k}$**, cuando en una secuencia ideal, completamente regular, dicho número debería ser** $k \cdot r_i$**. La regularidad implica que para toda** i **y toda** k **los valores** $X_{i,k}$ **sean lo más parecidos posible a los** $k \cdot r_i$**. Traducir la regularidad a una expresión matemática presenta indudables dificultades dado que se trata de un concepto intuítivo con un notable grado de ambigüedad. Miltenburg ha propuesto las siguientes:**

$$[MIN]\ z_1 = \sum_{k=1}^{K} \sum_{i=1}^{I} \left(\frac{X_{i,k}}{k} - r_i\right)^2$$

$$[MIN]\ z_2 = \sum_{k=1}^{K} \sum_{i=1}^{I} \left(X_{i,k} - k \cdot r_i\right)^2 = SDQ$$

$$[MIN]\ z_3 = \sum_{k=1}^{K} \sum_{i=1}^{I} \left|\frac{X_{i,k}}{k} - r_i\right|$$

$$[MIN]\ z_4 = \sum_{k=1}^{K} \sum_{i=1}^{I} \left|X_{i,k} - k \cdot r_i\right| = SDR$$

a las que hemos añadido:

$$[MIN]\ z_5 = \sum_{k=1}^{K} \max_{i} \left|\frac{X_{i,k}}{k} - r_i\right|$$

$$[MIN]\ z_6 = \sum_{k=1}^{K} \max_i \left| X_{i,k} - k \cdot r_i \right| = SDM$$

Es obvio que z_1 y z_2 corresponden a una distancia cuadrática, z_3 y z_4 a una rectangular (o Manhattan) y z_5 y z_6 a una minimax (posiblemente sería más coherente denominarla distancia máxima, pero adoptamos la denominación utilizada usualmente). z_2, z_4 y z_6 miden la distancia entre valores absolutos mientras que z_1, z_3 y z_5 lo hacen entre valores relativos. Habitualmente emplearemos la primera forma. Las formulaciones indicadas no son las únicas imaginables, y además no reflejan necesariamente todas las circunstancias reales, en las que los objetivos pueden ser diferentes; sin embargo, son suficientes para abordar el problema.

Consideremos un programa matemático que pretendiese resolver la minimización de la distancia cuadrática (z_2):

$$[MIN]\ SDQ = \sum_{k=1}^{K} \sum_{i=1}^{I} \left(X_{i,k} - k \cdot r_i \right)^2 \qquad (1)$$

s.a

$$\sum_{i=1}^{I} X_{i,k} = k \qquad 1 \le k \le K \qquad (2)$$

$$X_{i,k} \le u_i \qquad (3)$$

$$0 \le X_{i,k} - X_{i,k-1} \le 1 \qquad (4)$$

$$X_{i,k} \ge 0 \text{ y entero} \qquad (5)$$

con (3), (4) y (5) extendidos a $1 \le k \le K$ y $1 \le i \le I$ (adoptando para todo i $X_{i,0} = 0$). La expresión (2) indica que hasta la posición k se han secuenciado exactamente k unidades, la (3) que no se secuencian más unidades de cierto tipo de las que indica el programa y (4) que en cada posición se secuencia una unidad y sólo una. Para un programa cuadrático entero de la dimensión que corresponde al problema en su vertiente industrial no disponemos de un algoritmo de resolución adecuado, lo que descarta esta vía. Más asequible sería la correspondiente a las distancias rectangular y minimax que conducirían a programas lineales mixtos; por ejemplo, para la primera:

$$[MIN]\ SDR = \sum_{k=1}^{K} \sum_{i=1}^{I} \left(s_{i,k} + t_{i,k} \right) \qquad (6)$$

s.a

$$\sum_{i=1}^{I} X_{i,k} = k \qquad 1 \le k \le K \tag{2}$$

$$X_{i,k} \le u_i \tag{3}$$

$$0 \le X_{i,k} - X_{i,k-1} \le 1 \tag{4}$$

$$X_{i,k} - s_{i,k} + t_{i,k} = k \cdot r_i \tag{7}$$

$$X_{i,k} \ge 0 \text{ y entero} \tag{5}$$

$$s_{i,k},\ t_{i,k} \ge 0 \tag{8}$$

con (3), (4), (7), (5) y (8) extendidos a $1 \le k \le K$ y $1 \le i \le I$ (adoptando para todo i $X_{i,0} = 0$). La expresión (7) calcula la diferencia en valor absoluto entre $X_{i,k}$ y $k.r_i$, que es $s_{i,k}$ o $t_{i,k}$ según $X_{i,k}$ sea mayor o menor que $k.r_i$. Vamos a concentrarnos en la distancia SDQ, que es la preferida por la mayoría de autores.

Volviendo a nuestro programa cuadrático, si relajáramos la formulación prescindiendo de las expresiones (4) encontraríamos un problema conocido. En efecto, en este caso los valores $X_{i,k}$ para diversos valores de k *no estarían ligados por ninguna restricción***. Por consiguiente, podríamos descomponer el problema global en K subproblemas del tipo:**

$$[MIN]\ SDQ_k = \sum_{i=1}^{I} (X_{i,k} - k \cdot r_i)^2 \tag{1-k}$$

s.a

$$\sum_{i=1}^{I} X_{i,k} = k \tag{2-k}$$

$$X_{i,k} \le u_i \tag{3}$$

$$X_{i,k} \ge 0 \text{ y entero} \tag{5}$$

Donde k es fijo y (3) y (5) están extendidos a $1 \le i \le I$. Puede interpretarse como la determinación de un vector I-dimensional $X(k) = [\ X_{1,k}\ X_{2,k}\ \ldots\ X_{I,k}\ ]'$ de componentes enteras no negativas cuya suma es k que esté "lo más cerca posible" de un vector $R(k) = [\ k.r_1\ k.r_2\ \ldots.\ k.r_I\ ]'$ de componentes (cuotas) no negativas, en general no enteras, cuya suma es k y que resultan de descomponer k proporcionalmente a determinados valores. Este problema se presenta en muchas circunstancias reales y en especial en procesos

políticos tales como el reparto del número total de escaños de una cámara de representantes entre diversas circunscripciones ("proporcionalmente a su población") o la atribución de escaños a los partidos políticos tras unas elecciones ("proporcionalmente a los votos obtenidos"). Uno de los métodos más antiguos para asignar escaños, pues se remonta por lo menos a 1791 cuando fue propuesto por Alexander Hamilton, es el LF (*method of Largest Fractions***) que precisamente minimiza SDQ_k =** $5\ X(k) - R(k)\ 5$**. Dicho método consiste en lo siguiente:**

- se asigna a cada opción i **las unidades que resultan de truncar al entero más próximo por defecto** $k.r_i$ **(es decir INT(**$k.r_i$**))**

- las unidades restantes hasta completar las k **que deben repartirse se asignan una a una en orden creciente de las fracciones** $f_i = k.r_i$ **- INT(**$k.r_i$**).**

Claramente en esta forma satisfacemos (2-k) y (5); puesto que en cualquier caso $*\ X_{i,k} - k.r_i\ *$ **es inferior a una unidad también se satisface (3):**

para $k < K : u_i = K.r_i > k.r_i$ **y** u_i **es entero** $X_{i,k} \le u_i$

para $k = K : K.r_i = u_i$ **es entero,** $X_{i,k} = u_i$

La optimización de (1-k) es fácil de demostrar ya que el valor SDQ resulta de la suma de cuadrados de valores inferiores a uno en valor absoluto, y entre todas las posibilidades hemos elegido aquélla en que dichos valores absolutos son globalmente menores.

Si después de resolver para cada k el problema relajado obtenemos unos valores $X_{i,k}$ **que satisfacen la relación suprimida (4) dicha solución es óptima para el problema original. En caso contrario solo dispondremos de una buena cota inferior de** SDQ**.**

Ejemplo S-1. Consideremos el caso $K = 20,\ I = 4,\ u_1 = 6,\ u_2 = 4,\ u_3 = 5,\ u_4 = 5$

$r_1 = 0,3\ \ r_2 = 0,2\ \ r_3 = 0,25\ \ r_4 = 0,25$

(llamaremos A, B, C y D a los tipos de productos)

Los cálculos se recogen en la figura 7.1.7.1. Los continuos empates entre los dos últimos productos los hemos resuelto siempre en orden alfabético aunque sería indiferente adoptar otra regla y no necesariamente la misma en cada uno de ellos. Como los sucesivos valores de $X_{i,k}$ **satisfacen la condición (4) la secuencia obtenida en la columna "incremento" es la que minimiza** SDQ **en este caso:**

A-C-D-B-A-C-D-B-A-C-D-A-B-C-D-A-B-C-D-A

SDQ **= 8,25 (ver figura 7.1.7.3)**

Desgraciadamente esto no ocurre siempre, como puede comprobarse en el ejemplo S-2.

k	*R (k)*				*X (k)*				**incremento (secuencia)**
	$k \cdot r_1$	$k \cdot r_2$	$k \cdot r_3$	$k \cdot r_4$	**A**	**B**	**C**	**D**	
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	
1	**0,3**	**0,2**	**0,25**	**0,25**	**1**	**0**	**0**	**0**	**A**
2	**0,6**	**0,4**	**0,50**	**0,50**	**1**	**0**	**1**	**0**	**C**
3	**0,9**	**0,6**	**0,75**	**0,75**	**1**	**0**	**1**	**1**	**D**
4	**1,2**	**0,8**	**1,00**	**1,00**	**1**	**1**	**1**	**1**	**B**
5	**1,5**	**1,0**	**1,25**	**1,25**	**2**	**1**	**1**	**1**	**A**
6	**1,8**	**1,2**	**1,50**	**1,50**	**2**	**1**	**2**	**1**	**C**
7	**2,1**	**1,4**	**1,75**	**1,75**	**2**	**1**	**2**	**2**	**D**
8	**2,4**	**1,6**	**2,00**	**2,00**	**2**	**2**	**2**	**2**	**B**
9	**2,7**	**1,8**	**2,25**	**2,25**	**3**	**2**	**2**	**2**	**A**
10	**3,0**	**2,0**	**2,50**	**2,50**	**3**	**2**	**3**	**2**	**C**
11	**3,3**	**2,2**	**2,75**	**2,75**	**3**	**2**	**3**	**3**	**D**
12	**3,6**	**2,4**	**3,00**	**3,00**	**4**	**2**	**3**	**3**	**A**
13	**3,9**	**2,6**	**3,25**	**3,25**	**4**	**3**	**3**	**3**	**B**
14	**4,2**	**2,8**	**3,50**	**3,50**	**4**	**3**	**4**	**3**	**C**
15	**4,5**	**3,0**	**3,75**	**3,75**	**4**	**3**	**4**	**4**	**D**
16	**4,8**	**3,2**	**4,00**	**4,00**	**5**	**3**	**4**	**4**	**A**
17	**5,1**	**3,4**	**4,25**	**4,25**	**5**	**4**	**4**	**4**	**B**
18	**5,4**	**3,6**	**4,50**	**4,50**	**5**	**4**	**5**	**4**	**C**
19	**5,7**	**3,8**	**4,75**	**4,75**	**5**	**4**	**5**	**5**	**D**
20	**6,0**	**4,0**	**5,00**	**5,00**	**6**	**4**	**5**	**5**	**A**

Fig. 7.1.7.1 Aplicación del método LF al ejemplo S-1. Los valores $X_{i,k}$ satisfacen la relación (4); por tanto, hemos obtenido una secuencia óptima

Ejemplo S-2. *$K = 13$, $I = 3$, $u_1 = 6$, $u_2 = 6$, $u_3 = 1$*

$r_1 = 0,4615$ $r_2 = 0,4615$ $r_3 = 0,0769$

(llamaremos A, B y C a los tipos de producto)

Los cálculos se han recogido en la figura 7.1.7.2. Dado el incumplimiento de la relación (4) hemos realizado, aparentemente, dos intentos fallidos de asignar el único ejemplar del producto C en *$k = 5$* **y** *$k = 7$***, aunque a todas luces en la segunda ocasión, dada la simetría**

existente, la asignación era totalmente justificada. En la terminología de los procesos electorales se dice que LF no es *house monotone***, lo que significa que al aumentar los escaños a repartir pueden disminuir los atribuidos a alguna opción. En cierta ocasión esta circunstancia afectó a los estados de Alabama, Colorado y Maine, por lo que se conoce con el nombre de "paradoja de Alabama". La paradoja de Alabama impide que el método LF proporcione la solución óptima del problema de secuenciación (con criterio** *SDQ***) en todos los casos, pero como se ha dicho proporciona una cota inferior del valor** *SDQ* **de la secuencia óptima (en el ejemplo S-2 dicha cota es la suma de la columna SDQ_k, es decir 4,3094). Para salir del "impasse" Miltenburg (1989) propone un análisis exhaustivo de todas las subsecuencias posibles, enumerándolas y evaluándolas entre la posición anterior y posterior al de aparición de la paradoja (por ejemplo, en la paradoja de la posición 6 habría que analizar las subsecuencias entre la posición 5 y la 7, que en este caso sólo son 6) lo cual puede resultar muy farragoso aunque conduzca a la solución óptima. Será preferible utilizar un procedimiento heurístico, y para ello presentaremos la traducción del problema de secuenciación a la búsqueda de un camino extremo en un grafo.**

k	*R (k)*			*X (k)*			incremento	SDQ_k
	$k \cdot r_1$	$k \cdot r_2$	$k \cdot r_3$	A	B	C		
0	0	0	0	0	0	0		0
1	0,462	0,462	0,077	1	0	0	A	0,5088
2	0,923	0,923	0,154	1	1	0	B	0,0356
3	1,385	1,385	0,231	2	1	0	A	0,5798
4	1,846	1,846	0,308	2	2	0	B	0,1423
5	2,308	2,308	0,385	2	2	1	C	0,5680
6	2,769	2,769	0,462	3	3	0	A + B-C	0,3202
7	3,231	3,231	0,538	3	3	1	C	0,3202
8	3,692	3,692	0,615	4	4	0	A + B-C	0,5680
9	4,154	4,154	0,692	4	4	1	C	0,1423
10	4,615	4,615	0,769	5	4	1	A	0,5798
11	5,077	5,077	0,846	5	5	1	B	0,0356
12	5,538	5,538	0,923	6	5	1	A	0,5088
13	6,000	6,000	1,000	6	6	1	B	0,0000

Fig. 7.1.7.2 Aplicación del método LF al ejemplo S-2. Los valores $X_{i,k}$ no satisfacen la condición (4) en las posiciones 6 y 8. No disponemos, por tanto, de ninguna secuencia, óptima o no

Denominaremos *grafo asociado al problema de secuenciación G* **a un grafo orientado conexo, sin bucles ni circuitos polietápicos, con** $K+1$ **niveles. Los vértices al nivel** k **(**$0 \le k \le K$**) quedan definidos por todos los vectores** l**-dimensionales de componentes enteros**

no negativos cuya suma es k**, y que cumplen la condición** $X_{i,k} \leq u_i\ (1 \leq i \leq I)$, $X(k) = [\ X_{1,k}\ X_{2,k} \ldots X_{I,k}\ ]' \in N^I$.

Al nivel 0 existe un único vértice $X(0)$ **= [0 0 .... 0]' lo mismo que al nivel** K**,** $X(K) = [u_1\ u_2 \ldots u_I]'$**. Existe un principio de simetría por el que el número de vértices al nivel k es el mismo que al nivel** K**-**k**.**

Los arcos del grafo unen vértices de niveles contiguos, del nivel k**-1 al nivel** k**. Existe un arco del vértice** $X^{(1)}(k-1)$ **al vértice** $X^{(2)}(k)$ **si:**

$$X^{(2)}(k) \geq X^{(1)}(k-1)$$

lo que implica que todos los componentes de $X^{(2)}(k)$ **son idénticos a los de** $X^{(1)}(k-1)$ **salvo uno superior en una unidad (que corresponde al tipo de producto situado en la posición** k **de la secuencia). A cada vértice al nivel** k **puede asociarse un valor (correspondiente a** SDQ_k**):**

$$v(X(k)) = \sum_{i=1}^{I} (X_{i,k} - k \cdot r_i)^2 = \| X(k) - R(k) \|$$

Hallar la secuencia que minimiza SDQ **es equivalente a hallar el camino mínimo de** $X(0)$ **a** $X(K)$ **en el grafo** G**. Esta concepción es adaptable a gran número de criterios distintos de** SDQ **(por ejemplo,** SDR **y** SDM**). Dado que el número de vértices de** G **puede ser muy elevado los procedimientos de determinación de caminos mínimos que consideren explícitamente todos los vértices del grafo (o incluso aquellos que consideren simultáneamente sólo todos los vértices de un nivel) pueden resultar impracticables. Sin embargo podemos diseñar gran número de heurísticas, de tipo constructivo, que concentren los esfuerzos de cálculo en unos pocos caminos del grafo (aun a riesgo de dejar de lado el óptimo). La más sencilla consiste en elegir a cada etapa,** k**, del algoritmo un arco (situa-do del nivel** k**-1 al nivel** k**), considerando el camino parcial formado por los** k**-1 arcos ya elegidos como inamovible: por consiguiente, el arco elegido debe emerger del último vértice del camino parcial, y entre todos los posibles se elige el que minimiza la aportación representada por la expresión (9) (que no está extendida a todos los vértices al nivel** k **sino sólo a los accesibles desde el último vértice del camino parcial a nivel** k**-1, que a lo sumo son** I**). Podemos formalizar más:**

Algoritmo_1.

Paso 1. *Inicializar***. Calcular** r_i**, hacer** $v_i = \mu_i$, $X_{i,0} = 0\ (1 \leq i \leq I)$**. Hacer** $k = 1$**.**

Paso 2. *Iteración*. **Sean** v_i **las unidades de** i **que quedan por secuenciar en la etapa** k**; si** $k = K-1$**,** *ir a Paso 4*; **en caso contrario elegir el producto** s **tal que:**

$$d_s(k) = \min_{v_i > 0} \{ \sum_{j=1}^{I} (X_{j,k-1} + \delta_{j,i} - k \cdot r_j)^2 \}$$

donde $\delta_{i,j}$ **= 1 si** $i = j$ **y en caso contrario** $\delta_{i,j}$ **= 0**

en la posición *k se secuencia una unidad del producto s*

Paso 3. *Actualización*. **Hacer** $X_{i,k} = X_{i,k-1} + \delta_{i,s}$ **para todo** i**; hacer** $v_s = v_s - 1$

Paso 4. *Bucle*. **Si** $k = K-1$**, secuenciar en la posición** K **la unidad del producto** s **tal que** $v_s = 1$ **(el resto de valores** v_i **serán iguales a 0),** *finalizar*; **en caso contrario hacer** $k = k + 1$**,** *ir a Paso 2*.

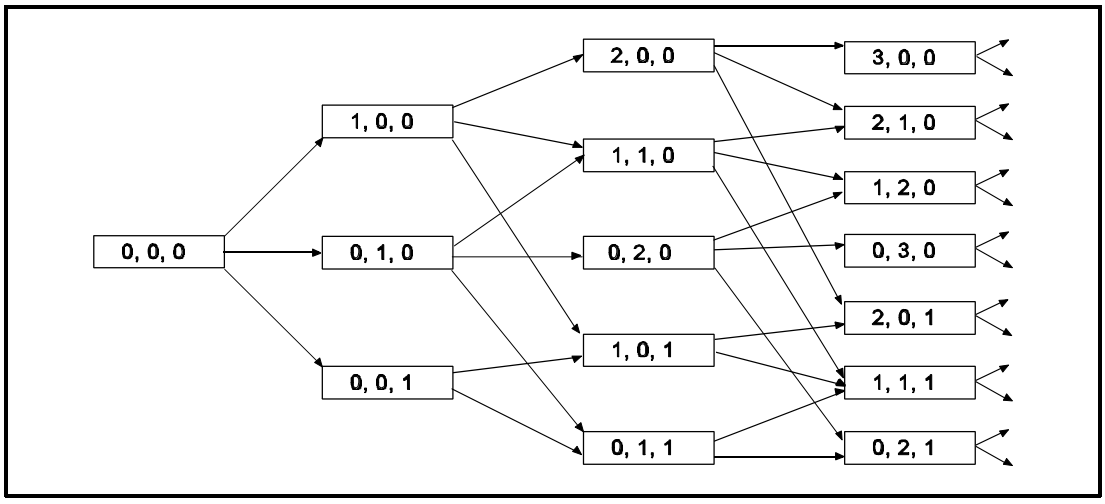


Fig. 7.1.7.3 Grafo asociado

k	$d_i(k)$				$s(k)$	$X(k)$				$d_s(k)$	$\sum_{h=1}^{k} SDQ_h$
	A	B	C	D		A	B	C	D		
0						0	0	0	0		0
1	0,655	0,855	0,755	0,755	A	1	0	0	0	0,655	0,655
2	2,620	1,020	0,820	0,820	C	1	0	1	0	0,820	1,475
3	2,195	0,795	2,495	2,495	D	1	0	1	1	0,495	1,970
4	1,280	0,080	1,680	1,680	B	1	1	1	1	0,080	2,050
5	0,375	1,375	0,875	0,875	A	2	1	1	1	0,375	2,425
6	1,980	1,180	0,580	0,580	C	2	1	2	1	0,580	3,005
7	1,595	0,995	2,295	2,295	D	2	1	2	2	0,295	3,300
8	0,720	0,320	1,520	1,520	B	2	2	2	2	0,320	3,620
9	0,255	2,055	1,155	1,155	A	3	2	2	2	0,255	3,875
10	1,500	1,500	0,500	0,500	C	3	2	3	2	0,500	4,375
11	1,155	1,355	2,255	0,255	D	3	2	3	3	0,255	4,630
12	0,320	0,720	1,520	1,520	A	4	2	3	3	0,320	4,950
13	1,695	0,295	0,995	0,995	B	4	3	3	3	0,295	5,245
14	1,180	1,980	0,580	0,580	C	4	3	4	3	0,580	5,825
15	0,875	1,875	2,375	2,375	D	4	3	4	4	0,375	6,200
16	0,080	1,280	1,680	1,680	A	5	3	4	4	0,080	6,280
17	1,095	0,495	0,795	0,795	B	5	4	4	4	0,495	6,775
18	1,020	-	0,820	0,820	C	5	4	5	4	0,820	7,595
19	0,755	-	-	0,655	D	5	4	5	5	0,655	8,250
20	0,000	-	-	-	A	6	4	5	5	0,000	8,250

Fig. 7.1.7.4 Aplicación del algoritmo 1 al ejemplo S-1. Obtenemos la misma solución que en la figura 7.1.7.1

k	$d_i(k)$			$s(k)$	$X(k)$			$d_s(k)$	$\sum_{h=1}^{k} SDQ_h$
	A	B	C		A	B	C		
0					0	0	0		0
1	0,5089	0,5089	1,2781	A	1	0	0	0,5089	0,5089
2	2,0355	0,0355	1,5740	B	1	1	0	0,0355	0,5444
3	0,5799	0,5799	0,8876	A	2	1	0	0,5799	1,1243
4	2,1421	0,1420	1,2190	B	2	2	0	0,1420	1,2663
5	0,7220	0,7220	0,5680	C	2	2	1	0,5680	1,8343
6	0,9351	0,9351	-	A	3	2	1	0,9351	2,7694
7	0,3197	0,3195	-	B	3	3	1	0,3195	3,0889
8	0,7220	0,7220	-	A	4	3	1	0,7220	3,8109
9	2,1421	0,1420	-	B	4	4	1	0,1420	3,9529
10	0,5799	0,5799	-	A	5	4	1	0,5799	4,5328
11	2,0355	0,0355	-	B	5	5	1	0,0355	4,5683
12	0,5089	0,5089	-	A	6	5	1	0,5089	5,0772
13	-	0,0000	-	B	6	6	1	0,0000	5,0722

Fig. 7.1.7.5 Aplicación del algoritmo 1 al ejemplo S-2. Obtenemos una secuencia (que no es óptima)

Este algoritmo tiene muchos puntos de contacto con el método de persecución de objetivos de Toyota que veremos más adelante. Su aplicación al ejemplo S-1 (figura 7.1.7.4) conduce a la misma solución óptima ya hallada. La aplicación al ejemplo S-2 (figura 7.1.7.5) proporciona una secuencia:

A-B-A-B-C-A-B-A-B-A-B-A-B

que no puede ser la óptima (por simetría la unidad de C debe estar, en la solución óptima, en la posición $k = 7$ y no en la $k = 5$). El valor de SDQ obtenido, 5,0722, es superior a la cota obtenida 4,3094 y también al valor de la secuencia óptima que como veremos es 4,6156.

El algoritmo_1 no conduce siempre a buenos resultados a causa de su marcada "miopía": toma las decisiones de secuenciar una unidad sin mirar las consecuencias en las asignaciones siguientes. Una forma de reducir dicha miopía consiste en ampliar el "campo de visión" en la toma de decisiones, analizando el efecto de las mismas más allá de la próxima posición (análogamente al jugador de ajedrez que considera una secuencia de varias jugadas antes de decidirse por la primera de ellas). La heurística 2-etapas se aplicaría de la forma siguiente:

Algoritmo_2.

Paso 1. *Inicializar*. **Calcular** r_i, **hacer** $v_i = u_i$, $X_{i,0} = 0$ $(1 \le i \le I)$. **Hacer** $k = 1$.

Paso 2. *Iteración*. **Sean** v_i **las unidades de** i **que quedan por secuenciar en la etapa** k; **si** $k = K\text{-}1$, **ir a Paso 4; en caso contrario elegir el producto** s **tal que:**

$$d_s(k) = \min_{v_i > 0} \{ \sum_{j=1}^{I} (X_{j,k-1} + \delta_{j,i} - k \cdot r_i)^2 +$$

$$+ \min_{v_h > \delta_{h,i}} \{ \sum_{j=1}^{I} (X_{j,k-1} + \delta_{j,i} + \delta_{h,j} - (k+1) \cdot r_j)^2 \} \}$$

en la posición k *se secuencia una unidad del producto* s

Paso 3. *Actualización*. **Hacer** $X_{i,k} = X_{i,k-1} + \delta_{i,s}$ **para todo** i, **hacer** $v_s = v_s\text{-}1$.

Paso 4. *Bucle*. **Si** $k = K\text{-}1$, **secuenciar en la posición** K **la unidad del producto** s **tal que** $v_s = 1$ **(el resto de valores** v_i **serán iguales a 0),** *finalizar*; **en caso contrario hacer** $k = k + 1$, *ir a Paso 2*.

La aplicación al ejemplo S-2 proporciona la secuencia:

A-B-A-B-A-B-C-A-B-A-B-A-B

que es óptima y a la que corresponde $SDQ = 4{,}6156$

7.1.7.1 Transformación de la distancia cuadrática

Dada la expresión (9) de v($X(k)$**):**

$$v(X(k)) = \sum_{i=1}^{I} (X_{i,k} - k \cdot r_i)^2 = \| X(k) - R(k) \|$$

podemos transformarla en una forma cuadrática. En efecto, teniendo en cuenta que la suma de las componentes de X(k) es k podemos escribir:

$$R(k) = k \cdot R = R \cdot O' \cdot X(k)$$

donde $O = [\,1\ 1\ 1\ \ldots\ 1\,]'$**; por tanto:**

$$v(X(k)) = 5\ (I - R \cdot O') \cdot X(k)\ 5 = X(k)' \cdot A \cdot X(k)$$

donde I **es la matriz unidad y** A**:**

$$A = (I - R \cdot O')' \cdot (I - R \cdot O')$$

es una matriz semidefinida positiva.

En el ejemplo S-1 tenemos:

$$I - R \cdot O = \begin{bmatrix} 0{,}7 & -0{,}3 & -0{,}3 & -0{,}3 \\ -0{,}2 & 0{,}8 & -0{,}2 & -0{,}2 \\ -0{,}25 & -0{,}25 & 0{,}75 & -0{,}25 \\ -0{,}25 & -0{,}25 & -0{,}25 & 0{,}75 \end{bmatrix}$$

$$A = \begin{bmatrix} 0{,}655 & -0{,}245 & -0{,}295 & -0{,}295 \\ -0{,}245 & 0{,}855 & -0{,}195 & -0{,}195 \\ -0{,}295 & -0{,}195 & 0{,}755 & -0{,}245 \\ -0{,}295 & -0{,}195 & -0{,}245 & 0{,}755 \end{bmatrix}$$

7.1.8 Fabricación ajustada a fechas contractuales

Inman & Bulfin aceptan y dan por justificada la utilidad de las funciones z_2 y z_4 de Miltenburg; sin embargo, su enfoque da lugar a una interpretación del problema conceptualmente distinta. Asignan a cada unidad una fecha contractual de lanzamiento (o entrada) a la cadena y pretenden reducir al mínimo los adelantos y los retrasos (huelgos) entre fechas reales y fechas ideales calculadas (como en un problema n/1//L_{max}). La formulación es:

$$[MIN]\, z_7 = \sum_{i=1}^{I} \sum_{h=1}^{u_i} (s_{i,h} - t_{i,h})^2$$

$$[MIN]\, z_8 = \sum_{i=1}^{I} \sum_{h=1}^{u_i} |s_{i,h} - t_{i,h}|$$

$$[MIN]\, z_9 = \underset{i,h}{MAX} |s_{i,h} - t_{i,h}|$$

donde:

$s_{i,h}$ **es el instante en que se secuencia realmente la** h**-ésima unidad del tipo de producto** i**,** $t_{i,h}$ **es el instante idóneo para lanzar la** h**-ésima unidad del tipo** i **en la línea.**

El instante idóneo $t_{i,h}$ **(o fecha contractual) debe establecerse con la idea de que las unidades pertenecientes a un mismo tipo de artículo se introduzcan en la línea a intervalos regulares de tiempo o, visto de otra forma, que ocupen en la secuencia posiciones equidistantes. Aunque no es la única posibilidad, parece adecuado definir dicho instante mediante el índice de Webster:**

$$t_{i,h} = (h - 0{,}5) \cdot \frac{K}{u_i}$$

en el que hemos supuesto implícitamente que el intervalo entre unidades sucesivas de la secuencia es la unidad.

En el caso *SDQ* **(función** z_7**) la regla de secuenciación es muy simple:**

$$SDQ = \sum_{i=1}^{I} \sum_{h=1}^{u_i} (s_{i,h} - t_{i,h})^2 =$$

$$\sum_{i=1}^{I} \sum_{h=1}^{u_i} s_{i,h}^2 + \sum_{i=1}^{I} \sum_{h=1}^{u_i} t_{i,h}^2 - 2 \cdot \sum_{i=1}^{I} \sum_{h=1}^{u_i} s_{i,h} \cdot t_{i,h}$$

Los dos primeros términos del desarrollo son constantes, independientes de la secuencia (dado que en cada posición habrá una unidad y sólo una). Por tanto, minimizar *SDQ* **equivale a maximizar el tercer término; para ello basta ordenar (i,h) en orden creciente de** $t_{i,h}$ **y por tanto asignaremos a los** $s_{i,h}$ **así ordenados valores también crecientes, con lo que la suma de los productos binarios alcanzará su máximo. Se trata del orden** *EDD***, que por tanto minimiza el huelgo máximo y el retraso máximo también. Si aplicamos el procedimiento al ejemplo S-1 (figura 7.1.8.1) obtenemos la misma secuencia que con el método LF.**

$t_{i,h}$				
h/i	**A**	**B**	**C**	**D**
1	**1,67**	**2,5**	**2**	**2**
2	**5**	**7,5**	**6**	**6**
3	**8,33**	**12,5**	**10**	**10**
4	**11,67**	**17,5**	**14**	**14**
5	**15**	**-**	**18**	**18**
6	**18,33**	**-**	**-**	**-**

Fig. 7.1.8.1 Determinación de los valores $t_{i,h}$ *para el ejemplo S-1. En orden creciente determinan la secuencia A-C-D-B-A-C-D-B-A-C-D-A-B-C-D-A-B-C-D-A*

7.1.9 Equilibrado de las tasas de las necesidades

Vamos a concentrarnos en la regularidad de las tasas de consumo de componentes. La nomenclatura utilizada es la siguiente:

I **número de tipos de productos diferentes a secuenciar;**
i **producto genérico (**$i = 1, 2, \ldots, I$**);**
u_i **número de unidades del producto** i **a secuenciar;**
$K = \sum u_i$ **total de unidades a secuenciar;**
J **número de componentes (distintos) a tener en cuenta;**
j **componente genérico (**$j = 1, 2, \ldots, J$**);**
$n_{j,i}$ **número de unidades del componente** j **que se incorporan a una unidad del producto** i**; esta cantidad se deduce a partir de la lista de materiales.**

El orden deseado de las unidades en la secuencia es aquél que garantice un flujo o consumo lo más regular posible de los componentes, lo que facilitará la reducción de stocks. Llamando T_j **al número de componentes** j **necesario para cumplir el programa de**

montaje:

$$T_j = \sum_{i=1}^{I} n_{j,i} \cdot u_i$$

y el consumo medio por unidad montada será:

$$\mathbf{r_j = T_j/K}$$

En la medida en que las diferencias (r_j - $n_{j,i}$) sean importantes, más difícil será lograr una secuencia de productos que se aproxime a un consumo regular de componentes. Sea una secuencia *S* **cualquiera y** *k* **una posición cualquiera de la misma. Según el número de unidades de cada tipo que hayamos situado en las** *k* **primeras posiciones de la secuencia, la cantidad de componentes** *j* **consumida hasta dicha posición será diferente. Llamaremos a dicha cantidad:**

$$Y_{j,k}(S) = \sum n_{i,j} \cdot X_{i,k}$$

siendo como antes $X_{i,k}$ el número de unidades del tipo *i* **secuenciadas en las** *k* **primeras posiciones de** *S***. El consumo regular sería:**

$$k.r_j$$

Deseamos que la secuencia elegida conduzca a unos valores reales de consumo lo más parecidos posible a los regulares, y esto para todo componente *j* **y toda posición** *k***. En otras palabras, deseamos que los vectores** *J***-dimensionales $Y(k) = [\ Y_{1,k}\ Y_{2,k}\ \ldots\ Y_{J,k}\ ]'$ estén lo más cerca posible de los $R(k) = [\ k.r_1\ k.r_2\ \ldots.\ k.r_J\ ]'$ = $k \cdot R$ con $R = [r_1\ r_2\ \ldots.\ r_J]'$. Este objetivo, que parece intuitivamente bastante claro, no lo es tanto en el momento en que queremos formularlo mediante una expresión matemática concreta. La adaptación de las formulaciones establecidas en 6.1.6 conducen a:**

$$[MIN]\, z_{10} = \sum_{k=1}^{K} \sum_{j=1}^{J} (Y_{j,k} - k \cdot r_j)^2$$

$$[MIN]\, z_{11} = \sum_{k=1}^{K} \sum_{j=1}^{J} \left(\frac{Y_{j,k}}{k} - r_j\right)^2$$

$$[MIN]\, z_{12} = \sum_{k=1}^{K} \sum_{j=1}^{J} |Y_{j,k} - k \cdot r_j|$$

$$[MIN]\, z_{13} = \sum_{k=1}^{K} \sum_{j=1}^{J} \left|\frac{Y_{j,k}}{k} - r_j\right|$$

$$[MIN]\, z_{14} = \sum_{k=1}^{K} \max_{j} |Y_{j,k} - k \cdot r_j|$$

$$[MIN]\, z_{15} = \sum_{k=1}^{K} \max_{j} \left|\frac{Y_{j,k}}{k} - r_j\right|$$

Aunque Monden propone, como criterio utilizado en Toyota:

$$[MIN]\, z_{16} = \sum_{k=1}^{K} \left(\sum_{j=1}^{J} (Y_{j,k} - k \cdot r_j)^2 \right)^{1/2}$$

El programa cuadrático entero equivalente a z_{10} sería:

$$[MIN]\, SDQ = \sum_{k=1}^{K} \sum_{j=1}^{J} \left(\sum_{i=1}^{I} n_{j,i} \cdot X_{i,k} - k \cdot r_j \right)^2 \qquad \textbf{(1)}$$

s.a

$$\sum_{i=1}^{I} X_{i,k} = k \qquad 1 \le k \le K \qquad \textbf{(2)}$$

$$X_{i,k} \le u_i \qquad \textbf{(3)}$$

$$0 \le X_{i,k} - X_{i,k-1} \le 1 \qquad \textbf{(4)}$$

$$Xi,k \ge 0 \text{ y entero} \qquad \textbf{(5)}$$

con (3), (4) y (5) extendidos a $1 \le k \le K$ y $1 \le i \le I$ (adoptando para todo i $X_{i,0} = 0$). Es decir, las restricciones coinciden con las indicadas en 7.1.7 variando ligeramente la función económica. Por tanto, podemos pues asociar el problema al mismo grafo G que hemos definido en 7.1.7 *modificando* **el valor asociado a los vértices que será ahora:**

$$v(X(k)) = \sum_{j=1}^{J} \left(\sum_{i=1}^{I} n_{j,i} \cdot X_{i,k} - k \cdot r_j \right)^2 = \| N \cdot X(k) - R(k) \|$$

Donde N **es la matriz** (J,I) **cuyos componentes son** $n_{j,i}$**. Recordando la definición de** $R(k)$ **podemos escribir:**

$$R(k) = k \cdot R = R \cdot (O' \cdot X(k))$$

donde O **= [1 1 1 ... 1]', ya que la suma de los componentes de** $X(k)$ **es igual a k. Por tanto:**

$$v(X(k)) = 5\ (N - R \cdot O') \cdot X(k)\ 5 = X(k)' \cdot A \cdot X(k)$$

donde
$$A = (N - R \cdot O')' \cdot (N - R \cdot O')$$

Por tanto, son aplicables los mismos procedimientos descritos en 7.1.7 (salvo LF) con la adaptación de la función objetivo.

Por consiguiente son posibles algoritmos constructivos "miopes" que en sucesivas etapas vayan añadiendo arcos al camino del grafo hasta unir los vértices extremos; cada arco añadido será aquél de entre los posibles que incremente lo mínimo la distancia elegida como función objetivo. Dichos algoritmos, en el caso que nos ocupa, y supuestas determinadas, por ejemplo, las cinco primeras posiciones de la secuencia, elegirían el producto a situar en sexta posición, y una vez elegido pasarían a la séptima, etc. Todos ellos son del mismo tipo que el algoritmo de "persecución de objetivos" (*goal chasing method***) presentado por Y. Monden como procedimiento empleado en Toyota.**

Ejemplo S-3.

- número de productos diferentes: I = 4 (A,B,C,D)
- número de componentes: J = 5 (1,2,3,4,5)
- estructura del producto $n_{j,i}$:

Componente	**1**	**2**	**3**	**4**	**5**	**Cantidad**
Producto						u_i
A	**3**	**3**	**3**	**2**	**1**	**6**
B	**4**	**1**	**2**	**3**	**2**	**4**
C	**2**	**2**	**1**	**3**	**4**	**5**
D	**2**	**2**	**2**	**3**	**3**	**5**
T_j	**54**	**42**	**41**	**54**	**49**	**K = 20**
r_j	**2,7**	**2,1**	**2,05**	**2,7**	**2,45**	

y sea la secuencia parcial S_p: D - B - D - A - C -

En la figura 7.1.9.1 hemos calculado, para cada valor de k entre 1 y 5, las distancias cuadrática, rectangular y minimax de la secuencia parcial.

		j					$\sum_{h=1}^{k} SDQ_h$	$\sum_{h=1}^{k} SDR_h$	$\sum_{h=1}^{k} SDM_h$
		1	2	3	4	5			
k		$Y_{j,k}$ (S_p)							
1	D	2	2	2	3	3	0,875	1,7	0,7
2	B	6	3	4	6	5	3,075	4,3	1,9
3	D	8	5	6	9	8	6,03	7,4	3,2
4	A	11	8	9	11	9	7,55	9,8	4,0
5	C	13	10	10	14	13	8,925	12,3	4,75

Fig. 7.1.9.1 Valores de las distancias cuadrática, rectangular y minimax para una secuencia dada

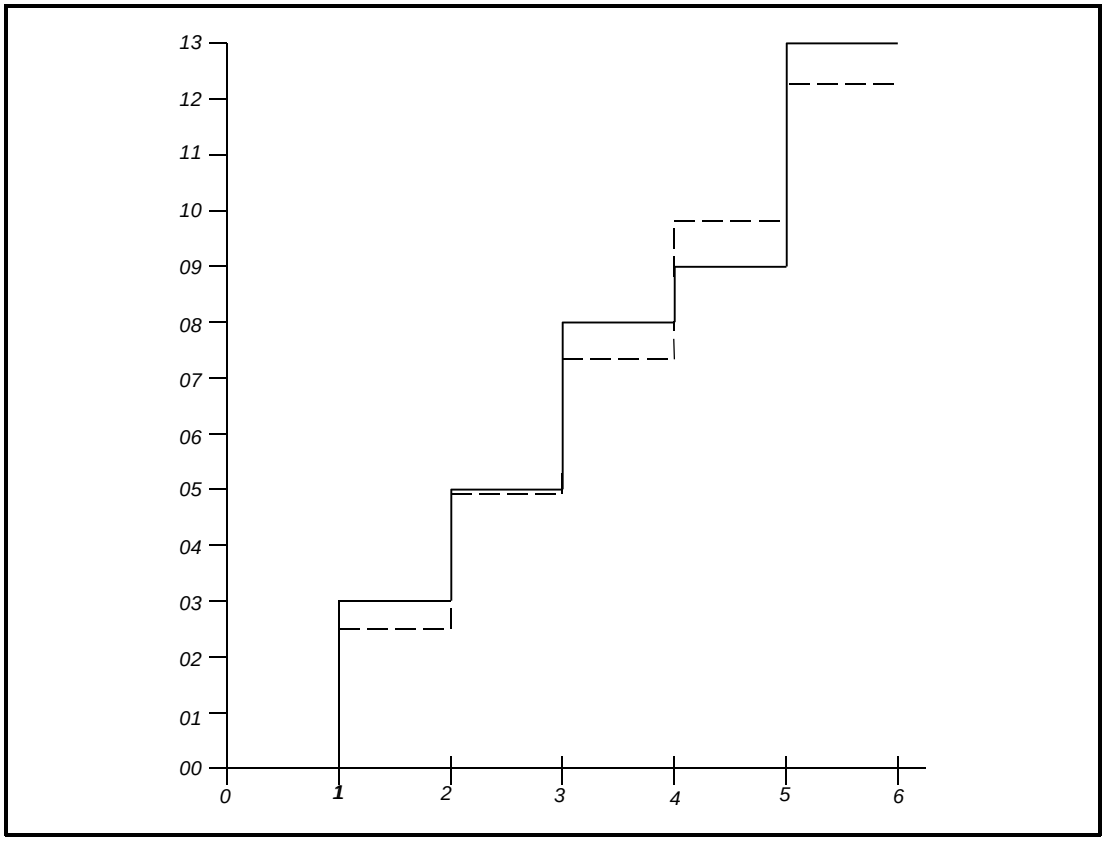


Fig. 7.1.9.2 Comparación entre el consumo teórico y real para el componente 5 ())) : teorico;)))))): real)

Deseamos la secuencia que tenga un perfil de consumo de componentes lo más cercano al consumo regular a lo largo de todos los valores de *k*.

Los algoritmos constructivos sitúan en cada posición *k* (1, 2, 3, ..., *K*) aquel producto (no agotado) cuya aportación es mínima, es decir, aquel que minimiza una de las expresiones:

Secuencia-Q

$$SDQ_k = [\underset{i}{MIN}] \sum_{j=1}^{J} (Y_{j,k}(S_p \; i) - k \cdot r_j)^2$$

Secuencia-R

$$SDR_k = [\underset{i}{MIN}] \sum_{j=1}^{J} |Y_{j,k}(S_p \; i) - k \cdot r_j|$$

Secuencia-M

$$SDM_k = [MIN]_i \ \max_j \ | Y_{j,k}(S_p \ \ i) - k \cdot r_j \ |$$

Aquí S_p representa la secuencia ya hallada para las *k-1* primeras posiciones, y $S_p^* i$ a la prolongación de dicha secuencia con una unidad del tipo *i*. El primer criterio tiene como base la distancia cuadrática o euclídea, el segundo la distancia rectangular o Manhattan y el tercero la distancia minimax.

Consideremos las cuatro posibilidades para la 6ª posición, que nos llevarían a un valor de $Y_{j,6}(S)$ y a un término a añadir a *SDQ*, *SDR* y *SDM* de acuerdo con la tabla de la figura 7.1.9.3. Los tres criterios conducirían a situar el producto A en dicha posición.

6a posición	$Y_{i,6}$ (S_p*i)					Incremento de la distancia		
j	**1**	**2**	**3**	**4**	**5**	SDQ_k	SDR_k	SDM_k
i								
A	**16**	**13**	**13**	**16**	**14**	**1,22**	**2,2**	**0,7**
B	**17**	**11**	**12**	**17**	**15**	**4,02**	**3,8**	**1,6**
C	**15**	**12**	**11**	**17**	**17**	**9,42**	**6,2**	**2,3**
D	**15**	**12**	**12**	**17**	**16**	**4,22**	**4,2**	**1,3**

Fig. 7.1.9.3 Aportaciones de las cuatro posibles continuaciones de la secuencia S_p

Generalmente, en lo que sigue, cuantificamos las secuencias de acuerdo con las tres distancias mencionadas así como respecto a la aportación máxima de las mismas para *k = 1,2….,K.*

Un programa construido para un PC nos determina fácilmente las secuencias completas del problema propuesto empleando dichas heurísticas (figura 7.1.9.4) a las que hemos añadido las halladas por otros procedimientos.

Las tres primeras secuencias corresponden a las heurísticas indicadas en el texto (derivadas del procedimiento "persecución de objetivos"), minimización de SDQ_k, SDR_k y SDM_k. El carácter "miope" de estos procedimientos respecto al objetivo global se advierte al comprobar las mejores prestaciones de la "secuencia_R" y la "secuencia_M" respecto a *SDQ* que la "secuencia_Q", a pesar de que el criterio empleado para construir esta última se basa precisamente en la distancia cuadrática, y en las otras dos no.

El resto de secuencias se han construido tomando como objetivo la minimización de *SDQ*. La heurística denominada "Monden revisado" utiliza una adaptación de la expresión SDQ_k; esta adaptación es posible sólo en distancias cuadráticas. La heurística "simétrica" corresponde a emplear el criterio de minimización de SDQ_k alternativamente al principio y final de la secuencia. La heurística "2-pasos" corresponde a analizar, de manera similar a lo realizado bajo el mismo nombre en 7.1.7.6, las aportaciones correspondientes a la

distancia cuadrática para las dos posiciones siguientes de la secuencia, decidiendo a partir de ahí el producto a situar en la primera de ellas. Tanto la heurística simétrica como la 2-pasos son adaptables a las distancias rectangular o minimax.

Secuencia_R (h. Monden) **D-B-D-A-C-A-D-B-D-A-C-B-D-A-C-A-C-B-A-C**		
max SDQ_k **= 4,095** SDQ **= 41,25**	**max** SDR_k **= 4,000** SDR **= 51,30**	**max** SDM_k **= 1,550** SDM **= 19,45**
Secuencia_R (h. Monden) **D-B-D-A-C-A-D-B-D-A-C-B-D-A-C-A-C-B-A-C**		
max SDQ_k **= 4,500** SDQ **= 39,05**	**max** SDR_k **= 3,700** SDR **= 48,40**	**max** SDM_k **= 1,550** SDM **= 19,00**
Secuencia_M (h. Monden) **D-A-C-B-D-A-C-B-D-A-D-B-C-A-D-A-C-B-C-A**		
max SDQ_k **= 4,395** SDQ **= 32,95**	**max** SDR_k **= 4,300** SDR **= 46,30**	**max** SDM_k **= 1,450** SDM **= 16,15**
Secuencia_Q (h. Monden revisado) **D-A-C-B-A-C-B-D-A-C-A-C-B-A-C-D-A-B-D-D**		
max SDQ_K **= 4,220** SDQ **= 34,95**	**max** SDR_k **= 4,200** SDR **= 47,90**	**max** SDM_k **= 1,400** SDM **= 17,10**
Secuencia_Q (h. simétrica) **D-B-D-A-C-A-D-B-C-A-C-A-B-C-A-C-A-D-B-D**		
max SDQ_k **= 2,955** SDQ **= 31,35**	**max** SDR_k **= 3,300** SDR **= 45,80**	**max** SDM_k **= 1,300** SDM **=16,80**
Secuencia_Q (h. simétrico/revisada) **D-A-C-B-A-C-B-D-A-C-D-A-D-B-C-A-B-C-A-D**		
max SDQ_k **= 2,875** SDQ **= 29,25**	**max** SDR_k **= 3,500** SDR **= 48,80**	**max** SDM_k **= 1,250** SDM **= 15,90**
Secuencia_Q (h. 2-pasos) **D-A-C-B-D-A-C-B-A-D-C-A-B-D-C-A-D-B-A-C**		
max SDQ_k **= 4,095** SDQ **= 28,85**	**max** SDR_k **= 3,700** SDR **= 43,30**	**max** SDM_k **= 1,550** SDM **= 15,85**
Secuencia_Q (BDP) **D-A-C-B-D-A-C-B-A-D-C-A-B-C-A-D-B-C-A-D**		
max SDQ_k **= 2,580** SDQ **= 27,65**	**max** SDR_k **= 3,400** SDR **= 43,60**	**max** SDM_k **= 1,100** SDM **= 15,10**

Fig. 7.1.9.4 Diferentes secuencias evaluadas

La última secuencia se ha obtenido aplicando un procedimiento original, desarrollado por el Laboratori d'Organització de la Producció del DOE de la ETSEIB-UPC dentro de un Proyecto de Investigación subvencionado por la DGICYT (PB 89-0504), denominado programación dinámica acotada (*bounded dynamic programming* **o BDP), que en este caso proporciona una secuencia óptima con el valor mínimo de SDQ para el problema considerado. Puede comprobarse el alejamiento respecto a dicho óptimo del algoritmo de "persecución de objetivos". El procedimiento BDP puede utilizarse con distancias rectangulares y minimax.**

7.2 Bibliografía

[01] BAUTISTA, J.; *Procedimientos heurísticos y exactos para la secuenciación en sistemas productivos de unidades homogéneas (contexto JIT)***, Tesis Doctoral, DOE, ETSEIB-UPC, 1993**

[02] BAUTISTA, J.; COMPANYS, R.; COROMINAS, A.; *Seqüenciació d'unitats en context JIT***; Edicions UPC, 1995**

[03] BEDWORTH, D. D.; BAILEY, J. E.; *Integrated Production Control Systems***, Wiley, 1982**

[04] BUFFA, E. S.; SARIN, R. K.; *Administración de la Producción y de las Operaciones***; Limusa, 1992**

[05] COMPANYS, R.; FONOLLOSA, J. B.; *Nuevas técnicas de gestión de stocks: MRP y JIT***; Marcombo, 1989**

[06] COVES, A. Mª; *Equilibrado de líneas de produción y montaje, DOE-UPC***, DIT 94/17, 1994**

[07] MIZE, J. H.; WHITE, C. R.; BROOKS, G. H.; *Planificación y Control de Operaciones***; Prentice Hall internacional, 1973.**

Comentarios

[07] y [04] constituyen una buena introducción al equilibrado de líneas. [06] describe los modelos analíticos, ampliando los conceptos vistos en este capítulo.

El problema de la secuenciación de unidades está tratado fundamentalmente en artículos. Una presentación elemental aparece en [05], siendo [01] y [02] un tratamiento más completo.

7.3 Enunciados

Enunciado 7.3.1

Equilibrar una línea de montaje a partir de los siguientes datos:

Tarea	Duración	Tareas previas
a	5	-
b	4	-
c	6	-
d	3	a
e	4	b
f	3	c
g	5	a,b
h	5	d
i	5	d
j	6	e
k	4	f
l	2	g
m	3	h,i
n	5	j
o	4	k
p	4	l
q	5	m
r	6	n
s	4	o
t	2	p,q

Utilizar los ciclos: 12, 11 y 17.

Enunciado 7.3.2

Preparar un programa para PC que permita el equilibrado de líneas de montaje con ligaduras compuestas únicamente por precedencias. Constará de los siguientes módulos:

1) Entrada de datos por teclado. La entrada debe ser cómoda para el usuario, permitiéndole corregir los errores de pulsación.

2) Escritura de los datos en un diskette para utilización ulterior.

3) Recuperación de los datos a partir de un diskette.

4) Consulta y modificación de los datos.

5) Cálculo de los pesos.

6) Equilibrado mediante la heurística de los pesos.

7) Opción a buscar una mejora por simulación.

8) Impresión de los datos y resultados.

Enunciado 7.3.3

Construir un programa para PC que busque el equilibrado de líneas de montaje con ligaduras generales; además de las precedencias pueden existir otros tipos de ligaduras: disyunciones, grupos de tareas que deben estar en el mismo puesto, tareas cuyo puesto deba encontrarse entre ciertos valores, etc. El algoritmo a emplear será básicamente la simulación.

El programa estará compuesto por los siguientes módulos:

1) Entrada de datos por teclado.

2) Escritura de los datos en un diskette.

3) Recuperación de los datos del diskette.

4) Consulta y modificación de los datos.

5) Simulación.

Enunciado 7.3.4

Un tema tradicional que ha readquirido recientemente interés es el de un grupo de máquinas en serie, con capacidad de stockaje intermedio limitado. Consideremos dos máquinas solamente A y B, y la capacidad intermedia c **unidades. No hay limitaciones antes de A o después de B (puede suponerse que delante de A hay una cola ilimitada, por lo que A podrá alimentarse siempre que quede libre). En el momento en que en el stock intermedio no haya ninguna unidad y B termine su operación, quedará parada por falta de alimentación. En el momento en que esté lleno el stock intermedio y A termine una operación, ésta quedará parada pues no podrá descargar la pieza. Por consiguiente si no existe una** *sincronización perfecta* **entre ambas máquinas la tasa de producción del**

conjunto puede ser inferior a la de una máquina aislada. Las causas que pueden producir la falta de sincronización son entre otras:

a) tiempos de operación aleatorios, con posibles variaciones respecto a la media,

b) averías aleatorias en las máquinas.

En el primer caso, *a***, existe una expresión teórica de la tasa resultante, si los tiempos operatorios se distribuyen según leyes exponenciales idénticas, que de acuerdo a unas hipótesis de costes puede conducirnos a determinar el tamaño óptimo de la capacidad de almacenaje intermedio. Este resultado Young (1967) lo extendió a más de dos máquinas idénticas, con tiempos distribuidos exponencialmente, con capacidades intermedias idénticas (aunque en el mejor de los casos su extensión es una aproximación). Asimismo dió una expresión analítica (aproximada) para el caso en que la distribución de los tiempos fuese normal. Para mayores detalles ver** *Mize, White y Brooks. Planificación y Control de Operaciones***, pp. 186-189.**

El trabajo consiste en preparar un informe sobre este tema. Para ello se deberá construir un modelo de simulación para experimentar el comportamiento del sistema al ir variando el número de máquinas y la capacidad de almacenaje (así como las leyes de distribución de los tiempos operatorios). Con los resultados obtenidos se juzgará la bondad de las expresiones analíticas y, en su caso se construirá las tablas y/o ábacos oportunos.

Enunciado 7.3.5 Efecto bol

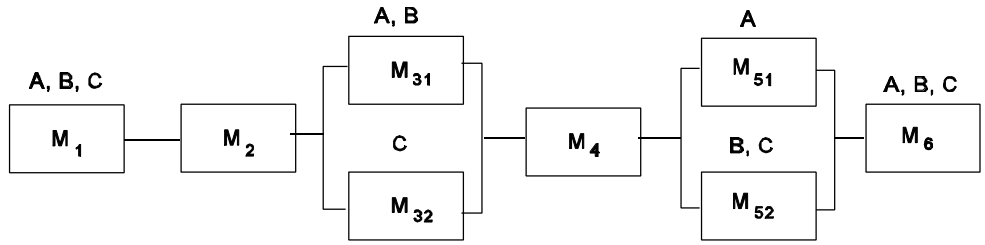
Una cadena de producción consta de tres estaciones de trabajo sin espacio de almacenamiento intermedio. Siempre hay material disponible para la primera estación, y los tiempos de operaciones pueden repartirse de cualquier forma entre las estaciones (el tiempo medio total debe ser igual a la suma de las medias de los tiempos correspondientes a cada estación y se considerará una distribución exponencial del tiempo en cada estación).

Se trata de establecer expresiones para calcular la producción media de la cadena por unidad de tiempo, en régimen permanente, y determinar cuál es la mejor distribución del tiempo entre las estaciones.

Enunciado 7.3.6 Análisis de reglas para el problema m-máquinas

Propósito

Un sistema productivo (muy sencillo) elabora tres piezas A, B y C según la estructura de la figura siguiente:



Cada máquina tiene un tiempo de preparación y una tasa de producción ligadas al tipo de pieza a producir, de la forma siguiente (tiempos de preparación en minutos, tasas en número de piezas por hora):

	M_1	M_2	M_{31}	M_{32}	M_4	M_{51}	M_{52}	M_6
A prep. tasa	100 60	120 70	150 60	-	130 70	140 30	-	110 50
B prep. tasa	110 50	130 60	140 70	-	120 60	-	120 50	130 40
C prep. tasa	120 70	110 50	-	120 20	110 50	-	130 60	120 60

Los lotes de trasferencia, cantidad de piezas producidas en una máquina antes de pasarlas a la elaboración de la máquina siguiente, y el rendimiento unitario de cada pieza son:

A	**100 piezas**	**3 Ptas/unidad**
B	**100 piezas**	**5 Ptas/unidad**
C	**100 piezas**	**4 Ptas/unidad**

Suponiendo una semana de 80 horas (dos turnos diarios de 8 horas cada uno), interesa saber la secuencia de lotes de piezas A, B y C a lanzar que conduzca al máximo rendimiento.

Formato

- el sistema preguntará la secuencia de lotes,

- utilizando el esquema anterior y unas reglas FIFO (sofisticadas en las elaboraciones en paralelo) calculará el número de piezas elaboradas de cada tipo sin esperas innecesarias,

- a partir de las piezas elaboradas el programa indicará el rendimiento obtenido durante la semana y, para cada máquina, la productividad,

- deseablemente, algún esquema gráfico mostrará la realización de los lotes antes de la presentación de resultados,

- los tiempos y tasas podrán cambiarse si se conoce los códigos y claves adecuadas.

Realización

- el programa se realizará en alguna variante de BASIC (por ejemplo QUICK-BASIC) y se entregará la fuente, el compilado y el manual del usuario,

- se emplearán las posibilidades de la pantalla en color, aunque el programa será utilizable en bicolor,

- podrá prolongarse la situación varias semanas, con indicación de la prolongación de la secuencia (si es preciso) y partiendo de la situación inicial de las máquinas correspondiente al final de la semana anterior (la secuencia indicada la semana anterior será obligada para los lotes ya en elaboración en la primera máquina, pero el resto de secuencia es revisable),

- se desea una estructura que favorezca los aspectos lúdicos y competitivos.

Enunciado 7.3.7 El problema BIMONDEN

7.3.7.1 Propósito

Probar por simulación diferentes reglas de asignación de productos a las diversas vías de un pulmón, previo a la línea de montaje. Una empresa fabrica un cierto número n de productos i, a los que asignaremos un peso p_i. Si N_i es el número de productos a fabricar un día dado, la tasa media $r=(\sum N_i \cdot p_i)/(\sum N_i)$ sirve para equilibrar la línea de montaje. Se intenta lanzar a la línea de montaje una secuencia de productos que se ajuste lo más posible a dicha tasa media, pero sólo son lanzables los que están en cabecera de las vías del pulmón.

El pulmón tiene m vías (m inferior a n) y cada una de ellas tiene capacidad para l unidades. Dichas unidades llegan al pulmón a instantes no regulares y aleatoriamente (dentro de las tasas establecidas) y salen regularmente.

Una regla debe decidir cuál de los productos situados en cabecera debe ser el que va a la línea de montaje, y otra, de acuerdo a la composición existente en el pulmón a cuál de las vías (con capacidad) debe incorporarse un producto que llega.

7.3.7.2 Formato

Dimensionado el pulmón y definidos N_i, p_i y la cadencia básica se generan llegadas. Cuando el pulmón está al 50% se comenzarán las salidas.

Las reglas pueden basarse en el conocido esquema de Monden.

7.3.7.3 Realización

- el programa se realizará en alguna variante de BASIC (por ejemplo QUICK-BASIC) y se entregará la fuente, el compilado y el manual del usuario,

- se emplearán las posibilidades de la pantalla en color, aunque el programa será utilizable en bicolor,

- los parámetros de base podrán cambiarse.