LA PROGRAMACIÓN DE LA PRODUCCIÓN EN EL SECTOR DEL AUTOMÓVIL. EL PROBLEMA DE LA SEGMENTACIÓN

Fco. Cruz Lario Esteban [1]
Cristóbal Miralles Insa [2]
José Pedro García Sabater [3]

Centro de Investigación de Gestión e Ingeniería de Producción (CIGIP)
DOE-ETSII-UPV. Camí de Vera, s/n 46022 Valencia

[1] fclario@omp.upv.es [2] crimiin@omp.upv.es [3] jpgarcia@omp.upv.es

RESUMEN

La Gestión de la Cadena de Suministro en el sector del automóvil, así como las relaciones entre ensamblador y proveedores de primera línea han adquirido una gran importancia en los últimos años. Se constata la importancia que los procesos de negocio de segmentación y secuenciación tienen en ese entorno, que resulta ser clave para aquellos proveedores secuenciados y sincronizados con el ensamblador.
*Se proponen una serie de Procedimientos de Resolución** *que permitan realizar la Programación-Secuenciación de la Producción, en el ámbito de un Sistema de Gestión de Producción de una empresa de fabricación de Unidades Homogéneas, utilizándose datos reales de una empresa concreta para su validación.*
Estos procedimientos están basados principalmente en la aplicación inicial de algunas heurísticas conocidas para posteriormente realizar un cierto número de particiones a la secuencia obtenida, aplicando a estas particiones otras heurísticas determinadas.

Palabras Clave:
Just In Time, Programación, Secuenciación, Regularidad, Segmentación

1 INTRODUCCIÓN

La evolución de los Mercados, particularmente en el Sector del Automóvil, exige una continua reducción de costes, al tiempo que se debe ofertar una gran variedad de productos con un tiempo de entrega más reducido y ajustado. Asimismo el automóvil es un producto cada vez más complejo en su diseño; los submontajes que conforman el producto final, son interdependientes entre sí, dificultando las tareas de montaje final, así como incrementando la variedad de subconjuntos que conforman el vehículo.

A medida que los subconjuntos adquieren mayor volumen, por el hecho de ser submontajes, aumentan sus costes de manipulación. De este modo, la evolución natural en Lean Production lleva a tener los Proveedores cerca de la planta de tal modo que sirvan estrictamente lo necesario y no sólo en pequeños lotes; es decir, aparecen como una necesidad los Proveedores que suministran los submontajes en Secuencia y sincronizados con las unidades que se fabrican en la línea. Así, estos Proveedores se convierten en pequeñas ramificaciones de la línea de montaje, creándose una estrecha vinculación de éstos con el Programa Secuencial. A este tipo de aprovisionamiento Monden lo denomina en [1] *Just-On-Time*.

1.1 El Programa Secuencial

* Este artículo se deriva del trabajo realizado en el proyecto FEDER-CICYT con referencia TAP 1FD97-1387, titulado "La Gestión de la Cadena de Suministro en Contexto de Integración Empresarial".

En este entorno tan ajustado de producción el Programa Secuencial debe evitar en todo momento producir sobrecargas puntuales en las estaciones tanto de la línea principal como en los proveedores. Dado que las estaciones están conectadas por la línea, la Secuencia de vehículos debe contemplar todas las estaciones simultáneamente. Y dado que la demanda de Opciones crece, el problema se complica para cada estación.

1.2 Posible consideración de los proveedores en el problema

En la actualidad las restricciones del sistema en el sector del automóvil se determinan de acuerdo con las circunstancias del Sistema Productivo del ensamblador, aunque podría mejorar la eficiencia del sistema global si estas restricciones se definieran atendiendo también a las circunstancias de los subsistemas productivos de los proveedores secuenciados-sincronizados, en lo que es denominado en [2] como Empresa Extendida o Empresa Virtual.

Para el problema concreto que aquí se planteará, las restricciones que se consideran serán exclusivamente restricciones de separación (p.ej. no poder fabricar más de 2 de cada 5 vehículos que consuman una determinada opción), por considerarse este tipo de restricción como el más relevante en la problemática específica de las plantas de montaje de automóviles.

Independientemente de qué circunstancias de los proveedores se tengan en cuenta o no para determinar las restricciones, una vez definidas, el problema se puede modelizar como sigue.

2 MODELIZACIÓN DEL PROBLEMA

La nomenclatura de las constantes, índices parámetros y variables necesarios para definir completamente el problema se exponen a continuación:

CONSTANTES:

- **T** = Número de Unidades a secuenciar
- **O** = Número de Opciones[1] consideradas
- **P** = Número de tipos de Productos[2]

ÍNDICES:

- **i** = Índice referido al tipo de Producto ($1 \leq i \leq P$)
- **j** = Índice referido a las denominadas opciones ($1 \leq j \leq O$)
- **k,t** = Índice referido a la posición k secuenciada ($1 \leq k \leq T$) ($1 \leq t \leq T$)

PARÁMETROS:

- $\mathbf{U_i}$ = Número de Productos de tipo i a secuenciar.
- $\mathbf{n_{i,j}}$ = La cantidad de opción j en el Producto tipo i.
- $\mathbf{r_j}$ = Tasa media del componente j o de la opción j, durante el programa de producción.
- $\mathbf{\Omega}$ = Subconjunto de las O opciones que tienen restriccion de separación definida.
- $\mathbf{L_j}$ = Longitud del Tramo en el que hay que considerar la separación de la opción $j \in \Omega$.
- $\mathbf{M_j}$ = Valor máximo de apariciones que debe tener la opción $j \in \Omega$ en el tramo L_j.

VARIABLES:

- $\mathbf{X_{j,k}}$ = Consumo total de Opción **j** hasta la posición **k** secuenciada.
- $\mathbf{y_{i,k}}$ = Variable binaria que vale 1 si en el instante k se ha secuenciado un Producto de tipo **i.**
- $\mathbf{s_k}$ = Tipo de Producto secuenciado en k-ésima posición.
- $\mathbf{S^A}$ = Vector formado por los valores de la secuencia ($s_1, s_{2\ldots}\ s_T$) obtenidos mediante la heurística A.

(1) En adelante las denominadas "opciones" englobarán indistintamente a componentes u opciones montadas (o no) en un producto.
(2) Un "tipo" englobará a todo producto que monte unas determinadas opciones definidas mediante $\mathbf{n_{i,j}}$

Tabla1: Nomenclatura

Se entiende como unidades homogéneas aquellas que aún siendo distintas entre sí, poseen gran cantidad de características comunes, tantas que no sólo pertenecen a la misma familia de productos sino que pueden ser consideradas como prácticamente idénticas, salvo variantes en determinadas opciones. Los automóviles, por tanto, se pueden considerar como unidades homogéneas, y tomando sólo estas determinadas opciones, el problema que aquí se caracteriza es un problema (tratado entre otros por Bautista en [3]) de:

"*Secuenciar **T** unidades homogéneas (automóviles), de las cuales cada una de ellas tiene un consumo determinado de cada una de las **O** opciones existentes; donde cada unidad de tipo **i** (1£ **i** £ **P**) tiene un consumo de opción **j** (1£**j**£**O**) igual a $\boldsymbol{n_{i,j}}$ y donde existen para cada tipo **i** un número $\boldsymbol{U_i}$ de unidades a secuenciar*".

Para cada posición **k**, la secuencia $\mathbf{s_k}$ determinará que tipo de vehículo irá secuenciado en k-ésima posición, de forma que se minimizen los dos objetivos que se formulan matemáticamente en la tabla 2 como:

1) <u>OBJ1</u>: Regularidad en cuanto a la aparición de las opciones, regularidad que resulta fundamental tanto para el ensamblador como para los proveedores sincronizados, como queda reflejado en [4]. Esto se plantea matemáticamente como minimizar en cada instante de secuenciación **k** la discrepancia entre el consumo real $\mathbf{X_{j,k}}$ y el ideal ($r_j \cdot k$) para cada opción **j** ([r.1] y [r.2]).

2) <u>OBJ2</u>: Minimizar el incumplimiento de unas restricciones de separación de aparición de opciones, definidas sólo para un subconjunto **Ω** de las **O** opciones (definidas en [r.3]). También fundamental, pues su incumplimiento llevaría a la sobrecarga de trabajo en las estaciones tratada en [5].

Así, el problema tiene simultáneamente dos objetivos no coincidentes y puede asumirse que el Modelo Matemático simplificado para este problema es:

OBJ1 [MIN] $\displaystyle\sum_{k=1}^{T}\sum_{j=1}^{O}\left(X_{j,k}-r_j k\right)^2$ OBJ2 [MIN] $\displaystyle\sum_{k=1}^{T}\sum_{j\in\Omega} d_{j,k}$

<u>sujeto a</u>:

$$r_j = \frac{\sum_{i=1}^{P}\left(n_{i,j}\cdot U_i\right)}{T}; \qquad \forall j,\ 1\le j<O \qquad [r.1]$$

$$X_{j,k} = \sum_{t=1}^{k}\sum_{i=1}^{P} n_{i,j}\cdot y_{i,t} \qquad \forall j,\ 1\le j<O;\ \forall k,\ 1\le k<T \qquad [r.2]$$

$$(X_{j,k} - X_{j,k-L_j}) - N_j \cdot d_{j,k} \le M_j \qquad \forall k,\ 1\le k<T;\ \forall\, j\in\Omega \qquad [r.3]$$

$$\sum_{k=1}^{T} y_{i,k} = U_i \qquad \forall i,\ 1\le i\le P \qquad [r.4]$$

	$s_k = \sum_{i=1}^{P} i \cdot y_{i,k}$	$\forall$k, 1≤k<T	[r.5]
	$\sum_{i=1}^{P} y_{i,k} = 1$	$\forall$k, 1≤k<T	[r.6]
con:	$N_j \geq M_j + 1$	$\forall j \in \Omega$	
	$y_{i,k} \in \{0,1\}$ Î ***Z*** *para i=1...P, k=1..T*		
	$d_{j,k} \in \{0,1\}$ Î *Z para j=1...O, k=1..T*		

Tabla 2: Modelización matemática del problema

Donde tenemos que la función de [r.1], [r.2] y [r.3] es calcular las diferentes variables que permiten la evaluación de las Funciones Objetivo:

- [r.1] define para cada opción j su ratio medio respecto a T
- [r.2] permite calcular la cantidad total de cada opción j secuenciada en cada instante k.
- [r.3] define perfectamente las restricciones de separación de cada una de las opciones $j \in \Omega$. Para ello se utiliza la variable binaria $d_{j,k}$, que será igual a la unidad cuando exista un incumplimiento en el instante *k* de la restricción de separación asociada a la opción *j,* y a cero cuando éste no exista.

Y en cuanto a las demás restricciones del modelo matemático:

- [r.4] obliga a secuenciar el total de Productos programado.
- [r.5] transforma la Secuencia de notación matricial simple a notación alfabética, estas notaciones, así como su transformación es ampliamente explicada en [6].
- [r.6] implica que en cada instante de secuenciación sólo se puede programar un Producto.

Una vez formulado el problema, en el siguiente apartado se concretará el procedimiento de resolución propuesto.

3 PLANTEAMIENTO GENERAL DE RESOLUCIÓN DEL PROBLEMA

Descartando los Métodos Exactos, cuando se trata de *problemas reales* de gran magnitud, en [6] se muestra que para este tipo de problemas ninguno de los Procedimientos Heurísticos allí propuestos puede garantizar, en todos los casos, un mejor comportamiento que el resto.

Cuando la cantidad de productos a secuenciar así como la cantidad de opciones de cada producto son altas, la doble necesidad de obtener unos buenos, aunque no óptimos, resultados y a la vez un reducido tiempo de computación es la mayoría de las veces difícilmente conciliable. Por regla general las heurísticas que proporcionan mejores resultados combinados, en cuanto a los dos objetivos no coincidentes OBJ1 y OBJ2 definidos en la Tabla 2, suelen ser las que mayor tiempo de computación emplean.

En la literatura existen una serie de procedimientos heurísticos que son realmente eficaces cuando el número de unidades a secuenciar y de opciones consideradas no es muy alto, pero cuyo tiempo de computación crece sustancialmente cuando la magnitud del número de ambos se acerca a las *magnitudes reales* en el sector del automóvil. Un ejemplo de esto es el *Algoritmo_3Rc* propuesto en [6] que se describe a continuación.

3.1 Algoritmo_3Rc

El *Algoritmo_3Rc* es una heurística de búsqueda en profundidad que en cada etapa selecciona aquel producto que sea el primero de la mejor subsecuencia de todas las posibles combinaciones de tres candidatos, de entre las unidades que quedan por secuenciar.

Si se define Δ3 como el número de incumplimientos registrados para cada una de estas subsecuencias de tres unidades, entonces dada una secuencia s_{k-1}, la elección del tipo de producto a secuenciar en la posición **k** consiste en tomar aquel producto, no secuenciado totalmente, que haga mínimo el valor de los incumplimientos $\Delta 3_{i,k}(s_{k-1})$:

$$\Delta 3_{\tilde{n}}(S_{k-1}) = \min_{\substack{i \le P \\ Y_{i,k-1} < U_i}} \Delta 3_i (S_{k-1}) \qquad (1)$$

Obviamente se producirán muchos empates, sobre todo en las primeras etapas, que se han de resolver. El mecanismo de tratamiento de Empates de *3Rc* evalúa para ello la aportación que, la incorporación de un producto **i**, y la consiguiente no incorporación del resto, tendría en OBJ1, mediante la Discrepancia $D_{i,k}(s_{k-1})$:

$$D_{i,k}(s_{k-1}) = \sum_{j=1}^{O} \left(\sum_{t=1}^{k-1} n_{s(t),j} + n_{i,j} - t \cdot r_j \right)^2 \qquad (2)$$

Básicamente se puede decir que el Parámetro Básico del Algoritmo de Persecución de Objetivos de **Monden,** relativo a la Discrepancia entre el consumo real y el consumo ideal de opciones, pasa a ser el criterio de desempate en este algoritmo. La expresión de esta regla de desempate quedaría por tanto así:

> *Sea* $\boldsymbol{Q_k}$ *el conjunto de todos los productos ñ tales que, no estando totalmente secuenciados, el valor de la aportación debida a la violación de restricciones es mínima:*
>
> $$Q_k = \{ \tilde{n} £ P \ / \ \Delta_{\tilde{n},k}(S_{k-1}) = \min_{\substack{i \le P \\ Y_{i,k} < U_i}} \Delta_{i,k}(S_{k-1}) \} \qquad (3)$$
>
> *Se secuenciará en k-ésimo lugar aquel* ***i Î*** Q_k *que cumple que* $D_{i,k}(S_{k-1}) = \min_{\tilde{n} \in Q_k} D_{\tilde{n},k}(S_{k-1})$, *con* $D_{i,k}$ *definido en la expresión (2).*

Es por tanto un algoritmo que construye eficientemente la secuencia en base a los dos objetivos definidos:

- en primer lugar construye la secuencia atendiendo al no incumplimiento de restricciones.
- en segundo lugar a la regularidad en la aparición de opciones.

En esta línea, el objetivo general del procedimiento que se propone a continuación es poder aprovechar la potencia de esta heurística para su aplicación a *problemas reales* de gran tamaño rebajando el gran coste computacional que supone su aplicación directa.

3.2 Procedimiento propuesto

Dado un problema Q definido perfectamente mediante los parámetros descritos en el apartado 2 (T, P, O, U_{i}, M_j, L_j …) el procedimiento que se propone consta de dos etapas:

- 1ª ETAPA : El objetivo de esta primera etapa será conseguir **H** subproblemas de menor magnitud con los mismos parámetros y restricciones de separación que Q, pero con $\left(T/H\right)$ unidades a secuenciar cada uno y con una cantidad de cada tipo de producto $Z_{i,h}$ (1≤ i ≤ **P**) en cada subproblema q_h (1≤h≤**H**) tal que:

$$\sum_{h=1}^{H} Z_{i,h} = U_i \quad \forall i,\ 1 \le i \le P \qquad \text{con: } Z_{i,h} \ge 0 \qquad (4)$$

  Para obtener estos H subproblemas se han planteado dos posibles alternativas (con heurísticas rápidas que relajan el OBJ2, ver tabla 2) que serán descritas en el siguiente apartado.

- 2 ª ETAPA : Una vez definidos éstos H subproblemas, se les aplica el *Algoritmo_3Rc*, obteniendo para cada q_h su secuencia parcial correspondiente. Estas secuencias se yuxtaponen por orden, conformando la secuencia global resultado del problema inicial, la cual puede ser evaluada y comparada con la que resultaría de aplicar a Q el *Algoritmo_3Rc* directamente.

Dado que el tiempo computacional empleado por *3Rc* aumenta con el número de posibles combinaciones de tres candidatos a comparar en cada iteración, es obvio que su aplicación a Q por partes resultará más eficiente computacionalmente. Esto se verá en la experimentación descrita en el apartado 6, donde se ha considerado el procedimiento para distinto número de particiones **H**, comprobándose posteriormente su eficiencia. Antes de pasar a ese apartado se definirá completamente el procedimiento en cada una de sus dos etapas.

4 PRIMERA ETAPA: HEURÍSTICAS EMPLEADAS

En la primera etapa del procedimiento propuesto el principal objetivo es conseguir una relativamente aceptable regularidad en el reparto de las opciones con el menor coste computacional posible. Por ello se emplearán heurísticas que se centran exclusivamente en minimizar OBJ1, relajando OBJ2 (ver tabla 2). Se proponen dos alternativas:

4.1 Primera alternativa: Algoritmo_0 y Reagrupamiento

En esta alternativa planteada, primero se aplica el **Algoritmo_0** al problema inicial Q, llegando a la secuencia $\mathbf{S^0}$.

Este Algoritmo presentado en [6] está basado en el Método de Persecución de Objetivos propuesto por Monden en [7]. Se basa en la construcción progresiva de la Secuencia siguiendo el criterio de minimizar la discrepancia entre el consumo real y el ideal para cada opción. Es por tanto un algoritmo que se centra exclusivamente en minimizar OBJ1 y construye la secuencia sin tener en cuenta las restricciones de separación (aunque puedan ser evaluadas a posteriori) y por otro lado resulta suficientemente rápido.
Una vez aplicado el **Algoritmo_0** a Q se divide la secuencia $\mathbf{S^0}$ obtenida en **H** particiones y para cada una de ellas se calcula el total de apariciones de cada tipo de producto en cada

partición $Z_{i,h}$. Con esto y con los mismos parámetros y restricciones del problema inicial quedarán definidos los H subproblemas. Este reagrupamiento por tipos se obtiene mediante:

$$Z_{i,h} = \sum_{k=(h-1)\cdot(T/H)}^{h\cdot(T/H)} y_{i,k+1} \qquad \forall i,\ 1 \le i \le P \qquad (5)$$

Con los H subproblemas generados ya se puede pasar a la segunda etapa del procedimiento en que se aplica *3Rc* a cada subproblema y se yuxtaponen las secuencias resultado de cada subproblema, tal y como se ha explicado.

Después de aplicar el procedimiento propuesto con esta alternativa a un *caso real*, se detectó que existen algunos tipos de unidades que por tener pocas opciones montadas son más "golosos" para el *Algoritmo_0*. Estas unidades se sitúan generalmente en posiciones avanzadas de la secuencia. Por otro lado hay otro tipo de unidades (los coches de versiones que llevan montadas muchas opciones) que resultan mucho más difíciles de secuenciar, soliendo quedar más retrasados en la secuencia obtenida.

Esto provoca en la segunda etapa del procedimiento que las primeras particiones contengan muchos elementos "golosos" muy fáciles de secuenciar sin incumplimientos por el *Algoritmo_3Rc*, mientras que las últimas contienen muchas más unidades de los tipos más complejos, siendo prácticamente imposible generar la secuencia de esas particiones sin incumplir restricciones.

Por todo esto se aplicó al *caso real* una segunda alternativa que paliaba en parte este defecto.

4.2 Segunda alternativa: Reparto Uniforme de Tipos de Unidades

Se puede decir que cuando el número de unidades a secuenciar **T** es suficientemente alto, o el número de particiones **H** es suficientemente bajo como para que todas las futuras particiones de la secuencia contengan unidades de todos los tipos (**U_1…U_P**), repartir carga de opciones entre las distintas particiones es equivalente a repartir tipos de unidades entre esas particiones.

Dado que en esta etapa se valora como más importante el conseguir un reparto uniforme de las unidades "golosas" entre los distintos subproblemas, se puede asumir que el modelo que definiría las cantidades de cada tipo de producto **i** que deben ir en cada subproblema **h** se puede obtener, sin necesidad de un secuenciado y posterior reagrupamiento, directamente de:

$$[\text{MIN}] \quad \sum_{i=1}^{P}\sum_{h=1}^{H}\left|Z_{i,h}-\frac{U_i}{H}\right|$$

<u>sujeto a</u>:

$$Z_{i,h} \ge 0$$

$$\sum_{h=1}^{H} Z_{i,h} = U_i \qquad \forall i,\ 1 \le i \le P$$

$$\sum_{i=1}^{P} Z_{i,h} = \left(T/H\right) \qquad \forall h,\ 1 \le h \le H$$

Tabla 3: Modelización Reparto Uniforme de Tipos de Unidades

Que es una variante reducida del problema propuesto en [8] y por tanto se puede utilizar para su resolución el *Algoritmo_1* allí propuesto con ligeras modificaciones. Por otro lado este procedimiento conduciría idealmente a la situación planteada en [9].

Con esta fórmula no es necesario generar una secuencia, partirla y agrupar por tipos de unidades para llegar a $Z_{i,h}$ ($\forall$i,1≤i≤P; $\forall$h,1≤h≤H). Estas cantidades son halladas directamente a partir de un reparto de unidades que respeta las proporciones de cada tipo respecto al total que habían en el problema inicial Q.

5 SEGUNDA ETAPA

Llegados a este punto conviene establecer una nomenclatura, acorde con la de la Tabla1, para las secuencias obtenidas mediante el procedimiento propuesto, tanto con una alternativa como con la otra:

- La secuencia final obtenida al yuxtaponer las subsecuencias de los H subproblemas creados mediante la Alternativa 1 será $\mathbf{S^{0\_3Rc}}$, y nos referiremos al método general como *Algoritmo_0+3Rc*.
- La secuencia final obtenida al yuxtaponer las subsecuencias de los H subproblemas creados mediante la Alternativa 2 de Reparto será $\mathbf{S^{Rep\_3Rc}}$, y nos referiremos al método general como *Alg.Reparto+3Rc*.

Así, el esquema general tanto del procedimiento propuesto como de la experimentación realizada es el siguiente:

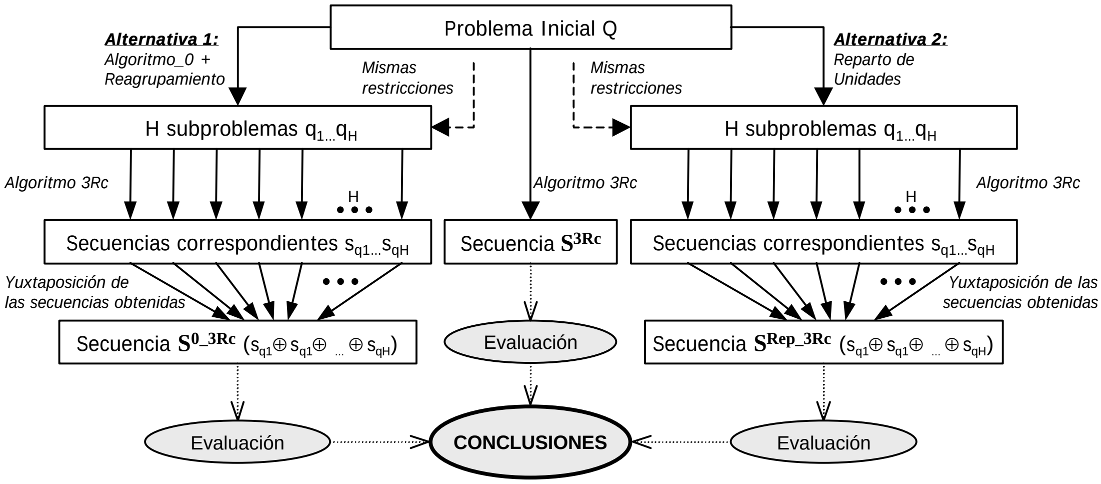

Figura 1: Esquema general del procedimiento

Una vez establecida esta nomenclatura se pasa a describir como se han tratado ciertos aspectos relacionados con la segunda etapa.

5.1 Resolución del problema de los extremos de las secuencias

En la segunda etapa del procedimiento, al aplicar el *Algoritmo_3Rc* a los **H** subproblemas, se debe tener en cuenta un elemento importante para poder comparar equitativamente la

secuencia yuxtapuesta final con la sec $\mathbf{S}^{3Rc}$
3Rc (ver Figura 1).

Dado que este algoritmo construye la secuencia en base al menor incumplimiento de restricciones de separación, al evaluar cual es la subsecuencia potencial de tres unidades de as posibles que menos incumple, el algoritmo debe considerar también los elementos ya secuenciados en el subproblema anterior.

Esto ha sido tenido en cuenta en el procedimiento propuesto, y así por ejemplo si tenemos una restricción de separación del tipo *"No más de 4 unidades con opción 28 de cada 7 unidades consecutivas"* y estamos construyendo con *3Rc* la segunda secuencia del quinto subproblema, el procedimiento considera también los últimos cinco elementos ya secuenciados del cuarto subproblema.

5.2 Mejora en la Concentración de los incumplimientos

El hecho de que el *Algoritmo_3Rc* elija atendiendo en primer lugar el cumplimiento de las restricciones de separación, hace que deje para el final las unidades más problemáticas, por lo que los incumplimientos en este algoritmo se suelen producir al final. En este sentido el procedimiento de partición propuesto compensa este defecto repartiendo también los posibles incumplimientos entre las distintas particiones (cuando estos sean inevitables).

Para ilustrar este fenómeno de forma sencilla se expone en la figura 2 las secuencias resultado de un problema, en este caso no real, que lo presenta: esta definido con 2000 unidades a secuenciar, 50 tipos de productos distintos, con combinaciones de 27 opciones a montar o no. Las restricciones son pocas y relativamente blandas.

En su resolución se han marcado los incumplimientos con una **X** y se han seleccionado por un lado los finales de las, en este caso diez, subsecuencias comprendidas en $\mathbf{S}^{Rep\_3Rc}$ y por otro el final de la secuencia $\mathbf{S}^{3Rc}$ obtenida con el *Algoritmo_3Rc* sin más:

2000_A_PROB_1_1_1_1_dRRB.pro
Algoritmo REPARTO+3Rc con H=10 PARTICIONES >> **tpo:** 978.5 seg. **OBJ1:** 23726.6 **Incumplimientos:** 10

$\mathbf{S}^{Rep\_3Rc}$**:**

.p8.p46.p7.p33.p12...	....p49.p27.p46.p32.p41.p48.**X**49.**X**30
.p24.p8.p9.p13.p35...	....p41.p49.p17.p27.p43.p48.p41.p49
.p24.p37.p9.p32.p35...	....p27.p36.p48.p28.p37.**X**50.**X**28.**X**38
.p24.p8.p39.p33.p35...	....p50.p48.p22.p49.p28.p48.p42.p50
.p4.p46.p29.p33.p12...	....p48.p42.p27.p50.p46.p48.p18.p49
.p24.p37.p9.p32.p35...	....p50.p27.p42.p50.p38.p48.**X**49.**X**28
.p24.p8.p9.p13.p35...	....p41.p49.p17.p27.p43.p48.p41.p49
.p24.p37.p9.p32.p35...	....p27.p36.p48.p28.p37.**X**50.**X**28.**X**38
.p24.p8.p39.p33.p35...	....p50.p48.p22.p49.p28.p48.p42.p50
.p4.p46.p29.p33.p12...	....p48.p42.p27.p50.p46.p48.p18.p49

2000_A_PROB_1_1_1_1_dRRB.pro
Algoritmo 3Rc >> **tpo:** 1897.67 seg. **OBJ1:** 67713.8 **Incumplimientos:** 7

$\mathbf{S}^{0\_3Rc}$**:**
.p8.p46.p7.p33.p1.... ...p27.**X**43.p48.p37.p36.p48.p37.p50.p43.p38.p50.p27.**X**48.p42.p28.p50.p48.**X**38.**X**49.**X**28.**X**28.**X**38

Figura 2: Ejemplo de aplicación

ridad y el tiempo computacional, y respecto a los incumplimientos se observa que:

- Se produce un reparto de los incumplimientos en los finales de cada partición y no todos juntos al final de la secuencia (como se puede observar en una secuencia de 2000 uni en **S** se acumulan los 7 incumplimientos en el tramo de las últimas 22 unidades).

- $\mathbf{S}^{\mathbf{Rep\_3Rc}}$ dos factores:
  - **T** mero de tipos de **P** unidades "golosas" respecto al total…)
  - **L** , **j**), ya que una misma restricción de separación ta más dura cuando el número de unidades a

  sucede en determinados subproblemas si las restricciones les resultan demasiado n **Rep_3Rc** **S** .

(T/H) respecto a los tramos de las restricciones de separación si se quiere obtener un buen resultado no sólo en regularidad sino también en incumplimientos. Esto se propone en el último apartado como una línea futura a investigar.

6 EXPERIENCIA CON UN CASO REAL

El procedimiento propuesto se aplicó a un *caso real* del sector del automóvil que se da al segmentar y secuenciar la producción de coches para 5 días de producción a dos turnos de una planta de montaje (1600 unidades por día).

Dada la magnitud de los parámetros de definición del problema sólo se describirá cualitativamente: se trata de un problema de 8000 unidades a secuenciar, con 170 tipos de unidades distintos, 53 opciones, que pueden ir montadas, o no, en cada tipo y algunas de ellas con restricciones de separación, que en ningún caso son más duras de "no más de 1 aparición en cada 4 unidades secuenciadas" (dado que ésta es la máxima restricción de separación que se suele encontrar en una planta de montaje de automóviles, pues en general pueden considerarse ligadas al número de vehículos que puede atender una estación).

Los resultados de aplicar por un lado el *Algoritmo_3Rc* al problema y por otro el procedimiento propuesto según las dos alternativas comentadas (*Algoritmo_0+3Rc* y *Reparto+3Rc*) y con diferente número de particiones **H**, han sido los siguientes:

Alg. Empleado	***Reparto+3Rc***				***Algoritmo_0 + 3Rc***				***3Rc***
Nº particiones	**4**	**5**	**8**	**10**	**4**	**5**	**8**	**10**	
T.Computación(s)	7674.9	7634.7	7241.7	7316.9	5624.2	5396.5	5093.2	4591	13701.6
OBJ1 (regularidad)	131 632	356 663	407 831	111 437	258 336	152733	183465	88836	1810100
OBJ2 (Incump.)	26	45	117	24	27	26	47	27	31
Tramo max Incump*	5	7	11	6	7	4	6	6	17

* Tramo de secuencias más largo de entre los que se dan incumplimientos de restricción seguidos (ver punto 5.2)

Tabla 4: Tabla de resultados para *caso real*

En la figura 4 se puede observar que el tiempo computacional mejora en todos los casos respecto a la aplicación del *Algoritmo_3Rc* al problema. También se observa que se han obtenido resultados diversos en cuanto a regularidad e incumplimientos según los algoritmos y el número de particiones empleados, pero varios de ellos mejoran ambos conceptos simultáneamente.
En este sentido, se puede considerar que para este tipo de problema concreto se consiguen los mejores resultados combinados aplicando *Reparto+3Rc* con 10 particiones.

7 APLICACIÓN AL PROBLEMA DE LA SEGMENTACIÓN

En el sector del automóvil la Segmentación de la Producción normalmente se realiza a partir de un *pool* de pedidos en firme tanto de clientes reales como de concesionarios. Los Departamentos de Planificación, a partir de este *pool* corporativo de pedidos, van confeccionando las distintas segmentaciones y asignándoles una factoría (en caso de fabricarse el modelo en distintas factorías) y una fecha determinada.

El procedimiento de particiones propuesto puede ser usado tanto en la Secuenciación como en el ámbito de la Segmentación de la producción de vehículos, siendo un mecanismo eficaz para conseguir una máxima regularidad en dicha segmentación. Esta regularidad previa de las particiones segmentadas puede, más tarde, facilitar enormemente la obtención de una Secuencia en cada uno de los bloques segmentados que minimice la aparición de incumplimientos, así como regularice internamente la regularidad en cuanto a opciones.

El procedimiento obviamente sólo puede ser empleado en coordinación con los objetivos marcados desde estos Departamentos de Planificación corporativos. Esta propuesta incide en el hecho de que la Segmentación que éstos realizan (que normalmente no prevé la posterior secuenciación de las unidades) puede reajustarse para buscar una mayor regularidad en vehículos como se ha descrito en el algoritmo de reparto. Es decir, una vez establecido el *pool* de pedidos, a la hora de realizar la segmentación en totales de vehículos de cada tipo para cada bloque de días o semanas segmentado, ésta puede hacerse dividiendo previamente en diferentes particiones que regularicen el reparto de unidades y por tanto de componentes.

La posterior Secuenciación de estos bloques uniformes resultará más eficiente y además, como se ha dicho, los posibles incumplimientos no serán acumulados al final del bloque segmentado, como podría suceder.

8 CONCLUSIONES Y LÍNEAS FUTURAS

Se ha establecido un procedimiento general de resolución de *Problemas Reales de Secuenciación Estática de Unidades Homogéneas* alternativo y complementario a los ya existentes; constatándose la posibilidad de mejorar la Segmentación y su posterior Secuenciación mediante este procedimiento. Se trata fundamentalmente de dividir un problema de grandes magnitudes en pequeños subproblemas para poder secuenciarlos mediante el *Algoritmo_3Rc* y yuxtaponer las secuencias obtenidas en una secuencia global resultado del problema inicial.

Este procedimiento es aplicable a cualquier problema y, aunque para la experimentación efectuada se ha realizado con el *Algoritmo_3Rc,* el procedimiento es abierto y fácilmente extrapolable para cualquier heurística lenta en la que se pretenda comprobar su posible mejora computacional mediante particiones.

Después de contrastar el procedimiento propuesto con un *caso real* de grandes magnitudes se concluye su validez en el entorno analizado, ya que el tiempo computacional mejora en todos los casos habiéndose obtenido resultados diversos en cuanto regularidad e incumplimientos. El análisis de los resultados en función de los algoritmos empleados en la primera etapa y del valor de (T/H) considerado permite establecer en que situación se mejora todas las funciones objetivo definidas de forma simultánea.

Una de las líneas futuras a investigar es precisamente el poder establecer a priori cual debería ser el número ideal de particiones H para cada problema concreto, según sus condiciones particulares de número total de unidades, dureza de restricciones, matriz de opciones…u otros parámetros que pudieran influir.

Otra de las líneas a explorar sería la posibilidad de emplear técnicas de Computación en Paralelo para la resolución de problemas de gran magnitud. En principio no parece haber ningún problema para ello, dada la completa modularidad de las etapas del procedimiento propuesto y la posibilidad de emplear máquinas distintas para computar la secuenciación de cada subproblema, para luego yuxtaponer el resultado final. Esto mejoraría los tiempos de computación conseguidos.

Referencias

[1] Monden, Y.; El Just-In-Time hoy en Toyota; Ediciones Deusto; 1994

[2] Centro de Investigación de Gestión e Ingeniería de Producción (CIGIP); Proyecto FEDER-CICYT TAP 1FD97-1387 “"La Gestión de la Cadena de Suministro en Contexto de Integración Empresarial"; 2001

[3]] Bautista, J. ; Procedimientos Heurísticos y Exactos para la secuenciación en Sistemas Productivos de Unidades Homogéneas(Contexto JIT); Tesis Doctoral; DOE, UPC; 1993

[4] Monden, Y.; What Makes The Toyota Production System Really Tick?; Institute of Industrial Engineers Press; 1981

[5] Yano, C. y Rachamadugu; Sequencing to minimize work overload in Assembly line with Product Options; Management Science, 37,5; 1991

[6] García-Sabater J.P.; “Modelos Métodos y Algoritmos de Resolución del Problema de Secuenciación de Unidades Homogéneas en el Sector del Automóvil”, Tesis Doctoral, UPV, 2000

[7] Monden,Y.; El Sistema de Producción Toyota; Ed. IESE; 1987

[8] Miltenburg, J.; Level Schedules for Mixed Models Assembly Lines in JIT Production System; Management Science, 35,2; 1989

[9] Kubiak, W; Product Rate Variation Problem and Greatest Common Divisor Property; 1998