



Ingeniería Industrial

ISSN: 1815-5936

Facultad de Ingeniería Industrial, Instituto Superior
Politécnico José Antonio Echeverría, Cujae.

Díaz-Cazañas, Ronald; De La Paz-Martínez, Estrella M.; Delgado-Sobrino, Daynier R.

Algoritmo para la programación integrada producción -
mantenimiento preventivo en máquinas paralelas idénticas

Ingeniería Industrial, vol. XL, núm. 3, 2019, Septiembre-Diciembre, pp. 260-271

Facultad de Ingeniería Industrial, Instituto Superior Politécnico José Antonio Echeverría, Cujae.

Disponible en: <https://www.redalyc.org/articulo.oa?id=360461152005>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

UAEI
redalyc.org

Sistema de Información Científica Redalyc

Red de Revistas Científicas de América Latina y el Caribe, España y Portugal

Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso
abierto



Algoritmo para la programación integrada producción – mantenimiento preventivo en máquinas paralelas idénticas

Algorithm for integrated programming of production and preventive maintenance on identical parallel machines

Ronald Díaz-Cazañas^I, Estrella M. De La Paz-Martínez^I, Daynier R. Delgado-Sobrino^{II}

^I Universidad Central “Marta Abreu” de Las Villas, Santa Clara. Cuba.

Correo-electrónico: ronalddc@uclv.edu.cu, estrella@uclv.edu.cu

^{II} Slovak University of Technology in Bratislava, Faculty of Materials Science and Technology in Trnava, Trnava.Slovak Republic

Correo-electrónico: daynier_sobrino@stuba.sk

Recibido: 17 de julio del 2019.

Aprobado: 28 de agosto del 2019.

RESUMEN

Se muestra un algoritmo para la programación integrada de producción y mantenimiento preventivo en una configuración productiva de máquinas paralelas idénticas, teniendo como objetivo minimizar el tiempo total de fabricación. El algoritmo fue estructurado en dos fases. En la primera se obtiene una solución inicial, resultado del diseño de una heurística constructiva inspirada en la regla de despacho *Largest Processing Time*, con algunas modificaciones para considerar la naturaleza estocástica del problema y la posibilidad de que varios trabajos puedan tener fechas de lanzamiento diferentes. En la fase dos se intenta mejorar la solución inicial utilizando un método de recocido simulado. Como forma de evaluar su factibilidad el algoritmo fue aplicado en una empresa productora de artículos plásticos, demostrándose mediante los resultados obtenidos sus potencialidades como mecanismo de mejoramiento de las capacidades de operaciones.

Palabras claves: producción, mantenimiento, programación integrada, heurística, recocido simulado.

ABSTRACT

The present paper introduces an algorithm for the integrated programming of production and preventive maintenance in a productive configuration of identical parallel machines, aiming to minimize the total manufacturing time. The algorithm has been structured in two phases; in the first one, an initial solution is obtained, resulting from the application of a designed and proposed constructive heuristic that is inspired by the Largest Processing Time dispatch rule, with some modifications allowing it to consider the stochastic nature of the problem and the possibility that several jobs may have different release dates; in phase 2, an attempt is made to improve the initial solution using a simulated annealing method. As a way of evaluating its feasibility the algorithm was implemented in a company producing products derived from plastic, demonstrating through the results obtained its potential as a mechanism to improve operational capabilities.

Key words: production, maintenance, integrated programming, heuristic; simulated annealing.

I. INTRODUCCIÓN

La degradación de algunas variables propias de la competitividad a menudo encuentra entre sus principales causas la falta de integración entre la gestión de la producción y la gestión del mantenimiento. Si bien este escenario se presenta frecuentemente en la práctica empresarial la situación en la esfera académica no resulta muy diferente, pues tradicionalmente los métodos desarrollados para la gestión de la producción no incluyen explícitamente los requerimientos de mantenimiento, lo que conduce a la generación de soluciones subóptimas [1; 2; 3; 4].

En una configuración productiva caracterizada por máquinas paralelas minimizar el *makespan* ($C_{\max}$) se convierte en un objetivo de especial interés, puesto que en la práctica los programadores de producción intentan frecuentemente lograr un equilibrio adecuado en la utilización de los recursos [5]. Dada su analogía con el conocido problema del *bin packing*, el cual pertenece a la clase de problemas NP-hard la existencia de un método exacto que en tiempo polinomial sea capaz de resolver de manera óptima todas las instancias resulta improbable [6, 7]. Si a esto se une el carácter aleatorio del tiempo de completamiento de los trabajos debido a las paradas por fallos del equipamiento la complejidad del problema se incrementa, por lo que el diseño de métodos heurísticos capaces de proveer soluciones cercanas a la óptima en un tiempo razonable parece ser la mejor alternativa.

En su versión tradicional (determinística) el problema de minimización del *makespan* en máquinas paralelas idénticas ha encontrado estrategias de solución basadas en algoritmos heurísticos tales como *First Fit Decreasing* (FFD) y *Best Fit Decreasing* (BFD) [8]. Así como la conocida regla de despacho *Largest Processing Time* (LPT) [9]. Años más tarde Coffman et al. (1978) proponen el algoritmo MULTIFIT, el cual posee un ratio de desempeño superior a LPT. Otras estrategias de solución más refinadas para este problema, basadas en el enfoque de ramificación y acotamiento, han sido propuestas por Carlier (1987) y Gharbi y Haouari (2002) [10; 11].

La extensión de este problema al considerar limitaciones en la disponibilidad del equipamiento fue abordada por Lee (1991; 1996; 2000) [12; 13; 14]. Este autor estudió dicha configuración productiva cuando las máquinas presentaban un único periodo de no disponibilidad durante el horizonte de planificación. Otras contribuciones importantes corresponden a Xu et al. (2008) y Lee y Wu (2008) quienes abordaron el problema desde una perspectiva determinística en lo que refiere a los instantes de comienzo de las intervenciones preventivas.

En los últimos años la mayoría de las estrategias de solución para este tipo de problema han estado basadas en algoritmos de búsqueda local, inteligencia artificial y en el uso de la simulación. En este sentido pueden citarse a autores como Mirabedini y Iranmanesh (2014) quienes aplicaron un algoritmo genético dinámico multiobjetivo y un esquema de optimización basado en enjambre de partículas como métodos de solución alternativos para un modelo donde se buscaba optimizar el *makespan* y otras medidas de desempeño [17]. Wang and Liu (2015) estudiaron este problema considerando que todos los productos estaban disponibles para iniciar el procesamiento al comienzo del horizonte de planificación. Diseñaron un modelo matemático no lineal y para la solución desarrollaron un algoritmo genético multiobjetivo elitista [18]. De modo similar Da et al. (2016) desarrollaron un algoritmo genético elitista pero para una variante de este problema en la que tanto el equipamiento como el tiempo de procesamiento de los trabajos se deterioran (alargan) en el tiempo [19].

A pesar del auge alcanzado en la utilización de los métodos procedentes del campo de la Inteligencia Artificial, las heurísticas constructivas continúan siendo una alternativa tentadora cuando se trata de desarrollar métodos relativamente sencillos, capaces de generar buenas soluciones con rapidez. Incluso las heurísticas han demostrado su utilidad en el desarrollo de métodos híbridos en los que frecuentemente un algoritmo de búsqueda local mejora la solución inicial provista por la heurística.

En este sentido, el presente trabajo tiene como objetivo desarrollar un método, relativamente sencillo, para la generación de un programa integrado de producción y mantenimiento preventivo en una configuración productiva de máquinas paralelas idénticas. Dirigido a minimizar el *makespan* ($C_{\max}$), que solventa algunas de las carencias de otros métodos encontrados en la literatura para enfrentar este tipo de problema. Donde no siempre se consideran fechas de lanzamiento diferentes para varios trabajos y, con frecuencia, los instantes de inicio de las intervenciones de mantenimiento preventivo (MP) se consideran conocidos previamente, constituyendo restricciones de entrada que limitan el carácter integrado de la solución.

Específicamente el método que se propone consiste en un algoritmo híbrido de dos fases, denominado HCSA. En la primera fase una Heurística Constructiva (HC) genera una solución

inicial, la cual se tratará de mejorar en una segunda fase mediante un enfoque de recocido simulado (SA). Se distingue que se considera que los diferentes productos pueden tener distintas fechas de lanzamiento y que el tiempo entre intervenciones de MP no es fijo, sino que constituye una variable de decisión dentro del algoritmo. Dicha variable trata de optimizar de conjunto con el programa de fabricación. Su implementación fue realizada en Matlab R2015a.

II. MÉTODOS

Se emplean métodos estadístico-matemáticos con la finalidad de desarrollar un programa integrado de producción y mantenimiento preventivo en una configuración productiva de máquinas paralelas idénticas. Para una mejor comprensión Esta sección se ha estructurado en 3 epígrafes. En el primero se plantea una breve descripción del problema, en el segundo se definen las variables y parámetros del algoritmo y en el tercero se proporciona una descripción detallada de su diseño.

2.1 Breve descripción del problema

El sistema productivo consta de un conjunto de m máquinas, dispuestas en paralelo, capaces de procesar indistintamente n tipos de productos. Cada producto recibe todo el procesamiento en una y solo una de las máquinas disponibles. El tiempo de procesamiento de los trabajos se asume constante y conocido, siendo el mismo en cada máquina (máquinas idénticas) y una vez iniciado el procesamiento de un trabajo este no será interrumpido a menos que ocurran perturbaciones inevitables en el proceso, en cuyo caso, luego de restablecidas las condiciones de operación, se continuará el procesamiento del producto. Se asume que el tiempo entre fallos del equipamiento puede describirse mediante una distribución de densidad probabilística Weibull.

El algoritmo propuesto se dirige a establecer una asignación de trabajos a las diferentes máquinas, indicando además el momento más conveniente para efectuar las intervenciones de Mantenimiento Preventivo (MP) sobre estas, de manera que se minimice el tiempo total de fabricación (*makespan*). Se considera una política de MP del tipo *as good as new*, o sea, luego de ejecutar la intervención preventiva sobre el equipo se restablecen totalmente las condiciones operativas del mismo.

2.2 Definición de variables y parámetros

n : Número de trabajos (productos o lotes de productos) a ser programados.

m : Número de máquinas.

J : Conjunto de trabajos (productos o lotes de productos) a ser programados.

M : Conjunto de máquinas.

r_j : Instante de tiempo en que el trabajo j se encuentra disponible para iniciar su procesamiento.

p_j : Tiempo de procesamiento del trabajo j .

j_l, j_t : Subíndices utilizados para denotar el l - ésimo o el t - ésimo trabajo dentro de un arreglo de trabajos ordenados de acuerdo con un criterio definido.

I_k : Número de trabajos asignados a la máquina k

Tt_k : Tiempo de terminación de la máquina k .

a_{0k} : Edad efectiva de la máquina k al inicio del periodo de programación.

$a_{k(lk^-)}$: Edad efectiva de la máquina k antes de iniciar el procesamiento del trabajo $l_{(k-ésimo)}$

$a_{k(lk)}$: Edad efectiva de la máquina k luego de terminar el procesamiento del trabajo $l_{(k-ésimo)}$.

t_{ilk} : Tiempo de inicio esperado del trabajo l en la máquina k

TCE_{lk} : Tiempo de completamiento esperado del trabajo l en la máquina k .

TE_{lkPM1} : Tiempo de elaboración del trabajo l en la máquina k si la actividad de mantenimiento preventivo se realiza en esa máquina antes de procesar el trabajo $l_{(k-ésimo)}$.

TE_{lkPM0} : Tiempo de elaboración del trabajo l en la máquina k si no se realiza mantenimiento preventivo en esa máquina antes de iniciar su procesamiento.

t_{pk} : Tiempo medio de duración de la intervención de mantenimiento preventivo en la máquina k .

t_{rk} : Tiempo medio de duración de la intervención de mantenimiento correctivo en la máquina k .

β_k : Parámetro de forma de la distribución Weibull que describe la variable tiempo entre fallos asociada a la máquina k .

η_k : Parámetro de escala de la distribución Weibull que describe la variable tiempo entre fallos asociada a la máquina k .

PM_{lk} : Variable binaria que indica si la intervención de mantenimiento preventivo se realizará en la máquina k antes de iniciar el procesamiento del trabajo l .

A_k : Arreglo que contiene el subconjunto de trabajos asignados a la máquina k .

(a): Cardinalidad del subconjunto a.

w_{lt} : Peso específico del l -ésimo trabajo respecto al t -ésimo trabajo

ω : Constante positiva definida experimentalmente para decidir si es conveniente programar primero un trabajo de menor r y menor p que otro de mayor p y mayor r .

$C_{\max}$: Valor del *makespan* asociado a la asignación de trabajos e intervenciones de mantenimiento establecida como solución.

2.3 Diseño de las fases del algoritmo HCSA propuesto

2.3.1 Fase 1. Generación de la solución inicial mediante el diseño de una heurística constructiva (HC)

1. Inicializar $Tt_k = I_k = 0 \forall k \in M$; $l = 0$; $A_{(1)} = A_{(2)} = \dots = A_{(m)} = \emptyset$; $r_j \forall j = \{1, 2, \dots, n\}$; $a_{0k} \forall k = \{1, 2, \dots, m\}$, ω .

2. Formar el conjunto J de trabajos ordenados de manera no creciente según p_j

3. Hacer: $t = 1$; $S = \emptyset$; $P = \emptyset$; $l = l + 1$

Si $r_{jl} \leq \max(\min(r_{jl}), \min(Tt_k)) \forall j_l = (l, l+1, \dots, n)$, $\forall k \in M$ asignar el trabajo j_l a la máquina k $|TCE_{lk} = \min(TCE_{lk}) \forall k \in M$. En caso de empate seleccionar la máquina k $|a_{k(lk^-)} = \min(a_{k(lk^-)}) \forall k$ $|TCE_{lk} = \min(TCE_{lk})$. Si persiste el empate seleccionar la máquina k arbitrariamente. Lo cual se refleja en las ecuaciones 1, 2, 3, 4, 5 y 6.

$$I_k = I_k + 1 \quad (1)$$

$$TCE_{lk} = \begin{cases} \max(Tt_k; r_l) + TE_{lkPM1} & \text{si } PM_{lk} = 1 \\ \max(Tt_k; r_l) + TE_{lkPM0} & \text{en otro caso} \end{cases} \quad (2)$$

$$PM_{lk} = \begin{cases} 1 & \text{si } \left(t_{pk} + t_{rk} \left[\left(\frac{p_l}{\eta_k} \right)^{\beta_k} \right] \right) < \left(t_{rk} \left[\left(\frac{a_{k(lk^-)} + p_l}{\eta_k} \right)^{\beta_k} - \left(\frac{a_{k(lk^-)}}{\eta_k} \right)^{\beta_k} \right] \right) \\ 0 & \text{en otro caso} \end{cases} \quad (3)$$

$$TE_{lkPM1} = p_l + t_{pk} + t_{rk} \left[\left(\frac{p_l}{\eta_k} \right)^{\beta_k} \right] \quad (4)$$

$$TE_{lkPM0} = p_l + t_{rk} \left[\left(\frac{a_{k(lk^-)} + p_l}{\eta_k} \right)^{\beta_k} - \left(\frac{a_{k(lk^-)}}{\eta_k} \right)^{\beta_k} \right] \quad (5)$$

$$a_{k(lk^-)} = \begin{cases} 0 & \text{si } PM_{lk} = 1 \\ a_{k(lk^- - 1)} + p_{lk^- - 1} & \text{si } PM_{lk} = 0 \text{ y } I_k > 1 \\ a_{0k} & \text{si } I_k = 1 \end{cases} \quad (6)$$

Hacer $Tt_k = TCE_{lk}$ y $A_k = A_k \cup \{j_l\}$

Si $PM_{lk} = 1$ programar una actividad de PM en la máquina k antes de iniciar el trabajo j_l .

Programar los tiempos o instantes de inicio (t_{ilk}) y de completamiento (TCE_{lk}) esperados del trabajo j_l en la máquina k utilizando las ecuaciones 7 y 1 respectivamente.

$$t_{ilk} = \begin{cases} \max(Tt_k; r_l) & \text{si } PM_{lk} = 0 \\ \max(Tt_k + t_{pk}; r_l) & \text{en otro caso} \end{cases} \quad (7)$$

Si $\sum_{k=1}^m I_k < n$ repetir el paso 3 si no obtener $C_{\max} = \max(Tt_k) \forall k \in M$

Si $r_{jl} > \max(\min(r_j), \min(Tt_k)) \forall j = (l, l+1, \dots, n)$, $\forall k \in M$ hacer $Tt'_k = Tt_k$, $I'_k = I_k$, $a_{k(lk^-)}' = a_{k(lk^-)}$,

$A'_k = A_k \forall k \in M$ e ir al paso 4

4. Generar el subconjunto P de trabajos ordenados de manera no decreciente según r_j , $P \subset J \setminus \{j_l\}$; $A(k) \forall k \in M$ $|r_l > r_t \forall j_t \in P$

4.1 Calcular el peso relativo del trabajo j_l respecto al trabajo j_t (w_{lt}) mediante las ecuaciones 8 y $\bar{p}$ mediante la ecuación 9.

$$w_{lt} = \frac{\frac{p_l}{\bar{p}}}{r_{jl} - \max(\min(r_{jt}), \min(Tt_k))} \forall k \in M, j_t \in P \quad (7)$$

$$\bar{p} = \frac{\sum_{j \in P} p_j}{\#(P)} \quad (8)$$

Si $w_{lt} < \omega$ ir al paso 4.2 si no ir al paso 4.3.

4.2 Calcular el tiempo de completamiento esperado del trabajo j_t en cada máquina k mediante la ecuación (1).

Identificar la máquina k $|TCE_{tk} = \min(TCE_{tk})$. En caso de empate seleccionar k arbitrariamente

Si $w_{lt} \leq \omega$ ó $TCE_{tk} \leq r_{jl}$

hacer $Tt_k = TCE_{tk}$; $I_k = I_k + 1$; $S = S \cup \{j_t\}$

si no ir al paso 4.3

4.3 Hacer $t = t + 1$. Si $t \leq \#(P)$ y $r_{jl} > \min(Tt_k) \forall k \in M$ repetir el paso 4.1, si no ir al paso 5.

5. Si $S \neq \emptyset$ hacer $Tt_k = Tt'_k$, $I_k = I'_k$, $a_{k(lk^-)} = a_{k(lk^-)}' A_k = A'_k$ y asignar los trabajos del subconjunto S aplicando la regla combinada $r_j - LPT$. Si $S = \emptyset$ ir al paso 6.

6. Hacer $Tt_k = Tt'_k$, $I_k = I'_k$, $a_{k(lk^-)} = a_{k(lk^-)}' A_k = A'_k$ y asignar el trabajo j_l a la máquina k $|TCE_{lk} = \min \{TCE_{lk}\} \forall k = \{1, 2, \dots, m\}$. En caso de empate seleccionar la máquina k $|a_{k(lk^-)} = \min \{a_{k(lk^-)}\} \forall k$ $|TCE_{lk} = \min \{TCE_{lk}\}$. Si persiste el empate seleccionar la máquina k arbitrariamente. Ajustar $A(k) = A(k) \cup \{j_l\}$ y $Tt_k = TCE_{lk}$.

Si $PM_{lk} = 1$ programar una actividad de PM en la máquina k antes de iniciar el trabajo j_l .

Programar los instantes de inicio (t_{ilk}) y de completamiento (TCE_{lk}) esperados del trabajo j_l en la máquina k utilizando las ecuaciones 6 y 1 respectivamente.

Si $\sum_{k=1}^m I_k < n$ ir al paso 3, si no obtener $C_{\max} = \max \{Tt_k\} \forall k \in M$ y terminar

Definición de la regla combinada $r - LPT$

1. Ajustar $t_0 = \max(\min(r_j), \min(Tt_k)) \forall j \in S, k \in M$, $JA = \emptyset$, $MA = \emptyset$

2. Formar el subconjunto de trabajos $JA \subset S$ $|r_j \leq t_0 \forall j \in S$.

3. Formar el subconjunto de máquinas $MA \subset M$ $|Tt_k \leq t_0 \forall k \in M$

4. Si $\#(JA) \leq \#(MA)$ asignar en el instante t_0 cada trabajo de JA a alguna máquina k de MA , si no asignar los trabajos de JA a las máquinas de MA según la regla LPT.

Actualizar los arreglos A_k e $I_k \forall k \in MA$ $|\exists j \in A_k$. Hacer $Tt_k = TCE_{jk} | j \in A(k) \forall k \in MA$.

Evaluar la variable PM_{lk} en cada máquina k previo a la asignación de cada trabajo j . Si $PM_{lk} = 1$ programar una actividad de PM en la máquina k antes de iniciar el trabajo j .

Programar los instantes de inicio (t_{ilk}) y de completamiento (TCE_{lk}) esperados del trabajo j_l en la máquina k utilizando las ecuaciones 6 y 1 respectivamente.

5. Ajustar $S = S \setminus j \in JA | j \in A_k \forall k \in MA$. Si $S \neq \emptyset$ repetir el paso 1 si no terminar.

2.3.2 Fase 2 Mejoramiento de la solución inicial mediante un enfoque de recocido simulado

El enfoque o algoritmo de recocido simulado que se propone está compuesto por 3 funciones características del mismo: una función que define la estrategia para generar una solución vecina a partir de un solución actual, la cual será denominada *neighbord function*; una función de recocido, denominada *schedules generation*, la cual toma como entrada una solución determinada y la modificará mediante una cantidad de cambios proporcional al parámetro de temperatura y, por último, la función objetivo (*fitness function*). La solución tomada como punto de partida para iniciar el proceso de búsqueda (solución inicial) se obtiene a partir de la heurística constructiva (HC) definida en el epígrafe 2.3.1. Seguidamente se presenta el diseño de las tres funciones que integran el algoritmo de recocido simulado.

2.3.2.1 Función generadora de soluciones vecinas (*neighbord function*)

La estrategia definida para generar una solución vecina a partir de la actual consiste en seleccionar aleatoriamente un par de máquinas m_1 y m_2 , de manera que la máquina que presente el mayor tiempo de terminación esperado en la solución actual tenga mayor probabilidad de ser seleccionada como m_1 ; mientras que la máquina que presente el menor tiempo de terminación esperado tenga mayor probabilidad de ser seleccionada como m_2 . Un elemento del vector que contiene la solución de m_1 se selecciona aleatoriamente y se transfiere a una posición aleatoria dentro del vector que define la solución de m_2 . Si el elemento seleccionado en m_1 es una actividad de PM esta será transferida hacia m_2 con una probabilidad de 0,5. El pseudocódigo que define la función *neighbord function*, implementado en MATLAB R2105a, se muestra a continuación:

1. Definir *sch_neighb*: matriz que contiene la solución inicial obtenida mediante la heurística constructiva (HC) definida en el subepígrafe 2.3.1.

Tt : vector que contiene el tiempo de terminación del conjunto de máquinas

$t_1 = \text{randi}(n)$

$x = \text{round}(\text{rand})$

2. if $x == 1$

$m_1 = m_k | Tt_k = \max(Tt_k) \forall k \in M$

else

$m_1 = \text{randi}(m);$

```

end
3. x = round(rand);
   if x == 1
        $m_2 = m_i | Tt_i = \min(Tt_i) \forall i \in M$ 
                                   else
        $m_2 = \text{randi}(m)$ ;
   end
   if  $m_2 == m_1$  repetir el paso 3 else ir al paso 4
4. Hacer j = randi (n)
5. if sch_neighb ( $m_1, t_1$ ) == PM
   if sch_neighb ( $m_2, j$ )  $\neq$  PM y j == 1
   Hacer d = round (rand)
   if d == 1 insertar la actividad de PM en la celda ( $m_2, j$ ) de sch_neighb y actualizar la posición de
       los restantes elementos de la fila  $m_2$  desplazando cada uno una celda a la derecha.
   end
end
   if sch_neighb ( $m_2, j$ )  $\neq$  PM y j > 1 y sch_neighb ( $m_2, j-1$ )  $\neq$  PM
   Hacer d = round (rand)
   if d == 1 insertar la actividad de PM en la celda ( $m_2, j$ ) de sch_neighb y actualizar la
       posición de los restantes elementos de la fila  $m_2$  desplazando cada uno una celda a
       la derecha.
   end
end
   if sch_neighb ( $m_2, j$ )  $\neq$  PM y j > 1 y sch_neighb ( $m_2, j+1$ )  $\neq$  PM y j < n
   Hacer d = round (rand)
   if d == 1 insertar la actividad de PM en la celda ( $m_2, j$ ) de sch_neighb y desplazar una
       celda a la derecha los restantes valores de la fila  $m_2$ .
   end
end
5.1 Eliminar la actividad de PM de la celda sch_neighb ( $m_1, t_1$ ) y actualizar las posiciones de
    los elementos de la fila  $m_1$  desplazando una posición (celda) a la izquierda los elementos
    ubicados en las celdas ( $m_1, u$ )  $\forall u > t_1$ .
else
    Hacer d = round (rand).
if d == 1 insertar la actividad de PM luego del último trabajo de la fila  $m_2$  y en la siguiente
    posición el trabajo ubicado en la celda sch_neighb ( $m_1, t_1$ )
else
    insertar el trabajo ubicado en la celda sch_neighb ( $m_1, t_1$ ) como último trabajo de la fila
     $m_2$ 
end
5.2 Actualizar las posiciones de la fila  $m_1$  cuidando que no queden 2 actividades de PM en
    posiciones adyacentes. De ocurrir esto eliminar una de estas.
end

```

2.3.2.2 Función de recocido (*schedules generation*)

Esta función toma como entrada la solución obtenida por *neighborhood function*, expresada en la matriz sch_neighb, y efectúa cambios sucesivos sobre dicha solución (determinados conforme a la estrategia definida en *neighborhood function*) en una cantidad definida por el parámetro temperatura. La solución obtenida constituye, igualmente, una solución vecina, pero con una estructura diferente a la provista por *neighborhood function*. El pseudocódigo que define a esta función se muestra a continuación:

```

Hacer sch_neighb = optimValues.x;
for i = 1:floor (optimValues.temperature) + 1
    [mach orders] = size(sch_neighb);
    sch_neighb = neighbor (schvecina, mach, orders);
end

```

2.3.2.3 Función objetivo (*fitness function*)

La función objetivo (*fitness function*) considera las premisas bajo las cuáles se diseñó HC, fundamentalmente en lo que refiere a la adopción de una política de mantenimiento preventivo

MODELO DE PONDERACIÓN DE ATRIBUTOS BASADO EN UN NUEVO ÍNDICE DE INCONSISTENCIA

de tipo *as good as new* y la variable tiempo entre fallos modelada mediante una distribución Weibull. El pseudocódigo correspondiente se muestra a continuación:

Inicializar $I = TE_{lk} = \text{zeros}(1, m)$, $a_{k(lk^-)} = a_{k(lk)} = TCE_{lk} = \text{zeros}(m, n)$

for $k = 1:m$

for $l = 1:n$

if $l == 1$ y $\text{sch_neighb}(k, l) \neq \text{PM}$

$I_k = I_k + 1$,

$a_{k(lk^-)} = a_{0k}$

$a_{k(lk)} = a_{k(lk^-)} + p_l$

$TE_{lk} = TE_{lkPM0}$

end

if $l > 1$ y $\text{sch_neighb}(k, l) \neq \text{PM}$ y $\text{sch_neighb}(m_2, j-1) == \text{PM}$

$I_k = I_k + 1$

$a_{k(lk^-)} = 0$

$a_{k(lk)} = a_{k(lk^-)} + p_l$

$TE_{lk} = TE_{lkPM1}$

end

if $l > 1$ y $\text{sch_neighb}(k, l) \neq \text{PM}$ y $\text{sch_neighb}(m_2, j-1) \neq \text{PM}$

$I_k = I_k + 1$

$a_{k(lk^-)} = a_{k(lk-1)}$

$a_{k(lk)} = a_{k(lk^-)} + p_l$

$TE_{lk} = TE_{lkPM0}$

end

$TCE_{lk} = \text{máx}(Tt_k; r_l) + TE_{lk}$

$Tt_k = TCE_{lk}$

end

end

$C_{\text{máx}} = \text{máx}(Tt_k)$

II. RESULTADOS

EL algoritmo fue aplicado en un taller de la industria del plástico, integrado por 7 máquinas de inyección dispuestas en paralelo. El sistema productivo reúne las características definidas en el epígrafe 2.1. El tiempo de fabricación de los lotes de productos y la fecha de lanzamiento estimada para cada uno a partir del inicio del periodo de programación se muestran en la Tabla 1.

Tabla 1. Tiempo de producción y fecha de lanzamiento de los trabajos a programar

Productos	Tiempo de	Tamaño del	Tiempo de	Fecha de
-----------	-----------	------------	-----------	----------

	elaboración (h/u)	lote (u)	producción (h)	lanzamiento (h)
1. Cubeta 4 lts	0,0193	14600	281,78	0
2. Cubeta 10 lts	0,0196	15760	308,90	24
3. Cubeta 19 lts	0,0210	10650	223,65	0
4. Tapa de teclado VE78	0,0073	1120	8,18	240
5. Asa inferior de la olla	0,0064	31586	202,15	0
6. Decoración del mango	0,0071	14300	101,53	0
7. Galón cónico	0,0155	13865	214,91	120
8. Cubierta de placa 2P	0,0017	113000	192,10	0
9. Pomo 1 lt	0,0135	21610	291,74	0
10. Recipiente de 4 lts	0,0137	18350	251,40	72
11. Recipiente medidor	0,0153	23740	363,22	0
12. Base del tomacorriente	0,0014	139250	194,95	0
13. Hélice 12' Repuesto	0,0195	22950	447,53	48
14. Botón de ajuste	0,0058	10400	60,32	72
15. Botón oscilador	0,0029	15980	46,34	240
16. Caja 2x4 vivienda	0,0072	102740	739,73	480
17. Asa superior de la olla	0,0064	31293	200,28	0
18. Caja 4x4 vivienda	0,0072	43857	315,77	0
19. Tapa lateral	0,0055	33084	181,96	48
20. Hélice VE 79	0,0229	19056	436,38	72
21. Columna base VE 79	0,0207	30694	635,37	288
22. Tuerca protectora	0,0043	17315	74,45	0
23. Base VE 79	0,0156	7000	109,20	288
24. Tapa base VE 78	0,0173	8550	147,92	120
25. Base VE78	0,0231	309	7,14	0
26. Hélice 16	0,0229	1115	25,53	0
27. Gaveta para vegetales	0,0229	2540	58,17	360
28. Panel frontal	0,0172	2200	37,84	0
29. Marco congelador	0,0229	2715	62,17	0
30. Panel superior	0,0533	1540	82,08	120
31. Anillo protector VE 79	0,0110	15630	171,93	168
32. Placa soporte 2 puestos	0,0069	37950	261,86	168

Para cada una de las máquinas se tomaron datos de las variables tiempo entre fallos (TBF) y tiempo de inactividad debido a fallos (TTR) expresados en horas (más de 30 datos por cada máquina). Para ambas variables las pruebas de bondad de ajuste realizadas (Kolmogorov – Smirnov (KS), Anderson – Darling (AD) y Chi cuadrado) reafirmaron la hipótesis de que los datos ajustaban a la distribución Weibull con un 5% de significación.

La Tabla 2 contiene la información, para cada máquina, referida a los parámetros de forma y escala de la distribución Weibull para las variables:

TBF

TTR

el tiempo medio dedicado a mantenimiento preventivo ($\bar{t}_p$)

el tiempo medio de la reparación del fallo ($\bar{t}_r$), la edad efectiva de cada máquina al inicio del periodo de programación (E_0)

MODELO DE PONDERACIÓN DE ATRIBUTOS BASADO EN UN NUEVO ÍNDICE DE INCONSISTENCIA

el tiempo entre intervenciones preventivas (t_i) según la política mantenimiento adoptada en la empresa, y el tiempo óptimo entre intervenciones preventivas (t_i^*), expresados en horas. El tiempo medio de reparación de cada máquina fue obtenido a partir de la ecuación 9 empleada por Mora Gutiérrez (2012) [20]:

$$\bar{t}_r = \eta * \Gamma(1 + \frac{1}{\beta})(9)$$

Donde:

$\bar{t}_r$: Tiempo medio de mantenimiento correctivo

η : Parámetro de escala de la distribución Weibull correspondiente a la variable TTR

β : Parámetro de forma de la distribución Weibull correspondiente a la variable TTR

Tabla 2. Parámetros de mantenimiento del equipamiento productivo

No.	Máquinas	β_{TBF}	$\eta_{TBF}(h)$	$\bar{t}_p(h)$	β_{TTR}	$\eta_{TTR}(h)$	$\bar{t}_r(h)$	$\bar{t}_i(h)$	$t_i^*(h)$	$E_0(h)$
1	TTI-130-1	2,06	585,97	33,45	1,42	127,62	116,06	400	311,41	332
2	TTI-130-2	2,20	911,78	33,45	1,41	135,72	123,56	400	463,43	292
3	TTI-300	1,87	397,64	33,45	1,29	61,43	56,82	400	322,69	168
4	IJ-300	2,03	1318,53	35,54	1,42	73,99	67,29	550	948,86	480
5	EM-300	5,67	2002,27	27,29	1,53	22,42	20,19	385	1608,94	288
6	EM-480	1,63	952,41	27,29	1,29	36,64	33,89	385	1107,18	316
7	JM-650	2,59	1454,74	27,29	1,36	97,57	89,35	425	515,45	412

Los valores del tiempo entre intervenciones preventivas $\bar{t}_i$ para cada máquina fueron obtenidos a partir del plan de mantenimiento preventivo establecido en la empresa. El tiempo óptimo entre intervenciones preventivas (t_i^*) fue calculado siguiendo el criterio de maximizar la disponibilidad del equipamiento, considerando que la variable tiempo entre fallos en todos los casos muestra un ajuste adecuado a la distribución Weibull. En este sentido se utilizó la expresión 10 planteada por Cassady y Kutanoglu (2003) y Pan et al (2010) [21; 22].

$$t_i^* = \eta * \left(\frac{\bar{t}_p}{\bar{t}_r * (\beta - 1)} \right)^{\frac{1}{\beta}} \quad (10)$$

Resultados obtenidos

En un primer momento se muestra la solución obtenida mediante la heurística constructiva (HC) que constituye la fase 1 del método híbrido HCSA. Posteriormente se muestran los resultados alcanzados mediante la aplicación de la fase 2.

3.2.1 Solución obtenida mediante la heurística constructiva (HC) de la fase 1

En la matriz Schedule se muestra la solución proporcionada por HC. Las filas corresponden a cada máquina, en el orden en que aparecen en la Tabla 2. Los valores se corresponden con el id de cada trabajo, en el orden en que aparecen en la tabla 1. El valor 100 indica una actividad de PM en la máquina correspondiente a la fila. Por ejemplo, de acuerdo con esta solución el programa integrado para la máquina 1 (máquina TTI – 130 – 1) iniciaría con una intervención de PM, luego se elaborarían los productos 17 y 22 y, seguidamente, el 32, el 8 y el 23, ejecutándose una intervención de PM antes de iniciar cada uno de estos 3 últimos productos. En este caso el valor del parámetro ω utilizado en el paso 4 de HC se ajustó a 0,005.

La matriz TTj muestra el id de cada trabajo en la fila 1, en el orden en que fueron asignados a las máquinas, y en las filas 2 y 3 los instantes de inicio y de completamiento esperados, respectivamente, para cada trabajo. Según puede apreciarse el valor de la función objetivo (Cmax) es de 1222,3 h., determinado por la máquina 5 tras concluir la elaboración del producto 16.

Schedule = Columns 1 through 9

100	17	22	100	32	100	8	100	23
3	12	100	2	100	19	100	30	0
5	6	100	10	100	7	14	0	0
1	29	20	31	15	0	0	0	0
11	16	0	0	0	0	0	0	0
9	28	25	21	4	0	0	0	0
18	26	100	13	24	27	0	0	0

TTj = 1.0e+03 *

Columns 1 through 13									
0.0110	0.0180	0.0090	0.0010	0.0030	0.0050	0.0170	0.0120	0.0060	0.0220
0.0290	0.0280	0.0260							
0	0	0	0	0	0	0	0.0335	0.2488	0.2405
0.2464	0.2952	0.3024	0.3272						
0.3633	0.3272	0.3024	0.2952	0.2488	0.2405	0.2464	0.4799	0.3705	0.3504
0.3612	0.3419	0.3541							
Columns 14 through 26									
0.0250	0.0160	0.0210	0.0130	0.0200	0.0020	0.0320	0.0100	0.0070	0.0080
0.0190	0.0310	0.0240							
0.3419	0.4800	0.4800	0.4800	0.4800	0.4800	0.4800	0.4800	0.7890	0.7974
0.8338	0.9519	0.9590							
0.3494	1.2223	1.1525	0.9590	0.9519	0.8338	0.7974	0.7890	1.0553	1.0346
1.0527	1.1420	1.1260							
Columns 27 through 32									
0.0230	0.0300	0.0140	0.0270	0.0150	0.0040				
1.0346	1.0527	1.0553	1.1260	1.1420	1.1525				
1.1809	1.1689	1.1506	1.1927	1.1937	1.1613				

3.2.2 Resultados obtenidos mediante la aplicación de la fase 2 del método híbrido (HCSA)

Se tomará la solución provista por HC como solución de partida para el algoritmo de recocido simulado definido en el epígrafe 2.3.2. La parametrización se definió de la manera siguiente: temperatura inicial igual a 1000, un valor relativamente alto que en las primeras iteraciones permite aceptar soluciones peores que la inicial con alta probabilidad. Se amplía con ello el espacio de búsqueda; definición del entorno de búsqueda mediante las funciones *neighbor function* y *schedules generation*. Condición de equilibrio definida ajustando a 100 el *reannealing interval*; y esquema de enfriamiento exponencial. Con $\alpha = 0,95$, para lograr un descenso de la temperatura más rápido a altas temperaturas y más lento cuando la temperatura es menor; criterio de parada correspondiente a 5000 iteraciones. El resultado obtenido luego de 5000 iteraciones se muestra a continuación.

```

100 18100 21250 0 0
5 100 8 1329240 0
10100 329 4 260 0
311 176 230 0 0
100 19160 0 0 0 0
123 27222015100 30
7 2 1128140 0 0

```

El valor del *makespan* ($C_{\max}$) logró reducirse a 1219,98 h, apreciándose una reducción considerable en el número de intervenciones de PM necesarias. La matriz TTj correspondiente a la solución anterior se muestra a continuación.

TTj = 1.0e+03 * h

Columns 1 through 13									
0.0180	0.0210	0.0250	0.0050	0.0080	0.0130	0.0290	0.0240	0.0100	0.0320
0.0090	0.0040	0.0260							
0.0335	0.4151	1.1876	0	0.2576	0.4537	0.9539	1.0289	0.0720	0.4083
0.6962	1.0674	1.0785							
0.3817	1.1876	1.1980	0.2242	0.4537	0.9539	1.0289	1.2131	0.3748	0.6962
1.0674	1.0785	1.1134							
Columns 14 through 26									
0.0310	0.0010	0.0170	0.0060	0.0230	0.0190	0.0160	0.0120	0.0030	0.0270
0.0220	0.0200	0.0150							
0.1680	0.3474	0.6465	0.8629	0.9738	0.0480	0.4800	0	0.2016	0.4352
0.4963	0.5747	1.0385							
0.3474	0.6465	0.8629	0.9738	1.0941	0.2300	1.2200	0.2016	0.4352	0.4963
0.5747	1.0385	1.0882							
Columns 25 through 32									
0.0300	0.0070	0.0020	0.0110	0.0280	0.0140				
1.1155	0.1200	0.3416	0.6689	1.0703	1.1133				
1.1982	0.3416	0.6689	1.0703	1.1133	1.1823				

En la figura 1 se muestra la evolución del algoritmo en base al resultado de la función objetivo cada 250 iteraciones, en la que puede verse el mejor y el último valor de Cmax durante toda la corrida.

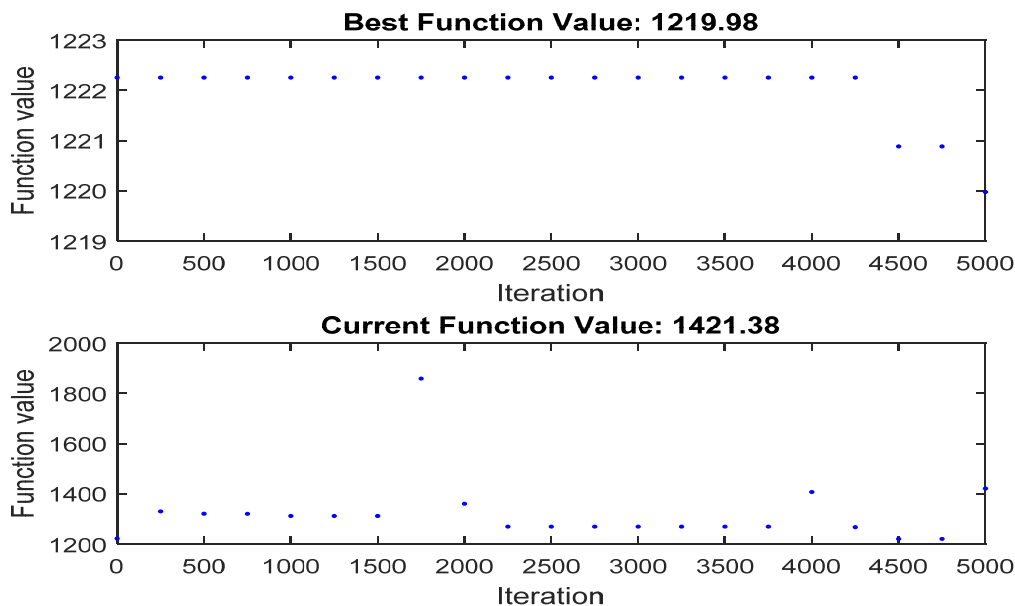


Fig. 1. Evolución del algoritmo HCSA en una corrida de 5000 iteraciones

IV.DISCUSIÓN

La estructura de la solución obtenida en la fase 1 del algoritmo, al aplicar la heurística HC, establece una correspondencia adecuada entre la frecuencia de intervenciones de MP para cada máquina y los valores de la variable t_i^* que se obtienen al optimizar la misma a partir de la ecuación 10, según se aprecia en la Tabla 2. Esto demuestra la efectividad de la heurística *greedy* que se expresa mediante la ecuación 2 del algoritmo, con lo que se logra una política de mantenimiento preventivo más racional que la adoptada en la empresa objeto de estudio. Sin embargo, debe considerarse que este resultado se obtuvo al considerar una política de mantenimiento del tipo *as good as new*, por lo que en investigaciones futuras debe trabajarse en lo que respecta al uso de una política que modele mejor el contexto real. La regla para la asignación provista por HC, para este caso de estudio, se asocia al esquema de lo que se denomina en la literatura un programa no retrasado, o sea, aquel en el que los trabajos se programan tan pronto como sea posible. No obstante, este esquema no siempre produce una solución óptima, de hecho, en la solución final obtenida mediante la fase 2 puede verse que los trabajos 1 y 17 son retrasados para dar prioridad al 31. Aunque la fase 2 no logró una reducción significativa del C_{max} , respecto al valor obtenido mediante HC, el número de intervenciones preventivas se redujo en 4, lo cual se traduce en una disminución del costo total y, si bien esta medida no fue el objetivo a optimizar, es un elemento importante dentro del proceso de toma de decisiones en el orden práctico, fundamentalmente en el contexto actual en el que se desempeña la empresa estatal cubana. Estas cuestiones reafirman la pertinencia del desarrollo de un programa integrado producción – mantenimiento, sobre la base métodos cuantitativos de la Investigación de Operaciones y la Inteligencia Artificial, como mecanismo para potenciar las capacidades de operaciones y la eficiencia empresarial.

V.CONCLUSIONES

1. Mediante la combinación de una heurística constructiva inspirada en la regla de despacho LPT y un enfoque de recocido simulado se desarrolló un algoritmo híbrido para generar programas integrados de producción y mantenimiento preventivo en ambientes productivos caracterizados por máquinas paralelas idénticas.
2. El algoritmo propuesto supera a otros enfoques encontrados en la literatura para abordar este tipo de problema al considerar posibles diferencias en las fechas de lanzamiento de varios productos. Se concibe el instante de inicio de las intervenciones de mantenimiento preventivo como una variable de decisión y no como un parámetro fijo y conocido previamente. Estos elementos realzan el valor práctico de nuestra propuesta, que modela situaciones frecuentes

en la práctica empresarial, que otras propuestas han considerado restricciones, lo que limita la capacidad de reproducir adecuadamente el contexto de manufactura.

3. Investigaciones futuras relacionadas con esta temática pudieran considerar otras medidas además del *makespan*, tales como el consumo de la energía, el nivel de riesgo ambiental y el costo, dando lugar a la adopción de un enfoque multicriterio capaz de aportar soluciones con un carácter más integral. Por otro lado, resultaría de interés extender este estudio a talleres de manufactura con otras configuraciones productivas tales como el *job shopflexible*.

VI. REFERENCIAS

1. Zahedi Z, Salim A. Integrating Preventive Maintenance Scheduling as Probability Machine Failure and Batch Production Scheduling. Computer, Mathematics and Engineering Applications. 2017;7(2):105-12. ISSN 2476-907X. DOI
2. Jing Z, Hua J, Zhu Y, 2017. Multi-objective Integrated Optimization Problem of Preventive Maintenance Planning and Flexible Job-Shop Scheduling. 2017 En: Proceedings of the 23rd International Conference on Industrial Engineering and Engineering Management. Sl.: Atlantis Press. p. ISBN 978-94-6239-255-7. DOI 10.2991/978-94-6239-255-7_25.
3. Boudjelida A. On the robustness of joint production and maintenance scheduling in presence of uncertainties. Journal of Intelligent Manufacturing. 2017 (1):1-16. ISSN 1572-8145.
4. Chansombat S, Pongcharoen P, Hicks C. A mixed-integer linear programming model for integrated production and preventive maintenance scheduling in the capital goods industry. International Journal of Production Research. 2018 (1). ISSN 1366-588X. DOI 10.1080/00207543.2018.1459923.
5. Pinedo ML. Scheduling. Theory, Algorithms, and Systems. 4. ed. New York, Estados Unidos Springer; 2012. ISBN 978-1-4614-2361-4. DOI 10.1007/978-1-4614-2361-4.
6. Coffman EG, Garey MR, Johnson DS. An application of Bin-Packing to multiprocessor scheduling. ,SIAM J COMPUT. 1978;7(1):1-17. ISSN 1095-7111.
7. Anily S, Bramel J, Simchi-Levi D. Worst-case analysis of heuristics for the Bin Packing Problem with general cost structures. Operations Research. 1992;42(2):287-98. ISSN 1526-5463.
8. Johnson DS, Demers A, Ullman JD, et al. Worst-case performance bounds for simple one-dimensional packing algorithms. SIAM J Comput. 1974;3(4):299-325. ISSN 1095-7111. DOI
9. Baker KR. Introduction to Sequencing and Scheduling. New York, Estados Unidos: John Wiley and Sons; 1974. ISBN 978-0853122999.
10. Carlier J. Scheduling jobs with release dates and tails on identical machines to minimize the makepan. European Journal of Operational Research. 1987 (29):298-306. ISSN 0377-2217. DOI
11. Gharbi A, Haouari M. Minimizing makespan on parallel machines subject to release dates and delivery times. Journal of scheduling. 2002 (5):329-55. ISSN 1099-1425. DOI 10.1002/jos.103.
12. Lee CY. Parallel machines scheduling with non-simultaneous machine available time. Discrete Appl Math. 1991 (30):53-61. ISSN 0166-218X.
13. Lee C, et al. Machine scheduling with an availability constraint. Journal of Global Optimization. 1996 (9):395-416. ISSN 1573-2916.
14. Lee C, et al. A note on "parallel machine scheduling with non-simultaneous machine available time". Discrete Applied Mathematics. 2000 (100):133-5. ISSN 0166-218X.
15. Xu D, Sun K, Li H. Parallel machine scheduling with almost periodic maintenance and non-preemptive jobs to minimize makespan. Computers & Operations Research. 2008 (35):1344 – 9. ISSN 0305-0548.
16. Lee W. Multi machine scheduling with deteriorating jobs and scheduled maintenance. Applied Mathematical Modelling. 2008 (32):362-73. ISSN 0307-904X.
17. Mirabedini S, N, Iranmanesh H. A scheduling model for serial jobs on parallel machines with different preventive maintenance. Int J Adv Manuf Technol. 2014 (70): 1579–89. ISSN 1433-3015.
18. Wang S, Liu M. Multi-objective optimization of parallel machine scheduling integrated with multi-resources preventive maintenance planning. Journal of Manufacturing Systems. 2015 (37):182-92. ISSN 0278-6125.
19. Da W, Feng H, Pan E. Integrated preventive maintenance and production scheduling optimization on uniform parallel machines with deterioration effect. 2016 En: Proceedings of the 2016 Estados Unidos. IEEM. p. 951-5. ISBN 978-1-5090-36. DOI 978-1-5090-3665-3/16/\$31.00 [Citado Disponible en:
20. Mora Gutiérrez A. Mantenimiento Industrial Efectivo. 2 ed. Medellín, Colombia: AMG; 2012. ISBN 978-958-98902.
21. Cassady CR, Kutanoglu E. Minimizing job tardiness using integrated preventive maintenance planning and production scheduling. IIE Transactions. 2003 (35):503-13. ISSN 1573-9724.
22. Pan E, W L, Xi L. Single-machine-based production scheduling model integrated preventive maintenance planning. Int J Adv Manuf Technol. 2010 (50):365-75. ISSN 1433-3015.