

Mejoras de las Heurísticas Greedy empleadas en el secuenciamiento de los Sistemas de Producción Multi-modelo y Multi-nivel

Juan José Areal Alonso, Julio Garrido Campos*, Ricardo Marín Martín

^a *Departamento de Ingeniería de Sistemas y Automática, Universidad de Vigo, Escuela de Ingeniería Industrial, 36200, Vigo, España*

Resumen

El problema de secuenciamiento de los sistemas de producción multi-modelo y multi-nivel reside en el alisado de las tasas de producción de los productos finales y también de las tasas de consumo de los diversos subconjuntos y elementos empleados en las etapas previas de producción. Las secuencias deben construirse de acuerdo a las opciones de cada producto, que requieren diferentes recursos y tiempo de producción, siendo el objetivo evitar sobrepasar el potencial de la instalación y de los medios humanos. Este artículo se desarrolla a partir del método Goal-chasing, que es una heurística Greedy desarrollada por Toyota para resolver este problema y que es ampliamente utilizado en la industria del automóvil. El artículo propone una mejora de dicho método con la introducción de pesos diferentes para cada opción con el fin de mejorar el ordenamiento de la secuencia. Profundizando en esta vía, se aplica el método Nelder-Mead de optimización no lineal para obtener los pesos de opciones que minimizan el coste de la secuencia resultante. Los resultados obtenidos aplicando los algoritmos a sistemas de producción multi-modelo y multi-nivel se concretan en el mundo del automóvil para una secuencia inicial, se generalizan a un conjunto de 30 secuencias representativas del entorno industrial del automóvil y se contrastan con referencias clásicas de la literatura. *Copyright © 2013 CEA. Publicado por Elsevier España, S.L. Todos los derechos reservados.*

Palabras Clave:

Heurísticas, estimación de secuencias, problemas de optimización, sistema de producción, industria del automóvil.

1. Introducción

En los sistemas de producción multi-modelo y multi-nivel es necesario mantener un cierto orden en los productos cuando discurren a través de la línea de producción. El término multi-modelo se aplica a la producción de una serie de productos diferentes pero que guardan una gran similitud entre los elementos que los componen. El término multi-nivel designa una estructuración por capas o estratos entre los diferentes elementos que componen el producto.

En el caso particular de la industria del automóvil, cada producto se caracteriza por un conjunto de opciones diferentes (techo-cielo, ABS, sistemas de navegación, etc.). Estas opciones deben ser ordenadas correctamente de forma que no se sobrepase el potencial de la instalación y de los medios humanos.

El flujo de vehículos se corresponde con el flujo físico de las carrocerías en una fábrica de producción, compuestas típicamente de tres talleres: soldadura, pintura y montaje (Figura 1). El flujo comienza en el punto de bautismo de la Orden de Fabricación en

el taller de Soldadura y se termina en el punto de comercialización del fin de fábrica. Cada uno de estos tres talleres está organizado para funcionar en FIFO (First-In-First-Out), pero tienen requerimientos diferentes a la hora de conformar las secuencias, lo que lleva a que sea necesario realizar un resecuenciamiento entre ellos. Así por ejemplo, la constitución de un convoy de vehículos permite responder a las necesidades de pintura, más particularmente a las del proceso de lacas. A cada cambio de color (por ejemplo que un vehículo rojo siga a uno blanco), los sistemas de pintura robotizados deben purgar los conductos que van desde los contenedores de pintura hasta las pistolas de aplicación. Esta operación tiene un coste no despreciable en cuanto al consumo de disolvente, por lo se que necesita el mantenimiento de un convoy de talla mínima de cada color. En la Figura 1 se muestra un esquema de flujo de vehículos característico entre los diferentes talleres, las zonas de retoques dentro de ellos y los sistemas de almacenamiento utilizados para cadenciar y re-cadenciar los vehículos de acuerdo a las necesidades.

* Autor en correspondencia.

Correos electrónicos: juanjose.areas@mpsa.com (Juan José Areal Alonso), jgarr@uvigo.es (Julio Garrido Campos), marin@uvigo.es (Ricardo Marín Martín)
URL: webs.uvigo.es/jgarr

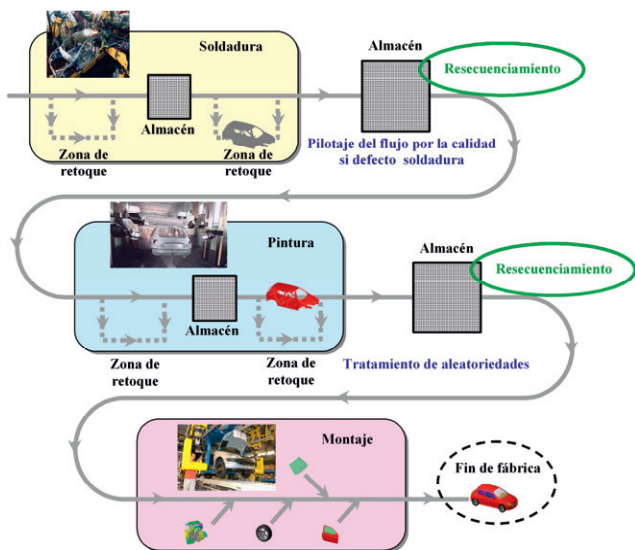


Figura 1. Esquema de flujo típico de vehículos entre los diferentes talleres y sus almacenes.

El lanzamiento de los vehículos en la entrada del taller de montaje supone que las piezas necesarias para realizar el vehículo serán efectivamente aprovisionadas con la suficiente anticipación, y ha de tener en cuenta la disponibilidad de personal. Se trata de prever y de respetar la carga instantánea de los puestos o sectores que permita un equilibrado de la línea. Para este equilibrado se emplean clásicamente técnicas de no-sucesión y el espaciamento simple. La no-sucesión consiste en no hacer que vayan seguidos dos vehículos dados, y permite responder a necesidades del montaje del tipo: impedir que un vehículo “dirección a derechas” siga a un vehículo “techo con abertura”, puesto que estos dos tipos de coches sufren operaciones largas que, acumuladas, pueden retrasar ciertos puestos de la línea si estas opciones no están espaciadas. Por otro lado, el espaciamento simple responde a la necesidad de los talleres de montaje de no ordenar más de x vehículos de un tipo dentro de una ventana de n vehículos, a fin de respetar la carga de los operarios o las limitaciones ligadas a las operaciones automáticas.

3 tipo A - 1 tipo B

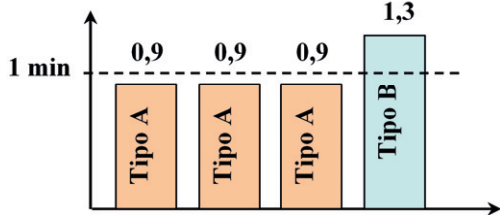


Figura 2. Ejemplo de equilibrado de puesto.

En la Figura 2 se muestra gráficamente un ejemplo básico de equilibrado de carga de trabajo atendiendo a un único criterio.

La búsqueda de la secuencia óptima entre las combinaciones posibles es un problema NP-hard (Kis, 2004). No existe ningún algoritmo de resolución polinómica para resolver problemas de este tipo, que se caracterizan por una explosión exponencial del tiempo de cálculo en función del aumento del tamaño del problema. Métodos heurísticos como Simulated Annealing (Kirkpatrick et al., 1983; Downsland, 1993), algoritmos genéticos

(Goldberg, 1989), búsqueda tabú, sistemas expertos (Parrello, 1988) y búsquedas en vecindad (Estellon et al., 2005; Cordeau et al. 2008), entre otros, proporcionan soluciones posibles que, aunque no alcancen la solución óptima, reducen la función de coste en un tiempo razonable. Un resumen del estado del arte de los métodos se describe en (Solnon et al., 2008).

De entre todos ellos, el método Goal-chasing (GCM) es una referencia en el sector del automóvil. GCM ha sido desarrollado y empleado por Toyota para ordenar el paso de los vehículos por las líneas de montaje y es también ampliamente empleado por otros constructores de automóviles dentro de sus sistemas de producción Just-In-Time (Monden, 1983).

En este artículo se parte del empleo del método Goal-chasing (descrito en la sección 3.2) para resolver el problema de búsqueda de la secuencia óptima para líneas de montaje de vehículos, y se propone una nueva función de coste C_{nesti} (sección 3.3) que mejora los resultados frente a otras funciones de coste. En la sección 4 se introduce el concepto de pesos diferentes para cada opción con el fin de mejorar el ordenamiento de la secuencia. Se presentan los resultados con cuatro vectores de pesos fijos. Posteriormente, en la sección 5 se propone la aplicación del método Nelder-Mead (Nelder and Mead, 1965) de optimización no lineal para obtener los pesos que minimizan aún más el coste de una secuencia. Finalmente, los resultados obtenidos para una secuencia inicial se generalizan a un conjunto de 30 secuencias representativas del entorno industrial y se contrastan con referencias clásicas de la literatura en la sección 6, mientras que en la sección 7 se realiza una última experimentación con datos reales extraídos de una línea de montaje industrial.

2. Modelo de secuenciamiento de vehículos y función de coste

El problema de secuenciamiento de vehículos ha sido primero descrito por Parello (Parello et al., 1988). Este problema implica el ordenar los coches a lo largo de una planta de fabricación, para instalar las opciones (por ejemplo, techo panorámico, radio, aire acondicionado) en ellos. Cada opción es instalada en una estación diferente, diseñada para realizar la operación como máximo en un cierto porcentaje de los vehículos que pasan a lo largo de la cadena de montaje, y esto implica que los coches que requieren esta opción deben ser espaciados de modo que nunca se sobrepase la capacidad de cada estación (Estellon et al., 2005).

Este requisito se puede formalizar mediante restricciones del tipo p/q : cada opción se asocia a una restricción en forma de ratio p/q que indica que cualquier sub-secuencia de q vehículos sucesivos puede incluir como máximo p vehículos que requieren esta opción.

Un caso del problema de ordenamiento de vehículos está definido por el conjunto (V, O, p, q, r) , donde:

$V = \{v_1, \dots, v_n\}$ el conjunto de vehículos que será producido.

$O = \{o_1, \dots, o_m\}$ es el conjunto de las diferentes opciones.

$p: O \rightarrow N$ y $q: O \rightarrow M$ definen la restricción de capacidad asociada a cada opción $o_i \in O$; esta restricción impone que, para cualquier sub-secuencia de q_i vehículos consecutivos en la línea, como máximo p_i de ellos pueden requerir o_i .

$r: V \times O \rightarrow \{0, 1\}$ define los requisitos de las opciones, es decir, para cada vehículo $v_j \in V$ y para cada opción $o_i \in O$, $r_{ji} = 1$ si o_i se debe instalar en v_j , y $r_{ji} = 0$ en caso contrario.

El objetivo consiste en definir el orden de los vehículos, de entre los del conjunto de los que serán producidos (V), en una secuencia determinada, estableciendo de este modo el orden en el cual pasarán a lo largo de la planta de fabricación. El problema de la decisión consiste en la búsqueda de una secuencia que satisfaga todas las restricciones de capacidad, mientras que el problema de la optimización implica el encontrar una secuencia de coste mínima, donde la función de coste evalúa las violaciones de las restricciones.

Este trabajo utiliza como punto de partida uno de los casos ya empleados por Delaval, Ohl, Gentina y Yvars (1995), que trata del ordenamiento de una secuencia de 100 vehículos con 10 opciones. Esta lista ideal de vehículos respeta la distancia mínima para cada opción restrictiva, y está construida de forma que presente un coste nulo. En esta secuencia los espaciamentos mínimos aceptables para cada opción limitativa de cada criterio son valores enteros.

A modo de ejemplo, la Tabla 1 muestra una parte de esta lista ideal correspondiente a los 10 primeros vehículos (de un total de 100), donde un “1” indica la presencia de una opción y un “0” su ausencia.

Puede verse en la Tabla 1 que para la opción 1 hay tres ceros (vehículos 2, 3 y 4) entre dos unos (vehículos 1 y 5), es decir, hay tres vehículos sin la opción limitativa entre dos que sí la presentan. La distancia se define como el número de vehículos sin la opción limitativa, en este caso de valor 3. De forma similar, la distancia mínima para la opción 9 es dos, mientras que para la opción 10 es uno.

Tabla 1. Lista parcial de vehículos antes de la mezcla aleatoria

Orden Vehículo	Opción									
	1	2	3	4	5	6	7	8	9	10
1	1	0	0	0	0	0	1	0	1	0
2	0	0	0	1	0	0	0	0	0	1
3	0	1	0	0	0	0	0	1	0	0
4	0	0	0	0	0	1	0	0	1	1
5	1	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	1	0	1
7	0	0	0	0	0	0	0	0	1	0
8	0	1	0	0	0	0	1	0	0	1
9	1	0	1	0	0	0	0	1	0	0
10	0	0	0	0	0	0	0	0	1	1
...

La Tabla 2 muestra las distancias mínimas para cada opción, mientras que la Figura 3 (arriba: secuencia ideal) representa, de forma gráfica, los espaciamentos ideales para el criterio (opción ocho). En el eje y de los gráficos de la figura se representa el espaciamento (dos para el caso de la opción 8), mientras que en el eje x se numeran en secuencia las sucesivas distancias entre vehículos que presentan la opción limitativa. Esta representación gráfica, aunque en el caso de la secuencia ideal es poco significativa (siempre plana), sí que aportará más información a lo largo del estudio ya que permitirá ver como las secuencias “no ideales” (Figura 3 centro y abajo: secuencia cadenciada) se acercan a la ideal, y el comportamiento en puntos significativos como el comienzo y final de la secuencia de vehículos.

Tabla 2. Distancia mínima para cada opción

Opción	1	2	3	4	5	6	7	8	9	10
Distancia mínima	3	4	9	8	19	7	6	2	2	1

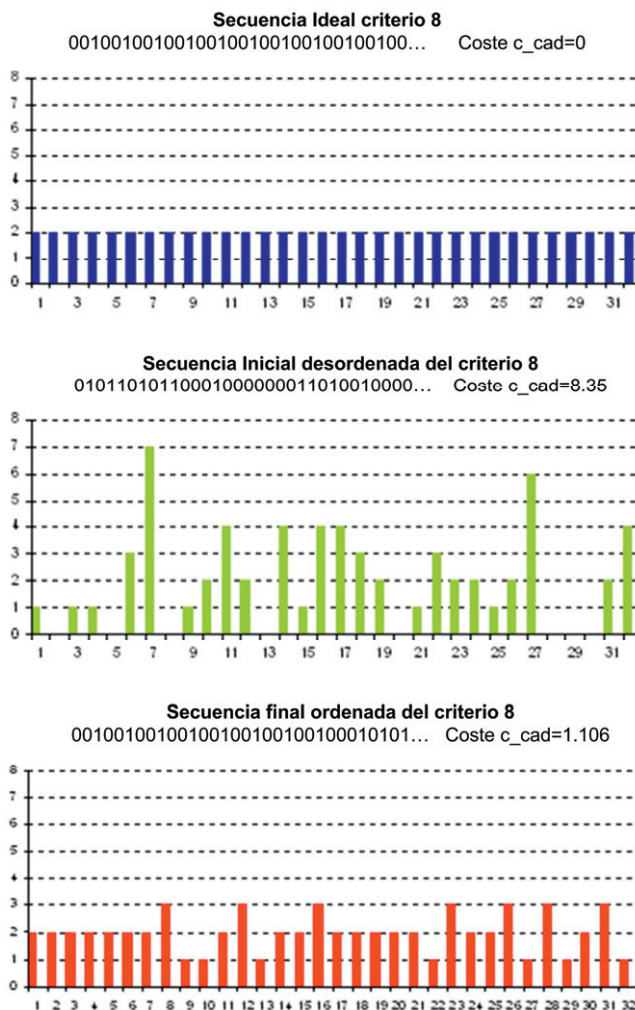


Figura 3. Arriba: espaciamentos de la secuencia ideal (opción 8). Centro: espaciamentos de la secuencia inicial de partida (opción 8). Abajo: secuencia cadenciada (opción/criterio 8).

Esta lista ideal de vehículos, después de haber sido mezclada de forma aleatoria, constituye la secuencia inicial que será utilizada como referencia para evaluar la convergencia de los algoritmos. En la Tabla 3 se muestran los valores de todos los criterios de los primeros ocho vehículos de esta secuencia inicial desordenada.

Como ejemplo, la Figura 3 (centro) muestra la distancia para una de las diez opciones (opción 8) de la secuencia inicial una vez desordenada. Las distancias instantáneas para la opción/criterio 8 son muy variables en la secuencia mezclada de forma aleatoria. Los 33 vehículos que presentan esta opción limitativa no están distribuidos uniformemente entre los 100 vehículos de la lista, con valores por encima y por debajo de la distancia mínima de valor 2.

Tabla 3. Nueva lista parcial después de la mezcla aleatoria

Orden Vehículo	Opción									
	1	2	3	4	5	6	7	8	9	10
1	1	0	0	1	0	0	0	0	0	0
2	0	0	0	0	0	1	0	1	0	1
3	0	0	0	0	0	1	0	0	1	1
4	1	1	0	0	0	0	0	1	0	0
5	0	0	0	0	0	0	0	1	0	1
6	1	0	0	0	0	0	0	0	1	0
7	0	1	0	0	0	0	0	1	0	0
8	0	0	0	0	0	0	1	0	1	1
...

La representación gráfica de la Figura 3 (centro) se corresponde con la secuencia siguiente: 0101101011000100000001101... en lugar del ordenamiento óptimo 00100100100..., cuya representación gráfica serían distancias iguales de valor 2.

En la Figura 3 se presenta una comparativa para el criterio 8 que muestra el espaciado ideal (arriba), el espaciado de la lista obtenida de desordenada de forma aleatoria la lista anterior y que constituye la lista de entrada a los algoritmos de optimización (centro), y el espaciado obtenido tras ordenar la lista anterior con un método de secuenciado (abajo). En la figura también se presenta el inicio de cada una de las secuencias anteriores para el criterio 8 en formato 0 y 1, así como el valor de la función de coste C_{cad} (introducida en la sección 3.3) para cada una de las tres secuencias del criterio 8.

Las funciones de coste sirven para evaluar de forma numérica la bondad de una secuencia. Se han utilizado históricamente diferentes expresiones relacionadas con los respectivos métodos de optimización (Solnon et al., 2008). En la sección 3.3 se presentan algunas de ellas, junto con la nueva función de coste C_{nesti} (Areal et al., 2011a), desarrollada en el marco de este estudio (Areal, 2011b), como mejora de las anteriores para el caso del secuenciado de vehículos.

La Figura 3 resalta cómo en la secuencia desordenada (medio) se ha degradado de forma importante el espaciado y ha provocado un coste C_{cad} no nulo. Al emplear un método de ordenamiento sobre la secuencia anterior se obtiene otra secuencia cadenciada cuyos espaciados han mejorado tendiendo al espaciado aceptable de la secuencia ideal. Asimismo, el coste C_{cad} se ha reducido.

La figura muestra una comparativa de un solo criterio pero en la práctica el problema abarca todos los criterios de la secuencia a ordenar, y el objetivo del ordenamiento es alisar el espaciado de todos ellos.

3. Algoritmos Greedy: Método Goal-Chasing

El enfoque ORV (Output Rate Variation) (Miltenburg y Sinnamon, 1989), está pensado para resolver el problema de secuenciado de los sistemas de producción multi-modelo y multi-nivel mediante el alisado de las tasas de producción de los productos finales y también de las tasas de consumo de los diversos subconjuntos y elementos empleados en las etapas previas del sistema de fabricación. Para el problema del ORV se

han utilizado heurísticas muy variadas : simulated annealing (Chew et al., 1992) (Xiaobo y Zhou, 1999) (Briant et al., 2008), algoritmos genéticos (Warwick y Tsang, 1995) (Ponnambalam et al., 2003) (Zinflou et al., 2007), seis heurísticas glotonas propuestas por Gottlieb (Gottlieb et al., 2003), método Goal Chasing (Monden, 1983), heurísticas de Miltenburg (Miltenburg y Sinnamon, 1989), métodos Branch-and-bound (Drexel et al., 2006), etc. Una comparativa del comportamiento de estos métodos para el problema del secuenciado de vehículos puede encontrarse en (Areal, 2011b)

3.1. Algoritmos Greedy

La estrategia de un algoritmo Greedy es que en cada una de las etapas se debe tratar de hacer lo mejor que se pueda hacer en ese momento, y para evaluar los posibles movimientos en cada paso del algoritmo se tienen en cuenta decisiones tomadas en pasos anteriores. Esto permite representar cada solución como una estructura obtenida de forma incremental a partir de una situación inicial, donde cada nuevo elemento se elige de un conjunto de posibles candidatos. Para ello se debe contar con una función que evalúe la bondad de añadir cada componente específico, cuidando además que la subestructura resultante forme parte de una solución factible (Díaz et al., 1996).

Desde este punto de vista se pueden considerar como algoritmos Greedy tanto las heurísticas de Miltenburg (Miltenburg y Sinnamon, 1989), como el método Goal-chasing (Monden, 1983).

3.2. Método Goal-chasing

Tanto las heurísticas de Miltenburg como el método Goal-chasing fueron desarrollados para resolver el problema de secuenciado de los sistemas de producción multi-modelo y multi-nivel. El término multi-modelo se aplica a la producción de una serie de productos diferentes pero que guardan una gran similitud entre los elementos que los componen y que el término multi-nivel designa una estructuración por capas o estratos entre los diferentes elementos que componen el producto. El producto (nivel 1) estará compuesto por una serie de subconjuntos (nivel 2), los cuales a su vez están integrados por componentes (nivel 3). Finalmente los componentes están formados por varias materias primas (nivel 4).

En un sistema de fabricación JIT (Just-in-Time) todos los subconjuntos, componentes y materias siguen el dictado del secuenciado de los productos, por ello, es la línea de ensamblado final la que posee el control y determina el secuenciado de los otros niveles inferiores. Uno de los grandes objetivos de un sistema JIT es que la cantidad de unidades de cada elemento (ya sean productos, subconjuntos, componentes o materias) consumidas por unidad de tiempo por el sistema de producción se mantenga tan próxima a un valor constante como sea posible (Miltenburg y Sinnamon, 1989).

El método Goal-chasing (GCM) considera solo la variabilidad en el nivel de subconjunto (nivel 2), aunque esté diseñado para sistemas de producción con dos niveles (niveles 1 y 2). El objetivo de este método es seleccionar, para cada etapa k , el producto i con el menor valor de:

$$GCM_{ik} = \sum_{l=1}^{n_2} [(x_{l,2,k-1} + t_{l,2,i}) - k d_{l,2} / DT_1]^2$$

El GCM puede extenderse a más niveles. La función objetivo (EGCM) en este caso será la siguiente:

$$\sum_{k=1}^{DT_1} \sum_{j=1}^4 \sum_{i=1}^{n_j} w_j (x_{ijk} - k d_{ij} / DT_1)^2$$

Donde:

j – número del nivel (1 producto, 2 subconjunto, 3 componente, 4 materia).

n_j – número de elementos diferentes en el nivel j , $j = 1, 2, 3, 4$.

d_{i1} – demanda para el producto i , $i = 1, 2, \dots, n_1$.

t_{ijl} – número de unidades del elemento i en el nivel j necesarias para producir una unidad de producto l . $i = 1, 2, \dots, n_j$; $j = 2, 3, 4$; $l = 1, 2, \dots, n_1$.

A partir de d_{i1} y t_{ijl} se determinan:

$$d_{ij} = \sum_{h=1}^{n_1} t_{ijh} d_{h1} \text{ – demanda de la salida } i \text{ en nivel } j, \\ i = 1, 2, \dots, n_j; j = 2, 3, 4.$$

$$DT_j = \sum_{i=1}^{n_j} d_{ij} \text{ – demanda total de producción en el nivel } j, j = 1, 2, 3, 4.$$

El secuenciamiento es la lista ordenada de los productos a fabricar. Está constituido de DT_1 etapas consecutivas a cada una de las cuales se asigna un producto determinado, y se representa por las variables siguientes:

x_{i1k} – número de unidades del producto i realizadas durante las k primeras etapas.

$x_{ijk} = \sum_{h=1}^{n_1} t_{ijh} x_{h1k}$ – número de unidades del elemento i en el nivel j producidas en las k primeras etapas.

Por último, las variables, w_j $j = 1, 2, 3, 4$. son utilizadas para ajustar la importancia relativa de cada nivel en su contribución al coste del secuenciamiento. En el caso de GCM, $w_1 = w_3 = w_4 = 0$; $w_2 = 1$.

El algoritmo se describe a continuación en forma de pseudocódigo:

- Seleccionar el número de elementos diferentes en el nivel 1, 2
- Inicializar los pesos diferentes para cada nivel
- Seleccionar la demanda para cada producto
- Seleccionar el número de unidades del elemento i en el nivel 2 necesarias para producir una unidad del producto l

$n1, n2$
 $w1 = w3 = w4 = 0, w2 = 1$
 $d(i, 1)$ con $i = 1, 2, \dots, n1$
 $t(i, 2, l)$

```

Calcular  $d(i, 2)$ 
Calcular  $DT1$ 
PARA  $k = 1$  HASTA  $DT1$  HACER
     $min\_temp = inf$ 
     $i\_min = 0$ 
    PARA  $i = 1$  HASTA  $n1$  HACER
        Calcular  $Delta(i, 2, k)$ 
         $GCM(i, k) = Delta(i, 2, k)$ 
        SI  $x(i, 1, k-1) < d(i, 1)$  ENTONCES
            SI  $GCM(i, k) < min\_temp$  ENTONCES
                 $i\_min = i$ 
                 $min\_temp = GCM(i, k)$ 
        FIN SI
    FIN SI
    FIN PARA
    PARA  $j = 1$  HASTA 2 HACER
        PARA  $i = 1$  HASTA  $n(j)$  HACER
            Actualizar  $x(i, j, k)$  (se ha elegido  $x(i\_min, 1, k)$ )
        FIN PARA
    FIN PARA
FIN PARA
    
```

3.3. Funciones de coste

Las funciones de coste se emplean para comparar los resultados generados por los métodos de secuenciamiento. Para problemas de naturaleza como los tratados en este estudio se han venido utilizando tradicionalmente diferentes expresiones relacionadas con los respectivos métodos de optimización como Miltenburg (Miltenburg y Sinnamon, 1989), Goal-chasing (Monden, 1983) e Inman (Inman y Bulfin, 1992).

Goal-chasing

$$GCM_{ik} = \sum_{l=1}^{n_2} [(x_{l,2,k-1} + t_{l,2,i}) - k d_{l,2} / DT_1]^2$$

Miltenburg

$$\sum_{k=1}^{DT_1} \sum_{j=1}^4 \sum_{i=1}^{n_j} w_j (x_{ijk} - XT_{jk} r_{ij})^2$$

Inman

$$\sum_{j=1}^J \sum_{i=1}^{n_j} \sum_{m=1}^{d_{ij}} w_j (c_{ijm} - \delta_{ijm})^2$$

Ccad

$$\sum_{j=1}^{n_2} (func(x_{j2k}, val_opt_j, acc_dis_j))$$

donde:

$$func(x_{j2k}, val_opt_j, acc_dis_j) = \min \left[0.1 \cdot \left(\frac{val_opt_j + acc_dis_j}{acc_dis_j} \right)^4 ; 1 \right]$$

Si $(x_{j2k} = 1) \text{ AND } (val_opt_j > 0)$
0 en caso contrario.

donde:

val_opt: vector de estimación de aparición de criterios.

acc_dis: vector de espaciamento aceptable de criterios.

La Tabla 4 muestra una comparativa del comportamiento de las funciones de coste anteriores respecto a características relevantes para el problema del secuenciamiento de sistemas de producción multi-modelo y multi-nivel. El estudio detallado de esta tabla y su génesis puede encontrarse en (Areal, 2011b).

Tabla 4. Vectores de pesos predeterminados

Función coste	Independencia origen tiempos secuencia	Seguimiento consumo constante en el tiempo	Igualdad tratamiento espaciamentos altos y bajos	Robustez penalización espaciamento unidad
Goal Chasing	Baja	Alta	Alta	Baja
Mitenburg	Media	Baja	Baja	Baja
Inman	Baja	Alta	Media	Media
c_cad	Alta	Media	Media	Baja

De la Tabla 4 se deduce que ninguna de las funciones de coste analizadas proporciona un comportamiento adecuado respecto a todas las características, por lo que surge la necesidad de definir un nuevo indicador que las satisfaga. En esta publicación se empleará la un nuevo indicador denominado *Cnesti* desarrollado por los autores (Areal et al., 2011a)(Areal, 2011b). Este se desarrolla a partir de valores estadísticos como la media y la desviación estándar que tiene en cuenta por igual a los diferentes criterios.

En primer lugar se define un indicador de coste para un solo criterio, dicho indicador deberá ser menor cuanto mejor sea el espaciamento.

La desviación estándar del espaciamento es una buena medida de la dispersión pero para poder compararla con la de otros criterios es necesario equipararla de algún modo. Para ello se procede a calcular el cociente entre la desviación estándar y la media aritmética de los valores de espaciamento. De esta forma se tiene una indicación de la dispersión de la muestra con relación a la media.

$$c_i = \frac{\sigma_{s_i}}{\bar{s}_i}$$

Volviendo al caso real que nos ocupa de equiespaciamento de criterios, para un mismo valor del ratio desviación estándar / media la repercusión en la línea de producción será distinta según el número de productos a los que afecte. Para tener en cuenta este efecto se añade un factor de multiplicación que representa el número de espaciamentos del criterio.

$$c'_i = (d_{i,2} - 1) \cdot \frac{\sigma_{s_i}}{\bar{s}_i}$$

Finalmente, se formaliza el indicador que tiene en cuenta a todos los criterios:

$$C_{nesti} = \sum_{i=1}^{n_2} \left((d_{i,2} - 1) \cdot \frac{\sigma_{s_i}}{\bar{s}_i} \right)$$

donde,

n_2 : número de opciones diferentes.

$d_{i,2}$: demanda de producción para la opción i.

σ_{s_i} : desviación estándar de la distancia de la opción i.

\bar{s}_i : media aritmética de la distancia de la opción i.

Esta función presenta un buen comportamiento frente a características importantes de una función de coste como son: la independencia del origen de tiempos de la secuencia, el seguimiento del consumo constante de cada elemento de producción, la independencia del valor de distancia de cada opción y la robustez para las opciones con distancia unidad (Areal et al., 2011b).

Cuando se aplica la función de coste a la lista mezclada de forma aleatoria utilizada en este estudio, se obtiene un coste *Cnesti* de valor 202,86 (frente a un coste *Cnesti* 0 si se calculase con las secuencias ideales antes de desordenar).

4. Mejoras del método Goal-Chasing: introducción de pesos diferentes para cada opción.

En el modelo del método GCM enunciado anteriormente es posible diferenciar la importancia relativa de cada nivel en su contribución al coste del secuenciamiento total. Sin embargo, no se enuncia ningún mecanismo para diferenciar la importancia relativa de cada opción dentro del nivel 2 en su contribución al coste del secuenciamiento total. La razón de ser de este nuevo vector variable de pesos diferentes para cada opción reside en la necesidad de priorizar la cadencia de algunas opciones frente a otras. Dándole más peso a algunas de ellas es posible obtener un mejor resultado para las mismas, pero hay que tener en cuenta la degradación generada en las otras.

A continuación se reproduce la expresión para el método GCM modificado con la introducción del vector de pesos de opciones *pen_opt(l)*:

$$\sum_{l=1}^{n_2} pen_opt(l) \cdot [(x_{1,2,k-1} + t_{1,2,i}) - k d_{1,2} / DT_1]^2$$

Resta ahora la tarea de definir el modo de asignación de valores numéricos a cada uno de los pesos para cada problema concreto. Se proponen a continuación cuatro métodos para el cálculo de estos vectores:

En primer lugar y como referencia se mantendrá el vector de pesos de valor unidad. En segundo lugar se empleará un vector de pesos de opciones donde cada componente tomará el valor inverso al espaciamiento aceptado de la opción.

$$pen_opt(l) = 1 / acc_dis(l)$$

En tercer lugar se empleará un vector de pesos de opción donde los coeficientes sólo pueden tomar tres valores: 1, x y $2x$. La asignación de pesos de opciones mayores se realiza inversamente al espaciamiento aceptable de opción. Con este vector se crean tres tipos de opciones:

- Una opción principal que es única y cuyo peso ($2x$) es igual a la suma de los pesos de las dos opciones importantes.
- Dos opciones importantes que presentan igual peso (x) y de valor mayor que la suma del peso de todas las opciones de base.
- Las opciones de base formadas por el resto de opciones, todas de igual peso y valor unidad.
- En cuarto lugar se empleará un vector de pesos de opciones donde los coeficientes toman valores decrecientes. La asignación de pesos de opciones mayores se realiza inversamente al espaciamiento aceptable de opción. Con este vector se crean dos tipos de opciones:
 - Una opción principal que es única y cuyo peso (igual al número de opciones) es mayor que la suma del peso del resto de opciones.
 - Las opciones secundarias que van tomando valores decrecientes progresivamente a partir del valor unidad. Para evitar que exista una sola opción con valor mínimo, las dos últimas opciones tomarán el mismo valor de peso.

En la Tabla 5 se muestran los cuatro vectores de pesos predeterminados para el caso del ordenamiento de la lista de 100 vehículos.

Tabla 5. Vectores de pesos predeterminados

Opción	1	2	3	4	5	6	7	8	9	10
Distancia mínima	3	4	9	8	19	7	6	2	2	1
Pen_opt 1	1	1	1	1	1	1	1	1	1	1
Pen_opt 2	.33	.25	.11	.12	.05	.14	.16	.5	.5	1
Pen_opt 3	1	1	1	1	1	1	1	8	8	16
Pen_opt 4	.9	.85	.65	.7	.65	.75	.8	.95	1	10

La Tabla 6 muestra los resultados obtenidos si se aplica el método Goal-chasing con cada uno de los cuatro vectores de pesos predeterminados para secuenciar la lista inicial de 100 vehículos desordenada de forma aleatoria. El empleo de Goal-chasing con pesos de opciones unidad presenta un coste de 63,25 que representa aproximadamente un 31,2% del coste *Cnesti* de la secuencia desordenada.

Tabla 6. Resultados de coste para la lista desordenada método Goal-chasing con pesos de opciones diferentes

	Pesos de opciones	Coste <i>Cnesti</i>
Lista Ideal		0.00
Lista desordenada		202.86
Lista ordenada (GCM)	[1 1 1 1 1 1 1 1 1 1]	63.25
Lista ordenada (GCM)	1/acc_dis	59.65
Lista ordenada (GCM)	[1 1 1 1 1 1 1 8 8 16]	61.68
Lista ordenada (GCM)	[.9 .85 .65 .7 .65 .75 .8 .95 1 10]	61.17

Los otros tres vectores propuestos mejoran ligeramente el resultado obtenido anteriormente, alcanzando un 29,4% para el caso del vector de pesos de opciones donde los coeficientes toman el valor inverso al espaciamiento aceptable de la opción.

En este artículo se empleará el método GCM con dos niveles modificado para impedir el secuenciamiento de un producto determinado una vez que su demanda ha sido satisfecha. Al aplicar este método sobre la lista desordenada de 100 vehículos, el valor de la función de coste *Cnesti* para la secuencia resultante es de 63,25.

5. Goal-chasing con optimización Nelder Mead

Después de haberse constatado la mejoría de los métodos de generación de la secuencia proporcionada por la introducción de pesos diferentes para cada opción, surge la pregunta de si se estarán empleando los vectores de pesos predeterminados realmente adecuados. En este apartado se propone una optimización progresiva del vector de pesos de opciones definido inicialmente a través de un proceso iterativo de búsqueda de un mínimo de la función de coste.

En este artículo se ha empleado el algoritmo de Nelder-Mead (Nelder y Mead, 1965) que es uno de los métodos más ampliamente utilizado para la optimización no lineal sin restricciones.

5.1. Método Nelder Mead

El método de Nelder-Mead minimiza una función no lineal escalar de n variables reales usando solamente valores de la función, sin ningún tipo de información de derivadas. Este método se encuentra dentro de la clase general de métodos de búsqueda directa. Un amplio subconjunto de los métodos de búsqueda directa, incluyendo el Nelder-Mead, mantiene en cada etapa del algoritmo un “simplex” o figura geométrica n -dimensional delimitada por $n + 1$ vértices.

Cada iteración de un método de búsqueda directa basado en un simplex comienza con un simplex especificado por sus $n+1$ vértices y sus valores de función asociados. Posteriormente se calculan uno o más puntos de test junto con sus valores de función, y la iteración termina con un nuevo conjunto de $n+1$

vértices y sus valores de función asociados obtenidos a partir de los puntos de test calculados.

Cuatro parámetros escalares son necesarios para definir un método Nelder-Mead: coeficiente de reflexión (ρ), de expansión (χ), de contracción (γ) y de “shrink” (σ). En este estudio se han empleado los valores utilizados universalmente (Lagarias et al, 1998):

$$\rho = 1, \quad \chi = 2, \quad \gamma = 1/2, \quad \sigma = 1/2$$

La operación Shrink consiste en calcular los nuevos n vértices v_i que junto con el actual x_1 formarán el nuevo simplex.

$$v_i = x_1 + \sigma \cdot (x_i - x_1) \quad , \quad i = 2, 3, \dots, n + 1$$

El algoritmo se ejecuta hasta alcanzar un número determinado de iteraciones, o bien hasta que los componentes de los vértices difieren menos que una tolerancia fijada de los componentes del mejor vértice, o bien hasta que los valores de las funciones para todos los vértices difieren menos que una tolerancia fijada del valor de la función para el mejor vértice.

5.2. Adaptaciones al método Nelder-Mead

Una de las particularizaciones realizadas al método Nelder-Mead en el caso de estudio consiste en la forma de calcular el simplex inicial a partir del vector de pesos de criterios. El cálculo del simplex inicial suele realizarse tomando un vértice donde todos sus componentes están muy próximos al vector inicial y los n vértices restantes se obtienen modificando solamente un componente del vector inicial:

$$v_1 = a \cdot v_{in}$$

$$v_j = (v_{in}(1), v_{in}(2), \dots, b \cdot v_{in}(j-1), \dots, v_{in}(n)),$$

$$j = 2, 3, \dots, n + 1$$

Tras diversos estudios para obtener valores de los coeficientes a y b apropiados (Areal, 2011b), se han retenido los valores siguientes: $a=0.9$ y $b=3$. Habitualmente se emplean los valores $a=0.9$ y $b=1.1$, pero un aumento del valor del coeficiente b facilita la optimización para los valores iniciales de los vectores de pesos de opciones del problema objeto de estudio.

Otra modificación realizada consiste en la prohibición de que los vértices obtenidos a lo largo de todo el algoritmo tomen valores negativos. El vector final resultante del proceso de optimización no sería válido dado que la idea de componente negativo trae asociada la disminución de la función de coste cuanto peor fuera la forma de cadenciar esa opción, comportamiento no deseado en el problema del secuenciamiento de productos.

Al aplicar el método Goal-chasing para disminuir el coste de secuenciamiento de la lista inicial de 100 vehículos desordenada de forma aleatoria, con cada uno de los cuatro vectores de pesos predeterminados optimizados con el método Nelder-Mead, se obtienen los resultados que se muestran en la Tabla 7.

Tabla 7. Resultados de coste para la lista desordenada método Goal-chasing con optimización Nelder-Mead

	Pesos de opciones	Coste Cnesti
Lista Ideal		0.00
Lista desordenada		202.86
Lista ordenada (GCM-NM)	[1 1 1 1 1 1 1 1 1 1]	58.76
Lista ordenada (GCM-NM)	1/acc_dis	56.17
Lista ordenada (GCM-NM)	[1 1 1 1 1 1 1 8 8 16]	57.79
Lista ordenada (GCM-NM)	[.9 .85 .65 .7 .65 .75 .8 .95 1 10]	56.17

De la Tabla 7 se infiere que el empleo de Goal-chasing optimizado con el método Nelder-Mead con pesos de opciones unidad presenta un coste de 58,76 que representa aproximadamente un 29,0% del coste C_{nesti} de la secuencia desordenada. La utilización de los pesos de opciones variables a partir de los otros tres vectores de opciones predeterminados mejora ligeramente el resultado obtenido anteriormente, alcanzando un 27,7% para el caso del vector de pesos de opciones con coeficientes inversamente proporcionales al espaciamiento aceptable y el caso de los coeficientes que toman valores decrecientes.

La Figura 4 muestra el gráfico de la distancia inicial de la opción 2 junto con las distancias obtenidas para la misma opción al aplicar Goal-chasing y Goal-chasing con optimización Nelder-Mead. Puede verse que las secuencias obtenidas tras aplicar los métodos de ordenamiento han reducido la dispersión de la distancia instantánea de la secuencia inicial, siendo Goal-chasing con optimización Nelder-Mead el que más se acerca a la distancia mínima para esta opción.

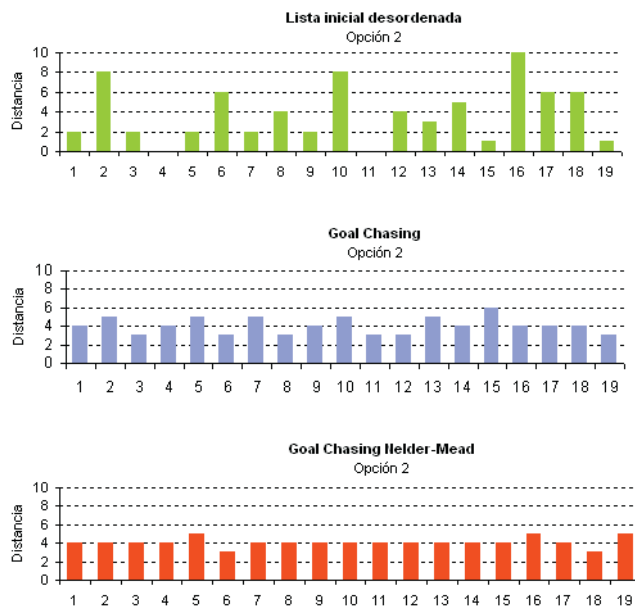


Figura 4. Distancia de la opción 2: lista inicial desordenada, Goal-chasing y Goal-chasing con optimización Nelder-Mead.

6. Generalización de resultados con datos representativos

Teniendo en cuenta que los resultados previos están basados en una única lista de referencia, las conclusiones deben ser generalizadas. Para ello, los cálculos se han extendido a un conjunto más representativo de 30 secuencias. Una lista ideal de 100 vehículos ha sido el punto de partida, cada uno de ellos con 10 opciones, la mayoría de ellas con distancias mínimas entre 1 y 9. El coste de esta lista ideal es nulo. Nuevamente, la lista de entrada para el algoritmo (secuencia inicial), es la resultante de haber desordenado la lista ideal de forma aleatoria.

Para generalizar los resultados y que éstos sean más estadísticamente significativos, se generan 30 secuencias de vehículos. Las longitudes de las secuencias generadas estarán en el intervalo [150, 250], siendo este rango representativo de la capacidad de acceso directo (plazas del almacén de carrocerías y primera posición de cada barra de clasificación FIFO) existente en las plantas de automóviles para determinar la siguiente carrocería en entrar en el taller de montaje. El número de opciones utilizadas para el ordenamiento de cada secuencia se moverá en el intervalo [10, 15], siendo estos valores ambiciosos en el contexto actual de la fabricación de automóviles. Finalmente, las distancias mínimas para cada opción estarán calculadas en el intervalo [1, 10], correspondiente a los valores más comúnmente obtenidos en la práctica.

En el mundo industrial de la fabricación de automóviles no se puede asegurar que para un horizonte de producción dado exista un ordenamiento de vehículos de coste cero. Con el fin de obtener una mayor representatividad del entorno industrial, en lugar de construir 30 listas iniciales con coste nulo y posteriormente desordenar de forma aleatoria los vehículos dentro de cada lista, cada una de las 30 secuencias generadas tendrá las opciones distribuidas de forma aleatoria. Este método de generación de listas iniciales es más severo que el empleado para la lista inicial de 100 vehículos y no permite conocer el valor de coste mínimo alcanzable para cada secuencia.

La Tabla 8 muestra la media de los porcentajes de coste *Cnesti* (con relación a los valores de coste de las secuencias iniciales) de las secuencias obtenidas después de aplicar los métodos de ordenamiento estudiados.

Estos resultados empleando 30 secuencias iniciales más representativas de la situación industrial son coherentes con los obtenidos al secuenciar la lista inicial de 100 vehículos desordenada de forma aleatoria.

Tabla 8. Resultados de coste de 30 secuencias generadas método Goal-chasing sin y con optimización Nelder-Mead

	Pesos de opciones	Coste <i>Cnesti</i>
30 Listas desordenadas		100
Lista ordenada (GCM)	[1 1 1 1 1 1 1 1 1 1]	34.88
Lista ordenada (GCM)	1/acc_dis	34.53
Lista ordenada (GCM)	[1 1 1 1 1 1 1 8 8 16]	33.97
Lista ordenada (GCM)	[.9 .85 .65 .7 .65 .75 .8 .95 1 10]	31.23
Lista ordenada (GCM-NM)	[1 1 1 1 1 1 1 1 1 1]	30.78
Lista ordenada (GCM-NM)	1/acc_dis	28.85
Lista ordenada (GCM-NM)	[1 1 1 1 1 1 1 8 8 16]	27.98
Lista ordenada (GCM-NM)	[.9 .85 .65 .7 .65 .75 .8 .95 1 10]	27.69

La Figura 5 presenta gráficamente la media de los porcentajes de coste *Cnesti* para las secuencias obtenidas (con relación a los valores de coste de las secuencias iniciales) al aplicar el método Goal-chasing y la optimización Nelder-Mead sobre Goal-chasing.

La utilización de vectores de pesos de opciones en el método Goal-chasing produce mejores resultados que el empleo de los valores unidad. La optimización Nelder-Mead de los pesos de opciones de GCM incrementa aún más la mejora del secuenciamiento. Los valores obtenidos con Nelder-Mead empleando cualquiera de los cuatro vectores predeterminados de pesos de opciones son mejores que el mejor resultado logrado con vector predeterminado sin optimización (31,23%).

Adicionalmente, el método Goal-chasing, con y sin optimización Nelder-Mead, ha sido contrastado con varias secuencias de referencia clásicas de la literatura, concretamente para resolver secuencias de la librería CSPLib de problemas de optimización sujetas a restricciones (Gent y Walsh, 1999). La Tabla 9 muestra los resultados obtenidos.

Los resultados para los dos problemas con secuencias no resolubles (ET1 6/76 y ET3_200_10) para la función de coste utilizada en CSPLib son consistentes con los mostrados anteriormente en la Figura 5, dando valores de *Cnesti* entre 34-37% con Goal-chasing y reduciéndose este coste un 3-4% adicional cuando se emplea también la optimización Nelder-Mead.

Hay que señalar que la reducción adicional de coste fue inferior en el problema de la secuencia resoluble (ET2 75-05), donde una reducción de coste más importante de partida unida al número reducido de opciones de la secuencia (cinco opciones) no han permitido obtener una mejora mayor al emplear Nelder-Mead.

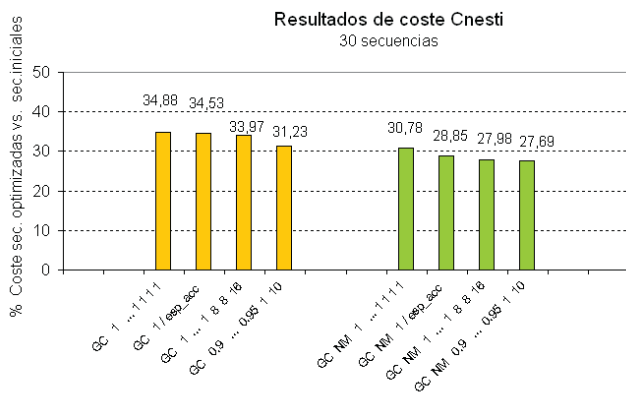


Figura 5. Resultados de coste de 30 secuencias generadas con método Goal-chasing sin y con optimización Nelder-Mead.

Tabla 9. Resultados de optimización de problemas CSPLib usando Goal-chasing y Goal-chasing con optimización Nelder-Mead.

Problema	Longitud de secuencia	Coste Ideal Cnesti	% Coste Cnesti Goal Chasing	% Coste Cnesti GH NM
ET1 6/76	100	227.69	33.92%	29.96%
ET2 75-05	200	381.61	28.04%	26.00%
ET23 200_10	200	503.87	37.62%	34.90%

7. Resultados con datos reales

En este apartado se utilizarán listas de secuenciamiento de producción reales procedentes de PSA Peugeot-Citroën Centro de Vigo (España). En concreto se tomarán valores de la producción realizada en la tercera semana de septiembre 2008. Se dispone de una lista de secuenciamiento para el Sistema de Montaje existente en la fábrica (Sistema 1). Para el Sistema 1 se trata de una lista ordenada compuesta por 5491 vehículos, cada uno de los cuales posee 10 criterios de ordenamiento montaje.

Para darle una mayor representatividad a nivel de cantidad de vehículos al estudio, se han tomado 30 listas de 180 vehículos para el caso del sistema 1. Se corresponden con los primeros 5400 vehículos ordenados de la lista de secuenciamiento real.

Los cálculos realizados a partir de los datos reales de ordenamiento de 5400 vehículos del Sistema 1 de montaje de PSA Centro de Vigo se encuentran resumidos en la Figura 7, mostrándose los valores medios obtenidos para la función de coste *Cnesti* al emplear el métodos de ordenamiento Goal Chasing con los cuatro vectores de penalización de criterios predeterminados y tras aplicar el algoritmo de Nelder-Mead para optimizar los componentes de los vectores de penalización de criterios predeterminados (Este análisis puede consultarse detalladamente en Areal, 2011b).

De la Figura 6 se deduce que los resultados obtenidos a partir de secuencias reales corroboran los resultados obtenidos a partir de datos generalizados (Figura 5).

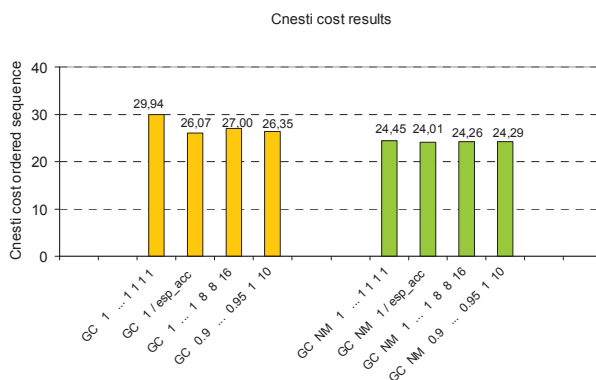


Figura 6. Resultados de coste de 30 sobre secuencias generadas con método Goal-chasing sin y con optimización Nelder-Mead partiendo de 30 secuencias reales.

Nuevamente, la optimización Nelder-Mead de los pesos de opciones de GCM incrementa aún más la mejora del secuenciamiento, y los valores obtenidos con Nelder-Mead empleando cualquiera de los cuatro vectores predeterminados de pesos de opciones son mejores que el mejor resultado logrado con vector predeterminado sin optimización (31,23%). En cuanto a que los valores de los coste sean mejores en el caso de las secuencias reales (Figura 6) que en el caso de las secuencias generalizadas (Figura 5) tiene su explicación en que las segundas se han hecho en condiciones más estrictas: listas de 250 vehículos frente a listas de 180, y número de opciones en el intervalo [10, 15] frente a 10 opciones.

8. Conclusiones y trabajo futuros

En este trabajo se ha presentado la utilización de pesos de opciones diferentes para mejorar el secuenciamiento obtenido por un algoritmo Greedy (método Goal-chasing). Se han introducido cuatro métodos diferentes de cálculo de pesos de opciones y los resultados de su empleo han sido mostrados para el caso concreto del mundo del automóvil con una secuencia de partida de 100 vehículos.

Se ha introducido el método Nelder-Mead de optimización no lineal y se ha modificado ligeramente para adaptarlo al problema de secuenciamiento de los sistemas de producción multi-producto y multi-nivel para optimizar el cálculo de los pesos de opciones. Los resultados logrados con estos pesos optimizados han mejorado los conseguidos con los pesos de opciones calculados anteriormente.

Los resultados obtenidos a partir de una única secuencia de partida se han generalizado utilizando 30 secuencias representativas del entorno industrial y se han contrastado frente a algunas secuencias de referencia clásicas de la literatura.

A la vista de los resultados obtenidos:

- El método Goal-chasing mejora de forma importante el ordenamiento de las secuencias iniciales desordenadas. En media presenta una reducción hasta el 34,9% del valor de coste inicial para la función de coste utilizada.
- El método Goal-chasing asociado al empleo de vectores de pesos de opciones mejora el Goal-chasing clásico. En media presenta una reducción suplementaria que alcanza hasta el 3,7% del valor de coste inicial.
- El método Goal-chasing asociado a la optimización Nelder-Mead de los coeficientes de pesos de opciones mejora los valores de coste anteriores. En media presenta una reducción suplementaria con relación al Goal-chasing clásico que alcanza hasta el 7,2% del valor de coste inicial.

Una futura línea de trabajo es la aplicación de la optimización Nelder-Mead con otros algoritmos Greedy no tan destacados como el GCM en lo que respecta al equilibrado de las opciones y para los que el empleo de los pesos de opciones diferentes proporcionaría a priori una mejora mayor. Otra vía a explorar es la utilización de otros métodos de optimización no lineal de los pesos de opciones que permitirían contrastar los resultados ya presentados en este trabajo utilizando el método Nelder-Mead.

English Summary

Greedy Heuristics improvements for sequencing Multi-model and Multi-level Production.

Abstract

The Car Sequencing Problem consists in maintaining a certain order in the vehicles as they pass through the assembly line. Sequences have to be built according to each vehicle's options, each one requiring different resources and production time, with the objective of avoiding to exceed the maximum human and facility potential. In this paper, we use a Greedy heuristic, the Goal-chasing method developed by Toyota, to solve the Car Sequencing Problem. The concept of different weights for each option is introduced to improve the ordering of the sequence. Nelder-Mead method of nonlinear optimization is applied to obtain the weights of options that minimize the cost of the resulting sequence. The results obtained for a initial sequence are expanded to a set of 30 representative sequences of the automotive industry and they are contrasted with some of the classical benchmarks from the literature. Finally, real data is used to validate the proposal.

Keywords:

Heuristics, sequence estimation, optimization problems, automobile industry

Referencias

- Areal, J., Marín, R., Garrido, J., 2011a. Production Line balancing with Genetic Algorithms, *International Journal of Manufacturing Technology Management*, 23 (1/2), 113-136.
- Areal, J., 2011b. Resolución del problema del secuenciamiento de vehículos en un entorno industrial. Tesis Doctoral. Universidad de Vigo, España.
- Briant, O., Naddef, D., Mounié, G., 2008. Greedy approach and multi-criteria simulated annealing for the car sequencing problem, *European Journal of Operational Research*, 191(3), 993-1003.
- Chew, T., David, J., Nguyen, A., Tourbier, Y., 1992. Solving constraint satisfaction problems with simulated annealing: The car sequencing problem revisited. *12th Int'l Workshop on Expert Systems and their Applications*, 405–416, Avignon, France.
- Cordeau, J.F., Laporte, G. and Pasin, F., 2008. An iterated local search heuristic for the logistics network design problem with single assignment, *International Journal of Production Economics*, 113, 626-640.
- Delaval, M., Ohl, H., Gentina, J.C., Yvars, P.A., 1995. Algorithmes de cadencement reactif des vehicules en entrée de ligne de montage automobile, *Revue d'automatique et de productique appliquees*, 8 (5), 663-682.
- Diaz, A., Glover F., Ghaziri, H., Gonzalez, J., Laguna, M., Moscato, P., Tseng, F., 1996. *Optimizacion Heuristica y Redes Neuronales*, Editorial Paraninfo.
- Downsland, K.A., 1993. *Simulated Annealing*, Modern Heuristic Techniques for Combinatorial Problems, C.R. Reeves, Blackwell Scientific Pub., Oxford.
- Drexl, A., Kimms, A., Matthiessen, L., 2006. Algorithms for the car sequencing and the level scheduling problem. *Journal of Scheduling*, 9 (2), 153-176.
- Estellon, B., Gardi, F., Nouioua, K., 2005. Ordonnement de véhicules: une approche par recherche locale à grand voisinage. In: *Actes de Premières Journées Francophones de Programmation par Contraintes*. Lens, France, 21-28.
- Gent, I. P., Walsh, T., 1999. CSPLib: a benchmark library for constraints. *Technical report APES-09-1999*. Available from <http://csplib.cs.strath.ac.uk/>. A shorter version appears in CP99.
- Goldberg, D.E., 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*, Reading, Mass.: Addison-Wesley Publishing Co.
- Gottlieb, J., Puchta, M., Solnon, C., 2003. A study of Greedy, local search and ant colony optimization approaches for car sequencing problems, *Lecture Notes in Computer Science*, 2611, 246-257.
- Kirkpatrick,S., Gelatt, C.D., Vecchi, M.P., 1983. Optimization by Simulated Annealing. *Science*, New Series, 220 (4598), 671-680.
- Kis, T., 2004. On the complexity of the car sequencing problem. *Operations Research Letters*, 32(4), 331–335.
- Inman, R., Bulfin, R., 1992. Quick and dirty sequencing for mixed-model multi-level JIT systems". *International Journal of Production Research*, 30 (9), 2011-2018.
- Lagarias, J., Reeds, J., Wright, M., Wright, P., 1998. Convergence properties of the Nelder-Mead simplex method in low dimensions. *Society for Industrial and Applied Mathematics Journal of Optimisation*, 9 (1), 112-147.
- Miltenburg, J., Sinnamon,G., 1989. Scheduling mixed-model multi-level just-in-time production systems. *Internacional Journal of Production Research*, 27 (9), 1487-1509.
- Monden, Y., 1983. *Toyota Production System*, Institute of Industrial Engineers Press, Norcross, Georgia.
- Nelder, J.A. and Mead, R., 1965. A simplex method for function minimization, *Computer Journal*, 7, 308-313.
- Parrello, B., 1988. Car Wars: The (almost) birth of an expert system, *AI Expert*, 60-64.
- Ponnambalam, S.G., Aravindan, P., Rao, M.S., 2003. Genetic algorithms for sequencing problems in mixed model assembly lines. *Computers and Industrial Engineering*, 45 (4), 669–690.
- Solnon,C., Cung V. D., Nguyen,A. and Artigues,C., 2008. The car sequencing problem: Overview of state-of-the-art methods and industrial case-study of the ROADEF'2005 challenge problem. *European Journal of Operational Research*, 191 (3), 912–927.
- Warwick, T., Tsang, E., 1995. Tackling car sequencing problems using a genetic algorithm. *Journal of Evolutionary Computation*, 3(3), 267–298.
- Xiaobo, Z., Zhou, Z., 1999. Algorithms for Toyota's goal of sequencing mixed models on an assembly line with multiple stations. *Journal of the Operational Research Society*, 50(7), 704–710.
- Zinflou, A., Gagné, C., Gravel, M., 2007. Crossover Operators for the Car Sequencing Problem, *Evolutionary Computation in Combinatorial Optimization*, 4446, 229-239.