

Programación n-cíclica en una línea de fabricación con puente-grúa

Manuel Mateo Doll, Ramon Companys Pascual

Dpto. de Organización de Empresas. Universidad Politécnica de Cataluña. Avda. Diagonal, 647, 7°. Barcelona 08028, Barcelona. manel.mateo@upc.edu, ramon.companys@upc.edu.

Resumen

La fabricación de grandes lotes de un mismo artículo en una línea de producción formada por tanques conlleva una programación cíclica para maximizar la producción. No obstante, queda por decidir cuál es el grado de ciclicidad óptimo, es decir el número óptimo de piezas que deben entrar y salir de la línea durante un ciclo. Para determinarlo se propone un modelo que considera tantas operaciones por tanque como piezas a tratar por ciclo. Dado que un puente-grúa, cuyos movimientos deben programarse, asegura la transferencia de piezas entre tanques, se trata de una variante del HSP (Hoist Scheduling Problem). El objetivo es determinar una secuencia que minimice el tiempo de ciclo. Para ello, hay una comparación entre los diversos grados de ciclos (problema n-cíclico). Se propone un procedimiento de branch-and-bound, con unas cotas apropiadas, que se aplica sobre unos juegos de datos. Los resultados de tiempos de ciclo obtenidos se analizan según los tipos de ventanas temporales y velocidades de grúa.

Palabras clave: Programación, *branch and bound*, *hoist scheduling problem*.

1. Introducción

Los sistemas modernos de fabricación a menudo comportan líneas de producción formadas por una secuencia de tanques. La manipulación de estos materiales se realiza por parte de puentes-grúa programados. Cuando una empresa afronta la programación cíclica de la producción, debido a la gran cantidad de unidades que forma cada lote, se puede plantear cuál es el grado óptimo de ciclo. Es decir, si compensa el hecho considerar más unidades dentro del ciclo de producción ya que aumenta la tasa de producción. Este trabajo puede incluirse en el marco del conocido problema *Cyclic Hoist Scheduling Problem* (CHSP).

Algunas de las características básicas del problema son:

- En cada uno de los tanques los productos reciben un cierto tratamiento. La fabricación se divide en una secuencia de operaciones, cada una realizada en un tanque ($i=1, \dots, m$).
- Cada operación del producto tiene asociada una ventana temporal. La ventana del tanque i tiene como límites un tiempo mínimo a_i y un tiempo máximo b_i .
- El transporte entre tanques se realiza exclusivamente mediante puente-grúa. El transporte con carga se realiza entre un tanque i y el siguiente, $i+1$. El transporte sin carga se puede realizar entre cualesquiera de los tanques (incluidas las estaciones de carga, tanque 0, y de descarga, tanque $m+1$).
- Si un puente-grúa llega antes del instante de recogida de un producto en un tanque, se produce un tiempo muerto.

Phillips y Unger (1976) desarrollaron el primer modelo para HSP. El problema de determinar la programación de operaciones del puente-grúa con el objetivo de optimizar la productividad es

NP-completo (Lei y Wang, 1989), incluso para la variante más simple de CHSP.

Algunos trabajos se han preocupado de analizar la influencia del número de grúas, como Lei y Wang (1991) que no permitían solape de movimientos de dos grúas. Cuando aparece un cuello de botella, es frecuente que más de un tanque se dedique a un único tratamiento (multi-tanques), para lo cual Zhou y Li (2003) propusieron un modelo de programación lineal entera mixta. Liu et al (2002) desarrollaron un modelo no sólo para multi-tanques, sino también para tanques multi-función (cuando en un tanque se realiza más de una operación diferente).

La producción simultánea de ítems múltiples (nuestro modelo lo sustituirá por piezas) ha sido tratada mediante diferentes tipos de procedimientos: heurísticas, para programación no cíclica, como Yih (1994) y Paul et al (2007); propagación de restricciones, como Hindi y Fleszar (2004); o *branch-and-bound* para programación cíclica, como Lei y Liu (2001).

En nuestro caso, se afronta el CHSP con una sola grúa, tanques con capacidad para una pieza, sin multi-tanques, sin tanques multi-función y sin ninguna otra restricción adicional. Las estaciones de carga y descarga se consideran a cada extremo de la línea, aunque el modelo también podría ser aplicado si coincidieran en el mismo lugar adaptando los tiempos de grúa. El modelo desarrollado permite tratar cualquier grado de ciclos. Recientemente, Manier y Bloch (2003) agruparon los diversos trabajos sobre este problema y elaboraron una clasificación de las diferentes variantes tratadas. Según su notación, nuestro caso puede clasificarse como: CHSP | mt/ /diss | /n,mt+2 | Tmin, siendo n el grado de los ciclos considerados y mt todos los tanques en que se realizan operaciones.

En la Sección 2, se describe el modelo 1-cíclico para presentar el modelo 2-cíclico en la Sección 3 y extenderlo al caso n -cíclico. La Sección 4 se propone el *branch-and-bound* en el cual en cada nodo se resuelve un grafo. La Sección 5 muestra la experiencia computacional y las conclusiones del trabajo se recogen en la Sección 6.

2. Formulación del problema cíclico básico

2.1. El problema de secuencias 1-cíclicas

En una línea compuesta por m tanques, donde suponemos que cada operación tiene lugar en un tanque, se debe producir una gran cantidad de piezas. Para cada operación, se define una ventana de tiempo. Para que las piezas fluyan a lo largo de la línea, una grúa mueve las piezas desde un tanque i al de la siguiente operación, $i+1$. Entre dos movimientos de grúa con carga, la grúa realiza a menudo viajes sin carga para ocupar el tiempo con traslados de otra pieza.

Sea:

i índice de tanques ($i=1,\dots,m$); estación de carga ($i=0$); estación de descarga ($i=m+1$)

$[a_i, b_i]$ ventana temporal en el tanque i ($i=1,2,\dots,m$)

f_i tiempo de movimiento de grúa con carga entre tanques i e $i+1$ ($i=0,1,\dots,m$).

$e_{i,i'}$ tiempo de movimiento de grúa sin carga entre tanques i e i' ($i,i'=0,\dots,m+1$).

La idea de trabajar con secuencias cíclicas se asocia a la fabricación de piezas idénticas desde Phillips y Unger (1976). La grúa repite movimientos cíclicos para tratar una serie de piezas, suficiente larga para considerarla infinita. El objetivo es minimizar el tiempo de ciclo, definido

como período consumido por la grúa para llevar a cabo una secuencia completa de movimientos (Shapiro y Nuttle, 1988).

Si una única pieza entra en el sistema productivo y una lo abandona durante cada ciclo, se habla de programación **1-ciclo** o de ciclo simple. En cambio, si entra y sale del sistema más de una pieza en cada ciclo, se habla de programación **n-ciclo**. Autores como Shapiro y Nuttle (1988) propusieron que sus algoritmos podían servir para el caso n-ciclo igual que lo hacían para el 1-ciclo, pero no sin aportar resultados. Lei y Wang (1989) afirmaron que el problema n-ciclo es también NP-completo.

Sea $\mathbf{H}=(h_0, h_1, \dots, h_m)$ una permutación circular de movimientos de grúa con carga, donde $h_{[l]}=h_i$ ($i, l=0, \dots, m$) significa que en la posición l de la secuencia se visita el tanque i . Se supone que $h_{[0]}=h_0$. A cada vector \mathbf{H} , se asocia otro vector \mathbf{T} con sus respectivos instantes t_i , en que la grúa toma una pieza del tanque i . Se establece $t_0=0$. Una secuencia cíclica (\mathbf{H}, \mathbf{T}) será factible si y sólo si hay una pieza en el tanque donde la grúa debe recogerlo y el tanque destino está vacío, y además las duraciones de las piezas en los tanques cumplen las ventanas temporales. El objetivo es minimizar el tiempo de ciclo C . Dada una secuencia \mathbf{H} , el problema de minimizar el tiempo de ciclo se define, a partir de los parámetros $ka_i, kb_i \in \{0,1\}$, como:

$$[\text{MIN}] C \quad (1)$$

s. a.:

$$t_i - t_{i-1} \geq a_i + f_{i-1} - ka_i \cdot C \quad i=1, \dots, m \quad (2)$$

$$t_{i-1} - t_i \geq -b_i - f_{i-1} + kb_i \cdot C \quad i=1, \dots, m \quad (3)$$

$$t_{[l]} - t_{[l-1]} \geq f_{[l-1]} + e_{[l-1]+1, [l]} \quad l=1, \dots, m \quad (4)$$

$$t_0 - t_{[m]} \geq f_{[m]} + e_{[m]+1, 0} - C \quad (5)$$

$$t_i \geq 0 \quad i=0, \dots, m \quad (6)$$

$$C \geq 0 \quad (7)$$

En (1) se define la función objetivo. Las restricciones de ventanas temporales son (2) y (3); las de movimientos de grúa en un ciclo son (4) y (5). Para el último movimiento, en (5) se incluye el tiempo de ciclo C . Las restricciones de no negatividad son (6) y (7).

2.2. De las secuencias 1-cíclicas a las n-cíclicas

Si la programación considera n-ciclos, esto es equivalente a considerar hasta “n piezas” diferentes que tienen que visitar cada tanque en un ciclo. El modelo presentado considera tanques “multiproducto” (con capacidad para más de una pieza, pero no simultánea), aunque de hecho sólo tengan capacidad para una única pieza. Por tanto, se puede resolver el problema transformando las m operaciones de los modelos 1-ciclo en $n \cdot m$ operaciones, ya que se multiplica la capacidad de tanques por el número de piezas por ciclo, n .

3. Modelo para el problema de secuencias 2-cíclicas

3.1. Definiciones, notación e hipótesis

En un modelo 2-cíclico, por ejemplo, se determina una secuencia cíclica que alterna piezas de un hipotético producto 1 (piezas impares) y de otro hipotético producto 2 (piezas pares). Es decir, se introduce una pieza impar y otra par en la línea y se acaba otras dos (una impar y otra par) durante cada ciclo. Así, se busca realizar los movimientos de grúa necesarios para que cada "tipo de pieza" (par o impar) esté en las mismas posiciones a inicio y fin de cada ciclo.

Previamente, se definen dos conceptos (que se muestran en la Figura 1):

- Un **tanque** es cada entidad física de la línea de producción, donde se realizan las operaciones; se considera tantos tanques como tratamientos u operaciones.
- Una **etapa** es cada una de las operaciones de una pieza cualquiera en un tanque cualquiera de la línea.

Sean $k-1$ y k dos etapas consecutivas del modelo 2-cíclico. Si k es impar, las etapas $k-1$ y k pertenecen a tanques diferentes; si k es par, son de un único tanque. Las restricciones de ventanas temporales, definidas por tanque, se convierten según las k etapas del modelo. De manera similar, se convierten los movimientos de grúa.

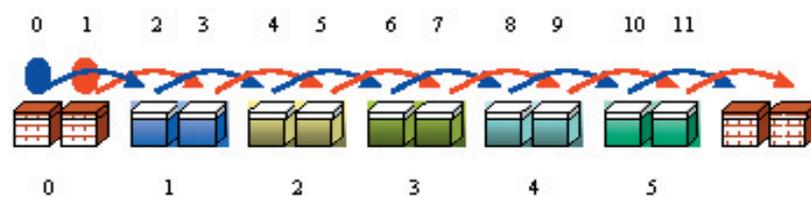


Figura 1. Representación de tanques (notación inferior) y etapas (notación superior) en el modelo.

Se define los siguientes parámetros de manera equivalente al caso 1-cíclico:

p número de etapas

s_0, s_1 etapas para la estación de carga (piezas impares y pares)

s_2, \dots, s_{p+1} etapas para los tanques $1, \dots, m$ (piezas impares y pares alternativamente)

s_{p+2}, s_{p+3} etapas para la estación de descarga (piezas impares y pares, respectivamente)

$a_k = a_{i*2+(j-1)}$ tiempo mínimo de ventana temporal en la etapa k , tanque i ($k=2, \dots, p+1$)

$b_k = b_{i*2+(j-1)}$ tiempo máximo de ventana temporal en la etapa k , tanque i ($k=2, \dots, p+1$)

f_k tiempo de movimiento de grúa con carga entre etapas s_k y s_{k+2} ($k=0, \dots, p+1$)

$e_{k,k'}$ tiempo de movimiento de grúa sin carga entre de la etapa k a la k' ($k, k'=0, \dots, p+3$)

Una secuencia cíclica, definida por dos vectores (\mathbf{H}, \mathbf{T}) , tiene $p+2$ componentes ($k=0, \dots, p+1$)

ya que desde las etapas $p+2$ y $p+3$ no hay movimiento de transporte de piezas: $\mathbf{H}=(h_{[0]}, h_{[1]}, \dots, h_{[p+1]})$ con $h_{[0]}=h_0$; $\mathbf{T}=(t_{[0]}, t_{[1]}, \dots, t_{[p+1]})$ con $t_{[0]}=0$ y $t_{[0]} < t_{[1]} < \dots < t_{[p+1]}$. Así, $h_{[l]}=h_k$ ($k, l=0, \dots, p+1$) es el l -ésimo movimiento con una pieza desde la etapa s_k a la s_{k+2} y $t_{[l]}=t_k$ ($k, l=0, \dots, p+1$) el instante en que empieza dicho movimiento. El modelo se plantea para trabajar en régimen permanente y omite el estado transitorio, con menor relevancia en la programación global.

3.2. Modelo de resolución de una secuencia

El tiempo de ciclo $C(\mathbf{H}, \mathbf{T})$ es el tiempo necesario para completar los $p+2$ movimientos de transporte de la secuencia \mathbf{H} . Las variables del modelo son $p+1$: una, para el tiempo de ciclo C , más p variables, de los valores del vector \mathbf{T} , ya que $t_0=0$ está fijado por hipótesis.

Sean $h_{[i]}=h_{k-2}$ y $h_{[j]}=h_k$ dos movimientos consecutivos sobre una misma pieza, ambos sobre la etapa k . Las restricciones (2) y (3) sobre duraciones de las operaciones en los tanques se convierten en (9) y (10). Sean $h_{[k]}$ y $h_{[k+1]}$ ($k=0, \dots, p$), las restricciones de tiempos de transporte (4) y (5) se convierten en (11) y (12). Así pues:

$$[\text{MIN}] C \quad (8)$$

s. a.:

$$a_k \leq t_{[j]} - (t_{[i]} + f_{k-2}) \leq b_k \quad k=2,3, \dots, p+1 \quad \text{si } j>i \quad (9)$$

$$a_k \leq t_{[j]} + C - (t_{[i]} + f_{k-2}) \leq b_k \quad k=2,3, \dots, p+1 \quad \text{si } j>i \quad (10)$$

$$t_{[k+1]} \geq t_{[k]} + f_{[k]} + e_{[k+2],[k+1]} \quad k=0,1, \dots, p \quad (11)$$

$$t_0 + C \geq t_{[p]} + f_{[p]} + e_{[p+2],0} \quad (12)$$

$$t_k \geq 0 \quad k=0, \dots, p+1 \quad (13)$$

$$C \geq 0 \quad (14)$$

En total, son $3 \cdot p+1$ restricciones: $2 \cdot p$ de (9) y (10) más $p+1$ al considerar (11) y (12).

Así mismo, considerando dos tanques consecutivos ($i, i+1$) y sus cuatro respectivas etapas asociadas ($k, k+1, k+2, k+3$), el orden de movimientos viene prefijado para que se puedan alternar piezas pares e impares (Figura 2).

Si se considera que la grúa tan sólo debe visitar los tanques 0 y 1, la única secuencia factible de movimientos es $\mathbf{H}=(h_0, h_2, h_1, h_3)$. El primer movimiento con carga (pieza impar) comienza en la etapa 0 (tanque 0) y acaba en la etapa 2 (tanque 1). La pieza par en la etapa 1 (tanque 0) no puede moverse al tanque 1 porque ya está ocupado por la anterior pieza. Es obligado el movimiento con carga desde la etapa 2 (tanque 1). Entonces, ya es posible mover la pieza par desde la etapa 1 (tanque 0) a la 3 (tanque 1). De nuevo, no puede moverse la pieza impar desde la etapa 0 (tanque 0). Así pues, el último de los cuatro movimientos es desde la etapa 3 (tanque 1). A esta condición la llamaremos de **subsecuencias coherentes**. Dicha condición también aparece en Lei y Liu (2001).

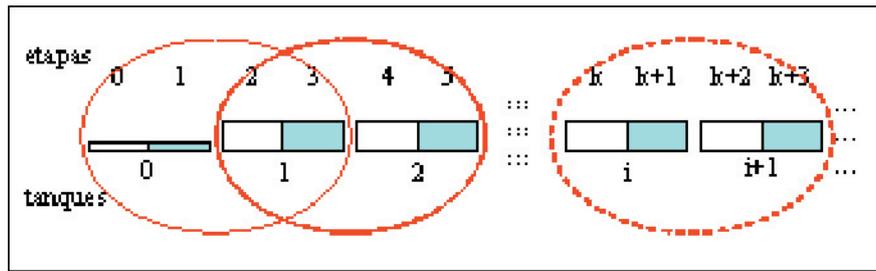


Figura 2. Relación entre etapas de dos tanques consecutivos (caso 2-cíclico).

Para añadir un nuevo tanque al modelo, al crear los vértices descendientes en el *branch-and-bound*, se añaden las dos siguientes etapas a las ya consideradas. En ese momento, sabiendo que se trata de permutaciones circulares, las cuatro últimas etapas en la línea parcial considerada deben cumplir el equivalente a la misma condición descrita para los tanques 0 y 1.

4. Procedimiento de resolución *branch-and-bound*

Uno de los métodos para alcanzar una solución óptima en un problema es el *branch-and-bound*. En este caso, el desarrollo del árbol implica el alargamiento de la línea con un nuevo tanque (y sus etapas asociadas) respecto al vector \mathbf{H} del nodo del nivel inmediatamente superior. La construcción de este tipo de árbol ya se define en Shapiro y Nuttle (1988).

4.1. Definición y resolución de nodos

El número de niveles del árbol coincide con los tanques del proceso, m . El nodo raíz está formado por la secuencia $\mathbf{H}=(h_0, h_2, h_1, h_3)$, único orden coherente para los tanques 0 y 1.

Sea un vértice de nivel r ($r=2, \dots, m$), definido por una permutación de los movimientos asignados a las primeras $2 \cdot (r+1)$ etapas. Su secuencia $\mathbf{H}_{[v][r]}=(h_{[0]}, \dots, h_{[1]}, \dots, h_{[2r+1]})$ corresponde a una permutación tal que $\{h_{[0]}=h_0, h_{[1]} \mid l=1, 2, \dots, 2r+1\}$ y $h_{[l]}$ debe respetar las condiciones de subsecuencias coherentes. El subíndice v indica en qué orden aparece el nodo en el algoritmo.

Para resolver la secuencia de cada nodo, se plantea un grafo con las $3p+1$ restricciones (9)-(12) bajo un mismo patrón, para representarlas como arcos (Mateo et al, 2002):

$$t_{Gto(q)} - t_{Gfrom(q)} \geq G_{time(q)} + G_{cycle(q)} \cdot C \quad q=1, \dots, 3p+1 \quad (15)$$

Los cuatro vectores de $3p+1$ componentes del patrón se llaman, respectivamente:

- $G_{from}(q) \in \{0, \dots, p+1\}$: vértice origen del arco, estado inicial de la restricción q .
- $G_{to}(q) \in \{0, \dots, p+1\}$: vértice destino del arco, estado final de la restricción q .
- $G_{time}(q) \in \mathbb{Z}$: valor positivo o negativo del tiempo a cumplirse entre el inicio de los movimientos desde las etapas $G_{from}(q)$ y $G_{to}(q)$ de la restricción q .
- $G_{cycle}(q) \in \{-1, 0, 1\}$: valor que indica la influencia de diferentes ciclos en la restricción q .

El grafo se resuelve aplicando el algoritmo de Ford. El tiempo de ciclo tentativo inicial C^0 es la mayor de las cotas inferiores que se pueden calcular con una selección de circuitos en el grafo

con valores no positivos, mezcla de restricciones de tanque (9) y (10) con otras restricciones de grúa, (11) y (12). No obstante, siempre hay dos cotas básicas: como cota inferior, la suma de tiempos de grúa de las restricciones (11) y (12), considerando posibles dominancias de restricciones (9); y como cota superior, la mejor solución en curso.

La evaluación de caminos óptimos en el grafo para C^0 puede conducir a la convergencia (y entonces C^0 es el valor óptimo esperado C^*) o no. Las diferencias en iteraciones sucesivas (ΔC) se usan para probar nuevos valores del tiempo de ciclo, lo cual puede demostrarse mediante propiedades utilizadas en programación dinámica (Mateo et al, 2002). El valor C puede incrementarse mientras no supere la cota superior.

4.2. Ramificación y acotación

Para la ramificación, si v es un nodo de nivel r ($r < m$), el vector \mathbf{H} en nodos descendientes se crea añadiendo dos nuevas etapas, las del tanque del siguiente nivel. En este momento, se comprueban las condiciones sobre subsecuencias coherentes y, así, las reglas de orden entre etapas permiten aceptar o rechazar el potencial nodo descendiente.

Para determinar una cota del valor C de las permutaciones que contiene dicho vértice v , se considera solamente los movimientos incluidos en dicho vértice y, si es necesario, un tiempo adicional para los movimientos aún no fijados. Sea una secuencia \mathbf{H} , una cota inferior de C que incorpore los movimientos no asignados ($h_1 \in \mathbf{U}$) a $\mathbf{H}_{[v]/[r]}$ e intente ocupar los tiempos ociosos de la grúa es:

$$LB(\mathbf{H}_{[v]/[r]}) = C(\mathbf{H}_{[v]/[r]}) + \max \left\{ \left[\sum_{h_1 \in \mathbf{U}} (f_1 + e_{1|1+1}) - \sum_{h_1 \in \mathbf{H}} (w_k) \right], 0 \right\} \quad (16)$$

donde $C(\mathbf{H}_{[v]/[r]})$ el tiempo de ciclo para dicha secuencia, $\mathbf{U} = \{h_{2(r+1)}, \dots, h_{p+1}\}$ el conjunto de movimientos no asignados a la secuencia y w_k el tiempo de espera de la grúa en una etapa k .

5. Experiencia computacional

5.1. Introducción, datos y objetivo

La experiencia computacional se basa en un conjunto de ejemplares (Mateo, 2001) con diferentes anchos de ventanas temporales y relaciones entre tiempos de movimientos de grúa con y sin carga.

Para clasificar las ventanas temporales, se ha considerado la diferencia entre tiempos máximo y mínimo ($b_k - a_k$) en cada etapa. Partiendo de Ng(1996), el tiempo mínimo de las operaciones se genera aleatoriamente según la ley uniforme: $a_k = U[20; 80]$. Para determinar las ventanas temporales en los baños, los ejemplares se dividen en 2 grupos: $V1$, para $b_k - a_k = U[1, 2a_k; 2a_k]$; $V2$, para $b_k - a_k = U[2a_k; 3a_k]$.

La segunda clasificación tiene en cuenta la velocidad de desplazamiento de la grúa. Tomando como referente a Ng(1996), el tiempo mínimo de movimiento de la grúa sin carga entre dos etapas k y $k+2$ se establece en $e_{k,k+2} = U[5; 10]$. Ng(1996) suma un valor constante para coger y dejar un objeto; en cambio, aquí se opta por establecer tres grupos considerando el cociente de tiempos de movimiento entre dos etapas k y $k+2$ con carga f_k y sin carga $e_{k,k+2}$: $G1$, para $f_k < 2 \cdot e_{k,k+2}$; $G2$, para $f_k = 2 \cdot e_{k,k+2}$; $G3$, para $f_k > 2 \cdot e_{k,k+2}$.

Estos grupos en ambos parámetros de entrada (ventanas y velocidades de grúas) comportan 6

colecciones de ejemplares. Se estudian los ejemplares entre 5 y 8 tanques, disponiendo para cada subgrupo de 10 ejemplares diferentes, lo que en total supone 240 instancias.

El objetivo es comparar los resultados de los ejemplares tipo para los casos n -cíclicos, donde $1 \leq n \leq 4$. Se determinará cuál es el tiempo de ciclo en los cuatro casos y qué incrementos de productividad pueden producirse al incluir más piezas por ciclo.

5.2. Análisis de resultados entre los casos n -cíclico y 1-cíclico

Sean $C^*(H_1)$ y $C^*(H_n)$ los tiempos de ciclo óptimos para los casos 1-cíclico y n -cíclico, respectivamente. Obviamente, los resultados esperados deben cumplir la relación (17):

$$n \cdot C^*(H_1) \geq C^*(H_n) \quad (17)$$

Para mostrar el creciente grado de complejidad para las secuencias n -cíclicas, la Tabla 1 ofrece los tiempos de cálculo para los cuatro tipos de ciclos.

Tabla 1. Valores medios de tiempo de ciclo en los casos 1-cíclico, 2-cíclico, 3-cíclico y 4-cíclico, así como porcentajes medios de reducción comparados con los valores del caso 1-cíclico.

Vent.	Grúa	1-ciclo	2-ciclo	2-c vs. 1-c	3-ciclo	3-c vs. 1-c	4-ciclo	4-c vs. 1-c
V1	G1	281,2	487	8,99%	764,3	6,08%	938,7	9,04%
	G2	358,5	584	13,29%	893,8	12,68%	1090,1	13,62%
	G3	405,3	673	13,86%	1063,0	10,95%	1268,9	14,19%
V2	G1	253,3	468	5,04%	710,8	4,51%	971,9	5,04%
	G2	311,4	540	9,17%	843,4	6,47%	1155,6	9,17%
	G3	411,7	683	12,69%	1068	10,00%	1439,8	12,70%

Se observa como casi siempre los ejemplares con ventanas temporales más reducidas (V1) pueden conseguir mayores reducciones del tiempo de ciclo con n -ciclos que con ventanas más anchas (V2).

La Tabla 2 muestra la reducción media porcentual del tiempo de ciclo según número de tanques (columnas) y ventanas temporales y velocidades de grúa (filas), comparando el mejor de los tiempos de ciclo (de los 4 valores n probados) con el tiempo de ciclo para el caso 1-cíclico.

En primer lugar, se puede apreciar una relación más o menos directa entre el potencial de reducción del tiempo de ciclo y el número de tanques. El tiempo se reduciría en menos del 5% en media para líneas con sólo 5 tanques y ventanas V1, pudiendo alcanzar para ejemplares de hasta 8 tanques una mejora de entre un 10% y casi un 30%.

Tabla 2. Reducción media (%) del tiempo de ciclo entre secuencias 1-ciclo y la mejor n -ciclo.

Ventana	Grúa	Tanques			
		5	6	7	8
V1	G1	5,64%	9,35%	12,07%	11,74%
	G2	6,01%	13,40%	16,91%	29,48%
	G3	7,13%	19,60%	13,00%	22,81%
V2	G1	0,62%	5,46%	7,22%	10,90%
	G2	2,28%	4,99%	5,32%	14,73%
	G3	5,59%	5,28%	4,65%	28,48%

En la Tabla 3 se incluyen los tiempos medios de cálculo, que van desde pocas milésimas de

segundo, para ejemplares con 5 tanques y caso 2-cíclico, hasta unos 4 minutos, para ejemplares de 8 tanques en el caso 4-cíclico y con ventanas V2 de mayor amplitud (lo que implica gran cantidad de subsecuencias).

Tabla 3. Tiempos medios de cálculo para las secuencias n-cíclicas (n=2,3,4), en segundos.

Ventana	Secuencia	Tanques			
		5	6	7	8
V1	2-ciclo	0,005	0,026	0,116	0,567
	3-ciclo	0,152	1,731	18,794	30,076
	4-ciclo	8,639	30,059	82,593	104,701
V2	2-ciclo	0,005	0,026	0,178	1,327
	3-ciclo	0,153	2,687	23,545	90,351
	4-ciclo	14,323	28,896	75,862	207,990

El algoritmo, escrito en Visual C++, se ha ejecutado en Pentium IV 2.60 Ghz, 256 Mb RAM.

6. Conclusiones

El modelo propuesto, basado en los más comunes para el problema HSP, propone diferenciar tanques de etapas, siendo una etapa cada una de las operaciones desarrolladas en un tanque. Que sea la primera o n-ésima pieza en pasar por el tanque queda contemplado en el modelo. El análisis de diferentes rangos de ventanas temporales y velocidades de grúa permite saber cuando es más ventajosa una secuencia n-cíclica que 1-cíclica, por ejemplo con ventanas de tipo V1. En la comparación de los diversos casos n-cíclicos, hasta n=4, se ha comparado la reducción media del tiempo de ciclo con el incremento de tiempo de cálculo, siendo a menudo provechosa.

Referencias

- Hindi, K.S.; Fleszar, K. (2004). A constraint propagation heuristic for the single-hoist, multiple-products scheduling problem, *Computers & Industrial Engineering*, vol. 47, pp. 91-101.
- Lei, L.; Liu, Q. (2001). Optimal cyclic scheduling of a robotic processing line with two-product and time-window constraints, *INFOR*, vol. 39, n. 2, pp. 185-199.
- Lei, L., Wang, T.J. (1989). A proof: the cyclic hoist scheduling problem is NP-complete, Working paper #89-0016, Rutgers University.
- Lei, L.; Wang, T.J. (1991). The minimum common-cycle algorithm for cyclic scheduling of two material handling hoists with time window constraints, *Management Science*, vol. 37, n. 12, pp. 1629-1639.
- Liu, J.; Jiang, Y., Zhou, Z. (2002). Cyclic scheduling of a single hoist in extended electroplating lines: a comprehensive integer programming solution, *IIE Transactions*, vol. 34, pp. 905-914.
- Manier, M.A., Bloch, C. (2003). A classification for Hoist Scheduling Problems, *International Journal of Flexible Manufacturing Systems*, vol. 15, n. 1, pp. 37-55.
- Mateo, M. (2001). Procedimientos de secuenciación y programación en un sistema productivo de estaciones en serie con transportadores asíncronos de material. Tesis doctoral, Universitat Politècnica de Catalunya, Barcelona.

Mateo, M., Companys, R., Bautista, J. (2002). Resolution of graphs with Bounded Cycle Time for the Cyclic Hoist Scheduling Problem. 8th International Workshop on Project Management and Scheduling, pp. 257-260.

Ng, W.C. (1996). A branch and bound algorithm for hoist scheduling of a circuit board production line. *The International Journal of Flexible Manufacturing Systems*, vol. 8, pp. 45-65.

Paul, H.J.; Bierwirth, C.; Kopfer, H. (2007). A heuristic scheduling procedure for multi-item hoist production lines, *International Journal of Production Economics*, vol. 105, pp. 54-69.

Phillips, L.W., Unger, P. S. (1976). Mathematical Programming Solution of a Hoist Scheduling Program, *AIIE Transactions*, vol. 8, n. 2, pp. 219-225.

Shapiro, G.W., Nuttle, H. W. (1988). Hoist Scheduling For A PCB Electroplating Facility, *IEE Transactions*, vol. 20, n. 2, pp. 157-167.

Yih, Y. (1994). An algorithm for hoist scheduling problems, *International Journal of Production Research*, vol. 32, n. 3, pp. 501-516.

Zhou, Z., Li, L. (2003). Single hoist cyclic scheduling with multiple tanks: a material handling solution, *Computers & Operations Research*, vol. 30, pp. 811-819.