

Procedimientos heurísticos para la determinación de secuencias en una máquina multiproducto sujeta a fallos y con costes cuadráticos.

Albert Corominas¹, Rafael Pastor¹, Ana Sánchez¹

¹ Instituto de Organización y Control de Sistemas Industriales. ETS de Ingeniería Industrial de Barcelona. Universidad Politécnica de Cataluña. albert.corominas@upc.edu, rafael.pastor@upc.edu, ana.sanchez@upc.edu.

Resumen

Se considera una máquina capaz de producir distintos tipos de pieza de forma simultánea. Dicha máquina está sujeta a fallos (en cuyo caso permanece temporalmente parada) y se desea minimizar los costes cuadráticos de almacenamiento y de carencia. En la literatura se han propuesto métodos exactos, basados en políticas PHP (prioritized hedging point), para la resolución del problema; pero presentan el inconveniente de que los tiempos de resolución aumentan de forma exponencial con el número de piezas. En este artículo se exponen procedimientos heurísticos y se muestran los resultados obtenidos, comparándolos con las soluciones óptimas.

Palabras clave: *Prioritized Hedging Points*, heurísticas, 1-máquina.

1. Introducción: definición del problema.

Se considera una máquina capaz de producir, de forma simultánea, n productos o tipos de pieza distintos y con unas tasas de demanda conocidas ($d_j; j = 1, \dots, n$). La máquina está sujeta a fallos, por lo que puede adoptar dos estados, funcionamiento o paro, cuyos valores siguen distribuciones exponenciales de parámetros conocidos ($1/q_F$ y $1/q_P$, respectivamente). Otros datos conocidos son la tasa máxima de producción de la máquina (μ , que es igual para todos los productos) y la particularidad que la máquina dispone de capacidad suficiente para satisfacer todas las demandas. El objetivo es minimizar la esperanza matemática de los costes de almacenamiento y de carencia, por unidad de tiempo y en régimen permanente. Estos costes son cuadráticos respecto a la cantidad de piezas almacenadas y se ponderan con un coeficiente c_j , que, en adelante, se denominará coste.

El problema se formaliza en Shu y Perkins (2001), como una extensión de trabajos previos. En dicho artículo, los autores se centran en las políticas de tipo PHP (*prioritized hedging points*) y establecen expresiones analíticas simples para determinar el valor óptimo del *hedging point* z_j^* (ecuación 1). Dichas políticas presentan el siguiente funcionamiento: siguiendo un orden de prioridad de las piezas previamente establecido, se va asignando toda la capacidad de la máquina a la pieza de mayor prioridad hasta que el stock de dicha pieza alcanza una determinada cota (que varía según el tipo de pieza) llamada *hedging point*; una vez alcanzada esta cota se mantiene un nivel de producción igual a la tasa de demanda de

dicho tipo de pieza para mantener constante el nivel de stock y el resto de capacidad de la máquina se asigna a la pieza de siguiente prioridad; y así sucesivamente. El coste total de almacenamiento y carencia es la suma de los costes parciales de almacenamiento (ecuación 2).

$$z_j^* = \frac{1-\gamma_j}{\lambda_j} - \frac{1-\gamma_{j-1}}{\lambda_{j-1}} \quad \text{con} \quad \lambda_j = \frac{q_u}{D_j} - \frac{q_d}{\mu - D_j}; \quad \gamma_j = \frac{\mu q_u - (q_d + q_u) D_j}{(\mu - D_j)(q_d + q_u)} \quad \text{y} \quad D_j = \sum_{k=1}^j d_k \quad (1)$$

$$J_j^* = C_j - c_j (z_j^*)^2 \quad \text{con} \quad C_j = 2c_j z_j^* \left[\frac{1}{\lambda_j} - \frac{(1-\gamma_{j-1})\gamma_j}{(1-\gamma_j)\gamma_{j-1}\lambda_{j-1}} \right] \quad (2)$$

En la Figura 1 se puede observar el comportamiento del tipo de políticas PHP, dado un orden de prioridad de las piezas A, B, C.

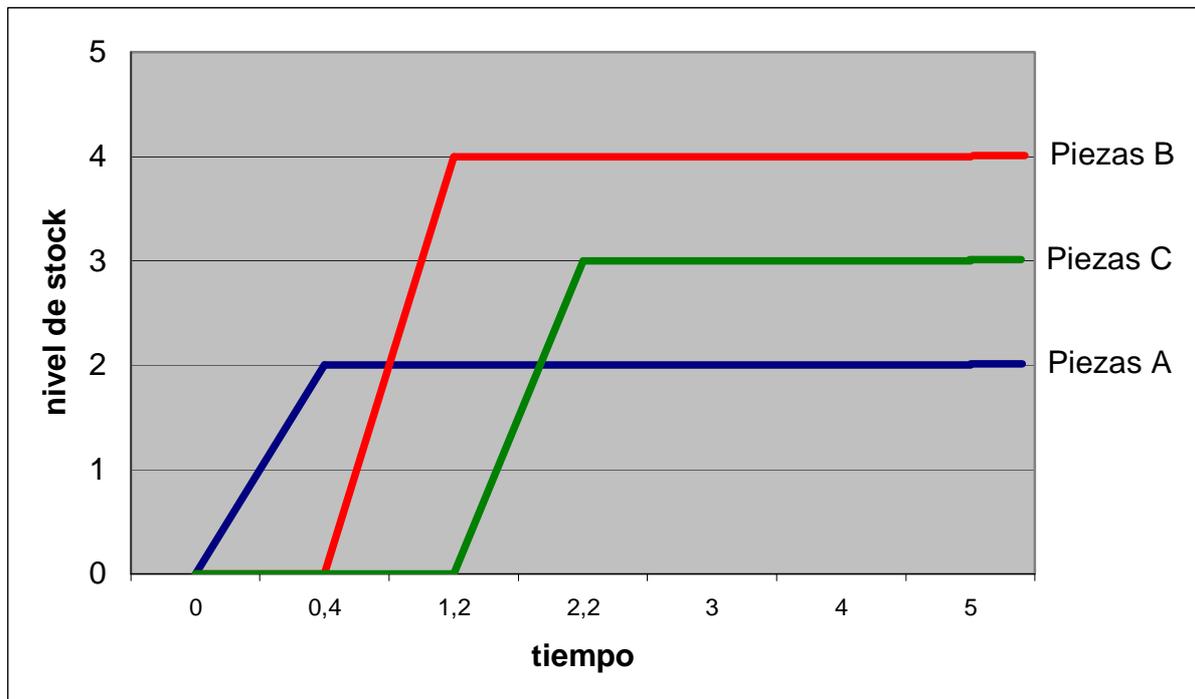


Figura 1. Ejemplo de política PHP.

En Corominas *et al.* (2004) se proponen dos procedimientos para determinar la secuencia óptima de prioridad de las piezas: se trata de una enumeración limitada, que puede asimilarse a la programación dinámica, en la cual se hace uso de unas determinadas relaciones de precedencia y de dominancia entre piezas, descritas en Shu y Perkins (2001). La única diferencia entre los dos procedimientos reside en el hecho que en uno de ellos se utilizan las relaciones de precedencia entre piezas (procedimiento A) y en el otro, además, se utilizan las relaciones de dominancia (procedimiento B). Asimismo, los autores realizan una amplia experiencia computacional en la que obtienen los resultados óptimos que muestran que, para un mismo procedimiento, los tiempos de cálculo son muy parecidos para un mismo valor de n , con un valor medio que crece exponencialmente (se multiplica por $2 \cdot n / (n-1)$, aproximadamente, para un incremento de n igual a 1, como es previsible a partir del análisis del algoritmo). Además, los tiempos de resolución del procedimiento B son menores que los del procedimiento A aunque presentan una mayor variabilidad.

El crecimiento exponencial del tiempo de cálculo pone de manifiesto la necesidad, para valores altos de n , de procedimientos de resolución heurísticos para lograr unos tiempos de resolución menores.

Con el objetivo de poder comparar posteriormente los resultados de los procedimientos heurísticos desarrollados con los resultados óptimos, se ha vuelto a realizar la experiencia computacional que se presenta en Corominas et al. 2004. Como resultado se confirman los resultados presentados en dicho trabajo: dado un procedimiento, los tiempos de cálculo son muy similares, para un mismo número de tipos de producto, n , y la media de dicho tiempo de cálculo crece exponencialmente (como estaba previsto) con dicho valor n ; por otro lado, los tiempos de cálculo se han reducido en un factor próximo a 10 debido al cambio del lenguaje de programación. En la Figura 2 se muestra la representación gráfica de la evolución de los nuevos tiempos de resolución para ambos procedimientos.

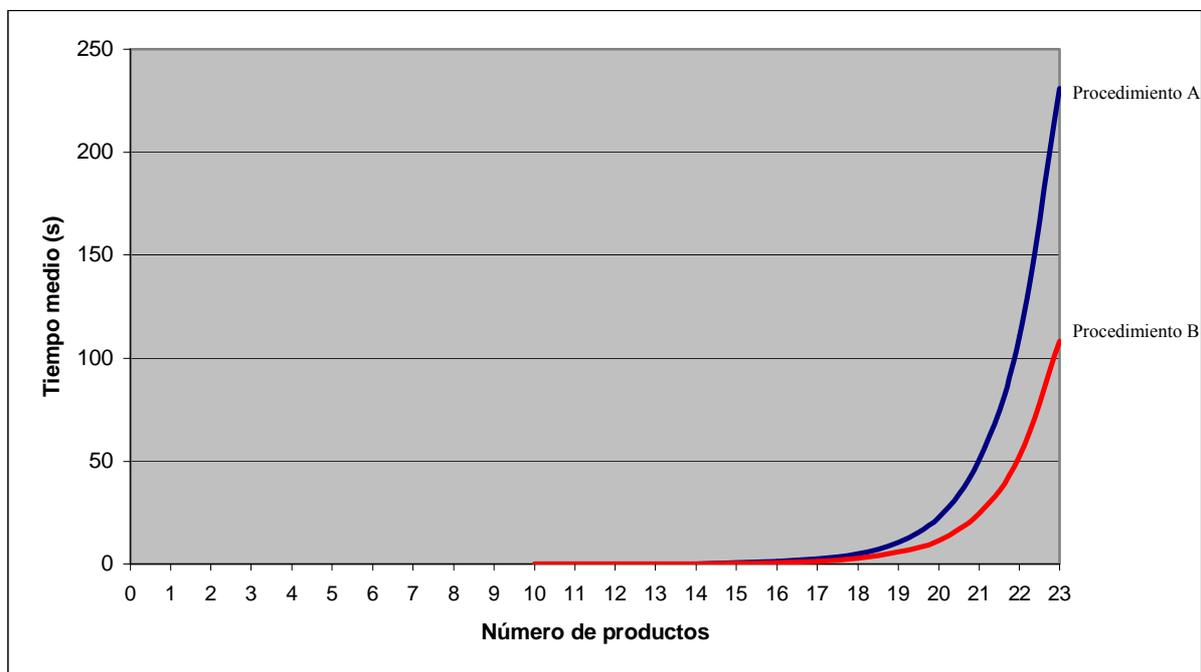


Figura 2. Evolución de los tiempos de resolución en función del número de productos.

2. Heurísticas

En el caso de los procedimientos heurísticos, aunque su ejecución no garantiza la consecución de las soluciones óptimas, es de prever que se acorten notablemente los tiempos de resolución (especialmente en los casos en que el número de productos distintos es elevado).

Los procedimientos heurísticos desarrollados consisten, básicamente, en establecer el orden de prioridad de las piezas mediante diferentes criterios. A continuación, en cada heurística, utilizando las expresiones propuestas por Shu y Perkins, se calculan y se guardan los resultados y los tiempos de resolución.

Los criterios propuestos para la determinación del orden de prioridad de las piezas proceden de un análisis exhaustivo de los factores que influyen en el coste de almacenamiento y carencia (según la expresión propuesta por Shu & Perkins).

Las heurísticas desarrolladas se basan en ordenar las piezas según los siguientes criterios:

- H1: por demanda decreciente (d_j).
- H2: por coste decreciente (c_j).
- H3: por producto “coste por demanda” decreciente ($d_j \cdot c_j$).
- H4: por coste parcial de almacenamiento (de cada pieza) menor (J_j). El coste parcial de almacenamiento J_j es el coste asociado a las piezas ya secuenciadas hasta el momento, incorporando al final de la secuencia parcial la pieza que está siendo evaluada (el resto no son consideradas).
- H5: por cociente “demanda / coste” decreciente (d_j/c_j).
- H6: por número de precedencias inmediatas (resultado de las relaciones de precedencia descritas en Shu y Perkins, 2001) decreciente.

Finalmente también se ha desarrollado la heurística H7, descrita a continuación:

- H7: calcular, en cada iteración, el coste parcial de almacenamiento para todas las parejas de piezas (considerando únicamente aquellas piezas que no han sido secuenciadas) y fijar la primera pieza de la pareja de menor coste; iterar hasta ordenar todas las piezas.

El disponer de la solución óptima de cada ejemplar permite evaluar la discrepancia media de cada heurística respecto a las soluciones óptimas (porcentualmente).

3. Experiencia computacional

Para evaluar los tiempos de computación conviene precisar que para la experiencia computacional se ha utilizado un Pentium IV a 3.2 GHz y 1 GB de memoria RAM, y que los procedimientos se han programado en Visual Fortran Professional Edition 6.6.0.

Para cada una de las 7 heurísticas se han resuelto los mismos 100 ejemplares de 14 valores de n distintos (con $n \in [10,23]$), en total 1400 ejemplares. En todos los experimentos, los tiempos de cálculo correspondientes a los distintos juegos de datos son muy pequeños (≤ 0.02 segundos) y no se observa ninguna tendencia clara en cuanto a su evolución (no se aprecia, en ningún caso, un aumento ligado al número de tipos de pieza distintos).

Para medir la calidad de las heurísticas, se han comparado las soluciones obtenidas con las soluciones óptimas; se puede observar que, claramente, algunas heurísticas se ajustan mucho mejor a las soluciones óptimas que otras (aunque todas ellas, en promedio, proporcionan cierta discrepancia respecto a las soluciones óptimas). Para medir dicho ajuste se calcula la discrepancia como la diferencia porcentual de las soluciones proporcionadas por las heurísticas respecto a los óptimos. Las discrepancias mínimas, medias y máximas respecto a las soluciones óptimas, para cada heurística, se muestran en la Tabla 1.

Tabla 1. Porcentaje de discrepancia mínima, media y máxima respecto a los óptimos.

	H1	H2	H3	H4	H5	H6	H7
D min (%)	0,0	0,1	0,0	6,8	10,3	0,0	3,2
D med (%)	138,7	18,6	1,8	886,1	342,6	6,6	636,5
D max (%)	732,9	109,6	43,9	2531,7	1364,6	62,7	2059,9

En promedio las heurísticas H3 y H6 proporcionan soluciones cercanas a las óptimas y mantienen una variabilidad no muy elevada; por otro lado, las soluciones proporcionadas por H4 y H7 presentan una discrepancia media muy grande y una gran variabilidad.

Además, también se ha podido observar que, dado un ejemplar de datos, la mejor solución entre las proporcionadas por las heurísticas, siempre proviene de H2, H3 o de H6; las soluciones proporcionadas por H1, H4, H5 y H7 tienen una discrepancia mayor en todos los casos. Por esta razón, se recomienda ejecutar H2, H3 y H6 y conservar la mejor solución de todas ellas; no obstante, y dado que los tiempos de cálculo son muy pequeños, si para una mayor seguridad se ejecutan las siete heurísticas, el incremento de tiempo es insignificante.

4. Perspectivas

Para disponer de heurísticas que presenten discrepancias reducidas, en la actualidad se está estudiando la posibilidad de calibrar heurísticas. Por otro lado, y como vías de posibles extensiones, se puede plantear la resolución del problema mediante el uso de cotas en un programa dinámico (de tipo *Branch & Bound*) o mediante metaheurísticas.

Referencias

Corominas, A.; Griñó, R.; Pastor, R. (2004). Determinación de secuencias óptimas en una máquina multiproducto sujeta a fallos y con costes cuadráticos. *XXVIII Congreso Nacional de Estadística e Investigación Operativa*, Cádiz.

Shu, C.; Perkins, J. (2001). Optimal PHP production of multiple part-types on a failure-prone machina with quadratic buffer costs. *IEEE Trans. Automat. Contr.*, 46 541-549.